Euro 5.33 USD 6.17

11/1999 The Magazine for true Computer Fans

GO64!

GO64! commodore WORLD

MUSIC

WITH THE C64 INTO THE CHARTS? -AN INTERVIEW WITH THE MUSIC-GROUP WELLE:ERDBALL

BEGINNERS COLUMN: JUST FOR STARTERS

GEOS MILLENIUM-PATCH

REPORT

GO64! ON THE FAIR IN STUTTGART

CLIPS, IDE-CONTROLLER, LUNIX NG - A UNIX ON C64 AND MUCH MORE!

DEMO-EFFECTS CODED ENTIRELY IN BASIC

HARDWARE PROJECT: A FLOPPY 1581 BUILT INTO A C128D

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

FLOPPY DISK DRIVES

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

HARD DRIVES

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

MISCELLANEOUS HARDWARE

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

UTILITIES

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

LANGUAGES & COMPILERS

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

GEOS

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

MIDI & SOUND

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

EDUCATIONAL

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

ACCELERATORS

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

RAM DEVICES

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

PRODUCTIVITY

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

TELECOMMUNICATIONS

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

GAMES

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

BOOKS

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

••• All Major Credit Cards Accepted •••

Mastercard • Visa • Discover • American Express

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/

THE HOTTEST UPGRADE EVER CREATED FOR GEOS

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

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

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

Novaterm

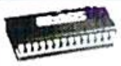
version 9.6

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

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

Just $29.95
(Specify 5.25" or 3.5" disk)

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



JiffyDOS™

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

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

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

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

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

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

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

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

Shipping and Handling Charges

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

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

* Other shipping methods available. Call for details.

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

CONTENTS

- November 1999 -

6

Trade Fair!

Picture: Volker Rust

Once more the time has come: The GO64! mag had a booth at the Hobby&Elektronik trade fair in Stuttgart, Germany. We were able to present new soft- and hardware for the C64 to 200,000 visitors. See page 6!

Picture: Welle: Erdball

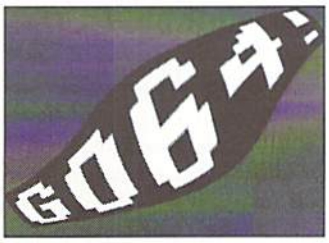
9

A Chart Position for the C64

Score a number one hit with the C64? Why, of course! The music group Welle:Erdball demonstrates how to do it. Our editorial bureau's very own Uncle got the chance for an exclusive interview. Read more on page 9!

19

Demo Effects in BASIC



Everybody knows the C64 scene's famous demo effects. Our new series will show you how to create such effects with nothing more than a bit of cunning BASIC programming. You don't always need assembly language to do great things with your C64. Starting at page 19!

Picture: Niko Malecki

27

C128D with 1581?

There's your nice compact C128D, and there's your 1581, but is there a way to integrate the two? Niko Malecki explains in his article how to get the American (metal cased) C128-D and the 1581 together into one casing. Read more on page 27!

[news]

News Flash:
New Web-Browser and more 5

Report:
The Incorrigibles 6

Interview:
Welle: Erdball 9

[tutorials]

Workshop:
How to GoDot, Part 2 13

Assembly - Language:
Part 8 of our tutorial 16

Effects in BASIC:
Lightsourced Stringrotator 19

Graphics - Tutorial:
What's in the can? 23

[geos]

GEOS - Millenium Patch:
Jan 1, 2000 12:00:01 25

[hardware]

Hardware - Forum:
The C128 DCR - a lot of hot air? 26

[cover disk]

Cover Disk:
Unzip, Wheels Update, Finder 1.1 5

[tips & tricks]

SuperCPU - Corner:
Coding in a compatible way 31

[departments]

Editorial 4
Just for Starters 22
GO64!-Bazar 26
For Free - Contact Corner 26
Masthead 26
GO64-Bazar-Form 26
Carrier Detect 29
World Watch 33

A

GFX - Workshop Files
GoDot - Module
BASIC Effects File
New Unzip
Y2k - Patches

B

Wheels Update
How To GoDot - Files

As the holidays draw near,

people start buying gifts for all their loved ones. However, there is probably one friend that we forget. That would be our Commodore computer. Wouldn't it be nice to bestow a present upon our favorite machine -- the machine that plays our games, does our work, picks up our e-mail, and cruises the World Wide Web?

C= gifts can be in any price range. It's the thought that counts. At virtually no cost, you can decorate your computer with some stickers or handmade covers for the keyboard and drives. If you fancy yourself an artist, you can give your computer a paint job! Years ago Compute!'s Gazette magazine had a competition on who could best repaint their computer, and at this year's World of Amiga show in London, the grand door prize was an Amiga 1200 computer, artfully airbrushed with a glossy black, outer space background complete with planets and nebulas.

Your Commodore would look even more decorative if you leaned a good magazine or book on it. By taking out a subscription to the GEOS Publication, GO64!/Commodore World, Commodore Scene, or Commodore Zone; or purchasing The Internet for Commodore C64/128 Users, you'd be assured of fine reading.

At the lower end of the price bracket, you could buy a disk or two from your C= club's collection of public domain disks. A step up would be to buy some used, commercial software or used, minor hardware, like a FastLoad cartridge or clone. New, unopened software is available from CMD, Centsible Software, and other retailers. Let's not forget the new software that comes from other sources, like Loadstar, Maurice Randall (Wheels 64/128, GeoFAX), Nick Rossi (Novaterm 9.6), and Arndt Dettke (GoDot), to name a few. Purchasing their products will encourage them to develop their software further.

Looking for something more substantial? If you only have one disk drive (or just a tape drive!), then buy one, two, or three extra disk drives. Provided that the drive device numbers are different, GEOS/Wheels can use the extra drives, and disk-copying becomes so much easier with more than one drive. An extra disk drive doesn't have to be of the 5 1/4 inch variety; Commodore 1581 and Creative Micro Design FD-2000's can use 3 1/2 inch disks -- the latter one using high-density disks. With the exception of the FD-2000, all of these drives can be readily found in the used market. If you really want brand, new equipment, CMD sells new 1541's and FD-2000's.

Perhaps your Commodore looks a bit lonely. Then you could buy an extra C64, C128, VIC-20, Plus/4, or PET to keep it company. The first three can easily be found in the used market, and CMD sells the C64 new. Also, CMD has new, repairable power supplies which will keep your C64/128's going much longer.

Not satisfied with small-time gifts? You can jump to the major league with the purchase of a faster, external modem; a Postscript printer, a RAMLink, a SuperCPU 64/128, the above FD-2000, or a CMD hard drive. Ranging in price from the low hundreds to the multi-hundreds, these hardware items will transform a C64 or C128 into a power machine.

O.K., there you have it. To drive home the point a little further, I've rewritten the song below to fit our Commodore lifestyle. Have fun singing it, and if you don't like the order of the items, you can switch them around. As the song stands, it would cost about $8,509 to buy all of the items! ...And that's according to prices and shipping from Creative Micro Designs, Centsible Software, Loadstar, Maurice Randall, and the used market. (My thanks to fellow Commodore user, Kevin Baysinger, for giving me inspiration and ideas to finish the lyrics. Oh, yes, I know that a FastLoad-type cartridge doesn't speed up the disk drive but the serial bus access. It just sounds better the way I wrote it.)

THE 12 DAYS OF COMMODORE X-MAS

On the first day of Christmas
I bought my Commodore
A cartridge that speeds up disk drives.

On the second day of Christmas
I bought my Commodore
Two ram expanders
And a cartridge that speeds up disk drives.

On the third day of Christmas
I bought my Commodore
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the fourth day of Christmas
I bought my Commodore
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the fifth day of Christmas
I bought my Commodore
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the sixth day of Christmas
I bought my Commodore
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the seventh day of Christmas
I bought my Commodore
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the eighth day of Christmas
I bought my Commodore
Eight 1 gig hard drives,
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the ninth day of Christmas
I bought my Commodore
Nine Loadstar diskettes,
Eight 1 gig hard drives,
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the tenth day of Christmas
I bought my Commodore
Ten sturdy joysticks,
Nine Loadstar diskettes,
Eight 1 gig hard drives,
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the eleventh day of Christmas
I bought my Commodore
Eleven RAMLinks running,
Ten sturdy joysticks,
Nine Loadstar diskettes,
Eight 1 gig hard drives,
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

On the twelfth day of Christmas
I bought my Commodore
Twelve games of action,
Eleven RAMLinks running,
Ten sturdy joysticks,
Nine Loadstar diskettes,
Eight 1 gig hard drives,
Seven speedy modems,
Six FD disk drives,
Five SCPU's,
Four megs of mem'ry,
Three Wheels upgrades,
Two ram expanders,
And a cartridge that speeds up disk drives.

Yours
Robert Bernardo
(Visalia, California)

News Flash

News for Wheels-Users

Maurice Randall has updated his WWW page. In addition to his usual features, you will also find the program "Finder" (A file-search utility for Wheels), plus new mouse drivers for Wheels 64 and Wheels 128. You can download Smart 4, Smart 4R, 128Smart 4, 128Smart 4R, C1351S, C1351SR, Toolbox 64, Dashboard 64 and Dashboard 128. The address is
http://people.delphi.com/arca93
(st)

New Web-Browser

Maurice has also been working on a new web browser for the C64 and C128. Once finished, the browser will be placed online, available for free download. Since the program is free, it may be distributed by other reasonable means as well. As of this writing, Maurice is still looking for some helping hands for this project, to allow him to finish the browser more quickly. Interested programmers, who need not be experienced in programming for GEOS and Wheels, may contact Maurice by regular mail, at:

Maurice Randall,
Click Here Software Co,
426 Sumpter St,
P.O. Box 606,
Charlotte MI 48813,
Tel: (517) 543-5202,
by e-mail at
arca93@delphi.com,
or by visiting his home page at:
http://people.delphi.com/arca93.
(st)

Competition in the Demo Scene

The initially Belgian scene group "Role" has created the "Starcompo". This demo competition is addressed to all active demo groups of the C64 scene. The demo may only consist of one file with a maximum length of 202 blocks. Those interested can send in their entries until May 25th, 2000. More than one entry per group is allowed. An independent jury, consisting of eight persons, will crown the winner, with a main prize of 100 Deutschmarks (about US $54). To get further information or to send in completed demos, you can contact Serge Engelen, Landwaartslaan 35, 3600 Genk, Belgium, or send an e-mail to **commander@role.de**.
(st)

Cover Disk

Unzip64/128 V2.14

A new unarchiving program is on it's way to the mailboxes - it's an updated version of the well known program "Unzip64" by Errol Smith. Along with the standard features, this version offers a special feature: you can now directly unpack a ZIPped d64 image file straight to a runnable 1541 disk. You don't have to worry about the name of the archive anymore. If you hit the return key instead of entering a file to unarchive, you'll get a listing of all of the zip archives in the current directory. You simply choose the desired file from here. With this program you can either unpack the archive normally or you can unpack it directly to a 1541 disk drive, something which saves a lot of time. It takes me 5 about five minutes to unpack an archive on my C128 with SCPU128, using a RAMLink as the source device and a 1571 as the destination. With a C64 you only have to wait another two minutes. *(me)*

Wheels 4.2 Update

Some small but noteworthy corrections have been released by Maurice Randall in the form of a few new files at his website. The files are available there (and of course, on our coverdisk). 'Dashboard 64', 'Dashboard 128', and 'Toolbox64' are not final releases; they're just corrected versions of the ones in Wheels64/128 V4.2. For SCPU owners there is a reworked mouse driver, which didn't work correctly in 1 MHz mode in the previous version.

Before copying the new Dashboard to the boot disk, you should make sure that point 3 of the 'boot-up settings' in the Toolbox is set; otherwise, complications can occur. In all cases you should have a copy of the master disk with the original Dashboard. I wouldn't mess with the original Wheels disk for this. The same goes for the replacement of the old Toolbox64 with the new version, which is differentiated in the infobox. A new Toolbox128 was not necessary.

To repeat, these updates are only for the Wheels Version 4.2. If you are using an earlier version you should contact Maurice to register it. With this, you can get a free upgrade to the newest version. The address is:

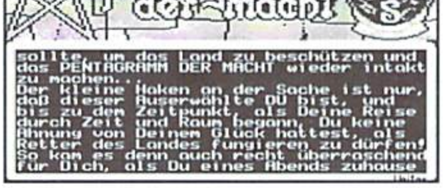
Maurice Randall,
Click Here Software Co.,
P.O. Box 606, Charlotte,
MI 48813,
USA,
Tel: 1-517-543-5202,
email: arca93@delphi.com,
website - http://people.delphi.com/arca93
(me)

Finder V1.1 (Wheels64/128)

We're sorry to say that lots of the cover disks to issue 9/1999 were delivered with a defective version of the Finder; most likely due to an error on our master copy. We'd like to apologize for any trouble this may have caused. Right after we became aware of the error, we decided to offer the program again, and of course we've taken the current version V1.1 this time. Unfortunately, it still can't copy files from different partitions in one go, so you still have to work through the partitions one at a time. *(me)*

New Byteriders adventure game: "Heroes & Cowards"

As mentioned in the February issue of GO64!, Steve Kups and Sebastian Broghammer (a.k.a. "the Byteriders") have an unreleased game in their diskbox, waiting there since 1991. We now have a test version which will be reviewed in detail in the next issue. For the moment, the game features the typical Byteriders humour and an excellent soundtrack made by Stefan Hartwig.

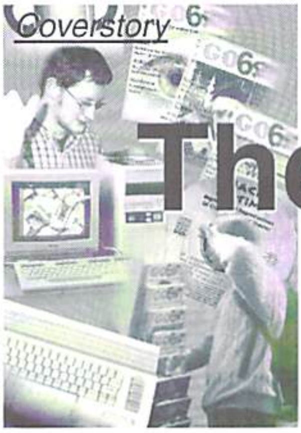


sollte, um das Land zu beschützen und
das PENTAGRAMM DER MACHT wieder intakt
zu machen..
Der kleine Haken an der Sache ist nur,
daß dieser Auserwählte DU bist, und
bis zu dem Zeitpunkt, als Deine Reise
durch Zeit und Raum begann, Du keine
Ahnung von Deinem Glück hattest, als
Retter des Landes fungieren zu dürfen!
So kam es denn auch recht überraschend
für Dich, als Du eines Abends zuhause

More about CMD in the net

Online now: "The Unofficial CMD Homepage" - **http://www.cmdweb.de**. Here you find information about the most important CMD products: The SuperCPU, the FD drives, the HD Series, JiffyDOS and of course the RAMLink. Next to the text, photos are shown to every product. Also interesting is the "About CMD" section: Finally you see the people behind the three big letters! Furthermore you will find a lot of interesting, important and partly funny information. Last not least there is a Download area in which you can find important programs for CMD devices. For CMD fans who maintain a homepage themselves, a large selection of CMD banners is available, which can be used to promote one's favourite hardware. *(Malte Mundt/ gb)*

Coverstory



Picture: Sebastian Bach

The Incorrigibles

1999 AD. The known world has submitted to the Empire of Wintel. All the world? No, a scattered band of incorrigible enthusiasts are insolent enough to yearly present their old breadboxes equipped with new hardware and software at Germany's second biggest consumer fair of more than 200,000 visitors!

by Günther Bauer

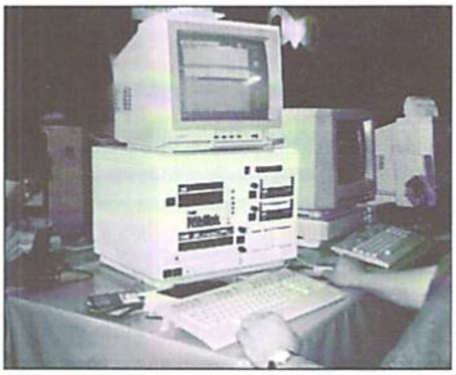
"Hey, look! It's the new movie!" A little boy together with some buddies had gotten lost in Hall 11 of the "Hobby&Elektronik" fair. He proudly pointed to a picture of a Lara Croft-styled woman. Half an hour later, a boy of about twelve years walked briskly over to exactly the same poster, his rather reluctant mother in his wake. "Look mom! That video shot of Tomb Raider looks absolutely realistic! Definitely!" Having scrutinized the picture from a few inches distance, he saw something at the next stall, and off he went. Superficial looks didn't reveal the truth. The Lara Croft poster was neither an advertisement for the new movie nor a well-rendered animation from one of the videos. In fact, it was GO64!'s cover from the August 1999 issue.

Once again, that special time had come. As in every October, GO64! took part in the Stuttgart fair with a stand of 400 sq. feet. Apart from GO64!, Oliver Biasin's Commodore Connection Line could be found in Hall 11, too. Two stands with Commodore themes in the year of 1999 at a fair with over 200,000 visitors overall. What an anachronism!

CCL

Commodore Connection Line (CCL) had a lot to offer, too, from the "Cube", a broad C128 tower; to GoDot, used for video entitling with the C64 and which was also demonstrated; to the range of C128 user software designed by Reinhard Kratzberg (MySoft). Furthermore, the KomaDat trade list of all traders who still support the C64 and a list of addresses of all registered C64/C128 fans were on sale at CCL's stand. A sure place to find flocks of people gathered around was at the computer which had been supplied by the fair; the computer had games, like "Crush" and "Bomb Mania". Markus Kanet and Manfred Grimm had come, too. They brought along the latest

Pictures (3): Günther Bauer



The Cube at the CCL-stand

Daniel Dallmann, LUnix and a null modem cord: The C64 is communicating with a PC

version of WinDesk, the new GEOS MP3 desktop. Even though it was not finished yet, the WinDesk interface was impressive in its use of context menus, taskbars, freely-selectable background pictures, lots of windows, and a very cunningly planned design.

CLiPS

The mysterious new SuperCPU operating system, CLiPS, attracted lots of amazed visitors at GO64!'s stand. Chester Kollschen, the programmer, unfortunately could not be present, but anyway CLiPS was one of the main attractions for the C64 people. Manuel Nikschas, who is going to offer his VirtualAss 16 as the development platform for CLiPS, took over Chester's place in demonstrating the new operating system to the wondering eyes of the public. Just like in the case of the alleged Lara Croft poster, we heard interesting, confused, though not-too-far-off remarks like, "GEOS for C64s; I had that, too, once". However, exclamations like, "They've got Windows 3.1" or "Look, it's an Amiga!", showed lack of knowledge concerning the basic platform and made the skin of insiders crawl.

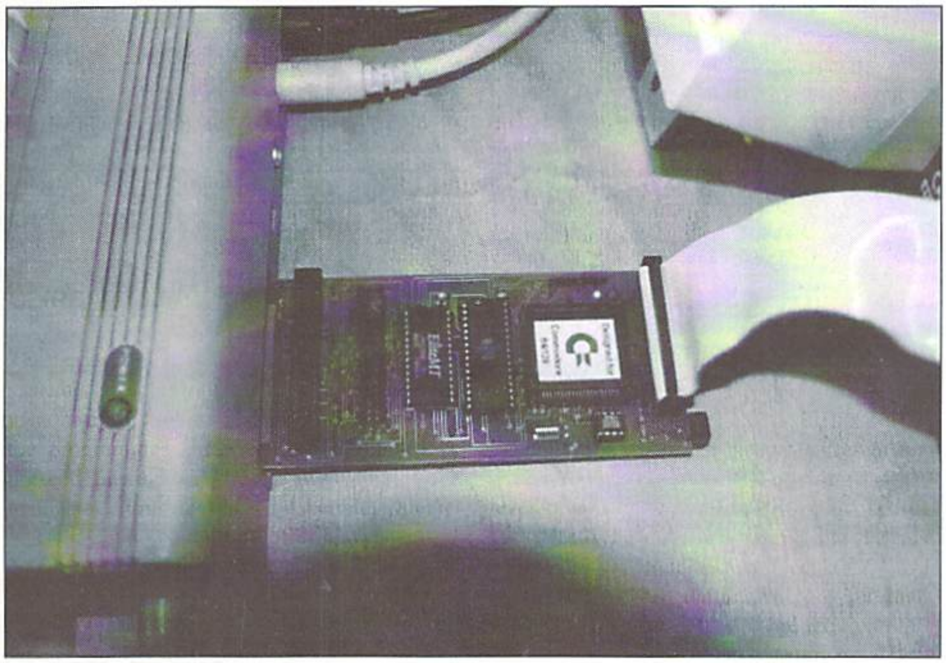
Unix for C64s

One of the guests of honor at GO64!'s stand was Daniel Dallmann. He presented LUnix Next Generation. LUnix NG is a Unix variant for C64s that offers the basic features of real Unix, although it's still in a beginning stage of development. The built-in, working TCP/IP stack, TelNet, and FTP seemed to be just waiting for an attempt to connect a C64 to a PC and access the Internet. The only problem was that the laptop Enno had brought with him didn't have a network adapter, so it couldn't be connected to the Internet line supplied by the fair. Daniel tried to install at least a small network with a null modem cord. After a few attempts to set up, he established a connection between the C64 working with LUnix and the PC's Linux. Thus, Daniel was able to impress the crowd with the demonstration of TelNet and sending data to and from by means of FTP.

