

TWIN CITIES 128

THE COMMODORE 128 JOURNAL

\$2.10

PROUDLY PRODUCED COMPLETELY ON A C-128!!!

One year with the C-128
Looking back...Looking ahead

September 1986

FEATURES

Software Thieves
Ruining it for everyone

CBM Engineers
Q&A Forum

RAM Packing
Programming the
1700/1750 expansions

Ingredients that
make up the best
C-128 software

Making Waves
Fine-tuning your SID

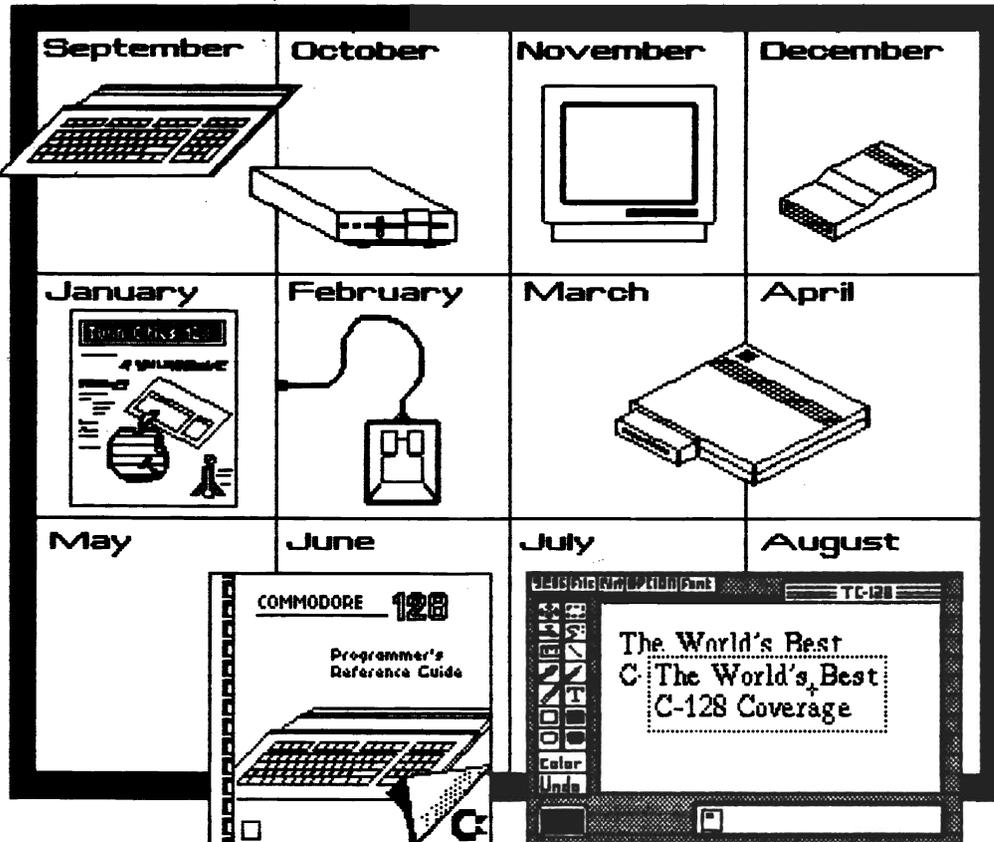
PLUS...

Forgotten BASIC

CP/M Expeditions

GEOS Update

Sparrow's Slick Tips



Reviews:

Partner 128, XREF 128,
Record Master 128,
ML for Commodores,
and the Omnitronix interface

AND FASTLOAD
MOD FOR THE 128

T W I N C I T I E S 1 2 8

Copyright September 1986

All material in this publication is protected by the laws of the United States and may not be reproduced without the written permission of the Managing Editor.

Loren Lovhaug.....Managing Editor
Avonelle Lovhaug.....Associate Editor
Hugh Morris.....Assistant Editor
Todd Madson.....Staff Writer

Contributing Authors: Bruce Jaeger, Curt Swanson

Photography by: Bruce Jaeger
Computer graphics by: Loren Lovhaug

Hardware and Software used to produce this issue:

Commodore 128 Personal Computer
Commodore 1571 Floppy Disk Drive
Commodore 1750 Ram Expansion
Comrex CR-5400 Monochrome Monitor
Amdek Color I Composite Color Monitor
Commodore 1351 Analog Mouse Controller
Commodore DPS-1101 Daisy Wheel Printer
Star Micronics SG-10 Dot Matrix Printer
Xetec Super Graphix Interface
Paperback Writer 128
Paperclip 128
Superscript 128
Superbase 128
Print Shop 64
Print Master 64
GEOS V2.0(?)

Twin Cities 128 is available by subscription to those who live outside of the Saint Paul/Minneapolis area. Send your \$22.00 check for one year (12 issues of TC-128) to:

Twin Cities 128 (International Headquarters)
P.O. Box 4625
Saint Paul MN 55104

Or call: Twin Cities 128 ONLINE
(612) 645-6482
300/1200 baud - 24 hours a day

Twin Cities 128
Issue 8 September 1986

Rumor/Opinion/Mayhem..... 4
Software Thieves..... 5
Ultimate C-128 Software..... 6
GEOS Update..... 8
Partner 128.....10
Butterfield's ML.....11
XREF 128.....12
Omnitronix Interface.....12
Record Master 128.....13
CP/M Expeditions.....14
CBM Engineers Forum.....16
Fastload Mod.....19
RAM Packing.....22
Making Waves.....24
Forgotten BASIC.....26
Sparrow's Slick Tips.....27

By: Loren Lovhaug
Managing Editor

HAPPY ANNIVERSARY C-128

That's right, the C-128 has been on the retail market for one whole year now. For me it seems much longer, because it is very difficult for me to think of not having this fabulously versatile tool. A great deal has happened in the C-128's first year of commercial distribution, here are just a few of the first year highlights:

1. The C-128 boasted North American sales of over 600,000 units and around 1 million worldwide (counting C-128d sales in Europe) making it one of the most successful selling computers of all time in the first year it was commercially available.

2. Software development for the C-128 (in C-128 mode) has focused mainly on productivity applications [and contrary to popular belief has been extremely brisk, at last count we had over 100 titles] giving C-128 owners who use their machines for so called "serious applications" an incredible number of quality choices for extremely low prices. I personally know of many Apple and PC clone owners who wish they had our productivity selection at the kind of prices we enjoy.

3. There has been a tidal wave of publishing houses producing quality texts about our machine. Just go to the computer section of any B. Dalton Bookseller or Waldenbooks or any other bookstore and I think you will be amazed at how many titles there are on the C-128. The grand-daddy of them all, "THE OFFICIAL COMMODORE 128 PROGRAMMER'S REFERENCE GUIDE" published by Commodore and Bantam, was a little late, but is a 744 page wonder reference on all aspects of the C-128 which amazingly contained just about everything you could ever want to know about the C-128 with numerous examples and most thankfully, very few errors!

4. The 1571 disk drive proved to be exactly what it was designed to be, the most versatile disk drive ever hooked up to a microcomputer. Unfortunately, a few bugs have reared their ugly head (they always do) having to do mostly with complicated operations involving the second side of the disk. But, in a bold and refreshing move by Commodore, these bugs are being addressed in the form of upgrade/replacement ROMs which are going to be made available to the end user for a nominal fee (more on this in future issues, stay tuned!).

AND NOW COMPUTER GOSSIP MONGERS, HERE ARE A FEW DEVELOPMENTS I SEE ON THE C-128 HORIZON:

1. In the next few months we should see commercial software appear which utilize the 1700/1750 RAM expansion units. We already have heard of several companies working on either upgrades of their current products or new products which will tap into the power these devices offer the C-128 world. Dropping a few names: Batteries Included, Digital Solutions, Solid State Software, Berkeley Softworks, Abacus...just to name a few!

2. Sometime in late October or early November Berkeley Softworks should be ready to release the C-128 version of GEOS [more on this in our monthly GEOS update column]. The C-128 version of GEOS could radically alter the C-128 world, and could very well give the C-128 the power and might it needs to remain a viable competitor in the shadow of the CLONES and the 16/32 bit glamour machines.

3. YES Virginia, THERE IS GOING TO BE A 1581 3.5" DISK DRIVE after all (thank goodness)! Jim Gracely and Jeff Porter of Commodore have sworn to me that this product IS A GO! This 880k slimline 3.5" 9000 cps disk drive will give C-128 owners a very nice mass storage alternative and make the C-128 even more competitive in the small business market. Unfortunately, RUN magazine's lofty claims that the consumer is going to see this drive in the fall of 1986 are totally unfounded. Both Gracely and Porter indicated that this drive should be on the retailer's shelves (though not very long, they will sell like hot cakes!) around February or March of 1987 (that is if the FCC cooperates!). So don't forego the mortgage payment this month, but file those tax returns early! [We have also heard that the major software developers should be receiving prototypes of this drive in Amiga 1010 cases sometime in the next month. We are going to beg, plead, grovel, cry, whine...etc. in an effort to get one of these babies so you can have the scoop, but don't hold your breath!]

4. Rumor has it that the retail price of the C-128 is going to increase sometime during the next few months. We certainly hope that this is not case, however, those sitting on the fence about purchasing their C-128 (or a second or third..for that matter) might consider doing it now. Around the country the going retail price for a C-128 is somewhere between \$250 - \$285, the 1571 seems to be stable at around \$250. On the more pleasant side of pricing, we have heard the price on the 1700/1750 RAM expansions are scheduled to drop sharply in the very near future, the going rates at most retailers seems to in the neighborhood of \$250 - \$275 for the 512k 1750, and \$130 - \$160 for the 128k 1700. It may pay to keep your eyes on all of these prices! [Note these price ranges were the consensus as of the 15th of August 1986]

TURNING THE FOCUS INWARD:

Those of you who have been reading Twin Cities 128 since our inception in January of 1986 have most surely noticed the progress we have been making in improving the appearance, thickness, and content quality of our publication. We are committed to bring you the C-128 owner the very best C-128 coverage possible. Unfortunately, as our quality increased, so has our costs. We have tried very hard to keep our costs low, so that you the reader can pick up Twin Cities 128 at a reasonable price. In a cost saving effort you may have noticed we cram our pages full with smaller than average typeface (If we printed using the same layout and type size as INFO as an example our last issue would have ran over 65 pages!). This practice saves a great deal on printing and mailing costs (we are still mailing via FIRST CLASS to our subscribers). But unfortunately, in order to continue to bring you the very best in C-128 coverage we have been forced to raise our cover price from \$1.75 to \$2.10. We apologize to those individuals who are on tight budgets, and we hope that you will still be able to afford us each month. We maintain that we are still one of the most inexpensive computer magazines in existence, and that we bring you more C-128 information than any other magazine for any price. We ask, that if you like what we are doing, please spread the word about us and encourage other C-128 owners to buy or subscribe to Twin Cities 128. If you do, it may not become necessary for us to raise our price again in the near future.

Software Thieves: Ruining it for everyone

by: Curt
Swanson
&
Avonelle
Lovhaug

Sept '86 Twin Cities 128

Editorial/Page 5

One of the most misunderstood and controversial topics in the microcomputer world is that of duplicating copy-protected software. The vast majority of commercial ("for sale") software is copy protected. The purpose of copy protection is to preserve the rights of the commercial software developers from the illegal distribution of their software. However, most copy protection schemes prove ultimately futile because they can be circumvented or eliminated by persistent thieves using sophisticated disk backup utilities.

Software and the Law

The copyright laws of the United States are indeed complex. But, the intent of the law is clear. The law prohibits the unauthorized reproduction or duplication of copyrighted material in any form by any means. (Since 1980, this has included computer software and accompanying documentation.) The penalties for being convicted of software theft are specific and harsh. They include provisions for treble damages, punitive damages, and prison terms for those convicted of software theft.

Why do they do it?

Certainly these penalties are sufficient to deter an otherwise law-abiding citizen from engaging in such illegal activities. And yet the software theft problem is widespread and spans all social and economic classes. Why do some people risk the penalties? Here are some typical unjustifiable excuses that people use to rationalize BREAKING THE LAW:

Transparent excuse #1: Everyone does it (the popularist argument):

The RIGHT stuff: 1) If that were true, nobody would be in the business of developing commercial software since "everyone" would just steal it anyway. 2) Two (or, for that matter, two million) wrongs still don't make a right. 3) Thousands of murders were committed last year in the United States, but my conscience just won't permit me to "run with the crowd" on this one. What's wrong is wrong, period.

Transparent excuse #2: What's the difference? Nobody ever gets caught!

The RIGHT stuff: Don't believe it! An August 18, 1986 INFOWORLD article by Scott Mace reports that the operator of an alleged "pirate" bulletin board in New York is being sued by a software publisher for copyright violations. A number of fellow software publishers are providing monies for the prosecution of this federal-court case. Arrests and convictions are well documented. Such stories are usually carried in "trade journals" and therefore (unfortunately) are not seen by the public at large. But, make no mistake, software theft is NOT a "safe" activity.

Transparent excuse #3: Software is overpriced, so publishers deserve to have it stolen.

The RIGHT stuff: Agreed, some software is priced too high. But consider the economic law of supply and demand. If a piece of software is in fact overpriced, low sales will dictate that either the price will come down, or the product will be displaced in the marketplace by a more affordable alternative product. Meanwhile, pricing carries no inherent right to STEAL anything that one cannot afford to buy.

Transparent excuse #4: I do not distribute copies to others, in turn, so no one is really harmed.

The RIGHT stuff: Most commercial software authors work under a royalty arrangement. Therefore EVERY illicit copy is money out of their pockets. You and I don't expect to work for free -- neither should they. Recently, two prominent software developers expressed their dismay at the rampant software theft, and said that they are having serious doubts about continuing in that profession considering the continual assaults on their pride and pocketbooks by software thieves.

Although software thieves (if pressed to do so) could probably add thousands more to this excuse list, the point should be clear. There is simply NO justifiable reason for "software mugging."

