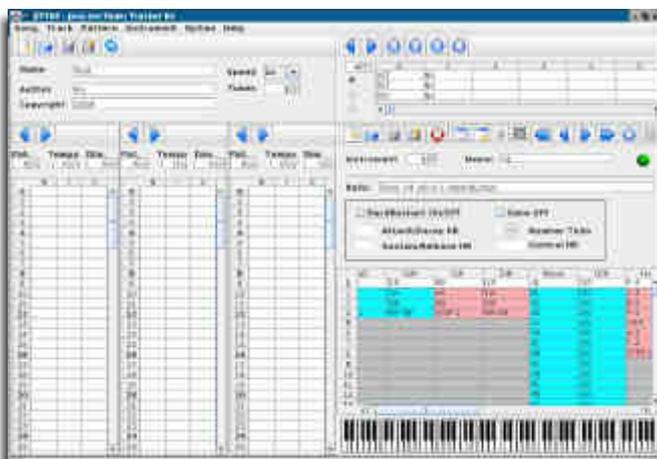


Commodore Free

Issue 22 August 2008

Free to download Commodore magazine
Dedicated to Commodore Computers
Available as PDF Text SEQ HTML and D64 image
www.commodorefree.com



Editor

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

This month I am plugging the Commodore computer club, you will remember from the last issue of Commodore Free that; we had our first meeting last month and to keep momentum going I have listed the membership form and general club rules in this issue of the magazine. A lot of thought has gone into setting up the club. The main point is the club will be run as a none profit organisation, so you can rest assured that any money invested will not be taken by members; we will have visuals on the spend and income listed to all members. Currently I am treasurer with Jason Kelk (tmr)as Webmaster and Alan Bairstow as Chairman (www.commodorescene.org.uk) and Shaun Bebbington from www.micromart.co.uk as secretary

What use is another Commodore club? as we are all very active Commodore users and enjoy promoting the machine we have some surprises in store for members various discounts from our online shop (still to be setup) special members access areas on the website and other benefits'. The main point is bringing people together something I felt the U.k. was lacking. I hope everyone reading would be interested in joining with fee starting from as little as £3, so you can test out the club, but I would think most will opt for a life membership

Please visit the website (still under construction, so you may need a hard hat to enter some areas) you will notice a mirror of the HTML version of Commodore free on the site and the forum is running along very nicely so take a look at <http://www.commodorecomputerclub.co.uk/> as stated last month we have already released a 1 file demo and this is available from the site to download, comments and feedback are welcome, please speak up.

I have asked for readers to send through information and I appreciate you will be reading this and think your comments and tutorials will not be welcome, nothing can be further from the truth, I would love to print tutorials for both beginners and experts alike. We all have particular talents and as a community these should be shared. Or maybe you would like to ask a question; and of course wherever possible I will find the answer, even if I have to "phone a friend"

Commodore is still alive let's keep the memory fresh, I am in the process of setting up a bank account for the Commodore Computer club. While giving details to the operator she said "Commodore are they still going I used to own one of those they were great machines". So even in banking I can't get away from the memories of the machine. Talking of which, if you have a fond memory of Commodore like when you first received your machine send write it down and send it in, some readers hate these but I love reading about your first experiences with Commodore.

If you are a club send me some information I will include the details as and when space allows, also if you are a company or have a product to advertise send me some promotional material and when space permits I will re-print it FREE! you cant get cheaper adverts than that! There is nothing better than a free advert, for promotion of course you may like to send me one of the devices for a full review in the magazine. Speaking of which, reviews are personal, I write the magazine so they are my personal reviews of the products, if an individual writes in they are there reviews and comments, if something is less than satisfactory I will say so, if its the best product in the world it will be reviewed as such. Bribes are welcome but ultimately they do not change the reviews.

Also; I finally had 3 entries to the Commodore music Competition to win a free copy of PRESS PLAY ON

TAPE's current cd. I will at some point print what I consider the best of these 3 entries, the disks have now gone out, thanks to PRESS PLAY ON TAPE for donating the disks and the 3 people who finally put pen to paper, I was however disappointed that more people didn't take part in the competition.

here we go then hold onto your hats it may be a bumpy read, especially with my spelling.

Regards
Nigel www.commodorefree.com

CONTENTS

EDITOR /CONTENT	PAGE 2
NEWS	
GENERAL NEWS	PAGE 3
UIEC Status Update	PAGE 3 - 4
UIEC INFORMATION	PAGE 4 - 7
TUTORIALS	
In the beginning part 7	PAGE 12 - 14
Making music with DMC	PAGE 14 - 19
INTERVIEWS	
STEFANO TOGNON (SIDIN Magazine)	PAGE 8 - 11
CLUB NEWS	
C.C.C. Membership form	PAGE 20
C.C.C. Membership Club rules	PAGE 21 - 22

HOW CAN I HELP COMMODORE FREE

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

WHAT ARTICLES DO YOU NEED

Well they vary, contact me if you have an idea but I am looking for

Tutorials

(beginners and Expert),

Experiences

with Commodore,
Why I love Commodore machines,

Interviews

maybe you have access to a power user

NEWS

NEWS =====

NEW C64 FRAMEWORK SOURCE RELEASED.
I've have just released my new C64 frame work (and sample app) for you to play with. Here is a list of the frameworks features.

- Initialisation, mainloop and interrupts all setup and ready to go.
- Production quality multiplexor.
- Three separate compile time switchable sorts.
- Simple collision detection system.
- General sprite animation system.
- Game based keyboard and joystick routines.
- Simple player control framework.
- Lightning fast 8x8bit multiply routine.
- Simple random number routine.
- Built in MMC64 FAT16 file loader.
- Simple example platform engine showing the framework in action.

You can grab it from the download links on my BLOG <http://daily.blogspot.com>

where you can also read about its development and any future additions there.

=====

SUB HUNTER

Another preview has been released of the game sub hunter, I would say this is one of Richards best games ever, and I personally am looking forward to the finished version with much anticipation. You can of course download a new preview from here <http://noname.c64.org/csdb/release/?id=68965>

and you can read about the games development here http://www.redesign.sk/tnd64/sub_hunter.html

=====

COMMODORE COMPUTER CLUB U.K.

<http://www.commodorecomputerclub.co.uk>
We're now accepting membership applications for the Commodore Computer Club (UK), and you can fill these applications electronically and pay with PayPal. We also accept cheques, postal orders and international money orders. Please contact me for details or for an application form and club rules. My email address is contact@CommodoreComputerClub.co.uk
Regards, Shaun.

More information about the cub is listed at the end of this magazine

=====

UEIC STATUS UPDATE

Subject: uIEC status update

I apologize for the delay in getting this out. Here is an update:

Since July 4th, I've been busy soldering up the remaining uIEC/IDE and uIEC/CF units. I've shipped all but one uIEC/IDE unit out (Wizard, it's yours, but I



need a padded envelope for it), and all but 5 of the uIEC/CF units out. I have updated the design to v2.3. This forced moving the activity LED to another IO pin, to free up pins for the disk swap buttons. All units shipped after C4 EXPO have this mod. They also have the 2008-07 uIEC bootloader. Thus, any unit sold at the C4 EXPO show needs to check with me for a timeframe when I can upgrade their units to v2.3 and install the bootloader and the newer 7.2.1 firmware.

The bootloader is very important, since it allows one to upgrade their firmware simply by copying a file to the root directory of the first partition of the primary IDE/CF drive. On boot, the bootloader will scan for a newer firmware and flash it into the uIEC if it finds one. FAT32 support is enabled now (I held off on shipping until I had that complete, since otherwise, folks would need to have a small FAT16 partition #1 to load new firmware, which seemed a waste) For the unfilled CF units, I plan to make the small IO mods of v2.3 and add a SD card option and run the next set of boards. I should be able to order the boards by next week. It takes 10 days to run a batch of boards, and a few days to solder. Thus, it'll probably be 14-21 days before I can sell more CF units. I only have 2 unfilled orders on the CF units. For the uIEC/IDE units, I need some help. With the v2.3 mods and a desire to add buffer drivers to the IEC lines (requires 4 more IO lines), I'm running too short on IO to offer printer port/IEEE port/parallel speeder port IO. I have two options: Make the minimal mods needed for the IEC lines, add the support for SD cards, and run a batch of boards. This would be quick and requires very little thought. Read: takes little time to prepare to run boards.

Make the larger mods needed to bring out the additional IO (requires the 100 pin version of the CPU I am using and some headers). Read, takes more time. 'm interested in feedback. I think 50 boards is

the smallest prod run I can economically run, so I'd need to ensure folks are OK with the decision. Right now, after mailing all the uIEC/IDE boards, I have 19 unfilled orders. I'm leaning towards option #1, mainly for timing, but I thought I'd ask the group. The uIEC/CF units were designed mainly for folks who plan to "embed" them in designs. As such, they do not have elegant connections. I'm not apologizing for that, just noting why the IEC wires and power wire are installed as they are. My intention was for the users to embed the unit and unsolder those wires when no longer needed. I will, though, apologize for the cassette connector. I forgot to plan for the little PCB that's needed for that connector. I will run a batch of those with my next order. If folks want one, I can ship one or solder one up when I see you next. Right now, there are two "bugs" I am working on. One is the start-up latency on uIEC/IDE units when only 1 drive is connected. Due to the way IDE devices are initialized, the initialization sequence must wait > 32 seconds to see if a drive is present. Thus, on a setup with only 1 drive, there will be a 30 second delay on start. It's not a "bug" per se, but unwanted behaviour. To counteract, I am working on a way to disable the second drive via DOS command. Since we know later designs might have more than 1 type of storage medium (ATA/CF/SD/etc.), I am making it flexible, so it's taking a bit longer to implement

The other bug is init issues with master CF card and slave IDE drive. I think it's just a timing issue. For now, I recommend putting the IDE drive as primary, and CF card as slave. I've set up a mailing list for the unit (questions, concerns, etc.) It's unmoderated at present, so play nice. Details are below. After I get the designs to the PCB house, I plan to work on image support. I have WIP code for D71/D81/DNP images that needs to be finished and committed. Although I tend to set my own priorities, I'm always interested in which features folks want.

Documentation: Yes, I do plan to write some up, but I simply need to get the units shipped first. Since the drive uses CBM and CMD syntax, I hope folks can easily find their way around. I will, though, send a cheat sheet to the mailing list in a few days. For the uIEC/CF owners, the two small pins on the backside of the unit are the disk-swap switches. Ground each (via a switch) for operation. I'm sure I'm leaving something out, but there's always more email to send.
Jim

Membership to the Mailing List uiec-users@jbrain.com

On this web page you can subscribe to, or unsubscribe from, the mailing list uiec-users@jbrain.com

UIEC INFORMATION

The device is now shipping and Jim has kindly let me reprint the draft of the read me guide I personally have one of these on order and will give a full and honest review as soon as the device arrives for my testing.

UIEC here is the most current README:

sd2iec - a controller/interface adapting storage devices to the CBM serial bus Copyright (C) 2007,2008 Ingo Korb <ingo@akana.de> Parts based on code from others, see comments in main.c for details. Jiffy-Dos send based on code by M.Kiesel Fat LFN support and lots of other ideas by Jim Brain crc7.c generated by pyrcr, see comments in it for pyrcr licence. Final Cartridge III fastloader support by Thomas Giesel

Free software under GPL version 2 ONLY, see comments in main.c and COPYING for details.

FIXME: This file still needs to be expanded. A lot.

INTRODUCTION:

sd2iec is firmware, used in hardware designs like MMC2IEC, SD2IEC, or uIEC, that allows the Commodore serial bus to access removable storage devices (MMC, SD, CF) - think of it as a 1541 with a modern storage medium instead of disks. The project was inspired by (and uses a few bits of code from) MMC2IEC[1] by Lars Pontoppidan and once ran on the same hardware before it grew too big for the ATmega32 used there.

Currently, the firmware provides good DOS and file-level compatibility with CBM drives, but much work still remains. Unless specifically noted, anything that tries to execute code on the 1541 will not work, this includes every software fastloader.

[1] Homepage: <http://pontoppidan.info/lars/index.php?proj=mmc2iec>

Supported commands:

- General notes: Any command not listed below is currently not supported.

- Directory filters: To show only directories, both =B (CMD-compatible) and =D can be used. On a real Commodore drive D matches everything. To include hidden files in the directory, use *=H - on a 1541 this doesn't do anything. sd2iec marks hidden files with an H after the lock mark, i.e. "PRG<H" or "PRG H".

CMD-style "short" and "long" directory listings with timestamps are supported ("=\$=T"), including time-stamp filters. Please read a CMD manual for the syntax until this file is updated.

- Partition directory: The CMD-style partition directory (\$=P) is supported, including filters (\$=P:S*). All partitions are listed with type "FAT", although this could change to "NAT" later for compatibility.

