The Magazine for true Computer Fans
Das Magazin für wahre Computerfreaks

3/2000

DM 9,17 USD 6,17

GO64!

GO64! commodore WORLD

CW

CLiPS CORNER

BEI DER ENTWICKLUNG LIVE DABEI

WITNESS CLiPS DURING ITS DEVELOPMENT

COVER

DRUCKEN AM C64

GRUNDLAGEN
POSTSCRIPT
GEOS UND DRUCKER
PC-DRUCKERKABEL AM C64

PRINTING WITH THE C64

THE BASICS
POSTSCRIPT
GEOS AND PRINTERS
PC PRINTER CABLE ON THE C64

DREI JAHRE GO64! MACHT MIT BEI DER GROSSEN UMFRAGE UND GEWINNT!

THREE YEARS GO64! PARTICIPATE IN THE BIG SURVEY AND WIN!

GODOT WORKSHOP, SUPERCPU CORNER, BASIC EFFECTS, ASSEMBLER, WORLD WATCH

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

THE HOTTEST UPGRADE EVER CREATED FOR GEOS

Wheels

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 online 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.

••• All Major Credit Cards Accepted •••
Mastercard • Visa • Discover • American Express

CMD

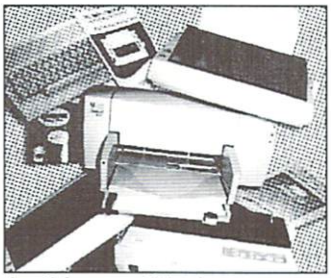
Creative Micro Designs, Inc.
P.O. Box 646
East Longmeadow, MA 01028
Info: (413) 525-0023
Fax: (413) 525-0147
Visit our Web Site at http://www.cmdweb.com/

CONTENTS

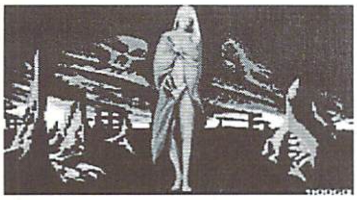
- March 2000 -

5

Printers



Printing with the C64 was always an important thing for many people. In this month's issue you will find everything from the basics to hardware projects and printing posters. Let's start on page 5!



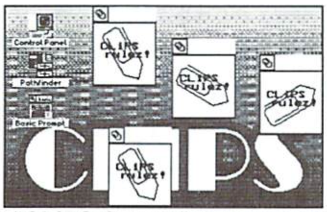
18

Graphic Tutorial

In this month's tutorial a real scene wizard will reveal some of the secrets of a professional artist. If you want to know how to combine interlaced hires graphics and sprites, better turn to page 18 now.

26

CLiPS - Corner



"CLiPS - live during its development" is the slogan of our new "CLiPS Corner" department. We will show you the features of this revolutionary new operating system for CMD's SuperCPU step by step, and you will get all the information directly from the source. The fun begins on page 26!

28

GEOS and the Iomega Zip

A CMD-HD is all you need to use exciting state-of-the-art hardware components. An Iomega Zip drive on your C64 is a perfect proof for this fact - you can stuff 100 MB of data on a single disk. Niko Metz will show you how easy it is to use a Zip drive under GEOS, beginning on page 28.

[news]

[hardware]

[cover disk]

[tutorials]

[geos]

[letters]

[departments]

In the next issue...

Commodore Cult

Have you ever heard of a Commodore 264, the VC-10, or a Commodore 232? No? Then be sure you don't miss the next issue of GO64!/ Commodore World! In addition to this, we will also feature a lot of background information on Rainbow Arts, one of the leading software companies of the late 1980s and early 1990s.

Plus: A special report on the HobbyTronic 2000 fair in

Dear reader,

would you believe it? We're glad to announce that with this issue of GO64!/ Commodore World, we celebrate our 3rd anniversary! I'm sure some of you will remember what I wrote in our premier issue back then, that we wanted to bring out "the best C64 magazine ever". With this in mind, we started the GO64! project three years ago; and now we can look back to a whole 37 issues that capture the Commodore fascination from many different sides. From the presentation of GeoDOS 2.0, Wheels vs. MegaPatch 3, reports about the HardSID and the SIDStation, to the game hit "Crazy News"; from the SuperRam option for the SuperCPU, to this year's sensations, CLiPS and "The Wave", the graphical web browser - all of these events have moved the minds of the C64 community, and we're proud to say that we were always a part of them, and surely have contributed our small share to the never-ending success of the C64 during the last years!

The joining of GO64! with CMD's Commodore World last August pushed us a great deal closer toward our dream of a global family. Three years ago our readers were in Germany, Austria and Switzerland, and now we're read in Argentinia, Australia, Belgium, Brazil, Denmark, Germany, Great Britain, Finland, France, Greece, Ireland, Italy, Japan, Yugoslavia, Canada, Luxembourg, Mexico, New Zealand, the Netherlands, Norway, Pakistan, Poland, Sweden, Switzerland, Spain, and the US! It's overwhelming for us!

And like back in 1997, when we were not aware of how much work it means to publish a regular magazine, we were just as surprised in the summer of 99, when we found out what an effort it was to switch to bilingual publishing, in spite of all the preparations we had made.

The work put into GO64!/Commodore World by our editors, translators, proofreaders, graphicians and lay-outers equals that of a full-time job. All funds the GO64!/Commodore World yields are fully returned into the magazine, and for years, our collaborators have been doing a job that goes far beyond a usual honorary post. I'd like to use this occasion to express my deepest, most honest thanks to all of them.

The delays that occurred during the last months are partly due to the fact that every one of us can only work for GO64! as a hobby, besides his "real" daily profession. I'd like to ask our readers to understand this, and I promise we'll do our best to catch up in the long run.

Of course, this issue isn't the end to all this. The 64 community doesn't rest - there are always new projects and new products, and we want to report on them in the quality you're used to. In order to be able to produce "the best C64 magazine ever" (which is still our aim), we have to continue developing the magazine and its contents, and we have to listen to the wishes (and the dislikes) of our readership. For this sake, we take our third anniversary as an occasion to start another big survey, which, for the first time, will also be available through the internet (**www.go64.de**). Of course, if you prefer the classical way, you'll find a questionnaire on this cover disk to print out. Once again, we're giving out prizes as a little teaser (JiffyDOS 64, RTC for your FD 2000, parallel cable set for your 1541 and the fullprice game Crazy News). I hope many of you will take part, as we at GO64!/Commodore World need this input from our readers. This is your chance to influence the appearance and content of the magazine!

Happy Birthday GO64!/Commodore World! 64 forever!

And now: have fun reading!

Enno Coners

News Flash

Dreamon News

By now, there is a new preview version of the SCPU monitor Dreamon (see our previous report) with Freeze Points correctly working. The VDC can be used as a second screen. It is fast advancing toward the state of perfection. So naturally we feel especially honored and happy by being able to tell you that this fantastic tool will be published exclusively in the GO64! magazine! Until that time we will of course keep you abreast of any news on the subject. *(ws)*

New SuperCPU demo

'It takes time to do a thing well' was probably the idea the programmers from Dmagic had concerning their first SuperCPU demo which took quite some time to finish. The name of this fine production is 'SuperCPU kicks' and it is available at the Internet address

http://come.to/supercpu

(st)

D4 adventure system available for free on the Internet

All those of you who have ever felt the urge to start designing adventure games by simple mouse clicks now have the opportunity to download Tectron's D4 adventure system from the Internet. The author Tobias Erbsland, who is especially famous for the highly prized mag system of the Digital Talk, has declared the D4 system's software to be freeware. The program, which has been public for some years now, makes it possible to design professional graphics adventures without any special knowledge of coding. Prototypic example: 'Leisure Suit Leo 2'.

ftp://ftp.profzone.ch/c64/d4

(vr)

Party results:

The results of the Out-of-Order party in Hoppstädten from December 28th to 31st:

Demo:

0. 'Banging Scorpe' by Out of Order (first place renounced)
1. 'Atomic Age 64' by Carlos/Cascade
2. 'Exhumed' by Silverfox/Sanity
3. 'Scorpe' by Spermatika

4K Intro:

1. 'Y2K' by Cyberpunx

Graphics:

1. 'Kellerkind' (Slum kid) by Zealot/Out of Order
2. 'Damnation' by Deathworks
3. 'Cevi' by Ollo

Music:

1. 'Pain remains' by Taxim/Banshee
2. 'Compotune' by Carlos/Cascade

(vr)

Picture: Sebastian Bach

Postscript on the C64

The year was 1440. The event changed the world. A great German named a Johannes Gutenberg created a new printing process. It lined up little pieces of metal type to create an entire page which could be quickly reprinted again and again. Up until his invention, the people in Europe had to write out books by hand. Gutenburg replaced scribes with typesetters and made handwriting books obsolete!

by K. Dale Sidebottom

The printing process was changed and improved many times over the years, but typesetters continued to be an invaluable part of the process until recently. Then suddenly a new revolution swept the publishing world! The World Book Encyclodpedia (Vol. 5, p 158) states, "Desktop publishing, often referred to as DTP, began in the mid-1980's

"Three technological breakthroughs...created the field of deskop publishing: (1) the Apple Macintosh personal computer, developed by Apple Computer, Inc., (2) Apple's LaserWriter printer, and (3) PostScript, a progrmming language developed by Adobe Systems, Inc."
Over half a century after Gutenburg's work made handwriting books obsolete, desktop publishing made typsetting books obsolete! PostScript was at the forefront of this movement. It was the first popular desktop publishing language!

For awhile it seemed that PostScript fell out of favor even as the Apple Computer company struggled, almost going under. Microsoft seemed to "own" the PC world, and it had little use for PostScript or any other product which represented competition. Then PostScript got a terrific "shot in the arm" from an unusual source. Somehow it "unofficially" became the document language of the internet. If you want to receive a document on the internet today, it is usual "packaged" in PDF format, which means Postscript Document File!

In part because of its constant use on the internet, PostScript is on the rise in popularity! Almost any print shop in the United States can process PostScript files for you. The question for us as Commodore users is this. Realizing that PostScipt helped revolutionize the entire publishing world, what can it do to revolutionize the printing of personal documents from our Commodores?

Berkeley Softworks opened the door by creating a geoPublish in 1987. (An improved version was reissued in 1988.) They not only enabled our platforms to perform desktop publishing, but they included specail files which made our machines PostScript compatible. The programs, geoLaser for geoWrite files and geoPublaser for geoPublish files, enable us to create and send GEOS documents as PostScript files. So what?

I asked a friend recently, who owns a print shop, what programs he uses to make a living. He mentioned five programs, one of them being Adobe's Illustrator. That is purely a PostScript program. In other words, PostScript was one of the first, and is still one of the finest publishing programs in the world, and it is available to US and our Commodores today!

Unfortunately, Berkeley Softworks abandoned our 8-bit machines soon after geoPublish was introduced. That was TWELVE years ago! Back in 1994, I began to wonder if I could modernize my geoPublish documents by learning the elements of the PostScript language and personally adding my own code. This would enable me to create document from my Commodore which could access the many amazing features of the PostScript language which cannot be accessed through geoPublish. For instance, I realized that due to its popularity, there are thousands and thousands of PostScript fonts and cliparts (called EPS files) which I would love to use. How can these be included? PostScript can rotate text and graphics, as well as print in color. How can I incorportate these into my Commodore printing!

I now know enough PostScript code to be able to accomplish all of these things, and I find my Commodore much more exciting because of it. I have written a booklet called "PSPFTC: PostScript Printing from the Commodore" which explores these things in detail. It is included in a disk called the Laser Lovers Disk which can be purchased for $25.00 in US currency.

Also, any person outside North America can receive our LUCKY Commodore Club newsletter, called the LUCKY REPORT. It demonstrate what Postscript printing from a Commodore can do. It is generally published 10 times a year and often includes a color cover. (We do not sell subsriptions. You actually pay $25.00 to join our club for a year and the newsletter is sent to you freely as a member.)

What makes PostScript exciting today is the addition of a program called Postprint II. This program is written by Maurice Randall, author of Wheels, and it takes off where geoPublish quit. It allows us a Commodore users to format 16 million color JPEGs into our geoPublish documents. More exciting improvements are in store for the near future.

In America, we have a saying, "I have good news, and I have bad news!" This also seems to apply to PostScript printing.

The good news for Commodore users is that PostScript is a device-independent language. If you can send the code, and we can, then you can create copy equal to quality to that produced by any personal computer on the planet!

In other words, the quality of your documents is determined by your printer, not your computer! In addition, PostScript can automatically access the best resoluation and quality in whatever printer you use. The same program will work in any PostScript compatible printer anywhere in the world! (This is true as long as there is not a conflict in the level of the PostScript language.)

The bad news is that PostScript costs a little more and its in English. I can buy a cheap color printer in the United States for as little as $100, but in order to purchase a quality PostScript color printer which can print color photographs, I would have to pay $500 or more!

Although the first printing revolution began in Germany, DTP began in the United States. This means that PostScript code and most PostScript materials are in English. It is more difficult to master for Commodore enthusiasts in non-English speaking countries. However, PostScript was designed to be an invisible language. Using geoPublish and Postprint II, you can yet do many things to create superb PostScirpt documents without having to read any code. However, even those programs are in English, and Postprint II requires the user to have Wheels. (Especially in Germany, Megapatch 3 is the preferred GEOS upgrade.)

If anyone has any questions about PostScript and the Commodore, or wishes to order the Laser Lovers Disk or join LUCKY to receive our popular PostScript printed LUCKY REPORT, you can write me at

K. Dale Sidebottom
P. O. Box 303
New Albany IN 47151-0303
U.S.A.

You can also email me at

luckykds@iglou.com

The important thing about PostScript, I believe, is that nothing in Commodore can compete with it for quality printing. It gives us the tools to print superb documents. It, therefore, makes us and our Commodores look good! That alone should make PostScript important to any Commodore enthusiast.

K. Dale Sidebottom

Load your Canon!

The following article may be freely distributed and reprinted without permission provided credit is given to its original publication in the July/August 1998 issue of "Double Click", the newsletter for the Lansing Area Commodore Club.

by Maurice Randall

How many of you purchased a new Canon BJ series printer only to discover you can't configure it from your Commodore? Yet, you hear stories about other people using them but you just can't seem to get the results you expected. Your old 9-pin dot-matrix printer sure was a lot easier to deal with. Your old printer had dip switches and front panel controls that allowed easy configuring for font type and pitch, margin settings, page size, automatic linefeeds, etc. This new printer you just bought only has two switches, one for turning the printer on and off and the other for form feeding a page through the printer.

How do you alter the settings on these new Canon BJ printers? Others are using them because they might also have a Windows-based computer. The printers are shipped with software that runs in Windows and allows settings to be changed using a program that you load into your computer. The computer then sends special instructions to the printer which causes the printer to set itself up just as if you flipped a dipswitch or pressed a control panel switch.

This is still inconvenient for someone to switch the printer over to another computer just to reconfigure it. Wouldn't it be nice if we could do something like this right from our 64 or 128?

Ten years ago, these printers were built with controls right on the printer. Those were the days when you could unplug the printer from one machine and plug it into another. If a setting needed to be changed for the other computer, a simple flip of a switch was usually all that was needed. Now, in order to use a printer, you also need software on each machine. The sad thing is the software is primarily only available for Windows and Macintosh machines. So, plug in the printer, load the software, and play. "Plug, LOAD, and Play."

If Canon used these Plug, Load, and Play printers ten years ago, they also would have included software for our Commodores in order to configure them. If you call the technical hotline at Canon, you'll get connected to a no-brain tech person that will tell you their printers can't be used on a Commodore. That tells you how much they really know about computers, let alone their own line of printers. In fact, you can call Canon all you want and you won't find a single individual working for the company that knows anything about any special codes or commands that can be sent from a computer to the printer for configuring it.

We're on our own, so let's get down to business and see how to set these printers up to get them working the way we need them to.

The first thing you need to do is to look at your printer. Is this the model of Canon that was made with dip switches? If so, then this article doesn't pertain to you. The older Canon BJ-200 had dip switches. You can configure those printers. The BJC-4000 also has dip switches. But then Canon introduced the BJC-4100 to take the place of the 4000. They got rid of the dip switches and cheapened up the printer. Every BJ model since then comes without dip switches.

Since our Commodores have BASIC built in, it's pretty easy to write a little program to configure the printer. The biggest problem is what goes into the program. What kind of commands are we supposed to send to the printer? Well, I did some investigating and came up with some of the commands we can use.

SWITCHING MODES

Most of these printers support two different emulation modes. The default is BJ mode. The other mode supported is Epson. Canon likes to refer to this as LQ mode, so we'll do the same in this article. Perhaps the software you use with your printer expects it to be in LQ mode. Study Program 1. This is a program that will switch the printer to LQ mode.

```
Program 1 - Configure for LQ mode.

10 c$=chr$(10)
20
```

```
d$=chr$(27)+chr$(91)+chr$(75)+chr$(
2)+chr$(0)+chr$(0)+chr$(31)
30 s$=d$+"bjlstart"+c$
40 e$="bjlend"+c$
50 open4,4,5
100 print#4,s$;
110 print#4,"@setcontrolmode=lq"c$;
120 print#4,e$;
999 close4
```

Each program that I'll discuss in this article will be very similar to Program 1. In fact, lines 10-50 will always be the same in each program. These lines set up some variable strings that we'll use throughout the program, making it easier to not only enter the program, but also make it easier to read. For instance, look at line 20. This defines d$ as a string containing a whole bunch of character string values. If we work on a program that needs to send all these more than once, it's much easier to just define the whole bunch in d$ and then use d$ throughout the program.

Line 50 opens a channel to the printer. This program and the others in this article assume you're using a printer interface that can be connected to the Commodore serial port. Using a geoCable would require a different method of communication. If you're a geoCable user, this might be the only difficulty you'll encounter, the fact that you'll have to plug in a separate interface each time you want to change the settings in your printer.

Line 999 closes the channel to the printer and ends the program. Everything in between opening and closing the printer are the lines that send commands to the printer. In this case, lines 100-120 are configuring the printer for LQ mode. Line 100 sends s$ which contains a series of codes that tells the printer some special commands will be coming. The chance of a this series of codes happening in normal everyday printer use is very slim and will likely never happen. So, when the printer sees this, it will be ready to accept the commands that will be following. And that is what we are sending in line 110.

This is the "@setcontrolmode" command. In the case of this program, we set this command's parameter equal to "lq". Following the command and its parameter is c$. Our program defines c$ as chr$(10). This is a normal linefeed character and the printer expects to see this character at the end of each command.

In line 120, we end the command sequence by sending e$ which was defined earlier in the program as "bjlend" and the linefeed character. When the printer sees this, it will resume normal operation along with the new settings it just received. In this case, the printer will now operate in LQ mode, otherwise known as Epson
mode.

You might find that some software works good in LQ mode, but there just might be some that work better in BJ mode. Why is this? Well for one thing, the LQ mode that is emulated is similar to a 24-pin Epson printer. When printing text, the printer will work great. If LQ mode works for you, then leave it that way. But if you use a software that prints in graphic mode and the results appear to be stretched vertically, it's likely due to the 24-pin emulation. The vertical resolution doesn't work the same as with an older 9-pin printer. But, when the printer is in BJ mode, it can simulate a 9-pin printer's graphics mode. The AGM mode within the BJ mode is used for this. But you need to watch out for some of the subtle other differences in BJ mode. That's why we have this article though, isn't it?

If you need to put your Canon into BJ mode, you only need to make one minor change to Program 1. Bet you can figure out what that change might be. Yup, just change line 110. Change the parameter from "lq" to "bj". When you run the program, you'll switch your printer to BJ mode.

To be continued in the next two issues of GO64!/Commodore World Magazine. (ad)

Picture: Sebastian Bach

A little Epson-Manual

In this article I'll try to give you an introduction to the basic ESC sequences used by modern Epson compatible printers. It's mainly aimed at those without a decent manual for their printer, be it because they bought it second-hand, or because the companies that make the printers deliver them without complete information, referring the user to the accompanying printer drivers for IBM compatible PCs.

by Ranjith Ekanayake Mudiyanselage

Of course, an article such as this one has certain limitations, so it won't be possible to delve into the furthest options of each and every sequence; however, after reading it, you should be able to use most of the available possibilities, as well as fully configurate your word processor to operate the printer correctly.

Attention: All sequences are shown as the ASCII codes sent to the printer. If you're using the PRINT# command, you have to send the codes via CHR$(), combining them with semicolons: for instance, the sequence 27 69 is sent with the command PRINT#1,CHR$(27);CHR$(69). The hexadecimal conversions are given as well, since some word processing programs expect their input in hex. As usual on the C64, hex numbers are preceded by a dollar ($) sign. All codes shown have been tested on a Canon ink jet printer (Canon BJC 4200) in Epson mode.

