

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 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GMPlay	1
1.1	Front Page	1
1.2	Release Notes	2
1.3	Updating from V1.2	3
1.4	What is MIDI?	3
1.5	What does GMPlay do?	4
1.6	System Requirements	4
1.7	Installing the Program	5
1.8	The Command Line Interface	8
1.9	Setting Default Options	11
1.10	About the 14 Bit Driver	12
1.11	About the Samples	13
1.12	Creating your own Samples	14
1.13	Tone and Drum Mapping	15
1.14	Solving Problems	18
1.15	Known Bugs and deficiencies	21
1.16	Program History	22
1.17	To Do List	23
1.18	Where to find MIDI files	25
1.19	The Authors	26
1.20	Donations welcome	27
1.21	Legal Stuff	27
1.22	MIDI implementation chart	28

Chapter 1

GMPlay

1.1 Front Page

--- GMPlay V1.3 ---

A MIDI file player for the Commodore Amiga Computer.

plays General MIDI, Roland GS and Yamaha XG files.

no additional hardware (synthesizer) needed!

Copyright © 1995-97 Alessandro de Luca and Christian Buchner.

Sonic Dreams Software

Documentation written by Christian Buchner

! Release Notes !

Updating from V1.2

User Manual - Table of Contents

-> What is MIDI? -> What does GMPlay do?

-> System Requirements -> Installing the Program

-> The Command Line Interface -> Setting Default Options

-> About the 14 Bit Driver -> About the Samples

-> Creating your own Samples -> Tone and Drum Mapping

-> Solving Problems -> Known bugs and deficiencies

-> Program History -> To Do List

-> Where to find MIDI files -> The Authors

-> Donations welcome! -> IMPORTANT LEGAL STUFF

Appendix A:

-> MIDI implementaton chart

>>> NEW! Visit our Internet support page! The URL is: <<<

<http://www.amigaworld.com/support/gmplay/>

Update versions, sample kits and MIDI files can be obtained here!

Please support development of this program by sending adequate

gifts to the authors! **Any support welcome.** Thanks.

1.2 Release Notes

Release Notes

Welcome back!

Some people were actually thinking that I had stopped working on GMPlay. You were absolutely wrong, guys! In the last 6 weeks I have done nothing else but working on this fine program (just compare the binary file sizes of V1.2 and V1.3!)

I was not planning to release this (actually unfinished) version, but I am going to Croatia for a week's vacation. And just imagine me being killed in a car accident or by a mine without having released the work of the last 6 weeks! ;-)

Sorry that the ADSR envelopes and the new tone samples are not yet finished as promised. This 1.3 release is soon to be followed by an 1.4 release which comes with all these neat features. See what's planned for release 1.4 in the [To Do List](#)

Check out my new GMPlay support page (URL see front page) to see the progress of the program. It will be updated every three days or so.

What's new in this release? Multiple drum kits and excellent new drum samples. Lots of cosmetic stuff. Lots of internal changes. Yamaha XG support (concerning the drums). A brand new timer (700 lines of source code!). Bugfixes in the MIDI loader. Countless other changes. Check out the [program history](#) for a complete list.

Six weeks is a hell lot of work. Hey, I am a poor student without regular income. And I really need a better turbo board. So any gifts are appreciated! Thanks to all people who have already sent me some money. Special thanks to José from Spain for the bottle of wine (the other one will go to Alessandro). *No thanks* to the editors of "CU Amiga" who did not send me their Cover CD-Rom containing GMPlay 1.2 although they promised to do so.

Version 1.4 will in fact become shareware and will be annoyingly crippled. Just to let you know. Have fun with this program, as long as it doesn't cost a buck.

Just a note: If you are experienced with electronics and have \$150 bucks left, you should check out my waveblaster hack in Aminet ([hard/hack/Waveblaster.lha](#)). It describes how to connect an excellent PC wavetable sound card to your Amiga. Check it out! It sounds far better than GMPlay, although my goal is to reach a similar sound quality with GMPlay.

1.3 Updating from V1.2

Updating from V1.2

You really should do this with the supplied installer script. It will automatically detect an old GMPlay installation and update it.

