

# TWIN CITIES 128

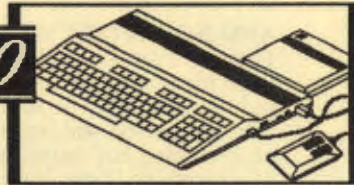
## THE COMMODORE 128 JOURNAL

ISSUE 24

\$2.50 U.S.

**GEOS 128 v2.0**

*The Graphics Environment  
Operating System From  
Berkeley Softworks*

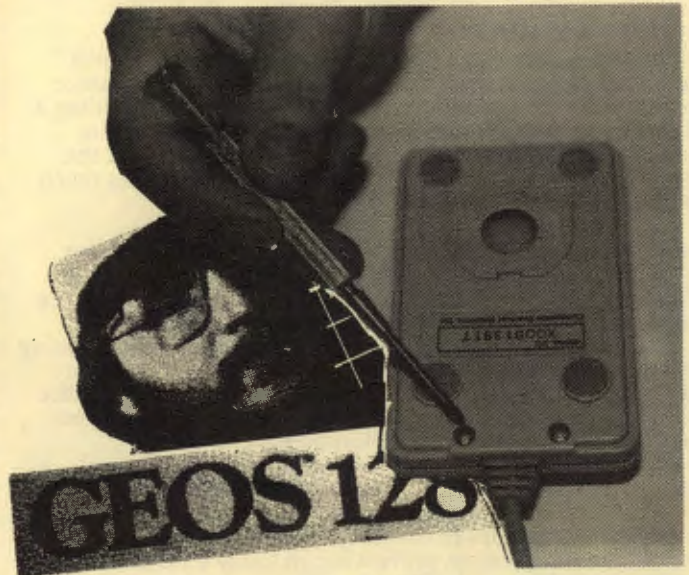


### IMPROVED GEOS ARRIVES...

It has been a long time since we have made anything more than a passing reference to GEOS in Twin Cities 128. However with the arrival of a significant upgrade to Berkeley Softwork's version of GEOS for the C-128, GEOS coverage returns to our pages in a big way. See Loren's candid examination of GEOS 128 v 2.0 on page 6.

### More than just a review...

We know without a doubt that over the next few months every Commodore publication in the universe that includes 128 coverage will have some sort of review of this upgrade, but our coverage does not stop there. After all, we know that you expect more from us! So included in this issue is an article on getting the most out of GEOS 128 v2.0 and a breakthrough program piece by Rick Kane that allows the transfer your **GEOS** images to **BASIC 8!**

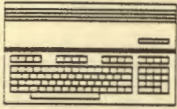


FRONT COVER	by Loren Lovhaug	Page 1
RUMOR/OPINION/MAYHEM	by Frank Hudson & Loren Lovhaug	Page 2
PRICE & PROGRESS REPORT	by Frank Hudson & Loren Lovhaug	Page 3
TAX AID 128 REVIEW	by Ray Bryan	Page 5
GEOS 128 v2.0 REVIEW	by Loren Lovhaug	Page 6
N.U.R.S.E. 128 REVIEW	by Bruce Jaeger	Page 9
JUGG'LER 128 REVIEW	by Bill Juliani	Page 12
GEOS TO BASIC 8 CONVERTER	by Rick Kane	Page 13
DUAL VIDEO FUN	by Miklos Garamszeghy	Page 15
REU FREE ACCESS	by Fred Bowen	Page 18
DISK DRIVE PROGRAMMING	by Mike Gilsdorf	Page 19
FKEY POWER	by Rick Kane	Page 23
SPARROW'S SLICK TIPS	by Loren Lovhaug	Page 25
DISKMON 128	by Anton Treuenfels	Page 26

Photography & Graphics by Loren Lovhaug, Photo Processing by Bruce Jaeger

*Advertisers Index:*

Briwall 21 & 22, Brown Boxes 10, Creative Micro Designs 20, Herne Data Systems 18, Micro Aided Designs 11  
Mountain Wizardry Software 12, Software Support International 4



## RUMOR/OPINION/MAYHEM

by Frank Hudson & Loren Lovhaug

Howdy, I'm Frank Hudson, the new employee around here. You might have seen my name around the last couple of issues, or seen my name on Qlink in the 128 support area. Twin Cities 128 is not new to me however, in fact I've read it since the very first issue. I even read TC128 before I had a 128, when I was a C-64 owner dreaming of the 128's added abilities. That glorious day I first hit the power switch on my own new 128, a stack TC128's was beside me, guiding me in my explorations. Yes, TC128 was one of the reasons I bought a 128 and make good use of it after purchase.

### IT'S NOT A MAGAZINE, IT'S A WAY OF LIFE:

Those early issues had a lot to offer the beginner, extolling the virtues of our favorite machine, explaining how to utilize the new software and hardware emerging for it, and holding it all together with a contagious sense of fun and excitement that helped me learn. Now that I'm here, getting to see the letters from subscribers, I find that many of you care as much about the 128 and TC128 as I do.

I'm no longer just another TC128 reader, or a 128 beginner either, but I never developed into much of 128 programmer. With easy to use tools like BASIC 7.0 and BASIC 8.0, this out come wasn't the 128's fault, it's just that applications such as Pocket Writer and GEOS series made working and creating with the 128 so attractive I rarely found the time to write my own code. As Twin Cities 128 developed, it became the place to find the latest in 128 programming tips and the magazine has played a big part in the exploration of the 128's hi-resolution graphics modes among other things. Since I don't program, do I resent this part of the magazine? No way. I want to have new 128 goodies, and TC128 getting the news out to programmers helps make that possible. And this non-programmer enjoys previewing, in today's TC128 articles, tomorrow's advances in 128 software!

If you are like me, a non-programmer, wondering if TC128 will continue to serve you, stay tuned! One of my chief jobs around here will be to maintain and strengthen our coverage of topics of interest to beginners and plain garden variety users. It's my hope you'll agree with me that this publication remains required reading for all serious users of the 128. We're going to continue reviewing all the 128 applications we can get our hands on: with honest, experienced reviews that will really help you decide which hardware and software fits your needs. And those power user tips on how to get the most from software you already have are going to continue too. When neat new stuff becomes available, TC128 is the place to read about it.

### IT'S NOT JUST A WAY OF LIFE, IT'S A JOB:

It's a tough, dirty job looking at all the new stuff being released for the 128, but hey, somebody's got to do it. Right now the staff at TC128 World Headquarters is enjoying the new V2.0 of GEOS 128 and exploring new ways to import graphics into the 128 environment. GEOS 128 2.0 is somewhat bittersweet experience for me after seeing what BSW did for the Apple // user last year. Maybe they felt sorry for those poor souls who paid so much for their Apples only to get an approximation of the 128's overall abilities--but the Alar coated brand got 80 column GEOS 2.0 first, and with a real manual, no DOS copy protection, and a couple of additional capabilities. Oh BSW--after all us Commodore users have done for you! You go and spend the kids milk money on that painted and fruited floozy! Still and all, GEOS is one of those disks that is always near the front of my file box. And our GEOS article is more than a review, it should help the GEOS user get more out of this package.

It's pretty exciting watching things happen around my new workplace. Graphics are constantly being swiped from more expensive systems and ported to the 128 with all their detail intact. There is the constant evaluation of new software, both commercial and public domain keeping all of the screens lit with excitement. Watching Rick Kane or Loren demonstrate their latest interlace mode discoveries is also a real thrill.

Logging on Q-link and GENie in an attempt to stay current, answering mail and processing orders, not just from the U.S. and Canada mind you, but from Australia, England, Ireland, and Norway, fill out the rest of my workday. With all the flurry of activity around here it is a wonder Loren and Avonelle managed to get a magazine out the door! And as I learn the ropes, I'm discovering that Twin Cities 128, is not just a job, it's an adventure.

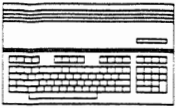
### AND NOW A FEW WORDS FROM THE BOSS...

Hi gang, I am Loren Lovhaug, and since January of 1986 I have been turning out North America's only C-128 specific publication. For the scores of you who are new to Twin Cities 128, I hope you will find every page of this issue enlightening and entertaining. Since our last issue, a number of interesting things have transpired, many of which are highlighted in our Price and Progress Report. During this period we have been talking with Commodore and a variety of developers about the long term C-128 users support plan I outlined in our last issue. I want to thank all of you who wrote to us and Commodore, in support of our proposal. I can tell you that there are indeed people reading your letters and that the plan is seriously being discussed and considered. I can also tell you that we have a lot of territory to cover if it is to become a reality. I also want you to keep in mind that it is a long shot, but regardless of what Commodore allows us to do for you, we continue to keep swinging for you!

Which brings me to a bit of a sad note. As the 8 bit computer market continues to mature and software Commodore software producers diversify into other markets you will no doubt begin to notice some changes within the Commodore media. The large glossy magazines that cover the 64 and the 128 which depend largely on advertising revenue for their profit will begin to shrink and in some cases disappear. Ahoy! was the first casualty of this trend. In a prime case of irony, their fifth anniversary issue extolling on their longevity was their swan song. Recent issues of RUN and Compute's Gazette have also hit the newstand quite a bit thinner with higher editorial content to advertising page ratios. In another case of irony, we have strong evidence that the 128 community is continuing to grow. Commodore expects to sell more than one million new 64s and 128Ds in 1989, and we are getting lots of information requests and new subscriptions from brand new 128 owners and our area on Q-link is always brimming with questions from new comers.

All of this, and the fact that we are starting to pick up some of the advertiser support that other publications are losing due to our relatively cheap advertising rates has continued to assure us that there is indeed a strong market for all levels of 128 support. In this vein, as Frank mentioned, we are going to continue to expand out beginning and intermediate 128 coverage, while still maintaining a high profile on the "leading edge" of C-128 programming developments. Along these lines we are also considering publishing a book entitled, "500 C-128 Questions...Answered". If you would like to see such a book, or, have some ideas or suggestions on what should be included in such a text we would love to hear from you.

One last note, last issue we changed our subscription mailing method from 1st to 3rd class within the US. This cut our mailing costs a whopping 80%! However there were a few minor difficulties. The post office told us that our subscribers should receive their issues within two weeks. While most of our readers who access Q-link told us they received their issues within two weeks, we did learn that for some readers it took as long as four weeks. We also learned that the post office does not forward third class mail when people have moved, so if you are moving in the near future, please be certain to send us your new address immediately! As always, if you suspect you have missed an issue please write us. Be sure to include your name and address information, plus a phone number where we can reach you.



## PRICE & PROGRESS REPORT

by Frank Hudson & Loren Lovhaug

**Looking for a 1750?...**For those tired of stalking the elusive RAM expansion unit, this spring brings news of two new sources for 512K REU's so vital for GEOS, C-128 productivity, and CP/M users. Several sources have reported availability of genuine 1750 RAM expansion units during March from Sears. Those who have found them at the stores say that Sears new lower price retail strategy extends to the hard-to-find 1750, with a \$180 price being reported by one Minneapolis shopper. Another report says that Sears still has a valid catalog number (3 A 32604 C) for this item and that as of this February they were available for \$199.99. I doubt Sears' stock of 1750's is very deep, so there are no guarantees you'll bag a REU by the time you read this. Sears, where America shops for expansion memory?

Software Support International has cleverly found another route to meeting the 128 owners REU needs. They've bought up a batch of the 256K 1764 units meant for 64 users and then installed additional RAM to bring them up to full 1750 status. Each cartridge is tested and warranted to be free from defects. Their price for this "1750 Clone" is \$199.95, and considering the amount of hand work that goes into such a conversion, it's a bargain. Have a 1764 hanging around that you'd like converted to 512K? SSI will do that for \$124.95. Wonder what they're doing with all those extra heavy-duty power-packs that Commodore ships with the 1764 (the 64's power supply, unlike the 128's, isn't up to the addition of a REU)? For \$29.95 your C64 using friends can get one of the excess power supplies from Software Support International. Software Support International 2700 NE Anderson Rd, Vancouver, WA 98661 (800) 356-1179.

**BASIC 8 available again...** By the time you read this Free Spirit Software Inc. will have entered into an agreement with Walrus Software Inc. to publish BASIC 8, the popular graphics programming system for the Commodore 128 computer written by Lou Wallace and David Darus.

BASIC 8 is a programming environment for the Commodore 128 (or 128D) computer in 128 mode with 80 column graphics display. BASIC 8 adds over 50 new graphic commands to standard C128 BASIC. BASIC 8 allows 640x200 resolution in monochrome and 640x192 resolution in 16 colors without any additional hardware. BASIC 8 even permits drawing in a three dimensional environment. Commands are included for windows, fonts, patterns and brushes. A RunTime Library, which allows programs written with BASIC 8 to run without BASIC 8, is included.

The version of BASIC 8 to be published by Free Spirit has been revised and updated. The package includes a two disk set containing the BASIC 8 language, RunTime library, demonstration programs such as BASIC Paint, BASIC Calc, BASIC Write, utilities and a new 200 page instruction manual. The list price for BASIC 8 is \$39.95.

**If I had a Hammer...**Free Spirit Software Inc. has also released a package called "Basic 8 Toolkit", a utility program for use with the Basic 8 hi-resolution graphics development system for the Commodore 128 computer. The Basic 8 Toolkit features a point-and-click operating system that allows the user to create custom Pointers, Fonts, Patterns and Icons. The Toolkit also allows the user to convert Print Shop Graphics into Basic 8 graphic files. Colors may be added to the Patterns, Icons and Print Shop Graphics with the color option. The Toolkit also contains a set of useful disk utilities such as make autoboot, convert icon file to brush file, scratch file, rename file, validate disk, toggle drive, directory and format. Files created with the Basic 8 Toolkit can be used in custom programs, Basic 8 programs or with graphic programs such as Sketchpad 128, Spectrum 128 or News Maker 128. The Basic 8 Toolkit is available from Free Spirit for \$19.95

**Does Whiz Kid Ever Sleep?...**Poster Maker 128 allows the user to create large posters ranging in size from two foot by two foot to five by five feet (25 pages!). The posters may be printed out or saved to disk as "picture files". Poster Maker 128 includes a graphic utility which permits it to import BASIC 8, Sketchpad 128 or Spectrum 128 graphic files. These graphic files can also be enlarged on both their X and Y axis. Also included is a utility to reduce the size of graphic screens in order to create clip art. Poster Maker 128 operates on the Commodore 128D or the Commodore 128 with upgraded 64K Video RAM in 80 column 128 mode. Resolution is 640x200 pixels. A 1351 or compatible mouse and a 1571 disk drive are required. Optional support is provided for a 1571 or 1581 disk drive a second drive. Poster Maker 128 has a suggested retail price of \$29.95. Free Spirit Software Inc. P.O. Box 128, 58 Noble St., Kutztown, PA 19530, (800) 552-6777.

**Missing in Action...**As of date we still have not seen a finished copy of the now legendary 100% assembly language paint package from Walrusoft or Pocket Writer 3. Our spies tell us the Pocket will not see daylight until May, while Walrusoft Paint is near completion, but in a holding pattern as a result of the frustrating legal tangles between Walrusoft and Patech, the former distributors of BASIC 8. Meanwhile we are sending out search parties.

**On the Homefront...**As the popularity of Twin Cities 128 continues to escalate, we find ourselves in the uncomfortable position of not being able to fill requests for some of our back issues. Unfortunately (or as our accountant says, fortunately) we have sold out of all of our issues except a limited supply of issues #4, #19, #20, and #23. To fill the demand for many of our earlier issues we have created the Twin Cities 128 Compendium Book #1 which contains the best material from our first 16 issues. This 175 page collection sells for \$16.95. We most definitely will be creating a second compendium book in the future, however, for those who simply can not wait we can send you photocopies of any article we have published. Just send \$3.00 for the first article and 50 cents for each additional article to: **Twin Cities 128, P.O. Box 11578, Minneapolis MN 55411.** Also, again as a service to our readers, we are offering all the programs from this issue on a disk for just \$5.00 which includes postage and handling.

**Overheard...**According to Les Lawrance at Software Support International, they are very close to creating a utility that will allow GEOS 128 v2.0 to boot off of a 3.5 inch disk while fully protecting Berkeley Softworks from those who are silly enough to endanger C-128 development through software theft. Meanwhile, for the last couple of weeks we have had the luxury of playing a full featured **terminal package** which runs under GEOS 128 and GEOS 128 v2.0! Geoterminal 128 includes a 30K buffer, Xmodem, XmodemCRC, and punter file transfers, 300 and 1200 baud support, a built-in GEOS to PRG or SEQ and vice versa file converter, and a fully configurable autodial phone book! This work of art was written by GEOS programmer first-class **Bill Coleman** aka Master Blaster and will be sold on the upcoming second GEOS productivity package from RUN magazine. On other front, our informants tell us that sometime later this spring **Free Spirit** will be marketing a whole series of high quality clip art disks for the 128 in BASIC 8 format, and something truly original: a **clip sound** collection and a basic wedge that will allow you to incorporate life-like digitized sounds in your own C-128 programs! Lastly, we keep hearing that a famous florida C-128 master programmer is very close to completing a pre-compiler system that will allow **BASIC 8** programs to be **compiled** with Abacus software's BASIC compiler. If this pans out...look for a flood of super fast 128 graphics software!

## 1541 RAMBOard\*

### Byte The Copy Protection bullet!

Let's face it - war is hell. And in the Commodore world, one of the ugliest wars of all is being waged by the copy protection industry. They're tough. They're clever. And they're very, very determined. If you've seen some of the current ice out there, you know just how determined they are. Well, we're just as determined. And we've got a way to walk right through the coldest ice out there.

What's our secret? The new RAMBOard, a small card that can easily be installed in your 1541 in 5 minutes using just a screwdriver. With this remarkable piece of firmware in place, backup hassles are a thing of the past!

RAMBOard gives you speed - it comes bundled with a fast copier that can archive an unprotected data disk in under 60 seconds. RAMBOard gives you power - working with our dynamic Maverick software, RAMBOard will back up programs that other systems can't even scratch the surface of. Best of all, RAMBOard won't become obsolete - when newer, even tougher protection schemes come along, we'll create new Maverick/RAMBOard parameters to cut right through them. Backed by our famous R&D, you know that we'll do whatever it takes to keep RAMBOard operating behind enemy lines!

Get your RAMBOard today. And the next time some company tries to charge you for an "authorized" backup of a program you've already paid for, remind them that war IS hell - and you know how to raise a little hell of your own.

**1541 & 1541C RAMBOard/ONLY \$34.95**

*In Most Cases, Plugs Right In-NO Soldering Required*

**1541II RAMBOard/ONLY \$44.95**

*This Version DOES Require Soldering*

*Professional Installation Available-Contact Us For Details*

**\*RAMBOard Is An Optional Accessory For Use With Maverick!\***

NEW V2.0

## MAVERICK™\*

### A New Tool-From Some Old Friends

You know us. When copy protection began throwing 1541 drive heads out of alignment, we gave you Kracker Jax. When V-Max™ reared its ugly head, we created the Kracker Jax Elite series. When the copy protection wars escalated, we responded with Bull's-Eye and the Hacker's Utility Kit. And when you decided to explore the mysteries of commercial-grade protection schemes, we were there with Kracker Jax Revealed.

Now the copy protection industry has upped the ante again. They've made it harder than ever to make legitimate backups of your expensive software. Do you think you should have to spend money to buy "authorized" backups of software that you already own?

Neither do we.

Introducing Maverick, the best weapon we've ever made to let you defend yourself from copy protection profiteering. Maverick is so advanced, it has actually replaced every other backup utility we've ever created. That's right - Maverick is now the ONLY archival software we manufacture. All of our R&D efforts go into making sure that Maverick remains the most powerful backup utility available - from ANY source! Take a look at some of these features:

- NEW - 1541 1571 1581 fast file copier - any direction
- GCR Editor for the experienced hacker
- NEW - GEOS™ file copier - works under desktop
- Error scanner with unique sector editor
- Directory editor helps you organize your disks
- Works with C-64 C-128 & 1541 1571 1581 drives
- Scrolling M L monitor with DriveMon
- EXCLUSIVE Popular subscriber service available
- Single or dual state of the art nibbler
- 300 - parameters for those tough to backup disks
- Capable of archiving Rapidlok protection
- Single of dual drive high speed data copier
- EXCLUSIVE GEOS™ Module PARAMETERS & TOOLS
- NEW - Dump a formatted parameter catalog to printer
- Byte pattern scanner for high speed searches
- Technical support from our staff experts

And this is just the beginning. Our future plans for Maverick include expanded RAMBOard support, RAM expansion module support, and even a 1541 1571 1581 relative file copier! And with our unique Subscriber Program, you can receive Maverick updates automatically - the instant they're released!

You know us. We're the company that has worked hard for years to establish an ironclad reputation for dynamic products, fair prices, and superb customer support. We've always been a part of your past. And as long as you have your Commodore, we'll always be a part of your future.

Attention: Registered Maverick owners!

Parameter Module #3 is now available, Only \$9.95

Maverick V2.0 Upgrade also available, Only \$9.95

**MAVERICK 2.0/ONLY \$34.95**

*Now Includes Parameter Modules 1, 2, & 3!*

**\*Maverick Was Formerly Known As "Renegade" From Kracker Jax Not To Be Confused With The Game Of The Same Name**

## THE 1581 TOOLKIT

### Without It, You Don't Know What Your 1581 Is Capable Of

It blew people away at the "World Of Commodore Show" held in Pennsylvania in November 1988. It's power and scope astonish anyone who sees it work. And, if you have a 1581 drive, it is the single most important piece of software that you can own. It is The 1581 Toolkit - and it will transform your 1581 into something remarkable.

