

Commodore Free

# Commodore Free

Issue Number 6



**Musician Richard Joseph  
Passed away 4<sup>th</sup> March 2007**

# Commodore Free

## Editor

Really got into some research this issue, I decided to do some reviews of Commodore vic 20 games, BUT guess what, I dug out Bomber and Blue meanies played them for 3 hours, and realised I hadn't done any actual reviewing!

So err no vic games reviews then in this issue, sorry about that, I did though find around 50 games I had in a box locked in the attic of my house, and fondly looked through the covers remembering the fun I had playing the games many years ago before the Commodore 64 came out. After the 64 the old vic games just moved to the attic never to see light of day again, well until now.

I have some concerns

I have been asked how I can justify the cost of the disk image, and copies of the magazine, turns out some members are copying the disk images and printing the pdf then charging for the copies this is disgusting, I can appreciate a small charge for cost of the disk and even ink but no more. If you are being charged more than a duplication cost please let me know.

The whole goal of the magazine is that its all FREE! So the disk image text and pdf are all free to download or read online, I suggest a local library or Commodore club should have some form of web browser if you don't have internet access.

Thanks this issue and last go to Al Jackson who has been mugged with the job of Disk creation, Al sort of volunteered but didn't realise it and now checks e text on the disk imaged D64

Remember the magazine is all by volunteers, ok 2 people myself and Al doing the disk text, it may not be professional but it's the best I can do, as I have said I am no writer, I also found I am not good at interviewing, if you feel you can do better then please do and send in some text or a review you have written

I just read the sad news about Richard Joseph Commodore musician

## Contents

Editor	Page 2
Contents	Page 2
Comments	Page 3
News	page 4
6502 Ramrom	Page 5-9
Rest of the Bible	Page 10-12
David Horrocks	Page 13
Hex files part 1	Page 14
Bits and pieces	Page 15
Diskimagery64	Page 16-17
New 64 Games	Page 18
Commodore128.org	Page 19-20
C= Preservation Project	Page 21-26
Andrew Fisher	Page 27-28
Richard Joseph	Page 29-30
The END	Page 31

**If you have an article you would like to share  
please feel free to send it to me**

[Nigelp2k@yahoo.co.uk](mailto:Nigelp2k@yahoo.co.uk)

**I have tracked 350 individual downloads of  
Commodore Free issue number 4**

More surprising is no one has actually criticised the magazine and ripped it to shreds am I doing everything right, I expect many will now correct this oversight.

# Readers Comments

**Pablo**

Hi,  
My name is Pablo and I'm a collector from Argentina. I'm really surprised about Commodore Free having copyright problems, I really cannot believe there are companies having problems with your magazine,

I guess Commodore users are still a "respectable force", so they still bothered about copyrights, but why don't they take your comments as free advertising?, I simply cannot understand they can have complains.

Ok, just keep going on and forget about them. Great Magazine and excellent work.

Pablo

## COMMODRE FREE

I agree with you that other users are braking copyright and never get "caught" I can sympathise with companies who still own copyright to images, text etc. the only thing I can say is that I have to be more careful

=====

**Yann**

Hi,  
Where can i get the 3 first magazines?  
best regards,

Yann

## COMMODORE FREE

Yann Due to copyright problems I can no longer distribut the issues, maybe you could find a user that has downloaded one of the first 3 issues and read his/her copy Thaks

=====

Damiano  
hi,  
i'm italian guy 1971 old ...

i lost the 1,2,3 number of fantastic!!!  
commodore\_free\_issue\_.pdf fromat is possible to received in my email

than'k for your good work!

bye Damiano

## COMMODORE FREE

Damiano  
I am glad you like the magazine, like many of the other questions here I am unable to distribute them due to copyright problems,

**Jocelyn**

Hi,  
I just discovered your magazine and i want to thank you very much for this great piece of art. I'm a huge fan of the commodore 64 (i programmed a lot in the 80's!) and I'm glad that i found your magazine. I was searching for a Commodore 64 dedicated magazines and this is the only one i found.  
Thanks again!

Regards,

Jocelyn

## COMMODORE FREE

Jocelyn  
Glad you like the magazine, why not contribute an article about some of the items you wrote for the Commodore, no one knows everything about the machine and its good to look at others experiences

As for other magazines there are a number of PDF downloadable but most are not in English, also there are a number of disk based magazines to run on real machines

Maybe I should feature an article on all commodore magazines Disk or paper based

Help would be very much welcome for this feature, if you know of a magazine disk, or pdf or paper based let me know, and how its ordered or downloaded.

=====

## For the record then on copyright

I was contacted by a number of company's claiming an infringement of copyright, or a misrepresentation of the company, the letters looked real and genuine and a number threatened court action if no further action was taken.

I decided to pull all copies and text from the website, the letters stopped, so I now ensure I try to clear all text and pictures and logos with the relevant copyright holders.

I am not going to list the companies involved as they may claim I am trying to start a vendetta against them, or a hate mail list, all I want to do is create a FREE magazine filled with relevant Commodore related items.

I make no money from the magazine, and in fact its costing me money to keep the magazine running, as well as the time it takes to compile the issue.

I can t blame companies for protecting themselves that's how they make money, unfortunately I am not motivated by money, I enjoy having fun and living rather than thinking all day about money

Regards  
Commodore Free magazine

## News and Information

### **Bomb Chase**

Richard bayliss releases and updated Bomb chase  
[http://www.redesign.sk/tnd64/games/Bomb\\_Chase\\_Revival.zip](http://www.redesign.sk/tnd64/games/Bomb_Chase_Revival.zip) With updated Graphics and music  
The idea of the game is to dash around collecting bombs before they explode. A simple game but the simple ones are usually the most addictive.

PRESS PLAY ON TAPE is proud to announce two upcoming concerts in the near future:

On Saturday april 7th we'll be playing in Frankfurt, Germany, at the Breakpoint'07 demo event. We have been wanting to play in Germany for a long time so we are really looking forward to this. If you live in Germany don't mis it! See the Breakpoint website for more info:  
<http://breakpoint.untergrund.net/>

### **Press Play on Tape**

On Saturday May 5th we'll be playing another concert in Copenhagen at "The Rock". The last concert in December went really well so The Rock wanted us to come and play again :) Hope to see YOU there! Tickets are available at <http://www.billetlugen.dk> (search for "PRESS PLAY ON TAPE") at 85 DKK + fee. At the previous concert the doors closed during the evening because the place was crammed, so be prepared and buy your ticket in advance :) Keep an eye on our event page for up-to-date info about support band and more:  
[http://www.pressplayontape.com/?pid=concerts\\_therock2007](http://www.pressplayontape.com/?pid=concerts_therock2007)

ROC=K ON!

PRESS PLAY ON TAPE

### **Cottonwood BBS**

I've been hard at work on adding "goodies" to my BBS, which is now running on Color 64 software. I have added multi-upload capability, 40/80 column selection for message entry, a "one-liner" feature, and best of all, some online games! I've added Horserace (a "classic" Color 64 game), Master's Empire (a very good upgrade to the classic Empire game), and Stock Market (another great game).

Master's Empire and Stock Market are both multi-player games. Everyone gets to take turns while they're signed onto the BBS, and over a period of weeks, the game progresses until one person reaches the pre-set goal and is proclaimed the winner.

I'll be on the lookout for more Color 64 mods to improve my setup... I'm always looking to improve... :-)

Check out what's new on Cottonwood BBS today by calling (951) 242-3593.

Take care...  
-Andrew

### **"New version of JSIDPlay available"**

A new version of the 100% Java sid player is available at [www.jac64.com](http://www.jac64.com). It plays much more sid tunes than previous versions since it is based on the latest JaC64 version and use Dag Lemms sidplay assembly routine. The full HVSC library is available to listen to!

<http://www.jac64.com/sid-music.html>

### **Pre-orders are now open for "The Commodore 64 Games Book 1982 - 199x".**

This is the follow-up to the ZX Spectrum book released in December 2006 by Andrew Rollings. (<http://zxgoldenyears.com>)

Author Andrew Fisher will review and describe over 200 classic (and not so great) Commodore 64 fans, revealing trivia about the machine, the programmers and the games.

There will be a limited print run, and pre-ordering will help guarantee the book goes to print.

<http://c64goldenyears.com>

### **More information later in this issue of Commodore Free magazine**

### **COMMODORE FREE Programming section**

I am pleased to announce that if all goes to plan! We should have a programming section in the magazine, I have asked the writer to produce an Introduction to programming.

So what's new with that there are loads of these, Yes dear reader and if you are a programming virgin you will realise they all start slow then seem to gather pace till it's a rush to the end and somewhere mid section you seem lost!

The Commodore Free programming guide will walk the user through the basics of Coding a game in BASIC using Commodore graphics then move to using sprites, and sounds, the adding music then finally to producing the whole game in machine code.

I will announce more after I have the full ok from the writer taking up the challenge.

Many readers have requested that there be a programming section and I hope this fills the gap, If you have any programming experience or advice please share it with other readers.

Again I would like anyone whatever background to submit an article on something Commodore related, Thanks to all readers who contributed to this issue.

# The 6502-RAMROM as drive expander

© Wolfgang Moser and Nicolas Welte

A short overview of the 6502 microprocessor based hardware extension, that was designed by Nicolas Welte . It contains a dedicated test report with my personal experiences and some application examples.

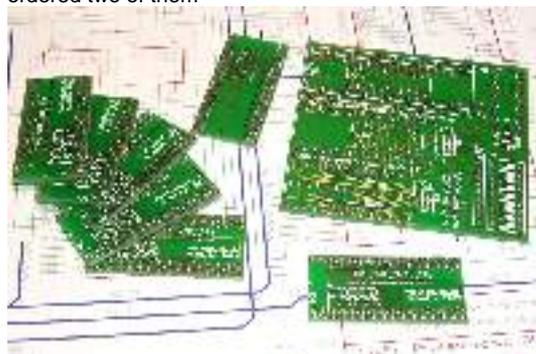


Originally the 6502-RAMROM  
[\[6502-RamRom inserted into 1541 drive, side view,](#)

was conceived by Nicolas Welte and Paul Förster for the Commodore PET series computers as a RAM and ROM expansion and diagnostic board, leading to it's first name: PETRAM. Based upon many discussions with Nicolas in the design phase, he (we ?) decided to make use of a Flash-ROM chip instead of just EPROMs. And other 6502 microprocessor based Commodore computers like the VIC20 or the 1541 disk drive should be taken into account.

The processor adaptor board can be configured for different systems by simply replacing one little GAL 16V8 programmable logic chip. To keep the hardware simple, the C64 computer or other non pure 6502 micro's are not supported.

6502-RAMROM prototype test report from Womo, 2003-02-07 HardwareSome weeks after Nicolas received his professionally manufactured printed circuit boards (PCBs) of the 6502-RAMROM I ordered two of them



[Delivered 6502-RamRom and Flash-Adaptor PCBs,

along with some of the needed components. Since I don't own a PET computer nor a VIC20, I ordered only two GALs for the Commodore 15x1 drive replacement configuration. Since I can call myself an experienced electronic craftsman, soldering the

board was no big problem. Although I have to state that, for a beginner, it may be too hard. Please look

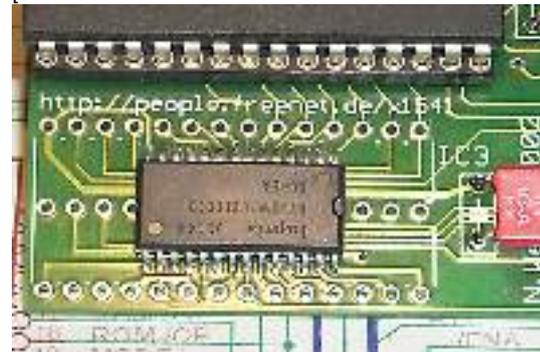
onto Nicolas' construction page and carefully decide, if you feel able to solder the board. If not, you should order the ready-built extension board



[6502-RamRom top view,



[6502-RamRom bottom view



[Soldered optional SMD-RAM,

At this time, the Flash update software wasn't available for disk drive in-circuit updates, so I had to use my IBM-PC based c't-Flasher for burning some contents into the ROMs. I tested several speeder ROMs, flashed different versions into the four ROM banks and switched them forth and back. Everything worked fine, cool.

# Commodore Free

## Dolphin-DOS 2.0

Before I could test the RAM options I needed to solve some problems. I wanted to test out the Dolphin-DOS 2.0 floppy speeder ROM which makes use of additional RAM. But for this and other speeder systems I needed to install a parallel cable from the floppy's 6522 VIA chip to my C64's user port. And I needed a simple solution to quickly test out different C64 Kernal replacements. The second was no problem, because I also ordered some Flash-ROM



[C64 Flash-Adapter Kernal ROMswitch, top view,

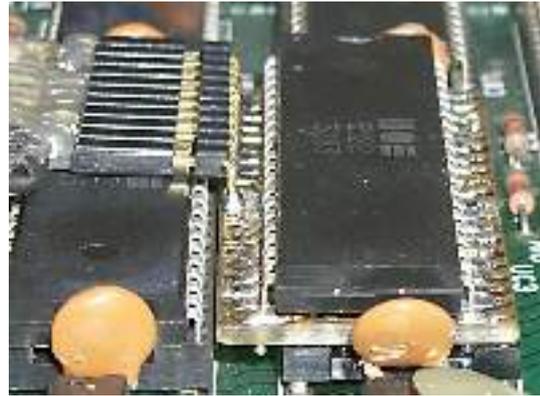


[Inserted C64 Flash-Adapter Kernal ROM switch.]

adaptor PCBs and could easily build a Flash-ROM to C64 Kernal adaptor. Inserting a parallel cable into my 1541 drive was a little bit problematic, because the 6502-RAMROM board overlays the 6522 VIA chip location. First I used some additional 40 pin sockets to lift the 6502-RAMROM a bit higher, but I didn't feel happy with it; the case couldn't be closed now. So I built a very special lowest-profile parallel cable adaptor.



[1541 low profile parallel cable adaptor, bottom view,



[1541 low profile parallel cable adaptor, top view,

This all done, I first tested my beloved SpeedDOS (35 and 40 tracks) and made sure, that it worked fine. I also tested some of the most important copy programs to make sure, that the parallel cable and the ROM replacements don't fail. Preparations for the test of Dolphin-DOS 2.0 ] were done, flashing the 1541 drive ROM and C64 Kernal. I configured the RAM option switches to DD2 and made the first steps. Anything looked fine. And wow, DD2 isn't as slow as I thought all the time. Although beaten by Professional-DOS working with it brings fun. I couldn't see any problems like load or CRC errors, all games loaded fine and were playable. So the RAM seemed to work fine also.



[Space between parallel cable adaptor and 6502-RamRom (view 1),



[Space between parallel cable adaptor and 6502-RamRom (view 2),



[Space between parallel cable adaptor and 6502-RamRom (lifted),

# Commodore Free

## In-circuit flashing

The next test phase came, when Nicolas was finished with the in-circuit drive flashing software . This piece of software was designed a very, very special way. The ROM, that is going to be burned is not loaded into the computer's memory and transferred back into the drive's Flash-ROM memory. That would be much too long winded. No, the ROM contents are transferred directly, block by block, from the disk's surface into the drive's buffer memory and further into the Flash-ROM.

