The Magazine for true Computer Fans

6/2000 Das Magazin für wahre Computerfreaks

DM 9.17 USD 6.17

GO64!

commodore WORLD

KURSE/ TUTORIALS

Assembler
Grafik/Graphics
How To GoDot

SUPERCPU CORNER

ABFRAGE/ DETECTING PAL & NTSC

TIPS & TRICKS

Register 38 of VDC 8568

HARDWARE

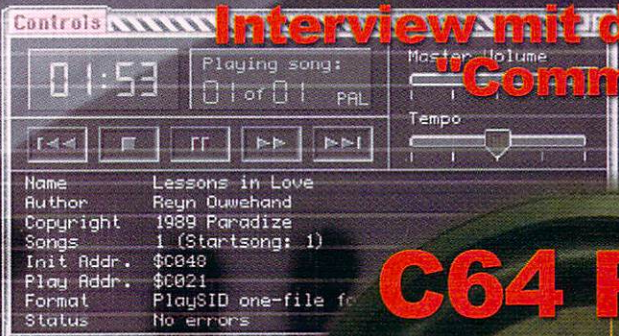
Erfahrungsbericht mit MODPlay

Experiences with MODPlay

COVER

DER C64 MACHT DIE MUSIK!

SID-Sounds im Radio
Der C64 im Musikkanal
Welle:Erdball-CD im Test
Interview mit der Gruppe "Commodore 64"



```
Controls
01:53        Playing song:          Master Volume
             01 of 01         PAL
                                    Tempo
Name         Lessons in Love
Author       Reyn Ouwehand
Copyright    1989 Paradize
Songs        1 (Startsong: 1)
Init Addr.   $C048
Play Addr.   $C021
Format       PlaySID one-file fo
Status       No errors
```

C64 PLAYS THE MUSIC!

SID Sounds On Air
The C64 On a Music Channel
Reviewed: Welle:Erdball CD
Interview With The Group "Commodore 64"

AUDIO CDs AM C64 ABSPIELEN
PLAY AUDIO CDs ON YOUR C64

CMD • HARDWARE • ORDERS: 1-800-638-3263 • SOFTWARE • CMD

FLOPPY DISK DRIVES

CBM 1541C Disk Drive (New w/JD)	$119.00
CBM 1541-II Disk Drive (New w/JD)	$129.00
CBM 1571 Disk Drive (Remanufactured w/JD)	CALL
CMD FD-2000 (800K and 1.6 MB)	$179.00
CMD FD Real-Time-Clock Option	$39.00
Box of 10, High Density Disks (1.6MB)	$14.95
Box of 10, Enhanced Density Disks (3.2 MB)	$29.00

HARD DRIVES

HD-KIT (Controller, Case, PS, Manual, Serial Cbl.)	$249.00
HD-1000, 1 GB (Special Edition)	$449.00
HD-2000, 2 GB (Special Edition)	$549.00
HD-3000, 3 GB (Special Edition)	$669.00
HD-4000, 4 GB (Special Edition)	$749.00

MISCELLANEOUS HARDWARE

Monitors (40 column and 40/80 column)	CALL
Aprotek 3-Way User Port Expander	$30.00
Aprotek User Port Extension Cable	$19.00
C-64 Computer (Remanufactured, w/JD & PS)	$99.00
C-64C Computer (Remanufactured, w/JD & PS)	$109.00
C-64C Computer (New)	$139.00
SX-64 Computer (Remanufactured w/JD)	$299.00
C-128 Computers	CALL
C128-D Computers (Remanufactured w/JD)	CALL
C-64/64C PowerSupply (New, 1.7 Amp.)	$29.00
C-64/64C RepairablePS (New, 1.7 Amp.)	$39.00
C-64/64C Heavy Duty Repairable PS (New, CMD)	$64.95
C-128 RepairablePower Supply (New, CMD)	$64.95
C-128D Internal Power Supply (New)	$39.00
1541-II/1581/RLv1 Power Supply (New, 1 Amp.)	$25.00
CMD EX2+1 3-Port Cartridge Port Expander	$39.00
CMD EX3 3-Port Cartridge Port Expander	$34.00
CMD GeoCable II Print Cable 6ft./15ft.	$34.96/$39.95
CMD Sega-style Gamepad for C-64/128	$24.95
CMD SmartMouse (1351 Compatible Mouse)	$49.95
Flip'n'File 25 Disk Holder (5.25-inch.)	$5.95
Floppy Disks (25 pack of DSDD 5.25-inch.)	$10.00
Inkwell Light Pen Model 170C	$75.00
Monitor Cables	CALL
Cannon BJC-4400 Color Inkjet Printer (Refurb)	$179.00
Panasonic KXP-1150 9-pin Printer	$199.00
Panasonic KXP-2130 24-pin B/W Printer (Refurb)	$199.00
Panasonic KXP-2130 24-pin Color (Refurb)	$299.00
Panasonic KXP-2130 24-pin Printer (B/W)	$329.00
Panasonic KXP-2130 24-pin Color Printer	$429.00
Smartline C-600 Printer Interface	$10.00

UTILITIES

Big Blue Reader V4.10 (SOGWAP)	$39.00
CMD Utilities	$24.95
JiffyMON-64 (ML Monitor)	$19.95

LANGUAGES & COMPILERS

BASIC 64 Compiler (Abacus)	$17.00
BASIC 128 Compiler (Abacus)	$25.00
Blitz! 64 Compiler (Skyles)	$30.00
Buddy 64/128 Assembler	$39.00
Cobol 64 (Abacus)	$17.00
Pascal 64 (Abacus)	$17.00
Power C (Spinnaker)	$19.00

GEOS

Collette Utilities (Handy Geos Utilities)	$19.95
Desk Pack Plus	$29.00
Dweezils Greatest Hits (NewTools2,Stamp,Label128)	$30.00
FONTPACK Plus	$25.00
gateWay 64 or 128 (Specify Version)	$29.95
geoBASIC	$20.00
geoCalc 64/128	$40.00/$45.00
geoChart	$29.00
geoFAX	$39.95
geoFile 64/128	$40.00/$45.00
geoMakeBoot (Makes Bootable copies)	$12.95
geoProgrammer	$45.00
geoPublish	$40.00
GEOS 64 v2.0	$44.00
GEOS 128 v2.0	$49.00
geoSHELL V2.2 (CLI for GEOS)	$24.95
International FONTPACK	$25.00
Perfect Print LQ for GEOS (Laser-like output)	$49.95
RUN GEOS Companion	$20.00
RUN GEOS Power Pak I or II (Specify)	$20.00
Wheels 64 (GEOS Upgrade, req. GEOS 64 v2.0)	$36.00

MIDI & SOUND

Digimaster	$34.95
SID Symphony Stereo Cartridge	$44.95
Dr.T's Caged Artist or AlgorythmicComposer	$15.00

EDUCATIONAL

Adventures w/ Fractions (ages 10-14)	$15.00
Expeditions (ages 8-15)	$15.00
Pre-Reading (ages 3-7)	$15.00
Spelling Bee (ages 8-13)	$15.00
The Glass Computer (ages 12+)	$15.00
The Market Place (ages 8-14)	$15.00
Where in Europe is Carmen SanDiego?	$29.00
Where in the World is Carmen SanDiego?	$29.00

ACCELERATORS

SuperCPU 64	$179.00
SuperCPU 64 w/SuperRAM (no RAM installed)	$219.00
SuperCPU 64 w/SuperRAM (4 MB)	$254.00
SuperCPU 64 w/SuperRAM (8 MB)	$278.00
SuperCPU 64 w/SuperRAM (16 MB)	$318.00
SuperCPU 128	$239.00
SuperCPU 128 w/SuperRAM (no RAM installed)	$279.00
SuperCPU 128 w/SuperRAM (4 MB)	$314.00
SuperCPU 128 w/SuperRAM (8 MB)	$338.00
SuperCPU 128 w/SuperRAM (16 MB)	$378.00
SuperRAM Card (no RAM installed)	$79.00
SuperRAM Card (4 MB)	$114.00
SuperRAM Card (8 MB)	$139.00
SuperRAM Card (16 MB)	$179.00

RAM DEVICES

RAMLink Base Model (ØMB, No RAMCard)	$169.00
RAMLink w/1MB RAMCard (Special!)	$209.00
RAMLink w/4MB RAMCard (Special!)	$249.00
RAMLink w/16MB RAMCard (Special!)	$369.00
RAMCard RTC Option/After Market Kit	$10.00/$19.00
RAMLink Battery Back-up (Optional)	$24.95
Parallel Cable (RAMLink to HD)	$19.95

PRODUCTIVITY

Bank Street Writer	$12.00
Cadpak 64 (Abacus)	$22.00
Cadpak 128 (Abacus)	$25.00
Chartpak 64 (Abacus)	$17.00
Chartpak 128 (Abacus)	$25.00
Data Manager 64 (Timeworks)	$16.00
GoDot	$34.95
I Paint v1.5 (128, 80-col., 64K VDC)	$39.00
I Port v1.54 (128, 80-col., 64K VDC))	$29.00
Personal Portfolio Manager (Abacus)	$16.00
RUN Productivity Pak I, II, or III (Specify)	$15.00
RUN Super Starter Pak 1541 or 1581	$20.00
RUN Works	$20.00
SEC Check Register 128	$50.00
SuperScript 128 (Precision)	$20.00
Suberbase 64 Version 3.01 (Precision)	$35.00
Superbase 128 Version 3.01 (Precision)	$35.00
SwiftCalc 64 (Timeworks)	$16.00
TWS 64 w/Speller (Busy Bee)	$29.00
TWS 128 w/Speller (Busy Bee)	$39.00
TWS Modules (HD/RL/Illustrator)	each $5.00

TELECOMMUNICATIONS

Aprotek Com-Modem Adapter (for ext. modem)	$19.00
Aprotek Com-Modem Adapter Cable	$9.95
Commodore 1670 Modem (1200 baud)	$19.95
Modem (9-25) or Null-Modem Cable (9-9 or 9-25)	$9.95
Novaterm 9.6 (specify 3.5" or 5.25")	$29.95
Turbo232 Cartridge (Up to 115Kbps)	$39.95
USR Sportster 33.6Kbps FaxModem	$169.00
USR Sportster Faxmodem w/Turbo232 & Cable	$199.00

GAMES

Atomino	$17.00
Ballistix	$13.00
Chomp!	$10.00
Cut Throats!	$10.00
Day in the Life of Prehistoric Man	$19.00
Escape Route	$19.00
Felony	$10.00
Frogger	$10.00
Grand Prix Circuit	$10.00
Guerilla	$10.00
Hardball	$10.00
Heavenbound	$19.95
Island of the Dragon	$19.00
Jordan vs. Bird: One on One	$10.00
Kings of the Beach	$10.00
Lions of the Universe	$19.95
Mainframe	$13.00
Mean Streets	$10.00
Menace	$15.00
RUN C64 Gamepak or C128 Funpak (Specify)	$10.00
Skate or Die	$13.00
The President Is Missing!	$10.00
The Three Stooges	$10.00
Tie Break Tennis	$16.00
Total Eclipse	$10.00
Walkerz	$19.00
War in Middle Earth	$10.00
Wings of Circe	$19.00

BOOKS

Anatomy of the 1541	$8.00
C-64 Science & Engineering	$8.00
C128 Computer Aided Design	$8.00
C128 BASIC Training Guide	$8.00
Commodore 64 Tricks and Tips	$8.00
Compiler Design & Implementation for the C64	$8.00
GEOS Programmers Reference Guide	$28.00
Graphics Book for the C-64	$8.00
Hitchhikers Guide to GEOS	$28.00
Ideas for Use on Your C-64	$8.00
Printer Book for the C64	$8.00
Mapping the C64	$8.00

••• All Major Credit Cards Accepted •••

Mastercard • Visa • Discover • American Express

CMD

Creative Micro Designs, Inc.

P.O. Box 646
East Longmeadow, MA 01028
Info: (413) 525-0023
Fax: (413) 525-0147
Visit our Web Site at http://www.cmdweb.com/

Wheels

THE HOTTEST UPGRADE EVER CREATED FOR GEOS

If you've been waiting for GEOS to get better, faster and more efficient, your wait is over! Click Here Software's *Wheels* takes GEOS to the next level, with full integrated support for all the latest hardware. *Wheels* also adds many features not previously available in GEOS, as well as some that could only be had by buying or adding other utilities and patches. GEOS is really going someplace, and it has the *Wheels* to get there!

Wheels64 **$36.00** *Wheels128* **$40.00**

Minimum requirements for Wheels: GEOS 64 v2.0 or GEOS 128 v2.0, 1541 or 1571 disk drive, mouse or joystick, and 128K or larger RAM expander.

Novaterm

version 9.6

Join the Telecommunications Revolution... get on the Information Super Highway TODAY!

- **Easy-to-use – novice user configuration**
- Now supports **UUencode and UUdecode** and converts files on disk or in a buffer. Also ASCII <> PETSCII and PETSCII to Unix.
- Protocal support: **Zmodem up/download, resume (crash recovery), streaming to buffer**; Ymodem batch; **Ymodem-g**; Xmodem-1k; **Xmodem-1k-g**(to buffer); Xmodem-CRC; Punter; Kermit; WXmodem.
- Supports High Speed Interfaces like Turbo232 for speeds up to 230Kbps.
- Supports ANSI color/graphics, VT102, and VT52 in 80 col. Plus Commodore color graphics and ANSI color. **All emulation modules now load from one window.**

Just $29.95

(Specify 5.25" or 3.5" disk)

- New faster 80 column modes: **Soft-80 on C-64 (enhanced scrolling w/ REU); and 25- or 28-line C-128 VDC 80-col. mode.**
- **Use any memory device as a buffer** including: 17xx REU, RAMLink partition, GeoRAM, C128 VDC, BBGRAM, and internal C64 memory.
- New text capture features:Capture text on-line and store in any device; High-speed transfers directly into memory and now **retain buffer contents with battery/power backed RAM devices such as RAMLink**.
- Enhanced full featured **Text editor can now load/save files from the buffer** and has an integrated script compiler.
- Script language for automatic operations
- Simple BBS mode - dial-in downloading
- Supports **real-time clocks in CMD devices**
- **All new user's manual**

JiffyDOS

Increase Disk Access Speed by up to 1500% While Retaining 100% Compatibility!

JiffyDOS 64 System **$39.00**
Includes JiffyDOS C64 Computer ROM and one JiffyDOS drive ROM.

JiffyDOS 64c System **$39.00**
Includes JiffyDOS C64c Computer ROM and one JiffyDOS drive ROM.

JiffyDOS SX64 System **$49.00**
Includes JiffyDOS SX64 Computer ROM and SX1541 drive ROM.

JiffyDOS 128 System **$49.00**
Includes JiffyDOS C128 Computer ROMs and one JiffyDOS drive ROM.

JiffyDOS 128D System **$49.00**
Includes JiffyDOS C128D Computer ROM and 1571D drive ROM.

JiffyDOS Drive ROMs **$19.00**
Available for Commodore 1541, 1571, 1581 and most compatibles.

Note: Computer Serial Number Required for C-64/64C Orders

Shipping and Handling Charges

	Continental US		AK,HI,PR	Canada	Others
Order Subtotal	UPS Ground	2nd Day	UPS 2nd Day Air	Air Parcel Post	AirMail or UPS
$0.01 to $19.99	$4.75	CALL	$16.00*	$5.00	CALL
$20.00 to $29.99	$5.75	CALL	$17.50*	$7.00	CALL
$30.00 to $59.99	$6.75	CALL	$21.00*	$9.00	CALL
$60.00 to $149.99	$7.75	CALL	$29.00*	$12.00	CALL
$150.00 to $299.99	$11.00	CALL	$38.00*	$20.00	CALL
$300.00 to $799.99	$16.00	CALL	$42.00*	$25.00	CALL
$800.00 and up	$21.00	CALL	$55.00*	$38.50	CALL

UPS C.O.D. add $5.00 (US only)

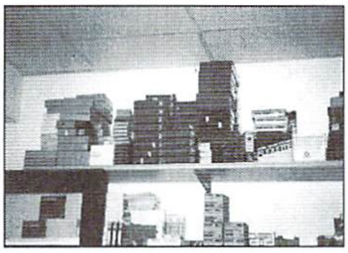
* Other shipping methods available. Call for details.

POLICIES: CMD accepts: Major Credit Cards, Money Orders, COD and Personal Checks. Personal Checks are held 3 weeks. Customer is responsible for shipping charges on refused packages. Most items are stock, contact CMD for firm delivery. Returns for merchandise credit only within 30 days with prior authorization. Credits are less shipping, handling, taxes, customs, duties, and a 15% restocking charge. No refunds or credits on opened software. All prices and specifications are subject to change without notice. CMD Business Hours: (Orders) M-F, 9AM-5:30PM EST (Tech Support) M-F, 1PM-5:30 PM EST.

CONTENTS

- June 2000 -

5
Commodore Oasis



They do exist, small oases for our Commodores - and we will uncover them. We start with Landolt Computer in Doerningheim, where Commodoreenthusiasts may discover a lot of goodies for their beloved machines.

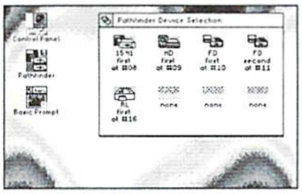
6
There's music in the air

The C64 makes the music - and its not that unsuccessful after all. Read more about this topic in this issue's cover story about radio DJ Hard HatMack, the Hip Hop group Commodore 64, the C64 on the music channel VIVA and about the new CD from Welle:Erdball, all beginning on page 10.

18
Graphics tutorial

It's none other than Deekay of the demo group Crest who is reveiling the secrets of those mysterious formats like SHIFLI and UIFLI. Special imageeditors included. More on page 18.

24
CLiPS - Corner: Drives



To work with more than just 4 drives is a dream for many GEOS users. Well, its not really a problem for CLiPS. Read more about drive management in this brand new operating system on page 24.

[news]

[tutorials]

[hardware]

[tips & tricks]

[cover disk]

[departments]

Vorschau auf GO64! 7/2000

Totally SuperCPU!

The C64 in turbo speed - real power users know how to astonish even hardcore PC freaks. The SuperCPU is able to accelerate software to unbelieveable speeds. This is why the next GO64!/Commodore World will deal with this topic. As a special surprise the newest version of the SuperCPU monitor "Dreamon" and an exclusive preview of the brand new operating system CLiPS is on our cover disk. Better don't miss it!

Dear reader,

Music has always been one of the strengths of our Commodores. Back in 1982 when the C64 was first presented to the public, the media was deeply impressed about the sound capabilities of this small computer. High-end PCs from that time couldn't do more than boring one voiced sinus beeps. But the cheap and revolutionary home computer from Commodore could do much more than that. The typical SID sound was part of the reason for the cult machine the C64 became a decade later on.

However, SID music is not only well-known from the countless games and demos from the eighties and nineties, but is now beginning to conquer the desktops of more 'mainstream' consumers in new ways. SIDStation and HardSID, as reported in earlier issues of GO64! magazine, are examples of this.

In this issue we will cover the SID in all it's aspects. Just imagine, a radio broadcast entirely reserved for SID tunes, or the newest CD from the group Welle:Erdball, are only two of the many stories we are covering in this issue. I wish you much fun in reading them.

Guenther Bauer

News Flash

BTX has to go eventually

Although earlier in the year 2000 the German company Telekom claimed otherwise (cf. the article on page 7, GO64! issue 12/99) the BTX network along with the special call number 01910 will be shut down by the end of 2001. We are truly disappointed that one of the oldest online services available for C64 users in Germany has to see this end. If you can access the Internet, this is the place to voice you complaint:

http://www.dtag.de/tool/mail/index.htm

(wg)

Uncle Wanja's report on "The Wave" causes stir in the press

At first, it was only one of the many reports on the c't news ticker: Maurice Randall's new C64 web browser "The Wave", information on which was forwarded by our reporter Wanja Gayk to a member of staff of the renowned computer magazine "c't". Even though our breadbox saw its transformation into a bread can there, the report quickly made its way e.g. into the video text of the German TV cable network "Kabel 1" and in the radio program of Delta Radio.

As a result, the number of visitors of the Internet home site of "The Wave" rose to new heights. The future holds interesting prospects, it seems to us...

http://videocam.net.au/colinjt/wave.html

(vr)

Internet text browser for every C64!

Cameron Kaiser is currently working on his "Hyperlink 2.5" text browser which is designed to make Internet access available to every ordinary C64. The only requirements are for you to have a shell account and a Hayes-compatible modem with a bit rate of at least 1200 bps. Any additional hardware such as a SCPU or a RAM extension are not necessary but will be supported. That is the reason why "Hyperlink 2.5" will run faster on a C128D, because it uses the VDC as a cache.

http://www.armory.com/~spectre/cwi/hl/web.html

(gb/vr)

Free Internet access

There's a free treat awaiting our American readers: Under the address www.teledyn.com/products/FreeWWW/ , they will find an offer of free Internet access. The page promises 80 hours per month with a maximum of 5 hours per single session. A PPP connection is required. So, this makes it possible to take advantage of the offer with "The Wave", the C64 web browser running under Wheels.

Under unix-shells.com you can set up a free telnet account. You will need a connection to the Internet, then you can use a terminal program to connect to unix-shells.com. An easy start menu will take you through the registration and you will be given your own account immediately afterwards. All main Internet applications can be used without graphics under telnet. Amongst these applications are a news reader (for news groups), an IRC client (for chat groups), various e-mail programs and, last but not least, Lynx as a text-based web browser.

(gb)

Last Ninja Music Compo

There is a new competition for all you SID artists out there. This time, the judges are looking for new interpretations of the brilliant tunes from the "Last Ninja" trilogy. A hand picked jury, featuring amongst other also Reyn Ouwehand who composed the music for "Last Ninja 3" will decide on a winner. The first three winners will receive money prizes. You can find further information on the Internet under:
http://www.lastninja.c64.org
or http://www.the-dreams.de

Email: ninja@the-dreams.de

(ws)

Construct your own four-player adapter

You can now find the construction instructions for the famous four-player adapter for 'Bomb Mania' by Classical Games on the home site of the games company Protovision. They also offer the game without the adapter at a much lower price.

More things to be got from Protovision's home site: the snowball fight game 'Ice Guys', based on the good old 'Bubble Bobble'.

http://home.t-online.de/home/bundyman

(vr)

New open mailing list for all issues concerning the Commodore

Gaelyne Gasson, member of the GO64! staff, has set up a new mailing list under the name of 'Homestead Commodore'. This open list in English language leaves room for an unlimited range of topics and wants to invite everybody to join in with an interesting few hours of discussion.