Royal Visit

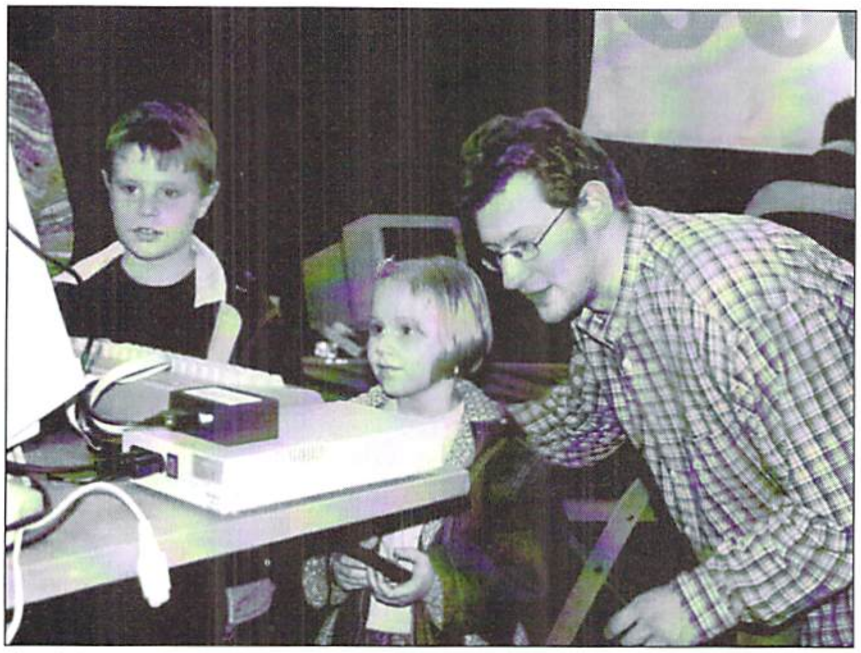
The world's greatest VIC programmer, Crossbow of the Crest group, alias Roland Toegel, took the liberty to pay a visit to GO64!'s stand in between lunch and "Star Trek: DS9" ("Of course I'm videotaping it, but there's nothing like watching it live!"). GO64!'s Mr. SuperCPU, Wolfram Sang, couldn't contain his enthusiasm, "We shook hands! We shook hands! I'm not going to wash my hand ever again!" After some technical discussion with Wolfram about the SuperCPU and VIC, Crossbow announced that he is thinking about a new graphics mode for the SuperCPU.

Jochen Adler had come to our fair stand with a C64 in his backpack, too. He demonstrated his S-JiffyDOS, which is an improved version of CMD's disk drive speeder chip. In different ways S-JiffyDOS is definitely better and faster than the original by CMD.



The IDE Controller for the C64

Mister SuperHiresIFLI: Crossbow of Crest



Uncle Wanja in a Bomb Mania duel with the younger generation

Emulators

The Commodore 64 Web.it computer with an integrated C64 emulator caused a lot of amazement because of its design, but unfortunately there wasn't much more to see due to the lack of an appropriate cord.

Many people came to the stand asking about C64 emulators for PCs. A recurring theme of the questions was how to use C64 diskettes on PCs. Many of the fair visitors were surprised by a 1541 connected with a X1541 cord to our laptop. For a small charge for our coffee cashbox, we made the construction plan available, along with the "StarCommander" software on disk. We also offered the complete version of the C64 emulator, CCS64, on CD-ROM at our stand. One particularly brazen visitor had the cheek to tell us that he had gotten the emulator on the Internet, but he didn't know how to operate it, so couldn't we give him the instructions?

IDE Hard Drives for the C64

Lots of people were fascinated by the new version of an IDE controller that was released just in time for the fair. The plug-in, expansion port card is a cheap, quality alternative to CMD's SCSI hard drive systems, even if it's not as compatible as CMD'S.

The C64 is back?

At the other side of the stand, Wolfram Sang devoted his time to playing "Creatures" in between slices of pizza. No matter if they were fans of the C64 or not, many gathered around to watch Wolfram's elegant and skilled handling of the joystick. One passer-by who noticed the flock of people stopped in his tracks and asked with an astonished face, "The C64 is back?" All in all, lots of people seemed very impressed to see a presentation of the C64 in front of so large a public gathering. An elderly man of about 60 years noticed GO64!'s stand and told his grandson: "Those are the right kind of boys!" Some of the long-lasting memories for us: People who couldn't believe that the PC64 tower is a C64 in a tower instead of a PC with a C64 emulator, those with unbelieving smiles who stood in front of the IFLI image of a Red Bull can, Playstation kids who wanted to try the "Rim Runner" game but didn't know how to use a Competition Pro joystick (which is not the same as one of those Playstation control pads), and a former Commodore employee who stood in front of GO64!'s showcase, pointing to a C64GS and exclaiming, "I helped develop that thing!" 'Til next year because in the next millennium we will still go on fighting the dark Empire of Wintel, like all brave Jedi Knights.

Adresses:

Commodore Connection Line
Oliver Biasin
Schwalbenberg 15
91220 Schnaittach
Germany
Phone/FAX +49 202 50 43 06
MySoft
Reinhard Kratzberg
Web: http://home.t-online.de/home/mysoft/mysoft.htm
KOMADat-Händerliste
Martin Kopetzky
Burgtreswitzer Str. 16
92709 Moosbach
Germany
EMail: Martin.Kopetzky@t-online.de
CLiPS
Web: http://www.clips64.de
EMail: c.kollschen@clips64.de
LUnix NG
Web: http://www.heilbronn.netsurf.de/~dallmann/lunix/lng.html
EMail: dallmann@heilbronn.netsurf.de
Commodore 64 Web.it
Web: http://www.commodore64.com
IDE-Controller
Web: http://sgi.felk.cvut.cz/~vorlicek/Ide/c64ide.html

Pictures (12): Welle: Erdball

welle: erdball interviewed

You might not believe it, but a number of musicians have found their way from the good, old C64 into the professional music business, for example, people like Thomas Detert / Xample, who successfully made it into the charts several times with his projects, "666" and "Activate", or like Andi Ypsilon from the "Fantastischen Vier" (Fantastic Four, name of a group) whose first contact with computers was a C64.

by Wanja Gayk

One day to our great surprise, we received a letter containing a CD from a group called "Welle:Erdball". They even used a C64 as an instrument on their CDs! Well, what we could hear on the CD was not at all unprofessional. But what else would you expect from a group that became number 1 on the German Alternative Charts (DAC) with their last album? That was all the reason our chief editor Enno Coners needed to ask me to meet these people.

***GO64!:** Hello and thank you for greeting us with your new CD "Der Sinn des Lebens" (The Sense of Life). What do you think of telling our readers who you are and what you are doing?*

Honey: First, a wonderful good morning to all GO64! readers. We are the presenters and employees of the imaginary radio station, Welle:Erdball. We have always hated to be declared as a band or pop group and see ourselves more as the workers of a radio station who send words, sounds, information, and entertainment to the listeners. I am mainly responsible for music, language, programming, and lyrics.

A.L.F.: My job at Welle:Erdball is more managing and planning, juggling with the money we don't have, preparing the stage performance, and handling the C64 on stage. Giving interviews is more Honey's part, and that's why I'm going to withdraw here....

```
C64: FIRST, A FRIENDLY "READY." TO ALL
THE READERS WHO HAVE MY FRIENDS (C64S)
STANDING ON THEIR TABLES AND HOPEFULLY
USE TO SUPPLY THEM WITH ENOUGH
ENERGY AND INPUT.
IN WELLE:ERDBALL I AM RESPONSIBLE FOR A
BIT OF EVERYTHING: SID SOUNDS, MIDI-
SEQUENCING, SAM OR MAGIC VOICE SPEECH,
GEOS-CORRESPONDENCY, THIS INTERVIEW,
VIC GRAPHICS...
LATELY, I HAVE EVEN STARTED WRITING
LYRICS WITH A RANDOM-TEXT PROGRAM.
```

***GO64!:** I must admit that I don't know much about this kind of music. Are there any popular groups comparable with your sound? Then our readers would get an impression of what is waiting for them on your CD.*

Honey: We produce pure synthetic, analogue, minimalistic, electronic music. Comparisons with other artists often lead to people thinking in certain ways. But you can find parallels to the music of the 80s, New Wave, or especially the "Neue Deutsche Welle" (New German Wave, German music movement), and bands like "Kraftwerk", "Grauzone", "DAF"....

***GO64!:** Well, I immediately thought of "Wolfsheim". Anyway, how did you find each other, and how did everything start?*

Honey: Because of commercial reasons, "our" music died out at the end of the 80s, and we were not quite happy with Milli Vanilli. So in 1990, we decided to make our music ourselves. We didn't have any idea how to do it at first, so we started working with our C64s that had already been filling our rooms with "Wizard of Wor" since 1984. From there it kept going on in tiny steps....

A.L.F.: Actually we met each other both playing "Wizard of Wor".

GO64!: *You've decided to write German lyrics. That is not unusual for your kind of music; I just think of groups like "Kraftwerk", "Kreidler", or others . Have you ever thought of singing in English?*

Honey: We speak German. We think, read, feel, eat, love, and live German. It would be nonsense to decimate this with a German-English translation. A good example is the English word, "steeldance". What is this supposed to be? It doesn't sound like its actual meaning. But "Stahltanz" expresses exactly what it means. Why should one sing English at all? Except for the "Scorpions", give me the name of one band that has become famous beyond the German border with English lyrics. But what about "Nena", "Falco", "Kraftwerk", "Andreas Dorau", "Einstuerzende Neubauten (Collapsing New Buildings)", "Rammstein", etc.? They all sing in German and are partly much more famous in other countries than in Germany.

GO64!: *Looking at your lyrics, I notice that they nearly always have something to do with computers or techniques on the one hand and romanticism and apocalypse on the other. How do your lyrics develop, and where do you get your ideas from? Is there anything special you want to express?*

Honey: Well, that differs from lyric to lyric. Sometimes we simply want to entertain, but mostly we try to inform the listener or to encourage him to think about certain things. Mostly, you can only achieve this if you cause anxiety. Of course, the computer revolution is the most important topic. If you don't move with the times, you have to go sometime. That does not mean that you should bring every kind of plastic rubbish into the house that destroys itself after the end of the six-months guarantee or that is already outdated today or that has never been up-to-date. And that is the essence: no Furby, no Smart, no MD player, no Tamagotchi, no PC, or other expensive scrap...what for? Better a VW Bug, a DUAL phonograph/ record player, or a COMMODORE 64. These were and are symbols for quality and perfectionism.

```
C64: IN VIEW OF SO MANY COMPLIMENTS,
I'LL MAKE A
POKE53280,2:POKE53281,2:PRINT CHR$(150)
AND TURN RED.
e,e
```

Honey: The bright future has been overtaken anyway. "The step back is the step forward!"

GO64!: *It seems that you can't forget the C64 in spite of all the possibilities that a modern recording studio offers. Why? Is it cult loyalty to the old box or its sound that keeps hold of you?*

Honey: Cult always sounds very negative. Motto: "Remember the old times... we were playing C64...". That's almost resignation. Rubbish! The C64 is not the third member of our band because it is old or because it used to be so nice, but because it is useful! Because, as you can read in the handbook (p.7), it is a "friendly computer", because it can do what it is able to do without compromises and even what it should not be able to, because it is different and has its own character (you can tell a SID sound from thousands of others), because somebody developed it with a big piece of humanity inside, because it is not a "personal computer" (PC) but a HC - a HOME computer. O.K., and now to the rational reasons. There's no good synthesizer existing anyway that can at the same time be used for playing, printing, programming, learning, painting, and drawing. My girlfriend is even writing her biology thesis in GEOS and uses an aerobic program (if anybody knows that?); that means she even does sports with the C64. But above all, a professional instrument would cost about 4000 DM. I bought my last C64 at a flea market for 10 DM, together with the standard peripherals (2 floppy drives, printer, mouse, joystick, software). So you could buy 400 C64s if you spent 4000 DM! And if you used these all at the same time - no problem in theory - you would not only beat every synthesizer or PC but have the option to rule the entire world!

C=64:

```
I JUST WANTED TO SAY THAT I'M GLAD
TO WORK FOR WELLE:ERDBALL.
```

GO64!: *I already knew your piece "Schweben, fliegen und fallen" (Floating, Flying, and Falling) before I knew your name, and it is still regularly played in discos that also sometimes play black music. I was quite surprised when I found it on the CD that some C64 freaks had sent us (some of my friends will be pretty jealous now that I have the chance to interview you). The piece seems to have become a great success. Did you ever expect that? What's the feeling like to know that half of Germany is crazy about your piece?*

Honey: No, we never expected such response. Well, how do we feel!? Maybe we are working at the right radio station. Or it's great to share our taste of music with so many people, because we are still the biggest fans of Welle:Erdball.

GO64!: *Xenia, on the disc "Tanzpalast 2000" (Dancing Palace 2000), you sing, "Bill Gates, come and f*ck me". Seriously, what would you do if he really came?*

Xenia: Good question...! First I would stop taking the pill, then give birth to a cheap PC, and pin a hefty paternity suit onto him. I would take the money to restart the production of the C64 and deliver a SCPU and a FD-2000 as standard.

GO64!: *Haha...good answer. Talking crudely from time to time seems to be typical for you anyway, at least concerning "Tanzpalast 2000". Why? Is it fun, provocation, or do you want to achieve anything in particular? Is it a way to stand out from the masses?*

```
C64: I THINK WELLE:ERDBALL HAS ITS OWN
HUMOR, AND IT'S MORE ABOUT REACHING THE
MASSES AND HOLDING UP A MIRROR TO THEM.
BUT YOU CAN'T REACH ANYBODY WITH
SELF-CRITICISM. SO YOU DO IT WITH A
CERTAIN  SARCASM OR IRONY. EVERYBODY CAN
LAUGH, EVEN IF HE ONLY NOTICES LATER
THAT HE HAS LAUGHED ABOUT HIMSELF.
```

GO64!: *Where did you get the crazy idea to cover pieces like "Robbi, Tobbi und das Fliewatueueuet" (a German puppet series), from the Christmas Carol "Alle Jahre wieder" (Returning Every Year), or the NDW hit "Fred vom Jupiter" (Fred from Jupiter) by Andreas Dorau? Don't they better belong to bad-taste parties than to your CD?*

Honey: Somehow you don't get very far nowadays without covering older pieces. Mostly, musicians choose quite good, popular pieces and don't reach half the original with their version, because it's simply difficult to make good things even better. So we prefer more unknown pieces that normally no bloody soul would cover. Then the whole thing is much more fun, and you can see that in the result.

GO64!: *Without a doubt, your roots are in the C64 community. Tell us something about that time. Who were your idols (e.g. musicians/programmers/graphics designers/ groups), what did you do, and so on.*

Honey: That sounds strange again! The good old times are not over, and for us they have never been. Of course, unfortunately, we no longer have a Chris Huelsbeck in the C64 scene, and we can guess Lucasfilm Productions won't bring Maniac Mansion II onto the market - just to mention two great things from the past. Not to forget TRIAD from Sweden, but they are still active (the MIDI-Manager is really great!), Dave Crane (Pitfall II), or what was the name of the guy from "Herz von Afrika" (Heart of Africa) and "Murder on Mississippi"? He and his games were even presented in Biggi Lechtermanns Computercorner (this used to be a German TV show) - imagine that today! Or Computerzeit (Computertime) when C. Kuersken - today he works for "Ford...die tun was!" (Ford...they get something done!) - presented the "Little Computer People" program. And today people are still talking about Furbies and Tamagotchis... Or the glorious 64'er magazine (lately, I even saw a picture of you, Mr. Gayk, in one of those old issues), or the Dynamic Duo, or when the letters G.C.S. floated along the screen in red-black-gold... Or "Creatures" from Thalamus - marvelous! (You should write an article about that!) Wonderful times, surely. But today, everything is much better, because not only can you use the old things, but you have such things like GO64! magazine, hardware specialists like N. Malecki, games like "Crazy News" or "Leo 2", hardware like the FD-2000, SCPU...and so on. And the best is that nobody can just withdraw some money from his bank account, go into the next computer shop, and buy all this, including [personal] memories. You have to search the computer flea

markets every two months for that...

GO64!: *Oh sh*t... that photo of me in the 64'er is from my Heavy-Metal times, with a PLO shawl around my neck and long hair... O.K., never mind. Apart from the studio music, does the C64 still play an active part in your lives?*

Honey: Active is not the right word. We live with the C64! The whole flat here is full of C64s, metal C128Ds, 2MB REU, 1541, 1571, FD-2000, Smart Mouse, 1351, piles of disks... We don't collect this stuff; we use it completely! We program in BASIC like world champions: games, music programs, programs to write song lyrics... Finally, we have even developed a program that uses the audio-in of the video port. Then - if you have the right wire - you can make the computer cause effects and, e.g., loop a CD player through the SID filter. We will improve that and then send it to you. Maybe you'll want to publish it?

GO64!: *Thank you for the offer; that would surely be interesting. Some of your pieces have completely been created with the help of the C64. Which software do you use, and how do you use the computer in your production? After all, the C64 naturally has a quite "dirty" sound, but you can't hear that on your CDs. I think some of our readers are interested in what other equipment you use.*

Honey: Actually the sound of the C64 is perfect as long as you keep a few things in mind. Never use the sound signal of the RF modulator but the audio-out of the video port. Never work with old, badly-shielded computers (breadbox, SX-64). There's a version (new casing) that isn't shielded with aluminum foil. It has a complete metal shield around the board that is inside. I think they are the ones you used to find at ALDI and Marktkauf markets (the last markets to sell C64s on a large scale). [These computers are] easy to identify when you have a look through the vent ducts of the plastic casing. They are the best! Finally, you could loop the whole thing through a stereo effect with a slight release sound effect in it, or you could use two computers synchronously... and there you have the best stereo sound (this sounds like TIPS & TRICKS from 64'er magazine). The Sound Monitor from C. Huelsbeck, TMS:The Music System (uses MIDI); Elektra 64, or the MidiManager from Triad are the best software. And that's incredible. Midi-Interface turns the C64 into a polyphonic, programmable synthesizer that has three voices and can be triggered with all sequencer programs by MIDI (with the screen as a "LC display"). Otherwise, we almost always use old analogue instruments, like Roland JX-3P, ORG Poly-61, KORG Poly-8000, Ensonic ESQ-1, and so on.

GO64!: *Many PC, Amiga, and Atari users, professional musicians, and people on the street laugh at the sound of the C64. Do you understand that or take it personally?*

```
C64: WELL, I DO TAKE IT PERSONALLY!
BUT JUST COME HERE... IF ANYBODY KEEPS
MAKING TROUBLE, I'LL COME OVER TO HIM,
CONNECT MYSELF TO HIS AMPLIFIER, LET
HIM FEEL MY WAVE MODE 17 TOGETHER
WITH THE RIGHT FREQUENCY, AND HIS
LOUDSPEAKERS WILL SAY GOODBYE.
THEIR DIGITAL, SOFT, MAD-DRIVING SOUND
CAN'T COMPETE WITH MINE ANYWAY!
```

GO64!: *What do other musicians say when they see or hear that you carry a Commodore 64 into the studio with you? I suppose there are some funny looks...*

Honey: Fortunately, we have our own small, broadcast, in-home studio. Otherwise, we couldn't manage it. And most of the people we know (e.g., Second Decay) know that the C64 once stood for really professional computer music and that many old synthies have the SID inside in a slightly different form. (So the SIDStation is nothing very special). Often it's funny during concerts when we show up at the promoter's or at the technician's with our C64s. Because the first, silly smiles stop when A.L.F. presses the right keys of his modified C64. (He only has the keys fastened to his arm; the board is connected to a long wire, so he can play the keyboard with the other hand - like a BORG.) Above all, we show various VIC graphics and animations on a videoscreen as a backdrop during a concert.

GO64!: *Concerning your concerts, when can you be seen live in the next months? Do you plan a bigger tour?*

A.L.F.: You can see us live more or less regularly throughout the year, for example...

