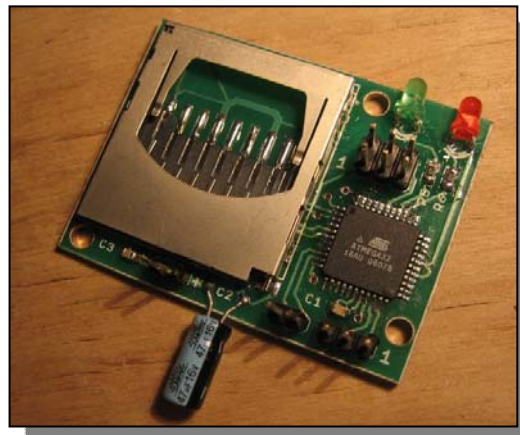


Commodore Free

Issue number 9
June 2007

FREE to download magazine dedicated to all Commodore Computers available FREE from
www.commodrefree.com



Editor

Another packed issue – although I have had to keep the page count down mainly due to the fact the text will no longer fit on a D64, I don't want to do anything like compressing the text or creating just D81 disk images as people with only 1541 disk drives will not be able to read the magazine.

Also it's a lot to edit and I am feeling rather unwell this month, flu and a viral infection has put me on a go slow for most of the month, and I feel tired although I am on the mend now.

In this issue I managed to track down WORLDLAM about his efforts to purchase "CMD" from Maurice although it seems Maurice is still quiet. Worldlam seems to have the money and seems genuine, whatever his reasons surely it can only be a good thing cant it?

If you buy a SCPU or other CMD product and receive it within 3 months at the same price Maurice sold the products then this must be a good thing, still no sign of my SCPU the order is over 3 years old, when I ordered Maurice assured me that 3 months or slightly longer was all I would need to wait, maybe he meant 3 years not months. – Maurice what is happening

WORLDLAM has been aggressive on Ebay buying everything CMD related and other Commodore hardware often for inflated prices; could this just be a late avid collector? See what you think after the interview

David "loadstar" Moorman has been having problems with his house being ripped apart by a Tornado so no part III to the basic programming course; this month although David is keen to restart where he left off. I am sure we all wish him well and hope he and family remain safe, and manage to rebuild and repair their house and lives after the damage.

Magazine has changed a little I removed the reverse block title in the main text, I thought it was using too much ink if people needed to print hard copies, what do you think? Better worse, comments about the magazine have dried up but users are still downloading and reading, basically without readers comments I cant produce a reader comment page.

I still like to receive any reviews links or projects you have seen or are working on, please send in ideas or comments to the usual address

Regards
Nigel
Commodore Free
www.commodorefree.com

Contents

Editor and Contents	Page 2
News from Protovision	Page 3 – 4
Search for a new generation	Page 5
Vic-20 Multi-Cart Prototype	page 6
Emulator File formats	Page 7
Terms and Acronyms	page 8
Interview with Worldlam	Page 9 – 10
Shredz64	Page 11– 13
Interview with Shredz creator	Page 14– 15
Hex files part 4	Page 16– 17
Commodore MMC2IEC	Page 18
Interview with mmc2ieccreator	Page 19-22
Interview with Shaun Bebbington	Page 23-26
CSS64 F.A.Q	Page 27-28
Interview with Hakan Sundell (Css64)	Page 29-30
The end	Page 31

Some general information about Commodore Free downloads, taken from My ISP log files

Issue number 8

PDF document downloads	381
TXT document downloads	93
D64 disk image downloads	17 (33 issue 7)

Other downloads

Quick menu total downloads	52
TND tools Disk downloads	33

Thanks to everyone who takes the time to read the magazine.

Protovision News May 2007

MMC64 UPDATES

Oliver Achten has done a lot of work on the MMC64: Since MMC and SD cards with a capacity of more than 4GB are already available, MMC64 now supports the FAT32 filesystem. This more modern filesystem also has advantages on smaller flashcards, because most C64 files are so small that FAT16 will only waste a lot of space.

The new Bios also made updates on the plugins necessary, most plugins experienced a complete re-write, the old plugins will be updated shortly and can currently only be used with FAT16. The MP3 plugin can now playback whole directories as playlists. The new AVF plugin plays an MMC64 proprietary video format that also utilizes the MP3@64 module. The Picture plugins have also been updated and they are available in version 1.3 at

<http://www.siliconsonic.de/t/bin> .

An example video file in AVF format is also located there. An AVF Converter for your videos is available as a beta test version.

MMC64 Bios changes from V1.04

Finally, a big update this time:

- FAT32 support!

Yes, Oliver managed to squeeze FAT32 support into the MMC64 Bios. Now you don't have to worry about cluster sizes anymore with big memory cards. Currently, cards up to 4GB are supported. Cards >4GB have yet to be released and require a change of the MMC/SD command set anyway.

- New plugin system:

FAT32 support required Oliver to rework the plugin system. When a FAT16 formatted card is used, all plugins work as usual. Using FAT32 formatted cards, old plugins won't work anymore, only new, so called "multifat" plugins will work, which are designed to support BOTH FAT16 and FAT32. With the release of this Bios, most plugins written by Oliver are re-released in the new format.

- Fixed loading of plugins with long filenames

- Fixed support for very small cards.

- Fixed some problems with FAT buffering.

- Fixed a bug in the volume name display routine.

- Instead of showing a computed size value, the

browser displays the card's model ID, which always includes the size.

Downloads

MMC64 Bios V1.10:

http://siliconsonic.de/t/bin/MMC64_V110.zip

Recovery Disk:

http://siliconsonic.de/t/bin/MMC64_Recovery_V110.zip

MMC64 MP3 Plugin V0.70 (requires MP3@64):

http://siliconsonic.de/t/bin/MP3_Plugin_V070.zip

MMC64 Picture Plugins V1.3:

http://siliconsonic.de/t/bin/MMC64_PicturePlugins13.zip

MMC64 AVF Plugin V0.3:

http://siliconsonic.de/t/bin/avf_plugin_V030.zip

AVF Testvideo:

http://siliconsonic.de/t/bin/avf_testvideo.zip

AVF Converter betatest V0.02b:

<http://noname.c64.org/csdb/release/?id=48707&show=notes>

MMC64 VERSIONS OF HOLLOWMAN DEMOS

Hollowman released MMC64 versions of seven of his demos. These demos will now work from MMC64 (even without Retro Replay): WWII, 26 Kg, Horsing Around, Romeo, Axis of Evil, Axis of Evil 2006 and Axis of Evil 2007. You only need to install the DFI Plugin. When starting, select "Dreamload" and confirm loading the boot file by pressing RETURN.

Downloads

WWII:

<http://noname.c64.org/csdb/release/?id=36864>

26 Kg: <http://noname.c64.org/csdb/release/?id=3015>

Horsing Around:

<http://noname.c64.org/csdb/release/?id=3017>

Romeo:

<http://noname.c64.org/csdb/release/?id=41459>

Axis of Evil:

<http://noname.c64.org/csdb/release/?id=18059>

Axis of Evil 2006:

<http://noname.c64.org/csdb/release/?id=29292>

Axis of Evil 2007:

<http://noname.c64.org/csdb/release/?id=48033>

DFI Plugin:

http://developer.berlios.de/project/showfiles.php?group_id=7602&release_id=11984

GUI4CBM4WIN UPDATES

Payton Byrd updated GUI4CBM4WIN, the graphical user interface for the transfer program CBM4WIN, several times. The latest version is 0.6.5. For further information and the download check <http://blog.paytonbyrd.com/?p=89>

CYBERPUNX RETRO REPLAY ROM WITH CODENET SUPPORT

Jockstrap released a modified version of Cyberpunch Retro Replay cartridge ROM V3.8p. This version supports CodeNet, the network server known from The Final Replay ROM. Download: <http://noname.c64.org/csdb/release/?id=48403>

ARTILLERY DUEL NETWORK PREVIEW V0.06

Leif Bloomquist released Preview V0.06 of the Artillery Duel Network.

Changelog:

Random wind for challenge. The game's now fun!!

Really fixed race condition at end of turns.

Tweaked power settings.

20% damage for direct hit.

10% damage for near hit.

PAL/NTSC detection for music and timing.

Homepage:

<http://home.ica.net/~leifb/commodore/duel>

Download (game):

<http://home.ica.net/~leifb/commodore/duel/duel006.d64>

Download (source code):

<http://home.ica.net/~leifb/commodore/duel/artillery20070428-006.zip>

SEQ PLUGIN FOR MMC64

Triad released a SEQ Plugin for the MMC64. With it you can display SEQ files, which are text files in PETSCII format. So it's really nothing fancy, just something that had to be done.

"In game" keys:

- a - abort
- shift - pause
- shift lock - steady pause
- 1-0 - speed
- <- - fast forward
- r - reload file (only works at the end of screen)
- space - exit at end of file

iopop must thank Doc Bacardi for the MMC64 libs.

Download:

<http://retrohackers.org/forum/viewtopic.php?t=254>

SINGULAR BROWSER V0.52

Soci/Singular strikes again at Breakpoint 2007 with Singular browser v0.52.

Highlights:

- supports extended memory up to 16MB (like swap files on Retro Replay RAM, REU, SuperRAM of SCPU, +60K, IDE64 and 1541)
- progressing rendering, it's possible to read the page while loading
- optimized color text rendering
- automatic header redirects
- fixed some bugs in css handling

Other updates are e. g. support of scroll back with cursor keys and detection of page end. Make Singular browser fit to your needs and download it at the new custom download system. Compiling for C128 is supported as well.

Unfortunately, the Custom Download System at <http://singularcrew.hu/browser> is not currently not available. However, you can find the disk only version at

<http://noname.c64.org/csdb/release/?id=48032> .
TURBO ACTION ROM V1 FOR RETRO REPLAY

SounDemoN spiced up good old Action Replay ROM with an improved Turbo Assembler that runs entirely out of Retro Replay ROM and utilizes RR RAM for the source, thus leaving the entire \$0801-\$ffff area available for code!

Retrohackers.org Forum topic:

<http://retrohackers.org/forum/viewtopic.php?t=249>
Download: http://www.dekadence64.org/tar_v1.zip

RAM-RESIDENT TURBO LOADER FOR C128 + RR

FMan released a RAM-resident Turbo Loader for C128 with Retro Replay. This is a disk turbo for 64 mode that resides in the second RAM bank of your C128. Also provides additional features.

Included is a stand-alone cartridge ready version. It has been successfully tested on a real Retro Replay. This is useful for any Commodore 128 user who frequently loads or develops C64 programs in the built-in monitor.

More information and download at CSDb:

<http://noname.c64.org/csdb/release/?id=48034&show=summary>

ONLINE SHOP UPDATE

Our Online Shop got updated: Now the availability of each product is visible. Happy shopping!
<http://www.protovision-online.de/catalog>

HARDWARE SECTION UPDATED

The design of our hardware section has been improved. Some new content has been added as well. Check it out!
<http://www.protovision-online.de/hardw/hardwstart.htm>

<http://www.protovision-online.de>



The Search Is On For The Next Generation

Although this isn't Commodore related I thought it may interest readers hey its only 1 page

Introduction...

WHAT is The Voyager Crusade?

The concept is simple, it is to design and program an original, innovative, but above all playable game using the technology from the golden age of video gaming. Nowadays game designers have 3d graphics, FMV and surround sound to play with, and too often we see games that look fantastic, but lack that all important holy grail of gaming...playability.

This is a challenge to all up-and-coming game designers...show us you can still do it!

The Voyager Crusade is the brainchild of Jonathan Thompson aka "JT", the man behind the "Voyager" arcade cabinets and game systems. The company supplies game machines to many top retailers, including Firebox.com and Selfridges & Co. The machines are also popular with celebrities, you can find Voyager cabinets in the homes of several Hollywood stars, and David Beckham recently took delivery of his very own custom gold cabinet! You'll find Voyager cabinets everywhere, museums, film premieres, exhibitions and more traditional venues like hotels and pubs and night clubs including the well known Cult TV chain clubs "FAB CAFÉ".

Who can enter?

THIS is NOT a competition to be passed over lightly; in fact if you want exposure for your talents, this is an opportunity not to be missed!

The competition is open to amateur programmers from all over the world, however you must be aged 18 upwards!

See rules for more information.

What will you win if your game is picked?

...First Prize...

IF your program is chosen as the winner you will have your game included on the Voyager Arcade System, so your game will get exposure to a huge audience! What an advert and something to add to your portfolio!

You also win a website for you to promote your games on, to other companies and also a customized one off Classic Games Machine designed to show off your games to potential clients. Thanks to Joystick Junkies you will also get a special T-Shirt...

This will be one of the top of the range JT Machine Cabinet. Only a few of these top of the range cabinets exist and they are all bespoke designs! A collector's item from day one.

The winners JT Machine Cabinet will also be signed by the man behind Darth Vader's mask, Dave Prowse and Jonathan himself for that personal touch!

To make the deal just that extra bit more exciting, you will also have a launch party put on for you in the heart of Liverpool City Centre, the European Capital of Culture in 2008...

TV and Film Stars will be invited as well as special celebrity guests and as prominent game manufactures and designers will be present.

Who knows, they might even take your design on further...

...Second Prize...

IF your program is chosen as second place then you will have your game demonstrated at the launch party and also added to the launch party machines so you can show it off to potential clients at the party.

You will also be given an opportunity to add your game to the Voyager play lists.

Thanks to Joystick Junkies you will also get a special T-Shirt...

...Third Prize...

IF you come third, you will be invited to the party and also you will be given an opportunity to demonstrate your game at the event.

Thanks to Joystick Junkies you will also get a special T-Shirt...

So what is the outline of the rules...?

WELL the game must be retro in style... The game must be simple, fun for all ages and easy to play. Remember the games "Space Invaders", "Bomb", "Mr Do", "Pacman", "Starwars" and other games like "Spy Hunter"; well these are the types of games we want to see. Simple, fun and nothing special, but it must be your original concept.

The games must be programmed to work on a PC Based machine running Windows XP, Graphics must be run in SVGA only, work with Electronic Coin System, 4 way and 8 Way, 3 button, Joystick configuration, 16K Sound and able to run 2 players as well as single player. See rules for full specifications and information...

Launch of the entries will start officially in January 2007, but if you are already interested you can email us your FULL name, date of birth, address, your Skype ID and email address. Information will be held under the UK Data Protection Act by both Retro Arcade Machines Ltd. and PCBT Photography and will only be used to send you information packs about the competition and also follow up information about the competition only.

For more information visit
<http://www.it-voyagercrusade.com>

Vic-20 Multi-Cart Prototype Project

<http://www.bjlyons.com/cbm/vic20/multicart/>



Project is based on concept that originated during discussions at WOC show in Dec 2006. This concept became a discussion item on an Internet discussion forum devoted to Vic-20 topics

"Denial" <http://sleepingelephant.com/denial>

The Multi-cart is intended to allow a user to run multiple Vic-20 ROM images from a single cartridge. This design builds on original concept to add features and functions

1. Concept

Allow multiple ROM images to be accessible from single cartridge

Reduce physical storage requirements of having multiple carts

Greater portability to take cartridge collection with you when you are on the go

Reduce number of times a cartridge is inserted & removed from Vic-20

2. Project Features

Ability to run up to 255 - 8K ROM based Vic-20 Cartridge games or utilities

Support for 4K, 8K, 12K, & 16K ROM games

Support for all known ROM images, irrespective of location of ROM image in Vic-20 memory map

Toggle from game to menu and back to game again via reset button

RAM expanded mode with additional 32K RAM

28,159 bytes directly available to Basic (3,583 core memory + 24,575 expanded) plus 8,192 bytes addressable by Basic for storage of variables, arrays, etc.

