The Magazine for true Computer Fans
Das Magazin für wahre Computerfreaks
2/2000
DM 9,17 USD 6,17

64'er

C64 commodore WORLD

HARDWARE

PROJEKT: HYBRID - DER ETWAS ANDERE C64 TOWER IM EIGENBAU

HARDWARE PROJECT: HYBRID - A SPECIAL C64 DO-IT-YOURSELF TOWER

GEOS

BRUCE THOMAS IST ZURÜCK MIT SEINER KOLUMNE GRAPHIC INTERPRETATION

BRUCE THOMAS IS BACK WITH HIS POPULAR COLUMN GRAPHIC INTERPRETATION

COVER

C64 IM EINSATZ

DER C64 IN DER ZAHNARZT-PRAXIS UND ALS BUCHHALTER IN DER DISCO

THE C64 ON DUTY

MANAGING A DENTIST'S PRACTICE AND A NIGHT CLUB WITH THE C64

GODOT WORKSHOP, SUPERCPU CORNER, BASIC EFFECTS, JUST FOR STARTERS PLUS WORLD WATCH

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

FLOPPY DISK DRIVES

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

HARD DRIVES

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

MISCELLANEOUS HARDWARE

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

UTILITIES

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

LANGUAGES & COMPILERS

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

GEOS

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

MIDI & SOUND

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

EDUCATIONAL

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

ACCELERATORS

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

RAM DEVICES

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

PRODUCTIVITY

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

TELECOMMUNICATIONS

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

GAMES

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

BOOKS

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

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

Creative Micro Designs, Inc.
P.O. Box 646 **Info:** (413) 525-0023
East Longmeadow, MA 01028 **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.

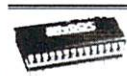
Novaterm *version 9.6*

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 online and store in any device; High-speed transfers directly into memory and now **retain buffer contents with battery/power backed RAM devices such as RAMLink.**
- Enhanced full featured **Text editor can now load/save files from the buffer** and has an integrated script compiler.
- Script language for automatic operations
- Simple BBS mode - dial-in downloading
- Supports **real-time clocks in CMD devices**
- **All new user's manual**



JiffyDOS™

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

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

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

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

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

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

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

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

Shipping and Handling Charges

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

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

* Other shipping methods available. Call for details.

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

CONTENTS

- February 2000 -

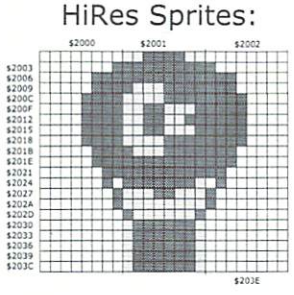
7
Practice

Although it is already 18 years old, the C64 still isn't exclusively a sentimental or nostalgic item. In two articles we will show you that professionals are still using their C64 system to maintain their businesses. Turn to page 7 to read more!

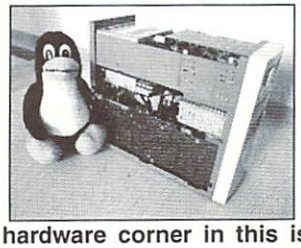
13
Monochrome GIF pictures

In this installment of Arndt's "How to GoDot", you will learn how to convert monochrome GIF pictures without any loss of quality from PC to C64. A hint and a few tools do the trick. More at page 13.

16
Assembler Tutorial, Part 11



Ever wanted to know how to control sprites and detect collisions in assembly language? If so, then part 11 of Wanja's Assembler Tutorial is right for you. The fun starts at page 16.



25
A special tower

...called "Hybrid" is part of our hardware corner in this issue. Consisting of two systems, this machine has a PC and a C64 with all of their devices living in perfect harmony in one and the same tower. A perfect tool for coders, graphicians and musicians. What else can you expect? To page 25 at once!

A

Assembler Tutorial - Files
How To Godot - Files
SuperCPU Corner
Contents of GO64! 1999

B

GEOS Stuff

A new year arrives.

The year 2000 arrived with owners of leading-edge computers worrying about whether their systems would survive. Meanwhile, the Commodore machines just kept on ticking. There's a certain satisfaction in having a simple computer whose architecture you know precisely.

There are disputes as to whether the third millennium begins in 2000 or 2001. Commodore's first personal computer was called the PET 2001. Those early machines still work well.

The Commodore computer world keeps its momentum. Users chat, meet, and swap notes. There are gatherings of computer enthusiasts. Although Commodore has been gone for some time, new and improved products still appear for these legacy systems.

I believe there's a Commodore secret here: their computers were fun. Somewhere along the line, the new-technology machines have lost it. The classic eight-bit computer magazines dealt with fun things: writing programs; discovering features of your machine; trying a new approach to handling files. Magazines that deal with new-tech computers are focused on product reviews and buying guides.

The Commodore computer story started almost 25 years ago. Chuck Peddle designed a single-board 6502 computer. When his company was acquired by Commodore, he wanted to build a full-sized personal computer. He pitched the idea to Jack Tramiel, Commodore's CEO, and the PET 2001 was born.

The PET was ahead of its time. It had upper and lower case, a feature not seen on competing machines. Its screen editing, coupled with a type-ahead keyboard, was brilliant. Intelligent peripherals - disk drives and printers - were a Commodore innovation. The serial bus, which came later, reappeared recently as the PC's USB or FireWire.

The sheer power of new computers should make them the dream machines we always hoped for. There can be no doubt that processing power, memory speed and capacity, storage devices, and, yes, even printers, have developed far beyond what we imagined back in the old days. Yet the new machines seem more hostile and harder to handle. Experimentation is hazardous. Programming is discouraged. Viruses (virii?), both real and imagined, are scary.

Arthur C. Clarke, the author of "2001: A Space Odyssey" said that he picked the year 2001 for several reasons. He said, "With advances in medicine, it's likely that those who see the year 2001 will also see the year 3001."

I have a feeling that the historic Commodore machines are likely to last that long.

Jim Butterfield

News Flash

German C64 disk magazines and the complete PD compilation by Reiser now available on one CD-ROM

'Magazineworms', here it comes: Ferdinand Gansberger, the owner of the 'C64 mag"s web page, is now offering the complete collection of magazines on CD-ROM. The CD costs DM 10 and amongst many other things you will also find included - apart from all issues of Digital Talk, Tiger Disk, Killer Drive, Mail Madness and Intel Outside - the PD series Rockford PD, Rainer's PD and other, less well-known magazines. Ferdinand Gansberger also makes available a CD-ROM with the complete public domain program by Tanja Reiser, former Master-MM-Soft. If you order the magazine CD you can also get the second CD for the small extra amount of DM 3. So act quick before Ferdinand has second thoughts!

Supplier address: **Ferdinand Gansberger, Grünwies 3, A-3232 Bischofstetten, Austria**, Email: **ferdinand.gansberger@utanet.at**, **http://www.c64-mags.de** *(vr)*

'Bubble Bobble Nostalgie' on the PC

In February, the software company 'Alawar Entertainment' published a PC version of the famous game 'Bubble Bobble' for C64 platforms, now called 'Bubble Bobble Nostalgie' (nostalgia). From now on, up to two players can also jump and run through 90 levels under Windows. You can download a playable demo version on the company's homepage for free. By the way: the game has followed the hype about computer improvement: the system requirements for it are 'ludicrous' 166MHz.
http://www.alawar.com *(vr)*

Annual meeting of the GIG Süd e.V.

This year, the annual meeting of the GIG will again be in Westphalia, Germany, to be exact, in Unna. The organizers' invitation reaches from 3rd to 5th of November. Among the main subjects will be GeoWork, DFÜ, magazines and demos. Accommodation will be in the Hotel Kraka in Unna's city center. Room charges for single rooms are roughly DM 89 and the Kraka family will personally take care of the physical well-being of the GIG's visitors with the preparation of the meals. Anyone who is interested please contact
Bodo Faros,
Grabengasse 16
D-59423 Unna, Germany
Telephone: (+049) 02303/237375 *(st)*

Revival of an old scene disk magazine

The scene group 'Role' from Belgium has decided to start to publish again their famous disk magazine 'Rock'n'Role', now at its 22nd issue. Those of you who'd like to see what's become of the cult magazine or who do not know it and are curious to know, you can download the magazine from the internet at **http://www.role.de**. *(st)*

Dear readers!

As you may have already noticed: this issue's cover is bilingual. This is a reaction to rising printing costs which could result in a higher subscription rate. To prevent you, fellow readers, from those higher rates, we reacted this way.

Additionally we want to apologize for the delays of the last issues. We are trying everything we can to catch up some time with the next couple of issues.

The GO64! staff

WORLD WATCH

by Rainer Buchty

This month's focus: Wireless

HELLO WORLD,

today, the C64 has lost a lot of its importance. It is hard to believe that, among other things, the very same computer was the most up-to-date communication computer 10 years ago, dominating the digital modes of amateur radio. Because the majority of readers may not be too acquainted with the subject, we will begin with a short introduction.

Digital radio modes?

What are the first ideas that spring to mind when a layman thinks of 'radio'? Maybe Morse code or radio-telephony - but these are only minor ones in the wide range of applications amateur radio offers. For fans the most interesting creative opportunities are the digital radio modes. The following are only the best-known ones:

(1) RTTY

This enigmatic acronym stands for 'Radio TeleTYpe'. Data transmission is executed with unbelievable data transfer rates of 45 and 75 baud respectively, in a special 5-bit code that is called baudot. Transmissions in ASCII with up to 100 baud are less frequently found. RTTY works by connectionless protocol; thus, it doesn't register possible defective transfers.

(2) AMTOR

AMTOR (AMateur Teletype Over Radio) resembles RTTY. One point of difference between the two is that AMTOR works in two modes; the call is made with the connectionless FEC mode (Forward Error Correction), and then the transmission changes to the ARQ mode (Automated ReQuest). The first mode's advantage over RTTY is that a certain fault tolerance is included in the alphabet used. For the latter mode, a simple connection protocol is kept so that faults occurring during transfer can be traced and deficient data packets can be resent.

(3) Packet Radio

Packet Radio was the shooting star amongst the digital modes, so to speak. Originally based on the X.25 protocol (slightly modified for amateur radio, so it should correctly be called AX.25 protocol), Packet Radio makes it possible to use TCP/IP, the communication protocols used for the Internet.

The ability to establish single computers as network node points especially added to the success of Packet Radio. For the first time, long distance connections were possible on VHF (144 or 453 MHz) and SHF (>1GHz), which before had only been feasible on shortwave.

Nowadays the low data transfer rates of 1200 to 9600 baud seem useless; due to its narrow frequency bands, shortwave only allows for 300 baud.

(4) Pactor

Pactor is one of the newest modes nin comparison to the others, that is) and tries to bring together the knowledge gained from the AMTOR and Packet Radio modes, especially considering transmissions on shortwave. Pactor thus guarantees extremely reliable data transfer with relatively fast transfer rates (300-1200 baud), at least as far as shortwave goes.

(5) Morse code

It was never conceived as a digital mode, but nevertheless, Morse code is the most digital of them all, because initially it only has two states: signal on or signal off. Furthermore, the length of each signal is measured. One long sound (dash) by definition has the length of three short ones (dots). Single letters are separated by spaces of three dots, single words by spaces of five dots. The length of one dot results from the speed with which the message is transmitted - if transmitted by humans, the average speed is 30 to 80 signals per minute. The best senders reach a speed of up to 200 signals a minute, using a special Morse key that only differentiates between dot, dash, and dot/dash.

This is not a difficult task for a computer - at least it seems so. But in fact, many of the Morse programs suffer from hearing problems due to the human factor involved in sending, and which exceeds the abilities of these programs. Was that a space of 3 or of 5 dots? Was it a dot, or was it maybe a dash? We as humans intuitively grasp the difference, but computers encounter insoluble problems.

The advantage

The interesting thing here is that at that time C64s could only work in these modes (except for the Pactor mode which only came into use after the C64's heyday) with the aid of special converter circuits or simple modems. PCs needed a special machine (TNC: Terminal Node Controller) for the operation of these modes. One way out of this dilemma was soon found by the programmers of DIGICOM (see below) who transferred the successful program from the C64 to the PC, where it has been setting standards ever since under the name of BAYCOM.

The revolution

Let's start with what is probably the most successful amateur radio software for the C64 - DIGICOM. Even as early as in 1985, it made the world of Packet Radio (by then dominated by the TNC solution for PCs) accessible to C64s. The fact that this software only required a one-chip modem was next to revolutionary. By the way, an interesting feature of the program is that the programmers, Florian Radlherr (DL8MBT) and Johannes Kneipp (DG3RBU), decided to establish the cassette port, which usually is not used much, as a modem port. The user port remained free at first, only to later serve as a port for status information data (active connections, current messages). From the users' point of view, DIGICOM was especially attractive because of its easy use and its excellent 80-sign software mode.

Happily, more development work was put into DIGICOM, so that today - if used together with a special modem - it compares with similar modem solutions for the PC.

The all-rounder

'Bonito' was the name of another popular amateur radio software in use by the mid-80s. It was a public policy of the program to unite all modes. Connections by Morse code, AMTOR, RTTY, and others were possible with the appropriate converter circuit. The plans for these converters were even published in the 64'er magazine (although they weren't all too helpful).

In contrast to many of the other amateur radio programs, 'Bonito' was downright commercial, a fact that was criticized since in the spirit of amateur radio, this kind of software should have been shared for free (roughly the same concept that inspired the GNU license) with fellow amateur radio operators (hams).

Anything else?

Apart from the usual AMTOR and RTTY tracks, a wide range of other software for special purposes was available. Besides additional modes like receiving pictures from the METEOSAT satellite (which gave radio amateurs a clear advantage over the people depending on TV weather forecasts), there were programs for image transmission via SSTV (Slow Scan TeleVision) or simple utility programs for keeping a protocol, for recording and analyzing QSL cards (confirmation cards for the connections made), or for even controlling X/Y rotors to automatically follow the OSCAR satellites or for EME links (earth-moon-earth).

Today the C64 has become outdated in nearly all areas because of the rise of PCs, but the reason we use the C64 is in its familiarity rather than in its computing capacity. Thus, there's really no reason for not using the C64 as your first computer for amateur radio (or also for CB in the case of Packet Radio). Nevertheless, the most important requirement is that you must be experienced in DIY work with your computer. In particular, mounting converter circuits takes more than the usual soldering knowledge. Ideally, you should also understand the circuit you are mounting, because later on you have to tune them. Those of you who are not particularly skilled in this respect can find all your heart desires at http://baycom.de where the original C64 products are still sold today.

So finally, there's only one thing left to say. Everyone is allowed to listen to the transmissions, but if you are toying with the idea of going on the air, you should definitely read the regulations that govern the subject in your country. The whole range of modes and corresponding frequency bands is only open to licensed radio amateurs.

Rainer Butchy (DL1GRA)

Addendum to issue 1/ 2000's cover disk

JPX/JPY/JPZ

Development on the revolutionary C64 JPEG viewer (with SuperCPU versions available) is finished for the moment. The new versions are very stable so that the programmers now want to take care of working on such JPEGs. But first, the confusion about the program's name should be cleared up.

JPX was the very first version, and it showed pictures only in black-and-grey. At the moment, this program has been abandoned. When compared with other, more recent versions, its results were not that spectacular; however, the name has remained, designating the entire program package. JPY is its successor's name and is able to present JPEGs in gray-scales. To achieve this, all shades of gray of the C64 are used in interlace mode. Finally, JPZ comes up with colors, its IFLI mode trying to make the best of the 16 standard colors.

These last two versions have been developed simultaneously; that means they both have the latest JPEG decoder routine, the only difference being in the presentation. What is conjured up on your screen really deserves our respect! Not only is the JPEG format decoded but also the picture data is adapted for the C64. A JPEG can have up to 16.7 million colors, and the way JPY/JPZ converts such data to the C64 should really be applauded! In addition, JPY/JPZ isn't that choosy anymore. In the beginning, not all JPEGs could be displayed, but with this version, the author and our editors don't know of any standard JPEG which can not be displayed (except CMYK images).

The exceptions to this are "progressive JPEGs". This special variant is very popular on the Internet, because the picture starts out being blurred but gets more and more detailed. This way, when dealing with big pictures, you can get a hunch of what to expect.

Such pictures can't be handled by JPY/JPZ directly; they have to be converted first. One program able to do the conversion is called JPEGTRANS and is available for all "big" platforms for free. At this moment, there is no such program for the C64.

Using JPX/JPY/JPZ is really simple. After you have loaded and started the desired version, you are asked for the picture's filename. By the way, it has to be in the currently active disk drive, and so, you may want to change the device first. The program always just shows a window of 320x200 dots, no matter how big the picture really is.

In order to move this "viewport" around a bigger picture, you will then be asked for the "Offset in Columns" and as well in "Rows". No offset (0,0) shows the upper left corner. For instance, an offset of (1,2) moves it to the right by 8 dots and down by 16 (the step size is 8 dots). Warning: JPZ still has a minor bug there, so you shouldn't enter odd numbers for the column offset, because this could lead to errors in the display. After you have given the information, the display process starts right away. When you have had your fill with viewing the picture, SPACE returns you to BASIC. To then restart the program, enter SYS4096. Another word on JPZ - memory is used by up to 99.9%. That's close to the limit; crashes are not always avoidable, but as long as the border is flickering, everything is all right. Now have fun with these great programs.

Jammon V3.2

There is also an update for the SuperCPU machine language monitor, JamaicaMon. It is now able to handle breakpoints, use single step tracing, and use 24-bit addresses for loading and saving so that you can easily work with RAMCard memory. Because this program is public domain, there is also the complete source code on disk. It is written in PETSCII format, which most assemblers should be able to import. A precise list of the commands is also provided in the file "JAMMON.DOCS".

Picture: Sebastian Bach und Marco Platzdasch

The C64 in (the Dentist's) Practice

Only thinking of the dentist makes many people grimace as if in pain. Once entering Dr. Barrut's office in Morbach, Rhineland-Palatinate, Germany, those imaginary aches are all forgotten. You want to know why? Only keep reading ...