This is possible, because Flash-ROMs from Atmel [1] were used, that can be flashed page by page. Unfortunately the Flash software does only support chips from Atmel until now, but you can use differently sized ROMs, if you want (AT29C256, AT29C512, AT29C010) and if you don't need that much ROM banks. For the first times flashing new drive ROMs went flawlessly, but whenever a SpeedDOS-ROM was configured via the main ROM switches I ran into problems with flashing other ROM banks.

Some tests revealed, that whenever a SpeedDOS-ROM was selected with the 6502-RAMROM and the Flash enable switch was set, the drive hung up, when a file was loaded or the error channel was retrieved with the '@' command. I discussed it with Nicolas and he told me, that he found similar problems with the original CBM VIC20-ROM. We found out that some misdesigned ROM code fragments were responsible for this (Note: the VICE emulator's built-in monitor was exceedingly helpful with this: <ALT>-m, device 8:, watch store 8000 FFFF, exit).

Whenever the Flash enable switch is set and parts of code executed by the desired 6502 processor store values to the ROM (to the Flash-ROM), the internal logic of the Flash blocks the whole chip so that even reading the ROM fails for some time. After discussing this point in the cbm-hackers mailing list [1], Nicolas managed to fix this ROM bug and so did I with the SpeedDOS-ROMs .

Now, with the patched ROMs, I was able to Flash other ROM banks without the need to switch back to the original CBM DOS 2.6 ROM (which worked flawlessly with all previous tests). Problems with the prototype another problem manifested when analyzing the »store-to-ROM« problem with the drive expansion 6502-RAMROM prototype. What happens, when someone wants to flash the whole ROM, but the DD2 RAM configuration is selected?

Because the DD2 RAM overlays the drive's address space from 0x8000 to 0x9FFF the Flash-ROM cannot be addressed there. Furthermore, DD2 went crazy, whenever the Flash enable switch was set. Nicolas detected the problem by carefully reviewing the 1541 GAL equations.

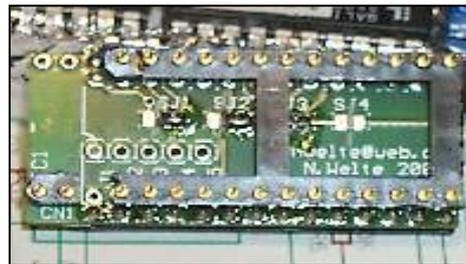
The Flash option switch disables the overlaying RAM for write operations. Not a big task, because only the GAL needs to be replaced, I thought. First we decided to patch the floppy DOS ROM part of Dolphin-DOS 2.0. The ROM code should not access the 8kB RAM at the memory address location 0x8000...0x9FFF anymore, but at the addresses 0x6000...0x7FFF. That way no changes to the 6502-RAMROM would be needed, because the RAMboard configuration could be used instead. And flashing the full 32kB ROM bank should become possible, too. But Nicolas reported, that the patched DD2-6 ROM didn't work, although other tests with the VICE

emulator showed no problems. He found out, that there was a real hardware problem, that depends on the special address decoding behaviour of the CBM drive series (RAM and especially VIA locations mirrored at different addresses; IRQ flag handling). Since the PET and VIC20 computers don't contain such mirrored device address spaces, this problem couldn't be detected with the PET/VIC20 6502-RAMROM prototypes.

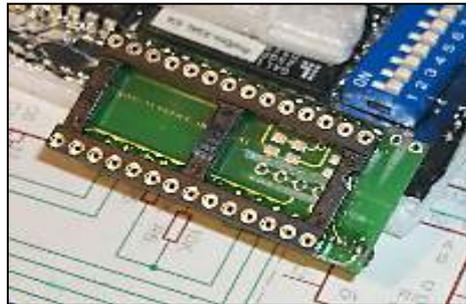
## The Solution

Nicolas worked out a solution, that needed the least possible changes to the PCB in combination with changed GAL equations. Without the GAL, this fix would have been a huge patch orgy. Unfortunately you have to redo the patch if you want to use the 6502-RAMROM in a VIC20 or PET again. Although not needed I installed some jumpers to easily configure the PCB for both options.

In the meantime I tested flashing other Flash-ROMs with different sizes. I had an AT29C020 as spare and tested the bigger flashing page size. Reading out the ROM contents with the c't-Flasher at my PC again showed no differences. Note: you can't access/flash half of the banks of this ROM type without further hardware modifications. Then I tested an AT29C256 as a small replacement for my C64 Kernal switch board.



[AT29C256 to 32 pin Flash-Adaptor, bottom view,



[AT29C256 to 32 pin Flash-Adaptor, top view,

Before I had to build a small Flash adaptor, because the 29C256 has got a different pinout (28 pins) than the standard 32 pin Flash ROM chips. I hacked Nicolas' Flash-ROM board a little bit by cutting some traces. Although the Flash-ROM PCB was not designed for this purpose it fitted my needs best (instead of building a cable wired adaptor).

Flashing the 29C256 with the least possible page size brought me a very small kernal switch board in a few minutes; thanks to the well designed Flasher software, that is able to burn different parts of the ROM chip in several steps (by selecting a start address of 0x8000, 0xA000, 0xC000 or 0xE000).

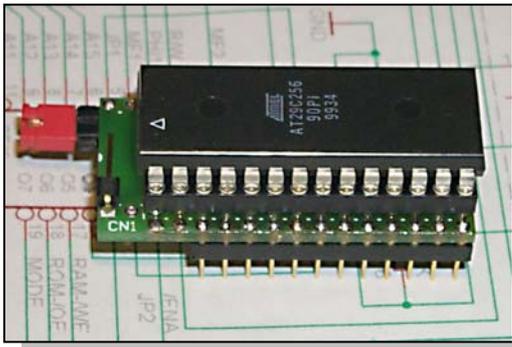
## Commodore Free

6502-RAMROM final release test report from Womo, 2003-02-16 Retesting different flashing scenarios to check, that the problems with Dolphin-DOS were gone After I got the new changed GALs I repeated all the earlier tests: Flashing the original CBM DOS V2.6 into bank 0 Selecting bank 0 and flashing the patched SpeedDOS 40 tracks ROM into bank 1 Selecting bank 1 and flashing the original Dolphin-DOS 2.0 ROM into bank 2. Because of the applied SpeedDOS ROM patch, the Flash procedure went flawlessly.

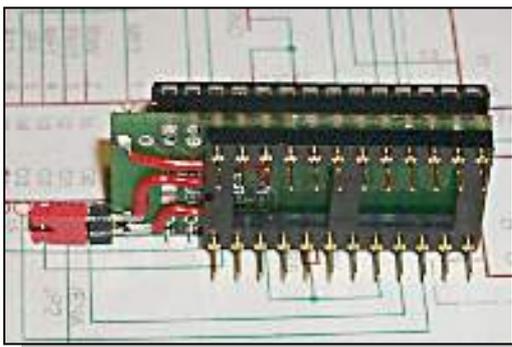
Selecting bank 2, configuring the DD2 RAM configuration and flashing the patched Dolphin-DOS-2.0-6 ROM (RAM a location 0x6000 to 0x7FFF) into bank 3. No problems with that and it shows, that you can now Flash the ROM locations from 0xA000 to 0xFFFF, when the original Dolphin-DOS 2.0 is enabled. But don't try to Flash the ROM location at 0x8000. Because it is used by DD2's RAM, the drive may become stubborn.

Selecting bank 3, configuring the RAM-Board configuration and flashing the Dolphin-DOS 3.0 ROM into bank 0. With the patched DD2-6 ROM the full ROM bank (0x8000 to 0xFFFF) could be flashed without any problem, thanks to the hardware modification and the new GAL.

After preparing some matching C64 Kernal ROMs



[AT29C256 to C64 Kernaladaptor, top view,



[AT29C256 to C64 Kernaladaptor, bottom view,]

I started testing the burned SpeedDOS and Dolphin-DOS versions. As mentioned earlier the patched SpeedDOS and also the original Dolphin-DOS ROM worked without problems. I also tested some special

Dolphin-DOS copy programs (Dolphin-Copy) without problems. The patched Dolphin-DOS-2.0-6 (RAM at 0x6000) is another task, with this ROM, the Dolphin-Copy program doesn't work, because the RAM location seems to be hardcoded into the copy tool. I also tested Dolphin-DOS 3.0, which worked fine with the exception that it was nearly as slow as the original CBM DOS.

No wonder, because DD3 makes use of an extra parallel port chip (6821 PIA) within the drive. It cannot communicate over the standard parallel port cable connected to the VIA 6522, therefore it has to use the slow serial bus transfer routines.

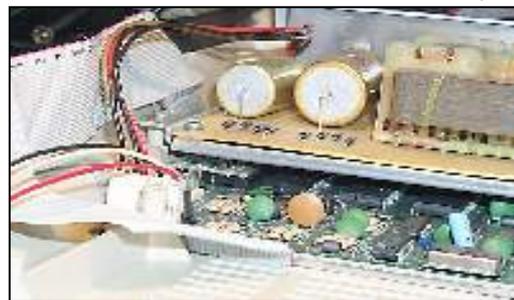
The future and some ideas Until now, mainly the ROM part of the 6502-RAMROM was tested, in the future I'll have to look at the RAM configurations, especially in some applications for the maxRam-Mode. Perhaps Markus Brenner wants to jump in here and extends his Burstnibbler based transfer tool »mnib«, so that much more sophisticated copy protections can be dumped to G64 disk images. I already started analyzing the Dolphin-DOS 3.0 DOS ROM looking for all the parallel port accesses to the 6821 PIA. Maybe I'm able to patch it, so that it makes use of the VIA 6522 parallel port.

This way people may use Dolphin-DOS 3.0 without hardware hacking; that means adding an additional 6821 PIA into the floppy. I should take my 1571 floppy drive sometime and test it with the 6502-RAMROM, too.

There are other floppy speeders, especially designed for this drive. Not only to mention Dolphin-DOS 3.0 for the 1571, but especially Prospeed-1571. With this drive, the maxRam mode of the 6502-RAMROM cannot be used, it would overlay the 1571's CIA and the registers of the WD177x controller.

It may be interesting what could happen, if the maxRam mode is used nevertheless. Not enough, there are more problems with the Commodore 1571 disk drive.

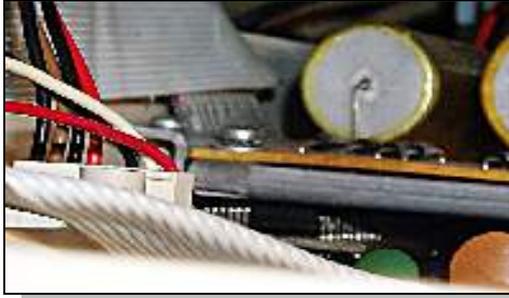
Since there is incredibly little space between the mainboard and the transformer assembled over it, I



currently don't know, how to insert the 6502-RAMROM into the 1571 drive best. One solution may be to connect the board with some flat cable to the processor socket. As for the parallel cable connector (1571 Burstnibbler cable compatible) I already built another lowest profile adaptor.

[Inserted 1571 low profileparallel cable adaptor,

## Commodore Free



[Space upon 1571 low profile parallel cable adaptor,



[1571 low profile parallel cable adaptor, top view,



[1571 low profile parallel cable adaptor, bottom view,

### Conclusion and application scenarios

Speeder system »emulator« »Emulating« different DOS ROMs and simple speeder systems like Jiffy-DOS, SpeedDOS or Dolphin-DOS is the playfield of the 6502-RAMROM, when inserted into Commodore's disk drives. You only need to add a simple parallel cable for some of these systems.

### Atmel Flash device programmer

In combination with a patched Dolphin-DOS (DD2-6), this hardware extends your floppy disk drive to a full blown high speed Atmel Flash device programmer. You will never need to handle with EPROMs and EPROM burners again. There is no need anymore for transferring the ROM files slowly to the computer memory before they can be written to the EPROM. The well designed flasher software makes your life even simpler.

Maybe you want to replace your system ROMs with DD2-6, which requires some little hardware modification (using address line A14 for switching the upper and lower part of the DD2 ROM at 0xA000/0xE000); this way you can flash the whole

chip. Nicolas is also working on more functions, like programmable bank switching about unused VIA/CIA port lines.

Maybe not only Dolphin-DOS-2.0-6 can speed up the Flash process a little bit, but other drive speeders like Jiffy-DOS, too. Because of the special designed In-System-Flash software, the whole process is not really slow on a 1541 drive equipped with a standard CBM DOS (35 seconds for 32kB).

So you don't need to install a speeder system or ROM in any case, instead it is an option. And Nicolas said, the Flash software contains some potential for speedups with the 1541 drive. Let's wait, test and see...

RAM (only) expander (1541/1571 RAMBoard) and RAM diagnostic module More applications of this board may rely on the RAM option only, you could save some money by not making use of the Flash-ROM chip. Copy software like the well known tool series Maverick already know how to use a RAMBoard equipped drive which the 6502-RAMROM can easily be configured to »emulate«.

Perhaps Michael Klein (cbm4linux or other people may want to develop special software, that makes use of a maxRAM equipped disk drive (nearly 32kB of RAM usable). Not to mention the RAM diagnostic functionality that replaces the drive's builtin RAM by the one from the 6502-RAMROM.

Things, the 6502-RAMROM can't do easily Because the hardware was mainly designed for universality and simplicity, it is not a general speeder system hardware emulator. Systems like Dolphin-DOS 3.0, Professional-DOS or Prologic-DOS cannot be used without further extensions or ROM patches. This purpose may be the job of another hardware extension board, one of the projects I am currently working on.

**Taken from the website**  
<http://d81.de/6502RamRom/>

© Wolfgang Moser and Nicolas Welte  
Re printed with owners consent

### Notes From Wolfgang Moser on the project

Email: -Wolfgang Moser

in general I agree about reprinting parts of my online article, if authorship is made clear and the origin of this article is named. Spelling corrections may be done. Then the article sometimes contains outdated informations, like e.g.:

"Nicolas is also working on more functions, like programmable bank switching about unused VIA/CIA port lines."

Nicolas is currently not working on extensions to the 6502-RamRom, it perfectly does its job as it is.

Since Nicolas also may want to agree to reprint my article or because he wants to point out other things, he got a Bcc copy of this mail.

**Commodore Free I received no email from Nicolas but decided to print the article**

## THE REST OF THE BIBLE

By Dave Moorman

OPEN lf,dv,ch(com)

Here is how you access the disk drive. LF is the Logical File number, which you pick. You can open more than one file, but each must have its own Logical File number.

DV is the device number (8 or higher for disk drive). CH is the Channel, again a unique number for each open file (we often just use the LF number, unless the OPEN is for a DISK COMMAND).

