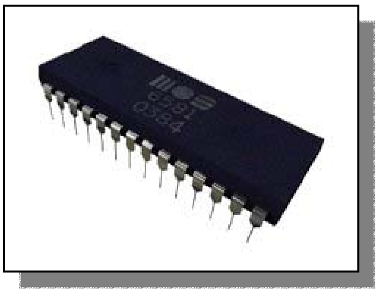
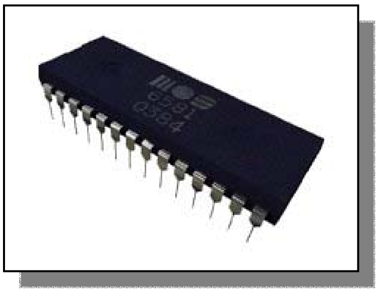
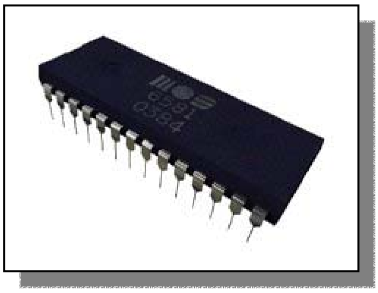
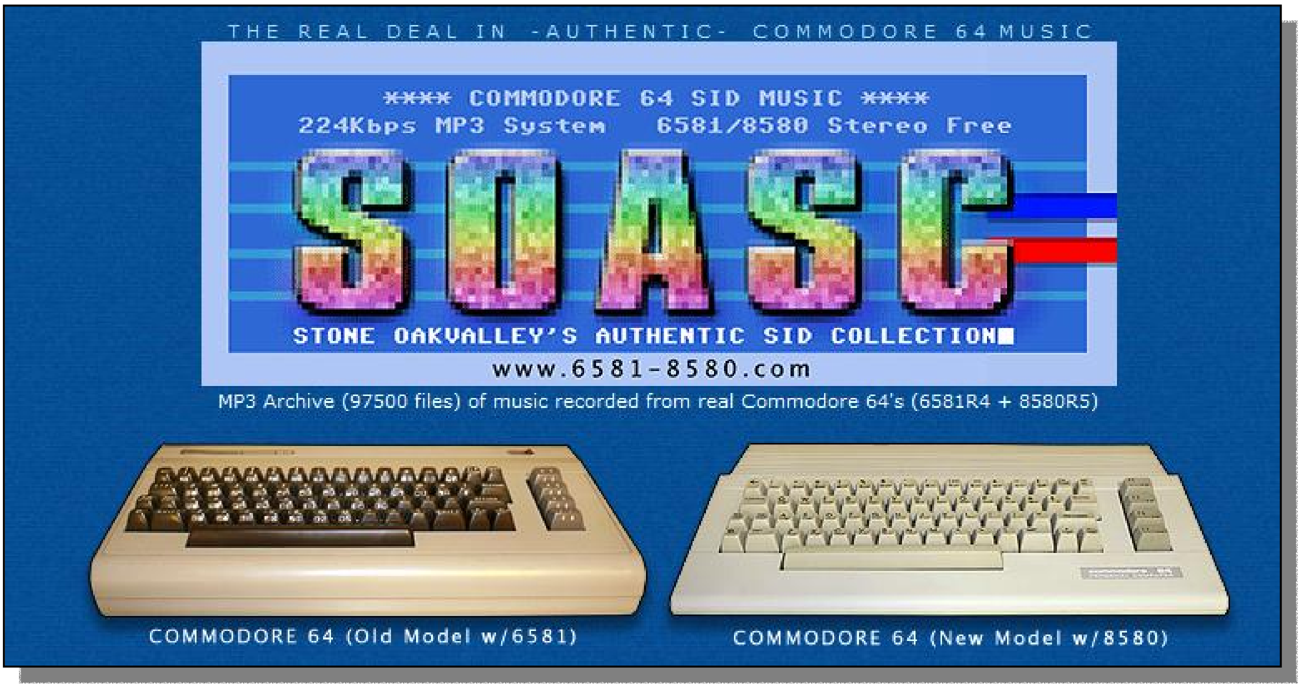
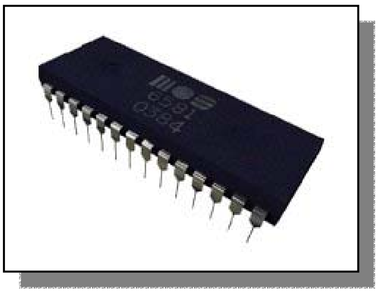
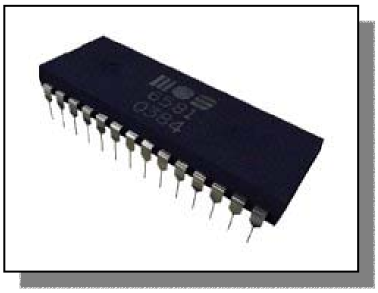
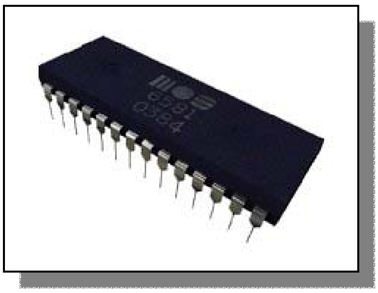
# Commodore Free

## Issue Number 12
## September 2007

Free to download magazine dedicated to Commodore Computers
Available from www.commdorefree.com in the formats of PDF TEXT HTML and D64 image


THE REAL DEAL IN -AUTHENTIC- COMMODORE 64 MUSIC
**** COMMODORE 64 SID MUSIC ****
224Kbps MP3 System 6581/8580 Stereo Free
SOASC
STONE OAKVALLEY'S AUTHENTIC SID COLLECTION
www.6581-8580.com
MP3 Archive (97500 files) of music recorded from real Commodore 64's (6581R4 + 8580R5)
COMMODORE 64 (Old Model w/6581)
COMMODORE 64 (New Model w/8580)

# Editor

Well some people may say the edition is over dominated by the **SOASC=** project. I feel this as an absolutely insane project and totally mad, a superb effort from a dedicated sid fan well done.
What's this SOASC= project all about? well you need to read the issue and all will be revealed

I am no artist as you can tell and have designed a new logo for the magazine, the old one was really a stop gap because, it was a case of using that or else nothing. I am fairly pleased with the finished logo being a none arty type person.

Time was not with me on this issue

Thanks
Nigel
www.commodorefree.com

HOW CAN I HELP COMMODORE FREE
Ok the best way to help would be "write something about Commodore" (yes for the observant I spelled the company correctly this time) _grin seriously though articles are always welcome,

WHAT ARTICLES DO YOU NEED
Well they vary contact me if you have an idea but I am looking for

Tutorials – (beginners and Expert)
Experiences with Commodore
Why I love Commodore machines
Interviews – maybe you have access to a power user

Thanks Nigel
www.commodorefree
commodorefree@commodorefree.com

# Contents

# NEWS

**New Version of WinUAE**
The Amiga Emulator for windows has been released
http://www.winuae.net/

Wanted: Non-emulated Freezer Cartridges
======================================

- X-Power Professional 500 (non-v1.2)
- Nordic Power (non-v2.0)
- Bus Stop
- Pro Access
- Action Cartridge loader disks/disk images. (non-v1.3 german)
- Action Replay (non v1.00, v1.50, v2.05, v2.12, v2.14, v3.09, v3.17)

(or cartridge ROM images if you have (EP)ROM reader. Note: ROMs are scrambled.)

WinUAE 1.4.3 (29.07.2007)
========================

New features:

- Built-in lha/lzh and lzx support.
- Mount archives as a harddrive with transparent, recursive (archives inside archive) decompression. Supported: zip, 7zip, rar (unrar.dll or archiveaccess.dll required), lha/lzh, lzx.
- A3000 Kickstart ROM and SuperKickstart disk support.
- A590/A2091 SCSI, A3000 SCSI and CDTV SCSI expansion harddrive (HDF) emulation (WD33C93 + (Super)DMAC based SCSI hardware).
- Action Cartridge Super IV Professional freezer cartridge emulation.
- X-Power Professional 500 (v1.2) freezer cartridge emulation.
- Nordic Power (v2.0) freezer cartridge emulation.
- Debugger improvements (improved deep trainer, copper memwatch points,CPU-model specific registers can be modified, illegal access logger improved, process breakpoints etc..)
- Paths-panel default paths selection improved.
- Separate native and Picasso96 vsync setting.
- GUI will "autoscroll" if fullscreen mode is smaller than GUI.
- Improved rtg.library, speeds up Picasso96 in high resolution modes (obsoletes picasso96fix)

Bugs fixed:

- CDTV emulation improved (DOTC2, Xenon2, ChaosInAndromeda CD player)
- CD32 CD emulation improved (Fightin' Spirit, Base Jumpers etc..)
- Ghostscript printing fixed (again).
- Floppy drive sound selection if fdrawcmd.sys was not installed.
- Video recording sound pitch issue.
- -datapath command line parameter fixed (again..)
- uae-configuration JIT on/off switching fixed.
- Sprite attachment fix, fixes "Great Demo" by "The Tremendous Trio" :)
- Some FPU fixes from Aranym.
- Directory filesystem locked files (most commonly s:startup-sequence) after software reset
- Filesystem emulation not initializing if JIT was enabled and no other expansions enabled (fast RAM, Z3 fast, etc..)

**VICE 1.22 Released**

* Changes in VICE 1.22
=====================

** C128 changes
---------------
- Added 2 MHz mode support (experimental).
- The cursor keys are mapped differently in C64-mode now.
- Fixed C64-mode autostart support.

** VIC20 changes
----------------
- Improved the sound emulation where the 'volume change click' is concerned, and normalized the audio output level.

** VIC-II
---------
- The VIC-II border mode can be selected now (normal, full, debug).
- Some sprite fixes needed for Krestage 3 demo.

** Drive changes
----------------
- Improved drive LED emulation.

** Unix changes
---------------
- Fixed the "black screen" bug caused by some X11 library security update.
- Fixed the usb support for bsd based platforms.
- Changed the preferred libdir and docdir for netbsd and freebsd.
- Xaw/XRandR fullscreen mode is supposed to work.

** MS-Windows changes
---------------------
- Positional keyboard mapping is used as default again.
- New volume slider control.
- The win32 port can now be compiled with openwatcom.

** OS/2 changes
---------------
- The os/2 port can now be compiled with openwatcom.

** RiscOS changes
-----------------
- Added a build script for the RiscOS port and all needed binary files are now part of the source distribution.

** AmigaOS changes
------------------
- Added netplay support for AmigaOS3 port.
- Added netplay support for AROS port.
- New VICE Volume control for all ports.

** C1541 changes
----------------
- Fixed some unlynx bugs.

http://www.viceteam.org/#vice

## SCACOM-Aktuell issue 1

SCACOM-Aktuell issue 1 (August 2007)

SCACOM-Aktuell is a new PDF magazine in German language for Commodore and Amiga fans.
Issue 1 (August 2007) got released today.

Topics are:
A magazine introduces itself...
Who is creating this magazine?
Renderering pictures - part 1
Tutorial - CF cards on Amiga
Where Commodore computers are being used nowadays?
How does Jack Tramiel look like these days?
PSP - The all-rounder

To download it, go to

www.scacom.de.vu,

click SCACOM in the menu on the left, then "SCACOM Aktuell Ausgabe 1 (August 2007)" and finally the link "August 2007".

## NEWS Project Update DC2N

> Hi Nigel,
>
> basically what I am still in need of is the design
> tool for the final circuit boards. I have been
> trying to design a cardedge (check the attached
> image) in the CAD I use (eagle by CadSoft) but I
> wasn't able to do so without a trick that requires
> manual cut of the board.
> I will research more on this because an on-board
> cardedge would be the choice for the final board
> (yes the one that will come inside a nice box with
> some buttons, the LCD, all the connectors, and maybe
> a PSU and a serial cable).
>
> It would help if you could show around this picture
> I sent you and ask people to give me help about how
> to design those cardedges in eagle, or have them
> designed with a different CAD software.
> Thank you.
>
> Cheers,
>
> Luigi

Hi Nigel,

I'm afraid there's something more interesting to publish. I told you I designed the missing part of the board, as per below:

http://digilander.libero.it/tcengineer/c64/dc2n/diary.html#29Aug07

Problem is that when we asked for a quote for 25 of them manufactured, they told us it would cost 6.30 pounds each!Way too much to afford them!

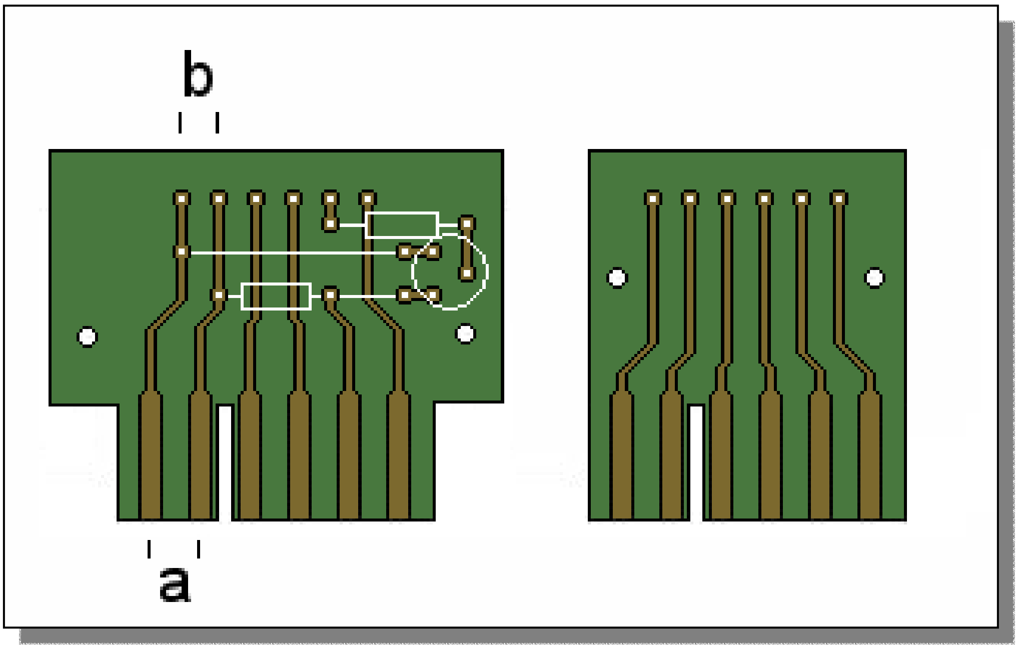
It may be helpful to find some hobbyist in UK that can help us to make them, instead of having them manufactured by a company that charges us so much.

Version 1.4 of the base DC2N board will incorporate that card edge connector, so there will not be reason of concern if we'll want to have it manufactured.
Question now is: can you please help me to figure out how many persons would like to have a DC2N?
It will cost 35-40 pounds about without any external lead or PSU if we manufacture 25 of these new boards (some components come from USA, shipping is 35 USD, customs charged me 18 euros the last time I ordered some parts). Of course, if 50 of them are manufactured, the price would be smaller...

Thank you for your help.

Cheers,

Luigi



# The SOASC= project

The SOASC= project is a non-profit and a private project.

The SOASC= project is an automated recording technique invented by me (Stone Oakvalley) in order to mass record music from the legendary Commodore 64 and its SID chips (6581 and 8580).

The goal is to record the entire HVSC SID collection played from REAL Commodore 64's (both old and new) as per collection #46 (January 21 2007.) With PSID64 as the REAL C64 player and 64HHD as fileserver, it all connects to multiple PC's with own tailored software, crude PAR: port control system/C64 keyboard interface and database structure toolswritten in PureBasic. Yes, its actually working.

Also, a strong point to consider in this project is that ALL SIDs are recorded on both Commodore SID chip models regardless of what HVSC or the author of the SID had recommended. Remember: There are people out there that probably NEVER heard the elite sound of the 6581 and its sample/filter defects, but only the sound of 8580 and visa versa. For those people, they ONLY rembember the tune as played by their model. And this is a strong point of the SOASC= project: PRESERVATION for ALL!

If it crackles and pops....well..it's the true and authentic sound of a real Commodore 64! This is what we had in the past, and now the past will be the present for all Commodore 64 fans out there.