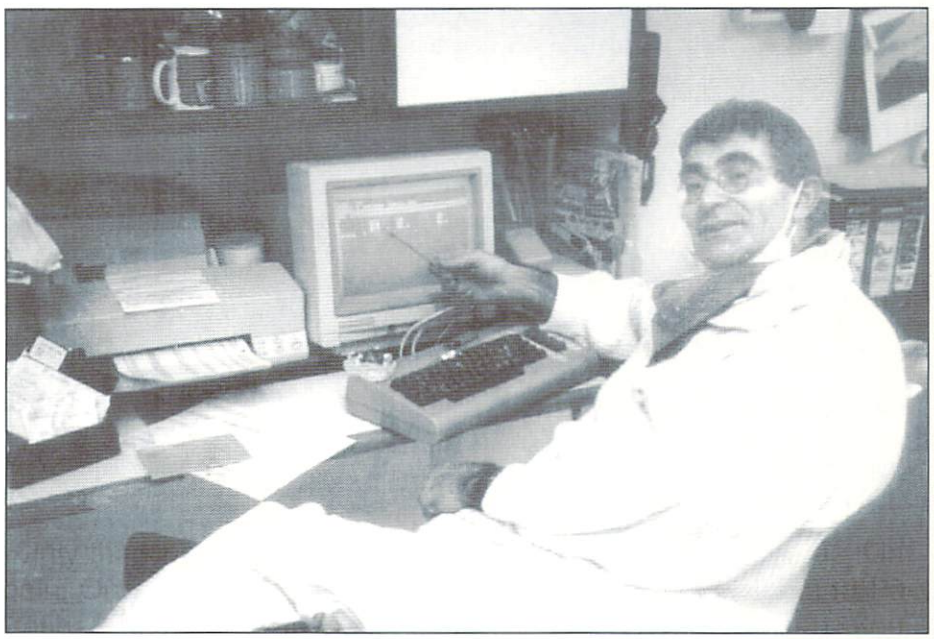
by Volker Rust

Everyone knows this situation. Each year you keep postponing the visit to the dentist's, since the examination could quite easily turn into a medically successful but nevertheless painful experience. This is not the case at Dr. Barrut's office: It's not only the drill that awaits you there, but also - you wouldn't believe it - a c64. It is not there to distract the patients in the waiting room with games as you might expect at first. Instead, it serves, thanks to self- designed software, a vital function in the daily works of the practice.

This was reason enough for us to ask Dr. Barrut a.k.a. "Mogui" for a short interview.

GO64!: *How come that there is a c64 in your office rather than a pc?*

Mogui: For one thing, it was a question of money. I recovered my old engine, a c128, from the attic in 1996. Two years ago, I "changed" to the c64. I had started designing the BASIC program managing my patients already in those heroic days around 1985. I have recently bought a pc for surfing on the internet, but the interest in continued programming keeps me with the c64. The c64 is the only machine that allows me to enter it in that way. In addition, I also have the ambition to become able to program the computer completely, that is to say also in assembly language.



Closeup: Mogui checks his accounts

Breitwiese 12 in Morbach: Things look completely harmless from the outside

GO64!: *What are your patient manager's functions like?*

Mogui: My manager solely consists of the cost accounting. As far as this is concerned, distributing the administrative costs to their respective accounts is my whole pride. In addition, I also use it to manage the patient accounts, which you can professionally print thanks to an Epson Stylus Color 640.

From the office routine: take the bare necessities

Screenshot from Barrut's software. It uses almost the complete BASIC memory, you can hardly find a free byte

GO64!: How do the patients react to the c64?

Mogui: Actually only few of my patients really notice that it is a pre-historic computer setup. But sometimes I even receive hard- and software for it from my patients as presents.

GO64!: What else do you do with the c64?

Mogui: I mainly use it for Mastertext, my accounts, and for learning programming. My great dream is still controlling my vacuum cleaner with the c64. For this purpose, I have built a driving frame already in 1986/87. I would like to turn on the vacuum cleaner when I am leaving the office in the evening and everything should be clean when I come back the next morning. One day, once I have expanded my assembly language knowledge, I will fulfill this dream.

GO64!: How did you get in contact with the scene?

Mogui: In the PCGo!- 64'er magazines I read about the "Digital Talk" disk magazine, which I subscribed to afterwards. By means of it, I heard about one of the c64 parties at Murphy's and there also about GO64! magazine for the first time.

GO64!: Afterwards, you were also at the Symposium Mekka 1999. What's the parties' charm for you?

Mogui: For years I have programmed BASIC at home in a corner and unfortunately never had any contact to the c64 scene. I missed a lot because of this.

GO64!: Thank you for the coffee and the friendly interview!

Inclined to have a somewhat different visit to the dentist?

Mogui alias Dr. med. dent. Luc Barrut
Breitwiese 12
54497 Morbach
Germany
Tel.: +49 6533/1592 and +49 6533/3464

C64 Bookkeeping

by Franz Kottira

Personal

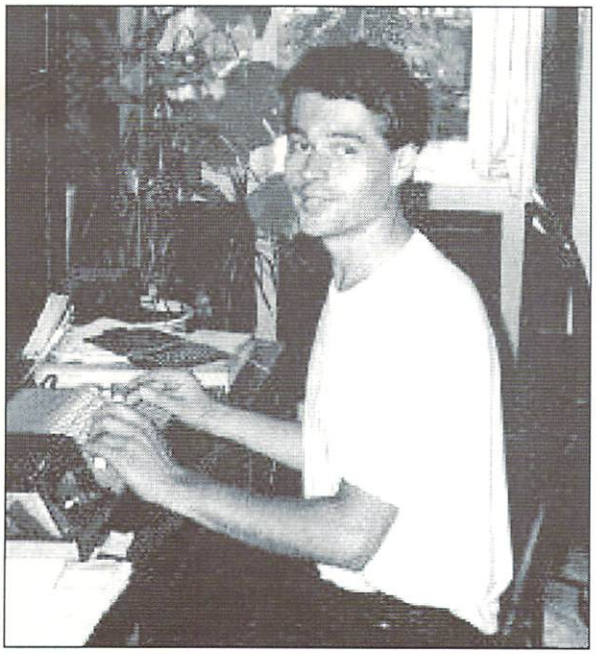
I am a 37-year old sociologist, though originally I received technical training as a mechanical engineer. My preference for working with the C64 even today comes from the fact that C64s are wonderfully easy to program. The advantages are obvious: I have the possibility to tailor the software just as I need it, I can repair the ever-emerging errors by myself and further develop programs along the lines that I require or prefer. This possibility has diminished with every new generation of computers. Unfortunately, I have to be unfaithful to my C64 in respect to things as surfing the Internet, processing long texts or printing nice graphics. But it is still absolutely fit to fulfil the original and basic computer tasks like data processing and computing. In our business office we have kept the C64, with which I do the bookkeeping, advertisement and business mail.

Business

The business, that's a disco with the name of "Wiener Freiheit". The customers are mostly gays and lesbians, but there is also a very mixed clientele. The bar's history is rather atypical and definitely too long to be recounted here as a whole (for further information see **http://members.teleweb.at/wiener.freiheit/t_home.html**

I have been involved ever since the inauguration in 1989, and for some reason or other, I've been stuck with the task of keeping the books, dealing with tax declarations and the like and, in general, doing the office work. With the years, the bar has become larger and larger (at the moment, we are working on another room for the discotheque), and thus the volume of administration work has increased as well.

The different fields of bookkeeping to be handled are these: recording revenue and expenses by means of a cash book (cash receipts and disbursements method of accounting) and recording our own consumption, calculating sales taxes (according to 10, 12, and 20 per cent tax rates), beverage taxes (for alcoholic and non-alcoholic beverages separately), entertainment taxes (according to the consumption in the discotheque on the number of business days) and corporation taxes (according to the business revenue for



```
                Steuerberechnung

Dateivolumen:  Kalenderjahr
               16 Einnahmenklassen
               16 Ausgabenklassen
               128 Buchungstitel
               5132 Buchungszeilen
Zahlenformat:  +- 0.01 bis +- 999999.99
Rechengrenze:  -2*10↑7 < n < +8*10↑7

Druckereinstellung: CBM-DIN Business
Ablage: intern, SKRIPTOR-Format
Tastenbefehle:
     ←          Vorgang abbrechen
   RETURN       Vorgang speichern

Standard-Betriebssystem bleibt intakt,
Wechsel zu Basic-Direktmodus moeglich.
─────────────────────────────────────────
✓ basic v4.0 * wird geladen...
─────────────────────────────────────────
✓ Programm gestartet
─────────────────────────────────────────
Datei laden:                      Jahr
```

Franz Kottira (left) in action. Above: His unique piece of software

one year). The holding society of the bar has to keep an index of members, e.g. in order to send out advertisement mail. And last but not least there's always some correspondence to do. I do all those things with my old C64 and the software that I myself have programmed. At the bar, I work at a special desk that was exclusively carpentered for the C64, two floppy drives, the monitor and a printer. In case the good old C64 should expire one day, we keep two emergency ones in reserve.

C64

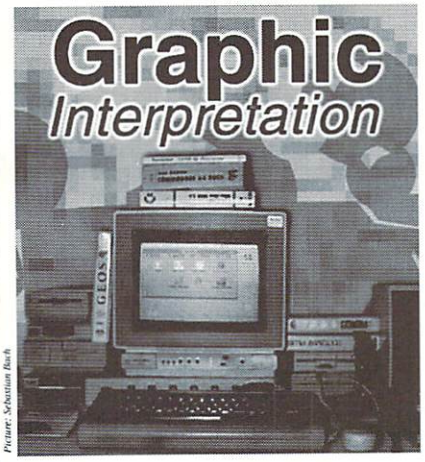
I got involved with the C64 rather late, about 1989 or 1990. A friend who bought it as a used computer gave it to me. (By the way, it is the flat case version.) In the beginning, I made use of the computer for statistics programs and as a comfortable replacement for a typewriter. When I was assigned the bookkeeping of the business - bookkeeping and tax declarations being two of the few things I have never been able to muster any interest whatsoever in - for me, the only way to do this was to look at the whole thing as a sort of programming problem.

The first program for bookkeeping was a mere BASIC program that worked with sequential files. One of the disadvantages was that it was very slow and could only store the data of one month. Together with the computer I had been given about 10 back issues of the 'Input64' magazine, which contained a course in machine language. I taught myself some of it and changed the data administration to a machine language program, which made it both faster and more compact. Version 2 of the program was able to store the data of one year and calculate the taxes with one press of the button. Version 3 included a special program for recording the bar's own consumption. In the meantime, I've reached version 3.4, which is still making life a lot easier for me.

Version 4.0, with a new design and numerous extras, is already in the planning stage. Along with the extension of the bar, it will be necessary to include programs for payroll calculation and depreciation. Most probably I'll start with the development by the beginning of 2000. Now as ever, I'm not very much interested in the subject, so that I have to enhance my interest by the detour of programming. On the other hand, these homemade developments are not just a sort of game; they fulfil their purpose. It takes me three to four hours on average to do the bookkeeping of one month, which is not at all much, in particular if you consider that this does not only involve the things that can be done with a computer but also ordering the records, writing tax declarations and various other things.

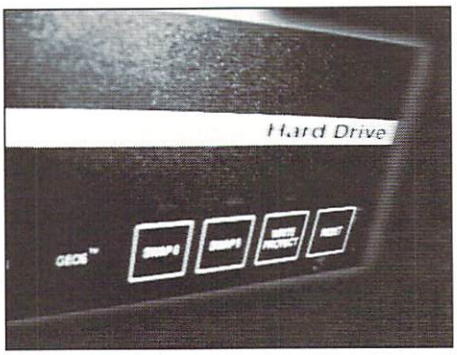
I bet you that (according to the exact requirements in our business) my C64 would win every competition with the fastest computer with the most professional bookkeeping software! *(gb)*

```
KASSABUCH WIENER FREIHEIT  VERSION 4.0
          gueltig ab 1994
──────────────────────────────────────────

     Einnahmen:                Ausgaben:

+ V Verein                  - V Verein
+ B Bar                     - B Betrieb
+ D Disco                   - G Gemeinkosten
+ E Eigenverbrauch          - K Kst
+ T Transfer                - T Transfer

    0   0% Ust   0% Gst        0   0% Ust
    1  10% Ust   0% Gst        1  10% Ust
    2  20% Ust   0% Gst        2  20% Ust
    3  10% Ust   5% Gst        3  12% Ust
    4  20% Ust   5% Gst
    5  20% Ust  10% Gst

──────────────────────────────────────────
Betraege in Schilling
←
```



Picture: Sebastian Bach

geoSpecific



by Bruce Thomas

Welcome to the Future! (or, Never Is a Long Time From Now)

When I bought my first C-64 System (with 1702 Monitor and 1541 Disk Drive) in November 1983 I was interested in bringing my family (at that time we had one'almost-2-year-old' daughter) into the computer age. It never dawned on me thatI would still be using Commodore computers when that child was ready to enter University. A lot has changed in that time but I still adore my 64 as we keep discovering things that a Commodore would 'never' be able to do. I find, as a user, that I like to push my system to the limits and that meanssome things get left behind. The original 64 and 1541 still work but have been 'retired' in favor of newer models; the 1702 is used daily by my 3 childrenwith their Sony PlayStation. My desk now holds a C-64C and a C-128D plus a lot of other hardware we will touch on throughout this article. In future articles I will do my best to include useful info for users who don't have power systems. In all of the time that I have used GEOS there were a number of years when I did so with a stock 64 and two 1541 disk drives. While that seems like a painful thing to attempt nowadays I know there are many users out there who don't have top-of-the-line systems and they still need tips on what software to use, and how to use it, to push their experiences to the limit.

Cheap Storage

One item that I keep on top of my 1802 Monitor is a 5 1/4" Disk Notcher. This little gem has gone from daily usage to being a dust collector - a throwback to times when my system was simpler and storage space was more expensive. I still use a 5 1/4" disk drive occasionally but these days that means the internal 1571 in my 128D. The presence of the 1571 (which writes to both sides of the disk with no 'flipping') means the notcher is history. The 128D has been fitted with a Drive On/Off switch and a Device 8/9 toggle allowing me to deactivate the 1571 and only turn it on when I need to access an old file. Another reason I can live without the notcher is that my system now includes a3 1/2" FD-2000 Disk Drive. When I use High-Density disks I can store almost ten 1541 floppies on one 3 1/2" disk. Most of the software that I use regularly can also be loaded and run from a 3 1/2" disk or I can load the software into my HD Hard Drive or my RAMLink and get terrific performance. Since I love using GEOS I have found, over the past 12 years, software to do almost anything I want to without leaving the GUI environment. The nature of GEOS and its copy protection scheme mean I am not forced to use a 5 1/4" drive. I can even boot my system from the FD, HD or RL. This wasn't always possible but the constantly evolving world of GEOS is what keeps me using my Commodore. Luckily for me, and the thousands of others like me, we have some highly skilled people keeping our systems current.

Improvements and Updates

I have already mentioned that I have a penchant for CMD hardware. The wonderfulfolks in East Longmeadow, Maine, provide the horsepower to make GEOS reallyshine. For a number of years I owned a 4 MHz TurboMaster Accelerator and was able to plow through projects in record time. Of course, the arrival of the 20MHz CMD SuperCPU left the Schnedler unit eating its dust. The highly-publicized Y2k bug also affects our GEOS systems. To keep our systems up-to-date (pun intended) we need software updates in addition to hotnew hardware. In 1995 I submitted a Y2k patch for the GEOS Calendar program (both versions) to Commodore World Magazine (Issue #10). Werner Weicht recently released (GO64!, Oct 99, Page 4) some patches for the Calendar (V1. 2 only) and also for geoWrite/geoPublish/geoMerge to allow the DATE function to print correctly. Todd Elliott also put out a patch for GeoWrite 128 (GO64!, Sept. 99, Page 4) to fully support four drives. GEOS itself has also received a big boost with the release of the Wheels and MP3 systems. Not only do these new systems provide us with more control over our advanced hardware but Wheels is a requirement for the new Graphical Web Browser, The Wave, that is currently under development. In order to graphically surf the Internet you will need Wheels, a SuperCPU w/1 MB SuperRAM (minimum), a Swiftlink or Turbo232 and a high speed modem.

GEOS Print Quality (no longer an oxymoron)

One area in GEOS that rightfully received a lot of flak was printer output. This has also changed dramatically over the years and users currently enGEOy the best output options they ever have. While a few of us have been using PostScript Lasers for over 10 years now the availability of inexpensive, good quality, used units has made it easier for Commodore users to own one. Dale Sidebottom, one of the PostScript Pioneers, chose another route and just recently bought a brand new LexMark Color Laser which he uses to print truly amazing newsletters for his LUCKY User Group (and the articles are terrific also!). If you would like to join the Louisville Users of Commodore of KentuckY (LUCKY) to see the wonderful work that Dale produces with his digital camera, ColorLaser and Commodore computer send e-mail to luckykds@iglou.com or snail mailto: LUCKY Editor, P. O. Box 303, New Albany Indiana USA, 47151-0303. The membership fee is $20 US per year. Some individual newsletter issues are available separately for a fee.

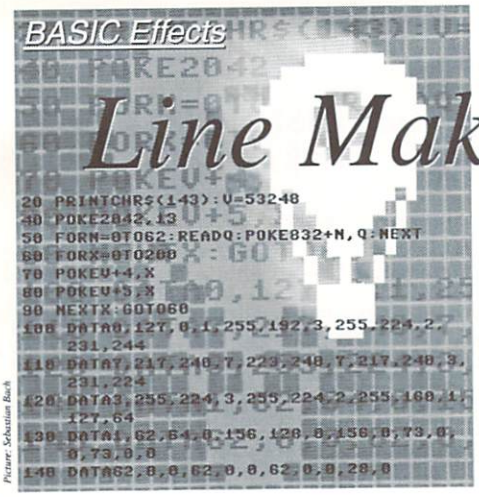
Surf's almost up

If you have access to the Internet there are lots of resources available to provide answers and help you along. One that is of particular interest to me is a mailing list called COPS. This stands for Commodore Only PostScript and is a list that Dale Sidebottom and myself started up with the able help of Gaelyne Gasson. The main focus of the list is, of course, PostScript printing including printing JPEG images straight from our Commodores (even adding them directly into printouts of geoPublish documents using Maurice Randall's PostPrint II). Lately we have opened the list up to be the Official Mailing List of The WaveBeta-Test. As you may guess, these two topics really bring out owners of a lot of cutting edge hardware and software and provide plenty for discussion. To subscribe to the list send an e-mail to: cops-request@videocam.net.au with the word 'subscribe' in the subject line and also in the body of the message. These are just two of many ways to stay in touch with new developments in the world of GEOS. Another one that I have always found useful has been magazines, like the one you are reading now, and I hope to be able to bring plenty of news to you regularly as we move into an exciting future.