We know the market, and we know that if you've bought any other 1581 utility in the past, you've probably been disappointed. But while other companies rushed ill-conceived and poorly executed products to market to "cash in" on the new 1581 drive, we did things differently. Our programmers took the time to become experts on the 1581 before they even looked at a drawing board. So when they finished the Toolkit project a full year later, they had accomplished what no one else had - the creation of a tight, integrated package of essential utilities that make the 1581 more than the sum of its parts. Take a look at these Toolkit utilities and see why we're so excited about it:

- Fast Disk Copier
- Directory Editor
- File Track & Sector Tracer
- Fast File Copier
- Byte Pattern Search
- Ultrafast Formatter
- Partition Creator
- Track & Sector Editor
- Error Scanner
- Relocatable Fast Loader

All 10 of these custom utilities use extremely fast read/write routines. Wherever appropriate, they allow full access to partitions. It's obvious we created The 1581 Toolkit to be the finest set of utilities available for the 1581 drive. But we didn't stop there. Order now, and we'll also send you the 1581 DOS Reference Guide FREE! With over 100 pages of concise, valuable information, you'll quickly find this book indispensable! Written by David Martin (INFO Magazine's Copy Corner reviewer), this manual represents a year's worth of research and investigation into the inner workings of the 1581 drive. We've also added extra utilities, including such handy items as a machine language monitor with DriveMon, the perfect companion for the 1581 DOS Reference Guide.

We've spent a lot of time working hard to make sure that nobody knows more about the 1581 than we do. Don't sell your 1581 short. Get the 1581 Toolkit today, and let us show you just what it's really capable of. We'll take your 1581 absolutely as far as it was designed to go. And then we'll take it farther.

**THE 1581 TOOLKIT (ON 3.5" DISK)/ONLY \$39.95**

*Includes Your FREE 1581 DOS Reference Guide*

*The 1581 Toolkit Works On The C-64 Or The C-128/C-128D In The 64 mode*

ALL NEW

## THE 1750 CLONE

### Thanks For The Memory

Most Commodore users are content to use their computers within the constraints placed on them by a limited amount of RAM. But some people want more. More speed. More power. And that means just one thing - more memory.

Power users have long known that the ideal solution to their memory problems is out there in the form of Commodore's incredible 1750 RAM expansion module. Just plug it in and you've got a whopping 512k of onboard RAM - the same amount of memory found on most IBM computers! This is the memory cartridge that all serious Commodore power users want. There's only one problem.

You can't find them.

That's right. These RAM units are almost impossible to buy. We've always specialized in finding solutions to Commodore related problems - here's what we came up with.

First, we bought brand new Commodore 1764 RAM expansion units, the ones with only 256k of RAM onboard. Next, we had Chip Level Design engineer and produce a custom upgrade that takes a 1764 to a full 512k! That's the same half meg of RAM as a regular 1750! Finally, we tested each and every cartridge, and warranted them to be free from defects.

Now there's nothing to stop you from taking your Commodore 64 or 128 to levels of power and sophistication that the original designers never even dreamed of! If you're using programs like GEOS from Berkeley; the Pocket Series from Digital Solutions; future versions of Maverick from Kracker Jax; Fleet System IV from Professional Software; or the potent PaperClip III from Batteries Included; if you're using ANY of these programs, you won't believe the difference that the extra memory makes!

#### IMPORTANT NOTES - READ CAREFULLY

\*C-64 64c (but NOT C-128 128D) owners MUST buy a heavy-duty power supply to use these units. The power supply is NOT included - it is available from us separately.

\*If you ALREADY own a 1764 RAM cartridge, we can upgrade it for you. Your unit MUST be in perfect working order for us to upgrade it. The turnaround time on upgrades is approx. 2 weeks. It will NOT be necessary for you to purchase a new power supply - the one that came with your 1764 will still work fine.

**THE 1750 CLONE  
ONLY \$199.95  
1764 UPGRADE  
ONLY \$124.95**

## ATTENTION C-128 OWNERS

### Solderless 64K Video RAM Upgrade

Now that Commodore has released the C-128D with 64K of video RAM, we should be seeing 128 programs address this fantastic new feature soon.

**BASIC 8** already has the capability of using all 64K of video RAM. If you own the C-128 in stock condition, you own all 64K of video RAM that Commodore felt was necessary. Using Basic 8 format and the full 64K of video RAM provides you with the ability to scroll through video memory as well as enhanced color resolution.

Up until now, to upgrade the C-128 to 64K of video RAM you would have to first search out the components, then find a competent repair outlet to desolder and install the parts. **What a hassle!**

**SOLUTION** - We have developed a module that simply plugs in to the mother board of your C-128. **No splattered solder - No heat damage - No hassle.**

This package is available in two different forms. Model A has all parts necessary to complete your upgrade, and Model B is identical except that you supply the RAMs. Again, no soldering is necessary. If you have access to low cost RAMs, we suggest Model B.

Both models include easy to follow installation instructions, a test program to verify proper installation, and the plug in Video RAM Upgrade.

**Model A (complete) only \$49.95**  
**Model B (no RAMs) only \$29.95**

## COMMODORE HEAVY-DUTY POWER SUPPLIES

*WATTS The Matter - Can't Get ample power?*

Sorry about that. Actually, it's no laughing matter. Our technicians tell us that the majority of Commodore computer problems can be traced back to an inadequate power supply. When the power supply fails, it can (and often does) cause additional damage to the computer itself - and a computer meltdown is no joke.

This doesn't have to happen to you. We are now selling a heavy-duty power supply that will greatly diminish your fear of a power supply failure. These are NOT aftermarket units - they are made by Commodore, and are ideal for use with RAM expanders. They will provide you with a more stable, dependable source of power. This unit will work on ALL C-64's and 64c on the market.

If your system is important to you, take the extra step to safeguard your investment by getting a Commodore Heavy-Duty Power Supply today. Power you can depend on - with no unexpected shocks.

**COMMODORE HEAVY-DUTY POWER SUPPLY  
ONLY \$29.95**

**SOFTWARE  
Support  
INTERNATIONAL**

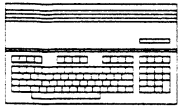
**PLEASE READ BEFORE ORDERING:** We accept money orders, certified checks, VISA, M.C. and Discover. Previous Software Support customers may use C.O.D. and personal checks. Orders shipped to U.S.A. (48 states), P.O., A.P.O. or possessions, please add \$3.50 per order for S & H. U.S. shipping is by UPS ground in most cases. FAST 2nd DAY AIR available, add \$1.00 per pound additional (U.S. 48 states only). Alaska or Hawaii (all orders shipped 2nd day air), please add \$7.50 per order for S & H. C.O.D. available to U.S. customers only (50 states); add \$2.75 along with your S & H charges per order. Canadian customers may calculate the S & H charges by including \$4.00 (minimum charge) for the first two pieces of SOFTWARE and \$1.00 for each additional piece per shipment. All monies must be submitted in U.S. funds. Canadians must call or write for hardware shipping charges. Foreign customers must call or write for shipping charges. Defective items are replaced at no charge if sent postpaid. All in stock orders are processed within 24 hours. U.S. SOFTWARE orders over \$100 will be shipped 2nd Day Air at our regular \$3.50 S & H charge (48 states only). Washington residents please add 7.6% additional for Sales Tax. All prices subject to change. All sales are final unless authorized by management.

Mail your order to: Software Support, Inc.  
2700 NE Andresen Road Vancouver, WA 98661

**Or call our toll-free order line at  
1-800-356-1179, 9am-5pm Pacific time  
Monday-Friday  
After hours orders accepted at  
(206) 695-9648 7 days a week.  
Technical support available. Call  
(206) 695-9648, 9am-5pm Pacific time,  
Monday-Friday.  
Orders Outside USA call (206) 695-1393.**



**DEALERS — WE HAVE THE SUPPORT YOU'RE LOOKING FOR!**



## TAX AIDE REVIEW

by Ray Bryan

The past two years I have complained so much about doing the taxes in slow and stupid mode on my C-128 using Timeworks' Swiftax that this year I swore I would find another, better way and I did. (I concede that Timeworks is second to none in their customer service - they even offered my money back after I had filed - but I felt I must keep the program just in case of audit; this year they have a C-128 version apparently). I had seen those little ads for Taxaid in the back pages of those 'other Commodore magazines' and thought if it is for the 64, Plus/4 and 128, how could it be better? (Actually, there are 3 C-128 mode packages to choose from the 3rd is Steve Karasek, 855 Diversey, St. Louis, MO 63126 314-961-2052, which does schedule F for farm income and 2106 Employee Business Expenses which Taxaid will not do).

Taxaid Software, Inc. doesn't spend their time and money on glossy ads, super salespeople and customer service support; they spent their time researching tax law and writing intelligent code. Unlike the 'name brand' product mentioned above this program supports the 1571 and 1581 disk drives and multiple drive units (it is distributed on a 5.25 inch floppy but was easily backed up to a 3.5 inch disk). Although it does not directly support the 1700 or 1750 Ram Expansion Unit and the manual's recommendation that with two drives the data disk be in the one configured as device 8 while the program disk is in 9, I found that it is usable and resides quite well in device 10 under RAMDOS. The manual has a caveat regarding non-Commodore printers with not 100% compatible interfaces, but it worked very well with my Silver Reed Typewriter and Micrographix MW350 interface in emulate mode. In short, it allows you to use the system you have, not contort your set-up to meet some lazy programmer's idea of the lowest common denominator.

I began by running RAMDOS from the 1581 disc drive (configured as device 9 in my set-up). I then used FILECOPY to copy the program into the Ram Expansion Unit. The manual states that a blank disc should be formatted before starting the process. That is because Taxaid provides no means to send disk commands to the drive other than load and save during its operation - a program called FORMAT is on the disk, however and can be used in between modules. I also tried Taxaid with the Ram Expansion Unit as device 8 and used FILECOPY between running the modules to save the data to the real drive. Using RUN"filename"onU(drivenumber) sure beats the pants off of LOAD "filename", (drivenumber) and RUN, for my money. (On the 1581 you can use RUN"f\*e"onU(drivenumber).

The program is divided into 5 separate modules which are each meant to run alone. If you have a small business, income property or capital gains income (or a combination of these as in my wife's and my case) you need to start with the separate schedules C, D or E (and SE if you have no W2 income or need to fill out the earned income credit). These results must be saved and printed out to file with the main 1040 form (you may file the printed out copy or transfer the data by hand to the government forms). The main bulk of the work is done in the 1040 module which includes schedules A and B and form 2441, the childcare credit form.

Each module starts out with a very readable, no frills menu from which to choose between data entry, data display, data changes, results display, loading or saving data, out-print of results or a work sheet and a quit/clear option. Choosing the quit/clear option brings up the only other choice menu with options to erase all data and start over, erase all data and quit or return to main menu (without disturbing the data).

The program uses what the manual calls 'entry numbers' as prompts that appear on the screen as (E#). The user is expected to enter a number and carriage-return. The program then asks for a specific datum. These prompt lines are all spelled out in the book, with a clear explanation. Any item

which does not pertain to your situation, you may skip over by simply not typing in the E# of that question. The numbers do not follow the IRS forms line numbers but with the Taxaid manual on one side and the form on the other it is easy to see what data is called for and where the process is leading. The book is concise and to the point. The manual suggests you print out the program work sheets to aid organizing your figures before you begin entering data.

Taxaid is very easy to use and fast (especially from the Ram Expansion Unit) in its calculations. With taxaid it is very fast and easy to decide what options will minimize your tax liability, for instance, you may want to know whether to itemize your deductions or take the Standard Deduction or how a contribution to your IRA will affect the tax you owe. Taxaid can do this type of speculation much quicker than the old calculator and paper method.

Weaknesses? Yes, it has some. No Schedule F or Form 4562 depreciation schedule are in the program. It would be nice to specify a data drive other than 8, and changes on the separate module schedules are not automatically plugged in to the respective 1040 lines. I was not put off by the entry number format but I suspect that some people would count it as a drawback.

Minimizing your taxes is what is important and Taxaid can help with that. Taxaid is not a program to meet the needs of every taxpayer. But if you have any sort of complexity to your tax status, a child in daycare, a house and a small business, a duplex, an even modest investment portfolio, it could be a great time saver for you. I am pleased with it and I recommend Taxaid. It costs \$49.95 with next year's update at \$18.00 plus \$2.00 shipping and handling.

Taxaid Software Inc., P.O. Box 100, Two Harbors, MN 55616.

C128 or C128D  
w/1571 or 1581

# FORMAT EXECUTIVE™

## Disk Format Conversion Utility

\$59.95 US

Version 4.0

File Transfers to 150+ 3.5" and 5.25" Disk Formats:

- MSDOS (IBM PS2, PC & PC Compatibles)
- CP/M-80
- CP/M-86
- Commodore CP/M
- Commodore 64 & 128
- Format Executive can also be used to transfer files to/from Commodore Amiga, Atari ST and Apple Macintosh.

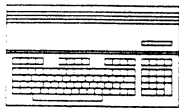
SUPPORTS:

- All Commodore Drive Devices, 154k, 157k, 158k Disk Drives & 17xx RAMDisks
- Single - Multiple - Dual Drives
- Read, Write & Format Disks
- Transfers Files (ASCII & OBJECT) Of Any Length
- ASCII - PETASCII Conversion
- Linefeed Adjustment
- Hi-speed "Burst" File Transfers
- CP/M User Areas
- 1581 CBM Disk Partitions
- COPY, RENAME, ERASE, DIRECTORY, LOGIN, HELP, RESET, SYSTEM
- Wildcards (\*,?)
- Disks Automatically Logged-In for Extra Security
- Backup Disks Permitted

On QuantumLink:  
POWERSOFT



INCORPORATED • P.O. BOX 7333 • BRADENTON, FLORIDA 34210



## GEOS 128 v2.0 REVIEW

by Loren Lovhaug

It seems like and eternity, way back in early May of 1986, I discovered something that radically changed the way I produced Twin Cities 128. At that time, Berkeley Softworks was an unknown entity, and GEOS 64 version 1.0 was being advertised with very small and modest black and white ads. Oh how times have changed. Looking back at issue #6 of Twin Cities 128, and the version of GEOS I was using, it is hard to believe I could have gotten the results I did. Version 1.0 of GEOS for the 64 was slow, clumsy, and bug ridden. There was no RAM Expansion support, in fact at the time the RAM expansions for the 128, let alone for the 64, were just a spot of vaporware on the horizon. The only disk drive supported was the 1541 which meant that I could only use single sided disks on my 1571, leaving me with a whopping 24K of disk space after I stripped my desktop of all unnecessary frills. Add to this the fact that the only input device support was a joystick ala the recently released "1350 roller joystick" and it seems like I was computing with knives and bear skins. And yet it was light years ahead of Doodle, which was the graphics package I used to render the illustrations in our previous issues.

In those days, GEOS was the new frontier. In fact, due to the fact that I was fortunate enough to get my hands on the pre-release version 1.0 of GEOS for the 64 in May of 1986, I can safely say that we pioneered the concept of desktop publishing with GEOS, as we used it to produce an issue of Twin Cities 128 weeks before it was readily commercially available. I am happy to say that both Twin Cities 128 and GEOS have come a long way since those days. But we remained on the cutting edge of GEOS developments for a long time to come. Later that summer, Commodore bestowed upon me a hard to come by 1750 RAM expansion unit (maybe things do not change) and a prototype 1351 "proportional" mouse so we could test them with a prototype version of GEOS that supported these new fangled gadgets. As time marched on we participated in the development of GEOS by beta-testing a variety GEOS add ons, including extensive testing of the early versions of GEOS 128. Then, sadly, during late fall of 1987, GEOS and I sort of broke company. It happened primarily as a result of our purchasing our Okidata Laserline 6 laserprinter and my fascination with BASIC 8 as an eighty column graphics tool. Because I was never able to get the HP compatible laser printer drivers for GEOS to work, I was forced to abandon it as a tool for the production of Twin Cities 128, even though I still was very fond of it as a graphics and desktop publishing tool. I also think that Berkeley's advertising, resplendent with immense gloss and hype, and some of their pricing, upgrade, and customer support policies (all of which show signs of significant improvement) also played a role in my disenchantment.

During the next year, I booted GEOS very rarely, quite a change from the days when it was a constant fixture on at least one of my 128s. Then in the late summer of 1988, I was asked to review GEOS 2.0 for the 64 for RUN magazine. Through my explorations I could see that Berkeley had come along way as far as GEOS development was concerned. Sure there were things I did nitpick about, as I will later in this article but, overall, GEOS had finally evolved into an operating system worthy of all the hype surrounding it. In my mind, only two pieces of the GEOS puzzle were now missing. First and foremost, the porting of this greatly enhanced version of GEOS to the 128, and secondly the support for my HP compatible laser. Two weeks ago those puzzle pieces arrived. And ever since, GEOS has again become a fixture on one my 128s. Forgive me for this poetic metaphor, but it does indeed feel like a long lost friend has returned. I assure you this friend is not perfect, mind you, however, I have never felt better about recommending GEOS to 128 owners than I do now.

GEOS 128 v2.0 comes on three double-sided 5.25 inch floppy disks. Unfortunately the boot disk for GEOS 128 v2.0 is copy protected, unlike the latest release of GEOS for the

Apple IIe which is not, even though the packages are nearly identical in scope and content. Apparently Berkeley has decided that it can "trust" Apple IIe owners but not Commodore 128 owners when it comes to software piracy. This decision is particularly ironic when one considers the fact that the Apple II community is notorious for heavy software piracy, particularly in elementary and secondary schools where Apple computers are so predominant. My own observation of the Commodore 128 community is that as a whole, Commodore 128 owners are far more likely to purchase a commercial productivity title like GEOS 128 v2.0 than either the Commodore 64 owners or Apple II owners. To Berkeley's credit, they have included a backup copy of the boot disk, just in case your primary boot disk fails. Berkeley also will replace failed boot disks for a nominal charge to registered GEOS 128 v2.0 owners.

Also included with the package is not one, but two manuals. The first is a perfect bound 307 page copy of the GEOS 64 v2.0 manual. As software manuals go this one is actually very good, and a nice improvement over previous GEOS documentation. The manual sports a table of contents, an index, a generous number of illustrations, and is organized in a logic progression from initial use to step by step instructions detailing the usage of the GEOS application programs. The second manual is a 106 page stapled "GEOS 128 v2.0" Addendum. While this manual shares the nice touches of the primary manual it is unfortunate that Berkeley did not make an attempt to integrate its material into the primary manual as their organization is nearly identical. As it stands, I find myself bouncing between the two manuals as both contain valuable information. Another nitpick I have about the manuals is that the perfect binding of the primary manual and the stapled binding of the addendum do show a lot of wear after moderate usage. It is a shame Berkeley did not see fit to bestow upon us the same loose-leaf, three-ring binder and slipcase, resplendent with tab index dividers that accompanies Apple GEOS. Again, it appears us Commodore types get second class treatment, even though recent price cuts in Apple GEOS put its pricetag in the same ballpark as GEOS 128 v2.0.

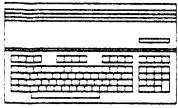
The enhancements to GEOS 128 can be divided into the following categories: desktop enhancements, geopaint enhancements, and geowrite enhancements.

### Desktop Enhancements

In my opinion, the enhancements to the GEOS 128 desktop are the most significant of the version 2.0 enhancements. As in the original GEOS 128, Berkeley fully supports the 1541, 1571, and 1581 disk drives as well as the 1764 and 1750 RAM Expansion units. However, the addition of support for an additional mass storage device, bringing the total to three is quite significant. This means that owners of two disk drives and a RAM Expansion unit do not have to sacrifice the use of one of these drives or go through the tedious process of constantly reconfiguring GEOS in order to utilize all of their equipment. Although the addition of support for a third mass storage unit is a big improvement, the way in which this support has been added is a tad clumsy. You see, as with GEOS 64 v2.0, you effectively only have access to two devices at any give time, labelled units A and B, while the third device unit C, lies dormant. In order to access the device C, you need to swap one of the two active devices, A or B, with unit C, thereby making device C active and one of the devices A or B inactive. This "shell game" can be accomplished by either dragging the icon for device C up into either the A or B device slot or by holding down the Commodore key and shift A or shift B to swap the respective device on the "on deck circle".

As was hinted at in the previous sentence, Berkeley has added a number of keyboard shortcuts for operations.

*Continued on Page 7*



## GEOS 128 v2.0 REVIEW

### Continued From Page Six

This keyboard supports allows those people who don't fancy themselves as "mouse jockeys" to navigate GEOS in most instances without lifting their fingers from the keyboard. To Berkeley's credit, many of these shortcuts are mnemonic in nature. In another very nice piece of Berkeley insight is the placement of a date and time read out on the desktop and a provision for setting the date and time from the desktop. Under previous versions of GEOS the clock could only be set from the preference manager. The end result is that very few GEOS owners took advantage of the time and date stamping features of GEOS because of the tedium associated with setting the clock and the fact that the preference manager was often considered "expendable" when it came to freeing disk space for other purposes.

But perhaps the most important improvements incorporated into GEOS 128 v2.0's desktop are its new file selection facilities. To begin, files may now be selected not only from the "icon" view mode, but in fact from all of the text type view modes as well. This means that if you want to, for instance, chose a number of font files to be copied to your RAM disk, you can quickly do so by instructing the desktop to display the files on your disk by "type" and all of your font files will automatically be grouped together making the selection and copy process much easier, in addition the file sizes are displayed as well which really helps when you are trying to cram those final couple of files into tight disk space. But the file selection enhancements to GEOS 128 v2.0 don't end there, GEOS 128 v2.0 now allows you to select multiple files for all file operations such as file copying, renaming, deletion etc. Multiple files can be selected in a number of ways: by holding down the commodore key and clicking on the filenames or icons of your choice, or by holding down the commodore key and pressing any of the numbers 1 through 8 which correspond to the filenames or icons currently being displayed on the desktop, or by utilizing the menu items (or their keyboard shortcuts) for selecting all files on the disk, desktop page, or those files on the desktop border. Clearly these new options make it much easier to conduct routine file maintenance on your GEOS disks.

