Fsk Encoder Mpf1Extension user guide

© The Fsk Encoder project

October 2025

Contents

Introduction

The Fsk Encoder application is a usefull tool for SW 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.

The extension (*Mpf1Extension*) enables the FskEncoder application to upload data to the **Multitech Microprofessor-I** (Mpf-1) board. It does the conversion of the source data to sound samples according to the *protocoll* designed by Multitech.

1 Description

A so called *protocoll* defines the frequencies to be used and the encoding rules for binary zeros and ones. In this case - for Mpf-1 - it is quit complex. Please refer to Appendix A for detailed information on this topic.

2 GUI

The Mpf1Extension's specific GUI panel is shown in the *Target specific information area* of the FskEncoder main GUI and offers a way to enter the so called *File Name* of the upload candidate.

The *File Name* is a 4-digit hexadecimal number, used to identify a program on tape in the earlyer days. In FskUploader there is only one file as upload candidate and the *File Name* can easily be left at the default setting.

Note:
Please consult the Mpf-1 user manual for more information how to start an upload on the boards monitor system.

The layout of the panel is shown in the picture on the right..
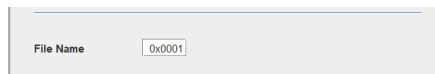
3 Installation

The Mpf1Extension comes as *Mpf1Extension-m.s.b.zip* file containing all the necessary folders and files, which must be unpacked in the installation folder of the FskEncoder application.

Note: The .zip file contains a version code consisting of

      m   the main line of the build,

      s   the stream of the build and

      b   the build number.

The main line of the extension **must** match the one of the application to function propperly.

The extension .zip file provides confiruration snippets which must be merged into the FskEncoder configuration to make the extension accessible (see next section).

4   Configuration

After unpacking the .zip file, three new files apear in the installation directories:

1. `./bin/Mpf1Extension.bat` containing the Java `CLASS_PATH` extension and must be merged into the `./bin/FskEncoder.bat` file,

2. `./cfg/Mpf1Extension.properties` containing a template for the configuration of the `./cfg/Plugin.properties` file.

3. And at least the extension file *Mpf1Extension.jar* in the `./extension` directory.

Both, the .bat and .properties files contain instructions on how to merge the extensions into the FskEncoder configuration.

A   Mpf-1 encoding protocol

```
Multitech Microprofessor MPF-I tape format


Bit format
--------------------------------------------------------------------------------


'0'    8 cycles 2000Hz   ( 8 * 0,5 ms = 4 ms )
     + 2 cycles 1000Hz   ( 2 * 1,0 ms = 2 ms )

     |                                 ^                  |
   _|⁻|_|⁻|_|⁻|_|⁻|_|⁻|_|⁻|_|⁻|_|⁻|_|⁻|_|⁻⁻|__|⁻⁻|__|⁻
     |0   1   2   3   4   5   6   7   0     1    |


'1'    4 cycles 2000Hz   ( 4 * 0,5 ms = 2 ms )
     + 4 cycles 1000Hz   ( 4 * 1,0 ms = 4 ms )

     |                 ^                                  |
   _|⁻|_|⁻|_|⁻|_|⁻|_|⁻⁻|__|⁻⁻|__|⁻⁻|__|⁻⁻|__|⁻
     |0   1   2   3   0     1     2     3    |


  1 bit equals 6ms



Envelope
--------------------------------------------------------------------------------

  1 start bit '0'
  8 data bits, lsb first (b0 to b7)
  1 stop bit '1'



Byte and Word format
--------------------------------------------------------------------------------

Word: Lo-byte,  Hi-byte
Byte: Lo-nible, Hi-nible

E.g.
```

```
The WORD  0x1234 0001.0010 : 0011.0100 (b7 ... b0 : b7 ... b0)
is send as   4321 0010.1100 : 0100.1000 (b0 ... b7 : b0 ... b7)



File format
--------------------------------------------------------------------------------

1.     4000 cycles 1000Hz Lead sync

2.     2 envlp    filename
3.     2 envlp    starting address
4.     2 envlp    ending address

5.     1 envlp    checksum of datablock start adr to end adr

6.     4000 cycles 2000Hz Mid sync

7.     n envlp    datablock

8.     4000 cycles 2000Hz Tail sync
```