

# COMMODORE FREE

A free to download Magazine dedicated to Commodore computers.

## Issue 82

Commodore 65 in FPGA Interview with Programmer

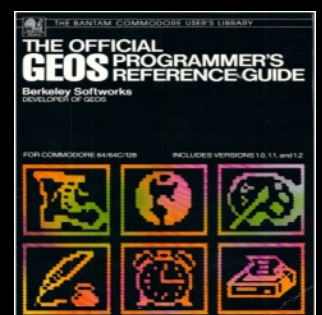


Interview with Melon 64 sysadmin

VIC games

SD card Review

Geos programming



# Contents

Editorial and Contents	Page 2
Commodore Free Cover Tape	Page 5
NEWS General	Page 8
NEWS C64	Page 9
NEWS Amiga	Page 15
NEWS Vic & plus 4	Page 21
Geos And Wheels Programming Basics By Terry Raymond	Page 23
Commodore 1 Church 0 By Lenard R.Roach	Page 25
Making a C64/C65 Comparable Computer FPGA Interview with Paul Gardner Stephen	Page 27
Commodore Free Interview Melon64 Admin	Page 33
Review Of Sd2Iec Device From Future Was 8 Bit	Page 35
The Assembler line \$05	Page 37

## Editor

Nigel Parker

## Spell Checking

Peter Badrick & Bert Novilla

## TXT, HTML & eBooks

Paul Davis

## D64 Disk Image

Al Jackson

## PDF Design

Nigel Parker

## Contributors

Richard Bayliss /Terry Raymond  
Lenard R. Roach /Bert Novilla (satpro)

## Website

[www.commodorefree.com](http://www.commodorefree.com)

## Email Address

[commodorefree@commodorefree.com](mailto:commodorefree@commodorefree.com)

## Submissions

Articles are always wanted for the magazine. Contact us for details. We can't pay you for your efforts but you are safe in the knowledge that you have passed on details that will interest other Commodore enthusiasts.

## Notices

All materials in this magazine are the property of Commodore Free unless otherwise stated. All copyrights, trademarks, trade names, Internet domain names or other similar rights are acknowledged. No part of this magazine may be reproduced without permission.

The appearance of an advert in the magazine does not necessarily mean that the goods/services advertised are associated with or endorsed by Commodore Free Magazine.

## Copyright

Copyright © 2014 Commodore Free Magazine  
All Rights Reserved.

# Editorial

Greetings!

Another mixed bag of articles again!

This time, we have Leonard Roach pondering whom he should continue writing for. In the past he has written articles for Commodore Free and even sold Commodore programmes; you may remember he has released some Commodore-related books. It's an interesting read although I know it won't be an article that suits everyone's taste as it touches on the subject of Faith.

Also, this issue we have an interview with the sysAdmin of Melon64, a forum for Commodore users. Some people I know seem to think that it's in direct competition to Lemon64, but read the interview and judge for yourself. It's not an exclusive interview and don't expect all to be revealed but it makes some interesting reading all the same.

Ever thought of making a shoe phone or an implementation of a Commodore 64/65 in FPGA logic with some extensions? Well one reader has, and he talks about his plans for such a project. I mean just think if you could own a portable Commodore 64, even if it was just a piece of FPGA logic with screen and keyboard? How cool would it be – working on Commodore Free while others looked over my shoulder wondering what operating system it was, and wowing at the fact it booted up in milliseconds rather than 15 minutes?

The more I read about the C65 the more I want one – not really one of the prototypes – but a fully working version. How I wished Commodore had released the machine to market, although they really needed to get it out faster, and I think that's one of the reasons they closed the project, because 16-bit machines were overtaking the 8-bit systems. Think of how the games would look and the extra sid for stereo soundtracks! Ok, I know it's never going to be a reality, but with an FPGA logic version, who knows? Anything could be possible!

We have a review of an SDuiec device, that lets you load and save programmes onto SD cards, although there are any number of similar versions I picked this mainly for the reason it looks like a real 1541 drive, it's also very neat and well manufactured. Concluding our magazine are two machine code programming tutorials, we continue our regular Machine code lessons and also we delve into the elite world of the Geos programmer with a look at Geos header files, and what tools you would need to code your own applications.

Thanks for reading

Best regards

Nigel (Editor)

[www.commodorefree.com](http://www.commodorefree.com)

Your source of Commodore news

# Commodore is Awesome

A place for everything Commodore

Magazines Scans  
Amiga Format Games  
Amiga Bladet Demos  
Zzap64 Applications



**CiA** 

<http://awesome.commodore.me>



# Commodore C64 Back to the Future!

Connect to TV and Play!

- 2 MByte FLASH
- 16 MByte SDRAM
- S-Video out
- Stereo Audio out
- PS-2 keyboard and mouse
- Micro SD card
- 2 Joystick ports



more than 100  
C64 games!



Online Order: [www.arcaderetrogaming.com](http://www.arcaderetrogaming.com)

More Information: [www.mcc-home.com](http://www.mcc-home.com)



# COMMODORE FREE E-COVER TAPE 15

by Richard Bayliss

Loader system by Martin Piper  
and Daniel Kahlin

all Loading tunes by Richard Bayliss



It's summer time once again and Commodore Free springs back with a HOT sizzling cover tape, crammed with spectacular goodies for you to enjoy. The past few E-Cover Tapes had mainly SEUCK compo entries. This issue also has a couple of SEUCK games which are EXCLUSIVE and so have not seen before. The majority of this tape consists of different programs. Huge thanks goes to CSIXX for giving us kind permission to use the C64 conversion of The Impossible Game on to this month's cover tape, which I have mastered using Daniel Kahlin's R-Loader.

You may have noticed that I reverted back to the regular TND loader system (Thunderload Series Six) for the majority of programs on this E-Cover Tape. I did do a brand new tape loading system, but a 3 colour logo which says "Thunderload 7" needs to be painted. So guess what else I have added on to this issue's cover tape? ... A logo editor ;)

Anyway, what is in store for you on the cover tape this month, before we take a break until after summer? First of all there are a couple of SEUCK games, which are fun and full of humour. There is also a 2 player car racing game, and of course (as pointed before) a Commodore 64 conversion of the flash PC game, "The Impossible Game". There's also 3 utilities (or should we count it as 4?) for your C64.

## RABID ROBOTS

(C)2014 by Alf Yngve, Music by Richard Bayliss

This is a funny game which was created by Alf Yngve, using the Sideways Scrolling SEUCK. This game consists of a comedy look into UK politics and the UK as well. This game was written by Alf only for fun, and wasn't intended to cause any insults to anyone. We made sure of that.

England is under attack by a series of robots from U.K.R.A.P (UK Robots Against People). They are spreading a message of 'hate' to the nation. All people have been held as prisoners in their own home. Michael Garage (who is also a robot) has built a robot empire to guard the whole country. His mission - to spread chaos across the country - and put an end to standard lifestyles.

Plucky schoolboy Vikram had a phone call from his sister Nita. She discovered U.K.R.A.P HQ as a school project, but she's lost in the park. Vikram decides to look for her, so he gets a water pistol and his pet bull dog Maztif to help locate her.

Help control Vikram through the city of London battling against the robots. Robots are spreading 'hate rays' around the city. Luckily for Vikram he can destroy the deadly robots by blasting water or throwing water balloons at them. Vikram's dog can also destroy the robots by confusing them with its barking.

During your travel across England you will come across a series of robots from other rogue robot parties. There are some other handy items which can be picked up to help you. Cans of fizzy pop will give you extra energy. Pick up a flame thrower. The dog can also help you pick flame throwers up. Try not to scare the cat, otherwise you'll be penalised by a reduction in energy.

Vikram cannot walk in water or across holes as he is pretty much accident prone. Luckily for you his dog can walk through both areas. Can you help Vikram save England from a robot invasion, defeat Michael Garage, and meet his sister? Good luck.



### Zappy Bird (C)2014 by Anthony Burns, Music by Richard Bayliss

The cult game meets its match with a sideways SEUCK parody of the game, called "Zappy Bird", written by Anthony Burns.

Infuriated at Flappy Bird's popularity various retro game characters summon the aid of an evil witch and her cohorts invade his game world and devastate it. Distraught, poor Flappy calls upon his psychopathic cousin "Zappy" to give the intruders a taste of their own medicine ...

Using a joystick in either port, help Zappy fly across poor Flappy's world fighting back against the evil retro characters and the evil witches. Watch out for their deadly bullets. Col-

lect black crosses to gain extra points with a possibility of an extra life for every 10,000 points scored.



### Dream Cars (C)2001 by Lubber/Padua

This is a public domain racing game for 2 players only. It was written back in 2001 for the Mekka Symposium party. Both players must enter their names. Select a car and choose the total number of laps to win the game.

It is now time to race your cars around the course. The first person who makes it around the course at the quickest time will become the winner. There are 4 different courses to race on. The player with the most wins through the 4 courses will become the overall winner. Beware, course 4 is the hardest course to get around. You'll see why when you play the game.



### The Impossible Game (C)2014 by CSixx, Music by Spider Jerusalem,

From one cult game - "Flappy Bird" - to another, the PC flash cult game called "The Impossible Game." This is a stunning Commodore 64 remake (or should we have called it a De-make) of the classic flash game. This game was going to be added on to last month's E-Tape but I was still waiting for permission from the author behind the game. Now with a huge thank you to C-Sixx, it is here.

Using just the fire button on your joystick in port 2, help your square jump across the moving maze and jump over any infinite obstacles on the ground as they scroll towards you. You can also jump on to other objects which are raised as platforms. Keep on going until you lose. :) Try to get the best score where possible :)



**CENTAURI LOGO EDITOR V1.2+**  
**(C)1989 by Henry/Centauri and Stad/Stash, Music by Drax**

This is a nice utility which allows you to design and create your very own 3 colour logos for your own C64 programs. This editor has a series of different options in which you can select to draw, fill copy and also enjoy some music by Drax/Vibrants in the background. The logo editor also allows you to convert logos painted from Koala paint into char and matrix form - and char and matrix to hires/multi.



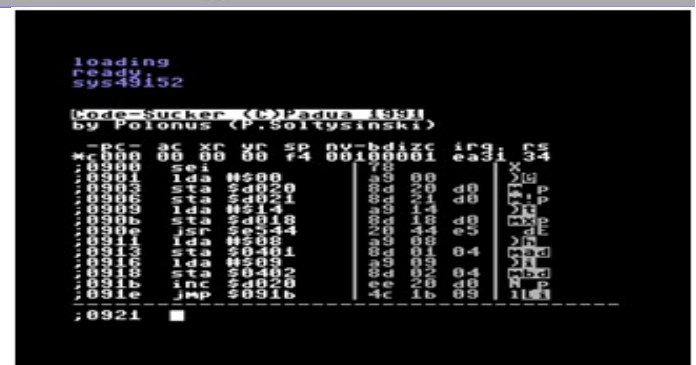
**SPRITE MACHINE**  
**(C)1993 by Iceball/Vision, Music by Richard Rinn**

Here's a handy utility that will allow you to draw, create and animate your own sprites for your very own programs. This editor consists of varied options which can be selected to help copy/paste and do other things to your sprites. The editor also consists of nice relaxing background music, while you are editing your own sprites



**CODE SUCKER M/C Monitor**  
**(C)1991 by Polonus/Padua**

This is a Machine Code monitor -- for anyone who wants to write their own machine code programs the difficult and very old school kind of way! The program consists of what you would expect in any machine code monitor. Load the installation menu from the tape. Select the memory location where you would like to save the M/C monitor to disk, then start programming. Instructions are also provided on the installation program.



**ON THE NEXT COVER TAPE**

We won't be compiling another cover tape for 2 or 3 months or so, due to the summer period. Just like last year we are taking a well-earned break until September 2014. However, on the next cover tape, we hope to have the usual good balance of programs. So if you have anything you would like to submit (even if it has already been released) then please email it to the usual Commodore Free address or to richardbayliss.c64(a)gmail.com.

# News General

## World of Commodore 2013 Videos

Leif Bloomquist announced that videos from the World of Commodore 2013 have been released for viewing on the internet. To see the videos head to the Youtube channel

### YouTube Channel.

<https://www.youtube.com/user/TorontoPETUsersGroup>

- MIDI for FastFingers (Dan Laskowski)

- Musical Experiments on the Commodore PET (Chiron Bramberger)
- Pixcen C64 Pixel Editor for Windows (John Hammarberg, Censor Design)
- M.U.L.E. Returns for iOS (Matt Brudzynski, Comma8 Studios)
- Motion-Sensing Gaming Glove (Leif Bloomquist)



## Programmable Logic I PLA & PAL

Bil Herd has created an article on hack a day about Programmable Logic. Bil explains how the PLA for the C64 was developed. The PLA in the C64 was used to create chip select signals from various other signals, The signals control which chip is connected to the data bus and is therefore responsible for the memory mapping of the C64.

Read more and watch the video here

<http://hackaday.com/2014/06/24/programmable-logic-i-plapal/>



## World of Jani WEBSITE

On this web page you can find hardware projects, documentation, preservation, and repairs, amongst many other Commodore related items. All information is provided "as is" says the author. Some interesting project to take a look at.

<http://blog.worldofjani.com/>



## Finnish Retro Game Comparison Blog

Not only does this website have a supper snappy name, it also writes about retro games. The site (blog) compares games between different computers like the C64, Amiga, MSX, NES, CPC, DOS

<http://frgcb.blogspot.co.uk/>



## Paula Powered - Vivienne Westwood

The French and British music combo Paula Powered have been making music with the help of a Amiga 1200. They make Low-Fi style punk using Octamed Soundstudio tracker. Check out the video on youtube

<https://www.youtube.com/watch?v=j8aTyiEX7jg>

<http://www.paulapowered.com>

<https://facebook.com/paulapowered>

<https://twitter.com/paulapowered>





# C64 News

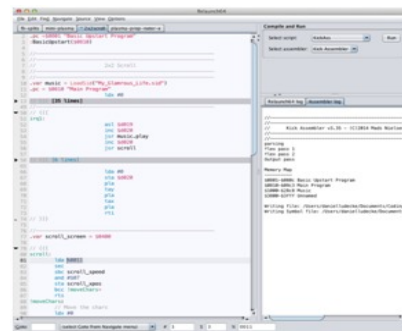
## relaunch 64 VERSION 3.0 RELEASED

*Relaunch64* is an IDE (text-editor) for C64 assembler-coding on *Windows, Linux and Mac OS X*. Relaunch64 has a clean and intuitive user interface, yet it offers many features that make coding faster and easier. This editor works together with common cross assemblers. Currently supported assemblers are:

64tass / ACME / ca65 / DreamAss /Kick Assembler

Other assemblers might work as well, but syntax highlighting may not be 100% correct

<http://www.popelgand.a.de/relaunch64.html>



## REVIVAL STUDIOS NEWS

[COMMODORE PET] - NEW GAME RELEASES: Rush and Boxing Champ Two more great-looking games released for the Commodore PET:

- Rush is a endless-runner type of game featuring parallax scrolling. You are a thief and you have to escape a heist by going from rooftop to rooftop.
- Boxing Champ is a side-view boxing game, in which you have to defeat multiple foes. Wait, find your opening and attack your opponent!

