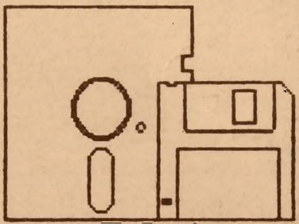
TWIN CITIES 128

THE COMMODORE 128 JOURNAL

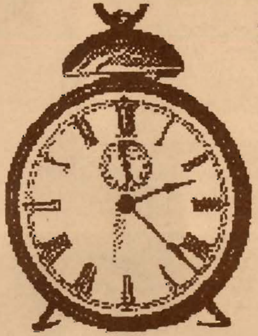
PROUDLY PRODUCED COMPLETELY ON A C-128 IN NATIVE MODE !!!!

ISSUE 16 $ 2.50

Our Second Bi-Annual
C-128 Product Compendium



The Most Comprehensive Listing of C-128 Products
Featuring Over 200 Commercial Titles ! ! !

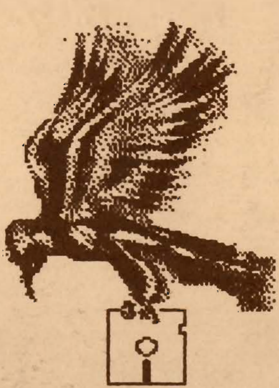


Time for C 128
Graphics Tools

Who says the C-128 is not a powerful graphics machine?

PLUS

A Special Double Installment of Sparrow's Slick Tips



Special Rumor/Opinion/Mayhem
NEWS CONFERENCE

C-128 Price & Progress
$ Report

Modify CP/M
for the 1581!

Using BASIC 7.0's
USR(X) Function

The Assembly Line
Rolls Onward

PLUS:
Find out how to create Spectacular Sixteen Color Bitmaps on your C-128's 40 column screen

Twin Cities 128 is North America's only C-128 specific publication

Twin Cities 128: The Commodore 128 Journal
North America's only C-128 specific publication

Issue #16
Copyright July 1987

T A B L E O F C O N T E N T S

Front Cover..Page 1 by Loren Lovhaug
Credits, Contents, & Stuff...Page 2 by Avonelle Lovhaug
RUMOR/OPINION/MAYHEM...Page 3 by Loren Lovhaug
C-128 Price & Progress Report......................................Page 4 by Loren Lovhaug
Time C-128 Graphics: Who says the C-128 is not a graphics machine?......Page 5 by Loren Lovhaug
1581 CP/M Modification: Use CP/M with your 1581....................Page 7 by Miklos Garamszeghy
C-128 PRODUCT COMPENDIUM #2...Page 10 by Staff
Using BASIC 7.0's USR Function.....................................Page 16 by Miklos Garamszeghy
The Assembly Line..Page 19 by John Kress
Sparrow's Slick Tips...Page 22 by Loren Lovhaug
Advanced VIC manipulation..Page 24 by Miklos Garamszeghy

S T A F F

Loren Lovhaug...........Managing Editor
C-128 Industry News & Developments, BASIC 7.0 & COMAL programming, Productivity Applications, Graphics & Cover design, Business & Correspondence

Avonelle Lovhaug.........Associate Editor
Productivity Applications, Proofreading and Layout, Business & Correspondence, Educational Applications

Bill Nicholson...........Assistant Editor
Productivity Applications, Proofreading and Layout, Music Applications, Business & Correspondence, Graphics, Entertainment software

Randy Margolis...........Staff Writer
Productivity Applications, Proofreading and Layout, Music Applications CP/M Applications

Miklos Garamszeghy.......Staff Writer
BASIC 7.0 and Assembly Language Programming, Productivity Applications, CP/M Applications

John Kress...............Staff Writer
BASIC 7.0 and Assembly Language Programming, Graphics programming

Twin Cities 128 is published 12 times a year and is available for $2.50 per single issue or $25.00 for 12 issues. To order Twin Cities 128 write:

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

Sorry, we cannot accept credit card or C.O.D orders.

If you would like to submit material to Twin Cities 128 for publication, send your article to us on a 1541, 1571, 1581, or CP/M formatted disk with an accompanying hard copy to Twin Cities 128 at the address listed above, or send your submission electronically via the GENIE telecommunications network or our BBS at (612) 699-1387 300/1200 baud.