Typeface options

The first and most basic options are the available typefaces. We'll start with setting the font; notice that the details (namely the maximum values allowed) differ from printer to printer. Another basic option is whether the printout should be printed proportionally or not. "Proportional printing" means that the horizontal space occupied by characters varies according to their width: for instance, an "i" would be much narrower than a "w". While most font options allow you to choose from several values, proportional printing can either be turned on or off. The commands are as follows:

select typeface: 27 107 x $1b $6b x
where x is the number of a typeface, which may vary between printers. Just try different values starting from 0.
These are the values on the Canon BJC 4200:

x=0 Roman
x=1 Sans Serif
x=2 Courier
x=3 Prestige
x=4 Script

enable proportional printing:
27 112 1 $1b $70 $01
disable proportional printing:
27 112 0 $1b $70 $00

There's another aspect to the choice of typeface, namely the choice of country. Since different languages have different sets of specific characters (like umlauts in German, and accents in French), the Epson font sets are ordered after countries. The character codes 35, 36, 64, 91-96 and 123-126, as well as 163, 164, 192, 219-224, change their meaning according to the country setting. The country code is set using the following command:

set country code: 27 82 x $1b $52 x
x is the country code

x=0 USA
x=1 France
x=2 Germany
x=3 Great Britain
x=4 Denmark I
x=5 Sweden
x=6 Italy
x=7 Spain I
x=8 Japan
x=9 Norway
x=10 Denmark II
x=11 Spain II
x=12 Latin America
x=13 Korea
x=64 Legal

Additionally, there's an italics and a graphics font: the latter has a set of graphics symbols stored in its second half, while the italics font contains a set of italic letters instead.

italics font: 27 116 0	$1b $74 $00
graphics font: 27 116 1	$1b $74 $01

Next subject is selecting the character pitch, which can be set to 10, 12 and 15 cpi (characters per inch). One of the three pitch settings is always active, so this function can't be disabled. Once you set the pitch, it's used until you change it. The sequences for this are as follows:

10 cpi: 27 80	$1b $50
12 cpi: 27 77	$1b $4d
15 cpi: 27 103	$1b $67

Additionally, there are modes for condensed (reduced character width), and expanded printing (double width). Both modes can be selected and de-selected independently. The resulting character size still depends on the character pitch selected. If condensed and expanded printing are on at the same time, they cancel each other, until one of them is turned off. Other print modes are superscripting, subscripting and expanding to double height. All these modes, once activated, have to be disabled explicitly, which is why each of them has separate enable and disable commands.

condensed print on: (27) 15	($1b) $0f
condensed printing off: 18	$12
double width on: 27 87 1	$1b $57 $01
double width off: 27 87 0	$1b $57 $00
double height on: 27 119 1	$1b $77 $01
double height off: 27 119 0	$1b $77 $01
superscript on: 27 83 0	$1b $53 $00
subscript on: 27 83 1	$1b $53 $01

superscript/subscript off: 27 84$1b $54

Character styles

The sequences in the last paragraph were a little more than basic. In fact, we could argue some of them belong to the character styles we're going to describe now. One thing common to the character styles is that they don't affect the size of the characters; rather, they influence the looks of the individual characters. For instance, a character can be printed bold or italic. Also, they can be printed as shadows, outlines or both. You can reach these modes using the following sequences:

bold print on: 27 69	$1b $45
bold print off: 27 70	$1b $46
italics on: 27 52	$1b $34
italics off: 27 53	$1b $35
outline characters on: 27 113 1	$1b $71 $01
shadow chars on: 27 113 2	$1b $71 $02
outlining+shadows: 27 113 3	$1b $71 $03
outline/shadow off: 27 113 0	$1b $71 $00

There's another character style, namely underlining, which I'm going to explain separately, as there are several options to it. Besides simply underlining text, you can underline it with a single or double line which can be either solid or broken. With the same command, you can also cancel or overline text, with the same line options available. Although it's one and the same command using different parameters (see below), it's actually three separate options. This means you can underline and cancel a text at the same time, even subject it to all three kinds of marking simultaneously, and if you do, you have to enable and disable each kind of marking separately. However, if you activate one marking several times, only the most recent selection counts, which means if you enable double underlining while single underlining is on, there will be two lines under your text, not three! The command looks like this:

```
27 40 45 3 0 1 x y     $1b $28 $2d $03 $00 $01 x y
```

x selects the position of the line(s):
x=1 to underline,
x=2 to cancel,
x=3 to overline the text.

y sets the line style or disables the chosen option:
y=0 disable the option
y=1 solid single line
y=2 solid double line
y=5 broken single line
y=6 broken double line

Shortcuts

Since the last command was a particularly long one, you'll be glad to hear that there are shorter versions of some commands. One of the abbreviations is for underlining. Underlining with the last chosen line style can be turned on or off with a simple command:

underlining on: 27 45 1	$1b $2d $01
underlining off: 27 45 0	$1b $2d $00

For some of the other functions you can use the master command. It consists of the command and one parameter, whose single bits are used to enable or disable several options, therefore it's a quick way to change several settings at once:

master command: 7 33 x	$1b $21 x

the bits of x have the following meanings:

bit	value	function
0	1	if set (1), sets character pitch to 12 cpi, if clear (0), sets character pitch to 10 cpi
1	2	turns proportional printing on (1) or off (0)
2	4	turns condensed printing on (1) or off (0)
3	8	turns bold printing on (1) or off (0)
4	16	turns double impact on (1) or off (0)
5	32	turns double width on (1) or off (0)
6	64	turns italics on (1) or off (0)
7	128	turns underlining on (1) or off (0)

According to the table, 27 33 11 would result in printing 12 cpi, proportional bold characters, and 27 33 194 would activate proportional, underlined italics text in 10 cpi. Cleared bits don't mean that the respective function isn't activated, rather it's explicitly disabled!

Colors

Another option concerning not only the text mode, but the graphics mode as well, is the option to change to a different color - provided that your printer is equipped for it. You'll notice the choice of colors is much smaller than on our C64. In text mode, you'll likely to be limited to the colors given, since mixing two colors hardly works. In the high-res graphics mode, however, you can get nice color compositions by using
alternating bit patterns. Here's the command to select the color:

set color: 27 114 x	$1b $72 x

x selects the color: x=0 for black x=1 for magenta x=2 for cyan x=3 for blue x=4 for yellow x=5 for red x=6 for green x=7 is not defined, and besides, only the 3 least significant bits are taken into account, so x=10 will result in a cyan printout!

Also, the table shows that you can only use one color at a time, and have to change it with this command whenever you need another one.

The graphics mode

Naturally, this is the most powerful option of them all. After all, it allows you to print pictures, and mix all available colors. The basic command has three parameters, one to select the desired resolution, and two others that represent the number of graphics columns to be printed, in the low-high byte format. After sending this command, you have to send the appropriate number of graphics data to fill the number of columns you specified. In the graphics data, any byte may occur, even those that equal an ESC command, and that's why there's no command to interrupt this mode by software! The command syntax is like this:

enable graphics mode: 27 42 x l h$1b $2a x l
h
different values of x result in different resolutions:

x	dpi vertical	dpi horizontal	bits image mode
0	60	60	8-bit single-density bit image
1	60	120	8-bit double-density bit image
2	60	120	8-bit high-speed double-density bit image
3	60	240	8-bit quadruple-density bit image
4	60	80	8-bit CRT graphics I
6	60	90	8-bit CRT graphics II
32	180	60	24-bit single density bit image
33	180	120	24-bit double-density bit image
38	180	90	24-bit CRT graphics II
39	180	180	24-bit triple-density bit image
40	180	360	24-bit sextuple-density bit image
71	360	180	48-bit triple-density bit image
72	360	360	48-bit sextuple-density bit image
73	360	360	48-bit sextuple-density bit image

l is the low byte, h the high byte representing the width of the graphics area in columns.

For programmers, the resolutions of 180x180 dpi (dots per inch) and up are likely the most interesting ones. As we said above, once the graphics mode has been enabled by the command, it stays active until either all columns have been printed, or the printer is turned off. The description of each mode tells us the height of a line in pixels: the number of bits in the table above is the number of bits in a column, which means that by selecting a vertical resolution of 180 dpi, we get pixel columns each consisting of 24 bits, or 3 bytes in a vertical row. If the printer was sent the command 27 42 39 16 0, it would interpret the next 48 bytes (16 columns x 3 bytes) as graphics data, and use them to print 16 columns at a resolution of 180x180 dpi. The bytes are ordered from top to bottom, meaning the first byte describes the upper 8 pixels, the second the 8 middle ones, and the third the lower 8 pixels. Bit #7 of each byte will be printed at the top, and bit #0 at the bottom. Here is the exact bit order at a vertical resolution of 180 dpi:

byte	bit	value
1	7	128
	6	64
	5	32
	4	16
	3	8
	2	4
	1	2
	0	1
2	7	128
	6	64
	5	32
	4	16
	3	8
	2	4
	1	2
	0	1
3	7	128
	6	64
	5	32
	4	16
	3	8
	2	4
	1	2
	0	1

At 360 dpi vertical resolution, the printer expects 6 bytes per column; doesn't sound like much, but remember that either 180 or even 360 of these columns are printed for an inch. That makes a whole 1080 or 2160 bytes for an inch!

Naturally, almost all of the aforementioned options are irrelevant while the graphics mode is active; the selected color should be the only exception. Besides just using the standard colors, it's also possible to mix several colors. The trick is to fill only a part of the pixels in an area with one color, then do a carriage return, change the color and fill another part of the area... at 360x360 dpi it will look like a unicolored area.

The following thoughts might be useful in case you want to use the graphics mode for drawings: The "address" of a pixel x,y (where x and y are the offset from the upper left corner) can be calculated like this:

```
line = y/48
byte = y-INT(y/48)*48+6*x
bit value = 2^(7-(yAND7))
```

These formulas hold for 360 dpi vertically. At 180 dpi, you've got to change 48 to 24 and 6 to 3. Also, if you work with this mode you have to consider that the horizontal and vertical resolution can be different, which you'll have to compensate for (otherwise, circles might end up looking like ellipses, and squares like rectangles).

Other commands

The following commands have more general effects on the text and graphics modes. First, there are commands that move the print head. They are mainly the carriage return (CR), line feed (LF), backspace (BS), and the horizontal tabulators. Although the carriage return uses the same code as the RETURN key on the 64 (whose name is derived from carriage return), sending a carriage return will only return the print head to the start of the current line! In order to get the same effect as pressing the RETURN key, you have to send an additional line feed, which will move the paper ahead one line. Backspace equals pressing cursor left, i.e. the print head will be moved one character to the left, and HT causes it to jump to the next horizontal tabulator position.

```
CR: 13 $0d
LF: 10 $0a
BS: 8 $08
HT: 9 $09
```

In order to make a line feed possible at all, the height of a line must be specified. There are several commands to do this which differ in the unit of the height. They all expect a single parameter which, divided by the appropriate factor, denotes the height of a line in inches. If a command expected the height in x/360 and a line were 48/360 (or 2/15), then x would be set to 48. If x was 60, this would mean 60/360 or 1/6th of an inch.

line spacing x/360:	27 43 x $1b $2b x
line spacing x/180:	27 51 x $1b $33 x
line spacing x/60:	27 65 x $1b $41 x

Then, there are shortcut commands to set the line spacing to 1/8th or 1/6th of an inch:

line spacing 1/8:	27 48 $1b $30
line spacing 1/6:	27 50 $1b $32

If you want to use the tabulators, you'll have to set them first, of course. This is done with the following command, which allows you to specify up to 32 parameters; each parameter is the position of a horizontal tabulator. A "0" parameter signals the end of the list.

set horizontal tabulators:
27 68 list $1b $44 list

But there are also commands to set the margins and the number of lines on a text page:

left margin:	27 108 x	$1b $6c x
right margin:	27 81 x	$1b $51 x
page length:	27 67 x	$1b $43 x

Now, only a few commands are still missing, namely form feed (FF), the commands to turn the printer online or offline, and the one to sound the printer bell. Form feed simply ejects the current sheet and feeds a new one into the printer. This command should be sent after a printout; otherwise, the user has to end printing and eject the sheet via the control buttons on the printer. Changing between online and offline should be self-explanatory; it's recommendable to turn the printer online before starting a printout, just to make sure. Finally, it's up to you what use you have for the printer bell:

form feed:	12 $0c
ONLINE:	17 $11
OFFLINE:	19 $13

This article was written using the printer manuals of the Star NL-10 and the Canon BJC-4000.

The following programs you'll find on the cover disk:

-"BASICLADER"

This BASIC program generates the machine language program described below on a medium of your choice. After running, it asks you for the drive number and the name for the target file. Line 500 contains the address of the machine program in the low/high byte format. Since it contains relative jumps only, the routine is freely movable just by changing line 500! It does, however, access some other memory locations which can only be changed in the code itself. The BASIC loader contains a simple source code to the routine, so it should be simple to adapt it to your needs.

-"PRINTERDRV"

This is the machine routine generated by BASICLADER. This version is located at address 834 ($0342); the other BASIC programs always assume the routine to be at this address. If you need the tape buffer for anything else, you can simply use BASICLADER to generate a version that's appropriate for you. The routine sends data from memory via the user port. Before calling it, set 828/829 ($033c/$033d) to the address of the first data byte, and store the number of data bytes in 830/831 ($033e/$033f). 832 and 833 ($0340 and $0341) are used to buffer the data direction registers, and 167/168 ($a7/$a8) hold a pointer. This routine may and should be passed on and used in your own programs. You may also alter it for this purpose.

-"TRANSFER"

This program assumes PRINTERDRV to be in memory at address $0342. The lines 100 and on hold data in DATA statements, which are sent to a printer connected to the user port. In the DATA statements, ASCII codes can be included as numbers. Strings have to be preceded by a -2, and the end must be marked with -1. A very short entry could look like this: 1000 DATA 13,10,-2,"Hi folks!",-1 This line will cause the program to send a CR, an LF and the text. The program performs a rudimentary conversion from PETSCII to ASCII. The table used (in lines 500 through 650) is kept very simple, and works for normal capital and small letters from a to z, punctuation marks, and digits. Umlauts aren't implemented.

Therefore, you might like to modify the table for your own projects. This shouldn't pose a problem. The table is ordered according to the PETSCII codes from 0 to 255, and every DATA statement contains 16 codes (those from 0 to 15 are in line 500, 16 to 31 in 510, etc.) By the way, the variable BA contains the address where the program puts the data to be transmitted; it defaults to 49152, but it can easily be adapted by changing the value of BA in line 100. If the machine routine isn't located at 834, the SYS command in line 200 has to be adapted. The other sample programs on the disk are all constructed on the basis of this routine, and thus have the same restrictions and options.

(ad)

Hardware - Forum

The Printer Looking for Connection...

... and this is realized in the simplest case by a printer cable. In the past, there had been printers with serial connections, which were connected via a serial cable with the system like a disk drive. Such printers (for instance Commodore MPS1230) haven't been sold anymore for a long time now, today's models all have a centronics port (36 pole).

by Niko Malecki

Our printer cable connects the C64/128's user port with the printer's centronics port. Such a cable isn't common anymore, and thus has to be made by ourselves. It is important that we are using a shielded round cable, because they are the only ones that allow for the greatest length of up to 2 meters (about 6.5 feet). I explicitly advice you not to use ribbon cables of any kind. Such cables are highly susceptible to exterior influences due to a lack of shielding. As components we therefore need an appropriate piece of cable, which needs to have 11 lines. Commonly the best match for sale is a version with 10 lines, the eleventh being ground (GND). If you used, for instance, a 14 line cable, then you ought to ground the lines not used. In addition we need a user port plug and a 36 pole centronics plug for the printer's side.

The connections are as follows:

Userport-Pin	Centronics-Pin
A	16
B	11
C	2
D	3
E	4
F	5
H	6
J	7
K	8
L	9
M	1

This setup is the same for all printers, no matter whether it's a dot matrix printer or an ink jet. From time to time you may come across some models (usually older ones), which need a connection from user port B to printer 10 instead of 11. In this case, some experimenting should help. Under no circumstances should pins 10 and 11 be simply short circuited. When you see a printer cable in any pc shop, then they are referring to a pc cable and not one for c64/c128. Such a cable can still be used, instructions for such a re-fitting have been devised by Colin J. Thomson for us (they can be found elsewhere in this issue).

An Interface Needs some Programs

There are a lot of programs out there, which can only print by means of a serial cable. In such cases, we need a printer interface (a "transformer") from serial to centronics. These interfaces, the most famous ones being Xetec in America and the Wiesemann Interface in Germany, are only found on used items sales.

Specialty by CMD

Let's not forget about the GEOCable by CMD. While the normal printer cable needs the user port all for itself and thereby blocks it, the GEOCable still allows for further usage of the user port. It hooks up the printer as an addition , so that the user port itself remains free. This is quite handy since there is equipment which is attached to the user port, for instance eprommers, IC tester, Handyscanner, and others. These devices do not always agree with the printer and with the GEOCable, the printer is simply switched off. By the way, GEOCable uses a commonly sold pc printer
cable (comes along with the GEOCable).

The GEOCable, just like any other CMD product, can be bought from the CSW publishing house (GO64!).

Ah, those Unfortunate switches ...

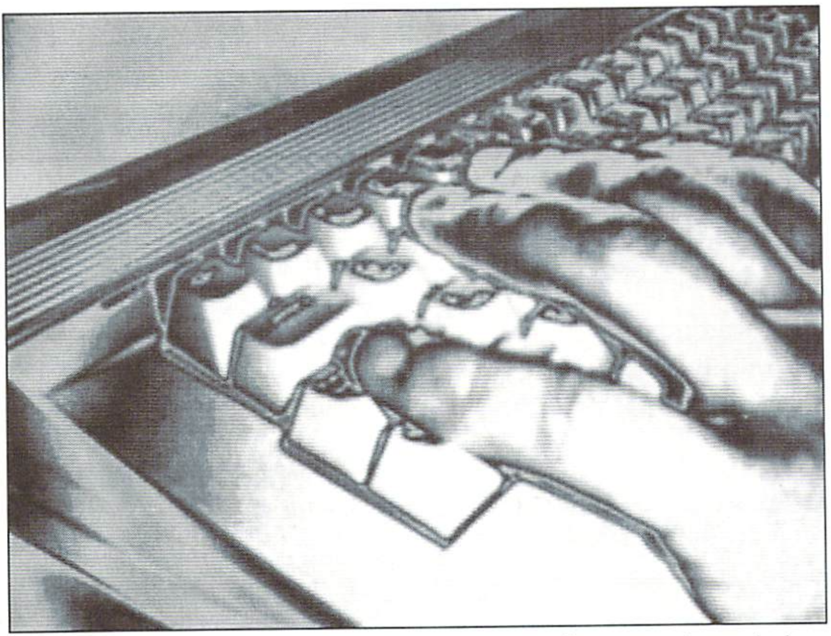
There used to be user port switches as well, for instance made by REX or Scanntronik. In most cases, those switches caused more problems than they cured.

Switch Boxes Are Great

And since we are already talking about switching off, on, and between, there is another fine solution if two or more printers are around. Just as life goes, first you have a dot matrix printer, and then at some point, you add a fine quality ink jet. The result: Remove the cable from one printer and plug it into the other. This is quite dangerous, because if it's done with the devices turned on, the computer's CIA chip is as good as done for. In addition, this maneuver is quite straining for the plug contacts. This constant plug switching can be evaded by using a commonly sold switch box. These switch boxes come in many variations. One for 36-pin centronics plugs is the one for our needs. There are boxes that can be switched 2- , 3- , or 4- way, that is there is one Input/Output (leading to the user port as it were) and 2, 3 or 4 ports for printers.

When using this solution, we have to remember that, when having two printers for instance, we would also need two additional cables connecting the printers and the switch box. These cable need to have at least all required lines, that's 1- 16, (s. connection scheme). We also need to check, if the aforementioned lines are switched within the switch box itself, since the switch box has enough lines, but the connection scheme is a bit different for a printer connected to a pc. If things go fine, we get a switch box which switches all 36 lines (full allocation). This case always works, there is nothing we need to change.

Special consideration should be given to the cables between switch box and the printers, since the total cable length between user port



Picture: Sebastian Bach

and printer should not exceed 2 meters (6.5 feet). If necessary, we should shorten the cable as required to prevent damaging the computer. So much for connecting cables.

Now Things Get Comfortable ...

Under adverse circumstances considering space, we get easily into a situation where a cable length of up to 2 meters (6.5 feet)just won't do. But there are good solutions for that problem. For one thing, there is the so-called printer booster, that is kind of an amplifier , which is "forcing" the data and therefore relieving the user port. With a printer booster cable lengths of up to about six meters (20 feet) can be handled. There had been building instructions for them in earlier 64'er issues and books, but we can't reprint those projects due to copyright reasons. Such a booster can be received via trade channels, though. Firma Reichelt (Sande, Germany), for instance, offers a "Line-Booster COM 621", which fulfills our requirements exactly, for about 40 to 50 DM?

The complete cable would then start out with our standard cable user port - printer, with the printer end going into the booster. Leaving the booster, a cable of appropriate length then connects to the printer. The booster has its own power supply, so that the amplifier energy demands are certainly met.

And now very Elegant

A much more elegant solution is transferring the printer signals by infra red sender and receiver. Until recently, such devices had been sold by Conrad Electronic (www.conrad.de) under the name "InfraLink". If needed, please ask Conrad. For transfer we need two devices, each of them being both sender and receiver (price about 30 DM??? per device). Our standard printer cable is simply connected to the sender, with the other InfraLink instrument standing on the receiving side which may very well be eight or more meters (26 feet or more) away. This is then connected to a fully connected 36 line centronics cable leading to the printer.

This set ups great advantage is bridging great distances without a cable.

The Crowbar

I will now interrupt my discussion in order to give the printer a chance to have a break from its job. I am talking here about a printer reset. When printing files there are times when we would like to interrupt the printing process. Very often, canceling is offered by the printer program's menu, but by then the printer has already accumulated lots of data into its own internal memory and keeps on printing in spite of the abortion. If we want to stop the printing process immediately, then there is nothing but turning off the printer, or ... yes, or adding a reset button. Every printer has a reset signal on a certain pin of the centronics port (see printer manual). If that pin is shorted with ground (GND), for instance by a button, then the printer is reset, that is it stops immediately.