More information under this address:
http://cbm.videocam.net.au/homestead/

Or simply send an e-mail with the subject and content (!) "subscribe" (without the inverted commas) to the address homestead@videocam.net.au to subscribe.

(vr/gg)

'DreamDreamBox' is no longer on-line

Because of a lack of time and users the 'DreamDreamBox' mailbox had to go offline. Active users are asked to switch to other mailboxes and offer their support there.

(ws)

Slide show in the JPEG graphics format

As a follow-up to his brilliant JPEG viewer, Steve Judd has designed a slide show program for JPEG formats. You can download it for free on the Internet under:

http://www.ffd2.com/fridge/jpeg/ssw/

(ad/vr)

Lots of remainders!

Landolt Computers in Doernigheim

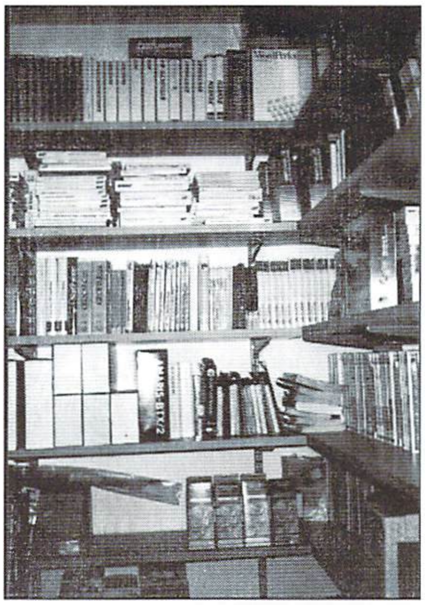
They are still out there - oasis for our beloved Commodore's. We will try to introduce shops that still carry software or hardware for the C64 and C128. If you know such an oasis, whereever in the world it may be located (GO64!/Commodore World is international!), please don't hesitate and drop us a line! This time we will feature Landolt-Computer in Dörnigheim.

by Hartmut "BUGJAM" Pachl

This story begins with a SuperCPU. When I bought this unbelievable machine complete with a SuperRAMCard I had to get the appropriate PS/2-SIMM for it - and I encountered more difficulties than I had expected. First, I searched in various smaller computer or electronics shops in Hanau (that's where I work) in Germany until someone told me about Landolt Computers in the nearby town of Maintal-Doernigheim: "They've got a lot of the old stuff, maybe they also have things for C64s. Why don't you give it a try there?"

Some time later I was on my way to the industrial area of Doernigheim. At the shop - a rather plain-looking, one-storey building with a large Commodore sticker on the front door (first flash of happy anticipation) - I was in for a real surprise. I didn't find the RAM I was looking for but instead I discovered the most astonishing treasures while I was rummaging through the shop. Tons of books, user software, GEOS stuff, a few games, diskettes (even some remaining eight inch ones!), datasette drives and lots more ... There were even some rarities for the VC-20 and the other Commodore 8 bit machines. Moreover, fans of the Amiga could also find things here. The owner of the store was not in that day, so I decided to come back some weeks later in order to get some more information (and, of course, to take away maybe one or two bargains).

Herr Landolt, an elderly gentleman with glasses and graying hair, was a little surprised by my request as well as by the fact that there are still people who are interested in this 'antiquated' system; but soon he agreed to answer my questions. He told me that some years ago he was the biggest Commodore dealer in the whole Rhine-Main area; even today he has some customers who bring their old 8 bit computers and peripherals from far away to have them repaired at his shop. And in fact the company's logo - still the same today - can be found in the advertisement sections of many back issues of the 64'er or Happy Computer magazines.



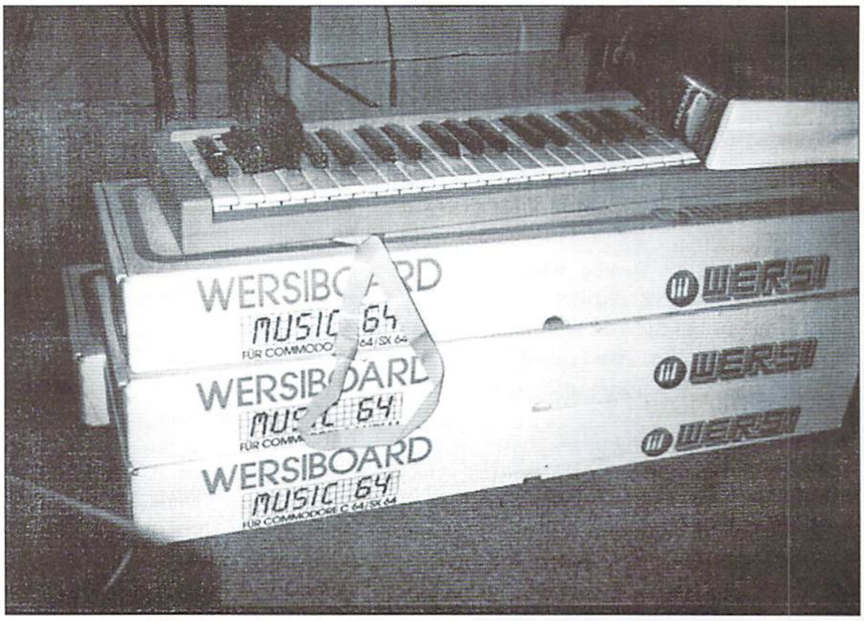
He admitted to have toyed with the thought of throwing the whole 'rubbish' out - stroke of luck that he didn't! But he would not be disinclined to sell the whole range for a fair price. He also keeps numerous replacement parts in store; he told me that people who are searching for anything in particular should just contact him. However, he asked that you please not block the telephone with the time-consuming requests but that you should rather contact him by e-mail, fax or snail mail; then he won't mind looking through his stock for the desired things. So here's the address:

Landolt-Computer
Robert-Bosch-Str. 14
D-63477 Maintal-Doernigheim
Germany

phone: +49 (0) 6181 45293
FAX: +49 (0) 6181 431043
e-mail: vertrieb@Landolt.de
Web: www.Landolt.de

As I live nearby, it would be no problem if you contacted me (+49 (0) 6188 6481), so I could drive to Landolt's for you. But if you are presented with the opportunity of going there in person: don't miss the fun of rummaging through this treasury!

(ad)



Picture: Sebastian Bach

Ich bin eine Commodore 64!

Commodore 64 - that's not only the name of the most successful home computer ever but also the name of a New York music group, calling themselves the "silliest hip-hop band ever". We get more proof in this interview.

by Volker Rust

GO64!: Hi! Maybe you should introduce yourself first...

H.M.O.: Hi, I'm H.M.O., also known as Hauser or Haus.

Teddy Ruxpin: I'm "Smart Money" MC Teddy Ruxpin, and I'm an Aries.

Professa: They call me "The Professa". I teach math at New York University.

GO64!: Let's start with your beginning. You formed in 1982. How old were you then, and what were your personal histories?

H.M.O.: Well, I'm the youngest. So when I was 7 years old, hip-hop was just filtering out to the rough streets of upstate New York. And around the same time my mother got her first IBM PC and my friends were getting Apple IIEs, I was bumping the Fat Boys. I told Teddy that we should be like the Fat Boys. Teddy became the human beat box.

Teddy Ruxpin: I was eight and destined for greatness. [Makes beat box sounds]

Professa: I met Haus and Teddy in the 4th grade in math club.

Teddy Ruxpin: As anyone who's heard the album will tell you, Prof's really the key member of the group. But we didn't call him Professa back then. We called him "MC Squared".

H.M.O.: We still do sometimes.

GO64!: In 1999 came your first album, "K-Minus Initiative". What did you do in these 17 years? Play "Summer Games"? :)

H.M.O.: More like California Games. We've been chillin'.

Teddy Ruxpin: We started performing at school talent shows in the 4th grade and stuff like that. We'd rap about like Popeye's flea circus and stuff like that. The rest is history, as we Americans like to say.

GO64!: Why did you choose "Commodore 64" as your name?

Teddy Ruxpin: We were thinking about important discoveries and inventions of the 20th Century. Commodore 64 just tripped off the tongue a lot easier than "Polio Vaccine".

Professa: (laughing)

H.M.O.: I'm actually a proud Commodore 64 user, or was. It's the funkiest computer. With a name like Commodore 64, the computer will live forever!

Professa: Same for us.

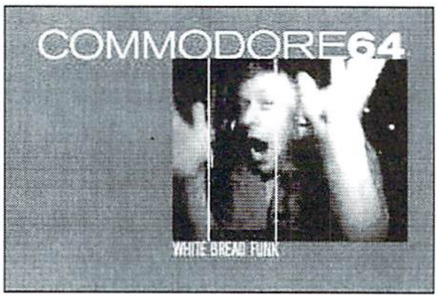
GO64!: Do you still have a C64 in use?

H.M.O.: Alas, I sold my souped-up Commodore 64 (running GEOS 1.2), floppy drive, and Okidata printer for $150 back in 1990. But now I'm running Power64 emulator on my G4.

Teddy Ruxpin: Emulators are great. You know you can download Karateka off the Internet? That game is the joint.

GO64!: What are your favourite games on the C64?

H.M.O.: All my friends had Apple IIs, so I was jealous. I loved Castle Wolfenstein. Or maybe it was Beyond Castle Wolfenstein. I still have nightmares about that f****r. Isn't that illegal in Germany? California Games was fun... I remember being good at surfing. Yie-Ar Kung-Fu was dope. I also had some "educational" time-travel cartridge.



Professa: Rush 'N' Attack, Pitfall, Ghosts & Goblins, Test Drive, Choplifter, Knight Games, Burger Time.

Teddy Ruxpin: My girlfriend Vandana used to play Seahorse Hide and Seek.

H.M.O.: In my final years as a C64 user, I came down with a severe case of Macintosh envy. Hence GEOS, which was great. But I tried to play "Deja Vu" and "Dark Castle" on the Commodore, and it just wasn't the same.

GO64!: C64 bands, like "Welle:Erdball" or "Mikron64", use SID samples in their songs and C64 video projections for their live performances. Did you do something similar? I remember "SAM" speaking the intro of your album...

H.M.O.: We had no idea about the persistence of the Commodore as a synthesizer. We're blown away. But we do use Commodore 64 software in making some of our beats. Our first single, Foam, has sound samples from California Games, or maybe it's Karateka, as a major element. We use Power64 software to export the 4-bit soundtrack to an AIFF file, which we can import into ProTools software on our G4.

Professa: I'm rocking Power64 and ProTools on my new PowerBook.

Teddy Ruxpin: A lot of people in England thought that that was "SAM" speaking on the album. Close. It's our DJ. He's an iMac, however. Tangerine. His name is DJ Goodbeats.

Professa: He hates us.

Teddy Ruxpin: He wants to go solo. He hates humans.

H.M.O.: Maybe he's related to SAM.

Professa: Also, for one track we've incorporated an old TV ad for the C64. It's really funny.

GO64!: What are your own musical preferences? Are there any bands and why?

H.M.O.: We all have our own idiosyncratic tastes, but we all love hip-hop.

Teddy Ruxpin: Prof's got a double-album of British Parliamentary speeches that he won't stop listening to. I've also been studying commercial jingles lately. They have that really annoying yet inescapable quality that we're trying to capture ourselves.

GO64!: Is your first album a commercial success? How is it with radio airplay in the States?

H.M.O.: We've sold more albums than we've ever dreamed. When we were featured on MemePool, a popular website, we jumped to the very top of Amazon.com's hip-hop charts. We've also been in Entertainment Weekly and Time magazines.

Teddy Ruxpin: We've sold several thousand CDs in independent record stores in New York City -- Other Music and Etherea, for example. We also have a lot of fans in the U.K., the Netherlands.

Professa: It actually seems most of our fans are from economically troubled countries in Eastern Europe.

H.M.O.: We've been on college radio all over the country, notably "The Breaks" on WREK in Atlanta.

GO64!: When will you quit your jobs for the music?

Teddy: I quit my job three weeks ago.

H.M.O.: He really did. I'm quitting in September.

Professa: I'm working in Los Alamos, New Mexico this summer.

GO64!: Do you have any future plans for this year?

Teddy: Oh, yeah. Big plans. We're making some more videos and talking about a full-length film. Plus we're hard at work on the follow-up album. And we're also writing a rap opera about Neptune, King of the Sea, but it's been sort of slow going. All we've got so far is "Neptune, King of the Sea / Got two fins where his feet should be."

Professa: I'm going to spend the next few months cooped away in Los Alamos working on beats.

H.M.O.: We've got some great material for the next album already, including the videogame-inspired, "Night on Mt. Mountain Smash", which is comprised entirely of animal noises. We're also going to sample Dr. Who.

Professa: Wasn't that a Commodore game, too?

GO64!: Many thanks for the interview!

Teddy: Thank you! Ich bin eine Commodore 64!

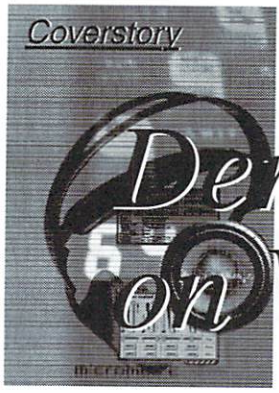
What Uncle Wanja thinks:

The introduction to the CD makes the promise of containing "sick music" by "sick people", one of them known by the name of Bass-IQ (pronounced as in 'BASIC'). And this is what we get: confusing lyrics, weird, monotonous beats overlaid with classic old-school rap and samples on strange instruments that make it hard to distinguish between the lyrics and the samples. The music often sounds as if a group of very stoned guys were jamming away in a garage. If you are expecting anything similar to old albums by Ice-T or A Tribe called Quest, think again. Beastie Boys with a huge bong could maybe convey the right idea - without my wanting to accuse anybody of illegal drug abuse. The tunes are not really good for dancing to but to me it feels like a perfect example of elevator music, which is sometimes exactly what you're looking for. Number 9 on the CD reminds one of a video game with a bit of hip hop jam added. All voices sound as if spoken (or sung, rather) through a megaphone and somehow are a bit lost in the music. Personally, I wouldn't listen to it at home because the CD has the effect of sending me to sleep from boredom, but we all know that tastes are different. Maybe the music would find its rightful environment in a late-night bar, as an accompaniment to that slow last quarter of an hour in front of your umpteenth glass of Tequila, before your heavy head finally hits the bar counter and the lights go off ... if you're lucky some drug-inspired vision might visit your sleep.

What Volker thinks:

US-American hip hop is not my usual subject but I quite enjoyed 'K-Minus Initiative' none the less and felt inspired to investigate further into the subject of hip hop in general. The first time you hear the album it is hard to distinguish between the numerous interesting samples and the actual rap because both elements go together so exceptionally well. In fact, I think the CD sounds rather like a soundtrack with the film going on in your head while you're listening to the music. At the same time, the music doesn't need your full attention but also serves well as a kind of background. The texts sometimes deserve the 'Parental Advisory' sticker, and especially the track sung by DJ Goodbeats wallows in bad language. Pay special attention to track number 8, it contains an extra gimmick: our rapping C64 fans have dragged out an old 80s commercial for the C64 hidden away in the farthest corner of the bottom drawer of bad taste's cupboard.

Commodore 64



Picture: Sebastian Bach

Der C64 on VIVA!

For C64 Fans with access to the television-music channel VIVA (the German counterpart to MTV), there was a considerable surprise last April when the video "Was ich weiss" (English translation: "Which I know") by the group Mikron 64 was first played. You may have heard the song on the radio. The sound of the typical triad-bells attributed to Martin Galway that have become famous in the C64 music scene as the "Galway-Effect" would be hard to miss. If this escaped you, the latest title of the album by Mikron 64 will make your ears ring: SYS 49152.

Mikron 64, recent gigs:

9-15-2000 - Mikron 64 as guest on NDR4 (NDR = Norddeutscher Rundfunk, engl.: Northern German Broadcasting Service)

Fall 2000 - Premiere of the Vinyl- 12" "Etwas Zeit" under storagerecords

10-15 to 10-25-2000 - Tour together with Kiss Kiss Bang Bang. More on their homepage!

by Wanja Gayk

SYS 49152 calls a machine language-program on the C64 at $C000, directly after the BASIC-ROM. Characteristically, this address is chosen in order to activate BASIC-expansions or small machine language-programs, so it won't disturb BASIC. As well, Chris H?lsbeck used this start-address for works of music created with his famous sound-monitor...

For us, this was reason enough to interview the group Mikron 64 ...

GO64!: What is Mikron 64 ?

Mikron 64: That's me, Markus.

GO64!: On the Mikron 64 web site, the Commodore is not named 64 directly, but it's unmistakable that it's a source of inspiration. Explain to us a little about the roots of your computer passion...

Mikron 64: I have been interested in computers in general since very early on. I remember a series of Daniel-D?sentrieb-Mags, there were always articles for children that were easy to understand. It also had a notebook of older computers with their history, etc. The series also explained how to write simple BASIC programs. I was so intrigued with this topic that I read through the notebook again and again and typed in the programs listed. Since the C64 was represented as having no equal, of course I absolutely had to have one. Incidentally a nice example, the C64 can be easily used as "toy" for children that's widely available around the world.

GO64!: Do you have any contacts in the C64 Scene, such as demo - or Cracking groups?

Mikron64: No.

GO64!: Were there any C64 people or groups, that impressed or influenced you ?

Mikron 64: I was always quite impressed from Rob Hubbard and Martin Galway, but they haven't really influenced me. Maybe in the respect that they've created certain sound standards for the programming of music on the C64. There was a phase where I wrote a few tracks with the 64 and these standards had a certain meaning for me of course. It comprises only a relatively small portion of their influence.

GO64!: If you could imagine that you had started with "normal" music rather than the home computer... How do you think you'd put video game and studio sounds together?

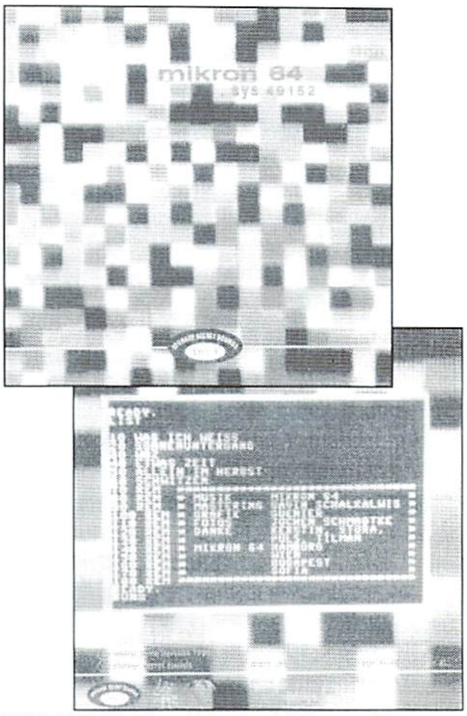
Mikron 64: Not exactly in a spectacular way. Behind Mikron 64 isn't the idea to pay homage to the C64; this happens on the side. I write songs, and then tone it with electronic methods. The 64 creates sounds that are used along with other sounds, and I use ist graphic aesthetic for the presentation of my music.

GO64!: What kind of equipment do you use in order to generate the sounds for the music? From where do the sounds come? It seems, you have also used parts from a few known games...

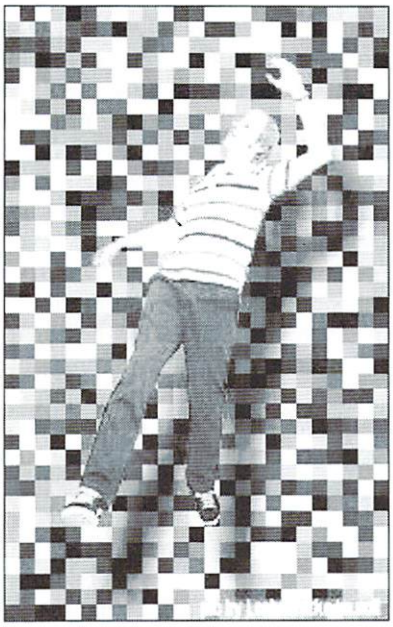
Mikron 64: None of the C64 sounds come from games. They are done with a program named "Sound-monitor" and then played live, therefore picked up unsequenced to the existing arrangement. Furthermore, only some sounds originate from the C64, and when mixed with our other sounds you may not be able to tell which are from the C64 and which aren't.

For example, the C64 was used only in the tracks "Was ich weiss" and "Sonnenuntergang". All other sounds are from a Roland Juno-106, a Yamaha TX 802, a Kawai K4 and a Casio-Sampler. The video-game-feeling that my songs have don't originate from the 64, but rather as a combination of sounds.

GO64!: Your lyrics are in German. Why is it that you haven't used English as it seems the computer-language quintessential language?



SYS 49152: The debut of Mikron 64 schows Commodore style on the back side.



Marcus in 16-Colorspace

Mikron 64: I have answered this already many times:

The overall picture of Mikron 64 admittedly works through using the C64 and video game aesthetics, and of course through the "64" in the band-name - quite harmonious and uniform.

Nevertheless, it wasn't a calculated concept to put the C64 in the center from the start. At the beginning, the songs always stand with me on basis of chant and piano or guitar, and the lyrics are only in German. At this stage, my music isn't "computer-music" and therefore I've made the decision to not to write my lyrics in English.

GO64!: The video for "Was ich weiss" is composed from C64 game and film-sequences. How did this idea occur to you and will you use this concept again?

Mikron 64: I've been very happy to have had this idea, but I don't want to over use it as it will wear out too quickly. There will be another video on a Maxi that will appear in autumn, but after it, I will have to think of something else for future videos. The idea had it's origin as I feel live appearances of electronic acts can be wearing visually. My next thought was then to show videos; as many other acts do, but many of those don't have much to do with the piece of music it's created for. Since I didn't (and still don't) have the money for approximately ten videos to be created, I got the idea of using cuts from video games.

GO64!: Can you share with our readers outside of Germany what the Mikron 64 lyrics are about?

Mikron 64: OK! Computer voices are probably the most striking feature of Mikron 64. In melodies which spin around themselves, they sing of loneliness, longing, love and euphoria. They are always optimistic, slightly naive, and sometimes not quite serious.

GO64!: Seriously, Would you have suspected that you would be interviewed in the year 2000 by a C64-Magazine?

Mikron 64: Of course not. Until recently I didn't know there was still such as big C64 scene.

GO64!: You're not the only musician who is still connected to the old C64. Have ever heard of Welle Erdball ?

Mikron 64: No.

GO64!: Could you imagine ever appearing with several C64-orientated groups?

Mikron 64: If the bands - apart from C64 sounds - are musically in common and I like the music, then why not?

GO64!: What should we expect of Mikron 64 in future?