Ability to switch to unexpanded Vic-20 Basic mode without removal of cartridge

System reset via software control or push button switch

Bank switching between RAM & ROM

Expanded RAM may remain hidden from kernel on start up and enabled via software control – no video memory address change from defaults

Software controlled write inhibit logic for R/W line on expansion RAM - no clumsy switches

150+ Games & Utilities currently loaded and fully functional (25% of EPROM capacity still available for further images)

3. Credits

Hardware Design
Testing

Brian Lyons
Anders Carlsson
Robin Harbron

Menu Software

Anders Carlsson

Hardware Concept & Ideas
Organization

Leif Bloomquist
KilrPir

Francois Leveille
Mark G. – aka saundby

ROM Images &

Leo – aka

NBLA000

Brent Santin

4. Current Status

Menu under development via project on Denial discussion forum

Updates may be found on Denial – <http://sleepingelephant.com/denial>

or Hardware project page

<http://www.bjlyons.com/cbm/vic20/multicart/>

Commodore free Comments

Brian Lyons sent me an email informing me about this project and said if I have free space to include it in Commodore Free magazine, so I did.

I emailed Brian in the hope of an interview about the project and an update but haven't heard anything from him so if you are reading Brian can you contact me again I would love to cover more about this project.

Maybe it was just that my emails have been blocked by your spam filters

Thanks commodore Free

Anyone else

If you have a project hardware or software let me know so I can tell everyone else about it
Thanks

Introduction to various Emulator File Formats

Compiled by: Peter Schepers

Started: August 24, 1996

Last updated: Nov 27, 2005

Commodore Free Peter kindly let me reprint 2 articles from his website, other are available and I suggest you study them for yourself, the attached 2 I though were worthy of a reprint

There are always questions asked regarding the various file formats which are commonly used on either the emulators or the real C64. Most often the Question involves conversion... "What do I do with LNX files?" or "How do I make these files work on the C64s emulator?" These documents attempt to explain their internal structure, what to do with them, and some of their respective strengths and weaknesses.

These documents were compiled and written in an attempt to unify all the other smaller files dealing with Commodore file types that are floating about the net, or that exist with other programs. They are by no means exhaustive (even though they look like it), but attempts will be made to keep them up-to-date, and correct anything which is wrong. If you spot something that needs correcting please make sure to email the author so that corrections can be made for future releases... the address is contained later in this document.

Some of the information contained in these documents may not be accurate as it could have been taken from inaccurate sources, and I have no first-hand experience with said format. However, use these, pass them around, upload them, whatever. Just be sure to leave them INTACT, don't remove bits.

I have attempted to categorize the filetypes involved using three basic categories: IMAGES, ARCHIVES and CONTAINERS. The definitions for each of these categories can be found at the bottom of this document.

Also, plenty of good information can be gleaned from the source code contained in the archive CBMConvert, which is on the FTP.FUNET.FI FTP site. Contained in it are the sources for UnZipCode, UnLNX, Ark, some LHA info, etc. It is an invaluable set of utilities put together by both Marko Makela and Paul Doherty.

So far, this document covers the following files:

- * D64 images (1541 disks and some variants)
- * X64 images (for the X64/Vice emulator)
- * T64 containers (for the C64s emulator)
- * T64 .FRZ (FRoZen Files, saved emulator sessions for C64s)
- * PC64 containers (P/S/U/Rxx)
- * PC64 .C64 (saved emulator sessions for PC64)
- * D71 images (1571 disks)
- * D81 images (1581 disks)
- * D80 (8050) & D82 (8250) floppy images
- * G64 images (GCR copy of a 1541 disk)
- * D2M images (FD2000 disks)
- * DNP images (CMD hard disk native partitions)
- * F64 (not an image file, but a companion file to D64's)
- * N64 (64NET's custom files)

- * L64 (64LAN's custom files)
- * C64 (PCLINK's custom files)
- * CRT images (CCS64 ROM cartridges)
- * 64x (PC64 ROM files)
- * TAP images (for CCS64, sampled cassette tapes)
- * VSF VICE snapshots (saved-emulator sessions for VICE)
- * WAV Audio RIFF files for the PC

...as well as the following native C64 types, some of which are also supported on the various emulators:

- * Extensive disk file layout (how files are stored on 1541/71/81 disks)
- * 4-file diskpacked ZipCode archives (or .Z64, 4 or 5 files, #!xxxxx)
- * 6-file SixPack ZipCode images (or .Z64, #!!xxxx)
- * Filepacked ZipCode archives (or .Z64, x files, x!xxxxx)
- * LNX containers (LyNX)
- * ARK containers & SRK archives (ARKive & compressed ARKive)
- * LHA & LZH archives (header description only)
- * SFX archives (Self-eXtracting LHA/LZH)
- * SDA archives (Self-Dissolving Archive)
- * ARC archives (ARChive)
- * ZIP archives (PKZIP)
- * CKIT archives (Compression KIT)
- * CPK containers
- * WRA & WR3 archives (Wraprot, version 1 to 3)
- * LBR containers (LiBRary, C64 only, not the C128 CP/M .LBR files)
- * GEOS VLIR files (Variable Length Index Record)
- * REL files (RELative)
- * CVT files (GEOS ConVerT)
- * SPY containers (SPYne)
- * C128 Boot Sector layout
- * Binary & PRG (ProGram, with load address)

Also included is a very basic look at some C64 graphic bitmap formats (in BITMAP.TXT), and the saved session layout of the Macintosh-based C64 emulator "Power64" (in POWER64.TXT). Thanks to Peter Weighill for the aboveinfo.

Joe Forster/STA has written up a description of how the various Commodore drives (1541/1571/1581) allocate sectors and directory entries when saving files (under normal mode and under GEOS). It is included as DISK.TXT

Right now there are several good utilities available to work with most of the mentioned formats. The first is 64COPY, my own conversion program. The second is Star Commander, by Joe Forster/STA. Included with his program are many smaller utilities such as Star ARK, Star LHA and Star ZIP, which will convert specific formats to D64 images.

Peter Schepers,
University of Waterloo.

<http://www.unusedino.de/ec64/technical.html>

Terms and acronyms by Peter Schepers

Many strange terms have come along with computers in general, and I will not attempt to explain them all, but some of the ones in this document may not be entirely clear. I will attempt to make things a little easier by explaining some of the more common ones.

<CR> - Short form for a Carriage Return (\$0D) symbol.

<LF> - Short form for a Line Feed (\$0A) symbol.

ARCHIVE - A file format which contains other files, and in which compression is an integral part of the design. Some examples are ZIP, SRK, SDA, ARC, LHA, WRA.

ASCII - This is an acronym for "American Standard Code for Information Interchange". The standard is a 7-bit code covering control codes, punctuation, alphanumeric (A to Z, 0 to 9) as well as math and a few other symbols. Since it is a 7-bit code, it ranges from \$00 to \$7F (0-127). This leaves the top 128-255 definable by the vendor. The PC world has corrupted this standard making it 8-bit.

BAM - An acronym for "Block Availability Map". Here is where the disk operating system keeps track of what sectors are allocated (or available) for each track.

BLOCK - This refers to sectors which on a logical level are grouped together. On a 1541 disk, it could be a series of sectors linked together in a file, or a partition on a 1581 disk. In the PC world it represents a "cluster" of sectors. Generally if I'm referring to a grouping of sectors that's *not* 256 bytes large, then I talk in blocks.

BYTE - A group of 8 bits, the contents of a memory location.

CHAIN - A series of sectors linked together. One sector will have a pointer to another, and that sector will point to another, until the chain has no more forward pointers. A file stored on a 1541 disk would be considered a chain of sectors, but it also has a directory entry explaining what the chain is for.

CONTAINER - A file format which simply contains other files, and no compression takes place. Some examples are T64, P00, SPY, ARK, LNX.

FILETYPE - In the Commodore world, this would be the kind of file, be it SEQ, REL, PRG, USR, GEOS etc. In the DOS world, this would possibly be the file extension, be it EXE, TXT, DOC. It tells the user what file it is, making usage easier.

GCR - An acronym for "Group Code Recording". This is the encoding method Commodore uses to physically store the information on most of the 5.25" disks (i.e. 1541). It encodes an 8 bit sequence (2 4-bit nybbles) into a 10 bit sequence (2 5-bit nybbles) so that long repeated sequences of 1's or 0's are avoided. These must be avoided so that the timing of reading/writing to the disk won't become "out of sync". As a user, you would not normally see the GCR information since the drive does all the conversion to normal HEX data before it gives it to you.

HIGH/LOW - The bytes here are stored backwards compared to the LOW/HIGH method. See LOW/HIGH for more information.

IMAGE - A file format which is a PC equivalent of a physical Commodore media. Some examples are D64 (1541), D71 (1571), D81 (1581), D2M (FD2000), X64.

LINK - This is the track/sector values, stored in the first two bytes of a sector, which point to (or "link" to) the t/s location of the next sector. A series of these links comprise a "chain" of sectors.

LOW/HIGH - This is how values are stored when they exceed one byte. A good example of this is the sector count of a D64 file. To calculate the actual value, take the second value, multiply it by 256 and add the low value. You will now get the real decimal value. i.e. (HIGH*256)+LOW=result. If you look at it as a HEX value, swap the bytes around and put them together for the 16-bit HEX value. i.e. \$FE \$03 would be \$03FE as a 16-bit HEX value.

LSB/MSB - See LOW/HIGH.

LSU - This is my own acronym meaning "last sector usage". It is the value stored in byte position \$01 (the "sector" value of the t/slink) of the last sector of a file. This value is the offset into the sector where the last byte is stored. It also represents the byte count + 1, since a value of 255 would actually mean only 254 bytes of file data exists (full sector less the 2 bytes for t/schain). Without reasonable knowledge of the disk layout, this byte can be confusing, and hard to explain.

NYBBLE - A grouping of 4 bits (half a byte), either the first or last 4 bits of an 8-bit binary number, or one half of a two-digit hexadecimal number. Typically, a byte will be broken down into two parts, the top 4 bits and the bottom 4 bits. These are referred to as the upper and lower nybble respectively, and are represented by two hexadecimal digits in base 16.

PETASCII - (or PETSCII) This is Commodore's version of ASCII (the PET part of the name comes from the first computer to use the code, the PET or Personal Electronic Transactor). Most of the codes from 0-127 are the same as ASCII, but there are differences, especially noticeable when converting text from a C64 to a DOS machine. Where ASCII has uppercase characters, PETASCII has lower case ones, and vice versa. Also, the top 128 characters (128 to 255) are quite different from the PC "standard".

RLE - An acronym for "Run Length Encoding". This is a simple compression method, employed by most compression programs, and also used by some file formats (ZipCode, CPK). It encodes sequences (or "runs", hence the name "RUN length...") of the same byte (i.e. 00 00 00 00 00 00) into a smaller string using a shorter code sequence, making the resultant file smaller than the original. This is the simplest form of file compression.

SECTOR - It is best described as the method that the drive uses to store the smallest group of bytes physically on the disk. On the 1541 this refers to a group of 256 bytes stored together in a single sector. On a PC disk, this is typically 512 bytes.

SIGNATURE - A group of bytes, usually near or at the front of the file, which are used to identify the type of file. I.e. a PC64 file will always have the signature string "C64file" contained at the beginning of the file.

TAR - An acronym for "Tape ARchiver", a UNIX application, and method of backing up information.

Commodore Free

Reprinted with the permission of Peter Schepers

<http://www.unusedino.de/ec64/technical/formats/intro.html>

Interview with WORLDRAM

A potential purchaser or CMD

Q - Please would you introduce yourself to our reader?

I am 36 years old, married, and have 4 children. I have been a resident of Oregon most of my life. I started getting in the Commodore scene back in the 80's when I was a teenager.

I remember at Christmas getting my first Commodore 64 back when I was 15 years old. I stayed up all night playing games. Joining different groups in my local area and staying up on BBS's most of the night I phreaked for a little while when you could get away with it on the Sprint Network to call long distance bbs's without my parents knowing about it. I eventually sold my C64 to get a Amiga and never got into it then I just kind went through my 20's a lost soul partying way too much. Getting married and having children brought me back to reality.

In my late 20's I was going to thrift store's finding old Commodore equipment and picked up some basic's and I played for it for a while and lost interest once again. I remember the old days of having meetings at our local community college to swap software and using Fast Hack'em to copy games. Still today I have fond memories of that software. Then about 2 months ago I found Alee650 in the UK that was selling Fast Hack'em and Kracker Jax and it brought back many memories I started buying and then I couldn't stop. Then I met Jerry Kurtz which had the biggest collection of rare games and I started getting back into games again. Remembering Phantasie and Space Taxi from my teenage years.

Q - What is the term "WORLDRAM" is it your real name or a "handle" what does it mean

It is short for my corporate name for one of my website's. Worldwide Laminate LLC.

Q - You seem to be purchasing everything with a Commodore name on it would you like to comment

I've been buying alot of rare items. I also collect Amiga, Apple, and Atari. My first choice is Commodore 64 of course.

Q - What is all the equipment going to be used for?

My goal is to end up keeping one of each item and eventually selling the rest back to the community. I really want to start playing with eproms and programming them I think it would be fun. I just haven't had the time to it yet. Maybe programming some cartridges as well.

Q - Some users are upset they get outbid on items and you always win can you comment

I sometimes overpay because of my aggressive bidding habits. However, I have always had a winning attitude and don't like to lose on anything. I can't figure out why they get upset as I took a 2 month window on ebay of stuff that has been available for years. I have quit trying to understand why people continue to flame me I just try to ignore them now. It's mainly on the message boards

I have encountered this problem.

Q - Are you just a collector or is this a part of childhood you missed out on.

Collecting items I couldn't get when I was a teenager. Like I said in my introduction.

Q - I find items on Ebay that were never released in the uk or were unavailable is this what you after.

I have noticed where theres a huge demand and whatever get's the most bid activity draws me to the auction. If other's want it really bad then I feel there is a reason for it so I sometimes bid overseas to get items that are rare. I got most of my Commodore Max items in Japan, Commodore 232 in the UK, etc. Where in the USA it's very hard to find them.

Q - Is there an item or items you desperately want and what you are prepared to pay for them maybe our reader could help out.

Part of my collection that I have the most emotional attachment to is the freezer cartridges i.e. Trilogic Expert, Action, Replay, ISEPIC. Eprom programmers i.e. Promenade C1, Merlin, Datel, etc. Fast Hack'em ALL VERSION's, Kracker Jax, Utilities Unlimited, etc. All the hacker tools that were very popular today as they were 27 years ago. You won't find me selling these item's any time soon. The games, computers, etc I feel I could always replace. Besides me and one other collector there isn't much competition on these items so I pick them up pretty cheap. The games on the other hand its very easy to spend \$50k and not get much in return. Also hard to resell them and make your money back its a very strange market. One day Space Taxi sells for \$520.00 and two weeks later it sells for \$242.00 huge price difference.

Q - Can you tell us a little about your Commodore Collection?