Until next time, enGEOy your Commodore!
(me)

BASIC Effects

Line Maker



Picture: Sebastian Bach

When using the original font, text display in BASIC looks kind of monotonous. What can be done to make the text look more exciting? A good solution would be to "decorate" by using graphic characters. Simple PRINT effects are widely known and don't impress anyone anymore. Why don't we do it differently for a change?

by Marcus Hinzmann aka Silver Fox/Satovia/ Sanity

Underlining text lines can be used quite effectively, and in this issue, I present an elegant way to do so. We will draw lines across the screen, taking the freedom of having them go either from left to right or from right to left. The interesting thing is that the lines seem to appear dot by dot. In text mode? Is that possible? Using a little trick, it turns out to be quite simple. Let's start our adventure!

Memory Address 646 ($0286)

There are different ways to influence the color when dealing with the normal PRINT routine. POKEing the color code into address 646 is the most elegant, though. You can find the color codes for the different colors in your C64 manual.

Memory Address 53265 ($d011)

We already worked with this address in the last issue. This time we use it in a different manner. Therefore, some explanation about the screen's composition...

The C64 creates its screen approximately 60 times per second on NTSC machines (50 times per second on PAL machines). Responsibility for this lies in the so-called raster beam. You can imagine this process as being similar to a typewriter permanently filling a sheet of paper with characters. It starts in the upper left corner and goes to the right char by char until reaching the margin. There it changes to the next line and starts over again at the left margin. The raster beam acts the same. It passes over the screen line by line and sets dots, colors, and so on. We can find out the raster beam's current position by means of address 53265. We will make use of this.

System Routine 58640 ($e510)

We already used this routine in the last issue (GO64! 12/99). It was used to set the cursor. In order to do so, we POKE the x position (column 0 - 39) into address 211 and the y position (line 0 - 24) into address 214. After having done that, we execute SYS 58640, and the cursor will be at the position defined before.

And This Is How the Trick Works...

Our routine for drawing line "dot by dot" is relatively short but offers us many options. It is supposed to draw lines for which we can select the color they should have, from which direction they come (from the left or from the right), in which line (0-24) they should appear, and with which chars they consist. How does it work then?

First thing is that we get the line drawn dot by dot, and not - as is usually the case - char by char. We get help with this from address 53265. As described above, it gives us the raster beam's current position while it creates a NTSC screen 60 times a second. If we put a line of 40 chars onto the screen using PRINT as usual, the raster beam will be often where the line string's next char should be drawn; we get in its way. As long as we don't take care of the output, the PRINTing will be visually unpleasant, as we notice when the line is being drawn. But if we wait for the beam to have passed that place and only then put the chars onto the screen, the next time the screen is created it seems as if the chars are being presented regularly dot by dot.

Those of you who haven't grasped the idea completely should really read the last passage again, because you can understand this trick only in that aforementioned way.

What the Program is doing...

Now that we know how to create the dot-by-dot composition, we only need to find a way to present the chars one by one but as fast as possible. Before PRINTing, we WAIT for the raster beam to be somewhere else. We define an output string and position the cursor. Then we can get started. We wait until the raster beam is in an unimportant part of the screen and then print the string char by char. The output not only prints the line character but also positions the cursor for the next output, using control codes. This is important in order to be fast enough.

After this, we wait again for the raster beam to finish showing the last characters. Then we PRINT the next one. We do this 40 times for an entire line of text.

Things Are Getting Serious...

After we exit the routine, the screen should look the same as before. The only difference ought to be the line.

The routine is entered with the following parameters:

M - Mode in which the line is drawn:
1 for from the left to the right
2 for from the right to the left
b$ - char that is to be printed 40 times (don't give more than one char!)
F - the line's color
Y - the row where it should be drawn

As usual, we design a routine starting at line 1000, which we jump at with our parameters. In the first line (1000), we start by defining the C64 addresses used:

W - 53265 (checking the raster beam's position)
S - 58640 (system routine for placing the cursor with x in 211 and y in 214)
Z - 646 (color for the PRINT command)

First, we set the mode variable M for the direction. Another parameter is the string B$. It may only be one char long. The line drawn later on consists of this char.

Because we can choose our line's direction, the routine needs to prepare the string B$ for this purpose internally. For direction 1 (from the left), we put B$ without a change into the

array A$(1). On the other hand, for direction 2 (from the right), A$(2) receives the char B$ and an additional two cursor-left control code that corrects the position during output in a way that B$ moves to the left.

We also have to prepare the starting position (x position). For mode 1, we have to set the x position in X(1) to zero, because we are starting at the left border. For mode 2, we set X(2) to 39, because we are starting at the right border.

In the following BASIC line (1001), the routine for outputting the string is executed. First, we POKE the color F to 646 and position the cursor. Since the row is independent of the mode, we can POKE the variable Y to 214 while - depending on the mode - 0 or 39 is sent to 211 for the column. The two addresses are set now, and we execute the system routine at 58640 in order to place the cursor.

Then we define loop T. T takes on all values from 1 to 40; in other words, the routine will execute that loop exactly 40 times. In the loop, we wait with each character for the raster beam to be away far enough from the output location using a WAIT command. This works, because the program continues only when the bits with the value given behind the command's comma (in our case 128, thus only bit 7) are set in the address in front of the comma (this time 53265). Then we print A$(M). At the PRINT statement's end, there is a semi-colon ensuring consecutive char output. If we left it out, the line's components would appear one beneath the other and not next to the other.

The NEXT command increases the loop counter T and causes a jump back to the WAIT command. If T reaches the value 41, the loop is finished, and the BASIC interpreter continues with the following command. In this case, we proceed to the next command line (1002) where our routine is ended by 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 w = 53265:
   s = 58640:
   z = 646:
   a$(1) = b$:
   a$(2) = b$+"2xleft":
   x(1) = 0:
   x(2) = 39:

1001 POKE z,f:
   POKE 211, x(m):
   POKE 214, y:
   SYS s:
   FOR t = 1 TO 40:
   WAIT w, 128:
   PRINT a$(m);:
   NEXT:

1002 RETURN
```

Terrificly short but still very effective! You can make use of this routine in many different ways. For instance, once you have PRINTed a line, you can have it deleted dot by dot by simply executing the routine again. When doing this, simply leave all the parameters as before so that the line is overwritten. Only the char should be changed to a space.

In order to test the subroutine right away, simply enter the following line in direct mode:

```
b$="commodore + t" : y=10 :
f=1 : m=1 : GOSUB 1000
```

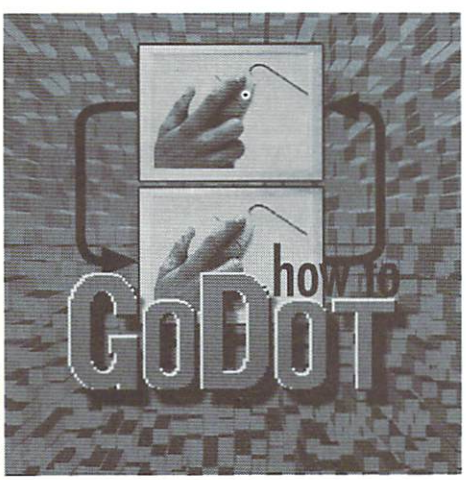
Well, have fun with the Line Maker!

Yours, Marcus Hinzmann aka Silver Fox *(ad)*

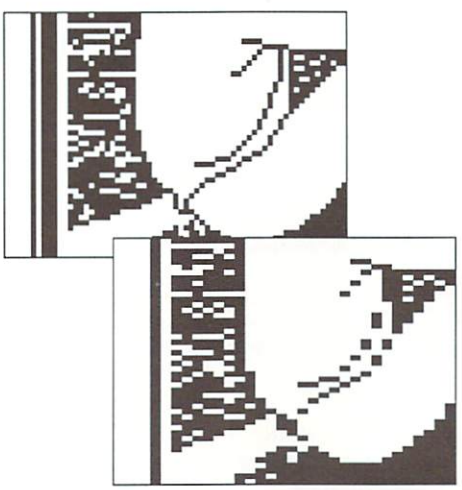
Memory Addresses Used:

53265 = Fetching the raster beam's position
646 = Setting the color for the PRINT command
211 = Defining the x position for placing the cursor (requires SYS 58640)
214 = Defining the y position for placing the cursor (requires SYS 58640)

System Routine Used:
58640 = Placing the cursor after POKE 211,x and POKE 214,y



Picture: Sebastian Bach



by Arndt Dettke

Monochrome GIF Pictures

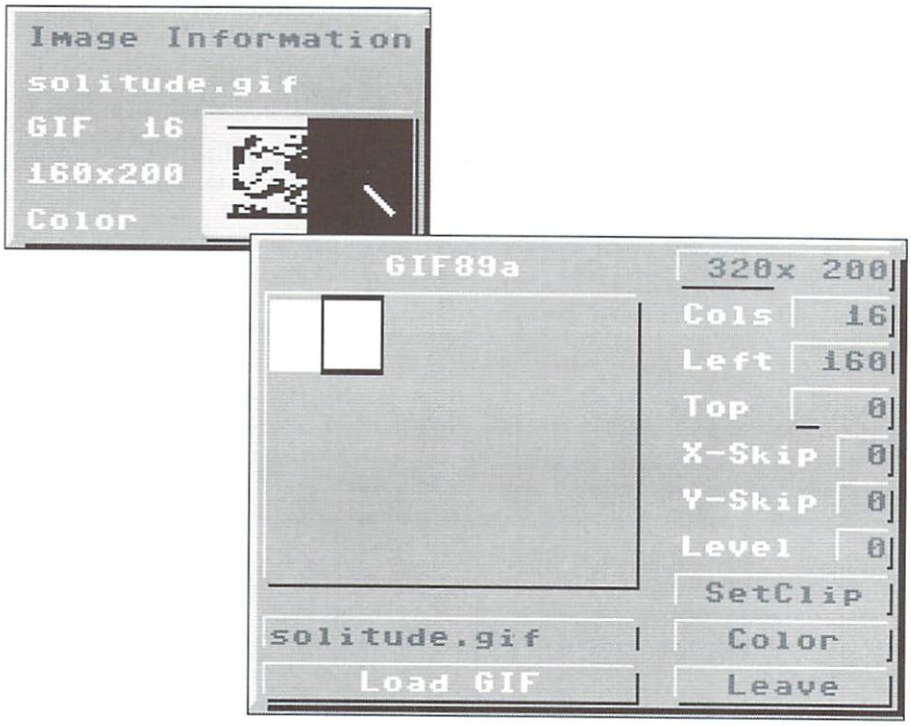
Today, we will be working with **ldr.GIF**, **mod.DecodeGIF**, **mod.ClipWorks**, and **mod.Rendered2Raw**.

GoDot's GIF loader has been designed for loading pictures created by photo cameras, video digitizers, or scanners. Its strength lies in converting colors the c64 screen doesn't have into colors or color combinations that it does have, which are as close to the GIF original as possible. In order to attain that goal, ldr.GIF transforms the individual color dots it reads into multicolor IFLI dots of twice their normal width. The loader therefore turns a 320x200 picture (that is, a hires image), into a 160x200 picture (multicolor resolution).

But what do we do if lines or dots that are just one pixel wide need to be preserved on the c64 side? Ldr.GIF levels them without mercy either into lines of twice that width, or into nothingness! This can be seen clearly in our cutouts when looking at the (actually double-line) left border and the shirt's folds. The doubled border becomes a not-so-wide border consisting of a single line, and the folds are lacking dots, while others appear too big. This can't be accepted, just imagine if you wanted to present circuitry which nearly exclusively consists of thin lines!

Well, GoDot wouldn't be GoDot, if there wouldn't be a simple solution for this problem as well! It is based on the GIF loader's option to be able to scale a picture (changing its size when

loading). For a better understanding, let's have a look at how the loader treats a pixel. It reads one pixel and turns it into an IFLI double-wide pixel. In order to prevent a picture that's 320 pixels wide from becoming 640 pixels wide, every other pixel is simply dropped. When scaling a GIF, we simply determine how many pixels are skipped. The registers required are the two skip gadgets, in particular, the "X skip" gadget.



X skip has the value 1 by default, so that a 320x200 picture keeps its size on a c64's Multicolor screen . One pixel is read, one is skipped. Now, the trick: We set this value to 0 so that the loader will retain every single pixel! It transforms each pixel into a doubled pixel. In order to prevent an overflow, it just fetches 160 pixels, though, before switching to the next input line. This means we have to process a picture in two halves s o that all 320 dots are scanned. This is a tad inv olved because we need to temporarily save, but this can't be helped. Sometimes only tangled paths lead to the goal. That things get done is all that matters.

This is how it works: After the usual preparations for loading GIF pictures, you start by setting X skip to zero and then reading the picture's left half.

```
Load: GIF
Inst: DecodeGIF
Load "solitude.gif"
X-Skip: 0 (click on the left
end of the gadget
once)
Load GIF
```

If we render the picture now using "display" (the graphics mode doesn't matter at the moment, since the pixels have been doubled), we would see a left half stretched to double width. One thing is important though: not a single pixel within this part of the image has been lost! Now we only need to get rid of the distortion.

```
Inst: ClipWorks
Execute
Hor: 20 (click there and enter
manually)
Shrink
Accept
(Screenmode:) Hires
Display
```

Terrific! The left half is already finished! The same procedure needs to be done for the half to the right. Since this would then overwrite the data we have created thus far, our next step needs to be to save our progress first - either onto disk (in this case use **svr.4BitGoDot** without fail!) or into additional RAM:

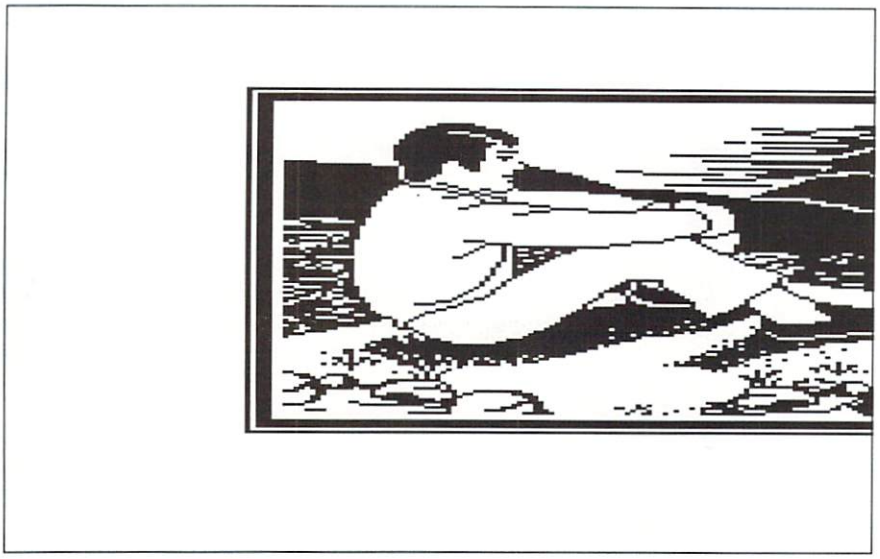
```
Save: 4BitGoDot (default)
Save
Units: RAM
Save "Undo 4Bit" (double click)
```

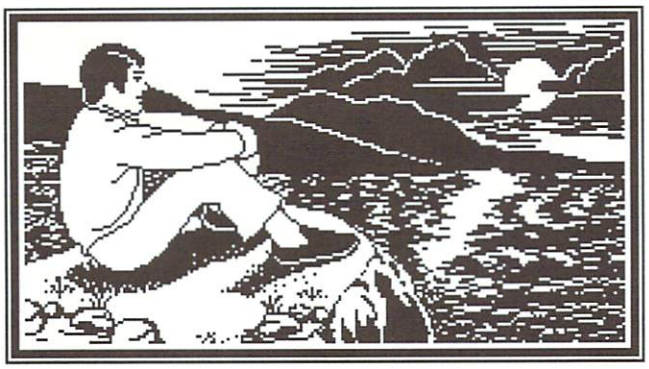
Those who click on the preview gadget now will see that there really is but half a picture in the 4 bit memory. Therefore a "reunion" action is required in the end. Well, now for the other half:

```
Inst: DecodeGIF
Load "solitude.gif"
X-Skip: 0
SetClip Left: 160 (CRSR RIGHT
until "160" appears behind
"Left" then press RETURN)
Load GIF
```

Now the half to the right is also in the memory, distorted again as clicking on preview or display would show. But we continue on anyway:

```
Inst: ClipWorks
Execute
Col: 20
(Hor: 20)
Shrink
Accept
(Screenmode:) Hires
Display
```





The value 20 for "Hor" should actually still be there. Those who set the Exec Area to "Full" on the main screen in between, need to enter "Hor" again, though, or recall it with the "Last" function in ClipWorks. After selecting Display, the screen's half to the left is filled with garbage left behind by ldr.GIF. Well, we now have the second half, but how do we make a single picture out of this? Let's get down to the next to last action (ClipWorks should still be installed):