Mikron 64: As mentioned, in the autumn a Maxi will be released. It will contain three new tracks besides the title piece "Etwas Zeit", that was already represented on "SYS 49152". With "Etwas Zeit", there will also be a video similar to the previous one with cuts from C64 games. From the 15-29 October 2000 we will be on tour with 'Kiss Kiss Bang'. The exact details can be found at www.Mikron64.de

GO64!: We thank you for the interview. Do you have another couple of words for Commodore64 fans?

Mikron 64: The life continues.

http://www.Mikron64.de

Titel: SYS 49152
Interpret: Mikron 64
Label: Edition Stora / Freibank 1999

Distribution: Edition Stora
Telephone/Fax ++49 (0) 4124 937281
E-Mail: stora@stora.de
Internet: http://www.stora.de

Uncle Wanjas Opinion:

If you've always liked the playful, groovy and small sounds of video games such as Giana Sisters, you'll probably love this CD. The synthetic voice accompanying the whole, fits like the proverbial fist in the eye: Better you cannot hit the voice, as it makes you think of sprites such as Rockford, Wonderboy, Bomb Jack or Pac Man. It's as if the tiny video game hero could sing with his artificial voice to his own in-game Sound track. The CD is suitable background music for computer club meetings or the morning after a night on the town. There are moments of the unreal, but not seriously enough to get on your nerves on a bad day.

Result: If you don't always want to hear the same mash and want to have one different item on your CD shelf, get this CD. There's nothing it compares to. SYS 49152 is a declaration of love to the small heroes in its small worlds from silicon and information.

Volkers Opinion:

Mikron 64 doesn't fit into a determined scheme, but mixes various influences into an unusual combination, which one can call earliest Vocoder Pop with loans from the 1980's and video games.

The computer voice seems very strange in the first moment and a bit mysterious. But once "acclimatized" it becomes more interesting. The 6-Track-EP has particularly strong moments with its faster numbers: "WGZ" e.g. convinced with its cool, thoughtful Synthie Pop, "sweating" against it, like a feverish Disco hit.

Coverstory

Interview with Hard Hat Mack

Picture: Sebastian Bach

A lot of C-64 fans like to switch on their machines just for listening to SID tunes once in a while, but what they don't ever expect is to hear them when they switch on the radio. If life somehow takes you to Davis, California by chance, you will get a big surprise, because this is Hard Hat Mack country and things are a bit different there...

by Wanja Gayk

GO64!: Hello, Justin! In the introductory text I already said that things are a bit different in Davis, California, and people may have a rough idea about what I was aiming at with my words. Would you tell them why exactly Davis is Hard Hat Mack country??

Justin: Oh boy, that's probably an exaggeration! I think, though, that building interest in this music is an important and fun thing to do, and I'm very happy to be doing it. I haven't been this obsessed with a hobby in a long time, maybe forever. Saying that this is "Hard Hat Mack country" makes me feel a bit self-conscious, because all I can safely take credit for is that I've played a lot of SID music on the radio in this area. The real core of the scene are all the people actually making this stuff, the demo groups and others, like the HVSC (High Voltage Sid Collection) crew. The radio thing is about developing a larger fan base for the music--a good thing, of course. Maybe one could say that a northern California scene is developing (or redeveloping?) a lot.

GO64!: When you first got the idea of making such a radio show, did you think it would become reality one day, or did you think that this was just way too crazy?

Justin: Since 1997 I had been doing a regular show on KDVS, playing electronic and experimental music. A friend of mine, who I'll just call communist64 and who is also a C-64 enthusiast, made me a CD-R of some tunes, and the obvious thing was to put the thing on the radio. It was a one-off at first. A few weeks later, after the Shark contacted me and I saw that a lot of this music was easily available, I figured why the hell not just do a weekly show? I didn't think it would be all that crazy, partly because KDVS is a station where doing something different is considered normal. Also, I think radio is the perfect forum for SID music. I was blown away by all the immediate interest in it! And I was grateful that people into the scene were so willing to help me out and re-introduce me to what was going on. I felt like a bit of an outsider at first, but I'm more comfortable now.

GO64!: I bet the people in the radio station looked quite baffled when you finally carried a C-64 into the room...

Justin: Imagine people's reaction when you run around a track with one (see my home page for the picture)! Yeah, people are sometimes confused by the whole thing, which I don't mind. I do mind if they treat it condescendingly, as if C-64 music is somehow not legitimate. Recently I played a couple of small sets of SID tunes in between a couple of local alternative rock bands at an outdoor university festival. It was so out of context, as I was there playing Swedish drum and bass songs and Norwegian funk on a C-64. A couple of people were cool and came up to talk to me about it, but mostly people were walking away or staring blankly. When I was starting the second set, one kid even had the nerve to say, "Oh no, not this Nintendo crap again." When I pointed to the Commodore logo, he couldn't believe it was an actual Commodore. It made him smile!

GO64!: Before I continue asking about your show, let's get back to the beginning of the story. Would you tell the readers about your "first contact" (with computers, the C-64, etc..)?

Justin: With computers, I think it was probably the Apple II, which I was introduced to in elementary school. The thing I remember most about that machine was the "Eliza" program, which was that computerized therapist thing. My first contact with the C-64 was somewhere between 1983 and 1984. A school friend of mine got one. Oddly enough, the first thing I remember seeing him do on it was paint an American flag on the screen. Incidentally, the first time I saw Pac Man was in a Shakey's Pizza parlor on the way to a trip to Hollywood to watch a taping of "Happy Days." I just felt I had to mention that.

GO64!: Does your nickname "Hard Hat Mack" have a special meaning?

Justin: It's just the name of the old video game. The C-64 version came out in 1983. It's a good DJ name because it expresses excitement, toughness, action, you know. Ironically, the game itself didn't have very

spectacular music. In fact, the music played only while your guy (Hard Hat Mack) ran around. I can't remember if I ever got past level two on that thing!!

GO64!: Did you ever have contacts to the C-64 demo/cracking scene ?

Justin: Only very indirectly. I knew very few people where I lived as a kid (Riverside, California) who were into all this. I did know one kid who was heavily into BBS activity and always downloading tons of stuff, but I never quite had the courage, I think, to get into that. It was rather intimidating, I guess! I was always impressed by the insane demos that crackers would stick at the beginning of a game.

GO64!: So, what is the actual equipment you use for your shows?

Justin: Usually these days it's my Pentium (ughhh) with the HardSID sound card, the SIDPlay program, and the HVSC. I run the HardSID through a Rane equalizer to shape the sound a little better. Happily, when I'm playing this stuff on the radio, I have the pleasure of hearing it in the studio on professional equipment, so the sound is incredible!

GO64!: Did you have any problems connecting a C-64 to the studio equipment?

Justin: Yeah, did you hear those embarassing shows where I couldn't get it hooked up correctly? That was kind of a nightmare at first! With the help of some friends, though, I managed to get it sorted out okay. I've recently begun to use a turntable mixer in between my HardSID and a C-64C (with the 8580 chip) so that I can more properly crossfade the tunes. Maybe a little beat-matching, mixing, etc. will come next.

GO64!: For those who haven't yet heard your show, could you briefly describe how such a radio show is made, what you're talking about in this show (if you're talking at all), etc.?

Justin: The show wouldn't be possible without the High Voltage SID Collection (that's where I get most of my material) and the SIDPlay tool. I spend a few days arranging a playlist, and then I just go to the studio, hook up the equipment, start SIDPlay, and the rest pretty much follows. I don't talk too much, because I only have an hour, and I want to fit in as much music as I can. When I do talk, it's usually just to announce the names of the tunes and the composers. On a few occasions I have interviewed famous people, like Rob Hubbard and Tim Follin, as well as the Shark, who founded the HVSC. I'd like to do more such interviews in the future.

GO64!: The "problem" with C-64 sound is that it is in mono. Do you keep it in mono, or do you somehow convert it to a pseudo-stereo signal for your shows to get a better sound?

Justin: I honestly much prefer the mono sound over the pseudo-stereo sort of effect! So, it's always mono and will probably stay that way.

GO64!: I bet you have played hundreds of C-64 sounds by now; do you still have favourites, or are there just too many good ones to remember?

Justin: Well, I'm still at the stage where I'm constantly discovering more and more tunes that I really like that sort of become instant favorites. Some old favorites kind of get forgotten but not all of them. Right now I'm listening to a lot of funk, disco, techno, and prog rock style tunes. Stuff that has an exciting sound to it and that has complex arrangements and rhythms is the sort that I'm most interested in hearing and playing. There's the nostalgia factor, too, but it's not as important to me as just well-written music.

GO64!: Do you get feedback on your show? What do people say about it?

Justin: When I first began the show, I posted a few messages to the newsgroups (alt.c64, comp.sys.cbm, comp.emulators.cbm). The response was overwhelming! I had no idea so many people would support such a project; in fact, I really wasn't aware that the C-64 scene was so alive and strong! Nearly all or perhaps all of the feedback I've received has been extremely positive. Usually people just say, "SID music rules!!" or something like that. Even people who had never heard SID music before listening to my show have been enthusiastic.

GO64!: What is so special about the SID sound in your opinion? I remember that years ago my mom always told me to turn down the volume of this horrible music. C-64 freaks seem to see (better, hear) that from a different perspective...

Justin: SID music was among the very first stuff I was interested in listening to. I remember once, back in the mid-eighties, trying to get a friend to listen to Ratt and Twisted Sister, while he wanted me to appreciate Paul Norman's "Forbidden Forest" music. Fifteen years later, it's pretty clear what has had more of a significant effect on me (ha ha). Beyond any effect of nostalgia, I think that the SID sound just has inherent appeal. I genuinely enjoy the sounds and

textures. I appreciate the simplicity on the one hand, and the complexity which people manage to accomplish on the other hand. I like the fact that people aren't afraid to make fun-sounding music on it and that they manage to create very sophisticated musical pieces using (seemingly) very primitive technology.

GO64!: Have you ever listened to a song by Welle:Erdball? What I'm about to ask is whether you think the SID has a good chance for a revival in the commercial music industry. I'm just thinking of the SIDstation Synthesizer...

Justin: No, I must admit I haven't heard any song by that artist. I would like to! Regarding your next question, I'm determined to help bring about the revival of the SID! In the techno/electronic scene especially, it has the potential to really shake things up. It easily dominates, in my opinion, over the long-worshipped boxes like the Roland 808/909/303, and eventually people are going to want different sounds. The SID is a great candidate. Projects like the SIDstation and HardSID are awesome! Whether people will be making a living by making SID music in the commercial industry is uncertain, but anything can happen. I'm confident, at least, that popular interest in the SID will continue to grow in the near future. If more SID radio shows happen around the world, too, things will progress faster.

GO64!: For how long do you plan to continue the show?

Justin: Far off into the future. Even if I move out of Davis, I'll try to find another station on which to do it. Beyond that, I continue to have faith that other people elsewhere will start similar shows elsewhere. I'd be their number-one listener!

GO64!: For all readers who might come through Davis, California, now you've got a chance to tell them the time, frequency, and your URL.

Justin: Tuesday nights at 8:00 on 90.3 FM and www.kdvs.org. My home page is **www.kdvs.org/6581SID/**.

GO64!: Any last words to the C-64 World?

Justin: I'd like to thank everyone who has supported my show and given me much-needed help and advice. The C-64 world is the coolest! Unite and conquer!!

GO64!: Thanks for the interview.

Picture: Sebastian Bach

Welle:Erdball Starfighter CD

We introduced you to the music group "Welle:Erdball" a while back. This group has been "dormant" for a while, but their new CD "Starfighter F-104G" blows the rest of their music away. It's currently on the D.A.C. (Deutsche Alternative Charts, i.e. German Alternative Charts) Top Hits list. This CD brings the C64 is back into the limelight.

by Malte Mundt

As we mentioned in our interview with "Welle:Erdball", they play the role of a host of an imaginary radio station, and want to inform the public. Of course, they want to broadcast more than just unnecessary babble (hostages being taken, fires etc.). No, they want to broadcast things that will make people think. This is the goal of their new CD "Starfighter F-104G", named for one of the biggest and increasingly forgotten disasters in the history of military aeronautics.

In 1956, the American firm Lockheed introduced the "Starfighter", a military aircraft which was conceived to fly at twice the speed of sound and deliver atomic bombs. Lockheed sold almost 1000 of these vehicles to the Federal Republic of Germany. Six years later, a crash took place during a training flight. This would be the first in a series of accidents (reaching a peak of 26 in 1965 alone). The cause of the crashes, not including those caused by military battles, was a somewhat faulty, overly complex technology. This was a blatant sign to the people who had over estimated themselves once again and believed that anything could be achieved with technology.

Due to the Starfighter affair, leading politicians of countries employing the vehicles were increasingly put under pressure. Ruling politicians in Germany were suspected of having accepted kickbacks. Only after 20 years of service was the last Starfighter decommissioned, the ultimate tally: over 250 crashes and 100 fatalities, one of whom was Joachim von Hassel, son of the National Defense Secretary who had arranged the purchase of the Starfighters...

Welle:Erdball's song "Starfighter F-104G" is sung from the perspective of this pilot, who flew to his death high above the clouds, imprisoned in a coffin of steel moving at twice the speed of sound. The musical instruments and onboard computer voice serve to create the mood of this pilot's fate perfectly.

In addition, the CD contains the song "Ich bin nicht von dieser Welt" (I am not of this world), a song with which many can identify, especially if they somehow feel strange and don't quite understand what's going on around them. Of course, the CD wouldn't be complete without a love song: "Verlieb Dich in mich" (Fall in love with me), in which electronic sounds are supposed to lull a potential romantic interest. Another highlight certainly has to be "23", named for the film "23 - Tod eines Hackers" (Death of a hacker), which dealt with conspiracy theories on a high level. An SX-64 is the hacker's computer in this relatively new film, which explains why the song "23" contains the words "Commodore 64 - ist das richtig?" (Commodore 64 - is that right?). To make this CD complete, a second version of "Starfighter F-104G" was included. This version conveys a less radical, more "soaring" atmosphere. The tempo of the music on this CD is just right for dancing. If you've ever experienced Welle:Erdball live, you won't be able to resist this one (Tour dates are listed on their homepage, below.)

The C64 and its SID sound chip is used as the main instrument more than usual. Even the voice of the popular speech synthesizer "SAM/Receiter", used to create speech on a C64 with 64K RAM and 1 MHz, is often heard on this CD. Of course, only true C64 fans would know that the computer voice in "Starfighter" is from the Microprose simulation "Solo Flight", which made its mark on the C64 long before Fighter Bomber or any others.

Welle:Erdball's musi shouldn't be confused with that of C64 remix CDs like "Back in Time", as they present something completely different. As a comparison, something like Kraftwerk would be more in order. In my opinion, however, "Starfighter F-104G" surpasses such classics.

The "Starfighter F-104G" CD isn't just available through "unofficial channels", and it's not just for "insiders". It should be available in any well-stocked music store (spotted in Mediamarkt). With this album, Welle:Erdball is helping to fulfill GO64!'s mission, i.e. to give the C64 some more publicity. The C64 isn't just named on the cover of the CD, but it's also on the CD - in the song "23" (see above), which was produced completely with the SID.

Welle:Erdball adds a new dimension to the C64 with their new CD. The intelligent, critical lyrics along with the ingenious flair make this an absolute must-have.

Homepage (MP3s available for download):
http://www.welle-e.de

(gb)

WORLD WATCH

by Rainer Buchty

This month's focus: Chip Cloning

HELLO WORLD,

cloning the CSG custom chips is a regular topic in some hardware-oriented forums. Enough reason for me to take it up this month and explain the possibilities and impossibilities.

Chip-cloning - what does it mean anyway?

Cloning a chip means making an equivalent one - either by emulating it or by copying it. The first means to build another chip which ideally behaves identically, while the latter means to re-implement the original circuit design.

But what components can we use to clone a chip, you might ask? I don't think that any of us has access to a chip factory (a foundry). And the production of customized chips (Application Specific Integrated Circuit - ASIC) is far too expensive for small production runs, only profitable for series of 1000 units and up. However, there's an alternative.

For some 20 years, there have been programmable logic chips, the so-called PLDs (Programmable Logic Device). There's even one of these in the 64, namely the PLA, which stands for "Programmable Logic Array". Today, there's a large variety of programmable logic chips available - from simple PLDs to complex PLDs (CPLD) to the highly flexible FPGAs (Field Programmable Gate Arrays). The latter two are particularly interesting, because they offer sufficient resources that would make it possible to imitate the more complex components of the C64.

And how do you do it?

As we said above, there are two alternatives - emulation and replication. In most cases, emulation will be the choice, because information on the inner construction of a chip is usually unavailable. To emulate a chip, we begin by creating a model of its behaviour which, while describing it as closely as possible, should also be as simple as possible. Usually, such a model comes surprisingly close to the actual design of the chip. A good example for this strategy is the analysis of the VIC-II chip performed by Andreas Boose and Marko Makela . Their findings had direct influence on the development of VICE, a C64 emulator for several different platforms.

But there are only a few cases in which the internal data of a chip is available. The PLA is the best example of this kind; this chip can be read out using a special device or a modified eprom programmer. With an eprom programmer, you only get an image of the chip, and the actual logic equations calculated by the PLA have to be extracted from this image. However, the result is the same for both ways. With the data obtained, a replacement PLA can be programmed, or another kind of logic chip can be programmed to behave identically.

The 1541 contains a logic array, too, which is not as easy to reproduce. However, an internal schematic of this chip can be derived by closely comparing the schematics of the 1540 and the 1541. Then, this schematic can be converted to logic equations, either manually or by using appropriate software. Again these equations can then be programmed into common, commercial logic chips.

So, what's the catch?

Now you might think, "oh, how easy!", as you look forward to new SIDs, VICs, and CIAs - but I've got to disappoint you. These chips are very complex ones, and furthermore, they contain certain design flaws. In many cases, it's just those flaws that make them interesting and enable all the programming tricks - like those used with the VIC. This means that the model must be constructed very cautiously, in order to be a close and error-free description of the component. You might argue that those models already exist - everybody can get the C sources to VICE and look at the descriptions of the C64 hardware contained in them. That is correct, but there are certain differences between a C program and a hardware description. The most aggravating one likely is the fact that computer programs are inherently sequential. You can always assume that the code lines are executed one after another. Hardware descriptions are exactly inverse; their "code lines" are processed in parallel, and sequential execution has to be enforced with certain constructs. Therefore, a source code written in C can surely be helpful as a basis to work from, but it's still a long and burdensome way to a hardware description.

Let's assume we finally have the description. The next step is to simulate it. However, building this simulation (a test bench) is at least as expensive as developing the hardware description. This is mainly because the simulation has to be realistic; it's far too easy to fall for the typical programmer's mistake, checking only particular situations which are deemed useful and incisive.

When the hardware description has been tested and probably corrected, it has to be synthesized. The description is used to generate logical equations, which are then fitted into a chip. At this point, you get the "bill" for using high-level languages; as a rule-of-thumb, the more complex the logic equations derived from a hardware description are and the more abstract your description is, the higher are the requirements for the logic chip. Of course, for economic reasons you want to go with the smallest chip possible, which will often mean that you have to change your hardware model at this point.

All in all, this is a time-consuming process. The time required to develop a complex chip is usually measured in man-years, and the companies doing so employ whole teams of developers.

Well, when will we see the new VICs and SIDs?

A very popular question, to which "Not at all!" is the very popular answer, together with the argument that it was impossible to clone them, because they contain analog components. But this argument has no basis. The analog components only prevent imitating a VIC or a SID with a single chip. Also, analog components can be digitally simulated; the catchword here is digital signal processing (DSP). But again, there is a matter of cost, which is why the filters and the final VCA of the SID, for instance, won't be integrated into the logic but built as external, analog
components.

That's the theory. The whole reason why there aren't any new CSG parts yet is simply a matter of cost. Even if you only want to reproduce a really primitive part like the PLA, in which all information is available, the production is much too expensive. You need an appropriate logic chip and an adapter board to make it fit into the PLA socket. I've made an illustrative calculation based on the M4-32/32 by Lattice/Vantis and got some estimates for the chip, the adapter board, and the production of the latter. The result - the mere material cost is in the $7-10 range, which is about four times the cost of an original replacement PLA. If a potential manufacturer wanted to make any money on a cloned PLA, he'd have to sell it at a minimum price of $15.

Talking about the VIC, SID, and the CIA, the logic chips needed to clone these are in the range between $25 and $50, even without the additional analog circuitry, and of course, the largest cost factor is the required time. And honestly, who would possibly buy a cloned VIC for 200 dollars?

The solution: OpenHardware

Cooperation. The Linux community has shown it, and the FreeCPU project uses the same approach.

There are many people who have invested an enormous amount of time into the construction of working models. A part of their findings was put down in texts (like Cristian Bauer's article on the VIC-II) and immortalized in emulator sources. However, the former is not sufficient as a hardware model, and the emulator sources are only partly suitable.

Picture Michael Steil

Therefore, I want to encourage you to work together! Work together to make more information available just like the sources integrated into the VICE system, more information like detailed internals of the chips in the form of cycle diagrams, and appropriate, detailed models! When this is achieved, the hardware people can sit together to derive a hardware description. This task requires the cooperation of many people as well, because the model generation is an extremely time-consuming process. All the cloning projects that I've known finally failed, because they were undertaken by single combatants who didn't have enough time and motivation to finish the project.

The devil's advocate

Finally, we should ask what can be achieved by cloning hardware. Spare parts would be easily available through flea-market sales for years to come - and true enthusiasts have already been collecting a whole arsenal of potential organ donors for the C64 they mainly use.

By use of chip-cloning, the supply would be ensured for all times, because a chip description is universal. If one logic chip isn't available anymore, you can simply use a newer one.

Another attractive idea would be integrating a complete C64 or 128 in only a few components; chip-cloning is an absolute prerequisite for the long-craved-for, new version of the breadbox. This would also be an opportunity to introduce some compatible enhancements, like the ZX team did with the ZX81 successors, ZX97 and ZX2000.

And finally, let's ask what benefit we would get from all this. After all, we already have a C64 with a VGA socket running at 50 MHz - and it's virtually for free, since the PC it runs on is practically base equipment in every household.

Time will tell if developing hardware is worth the effort.