- CD/MD/RD: Subdirectory access is compatible to the syntax used by the CMD drives, although drive/partition numbers are completely ignored.

Quick syntax overview:

```
CD:_      changes into the parent dir (_ is the left
arrow on the C64)
CD_      dito
CD:foo   changes into foo
CD/foo   dito
CD//foo  changes into \foo
CD/fool:bar changes into fool\bar
CD/fool/bar  dito
```

You can use wildcards anywhere in the path. To change into an M2I or D64 image the image file must be named after the ., it will not be recognized otherwise.

MD uses a syntax similar to CD and will create the directory listed after the colon (:) relative to any directory listed before it.

```
MD/fool:bar creates bar in fool
MD//fool:bar creates bar in \fool
```

RD can only remove subdirectories of the current directory.

```
RD:foo  deletes foo
```

CD is also used to mount/unmount image files. Just change into them as if they were a directory and use CD:_ (left arrow on the C64) to leave. Please note that image files are detected by file extension and file size and there is no reliable way to see if a file is a valid image file.

- CP, C<Shift-P> This changes the current partition; see "Partitions" below for details.

- C: File copy command, should be CMD compatible. The syntax is

```
C[partition][path]:targetname=[[partition][path]:]source
name[,[[p][p]:]sourcename...]
```

You can use this command to copy multiple files into a single target file in which case all source files will be appended into the target file. Parsing restarts for every source file name which means that every source name is assumed to be relative to the current directory. You can use wildcards in the source names, but only the first file matching will be copied.

Copying REL files should work, but isn't tested well. Mixing REL and non-REL files in an append operation isn't supported.

- G-P Get partition info, see CMD FD/HD manual for details. The reported information is partially faked, feedback is welcome.

- N: Format works only if a D64 image is already mounted.

- R Renaming files should work the same as it does on CMD drives, although the errors flagged for invalid characters in the name may differ.

- S: Name matching is fully supported, directories are ignored. Scratching of multiple files separated by , is

also supported with no limit to the number of files except for the maximum command line length (usually 100 to 120 characters).

- T-R and T-W If your hardware features RTC support the commands T-R (time read) and T-W (time write) are available. If the RTC isn't present, both commands return 30,SYNTAX ERROR,00,00; if the RTC is present but not set correctly T-R will return 31,SYNTAX ERROR,00,00. Both commands expect a fourth character that specifies the time format to be used, T-W expects that the new time follows that character in exactly the format returned by T-R with the same format char.

The possible formats are:

- "A"SCII: "SUN. 01/20/08 01:23:45 PM"+CHR\$(13)
The day-of-week string can be any of "SUN.", "MON.", "TUES", "WED.", "THUR", "FRI.", "SAT.". The year field is modulo 100.

- "B"CD or "D"ecimal:

Both these formats use 9 bytes to specify the time. For BCD everything is BCD-encoded, for Decimal the numbers are sent/parsed as-is.

Byte 0: Day of the week (0 for Sunday)

1: Year (modulo 100 for BCD; -1900 for Decimal, i.e. 108 for 2008)

2: Month (1-based)

3: Day (1-based)

4: Hour (1-12)

5: Minute (0-59)

6: Second (0-59)

7: AM/PM-Flag (0 is AM, other is PM)

8: CHR\$(13)

When the time is set a year less than 80 is interpreted as 20xx.

- U0 Device address changing with "U0">+chr\$(new address) is supported, other U0 commands are currently not implemented.

- U1/U2/B-R/B-W Block reading and writing is fully supported while a D64 image is mounted.

- B-P Supported, not checked against the original rom at all.

- UI+UI- Switching the slightly faster bus protocol for the VC20 on and off works, it hasn't been tested much though.

- UI/UJ Soft/Hard reset - UI just sets the "73,..." message on the error channel, UJ closes all active buffers but doesn't reset the current directory, mounted image, swap list or anything else.

- U<Shift-J> Real hard reset - this command causes a restart of the AVR processor (skipping the boot-loader if installed). <Shift-J> is character code 202.

- X: Extended commands. If you use JiffyDOS, you can send them by using @"X..." - without quotes you'll just receive an error.

- XEnum Sets the "file extension mode". This setting controls if files on FAT are written with an x00 header and extension or not.

Possible values for num are:

- 0: Never write x00 format files.
- 1: Write x00 format files for SEQ/USR/REL, but not for PRG
- 2: Always write x00 format files.
- 3: Use SEQ/USR/REL file extensions, no x00 header
- 4: Same as 3, but also for PRG

If you set mode 3 or 4, extension hiding is automatically enabled. This setting can be saved in the EEPROM using XW, the default value is 1.

For compatibility with existing programs that write D64 files, PRG files that have D64, D41, D71, D81 or M2I as an extension will always be written without an x00 header and without any additional PRG file extension.

- XE+/XE- Enable/disable extension hiding. If enabled, files in FAT with a PRG/SEQ/USR/REL extension will have their extension removed and the file type changed to the type specified by the file extension - e.g. APPLICATION.PRG will become a PRG file named "APPLICATION", "README.SEQ" will become a SEQ file named "README". This flag can be saved in the EEPROM using XW, the default value is disabled (-).

- XB+/XB- Enable/disable free block calculation for FAT32 drives. As the free block calculation for FAT32 can take a lot of time it can be disabled using XB-. If it is disabled, sd2iec will always report "1 BLOCKS FREE" for FAT32 drives. The free block calculation on FAT12/FAT16 isn't affected because it takes just two seconds at most. This flag can be saved in the EEPROM using XW, the default value is enabled (+).

- XJ+/XJ- Set or reset the JiffyDOS protocol support flag. This flag can be saved in the EEPROM using XW, the default value is enabled (+).

- X*+/X*- Enable/disable 1581-style * matching. If enabled, characters after a * will be matched against the end of the file name. If disabled, any characters after a * will be ignored. This flag can be saved in the EEPROM using XW, the default value is enabled (+).

- XCnum Set oscillator calibration value to num (must be between 0 and 255). Default is whatever your chip defaults to. This value can be saved in the EEPROM using XW.

- X X without any following characters reports the current state of all extended parameters via the error channel, similar to DolphinDOS. Example result: "03,J-:C152:E01+:B+:*+,08,00" The track indicates the current device address.

- XS:name Set up a swap list - see "Changing Disk Images" below.

XS Disable swap list

- XW Store configuration to EEPROM This command stores the current configuration in the EEPROM. It will automatically be read when the AVR is reset, so any changes you made will persist even after turning off the hardware.

The stored configuration includes the oscillator calibration value, the JiffyDOS protocol support flag, the

extension mode and the current device address. If you have changed the device address by software, sd2iec will power up with that address unless you have changed the device address jumpers (if available) to a different setting than the one active at the time the configuration was saved. You can think of this feature as changing the meaning of one specific setting of the jumpers to a different address if this sounds logical enough to you.

The "hardware overrides software overrides hardware" priority was chosen to allow accessing sd2iec even when it is soft-configured for a device number that is already taken by another device on the bus without having to remove that device to reconfigure sd2iec (e.g. when using a C128D).

- X? Extended version query This command returns the extended version string which consists of the version, the processor type set at build time and the suffix of the configuration file (usually corresponds to the short name of the hardware sd2iec was compiled for).

- M-R, M-W, M-E Memory reading returns random data of the requested length. Memory writing knows about the address used for changing the device address on a 1541 and will change the address of sd2iec to the requested value. It will also check if the transmitted data corresponds to any of the known software fastloaders so the correct emulation code can be used when M-E is called.

- E-R, E-W Both commands work like M-R and M-W, but instead of reading/writing RAM they allow access to a user-area of the EEPROM. This area currently holds 512 bytes and any access beyond this area will result in a 32 SYNTAX ERROR. It is strongly recommended to work on a protocol for sharing this area between multiple applications that want to store their configuration in there, but that is beyond the scope of this project.

As the contents of the EEPROM have to be copied to RAM before they can be sent to the computer, it is not possible to read more data with a single command than the error message buffer (default size: 36 bytes) can hold. Similarly, writing is restricted by the size of the command buffer (at least 42 bytes for compatibility, expected to be at least 100 bytes in release versions).

The user-area does not interfere with the stored configuration (XW) in any way.

Long File Names:

Long file names (i.e. names not within the 8.3 limits) are supported on FAT, but for compatibility reasons the 8.3 name is used if the long name exceeds 16 characters. If you use anything but ASCII characters on the PC or their PETSCII equivalents on the Commodore, you may get strange characters on the other system because the LFN use unicode characters on disk, but sd2iec parses only the low byte of each character in the name.

Partition:

sd2iec features a multi-partition support similar to that of the CMD drives. The partitions (which may be on separate drives for some hardware configurations) are accessed using the drive number of the commands sent from the computer and are num-

bered starting with 1. Partition 0 is a special case: Because most software doesn't support drive numbers, or always sends drive number 0, this partition points to the currently selected partition. By default, accesses to partition 0 will access partition 1, this can be changed by sending "CP<num>" over the command channel with <num> being an ASCII number from 1 to 255. "C<Shift-P" (0x42 0xd0) works the same, but expects a binary partition number as the third character of the command.

Software fastloaders:

Turbodisk

Turbodisk is detected by the CRC of its 493 byte long floppy code and the M-E address 0x0303. The same code seems to be used under various names, among them "Turbodisk" (both 2.1 and 2.2) and "Fast-Load". Unfortunately the timing requirements are extremely tight and cannot be met with the internal RC oscillator of the AVR even if calibrated. You really need an external 8MHz crystal or the data read by the C64 will be gibberish. It is not known if there is an NTSC-compatible version of this fastloader.

Final Cartridge III

Both the fast loader and the fast saver of Final Cartridge III are supported. The fast loader seems to work without a crystal, the fast saver was not tested. It is not known if there is an NTSC-compatible version using the same drive code, but at least for the fastloader the FC3 rom seems to contain two different C64-side implementations.

Action Replay 6

The AR6 reads a byte from the drive rom to check which fastloader it should use. sd2iec returns a value that should force the cartridge to use the standard kernel loader instead of its (still unsupported) fastloader.

Dreamload

Dreamload uses direct track/sector access, so it is only supported on D41 or similar disk image formats. As sd2iec has to wait for commands from the C64 constantly the disk change buttons may become unresponsive, try multiple times if you need to. Dreamload is a "captive" fastloader" sd2iec stays in Dreamload mode until it receives a "quit loader" command from the C64. To force sd2iec to resume normal operation, hold the disk change button until the red LED turns on (just like sleep mode).

Please note that Dreamload does not work with more than one device on the serial bus due to the way it uses the ATN line.

JiffyDOS:

The JiffyDOS protocol has very relaxed timing constraints compared to Turbodisk, but still not as relaxed as the standard Commodore IEC protocol. Jiffy seems to tolerate slightly mis-tuned RC oscillators, but you still shouldn't expect it to work without oscillator calibration. If the frequency error is too big you WILL get "wrong data" which usually manifests as a FILE NOT FOUND error, because the name the drive received was already garbled.

x00 files:

P00/S00/U00/R00 files are transparently supported, that means they show up in the directory listing with their internal file name; instead of the FAT file name.

Renaming them only changes the internal name. The XE command defines if x00 extensions are used when writing files, by default sd2iec uses them for SEQ/USR/REL files but not for PRG. Parsing of x00 files is always enabled even when writing them is not.

x00 files are recognized by checking both the extension of the file (P/S/U/R with a two-digit suffix) and the header signature.

Disk Images:

Disk images are recognized by their file extension (.D64, .D41) and their file size (must be one of 174848, 175531). If the image has an error info block appended, it will be used to simulate read errors. Writing to a sector with an error will always work, but it will not clear the indicated error.

M2I files:

M2I files are fully supported. sd2iec supports SEQ and USR files in this format in addition to PRG and DEL which were already implemented in MMC2IEC. For compatibility reasons the file type is not checked when opening files. Inside an M2I file the files are always shown as 0 (DEL) or 1 blocks; because calling stat for every file was slowing down the directory listing too much. For compatibility with existing M2I files the data files do not use P00 headers even when the file type is SEQ or USR.

Changing Disk Images

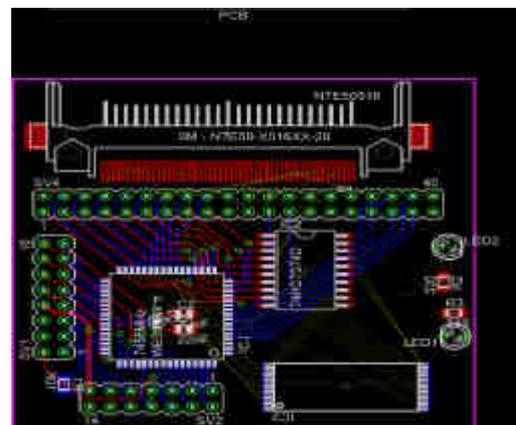
Because some programs require more than one disk side there is support for changing the currently mounted disk image with a button connected to the disk change pin.

