

Anonymous

Gaelan Griffin

Copyright © Copyright1998/1999 by Gaelan Griffin

COLLABORATORS

	<i>TITLE :</i> Anonymous		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Gaelan Griffin	April 14, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Anonymous	1
1.1	Table of Contents	1
1.2	What's an SPC700 anyway?	2
1.3	Cool legal routine, use it in your own code	2
1.4	How to support this program	4
1.5	What you need	4
1.6	How do I use this thing?	5
1.7	What kind of name is that?!?	6
1.8	Solutions to some possible problems you may have	6
1.9	Feathers in My Cap!!! !!	7
1.10	What is working so far	7
1.11	Black Eyes!!!	8
1.12	What doesn't work yet	8
1.13	Curses...	10
1.14	God willing...	10
1.15	The proper way to get SNES music	11
1.16	How you can help	12
1.17	Send me an email	13
1.18	The people that have contributed to this project	13
1.19	No, not government propaganda	14

Chapter 1

Anonymous

1.1 Table of Contents

Anonymous - The Amiga SPC700 emulator

Introduction

What's an SPC700 anyway?

License

Cool legal routine, use it in your own code

Support

How to support this program

Requirements

What you need

Usage

How do I use this thing?

Solutions

Solutions to some possible problems you may have

Features

Feathers in My Cap!!! !!

Misfeatures

Black Eyes!!!

Known bugs

Curses...

The Future

God willing...

Ripping

The proper way to get SNES music

Help wanted
How you can help

Contact
Send me email

Thanks
The people that have contributed to this project

History
No, not government propaganda

1.2 What's an SPC700 anyway?

This program is an emulator for the SPC700. The SPC700 is the sound co-processor of the SNES. The purpose of this program is to let you listen to music from SNES games on your Amiga.

I believe this is the 4th SPC700 emulator core in existence. Besides my own, there are 3 others from which all SPC700 emulators are derived. I hope that someday, this will be the most complete.

1.3 Cool legal routine, use it in your own code

-----Licensing Agreement-----

All files Copyright (c) 1998/1999 by Gaelan Griffin. All rights reserved.

License Agreement

This license applies to the computer program known as "Anonymous", which is referred to in this license as the "program".

The "author" refers to Gaelan Griffin who is the author (except where noted) of the program, documentation, and all other files included in the archive and who owns the copyright (1998/1999) to said files.

The "archive" refers to the package of distribution, as prepared by the author. The "files" or "data" refer to all elements of the archive.

Each licensee is addressed as "you" or the "licensee".

All conditions of this licensing agreement stated below must be met. If you do not accept ALL conditions of the licensing agreement you must destroy the archive and all accompanying data as well as any copies of the archive or data in your possession. If this licensing agreement is changed by the author then you must accept the new licence or destroy the archive and data as stated above.

You can not challenge the author's copyright.

No profit can be gained directly or indirectly by the licensee as a result of the use of or failure to use the program without the prior written permission of the author.

The program can not be used in any way that directly or indirectly supports the Microsoft corporation or any of it's subsidiaries or business partners (former or current) without the author's prior written permission.

The program can not be used by any employee or shareholder of the Microsoft corporation or any of it's subsidiaries or business partners (current or former) without the author's prior written permission.

The program and the data in the archive are freely distributable under the restrictions stated below, but are also Copyright (c) 1998/1999 by Gaelan Griffin. All rights reserved.

Redistribution of a modified version of the program, the archive or the contents of the archive is prohibited in any way, by any organization, regardless whether commercial or non-commercial. All files must be kept together, in their original and unmodified form.

Commercial distribution or inclusion of the software in any collection including, but not limited to, CD-ROMS or cover disks is forbidden without the author's prior written permission with the sole exception of Aminet and Fred Fish CD-ROMS. The exception previously stated is only applicable as long as the author does not state otherwise and as long as all conditions of the licensing agreement are met.

Distribution may not violate the licensing agreement or copyright in any way.

A nominal fee to cover copying costs is allowed, but none of the files may be sold for profit.

The author has the right to request other requirements (such as shareware fees) in addition to the licensing agreement as will be noted in the documentation contained in the archive. Acceptance of the licensing agreement does not imply acceptance of such requirements.

