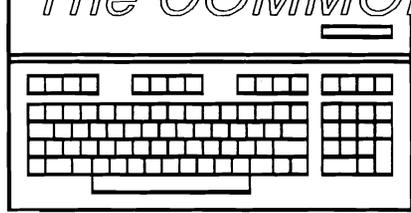


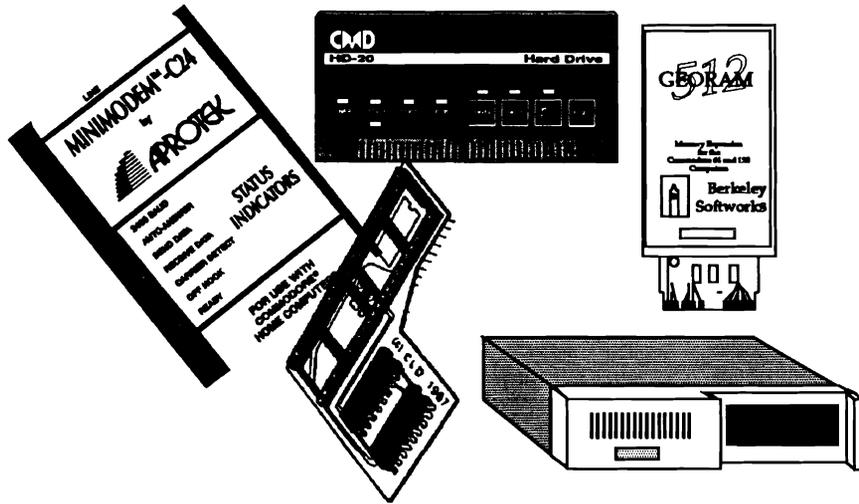
TWIN CITIES 128
 The COMMODORE 128
 Journal



ISSUE 28

September 5, 1990

Focus On Hardware



Rumor Opinion & Mayhem

by: Loren Lovhaug

One common barometer for measuring the health of a computer within its marketplace involves the production status of the computer and peripherals for it, specifically, are they still being manufactured and sold. Using this test as a guide your verdict would have to be that our 128s are "half-dead", or if you prefer, "half-alive". Although sources within Commodore tell me that a few months ago a small run of 128Ds were shipped to satisfy certain European commitments, as far as Commodore is concerned the 128 is a goner. It is clear that unless absolutely forced to do so by contractual obligations or by the need to clear out warehouse space, Commodore is not interested in selling 128s. Strike one.

With peripherals, the outlook is a somewhat less bleak. Although our friends at Commodore are making it difficult for folks to put their hands on 1571 and 1581 drives, (pfffft, pop, strike two) this TC128 issue is a testament to the fact that there are plenty of folks who sell hardware for your 128. Creative Micro Designs, Xetec, Software Support International, Dr. Evil Labs, Approtek, Brown Boxes Inc., all offer a host of quality products. However, like the batter standing in the box with two strikes against him our choices seem a whole lot more urgent.

Continued on Page 2

C128 PRICE AND PROGRESS REPORT...

Itsy-bitsy, GENie, Cheeple

Starting October 1, GENie will introduce "Star*Service", another national subscription online information service. Like other subscription services such as Prodigy for MSDOS computers or QuantumLink for the C64 mode, Star*Service differs from the classic "the meter's always running" on-line rate structure by charging a flat fee each month for unlimited use of certain services. In this case, for \$4.95 a month, you are entitled to unlimited non-prime time use of the following GENie services:

- NEWS, WEATHER, SPORTS
- TRAVEL INFORMATION & RESERVATIONS
- CLOSING STOCK QUOTATIONS
- ELECTRONIC ENCYCLOPEDIA
- GE MAIL (TM) EMAIL (SEND AND RECEIVE)
- LEISURE & PROFESSIONAL BULLETIN BOARDS
- CLASSIC GENIE SINGLE PLAYER GAMES
- SHOPPING SERVICES
- SUBSCRIPTION TO GENIE'S MAGAZINE

The unlimited Star*Service is available from 6 PM to 8 AM local time on weeknights, all day Saturday and Sunday and on certain holidays, the normal "non prime-time" hours for business telecommunications. Star*Service accessed during prime time hours is charged at the prime time rates. GENie says that if you are not satisfied after trying Star*Service for 30 days, they will refund your \$4.95.

Also effective October 1, GENie's is adjusting its normal connect rates to \$18 per hour prime time, and \$6 per hour non prime time (for 300, 1200, and 2400 bps).

Call a GENie With a Wizard!

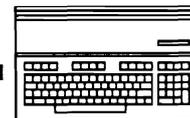
One of the hottest trends in the microcomputer industry today is the creation of dedicated "front-end navigators" for use with

Continued on Page 2

Table Of Contents

Rumor/Opinion/Mayhem	Page 1	Lovhaug
Price & Progress Report	Page 1	Hudson
RAM Expansion Alternatives	Page 5	Lovhaug
Minimodem C24 Review	Page 9	Hudson
Contriver M3 Mouse	Page 10	Hudson
Bible Search v3.0 Review	Page 11	Hudson
Maidstone Quest Review	Page 12	Hudson
Super-Charging Your CP/M	Page 13	Winchester
Internal ROMs Revealed	Page 16	Sullivan
A Small Memory Expansion	Page 18	Curio
Dynamic Keyboarding	Page 21	Treuenfels
Lt. Kernal Hard Drive	Page 23	Brier
CMD Hard Drive Overview	Page 26	Lovhaug

Want more information about TC128? Check out the information on pages 22 & 27! Thanks for supporting North America's Only C128 specific publication!



Price & Progress Continued

by: Frank Hudson

Bundled with the interface is a suite of 64 and 128 terminal programs (even one for CP/M that allows users to exceed 1200 bps). DesTerm is the C-128 native mode offering, but several other telecommunications software packages, including C-Net, Bill Coleman's geoTerm 128, BellTerm and Dialogue, are reported to be adding Swiftlink support. We did wonder about taking up the valuable cartridge/RAM expander port with Swiftlink, but since it allows its software to be relocated into at least three different internal memory locations and Dr. Evil specifically claims it to be compatible with the Commodore RAM expanders, it should co-exist on a Aprospond or similar multi-slot cartridge extender. Even more intriguing is the possibility that multiple Swiftlinks on one 128 could be managed with the proper programming, giving our favorite 8 bit multi-line capability.

We're looking forward to wringing out this inexpensive little breakthrough in a future issue of Twin Cities 128, but you can order yours today from Dr. Evil Laboratories P.O. Box 3432 Redmond, WA 98073-3432. They ship via 1st Class Mail, but you should allow four to six weeks for order turn-around.

Now Stepping Down to A Slower Transfer Speed...

Speaking of order turn-around, several readers wrote us to tell us it was taking a long time to get Dialogue 128, the terminal program we waxed ecstatic about last issue. That's disappointing to hear, since Dialogue is so good we'd hate to wait an extra minute to get it, but would wait a month of sundays for if we had to. Small firms sometimes have troubles in this area, whatever their good intentions. Besides finding the time to fill orders (the author/producers of many 128 products have "day jobs"), carrying the costs of appropriate inventory can strain budgets. From reports reaching us, orders are being filled eventually, but with some taking over six weeks. Readers who can document non shipment of Dialogue, can contact us via our mailbox and we'll report back on the issue in a future issue of TC-128. Further on in the caveat emptor patrol, a couple of readers have bought copies of B-Ware 128 Publisher and Illustrator previewed with screen shots last issue. Though we ourselves have not seen and evaluated these products yet, the reports from users who have are strongly negative. Dan Blackwood wrote that he found the manuals skimpy and the overall power of the program below the level he expected. Until we obtain a review copy of 128 Publisher or Illustrator we cannot recommend this package, no matter how nice the screen shots looked.

Which brings us to a point about this section of TC-128. We try to get information to you as soon as we receive it, and that means products mentioned here will often be listed on the basis of online reports, press releases, and other industry sources. We'd hate to think any of you confuse the C128 Price and Progress Report with a full-fledged TC-128 review. A lot of you write to tell us that you base your purchase decisions at least partly on TC-128 reviews, and that's not a responsibility that we take lightly. A full, fair and tough-minded evaluation takes place when we obtain and take time to test a 128 related item, and those pieces take longer to prepare than a notice in Price and Progress. Look here for the first note of new happenings in the

128 world, unless you want to be the first on your block with the new goodie, wait for a full review before spending your money.

Find Out If You Can Afford the Hardware In This Issue

Bremer Systems Inc. has a new personal finance program out now called "System 6". It needs a C128 with 80 column video, a 1571 disk drive, and a Commodore compatible printer. Among other functions, they claim that System 6 will manage your checking account, your budget, your income, your credit cards, your assets and your investments. And "You don't have to be an accountant to make the system work for you." Reports? "A great variety of report formats lets you examine each account in detail or provide general overviews of your financial standing. A budget analysis gives you insight into your financial performance, and a simplified balance sheet, available on demand, gives you information about your net worth." For current price and availability information you can write to Bremer Systems, Inc. 4242 Collie Court Lilburn, Georgia 30247 or send Quantum Link messages to "Bremer".

When the Going Gets Tough, the Tough Change Colors

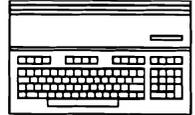
It won't help your net worth, but we saw it on QLink, honest. This August someone named "Person al" on Q was offering a revolutionary step in 128 hardware: replacement C128 case tops in custom colors. They say the standard Commodore beige lower case half and keyboard looks very sharp with oh say, pink! Or fluorescent orange! The price mentioned in the press release was \$35 and your old 128 upper half in exchange. No contact address was provided, but you can try to track down "Person al" on QLink. Sounds like just what you might need to match your 128 to your skateboard's color scheme.

Make Mine Brown Please

SRAM prices have dropped, so the Quick Brown Box prices have gotten lower too. The 256K size drops from \$289 to \$199, the 128K model from \$189 to \$149, the 64K from \$99 to \$89 and the 32K from \$79 to \$75. Add \$3 shipping and handling in the U.S. and Canada if you'd like to order these shirt-pocket battery-backed RAMdisk cartridges direct. Address for Brown Boxes Inc is 26 Concord Road Bedford MA 01730. Their phone numbers are 617-275-0090 and 862-3675.

Four Books, Three Dollars

Nope, this isn't the Book of the Month Club offer, but SOGWAP's software's new demo for their Bible Search program (see a full review elsewhere in this issue). The demo version has all the features of the whole program. The "catch" is that you get only the four gospels on the demo. It runs on any C64/128/128D with at least one 1541/71 disk drive. An on-line help function plus a separate help menu program is provided. We think you'll be amazed by the speed of the thing, even on a plain 128 with a single 5.25 inch disk drive. After seeing the demo, few serious Bible students will want to put off buying the whole Bible Search package. The Bible Search Gospel demo is available from SOGWAP Software 115 Belmont Road Decatur IN 46733 for \$3 shipping and handling. You may freely copy and pass on the Bible Search Gospel demo only provided you do not alter or change it in any way.



C128 RAM Expansion Alternatives

by: Loren Lovhaug

Back in 1985, when the 128 was first released, its standard complement of 128K of RAM was considered plentiful. But today, when most new computers are sold with a minimum of 1 megabyte (1024K) of RAM, the 128's memory capacity appears puny. Of course, comparing the relative merit of microcomputer systems solely on the basis of how much memory that system contains is foolish. After all, does it really matter how much memory a computer has if that computer cannot make effective use of that memory? For instance, until a few years ago most MSDOS applications could not make effective use of memory beyond 640K. This meant that a MSDOS computer with 1 megabyte of RAM was not significantly better than a MSDOS system equipped with 640K of RAM. On the Amiga the situation is quite different. Because of the nature of its operating system there is never a situation where the addition of memory does not enhance the performance of the system.

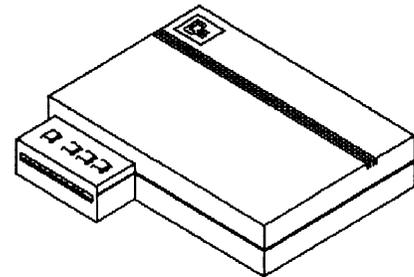
The 128's situation with respect to RAM expansion lies somewhere between the extremes outlined above. The 128 is not designed to accept direct, internal expansion of its 128 kilobytes of random access memory, even though the 128's MMU and other aspects of its architecture seem to hint otherwise. Therefore it should not surprise anyone that many 128 programs are totally oblivious to the addition of extra RAM. Non-programmers with restricted computing budgets should bear this in mind during the rest of this discussion: taking advantage of any C128 RAM expansion requires that your software knows how to take advantage of it. However, there are a number of applications that can benefit from the addition of RAM, and not entirely by coincidence, these programs include some of the very best applications available for the 128. There is even a small complement of applications that actually require some sort of RAM expansion. Programs which use RAM expansion often can alleviate some of the limitations of the 128's hardware design; most often the relatively slow speed and low capacity of its disk drives, but the right software/hardware combinations also can increase the resolution of your video display, add features and usefulness to programs you might already own, and even make your 128 more portable.

There are four types of RAM Expansion currently available to the 128 owner, with some additional prospects on the horizon. Each form of RAM expansion is unique in its implementation and integration into the 128's operating system as well as its functionality. Here is an overview of these options with an evaluation of the relative strengths and weaknesses.

Commodore's 1700, 1764, and 1750 REUs

The only 128 RAM expansion options that are officially supported within the 128's operating system itself are Commodore's 1700 (128k), 1764 (256k), and 1750 (512k) RAM expansion units. These were the first RAM expansion options available for the 128, the 1700 and 1750 appearing in mid 1986, the 1764 following about a year later. Though the 1700 and 1750 were designed and labeled as 128 peripherals, the 1764 includes a replacement heavy duty power supply for the Commodore 64 (unnecessary by the 128, designed from the start to support the

Commodore's 1700/1764/1750 RAM Expansion Unit



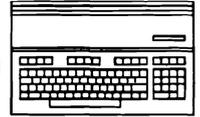
Lovhaug 1990

Commodore RAM expanders). A 1764 will however work fine with the 128, as it employs nearly the same design as the 1700 and 1750.

Although they differ in the amount of RAM they contain, all Commodore's RAM expansion units use a similar package. They are fairly large (13.7cm x 12cm x 2.5cm) cartridges which plug into the 128's expansion port located on the right rear backplane of the machine. Commodore did not provide a "pass through" of the cartridge port, so you must use a multi-slot cartridge port expander (such as the Aprospan available from Aprotek) if you wish to connect another cartridge in conjunction with a Commodore REU. Sandwiched inside Commodore's squarish clamshell-like plastic case is a single circuit board that holds each unit's complement of RAM, and a special custom integrated circuit called the RAM Expansion Controller (REC). The function of the REC is to copy or exchange blocks of memory between the 128's internal memory and the memory inside the RAM expander. The REC can do this at a very high rate of speed, as fast as 64K per one sixteenth of a second. This aspect of Commodore's REU design makes it very nice for graphics and animation applications, as well as the fastest of the RAM expansion options available for the 128.

From a programming standpoint, the copy and exchange functions of the REC are accessed with the BASIC 7.0 commands STASH, FETCH, and SWAP, as well as via the Commodore 128 kernel routine DMA_CALL (\$F7A5). The REC's registers are mapped into the 128's architecture at locations 57088 - 57098 (\$DF00 - \$DF0A). At no time is the RAM inside of the RAM expansion unit banked into the 128's memory map, therefore any data that is stored within the RAM expander must be copied or swapped into the 128's internal memory map in order to be accessed by the 128's 8502 microprocessor.

Due to their age and their "officially sanctioned" status, the 17xx



been commercially produced, but all of them take some dedicated memory (the VDC RAM) to accomplish.

Which brings us back to that 16K present in stock "flat" 128's. Expanding the amount of available VDC RAM allows properly designed programs to display more colors at higher screen resolutions on the C128. Applications such as the BASIC programming language extension, BASIC 8, paint programs such as IPaint and Spectrum 128, and even the new disk duplication system, Fastrac 128, take advantage of expanded 8563 VDC RAM when present. For this reason the latest 128 model (the 128D) has 64K of VDC RAM installed at the factory.

Okay, so now that you are convinced that the addition of video RAM to your older 128 is a good thing, you are probably asking yourself, "How do I accomplish this feat and how much is it going to cost me?" The answer depends on your affinity for electronics projects. If you feel comfortable desoldering small integrated circuits from motherboards in tight spots then you might consider doing the VDC RAM upgrade yourself. However you should be forewarned, that without the proper experience you could very easily ruin your 128's motherboard. In fact, Rick Kane, the author of IPaint and frequent Twin Cities 128 contributor, did just that the first time he attempted the upgrade. However, if you have a brave heart and a steady hand, the process is straightforward. First carefully desolder the 4416 DRAMs (they are just to the right of the 8563) from the motherboard. Then place sockets where the 4416 DRAMs once were. Finally, place 4464 DRAMs, into the sockets. You may use 150ns parts or faster. The current price for 4464 100ns DRAMs are approximately \$3.00 each (according to the August 1990 Issue of Computer Shopper magazine) and your socket cost will likely be less than \$1.00 each.

