Fsk Encoder user guide

The Fsk Encoder project

October 2025

Contents

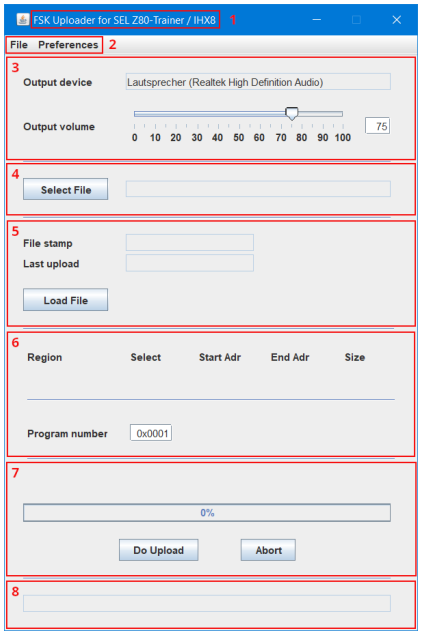
Introduction

The Fsk Encoder application is a usefull tool for SWw development in retro computing enviornment.
It's capable of converting binary code or data files into FSK encoded sound samples which can be played on the computers sound card. Together with an appropriate interconnection cable the sound output of the host computer can be connected to the sound input of a retro computer system to upload the data.

1 GUI

1.1 Main window layout

1. Titlebar, showing the selected target systen and inputformat.
2. Menu bar.
3. Output control showing the selected output device and offering a volume control.
4. Source File selector.
5. Source file info, showing the date and time of the last modification of the source file and the last time the file was loaded up to the target device. It also offers a button to reload the selected source file.
6. Target specific information area. It's contend depends on the selected target device.
7. Upload control section with a progress bar together with a start and abort button.
8. Status bar



1.2 Titlebar (1)

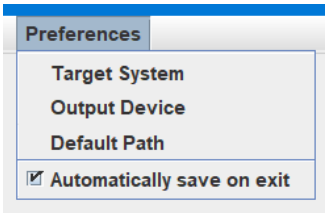
In the title bar the name of the currently selected target system is shown together with the currently selected input format, both seperated by a slash.

This information is obtained from the *Plugin.properties* file which is explained in detail in a later section.

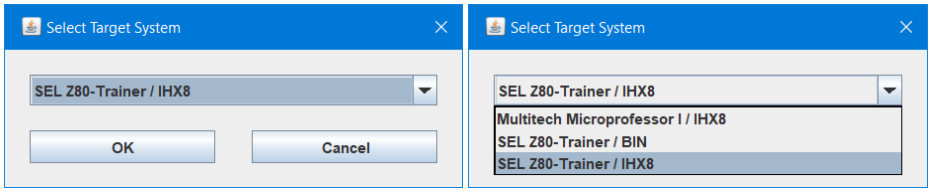
1.3 Main Menue (2)

The main menue offers two submenues:

1. **File**, which only has Exit as function.
2. **Preferences** with the functions
   (a) Target System,
   (b) Output Device,
   (c) Default Path and
   (d) Automatically save on exit.

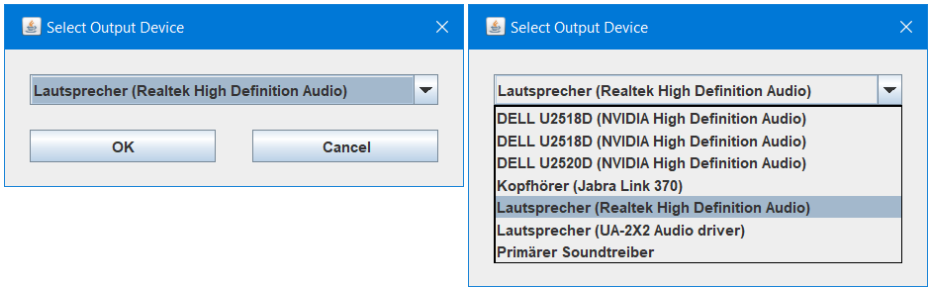


1.3.1 Target System selection



On selection a dialogue opens offering a drop down list which contains all currently available combinations of target system / input format. This list is set up from configuration file *Plugin.properties* which is explained in detail in a later section.

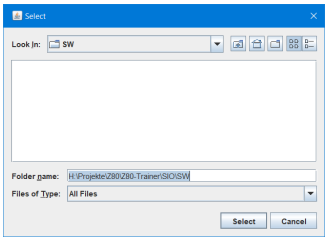
1.3.2 Output Device



On selection a dialogue opens offering a drop down list which contains all sound output devices of the host computer.

