

TURGEN SYSTEM DOCUMENTATION

version 8.2.8-01

Michael Kalouš

Contents

I	TURGEN SYSTEM	6
1	Introduction	6
1.1	Primary Functionality	6
1.2	Auxiliary Functionality	6
1.3	Program Characteristics	6
1.4	System Requirements	6
1.5	Target Audience	6
1.6	Plugins and Turbo systems	6
1.7	Conventions and Terminology	6
2	Distribution and Installation	8
3	Operations Guide	8
3.1	Starting TURGEN SYSTEM	8
3.2	Program Controls	9
3.3	Conversion of Files to Turbo Systems	10
3.3.1	Playlist Items	10
3.3.2	Working with Playlist	10
3.3.3	Playlist Item Manipulation	10
3.4	Wizard for Binary Files	11
3.5	Producing Output	11
3.5.1	Output of Electric Signal into WAVE File	11
3.5.2	Output of Electric Signal to Computer's Audio System.	12
3.5.3	Output of Tape Image	12
4	Program Configuration Facility	12
4.1	Modifying Configuration	12
4.2	General Configuration Entries	12
4.3	Output of Electric Signal to Computer's Audio System	12
4.4	Output of Electric Signal into WAVE File	13
4.5	Output of Tape Images	13
5	Advanced Settings	13
5.1	Active Plugins	13
5.2	Repository of Pulses	14
6	Tools	14
6.1	Merging Segments of Binary Files	14
6.1.1	Introduction	14
6.1.2	Merging Segments Step by Step	14
6.2	Turbo Decoder	15
6.2.1	Principles and Capabilities	15
6.2.2	Decoder Operations	15

7	Appendices	16
7.1	Hints and Tips	16
7.2	Data Recorders	16
7.3	Data Recorder Replacements	16
7.4	Postprocessing	16
7.4.1	Postprocessing of WAVE Files	17
7.4.2	Postprocessing of Tape Images	17
7.4.3	Examples	17
7.5	Data Transfer via Serial Port in Collaboration with CAS COM Utility	17
7.5.1	Introduction	17
7.5.2	Command Line Considerations	17
7.5.3	Temporary Files	17
II	BUILT-IN PLUGINS	18
8	Turbo 2000 and Super Turbo	18
8.1	Characteristics	18
8.2	Conversion types	18
8.3	Header Block	19
8.4	Inserting Silence After INIT Segments	19
8.5	Configuration Entries	19
9	Turbo 2000 - kilobyte blocks	20
9.1	Characteristics	20
9.2	User Interface	20
9.3	Binary Loaders	20
9.4	Configuration Entries	20
10	B-TAPE	21
10.1	Characteristics	21
10.2	User Interface	21
10.3	Binary Loaders	22
10.4	Configuration Entries	22
11	KSO Turbo 2000	22
11.1	Characteristics	22
11.2	User Interface	23
11.3	Configuration Entries	23
12	Turbo Blizzard	23
12.1	Characteristics	23
12.2	User Interface	24
12.3	Configuration Entries	24
13	Turbo ROM	24
13.1	Characteristics	24
13.2	User Interface	25
13.3	Configuration Entries	25
14	Atari Super Turbo	25
14.1	Characteristics	25
14.2	User Interface	26
14.3	Configuration Entries	26

15 Tape Image	26
III TURBO SYSTEMS	27
16 Introduction	27
17 Information encoding	27
18 Turbo Systems from Former Czechoslovakia	27
18.1 Common Information	27
18.1.1 Switching Data Recorder to Turbo Mode	27
18.1.2 Single Purpose or CIO	27
18.1.3 Pilot Tone and Data Separation	27
18.2 Turbo 2000	28
18.2.1 Description	28
18.2.2 Header block	28
18.2.3 Data block	28
18.2.4 Timing	28
18.2.5 Loaders	28
18.3 Super Turbo	29
18.3.1 Description	29
18.3.2 Header block	29
18.3.3 Data block	29
18.3.4 Timing	29
18.3.5 Loaders	29
18.4 Turbo 2000 - kilobyte blocks	30
18.4.1 Description	30
18.4.2 Header block	30
18.4.3 Data blocks	30
18.4.4 Timing	30
18.4.5 Loaders	30
18.5 Turbo Tape	31
18.5.1 Description	31
18.5.2 Tape modes	31
18.5.3 Structure of the blocks	31
18.5.4 Timing	31
18.6 B-TAPE	31
18.6.1 Description	31
18.6.2 Tape modes	32
18.6.3 Structure of the blocks	32
18.6.4 Timing	32
18.6.5 Notes	32
19 Turbo systems from Poland	33
19.1 KSO Turbo 2000 and Turbo 2000F	33
19.1.1 Description	33
19.1.2 Header block	33
19.1.3 Data block	33
19.1.4 Timing	33
19.2 Turbo Blizzard	33

19.2.1	Description	33
19.2.2	Header block	34
19.2.3	Data block	34
19.2.4	Timing	34
19.3	Turbo ROM	34
19.3.1	Description	34
19.3.2	Header block for binary files	34
19.3.3	Data block	35
19.3.4	Timing	35

Part I

TURGEN SYSTEM

1 Introduction

1.1 Primary Functionality

TURGEN SYSTEM is an acronym for “Turbo generating system”. It is a program whose main purpose is conversion of files (with focus on binary files) to various turbo systems used in former Czechoslovakia and Poland.

Output of the program is an electric signal that can be stored in WAVE files or can be sent directly to computer's audio system. Such electric signal can be then transferred to compact cassettes and read by data recorders equipped with particular turbo system.

TURGEN SYSTEM focuses on conversion of binary files. For such purpose, set of special miniature turbo loaders was developed and integrated to the program. Also a “Wizard for binary files” is available to provide convenience during the conversion preparation.

1.2 Auxiliary Functionality

Apart from primary functionality which is conversion of files to various turbo systems, TURGEN SYSTEM offers auxiliary functions for convenience. Such auxiliary functionality includes: 1. Turbo decoder which can be used to retrieve information from tapes, 2. Output of tape images and interpretation of tape images, 3. Data transfer via serial port in collaboration with CAS COM utility, 4. Tool for merging segments of binary files.

1.3 Program Characteristics

TURGEN SYSTEM is written in Java™ 2 programming language and it is free software distributed under the terms of GNU General Public Licence, version 2.

1.4 System Requirements

- Java Runtime Environment 1.4.2 or newer
- Solaris, Microsoft Windows 98/2000/ME/XP/Vista/7, GNU/Linux with kernel series 2.4 or 2.6

1.5 Target Audience

Target audience represents owners, users and fans of Atari 8-bit computers who use data recorders equipped with some turbo system (or some replacement devices) for data storage.

1.6 Plugins and Turbo systems

TURGEN SYSTEM provides relatively flexible architecture to support various turbo systems, using *plugins*. Each turbo system supported is handled by one plugin.

1.7 Conventions and Terminology

Conventions

If not explicitly noted, numbers are *decimals*.

Abbreviation *TS* is used for TURGEN SYSTEM.

Java Runtime Environment is abbreviated to *JRE*. Java Development Kit is abbreviated to *JDK*.

Terminology

Binary files¹. Files designed to store *programs* for Atari 8-bit computers. Binary files were introduced with Atari DOS and have specific, well-defined structure. Their purpose is the same as purpose of load modules on IBM mainframes or ELF files under GNU/Linux operating systems.

Binary files consist of *segments*, that are data blocks augmented with four byte header. The header specifies addresses where the contents of the particular segment will be placed when binary file is loaded.