If your circuit doesn't have a disk change pin/button you might be able to add it yourself:

- For the original MMC2IEC and the NKC MMC2IEC: Connect a button from PA4 to ground. PA4 is pin 36 on the DIL version of the controller or pin 33 on the surface-mount version.

- For Shadowolf's MMC2IEC 1.x PCBs: Connect a button from PC4 to ground. PC4 is pin 23 on the DIL version of the controller or pin 23 on the surface-mount version.

- Any other circuit without disk change pin on a convenient connector somewhere and no button dedicated to that function: Please check with the supplier of the board, and read config.h in the sources to find out how to connect it.



To use this functionality, create a text file that lists the file names of all disk images you want to swap between, one per line. The file names are parsed in the same way as the CD command, so you can include a path to the image if desired.

Examples:

```
=== example 1 ===
FOO.D64
BAR.D64
BAZ.D64
=== end of example 1 ===
```

```
=== example 2 ===
//NEATGAME:/DISK1A.D64
//NEATGAME:/DISK1B.D64
//NEATGAME:/DISK2A.D64
//NEATGAME:/DISK2B.D64
=== end of example 2 ===
```

The swap list is enabled by sending "XS:filename" over the command channel with filename being the name of the image list you created, parsed in the same way as any other file name.

After sending XS the first image in the list is automatically mounted. To switch to the next image in the list, push the button. If the new image was mounted successfully both LEDs will blink twice. When you've reached the last image in the list pushing the button will mount the first image again. All of this is completely compatible with normal image mounting/unmounting, so you can unmount the disk image any time you want, and resume the mount cycle later by pushing the button.

Due to the way this feature is implemented you are not limited to a swap list containing just D64 images, M2I and even FAT directories will work too. FIXME: Does that still work?

If you press the button when no list has been set before, or when the previous list was cleared by sending XS the software will look for a file called AUTOSWAP.LST in the current (FAT-) directory and use this as the current swap list until you deactivate it or manually change the directory (otherwise an AUTOSWAP.LST in the new directory would be ignored until you send XS, killing the nice "it just works" feeling).

FIXME: Integrate the following into the preceding

The second disk change button is on
 - PA5 for LarsP/NKC
 - PC3 for Shadowolf's MMC2IEC 1.x
 - PC2 for Shadowolf's sd2iec 1.x ("Reserve" on the header)
 - PG4 for uIEC

Either of those buttons will trigger the use of AUTOSWAP.LST. If a swap list is already active, the first button will switch to the next image in the list, the second button will switch to the previous image in the list and pushing both buttons together will switch to the first image in the list.

The confirmation blink is red+green followed by green for "next", by red for "previous" and by red+green for "first".

Sleep Mode:

If you hold the disk change button down for two seconds, sd2iec will enter "sleep mode". In this mode it doesn't listen to the bus at all until you hold down the disk change button for two seconds again which resumes normal operation. Sleep mode allows you to keep sd2iec connected to the serial bus even when you load something from a different drive that uses a fast loader that doesn't work with more than one device on the bus.

While sleep mode is active, the red LED will be on and the green LED will be off.

Other important notes:

- File overwrite (@foo) is implemented by deleting the file first.

- File sizes in the directory are in blocks (of 254 bytes), but the blocks free message actually reports free clusters. It is a compromise of compatibility, accuracy and code size.

- If known, the low byte of the next line link pointer of the directory listing will be set to (file size MOD 254)+2, so you can calculate the true size of the file if required. The 2 is added so it can never be mistaken for an end marker (0) or for the default value (1, used by at least the 1541 and 1571 disk drives).

- If your hardware supports more than one SD card, changing either one will reset the current partition to 1 and the current directory of all partitions to the root drive. Doing this just for the card that was changed would cause lots of problems if the number of partitions on the previous and the newly inserted cards are different.

Compilation notes:

sd2iec is set up to be compiled in multiple configurations, controlled by configuration files. By default the Makefile looks for a file named 'config', but you can override it by providing the name on the make command line with "make CONFIG=filename". If you are using *BSD you may have to edit the Makefile to use another awk implementation instead of gawk - unfortunately WinAVR compatibility requires using gawk in there.

An example configuration file named "config-example" is provided with the source code, as well as abridged files corresponding to the release binaries. If you want to compile sd2iec for a custom hardware you may have to edit config.h too to change the port definitions.

INTERVIEW WITH STEFANO TOGNON

SIDin magazine Editor

<http://digilander.libero.it/ice00/tsid/sidin/index.html>



COMMODORE FREE

Please would you introduce yourself to our readers?

Stefano Tognon

I'm Stefano Tognon and I live in Italy. I'm 35 years old and I'm an informatics engineer working into a farm as a programmer, system administrator and DBA maintainer. In the farm I program in Java, SQL and some custom scripting language: all computers here are installed with Windows. I always use Linux at home (and have done so since 1999), I programme linux in Java, C++,Perl, and cross assembler for the C64 part. I'm an open source person so I have no problem in sharing my code. I like to assemble computers from components (maybe a good future work...) and install them with Linux ;)

However my dreaming job would be to programming games, this is not so simple to achieve in Italy and I love my country to much to leave. I like Sid music (the only music I really listen to - my listening mileage



is at 120 days now, I listen while jogging looking at nature to cancel the working stress, I read many books about physic, math and science. I even like all sort of sports and F1. My parents have a farm with vineyards and I help them when possible.

Maybe someone in the net already me as ice00 of Ice Team (my group). Ice stay for Informatics Computer Engineer 00 is for the millennium bug remembrance...

CF. Can you tell our reader about SIDin magazine and about your website

ST. SIDin is a PDF magazine about the SID chip (from the Inside) (this is where the magazine's name came from), so expect to see some assembler code

in its pages ;) It is free to download and the purpose was to have two issues each year, but some numbers have been skipped in the last years....mmmm...I have too many activity around...The magazine has a news section containing what I find out about in the sid world from the previous issue, an interview with a musician/composer, and two (technical) articles. In the articles I speak about music routines and techniques on how to program the sid, how to rip a tune from a game, review of music editors and trackers, hardware cards, and music related projects.

The website (<http://digilander.iol.it/ice00>) is a portal to all my activities around the C64 or other projects I did in my free time. Nothing special in the site, just pure html code written by hand and the layout is a little similar to some C64 3D platform games. The background is made with Commodore related images

to give the right atmosphere. However you will find all my releases and even the source code of them.

CF. What would our readers recognise you for, apart from the magazine, for example have you written any Commodore demos or tunes

ST. I wrote two little demos: they took place at the Christmas Online Compo many years ago. Nothing technically special, they were created just for fun. The first is "Silent Night", a demo where there is a full screen Christmas image (just a little conversion in multicolor mode of a card) and the border color pixels changes according to the music. The main task here was to open all the horizontal borders. The second is "FD", where there is a better high resolution Christmas image that I hand created myself with a minor animation of snow, fire and light.

I created an image for an online Pixel Compo. It is a high resolution image of a Commodore-F1 machine. It is a very low quality production, I know, but only after having tried it yourself you can then find Leon's pictures are absolutely fantastic!! I wrote 13 tunes for the C64: one is from 86/87 and even if it's of poor quality by today's sid standard, it gives me the stimulus to learn computer music programming. Only 3 of those tunes are my compositions all the others are cover/remix's of other peoples tunes. Lot of those tunes where specially composed for online compos. Others are just for testing the sid sound. I think that some readers may know me form my minigames. I released this for the Minigames Online Compos. It took place to 6 editions producing 11 games. I like to program games and I find the minigame a big challenge as you have to reduce the size of the code as much as possible. For those games I have the help

of talented musician as Luca/Fire, Nata and Richard Bayliss. Maybe someone knows of Little Sara Sister (1 and 1.5), Haras, Elav, MArkanoid and Ann-Gong games. I have not finished Little Sara Sister 2, this features music from Dustbin and graphics logo from Luca/Fire. Maybe one day I will find the time to complete this title. I have another big game in development that many Italian people already know (it was to be for the previous minigame compo, but maybe it will be in the next one): a Frozen Bubble port to the C64. Something very hard to achieve. Many people know me for having held three editions of Sid-Wine online compo (maybe something unique for compos) and two Tiny Sid Compo (compo about writing tunes with only 256 or 512 bytes). Others may know me for having produced the High Voltage Music Engine Collection (HVMEC) <http://digilander.libero.it/ice00/sid/hvmec.html>

a big collection of many of the tools around for making sid music with the C64. Actually there are over 260 tools in it, but a lot are in my (soon to be released) archive. Fortunately Stephan Parth is helping me with this its a very time consuming task. Very few people know that I developed a C64 emulator (C64A) in 1994, mostly because I did not release it because it was in too early a state (however I found recently the source code and I will release it before the code gets lost). I even developed JC64, a Java C64 emulator in 1999, that it is available at Sourceforge site. It is still in a wait state, because it is totally based on a Vic Article for the emulation, but I find that the article is incomplete and you cannot emulate by following this article. From time to time I add more code to it, but I'm hoping that the true working of the Vic will be documented one day...

I even released the Time Sid Manager library that integrates with sid player for storing the listening to statistics about your sids. Recently I developed Linux Xsidplay as Michael left his development some years ago, I sued this sid player and of course found no more updates. It is at <http://sourceforge.net/projects/>



xsidplay2

Xsidplay has some new features like SDL support and the ALSA is being added (even a Windows port is available) and many more features are soon to come. The last project is JITT64 (Java Ice Team Tracker): <http://jitt64.sourceforge.net/>

This is my attempt to create my music routines and an editor for it. The programme uses Java. I'm moulding it as I need, without taking care of raster-time. I want to produce sid sounds without any restriction (for example one instrument definition can take up to 2KB of data). I have been working on the project from 6 months already, and I think that before

the end of the year it can reach the state where tunes can be produced using it...

I instead think that no one knows of this my embryonic project: <http://regame.altervista.org/>
It born as a paper magazine about C64 games, to be released free with 70dpi resolution (like for screen viewing) and on CD (with just a symbolic cost) for 600dpi resolution; as it is created using semi-professional paper related tools (actually 170MB of material for the first half number). The main problem is to find someone to help in creating reviews, and finding an Italian to English translator. Without this the project could stay in this state for ages :(

CF. The magazine is quite technical what level of user is this aimed for?

ST. I think a mix of programmer and musicians, most of the articles I show how the chip is programmed to produce a given sound. I try to give a little example of this point. For a musician that wants to produce and arpeggio, the only thing that he wants to know is that he has to put in sequence, at a given rate of 3 notes values like D E F into his editor /tracker. This should not be a problem for him. For a programmer, he has to know what is the effect to produce an (arpeggio) and then he has to write the appropriate code (e.g. converting the notes to hi/lo frequency values, and put them at the right rates in the sid registers 0 and 1 of one voice). However if the reader is not in one of the above categories, he can read the article descriptions so he has an idea of how that effect is generated by the SID. Also, I think that the articles like ripping a tune from a game (even though this is quite technical), could ever give a stimulus to other reader to try this from themselves. That could be a good positive for the sid community.

CF. The Magazine Covers just sound on Commodore 64 machines, and trackers/ editors, have you thought about covering any other items of Commodore related news?

ST. Yes and no ... as the magazine is based on my activities, it essentially to reflect this in the magazines pages. I do something today and it follows afterwards



in the magazine. I don't know for example, if the next issue will be formed in all these activities. Most depends of what I will do in the next months... For example, I have looked many times to try and buy a Plus4 computer without any success. I'm sure that if I have one, I will test the TED sound capability compared to the SID and an article will pop up for sure even if don't know all the Plus4 features. Also, recently I found that the AY-chip of a ZX SPECTRUM has some sound similarities to the SID: for example the "escape from the spaceship" tune at <http://www.8bitpeoples.com/discography/8BP084>

Maybe a comparison about AY and the SID chip could be a possibility for a future article, a way for me to learn how AY chip works.

CF Some articles cover ripping tunes from games etc, is this a passion of yours?

ST. yes, and this comes from ages ago, even though it was in another form... In around 1989/90 I found some computer viruses in our school computers. I started to study them (virus were some sort of mysterious programs at that time, and so not much information was available about them). The study consisted of disassembling the peace of code that was responsible of the infection and reverse engineering it, to understand how it worked. As I liked to program in assembly languages, this reverse task was very passionate for me. I have developed a certain ability or better a familiarity with this work. I remember that having new virus to analyze I would swap floppy disks using traditional mail with other people by answering magazine announcements (there was no internet available at that time...). All this was because I developed an antivirus (SVDL and MVDL) product and I wanted to make it grow. Unfortunately due to the difficulties found in virus and to add to the antivirus, I had to terminate this kind of (reverse engineering) work, until...I find the Big Sid Hunt many-many years after!! The Big Sid Hunt was (actually it is not updated now) a list of user requests for C64 tunes from games.