Some people on message boards started posting everything I was buying and the pricing I paid which I feel is nobody's business and I was notified from a seller I was being talked about and I went to answer why I was spending so much and buying so many rare items. But the flaming wars began so now I just try to lay low and not say much if I can. Some of biggest money items to date are: Commodore 232 (5 known to exist in the world), 2001 8k Pet Clicket Keyboard (pretty rare in mint condition), 3 Max Computers, about \$55k in rare games and then a bunch of misc stuff.

Q - Why Commodore what is the entire Buzz about.

Back in the 80's everyone had a Commodore 64 in their home. We all teased anyone that had a IBM or Apple. It was the machine. The love for them still hasn't died 27 years later. They are the best 8bit computer ever made. Period....

Q - You recently posted that you wanted to purchase CMD from Maurice Randel - do you have a business plan

I emailed him and didn't get a response. I saw a huge demand for CMD products on eBay and they were always going for high amounts. Yet on Maurice's website they are a lot less money. I almost ordered there but read all the complaints on the message board's that he hasn't delivered in a long time. I would need to fly out there and visit with him and see the whole story before I made a decision. There is always 2 sides to every story and I haven't heard his. I have no hard feelings for him because I never dealt with him before and he doesn't have any of my money. After hearing his side I could create a great business plan to present to him. I have been self employed for many years and know what it takes to run a business.

Q - Have you spoken to Maurice and or do you intent to visit him personally.

I have not spoken to him yet and at this time until I get some of kind of feedback via email I will not pursue this much anymore. Maybe it would be a good time for you to interview him and see what he thinks about the thought of selling CMD and if so I would be happy to re-contact him. I emailed him but will never beg for his business so I will not continue to email him over and over again I feel he will contact me back if its a interest to him.

Q - Would you be prepared to hint at the amount you offered for CMD

Nope. There is alot of factors I would have to see the inventory, his financial books, etc. Until I know the whole picture I would never buy something blind. You would never buy a house without seeing it first, correct?

Q - What did the offer cover Intellectual Property and software or was it the whole company including spares or components Maurice has in "stock". It has not gone this far yet.

Q - Our reader may not be aware of the term IP (intellectual property) can you explain the term

The term "intellectual property" denotes the specific legal rights which authors, inventors and other IP holders may hold and exercise, and not the intellectual work itself.

Q - When you start production - What will the products retail at

I will look at cost's of materials, labor to assemble, packing, and see what the profit ratio is. I would keep the pricing the same if I could.

Q - Will you keep the CMD website www.cmdrkey.com

Yes the domain itself is worth money and everyone knows it by the name. I would never change the domain name.

Q - Will Maurice still do repairs and programming?

I would love to keep him with the team of course. You can't replace his ideas, knowledge, and new product ideas.

Q - What is you technical background in electronics and programming

I have been a PC technician for 8 years. I can fix a PC with my eyes closed. But, anything I am not a expert in I would hire outside help or keep Maurice around to assemble the products until someone was trained or just keep him doing it if he choose to do so.

Q - Will the products be mass produced or do you intend to carry on hand production

I would have to see the manufacturing process to develop a plan to expand or keep things the way they are.

Q - Do you have any feelings for users that have ordered CMD products from Maurice and still after 2 - 3 years still not received any product or refund?

Of course I do. There is a lot of damage control that would need to be addressed but there is also fraud issues where anyone can say we owe them product. There would need to be proof. Credit card receipts, mailing tracking #'s, invoices, etc to address the problems.

Q - Have you looked into the complexity of the products - some of the designs are multilayer boards

I own almost every product I have not taken the time to take them apart. I am sure they are high quality products. My role would be more of a CEO and not the one assembling them I would hire someone that would do this work but I would also hire someone to do quality control and make sure everything is working before it ships out.

Q - Will the designs change, and have you spoken to any Electronics manufacturer about mass production.

If I was buying the business I would have access to all wholesale distributors, sales agents, etc I would ideally call them and tell them I have took over and introduce myself and develop a relationship. I would also take feedback from our customers on what mods they would like to see and if there was a big demand I would see if we could develop it and market it.

Q - Will the products be available to distributors for example "protovision" and should it return "Commodore Scene Import Service"

Distributors are always a good idea as it get's your product to more customers in different country's.

Q - If you were given 1 million pounds what would you do with the money (can be none Commodore related)

A castle what man doesn't want that.

Q - Is there a question you would have liked to have been asked?

Nope I think we covered it all.

Commodore Free
Thanks for your time and good luck with your purchase.

I open this for discussion in the magazine, would a purchaser of CMD be a good or bad thing for Commodore

The Shredz64 Project

By Toni Westbrook

The Shredz64 project is an attempt to build an interface to connect the Playstation Guitar Hero controller (and any other PS controller) to the Commodore 64 computer, then create a Guitar Hero like game on the C64 utilizing the guitar controller.

So far the project has been going well! As you can see, the interface, which I call the "PSX64", is 98% functional (There are some tiny bugs which don't really impact functionality on the whole). Not only are all guitar functions mapped over successfully, but it works great for playing C64 games with a normal dual shock controller. What's more, since the same DB9 port is used, the PSX64 can be used on the Amiga, Atari and



Sega Master System (All tested except SMS)..

The Shredz64 game itself is still in its infancy, but progressing nicely.

Most importantly, my journal of latest updates can be found on the "News Updates" page, for those interested. I don't have a ton of free time, but I'm very dedicated to this project, so it will progress, if only just slowly.

Shredz64 News

4/6/07

Lots of very promising progress tonight (and a few nights ago). First off, I implemented code that will read both note data for fret buttons and music data from SID files off of floppy disk. I also wrote some quick utilities on my Linux box to generate note files for testing. So all the static values are out of the program and its reading off files now. More importantly though, I implemented a good portion of the SID -> Shredz64 converter tonight - this sounds much fancier than it actually is. The issue is normal SID files are big blocks of 6502 machine code that have been ripped (or created), but they contain code for manipulating the processor, memory, etc as opposed to just being music data. Not only does this waste CPU time, but its also difficult to switch back and forth between executing Shredz64 code and executing SID code. My solution for now was to hack the libsidplay library on my Linux box, and intercept any memory writes to the virtual SID chip (0xD4XX) and dump the address and the value to a binary file, along with the relative time the memory write was made. E.g. I play a SID file on my computer, and in addition to playing it it dumps all the SID chip data to a file. Shredz64 then uses that file as its music data. Actually, I didn't implement the

timing tonight, but I did manually set the timing inside the game just to test it out, and it played the music pretty accurately! Very promising for first try. I do need to compress the files somehow, at least look for repeating sequences, since they get pretty big. Hopefully filesize won't end up being an impossible issue. But hearing music out of the game for the first time tonight sure was awesome!

3/26/07

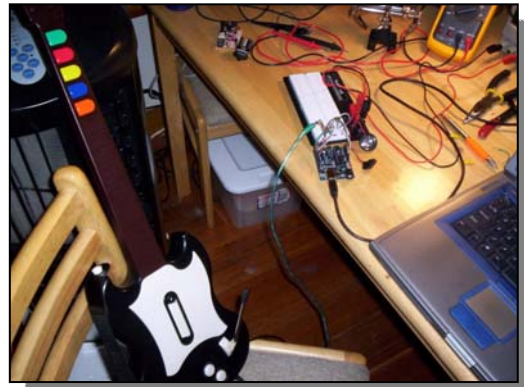
I added the applause meter tonight that shows how well you're doing overall and how close you are to failing. As can be seen, its a vertical bar separated into three sections, filling from bottom to top. When doing well, the bar fill is green, when okay its yellow, and when bad its red. When empty its game over! (Though I don't have a game over yet). Your applause increases when you correctly strum a note, and decreases if you miss a note or incorrectly strum. I also fixed up some bugs recognizing correct strumming, and separated the code out into multiple files now that things are getting bigger. Next up is reading note and song data from files, and then the SID converter plugin!

3/21/07

I snuck some hours in yesterday before I went to the dentist and worked on things. Shredz64 can now take a (static) list of button data (e.g. buttons to be pressed at what time and in what order) and correctly show it on the screen. For instance, I can encode data that says "A green button should be pressed at 00:00:10, 00:00:20, and 00:00:45, a red button should be pressed at 00:00:10, 00:00:30, etc, and it will correctly show and scroll the sprites on screen. It will also detect if you strum the correct chords at the correct times and increase your score upon success. I've also added a "Shredocity" meter that mirrors the functionality of Star Power in Guitar Hero for the Playstation. I will soon be adding a crowd meter that will determine how well you're playing..

The Shredocity meter is a bit chunky looking right now, I'll be thinning out the graphics.

Overall, things are going really well. The framework of all the controls and graphics are pretty much done, there is only the big project left of incorporating SIDs in as the source of the music, and the smaller task of reading the song data from files on disk. SID music is ironically incredibly hard to work with on a C64 since its just machine code that was ripped from some other program, but I have plans incorporating SID plugins for XMMS to convert it over to a different usable format. More on that later after some code has been written.



3/4/07

Well, I got a few more things done. As I mention on the Shredz64 page, graphics are simple and confined to PETSCII and sprites right now, no high res stuff. I want

to save all the processing power I can for good response and music. I'll tweak up the sprites later on, but for now this is fine. Anyway - I have the fret board showing, and button indicators at the bottom. They do respond to the guitar controller now, so if you press a button on the guitar it will light up on the screen. I have the strum bar just playing a beep out of the C64 right now to let me know its working. As can be seen, I also did a simple sprite to represent the note/button as it slides down the fret board.

3/2/07

I had a day off from work today, though I ended up sleeping pretty late and not getting everything done I wanted to. I did, however, build a PC64 cable and get cbmlink installed and running on my Debian box. Now I can successfully read/write files/disk images to my C64 from my PC, read/write memory, remotely execute programs, run disk commands, etc. The reason this is related to Shredz64 is now I can actually test it on a real C64 with the controller as opposed to only the emulator with no controller. The cable isn't too exciting looking to no pictures, but it does operate on a 4-bit parallel connection between the user port and the parallel port, as opposed to a serial connection.

2/27/07

Well, I updated the webpage a bit. Nothing too fancy, but I wanted to clean it up a bit so some sections were separated out. This page will still serve as a log of things I'm doing.

In technical news, I've got some graphical routines done and have a fretboard on the screen with scrolling buttons. Screenshots later when I have something a little more to show.

2/21/07

Lots of good work and news going on with this project, I figured it was about time to actually update this page with what's going on.

The soldered prototype for the PSX -> C64 adapter is now done, as can be seen here:

The converter now does the following:

1. Determines whether a normal controller is plugged in or a guitar controller, and maps buttons accordingly
2. Turns on analog mode for guitar controller and successfully reads whammy bar. (Updates later about getting this working for all those interested)
3. Maps Strum up, Strum down, and guitar lift up to static values on one of the POT lines, which allows the C64 to receive ALL information from the guitar (except start and select button)
4. Allows button macros to be programmed onto the R1/R2/L1/L2 buttons in controller mode, up to 127 button sequences PER macro
5. If the user selects analog mode on a normal controller, it will map the left analog stick to the normal digital directions
6. And converts all buttons successfully of course.

After a hefty debugging session, the physical adapter is solid and works great with both a guitar controller and normal dual shock controller. It was amazing to try all my old games with a PS2 controller (as well as some Amiga games and Atari 2600, haven't tried the Sega Master System yet).

Also, I've started writing the actual Shredz64 (guitar hero) game for the C64. I started out with a simple debugging routine to show the status of what guitar buttons were being pressed. Here are two screenshots (literally) of the C64 reporting what's happening on both the fret board and strum bar/lift sensor:

While this program was written in BASIC, the final Shredz64 program is being written in a combination of C and 6502 assembly using the CC65 cross compiler,

which I got up and running a few days ago. I'm currently developing in an emulator environment and am waiting for parts to arrive to build a cable to transfer the program to the C64 itself for testing.

Anyway, so much more to come, but things are progressing nicely! I've found out that this project is featured in Engadget which is great, I hope to be making updates more regularly now that I know people are reading. If you have any questions, feel free to email me: twestbrook AT synthdreams DOT com. Huzzah!

1/10/07 - 1/11/07

I recently ordered the arduino development board - it's a programming board for the ATMEGA8 microcontroller with a built in USB interface. It was extremely cheap (~35 for the board, 3 bucks per ATmega8 chip). The IDE allows you to program in C code and upload it straight to the flash memory on the ATmega, which is called by the bootloader on startup. The ATmega8 is awesome for 3 bucks, it runs at 16mhz, has 1K RAM, 8K flash memory, and 512 bytes of EEPROM memory. Great for a project like this. HOW DOES THIS RELATE TO THE GUITAR HERO CONTROLLER?

Well, long gone are the days when a joystick simply put voltage over a line to indicate a button being pressed. This is true on Atari/Commodore/Sega/Amiga controllers, but nothing recent. Back in the day, if you pressed LEFT, a circuit to a specific pin would be completed on the 9 pin connector. Press up, a different pin would be connected. A pin for every button, it was an easy life. But now PSX, XBOX, GC controllers have a billion buttons and only a limited number of lines, so they encode the data into a serial stream of packets, just like if you were sending data over a serial connection or network. In comes the ATmega8 microcontroller. The atmega receives the serial stream of packets, decodes them and figures out which buttons are being pressed. It then drops voltage on corresponding lines to the output to the commodore. But before I did that I had to test to make sure my decoding program was working! THIS WAS A NEAT TEST.

First I took a PSX extender cable:

I opened the male connector and got the pins out. I continuity tested each line so I could figure out which color was which pin. I then connected these pins into a solderless protoboard and mapped arbitrary pins off the arduino board into the protoboard. I also hooked a tiny loudspeaker up to the protoboard too and mapped the arduino into it. The goal to test this guy out was to have the ATmega play sounds on the speaker when I strummed with buttons held down on the guitar.

Then was the task of writing the decoder in C. I found some docs online that describe the protocol the PSX controller uses. It basically consists of sending data over a COMMAND line, listening on the DATA line, manipulating the CLOCK line to drive the data, checking the ACK line for good measure (not really necessary), and dropping the ATTENTION line when it was time to wake the controller up. The protocol was the following. Drop the attention line, Send 0x01 over the COMMAND line to the controller, the arduino should receive 0xFF back on the data line. Then the controller is sent 0x42 which is a REQUEST FOR DATA command, at the same time the controller responds back with what mode it is in (Digital or Analog, mouse or whatever). It then sends back 0x5A to indicate "Here comes the data SUCKA", and then sends 2-7 bytes depending on if the controller is in digital or analog mode. Those bytes contain a bit for each button, either set to 0 or 1 depending on if the button is pressed or not.

It took about two days to get this working, one because I needed to add a 10K pullup resistor across the COMMAND pin as the arduino was polling the controller too fast and the line noise was crapping out any packets I was sending. I found this tip online by someone who

had done another PSX controller project. The other was I wasn't reading and writing data properly with the clock cycle, I was doing more on the rise of the clock and I needed to do things on the fall of the clock. It was frustrating, but eventually I got it working. I then wrote a quick program to print out to my laptop (via the USB serial link) what button was being pressed, then I pressed each button on the guitar controller and saw what the corresponding PSX button was. Here's THAT:

Green = R2
 Red = O
 Yellow = Triangle
 Blue = X
 Orange = Square
 Start = Start
 Select = Select
 Lift guitar up = L2
 Strum up = UP
 Strum down = DOWN

I couldn't test the whammy bar out as I know for a fact it manipulates one of the analog sticks, and I don't know the command yet to force a controller into analog mode, so the guitar starts up in digital mode and disables the whammy bar.