If you insist on doing the upgrade manually, make sure you do the following:

- Back up your old Tones and Drums drawer as well as GM:ToneMap and GM:DrumMap if you want to keep them.
- Delete GM:Tables/#? (VERY IMPORTANT! Table format has changed.)
- Replace the GMPlay executable and documentation.
- Replace the 14 bit driver with the latest version.
- Install the Tones and Drums directories with the new samples.
- Install the new MapFiles directory.
- Install the supplied XPK library 4.13 and sub libs if you haven't already installed the latest XPK compression package (4.13) from Aminet.
- Copy the new example MIDI files to your MIDI-Directory if you wish.
- The GM: assignment can be replaced by the GMDIR environment variable which contains the name of your GMPlay directory. Copy this variable to ENVARC: for your convenience.

1.4 What is MIDI?

What is MIDI?

MIDI is an abbreviation for "Musical Instrument Digital Interface", a standard for linking sound synthesizers and equipment. This standard allowed mankind 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 detailed 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.

1.5 What does GMPlay do?

What does GMPlay do?

GMPlay is a program to load and play MIDI files with nothing but a plain Amiga. Unlike most other MIDI players for the Amiga, you do not require any additional hardware (MIDI interfaces, keyboards, MIDI sound cards etc).

This is why you can call this program a "Software Synthesizer".

The music is computed and simultaneously played with the Paula chip in 14 bit quality. In future versions you will have the possibilities to route audio output into sound files or to Amiga sound cards (e.g. Toccata, Maestro, Melody etc).

The program now also recognizes Yamaha-XG MIDI files. As I happen to own a Yamaha DB50XG sound card that I connected to my Amiga's serial port, I wanted to be able to play all my Yamaha XG files. These files offer some extras, like e.g. different instrument and drum kits and the ability to have more than one drum part.

The computation of music with a virtually unlimited number of voices is a difficult and CPU intensive job and this requires some horse power in your Amiga, which leads us to the system requirements...

1.6 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 28.603 kHz

(the default rate). Owing 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.

1.7 Installing the Program

Installing The Program

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

The GMPlay Directory

GMPlay and its data files reside in a directory (from now on referred to as the GMPlay directory). The environment variable "GMDIR" may specify this directory. If the "GMDIR" variable cannot be found, the program looks for an assign called "GM:". If this assign cannot be found, the program automatically looks in "PROGDIR:" which is the directory the main program resides in.

The installer script will automatically install a SetEnv command in your startup sequence that defines the GMPlay directory. Unlike earlier GMPlay versions, the "GM:" assign is no longer required.

This is the usual structure of the GMPlay directory after installation:

Tones (dir)

Drums (dir)

Mapfiles (dir)

SoundDrivers (dir)

Tables (dir)

MIDI-Files (dir)

14Bit_Calibration 14Bit_Calibration.info

GMPlay GMPlay-Eng.guide

GMPlay-Eng.guide.info MIDI-Files.info

Note that Tables directory and its contents will be created by GMPlay when run the first time. The SoundDrivers directory may optionally be installed in DEVS:

GMPlay can easily be used in conjunction with directory tools (Directory

Opus) and internet browsers.

Configuring Directory Opus for GMPlay

I only have Directory Opus 4.11 and will describe here how to configure it to play MIDI files with a mouse click. Isn't it really the best GUI for GMPlay? ;-)

I am using the German strings (localisation) for Dopus, so the actual button texts may differ from what I quote here.

Enter the configuration menu "file types". Klick on "New Type" and select "Standard MIDI File". Luckily this file type has already been defined and stored as a preset.

The actions/commands to define on this file type will probably be "Double Click", "Loop Play" and "Play". Perhaps you also want to define a User type which you later on assign to a Button.

Klick on "New Line" for the very first action (double click) and select "Amiga DOS". The command you define should look like similar to this:

```
GMPlay {F}
```

You can specify any option (frequency, etc) that you prefer. For a complete list of options see the section about the command line interface.

You may enable the "Output Window" and perhaps the "Start Asynchronously" flags.

A stack size of 10000 is adequate. Priority can be anything you like (may as well be defined in GMPlay's command line options). A pause value of 1 to 3 will leave the output window open for some seconds for you to read any error messages.

Now if you have defined the "double click" action, test it. If you like what you hear and see return to the configuration section and reselect the action "double click".

Choose "Copy" from the Intuition menu and select any other action or command you want to define (play, loop play, userx). Paste your settings to these actions/commands one by one using the Intuition menu.

That's it. Happy MIDI listening.