### Game features:

- Available on Tape and as Digital Download
  - Addictive and unique puzzle concept
  - Works on all PET models from 3016 and up.
  - Flickerfree gameplay
- Digital copy 3,50euro, on cassette tape 7,50

[COMING SOON]

Here's a sneak peak of what is coming up

- Finishing development on Stairrunner for the Commodore VIC-20

More formats and games have been released for other news check the website <http://www.revival-studios.com/>

## strikerterm 2014RELEASED

The notes say this is a terminal software package for the Commodore 64.

Striketerm 2014 is a streamlined hack of Novaterm for the modern day C-64 It is compatible with Nova9.6 modules

<http://csdb.dk/release/?id=130807>

The main website is here

<http://1200baud.wordpress.com/>

Also released is Striketerm Ultra (for a purchase price of \$20 USD) the notes say

Striketerm is NOW 9600 baud compatible (using the UP9600 driver). And as before, it's still fully user port compatible for running standard 2400 bbs systems. For dialing out on Striketerm, you can use the User Port OR UP9600 modem drivers without the need for additional hardware.

check the website for more information



## Komoda issue 9 released in polish

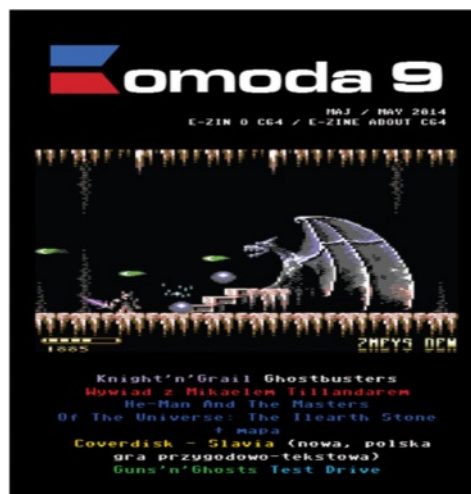
Issue 9 of the Polish commodore magazine has been released, you can download the magazine from the website. Rather sadly, I can't read it but the contents page says:

*fresh, block frenzy, scorpius, dark force, he-man and the masters of the universe: the ilearth stone, ghostbusters, test drive, knight'n'grail, mikael tillander 3, guns'n'ghosts, miszcz" i małgorzata 0*

*1985 - the day after, lode runner, harrier strike, top 10, slavia nadeszła, johnny przedstawia*

Looks rather good!

<http://dl.dropboxusercontent.com/u/16069876/index.html>



## Jonathan Temples interview

C64.com has interviewed Jonathan Temples, who was a graphic artist for Code Masters, Zeppelin Games, Choice Software and Genesis Software. He says his first computer was a Commodore VIC20 followed by a C64. Some games he worked on are: SpellCast, Spikey in Transylvania, CJ in the USA, Stuntman Seymour, DJ Puff's Volcanic Capers, Nobby the Aardvark. You can read the interview on the web page

<http://www.c64.com/>



## Chameleon 64 - 9a

A new core is available for the Chameleon  
Chameleon is an extremely user-friendly cartridge that can be used without opening the computer. It is just plugged to the expansion port of the C64. In addition to that, it can also be operated as a stand-alone unit, replacing the computer, the floppy drive and the heavy power supplies. If operated as a stand-alone unit, a USB power adapter or active USB hub can be used as power source (available separately). Even a USB-enabled computer (LCD-TVs, DVD-players...) can be used as a power source!

### featuring Improvements for:

RESTORE key,  
CIA timer B interrupt,  
Pagefox cartridge emulation,  
Disk drive emulation,  
VIC-II emulation, S  
DRAM refresh counter,  
SIDplayer and the file browser.  
a player for Compute Gazette SID player files (.mus/.str)

<http://wiki.icomp.de/wiki/Chameleon>  
[http://icomp.de/products/chameleon\\_e.htm](http://icomp.de/products/chameleon_e.htm)



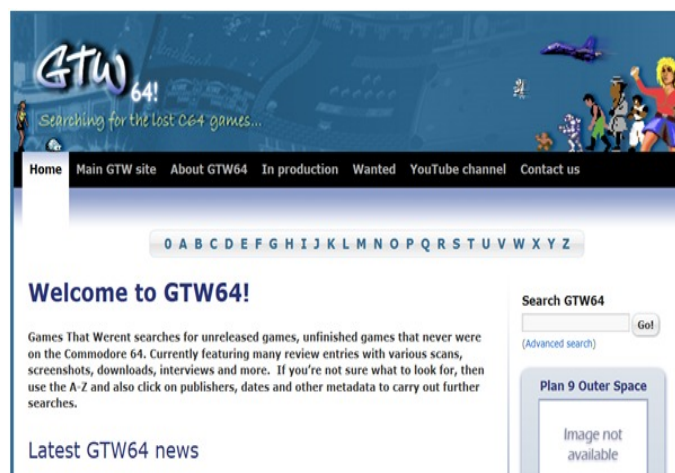
## Games That Weren't 64 UPDATES

<http://www.gamethatwerent.com/gtw64/>

The GTW64 web page has been updated

### New items include:

Pluff, Amity Island, Robotomania, Captain Future, Death Trap, It Came From The Desert and Night Walker. Update: Arcade Classics 2, Batman 3D, C64GS cartridge titles, Chuck Rock, Circus Fun!, Damocles, Darius +, Fire & Forget, Gi Hero, Heavy Barrel, Judge Death, Mach 3, Manhattan Dealers, Old Scores, Seal Cull, Search For Sharla, Sigue Sigue Sputnik, Spellcast, Streethawk, The Blues Brothers, The Seven Gates Of Jambala, Unnamed helicopter game, Victory Road (UK version), Viking Child, Virus and Whirligig.



## gateWay manual resurrected

-----Original Message-----

From: Glenn Holmer Sent: 16 June 2014 19:21

To: Commodore Free Subject: gateWay manual

I've reconstructed the gateWay manual with illustrations and the 2.5 addendum, and made it into a PDF:

<http://www.lyonlabs.org/commodore/onrequest/geos/gateWay25docs.pdf>



## C64 Network Game is Ready for Playtesting! AND Ultimate C64C Motherboard Swap

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

<http://jammingsignal.wordpress.com/2014/05/13/my-new-c64-network-game-is-ready-for-playtesting/>

Ok, it's not exactly a recent news item, but it is the first time I remember seeing something mentioned about the items in a magazine so here goes

Leif Bloomquist Released a new network game called "Vortex 2" at ECCG back in 2013. Itss a playable work-in-progress to get help with testing and feedback.

Leif says "You can't do much in the game yet. Enemy ships will fly around and chase you and each other, you can shoot at them and they will shoot back, but nothing will happen. But it's still pretty fun, especially with friends!"

### Notes:

- The game auto-updates. Every time you run it you will get the very latest sprites, graphics, music, and gameplay automatically.
- You need to forward UDP Port 3000 to the IP address given to your C64, displayed on the startup screen. The suggestion is a DHCP rule in your router to always get the same address. The MAC address used is 00:80:10:00:51:00.

- RR-Net or compatible card (64NIC+, FB-Net, etc) for your C64.
- The VICE emulator works but seems to have terrible lag.
- The server will be occasionally rebooted .
- Joystick in Port 2.

**You can download the boot program here**

<http://www.jammingsignal.com/c64/vortex/vortexboot.png>

**To see the list of players, some server stats and a real-time map here**

<http://vortex.jammingsignal.com/menu>

**Google Group for developers here:**

<https://groups.google.com/forum/#!forum/c64-vortex-development>



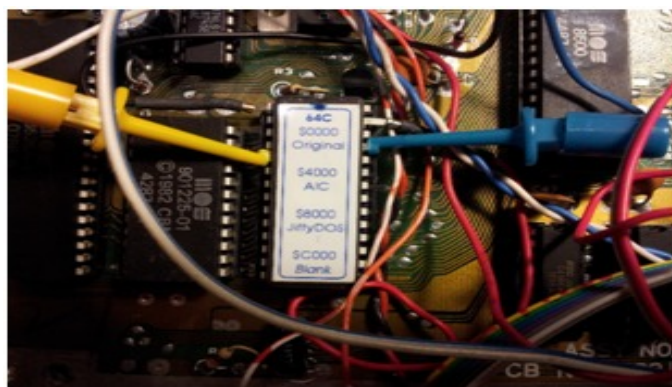
## ALSO

### It's Alive! Ultimate C64C Motherboard Swap

A couple of years ago Leif put together the "Ultimate Commodore 64" with multiple kernels, dual SID chips for stereo sound, a reset button, a USB, Ethernet, a 16GB flash drive, 4 Joystick Ports, and then painted it blue! Although... it seems Leif fried the board .

**More information is available here**

<http://jammingsignal.wordpress.com/2014/04/06/its-alive-ultimate-c64c-motherboard-swap-2/>



## GLINK 232 CARTRIDGE

Both the commodore 64 and the 128 are limited to 2400 bauds. Dr. Evil Labs and then later CMD produced the SwiftLink232 and turbo232 cartridges based on the 6551 ACIA chip as a workaround to this limitation. Sadly Both of these items have been out of production for a very long time. GLINK232 is a modern replacement for the swiftlink232 so your Commodore can communicate at speeds up to 38400 bauds

<http://gglabs.us/>



## C65 USING FPGA

Oh boy! Wouldn't the C65 have been a cool machine? Anyway, Paul Gardner-Stephen is working on an FPGA version of the C65. Features planned for the C65 - FPGA are: 1920x1200 @ 60Hz resolution, 256 color palette of 4096 colors, 8 normal or enlarged sprites, faster CPU, 16MB RAM, multiple SIDs and more.

<http://c65gs.blogspot.com.au/>



## D64 editor UPDATES

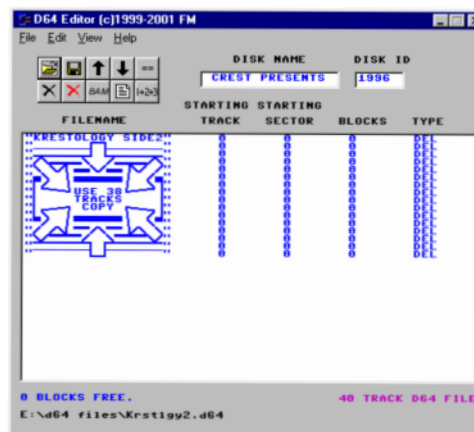
Want to edit D64 images? Well, you couldn't do any better than to grab yourself this piece of software, and even though other products are available, I still seem to grab this guy and always have it handy with my Windows toolkit.

Here is a quick rundown on some of the features:

- Read D64 files, display the directory, move files around the directory, rename files.
- Export files from the D64 into PRG files
- Import files into the D64 with autodetection for P00 files.
- View/Edit the BAM
- View the block chain of files on the D64
- Create new blank D64 files, clean existing D64's
- View and modify blocks on the disk in hex.
- Automatic check for crosslinked files on the D64.
- View the directory using the native c64 font
- SEQ File Viewer using the c64 font with PETASCII->ASCII conversion option

- Drag and Drop functionality. Drag D64's onto the main screen to load them, drag other files to import them.
- Preliminary support for importing T64's
- A filename builder to enable easy use of the C64 Characterset in filenames

<http://www.d64editor.com/>



## Sokoban FOR THE C64

Steve Ody has released the classic game Sokoban for the Commodore 64. Sokoban is a classic strategy game where the player must push boxes onto their storage locations. Once all boxes are in place, the level is complete.

Sokoban64 contains 100 puzzles divided into two difficulty levels. I know you know that you like playing this game, so stop denying yourself the fun and download a copy!

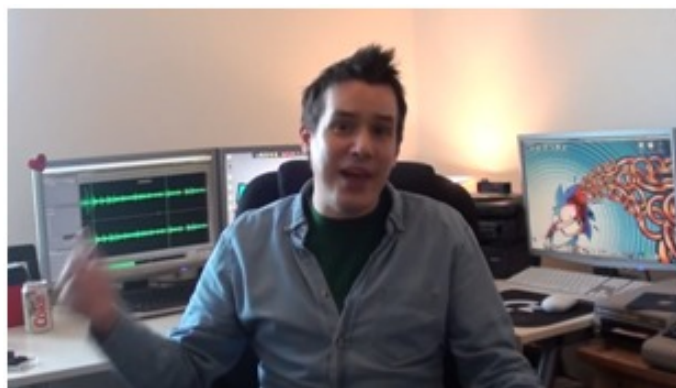
<http://steve.odyfamily.com/?p=556>



## Commodore 64 AND THE Internet

AS a follow-on from last issue's Contiki tutorial, we have Dan Wood with his YouTube video on how to connect a Commodore 64 8-bit to the Internet, using BBS, IRC And WWW and also WarpCopy. He uses the 64NIC + cartridge and the Contiki software. So even people who don't want to be bothered to learn can copy his actions.

<https://www.youtube.com/watch?v=iccQZCxDd60>



## ArnoDash Number 21 released

Arno Weber has released another new Boulder Dash game variation. This game is the 21st in the Arno Dash series; there are 16 caves and four international missions. You can download the brand new game here

<http://www.boulderdash.nl/>

## Simple Writer V1.0 released

Lux of Delta machine has released a C64 text writer. The download has the D64 and also a text file of how to use the program. In the text file is a plea for help:

Please note, we are a small group that has no C64 music coders (musicians), so we are looking for coders or groups that might give us your SID files for use in our projects with your kind permission. In return, we would like to properly credit and promote your work.

You can contact us on our email's or find us on csdb.dk (<http://csdb.dk/group/?id=7940>).