Anyway, I got the decoder working, and using the button map data, I created a little program for the atmega that plays a note through the loudspeaker when the strum bar is hit depending on what key combination is held down during the strum. It basically assigns a frequency value to each key, then adds them together, and plays that. It's not aligned to real notes right now, I did it more for just a fun test to actually see the thing working. There really is no other steps between here and hooking it to the Commodore 64 other than wiring up a DB9 connector.

That's it for now! Soon the interface will be done and it will be time to start on the game.

The Interface

The PSX64 interface connects playstation controllers to computers that use a DB9 port and C64 pin configuration (such as the Amiga, Atari 2600, Sega Master System, etc). Additionally, if a guitar controller is detected, it will encode strum up, strum down, lift up, and the whammy bar onto the two potentiometer lines of the joystick port. While this extra guitar functionality is usable by the Shredz64 game only (and any future projects), the normal functionality of the PSX64 interface can be used with any game, and works well!

The PSX64 features the following:

1. The aforementioned ability to hook a playstation controller up to a DB9 joystick computer. It maps up, down, left, right and X on the PSX controller to up, down, left, right, and fire on the computer, respectively.
2. If the playstation controller is put into analog mode, it will also map over the left analog control stick to the 4 up/down/left/right directions.
3. On a normal playstation controller, if the start button is pressed, the PSX64 goes into programming mode. The user can then hit L1, L2, R1, or R2, and program up to 127 buttons in for a macro mapped to that button. Hitting start again ends programming mode and saves the macro. These macros are saved in the PSX64's EEPROM and will be retained after power off.
4. If a guitar is plugged in, the PSX64 goes into guitar mode and maps the fretboard buttons into up, down, left, right, and fire. It encodes strum up, strum down, and lift up into one of the POT lines. It encodes the value of the whammy bar into the other POT line.
5. The PSX64 uses a socket mounted ATmega8 microcontroller, flashable with firmware updates with

easy to build programmers. It makes use of two digital potentiometers for accurate strum and whammy encoding. It does require a 9v-15v DC power supply to operate.

Shredz64 the game

Shredz64 is (will be if successful!) a game for the Commodore 64 that makes use of the playstation guitar controller hooked through the PSX64 interface. It is in the beginning stages right now, with only button testing and some graphics routines done, but is on a steady track. There is a good chance Shredz64 will make use of SID and/or MID files for its music, though with channel constraints,

SID files are more likely than MIDI files off the bat. Some obstacles to be worked through include presentation of music given the 3 voice nature of the 6581 and 8580. Calculation, whether static or dynamic, of fret button mappings to notes as well. Also, efficiency needed for quick polling of the guitar controller.

I'm keeping the graphics very simple, confined to PETSCII and sprites during the early phases, since I want to save all the processing power I can for the music and response. If I have some extra cycles left over at the end, I'll tweak and polish up the graphics. Anyway, you can see the (very) beginnings of how the main screen will look.

Why do it

Ha, this seems to be a question I get a lot. In December of 2006, I attended the TPUG World of Commodore convention in Toronto, Canada, and found hundreds of people still very much interested in the Commodore 64 and expanding its capabilities, even today. I saw some awesome demos of people building network adapters, online games, MIDI interfaces, and other cool stuff for their C64s.

I fell in love with the idea - I love the C64 and I love hacking around with software and hardware alike, so I figured this would be the perfect project! Around the same time I had also read about Jeri Ellsworth and her amazing work and success with recreating the C64, and I was just really inspired all around. If all goes well, perhaps I can present my little project at next year's TPUG conference!

Really, when it comes right down to it, I work on "more serious" projects all the time, for my day job, for private contracts, all the time. They pay the bills, and I do enjoy them, but I also like to do fun, wacky stuff. We live in a world where a computer older than 6-7 is considered useless, let alone 25 years.

Well, I absolutely love my Commodore 64, it has a lot of meaning and memories for me, and I love doing fun things with it - and there's no reason why it can't do a lot of cool things, even still. You don't always need a piece of equipment that can perform 4 billion calculations per second, sometimes 1 million is more than enough. Plus, come on, hooking a playstation guitar up to a Commodore 64 is just damn cool. :)

For more information head over to the SHEDZ64 website

<http://freedomirc.net/~megaboz/shredz64/>

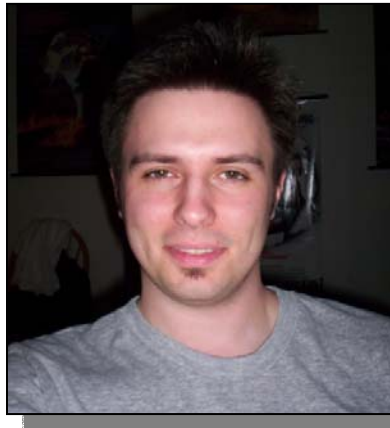
Interview with Toni Westbrook Shredz64 Creator

Q - Please can you tell our reader a little about yourself

I'm a 26 year old database developer living in a great little city on the seacoast of New Hampshire. I've been a Commodore (and classic computers in general) fanatic since my first C64. I'm a pretty big fan of life in general, and "geeking out" whenever possible.

Q - Please tell our reader about your full time job

I program database and web applications for Concord School District - (Concord is New Hampshire's state capital). I enjoy it as it allows me to work on a wide range of projects. I also run a software development company called Synthetic Dreams in my off hours which specializes in database development and neural network simulations. I'm pretty busy in general, but its fun stuff.



Q - How did you become involved with the Commodore 64

My family purchased a C64 when I was about 4, and it (and computing in general) pretty much became the focus of my life. It was the only computer I had until age 12, and I did everything I could with it. That little machine will always have a place in my heart, as corny as it sounds - even though parts are failing now, I'll always keep it.

Q - The Shredz64 Project what is Shredz64,

The Shredz64 project is a (successful now) attempt to connect the Playstation Guitar Hero controller to a Commodore 64, as well as write a Guitar Hero like game using the controller. The adapter I built to connect playstation controllers to Commodore machines I've named the PSX64, and the Guitar Hero like game I've named Shredz64. The adapter is great, as in addition to having extra functionality specific to the Guitar Hero controller, it also allows you to use normal dual shock controllers with any C64 (Or Amiga, Atari, or Sega Master System) game, and allows programmable macros and some other neat stuff. It's great to play old games with the new precision of a game pad.

Q - What would our reader need to try out your project

Currently I haven't released the source code or schematics for the PSX64 adapter yet - I'm waiting for the whole project to be done and all the little bugs to be worked out. I'm not an electrical engineer by any means, I have moderate knowledge of microcontroller programming and digital electronics, so even though everything works, I'd like to clean

some things up prior to public consumption, maybe get a few opinions of some EE people more knowledgeable than myself. Right now I'm focusing on the game since the adapter is 99% done; I have a target date of being done with everything by October/November.

Q - our reader may not have heard of the playstation can you enlighten him about the machine

The Playstation (Or more specifically, the PS2/PS3) is one of the most popular consoles on the market, along with Microsoft's X-Box line and Nintendo's GameCube / Wii systems. The PS line is great fun, as imaginative accessories like the Guitar Hero controller, or DDR pads, Or other devices are abundant on the system.

Q - The Shredz64 Project ok WHY ?

I went to a Commodore conference (TPUG's World of Commodore) in Toronto last December and realized for the first time that there were still a lot of people interested in using Commodore machines, even to this day. There were just a lot of imaginative things being done, network games and midi adapters and just really cool new uses and ways of pushing the C64 to its limits. Around the same time I had read about Jeri Ellsworth's amazing contributions to the field, and I just got (and still am) really excited that I wasn't the only person who still loved his/her Commodore, and decided that I wanted to actively work on something like these other bright people were. I think Guitar Hero is an awesomely fun game, so I figured there was no reason I couldn't combine the two. Then I realized how freaking cool it would be to play SID tunes with a Guitar Hero controller, and I was off and running. ;)

Q - the website says its about 98% functional can you explain what this means

Yeah, the 98% functional part describes the PSX64 adapter. It currently detects if you plug in a normal dual shock controller or a guitar hero controller, and operates differently accordingly. If you plug in a Dual shock, it allows you to assign button macros to the L1/L2/R1/R2 buttons using the start button, and converts L/U/D/R and X over to L/U/D/R and Fire, respectively. It also converts the left analog stick over to L/U/D/R when in analog mode. If you plug a guitar hero controller in, it maps all the buttons over, and also encodes strum up, strum down, lift up, and the whammy bar into the two analog pot lines of DB9 port, which Shredz64 reads. Really, the only thing I want to change about the adapter is protecting the MCU - right now due to a small design glitch, if you plug it into the computer without powering it from the wall, it tends to erase the programming on the MCU. It just needs a few diodes and all will be "perfect".

Q - What other developments are there to be made to the project

The game just needs to be finished. Currently, it draws the fret board, reads note data off floppy disk, glides the buttons down the fret board at the right times, reads in buttons being pressed on the guitar controller, increases your score and crowd meter for correctly playing notes, decreases your crowd meter for incorrectly played notes, and reads music (SID) data from disk. It currently plays the music too, but the timing is off as its not done. I need to complete the SID converter/player, and then the Shredocity meter (like star power), menus, general cleanup - add better graphics if there is any processing power left. Its coming along really nicely in general though. I've been programming it in a mix of C and 6502 assembly using the CC65 compiler on my Debian Linux box and a PC64 cable.

Q- If our reader is wanting to become involved in the project what would he need to do to help out

Just friendly words of support are great! I've had a couple people write with some helpful suggestions, or willing to do some graphics, that's always nice too. Honestly, until it's in the end phases, there won't be too much need of extra stuff, but until then I like to get notes just to know people are interested.

Q - Will this project be "mass produced" or is it just Designed to be a DIY effort

I haven't really decided about this yet. I will probably release everything as public domain, and then offer to build the PSX64 adapter for a fee. It's rough with the PSX64 adapter, as if you wanted to build one, you'd need an Atmel Atmega MCU programmer, which some people have, but I imagine many people don't. So those that do could build the interface and download the hex/source for the MCU and burn it on, and those who didn't could purchase it from me. Honestly, I'm not sure yet, but I will get everything out there one way or another! :)

Q - Our reader may be thinking "i have all the parts needed but cant solder" can you provide him with an interface

Thats the other thing, some people aren't big fans of soldering - so yeah, for these people assembling the PSX64 for a fee would work out.

Q - Was this an easy project or did you encounter some unforeseen problems

Parts were easier than I thought they were going to be, and other parts were very difficult. I had a hard time finding documentation about the protocol the PSX controller's use. There are lots of sites with general information about the pins and basic control codes for reading in directions, but when it came to sending commands to the controller, (eg forcing it into analogue mode or reading current mode, vibration, etc), I searched long and hard until I found it. Converting SID files to something usable in a very controlled application such as this one has been very difficult to say the least too.

Q - Please tell us a little about the design project - How does someone start to create such a piece of Hardware

Whenever I work on projects like these, I just take little steps. With this one, first I created a little circuit and MCU firmware to see if I could read the buttons being pressed on a PSX controller. Then I added to read the Guitar Hero controller. Then I made it play notes on a small speaker. Then it basically just grew and grew until the adapter was fully functional, then I started on the game. Its easy to get overwhelmed, but its fun when you split it into smaller projects, and easier to deal with the frustration when things don't work sometimes (often times).

Q - Are you working in any other projects Commodore or none Commodore

I'm working on a number of non-Commodore projects, but they aren't as fun - mostly database related. I have an ongoing project of building An accurate biological neural network emulator which is one of those lifelong things, I've been doing it for about 10-12 years. AI is another one of those areas that's always interested me.

Q - Do you own any other Commodore Computers apart from the Commodore 64 used in the project

Oh goodness yeah - I collect classic computers. For Commodores, I have my first C64, as well as another C64 and a newer C64C. I have a Vic-20, a C-128, and a SX-64 (great machine). In the Amiga line, I have a A1000, A500, A2000, A2500, and an A1200. I also have a Commodore 286 Special Edition. I have a range of Apples, Ataris and TRS-80s, but the Commodores are my favourites by far.

Q - Do you actively use Commodore machines

In my work area, I have my C64C and my Amiga 1200 right next to my Debian and Win2003 Server I use for my business. I don't necessarily use them in my day to day activities, but I can't go too long without turning them on and playing some games or just messing around.

Q - Ok I always like to ask "why did you think Commodore went wrong"

I think this is a very complicated question, and I only know what I've read, but a lot of it had to do with people in power like Irving Gould and even Jack Tramiel in the early days to a certain extent. It just seems like Commodore had all these brilliant minds, amazing potential, the power to absolutely rule the entire market, and the people who controlled the money just kept making absolutely bizarre and Destructive decisions. Here was a business that had its own chip making facility, had the #1 selling computer (of all time, to this day), and could have kept going with the Amiga - a machine that was years ahead of its time. It's really sad, and what's even more sad is the revisionist history of Apple being these visionaries and all the rest of the garbage you hear. Most people today think of Commodore as a gaming machine, if they think of it at all. I would love to see a world where Commodore made it.

Q - And " what would you do i you won 1 million pounds"

First thing I'd convert it to American currency, run my business full time, and hopefully make as big of an impact on the world as Commodore did. :)

The HEXFILES part 4

By Jason Kelk

'Ere mister, wanna learn machine code? Well you've come to the right place ain't ya? Yup, welcome to another installment of the Hex Files and, as is becoming fairly common ground now, I'll start off by having a look at the solution to the "challenge" (since some people have said they are too easy) set in the previous installment. If you remember, we had a simple scroll routine and I asked you to move it down a line and change where the data was coming from to \$e460. Here's the modified routine:

```

                * = $0900
                ldy #$00
main            ldx #$00
                lda #$fe
raster         cmp $d012
                bne raster
move_loop      lda $0429,x
                sta $0428,x          ;
read/write moved from $0400/1
                inx
                cpx #$27
                bne move_loop
                lda $e460,y          ;
reads the new "text" from $e460
                sta $044f          ; and this
used to be $0427
                iny
                jmp main

```

Right, at this point you're all saying something along the lines of "that's all very well, but what good is it?" On it's own not an awful lot, but if you start adding other things to it you get a little demo and that's just what we are going to do. Now I expect you are all wondering how I expect you to be able to understand the code for a demo, right?

Well, it's not going to be a complex demo, just something simple with sprites, a piece of music and a slightly more complex scrolling message than the one above. And you don't even have to type the code in, it's all available to download as we go if you feel lazy! So first, download this installment's sample data, extract everything to the same directory and drop demo_1.asm into a passing text editor so we can have a prod around.