### DISK COMMANDS

We can send commands to the disk to Scratch a file, Format (New) a disk, Rename a file, or Copy a file. Each is sent over the Command Channel -- 15. Here are some examples.

```
OPEN 1,8,15,"S0:FILE":CLOSE1
This will Scratch the file named FILE from the disk.
```

```
OPEN 1,8,15,"N0:DISK NAME,ID":CLOSE1
This will format a new disk, preparing it to accept data.
```

```
OPEN 1,8,15,"N0:NEW NAME":CLOSE1
NOTE: no ID characters. This will wipe a formatted disk clean and give it a new name.
```

```
OPEN 1,8,15,"C0:NEW FILENAME=OLD
FILENAME":CLOSE1
This copies the file OLD FILENAME to another file called NEW FILENAME.
```

```
OPEN 1,8,15,"R0:NEWNAME=OLDNAME":CLOSE1
This changes the name of the file OLDNAME to NEWNAME.
```

Channel 15 also lets the disk communicate with the computer. Here is a trick we use all the time to find out whether a certain filename is on the disk.

```
10 F$ = "FILENAME"
20 OPEN 1,8,15,"R0:"+F$+"="+"F$
30 INPUT#1,EN
40 CLOSE1
```

When this is done, EN will contain either 62 or 63. If EN=62 then the filename is not on the disk. If EN=63, the file name IS on the disk.

Once the command channel is open, we can use PRINT#lf to send further commands to the drive. We do this in our Scratch and Save routine:

```
60000 D=PEEK(186):IFDV<8THENDV=8
60001 OPEN1,D,15,"IO":N$="PROGRAM"
60002 PRINT#1,"S0:"+N$:CLOSE1
60003 SAVEN$,D:VERIFY N$,D:END
```

OPEN1,D,15,"IO" (should be "I" zero) makes sure the disk drive is awake and aware. Then the program name is put into N\$. Line 60002 scratches the program name, and CLOSEs the logical file. Finally, we SAVE N\$,D, then VERIFY N\$,D to make sure we got a good save.

We also use OPEN to open a file for reading or writing. Here is code that will save three variables to a file, followed by the routine to get those three variables.

```
10 DV=PEEK(186):IFDV<8THENDV=8
:
1000 OPEN 1,DV,15,"S0:DATAFILE":CLOSE1
1005 OPEN4,DV,4,"DATAFILE,W,S"
1010 PRINT#4,A$
1011 PRINT#4,B$
1012 PRINT#4,C$
1013 CLOSE4
1014 RETURN
:
2000 OPEN4,DV,4,"DATAFILE,R,S"
2001 INPUT#4,A$
2002 INPUT#4,B$
2003 INPUT#4,C$
2004 CLOSE4
2005 RETURN
```

When we OPEN a file, we need to tell it the filename, whether we will be Writing it or Reading it (W or R), and what kind of file it is. We have two normal kinds of files: Program and Sequential. For short data information, use a Sequential file.

So, after scratching the file, we open it to Write, Sequential (.W,S). Then we use PRINT#lf to print the data to the file. The best way is shown above, with each variable printed separately. We CLOSElf and RETURN when finished.

To get the data back into the variables, we open the file to Read, Sequential. (You can leave off the ",R,S" when opening a file to Read.) Then we INPUT#lf each variable exactly the same order as we did the PRINT#lf, CLOSElf, and RETURN to the main program.

OPEN is also our way to get data to the printer.

```
100 OPEN4,4,7
110 FOR X = 0 TO 50
120 PRINT#4,A$(X)
130 NEXT
140 CLOSE4
```

Here we assume that each element of the A\$ array has one line of text for the printer. You will have to experiment with your printer on this -- and read the manual for special characters and effects. At LOADSTAR, we assume a 66 line page with 80 characters to a line -- and don't do much that is fancy.

[NICKEL GAMES has a built in function to turn a C-64 print-out into a PC TXT file. After doing a print-out in VICE, press <Alt-Tab>, and click on File>Print. In a moment, your print-out will be displayed. You can copy and past it to a word processor, save it as a TXT file, or send it to your printer.]

NOTE: Always CLOSE the logical file after use. Getting proficient with OPEN takes a lot of practice! See SECRETS for other stuff about disk access.

# Commodore Free

## PEEK(loc) (fun)

Returns the contents of the memory byte at LOC. We use PEEK(186) to discover which disk drive device was last used. You will use it a lot for advanced tricks.

```
10 DV = PEEK(186)
```

## POKE loc,byte

Puts BYTE value into memory location LOC. Especially important for controlling C-64 features not included in BASIC 2.0.

```
10 POKE 53280, 0:POKE 53281, 0
(Turns screen border and background black)
```

## POS(0) (fun)

Returns the current position of the cursor on the screen row (0 - 39). I have no idea what this would be good for!

## PRINT (com)

Probably the most important and versatile command in BASIC 2.0. It puts characters, strings, and values on the screen.

End the PRINT with a semicolon to keep the cursor from automatically dropping down to the first column of the next row (called a carriage return). Use a comma to move the cursor to the next pre-set tab location on the line.

Semicolons and commas can separate data in a PRINT command. This is a command you will have to play with!

## READ (com)

Reads values or strings in DATA statements into variables.

Make sure the number of data items in the DATA statements match up with what you are expecting! For string data, enclose each string in double-quote marks.

```
100 FOR X=1TO4
110 READ A$(X)
120 ? A$(X)
130 NEXT
140 END
20000 DATA "DADDY", "MOMMY"
20001 DATA "JUNIOR", "SIS"
```

## REM (com)

Short for REMark. Everything on the program line after REM is ignored by the computer. Good for commenting on what the program is doing at a particular point.

```
10 REM BEGINNING OF PROGRAM
:
199 REM MAIN LOOP AT 200
:
60100 REM (C)2005 DAVE MOORMAN
60110 REM COPYING THIS PROGRAM BY
60111 REM ANY MEANS WILL GET YOU
60112 REM A SLAP ON THE HAND!
```

## RESTORE (com)

Resets data point so that the next READ receives the first data element from the first DATA statement. Not very useful, really, since we usually READ data into arrays, which are a lot easier to handle.

## RETURN (com)

Ends a subroutine and sends the program back to the GOSUB that jumped to this place.

## RND(n) (fun)

Returns a random number between (but not including) 0 and 1. If N is a negative value, the value is used to seed the random number generator. The result will always be the same for each negative number. If N is 0 or positive, a new random number is generated. Most say using 1 is best.

Actually, computers cannot do truly random numbers. Everything inside a computer is far too logical. However, with a bit of math, it can generate a list of unknown numbers. When you use a negative argument, you reset the generator.

I have heard some arguments about how an argument of 0 is not as random as a positive argument. Who knows?

To get a useable number, you might use this custom function:

```
10 DEF FNR(X)=INT(RND(1)*X)+1
```

If you want values from 0 to X-1, leave off the last +1. Now any time you need a random number, just use the function. Here is a routine to shuffle 52 "cards".

```
10 DIM CD(52)
15 DEF FNR(X)=INT(RND(1)*X)+1
20 FOR X = 1 TO 52: CD(X)=X:NEXT
30 FOR X = 1 TO 52: R=FNR(52)
40 C=CD(R): CD(R)=CD(X): CD(X)=C
50 NEXT
```

Please do NOT do this:

```
10 DIMCD(52)
:
200 DEF FNR(X)=INT(RND(1)*X)+1
210 CD=FNR(X):IFCD(CD)<>0THEN210
220 CD(CD)=1
230 RETURN
```

You will be able to quickly "pick a card" at first, but as the cards are taken, each pick will take longer. To find the last card, you are likely to look at about 52 taken cards!

## RUN [ln] (com)

Runs a program. You can begin a program at a given line number. You can also use RUN in a program to start all over from the beginning.

## RIGHT\$(s,n) (fun)

Return the rightmost N characters from string S.

```
10 A$ = "THIS IS A TEST"
20 ? RIGHT$(A$,4)
(This will print "TEST".)
```

## SAVE (com)

We have already covered the correct way to put a Scratch and Save routine in every program. In a pinch, you can simply

```
SAVE "FILENAME",8
```

## SGN(n) (fun)

Returns -1 if N is negative, 0 if N is 0, and 1 if N is positive.

## SIN(n) (fun)

# Commodore Free

Returns the sine of N in radians.

```
10 A = SIN(1)
(A contains the value 0.841470985.
```

SQR(n) (fun)

Returns the SQuare Root of N.

```
10 A = SQR(9)
(A contains 3)
```

STOP (com)

Causes a break in the program. Line number is reported. Useful for debugging.

STR\$(n) (fun)

Returns a string of the value N.

```
10 A$ = STR$(123)
(A$ contains " 123")
15 V = 4
20 B$ = RIGHT$("0"+MID$(STR$(V),2),2)
(B$ will contain "04")
```

SYS loc (com)

Executes an ML routine at memory LOcAtion. SYSie commands are often used with ML Modules just like BASIC commands, complete with parameters. Be sure to read the documentation that comes with modules.

TAB(n) (print com)

Moves cursor to N column when printing.

```
10 PRINT TAB(17)"CENTER"
(The word "CENTER" is printed in the center of the line.)
```

TAN(n) (fun)

Returns the tangent of N in radians

```
10 A = TAN(1)
(A contains 1.55740772.)
```

USR(n) (fun)

Executes ML routine in memory pointed to by locations 784/785. N is placed in the Floating Point Accumulator. Returns the value in the Floating Point Accumulator. This is a function, not a command, so you must either PRINT USR(N), or put the result into a variable -- A = USR(N).

VAL(s) (fun)

Returns the value of string S. Non-numeric characters return 0.

```
10 ? VAL("12A7")
20 ? VAL("NAME")
(Prints 12 and 0.)
```

VERIFY (com)

Like LOAD, except VERIFY compares the program in memory with that on disk without changing either. If the memory matches the disk file, OK is printed.

WAIT loc,v [,eor] (com)

Causes the program to pause while waiting for the byte at memory LOcAtion ANDEd with V and EOReD with C is not zero. C is optional. This is a great command for waiting for a keystroke.

```
10 POKE 198,0
20 WAIT 198,1
30 GETZ$
```

The program pauses while WAIT watches location 198 (which holds the number of keystrokes currently in the keyboard buffer). When a key is pressed, the program continues, GETting Z\$.

## SECRETS

Here are some "secret" routines that do useful things.

BLOAD

To load a binary file or block of data to any place in memory (except between 53248 and 57343).

```
1 DV=PEEK(186):IF DV<8 THEN DV=8
5 DEF FNH(X) = INT(X/256)
6 DEF FNL(X) = X - FNH(X)*256
9 ADDR = 49152: REM ADDRESS WHERE FILE
WILL BE LOADED
10 SYS57812"FILENAME",DV,0:POKE780,0
20 POKE781,FNL(ADDR)
25 POKE782,FNH(ADDR)
30 SYS65493
```

BSAVE

To save memory to a file. Cannot save from 40960-49151, or above 53248.

```
35 B = 49152:REM BEGIN ADDRESS
36 E = 53248: REM END ADDRESS + 1
40 SYS57812"FILENAME",DV
50 POKE 193,FNL(B):POKE 194,FNH(B)
60 POKE 174,FNL(E):POKE 175,FNH(E)
70 SYS62954
```

POSITION CURSOR ON SCREEN

his handy routine will put the cursor anywhere on the screen, where X = 0 through 39 and Y = 0 through 24.

```
1000 IF Y=0 THEN ?"<Shift-HOME>";:GOTO1020
1010 POKE214, Y-1:?
1020 ? TAB(X)"WHATEVER!"
```

Note: If the printing wraps around, an extra screen line may be inserted. Avoid printing to column 39 if possible!

Read through these commands and functions again! Try the code. Try experiments. Programming requires a broad knowledge of possible commands and functions and ways they can go together.

Dave Moorman  
Editor, LOADSTAR  
September 12, 2005

# Interview with HOXS-project David Horrocks

**Q** Hello David, and thanks for taking the time to answer our few questions. Perhaps you want to introduce yourself first?

**A** I am David Horrocks and I live and work in England, Cheshire as a Software Engineer. I currently design and maintain data management applications which makes Hoxs64 a very different type of project to what I do at the office

**Q** When did you start the HOXS-Project (we may assume that the name of the Emulator comes from the surname of the author?) and what has been your aim?

**A** Yes. The name Hoxs comes from my surname. I sometimes wonder if I should have come up with something more snazzy but at least the originality presented no problems getting a .com domain name. The Hoxs64 project starting during a period of annual leave early in January 2001.

The initial aim was for me to understand what made a computer like the C64 tick. I was inspired by seeing other emulators and I mistakenly thought it would be a quick job to throw together a CPU and graphics emulation based on the programmers reference guide.

But things were more complex as I delved deep into the inner working and then the project just grew and grew. The CPU emulation being the first part to be designed was quiet tedious. There was no visual result for me see how things were progressing. The disassemble window was a necessary item to help view the progress of the CPU.

I remember getting the CPU to execute the initial C64 ROM boot code but would get stuck looking for a non existent VIC graphics chip. That was my cue to begin coding the VIC emulation. It was major milestone just to get to the point where Hoxs64 could display the words

\*\*\*\* COMMODORE 64 BASIC V2 \*\*\*\*\* as every C64

emulator author will surely testify. A large amount of work takes place before there is even just one pixel dot to show for. There is no half way house to show. C64 software simply will not run until you have all of a RAM, ROM, CPU, VIC and CIA emulation all put together in the right way

**Q** Why do you think there is a need for yet another C64 Emulator for Windows? What can HOXS do that WinVice or CCS64 can't?

**A** With good emulators like Vice and CCS it is tough to argue the need for a new emulator. These emulators have features that are currently missing in Hoxs64 such as ROM cartridge support, mouse and tape seek interface just to mention some. But there are a couple of unique things about Hoxs64. Vincent Joguins' <http://www.oldskool.org/disk2fdi/> FDI support has recently been added. FDI support gives Hoxs64 better accessibility to copy protected disks.

As far as I know, Hoxs64 is the only C64 emulator in the world to emulate cycle based sprite collision and the only one that can run Emu-Fuxx0r protected

software  
[http://www.btinternet.com/~hoxs64/Plush\\_Emu-Fuxx0r\\_V2.zip](http://www.btinternet.com/~hoxs64/Plush_Emu-Fuxx0r_V2.zip)

**Q** Is HOXS rather addressed to the "skilled-programmer" than to the "quick-gamer"? Because there are several games that do not run on the emulator - how compatible do you think is HOXS in its present version?

**A** Hoxs64 is aimed at all C64 users both programmers and gamers. I will admit that until I revamp the debugger then programmers are not as well catered for as compared to the fully featured debugger in WinVice.

I am not aware of any games that do not run in Hoxs64. If anyone finds a game not working then I would be happy receive the file image for the purpose of improving the emulation. Occasionally a user will mail a game that does not work. One such example was MicroLeague Baseball which recently got fixed. This game required D64 custom track format support which was subsequently added to the disk drive emulation.

In my humble opinion the C64 side of the emulation has reached such a high level that no legacy software should fail. It is still possible that there is an inaccuracy in the C64 emulation or 1541 disk emulation that will cause some software to fail and I would be happy to inspect such software.

**Q** What fixes/Updates for HOXS are waiting for us in the immanent future?

**A** Updates for the future hope to include a better debugger, speed optimisation, tape seek and save. But these features are not immanent at present. The only thing that controls progress is how much time I have to spend.

**Q** Is it planned to establish a HOXS-port on Linux or even MacOS Computers?

**A** No Linux or Mac port is planned largely due the fact I don't have Linux or Mac. My priorities are to improve the emulation accuracy and user accessibility to both gamers and programmers.

Have a big "Thank You" for the interview!

Regards  
David

Hoxs64 is a Commodore 64 direct X emulator written by David Horrocks with the help of documents written by the following people.