1.3.3 Default Path

Selecting this option opens the system's standard *File Open* dialog and offers a way to set the default working directory for the file search.

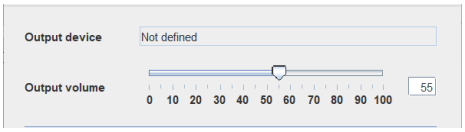


1.3.4 Automatically save on exit

If this option is checked, all preferences are written to the configuration file *FskEncoder.properties* when the programm exits.

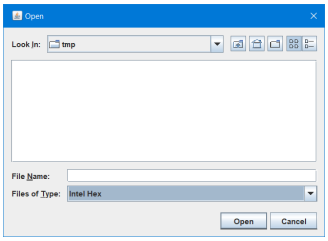
1.4 Output control (3)

This section displays the currently selected output device and provides a control for the output volume. The output volume can be adjusted by the slider or by entering the desired value in the textbox at the right side of the slider. Both controlls are kept in synch.



1.5 Source File selector (4)

By klicking the [**Select File**] button system's standard *File Open* dialogue opens and offers a way to select the desired source file for upload.
After selection of a file, it's path and name is displayed. It is automatically read and it's filestamp (date and time of last change) is displayed in the *Source File Info* pane.



1.6 Source file info (5)

This section provides information about the last time the source file was changed and the time of the last upload (if any).

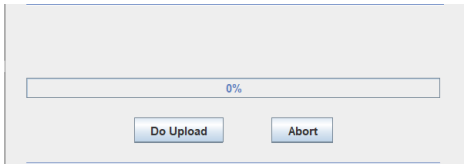
With the button [**Load File**] it also offers a way to reload the currently choosen source file if it has been modified.

Note: If a file has been uploaded to the target system, the *Last Upload* field changes but is left unchanged if the file is reloaded.

1.7 Target specific information (6)

The content of this pane depends on the selected target system and is discussed in the subsequent chapters for the target systems for the two targets which are built in (see section 4 on page 8).
For third-party targets and / or filereader, the explanation of this section should be found in the contributors documentation.

1.8 Upload control (7)



By clicking the button [**Do Upload**] the upload procedure starts. This means that the content of the file is converted into soundsampels (according to the protocoll that have to be used). Its progress is calculated based on the ratio between generated and played sound samples and then shown in the progress bar.

A running upload can be canceld by klicking the [**Abort**] button but abortion takes a little while to perform.

1.9 Statusbar (8)

Status and/or error messages are shown in the statusbar to give feedback.

2 Installation

The Fsk Encoder comes as *FskEncoder-all-dist.zip* file containing all the necessary folders and files, which must be unpacked in the chosen installation folder. It contains not only the application but also two target system and two input reader extensions, ready to use.

The resulting structure is:

```
<install_dir>
|
+-- bin
+-- cfg
+-- extensions
+-- lib
```

The *bin* folder contains the application itself and a .bat file for starting the application. The configuration files (see below) are stored in the *cfg* folder while required libraries are stored in the *lib* directory. The *extensions* folder is for the provider and reader modules, including the two provided and the 3pty contributions.

3 Configuration

The configuration of the application is held in the .bat file and two standard Java property files:

1. FskEncoder.properties
2. Plugin.properties

The .bat file is located in the *bin* directory while both .properties files are stored in the *cfg* subdirectory of the installation folder.

3.1 .bat file

This file contains the settings needed to configure the Java classpath. To make the application startable, the `INSTALL_PATH` variable must be set up correctly to the *bin* subfolder of the installation directory.

Also, if extensions are added, the `CLASS_PATH` variable must be updated to point to the added .jar files (extension and / or libraries).

3.2 FskEncoder.properties

In this file, the application saves it's curernt state (if enabled in the Properties menu). It's human readable but should not be edidet manually and it's O.K. if this file is missing because it's automatically created if requested by the *Save settings on exit*.

3.3 Plugin.properties

This file contains the available target systems and their internal configuration.

According to the Java property file standard it contains key-value pairs using an equal sign '=' as seperator.

The abstract syntax of this file is:

```
<unique_system_name>.name = <unique_name>
<unique_system_name>.provider = <unique_provider_class>
<unique_system_name>.inputFormat = <unique_provider_class>
```

`<unique_system_name>` is a unique name chosen freely, which is used to group the configuration items of one single target system.

The `<unique_system_name>.name` is the key for the visible `<unique_name>`.

`<unique_name>` is a unique and descriptive name which is used in the *Target System selection* and is displayed in the caption bar. Usually it is a combination of the provider and the assocciated input format.