If you feel inspired after reading this and would like to try working with logic chips, you can get additional information and software from the following sources:

http://www.xilinx.com

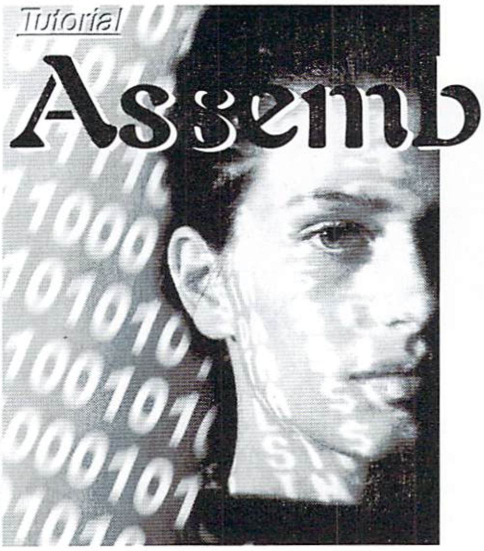
Xilinx is a company that makes FPGAs and CPLDs. For CPLD developers they offer a service that is unique, as far as I know; they have a web interface that allows you to have your own developments synthesized and fitted, which means you don't need any software on your own computer, except a text editor to write the hardware descriptions. There's a catch, however; you've got to register with them. But it's for free, and you get a t-shirt as a thanks for filling in their questionnaire.

Surely, the Spartan family of FPGAs is particulatly interesting for cloning projects. However, the required software is not for free; you need appropriate development environments whose costs range between some hundred and several hundred thousand dollars.

http://www.latticesemi.com

Lattice/Vantis is THE company for SPLDs and CPLDs. The software necessary for development is available for free download, except that the 50 megs will increase your web-surfing fees. This software must be registered as well, but it's also for free. You will get a six-months license, which you can extend free of charge after the initial period has passed.

Tutorial



Picture: Bernhard Pribil/Günther Bauer (www.hobbesgerm.at)

Assembly Language

Last time we were engaged in the basics, the representation of negative numbers and some fundamental mathematics. As promised, this time we will try to pump up our small range of numbers from just 8 bits to 16 bits.

by Gunnar 'Krill' Ruthenberg

Part 15

Our small 8 bit range may be enough for beginners, but soon it will be too little. It would be fine to increase the range - but how? Very simple. Let's say we want to use a 16 bit number. This will consist of two bytes, or 16 bits. This means our (unsigned) number may now take values from 0 to 65535 (2^16-1). Our new second byte is just the continuance of the first byte. In the first byte we store the bits 7 through 0, and in the second byte, we store the bits 15 through 8. But how do we deal with this multi-byte number?

Calculating with 16 bits

For starters, we store the number in memory the way our computer likes it, that is, with the order lower byte (LoByte) first, followed by the higher byte (HiByte). This means in the first byte we store bits 7 to 0, and in the the following byte, the bits 15 to 8. When we are actually working on these bytes, the connection between bytes is the processor's carry-flag. Let's take a look at an example now to see how a simple addition works with 16 bit numbers:

```
CLC                 ; clear carry-flag
LDA number1low      ;
ADC number2low      ; add the low bytes
STA number3low      ; store result low
LDA number1high     ;
ADC number2high     ; add the high bytes
STA number3high     ; store result high
```

At first the carry-flag is cleared (we will need it to 'carry' information from the low byte to the high byte), then the two low-bytes are added and the result is stored. Before the addition of the high-bytes we purposely avoid manually clearing the carry-flag again, for the following reason: If an overflow occures during the addition of two 8 bit numbers (in this case our low-bytes), that is, if 8 bits would not be enough for the result, then the missing 9th bit is stored in the carry-flag. Now when we add the high bytes, the carry-flag is automatically added as well. As an example: Let's add $0340 and $05C1. First $40 and $C1 (the low bytes) will be added, which will result in $0101. Since the result is bigger than $FF, the processor sets the carry flag and puts the low byte of the result ($xx01) in the accumulator . After the addition of $03 and $05 (the high bytes), accumulator is $09 (not $08!), because the carry-bit was set to 1 and and thus and extra 1 added. By the way, ADC is a shortcut for "add with carry-bit". So we get the correct result of $0901.

The same thing also works for the subtraction, but we have to use an SEC instead of the CLC to take an underflow (a result of less than $00) into consideration. It also works for signed numbers, so we have nothing else to do there. We could even increase our numbers' ranges to more than 16 bits (then we have to use more LDA/ADC/STA-sequences of course) if we do not clear or set the carry-flag manually in between each addition/subtraction step. Let the computer take care of this. But enough of addition and subtraction, now we go over to multiplication and division. During halving and doubling (these operations are very easy in assembler) we have to take extra care.

Shifting with 16 bits

If we have for example the number $17C4, and we want to double it, the program would look like this:

```
LDA number1low  ; $C4
ASL
STA number2low  ; $88 (result was $0188)
LDA number1high ; $17
ROL
STA number2high ; $2F (the set carry-flag
was taken into consideration with ROL)
```

The result of $2F88 becomes stored in number2. After the left shift the accumulator is $88 and bit 7 is shifted into the carry-flag (it's now set). During the left rotation the carry (which was still set) gets rotated back into the accumulator (into bit 0) and we get the correct result. Also in this case we may expand the routine to use more than 16 bits without problems, simply add more rotations and more bytes. In case we want to halve our number we have to take care of the sign again (see last part of this course). The sign bit is always the highest bit in the number, called the MSb ("most significant bit"), so in our case in bit 15 (the highest bit of our highest byte). Furthermore when we halve a number, we start with the high byte, because the bit that is shifted out of it needs to be rotated into the low byte. We can do that like this:

```
LDA number1high
PHA
ASL                 ; rotate sign into carry
PLA
ROR
STA number2high
LDA number1low
ROR
STA number2low
```

Now we are able to carry out simple calculations with 16 bits, but what exactly is the use of the additional eight bits anyway?

The interpretation of our numbers

The usage of these 16 bits depends on us alone, because the computer just calculates blindly while we interpret the results. For example we could say our numbers run from 0 to 65535 if unsigned, or from -32768 to 32767 if the numbers are signed. But what if we interprete the high byte as a number to the left- and the low byte as number to the right of a decimal point? Then we have a fixed point number with a precision of 1/256. Now the lower byte represents values from 2^-8 up to 2^-1 minus 1 (that is, 0/256 to 255/256)... So, what is the use of it?

The advantages of fixed point numbers

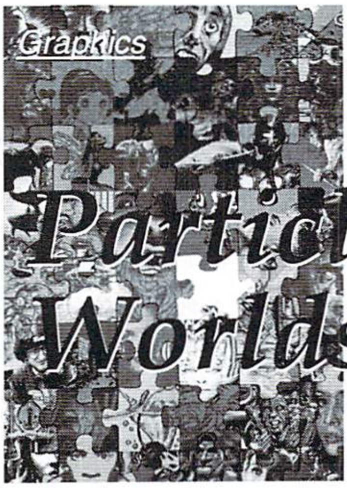
Of course we get a higher accuracy. An example for the use of it: Assuming we are using an NTSC machine, which has a video refresh of 60 FPS (60 Hz), let's say we have an animation in memory, which consists of many separate images. We want to play this animation at 45 FPS. The normal way to do things is to synchronise our program to the refresh, but if we were to show a new image for each refresh, then the animation would run way too fast at a blazing 60 FPS. If we show a new image every other refresh, then its too slow at only 30 FPS. So, we have to show the next image of our animation every "one and a half" refreshes, or 45 FPS. Here we benefit from our 16 bit fixed point number. In the beginning its value is set to $0000, and we'll treat the high byte as the number of the image to be shown next. If we wanted to run the animation at 60 FPS, we would add $0100 to our number on each refresh (since we are synchronised to it). If the animation should run at half this speed, we would add $0080. But what we need is a value halfway between these two, so $00C0 is a good choice. Our fixed point number was set to $0000 when we started, and we will be adding $00C0. So, our number becomes $00C0, then $0180, $0240, $0300, $04C0, $0480, and so on. If you pay attention only to the high byte (our image number), you'll see that it becomes $00, $00, $01, $02, $03, $03, $04 and so on. The speed of the animation is thus set to 45 FPS.

More than 16 bits

As already mentioned above it is easy to expand the routines to use more than 16 bits. Then all we have to do is to decide where our decimal point should be. For example we can use 24 bit numbers with two numbers to the right of the decimal point, which gives us an accuracy of 1/65536. But the advantages of it are very limited when it comes to addition, subtraction, halving and doubling.

In the next part of this course we will deal with multiplication. The precision of fixed point numbers will be more usefull there.

(ad)



Graphics

Particle Worlds

Part 7

In the next two parts of our graphic course Deekay of Crest will help all those people which until now have hesitated to use the possibilities of the very special graphic formats UIFLI and SHIF (Super Hires Interlace FLI). Especially interlacing in Hires mode will be covered.

by Daniel "Deekay" Kottmair

I've been thinking about this for a long time, but since no one of you graphicians out there seem to get into any Interlace editor besides Drazlace and Gunpaint I will finally "share my secret knowledge" about interlace with you! Sounds pretty arrogant, doesn't it? Nevertheless: Welcome to my Interlace Workshop!

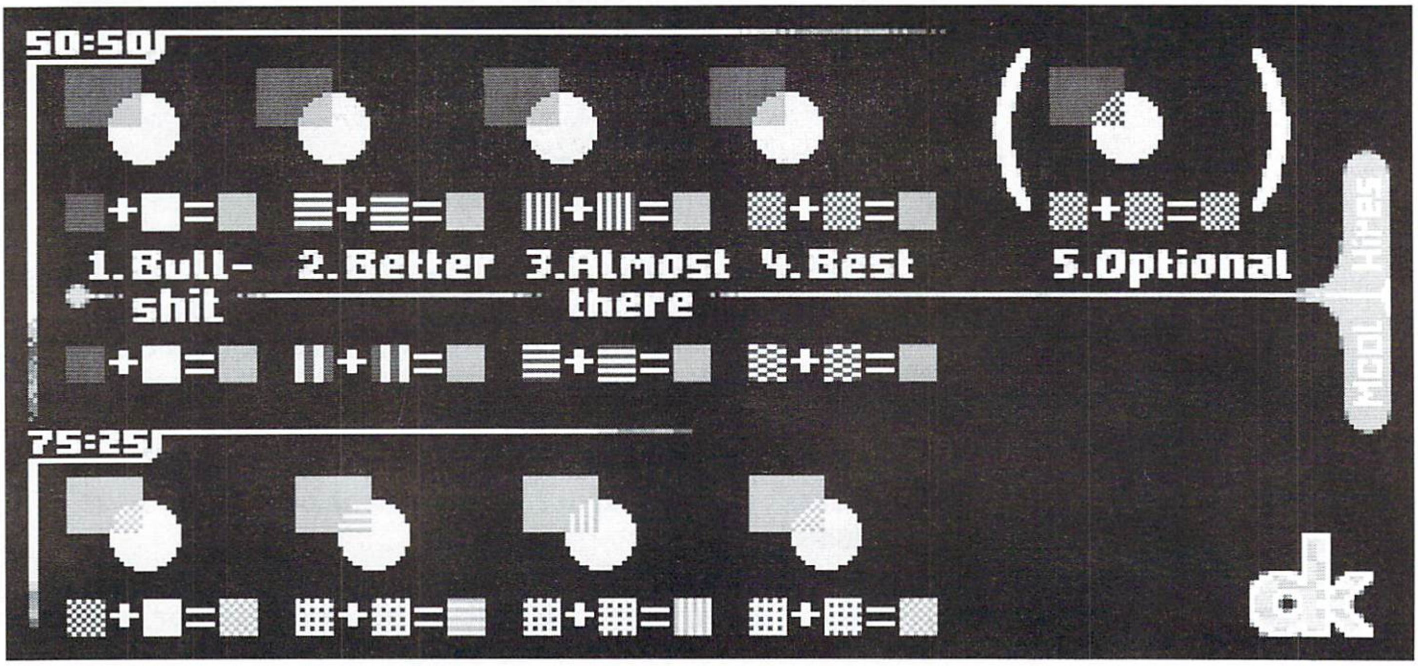
I've been pixelling a file which should be on the cover disk that explains all the things I talk about here. You will find the picture printed here as well, but the real effect you can only see when viewing it on a C64. The best to do is to have the picture on screen while reading this article! In the first chapter, at which we will look at today, we want to focus on mixing colors 50:50 in Hires and Multicolor mode.

1.1 Hires

Let's begin with Hires: Our main aim is to make it flicker as little as possible. So possibility 1 explains how NOT to do it: putting one color completely into the first bitmap and the other into the second bitmap. On most monitors/TVs this extremely flickers. If you ever worked in IFLI this would represent drawing alternating vertical lines when mixing colors instead of the commonly used chessboard.

So we move on to possibility 2, switching alternating horizontal lines. This is how IFLI normally looks when displayed. It's better than possibility 1 but still, since C64 pixels are not exactly squared (they are more high than wide) and due to the monitor's nature of blurring horizontal stuff less than vertical, I say this definitely flickers more than e.g. possibility 3, which is alternating vertical columns. This is almost the same as my favorite method, version 4, but I would still say that possibility 4 flickers less (version 4 means alternating chessboards).

Of course you could always use version 5, which doesn't flicker at all, it's just a plain



The effect of the combination possibilities can only be seen on a real C64/128.

GO, GO MULTIMEDIA!!
ADOBE
Suckers
Multimedicus
dk

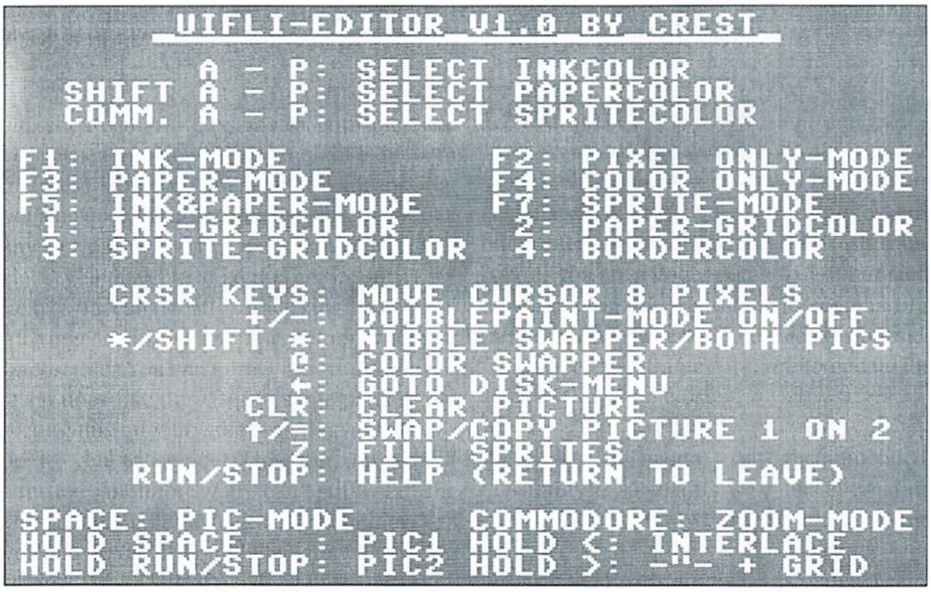
An example for the graphical possibilities is integrated in the editor

chessboard that's the same in both pictures. Or vertical columns that are alike in both pictures. However, this method means that you can't use the color like in version 3/4, like any normal color, but instead it means that since you have to mix the colors in a 2x2 field, you can't really draw single-pixelled. Making details using these colors is almost impossible: You can't draw nice roundings or steep lines like this. But in some parts of the picture this might be the better advantage, it's up to you to decide! However, on a well adjusted monitor you can still see the chessboards/columns - with technique 3/4 you can't! And after all, what are we working for in UIFLI/SHIF if we don't make use of its advantages - you could just load your favorite Hires-FLI editor instead then!

1.2 Multicolor

This might be interesting if you work e.g. in "Drazlace" without $d016 shifting or any other interlace editor without $d016 shifting. Also if you are a coder doing 4x4 dither stuff I still suggest the use of interlaced charsets (no additional rastertime needed for routines which need two frames or more to update the screen!), so listen closely!

Possibility 1, again isn't recommended. Possibility 2 is in this case the alternating columns, cause now we are in Multicolor, and in this case possibility 3 is the better one, because the pixels are twice as wide! Again, I'd favor alternating chessboards (4) as the best possibility, this time even much better than possibility 3 (line wise (IFLI)).



```
           UIFLI-EDITOR V1.0 BY CREST

            A - P: SELECT INKCOLOR
      SHIFT A - P: SELECT PAPERCOLOR
      COMM. A - P: SELECT SPRITECOLOR

F1: INK-MODE                F2: PIXEL ONLY-MODE
F3: PAPER-MODE              F4: COLOR ONLY-MODE
F5: INK&PAPER-MODE          F7: SPRITE-MODE
 1: INK-GRIDCOLOR            2: PAPER-GRIDCOLOR
 3: SPRITE-GRIDCOLOR         4: BORDERCOLOR

    CRSR KEYS: MOVE CURSOR 8 PIXELS
          +/-: DOUBLEPAINT-MODE ON/OFF
    */SHIFT *: NIBBLE SWAPPER/BOTH PICS
            @: COLOR SWAPPER
            ←: GOTO DISK-MENU
          CLR: CLEAR PICTURE
          ↑/=: SWAP/COPY PICTURE 1 ON 2
            Z: FILL SPRITES
     RUN/STOP: HELP (RETURN TO LEAVE)

SPACE: PIC-MODE         COMMODORE: ZOOM-MODE
HOLD SPACE    : PIC1    HOLD <: INTERLACE
HOLD RUN/STOP: PIC2     HOLD >: -"- + GRID
```

The UIFLI editor help menu

See you in the next part, in which we will look at fine mixing of colors!

(vr)

The original text was published in the column "Particle Worlds" in the disk magazine "Newspress", issue 19. Thanks for the cooperation.

Picture: Sebastian Bach

GoDots unterschiedliche Bildspeicher

Due to the large number of inquiries I've received regarding GoDot I'll reiterate: GoDot's image data and the image displayed on the screen are located in different sections of memory, and don't necessarily concur. But to the user who isn't savvy to this fact misses out on a lot of great opportunities made possible by this situation.

by Arndt Dettke

Both manuals already allude to the subject, but are written in an unclear manner. The English version contains quite a few errors due to editorial abridgments during translation. The purpose of this workshop is to clear up the confusion about this whole matter. You can also find an a somewhat shorter version on the Internet at:

http://members.aol.com/howtogodot/wshptuto.htm

We're going to use a wide variety of GoDot modules today, including: ldr.4BitGoDot, ldr.IFF, mod.ApplyColors, mod.ApplyDither, mod.ClipWorks, mod.FrameClip, mod.Rendered2Raw, mod.Scroll, mod.Squeeze2Clip, svr.4BitGoDot and the functions Redisp, Preview and Undo (the last of which requires an REU). As promised in issue 10/99, ApplyColors appears in a new version: After use, the changed palette automatically returns to the default values, saving a few mouse clicks. Our project for today entails creating one image from three, all of which can be found on the cover disk (see pictures 1 to 3). The new mod.ApplyColors is also included

Loading, ApplyColors and Scroll working together

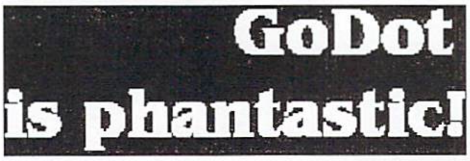
First I'll show you how to produce color text effects with GoDot. In order to prevent going into too much detail (this doesn't really relate to our subject today), I'll assume that you've already prepared the text as white on black with a paint program (using GoDot to "write" text will be explained in another issue). Position your text as I have, so that your results will be similar to mine. Alternatively, you could use the included 4-bit file "text.clp".

```
Inst: ClipWorks
Execute
Full
ClrClp
(Pick:) black
Inside
Leave
```

This deletes any remaining pictures from memory, and also cleans it of any "startup rubbish".

```
Hor: 28
Ver: 9
Row: 16
Col: 12
Accept
```

If you enter this into the corresponding ClipWorks number field, there should be a Clip in the bottom right corner of your screen.



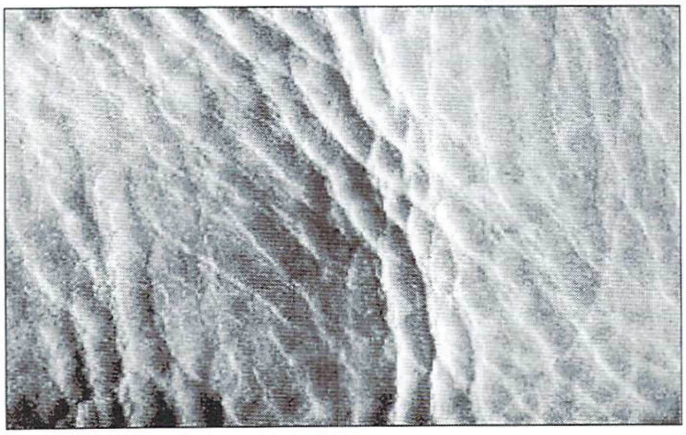
It's size should correspond to the "text.clp" file. If you're working with your own text in a 320x200 field, don't worry about the last step. The Clip will be loaded six times, shifted and colored in. If you're using an REU with your system you should load the applicable module into RAM, as this will save you a lot of time. I'll assume that the loader 4BitGoDot is installed ("Load: 4BitGoDot") and that the loading destination of the clip is set ("Load to: Clip").

Once more...

```
Load
Compose Foreground
"text.clp"
Palette
(pick:) blue (dark gray,
red, light red, turquoise)
(set:) white
Accept
Inst: ApplyColors
Execute
```

Up to this point, we've loaded the Clip and colored the text blue. In the other iterations, the other colors get a chance. Now it will be shifted so that the next text seems to lay above it:

```
Inst: Scroll
Set Amount: 6
Direction: left
Execute
```



```
Set Amount: 1
Direction: upward
Execute
Leave
```

...through all the colors.