Many thanks to:  
Christian Bauer for the VIC-II 6569 info.  
Marko Makela for the CPU 6510 info.  
Wolfgang Lorenz for the CIA 6526 info.  
Ruud Baltissen for the floppy disk controller info.

## THE HEX FILES - PART 1

Written by Jason Kelk

A few people out there have expressed an interest in learning machine code. And after all, if you want to write successful games, demos or utilities it is the best language to learn, albeit a bit unfriendly. The main problem with machine code is its simplicity. There, I bet that confused you a bit eh? What I mean is that almost anything can be done in a single BASIC command can take a bit more work in machine language but at a greatly increased speed.

Right, that's the intro over with so lets start looking at some commands. The first one I've decided to explain is RTS. RTS is short for ReTurn from Subroutine and its job in life is just like the BASIC return command.

So why am I explaining it first? Well, if you just use it without subroutines it also acts like the end command and for the purposes of this course we will be using RTS to finish our code. So why does it have such a short name? That's due to the fact that all assembly language commands are only three letters long! Now, if you were to just put the command RTS into an assembler and try to run it all that would happen is the "READY." message would come up. Dull, huh?

For the next bit I need to explain a bit about the workings of the C64; imagine in your mind for a moment a long line of little boxes and that each box has a number on it from 0 to 65,535. A good example is box number 1,024 which controls the character at the top left of the screen.

If you type POKE1024,1 and press return the letter A appears at the top left of the C64's display over whatever happened to be there. All of these boxes can hold a number from 0 to 255 (a byte). In the same way, box 53,280 controls the border colour of the screen so for example POKE53280,4 will put colour four into the border making it go purple.

But to do this in machine code is a little more complex. First off, all numbers are stored in hex, which is base 16. We all count in base 10 (mainly due to that being the number of fingers most of us have to count with) but base 16 is a little more tricky. You count to 9 as normal, but instead of saying 10 you use the letter A.

Similarly you would use B to represent 11, C for 12 and so on until F (which is 15) when you would finally use 10 (pronounced "one zero" or "one oh"). But in hex 10 is 16 which would be incredibly confusing so from here on any hex number will have a dollar sign in front of it to say which base its in, for example \$64.

So why do we have to use hex? The benefits will become apparent later on but as it's a good habit to think in hex we will start now to get everybody used to the idea and move on to our next two commands, which are Load Decimal Accumulator, or LDA for short, and STore Accumulator, STA to its friends. LDA is machine code's equivalent of a cross between the LET and PEEK commands, so LDA #\$04 is the same as LET A=4. But LDA can also be used for reading the contents of those little memory boxes we discussed earlier, so if we were to use LDA \$C000

we would be putting whatever was in location \$C000 (box 49,152) into A. The use of the # tells our C64 that we are giving it a direct number to put into A and without the # the C64 will read what is in box 4 in the memory instead. STA is the reverse, it can put whatever is in A back into a memory box. So STA will put whatever A contains into box \$0400 (or 1,024, which is the top left of the screen remember?). A good example would be:

```
lda #$04
sta $d020
rts
```

Basically this is the same as the "POKE53280,4" command we saw earlier and shows you what I meant by the simplicity. One simple BASIC command takes two in machine code for this particular job and each step has to be followed. Now an example of the second version of LDA:

```
lda $d021
sta $d020
rts
```

Now this is slightly different. The first command reads from 53,281 (the screen colour, which is normally dark blue) and then the second puts that colour into 53,280 (the border colour again) so basically this will turn the border the same colour as the screen! So this is the same as POKE53280,PEEK(53281).

BASIC programmers will be wondering why we are always using A and not another letter for the variable. The reason is that A is not just a variable in this context, its the Accumulator. But we do have two other letters available and they are X and Y, known technically as the X and Y registers. Both can be used in a similar way to A in that:

```
ldx #$04
stx $d020
rts
```

Will do the same thing as the first example and replacing LDX with LDY and STX with STY will also work. The X and Y do have a couple of different features which will be covered in detail later, but one incredibly useful thing they can do is add or subtract 1 from their contents in a flash! This trick is done with the INX and DEX commands for the X register and INY and DEY for the Y. So how do we use them? Time for another example methinks:

```
ldx #$04
dex
stx $d020
rts
```

This looks similar to the previous example, but the result of running it would be to make the border turn cyan! What actually happens is that first X is given the number 4 to look after. Then we tell it to go down by 1 with the DEX command leaving it with 3. Finally X is told to put its number into the border colour but because it only holds a 3 now the colour is different. I bet you can't guess which colour 3 makes!

INX will have the reverse effect to DEX so replacing one with the other in the above example will cause X to end up holding a 5 so this time the border would be dark green. The Y register is exactly the same, so replacing all of the references to X with Y in the example will still work and produce the exact same result. That's all for this first instalment, but if you want to head on to the next part I'll show you what to do with an accumulator, two registers and an old washing up liquid bottle. If you have any questions about this article or machine code, email me and I'll do the business with the shooters. Erm, try to help

Continues Next Issue

Written by Jason Kelk (Published with Permission)  
<http://www.oldschool-gaming.com/>

# Commodore Free Article entitled "Bits and Pieces" Written over a 2 month period

Hi everyone, here is Luke Lynde again with another article. In keeping with tradition, I am typing this article on a Real C64 using Mini Office 2 - saving it directly to D64 disk image, and then converting it to PC text (.txt) using the Printer Emulation in WinVice! That way, I make it easy on Nigel ;) So while the text is authentic and real, the rest of the process involves the PC! Yes, PC is very handy for all the fake stuff...

Anyway, I like to do my typing this way more preferably! This article is about anything C64 that comes to mind, because so often when writing about C64, I say to myself "I should have talked about that in article." So, this text here, is about things I forget to talk about - which will be included here, as I give a bit of time (a month and more) for updated ideas and brainstorming to come through, before you see the finished product - as you will be reading it (now) upon completion.

First thing I would consider a great idea would be like a portable MP3 player that plays SID files. That would be the ultimate music hardware for me. Imagine walking around the streets with a miniature device that contains over 33000 SID tunes. It would be great. It's funny, because I read somewhere (forget where) about someone playing SID music, and everyone around them thinking their mobile phone is ringing.

Before my latest phone, it would happen to me all the time, because I guess the SID chip has that basic kind of sound like a telephone ring now and again! SID music as a ringtone, that sounds a good idea, and it is even being done now. Even some CD's I have, sometimes make me think the phone is ringing. It was funny reading that comment, because it is so true. I think the composer Smalltown Boy made that remark somewhere. Oh, Randall!

Next, I really wonder about EBay. If you don't know, it is an Internet auction site. In Australia, you usually pick up a C64 keyboard and Power Supply for around \$25AU. That's pretty good, sometimes you get it for less than that - postage ends up being around \$15AU. Whenever a C64 system is advertised with a disk drive, it goes above \$100AU. I can't understand this crap, because disk drives are so unreliable. I don't know how to service them, so I am more than happy with my 1541 emulation PC system.

There are also rare odd C64 hardware that goes for over \$100AU, once again - ridiculous as far as I am concerned. I think there is a level people reach when they become "fanatical" about it all, whether it be C64 or even something like religion. Let's face it, C64 hardware is not going to last forever - and due to it's age, it should get cheaper and cheaper every year. It is not a collectable like antiques - computers have never been classified as a collectable in the strict antique sense of the word.

I think C64 is a hobby for most of us. Enough about all that, I love C64. I am not saying C64 is crap, I just say that from antique collector theories - electronics are not really included. Excluding antique references, it is still up to debate how collectable electronics are - if at all. I tell people I collect C64 stuff, but I think it would be erroneous to say they are collectors items themselves. I mean, a piece of furniture from the 1800's is a different thing entirely!

The C64 is a great computer, that I will always love. Not because of people in the scene, as alot of them are rude and abusive towards others. Because of the games and magazine publications? Certainly. The nostalgia plays a big part in it all too, when you grow up with something - sometimes you never want to let that something go. The scene? Well I hate the arrogance of recent disk magazines, for sure, and people who think they are in control of everything C64 and pretend to be God. The scene is something that evolved over the decades, bring a mixture of joy and irritation to all involved.

Joystix? Yes, I did a PDF mag also, it is available for download from CSDB. There are 2 issues, the first issue was a really bad rush job - but the text was okay. In Issue 2 I made sure everything looked nice and aligned, and that the screenshots were with better palette, etc. I like black text on a grey background. Issue 2 was simple in design, but more polished than Issue 1. Issue 3? I don't know about that, I think I will do it, but... I need Inspiration, and until then, I will leave it for a while. If more people download Issue 2, I will do another one - if not, what is the point? Ho, hum.

C64 emulation on the PC is pretty good at the moment, I guess. But you can never compare it to real C64. I wrote an article on Emulation in Commodore Free #4. Obviously an Emulator is not using the real hardware of a C64, so it is only going to reach a certain level - and it can go no further. I still use PC Emulators, as most do, even though I have some real C64's. Comparing an Emulator to the Real thing, might be like comparing a cheap copy of the Mona Lisa painting to the original. In some ways it may look the same, but the final product and outcome is different.

Commodore Free on a .d64 file? Yes, and even compatible with 64hdd - because of no fast load routines. So read this mag also on a Real C64! Good work, Nigel! I hope the readers of this Commodore Free Publication find it as a breath of fresh air, in the wilderness comprised of many different aromas. My first and major thoughts about Commodore Free PDF is: this is informative and well presented, I hope it continues on and on! As you may know, there aren't many modern English PDFs dedicated to Commodore? Anyway, this is the end for now, I will catch you around in another article some time. CUL8R SK8R H8R! zzz

Thanks Luke Regard Commodore Free

## DiskImagery64

A drag & drop Disk Image Editor for Commodore 64 D64 Images

Written by Christian Vogelgsang  
under the GNU Public License V2  
Official Page is: <http://www.lallafa.de/blog> (DiskImagery64 Page)

### Introduction

If you are a C64-addict then you usually like to setup own disk images with new software or with new arrangements of existing collections. I often use the great "c1541" command line tool of VICE, ([www.viceteam.org](http://www.viceteam.org)) to create and modify disk images on my Mac. Working this way gets cumbersome if you arrange new images from a large collection of other images or if you want to hand-craft an image with many files from different sources. For this task I wrote DiskImagery64...

DiskImagery64 is a GUI application that allows to quickly view and edit a large set of disk images and allows to copy or move files by simple dragging them from one image to the other. Furthermore, a file browser allows access to your local file system and from there you can also drag files to an image and vice versa.

### Installation

DiskImagery64 is written with the portable Qt-library ([www.trolltech.com](http://www.trolltech.com)) and is open source under the GNU Public License V2. You can either download the source code or a compiled binary. The source code compiles on all Qt-platforms: Mac, Linux/Unix with X11, and Windows. Simply call "qmake" and "make" in the source tree to build it. Make sure to use at least Version 4.2.x of Qt.

For the low-level disk image I/O DiskImagery64 uses the "diskimage" library written by Per Olofsson. For your convenience the source code is included in this source distribution, but you can also find the source with documentation on Per's official "diskimage"-Page: <http://www.paradroid.net/diskimage/>  
diskimage - D64/D71/D81 library Copyright (c) 2003-2006, Per Olofsson All rights reserved.

### Fonts

For authentic reproduction of file names in disk images, DiskImagery64 uses two fonts: CBM and CBMShift for the unshifted and shifted commodore char set. In this distribution you find both of them as scalable TrueType fonts. Install them on your system before running DiskImagery64.

Mac users simply double-click on the CBM.dfont and CBMShift.dfont files and select "Install" in the Font Manager to install them on their systems. Windows and Linux users can use the provided CBM.ttf and CBMShift.ttf files.

The fonts were not created totally by myself. I had a scalable commodore TrueType font lying around on my hard disk and I used it as a starting point. The existing font told me its origins in the header: based on a font by Devin D. Cook with fixes from Chris McBride.

I loaded this font in the great free font editor FontForge (<http://fontforge.sf.net/>) and started fixing it again: I wanted a one-to-one petSCII mapping and that was not present. Furthermore, this requires to have

two fonts: one for unshifted and one for the shifted commodore char set. I created the two new fonts CBM and CBMShift with a "Unicode" TrueType mapping but with each petSCII character at the correct hex position. The characters were copied from the other TrueType font and missing characters were drawn by myself. I hope the mapping of all petSCII characters is correct now. For reference I have provided the editable Font Forge font (\*.sfd) files in the "fonts" directory. If you find any errors then please send me your fixes!

The fonts are NOT suitable to use them as a replacement for any other unicode font as the embedded mapping is called unicode but petSCII actually. Nevertheless, they are perfectly suited to directly print petSCII strings.

### Usage

#### 1. Startup

There are several ways to launch DiskImagery64:

- Double-click on the application icon
- Double-click on a \*.d64 image file (on Macs)
- Drag a \*.d64 image file onto the application icon

If DiskImagery64 is launched with a disk image then an Image Browser window opens and shows the directory of the image. If no image is given then the File Browser window opens.

#### 2. Image Browser

An Image Browser window shows the contents of a disk image. You can open as many image browsers as you like. You can create a new disk image with the "File/New Image" menu command. A new and empty image browser is opened. The new disk image is empty and has a default disk name and id.

"File/Open Image" opens a new image browser with an already existing image.

You can change the disk name and id by formatting the image with the "Tools/Format Disk" command. Enter a new disk name and id in the dialog. This will also erase all files on the disk.

A disk image is altered by copying files from and to the directory shown in the corresponding image browser. Simply select one or more files in one disk image and drag-and-drop them to another image. It is also possible to drag files from the File Browser to the image. Then a local file is copied onto the image.

Files on a disk image can be altered with the common "Cut", "Copy", "Paste" and "Delete" commands found in the "Edit" menu. First select one or more of them and issue a command.

If a disk image is modified the changes are at first only performed in memory. You need to save your image to disk to make the changes permanent. Either select "File/Save" to save the image with the already given name or use "File/Save as..." to save a copy or a currently unsaved image.

# Commodore Free

"File/Close" closes the current image browser. The image is not automatically saved onto disk in this case. The application warns you if you want to close an unsaved image. If the last browser in DiskImagery64 is closed then the application quits.

## 3. File Browser

The File Browser shows you a directory of your machine's file system. The browser is used as a drag source or a drop target if you want to move local files to or from a disk image.

You can browse your file system by opening and closing the tree hierarchy shown in the browser. Furthermore, the root of the shown file tree can be entered in the top line edit field. Simply enter a valid path there and press enter. Additionally, a click on the directory icon lets you select the root directory in a file dialog.

Select a single or multiple files in the file browser and drag them onto a disk image to copy them there. The file names are automatically converted from Unicode to PetSCII. Additionally, known extensions like ".prg", ".del", ... are automatically stripped for the PetSCII name and converted into the corresponding CBM file type.

The transfer of files from a disk image to the local file system works similar: Simply select one or more files in the image browser and drag them onto a directory in the file browser. Again, the file names are converted automatically and the CBM file type is added as a file extension. Invalid characters (":", "/", "\") are automatically stripped from the name.

## 4. Tools

The Tools Menu offers some tools while working with a disk image. "Format Image" allows to completely format the virtual disk. Enter the new name and disk id.

"Add Separator" is used to add a separator special file to the current disk image. A separator file is usually empty and only used in the directory listing to separate entries or to group application files. The Add Separator command has already some predefined separator styles available, but you can design your own separator (see Preferences).