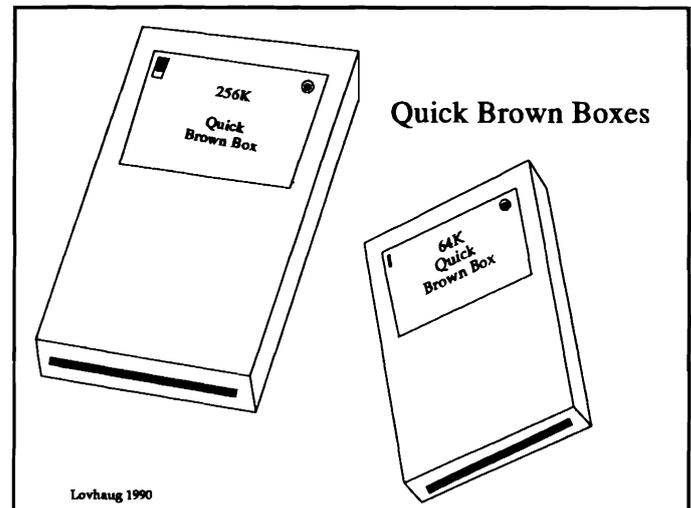
As inexpensive as the do-it-yourself approach to upgrading your video RAM is, I cannot recommend that you attempt it unless you have an immense amount of confidence in your abilities. Fortunately there is another, much easier method. Software Support International sells a solderless VDC RAM upgrade kit that allows you to quickly and simply upgrade your video RAM without hacking at your machine. For under \$50 you get an easy to install daughterboard that fits into the 8563 socket on the 128's motherboard. To install this daughterboard all you do is: open your machine, remove your 8563 from its socket, plug the daughterboard into the 8563 socket, and then finally plug the 8563 into the socket on the daughterboard.

Quick Brown Boxes

Another C128 RAM Expansion option comes in the form of the Brown Boxes Inc.'s Quick Brown Box (QBB) cartridges. The Quick Brown Boxes are the only currently available form of C128 RAM expansion option that is non-volatile, that is, the data you store within the Quick Brown Box is not erased when you turn off your computer. This impressive feat makes the Quick Brown Box even more of a mini-hard disk than the traditional RAMdisks implemented on the Commodore REUs. Not surprisingly, the Quick Brown Boxes' ability to serve as compact non-volatile storage combined with its ability load programs and data very quickly have caused most QBB owners to use it as a disk drive replacement.

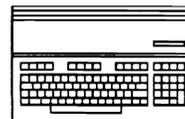
Though the QBBs are justifiable for this purpose alone, they are

also the only currently available C128 RAM expansion option which can actually be switched directly into the C128's memory map itself. No other RAM expansion product has the ability to "live" within the internal domain of the C128's MMU. This unique ability to be switched in offers enormous potential to the C128 programmer who wishes to create custom applications.



Now that we've got the programmers' attention, here's how the QBB does it: Each QBB consists of 16K banks of battery backed CMOS memory. Only one of these banks is "switched into" the 128's memory map at a time. Any QBB RAM that is "switched into" the system appears between locations 32768 and 49151 (\$8000 - \$Bffff) of BANK 0. Because the C128's architecture maps the BASIC 7.0 ROM in this address space, there is no simple way to execute a BASIC program solely in "box RAM" itself, however you can run an assembly language program that is fully self-contained within the box. Of course you can use the QBB for data and overlays with a BASIC 7.0 program, in addition to loading the program itself from the QBB. What bank is switched into the 128's memory map and all other QBB related operations are controlled via a write only control register at location 56832 (\$DE00). Besides selecting the QBB bank that is currently selected to be inside the context of the C128's memory map, this register is used to designate whether the QBB is in read-only mode, read-write mode, or switched out. This control register is available from all C128 bank configurations except those banks where internal system RAM is active in the address space (banks 0-3) or when the control register itself is switched out (bank 14).

Unfortunately for non-programmers, there are currently only a few applications that feature direct support of the Quick Brown Box, though many programs can be loaded from the QBB. Of those applications that do support the QBB directly, the most notable is the popular C128 word processor, The Write Stuff. Using the largest of the Quick Brown Boxes, the 256K model, the entire Write Stuff word processor and its superb spelling dictionary, can be loaded into the box leaving over 300 blocks free for storage of your documents or other utilities. Even the smallest Quick Brown Box, the 64K model, allows you to store the Write Stuff and still have a small amount of storage space left for a document. Also, Miklos Garamszeghy has written utilities



The Arotek Minimodem C24

Reviewed by: Frank Hudson

Commodore computers took a vital part in the pioneering of microcomputer telecommunication in the early 80's, not just because the Vic 20 and C-64 were home computing bargains, but because they were soon supplied with good low-cost modems. Once 300 baud was the rate at which most telecommunication took place, and Commodore owners with a VicModem or the autodialing 300 baud C1650 that succeeded it communicated on an equal level with those who spent more to go online. The classic Commodore 1670 1200 bits per second ("bps" is technically a more accurate term than the commonly used "baud") modem came along just as the rest of computerdom was making the move to 1200 bps. At its current below \$100 price, the 1670 remains one of least expensive 1200 bps modems around. Widely available cheap modems meant that Commodore users were more likely to go online than their fellow 8 bit computerists. I don't know if anyone has noted this before, but I believe this helped to shape the close-knit user community that continues to be the leading source of support for our computers.

Alas, this hardware advantage has eroded recently. While low cost 300 and 1200 baud modems for the C-128 are still in good supply, much telecommunication these days takes place at 2400 bps. By the late 80's modems designed to operate at this speed proliferated for machines with the widespread "industry standard" RS-232 port found on the IBM XT and its many clones. Through the use of special cables or interfaces C-128 owners could avail themselves of these standard RS-232 modems, but now we paid the same price as the clone owners did and then needed to spend a bit extra for the special C-128 connection cable.

But there is hope yet for the tradition! Arotek has introduced a direct connect 2400 bps modem that plugs right into C-128's user port. The Minimodem C24 takes its power from the user port (like the 1670) and includes an internal low-fi speaker to monitor dialing progress. The C24 answers to the "Hayes compatible" AT command set (again like the 1670) and goes further than the 1670 in that it includes a set of seven LED's indicators on top of its low blue plastic case. Although the Arotek has some small inconveniences and rough edges - and I'm going to mention every one of them - it is a fully workable 2400 bps modem. The best part is the C24's street price: less than \$100. If you've shopped standard RS-232 external modems lately you'll know that this is a good value, and of course the direct connect Minimodem requires no additional cable or interface. Once more it is possible for C-128 users to move up in telecommunications speed at a cost below that paid by other computer owners.

The Minimodem C24 comes shipped with a pair of 1541 "flippy" disks including several of the most popular freely-distributable C-64 and C-128 terminal programs and a 16 page saddle stapled softcover manual. The manual contains little in the way of hand-holding for the new telecommunicator, but an included loose sheet of paper does give explicit instructions for getting online with CGTerm, one of the programs included. 128 owners are also provided with a copy of Desterm, a powerful if not always fully intuitive 128 terminal program whose important documentation file is in Arc'ed form on the disk. The manual

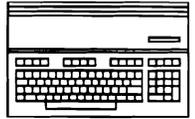
does include a short rundown of what each command and software register accessible in the modem does. All in all, the documentation for the modem and included software is adequate for experienced telecommunicators, but beginners will find many of their questions unanswered.

Installation of the modem is less convenient that you might think. First off, the seven foot long modular phone cable is permanently attached at the modem end to the Arotek C24. If you (like I) need a longer length of cable to reach your phone jack, you'll be in the market for an extension adaptor. If this attached cable ever goes bad or the jack at the end breaks, you'll need to do some cable splicing to continue to use the modem. Unlike the 1670 or other popular modems, the Arotek does not include a pass-through jack for your telephone. If you want to have a phone nearby you're going to be either stressing that jack often as you plug and unplug the modem, or buying another adaptor. Even plugging the modem into the user port is tricky. On first attempt, I managed to get the C24's smallish user port connectors misaligned with my 128's port, temporarily disabling my disk drive. This error is impossible to accomplish with any other C-128 modem or RS-232 interface I own. Although no damage resulted, I encourage Arotek to redesign the connector, or at the least mention this problem in the troubleshooting section.

Once plugged in, getting online with the C24 is normally a simple matter of setting your terminal up to work with a Hayes compatible modem and selecting 2400 baud. The C24 supports not only the Hayes AT commands, but most of the S and & registers too. These latter two parts of the Hayes compatible command set are used to reconfigure the modem to your personal preferences or to meet unusual needs (a process some other modems use DIP switches to accomplish). Unfortunately, the Arotek does not have non-volatile memory onboard to save your personal settings after your system is powered down. The modem works fine with the settings as fixed at the factory, but if you like or need things to be different, you will want to use a terminal program that lets you send a longish modem initialization string to set things up each time you use the modem.

I used the Minimodem C24 with all of my collection of C-128 terminal programs, from Bob's to Des to Nez to Ultra Term without a hitch. Our current favorite term at the TC-128 office, Dialogue 128, also worked well with the C24. However, QuantumLink's proprietary software wouldn't work configured to its Hayes setting. The same loose sheet that contains the beginner's instructions tells you to use the C24's 1670 emulation mode with Qlink. 1670 mode? Yes, though the Arotek modem worked with every C-128 term I threw at it, there is an internal switch that can be set inside the C24 to make it emulate the Commodore 1670. This QLink problem is puzzling. If you use QLink often, constantly having to open the C24's case to move the switch gets annoying, but if you don't use "Q", I doubt you'll ever have to think about it. It wasn't just that QLink doesn't have a 2400 baud option. The C24's native mode was capable of working at 300 or 1200 baud whenever another speed was

Continued on Page 10



Bible Search v3.0

Reviewed by: Frank Hudson

Bible Search from SOGWAP Software has always been one of the TC-128 staff's favorite applications, not only because it serves as an invaluable key to using the Bible, but because it also demonstrates how effectively the C128 can find and retrieve data, even with floppy disk drives.

That is why I was so happy to see a package from SOGWAP in the mail containing a review copy of Bible Search Version 3.0. It didn't take much investigation to discover that this was a major upgrade. When we last looked at Bible Search in 1989 it contained only the New Testament, and our reviewer Tom Wright, suggested that the ability to save references as a text file would be a valuable aid to merging them within a word processor. Well, that's Bible Search V3.0. Bible Search is now available with the entire Old and New Testament and a PETSCII file output option. In my opinion these improvements more than double the utility of the package.

For those that are not otherwise familiar with this program, let's go back and review what Bible Search can do and how fast it can do it. Bible Search is at its core a compressed and indexed text database, containing (in the version I'm reviewing) the entire Bible in your choice of either the King James Version or the New International Version translations. While it could be used as a way to read the text a verse at a time, it is of course the indexing and retrieval functions that make this computer version so valuable. Any word or combination of words can be searched for and Bible Search will return with a list of applicable verses or the actual text of each verse containing your searched-for words. The verses obtained by any search can be understood in context by using the scroll keys to move backward or forward in the text from where the verse was extracted. If you select a word not in the section you are searching, Bible Search will tell you it finds 0 references. If you select a word not in the Bible at all, Bible Search acts something like a spelling checker and offers you a screen full of alternative words that you might have really meant to type.

Despite the admirable compression achieved by Bible Search the entire Bible still takes up four 1581 disks or seven 1571 floppies. Bible Search handles this unavoidable multi-disk situation well. The Bible is cut up into contiguous blocks of books (1581 Disk One for example goes from Genesis to First Samuel), so that knowledgeable users can often select the proper disk right off the bat. Multiple disk searches are easily done, simply by putting another disk in the drive and directing Bible Search to switch to that disk's sub-index. If you flub up and have the wrong disk in the drive, the program traps the error and tells you which disk to insert. Commodore REU owners can load one of the disks onto the REU for automatic disk-less searches of one section of the Bible, and all the Bible text sections can be installed in one native mode CMD hard drive partition, eliminating the need to swap disks altogether.

As a trivial example of the program's search facility I attempted to find what Biblical advice might pertain to TC-128's field. God evidently desires for us to abstain from graphic user interfaces

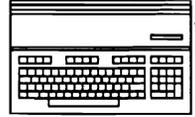
and the input devices particular to them, for Leviticus 11:29 tells us "These also shall be unclean unto you among the creeping things that creep upon the earth; the weasel, and the mouse, and the tortoise after his kind." I don't know about you, but that tortoise comment may mean He's only used GEOS without an REU. On the other hand, He approves of the C128's bank switching arrangements, for in Exodus 29:17 we find "And thou shalt cut the ram in pieces". Although I must admit there is no evidence of direct Divine inspiration in the 128's design, God Himself must be the Ultimate Programmer. I'm not sure what language he prefers, but Bible Search finds no references to the word "BASIC" in the King James Version, but hundreds of verses with "Forth".

More serious use of the Bible is just as easy. Shortly after the program's arrival we were able to use Bible Search to find the location of a particular parable which only one of us could recall. In less than five minutes from the time we opened the disk box, a search on the word "Vineyard" found it for us.

Yes, though there is now much more text to search, Bible Search is still surprisingly fast, even from a 5.25 inch disk drive. Any reference found on a particular disk pops up in seconds, and a worst case multiple disk search across all the disks in the set can be done in a minute or two. The performance of this product continues to amaze us, and programmer Michael R. Miller seems to have done it without resorting to any "hot-rod the disk drive" routines, instead relying on what must be very good search algorithms and a highly efficient index. The user interface is more pedestrian, function key and command oriented, but it works well enough and we wouldn't want to see anything detract from the speed of this application. The data is stored in large REL files, but the disks are not copy protected, allowing installation of Bible Search on a variety of drive types. Anyone who reads the Bible for more than its literary value shouldn't need to be reminded that this does not mean that it's OK to steal!

If you wish to save a particular verse to a disk file for use in a word processor, V3.0 will allow you select a second drive as the destination and then from within the program a function key allows one to open the file for writing. A press of a F or D key after a verse's display saves it to this open file. New verses can be appended to the disk file as they are found, and any Bible Search session can work with several different disk storage files. It shouldn't take much inspiration to see that this is an enormous aid to any writer who works with The Word and a word processor! Furthermore, importation into GEOS could lead to some nice posters of your favorite verses and via The Write Stuff 64's "talking word processor" module you could even generate computer speech Bible stories.

The files created from the New International Version (NIV) we received were straight PETSCII with no hard carriage returns, and I found they imported well into The Write Stuff and even my Amiga and MS-DOS word processors (with the appropriate PETSCII to ASCII translation in the last two cases of course). A
Continued on Page 12



Supercharging Your CP/M BIOS, CCP, & Bootdisk Utilities

by Randy Winchester

Too often, Commodore 128 owners will boot up CP/M, take a quick look at it and think "This sure looks slow, dull, and boring." The boot disk goes back in the box, never to see the inside of a disk drive again. CP/M on the C128 doesn't have to be a creaky crawler. The problem is with the default configuration of CP/M on the boot disk. Here are some suggestions for speeding up CP/M and adding new features.

A New CP/M Boot Disk

All that is required to make a C128 CP/M boot disk is to format a disk with FORMAT.COM and copy CPM+.SYS and CCP.COM to it. The disk must be formatted as either a C128 single or double sided disk. FORMAT.COM writes a boot sector on the disk that enables the C128 to load and install the system files.

Fixing CPM+.SYS

The standard CPM+.SYS provided on the C128 CP/M boot disk has a couple of crippling inadequacies. The 40 column screen is always enabled, slowing down the system by about 10%. The interrupt driven RS232 routines are always enabled, and set for an unreasonably high 300 baud. The higher the baud rate, the more interrupts are needed to service the RS232 port, and the less time the processor has to do its work.

Both problems are easy to fix. The public domain CONF.COM can remedy both with the command line:

```
CONF BAUD=75, FEEL=0, 40COL=OFF
```

The routines to scan the keyboard are also interrupt driven and depend on the baud rate setting. Although the baud rate can be set as low as 50, at this setting the keyboard starts to miss key presses. The FEEL parameter adjusts the number of interrupts used by the keyboard driver. The value indicates the number of interrupts skipped. A value of 1 tells the driver to ignore every other interrupt. At 75 baud, FEEL should be disabled by setting it to 0. 40COL=OFF disables the 40 column screen.

The baud rate can be hardwired into the BIOS by patching CPM+.SYS. The byte specifying the baud rate is found after the label "RS232" in the device table. The address is \$0da7 for the DEC 85 version or \$0d98 for the MAY 87 version. Normally, this byte will be set to 06 for 300 baud. The values to choose from are:

0	0
1	50
2	75
3	110
4	135
5	150
6	300
7	600
8	1200

Setting this byte to 02 for 75 baud is the single most important speedup you can make, with an obvious improvement in performance.

Unfortunately, the FEEL and 40COL parameters aren't as simple

to patch. The easiest way to make these permanent is to edit the CP/M source code and generate a new system. This requires MAC and RMAC, assemblers from the utility disks offered in the C128 System Guide. Although these disks also contain complete system source code, I recommend generating a system from the files in the widely available BIOSR4.ARK by James Waltrip, or my own BIOSR5.ARK.

BIOSR4 and R5 are archives containing modified source code. They've been reworked to remove all 40 column code, replacing it with an 80 column screen dump. Routines have been added to define the drive search chain and default drive, the shape of the cursor, and the keyboard repeat rate. The code is thoroughly commented, pointing out bytes to set for the default printer and secondary address, screen colors, and default baud rate. Additional disk types have been added to the disk parameter table in BIOSR5, including Miklos Garamszeghy's "Maxi 71," which provides a remarkable 398K per disk.