THE PROGRAM IS PROVIDED TO YOU "AS IS", WITHOUT WARRANTY. THERE IS NO WARRANTY FOR THE PROGRAM, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM AND ALL RELATED FILES IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE

PROGRAM INCLUDING, BUT NOT LIMITED TO, LOSS OF DATA OR DATA BEING RENDERED INACCURATE, LOSSES OF ANY KIND SUSTAINED BY YOU OR THIRD PARTIES, OR THE FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IF YOU DO NOT ACCEPT THIS LICENCE, YOU MUST DELETE THE PROGRAM, THE ARCHIVE AND ALL DATA CONTAINED IN THE ARCHIVE FROM YOUR STORAGE SYSTEM. YOU ACCEPT THIS LICENCE BY USING THE PROGRAM AND/OR REDISTRIBUTING THE ARCHIVE OR ANY DATA CONTAINED WITHIN THE ARCHIVE.

-----End of Licensing Agreement-----

1.4 How to support this program

Read the
Licensing Agreement
if you have not already done so.

Whew! Now let's get down to the fun stuff. This section will contain information on any requests that I ask for in exchange for using my program.

There are no other requirements in addition to the Licensing Agreement at this time.

(Of course, if you want to send me an email or something anyway, without being asked, that's perfectly acceptable and will be greatly appreciated!)

This may change in the future; it's possible that I might request that you send me an email for using my program or perhaps a small gift. It might even become "shareware" someday.

One thing I have no intention of doing (but reserve the right to do anyway) is making this program crippled in any way. This means no disable features for unregistered users, no keyfiles, no evaluation versions, and no copy protection. Registered users will get the same program everyone else has. Maybe this seems unfair to anyone who actually registers the program but at least I'll know they did it because they wanted to, not just to get a keyfile. I'll probably do something nice for registered users anyway. Also, certain people will never be required to pay anything. If they do, I will send the money back :)

That is, IF I ever accept registrations. It's possible that I never will. I guess it depends on how much support I get.

1.5 What you need

Minimum requirements:

- Any Amiga with a 68020+ processor and WB 2.0 or greater
- About 100k of RAM (preferably fast RAM)
- AHI V4 or greater installed. Get it from Aminet: mus/misc/ahiusr.lha
- Some SPC files to emulate. Try www.chez.com/raist/spc700 or come.to/amispc

Optional:

- XPK installed

1.6 How do I use this thing?

The only installation necessary is to put the program somewhere in `your` path. You may copy it to `C:` or `RAM:` for example, or anywhere you like just as long as it is in the system path. Use the "path" command if necessary.

At the command line simply type

```
Anonymous  
<file.spc> [volume boost] [additional options]
```

This is the argument template: `SPC/A,B=BOOST/N,P=PORT/K/N`

File is the name of the SPC file to emulate. It may be packed with XPK. It does not need to have the `.spc` extension.

You may add an optional volume boost number anywhere on the command line after the file argument. The lowest value allowed is 2, and the highest is 8. Values outside this range will be ignored. Also, you should not use values greater than 4 when using a stereo mode that doesn't support panning as the results are undefined.

You can stop emulation at any time by hitting the return key.

During play, you can hit the number keys to play subsongs, and the shifted number keys (may not work properly on all keyboards) for jingles or sound effects. Note that the effects may differ with each SPC file since these key presses just write data into the IO ports and each replay may assign a different meaning to them.

Let me explain:

When you press a number 0-9 that number is put into the first data port. When you press the shifted numbers '!'-'')' the equivalent number is put into the second data port. Now, it's completely up to the replay to determine what the data means. It may mean to play a subsong, a sample, or to stop the music. Play around with these and see what happens. It's often possible to eliminate unneeded SPC files this way.

Not all keymaps work the same, as a result, subsongs may not work as

intended. For your reference, these are the keys:

```
Port 0 - 1 2 3 4 5 6 7 8 9 0
Port 1 - ! @ # $ % ^ & * ( )
```

You can optionally specify an optional port handler type by using the P=PORT/K/N argument. The port handler is simply the emulator's response to any I/O port writes by the SPC700. Using this option can improve the emulated replay's response to any of your I/O port writes and theoretically could improve the replay emulation, although this is unlikely in most cases.

The default is to ignore all SPC700 I/O port writes.

Specifying port handler 1 will clear the corresponding read port on an SPC700 write. This is sometimes helpful, especially for playing sound samples.

Specifying port handler 2 will cause the same value the SPC700 writes to appear in its read port. This is not known to be useful, however it might work in some rare cases.

The interface could use some improvement, but it's not a high priority right now. When the emulation has improved significantly it will become a deli/eagleplayer.