Looking at this list I found that some games like Lazarian and Dig-Dug were not ripped into PSID files. That was amazing as the games were very old and even because Lazarian was the first game I played on the C64: I cannot stay without a PSID file of them! So I soon took up the challenge of ripping them, using the way I knew best: disassemble the code and reverse engineering it like I did for the viruses. Doing this I have found again, the good vibe feeling (I even programmed my own utility in Java for helping me in this task). Maybe this wasn't the best way of ripping, as it took a lot of time, but for sure it give a true view of the music from the inside the sid chip. I always like to see how the sid chip is programmed for producing such wonderful sounds. LaLa gave me a music CD as a present for my work in ripping tunes for the Big Sid Hunt some years ago. Thanks.

CF. Do you think Beginners would benefit from reading some of the articles?

ST. Yes, not all the articles are technical like we discuss before. In some articles I describe music editors and this is done from a user perspective (maybe only some parts could be a little more complex like in Dig-organizer), so I think that those articles, give the stimulus to the reader to test the program (and maybe to produce some tunes). In the other articles, maybe the beginner could however have an idea about what I'm talking about, in future when he gets interested about technical details, he may like to re-read the article again. I remember that I read and read many times the Rob Hubbard routine explained in C=Hacking when I was learning the sid, and only after some time I found the way it actually worked.

CF. What motivated you to produce such a magazine and give it away FREE?

ST. In the early months of 2002 the C=Hacking magazine <http://www.ffd2.com/fridge/chacking>



released the last issue. Maybe at that time it was not known that other issues will never be produced, but from 1999 the magazine released only one issue a year, instead of 3 like in the previous years. C=Hacking contains many fantastic articles about the sid chip, so having to wait all this time between issue numbers was a little whipping from the point of view of an enthusiastic reader (less from the editor that had to write the issue having little free time...I know this..) At that time I had many music routines reversed engineered near to be completed, some came from ripping games music, other just for seeing how they were made. I was thinking that, although C=Hacking articles would not have been released, maybe some sid users could appreciate them if I released them in some way. This was the birth of the SIDin magazine: its goal to give sid people something that looks at the inside of the sid: It is free as I wanted as many people as possible to learn about sid programming. I learnt a lot from C=Hacking and maybe if it was not free I probably would have never read it. (not for the price, but for the opportunity to know about it). SIDin is free as I hope that people will want to contribute with their own articles. Until now this is has not happened, but if you have a reader with something related to the sid that came from your activities, don't hesitate to contact me and we can include it in the magazine. Your help is welcome.

CF. Do you know how many people read the magazine?

ST. No as I can't access to the statistic of the web server that store the SIDin site. However, I think that there are at least 200 people read the magazine (this is from the average download link clicked at csdb). As a side note, this is double the value of readers I say to my friends for having the permission to publish her photo into the cover of the 11th number... ;) Another side note: number 4 was read by Martin Galway and he spoke to me about how it, this was a wonderful moment for me :) Thank you Martin.

CF. Have you read Commodore Free and what did you think

Yes, I have read every issue: it was very exciting to have a new magazine for the C64 pop up, and I don't miss a issue I have even the first 3 "ghost" numbers). I like to read the interviews: you interview lot of people related to the C64 world and it is a good objective you give to the magazine. The News section is a must read: I always find some news that I missed reading from the net. The Internet is a fast way to get informed, but it is very easy to miss something and not be able to find again even after just a couple of days. This is the same thing that makes me have a news section in SIDin: to let people find important facts, even if they are old in respect to magazine's release date. I like reading in Commodore Free, articles related to Commodore hardware: I like this very much. I would like to thank you for creating Commodore Free magazine that has had a big positive im-

pact to all the C= fans. I know how hard is creating a PDF magazine and having to release it each month (and in many formats, not only in PDF). Thank you :)

CF. What version is SIDin magazine in, I recognise PDF, what is the other versions are available?

ST. The magazine is in PDF for quick view and/or print. But, as I'm an open source person, I have released the magazine in his native format: Openoffice document. Old numbers are saved into SXW (Star Writer format) is the portable format that OpenOffice used in his early years. The latter are all saved into ODT that it is a standard format for office suites, and you can now open it with a large number of (office related) programs.

CF. Have you thought about a Disk image, of the magazine?

ST. Yes. Having a disk magazine that talks about the sid and even letting the reader listen to sid tunes while reading is very good idea. But the magazine cannot have all the listing I put in SIDin and so it would need to take a different approach for the technical questions. This would be for me, a very difficult and time consuming task. I had wondered about creating other C64 disk magazines, with more soft argument, but the work for starting that project was too much (and I don't have actually a true disk drive for testing magazine in the real hardware)

CF. I see emulation plays a big part of your Commodore usage, but do you still utilise real hardware?

ST. I have now a great number of real machines:
3 C64 (6581),
3 C64C (8580)
2 C128 (6581).

I still have my original C64 (that I don't use to preserve it as much as possible). One of the C64, old models is broken but the sid chip is good. I even have a commodore datasette and a Commodore monitor. I always use the real hardware for testing my productions before releasing them. This is an important phase, essential especially for music creations. For example for my "Evil" music that uses some pc samples, I tested some versions of it on the C64 (6581) and I choose the one with a given volume level of sample that sounds better on the real hardware. Even the "Silent Night" tune that uses a strong filter for voice 3 (inaudible into a 6581) was tested on a real C64C (8580), I needed to be sure that the music sounded good. However testing in real hardware is a good choice for none music related projects. I remember a Vice 1.12 bug in the sprite emulation that makes my MArkanoid game run ok on the PC, but when I tested the game on real hardware I found to my surprise (that the sprites were totally wrong)! Even in years to come as emulator improve, if you are across-developer like myself, always try your program on real hardware. This is a good rule! For e.g. if you don't have a real machine for testing, then ask to a friend to test the application on there's. We are all here to help you :)

I use 2 of my C64C at the annual Spresiano Retrocomputing meeting <http://nrgo.altervista.org/> where I realize/test programs with the public look at all the development phases. There are some images

and videos on that site of my hardware and some images are on my website too.

CF. What piece of Sid music do you consider the most technical in terms of musical content or the code used to produce the track

ST. Good question and not simple to answer...Before I give the real answer, I would like to start by showing you that the SID chip only has 3 sid voices (every musician that composes with the sid chip has that limitation). Go to listen to the "Cover" tune of Phobos: it is very simple musically and this is helping me in showing you what I'm looking for. You will hear from the start two instruments are playing, you then recognize that there are more than 3 instruments playing at the same time. This is very common: you have to program a voice of the sid for playing more that one instrument by changing timbre during the execution of the timbre of another instrument: you have, for example, a percussion sound over a bass sound, but when listening to it, you get the illusion that there are two instrument playing the same time. Just with this example we have seen that every sid tune contains some technicality, and we can describe lots of other ticks used in all the sid tunes (maybe if you listen to some tunes from 1982-83 you think the tunes are "poor" compared to tunes coming after this date: this is because at that time the sid was not programmed to use all its capabilities). So, maybe we can consider technical the tunes that play (or try to play) speech that has been generated by sid voices (using ring modulation for example) not by samples. Some examples of this come from JCH with Triangles tunes but there are other examples. Achieving this is very complicated, but maybe this has nothing musical it is an imitation of a human voice.



So, if I have to choose one tune that I consider the most technical in terms of musical content, I would consider "Parallax" by Martin Galway. It is a cover, of course, but it has some unique characteristics in instruments and interpretation of the composition, that make it very suitable for this choice. The Next tune is "Driller" of Matt Gray, due to his atmospheric sound.

CF. If you could change 3 things about the sid chip what would you change?

ST.

Point 1: removing the ADSR bugs. Those bugs are extremely whipping. I think that you have not known the sid chip if you did not find those bug in action. You write you music and suddenly the voice is muted, without any apparent reason!!

Point 2: making the filter a digital part one, so all chips can sound the same. The Filter is implemented by analogical components and due to the variable precision of how a resistor is implemented into silicon (if you study microelectronic like me you already have the answer) every chips sound different when using filter.

Point 3: maybe something like to have a filter for each voice, or to have more voices to use. Too difficult for me to choose between the two cases...

CF. The Commodore sid chip seems to be having a revival with new software and hardware like the prophet 64, new sound trackers and external sound modules featuring the sid chip controlled by modern sequencers. Why do you think people are turning back to the SID chip?

ST. Because sid sound is almost unique! an irresistible sound...I remember that in 1994 I was studying the FM programming of the Adlib and Soundblaster cards of a pc computer for looking at how I could emulate a sid chip using a modern chip. I was shocked to see that with the FM programming I was totally incapable of achieving this task (this is even when you could use samples (generated by emulation of the chip) to reproduce sid sound in sidplayer: using the hardware capability of sound synthesizers of the pc sound card was almost a waste of time. I don't know to much about the new sound cards produced after 1995, but I think that this hasn't changed).

Actually, a lot of external hardware that hold a sid chip (and sometimes even more than one) and with the development of so many new programs for programming the sid, composers can have a fresh environment for producing music that is a "modern way" of creating sound. A modern way but with the charm of an old chip...Maybe there are still people who open a monitor program on the C64 and find inserting new ways for programming directly the sid chip in a manner that was not used 25 years ago. But having a better convent way of accessing the chip with today's hardware/software lets more composer be productive with smaller effort, and so they are coming back to sid music.



CF. Do you think the chip will ever be produced again to cover the needs of "new" SID users?

ST. This is a dream. I think the dream of all C= user around the world The fact that if one want to put a sid chip into

modern hardware, you have to take a sid chip from a C64 is is very depressing to me.. The same issue if you want to replace a broken sid chip in your working C64. There were some rumours in the past that a big quantity of Sid chip built for the "upcoming" C65 computer where buried somewhere after the Commodore bankruptcy. This would be like finding buried treasure

or gold if it was true! All our problems would be ended..

However, returning to the initial question, I was happy to see that Tulip developed the C64 DTV and that many people like Jeri Ellsworth and others well known in the Commodore community for there enthusiasm were selected for that task. It seems that the right way was chosen as the Monster sid emulation of the Sid chip could open the port for having future Sid style chips to use. But we already know how the story is going up (especially for the people that I have mentioned before that give their energy without recognition) and the new bankruptcy that is coming for the holder of Commodore. This is, unfortunate, even the business politics that released a Commodore Mp3 player with lot of features, but not with the ability to play a PSID file. This is just ignoring the user base :(

With this in mind I don't think that any future Commodore holder will release a new sid chip :(even if they had the original chip layout stored into their archive (maybe some thing to hold it!).

CF. Can you describe the Sids unique characteristics?

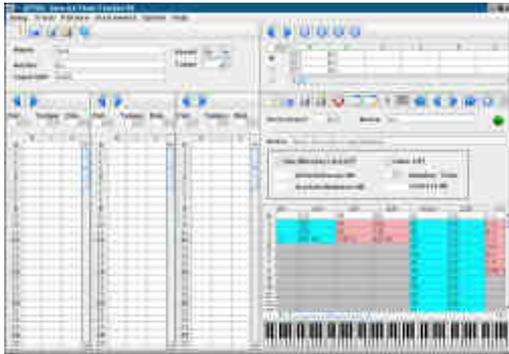
ST. The unique characteristic of the chip, compared to the other chips of the same periods, is that it was developed as a high-quality true synthesizer with lots of unique features. It has tree independent frequency oscillators of up to 4KHz frequency (8 octave) that can use four kind of waveforms (sawtooth, triangle, pulse and noise even combined together).It has three envelope generators that performs ADSR (Attack/Decay/Sustain/Release), tree ring modulators and the ability to synchronize the oscillators. It also has a programmable filter that can be applied to the voices. As it was developed in only 5 months and there were some problems to fit the chip into the silicon surface, many more features were not inserted (like 32 voices) or the use of a better filter. Sid it already so great, can you just imagine what would have been produced if it was created as the creator planned it...

CF. The Sid chip does suffer some design flaws can you describe them briefly to our reader, also some composers have used these flaws to there advantage can you comment on this?

ST. One point was mentioned before: ADSR bug (a flaw affecting the restart of internal chip logic). Many old school composers were able to produce wonderful music simple by setting the appropriate ADSR values to the sid for avoiding this bug and I can assure you that this is not an easy task (I worked on my Driller cover for many weeks trying to avoid the ADSR bug just changing those values: something terrible to achieve especially as you have to make the right timbre to your instruments and you cannot freely change those values...) However, many solutions were taken for bypassing this problem by musicians/editor programmers without having to modify your ADSR settings: the use of hard-restart the note. There are many way for restarting a note (just to mention that one is even sometimes called sexy-hard restart) but I don't want to describe them all here, maybe I could write up about them in a future issue of Sldin magazine.