There are special purpose segments loaded to the addresses 736-739, they are denoted as *jump segments*. There are three types of them summarised in table 1. Other segments (ordinary, common) will be denoted as *data segments*.

Note that other file formats were developed for program storage, for example those used by SpartaDOS or BW-DOS. These files are *not supported* by TURGEN SYSTEM.

Address	Name of jump segment and its description
736-737	RUN segment. When all segments are loaded, jump to the address specified by contents of the segment will be performed (JMP (736) in terms of assembler). Presence of RUN segment is not mandatory. If omitted, jump to the beginning of the first data segment is performed.
738-739	INIT segment. After this segment is loaded, jump to subroutine to the address specified by contents of the segment will be performed. In terms of assembler, virtual JSR (738) is simulated.
736-739	RUNINIT segment. Combination of the previous jump segments.

Table 1: Jump segments

Monolithic binary files. Binary files with a very simple structure. They consist of exactly one data segment and at most one RUN segment.

Segmented binary files. Binary files that are not monolithic.

Binary loaders. Programs, or routines (usually parts of operating systems) that load and run binary files.

Binary load. Name for the process of loading and running of binary files. This process is performed by binary loaders.

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

Data recorder. Device designed to read or write signal to or from compact cassettes, specially designed to be connected to computers, e.g. Atari XC-12.

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.

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

¹Binary files are sometimes referenced as Binary load files

Turbo system. This term denominates a system designed to speedup data transfer of original Atari data recorder. Typical turbo system consist of three parts: 1. Hardware modification of data recorder, 2. Software using such hardware modification, 3. File format. There is not necessarily 1:1:1 relation. For one hardware modification of data recorder, there can exist more software or more file formats.

TURGEN SYSTEM restricts this term to systems that use PWM (pulse width modulation) to encode data. Some examples of turbo systems are Turbo 2000 or Turbo Blizzard. Example of system that is not considered to be turbo system from the TS's point of view is IRON TURBO.

2 Distribution and Installation

Preliminary Operations

Before installing TURGEN SYSTEM, install Java Runtime Environment. JRE can be downloaded from the following address: <http://java.sun.com>. Download the newest available version of JRE for your operating system.

Distribution

Program is distributed in two forms:

1. Bzipped tarball. This form is devoted for users that do not use supported versions of Microsoft Windows or those who want to avoid overhead of automated installers for some reasons.
2. Automated installer for Microsoft Windows. This installer provides convenience and creates shortcuts in the Start menu or on desktop. Shortcuts won't be created if JRE (or JDK) is not installed.

Installation and Directories.

Installation represents unpacking of the bzipped tarball or running the automated installer. During the installation, program directory is created and populated with files.

The program directory contains all necessary files required for running TURGEN SYSTEM. TS never writes to this directory, it is recommended to set read-only access to this directory in order to provide security.

The configuration directory is created by TS when needed (usually during the first run). Its location is operating system dependent and TS requires read-write access for full functionality, however it can work if the read-write access is not granted. Location of this directory can be displayed by selecting *About* item from *Turgen* menu, if needed.

3 Operations Guide

3.1 Starting TURGEN SYSTEM

Various Ways How to Start

A general way how to start TS is to run launcher of interpret of Java language and pass it the file that holds byte-code of TS, which is `turgen.jar` located in the program directory. General form of a command line that accomplishes this task is the following:

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

Users of Microsoft Windows can use shortcuts created by the automated installer, or take benefit from the fact, that automated installer of JRE associates Java archives with launcher of the interpret, so TS can be started just by double-clicking `turgen.jar`'s icon.

Users of other operating systems can create shortcuts by means of their window managers or desktop environments for convenience.

Command Line Parameters

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

Log File

Exceptions, traces or other important information is stored to 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 reachable from the main menu.

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

Main Control Buttons

Under the playlist, there is a panel, where the *main control buttons* are located. The most used functions of TS are reachable using these buttons. Every button has its icon and name. The buttons are enumerated 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 to turbo systems can be monitored here or also stopped. At the very bottom of the program window, there is a status bar.

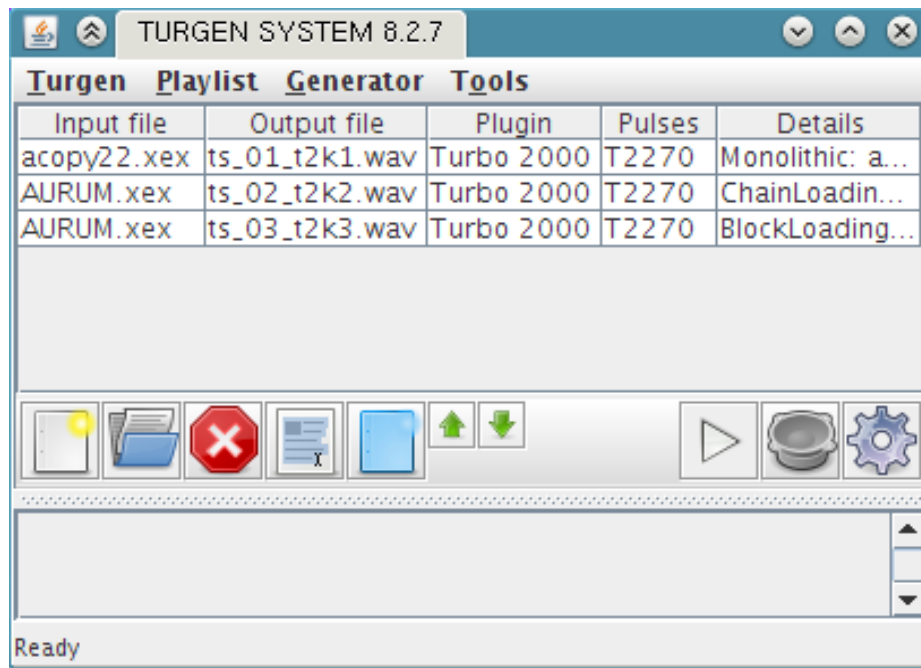


Figure 1: Program window

3.3 Conversion of Files to Turbo Systems

3.3.1 Playlist Items

In order to convert some file to turbo system, TURGEN SYSTEM requires information about such conversion which includes input file name, turbo system, output file name, what loader to prepend and various parameters specific to the particular turbo system.

Such information is held by *playlist items*, which can be somehow considered *unit of work*. Playlist items are elements of the *playlist*.

3.3.2 Working with Playlist

You can add playlist item to the playlist using the *Add item to playlist* main control button. Playlist items can be edited after clicking the *Edit selected playlist item* main control button. The *Select all playlist items* main control button will select all playlist items. Selection can be also performed using mouse dragging or keyboard. To remove selected playlist items, click the *Remove selected playlist items* main control button. Such functionality is accessible also from the *Playlist* menu.

Single playlist items can be moved using the *Move 1 playlist item up* and the *Move 1 playlist item down* buttons.

Playlist can be saved to file and retrieved. The items *Load* and *Save* from *Playlist* menu can be used to perform these operations.

3.3.3 Playlist Item Manipulation

TURGEN SYSTEM provides a dialog for playlist item manipulation which is depicted on figure 2.