It is called AUTHENTIC because the process will NOT attempt to enhance any of the recordings, it is recorded straight plain out from the mono Commodore 64 Audio/Video connector. No stereo, no funny mixes, no compression, no filtering, no remix, no software noise reduction, no crazy SID hacks or other unatural Commodore 64 elements.
If there is a poppy click in the recording its supposed to be there. The SID chip is unique as should be treated as so as well.

I've adjusted DC BIAS to 0 and executed Volume Maximizer with no clipping using NORMALIZE.EXE and SOX.EXE. Noise Reduction was done by improving the physical grounding inside Commodore 64. Thats it!

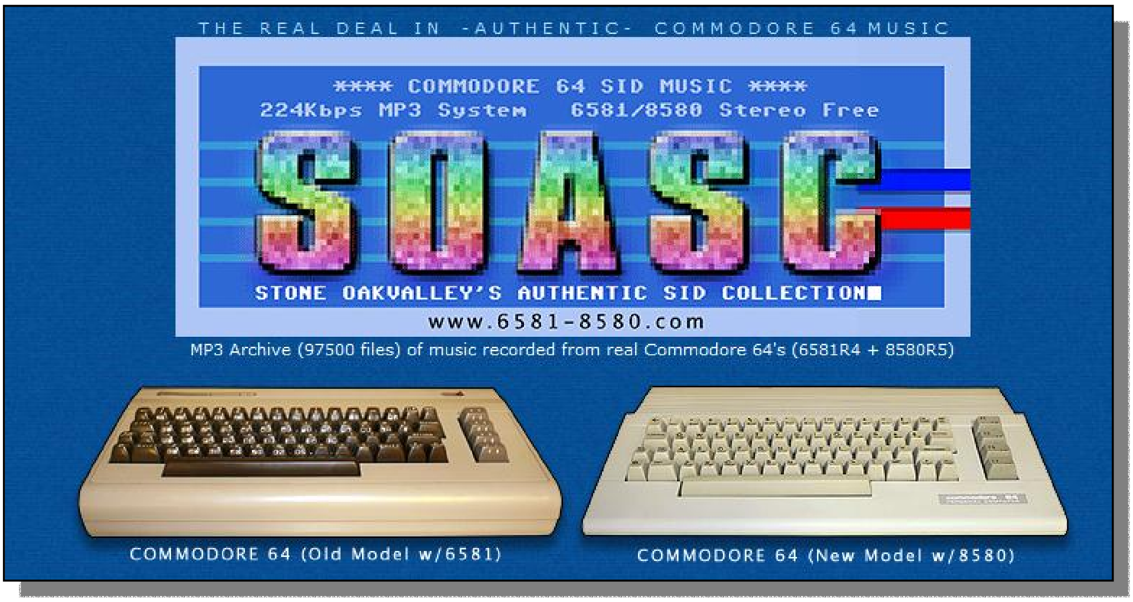
The final MP3 (224kbps, mono, 44100Hz) will contain all information from the SID itself, sorted in respect of the directory structure as defined by HVSC. Filenaming, title, author, copyright etc.

Forget PlaySID/SIDPlay2w....and all the others
Forget HardSID
Forget ParSID
Forget ReSID
Forget MMC64 (It has an intergrated SID player, for those who wonder)
Forget them ALL!

Just listen to the real deal instead with the help of the SOASC= project!

Note: I _LOVE_ all the software/hardware mentioned above..do not misunderstand me:-)

A project this big shouldn't be done...because it IS insane. But fret not: Because, ONE crazy man called Stone Oakvalley - will attempt this world record project in music conversion and preservation!


THE REAL DEAL IN -AUTHENTIC- COMMODORE 64 MUSIC
**** COMMODORE 64 SID MUSIC ****
224Kbps MP3 System 6581/8580 Stereo Free
SOASC
STONE OAKVALLEY'S AUTHENTIC SID COLLECTION
www.6581-8580.com
MP3 Archive (97500 files) of music recorded from real Commodore 64's (6581R4 + 8580R5)
COMMODORE 64 (Old Model w/6581)
COMMODORE 64 (New Model w/8580)

# SOASC=

**FAQ - Frequently Asked Qestions:**

There are bound to be people asking themselves why I did it THAT way? Or just people beeing incredible critical, just because they have a sound system that are 20 years newer and can TODAY bring forward that awful static noise, and details you'd never heard before in authentic Commodore 64 music! This FAQ below will logically answer ALL those questions.

http://www.6581-8580.com/index.htm

A: No, its not. The keyword is "emulate". It means to simulate or reproduce something in another environment that its original environment. An artificial environment. Is this real then? No. Therefore, the music MUST be recorded from the real environment to ensure the authentic and genuine sound of the C64. Recording from hardware cards or software emulators of the newest kind is NOT authentic. That's the easy way, I do it the hard way..so there! But, don't get me wrong. The work done for the software emulation of the SID chips are really a incredible task. Respect!

Another thing is that the SID chips have incoming capacitor lines which are made out of natural elements, and this means that the filters are impossible to simulate on a computer 100%. And do rememeber that 2 similar chips and C64's will NEVER have the exact sound on both! The filters are based on nautral ingredients (which are of the analogic world) and therefore there will naturally be deviations from emulation, clones vs the real thing! C64 is not living in a digital filter sound world... and thats why the sound is so incredible and this also apply for a lot of old synthesizer equipment from the 80's... like the Roland TB-303. Pure analog sound which is NOT comparable to software clones of today. (But that's another battle story...)

Q: "Cool, real hardware is the way to go, but what chip revisions & batch are you recording on?"

A: The chips used for recording is: "MOS 6581R4 3387 14" (Yes, no AR markings!) and the "C= CSG 8580R5 2689 25". Hopefully the whole 90000+ tunes will be recorded on these if they do not fail during the 3-4 month process!

Q: "Hey, my favourite tune sounds wrong, I can't remember this version!"

A: In most cases the tune are authentic and is exactly how it sounded when played on a real C64 with 6581 or 8580. Remember that composers designed tunes to be perfect on the 6581, and when 8580 came out a few years later the damage was of course not possible to fix. They could not have known that the filters was changed on the 8580. This is the most important part to remember. Tunes ARE and WAS specifically designed to that of the composer had in his machine. But how the hell can you know what the composer intended? I many cases, you don't - so please use the guidelines below for help! Guidelines for choosing the correct MP3 version to download/listen to:

1: Make sure you download the MP3 file (either 6581 or 8580) as suggested by HVSC and also indicated in our database. If still not sounding okey, proceed to step 2.

2: Download the SID file instead. Tweak settings in SIDPLAY2w to force 6581 or 8580 model to use. If the same problem (missing sounds or channels) is there, then the SID file is supposed to sound like that on the opposite model or the SID is a bad rip (difficult to determine).

3: There are known differences between 6581 and 8580 recordings. For instance the sample playback on 8580 may be low or missing. Please try the 6581 version instead. Even between revisions of the same chip the sound can be different, so remember that!

4: Please try and remember what kind of C64 you heard the tune on originally "back in the days". Download the appropriate MP3 file, and that should be it.

5: Also, to confuze you even more. The MOS6581R4 used for recording has some really strong filtering, so if thatis not pleasing or you can't remember, a safe say would be that the MOS6581R2 version is the one you seek!

6: Download all MP3 versions (6581 and 8580) and choose the most suiting one for your ears and stick to that!

General: The sound of C64 is analogic and sounds differently on each chip and even between the same revisions!

So a quick estimate from me:
1982-1988 = Go for the 6581 versions (100%)
1988-1993 = preferred model unknown
1993-2007 = preferred model 8580 (but still not 100% trustable)

BUT also remember that tunes made 1982-2007 WITH samples will not sound correct on 8580 (in most cases) so that too is a little bit confusing I'm afraid. If the problem is still there and you are certain that something HAS gone wrong during the recording, please post the bug and we shall investigate and give feedback.

Why we didn't do this intially during the pre-recording process is simple: People was used to the music they heard on whatever model they had regardless of what the tune was originally composed on "back in the days". So, if a tune was suggested to be played with a 6581 model (like tune 9 in Last Ninja) it would have a special filtersaw string in the background. But, on a 8580 the same sound would not be present. So, for people only listening to tune 9 in Last Ninja on a 8580 model, they would not have recognized the "new" filtersaw string in the background if they downloaded the 6581 MP3 version of today. So, it was a easy decision: We HAD to record everything on both models to suit everybodys childhood memories: )

Q: "Hey, my favourite tune is missing 1 to 0.5 second (start or end), and the looping is not perfect!"

A: The tune lengths were extracted from HVSC own songlengths.txt file which do not contain that much of precision. Furthermore, it is well known that a number of song lenghts are really wrong. The songlengths.txt is a beta project still. But in time, HVSC will probably take care of this, and even adjust the lengths to more precise numbers. Its rounded to the nearest second. I have added about 0.8 second to the recording in my SIDREC software to compensate. This means that your favourite tune or sound effect will either be perfect, missing 0.2 second or having 1.3 second too much making a seamless loop not possible. The other thing you have to remember is that the SOASC= recording is an AUTOMATED process and there is NO WAY I could load each tune into Adobe Audition and manage this for approx 95000 tunes. That is IMPOSSIBLE. BUT, of course such things are irritating, so requests for re-recordings and manual mastering is an option through our FORUMS!

Q: "Hey, my favourite tune is having a click/pop audio problem somewhere!!!!"

A: Some of the SID tunes played on a real C64 contains peaks beyond what you could believe. Its a analogic world on the C64 and things CAN get out of hand. I have not used any software to prevent clipping as this can destroy certain elements of sounds in other tunes probably. Furthermore, the audio in recording volume is ONLY about 25% of the REAL signal. This should prevent clipping in 99.8% of the songs. After the recording a NORMALIZE function is performed on the tune, maximizing it to the fullest volume possible. If the peaks are already recorded with 25% volume, the peak will naturally still be there. The click/pop problem is a minority of the tunes. You should try the 6581 or 8580 version to determine if the problem is on both of them and report back to the FORUMS. Maybe even request a re-recording where the clipping is manually removed by us. And again, remember the SOASC= project is an automatically based project, no human involment to fine tune each SID song. Some sacrifices will be present, but all things can be fixed. There is hope, "keep hope alive" (C) TCM!

Q: "Hey, almost all tunes have an annoying click in the beginning!!! WTF?"

A: Yes, this is the actual software INIT being done by PSID64 for the SID chip. Its just the same click you will hear on a real Commodore 64 if you were to start the actual game or demo yourselves. I tried to fine adjust the recording to avoid this, but there are some minor milliseconds to which to work on, so a lot of tunes were chopped off about 0.1-0.5 seconds in the beginning instead. So, I decided to adjust it back to make sure the tunes were not chopped off in the beginning. Sorry for this, its just the nature of the SID chip....and since the SOASC= is authenthic I guess it have to be in there...uhh. But maybe it wont play on all MP3 players both hardware and software, depends on how for instance you have the fading between songs (ie crossfading in WinAmp) or that your ipod ignores the first 0.1 seconds and skips it...you never know :).

Q: "Hey, Have you guys deleted some recorded files, where are the "_PSID" recordings?!"

A: The _PSID are files that were back in the Amiga days hacked to be played perfectly with samples when using Amiga PlaySID. Today, the result (when played on a real Commodore 64) is sample playback missing or totally screwed up etc by using PSID64 (which the entire SOASC= was used for). I think there are possibilities to record even the PSID files by using MMC64 or another dedicated SID player on the C64 itself, but this has not yet been researched upon. In time I will investigate and prepare them for new re-recording upon the next SOASC= major recording process during 2007. Today, the PSID files resulted in incredible noisy files when processed by the SOASC= recording technique for instance. This is also mentioned in HVSC FAQ, and they have intentions to re-rip all _PSID tunes and make them real C64 ripped files which will be played correctly on a real C64. They are not really suitable and can't be trusted at all, so we filter them out!.

Today, most tunes are duplicated with both the PSID and the regular SID format anyway, so this question will be null and void during time.

Q: "What actions did you take to ensure recording of PAL/NTSC based tunes? "

A: This is a bit of a troublesome point. All tunes were played on both 6581 and 8580 PAL timed machines. This beeing of course that I live in the PAL area and have only access to PAL machines. Furthermore, the PSID64 player (used to playback the tunes in this project) might ignore the NTSC flag bit set in the original SID file. As the "TODO.TXT" in PSID64 states: "compatibility mode for PAL SIDs on a NTSC computer and vice versa."

I guess the development of PSID64 never included that, or was never fully completed. What the result is for the SOASC= is this: The NTSC specifically timed tunes will sound (in the MP3 version) a little bit slower and also be pitched slightly down (factor 1.038). This might also interfere with the "songlengths.txt" which is probably based on the PAL/NTSC timing accordingly and therefore some tunes will be in a sense "wrong" in terms of the AUTHENTIC factor!

So if a tune had a NTSC flag in it by HVSC standard (like "Norman_Paul/ForbiddenForest.sid") it will sound slower and pitched down on a PAL machine (or in the SOASC= MP3 version).

For a PAL specific tune, it would then ofcourse be played faster and pitched up instead if you did compare it to your original NTSC machine or by using SIDPLAY2 with the NTSC flag speed set on.

However I tried "Sidplay64 v0.4" by Glenn Rune Gallefoss / SHAPE / Blues Muz' and that actually played NTSC tunes different from what PSDI64 did, so I'm not sure what effect I will let this have on the current SOASC= collection. If there are numerous reports that certain tunes sound wrong, I might re-record them using the Sidplay64 instead!

Q: "Why the 224Kbps, mono and CBR MP3 coding?"

(I used LAME 3.97 Command line version to convert the recorded wav to MP3)
A: CBR is constant bit ratio which means that a silent period in the MP3 have the same compression factor as a period with sound in it has. This results in larger size MP3, but size is NOT an issue here, and also it's supported by older equipment (which also represents the 224Kbps ratio) such as DVD players, Car Stereo MP3 players, MP3 players both software and hardware. It IS yesterday's compression scheme for sure, but not everybody are hip enough to follow the bandwagon and be cool and buy all the latest ipod MP3 players and do not care about dowloading the latest MP3 player software. SOASC= is about compability in probably all environments and situations. MONO is of course the only choice when recording C64 music. C64 does NOT play STEREO sound out of the AUDIO/VIDEO connector and for the tech freaks, the CHIP inside (6581 and 8580) has only ONE audio output. Please remember, 3 VOICES OUTPUT (as written in the C64 manual) is not the same as STEREO SOUND OUTPUT, and therfore what good will a STEREO MP3 with the same audio in both channels be any point to consider??!

Q: "Why MP3? and not OGG, FLAC, Plain WAV or "that other personal favourite" format, dude?"

A: It was a totally egoistic and personal choice. This is afterall a PRIVATE project intented for my own amusement.

MP3: It's the most common format for all kinds of people and hardware. End of story.

OGG: Might be better with those "unhearable high frequencies that you would NEVER miss if you had nothing to compare too anyway!" End Of Story.

FLAC: Why should I? Better go real WAV instead, and also FLAC is a non-typical format that are not "all over" compatible "all over" everywhere. End Of Story.

WAV: That would be dream, yes. But it would take 10 times as much space. So 400GB x 10 that is. For the years to come, not an option. But in future, why not! :-)

I really can't believe if anybody would be unsatisfied with the SOASC= audio quality of the MP3's. Those higher frequencies you claim to be issue with the MP3 are frequencies in the approx 18000Hz-20000Hz range.

Note that the Hi Freq cutout example was boosted by 17DB, just to get the maximum volume for you to listen too.The Original Recorded WAV (8580) vs The 18000Hz-2000Hz range (you claim you can here this?? Maybe your dog will!)The Processed MP3 Result (8580) vs The 18000Hz-2000Hz range (is there anybody out there at all?) And the best of all. If anybody have better recordings, they are free naturally to enjoy those. But why not do the same thing with the rest of the current available C64 music as well... ohhh don't worry...it's just about 90000 tunes of manual processing work for you. See you in another dimension and in a galaxy VERY far from here! Remember that the SOASC= project is an automated process, and can be restarted anytime, anywhere with any sound specification we would ever dream of. There is no more work involved, there is no problem having the computers and C64 record for 90 days straight with an 30 minute break each 3th day to extract the results. I have no further comments.

Q: "Hmm, I'm not really impressed...I can hear static noise in the recordings!"

A: Again, SOASC= is about the AUTHENTIC and GENUINE sound from real C64's. Its the key factor! The SOASC= project records both music played on a 8580 and 6581 SID CHIPS. It records them EXACTLY out of the AUDIO/VIDEO connector on the Commodore 64 itself...so naturally noise will emerge from the chips inside.