Due to the use of the AHI mixing routines, the output will be very soft so turn up your monitor/speakers if necessary and please don't forget to turn them down again when you are finished. This will be fixed in the future.

It uses a minimal stack.

It should only be run from the CLI.

That's about it.

1.7 What kind of name is that?!?

Yes, it's called Anonymous. This is because I haven't been able to decide yet between the many good names I've come up with. This will change in the future, probably when the emulation quality has improved.

1.8 Solutions to some possible problems you may have

If you get an XPK error -3, don't panic. You probably just typed in the filename wrong.

Try hitting the subsongs keys. It's possible that a particular file may be waiting for its very first song command.

Try turning up the volume on your monitor/speaker. Don't forget to turn the volume down again when you are done. ESPECIALLY if you like to listen to chip music! This may not be needed if you use the volume boost option. However, volume boost may occasionally cause distortion. Also, you should not use a higher volume boost than 4 with stereo modes that do not use panning (The stereo modes without ++)

Adjust AHI settings of the default music unit. Try setting it to one of the 8-bit fast modes and reduce the mixing frequency. Also, setting the AHI CPU usage to 100% can help in some cases, but this can be dangerous. If it gets overloaded, the system will slow down tremendously and your mouse pointer will probably be sluggish. If this happens, don't panic. Just keep hitting the enter key until it registers and wait until the system is functioning properly again, which may take a few minutes. There is also another effect of the CPU usage that deserves some attention. The music may slow down if it's too high in some cases, so if it's not playing at full speed try lowering the AHI CPU usage speed.

Try to give it as much CPU time as possible. Kill as many other tasks as you can and boost the priority.

You may have wait a few seconds for the music to kick in.

If all else fails try a different file. In particular, visit the website to see the recommended ones.

1.9 Feathers in My Cap!!! !!

- Completely system friendly.
- Uses AHI for output.
- Uses very little memory
- Written from scratch in 68020 assembler
- Multitasks
- XPK support
- Does NOT have an installation script (that's a GOOD thing!)
- Tested with Enforcer and Mungwall
- Excellent documentation. Please read it, it's not boring at all!

See also,

What's Emulated

1.10 What is working so far

CPU emulation

244/256 opcodes

NVPZC flags

DSP Registers

For each voice:

- Mono Volume
- Pitch
- SRCN

General:

- Key On
- Key Off
- EndX
- Dir

...and lot's more.

1.11 Black Eyes!!!

- Uses AHI for mixing (which prevents some DSP effects)
- Written in 68020 asm (not portable for, e.g. PPC)
- Doesn't multitask very well (uses all the CPU time it can get)
- Has a simple CLI interface with poor input handling

See also,

What's Not Emulated (yet)

1.12 What doesn't work yet

CPU emulation

The H, I, and B CPU flags. The I and B flags are used for interrupts which are rarely used, I haven't seen any replays that use interrupts yet. There are no instructions for testing the halfcarry flag, so it's a bit difficult for replays to use it. I have reason to believe that some replays depend on it, however, so I'll probably be adding support for it.

Unemulated CPU Instructions:

DAA
DAS

BRK
RET1

AND1/OR1

EI/DI
SLEEP/STOP

I haven't seen these instructions used by replays very often, so I'm going to leave them unimplemented to help detect emulation errors for now.

The extra RAM is not emulated

There are also 3 APU registers I'm unsure about: \$F0,\$F8,\$F9

Timer emulation is imperfect, but seems to work well enough for now.

Opcodes and their operands are not memory mapped for speed. This shouldn't cause any problems unless someone is trying to be cute and does a JMP \$F4 This can be fixed simply by setting a symbol during the assembly process, if needed.

DSP emulation

Many registers are not yet implemented:

For each voice:

- Stereo Volume
- ADSR
- Gain
- ENVX/OUTX

General:

- Main Volume (L and R)
- Echo
- Echo Volume (L and R)
- Echo Feedback
- Echo Delay
- Flag
- Mute
- Noise
- Pitch Modulation
- FIR Filter

The DSP itself isn't emulated, it is simulated. Emulation of the DSP is possible and would greatly increase the quality of the emulation, but the DSP roms need to be ripped first and this is very difficult to do. It is certainly beyond my skills.

Miscellaneous emulation

The SNES itself is not emulated. Although not necessary for most purposes, the 65816 code and data could be useful and may be even be required in some cases. This will be necessary for my vision of
ripped music.