The last two instructions will be repeated four times and only change the color. Last, we load the Clip and keep it white by not coloring it.

```
Load
Compose
Foreground "text.clp"
Display
```

The text file is finished, and must be saved as 4Bitimage. To do this:

```
Save: 4BitGoDot
Save "texteffect.4bt"
```

Image display and 4 bit memory working together

We will now shrink ("scale") an image down to the size of the screen so that it seems free to float above a sub layer (a so-called "texture"). The result will fill the entire screen. Let's begin with the texture which is in the IFF image format:

```
Load: IFF
Load Gray Scale
"LEATHER1.IFF"
Inst: ApplyDither
Execute
Dith: Pattern
Mode: Multi
Reduce to: 9 Cols
Accept
Display
```

At this point, the displayed picture and 4-bit memory are in agreement. The data in the 4-bit memory must be saved (either to a disk or an REU), because we will need it in exactly this form later on. Otherwise we'll load the picture which will soon learn to float:

```
Save
(Choose Drive or RAM)
Save "Undo 4Bit"
Load Gray Scale "NAGEL01.IFF"
```

After loading, the displayed picture and 4-bit memory are located in different positions. Go ahead and try it out: click the "Redisp" button (which will display the previously displayed image) and then activate the 4-bit Preview Window (located under the name of the picture). Both show completely different things. With an REU and the function "Undo Rendered" when saving, you can protect the image in the Redisp memory before it's lost. You should use start to do this at every chance

you get, after doing the types of things that we're doing today. The image is now shrunk and displayed in the upper left corner.

```
Inst: ClipWorks
Execute
Hor: 20
Ver: 13
Row: 3
Col: 4
Accept
Inst: Squeeze2Clip
Execute
```

Can you use the ClipWorks function "Shrink" instead of Squeeze2Clip to save a few mouse clicks? Actually, no. Not in this case. Shrink works on gray-scale images, whereas Squeeze2Clip deals with color images, i.e. what we want in this case. Click on Preview and Redisp once more. See? One area of memory is being processed and the other remains untouched. Now just a quick border around the shrunken picture and then I'll show you the advantage of both image memories:

```
Inst: FrameClip
(select:) yellow
Exec
Display
```

Surprised? The images look like they're blended together, don't they? Actually, nothing in the 4 bit memory has changed (click on Preview!), i.e. in the Redisp display. A clip is like an image display window in 4 bit GoDot!

```
Load: 4BitGoDot
Load Replace "Undo 4Bit"
Balancing
```

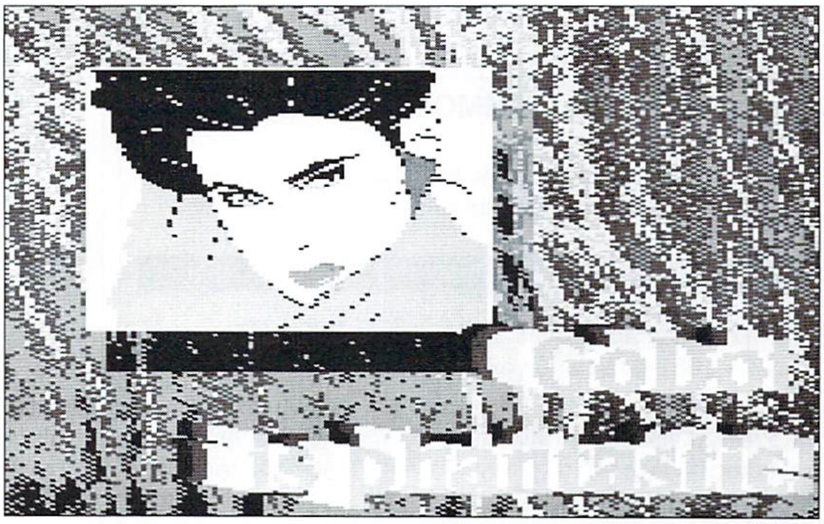
```
Brightness: -4
Accept
Inst: ClipWorks
Execute
Ver: 2
Row: 16
Col: 6
Hor: 20
Accept
Display
Execute
Row: 5
Hor: 2
Col: 24
Ver: 11
Accept
Display
```

The procedures described above plays around with loading in the 4 bit memory area; the Redisp area is altered with ClipWorks. Go ahead and try out "Redisp" and "Preview". See? Our Nagel girl is no longer in 4 bit! We see her in the Redisp display, but there are shadows below and next to the shrunken image - it's hovering. As the finale, we'll place our painstakingly created text effect over the whole thing. But both image memories must be combined first:

```
Inst: Rendered2Raw
Execute
Load
Compose Foreground
"texteffect.4bt"
Balancing
Reset
Accept
Exec Area: Full
Display
```

Until next time,

Arndt



The thing with the "boom"

A short report on my personal experiences with the "Modplayer" from the GO64! 5/2000 issue

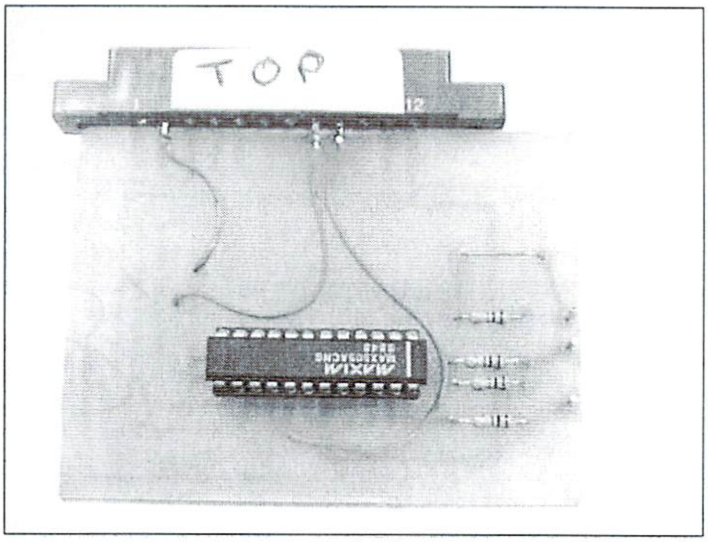
by Niko Malecki

One beautiful day Colin J. Thomson sent me the construction instructions for the Modplayer, or to be more exact, for the necessary user port hardware. When I saw the circuit diagram I couldn't help thinking: How can you ever make anything remotely sensational from a single IC and a couple of resistors?

Ever laying hands on an IC turned out to be the biggest difficulty because there is only one IC distributor in Germany. The company is called IC-Spezial (telephone: +49 (0) 5722 - 203111) and they take five to six months to deliver. But it is well worth waiting!

In the past we already showed you a way of obtaining an acceptable sound output for C64s and C128s, with the SID extension from the GO64! 9/97 issue. This type of SID player makes the C64 fit for music at a reasonable quality. But everything you ever heard with SIDs is only a poor imitation of what you can expect from Modplayers. The sound spectrum and the stereo channel separation are unbelievably perfect and instantly reminded me of those beloved fellow citizens who refurbished their cars as giant hifi stereo players, easily recognizable by the loud "boom, boom, boom" accompanying them wherever they drive ...

Modplayer as shown in issue 5/2000



But before you can enjoy all the fine qualities of the Modplayer, some requirements have to be met. The extension will only work together with a REU because the music is first loaded to the RAM and played from the RAM (the same way as PCs do it). The computer's own RAM is too small for this operation. While it works quite well with original C64s and C128s but you would still miss some of the sound purity. The distortions will eventually disappear when you insert a SCPU between the computer and the REU. Using a SCPU does not only have positive effects on the sound but makes loading the main program "Modplayer" and pieces of music much faster. Here's the proof: Without the use of a SCPU loading the piece of music we tested with will take about 90 seconds; if you use a SCPU, it will only take 20 seconds. And, well, music - as well as colors - are simply beyond description. If you want to have a fair impression of how great music sounds with the Modplayer, you will have to try it out for yourself ...

Okay, I'll let you start soldering then ... with only 20 connections to solder, you don't really need to etch a board, you could simply wire everything freely on a breadboard. However, if anybody out there wants to offer a board or even the complete kit, they're welcome to write to the GO64! editorial staff.

That's all for today, now start "booming"!

GO64! Bazar Form

Please insert the following advertisement in the next issue of GO64!

My advertisement is □ private/non-commercial (four (4) lines, 40 characters wide.)
US Dollar 3 (Euro 2.50) are □ included in cash □ included as cheque □ chargeable from my credit card (VISA, AMEX, Master Card)

□ commercial (any reasonably sized ad, US Dollar 7 (Euro 6) plus taxes

The item(s) I am advertising about are related to: □ C64 □ C128 □ Software □ Accesories □ Misc

Address
CSW Verlag
Goethestraße 22
D-71364 Winnenden
Germany

I assure that I own all rights concerning my offers.

From:

Last Name/First Name ____________________

Street ____________________

ZIP/Town/State ____________________

Credit Card no./expire date/holder ____________________

Date/Signature ____________________

Masthead

Publishing Manager: Enno Coners
Editorial Staff: Günther Bauer *(gb)*, Helmut Bieg *(hb)*, Rainer Buchty *(rb)*, Arndt Dettke *(ad)*, Wanja Gayk *(wg)*, Alexander Klock *(ak)*, Nikolaus Malecki *(ma)*, Nikolaus Metz *(me)*, Volker Rust *(vr)*, Wolfram Sang *(ws)*
Editorial Assistance: Gaelyne R. Gasson, Daniel Kottmair, Malte Mundt, Hartmut Pachl, Gunnar Tuthenberg, Nicolas Welte
Supervisor: Arndt Dettke **International Supervisor:** Robert Bernardo, Nate Dannenberg
Translations: Bettina Beck, Matt Bodnarchuk, Stephan Lesch, Ranjith Ekanayake Mudiyanselage, Kendra Thiemann; **NTSC-Fixing:** Scott Brockway, Wolfram Sang

Submitting Manuscripts: The editorial staff accepts most original Commodore-related manuscripts and programs. You must inform GO64!/Commodore World if any such programs or writings have been offered to be published or used commercially by a company other than CSW or one of it's subsidiaries. By sending in any manuscripts, programs or construction plans, the author agrees to allow the CSW Publishing House to print them. The author also agrees to allow any and all programs to be copied using whatever means is necessary (disk, online service, etc). Furthermore, CSW reserves the right to build and/or distribute any projects submitted to GO64!/Commodore World. The author shall assume no liability for damages or losses caused by the use of his/her programs or projects. Use any projects or programs contained herein at your own risk.

Layout: Günther Bauer
Cover Layout: Günther Bauer; picture by Sebastian Bach
Graphics: Christoph Bach, Sebastian Bach, Günther Bauer, Daniel Rac

Price for Advertisments: Price List No. 1 shall be considered valid unless specifically noted.

Publication: Monthly (12 issues per year).

Print: Berufsbildungswerk Waiblingen, Postfach 1129, 71301 Waiblingen, Germany

Copyright: All articles published in GO64!/Commodore World are protected by copyright. All rights, including translation and second commercialization, reserved. This publication may not be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publishers. The original authors shall retain any and all rights with regard to specific terms, passwords, and the like, unless otherwise noted.

Liability: If GO64!/Commodore World gives false information or one of the published programs or plots contains errors, liability can only be considered in case of culpable negligence of one of the assistants.

Circulation Manager: Enno Coners

2000 CSW Verlag

Address of the Publisher: CSW Verlag, Goethestr. 22, D-71364 Winnenden, Germany
Telephone/FAX: +49/(0)7195 61120, E-Mail: go64@c64.org

GO64! Bazar

Where can one buy or sell second-hand hardware, accessories, literature and software? Is there a convenient solution? Yes, there is! Check out our Classified Advertisements section, the GO64! Bazar! Above, you'll find a small registration form. Just clip it out, fill in the blanks, and send it to the publisher. For every advertisement you submit. For personal/private (non-commercial) advertisements, you may have up to four lines of text, each up to 40 characters in length. Please see the coupon for pricing and other details. We will make every effort to publish your ad in within the next issue of GO64!. If your ad doesn't make it by the next issue, it will appear in the issue following that.

For Free - Contact Corner

We're going to offer a free service to people who are looking for other C64 enthusiasts. Just send a contact advertisement in the form of a letter, FAX, or email to our editorial staff, care of: Contact Corner. The editorial staff reserves the right to shorten parts of letters which are not immediately important, in order to save space.

Misc

C128, 1581+71internal, Mon.1084S, 1571, RAM1764, StarLC24C, modules, more than 1000 disks, GEOS128, / Scanntronik- Scanner, video digitizer, C116, books and more Tel. +49(0) 5358258

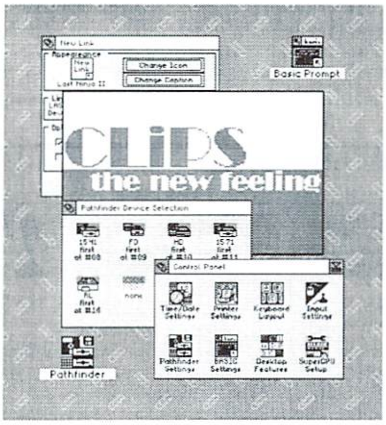
47 Magic Disk, 26 Game on; Brotkasten-CD and Brotkasten-CD Gold (original); mouse pad with C64 logo (new) together with p&p DM 59,- Tel: +49 (0) 6251-73166

Contact Corner

Join the Fresno Commodore User Group. $12 membership/$15 international. FCUG, 3487 E. Terrace Ave., Fresno, CA, 93703 USA. http://videocam.net.au/fcug.html

The editors kindly want to draw your attention to the fact that offering, selling or distributing copyrighted software is only allowed for original programs and materials. For your own protection we ask our readers not to offer or to distribute pirated copies of any copyrighted program or materials.

The publishers reserve the right to refuse advertisements that appear illegal in nature, including advertisements dealing with pirated material. Original programs which carry a copyright will be conspicuously noted as such in their accompanying text. The label on the disk (or other media) will indicate any copyright as well.



CLiPS - Corner live during its development

CLiPS, the new operating system for SuperCPU-equipped Commodore 64s, is continually being developed and improved. We recently visited the Protovision labs and took a look at CLiPS's disk drive management system, which offers never-before-seen flexibility and speed.

by Malte Mundt

Disk drive management is one of the most important elements of an operating system. Originally, the purpose of an OS, such as JiffyDOS and MS-DOS, was to act as an interface between the user and the disk drives. These OSes really were "Disk Operating Systems". CLiPS's strengths lie in its efficient memory management, utilization of system resources, and pre-emptive multi-tasking. But the control and management of disk drives of different types also play an important role in CLiPS.

Disk Drives in the Here and Now

You may have noticed that the FD-2000 isn't exactly a speed demon when accessing data while playing games. Data access from a MS-DOS formatted disk under GEOS seems to go on forever. If you think that the activity light on your CMD HD doesn't blink enough and have dreamed of faster data transfer rates between the HD and C64, you're not alone.

And who's not familiar with the problem of using four disk drives with GEOS and trying to start programs on each of the drives, not to mention the difficulty and sometimes inability to change partitions on a HD (hey, where is "CMDMove" anyway?)? And why does a 256K REU only show up as a 170K ramdisk anyway? If you've ever booted GoDot from a 1581 disk and then tried to save a picture to a native FD disk, you've probably gotten an illegal block error. These examples stress the need for good quality disk drive management.

The Future - CLiPS

CLiPS will manage all kinds of disk drives, from the 1541 to the FD-2000 to RAM disk drives, as efficiently as possible. During the boot process, CLiPS runs a fast, automatic hardware recognition routine. This routine takes less than a second to be completed, unlike Windows and its eternally dragging processes. Adding new components doesn't require complex or frustrating configuration programs, because changes are noticed by CLiPS automatically. Every disk drive is assisted by a so-called Resource Manager, which is loaded after the recognition of the device. A 1541 disk drive is obviously different from a RAMLink, which is why the use of a common loading routine would be extremely inefficient (the speed of programs like FCopy brings back painful memories). With CLiPS, every Resource Manager is adapted and optimized to its corresponding piece of hardware. This allows disk access speeds seldom reached by other systems, and in many cases, faster speeds are achieved.

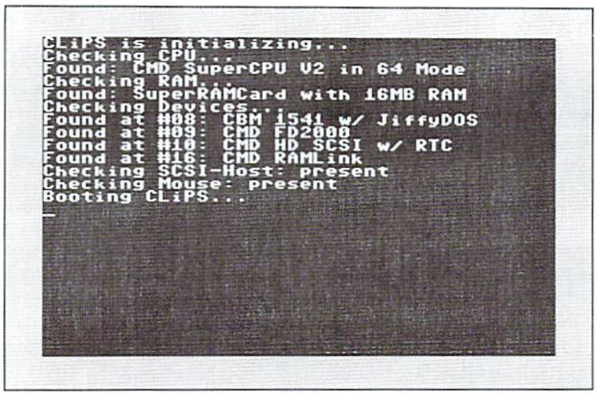
An Open System

CLiPS has been conceived as an open system in order to support every type of disk drive. Pathfinder, the disk drive management program, sends a simple message to the Resource Manager, like "give me a directory listing" (which is returned in standardized format) or "load this file" (which is loaded into the desired memory location at maximum speed). A closer inspection reveals features of CLiPS which might otherwise have been overlooked; the MS-DOS disk-specific Resource Manager is a good example. Simply insert a MS-DOS formatted disk into a FD-2000, click on the corresponding icon in the Pathfinder, and the disk appears as if it's a normal CMD disk. The disk's directory is read and displayed as per usual. Since Windows' long filenames are supported, files can be saved, loaded, and copied without regard to the older "8.3" filename convention. Additionally, all CLiPS applications can directly access PC disks. For example, if you were to save a Paintshop Pro image from the PC to a PC floppy with the intent of editing it with GoDot, you'd simply insert the PC disk into the FD-2000, read the file with GoDot, and away you go!

But that's not all. With the help of this system, CLiPS will be able to support additional types of hardware, namely the IDE64 interface, CD-ROM and ZIP drives connected to CMD SCSI controllers, and even 64Net. Once a corresponding Resource Manager is available, CLiPS will communicate with the piece of hardware sans patches, compromises, or decreases in speed. For RAMLinks, HDs, and FDs, CLiPS routines will be made available which function essentially as replacements for the 1 MHz JiffyDOS. In many cases, a 20 MHz C64 can accept data more quickly than a 1 MHz machine; in the case of a HD/FD with a serial connection, the transfer is accelerated despite the serial bus "bottleneck". RAMLinks also work faster under CLiPS, as do CMD HDs with parallel connections.

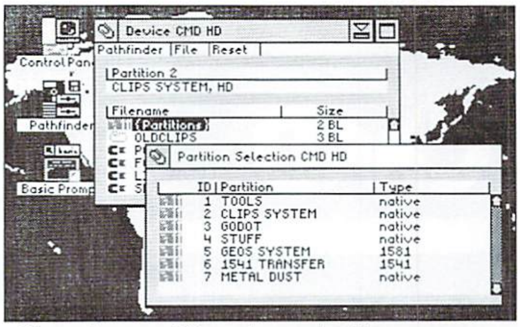
Limits? Not Anymore...

How many disk drives can CLiPS support? Are there limits similar to GEOS' inability to work with more than four drives at a time? You probably already know the answer; CLiPS can use as many drives as are present. For example, no problems were reported from a machine using two CMD-HDs, two FD-2000s, a 1541, a 1571, and a RAMLink simultaneously. Programs can be started from any one of the devices - and why not? Additionally, CLiPS offers a dynamic Ramdisk. Those who own a SuperRAM card with 16 MB of RAM will experience flawless temporary storage of applications and data. In contrast to the Acorn RiscPC, the CLiPS Ramdisk isn't static. That is,the



```
CLiPS is initializing...
Checking CPU...
Found: CMD SuperCPU V2 in 64 Mode
Checking RAM...
Found: SuperRAMCard with 16MB RAM
Checking Devices...
Found at #08: CBM 1541 w/ JiffyDOS
Found at #09: CMD FD2000
Found at #10: CMD HD SCSI w/ RTC
Found at #16: CMD RAMLink
Checking SCSI-Host: present
Checking Mouse: present
Booting CLiPS...
```

CLiPS' automatic hardware recognition



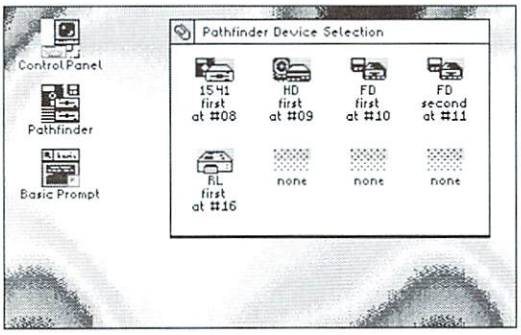
Partition 2
CLIPS SYSTEM, HD

```
Filename                 Size
{Partitions}             2 BL
OLDCLIPS                 3 BL
```

Partition Selection CMD HD

```
ID Partition             Type
 1 TOOLS                 native
 2 CLIPS SYSTEM          native
 3 GODOT                 native
 4 STUFF                 native
 5 GEOS SYSTEM           1581
 6 1541 TRANSFER         1541
 7 METAL DUST            native
```

Changing partitions is easier than ever before



Pathfinder Device Selection

```
1541      HD        FD        FD
first     first     first     second
at #08    at #09    at #10    at #11

RL        none      none      none
first
at #16
```

If you want to use more than four disk drives at a time, it's no problem with CLiPS!

Ramdisk uses as much memory as it needs at a certain time and isn't preset at a specific value (Amiga users should be familiar with this idea). For example, the Ramdisk is of great value to people who want to copy data from one FD disk to another but only own one FD-2000 drive (which is the rule). In this case, the Ramdisk is being used as temporary storage. Owners of Commodore and CMD REUs can still use these devices as ramdisks, which in no way affect the capacity or usability of the SuperRAM. A 512K REU will appear to have exactly that much memory available, as opposed to having the capacity of a 1571 disk.

Disk Structure

Since ramdisks are actually simulated disks, their subdirectories are also simulated. With real disk drives, however, subdirectories can only exist if supported by the EPROM of the drive - primarily the FD, HD, and RAMLink drives. Some programs can force a 1541 to work with subdirectories using special tricks, but CLiPS doesn't deal with this at all. CLiPS does not import any new disk formats which aren't still widely used outside the CLiPS environment nor files in a directory containing only unreadable USR files. CLiPS does not want to repeat what GEOS did, i.e., create a new world separate from the rest of the C64 universe. With CLiPS, for example, you would be able to use a paint program to edit multicolor, hi-res, and IFLI pictures which could then be read by Amiga Paint, Gunpaint, or GoDot. CLiPS also supports CMD HD, RAMLink, and FD partitions with ease. No special partition changing tools are needed; while in Pathfinder, you just click on "Partitions" above the corresponding directory. You then get a list of the drive's partitions, any one of which will give you a listing of its contents when clicked.

How are files loaded?

CLiPS loads files in two different ways - either all at once or as a stream. The first method loads an entire file into memory at high speed. This method is most often used to store frequently used programs and data. For certain instances, however, the second method must be called into action. Say you want to load a file which is larger than the amount of memory available - something that often occurs when working with JPEG images.

In this case, CLiPS does not fall back on the extremely slow byte for byte loading technique which, according to the rules, would make any hardware optimizations (e.g., interleave, track-caching) ineffective. Instead, CLiPS uses a streaming system in which part of an application is loaded into a 64k buffer a few bytes at a time. From then on, every required byte comes from this buffer, and anything that happens in the meantime (e.g., JPEG decoding/calculation) has no effect on the loading. The loading process repeats only when the buffer is completely empty and the next byte is required. It is possible to have more than one file open at a time in this mode, and if the computer does not have very much ram, data can be copied right to the hard drive.

