

GMPlay

Christian Buchner

COLLABORATORS

	<i>TITLE :</i> GMPlay		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Christian Buchner	April 16, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GMPlay	1
1.1	Front Page	1
1.2	What is MIDI?	1
1.3	System Requirements	2
1.4	Installing the Program	3
1.5	The Command Line Interface	3
1.6	About the 14 Bit Driver	4
1.7	About the Samples	5
1.8	Creating your own Samples	5
1.9	Tone and Drum Mapping	7
1.10	Solving Problems	8
1.11	Program History	10
1.12	To Do List	10
1.13	Where to find MIDI files	11
1.14	The Authors	12
1.15	Donations welcome	12
1.16	Legal Stuff	13

Chapter 1

GMPlay

1.1 Front Page

--- GMPlay V1.2 ---

A General MIDI file player for the Commodore Amiga Computer.

no external hardware (synthesizer) needed!

Copyright © 1995/96 Alessandro de Luca and Christian Buchner.

Sonic Dreams Software

Documentation written by Christian Buchner

User Manual - Table of Contents

[What is MIDI? System Requirements](#)

[Installing the Program The Command Line Interface](#)

[About the 14 Bit Driver About the Samples](#)

[Creating your own Samples Tone and Drum Mapping](#)

[Solving Problems Program History](#)

[To Do List Where to find MIDI files](#)

[The Authors Donations welcome!](#)

IMPORTANT LEGAL STUFF

Please support development of this program by sending adequate gifts to the authors! [Any support welcome](#). Thanks.

1.2 What is MIDI?

What is MIDI?

MIDI is a standard for linking sound synthesizers and equipment. This standard allowed us to computerize and simplify making music. Today, nearly no chart hit is produced without the help of synthesizers and MIDI sequencers.

Computerized music is often stored as a Standard MIDI file (often abbreviated SMF). On MS-DOS systems, they carry the extension ".MID". These MIDI files only contain the information about the notes that shall be played by the tone generator at a given time. This is the reason why most MIDI files are relatively short-sized compared to sound modules and other music storage formats.

MIDI files do not contain any information about the instruments themselves. Instead, every instrument has a unique number. The General MIDI standard defines 127 instruments and a similar number of drums. The set of instrument is standardized among General MIDI synthesizers. The instruments and drums are usually played by expensive keyboards and synthesizers, but there is no job, an Amiga cannot do. So this is why we wrote GMPlay.

Most MIDI files sound better than sound modules, if they are played on a good synthesizer. The MIDI files are usually recorded by people who can play the keyboard quite well. So this is why MIDI files often sound almost perfectly. Another advantage of MIDI files over sound modules is the (virtually) unlimited number of voices. Of course this requires some horse power in your Amiga, which leads us to the system requirements...

1.3 System Requirements

System requirements

This program requires OS2.0 and an MC68020 processor or better. Fast RAM is recommended for faster operation. A hard drive will be of great use for storing the samples and your MIDI files.

An MC68030 at 50MHZ should be able to play most MIDI files at 28.867 kHz (the default rate). Owning an MC68040 or better, you can try to use 44.1kHz - but this only makes sense if you have a sample database at a really good quality.

The program requires the mathtrans.library and mathffp.library. These are part of the Amiga ROM and the workbench disks.

The program makes use of an installed XPK data compression package when compressed MIDI or sample files are encountered. An XPK installation is not required, however, if only uncompressed data is provided.

1.4 Installing the Program

Installing The Program

Please use the supplied Installer Script. It will do the job for you.

1.5 The Command Line Interface

The Command Line Interface

GMPlay can currently not be started from Workbench, sorry.

* CTRL-C aborts playing the current MIDI file.

* CTRL-D quits the pattern matching routine

(similar to aborting CLI batch files)

For a complete stop press CTRL-D-C.

The CLI/Shell command line template is

FILES/M,ALL/S,VOL=VOLUME/K/N,FREQ=FREQUENCY/K/N,BUF=BUFFERS/K/N,POLY=POLYPHONY/K/N
PRI=PRIORITY/K/N,VER=VERBOSE/S,FL=FILTER/S,NFL=NOFILTER/S

/A means required option

/K means keyword

/N means numeric variable

/S means switch

= defines an alternative name for the option (in most cases an abbreviation)

FILES This specifies the MIDI files to play. You may specify multiple MIDI files and AmigaDOS patterns. The MIDI files may be XPK packed. SHRI gives best results, especially at 100% efficiency.