Configuring Web Browsers for GMPlay

Often you find links to .MID files directly on web pages. This document describes how to configure your web browser for playing the selected MIDI file when you click on such a link.

Each document that comes through the World Wide Web carries a so called "MIME type" which specifies the file type. MIDI files usually have the

MIME types:

"audio/midi" or "audio/x-midi"

Both MIME types are used for MIDI files and so you have to configure your web browser identically for both MIME types.

If a server does not send any MIME type, the web browser may also recognize the file type by checking the file extension. MIDI files carry the extension

".MID" or ".KAR"

.KAR stands for Karaoke MIDI files that also contain lyrics synchronized with the music (displaying lyrics is currently not yet supported by GMPlay, but the files will play well).

So this is what you enter in IBrowse (configuration for other web browsers is very similar):

Type : audio / midi <- two separate string gadgets

Extension: mid kar <- without the dots

Action : External Program

Viewer : GM:GMPlay <- maybe replace GM: with the full path

Arguments: <>CON:100/100/500/120/GMPlay %f V=400 F=20038 M DF B=2 PRI=21

Why the redirection to a CON window? If you used the standard IBrowse output window, breaking GMPlay with CTRL-C would lead to an immediate exit of the IBrowse program. The redirection avoids the problem. Adjust the window coordinates as you like.

GMPlay is supposed not to eat up all the computing power, so we use a trick: The options set a high volume and a relatively low output frequency. For better audio quality, the drums are filtered. The MONO option saves some CPU time. The program runs at an extremely high priority (higher than Intuition's input task). So the audio replay will never be interrupted except if GMPlay runs out of computing power. To prevent Intuition reacting slowly to user, we use a very small mix buffer of only 2048 samples.

Now you should have enough CPU power left to work well with IBrowse and surf the net while listening to "internetted" MIDI files. Lower the output frequency if you still have problems with the computing power.

If you wish, you may as well enable the VERBOSE switch to obtain more information about the current MIDI file. Contrary to this you might instead want to enable the QUIET option to stop GMPlay printing playtime progress information.

Unfortunately many servers offer their MIDI files only as zipped archives.

If you want to play these by mouse click, you should get an UnZip program from Aminet and write some batch files that automatically extract the zip archive and play all contained MIDI files.

1.8 The Command Line Interface

The Command Line Interface

GMPlay can currently not be started from Workbench, sorry.

* CTRL-C aborts loading or 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.

You can get a print of the command line template if you type "GMPlay ?" in the shell. Below you can see what the characters in the command line template mean:

/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)

The command line template has become very long since version 1.2.

Therefore I am listing and discussing all the options in grouped sections.

Program options

FILES/M

ALL/S

RND=RANDOM/S

PRI=PRIORITY/K/N

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.

RANDOM This will enable random play. GMPlay first scans all specified files using AmigaDOS pattern matching and builds a list of matching MIDI files. Then this list is played in random order, but each matching file is played only once.

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. This priority will only be used for playing, but not for loading MIDI files. If you do not

specify a priority, the priority of the shell's process will be used and left unchanged.

Audio options

O=OUTPUT/K

S=STREAM/K

V=VOLUME/K/N

M=MONO/S,S=STEREO/S

F=FREQUENCY/K/N

B=BUFFERS/K/N

OUTPUT Specified the output driver to use. It will be loaded from the SoundDrivers directory either in the GMPlay directory or in "DEVS:". Capital letters will be converted non-capitals and spaces will be removed. The default driver is "14 Bit".

STREAM The name of the audio stream. This may either be a filename or the name of an audio mode. Currently this option has no effect.

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. Large values may lead to slightly overmodulated signals. Note: This parameter has no effect on the software synthesizer (mixing routine). This volume change is done only by the output driver or hardware.

MONO Enables mono output. Gives you extra speed, but will disable stereo effects and sound panning.

STEREO The negative option of MONO. This is the default.

FREQUENCY The output frequency, default is 28603 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 (MONO) or six buffers (STEREO) of that size will be allocated. Decreasing the buffer size will result in a more constant system load, but will increase the chance of audio dropouts.