## 5. Emulator

DiskImagery64 allows to call your favorite C64 Emulator to mount an opened image. Use the preferences to setup your emulator. The "Mount Image" command mounts/attaches the disk image to a virtual drive in your emulator and launches the program.

"Run Program" allows to run the selected disk entry inside your emulator. Make sure to have a single program selected. The emulator is run and the disk image name and file name on the disk image is passed as arguments.

## 6. Networking

DiskImagery64 can directly work with a real Commodore 64 if its connected via ethernet. For the C64 a popular network adapter is the RR-Net, a 10 MBit NIC mounted on the Retro-Replay cartridge (available from Individual Computers <http://ami.ga>) which is an Action Replay clone.

To set up your network, I suggest to use a cross-cable to directly connect the C=64 to your Mac. This avoids traffic from other sources that can disturb the good old 8-bitter. Furthermore, The Final Replay ROM (short: TFR) image is required to use CodeNet or NetDrive features described below. The ROM is suitable for the Retro-Replay and available at <http://www.oxyron.de/html/freplay.html>. Setup the correct IP addresses for the C64 and your Mac in the ROM with the deliverd tool and flash the image on your cartridge. Finally store both IP addresses in DiskImagery64's preferences and you are ready to go! The following network services are available in DiskImagery64:

\* CodeNet: Press F6 on the C64 with TFR running to enter CodeNet mode. In this mode the C64 waits for instructions from the network. You can fill memory, download data directly to C64 memory, jump to memory or run a program. This is all implemented in DiskImagery64 but currently only used to download a PRG and run it.

In DiskImager64 simply select a program file in a disk image or a local file and select "Network/Run Program". The file is downloaded in a second and run on the real machine. This works only for one-filers as loading other files is not supported in this mode.

\* NetDrive: TFR allows to access a "virtual" IEC network drive on device id 6. So a "LOAD "\$",6" on your C64 will load the directory from the network drive. In DiskImagery64 you can create a network drive from every disk image or selection of files (also local ones) by selecting "Network/Share Files in NetDrive".

The NetDrive allows to use multi-file-programs as a program can load data files from the virtual device later on. The program must only use the kernel load routines (no fastloader, custom load routines...) as the NetDrive works on kernel level and is bypassed by custom routines. Kernel loading is often required nowadays (e.g. on IDE64, Dreamload on MMC64,...), so many multi-file progs are already available in patched versions.

\* WarpCopy64 support (<http://www.oxyron.de/html/wc64.html>). WC64 is a server program running on the C64 and a client on a host (PC/Mac). Now you can control your C64-attached real disk drives (e.g. 1541) directly via network from your Mac. You can format a disk, verify a disk, send direct DOS commands to the drive and of course copy disk images in both directions. You can then directly transfer a real disk into a disk image in DiskImagery64 or a disk image back onto a real disk. A slow (most IEC compatible) mode taking several minutes per side and a warp mode only taking tens of seconds is available.

First of all enter CodeNet by pressing F6 in TFR. "Network/WarpCopy64/Start WarpCopy" now transfers the WC64 server program to your C64 and launches it. Make sure to have the correct IP addresses selected in preferences! You can pick "Network/WarpCopy64/Read Disk" to transfer a real disk into a disk image of DiskImagery64. If you have a disk image opened then the "Network/WarpCopy64/Write Disk" command is available and transfers the image directly onto a real disk.

## "New Games on C64"

Hi here is Luke Lynde with a look at some fairly recent game releases for the C64. Some suprising titles here, I mean - who can ever say the C64 is dead and buried? Here we go...

### **Bomberman C64** **Coder+Gfx: Skull, Music: CRD.**

Well, when I noticed this PRG on a disk image I downloaded from the net - I really did not expect it to be this good at all. I have in the past played Bomberman on the NES, and it is a game I find quite addictive - but also equally frustrating. This new Commodore 64 version from Samar Productions is absolutely brilliant. The graphics are some of the best you will ever see in a C64 game. The title screen is professional and polished, with nice music+graphix - and a scrolltext from Jammer.

While playing the game, even though I am no big fan of the Bomberman series - this is perfectly playable in all aspects, I have yet to find any flaws with anything about this game. I still struggle to get past the second level, though. Now I realise why I dislike Bomberman on the NES! However, I will continue playing it - because it is on the computer that I adore, and it looks and plays like something that maybe nobody thought the Commodore 64 could do, as well as this.

Playability: 93%  
Graphics: 94%  
Sound: 91%  
Addictiveness: 92%  
Lastability: 90%  
Overall: 96% [Gold Medal!]

### **Beam** **Code: Skate, Gfx+Msx: Hydrogen.**

Well this is a simple type of idea, I have not really seen implemented in a game before. You must balance a ball on a type of beam, while hitting another ball with your bat which is situated a top of the beam. Sound simple? Well it requires some major type of reflexes and patience to be become proficient at it. The idea is simple, and it works well.

So, there is not much to this game. It gets very frustrating, when a game can be over in a matter of seconds, but perseverance pays off.

You have 3 balls to start, and there is a continuous high score feature - which will change the better you get. The initial design reminds me of a pinball screen layout, but different. I use the joystick in port 2, but apparently you can use a mouse with this game, nice.

Playability: 65%  
Graphics: 78%  
Sound: 88%  
Addictiveness: 81%  
Lastability: 82%  
Overall: 80%

### **Deep Sea Salvage 2** **Code etc: Richard Bayliss/TND**

I really can't see the point of releasing games like this, ad infinitum, like Richard does. The only thing you could compare it to, would be like: Frogger underwater - but without even the slightest rudimentary game concept to make it even slightly interesting. Basically, collect treasure from the bottom of the ocean while avoiding the fish. Advance a level after collecting a certain amount of treasure.

This game is so bad it would not even be good for a young child to learn computer and other skills, by playing it. The music is okay, but like all Richard's games appear the same, so does the sound of his tunes.

Playability: 50%  
Graphics: 60%  
Sound: 45%  
Addictiveness: 5%  
Lastability: 1%  
Overall: 10%

### **Greenrunner** **Code etc: Aleks Eeben**

A molested centipede game with minter-esque effects. This game is for the Game Over(view) freestyle jam competition, and features 100 levels of frenetic and bizarre blasting mayhem - unparalleled in any game I have played before. On initial inspection, I expected this game to be downright terrible. After playing it, I was filled with an excitement that I rarely experience playing a video game.

The sounds are very atmospheric, extremely suited to the game - and the voice samples are awesome. The gameplay is fast and addictive, and there is enough variety to add that extra polish. The variety of colors are okay, but on some C64's they may not look so good - on mine they are fine. This is the type of game you can play, and will play again, and again. It is a brilliant little game, and comes highly recommended.

Playability: 92%  
Graphics: 82%  
Sound: 94%  
Addictiveness: 91%  
Lastability: 90%  
Overall: 90% [Sizzler!]

Well that's it for now, I hope you enjoyed reading about some new releases on C64. As suprising as I was with Greenrunner, you could have knocked me over with a feather when I loaded up Bomberman. It's moments like these, that keeps my motivation to report to the world my C64 findings - and contribute to the C64 world at large. By the way, my reviews don't contain implicit reference about the game details, it's up to you to discover them. I provide an introuction, the middle and ending is up to you. Enjoy!

# Interview with commodore128.org Lance

By Commodore Free magazine

**Q** Lance could you introduce yourself formally to our readers

**A** Lance Lyon, most of your readers would be aware of me from comp.sys.cbm & within the wider C= community over the last 20 odd years

**Q** where do you live

**A** Katoomba, in the Blue Mountains, 110km west of Sydney, NSW, Australia

**Q** Are you a Commodore Free reader :-)

**A** Yes

**Q** How did you get into Commodore machines

**A** In 1978 my parents "decided" (read: I beat them into submission with my non-stop begging!) to buy me a computer & wanted to get me a TRS-80, we were using Exidy Sorcerors in school & I wanted one of those, so they visited Sydney & couldn't get either the Radio Shack or Exidy machine & bought a PET 2001 instead. (I still want a Sorceror though.....)

**Q** Do you still actively use Commodore machines

**A** Yes, my 128D & my original PET (still chugging along) & my Amiga 1200. I also have a Commodore PC-5 (XT compatible) that is used a little as well (my original BBS machine).

**Q** what machines do you own

A part from those mentioned above, several C64's (not used, just stored) A couple of Plus4's, a several C16's, two VIC20's, 3 Amiga 600's, 5 Amiga 1000's, several Commodore PC clones, various contemporary PC's & a whole plethora of other classic computers (mostly stored). No Spectrums though :-)

**Q** are there any active commodore 128 hackers

**A** Nothing like exists for the C64 scene, but there are a few coming out of the woodwork on the forum, so the scene is starting to get a (somewhat belated) boost

**Q** Tell our reader about your website

**A** In simplest terms, a "one stop shop" for all things Commodore 128

**Q** When was the site created

**A** The forums went online in July 2006, www.commodore128.org went online in October 2006, however the site is an outgrowth of my BBs that has been online since 1987, so it has a pretty long pedigree

**Q** Why Commodore 128 what is so special

**A** There are a plethora of sites for the C64 & even though there are a few 128 related sites, there are no specific community (forum) sites, I felt it was way past time that the machine had a "proper" community site. As for what's special about the machine, well, IMHO it's the best 8 bit machine from any company & is the penultimate 8 bit Commodore (I don't count the unreleased C65 as it was never finished). Add to that that the 128 has been underexposed & underutilised for so long, it's way beyond the time when it needed its own dedicated site

**Q** "what did Commodore do wrong"

**A** My opinion ? several things, getting rid of Jack Tramiel was the first big mistake, introducing the 264 line was another mistake, next mistake was producing the 128 & crippling it by including 64 mode & then continuing to produce the 64; that should have been canned once the 128D was released. And they dropped the ball big time with the Amiga - they had the most technically superior 16 bit computer of the time - even compared to the Mac - and they lost their lead, AGA was too little, too late. In markets like here in Australia they had a huge lead over every other company, even their PC line was selling better than other 'name' brands, they threw away an enormous amount of goodwill

**Q** What Commercial games were released

**A** Not many - Infocom had a few; Beyond Zork, A Mind Forever Voyaging, Bureaucracy & Trinity; then of course there were The Last V8, Thai Boxing, Kickstart II & The rocky horror picture Show - conversions all, but with Kickstart II (as an example) you could see that had more games been released, they would have been superior to the C64 offerings. In that game there were 27 courses compared to the 8 offered in the 64 version. Elite 128 was another that sticks out too

**Q** I suppose the 128's main advantage was backward compatibility with the Commodore 64 I also think this was its draw back as many programmers just made Commodore 64 versions never utilising the full 128's power do you think this was the case

**A** Absolutely, see my earlier comment about C64 compatibility

**Q** What is the 128's killer application

**A** That's a hard one, GEOS 128 springs to mind (80 column mode), and practically any of the 80 column business software (Timeworks was always a favourite label of mine)

**Q** Although the retro scene seems fixated on the C64 are there any new titles in development for the 128

**A** In development? Well, probably not, but one of the things that I've noticed since starting the site is that people are starting to program the 128 after having been in hiatus for many years, one of our members for example was spurred to re-write VICE's 2MHz mode, plus with the coding comp we're currently running (small to start with), I'm hoping that people will start developing for the machine. Remember that anything the the C64 can do the 128 can do twice as well & twice as fast too! <grin>

**Q** Thanks for your time is there anything you would like to add

**A** I'd like to thank the core group of my members who have put an enormous amount of their own time & effort into growing the site. Also, keep up The excellent work with the mag!

cheers, Lance

// <http://www.commodore128.org>  
Commodore 128 forums & more! //

# Commodore Free



## introduction to the c128

The Commodore 128 (aka C128) was Commodore Business Machines final 8 bit computer (the C65 doesn't count as it was never completed & never commercially released). Introduced at the January 1985 CES, it was the follow-up to the Commodore 64 & was a vastly expanded & more powerful successor to that computer.

## The C128 features :

CP/M mode via a secondary Zilog Z80 CPU in 40 & 80 column modes  
Enhanced & faster 6510 (the 8502) running at 2MHz  
Native 40 & 80 column modes in two speeds (1 & 2Mhz)  
Near 100% compatible C64 mode  
Basic 7.0 - more powerful & flexible than the limited C64's Basic 2.0  
Numeric keypad  
Burst mode disk access

128k of RAM with more available for programmers due to an MMU

The 128 was released in three models - the first looked like the a larger 64C (& the design was repeated in the later Amiga 500, 600 & 1200) - the "flat" 128. The second model was a rather nifty plastic box with a carry handle & slide in keyboard that allowed the machine to become a "luggable" (the 128D) & the last model was the metal cased cost-reduced variant (the 128DCR). The 128DCR also came with a 64k VDC as opposed to the earlier models 16k chip (they could be retrofitted).

Although the computer sold in excess of two million units during its lifespan, native software was thin on the ground due to the included 64 mode - developers didn't bother creating software that took advantage of the native modes. However, there are several fine business packages produced for the 80 column mode & due to the CP/M compatibility (& it's ability to handle multiple CP/M disk formats) a large library of software was available right from the start.

## about our site AND Mission statement :

To provide an active & vibrant community devoted to keeping the Commodore 128 alive (hence the name of the site) into the future.

To gather together in one place as much relevant material as possible & to provide a "one stop shop" for the C128 community.

To foster a friendly & sharing Commodore 128 community.

To provide accurate information & discussions about the Commodore 128.

To promote an active development scene for the Commodore 128.

To preserve & distribute Commodore 128 related software, manuals & code.

And above all - to have a good time while we're doing it!

## Commodore 128 hardware information

The C128's hardware basically remained the same across all three models. It consisted of :

128kb of RAM (externally expandable via an REU)  
8563 VDC chip driving the 80 column RGB display with either 16k of RAM or 64k of RAM in later models (& easily retro-fitted) - in the 128DCR this was replaced with an 8568

VT-100 style keyboard with a numeric keypad  
8502 CPU - an expanded 6502 which could run in either 1MHz mode or 2MHz mode (the 40 column screen turned off at the higher speed)  
Secondary Zilog Z80 CPU that controlled the startup of the computer & allowed it to run CP/M in either 40 or 80 column mode. Although this was a 4MHz chip, it was constrained to 2MHz due to the requirements of interfacing with the 8502

A reset button

MMU bank switching chip  
In the two later models, the keyboard was detachable  
The 128D models also contained a built in 1571 disk drive  
2 KB 4-bit dedicated color RAM for the VIC-II II  
MOS 8580 SID chip for sound - this was the cost reduced version, earlier 128's had the 6581 Sid as used in the C64

RGBI video output allowing the computer to connect to a standard CGA monitor but with an additional monochrome composite signal as well

There are three basic models of the Commodore 128

The original "flat" version, broadly similar in looks to the 64C, Amiga 500, 600 & 1200. This model came equipped with the 16k VDC & had no internal disk drive.

The Commodore 128D was an attempt to produce a more "business-like" looking PC and was in a low profile plastic box with an internal 1571, stowable keyboard & a retractable handle that made the computer a luggable. It still had the 16k VDC. This PC also had an internal fan.