Well, popquiz:
Did you really care about the noise back in the days? Did you actually hear it and did you hate the C64 for that? Could you really do something about it? Did it REALLY caused so much of a trouble for you? Do you use the same music equipment to listen to C64 music (MP3 version) TODAY as you did BACK then?

No. You were like the rest of us either connected to a crappy television with horrible speakers, or you did actually have a sound system with AUX input. Now....what do you really remember? The NOISE that WAS there (trust me it was, and even more that in the SOASC= recording!) or the music itself? ...the answer is the music of course, so easy!

Anyway, when doing tests during the recording, naturally noise was detected. But, by improving the internal grounding and also connect the AUDIO IN on the SID chip to GROUND, the noise apparent in the recordings was "completely" removed.

By writing "completely" its actually based on the sound material itself. Further studies during the recording showed that tunes did vary in volume. So lower volume means the "normalize" routine had to boost the final output more resulting in more noise, but if the volume was quite high, the noise would be really totally gone!

We could of course do some software post processing to remove the noise, humm etc...but we'd never know what a typical all over setting would do to THAT particular tune, so that idea was totally ignored too!

Furthermore I DID NOT add the -b option to PSID64 (screen blanking)! There were tips about this reducing noise, yes on the 6581 I noticed so just minorly, but a number of songs were detected to be out of sync completely with this option. (It may be due to defective SID files and was confirmed by HVSC that those tunes I noticed on ACTUALLY had bad bytes in them....typically a lot of the VARIOUS\M-R\Nilsen_Ronny\xx.sid).

But since I had already added a better physical grounding as mentioned above, this option did not have any relevance to the SOASC= project and where rejected.Furthermore in the "changelog.txt" for the PSID64 project, it states: "Implemented screen blanking to improve audio quality when using the RF modulated output of your C64.". It specifically mentions "RF modulated" and nothing about the AUDIO/VIDEO out connector! So confusion is here all right.

Q: "Why not build or buy dedicated newer hardware for the purpose, such as HardSID?"

A: Again, SOASC= is about the AUTHENTIC and GENUINE sound from real C64's. Its the key factor! All the old components and their analogic specifications make up the sound of the C64. Most of the key components around the sound chip is today totally obsolete not to mention the SID itself. You could get 95% close (with hardware) to the real sound of C64, but do I want that? No, I want the 100% real thing and so should others feel too.
Do you think the "AUTHENTIC" word is put into the SOASC= (Stone Oakvalley's Authentic SID Collection) just for fun? Ohh, yeah of course you could canibalize your precious C64 and put all the REAL components on a PCB card and play along, but man... you're destroying a C64 for gods sake, and why bother then when SOASC= MP3's is just what you will get anyway without any PCB contructing and insulting the soul of C64? Use common...no IMPROVED sense!

Q: "But why do you do it? What is the gain? What's the use, its outdated material and old!"

A: For starters, you are missing one word in the end there. It should say; "old school". That is about the things that matter when you grew up, what made you what you are today, your memories and your experiences with the C64 will be back again to say hello with the help of remembering the past and the old school days with crunchy sound and pixels in 8-bit, 16 color. Remember, todays equipment will be old school in 20 years, so its all about remembrance and the joy of "what was". There is no gain, except the result of the real authentic C64 sound:). This was an electronic and inventive challenge that I just HAD to do. Challenges make character, and it is incredible fun to be able to be creative and set no limits, what a boring life if not. Therefore, I just HAD to do it, and also because I have so many good memories from the C64 days that I would like to experience today in other situations and I still get a kick out of it. It's about human exploration and playing with your mind. If you don't understand, you're a zombie and have lost yourself...and that can't be repaired. Sorry.

Anyway, apart from that I do not trust ANYTHING else than the real authentic Commodore 64 playing the SIDS. Everything else is either emulated (by software or hardware) and I do not want to listen to music not intended to be played on anyting else than the original Commodore 64's. A lot of people seem to forget this incredible important detail when discovering the SOASC= project in general. It is just the hunt for the real authentic thing, and I stress that too much. AUTHENTIC SOUND IS ALL!

Q: "What do the future hold for the SOASC= project?"

A: The intial goal of SOASC= was to process the entire HVSC#45 archive, and then continue to update the archive whenever HVSC released a new pack of fixes/new tunes. This is still unchanged as of Feb 2007. Another goal is of course to fix recording errors and issues with the song lengths. This is a bit more tricky and manual work which will take some time to complete. It just has to be fixed when it is discovered, so patience is key.Also remember that this recording project is done entirely by one man, due to the hardware needed to process the tunes!

But there might be a innovative concept beeing presented during 2006: And this involves the possiblity for users to upload SID files to a server, specifiy for instance a new song length and other perferences, add it to recording queue, played and recorded by my automated recording techniques (on both 6581 and 8580), and then the MP3 file beeing presented as is on a different database. Upon manual verification the MP3 file will replace perhaps a bad recording, a wrong songlengths tune or just a plain new tune alltogehter, that are not even in the HVSC collection yet! This will at least let trustable users and friends of the SOASC= project to contribute and help fix the world largest and most authentic Commodore 64 music archive up to a perfect standard! This will without any doubt be a important asset to the SOASC= project....

Q: "Do you guys earn any money on this? And, you know what...you could by adding banners!"

A: Earn money on this project? Are you retarded?

Hosting this kind of massive collection online is really easy...if one cared to do it the right way!
And so we did, no CRAP - just pure, clean, serious and the tunes ready for download. No strings attached, no nags, no ads, no shit. This is a respectful archival project for the SID fans which includes myself and the website should reflect that.

So by bringing in a bunch of Google crap ads, stupid search engines plugins or ANY other kind of third-party webcode/functionalites beyond what I MEAN in my greatest belief would bring my site on a level that would be similar to SO many other archive sites out there is REALLY important. This site should be clean and 100% our design/functionalites only!If something would be added it should benefit the project...like the external embedded link to YouTube movie of the SOASC= project. Just a heads up, that I am VERY aware of what clean and serious concept/projects are all about! Death to banners, commercial ads, google ad crap etc. that DOES NOT benefit the project in question!Earning money on this project (in any way) would be subject to law as we do not own the tunes and are they are of course copyrighted to their respective composers/companies, so there will never be any banners or stuff making us rich on this. NEVER EVER, WE ARE SO CLEVER!

Q: "Are there any world records involved in this project?"

A: Hehe, well not officially recorded by Guiness Records but I guess my Commodore 64 suffered to most resets in the entire history. Each machine will during the entire recording sequence be reset via the userport about 95000 times..Guess that counts for something :) And not to mention the fact that both Commodore 64's will have played about 1450 hours of music during a 4 month period, and doing nuttin else. I guess its also the worlds longest project the Commodore 64 has ever been put through or even suffered through :) It did run for 24/7 with only a 30 minute break (power off) each 5th day. So did the 4 x PC's involved too...but we don't care about those. They are only workhorses, but the Commodore 64 is preciousssssssss...

# SOASC= Project
# HOW WAS IT DONE

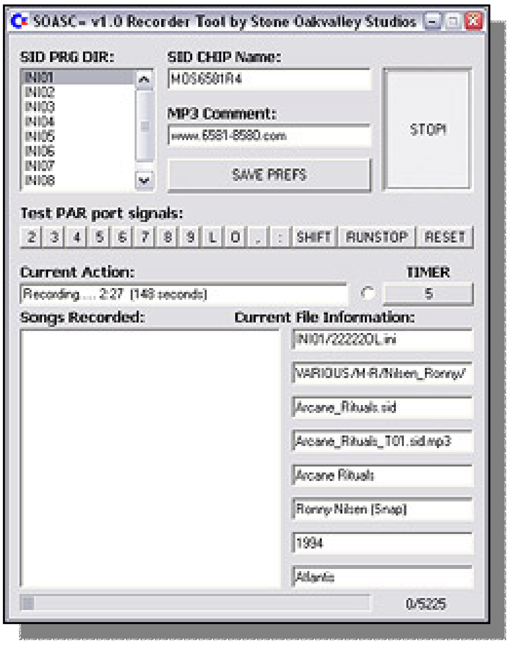
The SOASC= project is an automated recording technique invented by me (Stone Oakvalley) in order to mass record music from the legendary Commodore 64 and its SID chips (6581 and 8580). Realizing this project needed unique hardware solutions and software to back it all up. I spent approx 180 hours on research and how to figure out a plan that would help me automatically record this massive amount of Commodore 64 music. Here is a more or less complete detailed description of all the problems and solutions I encountered.

**INTRODUCTION**
First download the whole HVSC SID Collection and unpack :-)

**PREPARING MADNESS**
I then programmed a tool in PureBasic to interpret and extract out the Path+Filename i.e. "/20CC/Conquestador.sid" and then all the subtunes length from the songslength string: "3:28 3:42 4:39 0:01(G) 0:01(G) 0:09(G)" from the file in the HVSC collection "C64Music/DOCUMENTS/Songlengths.txt" I then made my own database .txt with and looped output like:

20CC/Conquestador.sid
3:28
3:42
4:39
0:01
0:01
0:09
#(Separator)
etc....into a large file containing all song information from the "Songlengths.txt".

**MAPPING THE SID'S**
Then I binary read each .SID file and checked the amount of sub tunes within it. I hacked the SUB-TUNE bit in the SID file to make the SID file start on this tune when played, and then I made a duplicate file of it. (This is because the PSID64 SID Player could not be used to skip tunes, and my system did not have any support to send any additional keys either to the C64. More on this later).Then I extracted out the required information I needed for the SIDREC recorder and when constructing the MP3 tags. I created a own unique .INI file for each SID sub-tune file like this, based on the information from the SID file itself:



SID CHIP Name: MOS6581R4

MP3 Comment: www.6581-8580.com

Current Action: Recording.... 2:27 (148 seconds)

TIMER 5

Current File Information:
INI01/22222OL.ini
VARIOUS/M-R/Nilsen_Ronny/
Arcane_Rituals.sid
Arcane_Rituals_T01.sid.mp3
Arcane Rituals
Ronny Nilsen (Snap)
1994
Atlantis

0/5225

Filename: "0000101.INI" etc etc (More about this filename later)

[SID-DATA]
PATH = 20CC/
FILE = Conquestador.sid
TUNE = 01%3:28 (This means Tune 01 is 3:28 long)
;
[MP3-DATA]
MP3-FILE = Conquestador_T01.sid.mp3 (The new filename to indicate Track 01)
MP3-TITL = Conquestador
MP3-AUTH = Edwin van Santen & Falco Paul
MP3-YEAR = 1991
MP3-COPY = German Design Group

With the above procedure the amount of files of course increased to 46668 .SID files and 46668 .INI files

**FILENAMES**
Then all the files was renamed (both .SID and .INI) into my own charset for usage towards the PAR: Relay C64 keyboard interface. The num of chars used in the filename created in step 3 above needed to be a little bit compressed, due to the fact that there were 50000 unique filenames and I also needed bits in the PAR: ports to send SHIFT and other special C64 keys, including reset and SCROLLLOCK detection for the 64HDD server. Anyway, a filename like "3207101.INI" was renamed to "32O7LOL.INI" Where "1" was replaced by "L" and "0" (zero) was replaced by the letter "O" This renaming caused the recording process to begin not at the top of the alphabet but really in the "VARIOUS/N-R/" directory, which contains approx 23000 tunes. And most of the "VARIOUS" tunes is not really what we all remembered from the good 'ol C64 gaming days! Gimme Rob Hubbard!

**BILL GATES AND HIS "oughta's"**
After the renaming of approx 95000 files was done, I had to convert each .SID file into Commodore 64's

.PRG format by using the excellent PSID64.exe dos software which makes an C64 executable out of a .SID file which can be executed on the real C64 containing player code + textual information from the HVSC tags.

A couple of files was detected as not playing, or had problems being converted during this conversion process. These files will be recorded after the main project is finished using alternative recording methods.Anyway, the amount of files in this project went to a whopping 142134! (.SID, .INI and now .PRG files)

Yeah, working with these amount of files was beginning to slow down Windows XP even. They were later sorted out into 9 different directories just because DOS and WINME on the ServerPC and RecordingPC could not handle this amount of files in one dir. Stupid Bill Gates and his "...oughta be enough for everybody" shit!!! Now, I had to start on the hardware part which was a really painstaking job....

**CABLES AND COMMUNICATION**
Now all the .PRG files was copied over to the ServerPC HD, and by running the 64HDD software which emulates a 1541 Disk Drive for C64 in DOS, the C64 could now access the files and load+run+play them. A own cable was made (XE1541 see pictures) to support the 64HDD and was connected to the Serial Port of the C64 and the PAR: port on the PC. Getting 64HHD up and running was not the most easiest part (also due to PAR: bios setups), and the diodes used in the XE1541 cable was hard to come by. I had to make 2 of everything, and that added some delays to the project.After 1 week of constant testing and configurating it finally worked like a charm! Also, a Audio/Video 5-pin cable had to be made and was connected to the AUDIO/VIDEO connector at the C64.

**KEYBOARD INTERFACE - MAGIC FINGERS TYPIN' ON THE C64'S!**
To be able to load automatically on the C64 by real typing chars, an easy and crude solution had to be constructed. The result was the homebuilt Parallel Relay Card connected to the C64 keyboard connector using a IDE 44pin cable (which fits 100% by the way) The interface consists of 8 relays which are each connected to the PAR: port. By programming again in PureBasic I could switch these relays on and off by command, and thus simulating keys to be pressed on the real C64, letting me load all the different filenames including the keys to LOAD & RUN the .PRG file as well on the C64. Since I had limited with PAR: bits to play with, it was a little bit tricky to optimize what chars I needed to get everything run and record automatically in an long lasting loop for about 13 weeks! The PAR: Relay C64 Keyboard Interface was designed as follows, entry after "=" is the actual C64 key:

PAR 01-BIT01 = 2
PAR 01-BIT02 = 3
PAR 01-BIT03 = 4
PAR 01-BIT04 = 5
PAR 01-BIT05 = 6
PAR 01-BIT06 = 7
PAR 01-BIT07 = 8
PAR 01-BIT08 = 9

PAR 02-BIT01 = L - for loading
PAR 02-BIT02 = O - for loading
PAR 02-BIT03 = , - for specify the C64 DEVICE num
PAR 02-BIT04 = : - : because ":" + SHIFT + RUNSTOP = LOAD & RUN in one go at the C64
PAR 02-BIT05 = SHIFT
PAR 02-BIT06 = RUN STOP
PAR 02-BIT07 = HARD RESET C64 - Userport pin 1 & 3...cold reset
PAR 02-BIT08 = SCROLL LOCK - Off when not loading tune, ON when loading on 64HHD Server

So, by resetting C64, loading, waiting and run a SID tune .PRG file, the PureBasic code was like this:

```
Procedure C64_Load()
; Will write out L+SHIFT+O+"
CallFunctionFast(*out,$378,254) ; KEY L
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,237) ; KEY SHIFT + O
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,239) ; KEY SHIFT ON
Delay(del)
CallFunctionFast(*out,$278,254) ; KEY 2
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)

;Will write out the .prg name

;7 letters to check
For a=1 To 7

If Mid(REC$(counter),a,1)="O"
CallFunctionFast(*out,$378,253) ; KEY O
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="L"
CallFunctionFast(*out,$378,254) ; KEY L
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
```

```
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="2"
CallFunctionFast(*out,$278,254) ; KEY 2
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="3"
CallFunctionFast(*out,$278,253) ; KEY 3
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="4"
CallFunctionFast(*out,$278,251) ; KEY 4
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="5"
CallFunctionFast(*out,$278,247) ; KEY 5
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="6"
CallFunctionFast(*out,$278,239) ; KEY 6
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf


If Mid(REC$(counter),a,1)="7"
CallFunctionFast(*out,$278,223) ; KEY 7
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="8"
CallFunctionFast(*out,$278,191) ; KEY 8
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf

If Mid(REC$(counter),a,1)="9"
CallFunctionFast(*out,$278,127) ; KEY 9
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
EndIf
Next a

;Will write out the extension part
",9:+SHIFT+RUNSTOP = Auto Load and RUN!
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,239) ; KEY SHIFT ON
Delay(del)
CallFunctionFast(*out,$278,254) ; KEY 2
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,251) ; KEY ,
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)
CallFunctionFast(*out,$278,127) ; KEY 9
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,247) ; KEY :
Delay(del)
CallFunctionFast(*out,$378,255) ; OFF
Delay(del)
CallFunctionFast(*out,$378,207) ; KEY RUN STOP + SHIFT = LOADING AND RUN!
Delay(del)
CallFunctionFast(*out,$278,255) ; OFF
CallFunctionFast(*out,$378,255) ; OFF

; Here we wait for a signal from C64 server thorugh the SCROLL LOCK which will light when loading is on, and off when done!

MeasureHiResIntervalStart() ; Start a timer to detect endless loop in case the SCROLLLOCK or 64HDD server crashes!
Repeat
Delay(1)
SetGadgetState(#Load,1)
If MeasureHiResIntervalStop()>120 ; Seconds. If timer reached this limit, then 64HDD server is down or the scrolllock is not working!
MessageRequester("INFO","Timeout: No response from 64HDD / SCROLL LOCK - Check connection! - Please restart")
CloseLibrary(0)
End
EndIf
Until CallFunctionFast(*inp,$279)=126 Or CallFunctionFast(*inp,$279)=255 ; ScrollLock (64 server loading) detection on

MeasureHiResIntervalStop()
Delay(1000)

MeasureHiResIntervalStart() ; Start a timer to detect endless loop in case the SCROLLLOCK or 64HDD server crashes!

;Another loop because 64HDD sends a double scroll lock ON/OFF signal quite fast.
Repeat
Delay(1)
SetGadgetState(#Load,1)
If MeasureHiResIntervalStop()>120 ; Seconds. If timer reached this limit, then 64HDD server is down or the scrolllock is not working!
MessageRequester("INFO","Timeout: No response from 64HDD / SCROLL LOCK - Check connection! - Please restart")
End
EndIf
Until CallFunctionFast(*inp,$279)=126 Or CallFunctionFast(*inp,$279)=255 ; ScrollLock (64 server loading) detection on

MeasureHiResIntervalStop()
SetGadgetState(#Load,0)
EndProcedure
```

Of course there is more code needed, but thats not important, it was just to mess up your reading! Hehe

**THE FIRST SETUP**

During the first 180 hours of research of the project, The first setup with just 1 pcs 8580 Commodore 64. This setup was used for several weeks while designing the software and constructing the hardware. It was really placed in a bad position and was really annoying to have around... well..that's life.