A Buffer Is Needed

When printing major text files, like for instance with the printer program "GEOS_LQ", or major PAINT documents, we will notice that the printer needs quite some time to process these data. During the entire transferal from computer to printer, the computer is blocked, that is we have to wait until the printer is (finally) finished. And this is just where a printer buffer gets convenient. Such a buffer is inserted between computer and printer and has the job of receiving the print data into its own memory, releasing the computer, and only then feeding the printer the data. In the course of time there had been many different kinds of buffers on the market, but it has shown that especially when printing with GEOS, buffers start to make sense starting at capacities of 1MB. A very good specimen is the printer buffer from the electronics magazine "Elektor" issue 11/91 (www.heise.de). This buffer can mount either a 1 or 4 MB SIMM memory chip, which should be sufficient for fast processing printer data in all situations. At the same time, a three digit digital display informs about the file size to be handled and about how much remains to be printed.

Older printer buffers with 64 to 256 KB of memory, on the other hand, are only advisable for minor amounts of data, for instance for old programs like Startexter, Protext, and the like.

Specialty

And now something special. Those using cartridges have an option called "printer menu" or the like depending on the model used. This menu allows for a more or less comfortable screen hard copy. The by far best printer option can be found in "Final Cartridge III". It can access just about any printer types, and even colored printing is possible. At the same time, the cartridge offers three interface possibilities: CBM serial, centronics, and RS232, so that just about any printer has its appropriate printing routine. More details on further printing options like inverse printing can be found in its manual.

So much for my visit to the world of printing today. There is a lot that can be said about printing software as well, but you will read about that another time.

Yours
Niko

Suppliers:

Reichelt, Sande Tel.: (+49) (0)4422-955333
Conrad, Hirschau Tel.: (+49) (0)180-5312111

A PC Printer cable on your Commodore?

I thought I would share this little PCB (printed circuit board) that I have used since about 1996. I was fed up with the "ugly" ribbon cable that I used for my "geoCable" (Centronics to user port) so I decided to make this little PCB/adapter.

by Colin J. Thomson

It's not as versatile as the CMD geoCable but it does the job!

Parts required are :

1 - 2 * 12 conductor Female Edge Connector
(.156" pin-to-pin spacing)
2 - 1 DB25 (Right Angled Socket)
1 - small single sided PCB approx 2 inches wide * 1 inch deep.
1 - Standard PC Printer Cable.

NOTE: In your printer cablre, make sure all of the pins connect through end-to-end. Some low-cost PC printer cables only have a few pins connected.

The connections...

Userport	A	B	C	D	E	F	H	J	K	L	M
DB25	19	11	2	3	4	5	6	7	8	9	1

There is no pin G or I on the user port connector, and although pin N is marked on the board layout, it is not used.

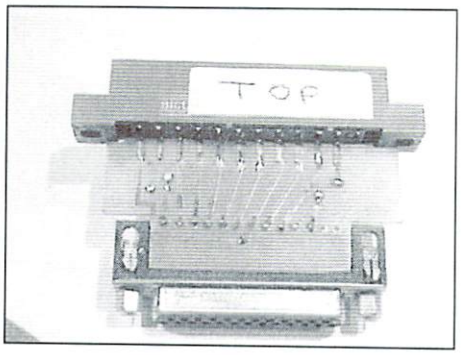
PLEASE NOTE. On some old Dot Matrix printers the "signal ground" is expected on pin 16. So I would advise that you check your printer manual and change the layout to suit.

I have tested this with a Citizen 120D+ and a Canon BJ10 inkjet with no problems using pin 19 as the ground.

Construction.

As you can see in Figure 1, the user port connector sits on top of the board and the DB25 right angle socket sits underneath. It was done this way for the easiest trace layout of the board. (Please remember the top of the board has the traces showing, with the two links and the DB25 socket underneath..) Cut the board to size and clean it, then mark out the points where to drill the holes for the DB25 socket, (on the NON copper side of the PCB). You can make this job easier by removing the unused pins, BUT BE VERY CAREFUL as you must get Pins 16 and 19 right!! This will leave fewer holes to mark out and drill. Once done you can drill the holes with a 1 mm to 1.2 mm drill depending on the diameter of the pins on the DB25 socket .

At the other end of the board goes the user port connector. Turn the board over so the Copper side is facing you. Then mark the points on the board where the user port connector goes. This is done by laying the pins on the board and marking them with an Etch Resist pen . The pads can be made thicker later on.

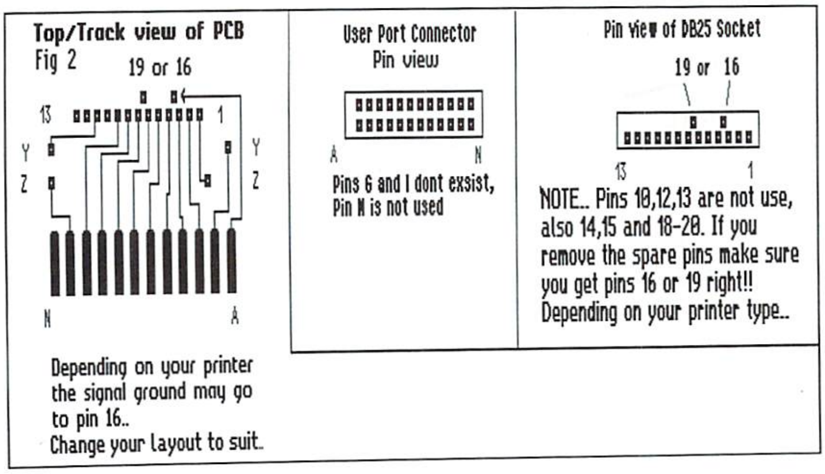


NOTE:
On the user port socket the bottom row of pins are the ones used (A to N), so I removed the top row. I would recommend that you mark the connector with "top" so we know which way round it plugs in just to be sure.

Now all being well, follow the layout for joining up the two sets of contacts (see Fig. 2) you can use either PCB transfers, or if you have a steady hand, draw the traces using an Etch Resist pen. You can make the Pads thicker for the sockets, so its easier to solder them onto the board. Alternatively you could scale down and print the layout on some Film and use "UV" board technique described in GO64! 9/1999. Once you are happy with the board you can etch it.

Once the board is etched and cleaned you are ready to fit the sockets and the two "links" to the PCB. (A small soldering iron is recommended.) I fitted the DB25 socket first, which sits underneath the board, then I turned the board over so the traces are facing me. Align the bottom row of pins (on the user port connector) up to the Pads and solder them. The last job is to fit the two links Z to Z and Y to Y with some thin gauge wire, these are fitted under the board.

And that's it! You could if you wish varnish the top of the board to hide the traces and prevent them from tarnishing, but as it is so small not much copper is exposed anyway. *(ma)*


Top/Track view of PCB
Fig 2

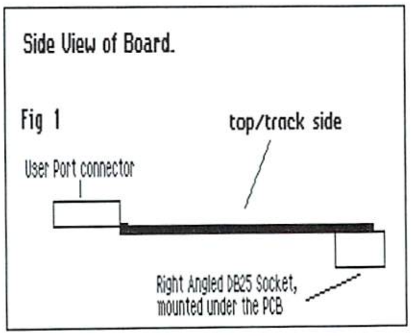
Depending on your printer the signal ground may go to pin 16..
Change your layout to suit.

User Port Connector
Pin view

Pins G and I dont exsist,
Pin N is not used

Pin View of DB25 Socket

NOTE.. Pins 10,12,13 are not use, also 14,15 and 18-20. If you remove the spare pins make sure you get pins 16 or 19 right!! Depending on your printer type..


Side View of Board.
Fig 1

Cover Disk

New...

You may remember a program called "MacBootMake", included on the July 1999 cover disk. Unfortunately, the file was corrupt, so we've included a working copy this month.

by Marco "Mac Bacon" Baye

Back then, the program was already corrupted by the time it was downloaded: Due to the combination of a badly configured webserver and an overzealous Windows/ Netscape combo, all $0a-Bytes through the sequence $0d $0a were exchanged (a conversion from "LF" to MSDOS type "CRLF"), something which quite effectively destroyed the BASIC program. The test C=128 used by the editor was defective, and thus the error was not detected in time.

Because the program used German screen text and was therefore unusable with the international version of GO64! (introduced with issue 8/99), it was not published anew.

Until now, that is. In addition to the new english-language version of the program, some new hot-keys have been designed. Nothing has changed in the user interface since version 3.10.

The How To's

The program "MacBootMake" is used to quickly and easily supply bootsectors for the 1540, 1541, 1570, 1571, 1581 and similar disk drives. The utility is written in pure BASIC V7, and works in both 40 and 80 track modes (the modes will be used accordingly, for example with the Directory function).

After you load and start the program,

```
run "mbm323"
```

the main menu appears. The area above deals with configuration, the bottom region with the actual work.

Adjusting the settings...

The first two points show which disk drive is active:

<+> and <-> change the device address in the range of 4 to 30. All corresponding functions of the program will use these addresses.

<1> changes the device type, either "1541/71" or "1581". CAUTION: If you make an error with this setting you can lose all of the data on your disk!

The next four lines determine the appearance of the Bootscreen:

<2> The local character set can be forced from here (only makes sense with non US machines).

<3> Determines if the system message "BOOTING" appears or not.

<4> Determines whether or not the key combo CBM + Shift will be active or not.

<5> Forces "small letter" character set.

The last three settings determine the parameters of the program which starts at boot time:

<6> Accepts the name of the file. Next, an alternative device address will be asked for. If such an address is entered, it will be saved in the boot block, so that the program can be run from another disk drive. Since this relatively esoteric functionality would only seldom be used, you would normally hit return at this request.

<7> Sets the boot program type: BASIC or Machine Language (this is done so that the correct calling command is used).

<8> Sets the memory bank which will be chosen before the boot program is started - this setting is only relevant if you've selected Machine Language in number <7>.

Functions

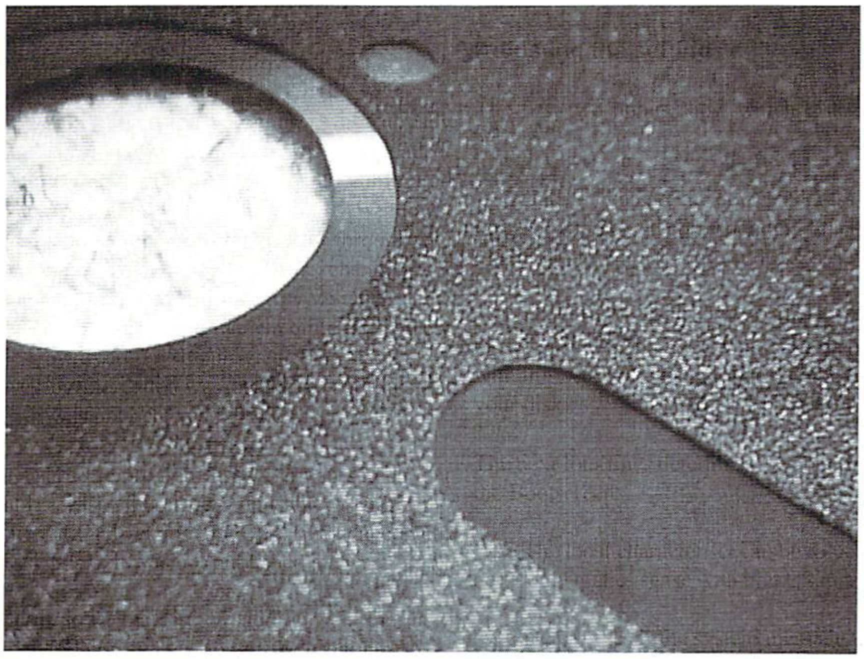
<i> displays information about the program (e.g. the Version)
<e> enter a boot greeting
<t> tests the boot greeting
<s> deals with the actual work: this function saves a boot block with the new words, boot greeting, etc... to the chosen disk drive.
<r> removes a boot sector from the chosen disk drive.
<c> sends a command to the disk drive (see the disk drive manual).
<$> shows current contents.
<q> quits the program.

That's it!

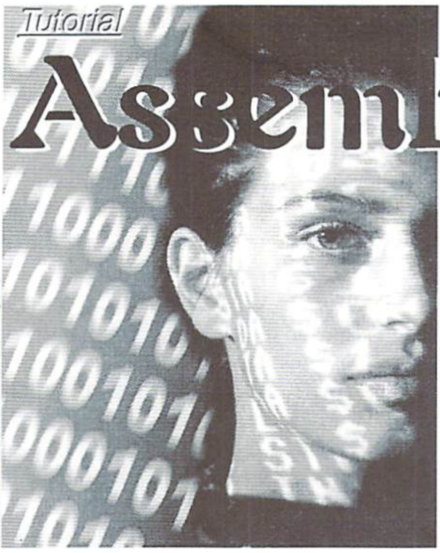
What this program *can't* do

The newly produced boot block will be shown in the BAM as being occupied, but would be removed by the command "v0" (COLLECT on the 128) due to the missing sector chaining. The disks should not be validated. With the 1581, there is a way to disable this (you have to create a partition which exits only on sector 0, track 1), but to get that sector back, the program must scan the directory of this "one block" partition. Such a scan is not possible.

Happy coding ! *(ws)*



Tutorial



Picture: Bernhard Peball/Günther Bauer (www.lichtbogen.at)

Assembly Language

After having dealt with sprites last time, I would like to discuss programming text input and output now, followed by the routines for loading and saving in the next issue. This way, those of you programming applications should also be able to benefit.

by Wanja Gayk

Part 12

What we really need to know are some specific ROM routines. Right from the start, the C64 offers quite a few useful routines that can make things easier for us. So why re-invent the wheel? The basics are offered by a routine at $FFE4 - CharGet (GET) - and a routine at $FFD2 - CharOut (BSOUT). You have to be aware of the fact that we are doing most of the following things using ASCII codes. Screen codes are rather unimportant when dealing with texts:

$FFE4 - GET
This routine checks for a key being pressed and returns its ASCII code in the accumulator. If no key is pressed, $00 is returned.

$FFD2 - BSOUT
This routine either prints an ASCII character at the current cursor position using the current color or executes a control code such as RETURN or CLEAR SCREEN.

Effectively, this little program is nearly a very simple text editor. Its disadvantages are that you don't see the cursor and that the text is saved nowhere else but the video RAM:

```
key   JSR $FFE4
      BEQ key
      JSR $FFD2
      JMP key
```

But this can't be the peak of the art, there is no cursor to be seen. This is why it is of crucial importance to know the cursor's current position. There are several methods to find out: To begin with, there are two bytes in the zero page containing the current cursor line and column: $D3 keeps track of the cursor column and $D6 of its line. It would be of much greater use, if we were to know, where the cursor was in the video RAM, though! For this purpose, there is a 16 bit address in $D1 and $D2. It contains the location where the line the cursor resides in starts. In the following examples I work on the basis that the video RAM is located at $0400, just like when the computer has just been started. If the cursor is in the upper left corner (after you have pressed {HOME}), $D1 returns $00 and $D2 $04. Thus the line starts at $0400. $D3 contains the column, so that the exact address can be determined by adding the value from there to the address in $D1/$D2. The simplest way to do this is to use indirect y indexed zero page addressing. Creating a cursor means in the most basic case that I replace the character it is on by some character representing the cursor. In addition, I save the original character somewhere, so that it can be restored when the cursor is moved away. The last thing to consider is that routine $FFE4 messes up the contents of A, X, and Y - but this is only a small problem. Here now is a routine with a cursor:

```
loop LDY $D3          ; Cursor column
                      into Y
     LDA ($D1),Y      ; Store screen code
     STA $02          ; beneath cursor
     LDA #$FF         ; Cursor character
                      (screen code)
     STA ($D1),Y      ; To cursor position
get  JSR $FFE4        ; GET routine
     BEQ get          ; wait for input
     PHA              ; Save input onto
                      stack
     LDA $02          ; retrieve former
                      character beneath
                      cursor
     LDY $D3          ; (Set column again)
     STA ($D1),Y      ; To old position
     PLA              ; retrieve input
                      from stack
     JSR $FFD2        ; And print using
                      BSOUT
     JMP loop         ; Back to the
                      beginning
```

It's a pity that you can't see the char beneath the cursor when using this routine. A blinking cursor can be created, but this is actually a bit exaggerated for such a simple task. It would indeed suffice to simply invert the character beneath the cursor. You can quite easily invert characters since the video RAM uses screen codes (whereas BSOUT and GET work with ASCII). You only need to invert bit 7 of the screen code beneath the cursor. For inverting bits we shall now meet a new command which is related to the logical operations AND and OR:

EOR - Exclusive OR
EOR always works on the value in the accumulator. If you modify a value in the accumulator by means of EOR, then those bits are inverted which are set in the EOR command's argument. The result can then be found in the accumulator. For example: The accumulator contains #$01. Combining it with #$80 by EOR leaves #$81 in the accumulator. Combining that #$01 with #$81 by EOR results in #$80 being stored in the accumulator. See also the diagram.

Thus, we simply insert an EOR operation instead of a new value for the cursor character. EOR #$80 (binary: %10000000) simply reverses bit 7, so that we can easily replace every screen code by its inverted counterpart. Here now the source code:

```
loop LDY $D3          ; Cursor column
                      into Y
     LDA ($D1),Y      ; Preserve character
     STA $02          ; beneath cursor
     EOR #$80         ; Invert character
     STA ($D1),Y      ; At cursor position
get  JSR $FFE4        ; GET routine
     BEQ get          ; wait for input
     PHA              ; Save input onto
                      stack
     LDA $02          ; Get old character
                      beneath cursor
     LDY $D3          ; (Fetch column
                      again)
     STA ($D1),Y      ; And put to old
                      position.
     PLA              ; Retrieve input
                      from stack
     JSR $FFD2        ; And print it via
                      BSOUT
     JMP loop         ; Back to the
                      beginning
```

Which method you use in the end - inverting the character or a distinct cursor character -

EOR - Exclusive OR

```
1 EOR 1 = 0
1 EOR 0 = 1
0 EOR 1 = 1
0 EOR 0 = 0

    01110000 ($70)
EOR 00101011 ($53)
  = 01011011 ($5B)
```

is a matter of personal taste, of course. I only wanted to use this function to illustrate the EOR command.

String Output in Assembly Language

A simple loop and a table with the letters' ASCII codes in the memory is all it takes to program a normal PRINT command as found in BASIC in assembly language. Such a table can be entered in plain languag in just about every machine language monitor (with the Action Replay Cartidge's monitor things look like this, for instance : ".i 1000 here is a text"). Assemblers make things even easier.

```
      LDX #$00
print LDA string,X
      BEQ out
      JSR $FFD2
      INX
      BNE print
out   RTS

string .TEXT "go64!assembly language
course"
      .BYTE $00
```

You can see that I have added a $00 byte to the text in order to mark its end, so that my loop knows when to stop. In addition, BSOUT at $FFD2 has the excellent characteristic of not changing the X or Y register. I can only print up to 256 characters because I am using normal X indexed addressing, but that isn't much of a problem. If I want to print more characters, I either use zero page addressing for accessing the text field, or I program the routine several times for different sub-texts.

Setting the Cursor Position

Granted, it wouldn't be nice to place the cursor only using the ASCII codes for {CRSR-UP}, {CRSR-DOWN}, {CRSR-LEFT}, {CRSR-RIGHT}, and {HOME}. The "LOCATE X,Y" command as found in the Amstrad CPC's Locomotive BASIC or the PC's QuickBASIC is missing in Commodore BASIC V2, but it is already there in the C64's ROM routines: The routine at $FFF0 (SET/GET Cursor Position) does the job. And with a few POKEs and a SYS command, it can also be used from BASIC.

$FFF0 - SET/GET Cursor Position

This routine either sets the cursor position to the values in x and y registers, or it returns the cursor position in the x and y registers. You choose the function by setting or clearing carry. If carry is set, the routine returns the cursor position, if it is cleared, the routine sets the cursor. Take Care: x and y are exchanged. The value in x gives the cursor's y-position (line) and the value in y the x- position (column)!

Setting the cursor

```
LDY #$0A        ; Column 10
LDX #$05        ; Line 5
CLC             ; Function "Set
                Cursor"
JSR $FFF0       ; ROM routine SET/
                GET Cursor
...
```

Get cursor position

```
SEC             ; Function "Get
                Cursor"
JSR $FFF0       ; ROM routine SET/
                GET Cursor
...
```

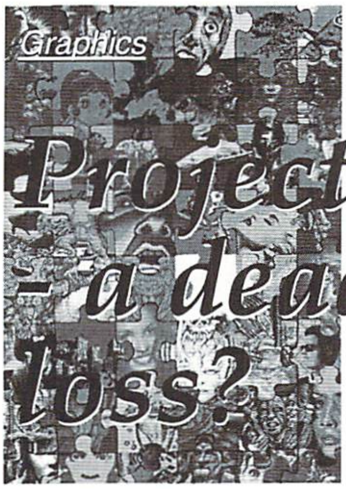
In order to clarify this example, I am going to paint a box using BSOUT and SET/GET Cursor:

```
       LDX #$03     ; Start in line 3
line   LDY #$04     ; Start in column 4
column CLC          ; Function: Set
                    Cursor
       JSR $FFF0    ; ROM routine SET/
                    GET Cursor
       LDA #$2A     ; ASCII character
                    "*"
       JSR $FFD2    ; BSOUT
       INY          ; Increase y
       CPY #$15     ; Already column 21
                    (decimal)?
       BNE column   ; If not, next
                    column
       INX          ; Increase x
       CPX #$0C     ; Already line 12
                    (decimal)?
       BNE line     ; If not, next line
       RTS
```

Well, now you know basically all important facts about handling texts in assembly language. With the routines BSOUT, GET and SET/GET Cursor, you can effectively do just about anything you like. Finally, let's look at a small but nice input routine you may for example use to enter file names for load and save routine, because this is exactly what the next part of our assembly language course will be about. Till then, have fun before your gleaming monitors...

```
; * Small input routine for entering
; * an ASCII string into 'name'.
; * Maximum length given in A.
; * Length of input returned in Y.

name = $0200

input    STA max
         LDA #$00
         STA prl

putloop  LDA #$BF      ; Cursor character
                       (ASCII)
         JSR $FFD2

reget    JSR $FFE4     ; Wait for
         BEQ reget     ; Key

         PHA
         LDA #$14      ; Clear cursor
         JSR $FFD2
         PLA

         CMP #$0D
         BEQ enter     ; RETURN key

         CMP #$14
         BEQ delete    ; DELETE key

         CMP #" "      ; Or #$20
         BCC putloop ; Test, if valid
         CMP #$80
         BCS putloop ; Character

         LDY prl       ; Check for input
         CPY max       ; Too long .
         BEQ putloop

         STA name,Y    ; Store character
         JSR $FFD2     ; And print it
         LDA #$00
         STA $D4
         INC prl
         BNE putloop

enter    LDA #$00      ; End of input
         LDY prl
         STA name,Y
         RTS

delete   LDA prl       ; Delete character
         BEQ putloop
         DEC prl
         LDA #$14
         JSR $FFD2
         JMP putloop

prl      !byte $00
max      !byte $00
```

Graphics



Project "Hopeless" - a dead loss?