29.01.2000 Markthalle / Hamburg / Germany
18.02.2000 Tower / Bremen / Germany
31.03.2000 Kollinsborg / Stockholm / Sweden
01.04.2000 Festival / Göteborg / Sweden
29.04.2000 Veranstaltungszentrum / Switzerland

...of course, everybody is welcomed!

GO64!: *Is there anything you want to tell our readers at the end of this interview?*

Honey: Actually everything's said. The most important thing is to stop complaining and thinking of the past. Make sure there will be a future instead!

A.L.F.: Besides, we would be happy if we could welcome some readers of GO64! at our concerts or at our homepage.

```
C64: DON'T FORGET! MY FRIENDS AND I
ALWAYS NEED NEW HARD- AND SOFTWARE.
IF YOU DON'T TRY AND PROGRAM YOURSELF,
THEN AT LEAST BUY SOMETHING!
```

XENIA: By the way, what's the phone number of Bill Gates...?

GO64!: *Bill Gates' phone number? No idea, maybe 1-800-SYSTEMCRASH. If I meet him, I will ask him for you. Anyway, thank you very much for the interview, and I wish you good luck in the future. May the SID be with you.*

Welle:Erdball: Thanks to you as well, and we wish you and the whole GO64! team and all readers a happy future. You can be sure about our subscription for the next millenium.

CD's by Welle:Erdball:

CD:
Der Sinn des Lebens, CD 085-62152
Alles ist möglich, CD 084-61792
Frontalaufprall, CD 084-61632
Tanzpalast 2000, CD 085-61452
Maxi:
Nyntändo-Schock, CDS 055-61613
Deine Augen/Arbeit adelt!, CDS 055-61983
W.O.L.F., CDS 055-61753

Published by:
Synthetic Symphony
SPV GmbH
POBox 72 1147
D-30531 Hannover
Germany.

Contact:
Mike Bätz & André Hesse
Gleisdammstr.3
D-96515 Sonneberg
Germany
Phone: 03675 / 829140

Radio Club Karl-Marx Stadt e.V.
Andrea Ulrich & Rolf Franke
Schüffnerstr. 11
D-09130 Chemnitz
Germany

Internet:
http://home.t-online.de/home/erdball/welle.htm
http://come.to/Mr-Ego

Picture: Sebastian Bach

by Arndt Dettke

From Multi to Hires

Converting pictures into a less powerful screen mode is difficult and sometimes just impractical. The hires mode on the C64 is at the lower end of graphical display abilities: it can't display more than two colors in an 8 by 8 pixel field. Although these two colors can freely be chosen for each field, there's an obvious problem: Let's say you'd like to display two adjacent, differently colored areas with a dividing line in a third color between them. You can only achieve this if the line runs across or vertically. Also, it has to run along the edge of a pixel field filled with the color of one area (see the image with the light blue and the two green shaded areas displaced with black).

Any image using more colors, or adversely ordered colors, (e.g. color areas with sloping edges) can be unsuitable for a conversion to hires. This conversion would result in unpleasant color artifacts, known as "clashes" (see image). You could try to post-process such images using mod.PixelEdit, or any good painting program, but this would require considerable effort, with doubious results.

So let's face it: Converting pictures to colored hires only makes sense with suitable source images.

Monochrome images

So, let's turn our attention to the only kind of conversion appropriate for the basic properties of the target display mode: monochrome hires, a single-colored image on a differently colored background. Conversions of this kind will always succeed, as GoDot offers a whole arsenal of tools and adjustments for this purpose. Even in the default configuration you can get monochrome hires images without further efforts. The appropriate windows are "Color Controls" and below, "Screen Controls".

First, we load an image:

```
Load: IFF
Load Color "HAWK2.IFF"
```

And select the following, if necessary:

```
(Screenmode:) Multi
Colors: 16
```

And finally:

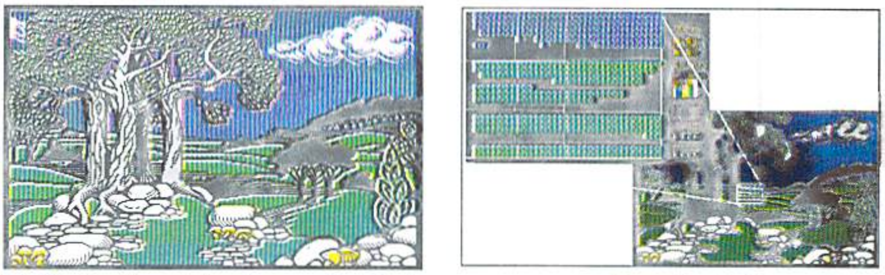
```
Display
```

This will render the image; if the screen mode were set to Hires (instead of Multi), we'd get lots of the aforementioned color clashes.

Here's how you convert to monochrome hires:

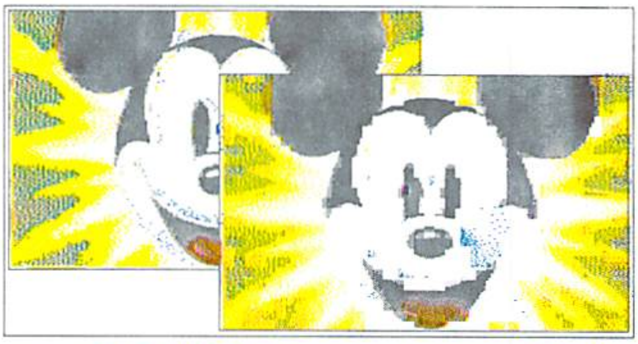
```
(Screenmode:) Hires
Colors: 2
Dith: Off
Palette Default
Accept
Display
```

The colors in the images are divided into two sets, one of them holding light blue and all brighter colors, the other containing the darker

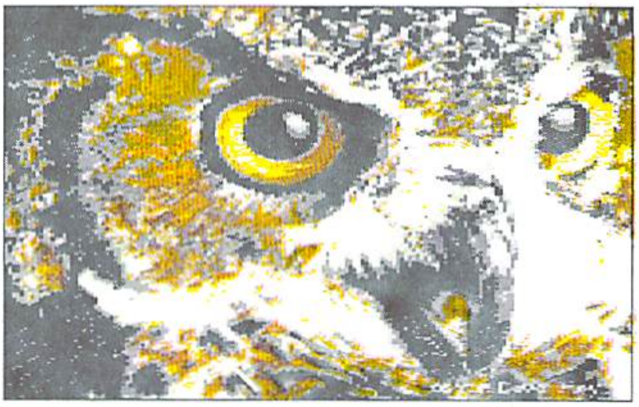


Middle Earth - a hires image. Detail: here you can see the color limits

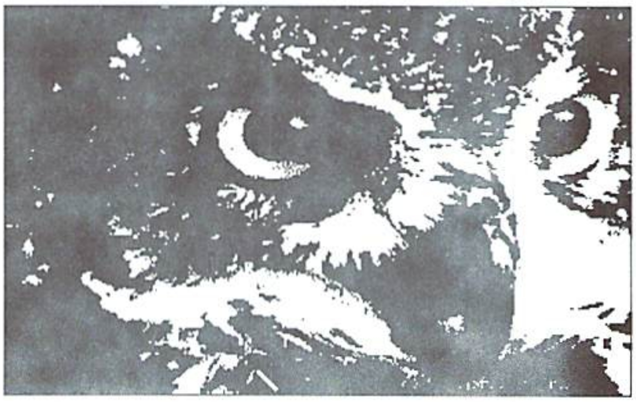
back: Mickey in multi mode, front: rendered in hires, lots of clashes



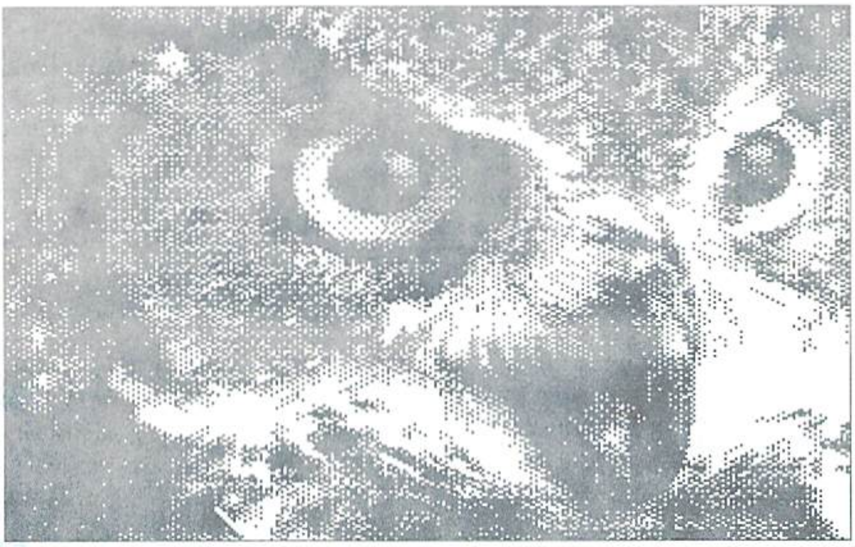
HAWK2, an image from the Amiga that we used to start from



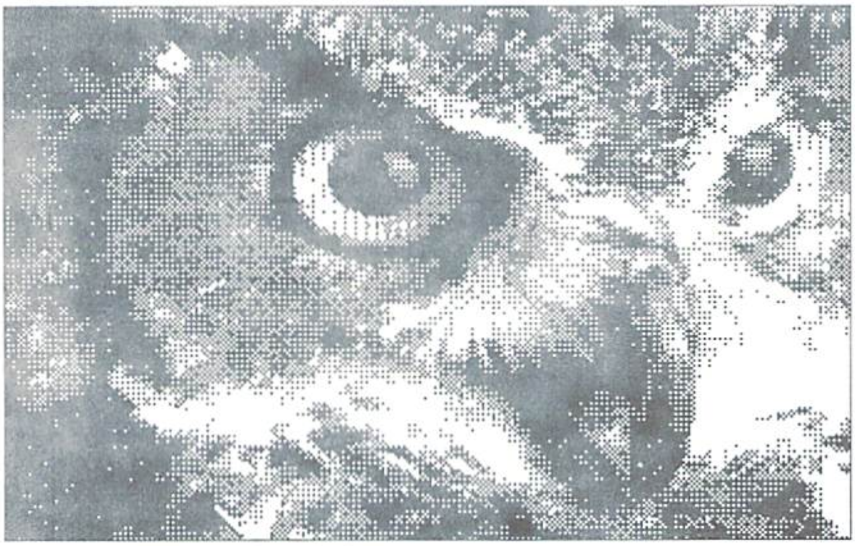
Dither: Off - an image without half tones (gray levels)



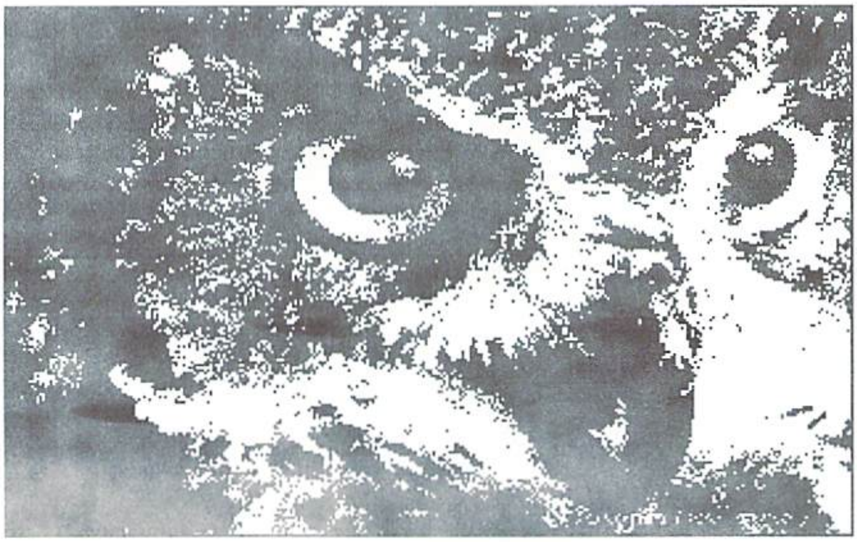
ones. The brighter colors are converted to white, the others to black. The threshold between black and white can be changed using the "Balancing - Brightness" controller. With the brightness value set to 8, even black will turn to white (the set of eight dark colors is exploited this way). And vice versa, even white will vanish in black if you shift the controller to -8.



Dither: Ordered - good for displaying on the screen



Dither: Pattern - nice for printing



Dither: Noise, Contrast: 7 - no disturbing patterns

If you find there's too much contrast, you could try to create a halftone image to your taste using GoDot's built-in dithering modules. Dithering converts the gray values assigned to the colors to differently ordered dot patterns that, overall, approximate the brightness of the original color to our eyes.

On this page you can see some examples showing the different dithering methods. The first image was created with the "Ordered" option. This method (inside GoDot) avoids patterns with vertically adjacent dots, as these may cause unwanted, yet noticeable patterns on the 64 screen.

The second method ("Pattern") orders the dots exactly in the way they shouldn't come out with the first method: this way, the image will look more regular if you print it. The pattern can be adapted to your needs using the mod.PatternEd, or exchanged with a pre-produced or self-made one.

The third option ("Noise") puts a heavy noise into the picture. The noise can be adjusted by changing the "Balancing - Contrast" setting. In the sample image, contrast was set to 7. The higher the contrast value, the lower the noise (and vice versa). The "Random" method is a special case of "Noise": using this option, only the medium gray level (the mid gray color by default) is dithered. You can select the color to be dithered by changing the "Balancing - Brightness" selection. We haven't prepared a sample picture of this method, as it's rarely used.

Changing the 4-bit data

All options described above don't change GoDot's four-bit image data; they merely affect the rendered display. This has to be considered when you save an image, since the rendered image is only used by the savers for standard C64 formats (in our case this means we're limited to using svr.Bitmap, svr.Hi-Eddi or svr.Doodle).

In order to also change the 4-bit data (with identical results on the display), we either use mod.Rendered2Raw (with one of the procedures described above), which is lossy due to its principle, or we prefer mod.ApplyDither:

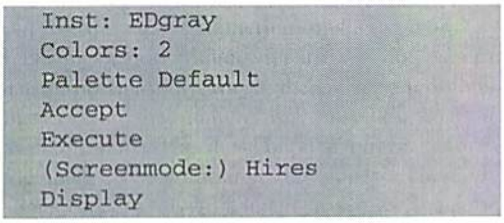
```
Screenmode:) Hires
Colors: 16
Inst: ApplyDither
Execute
```

Then, inside the module, we choose the desired dithering method, screen mode ("Hires" in this case) and number of gray tones (here: 2 for monochrome). After clicking on "Accept" the image is changed accordingly, for all savers, including others than those mentioned above. It's not necessary to render the image with "Display".

Error distribution algorithms

All of the above dithering methods work by replacing a gray value with a fixed dot pattern. That's not the only way - there exist algorithms that generate their pattern by themselves, from the available data. The most popular one is "Floyd-Steinberg", named after its developers. There are some versions of it included in GoDot, namely: mod.EDgray, mod.EDgrayMC (where MC means multicolor), mod.EDmono and two derivatives, mod.EDmono-g0.35 as well as mod.EDmono-g1.8. "ED" means "Error Diffusion" or "Error Distribution", a hint at the method used here.

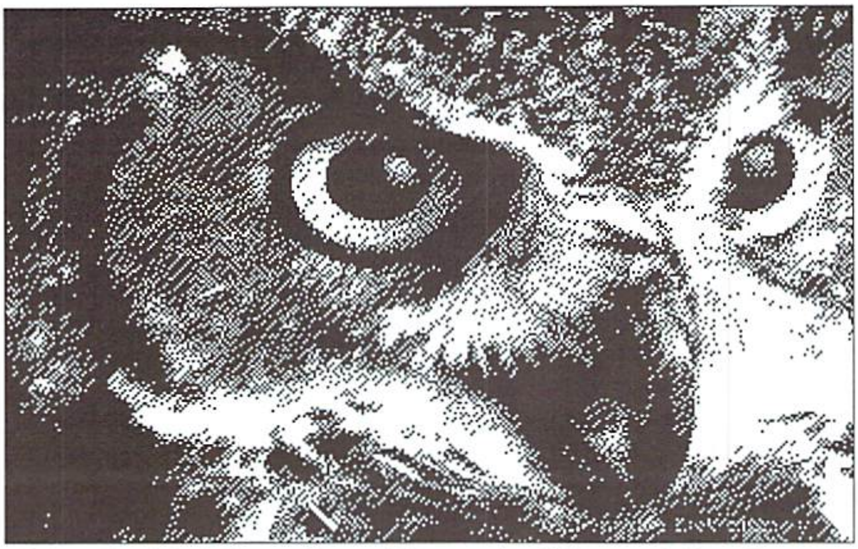
Mod.EDgray is the main module, the other three are derivatives. The version number (0.99) shows the module is still in development - it still lacks an interface which lets the user change its parameters. At the moment, this is still done in the main screen: the number of colors/gray levels the image shall be reduced to is selected beside "Colors". The colors/gray tones to be generated are defined under "Palette". Here's the procedure for monochrome images:



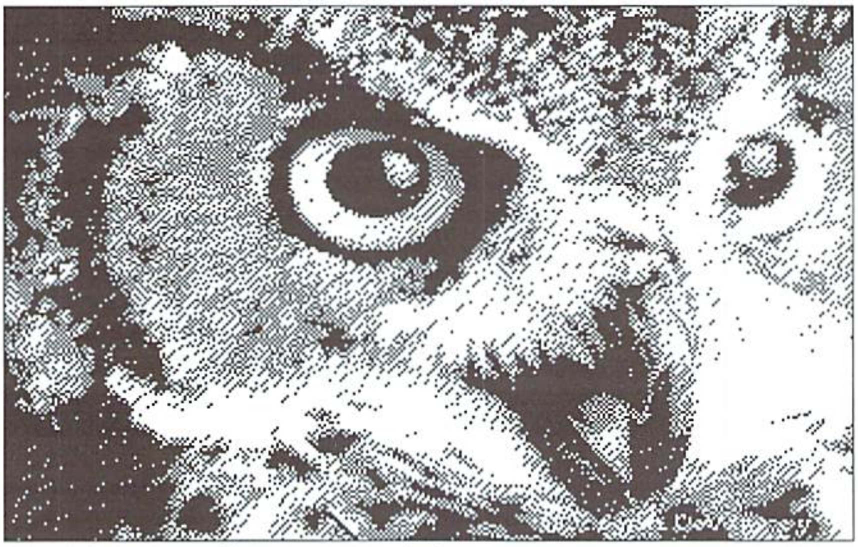
```
Inst: EDgray
Colors: 2
Palette Default
Accept
Execute
(Screenmode:) Hires
Display
```

The image produced will look like our example, dithered, but with no noticeable pattern. If you reset the number of colors to 16, you also have to adapt the palette itself accordingly (set to "Default"). Using mod.EDmono without changing any selections produces an identical result. The two other modules both affect the image's gamma curve (the distribution of brightness values in the gray spectrum). "g0.35" produces darker results (which will look better on the screen), while "g1.8" gives brighter results that will look nice when printed.

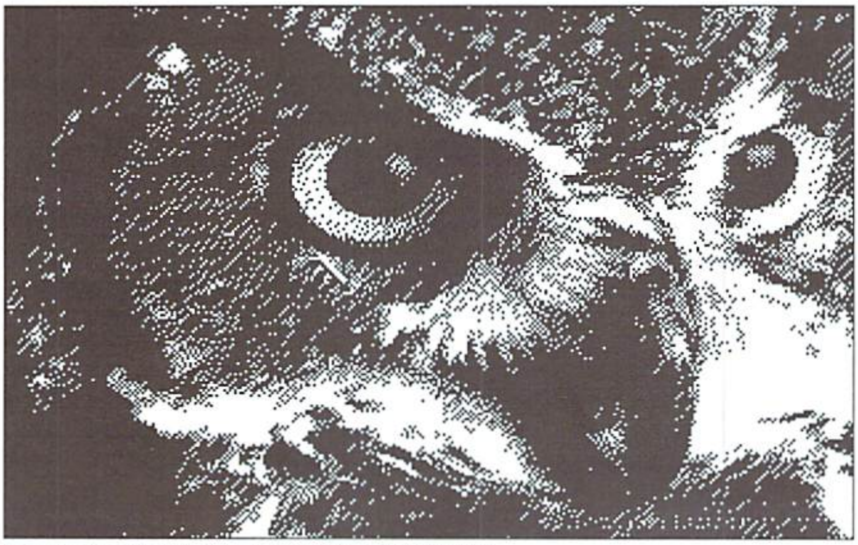
When you try it yourself, don't get confused by the names of the two modules; they were interchanged a long time ago, and never corrected! Simply rename one into the other with mod.FileCopy, and the results delivered by the modules will be consistent with the (correct) statements in this article...