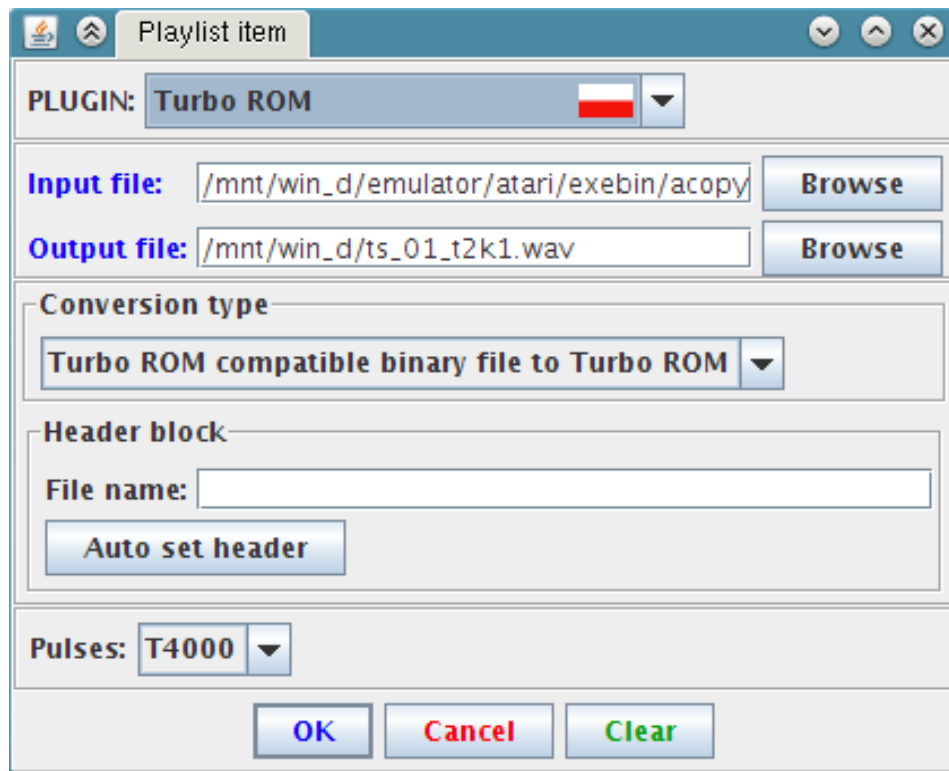


Figure 2: Manipulating playlist item

Use the Combo box *PLUGIN* to choose plugin that will be used for conversion to turbo system. There is a plugin for each turbo system supported by TURGEN SYSTEM. The names of plugins correspond to names of turbo systems. National flags can be used to distinguish from where particular turbo systems come from.

In the top part of the dialog, there is a common panel that is used to set input and output file. If the label *Input file* is clicked, a dialog with information about input file is shown. If the *Output file* label is clicked, the text field *Output file* is filled automatically, using Favourite output directory configuration entry.

In the middle part, there is a plugin-specific panel. Buttons at the bottom can be used to commit or cancel playlist item manipulation. The *Clear* button resets controls in the dialog, the effect is also plugin dependent.

3.4 Wizard for Binary Files

Wizard for binary files provides convenience when converting binary files to turbo systems. The wizard is started by clicking the *Wizard for binary files* main control button.

In the first step, the wizard asks the user to specify a binary file that he wants to convert and what are the capabilities of his data recorder. During the second step, the binary file is analyzed and the user is offered with a list of available methods of conversions to turbo systems.

The wizard saves laborious calculations, because the offered methods are ordered by pilot tone consumption and wizard checks (finitely of course) whether the binary file won't destroy a built-in binary loader that will be occasionally prepent.

3.5 Producing Output

3.5.1 Output of Electric Signal into WAVE File

To output electric signal into a WAVE file, click the *Generate WAVE file* main control button. A separate thread for each selected playlist item is created. These threads run in parallel and it is possible to stop them using 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 Computer's Audio System.

To output electric signal to computer's audio system, click *Generate AUDIO directly* main control button. Playlist items are processed serially. It is possible to stop the output using *Stop* and *Stop all* buttons that will appear in the progress monitoring panel.

3.5.3 Output of Tape Image

To output a tape image, click the *Generate tape image* main control button. Playlist items are processed serially. The output files are overwritten without a warning. The format of tape image can be configured using Program configuration facility described in section 4.

4 Program Configuration Facility

Program configuration facility provides various program parts with configuration capabilities in a uniform way. Program configuration consists of *configuration entries* that are grouped in *configuration classes*. Program parts use API to retrieve contents of configuration entries, user is provided with *Preferences dialog* which he can use to modify configuration. To provide persistence, configuration is stored in a properties file. Some configuration entries are used by the 'core' of TURGEN SYSTEM, others are used by plugins.

4.1 Modifying Configuration

There are two ways how the configuration can be modified:

1. Usage of the *Preferences* dialog. To make a modification permanent, it is necessary to save the configuration by selecting the *Save preferences* menu item from the *Tools* menu. This is the preferred way how to modify the configuration.
2. Manual adjustment of the `turgen.properties` file that can be found in the configuration directory. This is deprecated way and won't be described in this documentation.