Part 5

Hoogo on the "JMP $07D0" party held by Out of Order

This month, we got from Hoogo, known in the scene as coder and graphician of the group Padua, to support us. He's mostly known for his excellent multicolor painting tool, "Color-X". In this article, he'll tell us how he created his picture "Hopeless", and his own and rather unusual method to develop an image on the C64.

by Frank "Hoogo" Jürke

The picture, "Hopeless", was intended as an experiment to explore what quality a combination of hires and interlace might offer. Color blends look best if you use colors with similar brightness, the brighter and darker color constantly alternating. All those hires sprites lying above the bitmap allow for an interlaced picture with a fine checkered pattern, and only minor color restrictions at the same time. The background was done on a 64, and the girl was painted on PC. Besides the actual drawing, some technical work had to be done: selecting the good-looking color combinations for mixing; attune the colors of the PC monitor to those of the C64/TV set; obtain a good palette of skin colors (was done by converting photos); dividing the completed picture into half-frames; cutting those into sprite; writing the multiplexer, and surely some other stuff. It was published on TP97 in Denmark.

In the beginning, there was inspiration

Here's how I draw (roughly): After being hit by a sudden inspiration, I start looking for images related to the subject. During the years, I've collected lots of images from magazines and advertisements; in this case, however, some books about WWII, material about anatomy and a large mirror were much more helpful. After finishing my research, I continue with pencil sketches. I don't copy the pictures I got during research; they're only used for verification and correction. To tell the truth, I'm a rather bad artist; the first images I bring to paper look very unshapely and need lots of improvement. The sketches are converted to the 64 as very rough outlines; usually, I paint directly onto the screen, using a felt-tip pen; besides this, I've recently started experimenting with scanning, a graphics tablet, and a conversion method known as "wiring". Then, I go on doodling with a mouse, further developing the outlines (as seen in fig. 2); the surprises happening during the process are then refined into the final picture. In this stage, critics are very important to me, as others will often see some flaws in the picture that I don't notice because I've gotten used to them during the work.

Mixing, color gradients, and anti-aliasing

Large areas are a problem: they look boring if they're painted in a single color, and you can't get a smooth color gradient over lots of pixels with only 16 colors. Instead of an unicolored area with a gradient, you'll see several areas of different color; in this respect, an image is very similar to copper bars on the Amiga. Mixing colors halves the resolution which is low enough already, and usually gives the surface a dull look you don't want.

Anti-aliasing is used to prevent coarse transitions, however, it shouldn't be overdone either: instead of giving an impression of increased resolution, it might even reduce it, if you need many pixels for the transition color. I try to create small areas of irregular shape. The pixels used for the anti-aliasing effect are not only transients between two colors, but also determine the form. Also, it's possible to mix two colors without loosing resolution. If such a color mix covers only a few pixels, it might just as well be interpreted as a gradient, or a shape. In short: mixing, color gradients and anti-aliasing become one in case of small areas. For instance, take the orange-gray clouds at the horizon (fig. 1).

It's important to keep the overview in zoom mode

For me, it's essential to be able to watch the zoomed image as well as a portion of it in the original size - in zoom mode, color gradients just look like a lot of pixels and therefore are difficult to judge; on the other hand, drawing them directly into the full-screen view doesn't work precisely enough for me. For successful doodling, I need a cursor controllable at sub-pixel precision - otherwise, I'll slip too often. At those spots where color gradients make sense, I try to choose the three freely selectable colors for each card as differently as possible. There's a trick I can use to see which cards have a free color left: I use a clash mode that changes nothing if a pixel can't be plotted because there are too many different colors in its card, and a rectangle in a color not used in the picture. This way I can see all spots that can use some more color. Also, many details are oriented at the borders of cards, like the piece of wall with the blue halo and the horizontal window holes at the horizon (fig. 0)

A hard choice - the right background color

I mostly use black for the background. But the principle behind it is to use the color that appears in most cards that whose color capacity is exhausted. In the picture, there are many black details over colored areas, and the colored areas merely interrupt the black primary color. Besides, I prefer dark colors, and starting with black, I can paint a variety of color transients to colored areas.

A picture "grows"

The picture was created from 3 sketches, and it has also changed in content during the creative process. Sketch no. 1 (fig. 1) had a horizon, ground lines to indicate the vanishing point, and featured an angel that became the victim of negative criticism. The horizon developed out of some yellow doodling, which was later shaped with color gradients.

The ground lines were helpful in keeping the perspective, while I brought sketch no. 2 (fig. 2), the ruins, onto the screen using a felt-pen. There's no real principle behind the debris of the ruins - although I had an image in mind where the light should come from, where the debris would be lighted, and where the shadows would fall, but in the end I accidentally drew some good-looking parts which I re-used at other spots. It's no accident that the clumps to the very left and right of the wall look very similar (fig. 3). The restriction to two gray levels was done deliberately too, since using the third tone of gray would have resulted in an anti-aliasing effect, which would in turn make the stones look less jagged.

At first, I worked on the ruins and the horizon separately; after combining the pictures, I had to add details (like the far ruins); Also, corrections to the perspective were necessary (the right part of the wall had to be clinched). At this point, I had lots of work with Color clashes; in many cases, it was impossible to realize a color gradient as I had in mind, which meant I had to move small pieces of clouds every now and then. Sometimes, even larger moves were necessary: in order to realize the blue halo, the left part of the wall had to be moved and thickened.

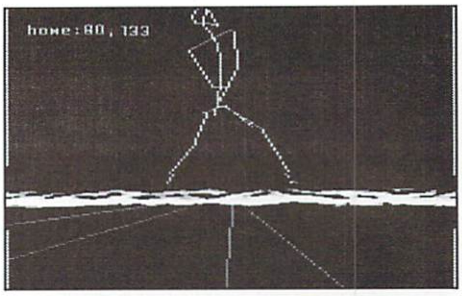
Trouble with the girls?

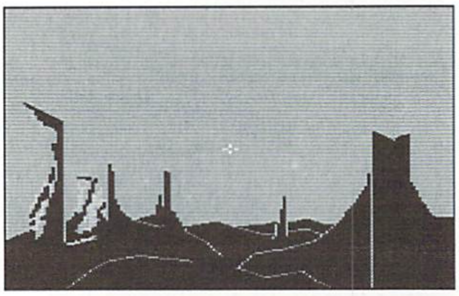
Sketch no. 3 (fig. 4-7), the girl, was first transferred to the PC monitor using transparent paper, then resized. The first tries to fill the outlines with color looked absolutely terrible. At this point, I tried to find good skin tones, and their distribution. The more I filled the sketch with color, the better I could check and correct the proportions. I didn't have to watch restrictions caused by clashes, but the figure became more slender, to enable later conversion to sprites.

Finished!

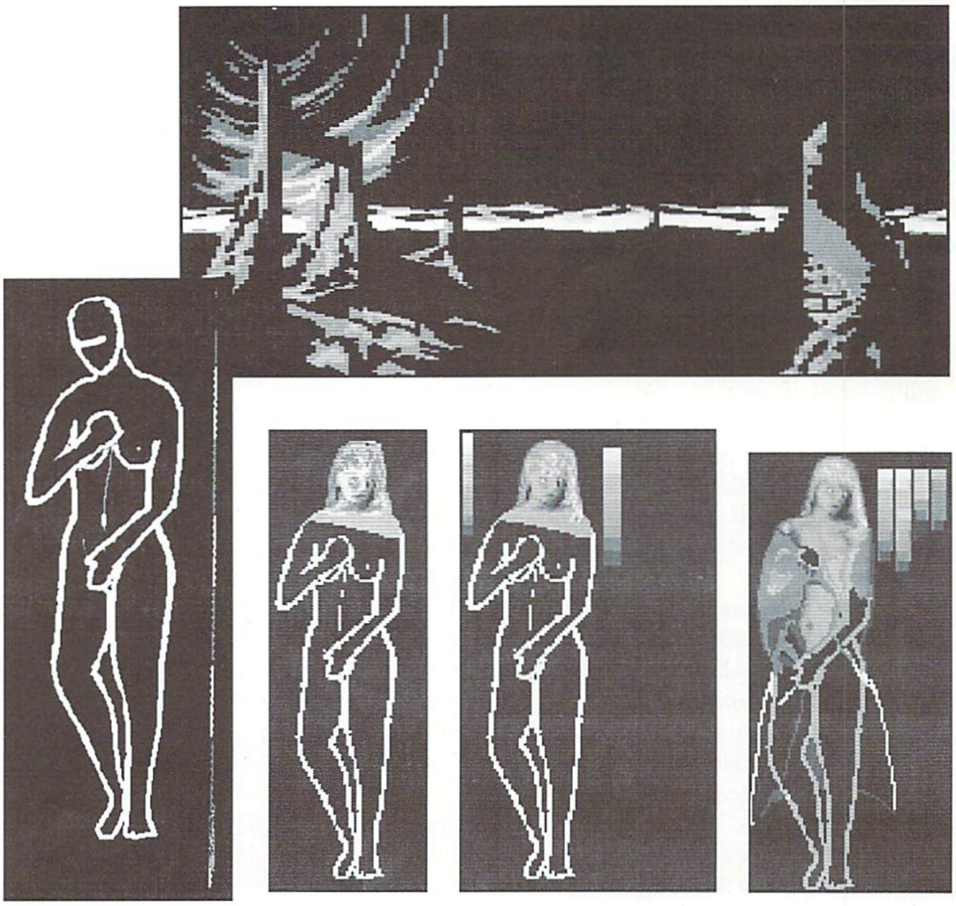
It took a small VB program, and lots of handiwork to combine all the parts, and display them together on the 64. Final polishing was done on the party after I received some critics: the sky was filled, and some stones were smoothed, when some people told me they looked as if they were scanned. All in all, the work took about four months.

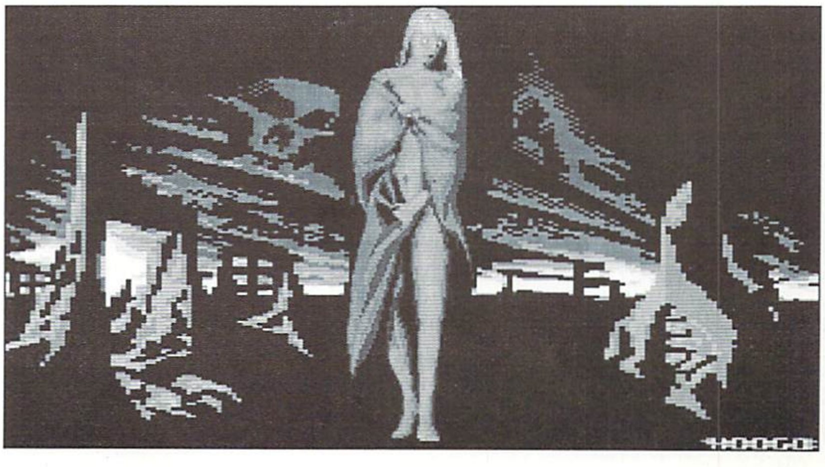
(vr)





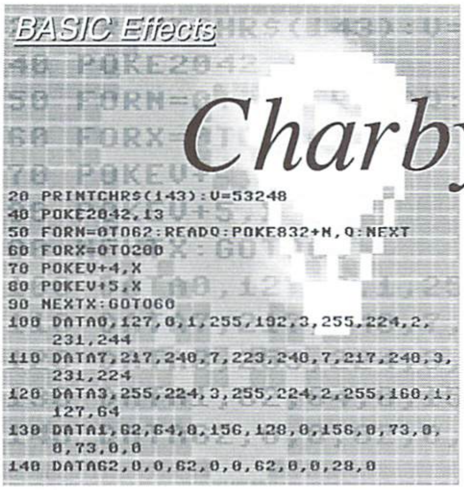
The creation of a masterpiece of art step by step from the beginning to the finished composition.





BASIC Effects

Charbyter



```
20 PRINTCHR$(143):V=53248
40 POKE2042,13
50 FORN=0TO62:READQ:POKE832+N,Q:NEXT
60 FORX=0TO200
70 POKEV+4,X
80 POKEV+5,X
90 NEXTX:GOTO60
100 DATA0,127,0,1,255,192,3,255,224,2,
    231,244
110 DATA7,217,240,7,223,240,7,217,240,3,
    231,224
120 DATA3,255,224,3,255,224,2,255,160,1,
    127,64
130 DATA1,62,64,0,156,128,0,156,0,73,0,
    0,73,0,0
140 DATA62,0,0,62,0,0,62,0,0,28,0
```

Picture: Sebastian Bach

By now we know for sure that you can make effective use of the computer under BASIC, given that you know how. The following example gives further evidence to this.

by Marcus Hinzmann alias Silver Fox/ Satovia/Sanity

Today we are dealing with an effect of the completely different kind. In the last issues, we have learned how to handle the video RAM and have experimented a lot with colors. This time we are aiming at something different, today we are processing a charset to our own designs. Those of you knowing noter programs are aware of the possibility of changing the C64's characters, that is loading a new charset into memory and then displaying it. You can make games much more interesting, for example, by turning the normal CBM charset's "A" into a self- designed "A", possibly in Gothic type (see also our assembly language course, parts 7 and 8 in issues 10 and 11 /1999). But what's this to our charbyter effect?

We want to animate the entire screen symmetrical to our liking today. How is this meant? Well, we paint a character "live", which is shown immediately on the entire screen. Thus we have 1000 (25 lines x 40 columns) facets which continuously change as if by magic, becoming circles or other figures, for example.

Memory Address 53272 ($d018)

As we know by now, memory address 53272 is responsible for the charset's and video RAM's location. Because of this, its value is split into half: all even number below 16 influence the charset's starting address (the C64 automatically adds 1, by the way, but this is then ignored later on). The real address results from multiplying this value by 1024. Multiples of 16 (up to 240) added to it, on the other hand, determine the video RAM's location. In order to get that final address, multiply by 64.

If you load a charset to 8192 (hex: $2000 in VIC bank 0), for instance, you only need to write 24 into that memory location. The value 24 consists of 16 plus 8 - the charset therefore resides at 8*1024 (8192), the desired video RAM at 16*64 (1024). And you can already see the new charset on the screen!

This memory location is also important for our effect. Since we want to keep editing a character from the charset at $2000, we have to put those 24 into 53272 so that this also becomes visible.

Thoughts in Advance

First we have to pick a character we want to edit continuously. You can find the way a character matrix is structured in issue 10/99 page 11. Just to give a swift sketch, every character is divided into 8 times 8 dots, one horizontal row of dots matching one byte, thus a value from 0 to 255. This means that we have eight rows of dots one under the other, so that we need eight bytes for a complete character.

We are therefore looking for a program that always reads eight values and the immediately assigns them to the chosen character. Let's take for our example the at symbol. It is the very first character in a charset. First we have to determine where the character's matrix (the eight consecutive bytes) resides in memory. If we place the charset at $2000, it will have the following structure. The first eight values ($2000-$2007) are for the at symbol, the following eight ones ($2008- $200f) are for A, the next eight values ($2010- $2017) are for B, and so on. Thus it is easy to find out, for instance, where the eight values for the letter D reside. D is the charset's fifth character, so that we calculate

starting address + 8 * (character position -1)

For D this would be:

8192 + 8 * (5-1)

resulting then in 8224 or $2020 in hexadecimal. By the way: a character's screen code gives us the correct value for the parenthesis in our formula right away; D has the screen code 4, so that starting address + 8 * screen code has the same result.

We had chosen the at symbol for our example, so that the eight values are at $2000, because 8192 + 8 * (1-1) still remains 8192. Thus memory locations $2000 to $2007 are designated for our at symbol. Now it is simple to create a program.

The program ...

We will need various addresses several times. Therefore we are defining the appropriate variables for them at the routine's beginning. An important address is 53272 (location of charset and screen). Since we will also be editing our character very often, we will rather work with a real base address, to which we are adding the values 0 through 7 as needed. This base value is, as we have computed before, at 8192 ($2000). We define those two addresses in the first line using

b = 8192 and
z = 53272.

Because of us really wanting to see "movement" on the entire screen, we poke our chosen character (the at symbol) into the entire video RAM (starting at 1024) in the second program line.

Here you could personalize the program to fit your taste, though. You could, for example, PRINT big letters consisting of at symbols onto the screen. In one demo, I have PRINTed the three letters/digits "C64" onto the screen, every one of them nothing but combined at symbols. But for our presentation's sake simply having the entire screen move should suffice.

Since we are not PRINTing onto the screen, the color of the former characters is not changed. Therefore we are adding 54272 to our loop counter from 1024 to 2023 in the program, which leads to this:

1024 + 54272 equals 55296

The memory starting at 55296 ($d800) is responsible for the displayed characters' colors. These memory locations (55296 to 56295) are to be thought of in terms similar to 1024 to 2023. It is only that we are not POKEing characters into these locations but rather the appropriate color. Thus this routine is POKEing each character's color, in this case "white" (value 1), as the character is POKEd. When the routine is done we have a screen filled with white at symbols ready for the effect itself.

In the third program line, the charset starting at 8192 ($2000) is made visible. The means of computing the value are given in the introduction.

Line 4 (1030) is the main routine. If we wanted to play an animation having eight different phases, for instance, we would define our loop counter p (= animation Phase) from 0 to 7. But what is happening during a single animation step? We need to read eight data values during each animation step, since a character is composed of eight bytes. This is done by the loop counter a. During each of the eight runs of a, a value is read from the data lines and written to the appropriate memory location. Then the phase is over.

The base value, in other words the first row of dots of each animation step is 8192. During a's first run (a is then 0), the routine reads a value from the data lines and puts it into the variable w. W is then POKEd into the address "base value b (=8192) plus loop counter a (=0)". Thus, the value in w is written into 8192 during the first run. During the second run, it is stored in 8193, since a no longer contains 0 but has become 1. And that way we are going through the at symbol's eight rows of dots.

The next animation step is done, by reading on. Once all eight phases have been completed, the following happens: We reset the data line pointer to 0 by means of RESTORE. Otherwise, the next run would obviously result in an error message displayed. Every time we are executing a READ, the pointer for data values is automatically increased by one. If we don't reset it, it will want to read more data, but finding none report an error. Once we have reset the pointer to zero, therefore, we jump back to the main routine and the animation begins anew.

The end of the program is made up of the data lines. To make things easier to read, I have put always eight values, in other words the eight bytes for an animation step's character, into a single data line. Eight phases need therefore eight data lines. The first one of the eight values per data line is the byte for the uppermost row of dots in the character. Value 2 defines the second row, and so on. If you write it this clearly, it will be easier to add or edit animation steps later on!

```
1000 b=8192:
            z=53272

1010 FOR t = 1024 TO 2023:
                   POKE t,0:
                   POKE t+54272,15
            NEXT:

1020 POKE z,24:

1030 FOR p = 1 TO 8:
            FOR a = 0 TO 7:
                   READ w:
                   POKE b+a,w:
                   NEXT:
            NEXT:
            RESTORE:
            GOTO 1030

1050 DATA 1,1,1,1,1,1,1,1
1060 DATA 3,3,3,3,3,3,3,3
1070 DATA 7,7,7,7,7,7,7,7
1080 DATA 15,15,15,15,15,15,15,15
1090 DATA 31,31,31,31,31,31,31,31
1100 DATA 63,63,63,63,63,63,63,63
1110 DATA 27,127,127,127,127,127,127,127
1120 DATA
255,255,255,255,255,255,255,255
```

A very simple and most importantly short effect, but you can prove again, that you can do a lot in BASIC!

If you have questions about the article, programming things (assembly language/ BASIC), or also about handling the DMC on the Internet, you can now reach me at

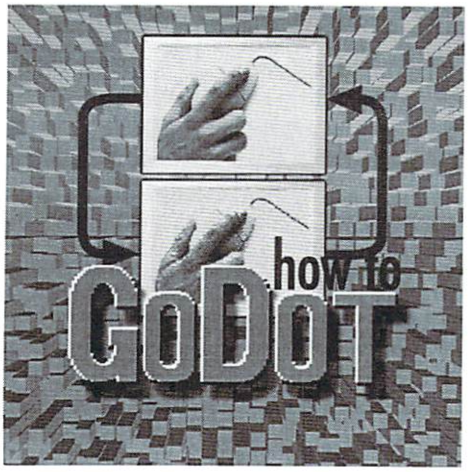
silverfox@foni.net

Good luck at experimenting says Marcus Hinzmann a.k.a. Silver Fox

(ad)

Memory Address 53272 ($d018):

53272	base address for video RAM and charset
8192	base address for the charset used



Picture: Sebastian Bach

by Arndt Dettke

Printing posters

Today, we're going to use the following modules: **ldr.GIF**, **mod.DecodeGIF**, **mod.ClipWorks**, **mod.OddSwap**, **mod.FrameClip** and **mod.CanonBJC.drv** (or another suitable printer driver). You can save some time if you install mod.ClipWorks and the printer driver into the REU (by using the **mod.REUTool** - I won't go further into this, though).