Upgrading the CCP

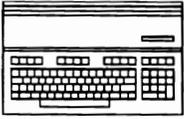
The Console Command Processor, or CCP, is the program CP/M relies on to communicate with the user. Its various responsibilities include printing the CP/M drive and user area prompt, interpreting console input, and loading and executing transient commands. The CCP.COM supplied on the CP/M boot disk is pretty bare bones. It has six limited resident commands: DIR, DIRSYS, ERASE, RENAME, TYPE, and USER.

A public domain CCP replacement that corrects some of the inadequacies of the standard CCP.COM is CCP Plus version 1.05. CCP+ doesn't include resident commands; they have been replaced with new features. CCP+ includes a time display in the drive/user prompt, and named directories similar to those in more modern operating systems. Commands can be executed from user areas other than the current one. An impressive feature is the ability to execute COM and SUB files from a command library much like MS-DOS.

If you use CCP+, you will have to select commands for your system. This can be fun, rather like going on a free software shopping spree. All the programs mentioned in this article, except QDisk, are public domain.

A RAM expansion, with its fast access and loading times is the wisest place to put frequently used commands, since they will load and execute almost instantly. By using a PROFILE.SUB on your boot disk, commands can automatically be copied to the RAM disk when the system is booted. For all purposes, commands stored on a RAM disk will appear to be resident commands. On the other hand, if you don't have a RAM disk, you might decide that CCP Plus's features don't make up for the lack of resident commands.

Another solution is to use a Quick Brown Box along with QDisk software as a CP/M RAM drive. The Quick Brown Box uses a battery to keep its 64K of RAM active between computing sessions. I use the box to store 64K of commands that are



available the instant I switch on the computer.

CCP Plus is available in the file CCP105P.LBR, along with many supporting programs and utilities.

Speeding Up Disks and Drives

A command my system runs when it first boots is C1571.COM. This shuts off the 1571's redundant write verify, reducing the time it takes to save files. Look for a version dated 24 Feb 86 or later. An earlier version has a bug that prevents the use of MFM disk formats.

Using MFM disk formats is another way to get maximum performance from CP/M. You'll still need Commodore GCR formatted disks for boot disks, but MFM formats can be used for all disks that don't need to be bootable. Not only do some MFM formatted disks provide more storage, they are typically faster than GCR disks. My recommendations are for the Epson QX10 format supported by the standard BIOS, or Maxi 71 mentioned earlier. QX10 disks offer 390K of storage; Maxi 71, 398K. That's 62K more than a C128 double sided CP/M disk.

Eliminating SETDEF

The SETDEF command is normally used to define the drive search chain and temporary drive. The drive search chain is a list of up to four drives that CP/M will search each time a command is issued. Sometimes CP/M needs to make temporary files in which it stores data while a command is being executed. SETDEF is also used to tell CP/M which drive to use for temporary files.

If you decide to generate a CPM+.SYS using BIOS R4 or R5, there is a section of code in the CXIO.ASM module for values for the drive search chain and temporary drive. These values also can be patched into an existing copy of CPM+.SYS.

The drive search chain is located from \$1268 to \$126b in the DEC 85 and MAY 87 versions, or \$0e68 to \$0e6b in the AUG 85 version of CPM+.SYS. The drives are numbered starting with 01 for drive A. The default or currently logged drive is 00. Positions that aren't used should be filled with \$ff. The byte immediately following the search chain, \$126c in the DEC 85 and MAY 87 versions or \$0e6c in the AUG 85 version, specifies the drive for temporary files. This byte defaults to 00, the currently logged drive. If you use a 1750, have the system make its temporary files there. It's much faster than writing and erasing them from a floppy disk. The suggested entries for RAM disk users are:
\$1268: 0d 00 ff ff 0d If a file isn't found on the RAM disk (drive M) CP/M also will search on the currently logged drive. Drive M is the temporary drive.

Since these patches cover the important functions of SETDEF, there is little need to run it every time you boot up.

Eliminating SUBMIT

When CP/M is first booted, the CCP searches for PROFILE.SUB, and if it is present, executes it via SUBMIT.COM. This is the usual method for automatically setting up the system. A typical PROFILE.SUB might consist of the following commands:

```
C1571 [A,B,D CP A1:.* M0: DATE S
```

PROFILE.SUB is read in line by line and each command is loaded and executed in turn. In this example C1571.COM turns off the

1571's write verify, all files in user area 1 of the boot disk are copied to user area 0 of drive M, and DATE.COM sets the system date and time.

Thanks to the public domain CHN31, SUBMIT.COM can be done away with, at least for PROFILE.SUB and other short lists of commands. CHN31 constructs a COM file consisting of any commands you might put into a SUBMIT file. In the example above, each command would be entered at CHN's prompt separated by an exclamation mark. CHN is a big convenience and time saver. Since CHN command files are themselves directly executed, SUBMIT doesn't have to be loaded.

CCP.COM requires a quick patch to use this trick. Look for the string 'PROFILE.S' somewhere in or around the seventh sector of CCP.COM. Simply change the string to 'PROFILE.C' to execute a file named PROFILE.COM created with CHN31. Here is the PROFILE.COM I use:

```
QD F/FIC1571 [A,B,DIRNAME ONIF:SETDIRIF:HISTIDATE S
```

QD (QDisk) activates the Quick Brown Box as drive F, a 64K RAM disk. C1571.COM disables the 1571 drive's unnecessary write verify. The letters following the command are a list of 1571 drives on my system. DIRNAME is a CCP Plus utility that starts the named directory feature. SETDIR is then loaded from drive F. It reads the list of directory names and installs them in memory. HIST provides recall of the last 20 commands typed to the system. DATE then sets the system time and date.

If you don't use a Quick Brown Box, you'll need to copy commands and other support files into your RAM disk. The way I do this is to place files I want to copy in user area 1 of my boot disk. I use the command 'CP 1:.* M0:' in my PROFILE.COM (CP is the name I use for ARCOPY.COM, an excellent public domain copy program) and everything in A1 is copied to M0.

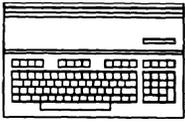
Laying Out a Boot Disk

User area 0 should include CPM+.SYS, CCP.COM, and PROFILE.COM. I also keep files in user 0 that are specified in PROFILE.COM but executed only when the system is first booted. These include C1571.COM, DIRNAME.COM, and DATE.COM.

User 1 should contain files that will get copied to the RAM disk. I suggest CMDRUN.COM, which is used by CCP+ to execute commands from COMMAND.LBR; ROOT.DIR, containing a list of directory names; and custom keyboard definition files created with KEYFIG.COM and saved with SAVEKEY2.COM. I use two keyboard files, DEFAULT.KEY for normal operation, and ZDE.KEY for use with the public domain ZDE editor. You'll also want to include a COMMAND.LBR.

COMMAND.LBR can be created with a library utility, such as NULU152.COM. It can have as many commands, and be as large as you have room to dedicate to it. It should include replacement commands for those missing from CCP+, and any other commands you routinely use.

Here is a list of the commands in my COMMAND.LBR along with some comments. All these programs are public domain.



Maidstone Quest

Reviewed by: Frank Hudson

Twin Cities 128 has always enjoyed correcting misconceptions about the Commodore 128, and some of these misconceptions seem to have always been with us. A quick glance at this issue should show that two companies are trying very hard to see to it that the old walls preventing the hooking up hard drives to the 128 are coming down. Now how about the perennial "If only there were some good 80 column games." complaint?

Well, that one may not be dead, but it should be feeling mighty ill these days. As TC-128 Issue 26 pointed out, there is plenty of grass roots programming going on with 128 games, and the associated excuses that the 80 column mode doesn't lend itself to graphics or animation have been dispensed with as clever programmers gain experience with the higher resolution screen. A case in point is this issue's game review, Maidstone Quest from Silvasoft.

Maidstone Quest is a colorful adventure saga that grabs you at first look with good animation and well done art. On loading (it autoboots) you are soon presented with a title screen and a jaunty tune. A minute or so later you are at the entrance to the Grey Mountain Mines, a 28 level maze that will be your character's home until you solve the game's puzzle or die trying. Unlike more complex adventure games, your computer alter-ego is always identical in power and ability on starting out, but as compensation, a click of the mouse or joystick will send him jogging along the mine-shaft tunnel in a convincing fashion.

If he finds anything, the classic adventure game options of take, drop, spellcast, etc. can be selected with your mouse or joystick, as is direction of movement, all from menus surrounding the action window. As you might expect, lots of what you encounter in the mine is as the philosopher said, nasty, brutish, and short. Gnomes pound you with medieval Louisville Sluggers, spiders nip at you, trolls are always stealing anything you are carrying, and my favorite, the Gargoyles, just fall on you and crush you flat enough to fit into a disk mailer.

Given enough "health" points you can survive a bit of this action and continue, eventually to find the right magic items and defeat (veteran adventure games are probably ahead of me here) the Evil Wizard. All the usual adventure game tactics are valuable in Maidstone Quest: good map-making skills, read and examine everything with an eye for clues, and when you pay a troll listen up to what he tells you.

Jaded adventurers will not find much to challenge or surprise them here, but those who have not played much of this type of game and patient children old enough to read (or play with a helpful adult) might like it. Throughout, programmer Roger Silva's talent for computer illustration and animation shines. The C-128's 80 column screen gives your eyes a level of color and detail superior to the entry-level MS-DOS clone's CGA display sported by many of that system's games. There is enough verisimilitude here to convince you that you really are stuck in the mine with the bad guys, and in fact the quality of the animation is so good that it almost hurts the game for me. You

see, although you are smoothly moving the hero around the screen with the mouse or joystick, there are no arcade aspects to this title at all. You can't jump over a troll or outrun a gnome run. Maidstone Quest looks like an arcade game, but plays like a text adventure with menu-driven parser.

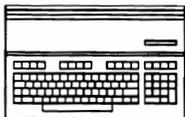
This last criticism, while emotionally accurate, is of course rationally unfair. Silvasoft doesn't claim that Maidstone Quest is an arcade game, and there are gamers who can resist the hyper-action of "Just Another Nintendo Equivalent". The simplicity of the story, strategy, and user interface may also be just what you are looking for. The beta version we tested plays well and reliably. Maidstone Quest requires a joystick or 1351 or compatible mouse hooked up to a 128D or 64K video RAM expanded flat 128, and as with all colorful games, a RGB monitor is recommended. As a further inducement, the game is not copy protected so it installs on any drive and programmer Silva says that if sales warrant he is considering more complex games to follow this example. I for one look forward to more C128 products as good looking as Maidstone Quest. To order Maidstone Quest, write: Silvasoft, P.O. Box 1006, Charlotte VT 005445, \$19.95

Continued From Page 11

bug in our review example the King James Version lead to scrambled text files when they were viewed within a C128 word processor, but SOGWAP reports that this has been fixed in current versions. If you own a King James Version Bible Search V3.0 that doesn't export PETSCII properly, SOGWAP tells us they will exchange your disks for free.

As before you also can dump any verse or verses directly to your printer and call up a particular chapter and verse directly by a direct verse read command. Although it makes for a few more disk swaps, the addition of the Old Testament greatly improves Bible Search in my estimation. After all we're talking about the addition of two-thirds more text and an important part of Western religious tradition.

With these additions, the purchasing options for Bible Search have grown. The complete Bible (Old and New Testaments, the version we tested) is available in King James or New International Versions for a list price of \$69. The New Testament alone, in either translation, goes for \$25 for King James or \$30 for the NIV. Commodore REU compatible disk sets cost \$10 extra, and you are asked to specify 1541, 1571, or 1581 disk formats. There are also combination deals for those that would like both translations. A sample version of Bible Search containing just the four Gospels is available for \$3. For more information or to order direct, SOGWAP's address is 115 Belmont Rd., Decatur IN 46733 Phone: 219-724-3900.



The Contriver M3 Mouse

Reviewed by: Frank Hudson

The Commodore 1351 mouse is one of my favorite shapes. Low and softly angled, it falls readily to hand. Though the Commodore Amiga and MS-DOS mice are not internally compatible with the 1351, I take small comfort in their external familiarity. If only everything were so comfortable when switching between different Commodore machines!

But this is only a personal feeling. Obviously the overriding consideration for C128 owners is the question of whether to use a mouse at all. In most cases this remains a matter of individual choice. Many competent 128 software packages don't support the mouse at all, and even those that do include excellent mouse drivers (such as the Pocket Series of productivity programs) may allow full use the program without a mouse or other "pointer device". The mouse may be the most successful alternative computer input device ever, but it has yet to eclipse keyboard-bound methods in many users' hearts.

That said, when it comes to graphical programs like GEOS and most paint programs, there is no question in my mind that you should operate these programs with a mouse. Trackballs have their adherents, but the common "click and drag" action is difficult to master on most layouts. Lightpens seem like a neat idea, but inevitably shift from registering consistently on the screen. Joysticks are great for games, but are maddening in their lack of precision and responsiveness when they are forced into duty as graphic input devices. Graphic tablets are excellent for drawing, but fall down when asked to select items from drop down menus.

Increasingly C128 users are reporting to us that the eight-bit peripheral shortage is extending to the once easy to get 1351 mouse. If you are late to GEOS or are looking to purchase one of the new breed of 80 column paint programs like Spectrum 128 or I Paint, what can you do?

Apparently Contriver Technology realized that Commodore owners increasingly want to operate their computers in the modern mouse-driven manner and moved to meet the need. Their M3 mouse (retail price, \$49.50) perfectly emulates the 1351 with programs that support the Commodore mouse. In fact, the M3's emulation is so perfect, that I face the possibility of having little more to say in this review, except to confirm that if you want or need a mouse for your 128 and cannot get a 1351, get the Contriver M3 instead. The M3 works with GEOS, Dialogue 128, and the Pocket Series. Holding down the right button while resetting the computer selects joystick emulation mode (just like the 1351). In addition, holding the M3's left button and resetting is said to select a paddle emulation mode, but since we know of no C-128 program that supports game paddles, we were unable to test this feature. So in general, plug it in and continue as if you had bought a 1351. If you are waiting to finish this issue of Twin Cities 128 and get back to your 128, you can stop here.

There are some small oddities I noted about the M3. It has an extremely short cord for instance, less than two feet long on my example. If you are right-handed you might consider that an

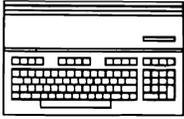
advantage of sorts, since the short cord can't curl back and into your way as can the 1351's longer tail. The left-handed computerist is forced by the 128's layout and the M3's cord length to use their less dexterous (or "less sinister" for all you entomology fans!) hand to operate this mouse. A disk of utilities is included with the mouse, but I noted nothing of value among the programs there. If you have a crying need for a sprite designer or a nondescript C-64 paint program and couldn't find them elsewhere perhaps you will disagree. Mice are among the C-128 hardware products that really don't need a manual, but the M3 comes with one anyway. It's something of a classic in the genre of Pacific Rim English documentation with howlers like "The M3 mouse have 3 modes" and "Congraturation! You have just purchased the finest product from contriver tech. inc. We take pleasure to recommend other items." Oh well, my knowledge of Chinese is certainly poorer than their knowledge of my language! I find the mouse's shape less natural than the 1351's and note that the roller ball (like that of many mice on the market) is under the heel of your hand rather than centered under your fingers. If you are unaccustomed to the 1351's shape and feel, I doubt you'll find these differences important.

Minimodem Review Continued from Page 9

needed, as long as it was under control of an ordinary terminal program. Perhaps somewhere out there is another program that won't work properly with the Arotek in Hayes mode, but then chances are that like the QLink software you'll still be able to use it (at 1200 bps only) with the modem's 1670 emulation mode.

If you're accustomed to using the C-128 at 1200 bps, 2400 bps is especially noticeable when doing file transfers, and my C-128 (with only a 1571 drive) pretty much kept pace with the higher speed. Even on message bases, the Arotek C24 is welcome, as it fills your screen with text snappily. Working with 2400 baud is like everything else associated with computing - once you've seen it done faster you will never want to step back to slower speeds. The modem seemed no more susceptible to line noise than my Commodore 1670 or Supra Modem 2400 during the short time that I used it, even when working at 2400 baud. It's a side issue for 128 owners, but the Arotek manual echoes what I've heard from other C-64 users, that the 64 has a harder time than the 128 keeping up with 2400 baud telecommunications.

Should you buy an Arotek Minimodem C24? If you frequently perform file transfers with a service that supports 2400 baud then you'll appreciate the extra zip. The C24 is the least expensive way to get 2400 bps performance on the 128, and shortcomings aside, it works well. I currently share a "standard RS-232" Supra 2400 between my C-128, Amiga 500 and MSDOS portable. The Arotek performs about the same as the Supra on the C-128, but its direct connection to the 128's RS-232 port does not, of course, match up to these other two machines. If you share (or think that someday you will share) your modem with a computer other than your 128 then the extra money you spend on a "standard RS-232" modem and adaptor cable may be worth it to you.



available from the QBB's manufacturer to allow its use as a CP/M RAMdisk. The QBB's non-volatile nature can make your 128 semi-transportable, since with it and the right applications it can eliminate the need to carry a disk drive.