### Geopaint Enhancements

In my opinion, the best reason for using GEOS has always been geopaint. Geopaint has always been one of the strongest drawing packages available for the 128 because it features a high level of sophistication combined with assembly language speed. Additionally, Geopaint is one of the few graphics programs for the Commodore 64 or 128 which is actually page oriented as opposed to screen oriented (see my discussion of these considerations in issue #16 of Twin Cities 128). As with the original C-128 version of Geopaint the GEOS 128 v2.0 version of Geopaint functions on either the forty or 80 column display, however color is only supported on the forty column display. This is indeed unfortunate. As readers of Twin Cities 128 know, over the past year some incredible strides have been in the realm of 80 column graphics on the 128 and it is hardly a "big trick" anymore to get color bitmaps on the 80 column display, even on "flat" 128s equipped only with 16K of video RAM. Taking this point one step further, just imagine for a moment how nice GEOS 128 v2.0 would have been if it supported the extended video RAM of the C-128D and upgraded flat 128s. C'mon Berkeley...we have published all the technical data for you! We know your programmers are bright...let's see more sophisticated 80 column support in GEOS 128 v3.0.

On the plus side however, Geopaint has been greatly improved. The litany of improvements includes: support of the cursor keys for pixel by pixel motion (a fantastic boon when working in pixel edit or zoom mode), a special "constrain" mode for most drawing tools which inhibits stray motion when trying to draw straight or diagonal lines, limited grid line support, much faster and more flexible system for region window movement and re-sizing, additional

pattern draw modes, and virtual image resizing and rescaling. While all of these features are welcome, the most impressive by far are the last three.

The new region movement and resizing really make the cutting, copying, and replacement of image data and "photo scraps" much easier. Under this new system whenever you are working with the region selection tool, (the outline box with the dotted lines) you are allowed to click on one of two control gadgets located on the upper left and lower right corners of the outline box. The gadget in the upper left corner allows you to move, copy, or relocate your selection area, while the gadget in the lower right corner allows you to re-size the area. Through the addition of these two relocation and resizing gadgets Berkeley has eliminated a lot of the tedium Geopainter's that used to be associated with cutting or placing graphic data precisely. Gone are the days when it was a matter of constantly redrawing the selection box until you got it just right. By using these gadgets, subtle adjustments are much easier.

The new pattern draw modes make it a lot easier to do texture, shading, and lighting effects. These modes are active when you use the paint brush or the air brush tools and allow you to render using patterns normally, or affecting only the foreground (black), or the background (white), as well as using the bit pattern normally or inverted. With a little practice and insight, the results can be very professional looking indeed.

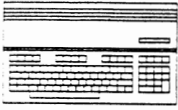
Finally, the new virtual image resizing and rescaling features of Geopaint are phenomenal. Essentially you can cut or load any image data as a photo scrap and paste it into any size area and the program gives you the option of having that image automatically stretched (or shrunk) to fit the area you define. In addition you can have the image scaled so that the horizontal and vertical aspect ratio remains in tact or even have the image smoothed. Also you can overlay the image "transparently" so that whatever is behind your image remains in view or place a fill pattern automatically in the background. I can honestly state that these the most sophisticated image cut and paste options I have ever seen in an eight bit paint program.

### Geowrite Enhancements

GEOS 128 v2.0 includes the enhanced version of Geowrite which was formally marketed as Geowrite Workshop 128. I applaud Berkeley in their decision to include their best version of Geowrite with the system. In reality the inclusion of a bare-bones version of Geowrite with the original GEOS 128 did nothing but frustrate and anger C-128 owners, especially since 128 owners have numerous first rate word processing choices. And given these choices, it is pretty hard to recommend even this version of Geowrite as their primary word processor. Text entry and reformatting when inserting is terribly slow and its ruler based mouse only approach to margins, while being visual and easy to comprehend, is also cumbersome when trying to create complex text formats such as outlines.

However, as a secondary word processing program, especially for certain situations it can be quite useful. Geowrite's two major strong points have always been its proportional font support and the ease by which graphics can be integrated with Geowrite text. The placement of graphics in a document is as simple as adjusting margins and selecting the paste graphics command. In addition, Geowrite has wonderful support for multi-line headers which can include graphics which are perfect for things like company logos. Because of these strengths Geowrite has always been a superior "printout" or "desktop publishing" word processing package. What I mean by this is that it is not unusual for Commodore 128 owners to compose text in another word processing package such as Paperclip, Pocket Writer, or the

Continued on Page 8



## GEOS 128 v2.0 REVIEW

*Continued From Page Seven*

Write Stuff and then transfer their text to GEOS for formatting, graphics inclusion, and printing. This type of Geowrite usage is further enhanced with GEOS 128 v2.0 by virtue of the fact that the Text Grabber program which accomplishes the conversion of alien word processing files to Geowrite is also packaged with this upgrade.

Also packaged with GEOS 128 v2.0 is the Paint and Overlay drivers which originally appeared with the Writers Workshop 128 product. These drivers add an incredible amount of flexibility to Geowrite. Essentially what they allow you to do is "print" a Geowrite document to a Geopaint file. This can be accomplished in two ways. The "paint" driver simply transforms a Geowrite document into a Geopaint file. Like the paint driver the overlay driver creates a Geopaint file from a Geowrite file, but the overlay driver does something unique. It allows you to create very complex Geopaint files by overlaying multiple pages of a Geowrite document. Using this technique you can create easily create multi-column documents. As an example, my mini-two column flyer on the front cover was created using this technique.

As an added bonus, Berkeley has seen fit to include Geomerge and Geospell with GEOS 128 v2.0. Of course most C-128 word processing packages include support for mail merge operations and spell checking, so in reality their inclusion in GEOS 128 v2.0 is nothing all that earth shaking, except when you consider that until the version 2.0 upgrades these items cost GEOS owners extra. I have mixed feelings about all of this. On the one hand I am glad that serious GEOS users will no longer have to spend extra to get the kind of facilities that come as standard options with other packages, but I can't help but feel sorry for those folks who did buy these facilities has add-ons or wonder how I would feel if I had spent the extra \$50 - \$60 it used to take to get these programs, now that these applications are standard with the system. Of course, those people who did purchase these add-ons did have the luxury of making use of them a lot sooner than the rest of us. As for the performance of Geomerge and Geospell, they do what they are supposed to do and operate quickly. The only remarkable features of these two standalone programs are the fact that Geomerge is capable of some very sophisticated conditional (essentially programmable) mail merges, and the fact that Geospell does have a very good word lookup/suggestion facility (somewhat of a rarity on 128). Also included with GEOS 128 v2.0 is Geolaser which allows you to print out your Geowrite and Geopaint documents to postscript laser printers.

### Conclusion

I am sure that very soon you will be seeing a lot of reviews and opinions on GEOS 128 v2.0. I am equally sure that none of them will be as tough as I have been. I want you to know that despite my some of my candid commentary and nitpicks, I think GEOS 128 v2.0 is a fantastic upgrade to the original version of GEOS 128. This version has made using GEOS fun again for me. While I can't say that GEOS is for everyone, I can say that I think many people including GEOS skeptics will truly enjoy this version of the package. For people interested in graphics, desktop publishing, or just an alternative way of using their 128, I think GEOS is money well spent. I am hopeful that Berkeley will continue to develop and enhance this product so that it encompasses some of the 128 developments of the past year as well as continue to develop add on GEOS applications.

### GEOS 128 V2.0 TIPS:

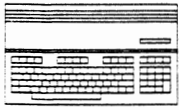
1. Using a RAM Expansion Unit with GEOS is a necessity. Typically GEOS users load up their RAM Expanders at the beginning of a session with the applications, fonts, and data they will be using. In previous versions of GEOS you were restricted to either copying a whole disk or single files at a time. GEOS 128 v2.0 allows you to batch (multi-file) copy multiple files to the REU at one time. However, even with this new capability it is still easier to diskcopy a pre-set REU disk to the REU. To do this, create a work disk you designate as your "REU loader". This disk should be a 5.25 inch disk rather than a 1581 disk. GEOS will not allow you to do a "diskcopy" from a 3.5 inch disk to the REU as the maximum available storage on even a 512K REU is far less than the capacity of a 1581 disk. Arrange or copy files onto the disk so that the icons for your GEOS application programs are on first page(s) of that disk. Then starting on a new page put your favorite fonts (if you are like me you will have several pages of fonts), then on the last page(s) of your disk place the desktop, the RAM Reboot file, your printer driver, the paint and overlay drivers and your desired desk accessories. Then use the page append command from the desktop's page menu to insert a new (blank) page 1 on your desktop. Likewise use this command to insert a new page at the page where your fonts begin. Once you have done these steps you will be able to quickly pre-load your REU via the diskcopy command.

Here is why: Essentially what we have done is placed all of the files you normally need to double-click on from the desktop on the frontmost pages of the desktop so that you do not have to scroll around in search of them. Putting all of your application programs together makes it much easier to switch between programs "on the fly". Because the first page is blank, GEOS will automatically place the icons for the subsequent application data you create or copy to the REU on that first page. This is convenient since you can double click on the application data icon(s) to automatically execute a particular application with your data loaded in it. Creating a blank page at the beginning of your font data makes it simple to change the fonts your applications "see" within the REU. GEOS applications are able to make use of the first eight fonts stored on a disk or REU. By leaving a blank page before the start of your font files, you can easily move a font file that is stored on your REU after the first eight font files into that "front eight" by placing it on the blank page at the beginning of the font files. Lastly, by putting the printer drives, desktop, the RAM reboot files and the desk accessories on the last pages of your REU you effectively "moves them out of the way" since these files ones that you do not normally utilize by double clicking them from the desktop. Desk accessories are usually started from the desktop or within applications via the "GEOS" menu as is printer selection and use of the photo and text managers.

2. One of the most tedious tasks that creeps up when using Geopaint is the centering of a "clipped region" or photo scrap. However, through creative application of the paint or overlay drivers and Geowrite you can save yourself a lot of headaches. After defining your region, use the cut option available from Geopaint's edit menu to cut the region as a photo scrap. Then exit Geopaint and from the desktop install the paint driver as your "printer" driver. Now start Geowrite (see why I say an REU is a necessity!). Paste the photo scrap into a fresh Geowrite document. You will notice that the photo scrap will be automatically centered horizontally across the page according to the margins that are established. If you like, you can change these margins for different positioning purposes. Once you have your graphic in the desired position, "print" your Geowrite document into a Geopaint file. An even more sophisticated application of this technique involves using the "overlay" driver to precisely position multiple images vertically as well as horizontally.

*If you like tips, catch Sparrow's return on Page 25*





## N.U.R.S.E. 128 REVIEW

by Bruce Jaeger

N.U.R.S.E. is a universal Commodore disk editor that should put to rest any further need to program disk editors. Period. While N.U.R.S.E. can be improved in certain areas (as can any program, regardless of price), I can't think of any disk-editor-type functions that have been left out!

First, a couple of questions. What is a disk editor, and why is this one called N.U.R.S.E.? A disk editor is a program that lets you look directly at the contents of a disk (hard or floppy), and inspect and change the bytes encoded on the disk directly in the original spot on the disk, without regard to whether they're in a sequential program, USR, random or other type file--or even if they're not in a file at all, like the Commodore boot header that is used to force an autoboot whenever the 128 is powered up or reset. Disk editors are used for "un-deleting" files, searching for lost data on scrambled disks, changing header names, disassembling and breaking copy protection schemes, creating copy protection schemes, modifying programs or data, and I'm sure several other things that I haven't mentioned.

A disk editor usually works by loading a 256-byte disk sector (from the track and sector you specify) from the disk into memory, and then letting you use familiar machine-language-monitor type commands to change any bytes you wish. The sector is then saved back to the disk, in its original location.

Why is this program called N.U.R.S.E.? (They had me in the hospital, until suddenly I took a turn for the nurse!) Loren and I couldn't figure it out. Maybe it's because you use it for working on bad disks--but then, I'd think you'd want to call it C.H.I.R.O.P.R.A.C.T.O.R. or something. Sorry, Dave, but the Reviewers Guild Rulebook requires at least one paragraph of feeble humor in every review.

The N.U.R.S.E. package is more than just a disk editor. Also included are the following utility programs (some with source code!):

Bootsector Utility 128, for creating, modifying, or killing autoboot.

Formatting the back side of 1541 disk in a 1571 drive.

1571 error checker.

Extramom 128; adds to the C128's built-in monitor up and down scrolling of disassembles, new and revised disk commands, interrogate command, current print device change, and checksum compute.

Basic print routines for the 64 and 128.

Source code formatter.

Screen dump for the 128

Super Search and Recover (an automated disk fix-up program for disks you've quick-erased by re-headering without giving an ID number).

Lots of miscellaneous auto-booting and machine-language loading routines to use in your own programs.

Much of the above doesn't even have much to do with "Disk Utilities," but it's sure nice to see them "gratis" on a disk. It's nice to get more than you paid for, rather than less, for a change.

But back to the N.U.R.S.E. disk editor. I sure wish that I had had this program a year or so ago, when I had to recover lots of lost files from source file disks trashed by the non-compatibility of 1541 disks and the 1571 drive with the original ROMs. I had to make do with a magazine type-in disk editor, and it would have been a heck of a lot easier

with N.U.R.S.E.. Just look at the information you get when you first load a sector:

A character-type dump of the 256 bytes. This is the main display, and it makes it easier to read the text located on the disk. Unprintable characters are designated with a reverse space.

The POSITION of the cursor in this block of 256 characters. (This is a number 1 to 256).

The HEX value of the byte the cursor is over, and of the eight bytes following it.

The DECIMAL value of the byte the cursor is over.

The machine-language DISASSEMBLY of the byte the cursor is over, starting with that byte. (If the cursor is located over a byte in the middle of an instruction, this could be nonsense, of course.)

The type of drive, DOS version, current track and sector, and the sector link.

If you can remember all the things that N.U.R.S.E. can do--and remember the key commands to do them--you'll find that this editor beats the eggs out of the rest. A help mode prints information about every command key (assuming you can remember which key you press to get the command you need help with!) Like all the best and most powerful programs I've seen, you have to invest some time in order to learn how to use all the features.

Every Commodore disk drive ever manufactured is supported, even drives 0 and 1 on dual drives, and all disk devices 8 - 11. You can dump a sector printout to any printer, on the "regular" print device #4, and including device #5 (used by many parallel-port print interfaces) and the "plotter" device 6. You can type in changes in decimal, hexadecimal, or (hurray!) text. (Finally, no more having to remember how to type in "4E" to change a byte to a capital "N".)

N.U.R.S.E. lets you do memory fills, ANDs, ORs, adds, subtracts, and XORs with any value, and you can do search and replaces--with wildcards. You can copy sectors to another disk, or to a different sector on the same disk. You can follow a file from sector to sector, or place the cursor over a line link in the directory, and jump to the start of that file.

I realize I'm jumping from feature to feature, and there are a lot of them. The manual covers everything quite well, and will explain for you that "line link" stuff mentioned in the previous paragraph.

Here's something neat. I hate the colors that N.U.R.S.E. comes up in: icky shades of dark blue that are hard to read, and other shades of blue/cyan that smear a lot next to them. But N.U.R.S.E. lets you change those colors, and in an interesting way. You don't load a "change preferences" screen, which the program then has to load each time it runs, but the manual shows you step-by-step how to use N.U.R.S.E. itself--and many of its more advanced features--to directly modify the color settings in N.U.R.S.E.. This gives you an interesting, hands-on practice session that actually does something useful! How many of you have skipped over "tutorials" that in the end are merely nonsense typing sessions?

Don't worry about wrecking your only disk. N.U.R.S.E. isn't copy protected, and you're encouraged to make backup copies and keep the original in a safe place.

Continued on Page 10


## N.U.R.S.E. 128 REVIEW

Continued from Page 9

I do have a couple of gripes about N.U.R.S.E., of course. (Remember that Reviewers Guild Rulebook I mentioned earlier? Another rule is that I have to have at least one paragraph of petty nit-picking.)

When N.U.R.S.E. first boots up, and when you initially load a track and sector, the program title "streams" onto the page with the characters running in rivers up and down and across the screen. This is cute. Once. After that, it seems to take forever. You can stop it by pressing a key--but you shouldn't be put in a situation where you have to. I thought I'd seen the last of this stuff back in my PET and TRS-80 days, but I guess I was wrong. The routines for making displays like this are included on the disk for you to play with in your own programs.

A more important shortcoming is that, while N.U.R.S.E. automatically supports 80 column and FAST mode on the 128, the 80-column display is just the 40-column display positioned in the center of the screen. Like an assembler I reviewed once, here we're paying for 80 columns, and not being allowed to use it. I can understand leaving the display the same from 40 to 80 columns, but the rest of the screen could have been put to better use; like a reminder of the command keys, for example.

To sum it all up, N.U.R.S.E. is a feature-laden, well-documented disk editor program for heavy-duty Commodore programmers. (If you just use your 64 or 128 to play Kill the Icons, you don't have much need for a disk editor.) The extra programs are an added bonus.



## NOTHING LOADS YOUR PROGRAMS FASTER THAN THE QUICK BROWN BOX !!

A NEW CONCEPT IN COMMODORE CARTRIDGES

STORE UP TO 30 OF YOUR FAVORITE PROGRAMS IN A SINGLE BATTERY BACKED CARTRIDGE FOR EASY AND INSTANT ACCESS! CHANGE THE CONTENTS OF THE BOX AS OFTEN AS YOU WISH. THE QUICK BROWN BOX ACCEPTS MOST UNPROTECTED AND "FROZEN" PROGRAMS INCLUDING THE ONLY WORD PROCESSOR THAT SAVES YOUR TEXT AS YOU TYPE: THE WRITE STUFF! THE BOX EVEN CO-EXISTS WITH GEOS, AND THE COMMODORE RAM EXPANSION UNITS. LOADER UTILITIES INCLUDED FOR BOTH THE COMMODORE 64 AND THE COMMODORE 128.

32K QUICK BROWN BOX...\$99    64K QUICK BROWN BOX...\$129

UTILITIES DISK..\$6    Q-DISK CP/M RAMDISK...\$10

### SPECIAL PACKAGES

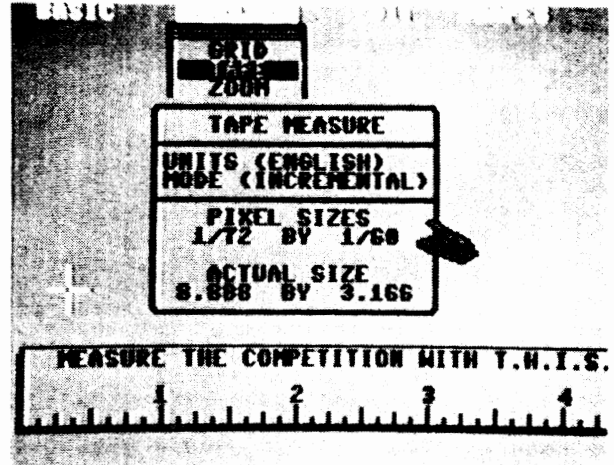
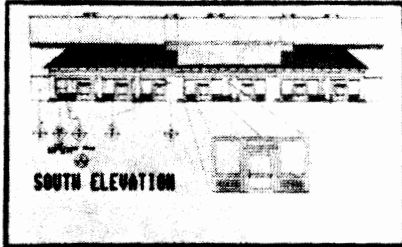
64K QUICK BROWN BOX & THE WRITE STUFF 64...\$139  
128K QUICK BROWN BOX & THE WRITE STUFF 128...\$144

**Brown Boxes Inc.**  
26 Concord Drive  
Bedford MA 01730

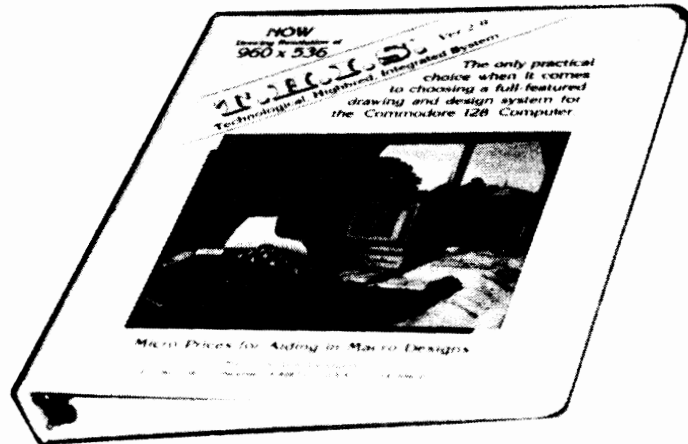
617-275-0090 617-862-3675

Add \$3.00 shipping and handling  
Mass. orders add 5% state sales tax  
C.O.D add \$3.00, overseas add \$5.00

Micro Aided Designs  
 P.O. Box 1982  
 Placentia, CA 92670  
 (714) 996-0723



**T.H.I.S.**



**THIS** (Technological Highbred Integrated System) is a program designed exclusively for the Commodore 128. Listed as one of the best programs of 1987 by "Commodore Magazine," **THIS** contains many advanced features not found in any other program. **THIS** proves to be the most user-friendly graphics program, complete with a step-by-step easy to understand manual. Using a 1700 or 1750 RAM and a light pen or mouse, **THIS** is a drawing system capable of producing a simple doodle or a sophisticated blue print.