Data Transfer over the Internet

In recent years, a disk image format know as "D64" has become standard. A D64 disk image is little more than a large file containing a sector by sector copy of a 1541 disk. Originally developed as a file format for emulators, many C64 users send their data over the Internet using the D64 format; impractical tools such as ZipCode and Lynx are used less and less as time goes on. But those who don't use the program Star Commander with a 1541 drise connected to their PC have to be patient; the tools needed to write a D64 file back to a disk are either slow or only work with DD disks or a HD. There are also reports of frequent errors with these programs. CLiPS does away with this mess too. The designers plan to implement a system whereby a D64 file is seen as a virtual disk drive, and users will be able to access information on the "drive" accordingly. Of course, the virtual drive will be copyable and open to editing. Special CLiPS routines will also finally allow D64 disk images to be written back to a 1541 disk in a reasonable amount of time.

Pathfinder

The Pathfinder is what is known as a file manager in other operating systems. In its expanded version, the Pathfinder functions as a file browser inside running programs. It offers directory management in scaleable windows with proportional scrollbars in which directories can be displayed, files deleted, and "dragged and dropped" from one drive to another. With this system, CLiPS offers the "comfort of a luxury sports car", that is, power under the hood and a pleasant ride at speeds never experienced before.

(gb)

A reliable PAL/NTSC check!

CMD SuperCPU Corner

"Where am I from?" - this quite philosophic question isn't uninteresting for C64 programmers. Even those without a SuperCPU may learn something here.

by Wolfram Sang

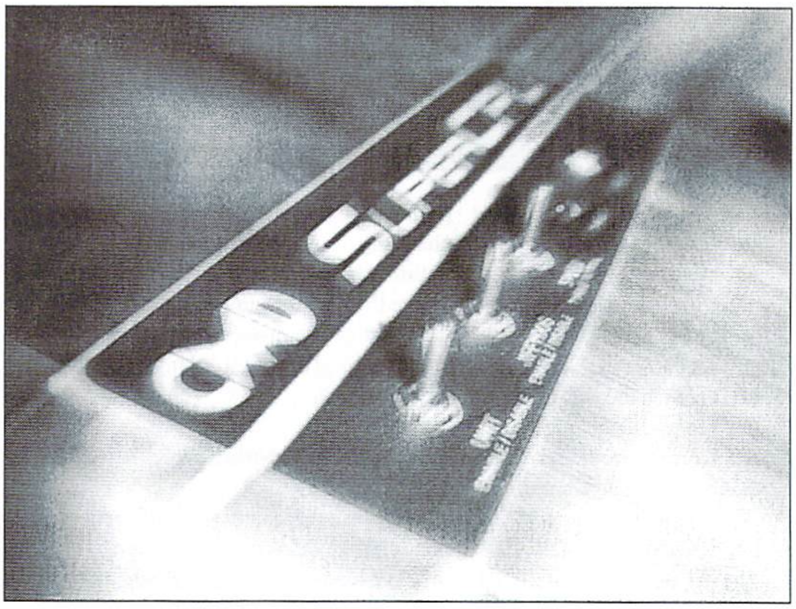
Our beloved Commodore exists in PAL and NTSC versions, which means they have differences in their clock frequency and picture generation (see box). A program which relies on this must know which kind of system it is currently running on. The kernal offers the memory location $02a6 for that purpose (0 = NTSC, 1 = PAL), which is accepted as reliable enough by many programmers. Unfortunately, this method has two major disadvantages: First it can lead to wrong results, since this location is only set in the Reset routine ($ff5b - $ff6a). Afterwards the value can be modified (by mistake or intentionally), for example with a simple POKE command. Second, this value will always indicate an NTSC machine when a SuperCPU is running and has been in 20 MHz mode during Reset. To explain that, let's take a further look at the system detection method of the kernal routine.

A good idea...

As you can see in the extra box, a PAL-C64 generates more rasterlines than an NTSC one, exactly 312 instead of 263. To check this, just write a number of a rasterline only existent on PAL into the latch $d011/12 (for example $137 = dec. 311). In the Interrupt Request Register (IRR) at $d019 bit 0 we can see whether the line we've indicated has been reached. We just wait until one entire screen has been refreshed once and then look if the rasterline $137 was displayed. Remember: This is only possible on PAL systems! So we have an absolutely sure decision criterium.

... badly realized

Why does this method now fail with a SuperCPU? Well, this is the fault of Commodore. The offending kernal routine doesn't explicitly wait until a full screen has been refreshed. Instead, it performs some other tasks in the meantime (clear screen etc.). On a standard C64 it is absolutely sure that these tasks take more time than one screen refresh. Unfortunately, the developers at Commodore couldn't know that a 20 MHz turbo board, introduced 15 years later, would be too fast to give the VIC time to reach rasterline $137. The following routine solves the problem. It delivers a 100% clear identification of the system, even with a SuperCPU in turbo mode. Moreover, only 6510 opcodes are used, so that this routine will also work on an unexpanded C64.



The improved version

Let's look at our new routine step by step. First we disable the interrupts and set the NMI vector to an RTI - we need silence for the detection of the video mode, any interrupt could lead to a wrong result. Now we wait until $d012 equals zero, meaning that we are either in rasterline 0 or 256. Which one doesn't matter, this check only prevents the routine from being called when the VIC is uncomfortably close to rasterline $137, which could lead to an error under certain circumstances.

Now we write the number of our test-line into the latch register, and then we reset bit 0 in the IRR (which is done by writing anything to it, in this case a 1) to clear out any pending interrupts. With the following waiting loop we make sure that an entire screen has been refreshed. Since we check the rasterline number itself for this, this routine will probably still work in another 15 years with the 200

PAL/NTSC-specifications compared

System	VIC Part No	Cycles per raster line	raster lines per field	field refresh rate	clock rate
NTSC	6567	65	263	~60Hz	1022727 Hz
PAL	6569	63	312	~50Hz	985248 Hz

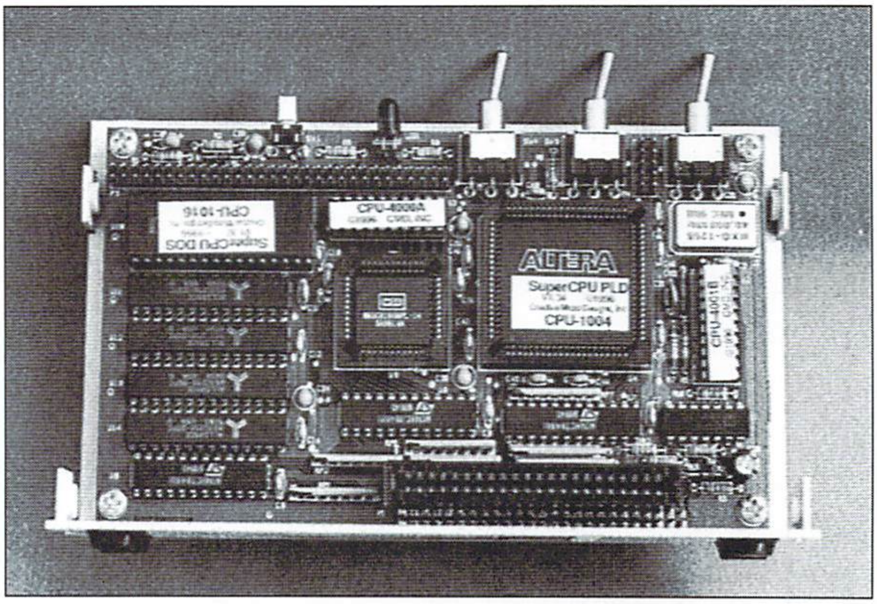
20 MHz power for your Commodore

MHz ultra turbocard. Now the only thing we need to do is mask out the crucial bit in the IRR, reset $d019 and leave the routine. The value in the Accumulator now represents PAL or NTSC, just like in $02a6, but much more reliable.

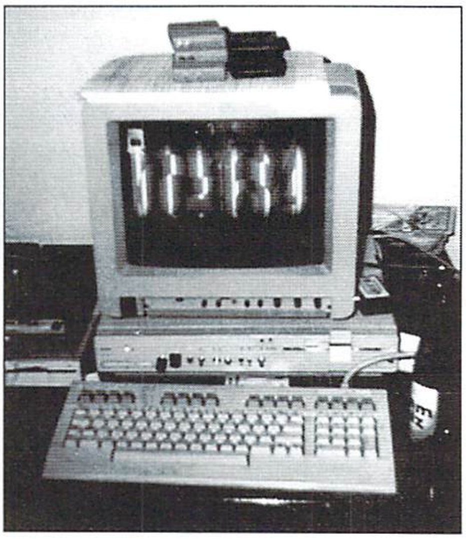
Finished!

As you can see, this check is neither long nor complicated. I urgently recommend the use of this one instead of $02a6, because who wants an orange to be identified as an apple?

```
nmivec = $031 ; NMI-vector

 org $0801

 adr $080b, 64
 byt $9e,"2061",0,0,0 ; Basic-
                         line

jmp_in:
 lda #lo(text)
 ldy #hi(text)
 jsr $ab1e   ; print startup-
               message

 jsr palntsc ; perform check
 sta $02a6 ; update KERNAL-
             variable
 beq ntsc ; if A=0, then go to
            NTSC

 lda #lo(pal_text)
 ldy #hi(pal_text) ; otherwise
                     print PAL-
                     text
 jmp $ab1e   ; and go back.
ntsc:
 lda #lo(ntsc_text)
 ldy #hi(ntsc_text) ; print
                      NTSC-text
 jmp $ab1e ; and go back.

palntsc:
 sei ; disable interrupts
 ldx nmivec
 ldy nmivec+1 ; remember old
 NMI-vector
 lda #lo(rti)
 sta nmivec
 lda #hi(rti) ; let NMI-vector
                point to
 sta nmivec+1 ; a RTI
wait:
 lda $d012
 bne wait ; wait for rasterline
            0 or 256
 lda #$37
 sta $d012
 lda #$9b  ; write testline $137
             to the
 sta $d011 ; latch-register
 lda #$01
 sta $d019 ; clear IMR-Bit 0
wait1:
 lda $d011 ; Is rasterbeam in
             the area
 bpl wait1 ; 0-255? if yes, wait
wait2:
 lda $d011 ; Is rasterbeam in
             the area
 bmi wait2 ; 256 to end? if yes,
             wait
 lda $d019 ; read IMR
 and #$01  ; mask Bit 0
 sta $d019 ; clear IMR-Bit 0
 stx nmivec
 sty nmivec+1 ; restore old NMI-
                vector

 cli  ; enable interrupts
 rts  ; return

rti:
 rti  ; go immediately back
        after
      ; a NMI

text:
 byt $93,$05,$0e,$0d
 byt "Reliable PAL/NTSC-
 Detector",$0d
 byt "by Ninja/The Dreams in
 2000",$0d
 byt "for GO64!/CW-
 Magazine.",$0d,$0d
 byt $9b,"You have a ",0
pal_text:
 byt "PAL-machine.",$0d,$05,0
ntsc_text:
 byt "NTSC-machine.",$0d,$05,0
```



UNCOVERED: THE VDC 8568'S 38TH REGISTER



Ok you,re probably already thinking, "but the VDC has only 37 registers, hasn't it?" This is quite correct for the old 8563 types inside the C128 and C128D. Inside the C128DCR, however, a later version with different new characteristics had been installed. One of them is an additional register!

by Nicolas Welte

The existance of register 37 (you start counting at 0) had already been reported about by 64'er magazine, but there hasn't even been a hint of an explanation of the register's function. This should finally be made up for with this article.

Only bits 7 and 6 are available, all other bits are 'reserved' and will always read back as 1. Following a hardware reset, though, bits 7 and 6 are also 1, so that reading the register will return 255 just as with all other unused registers. Only when writing into the register, the difference will become apparent, and in a really big way to boot: the screen loses synchronization and the picture runs through. This is not a useful effect, of course.

Some measuring at video output returned the following function: bit 7 controls the horizontal sync signal's polarity. That signal is found at the RGBI connector. In the same way, the vertical sync signal at pin 9 is controlled by bit 6. In this case, polarity refers to whether the synchronisation pulse is directed from TTL 0 to 1 (positive) or from 1 to 0 (negative). By default, the C128 works with positive pulses for both signals, and all monitors are designed for that. This leads to following register map, which you should glue into the manual:

Register	Bits	Function
37	7.......	HSync 0: negative polarity 1: positive polarity
	.6......	VSync 0: negative polarity 1: positive polarity

Is there any advantage we can get from this register? Yes, indeed there is! But in order to gain it, you need an EGA monitor. EGA monitors were the first high resolution monitors for PCs that really deserved the name. These monitors work with two different fixed frequency modes, one of which is comaptible with older CGA cards, and therefore with the C128's RGBI output. The other mode works at a line frequency of about 21kHz and a resolution of 640x350 at a vertical refresh rate of 60Hz, though. EGA also offers 64 colors instead of just 16 colors, but of course this feature can't be used by a C128. Some C128 owners may be using such a monitor, as they offer a better display quality than older CGA models. These days, they can be bought for a song as well.

How does the EGA monitor know in which of its two modes it should work? One option would be an automatic detection of the line frequency, but the time hadn't come for that technology yet. Thus, the monitor needed a signal from the outside indicating which mode should be chosen. This is realized by the sync signals' polarity! Four different modes could therefore be chosen, but my monitor in particular knows only two. There are also modern multi frequency monitors which ignore this signaling and switch automatically.

There are two example programs on the cover disk, which will only run on a C128DCR. Those of you who are sure you have a monitor which switches automatically, may remove line 5 from the program, which will allow the programs to run on old C128s and C128Ds. In all cases, you should be absolutely sure that a monitor is connected which can handle 21kHz, otherwise, there can be significant damage! Two different versions are given, a 50Hz version and a 60Hz version, sensibly named EGA50HZ and EGA60HZ. The 60Hz version displays a text screen with 77 columns and 43 lines roughly resembling PC's EGA standard. At least for me, the 50Hz version works better, perhaps due to the fact that I have a 50Hz mains current as well. This allows for 77 columns and 52 lines, without any interlace flickering! Unfortunately, 77 columns are the absolute maximum with attributes turned off.

[Perhaps he means on machines with only 16K of video RAM. On the C128-DCR, 64K is standard, and screen resolutions of 80x60 and higher are possible in full color. --Nate]

If you turn them on, only 70 columns are possible. In addition, accessing the video memory becomes tediously slow, so that you should turn off even more columns. These programs are only meant to demonstrate the additional register's workings anyway, and I don't believe that the new screen modes will ever really be useful. Please use the reset button in order to end the new screen modes. The C128's firmware doesn't know this register exists, and therefore doesn't try to reset it.

One small curiosity at the end: The German edition of the C128's programmers manual, simply called "Das C128 Buch" ("The C128 Book"), contains a register table for the 8563 (!) in chapter 10, which mentions register 37 and describes its workings correctly, if you can ignore a minor misprint. No version of the 8563 actually contained this register, though, and the 8568 became available only long time after this book had originally been printed. Obviously, Commodore had planned that register for a long time, but made it a reality only much later.

I wish all EGA monitor owners a good time experimenting.