```
Execute
Col: 0
Accept
Load
Units: RAM
Load "Undo 4Bit"
Display
```

Yeah! Super! There we have the picture! (Those not using a REU have to install ldr.4BitGoDot first.) But wait! Didn't we have only half the picture in memory? Clicking on the Preview gadget confirms this: the image memory contains merely half the picture, yet there is a complete version on the display! How do we get it into the 4 bit memory completely? No sweat, last action (which you must never forget!):

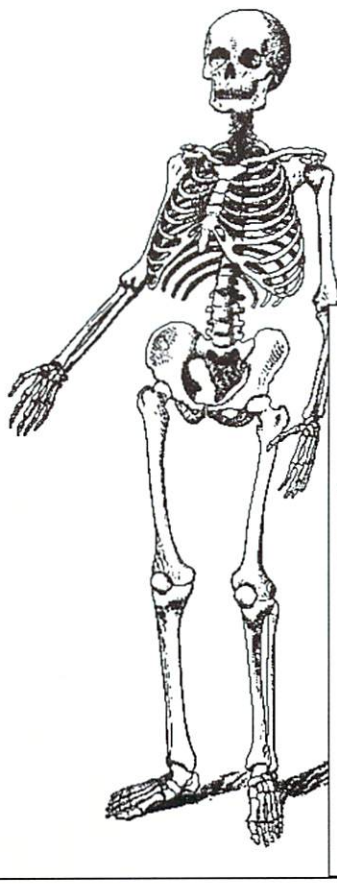
```
Inst: Rendered2Raw
Execute
Exec Area: Full
Display
```

GoDot has transformed the graphic display's content into 4 bit format using mod.Rendered2Raw and stored the data in the apropriate screen memory. Now the process "Reading a Monochrome GIF Picture" has been successfully finished. On the cover disk there are some more monochrome GIFs including "skeleton.gif" and "1351-mouse.gif" so that you can practice. They are far bigger than 320x200 pixels, by the way. Do also spend some effort on these pictures, because some mental arithmetic is required when you have to effectively take a picture apart before being able to have a look at it.

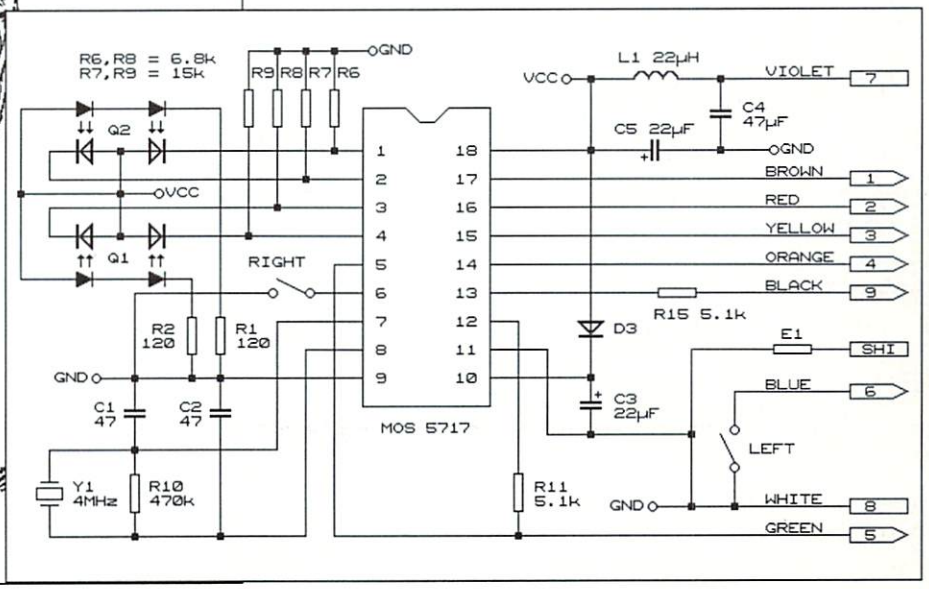
Next issue we will try printing a poster using GoDot. This also involves several of this issue's actions ...

Till then

Yours Arndt

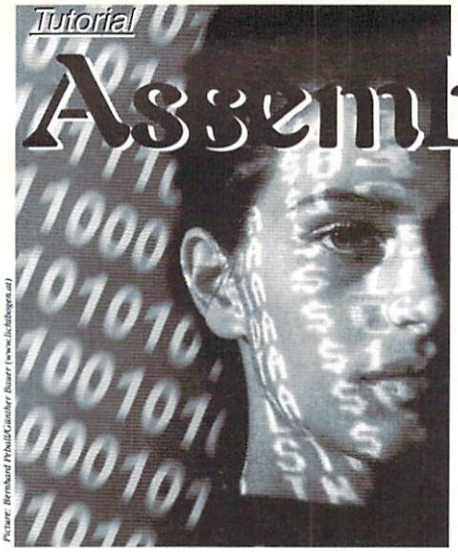


For practice purposes on disk: Skeleton.gif and 1351-mouse.gif



Tutorial

Assembly Language



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

As you may have expected, we're continuing with graphics today. I'm sure some of you know the legendary sprite balloon from the C64 user's guide and have acquired a basic knowledge about sprite programming. Now let's transfer that knowledge to assembler; you'll see it's not half as difficult as you may think.

by Wanja Gayk

Part 11

Sprites - what are those?

Unlike the Amstrad CPC and other machines of the time, the C64 has the ability to handle so-called sprites. A sprite is a little piece of graphic which can be moved on the screen independent from the rest of the visible image. Normally, the C64 can handle eight sprites, each with a size of 24x21 pixels. Why it's just 24x21 pixels will be explained later when we look at the construction of a sprite. Moving sprites on the screen is not the only thing you can do with them; there are easy ways to double their size and to check if they collide with each other or with the background image. Also, the multicolor mode we explained last issue is available for sprites, too, allowing a sprite to have three colors.

Sprites, pointers, and banks

Unfortunately, sprites are not constructed like bitmaps or character sets. They're not organized in "cards"; instead, three consecutive bytes form a row of 24 pixels, 21 of which make up the whole sprite. You might ask why it's made up of 21 lines, but the reason is simple. 21 lines with 3 bytes each makes 63 bytes in total (hex $3f). In order to tell the VIC as easily as possible where to find the sprite to be displayed, this number is rounded up to $40, and the active VIC bank is divided into segments of $40 bytes each. So all we have to do is to tell the VIC where the 64-byte segment of the sprite is located.

Well, since a VIC bank is $4000 bytes in size (16K), dividing this $4000 by $40 yields exactly 256 possible positions where we can hide the sprite. As any 8-bit connoisseur already knows, 256 is the exact number of combinations we can represent with eight bits, meaning one byte is enough to describe the position of a sprite in its bank. This byte is called a sprite pointer, and the C64 has eight of them. They're located at the end of video RAM, at the start address of the video RAM + 1016 ($03f8) to be exact.

There's a reason for this location; remember that the length of video RAM is only 1000 bytes ($03e8). However, to allow easy addressing of video RAM (by using only the upper four bits of register $d018, which hold a number between $0 and $f), this length gets rounded up to $0400, which leaves a 24-byte gap at the end of each $0400 byte segment - and instead of just wasting them, the sprite pointers are put there.

Observant readers may have noticed that if the sprite pointers are always located at the end of video RAM, moving it will also move the sprite pointers. This may either be a bit annoying or quite convenient, depending on the situation. If you move video RAM, you just have to copy those eight bytes to correctly restore the sprite pointers. Sprite data cannot be displayed from those memory areas which are invisible to the VIC (like $1000-$1fff), just like any kind of graphics data (video RAM, character set, and bitmaps).

Here's the formula you should keep in mind to calculate the value of a sprite pointer. It is: (memory location of the sprite - start address of the video bank) / $40. This value is stored at video RAM + $03f8 + sprite number. For instance, let's say we want to display sprite #0 from $2080 (which is in video bank #0 located from $0000 to $3fff). The sprite pointer value is calculated as follows: ($0280-$0000) / $40 = $82. Let video RAM be at $0400 as usual and use sprite #0 so we can get $0400 + $03f8 + 0 = $07f8. To make the VIC read the pixel data for sprite 0 from $2080, this means the command sequence is LDA #$82 : STA $07f8.

Registers, Registers...

The VIC needs a lot of registers to control the sprites, because each of the eight sprites requires its own register to hold its X and Y position. Besides these, the VIC also has to know which sprites are enabled and which are not plus several other sprite-related functions. Here's an overview of the registers and their functions:

$d000	Sprite 0 X Position
$d001	Sprite 0 Y Position
$d002	Sprite 1 X Position
$d003	Sprite 1 Y Position
$d004	Sprite 2 X Position
$d005	Sprite 2 Y Position
$d006	Sprite 3 X Position
$d007	Sprite 3 Y Position
$d008	Sprite 4 X Position
$d009	Sprite 4 Y Position
$d00a	Sprite 5 X Position
$d00b	Sprite 5 Y Position
$d00c	Sprite 6 X Position
$d00d	Sprite 6 Y Position
$d00e	Sprite 7 X Position
$d00f	Sprite 7 Y Position
$d010	Sprites 0-7 MSB of X position
$d015	Sprites 0-7 display enable
$d017	Sprites 0-7 Expansion, vertical (Y)
$d01b	Sprites 0-7 Background Display Priority
$d01c	Sprites 0-7 Multicolor Mode Select
$d01d	Sprites 0-7 Expansion, horizontal (X)
$d01e	Sprite to Sprite Collision Detect
$d01f	Sprite to Background Collision Detect
$d025	Sprite Multicolor 1
$d026	Sprite Multicolor 2
$d027	Sprite 0 Color
$d028	Sprite 1 Color
$d029	Sprite 2 Color
$d02a	Sprite 3 Color
$d02b	Sprite 4 Color
$d02c	Sprite 5 Color
$d02d	Sprite 6 Color
$d02e	Sprite 7 Color

Let's start by explaining how register $d015 works. Using this register, you can enable or disable each of the sprites. Each bit of $d015 represents a sprite; setting bit 0 enables sprite 0, setting bit 1 enables sprite 1, and so on. Well, merely turning on sprites doesn't help much; we also have to tell it where it's located on the screen.

In registers $d000 through $d00f, there's a position register for the X and Y position of each sprite. One problem remains, however. With a resolution of 320x200 pixels, it's no problem to store the Y position in a byte (as the maximum value is 199). But what if a sprite is located near the right border of the screen at an X position of, say, 290? And as if that isn't bad enough, sprites can also hide behind the screen borders, which means that position (0,0) doesn't mean the upper left corner of the visible graphic screen but the upper left corner of the whole display!

Well, 256 pixels in height are sufficient, so we can still use a single byte for the Y position. However, there are so many pixels in the X direction that eight bits aren't enough. We need a ninth bit (most significant bit, MSB) for each sprite to correctly represent its X coordinate, and these MSBs are stored in register $d010. In other words, each bit of $d010 determines whether the sprite it represents is located beyond the 256 pixel limit in the X direction. For example, if bit 0 of $d010 is set and the X coordinate of sprite #0 is set to $08, the sprite will actually be displayed at the X position $0108. On the other hand, if bit 0 of $d010 is clear, it will be displayed at $0008. In short, $d010 works just like $d015, with the difference that $d010 allows sprites to be placed at horizontal positions between $0000 and $01ff (decimal 511). (However, the whole screen including its borders are only 480 pixels wide, which means that values above 479 are of no use).

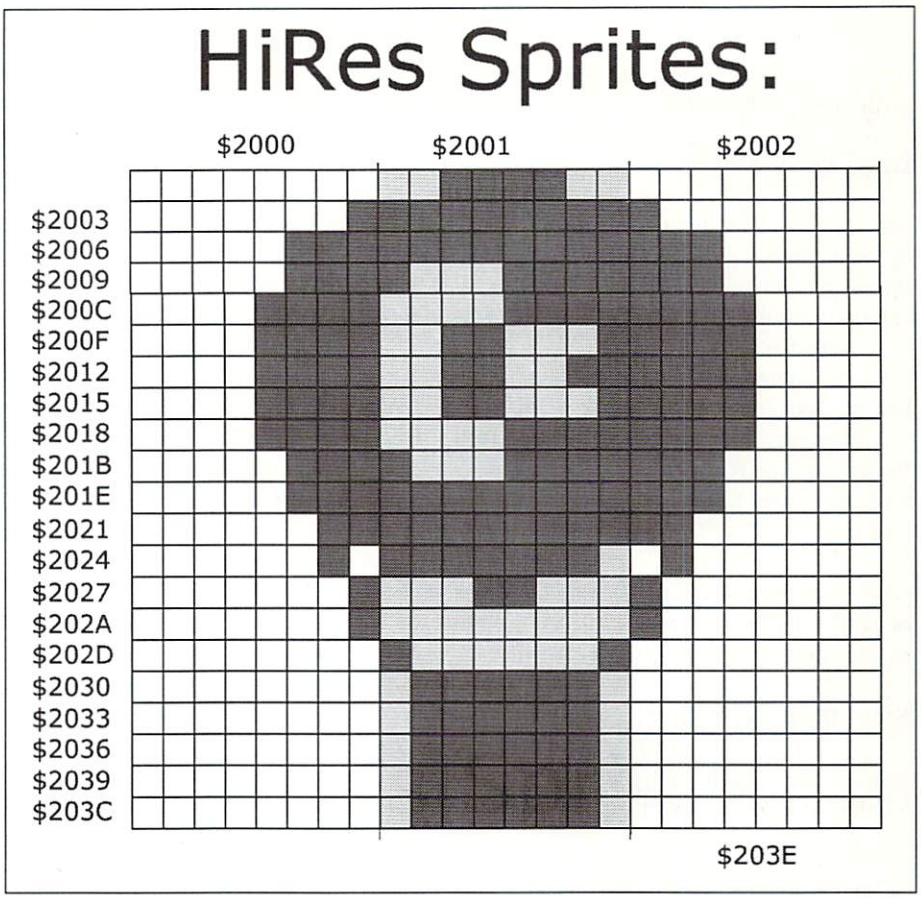
To clarify, here's a little table showing the position of the text/graphics window in screen coordinates:

Sprite position (Y,X)	Position on text screen
$0018,$32	upper left corner (first card)
$0157,$32	upper right corner
$0018,$fa	bottom left corner
$0157,$fa	bottom right corner (last card)

Now, we're able to turn on a sprite, tell the VIC where to find it, and tell where it should appear on the screen. But we've missed something important - color! The eight registers at $d027 through $d02e hold the color for each single sprite. They work just like any other color register (like $d020 which contains the border color).

Multicolor with Sprites

Register $d01c indicates which sprites are displayed as hi-res and which as multicolor.



It works just like $d015 and $d010. For instance, setting bit #6 of $d01c puts sprite #6 into multicolor mode. If it's cleared, it's a hi-res sprite. As we learned last time, two neighboring pixels are displayed as one in multicolor mode, which is twice as wide but may have one color out of a palette of three. For sprites, this means there are eight registers holding one individual color for each sprite which is also used to display it as hi-res, plus two global multicolor registers that influence all sprites. The bit pattern of two horizontally adjacent hi-res pixels determines whether the pixel color is taken from one of the registers $d027-$d02e or from one of the sprite-multicolor registers, $d025 and $d026.

Other effects

Registers $d017 and $d01d stretch the sprite to twice its size in the X or Y direction. Both of them are used just like $d015, $d010, or $d01c.

Using the register $d01b (which also works like $d015), it's possible to set whether a sprite has a higher or lower priority than the bitmap or character image. If a bit is set, the respective sprite has low priority; if it's cleared, it has high priority. A sprite with high priority covers everything else.

A sprite with low priority appears behind all bit combinations that start with a 1. In hi-res mode, this means it's behind the bitmap or characters. But watch it when you're in multicolor mode; the sprite will not only appear in front of 00 and 01 bit combinations but also behind any colors assigned to 10 or 11. Priority between sprites is determined by their number; sprites with a smaller number have priority over those with a higher number.

Crash, Boom, Bang! When sprites meet

Well, your screen won't explode, but the sprite might well do so, given that it's programmed and animated accordingly. The sprite-sprite collision register has one bit for each sprite, similar to $d015 where you turn sprites on and off. If a sprite collides with another, the respective bits in $d01e are set. Example: if sprite 0 and sprite 6 collide, bits 0 and 6 of $d01e are set.

But be careful; if the sprites part again, the bits are not reset automatically. You'll have to take care of that by reading it, e.g. with LDA $d01e. So, always keep in mind that sprite collisions are memorized in $d01e until you read them.

If a sprite-to-sprite collision occurs, bit 2 of $d019 is set, too - which means you can automatically issue an interrupt when a sprite/sprite collision happens.

When Sprites hit the wall

Normally, it would be a crazy idea to make up backgrounds, like walls or rocks from sprites, but you still want to know if a sprite collides with the background image. This is done with the register $d01f. It works like the sprite/sprite collision register; if sprite #3 hits the background, bit 3 of $d01f will be set. What we said about te $d01e also holds for $d01f; even if the sprite has long passed the background, the bit will stay set until you read it.

Also, a sprite/background collision sets bit 1 of $d019, similar to sprite/sprite collisions, so you can also control an interrupt with it.

Stop! My brain frizzles!

Don't panic. Usually, all you have to do is tell the VIC where to find the sprite data, turn the sprite on, set its position (don't forget the MSB in $d010) and its color, and you're done. For example, I've never used the collision interrupts and hardly ever used the collision registers, because I've coded mostly demos; sprite priority was high in most cases, so the sprite appeared in front of the picture, and I didn't have to worry. Yet, the potential for special effects is enormous. On the other hand, you should be familiar with the multicolor mode, because you're going to need it often. It's not really hard as long as you don't start playing with the priorities.

Sprites really enrich the C64. Therefore, let me wish you a good time experimenting with them until the next part of our assembly language course!

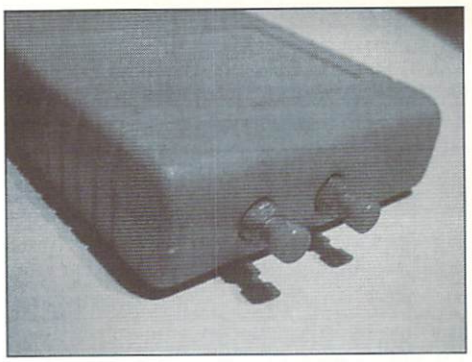
** A little anecdote. Few people know it, but no Formula 1 race car driver has ever taken part in more races than there are possible combinations of 8 bits. The record is held by Riccardo Patrese, who drove exactly 256 races. Just one more, and eight bits wouldn't be enough to count them. In order to break this record, a driver would have to take part in every Formula 1 race for almost 15 years. So it's perfectly correct to say that Riccardo Patrese is the world's fastest 8-bitter.*

*** To actually issue an interrupt upon a sprite/sprite or sprite/background collision, you have to set the corresponding bits (2 or 1) in $d01a, depending on which event will cause an interrupt. Bit 1 = sprite/background, bit 2 = sprite/sprite.*

Sources:

"All about your C64", Online Help (HTML), The Dreams.
http://www.the-dreams.de

Have you ever noticed that car owner's manuals warn that you should only use "genuine replacement parts" from the car's manufacturer? Much of the time those parts aren't any different from other third-party products, except for being three times the price. So smart shoppers buy the third-party products.



Just for Starters

Genuine Replacement Parts

by Jason Compton

Computers and cars aren't all that similar, of course. But the "genuine replacement part" issue is real. After all, Commodore made quite a few parts and peripherals back in the day, but they've been gone for six years. How can you be sure that a "third-party equivalent" will really work the same? Why, let Just For Starters tell you, of course!