With **THIS** you have a possible resolution of 960 by 536 dots per drawing page and seven (7) totally separate pages to draw upon. Giving the user a larger area to draw on not only allows for larger sized drawings but also increases the amount of details that can be added.

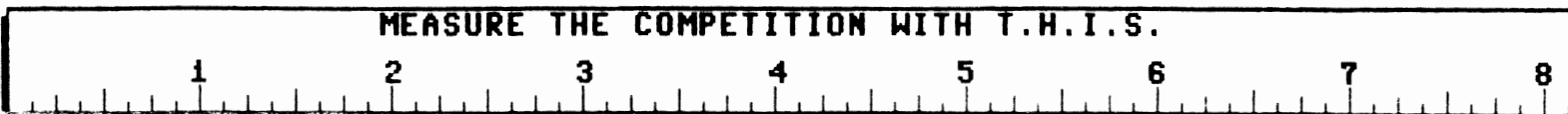
When it comes to the final step of printing a complete drawing, **THIS** offers exact-scaling. With exact-scaling the completed drawing can be printed to scale within 1/64 of an inch. **THIS** allows for the possibilities of templates to be created and used in real life applications. The printout sizes can vary anywhere between 2 by 2 inches to 480 by 321 inches.

### EXCLUSIVE FEATURES

- Up to 7 pages of graphic designs
- Maximum virtual screen resolution of 960 by 536 pixels per page
- Exactly scales printouts as large as 480 by 321 inches
- Real time object rotation and move
- Real time cut, copy, paste, and zoom
- Elastic modes include lines, boxes, circles, ellipses, arcs, and other geometric shapes
- Easily accessed graphics library including over 100 detailed electronic symbols
- Over 700 type sizes
- 10 font styles

Retail Price \$63.99

MEASURE THE COMPETITION WITH T.H.I.S.



## JUGG'LER 128 REVIEW

by Bill Juliani

Jugg'ler-128 is written by Miklos Garamszeghy of Herne Data Systems LTD, Canada. If you are a reader of the Twin Cities 128 and Transactor Magazines, you are all familiar with Miklos' qualifications of being the CP/M GURU of Commodore's 128. MIKLOS has picked up supporting this mode where Von-Ertwine has left us last May '87.

Juggler runs in CP/M mode, that allows the C-128's various 'smart' drives to read, write, and format to many different types of CP/M disks from other brands of computers. In effect, Juggler converts the C-128 into a universal CP/M computer and lets the user swap disks from one computer to another without worry. This is especially helpful for those who use different computers at work and can take data disks home, or that have hundreds of data disks on an older CP/M machine and want to upgrade their system, without going through the hassle of data transfer or null modem routines. This also applies to program disks. If you have a CP/M Plus (or CP/M 3.0) program from your old CP/M computer that includes an installation module for screen codes then these will work very well on your trusty C-128! The System Guide by Commodore gives a brief description of reading several types of MFM disks from other CP/M machines. Jugg'ler takes this valuable option from less than ten, to over 160 by patching the Disk Parameter Table.

I like menu driven programs, windows and flashing messages. Jugg'ler uses all of these. Typing errors can be corrected before hitting <return> and default values for prompts are set to abort the current operation. Selecting and installing a disk type allows you to work with that foreign disk for as long as the system is in CP/M mode. You can also save up to three different types of disk formats for re-use. These three 'saved' disk types are stored with the file JUGG.DEF on your copy of Jugg'ler, and several copies of Juggler can be made, each having different default files. Juggler even has a built-in PIP (copy) routine that will back itself up to another disk. Once you have decided on which three format types you will be using most often, and have selected the save option, Jugg'ler can be run by several different methods.

Many options for running Juggler are allowed, with the most useful being formatting a previously saved default disk type directly from the CP/M prompt. This option is as easy to use as one could imagine. Just hit 'JUGGF' at the CP/M prompt and a smaller menu will be displayed for choosing your selection. Another handy option is an auto-copy mode of Juggler into the REU (device M) with the command JUGGM or, into the QBB (device C) cartridge with JUGGB command. This is great for the speed freaks and RAM lovers. Similar to Q-Disk's self patching mode, Juggler will also allow the user to directly patch the CCP.COM for auto-booting. This patch, sadly, does not allow the use of PROFILE.SUB files, and can not be used with the auto-booting feature of Q-Disk. I would recommend staying with a bare bones CCP and use a PROFILE.SUB for both. In all, there are nine different Juggler shortcut options that will quickly access Juggler, and get you back to your current task.

When I first got my 1581 drive, and before my long awaited 28MAY87 disk arrived I desperately needed CP/M support for my new little/big drive. Miklos' programs that modified the 8DEC Systems Disk and '81 (MG1581) Format.COM solved this problem, but created a new headache, when I later switched to the 28MAY87 disk. I suddenly found that they were not compatible! My dBase files were in shambles! Jugg'ler's MG1581 disk option would have saved me hours of frustrations and file copying.

Also included are two unique Librarian and Maxi types for both the 1571 and 1581. The Librarian increases the number of files that can be stored on a disk, and is handy for those little 2K dBase files, while the MAXI increases the total storage disk space.

The versatile 1571 can also be modified to read Single Density (FM) disks by installing a switch and resistor to pin 26 of the WD177X controller chip. This modification used with Juggler opens another software door for the C128 CP/M user. While formatting Single Density disks are not possible, R/W operations let you use the many Osborne 1, TRS-80 Model I, Xerox 820, and Zenith 37 programs.

Juggler in itself a work of art, and its documentation is an excellent example for all software people to follow. Printed on 8 1/2 by 11 paper with an introduction, table of contents, beginner's area, and explanation of terms that is all typeset. The docs even point out subtle quirks in CP/M Plus giving the user some well needed insight in CP/M DOS. The section titles are large for quickly locating needed information, and under Technical Details, Miklos explains in simple language how Juggler works. However, the skews are missing from the Summary of Disk Types, and Miklos needs to eliminate the double notation for special drivers (!) and normally supported formats (\*) in the same table.

The most interesting option is for analyzing an unknown CP/M disk. Before you start cramming all sorts of 3 1/2" disks in for analysis, be warned that because of the '81 (and not Juggler) any GCR disk subjected for analysis will cause the '81's LED to stay on, indicating an open command channel. The docs mention an error message, but none appear, and you might think that the drive is locked up. Any disk access WILL clear the command channel, and any disk swapping while in this condition will be harmless. Future updates will clarify this in the documentation.

Juggler is a versatile, well written CP/M Program for the C-128. If you are a CP/M user, then this is a 'must have' for your library. With the loss of the 1571 drive, Juggler will surely play an important role with the 1541 and 1581 CP/M formats. Want IBM compatibility in 128 mode? Use Big Blue Reader. For CP/M compatibility, get Jugg'ler.

# MAS-128

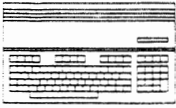
Full-Featured 6502/6510/8502 Symbolic Assembler for the C-128 based upon the Innovative C-64 Midnight Assembly System.

- ☆ 80 Column RGB Full Screen Editor.
- ☆ Executes from RAM 0 or RAM 1.
- ☆ Supports 1700/1750 REU with Built-in Proprietary RAMDOS, complete with commands for transferring entire disks to and from REU.
- ☆ Makes absolutely NO calls to ANY of the the C-128 ROMs (contains its own keyboard, video, and serial I/O drivers). Indispensable to the programmer who wishes to design C-128 ROM replacements.
- ☆ Multiple Statements per line of Source Code.
- ☆ Not Copy Protected! Archival Backup Encouraged.

Direct Inquiries to:

**Mountain Wizardry Software**  
P.O. Box 66134  
Portland, OR 97266  
Tech. Support (503) 265-2755





## GEOS to BASIC 8

by Rick Kane

We all love to show off what can be done with RGB graphics on the 128. But when it gets right down to it, it takes a lot of time (not to mention talent) to draw all those pictures to show off with. And many very nice graphics already exist, available on BBS's, having been either ported to a compatible format from Amiga or Macintosh formats, or created with one of the several fine paint packages available on the 128 and 64. Of course, each has a file format tailored to its characteristics. And just because you can download a graphic and view it is no guarantee that you'll be able to use it in your own programs.

So no one really needs to be told the value of having the tools to convert from one format to another. With this program, you will be able to take a Geos Photo Scrap (the file you get when you CUT or COPY from the Geopaint EDIT requester) and make an 8x8 color Basic 8 BRUSH file. The brush can then be pasted, cropped, and otherwise handled just as if it had been drawn with a Basic 8-based paint program such as Basic Paint, Spectrum 128, and others.

This brings up a bone I have to pick. I know Berkeley Softworks has just released GEOS 128 v2.0, and it has many improved features over its predecessor. But here we are, quite easily displaying color graphics in 80 column mode with only 16k of video RAM, crying for color support in arguably the best graphics program available in the 8 bit world. The standard answers are that it would take too much memory, it would be too slow, etc. But we're talking the 128 here, and while in 80 column mode there are 9000 bytes of programming space available that are normally occupied by the 40 column screen. Slow? Not to anyone who's been suitably impressed with the abilities of Basic Paint. Besides, the 80 column screen allows the use of FAST mode. I've captured an 80 column Geopaint screen and squashed it to 640x176 (with the same work screen area), just to prove how little the appearance would need to change. Ah well, I'm sure someone will hear us someday.

As to the conversion process, Geopaint presents special problems, with its compaction schemes to keep disk access times down, VLIR format, and USR files. By concentrating on Photo Scrap files, we cut down on the complexity in several ways. First, Photo Scraps are sequential in format (though labeled USR), so a relatively simple get# loop will read it. Second, a Geopaint file can represent 640 x 800 virtual resolution. Holding that document in memory (uncompressed) would require 64000 bytes for bitmap and 8000 bytes for attributes, which would require a fair amount of juggling. Then it would need to be broken into smaller parts anyway, for 16k systems and for any paint programs where the picture remains in memory. (The maximum 8x8 color document with 64k RAM would be 640 x 728).

As luck would have it, the Geos programmers chose to save their picture data from left to right across scanlines, just like the 8563 chip, rather than in the "8 bytes stacked per character" method used by the VIC chip. This was done because bit-mapped data compresses better horizontally. This makes sense, if you think about the case of a pattern fill. VIC patterns are generally based on 8x8 cards, so a pattern fill would place the same byte at the same scanline position in each card. This results in a higher probability overall that the byte in the position immediately to the right (or left) of one position will be the same, leading to better compaction. In any case, that takes some of the work out of getting the data onto the 8563 screen.

Geos uses three modes to maximize compaction. In each mode, the first byte is taken as the COUNT, followed by the appropriate amount of data. In the first mode, when COUNT < 128, indicates that the next byte is to be repeated COUNT times. The second mode is a string of data -- a 'word'-- used sequentially. COUNT is between 128 and 220 inclusive, and the data string is the next COUNT-128 bytes, used once

each. The third mode, BIGCOUNT, allows further compaction of patterns. If a string of bytes repeats several times, COUNT will be between 221 and 255. The number of data bytes is COUNT-220. The next byte is BIGCOUNT, telling how many times to repeat the string of data bytes.

The decompression routine is not fast, being written in Basic, but it uses FAST mode and really isn't that bad. A COUNT byte is gotten from disk, and the appropriate decompression routine is called for each mode, until the Photo Scrap's full dimensions have been addressed. When Geos creates the scrap, it doesn't distinguish between bitmap and color data; they are compressed the same way. So it is possible for a compression at the end of the bitmap to include both bitmap and color data. This will happen if, for example, the last row of the scrap is 00's (the scrap is slightly larger than the object you cut, perhaps, and contains blank space) and the first color data is the same. So each mode routine must check if data to repeat is left over when the last scanline is done, and finish off at the start of attributes. If no data is left, the switch to attribute memory is handled at the proper time by the main routine at 1000 - 1080.

Color bytes are in the VIC format, where 0 is black, 1 white, 2 red, etc. But the RGB colors don't match those numbers; so a conversion array is set up. Also, the color nibbles effect foreground/background in the reverse way; this is also handled when the byte is gotten from disk if the color change flag is set. If the Photo Scrap is monochrome, cut either from the 80 column screen or with COLOR OFF on the 40 column screen, there will be just one byte of color data for the entire scrap. In this case, the get-a-byte routine at 5000 will recognize the end of file marker (ST < > 0) and just leave the current byte to be handled until the scrap is filled.

Once the Photo Scrap is on the screen, it can be @STASHed in compressed form to a brush structure in a buffer. I chose bank 1 for the buffer because the data is relatively small (no more than 80\*176 + 80\*22, or 15840 bytes maximum uncompressed), and it is an easy matter to protect Basic's variables from corruption. After the brush is @STASHed, it can be written to disk with the @SSTRUCT command. To get a better understanding of the formats for Geos and Basic 8, I suggest 'The Official GEOS Programmer's Reference Guide' by BSW, and the Basic 8 Manual. You don't need to own Basic 8 to run the conversion program, but you do need the runtime library, which is available through most BBS's and clubs. Then use the program "GEOS/B8 CONVERT" to boot the runtime library and run "GEOSCRAP-2B8". If you own basic 8 you can simply boot the editor and run the latter.

When you run the program, you will be told that the menu is on the 40 column screen. It then asks for the source drive. If you answer '0', the program will end. After a catalog of all USR file on the appropriate disk, enter the filename you want to convert. Usually, you can leave it as 'Photo Scrap', because Geos won't let you rename it to anything else. You can rename them in Basic to put many scraps on one disk, but this is only practical if you have a RAM expansion unit to allow for quick reboots each time you exit Geos to rename. The program will handle 'not found' errors, but I make no guarantees about what will happen if you try to convert some other USR file, like the Desktop! When the file is found, the width and height in pixels is displayed, then the VIC screen blanks while the brush is drawn. When it is done, you'll be asked for a filename ('brus.' will be automatically added to the name you give) and the brus is saved. To display a brus quickly, exit the program, then type this line:

```
@clear,0,3,15:@display,0,device,0,brushname,0,0
```

Continued on Page 14

## GEOS to BASIC 8

Continued from Page 13

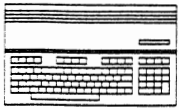
### GEOS/B8 CONVERT:

```
10 DV=PEEK(186):FAST:BLOAD"RTL8",BO,P4864,U(DV):SYS4864
20 BLOAD "P.PTRDEFS",BO,P2816,U(DV)
30 KEY8,""
40 RUN "GEOSCRAP-2B8V1.2",U(DV)
```

### GEOSCRAP-2B8V1.2:

```
0 REM GEOSCRAP-2B8V1.2
1 REM BASIC 8 FORMAT CONVERTER - 16K SYSTEMS
5 REM FROM GEOS PHOTO SCRAP
10 REM BY RICK KANE <C> 1989
13 BANK 15
15 FAST:@WALRUS,0:GRAPHIC5:@TEXT:PRINT"SEE 40 COLUMN SCREEN FOR
MENU":GRAPHIC0
25 POKE47,0:POKE48,DEC("94"):CLR:REM MOVE START OF VARIABLES UP
30 DIM WRD(35),CL(15):GOSUB5500:REM DEFINE COLORS
40 WR=DEC("CDCC"):WM=WR-2:REM WRITE 8563 REGISTER, MEMORY
60 DV=PEEK(186)
100 BM=0:AT=14500:REM BITMAP, ATTRIBUTE ADDRESSES
110 @SCRDEF,0,0,3,640,176,BM,AT:REM SET UP 16K 8X8 SCREEN
120 @BUFFER,1,1024,36863:REM BUFFER IN INT BANK 1 AT $0400
130 TRAP 9999
200 SLOW:SCNCLR:PRINT CHR$(14);"GEOS SCRAP TO B8 BRUSH IMPORTER"
201 INPUT"SOURCE DEVICE NUMBER OR 0";DV:IF DV=0THEN 9999:ELSE
IFDV<U8 OR DV>23 THEN200
202 IF DD=0 THEN DD=DV
203 INPUT" DEST. DEVICE NUMBER";DD:IF DD<8 OR DD>23 THEN202
205 PRINT"INSERT GEOS DISK IN DRIVE"DV:PRINT"THEN PRESS A KEY"
210 GETKEY$
212 CATALOG"*=U",U(DV)
214 PSS="PHOTO SCRAP":I=LEN(PSS):POKE 2592,I:FOR J=1 TO I:POKE
841+J,ASC(MID$(PSS,J,1)):NEXT J:POKE 208,I:INPUT"NAME OR
$";PSS:IFLEFT$(PSS,1)="$"THEN 200
216 IF PSS="PHOTO SCRAP"THEN GOSUB6000:REM CONVERT TO ASCII
220 PRINT"SEARCHING FOR PHOTO SCRAP...";
300 CLOSE2:CLOSE15:OPEN15,DV,15: OPEN2,DV,2,PSS+",U,R":
INPUT#15,EN,E$,ET,ES
310 IFEN<>0 THEN CLOSE2: CLOSE15:PRINT:PRINT"ERROR:"E$: SLEEP1:
GOTO200
315 PRINT"FOUND!":PRINT"READING..."
320 GET#2,WD$,HL$,HHS:WD=ASC(WD$):HT=ASC(HL$)+256*ASC(HHS):REM
WIDTH, HEIGHT
325 PRINT"WIDTH:"WD*8:PRINT"HEIGHT:"HT:SLEEP1
330 FAST:@SCREEN,0:@CLEAR,0,3,15:REM WHITE ON LIGHT BLUE
340 SYSWR,22,6:REM STOP DISPLAY AT 176/8 CHARACTERS
399 REM START OF DECOMPRESS/CONVERT ROUTINE
400 UP=BM:GOSUB900:HB=HT: REM UPDATE = BITMAP, HEIGHT.BYTES =
HEIGHT.PIXELS
405 CC=0:BOTSCR=0:REM COLOR CHANGE FLAG, BOTTOM OF SCREEN
410 GOSUB1000:REM CONVERSION ROUTINE
420 CLOSE2
540 SLOW:PRINT"PRESS A KEY...":GETKEY$
545 NAME$="":INPUT "OUTPUT FILE NAME ('BRUS.'+)";NAME$
546 IF NAME$="" THEN BEGIN:INPUT "CONTINUE WITHOUT
SAVING";Y$:IFLEFT$(Y$,1)<>"Y"THEN545:ELSE214:BEND
550 NAME$=LEFT$("BRUS."+NAME$,16)
555 AD=@STASH,0,1,0,0,0,WD*8,HT,1:REM STASH BRUSH IN BUFFER 1
560 IFAD<0THENPRINT"STASH FAILURE.":SLEEP4:GOTO590
570 @SSTRUCT,0,DD,NAME$:@SEND:REM SAVE IT
580 INPUT#15,EN,E$,ET,ES:IFEN<>0 THEN
PRINT:PRINT"ERROR:"E$:GOTO540
590 GOTO200
900 REM SET UPDATE ADDRESS
910 UH=INT(UP/256):UL=UP-256*UH
920 SYSWR,UH,18:SYSWR,UL,19:RETURN
999 REM START OF FILE READ/SCREEN WRITE
1000 XP=0:YP=0
```

```
1010 DO
1020 : GET#2,COUNT$:COUNT=ASC(COUNT$)
1030 : IF COUNT<128 THEN GOSUB 2000
1040 : IF COUNT>127 AND COUNT<221 THEN GOSUB 3000
1050 : IF COUNT>220 THEN GOSUB 4000
1060 IF BOTSCR=1 AND CC=0 THEN GOSUB1500:REM BOTTOM OF SCREEN
REACHED 1ST TIME
1065 IF BOTSCR=2 THEN EXIT
1070 LOOP
1080 RETURN
1499 REM SWITCH TO START OF ATTRIBUTES
1500 UP=AT:GOSUB900:HB=INT(HT/8):REM HEIGHT.BYTES =
HEIGHT.COLOR.CARDS
1510 CC=1:XP=0:YP=0:REM COLOR CHANGE ON, X, Y POS ZEROED
1520 RETURN
2000 REM MODE 0; SINGLE BYTE REPEAT
2010 GOSUB5000:REM GET A BYTE
2020 I=0
2030 DO UNTIL I=COUNT
2040 SYS WM,BT:I=I+1:XP=XP+1:IF XP<WD THEN 2050
2044 UP=UP+80:GOSUB900:XP=0:YP=YP+1
2046 IF YP<HB THEN 2050:REM NOT END OF BOTTOM ROW
2048 BOTSCR=BOTSCR+1:IF I<COUNT THENGOSUB1500:REM IF LOOP
UNFINISHED, SET TO ATTRIBUTE ADDRESS AND CONTINUE
2050 LOOP
2060 RETURN
3000 REM MODE 1; SINGLE WORD; 'COUNT' BYTES LONG
3010 COUNT=COUNT-128
3020 I=0:DO UNTIL I=COUNT
3030 GOSUB5000
3040 SYS WM,BT:I=I+1:XP=XP+1:IF XP<WD THEN 3050
3044 UP=UP+80:GOSUB900:XP=0:YP=YP+1
3046 IFYP<HB THEN 3050
3048 BOTSCR=BOTSCR+1:IF I<COUNT THEN GOSUB1500
3050 LOOP
3060 RETURN
4000 REM MODE 2; WORD REPEAT 'BIGCOUNT' TIMES
4010 COUNT=COUNT-220
4020 GET#2,BIGCOUNT$:BIGCOUNT=ASC(BIGCOUNT$)
4030 FORI=1 TO COUNT:GOSUB5000:WRD(I)=BT:NEXT:REM GET WORD
4040 J=0:DO UNTIL J=BIGCOUNT
4050 : I=0:DO UNTIL I=COUNT
4060 : SYS WM,WRD(I+1):I=I+1:XP=XP+1
4070 : IF XP<WD THEN 4080
4074 : UP=UP+80:GOSUB900:XP=0:YP=YP+1
4076 : IFYP<HB THEN 4080
4078 : BOTSCR=BOTSCR+11:IF I<COUNT THEN GOSUB1500
4080 : LOOP
4090 J=J+1:LOOP
4100 RETURN
5000 IF ST THEN RETURN:REM END OF FILE; BT UNCHANGED
5005 GET#2,BT$:BT=ASC(BT$)
5010 IF CC=1 THEN BT=16*CL(BT AND 15)+CL(INT(BT/16)):REM
REVERSE COLOR NIBBLES & CONVERT
5020 RETURN
5499 REM DEFINE RGB EQUIVALENTS FOR VIC COLORS
5500 RESTORE5510:FORCL=0TO15:READCL(CL):NEXT:RETURN
5510 DATA 0,15,8,7,10,4,2,13,11,12,9,1,6,5,3,14
5999 REM CONVERT PSS TO ASCII
6000 PP$="":FORI=1TOLLEN(PSS)
6010 P=ASC(MID$(PSS,I,1))
6020 IF P>192 THEN P=P-128:GOTO6050
6030 IF P>64 AND P<96 THEN P=P+32:GOTO6050
6050 PP$=PP$+CHR$(P)
6060 NEXT
6070 PSS=PP$:RETURN
9999 DCLEAR ON U(DV):GRAPHIC 5:@TEXT:FAST
```



## DUAL VIDEO FUN

by Miklos Garamszeghy

"BOMBS-AWAY" is a two player game for the C-128 requiring two joysticks and two video monitors (or one monitor and a television set). Each player controls their portion of the game from a separate screen.

