MOD 2 MID

AMIGA Modules to Standard MIDI files.

Version 1.4

© Paul Oliver 1993

Contents

1. Introduction

MOD2MID is a utility that translates the music contained in Modules (MODs) into standard MIDI files that can be played back by most sequencers on any multi-timbral MIDI synth.

The MOD format originated on the AMIGA computer where there are many varieties: Soundtracker, Noisetracker, Protracker and Startrekker being the most common. MOD2MID should be able to cope with 4-channel files from any of these provided they are not corrupt.

MOD2MID is a Freeware program which means that there is no registration fee and you are free to copy, give away or distribute this software provided you comply with the license at the end of this document.

2. What hardware and software is required?

MOD2MID runs under Microsoft Windows version 3.0 or 3.1. To hear the generated MIDI files you'll need a MIDI interface, a multi-timbral MIDI sound source and a sequencer that will accept standard type 1 MIDI files. You will also need some way of listening to the samples in a MOD in order to select instruments and pitch (a MOD editor is ideal). You'll also need a collection of modules to convert!

During the conversion process a number of temporary files are written to your hard disk (16 or 32 depending on how many samples there are in the module). These temporary files are erased automatically when the conversion is complete so as long as you have some free space on your hard disk of approximately the same size as the MOD being converted then there should be no problems.

3. How do MODs compare with MIDI files?

Modules are 4-channel music tracks designed to be played using a collection of sampled instruments contained within the module.
If the module only contains instruments and percussion (i.e. no effects or speech samples) then I thought it would be rather nice to be able to play the music back using my synths as the sound sources and thus achieve a much clearer sound.

MOD2MID enables this by generating a MIDI file from the song data contained within the MOD file. The MIDI file created has one track for tempo information and one track per sample in the module so in total there will be either 16 or 32 tracks. Several of these tracks are usually empty and can later be deleted.

The source MOD may also contain effects such as volume changes and pitch slides. Currently only a few effect commands are implemented as the information I have on them at present is rather sparse. I hope to implement more of the effects in future versions provided I can get detailed information on them.

4. Configuring MOD2MID

The first time you run MOD2MID you will be presented with a few options to give the program details about how you want the generated MIDI file to be created. These preferences are then stored in an INI file to be used next time MOD2MID is loaded. This configuration panel can also be reached at any other time via the menu bar on the main window.

The following information is required by the program:

• The drive and directory   This tells MOD2MID where the
   for output files.   generated MIDI files are to be placed.
                                                  Your working MIDI file directory is ideal.

- Whether track names should be created.

The sample names from the MOD file together with the sample number are used to name the tracks. This is very helpful for identifying the tracks particularly when the samples in the module are un-named.

- Whether Volume changes should occur in each track.

A volume change event (controller 7) at the start of each track is very useful for compensating for different instrument's loudnesses. You also need to supply a value in the range 0 to 127 that will be used as the default volume for each instrument.

- Whether Pan changes should occur in each track.

A pan change event (controller 10) can be created at the start of each track to place the instrument anywhere in the stereo field. The tracks can be set to sound in the centre of the stereo image, randomly scattered across the image or scattered with a normal distribution. I.e. more likely to be near the centre than at the left or right extremes (this sounds far more natural).

5. Converting a module

The converter simply generates a MIDI representation of the music in the module, it is left up to you the user to select appropriate patches for the instruments, transpose them as necessary, choose MIDI channels and adjust the volume and pan controls to get the best effect.

As an example of using the converter I have included the file "STORIESN.MOD" as it illustrates many of the common features of converted modules. Also included is "STORIES1.MID" which is my conversion voiced for the Roland MT32 family (including the CM32L, LAPC1 and SCC1). In this section I will describe how I generated this version.