Synthesizer options

XG=XGMODE/S,NXG=NOXGMODE/S

T=TONES/K

D=DRUMS/K

P=POLYPHONY/K/N

VELMAP=VELOCITYMAPPING/K

VOLMAP=VOLUMEMAPPING/K

TF=TONEFILTER/S,NTF=NOTONEFILTER/S

DF=DRUMFILTER/S,NDF=NODRUMFILTER/S

XGMODE Forces GMPlay and player into Yamaha XG mode. This option is only useful if you know that a certain MIDI file was composed in XG mode, but is not recognized as such. This may happen when the composer forgot to put a "XG System On" SysEx message in the MIDI file. NOXGMODE is the negative option and the default. Note that NOXGMODE will not disable the detection of XG MIDI files.

TONES Specifies the Tonemap file to use. "Standard" is the default. This will be forced into "XG" if the loader detects a XG MIDI file or if you specified the XGMODE option. Tonemap files will be looked for in the "Mapfiles" directory.

DRUMS Same as the TONES option, but for drum maps.

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

This option has no effect on the CPU load, except if you specify a too low polyphony value. This will result in note dropouts.

VELMAP Specifies the mapping of MIDI velocities to output amplitude. You have the choice between L=LINEAR, Q=QUADRATIC, E=EXPONENTIAL. The MIDI specs say that MIDI velocities should have a logarithmic scale. So EXPONENTIAL is the right choice and the default. Earlier GMPlay versions (up to V1.2) used a linear scale.

VOLMAP Same as VELMAP, except that this is for MIDI volumes (channel and expression volumes). EXPONENTIAL is the default. Earlier GMPlay versions (up to V1.2) used a linear scale.

TONEFILTER This option enables a low pass filter that is used on tone samples if the output frequency (FREQUENCY option) is lower than the tones' sample frequency. This option is recommended if you want to use low output frequencies (below

20kHz). This option will reduce aliasing (noise) effects, but at the cost of a longer sample loading time. The default is off because of the long loading times.

DRUMFILTER Same as the **TONEFILTER** option, but for the drums. Filtering really makes more sense for the drums, because these often contain high frequencies (think of hi-hats or cymbals). If you are not contented with the quality of the drums (when using low output frequencies), try enabling this option. Filtering the drums is faster than filtering tones because drums samples are much shorter. I notice an improvement in quality even at 28 kHz output frequency.

Miscellaneous flags

V=VERBOSE/S,NV=NOVERBOSE/S

Q=QUIET/S,NQ=NOQUIET/S

C=CHECK/S,NC=NOCHECK/S

VERBOSE Gives detailed information about the MIDI file, the audio and synthesizer options and the memory usage. The Default is **NOVERBOSE**.

QUIET Disables the output of playtime, measure and percentage while playing. Default is **NOQUIET**.

CHECK Enables checking of tone and drum mapfiles. Every non existing directory or file is reported. This is useful if you want to know if all your tones and drums are present. The default is **NOCHECK** because of the longer loading time.

1.9 Setting Default Options

Setting Default Options

Starting with GMPlay version 1.3, default options can be specified. This allows you to define your personal preferences. You may specify global defaults, but also file-specific ("local") defaults. These can be different for each MIDI file. Any defaults can be overridden by specifying options in the command line.

Global Defaults

Global defaults can be defined with the "GMOPTS" environment variable. For identification, the first token of this variable must be "GMPLAY". All

options of the CLI command template may be used, except FILES and ALL.

Note: All examples below are shell commands.

Example: SetEnv GMOPTS "GMPLAY VERBOSE FREQUENCY=20000 MONO"

will enable the VERBOSE option, set the frequency to 20kHz and enable the MONO mode.

You can also create or edit this environment variable with a text editor.

The filename is "ENV:GMOPTS". The file may contain linefeeds for better readability.

To make these settings permanent (survive resets), copy this variable to "ENVARC:"

Example: Copy ENV:GMOPTS ENVARC:

Local Defaults

Local default values can be stored in the file comment of each MIDI file.

The maximum length of this file comment is 80 characters, so use the abbreviated options if you are running out of space.

Example: Filenote mysong_xg.mid "GMPLAY XGMODE"

This statement forces GMPlay to recognize mysong_xg.mid as a Yamaha-XG MIDI file. Useful if the composer forgot to include a "XG System On" SysEx message.

You can also edit this file comment with a directory utility (e.g.

Directory Opus).

IMPORTANT: Do not forget to specify "GMPLAY" as the first token.

Otherwise your options will be ignored. This was necessary because some people (like me) prefer to use the file comments for other purposes.

