Turgen System
version 8.6.14 - "Sampling Rate"

Contents

Part I

Turgen System

1 Introduction

1.1 Mission and Features

TURGEN SYSTEM is a versatile utility that allows you to:

- Create your own tapes with software for Atari 8-bit computers.
- Transfer data from your PC or Mac to your Atari 8-bit computer using just a data recorder and a cassette adapter.

1.1.1 Primary Features

- TURGEN SYSTEM converts files to various turbo systems and also to standard tape records.
- TURGEN SYSTEM outputs WAVE files, tape images, or sends signal directly to the sound card.
- TURGEN SYSTEM provides special support for conversion of segmented binary files. This special support includes a Wizard for binary files and a set of special binary loaders.

1.1.2 Auxiliary Features

- Processing of tape images
- Facilities for automated production of multiple tapes - batch processing, automatic file name creation, and prefixing WAVE file names with tape side numbers
- Turbo decoder that can be used to retrieve information from tapes
- Tape image extractor tool that extracts data from tape images
- Tool for embedding tokenized BASIC files to binary files
- Tool for merging segments of binary files

1.1.3 Support for Standard Tape Records

Standard tape records and unmodified data recorders are fully supported. Tape images with baud, data, and fsk chunks can be read by the Tape image plugin. Monolithic binary files, segmented binary files, BASIC files, and plain data files can be converted to standard tape records by the *Standard* plugin.

1.2 Hardware that Can Be Used

Data recorders with or without turbo upgrades, compact cassettes, data recorder and compact cassette replacement devices, MEGA-CD interface (CD-LINK), SIO2PC, ATART.

Important information. If you are using devices other than data recorder and compact cassettes, please read section 7.5 that contains vital information about how to use such devices. Then read section 3.8.

1.3 Copyright

1.4 System Requirements

- Operating system supported by Java SE Runtime Environment (JRE) or Java SE Development Kit (JDK), version 8 or newer.
  This includes, but might not be limited to: Microsoft Windows, GNU/Linux distributions, Solaris, macOS, OpenIndiana, FreeBSD.

1.5 Abbreviations and Glossary

Data recorder. Device designed to read or write signal to or from compact cassettes, specially designed to be connected to computers, for example Atari XC12 or Atari 1010.

Tape recorder. Consumer electronics device designed to read or write electric signal to or from compact cassettes.

Standard Tape Records. Records that the Atari 8-bit computer can read and write using a data recorder that hasn't been modified.

Turbo system. A system designed to speed up data transfer of the original Atari data recorder. A typical turbo system consist of three components:

1. Hardware modification of the data recorder (extra circuitry connected to the original PCB)
2. Software that allows for usage of the hardware modification (loaders, tape operating systems)
3. File format or multiple file formats

Program directory. Directory or folder where Turgen System is installed to. Special symbol `<TSDIR>` is also used to reference this directory.

Configuration directory. Directory or folder where Turgen System stores user configuration files. Special symbol `<CFGDIR>` is also used to reference this directory.

JRE. Java SE Runtime Environment.

JDK. Java SE Development Kit.

Java home directory. Directory or folder where JRE (or JDK) resides. A special symbol `<JAVAHOME>` is also used to reference this directory.

1.6 Similar Utilities

Turgen System provides similar functionality as the following utilities: A8CAS, a8cas-util, CAS2WAV, XEX2CAS 2.4, and ATART.

1.7 Files Processed by Turgen System

The most important types of files processed by Turgen System are *binary files*, *tape images*, and *WAVE files*. Please note that Turgen System does not support disk images or cartridge images.

1.7.1 Binary Files

Turgen System provides extensive support for processing of *Atari DOS 2 Binary Files*. These files will be referenced as *binary files*. A binary file is a file with precisely defined internal structure that is mainly used to store *programs* for Atari 8-bit computers. *Please note that the knowledge of the binary files is critical for successful usage of Turgen System.*

Typical extensions for binary files are the following: COM, XEX, OBJ.

Internal Structure. A binary file always starts with a two-byte **header**. Both bytes have a value of 255 (0xFF). A binary file contains **segments**. A segment is a block of data prefixed with a **segment header**. The segment header holds two 16-bit addresses (segment start address and segment end address). These two addresses determine the addresses the segment data will be loaded to. The segment header can be optionally prefixed with two bytes of value 255 (0xFF).

Special Vectors. When a binary file is being loaded, certain addresses have a special meaning. These addresses (special vectors) are described in table 1.

Addresses	Special Vector
736 - 737	**RUN vector.** When all segments of a binary file are loaded, the binary loader will perform an indirect jump (JMP) using this vector. This vector defines run address of the binary file.
738 - 739	**INIT vector.** Whenever a segment that changes this vector is loaded, the binary loader will perform a jump to subroutine (JSR) using this vector. INIT vectors allow to execute code while the binary file is being loaded.

Table 1: Vectors

Segment Types. Segments that do not change any of the special vectors are called **DATA** segments. Segments that only change both bytes of the INIT special vector are called **INIT** segments. Segments that only change both bytes of the RUN special vector are called **RUN** segments.

Binary File Types. **Segmented Binary File** or just **Binary File** is a binary file without any special limitations or restrictions.

Monolithic Binary File is a special case of a binary file. Monolithic binary file that consists of exactly one DATA segment and at most one RUN segment. The structure of such binary file is very simplistic. It is one block of data with optional run address.

Loading Binary Files. **Binary Loader** is a program, or routine (usually a component of an operating system) that loads and executes binary files. **Binary Load** is a name for the process of loading and running of a binary file. This process is performed by a binary loader.

1.7.2 Tape Images

A tape image is a computer file containing the contents and structure of data stored on a compact cassette. A tape image is usually perfectly replicating the structure and contents of data stored on compact cassette regardless the file format and operating system used to store the data.

Turgen System supports *CAS*, a binary file format originally introduced by Ernest R. Schreurs as the output format of his WAV2CAS utility and later extended by the A8CAS project. The CAS format is intended to store Atari cassettes efficiently. For more information, refer to http://a8cas.sourceforge.net/format-cas.html.

1.7.3 WAVE Files

Waveform Audio File Format (WAVE, or more commonly known as WAV due to its filename extension) is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs. The WAV file is an instance of a Resource Interchange File Format (RIFF) defined by IBM and Microsoft. For more information, refer to https://en.wikipedia.org/wiki/WAV.

2 Distribution and Installation

2.1 Installation Requirements

Turgen System requires Java SE Runtime Environment (JRE) or Java SE Development Kit (JDK) to be installed.

Before installing Turgen System, download and install the JRE or JDK. Download from the following addresses:

- https://www.java.com/en/download/
- https://www.oracle.com/java/technologies/java-se.html

It is recommended to download and install the newest available version of JRE or JDK for your operating system.

2.2 Distribution and Source Code

Turgen System is distributed in two packages: **Installation package for Microsoft Windows** and **Operating system independent binary package**.

Source code is available from a GIT repository. For more information, refer to the project page at SourceForge.

2.3 Installation and Directories

To install Turgen System, decompress the operating system independent binary package or run the installation package to create the program directory and populate it with files.

The program directory contains all files required to run Turgen System. Turgen System requires read-only access to the files in the directory. The configuration directory is created by Turgen System when needed (usually when started for the first time). Its location is operating system dependent and Turgen System requires read-write access the directory in order to be fully functional.

3 Operations Guide

3.1 Starting Turgen System

Starting from Command Line

The program code of the Turgen System is located in the `turgen.jar` file. To run Turgen System, run the Java launcher and specify that the program code is in the `turgen.jar` file, using the following commands:

`<JAVAHOME>\bin\javaw.exe -jar <TSDIR>\turgen.jar` under Microsoft Windows or
`<JAVAHOME>/bin/java -jar <TSDIR>/turgen.jar` on Unix-like systems.

Microsoft Windows

Use the desktop or start menu shortcuts created by the installer or execute the `turgen.exe` launcher in the program directory.

Other Operating Systems

Start from command line or create launchers or shortcuts using your desktop environment

Command Line Parameters

One command line parameter is accepted - a file that stores a playlist.

Log File

Exceptions, traces or other important diagnostic information is stored into the `turgen.log` file located in the configuration directory. This file is recycled when it exceeds size of 4 MB.

3.2 Program Controls

Main Menu and Playlist

The main menu is located on the top of the program window. Almost all functions are accessible from the main menu.

In the middle of the program window, there is the *playlist* that stores information what files are to be converted and how they will be converted.

Main Control Buttons

Under the main menu, there is a toolbar, where the *main control buttons* are located. The most used functions can be invoked using these buttons. Every button has its icon and name. The buttons are depicted in table 2.

Icon	Name	Icon	Name
	Add item to playlist		Move 1 playlist item up
	Edit selected playlist item		Move 1 playlist item down
	Remove selected playlist items		Generate WAVE file
	Select all playlist items		Generate AUDIO directly
	Generate tape image		Wizard for binary files

Table 2: Main Control Buttons

Progress Monitoring Panel

This panel is located in the bottom part of the program window. Conversions of files can be monitored or stopped here. At the very bottom of the program window, there is a status bar.

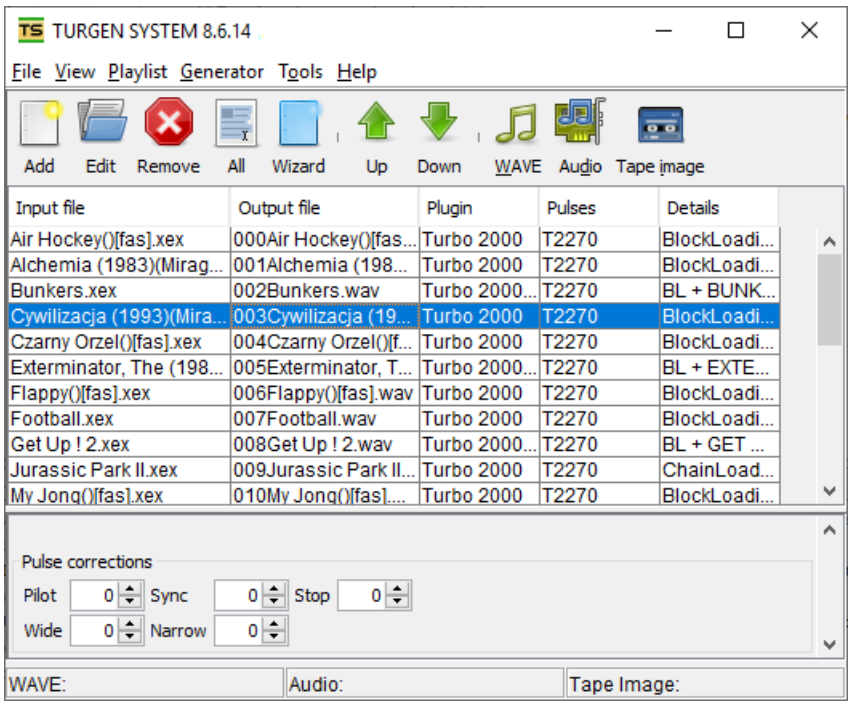

Figure 1: Program Window

3.3 Conversion of Files

3.3.1 Playlist Items

To convert a file, Turgen System requires information about the conversion that includes input file name, input and output format, output file name, what loader to prepend and various other parameters. That information is stored in the *playlist items*. The Playlist items are elements of the *playlist*.

3.3.2 Working with Playlist

To work with the playlist, use the main control buttons or menu items from the *Playlist* menu.

To add a playlist item to the playlist, click the *Add item to playlist* main control button. To edit playlist items, click the *Edit selected playlist item* main control button. To remove selected playlist items, click the *Remove selected playlist items* main control button.

To select playlist items, use your keyboard or your mouse. To select all playlist items, click the *Select all playlist items* main control button.

To move single playlist items use *Move 1 playlist item up* and *Move 1 playlist item down* buttons.

To load or save the playlist, use the *Load* and *Save* items from the *Playlist* menu.

3.3.3 Editing and Creating Playlist Items

Turgen System provides a dialog for creating and editing playlist items. The dialog is depicted in figure 2.

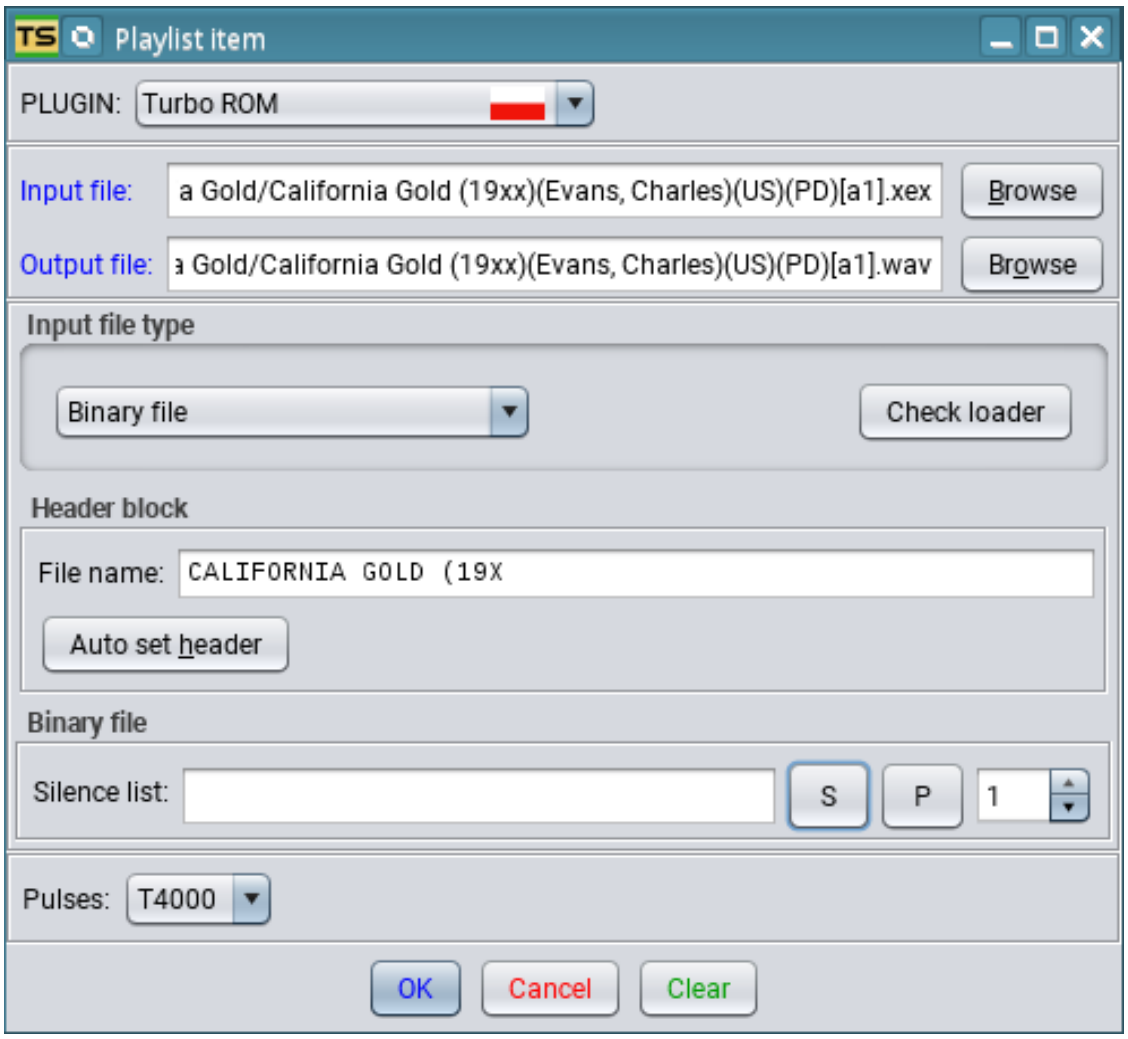

Figure 2: Manipulating Playlist Item

Use the *PLUGIN* combo box to select a plugin that will be used for conversion. Note the national flags that can help you distinguishing from where a particular turbo system comes from.

Use the common panel at the top of the dialog to choose the input and the output file.

To display a dialog with information about input file, click the *Input file* label. To populate the output file text field automatically, click the *Output file* label. To display a file chooser dialog, click the *Browse* buttons.

In the middle part of the dialog, there are controls specific for the selected plugin. These controls are described in part II.

Use the buttons at the bottom to commit or cancel the changes you have made to the playlist item. Click the *Clear* button to reset controls of the dialog.

3.3.4 Special Functions

To copy the playlist to the clipboard in the “comma separated values” file format, select the *Copy to clipboard as CSV* menu item.

To populate the playlist with items by processing a batch file, select the *Process batch file* menu item. For more information about batch processing, refer to section 6.4.

3.4 Wizard for Binary Files

The Wizard for binary files provides convenience when converting binary files.

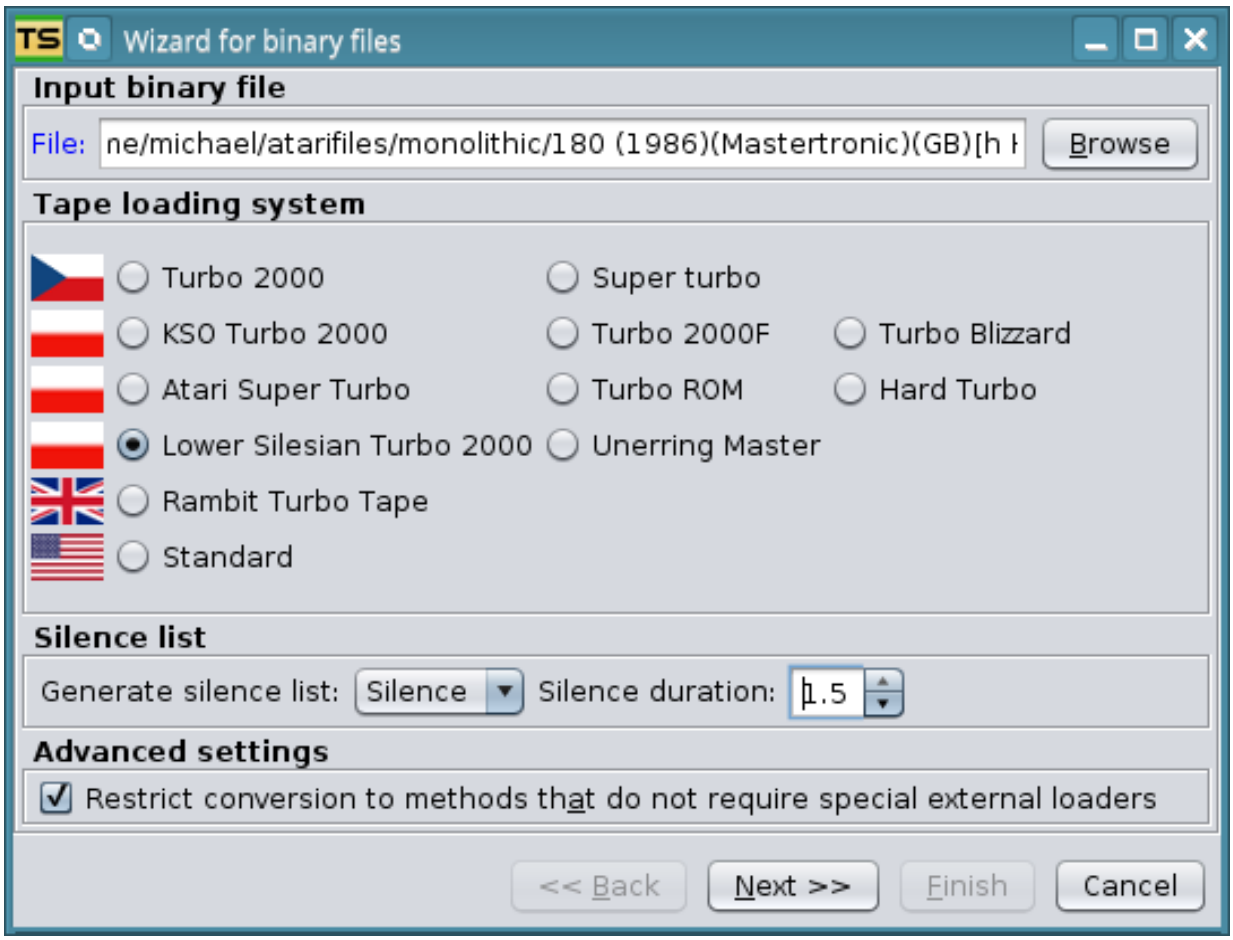

Figure 3: Wizard for Binary Files - Step 1

To start the wizard, click the *Wizard for binary files* main control button.

In the first step, specify a binary file that you want to convert and what are the capabilities of your data recorder. In the second step, choose one of the conversion methods offered by the wizard, output file name and pulses.

The offered conversion methods are ordered by the number of pilot tones or leaders. The wizard also checks (finitely of course) whether the binary file will destroy a built-in binary loader that will be prepended.

The wizard can also automatically generate silence lists (refer to section 3.8 for more information).

3.5 Output

3.5.1 Output of Electric Signal into WAVE File

To output electric signal to a WAVE file, select at least one playlist item and click the *Generate WAVE file* main control button. The WAVE files will be generated in one or more parallel tasks. To stop the generation, click the *Stop* buttons that will appear in the progress monitoring panel. The output files are overwritten without a warning.

3.5.2 Output of Electric Signal to Sound Card.

To output electric signal to the sound card, select at least one playlist item and click the *Generate AUDIO directly* main control button. Playlist items are processed sequentially. To stop the output, click the *Stop* and *Stop all* buttons that will appear in the progress monitoring panel. To resume an output that has been paused, click the *Resume* button.

You can press CTRL+SHIFT when clicking the *Generate AUDIO directly* main control button to output the electric signal in an infinite loop.