Finally, we would like to address the question of "backup utilities" -- those software utilities that enable the user to create duplicate copies of their software library. The computerist should and, in most cases can, make backups of their software. The same law which prohibits the illicit duplication of software allows for the OWNER of a program to make backups FOR THEIR OWN USE. Backups are generally available (to legitimate owners), for a charge, from the publisher of a program. But the ability to make your own copy allows you to use a new piece of software immediately, without the fear of something happening to your only copy. In addition, it is usually cheaper to make your own copy then pay the company for a backup. And in some cases, the copy with bypass the protection completely, leaving software that is faster to load, and won't knock the drive out of alignment. However, we are fooling ourselves if we think that the producers of copy utilities believe that their software is only used for legal copies. These companies would not be in business if it weren't for software thieves, because these are generally the people that will buy commercial protection-breaking copy utilities.

Heavily protected software makes it difficult for the legitimate user to make backup copies, and leaves them worrying about their own copy every time their disk error light starts flashing. But software producers and developers must heavily protect their software, or have their paychecks ripped out of their hands every time a thief steals one of their programs. What should be done? It is clear thieves must be stopped in their tracks. Hopefully, stiffer penalties are on the way, as well as improved scrutiny and increased litigation. This in turn, will serve as a warning to those already committing the crimes to STOP what they are doing and think about the implications of their actions. Meanwhile, increased support should be given to those companies who have attempted to make copy protection as painless as possible in the form of protection that removes the focus from disk-based protection (that leave our disk drives crying) and move towards hard protection (such as dongles, ROM-based, or optical key protection) that allows the legitimate user to make backups. However, the ultimate solution lies in making people aware of the damage software stealing is doing to the industry, to the developers, and to the users. Until people are aware of the damage that is created by thieves, the situation will only continue to prove destructive.

Ingredients that make up the best C-128 software

by: Loren Lovhaug

One of the reoccurring themes in Twin Cities 128 has been the role that quality commercial software has played in the C-128's success story. So far we have been generally very pleased with the C-128 software that has appeared during the C-128's first year in the market place. This is not to say that there haven't been a few dogs, but fortunately the vast majority of C-128 software has been very good. Reflecting upon the numerous winners and the handful of losers there arises a definite pattern or criteria which defines the attributes of the superior products and the pattern of failure inferior ones.

The pattern of success seems to be defined in three distinct areas: 1) Hardware and Software flexibility, 2) Utilization of the C-128 specific features, and 3) Usability and cost.

Hardware flexibility is crucial. The program must be able to perform effectively under a variety of system configurations. This means that the program should have the ability to adapt itself to the equipment I own and not vice-versa. For instance, most good word processors for the C-128 give the user a fairly large choice of printer configuration options to suit the huge array of different kinds of printers on the market today. Obviously, there is always going to be a off brand or newcomer appearing in the market so the better packages also incorporate an option for tailoring configuration drivers in addition to any presets that may be provided. This allows true flexibility because no matter what equipment the user owns it will work and insures that the program will not become obsolete directly as a result of the evolution of new peripherals.

Also extremely important is software flexibility. Flexible software allows the user to make his/her own choices according to his/her preferences. This means you can have your choice of colors, data entry templates, display modes, input devices etc. Flexible software also gives you a variety of choices as far as operation and automation goes. A superior example of this concept lies with Precision software's excellent database, Superbase 128. Superbase is perhaps the most flexible product available for the C-128 today. It allows you to completely design your own multi-page input screens and is capable of total automation and customization through its own built in programming language which is very similar to BASIC 7.0. So powerful is this user program facility that Superbase itself can be totally masked from the user, replaced by a customized application program which might be tailored to suit a specific purpose.

Another aspect of flexibility is software data exchange compatibility. Packages like Superbase and Superscript as well as Digital Solution's Pocket Series (Pocket Writer, Pocket Planner, Pocket Filer) allow the user to very easily exchange data between the integrated programs in their "family" as well as with other outside programs. This allows the user to choose the best tools for certain aspects of a task and then port the data over to another program which is better suited for another particular task. There is one rule however that one ought to remember about software flexibility, namely, the more flexible a program is, the complicated package tends to be, meaning that it may take a little longer to learn how to use these packages effectively.

Another factor that makes or breaks a piece of C-128 commercial software is its utilization of the features of the C-128. Perhaps foremost in this category is memory usage. Those of us who have been around computing for a long time realize that 128k of random access memory need not be considered confining. Certainly those programmers who wrote software to fit into the smaller environment of the C-64 should be able to crank out superior code now that the memory constraints have been loosened. But there is always a temptation when you have more memory to write less efficient code, thereby taking precious random access memory away from the user. Over the last year or so we have observed that the most successful software companies in the C-128 world have adhered to the axiom that "less is more" when writing code. For instance, the code for Pocket Writer and Pocket Planner, two efficient yet extremely powerful productivity packages fit entirely in one 64k bank of memory, leaving the user a whopping 64k of RAM for his/her data. Another excellent example of how efficient code benefits the user is Bobstern Pro which also utilizes an entire 64k of RAM for data (buffer) storage, while offering the user an incredible amount of power and flexibility.

Of course, efficient coding can have some other benefits besides leaving the user with a large amount of memory free for applications. For instance, Superbase and Superscript are written in such a manner that they can both be loaded into memory at the same time as co-resident applications which can share data.

Superior C-128 software designs make use of is the greater speed and storage capabilities of the 1571 disk drive. The challenge here lies in the fact that a software company must of course produce software which is compatible with the 1541 as well to maintain user flexibility (see above). The best programs allow the disk drive to decide whether we are working with a single or double sided disk and whether or not we can activate such wonderful features such as fast serial and burst mode.

The keyboard enhancements of the C-128 offer additional opportunities for the C-128 software developers. Use of the functions, the special keys (i.e. HELP, ALT, ESC, etc.), and the numeric keypad for special purposes can really enhance the sophistication and the ease of use of a product.

The final category which defines the excellence of a C-128 product may in many ways be the most important to the end user, namely usability and cost. Of course a good piece for commercial software should do what it claims to do, but there is more to it than just "working". So much of what makes a good program good, lies under the somewhat nebulous moniker of "style". A good piece of C-128 software should be able to meet the users requirements for functionality while still being pleasure to use. This means that the software should incorporate tasteful screen displays and outputs and a well conceived (read logical) command structure or menu system. The documentation should be clear and provide examples as well as command summaries and reference sections AND MUST INCLUDE AN INDEX as well as a table of contents.

And of course price plays a very important role in

the success and failure of a C-128 software product. We have found that the majority of superior C-128 pieces go anywhere from \$30 - \$75. One other consideration that ought to be made concerning price is software protection backup/upgrade policies. The most successful software companies in the C-128 world either do not copy protect their software or offer non disk based protection (dongle/key) or offer reasonable backup/upgrade policies.

In the end, of course, the key ingredient to successful C-128 software is the end user, whose personal tastes and pocketbooks ultimately choose the winners and the losers.

C-128 owners: Are you tired of second class treatment?

We certainly are, and we are doing something about it! The C-128 is a superior microcomputer with unique abilities that deserve extensive coverage. That is why we started Twin Cities 128, the only magazine devoted exclusively to the users of the Commodore 128. Check us out and get the FIRST CLASS treatment you and your C-128 deserve.

Twin Cities 128

P.O. Box 4625
Saint Paul MN 55104

Bouncing Dog Software Presents:



Data Master 128

A database designed
exclusively for the C-128 for
operation in 80 column mode.

Only \$29.95

Send Check or Money Order
plus \$2.00 postage & handling

Bouncing Dog Software

P.O. Box 6753

Minneapolis MN 55406

Minnesota residents must add 6%
sales tax

Easy to Use!

- No complicated formulas or syntax to memorize
- Menu driven - point and pick operation
- Online help at all times

Does the Job!

- Instant access to data
- Prints mailing labels
- Creates reports with breaks and subtotals

Full Featured!

- Full range of data types
text, numeric, logical, monetary
- Powerful record operations
define, add, delete, restore,
update, select, scan, sort
- Versatile file operations
create, display, index, erase
make subfiles, or even
password protect
- Access to disk operations

GEOS Update

Compiled by:
Loren Lovhaug

Since the first installment of this column last month I have had many readers ask me why we are devoting an entire column to developments concerning a particular product produced by a particular third party vendor. My answer lies in the following statements:

1. The Graphic Environment Operating System (GEOS) is the only alternative operating for the Commodore 64 and Commodore 128 microcomputers officially endorsed by Commodore.

2. The icon-based operating system GEOS brings to the C-128 makes the machine more competitive with machines like Apple's Macintosh and Atari's 520ST and 1040ST, offering potential computer owners the benefits of an icon-based operating system and presentation-quality graphics at a much lower cost. In many ways, this is just another demonstration of why the C-128 is perhaps the most flexible computer on the market today. In my mind, if the C-128 is going to have a long lasting niche in the microcomputing world, C-128 software development must remain on "cutting edge" of the marketplace.

3. GEOS brings the C-128 into the fastest growing area of computer applications, namely, desktop publishing. Until recently this field was closed to those who could not afford a \$2000 computer and \$400 software. The C-128's entry into this field represents a revolution in computing power for the masses. Don't believe me? Take a good look at the magazine you are holding in your hands. Without a tool like GEOS we could not afford to produce a magazine of the quality you possess.

4. GEOS is an amazing programming feat. Berkeley Softworks has certainly pulled off what many people would have believed was impossible on an 8bit microcomputer. But beyond being competent software developers, the people at Berkeley are going out of their way to support the end user, by making themselves available on national telecommunications networks to answer questions, continuing development to improve their products, and putting forth an "open book" attitude toward outside developers who are interested in creating applications which run under GEOS even as they themselves continue to improve and build upon their product.

Well there you have it, I hope you agree that our reasoning is valid. And now, on with the GEONEWS:

GEOS 128

We have not yet received our pre-release version of the C-128 version of GEOS (we are hoping to have a copy for review in our next issue) but Rob Siegel and Lee Llevano of Berkeley Softworks talked with us about it. First off, although there has been a lot of speculation to the contrary, GEOS 128 will run on the C-128's 40 column composite video screen. Lee Llevano cited the following reasons for their decision:

1. Compatibility with existing GEOS applications, as well as future ones. It is important to remember that GEOS is first and foremost is an operating system, and for this reason must operate in a "standard" manner so that applications can effectively run under it. The C-128 version of GEOS is simply an extension of the existing C-64 operating system which uses the some of the added C-128 mode facilities, such as added memory and

faster disk access. Because of this, developers can create their applications and market them to both C-128 and C-64 owners (thereby increasing the chances that we are going to see many GEOS applications because of the huge combined market), although C-128 owners will have a more powerful framework within which to use those applications.

2. Certain aspects of the 8563 video display controller (the chip which produces the 80 column RGBI video of the C-128) make it more hostile for GEOS type applications than the VIC II composite video controller. These include the lack of sprite animation facilities which make the movement of the mouse pointer and the dragging of icons swift and smooth, and the fact that the 8563s display memory is not directly accessible as part of normal system RAM.

3. Because of the nature of GEOPAINTs and other applications virtual (scrolling) screen techniques it was decided that the added on-screen resolution of the 640 x 200 screen was not as vital as with applications who do not rely on virtual screen operations.

Frankly, I agree with Berkeley's reasoning on this matter, especially because of the added potential for application development by making the C-128 and C-64 compatible, although it would have been nice to have the added screen resolution (especially as laser printing continues to decrease in cost) and the ability to have the C-128's CPU shifted into high gear (2 megahertz mode).

Here is a list of some of the enhancements to the existing version of GEOS that will be incorporated in the C-128 framework:

- Support for the 1571 disk drive. This is very important, especially since many GEOS applications are disk intensive. Many users of the current versions of GEOS complain about how slow the operating system is because of the amount of disk accessing that takes place. With the C-128 version of GEOS and a 1571 the speeds of these accesses will be greatly increased. Another benefit of utilizing the 1571 drive will be the doubling of the storage capacity of your work disks. This is also very important, since with any disk based operating system there a greater storage overhead for system files.

- Support for the 1750 RAM expansion. We have used a prototype version of GEOS that supports the 1750 RAM expansion (similar to the version shown at CES) for a few weeks now and we are thoroughly impressed. The support for the 1750 is in the form of a RAMdrive which works at blinding speed. Once GEOS applications are placed into the RAM drive they can be accessed instantly. Also since the RAM expansions are treated as ultra-fast disk drives the complaints some users have about disk accessing within in GEOS applications becomes moot. Scrolling in GEOPAINT becomes very smooth and instant and the accessing of options such as text or photo managers or the cutting and pasting of text or photo scraps becomes fluid and natural instead of somewhat slow and cumbersome as in the current release. With the RAMDISK, such applications as GEOPAINT can operate as fast as their counterparts.

- Concurrent operations. Because of the C-128 has twice the random access memory of the C-64, there

is room for special enhancements, one in particular is called the "applications switcher". This option will allow concurrent operation of GEOS applications, allowing you to immediately switch between application programs in memory. With the "applications switcher" and the RAMDISK, such applications as GEOPAINT can operate as fast as there counterparts on such high power systems as the Macintosh.

Okay, I know what you are thinking: 'Sounds great, but when and how much!' Well that is perhaps the best news of all, both Rob and Lee indicated that the C-128 version should be available on the retail level in late October or early November, just a little bit later than previously announced. And although no final suggested retail price has been announced, Lee told me that registered GEOS 1.2 owners can upgrade to the C-128 version for just \$15.00!

GEODEVELOPMENTS

Another product soon to be released from Berkeley is DESK PACK 1. DESK PACK 1 is a collection of applications utilities that run under GEOS. The most impressive of these utilities is a program called the GRAPHICS GRABBER which allows you to integrate graphics from PRINT SHOP, PRINTMASTER, and NEWSROOM into GEOPAINT. This utility is very easy to use and works flawlessly (Some of the results are presented on this page). This program really benefits two diametrically different groups of GEOS users, those who do not have the time, skill, or desire to create original graphics and those who want to use the superior editing capabilities of GEOPAINT to alter the "canned" designs. This program also expands the capabilities of GEOS while extending the life and value of other utilities you may own.