1.10 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 takes 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 using the 14 Bit notepayer.

The AHI sound system (similar project to CyberSound).

mpega v2.6 MPegAudio player (also supports AHI)

1.11 About the Samples

About the Samples

Starting with GMPlay version 1.3, the drum samples have been taken from my Yamaha DB50XG sound card using a self-made 8 bit parallel port digitizer. They are sampled in 44.1 kHz and stored as IFF 8SVX files. I am using a special self-written automated sampling program for this. It samples each sample at a maximum amplitude but remembers its original volume. This guarantees the best possible sound quality that is possible with 8 bit samples. Some of the original Yamaha samples are still missing because they use an ADSR envelope (e.g. snare rolls, samba whistles, most of the sound effects).

I am intending to grab the tone samples from the DB50XG as well. But this is more complicated. I want to extend the IFF 8SVX file format for supporting ADSR envelopes and "dynamic loops".

Until this project is finished, I provide the samples created by Christian Sauer (except the GrandPiano and the Overdrive Guitar). These samples can also be found in Aminet.

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.

In the future we may also support different sample formats, e.g.

- 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.

1.12 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 natural.

- 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 (preferrably 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.

1.13 Tone and Drum Mapping

Tone and Drum Mapping

The program distinguishes between tones and drums. Drums are usually 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 defined in ASCII config files. These config files have to be located in the "MapFiles" subdirectory in the GMPlay directory. The naming scheme is "xxx_Drums" and "xxx_Tones" where xxx stands for the name of the instrument kit. The different kits can be

accessed with the "TONES" and "DRUMS" statement. GMPlay uses the "Standard" kit per default, except an XG MIDI is encountered. In this case the "XG" kits are used. Comment lines must begin with a ";".

Tone Map Files

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.

With the DIRECTORY statement you can change the directory in which all subsequent samples are looked for. The default directory is "Tones", relative to the GMPlay directory. You can give more than one DIRECTORY statement in a map file. Any directory name you give here will be looked relative to the GMPlay directory, except if you specify a fully classified path (e.g. "Work:Samples").

The Template for the instrument definitions is as follows:

NUM/N,NAME/A,KEY/K/N,VOL/K/N,TUNING/K/N

NUM defines the MIDI program number (1-128).

NAME specifies a filename relative to the current tone directory (DIRECTORY argument). This has to be a IFF 8SVX sample, optionally Fibonacci Delta and/or 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 (12 semitones are one octave), so 72 would be a C-4. It often makes sense to sample a bass instrument at lower key value (e.g. 48).

VOL defines the relative sample volume. This volume is specified in percent. The range is from 0 to 255. I do not recommend to use values much larger than 100 because this could overmodulate your audio signal.

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.

Drum Map Files

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. This sample will be selected from the default drum kit (No. 1).

With the DIRECTORY statement you can change the directory in which all

subsequent samples are looked for. The default directory is "Drums", relative to the GMPlay directory. The same rules as in the tone map files apply here.

Drum kits are defined with the DRUMKIT argument. Its template is as follows: DRUMKIT/K/N/A,NAME/A,BANK/K/N,LSB/K/N.

DRUMKIT is the number of the drum kit (1 to 128). This corresponds to the MIDI program number. Make sure that the first drum kit you specify has the number 1.

NAME is the ASCII name of the drum kit.

BANK is the MSB bank number of this drum kit (MIDI controller #0). If no bank is specified here, any bank number selects this drum kit. Valid range is 0 to 127.

LSB defines defines the LSB of the bank number (MIDI controller #32). If this is not specified here, any LSB number is selects this drum kit. Valid range is 0 to 127.

Unspecified keys (drums) in any drum kit will automatically fall back to the corresponding key (drum) of the default drum kit. This allows you to obtain full-blown drum kits by redefining only the crucial keys (drums) in your "personalized" drum kits. Otherwise you had to re-type all that munbo-jumbo, even if it were identical to the default kit. If you don't understand this, take the provided map files as an example.

The template for the drum definitions is as follows:

NUM/N,NAME/A,SPOS/K/N,VOL/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 Fibonacci Delta and/or XPK packed.

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

VOL defines the relative sample volume. This volume is specified in percent. The range is from 0 to 255. I do not recommend to use values much larger than 100 because this could overmodulate your audio signal.