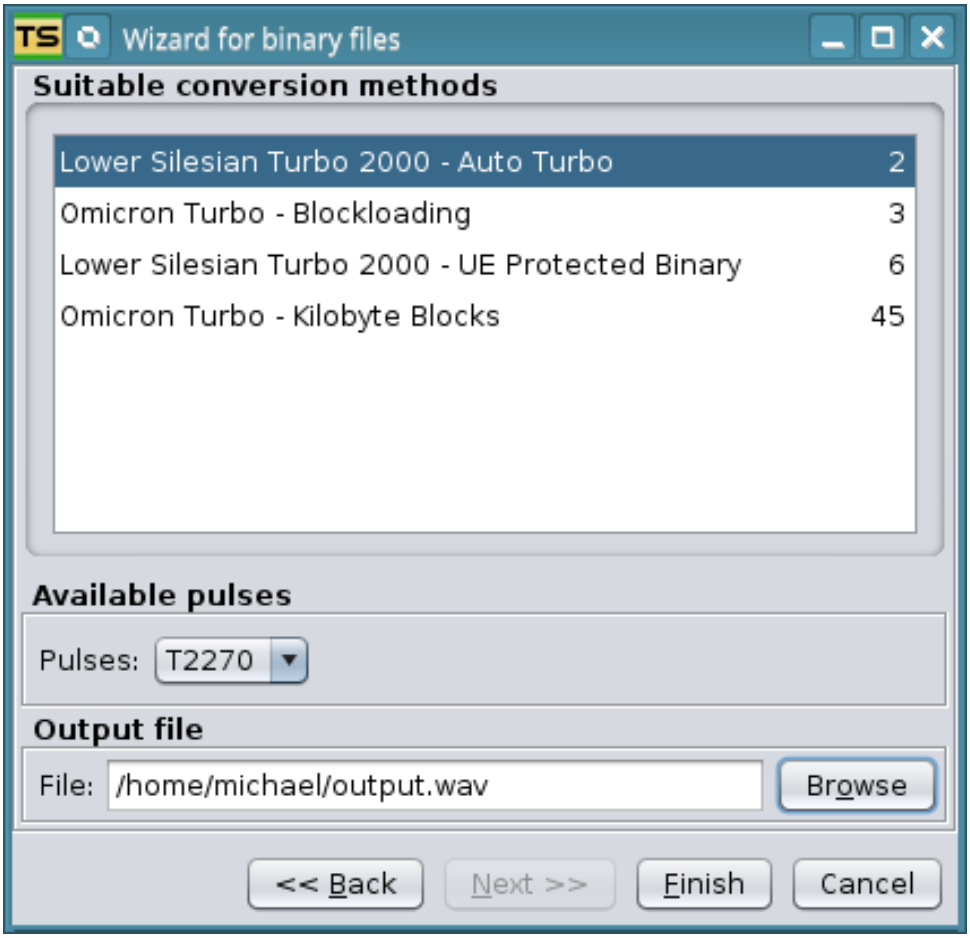

Figure 4: Wizard for Binary Files - Step 2

3.5.3 Output of Tape Image

To output a tape image, select at least one playlist item and click the *Generate tape image* main control button. Playlist items are processed sequentially. The output files are overwritten without a warning.

3.5.4 Waveforms

Turgen System allows you to choose shape and form of the electric signal. You can select waveforms described in the table.

Waveform	Description
Auto	Automatic selection. Pure sine wave is used for wider pulses, square wave is used for narrower pulses. This is the default.
Square	Ideal rectangular pulses
7th Harmonic	High precision approximation of rectangular pulses (Fourier series expansion, 7th harmonic)
5th Harmonic	Medium precision approximation of rectangulate pulses (Fourier series expansion, 5th harmonic)
3rd Harmonic	Low precision approximation of rectangulate pulses (Fourier series expansion, 3rd harmonic)
Pure Sine Wave	Harmonic pulses

Recommendations. Experimentation indicates that the pure sine wave is the best option for most types of recording equipment (as the distortion due to limited frequency ranges is negligible), except when very high transfer speeds (more than 4000 bd) are used.

Auto is the recommended and default setting. **Square wave** is recommended for high quality equipment (such as a Hi-Fi tape deck with minimum wow and flutter), high transfer speeds, and use with A8CAS-enhanced atari800 emulator or the Altirra emulator.

3.5.5 Output in Preview Mode

Output in the preview mode allows you to quickly check if your files can be converted and determine total duration of the generated signal. This is useful when you want to know if the converted files will fit to a tape side or how many tape sides will be needed.

To enable the Preview mode, select the *Preview mode* item from the *Generator menu*.

In the preview mode

- Playlist items are processed
- No output files are created
- No postprocessing is performed
- No signal is sent to the sound card
- Total duration of the signal or total size of the generated tape images is displayed in the status bar

3.5.6 Sampling Rate

For each output, you can select a sampling rate using the Program configuration facility. For output to the sound card and to the wave file, the sampling rate configuration entry determines the actual sampling rate. For tape image output, the sampling rate determines the smallest time unit used for pwm chunks in the tape image.

The default sampling rate is 44100 Hz. It is sufficient for most use cases.

Higher sampling rates (48000 Hz and 96000 Hz) are most useful for turbo systems using transfer speeds above 4000 bd, where more precision is required. Only some plugins take advantage of higher sampling rates, this is noted in the documentation for the plugins.

A lower sampling rate (22050 Hz) is available for usage with legacy hardware and software. In general, usage of this sampling rate is not recommended.

3.6 Input/Output matrix

Conversion capabilities of Turgen System are enumerated in the following table.

Input	Wave file	Sound card	Tape image with **pwmX** chunks
Regular file	YES	YES	YES
Binary file	YES	YES	YES
Tape image (**pwmX** chunks)	YES	YES	Passed through
Tape image (**baud, fsk** chunks)	YES	YES	Passed through

3.7 Pulse Corrections

3.7.1 Motivation

Turgen System has a repository of pulses for all supported turbo systems (see 5.2). The values in the repository have been tested using patched Atari800 emulator provided by the A8CAS project. However, users of Turgen System already reported minor problems when using real hardware. In light of this problems, an ability to comfortably correct the pulses is highly desired.

3.7.2 Permanent Pulse Corrections

To make permanent corrections of pulses, use the Program configuration facility. The plugins that support permanent pulse corrections are KSO Turbo 2000, Atari Super Turbo, Turbo ROM, and Turbo Blizzard.

3.7.3 Immediate Pulse Corrections

You can temporarily correct the pulses for all plugins. Select the *Immediate pulse corrections* item from the *View* menu. A special panel appears.

Use the controls on the special panel to adjust durations of the pulses. You can make the pulses narrower (negative values) or wider (positive values). One unit represents one audio signal sample.

Note that immediate pulse corrections always override any permanent corrections.

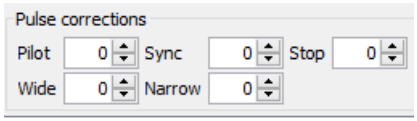

Figure 5: Pulse Corrections

3.8 Silence Lists

3.8.1 Overview

The silence lists allow comfortable usage of devices other than data recorders and compact cassettes (refer to section 7.5).

The silence lists allow the following:

- Insert silence (or other neutral signal) of defined duration after a block that contains one or more INIT segments
- Pause output of electric signal to the sound card after a block that contains one or more INIT segments

The silence lists are part of the playlist items (refer to section 3.3.1).

3.8.2 Silence List Syntax

A silence list consists of silence specifications (SPECs) separated by commas:
`SILENCE_LIST:=<SPEC>,<SPEC>,...,<SPEC>`

A silence specification is a number followed by optional P suffix or just a P suffix which is an equivalent of `1P` syntax:
`SPEC:=n[P]|P`

where n is the generated silence duration in seconds. The P suffix indicates that output of electric signal to the sound card will be paused. Tenths of seconds can be specified too.

3.8.3 Examples

Silence List	Explanation
`1,3,0,4.5`	1 second of silence after 1st INIT segment 3 seconds of silence after 2nd INIT segment No silence after 3rd INIT segment 4.5 seconds of silence after 4th INIT segment
`3.1,2P,P`	3.1 seconds of silence after 1st INIT segment 2 seconds of silence after 2nd INIT segment, output of electric signal to sound card will be paused 1 second of silence after 3rd INIT segment, output of electric signal to sound card will be paused

3.8.4 User Interface

To create a silence list, you can use the Silence list panels. Enter the silence specifications in the text field, or click the *S* or *P* buttons to populate the list automatically (the input binary file will be analyzed and a silence specification will be created for each INIT segment found). Use the spinner to set the duration of the automatically generated silence specifications.

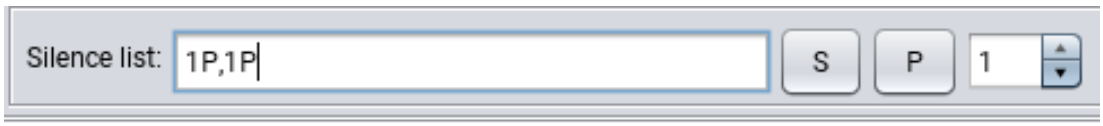

Figure 6: Silence List Panel

3.8.5 Cumulation of Silence

Some plugins allow to cumulate silence.

Cumulation can occur when multiple INIT vectors are stored in one data block.

- If the cumulation is enabled, duration of the resulting silence is calculated as addition of all silence specifications matching the INIT vectors stored in the affected data block. This is recommended for devices that ignore the MOTOR CTRL signal.
- If the cumulation is disabled, duration of the resulting silence is the longest of all silence specifications matching the INIT vectors stored in the affected data block. This is recommended for compact cassettes.

3.9 Automatic File Name Creation

Various turbo systems have different rules and limitations for file names. Turgen System provides configurable automatic file name creation facility that is used by selected plugins.

Each plugin that supports automatic file name creation provides a configuration entry that allows to configure the file name creation.

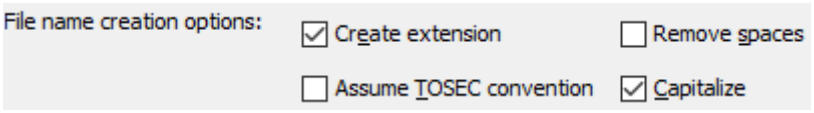

Figure 7: Automatic File Name Creation

- Select the *Create extension* check box to always generate file names with extension
- Select the *Remove spaces* check box to generate file names without spaces
- Select the *Capitalize* check box to generate capitalized file names
- Select the *Assume TOSEC naming convention* check box to create file names from only the first part of TOSEC file name (up to the first left parenthesis)

4 Program Configuration Facility

The Program configuration facility provides various program modules with configuration capabilities. Program configuration consists of *configuration entries* that are grouped in *configuration classes (or sections)*.

4.1 Changing Configuration

To change program configuration, select the *Preferences* item from the *Tools* menu. A dialog that allows to change the preferences will appear.

Click the *OK* button to save the changes. Click the *Cancel* button to cancel the changes. Click the *Defaults* button to reset the configuration entries to their default values for the currently selected configuration class.

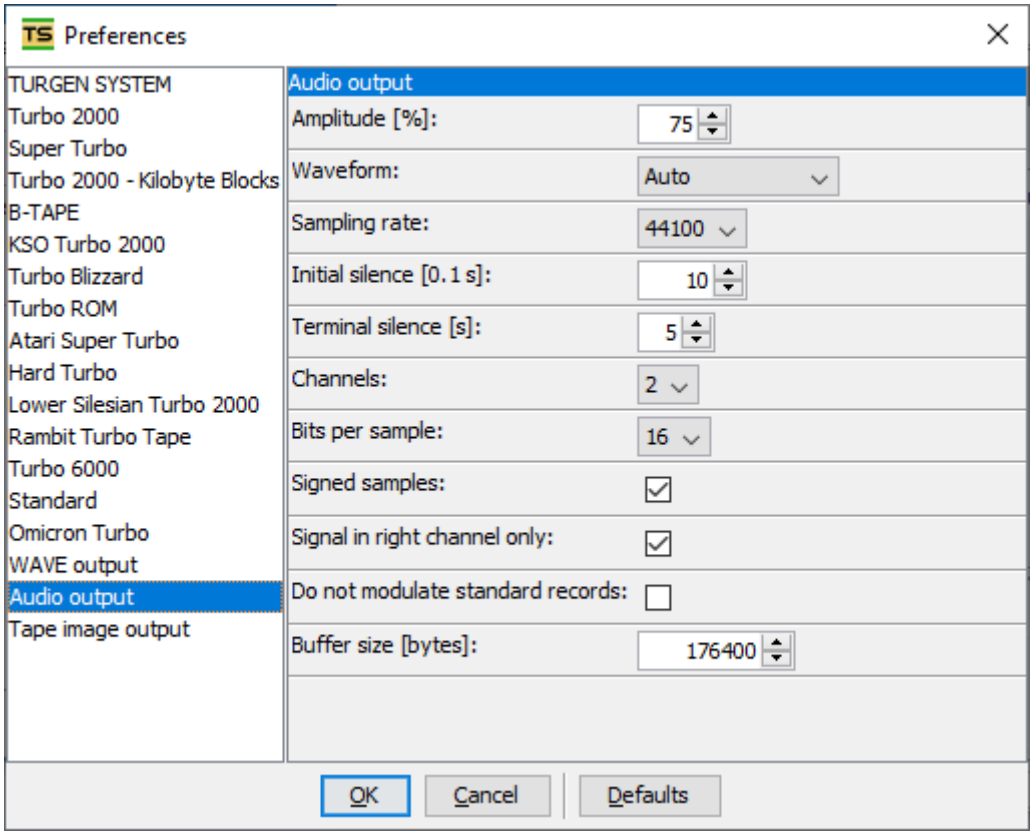

Figure 8: Preferences

4.2 General Configuration Entries

Turgen System/GUI Look and feel
Look and feel of the user interface, fully qualified class name. If the entry is empty, the default look and feel is used.

Standard looks and feels:
`javax.swing.plaf.metal.MetalLookAndFeel` (all platforms, default)
`com.sun.java.swing.plaf.motif.MotifLookAndFeel` (all platforms)
`com.sun.java.swing.plaf.windows.WindowsLookAndFeel` (Microsoft Windows only)
`com.sun.java.swing.plaf.gtk.GTKLookAndFeel` (only when GTK+ 2.2 or newer is available)
`com.sun.java.swing.plaf.mac.MacLookAndFeel` (only Mac, untested)
`com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel`

Turgen System/favorite output directory
Favorite directory for output files (wave files and tape images).

4.3 Output of Electric Signal to Sound Card

Audio output/Amplitude
Amplitude of the signal. 0-100%. It is not recommended to set the value too low.

Audio output/Waveform
This allows to select waveform. For more information, refer to section 3.5.4.

Audio output/Sampling rate

This allows to select sampling rate. Allowed values are 44100, 48000, and 96000 Hz. Note that only some plugins can take advantage of sampling rates higher than 44100 Hz.

Audio output/Initial silence
Duration of silence generated at the beginning of the output, one unit is 0.1 second.

Audio output/Terminal silence
Duration of silence generated at the end of the output. This allows to create gaps between programs.

Audio output/Channels
Number of channels.

Audio output/Bits per sample
Number of bits per one sample.

Audio output/Signed samples
Indicates whether to use signed samples.

Audio output/Signal in right channel only
If the number of channel is set to two, signal will be generated to the right channel only.

Audio output/Do not modulate standard records
If enabled, standard tape records will be generated without modulation. The resulting signal will be the same as it would be on the DATA-IN pin of the SIO port. This option should allow to pass standard tape records through turbo decoder circuitry.

Audio output/Buffer size
Size of the buffer for data being sent to the sound card. Turgen System can automatically adjust the selected value to match the sampling rate. Increase this value if there are clicks and pops in the audio output.

4.4 Output of Electric Signal to WAVE File

WAVE output/Amplitude
Amplitude of the signal. 0-100%. It is not recommended to set the value too low.

WAVE output/Waveform
This allows to select waveform. For more information, refer to section 3.5.4.

WAVE output/Sampling rate

This allows to select sampling rate. Allowed values are 44100, 48000, and 96000 Hz. Note that only some plugins can take advantage of sampling rates higher than 44100 Hz.

WAVE output/Initial silence
Duration of silence generated at the beginning of the WAVE file, one unit is 0.1 second.

WAVE output/Terminal silence
Duration of silence generated at the end of the WAVE file. This allows to create gaps between programs.

WAVE output/Channels
Number of channels.

WAVE output/Bits per sample
Number of bits per one sample.

WAVE output/Signal in right channel only
If the number of channel is set to two, signal will be generated to the right channel only.

WAVE output/Do not modulate standard records
If enabled, the standard tape records will be generated without modulation. The resulting signal will be the same as it would be on the DATA-IN pin of the SIO port. This option should allow to pass standard tape records through turbo decoder circuitry.

WAVE output/Change extension to .wav
Extension of output files will be changed to .wav in intelligent way if set to true.

WAVE output/Number output files
When selected, every output file name will be prefixed with a sequential number.

WAVE output/Prefix file names with tape side number
This allows to prefix wave file names with tape side number.
The wave file names are prefixed with s00_,s01_,... prefixes. File that wouldn't fit in one tape side are prefixed with "sovf_" prefix.

WAVE output/Tape side length
Defines tape side length in minutes. Used when prefixing wave file names with tape side numbers.
Set to 45 minutes for C-90 tape.

WAVE output/Tape side unused portion length
Defines unused portion of tape sides. Used when prefixing wave file names with tape side numbers. Set to 30-120 seconds to deal with varying tape lengths.

WAVE output/Maxiumum parallel tasks
Maximum number of parallel tasks that will generate wave files. Set to 0 for automatic determination (default, number of processors - 1), or specify 1-8 to set the number of tasks manually.

WAVE output/Postprocessing command
Command line for postprocessing. For more information refer to section 8.1.

4.5 Output of Tape Images

Tape image output/Sampling rate
This allows to select sampling rate. Allowed values are 44100, 48000, and 96000 Hz. The sampling rate is used for the pwm chunks to set the smallest time unit.

Tape image output/Change extension to .cas
Extension of output files will be changed to .cas in intelligent way if set to true.

Tape image output/Auto create temporary files
If the output file is not specified and this entry is set to true, temporary file is created automatically. This is useful for postprocessing.

Tape image output/Postprocessing command
Command line for postprocessing. For more information refer to section 8.2.

5 Advanced Settings

5.1 Active Plugins

The list of plugins that will be loaded is stored in the `plugins.list` file. To disable a plugin, remove or preferably comment-out (using the # character) the corresponding line. To change the order of the plugins (that is what users from Poland may want to do), change order of the lines.

Note: Some plugins allow to prepend a loader as a standard tape boot file. This function will not work when the Standard plugin is disabled.

5.2 Repository of Pulses

The repository of pulses is stored in `pulses/pulses.list` file. Repository can be modified, directions are present in the file itself as comments. It is not recommended to modify the repository without serious reasons. Changes take effect after Turgen System is restarted.

6 Tools

6.1 Turbo Decoder

6.1.1 Overview

The Turbo Decoder is a tool designed to retrieve data from tapes. The decoding algorithm is a simple Java rewrite of turbo loaders from assembler 6502.

The electric signal can be read from WAVE files. When creating WAVE files, it is recommended to set the tape recorder to get maximum signal amplitude. The WAVE file must be in the following format: PCM, 1 or 2 channels, 44100 - 96000 Hz, 8 or 16 bits per sample.

Another possibility is to read the electric signal directly from the sound card. This source of electric signal is supported, but not recommended.

Turbo systems supported by the decoder are the following: Turbo 2000, Turbo 2000 - kilobyte blocks, Super Turbo, Turbo Tape and B-TAPE, KSO Turbo 2000, Turbo Blizzard, Turbo ROM, Atari Super Turbo (AST format only), Hard Turbo, and Lower Silesian Turbo 2000.

The decoder can also work in so called “Monitor mode” that allows you to read raw turbo blocks.

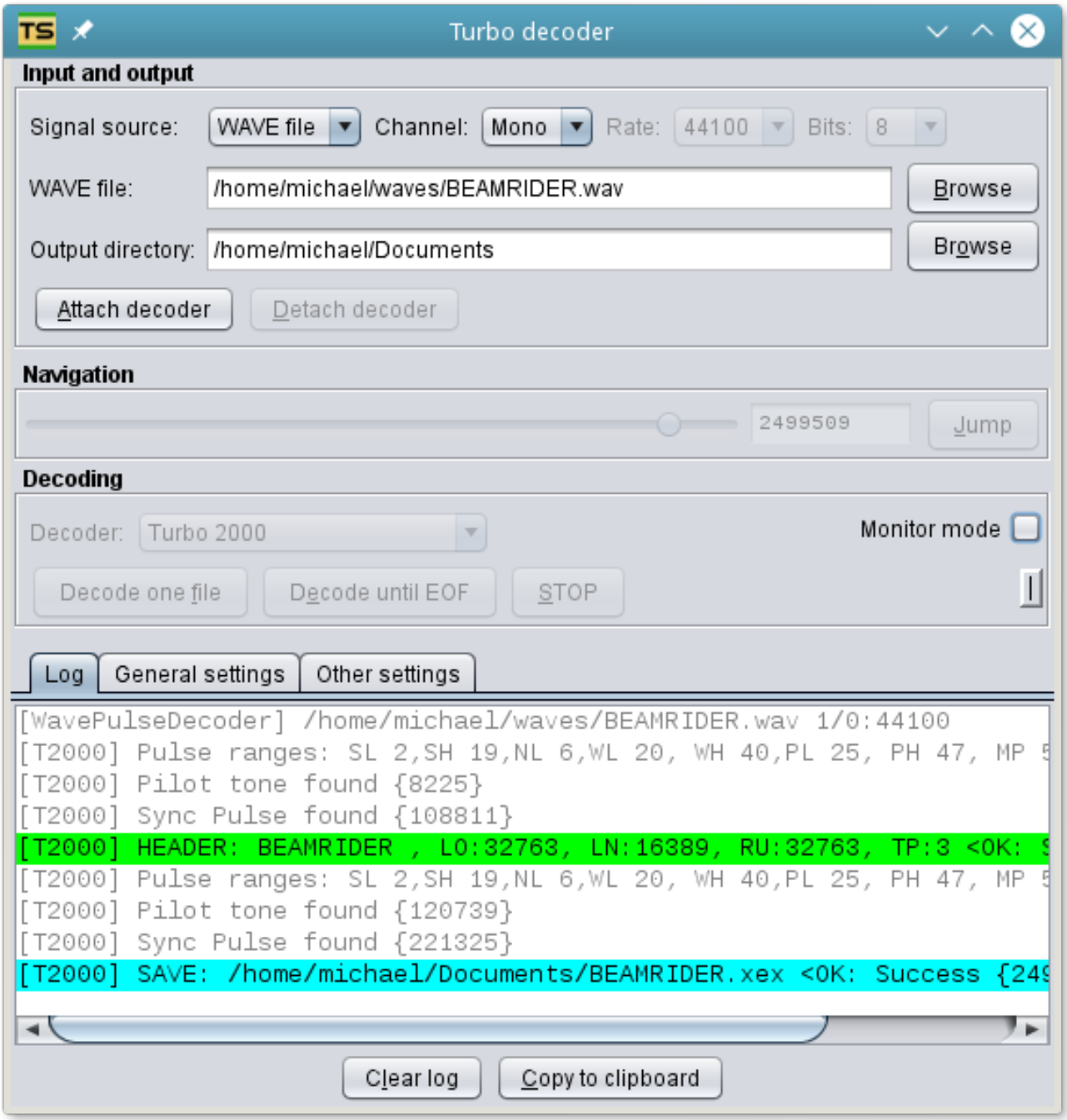

Figure 9: Turbo decoder

6.1.2 Preparing WAVE files

To prepare the wave file, you can use Audacity - free, open source, cross-platform software for recording and editing sounds.

Before decoding, applying of the following two filters is recommended:

1. High pass filter. Cutoff frequency 1000 Hz, rolloff 48 dB. This helps to remove DC offset.
2. Low pass filter. Cutoff frequency 12000 Hz, rolloff 48 dB. This removes small fluctuations of the signal.

Alternatively, you can use the Bass & Treble filter set to minimum bass and maximum treble.

To use a wave file with the A8CAS-enhanced atari800 emulator, amplify the signal with maximum amplification and allow clipping.