The C-128 has many interesting, often unique, features for a home computer. One such feature is its dual video capability. The C-128 has two separate, totally independent video output chips: the 8564 VIC II (used in 40 column mode and for the TV RF output) and the 8563 VDC (used in 80 column mode).

The ability to use two or more separate video screens is often found in high priced business micro and mini computers. It is particularly useful in graphics applications (such as computer aided drafting, or CAD) where menus and command prompts are often displayed on one screen, while the graphic image is displayed on a second. Besides providing a larger on screen graphics display area, use of a second screen leaves the primary graphics display uncluttered by menus and command prompts. This is very handy for making hi-res graphic screen dumps of just the image portion, or for taking photographs of a screen image.

Many commercial software packages for the C-128 will work with either 40 or 80 column mode individually but very few, if any, will take full advantage of using both display screens simultaneously to display different information. (Strictly speaking, only one video processor is actually used as the output device at any given time. The illusion of using two screens "simultaneously" is created by switching back and forth between the video processors, as required for updating each screen.) Because each video display processor has its own screen RAM on the C-128, switching screens will not erase either display.

### SWITCHING SCREENS WITH BASIC:

C-128 BASIC 7.0 contains several methods for switching video displays. Perhaps the simplest and most direct method is with the GRAPHIC command:

```
GRAPHIC 0 GIVES YOU 40 COLUMN TEXT MODE
```

```
WHILE GRAPHIC 5 GIVES YOU 80 COLUMN TEXT MODE
```

This works in either immediate or program mode. In program mode, you can also toggle (that is switch) displays, either 40 to 80, or 80 to 40, with a simple:

```
PRINT CHR$(27)"X"
```

This is equivalent to the command Escape (or <ESC>) X in immediate mode. A third method for use in immediate mode only, is to select the appropriate position of the "40/80" key and then press the <RUN/STOP> - <RESTORE> key combination for a soft system reset, or the reset button for a cold start.

Since each screen is totally independent, any character attributes, WINDOWing, or cursor positioning that you want displayed on both screens (such as colors, reverse text, etc) must be repeated for each screen. For example, printing the control code for white characters (CHR\$(5)) when the 40 column screen is on will not affect the color of the 80 column screen characters.

Most BASIC 7.0 screen output commands such as WINDOW, PRINT and CHAR work on the currently selected display screen only. With BASIC 7.0, the graphics oriented commands, such as LOCATE, CIRCLE, DRAW, etc., however, only work with 40 column graphics mode, regardless of what the currently selected display mode is. In order to be able to view the product of the graphics commands, however, you must select one of the appropriate medium or high resolution graphics modes for the 40 column screen. (To use 80 column graphics, you generally need an enhanced version of BASIC, such as BASIC 8.0.)

You can also "print" to the 40 column screen while in 80 column mode. This is done by POKEing the appropriate character codes to the proper locations of the 40 column chip video RAM (hex \$0400 to \$07ff; dec 1024 to 2047). A similar, although somewhat more complicated, procedure can be used for sending character information directly to the 80 column video RAM while in 40 column mode. (The 80 column chip has 16 K (normal C-128) or 64 K (C-128-D or enhanced C-128) bytes of dedicated display memory which can only be accessed indirectly through the control registers of the 8563. The memory locations cannot be POKEd directly by either BASIC or machine language.)

### SWITCHING SCREENS WITH MACHINE LANGUAGE:

There are also several ways to switch displays using machine language. The KERNAL has a routine accessed through hex \$ff5f (decimal 65375) which is equivalent to BASIC's PRINT CHR\$(27)"X" or <ESC> X statement. This can be called from within your ml code with a simple:

```
JSR $FF5F
```

or from BASIC with:

```
SYS 65375
```

This routine, which eventually ends up executing code at hex \$cd2e, swaps the contents of the "active" screen pointers at hex \$e0 (dec 224) to \$f9 (dec 249) (those for the current screen being printed to) with those of the "passive" screen starting at hex \$0a40 (dec 2624) (those for the screen being switched to). These pointers include things like current window positions, current cursor column and line, etc. They are needed for switching back again in a controlled manner. The passive data storage are is just a buffer used for temporary storage of the pointers for the screen which is not currently being used.

The KERNAL routine also sets the value of memory location hex \$d7. You can check which screen is active by PEEKing location hex \$d7 (dec 215). If this location has a value of hex \$80 (dec 128), the 80 column screen is active. If the value is 0, the 40 column screen is active. If you do not care about preserving current screen windows, cursor location, etc., you can switch screens directly in either BASIC or machine language by changing the value of this byte. You can also determine the current display mode from the RGR(x) function. RGR(x) equal to 0 indicates that the 40 column text screen is active, while an RGR(x) value of 5 means that the 80 column display is on.

### A SIMPLE DEMONSTRATION:

"BOMBS-AWAY" is a simple, yet fun, demonstration of using two screen video on the C-128. It is a two player game written in BASIC 7.0 that requires two joysticks and two video monitors (or one monitor and a television set). Each player controls their portion of the game from a separate screen. (The display screens should be positioned back to back so that opponents cannot see each other's screen.) Player 1 uses the 80 column screen and joystick port 1, while player 2 uses the 40 column screen (or TV set) and joystick port 2.

The object of the game is similar to the classic "Battle Ship" strategy game: each player "hides" a number of targets (10 in this case) on a 10 by 10 grid. The players then take turns dropping "bombs" on their opponent's grid. You score by hitting one of your opponent's targets. The first player to hit all ten of the opponent's targets is the winner.

Type in and SAVE a copy of the program in LISTING 1. It is quite short, but pay careful attention to the spacing and special cursor control characters contained in the PRINT and CHAR statements.

Continued on Page 16

## DUAL VIDEO FUN

Continued from Page 15

After a brief opening introduction screen, the game set up screen will appear on both displays. The top left quadrant of the screen contains a grid of 10 by 10 elements. This is where you will initially hide your own targets during the set up phase of the game; and then move about to hit your enemy's targets during the play portion of the game. It will initially be filled with graphic "X"'s (shift-v's). At the top right of the screen there will be a row of 10 circles (shift-w's). These represent the number of your targets which you have not yet hidden.

### HIDING YOUR TARGETS:

Player 1 will be asked to hide the targets first. This is done by moving the joystick controlled flashing cursor around the grid to the location where you wish to hide a target, and then pressing the fire button to insert the target. The number of remaining targets displayed at the top right of the screen will be decreased by one for each target hidden. The "X" in the grid will be replaced by a circle if a target is hidden there.

If you change your mind (before placing the last target) and want to change the location of a hidden target, press the fire button when the cursor is in the cell containing the target that you wish to move. The cell will change back to an "X" and the number of remaining targets displayed at the top right will be increased by one.

When player 1 has hidden all of his targets, the procedure will be repeated on the 40 column screen for player 2. When player 2 has finished hiding targets, the main game screen will set up and the game will begin with player 1.

### PLAYING THE GAME:

The object of the game is to locate and bomb all of your opponent's targets before he/she gets yours. Each player will be given a chance, in turn to drop a bomb on the other player's grid. If you score a hit, you get another turn, repeating until you miss. If you miss, the other player has a chance to bomb you.

The 10 by 10 main playing grid at the top left of the screen is initially filled with "X"'s. Under these "X"'s are your opponent's hidden targets. You cannot see your opponent's targets and must guess at their locations. As you score hits, the X's will be replaced by O's. Locations that you have bombed, but missed a hidden target, will be shown as blank spaces.

At the top right of the screen is a mini 10 by 10 status grid. This second grid contains the locations of YOUR targets and also your opponent's shots, so that you can keep track of how close your opponent is getting to your targets and how many you have left.

Move the cursor over the main grid to where you think your opponent has hidden a target and press the fire button. At this point several things happen. The corresponding location on your opponent's status grid (the top right one on his display) will be blanked out. If you scored a hit, the "X" in your main playing grid will be replaced by a "O" and you will get another turn. If you missed, the location on your main playing grid will be blanked out and play will switch to your opponent. In either case, the score display at the bottom right of the screen indicating number of hits and number of shots will be updated. Play ends when one player has hit all ten of his opponents targets. Simple, yet suitable, sound effects have been added at appropriate points to spice up the game.

Although the rules may sound complicated on paper, they are really quite simple once you have played a few rounds of "BOMBS-AWAY"!!

For those who are interested, the program works like this. Lines 10 to 100 set up the default values for the colors and various string variables used to draw the screens. Note

that both the 40 and 80 column screens are created in text mode using the block character set. That is, 40 column hi-res bit map graphics mode is not used.

Array P(2,10,10) contains the data on where each player has hidden targets. Array H(2,10,10) contains the data on where each player has dropped bombs. (Although it is recognized that these two functions could be readily handled in the same array, separating them has the advantage of simplifying the various video display routines for the game screens. Besides, the C-128 has oodles of extra memory anyway.)

The variable E\$ defines the <ESC> X key sequence for swapping display screens.

Lines 110 to 140 print the opening display on each screen. The illusion of simultaneous movement on both screens is created by switching between the displays and advancing one character position each time.

Lines 150 to 230 are used to set up the "hide target" screen on each display. This time, for variation, the GRAPHIC command is used to switch back and forth. The CHAR statement is used extensively for creating the screen images. This BASIC 7.0 feature allows you to print a character or character string at a given location on the 40 or 80 column screen. (In 40 columns, it works with both the text display mode and the various bit map graphics modes.)

Lines 240 to 380 contain the routines for moving the "cursor" and placing the targets. The flashing "cursor" is created by the combination of the CHAR command in line 250 writing a character in reverse video followed by the CHAR in line 290 writing the same character to the same location in normal video mode. A suitable delay for the flashing is created by the statements in between the two CHAR's.

The joystick is read and interpreted in lines 260 to 280 using the JOY(N) function. Positional information in terms of screen row (R) and column (C) is calculated in line 250 from the logical position variables X and Y (also used as indices for arrays P() and later H()), according to the current screen size (40 or 80 columns).

Lines 330 to 340 contain routines for keeping the cursor within the bounds of the 10 by 10 grid.

Lines 390 and 400 set up the 40 and 80 column displays for the main game.

Lines 410 to 550 are the main game routines for moving the cursor, firing a bomb, and detecting a hit or a miss. Line 490 updates the status grid on your opponent's display. Joystick control and other elements are similar to those used in the target hiding routines.

Lines 560 to 620 contain the sound effects and end-of-game routines. The sound effects use BASIC 7.0's advanced SOUND and PLAY commands for easy programming of the SID chip.

Although BOMBS-AWAY is a rather trivial use of dual screen video, it does serve to demonstrate many of the advanced features of the C-128 and BASIC 7.0.

See Bombs-Away listing on Page 17



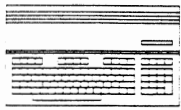
# DUAL VIDEO FUN

Continued from Page 16

```

10 COLOR 0,7:COLOR 4,7:COLOR 5,2:COLOR 1,2:GRAPHICS,1
20 FAST:DIM P(2,10,10),H(2,10,10):E$=CHR$(27)+"X"
30 T$=""
40 R$=""
50 M$=""
60 B$=""
70 T8$=""
80 R8$=""
90 M8$=""
100 B8$=""
110 PRINT"[CLR][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN]"
120 FOR I=1 TO 21:PRINT$ "M$[LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT][LEFT]";NEXT
130 CHAR,40,13,"BY":GRAPHICO:CHAR,18,13,"BY":CHAR,12,15,(M6$)
140 GRAPHICS:CHAR,34,15,(M6$):FOR I=1 TO 3:GOSUB 620:GOSUB 610:NEXT
150 GRAPHICS,1:GRAPHICO,1:PRINT"[DOWN]"T$:GRAPHICS:PRINT"[DOWN]"T8$
160 FOR I=1 TO 9:PRINTR8$:PRINTM8$:GRAPHICO:PRINTR$:PRINTM$:GRAPHICS:NEXT
170 PRINTR8$:PRINTB8$:GRAPHICO:PRINTR$:PRINTB$:GRAPHICS
180 CHAR,10,0,M8:CHAR,65,21,"PLAYER 1",1:CHAR,65,1,S$:CHAR,66,3,"TARGETS"
190 CHAR,60,15,"HITS >>":CHAR,60,17,"SHOTS >>":GRAPHICO
200 CHAR,1,0,M8:CHAR,30,21,"PLAYER 2",1:CHAR,28,1,S$:CHAR,29,3,"TARGETS"
210 CHAR,25,15,"HITS >>":CHAR,25,17,"SHOTS >>":GRAPHICS
220 FOR I=1 TO 10:FOR J=1 TO 10:CHAR,48 I-2,2+J,">:"
230 GRAPHICO:CHAR,28 I-1,2+J,">:"GRAPHICS:NEXT J,I
240 CHAR,10,24,Y8+S Y8:N=2:I=1:Y=1:NS=0:A=4:B=2:S=75
250 X0=X:Y0=Y:R=2+Y:C=A+I-B:CHAR,C,R,"[RVON]"CHR$(P(N,X,Y)+118)+"[RVOF]"
260 J=JOY(N):IF J AND 128 THEN P(N,X,Y)--(P(N,X,Y)=0):GOTO 310
270 IF J=3 THEN X=X+1:GOSUB 330:ELSE IF J=7 THEN X=X-1:GOSUB 330
280 IF J=5 THEN Y=Y+1:GOSUB 340:ELSE IF J=1 THEN Y=Y-1:GOSUB 340
290 CHAR,X0+A-B,Y0+J,CHR$(P(N,X0,Y0)+118)
300 IF NS<10 THEN S=S:ELSE S=60
310 IF P(N,X,Y) THEN NS=NS+1:CHAR,S-NS,1," ":ELSE NS=NS-1:CHAR,S-NS-1,1,">"
320 GOTO 290
330 IF X=0 THEN X=10:ELSE IF X=11 THEN X=1
340 IF Y=0 THEN Y=10:ELSE IF Y=11 THEN Y=1
350 RETURN
360 IF N=2 THEN S=80:ELSE CHAR,10,24,BL$+BL$:GRAPHICO:S=38
370 CHAR,1,24,Y8+S Y8:N=2:I=1:Y=1:NS=0:A=2:B=1:GOTO 250
380 CHAR,1,24,BL$+BL$:GRAPHICS
390 FOR I=1 TO 10:FOR J=1 TO 10:CHAR,48 I-2,2+J,">:"CHAR,64+I,J,CHR$(P(1,I,J)+118)
400 GRAPHICO:CHAR,28 I-1,2+J,">:"CHAR,27+I,J,CHR$(P(2,I,J)+118):GRAPHICS:NEXT J,I
410 X1=1:Y1=1:X2=1:Y2=1:NS(1)=0:NS(2)=0:NH(1)=0:NH(2)=0
420 CHAR,10,24,Y8+F8:N=2:A=4:B=2:S=27:X=X1:Y=Y1
430 X0=X:Y0=Y:R=2+Y:C=A+I-B:CHAR,C,R,"[RVON]"CHR$(H(3-N,X,Y)+118)+"[RVOF]"
440 J=JOY(3-N):IF J=128 THEN A=80
450 IF J=3 THEN X=X+1:GOSUB 330:ELSE IF J=7 THEN X=X-1:GOSUB 330
460 IF J=5 THEN Y=Y+1:GOSUB 340:ELSE IF J=1 THEN Y=Y-1:GOSUB 340
470 CHAR,X0+A-B,Y0+J,CHR$(H(3-N,X0,Y0)+118):GOTO 430
480 GOSUB 620:NS(3-N)=NS(3-N)+1:MS$=STR$(NS(3-N)):CHAR,34+N,17,NS$
490 PRINT$;CHAR,S+X,Y," ":PRINT$;:IF P(N,X,Y)=1 THEN P(N,X,Y)=0:GOTO 540
500 IF H(3-N,X,Y)<>1 THEN NH(3-N,X,Y)=42
510 IF N=1 THEN X2=X:Y2=Y:CHAR,1,24,BL$+BL$:GRAPHICS:GOTO 420
520 CHAR,10,24,BL$+BL$:GRAPHICO:CHAR,1,24,Y8+F8:X1=X:Y1=Y
530 N=1:A=2:B=1:S=64:X=X2:Y=Y2:GOTO 430
540 H(3-N,X,Y)=1:NH(3-N)=NH(3-N)+1:GOSUB 610
550 NH$=STR$(NH(3-N)):CHAR,34+N,15,NH$:IF NH(3-N)<10 THEN A=30
560 SLEEP 1:PLAY"V104IFGACSC.BC1DC04A.GF16AAGF"
570 PRINT"[CLR][DOWN][DOWN][DOWN][DOWN]"
580 PLAY".HGIFGACSC.BC1DC04A.GF16AAGGMF":POKE 209,0
590 INPUT"[DOWN][DOWN][DOWN]PLAY ANOTHER GAME <Y/N>";A$:IF A$="Y" THEN RUN
600 END
610 VOL15:SOUND 2,8192,45,1,1,150,3,0:RETURN:REM EXPLOSION
620 VOL15:SOUND 2,33333,100,1,1000,200,1:RETURN:REM BOMB DROP

```



## REU FREE ACCESS

by Fred Bowen

Anyone familiar with disk files knows about sequential (SEQ) or program (PRG) file types, how they are opened and how their data accessed. Folks are also familiar with the obvious advantages and disadvantages of these modes:

They:

- Provide easy access to individual bytes, but...
- Cannot access data already read or written (cannot "rewind" the file).
- Must be re-opened to change access mode (read file vs. write file).

RELative file mode is designed to get around these limitations and to do so with relative (pun intended) speed. It does so by imposing a data structure which must be established when the file is created; the user must define a record with a specific length (less than 255 characters) and specific data fields (reserving specific characters for field delimiters).

Wouldn't it be nice to have the capability to move the file pointer about a SEQuential or PRoGram file, forwards or backwards? And wouldn't it be nice to either read \*or\* write data on the fly? And wouldn't it be nice to use any character in any way you wanted? You need look no farther than the Commodore RAMdisk utility, version 4.0 or higher. There exists an undocumented file access mode in the RAMdisk which provides this capability.

It's called 'F' access mode, or 'Free' access. Simply open a new or old SEQuential (type = S) or PRoGram (type = P) RAMdisk file for Free access like this:

```
OPEN CHANNEL, UNIT, SA, "FILENAME,TYPE,F"
```

The file will be created if it does not already exist, so be sure to use the correct filename. You set the file pointer to anywhere in the file (even beyond the end of the file) much as you would access RELative file data, using the 'P'osition DOS command:

```
OPEN CHANNEL, UNIT, 15  
PRINT#15, "P"; CHR$(SA); CHR$(PAGE-LOW); CHR$(PAGE-HIGH); CHR$(BYTE)
```

Instead of record numbers, you need to know the address of the byte in the file. The first byte is page 0 byte 0, and this is where the file pointer is placed by default when the file is opened. The format of the data is totally free-your program can impose any structure desired. As data is either read (via GET# or INPUT#) or written (via PRINT#) the file pointer is automatically advanced to the next byte in the file to be read or written, just like the cursor on the display screen is advanced to the next character position when you type. The (byte) parameter is optional, and if omitted defaults to byte 0. If the file pointer is moved past the end of the file, the file is automatically expanded and the new area filled with \$FF (CHR\$(255)). As with RELative files, when you create a new file using Free mode you should position the file pointer to the last byte you expect to use and write something there (such as the CHR\$(255) pad byte).

I do not recommend C128 programmers use the BASIC 7.0 RECORD# command in place of the DOS 'P' command shown in the above example. The RECORD# command was added specifically for RELative files, and the BASIC parser limits the (byte) parameter to the range (1-254). The values 0 and 255 result in "? ILLEGAL QUANTITY ERROR", which is unfortunate.

So what is "Free" access good for? Clearly it is not without disadvantages- a program must keep track of its location within a file, and it is not supported by disk drives. "Free" access would appear to have some application for word processors, editors, and databases, and it would

seem to provide an ideal method for "patching" program files. "Free" access does not require side sectors like RELative files do, so it provides some space savings in that regard. Why was "Free" access built in? Because the mechanisms to implement it were already there, it was easy to add this capability. Also, a particular producer of third-party software had asked for it, but the request was never followed up. Why was "Free" access never documented? Because, while it is a neat idea, it had no immediate application and was destined to live alone, trapped within the REU RAMdisk driver.