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

1.14 Solving Problems

Solving problems

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 ;-)

You can also activate the MONO mode. This gains some speed but will deny you the experience of panoramic stereo sound.

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: At some very loud parts of a MIDI file I hear a violent "crack"!

The audio signal's amplitude got too high and produced an "overflow".

That means the 16 bit sample value was wrapped around from negative to positive sign. GMPlay does not check for this situation because of

speed reasons. Unfortunately this gives a loud cracking sound.

This is a very rare situation. GMPlay usually mixes at a quite low volume to prevent these situations.

Solution

Currently no easy solution exists.

Edit all the VOL statements in the key and drum maps and scale them down linearly (have fun with this ;-))

Or edit the volumes or velocities in the MIDI file with a MIDI sequencer.

Note: The VOLUME option has no positive effect here because it is just the volume of the output driver. The problem however occurs while mixing.

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

GMPlay mixes at a low volume to prevent overflow situations (see above).

In addition the two-channel 14 bit output in fact is not quite as loud as

the usual 4-channel Amiga sound.

Solution

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

Or use the VOLUME option. You can specify volumes of up to 500% here.

The 16 bit output sample will be stretched using tables, so no cracking overflow can happen here!

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 yet 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.

This is easy by using the TONES and DRUMS options as a local default.

(see the chapter about default values).

Alternatively 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: Tones are used instead of drums! Sounds terribly!

The MIDI file you have was written for a synthesizer that uses a different MIDI channel than channel #10 for its drums. GMPlay assumes that channel #10 is the drum part.

Yamaha-XG files also have the ability to assign other channels than #10 to drum parts.

Solution

If you are lucky, the MIDI file is a Yamaha-XG file that was not recognized as such. Try using the XGMODE option.

If you are unlucky, you will have to live with this. You may load the file into a MIDI editor and manually fix the problem there.

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.

The tone and drum kits provided with GMPlay release 1.3 should be complete. Use the CHECK option to check your tone and drum map files.

Perhaps the directory or sample names are wrong.

Solution

Correct the directory or sample names.

Create the missing samples yourself.

Problem: 44.1khz replay doesn't work!

44.1khz replay requires you to have an ECS or AGA chipset. The computer must be in "high scan" DMA mode.

The 14 bit driver checks for the active screen mode. If this is e.g a PAL or NTSC mode, it knows that the maximum DMA rate is 28.603 kHz. If the driver sees a "HiScan" screenmode, it knows that it can play up to 56kHz.

If the driver sees a Gfx board screen mode, it cannot decide what maximum frequency is allowed. It uses the internal default of 28.603 kHz.

Solution

You have to select a HiScan or MultiScan (31khz) screenmode to increase the DMA speed. WARNING: do not do this with a 1084 monitor or it will blow up!

Specify e.g. 44100 Hz frequency and open a Productivity screen right after or while starting GMPlay. If the 14 Bit driver does not reset the output rate to 28.603 kHz you made it!

If you have a Gfx board, you can now switch back to your graphic board screen. The 14bit.driver now remembers that the DMA has been set to HiScan mode until it is flushed out of memory (Avail flush).

But do not open a PAL screen or such in the meantime as this would reset your DMA mode to "slow scan" and limit the output frequency to 28.603 kHz.

Problem: 14 bit driver doesn't work at all with my Gfx board!

The 14 bit driver has been fixed for use with CyberGraphics and the Picasso-II emulation. There may or may not be problems with Picasso 96 however, according to the Picasso mailing list (PML).

Solution

Send me a bug report and describe your system hard- and software. I will see what I can do about it.

1.15 Known Bugs and deficiencies

Known Bugs and deficiencies

Known Bugs

* no real bugs known at the moment.

Unfinished stuff

The program is in an unfinished state:

* The program generates and allocates more tables than are actually needed at the current state of development. These tables were supposed to be used with the ADSR envelopes. The bad news is that these envelopes are not yet functional.

Other Deficiencies

Well basically everything you find on the **To Do List** could be seen as a deficiency. But this page is about things that are supposed to work, but refuse to do so.

And here's the good news: Everything seems to work fine. But I cannot tell for sure cause no one except me ever tested this program. ;-)

1.16 Program History

Program History

What's new in Release V1.3 ?