When using a software for editing sounds, *ensure that the resulting wave file is saved without any meta-data* (Artist name, track name...).

6.1.3 Preparing the Decoder

Open the decoder window by selecting the *Turbo decoder* item from the *Tools* menu.

Select a source of the electric signal using the *Signal source* combo box.

If the source of the electric signal is a WAVE file, fill-in the WAVE file name and select input channel using the *Channel* combo box.

If the source of the electric signal is the sound card, select input channel using the *Channel* combo box, select the sampling rate using the *Rate* combo box, and select number of bits per sample using the *Bits* combo box.

Enter the output directory and click the *Attach Decoder* button. The decoder is now ready to decode the electric signal.

6.1.4 Decoding Files

Use the *Navigation* panel to specify current position in the WAVE file.

The controls on the *Decoding* panel are devoted to perform decoding. Use the combo box *Decoder* to select a turbo system.

To start decoding of one file or all files until the end of the WAVE file, click *Decode one file* or *Decode until EOF* buttons. To stop the decoding process, press the *Stop* button.

You can see the results of decoding in the *Log* tab. If you use a digitized sound editor, you can use the sample numbers enclosed in the curly braces.

If you want to work with different WAVE file, or to select another source of the electric signal, detach the turbo decoder using the *Detach decoder* button.

If the decoder stops responding due to a bad sound card setup, press the *Stop* button while pressing the SHIFT key. This is called emergency stop. After an emergency stop, the decoder must be detached and attached to be ready again.

6.1.5 Monitor Mode

To work in the monitor mode, select the *Monitor mode* check box. In the Turbo monitor mode, turbo decoder reads raw blocks of the selected turbo system. If a block is found, it is decoded until an error occurs or end of block is encountered. No checksums are verified and internal format of the block is not validated.

To decode a single block, click the *Decode one file* button. To decode all blocks until end of file, click the *Decode until EOF* button.

The monitor mode is available for special purposes like decoding data stored in non-standard formats, retrieving corrupted files etc.

6.1.6 Configuring the Decoder

Use controls on the General settings and Other settings tabs to configure the decoder.

6.2 Tape Image Extractor

The tape image extractor is a tool designed to extract data from tape images. Tape image extractor allows you to do the following:

1. Display basic information about tape image chunks
2. Select tape image chunks for data extraction
3. Determine which portions of tape image chunks will be extracted

4. Extract data from tape image chunks to plain data files
5. Extract boot files to binary files
6. Extract data from tape image chunks to binary files
7. Save selected tape image chunks to a new tape image file

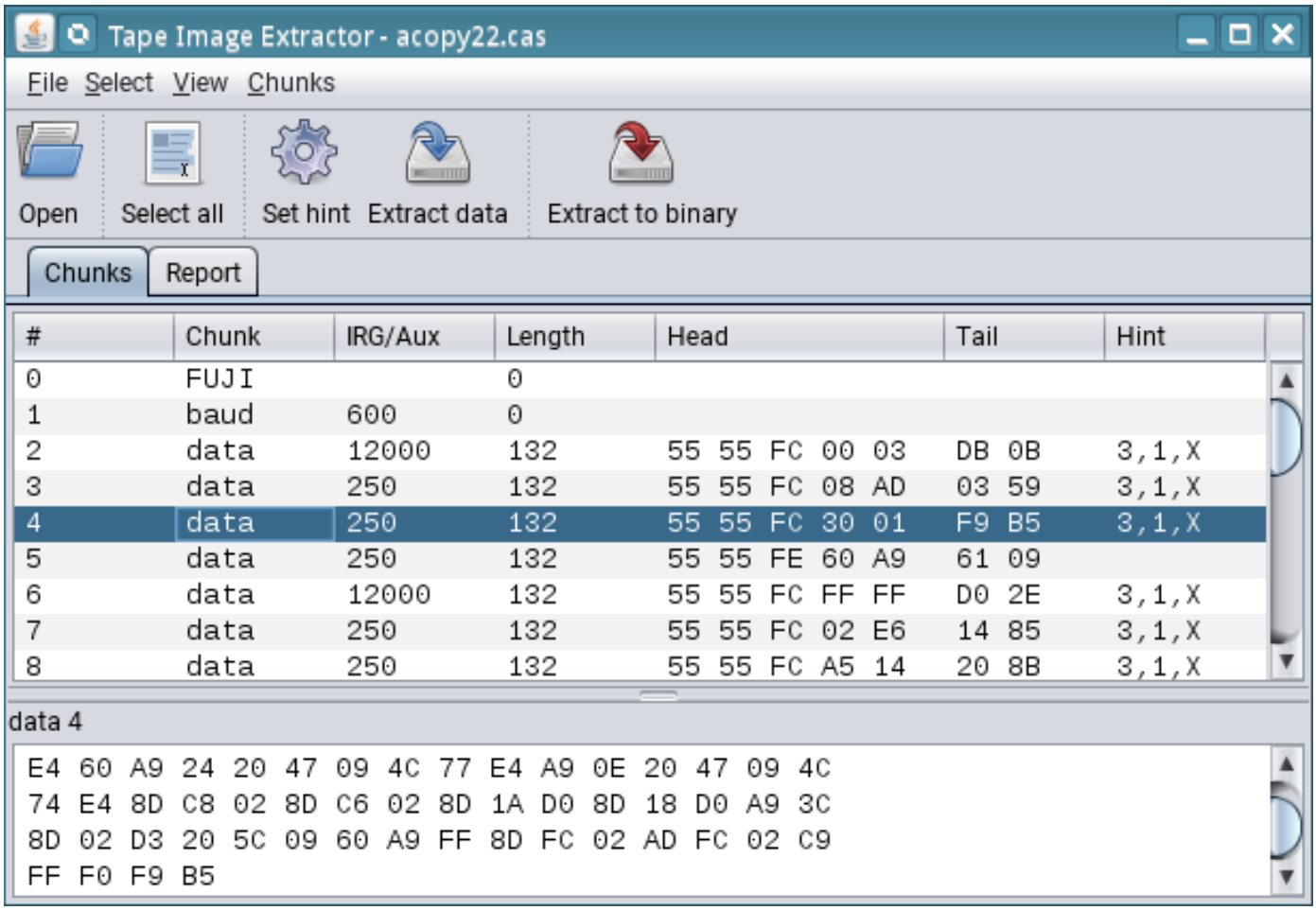

Figure 10: Tape Image Extractor

6.2.1 Operations

To open the Tape Image Extractor window, select the *Tape image extractor* item from the *Tools* menu.

To open a tape image, select the *Open tape image* item from the *File* menu.

To extract data from tape image chunks, select one or more tape image chunks. Then select the *Extract data* item from the *Chunks* menu. Select output file using the displayed file chooser. A report is displayed after the extraction.

To extract data from tape image chunks to a binary file, select one or more tape image chunks. Then select the *Extract data to binary file* item from the *Chunks* menu. A dialog that will allow you to select output file and parameters of the extraction will be displayed. Enter values and click the *Extract* button to perform the extraction.

To determine which portion of tape image chunk or chunks will be extracted, select one or more tape image chunks. Then select the *Set data extraction hint* from the *Chunk* menu. Enter values and click the *OK* button.

To save selected tape image chunks to a new tape image, select the *Save selected chunks* or *Save all chunks* item from the *File* menu. Select output file using the displayed file chooser. Note that FUJI chunk is inserted to the beginning of the tape image if needed.

6.3 Embedding Tokenized BASIC Files to Binary Files

6.3.1 Motivation

Some turbo systems define file formats than cannot store tokenized BASIC files. In order to circumvent such limitation, Turgen System has a tool that embeds tokenized BASIC files in monolithic binary files. **The purpose of the tool is embedding, not compilation**.

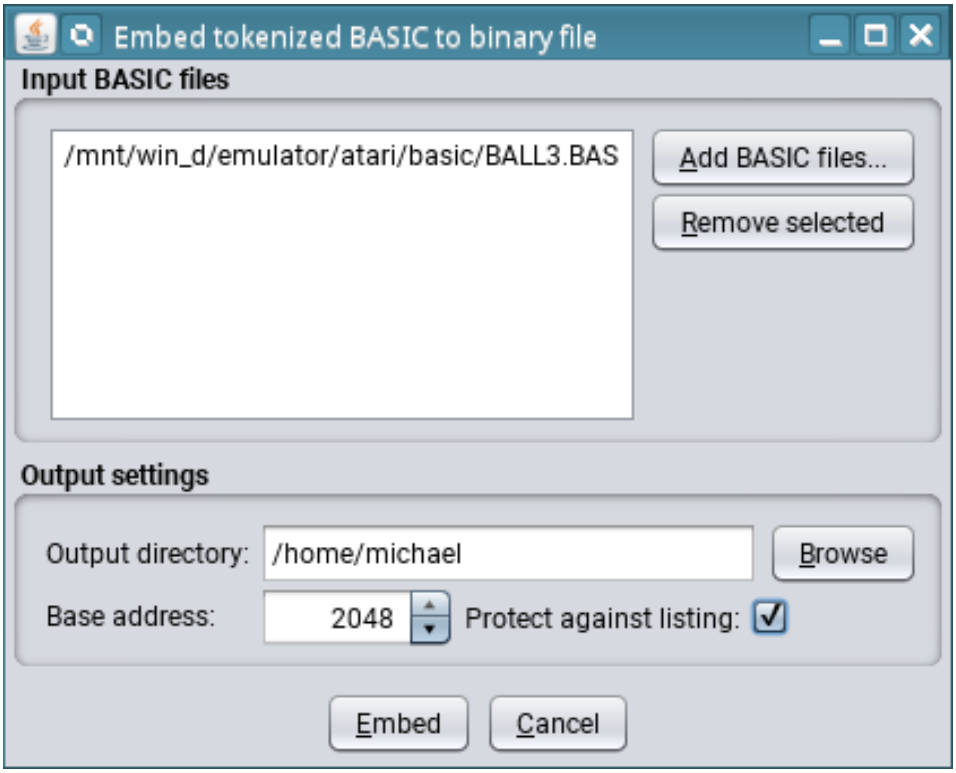

Figure 11: Embedding tokenized BASIC files to binary files

6.3.2 Operations

Open the embedding tool window by selecting the *Embed BASIC to binary file* item from the *Tools* menu.

Click the *Add BASIC files...* button to display a file chooser that allows you to select BASIC files to be embedded to binary files. Chosen files are added to a list. You can select multiple files to be embedded. Click the *Remove selected* button to remove BASIC files from the list.

Use the *Output directory* text field to specify into which directory the binary files will be placed. Use the *Browse* button to specify the directory using a file chooser.

Use the *Base address* spinner to specify address to which the BASIC files will be loaded (tokenized BASIC code is relocatable).

Use the *Protect against listing* check box to protect the BASIC program against listing by disabling the BREAK key and setting the COLDST flag. Note that such protection is only rudimentary and can be broken easily.

To embed the selected BASIC files in binary files, click the *Embed* button. All BASIC files in the list will be embedded in binary files. A report is displayed after embedding processing finishes.

6.4 Batch Processing

6.4.1 Overview

Batch processing allows to convert whole directories of binary files or tape images.

To invoke batch processing, select the *Process batch file* menu item from the *Special functions* sub-menu of the *Playlist* menu.

6.4.2 Syntax

Batch processing is controlled by a batch file that contains multiple *configurations*.

A configuration is defined by the following syntax:

```
BEGIN
INDIR <directory>
OUTDIR <directory>
CONVERTOR <convertor name>*
PULSES <pulses>
[CONVERTOR-ORDER <LIST|PILOT-TONES>]
[TAPE-IMAGE-CONVERTOR <convertor name>]
[SORT <NONE|ALPHABETICAL|LISTING>]
[AUTO-SILENCE-LIST] <duration>
END
```

Blank lines and lines beginning with # are ignored. Each keyword must be specified on a separate line.

Each configuration must begin with the **BEGIN** keyword.

The **INDIR** keyword identifies input directory with binary files. Only files with the .xex extension are processed. The directory specification can be absolute or relative.

The **OUTDIR** keyword identifies output directory (for wave files or tape images). The directory specification can be absolute or relative.

The **CONVERTOR** keyword identifies binary file convertor to be used. You can specify the CONVERTOR keyword multiple times to allow multiple convertors. A binary file is converted by the first convertor that accepts it, unless you specify CONVERTOR-ORDER other than LISTING.
Specify the CONVERTOR keywords in order from the most desirable convertor to least desirable convertor.

A convertor is identified by its name. Use listing of available plugins, convertors, and pulses to get the names of all available convertors.

The **PULSES** keyword identifies pulses to be used. Use listing of available plugins, convertors, and pulses to get the names of all available pulses.

The **CONVERTOR-ORDER** keyword determines order the convertors will be selected to convert binary files. Specify LISTING to select the first convertor that can accept the binary file (this is the default). Specify PILOT-TONES to select a convertor that generates minimum number of pilot tones.

The **TAPE-IMAGE-CONVERTOR** identifies tape image convertor to be used. You can specify this keyword multiple times to allow multiple convertors. A tape image is converted by the first convertor that accepts it. Specify the TAPE-IMAGE-CONVERTOR keywords in order from the most desirable convertor to least desirable convertor.

A convertor is identified by its name. Use listing of available plugins, convertors, and pulses to get the names of all available convertors

The **SORT** keyword determines order the files will added to the playlist. Specify NONE for order given by the file system. Specify ALPHABETICAL for alphabetical order. For full control of the order, specify LISTING for order determined by the listing.txt file located in the input directory. The listing.txt file contains list of file names.

The **AUTO-SILENCE-LIST** keyword allows to automatically generate silence lists with specified default duration.

Each configuration must end with the **END** keyword.

6.4.3 Sample Batch File

```
BEGIN
INDIR /mnt/win_d/emulator/atari/exebin
OUTDIR ../waves2
CONVERTOR Turbo 2000 - Monolithic Binary
CONVERTOR Turbo 2000 - Blockloading
CONVERTOR-ORDER PILOT-TONES
PULSES T2270
```

```
SORT NONE
END
```

This will convert all binary files from the exebin directory. The files will be converted to Turbo 2000 as monolithic binary files if possible. If not, the files will be converted using the Blockloading conversion. Transfer speed will be approximately 2270 bd. The output files will be placed to the waves2 directory.

6.4.4 User Interface

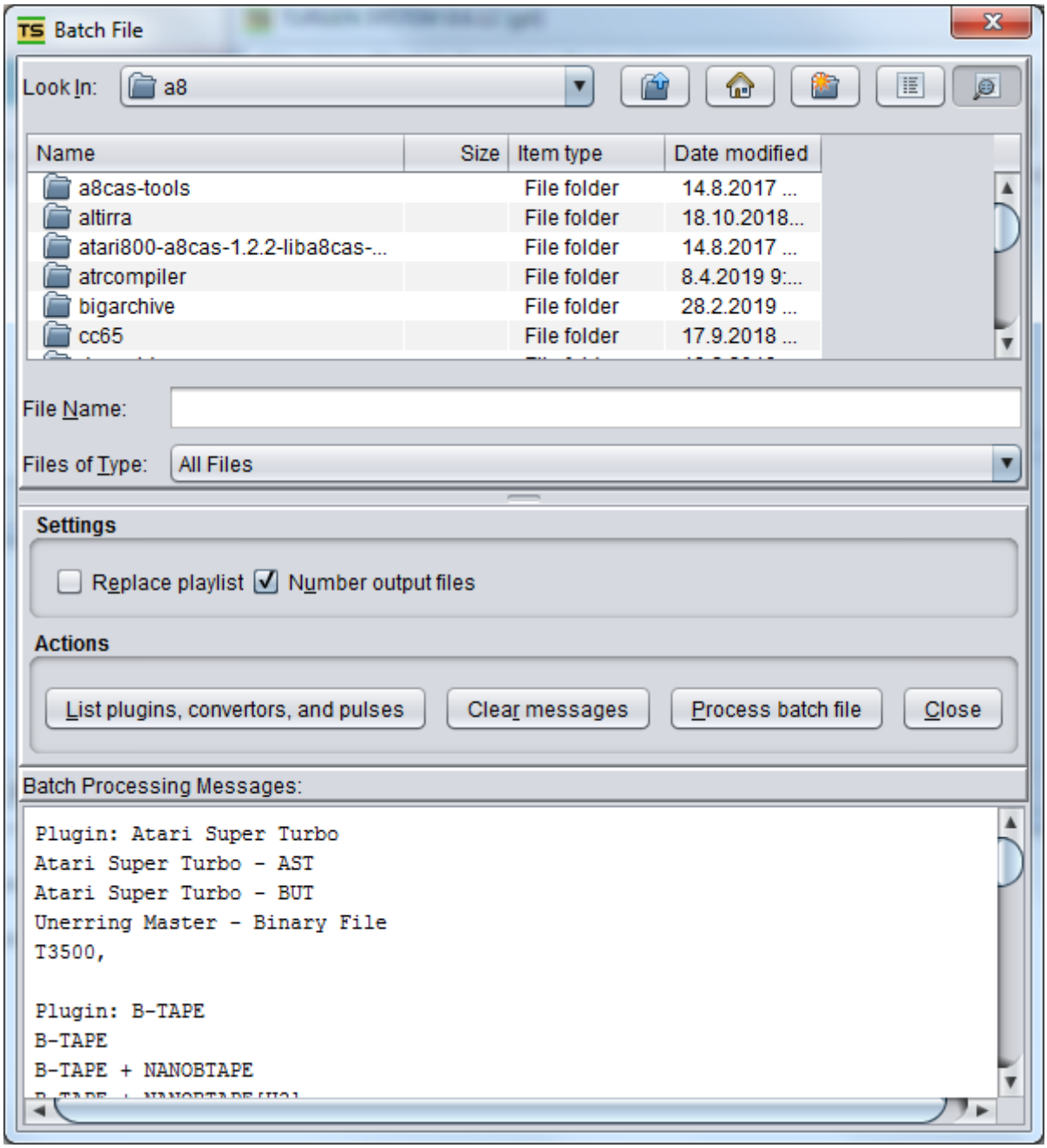

Figure 12: Batch Processing

- Click the *Process batch file* button to process the selected batch file and create playlist items.
- Select the *Replace playlist* check box to replace all existing playlist items with items created by the batch processing. Select the *Number output files* check box to create playlist items with numbered output files.
- Click the *Close* button to close the dialog.

- Click the *Clear messages* button to clear all messages.
- Click the *List plugins, convertors, and pulses* button to display listing of all available plugins, convertors and pulses.

6.5 Merging Segments of Binary Files

6.5.1 Introduction

Turgen System provides a tool that merges segments of binary files into one segment and adds extra code that replaces the INIT segments. This allows to convert segmented binary files that meet certain conditions to monolithic binary files.

Conditions that the segmented binary files must meet are the following:

1. Data segments must not overlap
2. Routines called through INIT vectors must work properly even when all segments of the binary file are already loaded
3. There is a free memory area for the code that replaces the INIT segments

6.5.2 Merging Segments Step by Step

1. Select the *Merge segments of binary file* item from the *Tools* menu to display the dialog for merging segments of binary file.
2. Enter file name in the *Input file* text field and click the *Analyze* button. The list of segments will be populated and attributes of the code that replaces INIT segments will be automatically detected.
3. Enter file name in the *Output file* text field. Verify address of the code that replaces the INIT segments and change it if needed. Note that the code is needed only when the binary file contains INIT segments.
4. Click the *Merge* button to merge the segments and to create the output binary file. Then test the resulting binary file.

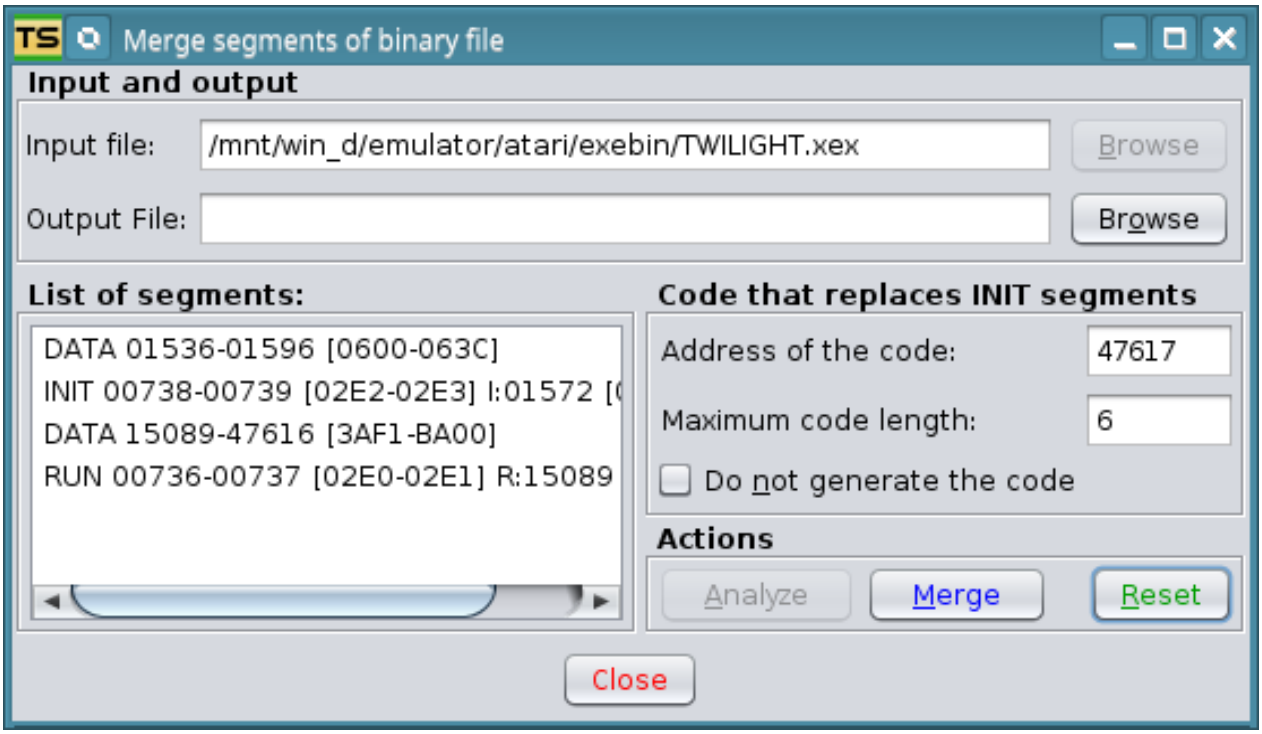

Figure 13: Merging segments of binary file

6.6 Pilot Tone Test

Turgen System can generate continous pilot tone. Select the *Pilot Tone Test* item from the *Tools* menu. The controls allow you to adjust width of the pulses. Parameter of the signal are those selected in the *Audio Generator* section of the *Preferences* dialog.