## Game Corner Q2/2014 released

A write-up disk magazine of the games released and cracked for the C64 in Q2 of 2014 has been released. You can download the file from here:

[http://csdb.dk/getinternalfile.php/130948/Game\\_Corner\\_Q2-2014\\_LAXITY.d64](http://csdb.dk/getinternalfile.php/130948/Game_Corner_Q2-2014_LAXITY.d64)

<http://csdb.dk/getinternalfile.php/130921/simple-writer-v1.0.zip>



## commando arcade RELEASED

<http://www.nostalgia.org/common/pages/releases.php?op=showrelease&id=321>

### Brilliant and finally a bugfixed release of Commando with more levels and some bug fixes and enhancements,

- \* 5 new levels added. Game now has a total of 8 levels like arcade
- \* New Barracks, Hangar, Chopper, Level transition, etc. graphics
- \* New Level features
- \* New glitch-free Sprite Multiplexer
- \* New Scroller
- \* Chopper that drops you onto the battle field and picks you up
- \* Level Transition screen animations
- \* Score, Lives and Bomb counter fixes
- \* Code bug fixes - no more crashes
- \* High Scores in attract mode
- \* Game Over sequence added
- \* Two new Sound modes with new SFX and new tunes by 6R6
- \* Loading screen (by STE'86) and Loading tune (by 6R6) added
- \* Pause added with Quit option
- \* Easyflash version with High Score saver to flash and back-up to disk
- \* Universal File version that should work on any device
- \* IFFL version with REU support
- \* PAL and NTSC supported

Trainers : +5

+D (Documentation/Manual)

+F (PAL/NTSC Fixing)

+H (High Score Saving)

+I (IDE64 Fix)

+R (REU Support)

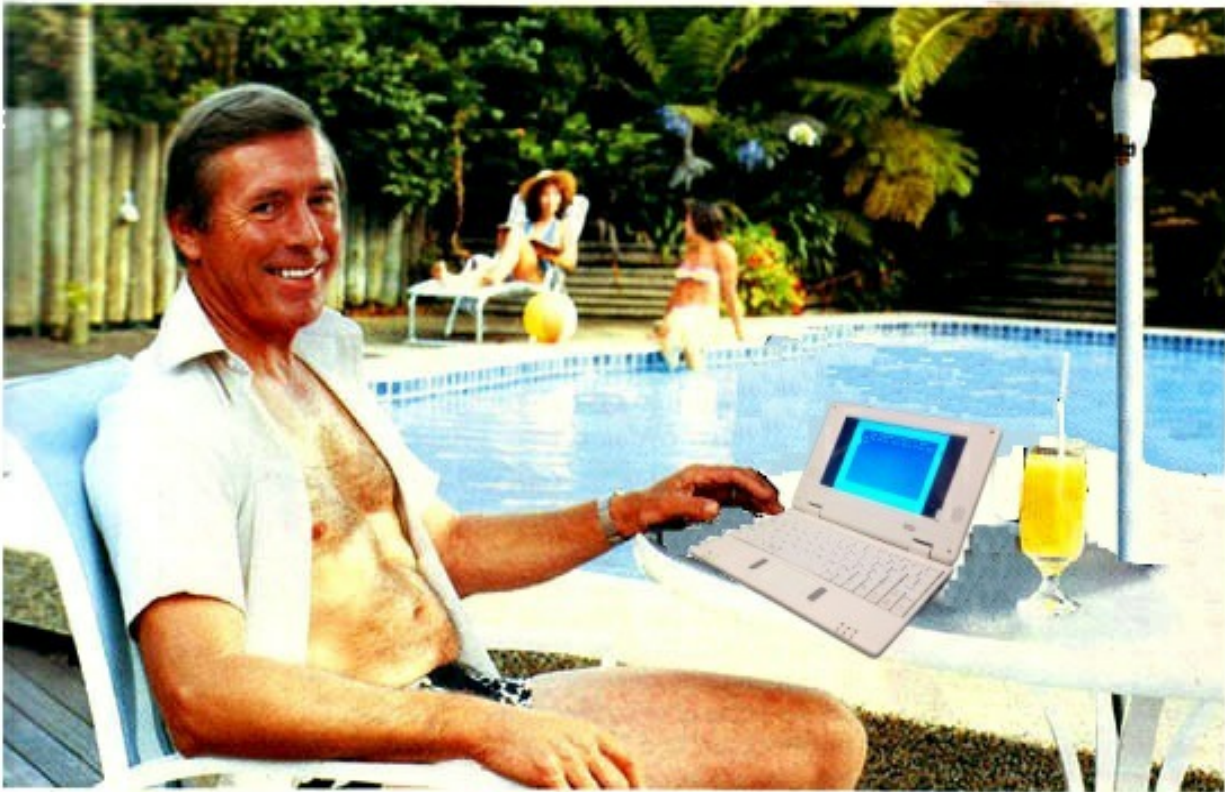
Loader Info : NOSDOS 1.X [W/b]

(1540/1541/1541B/1541C/1541-

II/1570/1571/1581/CMD HD/CMD FD2000/CMD FD4000)



Although I still see half a car/ truck / van thing appearing... hmmm... maybe its not 100% bug-fixed then.



## Who's keeping up with Commodore?



C64p



SD2IEC

 **commodore**  
COMPUTER  
Keeping up with you.

[www.thefuturewas8bit.co.uk](http://www.thefuturewas8bit.co.uk)  
Not just a place to buy SD2IEC's or a C64p  
The future really was 8 bit!



# Amiga News

## Amiga Forever 2014

Clonto have released the 2014 version of Amiga Forever emulation package

### Updates in this version include:

All Windows versions now come with a Windows Installer download. New Restore System Files feature, RP9 thumbnail provider, Visual Screen Clip Editor, Personal Paint 7.2 and the option for the use of the WinFellow emulator. Updates for AROS and the Enhanced RP9 Editor.

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



## AROS Vision 2.6.1 RELEASED

A new version of AROS Vision is available.

AROS is based on concepts and ideas of the Amiga and the 68000 processor and supports X86, ARM, PPC and 68k. Changes in this version: Added Amiga-E modules, AmiTwitter, AmiIRC, IBrowse Demo and Raystorm. Updates for the ROMs and DOpus Magellan. Changing Locale/Input is now more easy.

[http://www.aros-platform.de/html/distribution\\_download.html](http://www.aros-platform.de/html/distribution_download.html)



## A-EON Technology to distribute DVPlayer

A-EON Technology announced with the agreement of Stephen Fellner to become the sole distributor of DvPlayer, the premier multimedia player for AmigaOS 4.1. The software delivers the ultimate multimedia experience to AmigaOS 4.1. It is compatible with all Next-Generation AmigaOS computers including the AmigaONE X1000 and future models.

As part of the distribution agreement A-EON Technology will host the website and offer new versions of DvPlayer on AMIStore as and when they become available.

<http://www.amistore.net/>

The current version of DvPlayer runs under AmigaOS 4.1 and supports the following video formats:-

- MPEG-1/2 video streams
- MPEG-1/2 system streams
- MPEG audio files
- VideoCD
- AVI files (inc.OpenDML extens.)
- ASF/WMV files
- AIFF audio files

For more information please visit the official DV Player website: <http://dvplayer.amistore.net/>



## New products at Amedia Computer

Amedia Computer team has released some new products on his webshop,

### - PSX 1/2 pad / DB9 adapter :

for connecting your Playstation 1 or 2 gamepads on your Classic Amiga,

### - New CD32 GamePad in DB9 :

finally a new gamepad for your CD32 console and Amiga Classic compatible !

### - Rechargeable wireless optical mouse

for Classic Amiga : directly connected on your DB9 port of your favorite Amiga for more liberty !

<http://amiga.amedia-computer.com/>



## A-EON Technology Acquires Amiga.org

### OFFICIAL ANNOUNCEMENT

[A-EON Technology Ltd](http://www.a-eon.com) is pleased to announce that it is the new custodian of [Amiga.org](http://www.amiga.org), the world's oldest and longest running English language Amiga community web forum.

**A-EON acquired from the previous custodian, Bill Panagouleas of DiscreetFX, in an agreement which saw the transfer of the website and contents, the domain name together with all of the brands and rights including the official Facebook and Twitter accounts.**

**A-EON would like to thank Bill for his stewardship of Amiga.org over the past few years. Matthew Leaman, A-EON Managing Director, said of the purchase, "Our acquisition of is part of our long term strategy to help support and expand the Amiga community. We are really excited by this development."**

**Outgoing custodian Bill Panagouleas added, "I enjoyed my experience and am pleased to place the stewardship in the excellent hands of true Amiga enthusiasts. Amiga.org is safe and secure with A-EON Technology Limited."**

**Trevor Dickinson, A-EON's co-founder also commented, "speaking as an Amiga enthusiast, I am extremely proud to be associated with this**

**great piece of Amiga history. Amiga.org is oldest Amiga web portal, serving the Amiga community since 1994. This year marks the 20th anniversary of its founding and with the help of the Amiga community, whether it be Classic, Next-Generation, Emulation or new Retro hardware we hope to make an essential resource for all Amiga related news and information. Here's to the next 20 years!"**

**Full Press Release: [http://a-eon.biz/PDF/News\\_Release\\_Amiga.org.pdf](http://a-eon.biz/PDF/News_Release_Amiga.org.pdf)**

## A-EON Technology NEWS RELEASE

A-EON Technology acquires *AMIGA.org*

Cardiff 15th June 2014

A-EON Technology is pleased to announce that it is the new custodian of *AMIGA.org* the world's oldest and longest running English language Amiga community web forum.

A-EON acquired *AMIGA.org* from the previous custodian, Bill Panagouleas of DiscreetFX, in an agreement which saw the transfer of the *AMIGA.org* website and contents, the domain name together with all of the *AMIGA.org* brands and rights including the official Facebook and Twitter accounts.

A-EON would like to thank Bill for his stewardship of *Amiga.org* over the past few years. Matthew Leaman, A-EON Managing Director, said of the purchase, "our acquisition of *AMIGA.org* is part of our long term strategy to help support and expand the Amiga community. We are really excited by this development".

Outgoing custodian Bill Panagouleas added, "I enjoyed my *AMIGA.org* experience and am pleased to place the stewardship in the excellent hands of true Amiga enthusiasts. *AMIGA.org* is safe and secure with A-EON Technology Limited." Trevor Dickinson, A-EON's co-founder also commented, "speaking as an Amiga enthusiast, I am extremely proud to be associated with this great piece of Amiga history. *AMIGA.org* is oldest Amiga web portal, serving the Amiga community since 1994. This year marks the 20th anniversary of its founding and

with the help of the Amiga community, whether it be Classic, Next-Generation, Emulation or new Retro hardware we hope to make *AMIGA.org* an essential resource for all Amiga related news and information. Here's to the next 20 years!"

For more information please email: [contact@a-eon.com](mailto:contact@a-eon.com) or visit the A-EON Technology website at [www.a-eon.com](http://www.a-eon.com)

Supporting the Amiga community

## MUI 3.9 Released for AmigaOS 3.x (68k)

AmigaOS MUI development team NEWS:

<http://muidev.de/>

### Quote:

The MUI for AmigaOS development team is proud to announce the immediate release of version 3.9-2014R1 of the Magic User Interface for AmigaOS3/m68k. Please find the release archive in our download section <http://download.muidev.de/> This is the first release of MUI since MUI 3.8. At least a 68020 CPU is required. A faster CPU and a graphic board is highly recommended. Additionally MUI 3.9 will make use of 3rd party system extensions like AfaOS.

Like all former releases a keyfile is required to enable all available settings. Old keyfiles from MUI 3.8 can be used without any restriction. This release is explicitly declared as a beta release, because it been developed and tested on an AmiKit installation running on WinUAE only.

Please report any issues in the bug tracker along with a verbose description of the problem as well as detailed instructions on how to reproduce the issue. This was announced on Friday, but it looks like no one around here noticed. Nonetheless it's nice to see MUI updates for both 68k and PPC Amigas are back with an active development team.

MUI

AmigaOS  
Development Site



## 880 Gamer issue 3 released

In this English pdf magazine release for Amiga user are the following topics: Remake - Shadow of the Beast, Cover Disk - Putty Squad Game On: Alien Breed, Hybris, Putty Squad, the Quick Thunder Rabbit, Walker and Zany Golf, Bleeding Eyes: 40 and a Blunt, Tree, Energy Bar and State of the Art, Cheat and Talkback

<http://www.users.on.net/~stanners/>

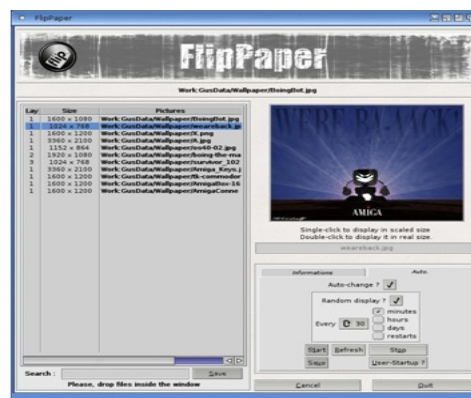


## FlipPaper Updated

Flip Paper is a MUI program allowing you to manage your wallpapers and to automatically change your wallpaper. The program is available in the following languages: English, Czech, Spanish, Dutch, Greek, Swedish and German.

**Recent changes:** Selected tag is always visible in the Workbench drawer. Added a picture Delete menu option and Send by mail menu. And an update to the Spanish translation.

<http://www.os4depot.net/index.php?function=showfile&file=utility/workbench/flippaper.lha>



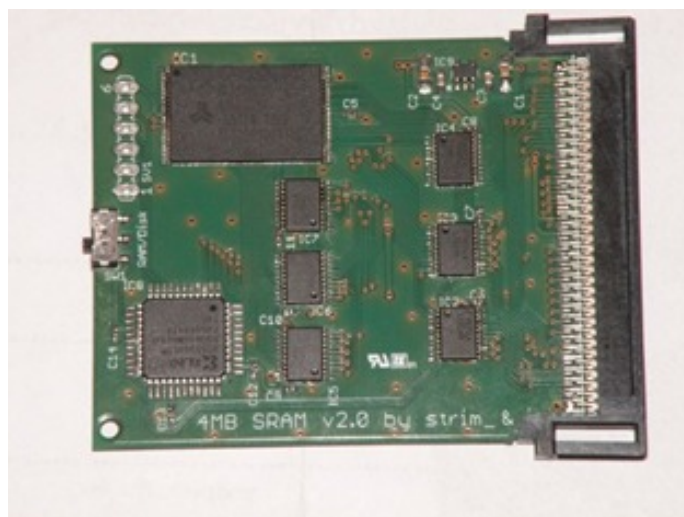
## Sakura 4MB PCMCIA SRAM

Sakura card is a prototype Fast RAM expansion for Amiga 600/1200. Add 4MB of Fast RAM to your unexpanded Amiga 600/1200. This additional memory will make your system much faster and allow to run more demanding games and applications.

### The product has following features:

- Adds 4MB of Fast RAM to Amiga 600 or Amiga 1200
- Accelerates unexpanded Amiga 1200 to 1.67 MIPS (according to SysInfo)
- Built using modern, high performance 55ns SRAM ICs
- Very simple installation - just insert the card into PCMCIA slot on the left side of Amiga
- Compatible with all PCMCIA friendly accelerators and memory expansions\*
- Open source design
- Made by Amigans for Amigans!

[http://retroami.com.pl/product.php?id\\_product=112](http://retroami.com.pl/product.php?id_product=112)



## Hollywood documentation available online

the documentation of Hollywood 5.3 is available online. You can access the documentation in HTML and PDF format in the "Help" section of the official Hollywood portal. MUI Royale documentation has also been made available online

<http://www.hollywood-mal.com/>



## Interview with Szilard Biro in Obligement magazine

Obligement magazine has recently published an interview with Szilárd Biró, a Hungarian developer who made many ports for AROS, MorphOS and AmigaOS 4.

French version :

<http://obligement.free.fr/articles/itwbiro.php>

English version :

[http://obligement.free.fr/articles\\_t...itwbiro\\_en.php](http://obligement.free.fr/articles_t...itwbiro_en.php)

Obligement - The Amiga online magazine

<http://obligement.free.fr>



The Adventure Continues In:

# REALMS OF QUEST IV

A Ghislain De Blois Production



PSYTRONIK SOFTWARE

**Featuring:**

- 10 Dungeon Levels which have been designed by hand with numerous rooms, secret passages, riddles and traps to traverse and overcome.
- 10 races and 12 classes to choose from to create 8 players for your party.
- You will meet other heroes to help and guide you during your quest.
- Over 100 unique graphical portraits that depict the various monsters that you'll encounter as you travel throughout the Dungeon.
- Over 135 types of equipment to represent weapons, armor, gems, jewelry, potions, scrolls and miscellaneous magical items.
- 60 magic spells that you can cast.
- 3 game fonts to choose from.
- A 16 page printed instruction manual is also included.

THREE full adventures for the VIC-20 along with Ultimate Quest, Catacomb for the Commodore 64!