Also included in the DESK PACK 1 package is a calendar and appointment organizer which allows the viewing of calendars past and future and the ability to create agenda entries on a daily basis. Another program on DESK PACK 1 is the first GEOGAME! This is a nifty BLACKJACK game which features animated cards and casino style wagering. This game is extremely slick and addicting, and a ton of fun! Finally included is an ICON EDITOR which allows you to customize the GEOS icons of your own programs. We do not at this time have a suggested retail price for DESKPACK 1, but we are told it should be available before the end of the year.

Another excellent product soon to be released by Commodore and supported by Berkeley is the 1351 "true mouse" controller. This analog mouse gives the GEOS user much greater control and is far more responsive than a joystick or the 1350 mouse controller. For delicate, professional graphics work with GEOS this baby is a must! Ever since we received our prototype from Commodore the staff here at Twin Cities 128 has been fighting tooth and nail over it. But alas, being the boss has its perks, so IT STAYS AT MY COMPUTER! No word on price or release date.

Also in the works are a variety of applications programs and input drivers which will greatly expand the GEOS system. A few of these include the following: A Computereyes applications program which allows digitized pictures to be incorporated into GEOPAINT, A Koala pad input driver, GEOFILE a database program, GEOCALC a spreadsheet program, and a greatly enhanced version of GEOWRITE.

**NOW AVAILABLE
RECORD MASTER 128
From WOODSoftware**

- * Automatic 40 or 80 columns
- * Supports FAST mode in 80 columns
- * Program fully self-contained
- * Compiled for speed
- * May have up to 40 fields per record
- * Send output to printer, screen or disk

ONLY \$49.95

**For More Information Contact:
WOODSoftware
P.O. Box 16193
Wichita, Ks. 67216
(316) 529-1861**

**To Order Contact:
Mid-Kansas Computers
P.O. Box 506
Newton, Ks. 67114
(316) 283-0208**



Review by: Avonelle Lovhaug

As a computer addict who uses my computer for just about everything, I find that I am often annoyed by the problem that one piece of software simply will not do everything that I need it to do. This is always very apparent when I am in the middle of a large database, and I realize that I really need to schedule an appointment with my doctor. Rather than loading up my schedule program, I will probably wait until I'm finished working with the database; this generally means that I will forget to schedule the time. However, Timeworks evidently understands what it is like to be thinking about 10 or 15 things at once, and has developed Partner 128, software that will help ease this difficulty.

Partner 128 is a cartridge that supplies its owner with a built-in calculator, memo pad, appointment calendar, address/phone list, a screen dump function, and a method of temporarily freezing your keyboard. All of these functions are available at the touch of a button located on the top of the cartridge. When you press the Partner 128 button, your current software is frozen temporarily, and the Partner menu is displayed in a window. The menu offers the previous examples as choices, plus a place to set up your current drive and printer device numbers, as well as access to DOS commands.

The appointment calendar is an important part of the Partner package. The calendar offers monthly calendars up to December 1999. When a day is selected, the program leaves the user ample space to record his "to do" list, appointments for the day, and key tasks. Unfortunately, because screen space is limited (especially within the window) the only thing which appears on the monthly calendar is the title for the day, which can consist of only 10 characters. The appointment calendar can also be printed, and you are given a choice of printing a single day, a week, or the entire month. The edit features within the calendar include the ability to copy information from one day to another (within a single month, only) and also to clear the entire display at once. Also, a disk directory can be obtained here, as well as just about anywhere in the rest of the applications. When you are finished with the calendar, hitting ESC once will take you back to the Partner menu, and once more will take you back to your previous software. Your calendar information is stored in the cartridge RAM. However, before powering down, you should save this information to disk.

The memo pad option offers the ability to quickly write and print a brief note or letter. It operates very much like a rudimentary word processor, offering tabs, insert and delete, and word wrap. The Partner manual suggests that the user could place help screens for his favorite software in the memo pad, to be recalled later when needed from Partner. The memo pad area can be printed in two ways. First, when you have finished with your memo, you can use the print option by pressing F7. Partner also allows you to make your printer into a typewriter, which will print each line immediately after pressing <RETURN>.

sort this list by any field (for instance, by zip code) and print this information or autodial your phone (if you have an autodial modem). When printing the information, you have a choice of printing the address records, printing labels or envelopes, or printing phone numbers. This is one of the best features of Partner 128, as it makes whipping out a quick address label or envelope a snap. Also, this information can be saved to disk. If you only need a simple database to maintain names and addresses of friends or relatives, this may be all that you ever need buy. Partner 128 can hold up to 60 addresses at a time in any given file, but you could save them to different disk files if you wished. One problem with this database is that it is designed for a United States address format, and foreign addresses will not generally fit well in a typical city, state, zip code format. This is especially true since you will probably want to include the name of the country the person resides in, and there is no feature within Partner to include this information. This is may not be important to the home user, but for the business user this is of vital importance, unless it is a VERY localized business.

An additional option of this software is the calculator. When selected, the calculator option displays a calculator in the Partner window. Operations are performed in a manner similar to a real calculator. To add or subtract, you must use the "+" and "-" keys, respectively; to multiply or divide, use the "*" or "/" keys. The up-arrow key raises a number to a power, and a colon will change the sign of a number. This calculator also has the ability to store a number in temporary memory, to be recalled later. One final feature of the calculator option is that your printer can print out the calculator's information in "paper tape" fashion.

There are three other features that Partner 128 offers. For those persons who own a 1541 drive, Timeworks claims that the "Swiftload" feature of Partner will allow the 1541 to load programs as fast as a 1571 drive. Although I didn't have an opportunity to test this with any 1541 drives, I do know that I had no problem loading any software with this cartridge installed. However, if you are an extensive user of Paperclip, BobsTerm Pro, or any other program which requires a dangle in joystick port #2, I suggest that you find a "Y" connector, as Partner 128 draws some power from this joystick port by a cable which runs from the cartridge into the port. (Timeworks is offering these "Y" connectors for a nominal fee.) Partner 128 also has a built-in screen dump utility which will let you print your current screen (no graphics screens, however). If you have had trouble with people accidentally bumping your computer when you were away from the keyboard for a moment, thus losing valuable data, Partner 128 also has a keyboard locking option which will freeze your keyboard until you type in a password which you choose. This option may be worth the program's price itself!!

Despite the fact that I really enjoyed using Partner 128, I found I did have some problems with

in Partner, for instance, a memo in the Memo pad. Another difficulty I had was that on one occasion, my cartridge worked its way loose from the computer, and when I pressed the cartridge button, it froze my software, but refused to call up the Partner menu. This meant I lost all of my data. One final quirk I found with Partner was the way it used the <INST/DEL> key. Generally, this key will erase characters to the left of the cursor. Instead, Partner uses this key to delete the character that the cursor is on, moving everything that is right of the cursor one to the left. I found this occasionally annoying.

The manual for Partner is concise and well written. Warnings are blazened everywhere with notes like: "DO NOT call up Partner if the disk drive is running", as well as some suggestions about when not to use Partner with certain software. Apparently database software tends to be the easiest to destroy, since this type of software tends to have heavy disk access. This is unfortunate, as I believe this software is definitely more useful for the business user, not the home user, and database software is an important part of business computing. Also, the manual clearly states that Partner will not work with Vizawrite or any other cartridge based software (forget using your RAM expansion), nor will it work with any 40 column programs or when there are high-resolution graphics on the display.

Timeworks seems to be very dedicated to this product. In addition to the fact that Partner 128 comes with a demonstration disk that gives the user some examples of how the program can be utilized, the manual also suggests that some of Partner's features can be re-programmed to create other useful utilities. This is only available through machine language, but if you are interested, Timeworks is offering instructions for this option, complete with a disk, documentation, sample files and a sample utility. This also will register the interested party as a user of the utility documentation, which will give them the option to receive help with utility writing from Timeworks.

If you are looking for software that can be coresident with your other programs, and can offer the kind of features I've been describing, I suggest you look into buying Partner 128. I make this recommendation with one qualification, though: please take a look at your most used software, and insure that it is not cartridge-based, or calls up high resolution screens, before slapping down around \$50.00 to purchase it. If it is compatible with your usual software, this could be a real deal.

It has been said that, "Real programmers do not code in high level languages, instead they gut it out on the level of the machine." Well I do not really believe that, but I must admit a certain admiration for those individuals who are proficient machine language programmers. There are many of these guys who know memory maps better than their home town street maps and can do binary, octal, and hexadecimal conversion in their head at a blink of an eye. But in the Commodore world, mention ML and one name comes to mind as standing head and shoulders above the rest, the Grandfatherly Guru, Jim Butterfield. I have met Mr. Butterfield a couple of times and attended his seminars while stumping Twin Cities 128 around the country, and I can personally attest that you will be hard pressed to find a better teacher or someone with more expertise. That is why I was overjoyed to learn that THE GURU had recently completed a revision of his venerable MACHINE LANGUAGE for COMMODORE COMPUTERS that included extensive coverage on C-128 ML programming. I own a copy of the original text, and for my money there was not a better text for beginners, until now. THIS BOOK IS A MUST HAVE for anyone who serious about learning to program in ML.

What makes this book superior is the way in which Mr. Butterfield presents a very complex subject in a manner which is easily understandable and nonintimidating. This text is divided into 8 chapters which are organized in a logical sequence from first concepts to more advanced concepts. MACHINE LANGUAGE for the COMMODORE 64, 128, and other COMMODORE COMPUTERS presents material on such topics as machine language arithmetic, addressing modes, and kernel routines that is not only informative and complete, but also in a manner that makes it ideal as reference guide long after the reader has mastered the concepts presented within. Each section is complimented with its own challenging, but worthwhile set of exercises. The answers to these exercises are also included in the text with extensive commentary on the solutions making this book ideal for review purposes. Mr. Butterfield has also included a rather hefty appendix section which contains a wealth of material on ever Commodore computer that has been produced to date, including memory maps and technical overviews, and superior illustrations. This technical history of Commodore computing is very interesting and of great value to anyone who owns more than one make of Commodore computer.

I have found only one minor flaw in this text. This is the slight inconvenience of having to often consult the C-128 appendix in the back of the book when working through the main body of the text to get to the C-128 commentary and exercises. However, the book makes it very clear within the text when a C-128 owner should consult the appendix, and a simple bookmark makes this aspect of an otherwise flawless text almost unnoticeable. 377 pages, \$16.95, A Brady Book, published by Prentice Hall Press.

Cross-reference ("XREF") programs have probably been around since the first high-level computer language was written. An XREF program reads your source-code, then prints out a listing of the locations of the use of all variables, GOSUB's, etc. This can be a very useful debugging or documentation tool. Suppose, for example, your program keeps coming up with an answer of "X=101," a result that you know to be nonsense because X was only supposed to be 0 or 1. Using a program like XREF 128 might tell you that you also inadvertently used X in a counting loop ("FOR X=1 TO 100") somewhere else in the program--possibly saving hours of frustrating debugging, especially if your BASIC program is quite large.

Abacus Software's XREF 128 is a powerful cross-reference program. In fact, its power can be somewhat frustrating! In its unmodified, all-options selected form, it generated a lengthy 10-page printout of my 55 block long program, including such "useful" information as 3/4 of a page telling me wherever I used an "=" sign. Fortunately, programmer R.C. Wainwright included a program on disk to add to (or subtract from!) the BASIC keywords and symbols that XREF 128 looks for; I simply deleted such super-common keywords as "=", "+", "-", etc. The program also gives you the option of not listing all the BASIC keywords, and of not listing what it calls "the rest," which are line numbers and variables. You can specify line ranges for either option, further reducing the size of the printed output if desired.

XREF 128 lists all of the numeric constants ("X=37.5") used in a program, as well as all target line numbers for GOTO's and GOSUB's. However, it inexcusably makes no distinction between the two in the listing! It treats "X=100" and "GOTO 100" as the same!

Powerful as a good cross-reference program can be, what is generally more useful when debugging a program is a good "FIND" command, which you can use when actually working on the program. I've done a good bit of commercial programming, and I can count on the fingers of one hand the times I've needed a cross-reference program.

XREF 128 is heavily copy-protected, somewhat user-hostile and overly cryptic in its screen prompts. (The Commodore 128 has memory to burn for user assistance--why won't more programmers get a book of matches?) The disk is auto-booting, that latest abomination to hit the Commodore camp. And, you have to turn the computer off to get out of the program, surely a lot of nonsense for a simple cross-reference utility that you can get from the public domain for free.

I recently gained an opportunity to pick up an Omnitronix Deluxe RS232C interface for my C-128. The RS232 interface will allow you to use standard RS232 devices with your Commodore. This includes modems, such as the Hayes Smartmodem and various true-ASCII printers. Commodore computers do have a type of RS232 connection, but the actual signals generated with it are quite different from other standard RS232 devices, and the connector is also very different than a standard RS232 (DB25) connection. With this interface plugged into the user port you can now use nearly any RS232 type device on the market.

The interface comes packed in a small cardboard box with protective styrofoam and includes a 17 page user manual. The manual suggests various types of connections, from RS232 modems, to Brother EP-44 printers, to the HERO robot! The Omnitronix interface allows you to "reach out" to the world of professional level devices that would be previously inaccessible by the C-128. The manual is thorough and guides you step by step in connecting the interface and how to access it.

The interface has three switches on the front panel that select which type of device you are using. If you are using a modem, set the first two switches to modem. If you are using a printer, set them to printer. The third switch, which is labeled "Normal" and "Inverted" is discussed in the chapter on programming, and should usually be set to "normal".

The manual discusses all sorts of intriguing ideas for using the interface, such as in word processing, the use of modems (a basic type of terminal program is included), and how to set the baud rate using the C-128's Command and Control registers. There is also a technical section on how to read and write directly to the RS232 bits, which has examples written in BASIC.

There is another BASIC program included which allows you to transfer a basic program from one Commodore computer to another directly using the RS232 lines. The program is somewhat small and very bare-bones, but gives you a good idea of how to use the interface. The manual concludes with a table of pinouts for the modem and printer settings. In this way, a programmer can determine exactly what lines are being used to send and receive data.