Use this function to:

- Test your recording equipment
- Find ideal signal amplitude when using cassette adapter

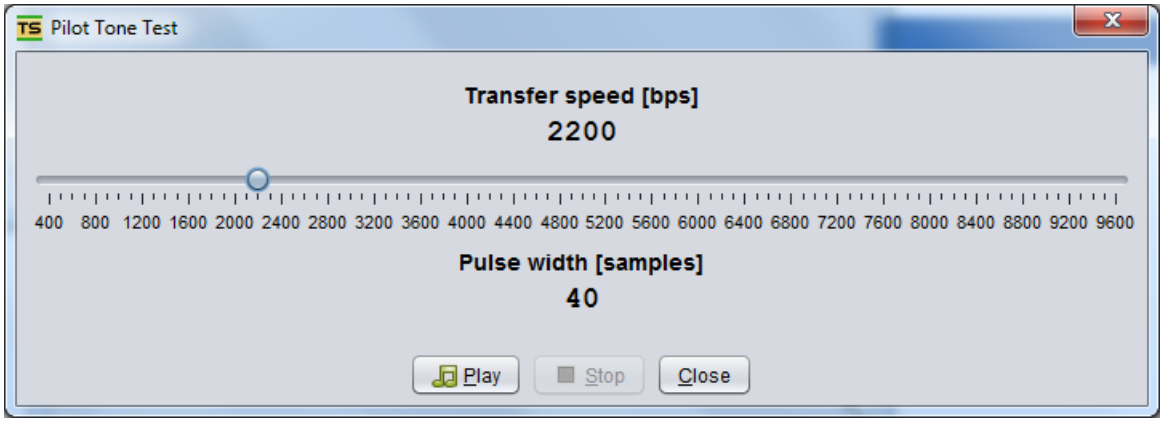

Figure 14: Pilot Tone Test

7 Hardware that Can Be Used with Turgen System

7.1 Data Recorders and Compact Cassettes

For best experience with Turgen System, use a data recorder with a turbo upgrade and quality compact cassettes.

7.2 Cassette Adapters

If you do not wish or cannot use compact cassettes, you can try an affordable and available cassette adapter. This device looks like a compact cassette, but there is no tape inside. Instead of using the tape, the cassette adapter is connected to an external source of signal using a thin cable with a JACK connector.

The cassette adapter is very versatile, as it can be connected to various devices like CD players, MP3 players or similar devices. Also do not forget that the cassette adapter will work with data recorders that can no more spin the reels, but their electrical circuitry is still intact.

The Cassette adapter is most useful when connected to a computer running Turgen System.

To use the cassette adapter, place it into your data recorder (ensure that its top side is oriented correctly) and connect it to an external signal source. Familiarize yourself with the concept of silence lists (see section 3.8) for reasons stated in section 7.5.

7.3 MEGA-CD Interface (CD-LINK)

This interface allows you to plug an external signal source as it would be a data recorder with Czechoslovak Super Turbo upgrade. In essence, it is an external version of Super Turbo circuitry with a pair of RCA connectors and a SIO connector.

You can use the MEGA-CD interface to transfer Turbo 2000, Super Turbo, Turbo 2000 - kilobyte blocks, and B-TAPE records to Atari. Atari Super Turbo and Turbo ROM should work as well.

The MEGA-CD interface was not originally designed to transfer standard tape records, but with Turgen System, this is also possible when a high-quality signal source is used (computer, CD player, or MP3 player). To transfer standard tape records, you must set *Do not modulate standard records* configuration entries to *true*.

7.4 SIO2PC and ATART

With appropriate software, both devices allow transfer of standard tape records.

7.5 Loading Binary Files with Replacement Devices

Replacement devices. The replacement devices are all devices that are intended to replace data recorders and/or compact cassettes.

Some examples of typical replacement devices:

- An MP3 player connected to a cassette adapter inserted to the data recorder.
- A SIO2PC device connected to a computer running a data recorder emulation (e.g. AspeQT).
- MEGA-CD interface

Background Information. The Atari computer uses a dedicated data recorder specially designed to co-operate with the computer. One advantage of this design is that the Atari computer can programatically control the data recorder.

There is a dedicated wire (MOTOR CTRL) in the SIO connector that allows the computer to switch the motor of the data recorder on and off. You might have already noticed, that if you just press the PLAY button, the cassette stands still until the computer is instructed to load data from the data recorder (e.g. by entering the CLOAD BASIC command or booting from tape).

When loading a binary file with an INIT segment, the computer will stop the data recorder, execute machine code identified by the INIT segment, and then re-start the data recorder. Execution of the code can take several seconds, or the code can wait for input from the user.

The Problem. Ther replacement devices **cannot be controlled by the Atari computer**. These devices are simply not connected to the MOTOR CTRL wire, so they cannot obey the computer's orders to start or stop.

This problem surfaces, when a binary file contains INIT vectors. When the machine code is executed, the replacement device simply keeps sending the signal to the computer (instead of waiting for the MOTOR CTRL signal).

This way, blocks or segments can be skipped and the loading fails.

The Solution. To partially solve the problem stated, Turgen System supports silence lists. The silence lists allow to generate silence or pause the signal generator after blocks that contain the INIT vectors.

If you are using a replacement device, it is strongly recommended that you make yourself familiar with the silence lists and use them.

For more information about the silence lists, refer to section 3.8.

8 Postprocessing

Turgen System allows to process its outputs by external programs. This is called *postprocessing*. After the output file is created, an external program according to a given *command line* is executed.

To reference the Turgen System's output in the command line, some special symbols are introduced. The symbols are described in table 3.

If the command line starts with the exclamation mark (!), output file is deleted after the external program terminates, if possible.

Processing of the command line is very limited. The command line is interpreted as a sequence of strings separated by spaces. The first string is name (possibly including path) of the external program, the remaining strings are parameters passed to the external program. Parameters containing spaces have to be enclosed in double quotes.

Command lines (after special symbol substitution) are always stored into the `turgen.log` file, so the correctness of command lines can be verified.

Symbol	Meaning
%OD%	Output directory
%ODS%	Output directory followed by file separator
%OFN%	Output file name without last extension
%OFNE%	Output file name with last extension

Table 3: Special symbols

8.1 Postprocessing of WAVE Files

To enable postprocessing of the WAVE files, select the *Wave postprocessing* check box menu item from the *Generator* menu. Specify the command line in the Wave output/Postprocessing command configuration entry.

8.2 Postprocessing of Tape Images

To enable postprocessing of the tape images, select the *Tape image postprocessing* check box menu item from the *Generator* menu. Specify the command line in the Tape image output/Postprocessing command configuration entry.

8.3 Examples

Example of the special symbols for output file: `/mnt/win_d/emulator/atari/wav/river_raid.wav`

```
%OD% = /mnt/win_d/emulator/atari/wav
%ODS% = /mnt/win_d/emulator/atari/wav/
%OFNE% = river_raid.wav
%OFN% = river_raid
```

Example of conversion to MP3:

```
lame %ODS%%OFNE% %ODS%%OFN%.mp3
```

Example of conversion to OGG Vorbis with deletion of the original file:

```
!oggenc -o %ODS%%OFN%.ogg %ODS%%OFNE%
```

Conversion to MP3 using LAME in the terminal emulator and consequent playback by the xmms media player. The original file is deleted:

```
!konsole -e bash -c "lame %ODS%%OFNE% %ODS%%OFN%.mp3 && xmms %ODS%%OFN%.mp3"
```

9 Hints and Tips

- If the SHIFT key is pressed when the *Add item to playlist* main control button is pressed, the contents of the playlist item manipulation dialog will not be cleared. This can be used to duplicate playlist items.
- A good way to reduce loading times is to reduce duration of the pilot tones.
- Do not forget that you can size most of the program windows as you want. The bounds of the windows are persistent, so if you exit the Turgen System and restart it, you will find the windows exactly where you left them.
- If you, for some reason, store the generated electric signal on compact discs (CDs) and you use compact disc player, try to switch off various shock protections and avoid nonstandard rotational speeds. Such features sometimes cause sample looses.

10 Solving Common Problems

When you cannot transfer your programs and games to your Atari 8-bit computer, try the following solutions.

Volume. Try to change volume (signal amplitude). Use available controls (*Amplitude* configuration entries, volume controls provided by your operating system, potentiometers etc.) to change volume. Use the Pilot Tone Test tool described in section 6.6 to find appropriate volume.

Polarity of the Pulses. Some turbo systems (Rambit Turbo Tape, Turbo 6000) require certain polarity of the pulses. Devices in your recording chain can sometimes reverse the polarity. Use the *Invert polarity of the pulses* configuration entries to solve this problem.

Waveform. Devices in your recording chain can have maximum signal frequency limits. Choose waveform that is suitable for the weakest device in your recording chain (it is usually the tape). For more information, refer to section 3.5.4.

Standard Tape Records. Before you try any of the turbo systems, check if you can transfer standard tape records. When you are not able to transfer standard tape records, there is probably a serious problem with your recording chain.

System Sounds. The electric signal can be disturbed by system sounds (notification bubbles, error messages, background tasks). When using the *Generate AUDIO directly* function, check that the system sounds are disabled.

Cassette Adapters. When using a cassette adapter, ensure that it is placed properly in your data recorder. Also do not use extremely cheap cassette adapters (sold as toys).

Cassette Type. Ensure that you are using cassettes of a reasonable quality.

Where to Buy New Cassettes. There are at least two manufacturers.
In Europe, you can buy Maxell UR-90.
In US, you can buy NAC Audio PRO cassettes.

11 Production of Multiple Tapes

Turgen System offers several facilities that will help you to produce multiple tapes with high degree of automation. Review the following sections to learn more.

1. Batch Processing - section 6.4
   Batch processing allows to automatically create playlist from a directory of binary files
2. Automatic File Name Creation - section 3.9
   This allows to automatically generate good file names that are following rules of a particular turbo system
3. Prefixing wave file names - section 4.4
   Names of the output wave files can be prefixed with side numbers. This allows to easily create tape sides of defined lengths.
4. Copying playlist as CSV - section 3.3.4
   You can copy playlist items to system clipboard as comma-separated values. This helps to create printed file listings or album art.

Part II

Plugins

12 Turbo 2000 and Super Turbo

12.1 Overview

The plugins convert input files to Czechoslovak turbo systems **Turbo 2000** and **Super Turbo**. Both plugins also provide means to circumvent main limitation of both systems - the incapability to store segmented binary files. The user interface is depicted in figure 15.

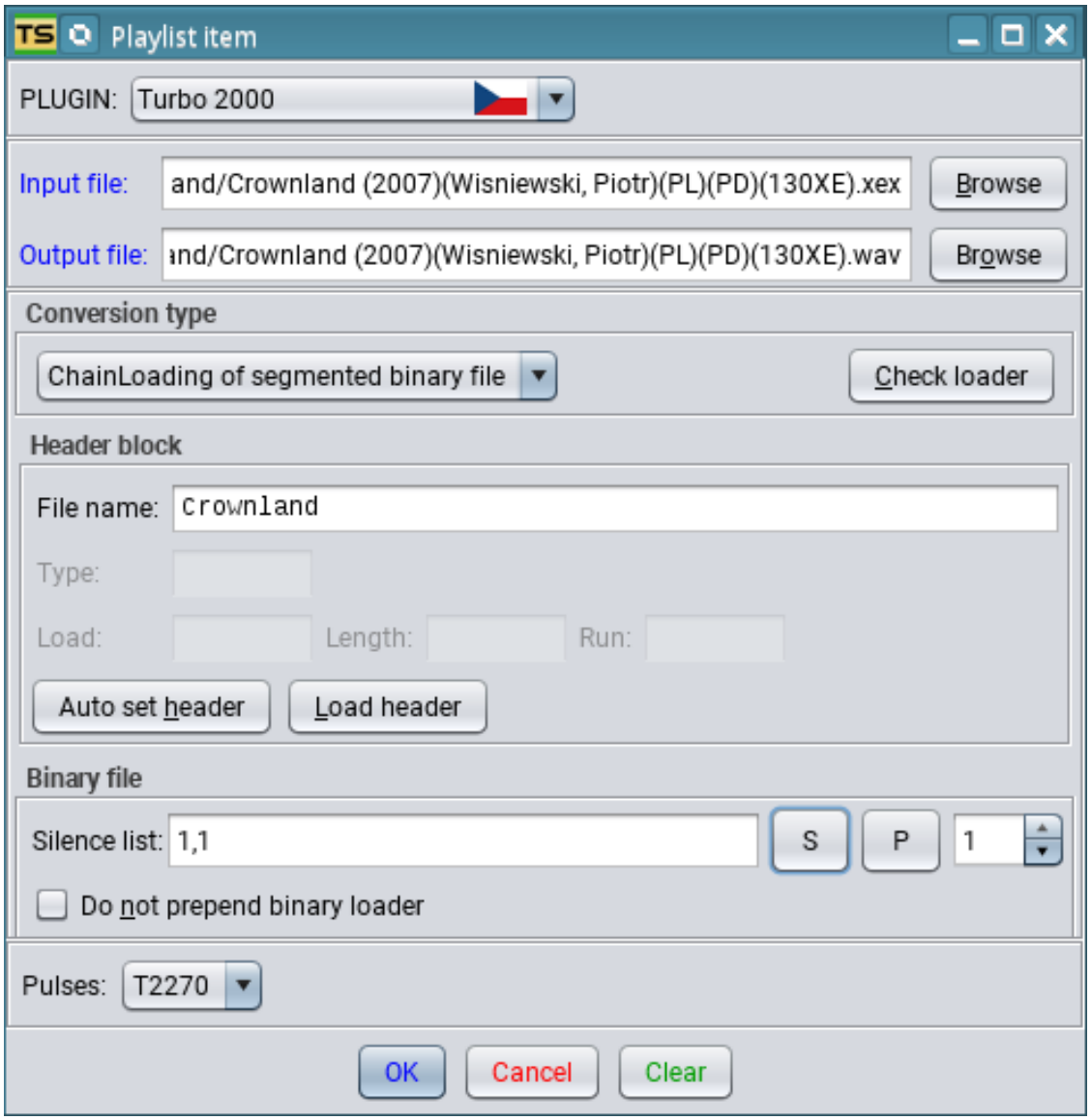

Figure 15: Turbo 2000, Super Turbo

12.2 Conversion types

Turbo 2000 and Super Turbo plugins support 6 conversion types. Select the conversion type using the *Conversion type* combo box.

Monolithic binary file to Turbo 2000 or Super Turbo. Conversion of a monolithic binary file (see 1.7.1). The input file must be a monolithic binary file. This is a natural usage of Turbo 2000 or Super Turbo

systems.

ChainLoading. Conversion of a segmented binary file to a *chain of Turbo 2000 or Super Turbo block pairs*. A special binary loader (Chainloader 2) that loads the block pairs is prepended before the chain. Click the *Check loader* button to determine if the binary loader can load the selected input binary file..

BlockLoading. Conversion of a segmented binary file to a *chain of Turbo 2000 or Super Turbo blocks*. A special binary loader (Blockloader) that loads the blocks (and stores information where to place the blocks in the memory) is prepended before the chain. The binary file can have up to 62 segments. Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

Tokenized BASIC to Turbo 2000 or Super Turbo. Conversion of a tokenized BASIC program. Note that many loaders cannot load such program directly. To circumvent such limitation, you can embed tokenized BASIC to binary file. See section 6.3.

Plain DATA to Turbo 2000 or Super Turbo. Conversion of plain data. You must enter values in the text fields related to the header block.

Binary file to binary turbo. Conversion of a binary file to the Binary turbo [1] (file type 4) format. Binary files can be stored in this format, but under normal circumstances, you cannot run them directly.

There are two known programs that provide special functionality for these files:

1. VisiCopy III (a commercial product by JRC) allows you to run such binary files, but only if they do not contain INIT segments.
2. ChainCopy 1.2 (auxiliary utility for Turgen System) can read these files and convert them as the ChainLoading conversion would do.

12.3 Header Block

Use the text fields *File name, Type, Load, Length* and *Run* to specify entries of the Turbo 2000 or Super Turbo header block.

Click the *Auto set header* button to automatically enter values in the text fields mentioned above. The information is obtained by an analysis of the input file and the selected conversion type. Furthermore, if the input file is not a monolithic binary file and according to the selected conversion type it should be, a warning dialog with recommendations is displayed.

Click the *Load header* to load the turbo header from an external file. The header is loaded automatically if it is stored in a file that is named same as the input file followed by a `.theader` suffix. In other cases, a file chooser is displayed.

12.4 Inserting Silence After INIT Segments

The text field *Silence list* contains a silence list (refer to section 3.8). This is applicable only for ChainLoading and BlockLoading conversion types.

12.5 Prepending Universal Turbo Loader

If you do not have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

[1] Also known as “Rožnovské turbo”

12.6 Using Custom Binary Loaders

Select the *Do not prepend binary loader* check box if you intend to use your own custom version of Chainloader 2 or Blockloader. The file will be converted, but the loader will not be prepended, so you will be able to use your own.

12.7 Configuration Entries

Turbo 2000/Invert polarity of the pulses
Invert polarity of the pulses

Turbo 2000/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-8192)

Turbo 2000/Data block pilot tone duration
Number of pilot tone pulses of the data block (256-8192)

Turbo 2000/BlockLoading pilot tone duration
Number of pilot tone pulses of data blocks for BlockLoading conversion type (256-8192)

Turbo 2000/Silence after header
Duration of silence that is generated after the header block (0-30). Note that many Universal Turbo loaders do not tolerate any silence after header.

Turbo 2000/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block. This can be useful for Turbo 2000 only loaders that wait for a key press. Note that many Universal Turbo loaders do not tolerate any interruptions or silence after the header block.

Turbo 2000/Prepend the Universal Turbo loader
Prepend the Universal Turbo loader as a standard tape boot file

Turbo 2000/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

Super Turbo/Invert polarity of the pulses
Invert polarity of the pulses

Super Turbo/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-8192)

Super Turbo/Data block pilot tone duration
Number of pilot tone pulses of the data block (256-8192)

Super Turbo/BlockLoading pilot tone duration
Number of pilot tone pulses of data blocks for BlockLoading conversion type (256-8192)

Super Turbo/Prolongate pilot tone
Increase number of pilot tone pulses with increasing approximate baud rate

Super Turbo/Prolongate pilot tone for BlockLoading
Increase number of pilot tone pulses with increasing approximate baud rate for BlockLoading conversion type

Super Turbo/Silence after header
Duration of silence that is generated after the header block (0-30). Note that many Universal Turbo loaders do not tolerate any silence after header.

Super Turbo/Convert binary loaders to Turbo 2000
If enabled, the Chainloader and Blockloader binary loaders are converted to Turbo 2000 instead of Super Turbo.

Super Turbo/Prepend the Universal Turbo loader
Prepend the Universal Turbo loader as a standard tape boot file

Super Turbo/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

13 Turbo 2000 - Kilobyte Blocks

13.1 Overview

The plugin converts files to the **Turbo 2000 - Kilobyte Blocks** Czechoslovak turbo system. There are no special restrictions on input files.

13.2 User Interface

The user interface is depicted in figure 16.

The text field *File name* corresponds to the same field of the Turbo 2000 - kilobyte blocks header block. The purpose of the text field *Silence list* is described in section 3.8.

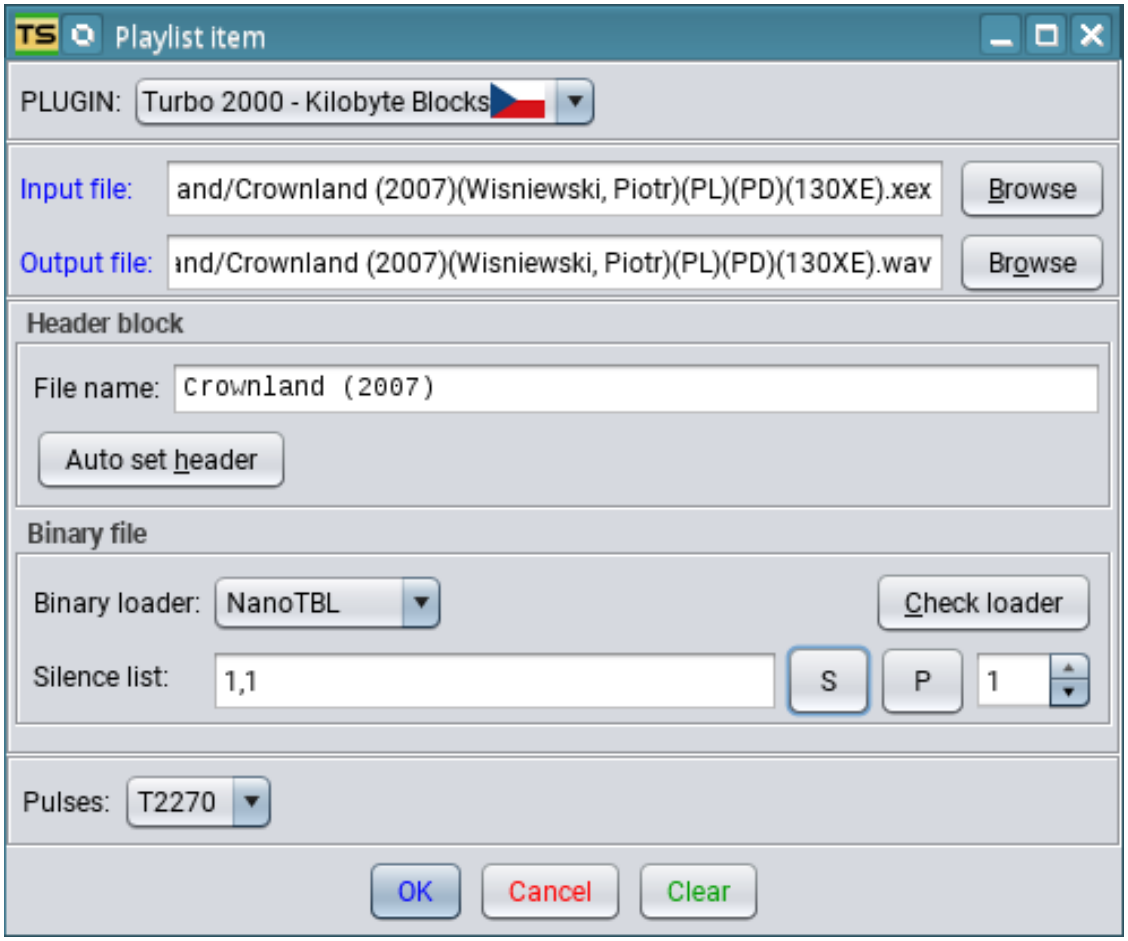

Figure 16: Turbo 2000 - kilobyte blocks

13.3 Binary Loaders

Turbo 2000 - kilobyte blocks turbo system can store binary files. You can prepend one of the special miniature built-in binary loaders. For this turbo system, you can choose MiniTBL, NanoTBL, NanoTBL[UR] or NanoTBL[U2] loader by using the *Binary loader* combo box. The loader is converted to the Turbo 2000 system.

The MiniTBL loader is a combination of stripped“T:” device handler that allows only READ operation and code for binary load using CIO.