GEORAM

The remaining C128 RAM expansion option is the 512K GEORAM product from Berkeley Softworks, the developers of GEOS. Though previously a software vendor, Berkeley introduced GEORAM as an alternative to the somewhat scarce 1750 RAM expander because GEOS is severely crippled without RAM expansion. Unfortunately GEORAM is not compatible with the 1750 and the only 128 software that supports GEORAM is GEOS and GEOS applications. GEORAM is significantly cheaper than the 1750, at \$125 compared to the typically \$200, 1750 pricetag.

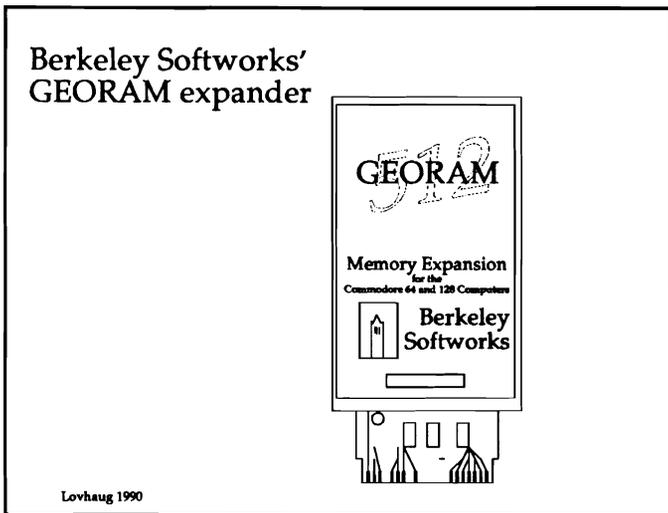
Conclusions

The worth of any or all of the RAM expansion alternatives mentioned above has a great deal to do with what you do with your 128 and the software you have chosen to do it with. Many 128 owners will be perfectly contented with their computers without ever adding additional RAM to their machines. However, for some specific C128 applications RAM expansion can make using the 128 much faster and easier. In particular, graphics and powerful productivity applications can be greatly enhanced with software that supports some form of RAM expansion. Environments such as GEOS and CP/M that must access a disk drive often make excellent candidates for a RAMdisk speed boost.

My recommendations? Because the 1750 is the best supported RAM expansion alternative as far as C128 software is concerned, it is the option I can most heartily recommend. The 1764 is pretty much the same story, although you may want to consider expanding it to a full 512K (or buying Software Support International's "1750 Clone" which has already had this 256K to 512K upgrade installed). Unless you plan to expand the model 1700 REU beyond its 128K you should be aware that some software (most importantly GEOS) will not work with it at all.

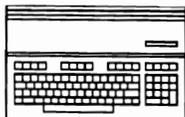
We can make a strong case for video RAM expansion and the purchase of a Quick Brown Box as well. Video RAM expansion in concert with the proper software noticeably increases the color and resolution abilities of the 128, and is required to allow the 128 to edit and display the multi-color 640 x 400 images created by IPaint. Users of The Write Stuff word processor and other QBB supporting packages, programmers, and people who take the 128 on the road with them will love the Quick Brown Boxes.

Interestingly enough, the Commodore RAM expansion unit, expanded video RAM, and the QBB, generally do not have any difficulty co-existing with one another. Of course using a Commodore REU and a QBB together requires the addition of a switched multi-slot expander for your cartridge port, but other than added expense we are not aware of any other problems. In fact, we have been using this "triad" of RAM expansion units at TC128 for quite sometime. Unfortunately, Berkeley's GEORAM does not continue this trend of "peaceful co-existence". In our tests, GEORAM would not function in concert with either the 1750 or the QBB on an Aprospand expander. Video RAM expansion does not conflict with GEORAM, but since GEOS is the only currently available application that is compatible with GEORAM, and GEOS itself does not provide any support for the expanded video RAM, it's a barren combination. Since all of GEORAM's RAM expansion abilities are duplicated in the more compatible and versatile 1750, I cannot recommend GEORAM, even when taking into account its lower price.



Beyond the compatibility question, perhaps the chief concern of potential GEORAM buyer is how fast does it operate with GEOS. Commodore's REU's, blessed with its special DMA circuitry provide impressive results when used in context with GEOS. Since the Berkeley expansion unit does not provide this kind of circuitry, it needs a special version of GEOS (supplied with the GEORAM) designed especially for use with GEORAM. The table below reflects comparative tests of the speed of GEORAM vs. the 1750 in several GEOS related situations. While the tests themselves show that the 1750 is indeed faster than the GEORAM by a considerable empirical margin, I'd question the significance of that speed increase in terms of normal use. After all, only in the cases of the largest files will the difference be anything more than a second or two.

Test	GEORAM	1750 RAM Expander	Speed Advantage 1750
Diskcopy 1571 GEOS disk to RAM 1571	2:27	2:16	8%
Filecopy 8 files (84K) 1571 to RAM 1571	1:06	1:01	8%
Filecopy 8 Files (84K) 1581 to RAM 1571	:39	:32	22%
File Duplicate geoPaint (50K) within RAM disk	:07	:05	40%
File Duplicate "big file" (147K) within RAM disk	:16	:10	60%
Full Page Preview of geoPaint picture in RAM disk	:08	:08	0%



expanders are the most widely supported RAM expanders as far as software is concerned. C128 applications typically utilize Commodore's RAM Expansion unit one of two ways. The first method uses the RAM expander to hold "overlays" (extra portions of a computer program that could not be accommodated in the 128's regular memory) or data. These are transparently swapped into memory as an application requires them. Paperclip III, Pocket Filer 2, Bible Search, and Big Blue Reader employ this method. Other applications treat the RAM expanders as if they were a super-fast imaginary disk drive or "RAMdisk". RAMdisks are established either through a custom implementation such as those found in GEOS, CSDOS, Pocket Planner 2, or Pocket Writer 2 and 3; or through the use of Commodore's freely distributable RAMdisk emulation software called RAMDOS. Commodore's RAMDOS allows you to use most Commodore DOS commands to perform file operations and maintenance using the RAM expander. Some programs such as Voyager Mindtools' IPAINTE and Dialogue 128 even allow you to choose whether you wish to use RAMDOS or the program's custom RAM expander support. Another fully-functional RAMdisk is built in to the C128's implementation of CP/M, allowing a 17xx expander to be used with any CP/M program you run on the 128.

RAMdisks generally accept the same commands used with "real" disk drives, but since they are really only moving memory around with the aid of the REC they are very, very fast, quicker than even a hard drive. RAMdisks are a great way to get more done more quickly with a 128, but they have their limitations:

1. They can hold no more data than the size of the RAM expansion unit, a significant limitation with the 1700/128K REU.
2. Incompatibilities exist between different kinds of custom RAMdisks. Therefore you can't share data between GEOS, Pocket Writer, and CP/M within a single RAMdisk, because they all create their RAMdisks differently.
3. Many pieces of software contain no provisions to use any kind of RAMdisk and frustrate even knowledgeable attempts to install a separate RAMdisk utility such as RAMDOS.
4. A RAMdisk disappears forever when the power switch is turned off, so data and programs must always be loaded onto the REU's RAMdisk from the computer or a "real" disk and saved back to a "real" disk at the end of a work session. The REU will hold its data however when the 128 is reset with the push button on the right side of the 128's case, allowing the RAMDisk to survive most system lockups and auto-booting of different software (as long as the new software doesn't automatically initialize a new RAMdisk).

Although Commodore's RAM expansion units were once the most widely distributed RAM expansion option for the 128, they are becoming difficult to find. Today, the most reliable source for the purchase of a new Commodore RAM expansion unit is Software Support International who will sell you a 1764 RAM expansion unit (without the extraneous power supply) upgraded to 512K for \$199. If you are willing to hunt through the used computer channels you may be able to come up with one for a lot less. Both the 1700 and 1764 can be upgraded to 512K, although the process requires the installation of additional memory chip sockets and/or soldering skill. If you'd like to really pack it full,

a technique has been discovered that will allow all three RAM expansion units to be upgraded to as much as 2 megabytes!

Video RAM Expansion

The next time you are talking with a fellow 128 owner, try stumping him or her with this question: "How much random access memory does a standard un-expanded flat C-128 come equipped with?" The vast majority of 128 owners, rookies and veterans alike, will most likely answer 128K, to which you can smugly reply, "Wrongo, bit breath! Every unexpanded flat 128 comes with 128K of main memory, and an additional 16K of 80 column video RAM for a grand total of 144K of RAM." Then once you have recovered from the bodily injury inflicted upon your person by the victim of your trick question, you can go on to explain the wonders of expanding that video RAM to 64k.

Software Support International's VDC RAM Upgrade Daughterboard



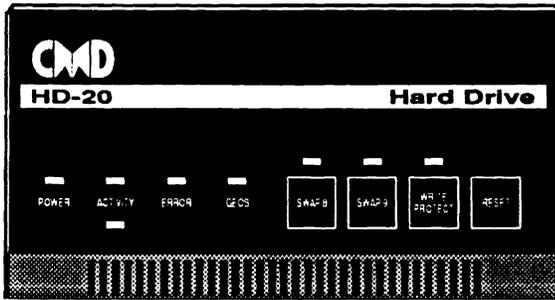
Lovhaug 1990

Yes, every 128 has as a little extra memory tucked away in the exclusive suburbs of its memory map. This memory can only be accessed by the 8502 microprocessor via the external video communications registers of the 128's 80 column 8563 Video Display Controller (VDC) chip at \$D600 and \$D601. This would seem to make the 128's VDC RAM almost useless for most applications, and indeed it would be, if it were not for the wonders of the 128's Video Display Controller itself.

The 8563 is an extremely powerful piece of silicon. Beyond handling the somewhat mundane job of controlling the 128's 80 column text display, the 8563 is capable of producing high resolution, color, bitmapped graphics displays. In fact the 8563 can produce overscan interlaced bitmapped graphics displays of up to 752 x 600 pixels, (however, due to the limitations of the monitors most Commodore 128 own, displays over 640 x 400 pixels have questionable use). Additionally, the 8563 has an incredibly fast block copy and move facility that in some respects mimics some of the "bit-blit" capacities found in graphics computers such as the Amiga. Finally, the 8563 was designed so that you could maintain alternative screen displays (either text or graphics) or alternative character sets and instantly "program" it to switch to your alternative display or font information. All these capabilities give the 128 the most powerful and flexible video display system of any eight bit microcomputer that has



Taking your 64 and 128 into the 1990's



CMD Hard Drives

- Four modes of operation: 1541, 1571, 1581 emulation modes for compatibility and Native mode with up to 16 Mb partitions.
- Built-in *GEOS™ and *CP/M™ compatibility.
- Standard capacities of 20, 40 and 100 Megabytes with custom capacities in excess of 200 Megabytes.
- Serial bus interface supports Standard Serial as well as high-performance Fast Serial and JiffyDOS protocols.
- 3 1/2" SCSI technology allows for quiet, cool reliable operation in a compact case about the same size as a 1581 drive.
- Can be interfaced with Amiga, IBM - compatible and Macintosh computers, allowing you to take it with you when you upgrade.
- Built-in real time clock for time and date stamping of files.

* GEOS is a Trademark of Berkeley Softworks • CP/M is a Trademark of Digital Research

Performance

CMD Hard Drives are designed to be the fastest serial mass-storage devices available for the C-64 and C-128. When connected on the serial bus to a completely stock computer, HD series hard drives outperform all other serial bus drives (including the 1581). If your computer is equipped with JiffyDOS, the HD delivers unbelievable speed, rivaling and in many instances outperforming even our competitor's parallel interfaced system. And when connected to the parallel port on CMD's RAMLink, HD series drives become as fast as the hard drives found on newer, more expensive machines.

Compatibility

Our specially designed operating system (HD DOS) gives HD series drives the ability to work with nearly all commercial software. This proprietary DOS fully supports all Commodore DOS commands, including block reads and writes, memory access commands, fast serial and burst commands. Complete compatibility with GEOS and CP/M coupled with the HD's ability to use partitions which emulate the popular 1541, 1571, and 1581 disk drives, result in a level of compatibility unparalleled in any other hard drive system available for C-64 and C-128 computers. Another feature of the HD's operating system is that it is stored on the hard drive itself and not in ROM. This means that DOS may be upgraded simply by sending the user a disk which is then loaded onto the drive.

Ease of Use

From the moment you plug in and begin using the HD, the simplicity of design and operation become apparent. We provide you with the tools to easily organize and access the large amounts of data you will be storing. Partitions and system defaults are effortlessly handled through our HD-TOOLS software. Use with GEOS is accomplished easily by installing a new CONFIGURE file on your GEOS boot disk. Utilities have been provided to automatically set the GEOS clock from the HD's Real Time Clock and to quickly navigate through and copy files between the HD's partitions. Easy to remember commands allow subdirectories to be created within native mode partitions and within other subdirectories, and enable easy movement to other areas within the drive.

HD-20 \$599.95 • HD-40 \$799.95 • HD-100 \$1299.95

- **Built-in two drive file copier.** Copy PRG, SEQ, REL and USR files between two drives of any type or to and from REU's. Great for quick backups and moving programs and files between 1541, 1571 & 1581 drives and REU's.
- **REU support.** The JiffyDOS commands now fully support Commodore REUs (RAM Expansion Units) running under RAMDOS. Access your REU just like a disk drive without having to load special wedge utilities.
- **Enhanced text screen dump.** Automatic screen mode recognition and printing of uppercase/graphics & lowercase characters.
- **Redefinable 64-mode function keys.** Allows you to easily redefine the function keys to suit your specific needs.

JiffyDOS™ Version 6.0

The Ultimate Disk Drive Speed Enhancement System

- **Ultra-fast multi-line serial technology.** Enables JiffyDOS to outperform Cartridges, Burst ROMs, Turbo ROMs, and Parallel Systems – without any of the disadvantages.
- **Speeds up all disk operations.** Load, Save, Format, Scratch, Validate, access PRG, SEQ, REL, USR & direct-access files up to 15 times faster!
- **100% Compatible – or your money back.** Guaranteed to work with all your hardware and software, including copy-protected commercial programs.
- **Uses no ports or extra cabling.** The JiffyDOS ROMs upgrade your computer and drive(s) internally for maximum speed and compatibility.
- **Easy Installation.** No experience or special tools required for most systems.
- **Supports C64, 64C, SX64, C128, C128D, 1541, 1541C, 1541-II, 1571, 1581 & more**
- **Can be completely switched out.** If ever necessary, the flip of a switch returns you to a 100% stock configuration – without resetting or powering down.
- **Built-in DOS Wedge.** Plus 17 additional commands and convenience features, including file copier, screen dump, directory menu, and single-key load/save/scratch.

- **Supports CMD HD Series Hard Drives and RAMLink.** Enhances the performance of CMD's new line of integrated C64/C128 products.
- **1581 support.** Copy programs and files from 1541 and 1571 drives to any partition on your 1581. Select partitions easily with just a few keystrokes.
- **Quick printer output toggle.** A simple 3-key command switches output from screen to printer and back with ease. Eliminates having to type the complicated OPEN#4,CMD#4 and PRINT#4,CLOSE#4 command sequences.
- **Adjustable sector interleave.** Enables you to increase disk-access performance even with hard-to-speedup software.

Hard Drive Ordering Information

Add \$25.00 for shipping and handling per drive. All orders shipped UPS ground. Please allow 3 to 6 weeks for delivery. U.S. prices quoted above.

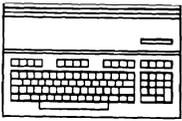
Canadian: 20Mb - \$649.95 • 40Mb - \$849.95 • 100Mb - \$1349.95 Shipping: \$30.00
Foreign: 20Mb - \$699.95 • 40Mb - \$899.95 • 100 Mb - \$1399.95 Shipping: \$35.00

JiffyDOS Ordering Information

C-64 SX-64 systems \$59.95; C-128 C-128D systems \$69.95; Additional drive ROMs \$29.95
Please specify computer (w/serial number) and drive(s) when ordering.

Add \$4.50 shipping/handling per order, plus \$4.00 for C.O.D. APO, FPO, AK, HI, PR & Canada. \$12.00 add'l for other overseas orders. MA residents add 5% sales tax. VISA, MC, COD, Check, Money Order. Allow 2 weeks for personal checks. U.G. & Dealer pricing available. Call or write for more information. Version 5.0 owners please call for upgrade information. Prices subject to change without notice.

Creative Micro Designs, Inc. 50 Industrial Dr., P.O. Box 646, East Longmeadow, MA 01028
Phone: 413-525-0023 FAX: 413-525-0147



Rumor & Progress Continued

Rumor From Page 1

With this urgency in mind, an increasing number of 128 owners are asking me these questions:

1. *Is it wise to invest any more money in 128 hardware, or should I begin saving for a new computer system?*

2. *Regardless of my future computer purchase plans, I want to make use of my 128 for a couple more years; what products should I consider purchasing while they are still available?*

These are valid questions. And when you consider that for most of you the purchase of your 128 was a major investment, they have serious financial ramifications. While I cannot provide a single set of answers which will apply to all of you, over the years I have gotten to know a lot about TC128 readers and C128 owners in general, and therefore I feel very confident about offering the following advice:

First, evaluate what you do with your current 128 system. Identify what you feel are the honest strengths and weaknesses of your computer setup. If you are like most 128 owners your list of positive will include the versatility of the 128's low cost software base, and the ease at which beginning and intermediate programmers can become proficient. Typically 128 complaints will focus on the relative slow speed and low capacity of the system, particularly when it comes to data storage. Also, many 128 owners, despite their appreciation for the quality and affordability of the 128's software base, find themselves wanting to utilize software which is available for mainstream platforms such as the Macintosh and MSDOS computers. After getting a handle on what you feel are your likes and dislikes about the 128, you have the basis for making some decisions about your future.