I found the Omnitronix Deluxe RS232 interface to be a good value. It allows the serious Commodore user access to devices such as 2400 baud modems and high speed plotters - devices usually inaccessible to the C-128 user. My only real gripe about the Omnitronix interface is the casing that plugs into the user port. It is made of a light, black plastic that allows you to open the casing and inspect the innards. The plastic is of a very light gauge and the hinge on the case looks as if it could break very easily if not handled carefully. At the very least, such an interface should be encased in either a harder plastic or perhaps metal, especially when the retail price sticker on my particular interface was \$69.95. On the other hand, I found the electrical workmanship to be first-rate, very neatly laid out and professional looking. As it is, it's a very nice interface. If Omnitronix retools their casing with a heavier gauge plastic they could have a real winner on their hands. You can contact Omnitronix at P.O. Box 43, Mercer Island, WA 98040. Phone (206) 236-2983.

Review by: Avonelle Lovhaug

I don't know why people keep complaining about a lack of software for the C-128. I've had no trouble finding new programs for our favorite machine, and recently came across a new database program that utilizes the features of C-128 mode. RECORDMASTER, by WOODSoftware of Wichita, Kansas is a well thought-out database program for the C-128.

Record Master comes in an attractive, red on gray three-ring binder, with useful (but not overzealous) documentation. The program is not copy protected, which relieved me because in addition to being able to make a back up of this program, I didn't have to worry about my drive heads being knocked all over the place. The program also takes advantage of the autoboot features of the C-128, and is available in 40 or 80 columns. After a brief introduction and a glossary of terms that the manual will refer to (pretty standard stuff for the computer junkie, but very useful to the novice), the manual instructs you to make a back up copy of the program. I would advise you to use a different back up utility than the one provided on the disk, I found it entirely too slow. Also provided on the disk is a "SETUP" program which will allow you to choose your default configurations for the disk drives, printer, and screen colors. After choosing your defaults, the information will be stored on your master disk for the program to use.

After self booting, the main menu offers 4 choices: open an existing database, create a new database, recover a corrupt database, or quit the program. Throughout the program, menus and prompts are clearly displayed, so that the program is easy to use and understand. To choose an item from a menu, you must type the first letter of the choice. Creating a new database is fairly simple. After choosing "C", the program will prompt, "Is this database like another database?" If you have similar databases (for instance, two similar databases that you keep separate for each different year) then you can use this option. To create a new database, Record Master will need to know approximately how many records the database will have, the number of fields per record, and how many characters per record. This is one of the unique features of Record Master. Instead of choosing the number of characters to be allowed in each field, the program requires you to do a little math on your own. Compute how many letters will be needed for each field (approximately), add them together, plus add 2 more for each field. This is the characters per record. Because of this method, if you have one record that has a very long first field, for instance, you won't have to restructure the entire database for that one record, or abbreviate the record to make it fit. Instead, Record Master adjusts the field size accordingly, only running into difficulties when the characters per record are too large. You shouldn't have a problem with this if you follow the directions carefully. (Incidentally, if you want to limit the size of a particular field, this is possible.) Next, you will choose a name for each field, and type them in at the appropriate prompts. The last thing required to create a database is choosing a key(s) for your database. Record Master will let you have as many keys as possible, and each key can include up to five fields. The keys are used to arrange the records in a specified order. I found that I appreciated the ability to choose more than one field for my key, as well as being able to give the program more than one key at a time. (When printing, the

program will then prompt you as to which key to sort by for the hardcopy.) After all of this information has been entered, the program will tell you the size of your file when it is full, and the amount of RAM used. Interesting statistics, although their usefulness has yet to be proven to me. Finally, the program will let you save the database format on a disk.

Now the program will take you back to the main menu, so that you can open your new database. The open database submenu will allow you to access the records, print mail labels, print reports, switch the case of all the records in your database (upper to lower and vice versa), use any disk commands, go back to choose another database, translate your database, or quit the program. Record access gives you many of the standard features, allowing you to add, change, delete, scan, search, or print a hardcopy of the current record. In addition, there is also an option to mass enter your data, to be merged with your database later on. This can save considerable time, especially if you don't like disk access after every record entry, or if you don't have information for all of your fields at the present time. Mass entry will let you choose which fields you wish to enter, and they will be stored in a temporary sequential file. This can then be merged with your database. You also have the option of reordering your records in any way that you choose, limiting the size of a particular field, or doing calculations to update certain fields. One final option, destroy database, is also available. Although the manual didn't mention this, the menu gives you the option to destroy a single record, or an entire database. However, when I tried to destroy a single record, the disk drive gave me an error, which eventually dropped me back to the main menu. This happened whenever I tried the "destroy single record" option, although destroying the entire database was not a problem. Personally, I would never use that option to delete a record anyway, when you have that choice in the "work on records" menu, where you can also see the record before you kill it. I suspect that this option was later eliminated since it is redundant, and was never entirely killed from my version of the program. Another anomaly in the program is the mention of a "user" option in the record access menu. This option supposedly gives the user the ability to alter the program. However, my version of the program did not offer this choice in the menu. Again, I didn't feel it was a necessary portion of the program, and I suspect that it has to do with a difference between upgrades. In fact, the "user" option is so rarely mentioned, it would be very easy to miss in the manual.

The mail labels and print report options from the open database submenu offer the user the two hardcopy options that are most needed. The mailing labels option will let you load, create, change, and save a layout that is suited to your labels and the fields they will contain. When creating a label layout, the prompts will ask questions regarding number of printed lines, number of lines between labels, number of labels across, and the field information that will be on the labels. The prompts make printing labels very easy, and these layouts can be saved to reuse. The print report option, again, will let you create, change, load and save layouts to disk. The layouts hold the following information: number and text for header lines, number of columns, information contained in columns, the

selection process to choose which records will be printed, and the ability to calculate certain columns (for instance, a total at the end of each report). The manual explains this all clearly, giving examples throughout. The ability to select records (which is also available with the mail labels option) has an entire chapter devoted to it in the manual.

The method the program uses to guide the user through the selection statements is quite unique, and highly useful. When creating selection statements, the program will first ask if you wish to select certain records. After pressing <Y>, the screen will list all of the fields in your database, and begin the selection statement with: "If (...)" and instruct you to enter the field number of your choice. The manual explains that the selection statement is made up of four parts with the form: "IF (field name) <equality> (argument) <connector>". After you enter the field, the statement will reappear, this time including your field name and your choice of 9 equity types. When you choose your equity type, the program again incorporates your choice into the statement, and requests your argument, which can be either alphabetical or numeric. Finally, you will be prompted for a connector, which is either "and", "or", or "then print". An example of a final selection statement might be: "IF Last Name equals "Smith" then print". I found this method of explaining and guiding the user through selection statements extremely helpful and valuable as a logic tool as well as a practical method to use the database. Several examples are also included in the manual.

Some other impressive features included in this program are: complete instructions for creating sequential files to be read by a word processor, as well as easy to understand prompts to guide you through, complete information on translating a database to a different format, an option within the regular part of the program to help recover corrupt files, tables in the appendix to give you an indication of how many records will fit on certain disks under certain conditions, a command summary appendix, a full index, and technical support. The end of the manual also suggests that data format disks will be made available for such tasks as payroll, accounting, inventory, check book balance, and other categories.

WOODSoftware will take phone calls for troubleshooting if you are having difficulty with the software, if you have sent in your registration card. Backup copies and free updates for two years are available for \$10.00. (Since the program is unprotected, this option may not be worthwhile, unless updates are very fine, although \$10.00 is not a bad price for backups compared to some companies).

The manual is fairly well written, although my only complaint is that some of the spelling and grammar need to be cleaned up a bit. I always understood what the author was trying to say, and found that the information provided was adequate and useful.

Record Master for the 128, at \$49.95, is a good buy for excellent productivity software. I would recommend this program to anyone looking for a flexible, easy to use database program. To get a copy of this program, write to WOODSoftware, P.O. Box 16193, Wichita, KS 67216, or call 316-529-1861.

ADD 40K TO YOUR DISKS!

The Commodore 1571 disk drive is a very versatile device, capable of reading and writing to many different types of MFM (Modified Frequency Modulation) disk formats. These formats, such as Kaypro (double and single sided), Osborn (single sided only), Epson (single and double sided), and IBM system 34 CP/M can be used under CP/M on the C-128. All of this is accomplished simply by reprogramming the disk controller. Up until recently, there was no way to format a diskette in one of these alternate formats - you were restricted to using a diskette that had been formatted on one of these other machines. That is until now. Miklos Garamszeghy, a Toronto programmer and journalist has released a program into the public domain that allows the formatting of all the above formats except the Epson. The program, "128 MFM FORMATTER" runs in the native mode of the C-128, and is beginning to show up on bulletin boards all over the country.

Why use any of these formats at all when the C-128 double sided disks holds a hefty 336K? The obvious reason is storage! The Kaypro DSDD format gives you 376K of formatted storage in CP/M mode! Now you can have an extra 40K for programs, utilities, text files, or whatever. While the most obvious use of MFM formatting on GCR (Group Code Recording - the standard Commodore disk format) disks is heavy copy protection, the benefits of extra storage can be seen by everyone. Naturally, there is a drawback: You cannot BOOT CP/M off one of these diskettes - you must boot CP/M off of a standard Commodore formatted CP/M disk and then switch to the new format once CP/M is up and running. When you place one of these MFM formatted disks in the drive, it will attempt to read it. The system will then sense the type of disk via the number of bytes per sector and sectors per track. If the format is not unique, a highlighted box is shown in the lower-left corner of the screen. You are then required to scroll through the choices via the upper set of cursor keys. To read this disk once, simply press return.

I have noticed that this feature sometimes behaves erratically, as swapping disks will sometimes cause the drive to lock up when reading an MFM disk. At this point, I remove the disk from the drive, turn the drive off, and turn it on again. By this time, CP/M has usually returned me to the system with a BDO5 error and it will be safe to attempt reading the disk again. For more information on reading MFM disk formats in CP/M mode, see page(s) 705-708 in the Commodore 128 Programmer's Reference Guide. Mr. Garamszeghy's BASIC 7.0 program is listed below:

```

10 PRINT "MFM DISK FORMATTER BY M. GARAMSZEGHY"
20 PRINT "OPTIONS:";FOR I=1 TO 5:READ A,A,A,A$:
  PRINT I";";A$;NEXT
30 INPUT "SELECT A FORMAT";F:RESTORE (F*10+90):
  READB6,BS,SD,F$
40 PRINT "FORMAT >> ";F$;PRINT "PRESS <RETURN TO
  CONTINUE>"
50 PRINT "OR PRESS ANY OTHER KEY TO ABORT":
  GETKEYA$:IFA$<>CHR$(13)THENRUN
60 B5=39:S(0)=128:S(1)=256:S(2)=512:S(3)=1024:
  SI=0 :IFSD=2THEN SI=32
70 B1=700RSI:FOR I=0 TO 3:IF BS=S(I)THENB4=I:
  ELSENEXT
80 OPENI5,8,15,"UO"+CHR$(B1)+CHR$(129)+CHR$(0)
  +CHR$(B 4)+CHR$(39)+CHR$(B6)

```

```

90 INPUT#15,A:DCLOSE:INPUT"FORMAT ANOTHER <Y/N>";
   FA$:IFFA$="Y" THEN40
100 DATA 8,512,1,"IBM CP/M-86 SINGLE SIDED"
110 DATA 8,512,2,"IBM CP/M-86 DOUBLE SIDED"
120 DATA 10,512,2,"KAYPRO IV DOUBLE SIDED"
130 DATA 10,512,1,"KAYPRO II SINGLE SIDED"
140 DATA 5,1024,1,"OSBORNE SINGLE SIDED"

```

RAM EXPANSIONS AND CP/M

One of my recent acquisitions for my C-128 has been the 1700 RAM expansion. It is a very useful device, and the fact that CP/M supports it as a high-speed ramdisk (drive M:) is an even better reason to buy one. I've found several ways to increase productivity and increase the speed of things that previously have taken much longer without it. Using the MEX terminal program, I have downloaded programs and files directly to the RAM expansion, and then later dumped the contents of the expansion to my 1571 when off line. This really saves time, since the RAM expansion transfers data at 1 megabyte per second. The 1571, even with sector verification disabled isn't quite this fast! Another nice application I've been using the expansion RAM for is backing up diskettes. Instead of using virtual drive E: with PIP on my single drive system, which results in many disk swaps, I simply place all the files on the diskette I wish to backup onto the RAMdisk. Then, when I have a blank formatted disk ready, I dump the contents of the expansion ram onto the blank disk. Poof! Instant backup. Disk intensive programs work really nice with the 1700, though I recommend either buying a 1750 or upgrading your 1700 to a 1750 (see the July/August issue of Twin Cities 128 on the upgrade procedure) since 128K is really too small for serious applications such as dBase II and the like. Although the RAM expansion isn't supported as a RAM disk in native mode, the BASIC 7.0 commands FETCH, STASH and SWAP also make it easy for you to "roll your own" routines.

Another little-mentioned trick of the ram expansion is that unless the computer has been shut off or the ram memory emptied, the files you've placed there will stay there. Even after a RESET, you can check the contents of the expansion, your files are still there. The expansion ram unit comes with a 25 page manual that covers native and CP/M mode use of the device, plus a diskette with some mind-blowing demos on side a, and the newest release of the CP/M operating system on side two, along with some utility programs that allow for optimization of the CP/M side of the machine.

A LITTLE HELP WITH PIP

PIP is one of those transient utilities you will find yourself using a lot under CP/M. PIP stands for "Peripheral Interchange Program", although many people find themselves renaming it "COPY", which is its basic function. PIP can copy single and multiple files from one disk drive to another, or you can use a single disk and do swapping using virtual drive E: for this purpose. This is random access memory which is configured like an pseudo disk drive. PIP can copy files from an entire disk to another, copy a text file and send it to a printer or other output device. PIP can, in the words of the Digital Research documentation, "transfer data from a logical input device to a logical output device, thus the name Peripheral Interchange Program".