### HARDWARE SETUP - MADNESS MANIFESTS ITSELF

1 x Commodore 64 BreadBox with 6581 SID Chip.
1 x Commodore 64 WhiteyBox with 8580 SID Chip.
2 x Server PC's (33MHz and 233MHz) running in DOS mode with 64HDD as fileserver for the Commodore 64 .prg's.
2 x Recording PC's (800Mhz and 933Mhz) running in WinME with own programmed record software (SIDREC).
2 x 1084 Monitors.
4 x 15 Inch Monitors
2 x Own constructed PAR Relay to Commodore 64 keyboard interface.
2 x Homemade XE1541 Cables.
2 x Homemade Audio/Video Commodore 64 Cables.

12 x 220V Power plug connections needed.

As you can see from the pictures, the hardware setup is a real mess. But if it works, who cares :)
The keyboard and top chassis of the Commodore's are removed.

A own tool called SIDREC (for Windows) was programmed in PureBasic which controls everything and keep track of the recording process. The function of the program is to RESET, LOAD, RECORD, CONVERT, MAKE MP3 (with tags from HVSC .INI files) and go in a loop for about 13+ weeks, 24 hour a day! It also detects when 64HDD has finished loading the .PRG file into the C64, at which point the SIDREC start to record the tune. A ServerPC and a RecordPC was setup (with no cabinets for easier access) and put on a huge table with 6 CRT's and a shitload of power connectors and my god the wires!

Pictures here will tell their own story, you can see all the software running and the 8580 C64 in action.
A fan system (running at 5v) was also made to cool down the SID chip, 2 HD's, C64 power supply and a old crappy 33MHZ SX Processor!

### GETTING THE BOXES

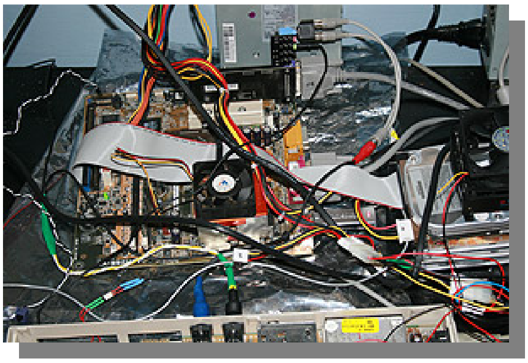
Getting hold of Commodore 64's wasn't as easy you would have thought. I searched some Norwegian net auctions pages, and ended up with a couple of defect C64 breadboxes. My own two C64's had a busted 8580 SID chip and something wrong with the video or char chip. Anyway, during a 2 months time in search for cheap C64's I ended up having 5 pcs 6581 C64's and 4 8580 C64's.. Well, they will make a great addition to my nostalgic showcase cabinet along with the Amiga 500 and Atari 2600 from the 80's together with some old magazines, game box casing and datasettes, disk drives and joysticks..hehe!

### FINAL WORDS

Well, here I am - Stone Oakvalley with the most comprehensive and insane project to this date. Looking on the project I must say the work involved kinda payed off. For who else can claim the throne of producing the most complete, real and authentic Commodore 64 SID music database in the world? From the 19th of November 2006 the SOASC= project will record automatically & process about 1000 tunes each 24 hour session for both the 6581 and 8580 SID models. Expected date of finalizing should be somewhere in March 2007!

### FACTS OF PROJECT:

Based upon HVSC #46 SID database, for both 6581+8580
66000 SID Files
97508 Files
178676 Minutes of music
2978 Hours of music
124 Days (24h) of music
One persistant human
End result? - Well: 97508 MP3 files with a total size of approx 300GB of Commodore 64 SID music.



See you around!
Regards Stone Oakvalley,

# Interview with “Stein Eikesdal” a.k.a Stone Oakvalley The SOASC= Project http://www.6581-8580.com/index.htm

**Q - Please introduce yourself to our reader**

A - My name is Stein Eikesdal a.k.a Stone Oakvalley, 33 years old, single for reasons unknown (okey, I live out on the country in a very small “town”) and I have been working as a Graphical Designer in a maritime display/computer manufacturer company since 2000. I have a very creative personal life filled with composing music, making short movies, music videos, designing crazy hardware, co-writing a 300 page black humor book about Norwegians words with lots of "puns intended" and then the latest SOASC= project.

**Q - What introduced you to Computing?**

A - I was introduced to computing during a Friday evening where my father had borrowed a TV game console at the local gas-station. I guess it was around 1983/84. To this date I can't remember what box that was I only remember a plane going around vertically and rescuing people and flying through cavers. I also remember some kind of base and the letters "FUEL" written somewhere. Later I was naturally hooked and my sister got hold of an Atari 2600 for the family (with PING-PONG, TANKS and the super classic Asteroid cartridges). The crazy thing, I found the Atari console at home during early 2006 and restored it back to life, but my favorite game Asteroids was gone. Well, luckily I got to play some PING-PONG and TANKS!

**Q - What is your first experience of Commodore?**

A - During 1987 or 1988 I finally got the beloved Commodore 64 Breadbox 6581 machine in my possession. I was so amazed about this machine at the age of 13-14. The first games I had was Sorcery and Winter Games on tape. But it was actually more interesting to read the C64 instruction manual at times than playing the games, because there was some kind of peek and pokes which really interested me. So, I started typing in basic programs and really got the hang of it. At school I would draw sprites on the grid patterned paper and writing down basic routines just out of the blue and when I got home type it into the C64 and created animated sprites and even multi colored sprites. I had a special interest of pixels really, kinda cool to see how just boxes of colors could be drawn to form known objects and humans floating around on the screen. And with the low resolution I think I actually started counting pixels by looking at demos and games. Then, other kids at school also got hold of the Commodore 64's and then the everlasting era of loading turbo tapes and watching demos took off in 1988. I was really amazed about the concept of demos. Just music, raster lines, flashing colors and scroll texts just blew me away. The greatest experience came when I heard real sampled drums on the C64. I could not believe it. I must have heard that tune for so many hours day after day. The demo was made by Lukullus and was on my system named "It's drum time!". The music was Dulcedo Cogitationis by Mr. Chris Huelsbeck. To this date, this is my most favorite track on the C64. I just love that tune and the drums was so kick ass. Luckily for me I had the 6581 machine, because if that song was played on the 8580 chip I would not hear the drums so much and it would not make the same impression. And having said that, the 6581 sound is of course my preferred choice of today. Funny how just how one moment of music decide everything for the future. Well, suddenly the Amiga 500 came to town in 1989 and just took my soul. I had the C64 for 6 months after that and sold it to a friend. And funny as it goes, this guy is now living in my neighborhood and he still has the box in the Attic.
Some day....some day I will recapture you - my bread boxed friend!!

And to add to the final conclusion, in 1992 I missed the C64 so much that I acquired not only the 6581, but also a 8580 machine including a 1541 disc drive and started playing around again during 1992-93-94, until Commodore died and I sold the A500 and A1200 in 1997 and bought a yuck 100mhz Pentium PC. But the C64s was never sold; they were just so precious so they were stored in the attic. Amazingly enough readers...those exact machines was used in the ENTIRE process of the SOASC= project! I must have had a evil futuristic and subconscious plan for them..hehe.

The Atari, Nintendo, SEGA and PlayStation etc etc never captured me in any way...for me it was always Commodore!

**Q - Do you still actively use Commodore machines (apart from the SOASC project)**

A - From 1997 until ca. 2000 I was not anywhere near the Commodore 64's. At that time I was composing music on the PC/midi/trackers/with my physical instruments (like TB-303 and Roland JP-8000). The need for hearing and seeing C64 again happened sometime during 2003, and at the end of 2003 I started on a crazy project called SOMAC (Stone Oakvalley's Multi Arcade Console 2000) which was at first intended for MAME games only, but the project just blew out of its bubble and today covers 16 different emulation software and about 26000 games in the database - all with genre specification and title screens along with my own 3D Matrix Core Menu System. And my Arcade Machine is a huge monster...But during 2006 I came over a comparison on emulation vs. hardware recordings of C64 sid music and naturally began the SOASC= project out of that single moment. Now, in the aftermath of the SOASC= project (Mid 2007) I started actually loading up turbo tapes and playing again for real on the C64. It still does the magic for me, and I have played some lovable forgotten games like FireAnt and Lady Tut from the past during my summer vacation 2007. So, I guess I still will be playing on the C64 again and try to pick up more on that in the future. But, naturally the recording of

HVSC updates will also require me to work on C64 for some time to come. And..oh.. yeah, just bought the MMC64 too, that is a impressive piece of hardware and it helps me with those cranky SIDs that could not be recorded automatically!

**Q - Please tell our reader about SOASC what does it stand for and what did you hope to accomplish**

A - SOASC= or SOASC stands for Stone Oakvalley's Authentic SID Collection. The '=' was put in there to recognize the C= brand in some way. I know it's copyrighted, so I do not rely so much on that on the website. Just as a subtle word/graphic reference. I must point out the the "authentic" word really is what the project was about. My goal was to bring forward the music just as it is heard on the C64 if you hooked a unmodified C64 to your telly/sound system today. It is also a step back from other ideas like "pseudo stereo", stereo-sid mix or other non-natural software processing of the sound itself like software noise reduction or any filters done via software to try and clean up or improve the sound material. I wanted to keep it gritty, noisy and of course authentic. The thing is that these gritty, noisy factors are just what you remember (hopefully) on your crappy old tv from the 80's. When doing test recordings the noise was naturally very present and luckily I found a site giving tips about connecting SID IN pin to GND and thus eliminating a lot or maybe even all noise that you could state as irritating to listen to for longer periods. I accepted that modification. The only other processing done was maximizing volume and adjust bias to 0.0 with DOS software like SOX and Normalize.

But also very important for me was to provide these MP3 files on a clean web database interface with no commercial banners or Google ads. I wanted the site to be just for the project and ONLY about the project. I see so much...cr** these days on other typical sites that provides historical information or have databases about games for old computers. I really want to stand away from all that and provide the entire collection just FREE and clean. FREE on the internet today is a totally messed up concept and been so for a decade at least. Its time to put out an example of what FREE used to mean. Get your stuff you seek and get out. So simple. Nothing more. Leave all the bells and whistles to somebody else.

And so it resulted in the sites we have today, both my own 6581-8580.com site and the pathway to the "file holder" site which holds the core of the online web database engine. The interface and background tools were programmed by Svein Engelsgjerd a.k.a Waxhead and he is still the
maintainer and network guru for the SOASC= project.

Another important thing is that I respect the people who have spent so much of their time on composing SIDs on the C64 during all these years. I have always loved the demo-scene community especially (although I never was a part of it). I want to show my respect for all those composers and to bring forward/present their work in a more suitable music format that we can all share and enjoy without having to type LOAD something and wait 30mins for the tape to finish in the worst case scenario :)

Further it was a crazy project that I also enjoyed for my own personal pleasure and the question of..."could I actually pull it off, or have I reached the limit of insanity just now?". Guess not.
And not to forget, both chips had to be recorded. Not, just my personal favorite 6581. There are people out there which only remember the 8580 of Last Ninja for instance. They would not have recognized the 6581 version where for instance on track 9 there is a filter sound present which is not there on the 8580 :) I just had to support those 8580 people as well. Further even two revisions of the 6581 chip will be recorded since there is a clear difference between R4 and R2 for instance. 6581R3 will be ignored as far as I know today.

**Q - So these are automated MP3 tracks recorded live of a REAL machine.**

A - Yes, all SIDs and all subtunes/sounds within the SID files was recorded in a automatic recording loop, which basically goes like; reset C64, type and load, run, play, record, reset, create MP3, update current progress and loop it until the end.

**Q - the mp3`s are recorded in quite a high quality 224Kbps why did you select this.**

A - The MP3 and CBR 224Kbps was chosen because it is in my strongest belief that this is a format suitable for a lot of different hardware and OS's just straight out of the box. DVD players in the living room, car stereo players, old MP3 players from 7 years back and a lot of other peculiar hardware (like MP3 plug-in for the MMC64) should be able to play the MP3's just so. The CBR 224Kbps was chosen because I know that for instance my car stereo player (from 2003) had problems with 320Kbps and also VBR MP3 files. The MP3 was encoded with LAME and the command line: "-c -h -m m -b 224 -q 2 --tg 52 --id3v1-only". Some people will disagree about the 224Kbps and CBR and why not using OGG and FLAC...but for those who say that, I only ask: Can all of your hardware equipment play OGG and FLAC straight out of the box (without any additional plugins) etc?. No, generally not so simple.... but also think of your friends, do they have THAT exact OGG supported MP3 physical player in their pocket when you accidentally give them the classic sound of the C64 for them to listen to? Guess not, ey?

And there you have it, my only point and statement for why choosing CBR 224Kbps MP3. Of course seen from a preservation view WAV should be they only right choice, but the amount of space needed for that is not that simple to deal with....yet.

OGG is claimed to have a better quality of sound vs. file size and really I don't know. I need a OGG plugin for my WinAmp maybe, maybe not…don't really care about OGG or FLAC formats. Only playable on specific software players with the plugins in place, and do I have hardware that support it, like DVD players, MP3 players..I have no idea and do not care. MP3 forvever. And since OGG is a later format and much more less commercialized, OGG was early on decided not to be used. But, I have made a nice experiment on what frequencies are being cut out from the MP3 files on my FAQ page, I guess there are some sound samples for your dog to listen to.

A quick opinion about the CBR (Constant Bit Ratio) too. PS, I'm no super expert on this, but here goes. It's like watching today's TV satellite streams. If you watching fast moving images and especially explosions you will see boxes of graphics shimmering/glimmering while the encoder tries quickly to uncompress the image material. That is the VBR (Variable Bit Ratio) in action. I just don't trust self adjusting real-time compression algorithms...I prefer solid stream of data...even if it takes 200kb for a blank image, I want it. I have worked with video editing, PJEG, MPEG compression on computers for 10 years now, so I've seen the dangerous settings to avoid. Variable Bit Rates scares me, so there.