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, 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 installed)
com.sun.java.swing.plaf.mac.MacLookAndFeel (only Mac, untested)
com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel (new since JDK 1.6 update 10)
```

TURGEN SYSTEM/Favourite output directory

Favourite directory for output files.

4.3 Output of Electric Signal to Computer's Audio System

Audio output/Silence between files

Length of silence in milliseconds generated between playlist items.

Audio generator/Amplitude

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

Audio generator/Channels

Number of channels.

Audio generator/Bits per sample

Number of bits per one sample.

Audio generator/Sampling rate

Sampling rate in Hz. It can be decreased to 22050 in case of audio system incapability, but it is not recommended.

Audio generator/Signed samples

Indicates whether to use signed samples.

Audio generator/Initial silence

Length of silence generated after audio system initialization, unit is 0.1 of second. This may be needed, because the very first data sent to the audio system can produce clicks or cracks.

Audio generator/Terminal silence

Length of silence generated before audio system termination. This may be needed, because the very last data sent to the audio system can be truncated.

Audio generator/Buffer size

Size of the buffer for data being sent to the audio system. Increasing this value can help when there are under runs and output from the audio system is repeatedly interrupted.

4.4 Output of Electric Signal into WAVE File

WAVE generator/Amplitude

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

WAVE generator/Channels

Number of channels.

WAVE generator/Bits per sample

Number of bits per one sample.

WAVE generator/Sampling rate

Sampling rate in Hz. It can be decreased to 22050 in case of audio system incapability, but it is not recommended.

WAVE generator/Postprocessing command

Command line for postprocessing. For more information refer to section 7.4.1.

WAVE generator/Change extension to .wav

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

4.5 Output of Tape Images

Tape image output/Change extension to .cas

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

Tape image generator/Postprocessing command

Command line for postprocessing. For more information refer to section 7.4.2.

Tape image generator/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 generator/Tape image chunks

This entry is used to specify, what type of chunks will be placed into generated tape images. trXX style chunks or pwmX can be specified. trXX style chunks are used by CASCOS software, pwmX style chunks are used by A8CAS project.

5 Advanced Settings

5.1 Active Plugins

The list of plugins that will be loaded is stored in the `plugins.list` file. Plugins can be disabled by removing appropriate lines or commenting them out by putting the # (hash) character to the beginning of the line. Order of the plugins can be changed just by reordering the lines, which can be useful for users from Poland.

When disabling plugins, user must take into account that plugins are also used to handle the trXX chunks in input tape images.

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 TS is restarted, because the repository is read only once.

6 Tools

6.1 Merging Segments of Binary Files

6.1.1 Introduction

TURGEN SYSTEM is equipped with a tool that merges segments of binary files into one segment and adds extra code that emulates the effect of occasional jump segments. This allows to convert segmented binary files that meet certain conditions to monolithic binary files.

This tool was created in early times of TURGEN SYSTEM in order to partially cope with limitations of Czechoslovak Turbo 2000 and Super Turbo turbo systems.

Conditions that the segmented binary files must meet are the following: 1. Data segments must not overlap, 2. Routines called by the jump segments must work even when all data segments are already loaded, 3. There must be place for code that emulates effect of occasional jump segments. Usually, the only way how to test this conditions is to run the resulting monolithic binary file in some emulator. It should be noted that capabilities of this tool must be considered limited.

6.1.2 Merging Segments Step by Step

Display the dialog for merging segments of binary file by selecting item *Merge segments of binary file* from the *Tools* menu.

Fill-in the *Input file* text field and press the *Analyze* button. The list of segments will be filled-in and you will be able to check whether the first condition is met. Then fill-in the *Output file* text field and specify address of the extra code that will emulate the effect of jump segments. The extra code is not needed only when there are no INIT or RUNINIT segments in the binary file.

Click the *Merge !* button to merge the segments and create the output binary file. Test the created binary file using some emulator.

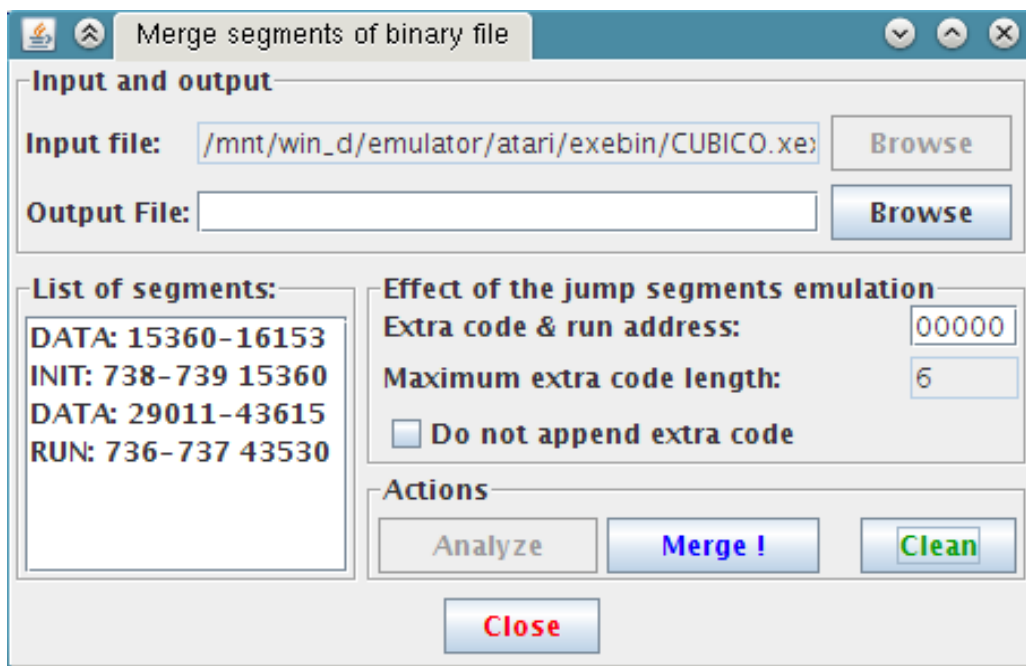


Figure 3: Merging segments of binary file

6.2 Turbo Decoder

6.2.1 Principles and Capabilities

Turbo decoder is a tool designed to retrieve data from compact cassettes. The decoding algorithm is a Java rewrite of turbo loaders from assembler 6502.

The electric signal can be read from WAVE files, these must be obtained by sampling of signal coming from cassette recorder connected to computer's audio system. It is recommended to set the cassette recorder to produce maximum signal amplitude. The WAVE file must be in the following format: PCM, 1 channel, 44100 Hz, 8 bits per sample.

Another possibility is to read the electric signal directly from the computer's audio system. This source of electric signal is available, but not recommended, because turbo decoding is CPU intensive process and the signal is coming in real time, so data loss can occur.

Turbo systems supported by the decoder are the following: Turbo 2000, Turbo 2000 - kilobyte blocks, Super Turbo and Turbo Tape or B-TAPE.

The decoder can also work as so called Turbo monitor (reading of Turbo 2000 or Super Turbo blocks) capable of reading blocks up to the size of 64 KB.

6.2.2 Decoder Operations

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

Select 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 press the *Attach decoder* button.

If the source of the electric signal is the computer's audio system, make sure that it is properly configured (capture is enabled and desired input line is selected - use means of your operating system), appropriate channel (mono, left, right) is selected and press the *Attach decoder* button.

Panels *Navigation* and *Decoding* will be enabled. Then fill-in the output directory, which must already exist. Using the *Navigation* panel, you can specify current position in the WAVE file.

Controls on the *Decoding* panel are devoted to perform decoding. Combo box *Decoder* is used to select the expected turbo system, buttons *Decode one file* and *Decode until EOF* will start decoding of one file or all files until the end of the WAVE file is reached or the *Stop* button is pressed.

Decoding can be stopped using the *Stop* button. When the decoder window is closed, decoding is stopped too.

If the source of the electric signal is the computer's audio system, the emergency stop can be performed if the SHIFT key is pressed together with the *Stop* button. This is useful when decoder hangs due to bad audio system setup. After an emergency stop, decoder must be detached and attached again to be operative again.

You can see the results of decoding in the *Log* tab. If you collaborate with some digitized sound editor, you may use the number of sample displayed in the {} style parenthesis.

Various decoding parameters can be set using the *Settings* tab.

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

7 Appendices

7.1 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 how to reduce loading times is to reduce length of pilot tones.
- Don't forget that you can resize program windows as you want. Bounds of the windows are persistent, so if you exit the TS 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 losses.

7.2 Data Recorders

In case you are using data recorder and compact cassettes for data storage, use fast transfer rates with care. It is perfectly OK to use fast transfer rates for single data transfers, but it is not OK in case of data preservation, because compact cassettes tend to age and the signal can degrade or disappear.

You must use standard compact cassettes, these are usually marked as:

IEC I/TYPE I/NORMAL BIAS/NORMAL POSITION

Compact cassettes marked as CHROME, METAL or HIGH POSITION are not suitable for data recorders.

7.3 Data Recorder Replacements

During the time, various data recorder replacements were invented. Some of them are simple adapters that allow to connect cassette recorders or compact disc players, some of them use built-in portable music players.

Most of such devices lack the MOTOR CTRL signal handling. This makes sometimes the binary load operation difficult to accomplish, because execution of routines invoked from INIT segments can take some time. Partial remedy to this problem is usage of the silence lists that allow to generate silence after blocks that hold INIT segments.

7.4 Postprocessing

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

To reference the TURGEN SYSTEM's output in the command line, some special symbols are introduced. Meaning of the symbols is 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 rest are parameters passed to it. 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

7.4.1 Postprocessing of WAVE Files

Postprocessing of WAVE files can be switched on using the *Wave postprocessing* check box menu item from the *Generator* menu. Command line is specified in the Wave generator/Postprocessing command configuration entry.

7.4.2 Postprocessing of Tape Images

Postprocessing of tape images can be switched on using the *Tape image postprocessing* check box menu item from the *Generator* menu. Command line is specified in the Tape image generator/Postprocessing command configuration entry.

7.4.3 Examples

Example of 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 xmms media player. The original file is deleted:

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

7.5 Data Transfer via Serial Port in Collaboration with CAS COM Utility

7.5.1 Introduction

CAS COM is a small utility for loading tape images to Atari via serial port. It supports standard and Turbo 2000 cassette images. CAS COM works with SIO2PC, ATART or compatible hardware interfaces.

CAS COM can be used as postprocessor of tape images created by TURGEN SYSTEM. Since CAS COM is currently available only for Microsoft Windows operating systems, the following text presumes that such operating system is used.

CAS COM has a web page: <http://sdq.czweb.org/atari/index.html>.

7.5.2 Command Line Considerations

Let us presume that CAS COM resides in C:\UTILS\ATARI directory. The configuration entry Tape image generator/Postprocessing command should have the following value:

```
cmd /x C:\UTILS\CASCOM\CASCOM.EXE %ODS%%OFNE%
```

It is important to add command line interpreter to the command line, because CAS COM uses console I/O to display important information. Users of Microsoft Windows 95/98/98SE/ME should use `command` instead of `cmd`.

CAS COM accepts other usefull command line parameters. /S starts tape image loading immediately, /e terminates program after data transfer. Moreover /1, /2, /3, /4 or /c<num> parameters allow to choose serial port.

7.5.3 Temporary Files

Setting Tape image generator/Auto create temporary files configuration entry can be useful when user's intention is only to transfer data via serial port.

Part II

BUILT-IN PLUGINS

In this part, functionality of the plugins that are shipped with TURGEN SYSTEM is described.

8 Turbo 2000 and Super Turbo

8.1 Characteristics

Both plugins convert input files to classic Czechoslovak turbo systems **Turbo 2000** and **Super Turbo**. Both plugins also provide means to circumvent main limitation of both systems - the incapability to hold segmented binary files. User interface is depicted on figure 4.

The screenshot shows a window titled "Playlist item" with a tab icon and standard window controls. The "PLUGIN:" dropdown is set to "Turbo 2000" with a Czech flag icon. Below are fields for "Input file:" (containing "/mnt/win_d/emulator/atari/exebin/acopy22.xex") and "Output file:" (containing "/mnt/win_d/ts_01_t2k1.wav"), each with a "Browse" button. The "Conversion type" section has a dropdown set to "Monolithic binary file to Turbo 2000" and a "Check loader" button. The "Header block" section contains fields for "File name:" (acopy22.xe), "Type:" (3), "Load:" (46621), "Length:" (2531), and "Run:" (46621), along with an "Auto set header" button. The "Binary file" section has a "Silence list:" text area. At the bottom, the "Pulses:" dropdown is set to "T2270", and there are "OK", "Cancel", and "Clear" buttons.

Figure 4: Turbo 2000, Super Turbo

8.2 Conversion types

In order to cover most of the needs, plugins Turbo 2000 and Super Turbo offer up to 6 conversion types. Choice can be made using the *Conversion type* combo box.

Monolithic binary file to Turbo 2000 or Super Turbo. Conversion of monolithic binary file. The input file must be monolithic binary file. This is classic usage of Turbo 2000 or Super Turbo systems.

ChainLoading. Conversion of segmented binary file to *chain of Turbo 2000 or Super Turbo files*. Generally, each segment of the binary file is converted to one file. Special binary loader that loads the files consequently is prepent before the chain. The binary file can have up to 252 segments. ChainLoading represents circumvention of incapability of Turbo 2000 or Super Turbo to hold segmented binary files. The *Check loader* button can be used to determine whether the special binary loader will be destroyed by the binary file.

BlockLoading. Conversion of segmented binary file to *chain of Turbo 2000 or Super Turbo blocks*. Each segment of the binary file is converted to one block. Special binary loader that loads the blocks consequently (and hold information where to put the blocks in memory) is prepent before the chain. The binary file can have up to 62 segments. BlockLoading represents circumvention of incapability of Turbo 2000 or Super Turbo to hold segmented binary files. The *Check loader* button can be used to determine whether the special binary loader will be destroyed by the binary file.

Tokenised BASIC to Turbo 2000 or Super Turbo. Conversion of BASIC program in tokenised form. Not all loaders are able to run such program directly.

Plain DATA to Turbo 2000 or Super Turbo. Conversion of plain data. Usually, there is a need for manual setup of header block.

Binary file to binary turbo. Conversion of binary file to so called “Binary turbo” (file type 4). This conversion type is available only for the Turbo 2000 plugin. Binary files converted this way usually cannot be loaded directly, VisiCopy III loader is capable of doing this, but there should not be INIT segments in the binary file.

8.3 Header Block

The text fields *File name*, *Type*, *Load*, *Length* and *Run* can be used to specify appropriate entries of the Turbo 2000 or Super Turbo header block. This is rarely needed.

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

8.4 Inserting Silence After INIT Segments

The text field *Silence list* can be filled with comma separated decimal numbers. Every number is interpreted as number of seconds of silence to be inserted after corresponding INIT segments. This is applicable only for ChainLoading and BlockLoading conversion types.