Products used in the production of this issue of Twin Cities 128:

Commodore 128 Personal Computer	Tandy Model 100 portable computer	Pocket Writer 2 by Digital Solutions
Commodore 1571 Disk Drive	Star Micronics SG-10 Dot Matrix Printer	Pocket Filer 2 by Digital Solutions
Commodore 1581 Disk Drive	Xetec Super Graphix Sr. Interface	Pocket Planner 2 by Digital Solutions
Commodore 1750 512K RAM Expansion	Comrex CR 5400 Monochrome Monitor	Superbase 128 by Precision Software
Commodore 1902A RGBI Monitor	Amdek Color 1 Color Composite Monitor	Bobsterm Pro 128 by RML labs
Commodore 1351 Mouse	GEOS 128 by Berkeley Softworks	Cassell's Latin Dictionary
Commodore DPS-1101 Printer	Writer's Workshop 128 by Berkeley Softworks	
Commodore 1670 automodem	The Reality Construction Set by LJL Inc.	

Special thanks to Fred Bowen of Commodore Engineering for being all-around great guy and to the spouses and loved ones of all Twin Cities 128 staff members for putting up with this nonsense.

You may contact Mr. Lovhaug electronically at the following addresses:
GENIE: Sparrow.J   Q-Link: LOVHAUG   UNINET: gryphon!pnet02!sparrow   TC-128 Online: Sparrow James

computo ergo sum
reductio ad absurdum, deus ex machina

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

Rumor/Opinion/Mayhem

by: Loren J. Lovhaug

During the last few weeks we have witnessed the downfall of Gary Hart and Jim Bakker. As I watched the scrutiny and destruction of these individuals I mused the following: Being the managing editor of a grass roots, 8 bit computer publication is kind of like running for president or being a televangelist. This is because I, like the aforementioned targets of the fourth estate, have been placed on the defensive by an angry mob, waiting with baited breath to devour me. True, my mob is somewhat tamer than the Miami Hearld, The Charolotte Observer, and the Washington Press Corps, but in their own way they are just as relentless in their pursuit. I am talking of course about angry Commodore computerists.

So ladies and gentlemen, I will quit ducking those tough questions and answer my critics. Here is the transcript of a mock press conference I had with myself just the other day.

LJL: Good afternoon ladies and gentlemen. I am glad you could all make it. Looks like you folks are going to get the scoop on this one since CNN, NBC, CBS, and ABC did not bother to show up. Ok, let's get this thing under way, Ms. Ross, you have a question?

PAULINE ROSS (Amigo News Service): Yes, Mr. Lovhaug, we have heard you are a critic of Commodore's new Amiga 500 computer: my question is, given the A-500's $650 price tag, and a performance ratio equal to that of the Amiga 1000, how can you justify the existence of the C-128?

LJL: Well Pauline, let me answer your question by first stating that I am impressed with the Amiga series computers and there is no doubt that they are the most technologically advanced microcomputer on the market today. But, let's take a look at things the way they really are.

Mr. Lovhaug walks over to a C-128 hooked up to a projection video system, and reveals:

Amiga 500 computer w/ disk drive	650	C-128 computer	250
		1571 drive	225
Amiga 1080 monitor	300	1902 monitor	280
Scribble	100	Pocket Writer 2	52
Maxiplan Plus	130	Multiplan 128	30
Superbase Personal	100	Superbase 128	70
Deluxe Paint II	120	GEOS 128	70
512K system plus software	1400	128k system plus software	977
A-501 512K expander w/clock	150	1750 512K REU	150
		1351 mouse	45
1 MB system plus software	1550	640K system plus software	1172

(Note: The above prices are fairly conservative; by shopping around and making the right software substitutions one can do better on both systems. But in general, Amiga productivity titles run 20-40% more than their C-128 counterparts.)