Or, in our case if the MP3 frame is almost dead silence we would save a few k's of bytes by using VBR instead of CBR. For a song, we would maybe save 300-900kb using VBR? Maybe more, did not really check. Well, since most of us are on broadband today, there should be no argument of why not doing it CBR's way. The 224kbps was chosen just out of the hat. I felt that 320kbps was really overkill, 192 could suffice, but lower would be just a waste of effort. So I kept the quality just a bit higher to make sure that if anybody would use the material for something they would at least get more quality than needed. I did not want to skip it down to the bone and optimize my files. Keep them good enough for some post-processing or whatever people would do to the MP3 files in the aftermath.

I guess the meaning of SOASC= is not that people can have all in one pack...like the SIDS. If people did some investigating they would NEVER manage to listen to all the SIDS anyway. You know...the length of the entire HVSC is 2978 Hours of music! People download their favorite or just picks at random. After a few hundred songs I guess most of them stop and just listen to what they have. Between most people, I really think they would have a problem (after lets say 5-10 songs) to tell any difference between 160, 224 and 320kbps. At first people will try as hard as they can to identify defects in the sound material. But, as logic has it people will soon ignore the quality, get used to the quality and just listen to the music anyway. Their brain and ears would focus on other elements like environments or other thoughts. If another person had only lets say 128kbps tunes on their MP3 while jogging they would in just 5-10 minutes after forget about the quality and their ears would adjust to the material and from there... it does not really matter anymore. They'd enjoy the view and the music would in many cases become subconscious vibes of sound and not something to analyze to death or nag about not choosing OGG or FLAC for the purpose of SOASC=.

**Q- are all the files recorded in mono or do you plan to have a pseudo stereo version ?**

A - No, recorded as intended by both Commodore and 99% of the composers. Plain mono. There will be no other versions recorded or produced.

**Q - Although the website lists how it was all done can you give our reader "a brief how to guide" on how you bulk record from a Commodore**

A - First you need a stand-alone player/software on the C64 that does not require any user input to play songs and change sub tracks. There are some players for the C64 which can play the RSID and PSID files, but it requires you to press return and use arrow keys. I choose PSID64 v0.7/v0.8 because the player routine is included together with the SID itself. PSID64 creates PRG executable files which start to play instantly. But, some of the tracks lock up the C64 so you can choose sub songs that easily and secure. I then had to duplicate the SID file with the START SONG bit set to increment for each track. Then, by loading them one by one, PSID64 would start playing on that particular sub song. I even think there is some kind of WinAMP plugin that sends the SID file to a C64 server program and plays it from there.

And so the playing system was really set up. Now comes the hard part. How do you load and run several PRG in a row. Well, first you need to LOAD a file, RUN it and then wait until the song ends and reset the C64. This can't be done without human interaction. Some specific tailored software with its own loading and resetting system would have to be programmed. Since, I don't know any advanced programming on the C64 this was no option. So, I figured I had to simulate key presses and resetting the C64 without me touching it. I drooled on robotics....but the solution was to use 2 x LPT PAR ports as relay triggers. Send a byte to the PAR: port voila you have a connection between a wire and thus simulating a key press. I made a custom relay card (thanks Waxhead for the schematics) and used a regular IDE cable with its original connecter and jacked that into the C64's keyboard connector on the main board. Since the C64 also have a handy reset functionality on the USER PORT I could reset the C64 also by sending a specific byte to the PAR: port. I had to use two PAR: ports, because I needed to type some filenames, reset C64, pressing shift together with : and compressed version of LOAD....which is L+SHIFT+O as you might remember. The physical "char set" I ended up with was; 2,3,4,5,6,7,8,9,L,O,COMMA,:,SHIFT, RUN STOP, USER PORT RESET and SCROLL LOCK detection for 64HDD.

So, where do the files come from? They came from another PC acting as a fileserver in DOS only. I used the freeware version of 64HDD to set up an simulated 1541 device by connecting a XE1541 cable from the PAR: port and to the SERIAL PORT of the C64. The nice thing about the 64HDD is that it tells you when its loading something. It turns on the SCROLL LOCK LED on the PC while loading (its an command line option) so when finished loading the PRG file it turns the LED off again. I then coupled two wires from my SCROLL LOCK LED on the PC keyboard to my own custom made relay PAR: card to detect whenever the LED was on or off. This was really an important factor of the project and without it I would have to estimate the loading time of each PRG and that could resulted in some damaged recordings.

Anyway, with the 64HDD up and running I could type LOAD"453453",9,1 and it all set to go. Then, to be able to control all this in a fluid automatic loop I created my own software which use my own tailored INI files (with actual SID FILE information) and file naming structure. My software (SIDREC) controls the PAR: ports and then typed everything automatically by sending bytes to the PAR: port. My software also uses information found in the HVSC songlengths.txt to determine when the song has finished recording. At that point the C64 is reset by SIDREC and the recorded WAV are converted to MP3 in the background on Windows. After converting, it loads the next INI file in the queue, extracts information about the filename to load, how long the song is, who composed it etc etc and stores that internally in SIDREC and then spits this information back into the MP3 file during encoding. Kinda strange and a akward way of doing it, but it did work.

**Q - What were all the songs recorded to**

A - Everything was recorded as 44,1Khz, MONO WAV to a 8.4GB and 6.4GB 7200RPM HDs which needed to be emptied out each 3rd day or so for 4 months. After the WAV was recorded it was encoded to MP3 and deleted. The PC hardware used on the project was nearly as old as the C64 themselves. I did not buy or have any large HD's at the time when I started recording. I spent a minimal amount of money on this project. Also, I though It would be a good idea to turn the entire system off (to copy the recorded MP3s) for 30mins each 3rd day to avoid corrupted memory or other issues with the C64 and its chips/power supplies.

**Q - Can all the Mp3 be downloaded for your website**

A - Yes and no. During March, May, June, July 2007 all was okay with our hosting provider...but in August 2007 they could not can handle us anymore. Actually during 2007 two hosting companies have been promoting false advertising about their accounts and storage capabilities and we had to cancel the accounts. We will alert people about this either through our forum or via hosting review companies and try to spread the word about this false advertising. We told them we intended to use the space and they agreed. Then, after some months of a 300GB packed website and download traffics (but far away from what they claim we are allowed) they asked us to leave etc. They offered us the world, and gave us only a straw....and they were all short!

**Q - Have you thought about providing the files on a DVDs - how big are all the songs as mp3`s I read somewhere about 300GB is this accurate**

A - The current collection as per August 2007 is 302.8GB which covers the MOS6581R4 and CSG8580R5 chips. We are planning to record the entire HVSC collection also with the MOS6581R2 version since it has a very good popularity in the community. Our 6581R4 chip has a very strong filter which is sometimes too strong, but this filter factors are very common knowledge amongst the hardcore SID lovers :)Well, I think regular DVDs are out of the question. Maybe in time when the HD-DVD or Blu-Ray discs are more common household material, such a solution might be possible?

**Q - I notice you have a forum and user have asked for a Bit torrent download will this be available.**

A - In fact, I use very rarly Bit Torrent files...for me its just so amazingly slow...if its not popular. So, if that is going to happen or how I have no clue about. As it is today, I'm the only one in the world with the entire collection in one place. When I have fixed some additional errors in the collection and recorded off the HVSC#47 and the MOS6581R2 chip I have some contacts that will be able to get a copy of the entire collection on a 400-500GB HD with the intent of presenting this on a radio stream or more preferable acting as a file mirror.But as it is today during 2007 I will keep the collection being spread minimalistically. I am a perfectionist and I won't give the whole pack away before I am really satisfied with the work.

**Q - You must be a perfectionist if you prefer the real machine other than emulation would you like to comment.**

A - In fact, before June 2006 I was quite happy with listening to emulated SIDs on the PC with like Sidplay2/w. But, the truth is that I was blown away while surfing to a emulation vs hardware recording site which featured the one and only track that ignited the whole SOASC= project. That track was Gloria by Dane and Mitch. Listening to the extreme cool difference in the beginning of the filter sequence I could not believe it. So, I loaded up my real C64 and recorded it myself and there it was. I started doing my own recordings vs emulation on other songs and found out that there is really a good difference between the real thing and emulation. But, of course in many many cases the emulation was just as the real thing..so respect for the work involved in the emulation world. But, I guess the old saying: "nothing beats the real thing" made my day...or months actually!"If emulation was perfect..." - a wise man stated once.

**Q - Will the hardware and software so our reader can DIY the project be available for purchase.**

A - No, I don't think so. My SIDREC software was specifically designed to work against a specific setup and it has a lot to do with the PAR: ports and their addresses and what kind of mainboard you'd have. I had a really difficult time finding old enough PC's to work with the 64HDD and to get the PAR: ports work correctly towards the XE1541 cable. The whole thing is somewhat a cruel mess and if you do not have a copy of my brain...you're in for some headaches and troubles.

**Q - What was the hardest part of the project?**

A - I guess the hardest and most "working against the whole universe" part was to get 64HDD up and running together with the homemade XE1541 cable and a DOS based pc. It was really picky about the main boards, processor and the BIOS setup for the LPT ports. Even the cable and the diodes I had problems finding so I tried some equivalents...nope, did not work too good. I messed with this a whole week before I almost went insane and just had to quit it all...I had no idea that the software could be so tricky to get to work. You just need "that" specific main board with "that" specific command line not to mention "that" specific setup in BIOS. And even, if you'd tried the same thing on another main board it did not work...and that was especially the BIOS/LPT setup part. But, when you finally get it to work I could not have done the project without 64HDD. Love it and its stable as a rock.

**Q - What`s next is there more to perfect on the project**

A - Next in line during 2007 is to setup a dedicated homemade recording rack where all the hardware (that used to lay on a 3 meters long table). It will also be TCP/IP connectivity this time in both DOS and WINXP for all the PCs. I have smacked together almost all the hardware during July 2007 which consists of: 2 x server DOS PC's, 2 x Recording WINXP PCS's, 1 x 6581 machine, 1 x 8580 Machine, 1 x 14 inch display, 2 x 12inch displays with touch screen, 5 x power supplys, 2 x cd-roms, 2 x floppy drives, 3 x PC keyboards, 1 x VGA switch, 1 x keyboard switch, 4 x HD's, 1 x 7inch CRT TV, 2 x homemade relay PAR: cards, 1 x 5 port hub and a lot of cabling! Its really called the FrankenStein rack for now :) The whole thing has been mounted into and onto a old Fisher stereo rack from the 80-90's..you know those with tape player and a record player on top.Other thing is to record the HVSC collection on a

MOS6581R2 chip, just because it has its own filter characteristics which can't be ignored. Other than that I don't know, news will be posted on my site.

**Q - When does the process finish is a HSVC update is issued will you re record the whole thing again or just the updates**

A - I will only record the update and add it to the collection. Recording of HVSC#45 took about 122 days to accomplish. Recording of HVSC#47 took about 4-5 days. My system is able to record about 500 tunes each 24 hour session for one Commodore 64 model. Since I have two setups, basically one for 6581 and 8580, the max tunes to be spit out are 1000.

**Q - Are you able to keep up with updates?**
Well, my system was based on HVSC#45 which was released in April 2006. My system recorded 24/7 from November 2006 through March 2007. In between here came the HVSC#46 during January 2007. Seeing how the HVSC updates was released, I thought I would be finished long time before HVSC#47 hit the market in June 2007. But, I had noted some problems with a couple thousands songs from the HVSC#45 collection and that was the BASIC tunes and some really large SID files which caused some aftermath recording work. A lot of manual work was done on these recordings. During July 2007 I managed to fix 3457 tunes that I had noted, or by others via the forums as bad ones. The only remaining bugs that still are "present" from the #45 release is the issue with NTSC specific tunes. I will have to purchase NTSC models and additionally fix and re-record about 2974 tunes for that cause. As I recorded everything on PAL machines and the fact that PSID64 does not handle NTSC flagged tunes correctly or not at all, I really never got finished with the full HVSC#45 release.And we are now in August 2007 at this point of writing, where I have recorded the 8580 version of HVSC#47 already, the recording rack is of the first priority (check our forum http://www.dirtcellar.net/phpBB3/viewtopic.php?f=8&t=146 ).Further, the HVSC updates are not just recording the update for me, but also own tools have to be created to filter out what other updates the HVSC crew had performed, like; songlengths fixes, copyright fixes, and information stored in the SID. They also sometimes delete files and remove dirs and move files around. All of these changes has to be either manually or half-automatic be performed on my MP3 collection as well. So for each update there is more than just hit record and go :)

Also, the HVSC crew has stated that the there is a current increase of SID being ripped etc, so that also brings the HVSC releases faster out than normally. During 2006 they only had one release, but during 2005 they had 4. In 2007 they already have had 2 releases, but I suspect a HVSC#48 release around Christmas or even before. Anyway, I'm still catching up, so hopefully I can release the updates much faster than during this year. But, as I said, that is because there is a lot of aftermath problems for the initial recordings.

**Q - On the site you can search the tune or composer - can you take our reader through the options**

A- Yeah, the main site you'd start from would be: http://www.6581-8580.com. Then on the top menu, click "Download MP3s". You will enter the database and the first page of "hits". From here you can either click on the insane bulk of numbers which represent pages on the database alphabetically. The options for searching are "Songtitle/Composer/Releaser/Date". I think the pulldown menu explains it all. Hitting search and you'd find your hits. You can download both the 6581 and 8580 versions but also as a bonus the SID file from the HVSC collection as well. This can be handy to quickly find out "that" particular tune is what you seek for. Of course you need some kind of SidPlayer software to play it. Also, the SID can be handy to determine if you found a possible damaged recording or a recording filled with noise or something that clearly differs from the SID file.

You will notice a GREEN "flag" on the "MP3 xxx xxxxxx" buttons to the left in the file table. This means that the tune is made for that specific chip, or is suggested to be the most preferable version to listen to. This flag comes directly from the SID file itself, so it follows HVSC many years of research :). Also, take notice that sometimes there will be no GREEN "flag" at all. That means nobody knows (yet) what kind of chip that tune was specifically composed on/for. In other instances, you will see several GREEN "flags", that should mean that the track sound good on both models of the SID chip. This is mainly to help listeners to find "that" favorite tune to download.

**Q - Can we bulk download for example if I search on "hubbard" Can I download all the Mp3`s that are retured in one go**

A -Not via our database as it is today no. I guess the smart people with Download Managers have already figured out an alternative way:) I have an idea on hold which is the SOASC= Downloader Tool which can be used to mass download songs and automatically create the correct directory structure for you as defined by HVSC initially. Today, the 6581 and 8580 files are named the same, we have just filtered them on our HD's and site as two different dirs with the HVSC file structure intact inside. The tool would be limited in how much you could download during a day, or based on amounts of data downloaded etc, but I do not have the plans ready on how and what to limit. The tool is a very dangerous tool so it has to be limited or else we would probably end up with traffic as high as YouTube or something:)

We thought about enabling FTP too, but the hosting companies will not like this. We have during 2007 only used VERY cheap solutions, so they must be bothered as little as possible or they will...erhh.. HAVE thrown us out!

So, today, there is only a straight forward interface for people to download some tracks...and not go mentally crazy and download everything they see. In our opinions, nobody could be that mad?...or?

**Q - What is the largest mp3 file on the website**

A - Hold on... I need to create a tool for that... Now, yeah, the largest file is 57815168 (57.8mb) MOS6581R4\MUSICIANS\B\Bjoernerud_Sebastian\Psykolog_end_T07.sid.mp3. It seems to be a "mix-up" of tune 1-6 in the SID file the STIL entry states, although I would rather call it a "messed-up-mix". The track lasts for about 34 minutes.