PSYTRONIK SOFTWARE VIC-20

www.psytronik.net

ONE COP. ONE DRUG LOAD. NO MERCY

# THE VICE SQUAD

A C64 Game by Achim Volkers & Trevor Storey



PSYTRONIK SOFTWARE

**GAME FEATURES:**

- Multiple stages to battle through!
- Superb varied backgrounds!
- Awesome weaponry!
- Mission briefings!
- Boss battles!
- Funky Soundtracks!
- Addictive arcade style gameplay!

PSYTRONIK SOFTWARE www.psytronik.net

www.rgcd.co.uk

NOW AVAILABLE ON C64 TAPE, DISK & DIGITAL DOWNLOAD FROM WWW.PSYTRONIK.NET AND ON C64 CARTRIDGE FROM WWW.RGCD.CO.UK

Trevor Storey and Georg Rottensteiner Present:

# BOULLESA

An Epic Arcade-Adventure For The Commodore 64!



PSYTRONIK SOFTWARE

**GAME FEATURES:**

- Massive Map To Explore
- Stunning Animation
- Searchable Objects
- Atmospheric Soundtracks
- Detailed Background Graphics
- Magic Effects
- Separate Animated Intro + End Sequences
- Spawn Points
- Spirit Stone Puzzle
- A3 Poster/Map
- Companion CD

UNLEASH THE BEAST FROM YOUR C64!  
Now Available For C64 On Tape, Disk, Cartridge + Digital Download

PSYTRONIK SOFTWARE www.psytronik.net

www.rgcd.co.uk

SIX C64 GAMES IN AN EXPLOSIVE NEW COMPILATION!

# SHOOT 'EM UP 3 DESTRUCTION SET



2014 C64 RELEASE!

AN ALF YNGVE PRODUCTION!

MUSIC & ENHANCEMENTS BY RICHARD BAYLISS  
LOADING BITMAPS BY CARL MASON • COVER ART BY OLIVER FREY

PSYTRONIK SOFTWARE www.psytronik.net

NOW AVAILABLE ON C64 TAPE, DISK & DIGITAL DOWNLOAD!

# Amiga News

## Artist H.R. Giger Passes (1940-2014) RIP

Surrealist artist Hans Rudolf "Ruedi" Giger passed away at 74. Widely known for his work on the Alien film series and of special interest to us: for the creepy and extremely tough Amiga game Dark Seed.

### Dark Seed long play on youtube

<https://www.youtube.com/watch?v=s3dGvTaQjd8>  
<http://memolition.com/2014/05/13/alien-designer-hr-giger-dies-at-74/>  
[http://en.wikipedia.org/wiki/H.\\_R.\\_Giger](http://en.wikipedia.org/wiki/H._R._Giger)



## The Best of MorphOS January/February 2014

The [WARmUp](http://www.warmup-asso.org/) association (world Association of Morph os users) <http://www.warmup-asso.org/> presents you with a PDF entitled the **Best of MorphOS March/April 2014**.

### Read the webzine.

[http://www.warmup-asso.org/WArMUp/BestOfMorphOS\\_03-04-2014\\_en.pdf](http://www.warmup-asso.org/WArMUp/BestOfMorphOS_03-04-2014_en.pdf)



World Association of MorphOS Users

## XU4 - Ultima IV Recreated

Ultima IV: Quest of the Avatar is the 4th instalment of the Ultima series by Richard Garriott originally released in 1985, and later ported to many platforms (including Amiga).

XU4 is an engine that can run the original datafiles, and also supports some enhancements. However, the game needs freely available additional data files to work.

The MorphOS port can be found here:

<http://jupp3.binaryriot.org/releases/ports/xu4-1.0beta3-morphos.lha>



You can read the original documentation online here:  
<http://paradroid.net/u4/docs.html>

## Dopus5.90 is released

After over 15 years of silence an updated, fixed, ported and whatever else version of dopus5 is released to whole amiga community.

### Download it from here

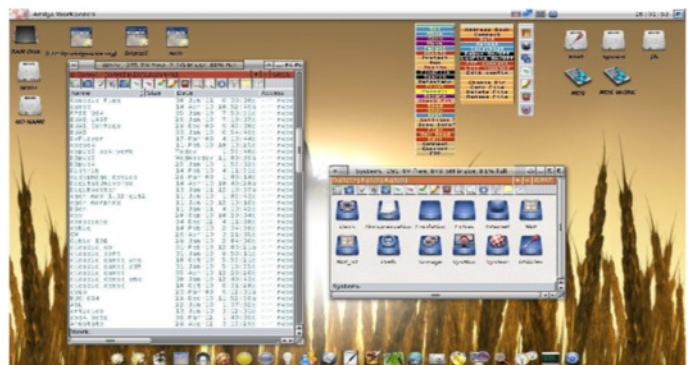
<http://dopus5.org>

Sadly no morphos version is available .

**68k version:** os3 with ks3.1 minimum.

**AmigaOS4 version:** Any version of amigaos4 may works fine

**AROS-i386 version:** latest deadwood's brunch or latest WIP's of ICAROS.



## Yerzmyey - Rave is illegal!

### Yerzmyey - Rave is illegal!

Yerzmyey has released a new album called Rave is illegal! All the music has been created on a AMIGA 500 (using just 4 channels) <https://soundcloud.com/yerzmyey>



## Full versions for download: THE Puzzle game Pinomania and Termite TCP

The puzzle game Pinomania <http://hol.abime.net/5725> and the TCP-Stack Termite TCP are available for free download. However the Termite TCP stack only supports Dial-Up-connections.

[http://www.amigafuture.de/dl\\_recent.php](http://www.amigafuture.de/dl_recent.php)



## MUI Royale 1.1 released

Airsoft Softwair has announced the availability of MUI Royale 1.1.

### This is an update and contains some minor fixes and new features.

MUI Royale is a plugin for Hollywood which allows the user to easily create MUI GUIs with Hollywood. MUI Royale supports over 40 MUI classes including popular third-party classes like TextEditor.mcc and TheBar.mcc. Creating and managing menustrips is also fully supported. The plugin sets new standards concerning the ease of use because the GUI layout can be conveniently defined using an XML file

that is converted into a MUI GUI by MUI Royale on the fly. GUI design has never been so easy! Of course, new features of MUI 4.0 are also supported.

MUI Royale requires at least Hollywood 5.2 and is provided free of charge for all users of Hollywood.

It is available for AmigaOS 3 (m68k), AmigaOS 4 (PPC), MorphOS (PPC), and AROS (i386).

Download now available at the official Hollywood portal @

<http://www.hollywood-mal.com/>



**MUI ROYALE**  
Easy GUI creation for you  
Kings and Queens of MUI!

## Official AmigaOS Documentation Wiki

The official AmigaOS Documentation Wiki has been online for a few years now and gone through many changes since its start.

With new developer material, the wiki also contains previous AmigaOS developer documentation including the RKRMs, AmigaDOS Manual, AmigaMail articles, AutoDoc style guide and Amiga User Interface Style Guide.

You will also find manuals for [ARexx](#), [AmigaDOS](#), and [Workbench](#), along with 3rd-party software manuals such as [Bars & Pipes Professional](#).

If you have something to contribute to The AmigaOS Documentation Wiki please contact the AmigaOS Development Team and volunteer.

<http://wiki.amigaos.net/wiki>



## WinUAE 2.8.0 Released

Version 2.8.0 of the famous Amiga emulator WinUAE has been released.

### New features include:

- Full A4000T and A4091 NCR53C710 SCSI emulation
- A590/A2091 and A4091 boot ROM GUI selection
- Separate graphics filter settings for native and RTG modes
- 256k ROM image inserted in floppy drive emulates A1000 KICK disk
- Super Card Pro image file support (.scp)
- SLIRP network inbound port support (default: 21, 22, 23 and 80, others can be added by editing configuration file)
- Input panel previously toggle-only events can be optionally set to on and off state, audio/video recording input event added
- Joystick/joypad can be used to control light pen cursor

### Updates:

- Remaining 68000 cycle-exact mode timing fixes
- 68000 reading from write-only or non-existing custom register compatibility improved
- Big chipset edge case compatibility update, for example demos with vertical \"copper\" bars work

perfectly, real hardware glitches in horizontal scaling are now accurately emulated and much more

- 68040/68060 without emulating unimplemented FPU instructions is now fully compatible with Motorola FPU emulation library, FSAVE special FPU exception stack frames implemented
- Programmed modes (Super72 etc.) now automatically select best fit horizontal resolution and is more compatible with different filter modes, also display positioning is improved
- Recently-dumped Arcadia arcade system ROM images supported
- Debugger full FPU and 68020+ bitfield, and other previously missing 68020+ only instructions supported in disassembler
- Magic mouse + mousehack mode now always stops keyboard input when mouse is outside of emulation window, even if window still has focus
- Hyperion Entertainment Blog

<http://www.winuae.net/>



# Vic and Plus 4 News

## Note-Game - Plus/4

A programmer known as Champ has created his first game on the Commodore Plus/4 and named it Note-Game. The game is written in Basic and available in both German and English languages. In the game you learn musical notation to earn points.

Seems to be a very "NOTE"-worthy release for the Plus 4. NOTE down the url and .... Ok, I will stop now with the puns...

<http://plus4world.powweb.com/software/Note-Game>



## Club Info issue 134 released

This is a German disk magazine (d64) for the Commodore Plus / 4 with the following articles: Tips and Tricks, Spaß Computer, Hardware, Total Eclipse, SVS Calc 2.0 Other systems, NSA, Katze ASCII Print, Meriday in the morning, Dunzhin - Warrior of RAS, ARCOS C16 Kristall-Labyrinth, 2048, Flashback, Calculate It, Indiana Jones PETSCII, Going Up PETSCII, Walk Sprite Animation and Far Cry.

<http://plus4world.powweb.com/software/Club Info 134>



## PULSE TAPE RELEASE DELAYED

The one thing missing from Pulse was a physical tape release. Having contacted many people to try to make this a reality and having no response, [Sven Klose](#) took the problem into his own hands and decided he could release the game on tape himself! Real life came in the way, but he is confident of a physical release of the game. Not only that, but it looks like the game will be heavily tweaked and re-worked.



### Taken from the website

Life is getting better

The optimistic release date for Pulse 2 passed and I'm very, very sorry about that. With moving into a new flat and office and metric cart-

loads of new work to do there's hardly any spare time left. But be ensured that it's still being worked on. Just adding a bit more level data doesn't do it for me. I want this thing to knock you off your socks - I love your applause. B) Am afraid you have to give me another couple of weeks.



### The free release of pulse is out - may the source be with you!

pulse is a well-received, fast, horizontally smooth-scrolling shoot-'em-up I wrote for the unexpanded Commodore VIC-20. Download the program file here

<https://github.com/SvenMichaelKlose/pulse/blob/master/pulse.prg?raw=true>.

You can get the source on Github

<https://github.com/SvenMichaelKlose/pulse>. There's also a video of the thing on real hardware; thanks, Mike! Pulse is being discussed at VIC-20 denia

<http://sleepingelephant.com/denial/>. Robert Hurst has set up the pulse highscore table and a download page with quite a review. I'm SO out of words here. I won't mind your upvote on pouet.net. 0:)

<http://hugbox.org/pixel/>

## VIC 20 GAMES

**Name:** FIFA World Cup 2014

**Author:** Jeff-20

**Released:** June 2, 2014

**Requirements:** VIC-20 (no memory expansion required), keyboard controlled

Description: A simple, arcade style soccer/football game.

- all 32 participating national teams
- one player elimination mode ("knockout rounds")
- two player vs mode
- two player vs mode
- adjustable skill levels



### Download

<http://www.sleepingelephant.com/denial/games/FIFA2014.zip>

### Discussion

<http://sleepingelephant.com/ipweb/bulletin/bb/viewtopic.php?f=10&t=6997&p=76137#p76137>



**Name:** Ice Hockey 2014

**Author:** Jeff-20

**Released:** June 2, 2014

**Requirements:** VIC-20 (no memory expansion required), keyboard controlled

Description: Essentially, a variation of World Cup 2014. A simple, arcade style hockey game.

- all 16 participating national teams in IIHF Championship
- all 30 NHL hockey teams with two defunct teams for added measure
- one player elimination mode ("knockout rounds")
- two player vs mode
- adjustable skill levels



### Download

<http://www.sleepingelephant.com/denial/games/FIFA2014.zip>

### Discussion

<http://sleepingelephant.com/ipweb/bulletin/bb/viewtopic.php?f=10&t=6998&p=76131#p76131>

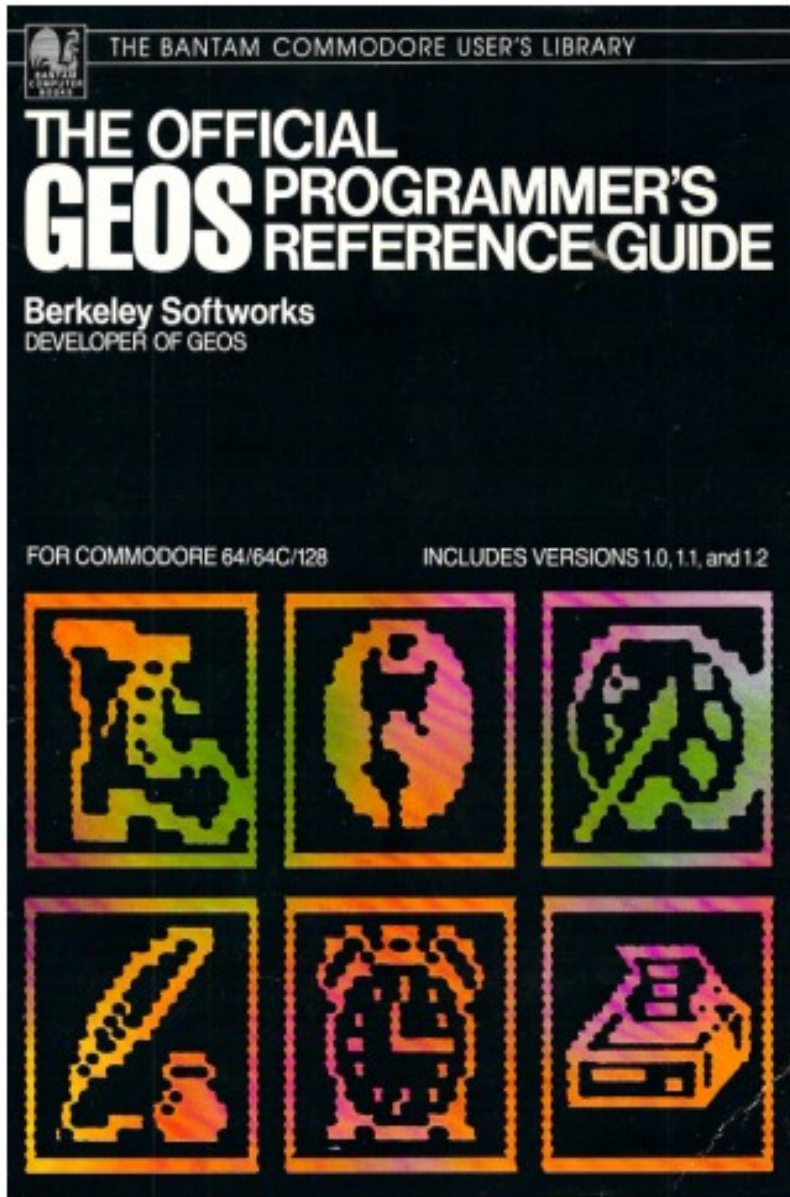
### Commodore Free

Two simple games tightly coded in Basic. However, the games are fast and, although sharing 90% of the same code base, are different enough for you to take a preference. For example, I can't understand football so I played the Hockey version. I found the game very entertaining, reminding me of something like the Nintendo game and watch variations that were available. The games are well-thought out, with just enough animations to make them interesting. The players are basically roted to the spot and just swing out to move the ball, puck or whatever. I didn't have time for a full review but I will rate them both together

<b>Graphics</b>	7/10 Simple, but well-animated	<b>Overall</b> 7/10 Fast, well-coded, and fun
<b>Sounds</b>	3/10 Just some blips	
<b>Gameplay</b>	7/10 You need a friend to really enjoy the fun	

# GEOS and Wheels programming basic code

## By Terry Raymond



There is hope though, as sometime around 1999 The author of the upgrade for GEOS (Maurice Randal) released Wheels OS he also upgraded the Geo Programmer package and with the first release fixed all original Geo Programmer bugs, the package was called: Concept. Concept is similar to Geo Programmer in that it's very limited on the size of the code, labels etc., etc., you can use, but one could at least code a decent 8-bit GEOS application.