The Commodore 128DCR was the final version released. The CR stood for "cost reduced". This release came in a sturdy metal case, dispensed with the fan, had a reduced number of chip components & had the 64k VDC.

**Text Reprinted from the website**  
<http://www.commodore128.org/index.html>

# Commodore Preservation Project

<http://rittwage.com/c64pp>

Welcome to the Commodore 64 (C64) Preservation Project. The main goal of this project is to archive pristine versions of original Commodore 64 software, including copy protection. A secondary goal will benefit of this will be to catalog and document all the different copy protection methods used. This information will be used to improve emulation, as well as remastering software onto disks for you to enjoy on the real thing. These goals are much like that of C.A.P.S. project for the Amiga. Of course, to reach these goals, we need your help. This software exists only on magnetic media from the 1980's, and as such has been disappearing into attics, yard sales, and landfills for almost 20 years. Floppy disks were also never made to last a lifetime, as I have found in purchasing them in auctions. Even disks that were stored and cared for by their owners all these years have pretty high failure rate, some where the magnetic material actually wipes right off, making the drive heads filthy. You can help preserve them for our own and future generations in a number of ways.

\* If you have a 1541 or 1571 disk drive and a XEP, XAP, or XMP cable (serial/parallel combination), we can send you the latest mastering software. You can then send us the resulting image and statistical data for analysis and inclusion in the archive.

\* We will happily pay for postage to get any original disks you have to us! We will promptly image them and send them back to you.

\* If you don't have the cabling or don't wish to send us your originals, you can make us nibbled copies of the originals by whatever the best means you have and send us those. We will pay for postage, as well as return these disks to you, of course. We can sometimes reconstruct the protection onto the image by looking at what it checks for, plus they can be used to verify the original image if nothing else.

## Copy Protection Methods

Many different protection methods were developed over the years in a cat-and-mouse game with those that wanted to make a copy of their own (or a friend's) disk. The methods steadily increased in complexity, but whether or not the protection was copyable or not, a way was always found to make a working backup. Descriptions of the drive hardware itself, as well as many of the methods that different companies employed to keep the disks from being copied are found here.

---

## Background

The Commodore 1541 disk drive is a stand-alone computer that talks to the C64 through a somewhat slow serial port. It is based on similar technology to the C64 itself, employing a 6502 CPU, two 6522 VIA I/O chips, and only 2k of memory (the limitation of which will be discussed later). It came with either an Alps or a Newtrionics 5.25" double-density floppy mechanism, both of which are functionally equivalent. This mechanism requires double-density 48 track-per-inch 5.25" magnetically-coated Mylar disks.

## Tracks

Tracks on the disk are organized as concentric circles, and the drive's stepper motor can stop at 84

different locations (tracks) on a disk. However, the read/write head on the drive is too wide to use each one separately, so every other track is skipped for a total of 42 theoretical tracks. The common terminology for the step in between each track is a "half-track" and a specific track would be referred to as (for example) "35.5" instead of the actual track (which would be 71). Commodore limited use to only the first 35 tracks in their standard DOS, but commercial software isn't limited by this. Most floppy media is rated to use 40 tracks, and the drives usually have no trouble reading out to track 41, although some will bump and not get past 40. Most software does not use any track past 35 except for copy protection, but alternative DOS systems like Speed-DOS used all 40 tracks in it's own DOS implementation.

CBM drives have no way in hardware to detect which track it is on, or where it is on any particular track. The software must handle these functions which leads to many of the more creative styles of copy protection. 5.25" disks contain an "index hole" which is the little hole you see diagonal to the hub ring on your disks. If you spin the disk around in it's shell, you'll see that there is a hole in that, too. Other drive manufacturers used an optical sensor to detect when this hole passed by, which would signal the start of a track. Commodore didn't implement this, so we have to use marks that are written to the disk during formatting. Later copy protection implementations took advantage of this because the devices used to master original software are not typically a 1541. They could master the disk using the index hole so it was lined up perfectly. As far as knowing which track we are on, the only way to tell is again in software. Each sector on a track has a header that contains (among other things) it's track number. DOS will read this whenever it needs to access a track so it knows how many times to step the motor to get there. The famous "drive-knock" is the software hack that Commodore employed to reset the drive to track 0 when we couldn't find any sectors. In this case, the motor is stepped out as far as it can go until it physically stops (causing the knocking noise) which guarantees it's at track 0. Unfortunately, this behavior is what is blamed for the terrible drive alignment problems of the early mechanisms.

## Sectoring

Tracks are further divided into sectors, which are sections of each track divided by the forementioned software-generated sync marks. The drive motor spins at 300rpm and can store data at 4 different bit densities (essentially 4 different clock speed rates of the read/write hardware). The different densities are needed because being round and the motor running at a constant speed, the disk surface travels over the head at different speeds depending on whether the drive is accessing the outermost or innermost tracks. Since the surface is moving faster on the outermost tracks, they can store more data, so they use the highest density setting. Consequently, the innermost tracks use the slowest density setting. Because it's recording at a higher density, of course more sectors are stored on the outer tracks, and fewer on the inner tracks. There is nothing stopping the hardware from reading/writing at the highest density across the entire disk surface, but it isn't generally done due to media reliability, and slight speed differences

# Commodore Free

between drives. The media itself is only rated for a certain bitrate at a certain speed.

## Drive Internals

The drive head, at its lowest level, can only detect and create polarity changes in the magnetic flux on the surface of the disk, which is interpreted by the drive as a binary 0 (no change) or a 1 (flux polarity change). Because of the specifics of the hardware used to access this magnetic flux, the bits stored on the disk have a couple of limitations (or "features"). This flux detection circuit feeds dual 4-bit "shifters" which convert the bits (detected from the flux changes) into a whole byte (8 bits), much like a serial port does. If this circuit detects ten '1' bits in a row, it knows that it has read a "sync". This sync is what tells the software that it's about to get a series of bytes representing either a sector header or sector data right after the sync ends. An anomaly of this circuit is that if it runs across three or more '0' bits in a row, it will clock in an extra '1' bit on each 4-bit shifter that doesn't really exist on the disk. When this happens, the track loses framing, meaning the data is bitshifted after this due to our new phantom bit. It will shift a new phantom '1' into the LSB once in a while (and wrap it around) according to drive speed and other environmental factors until the next real '1' bit is seen. This data is "random" in the sense that the data is never the same twice due to drive speed (and humidity, temperature) fluctuations. A good example is when you try to read a new, unformatted disk. If you try this, you'll get "random" data caused by this anomaly in the drive hardware because there are no flux changes on a blank disk. To the 1541 hardware, this is all '0' bits. Because of this weakness of the hardware, all data is stored on the disk in an encoding system called "GCR". This encoding assures that there will never be more than two '0' or '1' bits in a row in the data itself. This "feature" of the '0' bits is of course taken advantage of in some later copy protection schemes. (Rapidlok, Mindscape, and Datasoft, for example ;)

Differences between a "fast copier" and a "nibbler" Copy programs have been referred to by these two different types since the first nibblers came out on the Apple II. A little more background is required in how the disk is structured to explain the difference. A track is broken up into a number of sectors, which are further broken up into a header and data portion, and each of these has a filler "gap" at the end to give the drive software time to change modes. The actual GCR data is structured like this: SYNC (at least 10 bits of '1' in a row) header0 (8 bytes in DOS) header gap (usually 5 filler bytes, longer for nibbled copies) SYNC sector data 0 (CBM DOS uses 260 bytes) tail gap (usually 5 filler bytes, longer for nibbled copies) SYNC header1... (and on until the end of the track)

## Fast copiers

A "fast copier" reads the source disk on a sector-by-sector, normal CBM DOS level, and recreates them onto a new formatted disk. This method produces a "clean" copy of the disk, but does not copy any protection since it will ignore any sectors with errors, or any sectors that don't conform to the format explained above. It was really only used when we weren't going to try to copy the disk protection, but rather apply a "parameter" to the disk. A "parameter" is nothing more than a patch program that was run on the backup disk to remove the check for copy protection. Fast Hack'em, Kracker Jax, and later Maverick were all very popular copy programs that utilized parameters.

## Nibblers

A "nibbler" works at a lower level and will duplicate the actual headers and data, whether or not they are in CBM DOS format. It does have limitations, though. First, they don't duplicate the header or tail gaps or the original sync length, but rather creates its own, so protections that hide in the gaps or count sync lengths won't be duplicated. Secondly, the drive only has 2k of RAM, but a whole track is up to 8k, so it must be broken down and read in several passes, then stitched together on the destination disk. Protections that exploited this would use a sector or track larger than would fit in RAM at once, making it impossible to copy. Later, there were hardware solutions to this, such as extra drive RAM and parallel ports.

---

## Protection Methods

### Intentional Disk Errors

This protection consists of a disk error purposefully placed on a sector or an entire track, then its existence is checked for in the program. If the error isn't found, it's known it's a copy. In the beginning, there were no copy programs except Jim Butterfield's "COPY/ALL" that came on the 1541 test/demo disk. This program could not reproduce these errors (and even if it could, it took \*FOREVER\* to copy even regular disks). Like most protection, though, it only foiled casual copiers. Early, clever crackers figured out how to scan an original disk for these and the methods for recreating the errors were not difficult. Soon after, programs like "Copy II/64" came out that could effortlessly detect and reproduce these errors which marked the end of this era. Most publishers of disk software used this method before late 1984.

### Fat Tracks

This protection involves placing a track on the disk that is actually 2 tracks (4 half-tracks) wide. This has two effects.

- 1) The second track is invalid because Commodore DOS stores the track number as part of the track header, which is now for the wrong (previous) track.
- 2) The tracks are perfectly aligned on the disk since they were written at the same time.

The protection is checked by the program by stepping the head up and down between these 2 tracks and the halftrack between them while reading the whole time. Normally, a half-track will contain a garbled combination of its neighboring tracks. However, since these two tracks are now identical, this track should contain the same data as both of its neighbors. If it doesn't, it knows it's a copy. Electronic Arts used this protection from about 1984-1985 and Activision used a tougher variant of it called "XEMAG 2.0" for many years. These protections stand out from most of the early protections because they could \*not\* be copied 100% reliably even with \*any\* nibbler. As mentioned earlier, the 1541 has no way to know where it is on a track. Therefore, it has no way to know where it is in comparison to any other tracks on the disk. For this reason, it is impossible to reliably write these tracks out perfectly aligned with a 1541 drive. This was an idea that was greatly expanded on in future protections, but it still stands the test of time and still can't be duplicated reliably with the best copiers that ever existed. It is of note that it \*is\* possible to "unreliably" copy these, as you can just try over and over again copying just these two tracks until you happen upon "close enough" alignment for it to work. The EA protection in particular relies less on the exact track alignment, so it is less difficult to copy

# Commodore Free

this way. NOTE: Slowing your disk drive motor down to 298.5rpm or less seems to allow you to load a copy of any fat-track protected disk remastered with MNIB.

## Half Tracks

Along the same vein as fat-tracks, it was of course possible to read and write to the usually skipped "half-tracks" instead of the normal tracks, if you weren't worried about storing any data on the normal tracks. This foiled many copiers because they didn't search for and copy these half-tracks by default, so they instead got garbled, invalid mixes of the "real" data when trying to read the tracks as if they were normal. Copiers later came out that could scan for and detect these and copy them properly. I have not yet run into any disks which are known to use halftracks.

## Extra Tracks

This protection simply relies on using disk tracks beyond 35. Normally, copy programs default to copying only tracks 1-35, so these are missed. I am not sure why the programmers didn't extend this by default, except maybe some would lock up reading unformatted tracks, which these tracks usually are. Sometimes these tracks contained actual DOS-formatted data, but usually they were just "key" tracks, or the second part of a "fat" track from track 35. They could sometimes be nibbled, other times they could not, because they were combined with other protections.

## Changed Bitrates

As mentioned above, the bitrate is normally higher on the outer tracks and lower on the inner tracks. Some disks were protected by using a non-standard bitrate on different tracks. Copiers later came out that would scan for the correct density and these disks could be copied that way.

## Header and Tail Gaps

As mentioned above, when a disk is copied with either a fast copier or a nibbler, the gap data is not copied directly. Copiers would commonly just fill this space with 0x55 bytes. Gaps are "inert" bytes, meaning they aren't normally used as data, but just a space filler that is ignored by DOS. There are some protections that check for specific bytes here, or even the length of the gaps, and know they're a copy if it's different. This protection can be copied with hardware solutions that either extend the RAM in the drive to 8k or add a parallel port so it can read and write the entire track in one pass.

## Long Sectors

As mentioned above, the drive has only 2k and can't fit an entire track in RAM, so it must be broken down into 4 parts and stitched together on the destination disk. If you make a sector longer than will fit in RAM, it can't be copied with a normal 1541. This can be copied with hardware solutions that either extend the RAM in the drive to 8k or add a parallel port so it can read and write the entire track in one pass.

## Custom Formats

These usually depend on the long sectors trick above to be successful, but they also entail changing the way that GCR is interpreted in a custom DOS. If you write your own DOS, you can use your own disk format entirely, ignoring common sync, gap, header, and data conventions altogether and use whatever you want. You only have to adhere to the hardware limitations of sync and no more than two '0' bits in a row, but even that didn't stop later innovations. :)

## Long Tracks

Normal drives spin at ~300rpm and can "write" data at the highest density up to about 7700 bytes per disk track. However, the drive can "read" more than that, within some margin of error depending on motor speed. Some protection took advantage of this and put more data on the track than could be written back out at the normal speed, making it difficult to copy. Some copiers would "trim" the data in as non-destructive way as possible (reducing sync length and gaps) and it was also possible to slow down the disk drive to foil this, but usually these disk relied on a combination of other protections as well. A good example of this is Harald Seeley's "V-MAX!".

## Sync Counting

A sync mark must be at least ten '1' bits long to signal the drive hardware that it's about to see GCR data. However, sync marks are usually much longer than this (40 bits) and have no limit to their length. When copying a disk with a software nibbler, sync has to be reproduced rather than copied, since it is more like a "signal" than actual data stored on the disk exactly. For this reason, the length of a sync is also somewhat dependent on drive speed and will vary a little. Some protections count on this and measure the lengths of the sync, or the occurrences of sync on a track, to see if they match, and know they're a copy if they don't. Microprose used this method on some disks, and in fact "trimming" the sync will fail the protection checks.

## Track Synchronization

We mentioned above that the drive doesn't use the index hole, so lining up the tracks on the disk is nearly impossible on the 1541. Since disks weren't usually mastered with a 1541, but rather with machines that could do track sync based on the index hole, protections took advantage of this in different ways. They would move to a halftrack or the next track in the middle of reading to see if the data that should be there exists. If it doesn't, it knows it's a copy.

## Weak Bits/Unformatted

We mentioned before that if the drive gets more than two '0' bits in a row, it clocks in a random '1' bit occasionally. Single occurrences of this sequence won't always make it lose framing, it will return a "random" byte, but more than a couple in a row will result in the drive losing framing and reading these "random" bytes until it sees the next real '1' bit. Most nibblers can't duplicate this, so protections checked a byte purposefully on the disk somewhere and if it reads the same thing over and over, it knows it's a copy. Later copy programs did try to reproduce this--Burst Nibbler detects and writes a 0x01 byte in their place, which is enough to fool most software into working. It is also worth noting that this is the same thing as unformatted tracks, so you'll see this a lot when imaging in new disks. Since the disks aren't usually duplicated on a 1541, and the tracks aren't always as long as they could be, the machine leaves unformatted data on the end of all the tracks, causing a lot of software to appear to have these on every track, and making it difficult to verify the tracks from disk to disk, since the bytes will be different from sample to sample. I have several disks from Synapse that have unformatted parts on all the tracks, and none of the tracks above 25 are formatted at all! I've seen this used in Rapidlok (all titles), later Microprose disks, and Datasoft (Bruce Lee, The Goonies, Mr. Do!)