The main flow of the 6581 sid chip is that there is a (spurious) continuous offset voltage value that can reach the internal 4 bit D/A converter when resetting an oscillator. I think that you know how this flow is manifested: POKE 54272+24,15 and you have the answers... When you change the volume of the sid you get a click from the speaker. Blahh, something that is very annoying. But this is a key for having something very interesting... you just need to change the volume of sid with some "logic" and with some fixed "frequency". In other words you can play sample based music with 4 bit resolution at the frequency (something like from 4KHz to 8KHz for example) that you choose to use. And now the SID can even speak and sing :)

The first applications of this technique for generating sound from the SID was used around 1983-85 for generating speech synthesis. You probably remember the SAM (Software Automatic Mouth) program and maybe those games: Impossible Mission (1983), Slapshot (1984), Beach Head II (1985), Ghostbusters (1984) that have speech inside them (using the speak synthesizer from Berkeley Systems). But it is in 1987 that Martin Galway created Arkanoid music,



adding some real-time generated drum sound inside his music. A little time after that (even if it was ready before Arkanoid) Chris Huelsbeck's Bad Cat music that contains the first ever real sampled based music into it.

Many composers used this new technique, even was music produced using only samples, without the use or the power of sid sound generated sound. For example today there is the Polly Tracker editor that can use more than one sample (with 8 bit quality) and mix them in real-time. Before putting it with 4 bit resolution to the volume of sid. We could speak about samples more and more (for example, the resolution of 4 bit can be increased using other techniques), but the 8580 version of the chip had the flaw (partially) removed and so you can listen to the samples only if you apply a hack to it.

CF, Do you have any favourite sid musicians past and present?

ST. Just four: Matt Gray, Gray Matt, the author of Last Ninja 2 music and the author of Tusker music ... :) I am joking!! Sure Matt Gray is my favourite musician. Even if many people think that his sound is just bass and drums, I find his tunes are true masterpieces: Last Ninja 2, Tusker, Driller, Vendetta ... tunes that are absolutely fantastic. I like the Matt instruments, his style and the "atmosphere" he creates. Martin Galway is another composer I admire He has

squeezed lot the power from SID chip creating some sounds that are memorable like : Parallax, Yie Ar Kung Fu II, Green Beret, Times of Lore, Arkanoid, Rambo II and so on. I think that I could add many more composers ("old" and "new")but I will just nominate some others (in sparse order):Richard Bayliss (has a style that I like: I listen to his tunes for hours), Luca/Fire (his tunes have the best filtered 8580 sound ever for me), Turtle (he hasn't released many tunes from him, but the sound is very attractive for me), Chris Huelsbeck (Giana Sister music is a must),Jonathan Dunn, Dane, Cadaver, Thomas Danko, Soede brothers, Agemixer,Tel Jeroen, Rob Hubbard, ... and I will have to stop here or the list would take up to much space... :)

CF. Do you have any other comments you would like to add

ST. I would like to thank all the musician/composers out there that even today are still producing sid music for our happiness. Thanks to all the people that are working in the sid area (like HVSC crew for example) as this work is always appreciated by me.

For all the readers here: if you have some ideas for realizing something in the C= world, start it. Don't hesitate. All the C= community is behind you...

Finally, I can be reached via the email address ice00[at]libero.it

Best regards to all the people around ... and thanks Nigel for the interview you give me for your great magazine.

visit: <http://digilander.iol.it/ice00>



In the Beginning Part 7} \Lord Ronin from Q-Link\

FOR NEXT LOOPS

FOR NEXT LOOPS are what the Commodore user book has for the next piece of information. By this time I suspect that you have some idea that what the manual presents is not a full programming course, Only a taster of the subject. Yet the manual is an attempt to make the computer a friendly, rather than a scary thing. Each of these parts really deserve full chapters in programming manuals. Getting that information out of the way. Here is what the Commodore user book has for you to type in,

```
new <press return, clears the memory>
```

```
10 forct=1to5
20 ?"commodore 64"
30 nextct
```

Like a nagging yiddish momma, remember to press return after typing in a line. Forgetting to press "Return" is a common fault that we all share, and all forget to do from time to time. Forgetting to press "return" is an annoying user error to find that the line didn't take, when you are debugging 6 pages of a type in programme.

Yeah when you run this programme, it does the same thing as the IF THEN Just that it is smaller. Let's go over it a bit. Line 10, right they are called line numbers. Although I have a programming book collection that isn't American English that calls them line labels. Right back on topic, Line 10 says the same thing that three lines in the IF THEN programme we did recently. Says that "C" is going to be from 1 to 5. Line 20 says that the C= is to print that word line. Line 30 says to do the next ct. Looping from line 10 back through for the 5 prints

A bit of a pause here, I dig that this is all strange and new and to be polite <a rarity for me> odd for you, these commands and typing in the stuff. Alien as you see it for the first time. Personally I was scared to type any of this into the computer, yet this is the stuff from that manual that came in the box with the C=64 PC. Pretty much everyone that uses, used, and programmes; then as well as now, started with to learn how to use the machine. Thankfully this isn't a series on how to programme, I am not that good. Despite my vanity and ego. Here, when you type in and run the programmes, You are doing the same as a guy 25+ years ago did. Feeling the same things as he felt, across the years. Out of the philosophic musings and lets get back to the programming stuff.

What the manual doesn't say is that a FOR NEXT loop could be used for a timer in a programme. 100 for fz = 1 to 250 : next fz. If I wrote that correctly from memory. This will delay the programme for a short amount of time, then the next line of the programme will happen. Mainly I have used this as a delay in my little intro screens. I mean that I have a bit of text on the screen, done with that print statement, this will sit there for a time and then another thing will pop up, Like an entire new screen. That may sound like a big thing, Only because you haven't done it yet, and I write it vague enough to make it sound great, When

in reality it is just a few steps past what is in this manual.

Here is the next thing they have for a FOR NEXT.

```
new <return>

10 forn=1to10 step.5
20 ?nb,
30 nextnb
```

Tearing this apart. Before or after you have rU (run) it. Did the new part. Then declare on line 10 the variable will be "nb". I think that they did that to suggest Number. Saying that it is going to be starting at 1 and ending at 10. That is sort of familiar to you from the previous stuff. But what the smeg is "step .5"

Step is just what it says, a step, sort of like a step in walking, or a step in a set of stairs, One thing at a time. You don't need to tell the C=64 PC to go forward in the steps, If it is just one full step. If you want to run the programme above without the .5 part, Then it will act as if you had said "step 1". Gonna come back to that in a bit.

Next line tells it to print the nb variable, Last line tells it to go to the next nb variable. As you can see from the last few examples of variables, the value of the variable, or less techy said, what is in the variable, is changing each time through the programme. Note in the above one there is that comma. Explaining why there are 4 columns of printout on the screen. All this little thing does is count from 1 to 10, pretty much seen in that first line. Difference is that it is doing it in that .5. OK if you haven't run the programme, it starts off as 1 1.5 2 2.5 and so on until it reaches 10. Step .5 is the way its was told to count, Like I said above. If you remove that "step .5" it will count by ones. In fact if you want it to count by ones in a forward direction, you don't even need to put in the step section. If you are in an experimental mood, Then try changing the "step .5" to "step -1". Try it as a "step 2" or a "step 5". See how it works. Well the step -1 won't work, you should see something like ?syntax error line 10. Why? Kind of hard to count from a low positive number like 1, Up to a high positive number like 10, When you are counting in negative numbers <VBG>. Just change it from 1 to 10, over to 10 to 1 to make it work in line 10.

ADVANCED BASIC

Well we have made it to page 41, titled Advanced Basic. Compared to other books that I have seen. The commands may be advanced, but the lessons are not.

First thing to type in is this little animation programme, right there will be some sort of simple animation on the screen. Because the writing programme that I am using doesn't have the full character set. I am altering this programme a bit. Type it in and give it a run, I'll explain some things about it afterwards.

```
NEW <clears remember?>
```

```
10 rem bouncing ball
20 ?"<clear/home>" <they want you to
type the line number, the print
command, the first quote, then a
shifted clear home key, followed by
ending the quote.>
```

```
25 for x=1to10:"<do one cursor
down>";next
30 forbl=1to40
40 ?" *<do a cursor left>";
50 for tm=1to5
60 next tm
70 next bl
75 rem moves ball right to left
80 for bl = 40to1 step -1
90 ?" <cursor left><cursor
left>*<cursor left>";
100 for tm=1to5
110 nexttm
120 nextbl
130 goto20
```

Scary isn't it? Yet there are some things here that we have already covered in the previous parts. If I typed this correctly, and you entered it correctly, there should be a * symbol that is about 20 lines down on the screen and goes from left to right, then right to left. Actual programme uses the shifted Q to get that filled in circle or ball graphic. We are going to tear this apart, Because there are some points to be made which may have thrown off your typing it in.

Line 10 starts off with the term REM, this is short for REMark. When the 64 PC sees that in a programme, it is lazy and doesn't want to read remarks, so it acts as if it isn't there at all. This is a way for a programmer to insert information in the programme for others to see and work upon. I have seen it in menu programmes, A way to tell the user the how and what. Also seen it in BBS games, breaking the parts of the programme down for editing or fixing, That's the good part. The bad part is that it also eats up memory space.

Line 20 I had to add explanation. You already have the print statement; know about the fact that what is in quotes is what you are going to have printed to the screen. Notice that when you did the quote and then the shifted clear/home keys there was a symbol that appeared. That is the symbol for that shifted clear home, Generally it is called just clear home; When not shifted it is called clear.

Line 25 is wrong. Yuppers it is wrong in the book, in all 27 copies of the book that I have, This is written incorrectly! Shocked that I contradict the writers of the book? Well there is another line coming up that I will say is wrong as well. Look at the line, it says that X=1to10. So you know that there is a 10 something in this line for the programme. Shows you the ":" symbol, in fact the book states, that indicates a new command. Now you make a quote and then do just one down cursor movement ending the quote. That will be another strange symbol to you. A : follows and the next command.

Well if you haven't run it yet. Give it a try, I'll be here to explain.... OK see that it goes not 10 lines down, it goes 20 lines down, What the smeg? Many of the members of my users group have thought that they

did it wrong and couldn't fix or see the problem, well here is a problem. Says to do 1 to 10. Using a for next loop on that line, What you don't see is the fact that : with a next as a command, adds another line as if you had said 1 to 20. There is a typo on this line. In order to fix it you need to type ?"<cursor down>";next. The use of that ; will keep it on the same line of command. Basically it won't go down 20 lines just the 10 it is supposed to do for this programme.

Line 30 is another FOR NEXT loop, but only the FOR is on the line, BL is for the ball character. Says it is to be 1to40. This line is not wrong, run the programme, and take a hard look at the left and right side of the screen. Keep a careful eye out and you will see for a short moment, the graphic you used go up one line on one side and down one line on the other, Just one character. But it is going over the line on the screen, this is because of this line #80. OK scratch your head and wonder why, The programme is written so that you will the use it first. There are 40 columns on the screen, why is it doing 41 in each direction? The truth is that it isn't really it is doing 40 columns. Just that in my opinion, the author didn't want to go into a hard to understand the part about counting, in 1982ce.

I mentioned this a little bit earlier, there are 40 columns; But remember that in computer things we will start the count at 0, not at 1. The 64 is doing what it was told, so no problem there. Well it means that you typed the programme in correctly from the book. Only that it was written to sort of re-enforce the 40 column part of the screen, i take it as a something that was glossed over for the time it was written. OK to correct this, we just need to change that 40 to 39 in the line. Can tell you that this shows two things about programming, one is start on some, but not all things, at 0. The other is well ah, just don't believe everything that is in a book or magazine to always be 100% correct, You may have to tweak with it a bit.

Line 40 may have caused you trouble in the way I wrote it, we have the print command, then the quote. Yes there is really a space before the * symbol, that needs to be in your line, Followed by one cursor left command.

Line 50 sets another FOR loop, here it is TM and it is 1 to 5. Think of this as Time and you will see it is the delay, as done in line 60, where it does the NEXT part, looping between those two lines 5 times before it goes to line 70. Line 70 will print on the screen the next variable called BL, Or the graphic symbol that you used.

Line 75 is just another REMark, Indicating a new part of the programme, The area where the motion of the ball is reversed. Line 80 is the same as line 30 In that it is using the same variable label. Also that it is doing a FOR and it is doing that 40 column movement, change that 40 to 39 to keep it on one line. Note that it starts at 40 and goes to 1, using that step thing at -1 that moves the gfx from the right to the left.

Line 90 is another one that may have given you trouble in the way I wrote it. Got the print command and the quote, there really is a space before the first left cursor movement, followed by the second left cursor then the symbol and then another left cursor. 100 sets the time delay to be the same as line 50. 110 is the same as line 60. Line 120 is the same as 70. Line

130 simple tells the entire programme to start over again. The way the programme is laid out, the variable labels of TM and BL can be reused.