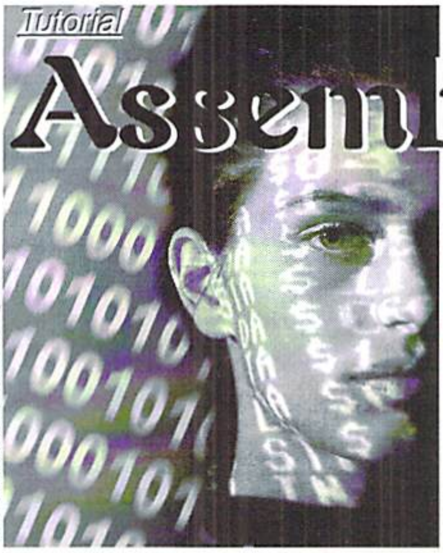
Dither: Floyd-Steinberg with mod.EDmono/mod.EDgray, Colors: 2



Floyd-Steinberg with low gamma: darker



Floyd-Steinberg with high gamma: brighter



Picture: Bernhard Pribal/Guenther Bauer (www.lichtbogen.at)

Assembly Language

I hope that I didn't scare you off totally the last time, because today we are starting nearly right where we left off. If you understood the last part well, then I don't have much to say about the theory of charsets, but I can try to bring you closer to its practical implementation.

by Wanja Gayk

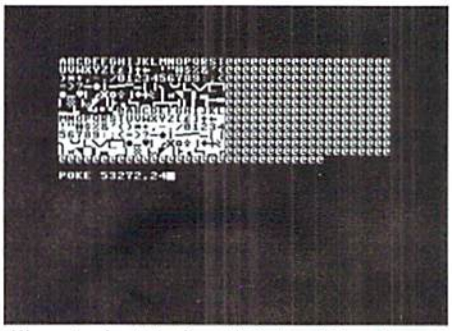
Part 8

Most assembly language courses don't say much about charsets, and if they do say something, then most of the time it is only about how you can use a charset to make the letters look a bit different from the design in the integrated ROM charset. But what I have in mind is to examine this "graphic" mode (actually, text mode) from a demo programmer's perspective, something I consider much more interesting and diverting.

Graphic Using Charsets

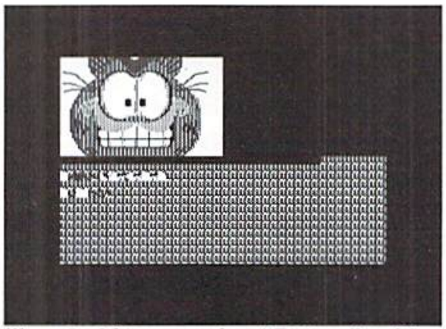
Using a changed charset, you can turn an "A" into an italic "A" or into a bold "A", but you can also give that letter a completely different shape which has nothing to do with the letter at all - somewhat similar to what the uppercase letter ROM charset does with its graphic characters, only on a much deeper level.

Let's just imagine...we draw a new charset, and in this charset an "A" wouldn't appear as an "A" but as the upper left quarter of a circle. The "B" would look like the upper right quarter. If we then were to write "AB" onto the screen and turn on the changed charset, we would see the upper half of a circle on the screen instead of the letters "A" and "B".



```
POKE 53272,24
```

Illus. 1: Just before "Return" has been pressed, you can see a bunch of characters: screencodes visible as ROM-charset



Illus. 2: After pressing "Return": with the confirmation of the poke-command to switch on our own charset the screencodes don't appear as single characters anymore. They are small graphic images giving a whole picture when put together.

In principle, every charset consists of 256 different patterns, each of which is assigned to an individual "screencode". A "POKE 1024,1" shows that the screencode 1 stands for the letter "A", if the ROM charset is activated. But if our charset now contains the pattern of a quarter circle (like in the aforementioned example) instead of the pattern of an "A" which now looks like a graphic part, you would see exactly that part of the graphic instead of "A" - if you had activated that charset, of course.

Therefore, you see that it is possible to write different screencodes to the screen and to assign each screencode in your own charset a different pattern which can (if done correctly) display a graphic, for instance, a circle, a face, or a special line of text.

The Programming

It is much simpler than you think, because it doesn't differ much from programming a normal charset. The only thing that changes is the data, that is, we change the charset. In addition to this, every charset of that kind has a correct "screen" full of screencodes to go with it. You can see an example in illus. 1 and illus. 2. The explanatory files for this are on the coverdisk ("GARFIELD.CHA" = charset, "GARFIELD.SCR" = the arrangement of screencodes on the screen).

They are located as follows - the charset has been memorized at $2000 (to $2fff), the screen at $2800 to $2bf7. Now there are two possibilities to turn charset and screen into a picture; either you copy the screencodes to $0400, which is the "normal" video RAM address and simply just turn on the charset at $2000, or you activate the charset at $2000 and relocate the video RAM to $2800. Each option can be the sensible choice. For ease of use, I present both methods:

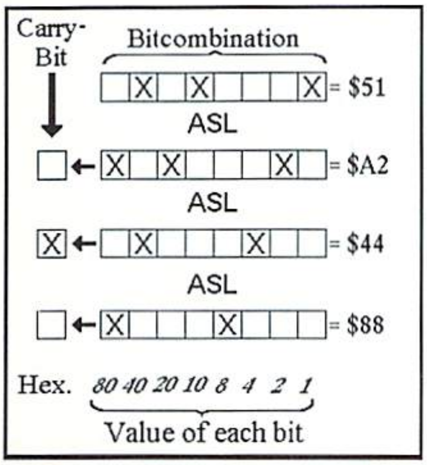
```
[Method 1]

 LDX #$00 ; copy screencodes to $0400
loop LDA $2800,X
 STA $0400,X
 LDA $2900,X
 STA $0500,X
 LDA $2A00,X
 STA $0600,X
 LDA $2B00,X
 STA $0700,X
 INX
 BNE loop
 LDA #$18 ; video RAM at $0400, charset
at $2000
 STA $D018
 RTS


[Method 2]

 LDA #$A8 ; video RAM at $2800, charset
at $2000
 STA $D018
 RTS
```

In both examples, $dd00 (basic address of the VIC memory areas) doesn't need to be



Illus. 3 The ASL-command

changed, since we remain in the preset bank 0.

Now you may wonder why the heck I showed you that seemingly complicated, first method. Well, it's possible you may need the space at $2800 where our screen data is located for something else, like sprites or a second charset. Then it is simpler to store the screen data beyond the range visible to the VIC and to copy it to the normal video RAM's position when needed there. In fact, the first question during a demo's or intro's conception is often where to store the various data without them being in each other's way. If you have video RAM at $0400-$07f7, the program code at $0fff, the music at $1000-$1fff, and the graphics at $2000-$3f3f, you only have space for three displayable sprites and must think of something else when planning a demo.

But let's get back to the assembly language course. This wouldn't be a GO64! assembly language course if we didn't have some small effects once in a while; therefore, here are some more goodies right now.

Short, effective, and simple: Animation

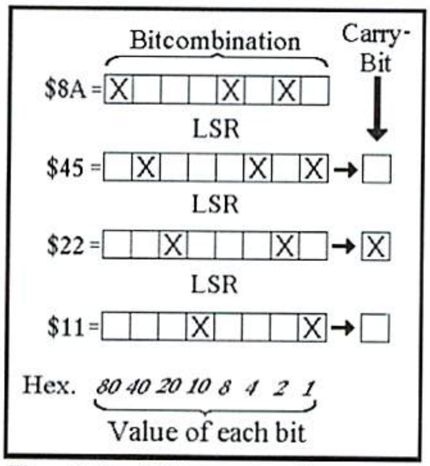
Let's say you've installed your own character set. If every screen position is occupied by a "@", for example, you will see the dots of the pattern that you chose for this "@" instead. If you understand this, then you also understand that you can change the view of large screen areas (those occupied by this "@") by simply changing a few bytes within this char's pattern (and the same for every other char). By doing so, it becomes possible to have an immense impact on the screen with relatively small effort and little computer time used.

As I already explained last time, you can easily find out which eight bytes of a charset are responsible for which character. The simplest method is perhaps to write the desired character to the upper left corner of the screen and enter "PRINT PEEK(1024)" one line below.

Example: Just press the following keys one after the other. SHIFT CLR/HOME C SHIFT RETURN PRINT PEEK(1024) RETURN. As a result, "3" is displayed, which means that the screencode 3 in the ROM charset displays a "C". The position of the eight bitmap bytes in your own charsets can be discerned by using the following simple algorithm: CHARSET'S ADDRESS+8*SCREENCODE. In our example (charset's address = $2000, dec. 8192): 8192+3*8 =8216 (hex. $2018). In a charset located at $2000, the bytes at $2018, $2019, $201A, $201B, $201C, $201D, $201E, and $201F are therefore responsible for the letter "C".

In our sample graphic (Garfield), we also have a charset at $2000, but the screen background is not filled with the normal space (screencode 32 or $20), because it has a different dot pattern in this charset. The Garfield charset uses the character "@" as the empty pattern instead. "@" has the screencode 0, and therefore, its pattern is located in the charset's first eight bytes (address+0*8 remains: address).

In the following example - an animation program - we roll a simple dot pattern from left to right through this character at the screencode 0. If you have still not understood everything, then I think you will understand everything by the time you start the program. However, this is still an assembly language course. Therefore, I am going to introduce and explain four new commands we will need right away. Also have a look at illustrations 3 to 6:



Illus. 4 The LSR-command

ASL - Arithmetic Shift Left

ASL moves all bits in a byte left by one place. At the right, a 0-bit is entered. Mathematically, the byte's value is doubled by this action. Of course, it is obvious that there is also an overflow flag if the doubling goes beyond the maximal value of a byte ($ff). The bit which is pushed out of the byte to the left (in effect the carry-over) is stored by the processor in the so called "carry-bit" or "carry flag".

LSR - Logical Shift Right

LSR is exactly the opposite action to ASL, because LSR moves all the bits in a byte to the right by one place. Mathematically, you cut a value in half by doing so (but we always round down, that is, 3 is "halved" to 1 since bytes don't have places after the point). Everything this command pushes out of the byte to the right is stored by the processor in the carry-bit.

ROL - ROtate Left

ROL is basically the same thing as ASL, because all the bits are moved to the left by one place, but we don't always insert a 0-bit to the left but rather the bit the processor has currently stored in the carry-bit. Everything pushed out of a byte to the left can be found in the carry-bit following the command's execution. With ROL you can save the carry-over when going beyond the $ff boundary to another byte, so as to enlarge the numerical range for this operation to 16 bits, 24 bits, 32 bits, 64 bits, and so on.

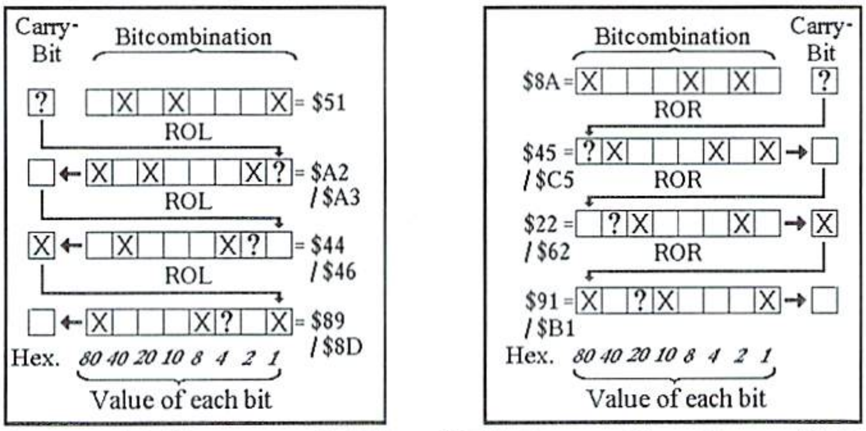
ROR - ROtate Right

ROR is the opposite of ROL, because ROR moves all the bits in a byte to the right by one place, inserts the carry-bit's content at the left, and transfers everything pushed out to the right into the processor's carry-bit, as before. This can then be stored in one or more additional bytes if need be.

The program:

```
[Listing 1]

 SEI ; bar interrupts
 LDA #$01 ; color white
 LDX #$00
adr1 STA $D800,X ; to the color RAM
 STA $D900,X
 STA $DA00,X
 STA $DB00,X
 INX
 BNE adr1
```



Illus. 5 and 6. The commands ROL and ROR: Because I haven't committed to the value of the carry-bit (1 or 0) before the first operation, I have used the question mark instead. There are two results. Depending on the status of the carry-bit before the first operation, the first result (with erased carry-bit) or the second result (with set carry-bit) is valid.

```
 LDA #$00 ; color black
 STA $D020 ; to border and
 STA $D021 ; background color
 LDA #$A8 ; screen to $2800,
 STA $D018 ; charset to $2000
 LDA #$00 ; clear unused pattern
 STA $2000 ; in charset so as to
 STA $2001 ; insure a correct
 STA $2002 ; display
 STA $2005
 STA $2006
 STA $2007
 LDA #$03 ; store bitmap %00000011
 STA $02 ; in the zero page
adr2 LDA #$FB ; wait for raster beam
adr3 CMP $D012 ; for constant timing
 BNE adr3
 LDA $02 ; load bitmap
 ASL ; leftmost bit into carry
 ROL $02 ; insert carry to the right of
the bitmap
 LDA $02 ; load moved bitmap
 STA $2003 ; and write to charset
 STA $2004
 JMP adr2 ; back to the timing
```

Optimizing the program

As printed above, the program has been written only for reasons of clarification and flexibility. Of course, it can be shortened by a few bytes if examined closely. Let's start with the color RAM. If you want to fill the entire screen with one particular color, you don't have to write a distinctive program that fills all the color RAM with that value. As long as there is nothing of importance on the screen yet, you can also use the ROM routine for clearing the screen, which is also called in the BASIC direct mode when SHIFT CLR/HOME is pressed. It automatically fills the color RAM with the current cursor color (but this does not apply to very old C64s; there the background color $d021 is used instead!).

The routine lies at $E544 and is called by JSR. Before doing so, you have to set the current cursor color, stored by the C64 at $0286 (or the background color at $d021 in ancient C64s). If we set our colors to have the foreground white and the background black, we can take advantage of black having the value 0 and white the value 1. We can then use the x-register in order to set the background color, and after that, simply increase the x-value to 1 (white) by using INX instead of setting it again, and thus we have saved a byte. The background color, black, has the value 0 - the same value used to clear the parts of the graphic block which is supposed to be empty. So why set the accumulator again? We can simply continue to use that value while it is in the x-register anyway, saving another two bytes.

In the loop which moves the dots in the graphic block, we fetch the contents of the zeropage address $02 twice into the accumulator every time we loop through, because the accumulator gets erased in the course of the delay loop that is responsible for timing. We can prevent this by simply using the x-register in the waiting loop, thereby not constantly changing the contents of the accumulator. Doing this, we have saved another two bytes, speeding the program up by three cycles as a positive side effect. Here now is the shortened and optimized program:

```
[Listing 1b]

 SEI ; bar interrupts
 LDX #$00 ; color black (X=0)
 STX $D020 ; as border and
 STX $D021 ; background color
 STX $2000 ; clear unused pattern
 STX $2001 ; in charset so as to
 STX $2002 ; ensure a correct
 STX $2005 ; display
 STX $2006
 STX $2007
 INX ; color white (X=1)
 STX $0286 ; as current cursor color
 JSR $E544 ; ROM routine "clear screen"
 LDA #$A8 ; screen to $2800,
 STA $D018 ; charset to $2000
 LDA #$03 ; bitmap %00000011
 STA $02 ; store in zero page
adr2 LDX #$FB ; wait for raster beam
adr3 CPX $D012 ; for constant timing
 BNE adr3
 ASL ; leftmost bit into carry
 ROL $02 ; insert carry to the right of
the bitmap
 LDA $02 ; load moved bitmap
 STA $2003 ; and write to charset
 STA $2004
 JMP adr2 ; back to the timing
```

But once optimized, you really quickly find a problem - the program becomes less flexible. While you were able to play around with the colors as you wished in the first, longer program, you have to stick to the chosen colors in the second one. Therefore, you ought to make a habit of optimizing a program only then, when the final function and final appearance have been defined. Those programming with a machine language monitor have a much harder time than those working with an assembler development program (Professional Assembler, Turboassembler, AssBlaster, Virtual Ass, ACME, and so on).

To finish, here are another three listings whose function you should try to discern on your own for training purposes. Next time we will continue with graphics.

```
[Listing 2]

 SEI
 LDX #$00
 STX $D020
 STX $D021
 STX $2000
 STX $2001
 STX $2002
 STX $2005
 STX $2006
 STX $2007
 INX
 STX $0286
 JSR $E544
 LDA #$A8
 STA $D018
 LDA #$03
 STA $02
adr2 LDX #$FB
adr3 CPX $D012
 BNE adr3
 LSR
 ROR $02
 LDA $02
 STA $2003
 STA $2004
 JMP adr2

[Listing 3]

 SEI
```

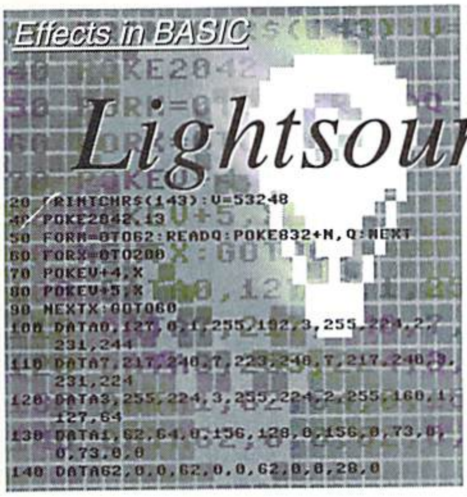
```
LDX #$00
STX $D020
STX $D021
STX $2000
STX $2001
STX $2002
STX $2005
STX $2006
STX $2007
INX
STX $0286
JSR $E544
LDA #$A8
STA $D018
LDA #$03
STA $2003
STA $2004
adr2 LDX #$FB
adr3 CPX $D012
 BNE adr3
 LDY $2000
 LDX #$00
adr4 LDA $2001,X
 STA $2000,X
```

```
INX
CPX #$07
BNE adr4
STY $2007
JMP adr2
```

[Listing 4]

```
SEI
LDX #$00
STX $D020
STX $D021
STX $2000
STX $2001
STX $2002
STX $2005
STX $2006
STX $2007
INX
STX $0286
JSR $E544
LDA #$A8
STA $D018
LDA #$03
```

```
 STA $2003
 STA $2004
adr2 LDX #$FB
adr3 CPX $D012
 BNE adr3
 LDY $2007
 LDX #$07
adr4 LDA $1FFF,X
 STA $2000,X
 DEX
 BNE adr4
 STY $2000
 JMP adr2
```



```
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
```

Lightsourced Stringrotator

If you have the knowledge, the computer can be used efficiently with BASIC. The following examples are easily understood if you know a bit about the C64, essentially understand BASIC 2.0, and show a bit of interest. Let's see what can really be done with our computer.

by Marcus Hinzmann alias Silver Fox/Satovia/ Sanity

Who hasn't admired this before?! In an assembler demo or a good game, some text is written on the screen, which is illuminated from right to left and back again as if by a flashlight. Letters in the middle of the circle of light are clearly readable, but towards the edge, the letters are illuminated more and more weakly until they cannot be seen at all in the depths of the color black. Now this illumination swings char by char from one border to the other. Something that looks this good can actually be realized in only six BASIC lines!

Memory Address 648 ($0288)

In order to understand this effect, we now go a bit beyond common basic knowledge. Let's have a look at memo.y address 648 first. Usually, video RAM starts at 1024 and ends at memory address 2023. This means that you have 1000, accessible, independent memory addresses. This number can also be confirmed by a simple calculation. The C64's screen has a raster of 25 lines with 40 chars each in text mode. Thus, 25 times 40 equals 1000.

And how does the starting address of 1024 come into being? By reading the usual content of memory address 648 (which is responsible for this), you get to know why. You find the value 4 in it, the number of the C64 memory page for the video RAM start. One page has 256 bytes, and 4 times 256 equals 1024. However, now it should also be possible to simply change this value. Indeed, it is.

