8to16 release 1.3 sample converter documentation

COLLABORATORS			
	TITLE : 8to16 release 1.3 sample converter documentation		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 16, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

8to16 release 1.3 sample converter documentation

1.1 8to16 v1.3 user's guide

```
                 **  8 to 16  interpolating sample converter
  **    by IMMORTAL Systems & Jens Kirk

  **  This program is FREEWARE
  **  U can copy and use it free

  **  WARNING:

  **  Use it on your !OWN! risk !!!
  **  (..but U know it: no risK no fun!!! :)))

Choose one of topics......


                Introduction - what is it why use it

                Requirements

                Installation

                Features

                Usage

                Supported file formats

                History & future plans

                Credits,bugs and another important informations
                NOTE1:  This is guide for v1.3 of 8to16
NOTE2:  Please excuse my english, working on it....
NOTE3:  LONG LIVE THE CZECH AMIGA SOFTWARE !!!!
```

1.2 Introduction

```
                  **  Introduction

8to16 is 8-bit to 16-bit sample converter.  :((( only?
* Why use it? U can covert it in almost every program !!!

But......

It doesn't work such as many of converting algorithms, which only
take the 8-bit number and store it to 16-bit.That algorithm works,
but quality is still.....same!

But 8to16 works other way. It improves the quality by using
the
                Standart linear interpolation
                 methods.   :oo

You can also use another method called
                Kirk's interpolation.
                This method allows U to use your old 8-bit samples even on your
new soundcard or in 16-bit music editors with a little bit
improved quality! :))

The result is most audible in bass and mainly in bass and quiet samples.
If U don't believe it, I have one suitable sample given to this
package, so U can try it and get to believe.

Note:  LI is *not* the miracle algorithm to correct samples!
It can only correct certain type of bug in sample caused by sampling
it in low 8bit resolution! So please do *not* expect miracles from it!
```

1.3 Standart interpolation methods

```
  ** Standart LI methods

LI is a simple algorithm.
How does it work?

First, sample is rescaled from 8-bit(256 points resolution) to 16-bit
(65536 points resolution).It means, that between every two points
converted from 8-bit resolution are 256 unused points.
(65536/256=256)

So,it will be good to use them to make the sound more smoothy,isn't it?

And I decided to join all points by LINES.
LINES are computed in new 16-bit resolution and...
 that's it: Linear Interpolation!

.....what?
```

```
I'll give U an example:

part of 8-bit sample:

-
    ***
| 256      ***
| points        ***
-

rescaled sample

-
|   ***
|   |
|   |256 points
|65536    |
|points      ***
|     |
|     |256 points
|         ***
-

and then, interpolated sample

-
|   *
|    *
|65536      *
|points      *
|       *
|        *
|         *
-

so, sound is more smoothy then:

:--)))) woooowww!!!!

.....but

These are limitation. Interpolation CANNOT be used everywhere in
the sample without changing its rate!
(Try imagine it: U cannot intepolate 2 not equal points beside -
U would have to insert 3rd point between them first.....)

NOTE: BELIEVE OR NOT, 16-bit sample is twice long than 8-bit!!!!!
So if your 8-bit have 400K, 16-bit would have 800K!!!

So,it's all, please read other parts......
```