5.1 Auditioning Sounds

In order to hear your MIDI version of the MOD tune you will need to select appropriate instruments for the different samples in the module. Currently the only way to do this is to load the module into a MOD editor or player that allows you to hear the individual samples. In future I would like to incorporate a sample player into the converter so if anyone out there has information on the WAV file format or a way of playing samples directly from Visual Basic then I'd really like to hear from you.

You not only need to select an instrument for each sample but also have to determine whether the sample is a chord, what key it is in and whether it needs to be transposed up or down one or more octaves.

For "STORIESN.MOD" I determined the following upon listening to the samples:

Sample Name:	Description: (MT32 patch numbers in brackets)
st-80:chris11	Synth with reverb. When played as middle C the note heard is the F above. (44)
st-05:akaimajor1	A warm major chord sample which is looped to produce a continuous tone. The chord sounds 1 octave down from the note played. (32)
st-05:akaiminor1	As above but this sample is of a minor chord.
st-04:Future2	A short, low electronic bell sound. This is used for bass notes in the song. (37)
st-03:DowSynth.TWI	An electronic "Doww" sound. This appears to sound an octave and eight semitones down from the note played. (30)
st-89:snare2	Standard snare drum sound.

sample slots 7 to 31 are unused.

As can be seen from these examples, not all instruments are sampled in the key of C so you may need to transpose them up or down several semitones when converting to MIDI. Many instruments are also sampled one or more octaves up or down due to the limited range of notes in modules. This is how low bass notes and high flute sounds are handled in modules.

Many modules will contain sounds that you won't have exact matches for on your synth(s). The music usually sounds fine though if you just pick the closest patch you synth has.

5.2 Using MOD2MID

The conversion itself is quite simple. Upon loading MOD2MID you are prompted to select a module to convert. The selected file is loaded and the main window is brought up which gives you information about the song. If you are happy with this then pressing the Make button generates the MIDI file.

The information you are given is:

•   The full song name which for the example is "stories never end2".

•   The names of all the samples.

•   The song length and number of patterns used. The example
        song is 42 patterns long made up from 36 different patterns.

•   Any effects used in the song. At present only a few of the effects
        are implemented in the converted song. See section 6 for details.

•   The proposed path and filename for the MIDI output file. This
        may be edited if necessary.

5.3 Editing the MIDI file

The converted MIDI file can now be loaded into your favorite sequencer where you need to set patches for the tracks, transpose them as necessary, expand notes to chords if required and remove empty tracks.

There are several things to be done before the song can be heard:

• 	 Erase all the redundant tracks that don't contain any notes or

tempo changes. In the example only the first 7 tracks are used
and the rest can be erased.

• 	 For each instrument track you need to assign patches, set MIDI

channels and transpose or expand to chords if necessary. When
creating STORIES1.MID, I set the following:

Name	MT32 Patch	MIDI	Transpose	Volume
chris11	44	2	+5 semitones	90
akaimajor1	32	3	-12 semitones	90
akaiminor1	32	3	-12 semitones	90
future2	37	4	-24 semitones	127
dowsynth	30	5	-20 semitones	100

The two chord samples need to be expanded. To do this make
two copies of each of them, transpose the copies +4 semitones
and +7 semitones for the major chords, +3 semitones and +7
semitones for the minor chords. Then merge the parts together
and erase surplus patch and control changes.

• 	 For percussive sounds you usually need to force all the notes into

a single note if like me your drum synth has one percussive
instrument on each note. For example the snare drum on the
MT32 is on note D2 so all the snare track has to be transposed to
this.

• 	 Adjust the volume and pan settings to achieve a well balanced

sound. The volume settings I used are in the above table but
they will depend on the loudness of the patches on your synths.

6. Effect implementation

The converter does not currently translate all the effects in a module as I do not have any detailed information on how they are specified or implemented.