One thing needs to be considered when doing this. Please don't ever change this memory address by "POKEing" in the direct mode. For instance, if you enter POKE 648,160, the video RAM doesn't start at 1024 (4x256) but rather starts at 40960 (160x256, $a000). But without any further preparations, there would be chaos, and the cursor would disappear. This indicates that you need to adjust more C64 parameters if you wish to work on other screens and to be able to see something.

Memory Address 53272 ($d018)

Memory address 53272 is responsible for the char set used and for the location of the video RAM. Therefore, the value contained is divided in two. All even numbers less than 16 influence the starting address of the char set (by the way, the C64 automatically adds a 1, but this is ignored), values added to this starting with 16 (in steps of 16 up to 240) define - just like memory address 648 - the video RAM's location. In order to calculate the final address, you multiply the values based on 16 by 64, whereas the values below 16 are multiplied by 1024. It has to be noted that this calculation's result is based on the currently chosen VIC area. If anything else but the normal one (bank 0, $0000 to $3fff) is activated, you also need to add its starting address as well. For instance, if you were to load a char set to 8192 ($2000 in bank 0), it would suffice to write 24 to this memory address (16+8 - the video RAM is at 64*16, the char set at 8*1024) in order to see the new char set on the screen. This memory address is also important for our effect. After having changed 648, you also need to adjust something in 53272 in order to see the screen.

Memory Address 56576 ($dd00)

Divided into two parts, this memory address defines the already mentioned VIC area. To the constant value of 148, you add the selection values of 0 through 3, with the highest value (3) selecting the lowest bank (0). Each bank contains 16384 bytes (16 KB). In order to be able to work on our switched screen from the example above (POKE648,160), you need to POKE 148+1 (equaling 149, for the VIC bank from $8000 to $bfff) to this memory address. While you work in normal mode (video RAM at 1024 in bank 0), this memory address contains 148+3 (equals 151).

All Memory Addresses in Practice

One thing should be said beforehand. The color RAM starting at 55296 ($d800) can't be moved. Thus, no matter which color you write to a switched screen, colors that are to be displayed on the screen later on can also be seen at the same location on the switched screen. Thus, the last screen color where a color had been "PRINTed" to has priority.

An example:

```
1 POKE 648,160: POKE 53272,128+4:
POKE 56576,148+1
```

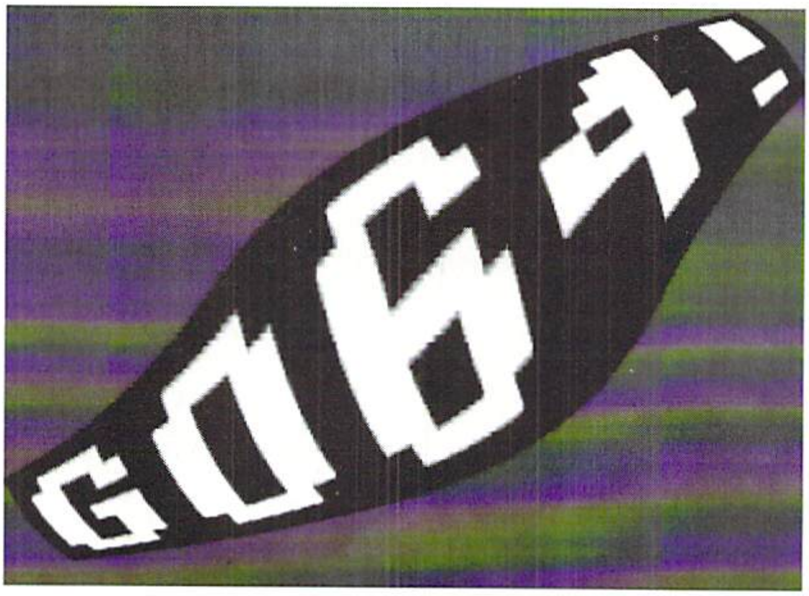
This switches the screen to 40960 ($a000) using 160, sets the start of the screen to $a000 (with 128*64 equaling 8192, this being $2000 higher than the VIC bank's start), and brings the char set to $9000 (with 4*1024 equaling 4096). The built-in ROM character set is usually 4096 ($1000) higher than the start of the VIC bank. The last BASIC command on the given line just chooses VIC bank 2 by using a parameter of 148+1.

```
2 PRINT "(Shift+CLR/
Home)(white)test"
```

(Writing the text, "test", in white paint)

```
3 POKE 648,4
```

Returning the video RAM at 1024 (4*256) but due to the change at 53272, you see the screen starting at 40960.



```
4 PRINT "(Shift+CLR/Home)(red)c-64"
```

The text "c-64" is written in a red color onto the screen at 1024, which is the one now chosen.

If you execute this program, you see the text written, starting at 40960 ("test") on the screen, but it appears in red. Why? At the same location (column 0/line 0), another color is displayed on the second screen. However, the colors are valid for all screens. Thus, the color last displayed is always the color seen. This is not the case with the text. This remains unchanged on its corresponding screen.

Memory Address 53266 ($d012)

This location provides the position of the raster beam the VIC is drawing on the monitor right now. By constantly waiting for one and the same value, the program achieves a small but effective synchronization delay that is even independent of the computer's speed (it takes the same time on a SuperCPU as on a normal C64).

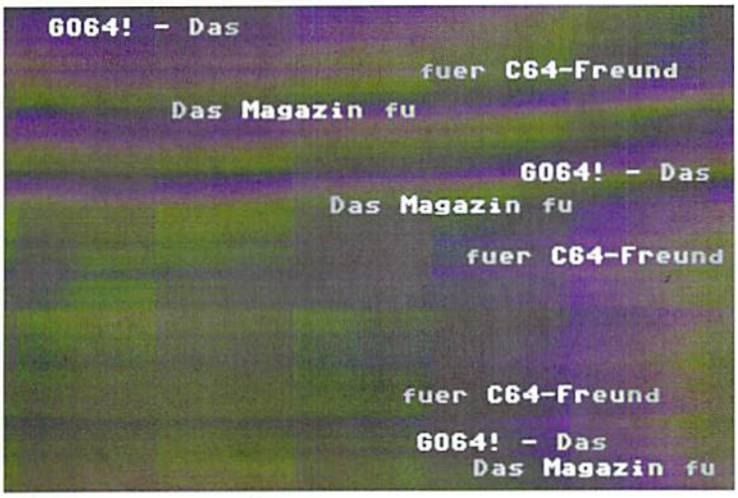
Analysing the Effect

Our flashlight trick is based on a quite simple idea. You have the possibility to switch the text screen, as discussed before. You write your text onto a secondary screen. Now you return to the normal screen, but 53272 is not changed, and thus you still see the initial screen. The colors now "PRINTed" to the normal screen can be seen through the other one's characters. Thus, you deal with two layers, one for the colors and one for the characters. The characters of one layer glow in the color of the other. By doing so, you may PRINT the colors on the color layer at a tremendous speed without erasing the characters on the text layer!

Analysing the Program

Roughly put, our little program, which has been designed as a subroutine, does the following. At first it computes a value for centering the string C$ (in line 1000). Then it defines the two strings A$ and B$ that will later be used for the color display (1001 and 1002), clears screen number one (the color screen), and sets border and background color (addresses 53280 and 53281 - see info box) to black (still 1002). Next it switches to the second screen's text layer, clears the screen there, and displays the string C$ in the center. Then the C64 is informed where the second screen (the one for the text) is to be found so that it can also be displayed (1003). Finally, we jump back to our color layer (screen 1). There the routine PRINTs the color strings, which have been structured in such a way that a swinging movement is created five times (1004 and 1005). In an execution delay, we wait for the raster beam at position 128 in address 53266. Once everything has been done, things will be reset to the normal mode, and we return to the main program (1005).

And now more precisely, we are writing a subroutine starting at line 1000 that will be accessed with the string C$ containing our text. The string may not be longer than 38 chars. In the first line (1000), a formula calculates the number of spaces required to center the string and then stores that number in the variable M. Then we define the following variables:



G064! - Das
fuer C64-Freund
Das Magazin fu
G064! - Das
Das Magazin fu
fuer C64-Freund
fuer C64-Freund
G064! - Das
Das Magazin fu

W as 53266	(Checking the raster beam)
S as 53272	(Address for char set switching and video RAM location)
V as 56576	(Address for switching the VIC bank)
D as 53280	(Border color)
F as 53281	(Background color)
R as 648	(Address for switching the video RAM)

In line 1001 string B$ is defined. Of course, the light source is nothing but a string of colored chars that is later moved in a looped char by char from the left to the right. In this string there are at first 17 cursor movements to the left, followed by a color progression consisting of 16 (arbitrarily chosen) characters. With 17 movements left and 16 movements right, one movement to the left is left over. If this string is PRINTed repeatedly, it continously moves to the left by one char, and with it, the color progression.

In line 1002 we define another string A$, this first displaying 16 colored characters and then moving 15 times to the left. By repeatedly PRINTing this string, we attain a constant movement to the right by one char. Following this definition, we POKE the screen (first background color, then border color) to become black and clear it.

Line 1003 now brings the address 648 into play (see above). At first, things are switched to screen page 160 at 648. This new memory for the screen needs to be cleared first. There we write the number of spaces as stored in M before, and we display our text in string C$ after having switched to lower case (by control code CTRL - n). Then the routine sets 53272 to 134 (128+6), POKEs 149 (148+1) to 56576 (in order to allow for working on the switched screen), and finally resets the screen to 1024 for displaying the colors. The readable screen (the one with our C$ text) remains at 40960 due to the values in 53272 and 56576 (see explanations at the article's beginning.)

In line 1004 we clear the color RAM and position the color string by using control codes. In addition, some loops are defined here with the counter H for the frequency of the swinging motion from left to right and back again, and T, as the number of PRINTs, is executed in which the color string moves by one char per update. At first A$ is displayed 50 times, and due to the WAIT command, the routine waits for position 128 of the raster beam every time, which creates a tiny delay. Thus, the entire affair moves 50 times to the right.

In line 1005 this procedure is repeated, but this time we don't display A$ but rather B$. The result is movement to the left for the color progression. The second NEXT in that line increases the counter H until it reaches the value 5, that is, until the loop has been executed five times. At the end, the char set is reset with the value 21 (16+4 plus an automatic 1) to the normal mode, and the work in VIC bank 2 is finished (POKE 56576,148+3). The last thing is to leave the subroutine by using RETURN.

Here is now the complete source code in BASIC 2.0 form (when entering the lines, please write consecutively. For greater clarity, the listing below has been formatted to break lines up into several lines.)

```
1000 m=((40-len(c$))/2):
 w=53266:
 s=53272:
 v=56576:
 d=53280:
 f=53281:
 r=648

1001
b$="(17xleft)(orange)**(1.red)**(yellow)
**(white)***(yellow)**(1.red)

1002
a$="(orange)**(1.red)**(yellow)**(white)
***(yellow)**(1.red)
 POKE f,0:
 POKE d,0:
 PRINT "(shift+clr/home)"

1003 POKE r,160:
 PRINT"(clr/
home)(7xdown)(black)ctrl+n)"SPC(m)c$:
 POKE s,134:
 POKE v,149:
 POKE r,4

1004 PRINT "(shift+clr/
home)(7xdown)(8xleft)";
 FOR h=1 TO 5:
 FOR t=1 TO 50:
 wait w,128:
 PRINT a$;:
 NEXT

1005 FOR t=1 TO 50:
 WAIT w,128:
 PRINT b$;:
 NEXT:
 NEXT:
 POKE s,21:
 POKE v,151:
 RETURN
```

Was this hard to understand? Constant testing, retrying, and rereading of this program/article should help. Someday you should be able to insert the following routine into a program:

```
c$="GO64! The true computer magazine":
GOSUB 1000
```

Lower case and upper case are always available, so let's make use of them. Those who would like to see this effect in a larger context should get my demo INDEPENDENT as soon as possible. There they will find it at work.

Have fun trying things out! Best wishes!
Marcus Hinzman alias Silver Fox *(ad)*

Overview of the mentioned addresses

648 - screen memory start (page)
53265 - position of the raster beam
53272 - addresses for screen memory and characterset
53280 - frame color
53281 - background color
56576 - VIC-bank-control

Column

Your Commodore is a valuable resource. Take care of your investment with a few, simple safety and survival tips.

Just for Starters

Survival School

by Jason Compton

Save your joystick ports

You probably know already that the exposed chips on the inside of a computer are "electrostatically sensitive"; they are very susceptible to damage from even minor electric shocks, which is why you are generally not supposed to touch the metal pins on chips. But did you realize that the metal pins on the joystick port are nearly as sensitive, especially when the computer is on?

The joystick ports speak to the 64's "CIA" chips which control the joystick and keyboard. When certain areas of a keyboard stop working or certain directions on all of your joysticks stop working, it's generally because a CIA chip has gone bad. Sometimes, these things just happen, but a fair amount of the time the cause is a shock that came through those joystick ports. The most common cause is reaching across those ports in order to flip the power switch or (on a 128) to press the reset button.

On some revisions of the 64 and 128, you can replace the CIA chips fairly easily. But why run the risk? Here are some easy tips to help you stay out of trouble:

1. Always keep joysticks in the joystick ports. Sounds simple, doesn't it? You may not think about it, for example, if you use GEOS extensively, which generally only uses one port. But by leaving two joysticks plugged in at all times, you eliminate the hazard.

2. Don't unplug/change the joysticks while the computer is on. This one is almost as simple; if you leave two joysticks plugged in all the time, there shouldn't be a real problem. But you may have a favorite joystick for a particular game, or suspect that one isn't working and want to make a quick change. The short answer - don't do it. When you change joysticks, you run the risk of causing a short, either with the joystick itself or with your fingers as you feel for the port. The worst offenders are joysticks or other devices that have a metal hood around their connectors; they're not very common, but that metal hood can cause a short if it touches the pins on the computer. So if you need to make a joystick change, turn the computer off.

3. Don't use joypads for other systems unless you're very, very sure they work on the 64. Joypads for the Sega Genesis/Megadrive and Amiga may have the same 9-pin connector, and in fact they are electrically similar, but they're by no means identical. The signals sent by those gamepads are somewhat different in order to accommodate the extra buttons, and the results on the 64 aren't pretty. At best, they may not work very well, and at worst, those different signals could damage the computer, especially the aforementioned CIA chips. Some people have published specifications that help you disable the extra electrical signals (and the extra buttons as a result) on Sega gamepads, but it's easiest to just stick with joysticks designed for the Atari/Commodore standard.

In an upcoming GO64!, we'll talk about giving a 64 keyboard a proper cleaning. Commodore World readers will remember from Issue 22 that a quick, non-intrusive technique is to take a stiff piece of paper like a business card and run it through the slats in the keyboard; you can dig out a lot of dirt, hair, pet fur, cigarette ash, etc. that way.

But the fun part is that you may not need to clean that keyboard at all, if you can find a clean keyboard to be a parts donor. Although the Commodore 64's innards changed from time to time and the keyboard changed in appearance somewhat from revision to revision, physically and electrically, all Commodore 64 keyboards are almost exactly the same, meaning you can interchange them if you have a "spare parts" computer. The process is very simple.

Remove all of the screws from the bottom of your computer (three on most models) and remove the top of the shell. You should be able to lift the keyboard and move it a little bit to reveal the metal shield underneath. Don't remove the metal shield unless you have to.

You'll see a large bundle of wires running from the keyboard to a single, long plastic plug into the computer motherboard. Note the orientation (remember which colored wire faces which side), and unplug the keyboard (carefully grip the keyboard connector with needlenose pliers and pull.)

Plug the new keyboard into the now-empty spot on the motherboard, being sure the connector is aligned the same way as the original keyboard.

Replace the top shell of the computer and replace the screws in the bottom of the computer.

That's all!

The original Commodore 64 power supply was one of the worst pieces of electronics ever devised. If you are still using one, you may want to consider replacing it with a heavy-duty power supply, preferably a "repairable" model so you can change fuses in the event of a problem.

Aside from being underpowered and inefficient, many original power supplies actually "leak" a damaging level of current into the 64 when it is plugged in but turned off. Over time, this can damage the 64's chips and ultimately make the computer unusable. So if you're stuck with the original power supply, completely remove it from the wall socket when it is not in use.

Now that Commodore World and GO64! are together, it's more important than ever that we hear from you, the reader, and find out what topics you want to learn more about. Please write to jcompton@xnet.com and share your thoughts and questions so we can cover them in upcoming issues. *(ak)*

Graphics - Course

What's in the can?

Part 2

The favorite on the monitor

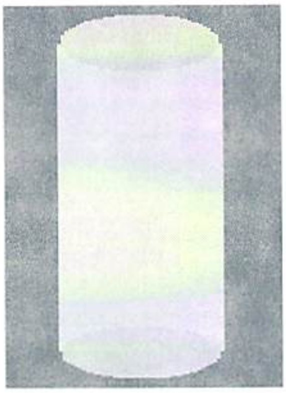


Image 1: It all begins with a rectangle and two ellipses

Last time we looked at the different graphics formats and their peculiarities. Today we practice. How do you proceed to draw an image of an object?

by Alexander "Mr. Quark" Teufel and Volker Rust

Well, first you'll have to consider what object you want. If you already have an image in mind, you might want to start drawing immediately, but since starting this way requires a lot of graphic imagination, it's difficult for beginners. It's much easier if you've got a model for orientation. In the case of a physical object, like a teddy bear, you can simply pose it. Take care it's well-lit and easy to see and compare with the image on the screen. If you want to have light and shadow effects in the picture, it's particularly helpful to light the model appropriately, and you'll likely get a more realistic result. However, if you don't have a model, it's recommended that you make a sketch on paper.

Which graphics mode is the most suitable?

Once you know what to draw, you'll have to decide which graphics mode offers the best conditions for your image. For the beer can which we're trying to create, we've chosen to use multicolor mode. This mode is the most suitable to begin your pixeling practice, because the low resolution of 160x200 pixels allows for quick results, and because it offers enough colors so you can learn how to handle them.

Which graphics program?

Next step - get your favorite image processor. It's extremely important to be familiar with the software and know its functions. If you haven't made yourself acquainted with it yet and tried out its options, drawing will be hard and time-consuming, and you won't get the result you desire. Although it barely takes any more than ten minutes to get familiar with a program, most artists fail at this point, because they want to begin drawing without knowing the software they're using. By the way, we used Amica Paint as a multicolor painting program to create our example pic. "Color-X" (included on this issues's coverdisk) is another graphics program we consider suitable for beginners.

Strategy

First try to draw the contours of your object. For irregular lines or curves you may use the line-drawing function and outline the object line by line. While you do this, it's best to imagine the object as a flat, uni-colored area. If it has regular proportions, use the circle, ellipse, and rectangle-drawing functions. The color should be similar to that of the actual object but with a medium brightness (you'll need brighter and darker tones to make light and shadows). Choosing the right color should not be too hard, since the 16 colors the C64 can display don't offer much choice. The next step is to fill the enclosed area you drew (see picture 1).

From contours to details

You can now start adding details; in the example picture, we begin with the upper lid of the can. First, the inner area is marked off with a second ellipse. Then the outer form of the object is roughly outlined (picture 2). After that, we enter zoom mode for the fine work. At this point we recognize the importance of light and shadow and the effects they can produce. Matching colors are best found out by trial-and-error; our color table gives you some orientation. For the label of the can, gray tones are most suitable. Some color combinations melt together on thescreen and appear as a single different color if you use them in a checkered pattern. This can sometimes work miracles, given the low color abilities of the C64 (picture 3).

Color limitations and clashes

Watch out when you want to use more than the three colors (plus background color) that are allowed in each field of 4 by 8 pixels; you'll encounter the so-called color clashes. This happens when you try to use a fifth color, although it's impossible for the hardware to display it in this pixel field. Painting programs react differently to this situation; some of them replace one of the colors used in the pixel field with the currently selected color, while others simply won't let you select a new color. At any rate, color clashes are always annoying, because you can't draw as you'd like to and you're forced to find a way around.