Maurice went one step further to support the CMD Super CPU and came up with: Concept Plus with a number of great improvements:

- 1- Faster assembling and Linking times.
- 2- Offers better Wheels memory management (little better than GEOS).
- 3- Creates both 8-bit and 16-bit apps.
- 4- Author includes in Concept package: New Macros (beyond my understanding).

Concept & Concept Plus, to my understanding, are now freely distributable (as Maurice has left the scene now). If, however, people want to try it out, and have more interest in GEOS/Wheels programming then feel free to contact me.

I would also suggest newbies get hold of the original GeoProgrammer manual and disks, because on the disks are the original GEOS Macros that are needed, but I won't go into that in this article. Now onto how to code a GEOS/Wheels File Header.

This header is kind of similar to a standard CBM file header with of course a few differences for the GEOS system, it contains information about the Application, including a small Icon that is clicked on to start any GEOS Application (more on that later).

Hello, I am Terry Raymond and together we will be delving into some articles on basic GEOS 8-bit programming, using Machine Language. Our code will also work with Wheels OS, too. To get you started I need to tell you about an update for Geo Programmer, then we will start with GEOS/Wheels Header file.

### The Geo Programmer update:

The original Geo Programmer package for programming your own apps, etc., was okay – but contained a lot of bugs. If you are a well-seasoned ML programmer you will probably know about all of the workarounds for these bugs and this is not a problem. However, for beginners this can be the most difficult and frustrating part to overcome, so many just give up.

I will mention here that I'm very new to 6502 ML and GEOS programming, but I will not try to teach that since I'm barely grasping that myself and learning.

With any 6502 ML and GEOS programming that use ML Opp codes, and GEOS/Wheels OS uses its own unique type of routines. Also ML uses in its code its own Comments that usually have a semi colon like:

*; Start of code*

This describes what each line of code is doing but sometimes hard to follow or understand.

To start the Header file code:

```
,*****
;My application
;Header file
,*****

.header          ;start of Header file
                ;section
        .word    0      ;first 2 bytes - zero ( after .word is a
zero)
        .byte    3      ;width in bytes
        .byte    21     ;height in scanlines of
(LEAVE 2 BLANK SPACES HERE) very important
                ?Use Geopaint to create your Icon, then cut
the image, then
paste the image into this code HERE. More on this in a later
article.

(LEAVE 2 BLANK SPACES HERE) very important

        .byte    $83    ;Commodore file type,
                ;with bit 7 set
        .byte    6      ;GEOS file type:
                ;APPLICATION
        .byte    0      ;GEOS file structure
                ;type: SEQUENTIAL
        .word    $0400   ;Start address
        .word    $03ff   ;Highest end address
                ;only need for
                ;Desk accessories
        .word    $0400   ;init address
```

```
;Permanent command name (or class of file) must be 12
characters
;including Spaces plus Version number 5 characters
followed by three
; zero values. The final byte here is the SYSTEM byte. $40
allows this
;command to run on any GEOS system. (also Wheels).
;(REFER TO GEO PROGRAMMER USER MANUAL
APPENDIX-C)
```

```
        .byte    "App name      V1.0",0,0,$00
```

```
; the final byte can be:
;$00 - Only 40 column mode (C64, C128 40 column mode
only)
;$40 - 40 and 80 column mode (both C64 and C128)
;$80 - GEOS 64 only
;$C0 - Only 80 column mode (80 column mode for C128)
```

```
;Twenty character author name including the Null-
Terminator
;Change this to your own name
```

```
        .byte    "Author Name      ";0
```

```
        .block   20     ;Free area for GEOS internal use ???
```

```
        .block   23     ;Free area for GEOS internal use ???
```

```
;The info text including the Null- Terminator :Max 96
characters
```

```
        .byte    "App's  Function",0d
        .byte    "Same      ",0
        .endh     ;end of Header section
```

BTW, the comments mentioned at the start are only for commenting in the code and do not use any of the System Memory for the Application etc.

The next article we will explain the small Graphical Icon and how to create it etc.

If anyone has any basic questions about GEOS (basic code) I may be able to answer some questions etc, so feel free to contact me:

Terry L. Raymond  
email: traymond20@gmail.com  
Enjoy GEOS programming till our next article. -Terry

### Further reading

Wheels the GOES upgrade  
[http://en.wikipedia.org/wiki/Wheels\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Wheels_(operating_system))

Geos programming wiki  
<http://commodore64.wikispaces.com/GEOS+Programming>

Geos programmers reference guide  
<http://www.zimmers.net/geos/docs/geotech.txt>

GEOS FAQ  
<http://www.zimmers.net/geos/GEOSFAQ.html>

Shadow m`s Geos  
<http://www.lyonlabs.org/commodore/onrequest/geos.html>



# Commodore: 1 -- Church: 0

## by Lenard R. Roach

It will be four years this coming December 27th that "Run/Stop-Restore: 10th Anniversary Edition" came out on the shelves as, I hope, a different way to approach Commodore computing and writing. The response to my collection of writings by the Commodore public has been far more gratifying and profitable than my two Christian publications combined. This does not bode well for the church but it speaks volumes about the Commodore community and how well they do support their beloved computer. In order to have a publication sell in the Christian world the book has to be endorsed or forwarded by a big name preacher or Christian ministry. My books have not. However, Mr. Parker at "Commodore Free" magazine has graciously granted me more than one interview over "Run/Stop-Restore: 10th Anniversary Edition" and even dedicated an interview for me in his publication on some of the programs I've written over the last twenty or so years that I've been sitting behind a Commodore keyboard. This bolstered sales of all my Commodore wares very well, which has brought me to a crossroads in my writing career ... do I continue to write books for a Christian world that doesn't know I exist, or do I change venues and write more for the Commodore machine where I know I will get an honest and fair review, and even a little pocket change?

The simplest answer to the above question is to just ... write for both. True and simple indeed, but there is the problem of cash flow that seems to rise it's head and makes itself pronouncement. The Christians I deal with here locally in the Kansas City Metro area are probably like those in any community around the world -- if they can get it for free, then heck yeah, by all means, let's use it. I've written two books that are out for the Christian public ("Skits For 2nd Hand Puppets Volume 1: The Ten Commandments" and "Skits For 2nd Hand Puppets Volume 2: Adventures In Courage") with a third book

sitting on a flash drive waiting for the outcome of the first two books on sales (working title: "Skits For 2nd Hand Puppets Volume 3: Twisted Parables"). I don't mind doing things for free once in a while, but when Mr. Bill Collector comes knocking at your soon-to-be foreclosed front door demanding payment on a debt, with a line of creditors behind him that flows well into the street and around the corner, you begin to wonder if all these "freebies" are making a difference in your world. I've received a lot of "Lenard, you do such a good job," or "Lenard, the kids love it," which is fine. I do the same thing for my cats when they decided to use the litter box instead of my living room carpet for a toilet; lots of prestige, but no payback that helps in the daily living expenditures of the Roach Center For BASIC Commodore Studies. No wonder I work three jobs and still try to have a writing career. That's one reason why I don't like doing both.

"But Lenard, aren't you giving freebies with Commodore writing, too?"

I did that for many moons until the idea of "Run/Stop-Restore: 10th Anniversary Edition" came to light. That's where I took a large collection of Commodore article writings and combined them into a short, easy to read volume and got them published under a bigger name publisher than just me. Also, I've been given plenty of commentary, reviews, and blurbs throughout the Commodore universe (like this magazine) that opened up sales and gave me a little pocket change to where I could afford a Hamburger Helper once in a while instead of Tuna Helper all the time. Can I get the same from the Christian world? Only if I start flashing them some greenbacks at them. Nobody seems to do things for Jesus anymore unless they can get a buck or two out of it for themselves. This does not speak for ALL Christian ministries, of course, but just listen sometime to either a TV or radio preacher and see if they don't ask the listeners for a "contribution."

Now let's look at Commodore users (and I'll use the example of the prepara-



tion of my latest program, "The Envelope Addressor," to illustrate.) When I came up with an idea to write a program for the Commodore 64 and laid down the BASIC code for it, I received an instant response for the Commodore community. Sure, I got a lot of "Good jobs" and "Can't wait to see the full blown version," but there were also those who came up with suggestions on what they think I should add to the program, like a directory reader or other subroutine. To tell the truth, most, if not all, of my later works after 1992 for programming involved the Commodore community at large, in some capacity or another. Those who helped me the most were mentioned in the credits at the end of the programs. The club or institution that helped me the most were added to the start up screen, so anyone using the product would have to read that at each boot up. That could be YOU, Commodore reader, if I ever get another inspiration to sit back and BASIC program again. Sadly, it seems like such epiphanies only come about once every three or four years, and my next one is overdue by about two years.

When it came to Commodore, the people were enthusiastic and helpful, rarely critical (except during extreme cases where a programming blunder would be inevitable), but always excited to see something new come out for the Commodore. Sometimes some users just didn't care if the application I was working on was something they would never use personally; they were just excited to see something new for the Commodore

The Christian community was different. They rarely gave feedback on work I was doing, except to tell me that my material was "unfit for their criteria." When I asked them what they were looking for, they would respond with something to the effect of "Never mind. We have decided to go a different avenue. Your services are no longer required." I decided that with all these unused skits and plays with a Christian theme sitting around the house, I would do with them as I did with Commodore and compile them into a book and try and sell them. Do you want to know my total sales of volumes for the working year 2013 for my Christian books? One -- and that one I had to

practically beg them to purchase, just so I could have a sale on the records. I have given away dozens of copies of my Christian play books hoping that word of mouth would generate sales, but all people are doing is taking the free book and copying the contents (not all the content; just the parts they want to use) and leaving my book on the shelves to rot. The Commodore community stepped right up when I had chances to advertise (in many cases, for free) in magazines and online book stores and didn't beg or ask for a "free" copy, but dug right into their wallets and paid me for my efforts on Commodore writing. Even as I write this article, I'm glancing at my latest royalties check from the book publisher that came in the mail.

"Run/Stop-Restore: 10th Anniversary Edition" has been on the market now for a while and I'm still getting residuals from it today. This is all thanks to the Commodore community (readers like you) and those Commodore enthusiasts who display my material at their conventions, even though I can't be there personally to sign autographs or talk to the readers. I hate to say this about God's church, but it seems like there is more unity and camaraderie within Commodore, as spaced apart as the users group might be these days, than there is in the walls of a church sanctuary.

So I'm back to the question of what to do with my writing career. I get sick of working late hours on my computer's word processor wondering if my latest work will meet the criteria of what readers are looking for today, whether it be Commodore or Christian or whatever. Still, I can "walk" into a Commodore magazine, interview, program demonstration, or book signing and be greeted with open arms, a smile, a handshake, and even a pat on the back for a work done. I try my best not to claim mastery over anything I do in either realm as a "final authority," but it sure is bad that only Commodore enthusiasts are willing to shell out comments -- and cash -- for the things that I have done.

Maybe the church needs to get off their "blessed assurance," march out into a Commodore convention, and see how things can be done just a little bit bet-

ter in the way people are treated. My ex-wife gave Robert Bernardo, president of the Fresno Commodore Users Group, some static a few years ago when it came to my first appearance at one of the club's CommVEx conventions because it was in Las Vegas, Nevada, better known to the public as "Sin City." Robert, in his wisdom, simply answered her query with one of his own: "And Kansas City is much better how?" She was quiet after that and retreated back into the sanity of her quiet church pew. Well after our divorce, I found out that she never enjoyed our family trip to Las Vegas that year because of all the explicit activity, and she would never go back to the city again.

Commodore has been, and always shall be, a venue wherein a person can always be met with consideration and compassion when it comes to computers. It was Commodore users that taught me to always be understanding of other formats that were being used out there and to be helpful when they ran into a problem on said format in hopes that my Commodore knowledge will be a possible solution for them. Even today, when some user runs into a problem, I always think of what I can do with my Commodore that will possibly be beneficial to them with an answer

So, in conclusion, I will decide to continue to work on both careers, Commodore and Christian, so that hopefully one or the other will profit me. Like it says in Ecclesiastes 11:2, "*Divide your portions into seven, yes, even into eight, for you do not know what disaster will fall upon the earth.*" Who knows? Maybe both will prosper well and I will be up in the money, or I'll stay struggling like so many other people around the world.

But before I go, let me once again say, "Thank you," Commodore reader, for making me a success in what I do.

# Making a C64/C65 compatible computer in an FPGA



humanitarian telecommunications at Flinders University in Australia (see <http://servalproject.org> and <http://servalproject.blogspot.com.au> for more details). I have also achieved some notoriety for creating a working shoe phone (<http://realshoephone.com>

<http://c65gs.blogspot.co.uk/>

Let's jump straight in as the interview will explain the title for those still confused about the FPGA acronym, as they say, all will be revealed, Well not everything but you know what I mean.....

Anyway I will leave you with the interview as I have a call coming in on my shoe!

## Q. Please introduce yourself to our Commodore Free readers.

My name is Paul Gardner-Stephen, and I work as a lecturer in computer science and researcher in rural, remote &

## Q. Can you give our readers a little history about yourself and your computing love?

I have been using Commodore 8-bit computers since the mid 1980s. I first used Commodore 64s at school, but the VIC-20 was the first computer I was able to buy for myself. In high school I wrote 64NET to allow my Commodore 64 to use the hard drive in a PC via a custom parallel cable. 64NET went on to be sold around the world, but mostly in Germany. The creation of 64NET also resulted in an invitation to join the tools section of FairLight64.

As time went on, I was able to upgrade, first to a SX64 and then a C128D, appreciating the internal drive and improved keyboard of both. I never really used the advanced features of the C128, partly because I found the memory management system to

be rather unappealing and overly complex, and the performance of the VDC to be uninspiring.

However, I remember hearing the persistent rumours about the Commodore 65 during the 80s and 90s, and imagining what an improved 8-bit computer, that really was a genuinely enhanced Commodore 64 might be.

I also started thinking about how to make a really good C64 accelerator in the late 1990s, that didn't suffer from slow memory write-back or glitching of the display if memory writes were cached and delayed. Looking at the bus arrangement of the C64 I realised that it was possible to avoid these problems by pulling the write line low during a VIC-II memory access, because nothing drives the write line during VIC-II memory accesses. This would make the RAM "listen" to whatever was put on the data lines. The VIC-II would of course still drive the address lines.