The `<unique_system_name>.provider` property gives in it's value part the full qualified classpath of the provider implementation. The provider implementation is the translator (or compiler) wich transforms the data from the source file into the sound samples.

The `<unique_system_name>.inputFormat` property gives in it's value part the full qualified classpath of the file reader implementation.

For both, the provider and the reader implementation the following condition must be met:

> Each extensiom must be provided in it's own .jar file which must be named after the class name of the extension.

3.4 Example

This part of the `Plugin.properties` file (which is included in the distribution) shall visualize what was said before.

It contains the configuration of the target system 'SEL Z80 Trainer' in two different versions (which differ in the input file reader), '*SEL Z80-Trainer / IHX8*' and '*SEL Z80-Trainer / BIN*', distinguished by their `<unique_system_name>` *z80trainerIhx8* and *z80trainerBin*.

The related .jar files for the provider and reader are located in the extensions directory and are named *Z80TrainerExtension.jar* for the provider and

Ihx8ReaderExtension.jar respectively *BinReaderExtension.jar* for the readers.

```
# ------------------------------------------------------------
#
# SEL Z80trainer

z80trainerIhx8.name = SEL Z80-Trainer / IHX8
z80trainerIhx8.provider = target.z80trainer.Z80TrainerExtension
z80trainerIhx8.inputFormat = source.ihx.x8.Ihx8ReaderExtension

z80trainerBin.name = SEL Z80-Trainer / BIN
z80trainerBin.provider = target.z80trainer.Z80TrainerExtension
z80trainerBin.inputFormat = source.bin.BinReaderExtension
```

The related parts in the .bat file outlines as

```
set CLASS_PATH=.;./FskEncoder.jar
...
set CLASS_PATH=%CLASS_PATH%;../extensions/BinReaderExtension.jar;
set CLASS_PATH=%CLASS_PATH%;../extensions/Ihx8ReaderExtension.jar;
set CLASS_PATH=%CLASS_PATH%;../extensions/Z80TrainerExtension.jar;
```

The first `set CLASS_PATH` statement initializes the class path variable and points to the *bin* directory and the application .jar file.
The subsequent `set CLASS_PATH` statements append the specific extension .jar to the previously defined class path.

4 Extensions

Two target systems came together with two input file readers in the distribution package and are preconfigured ready to use.

The 'Target specific information' (see section 1.7 on page 5) area has different input fields for various control informations. Unless specified otherwise, all input is taken as hexadecimal digits and must be entered in C-style notation (with a leading 0x).

4.1 Multitech Microprofessor I

The MPF-1 system needs the '*File Name*' parameter for an upload which must be entered in the related text field.

In the old imes where programms are read from tape this parameter was neccesserry to verif that the correct datablock was loaded. With FskEncoder it

can be set up with an default value of 0x0001 to satisfie the target systems requirement.

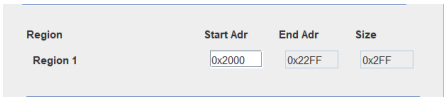
4.2 SEL Z80-Trainer

The Z80 Trainer system needs the '*Program Number*' parameter for an upload which must be entered in the related text field.

In the old imes where programms are read from tape this parameter was neccesserry to verif that the correct datablock was loaded. With FskEncoder it can be set up with an default value of 0x0001 to satisfie the target systems requirement.

4.3 Binary file reader

The upper part of the 'Target specific information' contains information about the binary source data.



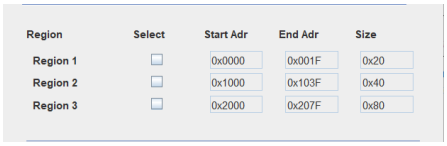
A binary source file can only consist of one single region.

Therefor, the panel contains only one data line labeld *Region 1.* This line contains a textfield '*Start Adr*' in which the start address of the data to be loaded must be entered.

Note: The start address is that address where the data get to be stored in the memory of the target device and depends on the type of the target system.

4.4 Intel-Hex file reader

The upper part of the 'Target specific information' contains information about the Intel-Hex source file.



A Intel-Hex (IHX) source file is record structured and can consist of one or more regions.

Because the IHX file contains address information, the start and end address are taken and calculated from the source file together with the block size.

To make a section an upload candidate, it's related '*Select*' checkbox must be checked. Multiple selections are treated as subsequent uploads and the upload is done automatically one by one in the order of the sections.

Because some target systems cannot handle an upload of multiple regions in one single task, they must be restarted for each region. This behaviour is supported by FskEncoder by displaying a Yes/No dialogue box for each selected region. The upload starts when the [**Yes**] button is clicked and can be skipped by clicking the [**No**] button.