**Q - I think its great to be able to download the Mp3`s you like stick them on a small MP3 player and listen at work/or on the bus etc people ask**

**"what are you listening to" and I can say "sid music" then they just blankly gave at me tapping my foot away**

A - When people catch me they ask to, what is this? Commodore 64 music I tell them, then they smile and think it sounds cool. For those who already know what it is, they say....do you have Commando and ....and...what was the name of that game where you...blah.blah and so it goes:)

**Q - You have a real recording studio "w w w . stone-oakvalley-studios.com" can you tell our readers about this**

A - Its not a real recording studio, really. I used to call myself Demonoid Productions(!DmD!) during my Amiga years and long into the 20's or 00's or whatever we gonna call this decade. The zeros, I guess. Anyway, since I do so much different stuff within editing, graphics, audio and file treatment all my life I decided to call it just Studios and use my real name as a igniter for my new "brand" so to speak. So, I created the Stone Oakvalley Studios name and environment in 2003. Today I do have a fully equipped music studio with just physical musical instruments. I can't stand softsynth, they do no good for my fingers. Like, have you tried using your mouse on a TB-303 software emulator....thought that was cool? Yeah?, well... try the real physical thing and you'll never go back to that lame way of making basslines:). Since I also had a full editing equipment for doing at least DV movies and before that S-VHS movies the Stone Oakvalley Studios name just sounded appropriate. Its also a creative studio where not just movies and music can come to life, but generally all that I like to do and spend my time on. Creative projects for the eye and ear and stuff that people don't do and wouldn't do even if they said they gonna. I wish I could have a proper music studio, but I don't have my own house and the need is not there either, as I have been doing so much else than making music since 2003. I promised myself to get back into the composing music through MIDI and all during 2007, but now....I guess we end up in the next year...as always :). Someday, I will be back on the keyboards and trancing out some cool tracks again. And, oh.. if you are curious about my music, they're all downloadable from my music section on the Stone Oakvalley Studios site. And, if you got the time, check out some of the other projects beside the Commodore 64 project. But...please make sure you have a couple of hours to spare...you gonna need it, it that much to go through:)

**Q - Do you run the studio commercially and have you had any famous people record there**

A - No, just for my own music. If I get famous some day...than at least I can count one.

**Q - I cant find anywhere to donate to the project are you going to accept paypal or similar**

A - Donations....donations. We had some questions about this, but I have stated that no donations will be accepted as it would interfere with my belief in the words FREE and NON-PROFIT. However, serving 300+ of GB does not come without a price we would think. Well, we did have all the files available for download through some really cheap webhosters for some months. The price was so low, I could not believe...and as you may already know.... such a cheap price does have its "small text" written somewhere and the keyword is "SHARED HOSTING" sites which falsely advertise about 300GB webspace and 3000GB of monthly transfer. We took them up on that offer and was used about 10% of this "allowed" traffic during 2 months and that was it. We had to move the files away, they couldn't take it. We did get refunds though. So, what if we accepted dontations? I guess I could pay for a couple of months, but sooner or later it would crumble down. So, our wish is that the other well established sites out there that rely on donations and even possible banners could host the SOASC= collection instead. So that we can stop being business men with the hosting companies. Also, I guess to set up an network for accepting banners etc would require some additional ground work something that I do not wish to do. I just wanna record the music and give it out.Continuing to improve the current sites we have is not really something we want to do. As the sites stand today, they are more than sufficient to provide users with the files they need. I'm also not too interested in having to deal with other people’s money and almost making a side-business out of this and dealing with the donations. There will quite possible be some work involved. I have so many other things to do, and if I one day can't hold the site up, the people who made recent donations would just throw away money and maybe even be disappointed and we don't want that responsibility on our souls.

So, for the time being we will try to provide the files using cheap hosting solutions, while I will record the HVSC#47 and fix the NTSC errors and even record the MOS6581R2. Heck, we might just go offline until I can get all this fixed, why not..I like to have a complete project to pass on to a mirror site. When that happens, we will contact our interested people that would like to provide a mirror or at least make it possible to bring the SOASC= collection through a streaming radio solution.

Regards
Stone Oakvalley Studios
http://www.6581-8580.com
http://www.stone-oakvalley-studios.com

# - Minimig -
# An Amiga in an FPGA

**What is Minimig?**
Minimig stands for Mini Amiga. Minimig is an FPGA-based re-implementation of the original Amiga 500 hardware. In it's current form, Minimig is a single PCB measuring only 12*12cm which makes it the smallest "Amiga" ever made and the first new "Amiga" in almost 14 years! Minimig is available for download as an open-source / open-hardware design under the GNU public license. This page describes the architecture and the inner working of the Minimig. All design files can be downloaded from the download section.

**History**
The idea to make Minimig started around january 2005. The C64DTV had just been released and the Amiga forums were buzzing disccussing the possibility of putting a complete Amiga with games inside a single joystick. Things like ASIC's, FPGA's and VHDL were discussed and being a hardware engineer, they immediately caught my attention. I remember that the discussions ended with the conclusion that it should be possible to put an Amiga in a joystick but that it would be a very difficult task. The first step of such an undertaking would be to reverse engineer the Amiga chipset and get it running inside an FPGA.

The following weeks I discussed this idea with a collaegue who also happened to be an Amiga enthusiast. He did some FPGA programming during his previous job. The more he told me about FPGA's and the more I dug into my old Amiga literature, the more I became convinced that it could indeed be done. And so it started, I learned Verilog, bought an FPGA board and started coding! It took me almost a year to get the Minimig to boot it's first game (which was Lemmings, by the way). It was and is the largest hobby project I have ever started.

The first version of Minimig was built around a Digilent Spartan-3 FPGA starter board, which I expanded with a real 68000, an upgraded vga output and a PIC-controller based floppy emulator. That version can be seen in the picture to the left. Later I moved the design to it's own custom-designed PCB which I called rev1.0. That is the version that is described here.

**Minimig rev1.0 technical description**
The Minimig rev1.0 is built on a single 12*12cm PCB that contains all components to make up a complete Amiga. It has no floppy drive or harddisk. Instead it is equipped with a MMC flash card slot and a microcontroller based floppy emulator. The flash card holds the disk-images which can be "loaded" into the Minimig using a convenient on-screen-display.

The (physical) hardware of the Minimig consists of 4 major parts:

The FPGA
The 68000
The RAM
The PIC controller

**The FPGA**
The FPGA is the heart of the Minimig. The FPGA used is a 400Kgate Spartan-3 by Xilinx. All the other major components (RAM and 68000) connect directly to the FPGA. The FPGA implements the Amiga custom chips Denise, Agnus, Paula and Gary as well as both 8520 CIA's. It also implements a simple version of Amber so that VGA monitors can be connected. Besides this, the FPGA also acts as an automatic joystick-mouse-switcher, a PS2-to-Amiga-keyboard converter, PS2-to-Amiga-mouse converter and as an OSD (on-screen-display) generator. All of these function were not present in the original Amiga, but make life much easier now that we are living in the 21th century! The Spartan-3 is a ram-based FPGA and must be loaded with a "core" upon startup. This is done by the PIC controller described below.

**The 68000**
The 68000 is the Minimig's main processor. The Minimig uses a special version of the 68000: the MC68SEC000. This version runs at 3.3V and is completely static (so it can run at any frequency between 0 and Fmax). This makes it an excellent companion for the Spartan-3 FPGA as there is no need for level-shifting between 3.3V and 5V levels. The MC68SEC000 connects directly to the FPGA.

**The RAM**
The Minimig rev1.0 board contains 2Mbyte of 70ns static ram. The RAM is organised as 2 524288*16 banks. Each bank has seperate enables for the upper and lower byte. The RAM is used to implement the 3 types of memory needed by the Minimig, namely: kickstart rom area, chip ram and (ranger) fast ram. As the Minimig has no kickstart socket, the kickstart image must be loaded upon startup. This is done by the PIC controller described below. Once loaded, writes to that area of the RAM are disabled and the area acts like a read-only memory. The remaining part of the RAM (1.5Mbyte) is divided up between chip and fast ram.
NOTICE: The ST RAM chips used in Minimig rev1.0 are obsolete. If you want to build a Minimig rev1.0, please check the availability of all used parts first. A suitable replacement for the ST type is the ISSI IS62WV51216BLL-55TLI. These chips can still be bought from Digikey (part number 706-1048-ND)

**The PIC controller**
The PIC controller fullfills the role of "bios". It is a single chip 8-bit microcontroller from Microchip. The PIC controller configures the FPGA (by loading a core into it), loads the kickstart image into the kickstart ram area and acts as an Amiga floppy emulator. Thus, the PIC controller really starts the system up as soon as power is applied, hence the "bios"-like function. The Minimig uses a PIC controller type 18LF252/SP.

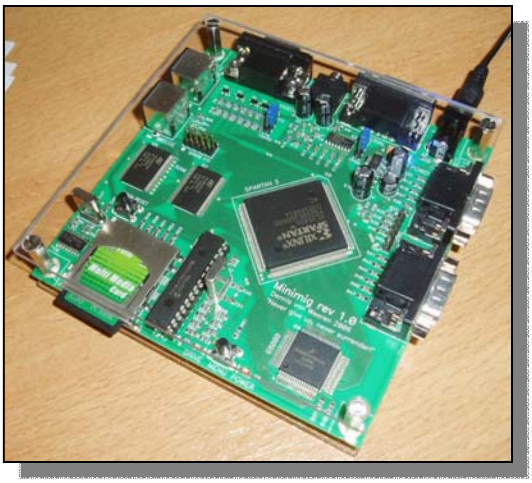
**FPGA general internal architecture**
Besides the physical hardware, there is the "programmed" hardware inside the FPGA. This hardware is described in Verilog. To keep the project manageable I have kept the same organization as a real Amiga. That is, I have kept the Denise functions in a Denise module, Agnus functions in an Agnus module etc. I have even kept a lot af the signal

names the same, so there is a dmal signal (as well as an extra dmas signal), an int2 signal, an ovl signal and so on. Besides these standard modules, there are also 2 bridge modules to connect the FPGA hardware to the RAM and 68000 chips. The code for the FPGA has been synthesized using the free webpack tool V9.1i from Xilinx.

FPGA internal bus structure and clocking scheme

This needs some explanation as it is quite different from a real Amiga 500. Whereas the Amiga 500 had a seperate chipram bus and fastram bus, the Minimig has only a single, synchronous multiplexed bus. To compensate for this, this bus is clocked at 7.09379MHz or twice the speed of an PAL Amiga 500 bus. This clock is the Minimig's main clock (called "clk" in the code). It is the clock far ALL Minimig sub-systems, including the CIA's. As the CIA's are normally run at the so-called E clock (709379Hz), special circuitry has been added to Gary to slow down CPU accesses to the CIA's to approximately E clock speed. Clk is also used as the Minimig's pixel clock. For hires screen-modes clk is "double-pumped", with new pixels put out at both the rising edge and the falling edge of clk giving an effective pixel rate of 14.18758MHz. Besides clk, two other clocks are generated; qclk and vgaclk. Qclk is clk shifted by 90 degrees. Qclk is used by the SRAM bridge to control read/write timing. Vgaclk is used as the pixelclock for the Amber scandoubler module. All clocks are derived from a single 4.433619MHz PAL crystal using the FPGA's DCM module (Digital Clock Manager)

The Minimig internal bus is used as both the chipram bus and fastram bus. All modules (including kickstart area and CIA's) connect to this bus. This bus is time-multiplexed between chipram and fastram/kickstart/CIA's. The mulitplexing is controlled by Agnus and the horizontal pixel counter "horbeam". The lower 2 bits of horbeam define 4 types of bus "slots":



slot 2'b00: fastram (68000)
slot 2'b01: chipram (disk, bitplanes, copper, blitter and 68000)
slot 2'b10: fastram and blitter (non-standard, gives cpu some more cycles in chipram to fix some compability problems)
slot 2'b11: chipram (disk, bitplanes, sprites, audio and 68000)

The Agnus module passes the signals dma,dmapri and dmawr to the Gary module to indicate the type of bus slot. Dma indicates that Agnus is doing a bus cycle (read or write) and dmawr indicates that that cycle is a write cycle. Dmapri indicates that Agnus only holds the bus bus does not write or read it (Agnus does a "dummy cycle). If both dma and dmawr are inactive, the CPU can use the bus if it wants to.

Because the FPGA does not supports internal tri-state busses, all devices are connected together using 'or" gates. The convention is thus as follows; when a device is not selected, it drives it's outputs low. When the device is selected, it drives it's outputs with the data it wants to write.

**Boot sequence**

The boot sequence is a 2-step process. The first step is to configure the FPGA. Like said, this is done by the PIC controller. The second step is to load the Kickstart image. This is done as follows;

Once the FPGA is configured, the system is booted in a special state. In this state, a small bootrom is overlayed at addresss #0. This bootrom loads the kickstart through the floppy emulator. Once the kickstart has been loaded, the bootrom resets the system. The bootrom then disappears from address #0 and the system boots as if it were a normal Amiga. The code from the bootrom is written in 68000 assembly. I have used the freeware AS32 assembler from the Freescale website. I have made it available for download here as I can't find it anymore on their (again...) redesigned website.

PIC controller firmware and FPGA to PIC communication

The PIC's firmware is written in Hi-Tech Ansi C. The firmware contains MMC (Multi Media Card) and FAT16 drivers to control the flash card. The firmware also handles the user-interface and on-screen-display. The PIC communicates with the FPGA through an SPI interface. The MMC card is also connected to this SPI bus. In it's current form, the FPGA has 2 SPI "addresses". Address #0 is selected by the fpga_sel0 signal and control the floppy emulator. fpga_sel1 controls the on-screen-display. fpga_sel2 is currently unused.

**Disclaimer 1: About the Kickstart**

To function, Minimig needs a kickstart rom image. The Kickstart is a copyrighted piece of software. Therefore, you are not allowed to just download it anywhere from the web. You must own the original kickstart roms and make an image of them using the method described in the UAE archives.

**Disclaimer 2: If you decide to built it...**

Do so completely at your own risk. This is not a beginners project.

What does the future hold for Minimig? I don't know. My hope is that due to the GNU public license people will debug it, expand it and generally make it better. What I would like to see first is the implementation of some form of harddisk support, ethernet support and offcourse a debugged sprite engine :-). It would also be nice if the verilog sources of Minimig would make it into a sourceforge project. I could really need some help there. And the rest? Only time will tell!

Reprinted with the Copyright holders permission information taken from
http://home.hetnet.nl/~weeren001/
Commodore Free would like to thank Dennis van Weeren for allowing the reprint of the article

# Commodore Disk Archive Project

**- by Bill Degnan -**

Here are directions for using the MMC64 with RR-Net to make backups of Commodore 64/128 disk libary. See below for a link to get most of the files you'll need.

**NOTE:**
You will need to turn the C64 off and on after each successful image extraction. I am looking for a way to avoid this, so far nothing I have tried works. For this reason, it might be best to use a C128.

1. Purchase a MMC64 and RR-Net from http://protovision-online.de (In Germany).

The MMC64 fits into the cartridge slot of the C64 (or 128). The RR-Net attaches to the MMC64. It has an ethernet jack.

2. Format a SD-memory card, FAT 16 or 32.

3. Download "Warp Copy." The software contains 2 components. One is WARPCOPY06.prg which is to be run on the Commodore c64. The other component is warpcopy.exe which is to be run on a modern Windows PC. Together they work to perform a special kind of TCPIP-like network. I moved WARPCOPY06.prg to the SD card. I installed the PC version of the warpcopy program. You may want to add warpcopy to your firewall rules, allow data to pass through your ethernet card.

3.5 Get a copy of MMC64_Recovery_V110.zip. With this file you can perform a MMC64 Bios upgrade to V1.10. You will need this to make the TCPIP connection and to extract D64 files using the card. Unzip and install recovery.prg on your SD memory card.

At this point you should have warpcopy06.prg and recovery.prg on the SD card.

4. Set up your C-64 with a 1541 disk drive. Test everything to make sure your drive and cables work, etc. Insert a known-working diskette with a program(s) on it.

5. Carefully attach the MMC64 with RR-Net and SD card installed into the cartridge slot of your C64. Carefully connect an ethernet cable to the RR-Net jack, and connect the other end of the cable to your PC. I have two ethernet cards in my PC to allow me to leave the systems connected indefinitely.

6. Activate the PC software. Change the IP address from 192.168.0.64 to 192.168.0.101 and hit enter. This is necessary for Windows XP because IP .64 is not available. Experiment for yourself.

7. Turn on the C64. If everything is working two things will happen

1) The RR-Net's right-hand light will shine, the left-hand light with blip every 3 seconds or so. If it does not, this is a clue that your network connection may not be active. I went into the "Network Connections" and turned on "allow other internet users to connect through this internet connection" in the advanced properties section. You should notice that your "Local area connection" icon will have activated if you've been successful.

2) The MMC64 Interface Bios 1.0 screen will show.

8. From the interface menu, select F1 - Start Filebrowser. Use the menus to locate and run recovery.prg. This program will upgrade your bios to 1.1. There has to be an easier way, but I can't find one. There in an bios upgrade file available, but I could not get it to work.