I hear there is also a hardware analog filter for the sound output ↔
, but
it may be difficult to simulate.

1.13 Curses...

There is a small memory loss the first time the program is run. I believe this is AHI's fault as I've seen other programs that use AHI exhibit this behavior. This mem loss is small and is one-time only so it shouldn't affect anyone. I don't know if it can be fixed or not. If you have any information concerning this, please contact me.

It can sound terrible if it can't get enough CPU time, especially on slower systems.

Other than that, it appears to be pretty stable. If you notice any enforcer or mungwall hits or any other bugs then please contact me, (Don't assume someone else will!) but try to troubleshoot the problem first.

When reporting bugs give me as much information about the problem as you can. Also send your system configuration, what files were causing the problem, any system patches that conflicted, etc.

1.14 God willing...

Better CPU and DSP emulation, of course. I need to add some extensions to my debugger to help track down the bugs.

PS3M based mixing routines -

This should greatly improve the sound quality, as well as the speed. Currently I use AHI to do the mixing, which isn't very good for most people (except maybe soundcard owners?) It also limits some of the DSP effects I can do. I will still use AHI for output, but it may be optional, since it can really slow things down if I've just mixed the sound data and then AHI has to remix it to a different frequency, as can happen sometimes. Also, the sound will no longer be so soft.

Fix the timings -

Currently, I don't even bother limiting the speed, the emulator just runs as fast as it can. This isn't very good for 060 owners. It's also the (partial) cause of the poor multitasking. It really would work better if I disabled multitasking, but then AHI wouldn't work.

Sample caching -

This should speed things up a bit, at the expense of memory. It will take approximately 300k, which isn't too bad, but will be implemented as an option because I like the fact that currently only 100k is needed.

CPU shutdown -

This involves detecting timer loops and stopping emulation until the specified time occurs. This should give a big speed boost in many cases, but will slow emulation down anytime the CPU is running (which is potentially all the time, if the loop isn't detected.)

CPU optimized decoding routines -

It should be pretty easy to optimize the decoding routines for various cache sizes which will speed things up a bit.

CPU optimizations -

Some things can be done here, perhaps at the expense of memory.

Of course, I plan to turn this into a deli/eagleplayer eventually.

I'd like to write some replays for the SNES someday:

- Protracker as well as some of the multichannel clones like 6CHN/8CHN
- MED/OctaMED but nothing with more than 8 channels

There is a technique that can be used to play uncompressed 8-bit samples on the SPC700. With proper use of this, it should be possible to play all 8-bit mod formats with near-perfect quality, as well as some more interesting features such as sample mixing (which is normally difficult to do as you have to uncompress the samples, mix them, and then recompress them. This takes a lot of time, especially the recompression, if you want the samples to sound good.)

Some fun ideas for later:

Make a special version for minimal systems. I think it may be possible to get the emulator running on an unexpanded A500.

Make it pure.

Turn it into a reentrant shared library.

Overlay parts of the code to minimize memory usage.

Add support for dynamic/static recompilation and other sophisticated emulation techniques.

1.15 The proper way to get SNES music

SPC files are SPC700 emulation save states, written out by some program. Personally, I find this method to be very poor, but at the moment, there is no other easy way to get the music. I'm working on a file format for ripped music which will be MUCH better than save states, but it will not be ready for quite some time. If you're interested in ripping, send me an email.

The file format will be very powerful and easily extensible. Unfortunately I need to rip a lot more music before I'll have a good idea of what's needed. I'll also need a SNES backup utility of some sort.

Some possible advantages of ripped music:

- Properly ripped files will be perfect, unlike memory saves which can have errors due to emulation deficiencies. Music will start at the actual beginning instead of whenever the SPC file was saved. Also, it is easier to compare two different ripped files than it is SPC files because it is highly unlikely that two different SPC files of the same

music will be exactly the same on the binary level.

- Ripped music can be contained in very small files that can get their data directly out of a ROM file, thus reducing file sizes tremendously.
 - Real subsongs can be implemented, as well as jingles and sound effects.
 - Files can contain information about the replay resulting in more efficient playback. Files can also contain other information such as author/ripper credits, alternative song names, annotations, etc.
- One of the biggest advantages would be the ability to hold information about where the CPU can be shutdown. Some replays use very complex methods for polling the timers which are nearly impossible to detect with 100% accuracy. This isn't a problem if the ripped music contains the necessary information obtained by analyzing the replay.