First off, since this is such a large piece of source I'll cover it in chunks and any commands we haven't covered already will be explained on the way past. The first part of the source is responsible for initialization, the setting up of the computer.

```

scrollcount = $0340
messcount   = $0341
sineposx    = $0342
sineposy    = $0343

sine_curve  = $0e00
sine_curve_2 = $0f00

```

The above assigns the names scrollcount, messcount, sineposx and sineposy to various locations in memory and specify where to look for the two sine

curves that are about to be loaded in. Apart from the curves, these are our counters for various pieces of

the code and the memory they have been given is part of the cassette buffer so will cause no problems.

```

.incbin sprites.prg
.incbin sines.prg
.incbin music.prg

```

These three are "INCLUDE BINARY" file commands; they're not 6510 machine code instructions but assembler directives, in other words they tell C64Asm to load those binary files and include them in the final PRG it makes. In this case, they're the sprite definitions (at \$0c00 to \$0dff), the sine curves we'll use to make the sprites move (at \$0e00 to \$0fff) and finally a piece of music by Sean of Cosine (at \$1000 to \$1xxxx).

```

                *= $0900

                sei

```

As usual the * command tells the assembler where we want our code put, in this case at \$0900. And next we have a new 6510 command, SEI means SET Interrupt disable status and basically it turns off the C64's interrupts which would otherwise interfere with the running of our code. You may have noticed the occasional jump in the old scroller as the system interrupts occurred at the same point on the screen as the routine and we don't want that. The actual use of SEI (for setting up interrupts) will be covered later on.

```

clrpage        ldx #$00
                lda #$20
                sta $0400,x
                sta $0500,x
                sta $0600,x
                sta $06e8,x
                lda #$0f
                sta $d800,x
                sta $d900,x
                sta $da00,x
                sta $dae8,x
                inx
                bne clrpage

```

Okay, this is one of those loop thingies isn't it? Yes, just like the loops we have seen before it's just putting a repeated character onto the screen. It's a bit different in that it does the whole screen and the colour memory at \$D800 as well. You may be wondering about the last location being \$06E8, well this is a little trick; the screen memory is 1,000 bytes long and runs from \$0400 to \$07E7 and the maximum value a register can hold is \$FF so \$06E8 + \$ff is \$07E7. It saves having to do a shorter loop for the last bit of the screen. The value \$20 is the space character so the whole screen is cleared and

the value \$0f is going to the colour memory to set all the characters light grey.

```
lda #$00
sta $d020
sta $d021
sta scrlicount
sta messcount
sta sineposx
lda #$40
sta sineposy
```

This next piece of code puts 0 into the screen and border colours, making them black and also zeroes off our counters - at least, all except sineposy which we want to be different for the sine effect to work so it has a value of \$40 instead.

```
lda #$ff
sta $d015
sta $d01c
sta $d01d
```

Right, I've assumed that you all have some knowledge of sprites from the C64's manual but we'll clarify a little to be sure; the above four lines will, in order, turn on all eight sprites (\$d015), make them multicolour (\$d01c) and also alter their priority to put them under any characters on the screen like our scroller (\$d01d). The eight bits in each of these locations represent one sprite, so the first sprite is the lowest bit and the last is the highest; since a value of #\$ff has all of it's bits set that means all eight sprites are being turned on and then all have their multicolour enabled and are set to "low" priority.

```
setsdp      ldx #$00
            lda #$30
            sta $07f8,x
            clc
            adc #$01
            inx
            cpx #$08
            bne setsdp
```

Next we set the sprite data pointers (S.D.P.'s for short) using another loop. But this time there are two commands we haven't covered again. These are CLC (CLear Carry flag) and ADC (Add Decimal Accumulator). The former's job is to empty the carry flag, which is one of a series of flags the C64 uses to remember things (in this case, it's an "overflow" for various calculations). The reason for it's use here will not be immediately apparent until I explain the ADC command that follows it.

ADC #\$01 will add one to whatever value is in the accumulator, so if it's presently \$30 as at the start of the loop the ADC command will make it \$31. It's like the INX command except that by changing the value you can alter how much the A changes by and, if the carry flag is set from any previous job, then it will be added in too! Since we don't want this to happen here because the carry will only have been set by something we don't want to affect this particular job, we use the CLC first.

But what does this loop actually do? Well, it puts a value of \$30 into \$07F8, a value of \$31 into \$07F9

and so on until it puts \$37 into \$07Ff, all of which point the VIC to \$0C00, \$0C40 and so on to \$0dc0 for the sprite data. If you look at those sprites in memory they spell H E X F I L E S (we have put in two E definitions to make this a bit easier for now).

```
spritec     lda #$0b
            sta $d025
            lda #$0f
            sta $d026

            ldx #$00
            lda #$0e
            sta $d027,x
            inx
            cpx #$08
            bne spritec
```

Now we have to set the sprite colours. \$D027 onwards govern the sprite colours and the value of \$0e will produce the same light blue as the border colour when the C64 is turned on. The multicolours are set to dark grey and light grey so the sprites can have a nice shading effect.

```
lda #$00
jsr $1000
```

The final part of the setup we will cover this issue is another new command. JSR means Jump to Sub Routine and it's the machine code equivalent of the BASIC command GOSUB. Up until now we have used the RTS command to drop back to BASIC after our code has finished, but if we were to JSR to a piece of code an RTS would return us back to the next command along so it allows us to pop off temporarily and do something else, then come back and continue where we left off.

In this case it calls the part of the music routine (it was included earlier with the INCBIN command, remember?) that sets the music up for playing and turns the volume on. The LDA #\$00 is to tell the music routine that we want tune 0, the first one it has and later on there's a JSR \$1003 that actually plays the music.

Right, that's yer lot for another time, I hope you're all following okay. If you want to, you can try to carry on alone and examining the rest of the program, although there are some new commands I will cover in the next installment. And your mission, should you decide to accept it, is to try and change the sprite colours to \$04 (for purple) and the message colour to \$01 (for white). Good luck and, as usual, you can email me if you have any questions on this installment. This web site will self destruct in ten seconds.

The source code for the routines above can be downloaded here

http://www.oldschool-gaming.com/files/c64/hex_files/part_4_files.zip

for easier reference.

© Jason Kelk
RE-printed with the permission of the copyright holder,

Commodore MMC2IEC DEVICE

The MMC2IEC device

It has always been tricky to transfer data between a PC and a Commodore 64. As the old machine lacks any even remotely modern interfaces, tricks of some sort are always required to do the transfer. The MMC2IEC device can be seen as yet another one of these tricks, but it does so much more.

Description

The MMC2IEC device tries to simulate a 1541 disk drive connected to the C64 using the IEC bus and accessing data from a SD/MMC flash memory card. The MMC2IEC is both greatly limited and greatly extended in features when comparing to the original 1541:

Features:

Supports C64 kernel LOAD and SAVE functions.
Supports wildcards.
Supports all? SD and MMC cards. Up to 1 GB tested.
Thorough FAT16 and FAT32 support.
Directory listing and changing supported through LOAD commands.
PRG file load and save.
D64 image readonly support. LOAD"\$" produces a realistic listing when a D64 image is active.
T64 image readonly support. LOAD"\$" produces a tape contents listing.

Limitations:

No fast loader support.
None of the Commodore DOS functions beside LOAD and SAVE works.
Few, if any, D64 games with loaders work.
No long filenames.
SD card write protect switch not (yet) implemented...

Background and credits

The idea of transferring data to the C64 from a flash media was conceived long ago as I did an university project. In cooperation with other students we created a data logger interfacing a SD-card using an AVR. The SD-card data access and FAT driver modules were created back then and they were thoroughly tested. The FAT driver is not completely my own invention, it is a partial rewrite of work done by Angelo Bannack and Giordano Bruno Wolaniuk who used work done by Pascal Stang. See fat.c for details.

At some point I discovered Jan Derogee's 1541-III device which was a major inspiration. Before that I considered creating a C64 datasette simulator, but the 1541-III made me realise that it was possible to simulate a 1541.

The MMC2IEC project thus started as an attempt to port Derogee's 1541-III from PIC to the Atmel AVR architecture such that I could combine it with my FAT and SD-card modules. Soon enough I gave up the porting and implemented the IEC and 1541simulation from scratch, but it would have been impossible without referring to Derogee's project all the time.

At a later point I realised that the MMC2IEC was perfectly suited for embedding in a C64-DTV mod. And so far I have only tested the MMC2IEC on a DTV in conjunction with my Wooden DTV mod. This project is open source under the GPL license.

Usage guide

The MMC2IEC device implements accessing FAT16/32 filesystems on MMC or SD flash media. The CBM IEC protocol is implemented and the kernel load and save routines on the C64 works, loading from either PRG, D64 or T64 files, and saving only to PRG type files in FAT.

Selecting directories and images is possible through LOAD"xxx" commands issued on the CBM.

This is a quick overview of the commands, say MMC2IEC is IEC device 8: Read < as the petSCII back-arrow.

LOAD"<<",8 Reset SD card state. Do this if the SD card is exchanged.

In FAT mode: (the default mode)

LOAD"\$",8 Gets directory listing, equivalent to LOAD".",8

LOAD"gamesdir",8 Enter the "gamesdir" directory, and get listing.

LOAD"..",8 Up one directory and get directory listing.

LOAD"tetris.prg",8 Loads the "tetris.prg" program file.

SAVE"example.prg",8 Save into "example.prg" which is a FAT file.

LOAD"disk.d64",8 Loads the disk.d64 image and enters D64 mode.

LOAD"tape.t64",8 Loads the tape.t64 image and enters T64 mode.

In D64 mode:

Load "\$", "!", wildcards, filenames works (almost) as expected on a 1541.

LOAD"<",8 (back-arrow). Escape D64 mode, and back to FAT mode.

SAVE"abc",8 Fools the CBM, but has no effect. Saves in D64 are not implemented

In T64 mode:

Load "\$", "!", wildcards, filenames works as if it was a D64.

LOAD"<",8 (back-arrow). Escape T64 mode, and back to FAT mode.

SAVE"abc",8 Fools the CBM, but has no effect.

Make your own MMC2IEC?

Releasing all this detailed information about my MMC2IEC device is also to encourage people to make their own devices. I would be happy if others benefit from my work on the MMC2IEC project. All I require is that proper credits are given. If you build a MMC2IEC device or just use some of the code modules, please email me with your experiences. I would also like to know if you improve the code. Happy Hacking!

By Lars Pontoppidan Feb 2007

<http://pontoppidan.info/lars/index.php?proj=mmc2iec>

Interview with Lars Pontoppidan Commodore MMC2IEC DEVICE

Q - Please introduce yourself to our Readers

Hi, my name is Lars Pontoppidan (larsp). I'm currently studying at the Technical University of Denmark, and I'm almost finished with my M.Sc. in Electrical Engineering.

> Q - What is your fascination with Commodore

I think there is a great deal of nostalgia involved, as the C64 was the computer I grew up with. But that's not all of it. Doing stuff with the C64 is always fun, because the software and hardware is so crude and simple, in stark contrast to modern PCs. I have had many great laughs with Spiff discovering the absurdities of the hardware and how hackers fooled the ICs to do more than originally intended.

Q - MMC2IEC device Please can you explain it function to our reader

The MMC2IEC device simulates a 1541 disk drive and allows loading and saving of program files to MMC/SD flash media. Only the kernel load and save is supported though.

Q - Hasn't this been done before though?

Yes, it has, multiple times I guess. For instance, Jan Derogee's 1541-III is another popular device with similar functionality, but it is more complicated as it requires a LCD and buttons.

Q - What makes your device Special

Well, I think MMC2IEC has the best file system support of the bunch: FAT16/FAT32 read/write with directories and fragmented files, though no Win95 long filenames. The flash card driver has also proven to be strong, as all MMC/SD cards I have encountered were accessible by the device, even a 4 GB one. And then there is the simplicity of the hardware.

Q - is the design easy to use, for example does it use standard Commodore basic load, save style commands

It is very intuitive I think. You can LOAD"\$",8 to get the current dir listing. If you load a directory, then you go into that and get the listing of it. LOAD"..",8 will get you back. If you load a .PRG, you load the program, if you load a .D64 you "mount" the disk image and get a realistic listing of the disk contents, etc. Only save to .PRG is supported though.

Q - is your design copyrighted

I did it, so I'm the copyright holder.

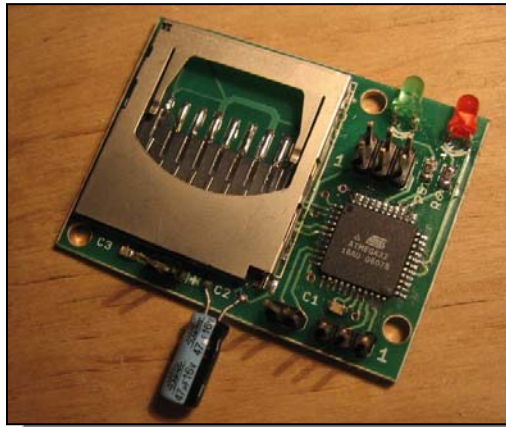
Q - why did you go open source for the project

Well, the FAT driver in MMC2IEC is "heavily inspired" by another open source FAT driver as I call it. I restructured the driver, fixed a (large) number of bugs, and implemented new features. It was only fair that I kept the GPL status for my improved FAT driver, and thus, the entire MMC2IEC project had to be open source as well, according to the GPL. But open source is fine with me. In general, when you do hobby projects like this, I think you should share the source. Unless you are ashamed of it, of course :)

Q - Are there plans to Sell the device commercially

COMMODORE FREE
Late breaking news See footnote about sales of this item

Q - Please explain to our reader what is an MMC



It's an early flash memory standard (MultiMedia Card). But MMC2IEC is compatible with both MMC and SD cards, as they both implement the SPI interface. I just thought that MMC2IEC had a better rhythm to it than SD2IEC ;)

Q - Are these Cards easily obtainable?

Indeed. SD cards are among the cheapest flash storage available.

Q - Can MMC cards be plugged and unplugged

while the Commodore is still running, like normal floppy disks

You can do that. Often, when the SD card is inserted, it sucks such a great deal of current to start up, that the microcontroller resets... This may sound like a problem, but it isn't. When the controller resets, I initialize the SD card and everything is fine. If the controller doesn't reset however, it is required to say "reload flash card" to the MMC2IEC device. In a future firmware version it should be possible to do this automatically.

Q - Can you talk our reader basically through the Design phase of the device

Well, I had the MMC/SD card driver and the FAT driver up and running on the AVR in connection with a project I did on the university. So "all" I had to do, was to implement the IEC protocol and simulate a 1541. The first attempt was to port Derogee's 1541-III from PIC to AVR, but I gave up on that approach. Instead, I chose to implement the IEC protocol and 1541 simulation from scratch. That was like a near-death experience to me.

IEC handshaking and the command communication is EVIL. But after a few rewrites, I got it working. After having the software working, I moved the circuit to

PCB, such that I could use my breadboard for other stuff ;)

Q - Why design the device what gave you the idea

As I did the SD card interfacing and FAT driver in the university project, I got the original idea: a datasette simulator loading from SD card. I had already messed with a little converter that enabled using a CD-player for datasette loading. Then I stumbled on Derogee's 1541-III, and realized it was possible, and actually made much more sense, to simulate the 1541 drive instead of the datasette. From that point the rest of the design just followed logically.

Q - Do you still use commodore computer then

On a hobby basis, yes. I spend some time with Spiff developing stuff for the DTV, and from time to time I get the C64 out of the closet to test stuff and perhaps play bubble bobble with a friend.

Q - You have other hardware and software on your site please tell our reader about some of your other projects

Well, recently Aske Olsson and I created a snake robot: SickSack, for the DTU Robocup contest. It was a super fun project, and we got a lot of exposure because the robot is quite unusual. Eight servos connect nine cars which have passive wheels and the snake motion itself makes it move forwards. We won the Best Design and Effects award!

Another project that got a lot of attention was my Denver DVD hack. I was mad at the user interface on a cheap DVD player. So, to fix it, I programmed a microcontroller to scan the buttons myself and generate fake remote control signals to the DVD. The controller was also programmed to decode incoming IR commands and pass them on. I also made it do super smooth 16-bit PWM on a led...

