

# COMMODORE FREE

Issue 59 2012

Free Commodore Computer Magazine

In TEXT, PDF, SEQ, D64 and HTML And Ebook formats

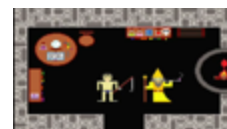
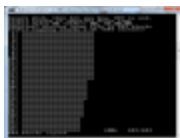
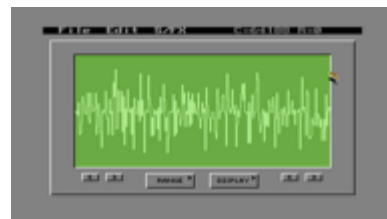
[www.commodorefree.com](http://www.commodorefree.com)



```
Administrator: C:\Windows\system32\cmd.exe - edit Scroly.asm
File Edit Search View Options Help
C:\NMCProgramming\C64Ass\dreamass\Scroly.asm
lda BUFFER ;this passes the bytes in the BUFFER and writes it to the
sta STRING,y ;last byte of the STRING data before the terminator (0)
ldy #000
tys
jsr DELAY
jmp LOOP
DELAY
pha ;This will slow down the text so that we can read it
lda $a2 ;Pushes 'A' onto the stack
sta $fd ;Loads A with the current value of memory location $00A2
;and stores it in memory location $00FD
DELAY1
sec ;SEts Carry flag (good practice when doing arithmetic)
lda $a2 ;Get's new value of $00A2 (used for the clock)
sbc $fd ;Subtracts with previous value stored above
cmp #$0f ;Compares result with 15 decimal
bcc DELAY1 ;Branches if Carry flag is clear back to DELAY1 marker
pla ;Restores old value of A from stack
rts ;Returns from Subroutine
BUFFER
.scr "
STRING ;One char buffer
;Data (40 characters long)
.scr "...micro mart available every thursday...",0
Fi=Help | Line:48 Col:50
```

# CONTENTS

<b>EDITORIAL</b>	<b>PAGE 2</b>
<b>READERS COMMENTS</b>	<b>PAGE 5</b>
<b>NEWS</b> Digimaster 128 released!	<b>PAGE 5</b>
<b>NEWS</b> OpenCBM Archiver Deeds of Yore Results of the Amiga Games Award 2011	<b>PAGE 7</b>
<b>NEWS</b> CBM-COMMAND UPDATED PixelJam Demoparty	<b>PAGE 8</b>
<b>NEWS</b> MORE GAMES THAT WERNT FOUND Angel of Hell II PenPalz	<b>PAGE 11</b>
<b>NEWS</b> A-EON Updates "Boing Ball" re-visited News from A-EON Technology Sam440ep_setup 1.3 Released	<b>PAGE 12</b>
<b>NEWS</b> PDXCUG.org Meeting Reminder C=FREE WELL I MISSED THE DEADLINE BUT..... iBatch V1.4: new Features/ Screenshots	<b>PAGE 13</b>
<b>NEWS</b> 14AmiDARK Engine Rel0.5 is Now Available!	<b>PAGE 14</b>
<b>NEWS</b> iBatch V1.4: new Features/ Screenshots EF3 USB Utilities v1.2	<b>PAGE 15</b>
<b>NEWS</b> FREESCAPE GAME NEW ISH	<b>PAGE 16</b>
<b>NEWS</b> AmigaOS 4.1 Update 5 CD for the AmigaONE X1000 "First Contact" system has gone GOLD!	<b>PAGE 17</b>
<b>S U D O K U</b> by John Borrowman	<b>PAGE 18</b>
<b>ASSEMBLER TUTORIAL PART 2</b> FROM MICROMART BY SHAUN BEBBINGTON	<b>PAGE 19</b>
<b>The Datasette</b> By Francisco Javier Gonzalez Cañete	<b>PAGE 25</b>
<b>Rediscovering Commodore CP/M</b> by: <i>Commodoreman</i> ©	<b>PAGE 31</b>



# EDITORIAL

The Second issue of 2012

and in this issue we have the continuation of the assembler tutorial for the Commodore 64 courtesy of Shaun Bebbington and Micro mart magazine, building on from what we learnt last issue. More news as usual for the magazine, and although I say it every time it still surprises me that for such a supposedly dead platform we have a large amount of news and projects going on.

Also we have a look at the datasette and how software was loaded from tape or more correctly the TAP format, for a number of years I only had a datasette to load games and programs from, although it could be tediously slow, (I remember some titles taking about 20 minutes to load if they decided they wanted to ) and very unreliable the process could be, the one thing about tape was the suspense of the loading especially when turbo loaders came in, with music and loading pictures. I know some of the disc games started using loading pictures but it didn't feel the same, and the horrendous knocking sound and watching my drive dance around trying to read the copy protection put me off discs. Some loading systems would even have a countdown timer until the game started, while other had mini games to play until the main game loaded. I still load from tape for this very reason, the waiting time (now down to minutes) still adds to the games appeal, and when the picture does finally come up and then the music starts; your literally trembling with anticipation. Of course should you only have a short time to look or play a game or use a utility then loading from some other system is more preferable, especially now with new hardware able to load applications very quickly indeed.

## MISSING

We have a reader's comment plea to find a missing book for the +4 computer about programming entitled "an introduction to basic part 2" of course if you can help please contact Commodore Free and I will pass on any information. I would

love to find this book or even notes about the book for the reader. Finding the author would of course be a major plus point.

## CP/M

We finish off this issue with a look into the world of CP/M and a users initial looks at finding resources and questions about the system and its use in the Commodore 128, If you feel you could expand this then let me know, I am sure many users are curious about CP/M

## Again I would like to say

I am planning a Commodore 264 edition; this will cover the commodore 116 commodore 16 commodore plus4 etc, If you would like to contribute to this issue, and so far I have only 2 people willing to help. Please contact me. This is now your chance to tell the world why this range of machines was so good and why you still use the machine today. It could be a memory about the machine; article or a technical article. Do you know why the range was started and why Commodore changed the design, creating the plus4 and c16 and do you know what system they were competing against (feel free to write to me at the usual address and let others know) Of course I have a good idea why the system was created by Commodore and its major competitor, but I feel this would be better written by an actual 264 fan rather than myself. Although I am a fan of the range I am no expert on the system.

Now is your chance to let the machines shine out and teach the masses!

Regards

Nigel Parker (pretend editor)

Commodore Free

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



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

Find a mouse for your modern or vintage computer  
\*Plus Commodore components and software

The Adventures of

# RAMSLAM

**NEW  
FOR C64!**

In **Dragonland**



- Password system
- Shoot power-up
  - Tube system
  - Keys
  - Teleporters
  - Trampolines
- Highscore saver
- 15 worlds with 3 levels each



**PROTOVISION**

More information at [www.protovision-online.com](http://www.protovision-online.com)

# Readers Comments

Has any-one ever seen, heard of or know the whereabouts of Prof. Andrew Colin?

You will of course be aware he wrote all the "Introduction to Basic" books for Commodore range of machines -I think all up to the Amiga. But certainly he created the series for the VIC, C64, & C16/+4s.

The thing is. Many of 'us' 16/+4 users are dying to know what-ever happened to part 2 of The Series of "An Introduction to BASIC..." for c16/+4, The burning question I keep asking is "Was part 2 ever made", does it exist, can we get a copy?

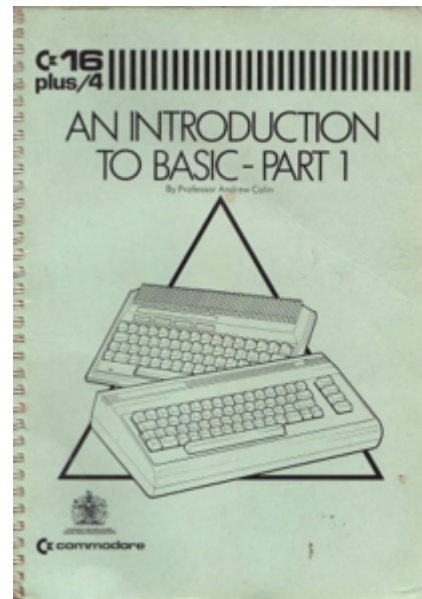
I have both the c64 books, haven't had chance to read them through yet though. I Never bothered with the VIC -bless it, perhaps I will someday in the none too distant future.

I think you're the last hope (Commodore Free) with above plea, Please can you ask the readers for help if you can.

I would love to read that book, and know how my main comp prog. teacher by books is?  
Best Wishes Jn.

## COMMODORE FREE

So you have it has anyone ever seen Part 2 of the book series "an introduction to Basic for the +4" does it exist in any format and is Prof. Andrew Colin still alive and well to ask?



# NEWS

## Digimaster 128 released!

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

From: Robert Bernardo

To: commodorefree@commodorefree.com

Subject: Digimaster 128 released!

Based on Autumn Technologies' Digimaster for the C64, Hydrophilic has released Digimaster 128 for the C128 in 40-column mode. Digimaster is the digital audio sampling and editing software for the C64 and now for the C128. With sampling rates up to 12,439 cycles, Digimaster has a graphical interface which makes cutting and pasting samples easy. Digimaster is meant to work with audio samplers, such as the Sound Ultimate Xpander 6400.

Digimaster 128 boasts improvements over its older C64 cousin. With the C128's greater memory, the program can now capture sound samples approximately twice as long as the C64 version. Now when the screen is blanked during highest-quality sound sampling, the program switches into the C128's 2 MHz. mode. Also when run, DM 128 auto-detects whether the Commodore is NTSC or PAL and then runs the program appropriately. DM 128 takes advantage of the C128's faster serial loading when loading samples.

To read more about Digimaster 128, go to

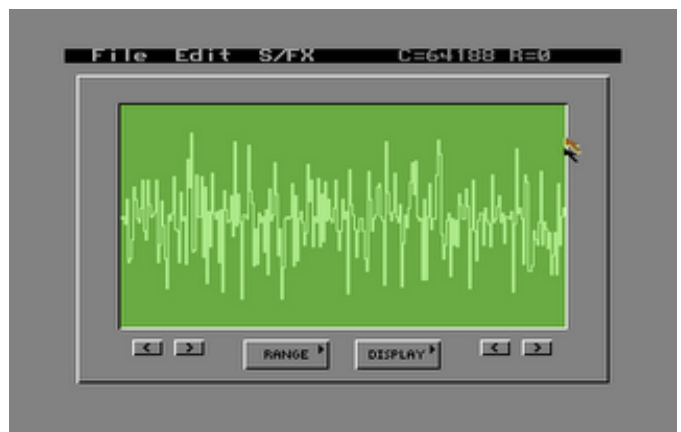
<http://www.commodore128.org/index.php?topic=4162.msg20510;topicseen#msg20510>

If you are registered at [www.commodore128.org](http://www.commodore128.org), you can see a screenshot .gif of Digimaster 128 and download a zip of the program and its development notes. Digimaster 128 is shareware; keep reading the above thread in order to find out how to PayPal money to Hydrophilic.