My ultimate goal is to create a complete collection of all SNES music, properly ripped. Then I'll start working on some other console sound chip.

1.16 How you can help

I REALLY need some sort of "backup utility" for my SNES in order to do research on the SPC700. Anyone willing to donate one to the cause? Or sell cheap? I live in America, so preferably someone from this country to reduce shipping costs.

Betatesters wanted, email me to apply. Some of the types of people I'm looking for:

- Anyone with good debugging skills, knowledge of AHI, digital sound theory, or emulation experience. It's also helpful if you have a wide range of systems to test it on. You also must have the time and willingness to conduct tests as often as needed.
- People with soundcards.
- People with lots of SPC files who would be willing to test them with every release of the emulator and record their observations. To reduce the workload, you will probably only be responsible for testing a certain group of files with the rest being tested by other people.

Also, if you have the ability to run an SNES emulator that supports the saving of SPC files let me know. If you have a PPC card you should be able to run SNES9x (WITH sound) very soon and that will do nicely.

If you have any SPC700 related files, let me know. Check the webpage first to see if the files you have are already there. In particular, I'm looking for The Legendary Amiga SPC700 Ripper, but anything that isn't in the download section will do. I don't mean SPC save state files, but SPC700 related documents or programs (amiga only, unless they include source.)

I have the source to an SPC700 assembler, but it's unusable. If someone would like to work on it, or write their own assembler let me know. It's a real pain poking in programs with my debugger. I am certainly not

planning on writing any major replays that way.

Any information on deli/eagleplayer programming would be nice. Seems the EP docs have not been completely translated and I seem to be missing some stuff. I'd like to be able to use my debugger with the player, so docs on Genies/Engines would be most helpful. I have already done some player work (get my NES player!), but nothing this complex.

Also, if you have any NEWS, then please contact me as soon as possible.

1.17 Send me an email

Raist is the webmaster of the official webpage at www.chez.com/raist/spc700 or come.to/amispc I also help update the news.

Be sure to visit it for the latest news on ALL my SPC700 emulators, as well as to download other SPC700 and SNES related files. There is also a forum where you may leave public messages. This webpage will probably be at the center of my SPC700 work, so check it often.

Feel free to contact me for any bug reports, constructive criticism, ideas, praise, donations, information, or whatever.

My email addresses:

chinoclast (at) softhome.net

chinoclast (at) mailexcite.com (The upgrade sucks; I don't check it often)

NOTE: If you haven't received an email reply in more than a week, write me again. It's possible that it got lost somewhere or that I didn't realize you wanted a reply.

1.18 The people that have contributed to this project

David Gonneau (Raist) - For making a great webpage!

Dave Hng - For providing useful info and data

Gau - For his SPC700 programming information

Jesper Svennevid - For his amiga port of SNES9x

Ledi - For his APU manual

Martin Blom - For AHI

Savoury SnaX - For releasing the source to SNEeSe

Steffen Haeuser - For WarpSNES, his WarpOS port of SNES9x

The SNES9x team - For releasing the source to SNES9x

The XPK team - For the best packing system in existence

.Z-chan - For providing SPC700 information

Special thanks to my Amiga emulation friends:

Morgan "A/NES" Johansson
Ville "Wzonka-lad" Helin

Special Guest Star: Peter "Delirium" Kunath

Very special thanks to:

My family, especially my father
The Trinity

1.19 No, not government propaganda

V0.40 19/Mar/99

- CPU emulation has improved significantly, I don't know of ANY replays that fail with this version. If you find any SPC files that fail, let me know.
- Dramatic improvements in sound quality, SPC files should sound much better
- Added support for ReadArgs and optional port handlers
- Documentation updated

V0.31 12/Jan/99

- Added optional volume boost, append a number [2-8] to the command line
- I REALLY need a console copier for my SPC700 research

V0.30 9/Dec/98

- First "real" release.
- Several bugs fixed, more files work now.
- Subsong support
- Emulation is faster.
- Extensive documentation written.
- FANTASTIC Webpage by Raist at www.chez.com/raist/spc700/

Compatibility is now greatly improved; many SPC files will run now. This is the first official release and it is accompanied by an official webpage. Excellent documentation is included. Please read it.

V0.0 10/Sep/98

First preview version uploaded to Aminet. I fixed a bug that crashed PAL machines and re-uploaded it again later that night. This was a very poor version; I just removed the debugger and put some code in to keep it running alone.