Do you have patience? Are you skilled with scissors, glue and paper? Yes? Then you can use GoDot to print colored or black and white posters. You can obtain good-looking results up to a size of 90x56 cm (b/w), and 81x51 cm (color). If you don't mind getting very large pixels (depends on the usage, and where you're going to display the poster), you can double the size to 180x112 cm (b/w) or 162x102 cm (color). For exact sizes, take a look at the table, since the possible sizes differ between printers.

If you're using a color printer, GoDot lets you choose from three base sizes (tiny, norm and lrge). A poster is made up of several enlarged image clips printed separately. Unfortunately, mod.ClipWorks requires the edges of a picture clip to be aligned with the borders of the bitmap tiles, which means it's not possible to zoom to just any desired level. I suggest using zoom factors of 2, 2.5, 5 and multiples of these. The first two are a bit complicated to handle, or they lead to overlaps or omissions in the image. The 5x zoom is the simplest, and that's why we're going to use it for the moment.

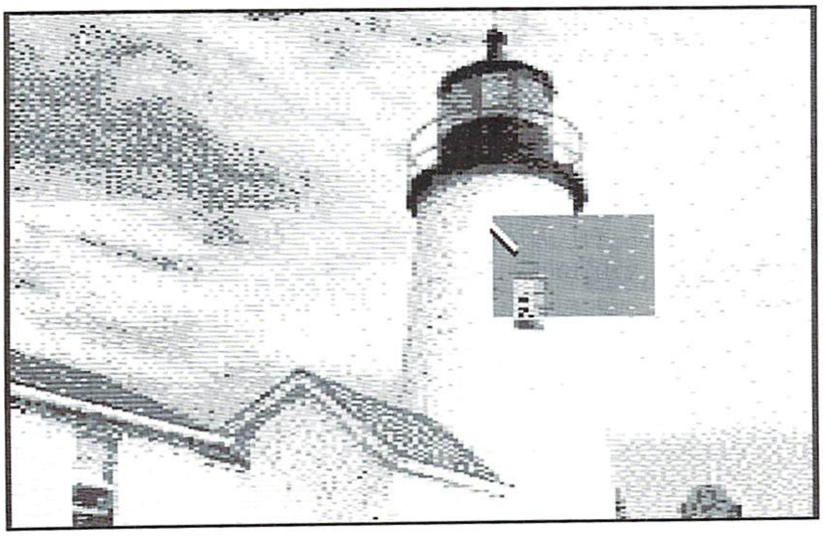
At zoom level 5, a 320x200 pixel image can be divided in 25 clips each measuring 8x5 tiles. Fig. 1 shows how part #14 is defined. It's located at line 10, column 24 (fig. 2). Each part is enlarged and printed separately, and then all 25 are combined to one large image. For this tutorial, I chose an image that doesn't use too much ink. Of course you can use any image to print a poster, but always mind that printing large dark areas requires enormous amounts of ink! And, finally, you should have an REU if you want to try this yourself, or at least a very fast drive (RAMLink or HD), as the image and the image processing modules have to be reloaded 25 times over.

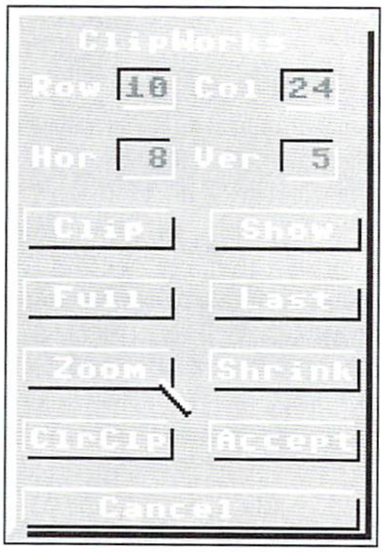
We'll proceed like this:

```
Load: GIF
Inst: DecodeGIF
Load "pcd019.gif"
Load GIF
```

This will fetch the image into memory. Now we'll prepare it for printing. When a GIF image is read by GoDot, it's prepared to be displayed as an IFLI image, which would result in stripes in a printout (you can read more on this in issue 1/2000). The following operations will remove these stripes, and put a border around the image (you should always do this!):

```
Inst: OddSwap
Execute
Inst: FrameClip
```




ClipWorks

Row 10 Col 24

Hor 8 Ver 5

```
Execute
(Select) red (or whatever
you like)
Exec
Display
```

The image displayed is the source image for our poster. We must save it now: in the following it's going to get under the scalpel 25 times.

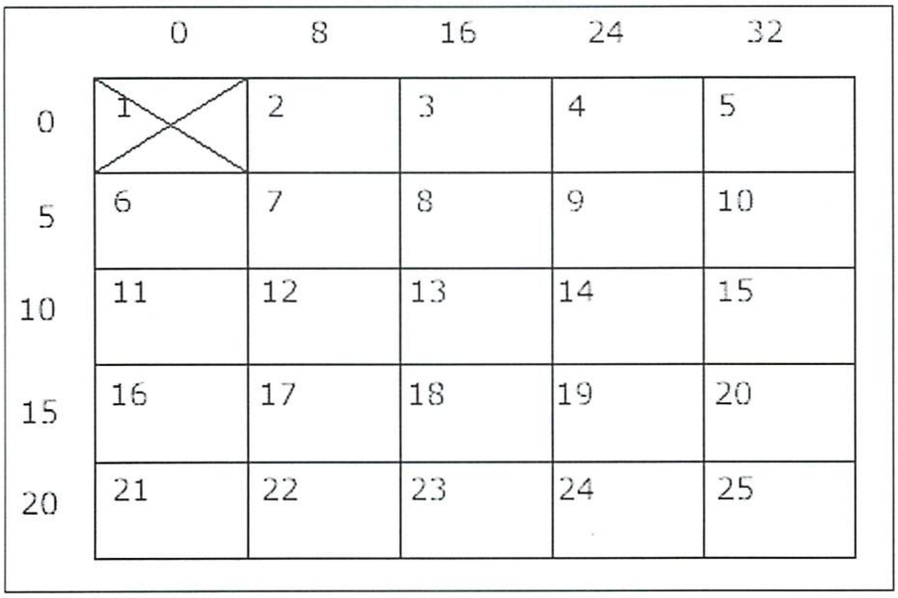
```
Save
Unit: RAM (or any other
fast storage drive, if you
don't have an REU)
Save "Undo 4Bit"
```

Well, that's it for the preparations, now lets get serious. During the following procedure, we'll be changing back and forth between the mod.ClipWorks module, and the chosen printer driver (in these examples, it's one for the Canon BJC). Since we'll do this 25 times, you should plan ahead a little to avoid getting confused. It's recommended to prepare a rough drawing showing how the image is divided into clips, and their individual row and column coordinates, if possible. The figure shows an example.

Let's go now!

```
Inst: ClipWorks
Execute
Hor: 8
Ver: 5
Show (just to verify - do
this for every clip,
you'll be thankful you
did!)
Zoom
Accept
```

Please don't choose "Display" now - the "Show" command would produce garbage during the other 24 turns. If you want to



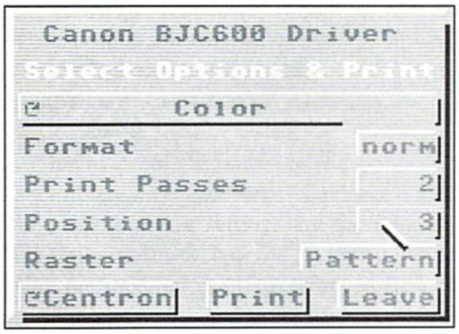
	0	8	16	24	32
0	1	2	3	4	5
5	6	7	8	9	10
10	11	12	13	14	15
15	16	17	18	19	20
20	21	22	23	24	25

make sure that "Zoom" has done its work correctly, you can click the Preview gadget, that will suffice normally.

Now it's time to print:

```
Inst: CanonBJC.drv
Execute
Color
Format: norm
Print Passes: 2
Position: 1
Raster: Pattern
Centron
Print
```

Most of the above settings are the defaults. What they mean is that you'll print in color, using the medium print size (8 clips fit onto an A4 page this way), printing two-passes for higher color saturation, starting in the top left corner of the printed page, dithering with the system pattern (which can be modified in mod.PatternEd), and printing via a parallel cable (geoCable or similar). Those who have to use a printer interface, select the appropriate mode (usually transparent mode). The Xetec interface, which is a widely popular one in the US, is compatible with the Merlin C+. SuperCPU users, please note that GoDot doesn't care if the printer can actually process all the data it gets. This shouldn't matter if the printer has a buffer RAM, as it will be fast enough. If not, you should set the SuperCPU to 1 MHz while printing; this will avoid interruptions and ugly printouts. So! The first part of the poster is printed, and you'll have to repeat the following operations 24 times:
Repeat...



Canon BJC600 Driver
Select Options & Print
Color
Format: norm
Print Passes: 2
Position: 3
Raster: Pattern

```
Load
Unit: RAM
Load "Undo 4Bit"
Preview (see if all went
well)
Inst: ClipWorks
Execute
Col: 0, 8, 16, 24 oder 32
(depending on the location
of the image clip)
Row: 0, 5, 10, 15 oder 20
Zoom
Accept
```

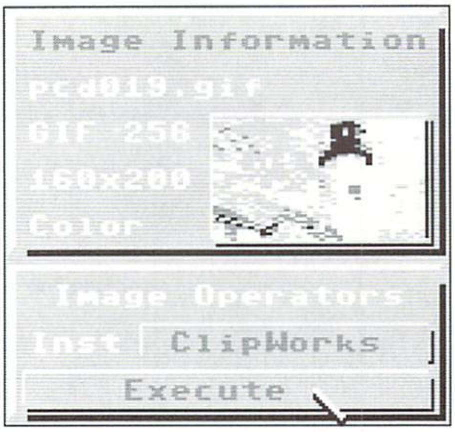


Image Information
pcd019.gif
GIF 256
160x200
Color
Image Operators
Inst: ClipWorks

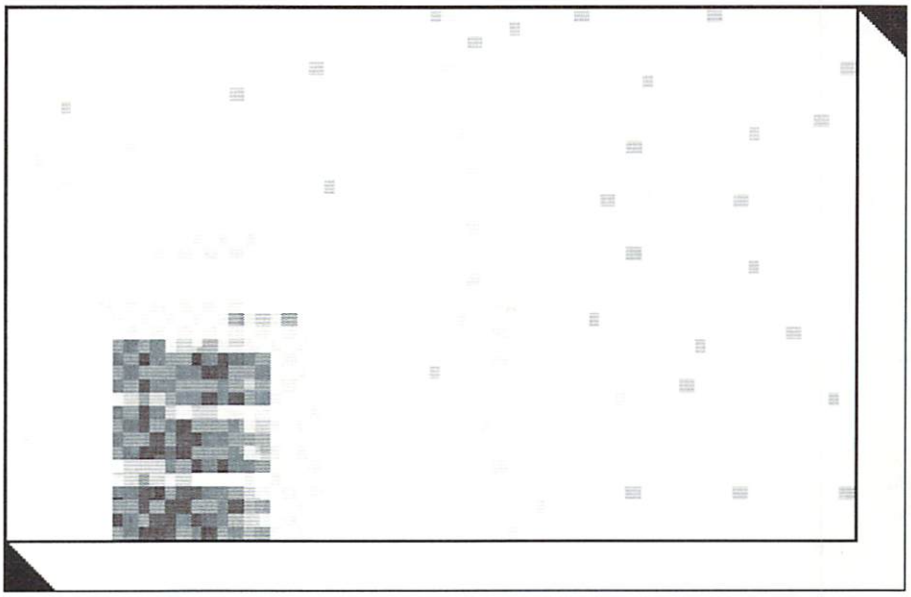
At this point, re-insert the paper into the printer in the correct orientation, unless it's fully printed, of course - in this case, use a new sheet.

```
Inst: CanonBJC.drv
Execute
Position: 1, 2, 3, 4, 5, 6,
7 oder 8 (depending on
printing progress)
Print
```

... until done 24 times.

You should now have four sheets lying in front of you; three fully printed ones, and another with just a single image clip. GoDot, the 64 and the printer have done their job now, the rest is up to you. Get your pencil, scissors and glue ready and move to a large table, where you have all the space you need. To avoid a chaos, it's best to number the pieces on their back before cutting them apart. Please try not to press on the paper too hard, or the numbers will show through the paper!

Now you can begin the scalpel work. There



Printing Posters

	Standard		
	Tiny	Norm	Lrge
9-Pin	-	-	215 x 5.55"
24-Pin	-	-	180.5 x 4.44"
300 dpi (HP)	54 x 1.33"	108 x 2.66"	162 x 4.00"
360 dpi (Canon, Epson)	45 x 1.10"	90 x 2.21"	135 x 3.31"
	Five times as big		
	Tiny	Norm	Lrge
9-Pin	-	-	1075 x 27.75"
24-Pin	-	-	902.5 x 22.20"
300 dpi (HP)	270 x 6.65"	537.5 x 13.29"	810 x 19.98"
360 dpi (Canon, Epson)	225 x 5.51"	450 x 11.07"	675 x 16.54"

Other scale values are possible. All given values are fixed with two digits.

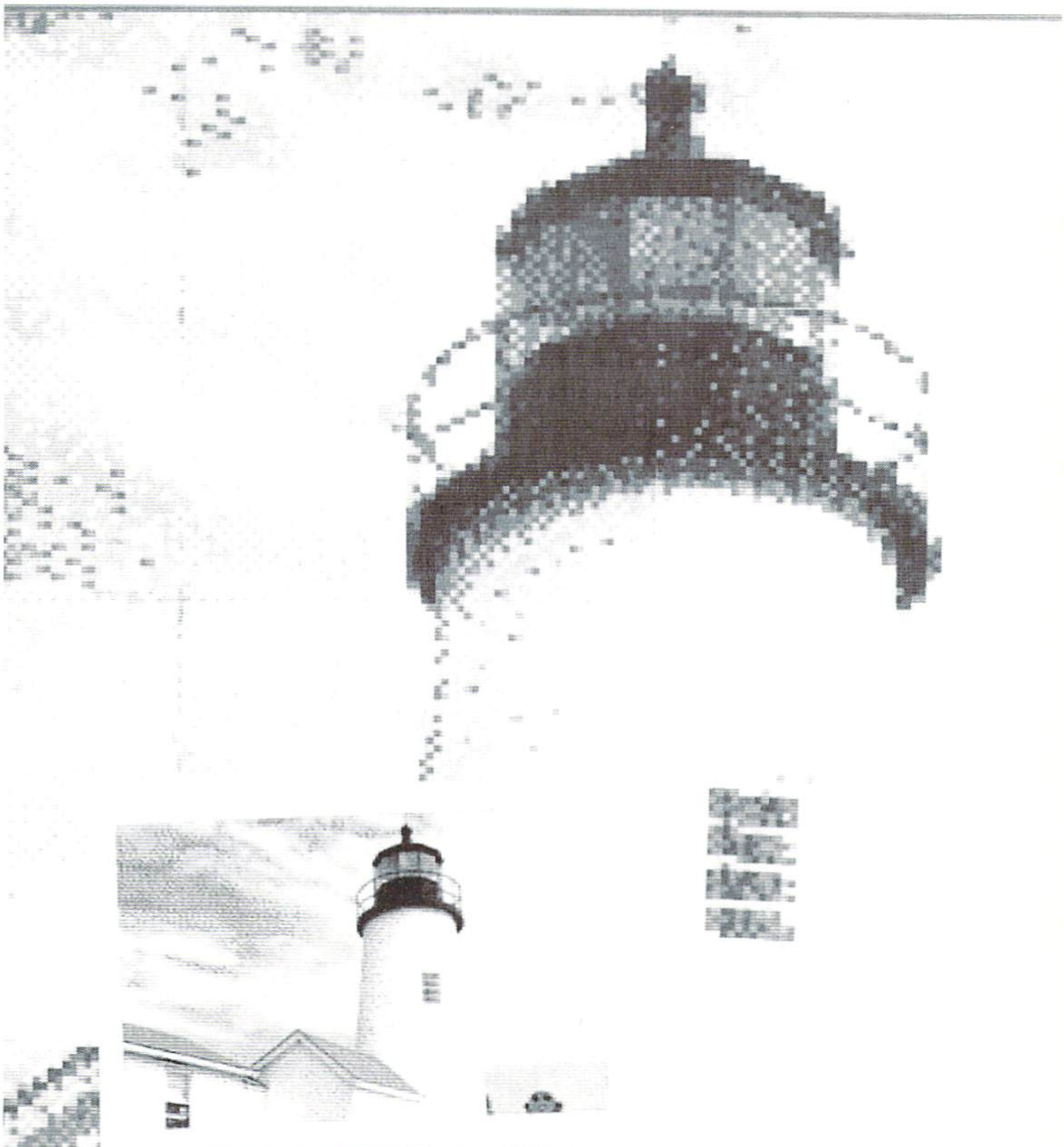
are some things you should keep in mind. Of course, you have to leave a margin where the parts will be glued together, but only on 2 sides, not four. I'd suggest to leave a margin at the right side and at the bottom of every piece. The left and upper edge have to be cut accurately to the millimeter, without leaving any white border, however narrow it may be! This is particularly important for dark image pieces. On the other hand, a little inaccuracy will probably not hurt in light areas of the motif. Also, I'd recommend you cut out the lower left and upper right corners of the margin. By doing this, you avoid overlapping more than two layers of paper when the pieces are glued together. The image shows what I mean (it's clip #14 of image 1 again). But keep this in mind: Don't cut the edges with a red border - these are the border of the poster itself. In fact, you should leave a large white space around these edges, it's going to be our mounting frame. Well now, After a long time, you'll have 25 little image pieces lying there, and all that's left is glueing them together. If you like, put them in the right order before you start glueing.

Now put the glue on the margins of each part, adjust the next part to it, repeat 25 times, and cut an even edge to the poster - and you're done!

To print a double-sized poster, the source image is read in twice as large. You can do this with the skip gadgets in ldr.GIF: X-Skip must be set to "0", and Y-Skip to "d". Please note you can't select the lower half of the image precisely, since you can only select complete tiles. Therefore you should leave a little overlap between image clips, by setting "Top" (the distance from the upper edge) to 96 (use SetClip). Unfortunately, some of the bottom area will be missing afterwards, as well as the red border. It's up to you how you handle this.

That's it. Next time, we'll try to give our images a more atmospheric look. We'll see what this means...

Yours,
Arndt



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 and Chief Editor: Enno Coners
Editorial Staff: Sebastian Bach *(sb)*, 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)*, Sven Tewes *(st)*
Editorial Assistance: M. Baye, M. Hinzmann, F. Jürke, R. Ekanayake Mudiyanselage, M. Mundt, M. Randall, K. Dale Sidebottom, C. J. Thomson, F. Wagenknecht
Supervisor: A. Dettke **Int. Supervisor:** R. Bernardo, N. Dannenberg, G. Gasson, D. Mohr
Translations: Bettina Beck, Matt Bodnarchuk, Stephan Lesch, Ranjith Ekanayake Mudiyanselage, Kendra Thiemann; **NTSC-Fixing:** Wanja Gayk, 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

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, cosowi@t-online.de

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.

Commodore 64

CMD HD 52 MB with PSU and user's guide, disk, cables, all like new for DM 450; SuperCPU 64 with user's guide, disk like new DM 280. Floppy CMD FD 2000 like new DM 280, Floppy 1541-II with PSU, user's guide, cables, disk in best shape DM 60. RAM expansion Commodore REU upgraded to 1 MB in best shape DM 150, keyboard for Commodore C128D, new, DM 60. GEOS games Vol1-4 new, still enclosed in their originally state all together for DM 30. Modem 14400 with RAM, GeoFAX compatible, new for DM 70. Tel: +49 2053 7765 from 7 p.m. local time (Central European Time).

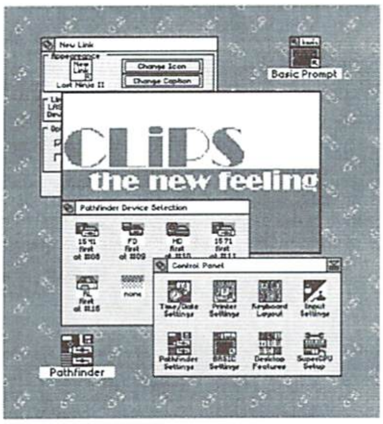
For free - Contact Corner

Better check this out: one of the last German C64 Mailboxes, fast, beautiful and colorful: +49 202 504 306

Hi! I own a C64 for about 3 years and haven't got that much experience though. Therefore I want to find people who share this enthusiasm for the C64. Contact me under: Marcel Krense, Str. d. Freunschaft 10, 19089 Demen, Tel +49 38488 20 261

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 gets ready to revolutionize the C64's world of operating systems. Though CLiPS is still in development, the design of the parts which are not yet programmed goes on. From now on in each issue, we are going to report about news from the CliPs "development labs", and we will keep you informed about any new, great approaches, ideas, and innovations. In this issue we want to explain the multi-tasking we touched upon in our last CLiPS report.

by Malte Mundt

CLiPS will be able to run several programs quasi-simultaneously. Pre-emptive multi-tasking makes it possible; for iastance, one application can just cheerfully calculate and calculate while the process manager interrupts its action and passes the baton to a second application. It is stopped a bit later on, and a third program gets the opportunity to work on its task. After this happens, the SuperCPU switches again to the first application, which continues at the point just where it was interrupted a short while ago. This process happens in a split second, and to us it looks like everything happens at the same time. Sounds like trickery, eh? But it's not; every pre-emptive multi-tasking operating system works exactly this way, unless it controls several processors!