Q - On your website you say you are looking for work would you like to advertise your skills to our reader

Well, I will finish my Master's of Science in Electrical Engineering degree May the 9th, and then I'll be visiting California for vacation and to talk to companies. I hope to find a job there.

My strongest skills are probably that I'm a quick learner and a talented programmer. At least, those abilities got me through the Master's degree with relative ease, and with enough spare time to do a lot of hobby projects.

I'm interested in electronics hardware and low-level programming matters, and so far I have focused on microcontroller programming/development. Read my resume:
http://pontoppidan.info/lars/maincontent/cv_english.pdf

Q - Do you plan to have any other Commodore related projects

Spiff and I are currently hacking together a menu-loader for the DTV that will enable loading from the DTV flash as well as browsing MMC/SD cards via MMC2IEC, controlled by the joystick alone. One of the goals is to enable loading games from MMC2IEC without hooking up a keyboard.

Q - What would you like to design for Commodore

machines assuming you had unlimited time, staff and funds

I think that's a paradoxical question, because doing stuff with a C64 is all about doing incredible things with limited resources. If you had unlimited resources, well, I honestly think you should spend them on something else than a C64.

Q - Have you had any good or bad feedback from Commodore users

Yeah, both, most good feedback fortunately :). A couple of other persons have created MMC2IEC devices and we have discussed problems and stuff in the petscii DTV forum.

A nice thing about MMC2IEC is that the hardware is so simple. When interfacing the 3.3 V DTV no level conversion is needed, so the hardware consists of nothing more than the Atmega32 controller, a SD-card socket and some decoupling capacitors basically.

Some have had problems with the stability, and some had no problems at all. With the newest firmware stuff seems to be work quite well though.

Q - I noticed the following statement on your website "Out of the box, the PAL DTV creates horrible colours. This is not because of price cuts or design intensions, but because of mistakes on the PCB." Please can you explain the problem to our reader

Well, to generate the PAL video signal, the DTV utilizes a D/A conversion principle known as a R-2R ladder. Or rather, it is supposed to. A correct R-2R ladder is a network of resistors with values R and 2R arranged such that an analog voltage is created from digital signals with weights 1/2, 1/4, 1/8, 1/16 etc. This is a simple and efficient way of D/A conversion. The problem with the DTV is that the 2R resistors have value R, and the R resistors have value 2R! That mistake pretty much ruins the linearity of the D/A conversion, and results in the strange colors on a vanilla PAL DTV. It can be fixed by adding some SMD resistors.

UPDATE

**Lars has informed me these devices are now for sale I have ordered one and will update you with how I get on using the device thanks Lairs
COMMODORE FREE**



<http://pontoppidan.info/lars/index.php?proj=mmc2iec>

Interview with Arno Weber

Boulder Dash fan

NP: Please introduce yourself to our reader and give a little background about yourself.

AW: Hi, I'm Arno, born in 1981 in Utrecht, The Netherlands. In daily life I'm a student in Mathematics, but right now, in order to finish the study, I am doing my master thesis at some consultancy company. Next to this work I have several hobbies. One of them is retro gaming, especially Boulder Dash!

NP: What is your website all about?

AW: My site is dedicated to the classic game Boulder Dash. Although the focus is on the Commodore 64 version, I will probably pay some attention to other versions and clones as well. For those readers who don't know the game, it's a platformer in which you control a guy called Rockford through a series of caves.

In order to complete a cave a certain amount of diamonds has to be collected. When this amount is collected an exit opens which leads you to the next cave. During the search for diamonds you meet different obstacles like falling boulders, walls, and deadly fireflies and butterflies. On the other hand, you often need these items to reach the diamonds, or to create new diamonds.

NP: So Boulder Dash what is all the fuss about, what makes this a great game?

AW: I think it is so popular because the game is both simple and complex. The levels of the original BD (also called "caves") are made up of only 10 different items. Still, by combining these items, it is possible to create many different types of caves.

Because each element has its own specific properties, a Boulder Dash cave is like a puzzle that must be solved by logic/strategic reasoning. This makes the game addictive, which is probably the basis of the success. For example, a firefly is a deadly enemy, but at the same time, these animals must be used to get behind a wall (by explosion).

Also the difficulty of BD caves is extremely varying. Some of them are easy ("just get a few diamonds and find the exit"). Other caves are a challenge for the most experienced BD players.

Another reason for the success is the existence of Boulder Dash editors and many other tools, which enable the fans to create their own (standalone!) Boulder Dash games. Thanks to these tools a lot of fan games have been created over the years.

NP: Can you tell our reader a little about the Boulder Dash history?

AW: The first Boulder Dash game was brought out in 1983 for the Atari 800. Peter Liepa wrote it together with Chris Gray, and First Star Software released the game. The conversions for Commodore 64, Amstrad CPC, ZX Spectrum, and other platforms, were released a bit later. 1985 was the year of Boulder Dash II and 1986 the year of the Construction Kit. But

even before that time a lot of fan games were created because the authors simply copied the Boulder Dash I engine and edited the caves.

Most fan games were made for the Commodore 64. Also for this machine a lot of useful BD tools were

created. By the late 80's and early 90's some tools appeared that could be used to create standalone games out of caves designed by the Construction Kit.

Since the rise of the internet in the late 90's, Boulder Dash seems more popular than ever before. Many websites devoted to Boulder Dash were launched, and via these sites the tools were more spread out than before that time. At

<http://www.gratissaugen.de/erbsen/>

you can read more about the history of BD at the section "Highlights and flops" (unfortunately this is only in German at the moment, but very nice to read). Last year 4 new BD fans have released their first games – all inspired by the websites! As for now (2007) more than 1000 fan games are available.

For those who want to download them: the most complete collection can be found at Martijn's BD fan site:

<http://www.bd-fans.com/>

NP: I notice you design your own Boulder Dash games. Can you let our reader know more, are they freely available for download?

AW: At the moment I have created 24 Boulder Dash games, each consisting of 16 caves and 4 intermissions. (Intermissions are small "caves" in which you gain a free bonus life, and mostly get more points per collected diamond.)

All games can be freely downloaded at the "games" section. The biggest series consists of Arno Dash 1 to 20. These games usually have medium difficulty. Two other games are Exploding Dash 1 and 2. If you play these you shouldn't be afraid of fireflies and butterflies since there are many of them! Furthermore I have created 2 games with a special theme. Santa's Boulder Dash Party is a game with X-mas graphics. Future Dash contains levels of which the border is "open".

So if you leave the screen at one side you return at the other side. The aim of this game originally was to show how many possibilities for new levels arise when just one new feature is added to Boulder Dash!

NP: What tools are available for Boulder Dash?

AW: For C64 there exist some very important tools. Let's start with the Construction Kit. This tool can be used to design your own BD caves and save them on disk. Some authors have released alternative construction kits that feature more items and more effects. A recent release is the Crazy Light Construction Kit 3.0 by Marek Roth. This kit is most complete with respect to the available items, cave properties and effects. Although it is still in development it's currently possible to use it to create BD games.

There are also tools to make a standalone game out of a set of caves; so called packers. A famous packer is Deluxe Packer, which I mostly use to pack my games. Also the Crazy Light Tools package contains a program to pack the caves designed with the Crazy Light Construction Kit.

Furthermore there are tools available to design your own graphics-sets and font-sets. Using such a tool you can create a game with Rockford in space, or dressed like a pirate, or whatever you can think of!

Another useful tool is the Boulder Remake converter. This tool belongs to Boulder Remake, a very good remake from BD for pc (more info at

<http://home.deds.nl/~bremake/>)

With this tool you can convert a C64 BD game to the format of Boulder Remake.

NP: I notice the website has maps for the various levels. How easy was the process of creating these maps?

AW: Well, in fact I was already dreaming of having such maps when I played BD as a kid... Maps were fascinating me at a young age (I was drawing maps of all places I visited, and also designed my own cities and other things by making maps). So having a map of the Boulder Dash game would be fantastic!

A bit later, in 2004, I noticed that Boulder Dash games could be converted to some special format, called BDCFF (Boulder Dash Common File Format). This is a text format, which describes the levels of a BD game by its properties and the rules that generate the map of each level. There were some tools that could read BDCFF files and print maps of the levels on screen,

but unfortunately, these tools did not use the original colors of the caves. Therefore I started in 2006 to write a program by myself: the BDCFF2BMP converter. The program reads a BDCFF file, creates BMP pictures of the maps and produces an HTML file in which all the maps can be viewed together with the level properties. After creating some the maps using the converter, I printed out these files and put them in a "book". This book was exactly what I wanted to have when I was an 8 years old boy... :)

Now the maps of 190 different Boulder Dash games can be viewed and downloaded at my site. The hardest step in the conversion process was actually the creation of the BDCFF files. This happens technically by using a C64 memory snapshot, which is taken after a game has been loaded. But since the caves are generally not stored in a standard way in the snapshot, it is not always trivial to make a BDCFF out of it. Some games need an individual "treatment" before the BDCFF can be created.

NP: Do you have a How-to guide that walks the player thought each level?

AW: Sure, two pages are devoted to Boulder Dash I and II, respectively. Each page contains a walkthrough of all the caves and intermissions, complete with screenshots. In addition, I have video recordings of the solutions to both BDs. You need a media player to watch the solutions.

NP: I find early levels easy then get frustrated as the difficulty curve kicks in, the problem is the puzzles seem so easy when the level loads, I suspect this is good game design keeping the player guessing and going back for more, would you like to comment?

AW: As I said, in Boulder Dash there are both extremely easy and extremely hard puzzles. And it's true that many caves contain one or a few "surprising effects", such that it turns out to be harder than it looked like at the beginning. I can't say whether or not this is the feature that keeps the players in front of the screen. To take myself as an example, I mostly play BD games to try the caves one by one. My goal is usually to complete all the caves, rather than going for a high score. If the caves are well designed and contain nice puzzles, then I will keep the game and play it regularly amongst others.

NP: How would our reader create his or her own Boulder Dash game?

AW: Everyone can make his/her own Boulder Dash games! You have to use a few of the tools mentioned above. Most of these tools are easy to learn and quite self explanatory. Tutorials are currently not available at my site (there will be later on), but of course I'm always willing to help in case of problems. We don't want to miss any BD creations simply because the tools seem too complicated to the public!

NP: Are there any tricks or tips that are useful for playing the game?

AW: Well, euhh... there are many tricks that can be useful in general. An example is that you can get rid of a firefly by, for instance, killing it with a boulder, or dig a round tunnel such that it keeps spinning around. In caves which are very tight because of many boulders I always try to work from the lower area to the upper area, because it's much simpler to drop boulders than trying to make a way through the cave when the spaces are very tight.

But overall I think each cave is unique and needs its own approach. Some caves have unavoidable "bobby traps". Then you first have to "learn" the cave at the cost of one or a few lives.

To give some hints: when I play a new cave I usually try to get an overview by just walking around. But first I look at the information bar above the screen, saying how many diamonds I need to collect and how much time I have. Also I try to find out quickly where the exit has been placed. Sometimes this is easy because the exit looks like a titanium wall. But in other caves the exit is hidden somewhere on the border, or it could be placed anywhere because the cave contains many titanium walls.

NP: Do you play any other games?

AW: I have been playing uncountable C64 games. My favourites seem to be the platformers like Ghosts 'n Goblins, Montezuma's Revenge, Yie ar Kung Fu, and Who Dares Wins. One of the best games (i.m.h.o.) to play against the computer or with 2 players is Hat Trick Ice Hockey.

Another game, which is less known, but I like very much (because of the puzzle aspect), is Sensitive. In each level you have to move an ufo-like object from one place to another by flying over tiles. The tiles explode when you fly over it. Some of the tiles can be used only once, others twice. Unfortunately this game has no sequels, and there is no editor. Otherwise I would have designed tons of levels!

Next to these C64 games, I play one pc game quite often, which is Revolt (1999). This is a racing game with small RC cars. There are race tracks in which the cars are driving through supermarkets, museums, wild west towns, and cruise ships. Optionally there are pickups that you can use against your opponents. For example, you can shoot at other cars, or let them slip through an oil puddle. It is also possible to create your own race tracks. There is a community of fans releasing their self-made tracks at some websites. I think about a few hundreds of tracks can be downloaded there.

NP: Do you own any Commodore machines?

AW: Yep, I have a real Commodore 64, but I don't use it very often. Most of the C64 games that I'm playing regularly are downloaded somewhere from the internet, so I use the VICE emulator the play them. Also I design my Boulder Dash levels within VICE.

NP: The best till last! If you had 1 million pounds what would you do with it?

AW: Euhhh.... change it for euros! ;

Interview with Shaun Bebbington

Retro promoter and Writer

Commodore Free)

Please introduce yourself to our reader.

Shaun Bebbington)

Hello all. I am Shaun Bebbington. Just call me Shaun, Bebbers, Fluff or Picto. I am a part-time writer and columnist for Micro Mart magazine, available throughout the UK via WH Smiths and the likes.

CF) What are your first memories of Commodore?

SB) That would be one Christmas, I think 1985 though I'm not entirely sure. My Dad brought a Commodore 64 from his brother who had recently upgraded to a flat Commodore 128 with a 1541 disk drive. The first game I remember was Beech Head II. Brilliant loading music, graphics and speech it had. The cries of "Medic!" still ring today.

CF) Do you still use Commodore machines actively? And can you tell us about your various set-ups?

SB) Currently, I don't have a place of my own as such, so I am a "sleeping" Commodore user, as all of my machines and hardware items are in storage. By the time you are reading this, I would like to have my Commodore 128D-Cr, which has the internal 1571 removed and attn line cut, up and running, with a SuperCPU 128 and RAMLink MK-II as well as a CMD-HD. I also have a 1581 and FD-2000 (note, no 5.25" floppy disks here), IDE-64, SCSI CD-ROM and ZIP drive, SmartMOUSE and a few other hardware items. I hope to add a StereoINSid into this mix, just for the fun of it. In for a penny, in for a pound, that's what I say.

When this has happened, I'll be running Wheels 128, which is basically a step up from GEOS. To find out more about GEOS, www.cmdrkey.com should have the answers, though I'm not sure if you are able to order anything from the site yet.

CF) Do you own other retro machines?

SB) Yes, quite a few. I own every Sinclair machine less the ZX80 (and to be honest, I don't want a ZX80 anyway), most Commodore machines except the rare ones and the C16 (I have a few Plus/4s though), and an Amstrad CPC for shame. It's rubbish, of course!

CF) You have written for many magazine would you like to tell our reader about some of them?

SB) Well, I've been published in PC Mart, Micro Mart, Retro Gamer, gamesTM, PC Extreme and PC Action Emulate, and a few fan-based publications such as ZXF, ZX Shed and Commodore Scene. I've been regularly published for over five years, though I was first published some 10 years ago in PC Mart. I first started regular writing for Micro Mart, issue 686 to be exact, and through that I was asked to set up a "Retro area" for the two Micro Mart computer shows that they ran in 2002 and 2003 I think. During the

latter one, a guy named Martyn Carroll introduced himself to Allan Bairstow and was talking about this new magazine called Retro Gamer. I butted in, telling Martyn that I was a writer, and importantly that I don't

write for free. I even got a mention in the very first issue.