## Signature (key) Tracks

# Commodore Free

This protection is seen from time to time and revolves around a track being mastered in a non-standard way. It can be all sync '1' bits (which we call a 'sync-killer' track), unformatted (or all '0' bits), filled completely with some other byte, or filled a special byte sequence like EA's PirateSlayer, or a combination of all of them. These tracks generally foil most all copy programs because they either cannot handle all or long sequences of '1' or all '0' bits, or they are just read incorrectly/out of framing. We can remaster most all of these with our development version of 'mnib'. :)

## No SYNC

The meanest protection that came about used tracks on the disk that had \*no\* sync marks. Most copiers cannot read these tracks and most times think they're empty. The later Vorpil uses this. We believe there is a way to find the track cycle in these and reproduce them, as Maverick had custom copiers for these titles. It needs disassembled and examined.

## Frequently Asked Questions

Q: Where do I download the games from?

A. You can get back images for games you have contributed, games for which can be proven you own the originals, or games that the publisher has said can be redistributed. We cannot legally redistribute the games, as we are not the copyright holder. Most of the games we have preserved already have existed on the internet in other forms for over a decade, but still we cannot open up any sort of legal implications.

Q: How do I use the images I get back?

A. The images are in "NIB" format. This format contains a little more information about the media than a G64 file does that helps the remastering process. In most cases, the image can be converted to G64 and either VICE or CCS64 will run them. There are some tougher protections that won't run due to the inexact way 1541 emulators work. There are also some images that work on the emulator that can't be put back on a real disk.

Q: What equipment and/or cables do I need to dump my disks?

A. You need a 1541, 1541-II, or 1571 disk drive and an XAP1541, XEP1541, or XMP1541 cable. This cable has both the standard IEC connector as well as the direct parallel connection to the VIA chip inside your disk drive. You can build them yourself using the guides on Joe Forster's Pages or purchase them from his shop. Note with the latter that you still have to solder in the parallel cables to the VIA chip inside your drive, but the cables are top quality and you'll save a lot of time.

Q: You already have all the games I own in your database. Should I bother to redump them?

A. Yes! There are many different reasons.

\* Not all disks even with the same publisher, region, and disk label have the same protection. The disk we have might be a different version, from a different region, from a different mastering run, or from an alternative publisher (like a budget release). The disk we have might even be bad and we don't know it yet.

\* Commodore's checksum for DOS sectors is not strong (it's a simple rolling XOR) and prone to flag a sector as good when it is not, and heavily protected disks are even worse since they don't even have these checksums. This can happen on old disks that have deteriorated and caused some of the bits to go

"weak". The only way we have to verify that an image is "good" is to compare it to other dumps of the same exact game. We are working on a tool to create CRC32 checksums of just the actual data on the disk without regard to sync, gaps, and weak runs. This will help verify and compare images going forward.

## MNIB Development

MNIB - Commodore 1541/1571 disk image nibbler  
(C) 2004-07 Peter Rittwage, Markus Brenner, and friends (C) 2000-04 Markus Brenner.

1/7/2007 - Released version 0.45.1 Added halftrack support as the default behavior for NB2 images.

1/7/2007 - Released version 0.45 Added support for new NB2 format, specified by extension. (i.e. 'mnib test.nb2') This format contains 16 samples of each track, 4 at each possible density. Processing tools are forthcoming. Noticed the new writing tool 'pwrite' was missing from the web distribution. This has been added to the binary releases. Density and 'killer' track detection has been improved/changed. (again) Interactive mode should be working 100% now. Command line interface to this is clunky, but functional. Allows 9 second backups. :)

The C64PP project originally started when I took Markus Brenner's GPL source for MNIB and figured out how to add a routine to make it write the disk images back out to real disks. This part turned out to be easy, but I opened up a can of worms that has taken lots of time to perfect. :) Since then, I have actively developed the program, adding in various routines to handle foreign disk formats and protections, as well as improving the read routines and conversion tools. At this point, we can read and reproduce pretty much any disk that was/is possible with 1541 drive hardware. Most converted protections will work in the emulators as well, with a few exceptions. I have been handing out the test versions over time to contributors (of time, advice, disks, and images) and at this point it is pretty stable. Markus as well as many other people from around the world have kindly contributed their time and expertise to expand this into a powerful tool. I would never have gotten out of DOS and into Windows/Linux without Tim Schurmann, Nate Lawson, and Spiro Trikaliotis.

The latest sources are always available via anonymous CVS and can be fetched with CYGWIN as follows: set  
CVSROOT=:pserver:anonymous@rittwage.com:/root  
cvs login (password is 'mnib') mkdir mnib36 cvs  
checkout mnib36

Special thanks for code, bug reports, and suggestions: Tim Schurmann, Spiro Trikaliotis, Nate Lawson, Jerry Kurtz, Matt Larsen, Mat Allen (Mayhem/UK), Wolfgang Moser (Womo), Quader, Jani, Curt Coder, Roger Cousin, Christian Hedemann, Steppe, Victor Castro (and many more I'm sure I've forgotten...)

The MNIB DOS version contains code originally based on CBM4Linux from these authors:  
Michael Klein, Andreas Boose

**All content copyright (c) 1971-2061 by Peter Rittwage. All programs mentioned are copyrighted by their respective owners**

Text reprinted with thanks from the website with the Permission of Peter Rittwage

# Interview with Pete Rittwage

C64 Preservation Project

Q - Please Introduce yourself to our Reader, Where do you live, What do you do for a living

A - I am a Senior Network Engineer from Augusta, Georgia, USA that grew up in the 8-bit computing era during the reign of the Commodore 64. My original goal was probably that of a lot of retro users.

I wanted to recollect all the games I had for the C64 when I was a kid, which for most part were games that I either copied (or later cracked) from friends in the Northeastern Ohio (USA) area.

It always bothered me when I got a copy of a game where the text was edited with silly "BROKEN BY" or "CRACKED BY" phrases, so I would always remove them and put back the original text. Later, we were inundated with the "intro demo" which usually saw the loss of the title screens and original loading graphics. I sought to remove those as well.

From there it expanded into collecting original disks. After discovering Markus Brenner's MNIB project, I decided it was better to just try to collect all the games as they were originally distributed. That way you avoid any errors or missed protection checks that the 'cracker' left behind. This opened up into the can of worms you see in the software and on the site today.

Q - What introduced you to Commodore

A - I received a Vic-20 for my birthday in 1983 or so. After the C64 came out, then VIC was marked down to less than \$100, so my parents were able to afford to get me one. I had no tape drive for about 6 months until I got one for Christmas. During that time, I would type in programs from magazines and debug all my typos, which is how I learned programming.

But of course when the computer was turned off, they were gone. :( I purchased a C64, 1541, and 1525 in 1985 or so (used) from a friend of my parents who was "upgrading" to an IBM PC. I moved to Georgia a couple of years later, sold all my C64 equipment, then bought an Amiga 500 in about 1988. Later (1991) I went to work for a Commodore dealer and was surprised how many people still used the C64. I had a almost a full time job repairing them for a year or two.

Q - What Commodore machines do you own

A - Pretty much anything they made that was in color. VIC-20, C16, a couple of +4's, an original badge breadbox (S/N 25000 or so), many rainbow badge 64's, a pristine SX-64, a couple 64C's, a couple 128's, Amiga 600, Amiga1200, Amiga 1000, a couple of Amiga 500's, a dozen or so 1541's, 1571's, and 1541-II's. I have plenty of parts to last my whole life. And a couple thousand original disks at this point, about 1/3 with boxes. Nothing sealed, of course. :)

Q - where did Commodore go wrong

A - Well, I worked for a CBM dealer when they went under, and the owner of the store owned CBM stock (and actually went to the last stockholder meeting in the Bahamas). It became clear that the leadership at CBM did not want to invest the money in becoming a

mainstream computer company in the 90's. So they just drained what they could until it collapsed. I could write a whole essay on their mistakes, but this was the ultimate downfall in the end.

Q - Commodore Preservation Society - please tell our reader the main aim of this website

A - To collect and document all floppy disks released for Commodore computers.

Q - I have printed the FAQ file from the site but you must receive some really dumb questions can you Enlighten us

A - Actually, you'd be surprised. Most people that still use C64's or play as a hobby are quite computer literate, so I don't get many dumb questions at all.

Q - would it ever be possible to recreate a duplicate disk from a Nib Copy

A - Most images, yes, you already can. Some won't work due to the limitations of the drive hardware, but some of that can be worked around. Probably 97% of images can be remastered without problems.

Q - Please tell our reader about the process and the NIB copy format

A - The NIB "format" is really nothing more than about a cycle and a half of raw GCR data as it's seen passing over the drive heads on each track. The parallel connection inside the drive sends this GCR data out in real time to the PC which stores it in this simple file format.

This file is then analyzed, the track cycles are determined, and then converted to G64 (or D64 if appropriate) for use in emulation. I usually briefly analyze the copy protection (manually) and make note of it. I then use other tools I've written to compare the data to other known images of the same title to see if it's unique in some way.

Q - What will happen to the game copies if the website closes

A - There are many contributors to the project that have backup copies of everything, so if I disappear hopefully someone will carry the torch.

Q - How can our reader help your project

If you have any original disks and have access to the hardware to read them (XAP, XMP, or XEP combo cable) then feel free to contact me to contribute images. If you don't have the hardware, then you can send the disks to me or other project members that may be in a country nearer to you. I can even pay for postage both ways, if you wish.

Q - Do you copy the documentation and disk covers?

No, but there is a site called "The SixtyFour Originals Database" that does this. It's too much for me to handle all of that too, so it's good to have different projects.

## Commodore Free

Q - What about tape preservation or have you deliberately left this to others

A - There are a couple of other projects doing this as well. Tapes aren't plentiful in the USA (most software was on disk here) so I leave that to people with more expertise in tape preservation.

Q - What is the worst copy protection scheme you have come across and has any scheme had you unable to produce a disk copy

A -Big Five's "Bounty Bob Strikes Back" is written in a format like "Spiradisk" (from the Apple II). 1/2 of each track is written, and the head bumped 1/2 a track in the middle, then repeated again and again For about 12 tracks. This is very difficult to copy or write back out because of timing issues. No copy program ever succeeded in making an exact duplicate (without cracking it).

There is a special copy program for the SuperCard+ that is supposed to work, but we've tried it on two different originals with 3 different drives and never gotten it to produce a working copy. The other very difficult ones are mostly due to track skew. On the 1541, there is no way to accurately align tracks to one another without adding an index hole sensor.

Some protections (like Rapidlok) check timing between certain patterns on neighboring tracks and fail if they are even a bit off For this reason, these games don't completely work in emulators, as they don't yet emulate track skew or exact stepper motor timings. Long tracks, such as those used in V-MAX or the newer Vorpak-protected disks (like California Games, etc) are no problem for emulators, but The disk drive motor must be slowed way down to fit all the data on the disk when remastering them.

Q - So where does it go from here, is this just to Continue collecting disk images

A - That's pretty much it. Nobody knows how many games were released on disk for the 64, so we'll never know when we're done. We're about to reach about 2,000 unique titles, and there are about 3500 disk sides that make up those titles.

Also, there are many games which were released in different versions, different regions (PAL/NTSC and language differences) as well as re-releases on budget labels, etc. It's a big job. I probably spend a between 3-10 hours a week on it just going through images and disks trying to keep up with our kind contributors.

Q - Can a user create an image and send it to you for the Database if so how would they create the images

You need a 1541 (preferably an original model as they are more reliable), a PC (running DOS, Windows, or Linux) and an XAP, XMP, or XEP combo cable. This cable has the extra parallel connection soldered into the drive like the 'Datel Burstnibbler' or '21 Second Backup' used back in the old days.

Q - Many of the Commodore games can be downloaded from various sites, these are normally cracked with intros and cheats, although this does allow in many cases the games to be played with newer or extra hardware for example run from Cmd products, like hard disk and Ram - its nice to run the real version of the software, the problem is how long

will my disk drive last constantly bumping the heads about,

A - Only the earliest copy protections that check for an intentional disk error have this problem. The old original drive mechanism with the pull-down (as opposed to the twist-down) are less reliable and should be avoided for these earlier disks.

I've got a 1541 drive with the twist-down door that I've used to read probably 10,000 images with and it still works fine, so the hardware is pretty reliable if you have a good drive. If it breaks, though, there are plenty left 'in the wild' for years to come. :)

Pete Rittwage  
C64 Preservation Project  
<http://c64preservation.com>

# Interview with Andrew Fisher

Freelance Writer Musician and Commodore Programmer

Q. Please tell our readers a little about yourself.

A. My name is Andrew Fisher, I'm 32 and a freelance writer. I live in Cambridge, England.

Q. Tell our readers about your computer memories.

A. My earliest computer experiences would be playing games at a friend's house on his birthday, on his VIC-20 and BBC (he was a lucky kid). As we were living near Great Yarmouth at the time, I also started to play arcade machines like Gorf and Pole Position.

Q. When did you first receive your Commodore machine?

A. It was October 1985, in a package with two games and a Datasette. Sadly the powerpack was faulty, meaning it was several weeks before we got it back working and had the chance to play the games - Arcadia 64 (with dreadful flickering sprites, and it took 18 minutes to load!) and 3D Time Trek (without a joystick).

Q. What machines do you own?

A. I still have that original C64, a couple of C64C's and a C128. Plus a Spectrum +3, NES, SNES, N64, Gamecube, PS1, PS2, Xbox and a JAMMA arcade cabinet. And a Gameboy Advance, Gameboy Advance SP and Nintendo DS Lite. As you can see I own quite a few machines...

Q. Tell our readers about your writing work on magazines.

A. Well, back in 1993 I started writing a technical column for Commodore Force as "Professor Brian Strain", or The Mighty Brian. That was a caricature based on a real photo of me. When Commodore Force closed, I wrote for its rival Commodore Format for a few issues, mostly on GEOS and user groups. Then I spent most of the 1990's writing for fanzines like Commodore Scene, before Retro Gamer appeared...

Q. You recently worked on Retro Gamer, tell our readers a little about the articles you wrote.

A. Well, the first one was sent in on spec, and printed so I guess you could say I was lucky! That was THE RETRO RYDER CUP, looking at golf games through the years. On a Commodore theme I wrote about cartridges and the Commodore 128. I contributed to a couple of other articles, and had more underway.