If you are not registered at [www.commodore128.org](http://www.commodore128.org), go to

<http://retro-link.blogspot.com/2012/02/digimaster-128-released.html>

in order to see a screenshot of the program. (Commodore Free see attached photo)



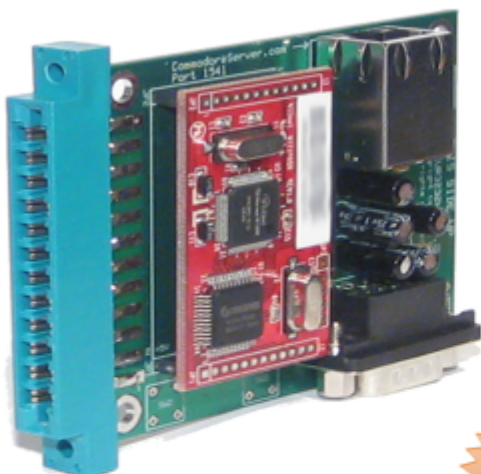
Truly,

Robert Bernardo

Fresno Commodore User Group

<http://videocam.net.au/fcug>

# COMMODORE INTERNET MODEMS ARE HERE



starting at  
**\$66<sup>99</sup>**

## AVAILABLE IN 2 OPTIONS:

- **INTERNET ONLY**  
Connect to Internet Services from a stock Commodore computer.
- **INTERNET +RS-232**  
Same as above, but includes a standard RS-232C port to connect to other RS-232 equipment

**USER PORT DEVICE  
COMMUNICATES OVER ETHERNET  
INSTEAD OF TELEPHONE LINE**

## EASY TO PROGRAM

Simply open an RS-232 file and then send data.  
For example:

```
10 OPEN 2,2,2,CHR$(10)
20 PRINT #2, "MY DATA"
30 CLOSE 2
```

## FIND OUT MORE AT:

<http://www.CommodoreServer.com>



# ARE YOU DISK ORGANIZED?

TAKE CONTROL OF YOUR DISK COLLECTION

## D64 Disk Management for your Commodore Computer

Never before has there been an Internet-based disk management system and software download center for Commodore computers!  
Access, upload and view disks right from your Commodore computer.

**UPLOAD YOUR DISKS TO:** <http://www.CommodoreServer.com>

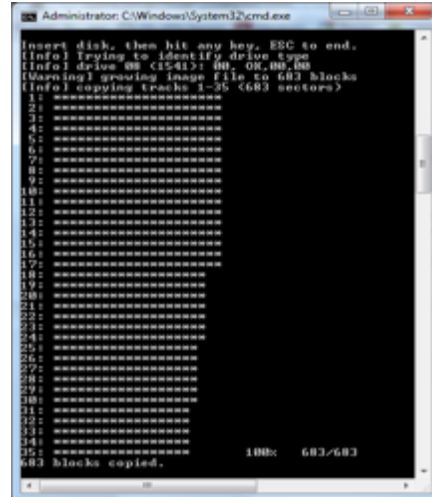
# NEWS

## OpenCBM Archiver

OpenCBM Archiver is a small program to help automate the creation of disk images using OpenCBM. With OpenCBM Archiver, you only need to press a key between disks and a new image will be created using the next number in a sequence.

You can download OpenCBM Archiver at:

<http://opencbmarchiver.codeplex.com>



## Deeds of Yore

### Ostrog Productions Presents...

The story of David Grigsby, a young man in 1987 who loves video games. One night, after losing his car, his job and his girlfriend, David gets sucked into a video game called... DEEDS OF YORE!

The first 2 episodes are live action featuring the C64, the rest seems to be Commodore style animations

<http://www.youtube.com/user/ostrogproductions>



## Results of the Amiga Games Award 2011

On February 4th 2011, the magazine Obligement have announced the results of the Amiga Games Award 2011. This represents the best Amiga games voted by amigans on all Amiga platforms.

### For AmigaOS 68k :

1. Mr Beanbag (Jigsaw Lounge)
2. Bermuda Syndrome (Century Interactive)
3. Barrage (Michael Speck)

### For AmigaOS 4.x

1. 1941 Extreme Deluxe Dual Players 2.0 (Badblocks, Hugues Nouvel)
2. Swamp Defense (EntwicklerX).
3. River Pirate (EntwicklerX).

### For MorphOS

1. Return To Castle Wolfenstein (ID Software, Gray Matter Interactive)
2. 1941 Extreme Deluxe Dual Players 2.0 (Badblocks, Hugues Nouvel)
3. Simutrans (Hansjoerg Malthaner, Simutrans Team)

### For AROS

1. UFO: Alien Invasion (UFO: Alien Invasion Team)
2. Assault Cube (Rabid Viper Productions)
3. Cube 2: Sauerbraten (Wouter Van Oortmerssen)

More information on :

<http://obligement.free.fr/hitparade/award2011.php>

# NEWS

## CBM-COMMAND UPDATED

### What is it?

CBM-Command is a disk manager for the Commodore 64, Commodore 128, Commodore VIC-20, Commodore Plus/4, and Commodore Pet/CBM computers. It is written in the vein of Norton Commander or Midnight Commander; but, it is much simpler due to the target platforms. Each Commodore model has its own native version of the application.

### New Features:

- It can detect the formats of Creative Micro Design's partitions and sd2iec's disk images.
- It knows automatically when to create either single-sided (D64) or double-sided (D71) images.
- CBM 8050 and CBM 8250 (SFD-1001) disk copier.

### Changes from 2.1rc2:

- After showing a message, the status line always will let you know when you have tapped the RETURN-key.

<http://cbmcommand.codeplex.com/>



## PixelJam Demoparty

**From:** Robert Bernardo

**Sent:** February

**To:** Commodorefree@commodorefree.com

**Subject:** PixelJam Demoparty at Notacon 9, April 12-15

Demoparty time! The PixelJam Demoparty crashes through again during Notacon 9 in Cleveland, Ohio, U.S.A, April 12-15. Join the 300-400+ attendees who will be descending upon the Hilton Garden Inn Hotel there.

Get those demo entries in! PixelJam is looking for demos ranging from the most modern to old-school. For the Commodore and Amiga enthusiasts, your entries can be in the old-school, wildcard, or music compos. Here is some more info, straight from the Notacon blog, "PixelJam will feature a total of 9 compos ranging from music to animation to traditional demos. Notacon itself has pledged over \$1,500 in total prize money, with more prizes to be announced as we secure sponsorship deals. We are also dedicating 2,500 square feet of conference space to a coding lounge and projection space. This space will be expanded by an additional 3,000 square feet for demo screenings on Saturday night the 14th."

Play with the other happy enthusiasts in the Video and Board Game room. Open 24 hours a day, the game room will have everything from the most modern consoles to Commodore and/or Amiga computers to board games. (The Fresno Commodore User Group will not be able to attend this time, but Keith M. and others will fill in.) Listen to the various presentations from those at Notacon 9. Speak out at the broadcast at Notacon Radio. Check out the Notacon hacking room. Or hang out in the Notacon Ham Radio room or in the coding lounge.

Early registration closes on Feb. 10. Register now, because

after the 10th the registration fees go up.

For more information, go to the Notacon website at

<http://www.notacon.org>

For more demoparty info (though not yet updated with this year's bits), go to

<http://www.pixelj.am>

Truly,

Robert Bernardo

Fresno Commodore User Group

<http://videocam.net.au/fcug>







# 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)



# Multiple Classic Computer



C64



AMIGA



Connect to TV  
and Play !

The ultimate  
Retro Gaming  
Device

ATARI 2600



Online Order:

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

More Information:

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



GTW 64!

## NEWS

GTW 64!

## MORE GAMES THAT WERNT FOUND

GTW 64!

## 'ANGEL OF HELL II'

Angel of Hell II is a preview of a game by 576 KByte which was produced back in 1996. Sadly it never got completed. It was the sequel to this game.

The preview spans a very impressive 3 disk sides, and comes with a superb introduction sequence which then brings you onto the main game itself.

The game needs an English translation, so I cannot describe much about the game's story at present - but the game is an impressive looking icon driven graphic adventure game - which looks very promising indeed.

Development of the game overall got quite far, but it seems that it was quite late into the C64's life, and possibly due to poor sales of previous titles - the game was abandoned as a result. Recently Jazzcat managed to get hold of the remains on the game which we can now present to you to check out for yourselves.

It seems a shame that this one never quite got finished, as it looked very promising and might not have been that far away from completion. Still, at least it is now preserved and people can get an idea of what might have been. Please refer to the text file for instructions on how to get some of the separate parts working.

We hope to hear from the developers themselves about the game and the intentions very soon!



Credits

Code - Robért Olessák

Music - Olivér Gáspár

Graphics - Unknown

[http://www.gtw64.co.uk/Pages/a/Review\\_Angelofhellii.php](http://www.gtw64.co.uk/Pages/a/Review_Angelofhellii.php)

## 'PEN PALZ'

1997 John Pericos and Nathan Perry

PenPalz was a neat little game that was being developed back in Jan 1997 by John Pericos and Nathan Perry - intended for GO64.

The idea is simple - draw on the screen as much as you can and try to fill in a certain percentage of the screen before time runs out. Sort of like Qix in many ways - and with the same problem of having creatures on the screen which are trying to stop you. Collecting the bonus colour blocks on the screen are there to make your life far easier. Although the game got to a promising early stage (with titles, music and at least a level) - unfortunately the game was never finished due to both developers getting tied up and busy at University. After that, the game was shelved and forgotten about, and John went onto developing for the GameBoy.

In early 2012, John got in touch with GTW64 and offered the game to us. He had been cleaning up stuff in his old room at his parent's house and found a bunch of disks - one of which had the last version of PenPalz. To ensure it got preserved and share it with others, John kindly passed on a copy of the game to put on the website. And here it is! The game is fully playable, and contains one level for you to try. This is as far as the game got before it was shelved. The download also includes a text file which explains more about how to play the game.

John offers hope that maybe one day he may even complete the game - however, John lost contact with Nathan Perry and hopes that maybe this entry will get Nathan in touch again to maybe get this project moving again. If you're out there Nathan, then please get in touch! For now, check out this superb preview of a very promising game!

Frank (Additional source credits - John Pericos)



Credits

Code - John Pericos

Music - Sidchip Scratchers

Graphics - Nathan Perry

[http://www.gtw64.co.uk/Pages/p/Shots\\_Penpalz.php](http://www.gtw64.co.uk/Pages/p/Shots_Penpalz.php)

# NEWS

## A-EON Updates "Boing Ball" re-visited News from A-EON Technology

### Hardware Accelerated 3D using 2D Compositing

AmigaOS graphics guru, Hans de Ruyter, has issued another update to his RadeonHD graphics driver for the AmigaOS. Version 0.32, which makes its official debut on the AmigaOS 4.1 Update 5 CD especially created for AmigaONE X1000 "First Contact" system, delivers full 2D support for all Radeon graphics cards from the Radeon X1300 through to the RadeonHD 4890. When asked about the improvements in his new driver De Ruyter said, "I'm quite pleased with the updated driver. Apart from fixing a few bugs, the biggest change is the addition of homogeneous texture coordinate support to the compositing function. This allows 2D warping and even perspective-correct texturing". In reply, A-EON Technology's Trevor Dickinson commented, "I am pleased to help support Hans' graphics driver work and look forward to the future release of hardware accelerated 3D graphics drivers which should benefit the AmigaONE X1000 all other AmigaOS 4 users".

To illustrate the improvements in his new driver De Ruyter created "Composite3DDemo", an interactive 3D "Boing Ball" demo remake that renders everything using hardware accelerated 2D graphics without the use of 3D drivers. Originally developed as a test application for the RadeonHD driver running on the AmigaONE X1000, it demonstrates what can be done with AmigaOS 4.1's powerful compositing feature and a little creative thinking. In other good news De Ruyter revealed that he is releasing the source-code which he hope will help others to use these features, and encourage them to see what they can do. The "Composite3DDemo" is included on the AmigaOS 4.1 Update 5 CD and can be downloaded from the web link shown below. Please also check the link to the "YouTube" video.



About Hans de Ruyter & HDRLab: Hans de Ruyter is an electrical & electronics engineer and leading AmigaOS developer and Amiga enthusiast. His website - HDRLab - details various projects, many of which are AmigaOS related. The biggest of these is the RadeonHD driver for AmigaOS, an ambitious project that aims to deliver modern graphics capabilities for the AmigaOS platform.