For issue two, I wrote the Commodore feature. This was a bit of a nightmare. Firstly, Martyn asked me to write 7,000 words. I had never written that many words before or since, not for a single feature. Anyway, at first I was told that the deadline would be ages away, some two months. But things were happening at Live Publishing, and the editorial deadlines were brought forward to increase the frequency of the magazine.

All of a sudden, I had two weeks to write more words on a single feature that I had ever done. I booked a few days off from work, and was even working at my partner's parent's house whenever I could. Alas, I lost some of the text during the process as I was using Notepad, and I didn't realise that this package on Windoze 98 wouldn't allow you to write much more than 5,000 words without loosing some of the text. So, I had to rewrite a whole section before sending my work to the three Commodore experts that I trusted, being Allan Bairstow, Bo Zimmerman and Robert Bernardo, to mull over. Unfortunately, the deadline finally caught up with me and I had to submit it warts and all. I didn't realise that Retro Gamer were going to put me as a Commodore expert. For the record, I AM NOT a Commodore expert. I am merely an enthusiast. Okay!

Anyway, from there, I became a staff writer on Retro Gamer, and officially worked until issue 12 when I was made redundant. Though I continued writing for the magazine unofficially until issue 15. I just didn't allow Martyn to use my name.

After being made redundant, I did some freelance PR work for Cronosoft and the Alten8/Retro-Soft group. By doing this, I got to know Darran Jones, who was the "Retro" editor on the magazine gamesTM. Shortly after, I was writing for gTM. It was quite a relief as I felt frustrated after my RG experience. Anyway, I wrote an article about the Commodore VIC-20, Atari 2600, Amiga A500 and ZX81 (not necessarily in that order). The VIC-20 piece is my best work ever, and I'm not sure if I'll ever write anything more worthwhile, though the ZX81 feature comes close. I also represented gTM to a degree at the CGE-UK 2005, which was great fun though I was a bit hung over. Did anyone notice?

I'm happy to consign the rest to history for now. It was good while it lasted.

CF) Which magazines do you currently work for?

SB) Just Micro Mart presently. I have no plans to work for anyone else.

CF) Is all you work Freelance?

SB) It currently is, though I was a staff writer on Retro Gamer for all of nine months. Those were the days!

CF) Some of our reader will recognise you as the voice of the retro section in Micro Mart - although its only a one page section in a PC magazine. Do you

think it's important to keep pushing the retro aspect of computing?

SB) Absolutely. With so much happening, it's important that there is at least one national magazine which is taking retro computing and gaming seriously beyond the usual collectors/eBay/nostalgia slant. At first I didn't have much faith in the home-brew scene, but then I played a demo of the forth-coming Metal Warrior IV.

This was a brilliant little game, and I instantly downloaded the first three titles, which were all equally as good. Around that time, Cronosoft were just starting up. Their aim was to offer new software on computers such as the Sinclair ZX Spectrum and of course the Commodore 64. Their first game was EggHead in Space, which was a nice little game for the Speccy. I asked Colin Woodcock to review it for me as he was a Speccy enthusiast and I wasn't, and he was and is one of the best writers that I know. His review was brilliant.

After this, I tried to promote home-brew software as much as I could. My aim was and is to create a big enough market for bedroom programmers to earn a few quid and for Cronosoft's activities to be viable beyond a small niche. The fact that Retro Gamer seem now to be taking home-brew software a little more seriously is a good thing.

Hopefully, sales will continue increase steadily and people will start using their old machines again rather than just collecting them. Speccy fans are especially spoilt at the moment I have to say, with Jonathan Cauldwell the standard barer not just for that machine, but for all 8-bit home-brew programmers. His ideas are thought provoking and well worked even if you don't like his games. Even an ardent Commodore enthusiast like me has to acknowledge that.

CF) Can you tell our reader a little of the history of Micro Mart, and how you came to work on the retro section?

SB) Micro Mart started in 1984 or 1985 I think, offering "free adds" and numerous typos. It was like the eBay of it's time for people interested in trading or selling computers. It's done extremely well to have survived over 20 years, and as the small-adverts have dropped, the editorial has increased. MM is a free-thinking PC magazine that also covers Linux, Amiga and even MAC.

Freedom of speech, or in this case freedom of print, doesn't work unless the editorial process allows it. Thankfully, in my case it does, and I'm given a free hand in what to write about. In other words, I write the column because I want to cover what interests me and not what my editor tells me to write. It's a very good situation, and is like working on a fanzine only on a bigger and more professional scale.

As for how my Retro coverage started, well I did write into Micro Mart after seeing some retro features that they wrote, and noting their Amiga page. I told them about the SuperCPU and GEOS, Protovision and other such things. Simon Brew did respond to me personally, but was thinking of how to implement it, and after all I could only cover Commodore stuff at the time, or so it would seem.

A few months later and I started working on an electronic fanzine called Retro Computing Today.

This was an eight-page sampler for people to download, and covered formats such as the Atari Jaguar, MSX and Commodore Plus/4. As it turned out, Simon (Brew) downloaded a copy and liked it. He could now see how a regular retro column would work, and asked me to work for him on an eight-week trial basis. I did, and I was never asked to stop, so I guess the trial is still on-going.

That is, as long as I continue to find new content, the column will thrive. Though I can't do that unless programmers and hardware hackers continue to do what they do. I'm sorry to say that I don't tend to cover demos, but I do cover new games and hardware, as well as modifications, such as adding an internal CD-Rom drive to a C64c. So, if you're up to anything in this regard, please get in touch. My email is shaun@micromart.co.uk.

CF) Do you want to plug the Micro Mart website?

SB) Looks like you want me to. Okay, head over to www.micromart.co.uk and if you're lucky you might be able to read my column without buying the magazine, though you'll have to do a site search for either "retro" or "shaun" – while you're there, also check out the retro forum. Everyone there is friendly, honestly.

CF) Can you tell us about retro gamer, you were a staff writer, what went wrong?

SB) I was a staff writer for nine months. As for what went wrong, well I probably wasn't good enough and I couldn't compete with the freelancers. Martyn was trying to keep peace with them as best he could as far as I could tell, so if someone emailed in suggesting a feature, he wouldn't say no in case someone would kick up a fuss if he, Aaron Birch or I wrote "their idea". This had already happened anyway in the very early days when Martyn was working on his own. Anyway, I was made redundant because I was redundant. But if I had been there until the bitter end, I would have been worse off, that is for sure.

CF) Were you upset and bitter about your experience at Live Publishing?

SB) It seems popular to hate Live Publishing to be honest, and although there was one or two people I wouldn't want to speak to again, the vast majority of the staff there were friendly, creative and just good people. I was more frustrated with the job than anything else if truth be known.

The freelancers had too much power in my opinion, however that was Martyn trying to keep the peace as much as anything else. But of course loosing your job is always upsetting; however there was simply no more I could do there. I had written all of the features that I wanted to, and didn't like penning anything above 3,000 words anyway as I would loose interest quickly. Live liked long features and that was that.

The gTM word limits and house style are more like what suites me, as I proved with the VIC-20 and ZX81 features that I wrote for them. But now I really have written all of the features on my wish list, so there is nothing else but Micro Mart. It is time for others to shine.

CF) Are you still owed any money, and did it all finish on friendly terms?

SB) I'm owed money from Live and Paragon, not that I care much about that now or even then. Nothing I can or could do, you see. Though I wasn't happy at first when Imagine purchased gTM as they printed my work without crediting me. They've reprinted stuff since without any credit to me, but this works out for the best anyway. People ask questions when my name appears in other places apart from Micro Mart, and I only intend to write new work for Micro Mart these days as I feel much freer that way.

CF) You promote all retro machines. What is your favourite machine?

SB) Many for different reasons. I have a real soft spot for the CBM/PET and especially the VIC-20. The VIC is my favourite Commodore machine I think, closely followed by the C128. I'm probably more of a fan of Jonathan Cauldwell's work than of the Speccy, but then I prefer American machines with real hardware such as sound and graphic chips. Anyway, that's not saying that I hate the Speccy. So there you go, I don't have a definitive answer, which says a lot. As long as there's something going on, I'm interested.

CF) What do you think of the current retro scene, for example, new releases from Protovision? And how do you rate the quality of such software?

SB) I love new software as long as it's worthwhile. I don't want to pick on Richard Bayliss specifically, but a lot of his work seems to be the same idea again and again with very little thought put into the whole thing. Move a sprite and collect diamonds, that sort of thing. Protovision at least try to think of what to do next. Four-player PAC-MAN, for instance, if you don't like the game at least it's not just another basic PAC-MAN clone.

But then I suppose something is worth doing as long as you get the important game mechanic right, which happens more often than not with regard to homebrew. Take Pinball Dreams C64, this wouldn't have even been possible 15 years ago, or wouldn't have been attempted at least. But when you look at the demo you can see it's going to be good. So quality is varied, but that's always been the case for as long as the software industry has been in existence.

CF) Protovisions games are of a very high standard. Do you think they have a good market or a niche, are there room for other software producers?

SB) It's a niche, but I want to help to spread the word and ultimately grow that niche, which will in turn bring about more new software, so everyone's a winner. As for room for others, of course there is. Cronosoft sell Commodore software too.

CF) What other current software creators do you rate for Commodore? Maybe our reader has missed some products.

SB) I like Jason Kelk's work. ViColumn was excellent for the VIC-20, and he seems to get it right more often than not even if they're not quite as original as say a Cauldwell production. To be fair, I can't think of many games that are as original as something that JC has produced for the Speccy with its non-existent support hardware. VIC-20 fans should also check out Astro Nell from Cronosoft, which is amazing.

There's also CovertBitOps Metal Warrior games – every C64 fan should play these, and it feels like everyone is awaiting Pinball Dreams C64. Seriously,

look around and get downloading. If you download something that you don't like then you haven't lost anything. It's a bit different if you plan to buy, which is why reviews are so important. So tell me if you disagree with any review that I write please, either by emailing in or using the Micro Mart forums.

CF) What is your favourite game?

SB) Okay, my five favourites for 8-bits would read something like Turrigan (C64), GameX – The Games Exchange (ZX Spectrum), Astro Nell (VIC-20), Stunt Car Racer (C64) and Delta (C64).

CF) You were a staff writer for Commodore Scene, are you sad now the magazine closed?

SB) Staff writer is a bit rich. I was a semi-regular contributor, yes. It was a dark day when Allan closed Commodore Scene, but who could blame him when no one wanted to support it? Not I.

CF) Allan said he emailed over 300 users but only 12 took subscription would you like to comment?

SB) In advertising terms, that's a good return, but unfortunately not good enough for Allan to continue, which is sad.

CF) I know you have been pushing Commodore Free magazine, and I suspect this is why I have had so many letters and readers. How do you feel the magazine has progressed? And how could I improve the magazine?

SB) Commodore Free is progressing really well. It lacks the polish of a professional production, which is what you would expect to be honest, but it has bags of enthusiasm and interesting features. As for improving, well just let things flow and if in four or five issues time you feel you've made little or no progress, then we can talk.

CF) Do you read Commodore Free magazine? I know you are pressed for time with other projects.

SB) At the moment, I work seven days a week, but I do read it. Not cover to cover, I'd have to admit. Time won't allow unfortunately. I like what I see though.

CF) Are you still active as a coder and musician? What was the last thing you worked on?

SB) I can code a little, and I would like to finally write something decent. But I need first my own place and I think things could happen from there. As for my music, I haven't picked up a guitar or bass for a good while. Again, something I hope to put right soon.

CF) Can you tell our reader about some of your coding and music work?

SB) I have a few scraps of code knocking about. I once wrote a noter for my own purposes but I've never really done anything good. Real life takes over unfortunately. As for my music, I did compose some SID tunes a long time ago – 14 years to be exact. I don't know if they were any good, but I know that they have been lost forever at least in their binary form. I can remember the composition though. I don't tend to forget a piece of music that I have written.

CF) Do have a handle our reader would recognise?

SB) No.

CF) Why do people use a Handle and not just your real name, what is the point?

SB) Something to do with the heritage of the scene. I don't bother because I've not actually done anything worthwhile but write for some magazines.

CF) My wife thinks your handle should be "Fuzzy Bear" as you remind her of the Muppet character because you had a stubby beard and long curly hair. I think it was meant in good jest; maybe you could use this for your Handle?

SB) Interesting, I'm currently cleanly shaven with straight hair. Perhaps people will start calling me that now.

CF) Back to Retro gaming, Retro Gamer magazine seems to cover platforms and games that I would consider none retro for example Tomb Raider. I know getting revenue is important but the magazine seems to spend a lot of pages covering remakes and "new games". I don't mind new games for retro machines but covering PlayStation 2 and Xbox games are these really retro - would you like to comment?

SB) The best period of RG in terms of balance was the "Vol 2" era between issues 12 and 18 if I remember correctly. The current magazine is doing a lot right, but also missing something. It seems to react rather than to pre-empt and there doesn't seem to be a grand plan for the magazine. Not that there ever was, but when Martyn was in the editors chair, he would edit stuff to death so that the magazine had a certain flow to it, which is perhaps what is missing.

CF) I still wish they would change the magazine to retro computing and cover retro machines and new hardware and operating systems. What do you think?

SB) There should be space for everything in the magazine at some point. Retro is in itself a niche market, so why ignore niche parts of that niche market? That to me doesn't make sense.

CF) You were trying to publish a retro magazine, but the backing was pulled. Can you tell our readers about the project? Also there was a sample issue 1 that was a really good read. Is this still available and will there be others?

SB) The project was called Retro:Bytes. I was trying to achieve a publication that had well balanced content, not necessarily to please everyone all of the time as this is simply impossible, but to cover as many niches as possible, including demo coding and remakes, even though I'm not a big fan of either. The idea was to theme each issue around a few main subjects and to have regular news and such like. The sample issue should still be available from Click Gamer, but was produced simply as a stop-gap. As for the future of the project, I'm unable to comment at the moment.

CF) If you were given 1 million pounds what would you do with the money?

SB) Clear debts and look to do some sort of project like the Game Over(view) Freestyle Gaming Jam, as in offer money for the best game(s) over different formats. For me, home-brew gaming is the future at least for people who choose to use their old computers rather than just sell/collect/trade them.

CF) Commodore Scene's Doom challenge has now ended because of lack of interest from developers, but do you still feel it is a valid project do you think we will ever see Doom running on the Commodore 64?

SB) Every project or challenge is valid. Imagine if Dave Macleod had not bothered with his little E-11 project, for instance. The world of climbing would be a worse place for it. But he did it in the end, and now the most difficult traditional rock climb in the world is officially in Scotland thanks to his magnanimous efforts. So, Doom is impossible to do on a C64 with a SuperCPU; is there definitive proof for this?

CF) You were promoting a SCPU coding competition what happened has this ended? Some users may need reminding what SCPU is could you enlighten them?

SB) The coding competition was aimed at simply promoting development for a piece of Commodore 64 and 128 hardware called the SuperCPU. This adds a 20Mhz 65c816s processor to the machine and gives any coder a little bit more processor time to play around with. You can also add 16Mb's of RAM, and there's a whole new set of op-codes to learn should you want to. The competition was halted due to the changes in my life, but it might be reopened when everything is finally sorted.

CF) Do you have any final things to say?

SB) Yes, if you own a piece of hardware then use it. If you like playing retro games on your Commodore, why not try some of the games available from Protovision or Cronosoft? You might be surprised. And don't moan when things disappear. People moan that they can't buy CMD hardware anymore.