With fast enough logic and external memory, it would be possible to watch the address lines, and present the appropriate byte of memory to the bus, and so create a just-in-time memory synchronisation system.

I did some early experiments on making such an accelerator, proving that the system could work, but didn't have the equipment or other resources at the time to make it a reality. Also, around the same time, the German distributor of 64NET sent me a Commodore 65 prototype. My immediate need for increased speed and performance were met, and I used the C65 as my main development machine for much of the late 1990s. Turbo Assembler was a dream at 3.5MHz, and the internal 3.5" floppy, while still rather slow (more on this in a moment) was still a considerable improvement on a 1581. This meant that I gained quite a good working knowledge of the C65, and even wrote a few C65-only applications, including a simplistic micro-emacs like 80-column text editor.



**Q. Can you explain some of the features of the Commodore 65 and what happened to it for the readers who are unaware of the machine?**



To me, the C65 has the sense of being what the creators of the C128 might have liked to have done, given enough time. It also contains obvious lessons learned from the C128, principally that all advanced features can be accessed from C64-mode. In fact, the C65 boots in C64 mode first, and a modified C64-mode kernel decides whether to jump to C65 mode.

In terms of specification, the C65 is like a cross between the C64 and Amiga 600. It has stereo SIDs (the newer 8581 type).

The units shipped with 128KB of RAM, and an Amiga-like trapdoor slot that theoretically take another 8MB of RAM, but not without some rather scary memory banking tricks, involving two nested levels of memory banking. The trapdoor slot mapped 512KB at a time into the 1MB address space of the 4510 processor, and required the memory expansion to include a special register to select which 512KB bank was required.

The byzantine logic to make use of this was built into the DMA magic DMA controller, or at least was supposed to be. I don't know if anyone knows whether it was actually all implemented in the experimental units. The DMA unit is however handy for memory fills and copies, and is heavily used in C65 mode for screen clearing, screen scrolling,

and accessing BASIC variable and program storage.

The C65 has an internal 3.5" floppy drive which is compatible with disks from the 1581, but the DOS runs in the C65 itself. The DOS itself was rather buggy; erasing files or validating the disk is not recommended unless you want to mess things up.

This may have been improved in later versions of the ROM, of which there were many.

The CPU is a 3.5MHz 4510, which is an extended version of the 6502 (the VIC-III provides the 6510-like CPU IO port in the C65), but also including the 6526 CIA adapters. The 4510 has a number of interesting new addressing modes, as well as an extra index register (Z), and some special registers for memory mapping. The B register sets where zero page lives, now renamed to "base page". There are also unnamed memory offset registers that allow access to

1MB of memory. The 64KB of address space is divided into eight 8KB chunks, in two sets of four. Each chunk can point to its normal address in the first 64KB, or have an offset applied that allows it to point to any 256-byte address in the 1MB range. The offset is common for each of the four chunks in one half of memory, which places some practical limits on the memory mapping. The bottom few pages of memory have no special treatment, unlike on the C128.

A lot of the extra opcodes in the 4510 (all 256 are defined) are dedicated to more efficient bit fiddling operations which make for more compact and/or faster code in many cases. A number of the new opcodes cover 16-bit relative branches, a relative version of JSR, allowing for completely relocatable code.

Finally, there are new address modes, like stack-indirect, and a 16-bit stack mode that make it much, much easier to write a C compiler for the 4510 than for the 6502. Many of these new instructions are common to other variants of the 6502, but not always with the same opcode.

This means that 6502 illegal opcodes don't work on the C65. This isn't much of a problem for most C64 software.

However, the engineers also shaved a cycle off INC, DEC, ASL and the other read-modify-write instructions, remov-



ing the dummy write that happens on the 6502. Unfortunately, this broke lots of software that expects ASL \$D019 or DEC \$D019 to clear raster interrupts -- because it was the dummy write that cleared the interrupt. This is probably the single greatest source of incompatibility between the C64 and C65 for common software. Technically simple games like Impossible Mission just wouldn't work. I do recall patching a few games to clear interrupts on the C65 and getting them to work, or nearly work.

The VIC-III video controller is the other really interesting advance in the C65.

It keeps all VIC-II modes, although to "make life easier for the programmer", raster splits that change the video mode only take effect at the start of each character row. This means that clean raster splits are easy to write, but it probably means that some advanced tricks, like FLI and DMA-delay based effects may not work on the VIC-III. It also has a 256-colour palette with 12-bit colour (4,096 possible colours like the Amiga 500).

32 colours can be used in text and normal bitmap modes, through the use of an extra "bold" flag bit in the colour RAM (which is not 8-bit wide, and is really just 2KB of the 128KB internal RAM, unlike on the C64 where it is a separate 1KB x 4 bit RAM). The palette can be changed at anytime, so it is possible to have many more than 32 colours visible in a single frame. To have more than 32 colours, it is necessary to use new bit planar modes, like on the Amiga. Unfortunately, with only 128KB of RAM, this is really memory inefficient, and with an 8-bit CPU at 3.5MHz, really slow. The focus on planar graphics, even though allowing 8 bit planes and hence 256 colour graphics, is in my view a serious design mistake. Having a super-character mode with 256-colour characters and allowing more than 256 characters is more memory efficient for games where there is a lot of repeated graphics, and also much faster to manipulate the display. More on that when I talk about my C65GS design.

**Q. I often felt that Commodore missed out by not manufacturing (apart from a few prototypes) the Commodore 65 I know I would have bought one.**

If Commodore had done a really good job on the C65, and got it out quickly, it might have had a chance. But by 1992 it really wasn't competitive, and the shockingly slow DOS and crippled bit-planar graphics and really ensured that it was never going to be a commercial success. Of course, for us enthusiasts, this doesn't matter so much, and just the thought of using what was possibly the most powerful 8-bit computer ever built has an intrinsic appeal.

**Q. You mention the Commodore 128 feels more hacked together, and looking at the history of the machine it would seem the designer wasn't told by the marketing department that it would support larger memory upgrades and also be backwards compatible with the c64, (another of Commodore's failing letting-marketing run the company). Do you think this is why the C128 felt like a hotchpotch of design implementations?**

My reading of the history suggests that, while I don't personally like the result, the C128 was pulled together insanely quickly, and managed to achieve some seemingly contradictory and almost impossible goals. The machine could run CPM on its Z80 processor. It was so amazingly close to 100% C64 compatible that the designers deserve some kind of medal. And it was developed from scratch in only about five months.

**Q. Can you explain to reader what FPGA is?**

The easiest way to think of an FPGA is as a programmable chip. In an FPGA you can create a real CPU and other digital circuits, and you can change them as often as you like. Basically it means that someone with a bit of skill, the right software and a few hundred dollars can design an interesting new computer, without needing a chip fab or millions of dollars.



**Q. Many will ask why implement the c64/65 in FPGA. Also, hasn't the c64 already been implemented in FPGA?**

For me, there are several reasons.

One of the biggest reasons is that the existing C64 implementations in FPGA are not free. That is, they are not released under open-source licences that let someone take and improve them.

For me, this is really important philosophically, and also for the longer-term survival of the C64 ecosystem. This is also why I have designed it to be able to work on an off-the-shelf FPGA board, so that as a community we don't need to rely on custom PCB runs and other barriers.

A second is that the existing C64 FPGA implementations are not really 100% compatible. I want to create a basis for making a 100% compatible reimplementation. The visual-6502 project is a key part in this, because they will soon have a gate level reverse-engineered design for the VIC-II and SID chips, which will make it much easier to make 100% compatible implementations in FPGA. People hoping to see such should consider donating to that project.

**Q. Is this just a collage project or do you plan a physical machine at the end of the design?**

I have a few end points in mind.

One, I intend to make a computer that lives in a C64 case that is a fast, modern 8-bit computer, and that is as compatible with the C64 as possible.

Two, I would like to have a laptop form-factor 8-bit computer that I can take with me when I travel for work, and generally enjoy using in a productive manner. Writing notes and simple word processing is quite feasible on a 32MHz 8-bit computer with SD-card storage and ethernet/Wi-Fi. While other laptops have to worry about suspend and hibernate, and long boot times, with an 8-bit laptop it will be possible to cold boot in <3 seconds, and potentially <0.3 seconds if I can make the SD-card reset happen faster.

Three, for teaching computer science, a modern 8-bit computer is I think a

good way to quickly and interestingly teach the fundamentals of computers.

Four, the experience of refreshing my FPGA programming skills will be helpful in designing a completely open-source mobile phone that uses just an FPGA and radio front-end, and has no closed-source binary blobs. This will be of great use in my humanitarian telecommunications work, because it will allow me to make mobile phones that work like digital CB-radios, communicating with each other directly, as well as with phone towers.

**Q. Many will question "where do you start" and what tools / software and hardware would you need to program such a project.**

All I needed was a \$200 FPGA development board and the free license for the proprietary VHDL programming and synthesis software. The rest is knowledge and a lot of patience, as VHDL is not a nice language, and the tools have many bugs and short-comings.

**Q. You plan to implement new features for the machine. Do you think this will alienate your creation from the purists of the machines (although to be fair if they were that bothered they would just use physical machines)?**

Purists will, as you note, not see what I am making as being "proper", and less so because I am extending and changing the specification in some ways. Of course, for the C65 there is no solid uniform specification, and the various prototypes differ in various ways. But for me, this isn't my concern. I have a real C64 for when I want to use it, but when I want updated 8-bit fun, I will be very happy using what is undeniably an 8-bit computer (there is no other CPU in there anywhere). They are different and complementary. I have already had fun writing 4,096 colour raster bars and playing with stretching pixels using the hardware scaler in the C65GS's VIC-IV. For me, the important thing is that it is still purely 8-bit, and feels like an 8-bit computer. So all IO is direct mapped, POKE is a skeleton key that lets you do anything, and "booting" is mostly a euphemism for letting the ROM clear the screen and say "READY."

**Q. At some point the old 8-bit hardware will fail and become impossible to repair. Could you see this happening in your lifetime, and do you think an FPGA should we say "clone" or implementations is the best way to preserve the machine historically?**

Spare parts, especially SID chips, are already getting very hard to source. FPGA replacement of ICs is a good backup plan. However, I have been doing a bit of digging around, and if we can raise the necessary funds, there is a chip fab in a Canadian University that uses the right kind of process that would allow us to recreate real SID chips, real VIC-II chips and all the other custom chips. As a University fab, the costs are surprisingly low, especially if it is connected in and becomes part of their VLSI teaching. If this can succeed, then it will be the golden standard for preserving old computers, but as I say, FPGAs are a good backup plan, and have the advantage that we have them now.

**Q. talking of cloning, is that how you see the designs in FPGA? Is it cloning of the machine?**

Well, I hope to provide as close to 100% compatibility for C64 mode as I can. This might mean having a 4510 and 6510 CPU and a VIC-IV and a VIC-II in the design to achieve this. However, I don't intend to exactly clone the C65. It was never finished, and so I am re-imagining what it could have been, and giving it some capability and performance boosts along the way, so that the end result, while very C65-like, will be a unique machine, and I think a very fun machine. That's really the goal, that this be a machine that is similar enough to be nostalgic, and give an authentic experience of a real C65, but with some niceties (like SD storage instead of actual floppy disks) and extras that can be accessed (like extra RAM, faster CPU, ethernet, more colours, higher resolutions and so on) if desired. But when it turns on, you will be thinking, "wow, I get to play with a C65".

**Q. Someone once said to me the holy grail of cloning (or implementations) would be to create a SCPU in FPGA. I see one of your goals is to provide a machine faster than the**

## SCPU, so will SCPU "compatibility" be programmed into the design?

Well, that is the holy grail for some at the moment, and I am sure will continue to be for a number of people. However, the C65GS is already faster than the SCPU, especially for video-intensive tasks. The only real advantage of the SCPU right now, is that GEOS works on it. It won't be hard to add C65GS support to GEOS, and once that is done, there will be only a very few SCPU specific programmes that people will miss out on. So my personal view is that SCPU compatibility will not be high on most people's agenda if they had a C65GS.

Talking about GEOS, I am actually very looking forward to seeing it ported to the C65GS. Pretty much all the functionality required is already there for it to work, give or take sprites which are one of the next things I wish to implement. The VIC-IV video modes are very like the VIC-II bitmap modes, but with resolutions up to 1920x1200 and separate 256 colour palettes for bitmap and sprites. If someone would like to take a shot at porting GEOS to the C65GS, I'd be happy to provide the necessary technical information and assistance.

## Q. Some feel all this tinkering isn't the Commodore way, and the machine isn't in fact a Commodore 64 /or 65 because its doing things that were not implemented in the machines original design, or that were impossible to implement. Would you like to comment to those readers?

I have probably mostly answered this in my response to one of the questions above. But I would say that this project isn't about recreating the machine in a pure sense -- for that we need chip fabs. It is really about making a new C64-like and C65-like (and hopefully compatible) 8-bit machine that is fun to use in the 20th century. For me there is also a sense of the less-is-more philosophy, in that modern computers are so complex that they have lost a lot of the charm that we enjoyed in the 1980s. Even my phone takes more than a minute to boot these days. Rather than just yearn for the simple old days, I am trying to recreate some of the the nice parts of them, but

re-imagined through the hardware that is available today.

Of course my secret long-term plan is to implement a complete 8-bit computer using a GaAs chip fab. Modern GaAs digital circuit processes are currently at about 1 micron, the same size as was used for most of the custom chips in the C64. The big difference is that for the same power consumption as the original chips it is possible to clock a design at somewhere around 10GHz. Now that would be a fast 8-bit computer :)

## Q. Personally I think this sort of implementation or "machine cloning" leads to a better understanding of the machine and sees demos and programs that would not have been realised if the coder hadn't been, should we say, "tinkering with the dark side" of FPGA implementations.

I think you are right in that; the more we understand the hardware, the more we can stretch it to its limits. This is part of why I am really looking forward to a complete and 100% accurate reverse engineering of the VIC-II.

## Q. On the Commodore 64 and 65 what component or software on each machine would you change and why (assuming you could do this when the machine was created)?

I think the C64 is best left as it is, because it is so iconic. It would be like me trying to say what should have been different in the original VW beetle.

But with the C65, I think it is easier to make some comments.

On the hardware side, planar graphics modes should have been ignored, in favour of better bitmap and character modes so that interesting graphics and animation could be done in 128KB RAM, and with a 3.5MHz CPU. Also, the 4510 should have included a 6502 compatibility mode that kept in the dummy writes. This would have been fairly easy to implement. One could also argue that it should have come with more than 128KB RAM, but that was a practical limitation of the time. Otherwise, I think some more advanced sprites would have been a nice touch.

On the software side, Commodore should have put some extra effort in making the DOS more efficient. The internal drive loads at only around 1KB/second, because it switches memory maps for every byte read. They could have at least made LOAD and SAVE copy a sector at a time, and so run 10x to 30x faster. They may well have planned to do so, but the ROM was never finished. This is an optional change to the ROM that I may well implement at some point. There is space in the modified C64 kernel to do such a thing, but it isn't too high on my list because when the C65GS is running at 32MHz it loads at around 8KB/second which is pretty comfortable. Any programme wanting to load faster can just make a fast loader that prods the emulated floppy controller or even the SD card controller directly. Already the kickstart ROM that loads the main 128KB C65 ROM is able to load at several hundred kilobytes per second on a decent SD card.