PIP can even rename a file or files after the

copying has been completed. But first, let's get down to basics with the program. To copy the files from one diskette to another using a single 1571, all you need do is the following (provided the PIP.COM program is on your diskette): PIP E:=A:*.*. Note the wildcard used for the file name and extension fields. You can even copy all of the COM files on one disk to another by doing PIP E:=A:*.COM. You will be prompted to swap diskettes by the system with highlighted inverse bars at the lower portion of the screen. You will note the files being copied will be printed on the screen as the process continues. This will be a somewhat slow and tedious process if you have a full CP/M disk in the drive.

For those lucky users with two drives, all you need do is the following: PIP B:=A:*.*. This will then copy all of the files from the disk in drive A to drive B. But what if you want to copy just a single file from one diskette to another? Simply substitute the filename and its extension for the wildcard like this PIP B:=A:FILE.EXT.

Renaming a file after copying is another neat feature. You can copy it to the same drive/usernumber, or a different drive/usernumber. But make sure you have enough storage on the disk for the file, as PIP has to have enough room for the new copy before it deletes the old one. This is accomplished like this: PIP B:newfile.txt=A:oldfile.txt. This copies the old text file from drive A: to drive B: with the name NEWFILE.TXT.

PIP even allows you to COMBINE files by concatenating them into one massive file. One thing you should note is that PIP will stop when it reaches a control-z in a file, which is usually the default end-of-file character. You should use the O option for copying machine code files in which a control-z might be a valid data item. Here's an example that lets you concatenate three files together: PIP BIGFILE=FILE1,FILE2,FILE3. The three smaller files should be on the default drive A.

You can also copy files to and from DEVICES in the system, but the file must contain printable characters. You can type a control-c to abort any PIP option. The logical devices you can use for PIP are CON:, which is your console input-output device (your keyboard), AUX:, which is any auxiliary input or output device, and LST:, which is the destination device assigned to the list output device (usually a printer). Here is an example of how to send everything you type to the printer (other than the usual control-p): PIP LST:=CON. You can terminate this by typing a control-z. You can also send whatever you like into a textfile like so: PIP A:TEXTFILE.TXT = CON: this will then send any text you type into a file. You end this with a control-z. Here is how you can send one of your files to the printer using pip: PIP LST:=A:BBSLIST.TXT[T8]. The T8 option expands any tab characters to the nearest column that is a multiple of eight.

PIP also can be used as a program. You merely type PIP without any options, and the CP/M 3 PIP VERSION 3.0 will appear. The command prompt is an asterisk (*). At that point, you can type any command line and press return like so:

```
*E:=A:MEX.HLP
```

```
*E:=A:CONF.COM
```

To return to CP/M, all you need do is then press an empty carriage return.

Question: Is the assembler for the C-128 going to be out soon?

DAVE: Some of the folk who brought you the C-128 system software have written a development package with a very good editor and two assemblers. The only question is how the package will be sold.

Question: What cable can I use with a Zenith 11 monitor and C-128 keyboard so that I might take advantage of the 80 columns on my C-128?

DAVE: I'm not familiar with the Zenith 11 monitor. If it supports RGBI, then you'll need a standard 9-PIN (IBM style) connector that'll hook up to the C-128 just fine. Otherwise, you'll only get monochrome, from the RGB connector's MONO pin, pin 7. Usually this is set up as an RCA type jack on the other end, center pin to 7, shield to ground (1 or 2 on the C-128).

Question: Why didn't the C-128 come with built-in software for the 80 column screen?

JEFF: We didn't really have time to develop this software, and it didn't seem that important at the time, since the 40 column screen is mostly for graphics, but if you are a serious application programmer, you still have this capability in hardware. Seems like a perfect niche for some enterprising guy to sell a ROM that plugs into this socket that gives you 80 column graphics.

Question: Are any new peripherals planned for the C-128?

DAVE: Yes, there's the 1581 3 1/2" disk drive (stores 800K), and a new mouse which is much smoother than the current mouse. I don't have any official dates on the release of either.

Question: When will the 1581 be released?

JEFF: We are trying for X-mas, but no guarantees. Tooling has started, but it is still quite a long lead time item. Software developers should be getting prototypes soon however, in Amiga 1010 cases. (soon = one month or so)

Question: Is the 1670 modem 1650 compatible in some respects?

JEFF: Yes and no. Transmit data and receive data along with +5v and carrier detect are all on the same lines. The control of the switch hook is completely different. The 1650 is a DUMB modem. The 1670 is Hayes compatible SMART modem. The 1670 will work with nearly ANY terminal program you can imagine, regardless of modem support. The 1650 must have special code in the modem program that knows about how to dial the phone etc. That's about it.

Question: Will this assembler be able to also do Z80 code?

DAVE: The assembler package consists of an assembler compatible with the C-64 assembler, and another very similar to the one we have on our VAX. No Z-80 assembler. The CP/M package you order from DRI has a Z-80 compatible macro assembler, and there are many others available in the public domain (I've got two myself).

Question: I can only get the eight CTRL colors on my Toshiba monitor. Is something wrong with my

monitor, computer, or cable? I am using an IBM-type cable. The Commodore colors only work in 40 column mode.

JEFF: Sounds like the "I" in your RGBI cable is not connected. This is what gives you the other 8 colors. If that doesn't work, your monitor is screwed up. I would bet that the cable is hooked up wrong though.

Question: Someone figured out how to activate the interlace mode on the 80 column chip. (The program is in the 128th Precinct.) However, only the first 25 lines are used on the display (which makes sense, since the OS is designed for 25 lines). Is it possible to copy OS ROM to RAM and insert a patch so that all 50 lines would be used?

JEFF: I think the problem is that you have just run fresh out of display RAM. If you would replace your 16K chips with 64K chips, the interlace mode should have enough RAM. No other patch to this that I know of. This is a hardware restriction to the 856.

Question: Is the first 128K of the 1750/1700 expansion cards directly addressable with BASIC 7.0 banks 2 and with bits 6 & 7 of the MMU configuration register? If not, how does the C-128 MMU handle the extra memory?

JEFF: The memory in the 1700/1750 is NOT REALLY in the memory map of the system. This is a DMA RAM that can be swapped with any area of RAM that is in the machine already. You give it a starting address and how many bytes and poof! It's loaded before your eyes.

Question: What's the word on the upgrade ROM for C-128 (to correct that "q" problem and other things...)? Will CBM provide upgrading through the RCA service centers?

JEFF: We are definitely working on the ROM upgrades for the 1571 and C-128. We still have a minor problem with the math package on the C-128 that we are trying to solve. As soon as this is solved, we will release it. We also have 1571 ROMs out to many of the software developers for the C-128. We are eager to hear their reaction to this new ROM. All things look very good so far, but I would really HATE to have to do this again a few months down the pike.

Question: Why doesn't CBM come out with an improved 80 column chip for the C-128 since it really wasn't really 'finished'?

JEFF: The person that designed this chip is no longer with us, and it would be VERY difficult to fix without him. He spent YEARS working on this beast, long before the C-128 was even conceived! Sorry, I wish I could tell you something else, but he left of his own free will. I suppose you would then complain about the flickering in 640x400 mode even after it was fixed. It has the same problems with interlaced mode as the Amiga. It simply follows the NTSC standard.

Question: Does something have to be done to the older C-128's so that the 1700 and 1750 expansions will work? I have been told a cut has to be made in the computer.

DAVE: That's probably the same problem you'd have using a 1700/50 with a C-64. The first 5000

C-128s, of which maybe 4000 (?) were actually sold on the retail market, have a slightly different DOT clock output than the later C-128s (more like the C-64s in some respect). The 1700/50 has a pull-up resistor on its DOT clock input. Removing this would improve things with an older C-128. Chances are about 4000/600,000+ that you have one of the earlier boards.

Question: Is it possible for the C-128 memory be expanded more than 512k?

JEFF: No not easily, but then, that's why Dave and I have a job here at CBM!!!

Question: In a recent issue of "the guide" magazine, an unnamed CBM engineer was quoted as saying CBM has/had developed a PC emulator cartridge for the C-128. It has an 8088 chip in it. Is this true? If not, is this possible? If so, will it ever be marketed?

DAVE: That's total bunk. There's no 8088 cartridge for the C-128, at least from CBM.

Question: About the 1581, could I use the 1581 for BBS programs to use as a file transfer drive, or do I have to change the program?

DAVE: You can use the 1581 for just about anything you can use a 1571 for. The exception is that a 1581 won't emulate a 1541; the disk format and related software are different. The 1581 would make a fine drive for a BBS, with its 800K storage and faster transfer.

Question: In regards to your 128 memory expanders, when will the "ramdisk" software for use under BASIC 7.0 be ready?

JEFF: We are working on this, but there is no planned release date yet.

Question: Any C-128 CP/M updates in the works?

DAVE: The only planned upgrade is for support of the 1581, I believe.

Question: Is there any truth to the rumor that the 1581 is running too hot and needs a new case? Also, why does CBM stuff in general run so hot?

DAVE: No, the 1581 is cool. It does need a case, currently the case tooling is the main delay factor in its introduction; the circuitry is rather simple. There are two reasons why CBM drives have run hot. First is that lots of things in them give off heat, the built-in power supplies and a large number of NMOS chips; most other computers have their drive intelligence supplied by the host computer. Second reason is that many user's don't allow the drives adequate ventilation.

Question: Any other printers in the making?

DAVE: A new one, called the MPS-1200, is coming out soon. Even better than the MPS-1000.

Question: I have 2 1541's (8 & 9) and 1 1571 (8). I only have 1 device 8 on at a time. I have them chained from C-128 to 1571, 1571 to 1541 (8), 1541(8) to 1541 (9), and 1541 (9) to printer interface. My problem is this: with everything hooked up my modem emits white noise from the speaker and will not operate. If I disconnect the

cable from the 1541 going to printer interface everything works ok. I have a switch on the power line to the interface and this happens with power on or off. Do you know of any other solutions other than unplugging the cable each time I want to use the modem?

JEFF: The modem is known to get confused when there are other devices connected that are not turned on - there is no real good solution to the configuration you are trying to set up.

Question: Will it be possible to get an 'update' to the 1670 when the new one's have the ability to NOT answer the phone at default?

JEFF: Sorry, there are no plans for an update - this is just an enhancement that will show up in production units at some unspecified future date.

Question: Is the 1660 a smart modem?

DAVE: Not really. The 1660 is a dumb modem, but it has the capability for intelligent action driven by the C-128 or C-64. The 1670 is truly intelligent, capable of autodialing without the help of the host machine.

Question: With the IIX to be released very shortly do you think CBM will begin moving towards a 65816 upgrade with 64/128 compatibility and some of that wonderful 16 bit speed?

DAVE: Well, that's possible. But the 65816 is not so wonderfully fast as you might have been led to believe. It's actually an 8 bit machine externally, like an 8088; and the 16 bit instructions are not all that powerful (no multiply). Plus, the 65816 is currently more expensive than a 68000. The latest rumors put the Apple IIX at a \$1600 list price, in direct competition with much more powerful machines like the Amiga.

JEFF: Yes - the Amiga is our 16-bit powerhouse - with coprocessors Apple isn't likely to get.

Question: Which monitor is better? I've heard the 1902 is better than the 1902a. Is it true? If so is Commodore still making the 1902?

DAVE: It may be nothing more than a matter of opinion. The 1902 has more standard connectors, the 1902a has a weird connector, but it adds a green screen mode missing on the '02. I'm not sure if we still buy 1902s or not.

Question: Is it true that Commodore is coming out with a 1680 1200BPS modem? If so why?

JEFF: YES, the 1680 is coming out very soon - it is nominally an AMIGA modem, but in reality is a standard RS22 interface HAYES compatible modem, that can be used for almost any application - PC's and the like!

Question: Can you upgrade the C-128 to 256K internally without 'losing' the '64 mode', etc ?

DAVE: No. The C-128s MMU can only physically address 128K of internal memory. Any internal addition to this would be some kind of kludge, and it wouldn't be supported by any C-128 programs.

Question: Any chance of a mass storage device for the C-64 or C-128, say in the 1mg or larger range.

How about the NEW 3.5" drive?

JEFF: The 1581 drive is around 1MB capacity and should be available later this year or early next year. Hard disks are still in the R&D stage, although some are available from third party vendors.

Question: What's an unimplemented command error in BASIC 7.0?

DAVE: BASIC 7.0 has a few unused tokens. Instead of making these illegal, they return the afore-mentioned error.

Question: You said something about the 1571 ROMS Are you writing them over also?

JEFF: No, we just fixed the known bugs - it isn't practical to rewrite the software, since there is too much software out there that thinks it knows what is in the ROM. You would have to decide what to break to permit your 'improvements'.

Question: What does upgrading video RAM to 64x4 give you?

DAVE: Upgrading to 64K (x8, by the way) of Video RAM will allow (1) Color 640x200 bitmaps, (2) 80x50 interlaced character displays, and (3) a somewhat broken 640x400 bit mapped mode. It will also allow for multiple screens to be stored simultaneously.

Question: Might Commodore think about upgrading the VDC's memory with 64Kx4 chips and great graphics software in the empty 2K ROM sockets?

JEFF: This is not likely at the present time. As pointed out earlier, this would be a nice niche for a small third party developer.

Question: Can additional internal RAM be installed in a C-128 to take advantage of the unused bank configurations?

JEFF: Sorry no, you can only do what the MMU chip provides for - it does not have any pins/signals to support additional RAM.

Question: Why haven't more games for the C-128 been released?

DAVE: There's no reason to release a game for C-128 mode that's the same as for C-64 mode. Only newer games that require the extra memory or new graphic features will show up for C-128 mode. A 64K program takes a LONG time to write.

JEFF: Remember that a vendor who writes a game that will only work on a C-128 is writing a product that he can't sell to the 90% of CBM customers who still have C-64's. For games, C-64 is still the way to go.

Question: When will the PC emulator using the 8088 be out?

DAVE: For the C-128, NEVER. This is fiction. For the Amiga, the SideCar IS a PC compatible, and it'll be out maybe in the fall.

Question: Can I record programs saved on the 1571 disks to a 1581 disks by loading 1571 disk first and then save it on the 1581 disk?