The NanoTBL loaders are single-purpose binary loaders reading data blocks, not using CIO at all. NanoTBL stores data blocks to the freely available RAM. NanoTBL[UR] stores the data blocks to the beginning of the “RAM under ROM”, NanoTBL[U2] to the end of “RAM under ROM”.

Click the *Check loader* button to determine if the selected binary loader can load the input binary file..

13.4 Prepending Universal Turbo Loader

If you don't have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

13.5 Configuration Entries

Turbo 2000 - Kilobyte blocks/Invert polarity of the pulses
Invert polarity of the pulses

Turbo 2000 - Kilobyte Blocks/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-8192)

Turbo 2000 - Kilobyte Blocks/Data block pilot tone duration
Number of pilot tone pulses of the data blocks (256-8192)

Turbo 2000 - Kilobyte Blocks/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

Turbo 2000 - Kilobyte Blocks/Silence after loader
Number of seconds of the silence inserted after the binary loader (0-30)

Turbo 2000 - Kilobyte Blocks/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 3.8.5.

Turbo 2000 - Kilobyte Blocks/Name the loader same as file
The binary loader will have name same as the converted file

Turbo 2000 - Kilobyte Blocks/Prepend the Universal Turbo loader
Prepend the Universal Turbo loader as a standard tape boot file

Turbo 2000 - Kilobyte Blocks/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

14 B-TAPE

14.1 Overview

The plugin converts files to the **B-TAPE** Czechoslovak turbo system. There are no special restrictions on input files. The B-TAPE system is backward compatible with the Turbo Tape system used by the TT-DOS operating system.

14.2 User Interface

The user interface is depicted in figure 17.

The text field *File name* corresponds to the same field of the B-TAPE data block. Use the *Tape mode* combo box to select tape mode. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

14.3 Binary Loaders

The B-TAPE turbo system can store binary files. You can prepend one of the special miniature built-in binary loaders. For this turbo system, you can choose NanoBTAPE, NanoBTAPE[UR] or NanoBTAPE[U2] loader using the *Binary loader* combo box. The loader is converted to the Turbo 2000 system.

The NanoBTAPE loaders are single-purpose binary loaders reading blocks not using CIO. NanoBTAPE stores the blocks to the freely available RAM. NanoBTAPE[UR] stores blocks to the beginning of the "RAM under ROM", NanoBTAPE[U2] to the end of "RAM under ROM".

Click the *Check loader* button to determine if the selected binary loader can load the input binary file.

14.4 Prepending Universal Turbo Loader

If you don't have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

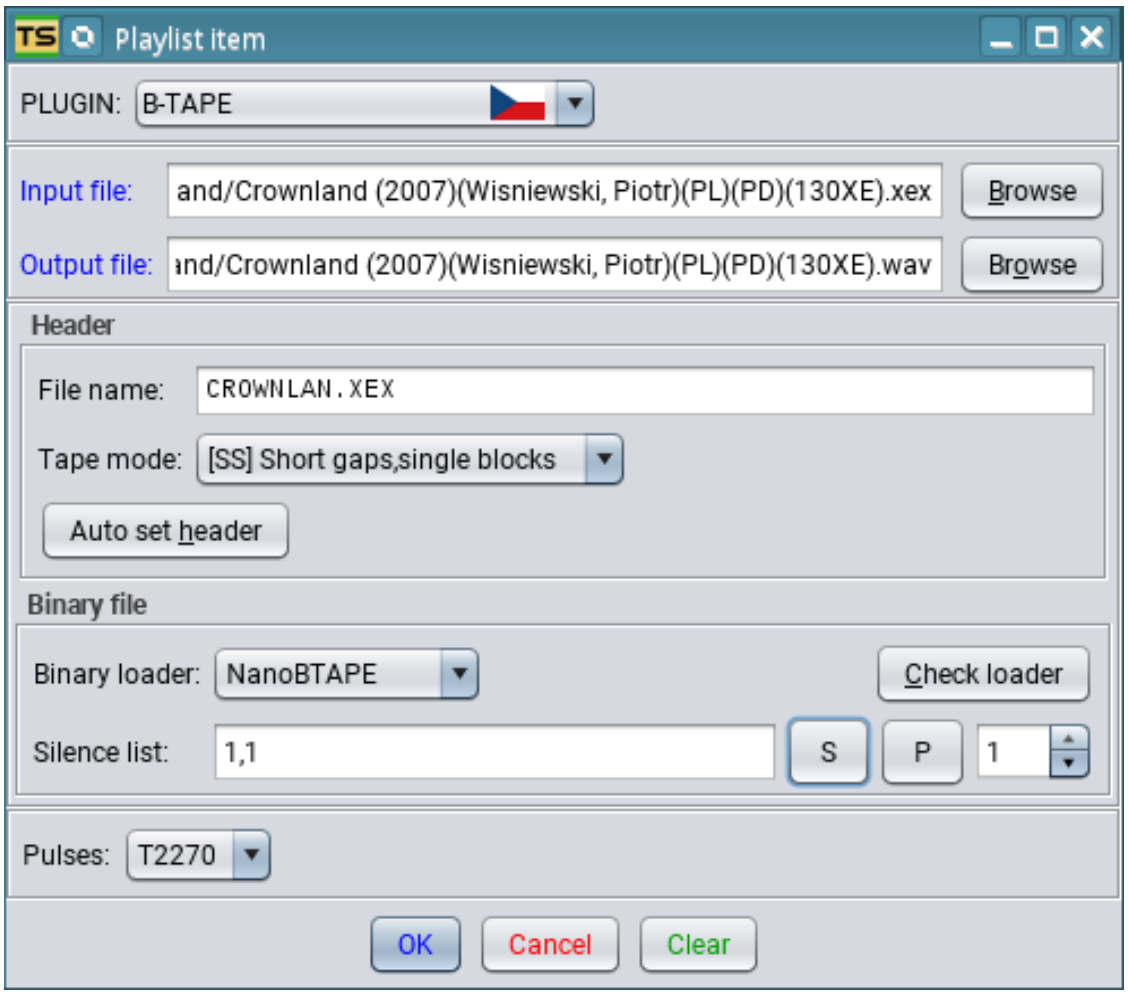

Figure 17: B-TAPE

14.5 Configuration Entries

B-TAPE/Invert polarity of the pulses
Invert polarity of the pulses

B-TAPE/Pilot tone duration
Number of pilot tone pulses of the blocks (256-8192)

B-TAPE/Prolongate pilot tone
Increase number of pilot tone pulses with increasing approximate baud rate

B-TAPE/Silence after first block
Number of seconds of the silence inserted after the first B-TAPE block (0-30)

B-TAPE/Silence after loader
Number of seconds of the silence inserted after the binary loader (0-30)

B-TAPE/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 3.8.5.

B-TAPE/Name the loader same as file
The binary loader will have name same as the converted file

B-TAPE/Prepend the Universal Turbo loader
Prepend the Universal Turbo loader as a standard tape boot file

B-TAPE/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

15 KSO Turbo 2000

15.1 Overview

The plugin converts files to the **KSO Turbo 2000** Polish turbo system. The KSO Turbo 2000 system is compatible with various similar turbo systems (e.g. Turbo 2000F).

15.2 Conversion types

15.2.1 Natural Format

This conversion converts any file to the natural file format defined by the KSO Turbo 2000. There are no special restrictions on input files.

15.2.2 Binary File to L3 Format

This conversion converts binary files to a special L3 file format designed for loading of binary files with long segments. The conversion is performed as follows:

1. L3 Binary loader is converted to the natural file format
2. Input binary file is converted to the special L3 file format

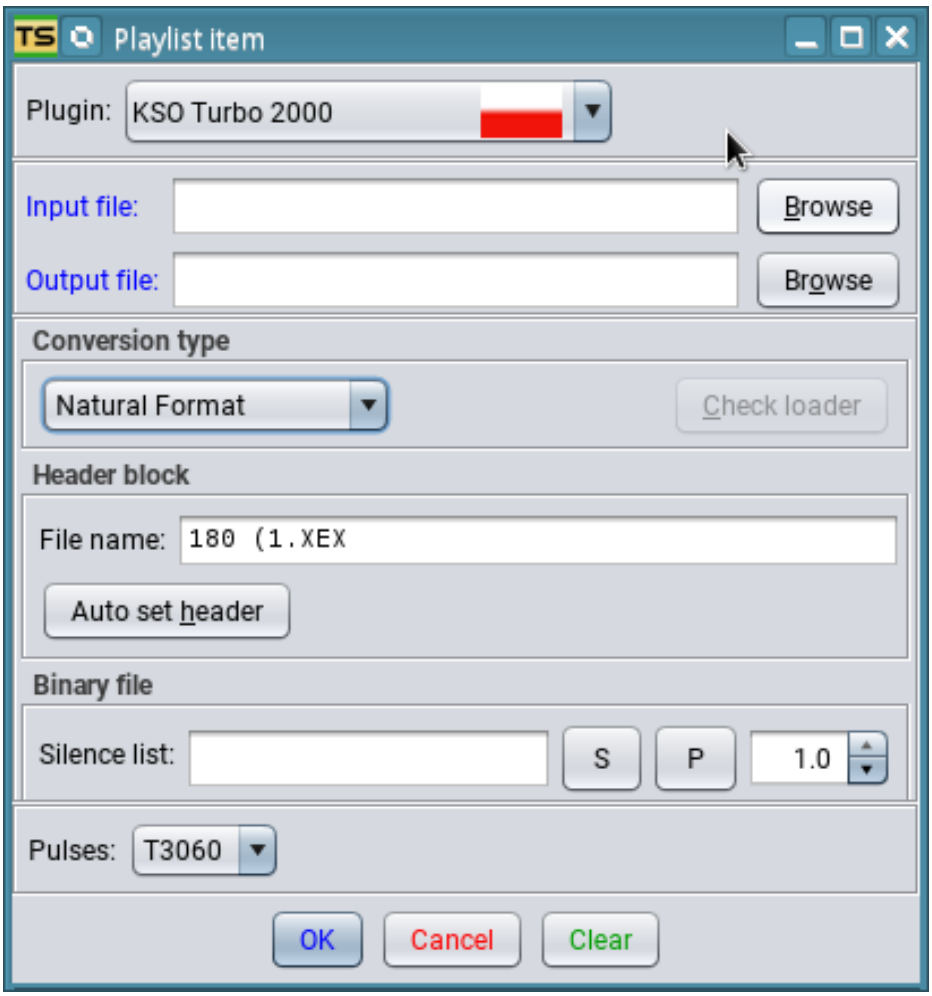

Figure 18: KSO Turbo 2000

15.3 User Interface

The user interface is depicted in figure 18.

Use the *Conversion type* combo box to select conversion type. Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Enter file name in the *File name* text field. Click the *Auto set header* button to automatically create the file name. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details..

15.4 Configuration Entries

KSO Turbo 2000/Invert polarity of the pulses
Invert polarity of the pulses

KSO Turbo 2000/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-8192)

KSO Turbo 2000/Data block pilot tone duration
Number of pilot tone pulses of the data blocks (256-8192)

KSO Turbo 2000/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

KSO Turbo 2000/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 3.8.5.

KSO Turbo 2000/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

KSO Turbo 2000/Pulse Corrections
Corrections of pulses

KSO Turbo 2000/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

16 Turbo Blizzard

16.1 Characteristics

Plugin converts files to **Turbo Blizzard** Polish turbo system. There are no special restrictions on input files.

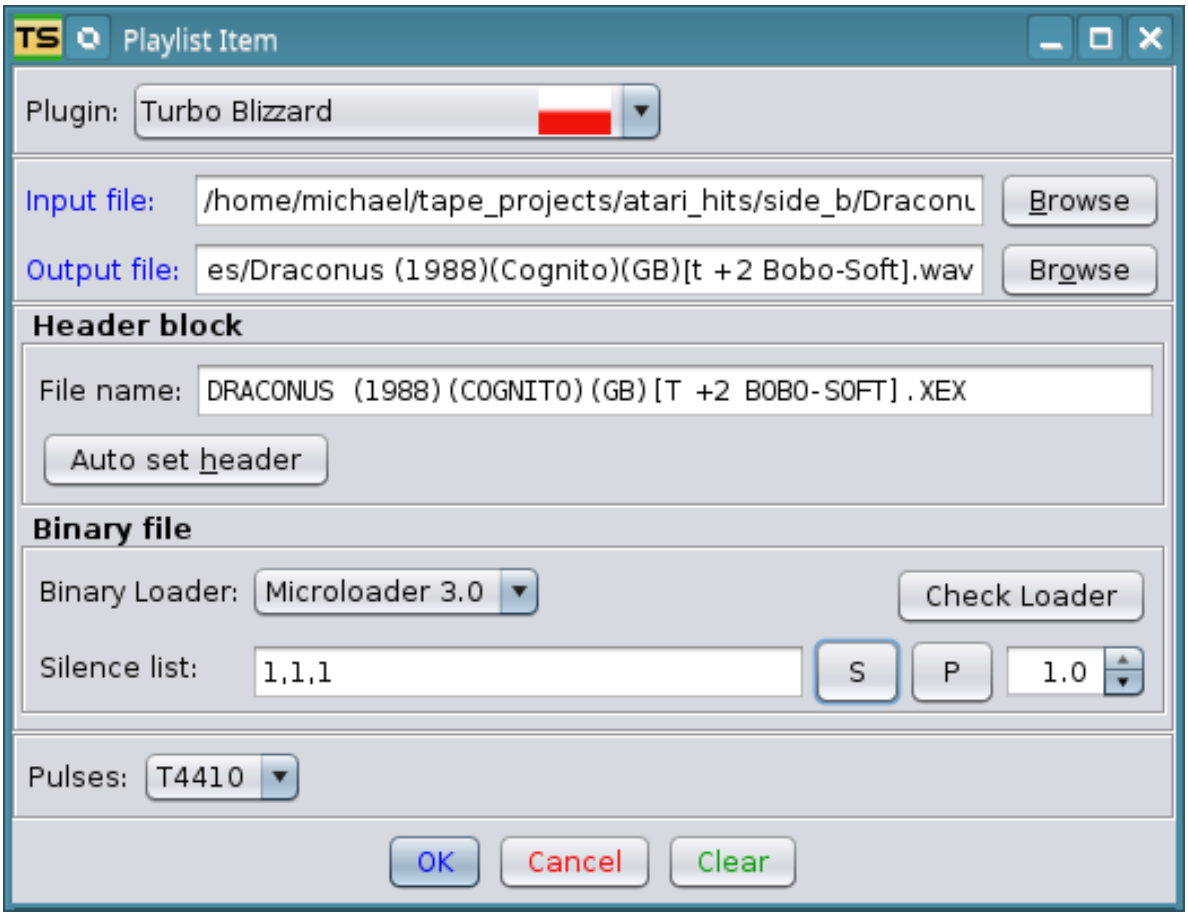

Figure 19: Turbo Blizzard

16.2 User Interface

The user interface is depicted in figure 19.

The text field *File name* corresponds to the same field of the header block. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details..

Use the *Binary loader* combo box to prepend a binary loader. You can choose to prepend none, Microloader 3.0, or ShortKOS 2.0. Click the *Check loader* button to determine if the selected binary loader can load the input binary file.

16.3 Sampling Rate

The Turbo Blizzard plugin take advantage of the sampling rate of 48000 Hz.

16.4 Configuration Entries

Turbo Blizzard/Invert polarity of the pulses
Invert polarity of the pulses

Turbo Blizzard/Long gaps between blocks
Make long gaps between blocks

Turbo Blizzard/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

Turbo Blizzard/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Turbo Blizzard/Convert binary loaders to tape boot files
If enabled, the binary loaders are converted to tape boot files. Otherwise, the loaders are converted to Turbo Blizzard.

Turbo Blizzard/Silence after loader
Duration of silence generated after binary loader

Turbo Blizzard/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 3.8.5.

Turbo Blizzard/Pulse Corrections
Corrections of pulses

Turbo Blizzard/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

17 Turbo ROM

17.1 Characteristics

The plugin converts files to the **Turbo ROM** Polish turbo system. The following input files are supported:

- Turbo ROM compatible binary files
- Binary files
- Tokenized BASIC files

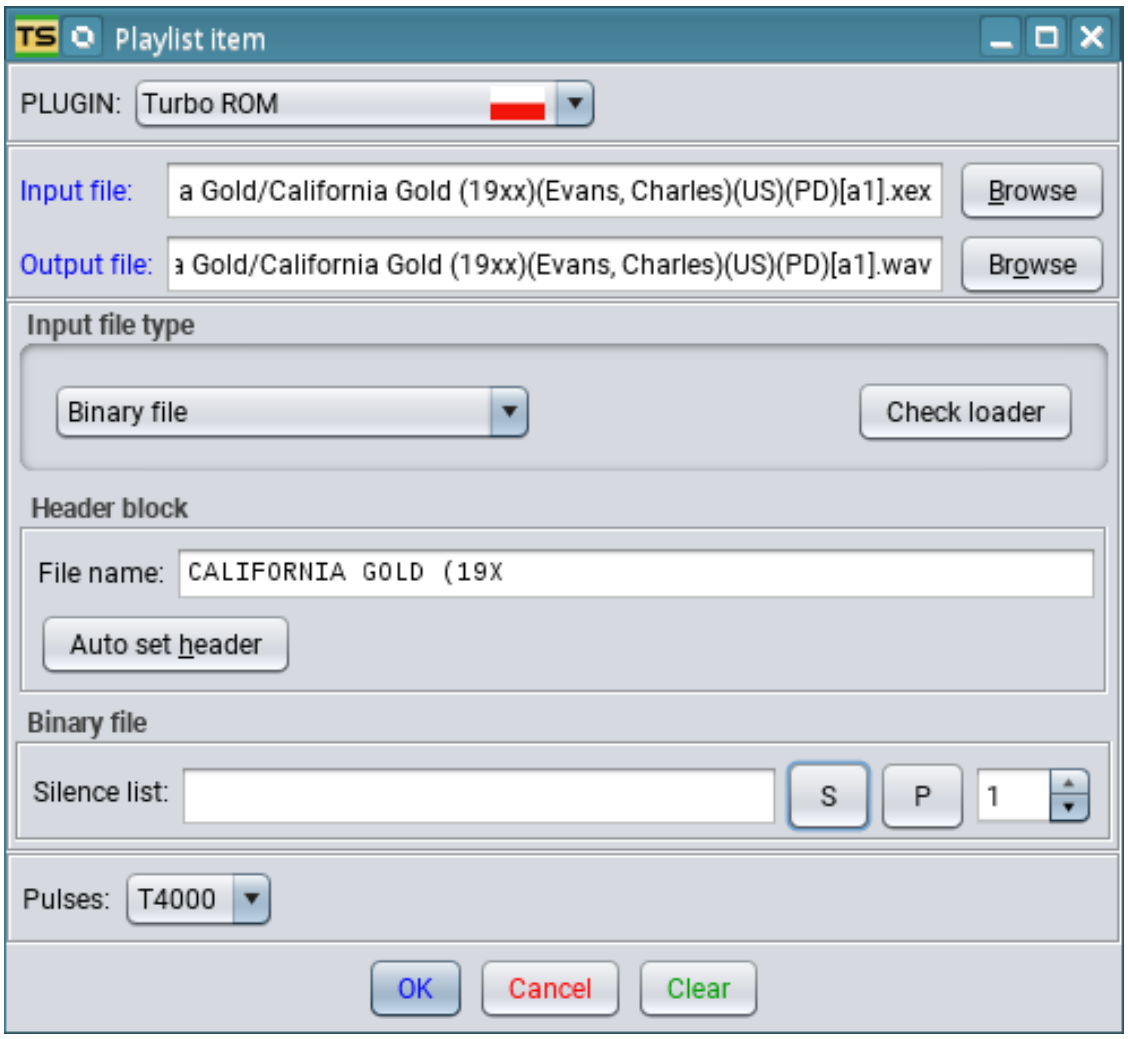

Figure 20: Turbo ROM

17.2 User Interface

The user interface is depicted in figure 20.

The text field *File name* corresponds to the same field of the header block. Use the *Auto set header* button to automatically set file name and also to verify whether the input file is Turbo ROM compatible binary file.

Use the *Input file type* combo box to select input file type.

17.3 Conversion of Turbo ROM Compatible Binary Files

These binary files consist of exactly one DATA segment and at most one RUN segment and at most one INIT segment. Files are converted to the Turbo ROM natural format.

17.4 Conversion of Binary Files

17.4.1 Processing

Binary Files are converted to a special format. A special binary loader in the Turbo ROM natural format is prepended. The binary files are normalized before conversion (Combined RUN+INIT segments are split, RUN segment is moved to the end of the file).

17.4.2 User Interface

Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

17.5 Conversion of Tokenized BASIC Files

Files are converted to the Turbo ROM natural format for tokenized BASIC files.

17.6 Prepending Loaders

If you do not have a cartridge with Turbo ROM loader, you can prepend a tape version of such loader. Set the *Prepend Turbo ROM loader* configuration entry to deserved value. Set the *Convert binary loader to a standard tape boot file* configuration entry to *true*.

17.7 Sampling Rate

The Turbo ROM plugin take advantage of the sampling rate of 48000 Hz.

17.8 Configuration Entries

Turbo ROM/Invert polarity of the pulses
Invert polarity of the pulses

Turbo ROM/Header pilot tone duration
Number of pilot tone pulses of the header block (1024-8192)

Turbo ROM/Data block pilot tone duration
Number of pilot tone pulses of the data block (128-2048)

Turbo ROM/Binary load pilot tone duration
Number of pilot tone pulses of blocks when binary files are converted

Turbo ROM/Silence after header
Number of seconds of the silence inserted after the header block

Turbo ROM/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Turbo ROM/Silence after binary loader
Number of seconds of the silence inserted after binary loader

Turbo ROM/Convert binary loader to a standard tape boot file
If enabled, the binary loader is converted to a standard tape boot file instead of a Tubo ROM compatible binary file.

Turbo ROM/Prepend Turbo ROM loader
Determines if a Turbo ROM natural format loader will be prepended as a standard tape boot file. The values are the following: “None” - no loader will be prepended, “Standard” - a plain loader will be prepended, “Diagnostic” - a verbose loader that displays hexadecimal dump of the header block will be prepended.

Turbo ROM/LOAD address of a BASIC program
Base address to which BASIC programs will be loaded

Turbo ROM/RUN address of a BASIC program
Address of a routine that executes BASIC programs (default is 41086)

Turbo ROM/Pulse Corrections
Corrections of the pulses

Turbo ROM/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

18 Atari Super Turbo

18.1 Overview

The plugin converts files to the Polish **Atari Super Turbo** and compatible systems. The following conversions are supported:

- Conversion to the **Atari Super Turbo AST** format that can store binary files that have up to 44 segments and no INIT segments.
- Conversion to the **Atari Super Turbo BUT** format that can store binary files that have up to 254 segments of any type. A generic BUT loader is used to load such files. The loader occupies addresses 1926 - 2185.
- Conversion to the **Unerring Master Atari Turbo System**. This turbo system can store binary files.