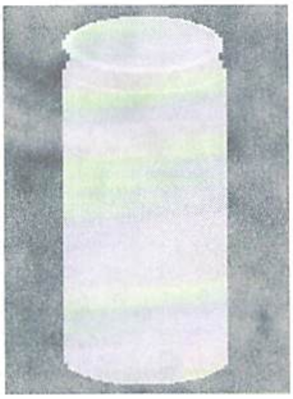
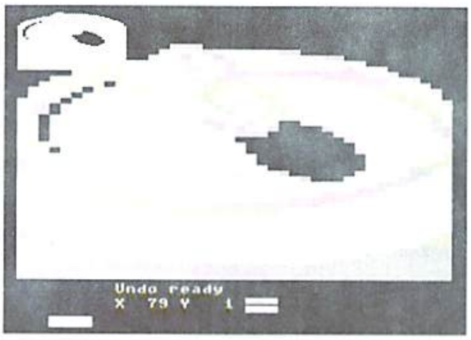


Image 2: The basic form of the can



Undo ready
X 79 Y 1

Image 3: Using brighter and darker colors for a more plastic result

This example shows that it's important to think in terms of the computer's capabilities; if you are aware of the edges of the 4x8 tiles and adjust the image to them, you'll be able to use three new colors in each tile, which guarantees an optimal color density.

Writing can be tricky

Now, we have some problems with the writing which flows from top to bottom. It's best to prepare it in the second graphics screen - if there is one - and insert it onto the can with the copying function (picture 4). Before drawing any letters, you should know exactly where the writing is to be placed and divide it properly so all letters fit in. If you want an exact equivalent to the original writing, you'll have a lot of real pixel work (i.e., you'll have to set a lot of single pixels), constantly comparing it to the real beer can.

When copying the writing onto the can, take care that the writing fits into the same tile limits as on the screen it was drawn in; otherwise, you'll likely generate a color clash, which will take unnecessary time to fix.

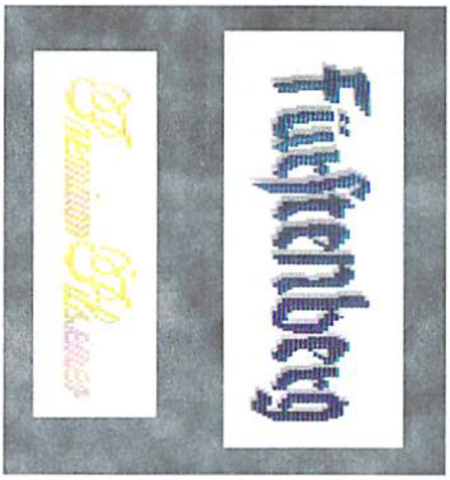


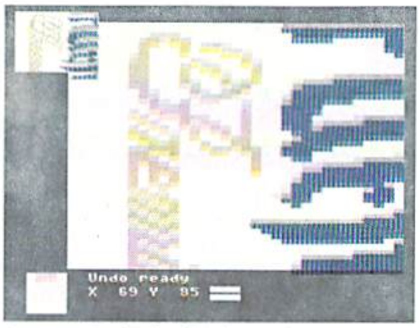
Image 4: The writing is pre-drawn in its own graphics screen

Anti-aliasing

Once you've overcome this hurdle, you'll have quite some detail work left over. The most important task might be trying to "round off" the corners and pixel staircases caused by the low resolution. A corner in front of a black background, like in the example pic, can be rounded off by setting a darker pixel exactly at the point where the edges meet. This method is known as anti-aliasing; it lets you create a soft transition between zones that have a large difference in brightness, and this reduces the ugly sharp edges to a minimum or even makes them invisible.

On white or similarly bright backgrounds, the same principle works conversely; the brightness of the color you use for anti-aliasing has to be between those of the edge to be rounded off and its background. Use of this technique can even give the viewer the impression of a higher resolution, because the sharp edges expected in a multicolor picture aren't visible (picture 5).

On our beer can, the blue writing that looked rather edgy at first was rounded off with dark gray pixels; also the label appears rounder than normally possible in multicolor mode.



Undo ready
X 69 Y 85

Image 5: Anti-aliasing can work wonders with the resolution

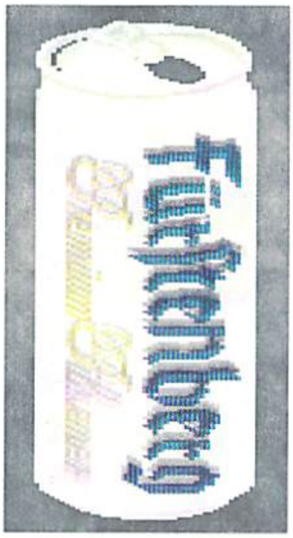


Image 6: Finally! This is ready to show off!

It's up to the painter's patience and ability to finish up the picture. Little details, like the edge and bottom of the can (picture 6), are usually added after everything else is done. During the final pixeling, you'll reach a point where you can't visibly improve it anymore by setting further pixels. At this point, you can be sure it's done... at last!

Hints from the pros

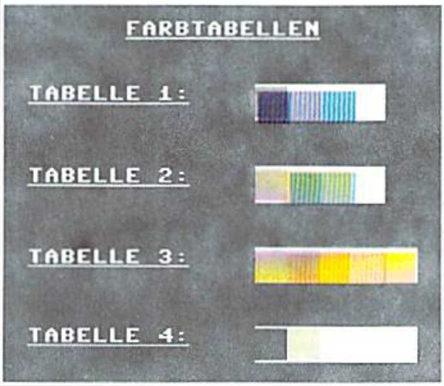
André Seeliger, better known in the scene under his handle, "Brainsmasher", talks about his newest logo he did for the group, "Axicon":

"Such images, as my logo shown here, can hardly be done in one go. They take quite some time. So when you work on such a pic, for heaven's sake, take all the time you need. If you happen to lose any motivation to continue the picture, save your unfinished work and get at it again a few days later."

"Of course, you get the most experience if you try out lots of ideas. For example, try drawing a simple image in several color combinations and find out which colors fit together well and which don't."

Overview: the most common color combinations

1: blue, light blue, cyan, white
2: dark brown, green, light green, white
3: dark brown, red, brown, pink, yellow
4: black, dark gray, medium gray, light gray, white



FARBTABELLEN

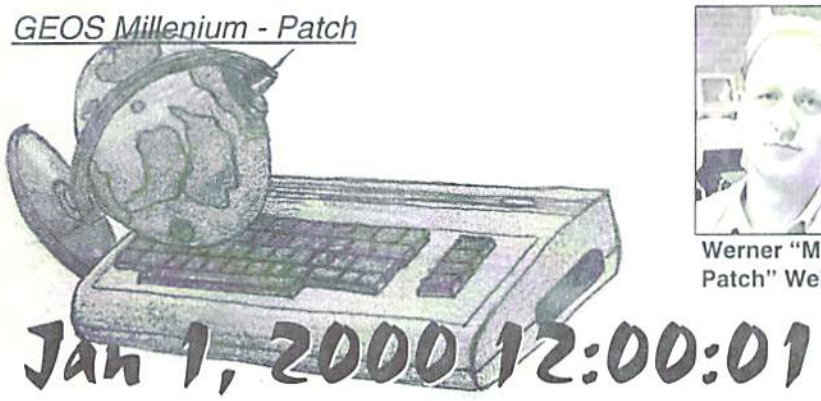
TABELLE 1:

TABELLE 2:

TABELLE 3:

TABELLE 4:

GEOS Millenium - Patch



Werner "Mr. Patch" Weicht

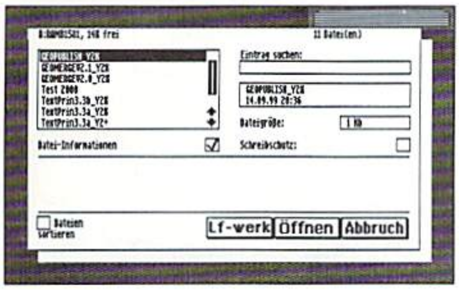
Burning toasters ... computers that won't start ... exploding monitors ... is anything going to be spared the horror of the change of year from 1999 to 2000...

by Nikolaus Metz

Luckily, we Commodore 64/128 users are safe - but we do have to pay attention to a few small things. As the GEOS developers were hard at work creating a revolutionary product for the C64, no one had the notion that they would still be in use in the year 2000.

Necessary Changes

Until a few short years ago, it was common programming practice to record the date as only the last two digits. The fact that this would have a boomerang effect was first seen with the use of the program 'Updater', used to save data from a RAM disk to a diskette or hard drive. The 'Updater' actually had problems with the year '1999' and they had to be corrected. Additional problems arise with applications which contain the date function: geoWrite, geoMerge, geoPublish, TextPrint and GeosLQ. In order to save valuable memory, '19' was automatically prepended to the user-entered last two digits of the year. To make sure that everything will work correctly in the new year, we have the program PatchSystem V1.2.


B:RAMD1581, 148 frei

11 Datei(en)

GEOPUBLISH_Y2K
GEOMERGEV2.1_Y2K
GEOMERGEV2.0_Y2K
Test 2000
TextPrin3.3b_Y2K
TextPrin3.3a_Y2K
TextPrin3.3a_Y2+

Eintrag suchen:

GEOPUBLISH_Y2K
14.09.99 20:36

Dateigröße: 1 Kb

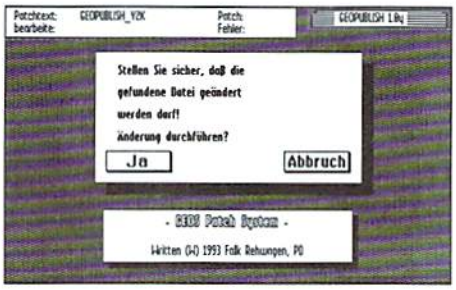
Datei-Informationen ☑

Schreibschutz: □

□ Dateien sortieren

The Corrections

The patch text is available in US and German versions for the programs geoWrite 64 and 128 V2.1/V2.2, geoMerge V2.0/V2.1 from the DeskPack, GeosLQ V2.16/V2.17/V2.18c (German) and V2.5 (US), various TextPrint versions, Updater V2.0/V2.1 and for the Calendar from DeskPack. The programs are changed thusly: the '19' is replaced by '20' in the date function. This simple solution has a small disadvantage, however. After using the patch, the programs may only be started on 1/1/2000 or later, otherwise the date will read '2099'. Additionally, the program TextPrint must be altered in the year 2010, due to another date entry error.


Patchtext: GEOPUBLISH_Y2K
bearbeite:

Patch:
Fehler:

GEOPUBLISH 1.0y

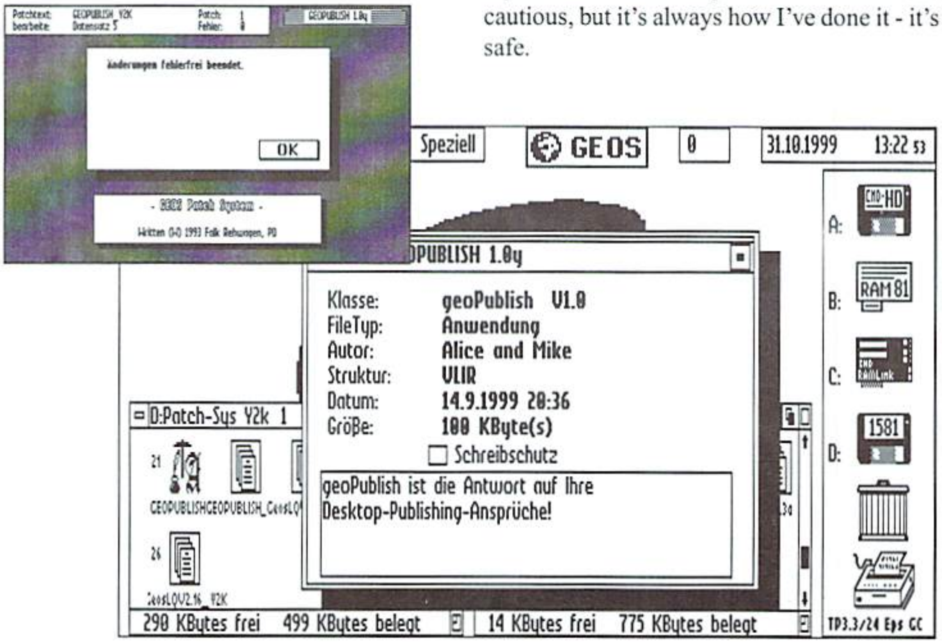
Stellen Sie sicher, daß die gefundene Datei geändert werden darf!
Änderung durchführen?

- GEOS Patch System -

Written (c) 1993 Falk Rehwagen, PO

Get Ready for the Patch

First off, I have to warn you to never use the Patchtext on the original software diskettes. I always have a 1581 disk drive on hand for the purpose of copying a program and fixing it with Patchtext. I do the actual patching on my RAM disk and then change the name of the patched program so that I can tell the difference between files. The patched program is then saved to a diskette with a report file. This may sound a litte too cautious, but it's always how I've done it - it's safe.


Patchtext: GEOPUBLISH_Y2K
bearbeite: Datensatz 5

Patch: 1
Fehler: 0

GEOPUBLISH 1.0y

Änderungen fehlerfrei beendet.

- GEOS Patch System -

Written (c) 1993 Falk Rehwagen, PO

GEOPUBLISH 1.0y

Klasse: geoPublish V1.0
FileTyp: Anwendung
Autor: Alice and Mike
Struktur: VLIR
Datum: 14.9.1999 20:36
Größe: 100 KByte(s)
□ Schreibschutz

geoPublish ist die Antwort auf Ihre Desktop-Publishing-Ansprüche!

D:Patch-Sys Y2k 1

290 KBytes frei 499 KBytes belegt

14 KBytes frei 775 KBytes belegt

The Public Domain Idea

The Text for Patchsystem V1.2 (and on) are located on Werner Weicht's homepage (http://home.t-online.de/home/wweicht) and in mailboxes with CBM and Fido Net association. Since not all of our readers have access to the Internet, this program can also be found on our cover disk. If you use the program, please send the small shareware fee of $5 US to make sure that the author can continue work on this program in the future. If there's a program for which you'd like to see a patch, contact Werner Weicht and discuss the possibilities.

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: Gaelyne R. Gasson, Markus Hinymann, Malte Mundt, Maurice Randall, Errol Smith, Alexander Teufel, Werner Weicht

Translations: Bettina Beck, Matt Bodnarchuk, Stephan Lesch, Ranjith Ekanayake Mudiyanselage, Kendra Thiemann; **NTSC-Fixing:** Wolfram Sang; **Supervisor:** Arndt Dettke

International Supervisor: Robert Bernardo, Nate Dannenberg, Gaelyne R. Gasson

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, photos by *vr* and *gb*

Graphics: Christoph Bach, Sebastian Bach, Günther Bauer, Daniel Rac

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

Publication: Monthly (12 issues per year).

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

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

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

Circulation Manager: Enno Coners

1999 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

Do you want to exchange data between your C64 and your PC? It is easy to do with the X1541/XE1541 cable, which is connected between the PC printer port and the C64 drive. +49 (0) 234/332887 (Stefan)

Who is visiting bazars and is finding me some electronic games? I will pay for every single game. I will send you my search list with pictures. Tel. +49 (0) 2831/2273

For sale: C64, C64C, monitor 1901, printer Seikosha SL80VC, 1541 I+II, 1581, REU, 1541, 1571, Handyscanner by Scanntronik +49 (0) 7486/963636

For sale: JiffyDOS C64/128/1541, original games, PD, cartridges, books (German), C64, 1541 I+II, printer, datasette, C16-original software, Action Replay MK VI Tel. +49 (0) 202/504306

For sale: C128 programmers. guide, $50. Newsmaker 128, Home Designer CAD 128, Magic Draw 64, $10ea. Cobal 128, $15. James Lambert, 1121 Williams Ave. James Lambert, Hastings, NE 68901-3820 USA

For free - contact corner

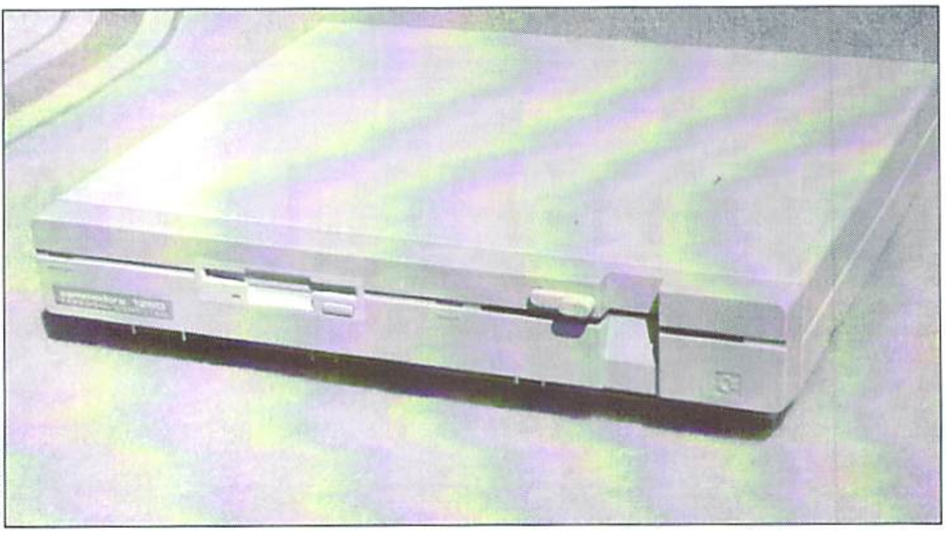
I am in search for a software-swap for C64/C128 and C16/116/+4 Tel +40 (0) 202/504306

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.

The C128 DCR - a lot of hot air?

Of course not - but there's surely enough air inside. If the casing were a food packaging and the inner parts were food, Commodore would almost certainly have gotten into trouble, since in the food branch, the contents are to fill at least 80% of the total space, leaving empty no more than 20%. The 128DCR shows the contrary ratio - if we open the casing, we see mostly an empty space with some electronics in it.



Pictures (4): Nikolaus Malecki

by Niko Malecki

New ideas

This sight gives rise to new ideas in the head of any enthusiastic hobbyist. Wouldn't a RAM expansion fit in there just great, leaving the expansion port free? All we need to do is solder an additional connector to the inner pins of the expansion port! And now, the most important idea: Although the C128DCR has a built-in 5.25" drive (1571), any professional user of the C128 will certainly need a 3.5" drive. Usually, he'll put it somewhere next to the computer, but neither does this set-up look aesthetic, nor the cables attached to it.

With such large hollow space inside the computer, what's more suggesting than putting the 3.5" floppy into it. In issue 2/99 you can see a photo showing my computer with a 1581 floppy inside. This conversion is a fairly easy one, and in this article I'll show how it is done.

At first, we open the housing of our 1581 to take out the 3.5" Floppy and the board, just to notice that the drive has no front panel of its own... which will make it a bit difficult to get a good-looking installation.

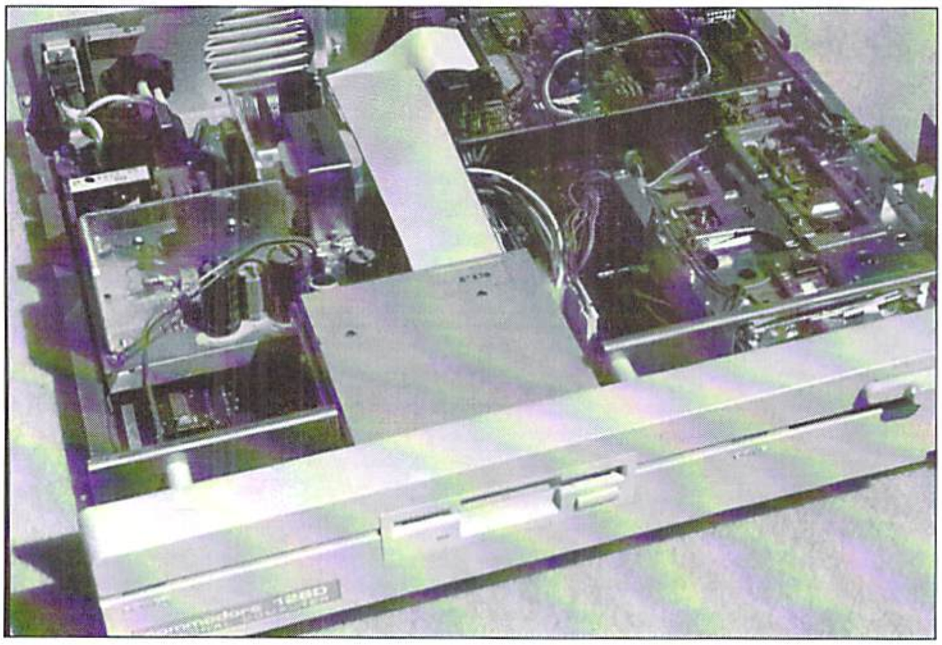
Gimme that drilling machine!