#### Links:

Composite 3D Demo download

<http://hdrlab.org.nz/projects/amiga-os-4-projects/c3d/>

RadeonHD development log

<http://hdrlab.org.nz/projects/amiga-os-4-projects/radeonhd-driver/radeonhd-development-log/radeonhd-chip-0-32-bug-fixes-and-fully-working-2d/>

Composite 3D Demo video

<http://www.youtube.com/watch?v=pDDBhGLTPPI>

A-EON Technology

<http://a-eon.com/>

YouTube Link

[https://www.youtube.com/watch?v=pDDBhGLTPPI&feature=player\\_embedded#](https://www.youtube.com/watch?v=pDDBhGLTPPI&feature=player_embedded#)

## Sam440ep\_setup 1.3 Released

### from m3x via

Sam440ep\_setup is a little utility to optimize performance on your Sam440ep board (mini-itx and flex versions)

#### What's new:

use a more aggressive instruction cache speculative prefetching setting. This allows a faster execution for some software, for example the following Blender speed test increased performance by 10%-20%:

```
bldyn -b projects/bricks_and_water_plugin.blend -o ram:pool
-F JPEG -f 0
```

As a side note some values returned by Ragemem are incorrect (MIPS) or lower (L2 cache read64/write64) when this setting is enabled. To disable this setting, add NOCCR argument to the command line.

Currently on OS4Depot upload queue. <http://os4depot.net/>



# NEWS

## PDXCUG.org Meeting Reminder C=FREE WELL I MISSED THE DEADLINE BUT.....

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

Subject: PDXCUG.org Meeting Reminder  
From: "PDXCUG Admin"

Hello, PDXCUG.org Fans!  
Missed last month? See the meeting highlights:  
<http://pdxcug.org/meetings.asp>

### Unable to attend the meeting in person? No worries.

Come chat with us using your RR-NET or Comet64 (or use either in VICE).

- We'll be chatting with you live from the meeting.
- Interact with others Commodore enthusiasts around the world.
- Play online games with others

<http://pdxcug.org/Portland-Commodore-Users-Group-Meetings-Live-Online.asp>

Find us in the c64 chat room

### PDXCUG.org Meeting:

Thursday, February 9, 2012 @6:00 pm - come as early as 5:00 pm  
More info & location: <http://pdxcug.org/meetings.asp>

### This Month's Meeting Agenda:

Feature: Commodore 16

Demo: C64List 3.0

Demo: TBA

Tinker Time: Time for YOUR projects

Music Time: We'll have some software to make and play cool music  
Gaming Time - Countdown to Shutdown: Prevent meltdown at the nuclear facility  
Show & Tell - Bring your cool Commodore items

### As usual, we'll have plenty of time for these other activities:

- Gaming Time
- Show & Tell: Bring anything you'd like to show off
- Tinker Time:
  - Soldering
  - Hacking
  - Coding
  - Repair
  - Cleaning of equipment

### What you can bring:

- Show & Tell item - have something neat to show everyone?
- Equipment that you want diagnose, fix, or just show off
- Cool Commodore gear or software to share
- Your significant other - plenty for all to do and enjoy

See you there,- Goog

PDXCUG.org is Portland's Official Commodore Users Group  
Serving the greater metropolitan area - Portland, OR USA  
Meetings every 2nd Thursday of the month

Twitter: @pdxcug

<http://pdxcug.org>



## iBatch V1.4: new Features/ Screenshots

News from gerograph

Hello,...

As promised, find feature list of upcoming iBatch V 1.4 below. Detailed infos, pdf Documentation and screenshots can be viewed on iBatch Website. However, iBatch V1.4 will not be available on that website for download, so check [www.boingsworld.de](http://www.boingsworld.de) website for updates. If you are into German language make sure to listen to boingsworld.de Podcast/Episode 24. All the news will be in there, including an interview with the iBatch author.

### New Features...

- ignore aspect ratio when resizing
- use wildcards when renaming
- automatic selection of suitable scaling method (in "Quality Mode")
- bicubic
- linear
- window
- with or without lowpassfilter
- sharpen and soften your images
- FXPlugins

- Gamma correction for all images in batchlist
- image too grayscale for all images in batchlist
- add watermark

All this resulted in a complete rewrite of the so called "conversion profile editor". A complete new GUI (see Screenshot) will allow you to define different conversion profiles even easier than before.

### Main Features since release...

- batch converting, renaming, resizing, turning
- jpg, png, bmp, iff-ilbm in/output
- autonumbering when renaming
- add capture date when renaming
- all bundled in predefinable conversion profiles
- built-in thumbnail viewer/batchlist creator
- built-in batchlist browser
- exif and embedded thumbnail support
- plugin capability
- html album plugin
- image subtitle plugin (e.g. for html album)
- workflow manager plugin
- slideshow plugin (Win/AROS/Mac/Amiga)
- transfer EXIF Header plugin

# NEWS

## AmiDARK Engine Rel0.5 is Now Available!

News from AmiDARK

Hi All,

The new alpha build of the AmiDARK Engine is now available for both AmigaOS4.1 and MorphOS 2.7.

Here are all the major changes between this build and the previous one (that was only for AmigaOS4.1)

I have Restructured entirely the archive content with more "specific" folders.

1.Folder SDK that contain the .a library needed, it's the game engine. Also contain the .h definition files needed to know which functions are available.

2.Folder HELP that'll contain all help in HTML file format.

3.Folder RELEASE contain files like default app icon and an "AmiDARK Engine" logo you can put if you want in your application.

4.Folder "Source Code & Binaries" that contain all actual source code and binaries :

4.a : "Commands sets" will contain "function use" samples

4.b : "DarkBASIC Professional" contain demonstration ported from DarkBASIC Professional to the AmiDARK Engine.

4.c : Technical Demo" will contain special technology demonstration.

4.d : Will contain "default" project "skeleton" definition for various tool (like AmiDevCPP, CodeBench, etc ... ) that can be used to create new projects

Many news sample, source code, technical demonstration and DarkBASIC Professional portage will appear with the next release of the Game Engine.

I have also made many changes/improvements in the game engine.

Don't forget that under MorphOS, the Escape Key don't work under GLUT so, until this bug is fixed, I've put the "Quit Application" key to the Function Key F12.

Under AmigaOS 4.1, the "Quit Application" key is always "Escape Key" as it work correctly.

**Here are the details of these 2 archives.**

What is important to know before downloading the AmiDARK Engine :

The engine is under development, "Work in Progress". the author/developer may not be responsible for any problem that can happen/occur when you Download/uncompress/use the AmiDARK Engine alpha version. You use it entirely at your own risk.

If you want to know all changes made from the previous AmiDARK Engine release 0.4c,

Simply jump here : <http://www.amidark-engine.com/spip.php?article28>

If you want to support the project, simply jump here : <http://www.amidark-engine.com/spip.php?article6>

Here are the direct links to download these 2 builds :

MorphOS 2.7+ build :

[http://files.amidark-engine.com/AmiD...ne\\_Release0.5c\[MorphOS2.7\].lzh](http://files.amidark-engine.com/AmiD...ne_Release0.5c[MorphOS2.7].lzh)

AmigaOS4.1u4+ build :

[http://files.amidark-engine.com/AmiD...ne\\_Release0.5c\[AmigaOS4.1u4\].lzh](http://files.amidark-engine.com/AmiD...ne_Release0.5c[AmigaOS4.1u4].lzh)

I hope you'll enjoy test/development under the AmiDARK Engine.

Kindest Regards,

Frédéric "AmiDARK" Cordier



# NEWS

## iBatch V1.4: new Features/ Screenshots

News from gerograph

Hello,...

As promised, find feature list of upcoming iBatch V 1.4 below. Detailed infos, pdf Documentation and screenshots can be viewed on iBatch Website. However, iBatch V1.4 will not be available on that website for download, so check [www.boingsworld.de](http://www.boingsworld.de) website for updates. If you are into German language make sure to listen to boingsworld.de Podcast/Episode 24. All the news will be in there, including an interview with the iBatch author.

### New Features...

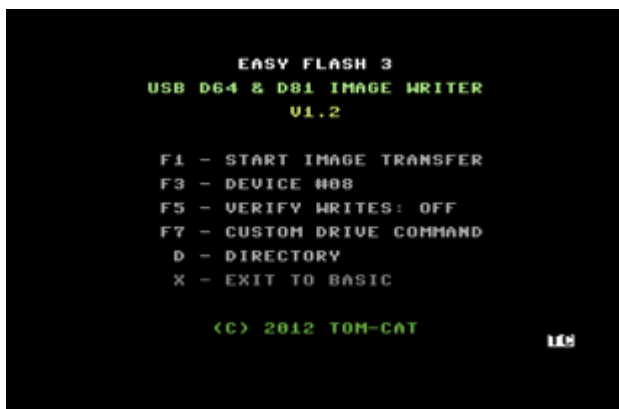
- ignore aspect ratio when resizing
- use wildcards when renaming
- automatic selection of suitable scaling method (in "Quality Mode")
  - bicubic
  - linear
  - window
- with or without lowpassfilter
- sharpen and soften your images

- FXPlugins
  - Gamma correction for all images in batchlist
  - image too grayscale for all images in batchlist
  - add watermark
- All this resulted in a complete rewrite of the so called "conversion profile editor". A complete new GUI (see Screenshot) will allow you to define different conversion profiles even easier than before.

### Main Features since release...

- batch converting, renaming, resizing, turning
- jpg, png, bmp, iff-ilbm in/output
- autonumbering when renaming
- add capture date when renaming
- all bundled in predefinable conversion profiles
- built-in thumbnail viewer/batchlist creator
- built-in batchlist browser
- exif and embedded thumbnail support
- plugin capability
- html album plugin
- image subtitle plugin (e.g. for html album)
- workflow manager plugin
- slideshow plugin (Win/AROS/Mac/Amiga)
- transfer EXIF Header plugin

## EF3 USB Utilities v1.2



This is the v1.2 release of the EF3 USB Transfer Utilities. This time with only bug fixes and a few additional features. The following utilities are present:

### USB Test

This utility will test your USB connection between PC and the EF3/C64

### EF3 Send

This will send a .PRG file over to the C64 and start it. This is very fast: less than 3 seconds for 200 blocks. Files with sizes up to 250 blocks are supported. If the file does not load to \$0801 (basic start) then it will be started from its loading address (replacement for CodeNet).

### EF3 Copy

This will copy FILES that are inside of the .D64, .D81 or .D71 images over to any drive connected to the C64. Only PRG and

SEQ file types are supported. Also single .PRG or .P00 files can now be transferred over.

### EF3 Image Writer

This will write any .D64 or .D81 image on your PC over the USB cable to the 1541/1581 drive on your c64. You can turn on VERIFY option which will verify each sector after the write. Upton 3 retries will be done.

### EF3 Image Reader

This will read a floppy disk in your 1541 or 1581 floppy drive on your c64 into a .D64 or .D81 image on your PC.

Only 35 tracks .D64 images are supported. Also note that the Transfer (especially of the images) might be slow since Kernal routines are used for maximum compatibility (and because of my lack of 1541 knowledge). So use Jiffy DOS or similar in your C64 \*and\* your drive to make the transfer faster.

There is a readme file present in the archive where you can find more information.

Link: <http://pc.sux.org/tomcat/ef3utils12.zip>

Note: I cannot take responsibility for any damage to the data or hardware these utilities might cause. Just to be on the safe side

New in 1.2: removed 40 track .D64 support since it was not working. Added VERIFY option for EF3 Image Writer. Added .PRG and .P00 support to EF3 Copy utility.