8.5 Configuration Entries

Turbo 2000/Header block pilot tone length
Number of pilot tone pulses of the header block (256-8192)
Turbo 2000/Data block pilot tone length
Number of pilot tone pulses of the data block (256-8192)
Turbo 2000/Pilot tone length for BlockLoading
Number of pilot tone pulses of data blocks for BlockLoading conversion type (256-8192)
Super Turbo/Header block pilot tone length
Number of pilot tone pulses of the header block (256-8192)
Super Turbo/Data block pilot tone length
Number of pilot tone pulses of the data block (256-8192)
Super Turbo/Pilot tone length for BlockLoading
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

9 Turbo 2000 - kilobyte blocks

9.1 Characteristics

Plugin converts files to **Turbo 2000 - kilobyte blocks** Czechoslovak turbo system. There are no special restrictions on input files.

9.2 User Interface

The user interface is depicted on figure 5.

The text field *File name* corresponds to the same field of the Turbo 2000 - kilobyte blocks header block. The meaning of the text field *Silence list* is described in section 8.4.

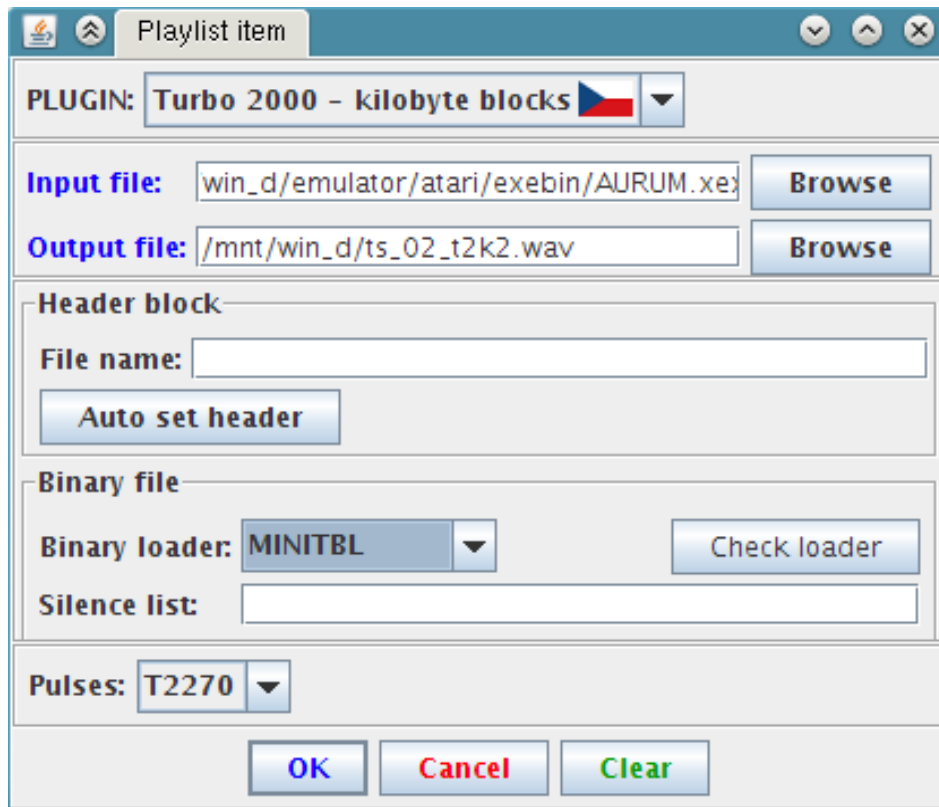


Figure 5: Turbo 2000 - kilobyte blocks

9.3 Binary Loaders

Turbo 2000 - kilobyte blocks turbo system is suitable to hold binary files. There is a possibility to prepend one of the special miniature built-in binary loaders. For this turbo system, MINITBL, NANOTBL, NANOTBL[UR] and NANOTBL[U2] can be chosen using the *Binary loader* combo box. The prepend loader is converted to the Turbo 2000 system.

MINITBL loader is a combination of stripped "T:" device handler that allows only READ operation and code for binary load using CIO. 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".

The *Check loader* button can be used to verify whether the binary file will destroy the selected binary loader.

9.4 Configuration Entries

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

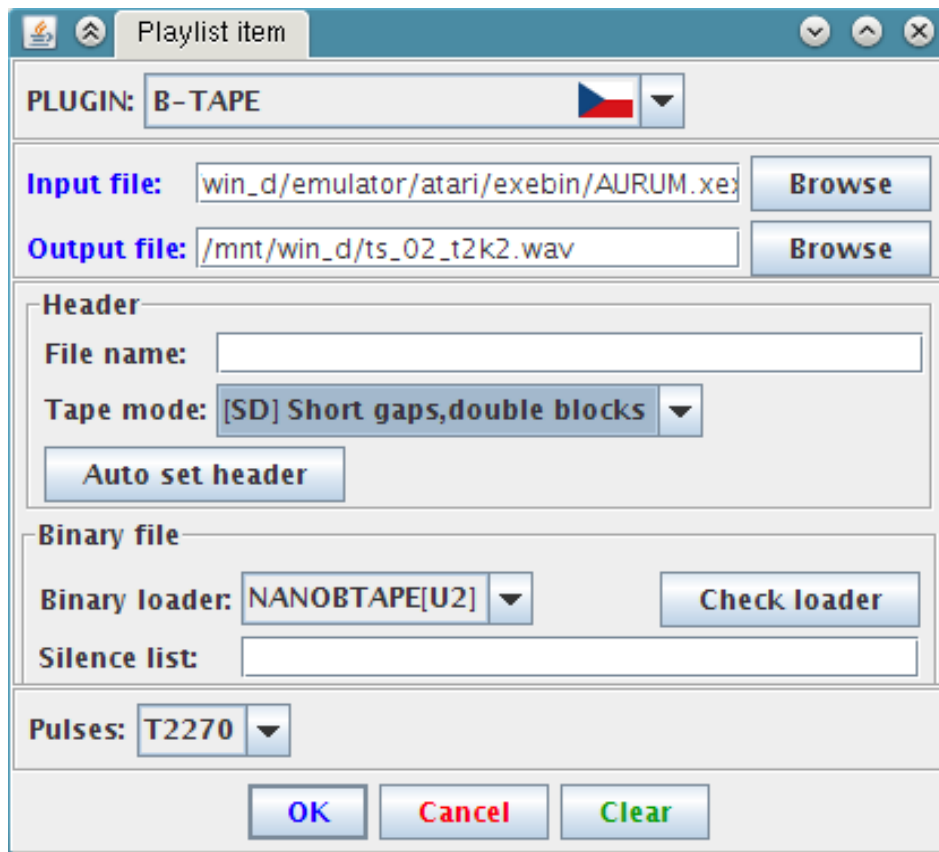


Figure 6: B-TAPE

Turbo 2000 - kilobyte blocks/Data block pilot tone length

Number of pilot tone pulses of the data blocks (256-8192)

Turbo 2000 - kilobyte blocks/Loader header block pilot tone length

Number of pilot tone pulses of the header block of the binary loader (256-8192)

Turbo 2000 - kilobyte blocks/Loader data block pilot tone length

Number of pilot tone pulses of the data block of the binary loader (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/Name loader same as file

The binary loader will have name same as the converted file

10 B-TAPE

10.1 Characteristics

Plugin converts files to **B-TAPE** Czechoslovak turbo system. There are no special restrictions on input files. B-TAPE system is backward compatible with Turbo Tape system.

10.2 User Interface

The user interface is depicted on figure 6.

The text field *File name* corresponds to the same field of the B-TAPE data block. The meaning of the text field *Silence list* is described in section 8.4. The *Tape mode* combo box allows to select tape mode.

10.3 Binary Loaders

B-TAPE turbo system is suitable to hold binary files. There is a possibility to prepend one of the special miniature built-in binary loaders. For this turbo system, NANOBTAPe, NANOBTAPe[UR] and NANOBTAPe[U2] can be chosen using the *Binary loader* combo box. The prepent loader is converted to the Turbo 2000 system.

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

The *Check loader* button can be used to verify whether the binary file will destroy the selected binary loader.

10.4 Configuration Entries

B-TAPE/Pilot tone length

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/Loader header block pilot tone length

Number of pilot tone pulses of the header block of the binary loader (256-8192)

B-TAPE/Loader data block pilot tone length

Number of pilot tone pulses of the data block of the binary loader (256-8192)

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/Name loader same as file

The binary loader will have name same as the converted file

11 KSO Turbo 2000

11.1 Characteristics

Plugin converts files to **KSO Turbo 2000** Polish turbo system. There are no special restrictions on input files. KSO Turbo 2000 system is compatible with various similar turbo systems (e.g. Turbo 2000F).

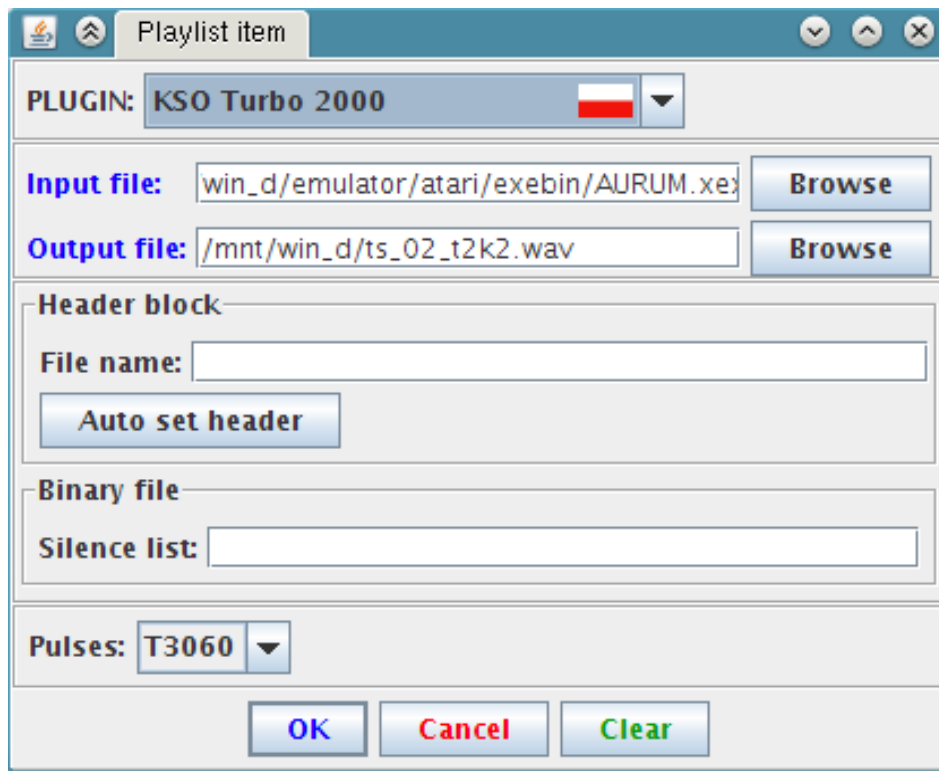


Figure 7: KSO Turbo 2000

11.2 User Interface

The user interface is depicted on figure 7.