ALL This switch will cause GMPlay to enter subdirectories recursively, looking for MIDI files.

VOLUME This defines the output volume. The volume is specified in percent of the standard hardware level. So 100 is the default volume. The permitted range is 0-500. The 14 bit driver is able to boost the output volume if values >100 are given. However, this may lead to overmodulated signals.

FREQUENCY The output frequency, default is 28867 Hz (maximum for the OCS display modes). If your computer is too slow, try lower values. Range is 4000 to 80000.

BUFFERS The size of the mix buffers in KiloSamples. Defaults to 16.

One sample uses 2 bytes. Three buffers of that size will be

allocated. Decreasing the buffer size will result in a more constant system load.

POLYPHONY Maximum polyphony. In other words the number of voices to play simultaneously. Range is 8 to 64. Default is 32.

PRIORITY Defines the task priority of the player. Try to use a priority greater than 0 if the player is often interrupted by heavy multitasking load.

VERBOSE Gives some extra information about the song.

FILTER Turns the audio filter on (currently ignored)

NOFILTER Turns the audio filter off (currently ignored)

1.6 About the 14 Bit Driver

About the 14 Bit Driver

My 14 bit driver allows the Amiga to play one Stereo channel in 14 bit dynamic resolution. This is almost as good as a CD player! Depending on the screen mode, you can play at 28khz, 56 or 78 khz. You will have to use a 30khz screen mode for using 44.1 kHz CD quality.

The 14 Bit Driver is part of a project called "CyberSound", which will probably be never completed (unfortunately). It is supposed to allow device independent audio output. A similar sound driver will probably be created for the Toccata audio board.

The 14 bit quality is created by using a cute trick. Two 8 bit resolution channels are added for gaining 14 bits. The second channel is played at 1/64 of the volume of the first channel and gives 6 additional bits. The low pass audio filter (coupled with the computer's power led) will be turned off in order to achieve a more brilliant sound.

The 14 Bit Driver can be calibrated for minimizing noise distortion. Use the 14Bit_Calibration program for this purpose. The calibration process will take approximately 15-30 minutes and requires a good HIFI equipment or headphones. By calibrating, you will measure out the non-linearities of your computer's D/A converters. My software then take respect of these settings and can improve its sound quality.

The 14 Bit Driver is still under development. You can find an earlier version on Aminet (including Source) in an archive called "14Bit_CDPlayer.lha".

The following programs make use of the calibrated sound output:

Play16 by Thomas Wenzel.

DeliTracker by Peter Kunath.

The AHI sound system (similar project to CyberSound).

1.7 About the Samples

About the Samples

Most of the tone and drum samples were created by Alessandro de Luca. The samples are quite noisy because they were sampled in 8 bit using a parallel port sound sampler. All the samples in the 16.7kHz drawers were created by him. His YAMAHA synthesizer unfortunately did not provide all instruments that are needed for General MIDI.

The instruments are in IFF 8SVX format, 8 bit MONO. Some of the samples are Fibonacci-Delta coded, which unfortunately adds some noise. If a sample cannot be found, the default sample is loaded. The default sample can be specified in the map files. If the default sample cannot be found, the program exits.

I crunched all the samples with XPK SQSH because this packs better than LHA and the resulting archive will be somewhat shorter. This is achieved by first applying a delta algorithm followed by a huffman coding (like LHA). You will have to decrunch the samples before you can load and edit them with other programs than GMPlay.

We are currently working several different approaches for gaining new sound resources. The next GMPlay releases will offer you new sample formats and hopefully one or several complete, high-quality sample banks.

- by sampling off whole instrument and drum sets from high quality synthesizers, using a Toccata audio board. We are trying to automatize sampling. One just has to enter a config file and the computer samples all the instruments that were requested.

- by implementing the SoundFont Format © EMU Systems. The SoundFont file format is currently used in the AWE32 sound card for PC's. Work is currently in progress to implement a loader for these instrument banks.

There are huge collections of interesting SoundFont banks in the Internet containing lots of HIFI instruments.

- by convertig or loading GUS (Gravis UltraSound) MIDI patch files.