JEFF: You should be able to do this without any particular difficulty.

Question: Will getting rid of call waiting eliminate the problem of being knocked off the line?

JEFF: If you are being knocked off because of incoming calls, yes - some telephone exchanges now use a special dial-code to disable call waiting for the duration of the call - check with your local telephone business office.

Question: Are we going to see a C-256 or C-512 any time soon? Hopefully in the Apple IIx class?

DAVE: A C-256 or C-512 follow up will depend both on the success of the C-128 and of the changing economies of performance (like the 68000 that costs less than a 65816). We don't really know what an Apple IIx is yet, though it's likely to be as expensive as an Amiga and significantly less powerful. Really, though, there's no way to predict this; we have considered it.

Question: Can you speed up the clock beyond 2 mgz?

DAVE: No. The 2.04MHz clock rate is tightly controlled by the VIC chip and the 8701 clock chip. All of the memory and peripheral chips are timed tightly to run at this speed, which is actually faster than an 8MHz IBM PC as far as memory access time goes. The timing is also a factor in generating the VIC color; if its slightly changed, you don't get any color.

Question: Why in 80 columns do some of the colors change colors? Like Orange= Dark Purple

JEFF: Monitor limitations - it takes a high dollar monitor to do 80 column color flawlessly - you might take your monitor to a radio/tv shop and have the convergence checked and perhaps have the monitor degaussed. On some it is accessible, however CSA (Canadian Safety Assoc.) requires that there be no user serviceable parts in the power supply - so, newer units do not have accessible fuses.

Question: If new 1571 ROMs, any chance of fixing second side write to speed it up?

JEFF: Yep. This is definitely fixed.

Question: I have my 128 interfaced to an IEEE bus via a Quicksilver and I'm concerned as to the likelihood of the 1750 and the QS getting in each other's way. Have you any thoughts on the subject?

JEFF: I don't think that they should interfere, as long as QS stay away from the DMA line.

Question: I'm sure it does, Jeff. The 128D IS for sale in England, why not here?

JEFF: In the infinite wisdom of CBM-North America. It costs too much.

Question: CBM Have any plans for hard drive support?

JEFF: We have a design, but we are waiting for someone upstairs to say GO. Hard disks are a problem. Let me explain. You can get a HD for

Fastload Modification for use with the C-128

by: Bruce Jaeger

bad day. A large portion of this is the actual cost of the HD, and it is difficult to make any profit on these things, since they are already cutting the price so low and HD makers are selling directly to computer stores. So, we would have to invest significant \$\$\$\$\$\$ in the mere HD alone, not to mention the interface, which by the way is the same custom chip as in the RAM expansion.

Question: We have talked some about the 1581. Are we, as consumers, ever going to see it, or will it go the way of the 1572, LCD, etc.

JEFF: EVERYONE LIKES the 1581. It is a GO project, from the top.

Question: Any plans for a new 1670 upgrade, or another Hayes-compatible modem that is more reliable?

JEFF: We do have a plan for an upgrade for the 1670, but I am not sure when this new version will get phased into production. Probably later this year, but it depends on what the demand is for the product. The changes are pretty small, but it's what you guys have been asking for. MORE HAYES COMPATIBILITY! It won't answer on the first ring anymore. It will have call progress monitoring. It will have more S-register implemented. It will implement ATH and other Hayes codes that got left off before. That's about it, it is VERY HAYES compatible.

Question: What are the differences in the 128 ROM upgrades and what is the status of the 65816 machine Dave Haynie was working on?

JEFF: C-128 upgrade roms fix some round off errors in the math package like $25=32768$ not 32768.000000000001 or whatever. Also the caps lock q bug. Some other small ditties like that. Nothing major. Haynie's 65816 machine is on the back burner until the Yen decreases again. Haynie's '816 machine is on the back burner until someone really asks for it. We just can't make computers that cheap anymore with the Yen the way it is. The electronics industry has been hit hard, whether you know it or not. I personally like the idea, but that doesn't always hack it. It may surface again if Apple introduces a machine with the '816 in it. I know we could beat them on price. Haynie's machine was going to (sell for) about the same as a C-128 is today, but with a better microprocessor and more ram, but no 64 mode.

Question: I'm confused, Jeff. You are working on a RAM disk wedge?

JEFF: Yes, for SEQ and PRM files only. It will also handle the error channel like a real disk.

Question: The MMU in the C-128 appears to be able to address 256k DMA, any comments? Could the system therefore be modified?

JEFF: Yes, originally there were plans for 256K, but I not sure what happened. It would require significant software mods, the hardware is easy.

Question: Which machine do YOU use the most!

JEFF: The C-128.

We don't like to talk much about the C-128's Commodore 64 mode in Twin Cities 128. Occasionally, though, we realize that there will always be more C-64's than C-128's (because every C-128 sold also adds to the C-64's total!) and that the software publishers will almost always opt for the most common machine. Hence, we often have to lobotomize our C-128 down to 64K, and take the 1571 Indy Car-speed disk drive, install a lawn-mower engine with bad rings and a shot magneto, and call it a 1541.

Epyx's Fast Load cartridge is a pretty effective product for getting those Commodore 64 games and utilities to load faster. While not all programs can be loaded with it (due to "protection" schemes that further confuse the poor 1541), the Fast Load cartridge will load most programs much faster than the standard DOS, with the added bonus of making the famous "DOS Wedge" disk command shorthand available to the 64 user, as well as several other utilities.

On the minus side, to use the Fast Load cartridge with the C-128 means that you have to install the cartridge when you go to C-64 mode, then remove it to go back to 128 mode. If you're a software developer (or just an avid C-128 user) this is going to mean a lot of wear and tear on the cartridge port, as well as frustrating waits for power-up if you turn off the computer between cartridge changes (as you're supposed to), or risk frying a chip or two by yanking the cartridge in and out under power (as anyone with a normal human's patience tends to start doing eventually!)



FIGURE 1



FIGURE 2

Fortunately, there's an easy way out, and you don't have to be overly handy with a soldering iron. By putting a switch in the EXROM line between the Fast Load circuitry and the circuit board pads, the C-128 can be made to ignore the presence of the cartridge. Open the switch, and the C-128 will come up in 128 mode, because the cartridge can't assert the EXROM line to the cartridge port. Close the switch, and the C-128 will come up as a C-64 (after power-up, or simply pressing the reset button), with the Fast Load utilities enabled. And as a little extra bonus, the Fast Load cartridge gets you to C-64 mode a lot faster than "GO 64" or the reset-with-Commodore-key methods!

Here are the simple step-by-step instructions for adding the switch to your Fast Load cartridge. Not that the photos may look slightly different from your own Fast Load; that's because I made the modification to mine, then unsoldered everything (once I knew it worked!) to take the photographs.

STEP 1: OPENING THE CARTRIDGE:

Slit the paper cover where it wraps around the back of the cartridge. Pick away the paper from the screw holding the two cartridge halves together (Figure 1). Unscrew the screw, using a small Phillips-head screwdriver, and carefully pull the two cartridge halves apart.

STEP 2: DRILL THE SWITCH MOUNTING HOLE:

There's plenty of room in the top portion of the cartridge (near the Epyx name) for a small SPST toggle switch. (SPST means "Single Pole, Single Throw." It's the cheapest, least confusing type. Get a small one at Radio Shack.) Using a drill appropriate for mounting the switch you have on hand (mine was 1/4"), drill a hole in the top half of the cartridge where you want the switch to go. (See Figure 2.) If you like, you can install the switch now in the cartridge top; it will make it easier to solder to later.

STEP 3: CUT THE EXROM TRACE:

The trace (that's electronics talk for those little copper paths on the circuit board) that we want to cut is the EXROM line. No, it's not labelled on the circuit board! Looking down at the board with the PC pads (that connect to the 128's cartridge port) towards us, count nine pads from the left; that's the EXROM line. (Also see Figure 3.) Using an X-Acto knife or similar tool (CAREFUL!), cut away at the trace until you are just through the copper, and it is no longer a good connection from the pad to the 7407 chip. Don't make your cut too near the chip, or too near the edge pad, as we'll need room to solder wires leading to our switch.

STEP 4: SCRAPE INSULATION:

You've probably noticed that the copper trace was coated with some translucent green varnish-like stuff. We can't solder to that, so use the X-Acto to carefully scrape off the varnish on either side of the cut we made.

STEP 5: SOLDERING WIRES:

This is probably the trickiest step, because the tiny traces aren't really designed to be soldered directly to, and can be pulled away from the board if we aren't careful, or use too much heat. If you're at all unsure about soldering in close quarters, get someone with more experience to help you.

First, find two 3" long pieces of thin, flexible wire. (Remember, it's got to be tucked up inside the cartridge with your switch, and the whole mess screwed back together!) Remove about 1/8" of the insulation from each end of each wire. "Tin" the ends by heating the wire with your soldering iron, and apply ROSIN-CORE solder until the ends of the wires are shiny.

Now, carefully tin the exposed copper of the scraped traces, being extra careful not to flow solder into the gap you laboriously cut! DO NOT do this soldering on the circuit board with a big, clumsy soldering gun -- they provide too much heat, which could buckle the board or make the trace pull away. (And if you even considered using a torch to do your soldering, please sell your computer and go get a job as a tornado-rod salesman.)

Tinning your wires and traces will make soldering them together easy. Lay the exposed end of a wire on top of a tinned trace, press the iron on top of them and Presto!, they're soldered together. Repeat for the other trace.

Figure 4 shows my wires soldered in place. They're not so neat, because I reheated my joints several times in installing and removing the wires for the photos. Do yours once!

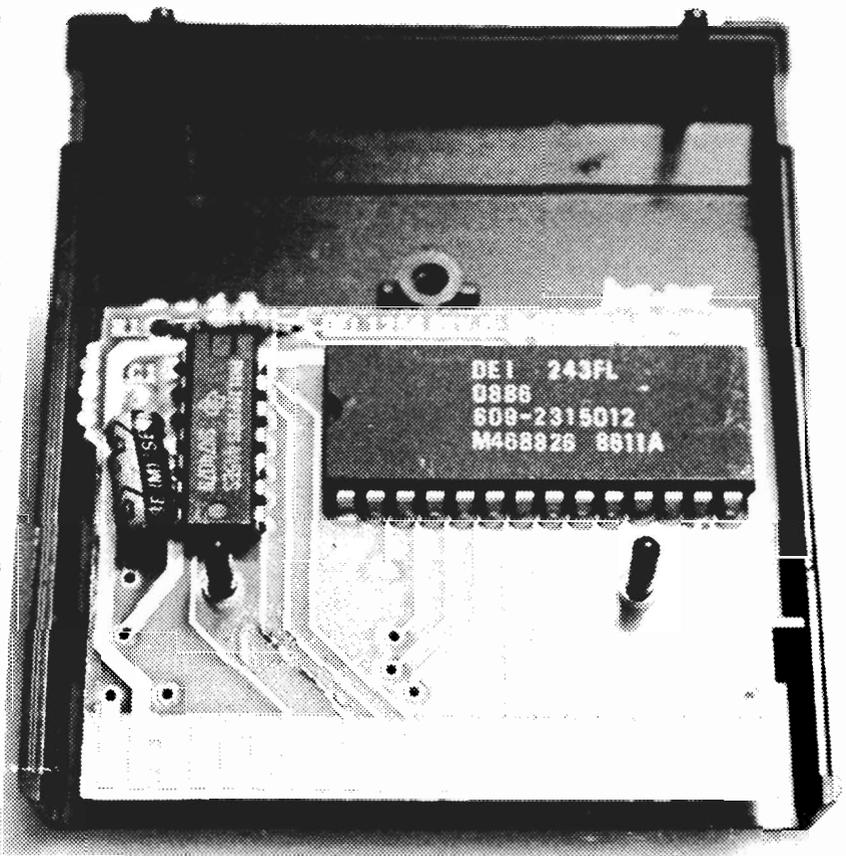


FIGURE 3

STEP 6: SOLDER THE WIRES TO THE SWITCH:

This is easy, unless you made your wires a bit too short! Just "tin" the terminals on your switch, then solder the other ends of the wires to the terminals. (See Figure 5.) It's not overly important which wire goes where, as long as the connections are good.

If you haven't already mounted the switch to the top half of the cartridge box, do so now.

STEP 7: PUTTING IT ALL TOGETHER:

CAREFULLY put the two cartridge halves back together, taking extra care that your wires don't stick out the sides. Be especially careful that you don't inadvertently pull the wires off their soldered connection to the traces--that's a very fragile connection!

When everything fits, screw in the locking screw (Figure 1) and tighten. Now for testing!

STEP 8: TESTING, LABELLING THE SWITCH

Turn OFF your computer. OFF! Now plug in the cartridge. If you are using a 1902 monitor, turn it to 40-column mode (so we can see the C-64 mode

screen if it comes up).

Turn on the computer. It will come up in either 128 or 64 mode. Flip the switch in the opposite direction, and press the reset button on the side of the C-128. The computer should now come up in the opposite mode. Label your switch appropriately.

TROUBLESHOOTING:

If, for some reason, the computer keeps coming up in C-64 mode, you've got a "short" in the new wiring somewhere. Make sure solder hasn't bridged the gap you cut in the trace, and that the wires are distinct and separate all the way to the switch terminals and back. (Also make sure, of course, that it was the EXROM trace you cut and installed wires on! Refer to Figure 3 again to make sure!)

If, on the other hand, the computer keeps coming up in C-128 mode, no matter where you set the switch, it means that the switch is not re-connecting the cut trace. Check for bad solder joints, a broken wire (they can do that sometimes, without you suspecting it) or for a bum switch if you pulled it out of your scrap box.