Joysticks and Mice

Commodore manufactured and sold some truly awful joysticks in the 80s, but naturally you know that the Commodore joystick port is completely compatible with joysticks for the original Atari computers as well, and literally dozens of companies built some very fine custom controllers for the 64 and 128. You need to be careful of Sega and 3DO joypads, which have the same plug as Commodore-compatible joysticks but could damage your computer. Some Sega joypads can be modified to be Commodore-friendly, but it's best to let an experienced electronics whiz do the work.

Commodore introduced a pair of mice, the 1350 and 1351. The 1350 worked just like a joystick, which meant it did not need special software support, but it also didn't work very well. The 1351 was far more successful. Only a few Commodore 64-compatible clone mice have been built, including the SmartMouse and SmartTrack (trackball) from CMD. Unfortunately, the SmartTrack is no longer available from CMD. Amiga and Atari ST mice, while they look the same, are not compatible.

Disk Drives

Last time, we discussed the differences between the standard Commodore 1541 and 1571 drives, and the clones and copies other companies put out. However, there are other disk drives besides the humble 5 1/4" workhorses.

To address the growing need for more storage on a single disk and to take advantage of the increasing popularity of the 3.5" disk, Commodore released the 1581 drive, which could store over 800K of data on a single disk. The 1581 was generally reliable, but it has been surpassed by the CMD FD-2000, which can accept high-density 3.5" disks for a full 1.6 megabytes of space. The FD-2000 is also completely backwards compatible with the 1581 and with what little 1581-specific software there is.

Commodore didn't really get into the business of building hard drives, but you still need to be conscious of the standards Commodore set for disk access. Today you can choose between two types of hard drive: the IDE64 interface or the CMD HD series. IDE64 is an interface board that plugs into your cartridge port and allows you to attach common PC hard drives to the computer. CMD's hard drives plug into the disk drive serial port - the same place your floppy drive connects.

CMD went to great lengths to make working with their hard drive very similar to using a floppy. The drive works equally well on a Commodore 64 or 128. CMD replicates the functions of a disk drive whenever possible, and it is even possible to create a mirror image of a 1541 or 1571 disk on the hard drive, which makes it easier (but not foolproof) to install commercial software, including GEOS and some multi-disk games, on a CMD HD. The HD uses SCSI hard drives, up to 4 gigabytes in size.

IDE64 is less expensive (it's about half the price of an empty CMD HD and uses more common, cheaper IDE drives). It does have a couple of drawbacks, however. It occupies the cartridge port and cannot share with "super cartridges" like the Action Replay, even if you use a cartridge expander. It does not replicate the Commodore floppy routines as thoroughly, so it cannot be used with the same range of commercial software as a CMD hard drive can. Finally, the IDE64 is not compatible with the Commodore 128. It is possible to go up to 8 gigabytes of storage with an IDE64.

RAM Expansion

Although Commodore's RAM expansion isn't as flexible as some people would like (it's not like tacking extra memory onto the end of the built-in 64K), they did set a very important standard. If you can't find an authentic Commodore 1700, 1764, or 1750 RAM expansion unit (REU), there are three 100% compatible alternatives.

SSI's Super Clone 1750 duplicated the 1750's 512K of extended RAM but in a smaller case and with less of a power drain. More recently, CMD put out their own 1750 and 1750XL, offering 512K and 2 megabytes respectively. All three are fully compatible with Commodore's own REUs, because they use the REC, the special memory control chip developed by Commodore.

Other RAM expansion units do not have an REC, and as a result, they are not the same. Berkeley Softworks, the original developers of GEOS, built the GeoRAM to add 512K to GEOS. Within GEOS, you'll hardly notice the difference between the GeoRAM and a 1750, but other software does not work.

Then there are RAM devices that act more like disk drives than REUs. Models, like BBGRAM and CMD's RAMLink and RAMDrive, simply store files that can be retrieved at a very high speed, even faster than a hard drive.

Modems

Commodore never broke the 1200 baud barrier with their official brand of modems. Although a couple of other companies built 2400 baud modems that plugged directly into the user port like Commodore intended, most modems connect to computers using a 9 or 25-pin port known as RS-232.

By now, most Commodore modemers own RS-232 expansion boxes which connect either to the user port or the cartridge port, depending on the model and capabilities. Virtually every modem built at 2400 baud or above is "Hayes compatible", which is the standard modem command set supported by almost all Commodore terminal programs.

The only major issue you might have is if you are using a cartridge port RS-232 adapter, like CMD's Swiftlink or Turbo232, because some very old terminal programs may not support the way those adapters talk to the modem. Newer terminal programs, like Novaterm, have no problem finding your modem, wherever it may be.

Next month, we'll talk about how you might want to replace the brain of your 64 - the CPU - and what you can expect if you take such a leap!

(ak)

SCSI-MOs usable with HD-controllers

In this article, I'm going to describe the advantages of using exchangeable media with CMD's HD controller instead of hard disks (the advantages offered by the controller itself have already been explored in depth by GO64!).

by Johannes Schulze-Oechtering

The HD allows us to create a wide variety of different kinds of partitions - 1541, 1571, 1581, 1581-CPM, native, print buffer... Unfortunately, the total number of partitions is limited to 254. In general, we can say that using emulation partitions reduces the maximum amount of hard disk space the controller is able to handle.

Partitions

If we use 254 native partitions with 16 megs each, we have 4.3 GB available; using 254 1541 partitions, we can only handle a little less than 50 megs. This way, we lose much capacity, since a rather large part of the Commodore software requires the 1541 disk structure.

Shock Sensitivity

Another disadvantage of hard drives is their shock sensitivity, and the limited lifetime of the media.

Given these facts, it's no wonder Commodore users show growing interest in exchangeable storage media; for instance, ZIP drives are available at a low price and allow up to 254 partitions on each disk.

After I got familiar with magneto-optical drives, I wanted one of those for my computer. I bought a HD controller along with a SCSI MO drive and connected the drive to the controller. Up to that point, I didn't encounter any serious problems. The SCSI MO-drive made by Fujitsu has the same 50-pin connector as found on the hard drives used with the HD's; the power cord fits, too.

How to connect

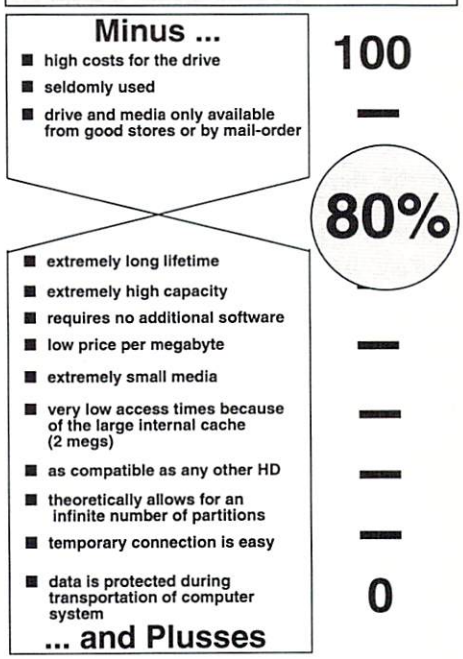
In order to temporarily connect the MO drive to the HD controller, you first loosen the four screws and open the housing. You find a neat inner construction, as we're used to from CMD. There are two loose cables inside. The one with the four-pin plug is the power connector; there's only one way it fits into the MO's power socket. The other one with the large 50-pin connector is the data cable. It has a notch indicating how it has to be plugged into the MO drive. (Note that a MO is only one of several possible hard drives).

The first try to put it to use failed, due to a slightly complicated reason. The HD controller expected a hard drive that uses sectors 512 bytes in size, while the 640 Meg media I used with the MO drive has 2048 bytes per sector. It took a while until I found a way out; MO drives are compatible with older 128, 230 or 540 meg media, and the 540 meg kind are just perfect for us! Think of it, 540 megabytes, almost as much as a CD - all this on a small 3.5" media twice as thick as a 3.5" disk.

GEOS and MOs

After I finally got my first 540 MB media, the actual testing could begin. I tried to use GEOS 64, CPM 3.0... without any failures. Even the old CPM 3.0 from 1985 worked with this combination, in contrast to the FD-2000. This way, both user port printers and the German charset could be used. (guess he's referring to a 128 system? -translator) The SCSI-MO drive behaved exactly like a hard disk; I didn't encounter any problems. Regarding the guaranteed lifetime of ten years for a MO disk, you can now start your ultimate backup!

Product Name	SCSI-MO Drive
Producer	Fujitsu
Released in	1999
Tested with	C64, CMD HD, MO Drive
Price	US $ 330, media: US $20 in computer stores for 540 MB, US $11 via mail order



CMD SuperCPU CORNER

20 MHz is super-fast - almost a little too fast for the memory installed on a SuperRAM Card, but it still works. In this SuperCPU Corner you can read how it works and what it means for timing.

by Wolfram Sang

First, let me emphasize that the following information applies to the 20 MHz mode only. Of course, you can access the RAMCard in 1 MHz mode as well, and there won't be any timing problems; the memory behaves just as you'd expect from a C64. But when you're in turbo mode, it's useful to have at least a rough picture of what goes on in the RAMCard. Knowing about these peculiarities and taking them into account while planning your programs will surely have a beneficial effect on their performance. Now, let's have a look at the structure of SIMM memory as used in the RAMCard.

Precharged?

As CMD mentions in their user's guide to the SCPU, SIMM memory isn't as fast as the internal SRAM on the accelerator card. On the other hand, SIMMs are significantly cheaper and easier to get. The reason for the relative slowness of SIMM memory compared to SRAM is called "precharge". It means that a memory position or its cell has to be prepared before it can be read or written into. Now what exactly is a "cell"? On the SIMMs used with a RAMCard, a cell always contains four consequential bytes. These cells are arranged in rows and columns, like a grid (see table 1). Register $d27b tells you what type of SIMM you're using. As an example, let's say we get a 3; what does that tell us? From table 1 we can see that we have a 12/10 SIMM. The second number is the number of bits used to address the columns (=cells). In our case it's 10 bits, meaning there are 2^10 = 1024 cells in a row. Since one cell always consists of 4 bytes, the storage capacity of a row is 1024 * 4 bytes = 4096 bytes = 4 KB. The first number of the SIMM type tells us that the row addresses are 12 bits long, so we have 2^12 = 4096 rows. Each row equals 4 KB, which makes a total capacity of 4096 * 4 KB = 16384 KB = 16 MB. This calculation is easy to repeat with other possible values, and on the coverdisk, you'll find a little example program written in BASIC for this.

Well, now we know some more, but will it pay off in any way? Let's just say that it's crucial for the timing if we have to change to another cell or even another row.

Cycle counting

Let's first examine an easy command like LDA $0400. Its execution takes four cycles, one for decoding the instruction LDA, two to fetch the address $0400. Finally, in the last cycle the value is read from the memory position $0400. And this actual read access is the crucial one; on a SuperRAM Card, this one may take up to 8.5 cycles! 8.5 cycles - did I miss something? Oh yes, half cycles do exist. To understand this, you have to imagine the clock signal as a square wave. Each pulse has a rise and a fall. Normally, the processor becomes active after each rise, but after such an ominous semi-cycle, it acts after a fall. This has no particular consequences, except that the CPU has to be re-synched for certain internal procedures, but all this happens without any action by the programmer.

This makes it obvious that routines with critical timing should avoid the RAMCard and use the SRAM instead. But since most programs aren't likely to be so terribly sensible, we'll now take a closer look at how long routines take on a RAMCard (see table 2). You'll notice that memory accesses take a minimum time of 1 cycle as long as you stay in a precharged cell. If you change to neighbouring cells sequentially (without skipping any), there's no delay either. That's because the RAMCard has an electronic controller that "assumes" this kind of access and optimizes it, which is very reasonable because this is the way code is usually executed.

However, there are some occasions (like branches, for instance) where you have to change to a non-adjacent cell located in the same row. This will cost you two cycles or even 3.5 if you have to change to another row. These values hold for read accesses. Strangely, the timing for write accesses is simpler; writing into the current row takes you one cycle or otherwise 3.

As we can see, accessing a new row takes the most time. Looking at the two possible types of a 16 MB SIMM, you might conclude that an 11/11 one is slightly faster than the 12/10 type. That's right; since it has fewer and longer rows, this reduces the probability of row changes. However, this speed advantage is likely a very limited one - so get the 11/11 if you have a choice but don't despise a 12/10 just because of this!

Refresh

Unfortunately, we're not done yet; we still have to know how to deal with the so-called refresh. It's vital for the computer; without it the RAM memory would forget the data stored in it. In a C64, the refresh is handled by the VIC, and it normally does the job in the background. However, at 20 MHz this activity can't stay hidden; therefore, a refresh signal is generated every 10 microseconds or every 200 cycles, which at worst can prolong a read cycle up to 8.5 cycles. The worst case occurs if the CPU wants to read the RAM right after a refresh has started. If we're lucky and the read access comes near the eni of the refresh, the delay will be shorter. There's no way to predict the refresh or adjust the program timing for it (at the present time, that is). But that's not too tragic, as the power of the SCPU has not

nearly been exploited at the moment. Yet, it's another reason why routines with crucial timing should not be stored in the RAMCard.

An example

In order to illustrate the above, let's examine the following code which is running in the fast SRAM. The example is constructed such that the type of the SIMM doesn't matter:

```
020000 SEP #$30
020002 LDA $020100
020006 STA $030000
...
```

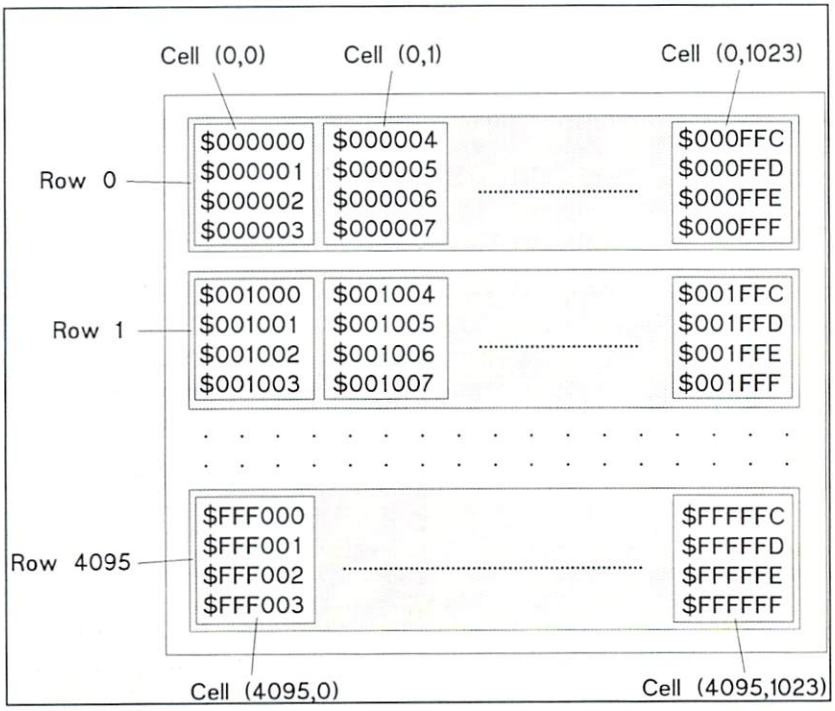
Before the SEP opcode can be fetched, its cell ($020000 - $020003) has to be precharged. The time used is 3.5 cycles plus another one for the #$30 operand, which is stored in the same ell; this means the command takes 4.5 cycles in total instead of 2. The LDA command and its operand bytes are stored sequentially, so only 4 cycles are needed to read these 4 bytes. However, reading the value in $020100, which is in another cell of the same row, takes 2 cycles according to table 2; therefore, this command takes 6 cycles instead of 5 (the accumulator is only 8 bits wide). Then the processor has to change back before it reads the next opcode at $020006, taking another 2 cycles. Again, the target address for the STA is stored sequentially and takes 3 cycles to read, which means the STA command takes 8 cycles instead of five. The whole routine takes 18.5 cycles (without any refresh), while it could be executed in 12 if it ran in the SRAM.

This topic may be somewhat different than usual. But if you just get a bit used to the ideas presented here, you'll get a feeling for what's better put into the RAMCard and what's not suitable for it. I hope it has become clear why data put in the RAMCard should be stored as sequentially as possible and why you should take care to use the fast SRAM wisely.

Value of $D27B (53883)	SIMM-Type	Number of Cells	Length of one row (in Byte)	Number of Rows	Total memory (in Byte)
0	9/9	2^9=512	512*4=2048=2K	2^9=512	512*2K=1024K (1MB)
1	10/10	2^10=1024	1024*4=4096=4K	2^10=1024	1024*4K=4096K (4MB)
2	11/10	2^10=1024	1024*4=4096=4K	2^11=2048	2048*4K=8192K (8MB)
3	12/10	2^10=1024	1024*4=4096=4K	2^12=4096	4096*4K=16384K (16MB)
4	11/11	2^11=2048	2048*4=8192=8K	2^11=2048	2048*8K=16384K (16MB)

Read inside a cell	1 Cycle
Sequential read inside row	1 Cycle
Read from a new cell inside row	2 Cycles
Read from a new row	3.5 Cycles
Write inside row	1 Cycle
Write to a new row	3 Cycles
Read during Refresh	up to 8.5 Cycles
Write during Refresh	up to 8 Cycles

Memory map of a SIMM



GO64! Bazar Form

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

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

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

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

Address
CSW Verlag
Goethestraße 22
D-71364 Winnenden
Germany

I assure that I own all rights concerning my offers.

From:
Last Name/First Name

Street

ZIP/Town/State

Credit Card no./expire date/holder

Date/Signature

Masthead

Publishing Manager: Enno Coners
Editorial Staff: 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: P. Assion, O.Biasin, J. Butterfield, J. Compton, G. R. Gasson, M. Hinzmann, M. Jacobi, F. Kottira, J. Schulze-Oechtering, B. Thomas
Supervisor: A. Dettke **Int. Supervisor:** R. Bernardo, N. Dannenberg, G. Gasson, D. Mohr
Translations: Bettina Beck, Matt Bodnarchuk, Stephan Lesch, Ranjith Ekanayake Mudiyanselage, Kendra Thiemann; **NTSC-Fixing:** Wolfram Sang, Wanja Gayk
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 and Marco Platzdasch
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

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

C128, 1581+71internal, Mon.1084S, 1571, RAM1764, StarLC24C, cartridges, >1000Disks, GEOS128, / Scanntronik- Scanner, Videodigitizer, C116, books, and more Tel. +49 (0) 40 / 535 82 58

Contact Corner

Searching for contact to other C64 users? I can help you. Commodore Connection Line, Oliver Biasin, Schwalbenweg 15, 91220 Schnaittach, Tel-Fax +49 9153-1831
E-Mail Connection-Line@gmx.de,
http://home.t-onlie.de/home/mysoft/ccl.htm

Visit one of the last BBS' of Germany. Fast and colored, just take a look! +49(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 Demo Scene

I still remember how it began. It was in 1989, and I was sitting in front of Amy TV, staring fascinatedly at the effects shown in a well-known C64 demo, and began to wonder. "What kind of people create such a show? How do they do it, what programs are they using? And even more, what's their point in it?"

by Marko Jakobi

As I guess many of you have similar questions, I'm writing about the demo scene, how it came into existence, its purpose and about its organization. It's a rather complicated matter, but I'll try.

How demo groups developed

In the middle of the 80s, when more and more interesting games appeared on the market, the developers started inventing methods to protect their games against illegal copying, and implementing them in their programs. The users, especially those with small incomes, didn't find it acceptable at all, and simply began to (illegally) remove or circumvent the copy protection schemes. The people who could do this were soon known as 'crackers'. In order to immortalize their work, the crackers put so-called intros in front of the games they'd cracked. These intros presented the cracker's alias, his greetings to other members of his guild, and similar things. As time went on, the graphical effects shown became more and more impressive, and that's why these effects were soon made into whole programs of their own that presented a series of effects along with music, instead of mere intros. The demo was born, along with the demo scene.

The purpose of the demo scene

The 'why' is easy to explain: Demos are fun, and are meant to be fun! That's the common hobby shared by the countless members of this scene, and naturally, there's also some competition involved. Most of the time these coders do it for prestige, rather than commerce. These competitions are held at computer parties, where visitors numbering as high as 3000 are not uncommon. But, there are also lots of small meetings held by sceners who know each other personally. These meetings have one thing in common, and that's the fun in them. That's how it's meant to be.

What's a demo?

In simple terms, a demo is a program showing graphical effects accompanied by music. While early demos were a day's work, they soon became larger projects that took weeks to finish. This caused a need for specialization - instead of a single person doing the whole demo, there were specialists for each detail - programming, graphics and music - always trying to get the last out of the computer in their field.

How the demo scene and groups are organized

Therefore, most scene members are also members of a group. These groups are constantly trying to outplay each other with new and better effects, and so the quality of the demos is raising constantly - at least that's how it should be. Each member of a group fulfils one or several tasks.

1. The coder (programmer)

The coder is the core of a demo group, the one who gets the works done by the other artists (graphicians, musicians) and uses them in his code. It's usually the coder who decides about the design of a demo.

2. The graphician (gfx'er, painter...)

He's responsible for the pictures and logos shown in the demo, which he can't always style to his own liking, as he has to bear in mind what the finished demo is planned to look like (gloomy or colorful, etc).

3. The musician (msx'er)

The same holds for the musician, with respect to music.

4. The swapper (trader, spreader...)

It's the swapper's task to spread the releases of his group as wide as possible. These products may be demos, but also disk magazines, votesheets for charts (read on below), or productions by single group members. The basic rule for a swapper is, the more active he is, the more contacts he has, the better.

5. The magazine editor

If a group releases a magazine, they need an editor to write the text. He looks out for news from the group or the whole scene, writes party reports, etc. In fact, writing takes most of his time, very seldom does he manage to other activities like coding, painting, composing or swapping.

Disk magazines

A diskmag, or 'mag' in short, is simply an electronic newspaper with articles concerning the scene, including all kinds of relevant news, party reports, and other information. One specialty are the charts (rankings), which cover most categories, like demos, coders, graphicians, musicians, and so on. The size of the charts varies between mags, but usually they're about ten positions for each category. In order to determine the rankings, each mag has its own voting sheets, which are filled in by the sceners and returned to the mag editors. The address column is mainly of interest to swappers, since it lists lots of potential contacts in the scene. The outward form of mags vary strongly, as each demo group strives to make the visuals and user interface of their mag as outstanding as possible. Another point that gets special consideration (in most cases, that is), is to ensure the objectivity of the texts published in the mag.
(ad)

hYbRiD 1 - A Common C64 Tower?

"This is really old hat", you will certainly say by now. As is very often the case, things are not what they seem.

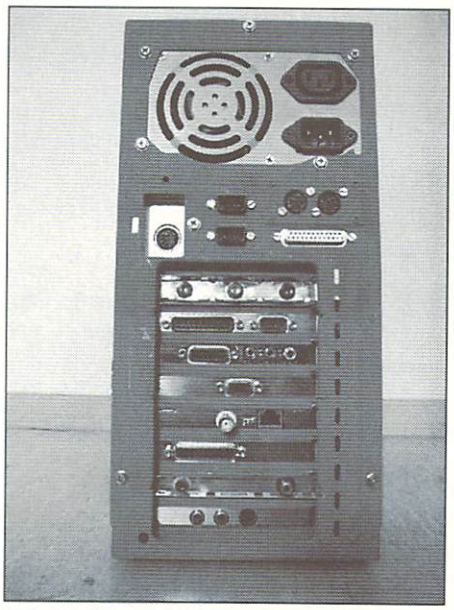
by Patrick Assion

Prelude

Once again there was a computer party, and as usual, there was the question of "What hardware parts am I going to take with me?". To begin with, I was bringing a C64C with an Action Replay MK V, 1541-II, 1084 monitor, as well as the respective power supplies and cables, disks, sleeping bag, provisions, and so on. So far, things were always as they had been, but even with this minimal amount of equipment, together with three passengers, it went far beyond the room in my VW Golf 'limousine'.

And if my party system was to have a capacity similar to what I was used to at home, the number of individual components would increase even more: a second 1541 for easy copying, a 1581 for archiving bigger amounts of data, PC laptop with link cable and modem for data exchange via Internet and e-mail, and so on, and so on, and so on.

Reflecting on all these items, you might think that I needed a van for transporting this equipment. There had to be a way out. And it wasn't long before the basic idea for the hYbRiD 1 was born - a combination between a C64 and a PC in a single, compact case.



TO BE CONNECTED - connectors as far as you can see

Well-planned is half-way built

As the days progressed, it became more and more clear to me what features were needed. Taking into account my existing hardware, I first checked these features, whether they were technically possible. Then I scheduled them accordingly and wrote them down in a what-to-do booklet:

- two systems, one case
- one main power switch for both
- one reset button each for PC, C64, and disk drives
- speakers common for both computers
- C64 with C128D keyboard connection
- C64 with internally hardwired Action Replay MK V
- C64 screen on PC monitor (via video capture or TV card)
- C64 serial port, video, keyboard connection, and joystick ports transferred to the outside of the new case
- linked system use possible (C64 user port to PC's LPT port)
- PC with additional LPT port for printer access

Every part to be produced, changed, and inserted was carefully considered, measured, and written down in all its details - work that really did pay off later on. The resulting material requirements (see box) was then by far more comprehensive than I initially expected, and due to the short time available (only two weeks to the party), not easily implemented.

After this rather theoretical part of the project, the practical part followed, which as it turned out, had some less pleasing surprises to offer.

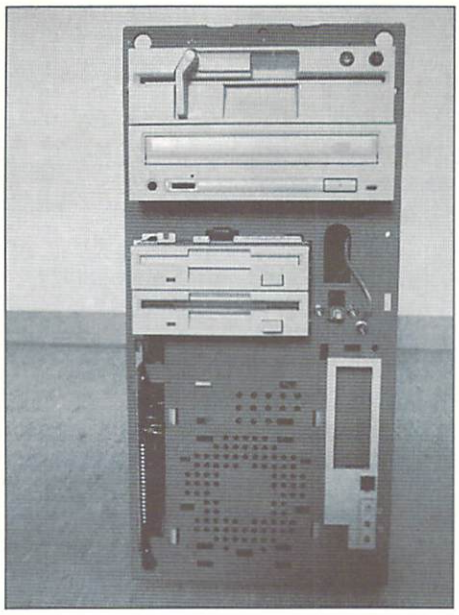
There is room even in the smallest cottage

While the placement of the disk drives was a direct result of the slots available in the tower (the drive boards being attached above or below the drives), finding an appropriate location for the C64 motherboard proved to be much more difficult. The case had been designed for housing a standard PC in BAT design, and it wasn't too big. The C64 board had to be accommodated in the remaining space, but this was restricted to an area in the lower left part of the case, 30 mm wide, 380 mm long, and ranging from 150 mm (at the front) to 180 mm (in the back) at height.

Just as expected (Murphy's Law), the C64 motherboard was just a few millimeters too long for the standardized PC mini-tower case. In addition, some metal work was required, in order to add the missing 5 mm clearance in the interior.

To this end, a recess of 148x24mm was worked into the front of the chassis at the left bottom, and the corner brackets required for

The hYbRiD 1 in all its glory (although still without 1581) as first seen on Computer's Hell Party 1999 in Berlin



Front view of the hYbRiD 1. The recess at the lower left side for the C64 motherboard is clearly seen

the board were attached. Doing this, I made sure to use one of the newer, narrow C64 boards, because there was only enough room for this below the 3.5" disk drive slots.

The electric connectors of the C64 also did't really fit the standardized world of modern PC cases, which was why a saw and rasp was put to use again. A part of the case's back was removed and replaced by a GFK plate with round DIN connectors for the video and serial ports.

For the remaining C64 connections that needed to exit from the case (joystick ports and keyboard), the existing openings in the case's back could be employed.

Plastic Surgery

During the next step, the disk drives were given a new "face", so that I could integrate the look into the casing.

My 1581 proved to be very cooperative here, since it was a standard disk drive which has been used in Amigas and PCs. After changing the eject button, the front panel of a TEAC 3.5" disk drive could be put onto the case without any problems.