9. Resart the C64/1541 drive. Enter the MMC64 filebrowser again, this time start up WARPCOPY06.prg.

10. Once in the program, hit N to change the IP address. Use Inst/Del to backspace over the "64" and replace with 101 so that your IP is 192.168.0.101, (port 6644) just like your PC's Warpcopy is using on the other end. In theory at this point you have created the connection between the two systems.

11. Verify that there is a commodore disk in the 1541. Using Warpcopy on the PC click on the "directory" button. The program should return a diretory of the diskette. The time to read a disk is about 5 seconds.

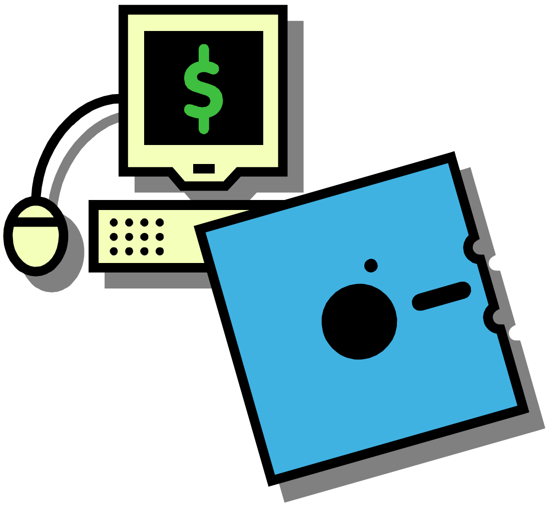
You may wish to take a few minutes to reflect to yourself of the possibilities!

12. Lastly, click on the Read Image button. Warpcopy will ask for a filename, and then perform a disk image extraction from the diskette in the 1541 drive to the destination file on your PC in about 20 seconds!

Here are some useful files, plus the start of what will be a massive D64 library, more on that later.

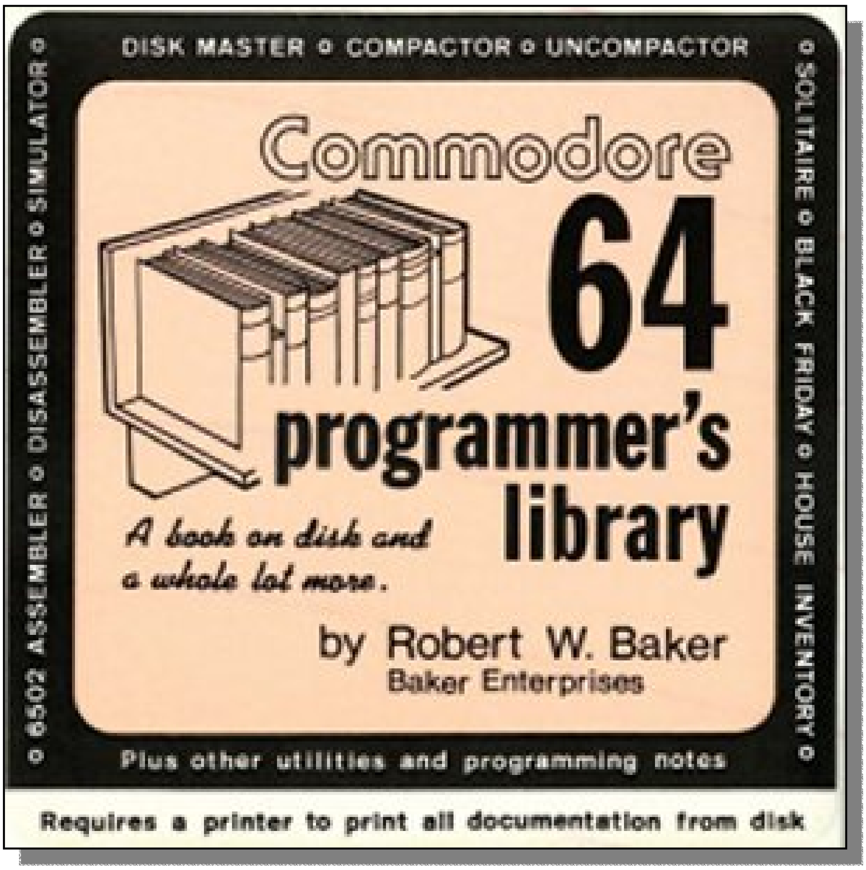
http://www.vintagecomputer.net/commodore/64/

Article Copyright to Bill Degnan
Commodore Free would like to thank Bill Degnan for permission to reprint this article



# The Commodore-64 Programmer's Library

**Robert W. Baker**
http://home.comcast.net/~c64proglib/


DISK MASTER ○ COMPACTOR ○ UNCOMPACTOR
○ SOLITAIRE ○ BLACK FRIDAY ○ HOUSE INVENTORY ○
○ 6502 ASSEMBLER ○ DISASSEMBLER ○ SIMULATOR ○
Commodore
64
programmer's
library
A book on disk and
a whole lot more.
by Robert W. Baker
Baker Enterprises
Plus other utilities and programming notes
Requires a printer to print all documentation from disk

The Commodore-64 Programmer's Library was originally meant to be published as a book by Wayne Green Publishing, publishers of Kilobaud Magazine -- who published my PETpourri column in the very early Commodore years.

Just before going to print, after all the proof reading and editing were finalized the publisher was bought by another company and the book was cancelled.

A close friend of mine, Jim Oldfield, who published the Midnite Gazette for Commodore users, help me self publish the material on floppy disks through a distributor in IL for a few years. Eventually interest diminished and the package was taken off the market.

Just recently I found the original Proof copies for the book and the accompanying floppy disks, and decided to try and make the information available for anyone interested. It's now available from my simple website or on eBay for a very modest fee to help cover my internet costs and maintain ownership of the material.

Besides the http://home.comcast.net/~c64proglib/ website you may also want to look at my RBakerPC blog at http://rbakerpc.blogspot.com/ that documents much of my writings and various activities throughout the computer industry.Besides all the writing, I also managed an extensive newswire service on QuantumLink, AOL and Delphi that was also syndicated on BBS's nationwide. Plus I created a computer industry database that was published for years on AOL, on CD from EMS and in print by No Starch Press. And Omnigraphics used my data to help publish their own computer industry reference books for libraries.

**The Package**
This package includes many of my early original magazine articles, programs and programming ideas for the Commodore-64 that appeared in various Commodore related magazines of the 1970's and 1980's when I was writing the PETpourri column for Kilobaud Microcomputing magazine. A fair amount of the material was originally written for the Commodore PET and CBM systems, then rewritten and updated for the VIC-20, Commodore-64 and Commodore-128 systems.

The 10MB C64ProgLib.zip file contains a number of Adobe Acrobat (.pdf) files that can be easily viewed and printed, either directly from this website or after downloading to your local hard drive. Program listings are included but actual loadable programs are not included in this file.

The 273KB C64ProgLibD64.zip file contains standard D64 disk image files for each of the three 1541 floppy disks distributed in the original package. Two disks provide electronic copies of all the documentation files included in the first zip archive while the third disk provides loadable copies of all the actual program files.

Note that you will need a password to access the actual files included in the archives and those passwords will be emailed to you after payment is received.

All material Copyright (c) 1984 -- Robert W. Baker

For more information about me personally, my past writing accomplishments and other activities, please check my blog at www.rbakerpc.blogspot.com

# The Commodore-64 Programmer's Library - DISK IMAGE PACKAGE

Note: This particular Zip archive includes standard D64 disk image files for the three 1541 floppy disks distributed in the original Commodore-64 Programmer's Library package. Two of the disks provide copies of all the printed documentation files while the third disk provides all the loadable program files plus sample data files There are also three text files included that provide copies of the directories of these three floppy disk images plus a copy of the original package label in another Adobe Acrobat (.pdf) file as well. Additional Adobe Acrobat (.pdf) files providing scanned copies of al lthe printed documentation files plus program listings are provided in a separate Zip archive.

This package includes many of my early original magazine articles, programs and programming ideas for the Commodore-64 that appeared in various Commodore related magazines of the 1970's and 1980's when I was writing the PETpourri column for Kilobaud Microcomputing magazine. A fair amount of the material was originally written for the Commodore PET and CBM systems, then rewritten and updated for the VIC-20 and Commodore-64 systems. This is a collection of many different ideas written over several years so there is no specific theme to this package. However, I tried to organize the material into useful sections that hopefully make it easierto digest. This package was originally written to be published as a book but the deal fell through at the last minute. The package was eventually distributed in electronic form on diskette by a publisher in Illinois for several years. By the way, a copy of the original cover label is included in a separate Adobe Acrobat (.pdf) file in this package.

Just recently I came across the original printed manuscript for this package and decided to scan the material into an Acrobat file and make it available again to anyone interested. Those files are available in a separate download file. This package includes D64 disk image files for each of the three original 1541 format floppy disks. Two disks provide the original text files for the book while the third file  provides all the original program files in loadable form plus a simple utility to print the text files for the book.

**brief overview of material included in this package:**

**Common Sense Programming**
- Saving Space
- Saving Time
- Programming Style
- Debugging Hints

**Getting Into Basic**
- Basic Quirks
- Poking Around in Basic
- Computed GOTOs

**Tape Quickies**
- Hex File Dump Utility
- Data File Copy Utility
- Tape File Notes

**Disk Basics**
- Disk Command Conversions
- Disk Hint
- The OPEN Command

**Disk Internals**
- Disk Programming Tips
- Symbol List – Variable

**Crossreference**
- GOTO/GOSUB Crossreference
- Disk Master - Disk Cataloging Utility

**From Basic to Assembly Language**
- Getting Started in Assembly Language Programming
- Disassembler Program
- DASM - Editor/Assembler

**Programs**
- Data Builder Utility
- The CHRGET Routine
- Intermixing Basic & Machine

**Language**
- 6502 Simulator Program

**Miscellaneous Programs**
- Program Finder
- House Inventory
- Date Book
- Time Billing
- Solitaire
- Black Friday - Stock Market Game
- Easy Script Printer Utility
- Easy Script Word Counter

**Miscellaneous Info**
- The PET Emulator Program
- The Commodore-64 CIA Chip
- Using Zenith TVs

**Additional material later added to the original collection:**

**Compactor** - Basic program compactor utility
- Uncompactor - reverse utility to uncompact

**Basic program**
- Word Pro Disk/Tape Print Utilities
- Finance - simple financial calculator

**The following loadable program files are included:**

- Doc Printer (prints the book text files)
- Disk Master
- Compactor
- Uncompactor
- Basic Program Symbol List
- Basic Program GOTO Crossreference
- Hex Dump
- DASM Editor
- DASM Assembler
- ** Sample data files for DASM
- Machine Language Disassembler
- 6502 Simulator
- Data Builder
- Tape-to-Disk Copy
- Disk-to-Tape Copy
- Tape Reader
- Program Data
- Program Search
- WordPro Word Counter
- Easy Script Word Counter
- WordPro Source File Printer
- Easy Script Source File Printer
- Finance
- House Inventory
- Date Book
- Time Billing
- Solitaire
- Black Friday

**Also included in this package:**

- Three text files with the disk directories
- Image of Original package label (.pdf)

# SID STILLS SOUNDS SO SPECIAL

**By Andrew Fisher**

Note: this is an updated version for “Commodore Free Magazine” of an article first published in the UCUGA Commodore Digest in 2003

Inside every Commodore 64 (and 128) is a special chip. The Sound Interface Device or 6581 chip (later to be replaced by the 8580) gave it the best sound of any 8-bit machine, and has left a legacy of music and sound that is still important today.

WIRED FOR SOUND
First of all, it is important to look at the specifications of the chip created by talented Commodore engineer Bob Yannes. Three separate channels of sound can play at once, giving depth and range. Each channel’s envelope (shape of the sound) and waveform (structure of the sound) can be set independently – or, through the synchronisation and ring modulation registers, work together to create a new type of sound. There is also another feature, the filter, which can dramatically alter the tone and resonance of a sound. At the time it was only an option on expensive synthesizers or through sound editing, but as we shall see there are drawbacks to its use.

Finally, it is important to talk about sound output. From the start SID had an advantage – the audio/video port on the C64 allows connection to exterior audio equipment or a monitor for better quality output. That same port also allows an exterior INPUT into the SID chip, which was uncommon at the time.

SCALES
The earliest form of SID music was a BASIC program. Notes were stored as DATA, read into an array and played back. Many of these programs used a frequency table to generate output, relying on the mathematical properties of sound. This also meant the data could be entered in something approaching musical notation – a note could be stored as C4, read back by the program and converted into the numeric values that SID needed. There were many programs like this, transcribing famous music such as Rhapsody in Blue or classical works.

Commodore also worked on peripherals that allowed the user to make music – the earliest being the Music Maker keyboard overlay. This plastic frame fitted over the computer keyboard and looked like a piano keyboard, pressing down the appropriate key underneath. Later came the play-along albums, and the more advanced Sound Studio. The full-sized keyboard that came with the Music Expansion System (promoted by famous keyboard player Rick Wakeman) works with the Sound Expander cartridge and its built-in Yamaha FM chip. While the software was limited, the output was of a very high quality for the time.

EFFECTS
Another area the SID chip excelled in was generating sound effects. Early arcade games had relatively simple beeps, which developed into more realistic noises by the time the C64 came along. The C64 could replicate them perfectly, and even improve on them. By combining different voices complicated sounds like sirens and ticking clocks were possible. One pertinent example is the game CHAMELEON by Martin Walker. As you play you will hear dripping water, roaring fires, ticking clocks and many other clever sounds.

To start with, many programmers did everything – music, code, graphics. Then a few talented individuals became famous for their music, and were employed solely on that basis. Years later, their names are still important – people like Rob Hubbard, Martin Galway, David Whittaker and the late (and sadly missed) Richard Joseph.

REGISTERS
Of course, controlling SID was never easy. With 30 different registers (including the read-only settings) and many of them doing multiple tasks, there had to be an easier way.

Machine code and the design of the Commodore 64 offer an excellent way to reproduce music. Using the raster register, SID can be updated as fast as the TV screen (50 or 60Hz depending on the broadcast standard). So, the earliest musicians wrote their own “player” routines, entering the note data in an assembler and then playing it back, each frame of the screen update changing SID values. More complex sounds could be generated, leading to better tunes.

The next step forward was a dedicated editor that allowed you to use the music it produced in your own programs. Among the earliest was ELECTROSOUND, which changed the jargon. Instead of voices, we now had 3 “tracks”, just like a recording studio. Each tune was built up of “sequences”, allowing you to repeat sections of the tune. The only drawback was that it used large amounts of memory.

When the demo scene started in Europe, there was a need for more music. The fledgling Compunet network saw lots of demos featuring music “ripped” from games, but now people wanted to start composing their own. Along came utilities like FUTURE COMPOSER and JCH EDITOR, allowing more people to write their own tunes – and to cover “real” music.

SIDPLAYER
Craig Chamberlain was also an important name for SID; in a book on programming from Compute’s Gazette he published the SIDPlayer format and the editor. Now it became easier for American music fans to cover their favourite songs or compose their own music.

Over time the format and the accompanying player application developed. At first you could read information on the tune and see the notes on a piano keyboard. Later players added options to see an accompanying picture, read the lyrics as the song played, and even hear the tune in stereo (with the addition of a second SID chip – more on that later). SIDPlayer music is still used by the Loadstar disk magazine, and a large Internet archive of the tunes exists at www.c64music.co.uk.

FREE SPEECH
Another interesting development in the early years of the C64 was speech synthesis. Commodore's Magic Voice and the Currah Speech cartridge plugged into the expansion port and the audio/video port (remember I mentioned the external input?) to allow programs to create speech. Words and phrases are broken down into phonemes, short groups of sound, and "spoken" by an artificial voice.
The next step was software synthesis, with games like IMPOSSIBLE MISSION and BEACH HEAD II reciting memorable phrases such as "Another visitor" and "Medic! I'm hit" There was also the fantastic GHOSTBUSTERS game by David Crane of Activision, which had several speech samples – and a sing-along version of the Ray Parker Jr. theme tune that displayed the lyrics onscreen with a bouncing ball…

Then another technique became available. The new digital music format of compact discs gave programmers the idea to break sounds down into bits of data. A rapid series of clicks and silences can then re-create sounds. This is sampling, and the Commodore can do it too. Commodore's own Sound Sampler, the high-quality Microvox unit and the Datel Sampler all allowed you to use a microphone or line input to sample sounds into memory and play them back. Typically only a few seconds could be captured due to the limited memory of the C64.