The text field *File name* corresponds to the same field of the header block. The meaning of the text field *Silence list* is described in section 8.4.

11.3 Configuration Entries

KSO Turbo 2000/Header block pilot tone length
 Number of pilot tone pulses of the header block (256-8192)
 KSO Turbo 2000/Data block pilot tone length
 Number of pilot tone pulses of the data blocks (256-8192)
 KSO Turbo 2000/Invert polarity of pulses
 Invert polarity of pulses

12 Turbo Blizzard

12.1 Characteristics

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

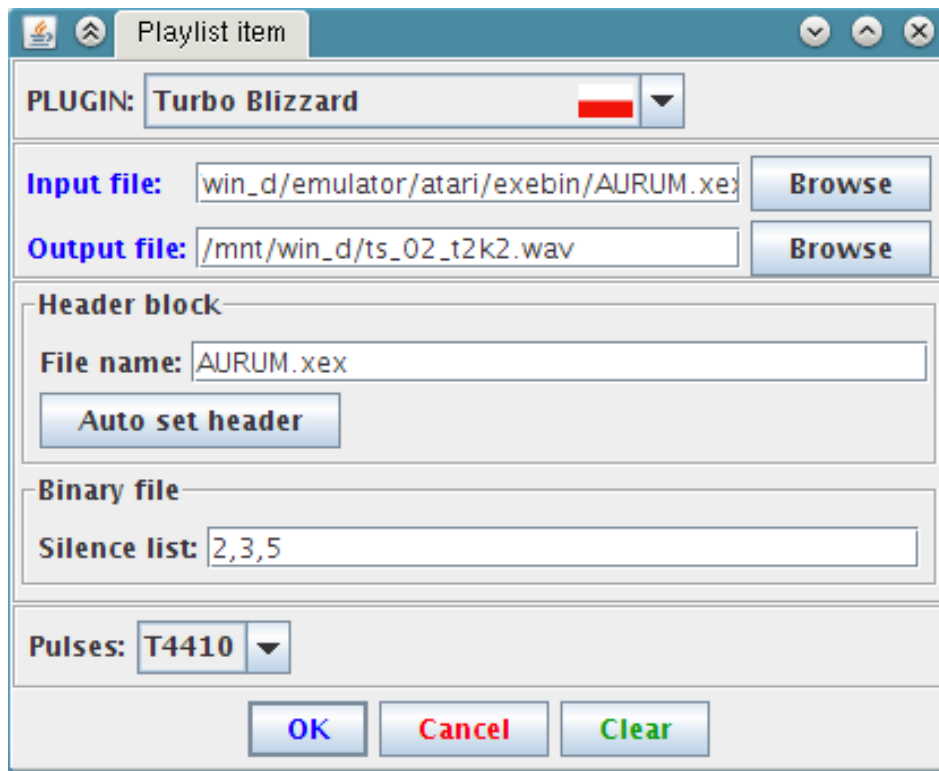


Figure 8: Turbo Blizzard

12.2 User Interface

The user interface is depicted on figure 8.

The text field *File name* corresponds to the same field of the header block. The meaning of the text field *Silence list* is described in section 8.4.

12.3 Configuration Entries

Turbo Blizzard/Invert polarity of pulses
 Invert polarity of pulses
 Turbo Blizzard/Long gaps between blocks
 Make long gaps between blocks

13 Turbo ROM

13.1 Characteristics

Plugin converts files to **Turbo ROM** Polish turbo system. Only conversion of Turbo ROM compatible binary files is supported (those binary files consist of exactly one DATA segment at most one RUN segment and at most one INIT segment).

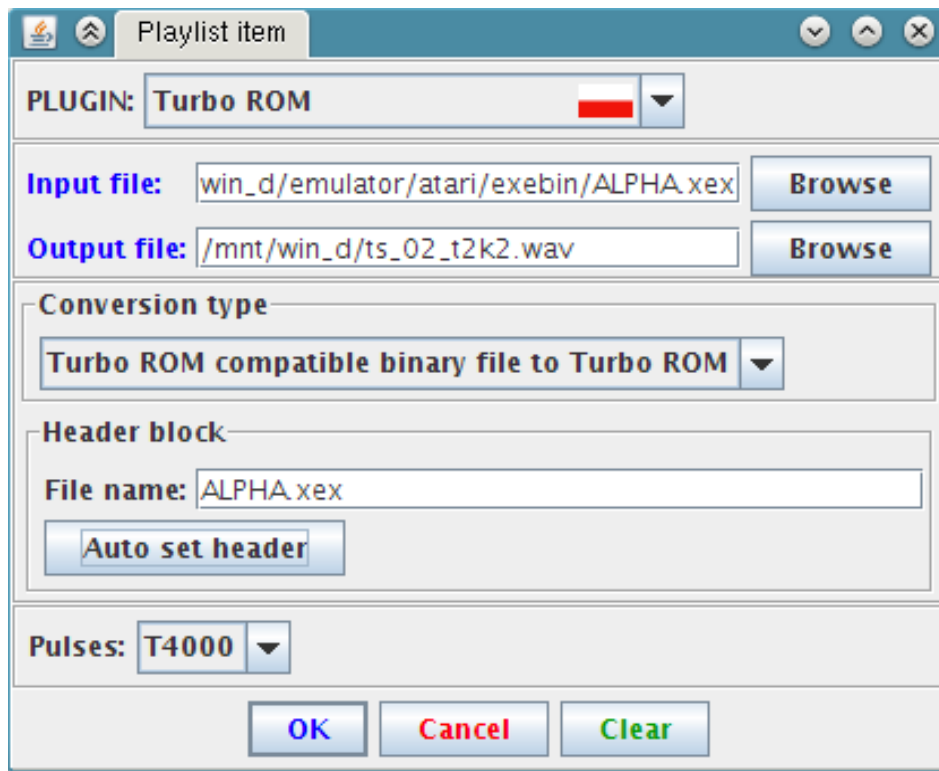


Figure 9: Turbo ROM

13.2 User Interface

The user interface is depicted on figure 9.

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 Turbo ROM compatible binary file.

13.3 Configuration Entries

Turbo ROM/Invert polarity of pulses
Invert polarity of pulses

14 Atari Super Turbo

14.1 Characteristics

Plugin converts files to **Atari Super Turbo** Polish turbo system. Conversion of binary files to the following formats is supported:

- AST format that can hold binary files that have up to 44 segments and no INIT segments.
- BUT format that can hold 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.

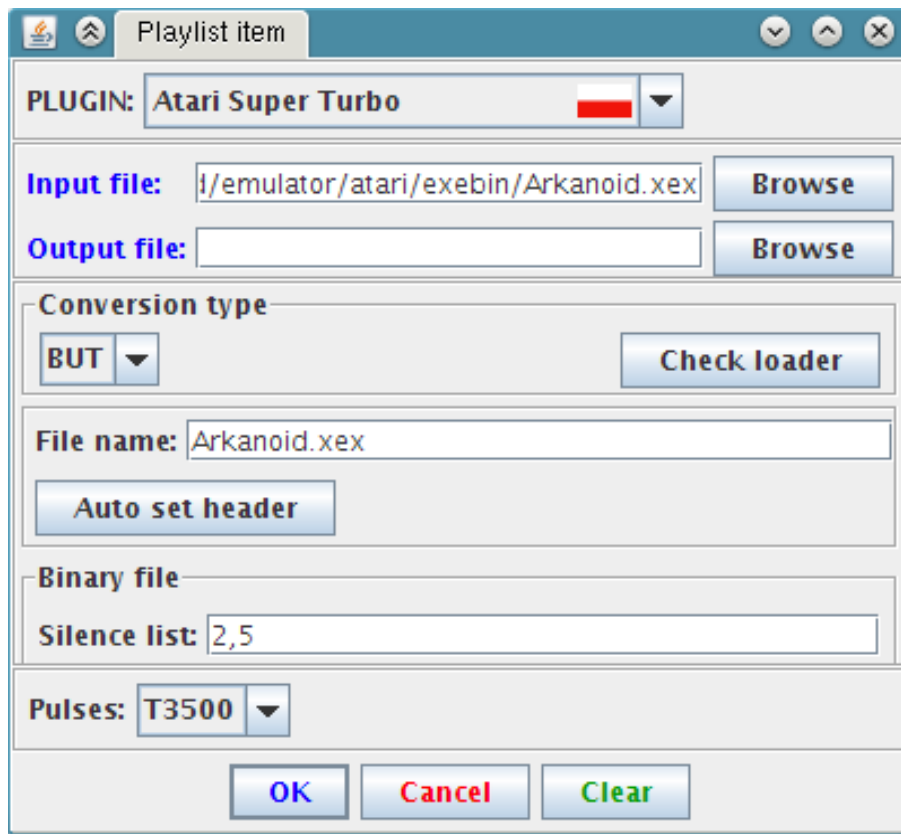


Figure 10: Atari Super Turbo

14.2 User Interface

The user interface is depicted on figure 10.

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 format (AST or BUT). The meaning of the text field *Silence list* is described in section 8.4.

14.3 Configuration Entries

Atari Super Turbo/Invert polarity of pulses

Invert polarity of pulses

Atari Super Turbo/AST header pilot tone length

Number of pilot tone pulses of the AST header block (256-16384)

Atari Super Turbo/Data block pilot tone length

Number of pilot tone pulses of data blocks (256-16384)

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 BUT loader

15 Tape Image

Plugin interprets tape images. Only trXX tape image chunks are supported.

Part III

TURBO SYSTEMS

16 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.

17 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. See figure 11.



Figure 11: Pulse and its width

18 Turbo Systems from Former Czechoslovakia

18.1 Common Information

This common information is related only to turbo systems described in this document. These turbo systems are or were widespread in former Czechoslovakia, but of course, there are or were also others.

18.1.1 Switching Data Recorder to Turbo Mode

Switching data recorder to turbo mode is done using electronic switch, that switches to turbo mode if signals COMMAND and MOTOR CTRL are active. Reading of data is done by monitoring signal at pin DATA-IN of the SIO connector. Writing of data is performed by direct change of logical value at DATA-OUT pin of the SIO connector.

18.1.2 Single Purpose or CIO

Two main kinds of Czechoslovak turbo systems can be distinguished:

1. Systems that use single purpose loaders and simple file format that can hold only one contiguous block of data (Turbo 2000, Super Turbo). These systems are unable to support binary load operation, although capable to hold binary load files themselves.
2. Systems that were created together with tape operating systems: Turbo 2000 - kilobyte blocks, Turbo Tape, B-TAPE. Aim of these systems is to partially or fully replace the disk drive.

18.1.3 Pilot Tone and Data Separation

All Czechoslovak turbo systems use same way how to separate pilot tone and data. One sync pulse (which is very narrow pulse) or narrow pulse.

18.2 Turbo 2000

18.2.1 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 distinguished: 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 pilot tone (series of at least 256 pilot tone pulses) which is followed by 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.