Use it advisedly- "Free" access was never heavily tested, and it was not reviewed by the QA (quality assurance) people. Like RELative file operations on the RAMdisk, Free mode is not very fast. I am working on a neat hack that promises to help out in that respect- a way to make the RAMdisk controller stay resident, eliminating the time consuming swapping that occurs every time the controller is needed (for every sequential byte read, nearly 16k bytes of data are moved!). Cognito ergo equus sum.

## Herne Data Systems Ltd.

Featuring C-128 CP/M software by:  
Miklos Garamszeghy

**Jugg'ler-128** Now shipping version 3.4.

Let your C-128 CP/M system read, write and format over 140 types of CP/M disks with this easy to use utility. Only \$19.95\*

**QDisk** Now shipping version 2.1.

Now you can use your Quick Brown Box as a non-volatile RAM disk in CP/M mode with this memory resident driver. Only \$11.95\*

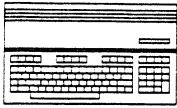
**Scramb'ler-128** Now shipping version 2.0

High security data encryption system for C-128 CP/M. Protect all of your vital and confidential data from un-authorized snooping. Only \$19.95\*

Herne Data Systems Ltd.  
P.O. Box 714, Station 'C'  
Toronto, Ont. M6J 3S1  
(416) 535-9335

\*US \$, including postage & handling

**Software worth hunting for!**



## DISK DRIVE PROGRAMMING

by Mike Gilsdorf

Have you ever wished you didn't have to load those DOS utility programs into your computer everytime you wanted to change a file type, lock/unlock files, append BASIC programs, etc.? Wish you could add new commands to your disk drive and execute them in the same way you rename and scratch files? How would you like to unscratch a file with PRINT#15, "UNSCRATCH:Filename"? Well, if you thought that these things were only possible if you replaced the ROM, used a cartridge or utility program, then surprise!

As most of you are probably aware, Commodore disk drives like the 1541, 1571, and 1581 are smart devices. Each have their own RAM, ROM, CPU, and I/O chips. They can be programmed to perform tasks in the same manner your computer does. More importantly, these tasks can be done without loading or disturbing a current program in your computer's memory. Its sad to see so many utility programs written for the computer when in fact many are better suited for the drive. Generally speaking, drive programs are faster, shorter, and more efficient, and they don't need to be loaded over the slow serial bus.

### WHAT IS A DRIVE PROGRAM?

Drive programs are loaded and run in your disk drive in much the same way utility programs are in your computer. After receiving the appropriate computer command, the drive loads the ML program off disk and stores it in its memory (RAM). Depending upon the command sent, the drive either immediately executes the program, or waits until it receives another command such as M-E or Un (where n is a number from 3 to 9). Programs which auto-execute in the drive on command are referred to as "Utility Loaders". Those which auto-execute when the drive is reset or initialized (as in the case of the 1581) are called "Auto Boot Loaders". Both types of loaders are made up of one or more packets (groups of bytes) which hold the ML program. Each packet is constructed as follows:

BYTE 0	LOW BYTE LOAD ADDRESS
BYTE 1	HIGH BYTE LOAD ADDRESS
BYTE 2	NUMBER OF PROGRAM/DATA BYTES
BYTES 3 TO X	PROGRM DATA BYTES
BYTE X+1	CHECKSUM

Notice that since byte #2 can not be larger than 255, the maximum length of a packet is 259 bytes (2 address bytes, 1 byte counter, 255 program/data bytes, and 1 checksum byte). Notice too, that this limits X to a maximum value of 257. You can combine or APPEND packets to construct longer files. Bytes 0 and 1 in the first packet also serve as the address to begin executing once the entire program file is loaded. Depending upon its load address the next packet is stored in the drive's memory usually (not always) immediately following the last byte of the previous packet. The checksum byte is calculated by adding bytes 0 to X and adding a 1 for every carry (over 255) that results.

Utility loaders must be named as USR type files. You can call the program file any legal name you wish, except if your using a 1541. Due to a bug in the 41 drives, filenames for utility loaders had to begin with the "&" character. Whether this is still true for the newer 1541 IIs, I'm not sure. Personally, I like to name all my utility loaders beginning with the "&" character anyway since it makes it easier to distinguish them from my other files on disk. Also, if the program is written to run only on a particular type drive (which is true for most utility loaders) I'd recommend you include the name of the drive in the filename (e.g., &filename.41 for the 1541). Drive programs are executed by sending the following command:

```
OPEN15,8,15,"&O:FILENAME": CLOSE 15
```

If you're using the DOS wedge, the command is sent as follows:

```
&&O:FILENAME
```

If you're using a 1541 the commands are:

```
OPEN1,8,15,"&FILENAME": CLOSE1
```

or:

```
&&FILENAME
```

Note: Many application programs (such as word processors) will allow you to send commands directly to your drive without having to exit the program. This is one of the nice advantages of utility loaders and Un commands!

### WHERE TO LOAD:

Although there are products available which allow you to add extra RAM to your drive (e.g., RAMBOard), most of us are limited to the amount of memory initially installed.

1541/71	2K	\$0000-\$07FF
1581	8K	\$0000-\$1FFF

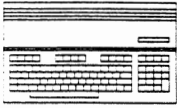
Unfortunately, not all of this is available to us for programs. Some is reserved for DOS variables, BAM, work area, etc. Programs written for the drive generally load into one (or more) of the drive's 256-byte data buffers.

BUFFERS		1541/71	1581	
BUF0	\$0300-\$03FF	X	X	
BUF1	\$0400-\$04FF	X	X	
BUF2	\$0500-\$05FF	X	X	(UN COMMANDS JUMP HERE)
BUF3	\$0600-\$06FF	X	X	
BUF4	\$0700-\$07FF	BAM	X	
BUF5	\$0800-\$08FF	N/A	X	
BUF6	\$0900-\$09FF	N/A	X	
BUF7	\$0A00-\$0AFF	N/A	BAM	
BUF8	\$0B00-\$0BFF	N/A	BAM	
BUF9	\$0C00-\$1FFF	N/A	TRK	

Do NOT load/execute drive programs into the BAM or TRK buffers or when there are open active disk files. If you do, you run the risk of corrupting your disk making it unrecoverable. I usually prefer having my utility loaders load into BUF2 since it offers the option to execute them again on the same or different disk using the Un commands.

To show you how all this works, let's write a simple ML program for the drive which will turn the drive's activity light (LED) on, then off. We'll design the program to load and execute in BUF2, and we'll also make use of the U3 and U4 commands. Note that these commands are sent over the drive's command (error) channel #15, and work similar to sending an M-E command to addresses \$0500 and \$0503. (See your drive's user manual for other Un commands and their jump addresses.) The DOS routines that turn the LED on and off are called SETLDS and ERROFF. For the 1541/71 they are located at \$C100 and \$C123. Simple enough? Well... there is a slight quirk. If we RTS immediately after turning the LED on, the drive will immediately turn it back off again once it returns to its normal housekeeping chores. If we didn't have a way to keep the LED on, we might only see a very fast flash - perhaps too fast to be seen. One way to overcome this is not to exit our drive program after LED turn-on, but continue looping until a new command has been detected. Location \$7C (ATNPND) on the 1541/71 is used as a

Continued on Page 20



## DISK DRIVE PROGRAMMING

Continued from Page 19

flag to signal the presence of a new command. It holds \$01 if a new command has been received, otherwise it holds \$00. This location is checked/updated with every interrupt. Here's how our program will appear once it is loaded into the 1541/71 drive.

```

$0500 JMP $0507 ; U3 ENTRY POINT
$0503 JSR $C123 ; ERROFF - TURN OFF LED (ALSO U4
                ; ENTRY POINT)

$0506 RTS
$0507 JSR $C100 ; SETLDS - TURN ON LED
$050A LDA #$00 ; CLEAR THE
$050C STA $7C ; ATTENTION PENDING (ATNPND) FLAG
$050E CMP $7C ; CHECK THE ATNPND FLAG FOR THE
                ; COMMAND
$0510 BEQ $050E ; JUMP IF NO NEW COMMAD - KEEP
                ; LOOPING
$0512 RTS ; COMMAND IS WAITING - EXIT

```

For the 1581 drive the program looks similiar.

```

$0500 JMP $0507 ; U3 ENTRY POINT
$0503 JSR $81E5 ; ERROFF - TURN OFF LED (ALSO U4
                ; ENTRY POINT)

$0506 RTS
$0507 JSR $81F6 ; SETLDS - TURN OFF LED
$050A LDA $76 ; GET PRESENT VALUE OF FLAG
$050C CMP $76 ; SEE IF FLAG CHANGED
$0510 BEQ $050C ; NO CHANGE SO KEEP CHECKING
$0512 RTS ; COMMAND IS WAITING - EXIT

```

Now enter the BASIC program below. This program creates a utility loader on disk (device #8) by assembling our ML program into a single packet. Be sure to insert a disk in the drive before RUNNING it. As a precaution use a newly formatted disk.

```

10 REM LED - BY MIKE GILSDORF (C) 1988
20 DV=8: OPEN 2,DV,2,"O:&LED,U,W": REM DV IS THE DEVICE NUMBER
30 READ A: IF A<>256 THEN PRINT#2,CHR$(A): GOTO 30
40 CLOSE 2
50 REM UTILITY LOADER FOR BUF2
100 DATA 0 :REM BUF2 LO BYTE ADDRESS
110 DATA 5 :REM BUF2 HI BYTE ADDRESS
120 DATA 19 :REM NUMBER OF PROGRAM/DATA BYTES
130 DATA 76,7,5 :REM JMP $0507
140 DATA 32,35,193 :REM JSR $C123
150 DATA 96 :REM RTS
160 DATA 32,0,193 :REM JSR $C100
170 DATA 169,0 :REM LDA #$00
180 DATA 133,124 :REM STA $7C
190 DATA 197,124 :REM CMP $7C
200 DATA 240,252 :REM BEQ $050E
210 DATA 96 :REM RTS
220 DATA 243 :REM CHECKSUM BYTE
230 DATA 256 :REM END-OF-DATA MARKER

```

NOTE: If your using a 1581 verify the device number in line 20 is correct, and replace the following program lines:

```

120 DATA 17 :REM NUMBER OF PROGRAM/DATA BYTES
140 DATA 32,229,129 :REM JSR $81E5
160 DATA 32,246,129 :REM JSR $81F6
170 DATA 165,118 :REM LDA $76
180 REM DELETE
190 DATA 197,118 :REM CMP $76
200 DATA 240,252 :REM BEQ $050C
220 DATA 149 :REM CHECKSUM BYTE

```

After RUNNING the BASIC program, if everything went correctly, you should now have a utility loader stored on disk. Here's how it should appear in the directory:

```
1 "&LED" USR
```

We're ready now to have the drive load and execute the utility loader. In direct mode type:

```
OPEN15,DV,15,"&O:&LED"
```

For the 1541 use:

```
OPEN15,DV,15,"&LED"
```

You should see the activity light on. To turn it back off, type:

```
PRINT#15,"U4"
```

To turn it back on again without reloading, remove the disk and type:

```
PRINT#15,"U3"
```

Type:

```
PRINT#15,"U4"
```

to turn it back off. Or as an alternative, to see the drive turn the LED off as part of its housekeeping duties, type:

```
CLOSE15
```

If these commands didn't work properly, recheck your typing, and be sure the DATA statements are entered correctly. Don't overlook the semicolon appearing in line 30 after the PRINT # command. In a future issue we'll show you how to use more DOS ROM routines, and present an easier way to construct utility loaders. Until then, easy DOS it!

(Mike Gilsdorf can be reached on Q-Link under the handle MIKE ALL).

### Faster than a Speeding Cartridge More Powerful than a Turbo ROM

*It's Fast, It's Compatible, It's Complete, It's...*

# JiffyDOS™

Ultra-Fast Disk Operating System for the C-64, SX-64 & C-128

- Speeds up all disk operations. Load, Save, Format, Scratch, Validate, access PRG, SEQ, REL, & USR files up to 15 times faster!
- Uses no ports, memory, or extra cabling. The JiffyDOS ROMs upgrade your computer and drive(s) internally for maximum speed and compatibility.
- Guaranteed 100% compatible with all software and hardware. JiffyDOS speeds up the loading and internal file-access operation of virtually all commercial software.
- Built-in DOS Wedge plus 14 additional commands and convenience features including one-key load/save/scratch, directory menu and screen dump.
- Easy do-it-yourself installation. No electronics experience or special tools required. Illustrated step-by-step instructions included.

Available for C-64, 64C, SX-64, C-128 & C-128D (JiffyDOS/128 speeds up both 64 and 128 modes) and 1541, 1541C, 1541-II, 1571, 1581, FSD-1&2, MSD SD-1&2, Excel 2001, Enhancer 2000, Amtech, Swan, Indus & Bluechip disk drives. System includes ROMs for computer and 1 disk drive, stock/JiffyDOS switching system, illustrated installation instructions, User's Manual and Money-Back Guarantee.

C-64/SX-64 systems \$59.95; C-128/C-128D systems \$69.95; Add'l drive ROM's \$29.95

Please add \$4.25 shipping/handling per order, plus \$2.50 for AK, HI, APO, FPO, Canada & Puerto Rico. \$10.00 add'l for other overseas orders. MA residents add 5% sales tax. VISA/MC, COD, Check, Money Order. Allow 2 weeks for personal checks. Call or write for more information. Dealer, Distributor & UG pricing available.

Please specify computer and drive when ordering

### Creative Micro Designs, Inc.

P.O. Box 789, Wilbraham, MA 01095 Phone: (413) 525-0023  
50 Industrial Dr., Box 646, E. Longmeadow, MA 01028 FAX: (413) 525-0147

# BRIWALL

Our Fourth Year  
In Business!  
We Know How  
To Service  
Our Customers!

## SOLID PRODUCTS & SOLID SUPPORT

P.O. Box 129 / 58 Noble Street  
Kutztown, PA 19530

24 HOURS — TOLL FREE

1-800-638-5757

OUTSIDE USA CALL 1-215-683-5433

SATISFACTION  
GUARANTEED

No Surcharge On  
Charge Orders

Friendly Service

Customer Service

Business Hours

M-F, 9-5 EST

### ACCESSORIES

40/80 Column Switch Cable	18
Apro Ext.64-Crt. Port Ribbon	17
Apro Ext.64-User Port Ribbon	19
Aprospan 4 Slot Cartrdg Holder	25
Cover 1541	7
Cover 1571	9
Cover C128	9
Cover C64	8
Drive Box	27
Final Cartridge V3	49
Hotshot Plus Interface	69
Joystick-Bat Handle	22
Joystick-Ergo Stick	22
Joystick-Super 3 Way	25
Joystick Y Adapter	7
Leroy's Cheat Sheets C64 each	3
Leroy's Cheat Sheets C128 each	7
Lightpen-Model 170c	69
Lightpen-Model 184c	49
M-3 Mouse 64/128-Proport.	39
Mach 128 Cartridge	35
Mousepad	8
Paper Banner Colors each	10
Power Supply-C128-Repairable	55
Power Supply-C64-Repairable	39
Quick Brown Box 32K	79
Quick Brown Box 64K	99
RGB to RCA Cable	9
RS232 Deluxe Interface	39
Ribbons for Printer	Call
Ribbon Renew Re-ink w/no eqp.	5
Serial Box 2 for 1	29
Serial to Serial Cable-6 ft.	9
Super Chips 128	45
Super Chips 128-D	45
Super Chips 64	25
Super Chips 64 mode on 128	25
Super Graphix Gold Pmtr Intrc	89
Warspeed 128	35

### BOOKS

1541 Troubleshoot & Repair Guide	17
1571 Internals*	17
Anatomy of C64*	17
Anatomy of the 1541 Book	17
Basic 7.0 Internals Book	19
Basic 7.0 for C128 Book	10
Beginner's Guide to BASIC 8	15
C128 Assembly Language Prog Book	14
C128 Internals*	17
C128 Programmer's Ref. Book	20
C128 Tips & Tricks Book	17
C128 Troubleshooting & Repair Book	17
C64 Basic Programming	25
C64 Programmer's Reference Guide	17
C64 Tips & Tricks*	17
C64 Troubleshoot & Repair Guide	17
C64/128 Asmbly Lang. Prog Book	14
GEOS Get Most Book	9
GEOS Inside & Out*	17
GEOS Programmer's Ref. Guide	17
GEOS Tips & Tricks*	15
Hot to Program in BASIC	24
I Speak Basic to My C64 Book	8
K Jax Book Revealed 1	23
K Jax Book Revealed 2	23
Machine Language 64 Book	13
Superbase The Book 64/128	15
Troubleshoot & Repair C64 Book	18
Twin Cities 128 Compendium 1	15
Abacus Disks each*	12
Bankstreet Writer	35
Business Form Shop/64	30
CMS Accounting/128	129
CMS Inventory Module 128	53

Chartpak 128	29
Chartpak 64	29
Datamanager 128	39
Datamanager 2 + /64	17
Fleet System 2 w/64	39
Fleet System 4	52
Fontmaster 128 w/Speller	39
Fontmaster II/64	35
KFS Accountant 128	109
Leroy's Label Maker	24
Microlawyer/64	39
Paperclip 3	39
Paperclip Publisher	39
Partner 128	39
Partner 64	30
Personal Portfolio Manager 128	42
Personal Portfolio Manager 64	29
Pocket Dictionary 64/128	10
Pocket Filer 2	29
Pocket Planner 2	29
Pocket Superpak 2	67
Pocket Writer 2	39
Security Analyst 128	35
Superbase 128	39
Superbase 64	29
Sprbase/Sprscprt/Book 128 PAK	69
Sprbase/Sprscprt/Book 64 PAK	59
Superscript 128	29
Superscript C64	29
Swiftcalc 128 w/Sideways	39

Kindercomp	18
Linkword: German	19
Linkword: French	19
Linkword: French 2	19
Linkword: Italian	19
Linkword: Russian	19
Linkword: Spanish	19
Little Computer People	12
Mathbusters	6
Mavis Beacon Teaches Typing	30
Peter & Wolf Music	19
Peter Rabbit (Reading 1)	19
Protutor Accounting 128	69
R.S.V.P.	21
Sky Travel	35
Stickybear ABC's	22
Stickybear Math 1	22
Stickybear Math 2	22
Stickybear Numbers	22
Stickybear Opposites	22
Stickybear Reading 1	22
Stickybear Reading Comprehension	22
Stickybear Shapes	22
Stickybear Spellgrabber	22
Stickybear Townbuilder	22
Stickybear Typing	22
Success w/Math Series each	23
Toy Shop 64	22
WhereinEurope is Carmen Sandiego	29
Where in USA is Carmen Sandiego	29

Cadpak/64	29
Certificate Maker	17
Certificate Maker-Library	12
Coloriz 128	12
Computer Eyes	105
Create a Calendar	22
Doodle/64	29
Flexidraw 5.5/64	29
Flexifont/64	24
Graphics Art Disk 7-12 each	9
Graphics Galleria Pak 1	24
Graphics Galleria Pak 2	24
Graphics Intergrator 2/64	24
Home Designer	45
Home Designer/Circuit Symbol Lib	10
Icon Factory/64	25
Label Wizard	21
NewsMaker 128	24
Newsroom	17
Newsroom Clip Art Disk 1-3 each	13
PC Board Maker 64	79
Perspectives II/64(3D)	39
Photo Finish	19
Postcards	19
Printmaster-Fantasy Art Gallery	15
Printmaster Gallery I/64	15
Printmaster Gallery II/64	15
Printmaster Plus/64	25
Printshop/64	35
Printshop Companion	27

Assembler/Monitor/64	29
Basic 8	25
Basic Compiler/128	42
Basic Compiler/64	29
Big Blue Reader 64/128	32
Bobstern Pro/128	42
Bobstern Pro/64	35
CP/M Kit	22
Cobol/128	29
Cobol/64	29
Gnome Kit 64/128	29
Gnome Speed Compiler 128	29
Kracker Jax Super Cat	22
Kracker Jax-Hacker's Utilities	17
Merlin 128	45
Merlin 64	35
Oxford Pascal 128	29
Oxford Pascal 64	25
Physical Exam 1541	29
Physical Exam 1571	29
Power Assembler 64/128	30
Power C 64/128	30
Programmers Toolbox/64	15
Protolinc BBS 128	19
Prototerm 128	12
RamDOS 128	29
RenegadeCopier V2	29
Super 64 Librarian	19
Super 81 Utilities for C128	29
Super 81 Utilities for C64	29
Super Aide 64	25
Super C 128	45
Super C 64	45
Super Disk Librarian for C128	19
Super Disk Utilities 128	25
Super Pascal 128	45
Super Pascal 64	45
Super Snapshot 4 w/C128 Disable	65
Super Snapshot V4	59
SysRES Enhanced	30
Z 3 Plus	59

### OUR POLICY

Our policy is to stock what we advertise and carry the best products available for your C64 and C128 computers. Over 400 of the best productivity, educational & technical software and a host of accessories in stock now! And CHECK OUT OUR C128 LINE! Your will be amazed at the number of products that we carry for this fantastic machine!