ONE, TWO, THREE
Of course, someone took it a step further. How about playing digitised sounds at the same time as SID music? Games that talk like I, BALL and SLIMEY'S MINE have a lot of atmosphere generated from their funny samples. Then along came ROCKMONITOR, giving a fourth "track" for digitised sounds to play alongside your SID composition. And the conversion of arcade smash hit TURBO OUT RUN has two amazing tunes with digitised sounds, ranging from speech samples to scratching records and all sorts of percussion. It was created by the Maniacs of Noise, famous for their demo tunes and later work on many hit games.

BACKGROUND MUSIC
At first, demos just played the music. Later routines allowed the music to fade in or out, so the demo faded in or out with it. Timing effects to the music was also popular, from a simple graphic equaliser flashing in time with the 3 channels, then on to larger movements of whole pictures. The game DELTA also added a new phenomenon – MIX-E-LOAD, a clever piece of software that allowed you to mix drum & music patterns while the main program loaded.

With the invention of the IRQ-loader, music could carry on while something was loading from disk. That also lead to the development of the trackmo (track-demo) with its continuous effects. Epic pieces of music lasting many minutes were required, often in the techno style.

MIDI
While the C64 did not contain a MIDI port like the Atari ST, interfaces soon appeared to take advantage. Programs like the Advanced Music Studio can playback or record from a MIDI keyboard, while more advanced software from companies like Steinberg turned the C64 into a basic recording studio. One drawback is that there is more than one type of interface available, and they are not all compatible with each other or all the software. One of the rarer items to look out for is the Moog Sound Producer, which came with its own software and no less 4 MIDI OUT ports.

THE SAME, BUT DIFFERENT
In 1987, Commodore introduced a new model – the Commodore 64C. As well as changing the outer case (from the classic breadbox style to a sleeker off-white) there were differences inside. Most dramatic was the SID chip – no longer the 6581 SID but a newer 8580 model. There were also changes on the board, meaning you cannot put an old SID into a C64C and vice versa. Even the manual was different, missing out the sync and ring mod features.

Most dramatic was the effect on samples. The new chip was "quieter", in that it suppressed voltage "noise" better. This meant that samples sounded very faint on the new chip, so you had to turn up the volume on your monitor/TV to hear them. (There are ways round this, including a technique called "digiboost"). The Commodore 128 also has a similar problem, in that accessing multiple waveforms on the same voice can cause the channel to lock or sound at a very low volume.

FILTERS
This also brings us nicely to a discussion on filters. A filter is a way of altering the sound passed through it, and the Commodore 64 has four types. LOW PASS means anything below the set frequency is unaltered, above it is filtered. HIGH PASS does the opposite. BAND PASS accentuates a narrow range of frequencies, while combining LOW and HIGH PASS creates a NOTCH filter, passing only a narrow range of frequencies (the opposite of a BAND PASS). The RESONANCE level affects the strength of the filter.

The trouble is, the filters are different on every machine. As they are analogue filters rather than digital, their effectiveness changes as the chip heats up. Commodore's official line was that the chips could vary by as much as 20% between machines and that it was an acceptable margin to work within. Some games and demos allow you to alter the filter, or they detect which model of SID chip is being used and alter the soundtrack accordingly.

EDITORS
In time, the editors became more complex with more commands. These included the trackers like VOICETRACKER, the Dutch USA Team's MUSIC ASSEMBLER and the famous DEMO MUSIC CREATOR (or DMC for short). And as musicians stretched them, new and updated versions of these tools would regularly appear.

The language of all the editors is similar. You create PRESETS or INSTRUMENTS or SOUNDS – these are the collections of settings for the SID registers. Each of the 3 TRACKS is built up of SEQUENCES, which can be repeated or transposed. Each SEQUENCE is a series of COMMANDS (e.g. SND00 to change to SOUND 00, DUR08 for a note of duration 8 beats) and NOTES (D#4, C-4 and so on).

Even today, new editors are being written. Recent additions include the PC based NINJA TRACKER and the excellent SID DUZZ IT (or SDI for short).

LIFE AFTER DEATH
When the commercial life of the Commodore 64 came to a close, the music lived on. First came a clever Amiga demo featuring converted tunes. Then came SIDPlay, a utility for playing individual tunes. This was later ported to many different operating

systems including PC, Macintosh and UNIX. Alternatives include the SIDAmp plug-in for Winamp, and Deliplayer. Websites appeared devoted to the music, the composers and keeping the spirit of SID alive. Among the most successful is the HIGH VOLTAGE SID COLLECTION at www.hvsc.c64.org, which now contains over 30,000 SID files from games and demos. Working alongside the collection is the SID TUNE INFORMATION LIST (STIL), telling you more about each tune, the composer and identifying which tunes are covers.

There are also many fascinating bits of trivia contained in the STIL, contributed by the composers themselves or the dedicated HVSC team, and most SID players will display the text as a tune plays. And if you want to see what the composers look like, check out composers.c64.org for Peter Sanden's archive of photographs and choice of the best tunes. The HVSC also links up with the incredible Gamebase collection, allowing you to play a SID from a game and see the picture of the composer.

There are also devoted fans remixing and re-making their favourite SID tunes using new technology and musical equipment. From straightforward covers to modern-sounding dance mixes with vocals to orchestral interpretations, there are remixes of every sort. REMIX64 ( www.remix64.com ) and RKO ( remix.kwed.org , maintained by Jan Lund Thomsen ) continue to be the showcase for amazing new takes on old SID tunes. (The recent controversy over producer Timbaland's use of a sampled C64 tune in a song for Nelly Furtado is an example of how the scene stays together, and how the "out of date" machine still influences music).

You can even buy remix CD's, mainly thanks to the efforts of Chris Abbott. Chris used his experience in studio work to create BACK IN TIME, the first ever remix CD. More have followed, along with CD's published by Chris for other artists. Reyn Ouwehand, a famous composer himself, tackled Martin Galway's famous tunes and more recently released his new album "The Blithe, The Blend & The Bizarre". Instant Remedy made a CD of dance mixes, which sounds like a commercial dance album. Press Play on Tape, a group from Denmark, play SID music on guitars, keyboard and drums.

The REMIX64 team created a 1980's themed CD (volume 1), remixing famous game tunes in the style of 80's composers like Vangelis and the Art of Noise, then followed it up with the orchestral splendour of volume 2 subtitled Into Eternity. The revamped C64 Audio website at www.c64audio.com sells many of these CD's, along with digital downloads and bonus tracks for those who make a purchase.

One of the more unusual CD's was PROJECT GALWAY. This 2-CD set gathers together every tune created by Martin Galway, but it is not a remix album. Instead every track is a recording of the original tune playing on Martin's own SID chip. As an added bonus there are some previously unheard tracks, such as the soundtrack to the unfinished STREET HAWK game by Ocean, recovered from the original source disks. The Digital Memories DVD contains footage of many classic demos, along with a separate audio jukebox.

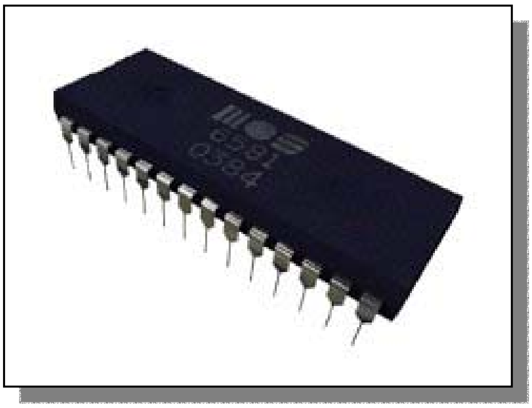
And there's more. BACK IN TIME LIVE has been a series of events aimed at launching the new remix CD's AND getting together the fans and composers. The first two events saw DJ's mixing together SID dance music, the third had live performances from groups like Press Play on Tape, which included Ben Daglish joining them on flute for a rendition of his tunes. Famous composers like Martin Galway and Rob Hubbard, who both now work in the USA, flew back especially for the events. Heroes like Jon Hare of Sensible Software and Jeff Minter mingled with fans. The events also went international, with Back in Time Live Germany and 2005's Retro Concert in Copenhagen. The latest event in London saw Reyn Ouwehand put on a masterful performance of "live remixing" as he played multiple instruments.

MORE, MORE, MORE
SID music does not stand still. Multiple speed players, where the sounds and notes are updated more than once a frame, allow better quality. Hardware experts worked out how to add a second SID chip using a different area of memory, giving you six voices instead of 3. This also led to the Stereo SID cartridge from CMD, which is unfortunately no longer available. But that didn't stop the development.

The HardSID and QuattroSID boards for PC allow perfect reproduction of SID sounds through an emulator or SIDPlay. The SIDStation allows you to compose music via MIDI, and VSTi plug-ins like the QuadraSID recreates the 8-bit sound.

CMD's SuperCPU added another dimension to sound, with its DMA capabilities. With extra memory and speed the use of samples becomes easier and faster, and the amount that can be sampled increased. The game METAL DUST released through Protovision proves what is possible – digitised music plus speech all playing while large amounts of graphics are moved around onscreen, thanks to the power of the SuperCPU's 20MHz processor. With an IDE64 interface it is even possible to stream music from a CD and output it through SID.

LIFE GOES ON…
As long as people are producing demos, games and diskmags for the C64, there will be musicians making music. As long as demo parties continue their competitions for the best music, there will be people trying to do something new and interesting with SID. As long as the remix community keeps expanding and broadening the horizons, people will listen to the tunes and say, "I remember that". As long as the Internet survives, there will be digital data that can be converted into the melodies and sounds of SID. It's a comforting thought.



# Hexfiles Part 7
# http://www.oldschool-gaming.com/

**By Jason Kelk**

Okay, so what happened last time?

Ah, yes I remember, I was going to explain $D018 does, wasn't I?

Okay, well while I'm here I might as well explain some of the rest of the video chip at the same time. First off, there are some conventions that we'll use use here, the bits of a byte are referred to as 0 for the lowest bit through to 7 for the highest.

Okay, so as promised lets start with $D018, which is a pointer to where the screen and character set are stored. And before I go on, a little lesson in architecture is needed. The C64 has, as we know, 64K of memory but the VIC-II chip isn't able to use it all at once, instead it uses one of four "banks" of 16K, bank 0 is the lowest, from the start of memory to $3FFF (and the default, so for now we will keep working in it) whilst bank 3 is the last, at $C000 to $FFFF.

A C64 character set contains 256 characters, so with eight bytes per character thats a total of 2K needed and it's possible to point at any 2K block in the present bank. Bits 1, 2 and 3 of $D018 control this, allowing the VIC-II to see any one of the eight possible starting points for the character data and setting the bits to $0 will put the character set at $0000, using $2 will mean the font is at $0800 all the way up (in steps of two) until we get to $E, which means $3800 has the font data.

Similarly, the C64 needs 1K for a screen and bits 4 to 7 of $D018 are used to represent where the screen is living. Since there are sixteen possible screens in each bankading from, $0 will put it down at $0000 and $F locates it at $3C00. The actual default for $D018 is $14, the upper four bits are set to $1 so the screen is at $0400 and the lower four are $4 so the fifth character set in is used, at $1000. The reason the C64 defaults like this is because a "shadow" copy of the ROM character set is stored at $1000 with the lower case version at $1800.

A quick definition of shadowing is in order, I think. The C64 only has 64K of RAM but the ROM takes another 32K. Because the 6510 can't access more than 64K at any one time (it can only look at $0000 to $FFFF) the ROM has to exist within that space. But that would take up valuable memory from programs.

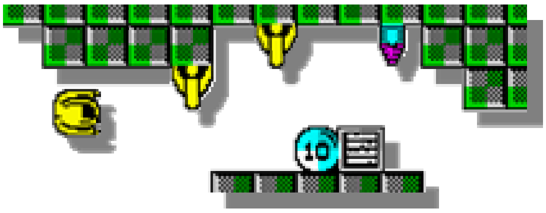
So the ROMs are all shadowed over other memory, the BASIC ROM is at $A000 to $C000 and if you try reading through that memory with the PEEK command you'll be able to see it. But it's *also* possible to store data under that memory, for example graphics, and not have the ROM overwrite it and if we point the VIC-II chip at it we don't see the ROM data, just our graphics.

This means that we get all the ROMs we need and *still* have the space they occupy in the RAM, they're there but in an insubstantial way, hence the name ROM shadow. The character set at $1000 works the other way around, if we point the VIC-II at it we only see the ROM font, no matter what data we put there. But we can happily put data into that RAM, read it out or run program code from there without that font affecting what we're doing (in fact, $1000 is a very commonly used area of memory for music routines for this reason, a convenient 4K block for two fonts that can't be used for graphics).

Another useful location is $D011 and it has a number of uses, bits 0, 1 and 2 are used to set the vertical smooth

scrolling and bit 3 controls the scroll masks, when this bit *isn't* set, the border extends into the screen area slightly and masks hides the edges of the screen so that the new data coming on can't be seen until it's ready. Bit 4 controls the screen, if it's set the screen is on, otherwise the border covers it. Turning the screen off may seem a little pointless, but there are purposes, for example hiding the screen until the graphics have been set up.

Bit 5 of $D011 will activate bitmap screen mode if set, instead of the normal 256 character set the VIC chip takes a block of 8,000 bytes, 8 for each character square of the screen, so really it's just like having a *very* large character set. Bitmap mode has a few disadvantages over normal screen modes, it's memory consuming (a bitmapped picture such as a loading screen needs 10,000 bytes of memory all told when the colour is added) and it's difficult to scroll. Not impossible, just difficult. The *advantage* of bitmaps are that they have more colours available to them. I will be covering the basics of bitmaps in more detail a little further on.



$D016 has been covered before, we used it in the Ferris demo to scroll the message across the screen and smooth scrolling is the job of bits 0 to 2, along with bit 3 which controls the masks at the sides of the screen in the same way as $D011. Bit 4 is used to set multicolour mode, one of those little quirks of the C64's video system.

When multicolour mode is on the pixels don't just appear as they are stored, instead the bits are grouped together in twos horizontally and the 2 bit number they produce represents one of four colours. This reduces the horizontal resolution of the screen by half because each character is now only four pixels wide (physically they don't change size, but the pixels are now twice as wide) but we do have more colour control.

The character colour, in locations $D800 onwards, is used to control multicolour mode and, as we have done before, when we set it normally we have sixteen colours, values $00 to $0F. But multicolour mode allows us to mix standard characters and multicolour ones because the first eight colours (black, white, red, cyan, purple, green, dark blue and yellow, values $00 to $07) will produce their normal eight pixel wide characters but using colours $08 to $0F produces those eight colours again, but in multicolour mode. So where does the colour information for our two new colours come from? $D022 and $D023 are the multicolour registers. They function just like $D021 (the background colour) but only affect multicolour areas on characters using colours $08 to $0F. Our four possible colours therefore come from the character colour, $D021 for the background colour and the two multicolours, giving four in total.

Bitmaps (and I said I'd get back to them, didn't I?) work a little differently. When we use bitmap mode, the screen becomes the colour information for the bitmap. So, if we set $D011 to $3B to switch bitmap mode on and $D018 to $18, we now have 1,000 "characters" on the screen, each with it's own individual colours.
The lower four bits of the screen (in this example at $0400) represent one of the colours and the upper four a second, $00 will set both to black and $FF will make them light grey for example. But the most interesting use for bitmaps is with multicolour on, so we set $D016 to $18 and now we have lots of colours.

Why? Well, at $0400 to $07E7 we have 1,000 bytes of colour information, two colours per character. But in multicolour bitmap mode we *also* have the $D800 colour map, as used by the screen normally to give us a third colour and the background for our forth. And the $D800 colours are not limited to the first seven as with multicolour characters, they can be any colour from $00 to $0F.

The only drawback to having this much colour (three colours per character square) is that there are some 10,000 bytes of data required, 8,000 for the bitmap itself and 2,000 over two areas for the colour data, so moving it around is a no-no without some trickery which we will have to leave for a while yet.

Right, well thats the theory over with and indeed the article, but next time I'll be performing a more practical demonstration of how bitmaps work with some example code and a picture to play with.

In the meantime, a small challenge regarding $D018, can you work out what the values would be to put the screen at $3400 and the character data at $2800? Work it out on paper then check yourself against the start of the next part and if you have any questions, suggestions or whatever about this or C64 code in general, contact me.

**Commodore Free**
would like to thank Jason for allowing the reprinting of the HEXFILES the article was taken from
http://www.oldschool-gaming.com/