Best regards, Tomaz

# NEWS

## FREESCAPE GAME NEW ISH

From the website:

### Freescape Games

This page is dedicated to the one of the best 3D Engine to come out for the Commodore computer (Well, the best I've seen so far). Here are a collection of games that uses Freescape engine.

<http://www.commodorecheetah.co.uk/FreescapeGames/Freeescape.htm>

### *A Chance In Hell*

2011, Steven Flanagan.

A Chance In Hell, Released 2011, Written by Steven Flanagan

This is a Freescape game created using the 3D Construction Kit. The game is public domain. You are free to copy it, host it, etc.

### Two versions exist:

AChanceInHell.d64 - Standard C64 version - can be run on any C64 or emulator.

AChanceInHellEXtreme.d64 - EXtreme version for SuperCPU or fast emulation (Eg 25x)

### Essential Game Controls:

You will need these controls to play and complete the game:

O – forward (or joystick)

K – back (or joystick)

Q – left (or joystick)

W - (or joystick)

B – shoot/get/use/push/open (or joystick)

P – look up

L – look down

I - Face forward

### Additional controls are as follows:

Z - sidestep left

X - sidestep right

R = Move Up (Rise/Stand)

F = Move Down (Fall/Crouch)

U = U-Turn

CTRL+key = When used in conjunction with movement keys will accelerate the movement.

ESC/RUN-STOP = Restart environment.

CTRL+key = When used in conjunction with movement keys will accelerate the movement.

Control only required for the standard version:

SPACE = Toggle Sights Mode/Movement Mode (not needed in the EXtreme version)

Controls that exist but you don't need:

M = Tilt Right

N = Tilt Left

A = Activate Object (not needed in this game)

### Game Story:

The Devil has invaded your coastal village, and has kidnapped the village population. As the village priest, you alone must defeat the devil and rescue your flock. A main road goes through the village, but don't expect anyone to stop for you as they have heard the news.

This is a short game that can be complete in one sitting. The puzzles are straightforward - there are no 'obscure' puzzles. The game is always completeable - there is no situation where the player cannot complete the game.

### Hints:

\* Walk into doorways to get through them.

\* If anything looks suspicious/useful, then try shooting it.

\* Make use of the look down and look up to see where you are going.

\* Remember this is a coastal village, with a cliff edge somewhere!

\* There has been a twister recently so light objects may have been blown to high places.

The EXtreme version of A Chance In Hell is revamped to run with the SuperCPU or with a C64 emulator running at maximum speed. A speed of 2500% (25x) is ideal. Any speed above 10x is recommended. VICE and Frodo tend to be the best emulators at reaching these speeds. Tip: Increasing 'skip frames' to 4 or 6 can help the emulator achieve these speeds if you have an older PC. At a speed of 25x, you will be rewarded with a very smooth Freescape experience.

### Emulator speed guide for the EXtreme version:

1x to 5x - too slow

5x to 10x - ok

10x to 15x - good

15x to 20x - great

20x to 25x - best





# NEWS

## AmigaOS 4.1 Update 5 CD for the AmigaONE X1000 "First Contact" system has gone GOLD!

News from A-EON Technology News

<http://www.a-eon.com/>

### AmigaOS 4.1 Update 5 CD for the AmigaONE X1000 "First Contact" system has gone GOLD!

A-EON Technology and Hyperion Entertainment are pleased to announce that the AmigaOS 4.1 Update 5 CD, exclusively created for the AmigaONE X1000 "First Contact" system has gone gold. The professionally re-mastered CD will be supplied in a printed boxed set along with copies of the AmigaONE X1000 and AmigaOS 4.1 "Quick Start" guides. Steven Solie, Hyperion Entertainment's AmigaOS Development Team Lead said, "The team worked hard to reach this important milestone in the development of the Amiga Operating System. With AmigaOS 4.1 Update 5 completed for the AmigaONE X1000 we can concentrate on versions for all the other supported hardware platforms as we continue to push towards the future release of AmigaOS 4.2." Trevor Dickinson added, "I would like to thank the AmigaOS developers and the whole AmigaONE X1000 beta test team who have worked so tirelessly behind the scenes to make this a reality. I would also like to give special thanks to a few key members of the team who went above and beyond the "call of duty" to ensure the success of this project. You know who you are." The AmigaOS 4.1 Update 5 boxed set will be supplied to all AmigaONE X1000 "First Contact" customers who will also receive a free update to AmigaOS 4.2 when it is commercially released by Hyperion Entertainment.

### AmigaONE X1000 "First Contact" systems now shipping

A-EON Technology and AmigaKit are pleased to announce that the deliveries of AmigaONE X1000 "First Contact" systems have commenced. Matthew Leaman, Managing Director of AmigaKit commented, "we are pleased to finally supply this eagerly awaited AmigaONE computer system to our customers". Trevor Dickinson added, "I have just received my own AmigaONE X1000 system and I hope "First Contact" owners will have as much fun as I'm having with my new "Amiga". All that remains to be said is let's "keep this party going!"

### A-EON Technology

About AmigaKit: AmigaKit is a trading division of Leaman Computing Ltd based in Cardiff, UK. It acquired the stock of Eyetech Group Ltd in 2006 and quickly established itself as a world-wide market leader in the retail and distribution of Amiga hardware and software. A-EON Technology appointed AmigaKit as its primary distributor for all its AmigaONE products.

About A-EON Technology and the AmigaONE X1000: A-EON Technology is the developer of the AmigaONE X1000 computer system designed by Varisys Ltd for running the AmigaOS. The AmigaONE X1000 is not like other computers. It is a culmination of efforts by real Amiga enthusiasts and

developers to create powerful, modern desktop hardware for the Amiga Operating System. It is the natural evolution of the Amiga's PowerPC lineage and is based on the PA-Semi Dual-core PA6T-1682M CPU and includes Xena, a "Software Defined Silicon" co-processor. Above all it runs the latest version of the AmigaOS.

About Hyperion Entertainment: Hyperion Entertainment is the developer of the AmigaOS.

AmigaKit website: <http://www.amigakit.com/>



# AmigaONE X1000

# S U D O K U

## by John Borrowman

The original version of this program was published on LOADSTAR #60 as 'digithunt' by Dave Johannsen. He was no doubt ahead of the times because a few years later, puzzles like this became the latest craze. These puzzles have appeared in almost all daily newspapers for the last decade.

I modified the program to use a custom font with a large 2x2 character set of numbers in a proportionally larger grid. I also used brighter colours and included ML routines from several of the Loadstar toolboxes. I did some 'tweaking' here and there, but the main algorithms remain (nearly) unchanged.

Sudoku puzzles consist of a 9x9 grid, broken down into nine 3x3 boxes. To solve a Sudoku, the numbers 1 thru 9 must fill each row, column, and box. Each number can appear only once in each row, column and box. You can figure out the order in which the numbers will appear by using the numeric clues already in the boxes. The more numbers you name, the easier it gets to solve the puzzle!

A box cursor appears in the centre of the grid. Use the cursor keys to move the box cursor to an open space. Press a number key to put that number in the space. A ZERO will clear any space not containing one of the original puzzle clues.

Whatever number you insert in the puzzle will be checked. A buzzer will sound if the number entered is incorrect, and the number will not be entered in the square. A tally is kept of all errors and a timer clock tracks how long it takes to solve the puzzle. The objective is to get the fastest time with the least errors.

There are nine different levels, each level more difficult than the preceding one.

From the initial screen, you can:

1. choose the puzzle level to play
2. clear the high scores file
3. LOAD a previously SAVED puzzle
4. Build your own puzzle

### Enter skill level (1-9)

The cursor keys can be used to scroll up and down thru the levels, and press RETURN on your choice, or just type the number and then press RETURN.

### CLR to clear scores

Press the HOME key to clear the file of high scores.

### Build your own puzzle

This option can be used to enter puzzles from newspapers, magazines, etc. First enter the printed puzzle (unfinished) and press F1. It will then ask if you want to enter the solution, which can be skipped. If you do not plan on completing the puzzle right away, press F3 to SAVE it for later.

### LOAD a previously SAVED puzzle

Selecting this option will show you a list of all SAVED puzzles on file. Use the cursor keys to pick a file to LOAD and press RETURN.

While solving a puzzle, there are several option to assist you. The '?' key will show a random hidden digit, while the '/' key will show the digit under the cursor. Both of these will cost you 2 error-points'. The F1 key will offer hints about which rows, columns, or boxes may hold keys to the puzzle solution. Each 'hint' costs 1 'error-point'.

F7 or 'Q' will end the puzzle.

By the time this project is done, the program files should be linked and packed by Loadstar's Star Trio programs.

Have fun!!

### COMMODORE FREE

I will place a download for this file on the Commodore free website look in the downloads section



# ASSEMBLER TUTORIAL PART 2

## FROM MICROMART BY SHAUN BEBBINGTON



**Author's notes:** This is the second lot of tutorials (parts 5 through to 8 inclusive) that were published in Micro Mart magazine between February and July 2011 in the 'Specialist' section, which also includes Amiga, Apple, Linux and gaming news and views – see [www.micromart.co.uk](http://www.micromart.co.uk) for more information about this publication.

**Disclaimer:** The tutorials assume no prior knowledge of machine code, so if this means you, read the last issue of Commodore FREE which has parts one through to four. None of this will not be very useful if you are not a complete beginner and I'm only providing a foundation on which to build. I'm sure all of the example code could be improved greatly. Remember: the more you experiment with your code, and the more you read up about the Commodore 64's hardware, the more you will learn. Even better if you enjoy programming, because this will aid your progression as much [or more] than anything else.

I retain the copyright for these articles, which are used in Commodore FREE with permission. If you would like to contact me about them, then you may do so through the Micro Mart forums and a link is provided. Without any further ado, here is the first four weeks, which covers some of the very basics. The original images that I provided with the articles are included for illustration purposes only.

### Doing things indirectly – Part 5

For the first instalment of this series, I briefly gave you an overview of the Commodore 64 and also looked at some examples of writing text to the screen, using a routine in the Kernal (notice the spelling) at memory location FFD2 in hexadecimal, and also writing to the screen RAM directly. Well, it may not seem like progress, but this week's example is writing text to the screen RAM again.

As with nearly every programming language, there is always more than one way to solve a problem, and in this respect, how you've constructed the code doesn't often matter as long as you have met the objectives you set out to achieve. But in the context of the limited resources available to the C64 (and other 8-bits), efficiency should always be a consideration. Making your program smaller not only saves memory, but processor time, and the more of that you have, the more you can make the computer do. Try this example:

```
; Here's where our program will reside in memory (12*4096)
*=$c000
lda #$00      ; Let A=0
sta $fb      ; Stores it at zero-page location hex fb
lda #$04      ; Let A=4
sta $fc      ; Store at hex fc
lda #$00      ; And again...
sta $fd      ; Stores it at zero-page location hex fd
lda #$d8
sta $fe
ldx #$0f      ; We'll be using X for to write to the colour
nybbles
ldy #$00      ; let Y=0
tya          ; Transfers Y value to A (Let A=Y)
sta $d020
sta $d021     ; We may as well change the border and
screen background colour
LOOP         ; Loop marker
lda STRING,y ; Read byte at location STRING+Y into A
cmp #$00     ; Compare contents of A to zero (terminator)
beq END      ; If equal, branch to END
sta ($fb),y  ; Using indirect indexed addressing
txa         ; Transfers value of X to A
sta ($fd),y ; Again, indirect indexed addressing used
iny         ; Increase Y by one (Y++)
jmp LOOP     ; Unconditional jump to LOOP
END         ; END marker
rts         ; ReTurn from Subroutine (back to BASIC)
STRING     ; Data marker
; Test data - 40 characters long. Zero used as terminator:
.text "0123456789012345678901234567890123456789",0
```

What we're doing is writing to the screen and colour RAM (or 'colour nybbles', if you prefer, again note the spelling), but using indirect indexed addressing mode. How this works is that the memory location for both screen and colour RAM are written to the zero-page at 251 and 253 respectively (or in hex, fb and fd). We set this first because it'll be used in our conditional routine later. After storing 15 in X (light grey) and changing the border and background to black, the program reaches our main routine.