If you decide the purchase of a new computer system is in order, I suggest that you address your perceived 128 deficiencies with your new system. With the experience you have gained with at least one computer system, you should be able to make better informed judgements as to what is important as far as your computer usage is concerned. But be forewarned, replacing your 128 will not be cheap, especially since many of your desires may take quite a bit of computer. In addition, the trend in computer sales and packaging today has definitely moved away from the component or modular approach (like the 128) to more of a system approach. Items such as hard disks, monitors, and memory are sold in bundles immediately with the system rather than later as add-ons.

Naturally, this adds to the upfront costs of a new computer system. Additionally, unlike the upgrade from the C64 to 128, the dissimilar nature of many upgrade options to the 128 will preclude the use of most 128 peripherals directly with a new computer. Both of these factors mean that while the price of some extremely powerful computing power is eminently more affordable now as opposed to several years ago, it is a sure bet that you will spend quite a bit more to replace your 128 than you did to purchase it initially. With this in mind

The most inexpensive computer you will ever own is the one that is paid for.

my recommendation is that you take your time and do a good deal of research concerning your next computer purchase. Especially since you already have a perfectly functional computer, and the time-honored tradition of today's technology being priced lower tomorrow shows no sign of disappearing. (As an example, the HP compatible laser printer which cost \$2500 two years ago, now retails for less than \$1000). After all, the most inexpensive computer you will ever own is the one that is already paid for.

For those of you who are not ready (or may never be ready) to make any major decisions concerning a new computer, you have lots of choices as well. First, while there are plenty of 128 products available today, that may not be the case next year. Therefore, you should definitely consider buying those items which will significantly enhance the productivity and pleasure you derive from your 128 during the next few months. In addition, with a number of 128ers considering the purchase of new computers, the used hardware and software market will be ripe with all sorts of attractively priced goodies. So in many respects the so-called decline of the 128 marketplace could be a great boon for 128 bargain hunter.

It is our feeling that most of you will be using your 128s for some time to come, even if you have bought (or are planning to buy) another computer. Regardless of the status of your "upgrade" plans, TC128 will continue to be your source of 128 information. In our view dedicated 128 journalism in the 1990s needs to address techniques for using the 128 in conjunction with other (and multiple) computer systems.

Price & Progress Continued from Page 1...

telecommunications networks. These network-exclusive dedicated terminal programs allow you to bypass obtuse menu structures and to play in special gaming environments. While most networks are concentrating their development on the Macintosh and MSDOS computers, one system now has a custom front designed for the C128, called "Wizard". Written by Bill Coleman of GEOS programming fame, Wizard has a mouse menuing system which automates getting on GENie and then aids with online activities such as checking EMail, downloading files, and reading messages in various areas of the system. Wizard is now available for download in GENie's Commodore Roundtable section, which is at Page 625 for you GENie shortcut fans. Look for file number 7919 in Library 40. If you're not a GENie member and would like to be, you can write to GE Information Services, 401 North Washington St. Rockville, MD 20850 or sign up on-line with your credit card by setting your modem to half-duplex, 300 or 1200 baud, and dialing 1-800-638-8369. At connect, enter HHH and then when asked for a U# put in XTX99656, GENie and hit RETURN again. Tell'em you heard about it in TC128.

Why You May Want to Plug Your Modem Into the "Wrong" Port

One more C-128 wall has been pulverized lately with the issuance of the newest hardware product from Dr. Evil Labs (of Stereo SID fame). It's called Swiftlink 232 and its job is to allow the 128 (or the 64 for that matter) to transfer data reliably at greater than 2400 bps speeds. With this little hummer plugged into your cartridge port you can at last use that 9600 baud modem which you chucked into the closet when it wouldn't work the 128. What, you say that your closet is so full of old 300 baud modems that you can't find that high speed model? Well what if we told you that an inexpensive null-modem adaptor attached to the 128 makes porting files between unlike systems possible, and that at the high transfer speeds Swiftlink makes possible, this method can beat disk format transfer to the finish? Furthermore, RS232 transfer works with any machine that has a serial port, even those chic little portables without disk drives.

Price for the Swiftlink is \$29.95. It plugs into the cartridge port (not the user port which one normally thinks of as the "modem" port of the 128) and then terminates in a 9 pin "IBM AT" style connector, a type that is increasingly used by the better MSDOS clones and portables. If you already own a standard RS232 modem, an off-the-shelf 9-to-25 pin modem cable should work, but Dr. Evil Labs will sell you a perfectly good one for an attractive \$5.50 price if you order it with the Swiftlink. One way to look at this product is as a competitively priced way to get a Commodore-to-standard-RS232 interface with some very interesting extra capabilities, even if you are planning to use it today with a 2400 bps modem.

Continued on Page 3

**SD137.COM**

This is a versatile directory program that can show files in all user areas, on all drives, files in libraries or archives, and files with the system attribute set. It can send output to a file or printer. SD137 replaces both DIR and DIRSYS.

ERASE.COM (version 5.3)

This ERASE replacement allows scratching files in user areas other than the current one and asks for confirmation before erasing Read-Only files.

QL41.COM

Quick Look replaces both TYPE and DUMP. QL can view binary as well as text files. If a file is crunched or squeezed, or is a member of a library, QL will unpack it before displaying it. It features random access paging through files, a string or hex byte search, memory display, and can extract files from libraries.

LD.COM and CREATE.COM

Library Disk opens LBR files and causes CP/M to treat them as if they were disk drives. If used with CCP+, the library name is displayed in the prompt with the directory name. This gives the effect of working in a disk partition or subdirectory. CREATE, which I've renamed to MKDIR, makes empty libraries.

CHN31.COM

CHN creates a COM file from a chain of CP/M commands. CHN can take on many of SUBMIT's functions.

ARCOPY.COM version 2.1

This program replaces PIP, RENAME, and TYPE. It's feature packed, including copying to/from different user areas, to the printer/console, and renaming or moving files (erase after copy).

DA.COM

Directory Attributes can set or reset the system, read only, archive, or special 1 - 4 attributes for a file or group of files. It also functions as a directory program.

HIST.COM

This history program for CP/M+ quickly recalls any of the last 20 commands typed to the system. Command lines can be edited and reused.

SETDIR.COM

A CCP+ utility for loading directory names into memory.

LOADKEY2.COM

This program will load a custom keyboard file created with KEYFIG and saved with SAVEKEY2.COM.

ZDE13.COM

ZDE is the latest version of VDE, the Video Display Editor. ZDE uses Wordstar commands, is memory resident, and includes programmable macros. It is better than many commercially available word processors for 128 mode and is by far my favorite C128 editor.

E.COM

This is a command file created with CHN.COM. Its syntax is:
E du:filename

The definition of E is:

```
LOADKEY2 ZDE.KEY1ZDE13 $1 $2!LOADKEY2 DEFAULT.KEY
```

What this does is load a special keyboard file that places ZDE commands on the keys I want them on, then loads ZDE along with the file to be edited. After using ZDE, standard keyboard definitions are loaded back into the system. This method can be used to customize any application or program by automatically loading keyboard layouts that change, simplify or combine commands and assign them to almost any key. I use a similar CHN command file for loading terminal software. This is the command line:

```
LOADKEY2 TERM.KEY!MEX114 $1!CONF  
BAUD=75,FEEL=0!LOADKEY2 DEFAULT.KEY
```

This line loads the keyboard definition file TERM.KEY, then loads MEX114. When I quit MEX, CONF resets the system baud rate to 75 and LOADKEY2 reloads the standard keyboard layout. All CP/M terminal programs I've seen set the system baud rate to 1200, but don't reset when finished. If you think the system is slow at 300 baud, you don't want to see it outside of a term program at 1200 baud. If you use this command, you never will.

CP/M can be plenty of fun to tinker with. Both the operating system and applications give ample opportunities for customization. Once you get CP/M up to speed and discover some of the excellent software available for it, you might find your new boot disk spinning in the drive more often.

Raise the Commodore Graphic ante
from 40 to 80 Columns with . . .

ANTE UPTM

— THE BASIC 8 GRAPHICS TRANSLATOR —

NOW . . . convert GEOS, GEOS 128, Koala, Doodle, Computereyes and Print Shop graphics into 80 column, Basic 8 compatible pictures!

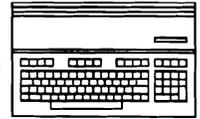
"A very slick utility"

"If you use Basic 8, you need Ante Up." --- Twin Cities 128

Price: \$19.95 + .85 postage

Starfollower Productions

4224 Bakman, N. Hollywood CA 91602



You can arrange the code on your function ROM in almost any way you like, as long as you reserve the first ten bytes for information the system needs:

Bytes	Purpose
00,01,02	Cold start entry point
03,04,05	Warm start entry point, not used by the system
06	Cartridge ID (\$01 - \$FF)
07,08,09	ROM identification string ("CBM")

The "C-128 Programmer's Reference Guide", page 472, mistakenly calls the cold and warm start entry points "vectors". The system does not jump "through" an address stored at these locations (the definition of a vector). Instead, the cold and warm start locations provide a place for the system to jump "to".

The system only uses the cold start entry point. It should contain a JSR or JMP to code on the ROM you want executed on startup or an RTS if you don't want any code on the ROM to be executed upon startup. The warm start entry point is left for your convenience--you could modify the system IRQ routine to jump there during a warm start of the machine, for example.

You have several choices on how to arrange code on the ROM and how it is to be accessed. In the example discussed below, for example, the ROM merely acts as a place to store code--everything is copied into RAM before being executed. The main reason for this is the other system ROMs are needed, so the function ROM must be switched out.

With the powerful bank-switching capabilities of the C-128 and some careful planning, you can have code execute from the function ROM as well as access the system ROMs.

BASIC-8: An Example Function ROM Startup Sequence

When I first became interested in function ROMs, I looked around for good examples of ROM startup routines. One of the most useful for me to study was that of BASIC-8⁶. I have reproduced the startup routine here (with permission and thanks to Lou Wallace, one of BASIC-8's authors) in a slightly modified form to make it a bit more robust and therefore be a slightly better model. Let's take a look at the code:

8000	4C 0A 80	JMP \$800A	;Cold start entry
8003	60	RTS	;Warm start entry (not used)
8004	EA	NOP	
8005	EA	NOP	
8006	BB	???	;ROM ID byte
8007	43	???	;ROM Signature "CBM"
8008	42	???	
8009	4D	???	

These first ten bytes are the startup bytes the C128 expects to see. Note that an RTS was placed at the warm start entry point--it provides an easy way to leave the ROM (used at \$800E below). The ROM ID of \$BB was an arbitrary assignment.

800A	A5 91	LDA #91	;Check to see if <Control> key is down
800C	C9 FB	CMP #\$FB	
800E	D0 F3	BNE \$8003	;if not, exit
8010	A0 00	LDY #00	;Copy rest of routine to cassette buffer
8012	B9 20 80	LDA \$8020,Y	;Code needs to be in RAM to execute
8015	99 00 0B	STA \$0B00,Y	;since ROM is switched out near the end
8018	C8	INY	
8019	C0 52	CPY #\$52	;52 bytes
801B	D0 F5	BNE \$8012	
801D	4C 00 0B	JMP \$0B00	;Jump to beginning of code

This section first checks to see if the <Control> key is being held down. If it is, then the user must want to have BASIC-8 turned on so we continue. If not, we stop here. The next routine copies the rest of the startup code to low RAM. This is done because the ROM is switched out near the end of the routine to return things to normal. I am not completely sure that this is necessary as it seems that Phoenix/BASIC would turn the ROM off anyway later. Some more research needs to be done here.

8020	78	SEI	;Disable interrupts
8021	AD 00 FF	LDA \$FF00	;Save current MMU configuration
8024	85 FE	STA \$FE	
8026	A9 17	LDA #\$17	;Switch in top 16K of function ROM
8028	8D 00 FF	STA \$FF00	

This section does two interesting things. First, it saves the current MMU configuration in case the ROM is being called from within an unusual bank configuration. Next, it switches in the top half of the 32K ROM. This is necessary because the Kernel treats a 32K ROM like two 16K chunks during initialization. Since we are using the 32K space as one big ROM, the reset routine only found one "CBM" signature in the lower 16K. Phoenix then turned that 16K space on before jumping to the cold start entry point.

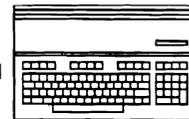
802B	A0 00	LDY #00	;Copy default printer definitions
802D	B9 00 81	LDA \$8100,Y	;on ROM
8030	99 00 0E	STA \$0E00,Y	;to RAM
8033	C8	INY	
8034	D0 F7	BNE \$802D	
8036	A0 00	LDY #00	;Set up to copy BASIC-8 Kernel and
8038	84 FA	STY \$FA	;startup BASIC program
803A	84 FC	STY \$FC	
803C	A9 82	LDA #\$82	;Source = \$8200 (ROM)
803E	85 FB	STA \$FB	
8040	A9 13	LDA #\$13	;Destination = \$1300 (RAM)
8042	85 FD	STA \$FD	
8044	A2 80	LDX #80	;80 pages
8046	B1 FA	LDA (\$FA),Y	
8048	91 FC	STA (\$FC),Y	
804A	C8	INY	
804B	D0 F9	BNE \$8046	
804D	E8 FB	INC \$FB	
804F	E8 FD	INC \$FD	
8051	CA	DEX	
8052	D0 F2	BNE \$8046	

This section does some standard memory moves to copy the needed code out of the ROM to the appropriate locations in RAM.

8054	A9 70	LDA #\$70	;Relocate start of BASIC to \$7000
8056	85 2E	STA \$2E	
8058	A9 03	LDA #\$03	;Put three characters in keyboard queue
805A	85 D0	STA \$D0	;to start BASIC program at \$7000
805C	A9 52	LDA #\$52	;"
805E	8D 4A 03	STA \$034A	
8061	A9 D5	LDA #\$D5	;"U"
8063	8D 4B 03	STA \$034B	
8066	A9 0D	LDA #\$0D	<RETURN>
8068	8D 4C 03	STA \$034C	
806B	A5 FE	LDA \$FE	;Restore MMU configuration (switch out
806D	8D 00 FF	STA \$FF00	;function ROM)
8070	58	CLI	;Turn on interrupts
8071	60	RTS	;Done!

This final section first relocates the start of BASIC to be above the BASIC-8 Kernel in RAM. It then pokes a RUN command into the keyboard queue to set up the execution of the BASIC startup program downloaded from ROM. Finally, the MMU configuration is restored and the routine is exited. If the ROM was entered during a system reset, control returns to the Phoenix routine. It

Continued on Page 20



With small pliers, bend the necessary pins away from the body of the new 4066. Don't try to make them completely horizontal, just enough to clear any underlying components. You can put a small piece of electrical tape over any component a pin is in danger of touching. Put the new 4066 on top of U27 in a 128 (or U20 in a 128D,) with the notch in the same direction. Solder the un-bent pins to the bottom chip. As shown in the illustration, for a 128 only pins 7 and 14 are un-bent. Use small gauge wire, wire-wrap, or single conductors from "ribbon" cable; to connect pins 2, 3, 9, and 10 to D4-D7 at the feed-throughs as shown (C128,) or pins 14-17 of character ROM U18 (C128D). Wire pins 1, 4, 8, and 11 to color RAM U19 pins 14-17. It will not matter if you scramble the data bits. For 128 installation, wire pins 5, 6, 9 and 10 together and connect them to pin 5 of U20. We're almost done.

On the underside of the board, locate pins 14-17 of U19. Remember that you'll be looking at the chip upside-down and backward. Follow the traces from those pins to resistor package RP7. These traces are very close together and you may need magnification to make certain you have the right ones. Use a single-edge razor or X-acto knife to cut those four traces at RP7. Other resistors in RP7 are used elsewhere and if you break the wrong connections your 128 won't function properly. An alternative to cutting on the bottom is to snip pins 14-17 of U19 on the top side and bend them up away from the board, functionally equivalent, but crude.