Line 40 is the left to right work horse of the programme, they Prints a space that will erase the previous symbol on the line, then print the symbol, and then does a left cursor over the symbol, getting ready to do the erase and print all over again. Line 90 is the same except it is doing everything in the opposite direction. See how it is moved with the need of the extra erasures of the symbol to move delete and print.

OK when you have this working, feel free to save it. Feeling a bit experimental? Well then change the symbol, do this all in upper case for now. Tired of the speed that it moves, try altering the TM variable. One thing that I have the local group members do is to make a black spade go from left to right. But how about a red diamond going from right to left. No problem in changing the symbol, How to do the colour?

Well that takes retyping out the line, just need one more thing in the line. Need to press the control and 1 key for black. You will see a different graphic symbol show up. If not and you see the cursor turn black, you forgot to open the print statement by retyping the " symbol. Same for the diamond or heart, retype the line and add the control 3 and a symbol will appear, run the programme. Did you put them in the right place? If not play with it a while. Maybe you want to change the screen and border colour, make a line 22 that says "my animation" or something like that. We have covered the basics on how to do that. Save it if you wish, the thing is, that you have made it on your own. There is that thing moving back and forth on the screen, You did it! You have just programmed a visual thing.

Now we come to the INPUT command. A way to have the user interact with what you have written. If you went through my thing on programmes for the C=. You may have not fallen asleep on the Text Adventures part. Simple and easy early to learn Text Adventures, use INPUT as the way to have the player enter information for the game.

Lets new the computer and do the following from the book.

```
10 input a$
20 ?"you typed: ";a$
30 ?
40 goto10
```

That is simple and there are two things here that are new, I'll get to the input in a moment. That print command on line 30, what is it going to print? The answer is Nothing! As you see on the line there is nothing to print, so all it is going to do is drop one line down. Since this programme can be run over and over again, that space just separates what you have done from the next part. No matter what you type a word, a number, a gfx, a letter, as soon as you hit the return key what every you input, is output to the screen. Can you see at this time the use of the IF THEN with INPUT? I mean you write a question and IF the answer is wrong THEN the programme prints a nasty response? <VBESG>

Well the secret isn't a big one, you can see on the first line that there is a variable, this will be for the input. Line 20 prints out both the message and what you printed, just that simple. Putting it together with other things is where the "fun" enters the game. I'll add here that there are some people who dislike the input command. Also there is available some code to remove that question mark at the end of the input line when you run the programme. Seen it in action, never found my copy of that command after a move. OK here is a more understandable use of the INPUT command, the following will convert standard Fahrenheit to Centigrade and the reverse. New the machine and lets start.

```
1 rem temperature conversion programme
5 ?"<clear/home>"
10 ?"convert from farhenheit or celsius (f/c)":input a$
20 ifa$=""then 20
30 ifa$="f"then100
40 ifa$<"c"then10
50 input"enter degrees celsius: ";c
60 f=(c*9)/5+32
70 ?c;" deg. celsius = "; f;" deg. fahrenheit"
80 ?
90 goto10
100 input"enter degrees fahrenheit: ";f
110 c=(f-32)*5/9
120 ? f;" deg. fahrenheit = ";c;" deg. celsius"
130 ?
140 goto10
```

A lot of work to do something an old calculator I had would do with a few pushes of the keys, but there is a lot to see here. I mean we are familiar to some degree with many of the components of presented commands, See variables, math work, print statements. Now lets hope that I wrote it down right from the book, as we are going into this one. REM part we covered, doing that shifted/clear home as well, Line 10 is a little different, We have the familiar print statement The : to separate commands on a line, But now we have an INPUT statement, Creating the variable a\$ which should be either a f or a c.

Lines 20-40 are a new thing, sure they are the IF THEN statements, but on line 20 there is something new those "", and no space between them. The line is saying that if a\$ is nothing, (no key press), then go to line 20. Note here that I wrote it as go to line 20, in programming you do not have to write that go to part, just the line number. OK what is happening is that the programme is going to sit there and wait for you do type in the right thing, before it goes anywhere else.

Line 30 works if a\$ is the f, inputted from the keyboard, ff so then the programme will jump to line 100. Ah here is a new one, Line 40 has the does not equal c part, reading the line says that if the input that the user does from the keyboard doesn't equal c, the programme goes back to line 10 saying printing for the (f/c), and maybe the user will understand what he is to type in for the input. Covers here the input of the variable c as well as rejecting all other non recognised inputs, done in just one line.

Line 50 is an input and takes the centigrade degrees making them into the variable c. Line 60 does the math for us, Line 70 prints out what the centigrade is in standard. Here you may need to alter this for spaces to have things line up correctly. Line 80 is a print statement to move down one line. Line 90 takes

it back to line 10 to start over again. Line 100 is the same with the input for Fahrenheit, giving f as the variable. Line 110 does the math, Line 120 prints out the variable as a number and the text, again here you may need to adjust for spacing. Line 130 does that drop down again and Line 140 sends the programme back to line 10 to start over again.

A few things need to be added at this point, you can change the words in the statements. As I am obviously not a fan of metric, I wrote mine as standard for Fahrenheit, and centigrade for Celsius, as that was the term I was taught for the metric measurement. Doesn't hurt a thing if it makes you more comfortable, if you feel like experimenting, try changing the colours of the text, the border and the screen. Be surprised at how that will change the look and feel of the programme. Now that go to Line 10 on lines 90 and 140. Takes you back to the start of the input for the programme. What it will also do is keep making a space from the print line, and then print the first line over again on your screen. If you don't want a long list of that and the answers on your screen, change that goto 10 to goto 5. That line clears the screen and puts the cursor at row 0 column 0.

Sort of like input, we have GET. This one works with each key press. New the PC and type in.

```
1 ?"<clear/home>"
10 geta$:ifa$=""then10
20 ?a$;
30 goto10
```

Run it and see what happens, ok I'll let you know now, but it is more fun to run it. <G> Each character from the keyboard you type shows up one by one on a single line. WHY? OK lets tear it apart. Got what happens in line 1 as the clear screen and start at the top left, Line 10 has a variable called a\$ and the if then statement. Also in that if then statement the "" placed together, A point to bring up for those that intend to use type in books, Font changes in the different books may confuse you to if there is or isn't a space between the quotes. Most of the time there isn't, but it can look like it on the book page. We have a new command called GET, What does it do? Pretty much as it says, it is going to GET something. In this example, It wants to GET the keyboard input, and won't move along, hmm as stubborn as me, till it GETs what it is looking for. That is why nothing happens till you press a key. Works that way for every time you press a key, patiently waiting for you to enter a something, but takes it only one character at a time. Once you press a key, the programme goes right to line 20, where it prints that character where it is the variable a\$. Next is line 30 that loops it back to the start.

The reason that it prints the characters all in a line is the use of that ; symbol. Change that to , and you will have the characters in neat rows at the 4 tab points on the screen. Drop the , and or the ; and it just goes along the left edge of the screen. A simple programme that with a slight addition is how I have people type in their name or other information on the little intro screens I make from time to time for the club. Also basically the same way that callers do when they log into the BBS and type anything in the message bases. The Commodore book thinks that you saved that previous temperature convention thing. They want to show you how to shorten the pro-

gramme down a bit, so they show in the book that you can alter lines 10-40 from all those IF THEN statements to

```
10 ? "convert from Fahrenheit or Celsius (f/c)"
20 get a$: if a$ = "" then 20
40 if a$ <> "c" then 20
```

I dig RPGames. Have done since 1978ce when I was in my late 20s, no idea how many dice I have rolled or how many times I have rolled them. Nice to know that the C= can do dice rolling, but how to do that? Step by step me laddieo. <G> New the system and lets do the following.

```
10 for x=1to10
20 ?rnd(1),
30 next
```

Well run that and you see a pair of columns of numbers less than 1. OK that RND thing is pretty much a random number generator, but there is more to it than that, if you saw the same information on your screen as mine, then it wouldn't be random would it? But that is rather useless for dice rolling. The book now wants you to change line 20 to read.

```
20 ? 6*rnd(1),
```

Well running that and you still have some weird numbers, but you do have something to the left of the decimal point. OK lets change line 20 again to read...

```
20 ? int(6*rnd(1)),
```

Ah now we have solid numbers, But wait a moment 0-5 is not what we want for a D6. So change the line again to read...

```
20 ? int(6*rnd(1))+1,
```

Yeah here we go, from 1-6 gamers you can change the 6 to be a 4, 8, 10, 12, 20, 30 and 100, Reflecting the different game dice. Really advanced stuff is using the KB GFX to show the numbers. I've seen animated bits of dice rolling, along loads of other pretty pretties on the screen. All comes from this little part.

```
int(lower limit+(upper-lower+1) *rnd(1))
```

Ah that is supposed to be on one line, i know that it doesn't make any sense looking like that and there is a thing that will be in the next section that will require you to use this part. So I'll put it out in that area. You see the next area is where we make a guessing game. Simple but remember you are the one creating it, I'll go over the parts for us, and then we will do that weird line.

Continued next month.

TND Creating music with DMC

http://www.redesign.sk/tnd64/music_scene.html

Created by Richard Bayliss

with Special thanks to Rio / Rattenrudel



Final installment

2.5.1 Options

We can step with CRSR Keys up and down in Sector Editor. The HOME Key jumps to first line of sector. SHIFT + [or SHIFT +] will change the current sector to previous or next sectors.

If you want to compose something, first you will need to set up Commands like Duration and Sound, Then you are able to write down notes in the sectors. Otherwise it will use the last settings of a sector before. (-----) represents an empty step. If you want to change all notes to another pitch value from current position, then you should push SHIFT + < or SHIFT + >. All Instruments below will now transpose up or transpose down.

+ key will switch the VOICE ON or OFF, so that you can listen to your placed notes. - key will set an empty step (-----). DEL key deletes a line, INST will insert an empty line instead. CLR (SHIFT + HOME) will delete the whole sector.

Like in Track Editor - a COPY / PASTE functionality is integrated in sectors too. ? Arrow (Copy - part of the screen flicker in green) and @ (Paste - part of the



screen flicker in red). Sectors will be copied only to an END! Command in Tracks (Everything behind that will be ignored).

In DMC7 you are able to trace a sector by pushing ?. But beware, i noticed some crashes if we do it to often, looks like a bug. SHIFT + RETURN will close the sector editing.

2.5.2 Commands

The sector editor allows us to use different commands:

DUR.xx Sets the duration for passing one step (C= + D)

SND.xx Sets an instrument number (C= + S)

GLD.xy Slides/glides between two notes (C= + G)

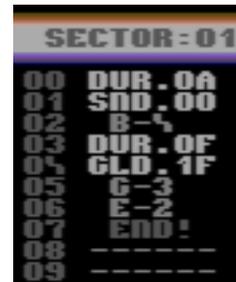
VOL.0x Sets a volume (works if no Attack or Decay is set) (C= + V)

SWITCH Played notes in that section will not be re-set by ADSR (C= + X)

-GATE- Set gate for a played note (£ key)

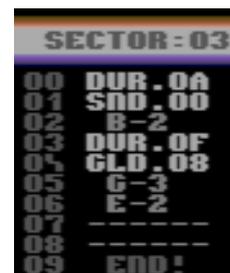
END! End of Sector will switch into the next sector of track (= key)

If you are writing your first tune, you can try to change the duration time (DUR.xx) before the next note is played. DUR is a command like SND etc., which will be noticed by the play routine - but only interpreted as a command



(not as a step!). By setting DUR Command it will jump to next note after waiting of xx Steps. The DUR value represents the duration between 2 note-steps.

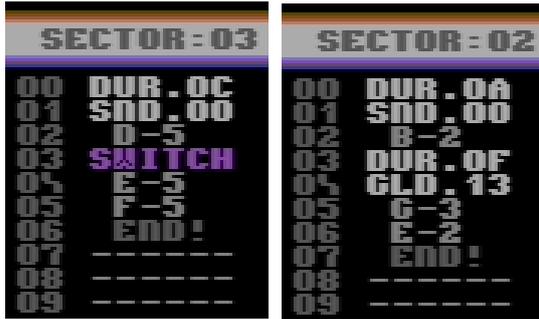
Let's take a look to following example:



```
00 -----
01 DUR.03
02 D-3
03 D-4
04 END!
```

It will do the same like this:

```
00 -----
01 DUR.01
02 D-3
03 -----
04 -----
```



```
05 D-4
06 -----
07 -----
08 END!
```

Ok? But the first method has a better raster timing and is more space efficient. So you can optimized your composed code with using DUR command.



Using GLD.xy command isn't so simple, but here a quick example in two ways:

1. Slide from current note:

We are able to slide from a current playing note to another note. We have to adjust the GLD.xy parameters with x=1 and y=speed (0 - none, 1 - slowest, F - fastest) and we need to set the destination note. You can try to play these sectors (01, 02) in a loop:

You will hear a downslide to G-3 (from B-4) and an up slide to G-3 (from B-2) in the second sector, but both in different amounts of speed, and you also have to watch out how many steps are between the notes. If you slow down

the speed of slide in the first sector (instead F e.g. y = 3), you will notice that the duration of slide will be execute too affecting the next note. Then the next note will slide, which actually shouldn't be slid.