1.8 Creating your own Samples

Creating your own Samples

If you want to create your own samples, I want go give you some recommendations that can make life more enjoyable:

- Use a good sampler (16 bit cards like Toccata are ideal) or good raw material, e.g. from special sampling CDs.
 - Try to use CD-quality sampling frequencies (44.1kHz). This makes the samples long, but actually they can be crunched better with XPK SQSH.
 - Sample instruments at their "native" pitch. e.g. use a C-2 for sampling a bass sound, and C-4 for sampling a whistle or something. This way you can avoid that the samples have to be transposed so much when they are re-played with GMPlay. This way they sound more naturally.
 - You are welcome to sample instruments at multiple pitches. Keep the raw material because later on I will add a new feature of supporting key range splitted samples.
 - Try to sample instruments without too much vibrato (turn it off if you have a synthesizer). Vibrato sounds awful when transposed up- or downwards. GMPlay might get a vibrato option later where the program does the job.
 - You currently have to use IFF 8 bit MONO samples. But please keep your un-edited raw material (preferably 16 bits) because you never know when you might need it again!
 - Try to use the full dynamic range when sampling. Do not sample at a too low volume because this reduces the dynamics and increases the noise levels. However, do not overmodulate when sampling (that means the sample must not hit the limits) - or you will get audible distortions.
 - Centralize and normalize the samples. Use adequate software for this step. Centralising means eliminating any DC levels by putting the sample right into the middle. Normalising means stretching the sample to full dynamic range.
 - Spend a great deal of time for looping the samples. Try to create clean loops without glitches and artefacts. Be careful: Most sampling programs discard the part after the loop part when saving IFF 8SVX samples. So make sure to keep a copy of your raw material (I cannot stress this often enough).
 - Do NOT use Fibonacci Delta coding when saving! Instead, you might consider crunching 8 bit IFF samples with XPK SQSH. This gives you 40-80% gain without any loss in quality, depending on the sample frequency.
 - When saving samples with programs like AudioMaster IV, be careful: AudioMaster likes to round the tuning frequency of the sample to the value of the nearest DMA frequency. So after saving a 44.1kHz sample,
-

it is necessary to patch the frequency in the VoiceHeader back to 44100 (\$AC44) with a file editor. Before saving samples with the program "Samplitude", do not forget to activate the TOCCATA 44.1 kHz audio output in the Parameters (p) or the frequency will be rounded up/down to the nearest DMA frequency as well. ;-(

Of course you are welcome to distribute your sample banks together with new map files. You can also send them to us. We will distribute the samples if they are better than ours.

1.9 Tone and Drum Mapping

Tone and Drum Mapping

The program distinguishes between tones and drums. Drums are located on track 10 of the General MIDI file. They are mapped to individual keys on the keyboard. All the other General MIDI instruments are referenced as Tones.

Instrument and Drum Mappings are currently defined in two ASCII config files. These are called 'GM:ToneMap' and 'GM:DrumMap'. GMPlay does currently not support multiple banks, but it will in the future.

The default tone sample is defined by the DEFAULTTONE statement. This sample will be used, if there is no tone specified for a specific MIDI program number or if the tone could not be loaded. The default tone must exist or the program will refuse to work.

The Template for the ToneMap is: NUM/N,NAME/A,KEY/K/N,TUNING/K/N
NUM defines the MIDI program number (1-128).

NAME specifies a filename relative to the 'GM:Tones' directory. This has to be a IFF 8SVX sample, optionally XPK packed.

KEY defines the MIDI base key of the sample. The default key is 60 which equals a C-3. The value is given in semitones, so 72 would be a C-4. It often makes sense to sample a bass instrument with at lower key value (e.g. 48).

TUNING allows you to fine-tune the sample. The value is specified in cents. A cent is a 1/100 of one semitone. So a value of 100 would transpose the sample up by one semitone.

Currently it is NOT possible to use different samples for different key ranges.

The default drum sample is defined by the DEFAULTDRUM statement. Equally to the default tone, this sample must exist or the program will refuse to

work.

The template for the DrumMap is: NUM/N,NAME/A,SPOS/K/N,LINK/K/N/M

NUM is the MIDI key number for the specific drum (0-127).

NAME is the filename relative to the 'GM:Drums' directory. This has to be the name of an IFF 8SVX sample, optionally XPK packed.

SPOS is the stereo position. 0 is far left. 127 is far right. 64 is in the center. Default is 64.

LINK is a list of links to other drum samples. Linked samples mutually exclude each other. This makes sense for hi-hats and cymbals.