Q. Retro Gamer closed, were you left unpaid for some of the articles?

A. Yes, I was owed money for one article.

Q. Did you look to recover your money?

A. Administration is a difficult process (which I've unfortunately been dragged into several times with my magazine work) and there was no chance of the freelancers being paid back with the banks and other companies owed so much. Instead, the freelancers got together to create the Retro Survival CD, featuring some of the articles that would have been in the

issue 19 that was never printed. The readers were so enthusiastic, they pre-ordered enough to cover the cost of getting the CD produced, so we have been able to pay money back to all the writers. If you are interested in Retro Gamer and want to see some interesting articles, the CD (which is compatible with PC or Mac) is available from: [www.retrosurvival.co.uk](http://www.retrosurvival.co.uk)

Q. Retro Gamer was purchased by another publisher, have you written for this publisher?

A. Yes, Imagine Publishing stepped in with new editor Darran Jones taking the helm (he had previously worked on gamesTM's retro section). I had work in the first few issues - interviews

with Shaun Southern and Andrew Morris, plus C64 artist Stephen Robertson (known as SIR). Then there was my biggest article ever, a look back at the history of Gremlin Graphics. Hopefully I will have more in the magazine soon.

Q. You work freelance, has this always been the case?

A. Yes, the freedom is quite nice but at the same time there are downsides - like not being paid, tight deadlines and trying to find an outlet for the work.

Q. If our readers wanted to break into writing for a career, what would you suggest?

A. The first step would be to write as much as you can - if you want to get into games reviews, find a website that is looking for reviewers and volunteer. (I'm currently writing reviews for a site called Console Obsession, for example.) Once you've built up a few contacts like that and got some good feedback on what you write, try approaching editors with ideas. Rejection is tough at first, but keep at it. The most important thing is to make sure you read a lot and write a lot; over time your work will get better.



## Commodore Free

Q. Do you have any favourite software?

A. In terms of games, my favourites on the C64 include The Sentinel, Wizball, Paratroid, Impossible Mission and Pogo Joe. On other machines I tend to play a lot of sports games, like the Tony Hawk series and American Football games. Just recently I've been hooked on the DS puzzle game Meteos, Nintendogs and the two Lego Star Wars games for Xbox.

Q. Tell us about POL, how were you involved and what POL is?

A. People of Liberty is a group formed by Joerg Droege, a German C64 fan. He was looking for contacts to help him with his disk magazine and I volunteered. It's amazing to think that is nearly seven years ago.

Q. Do you belong to other groups?

A. Yes, I've been a member of two other groups. I first joined the Irish group Ozone back in the mid-90's and then released a lot of my own demos under that group. I was also invited to join ROLE back in 2002/3, writing music and text for them.

Q. You write music, is this all "SID" music?

A. Yes, the majority I have written has been on the C64. I used programmes like Dutch USA Team's Music Assembler, Cosine's Electronic Music System (EMS) and DMC v4 (Demo Music Creator). I started out doing a lot of covers from sheet music, but then Warren Pilkington persuaded me to try writing my own. Over the last couple of years I've also done a bit of remixing on the PC, taking my SID tunes and turning them into music for Ovine by Design or releasing it on [remix.kwed.org](http://remix.kwed.org)

Q. Can our readers listen to some of your music, and where would our reader need to look?

A. The vast majority of my SID tunes are in the High Voltage SID Collection at [www.hvsc.c64.org](http://www.hvsc.c64.org). If you already have the latest update, it's at /MUSICIANS/M/Merman (in older editions it's at /VARIOUS/M-R/MERMAN). SIDPlay is one of the easiest ways to listen to it on PC or Mac. Alternatively, a lot of the tunes are at: <http://www.geocities.com/andrewfisher/> in my old website called "Merman's Kingdom"

Q. At <http://c64goldenyears.com/> you have announced a book looking at gaming history, can you enlighten our readers?

A. This is actually a follow-up to a Spectrum book that was published in December 2006. I approached the author of that book, Andrew Rollings, the previous year about doing a Commodore 64 version and he agreed. But with 2006 being a busy year for me, I didn't have the time to work on it. But a new year proved the perfect stimulus to get me to start it up again. The basic format of the book is that it is split into chapters each covering year (with smaller chapters covering 1993-1994, and from 1995 onwards). Each year has a short introduction and a famous loading screen from that year, before each game gets a full page review with screenshots, trivia about the game or the programmer and a description of how it plays. With over 200 games chosen to be in it (and many favourites having to be left out to fit it all in) there is a lot of reading.

Q. How would our reader order this book - when will it be available and what is the intended print run?

A. Pre-ordering can be done with PayPal or credit card via the website - <http://c64goldenyears.com>. The current plan is to get the book finished and printed, ready for sale in summer 2007. Sorry I can't be more specific than that, because there are so many factors that will affect how long it takes. The final number to be printed has not been decided, but the nature of the book means it will be a limited amount.

Q. Who is the publisher of this book and will it be available over the counter in our local book shop?

A. Publishing is being done by Hiive Books, set up by Andrew Rollings to print the Spectrum book. It won't be on general sale, but it will have an ISBN number allowing overseas readers to find it. (Printing is being done in the United States, and the books will be sent from there).

Q. This is a follow-on book; would our readers need both books?

A. No, it will stand on its own as it is about Commodore games. The layout and look of the book will be broadly similar... but it will have better graphics and more shades of grey...

Q. Do you have a sample page our readers could look at?

A. A couple of sample pages are currently available on the website, but those are very early previews and Subject to change.

Q. Do you have any other projects in the pipeline?

A. I'm doing more work on my website, [www.seuckvault.co.uk](http://www.seuckvault.co.uk) I'm writing more stuff for Retro Gamer. And long term there could be more books, depending on how successful this one is.

Q. Any question you wished you had been asked?

A. Hey, that's my question! Seriously, good luck with Commodore Free, and the only question I would like to have been asked is "Do you know someone who wants a million pounds?"

### Commodore Free

*DOH guilty I pinched one of your questions – I should be ashamed – although Yes I could use the Million Pounds I think it would come in handy*

Q. Would you like to plug the book again to our readers - why should our readers purchase this book?

A. If you are a Commodore 64 fan who wants to know more about the great games, discover games they may have missed, or find out some fascinating trivia about the games and machines, then you need this book. Or even if you don't like games it's something you can pick up and flick through When you're bored.

Q. Thanks for your time. How does it feel to be on the opposite end of the interview? :-)

A. It makes a change to be answering the questions!

# An Interview with Richard Joseph

by Neil Carr

Defender Of the Crown, Antirad, Barbarian and more recently 11 tracks for a lego game. Although he follows a different lead with his company Audio Interactive he still writes music to this very day. Talking of music.. Keep an eye out for a remix of Barbarian which may feature in a future BIT Album by Richard himself.

Real name: Richard Joseph  
Nationality: British

Q What other c64 composers did you like?

A Martin Galway is still my favourite even now. Although Rob's was technically groundbreaking there's more emotion in Martin's stuff. Very musical.

Q What non Richard Joseph sids did you like?

A Knucklebusters - Rob H  
Martin Galway - Wizball (awesome music AND fx. The whole soundtrack is brilliant)

Q What tune that you created were you most pleased with?

A It was an ending section for Antirad that I decided to change to the piece you hear now. It was abstract and surreal and I felt that the reviewers and gamers alike would hate it. But I still love it. Also, there was a collection of great tunes written for a game called 'Monster Museum' which was never released. Annoying really, I was just starting to get good when the Amiga came along and we all had to change or lose our jobs.....

Q What were your likes/dislikes about the sid chip?

A I hated the fact that the filter was changed a number of times throughout various production runs of the hardware. This meant that we couldn't really use it at all.

Q You created music for some of the best games on the c64, including Barbarian 1&2, Defender Of The Crown, Antirad. How pleasing for you personally was it for you to be associated with these fine titles?

A I felt honoured. After the 64 I went on to work on very many top titles and I still feel honoured to work on them even now.

Q You worked mainly at the now no longer Palace software, could you explain that period to our readers?

A It was a very exciting time. We all knew we were doing something special and there was a real concentration of great talent. There were only a few games developers in those days and it was very cool to work for indie houses like Palace and System 3. A lot of very high up names in the current games industry came from those studios originally. Palace was a part of the film company that made Evil Dead and Absolute Beginners so the emphasis was on cinematic games from the start. Palace liked getting publicity and one stunt was to feature Page 3 girl Maria Whitaker as the Princess in Barbarian. I

remember the uproar it all created in the tabloids. Brilliant. The Barbarian himself was of course a yet to be discovered Wolf out of Gladiators.

Q Why did you start writing c64 music?

A It was the flagship format for the first game I did for Palace, Cauldron II. Out of the three we working on- Spectrum and Amstrad being the others- the C64 was by far the best to originate a soundtrack on. I never used it for anything but games work, but then the games stuff used to take up all of my time anyway.

Q How did you become involved in the games industry?

A I had spent many years working in the pop music industry and was writing for mainstream entertainment when I answered an advertisement in Melody Maker (sadly no longer with us) put in by Palace who were looking for a composer to work in games. Nothing unusual about that, but this was 1986 of course and there weren't very many computer musicians around at that time. I'd just spent a year composing about 100 tunes on a Yamaha CX5 music computer (including the 'famous' Robocod one), and had tinkered with a Spectrum so it wasn't that hard to convince Palace.

Q What are your thoughts on your music being re-created using modern sounds?

A I'm always interested to hear other people's interpretations but I don't believe that any of these, or remixes of other's stuff is necessarily 'better' or 'improved' just because more modern sounds are used. I think ultimately it will be the original c64 renditions that go down in history whatever remixes exist.

Q Have you heard any of these remixes/covers that has impressed you?

A Well they all have to some degree. I just like the fact that people are enjoying doing them, although I do wonder why- my own stuff doesn't exactly fit into the mainstream of remixable soundtracks being largely orchestral.

Q Is there a tune, you wished you could have worked more on?

A I only had a short time to do Cauldron II. There is a longer version but sadly it didn't get finished quickly enough to make it to the game. I had just two weeks from booting up the c64 for the first time to delivering a finished tune and 20 sound effects.

Q Probably your most remembered tunes were for Defender Of The Crown and Antirad. Would you agree with this, and what can you tell our readers about these two sids?

A Antirad was the second soundtrack after Cauldron II. I took the tune from something I'd written a year or so earlier on midi instruments and stuck a new bit on the end. As I said in another answer there was an alternative part B for Antirad that never made it, and I still listen to it to this day and wonder what might have been. Defender Of The Crown was more remote for me, as Cinemaware who developed it were based in the States. It was weird working that way but I really enjoyed it- the game itself is brilliant fun.

Q What other formats have you worked on, and what was your preference/least preferred?

A  
Spectrum / Amstrad 464 / Amiga / Atari ST / Sega Megadrive / SNES / Gameboy / GBC / Playstation  
Playstation 2 / N64 / PC

# Commodore Free

Favourite was Amiga. Bitmap Brothers, Sensible Software, those were cool times. Least favourite was Megadrive. One of the most interesting moments was listening to Rob Hubbard's Megadrive conversion of my Robocod tunes!

Q So I see you are still involved in the game music industry with Audio Interactive. Do you still write music for games, or do you follow a different lead now?

A I am more involved with the production of a game's audio. There is so much material needed nowadays that I prefer to leave the specialized work, like writing Orchestral music for instance to experts in those particular fields. I did write 11 tunes for a Lego game last year but as a rule I don't write as much. Probably more for myself in fact. I've been working on an orchestral arrangement of Barbarian in the style of Ennio Morriconi, which is what I had aimed for in the c64 version, and its turned out very well. If all goes well it could appear on a future Back In Time CD.

Q What can you tell our readers about Audio Interactive?

A We do everything audio for games. We compose and record music, both linear and interactive. We design and implement sound effects. We do dialogue production, recording and processing. It sounds like a big industry but its not really. Its just the same as it was in c64 days but on a larger scale. If anything there was more pressure in those days, as every single sound was critical. These days we have the luxury of being able to use sampled sounds which makes the job an absolute doddle compared to the old times.

Q If there was a tune that you wished you could as your own, what would it be and why?

A Probably Rob's Knucklebusters. The tune maintains the listeners interest for what, thirteen minutes? That's no mean achievement.

Q What are your fondest memories of the c64?

A The day I booted it up for the first time, after bringing it back from Palace, and playing Beach Head and Impossible Mission with my girlfriend's kids (work it out.... yes I am very old!).

Q What has been your latest project?

A The latest project is Republic: The Revolution for Elixir Studios. The sound has lots to do including conveying the various atmospheres of a city at different times of day and night, with ever-changing orchestral /industrial film music to suit each action within the game.

Q Your music on the c64 was generally more slanted towards the orchestral side rather than heavy drums and baselines. Was this due to the games you made music for or was this your style?

A This actually was quite a problem for me, with the SID only having very limited resources for recreating real instruments. Most of the early games I worked on needed the kind of ambitious soundtrack we are creating now.

Q What did you do after leaving Palace software?

A I went freelance and signed up to do games for The Bitmap Brothers, Sensible Software and Millennium. In 1995 I formed Audio Interactive and we picked up companies like Sony and EA. Last year we won a BAFTA Interactive award for best sound on Theme Park World. Another of our soundtracks, Codemasters' Cannon Fodder (GBC) was nominated for the same award.

Q How would you best prefer to be remembered?

A For the early stuff- not only the c64 but the Amiga soundtracks I did for Sensible and the Bitmaps. Nowadays I would say that Theme Park World is probably the best but I have higher hopes for Republic...

Q Have you ever considered Remixing or re-arranging some of your old c64 sids into modern sounds?

A Yes, the Barbarian thing. Otherwise no, I like them the way they are.

Q How did you get your inspiration for the music for Defender Of the Crown?

A I don't think anything was an inspiration for DOTC. Palace just told me that another company wanted to use me and that there was some tie-in for them. In a lot of ways it was good to work on something, for a change, that wasn't 'Palace' in feel.

Q What does the future hold for you and your music?

A I have no idea to be honest. There is a whole CD of music and mayhem from the ill-fated Sensible game Sex n Drugs n Rock n Roll that we are trying to place with a record company right now. It's quite wonderful really! I just don't want it put out as an mp3 yet- we must have spent nearly six years on it from start to finish and I want to get something out of it. As for the future, Jon Hare (ex Sensible) and I are creating a sequel to the SDRR CD but there is little pressure on us to complete it and neither of us have much time. If SDRR was a success then things would be different obviously.

Q Lastly, what would you like to say to the c64 community?

A I would like to say that it's brilliant that there's a community at all. After the c64 disappeared into commercial history and the industry was pushing forward I just thought that everyone's pioneering work would simply be forgotten. Not so! The appreciation of retro games and the people who made them is wonderful, as is the continued use of the machine and emulators, with remixes and CDs... wonderful.

Richard always followed the orchestral side of music.. Which with 3 voices was as richard explained was very difficult. However he came out of it with flying colours, creating some of the most ambitious music on the c64. Palace software was a small player in the games market, but didn't they produce some fantastic games and who could forget the publicity campaign for Barbarian where they pictured Maria Whittaker scantily clad. This caused a real stir amongst the press. This however must have been exactly what Palace had wanted. The Barbarian soundtrack was exceptionally fitting for the game too. Who could forget the move in Barbarian where the player could spin around and chop an enemies head right off in full gore. Wikid!!!

- Neil  
Interview date: 30.04.2001

Interview printed from  
[http://www.remix64.com/interview\\_richard\\_joseph.html](http://www.remix64.com/interview_richard_joseph.html)  
With Permission via email

Subject: Re: Interview Reprint request  
Hello Nigel,

yes, please go ahead.

Regards,  
- Markus

Commodore Free

**THE END**