Yours, Nicolas Welte
welte@chemie.uni-konstanz.de

The ROM-Listing

Part 3

There are commented ROM listings of the operating system of the 1541 disk drive in various books. For the 1581 disk drive, such a thing has never been published. Since a book about a C64 disk drive would probably only be sold in small numbers today, we bring you exclusively a fully-commented ROM Listing of the 1581 disk drive in GO64!. Because itis very large, we are spreading it over several issues with one page per issue -- this listing certainly not being interesting for everyone. To help collect these pages, we have specifically provided a margin for hole-punching on the left side.

```
,80a2 20 99 80 jsr $8099    Look for ':' in command line
,80a5 d0 05    bne $80ac    found, ==>
,80a7 a9 34    lda #$34     34, Syntax Error
,80a9 4c 7c 80 jmp $807c    (':' missing)

,80ac 88       dey
,80ad 88       dey          Memorize Pointer to drive number
,80ae 8c 91 02 sty $0291    (if existant)
,80b1 8a       txa          Number of commas found > 0 ?
,80b2 d0 f3    bne 80a7     Yes, error: ',' before ':' ==>

Analyze command line between ':' and '='

,80b4 a9 3d    lda #'=
,80b6 20 1c 81 jsr $811c    Look for '='
,80b9 8a       txa
,80ba f0 02    beq $80be    Is there a ',' before the '=' ?
,80bc a9 40    lda #$40     Yes, set ', before =' flag
,80be 09 21    ora #$21     b0-1 = 01: set 'no =' flag
,80c0 8d a2 02 sta $02a2    Memorize bit mask
,80c3 e8       inx          Number of commas found +1
,80c4 8e 2e 02 stx $022e     = number of parameters
,80c7 8e 2f 02 stx $022f    (e. g. 'xxx,p,r')
,80ca ad a1 02 lda $02a1    Is there any joker?
,80cd f0 0d    beq $80dc    No, ==>
,80cf a9 80    lda #$80     Set flag 'joker before ='
,80d1 0d a2 02 ora $02a2    Enter into bit mask
,80d4 8d a2 02 sta $02a2
,80d7 a9 00    lda #$00     Clear joker flag
,80d9 8d a1 02 sta $02a1
,80dc 98       tya          Has an '=' been found?
,80dd f0 29    beq $8108    No, ==>
,80df 9d 91 02 sta $0291,x  Memorize position of '='
,80e2 ad 2e 02 lda $022e    Memorize number of parameters before '='
,80e5 8d 30 02 sta $0230

Analyze command line from '=' to end of line

,80e8 a9 8d    lda #$8d     $8d=<Shift Return>
,80ea 20 1c 81 jsr $811c    Examine from '=' until end of line
,80ed e8       inx          Memorize number of parameters
,80ee 8e 2f 02 stx $022f
,80f1 ca       dex          = number of ','
,80f2 ad a1 02 lda $02a1    Joker existant?
,80f5 f0 02    beq $80f9    No, ==>
,80f7 a9 08    lda #$08     Set flag 'Joker after ='
,80f9 ec 2e 02 cpx $022e    Number of commas = number of commas before '='?
,80fc f0 02    beq $8100    Yes, (no commas found) ==>
,80fe 09 04    ora #$04     Set flag ', after ='
,8100 09 03    ora #$03     b0-1 = 10: set flag '='
,8102 4d a2 02 eor $02a2    If appropriate, set b7-8
,8105 8d a2 02 sta $02a2    Turn b0-1

,8108 ad a2 02 lda $02a2    Combine bit mask with control mask
,810b ae 2a 02 ldx $022a    (Nunber of current command)
,810e 3d 94 db and $db94,x  at $db9c
,8111 d0 01    bne $8114    Needs to become 0, otherwise error ==>
,8113 60       rts          Format OK

,8114 8d ab 02 sta $02ab    Set error flag
,8117 a9 30    lda #$30      30, Syntax Error
,8119 4c 7c 80 jmp $807c
----------------------------------
Analyze input line up to a certain character ($c268)
The number of ',' up to the location found is counted, and wild cards are noted.
x should always be set to 0 prior to the first call.

I: a: end character
   x: old nunber of commas found
   y: starting position

O: a: position of character + 1 (0 = character not found)
   x: number of commas found
   y: (=a)
   n: 1: character not found
----------------------------------------

,811c  8d  8f  02  sta $028f      character to be found
,811f  c4  29      cpy  #$29      reached end of command line?
,8121  b0  2e      bcs $8151      yes, ==>
,8123  b1  cd      lda ($cd),y    get character from command line
,8125  c8          iny
,8126  cd  8f  02  cmp $028f      = character to be found?
,8129  f0  28      beq $8153      yes, ==>
,812b  c9  2a      cmp #'*        wild card ?
,812d  f0  04      beq $8133      yes, ==>
,812f  c9  3f      cmp #'?
,8131  d0  03      bne $8136      no, ==>
,8133  ee  a1  02  inc $02a1      set wild card flag
,8136  c9  2c      cmp #',
,8138  d0  e5      bne $811f      ',' found
,813a  98          tya            yes, store position of next
,813b  9d  92  02  sta $0292,x    parameter
,813e  ad  a1  02  lda $02a1      did the parameter include wild cards?
,8141  29  7f      and #$7f
,8143  f0  07      beq $814c
,8145  a9  80      lda #$80       yes, set wild card flag
,8147  95  f4      sta $f4,x
,8149  8d  a1  02  sta $02a1      clear wild card flag for new search
,814c  e8          inx            b7=1: wild cards have been found
,814d  e0  04      cpx #$04       have too many parameters been given ?
,814f  90  ce      bcc 811f       no, ==>
,8151  a0  00      ldy #$00       yes, end of search
,8153  a5  29      lda $29        remember length of line, or position of
,8155  9d  92  02  sta $0292,x    character searched
,8158  ad  a1  02  lda $02a1      did the last parameter include a wild card (b7=1) ?
,815b  29  7f      and #$7f
,815d  f0  04      beq $8163      no, ==>
,815f  a9  80      lda #$80       set wild card flag
,8161  95  f4      sta $f4,x
,8163  98          tya
,8164  60          rts

------------------------------------------
Determine end of command line (remove CR (/LF)) ($c2b3)
(if u0 then don't change end of line)
clear command line parameters
------------------------------------------

,8165  a4  cd      ldy $cd        length of command line
,8167  f0  36      beq $819f      = 0 ? ==>
,8169  88          dey
,816a  f0  32      beq $819e      = 1 ? ==> (can't be u0)
,816c  ad  00  02  lda $0200      (slightly complicated) check for 'u0'
,816f  c9  55      cmp #'u
,8171  d0  07      bne $817a      (u0)
,8173  ad  01  02  lda $0201
,8176  c9  30      cmp #'0
,8178  f0  04      beq $817e      = u0 ? ==>
,817a  b9  00  02  lda $0200,y    compare end of line to CR
,817d  2c           b $2c         (u0)
,817e  a9  00      lda #$00       u0: ignore CR
,8180  c9  0d      cmp #$0d
,8182  f0  1b      beq $819f
,8184  88          dey
,8185  ad  00  02  lda $0200      (check for u0) (why ?)
,8188  c9  55      cmp #'u
,818a  d0  07      bne $8193
,818c  ad  01  02  lda $0201
,818f  c9  30      cmp #'0
,8191  f0  04      beq $8197      = u0 ? ==>
,8193  b9  00  02  lda $0200,y    compare last but cne character to CR
,8196  2c           b $2c
,8197  a9  00      lda #$00       u0: ignore CR
,8199  c9  0d      cmp #0d
,819b  f0  02      beq $819f
,819d  c8          iny            <= either no CR or u0
,819e  c8          iny            <= CR exists / length = 1
,819f  84  29      sty $29        <= CR, LF exist / length = 0
,81a1  c0  2a      cpy #$2a       line too long ?
,81a3  a0  ff      ldy #$ff
,81a5  90  08      bcc $81af      no, ==>
,81a7  8c  2a  02  sty $022a
,81aa  a9  32      lda #$32       32, SYNTAX ERROR
,81ac  4c  7c  80  jmp $807c

Initialize command line parameters ($c2dc)

,81af  a0  00      ldy #$00
,81b1  98          tya
,81b2  85  cd      sta $cd        pointer to INPUT buffer lo = 0
,81b4  8d  6f  02  sta $026f      current record length = 0
,81b7  8d  2d  02  sta $022d      current file type = DEL
,81ba  8d  33  02  sta $0233      clear file type for file search
,81bd  85  67      sta $67        clear counter (for instance for block commands)
,81bf  8d  30  02  sta $0230      number of parameters before '=' or number of file
                                  names
,81c2  8d  2e  02  sta $022e      clear parameter counter (before '=')
,81c5  8d  2f  02  sta $022f      clear number of parameters in command line
,81c8  8d  a1  02  sta $02a1      clear wild card flag
,81cb  8d  ab  02  sta $02ab      clear error flag
,81ce  a2  05      ldx #$05
,81d0  9d  90  02  sta $0290,x    parameter positions
,81d3  95  e4      sta $e4,x      directory block with file entry
,81d5  95  e9      sta $e9,x      pointer to entry
,81d7  95  ee      sta $ee,x      drive numbers
,81d9  95  f3      sta $f3,x      clear wild card flags
,81db  9d  96  02  sta $0296,x    T&S of files = 0, or
,81de  9d  9b  02  sta $029b,x    block commands' parameters = 0
,81e1  ca          dex
,81e2  d0  ec      bne $81d0
,81e4  60          rts

---------------------------------
LED routines ($c100)
---------------------------------

,81e5  a9  00      lda #$00       turn of LEDs
```

```
,81e7  8d  ab  02  sta $02ab      clear error flag
,81ea  a5  79      lda $79        turn off drive LED
,81ec  29  df      and #$df
,81ee  85  79      sta $79
,81f0  60          rts
----------------------------------
,81f1  a9  50      lda #$50       LED blinking
,81f3  8d  ab  02  sta $02ab
,81f6  a5  79      lda $79        turn on drive LED
,81f8  09  40      ora #$40
,81fa  85  79      sta $79
,81fc  60          rts
----------------------------------
Fetch and set drive number ($c312)
(for commands with a file name and possible other parameters
for instance OPEN 1,dn,2,"0:xxxx,p,r", "n0:xxxx,yy", ...)
----------------------------------

,81fd  ad  2f  02  lda $022f      temporarily store
,8200  8d  2e  02  sta $022e      number of parameters
,8203  a9  01      lda #$01       only the first parameter is a file name
,8205  8d  2f  02  sta $022f      and should be checked
,8208  8d  30  02  sta $0230      1 file name in command line

----------------------------------
Check all parameters for drive numbers, and erase them if found ($c320)
(for instance "c0:xxxx=0:yyy,0:zzz", "s0:xxx,0:yyy", LOAD"$0:xxx,0:yyy",dn ...)
----------------------------------

,820b  a2  00      ldx #$00       start with first parameter
,820d  86  67      stx $67
,820f  bd  91  02  lda $0291,x    get parameter's position
,8212  20  24  82  jsr $8224      and check
,8215  a6  67      ldx $67        write back new position
,8217  9d  91  02  sta $0291,x    (in order to remove drive number)
,821a  98          tya            remember drive number
,821b  95  ef      sta $ef,x
,821d  e8          inx            check next parameter
,821e  ec  2f  02  cpx $022f      < number of parameters ?
,8221  90  ea      bcc $820d      yes, ==>
,8223  60          rts

----------------------------------
Test for drive number and remove it ($c33c)
----------------------------------

,8224  aa          tax            a: position, where a drive number is
,8225  a0  00      ldy #$00       being expected. y=0 (default drive number)
,8227  a9  3a      lda #':        determine the kind of drive specification
,8229  dd  01  02  cmp $0201,x    the position of ':' (for instance ':' or '0:')
,822c  f0  0c      beq $823a      drive number exists, ('0:') ==>
,822e  dd  00  02  cmp $0200,x    only ':', default drive
,8231  d0  17      bne $824a      no ':', ==>
,8233  e8          inx            remove drive number (':')
,8234  98          tya            not necessary,
,8235  29  01      and #$01       is 0 anyway
,8237  a8          tay
,8238  8a          txa            x points after the drive number now
,8239  60          rts

,823a  bd  00  02  lda $0200,x    read drive number
,823d  e8          inx            remove drive number ('0:')
,823e  e8          inx
,823f  c9  30      cmp #'0
,8241  f0  f2      beq $8235      drive 0, then ok ==>
,8243  c9  31      cmp #'1        drive 1, then '31, SYNTAX ERROR'
,8245  d0  ed      bne $8234      neither 0 nor 1, then drive 0
,8247  4c  2d  80  jmp $802d      31, SYNTAX ERROR

,824a  98          tya
,824b  09  80      ora #$80       flag for  'no drive number' (a = $80)
,824d  29  81      and #$81
,824f  d0  e6      bne $8237      always ==>
----------------------------------
Set drive number, turn on LED ($c368)
----------------------------------

,8251  a9  00      lda #$00       clear command syntax flag
,8253  8d  a2  02  sta $02a2
,8256  ac  91  02  ldy $0291      fetch the drive number's position
,8259  b1  cd      lda ($cd),y    fetch drive number
,825b  20  95  82  jsr $8295      compare to '0' and '1'
,825e  10  0d      bpl $826d      drive number ok, ==>
,8260  c8          iny            reached end of line?
,8261  c4  29      cpy $29
,8263  b0  05      bcs $826a      yes, ==>
,8265  a4  29      lda $29        look for drive number at end of line
,8267  88          dey            (for instance 'I0', 'V0', '$0', ...)
,8268  d0  ef      bne $8259      length of command <> 1, ==>
,826a  ce  a2  02  dec $02a2      = $ff
,826d  4c  f6  81  jmp $81f6      turn on drive LED

----------------------------------
Determine file type ('s,p,u,r,c') (for instance '$*=s') ($c398)
----------------------------------

,8270  a0  00      ldy #$00
,8272  ad  2e  02  lda $022e      number of parameters before '='
,8275  cd  2f  02  cmp $022f      = number of parameters in command line ?
,8278  f0  16      beq $8290      yes, no type given, ==>
,827a  ce  2f  02  dec $022f      remove pointer to end of line
,827d  ac  2f  02  ldy $022f      fetch last parameter (soon to be end of line)
,8280  b9  91  02  lda $0291,y
,8283  a8          tay            fetch parameter's first character
,8284  b1  cd      lda ($cd),y
,8286  a0  05      ldy #$05       compare to all
,8288  d9  ab  db  cmp $dbab,y    file type denominators
,828b  f0  03      beq $8290      equals, ==>
,828d  88          dey            if type not found, or
,828e  d0  f8      bne $8288      type = del:
,8290  98          tya            store 'type not found or not given'
```

```
,8291  8d  33  02  sta $0233      store file type
,8294  60          rts

----------------------------------
Check drive number ($c3bd)
----------------------------------

,8295  c9  30      cmp #'0
,8297  f0  06      beq $829f
,8299  c9  31      cmp #'1
,829b  f0  02      beq $829f
,829d  09  80      ora #$80       no or incorrect drive number
,829f  29  81      and #$81       return 0, 1, or $80
,82a1  60          rts

----------------------------------
Initialize drive, turn on LED ($c3ca)
----------------------------------

,82a2  a9  00      lda #$00
,82a4  85  40      sta $40        (no obvious usage)
,82a6  8d  a4  02  sta $02a4      (drive number ?)

,82a9  8d  a3  02  sta $02a3      number of drives = 0 (= 1 drive)
,82ac  20  ae  84  jsr $84ae      initialize partition if needed
,82af  f0  05      beq $82b6      ok, ==>
,82b1  a9  74      lda #$74       74, DRIVE NOT READY
,82b3  20  7c  80  jsr $807c      end
,82b6  4c  f6  81  jmp $81f6      set drive LED flag
----------------------------------
Search all provided files in the directory ($c44f)
O: file tables $e5- $f8
   file start  $0297- $02a1
   $02a5: $ff: all finds found (no name contains wild cards)
          $00: - not all files found, or
               - files contain wild cards (further matches possible)
----------------------------------

,82b9  20  a2  82  jsr $82a2      initialize partition
,82bc  a9  00      lda #$00       set flag for 'search for entries in usage'
,82be  85  73      sta $73
,82c0  20  24  84  jsr $8424      fetch 1st directory entry
,82c3  d0  0b      bne $82d0      match ? yes, ==>
,82c5  ce  a3  02  dec $02a3      number of drives -1 = $ff
,82c8  10  f2      bpl $82bc      (is never executed)
,82ca  60          rts

,82cb  20  89  84  jsr $8489      fetch next directory entry
,82ce  f0  10      beq $82e0      no more available ? yes, ==>
,82d0  20  27  83  jsr $8327      compare entry to desired entries
,82d3  ad  a5  02  lda $02a5      not all files found yet ?
,82d6  f0  01      beq $82d9      yes, ==>
,82d8  60          rts
,82d9  ad  6d  02  lda $026d      was the entry among those desired ?
,82dc  30  ed      bmi $82cb      no, ==>
,82de  10  f0      bpl $82d0      yes, ==>

,82e0  ad  a5  02  lda $02a5      all files found ?
,82e3  f0  e0      beq $82c5      no, (end) ==>
,82e5  60          rts

----------------------------------
Look for files one by one ($c48b)
O: n: 1: no more files found; 0: search was successful
   file tables $e5- $f8
   file start $0297- $02a1
----------------------------------

look for next file (reread directory block beforehand)

,82e6  20  77  84  jsr $8477      fetch next directory entry
,82e9  f0  0e      beq $82f9      another file found ? no, ==>
,82eb  d0  1c      bne $8309      yes, ==>

1. look for file

,82ed  a9  00      lda #$00       set flag for 'search for entries in usage'
,82ef  85  73      sta $73
,82f1  20  24  84  jsr $8424      search for 1st directory entry
,82f4  d0  13      bne $8309      found ? yes, ==>
,82f6  8d  a5  02  sta $02a5      a=0: some files still not found
                                  (set n=1 and end)
,82f9  ad  a5  02  lda $02a5      all files found ?
,82fc  d0  28      bne $8326      yes, (definite end of search) ==>
,82fe  ce  a3  02  dec $02a3      number of drives -1 = $ff
,8301  10  ea      bpl $82ed      (is never executed)
,8303  60          rts            n=1, end

look for next file (directory block is still in buffer)

,8304  20  89  84  jsr $8489      <-- fetch next directory entry
,8307  f0  f0      beq $82f9      none left ? yes, ==>
,8309  20  27  83  jsr $8327      compare entry to desired entries
,830c  ae  6d  02  ldx $026d      was it a match ?
,830f  10  07      bpl $8318      yes, ==>
,8311  ad  a5  02  lda $02a5      are the file names not yet found ?
,8314  f0  ee      beq $8304      yes, ==>
,8316  d0  0e      bne $8326      no, n=1, end ==>
,8318  ad  33  02  lda $0233      is a specific file type required ?
,831b  f0  09      beq $8326      no, (0 would be DEL) ==>
,831d  b5  f4      lda $f4,x      compare file type
,831f  29  07      and #$07
,8321  cd  33  02  cmp $0233
,8324  d0  de      bne $8304      not equal, ==>
,8326  60          rts            file found (n=0), end ==>

----------------------------------
Compare entry in directory to desired entries ($c4d8)
----------------------------------

,8327  a2  ff      ldx #$ff
```

Efficient Searching in Texts and Lists

In the early years of the computer age, computers were slow compared to today's machines, but it was exactly this necessity that knows no law.

by Wanja Gayk

This is why probably the most efficient algorithms for tackling standard problems, like exact search in texts, sorting data and so on, were developed especially in the 70s. Algorithms you should teach some of today's programmers by force, if you have a look at how slow some programs are even on a 300 MHz PC.

The Problem

Let's imagine we had a relatively long text or some other accumulation of data, like for instance a picture or a list of addresses. Let's continue to imagine that we were looking for the place where a specific word, pattern, or name occurs in our data. What we need is therefore an appropriate search algorithm that not only does as told, but is also fast. Because, who likes programs that need ages to produce a result?

The Naive Solution

Those searching in a text will figure out the naive solution first, which looks like this: *Look for the first character of pattern (P) in text (T). If they natch, compare the following characters. If all characters match, the pattern has been found. If it has been found, look for its next occurance. Start at one place later in the text than with the attempt before. If there is a mismatch, restart the search as well one place later in the text.* This approach is quite simple and seems to be quick fast as well. Having a closer look, you would realize, though, that many patterns would have no problem with starting not one place later in the text but increasing the step width.

An example:

A text T and a pattern P are given. The naive method would handle the comparison as follows ("*" marks cancel, "!" success):

```
T: ABRAHAM SAYS ABRACADABRA
P: ABRAC*
    A*
     A*
      AB*
       A*
        ........
                ABRACADABRA!
```

I have saved myself the trouble of writing down the complete search, but you can see that the pattern has to be "shifted" 14 times altogether in comparison to the text, until the comparison leads to a success. There is a better solution, of course. The following search algorithm stems from the year 1977, and has been named after its inventor. The ...

Boyer- Moore- Horspool- Algorithm

The BMH algorithm is mainly based, on you being able to move the pattern (in relation to the text) by more than just one character. But this makes it necessary for you to preprocess the pattern, that is: get to know the pattern better, instead of just knowing which letter are contained. Let's start by having a look at how the algorithm works:

```
T: ABRAHAM SAYS ABRACADABRA
P: ACRACADABRA
            *A
                       *A
               !ABRACADABRA
```

The goal has been reached in three steps!! But what causes this strange behaviour? The first thing you will notice is that the pattern is compared with the text from right to left. Then you will notice that the steps are very large, with which the pattern is moved over the text. In order to understand this, let's have a closer look at the pattern:

```
P: ABRACADABRA
LO:    C DABR
```

I use LO to denote the Last Occurance of a character in the pattern or the first time, you meet a letter, when looking through the pattern from the right to the left. The last character in the pattern **must** be ingnored.

But what's this all good for? Well, I compare from the right to the left and right during the first comparison (see example), I find a Y in the text, where the pattern contains an A. Since the pattern doesn't have a Y, I can move by the pattern's entire length right away, without risking missing a match. During the next attempt, I find a B in the text where the pattern has an A. Now I take a look where it would be best to move to, without missing a possible match. In order to do so, I check which is the letter in the text being covered by the last letter in the pattern (that is the letter compared first). In our case it is the aforementioned B. The distance between the **last** B in the pattern and this position is noted in my LO table. As we can see in the LO table, the B is the second but last letter in the pattern. I can therefore move the pattern by exactly 2 positions so that the pattern's B comes to rest exactly under the text's B. On the third attempt, all letters match, and we report a success.

The algorithm can be programmed relatively easily, but it results in a rapid increase of speed with increasing pattern length. The following table illustrates this quite clearly:

Pattern	Naive-matcher.bas	BMH-Matcher.bas (Search+ Prepro-cessing)
ab	129	260 (246+14)
abra	127	159 (143+16)
test	125	163 (147+16)
table	125	137 (121+16)
abraham	124	114 (96+16)
strings	124	99 (81+18)
decides	124	109 (91+18)
abracadabra	121	89 (69+20)
abracadabrx	121	74 (53+21)
forget about the table	113	71 (42+29)
forget about the abracadabra	110	63 (29+34)

!!Note: The smaller the value, the smaller the time needed.

Starting at a pattern length of 5 (see pattern "table"), the BMH- algorithm becomes faster than the naive approach as you can see. The pattern's preprocessing still eats up that advantage. With increasing text length, the BMH algorithm would catch up with the naive approach, though.

The BMH- Algorithm is slowed down by

C64 BASIC especially by the fact that the Commodore BASIC V2 doesn't allow for indexing arrays by characters, as offered by Turbo Pascal, for instance. Thus you have to convert ASCII characters to integer values very often and in a complicated manner during the program loops. In assembly language, things look different, though, and the BMH algorithm can really show its strength.

There are even better search algorithms, of course, like the Knuth- Morris-Pratt-algorithm, but the BMH solution excels by its extraordinary simplicity, so that I found it most appropriate for this article. The programs I used for the timing are on the cover disk and can be started with RUN.

Optimizing for Lists

Those very often searching in lists like addresses, is well advised to optimize them in a way to allow the search algorithm to find the desired object especially fast. One example would be an address directory: You can organize an address directory in such a way to have it sorted alphabetically by name. If you very often looked for a name starting with S or T, though, it could take quite some time, until you found it with such a search algorithm, since these entries would be quite close to the end of the list. In this case it would probably better to put the entries you look for very often closer to the beginning of the list then would be the case with an alphabetical order.

You don't necessarily know if you are going to access a given entry often or not when establishing a list. Therefore, it seems advisable to change the list dynamically, so that an entry often searched automatically moves to the front so that it can be found faster in the future.

The so- called MTF procedure (Move To Front) has proven to be especially efficient and easy. It's mechanism is simple: If an entry has been searched and found, it is moved to the beginning of the list. This method has one disadvantage, though: An entry could also be moved to the front of the list, although you don't really look for it regularly. This "error" is corrected only slowly in the MTF scheme, but this shouldn't be too much a problem, since entries often looked for still remain on average more towards the beginning of the list than those seldom wanted.

You may wonder now, if this really made sense. After all, you would have to reorganize the list every time your search succeeds, which can take some time, especially on the C64.

The answer is "yes". For one thing, moving data is usually faster than searching for them. And secondly, it is only necessary to move a lot of data if the entry found is very far towards the back of the list and therefore seldom looked for. Accordingly, this also occurs only seldom. All in all, this means that rather common search requests are dealt with very quickly, while uncommon search request take very long. The program adapts perfectly to the user's habits.

I hope that I was able to give some of you "new" ideas for your own programs with this article. Having said this, I wish you much fun while programming and trying things out!

Recommended reading for this and similar problems:

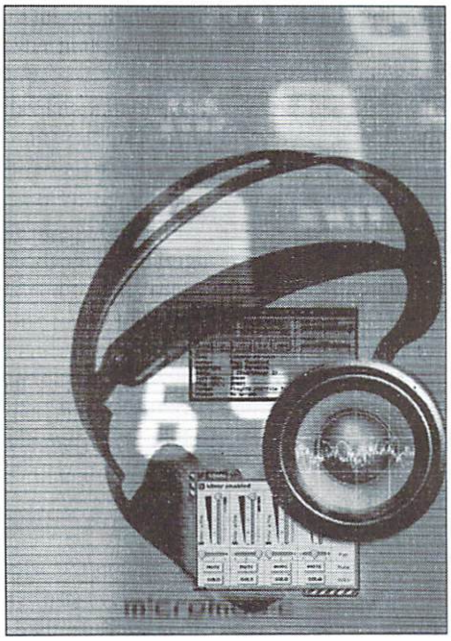
Cormen, Leierson, Rivest: /Introduction to Algorithms./ MIT Press, 1990.
H. Klaeren: Vom Problem zum Programm, 2. edition, B.G. Teubner Verlag, 1991.

Cover Disk

Audio CDs on the C64

by Torsten Baade

The utility "AUDIO" ("AUDIO-D" for German users) is meant for playing audio CDs via a CD-ROM drive which is connected to a CMD HD. It is a C64-adapted and extended version of the audio CD player developed in BASIC by Achim Taege during his SCSI tutorial, which was published in GO64! issues 5/98 to 10/98. It runs on a C64 as well as a C128, the latter using 40 or 80 columns. However, the very first time you try it should be in C64 mode, because it configures itself with some code. With just a little BASIC knowledge, you should be able to adapt and extend this program to your own needs quickly and easily.

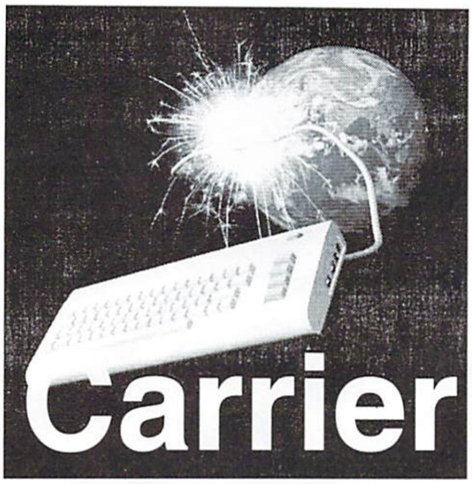


After you start the program with "RUN", it will look for a CMD HD and then for connected SCSI devices. Unfortunately, this takes some time. If a CD-ROM drive is been found, you can save the slightly modified version of this tool on the latest used drive by pressing <Arrow left>. In this version, the current device numbers are stored in line 7010. The new program is now called "AUDIO CD PLAYER" and should be used instead (on both C64 and C128), because you don't have to wait for the drive check anymore.

If your device numbers should change (alterations, new drives...), just delete line 7010 before starting this program, and the player will scan for the drives again. The rest of this tool should be self-explanatory. By pressing <P>, the whole CD starts playing, and the program will end so that the computer is free for other applications/ games, even if these have to access the hard drive.

Up to now, there has just been one known problem with an Apple CD-ROM drive. Thanks must go to Oliver Weißflach for his suggestions and bug reports.

(ws)



Picture: Sebastian Bach

Carrier Detect

Instant Messaging, Part 2

Due to a mistake we accidently didn't print part 2 of Gaelyne's 'Instant Messaging' (see GO64!/ Commodore World 10/1999). Well, here it is, finally.

by Gaelyne R. Gasson

Getting Started

Using Micq is fairly straight forward but you'll need to know what a UIN is before going further. Every user is given a UIN (Universal Internet Number) to use, and basically this is you're ID number for using ICQ and ICQ clone programs. As we don't think of ourselves as numbers, a nickname is also used to associate a name with the number. For instance, my nickname online is Gaelyne (original, eh?) and my UIN is 5069384. Another thing you'll need to know before hand is what password you want to use.