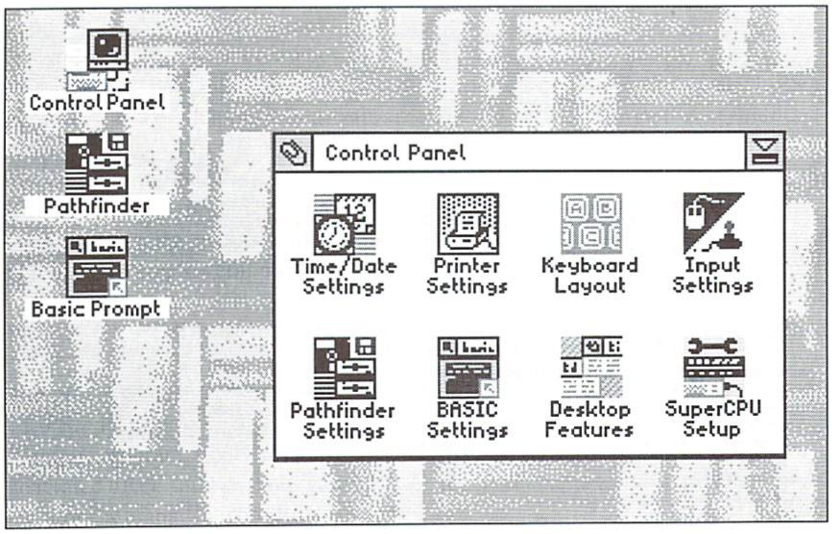
How is CLiPS 'multi-tasking?'?

The process manager is an important part of CLiPS. It controls the multi-tasking. A running program gets stopped by a CIA-generated IRQ, thus passing control over to the process manager. The IRQ is triggered by the countdown of a CIA timer. This timer is now loaded with a new value which determines how long the next application is allowed to run unless the process manager is called by the IRQ again. This can be defined individually for every program. It hasn't yet been decided whether a CLiPS user should be able to change this at runtime (like he can when tweaking the task priority on the Amiga) or whether a CLiPS programmer should determine this.

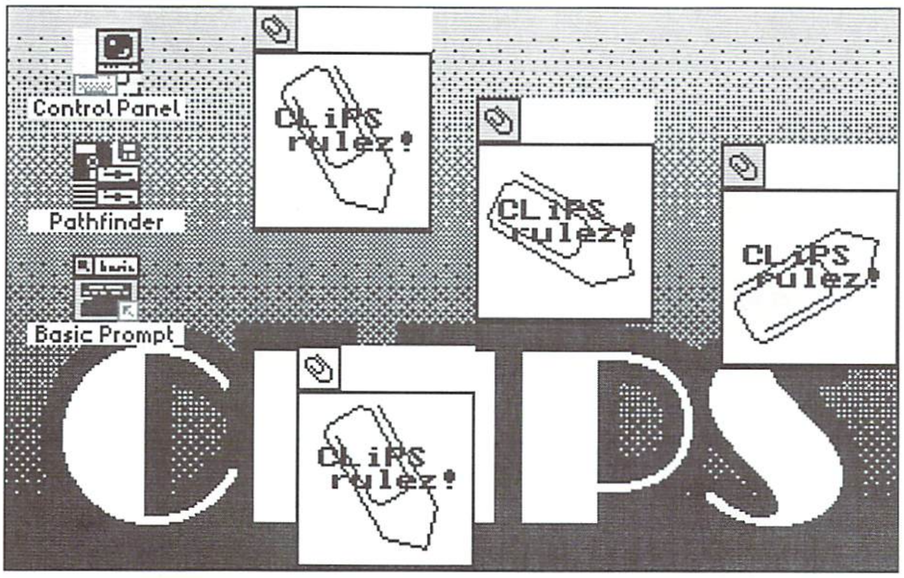
By means of this simple principle, the foundations of pre-emptive multi-tasking are laid. Up to a certain limit, this system also allows a program to always run at the same speed, no matter how many other programs are launched. To achieve this, the program has to always get the same percentage of the available time.

There is no time wasted

Task changing happens extremely fast; a copy of the current stack and zero page are dropped, and the corresponding registers have to be changed to the locations corresponding to the program. The suitable variables - the stored registers and the return address - can be found exactly there. As already mentioned in the last CLiPS report, a program which is waiting for input (from mouse or keyboard) will not get any time at all. For instance, when the process manager recognizes that a corresponding window is clicked upon, then time is given to the program to evaluate the mouse click. A new timer feature won't always give time to an application during its 'turn' but only when a certain amount of time has gone by.



A simple example illustrates this capability clearly. Why should a clock that must move its second hand only once per second obtain computing time many times a second in order to distinguish that this very second has not passed yet? It is true that it could pass control back to the process manager, but the timer method is much more efficient. That way, the clock is allotted time after exactly one second, redraws its hand, and goes to bed for a whole second again. And this is half an eternity for our SuperCPU in that it can work on other programs while running under CLiPS! Of course, the timers used are 'soft' and are synchronized using the monitor's video raster beam. Among other things, the real CIA timers located in the C64 are used to control multi-tasking. Besides, a NMI using the timer in CIA 2 is also possible, for example, when playing digitized music or speech with enabled display and running under multi-tasking.



A dream comes true: preemptive multitasking on a C64

Code callable several times

In multi-tasking the kernal itself gains particular importance; its code must be 're-entrant'. That means if a program calls a CLiPS kernal routine and the process manager switches to another task calling the same CLiPS function, it will not become a mess of variables. When it is the first program's turn again, the currently running CLiPS routine must find the same values in its variables again which were stored by the time of the task change. At the call of the same routine from within another program, the old values must somehow be preserved. Luckily, the 65816 comes to our aid again. Each program has its own direct page (formerly zero page). A small area of it can be used in our own programs. The CLiPS routines store variables exclusively in this area. This is the only way to guarantee that a CLiPS routine leaves alone the variables created during a call from another application. You can say CLiPS doesn't 'know' that it was actually running and that it was interrupted right in the middle by the multi-tasking, just to be accidentally called by anothSr program. The alternative would be to stop the multi-tasking at kernal calls until the routine has left again, but this is unthinkable for an efficiently working operating system.

Also self-modifying code can't be used in the CLiPS kernal. Of course, this doesn't mean that CLiPS programmers must do without this often very efficient method in their programs. For example, instead of checking flags in a complicated way, you can simply modify an op-code or something like that. Exception - if you want to launch exactly the same code several times while holding it only once in memory, self-modification must be avoided.

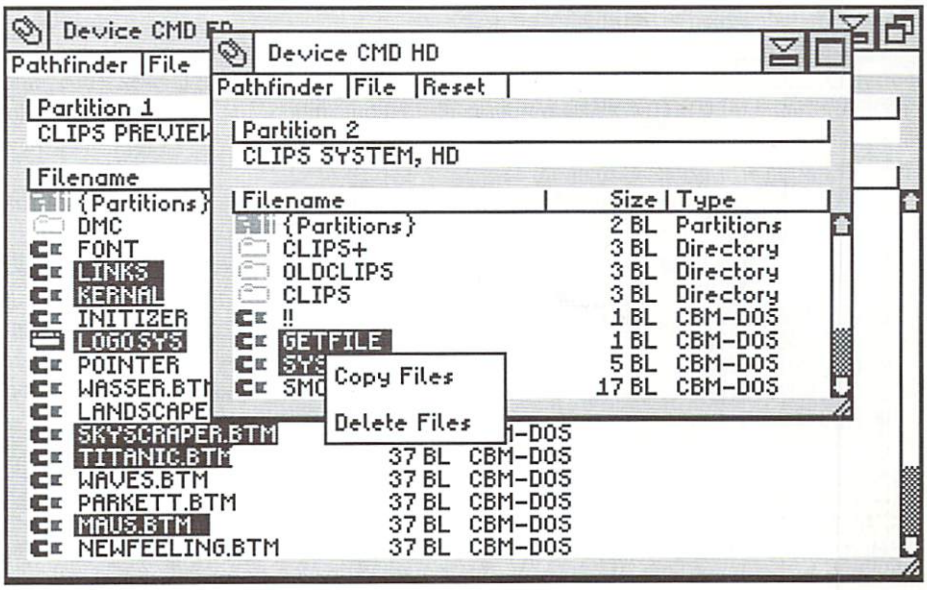
Multi-threading?

As if this isn't enough, CLiPS will also provide so-called multi-threading. Said in a simplified way, this means 'program-internal multi-tasking.' Multi-threading offers two enormous advantages. On the one hand, an assembler can assemble a source code while editing another - editor and assembler being two separate threads. On the other hand, you can now also assemble a second source code; a second assembling thread is launched. Then the assembling part is not copied in the memory so that it exists two times, but the same program code is assigned a second direct page and a second stack. A simple example is represented by graphical web browsers on other computer platforms; many pictures are created as if they were drawn at the same time. In the course of this, you can still press the stop button or call a find function. This is possible without multi-threading, but then it would rather be equivalent to a C64 BASIC program having some more instructions between a GET A$ and an IF A$="" GOTO, like when creating a flashing message while waiting for a key press.

Disk access and multi-tasking?

Anyone who works with a PC or Amiga naturally knows that currently running programs are not stopped while the hard drive is accessed. Unfortunately, this is not possible on the C64 without further work. Of course, a kind of IRQ loader could be developed which loads while the multi-tasking of CLiPS is active, but this would slow the computer tremendously; a 1541 or FD-2000 would need just as much in system resources as a CMD HD in order to be going full blast. However, the RAMLink may be an exception here, because it is not an IEC bus device and does not have critical timing needs. Yet, it is fast enough so that it wouldn't pay to develop a RAMLink load routine that works during multi-tasking. CLiPS prefers another way; for each drive, separate loading and saving routines are written which are especially programmed for the corresponding device. This way, speeds are reached which are beyond the transfer rates known under GEOS or even JiffyDOS.

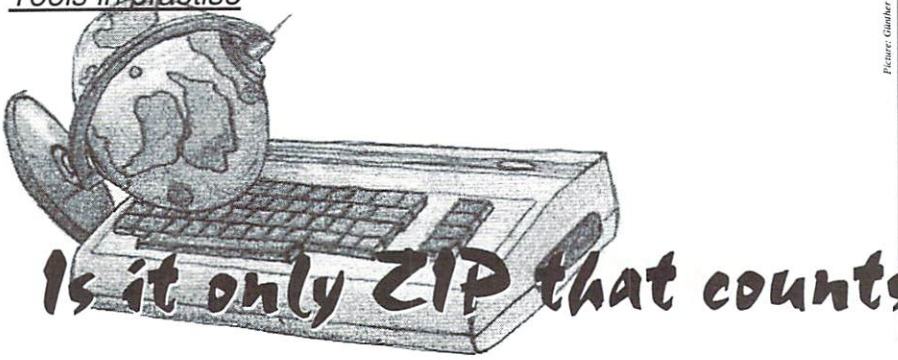
In the next CLiPS corner we will have a closer look at the handling of windows within CLiPS. Here too, there are things waiting for us which have never been seen before on the C64. (gb)

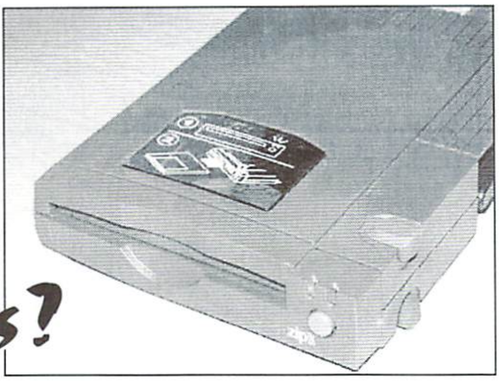


```
Device CMD FD
Pathfinder |File
Partition 1
CLIPS PREVIEW
Filename
{Partitions}
DMC
FONT
LINKS
KERNAL
INITIZER
LOGOSYS
POINTER
WASSER.BTM
LANDSCAPE
SKYSCRAPER.BTM              -DOS
TITANIC.BTM          37 BL  CBM-DOS
WAVES.BTM            37 BL  CBM-DOS
PARKETT.BTM          37 BL  CBM-DOS
MAUS.BTM             37 BL  CBM-DOS
NEWFEELING.BTM       37 BL  CBM-DOS
```

```
Device CMD HD
Pathfinder |File |Reset |
Partition 2
CLIPS SYSTEM, HD
Filename                 Size | Type
{Partitions}             2 BL  Partitions
CLIPS+                   3 BL  Directory
OLDCLIPS                 3 BL  Directory
CLIPS                    3 BL  Directory
!!                       1 BL  CBM-DOS
GETFILE                  1 BL  CBM-DOS
SYS                      5 BL  CBM-DOS
SMO                     17 BL  CBM-DOS
```

Tools in practise





Picture: Günther Bauer

Is it only ZIP that counts?

"Not another article about file compression!", I hear some readers cry. Don't worry, it's not one of these, it's about the well-known ZipDrive by iOmega.

by Nikolaus Metz

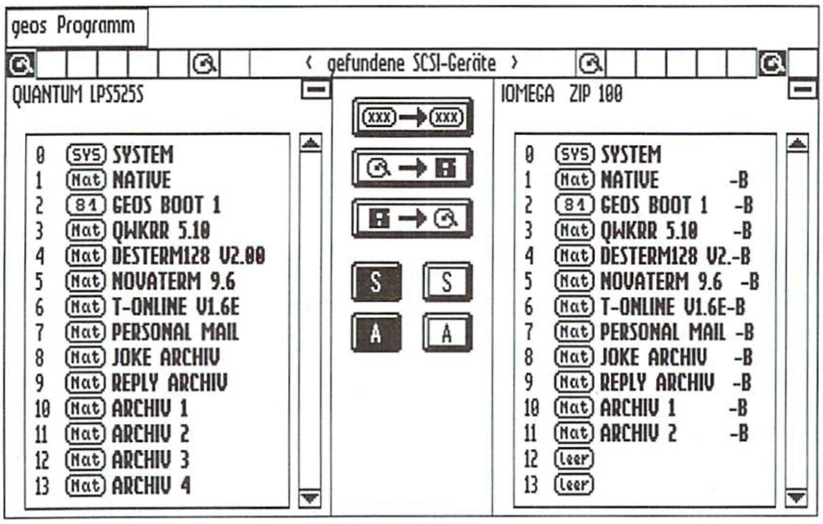
If you believe it's impossible to use modern hardware with Commodore machines, you should either turn to the next page, or read on and find out what we present in this article. There are always some innovative programmers who try to make hardware designed for PCs accessible to the C64/C128. One of their results is Ronny Bachmann's SCSI manager and the 'HD_ZIP' package, which works with a CMD hard disk and any SCSI devices connected tȯ it.

What's it good for?

Many hard disk users live in fear of a headcrash and of loosing their data forever. PC users can resort to fast and reliable utilities which save data on tapes or disks, an option that Commodore users didn't have until the end of 1998. With issue 10/98 of Go64!, Achim T"age introduced for the first time a backup program for the 128. However, it had a major drawback: it could only write an identical image of the hard disk contents to ZipDisks, or restore them from the disks. Since it's written in BASIC 7.0, it takes a long time, and lots of Zip disks, depending on the size of the hard disk. During a test run of my HD-500, saving the first 100 MB took more than an hour, which means the program is not suitable for general use yet. However, it wasn't intended as such anyway, but rather as an impulse, to show what can be done with simple means.

Alpha Version

It was purely by chance that I found the SCSI manager for Geos128 in my favourite mailbox (SachsenGEOS), and since I've got an iOmega-SCSI-ZipDrive, I connected it and tried the software. Naturally, I was surprised to find a program bearing a version number of 0.01, but as long as it works, I won't complain. In order to use the utility, you need a CMD hard disk, which allows for direct access to other SCSI drives (up to 7 without an additional controller). The ZipDrive is connected to it via a SCSI cable. Note that the drive must be terminated, which when preset, keeps all signals at proper levels when there is no data present so that this drive becomes the physical end-of-chain (that is, it "terminates" the SCSI chain). After all, we might get silly ideas and use modern devices like a CD ROM drive or a scanner with our Commodore...



```
geos Programm
                    ‹ gefundene SCSI-Geräte ›
QUANTUM LPS525S                         IOMEGA  ZIP 100

 0  (SYS) SYSTEM                         0  (SYS) SYSTEM
 1  (Nat) NATIVE                         1  (Nat) NATIVE          -B
 2  (81)  GEOS BOOT 1                    2  (81)  GEOS BOOT 1     -B
 3  (Nat) QWKRR 5.10                     3  (Nat) QWKRR 5.10      -B
 4  (Nat) DESTERM128 V2.00               4  (Nat) DESTERM128 V2.-B
 5  (Nat) NOVATERM 9.6                   5  (Nat) NOVATERM 9.6    -B
 6  (Nat) T-ONLINE V1.6E                 6  (Nat) T-ONLINE V1.6E-B
 7  (Nat) PERSONAL MAIL                  7  (Nat) PERSONAL MAIL   -B
 8  (Nat) JOKE ARCHIV                    8  (Nat) JOKE ARCHIV     -B
 9  (Nat) REPLY ARCHIV                   9  (Nat) REPLY ARCHIV    -B
10  (Nat) ARCHIV 1                      10  (Nat) ARCHIV 1        -B
11  (Nat) ARCHIV 2                      11  (Nat) ARCHIV 2        -B
12  (Nat) ARCHIV 3                      12  (Leer)
13  (Nat) ARCHIV 4                      13  (Leer)
```

Both drives are now activated

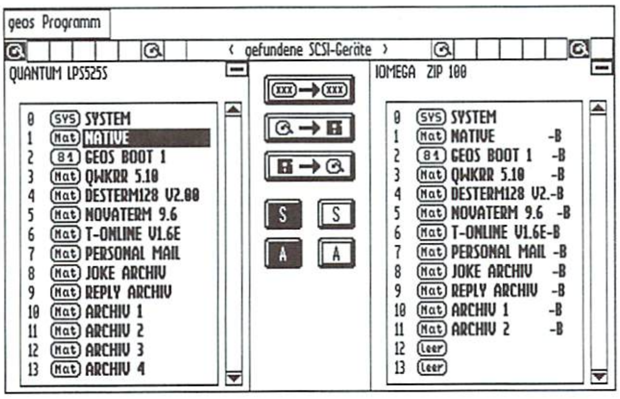
The Preparations

Before we can use the Zip disks, we have to format them and create partitions on them. Don't worry, nothing can happen as long as you heed all the advice presented here. First, you should take the time to read all of the article. Formatting is done in BASIC mode, using the 'LLFORMAT' tool, which can be found on the HD utilities disk. After starting it, you have to set the SCSI controller into install mode by pressing the HD's 'Reset' switch while holding the 'Swap8' and 'Swap9' switches. After you've confirmed this in the program, with the 'Return' key, the program will search for any connected SCSI devices.

Always Be Careful

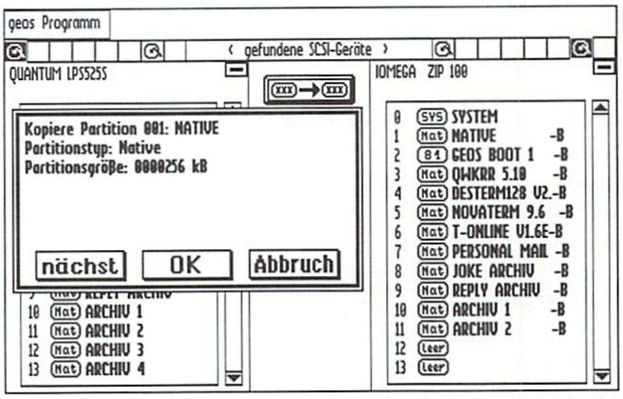
At this point you have to be careful when you select the device to format - in my system for example, the ZipDrive's device number is 6; 5 is also possible - this is selected with a switch on the back side of the drive housing. Attention: 0 is always the main hard disk itself!!! If no second SCSI drive is recognized, simply abort by pressing the 'RUN/STOP' key, and rerun LLFormat. If you do that, you don't have to enable the install mode again, unless you had pressed the 'Reset' key on the HD in the meanwhile (The program will ask you to when you exit, usually).