1.4 Kirk's linear interpolation method

```
  ** Kirk's linear interpolation

  The whole idea of '8to16' is to make old samples sound better on a
16 bit soundcard, than they would if you just scaled them to 16 bit.
  The heart of the system is the LI algorithm.

  The benefits of using a LI algorithm is that you get high frequency
noise filtered away digitally (almost like the analog filter in the
Amiga itself, except that you keep the the high-frequencies that belong
to the sample). The result is a 'softer' sample.

  The current LI algorithm used for the purpose is based on methods
used in many digital low-pass filters. It uses a approximation that is
as follows:

  A value between two points is calculated on the surrounding 6
original points:

t[x]=  1/32 * t[x-5] - 5/32 * t[x-3] + 5/8 * t[x-1]
     + 1/32 * t[x+5] - 5/32 * t[x+3] + 5/8 * t[x+1]

  The calculation is done in 2 levels so the resulting sample will have
4 times the samplerate. (The first time all values between the 8-bit
sample-points, the second time beteen these new points and the 8-bit
sample-points).
  To make it clearer, I have included an illustration of a downcut
8-bit sample and the resulting 16-bit.

  Everything is scaled by 32768/256 = 128. Why not 65536/256 = 256 (for
16bit/8bit)? Well ... As you can  see from an examination of the first
level, it can give a factor  (compared to a maximum normalized \ensuremath{\pm} ↩
   value)
\ensuremath{\pm}(1/32 * 2 + 5/32 * 2 + 5/8 * 2) = 52/32 = 13/8, second level is
restricted by first level and will not get above this value.

  To scale the sample the rest of the way you could multiply by 16 and
divide by 13 (~= *1.23) [remember it must be signed]. The LI algorithm
is made to be fast and is not surposed to handle things that could be
called options.
  A little precision is lost in second level (the value is rounded),
this means that all 16 bits is used. The implementation does not make
use of divisions or multiplications at all, resulting in very fast
translation. The data is not buffered.

NOTE: BELIEVE OR NOT! Since 16-bit sample is twice as long than 8-bit
and the samplerate 4 times as much, a 8-bit sample of the size 400K
becomes a 16-bit sample that requires 3200K = 3.125MB!!! This is why
it is so important that this routine gets buffered!

So,it's all, please read other parts......
```

1.5 Requirements

```
  * Requirements
```

```
 - OS and CPU:
=================

Although it was not tested , it may work *well* on
68000 CPU and OS 1.2 as well as on 68060 and OS 3.1.

It was tested on 68020 and 3.0.

 - MEMORY USAGE:
=================

It doesn’t use buffered loading now, so length of processed
sample still remains limited by free memory.

!!!!
You REQUIRE at least 3*<sample length> free mem for normal interpolation
mode and 5*<sample length> for DOUBLE RATE mode and 9*<sample length>
for Kirk’s interpolation mode.
!!!!

Example: If U want to process 200KB raw sample,U need at least:

  - 600 KB free mem for standart interpolation mode !!!!!
  - 1.0 MB free mem for DOUBLE RATE interpolation mode !!!!!!!
        - 1.8 MB free mem for KIRK’s INTERPOLATION mode !!!

 - GENERALY SPEAKING:
========================
Only what u need is ANY amiga and BIG piece of free memory......
```

1.6 Installation...

```
  * Installation

Installation of 8to16 is pretty simple.
First, extract the archive anywhere u want to have it.
It’s all ! :--)))

But then I recomment u to copy the main command file 8to16 to some
command directory for more availability.

[I have it in C:]
```

1.7 Features....

```
  * Features

  - Improves quality of samples using sample interpolation
  - Three interpolation modes: standart, double rate mode, Kirk’s special
  - Efficiency indicator
  - Imports 8-bitraw and 8SVX, exports Raw16bit and IFF-MAUD samples
  - Easy usage & installation
  - Low requirements
```