2. Glide between two notes:

The second way is to glide between two fixed notes. We need to set the source and destination note after GLD.xy Command. We have to adjust the GLD.xy parameters with x=0 and y=speed (0 - none, 1 - slowest, F - fastest). We have also pay attention to duration and speed of glide. If another note is to be played and the glide isn't finished, it will influence the following notes too! The both notes, which are followed after that GLD Command will be interpreted together as one step. The following example will demonstrate a glide between G-3 and E-5:

Volume Command VOL.0x can be used to control volume in channel independent from instrument's Sustain level. The maximum level is F. Volume changes can only be used for instruments which don't use Attack and Decay. The last volume adjustment will be stored globally for that channel. If you want to reset volume settings, so that instruments will use their own ADSR, you have to set Volume parameter back to 00.

SWITCH Command won't reset the ADSR for following notes. It can be used in two ways, if we set one SWITCH command, then every following note will use the current ADSR setting until a further SWITCH or END - Command is issued. Then the ADSR will be reset, Take a look at the two examples:

GATE Command will activate the GATE of a played instrument, It should only set after a played note.

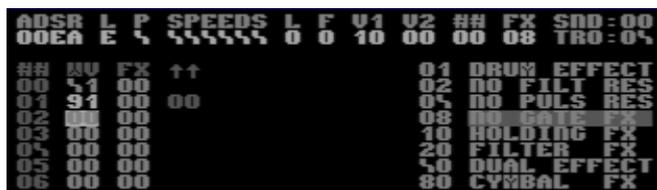
END Command have to be set in every sector and represents the mark of ending a sequence. It will define the total length of your sector too.

2.5.3 Time Control

So the main question is: How can i synchronize all three channels with different played sectors. Therefore the total duration time is necessary, If your "tune speed" is 00 and if you setup your editor as follow, you should find following values on screen: L: 0000 0008

The first 4 numbers represents the current played position. The Second 4 numbers represents the total duration time of a sector. It should be same in all the other sectors (or a multiple or halve of that). In the example above you can calculate the sum for every note: DUR04+DUR02+DUR02 = 8. Notice that this number is always hexadecimal! After Editing the number it'll be refresh if you open again the sector or you start the play routine.

Take care to choose a good divisible value, You'll understand it if you try drums. Notice that at the end of a sequence should stand an -END-, so that the





player knows that you want change to the next sector in track. END, SND, DUR ... are commands, which are only read in from player and so they shouldn't stand in front of an END. These commands are not counted! Notes, GATE and --- are interpreted as full played notes, which are counted in a total duration of a sector.

In menu setup you'll find "tune setup". Here you can adjust the whole song speed. What does it means? A higher value will halve the speed. So that you can calculate the total duration time as follows: L:0000 XXXX = (tune SPD + 1)* all DURs (for Notes/Gate/---)

2.5.4 Global Filter and Pulse
There is an easy way to create cool Filter Effects and Low Frequency Modulation for sequences, which are using filter.

First, you have to create one instrument in Sound Editor. Choose the first instrument number. There you have to choose a filter number (F) and activate the filter (FX: 20) for that instrument. Then go in Filter Editor, choose that number and define these parameters for testing that:

```
F 3 09 00 00
01 00 00 00 00 FF
25 01 01 01 01 25
```

All right, now go back into the Sound Editor again. Copy your instrument-settings on a free place (another instr. number = 2) and paste your buffer. Now set FX: 22 - that means NO FILTER RESET. Go into Sector editor and put following sequence:

```
00 DUR.03
01 SND.01 ;FX: 20 you have to reset first the filter settings ;)
02 C-1
03 SND.02 ;FX: 22 NO FILTER RES to start global filter for next notes
04 C-2
05 C-2
06 C-1
07 C-2
08 C-2
```

```
09 C-1
0A C-2
0B END!
```

Ok, so you can create LFO envelops for all following notes, the filter envelopes are transferable on other instrument, which use these filter settings too. Notice that all these setting are different for both SIDs (6581 /8580). You have to choose higher values on the 6581, otherwise the filtered instruments sound to deep, pay attention to set filtered sounds only in one channel!

Ok, you can try these settings for Pulse instruments too. You only have to choose an instrument with Pulse Wave (41) with defined SPEED values and set FX to NO PULS RES instead NO FILTER RES.



2.6 Quick Recaps and Examples
Familiarize yourself with the sound tables, now it is time to do some serious work on those tables. For a start off re-init DMC and pick a normal instrument, which can be the main instrument for any tune, that you compose in future. Because you're at sound #00, enter the following:

After setup wavetable press SHIFT + RETURN then press

SPACEBAR to test the sound. There, we are, a perfect working sound. The octave of your test sound can be incremented and decrement by pressing SHIFT and '+' or SHIFT and '-' (depending on which way you would like your sounds' octave to go).

Okay, now press the UP ARROW key on your C64 to put the current sound into memory buffer. Then press '@' to paste to the next instrument. Now all you need to do is add a filter. The next diagram shows how a filter can be turned on.

Okay, so not much of a difference to this diagram. All I've done here was added '28' to 'FX'. Basically using '2' or 'A' underneath 'F' turns on the filter and the number under 'X' either emulates as a drum, or just holds your note down, until a GATE is set (GATES will be looked at later on in this feature).

Okay, now press the BACK ARROW (or CTRL in DMC7) yet again to enter the flashing bar menu, and

then highlight the Filter Editor, using cursor keys, followed by RETURN to activate that editor.

The diagram below shows you what the filter editor looks like.

Now try the example above. There, now we have a filter ready for the sounds. Press SPACE to test your sound. There you are, what you think? Quite a nice sound eh? Press BACK ARROW (or CTRL in DMC7) to move back to the main menu.

Anyway, lets' go and create a new instrument, using the sound editor. Enter as follows:

```
ADSR L P SPEEDS L F V1 V2 ## FX
00AE 0 0 000000 0 0 00 00 02 08
```

Move your cursor to ## and then press SHIFT & RETURN. You're able to do this with any sound you decide to create, as well as change the ## values. The ## Values is where your put your sound type. Say for example we use ##66, this indicates the DMC player to jump along to section ##66 in the sound data tables.

Anyway, at ##02 we want to do a C64-style minor chord, so now copy as follows in the data tables:

```
02 21 00
03 21 03
04 21 07
05 21 0C
06 94 00 ;94 loop four sounds.
```

The last line controls the octaves. #03 is octave 3, #07 is octave 07, and #0C is octave 12. #94 repeats the four octaves for a while (depends on the attack/decay used).

Now do the same for the major chords (remember copy & paste to the next sound no).

```
ADSR L P SPEEDS L F V! V2 ## FX
00AE 0 0 000000 0 0 01 02 07 08
07 21 00
08 21 04
09 21 07
0A 21 0C
0B 94 00 - Not hard is it ;o) ?
```

Next we'll introduce some more instruments, for you to try:

Drums and Bass

This chapter takes a look at creating your own C64 drum kit. Let's get back to the sound editor and try as follows.

```
ADSR L P SPEEDS L F V! V2 ## FX
00E8 0 8 000000 0 0 01 02 0C 01
```

Now edit the table below:

```
0C 81 FF ;Drum 1
0D 81 FF
0E 41 0C
0F 41 0A
10 41 02
11 91 00 ;End
```

Now let's make cymbals:

```
ADSR L P SPEEDS L F V! V2 ## FX
00E9 0 0 000000 0 2 00 00 12 A1
```

```
12 81 FF
13 91 00
```

Enter the filter editor and enter in table 2. the following:

```
R T ## RT ST S1 S2 S3
0 4 A0 00 00 FF 00 00
```

Now let's create drum 2:

```
ADSR L P SPEEDS L F V! V2 ## FX
00A9 0 8 000000 0 0 01 02 14 01
14 81 7f
15 41 0E
16 41 0C
17 41 0A
18 41 08
19 81 55
1A 91 00
```

Now test all your drums.

Writing your First Tunes

Let us take a look at what we have done so far. Well, we played around with sounds and filters, now we shall start writing a little ditty. For a start off, hop on to the track editor, by cancelling the main menu (pressing BACK ARROW again). Now move into track editor. Pressing 'C=' and 'E' sets STOP and 'SHIFT + E' puts end and loops to the very first sector.

Now go to track 1, highlight the sector at \$00 and then press SHIFT & RETURN to enter the sector editor. Now do as follows in sector \$00.

```
00 DUR.06
01 SND.XX ;Use the sound number which first drum is
02 A-1
03 SND.YY ;Use the sound no. which are cymbals
04 A-1
05 SND.ZZ ;Second Drum
06 A-1
07 SND.YY ;Cymbals again
08 A-2
09 END!
```

Quick recap: 'C=' and 'S' sets SND.xx, which indicates the sound number which you want to use, while 0-9 and SHIFT A-F sets the sound no. 'C=' and 'D' sets DUR.XX, which is the duration. Yet again, use the same number, and letter keys to set the sound. 'E' sets a GATE (We'll look at this later), 'UP ARROW' copies the sector in to memory buffer, '@' pastes from memory buffer to current sector, 'SHIFT+X' sets SWITCH (Also will be looked at), while 'SHIFT+G' makes GLIDE. To place 'END', just press '='.

Let's edit SECTOR 1 to do a bit of music.

```
00 SND.00
01 DUR.06
02 A-1
03 A-1
04 A-2
```

05 A-1
 06 A-1
 07 A-2
 08 A-1
 09 A-2
 0A END!

Now press F1 to listen to what you have done so far. There we are, your first tune done on DMC V4.0. Easy huh?

Introducing Volume Control

Not much will be happening this time, but we will be taking a look at some really cool things that can be done, when using the DMC V4.0 player. Well, i talked about installing a gate, I also talked about installing, what now? Ah yes, we'll take a look at the volume feature. The volume feature can do amazing effects with your sounds. Especially when you are using the block/sequence editor. For example, you can do a sort of fading effect to your note. Why not try this:

SECTOR 00:
 00 SND.XX
 01 DUR.08
 02 VOL.0E
 03 A-4
 04 VOL.0D
 05 A-4
 06 VOL.0C
 07 A-4
 08 VOL.0B
 09 A-4
 0A VOL.0A
 0B A-4
 0C VOL.09
 0D A-4
 0E VOL.08
 0F A-4
 10 VOL.07
 11 A-4
 12 VOL.06
 13 A-4
 14 VOL.05
 15 A-4
 16 VOL.04
 17 A-4
 18 VOL.03
 19 A-4
 1A VOL.02
 1B A-4
 1C VOL.01
 1D A-4
 1E END!

However the problem is, if you jump from one sector to another. Let us say for example this is your first sector, and you would like to edit your second sector, then you would need to set a volume, else the second sector will use the last volume which was used and we don't really want that do we? Still; not to worry. We can sort this problem out by going to the next sector and set the correct volume or we set the volume back to 00. Okay then, let's try this:

SECTOR 01:
 00 SND.XX
 01 DUR.08
 02 VOL.0E
 03 A-2
 04 -----
 05 -----
 06 -----
 07 -----
 08 -----
 09 -----
 0A -----
 0B -----
 0C -----
 0D -----
 0E -----
 0F -----
 10 -----
 11 -----
 12 END!

Now combine these in track form. like this:

TRACK 1	TRACK 2	TRACK 3
00 - 00	00 STOP	00 STOP
01 - 01	01 - 00	01 - 00
02 STOP	02 - 00	02 - 00

Okay, now press F1 to play the small sample you created. If you find that there are no sounds coming through then the reason behind this is because you have not loaded or created any sounds. Else, you should hear the sample more easily.

Now then, here's a challenge for you. Try and make the volume go UP instead of down. Heh, heh. I don't know if you'll manage this, but why not give it a go.

Introducing Filter Control

Okay, now for something new. We are going to play around with filtered effects coming through with your sounds. When I was writing a piece of music for Commodore Scene's game 'Stealth 4', I originally used the DMC player to do the superb effects, but sadly I could not use the tune in the game, due to part of the player's routines being snipped off. So, I used the Dutch USA Team Music Assembler. But I still love the DMC player. Anyway, let's create a few filters and import them into some of our sounds shall we? Yeah, why not. We'll do at least six or seven filters for this bit. Before we do this initialize everything so that we have no sounds at all. Go to the sound editor and then enter as follows:

SOUND 00:
 ADSR L P SPEEDS L F V1 V2 ##
 00AA 0 0 000000 0 0 00 00 00
 FX
 20

Once you have done the sound data, go to '##' and then press Shift + Return and edit the sound data tables.