Now if you examine this chart closely you will see that in reality the basic Amiga 500 system is not $650, but instead is $1400 when you add the cost of an RGBA monitor and four productivity titles (word processing, spreadsheet, database, and graphics). The equivalent C-128 system, costs roughly $420 less. It should also be noted that many Amiga 1000 owners have found it necessary to expand their systems beyond the 512K of random access memory provided in the Amiga 500, while the vast majority of C-128 owners get a long just fine within the confines of their machine's standard 128K. However, if you expand the C-128 by adding 512K of RAM and the 1351 mouse, you greatly enhance the C-128's speed and abilities at about 25% off the the equivalently expanded Amiga 500 system. Now granted, the Amiga system is faster, and more powerful, but I submit that for the needs of the average computer user, especially those on a budget, the C-128 system provides an affordable solution.

JANE MEANHAND (TBBS NEWS): In light of the introduction of the new Amiga products, will you consider adding Amiga coverage to Twin Cities 128?

LJL: No. I believe the single most important factor in the success of Twin Cities 128 has been the fact that we cater exclusively to the C-128. Our readers look to us for information they simply cannot find elsewhere, and they appreciate the fact that our publication is not diluted with coverage of other machines. Besides, there are already plenty of publications for the Amigas.

BELLI HANLEYSKI (Moscow Computer News): In your publication, I fail to notice anything but positive commentary about the C-128 and Commodore. Don't you feel you are doing your readers a disservice by continuing your pro-128 propaganda without balancing it with some objective criticism of the C-128 and Commodore?

LJL: Belli, that is an interesting assertion, one that I will not entirely deny. Yes, it is true that in a typical issue of TC-128 you will not find much complaining about the C-128. But I submit that there is not a great deal about the C-128 worth griping about, and that our pages are much better spent discussing ways to help our readers make the most out of their computer investment. However, you will find that I have been critical of Commodore for various marketing policies and the lack of any tangible user support.

EL KURTO SWANSO (El Commode): Mr. Lovhaug, over the past few months you have reported on items like the 1581 disk drive, and GEOS 128 which have not until just recently been available to the ordinary user. Don't you think most 128 owners would rather not read about things they can't have?

LJL: So under your reasoning: If it is not yet available, then people are not interested in it? I think people are very interested in new products for the C-128, and I am proud of the fact that Twin Cities 128 is often months ahead of the so-called "leading publications". By reporting on items like GEOS 128, BASIC 8, and the 1581 disk drive well in advance of their availability, I think we help our readers make informed decisions about future computer purchases. Also, I always state in any preview article that it is just that, a preview of an item that is yet to be released.

C-128 Price & Progress UPDATE

The C-128 Price and Product Report is a new feature of Twin Cities 128 which is designed to keep you updated on the status, availability, current pricing, and upgrade information about various C-128 products.

Current Pricing of Popular Commodore C-128 System Components
(in U.S. Dollars)

	High	Low	Average	Used	Description/Availability/Comments
C-128	$299.95	$229.95	$249.95	$175.00	Classic "flat" C-128
C-128D	$550.00	$499.95	N/A	N/A	C-128 with built-in 1571 and separate keyboard
1541(c) disk drive	$219.95	$159.95	$175.00	$100.00	Used price refers to old 1541
1571 disk drive	$285.95	$215.95	$242.00	$175.00	Be sure to check alignment of used drives
1581 disk drive	$299.00	$209.95	$259.95	N/A	808k super-fast 3.5 inch disk drive
1700 128k expansion	$179.95	$89.95	$107.00	$85.00	Baby RAM expansion unit can be upgraded
1750 512k expansion	$259.95	$149.95	$155.00	$125.00	Cheaper than upgrading new 1700s in most cases
1660 300 baud modem	$49.95	$29.95	$32.00	$20.00	Older 1660s had some design flaws
1670 1200 baud modem	$169.95	$89.95	$117.00	$85.00	The 1670 is excellent bargain
1350 joystick mouse	$49.95	$22.00	$31.00	$15.00	Old mouse acts exactly like a joystick
1351 analog mouse	$69.95	$42.00	$49.00	$40.00	New mouse is more precise, has 1350 mode
MPS 1000 printer	$299.95	$229.95	$237.00	$200.00	Works best in IBM mode, but has 1525 emulation
MPS 1200 printer	$299.95	$259.00	N/A	N/A	Upgraded MPS 1000 with more features
1902A RGBI monitor	$299.95	$239.95	$275.00	$210.00	"The C-128 monitor" has NTSC mode
2002 RGBI/RGBA monitor	$359.95	$299.95	N/A	N/A	New super monitor, supports everything

Notes:

1. The high, low, and average prices were obtained from a random sampling of advertisements in Commodore oriented publications and by telephone conversations with various dealers and distributors in the U.S.