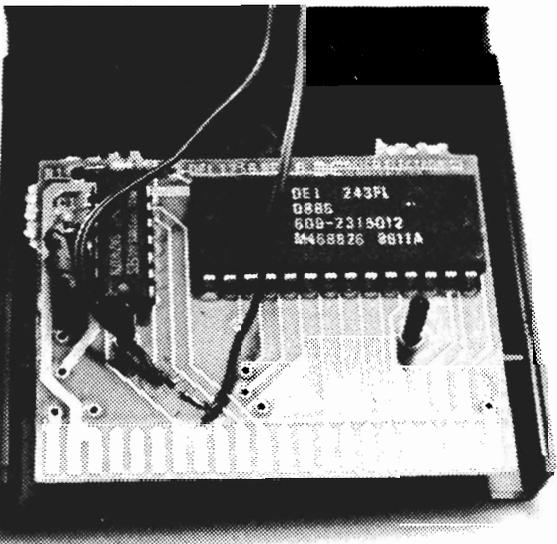


FIGURE 4

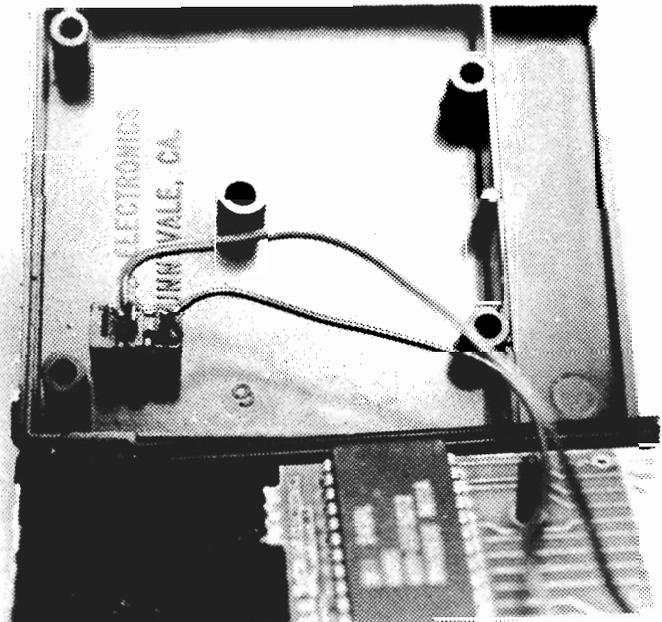


FIGURE 5

Last month, caught in the fury of TECHNOJUST, I told you a little bit about my new 1750 RAM expansion. This month it is time to "get down to business" with the 1700/1750 RAM expansion by exploring how to use them in and with our BASIC 7.0 programs.

First off let's try to better understand exactly how BASIC 7.0 uses its own internal random access memory and accesses the memory in your RAM expansion. As I said last month the RAM expansions give you indirectly accessible random access memory. This means that the operating system of your computer does not control this random access memory in the same way it accesses and allocates the 128k of random access memory installed within your C-128. The operating system of the C-128 can only "see" (directly access) the C-128's internal random access memory. This memory is divided into two 64k sections called BANK 0 and BANK 1. The C-128's BASIC programming language uses these 64k banks for two distinctly different purposes. BANK 0 is used by BASIC for the storage of various system information and data and the text of your BASIC programs (the vast majority of this space is reserved for the text of our BASIC programs). BANK 1 on the other hand is used for the storage of variables in your BASIC programs. We will discuss which sections of these banks are particularly useful later in this article.

Since the C-128's operating system can only "see" the contents of the memory in BANK 0 and BANK 1, the designers of the C-128 have given you the capability to copy or replace part of or the entire contents of either BANK to/from the external RAM in the expanders. This is accomplished through the use of three BASIC 7.0 commands specifically to transfer data between the internal memory of your C-128 and the external memory of your RAM expansion.

The STASH command copies a specified section of the C-128's internal memory (BANK 0 or BANK 1) to a specified part of the external RAM expansion memory. This copying process does not disturb the contents of the internal memory. The FETCH command performs the exact opposite function of the STASH command, copying the contents of a specific part of the external memory to a specific section of BANK 0 or BANK 1. Like the STASH command the FETCH command does not disturb the source memory for the copy, in this case the specified section of the expansion memory. The SWAP command exchanges a section of internal memory with external memory. This means that with the SWAP command both internal and external memory ARE altered. The parameter list for each command follow exactly the same pattern, namely:

NUM BYTES, INTERNAL LOC, EXTERNAL LOC, EXT. BANK

Where: NUM BYTES - is the number of bytes involved in the operation, INTERNAL LOC - is the beginning location in BANK 0 or BANK 1 (BANK 0 is the default) involved in the operation, EXTERNAL LOC - is the beginning location inside the memory expansion involved in the operation, EXT. BANK - is the bank inside the RAM expansion where the operation is to take place.

After looking at the above parameter descriptions you may have wondered what all of that means. Let's consider a specific instance such as:

SWAP 1024,1024,0,0

This SWAP command tells the C-128 to exchange 1024 bytes (parameter 1) beginning at location 1024 in BANK 0 of the C-128's internal memory (parameter 2) with the expansion memory beginning at location 0 (parameter 3) in bank 0 in the expanders. The last parameter is necessary because the memory inside the RAM expansion units is organized into 64k sections, sometimes also referred to as banks (be careful not to confuse these external "banks" with the internal BANK 0 and BANK 1). The 1700 RAM expander is organized into two 64k banks numbered 0 and 1, while the 1750 RAM expander is organized into eight 64k banks numbered 0 through 8. That's all there is to it.

About this time, I envision many of you scratching your heads and saying, "Okay, I understand the principles behind utilizing the RAM expansion from BASIC 7.0, but I am still not sure how to actually do the NEAT stuff?" The answer to that question lies in your understanding of the C-128's architecture or your ability to utilize various reference materials on the C-128's architecture. In order to use the RAM expanders effectively in your own programs you must be able to locate the areas in the C-128's internal memory which store or control various operations that you may wish to store and later recall from the RAM expander. This is best done by examining a detailed memory map of the Commodore 128 and studying various explanations of how things work on the C-128. Perhaps the best source for this kind of material is the Commodore 128 Programmer's Reference Guide published by Bantam. But here are some handy memory locations and ideas to get you started:

(all memory locations are in BANK 0 unless specified differently)

40 column text screen (Location 1024 - 2023)

Transfer 1024 bytes beginning at 1024.

(example: STASH 1024,1024,0,0)

Idea: Multiple instantly accessible help screens

40 column graphics screen (Location 7168 - 16383)

GRAPHIC screen must be allocated.

Transfer 9216 bytes beginning at 7168.

(example: STASH 9216,7168,0,0)

Idea: Multiple graphic screen animation

Comments: GRAPHIC screen must be allocated.

Background color: 7168 - 8191 Bitmap: 8192 - 16383

Function keys (Location 4106 - 4351)

Transfer 245 bytes beginning at 4106.

(example: STASH 246,4106,0,0)

Idea: Instant function key overlays for menus

Comment: Transfers all function keys regardless of definition or lengths

Sprite definitions (Location 3584 - 4095)

Transfer 512 bytes beginning at 3584.

(example: STASH 512,3584,0,0)

Idea: Super animation with multiple definitions

Comment: Use of multiple backgrounds with multiple moving sprite sets would produce fantastic results.

BASIC Program text (Normal Location 7168 - 65279)

(Location w/graphics 16384 - 65279)

Transfer number bytes necessary for BASIC text.

The end of BASIC text can be found as follows:

EB=(PEEK(4625)*256)+PEEK(4624)

So sample transfer might look like this:

STASH EB-7168,7168,0,0

Comment: This is the technique one would use for storing multiple BASIC programs in the expansion. First you load the program into memory using the DLOAD command, then find the end of the BASIC text in BANK 0 using the formula above and STASH it in the RAM expansion. Be sure to keep track of the location and bank where your programs are stored so they can be recalled.

BASIC Variables storage in BANK 1

This is a little tricky because the STASH, FETCH, and SWAP commands are designed to operate on BANK0, although the operating system can be convinced to let you access BANK 1 with them. The method involves manipulating bits 6 and 7 of the MMU's RAM control register. These bits are used to control the C-128's VIC RAM bank pointer as well as the C-128's DMA bank pointer. To find the location in BANK 1 of specific variable use the POINTER function. The following example sets the register so we can access BANK 1, performs a STASH, and then returns the default bank for the STASH, FETCH, and SWAP commands back to BANK 0. It is important to return these bits back to their original values after performing a STASH, FETCH, or SWAP or your 40 column screen will display garbage.

10 POKE 54534,PEEK(54534) OR 64:REM set to bank 1
 20 STASH1024,21,0,0: REM perform stash
 30 POKE 54534,XOR(54534,64):REM set to bank 0

TWIN CITIES 128

Available in Minnesota at:
 Computer Stuff
 Computer Exercise World
 Computer Outfitters
 Valiant, Inc.

Available throughout
 the rest of the universe

\$ 2.10 per single issue
 \$22.00 per subscription (12 issues)

Send your check to:

Twin Cities 128
 P.O. Box 4625
 Saint Paul MN 55104

*Dealer and investor inquiries
 welcome*

Providing quality information since
 January 1986

September Specials

Generic DS/DD 5.25" Floppy Disks BOX OF 10
 REGULARLY \$10.95 **\$ 7.99**

Monitor Stand Regularly \$18.95 NOW JUST **\$ 9.99**

Now in Stock

Rebel Assembler Package Both C-128 and C-64
 versions included for one low price! **\$ 26.99**

Gnome Speed C-128 BASIC Compiler **\$ 51.99**

Gnome Kit Programming Tool Kit Both C-128 and C-64
 versions included for **\$ 35.99**

Computer Stuff
 The Twin Cities Commodore Specialists

4606 Chicago Avenue S.
 Minneapolis MN 55407
 612-825-2674

Music on the Commodore 128 is much easier than on any previous 8-bit microcomputer. The Sound Interface Device, usually referred to as the SID chip, plays three music sounds simultaneously, with a wide variety of tonal variations available. In fact, with the PLAY, SOUND, FILTER, and ENVELOPE commands readily available from BASIC 7.0, you can re-create any sound or noise that you can hear. Granted, the more difficult the sound (like a dog barking, a human voice, or the wind) takes great skill, and is much easier on some other microcomputers and synthesizers. But, working with it can prove to be quite entertaining and rewarding.

The C-128 has 10 predefined sound settings, which resemble 10 different instruments. These resident sound settings are the following, with the instrument the emulate: 0=piano, 1=accordion, 2=calliope, 3=drum, 4=flute, 5=guitar, 6=harpsichord, 7=organ, 8=trumpet, and 9=xylophone. To hear an example of each of these sounds, type in the following program and RUN it:

```
10 FOR N=0 TO 9
20 A$=STR$(N)
30 A$="V104T"+A$+"U8X0"
40 PLAY A$
50 PLAY "V104QCDEFGAB05QC"
60 NEXT N
```

This will play a standard octave 9 times, each time with a different predefined sound.

Since humans describe sound using examples of sounds around us, it was fairly easy for you to anticipate what each of the those sounds would be like, even before running the program. However, the computer must use different methods for describing sounds. The C-128 defines a particular sound with ADSR settings, which refer to Attack, Decay, Sustain and Release, as well as the shape of the waveform, and the pulse width. All of these settings are incorporated into the ENVELOPE command, which will redefine any of the preset values for a particular sound to the new values you assign it. The format for the ENVELOPE command looks like this:

```
ENVELOPE n,[a][,d][,s][,r][,wf][,pw]
```

where:

n is the envelope number (either a preset or a new setting)

a is the attack rate (how fast the sound reaches its peak volume level, described by a number from 0-15)

d is the decay rate (the speed that a tone falls from its peak volume to the sustain level, again described by a number from 0-15)

s is the sustain rate (the volume level that a note can decay to, described by 0-15)

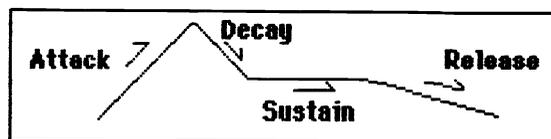
r is the release rate (the length of time a note is held at the sustain level before being dropped to zero, again 0-15)

wf is the description of the wave form:

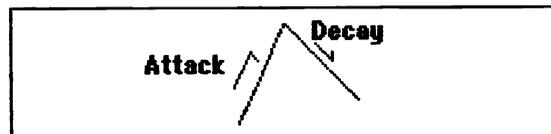
- 0=triangle
- 1=sawtooth
- 2=variable pulse (square)
- 3=noise (white and pink)
- 4=ring modulation

pw is the pulse width (the length of time between notes for a variable pulse wave)

To get an idea of what this looks like graphically, take a look at this chart, which exemplifies ADSR levels all at average settings. The line across the bottom represents the amount of time that has gone by.



Let's take a look at the most common envelope, the piano. The envelope setting of the piano is as follows: ENVELOPE 0,0,9,0,0,2,1536. The first "0" tells the computer which preset we are going to be altering. The second "0" is the attack. This would be a very fast attack, a sharp strike. The "9" is the decay, and this is not a fast drop-off (0) or a slow one (15). The next "0" is the sustain, and the zero means that the note is not sustained but instead decays to zero (very good for staccato, or clear-cut, notes). The last "0" is the release rate, which in this case is released immediately. Here is a picture of this particular envelope's ADSR settings.



The "2" in the envelope setting above describes the wave form, here a square wave. This is probably the most confusing description of sound to beginners. The word "square," "triangle," and "sawtooth" all describe how the tone looks when displayed on an oscilloscope, a tool used to examine waveforms of various types. The square wave generally describes the sound of a percussive string or wind instrument, a sawtooth wave usually describes the sounds of reeds and picked stringed instruments (guitar, double bass), and the triangle wave is a purer, cleaner form of the sawtooth wave, that describes the clearer instruments like flute, calliope, and cello. The primary thing to remember when describing a tone with one of the terms above is that most instruments are combinations of the different wave forms. (To see ADSR settings for the other preset envelopes, check page 142 of your System Guide.)