Things were more difficult with the 1541-II. It had been produced with two different mechanisms that mainly differed in the location of the eject lever. I chose the CHINON drive, because its mechanism - contrary to common belief - was frequently used in the PC arena (e.g., the Commodore PC10-III/PC20-III), which made it easy to find a fitted front panel from the flea market.

As an alternative, I could have also used an old 1541 (brown front panel, fawn case). This drive was already equipped with a matching front panel, and there were tons of them available. Some adjustments in the electronics would have been required, though. Since all changes had to be reversible, this option was therefore not considered.

In addition, the CHINON mechanism of the 1541-II already had two tapped holes on its upper side, which at least helped me with the problem of attaching the board.

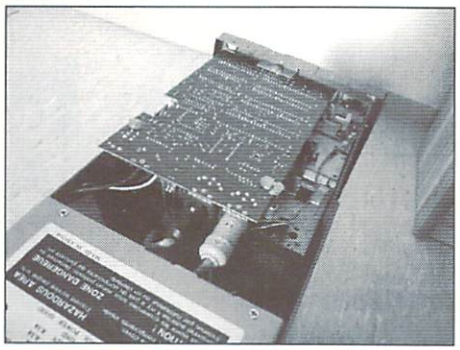
With no fitted DIN connectors available, and in order to avoid possible defects, the power connectors were removed from the drives' boards and replaced by common four-pin PC power connectors. Adapter cables (see sketch 1) were also an alternative, but because of the danger of shorting out other modules, these would have been made safe against slipping off.

The last change on the disk drives was to relocate the power/activity LEDs on the front panel. While the 1541 activity LED was really all that's necessary to view all operating states, the 1581 also required mounting the power LED, because it also showed drive errors in addition to general readiness.

"Houston, we have a problem."

When testing the disk drives afterwards, I found out that my 1581 had unfortunately not survived this operation.

The drive had worked without any problems in an Amiga 500, and the Amiga's identical drive failed to work with the 1581 board as



The 1541 disk drive's board installed upside-down

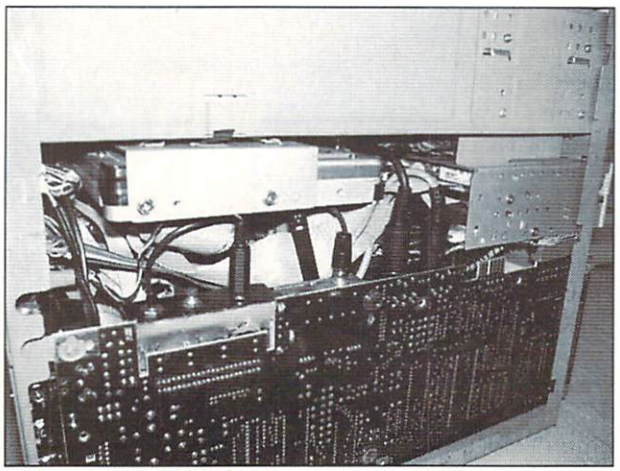
well. This would also explain why Commodore replaced the mechanism in the course of its production. From the same manufacturer, they took another one that would work properly with the board.

Having passed all these tests, and exchanging both the board and the drive, I found it was a problem in the communication between the electronics and this specific type of Matsushita mechanism. Strangely enough, the following tests with my second 1581 showed the same problems again but not as frequently as before. Anyway, the problem couldn't be solved in the short time remaining (just two days to the party), so that I abandoned inserting a 1581 for the moment.

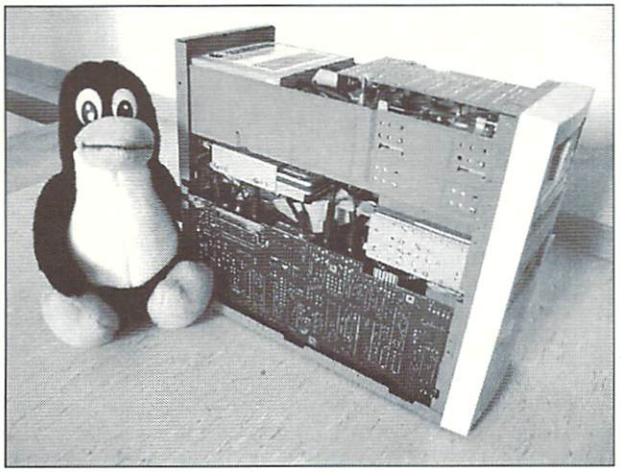
By now I can say this - the problem nearly always manifests with disks that had either been used with other systems (Amiga, PC, etc.), or that are defective.

Chip Tuning

Now it was time for the modifications to the C64 motherboard, which had to be undertaken due to the really small space available in the tower case. The power connector was removed and replaced by a 4-pin (5V) and a two-pin (9V) PC power plug



Having a look at the inside quickly explains the hYbRiD 1's nearly 15 kg of weight



Compact Class - as if of one body

(see sketch 1).

Since the C64 required additional 9 volts AC, which the PC power supply didn't provide, a matching transformer had to be mounted. On its input side, this device was connected internally to the switchable 220V output from the power supply, and on its output side with a 2-pin plug fitting to the modified C64 board.

After being extracted, the two original joystick ports were relocated to the case's back and were connected with the C64 board by a 9-core ribbon cable.

Now another problem intruded. In spite of my best efforts, the space was insufficient to mount a user port plug. Fortunately, I didn't really need the last pair of lines (12 and N). They were just ground contacts which were also connected to the port's pins 1 and A. Thus, the plug was simply sawed to the proper size until it fit in beside the case's front panel. In order to keep the plug from shifting during its later operation, it also received notches (planned anyway) and two small GFK stops (see sketch 2a).

Then followed the rebuilding of an even earlier rebuilding. As mentioned before, hYbRiD 1 was supposed to be equipped with an Action Replay MK V. To be precise, it had already been equipped that way. I had purchased the cartridge some years ago, but very soon after that, it ran into contact problems. Being annoyed by permanent system crashes, I had simply soldered it onto the board and mounted/moulded the two buttons (reset and freeze) into the former expansion port.

I realized that this was not reversible without some effort. I left the module in its hardwired state and simply branched out the connection lines for the two buttons so that operation from the outside was possible.
As you can surely imagine, things became more than cramped in the case by now, which resulted in excessive heating up of things. In order to fight this, I gave the most important chips (CPU, VIC, SID, and CIA) heatsinks and attached them by using double-sided, self-adhesive, heat conductor foil from the PC processor arena.

An existing, additional fan built into the mount in the front panel of the case couldn't do any harm either. While the power supply ventilation pushed the air out of the back of the case, the additional fan pulled fresh air in from the front which ensured sufficient cooling.

Cable Harnesses and other Devices

Now the appropriate connection cables were missing. Theoretically, many of the original cables could have been used, but I didn't want to further fill up the already cramped casing.

Now the comprehensive preparations paid off, since nearly all the cables had to be changed in one way or the other.

In detail these were:

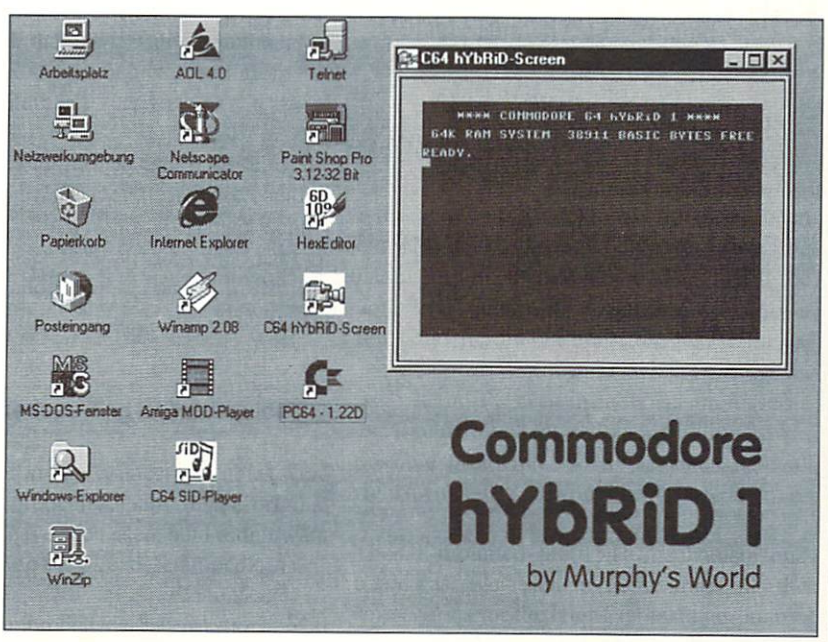
- Video cable (luma/chroma plus composite):
  C64 video output to DIN connector 8-pin and cinch connector (yellow) for connecting to the PC video capture card
- RF cable:
  C64 RF modulator to cinch connector (black) for connecting to a TV's antenna input
- Serial cable (no reset line):
  C64 to 1541-II
- Serial cable:
  1541-II to 1581
- Serial cable (with reset button):
  1581 to 6-pin DIN mounting connector for connecting other external devices
- PC64 link cable:
  C64 user port to PC LPT (see sketch 2)
- Connection cable for C128D keyboard:
  (see sketch 3)

Moving Day

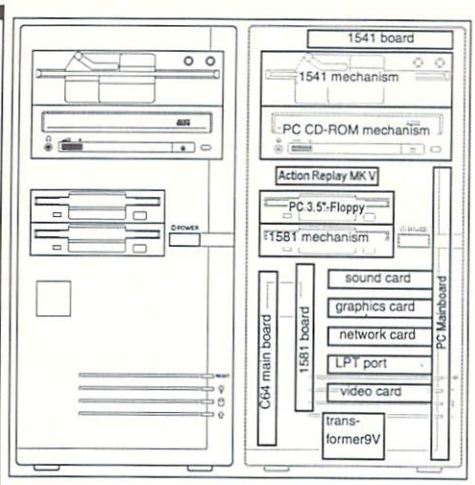
After all preparatory work had been completed, I was now able to move the hardware into its new home. This had to be done very quickly now, since it was only a few hours before our departure for Berlin.

At first the PC was installed in the case as usual and tested for proper operation, since possible problems had to be taken care of right away before it would be too late later on.

Then I mounted the 1541 mechanism and board as well as implementing all the connections for the later C64 installation into case with power supply, PC motherboard, and drive.



The C64 screen can be shown on a PC monitor by means of a video card



Optimal utilization of space - nearly no corner remains unused

Now there was only the C64 motherboard with the Action Replay Module left, which took some time getting everything hooked up due to the case being so small.

The installation was finished by locking all clamping bolts with hot adhesive, inserting isolated spacer bolts, and closing the case, of course.

Crucible

As already mentioned before, the idea for this reconstruction developed from practical considerations - namely reducing the transport problems when going to computer parties. And this was just about to be put to the test.

I assembled the equipment and found it "travel-friendly". Besides the computer, the system only needed a VGA monitor, two keyboards, and a mouse to operate. Just to make sure, I also took my 1084 and some other equipment with me, because the system had yet to be tested.

After being switched on at the party, hYbRiD 1 ran for more than 24 hours non-stop and was used intensively by different people during that time. From demos to coding to converting, both the C64 and the PC subsystems were active without any difficulties arising. Only the heat which developed turned out to be bigger than expected, but it still remained in tolerable limits. The exhaust air's temperature was at about 35 degrees to 40 degrees centigrade when the surrounding temperature amounted to 20 degrees centigrade.

Rebuilding Advice

It was quite a lot of work to turn the original, spontaneous idea into a working computer.

From a technical point of view, an idea like this didn't have high requirements; most changes only consisted of adapting the different connectors and creating new case mountings.

It is of greatest importance that you take your time in precise planning, because otherwise, you may very well fail due mostly to minor problems. For instance, acquiring 4-pin DIN plugs that fit the disk drives' power supply proved to be a nearly unsolvable problem.

So as not to risk the "health" of your beloved Commodore hardware, I advise the use of identical, defective boards for measuring the mounting points. The importance of such safety measures was demonstrated when I slipped a punch while marking the drill holes; I punched an additional hole into the C64 test board. That's life!

Of course, I am also willing to help with advice should there be problems with any of the described construction steps.

For those of you who would like to have a closer look at the hYbRiD 1, check out the "JMP $07D0 Party" from December 28 to 31 this year or the website of Murphy's World -

The COMMODORE Place
http://www.murphys-world.de/cbm

And now have fun tinkering...

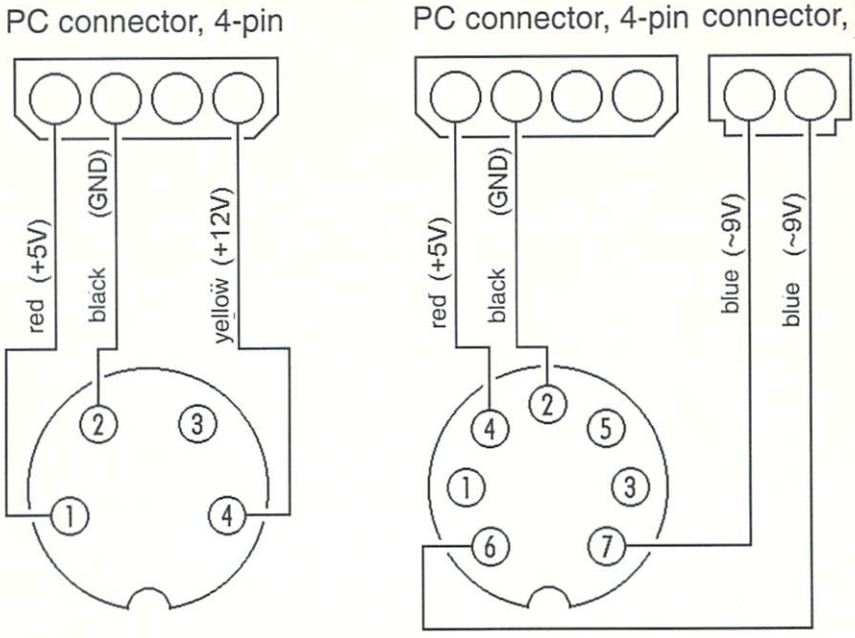
Yours,
Murphy

Materials required

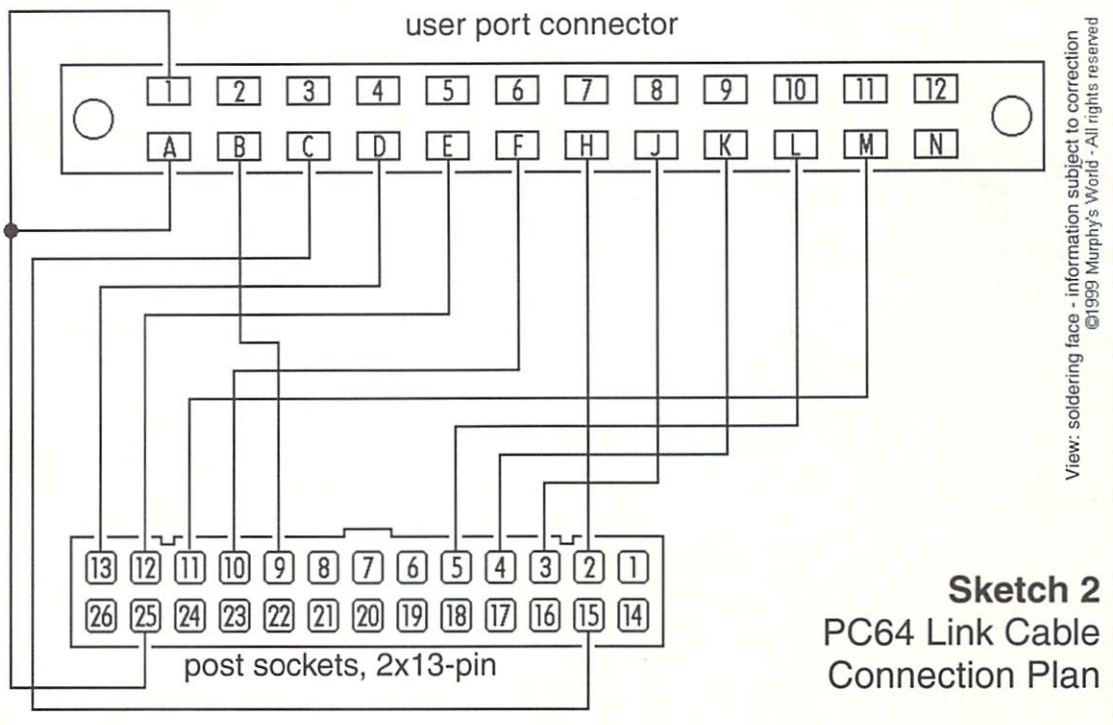
Case:
1x PC mini-tower with 2x 5.25" and 2x 3.5" mounting slots
1x 200 watt power supply
1x transformer 9V, 1.1 A
1x Case fan 12V
2x GFK plate, 100 x 50 mm, 1 mm thick
5x plate metal strips, 50 x 10 mm, 1 mm thick
1x brass pipe, 4 mm diameter, 60 mm long
1x can of zinc spray
1x adhesive text "hYbRiD 1"
1x adhesive Commodore Logo
- screws and nuts (3 mm fine thread)
- one bar of hot adhesive for locking the screws
- several small cable wraps

PC Subsystem
1x Pentium motherboard
1x Pentium 133 processor with fan
2x 64 MB EDO-RAM
1x 850 MB IDE-HDD
1x CD ROM 6x ATAPI
1x 3.5" disk drive
1x VGA graphics card
1x video capture card
1x sound card with two internal inputs
1x Commodore PC keyboard
1x serial mouse

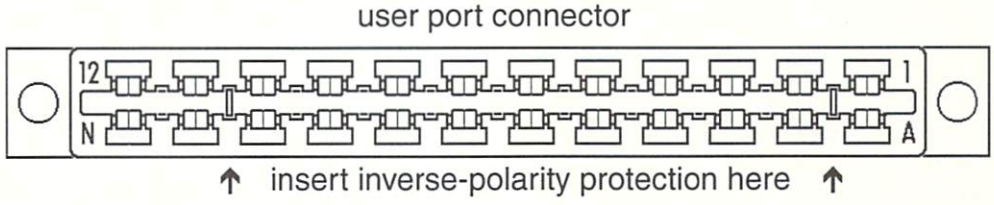
C64 Subsystem:
1x C64 motherboard, narrow design
1x 1541-II mechanism (Chinon) and board
1x 1581 mechanism (Matsushita) and board
1x Action Replay MK V (board)
1x 5.25" PC disk drive front panel
1x 3.5" PC disk drive front panel
1x C128D keyboard
1x 1351 proportional mouse
3x CMOS heatsink long for CPU, VIC, and CIA
1x CMOS heatsink short for SID
1x adhesive heat conductor foil 50x50 mm
1x Sub-D connector, 25-pin female
5x DIN connector, 6-pin
1x DIN connector, 8-pin
1x DIN mounted connector, 6-pin
1x DIN mounted connector, 8-pin
1x mounted jacks, 3.5 mm Stereo
1x jack plugs, 3.5 Stereo
3x PC power connector, 4-pin male
1x PC power connector, 2-pin male
1x PC power connector, 2-pin female
1x board connector, 2x 12-pin
1x post sockets 2x 13-pin
1x post socket 20-pin
3x switch, pushbutton - momentary contact
2x LED 5 mm with socket (1x red / 1x green)
2x LED 3 mm (1x red/ 1x green)
1m control line, 6-core shielded
1m control line, 10-core shielded
1m ribbon cable, 9-core
1m ribbon cable, 20-core
2m braided wire, .75 mm, sorted by colors



Sketch 1
Power Supply
Connection Plan

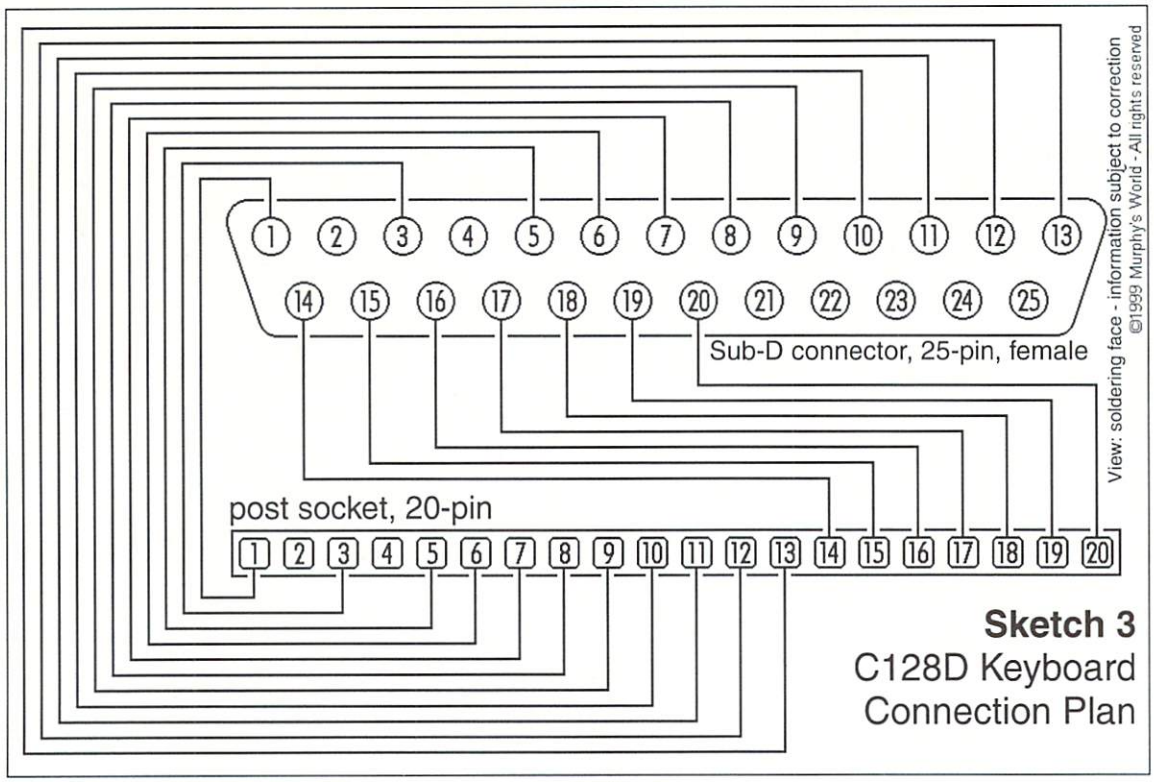


Sketch 2
PC64 Link Cable
Connection Plan



Sketch 2a
User Port Inverse-
Polarity Protection

View: connector side - information subject to correction



Sketch 3
C128D Keyboard
Connection Plan

Routine $ab1e

Our answer to Daniel Kelber's question in GO64! issue 6/1999 where he was asking for the C128 counterpart for the C64 routine $ab1e:

The counterpart for the C64 ROM routine at $ab1e is located at $55e2 in the C128.

John Selck (Graham/Oxyron)

Questions?

Don't hesitate, just ask the many readers of GO64!/Commodore World Magazine! We're pretty sure there is someone out there who knows the answer to your question. It doesn't matter if your question is hardware or software related, just write it down and send it to us. We will publish as much as possible, and if you wish to, we will also include your address so that people can contact you directly.

Our address

CSW Verlag
Readers' Corner
Goethestrasse 22
D-71364 Winnenden
Germany

You can also contact us via E-Mail under

`go64@c64.org`

There is also the possibility to participate in our new mailinglist. For this you will need an E-Mail access. To subscribe, send a mail to:

`listserv@giga.or.at`

with the content:

`SUBSCRIBE GO64-LIST Your Name`

The mailinglist-address is:

`go64-list@giga.or.at`

Ingeniously Simple Conversion

In reaction to a question by Torsten Baade looking for a conversion method from decimal to 'normal' byte in GO64! issue 7/1999 and the answer received from Marco Baye:

In issue 7/99, two quite daring methods for converting BCD numbers to normal binaries were presented. I consider neither the method using a table of 256 bytes nor the method of "counting" to be of any use. The first needs too much memory for such a tiny problem, while the other method very quickly eats up computing time, since the average conversion of a single BCD number costs 400 cycles.

The "best" method for converting BCD numbers to hexadecimal:

```
dec2bin_best

 lda dec_byte
 pha
 and #$f0
 lsr
 sta $02
 lsr
 lsr
 adc $02
 sta $02
 pla
 and #$0f
 adc $02
 .end
```

This method is but 5 bytes longer than the counting method, but the speed is far greater. It uses exactly 30 cycles in all cases. In addition, the x register is not used at all.

If you wish to save another 3 cycles, though, you should replace PHA and PLA by TAX and TXA.

John Selck (Graham/Oxyron)

FD 2000

Is there anyone in the know about FD 2000 disks? The Nordic Power's loader for the 1581 works, but saving destroys at least the directory. Has someone examined the disk format? Can anyone save some of the data?

Hoogo/Padua

The 1581 stores its BAM and directory on track 40. When emulating a 1581 disk, the FD-2000 necessarily does the same. However, when you're working with a Native-Mode partition, the directory and BAM are stored on Track 1 instead. When saving, Nordic might be reading track 40 of your Native Mode partition, which of course contains data and not a directory/BAM. In so doing, it is probably getting the idea that track 1 of the disk is empty, and hence it overwrites your directory with the data you are trying to save. Alternatively, it could be using areas of memory inside the 1581 that either don't exist or have been re-assigned in the FD2000, which can also lead to a trashed disk.

Nate Dannenberg

Flawed Colors and Power Supply Question

Answer to Sirko Neuenfeldt who complained about his C128D's flawed colors in 80columns mode in GO64! issue 9/1999 and who also had a question concerning the power supply:

Concering the monitor problem: It seems as if the green color was defective. Check if there is a contact on both ends of the cable connecting pin 4. If not, then the cable is defective and needs to be replaced. If it's not the cable, try running the computer with another monitor (if you can't find anything else use an old pc CGA/EGA monitor). The computer is defective if the problem remains, otherwise it is the monitor itself.

Peter Karlsson

Happy flashing

If I turn on my 1541-II, the LED would blink once per second and the drive won't respond to the computer's commands.

Andreas Gansen

When the 1541 can't start up, it will report an error condition according to the number of flashes on it's activity LED.. I can't remember the correct error codes, but I believe they're listed in the 1541 manual. One place to start looking would be the 1541's DOS ROM. Check this and the two 6522 VIA chips by substitution. For your information also, the CMD Hard Drive exhibits the same behaviour if it can't start up, it flashes an error code with its lights.

In your case, you're getting an Error Code of 1 (one LED flash followed by a long pause, and repeat). Hope this helps!

Nate Dannenberg

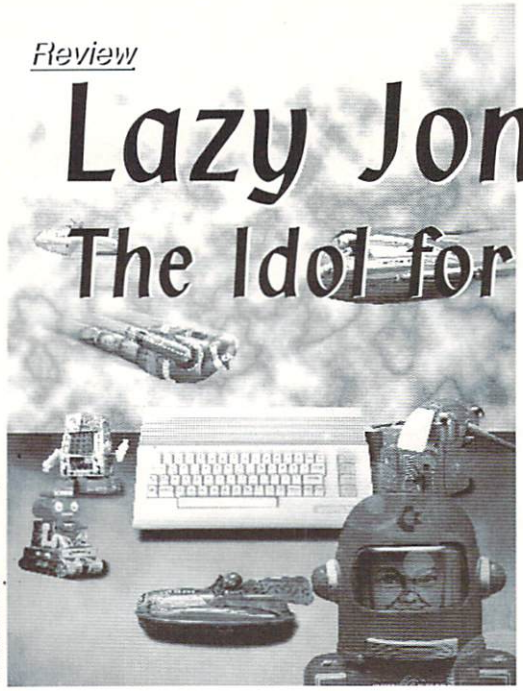
Do you know it?

Do you know the answer to a published question? Are you able to help? Write it down, mention the issue number and the title of the question. We are happy with every single letter we get, and by writing, you are actively supporting other readers and therefore the C64 scene. Don't hesitate! Send your answer to our address via snail mail or E-Mail, or participate in our mailinglist.

Review

Lazy Jones

The Idol for Every Gamer



Picture: Sebastian Bach

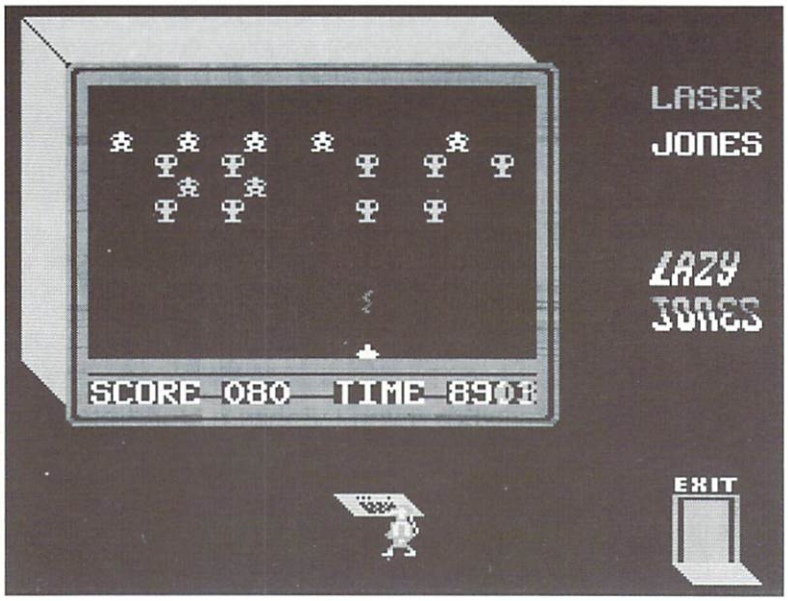
Lazy Jones's Dancefloor Hit

At this very moment the melody playing on the "Stardust" arcade machine, which is one of the catchiest tunes of the whole game, is scoring a hit in the German dancefloor charts. The group Zombie Nation published their song based on this melody under the title of "Kernkraft 400" (Nuclear Power 400). This techno version stays very close to the original and can only be recommended to all C64 fans.

David Whittaker is a widely known music composer for computer games (his music accompanies games such as Back to the Future II, Exterminator and Super Wonderboy). His sounds reappear in various other surroundings in altered form and are easily recognizable. In 1984, this same composer set out to design a game entirely on his own. The result is a gem he calls "Lazy Jones."

by Helmut "Katakis" Bieg

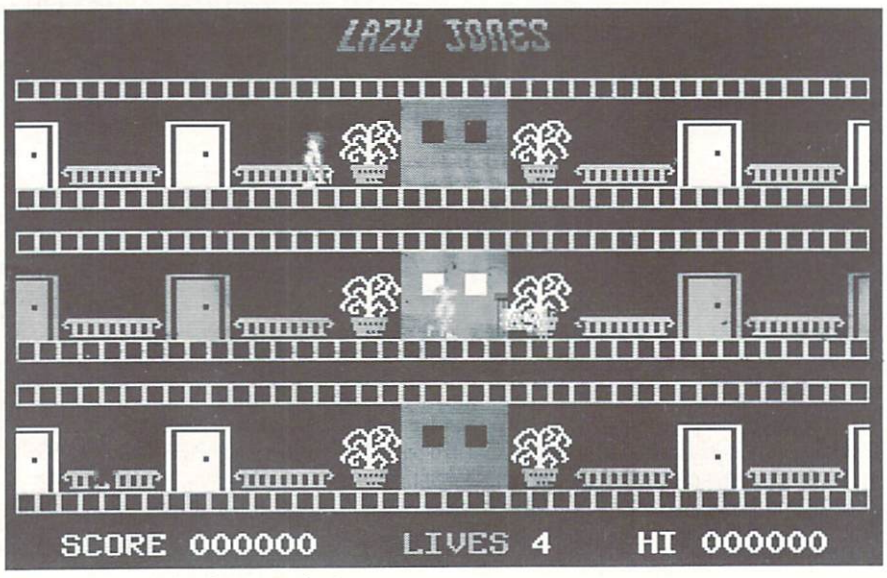
Some of you might now ask: "What's that, Lazy Jones?" For all those who are plagued by that question, here's the enigmatic answer: "There's a Lazy Jones in all of us." Lazy Jones is one of the games that too few people have ever thought of coding. The protagonist himself, from whom the game got its title, is a game addict who roams a hybrid of a hotel and amusement arcade in search of good arcade machines. Behind every door of the three-story building (the floors are connected by elevators) there's a small arcade game hidden which can be used once per game.



The one true classic of all arcade games

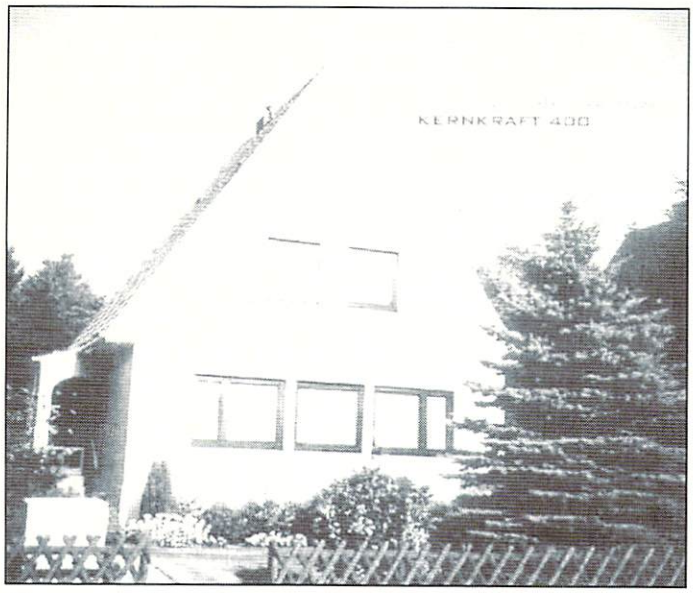
Lazy Jones - a game of games

This is about all one needs to explain the story - one Lazy Jones game consists of various games. The basic idea sounds simple - but it is, in fact, brilliant. To design a computer game in which the protagonist does nothing but play games is a stroke of genius.



Five stars for playing comfort

The background of the game, i.e. the hotel scenario, works as follows: Lazy Jones saunters - directed by the player - in his characteristic 'easy walk' through the hotel and has to evade the bell boy, an ominous hotel guest in black and the chamber maid's service cart. The aim of the game is to explore all rooms behind the doors. If Lazy Jones is hit by one of his three adversaries he loses one of his five lives. Behind the doors various games can be found, ranging from well-known games like 'Space Invaders' to small 'shoot'em-up' games called 'The Hills are Alive'. Moreover there's David Whittaker's own creation '99 Red Balloons' where the player has to catch a balloon and hang on to it in order to reach his lover's balcony, or the



The cover of the 'Kernkraft 400' CD by Zombie Nation

extremely weird 'The Turk' where players have to skewer a turkey with a fork while fighting off a nasty telephone.

Some of the hotel's doors hold other things like a broom-closet, which logically is boring to death, a bathroom (just as uninteresting), or a bedroom (where a nightmare lies in wait for Lazy Jones), and last but not least the bar. There, players have to attempt to order as many drinks as possible at the bar, which is made much harder than it sounds by a badly swaying drunk.

So how's the implementation?

When looking at so many droll ideas you naturally get to the question of whether or not the implementation of the game can keep up with them. As Whittaker programmed the game completely on his own, he had every possibility to implement the basic game idea exactly as he had in mind.

The first thing to catch your ears is of course the music. Naturally, in a game by David Whittaker, you'll find the composer put together all of his own sounds, which is easy to notice: Every game within Lazy Jones has its own music, all very well composed. The basic rhythm stays the same in all of them and the pieces merge into one another every time the scene changes. There are no objections to be made on the technical side, especially if you take into account the short time it took Whittaker to design the game. Overall, the music alone is worth buying the game. Whittaker transformed the main themes of various famous pieces into music for the Lazy Jones. Present of course is the background to "99 Red Balloons" (that is, Nena's "99 Luftballons"), and in another game you can hear "Fade to Grey" by Visage.

Of course the graphics of Lazy Jones won't exactly take your breath away, even though they look quite acceptable for the mid 1980s. On the other hand, the graphics won't make you abandon your C64 in despair and they fit in with the game, though naturally little more polishing would have done them good.

The game itself is technically thoroughly acceptable and, what's most important, it is fun to play. Because of the entertaining little games behind the doors, Lazy Jones is great for either a short play in between other things or a longer, more involved session. There aren't many negative points to be made, taking into account that whether you like the various little games or not depends on your personal preferences. You might find one or two of them flops, but that doesn't really matter much as there are so many of them. Neither wandering through the hotel nor the various games are difficult, so players have no reason for frustration. Searching the hotel for the ultimate game keeps you in a good mood all the time.

Lazy Jones is one of the best games for the C64, a true classic.

Title	Lazy Jones
Designer	David Whittaker
Year of publication	1984
Tested with	C64, 1541, Joystick
Genre	Arcade-Action
Where to get it from	the second-hand market

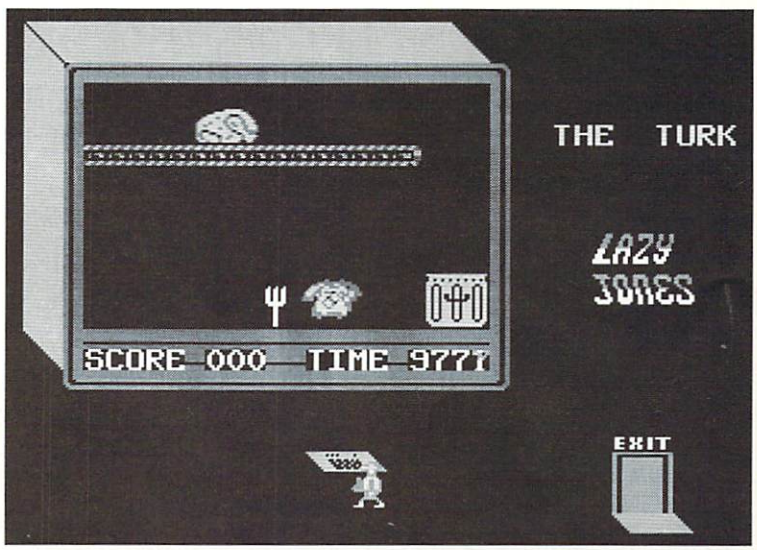
Minus ...

- graphics aren't exactly astounding
- some of the games might not be to everybody's liking

... and Plusses

- fantastic game concept
- great music
- many different games
- very entertaining
- perfect for either a short game on the side or for a longer play
- good fun

100 — 78% — 0



Get your Thanksgiving dinner before the nasty telephone gets you!

The Commodore Connection Line

by Oliver Biasin

A few years back, there were no problems; if you had any questions about your C64 or C128, you could simply ask your neighbor or the man in the street. Just about everyone had a "breadbox", and those who hadn't knew at least someone who had. Even those who couldn't afford one, knew about them. After all, you could use the computers on display in every good department store. This was often without permission, but who cared those days? Today, everything has changed. In the department store, there are usually only PCs, and you're just allowed to look at them. Programs for PCs are written in big firms. And when you are looking for someone knowing about the c64, you only receive a pitying smile. Those few magazines that stayed in touch with the c64 users ceased publication one by one, and the c64 users were totally deserted.

A terrific idea was needed. Some means of learning where to find c64 users. I had always asked myself, why no one has ever thought of collecting the addresses of the c64 users in order to allow them to get into contact with each other. Then the last one of the old German c64 magazines ceased to exist, and I finally realized that my idea wouldn't become reality unless I did it myself. Therefore I asked GO64! magazine, which had just started then, to write an article about my Commodore Connection Line (CCL). They did so right away, and I was amazed by the readers' feedback.

After only a short time, I had so many addresses of Commodore users, that postage just grew too expensive. There was need of an affordable way to provide all the addresses to all those in the list. Therefore I changed the address list's concept. From that point on, the addresses were not sent every three months to all, as had been the original routine, but rather only to those, who had given me the postage in advance. This made things a bit more complicated, but also helped me get control of postage costs.

In addition, I also thought of distributing the list by help of GO64! magazine. Since nearly all of the list's participants also had subscribed to GO64! magazine, this idea seemed to be nearly perfect. Soon thereafter, we had the first issue of my address list on the cover disk. The success was terrific. Within a short period of time, more people had entered the list than ever before. At the same time, Ralf Knabe (a.k.a. "Stressie") offered to convert the list for GEOS.

For about two years now, the new CCL concept has worked perfectly well. At computer parties and the Hobby & Elektronik fair at Stuttgart, I got more and more names on the list. In those two years since the original idea was put to work, the CCL's members grew to more than 100 people. Therefore, I thought that the time has come to publish the addresses on a GO64!'s cover disk again. At this time, Commodore World had just announced their fusion with GO64! magazine.

This made me consider to open the Connection Line for international "traffic" as well. I am planing to create an individual CCL list for every country seperately, so as to ensure clarity. Maybe I will start with having several countries in one list, depending on how others perceive the idea. Since postage into other countries is more expensive than within Germany, I can't send the CCL list to everyone individually. But you will see it regularly on the GO64!/CW's cover disk.

For those of you who would rather receive the list more frequently, I installed the CCL mail service. Simply send me an e-mail, and I will send you the CCL as a reply. You can transfer the CCL file via transfer cable, Blue Big Reader or other means to your c64 to examine it, or if you like, you can use a PC with a c64 emulator. Through the CCL, I hope to reach C64 users abroad, not just those in Germany, and establish communication among all readers. I further hope this will generate a better sense of community, and bring us all a little closer together.

In order to allow all to enter right away, the required questionnaire is included in this GO64! magazine's issue. These addresses are meant strictly for use between individual C64 users, of course, and are absolutely NOT to be used for any form of advertising. I encourage all users to copy and distribute this questionnaire as appropriate, as I would like the CCL to spread as far and as fast as possible.As far as the questionnaire is concerned, on the other hand, you are not In order to join, simply fill in the questionnaire and send it to the following address:

Oliver Biasin
Schwalbenweg 15
D-91220 Schnaittach, Germany.
Fax +49 (0) 9153-1831

Will I have the pleasure of welcoming new members to the Commodore Connection Line soon?

(ad/gb)

In the next issue...

Everything about printers

- **C64 compatible printers and how to connect them**
- **doing a printerinterface on your own**
- **GEOS and printers**
- **printing posters with GoDot**

plus:

ZipDisks under GEOS
SuperCPU - Corner
Tutorials and workshops
and...

New: CLiPS Corner - a regular report of the development status of the new multitasking SuperCPU OS CLiPS

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.