Fair enough if you've just returned to Commodore and have just found out about the hardware, but if you were using your Commodore when CMD were still operating, why didn't you buy the hardware then? You don't know what you've got until it's gone, never a truer saying that. Ranting aside for a minute, I'd like to say thank-you all for putting up with my ramblings here, and thank-you to those of you who have read and enjoyed my work these past years. I hope to continue for a little while yet.

CF) Shaun thank you, readers I suggest you email editor@micromart.co.uk and ask why is there only 1 page dedicated to retro computing. You may be interested to learn I wrote an article for Micromart magazine, Shaun decided the article was good enough to publish and so I was famed in print in a real magazine. If you would like to read the article it can be viewed from here

<http://www.micromart.co.uk/default.aspx?contentid=0a437af8-b1f5-4ffc-a6c8-f567007946de&featureid=1421>

I evaluated the current state of computer games with the title

"Nigel Parker evaluates retro gaming for the Commodore 64 and ponders the future of the technology."

Then it all went downhill and I started producing Commodore Free magazine

CSS 64 F.A.Q

A Commodore 64 Emulator

By Per Håkan Sundell

Question: Where do I connect my real 1541 drive for use with CCS64?

You don't. CCS64 doesn't support connection of any original hardware compatible with C64. Instead the 1541 disk drive is also emulated in software, and behaves as alike the real thing as it is possible. The original 5 1/4" disks is replaces by ordinary PC files, disk images.

To transfer your disks to disk images use one of the existing programs available, together with an special cable for the connection. The best program for PC is called "Star Commander", and is available at the Internet.

Question: When will CCS64 support connection to a real 1541 drive?

Probably never as there exists very good tools for transferring the data on 1541 disks to system independent file formats (like .D64). As I see it the main benefit of using a real 1541 disk drive connected to the emulator would be to use copy-protected software that uses fast-loaders and/or special formatting that the transferring tools can not handle.

But as I am pretty shure that it is theoretically impossible to get the connection 100% compatible so that it will support all kind of fast-loaders, I think it is not worth to implement a half support. The support that should be possible is the 1541 normal loading protocol and perhaps some special selected fast-loaders.

To understand the problem you have to consider that the 1541 is not just an ordinary device with a specified communication protocol. In fact it is a complete computer of its own and can therefore be re-programmed to send signals in any way and in any speed wanted, and mostly with very little or none handshaking control (controlling that the data actually has been recognized by the C64).

A fast-loader normally works in that way that it first sends over a special program to the 1541 and then starts the execution of this program (inside the 1541!). This program has full control of all the hardware that is inside the 1541 and executes on an CPU clocked on 1.0 MHz. On the C64 is also a program executing, that is running fully in parallel with the other program and is synchronized down to the level of some CPU cycles (appr. 1/1000000 seconds) .

So the problem is mainly because of the synchronization problem of down to 1/1000000 seconds. That kind of synchronization is not supported by any operating system on PC, and neither does the PC's hardware allow that accuracy - for instance the SoundBlaster hardware interrupts the main processing about 50 times per seconds for quite long period when it is moving data from the RAM and reinitializes the looping.

To be able to emulate the connection fully you must have a real-time OS and a real-time computer

system, where you can be guaranteed not get your program execution to be interrupted by more than 1/1000000 seconds. And in a normal PC, your programs are normally interrupted by a lot of things that takes a lot more time than that, like harddisk access, keyboard, mouse, scsi, eide, usb, networks etceteras. For example in Windows 3.1 you can be guaranteed a response time of 1/18 seconds - and in that time the program that is inside the 1541 has crashed for a long time ago, blink blink!

Question: Why can't I use my original 1541 disks in my PC 5 1/4" diskdrive (B:)?

It is technically impossible to read a 1541 disk in a PC diskdrive. The reason is that PC and 1541 uses completely different systems for magnetic recording, MFM versus GCR. Note also that the original 1541 is much more than just a disk drive, in fact it is a complete computer, with its own CPU and RAM/ROM. The PC's diskdrive is just mechanics with simple read/write electronics, with no intelligence of its own.

Question: Why doesn't CCS64 work on my Windows NT system?

The reason is that Windows NT is a system that does not support programs that accesses the PC's hardware directly, probably to ensure system stability. Because when programs deal with the hardware directly, any program fault can mess up the system completely. CCS64 V1.09 is developed for DOS using a 32-bit Dos-Extender. Windows NT 4.0 can only tolerate 16-bit DOS programs.

Because this problem isn't only arising with CCS64, but also with many commercial games, Microsoft have developed the DirectX interface to Windows 32-bit. This enables program to get nearly direct access to the hardware under controlled manners, thus keeping system stability.

CCS64 V2.0 therefore is distributed as two versions for PC, one DOS and one Win32/DirectX version. Win32 is a Microsoft name-convention for Windows95/98/NT which all is built around a similar 32-bit architecture. To have DirectX on WindowsNT 4.0 you must have installed Service Pack 3 or later. To have DirectX for Windows95 you must install a special package from Microsoft. In Windows98 there is an pre-installed version of DirectX 5.0.

Question: Why don't I get any sound with CCS64?

If you are using the DOS version, you should ensure that you have properly installed your soundcard for use with DOS, and that you are using either a SoundBlaster compatible or Gravis Ultrasound soundcard. With your soundcard package there should be a 1.44" disk containing DOS drivers.

When this is properly installed, the systems boot files AUTOEXEC.BAT and CONFIG.SYS should be changed to contain an environment variable setting. Often this will look like SET BLASTER=A220 I5 D1 H5 T4 etceteras. When CCS64 starts it will look after this environment variable setting and configure its sounddrivers according to that. For more information look in your soundcards manual at the topic "How to use my soundcard with DOS games" or visit your soundcards vendors homepage on the Internet.

If you are using the DirectX version, you should ensure that you have the latest drivers for your soundcard installed, and that they are compatible with DirectX. Also ensure that you have the latest version of DirectX installed. Your soundcard may also have different modes of DirectSound interface, of which only some works with CCS64. Enter the Sound-menu in CCS64 and try to change the Sound Device settings, hopefully some other setting will work.

Question: Why doesn't my joystick work with CCS64?

For use with the DOS version, you must have a DOS compatible joystick. Many new advanced joysticks are only designed for use with Windows and DirectX with special drivers. In this case one possibility is to use the Win32/Direct version of CCS64. The DOS version can sadly not use any features of Windows even if it is running under Windows.

Also ensure that you have calibrated the joysticks correctly in the emulator. In CCS64 V2.0 there is a special sub-menu for this purpose.

Question: Why don't I get any graphics display with CCS64 for Win32/DirectX (DirectDraw-Init Error)?

CCS64 for DirectX uses a double-buffering system for the video display, to minimize possible flicker. This can consume quite a lot of video memory, perhaps your video-card has too little. Different video-modes requires different amount of videomemory, so try with different settings. Also ensure that you have the latest DirectX version and the latest version of drivers for your video-card.

It seems also that it matters if the MS-DOS prompt window is in windowed or full-screen mode, DirectX sometimes fails to initialize properly when there is a full-screen MS-DOS window. So try to change the settings to windowed mode for the MS-DOS prompt (Command prompt in NT).

Question: Why doesn't games (scrollers in particular) run smoothly on my computer - even if I have selected the 384x282 screenmode and have a powerful CPU?

The reason is that the PAL C64 is making frames with 50.12 Hz (985248/63/312) and the screenmode you are using is probably not exactly 50.12 Hz but rather perhaps 50.00 Hz. That means that after about 8 seconds (1/0.12) it will be produced one more frame than have been showed, and that frame will never be displayed. And because of the missed frame the scroller will make a visible jump.

If your displays refresh rate is faster than 50.12 Hz, perhaps 70.00 Hz, this means that several frames will be displayed twice but others are displayed only once.

Question: Why don't I get any flashing Scroll Lock LED with CCS64 for Win32/DirectX when the 1541 drive is working?

The reason is that current version of DirectX or Windows doesn't support changing of the keyboard leds from software. According to Microsoft rumours, there will be support for this in future version of DirectX together with Windows98/NT5.0.

Question: What should I do when the game asks me to "Insert Disk 2"?

Well, you should insert disk 2 into your virtual 1541 diskdrive. When you started the game you probably "inserted" disk number 1, so the answer lies in the same menu, where it is probably called "select". In version 1.09 the disk image (probably with .D64 extension) is selected as soon as you look at its contents.

Sometimes you could have this message repeating - even if you have selected the correct disk image in CCS64. This is most probably because the normal kind of format for disk images, .D64, doesn't contain all information that is on a real 1541 disk. There are quite many programs for C64 that check for this information to detect whether the correct disk is inserted or not.

In this case search the Internet for a better disk image, that either contains the missing information, or where the program check functions are fixed.

It could also be the case that you have too little knowledge of the C64 and its equipment to do a correct decision. In this case there are a lot of literature available at the Internet, search for "Project 64".

Question: I have successfully started a game, but all I get is a blank or static screen?

As the virtual 1541 disk drive is emulated to be as alike as a real 1541 disk, it is also as slow or fast as the real thing. On many games it takes several minutes on a real C64 with a 1541 diskdrive before the game is fully loaded.

In version 2.0 of CCS64, there are several possibilities to speed up the loading procedures. One is to use a fastload ROM-set or cartridge, as you would on a real C64. There is also an emulator fastload functionality that will trap normal loading demands from games which uses the C64 Kernal \$FFD5 vector, and then will load the file into virtual RAM in no-time.

Question: I have successfully installed CCS64, but how do I use a C64?

You should really start reading the user manuals and other useful papers for C64. There are a lot of literature available at the Internet, search for "Project 64".

Question: Where do I get game ROMs for use with CCS64?

You don't. ROM stands for Read Only Memory and is normally the basic part of a computers operating system (BIOS). The ROM is an integrated electronic circuit and is located on the computers motherboard, and on expansion cartridges. Games for C64 were most often distributed as 5 1/4" disks or on tapes, as most C64 systems were connected to a 1541 diskdrive or a C2N cassette recorder.

In CCS64 are instead disk images and tape images used, as the original media can not be used directly. They are available at the Internet in very large quantities, a good starter is www.c64.com.

You should really start reading the user manuals and other useful papers for C64. There are a lot of literature available at the Internet, search for "Project 64".

Question: Why doesn't this particular game work on CCS64?

Even though I aim to get CCS64 to be perfect, there may be some programs that doesn't work correctly. The first thing you should try is to see if you could change some of the parameters like disabling the REU or 1541, different reset memory patterns, system modes etceteras. If you have tried all possible combinations and it still doesn't work, then perhaps try with some other emulator, and most importantly - if possible - try the program with a real C64. There are actually a lot of damaged disk and tape images spreading around, that were damaged during the conversion from the c64 media or perhaps lost some information that were on the original C64 media (like copy protection).

Question: When will you implement this particular feature and release the next version of CCS64?

I actually do not know, but you can count on that I will do it as fast as possible. This is a one-man project, I am the single author of CCS64 and I try to program everything myself from bottom and up. The project is done solely on my free-time as I have an ordinary work on day-time.

Interview with Håkan Sundell

Computerbrains C64

<http://www.computerbrains.com/>

Q - Please introduce yourself to our reader

Born 1968, family father, both professional and academic carrier (PhD), currently works as a senior lecturer (associate professor) in programming. Have been programming since 1982, basically self-learned, started with assembler.

Q - Can you tell our reader how you were first introduced to Commodore

In the beginning of the 80's I saw the PET, which unfortunately was by far too expensive. In 1982, we borrowed a Vic20 over the summer, which I was really fascinated with and used 24/7.

Q - Do you own any Real Commodore hardware and do you still actively use it, and what for

I own a lot, like a Vic20, several C64's and several Amiga's, together with a lot of related stuff. Basically I keep them for reference, as well as for nostalgia purposes.

Q - i am still trying to find a company that operates solely by using Commodore machines, do you think I will ever find one

No, that wouldn't be very likely. However, there could probably still be some C64 out there still controlling some connected machine or equipment.

Q - Please Tell our reader about Computerbrains, who are you all and is the CCS64 emulator the only project you focus on

Computerbrains was basically about the scene in the past. Nowadays, CCS64 is the only project in focus.

Q - when was the software first developed

In 1995.

Q - The CCS64 software, some readers may not be familiar with it can you explain what it does

It makes it possible to run old C64 software on modern computers, like a modern PC.

Q - Was CCS64 the first software emulation of the Commodore 64,

No. I think one of the first came for the Amiga, like A64. The PlaySID project from 1989, which I developed together with Ron Birk, also kind of emulated the C64, although in a more limited way.

Q - The logo with CCS64 and a commodore 64 over what looks like the world, what does this mean, Commodore 64 world domination?

It is just a logo, no special thoughts involved. Maybe like c64 forever.

Q - Have you tried other Commodore emulation software?

Yes, probably most of them.

Q - What platforms does the software run on, and do you plan to port the system to other operating systems

Currently only Windows is supported. It actually depends of what system I am using, if I will migrate to another platform then CCS64 will as well.

Q - Can you explain why our reader should us CCS64 instead of other emulation software?

Nowadays it is basically a matter of taste, maybe related to the user interface and some special and unique features that are of interest. In the past it was also much about compatibility, which the others now have now caught up..

Q - Following on can you give our reader a list of the software's strengths and weaknesses?

There are a lot of comparisons already on the net, or try it yourself.

Q - The software is free to download, although some of the features are locked, you ask readers to register, what benefits would registration give our reader

Currently there is nothing locked. The benefits is basically more influence and increased speed of further development.

Q - How much is the registration fee

30 USD.

Q - What do you spend the registration fees on

Running the site and computer equipment.

Q - Is the CCS64 software still being developed or do you think its now 100% complete

There is always more to do.

Q - Can software emulation ever be 100% finished?

No, there is always something more to do, like nice extra features etc.

Q- Have you found anything surprising out about the C64 while making the software, things like undocumented bugs or intra machine incompatibilities, like problems with the various c64 models and revisions

Yes, a lot. And still do.

Q - I read someone saying there is a plan for CSS128, can you comment about this

Who knows...

Q - Apart from running Commodore 64 games and applications what practical use is a Commodore emulator

Learning to program.

Q - Are there plans for other emulation for example the Commodore plus 4 and 16 machines

Who knows...

Q - The software uses Direct x on the pc for windows what advantages does this give you, and does this cause problems for the DOS version that doesn't have direct X application layer

The only advantage with DirectX is that I don't have to write drivers for particular graphic/sound cards etc. DirectX is also faster than the old Windows Win32 API for drawing. With the DOS version I had to write for example special code for Soundblaster and Ultrasound with various revisions etc.

Q - If you had someone to fund a project and they had unlimited amounts of money to invest, and also a

machine to freeze time so you had unlimited time on a project what would you create and why

I would try to realize the HAL 9000 computer that was mentioned in 2001 by Arthur C Clarke. It is the ultimate goal to make the computers think like us or even better...

Q- If you won 1million pounds what would you spend the money on

I would buy a house so that I and my family could move from our small rented flat. Probably also a nice boat. And lastly (of course) also a lot of nice computer equipment.

Q - Finally I use winvice on the PC, what would you say to convince me to use CCS64 instead

Try it yourself and discuss with other CCS64 users.

Thank you for your time and software

> No problems.
>
> Regards,
> Håkan Sundell



THE END

Commodore Free Magazine

Commodore Free magazine is a FREE to download magazine for Commodore computers available in PDF TXT and as a Commodore 64 D64 image file

www.commodorefree.com