Swiftcalc 64 w/Sideways	17
Sylvia Porter's Financial Plan 128	39
Technical Analysis System 128	42
Technical Analysis System 64	29
Timeworks Account Payable 64	39
Timeworks Accounts Receivable 64	39
Timeworks General Ledger 64	39
Timeworks Inventory Managemnt 64	39
Timeworks Payroll 64	39
Timeworks Sales Analysis 64	39
Vizistar 128	60
Vizivrite 128	60
Wordpro 128 w/Speller w/Filepro	30
Wordpro 64 w/Speller w/Turbo Load	30
Wordwriter 128	39
Wordwriter 3 for C64	30
Write Stuff 64	19
Write Stuff 64 w/Talk	24
Write Stuff C128 Version	24

Alphabet Zoo	18
Alphabuild	6
Calculus by Numeric Methods	27
Counting Parade	6
Designasaurus	6
Early Learning Friends	6
Easy Sign	17
Evening Wood Dynamic Reader	17
Facemaker	16
First Men on Moon Math	19
Grandma's House	6
Hayden SAT Preparation	32
Jungle Book (Reading 2)	19
Kids on Keys	18
Kidwriter	22

Where in World is Carmen Sandiego	29
Widham Classics-Alice/Wonderland	32
Widham Classics-Below the Root	12
Widham Classics-Swiss Family Robs	12
Widham Classics-Treasure Island	12
Widham Classics-Wizard of OZ	12

Becker Basic for GEOS 64	35
Desk Pak Plus-Geos	22
FontPak Plus-GEOS	22
GEOS 64 v 2.0	39
GEOS Programmer/64	45
GEOS Write Workshop/64	39
GEOS Write Workshop/128	45
Geocalc 128	45
Geocalc/64	35
Geofile 128	45
Geofile/64	39
Geopublish 64	45
Geos/128	45
Geospell 64/128	19
Wordpublisher 64/128 (for GEOS)	28

Animation Station	65
Award Maker Plus/C64	29
Billboard Maker/64	25
Blazing Paddles	25
Bumpersticker Maker	35
Business Card Maker	29
Button & Badge Maker	39
Cad 3D/64	39
Cadpak 128	42

Screen F/X	25
Sketchpad 128 NEW	24
Slideshow Creator	13
Video Title Shop w/Gr Comp 1	25
Video Title Shop s/Gr Comp 2	17

Boston Bartender's Guide	12
Bridge 5.0	22
Cardio Exercise	75
Cardio Exercise & Heartlab Combo	85
Celebrity Cookbook/64	19
Dr. Ruth's Book of Good Sex	22
Dream Machine Analyze/64	19
Family Tree 128	39
Family Tree 64	39
Heartlab	49
MK America Cooks Chinese	12
MK America Cooks Italian	12
MK Great Chefs of PBS Vol 1	12
MK Great Chefs of PBS Vol 2	12
MK Great Chefs of PBS Vol 3	12
Memory Academy 128	15
Micro Kitchen Companion	29
Monopoly C64	25
Muscle Development Package	54
Sexual Edge/64	19
Stress Reduction-Enhanced	299
Stress Reduction-Standard	89
Stnder's Classic 1-10 each	5
Tarot 128	15

1541/1571 Drive Alignment	25
1581 Toolkit	32

### ENTERTAINMENT

AD&D Dungeon Master Assistance	22
AD&D Pool of Radiance	29
Air	12
Battleship	24
Blackjack Academy	29
California Raisins	19
Caveman Ugh*lympics	22
Chessmaster 2001	30
Crossword Magic	39
Double Dragon	27
Faery Tales	19
Jeopardy 2	32
Jordan vs Bird	24
Lord of the Rising Sun	25
Main Frame	25
Monopoly	24
Nerwomancer	29
Ocean Ranger	0
Operation Wolf	9
Platoon	25
Powerplay Hockey	24
Rampage	27
Rocket Ranger	17
Roger Rabbit	25
Scrabble	26
Scraples	29
Skate or Die	22
TV Sports Football	25
Three Stooges	27
Ultima V	12
Wheel of Fortune 2	42
Zach Macrackin	25
Zoom	22

# Home Designer

When the details are important!

## Home Designer CAD 128

Given glowing ratings by every major Commodore magazine, this CAD system outclasses every other CAD program, because of its object-based design. With over 50 powerful commands, 5 drawing layers, superb support of library figures and lazer-quality printouts at ANY scale on your dot matrix printer or plotter, you can create drawings so accurate that a blueprint can be made from them!

Tired of working with poor quality/inaccurate printouts, manipulating little dots on a bit-map, zooming up on detailed work because you can't zoom in close enough? Join the professionals!

**only \$45.00**

mouse or joystick required

NEW! Engineering Library disk available now. \$10.00



## SKETCHPAD 128

Brand new from Free Spirit, Sketchpad 128 fully supports your C128 and takes advantage of its crisp 80 column graphics capabilities. It is packed with all the features of a professional drawing package such as drawing SMOOTH freehand lines, 3D Solids, creating Slideshows, Cut & Paste, Clip, Flip, Enlarge, Shaded Fill Patterns, a variety of Fonts, Air Brush and more! It supports Printshop graphics and is completely compatible with all BASIC 8 files.

Sketchpad 128 unleashes the graphics power of your C128! It supports your 1351 Mouse, 64K Video Chip, 1581 drive and 80 column display. **What more could any real C128 user ask for?**

**ONLY \$24.00**

## SIZZLING HOT ENTERTAINMENT TITLES

AD&D Dungeon Master Assistance	\$22
AD&D Pool of Radiance	29
Battleship	24
Caveman Ugh*lympics	22
Double Dragon	27
Jordan vs Bird	24
Lord of the Rising Sun	25
Monopoly C64	24
Newromancer	29
Operation Wolf	29
Platoon	25
Powerplay Hockey	24
Rampage	27
Rocket Ranger	27
Roger Rabbit	15
Scrabble	26
Scraples	29
Skate or Die	22
TV Sports Football	25
Three Stooges	27
Ultima V	42
Zach Macrackin	25
Zoom	22

Buy any 3 of the above titles and deduct \$5 from your order total! Offer until 3/31/89.

## BRIWALL

SOLID PRODUCTS & SOLID SUPPORT  
PO Box 129/58 Noble Street  
Kutztown, PA 19530

Toll-Free 24 Hour Order Line  
1-800-638-5757

24-Hour Order Line Outside USA  
1-215-683-5433

24-Hour FAX Line - 1-215-683-8567

Customer Service - 1-215-683-5433

Technical Support - 1-215-683-5699

PLEASE COME VISIT US!

Monday - Friday 9AM-6PM EST

Saturday 9AM - 12 Noon EST



Write or Call for our

GREAT AMGA OR C64/C128 CATALOG



HOT NEWS!

## NEWS MAKER 128

Finally somebody (Free Spirit Software) has introduced a desk top publishing program for the 128 in its native mode! Take a few moments to think about what should be included! Would you like the crisp & clear 80 column screen? Would you like to zip across the screen with a 1351 mouse? Could you use a Ram Disk that can store multi-page documents & graphics available at the click of a mouse? Also how about storing up to 45 fonts in the Ram Disk for instant access? How about if you could use Print Shop graphics directly from the Print Shop disk? Also be able to use pictures and graphics created with Sketchpad 128 and Basic 8.0. Would you prefer graphic tools that would allow you to draw custom art directly? How about supporting 2 text modes and 9 font sizes? Pour sequential files into columns? Adjust columns around graphics! or type text directly to the screen?



**ONLY \$24.00**

C128D (or 64K video) and  
80 column monitor

## SCREENFIX

Create incredible presentations using 100,000 different combinations of effects. Create fantastic animated scripts using the most powerful C64 editor ever created.

**OUR PRICE \$25**

## 1541/1571 DRIVE ALIGNMENT

### 1541/1571 Drive Alignment

This excellent alignment program is a must have for every Commodore owner. Easy to use, it helps you to align your drive so that it runs just as if it were new! The simple instructional manual and on-screen help prompt you thru the alignment procedure and help you adjust the speed and head stop of your drives. It even includes instructions on how to load the alignment program when nothing else will load. Don't be caught short! We get more RED LABEL orders for this program, then any other program we sell. Save yourself the expense! Order now, keep it in your library and use the 1541/1571 Drive Alignment program regularly!! Works on the C64, C128, and SX64 for both the 1541 and 1571 drives.

**STILL ONLY \$25.00**

## BASIC 8 IS BACK!!

IN NEW PACKAGING!  
IN NEW SUPPORT!

This popular package adds extensive 80 column graphics capabilities to your C128. A must for C128 programmers! As an added bonus several preprogrammed BASIC 8 applications, such as BASIC PAINT, WRITE and CALC, are included on the flip side!



**JUST \$29**

## ICON FACTORY

Convert, Enlarge, Alter, Enhance, Crop, Invert, Flip, Reduce and even Smooth graphics with this great graphics conversion utility.

**OUR PRICE \$25**

## Photo Finish

"Optimize" your Hi-Res graphics producing hardcopy with FOUR TIMES the resolution of standard printer dumps.

**OUR PRICE \$19**

### SHIPPING POLICY

All ORDERS received before 3PM EST will normally be shipped same or next business day. Out Of Stock items will be shipped by the same shipping method as original order, normally within 3 or 4 business days. All UPS shipments are sent SIGNATURE REQUIRED/NO DRIVER RELEASE. ADD the following shipping charges to your TOTAL software order:

UPS: Ground \$4.00 (cont USA ONLY); Air/Rush \$7.00 (Includes Puerto Rico/Hawaii); Overnight \$12.00 + \$2.50 per item (must be received by 12 NOON)

PRIORITY MAIL: USA \$4.00 (includes APO/FPO); Canada/Mexico \$6.00; Other Foreign \$4.00 Handling and Actual Shipping (minimum \$12.00)

### ORDERING INSTRUCTIONS

For your convenience, you can place a phone order 24 HOURS/7 DAYS A WEEK on our TOLL-FREE LINE 1-800-638-5757. Outside the USA, please use (215) 683-5433. Price, Availability and Compatibility Checks are also welcome on our order line, Monday thru Friday, 9:30AM - 4:30PM EST. AFTER HOURS, Orders Only, Please!! When placing an order, please specify your Computer Model, Home & Daytime Phone Number, Shipping Address, Charge Card Number, Expiration Date, & Name of Bank. To help us serve you better, please have all your information ready before you call us.

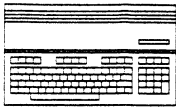
ACCEPTED PAYMENT METHODS: We welcome payment by PREPAID (by Personal Check or Money Order), MASTERCARD; VISA; SCHOOL PURCHASE ORDERS. CDD orders are welcome for established Briwall Customers Only (\$4.00 additional). All payment must be in USA DOLLARS. There is no Surcharge for Credit Cards and your card is NOT charged until we ship!

### RETURN POLICY (for software and accessories ONLY)

We have a liberal return policy to better service your needs. If within 15 days from the time you receive an item, you are not satisfied with it for any reason, you may return it to us for either a refund, exchange or open credit. REFUNDS ARE SUBJECT TO A 10% RESTOCKING FEE PER ITEM RETURNED (\$5.00 MINIMUM PER ITEM). EXCHANGES/OPEN CREDITS will gladly be issued for the Full Purchase Price of the item. DEFECTIVE items, Hardware items and Special Order items will be replaced with the same item only.

### TECHNICAL SUPPORT

We do our very best to help you with your product selections, before you order and after you receive your product. General questions, price, compatibility with your computer, etc. will be handled by our order staff at the time you place your order. BUT, if you have specific, detailed questions about a product, printer, compatibility questions etc., you will get the most help from our TECHNICAL SUPPORT LINE at 215-683-5699. Call Monday thru Friday, 9:30AM-4:30PM EST, and our trained tech staff will be happy to help you.



## FKEY POWER

by Rick Kane

The C-128's programmable function keys can be a handy and welcome feature for power users. It's possible to speed many operations by simply redefining a key or two for your particular needs. I'm sure most of us already have the 'perfect' set of definitions for our tastes.

But there are a couple of areas that the 'key' command doesn't address. What about the HELP and shift-RUN/STOP keys? These are also programmable keys, but the 'key' command will not let you near them. And what about programs that use the function keys in a menu? They expect the chr\$ codes from 133 - 140. If you're testing and debugging such a program, it can be a problem. You have to define the function keys as the chr\$ codes (you probably should at the beginning of such a program anyway), then you have to retrieve your programming-aid definitions each time you test a change you've made. You could write that into the program you're working on, but that clutters things up (I think unnecessarily).

Yes, the 'key' command is powerful and useful, but it simply doesn't go far enough. What it needs is an enhancement that allows you to redefine the HELP and shift-RUN/STOP macros. An enhancement that allows you to toggle between the definitions YOU want and the ones the program wants. What it needs, if you haven't figured it out by now, is 'FKEY.POWER', the subject of this article.

When 'FKEY.POWER' is loaded and run, you will get a list of key command enhancements, followed by 'keys:on'. Any definitions you had entered are intact, and are considered the 'on' keys. If you type 'key' <return>, you will see them (but not the HELP or shift-RUN/STOP macros). Now type 'key off' <return>. The space is not needed, but be sure you start at the beginning of the line, or you will get a syntax error. Then type 'key' <return> again. You will see the graphic symbols for the chr\$ codes 133 - 140. HELP is defined as 'help' <return>, and shift-RUN/STOP as 'run'\*' <return>, and these work as on power-up. Type 'key on' <return>, then 'key' <return>, and your definitions are back!

To redefine the shift-RUN/STOP key type 'key 9,' followed by your definition, the same way you do for keys 1 - 8. Similarly, the HELP key is defined with 'key 10,'. This will work for either the 'on' or 'off' key definitions, but there is one thing to keep in mind. The 'off' keys are stored in an area of 40 bytes. The first ten are key lengths for each key, leaving 30 for definitions. Eight are used by the function keys as defined, leaving 22 bytes for HELP and shift-RUN/STOP. These use 11 as defined, leaving 11 'spare' bytes for changing the 'off' definitions. Any extra will overwrite some part of your 'on' macros (if they are longer than 40 themselves), so be careful if you redefine the 'off' keys. Any changes you do make are swapped into the 'off' buffer with the next 'key on', so they could be saved with the machine language if you prefer a different set of definitions. I'll say more about that later.

The last new command, 'key x', disables the enhanced commands, leaving whichever set was working intact. To re-enable FKEY.POWER, type 'sys3072'. (If you type 'sys3072' when fkey.power is active, it will be disabled. Just type it again.) If 'key off' was issued before 'key x', a 'key on' command issued now will restore your 'on' definitions, if they haven't been overwritten; most likely by a 'boot'. If the keys were left 'on' before 'key x', the current definitions will be the new 'on' keys. FKEY.POWER should be quite transparent, but if you want to test a programming change in a program that is sensitive to it, issue a 'key off' before 'key x', run the test, then type 'sys3072' to re-enable fkeys, and 'key on'. This will bring back the definitions you were using before the test. If you aren't sure whether the keys were off before disabling, you can bring back the working macros by 'poke 3260,0' before

the 'sys 3072'. FKEY.POWER works by intercepting the ERROR vector. On an error, it looks at the input buffer, looking for the 'key' token. If it was a syntax error, it checks for the 'x' character at the point of the error, and disables itself if found. If not, it looks to define key 9 (RUN/STOP) or 10 (HELP). If '9' caused the error, it defines key 9, otherwise it defines key 10. It does so by re-entering the PFKEY routine where it exited to the ERROR routine. 'key on' and 'key off' cause the 'unimplemented command' error. In this case, FKEY.POWER checks for the tokens for 'on' or 'off'. If 'off' caused the error (and the keys are on), it stores the first 40 bytes at \$1000, the current definitions, to KEYON (\$0bd8 - \$0bff), and installs the chr\$ code definitions from KEYOFF (\$0cb8 - \$0cff) to \$1000. If 'on' caused it, and the keys were already off, it swaps out the 'off' keys to KEYOFF and installs KEYON back to \$1000, reuniting those 40 bytes with the 'on' definitions. The length bytes reflect the length of these definitions, and 40 captured bytes fit back into place.

If you want, you can save the machine language as a bootable file, one which can be activated without disturbing Basic with 'boot "name"'. This also allows you to save any changes you might make to the 'off' definitions, just type:

```
KEY ON <RETURN>
BSAVE"NAME",P(3072)TO P(3328),80
```

If your 'on' definitions are 40 bytes total or less, you can save them with the machine language with the following:

```
KEY OFF <RETURN>
POKE3029,76:POKE3030,0:POKE3031,12 <RETURN>
BSAVE"NAME",P(3029)TO P(3328),80 <RETURN>.
```

The pokes above create a 'jmp \$0c00' at the start. The file created will then boot normally, and your 'on' keys will be available immediately.

See listings on Page 24

Twin Cities 128  
The Commodore 128  
Journal P.O. BOX 11578  
MINNEAPOLIS MN 55411

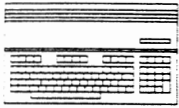


Find out why we are  
**growing** while the  
others are **shrinking!**

Six issues \$12.50  
Twelve issues \$25

---

Twin Cities 128 Compendium  
Book #1 the best of issues 1-18  
\$76.95



# FKEY POWER

Continued from Page 23

\* FKEY.POWER 2.1 \*  
\* MERLIN SOURCE CODE \*

TXTPNTR = \$3D  
BUF = \$0200  
IERROR = \$0300  
PKYBUF = \$1000  
PFKENTRY= \$60EC  
PRIMM = \$FF7D

ORG \$0C00  
  
JSR PRIMM  
HEX 0D  
TXT '<C>89 R.KANE'  
HEX 0D  
TXT 'KEY9/10/ON/OFF/X'  
HEX 0D00  
TOGGLE JSR PRIMM  
TXT 'FKEYS:'  
HEX 00

LDX IERROR  
LDY IERROR+1  
CPX #<TRAPERR  
BNE INSTALL  
CPY #>TRAPERR  
BNE INSTALL  
LDX ERRVEC  
STX IERROR  
LDX ERRVEC+1  
STX IERROR+1  
JSR PRIMM  
TXT 'OFF'  
HEX 00  
LDX #\$FF  
BNE DONE  
INSTALL STX ERRVEC  
STY ERRVEC+1  
LDX #<TRAPERR  
STX IERROR  
LDX #>TRAPERR  
STX IERROR+1  
JSR KEYON  
JSR PRIMM  
TXT 'ON'  
HEX 00  
DONE RTS

TRAPERR JSR NEWERROR  
HEX 4C ; JMP  
ERRVEC HEX 3F4D

NEWERRORLDA BUF  
CMP #\$F9 ; 'KEY' TOKEN  
BNE DONE  
CPX #\$0E ; SYNTAX ERROR  
BNE KEYTOGL  
LDX #\$09  
LDY TXTPNTR  
LDA BUF-1,Y  
CMP #'X'  
BEQ TOGGLE  
CMP #'9'  
BNE DEFINE  
DEX

DEFINE JSR PFKENTRY ; CONTINUE KEY DEFINITION  
LDX #\$FF ; X>127 = NO ERROR  
RTS

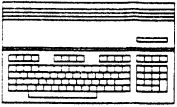
KEYTOGL CPX #\$28 ; UNIMPL. COMMAND ERROR  
BNE DONE  
LDY TXTPNTR  
LDA BUF,Y  
CMP #\$24 ; 2ND BYTE OF 'OFF' TOKEN  
BNE KEYON  
KEYOFF LDX ONFLAG  
BEQ OFFDONE  
LDX #\$D00-KEYOFFS ; # BYTES IN KEYOFFS  
OFFLOOP LDA PKYBUF-1,X  
STA KEYONS-1,X  
LDA KEYOFFS-1,X  
STA PKYBUF-1,X  
DEX  
BNE OFFLOOP  
STX ONFLAG  
OFFDONE DEX ; NO ERROR MESSAGE  
RTS  
  
ONFLAG HEX FF ; FF =ON TO START  
  
KEYON LDX ONFLAG  
BNE ONDONE  
LDX #\$0D00-KEYOFFS  
ONLOOP LDA PKYBUF-1,X  
STA KEYOFFS-1,X  
LDA KEYONS-1,X  
STA PKYBUF-1,X  
DEX  
BNE ONLOOP  
DEX  
ONDONE STX ONFLAG  
RTS  
  
KEYOFFS HEX 01010101010101010605  
HEX 8589868A ; FUNCTION KEYS 1-4  
HEX 878B888C ; FUNCTION KEYS 5-8  
HEX 52D522A0D ; 'RUN'\*<RETURN>  
HEX 48454C500D ; 'HELP'<RETURN>

KEYONS = KEYOFFS-\$100

BASIC LOADER FOR FKEY.POWER

10 REM FKEY.POWER V2.1  
20 REM <C> 1989 RICK KANE  
30 TRAP 60  
40 RESTORE:I=DEC("0C00"):X=0  
50 DO:READ A\$:POKEI+X,DEC(A\$):X=X+1:LOOP  
60 SYS I  
70 END  
100 DATA 20,7D,FF,0D,3C,43,3E,38,39,20,52,2E,4B,41,4E,45  
110 DATA 0D,4B,45,59,39,2F,31,30,2F,4F,4E,2F,4F,46,46,2F  
120 DATA 58,0D,00,20,7D,FF,46,4B,45,59,53,3A,00,AE,00,03  
130 DATA AC,01,03,E0,6D,D0,1B,C0,0C,D0,17,AE,71,0C,8E,00  
140 DATA 03,AE,72,0C,8E,01,03,20,7D,FF,4F,46,46,00,A2,FF  
150 DATA D0,1A,8E,71,0C,8C,72,0C,A2,6D,8E,00,03,A2,0C,8E  
160 DATA 01,03,20,BD,0C,20,7D,FF,4F,4E,20,60,20,73,0C  
170 DATA 4C,3F,4D,AD,00,02,C9,F9,D0,F2,E0,0E,D0,16,A2,09  
180 DATA A4,3D,B9,FF,01,C9,58,F0,9A,C9,39,D0,01,CA,20,EC  
190 DATA 60,A2,FF,60,E0,28,D0,D4,A4,3D,B9,00,02,C9,24,D0  
200 DATA 1C,AE,BC,0C,F0,14,A2,28,BD,FF,0F,9D,D7,0B,BD,D7  
210 DATA 0C,9D,FF,0F,CA,D0,F1,8E,BC,0C,CA,60,FF,AE,BC,0C  
220 DATA D0,12,A2,28,BD,FF,0F,9D,D7,0C,BD,D7,0B,9D,FF,0F  
230 DATA CA,D0,F1,CA,8E,BC,0C,60,01,01,01,01,01,01,01  
240 DATA 06,05,85,89,86,8A,87,8B,88,8C,52,55,4E,22,2A,0D  
250 DATA 48,45,4C,50,0D





## Sparrow's Slick Tips

by Sparrow James?