Each character in the data area (STRING) is passed through the accumulator (A) and then it's written to the 16-bit address indexed in the zero page at hex fb. Note the beginning of the program which stores '00' first and '04' in the next location. Well that's because the 6502 writes 16-bit addresses in reverse order, so the processor will see it as 0400 in hex, or 1,024 in decimal, which is where the screen starts. We then pass the value of X to A and do the same (X is constantly 15 in this case) but writing to the colour RAM instead. Again, at the beginning of the program, '00' and 'd8' will be read as d800 by the processor.

We're using the Y register for this addressing mode, so as Y is increased by one, it is 'added' to the indexed 16-bit memory locations stored in the zero page for each pass until it reaches the terminating value, so when Y=5, sta (\$fb),y will write A (the accumulator) to hex 0400+05. Once all of the data is written, you're returned to BASIC.

Further explanations about this are at [tinyurl.com/C64-Coding](http://tinyurl.com/C64-Coding); it's really very simple, but I know it may take a while to get your head around. Feel free to ask any questions there too

```

sta $fd
lda #$08
sta $fe
ldx #$0f
ldy #$00
tva
sta $d020
sta $d021
LOOP
lda STRING,y
cmp #$00
beq END
sta ($fb),y
sta ($fd),y
iny
jmp LOOP
END
rts
STRING
; test data - 40 characters long. Zero used as terminator:
; text "0123456789012345678901234567890123456789"

```

### The joys of indirect indexed addressing

#### Simple scrolling – Part 6

In part five, we looked at indirect indexed addressing mode, and used it to write characters to the screen, and also the colour RAM (or colour nybbles, in Commodore-speak). In essence, this [addressing mode] is another way of moving bytes around the computer's memory which could also be a real byte-saver as your programs become bigger and more complicated. For now, don't worry about this.

This tutorial is a basic 'scrolly-text', more sophisticated variants have seen in many demos and scene intros on the Commodore 64 over the years. This code is shifting the characters only (no pixel scrolling yet, which is actually quite east with the VIC-II chip). Let's have a look at the code:

```

*=$8000 ; Program is stored at 8*4096
lda #$04 ; A=04
sta $fc ; Store A at hex fc
lda #$00 ; A=00
sta $fb ; Store A at hex fb (Processor will see it as
$0400, or start of screen RAM)
sta $d020
sta $d021
tay ; Let Y=A
LOOP ; Loop marker
lda STRING,y ; Read byte at location STRING+Y into A
cmp #$00
beq SHUFFLE ; If equal, branch to SHUFFLE
sta ($fb),y ; Using indirect indexed addressing again
iny ; Increase Y by one (Y++)
jmp LOOP ; Unconditional jump to LOOP
SHUFFLE ; This will move each character to the left
tay
LOOPY
lda STRING,y
cmp #$00
beq AGAIN

```

```

sta BUFFER,y ; This takes the character at STRING+Y and
stores it at
iny ; BUFFER+Y, which is a one-byte off set, then
Y=Y+1
jmp LOOPY
AGAIN
dey ; DEcrease Y by one (Y--)
lda BUFFER ; This passes the byte in the BUFFER and
writes it to the
sta STRING,y ; last byte of the STRING data before the
terminator (0)
ldy #$00
tva
jsr DELAY
jmp LOOP
DELAY ; This will slow down the text so that we can read
it
pha ; 'Pushes' A onto the stack
lda $a2 ; Loads A with the current value of memory
location $00a2
sta $fd ; and stores it in memory location $00fd
DELAY1
sec ; SEts Carry flag (good practice when doing 6502
arithmetic)
lda $a2 ; Gets new value of $00a2 (used for the clock)
sbc $fd ; Subtracts with previous value stored above
cmp #$0f ; Compares result with 15 decimal
bcc DELAY1 ; Branches if Carry flag is clear back to
DELAY1 marker
pla ; Restores old value of A from stack
rts ; ReTurns from Subroutine
BUFFER
.scr " " ; One char buffer
STRING ; Data (40 characters long)
.scr "...micro mart available every thursday.."

```

Okay, so there's some code to play about with there. The first thing to do would be to alter the text, but you could also try changing the speed at which the text is scrolled by looking in the 'DELAY' sub-routine. And if you want a real challenge, why not see if you can change the length of the message so that it's greater than 40 characters, and see if you can make the whole text scroll seamlessly across the top line of the screen, or even change the line at which it appears. Further explanations about this are at [tinyurl.com/C64-Coding](http://tinyurl.com/C64-Coding), with some hints on how solving these problems. Feel free to ask any questions there, or email me at [shaun@micromart.co.uk](mailto:shaun@micromart.co.uk). See you next time.

```

lda BUFFER ; This passes the byte in the BUFFER and writes it to the
sta STRING,y ; last byte of the STRING data before the terminator (0)
ldy #$00
tva
jsr DELAY
jmp LOOP
DELAY ; This will slow down the text so that we can read it
pha ; 'Pushes' A onto the stack
lda $a2 ; Loads A with the current value of memory location $00a2
sta $fd ; and stores it in memory location $00fd
DELAY1
sec ; SEts Carry flag (good practice when doing arithmetic)
lda $a2 ; Get's new value of $00a2 (used for the clock)
sbc $fd ; Subtracts with previous value stored above
cmp #$0f ; Compares result with 15 decimal
bcc DELAY1 ; Branches if Carry flag is clear back to DELAY1 marker
pla ; Restores old value of A from stack
rts ; ReTurns from Subroutine
BUFFER
.scr " " ; One char buffer
STRING ; Data (40 characters long)
.scr "...micro mart available every thursday.."

```

# ASSEMBLER TUTORIAL PART 2

## FROM MICROMART BY SHAUN BEBBINGTON

Happy face – Part 7

By now, you should be feeling more comfortable with assembly language, and hopefully you'll also want to be making your Commodore 64 do more and more exciting things.

This week, we'll look at user defined graphics by manually changing the @ symbol on the character set. Of course, this isn't the only way to draw graphics on the machine; it has hardware sprites and a bit-mapped mode too. I'll cover these at a later time.

What I recommend you do is to get a copy of the machine's memory map. Some important memory locations can be found at the website [codebase64.org](http://codebase64.org), or also on Bo Zimmerman's site at [tinyurl.com/C64-Memory-map](http://tinyurl.com/C64-Memory-map). As we'll be using the graphics chip (VIC-II), so it's important that you look at the locations at d000 in hexadecimal, or 53248 in decimal, even if you don't fully understand the explanations. Also, do a general Internet search to see what logical AND and OR gates do. Here's this week's routine:

```

*=$8000 ; Where our program resides in memory
sei ; turn off interrupts
lda $01 ; This uses the machine's IO register to tell. We
want to
and #%11111011 ; switch off bit 2 to tell the C64 to read
from the ROM
sta $01 ; as we're copying the character set stored in
ROM
lda #$d0
sta $fc
lda #$38 ; $fb will indirectly index $d000 (where the
character ROM is
sta $fe ; stored) and $fd will point to $3800, where
we're copying it
lda #$00
sta $fb
sta $fd
ldx #$08 ; We need to copy 8 'pages' of RAM, so X will
be our counter
tay ; Transfers value of A to the Y register
LOOP
lda ($fb),y
sta ($fd),y
iny
bne LOOP
inc $fc
inc $fe ; This increases our indirect index to point at
the next page
dex ; so we've copied one page, X is decreased
bne LOOP ; This will check if X is zero. If not, it'll branch
to LOOP
lda $01
ora #%00000100
sta $01 ; This points the IO register back to ROM
cli ; and then clears the interrupt flag, allowing
normal service

```

```

REDEFINE ; We'll redefine the first character in RAM (the
@ sign)
ldy #$07
DEFLOOP
lda DATA,y
sta $3800,y
dey
cpy #$ff
bne DEFLOOP
lda $dd00
and #%11111100
ora #%00000011
sta $dd00 ; This will point the VIC-II chip at bank zero
on the C64
lda $d018
and #%11111000
ora #%00001111
sta $d018 ; And this tells the VIC-II to display our
character set
rts
DATA
.db %00000000, %00111100, %01011010, %01111110
.db %01011010, %01100110, %00111100, %00000000

```

I've optimised the code for you to save a few bytes and so you have fewer lines of typing (I'm good like that). What it does is to take a copy of the character ROM (at d000 hex) into RAM (at 3800 hex), where we can manipulate it. On running the routine (by SYS 32768), you won't notice any difference until you press the @ key. Because this is the first character mapped in memory (from 3800 to 3807 hex), we changed the first 8 bytes to the bit pattern at the end of the program (after our DATA marker). Because of optimisation, or seeing so much binary, you may feel a little lost, but help is at hand by asking questions over at [tinyurl.com/C64-Coding](http://tinyurl.com/C64-Coding).

```

1 *=$8000 ; Where our program resides in memory
2 sei ; turn off interrupts
3 lda $01 ; This uses the machine's IO register to tell. We want to
4 and #%11111011 ; switch off bit 2 to tell the C64 to read from the ROM
5 sta $01 ; as we're copying the character set stored in ROM
6 lda #$d0
7 sta $fc
8 lda #$38 ; $fb will indirectly index $d000 (where the character ROM is
9 sta $fe ; stored) and $fd will point to $3800, where we're copying it
10 lda #$00
11 sta $fb
12 sta $fd
13 ldx #$08 ; We need to copy 8 'pages' of RAM, so X will be our counter
14 tay ; Transfers value of A to the Y register
15 LOOP
16 lda ($fb),y
17 sta ($fd),y
18 iny
19 bne LOOP
20 inc $fc
21 inc $fe ; This increases our indirect index to point at the next page
22 dex ; so we've copied one page, X is decreased

```

# ASSEMBLER TUTORIAL PART 2

## FROM MICROMART BY SHAUN BEBBINGTON

### Paging and logic – Part 8

If you recall, from part five of the series, I introduced you to how the 6502 processor indirectly indexes 16-bit memory addresses in the zero-page, as well as introducing (at least in the code) AND and OR gates last week. You may have been put off by the terminology that I've been using, but don't be as it's all very simple. But just in case you're lost, I'll explain further this week, starting off with memory paging.

On 6502-based micro computer systems (any 8-bit Commodore, the BBC Micro, Oric-1 or Atari XE/XL, for instance), a 'page' of memory is 256 bytes long from zero (or 00 in hexadecimal) to 255 inclusive (or FF hex). So, 'zero page' simply means the first 'page' of memory from 0 to 255, which in hex can be expressed as either 00 to FF or 0000 to 00FF. Therefore page one is 256 to 511 in decimal, or 0100 to 01FF in hex, and page two is from 512 to 767, or 0200 to 02FF hex et cetera. So each page number related to the first two hex numbers in hexadecimal address (00xx to FFxx), with the specific memory location being the latter two (xx00 to xxFF).

When you use indexed indirect addressing though, the 6502 processor will read the memory location in reverse order, which is low-byte, high-byte. So, if the zero-page address FB in hex has 10 stored there, and FC has hex 20, the processor will read 2010 hex (and not 1020), or 8208 in decimal. This is page 32, as 20 in hex is 32 in decimal. See part two for how to convert hexadecimal to decimal, or simply use an online converter or your calculator if you get stuck.