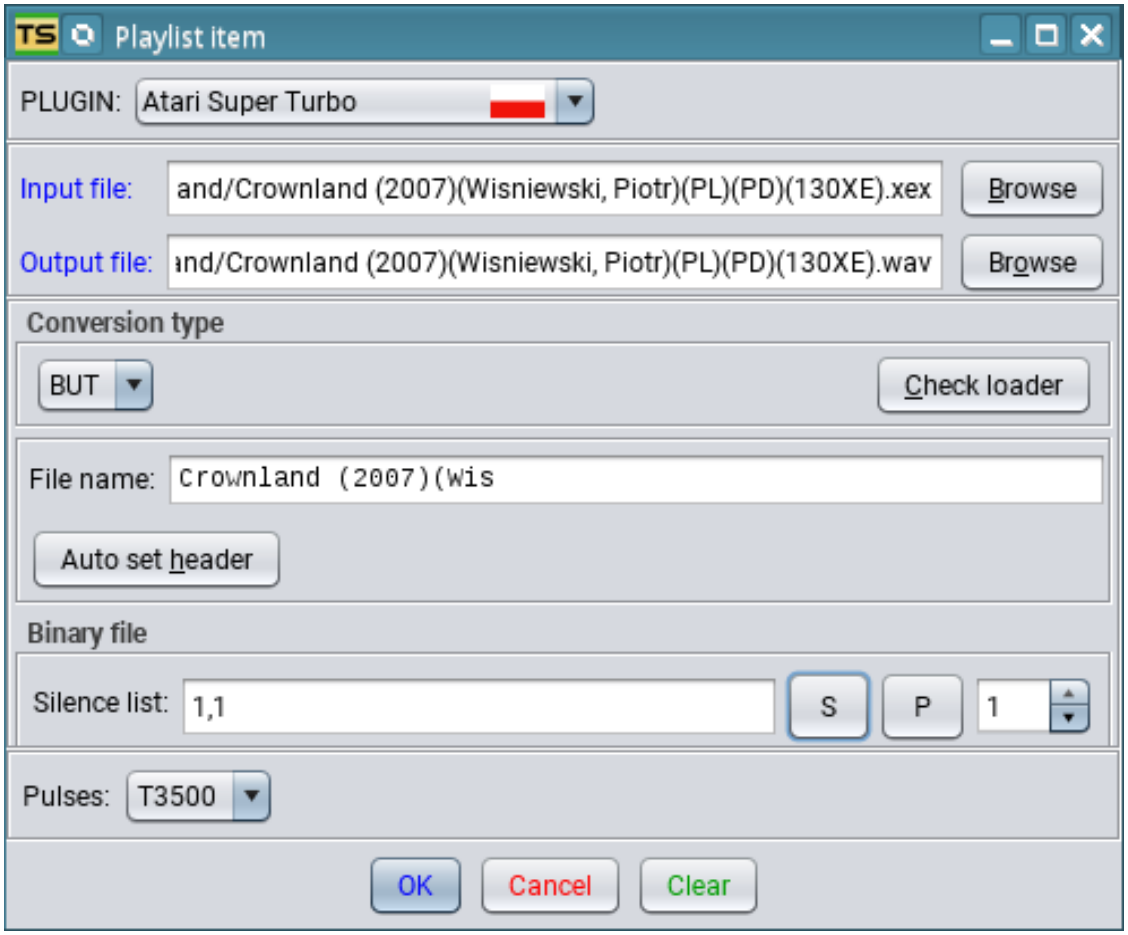

Figure 21: Atari Super Turbo

18.2 User Interface

The user interface is depicted in figure 21.

The text field *File name* corresponds to the same field of the header block. The *Auto set header* button can be used to automatically set file name and also to verify whether the input file is compatible with the selected conversion (AST, BUT, or UM). Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

18.3 Configuration Entries

Atari Super Turbo/Invert polarity of the pulses
Invert polarity of the pulses

Atari Super Turbo/AST header pilot tone duration
Number of pilot tone pulses of the AST header block (256-16384)

Atari Super Turbo/Data block pilot tone duration
Number of pilot tone pulses of data blocks (256-16384)

Atari Super Turbo/Silence after AST header
Number of seconds of the silence inserted after the header block (0-30)

Atari Super Turbo/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Atari Super Turbo/BUT loader will wait for any key
Determines whether the BUT loader will require a key press to proceed

Atari Super Turbo/Silence after BUT loader
Duration of silence generated after the BUT loader

Atari Super Turbo/Pulse Corrections
Corrections of pulses

Atari Super Turbo/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

19 Hard Turbo

19.1 Overview

The plugin converts files to the **Hard Turbo** Polish turbo system. According to the nature of this turbo system, the input files must be binary files.

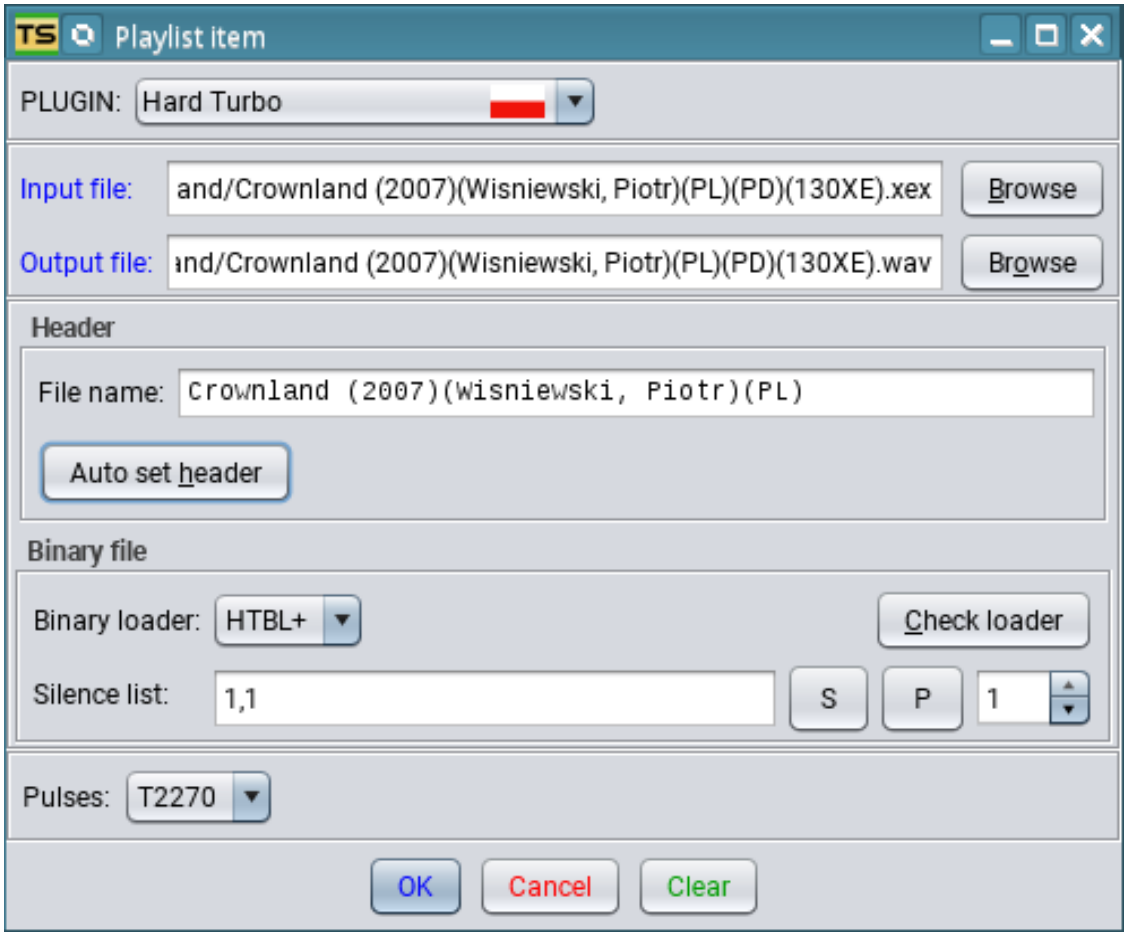

Figure 22: Hard Turbo

19.2 User Interface

The user interface is depicted in figure 22.

The text field *File name* corresponds to the same field of the main header block. Use the *Auto set header* to automatically set file name. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

19.3 HTBL+ Loader

You can add HTBL+ loader before the binary file converted.

HTBL+ is an extended binary loader that has the following enhancements

- Data recorder motor is switched off after segment data is loaded. This enables loading of binary files that execute lot of code from INIT segments
- RUNAD is set in a way that allows RUN segment to be the first segment of the binary file
- The loader never waits for any key under normal circumstances. This makes the loader easier to use with devices that do not support MOTOR CTRL signal
- The loader forces warm start during its initialization. This allows to load and run programs that rely on the display configuration set up by the warm start routines
- BASIC is automatically disabled

To add the HTBL+ loader, select HTBL+ from the *Binary loader* check box. Click the *Check loader* button to determine if the HTBL+ binary loader can load the selected input binary file..

19.4 Configuration Entries

Hard Turbo/Invert polarity of the pulses
Invert polarity of the pulses

Hard Turbo/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-16384)

Hard Turbo/Data block pilot tone duration
Number of pilot tone pulses of data blocks (256-16384)

Hard Turbo/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

Hard Turbo/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Hard Turbo/Name the loader same as file
The binary loader will have name same as the converted file

Hard Turbo/Convert the HTBL+ loader to a standard tape boot file
If enabled, the HTBL+ binary loader will be converted to a standard tape boot file. This helps when you do not have a cartridge with a loader.

Hard Turbo/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

20 Lower Silesian Turbo 2000

20.1 Overview

The plugin converts binary files to the **Lower Silesian Turbo 2000** turbo system from Poland.

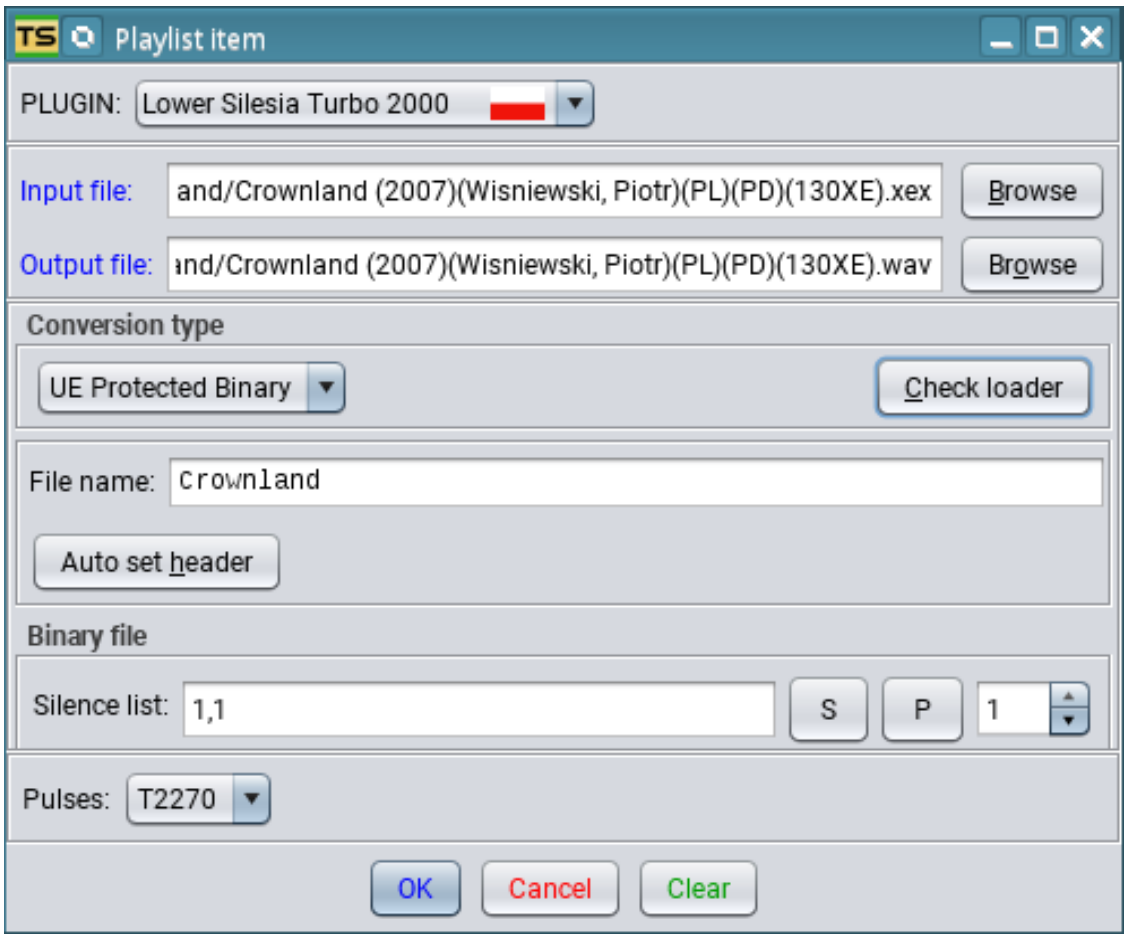

Figure 23: Lower Silesian Turbo 2000

20.2 Conversion Types

Two conversion types are supported.

Auto Turbo. Conversion of an input binary file to the Auto Turbo format (see section 29.6.2). The Auto Turbo format can store any binary file, but only binary files that have no INIT segments can be loaded and executed.

UE Protected Binary. Conversion of an input binary file to the Protected Binary Format (see section 29.6.3) use by the "Unknown Exterminator" copier. A binary loader is converted to the Auto Turbo format. Then the input binary file is converted to the Protected Binary Format.

20.3 User Interface

The user interface is depicted in figure 23.

The text field *File name* corresponds to the same field of the header block. Click the *Auto set header* button to automatically set the file name. Select conversion type using the *Conversion type* box. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

Click the *Check loader* button to determine if the binary loader used for given conversion type can load the selected input binary file.

20.4 Configuration Entries

Lower Silesian Turbo 2000/Header block pilot tone duration
Number of pilot tone pulses of the header block (256-8192)

Lower Silesian Turbo 2000/Data block pilot tone duration
Number of pilot tone pulses of the data block (256-8192)

Lower Silesian Turbo 2000/Segment header pilot tone duration
Number of pilot tone pulses of the segment header block (128-768)

Lower Silesian Turbo 2000/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

Lower Silesian Turbo 2000/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Lower Silesian Turbo 2000/Silence after loader
Duration of silence generated after binary loader

Lower Silesian Turbo 2000/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

21 Rambit Turbo Tape

21.1 Overview

This plugin converts files to the **Rambit turbo system** from the United Kingdom. Given the nature of the Rambit turbo system, only binary files can be converted.

21.2 Conversion Types

Two conversion types are supported.

Monolithic binary file to Rambit. Conversion of monolithic binary file to the natural format for monolithic binary files defined by the Rambit turbo system. A monolithic binary file loader is prepended as a standard tape boot file.

Segmented binary file to Rambit. Conversion of segmented binary file to the natural format for segmented binary files defined by the Rambit turbo system. A segmented binary file loader is prepended as a standard tape boot file.

21.3 User Interface

The user interface is depicted in figure 24.

Use the *Conversion type* box to select conversion type. Enter file name that will be displayed by the loaders in the *File name* box. Click the *Auto set header* button to automatically set the file name. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

Click the *Check loader* button to determine if the selected binary loader can load the input binary file. Use the *Do not prepend binary loader* box not to prepend the binary loader (use this when using your own binary loader).

21.3.1 Setting Look and Feel of the Loaders

Click the *Color 1* and *Color 2* buttons to set the look and feel of the loaders.

- Original loaders do not allow to change look and feel.
- Modern loaders display horizontal bars of two colors selected with both buttons
- Modern loaders 2 display shades of one basic color selected with the *Color 1* button.

21.4 Loaders

This plugin provides three sets of loaders.

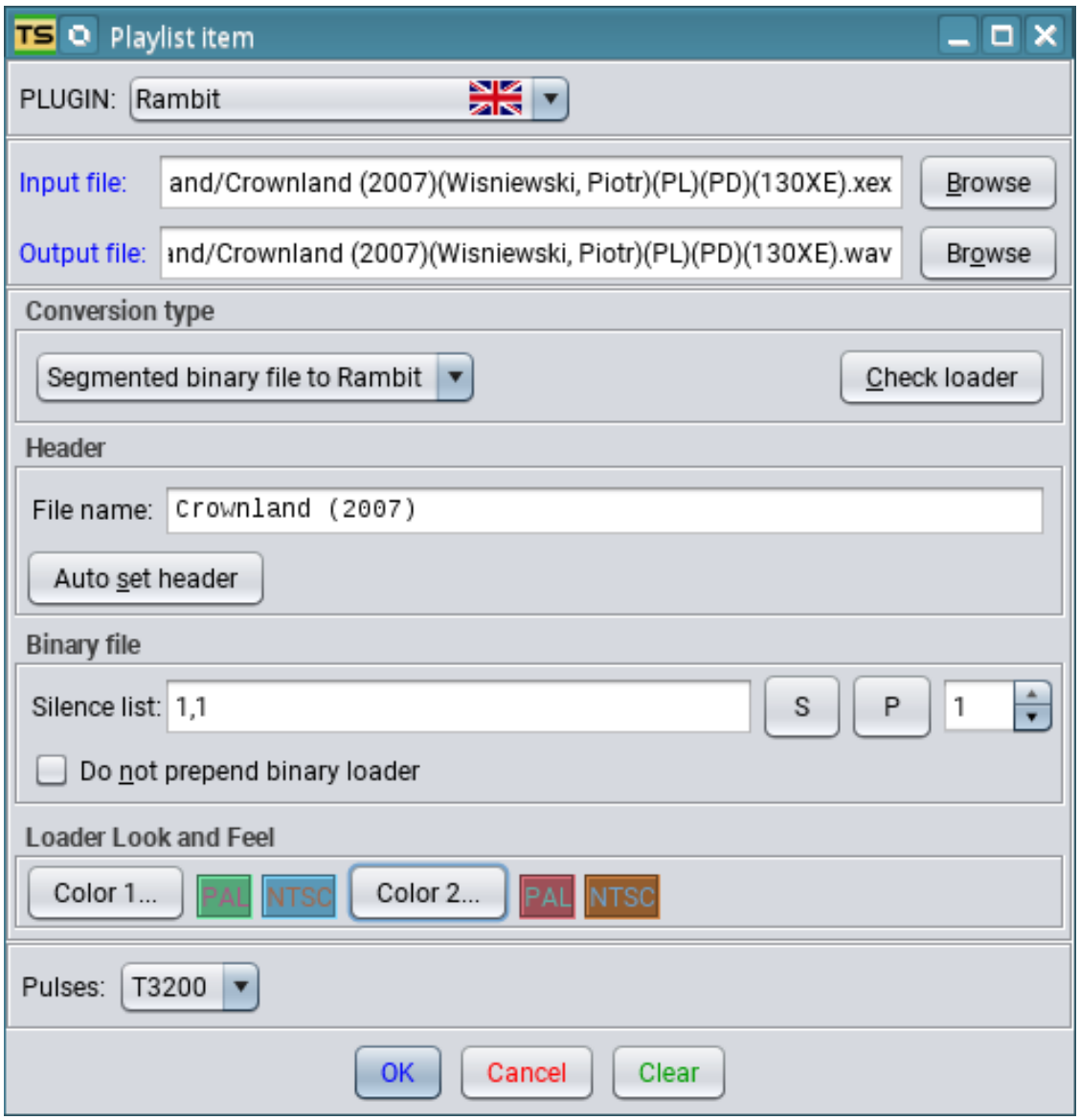

Figure 24: Rambit

Original Loaders. The original loaders were shipped with the Rambit turbo system. These loaders display characters on screen to indicate progress of the loading.

Modern Loaders. Modern re-write of the original loaders with improvements and bug fixes. The modern loaders display name of the loaded program, have adjustable look and feel, and load binary files with INIT vectors properly. These loaders display thick horizontal bars of two colors to indicate progress of the loading.

Modern Loaders 2. Other version of the modern loaders. The capabilities are the same, but the look and feel is different. These loaders display thin horizontal bars to indicate progress of the loading. Shades of a single color are displayed.

It is strongly recommended to use the modern loaders, especially for conversion of segmented binary files. The original loaders handle binary files with INIT segments poorly and also can cause screen display corruption.

21.5 Polarity of the Pulses

The Rambit turbo system is sensitive to the polarity of the pulses. With incorrect polarity, the files will not load. When the files do not load for no obvious reason, use the *Invert polarity of the pulses* configuration

entry to invert the polarity.

21.6 Configuration Entries

Rambit/Invert polarity of the pulses
Invert polarity of the pulses.

Rambit/Loaders
Determines what set of loaders will be used. Modern (default), Modern 2, or Original.

Rambit/Faster Boot
If enabled, the loaders are converted with transfer speed of 720 bd instead of 600 bd.

Rambit/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

22 Turbo 6000

22.1 Overview

This plugin converts files to the **Turbo 6000 system** from the German Democratic Republic.

22.2 Input Files

22.2.1 Compatible Binary Files

Compatible binary file can be converted to the *natural file format* of the Turbo 6000 system. Binary file compatible with Turbo 6000 must meet the following criteria:

- File size is up to 46592 bytes
- The segments in the binary file must be ordered by their first addresses
- File has exatcly one RUN segment
- FIle has at most one INIT segment

22.2.2 Binary Files

Any binary file can be converted to a special file format (Turbo 6000 - ChainLoading) that was designed to allow for loading of binary files with minimum restrictions.
The binary file is converted as follows:

- A special binary loader (Turbo 6000 - ChainLoader) is converted to the natural file format
- Binary file is converted to the special file format

22.3 User Interface

The user interface is depicted in figure 25.

Use the *Input file type* box to select input file type. Enter file name that will be displayed by the loaders in the *File name* box. Click the *Auto set header* button to automatically set the file name. Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

22.4 Polarity of the Pulses

The Turbo 6000 system is sensitive to the polarity of the pulses. With incorrect polarity, the files will not load. When the files do not load for no obvious reason, use the *Invert polarity of the pulses* configuration entry to invert the polarity.