However, this is just the beginning. Now that you understand some of the principles behind the ENVELOPE command, you can use this to create your own sounds. Let's try experimenting with the different ADSR settings, to hear the effect of changes made. First, in direct mode, type in the following command:

PLAY "TOV1AAA"

This will play the note "A" three times with the preset piano envelope. The "TO" tells the computer to use envelope 0, but you could substitute any number 0-9. Next, let's reset this envelope to create a different sound. Type in (again in direct mode) this command:

ENVELOPE 0,12,0,0,0

This envelope will reset the "0" preset to a note with a slow moving attack, which will reach its maximum volume and then fall immediately to zero. To hear this note, retype the PLAY command. The sound created is a kind of "whoop" sound. The opposite of this, of course, would be an envelope with no attack and a large decay. Try typing this:

ENVELOPE 0,0,13,0,0

This envelope creates a different kind of noise,

an almost "ping." (You'll have to retype the PLAY command again to hear it.) The same kinds of experimentation can be tried with the sustain and release levels, and different combinations of the four ADSR settings. The more you attempt different levels for these rates, the more you can see how these settings affect how a note sounds.

While you are trying different settings, you might try changing the waveform also. For instance, setting the waveform at "3" will create white noise, which, if you check the settings of the predetermined envelopes, you will find that "3" is used for the "drum" setting. You will also see that both the guitar and the accordion have a sawtooth waveform, whereas the trumpet and the organ both have a square or variable pulse waveform, although their pulse widths are radically different.

Try fooling around with the ENVELOPE command, as well as the PLAY command, and next time I will show you how to make the FILTER work for you.



COMPUTER OUTFITTERS



OUR PHONE LINES ARE OFTEN BUSY, SO PLEASE KEEP TRYING!

2730 Snelling Ave. N. Roseville

631-1763

Forgotten BASIC

Exploring little known areas of
BASIC 7.0 on your C-128

by Loren Lovhaug

Sept '86 Twin Cities 128

Technical/Page 26

This month's installment of "Forgotten BASIC" explores another aspect of the BASIC 7.0 that while offering a great deal of power and potential to BASIC programmers is often ignored.

Many maids and housekeepers refuse to do windows but that is no reason why C-128 programmers should not do WINDOWS. The WINDOW command provides a very nice method for visually organizing output to the screen. The WINDOW command is used to specify the position and dimensions of the area in which output can be placed onto a text screen. The WINDOW command has five parameters which are as follows:

WINDOW X,Y,A,B,C

X represents the upper-leftmost character position of the area in which text is to be displayed. When the forty column screen is active X may range from 0 - 39; however, when the 80 column screen is active X may range from 0 - 79 (zero represents the absolute leftmost character position). Y represents the line ranging from 0 at the top of the screen to 24 at the bottom on which the upper-leftmost character of the display area may be placed. A and B are similar to X and Y respectively except that they represent the horizontal and vertical positions of the lower-rightmost position of the display area. The last parameter C, indicates whether or not the newly defined display area should be cleared or not. (1 clears, 0 does not clear).

The WINDOW command is destructive, so it cannot be used for true overlaying pop-down menus, but once a window is defined text will scroll within the defined display area. Another interesting aspect of the C-128's WINDOW facility is the fact that BASIC 7.0's CHAR command adapts its display coordination to the currently defined WINDOW. Type in and study the following example to get a better understanding of the WINDOW command and some the abilities of the CHAR command:

```
100 ws=RWINDOW(2)-1
110 DEF FNr(x)=INT(RND(1)*x)+1
120 t$="Watch these windows get dirty!"
130 FOR i= 1 to 10
140   x=FNr(ws-1)
150   y=FNr(12)
160   a=x+FNr(ws-x)
170   b=y+FNr(12)
180   FOR j= 1 to 5
190     WINDOW x,y,a,b,0
200     CHAR 1, FNr(RWINDOW(1)), FNr(RWINDOW(0)), t$, 1
210     SLEEP 1
220     WINDOW 0,0,ws,0,1
230     a$=""
240     DO WHILE a$=""
250       GET a$
260       PRINT"press any key!  ";
270       LOOP
280     NEXT j
290 NEXT i
```

This program upon execution will place the message: "Watch these windows get dirty!" in various places on the screen. This is how the program works:

Line 100 uses the function RWINDOW to decide whether you are executing this program on the 40 or 80 column screen. RWINDOW returns various information about the current display area depending upon the argument you provide. If you use an argument of zero with the RWINDOW function, the function will return the number of vertical positions in the current display area. If you use an argument of one, the function will return the number of horizontal positions in the current display area. And as we see in line 100 an argument of two returns the display mode we are in (40 or 80).

Line 110 creates a function which chooses a pseudo random number between 1 and whatever number is used as the argument of the newly created FNr function. Line 120 simply defines the string variable T\$ to our message. Line 130 defines the primary loop in this program. This loop in reality keeps track of how many windows we have placed on the screen. Once we have placed ten windows and their messages on the screen this program terminates. Lines 140 through 170 are used to define randomly chosen dimensions of each new window. Line 180 defines a nested loop which will control how many messages are placed in each window. The WINDOW command in line 190 defines the current display area to the one chosen in lines 140 through 170. Interestingly enough this WINDOW command will also allow us to "jump between" windows as the "j-loop" executes. The CHAR command in line 200 places our message in a random position within the defined window. You might notice that the message is printed in reversed video, this is because the reverse video flag is set to one (note the last parameter of the CHAR command). You might also notice how the text occasionally scrolls. Line 210 defines a new window which will be used for a one line travelling message (this is the window we "jump between" as the "j-loop" is executed. Line 230 defines the string variable A\$ as a null string so we may use it as a condition in the DO loop beginning in Line 240. This loop continues to display the "press any key message" in the one line display window until a key is pressed. When a key is pressed the program continues until all loop conditions are satisfied.

While the above program does not seem very significant, it does demonstrate some of the neat things you can do with the C-128's WINDOWing capabilities. Why not try them out in your programs?



Pssssst!
Hey buddy, I've
gotta tip for ya..

Welcome to this month's addition of Sparrow's Slick Tips. Our resident hacker has been on the road as of late visiting various Commodore computer gatherings around the country, compiling valuable C-128 information for you to use to take your C-128 farther, faster. If you have an item you think would make a good Slick Tip, send it to Sparrow James, P.O. Box 4625, Saint Paul MN 55104.

00101010: INDENTED LISTINGS

It is often desirable to indent your BASIC program listings to make the program's structure more apparent. However, this cannot simply be done by adding blank spaces between a line number and the command on a particular line since the BASIC parser will "throw out" your "unnecessary spaces". Fortunately there is a way to easily create indentations. Simply put a graphic character (the characters printed on the front edges of your keyboard) following the line number in your program and put as many blank spaces as you like and type your command and press return. When you list the program you find that the graphic character will have disappeared but your indentation will remain intact.

00101011: BASIC MERGER

As I have said many times in this column, the C-128's BASIC 7.0 is one of the richest BASICs available for any microcomputer. But there is one feature noticeably absent: the ability to merge program files from disk. With this capability you can write a set of often-used generic subroutines (sometimes referred to as "primitives"), save them to disk and later merge them into other programs, thereby saving the "reinvention of the wheel syndrome". Many authors have attempted to simulate a program merge on the C-128 by loading a program, then moving the start of BASIC pointer to just beyond the end of that program in memory, which will convince the C-128 to load the second program in just after the first in BANK 0. Then, once the second program has been loaded into memory, the start of BASIC pointer is moved back to the beginning of the first program thus convincing BASIC that you now have a longer "merged" program in memory. This is all well and good, except that this type of merge is often incredibly inconvenient. The biggest drawback this type of merge requires that the subroutine you are merging be numbered with line numbers greater than those of the first program in memory. If you don't obey this rule, BASIC will do some very strange things with your new pseudo-merged program. Fortunately there is a better method:

1. Save the program or subroutine you wish to merge to another program to disk as a sequential file like this:

```
DOPEN#1,"filename",W:cmdl:list:print#1,"DCLOSE":print#1,"":DCLOSE
```

2. Load the program that you wish to attach the above program to.

3. Type the following and press return:

```
DOPEN#1,"filename":SYS 65478,0,1:LIST
```

You will now have a perfectly merged program (you should see it on the screen). Don't believe me, LIST the program and see for yourself! Here is how it works: Step 1 saves your BASIC program listing to disk as a sequential file, and then adds the command DCLOSE at the end of the file. By putting this entire sequence on one direct mode line we will avoid having the "READY." message placed in our file at the end of the list. Step 2 brings in the program you are about to append to. Step 3 opens the sequential file and calls the kernel routine CHKIN which in this case tells the computer to use logical file #1 as an input device (the default input device is the keyboard). Then the program is listed, but since the DOPEN'd file is the input device your listing is in effect "typed in" from disk, thereby merging it with the existing program in memory. The DCLOSE we placed at the end of the file in step 1 in turn closes the file automatically for you. This method for merging programs is much more convenient because it allows you to merge your routines freely without having to make sure that their line numbers are greater than those of the program you are merging to. (One caveat however, this merging technique will also replace lines in the program with lines for the "merge" file if they have the same line number.) Another side benefit of this merge technique is that it allows you (if you so choose) to use a word processor to edit BASIC programs or subroutines. Consider altering programs with the powerful search and replace and macro capabilities of most C-128 word processors.

00101100: CREATING A NEW DEFAULT PRINTER DRIVER WITH GEOS

Many GEOS users have expressed to me sheer aggravation over the apparent inability to set up your own default printer driver. It is irritating to have to select your printer manually each time you begin a work session with GEOS. The GEOS manual suggests vaguely that GEOS will select automatically the first printer driver on the diskette, however we have found this only to be true with the BOOT disk and NOT your individual work disks. So, the solution is to rearrange your boot disk so that the printer driver of your choice is first on the BOOT disk and then your printer driver will be selected as the system's default printer driver.

00101101: RESTORING YOUR FUNCTION KEYS

It is often desirable to redefine the C-128's programmable function keys for use in your BASIC programs. However many programmers have a bad habit of not restoring the function keys back to their default values before the program terminates. This is unfortunate since it can very easily be done without even having to go through the drudgery of having to use the KEY command eight times. All you have to do is execute this simple loop:

```
FOR X= 0 TO 76: POKE 4096+X,PEEK(52904+X):NEXT X
```

This loop reads the default definitions from the C-128's read only memory and POKES those values in the function key buffer.

Commodore Owners

C-128, C-64, SX-64, Plus/4

Learn how to program your computer in BASIC I

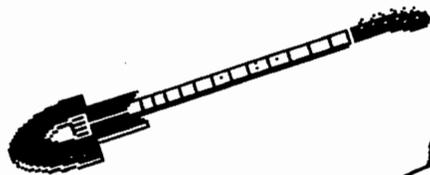
Beginning BASIC programming for Commodore microcomputers.

8 consecutive weeks beginning Wednesday, September 24 1986
6:30 - 9:00pm at Como Park Senior High School Only \$25.00*

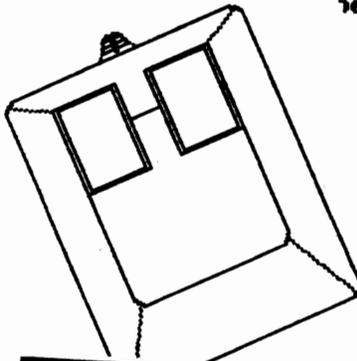
*plus materials available at Computer Stuff on September 15, 1986

Call Como Community Education at:

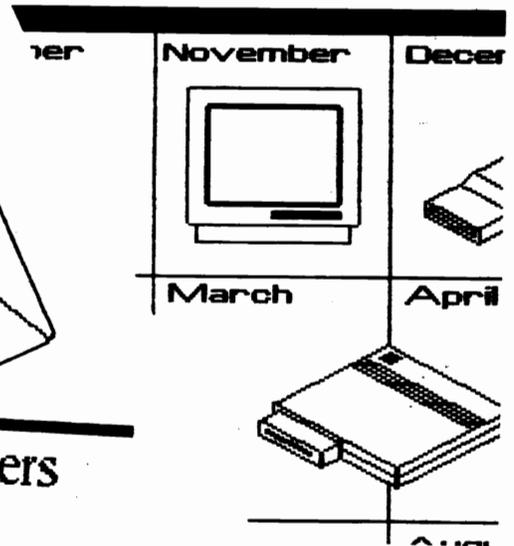
489-4523



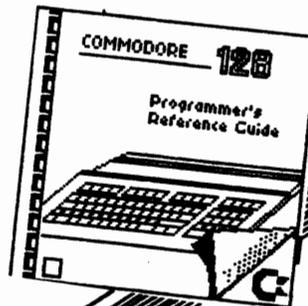
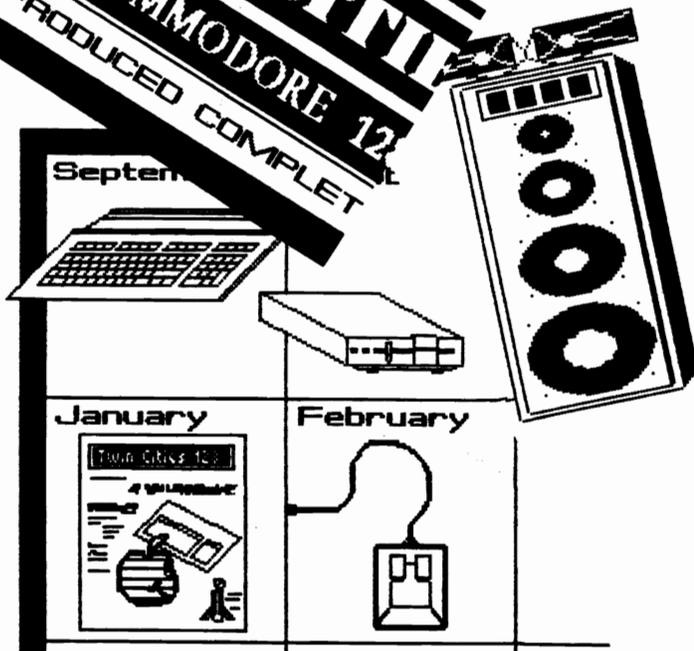
The new 1351 "true mouse" controller gives the GEOS user extremely delicate control and excellent movement response.



CBM Engineers
Q&A Forum



THE COMMODORE 128
COMPLETELY PRODUCED COMPLETE



3i 15...basb zi warrsq