So, indirectly indexing a memory address simply means a 16-bit memory pointer stored in the zero page, which may not be readable in the code, i.e., sta (\$fb),y rather than sta \$2010,y - this is why comments are important as you may not be able to revisit your program you for some weeks or months.

From last week's example, you will have seen in the code the use of simple Boolean logic, using 'or' and 'and' gates. An 'or' gate will return 'true' when one or the other or both are true. As we're dealing with binary, this sets the bit to 1 when one or the other or both are true. For example:

```
*=$1000 ; Code assembles to 4096 decimal in memory
ldy #%10101010
sty $c000 ; Stores bit-pattern in Y at memory location
C000 (49152 decimal)
lda $c000 ; Get value of 49152
ora #%01010101 ; Logical 'OR' with alternative bit pattern
sets
sta $c001 ; all bits (11111111) to memory location
49153
```

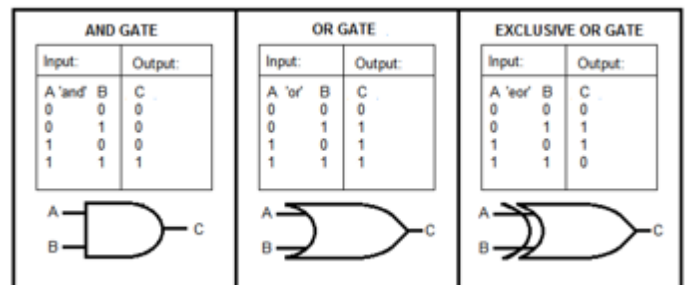
We can test this by reading the byte stored at 49152 and 49153 in BASIC by 'peeking' their values: PRINT PEEK (49152), PEEK

(49153) - the number will be returned as decimal, but you can easily convert each to hex or binary with a little help. 'And' gates work only when both are true, so continuing the program:

```
lda $c001 ; Passes value of 49153 to A
and #%11110000 ; Any bits set as 1 at 49153 will remain as
1 where
; 1 is already set in both
; binary octets
sta $c002 ; Stores new value at 49154
rts
```

Again, peek at the relevant value (and do the necessary conversion) to see how the bits have changed.

The other handy logic command in 6502 is 'exclusive or' (eor), which will set one or the other, but not both. To give you an example of what this means, if you eor a binary octet of 01111110 with 11111111, will set all bits to zero that are one and vice versa (answer will be 10000001). You could use this to create a reversed pattern of the smiley face from last week's example (part 7). Happy coding!



```
File Edit Search View Options Help
C:\MCP\programming\664\asm\dreamass\AND OR.asm
*=$1000 ; Code assembles to 4096 decimal in memory
ldy #10101010 ; Stores bit-pattern in Y at memory location
sty $c000 ; C000 (49152 decimal)
lda $c000 ; Get value of 49152
ora #01010101 ; Logical 'OR' with alternative bit pattern
sets
sta $c001 ; all bits (11111111) to memory location
49153
lda $c001 ; Passes value of 49153 to A
and #11110000 ; Any bits set as 1 at 49153 will remain as 1
; where 1 is already set in both binary octets
sta $c002 ; Stores new value at 49154
rts
```

# AMIGA FOREVER AND COMMODORE 64 FOREVER

Amiga Forever

<http://www.amigaforever.com>

<http://www.facebook.com/AmigaForever>

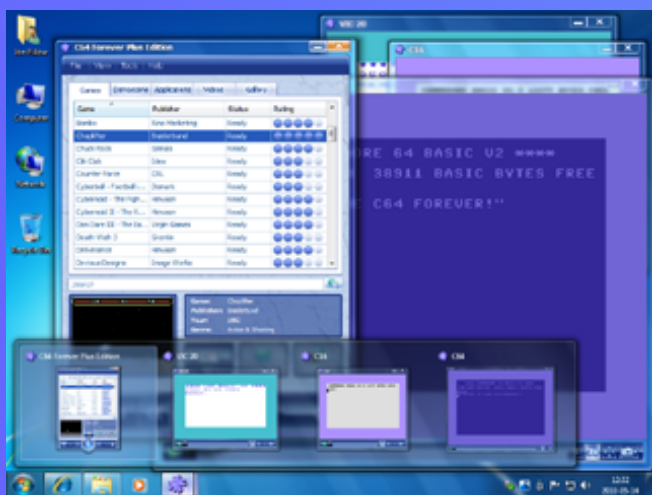
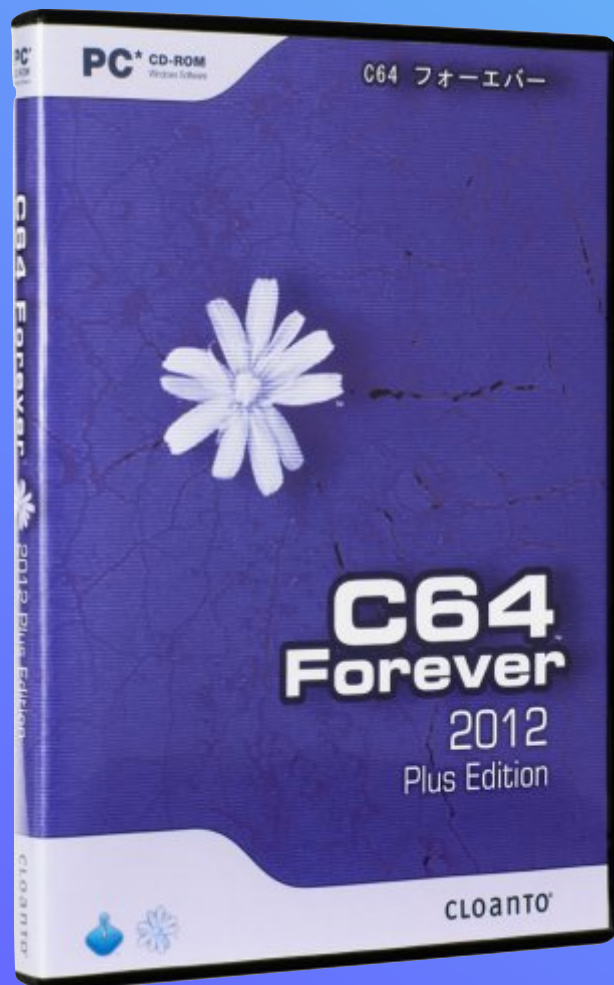
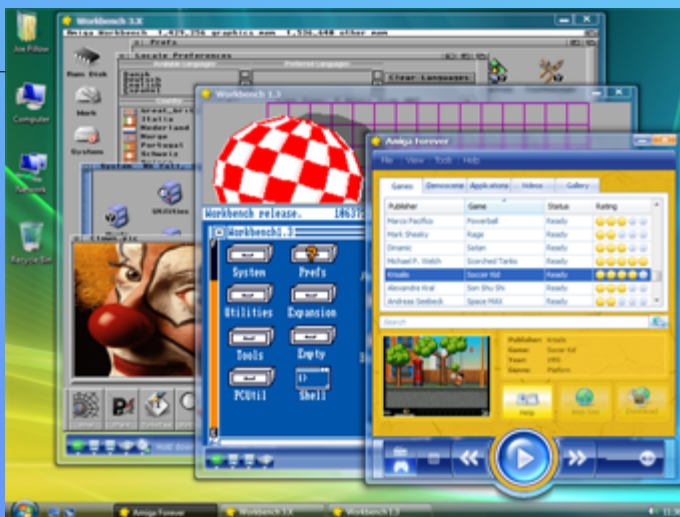
C64 Forever

<http://www.c64forever.com>

<http://www.facebook.com/C64Forever>

RetroPlatform Project

<http://www.retroplatform.com>





<http://www.cmdweb.de/>

Here you find information about the [SuperCPU](#), [FD disk drive](#), [HD series](#), [JiffyDOS](#) and [RAMLink](#). To order, pick your closest [distributor](#).

Furthermore, you can visit our [About CMD](#) section or the [Download area](#).



[www.commodorecomputerclub.co.uk/](http://www.commodorecomputerclub.co.uk/)

Welcome to the home of the **Commodore Computer Club UK**. We have set up to support all popular Commodore 8-bit computers, from the Commodore (CBM) PET range, through to the C128, so if you have a C64, VIC-20, C16, Plus/4 or even a C64DTV, and would like to know more about the club, please read the [club FAQs](#) or visit [our forums](#) if you have any further questions.





# The Datasette

## By Francisco Javier Gonzalez Cañete

The Commodore 1530/1531 (C2N) was/is the peripheral used by the Commodore computers to store software in magnetic tapes. It was the storage medium most used in Europe because it was cheaper than the disk drive that was, at least, five times more expensive.

The first datasette model appeared for the Commodore PET. There are four different models of the Commodore 1530, but all of them share the same functionality and they are fully compatible between themselves. The first of the units was integrated in the Commodore PET and it had a black lid with five buttons, without a counter and without a Save led light.

The first external model for the PET was black with five buttons, again without counter and without Save led light. The next model was also for the Commodore PET. This one was white with a black lid, but this time with counter although without a Save led light.

Finally, the most common and most often sold version was the model that was developed for the Commodore VIC-20. This model was white with a silver lid, it had six buttons, with a counter and a Save led light.

The last model (C2N 1531) was developed for the Commodore 16/116 and Plus/4 computers. It was identical to its predecessor but it was black in colour and the connector to the computer was different.

The C2N 1530 is connected to the computer using a special six pins border connector as shown in Figure 4. The C2N 1531 used a seven pin mini-DIN connector and is pin-by-pin compatible with the 1530 model (see Figure 5).



Figure 1. C2N 1530 connector pinout

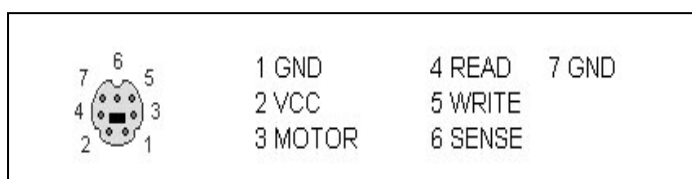


Figure 2. C2N 1531 connector pinout

The communication signals of these connectors are GND, VCC, Motor, Read, Write and Sense:

- VCC and GND: They are the power supply pins for the datasette (but not for the engine that moves the tape).
- Motor: This is the control signal and power supply for the engine that moves the tapes. This signal is controlled by the computer. When it is active (5V) the engine can start working.
- Read: This signal is used to send the serial data from the datasette to the computer.
- Write: This signal is used to send the data to be stored in the tape.
- Sense: This is a control signal that is active (0V) when any button have been pressed in the C2N. In that way, the computer is informed about that fact.

The Read and Write signals are used to transmit and receive the data from the datasette to the computer and vice versa. A serial frequency modulated protocol is used, that is, each pulse has a different period and each duration represents a piece of data. There are a lot of loaders and each of them uses different pulse lengths.

The pulses that are sent from the computer to the datasette using the Write signal are stored in an analogue form in the magnetic tapes; using a magnetic head. This head is a metal ring that is the core of a coil. If a current is applied to the coil, it will produce a magnetic field proportional to the electromagnetic induction. This magnetic field aligns the tape particles in the form that they can be read in the future using the inverse process.

Because the stored data are digital, the values finally saved are 0 or 1. Figure 3 shows an example of a recorded magnetic tape.