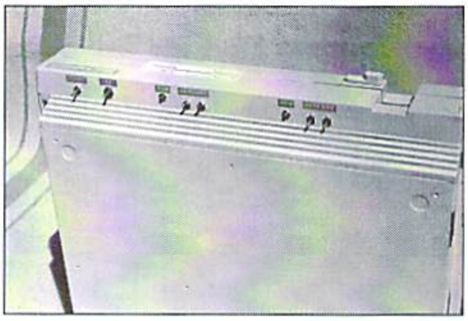
There are two ways to solve this: we can either cut a slit for the disk and a hole for the eject button by ourselves, or we can get an original drive that still has a front panel (see photo). These 800KB floppy drives are available from Amiga retailers (e.g. Vesalia, see Amiga mags). Alternatively we can connect a PC drive to the 1581 board. In issue 2/98 of GO64! I showed how to do this conversion.

The second part, namely the board, can easily be fastened anywhere in the computer cabinet. So, after we've provided the right cut-outs in the front of the computer we put in the drive and install it with metal elbows. How you do this in your particular case, is left to your own constructing skills, as there are several ways. After installing the drive and floppy board, we now provide the electric connections.

Soldering iron, at your service

Didn't you always feel annoyed by the fact that the 1571 couldn't be set to different addresses than 8? While we are at it, we might just as well add an address switch, which enables us to set any address from 8 to 11. The circuit schematic accompanying the article shows how it's done. The 3.5" floppy has address switches, too; however, as we can't reach them once the board is installed inside the computer, we have to put them to the outside. The bottom edge of the computer's front panel is a good position (see photo). So, this bottom edge is to hold two address switches for the 1571, another two for the 1581, and a two-pin on/off switch for the 1581. We also add reset buttons for each of the drives, since the drives need to be reset

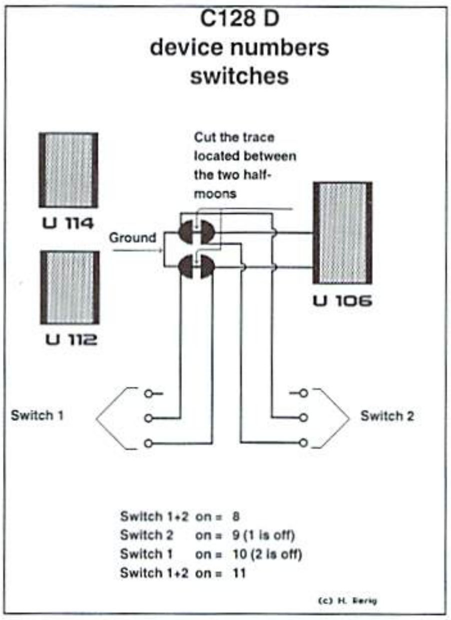




in order to assume the address we set on the switches.

Whoops!

Now we handle the rest of the electrical stuff: first, the serial connection. To obtain it, we connect the six contacts of one of the two serial sockets on the 1581 board, with the corresponding ones on the serial connector on the the computer board. The power switch on the 1581 board is set to "off" and a new switch is connected to the equal pins of the original switch, using at least 16 guage wire. Three lines are needed for the power supply: Ground (GND), +12 Volts, and +5 Volts. These voltages can be taken from different points on the main board, near the power supply connector (check with a multimeter). Again, these points are connected to the corresponding pins of the power switch on the 1581 board, using at least 16 guage wire.



Switch 1+2 on = 8
Switch 2 on = 9 (1 is off)
Switch 1 on = 10 (2 is off)
Switch 1+2 on = 11

Our next task is to connect the drive to the 1581 board: for this, we need a flat ribbon cable with 34 strands, and the power connector. Both of them have to be significantly longer after the conversion, which means we have to build our own cables. And at last, we have to install the reset buttons: for the 1571 reset, we simply tap the orignial reset key for the 1571 located near the rear end of the main board, and the one for the 1581 is conected to ground and pin #10 of U12 of the 1581 board (a 7406 chip).

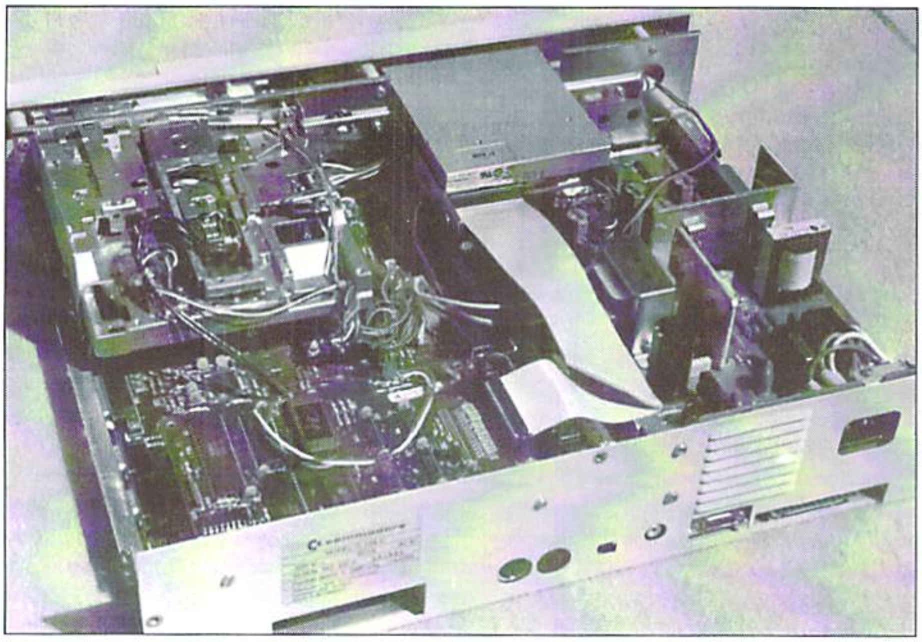
Surely you've noticed that this conversion doesn't require great skill, but even in this case, you can't do without some thorough basic knowledge.

Knowledge + capability = power!

Those who know how to do it, just need the idea, but no instructions, and those who don't know how, won't find much help in even the most detailed instruction guide. When using a technical device, be it a computer, a motorcycle or a lawnmower, you should ALWAYS acquire enough basic technical knowledge to be able to intervene in case of emergency. It's always a disadvantage if you understand little or nothing at all of a technical matter, because you'll be bound to other people.

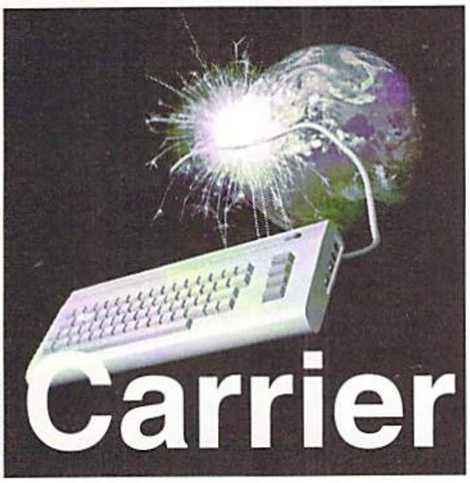
And now, let's be proud of what we've just accomplished.

yours,
Niko



If you have ideas or wishes for new hardware projects, don't hesitate, but let me know. Contact me under
CSW-Verlag
Goethestr. 22
D-71364 Winnenden
Germany

E-Mail: go64@c64.org



Carrier Detect

Picture: Sebastian Bach

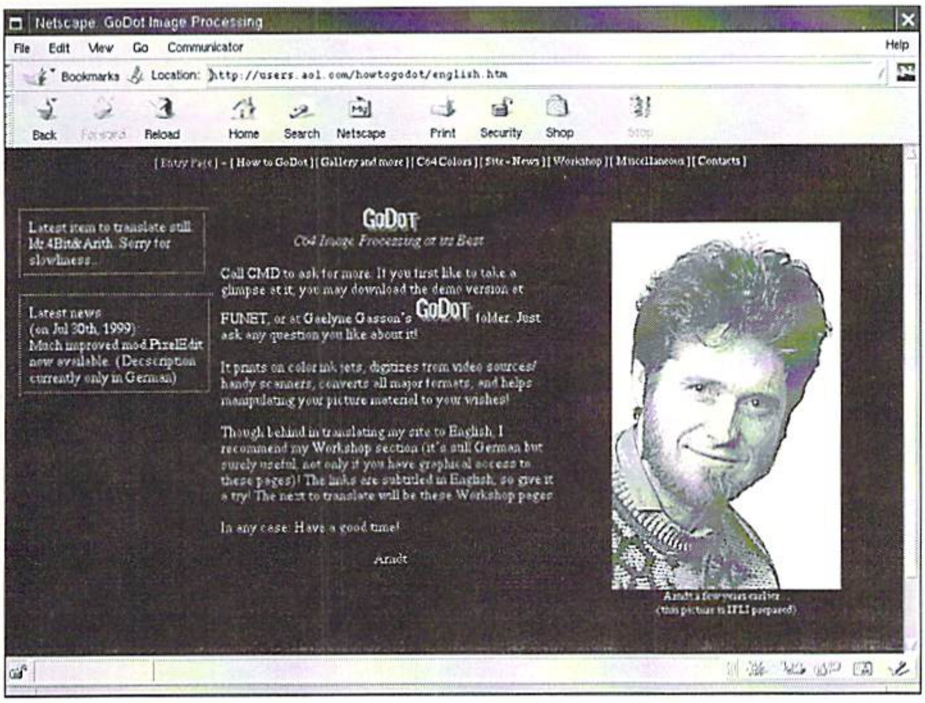
by Gaelyne R. Gasson

W3M

Maurice Randall recently posted an announcement in the comp.sys.cbm newsgroup acknowledging that a new web browser is in the works for our 64s and 128s. His announcement was aimed towards programmers interested in helping with the project, which will be open source, allowing for further development. While we wait to see what develops in the Commodore realm, we can explore a "new" text based web browser, named "W3M", written by Akinori Ito. This isn't really a new browser, as it started out as a UNIX 'pager' (or text viewer) in 1991. W3m is an abbreviation of "WWW-to-Miru", Japanese for "See the ".

What's truly magic about W3M is that it allows us to view web pages that use frames and tables. It 'renders' the page in a format that's close to perfect. It can also handle cookies and recent versions can browse secure web sites, such as for banking or online ordering.

To demonstrate the differences between Lynx and W3M, let's visit Arndt Dettke's English GoDot page **http://members.aol.com/howtogodot/english.htm** Using Lynx, the GoDot page appears with a menu at the top, two news items are listed, and then we see the word "GoDot" centered on the page, followed by the main text. Below this there is a link to a picture of Arndt, centered horizontally in the middle of the screen with some descriptive text. At the bottom of the page, there's a horizontal line, then on the left hand side, there's a link to send Arndt Email, and his copyright information.



http://users.aol.com/howtogodot/english.htm

[Entry Page] - [How to GoDot] [Gallery and more] [C64 Colors] [Site - News] [Workshop] [Miscellaneous] [Contacts]

Latest item to translate still: Mr.4Bit&Arith. Sorry for slowliness..

Latest news (on Jul 30th, 1999): Much improved mod.PixelEdit now available. (Description currently only in German)

GoDot

C64 Image Processing at its Best

Call CMD to ask for more. If you first like to take a glimpse at it, you may download the demo version at FUNET, or at Gaelyne Gasson's GoDot folder. Just ask any question you like about it!

It prints on color ink jets, digitizes from video sources/ handy scanners, converts all major formats, and helps manipulating your picture material to your wishes!

Though behind in translating my site to English, I recommend my Workshop section (it's still German but surely useful, not only if you have graphical access to these pages)! The links are subtitled in English, so give it a try! The next to translate will be these Workshop pages.

In any case: Have a good time!

Arndt

Arndt a few years earlier...
(this picture is IFLI prepared)

GoDot Homepage and Netscape

When we view the same page using W3M, we see the menu at the top of the page, plus three separate columns, the left side contains the news items, the center has the main text of the page, with the word GoDot appearing at the top of the page, centered horizontally. The right-hand column has a link to Arndt's picture with descriptive text. Underneath the three columns is a line, and then the link to send Email appears on the right side, while the copyright information is on the left.

It's the same page, but W3M has displayed it closer to how a graphical browser shows the web page. I've included screen shots of the page as rendered by W3M, Lynx, and Netscape to show how the same page is displayed by all three web browsers.

Installing W3M

Your System Administrator will need to install W3M. It's available for most types of UNIX systems, and there's a Windows version too, so most Internet providers should be able to install W3M and have it available for online use. When you ask your SysAdmin to install it, make sure to tell them it's at http://ei5nazha.yz.yamagata-u.ac.jp/~aito/w3m/eng/.

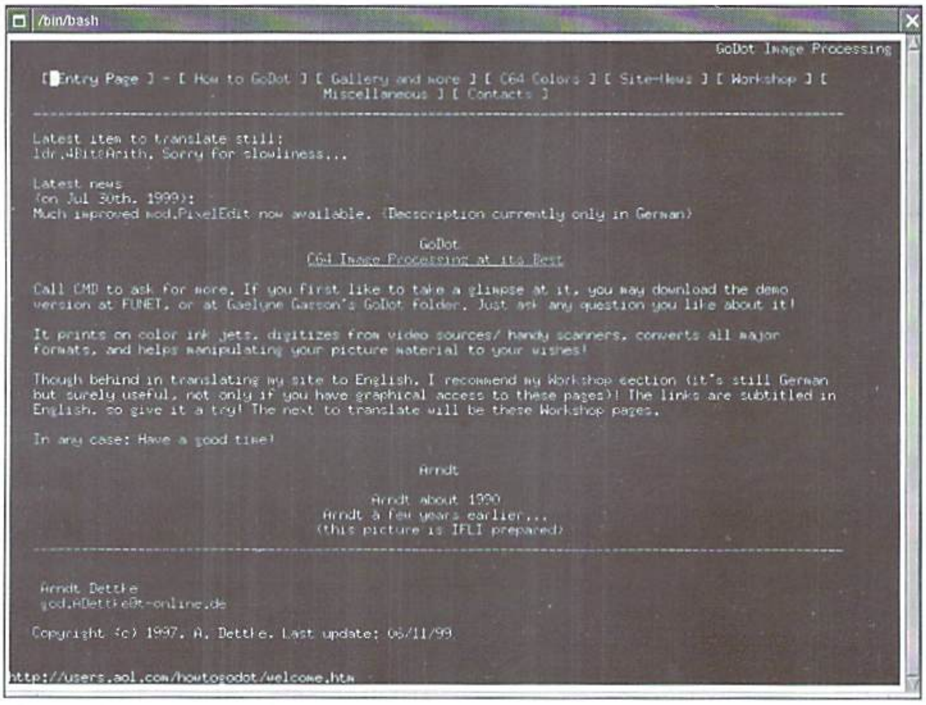
When W3M is compiled, there's an option to set the type of keybinding it uses - its own, or a set that's similar to Lynx's default settings so that most of the common keys you type in Lynx would be the same in W3M. This is an area of personal preference, but unfortunately it's also something that can only be set when the program is compiled. If in doubt, ask your SysAdmin to install it with "lynx-like keybindings". He (or she!) will be prompted with this question by the software, so it doesn't hurt to let your sysadmin know your preferences ahead of time.

Using W3M

Unlike Lynx, W3M must be started with a file name or URL (web address), such as:

```
w3m http://cbm.videocam.net.au/
```

Getting used to cursoring around a web page with more than one column may take some practice. In my experiments, I found that when the cursor reached the bottom of a column, it would then jump to the top of the next column. Sometimes a viewed page will



```
/bin/bash
                                                                  GoDot Image Processing
  [ Entry Page ] - [ How to GoDot ] [ Gallery and more ] [ C64 Colors ] [ Site-News ] [ Workshop ] [
                                     Miscellaneous ] [ Contacts ]

 -----------------------------------------------------------------------------------------------

 Latest item to translate still:
 ldr.4BitsArith, Sorry for slowliness...

 Latest news
 (on Jul 30th, 1999):
 Much improved mod.PixelEdit now available. (Decscription currently only in German)

                                              GoDot
                                 C64 Image Processing at its Best

 Call CMD to ask for more. If you first like to take a glimpse at it, you may download the demo
 version at FUNET, or at Gaelyne Gasson's GoDot folder. Just ask any question you like about it!

 It prints on color ink jets, digitizes from video sources/ handy scanners, converts all major
 formats, and helps manipulating your picture material to your wishes!

 Though behind in translating my site to English, I recommend my Workshop section (it's still German
 but surely useful, not only if you have graphical access to these pages)! The links are subtitled in
 English, so give it a try! The next to translate will be these Workshop pages.

 In any case: Have a good time!

                                              Arndt

                                        Arndt about 1990
                                   Arndt a few years earlier...
                                  (this picture is IFLI prepared)

 -----------------------------------------------------------------------------------------------

  Arndt Dettke
  god.ADettke@t-online.de

 Copyright (c) 1997, A. Dettke, Last update: 06/11/99

http://users.aol.com/howtogodot/welcome.htm
```

GoDot Homepage and Lynx

appear off the side of the screen. To see it without cursoring up and down columns, you can use either the "<" or ">" keys (or their corresponding unshifted keys such as the coma and period. If W3M was compiled using it's own keybindings, instead of the up/down cursor keys, use 'l' (the letter 'el') and 'h' for right and left cursor movement. For up/down cursoring, use the 'j' and 'k' keys. You can also use the TAB key to move between hyperlinks. If ever in doubt of how to use W3M, you can press "H" (shifted 'h') to view the Help page. It lists keystrokes and commands that are used within the program.

When filling out web forms (such as participating in online forums), when the cursor is over a line to be filled in, press the return key. In the bottom left-hand corner, the word "TEXT:" will appear with the cursor next to it. Just type what you want and press return again. The text will appear in the web form. For longer text fields, W3M will start your text editor. Once you've typed your text, save the file and you will be returned to the web browser with your words of wisdom in the text box. This is useful as it allows you to use any of the tools your text editor has available, such as spell checking or importing text from another file.

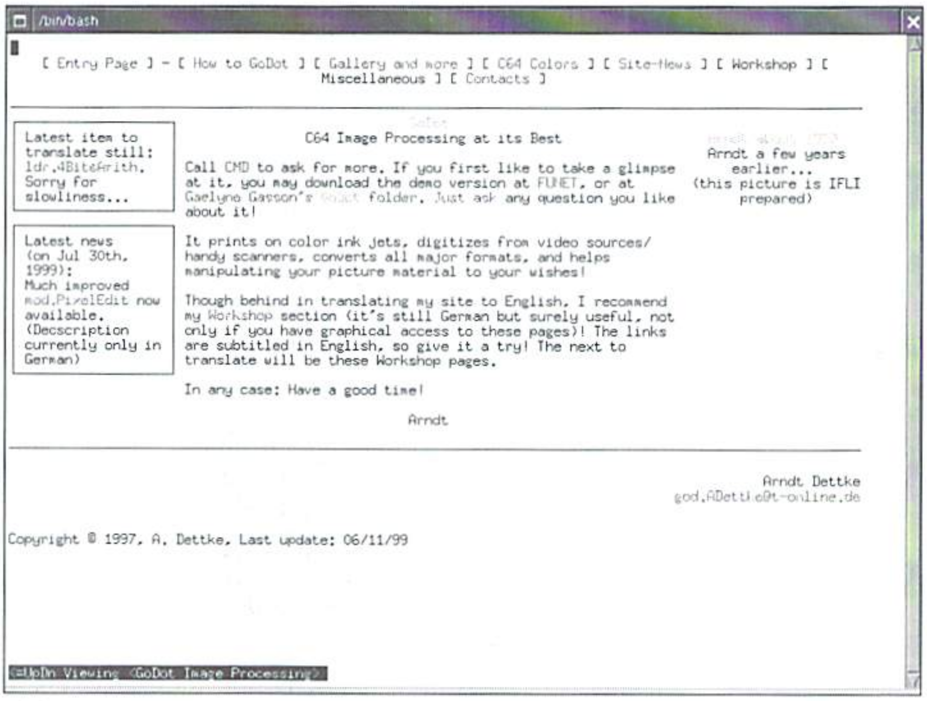
Configuring W3M Options

After giving W3M a try, the next step is to set it up to suit your tastes. To set your preferences, press the "o" key for "Options". A web page will appear with several different sets of options. After each set of options is a link with the word "OK". If you're only changing one setting you can select "OK" in that section - your preferences will be saved, and you'll be back at the previous web page you were visiting.

A couple important items to set in the options are automatic rendering of frames, and the colour choices. This is one of the areas where W3M shines, as it's difficult to change the colour choices with Lynx as it requires editing the lynx.cfg text file. Near the bottom of the options page, you can set the mailer and editor that W3M will use. If you use Pine for Email, type "pine" for the mailer. The editor is used when filling out forms online. If you're not sure what editor to use "pico" is a safe bet.