## Q. On the 128, if you were designing the original machine without any constraints what would you have done differently given the limitations of the day?

Ok, so If we assume we had to work with the technology of 1985, but had plenty of time and money, then I would have used the fastest 6502 compatible CPU of the time, probably 8MHz or 16MHz, and made a new VIC-II that did 40 and 80 column output to either a composite or RGBI video output. Also I would have made C128 mode re-accessible via secret knock register like on the C65. The memory management would also be a lot different!

## Q. Finally I would like to thank you for your time and do you have a comment you would like to add?

You're welcome. The only comment I have is to emphasise that this is an open-source project, and so other people are more than welcome to contribute with VHDL programming, testing, documentation or any other activity that interests them. My hope is that others can derive some fun from what I have begun to create.

Paul.





# COMMODORE FREE INTERVIEW WITH MELON64 SYSADMIN

Some people were worried Lemon64 would be no more after attempting for days to access the forum and with no success, then suddenly another forum formed up called Melon64. Soon enough though the Lemon64 forum was back, but does this mean we have to choose our forum or can both live in harmony? WE asked Melons sysadmin to shed some light on the new forum. Although I did obtain an interview the MELON Admin at this time wished to remain anonymous, or as they say in business speak "remain under the radar" so blindfolded and lead into a small shed, I sat down to speak to the admin and see if any information could be obtained, although the results are classified I printed them anyway!

**Q. Can you tell us something of your background and retro love?**

I've always enjoyed anything retro, but the Commodore 64 has always been my passion.

**Q. So Melon64. Was this just a fun prank because Lemons forum was down, or had you fully intended starting a separate forum, and with lemons system down again decided now was the right time?**

I had absolutely no notion of starting a forum. I've lurked on Lemon for a few years and when it recently went down I considered putting a temporary forum up, bearing in mind this was the third time in as many months it had completely disappeared. There wasn't much information around regarding why it was down, when it would be back, if it would be back. As time went on, into the 2nd week of downtime I decided I would take the plunge and put a forum online. Melon64 was born!

**Q. Where did the name Melon come from? And yes, I can see it looks very like Lemon 64.**

The Melon64 name did, obviously come from a play on Lemon64.

**Q. How many admins monitor the forum?**

At the moment there is only myself administering the forum. The forum is nowhere near large enough to justify taking up anyone else's time. When the Melon64's user base gets to the size where I can't take care of any issues, then moderators will be sought.

**Q. Some of the users on Lemon are quite, let's say aggressive! Was this an incentive to create a less hostile environment for users?**

Most forums have their share of overly passionate users. Aggression however won't be accepted on Melon64 and fortunately there have been no incidents to date. I'm sure in the future someone may need their wrists slapped.

**Q. What sort of take up have you had on the forum, how many users and for how long has it now been running**

A, Melon64 hasn't been around for three months yet. We've a small user

base, just over 160 users at present. I'm enjoying watching the user base grow, albeit slowly but the users that are taking the time to register do seem to be contributing. We have quite a few users that are very knowledgeable, and are more than willing to share that knowledge.

**Q. Do you fully intend the forum to continue?**

The forum will continue for as long as users feel it's a useful resource.

**Q. You must have had some backlash from the Lemon users. Would you like to comment on this?**

I honestly think that most Lemon users aren't aware of Melon64. There are some familiar users though, everyone's welcome!

**Q So you were expecting Lemon not to come back, it's had issues before, I think this time they say they**



### moved servers. How confident are you that your forum can survive with Lemon's resurrection?

Three months in and we're still growing. There definitely needs to be more than one source for our retro chit chat.

### Q. With a new forum you don't have the baggage of the old system. How did you start the forum? I noticed some place holder texts in there to keep things neat and tidy.

Everything is maintained by myself. Users are providing constructive feedback on a regular basis which I've taken on board and have resulted in various changes to the forum.

### Q. Was the initial skeletal format easy to set up? Was it custom-programmed or is it "out of a box"?

Melon64 is ran on PHPBB which is generic open-source forum software, running on Linux. I've added quite a lot of modules that modify how the forum operates; hopefully these don't detract from the user experience.

### Q. What's still to implement? Can you give our readers any exclusive gossip?

Well, I'd really like to implement a game database, which users will be capable of modifying. I'd also like to put together a decent Wiki or Knowledge

Base.

### Q. Can anyone join Melon64? If so, how do you join the forum?

Everyone is welcome to join Melon64. Head on over to [www.melon64.com](http://www.melon64.com), register, and confirm your account via the email you'll receive. That's it! Once registered, you'll be able to contribute to the discussions.

### Q. Is the forum exclusively Commodore 64, or can other machine users join in the debates, or do you intend to start forums for other machines, Commodore and non-Commodore systems?

Currently Melon64 provided forums for various other Commodore computers. There is a dedicated Amiga forum. I'm more than willing to add other Commodore-related forums upon request.

### Q. What are the main forum rules and how do you treat bad behaviours? Also, if you ban someone – how can you stop them coming back with another username ?

This is something I haven't set in stone just yet. At the moment everyone is well-behaved! The rules will be the usual generic rules you'd expect to agree to on any forum out there. The only rules in place at the moment are the obvious ones. No baiting, trolling, racism etc.

### Q. Are you a Commodore Free reader?

YES I am a Commodore Free reader, of course! I actually print it out and read it! The magazine is excellent, keep up the good work!

### Q. Finally, do you have another comments, and do you have a closing comment to the commodore community?

If you haven't already registered, please do! The forum is there for everyone who has an interest in the Commodore 64 and other Commodore computers. Thanks for giving me this opportunity Nigel, I appreciate it.

Cheers,  
Melon

### Hmm... I was going to say something about leaving your melons out but I don't think I will now.

Anyway, with the interview over I was escorted out into the open space. The blindfold was removed and I was alone, although thankfully – I found myself fully-clothed. I only had my melon in my hand, and after examining it, I set off home, 3 days later tired, and melon-less, I arrived, and wrote down my memories. This information remains classified.


Commodore C64 forums

Board index

[FAQ](#) [Register](#) [Login](#)

It is currently Thu Aug 07, 2014 12:09 pm  
 Page 1 of 42 • [1](#) [2](#) [3](#) [4](#) [5](#) ... [42](#)

RECENT TOPICS	REPLIES	VIEWS	LAST POST
 <b>beginners text adventure problem</b> by drpump » Mon Aug 04, 2014 6:02 pm in <a href="#">10 PRINT "HI!"</a> » <a href="#">BASIC</a>	11	106	by Shaun_B  Thu Aug 07, 2014 7:29 am
 <b>gosub error</b> by drpump » Wed Aug 06, 2014 8:49 am in <a href="#">10 PRINT "HI!"</a> » <a href="#">BASIC</a>	2	28	by drpump  Wed Aug 06, 2014 9:41 pm
 <b>What Mike said...</b> by Shaun_B » Tue Aug 05, 2014 7:57 pm in <a href="#">Chit Chat</a> » <a href="#">Chit Chat</a>	4	43	by dudz  Wed Aug 06, 2014 3:57 pm
 <b>Banana Pi</b> by Melon » Tue Apr 22, 2014 9:09 pm in <a href="#">Chit Chat</a> » <a href="#">Off Topic</a>	6	188	by judyhuang  Tue Aug 05, 2014 5:07 am
 <b>Relaunch64 3.2 - your IDE for Windows, Linux and OS X</b> by Testicle » Sun Jun 22, 2014 8:34 am in <a href="#">Chit Chat</a> » <a href="#">Software</a>	1	313	by Testicle  Fri Aug 01, 2014 8:01 am

# REVIEW OF THE FUTURE WAS 8 BIT SD2IEC DRIVE

Now, in the modern Commodore age you have a few options to load applications and games, one being to try and find an original 1541 Disc drive. This will probably come from Ebay; after the seller has fleeced you for around £100, and the post man has put his back out delivering it. You may find it works for a while, but where will you buy the disks from? Who will repair it once it breaks? Hence, the need for a more suitable storage option. SD cards are widely available, and it makes sense to use this option as a storage medium. They are hard-wearing, easy to buy, and available from everywhere. While the concept of using an SD card isn't new, we have had SD2IEC devices for a number of years in the Commodore world from a number of different retailers. The device is usable on virtually all 8-bit Commodore machines. For this article I am concentrating on just one device.

So simply, this device permits your Commodore 64 or other 8 bit Commodore to use an SD card as a disk drive, This particular one has a standard IEC connector for the disk drive port, and this also has a connector that just plugs neatly into the tape port on your machine for its power. The device is sadly not a full implementation of the 1541 disk drive, nor is it 100% compatible, but most games you will be putting on the device will be Fixed or cracked version. Of course, you will be unable to copy the "real disk" to the device. If you own the original you will be safe to use a copy from the internet. I wouldn't like to condone any sort of copy theft, even if the files are over 30 years old! I leave this to your conscience.

The beauty of this particular system is its size, just a little longer than an SD card itself. It goes without saying that

owning one of these will change your computing experience with your Commodore forever. It will of course stop any sort of loading problems from magnetic media as the device and files are digital and it stops that dancing disk drive effect as your 1541 fights with the copy protections thrashing the disk heads to load the game.

The Future was 8-bit - has a huge collection of games guaranteed to be compatible with the device available to download from their website. You can download the files quickly as a large ZIP image but you must then decompress this onto the SD card. (Hey remember you need to do some work yourself!) The website also contains a wealth of useful information to help you get the SD2IEC up and running so it's a first stop when you receive your device.

The SD2IEC is still as slow as a 1541 drive. But you can however speed this up, as the device is fully compatible with the JiffyDOS fast loader ROM, along with a number of cartridge-based

fast loader solutions again you can find more details on website about these and the compatibility may change with upgrades to the devices firmware.

To load the File Browser software for example you'd type 'LOAD "FB",8,1'. (Because the file is called FB) Once loaded, simply type type 'RUN' to start the File Browser interface. Once running you can pick your game. If you have a game that comes on more than one disk, you can use the disk swap switch. This automatically loads in the second disk with just a button press on the device to "SWAP" disks. All of this functionality is configured by a special file in the game's directory. Full Details of how to create this and what it needs to contain are available on **The Future was 8-bit** website .

The device can be bought either cased or uncased. The cased version will be the most popular; however, those wishing to attack there machines with a soldering iron can install the device inside their C64 or build their own case and purchase the cheaper uncased version,



but this will need modification to add a swap button.

The cased version in use has two LEDs to indicate the devices status. A green LED shows disk activity so as long as this is lit, you can assume a program is loading. The second red LED flashes whenever there's a disk load error. This could happen if there's some incompatibility between the disk image you've chosen and the SD2IEC's emulated 1541. If this happens then you will need to find a version that will work with the device.

Along with the device is a useful "getting started guide" I suppose most people will just use the device to load a program or run something from a d64 image.

The build quality is fantastic, and the device does look like a miniature 1541 disk drive, the cables look professionally terminated, and the price is justifiable for what the device can do and the work that's gone into manufacturing it. If this review has teased you enough you should visit the website and buy

the product, for a few pounds you could change your commodore experience forever!

<http://www.sd2iec.co.uk/>



*...The Future was 8bit ...*

Home My Collection At last.. The C64p SD2IEC SD2IEC Info Projects/Prototypes Repair/Upgrade Shop Forums Links Old Stuff Contact Guestbook

**\*\*\* LIMITED edition GENUINE recycled C64 plastic SD2IEC Case now in stock \*\*\***  
**Updated SD2IEC case design now available in SIX COLOURS!**  
**Turbo Loader Cartridge for C64/SX64/C128 - Epyx Fastload RELOADED now available**  
**\*\*\* Commodore 1520/Atari 1020/Tandy CGP-115/Oric PCP40/Sharp CE-150 pens in stock \*\*\***  
**\*\*\* C64p what dreams are made of, Commodore 64 DTV based laptop.. Userguide \*\*\***

**Wow, you've ended up on my landing page. Well done you!**

As I spend far too much of my time saving old computer tat from landfill, it was inevitable that I'd throw together some sort of website to show all 3 of you what I do.  
 You just can't beat the old 8bit machines... Keeping them alive can be as addictive as keeping vintage cars on the road, but with less rust, unless its a Commodore Pet.  
 If you've got an old sick computer that you'd like to reseraect, drop me a line!  
 You'll also notice, that I have a slight leaning towards Commodore machines.. But I still find time to fiddle with all maner of other makes and models of vintage computers.

**Back Story... Shortly after the earth cooled...**

My first computer was a Vic-20, second was an Amstrad PCW8256 (took it back a week later tho, third was a C128 (yay!)). The list went on and on.. Till I finally switched from Amiga 4000 to PC permanently..ish  
 Thanks to being a bit mental, and a thing called ebay.. I have managed to fill my house with all kinds of vintage computer greatness..

***I also offer a repair service for Commodore Vic-20 C64 C128 C16 Plus4 and Amiga, as well as Sinclair, Amstrad, Oric, BBC, Atari and Dragon computers.. Also free advice for those who want to fix it themselves! Check out the Repair pages as well as the forums.***



## The Assembly Line

### *\$05: Look at Us, All PRIMM and Proper!*

By Bert Novilla (satpro)

The last time out I did something I said I wouldn't do: bog the reader down. I apologize for that, and as soon as I can figure a way to write a good sequel I promise we will continue on the subject from Assembly Line #04. This time however, we are going to explore and explain a related task: printing in-line text strings

#### What is PRIMM?

Many of us have seen or heard of the C-128 text-printing function called "PRIMM." It has been used in many applications, and quite prominently by most of the major turbo utility cartridges. The Buddy assembler (and accompanying text editor) is another good example. What is PRIMM, or "print immediate?" In short, it is a function which allows the 65x programmer to print an inline, 0-terminated petscii text string. That is, placing the string characters (and optionally, screen location coordinates) directly after the call to print our string. Normally we tend to place our code in one location and our data in another, separate space. This can make

for cleaner, more understandable code, but other than for aesthetics or reasons related to personal preference, there is really no rule which prevents us from placing code with (and around) data, or from printing an in-line string. The C-64 does not have a function for printing in-line strings, but we can borrow from big brother C-128. So let's see how PRIMM works -- and let's improve upon it some, and even see how we can add a screen plotting enhancement. The code which follows makes a very nice addition to the programmer's toolbox.

#### The Code and What it Does.

The call to PRIMM is actually quite simple. It takes no parameters, preserves the registers, and is very economical with memory usage. You structure the call to PRIMM like this:

```
JSR PRIMM          ;a call to our in-line string printing function
.text "Hello World.", 0 ;the petscii string with a terminating 0

... more code follows ;this could be anything

RTS                ;exit the program
```

Instead of returning to execute the next instruction PRIMM instead does some parsing to retrieve and print the in-line text string. Once PRIMM encounters the zero byte (0) terminator, it returns to the code stream. This is really very straight-forward, but it takes a moderate understanding of both the stack and the program counter. Okay... let's take a first look at the C-128 version (on the left) and our modified version (to the right).