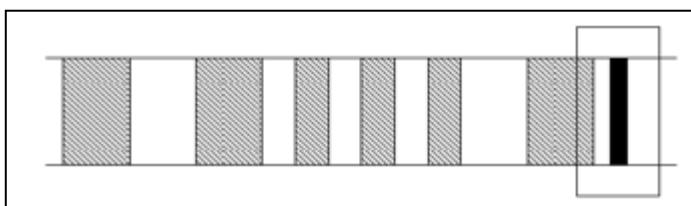


Figure 3. Aligned head

This way of storing data was very dependant of the mechanic factor. The head has to be very well aligned; at least with the same angle as the one that saved the tape, if not, some reading errors could be presented. Figure 4 depicts this situation.

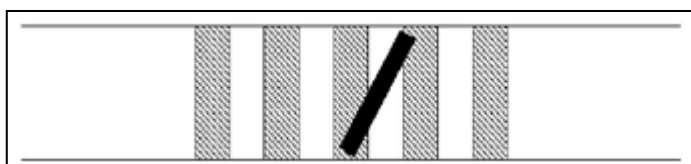


Figure 4. Bad aligned head.

In this situation, the reading process could produce some error because of the overlapping of some areas. Figure 5 shows an example of a correct and incorrect read because of a bad aligned head.

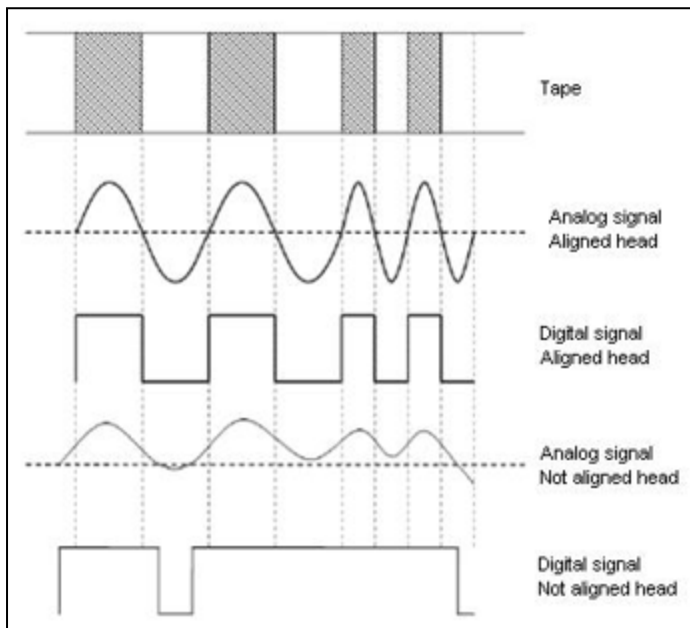


Figure 5. Example of a bad aligned head

In order to avoid this situation the datasette has a gap in the upper side with a screw that allows the user to adjust the head.

### The loaders

The loaders are the way to code the pulse length to save the information in the tapes. The original Commodore loader, the ROM loader, uses three kinds of pulses with different periods: Short (S) of 352  $\mu$ s (microseconds), Medium (M) of 512  $\mu$ s and Long (L) of 672  $\mu$ s. These pulses are interpreted in pairs in that way:

- (S,M) = 0
- (M,S) = 1
- (L,M) = New data mark
- (L,S) = End of data mark

Each byte is coded using ten pairs of pulses: the new data mark, eight pairs of pulses representing the eight bits and a pair of checksum pulses (this information is the XOR operation of all the bits transmitted in the byte 1 xor bit0 xor bit1 xor bit2 .... ). Figure 6 shows an example.

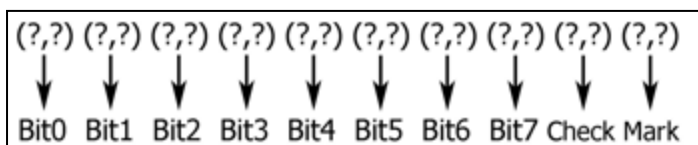


Figure 6.

Two blocks are recorded; the header, that contains a synchronization part and the name of the program and the data itself. Those blocks are stored twice with a space between them. Because of that, this loader is robust but too slow. Some software firms developed new faster and more reliable loaders,

reducing the loading time and allowing them to show pictures and play music during the loading process. Those loaders use a different pulse length and codes. Most of them represent each data using only one pulse.

### TAP files

The TAP format was created by Per Håkan Sundell in 1995. This file format stores a digital image of the data stored in the magnetic tapes. Usually, the .tap extension is used. The TAP files are divided into two parts: a header with information about the Commodore computer model, video mode, tap version and length of the data; and the data itself. The data section contains the information about the pulse lengths to play.

Figure 7 shows an example of a TAP file header.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000h:	43	36	34	2D	54	41	50	45	2D	52	41	57	01	00	00	00
00000010h:	02	F5	0A	00	30	30	30	30	30	30	30	30	30	30	30	30
00000020h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30

Figure 7. TAP file header

This header has the following fields:

Position	Data	Description
00-11	Signature	- C64-TAPE-RAW - C16-TAPE-RAW
12	Version	00, 01 or 02
13	Machine	- 00 - C64 - 01 - VIC-20 - 02 - C16/Plus4
14	Video	- 00 - PAL - 01 - NTSC
15		Reserved
16-19	Size	Data block size

From byte 0 to 11 there is a signature that can be C64-TAPE-RAW or C16-TAPE-RAW depending on the machine the TAP file was created on. The C64-TAPE-RAW signature corresponds with the VIC-20, C64 and C128 computers and the C16-TAPE-RAW corresponds to the C16, C116 and Commodore Plus/4 comput-

From byte 0 to 11 there is a signature that can be C64-TAPE-RAW or C16-TAPE-RAW depending on the machine the TAP file was created on. The C64-TAPE-RAW signature corresponds with the VIC-20, C64 and C128 computers and the C16-TAPE-RAW corresponds to the C16, C116 and Commodore Plus/4 computers.

In the byte 12 the tap version is stored. Depending of the TAP version, the pulses and silences have to be coded differently.

The byte 13 contains the Commodore model for which the TAP file is created. If this byte is 00 then the file is created for the C64 and C128 computers. If this byte is 01 then the file was created for a VIC-20 and finally, a value of 02 means that this file was created for a C16/116 or Plus/4 machine.

The byte 14 stores the video system of the target machine. 01 means the PAL system (for Europe users) and 02 means the NTSC system (for USA users).

The 15<sup>th</sup> byte is reserved for future use.

Finally, from byte 16 to 19 the data block size (without the header) is stored in little-endian format.

### The TAP version 0

Each data different to 00 represents the number of machine cycles multiplied by 8 that each pulse has to last. In that way, to calculate each pulse length the next equation must be applied:

$$Pulse\_length = Data \cdot \frac{8}{frequency}$$

where Data is the byte read from the TAP file and frequency is the clock speed of the Commodore computer that is defined using the information of the header according to the next table:

Computer	Video mode	Frequency (Hz)
VIC-20	PAL	1108405
	NTSC	1022727
C64 and C128	PAL	985248
	NTSC	1022727
C16/C116 and Plus/4	PAL	886724
	NTSC	894886

If, for example, a 0x30 value (in hexadecimal format, which is a decimal value of 48) is read from a TAP file created for a PAL Commodore 64 which has a frequency of 985248 Hz, the pulse width will be:

$$Pulse\_length = 48(\$30) \cdot \frac{8}{985248} = 389.74\mu s$$

This value means that for the first half of the pulse length a value of 0 will be transmitted and in the other half a value of 1 will be transmitted. Figure 8 depicts this situation

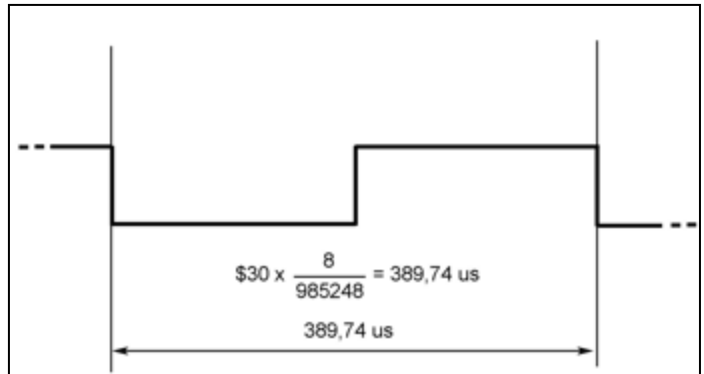


Figure 8. TAP version 0 pulse

If a 00 value is found in the file, it means a long pulse, this is a pulse that is longer than the value that can be coded using one byte (256). In that case, the pulse length is calculated considering that the Data value is 255.

### The TAP version 1

The TAP version uses the same way of coding the data but changes the way that the long pulses are coded. When a 00 value is found in the TAP file, the next three bytes represents the number of clock cycles that the pulse lasts. Figure 9 shows a section of a TAP file. The rectangle marks a long pulse.

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
0050: 56 36 20 2D 54 41 50 45 2D 52 00 57 16 03 45 56
0060: 51 21 00 54 2F 0F 20 0D 31 01 26 0D 07 21 0A 12

```

Figure 9. Example of TAP version 1 long pulse

The period of the pulse is calculated (for a PAL Commodore 64) using the next equation:

$$LongPulse\_length = \frac{995156 (\$0F2F54)}{985248} = 1.010s$$

As can be observed, the three bytes are concatenated using the reversal order and divided by the clock computer frequency. The pulse of this example is depicted in Figure 10.

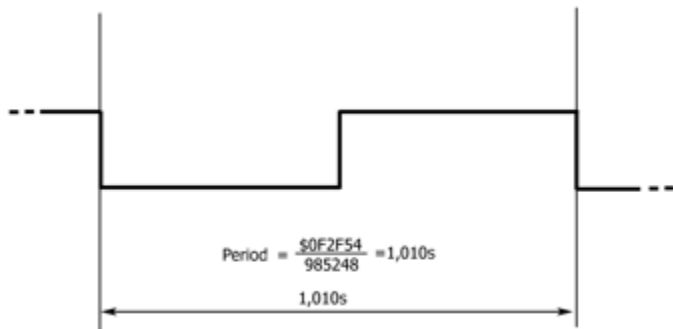


Figure 10. Example of long pulse in a TAP version 1

$$LowPulse\_length = 32 (\$20) \cdot \frac{8}{886724} = 288.70\mu s$$

$$HighPulse\_length = 48 (\$30) \cdot \frac{8}{886724} = 433.05\mu s$$

And the pulse will be as shown in Figure 12.

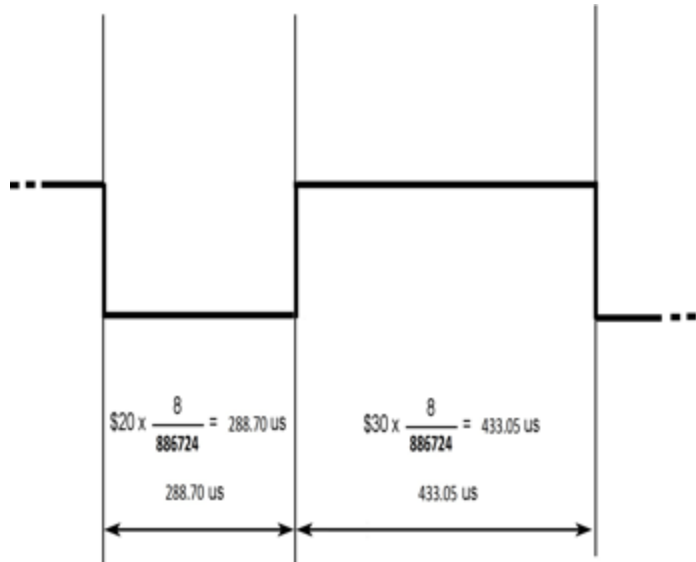


Figure 12. Pulse of a TAP version 2

**The TAP version 2**

The TAP version 2 codes the long pulses as in the TAP version 1 but it changes the codification of the standard pulses. The TAP version 2 codes each semi period using one byte. This method enables it to create pulses where the duration of the low (0) and high (1) part are different. Figure 11 shows an example of TAP version 2 file.