Make sure no little bits of wire or solder are left on the circuit board; blow on it, turn it upside down and shake it. Lay it on a newspaper and reconnect the power connector, video cables and keyboard. Power up in 40 columns and you should get the familiar start up message. If not, something's wrong. Power down and inspect your work. Look for solder "bridges." Did you use the right feed-throughs? Did you break the wrong RP7 connections? Correct the error(s) and try again. Enter the machine language monitor and examine the last 32 bytes of color memory:

```
.m fdbe0 fdbff (return)
```

Cursor up to that same statement and press the return key again. If the upper nybbles change, there's still something wrong; the upper nybbles of U19 aren't getting connected to the data bus. Maybe the connection between the new 4066 and pin 5 of U20 isn't right. Power down, correct any error(s) and try again. When you're satisfied that everything is wired properly, try filling that portion of color RAM with some value, say \$55 or \$AA:

```
.f fdbe0 fdbff 55
```

Examine that portion of memory with the ".m" command. If you get all \$55's or \$AA's then your modification is successful. Clear the screen with shift CLR/HOME and examine that portion of color memory again. Locations \$dbe0-dbe7 should contain the current character color while \$dbe8-\$dbff should retain the value you filled it with previously. As stated earlier, the text screen is cleared by printing blank lines, so those last 24 bytes are never touched. However, the 128 contains two 1K blocks of color RAM, the second being for multi-color bit maps. When a multi-color bit-map is cleared, that color block is filled with COLOR3 (multi-color2,) by a routine that fills whole pages of

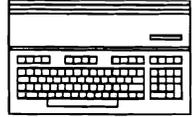
memory. Also, some programs may manipulate text colors in the same manner. Bear this in mind when stashing short programs or data in those last 24 bytes.

Having two 1K color blocks is not so complicating a factor as it appears to be. The processor changes bit 0 of location 1, to gain access to the multi-color block, only when actually drawing on the bit-map (DRAW, BOX, etc.). Then location 1 is set back to text color access. The bit-map routines do this byte by byte. Even split screens are no problem.

For strictly 80 column programs, the full 1K of 40 column color memory can be used, or 2K with color bank switching. (Indeed, all of the 40 column text screen can be used as well for an additional 1K. The last 24 bytes of text screen can always be used. This area, however, is normally accessible only in BANK 15 or BANK 0.) However, STOP-RESTORE clears both the 40 and 80 column screens, regardless of which is active. If you don't want to lose whatever you've stored in 40 column color or screen memory, you'll have to divert the indirect NMI vector so that only the 80 column screen is cleared. A program to do this is included here. It is useful even if you don't perform the hardware modification. The source listing for "Eighty-Only" is deceptively short. All the zeroes in the BASIC loader comprise buffers to hold RAM-resident copies of the NMI handler and the Screen Editor initialization routine, CINT. When Eighty-Only is initialized it copies those routines to the buffers (after switching to 80 columns), modifies the NMI copy to call the CINT copy, instead of the ROM version, and modifies the CINT copy so that only the 80 column screen is cleared when STOP-RESTORE is pressed, thus preserving anything in 40 column color and screen memory. The new CINT ignores the 40/80 key, always selecting 80 columns. The NMI indirect vector is pointed to the NMI copy. The ESCape vector is also changed, so that ESC-X, switching display modes, is ignored as well. (NOPs are included at this point. You may modify this portion to JMP to a routine to give ESC-X a new function.) The last step of the new CINT again changes the vectors, which were reset to their default values by earlier portions of the CINT and NMI copies. You can still change display modes with the GRAPHIC n statements, and RESET will set everything back to normal.

This I/O-resident memory is best used by "transient" routines, short programs stored elsewhere and moved into color RAM as needed. I have in mind those situations where there is no common memory as set by the MMU. As long as I/O is visible, you have a way back to the configuration you came from.

```
100 REM *** EIGHTY-ONLY ***
110 SA=4864:REM RELOCATING
120 BANK15:CK=0
130 READD:CK=CK+D:IFD=999THEN150
140 GOTO130
150 IFCK<>9957THENPRINT"*** ERROR IN DATA ***":END
160 RESTORE:AD=SA
170 READD:IFD=999THEN210
180 IFD=>0THENPOKEAD,D:GOTO200
190 GOSUB280:POKEAD,LB:AD=AD+1:POKEAD,HB
200 AD=AD+1:GOTO170
210 D=134:GOSUB280:POKESA+35,LB:POKESA+37,HB
220 D=80:GOSUB280:POKESA+57,LB:POKESA+59,HB
230 D=90:GOSUB280:POKESA+69,LB:POKESA+71,HB
240 IFRGR(0)<>5THENPRINT"SWITCH TO 80 COLUMNS.":GRAPHICS
250 PRINT"NEW NMI INSTALLED'SA'TO'SA+318
260 PRINT"TO ENABLE, SYS'SA
270 END
280 DD=SA+ABS(D):HB=DD/256:LB=DD-INT(DD/256)*256:RETURN
```



The Dynamic Keyboard

by: Anton Treuenfels

Many programming languages are designed in a way that separates the editing phase of creating a program from the execution phase which runs it. BASIC, which makes a distinction between "immediate" and "program" modes, is such a language. One result of this design is that a program is prevented from changing itself while it is running. In general this is a good idea, since among other things it guarantees that the executing program is the same one the programmer wrote. In the hands of a knowledgeable programmer, however, an occasional self-modifying program can be extremely handy.

On the 128 it is possible to allow a BASIC program to stop executing and then to restart itself without user intervention. While the program is stopped the interpreter is in immediate mode. New lines can be entered during this time, allowing a BASIC program to modify itself. The technique, known as "dynamic keyboarding", takes advantage of the interaction between the BASIC interpreter and the Kernel operating system. Whenever the interpreter needs to obtain a line of input from any source it calls an operating system routine known as BASIN. When input is expected from the keyboard, as it is in BASIC immediate mode, the BASIN routine stores keyboard input in a buffer until the <RETURN> key is pressed. The entire line is then returned to the interpreter as one piece. This behavior itself is not unique, many computers set aside memory to be used as a line input buffer. What is unusual about Commodore 8-bits is that they can use the screen as a keyboard line input buffer.

The process starts with the keyscan routine, which is executed 60 times a second during the system "heartbeat" interrupt. If a key is being pressed a character code for it is placed in a small keypress buffer. Whenever the BASIN routine is called to obtain a line from the keyboard it spends most of its time watching that buffer to see if anything is in it. Any character found is removed, and if it is not <RETURN> it is sent to the screen. If the character is <RETURN>, the BASIN routine retrieves the line from the screen and gives it to the caller (usually the BASIC interpreter or the built-in machine language monitor). The fact that BASIN sends all characters except <RETURN> to the screen during keyboard input is what gives Commodore machines such remarkably flexible screen editors relative to many other 8-bit computers. The screen can be used to buffer as many lines as can be displayed at once. Since some characters perform actions such as moving the cursor, inserting or deleting characters, entering or exiting quote mode, a line on the screen can easily be edited into any form desired. Only when <RETURN> is pressed are the contents of the logical line the cursor is on, up to the last non-space character, handed over by BASIN. The key to the dynamic keyboard technique is realizing that by using POKE and PRINT statements BASIC programs can place characters into the keypress buffer and onto the screen as easily as system routines do. The following short program demonstrates the technique:

```
10 PRINT CHR$(147);"GOTO 10"
20 POKE 842,19
30 POKE 843,13
40 POKE 208,2
50 END
:REM IMMEDIATE MODE INSTRUCTION
:REM PRESS <HOME>
:REM PRESS <RETURN>
:REM TWO KEYS PRESSED
:REM STOP EXECUTION
```

What this program does is print "GOTO 10" on the top line of the screen, place <HOME> and <RETURN> characters in the keypress buffer, and stop. Next the BASIC interpreter returns to immediate mode, prints "READY." a few lines down on the screen, and calls BASIN. BASIN notices that the keypress buffer is not empty and retrieves the first character. Since <HOME> is not <RETURN> it is sent to the screen, which causes the cursor to return to the top line. The next character is <RETURN>, so BASIN hands the line "GOTO 10" to the interpreter. The interpreter tokenizes the line and, since it is not preceded by a line number, executes it. This causes the whole cycle to begin again, achieving with great effort all the useful effect of the following program: 10 GOTO 10

There are additional considerations that turn the dynamic keyboard technique into something really useful. More than one line can be printed before the program stops. Any line preceded by a line number will be added to the program text instead of being executed. Variables can be printed as well as literal strings. It is possible to print variables representing line numbers and program statements on the screen, print a GOTO statement under them, and then prime the keyboard buffer so that the first line is added to the program and the program restarted. Since restarting by GOTO does not destroy the contents of variables, the string and line number variables can be changed and the process continued as long as desired.

The "HEXIFIER" program below scans any range of the C128's memory and creates DATA statements representing that range. The program also creates a "POKER" program necessary to put that information back into memory. This can be used to add DATA statements representing sprites or machine language routines to BASIC programs. A convenient method of using "HEXIFIER" is to LOAD it, LIST it on the 80-column screen and then LOAD the target program to which the DATA statements are to be added. At this point there is a copy of "HEXIFIER" buffered in 80-column screen memory and a copy of the target program in BASIC program text space. Add "HEXIFIER" to the target program by moving the cursor to the first line and pressing <RETURN> on each line. Once the lines have been added, type "RUN 60000" to create the DATA statements and the "POKER" program to read them. Once all the lines have been created, type "DELETE 60000-" to get rid of "HEXIFIER". SAVE the target program. "HEXIFIER" has been arranged in such a way that its first line should be on the top line of the screen after the second LOAD. This first line contains several variables that must be properly set. "BN=" the bank of memory to scan, "AD=" the starting address of the scan, "SZ=" the number of bytes to scan, "SI=" the number of bytes to include in a single data statement, "LN=" the line number of the first statement added, "LI=" the number to add to get the next line number, and "BN\$=" and "AD\$=" the names of variables in which "POKER" will store the bank and starting address of the data, respectively (the names should not be "A", "B", "I" or "J", which are used internally by "POKER").

Continued On Page 22



The Lt. Kernal Hard Drive: How It Works and How It Got Its Name

by: *Bill Brier*

Adapting a hard drive to work with a C-128 is not easy. Most hard drives do not have the on-board microprocessors present in Commodore disk drives. This means that a disk operating system (DOS) must be developed that will cause the hard drive to emulate a Commodore floppy drive with a maximum of transparency. However, since even a small hard drive has tremendous storage capacity, the DOS must also provide a means of partitioning the drive and managing the files if the unit is to be truly useful. The DOS should implement a command structure that will not be burdensome to the operator. Ideally RAM must be provided for the DOS, as it will be constantly executing while the system is in use.

A physical connection must be provided between the drive and the computer capable of high-speed data transfer. One method used by many non-MSDOS hard drives is a type of parallel data bus known as the small computer system interface (SCSI, pronounced "scuzzy"). Since the C-128 is not provided with a SCSI port at the factory, a route to the SCSI interface itself must be provided.

The Xetec Lt. Kernal uses unique strategies to integrate a hard drive into the C-128, both in terms of hardware and software. In the Lt. Kernal approach, the drive actually becomes a modification and extension of the computer, not an accessory. The DOS works in conjunction with the Kernal operating system and BASIC interpreter in such a manner that English language DOS commands may be issued directly from the keyboard. At the same time, the Lt. Kernal DOS transparently implements all Commodore file-type DOS commands that are relevant to a hard drive environment.

Networking is inherent to the Lt. Kernal concept and its implementation of the SCSI bus, permitting multiple computers and multiple drives to operate as a single integrated system. A Lt. Kernal setup consists of at least one computer, one host adaptor and the drive unit itself. Additional computers (up to a maximum of 16) are connected to the system by use of additional host adaptors and one or more multiplexers (MUX). Additional storage capacity over the basic 20 or 40 megabytes provided by the system can be attained by attaching more drives to the bus (up to a maximum of 160 megabytes).

The interface between the SCSI bus and the computer is provided by the host adaptor. This device plugs into the cartridge port of the computer and also requires several hardware connections to the computer's motherboard. A C-128 requires the installation of an MMU daughterboard, which plugs into the socket occupied by the 8722 MMU chip. A modification on the 128 board is required to retain the burst mode functions available with 1571/1581 floppy drives. If you do not wish to make this modification, you must use a serial cable provided by Xetec which disables the SRQ line. The burst mode mod is required on a C-128D, and it can be performed without removing the motherboard from the chassis.

The host adaptor connects to the system address and data lines and thus becomes visible to the microprocessor (CPU) as another

device in the I/O block. By this means the parallel interface is implemented. The host adaptor also has 16 kilobytes of RAM divided into two 8 kilobyte blocks, one addressable in the \$8000-\$9FFF range and the other in the \$E000-\$FFFF range. Wait a minute, you say, the \$E000-\$FFFF range is the Commodore Kernel! That's right, and that's the reason why the drive is called the Lt. Kernal.

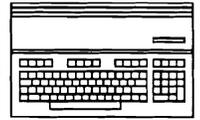
The host adaptor RAM is where the DOS executes. The upper 8K is actually a modified form of the CBM Kernel, referred to as shadow RAM. All of the standard jump table entries are there, but they lead into a complex series of software traps that are the gateway to the Lt. Kernal DOS. The traps are set by copying the CBM Kernel ROM to the host adaptor RAM and modifying it. Then the system is configured so that the modified Kernel replaces the ROM version. The result of this arrangement is that the traps analyze where I/O calls are supposed to be directed and if not intended for the hard disk, simply pass such calls on to the standard Kernel routines.

When an I/O request is intended for the hard drive the traps remap the computer, causing the standard operating system to vanish and be replaced by the Lt. Kernal DOS (a condition that Lt. Kernal programmers refer to as "banked out"). At the same time, the normal computer RAM in the range from \$8000-\$9FFF vanishes and that area becomes a window into the DOS, providing access to a variety of functions.

With the system banked out, the CPU can directly communicate with the host adaptor RAM (and thus indirectly with the SCSI bus). This results in I/O calls being handled at high speeds. In fact, the data transfer rate on the SCSI bus on a C-128 running in FAST mode is over 60 kilobytes per second, 12 to 15 times faster than the serial port operating in burst mode. A 25K program can be loaded in about one second, including the time required to search the disk directory and locate the file. The Lt. Kernal can copy a block of data (512 bytes) into system RAM in about 10 milliseconds. Even at one megahertz, the SCSI bus still runs at over 38 kilobytes per second.

When an I/O call has been completed the Lt. Kernal restores the system to its normal configuration (a condition referred to as "banked in"). The DOS is put back to sleep and the computer now appears to be normal to the CPU. The beauty of this arrangement is that no system RAM is ever disrupted, including the RAM that is underneath the Kernel ROM itself. The DOS will temporarily steal system RAM for some immediate mode commands but will always restore that RAM to its original condition upon completion of the function. No system RAM is ever touched during run-mode.

The Lt. Kernal DOS runs in one of two ways, depending on whether the computer is in immediate mode or in run-mode. In run-mode, the drive appears to be a fast and enhanced 1541, faithfully emulating the file-handling part of the Commodore DOS. All CBM file types are recognized, including an enhanced RELative file structure. The standard OPEN, CLOSE, PRINT#,



suppose that you are maintaining a parts inventory. You've created a RELative file whose records will hold the data about the parts (part description, number in stock, cost, etc.). You intend to index these parts by using the part number as a reference. When the user types in the part number you will look it up in a list or index, determine the record number in the RELative file, go to the file and retrieve the record. Obviously, the part number index should be in alpha-numeric order so as to facilitate finding the item. Also, to most rapidly respond to the user's request for data, the index should be in memory. But putting the part number index in memory gives rise to several problems. The first, of course, is keeping it sorted. BASIC isn't very good at such a task. Secondly, the index is maintained only so long as the power to the computer is. The third and most onerous problem is making the same version of the index available to all ports on a MUXed system.

The KEY file system solves all of these problems because it is entirely disk-resident and thus is "visible" to all ports on a MUXed system. The KEY file processor in the DOS takes over all aspects of managing the file and performs such tasks as inserting and removing entries ("keys"), keeping the index in order and providing search functions that return a record pointer if a match is found (there are also GREATER or LESSER searches, which allow a program to sequentially step through the index). Because KEY files are disk-resident they are relatively immune to harm from a power failure (unless the file is being sorted at the exact instant that the power goes down).

The internal structure of a KEY file is very complex but the management of such a file is easily accomplished from BASIC. Any given KEY file may be partitioned into a maximum of five separate indexes or directories (which have nothing to do with the disk directory). Keys up to 30 characters in length may be stored and depending on the actual key length, up to 65,535 keys may be indexed in a single directory (which means that a theoretical total of 327,675 keys could be stored in one file). Despite the enormous number of keys that may be stored in a file, the search routines can locate a single KEY and record number in about 1/2 second! It would take a binary search in compiled BASIC just as long to wade through maybe 1,000 keys in memory.

The other special Lt. Kernal file type, the contiguous data file, is a bit of an oddity. It is so-named because the disk blocks used to make up the file are contiguous to each other (CBM file types, on the other hand, are stored in scattered blocks in a manner similar to that used on any other Commodore drive). The DOS provides no specific handling routines for this file type and there are no references to it in the Lt. Kernal manual. However, there are DOS mechanisms available to a machine language programmer that allow the contiguous file to be used as a binary file (and thus loaded into the computer at incredible speeds), a sequential data file (such as for a word processor), or as a true random access file, complete with record arbitration (file locking) to allow multiple users simultaneous access.

Nothing loads your programs faster than:

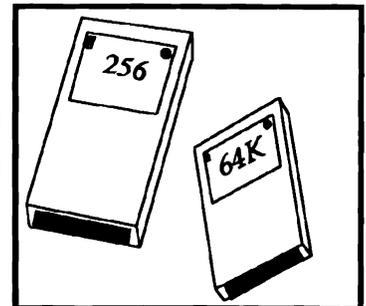
THE QUICK BROWN BOX

A new concept in Commodore cartridges! Store your favorite programs in a battery-backed cartridge for easy and instant access. Change the contents of the box as often as you wish. The Quick Brown Box accepts most unprotected and "frozen" programs including the only word processor that saves your text as you type: The Write Stuff! The box even co-exists with GEOS, and the Commodore RAM expansion units. Loader utilities included for both the Commodore 64 and Commodore 128.