The following chart lists all the effects in modules and what effect they currently have when converted to MIDI:

Effect	Comments
0. Note with Arpeggio	Arpeggio is sometimes used in modules to simulate chords. It tends to sound very noisy though and is currently not implemented.
1. Portamento Up 2. Portamento Down 3. Tone Portamento	These slide the pitch of a note up or down. I do not have any information on how fast they slide so they are not currently implemented. I hope to create MIDI pitch bends from them in a later version.
4. Vibrato	This works by varying the pitch of a note up and down. I do not know how the frequency or amplitude of the variation is defined so this also is not implemented.
5. Tone Portamento + Volume Slide	See Portamento or Volume Slide entries.
6. Vibrato + Volume Slide	See Vibrato or Volume Slide entries.
7. Tremolo	This varies the loudness of the sample in a similar way to Vibrato. Again not implemented.
9. Set Sample Offset	This adjusts the playing of the sample and is unnecessary in a MIDI file since samples are not used.

A. Volume Slide   Again I have no details on how this is specified so it is ignored.

B. Position Jump   This is generally used to loop the song at the end to create infinite length songs. Definitely not to be implemented!

C. Set Volume   The converter only acts upon volume changes at the start of a note and so cannot slide volumes. The volume is set in the note-on MIDI event and not varied by a controller since percussive sounds generally all occur on one MIDI channel and changing the volume of one would change all the others if a controller was used.

D. Pattern Break   Implemented.

E0. Set Filter   This sets a hardware filter on the Amiga and so is not acted upon.

E1. Fine Slide Up   )
E2. Fine Slide Down   ) As portamento.

E3. Glissando Control   Adjusts step size for effect 3.

E4. Set Vibrato Waveform   Selects waveform for Vibrato.

E5. Set Finetune   Adjusts sample playback - not used.

E6. Pattern Loop   Not implemented due to lack of information.

E7. Set Tremolo Waveform   Selects waveform for tremolo.

E9. Retrigger Note   Not implemented due to lack of information.

EA. Fine Volume Slide Up   )
EB. Fine Volume Slide Down   ) As Volume Slide.

EC. Cut Note) Not implemented due to lack of
ED. Note Delay) information.
EE. Pattern Delay)

EF. Invert Loop Plays sample backwards. Unused.

F. Set Speed Implemented.

In future versions I would very much like to allow pitch slides, vibrato and possibly volume slides and delays. If anyone has a detailed specification of these effects then I'd be very grateful if you could get in touch.

7. The legal bit

The software (MOD2MID.EXE) and this document are copyrighted by the author (Paul D Oliver). You are licensed to use this software, make copies and to distribute the software via any means provided:

• THIS PROGRAM IS NOT MODIFIED IN ANY WAY.

• THE PROGRAM AND DOCUMENTATION CARRY THEIR COPYRIGHT NOTICES.

• THE PROGRAM IS ALWAYS DISTRIBUTED WITH THIS DOCUMENTATION AND THE ACCOMPANYING FILES.

• NO FEE IS CHARGED, APART FROM DISK COPYING / LIBRARY CHARGES.

This software is provided "AS IS" without warranty of any kind. The copyright holder accepts no liability whatsoever for problems or damages resulting from the use or misuse of this software.

By owning / using this software you agree to all the above terms.

All trademarks are the property of their respective companies.

The software and documentation is
Copyright © Paul D Oliver 1993.

8. Version history

Version: Comments:
 1.4   First release of MOD2MID. (I hope I've eradicated most of
          the bugs by testing on my collection of 200+ MODs)

9. Contacting the author

If you have any comments about MOD2MID then I'd like to hear from
you. If you find any bugs then I would be grateful if you could let me
know and if possible send me the MOD that causes the bug to appear.
(sorry 3½" disks only!)

I would also like to hear from you if you have full details on effect
implementation / details of other music file formats that I could support /
suggestions for improvements.

I will try to reply to anyone who writes to me but those who send disks
of MODs and/or post-paid return envelopes will take preference!

Paul Oliver
6 Meadow Rise
Barton Under Needwood
Burton On Trent
Staffs, UK
DE13 8DT