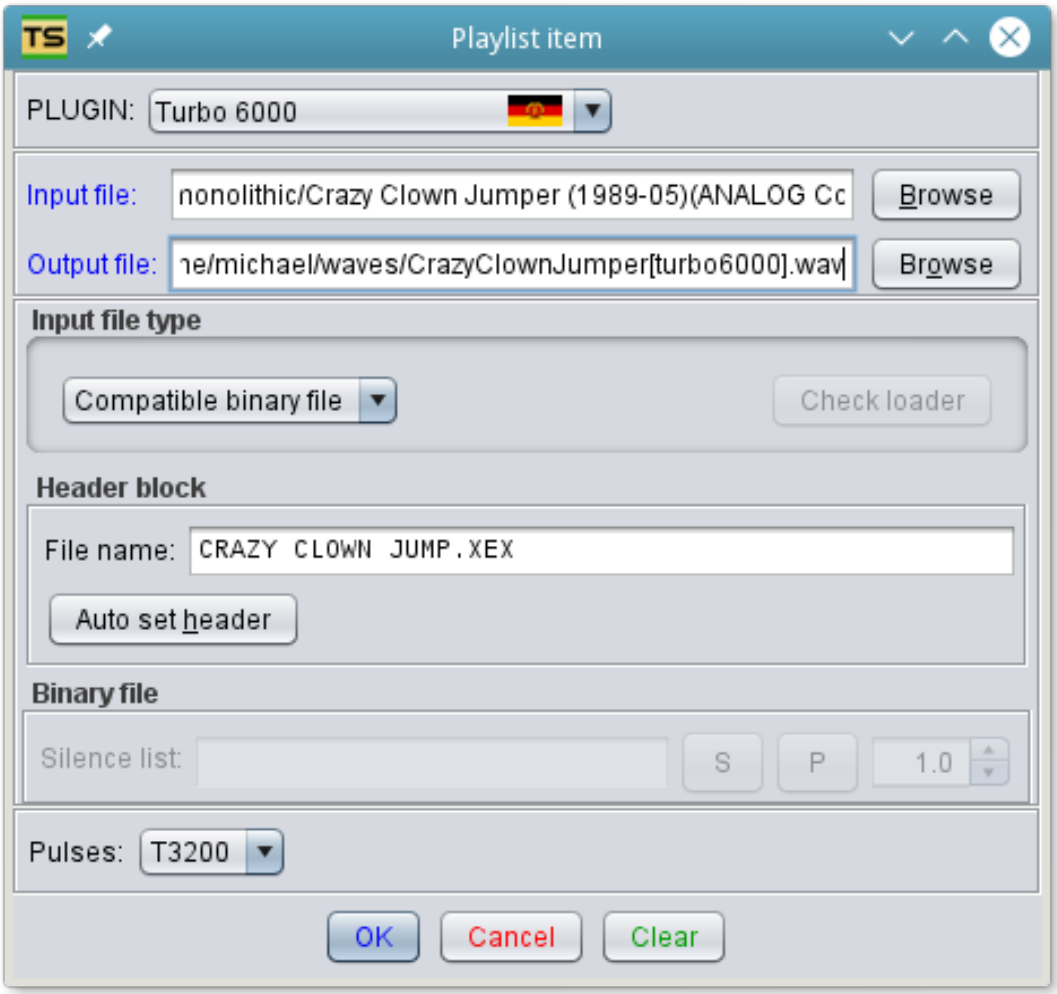

Figure 25: Turbo 6000

22.5 Sampling Rate

The Turbo 6000 plugin take advantage of the sampling rate of 48000 Hz.

22.6 Configuration Entries

Turbo 6000/Invert polarity of the pulses
Invert polarity of the pulses.

Turbo 6000/Silence after header
Number of seconds of the silence inserted after the header block (0-30)

Turbo 6000/Pause after header
If enabled, output of electric signal to sound card will be paused after the header block

Turbo 6000/Pulse Corrections
Corrections of pulses

Turbo 6000/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

23 Standard

23.1 Overview

The plugin converts input files to the standard (FSK) tape records, therefore you can use it with an unmodified data recorder. The transfer speed is limited to 600-800 bd. You can perform the following conversions to the standard tape records.

- Monolithic binary files to tape boot files
- Binary files
- Tokenized BASIC files and BASIC source code files
- Plain data files

23.2 User Interface

The user interface is depicted in figure 26.

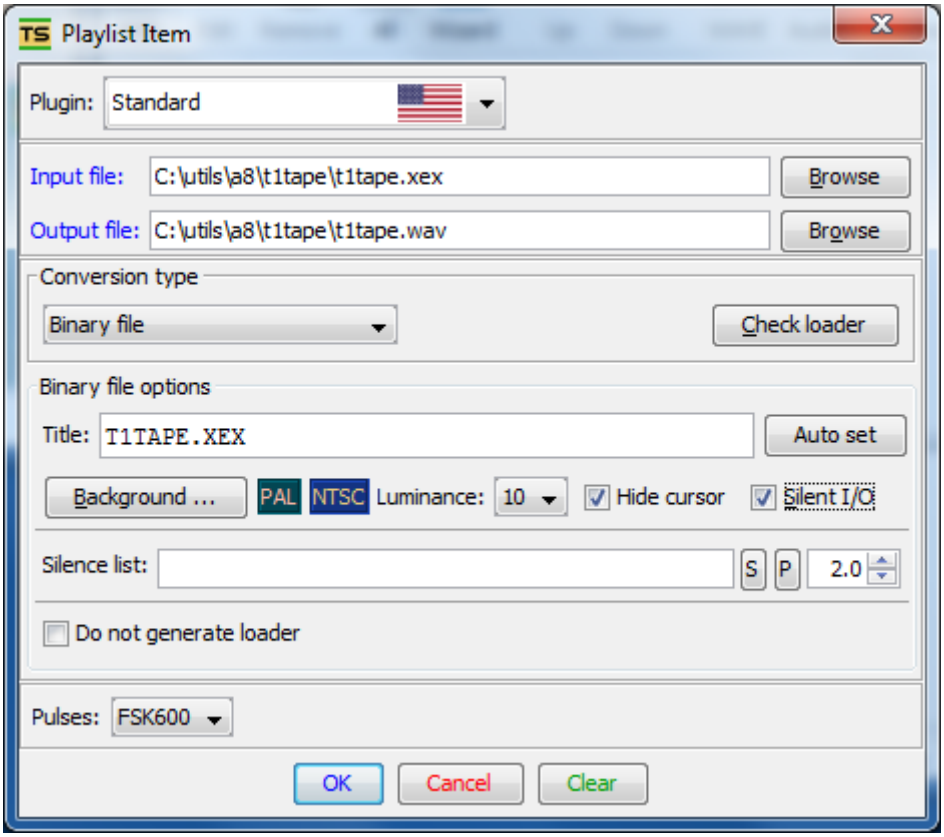

Figure 26: Standard

Use the *Conversion type* combo box to select input file type. Use controls in the *Options* boxes to set additional options for the conversion. Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Use the *Pulses* combo box to select transfer speed.

23.3 Conversion of Monolithic Binary Files to Boot Files

The monolithic binary files are converted to standard tape boot files. The following limitations apply to monolithic binary files:

- The DATA segment cannot occupy memory below address 1536
- Length of the DATA segment must not exceed 32649 bytes

If your monolithic binary file is not within the limitations, convert the file as a (general) binary file.

23.4 Conversion of Binary Files

23.4.1 Processing

The binary files are converted as follows:

- A binary loader skeleton is customized and converted to a tape boot file.
- The binary file is converted to the standard tape records with short IRGs.

23.4.2 Binary Loader Skeleton

This plugin uses the TSCBL customizable binary loader skeleton. The TSCBL binary loader skeleton is compatible with both pre-XL/XE and XL/XE computers and also displays program title on screen.

Binary files with INIT segments are fully supported. IRGs after blocks that hold INIT segments are automatically elongated to at least 1500 miliseconds (even without a silence list). The loader switches off the motor when a block with an INIT segment is read, executes the code pointed to by the INIT vector, switches the motor on and then continues loading.

23.4.3 User Interface

Use the *Title* text field to set a title that will be displayed by the binary loader. Alternatively, you can enter space-separated hexadecimal ATASCII codes preceded by the HEX keyword, e.g. `$HEX$ 32 FC 24 45 75` to display characters not available on the keyboard, or inverse characters. Click the *Auto set* button to set the title automatically. The binary loader skeleton is displaying the characters through the E: system device. The control characters are interpreted, so you can use them to position the title.

Click the *Background color* button to set background color of the loader. Use the *Luminance* box to select luminance of the title text. Select the *Hide cursor* box to hide cursor.

Select the *Silent I/O* box to disable the SIO beeping (useful when creating dual track tapes). Use the *Silence list* panel to create a silence list. Refer to section3.8 for more details.

Select the *Do not generate loader* box to convert just the binary file without loader.

Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

23.5 Conversion of Plain Data files

Plain data files are converted as they are. You can select duration of the IRGs.

Use controls in the *Plain data options* panel to select either short IRGs or long IRGs. Select the *"Use data in EOF block" trick* box to place data to EOF block. Use this special option only when converting tape boot files that require it.

23.6 Conversion of Tokenized BASIC Files

23.6.1 Formats

You can convert tokenized BASIC files to the following formats:

- Format loadable by the `CLOAD` command. Short IRGs are used.
- Format loadable by the `LOAD"C:"` or `RUN"C:"` commands. Long IRGs are used.
- Bootable format. First, a BASIC initializer (LAUNCHBAS) is converted to a boot file. Then the tokenized BASIC file is converted to the format loadable by the CLOAD command. The BASIC initializer forces a `RUN"C:"` command and also tweaks the OPEN routine to use short IRGs once. This allows you to load and run protected programs. The BASIC initializer displays the B letter on screen during the loading process.

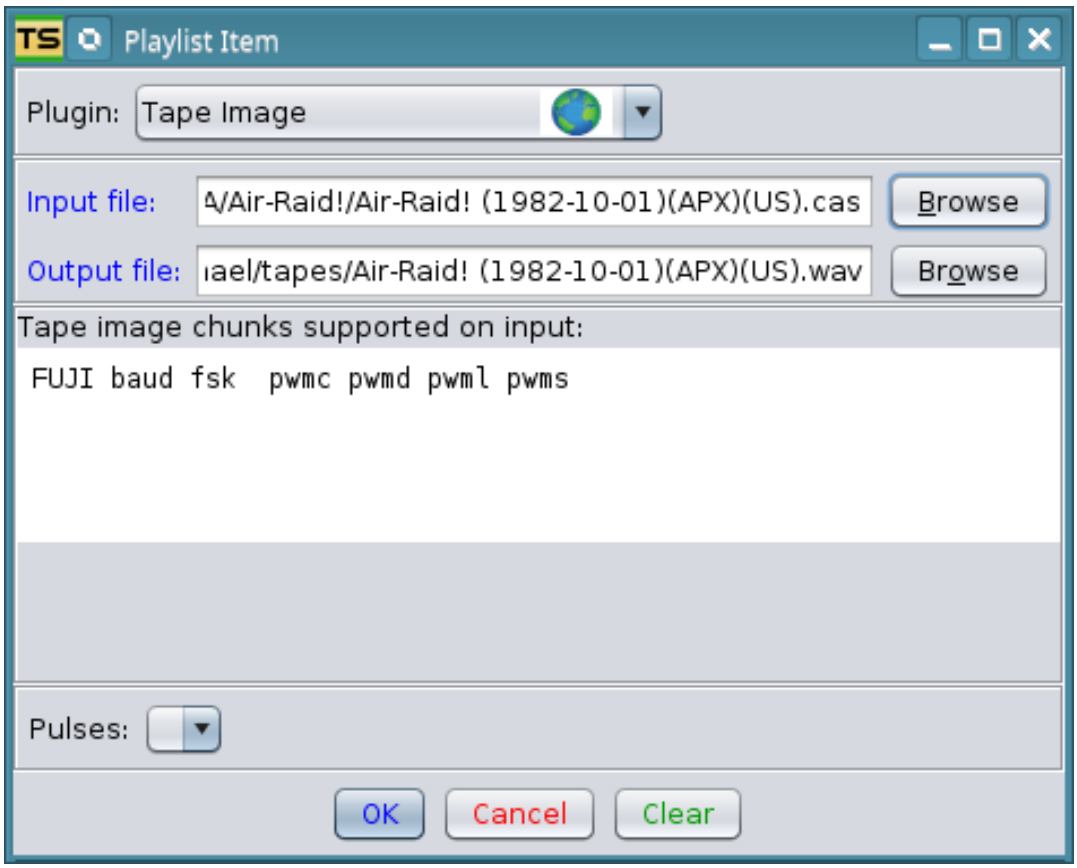

Figure 27: Tape Image

23.6.2 User Interface

Use controls in the *Tokenized BASIC options* box to select the desired file format.

23.7 Conversion of BASIC Source Files

The files are converted to a format loadable by the ENTER”C:” command.

23.8 Sampling Rate

The Standard plugin take advantage of the sampling rates of 48000 and 96000 Hz.

23.9 Configuration entries

Standard/Leader duration
Duration of the leader signal in seconds (14-30).

Standard/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 3.8.5.

Standard/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

24 Tape Image

The plugin reads and interprets tape images. The user interface displays a list of supported tape image chunks.

25 Omicron Turbo

Omicron Turbo is a special turbo system that is not associated with a particular turbo upgrade. Its design allows to use Omicron Turbo with various existing turbo upgrades.

Omicron Turbo provides a set of unifying file formats and binary loaders that will work with the following turbo upgrades (this list might not be exhaustive):

- Turbo 2000 (CZ/SK), Super Turbo (CZ/SK)
- KSO Turbo 2000 and compatibles (input from joystick port or SIO port)
- Turbo Blizzard
- Atari Super Turbo, Hard Turbo, Lower Silesian Turbo 2000, Unerring Master

Omicron Turbo was designed for creating sellable gaming software distributions on tape. With Omicron Turbo, you can distribute software on tapes with high transfer speed and great compatibility with various turbo upgrades. The only small disadvantage is a very small standard tape boot file prepended before each program.

25.1 Conversion Types

The Omicron Turbo plugin can convert binary files only.

25.1.1 BlockLoading

The input file is converted to blocks of varying lengths. A block is generated for each segment of the input binary file. Maximum number of segments is 62. A binary loader is prepended before the blocks.

25.1.2 Kilobyte Blocks

The input file is converted to 1024-byte blocks. A binary loader is prepended before the blocks.

25.1.3 Twokilobyte Blocks

The input file is converted to 2048-byte blocks. A binary loader is prepended before the blocks.

25.2 Processing

The input file is converted as follows

1. Stage 0 loader is converted to a standard tape boot file (12-second leader tone followed by 4 blocks, with data in the EOF block).
2. Binary loader (Stage 1) is converted to a single Omicron Turbo block that is read by the Stage 0 loader. This binary loader can be also loaded directly by cartridge or file version of Omicron Turbo Stage 0 loader.
3. The input binary file is converted to Omicron Turbo

25.3 User Interface

The user interface is depicted in figure 28 .

Click the *Check loader* button to determine if the binary loader can load the selected input binary file. The purpose of the text field *Silence list* is described in section 3.8.

Use the *Conversion type* combo box to select conversion type. Click the *Auto Set Header* button to automatically set program name displayed by the stage 0 loader.

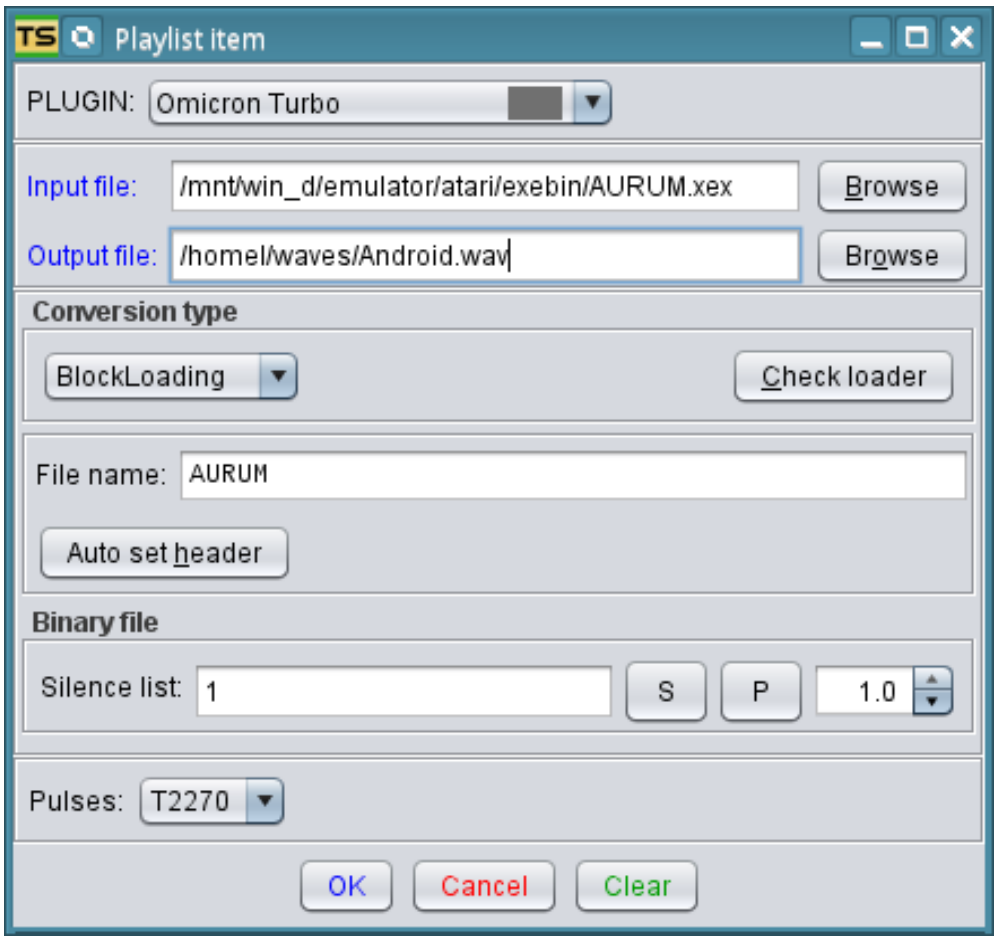

Figure 28: Omicron Turbo

25.4 Configuration Entries

Omicron Turbo/Invert polarity of the pulses
Invert polarity of the pulses

Omicron Turbo/BlockLoading pilot tone duration
Number of pilot tone pulses of data blocks for the BlockLoading conversion type (256-8192)

Omicron Turbo/Pilot tone duration for Kilobyte Blocks
Number of pilot tone pulses of data blocks for the Kilobyte Blocks and Twokilobyte blocks conversion type (256-8192)

Omicron Turbo/Prolongate pilot tone
Increase number of pilot tone pulses with increasing approximate baud rate

Omicron Turbo/Generate loader as a standard tape boot file
When enabled, the stage 0 loader is prepended as a standard tape boot file (default). When disabled, no stage 0 loader is generated (useful when cartridge or file version of Omicron Turbo will be used).

Omicron Turbo/Faster Boot
When enabled, the stage 0 loader is generated with transfer speed of 720 bd and inter-record-gaps (IRGs) that are only 180 ms long.

Omicron Turbo/File name creation options
Configure the automatic file name creation. Refer to section 3.9.

Omicron Turbo/Enable this plugin in the wizard for binary files
Makes the plugin visible to the wizard for binary files. Refer to section 3.4.

Part III

Turbo Systems

26 Introduction

This part contains information about various turbo systems that have been created in Czechoslovakia and Poland. Descriptions of the systems are sufficient to provide ability to write turbo generators, loaders, copiers and turbo decoders.

Note: Some of the information provided is not based on official documentation, but on observations and reverse engineering, thus it might be inaccurate. If you have any corrections to suggest, please visit http://turgen.sourceforge.net/support.html.

27 Information Encoding

Information is encoded using pulse width modulation (PWM). In this document, width of pulse is defined as a distance between two transitions from logical zero to logical one.

28 Turbo Systems from the Czech Republic and Slovakia

28.1 Turbo 2000

28.1.1 Description

Turbo 2000 was the first turbo system available in former Czechoslovakia. It was developed in 1987 by Jiří Richter, student of Czech Technical University in Prague and member of Prague Atari user club.

Four types of pulses are defined: Narrow pulse, wide pulse, pilot tone pulse and sync pulse. Bits are stored in MSB to LSB order.

File stored using Turbo 2000 system consists of two blocks - header block (HEADER) and data block (DATA).

Both blocks are preceded by a pilot tone (series of at least 256 pilot tone pulses) which is followed by a sync pulse (which is very narrow pulse). Recommended number of pilot tone pulses is at least 2000 in order to provide compatibility with “Universal turbo” loaders that require more pulses to measure transfer speed.

Signal is expected on the DATA-IN pin of the SIO connector.

28.1.2 Header block

Offset	Description
0	Always 0
1	File type (1 - plain data, 3 - program, 4 - binary file, 255, 254 - tokenized BASIC)
2-11	File name
12-13	Load address
14-15	Length of file
16-17	Start address
18	Check sum = (HEADER[0]) xor ... xor (HEADER[17])

28.1.3 Data block

Offset	Description
0	Always 255
1-?	Data itself
Last	Check sum = (DATA[0]) xor ... xor (DATA[?])

28.1.4 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of all pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

28.1.5 Loaders

In early times, Turbo 2000 loaders were distributed on tapes, stored using standard 600 baud system. Then loaders on cartridges became widespread.

28.2 Super Turbo

28.2.1 Description

Super turbo system is an enhancement of Turbo 2000 system, developed by Jiří Richter. Supported transfer speeds are approximately from 2725 to 6411 bauds.

Two types of pulses are defined: Narrow pulse and wide pulse. Bits are stored in MSB to LSB order.

File stored using Super Turbo consists of two blocks - header block (HEADER) and data block (DATA). Both blocks are preceded with pilot tone (series of at least 1200 wide pulses) which is followed by narrow pulse. Recommended number of pilot tone pulses is at least 2000 in order to provide compatibility with available loaders.

Signal is expected on the DATA-IN pin of the SIO connector.

28.2.2 Header block

Offset	Description
0	Always 183
1	File type (1 - plain data, 3 - program, 4 - binary file, 255 - tokenized basic)
2-21	File name
22-23	Load address
24-25	Length of file
26-27	Start address
28	Check sum = (HEADER[0]) xor ... xor (HEADER[27])

28.2.3 Data block

Offset	Description
0	Always 237
1-?	Data itself
Last	Check sum = (DATA[0]) xor ... xor (DATA[?])

28.2.4 Timing

Various transfer speeds are supported. For given speed, width of wide pulse is width of narrow pulse simply doubled.

Pulse	Width
Wide	(6/44100 - 22/44100) s
Narrow	(3/44100 - 11/44100) s

28.2.5 Loaders

Special loaders for Super Turbo only are very rare. So called “Universal turbo” loaders capable to load both Turbo 2000 and Super Turbo are widespread, mostly on cartridges. Universal turbo loaders measure speed using examination of three consequent pilot tone pulses. Measured speed is also used to distinguish between Turbo 2000 and Super Turbo.

During the time, so called “Visiloader” has been invented. This loader is able to display progress of loading using PMG.

28.3 Turbo 2000 - Kilobyte Blocks

28.3.1 Description

Turbo system designed together with various versions of “Turbo operating system” (TOS).

Four types of pulses are defined: Narrow pulse, wide pulse, pilot tone pulse and sync pulse. Bits are stored in MSB to LSB order.

File stored using this turbo system consists of many blocks. First block is a header block (HEADER), remaining blocks (BLOCK) are data blocks that have 1026 bytes. Every block is preceded by pilot tone (series of at least 256 pilot tone pulses) which is followed by one sync pulse. Recommended number of pilot tone pulses is at least 2000 in order to provide compatibility with available loaders.

Signal is expected on the DATA-IN pin of the SIO connector.