Even though W3M does an excellent job at displaying web pages with tables or frames, it still has some of the same drawbacks as Lynx, but for the same reason - if a web designer elected not to make use of the "Alt" tag to display alternative text, we see only the name of the image. Unfortunately even some Commodore related web pages fall into this category. Another drawback is that W3M only comes with help files and configured for two languages - Japanese, and English. The author states that W3M is not a replacement for Lynx, and this is fair. It's not a replacement, but it's a tool we can use for seeing the web from a different perspective. *(ad)*

Gaelyne R. Gasson is the author of The Internet for Commodore C64/128 Users. You can find her online at http://cbm.videocam.net.au



```
/bin/bash
█
  [ Entry Page ] - [ How to GoDot ] [ Gallery and more ] [ C64 Colors ] [ Site-News ] [ Workshop ] [
                                  Miscellaneous ] [ Contacts ]
 ________________________________________________________________________________________________
 ┌──────────────────┐
 │ Latest item to   │          C64 Image Processing at its Best
 │ translate still: │                                                         Arndt a few years
 │ ldr.4BitsArith,  │ Call CMD to ask for more. If you first like to take a glimpse      earlier...
 │ Sorry for        │ at it, you may download the demo version at FUNET, or at   (this picture is IFLI
 │ slowliness...    │ Gaelyne Gasson's GoDot folder. Just ask any question you like       prepared)
 └──────────────────┘ about it!
 ┌──────────────────┐
 │ Latest news      │ It prints on color ink jets, digitizes from video sources/
 │ (on Jul 30th,    │ handy scanners, converts all major formats, and helps
 │ 1999):           │ manipulating your picture material to your wishes!
 │ Much improved    │
 │ mod.PixelEdit now│ Though behind in translating my site to English, I recommend
 │ available.       │ my Workshop section (it's still German but surely useful, not
 │ (Decscription    │ only if you have graphical access to these pages)! The links
 │ currently only in│ are subtitled in English, so give it a try! The next to
 │ German)          │ translate will be these Workshop pages.
 └──────────────────┘
                      In any case: Have a good time!

                                       Arndt
 ________________________________________________________________________________________________

                                                                                  Arndt Dettke
                                                                       god.ADettke@t-online.de

Copyright © 1997, A. Dettke, Last update: 06/11/99



≡UpDn Viewing <GoDot Image Processing>
```

GoDot Homepage and W3M

How to code SuperCPU compatible

CMD SuperCPU Corner

Learn to program in a SuperCPU-compatible way. This is important even if you don't own a SuperCPU yet. Make your programs benefit from the accelerator when it is present. Here is how it's done.

by Malte Mundt

Whether you own a SuperCPU or not - your programs should work with it. And while the engineers at CMD have managed to make their accelerator as compatible as it was technically possible, there are still some things to keep in mind when writing programs that are to be SuperCPU compatible.

Don't use illegal opcodes

The first thing, which is obvious and therefore quite known, is the fact that you must not use any of the so called "illegal" opcodes of the 6510. These opcodes usually save two or three cycles by performing several operations which normally would have required more than one instruction code, thus needing more cycles. Such opcodes do not work on the 65816. The reason for this is simple: These "illegal" opcodes are undocumented and the developers of the processor did not even intend them to exist. So, the people at Western Design Center, makers of the 65816, defined and implemented all of the opcodes not already in use by one of the standard 6510/02 processors, to include new, more powerful and more useful instructions. Thus, all possible opcodes from $00 to $FF are used. The values of the "illegals" now have documented meanings which are, of course, completely different. This usually leads to a program crash when illegal opcodes are used. Therefore, please avoid them to keep your program SuperCPU compatible; you probably don't need to save those two cycles anyway.

Load at 1 MHz

Loading routines (and other disk routines) are timing dependant. Small waiting loops or NOPs are used to create precise timing - for example to get rid of the necessity of an extra clock line for the data transfer. This enables specially written loading routines to perform disk operations much faster than the stock routines supplied in the ROMs by Commodore. JiffyDOS is also based on the timing of a plain C64, this is why the SuperCPU switches down to 1 MHz during disk access. Note that it is not the SuperCPU itself (or some hardware component inside it), but the kernal that does the switching, as soon as the disk routines are used. The conclusion: If you don't use the kernal in your own fastload routine or IRQ-enabled-loader, you must switch down to 1 MHz manually. This is easily done with:

```
STA $D07A
```

After loading, switch back to turbo mode with a simple:

```
STA $D07B
```

The value of the accumulator is unimportant in this case, as the registers are write-sensitive. Any write to the register will trigger it's function, regardless of the value written.

Perform timing critical stuff in 1 MHz mode

What applies for loading routines is also important for everything else which is dependant on 1 Mhz timing. For example, if you want to display a picture in FLI mode and cannot (or don't want to) write a routine which generates the FLI in turbo mode, switch to 1 MHz before starting the display routine, and switch back to turbo afterwards. The same goes for similar stuff like opening the side borders or even just creating color rasterbars - the original 1 MHz timing of the 6510 is the base you can always return to if needed. However, all processor-intensive routines of course should be executed in turbo mode whenever possible. When it comes to special routines only used when a SuperCPU is present, for example an FLI routine which is designed to run in 20 MHz, it frees a LOT of processor time for other tasks. Please consult the paragraph above (about loading) on how to switch between 1 MHz and 20 MHz.

Don't calculate in the IRQ routine

SuperCPU owners have experienced it: Some programs simply don't get faster with the SuperCPU! One of the main reasons, especially when the program contains vector calculations or similar routines, is that the main work is done within an IRQ routine. A frame oriented way of programming is very common among demo coders, however, it is not very efficient, especially on the SuperCPU. In 20 MHz mode, a heavy calculation would then simply take less raster time to execute, leaving a lot of unused time until the VIC (or CIA in some cases) triggers the next IRQ. So if you calculate something, scroll, move, whatever - do it in the main program. Do not "split" the calculations on frames. Let your routine calculate as much as it can in a given time - it will manage to do a lot more with a SuperCPU then.

The memory doesn't end at $FFFF

Some programmers assume that the memory is like a globe. They leave Europe to sail westwards and expect to reach India. In other words, it is assumed that after $FFFF, the next address is $0000 again. Consider this: a program initializes the IRQ/NMI vectors (located at $FFFx) and some zero page addresses with a single loop like:

```
 LDX #$00
loop: LDA data,x
 STA $FFFA,x
 INX
 CPX #$2A
 BNE loop

data: .byte
$00,$c0,$00,$c0,$00,$c1,$2f,$37,$a5,$76,
$a1,$98 ...
```

The loop starts with a value of $00 in the x-register. The NMI vector ($FFFA/FB) and later the IRQ vector addresses at $FFFE/FF are filled with the desired values. With the next increment of the x-register, a kind of overflow occurs. On a plain C64, the address wraps to $0000. However, the 65816 is capable of addressing up to 16 MB directly. Since the SuperCPU has a second 64K bank of SRAM to keep the ROM images there for fastest execution, (located at $010000 upwards!), the little loop above will store the values intended for the zero page at $010000+ instead. After that, your program assumes that the correct values are in the correct zero page addresses, which is not the case when a SuperCPU is active - and because of this it will crash or at least not work as it should! The solution is simple: A bit more "proper" programming. Fill the two areas ($FFFx and the zero page) with the desired values in two separate initialization loops.

Problems with 2 MHz

In some programs, a manipulation of $D030 on a C128 in C64 mode (with the screen turned off) is used to switch to 2 MHz and gain speed. Unfortunately, this may create problems with the SuperCPU. So avoid using $D030 if you find a SuperCPU on the system. Surely users will love programs which run not only at 2 MHz, but also at 20 MHz!

Use only documented locations

While most cases of SuperCPU incompatibility resulting from the usage of illegal opcodes and the other compatibility problems outlined above only apply to a minor number of programs, usually we find them right in those applications and games we would like to run with a turbo enabled SuperCPU.

However, there is one point left to mention: Please only use documented locations in the I/O-area! Although it may sound unbelieveable to some, there are programmers who write to $D220 instead of $D020, $D212 instead of $D012 and so on! Due to the nature of the C64's design, these locations do work and do what their counterparts at the real addresses usually do. But on the SuperCPU, methods like this cannot work. The SuperCPU handles I/O accesses in a very special way and was not designed to support writes or reads to undocumented mirror locations. In addition, the SuperCPU has it's own little portion of RAM in the I/O area - $D200-$D2FF is used by the system, while at $D300-$D3FF is an extra block of RAM available for SuperCPU-supporting programs. The conclusion is simple: Only use documented address locations - this applies not only to the I/O area, but also to Kernal routines.

Detecting a SuperCPU

With the knowledge we have gained so far we are able to write programs that are 100% compatible to CMD's accelerator board and even can benefit from the presence of a SuperCPU. If you want to let your program "know" whether it is executed by a SuperCPU or a plain Commodore system, this can be done in a simple way: Just check whether bit 7 of $D0BC (a special SuperCPU register) is zero. A plain C64 always returns a one, like every read from an I/O address that is not used. But on a SuperCPU, this bit is normally zero - thus giving us the possibility to easily detect a SuperCPU. Once detected, we are able to select alternative routines for SuperCPU owners, or even use full SuperCPU only programs which could take advantage of all the powerful new opcodes and addressing modes the 65816 offers. *(ws)*

WORLD WATCH

by Rainer Buchty

This month's focus: USB

HELLO WORLD,

a new bus system has appeared in the world of computers and is becoming the new, platform-spanning standard for (slower) peripherals: the universal serial bus, in short - USB.

In the beginning, there was frustration...

Developing hardware would be so easy, if not every computer system had come with its very own proprietary bus system. Even on the C64, hardware designers have to handle a variety of ports like, the expansion port, the user port, as well as the datasette and joystick port. And there's not much difference on other systems. The Commodore Amiga comes with many different expansion slots - depending on the version of the machine - and PCs have a sheer babylonical diversity of different slots, like 8-bit ISA, 16-bit ISA, EISA, MCA, VLB, and PCI... but all those systems have one thing in common - you never have enough expansion slots!

Serial?

The name itself is the difference, compared with that of traditional bus systems. The classical expansion buses, which usually derive from the system architecture of a computer (expansion port on the C64, Zorro on the Amiga, ISA and VLB on PCs...), are parallel buses consisting of an address and data bus as well as various signals to control the bus transfers. This design had practical reasons, since it allows any kind of hardware to be easily integrated; however, portability is affected by the dependency on the system's architecture.

Serializing a bus is no new invention, since Commodore already did it with the IEEE488 bus back in 1980, creating the serial IEC bus we all love (easy cabling) and hate (no! it really wasn't that fast!). Insiders know the IEC bus protocol is identical to that of its big brother, the IEEE488 bus, except that it has fewer lines, which means that some lines have to be employed for several functions at once.

Well?

USB goes a step further by reducing the number of bus signals required to two, besides two power supply lines. The times are gone when you needed an additional power supply unit for each external device. Today any device that's not too hungry for power can simply be driven via the USB connection.

For those who believe USB has a slow bus, here's a nice surprise for you. These are the data rates that USB can provide:

- 12 MBit/s
- 1.5 MBit/s (for "slow" devices like keyboard or mouse)

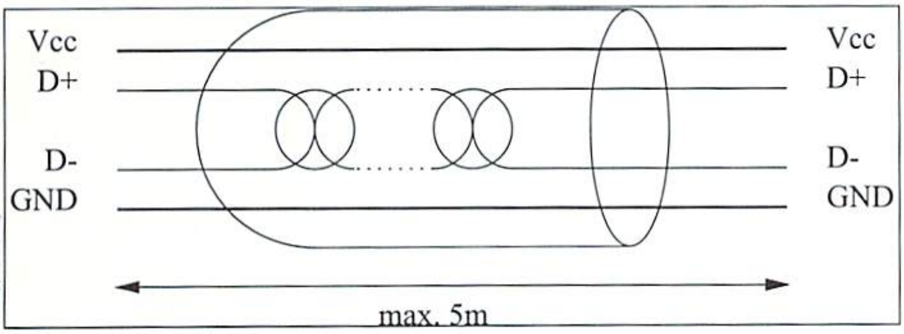
This equals to maximum transfer rates of 1.5 MB/s or 187kB/s (the IEC bus, using speeders, provides around 10 kB/s). It's interesting to note that both data rates can be used on the same bus.

While we're at it, strictly speaking, USB is not even a bus in the actual sense (where all devices have equal and parallel access to the bus, like on the IEC bus), but rather a peer-to-peer network. This means a USB device can only communicate with its immediate neighbor, and any locations lying further away can only be reached by sending the packets through all devices between the sender and the receiver. For this purpose, every USB device has a dedicated data input and a data output port. However, it's possible to build star-layout networks - as known from networking technology - by using so-called hubs. This is particularly useful in order to separate devices with slow (1.5MBit/s) and fast (12MBit/s) communication, since this way the faster devices aren't slowed down by the slower ones. USB has some other interesting features, like the possibility to reserve bus bandwidth for real-time transfers (asynchronous transfer), but we won't go any further into this at this point. Those who are interested may turn to the USB specifications [1].

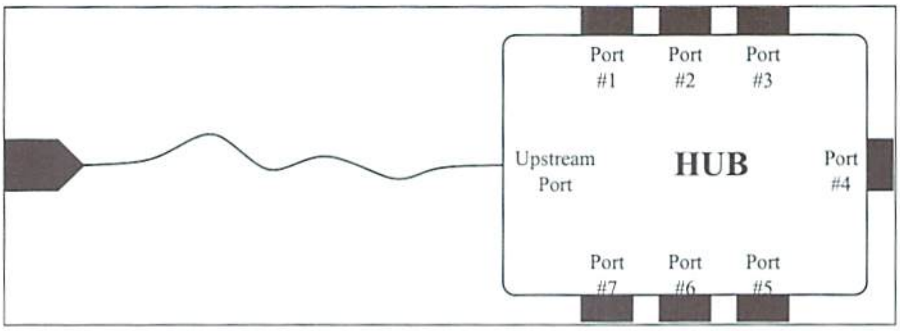
Can we have that, too?

Alas, no, at least not at the moment. However, it could be possible, because there are appropriate (and low-priced) chips on the market that can receive and process a USB data stream [2]. Using these, a connection to the C64/C128 could easily be achieved.

But is the C64 fast enough to receive the data? There's no doubt that the slower USB mode is no problem. 187 kB per second are easily transferred into memory by using the DMA; we just have to coordinate it with the VIC. 1.5 MB (or 12 Mbit) per second are a little harder to handle, since that's more than 50% above the bus bandwidth of a (PAL) C64. We could still create a connection, given a USB interface with sufficient buffer memory that would transfer the incoming



Picture (3): Rainer Buchty

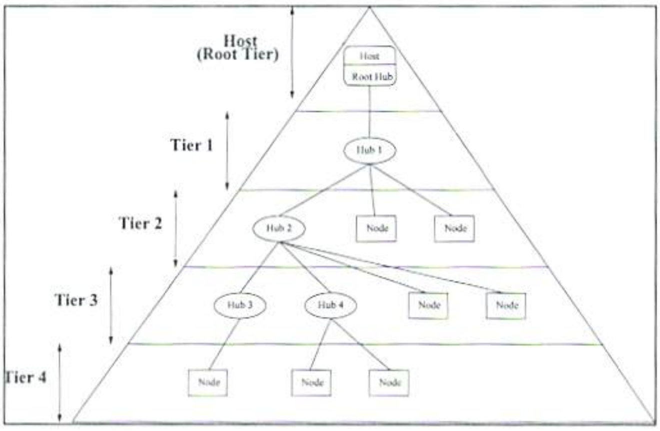


data into the C64's memory in chunks. Of course, this would reduce the effective data rate. Instead of being constantly blocked by the DMAs, the C64 should still be able to process the data, which means the actual maximum data rate would depend on the kind of application and data. If data is only received, about 1 MB per second (or 9 Mbit/s) is possible; with additional data processing, 500 kB/s to 750 kB/s (4 to 6 Mbit/s) should not be exceeded, in order to preserve enough computing time (bus cycles) for the actual processing.

Unfortunately, there's more to do than just receive data. The processing of USB data causes a considerable workload on the memory as well as the rest of the machine. Even on a state-of-the-art PC, a constantly active USB interface generates around a 5% cpu load. You can guess what this means on a C64. From this point of view, a SCPU seems to be an absolute requirement for using the USB, because it offers the required cpu power. Also, a REU or SuperRAM is an absolute minimum requirement, due to the memory intensity of the USB management structures.

An interface such as this would be interesting at any rate, because it would allow us to use keyboards and mice from PCs on the C64, for example. On the other hand, there are other ways to achieve this using less resources. It would enable us to have the use of modern sound cards and other USB peripherals, like scanners, but appliances such as these require the full bandwidth of 12 Mbit/s. What makes USB a practical concept is the fact that only drivers have to be written in order to support these devices; the devices themselves (finally!) don't have to be customized to a particular platform anymore. This would allow easy access to modern peripherals with the C64.

Rumours are that there might be a USB adapter from CMD that supports the slower transfer mode. However, due to their recent announcement [3] to withdraw from developing new Commodore hardware in the medium term and only consider new developments based on careful market analyses, it remains questionable if this adapter will ever see the light of the day.



Sources

[1] USB Specifications V1.0
http://www.usb.org

[2] Anchor Chips EZ-USB
http://www.anchorchips.com/ez_usb.htm

[3] Rainer Buchty, "The Last Resort", #5!, published in GO64! Issue #5/99
http://www.buchty.net/articles/cmd.html

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

Shatter the 1 MHz Barrier

SuperCPU

20 MHz Accelerators for the Commodore 64 & 128

- **Breathes New Life into your Commodore 64 or 128**
- **Increases System Performance by up to 22 Times**
- **Makes Sluggish Old Programs Quick and Useful**

Take Your Commodore Into The Next Millenium!

Programs will suddenly spring to life, screens and menus pop up instantly, scrolling becomes fluid, and graphics fly. A Commodore equipped with a SuperCPU is like having a Volkswagon with a V-8. There's one small difference: the SuperCPU rides much smoother. And while the SuperCPU provides your computer with a radical increase in horsepower (up to 22 times faster), its operating system makes that power easy to use by maintaining compatibility with most existing programs and by making itself virtually transparent to the user and the computer.

Looking to the future, the powerful 65816 microprocessor gives programmers vast new resources to create powerful and sophisticated software. With its 16-bit registers, up to 16 Megabytes of system memory and an enhanced instruction set based on the 6502 core, the SuperCPU makes it easy to enhance existing software while providing the muscle needed to create a whole new generation of Commodore programs integrating more extensive graphics, sound and animation.

To get the most from your existing programs and to get ready for the future with just one puchase, get a SuperCPU. You'll wonder why you waited this long!

SuperCPU 64	$179.00*
SuperCPU 128	$239.00*

*prices do not include shipping

Features:

- High-speed microprocessor, 128K or 256K of fast RAM, advanced hardware logic, and up to 512K ROM enable you to run all types of existing commercial software at speeds up to 22 times faster than stock machines.
- BASIC programs, including most Bulletin Board Systems, seem to run as fast as machine language, and compiled BASIC programs run faster than ever!
- Commercial machine language programs (databases, spreadsheets, word processors, telecommunications, graphics, etc.) run so fast that you'll think you're running them on a completely different computer!
- GEOS comes alive—No typing or scrolling delays, graphic operations happen instantly, plus faster printing and disk access! Includes an enhanced GEOS setup utility!
- Easy-to-use: Plug it into your cartridge port** and start working—no special adjustments needed.
- Built-in JiffyDOS Kernal boosts disk access speed up to 15 times faster with JiffyDOS-equipped drives.
- Pass-thru port provides compatibility with RAMLink, SwiftLink, Turbo-23 2, REUs and other cartridges.
- Unit Enable/Disable, JiffyDOS On/Off, Normal/Turbo, and Reset switches make using the SuperCPU a snap!

**Additional MMU SuperAdapter installation required for 128 version.

SuperRAM Card (0 MB) $79 • 4 MB $114 • 8 MB $139 • 16 MB $179
SuperCPU 64 w/SuperRAM (0 MB) $219 • 4 MB $254 • 8 MB $278 • 16 MB $318
SuperCPU 128 w/SuperRAM (0 MB) $279 • 4 MB $314 • 8 MB $338 • 16 MB $378

Prices do not include shipping and are subject to change without notice. Not responsible for typographical errors.