00 21 00
 01 91 00

When you have done your sounds, press Shift + Return to go back to the sound editor. Now we are going to the filter editor. Press the 'Back Arrow' key,

and highlight 'Filter Editor' with your cursor keys. Once you have done this, select the first filter and do as follows:

Filter 0:

```
R T ## RT ST S1 S2 S3
F 1 08 00 08 00 00 00
```

Filter 1:

```
R T ## RT ST S1 S2 S3
F 1 18 00 18 00 00 00
```

Filter 2:

```
R T ## RT ST S1 S2 S3
F 1 28 00 28 00 00 00
```

Filter 3:

```
R T ## RT ST S1 S2 S3
F 1 38 00 38 00 00 00
```

Filter 4:

```
R T ## RT ST S1 S2 S3
F 1 48 00 48 00 00 00
```

Filter 5:

```
R T ## RT ST S1 S2 S3
F 1 58 00 58 00 00 00
```

Filter 6:

```
R T ## RT ST S1 S2 S3
F 1 68 00 68 00 00 00
```

Now, go back to the sound editor and copy each sound by pressing 'Up Arrow' and paste to the next sound and change the value of 'F'. This is because 'F' represents the filter number your C64 tries to read.

Now let's go back to the track editor and play around with the filtered sounds. First of all, in your track editor, do as follows:

```
TRACK 1 TRACK 2 TRACK 3
00 - 00 00 STOP 00 STOP
01 STOP 01 - 00 01 - 00
```

Now enter the sector editor and do as follows:

SECTOR 00:

```
00 DUR.03
01 SND.00
02 C-1
03 C-2
04 C-2
05 C-1
06 C-2
07 C-2
08 C-1
09 C-2
0A SND.01
0B C-1
0C C-2
0D C-2
0E C-1
0F C-2
10 C-2
11 C-1
12 C-2
```

And keep repeating until you have done SND.06, then go back down again to SND.01. Well basically that is all I'll be leaving with you with in this chapter. Just play around with DMC.

2.7 Packer DMC 4 and DMC 7 have an integrated packer (inside). If you store your tune, the tune will be relocated to \$1000 and packed. There is no need for any external packer.

2.8 Relocator Often if you do a demo or a game, it's necessary to relocate your tune(s) to another position in memory. DMC 4 and 7 sets the init routine automatically to \$1000. The play routine will start at \$1003 for normal or at \$1006 for multiple player settings. Check out several relocator tools, which are free in internet to relocate DMC 4 tunes. Watch out for the right version too. DMC 7 hasn't any Relocator tool, but Hoeppie (a member of forum64) is still working on it (in 2007 :)). So watch out for next releases.

That's the end of the tutorial.

COMMODORE FREE

Commodore free would like to thank Richard for his permission to reprint this article

http://www.redesign.sk/tnd64/music_scene.html



If you are filling in this form electronically (using a word processor), then please send it to shop@CommodoreComputerClub.co.uk stating that the subject is CCC (UK) membership. Call the file CCC-XXXXXXXXXXXXX when saving - replace the Xs with your name. We will assign you with a membership number.

For those people who have printed out this application form to fill in, please send it with a cheque payable to 'Commodore Computer Club', to Commodore Computer Club Treasurer, 2 Willis Road, Blackburn, Lancashire, BB2 2UA - United Kingdom.

Fees can be arranged electronically (via PayPal), or by personal cheque or postal order. Once you have filled in and submitted this membership form, an invoice will follow with relevant payment details, so please don't forget to fill in your contact details.

Please sign me up for a membership to the CCC (UK) - I wish to be a member for:

6 months* at £3

1 year* at £5

life member at £30

My personal details:

**Name:

**Address:

**Postal code:

Country (if outside of the United Kingdom)**:

**Date of application (Please use dd/mm/yy format):

[FOR INTERNAL USE: Date membership fee received:]

**PayPal account (for those paying by this method):

If you have already registered on the CCC (UK) forums (<http://www.commodorecomputerclub.co.uk/forums>), please tell us your username here:

Email address:

Please tick or cross the box below once you have read and understood the club's rules and regulations:

** I declare that I, the named applicant above, have read and understood the CCC (UK) rules and regulations, and agree to abide by them fully and co-operatively. I understand that I am joining this club on a personal level, and not as a representative of any group, developer, publisher or vendor that I belong to.

Please tick which computers you own:

CBM/PET 40 columns CBM/PET 80 columns VIC/VC 20 C64/64c
 C64GS SX-64 C16/116 Plus/4 C128/128D
 C64DTV Other Commodore 8-bit Other Commodore 16-bit

If you have ticked 'Other Commodore 8-bit' or 'Other Commodore 16-bit', please list these machines below:

Please tick from the list below your interests from the following:

Gaming BASIC programming Machine language coding
 GEOS JOS/WiNGs Tech/scene demos
 Collecting Archiving/preservation Other applications

If you have ticked 'Other applications', please give details below:

Please tick if you use any of the following peripherals below:

Datasette 1541 compatible disk drive 1581 compatible disk drive
 FD2000/4000 CMD HD or RAMLink 1351 mouse or compatible
 Commodore REU Other RAM expansion SuperCPU 64/128
 Other accelerator MMC/Retro Replay SwiftLINK/Turbo232
 RR-Net or FB-Net Other networking device 1541Ultimate/+
 Action Replay Trilogic Expert Cartridge Other cartridge upgrade
 Other speed loader Commodore VDU Other hardware

Please list any items not mentioned above which you have (especially for other Commodore 8-bit machines):

Thank-you for taking the time to fill in this membership form. Your membership will be processed on receipt of the relevant membership fee as stated.

* Membership fees for 6 months or 1 year will be back-dated to the beginning of the month that the membership application has been made and relevant fee has been cleared into our account. This can be paid either by personal cheque, postal order or PayPal. Membership fees are currently £3 for six months, £5 per year or £30 for life membership.

** It is essential that you fill these details, otherwise your membership will not be processed



Commodore Computer Club (UK) (CCC(UK)).

Club rules:

(1) Membership subscriptions, raising funds and re-selling items.

(a) All members will pay a membership fee as follows: £3 for six months, £5 per year and life membership at £30. This fee will entitle the holder to free entry to the meetings, and special limited areas of the website, such as private forums and exclusive downloads, should we get anything exclusive to download. Membership will always be back-dated to the start of the month in which the member took out the subscription, so that everyone joining in the month of April for one year will see their membership expire on the 31st of March the following year.

(b) We should have a 'Commodore Computer Club Shop', which will stock all of the latest hardware mods and sods for Commodore computers. To stop the 'Maurice Randall' effect, in which the club will have to repay people for not receiving their goods because they haven't been delivered but have been paid for, items will only be on sale if they are in stock.

(c) There will be two prices, one for members (cost of item + postage and packing + 10%), and one for none members (cost + postage and packing + 20%).

(d) Any members that do work for the club, organising events, donating items for auction, coding, or are otherwise active, with exception to posting on forums and turning up to meetings, will be considered for free membership and/or lifetime membership on merit based only on work they have done for the club. In certain instances, will include what they have contributed overall to Commodore computing or gaming during their life-time, should any 'Commodore legends' show sufficient interest to join the club.

(2) Events, software and other developments.

(a) Any money that is raised by the club should be used primarily for setting up events, or bolting onto other events as appropriate. This is to go towards, or cover costs with van hire, hotels, food and drink, so the person or people who are willing to travelling to these events, man stalls and generally promote the club and its work are not be out of pocket as far as possible.

(b) Profits made from items sold at events should contribute to cover the costs of attending, or hosting, and/or expenses acrued during the event. This will not include monies raised from membership subscriptions paid for during the event.

(c) The club should seek to raise money for the purpose of developing hardware and/or software that will benefit Commodore users in the UK and world-wide, and that could therefore be sold through the club Shop.

(d) Payments to developers who are commissioned to work on behalf of the club should not be made in advanced or up front unless otherwise agreed by the treasurer and chair-person, and any other two members. This should be openly discussed with all members either in private members areas of the site, or at an organised meeting as appropriate.

(e) Hardware that is commissioned on behalf of the club which reaches production should be sold at a small profit, and monies raised to put back into the clubs funds.

(f) If it is agreed that the club should commission entertainment software, the productions should be available to download for free from the site for members only. Real-media versions should also be sold through the shop with non-members able to buy copies, though at a higher price than members.

(g) Any software commissioned by the club will either be purchased outright, paying the programmer an agreed fee on completion, or paying a lesser fee and splitting the profits at an agreed rate. This should be discussed on a case-by-case basis. The chair-person and treasurer, and two other members, must agree which method should be implemented.

(3) Meetings and monies

(a) The club should hold an annual general meeting in which members have a say in its running, and are able to make suggestions and table official club business for the year ahead. Membership subscriptions should be reviewed at the annual general meeting, and any price increases must be agreed by the chair-person, treasurer and at least two other members.

(b) There should be an annual audit of the clubs finances, with a news-letter at least every three months. The audit should be published before the annual general meeting, and this and the news-letter should be available to current members online in the private member areas. Former members may request this information, which will be granted on a case-by-case basis.

(c) Members will be able to attend any events that organised and run by the club for free when ever possible, whilst non-members will pay a small signing in fee of at least £2.50. With agreement with other event organisers, and at events that the club is attending in an official capacity, we will work towards getting members a discounted entry fee.

(d) All monies raised will go back into club funds.

(4) End of line

(a) If it is apparent that the club is not running within its means to the extent that it is likely to fold, or that legal action against it will lead to the club being dissolved, all club assets should be sold or auctioned off, and members will be refunded their current subscriptions based on the length of time they have been members. The longest-serving paying members will be refunded first as appropriate, either particularly or fully depending on the financial circumstances at the time. The newer members, who will be dealt with last.

(b) Personal donations to the club's funds can never be fully refunded.

Club complaints procedure:

Phase 1: Where a complaint is made against the club, or one of the club members, there should initially be a private apology between the club or individual and the plaintiff. This apology should be for 'any undue harm or upset caused', and will not amount to an admission of guilt or a retraction in any way. The club will not be able to force any of its members to make this initial apology except in the instance that the individual has clearly and admittedly worked on the clubs behalf in the matter specifically relating to the complaint that has been lodged.

Phase 2: The matter should then be investigated to establish the facts. If it is deemed that an individual club member has not been acting on the clubs behalf with regards to the specifics of the complaint, then this becomes a personal matter between the two parties. The club should therefore stop any further investigations or involvement in the matter.

Phase 3: If the complaint lacks any real evidence, or it is felt that the findings are not conclusive, then the matter should be closed. Neither the club, nor any of its members, should therefore discuss the matter publically. All findings should be reported to the plaintiff, and the matter should be considered closed from the club's point of view.

Phase 4: Where a complaint is upheld, a public apology and/or retraction should be published through the official website, and in the newsletter. The club should also give the plaintiff the opportunity to give his or her point of view through the website and/or newsletter as appropriate. In this instance, the case will be considered closed from the club's point of view unless the plaintiff wants to take the matter further through due legal process.

Emergency phase: If at any point during this process the plaintiff feels aggrieved to the extent that he or she instructs a solicitor to take the matter up against the club or club members who have clearly being acting on behalf of the club in this instance, the club should then consider its legal position on the matter, and a meeting should be set up with the principle members of the club within two weeks of receiving legal notice to discuss the matter, and what to do next. Obviously, one would hope that any complaint would ever get to this stage.

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

Commodore Free THE END...

Home Magazine Documentation Software Forums Links

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, Plus4 or even a C64DTV, and would like to know more about the club, please visit [our forums](#).

CCC UK MEMBERSHIP Fri, 07 Aug 2008 11:05:00 GMT

We're in the final stages of setting up the Commodore Computer Club (UK) (CCC (UK)) which also means that we are taking formal membership. Fees start from just £3 for six months membership, up to £30 for life membership. For further information, please [contact me](#), or ask questions on [our club forums](#).

COMMODORE FREE ISSUE 21 RELEASED Wed, 30 Jul 2008 15:00:00 GMT

The 21st issue of Commodore Free has been released, this issue contains articles about Netazar, games with No Ending, Paper Sky 64 model, The Commodore Computer Club UK, in the Beginning Part II, Creating music with DMC and interviews with Vinyr Mairak, Antonino Puzone and Erik Schuback. It can be downloaded from [CommodoreFree.com](#) in PDF, text, HTML, D64 or SEQ files and the HTML version is mirrored [here](#) on the CCC UK website.

FIRST COMMODORE COMPUTER CLUB VIDEO RELEASED Tue, 29 Jul 2008 10:10:10 GMT

The first Commodore Computer Club UK demo, which was released at the first CCC UK meeting on the 29th July 2008 and developed by Conrad of Onslaught and Sennar, has been added to the [software](#) page of the website.

[The most recent 5 articles of 5 news items available | [view all news](#)]

Commodore Apache PHP RSS Subscribe

Home Magazine Documentation Software Forums Links

Content copyright © 2008 Commodore Computer Club. Designed and hosted by Enttec Design.