```

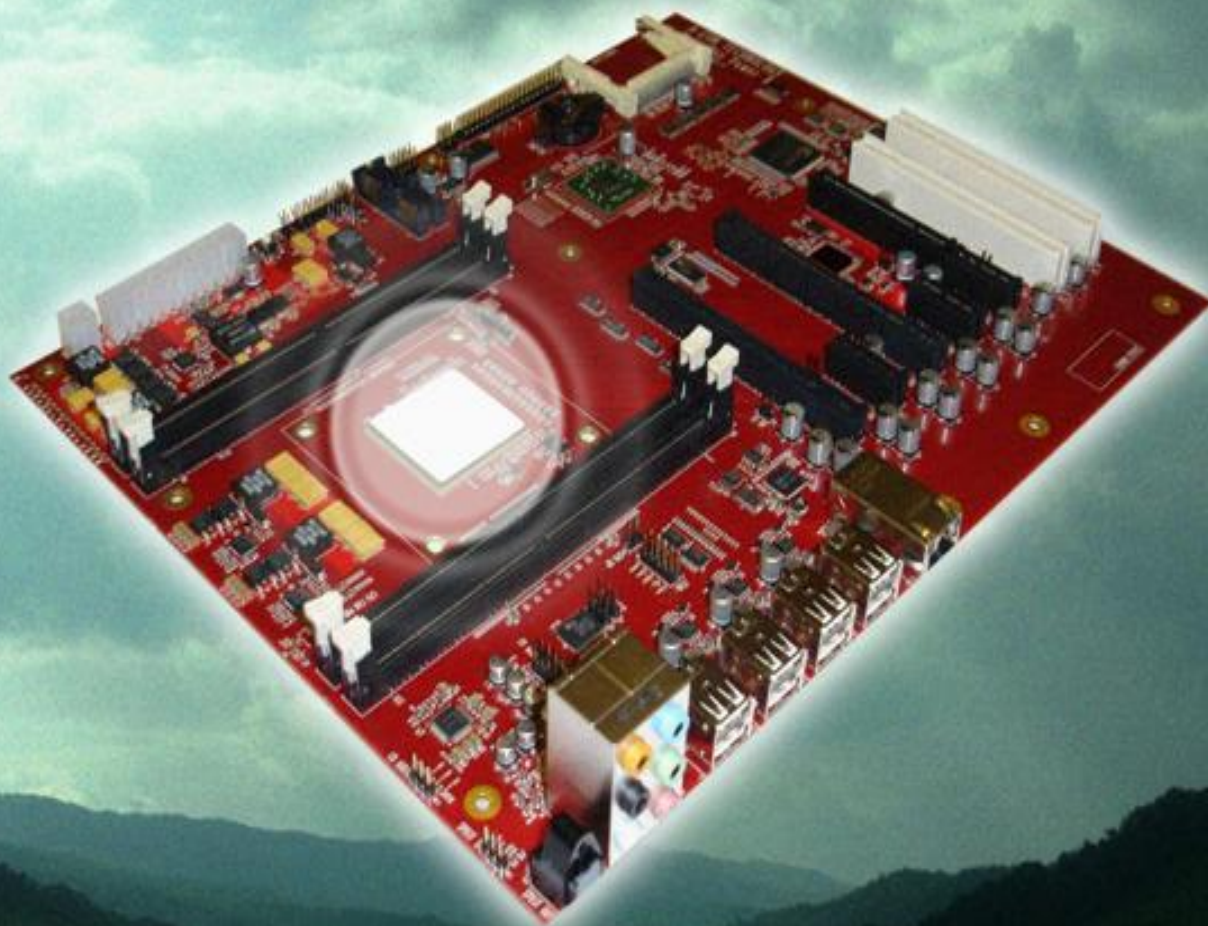
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
0050: 56 36 20 30 54 41 50 45 2D 52 00 57 16 03 45 56
0060: 51 21 00 54 2F 0F 20 0D 31 01 26 0D 07 21 0A 12
    
```

Figure 11. TAP version 2

The length of each semi period will be calculated as the length of the period that was calculated in the TAP version 1. If a PAL Commodore 16/116/Plus 4 is used, the length of each semi pulse of the bytes marked in the Figure 11 will be calculated as:



# AmigaONE



**X10000**  
**I BELIEVE**

**On the Way to Fun:  
An Emotion-Based Approach to  
Successful Game Design**

**Roberto Dillon (Author)**

**Product details**

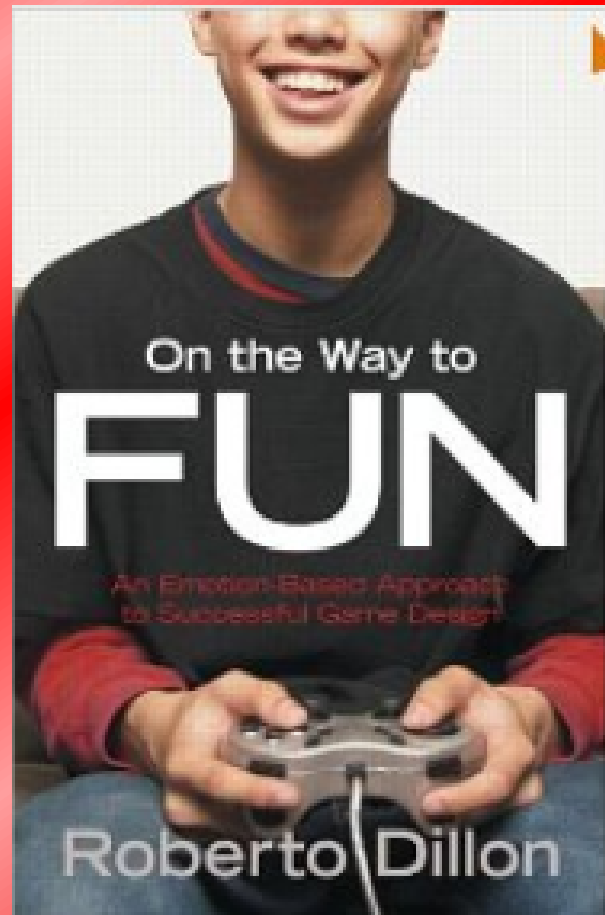
**Paperback: 200 pages**

**Publisher: A K Peters/CRC Press  
(8 Mar 2010)**

**Language English**

**ISBN-10: 1568815824**

**ISBN-13: 978-1568815824**



**Golden Age of Video Games:  
The Birth of a Multi billion  
Dollar Industry [Paperback]**

**Roberto Dillon (Author)**

**Product details**

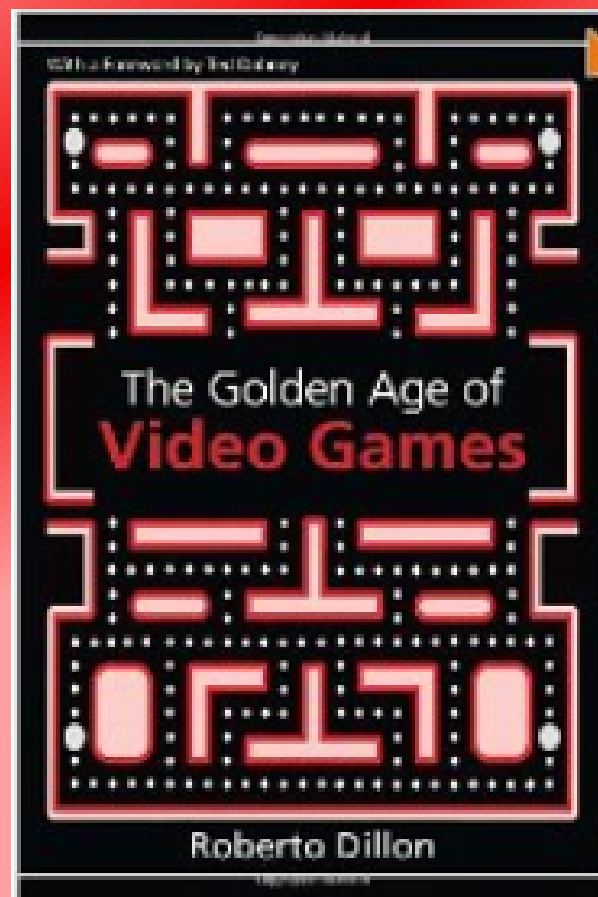
**Paperback: 209 pages**

**Publisher: A K Peters/CRC  
Press; 1 edition (28 April 2011)**

**Language English**

**ISBN-10: 1439873232**

**ISBN-13: 978-1439873236**



# Rediscovering Commodore CP/M

## by: *Commodoreman* ©

### PART 1

My first Commodore was a standard 128. At the time (approx. 1987), it was a used machine that I bought from a friend. I was mostly interested in the Commodore 128 and 64 modes and CP/M mode happened to be only a curiosity. I believe this was due to the fact that I didn't have any software for it other than the system disk. I remember reading about CP/M mode in the System Guide, but really did not get very far.

As time went on, I found a local Commodore club (Mountain Computer Society) and sure learned a lot more about the Commodore 64 and 128, but there was little if anything done with CP/M mode. I did manage to obtain the Digital Research book from a member (This is the one that has the User's Guide, Programmer's Guide, and System Guide) and a genealogy program. However, I didn't get much farther than formatting a few disks and tinkering with a few .com files.

Eventually, CP/M was shelved for some future day and since



then, I focused primarily on the 64 and 128. I would say that the difficulty in access to resources was the biggest reason why I didn't pursue it. Sure, I did have an interest, but that interest eventually faded – I put this project on the “back burner”, so to speak. I remember crossing the paths of bits of information here and there, somewhere I heard there were supposed to be thousands of programs and lots of other resources, but I just didn't see it like I did with Commodore.

With the advent of the internet, I found access to information and resources magnified with great intensity. While working on the 64/128 scene, I would see some things about CP/M here and there. However, CP/M still remained a “future project”.

Enter a new phase...While engaged in educational pursuits, I took a course on business and the internet. This is where I started my website – focusing on Commodore computers. I felt that I could not have a proper website about Commodores and not have something about CP/M. I felt it was only proper to include CP/M 3 (I have pages for programs, software, and programming, albeit with few resources).

I learned the Zilog Z80 processor is an integral part of the Commodore 128. The way the 128 was designed, the Z80 played an important role – being first in control when the computer was turned on. If the system disk was in the primary drive, that would be the system to initiate. Anyway, still to this date, I do not have much in the way of resources for CP/M. I do have access to a few links and managed to find a few programs. I guess it's a start. Perhaps that is why I thought this would make an interesting article. Rediscovering would be a proper title since I have forgotten most of what I had learned before.

Perhaps there are other Commodore computer owners who may feel the same way, that there is an interest, but never really got around to seriously tackling it and not really being able to do anything substantial. Un-shelving this project could be interesting...what could go wrong? What was there to lose? Should be easy, right? There is only one way to find out...

If I have caught your interest, tune in to Part 2. Here is where I'll be starting out again and things just might get to be interesting....

Commodoreman



# Issue 59

## 2012

**Editor**

**Nigel Parker**

**Spell Checking**  
**Peter Badrick**

**Text & HTML Conversion**  
**Paul Davis**

**D64 Disk Image**  
**Al Jackson**

**PDF Design**  
**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 © 2011 Commodore Free Magazine**