1.8 Usage of 8to16...

```
  * USAGE

Usage of 8to16 is very simple. U can use it from cli or better way
is to built it to some file manager - I have it built to my
DirWork and it works very well.

USAGE:    8to16 [-qdMk] <filename>

OPTIONS:

  -q : QUIET, doesn't open any window, just working :)
  -d : enables DOUBLE RATE MODE
       In double rate mode, sample rate is doubled, and sample
       octave is halved down.
       - efficiency is SIGNIFICALLY increased
       - new length of sample is 4*higher than source !!!
       - octave is halved
             When disabled, standart mode is set.In it:
             - efficiency is lower
             - new length is only twice higher than source
       - sample frequency ISN'T changed
  -M : use IFF-MAUD as the output fileformat
       (default is Raw16Bit)
  -k : use Kirk's interpolation
       In this mode, sample rate is multiplied by 4 and sample
       octave is  halved down 2 times. Remember, new length is
       8*longer !!!!


NOTE:   ALL options are case-sensitive!


EXAMPLES:

8to16 FukU.8svx   -will process FukU.8svx in normal mode
8to16 -q FukU.8svx  -same, but nothing will be displayed
8to16 -d FukU.8svx  -will process FukU.8svx in DOUBLE RATE mode
8to16 -qd FukU.8svx -same, but nothing will be displayed
8to16 -dM FukU.8svx     -double mode, uses IFF-MAUD as output fileformat
8to16 -kM FukU.8svx -kirk's interpolation, uses MAUD as export filetype.

After process, if no quiet mode , efficiency meter is displayed.
It signs the percentage of used interpolation.
It looks like this "[*****]".
One * means 10%.
In Kirk's interpolation, efficiency indicator doesn't work (his routine
works another way).

Example: [****] means that interpolation was used on 40% of sample.
Believe or not, result quality DEPENDS of efficiency!!!!
```

```
NOTE: In DOUBLE RATE MODE, efficiency is ALWAYS pretty high,
and it’s ALWAYS at least 50% !!!!!!!!!!!!

Now some notes :

Efficiency[%]
=============
>50   Very good, use it!!!!
50,40   Good, hear it and than use.
20,30   Average.
<20   Poor. Use 8-bit source rather.

Note1: In DOUBLE RATE MODE, 50% is the minimum.
So efficiency is ALWAYS [*****] at least !!!!!!! :))))

MAIN NOTE:
----------

Remember: After processing, SOURCE SAMPLE WILL BE DELETED AND
    WILL BE REPLACED BY NEW 16-BIT!!!!!!!!!!!!

So, don’t forget to backup source samples before experimenting....

Please, read SURE file-formats chapter!
```

1.9 Supported file formats...

```
  * File formats

In this version, only certain file types can be loaded
and saved.


Supported input file formats:
=============================

IFF-8SVX: 8to16 CAN load these files and it seems it handles
mono only them correctly.But it can load only old 8svx without
    delta compression.

RAW-8bit  Any other formats than 8SVX will not be recognized
    and will be loaded as 8-bit RAW files without header.

IMPORTANT:8to16 CANNOT handle ANY crunched files itself,so decrunch
    them first.

Some advices for power- and xpk- crunched files:

PP: Decrunch them first or use PPPatcher.

XPK:  Decrunch them first or mount some unpacking device
  (use xfd or xpk handler).

Supported output file formats:
==============================
There is only two now - Raw16Bit mono and IFF-MAUD mono.
(many programs support them, tested in Octamed Soundstudio)

Description of Raw16Bit:

There is an 8-bytes long header "Raw16Bit" and then follows
16-bit raw data in Motorola(tm) signed format.
```

1.10 History of 8to16

```
  * History and Future plans

HISTORY
=======

v1.0: Initial release (but not on Aminet)
  BUGS: - 8SVX were not handled properly
    - Error in command line scanning

v1.1: First release on aminet
  - new DOUBLE RATE MODE added [-d option]
  - all bugs from 1.0 fixed
  BUGS:   - Efficiency indicator bug in DOUBLE RATE MODE
      (it often said more than 180% :---)))

v1.2:   (30.1.1997)
  - new IFF-MAUD support added (thanx to FBY,that's it!)
  - Indicator bug from 1.1 fixed
  BUGS:   Not yet known

v1.3:   (17.4.1997)
  - new Kirk's interpolation algorithm added
  BUGS:   Not yet known

FUTURE PLANS
============

- more file formats,decrunch support
- buffered loading
- more options, more resolution-bug-correcting algorithms
- a xpk-based sublibrary for loading 8-bit as 16-bit interpolated
  in real time!
```

1.11 info

```
  * Info

8to16 was created on A1200HD in PhxAss.All the program is
written in it.
```

```
8to16 (c)1997 IMMORTAL Systems

FREEWARE VERSION

Thanx,suggests,money,JPGs,modules and BUG report to:
-
xvavra1@br.fjfi.cvut.cz
-
Please write "8to16 v x.x" to SUBJECT field before mailing.


Thanx to FBY 4 report and MAUD description.
Thanx to all others who did it something for u and I forgot them.
Thanx to Jens Kirk for his interpolation algorithm and for his support.

GREETINGS TO:
=============

DJ OKO,Alien,Dean Allen(Muzik),Lubooo,DNA and to all #amiga,#amigacs
IRC users.

NOTE: I’m looking for AIFF,WAVE,16SV samples description.

.
.
.
```