- 32K Quick Brown Box.....\$75
- 64K Quick Brown Box.....\$89
- 128K Quick Brown Box.....\$149
- 256K Quick Brown Box.....\$199
- Utilities Disk.....\$6
- Q-Disk CP/M RAMdisk.....\$10

New Lower Prices And New Larger Quick Brown Boxes. Ask us about our new Write Stuff Combination Deals.

Add \$3 shipping and handling
 Mass. Orders Add 5% state sales tax
 C.O.D. add \$3, Overseas add \$5



BROWN BOXES INC.
 26 Concord Drive
 Bedford MA 01730
 (617) 275-0090 (617) 862-3675

Commodore 128

***INTO THE
90's***

***More Products from
Twin Cities 128:***

***Now paint
on 640 x 400
full color***

interlace screens

PLUS

***128 information in handy
book form right from the source***

I Paint, Earth's Only C128 Interlace Paint Program, is now shipping. Tired of monochrome only graphics, low resolution, tall and skinny pixel shapes, 8x8 color cells, and graphics that look like they were done on a 64? I Paint brings to the Commodore 128 the ability to create pictures that your friends will think could only be done on an Amiga. I Paint works with 640 x 400 pixels *all on the screen at one time* using a mouse controlled icon toolbox to select operations. While I Paint is BASIC 8 graphics format compatible, Rick Kane wrote it in a hybrid of BASIC 8 and machine language for improved speed. An 128D or 64K VDC RAM equipped flat 128 and a 1351 compatible mouse required. This breakthrough C128 program is available directly from us for \$39.95 postpaid.

500 C-128 Questions: Answered is over 160 pages of spiral-bound knowledge. It answers those "beginner's questions" in light of the latest Commodore developments, and goes from there to provide programming hints, an extensive discussion of disk and graphics formats, and a frank appraisal of which hardware and software solutions work - and which don't. Every 128 owner can find something useful in this brand new, 1990 book. **500 C-128 Questions: Answered** costs \$16.95 postpaid.

Twin Cities 128 Compendium Book #1 collects the first 18 issues of the magazine in an "all-in-one" spiral-bound volume. It's a great way to get "more TC-128" in one chunk and catch up on the history and development of "The best 8 bit computer ever made." You can order Compendium Book #1 for \$16.95 postpaid.

Twin Cities 128 The Commodore 128 Journal is the only American publication focusing on the Commodore 128 and its use and improvement. If you're wondering where to get the latest information and uncompromising reviews of new products, this is the place. Programmers can turn to TC-128 for articles by experts like Fred Bowen, Miklos Garamszeghy, Rick Kane, Loren Lovhaug, and Anton Treuenfels. Beginners can find articles that help them understand more. Subscriptions to TC-128 run for 6 issues and cost only \$12.50. If TC-128 guides you to one good new product or away from one poor one, you'll have saved the subscription price and more!

***Order Twin Cities 128/Voyager Mindtools products from:
TC-128 P.O. Box 11578 Minneapolis MN 55411***

The above prices are good for US orders only--foreign orders welcome--write us for current rates.

LET'S FACE IT - MAIL ORDER HOUSES ARE NOT ALL THE SAME!
SOFTWARE SUPPORT INTERNATIONAL
CONVENIENCE - FAST SERVICE - RELIABILITY - SUPPORT



Maverick V5 -- Next Generation Software For The Next Generation Hacker

The magazines and dealers will tell you that in the Commodore world, nothing copies protected software better than Maverick. But that's just the beginning of what Maverick can do for you.

Maverick is the most powerful and full featured complete archival utility system money can buy. There are so many features and

modules and utilities included that they can't be listed here. And our latest release, Version 5, is so far ahead of everything else that there simply is no competition, period. Bold talk? Try it for yourself -- learn first hand why Maverick is #1.

CO2475 • \$39.95

1581 Toolkit V2 -- The Best Reason Yet To Own A 1581 Drive

Owning a 1581 drive and not having the 1581 Toolkit V2 is like owning a sports car and not having gas in the tank. The 1581 Toolkit gives you all the utilities you need to take total control of your drive! In addition, features like REU support for fast one pass

disk copies, 64K video RAM support, and hyperfast read/write routines make this a "must have" package for any 1581 owner! Includes the famous book "The 1581 DOS Reference Guide," a \$14.95 value!
 CO1084 • \$34.95



The 1750 Clone -- Half A Megabyte Of Instant Power

Get a staggering 512K of RAM for your Commodore with our 1750 Clone! Designed to address the severe shortage of 1750 REU's, our 1750 Clone is made from authentic Commodore components and is indistinguishable from a 1750 to all your software!

You've got one of the finest computers ever built. But today's generation of advanced software wants more memory. Give your Commodore the room it needs to get serious. Step up to a full 512K of computing power with the 1750 Clone.
 CO1260 • \$199.95

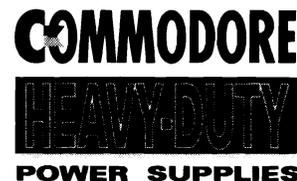
Commodore Heavy Duty Power Supplies

Having a hard time finding a decent replacement power supply for your Commodore 128? So are a lot of people. And when they do manage to find one, it costs a small fortune! Well, not any more....

Picture a heavy duty Commodore 128 power supply, made by Commodore themselves, that's fused, repairable, and

readily available from us -- and best of all, available at a *great price!*

Don't wait for your present power supply to fail (and possibly fry your precious computer in the process). Get your genuine Commodore Heavy Duty Power Supply today! CO0920 • \$54.95



Commodore 1541 or 1571 Disk Drive Repair Videos

Imagine having a technician actually show you, step by step, everything you've ever wanted to know about the inner workings of your Commodore 1541 or 1571 disk drives! Learn how to fix logic board problems, remove foreign objects from the drive, adjust drive speed and head

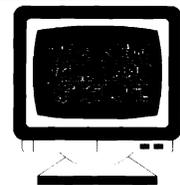
alignment and much more! Even see firsthand exactly how to install RAMBOard in your drive! Don't waste your time and money with the repair shop -- our video expert will show you how to do it yourself! 1541 Video: CO2567 • \$29.95
 1571 Video: CO2745 • \$29.95

Free* -- The Software Support International Fall 1990 Catalog -- Our Best Yet!

Just pick up the phone, call our TOLL FREE number, and we'll rush you a FREE copy of our huge brand new Fall 1990 Catalog!

It's jammed with hundreds of Commodore products, from hardware to software to accessories. The detailed descriptions and technical spec listings will provide you with all the information you need to

make the right choices for your specific needs. And the prices are among the lowest in the entire industry! How has Software Support International has grown over the last five years to become the most trusted mail order resource the Commodore user has? Just call our FREE phone number and ask for your FREE catalog -- and find out for yourself! * Overseas please send \$2.00 to cover S&H charges.



SOFTWARE SUPPORT
TOLL FREE LINE FOR USA & CANADA
800-356-1179
 Monday - Friday 9AM/5PM Pacific Time
 NEW - Saturday 7AM/3PM Pacific Time
INTERNATIONAL ORDERS
 206-695-1393
Some Hours As Above
NOW ACCEPTING FAX ORDERS
 206-695-0059
 24-Hours A Day!
TECHNICAL SUPPORT LINE
 206-695-9648
 Monday - Friday 9AM/5PM Pacific Time

If You Prefer, You May Mail Your Order To:
Software Support International
 2700 NE Andresen Road, Suite A-10, Vancouver, WA 98661

METHODS OF PAYMENTS • We accept money orders, certified checks, Visa, M/C, and Discover. Previous customers may also pay by COD or personal check. All monies MUST be paid in US funds.

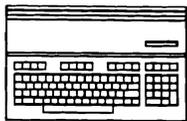
SHIPPING & HANDLING CHARGES - US/4 (48 states), FPO, APO, US Possessions: Please add \$3.50 per order. US shipping is usually by UPS ground. Fast UPS 2nd Day Air is available (US 48 states only) by adding \$2.00 per pound (1st lb.) and \$1.00 per pound (each additional lb.) more. Alaska & Hawaii: Shipping is by UPS 2nd Day Air. Please add \$8.50 per order. Canada: Software - \$4.00 for the first piece & \$1.00 for each additional piece per shipment. Canadian Hardware/Overweight orders & Foreign Countries: S&H varies per order - please call or write.

COD CHARGES • COD available to previous customers only in all 50 US states. Please add \$3.50 in addition to your S&H charges.

OTHER POLICIES • Washington State residents must add 7.6% to their order for state sales tax. Defective items are replaced at no charge, but must be returned to us postpaid within 30 days of invoice date. All in stock orders are processed within 24 hours. US (48 states) software orders over \$100.00 will be shipped 2nd Day Air at no additional charge above the normal \$3.50 S&H fee. All prices, policies, and specifications are subject to change without notice. All sales are final unless authorized by management.



We Accept VISA, M/C, & Discover



Overview of The Creative Micro Designs Hard Disk

by: Loren Lovhaug

Editor's Note: Since our cover story in Issue #27, a lot has happened with the CMD hard drive. At that time, we were using a preliminary "beta-test" release of the CMD's operating system. We tried with our preview coverage to describe both the capabilities which were present in the CMD hard drive, as well as those which we were told would be present in the release version of the operating system. Unfortunately, much to the consternation of TC128 readers who had read our preview and pre-ordered CMD hard drives, many of the capabilities we described were not present in the initial release version of the CMD hard drive operating system. During the interim, CMD released a full-featured version of the operating system which included the features we described in issue #27 and more. We apologize to anyone who was confused or misled by our coverage. With the full implementation CMD's operating now in place, the Lt. Kernal and CMD hard drive articles in this issue begin our extended coverage of these hard disk systems. The articles in this issue are designed to give you insight into how these products work. However, due to the extensive coverage of the CMD HD in issue 27, we are limiting our coverage of the CMD product to a simple overview. The next issue of TC128 will examine comparative criteria such as speed, application compatibility, overall functionality and ease of use.

The New, "Old" Approach

As I indicated last issue, the prime problem facing any developer who wishes to create a hard disk system for the Commodore 128 is the lack of support for such a device in the 128's operating system. Creative Micro Design's (CMD) approach to solving this dilemma is quite different from its older rival, Xetec & Fiscal Information's Lt. Kernal Hard drive described elsewhere in this issue. The Lt. Kernal approach involves the integration of a hard disk operating system within the architecture of Commodore 128 itself, aided by external programs (or in CP/M and MSDOS speak transient commands) which are stored on the hard disk itself. In effect, this transforms the 128 from a computer which was totally oblivious to a hard disk into a new, hybrid computer which is hard disk ready. This integrated approach, not coincidentally mimics the manner in which the majority of most hard disk systems are implemented (hardware hooks to a disk based transient operating system) on other computing platforms.

CMD's approach is radically different, and reflects a more traditional approach to disk storage on the 128. The CMD hard drive like all other Commodore disk drives has its operating system based primarily within the disk drive firmware itself. Although portions of the CMD HD's operating system are stored within a hidden partition on the hard disk, this component of the hard drive's operating system like the firmware inside of a typical Commodore disk drive such as the 1571 and 1581, are not meant to be accessed by the typical user. Also like Commodore disk drives, DOS commands are sent via the command channel and interpreted and executed totally within the drive itself.

While contrasting the differences between the Lt. Kernal and CMD hard drive operating system implementations may seem like useless technical trivia, knowledge of this difference is fundamental to comparing these two devices. The prime advantage to CMD's approach is that by operating within a framework that closely resembles Commodore's own framework, a greater degree of compatibility with off the shelf applications is

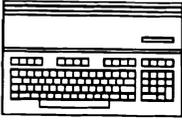
achieved. The penalty CMD's product pays for this compatibility is the loss of some convenience when it comes to user-interaction with the hard disk. Because communication with the drive is achieved via the command channel, the command channel must be opened to send a command to the drive. This is accomplished via the familiar syntax: OPEN 15,UNIT NUMBER,15,"COMMAND":CLOSE 15. The result is a lot of typing, unless you are using Creative Micro Designs' Jiffy DOS (which we highly recommend you use in conjunction with the hard disk for this reason and the fact that it increases the performance considerably) which allows you to bypass the entire OPEN and CLOSE sequence with a single character such as the "@". An added benefit to this approach is that you can send any hard disk command from applications which allow you to issue a Commodore DOS command.

Native vs. Emulation

In addition to its making its operating system implementation mimic Commodore's floppy disk drive, CMD has gone one step further to insure compatibility. CMD has provided a method by which you can divide up your hard disk into special floppy disk sized partitions. These "emulation" partitions are handled by the disk controller firmware in such a way that applications are fooled into thinking that the media they are dealing with is a floppy disk even though it is actually a small section of your hard disk. Emulation partition types options are available for mimicking 1541, 1571, and 1581 disk drives. Using these emulation partitions we have been able to get nearly every piece of C128 mode software to function with the CMD HD, including programs that involve direct access to disk files such as Superbase 128, GEOS, and Big Blue Reader. Additionally using various utilities we have even been able to install heavily copy protected applications so that they would boot directly off the hard disk, including Superbase 128 and GEOS.

However the use of these emulation modes is not the only way to tap the power of the CMD hard drive. Creative Micro Designs has also provided support for a more conventional type of hard disk partition on the CMD, referred to as "native mode" partitions. These native mode partitions can be as large as 16 megabytes in size and are not limited to the 144 and 296 file limits imposed by Commodore DOS on the 1571 and 1581 respectively. In addition to their ability to defeat this file limit, native mode partitions also provide support for an unlimited number of dynamically allocated MSDOS style subdirectories. These subdirectories allow you to create logical divisions within the native mode partition. In general, most applications which use "high level" file access routines rather than low level or direct access routines can utilize the enhanced storage and subdirectory capabilities of native mode partitions. In our tests we have had no problems with applications such as Sidplayer 128, IPaint, and Bible Search in these partitions, and CMD is working to develop strategies for enhancing native mode compatibility with popular applications.

Finally, all CMD hard drives are equipped with a battery-backed real time clock which is used to time and date stamp files. Hard drive commands are provided to access this clock, display custom directories by time etc.



Continued from Page 21

```

60000 BN=15:AD=DEC("1300"):SZ=250:SI=20:LN=9105:LI=5:BN$="MB":AD$="MA"
60002 DATA"%8=%0:%9=%1:BANK%8:IFPEEK(%9)=%4ANDPEEK(%9+%2)=%5THENRETURN"
60004 DATA"RESTORE%6:FORI=%9TO%9+%2STEP%3:READA$:A=0"
60006 DATA"FORJ=0TOLEN(A$)/2-2:B=DEC(MID$(A$,J*2+1,2)):A=A+B:POKEI+J,B:NEXTJ"
60008 DATA"IF(AAND255)=DEC(RIGHT$(A$,2))THENNEXTI:RETURN"
60010 DATA"PRINT%7CHKSUM ERR IN LINE%7:PEEK(65)+256*PEEK(66):END",""
60012 DIMS(6),S$(9),D$(5):S(0)=BN:S(1)=AD:S(2)=SZ-1:S(3)=SI:S$(8)=BN$:S$(9)=AD$
60014 BANKBN:S(4)=PEEK(AD):S(5)=PEEK(AD+SZ-1):S(6)=LN+6*LI
60016 FORI=0TO8:S$(I)=MID$(STR$(S(I)),2):NEXTI:S$(7)=CHR$(34)
60018 RESTORE60002:FORI=0TO5:READA$:DO:A=INSTR(A$, "%?"):IFA<1THENNEXT
60020 A$=LEFT$(A$,A-1)+S$(VAL(MID$(A$,A+1,1)))+MID$(A$,A+2):LOOP:D$(I)=A$:NEXT
60022 IFA<6THENA$=D$(A):A=A+1:GT$="GOTO60022":GOTO60030
60024 A$="DATA ":A=0:GT$="GOTO60024":IFSZ<SITHENSI=SZ
60026 BANKBN:FORI=ADTOAD+SI-1:C=PEEK(I):A=A+C:A$=A$+RIGHT$(HEX$(C),2):NEXT
60028 AD=AD+SI:SZ=SZ-SI:A$=A$+RIGHT$(HEX$(A),2)
60030 PRINTCHR$(147):LN:A$:IFSZ>0THENLN=LN+LI:PRINTGT$
60032 BANK0:POKE842,19:POKE843,13:POKE844,13:POKE208,3:END

```

We hate getting issues of Twin Cities 128 back in the mail.

If you are moving be sure to send us a change of address notice. Just drop us a letter or postcard with your name as it appears on the mailing label and your old and new address; or request a change of address form from your Post Office.

Issues of Twin Cities 128 are published as often as the small staff of dedicated Commodore enthusiasts can get them done. Honest, we wish we could become Commodore Week or even the Daily 128, but it just isn't possible. For this reason we don't purport to be a monthly either, basing our subscriptions on a set number of issues instead of a period of time. A subscription gets you six issues as they are published, but the interval between issues will continue to vary. We know how anxiously many of you wait for your next issue, but it is partly this policy that has allowed us to continue to offer concentrated C128 coverage while many other Commodore publications have been forced to cease publication.