1.10 Solving Problems

Solving problems

Problem: The sound output is too soft (not loud enough)!

The 14 bit output in fact is not quite as loud as the usual 4-channel Amiga sound. This also helps to prevent overflow situations, where the sound may get distorted.

Solution

You may turn up the volume of your HIFI set. Yeah, pump it up!

Or use the new VOLUME option. Please note that any volume setting above 100% may cause an overmodulated signal which can lead to an audible distortion in the audio signal.

Problem: Some MIDI files sound ***terribly*** ugly!

This may have several reasons:

- 1) You don't like the style of the music? ;-)
- 2) The author of the MIDI file was anything else but a good composer.
- 3) The MIDI file was created for a synthesizer which does NOT use the General MIDI instrument mappings. In this case GMPlay will of course use wrong instruments.
- 4) The samples of GMPlay do not sound too good (known problem).

Solution

1+2) In this case you will probably want to delete the file (and punish the composer?)

3) You may create a custom ToneMap or DrumMap for this special MIDI file. Currently, GMPlay only loads 'GM:ToneMap' and 'GM:DrumMap', sorry. Or edit the MIDI file with a MIDI sequencer/editor and correct the

assignments of the instruments.

4) Try to create better samples and/or instrument mappings.

Problem: The player refuses to load MIDI files and/or samples, although
----- they exist.

The MIDI files and samples may be crunched. Please check if you have installed the XPK compression package. GMPlay needs these libraries for loading crunched files.

Solution

Copy the supplied libraries to your LIBS: directory and try again.

GMPlay V1.1a contained a bug that made it refuse to load any instruments on OS3.0 equipped machines. It worked on OS3.1, though. Starting with version 1.2, this bug should be gone.

Problem: My MIDI editor cannot load the supplied MIDI files.

----- My sample editor cannot load the supplied sample files.

The MIDI files and samples have been XPK compressed for maximum storage efficiency.

Solution

Unpack these files first. Therefore you can use the "xPack" or "xDrop" programs that are available in the original XPK distribution on Aminet. Then your editors will be able to load these files.

Problem: The player complains about missing tones or drums.

Currently, the sample database is neither complete nor of a good quality. It does not yet contain all of the General MIDI instruments. We are working on this.

Solution

Wait for future GMPlay releases or try to create your own samples.

Problem: The sound stops and restarts in short intervals.

Your computer may be too slow for calculating audio data in real time. This can even happen on a MC68030/50MHZ when playing complex MIDI files. Perhaps your computer was only busy with too much multitasking. ;-)

Solution

Try to reduce the sampling frequency by using the FREQUENCY option. Try values of about 16000 or 10000 or lower. This will of course reduce the quality of the audio data, but at least you will get a continuous sound experience ;-)

If your problem is only caused by multitasking load, you can try to use a higher priority for the player. Use the ChangeTaskPri shell command or

the PRIORITY switch. Try e.g. a priority of 5. Now the player will have precedence over most other applications.

Problem: 14 bit driver and Gfx boards

The 14 bit driver might have some trouble with of Graphics Boards (CyberGfx etc). This is because the driver tries to read the minimal sampling rate for the current screen mode from the Monitor/DisplayInfo Data. This works for ECS and AGA modes, but some Gfx emulations do not give reasonable information on their screen modes. So the driver will be limited to a 28.867kHz replay rate, even if the DMA would allow more.

Solution

Currently none. I am working on this.

1.11 Program History

Program History

History since V0.9a

- New OS2.0 style command line interface
- ToneMap and DrumMap are now ASCII ConfigFiles
- Made sample mixing *a lot* faster
- XPK Support for MIDI files and samples
- Creates a 16 bit audio stream, supports external stream drivers
- Countless small and large changes in the code

History since V1.1a

- Random crash bug fixed (uninitialized stack variable in the MIDI loader)
- Executable is now optimized with the SAS-C global optimizer
- Fixed bug in command line parsing (BUFFER option was broken)
- Added AmigaDOS pattern matching for loading MIDI files.
- Added the ability to specify a volume from 0-500%
- Fixed a bug that made GMPlay fail on Kickstart 3.0 systems (the config scanner did not parse the mapfiles correctly)

1.12 To Do List

To Do List

In the future, this program will become *real* shareware which will probably force you to pay if you want to use it regularly. So enjoy the

program right now.

Currently it is still giftware, which means that you are encouraged to send us an adequate donation of any kind to support this program.