18.2.2 Header block

Offset	Description
0	Always 0
1	File type (1 - plain data, 3 - program, 4 - binary file, 255, 254 - tokenised 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])

18.2.3 Data block

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

18.2.4 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

18.2.5 Loaders

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

18.3 Super Turbo

18.3.1 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 distinguished: 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.

18.3.2 Header block

Offset	Description
0	Always 183
1	File type (1 - plain data, 3 - program, 4 - binary file, 255 - tokenised 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])

18.3.3 Data block

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

18.3.4 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

18.3.5 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 capable to display progress of loading using PMG.

18.4 Turbo 2000 - kilobyte blocks

18.4.1 Description

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

Four types of pulses are distinguished: 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.

18.4.2 Header block

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

18.4.3 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 zeroes. If the block is a partially filled block, there is data padded with zeroes 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]

18.4.4 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

18.4.5 Loaders

This turbo system is integrated into 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.

18.5 Turbo Tape

18.5.1 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 distinguished: 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.

18.5.2 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.

18.5.3 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	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]

18.5.4 Timing

Timing is compatible with Turbo 2000 and Super Turbo.

18.6 B-TAPE

18.6.1 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 distinguished: 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.

18.6.2 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.

18.6.3 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 zeroes if needed.
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1025]

18.6.4 Timing

Timing is compatible with Turbo 2000 and Super Turbo.

18.6.5 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.

19 Turbo systems from Poland

19.1 KSO Turbo 2000 and Turbo 2000F

19.1.1 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 distinguished: 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 joystick port (KSO Turbo 2000) or on DATA-IN pin of the SIO connector (Turbo 2000F).

19.1.2 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

19.1.3 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 zeroes 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.

19.1.4 Timing

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

19.2 Turbo Blizzard

19.2.1 Description

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

Three types of pulses are distinguished: 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. Block does not hold 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.

19.2.2 Header block

Offset	Description
0-75	File name
76	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[75]) mod 256
77	Spare byte, always 0

19.2.3 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 zeroes if needed.
1026	Always 0
1027	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[1026]) mod 256
1028	Spare byte, always 0

19.2.4 Timing

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

19.3 Turbo ROM

19.3.1 Description

Turbo system used in Poland, with limitations similar to Czechoslovak Turbo 2000 and Super Turbo Systems. This turbo system can hold Turbo ROM compatible binary files (those 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 distinguished: 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 two narrow pulses. Then data follow. Data blocks are separated by short gaps. After data block, there are four wide pulses and one stop pulse.

19.3.2 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 init address, 1 - No JSR to init address
37-39	Padding zeros
40	RTS opcode (96)

19.3.3 Data block

Offset	Description
0-?	Bytes of data

19.3.4 Timing

Pulse	Width
Wide	16/44100 s
Narrow	6/44100 s
Stop	48/44100 s