If you can't wait for more TC-128, we do make available limited supplies of back issues. Back issues are \$2.50 per single copy, but many issues are "sold out" at this time. Write us for our current list of "in stock" back issues. Almost all of our first 18 numbers are gone, but you can get them repackaged and bound into our Twin Cities 128 Compendium Book #1 for \$16.95. If you have any questions about your subscription or ordering any TC-128/Voyager Mindtools product, you can write us at our address.

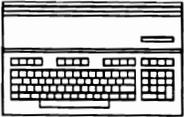
All articles published in this issue are copyrighted, and may not be reprinted without express written permission from the Commissioner of Baseball - oops! - I mean from us. This condition does extend to user group newsletters, so please, write us first.

If you want to write for us, we welcome submissions. The typical TC-128 article is from one to three thousand words long and we expect thorough and understandable coverage of topic within this length. Payment rates can vary, but are normally between \$10-\$25. We retain all rights to the material we accept. TC-128 strongly prefers that text comes on disk and in Commodore PETSCII format with no formatting codes included. If you have illustrations or diagrams, make them black and white (monochrome) in geoPaint or BASIC 8 format. Include your address and phone number for response. You will be notified if your article has been accepted, but we regret that we cannot return submissions.

If you are ordering a subscription or other TC-128/Voyager Mindtools product for delivery to a foreign address, be aware that we have to charge a bit more to cover the postage. In return for the higher rate, we send our Canadian issues under first class mail and our overseas subscribers get their products via air mail. Current Canadian subscription rates are \$15 for six issues and equivalent overseas rate is \$25 for six.

In conclusion, thanks for waiting for Twin Cities 128 #28. We hope you enjoy the issue and continue to learn about using your Commodore 128. It is the continued support of reader/users such as yourselves that makes the task of producing this magazine and our other products worthwhile!

Our Address: Twin Cities 128 P.O. Box 11578 Minneapolis MN 55411



INPUT#, GET#, RECORD# and so forth are transparently implemented. So, in run-mode the Lt. Kernal seems to be just a great big 1541 with one heck of a speedup kit installed.

In immediate mode, the Lt. Kernal takes on a whole new character. No longer are you required to utilize Commodore-style commands to execute housekeeping functions on the system (although you may if you like to work with a stone ax). Instead, the Lt. Kernal DOS provides a full complement of its own DOS commands that make such things as moving, copying and renaming files easier. For example, in this new environment, programs are loaded and executed simply by typing the filename.

Other commands allow you to verify the integrity of your files or verify the integrity of DOS itself. (Like most high performance computer systems, the Lt. Kernal DOS is disk-resident and is loaded into RAM at boot-up time.) Still other functions allow you to sort and print directories, output a file to screen or printer and perform a variety of tricks on a BASIC program currently in memory. And, if your C-128 is running in C-64 mode you can return it to 128 mode with the command GO128.

All immediate mode commands implemented by the Lt. Kernal DOS are executed by what are referred to as DOS processors. Each processor is a contiguous block binary file stored in a drive partition assigned to the DOS. Upon pressing the <RETURN> key, the RAM-resident portion of the DOS, always executing in the host adaptor, examines the typed command, starting at the left margin of the screen. If the command doesn't start at the left margin it is simply passed to the BASIC interpreter.

Otherwise, the DOS searches its partition for a processor of the same name as the first part of the command. If no such processor is found, the DOS will attempt to load and execute a file of the same name. However, if a DOS processor is found to match the command it is loaded and executed. At this point, two different actions may occur, depending on the command being executed.

Some processors are relatively simple in nature and thus are executed in an area in the host adaptor reserved for "transient" functions. Other processors, however, are too large to fit into the RAM in the host adaptor and thus are executed in the \$C000-\$CFFF range of computer RAM. To avoid destroying the contents of that area, a low-level DOS routine buffers that entire block of memory on the drive and later restores it when the processor has completed its task. This all happens so fast that the user is never really aware of what is going on.

Let's suppose, however, that the DOS cannot find anything in its partition that matches the command that has been typed. The next step is for the DOS to search the currently-logged "logical unit" or LU. An LU is a software-defined partition on the drive, and roughly corresponds to the physically separate drives that are found in such units as the Commodore 8050 or the MSD SD-2. However, unlike the 0 or 1 choices available in a dual floppy drive, the Lt. Kernal provides 0 through 10 (LU 10 belongs to the DOS but may be used to store user files). LUs 0-9 may be defined and configured by the user as required (the limit on the size of any one LU is 32 megabytes).

If the DOS is not able to find a match for the command on the currently-logged LU and subdirectory (defined as a "user" [USER]

area), it then passes the command to the BASIC interpreter (the DOS command syntax often allows specification of LU and USER, thus allowing one to cross partition boundaries). The result is that the DOS acts as an integrated part of the computer's operating system, and no files have to be opened to the command channel in order to tell the drive what to do.

The beauty of this type of user interface is that it allows the use of English language commands instead of the usual terse and non-mnemonic Commodore DOS channel commands. It also allows the addition of new functions as desired. In fact, if the user is capable of doing some simple programming they can create new commands on the Lt. Kernal to perform specific tasks or to automate existing ones.

A networked computer system considers each connection to an external device (such as a terminal or printer) a "port." So it is with the Lt. Kernal. Each computer on a multiplexed (MUXed) system is assigned a port number (0-15), which is determined by hardware in the host adaptor. The port number is displayed as part of the Lt. Kernal system prompt, along with some information that tells the operator which LU and USER area in which they are working and the device number of the drive (the device number can be changed with a typed command - no switches need be thrown or pressed).

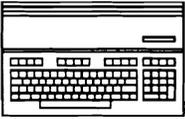
The operation of a multiplexed system is contingent upon each computer being uniquely identified, which is why host adaptors are equipped to provide port numbers. Port 0 is considered the "master" port and all others are considered "slaves." The master must be present on the system or else the SCSI bus will crash. Slaves do not have to be on-line, however, for everything to work.

The MUX itself time-shares the SCSI bus, giving each port its turn at accessing the drive, without any risk of a bus collision. This happens so fast that users on a multiplexed system are seldom aware of any other activity. If a port requires access to the drive, the MUX will service that request and then move on. No additional software or hardware is required to accomplish this, as the multi-port concept is inherent in the design of the Lt. Kernal.

A low-level DOS function (callable from machine language) makes it possible to determine the port number of a particular computer, which facilitates the development of software whose features can be varied depending on which port is being used. For example, you might have one C-128 in the office and another out in the stock room. You could write an inventory program that would allow either location to examine the inventory database, but only allow the computer in the stock room to actually change the parts counts.

I've mentioned that in run-mode the Lt. Kernal appears to be an extremely large and fast 1541. What I haven't talked about are some file types that are unique to the Lt. Kernal. They are the KEY-INDEX file (KEY file for short) and the contiguous data file. Both of these are without counterpart on any other CBM-compatible drive.

A KEY file is the disk-based equivalent of a RAM-based variable array (subscripted variables). Describing a typical programming task will explain what is so valuable about this sort of file. Let's



Function ROMs: The Mystery Revealed

by: Kent Sullivan

Introduction

The Commodore 128's architecture, in terms of the mixture of ROM and RAM that it can manage, is quite flexible and powerful. Numerous articles have been published on the features and use of banked memory for programming, but little exists on the subject of function ROMs.

The C-128 can have up to 64K of "extra" ROM plugged in, with up to 32K available at a time. The ROM can be divided into 16K or 32K chunks as well. Even nicer is the way in which such ROM can be added to the system: you don't have to build or buy an external cartridge to add an EPROM like you do with the C-64-- you can simply plug a chip into an empty slot on the machine's motherboard. Finally, function ROMs are attractive because the programs they hold are immediately available you don't have to load anything from disk.

Despite the power and benefits of function ROMs, very few ROM products have been produced. Of the few products that have appeared, the most notable was perhaps BASIC-8. However, it is no longer available on ROM.

Internal vs. External

As described above, the C-128 can manage up to 64K of function ROM. The 64K is divided into "internal" and "external" ROM areas, 32K each. The internal ROM plugs into the previously-mentioned empty socket inside the C-128 (labeled "U36" on the motherboard), while the external ROM plugs into a cartridge in the expansion port (similar to C-64 cartridges). Since everyone who has a C-128 has the internal socket, it is the most attractive option for program development.

However, a little-known fact is that many people also have a "half-finished" external ROM socket: on their RAM Expansion Units (REUs)¹. Commodore included all the necessary connections on the REU circuit board for an external function ROM socket. You need only solder a 28-pin low-profile socket onto the board and you're in business. According to Fred Bowen, Senior Systems Engineer at CBM, the empty spot on the board was to hold RAMDOS, but this was never implemented.

Selecting the ROM Window

Although you can have a full 64K of function ROM plugged into the system, only 32K of it can be used at a time. A look at a memory map for the C-128 (like on page 469 of Abacus's "C-128 Internals"² explains why. Internal and external function ROMs "compete" for the same 32K address space in memory: from \$8000 to \$FFFF. The space is divided into two 16K chunks: a "lo" section (\$8000 - \$BFFF) and a "hi" section (\$C000 - \$FFFF). In addition, much of the system ROM maps into the area beginning at \$8000, which further complicates matters.

You can divide each 32K ROM into two separate 16K ROMs if you wish or use it as one 32K area. The configuration you want can be selected via the Memory Management Unit (MMU). The 16 most popular configurations (banks) can easily be chosen, but you also can create many other possibilities (see Appendix D of

Abacus's "BASIC 7.0 Internals"³ for a complete list of MMU configurations).

Reset, Phoenix, and Logging in ROMs

To understand how to put programs on function ROMs, we first need to understand the C-128 system reset routine. The reset routine begins at \$E000 (and can also be called at \$FF3D) and does many important functions crucial to starting (or restarting) the machine properly. We are only interested in the part of the routine that begins at \$E26B. The routine checks each of the four possible 16K ROM areas for a specific pattern (uppercase "CBM"). The order it checks the areas is:

1. external lo
2. external hi
3. internal lo
4. internal hi

For each area that it checks, if it finds the pattern, it then "logs in" the ID byte of the ROM in the Physical Address Table (PAT) in low memory (begins at \$0AC1). All function ROMs are programmed with an ID byte (\$01 - \$FF). This byte is supposed to be unique for each different ROM, but as far as I know manufacturers have not coordinated on the IDs for ROMs that have been produced.

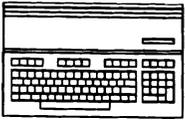
For all possible ID values except \$01, the routine simply moves on to check the next area then continues doing other things. However, if it finds a ROM with an ID of \$01, it immediately "drops" everything and jumps to the "cold-start" address on the ROM. In other words, an ID of \$01 is special and allows a programmer to get control of the machine before BASIC has been started or any other function ROMs have been executed.

For all other ID values, though, the reset routine continues. Next, BASIC calls the Kernel routine Phoenix during its own reset routine. This routine, located at \$F867 (and accessed through the Kernel jump table at \$FF56), keeps track of function ROMs and autoboot disks. Phoenix checks each space in the PAT (using the same order as the Kernel does for checking the actual ROMs), and if it is non-zero (indicating the Kernel found a ROM there), jumps to the ROM's cold-start address.

If you had the maximum number of ROMs (4) logged in, Phoenix could execute each one in turn. Realistically, though, it is quite probable that one of the ROMs would conflict with something that another ROM had done. So, if you want to make sure that your ROM gets executed first, it needs to be in the "external lo" slot (short of having an ID of \$01).

Constructing a Function ROM

With all the above possibilities in mind, let's now discuss how to arrange the program on the function ROM. I won't explain how to use an EPROM programmer in this article because of space limitations. However, I will say that by far the best programmer is the "Promenade", made by Jason-Ranheim⁴. Version 2.0 of their programming software, "Promos", really makes EPROM programming easy with just a little practice.



A Small Memory Expansion for the 128

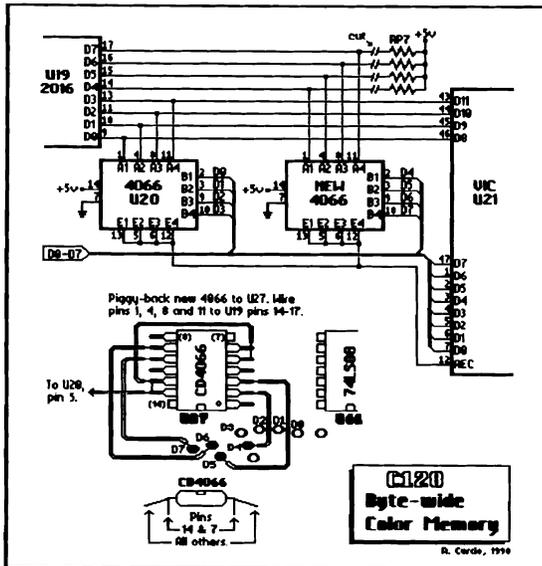
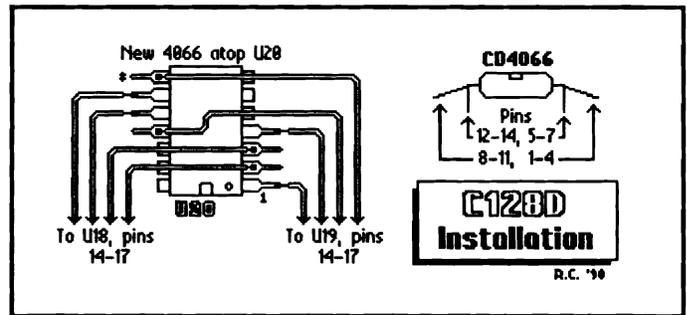
by: Richard Curcio

It is possible to increase the C128 memory internally by the addition of a single chip. Now that I have your attention, I hasten to explain that the increase is very minor, 1 or 2K. With a little imagination, this small amount of memory should prove quite useful.

Details

Both the 64 and 128 use memory 4 bits wide to hold 40 column color information. In the case of the C128, two 1K blocks are provided, one for text, and another for multi-color bit-mapped displays. In that machine, a 2K-by-8 bit static RAM was used, with the upper 4 bits unused and terminated by pull-up resistors. All we have to do to make the upper nybbles available is un-terminate those bits and tie them to D4-D7 of the data bus to acquire byte-wide color memory. This will not provide 256 colors. The VIC chip can display only sixteen colors, which is why color memory is normally only 4 bits wide. One thousand nybbles are needed for the display, and since 1K is actually 1024, the last 24 nybbles of text color RAM are unused by the system. The screen is cleared by printing blank lines, so those 24 nybbles aren't touched. By expanding color memory to full bytes, some interesting color effects can be created with shift and rotate instructions and, in an (internally) expanded memory system, the last 24 bytes can hold short routines and serve as a kind of quasi "common memory." I call it quasi because the I/O block must be visible for this memory to be utilized. In an internally expanded C64, this I/O-resident common memory should ease the bank switching overhead when there is no "normal" common memory.

D8-D11. When AEC is logic 0, the VIC has control of the address and data busses and the quad analog switch U20 is "open", disconnecting the RAM from the system data bus. When AEC is high, logic 1, the CPU has control of the busses and the analog switches are "closed", allowing the CPU to write or read color memory. To make the upper nybbles available, we disconnect the pull-up resistors and wire a second analog switch between the static RAM D4-D7 and the data bus D4-D7.



C128 disassembly is as follows: Begin by disconnecting all peripherals and the power supply from your computer. Remove six screws from the bottom of the case, noting the different sizes. Partially lift the top cover and remove the three wire LED power indicator on the left side. (When you reconnect it, the polarity won't matter.) At this point, you might want to discharge any static charges from yourself by touching a cold water or radiator pipe. Unscrew the keyboard grounding strap on the right side of the case and carefully remove the keyboard connector. Do this by gently rocking it from side to side while maintaining an upward pull. This connector is "keyed" and will go back on only one way. If any of the pins bend, you can gently bend them back to the vertical position. If you break any pins, you will lose use of your keyboard. Remove the screws holding the RF shield in place, again noting any difference in sizes. The metal shield is soldered together at one point. This must be unsoldered, using a solder vacuum or solder braid. There are also small metal tabs holding the top and bottom metal covers together. Gently un-bend these and expose your computer's innards. The white gooey stuff on the larger chips and the underside of the top metal cover is heat-sink compound. Be careful not to get any on clothing, carpets, etc. The C128 circuit board is fairly large. Avoid flexing it.

The IC designations U19 and U20 are the same for the C128 and the C128D. The 2K-by-8 color RAM may be a 2016, 3517 or a 2028. The new analog switch can be a CD4066, MC14066 or a 74HC4066. In a 128D it can be "piggy-backed" to U20, the original 4066, after bending out pins 1-4, and 8-11. In the 128 this is an inconvenient location, right next to the metal box housing the video circuitry. Nearby is U27, which should be a UA556. This provides more room to spread out the pins and the feed-throughs holding the data lines are close by, as shown in the drawing. On a C128D, data bits D4-D7 can be obtained at pins 14-17 of the character ROM, U18.

C128 Disassembly and Modification

The large illustration shows the C128 modification in both schematic and mechanical representation. (Mechanically, 128D installation is a little simpler. I thank Terry Cripe of Defiance, Ohio, for the loan of his 128D service manual.) Bits 0-3 of the static RAM U19 are always connected to the VIC's color data bus,