This is what I want to implement, but I cannot promise anything.

- Fix the progress indicator (sometimes the song ends before reaching 100%)
- Add an instrument decay and relative sample volumes
- Add multiple bank supports for drums and tones
- Replace the samples with a complete HIFI collection at 44.1kHz
- Add the possibility to load 16 bit samples (WAVE, MAUD, etc) as well.
- Add compatibility for SoundFont banks (see AWE32 sound cards for PCs)
- Write a loader or converter for GUS patch files
- GUI for the player and the instrument maps
- Workbench interface
- DeliTracker interface
- Allow saving of the created audio stream
- Toccatà and Maestro Soundcard drivers

Some extra utilities are in the works:

- A resampling utility for up/downsampling samples without losing quality or information, based on spectral analysis and resynthesizing.
- A utility for finding and eliminating equal midi files in a collection of many thousands.
- A utility for the Toccatà audio board that can do automatic sampling of a MIDI synthesizer's instruments.

1.13 Where to find MIDI files

Where to find MIDI files

The internet offers huge resources of MIDI files. Many people offer their favourite MIDI files on their home pages.

When searching for a specific group and title, you might consider using a search engine like AltaVista (<http://www.altavista.digital.com>).

Some MIDI WWW-Servers:

<http://www.midifarm.com>

Some MIDI FTP-Servers:

<ftp.cs.ruu.nl>

This list will be extended soon, but I do not have many addresses handy right now. You can EMail me the addresses of good MIDI servers. I will include them into this list next time.

1.14 The Authors

The Authors

Christian Buchner

A 22 year-old student of Electrical Engineering and Technical University of Munich. Special interests: Amiga, Internet, Computer Music, Digital Signal processing, Hardware drivers, etc...

Hey, guys! I really need a new CPU board. Mine is causing trouble all the time. Please help us developing this program (any donations welcome).

EMAIL: flowerp@eikon.e-technik.tu-muenchen.de

Snail: Christian Buchner

Steinbergstraße 2

82110 Germering

Germany

Tel.: +49-89/849868

Alessandro de Luca

Sorry, I don't know much of him, except that he lives in in the south of Italy (near Messina) and started to code this marvellous MIDI player. ;-)

EMAIL: hairy@alicudi.unime.it

hairy@asterix.unime.it

jafo@eden.it

Snail: Alessandro De Luca

Via degli Oleandri s/n

98035 Giardini Naxos (ME)

ITALY

1.15 Donations welcome

Donations welcome

Currently, GMPlay is GiftWare. What does that mean?

If you like the program, you are encouraged to send us a gift (also called a present - just to clarify the meaning). You can send the gifts to Christian Buchner or to Alessandro de Luca - or to both of us (just as you like).

We will take everything (as we are students) - but of course monetary donations are welcome best. You can send us nearly anything that will make us a joy. Bottles of wine, some specialities of your region, fancy

computer hardware, cool Amiga games, sex toys. But please no drugs! ;-)
Well, except alcohol - and perhaps some cigarettes for Alessandro. Do what you like!

In case you want to send us money: Please tell the receiver to whom of us you sent the money (only to Christian, only to Allesandro or to both of us). This is because I want to share any monetary donations with Alessandro. I don't want to keep it alone, because this would be unfair - we both invested about an equal amount of work in this program. So please let us know if you sent any money to both of us - or just to one of us.

IMPORTANT: GMPlay will become Shareware in the future.

Any money that you now send to us will be taken into account for a later Shareware release of GMPlay. So you will have to pay less (or nothing) if you have sent us any money before.

Want to send us anything? Click [here](#) for our addresses.

1.16 Legal Stuff

DISCLAIMER

WARNING: We are not responsible for any loss or damage caused by this program to your software, your hardware, your house, or your life, even if you use it in perfect respect of my directions.

YOU CAN USE THE PROGRAM AS IS, AT YOUR OWN RISK.

You can freely distribute this program for non-commercial purposes only, but you must not modify the archive contents. However you can replace instrument samples in your installation to fit your own taste.

COMMERCIAL DISTRIBUTION OR INCLUDING IN MAGAZINES OR COVER DISKS IS FORBIDDEN WITHOUT OUR WRITTEN PERMISSION.

Explicit permission to Fred Fish and Urban Dominik Müller to include this software on Fish disks, Aminet and Aminet CDs. Thanks for keeping Amiga alive, guys.