To start Micq, type "micq" on the command line. The program will ask: "Enter UIN or 0 for a new UIN:". If you've never used ICQ before, type "0", and you'll be prompted twice for a password (the second time is to verify that it's password you meant to use). It will request your UIN from the ICQ servers and you'll see something similar to this, but with actual numbers instead of the x's:

```
Your new UIN is xxxxxxxx!
Matt's ICQ clone compiled on 12:38:16 Jul 16
1999
Version 0.4.1 Now with a new version number.
No  Mirabilis  client  was  maimed,  hacked,
tortured, sodomized or otherwise harmed
in the making of this utility.
Using intel byte ordering.
Login successful! UIN : xxxxxxxx
 IP : 139.130.137.106           00:15:08
============================================
xxxxxxxx: Your status is Online
============================================
Users offline:
 Micq Author                     (Offline)
============================================
Users online:
============================================
Micq>
```

Giving Yourself a Nickname

So others will have a name to go with your UIN, type "update" and press return. You'll be prompted for your new nick name. It will then prompt for your first and last names, and your Email address. You can leave these blank if you like. Adding your Email address can help others find you online with a feature that's explained in more detail below. The next question is "Do you require Mirabilis users to request your authorization?" Mirabilis is the company that owns ICQ, and one of the features their software offers is the ability to not accept messages from users unless they've been authorised by you. Not all of the 3rd party clone ICQ programs have this option, but it can be useful if you don't want to get messages from total strangers. Answer "yes" or "no" to this prompt, and the information will be updated in your .micqrc file.

Your Contact List

At this point, your next step would be to add a few people to your contact list. The contact list appears above and starts off with just the author of the Micq program listed. When a user in your contact is online their nickname would appear under the "Users online" display. To find someone you know online, you can use the command: "search user@email.address". You'll see the "Micq>" prompt again, and in a few seconds it will either respond with the nickname and UIN matching the Email address, or it will tell you that the search is done and all users have been found (without showing the information as there isn't any to show). Once you have a friend's UIN, you need to tell the program to add it to your contact list. The command for this is "add uin nick". For instance, to add me to your list, you would type "add 5069384 Gaelyne". Micq will respond by showing your newly updated contact list:

```
Users offline:
 Micq Author                    (off-line)
 Gaelyne                 (off-line)
============================================
Users online:
============================================
Micq>
```

Even though someone is "off-line", they may be online, but marked as 'invisible'. You can send a message to a user even when they aren't online. If they are online, they will receive the message almost immediately, and if not, the next time they start their ICQ program, the message will be there waiting for them.

Sending Messages

To send a message, the quick way is to type "msg nick/Start typing your message here". For example to send a one line message to me, type: msg Gaelyne/Hi. I'm trying this Micq thing out.

Sometimes you might want to send a longer message. In some ways, this is even easier. Just type "msg nickname" and press return. It will tell you you're typing a message as the prompt will change to "msg>". When finished writing your message, type a period (".") on a line by itself. If you get halfway through typing your note and decide not to send it, type a hash ("#") on a line by itself to cancel.

You can type about 4 full 80 column lines of text, a 'limitation/feature' of ICQ itself (it's not intended for monologues). Once you reach this limit, Micq will automatically send your message. This means your last line of prose may not be sent, so it's best to stick with 4 lines. One of the most useful features of Micq is that cursoring up will cycle through lines you've typed. This way you can send the same command or message line again with minor changes.

Naturally, there are times when you just don't feel like chatting, and there are features you can use to set your online status (away, do not disturb, occupied, etc.). You can even set it so you always appear as offline to certain people (or everyone) but show up as online for people that you want to communicate with.
The commands to change your status are listed in the side bar.

New Files and Directories

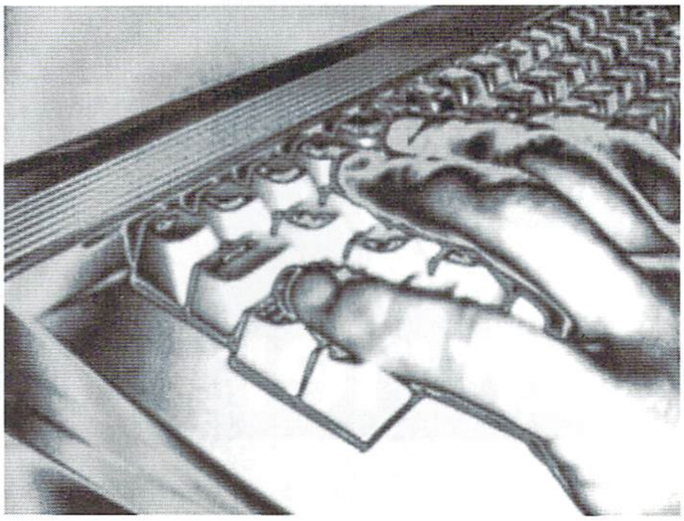
The first time you use Micq, it creates a new file in your home directory named ".micqrc". This is your configuration file that you can change some of the defaults the program uses. The next time you use the program, it will automatically connect with your UIN and password, as this information is stored in the .micqrc file. For this reason, it's a bad idea to use you the same password as you use for your ISP.

Micq keeps a log file of your conversations in the .micq.logs directory. The log files are listed by the UIN of the person you've chatted with. Knowing where the log files are can be useful for double checking information, especially if someone told you about a web site or gave you information you want to see again.

The Rumour Mill

One of the annoying things about ICQ is that some people (including friends and family members) truly need to get a clue. Have you ever heard a juicy rumour and immediately phoned a friend to tell them? Imagine that instead of phoning ONE friend you could send all 10,000 of your friends the same rumour with just a few key presses? This happens everyday with ICQ. People that I would have thought knew better send me messages like "Don't accept messages from Joe Blow, he's bad news. Pass this on to everyone on your contact list." Chances are, the person who sent me the message don't even know Joe Blow. The rumour problem has got so serious that ICQ actually has a web page devoted to it at http://www.icq.com/support/rumors.html and another useful resource is http://diamond-back.com/icqlies.html. Another annoying habit well meaning people have are sending URLs (web page addresses) of every cute web page they run into. One well meaning fellow sends me at least three a day.

It's not all that bad though. I use ICQ to be able to offer immediate online support and for this, it does the job well. The annoyances of a few people sending me rumours and web pages really doesn't detract from the usefulness of the utility. Some people really enjoy it to the extent that there are now ICQ groups covering just about every topic under the sun. Mirabilis has a web site at http://www.mirabilis.com devoted to the ICQ network that you might find useful in finding friends, family and people of like mind.



ICQ All or Some of the Time?

When we use Micq we need to make the choice of whether to have it running the whole time we're online or starting it every once in awhile to see if any friends or contacts are online or to send a quick note to someone. To have it all the time you're online, you'll need to learn to use a few new keystrokes to put it in the background while you do other things like read mail and such. If you receive any messages, Micq will show them to you on the screen.

To set it up, once you've started Micq, press CTRL-z to suspend it. You'll see your normal Unix shell prompt. Now type "bg" (for "background") and press return. What this does is let the system know that you didn't really want to suspend Micq, you want to let it continue to run, but in the background. Now you can read your mail or do other things and other people using ICQ will know that you're online. When you want to return to Micq again, type "fg" (for "foreground") to bring it back. Use the same CTRL-z and then "bg|" sequence to send it to the background again.

After a few minutes of inactivity, Micq will automatically set your status to "Away". This is ok, as you still appear as being online, but other ICQ users would know you're busy doing other things. This text will appear on your command line, it looks as though you're within the program, but if you just press return you'll see your normal command prompt. Any messages you receive are also displayed on the command line while the program is in the background too. Pressing return will bring the command prompt back. There are times when the ICQ server restarts itself, and under this condition, the message that Micq displays is "Server sent "Go Away!" command. Most ICQ programs reconnect, but unfortunately Micq isn't one of these. In this case, simply restart the program.

Wrap Up

Believe it or not, I'm really not all that keen on chatting, even though I've found ICQ to be very useful when online. One of the things I like about ICQ though is the ability to see if family is online, and it's nice for when you really only want to say hello or ask a quick question, or in the case of some people I know, sending virtual hugs and kisses. It's quick, it's not too difficult to use, and it helps me stay in touch with friends and family all around the world in "real time" (unlike Email, which has delays). I hope you give this new world a try and see how you enjoy it. The good news is that when SLIP/PPP becomes available, our own Commodore ICQ clone will likely be available too.

Miribilis, ICQ, and, THE ICQ NETWORK are Trademarks and/or Servicemarks of Mirabilis. Micq and its author are in no way affiliated with ICQ or Mirabilis.

Gaelyne Gasson is the author of The Internet for Commodore C64/128 Users and spends her days as co-system administrator for VideoCam Services. She can be reached via Email (gaelyne@videocam.net.au) or ICQ # 5069384.

CMD WAREHOUSE BARGAINS
1-800-638-3263

CMD's Warehouse Bargains include many hard-to-find used and (where noted) new items. In most cases we have only one or two of each item in stock, so be sure to place your order before your selection has been taken! A minimum order of $20.00 applies to all Warehouse Sale orders. Shipping charges are not included in the prices shown. (*See our main advertisement elsewhere in this issue for shipping information*).

TELEPHONE ORDERS ONLY ON THESE ITEMS

Mailed in orders for non-reserved out-of-stock items will result in a merchandise credit. No cash refunds. All items listed here are sold "as-is", and all sales are final. Prices and specifications subject to change without notice. Not responsible for typographical errors.

BOOKS

32 BASIC Programs for the C64	$8.00
Anatomy Of The 1541	$8.00
Basic Computer Games	$6.00
C128 Computer Aided Design (new)	$8.00
C64 Programmer's Reference Guide	$25.00
Commodore 64 Favorite Programs Explained	$10.00
Commodore 64 For Kids from 8 to 80	$8.00
Commodore 64 Games Book	$5.00
Commodore 64 Programs for the Home	$8.00
Compute's First Book of C=64 Sound &Graphics	$10.00
Computes Second Book of Commodore 64	$10.00
Computes Second Book of Machine Language	$10.00
"Guide to Care, Feeding and Training of C-64"	$6.00
How To Use The Commodore 64 Computer	$10.00
Inside Commodore DOS	$10.00
Osborne 1 User's Reference Guide	$15.00
Superbase The Book	$10.00
The Elementary Commodore 64	$10.00
The Official Book Of King's Quest-Daventry	$10.00
Using & Programming the C64	$8.00
Using Newsroom	$10.00
Your Commodore 64-A Guide to the C 64	$10.00

BUSINESS

64 Powerpack-NEW	$10.00
Better Working File & Report (new)	$15.00
Business System- Sales Analysis Management	$10.00
Datamanager 128	$15.00
Datamanager 128	$20.00
Datamanager 2 (new)	$15.00
Easy Finance 1	$5.00
EPYX Microsoft Multiplan Spreadsheet	$10.00
Financial Cookbook-NEW	$8.00
General Ledger for the C=64-NEW	$15.00
Paperback Filer 64	$10.00
Paperback Planner 64	$10.00
Payables for the C=64-NEW	$15.00
Planner's Choice -Personal Planning System	$8.00
Power Assembler-(new)	$10.00
Practicalc 64	$5.00
Practicalc-(new)	$10.00
Practifile 64-(new)	$10.00
Receivables for the C=64-NEW	$15.00
RUN Productivity Pak II	$8.00
Silent Butler (Check Writing system)	$15.00
Superbase Starter-NEW	$10.00
Swiftcalc 128	$20.00
Swiftcalc 64	$10.00
The Consultant (Database Management)	$15.00
The Home Accountant	$15.00
The Home Banker	$10.00
The Home Manager	$10.00
The Manager	$10.00
The Word Machine/Name Machine (new)	$5.00
Valuecalc	$8.00

CARTRIDGES

Alf in The Color Caves	$10.00
Alpha Build	$10.00
Big Bird's Special Delivery	$10.00
Bubble Burst	$10.00
Dragonsden	$8.00
Facemaker-(new)	$10.00
Heskit	$10.00
Kids on Keys-(new)	$10.00
Kindercomp	$10.00
Lemans	$8.00
Linking Logic	$10.00
Mach 5 Cartridge	$15.00
Mastertype (new)	$10.00
MS Pac-Man	$8.00
Pinball Spectacular	$8.00
Radar Rat Race	$8.00
Sea Speller	$10.00
Speed Bingo/Math	$10.00
The Final Cartridge	$15.00
Up & Add 'em	$10.00

CARTRIDGES (No Docs)

Blueprint	$8.00
Clowns	$8.00
Ducks Ahoy	$8.00
Kindercomp	$8.00
Lemans	$8.00
Linking Logic	$8.00
Magic Desk I	$8.00
Maze Master	$8.00
Memory Manor	$8.00
Popeye	$8.00
Speed/Bingo Math	$8.00
Up & Add 'em	$8.00
Vic 20 -Hesware-Gridrunner	$5.00
Vic 20- Avenger	$5.00
Vic 20-Jupiter Lander	$5.00
Vic 20-Omega Race	$5.00

DESIGN

Awardware	$8.00
Delta Drawing-(new)	$6.00
Jingle Disk-NEW	$6.00
Printer's Devil for Print Shop	$8.00
The Christmas Kit-NEW	$10.00
The Newsroom	$10.00
The Newsroom (new)	$15.00
The Print Shop	$15.00
The Print Shop Companion	$10.00

DISK UTILITY

Commodore 64 Software Bonus Pak	$5.00
Disk Utilities	$5.00

EDUCATIONAL/CHILDREN

Computer Science	$5.00
Early Learning Friends	$10.00
Easy Lesson and Easy Quiz	$5.00
English I	$5.00
ENGLISH II	$5.00
First Men on the Moon -Math-(new)	$6.00
Grade A Math	$5.00
In Search of the Most Amazing Thing-(new)	$6.00
Jungle Book Reading-(new)	$6.00
Koalagrams Spelling 1-use with Koala Pad-(new)	$6.00
Logic Builders	$8.00
Math Busters	$8.00
MECC Expeditions	$8.00
MECC Odell Lake	$8.00
Paint-A-Rhyme -for use with Koala Pad-(new)	$6.00
Peter Rabbit Reading-(new)	$6.00
Snooper Troops Case #1 (new)	$6.00
Snooper Troops Case #2	$5.00
Snooper Troops Case #2 (new)	$6.00
Snoopy's Skywriter Scrambler	$10.00
Story Machine	$8.00
SUNBURST-Memroy Building Blocks	$10.00
Tales of Mystery	$8.00
The Argos Expedition	$8.00
Up for Grabs-(new)	$6.00
Weather Tamers	$8.00
Where in Europe is Carmen San Diego (new)	$20.00
Where in the USA is Carmen Sandiego	$20.00
Where in the World is Carmen San Diego (new)	$25.00
Wizard of Words	$10.00

GAMES

20 Solid Gold C64/128 Entertainment Hits	$10.00
20 Solid Gold Hits -5 Disks	$10.00
Aliens	$4.00
Amnesia	$6.00
Archon	$10.00
Ardok-The Barbarian	$6.00
Batman	$10.00
Batman-The Caped Crusader	$10.00
Beach-Head	$6.00
Below The Root (NEW)	$10.00
Beyond Zork-NEW	$6.00
Blood Money-NEW	$6.00
Blue Angels-Formation Flight Simulation	$10.00
Bop'N Wrestle-NEW	$6.00
Bureaucracy	$10.00
CavemanUgh-lympics	$4.00
Championship Baseball	$6.00
Championship Golf (new)	$6.00
Championship Wrestling	$6.00
Chomp (new)	$4.00
Classic Concentration	$6.00
Commodore 64C Test Pilot-NEW	$6.00
Conflict in Vietnam-NEW	$6.00
Countdown to Shutdown	$4.00
Crossfire	$4.00
Cyrus Chess	$6.00
Deadline	$4.00
Demon's Winter	$6.00
Dig Dug	$4.00
F15 Strike Eagle	$10.00
Feud	$6.00
Fist - The Legend Continues	$6.00
Flight Simulator II	$15.00
GBA Championship Basketball-2 on 2-NEW	$6.00
GFL Championship Football	$6.00
Gold Record Race-(new)	$6.00
Grand Prix Circuit-(new)	$10.00
Guerilla (new)	$6.00
Hacker	$4.00
Hardball (new)	$6.00
Heartland (new)	$6.00
High Roller-NEW	$6.00
Howard the Duck-Adventure on Volcano Island	$4.00
Invaders of the Lost Tomb-(new)	$6.00
Jack Nicklaus Greatest 18 Holes of Golf-NEW	$6.00
Jaws	$6.00
Jeopardy-Junior Edition	$6.00
John Elway's Quarterback	$6.00
Jordan vs Bird	$10.00
Karate Chop (new)	$6.00
Kings of the Beach (new)	$6.00
Knight ORC	$6.00
Kung Fu II-Sticks of Death (new)	$6.00
Last Ninja 2-NEW	$6.00
Legionnaire	$5.00
Lords of Conquest-(new)	$6.00
Lords of Conquest-NEW	$6.00
Mean Streets	$10.00
Megaplay Volume 1	$4.00
Minit Man	$6.00
Murder by the Dozen	$8.00
Navy Seal	$6.00
Navy Seal (new)	$8.00
Operation Whirlwind	$6.00
Pathwords	$6.00
Power Drift	$4.00
Predator	$6.00
Presidential Campaign	$6.00
Project Space Station	$4.00
Q-Bopper	$6.00
Qix (new)	$6.00
Questron II	$4.00
Raid Over Moscow	$4.00
Raid Over Moscow	$6.00
Rambo III	$4.00
Rick Dangerous	$4.00
Robbers of the Lost Tomb	$4.00
Sanxion	$6.00
Sargon II-(new)	$6.00
Sargon III-(new)	$8.00
Serve and Volley (new)	$6.00
Shinobi	$6.00
Silent Service	$4.00
Sky Fox	$6.00
Snow strike-NEW	$6.00
Spiderman	$4.00
Sporting News Baseball	$6.00
Star Trooper-(new)	$6.00
Stationfall	$6.00
Summer GamesII	$4.00
Superman	$4.00
Superstar Ice Hockey	$6.00
Suspended -A Cryogenic Nightmare	$5.00
Sword of Fargoal	$4.00
Tag Team Wrestling	$4.00
TETRIS	$6.00
The Last Ninja	$4.00
The Lords of Midnight	$6.00
The President is Missing (new)	$5.00
The World's Greatest Baseball Game	$4.00
Thud Ridge-NEW	$6.00
Times of Lore	$4.00
Titan-NEW	$6.00
Total Eclipse (new)	$5.00
Ultimate Wizard	$6.00
War in Middle Earth-NEW	$6.00
Wheel of Fortune	$8.00
Wild Streets-NEW	$6.00
Word Flyer	$6.00
Z-Pilot	$4.00
Zenji	$6.00
Zork I	$6.00
Zork II	$10.00
Zork III-NEW	$8.00

GEOS

Deskpack 1-NEW	$10.00
Fontpack 1-NEW	$10.00
GEOBASIC (new)	$10.00

HARDWARE

1351 Mouse w/Manual-no disk	$20.00
1702 Monitor -Color 40 Column-refurbished	$99.00
1802 Monitor-Color Monitor 40 Column-refurb	$139.00
Aprotek 1200 Baud Mini-Modem (new)	$30.00
Assorted Joysticks	$5.00
C128D Keyboards (NEW)	$69.95
C128D Keyboards-REFURBISHED	$59.95
C64 Keyboard (NEW)	$49.95
C64 Keyboard (refurbished)	$39.95
CPU64 Version 1 Base Model	$139.00
Datasette	$5.00
EPYX Fast Load Cartridge w/Manual	$10.00
Homewriter 10 Printer w/manual	$45.00
Koala Pad Only no docs	$20.00
KXP 1080I Printer	$75.00
Laser 190E Printer	$30.00
Magic Mouse	$10.00
Magic Voice Speech Module	$50.00
Multiplexer	$50.00
Okimate 10 Color Printer w/manual,disk,ribbons	$45.00
STAR Gemini 10X w/Manual	$25.00
Super Expander 64	$8.00
The Voice Messenger -Speech 64	$20.00
Used SlimLine cases for Commodore 64	$10.00
VICMODEM for VIC20	$5.00

MISCELLANEOUS

10 Assorted Commodore Magazines 1985-86	$10.00
64 Doctor	$5.00
Home Video Producer	$10.00
Instant Music-NEW	$6.00
LOADSTAR Disks#46,65,66,67,68	$10.00
Synthesound 64	$5.00
Vic 20 Cassette-Intro to Basic Part 2	$10.00

MISCELLANEOUS SOFTWARE

Aerobics	$6.00
Bobsterm Pro (new)	$10.00
Commodore	$5.00
Commodore Technology	$5.00
Music Construction Kit	$5.00
SAT Score Improvement System	$5.00
Writer/File Pak 1541	$8.00

PROGRAMMING

Better Working Power Assembler	$10.00
GEOBASIC (new)	$10.00
Hesware - Graphics Basic	$10.00
Introduction to BASIC/Part I	$8.00
Oxford Basic	$10.00
Oxford Pascal	$10.00
Simon's Basic	$10.00
Start Programming w/Gortek and the Microchips	$5.00

WORD PROCESSING

Easy Script	$5.00
Mastertypes Writer	$8.00
Mirage Concepts-Word Processor -Professional	$10.00
Outrageous Pages-NEW	$20.00
Super-Text Professional	$5.00
Supertext Word Processor	$10.00
The Printed Word	$8.00
The Write Stuff 128-NEW	$20.00
Word Writer 128	$20.00
Word Writer 6-NEW	$15.00
Wordpro 3 Plus/64-NEW	$15.00
Wordpro w/turbo Load & Save-NEW	$15.00
Writers Choice	$8.00

Computer...

Party...

Compare?

ComPARADE

C64 and AMIGA Computerparty

- From/Von: Fr. 10/27/2000
- To/Bis: Su. 29/10/2000
- Where/Wo: Amperhalle Emmering (Germany)
- Special C64 entrance fees/ Spezieller C64 Eintrittspreis
- Games(No LAN-GAMES!)/ Spiele(Keine LAN-Spiele)
- No Quake, etc.
- Democompetitions & -presentations!
- Visit our homepage for more information/ Weitere Infos im Internet:

http://www.comparade.de

Reservations(lower entrance fee)/
Reservierungen(kleinerer Eintrittspreis):

comparade@gmx.net

Since 1995 - Seit 1995

Comparade 9 - The C64 and Amiga Scene event in south Germany

Comparade 9 - Das C64 und Amiga Szene event im Sueden

Amiga-Democompetition

C64-Democompetition

OnStage-Competitions

2 Videobeamers

44 hrs. Computer Fun

KEINE MACHT DEN QUAKERN

Post-Info:

Andi "homecat" Brandmair
Am Eichenhain 3
D-82275 Emmering
Germany
Alemania
Germania