Now insert a new Zip disk and start the format procedure: choose the ZipDrive with



```
QUANTUM LPS525S
0  SYS SYSTEM
1  Nat NATIVE
2  81  GEOS BOOT 1
3  Nat QWKRR 5.10
4  Nat DESTERM128 V2.00
5  Nat NOVATERM 9.6
6  Nat T-ONLINE V1.6E
7  Nat PERSONAL MAIL
8  Nat JOKE ARCHIV
9  Nat REPLY ARCHIV
10 Nat ARCHIV 1
11 Nat ARCHIV 2
12 Nat ARCHIV 3
13 Nat ARCHIV 4

IOMEGA ZIP 100
0  SYS SYSTEM
1  Nat NATIVE         -B
2  81  GEOS BOOT 1    -B
3  Nat QWKRR 5.10     -B
4  Nat DESTERM128 V2.-B
5  Nat NOVATERM 9.6   -B
6  Nat T-ONLINE V1.6E-B
7  Nat PERSONAL MAIL -B
8  Nat JOKE ARCHIV    -B
9  Nat REPLY ARCHIV   -B
10 Nat ARCHIV 1       -B
11 Nat ARCHIV 2       -B
12 Leer
13 Leer
```

We've selected partition 1



```
Kopiere Partition 001: NATIVE
Partitionstyp: Native
Partitionsgröße: 0000256 kB
```

```
QUANTUM LPS525S
10 Nat ARCHIV 1
11 Nat ARCHIV 2
12 Nat ARCHIV 3
13 Nat ARCHIV 4

IOMEGA ZIP 100
0  SYS SYSTEM
1  Nat NATIVE         -B
2  81  GEOS BOOT 1    -B
3  Nat QWKRR 5.10     -B
4  Nat DESTERM128 V2.-B
5  Nat NOVATERM 9.6   -B
6  Nat T-ONLINE V1.6E-B
7  Nat PERSONAL MAIL -B
8  Nat JOKE ARCHIV    -B
9  Nat REPLY ARCHIV   -B
10 Nat ARCHIV 1       -B
11 Nat ARCHIV 2       -B
12 Leer
13 Leer
```

... and logged it on for copying

the cursor, and press the 'Return' key to confirm your choice. You can still discontinue at this point; remember that formatting a disk (Zip or otherwise) will irrevocably destroy all data. When you've made the right choice, you have to confirm the two security queries by entering 'Y' and pressing 'Return'; once again you can see which device you've chosen. After that, the program starts the format procedure, which should take about 10 minutes including testing. This is indicated by the wild flashing of the yellow data LED. If the program shows a message other than 'Format Successful' after formatting, you've got to repeat the process (with another disk, if it was a bad media), checking each and every step. (never happened to me, though).

Creating Partitions

Next, we need to create a system partition on the freshly formatted disk, which is going to hold infomation about the type, size, position and names of the other partitions we're going to create later. For this purpose we need the programs called 'create sys_e(al)' and 'HD-ZIP(al).ass' from the cover disk, as well as 'rewrite dos', 'system header', 'hdos Vx.xx' and 'geos/hd Vx.xx', which can be found on the HD utilities disk. Since there are different versions of the two latter files, the X'es stand for their version numbers; an HD-DOS V1.92 or above and GEOS/HD V2.00 or higher would be ideal. I'ts best to copy these six files to an empty 1541 or 1581 disk, together with the other files, since some of them are loaded on the fly - they only take 134 blocks anyway, and this way, we always have them at hand.

Now we run 'create sys_e(al),' and if necessary, switch the controller into install mode (Swap8+Swap9+Reset). The program recognizes this and asks for the device address of the drive - which means its SCSI address (6 or 5). Additionally, we have to input the Commodore address of the hard disk - 8 on my system, 12 is CMD's factory default setting. The following question for the start address can simply be answered with 'Return'; erasing this lower area is also unimportant for us, so this question can be confirmed with 'Y', since we don't want to share our Zip disk with another system.

Careful now: do not answer any of the questions before you've checked the number next to 'total storage' and made sure it reads 196 607 blocks, or 100 662 784 bytes (the block count seems to refer to the physical block size of 512 bytes). These numbers don't vary unless the post-format check has found bad spots on the disk and excluded them for security, something which never happened to me. However, should you find totally different values, maybe resembling the size of the hard disk, abort the program immediately, (best reset the computer), and restart the program, checking all steps carefully!

You have to explicitly run the FIX-BLOCKS program, found on the CMD HD Utilities disk, to go in and correct bad blocks in this case.

Ready for Backup?

After about half a minute, the system partition has been written to the disk, and the computer tells you to reset the hard disk controller, a request you can safely ignore at this point. We will now run 'HD-TOOLS.64[128]', which normally calls for the drive to be placed in "Configuration Mode" (which can normally be set by pressing Write Protect along with Reset). We didn't reset the controller after formatting, so it's still in "Install Mode", which will also work in place of Configuration Mode. RamLink owners who use the parallel cable for faster transfers, have to disable it with '@P0' before starting, otherwise access to the partition list will be denied. The program 'HD-TOOLS.64(128)' must be on some driver other than the HD, (like a 1541/71/81/FD), because we've switched to the ZipDrive before running it, and we temporarily can't access the data on the hard disk afterwards.

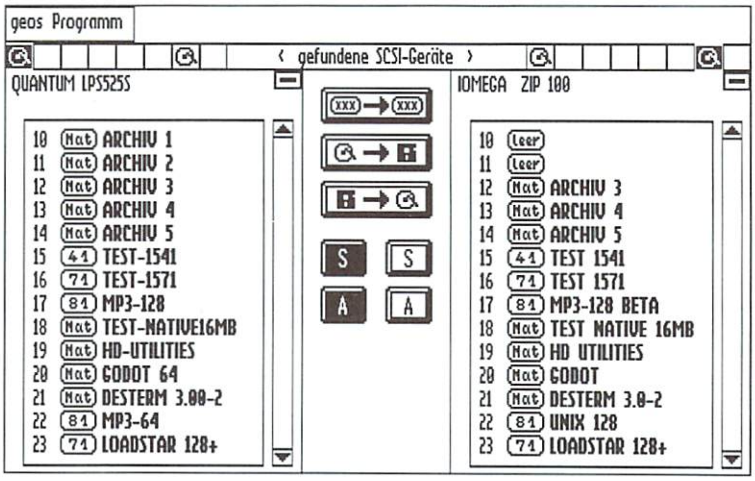
To allow for a reasonable use as a backup drive, it's recommended that you create the same partitions on the Zip disks as you've got on the hard disk (with native partitions, be careful to use the correct size!) I've appended a '-b' (which stands for 'backup') to the partition names on the Zip disk, so I've got a last resort for distinguishing them. Once the first Zip disk is partitioned, go on with the other disks, until you've got copies of all partitions on the Zip disks, in the same order as on the hard disk. I doesn't matter if there's a gap in the order on the hard disk; simply omit the missing partition numbers. You should, however, leave the same gaps on the Zip disks as well (that is, really omit the numbers), so you can add them later. Also, it's recommended that you note down the partition numbers on the disk label, in order to further avoid confusion.

Zip available under BASIC

If for any reason, you've pressed the reset button on the HD in the meantime, you can use the program 'HD-ZIP 64(al)' or 'HD-ZIP 128(al)' to switch back to the Zip drive. These small programs are a crucial point, and should always be at hand on a floppy disk. For instance, if you erase a partition from the ZipDisk (device 6) in the 'HD-TOOLS.64(128)' program, the controller will automatically reset to the hard disk (device 0) - it seems that the Bachmann brothers were the first to notice this bug. You can verify this by looking at the partition table after deleting a partition. This makes it clear that you have to be extremely careful if you want to avoid losing lots of data, or at least time.

Now we can use 'HD-ZIP 64/128(al)' to switch between the hard disk and another connected SCSI device in the normal BASIC mode of the C64/128, and access it directly. Note that you also have to run this program after a disk change under BASIC, to make the controller read the system partition of the new disk. Unfortunately, only one SCSI drive is available at a time, since the controller can't handle both devices at once. But, that's no problem for our purpose. Later, during backup when both drives have to be active at the same time, the SCSI-Manager will handle the switching automatically.

The second back-up disk



```
QUANTUM LPS525S                    IOMEGA  ZIP 100

10 (Nat) ARCHIV 1                  10 (leer)
11 (Nat) ARCHIV 2                  11 (leer)
12 (Nat) ARCHIV 3                  12 (Nat) ARCHIV 3
13 (Nat) ARCHIV 4                  13 (Nat) ARCHIV 4
14 (Nat) ARCHIV 5                  14 (Nat) ARCHIV 5
15 (41) TEST-1541                  15 (41) TEST 1541
16 (71) TEST-1571                  16 (71) TEST 1571
17 (81) MP3-128                    17 (81) MP3-128 BETA
18 (Nat) TEST-NATIVE16MB           18 (Nat) TEST NATIVE 16MB
19 (Nat) HD-UTILITIES              19 (Nat) HD UTILITIES
20 (Nat) GODOT 64                  20 (Nat) GODOT
21 (Nat) DESTERM 3.00-2            21 (Nat) DESTERM 3.0-2
22 (81) MP3-64                     22 (81) UNIX 128
23 (71) LOADSTAR 128+              23 (71) LOADSTAR 128+
```

GEOS Comes Into Play

Now we can finally deal with the real object of the article, the SCSI-Manager. Unfortunately, there only exists a version for the C128, but it works under GEOS, Wheels and MP3 as well, with some small differences that will be explained later. It runs only in 80 column mode, and requires the larger 64K VDC memory. After running, you'll find yourself with a user interface that reminds strongly of DoubleDesk; which is no wonder as it's by the same author.

The Displays

The 'geos' menu contains some 'info' about the SCSI-Manager, and the only option in 'Programm' is to quit ('Ende'). But that's sufficient, since everything else can be done with the buttons between the drive displays. For our task, I've activated the hard disk on the left-hand side, and the Zip drive on the right. To choose a drive, click on it's symbol in the bar reading 'gefundene SCSI-Ger"ate' (SCSI devices found), which displays all connected SCSI drives under their device number, starting at 0. Of course, you've got to insert the disk before, or the system partition won't be found. If the program has difficulty reading this partition, simply click the symbol again. In order to deactivate a drive, use the 'close' symbol in the upper right corner of its window. You can scroll the partition list with the arrows next to the display. The 'active' source drive is highlighted by a light green background, while the inactive has a dark green background.

The Backup

There are three upper icons, but only the first one (copy partitions - 'Partitionen kopieren') is important for our task. The other two ('gesamte Festplatte sichern bzw. wiederherstellen' - backup or restore whole hard disk) have no function yet. To transfer the contents of a hard disk partition to the Zip drive, we choose the partition and click on the upper symbol. A dialogue box appears, showing the contents. Now we choose the target partition on the second drive while the dialogue box is open, and information about it is displayed, too. After we click 'OK', a new box showing some garbage appears if running under MP3-128, which we can safely ignore; the only correct information here are the drive infos. Wheels128, on the other hand, displays the time needed for the copy, which depends on the size of the partition. A test at 20 MHz shows that the SCPU can hardly save any time, since the actual copying process is performed by the SCSI controller itself. Therefore it's not necessary to compare performance between the three systems. Another confirmation by clicking on 'OK' and the copy procedure starts; the times for the different partition sizes are shown in the table.

Of course, we could speed up the process by choosing several source partitions at once, assigning the target partitions separately; however, this only worked under Wheels when I tried it. The program simply needs a thorough inspection, and adaption to all user interfaces. By the way, the only thing you may never copy is the system partition, since it contains all partition information of the hard disk, while the several backup disks contain the disk information only, which differs strongly. If you want to save the system partition as well, it's recommendable to use CMD's 'HD Power Tools' (Backup Partition Table); using a separate ZipDisk just for that seems like a waste of space to me.

Conclusion

Surprisingly powerful for a freeware tool, the greatest advantage of the SCSI-Manager is that you don't have to backup the whole hard disk. This way, it's easy and quick to backup single partitions; the required time only depends on the size of the partition. However, saving the contents of the entire hard disk still takes some time. Filling a whole ZipDisk takes about 56 minutes, but usually, this is only necessary the first time. Your weekly backup-refreshes normally only take a few minutes, depending on the amount of data.

Another large plus is the ability to use the ZipDrive like a virtual hard disk, which means loading programs, scratching, copying, etc. works just as with any normal drive. Naturally, GEOS and its successors can be booted as well, without any changes or add-ons. All you have to do is copy the existing boot partition to the ZipDisk, switch to it, and you can immediately boot from it. Booting the newest MP3 version takes 26 seconds, which is only six seconds longer than booting from the hard disk.

This is actually a program that doesn't deserve collecting dust in a drawer - however, its developer Ronny Bachman has no interest in finishing it. Maybe there's someone else who could take on this task, and probably even enhance the software - I already have some ideas in store. Personally, I'd see it as another proof that our old breadbox doesn't have to stand in the corner, because the modern machines are supposed to be so much better.

GEOS_LQ

The history of printer drivers running under GEOS is a story full of misunderstandings... Most people probably have at one time or another been confused by the vast number of printer drivers and print programs available for GEOS. Lots of articles on printing have been published in the 64'er magazine and in GO64!, and it would be easy to write more than one book on the theme.

by Frank Wagenknecht

Therefore, this article logically cannot recount the diversity of this field. I am going to restrict myself to the great moments in the history of printer drivers under GEOS and to concentrate on text printing.

It came to pass in those days when you could print your documents under GEOS with a 9-pin dot matrix printer and unbelievable 80 dpi (special terms see list below; modern ink jet printers, in comparison, print up to 1440 dpi) that Thilo Herrmann radically changed the world (of GEOS) with his program GEOS_LQ.

There had been many attempts to improve the low quality of dot matrix prints: The very same line was overwritten various times or printed slightly dislocated in each pass. But the basic problem remained: the poor resolution. Prints could be made blacker with the aforementioned procedures, but at the same time they became more coarse-grained.

GEOS_LQ tried to cope with the problem differently. Printing is not executed directly from GeoWrite via the print routines found there. GEOS_LQ is a separate program that restructures GeoWrite documents on its own. To start the program, however, you do not have to change to the desktop but instead can use the 'Start_LQ' feature to call up GEOS_LQ directly from GeoWrite.

GEOS_LQ then exchanges the document's character sets for character sets with a higher resolution during printing. The idea is simple, but it provides astonishing printing results. Back then, advertisements for GEOS_LQ claimed that the printing quality was comparable to that of laser printers. Well, I never witnessed laser quality on my matrix printer, but the prints looked impressive nevertheless.

Of course, graphics in GeoWrite documents were also printed. The program did not have the same high resolution for graphics as for character sets, but the interpolation routines produced an acceptable degree of quality. GEOS_LQ surely provides WYSIWYG (see below). Nevertheless, printing with this program takes an awful lot of time ...

TextPrint

TextPrint follows another approach. Ordinary NLQ printing was nothing new under GEOS but at the time, NLQ printing features were very limited by practical restrictions. Thus, it was hardly possible to format the text: the programs did not allow for bold or outline print, italics or underlined sections of texts. Moreover, the graphics included in documents were not printed with the documents and WYSIWYG was completely out of the question.

TextPrint, too, is a separate program that can be started with an additional feature directly from GeoWrite, just like GEOS_LQ. TextPrint independently analyzes GeoWrite documents in the manner described above. At the same time it sends the appropriate codes for bold and outline print, italics or underlined text sections to the printer. In the TextPrint package, a special micro font is also included.

TextPrint prints graphics, and WYSIWYG is possible in most cases, although TextPrint does not offer as many different character sets as GEOS_LQ. On the other hand, it has the advantage of being much faster than GEOS_LQ because it uses the printer's built-in font library.

PrintText

PrintText? Is this a typing error? No, PrintText is yet another program. As ink jet printers became less and less expensive the importance of NLQ printing increased. GEOS_LQ prints looked very pale on ink jet printers whereas TextPrint made it possible to use the internal fonts of those printers, too. The prints were of high quality and very fast.

But not much later color ink jet printers became very cheap and affordable for C64users. There was just one problem: TextPrint did not support color printing. PrintText (test in GO64! 10/97) is a program fashioned on TextPrint and designed by Ro(nn)y Bachmann to solve the problem mentioned above: it is possible to color the text in paragraphs. Similar to the two other programs, PrintText is a separate program that also prints graphics.

Stylus_NLQ

Stylus_NLQ by Markus Kanet is a special case - it is not an additional program for GEOS. Instead, it is a package of NLQ printer drivers that work directly from GeoWrite. Control commands in the text give users the opportunity to use various fonts and styles. The version from the 64'er mag 1/97 also includes color printing. Colors are not only available for each paragraph - every character of one word can be colored differently.

Overall, Stylus_NLQ offers the most possibilities for text editing, though graphics printing is not supported by those drivers. Moreover, the WYSIWYG feature is completely lost. On the other hand, the program is not restricted to GeoWrite in contrast to the other programs described above. All GEOS applications that support NLQ printing, e.g. GeoCalc, can use these drivers. This is probably one of the biggest advantages of Stylus_NLQ.
(ad)

Special terms:

dpi - dots per inch - Dots Per Inch.the higher this value is, the better the printer's resolution.
WYSIWYG - What you see is what you get - he printed document will look exactly like the version you edit in the text editing program.
Interpolation - mathematical procedure in which in between every two existing dots further dots are placed. This procedure tries to smooth out the 'staircase' look in prints.
NLQ - Near Letter Quality - Near Letter Quality.A printing process which uses the internal special fonts many printers have

WORLD WATCH

by Rainer Buchty

This month's focus: Printing

HELLO WORLD,

in the Middle Ages, times were rough. If you wanted a facsimile of a book, you had to copy it by hand. But approximately in 1450 a certain Mr. Johannes Gutenberg from Germany invented letterpress printing. This hardware development could rightly be designated as the first printer. Of course, there has been an obvious evolution from the first letterpress printing machine to today's printers. For that reason, today we are going to take a closer look at the various types of prilters. How do they print on the paper?

Daisywheel printers

Although I do not have any exact dates, this type of printer certainly is the oldest one, because strictly speaking, it is nothing more than an electric typewriter connected to a computer. In daisywheel printers - contrary to conventional typewriters where each key is responsible for only one character - all printing types are arranged on a disc, the daisywheel. For each character to be printed, the wheel has to be positioned correctly, and the printing type is hammered onto the paper.

This procedure has the advantage of providing a high-quality print, but at the same time it only offers a limited set of characters. For printing language-specific characters, additional daisywheels are needed while printing graphics is not feasible at all (unless you accept ASCII art as graphics printing). Moreover, daisywheel printers naturally do not excel in outstanding printing speed because of the mechanism's limitations encountered when printing with a daisywheel (positioning the wheel, arresting it, and hammering the printing type onto the paper).

Ballhead printers

These printers work very similarly to daisywheel printers. However, the printing types are not arranged in linear order on a wheel. Instead, the printing types of ballhead printers lie in various horizontal lines on a spherical printhead.

Thus, the compact mechanism makes each printing type more easily accessible since the distance each type has to travel is shorter. Sometimes it is not even necessary to rotate the ballhead at all. Due to the parallel arrangement of the types, raising or lowering the ballhead is enough to reach another printing type.

In contrast to the daisywheel printers that print with a sort of hammer, in ballhead printers the printhead itself hammers the printing types onto the paper. This results in considerable mechanical strain, which makes this type of printer especially susceptible to mechanical failure.

Dot-matrix printers

Why should you work with fixed printing types? This question led to the development of dot-matrix printers. Instead of providing a single printing type for each printable character and printing it onto the paper with a hammer, dot-matrix printers work the other way round. There is only one single type (a dot) with a large number of pins. To make the printing process more economical, in most cases dot-matrix printers have only one vertical band with printing types instead of a whole matrix; the printhead moves over the paper in a horizontal line anyway.

Because each of the printing pins can be controlled individually, it is possible to print any character, even graphics. Nevertheless, dot-matrix printers have a big disadvantage that should not go unmentioned; their noise when printing can drive you nuts.

Thermal printers

The concept of thermal printers is similar to that of dot-matrix printers. However, instead of using a complicated mechanical printing system, thermal printers work with horizontally-arranged heating elements. Naturally, it is not advisable to heat up those elements to a temperature that would make oormal paper turn black, so thermal printers print on special thermal paper. This paper develops black or lilac spots in the corresponding places, even when only moderate temperatures are applied. Unfortunately, these prints do not last for a long time; they begin to fade after a while and are very susceptible to damage. One cup of hot coffee placed on the paper completely ruins a thermal print. Theoretically, it is possible to print graphics with this type of printer, but in practice you shouldn't do this, because the thermal elements tend to heat up excessively when printing completely black areas.

Nowadays the majority of thermal printers are used for cash registers, where mechanical robustness, and above all, compactness are chief requirements. On the other hand, thermal printers do not play an important role anymore for faxing. Thermal fax paper has long since been replaced by plain fax paper which is printable with inkjet or laser printers.

Inkjet printers

Inkjet printers are another type of printer that use the principle dot-matrix printing is based on, with the addition of an actual X/Y printing matrix instead of a single band of pins. A special kind of ink is sprayed onto the paper via capillary jets and there dries very quickly due to the additives mixed into the ink.

Two main methods are being employed for spraying the ink onto the paper. One of them works with tiny thermal elements that vaporize minimal amounts of ink. The ensuing gas pressure then blows the ink via the jet onto the paper. The other method uses minute piezoelectric elements instead of thermal elements. Piezoelectric elements are special materials which undergo a change in their nozzles when a voltage is applied to them (by the way, vice versa it is possible to 'squeeze' electrons from piezoelectric elements; gas lighters with piezoelectrical gas ignitors do this). The nozzle 'extends', catapulting the ink from the jet.

And why does the ink stay in the jet instead of running out? The reason for this is that the inkjets are as fine as a hair. Due to capillary forces, the ink normally fills only the jet without running out and directly refills the jet after 'firing'. Anyone who has had a blood sugar test has seen this effect; the small glass tube draws up enough blood to fill it without any mechanical aid.

Inkjet printers suffer from one major problem. Not every sort of ink can be combined with any sort of paper. In many cases, only the use of a special type of paper considered best by the respective manufacturer will produce perfect prints. In extreme cases, if you use standard writing paper, the result will look more like a water-color painting.

Laser printers

Star Wars on your computer? No way. It might sound like fierce space battles at first, but in fact the terminology turns out to be a lie; laser printers do not print with laser beams. Those are only used to ionize determined spots of the printing drum. Due to electrostatic attraction, the finely ground toner sticks to those ionized spots, is then heated up, and is transferred to the paper. There the toner fuses to the paper after cooling off and can only be removed by a chemical reaction.

The advantages of this kind of printing are self-evident. There is hardly any mechanical strain, and the resolution is extremely good since the size of one printed dot does not depend on the size of the hammer used for printing, on the size of the thermal elements, nor on the reaction of the paper to the ink.