28.3.2 Header block

Offset	Description
0-1	Always 0
2-17	File name
18	Check sum = HEADER[0] xor HEADER[1] xor ... xor HEADER[17]

28.3.3 Data blocks

Offset	Description
0	255=Full block, 250=EOF block. Numbers 251-254 indicate partially filled block. If we subtract 251 from this number, we obtain a difference that will be denoted as Z. This difference can be 0,1,2 or 3 and represents higher byte of number of valid data bytes in the block.
1-1024	Data itself. If the block is a full block, there is 1024 bytes of data. If the block is EOF block, there are all zeros. If the block is a partially filled block, there is data padded with zeros up to the length of 1023 bytes. Last byte is lower byte of number of valid data bytes in the block. If we denote this lower byte as X, the number of valid bytes in the block is $Z * 256 + X$
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1024]

28.3.4 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

28.3.5 Loaders

This turbo system is integrated into to various version of “Turbo operating system” (TOS) and also to one special version of TURBO BASIC XL widely used in former Czechoslovakia. Usual CIO device installed is T: or D:. CIO functions supported are OPEN, READ, WRITE and CLOSE.

28.4 Turbo Tape

28.4.1 Description

Advanced turbo system introduced with TT-DOS operating system sold by JRC company. Aim of this turbo system is to fully replace disk drive. Usual CIO device installed is B: or D:.

Four types of pulses are defined: Narrow pulse, wide pulse, pilot tone pulse and sync pulse. Bits are stored in MSB to LSB order.

File stored using Turbo Tape consists of blocks (BLOCK) that are 1026 bytes long. Every block is preceded by pilot tone (series of at least 256 pilot tone or wide pulses) which is followed by one sync or narrow pulse. Recommended number of pilot tone or wide pulses is at least 2000 in order to provide compatibility with available loaders.

Signal is expected on the DATA-IN pin of the SIO connector.

28.4.2 Tape modes

Mode	Description
SS	Short gaps between blocks. First block is written twice, others once.
SD	Short gaps between blocks. All blocks are written twice.
LS	Long gaps between blocks. First block is written twice, others once.
LD	Long gaps between blocks. All blocks are written twice.

D modes provide data redundancy for safe data storage. First block is always written twice in order to provide convenient support for READ DIRECTORY CIO function.

28.4.3 Structure of the blocks

Offset	Description
0	Sequential number of the block. Numbering starts from 1. In case of D modes, pairs of blocks have same sequential number.
1	Tape mode: SS=128, LS=0, SD=192, LD=64
2,3	Bits 0-11: Offset of last valid byte in the block (16-1024), this offset will be denoted as B. Bit 15: Last block flag. Byte at offset 2: $B\%256$ Byte at offset 3: $[B/256] + [128 * (EOF\,is\,true)]$
4	Undefined number. All blocks of one file should have same number here.
5	Undefined.
6-16	File name and extension padded with spaces. First 8 bytes are devoted to file name, last 3 bytes are devoted to extension.
17-1024	Data itself (1008 bytes). Data must be padded with any bytes.
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1025]

28.4.4 Timing

Timing is compatible with Turbo 2000 and Super Turbo.

28.5 B-TAPE

28.5.1 Description

Advanced turbo system introduced with B-TAPE extension for operating system BW-DOS. Aim of this extension is to fully replace disk drive. B-TAPE allows to use both CIO and SIO to exploit data recorder turbo modification. The disadvantage is a big size of the device handler. B-TAPE was designed as improvement of Turbo Tape system.

Four types of pulses are defined: Narrow pulse, wide pulse, pilot tone pulse and sync pulse. Bits are stored in MSB to LSB order.

File stored using B-TAPE consists of blocks (BLOCK) that are 1026 bytes long. Every block is preceded by pilot tone (series of at least 256 pilot tone or wide pulses) which is followed by one sync or narrow pulse. Recommended number of pilot tone or wide pulses is at least 2000 in order to provide compatibility with available loaders.

Signal is expected on the DATA-IN pin of the SIO connector.

28.5.2 Tape modes

Mode	Description
SS	Short gaps between blocks. First block is written twice, others once.
SD	Short gaps between blocks. All blocks are written twice.
LS	Long gaps between blocks. First block is written twice, others once.
LD	Long gaps between blocks. All blocks are written twice.

D modes provide data redundancy for safe data storage. First block is always written twice in order to provide convenient support for READ DIRECTORY CIO function.

28.5.3 Structure of the blocks

Offset	Description
0	Sequential number of the block. Numbering starts from 1. In case of D modes, pairs of blocks have same sequential number.
1	Tape mode: SS=128, LS=0, SD=192, LD=64
2,3	Bits 0-11: Offset of last valid byte in the block (16-1024), this offset will be denoted as B. Bit 15: Last block flag. Byte at offset 2: $B\%256$ Byte at offset 3: $[B/256] + [128 * (EOF\,is\,true)]$
4	Undefined number. All blocks of one file should have same number here.
5	Random number. All blocks of one file must have same number here. This random number allows to distinguish files with same file name.
6-16	File name and extension padded with spaces. First 8 bytes are devoted to file name, last 3 bytes are devoted for extension.
17-1024	Data itself (1008 bytes). Data must be padded with zeros if needed.
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1024]

28.5.4 Timing

Timing is compatible with Turbo 2000 and Super Turbo.

28.5.5 Notes

B-TAPE device handler is able to read files stored using Turbo Tape system. In order to circumvent problems with big size of the device handler, special minimalistic binary loader called MICROB was shipped with B-TAPE.

29 Turbo systems from Poland

29.1 KSO Turbo 2000 and Turbo 2000F

29.1.1 Description

Turbo systems used in Poland, originally designed together with KSO Turbo 2000 (Tape operating system) and then adopted for other tape operating systems.

Three types of pulses are defined: Pilot tone pulse, wide pulse and narrow pulse. Bits are stored in MSB to LSB order.

File stored using KSO Turbo 2000 system consists of blocks. First block is a header block (HEADER), other blocks are data blocks (DATA).

Every block is preceded by pilot tone (series of pilot tone pulses).

Signal is expected on some pin of the second joystick port (KSO Turbo 2000) or on the DATA-IN pin of the SIO connector (Turbo 2000F).

29.1.2 Header block

Offset	Description
0	Always 0
1	Always 255
2-11	File name (10 characters)
12	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[11]) mod 256

29.1.3 Data block

Offset	Description
0-1	Number of valid bytes in the block, 0-3072.
2-3073	Up to 3072 bytes of data padded with zeros if needed.
3074	Check sum = (BLOCK[0] + ... + BLOCK[3073]) mod 256

File ends with data block that has less than 3072 valid bytes. If the total file size can be divided by 3072 without a remainder, file must end with block that has 0 valid bytes.

29.1.4 Timing

Pulse	Width
Pilot tone	44/44100 s
Wide	22/44100 s
Narrow	11/44100 s

29.2 Turbo Blizzard

29.2.1 Description

Turbo system used in Poland, suitable for storing of binary files (small blocks, high transfer speed).

Three types of pulses are defined: Pilot tone pulse, wide pulse and narrow pulse. Bits are stored in MSB to LSB order.

File stored using Turbo Blizzard system consists of the following blocks.

1. Synchronization block. This block is preceded with pilot tone (3072 pilot tone pulses) and two narrow pulses. The block does not store any data. Block is followed with silence lasting for 0.1 second.
2. Header block (HEADER). This block is preceded with pilot tone (302 pilot tone pulses) and two narrow pulses. Then 78 bytes of data follow. Block is followed with silence lasting for at least 3 seconds.
3. One or more data blocks. These blocks are preceded with pilot tone (302 pilot tone pulses) and two narrow pulses. Then 1029 bytes of data follow. Data blocks are separated by short gaps.

Signal is expected on the DATA-IN pin of the SIO connector.

29.2.2 Header block

Offset	Description
0-75	File name
76	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[75]) mod 256
77	Spare byte, always 0. This byte is never read by the loaders.

29.2.3 Data block

Offset	Description
0-1	Number of valid bytes in the block, 0-1024.
2-1025	Up to 1024 bytes of data padded with zeros if needed.
1026	Always 0
1027	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[1026]) mod 256
1028	Spare byte, always 0. This byte is never read by the loaders.

29.2.4 Timing

Pulse	Width
Pilot tone	22/44100 s 24/48000 s
Wide	12/44100 s 12/48000 s
Narrow	8/44100 s 8/48000 s

29.3 Turbo ROM

29.3.1 Description

Turbo system used in Poland, with limitations similar to Czechoslovak Turbo 2000 and Super Turbo Systems. This turbo system can store Turbo ROM compatible binary files (these binary files consist of exactly one DATA segment at most one RUN segment and at most one INIT segment) or tokenized BASIC programs.

Three types of pulses are defined: Wide pulse, narrow pulse and stop pulse. Bits are stored in LSB to MSB order.

File stored using Turbo ROM system consists of two blocks.

1. Header block (HEADER). This block is preceded with pilot tone (4884 wide pulses) and one narrow pulse. Then 41 bytes of data follow. After header block, there are four wide pulses and one stop pulse.
2. Data block. This block is preceded with pilot tone (516 wide pulses) and one narrow pulse. Then data follow. Data blocks are separated by short gaps. After data block, there are four wide pulses and one stop pulse.

Signal is expected on the DATA-IN pin of the SIO connector.

29.3.2 Header Block for Binary Files

Offset	Description
0	Header block check sum = (HEADER[1] ... xor ... HEADER[40])
1-2	Header load address (1537)
3-4	Header length excluding first byte (40)
5	Data block check sum = (DATA[0] xor DATA[1] xor ... DATA[?])
6-7	RUN address
8-9	INIT address
10-11	Load address
12-13	Data block length
14	Padding 0
15-34	File name. Internal code is used.
35	Program type flag. For binary files there is 1.
36	0 - JSR to the INIT address, 1 - No JSR to the INIT address
37-39	Padding zeros
40	RTS opcode (96)

29.3.3 Data Block for Binary Files

Offset	Description
0-?	Data bytes

29.3.4 Header block for BASIC files

A BASIC file consist of two parts. A header part (14 bytes long) and a main part. Data from the header part is stored in the header block and data from the main part is stored in the data block. The tokenized form of the BASIC file is relocatable and can be loaded to any address. Such address is denoted as base address.

Offset	Description
0	Header block check sum = (HEADER[1] ... xor ... HEADER[77])
1-2	Header load address (1537)
3-4	Header length excluding first byte (77)
5	Data block check sum = (DATA[0] xor DATA[1] xor ... DATA[?])
6-7	RUN address (41086). Points to a routine in the BASIC ROM.
8-9	INIT address. Always 0 for a BASIC program.
10-11	LOAD address. Calculated as base address + VNT.
12-13	Data block length. Length of the main part of the BASIC file.
14	Padding 0
15-34	File name. Internal code is used.
35	Program type flag. Always 0 for a BASIC program.
36	1 - No JSR to the INIT address
37-39	Padding zeros
40-59	Routine that copies the header part to the zero page `0xA2, 0x11, 0xBD, 0x3C, 0x06, 0x95,0x80, 0xCA, 0x10,0xF8,0x60`
60	Header part of the BASIC file (LOMEM, VNT, VNTE, VVT, STMTAB, STMCUR, STARP)
74-75	Address right past the loaded BASIC file (LOAD address + length of the main part)
76-77	Address right past the loaded BASIC file (LOAD address + length of the main part)

29.3.5 Data Block for BASIC Files

Offset	Description
0-?	Main part of the BASIC file

29.3.6 Timing

Pulse	Width
Wide	14/44100 s 18/48000 s
Narrow	6/44100 s 8/48000 s
Stop	48/44100 s 52/48000 s

29.3.7 Special Notes

The Turbo ROM loading routines calculate check sum *after a block is fully loaded*. This can cause false check sum errors for blocks that are loaded to the I/O area (0xD000 - 0xD7FF).

29.4 Atari Super Turbo (AST)

29.4.1 Description

Turbo system used in Poland that can store binary files. Note that the following information is incomplete.

Three types of pulses are defined: Wide pulse, narrow pulse and STOP pulse. Bits are stored in LSB to MSB order.

Signal is expected on the DATA-IN pin of the SIO connector.

29.4.2 Block

Blocks have the following structure:

1. Pilot tone (narrow pulses)
2. Narrow pulse
3. Data (wide and narrow pulses)
4. Terminator 1(4 narrow pulses)
5. Terminator 2 (128 STOP pulses)

29.4.3 AST File Format

This file format can be used to store binary files that have up to 44 segments and no INIT segments. File consists of several blocks.

1. Header block, 256 bytes of data
2. Data blocks. For each segment of the binary file, there is one data block.

29.4.4 BUT File Format

This file format can be used to store binary files. File consists of several blocks.

1. BUT loader header block. 6 bytes of data. It appears to be a boot header.
2. BUT loader data block. 576 bytes of data.
3. Pairs of data blocks for each segment. First block in a pair is a segment header, second block in a pair holds segment data.
4. Termination segment header[2]

29.4.5 Unerring Master Atari Turbo System

This turbo system can be used to store binary files. A file consists of several blocks.

1. Header block
2. Binary loader block
3. Binary file data blocks. Each data block stores one or more segments of a binary file. A block ends either with a special termination header or with an INIT segment.

[2] Turgen System is using slightly different format. There is no termination segment header, but the loader knows the number of segments.

29.4.6 AST Header Block

Offset	Description
0	Number of segments (1-44)
1	Header check sum Check sum = (HEADER[2] xor HEADER[3] +xor... HEADER[255])
2-177	44 segment headers (start and end addresses)
178	Always 102
179	Always 85
180-199	File name. Internal code is used.
200	Always 0
201-244	Check sums for each segment block Check sum = (DATA[1] xor DATA[2] +xor... DATA[?])
245-255	Unknown. 0x70, 0x4b, 0x00, 0xd6, 0x47, 0xb4, 0xcf, 0x4b, 0x00, 0xd6, 0x41.

29.4.7 BUT Loader Header Block

Offset	Description
0	Always 0
1	Length of the BUT loader data block in 128 bytes blocks. Always 5.
2,3	Loader load address. 1920.
4,5	Unknown. 182, 9.

29.4.8 BUT Loader Data Block

Offset	Description
0-575	BUT loader code

29.4.9 AST and BUT Data Block

Offset	Description
0-?	Data

29.4.10 BUT Termination Segment Header

Offset	Description
0-3	255, 255, 255, 255

29.4.11 UM Header Block

Offset	Description
0-255	Header title and file name. All 255 bytes are displayed on screen beginning from SAVMSC. Internal code is used for the characters displayed. Offset | Description 167-198 | File name 56 | Always 'U'. The loader checks this field
256,257	Load address of the next block
258,259	Length of the next block
260,261	Run address. DOSINI is set to this address.
262	Checksum of the next block (EOR of all bytes of the block)
263-269	Always 0

29.4.12 UM Binary Loader Block

Offset	Description
0-?	Binary loader code and data

29.4.13 UM Binary File Data Block

Offset	Description
[	This part of the block can repeat
0,1	Segment header 1 (first address of a segment) Special value 0xDE,0xDE indicates end of file, no segment header 2 and no segment data.
2,3	Segment header 2 (last address of a segment).
4-?	Data of the segment
]	End of part that can repeat
?+1	Checksum (EOR of all previous bytes of the block)

29.4.14 Timing

Pulse	Width
Wide	22/44100 s
Narrow	12/44100 s
Stop	72/44100 s

29.5 Hard Turbo

29.5.1 Description

Turbo system used in Poland that can store only binary files.

Three types of pulses are defined: Pilot tone pulse, wide pulse, and narrow pulse. Bits are stored in MSB to LSB order.

File stored using Hard Turbo system consists of many blocks. There are three types of blocks.

1. Main header block (HEADER). This block is preceded with pilot tone (at least 512 pilot tone pulses) and one narrow pulse. Then 41 bytes of data follow.
2. Segment header block (SEGHEAD). This block is preceded with pilot tone (at least 512 pilot tone pulses) and one narrow pulse. Then 6 bytes of data follow.
3. Segment data block (DATA). This block is preceded with pilot tone (at least 512 pilot tone pulses) and one narrow pulse. Then data bytes follow.

A Hard Turbo file starts with the main header block. Then pairs of segment header and segment data blocks follow. The file ends with one special segment header block.

Signal is expected on the DATA-IN pin of the SIO connector.

29.5.2 Main Header Block

Offset	Description
0	Identification byte, always 0
1-39	File name. ATASCII string terminated with EOL (155)
40	Check sum = (HEADER[0] xor HEADER[1] xor ... HEADER[39])

29.5.3 Segment Header Block

Offset	Description
0	Identification byte, always 255
1,2	First address of the following segment. Value of 65535 (255 255) indicates end of file
3,4	Last address of the following segment (increased by one)
5	Check sum = (SEGHEAD[0] xor SEGHEAD[1] xor ... SEGHEAD[4])

29.5.4 Segment Data Block

Offset	Description
0	Identification byte, always 255
1-?	Segment data
Last	Check sum = (DATA[0] xor DATA[1] xor ... DATA[?])

29.5.5 Timing

Pulse	Width
Pilot	34/44100 s
Wide	22/44100 s
Narrow	12/44100 s

29.6 Lower Silesian Turbo 2000

29.6.1 Description

This turbo system appears to be very similar to Turbo 2000 from former Czechoslovakia.

Four types of pulses are defined: Narrow pulse, wide pulse, pilot tone pulse and sync pulse. Bits are stored in MSB to LSB order.

Signal is expected on the DATA-IN pin of the SIO connector.

There are several file formats used.

29.6.2 Auto Turbo Format

This format is used to store binary files with limitations. Binary files without INIT segments can be stored and executed. Binary files with INIT segments can be stored, but not executed.

File stored using Auto Turbo format consists of two blocks - header block (HEADER) and data block (DATA).

Both blocks are preceded by pilot tone (series of at least 256 pilot tone pulses) which is followed by a sync pulse (which is very narrow pulse). Recommended number of pilot tone pulses is at least 2000.

29.6.3 Unknown Exterminator Unprotected Binary Format and Protected Binary Format (UE UBF and UE PBF)

These file formats can be used to store binary files. A binary loader is prepended before the file. File consists of several blocks.

1. Dummy RUN block. 2 bytes of data (UE PBF only).
2. Pairs of data blocks for each segment. First block in a pair is a segment header (SEGHEAD), second block in a pair holds segment data (SEGDATA).
3. Termination segment header

29.6.4 Auto Turbo Header block

Offset	Description
0	Always 0
1	File type. Always 4.
2-11	File name
12-13	Always 0
14-15	Length of file
16-17	Always 0
18	Check sum = (HEADER[0]) xor ... xor (HEADER[17])

29.6.5 Auto Turbo Data block

Offset	Description
0	Always 255
1-?	Data of the binary file (includes segment headers)
Last	Check sum = (DATA[0] xor ... xor DATA[?])

29.6.6 UE PBF Dummy RUN block

Offset	Description
0,1	0x02 0xE0
2	Check sum = 0xE2

29.6.7 UE UBF and UE PBF Segment Header Block

Offset	Description
0,1	First address of the following segment
2,3	Last address of the following segment
4	Check sum = (SEGHEAD[0] xor ... xor SEGHEAD[3])

29.6.8 UE UBF and UE PBF Segment Data Block

Offset	Description
0-?	Data
Last	Check sum = (SEGDATA[0] xor ... xor SEGDATA[?])

29.6.9 UE UBF and UE PBF Termination Segment Header

Offset	Description
0-3	0x00,0x00,0x00,0x00
4	Check sum = 0x00

29.6.10 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of all pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

30 Turbo Systems from Other Countries

30.1 Rambit Turbo Tape System from UK

Turbo system from United Kingdom that can store monolithic and segmented binary files.

Two types of pulses are defined: wide and narrow. Bits are stored in MSB to LSB order. Signal is expected on pin #13 of the SIO connector (INTERRUPT). The default polarity of the pulses is HIGH-LOW.

Rambit Turbo System defines two file formats

30.1.1 Standard Boot Format

File stored using this format consists of one single block (BLOCK). The block is preceded by pilot tone (series of at least 8 wide pulses) which is immediately followed by data. Length of the block, load address, and run address are not stored in the block, but in a loader.

Offset	Description
0	Identification byte. Always 0x5A
2-?	Data (length may vary)
?+1	Check sum = (BLOCK[2] xor ... xor BLOCK[?])
?+2	Termination marker. Always 0x00,0x00,0x00,0x00

30.1.2 Pseudo-Disk Binary File

File stored using this format consists of one or more blocks (BLOCK). The blocks are preceded by pilot tone (series of at least 8 wide pulses) which is immediately followed by data. New block is created after each INIT segment. This allows proper loading of segmented binary files with INIT segments. There is also no check sum.

Offset	Description
0	Identification byte. Always 0x5A
1-2	Always 0x00, 0x00
[	This part of the block can repeat
3-4	First address of a segment. If there is special value of 0x??, 0xFF, the block ends right here, INIT segment is executed and new block follows
5-6	Last address of a segment If there is special value of 0x00,0x00, end of file was reached
7-?	Data of the segment
]	End of part that can repeat

30.1.3 Timing

Transfer speed is approximately 3200 bauds.

Pulse	Width
Wide	19/44100 s
Narrow	10/44100 s

30.2 Turbo 6000 from GDR

This turbo system can store compatible binary files (refer to section 22.2).

Two types of pulses are defined: wide and narrow. Bits are stored in MSB to LSB order. Signal is expected on pin #9 of the SIO connector (PROCEED). The default polarity of the pulses is LOW-HIGH.

One file format is defined.

30.2.1 Natural File Format

File stored in the natural file format consists of a header block (HEADER) and a data block (DATA) that holds a binary file (including binary file header and segment headers).

Both blocks are preceded with pilot tone, synchronization sequence and a marker byte.

The Turbo 6000 loader works as follows:

1. Load the data block to a memory area determined by the header
2. Read segments of the binary file from the memory area and place the data of the segments to memory locations determined by the segement headers
3. Execute initialization code via the INITAD vector
4. Run the binary file via the RUNAD vector

This processing imposes severe limitations to the binary file:

1. Maximum binary file size is limited to 46592 bytes
2. The binary file is limited to one INIT segment
3. The binary file must have a RUN segment
4. The segments in the binary file must be ordered by their first addresses

30.2.2 Header Block

Offset	Description
	Pilot tone. Series of at least 64 0x02 bytes.
	Synchronization sequence 0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x01
	Marker byte. Must be nonzero
0,1	First address of buffer (BUFRLO,BUFRHI)
2,3	Last buffer address (BFENLO,BFENHI) increased by 1
4-23	File name (internal code)
24	Padding byte (always 0)

30.2.3 Data Block

Offset	Description
	Pilot tone. Series of at least 64 0x02 bytes.
	Synchronization sequence 0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x01
	Marker byte. Must be zero
0-?	Bytes of the binary file (including segment headers)
?+1	Checksum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[?]

30.2.4 Timing

Pulse	Width
Wide	14/44100 s 16/48000 s
Narrow	8/44100 s 10/48000 s