C-128 Version	Our Version
PRIMM =*	PRIMM2 =*
PHA ; save .A on the stack	PHA ; save .A on the stack
TYA	TXA
PHA ; save .Y on the stack	PHA ; save .X on the stack
TYA	TYA
PHA ; save .X on the stack	PHA ; save .Y on the stack

This first section (above, left) preserves the 65x registers A, Y, and X. If, as a conscientious programmer, you always preserve the registers your function modifies, you increase greatly the chances your code will work later for both you and anyone who wishes to use your function in their own application. Why? Well, for example, what if we wanted our PRIMM function nested within a larger loop, and the loop is counted by the .X or .Y register? In lieu of preserving registers we would be required to save the loop index register before making the call to PRIMM, and then we would have to restore it later, wouldn't we? It would place an undue burden on the person making use of such a function, and it's just one more needless thing he would have to remember.

It has long been considered good practice to save any registers which are modified in a function, especially if someone else will use that function in their own program. We are going to modify all three registers in our function, so first let's just save them on the stack and be done with it. Our initial modification won't make or break us, but in the name of consistency let's change the order in which we push our registers to the more standard (and natural) ordering; A, then X, and then Y. Sorry, I'm just funny about things like proper ordering of stack pushes, primarily because it can foul you up on the other end of this function when it is time to pull and restore these registers. Kernal interrupts save registers in A/X/Y order, as do most competent 65x assembly language programmers. It doesn't really matter to the computer which order you push the registers, but we should always strive for consistency in assembly language.

### Adjust PC to Point to Our Text String

Next, we have to calculate where the string is in memory, which (we know) is directly after JSR PRIMM. How do we do this? Well, we can get the address by looking on the stack. When you perform a JSR the processor needs to save a return address so that it will know where to continue executing instructions after the function terminates. The 65x family of processors place the address of the *last* byte of the JSR \$XXXX instruction, which in this case is one byte *before* our string. Later, when the processor encounters the RTS instruction it will automatically increment (add one) to the address it retrieves from the stack, thereby pointing to the next instruction in the code stream. The C-128 method is to increment .X four times, store the address in a zero page pointer, and then increment the address pointer. The four INX instructions are necessary to skip past the three register pushes (A, X, and Y) and to also account for the fact that

the 6502 SP always points to the next *lower* address on the stack a future pushed byte will be placed. The last byte that was placed on the stack is accessed as SP+1, so the C-128 method must include that extra (4<sup>th</sup>) register increment.

So now we have the location of the return address, *minus* 1. The C-128 method then places this address into the zero page pointer at \$BC-\$BD, then does a double-byte increment of the address to make it point to the first byte of our string. We don't need all that, however. We will just start our Y-index at 1, which accounts for the return address (on the stack) initially pointing to one byte *before* our string.

Our improvement (as seen below) is to eliminate the four INX instructions and simply hard-code the absolute indexed instruction directly → \$0104,x/\$0105,x as opposed to \$0100,x/INX/\$0100,x. You can see the significant memory (and time) savings in the listing. Each INX, by the way, requires two cycles of your processor's time, so let's save cycles where we can. Incidentally, I often see guys jump through hoops trying to cut a cycle or a byte in places where it's not really necessary. Don't let cycle counting get in the way of a good program. It's often not worth the effort and can become all-consuming and hard to debug, although there are times you may have to in order to make something work right. Good examples where "hoop-jumping" might be necessary are in serial bus timing and of course, chasing the raster in a complicated demo.

One other thing I almost always do is use labels instead of hard-coding computer addresses such as \$0104, \$BC, etc. Why? Because, for example, you may later find that the addresses you choose in zero page (\$BC-\$BD here) are in conflict with the Kernal or BASIC and you have to change locations for the pointer, which may now be used in many places scattered throughout your program. Using labels means you have to make the change in *exactly* and *only* one place – the label at the top of your source. This is a fundamental principle in symbolic assembly language. Using labels liberally is much more elegant and requires much less work in the long run, and is certainly much more descriptive. Remember this tip – it will save you countless debugging (and search/replace) sessions later.

<pre> TSX          ;get the stack pointer (SP) INX          ;increment the index INX          ;four times to account for INX          ;pushed A/X/Y INX          ;and 6502 stack return address peculiarity LDA \$0100,X ;use absolute indexed addressing STA \$BC     ;to retrieve the return address-1 (lo) INX         ;increment index LDA \$0100,X ;retrieve return address-1 (hi) STA \$BD     ;store into zp pointer (hi) INC \$BC     ;increment pointer at \$BC BNE PRIM1  ;branch ahead unless lo byte wraps to \$00 INC \$BD     ;increment hi byte (\$BD) if needed </pre>	<pre> RETADDR     = \$0104 ;label referring to stack address PTR_STRING  = \$BC   ;label referring to zp address CHROUT      = \$FFD2 ;well-known Kernal PRINT function PLOT        = \$FFF0 ;Kernal PLOT function  TSX          ;get the stack pointer (SP) LDA RETADDR,X ;aka LDA \$0104,X (return address-1 lo) STA PTR_STRING ;store into zp pointer (lo) LDA RETADDR+1,X ;aka LDA \$0105,X (return address-1 hi) STA PTR_STRING+1 ;store into zp pointer (hi) </pre>
--	---

Now, with our setup out of the way, it's time to print the string. We are going to use the Kernal function CHROUT located at \$FFD2. CHROUT prints the petSCII character in the A Register, for those few that are not familiar. We are going to use **Indirect Indexed, Y** addressing for this, and the Y Register is going to do double-duty as a length counter for our string. We will need the length in order to correct the

program counter (PC) when it comes time to return to the main program. We will not modify this section of code as it is fine as is. The posted assembly listing is what might be considered a "generic" listing, so in place of local (cheap) labels (look up usage in your assembler's manual) I have used "=" to denote label locations. This will work in nearly all 6502 assemblers.

<pre> PRIM1 =* LDY #\$01      ;string index (and string length counter)  PRIM2 =*      ;top of the loop LDA (\$BC),Y  ;get byte (a character) from string BEQ PRIM3    ;exit when terminating 0-byte is reached  JSR \$FFD2    ;print the character in .A INY         ;increment index/length counter BNE PRIM2    ;branch back to top of loop (256 chars max!) </pre>	<pre> PRIM1 =*      ;label not needed for our version LDY #\$01      ;string index/string length counter  PRIM2 =*      ;top of the loop LDA (PTR_STRING),Y ;get byte (a character) from string BEQ PRIM3    ;exit when 0-byte is reached  JSR CHROUT    ;print the character in .A INY         ;increment index/length counter BNE PRIM2    ;branch to loop top (256 chars max!) </pre>
--	--

We have now displayed our text screen on the screen at the current cursor location.

Earlier I mentioned embedding screen location coordinates into the string. This is a very handy (and convenient) way of putting a string right where you want it without any additional code in the main program and is commonly known as

"PRINT AT" or something similar. I call mine "PlotString" and you may have your own name for yours. So, how is this done?

The idea is two make the first two bytes determine the position your string is displayed on-screen. We add the coordinates to the beginning of the string as follows

```
.byte X,Y           ;replace X (0-39),Y (0-24) with the actual coordinates you need
.text "Hello World.", 0 ;the petscii string with a terminating 0
```

That's it as far as the string is concerned, but we have to add a wee bit of code to our PRIMM function. The listing below shows the process. The odd thing about the Kernal function PLOT (\$FFF0) is that the X/Y coordinates are transposed, so we must get them from the string in reverse order to preserve the Y Index Register.

### Modification to Include Screen Coordinates

```
PRIM1 =*           ;our version doesn't need this label, it's here as a
placeholder
LDY #$02           ;string index/string length counter
LDA (PTR_STRING),Y ;get the Y-coordinate from coordinate position 1
TAX               ;PLOT requires Y-coordinate in .X
DEY              ;now .Y = 1
LDA (PTR_STRING),Y ;get the Y-coordinate from coordinate position 0
TAY              ;PLOT requires X-coordinate in .Y
CLC              ;PLOT requires carry = 0 to SET cursor, 1 to GET
cursor
JSR PLOT          ;aka $FFF0

LDY #3            ;first real text character is at string position 2
(INY:INY does the same thing)

PRIM2 =*         ;top of the loop
LDA (PTR_STRING),Y ;get byte (a character) from string
BEQ PRIM3       ;exit when 0-byte is reached

JSR CHROUT      ;print the character in .A
INY             ;increment index/length counter
BNE PRIM2       ;branch to loop top (256 chars max!)
```

So, you see that with just a bit of additional code we made ourselves a very nice string printing routine! The tricky part (not too bad, though) is possessing the knowledge that the Kernal PLOT function has mixed-up cursor coordinates with respect to registers X and Y, but we will assume that as good assembly language programmers you took the time to look up the function before using it, right?

#### Cleanup Time

The final thing we have to do is get the program counter "fixed up" so that we can return to the main program. Once again we will make a few changes to the C-128 method in order to save some space and time. If you choose not to add the PlotString mod you won't need to get the stack pointer

with TSX as we did not modify .X anywhere in the original PRIMM-only function.

We will take our length counter (in .Y), add it to the pointer located at \$BC-\$BD, and place it back on the stack where the original return address is located. This has the net effect of adding the number of string characters to the program counter – skipping the string (and the optional screen coordinates) altogether. And finally, make absolutely certain you pull the registers from the stack in the REVERSE order. Failure to do this could result in a hard-to-find bug later. Also, since this function modifies the stack to get at the program counter so you won't be able to JMP to it. Instead, remember to always JSR.



PRIM3 =* ;all done printing	PRIM3 =* ;all done printing
TYA ;put the character count in .A	TYA ;put the character count in .A
TSX ;get the stack pointer (SP)	TSX ;only necessary if PLOT mod is used
INX ;increment the index	CLC
INX ;four times to account for	ADC PTR_STRING ;add count to string location (-1)
INX ;pushed A/X/Y	STA RETADDR,X ;and store it where return address is
INX ;and 6502 stack return address peculiarity	LDA #\$00 ;LDA #\$00 accounts for word-sized addr
CLC	ADC PTR_STRING+1 ;do the hi byte
ADC \$BC ;add count to string location (-1)	STA RETADDR+1,X ;and store it
STA \$0100,X ;and store it where return address is	PLA ;all done, so pull and restore
LDA #\$00 ;LDA #\$00 accounts for word-sized addr	TAY ;(.Y) the modified registers
ADC \$BD ;do the hi byte	PLA ;in the REVERSE order
INX	TAX ;(.X)
STA \$0100,X ;and store it	PLA ;(.A)
PLA ;pull the registers in REVERSE order	RTS ;cpu will add a byte with RTS
TAX	
PLA	
TAY	
PLA	
RTS ;and return to the main program	

And that's that.

#### Bonus: 65816 Version

Look below to see how we might do the same thing on the SuperCPU. Notice how much shorter and less complicated the function becomes. I have included a

few directives and instructions to pre-set the environment for native, 16-bit mode, although later we will make some register size adjustments in order to accommodate 8-bit Kernal requirements. Although not included in this sample, it is very easy to extend this function to accept strings of any length up to 65536 bytes.

```

.65816
.al (longa)
.xl (longi)

ptr_string    = $bc
PLOT          = $fff0
CHROUT        = $ffd2

;-----
;SETUP                ;this can be done earlier
;-----
    clc
    xce                ;go to 65816 native mode
    rep #00110000     ;go to 16-bit A/XY
    jsr PlotString
.byte X,Y            ;replace X (0-39),Y (0-24) with the actual screen
coordinates you need
.text "Hello World"
.byte 0

    rts

```

```

PlotString =*
    pha                ;save .A on the stack
    phx                ;save .X on the stack
    phy                ;save .Y on the stack

    lda 1,s            ;get the return address (stack relative
addressing)
    sta ptr_string     ;store into DP (Direct Page) pointer

    sep #00010000     ;go to 8-bit XY
    lda (ptr_string)   ;get x/y-coordinates both at once (DP Indirect
addressing)
    tay                ;Kernal PLOT requires x-coordinate in .Y
    xba                ;swap A/B registers
    tax                ;Kernal PLOT requires y-coordinate in .X

    clc                ;Kernal PLOT requires carry = 0 to SET cursor
    jsr PLOT           ;aka $FFF0
    sep #00100000     ;go to 8-bit A

    ldy #2             ;string length counter - first real text
character is at string position 2
prim2 =*              ;top of the loop
    lda (ptr_string),y ;get byte (a character) from string
    beq prim3          ;exit when 0-byte is reached

    jsr CHROUT         ;print the character in .A

```

## In Conclusion

While thinking of ways to present today's topic I consulted several different sources to see how others have approached this in the past, primarily because my own projects have managed (over time) to take a very straightforward function such as PRIMM and turn it into something very different from the code you see here, and it has all been based on my own needs in different situations. You will find the same will happen to you as you find new ways to expand and enhance your own efforts. These days I work almost exclusively on the SuperCPU, and with that comes liberties (primarily in the form of a greatly enhanced instruction set) which tend to spoil the assembly language programmer. So something like this became somewhat of a challenge

as I had to think only in terms of 8-bit 6502. I was amazed to see the number of ways this function has been written – some good, some not as good. In the end I had to dig deep into the bowels of an old hard drive to find an untarnished source copy from a long time ago, one which I wasn't sure would even work as written! I had to type the whole thing into CBM prg Studio to make sure it would do what it is supposed to do. To my surprise, the only register you can push onto the stack is the Accumulator! Well, okay, it wasn't quite *that* bad, but it was sort of fun to work in 8-bits again.

I hope you can make use of this technique – it's a fast (and efficient) way to get a screen up and running. While nothing earth-shaking, PRIMM does demonstrate how to access the stack and

manipulate the program counter. Also, I included the bonus 65816 code for those who are interested in programming the SuperCPU. The VICE SuperCPU emulator is impressive and a lot of fun – give it a try! Arthur Jordison's **CBM prg Studio** now includes a 65816 assembler which makes writing programs for the SuperCPU a breeze. We have an entire section devoted to SuperCPU coding over at [www.Melon64.com](http://www.Melon64.com) – and all are welcome. The admin allows uploads, so you are free to exhibit your code!

See you next month...

Please send errors, omissions, or suggestions to [bert@winc64.com](mailto:bert@winc64.com) or on [Lemon64](#), username [satpro](#) or at [www.melon64.com](http://www.melon64.com), username [satpro](#).



# Commodore Free Magazine

[www.commodorefree.com](http://www.commodorefree.com)

**Editor**  
Nigel Parker

**Spell Checking**  
Peter Badrick and Bert Novilla

**Text , HTML & Ebook Conversion**  
Paul Davis

**D64 Disk Image**  
Al Jackson

**ISSUU formatting**  
Alessandro Di Nepi

**PDF Design /Editor /webhost /text collector**  
Nigel Parker

**Website**  
[www.commodorefree.com](http://www.commodorefree.com)

**Email Address**  
[commodorefree@commodorefree.com](mailto:commodorefree@commodorefree.com)

#### **Submissions**

Articles are always wanted for the magazine. Contact us for details .We can't pay you for your efforts but you are safe in the knowledge that you have passed on details that will interest other Commodore enthusiasts.

#### **Notices**

All materials in this magazine are the property of Commodore Free unless otherwise stated. All copyrights, trademarks, trade names, internet domain names or other similar rights are acknowledged. No part of this magazine may be reproduced without permission.

The appearance of an advert in the magazine does not necessarily mean that the goods/services advertised are associated with or endorsed by Commodore Free Magazine.

#### **Copyright**

**Copyright © 2014 Commodore Free Magazine  
All Rights Reserved.**