After a few issues on hiatus, the tricky bird returns with a few slick ones...

**CLIP...CLIP...PAPERCLIP...** One of the nicest things about C-128 programs which are not copy protected is the ease at which most of them can be adapted to alternative mass storage situations such as the RAMDOS and probably more importantly, the 1581 disk drive. One such program is Paperclip III from Electronic Arts. What is particularly attractive about this prospect is the fact that by virtue of the 1581's large storage capacity you can place all of Paperclip's "overlay" files and its entire dictionary onto a single 1581 disk and still have ample room for documents. In addition, by virtue of Paperclip III's non-copy protected state you can hide these system files in a partition so they are not cluttering your main (root) directory. Here is how you accomplish this feat: First format a fresh 1581 disk. Then using the partition creation utility on the 1581 test/demo disk, or any other partition creation utility create a disk partition of at least 1200 blocks and format the partition. Then use a file copier to copy the Paperclip program/overlay files, your printer configuration file, and the Paperclip dictionary files into the partition you just created. Then place the following BASIC program in your root directory: :

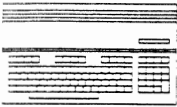
```
10 U=PEEK(186)
20 FAST:OPEN,15,U,15,"/O:PCLIP"
30 DLOAD"PCIII128G/RC" ON U(U)
```

(Note, If you do not have a RAM Expansion Unit, the filename in line 30 should read: PCIII128G) Running the above program automatically selects your Paperclip partition (in this case called "pclip") and loads the program and its overlays from the pclip partition. Once Paperclip is loaded, you can then use the Commodore DOS "IO" command (via the control > option) to return to your root directory. REU owners may want to load the dictionary into the REU before exiting the pclip directory.

**BIRDS JUST WANT TO HAVE FUN...** You know sometimes the staff at Twin Cities 128 is so caught up with interlace whatchamacallits, and high level conversations with C-128 heavy weights, that they just plain forget to loosen up and have a little fun with their C-128s! That is why I wrote this little program and slipped it in the issue while Loren and Frank were not looking.

```
10 WR=52684:RR=52698:SCNCLR:WINDOW 20,10,60,20,1
20 PRINT:PRINT"LADIES AND GENTLEMAN...":PRINT:PRINT"SUBMITTED FOR YOUR VIEWING PLEASURE":PRINT
30 PRINT"A FEW NIFTY 80 COLUMN TRICKS":PRINT:GOSUB 390:SLEEP 1
40 PLAY"TOO4HASBBBHWB":SLEEP 2:SCNCLR:GOSUB 590:WINDOW 10,10,40,20,1:PRINT"GEEZ...I FEEL FAT!!!":PRINT"TIME TO GO ON A DIET!"
50 PLAY"O4T7HAAO2W$E":SLEEP 3:GOSUB 630:GOSUB 390:PRINT:PRINT"OH YES..THAT IS MUCH BETTER..."
60 PLAY"O2T7HAAO4W$E":SLEEP 2:WINDOW 0,10,79,20,1
70 PRINT:PRINT"WAIT A SEC..I AM NOT FEELING SO GOOD...I FEEL AS IF I AM ABOUT TO DISAPPEAR!!!":SLEEP 2
80 PLAY"T205HGW.O3BT3S05AAAAA":GOSUB 450:SLEEP 2:SCNCLR:PLAY"T3S05AAAAA"
90 PRINT:PRINT:PRINT"AH...I AM COMING BACK NOW...YOU HAVE TO BE CAREFUL WITH THOSE CRASH DIETS"
100 GOSUB 520:PLAY"TOO4HASBBBHWB":SLEEP 2 :WINDOW 0,0,79,24,1
200 END
390 REM CHARACTER SWIPE
400 FOR L=0 TO 8:SYS WR,L,23
410 FOR X=1 TO 100:NEXT X
420 NEXT L
430 RETURN
450 REM CLOSE CURTAIN
460 FOR L= 100 TO 0 STEP -1:SYS WR,L,35
470 FOR X=1 TO 10:NEXT X:REM DELAY LOOP
480 NEXT L
490 RETURN
520 REM OPEN CURTAIN
530 FOR L= 0 TO 100:SYS WR,L,35
540 FOR X=1 TO 10:NEXT X:REM DELAY LOOP
550 NEXT L
560 RETURN
580 REM 40 COLUMNS?
590 SYS WR,63,0:SYS WR,40,1:SYS WR,55,2:SYS RR,,25
600 RREG A:SYS WR,(A OR 16),25:SYS WR,137,22:SYS WR,40,27:POKE238,39
610 RETURN
630 REM BACK TO 80
640 SYS WR,126,0:SYS WR,80,1:SYS WR,102,2:SYS WR,120,22
650 SYS RR,,25:RREGA:SYS WR,(A AND 239),25:SYS WR,0,27
660 POKE238,79
670 RETURN
```

**Sparrow's Parting Shots...** I suspect that once they realize that I pulled the fast one above, I will be put back in my cage for a long, long time. But before I go, here is a few parting words of advice: 1. Get in the habit of setting your clock on the GEOS 128 v2.0 desktop, everytime you boot it! GEOS automatically will time and date stamp your files, and now that Berkworks has made setting the clock from the desktop so easy, there is no reason not to utilize this powerful feature. 2. Remember that the latest version of Superbase 128 (version 3.0) in addition to supporting the 1581 disk drive, supports exporting, importing, sorting, finding, and outputting between multiple disk units! Also, the Superbase utility program now works in 128 mode making it much faster! If you have not upgraded you don't know what you are missing. Contact: Precision Software, 8404 Sterling Street - Suite A, Irving TX 75063, 214-929-4888. As the blinking cursor fades...remember what your C-128 says: "Link Arms Don't Make Them!"



## DISKMON 128

by Anton Treuenfels

There are times (often shortly after scratching a file I really didn't mean to) when I find it very useful to have a disk monitor program handy. Using one of these widely available programs I can examine and, if I wish, modify the contents of any disk sector. On the other hand, I felt that many of these programs were big, clumsy, rigid, and inconvenient to use. The program presented here was designed to overcome these perceived problems. Although it does not do everything that could possibly be wished for, it is relatively small, nimble, flexible, and easy to use.

A disk monitor program must at least be able to read, write, display and edit disk sectors. The problems of display and editing are shared by monitor programs in general, and "DISKMON128 deals with them by wedging itself into the main command loop of the C128's built-in machine language monitor. This reduces display and editing to problems that have already been solved (always a useful programming technique). As bonuses DISKMON128 gains use of the "@" disk wedge command and the ability to examine and modify sectors by disassembly and assembly as well as by simple memory dump. After employing so much of the power of the built-in monitor, about all that is left for "DISKMON128 to concern itself with is the proper reading and writing of disk sectors to and from the C128's memory.

### USING THE PROGRAM:

This version of DISKMON128 is designed to work with RLO, a linking loader program for Merlin 128 format link files. A copy of DSKMON128.RLO.O must be attached to a copy of RLO128.O (eg., by using the DOS command "c0:diskmon128=rlo128.o,dskmon128.rlo.o") to function properly. The resulting program may then be LOAded and RUN like any normal BASIC 7.0 program (a caution: the bitmapped graphics area must not be in use at the time). DISKMON128 is installed into the C128's free RAM block which starts at \$1300 in bank 0.

DISKMON128 functions as an extension of the C128's built-in machine language monitor. When operating within the monitor there are four new commands in addition to all the usual ones:

```
/R [<track> <sector>] - read sector
/W [<track> <sector>] - write sector
/# <device> [<drive>] - set device and drive numbers
/Q - disable disk monitor wedge
```

All parameters are numeric and (since a built-in monitor ROM routine is used to collect them) may be specified in any convenient base (hexadecimal, decimal, octal, or binary). Square brackets indicate optional parameters.

The read command ("/R") copies a disk sector into a buffer in the C128's memory. Once in the C128's memory, the contents of the sector may be displayed in hexadecimal and ASCII form by using the built-in monitor's memory dump command ("M"). The buffer is located at \$B00 in bank 0, so the command to display the entire sector is "M B00 BFF" (\$B00 is the autoboot disk sector buffer. Note that this page of memory is also used for the cassette buffer, and so is incompatible with any routines which might want to reside there). The memory display may be edited in the normal manner. Alterations made in this way affect only the copy of the sector in the C128's memory however, and no changes are made to any actual disk sector until the contents of the disk sector buffer are deliberately copied back to disk using the write command ("/W").

Both the read and write commands may optionally be followed by disk track and sector numbers. If a track and sector are specified they are checked only to see if they are each in the range 0-99, which allows the program to create a syntactically legal direct access command. It is left up to a drives' DOS to complain if the command cannot be complied with (usually because the DOS does not recognize the existence of the requested track and sector). This approach

is designed to avoid having to hard code into the program the internal arrangement of any past, present or future disk format by taking advantage of the user's knowledge and the DOS' intelligence.

If a track and sector are not specified, both the read and write commands use default track and sector values. In the case of the read command the contents of the first two bytes of the disk sector buffer are used as the track and sector to read. This is based on the assumptions that the contents of the buffer represent one sector in a series of sectors logically linked together in a single file, and that the first two bytes in the buffer represent the link to the track and sector of the next sector of the file. These assumptions are often true, making it possible to easily trace forward through the system of links tying files together under Commodore DOS. Particular conditions under which the assumptions are untrue include reaching the last sector of a file and before the program has read its first sector.

The write command defaults to using the values found in the current track and current sector variables maintained by the program. Since these variables will usually have last been set by the most recent read command, the default action normally amounts to putting the (possibly modified) sector currently in the disk sector buffer back where it came from.

The "/"# command is used to set the device and drive to which the program will read and write. The program defaults to device 8, drive 0. The "/"Q command resets the built-in monitor's command indirect vector to the value it had when the disk monitor was first installed, which effectively disables the disk monitor.

### ABOUT THE PROGRAM:

The main command loop of the built-in monitor is designed to accept a line of input, find the first non-space character on the line, and then jump through an indirect vector. Normally this vector points to a routine which tries to match that first character to the commands the monitor knows. DISKMON128 re-points the indirect vector to its own match routine, which checks if the character found is the disk monitor wedge character ("/"). If not, control passes to the original command match routine. If so, the program attempts to match the second non-space character of the input line to a known disk monitor command, and reports an error if it cannot do so (two-character rather than one-character commands are required mainly because the letter "R" is already employed by the built-in monitor for its "R"egister command and no alternative one-character command seemed to make as much intuitive sense as "/R"ead).

DISKMON128 opens and closes a direct access channel to a drive for every read or write attempt rather than opening and closing once for each session with a disk. The time cost of doing this is virtually unnoticeable, and it actually saves code space since session management commands (eg., changing disks) are not needed.

High-level Kernel file routines are used exclusively rather than going to the low-level Kernel serial bus routines. There are a number of mass-storage devices which patch themselves into the indirect vectors of the high-level routines, and the program should work with any of them which recognize the normal Commodore DOS direct access commands (eg., an SFD-1001 drive with a parallel cable should, but an REU with a Commodore RAMDOS program won't).

It is unfortunate that after having provided a documented method of intercepting the built-in monitor, Commodore did not provide a documented method of returning to it. The

*Text Continued on Page 28  
Source Code listed on Pages 27 & 28*

\* C128 Simple Disk Monitor  
 \* last revision: 12/08/88  
 \* written by Anton Treuenfels  
 \* 5248 Horizon Drive  
 \* Fridley, Minnesota USA 55421  
 \* 612/572-8229

\* assembly flags

LST OFF ;monitor listing  
 REL ;link file

\* program constants

CR = \$0D  
 CSR = \$1D  
 SPC = \$20  
 QUE = \$3F

DACL = 13 ;direct access  
 ;channel file#  
 CMDLF = 15 ;command channel  
 ;file#

RLOINT = \$40 ;call init  
 RLOIGD = \$04 ;ind guard  
 RLOIEX = \$02 ;ind execute  
 RLODWN = \$01 ;end address

\* program memory use

ADDR = \$60 ;pointer  
 REAWRI = \$A98 ;read/write flag

OLDMON = \$ABA ;old command vector  
 CRRDVC = \$ABC ;current device  
 CRRDRV = \$ABD ;current drive  
 CRRTRK = \$ABE ;current track  
 CRRSCT = \$ABF ;current sector

DSKBUF = \$B00 ;disk buffer (boot  
 ;block buffer)

FREEND = \$1C00 ;free RAM block end

RLOFRE = \$1C03 ;free RAM  
 ;'arbitrator'

\* monitor memory use

STKPTR = \$09 ;stack pointer save  
 ACC1 = \$60 ;numeric accumulator  
 MONPTR = \$7A ;input buffer  
 ;pointer

BUF = \$200 ;input buffer

IMON = \$32E ;command execute  
 ;vector

MSGBUF = \$A80 ;disk command buffer

\* monitor ROM

MNLOOP = \$B08B ;monitor main loop  
 NUMPRM = \$B7CE ;get numeric  
 ;parameter

\* Kernel vectors

CLSALL = \$FF4A ;close all files on  
 ;device  
 SETBNK = \$FF68 ;set i/o bank  
 PRIMM = \$FF7D ;print immediate

SETLFS = \$FFBA ;set file#, device,  
 ;command  
 SETNAM = \$FFBD ;set filename

OPEN = \$FFC0 ;open file  
 CHKIN = \$FFC6 ;set input file  
 CHKOUT = \$FFC9 ;set output file  
 CLRCHN = \$FFCC ;set default i/o  
 ;files  
 CHROUT = \$FFD2 ;output byte  
 GETIN = \$FFE4 ;input byte

\*\*\*\*\*

DSK "DSKMON128.RLO"

\* RLO header

RLOHDR TXT 'rlo'  
 DA FREEND ;stay in free RAM  
 DA RLOFRE ;RLO arbitrates  
 DFB 0 ;bank 0  
 DFB RLOINT.RLOIEX  
 DFB CR  
 TXT 'diskmon128 v120888'  
 DFB CR  
 TXT 'by anton treuenfels'  
 DFB CR,00

\* install

INSTAL LDX #3-1  
 LDA #\$00  
 JBB1 STA CRRDRV,X ;current drive,  
 ;track, sector  
 STA DSKBUF,X ;for /r or /w  
 ;without params  
 DEX  
 BPL JBB1  
 LDA #8 ;default device  
 STA CRRDVC  
 CLC  
 JSR SETVCT ;set wedge vector  
 RTS

\* Look for disk monitor command

DSKMON CMP #'/' ;wedge token?  
 BEQ MONTOK ;b:yes  
 JMP (OLDMON) ;to normal handler  
 MONTOK JSR GETSPC ;look for monitor  
 ;command  
 BEQ RETERR  
 LDX #MONADR-MONCMD-1  
 JBB1 CMP MONCMD,X  
 BEQ HAVCMD ;b:found command  
 DEX  
 BPL JBB1  
 RETERR LDX STKPTR ;restore stack  
 TXS  
 JSR PRIMM ;report problem  
 DFB CSR,QUE,\$00  
 RETMON JMP MNLOOP ;back to main loop

\* found command

HAVCMD TXA  
 ASL  
 TAX  
 LDA MONADR+1,X  
 PHA  
 LDA MONADR,X  
 PHA  
 RTS

\* monitor commands

MONCMD TXT 'qrw#'  
 MONADR DA EXQUIT-1  
 DA EXREAD-1  
 DA EXWRIT-1  
 DA EXDEVC-1

\* execute quit

EXQUIT SEC  
 JSR SETVCT  
 JMP RETMON

\* set indirect vector(s)

SETVCT LDX #2-1  
 JBB1 LDA OLDMON,X  
 BCS :1 ;b:replace old  
 ;vector  
 LDA IMON,X ;save old vector  
 STA OLDMON,X  
 LDA RPLVCT,X ;set new vector  
 :1 STA IMON,X  
 DEX  
 BPL JBB1  
 RTS

RPLVCT DA DSKMON

\* execute read/write

EXREAD LDA #'1' ;'u1'  
 DFB \$2C  
 EXWRIT LDA #'2' ;'u2'  
 STA REAWRI ;read/write flag  
 JSR TRKSCT ;get track/sector  
 JSR MAKDAC ;make command  
 JSR CLSFIL ;close files on  
 ;device  
 JSR OPNDIR ;open direct access  
 ;file  
 BCS :1 ;b:can't open  
 JSR RWSECT ;read/write sector  
 :1 JSR CLSFIL  
 JMP RETMON

\* open direct access disk file

OPNDIR LDA #\$00  
 JSR SETNAM  
 LDA #CMDLF ;command file  
 JSR OPNFIL  
 BCS :1 ;b:error  
 LDX #<DACNAM  
 LDY #>DACNAM  
 LDA #1  
 JSR SETNAM  
 LDA #DACL ;direct access file  
 JSR OPNFIL  
 :1 RTS

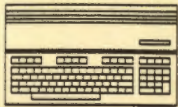
DACNAM TXT '#'

\* open disk file

OPNFIL TAY ;secondary address  
 LDX CRRDVC  
 JSR SETLFS  
 LDA #\$00  
 TAX  
 JSR SETBNK  
 JSR OPEN  
 BCS :1  
 JSR DISKST ;check status  
 ;message  
 BCC :1 ;b:ok  
 JSR CLSFIL ;close files  
 SEC  
 :1 RTS

\* close disk files

CLSFIL LDA CRRDVC  
 JSR CLSALL ;close everything  
 ;on device  
 RTS



# DISKMON 128

Text Continued from Page 26, Code from Page 27  
 the built-in monitor this routine is at the same level as the main loop, so calls from and returns to the main loop are handled as direct jumps (as opposed to BASIC 7.0's command dispatch routine, which is a subroutine of the main loop. After being intercepted, command can be returned to the main loop simply by executing an "RTS" instruction). This is a workable, if not exactly academically sanctioned, method of doing things. However it would be handier if in the jump table at the start of the built-in monitor's ROM Commodore had provided another entry, one which pointed to the start of the main command loop. This would make returning after interception less of a chancy business.

As it is, DISKMON128 is vulnerable to a ROM revision (in for a penny, in for a pound - the decision to use a ROM routine to collect numeric parameters must be blamed solely on a desire to save about 80 bytes of code. It is just as vulnerable to ROM revisions, and there is really no excuse for it. I simply yielded to temptation on this one "as long as I have to use an undocumented return call anyway...").

**ABOUT THE SOURCE:**  
 DISKMON128 was written using the Merlin 128 assembler. The most notable idiosyncratic pseudo-ops employed in the source are REL and DSK. REL instructs Merlin to create a linkable file instead of a normal object file. Merlin requires that linkable files be assembled directly to disk, which is accomplished with the DSK directive.

```

===== 2=====
* read/write sector          :1      JSR DISKST          BCS TKS1          ;b:not found
                                RTS          STA CRRTRK
                                * send disk command      JSR GETBYT      ;get sector#
                                DSKCMD STX ADDR          BCC TKS2          ;b:found
                                STY ADDR+1          NUMERR JMP RETERR
                                LDY #CMDLF          TKS1 LDA REAWRI
                                JSR CHKOUT          CMP #'2'          ;write?
                                BCS :1              ;b:error          BEQ TKS3          ;b:yes - use
                                LDY #0              ;current values
                                LDA (ADDR),Y        ;follow link to
                                ;first char          ;next sector
JBB1 JSR CHROUT          STA CRRTRK
                                INY          LDA DSKBUF+1
                                LDA (ADDR),Y        TKS2 STA CRRSCT
                                BNE JBB1          TKS3 RTS
                                JSR CLRCHN          ;'unlisten'
                                CLC
                                :1 RTS          * get byte value
                                * disk command messages  GETBYT JSR GETNUM      ;get number
                                DCREAD TXT 'u:13,0,01,00',00 BCS :1              ;b:no number
                                DCPTR0 TXT 'b-p:13,0',00 LDA ACC1+2
                                * check disk status      BNE NUMERR        ;b:too big
                                DISKST LDX #CMDLF        LDA ACC1+1
                                JSR CHKIN          ;'talk'          BNE NUMERR
                                BCS :2              ;b:error          LDA ACC1
                                JSR GETIN          ;first byte of      :1 RTS
                                CMP #'2'          ;status message
                                ;is this an error
                                BCC :1              ;message?
                                JSR PRIMM          ;b:no
                                DFB CR,$00
                                JBB1 JSR CHROUT        ;display error
                                ;message
                                JSR GETIN
                                CMP #CR
                                BNE JBB1          ;sets carry when
                                ;true
                                :1 PHP
                                JSR CLRCHN          ;'untalk'
                                PLP
                                :2 RTS
                                * get numeric value
                                GETNUM JSR NUMPRM      ;get numeric
                                ;parameter
                                BCS NUMERR          ;b:too big
                                BNE :1              ;b:found number
                                SEC                ;flag not found
                                :1 PHP              ;save flag
                                JSR GOTDLM          ;check last char
                                BNE NUMERR        ;b:not legal
                                ;terminator
                                PLP
                                RTS
                                * character fetches
                                GOTDLM DEC MONPTR
                                GETDLM JSR GETCHR
                                BEQ :1              ;b:end of input
  
```