Exotic variants

The following special printer types will only be mentioned briefly without going further into details:

- Thermal transfer printers / Thermal sublimation printers
Strictly speaking, thermal transfer printers are wax printers. They work with a wax-like substance that is heated up and then transferred - in the past, from a specially-inked ribbon, nowadays from a storage cartridge similar to the ones inkjet printers have - onto the material which is to be printed. Unlike the aforementioned methods, here it also possible to print onto other materials, e.g., t-shirts.

- Line printers
Line printers are the fastest of all printers because they not print only one character at a time but also one complete line of characters (text, not pixels!). The sight of such a machine in full action leaves you quite astonished; the paper is practically catapulted through it. Unfortunately, the similarly astonishing consumption of electricity probably impedes private use.

- Plotters
True artists... instead of hammering single dots onto the paper, plotters have pens that write on the paper. Plotters really are vector printers; therefore, they are especially apt for printing technical drawings. These printers can also handle texts, which naturally take very long to print since every letter of a word literally has to be 'written', as in handwriting.

...and how does the data reach the printer?

In the past, all of the manufacturers had their own idea of how data transfer should work; however, nowadays the following 'dialects' are generally established:

- ESC-P (Escape P)
This format was developed in the early eighties by Epson. It has mainly been used for dot-matrix printers and similar printer types. The name was inspired by the fact that the printer commands begin with the ASCII character <ESC> ($1b).

- PCL5 (Hewlett-Packard Printer Command Language 5)
This is Hewlett-Packard's printer command language (as you can see from the name). This language was about to conquer the world of printers in the heyday of LaserJet printers. Comparable to PostScript, PCL5 is a rather abstract page-description language, and so, a relatively large memory is required in order to compute the descriptive printing information back into graphical objects.

- PostScript
This language was developed by Adobe and is a standard printer control language for all platforms. PostScript is not limited to printers; it is also applied in outrageously expensive systems as a page-description language for the display - the so-called Display PostScript.

But didn't printer paper use to have holes?!

Veterans might still remember the good, old tractor feed; sprocketed wheels engage with the holes at both sides of the paper. Th s kind of feed was perfect for continuous form paper (listings), and a real pain in the neck when printing letters, because the printer simply had no idea of where a page began or ended. Things got really amusing when one committed the fatal error of clamping the paper not straight but crooked, too hard, or not tard enough. In those cases, the printer turned immediately into a high-tech shredder. Nowadays, all printers work with a friction feed similar to the one found in typewriters. In the meantime, this technology has been improved enough to enable printers to pull in and print on the paper in straight lines. Because this was not always the case, it was easy to recognize early single sheet feeds by way of their more or less pronounced 'diagonal' printing.

So how do C64s handle the data?

You are right to ask. Commodore's favorite, the serial IEC bus, has been competely neglected by printer manufacturers. Theoretically, you have two options. One, a special interface 'interprets' Commodore's serial print protocol into one of the more common languages, e.g., ESC-P. Apart from interpreting the protocol, a serial-to-parallel conversion is executed, because very few printers offer a serial interface. The world works with parallel. The advantage of this solution is that no special printer drivers are needed; any software suitable for Commodore printers can also be combined with other standard printers. A problem of this method is that such interfaces have been out-of-production for some years now, so now you have to search for used interfaces.

The simpler solution is the second one - a special printer cable that connects the C64's user port with the parallel interface of the printer. Unfortunately, these cables need complementary software support, which older software in particular does not always offer.

What's in the future?

Laser color-printing is the future, but until now this method has remained too expensive for private use, so that today the field is dominated by inkjet printers of different types, and to a much smaller extent, by thermal transfer printess. In business use, the fusion of three separate but similar technologies can be observed at the moment. These three are printers, fax machines, and copiers/scanners. The first combinations are already on the market and have made their way into the private sector.

Letters

Uncle Wanja's Mailbox

10 PRINT "HALLO GO64! ";:GOTO 10

The first program everyone of us wrote probably looked like this, or at least very similar. But what my introduction to the first mailbox section looked like - I don't remember. Maybe I could leaf through the back issues and find out. Instead I've leafed through my mail and here are some of the letters I found ...

First, on behalf of ourselves I would like to comment on a special issue. In some letters and in the GO64! forum we have been 'reproached' for our mag often being somewhat 'late'. I would like to briefly explain the problem: The problem arises when any one of the authors, translators, proofreaders, NTSC fixers or the layout is late with his/her work - even if it is only one day - so that we don't have a fixed date for sending the magazine to the printer's. Naturally the printing office has to keep fixed dates for bigger clients than we are, so they can only print the GO64!/CW when there's enough time in between and at times we have to queue up at the end. As a result, we are again delayed for some days. But due to the amount of work for one issue it is not feasible to make up the time delay at another point. Finally, the magazines have to be packed up and brought to the post-office - if we are lucky we are just in time for the week-end, when post-offices in Germany are closed. Thus, two more days delay. In about two years there have been various occasions when we have been one day late and probably you can imagine that those days have accumulated to a fair amount. By now, we are forty days behind. And last but not least, there is one more thing that brakes us: the speed with which the post works. We are a little upset about the fact that delivery to the US takes up to one month (or even more), but unfortunately we are not in the position to change anything about that. So, in order to catch up our backlog we have decided to bring the time of going to press forward one week this year. Let's hope it works out! We certainly won't manage to do it from one day to the next, but you can always

be sure that you get as many issues as you have subscribed for - no matter if they are late or not. And now for something completely different ... our first letter to the editor:

Hi Go64!
First of all, I have to praise you for the fantastic work you are doing with your great C64 magazine in times of AMD and Pentium. I can only take my hat off to you! Keep going on like this. I hope you can continue your successful work at the GO64! in the future. The reason I started writing to you? There are so many pages on C64s in the Internet, e.g. on hardware, on software, on the installation of the C64 in a tower and much more. But very often the problem is that you just don't find the exact thing you were searching for. And so, I have a request to you ... couldn't you write an article or compose a list of the various web pages dealing with C64s, and what the exact addresses of theses pages are, what kind of information they offer (e.g. the page on C64s in a tower)? I also would be very grateful for a list of C64 suppliers with the products they offer. Thanks a lot for your time and work!
A. Niehoff

Many thanks for your praise! If it is possible, we include Internet addresses that offer further information in every article. There is a very long list of links on our homepage at http://www.go64.de. Concerning the search engines in the Internet, I have some useful tricks for you: At www.altavista.com e.g. you can put the item you are searching for in inverted commas ("C64 tower" or "C-64 tower"). That way, the search engine does not only search for the two words separately, but only for the complete expression in the inverted commas and you will get better search results. We also have links to special C64 search engines like 'Commodore FTP Search' on our list of links. Moreover, you can get a very good list of C64 suppliers (up to now, mostly German addresses) from Martin Kopetzky, Burgtreswitzer Str. 16 in D-92709 Moosbach, Germany. E-mail: <Martin.Kopetzky@t-online.de>, Tel.: +49 (0)9656 - 1772, Internet address: <http://www.home.t-online.de/home/Martin.Kopetzky/>. I hope this information is useful to you.

Dear sirs,
about 3 months ago I received my first copy of your English version of GO64! magazine. Overall, I liked the magazine very much. It had good articles and information. My major problem was some of the ads were in German or English and German. This made it hard to know the price of an item in U.S. Dollars and the advertisers address. The ads should have the prices and other information in English and not necessarily in German. Also, your magazine does not have many of the American advertisers and suppliers that Commodore World used to have for the North American readers, you should probably have these American and English advertisers listed in English in your magazine, otherwise your English speaking readers would have to get most of their supplies from European sources and have to pay possibly larger international shipping fees, international calls, etc. Also in your magazine you do not describe all the programs that you have in each cover disk. You also don't have any loading instructions with the disk. You should have more documentation for the programs on the disk. In closing, I hope the above comments might be of some help in publishing the English version of your magazine and invite you to respond to the comments I have written. Irregardless of what I have written above, I will continue to subscribe to GO64! in the future since it is still a very good magazine. Respectfully,
-Steven Klein-

Thanks for your criticism, Steven. Naturally, we have been asking ourselves what features our English-speaking readers like or do not like about our magazine. That is the reason why we wanted not only to integrate as many 'Commodore World' authors as possible in our team, but at the same time we have also been trying to get American, British and other C64 suppliers to advertise in our magazine. Of course we are dependent on the cooperation of these companies. So, if any of our readers does business concerning C64s and is interested in placing an ad in the mag, he/she is welcome to contact us! We are of course also actively searching for advertisers, but you can probably imagine that the search is not always easy. The reason for the German adverts in the English-language edition of the GO64! is simply that advertisers over here pay for the ads. What these advertisements look like is in the responsibility of the people who place them, so we have no influence on the way prices are indicated (in Deutsche Marks, dollars or Euro). - The application of the programs belonging to a certain article should be self-evident from the article itself. Nevertheless, we are trying to do better concerning the magazine disk in the future, even though unfortunately we won't be able to give a description of every program on the disk for reasons of space. As a rule, the files without description will be files that can be started with RUN and hopefully won't require further explanations.

-Uncle Wanja-

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

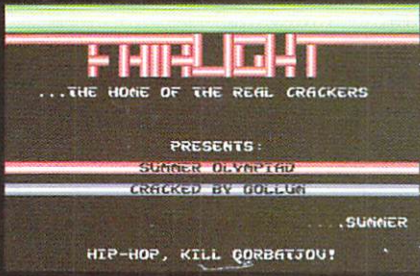
PROGRAMMING

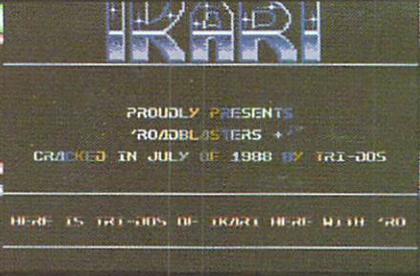
Better Working Power Assembler	$10.00
GEOBASIC (new)	$10.00
Hesware - Graphics Basic	$10.00
Introduction to BASIC/Part I	$8.00
Oxford Basic	$10.00
Oxford Pascal	$10.00
Simon's Basic	$10.00
Start Programming w/Gortek and the Microchips	$5.00

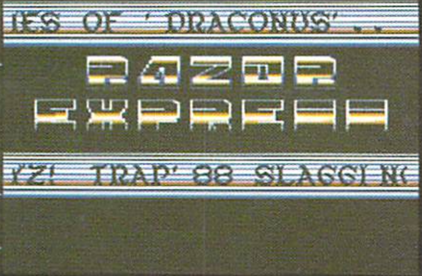
WORD PROCESSING

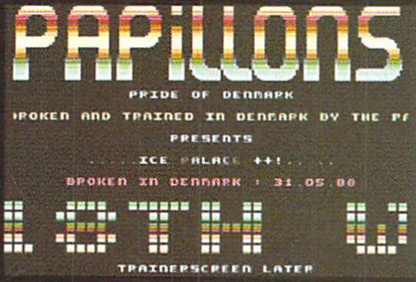
Easy Script	$5.00
Mastertypes Writer	$8.00
Mirage Concepts-Word Processor -Professional	$10.00
Outrageous Pages-NEW	$20.00
Super-Text Professional	$5.00
Supertext Word Processor	$10.00
The Printed Word	$8.00
The Write Stuff 128-NEW	$20.00
Word Writer 128	$20.00
Word Writer 6-NEW	$15.00
Wordpro 3 Plus/64-NEW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

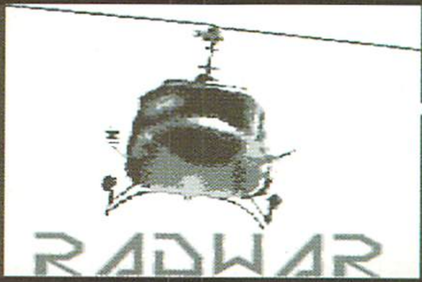
Radwar Party 2000

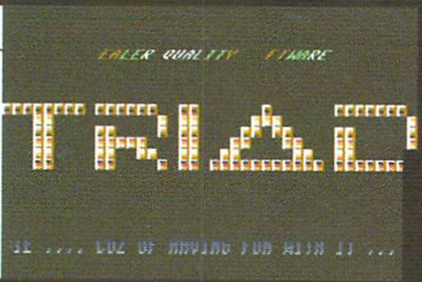

FAIRLIGHT
...THE HOME OF THE REAL CRACKERS
PRESENTS:
SUMMER OLYMPIAD
CRACKED BY GOLLUM
....SUMMER
HIP-HOP, KILL GORBATJOV!

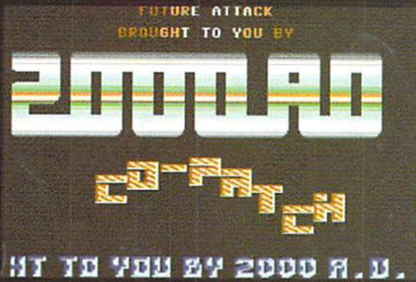

IKARI
PROUDLY PRESENTS
'ROADBLASTERS +'
CRACKED IN JULY OF 1988 BY TRI-DOS
HERE IS TRI-DOS OF IKARI HERE WITH 'RO

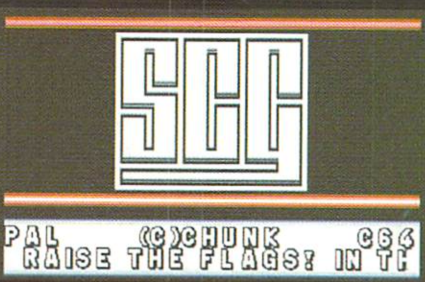

IES OF 'DRACONUS'..
R4ZOR
EXPRESS
YZ! TRAP' 88 SLAGGING

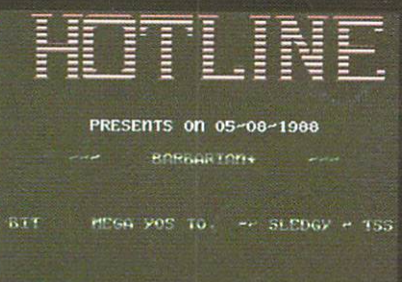

PAPILLONS
PRIDE OF DENMARK
ROKEN AND TRAINED IN DENMARK BY THE PA
PRESENTS
.....ICE PALACE ++!.....
BROKEN IN DENMARK : 31.05.88
10TH W
TRAINERSCREEN LATER

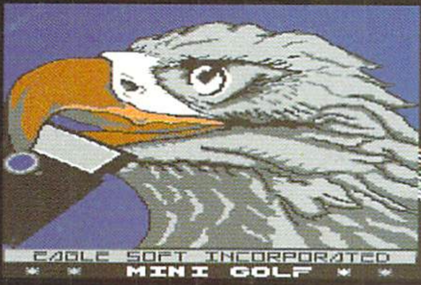


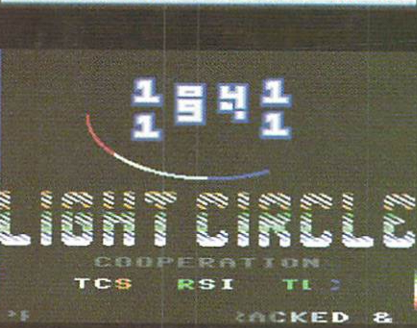

EALER QUALITY FTWARE
TRIAD
IE .... LOZ OF HAVING FUN WITH IT ...

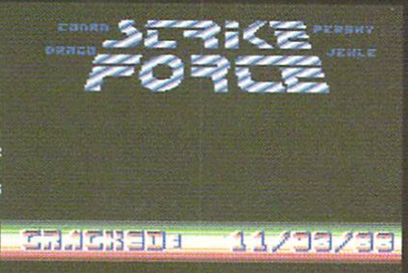

FUTURE ATTACK
BROUGHT TO YOU BY
2000AD
CO-PATCH
HT TO YOU BY 2000 A.D.

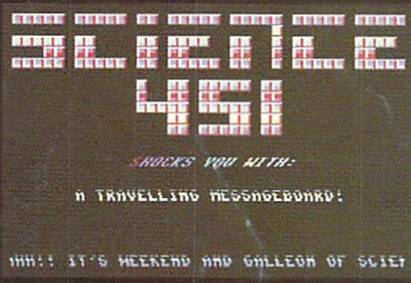

SCG
PAL (C)CHUNK C64
RAISE THE FLAGS! IN TH

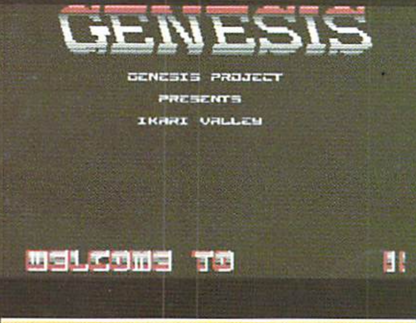

HOTLINE
PRESENTS ON 05-08-1988
--- BARBARIAN+ ---
BIT MEGA YOS TO. -- SLEDGY - TSS

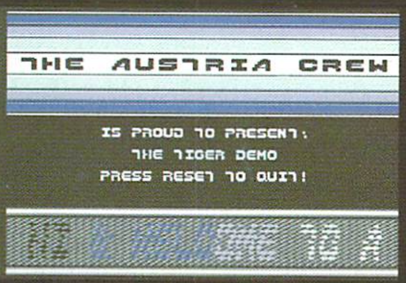

EAGLE SOFT INCORPORATED
* * MINI GOLF * *

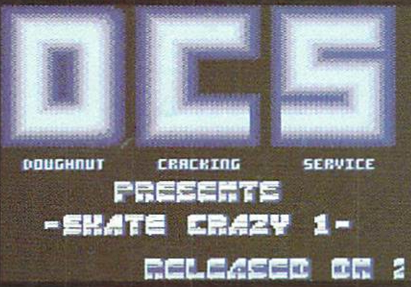

1941
LIGHT CIRCLE
COOPERATION
TCS RSI TL
ACKED &

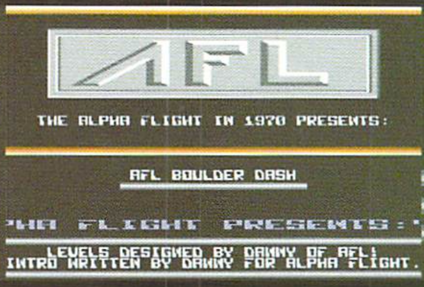

CONAN STRIKE PERSKY
DRAGO FORCE JEALE
CRACKED: 11/03/88

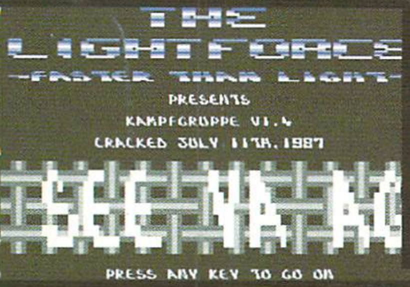

SCIENCE
451
SHOCKS YOU WITH:
A TRAVELLING MESSAGEBOARD!
HHH!! IT'S WEEKEND AND GALLEON OF SCIE

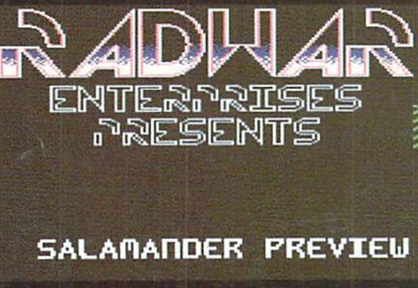

GENESIS
GENESIS PROJECT
PRESENTS
IKARI VALLEY
WELCOME TO

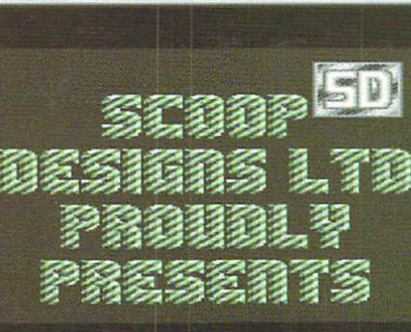

THE AUSTRIA CREW
IS PROUD TO PRESENT:
THE TIGER DEMO
PRESS RESET TO QUIT!
ME TO A

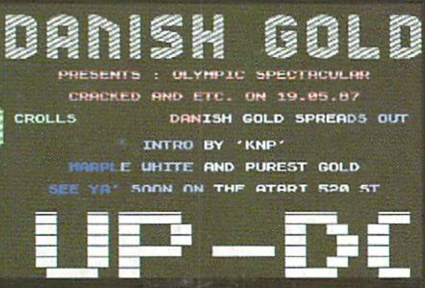

DCS
DOUGHNUT CRACKING SERVICE
PRESENTS
-SKATE CRAZY 1-
RELEASED ON 2


AFL
THE ALPHA FLIGHT IN 1970 PRESENTS:
AFL BOULDER DASH
PHA FLIGHT PRESENTS:
LEVELS DESIGNED BY DANNY OF AFL!
INTRO WRITTEN BY DANNY FOR ALPHA FLIGHT.


THE
LIGHTFORCE
-FASTER THAN LIGHT-
PRESENTS
KAMPFGRUPPE V1.4
CRACKED JULY 11TH.1987
SEE YA AG
PRESS ANY KEY TO GO ON


RADWAR
ENTERPRISES
PRESENTS
SALAMANDER PREVIEW


SD
SCOOP
DESIGNS LTD
PROUDLY
PRESENTS


DANISH GOLD
PRESENTS : OLYMPIC SPECTACULAR
CRACKED AND ETC. ON 19.05.87
CROLLS DANISH GOLD SPREADS OUT
INTRO BY 'KNP'
MARBLE WHITE AND PUREST GOLD
SEE YA' SOON ON THE ATARI 520 ST
UP-DC

Meet Them All ! 27-05-2000
www.radwar.com