-
- Having more than one tone and drum set is possible now. (TONES, DRUMS options). Tone and Drum map files now reside in <GMDIR>/MapFiles
 - Added optional MONO mode for faster mixing and replay. 14bit.driver also supports faster mono replay.
 - Fixed 14bit.driver to better accept graphic boards. You must activate a Productivity or any other HiScan screen for about two seconds when starting GMPlay so that it can be recognized by the driver.
 - Added GMDIR environment variable. This specifies the player's base directory. If this variable does not exist, the default will be "GM:". If "GM:" does not exist, the program will fall back to PROGDIR:
 - A DIRECTORY option can now be given in the tone and drum map files to change the directory and group the samples together. This makes mapfiles shorter.
 - Improved output of playtime, measure and percentage, synchronized to the audio stream.
 - Gained some speed in the mixing routine.
 - Improved the loading routine for MIDI files. It should never crash and checks for corrupted data and exits immediately on every intolerable error.
 - The sounddrivers will now also be looked for in DEVS:SoundDrivers if not found in <GMDir>/SoundDrivers.
 - Tones and Drums can have relative volumes now. This allows to sample each instrument at full 8 bit level and obtain maximum signal-to-noise ratio.
 - Multiple drumkits have been implemented (DRUMKIT command in the mapfiles).
 - An excellent drum kit sampled from my Yamaha DB50XG board is provided. One kit for GM/GS mode, another kit for Yamaha-XG mode. Some samples however are still missing (would require envelopes to work)
 - The program now recognizes Yamaha XG MIDI files and enables special features of the XG MIDI standard (any channel can be changed into a drum channel etc.)
 - Global defaults can now be specified in an environment variable
 - Local defaults can be specified in the file comment of any MIDI file.
 - CTRL-C is now checked for during every time consuming action (loading MIDI
-

file, mapfiles and samples).

- Implemented selection of linear, quadratic or exponential mapping of MIDI volumes and velocities.
 - An optional lowpass filter improves audio quality when using low output sampling rates. Slows down loading the samples however.
 - The speed control and timing of the MIDI files is now much more exact than before.
 - GMPlay now informs you about what it is currently doing/loading (status reports).
 - Fixed problems with the Sustain Pedal (hanging notes) and added some MIDI commands.
 - Added a random play option that will play all songs specified as filename or DOS patterns in random order.
 - Implemented memory pooling that works under both Kick2.0 and Kick3.x. Reduces memory fragmentation and speeds up the memory allocation process.
- What was new in V1.2 ?

-
- 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)
- What changed first when I took over the project ?

-
- 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

1.17 To Do List

To Do List

As I said, this release is unfinished. I was intending to add many more features. According to my plans, release 1.4 should have been finished in

the end of April'97. Now it's end of May. Oops? I ran too much into details and neglected the really important things. Well, I'm a perfectionist ;-)

This is the feature list of the 1.4 version (June or July'97)

-
- an extension in the concept of the sound driver modules to allow the output of named audio streams (AIFF file name or AHI mode)
 - an AHI driver and an AIFF driver
 - "Dynamic Loops" in the samples. These are looped samples that change over time. An extension to the IFF 8SVX standard.
 - ADSR envelopes for the samples. Attack, Decay, Sustain, Release. No more chopped off instruments!
 - Karaoke MIDI support (displaying the song's text and syllables synchronized with the MIDI replay. Should be easy because the timer routines are already finished (playtime, measure and percentage stuff).
 - A German (and a french?) documentation.
 - Perhaps a special MC68000 version. This is a bit difficult because I am using some MC68020 specific instruction in the assembly routines that are difficult to replace.

Intended features for very future versions

-
- realtime effects (fast 68040 or better required, PPC?)
- Reverb, Chorus, perhaps a basic emulation of some XG specific effects
- 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
 - Localisation of the program
 - Workbench interface
 - DeliTracker interface

Shareware Plans

As this release is not yet fulfilling my expectations, I do not yet want to take a shareware fee for it. But 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.

1.18 Where to find MIDI files

Where to find MIDI files

That's easy. Mailbox systems, CD-Rom collections, PD disks etc offer huge resources of MIDI files. The only unfortunate thing about this is that most of these files follow the MS-DOS 8+3 naming scheme which often makes it difficult to guess their real names.

There are a lot of excellent MIDI files. Sorting out the trash is often very time consuming job. It's best to look for well-sorted and high quality sources of MIDI files.

Internet Support Site

If you have access to the Internet, your a regular look should go to our GMPlay support site at "<http://www.amigaworld.com/support/gmplay/>". Update versions, sample kits and MIDI files can be obtained here!

Internet MIDI Resources

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

Some links to MIDI resources. From there you can follow the countless links and the world of MIDI is open to you

URL Title

<http://www.midifarm.com> | The MidiFarm

<http://www.midifarm.com/midifarm/free.shtml> | MIDIFarm: Free files

<http://homepage.interaccess.com/~midilink/> | MIDI Link...

<http://www.midiweb.com> | MidiWeb

<http://www.aitech.ac.jp/~ckelly/SMF.html> | Standard MIDI Files on the Net

<http://www.prs.net/midi.html> | The Classical MIDI Archives

Yamaha-XG Internet MIDI Resources

Have a look at www.yamaha.com or the local YAMAHA web server in your country. They offer (PC) software and MIDI files as well as technical data and valuable tips.

There are a lot of internet sites that offer song collections for the Yamaha DB50XG. The best site is probably "The XG Home" at "<http://www.ids.net/~marshall/xg.htm>"

The link "The Songs" at the bottom of the page will take you to the

collection of midi files. Links to the composers' pages are also provided.

The songs on this server are especially written for the Yamaha DB50XG sound card and may sound very different on other MIDI equipment.

There are more web sites that offer XG midi files. Another excellent German site was created by Michael Banz. His site offers more than 500 XG midi files (even more that can be found on Marshall's XG Home page). Be sure to check out this URL:

"<http://www.castrop-rauxel.netsurf.de/~michael.banz/>"

A lot of cover version of original chart hits and old C-64 songs in XG MIDI format can be found on "<http://www.cs.vu.nl/~arthur/xg.html>"

1.19 The Authors

The Authors

Christian Buchner

A 23 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: hairly@alicudi.unime.it

hairly@asterix.unime.it

jafo@eden.it

Snail: Alessandro De Luca

Via degli Oleandri s/n

98035 Giardini Naxos (ME)

ITALY

1.20 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 Allessandro 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.

1.21 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 INCLUSION IN MAGAZINES OR COVER DISKS IS FORBIDDEN WITHOUT OUR WRITTEN PERMISSION.

Especially the British "CU Amiga" magazine is *not* allowed to put this 1.3 release of GMPlay on their coverdisk or CD-ROM until I receive my promised CD-Rom with the 1.2 release and the collection of MIDI files.

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.

1.22 MIDI implementation chart

Appendix A:

MIDI implementation chart

Almost every vendor of MIDI products offers a MIDI implementation chart that clearly states what MIDI commands are supported by their product.

GMPlay is a MIDI product and I guess that some of you really want to know what MIDI commands GMPlay supports.

This appendix is very technical and you should have a profound knowledge of the MIDI standard to really understand this.

MIDI commands Hexcode Comments

n Channel number (0-f)

Key Off 8n velocity is ignored

Key On 9n fully supported, user-defineable scale

keyrange 0-127, velocity range 1-127

velocity 0 causes a Key Off

Poly Pressure an ignored

Control Change bn supported, see list of supp. controllers

Program Change cn supported, selects active tone or drum

Channel Pressure dn ignored

Pitch Bender en fully supported

Supported Controllers No. # Comments

Bank Select 0 supported, currently only for drum parts

Modulation 1 received, but not yet implemented

Data Entry MSB 6 supported, see list of available RPNs

MainVolume 7 supported, user defineable scale

Pan 10 supported (not if MONO mode enabled)

Expression 11 supported, user defineable scale

Bank Select LSB 32 supported, currently only for drum parts

Data Entry LSB 38 supported, see list of available RPNs

Hold1 64 received, but not yet implemented

Sostenuto 66 supported

RPN LSB 100 supported, see list of available RPN's

RPN MSB 101 supported (->Registered Parameters)

Reset All Controllers 121 supported

All Notes Off 123 supported

Registered Parameters MSB LSB Comments

Pitch Bend Sensitivity 00H 00H supported (even more than +/-12 semitones)

MSB defines semitones, LSB defines cents

Recognized SysEx commands

XG System On : enables Yamaha-XG mode

Future implementation

really depends on your needs. just ask me for implementation.

Possible future support for:

Universal Realtime Messages: Master Volume, Master Tuning

Data Increment/Decrement for RPN (Controller No. 96, 97) ?