2. As of this writing, the 1581 disk drives had just arrived at retailers. The C-128D had not begun shipping however.

3. The 2002 is Commodore's new "everything" monitor. This monitor supports separated color composite, digital RGB (RGBI), and analog RGB (RGBA). This monitor is idea for the computerist who may consider changing or upgrading computer systems in the future because it has a mode that supports just about every computer (Commodore and non-Commodore alike) on market today.

C-128 Progress Update

Berkeley Softworks, 2150 Shattuck Ave., Berkeley CA 94704, 415-644-0883, finally began shipping GEOS 128 to distributors during late July. The suggested retail price is $69.95. This version of GEOS supports both the 1571 as well as the new 1581 disk drive, the 1350 and 1351 mice, and the 1700/1750 RAM expansion units, and works on either the 40 or 80 column screen. Registered owners of the C-64 version of GEOS will receive a card in the mail which will entitle them to order the C-128 version of GEOS as an upgrade for $19.95 plus shipping and handling.

Berkeley is also in the final stages of development of the first GEOS 128 specific application package entitled Writer's Workshop 128. The package includes four programs: GEOwrite 128 version 2.1, GEOmerge 128, GEOlaser 128, and Text Grabber 128. GEOwrite 128 version 2.1 is a greatly enhanced version of GEOwrite which supports many features that people with serious word processing needs have come to expect but are not found in the version of GEOwrite that comes with GEOS 128. GEOmerge 128 is a very sophisticated programmable external mailmerge program for use with any version of GEOwrite 128. GEOlaser 128 is a program for outputing GEOS created text and graphics to an Apple Laserwriter for near typeset quality documents. And finally Text Grabber 128 is a valuable program which converts text created with other C-128 word processors to GEOwrite format so that those files can take advantage of GEOS' font and graphics ability. The final price was not available as of this writing.

Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380, 1-215-431-9100, began shipping the long awaited 1581 disk drive to distributors and retailers in late June. The suggested retail price for the 808K, 3.5 inch unit is $299.95, although several retailers are advertising them for significantly less than that (see above). The 1581 comes packaged with a test/utilities disk and an order form for a 3.5 inch version of CP/M 3.0. The new 3.5 inch CP/M 3.0 allows the entire formatted capacity of the 1581 to be used for CP/M program and data storages, and is available for $19.95.

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

C128 Graphics

by: Loren Lovhaug

Your C-128 can do more than most people think

In the dark times, before the age of affordable computing, I spent the majority of my time in hot, poorly lit gymnasiums throwing an orange spheriod at a ring elevated ten feet above the floor. Those were the days when my dreams had very little to do with microprocessors, random access memory, pixel resolution, or money. But even though my life has drastically changed since then, at times it seems as though I am replaying contests of old in a different, but parallel setting.

Like the 6'2 forward who's competitive basketball career ended in high school, the C-128 will never make it to the "big time" of computer graphics. Neither of them possess the size, speed, and muscle it takes to attain the highest level of achievement. But all is not lost just because you will never find my name on an NBA roster or the C-128 in the credits for a feature film, rock video, or episode of Miami Vice. After all, I still get a charge out of driving past an opponent in pickup games, and recent C-128 software developments allow you to achieve some equally satisfying and at times stunning results graphically with your C-128. In this article I will discuss three 80 column graphics products which can give you the kind of performance you thought you could get only from 16 bit micros.

When we discuss computer graphic programs we should be aware that there are really two kinds of computer graphics software. The first division encompasses those programs which are designed to create images that are to be displayed or animated on some kind of video display. The other division involves programs which are used to prepare graphics for output to a printed page.

Pagemaking

Over the last two years the hottest buzz words in the personal computing industry has been "desktop publishing". Desktop publishing infers the use of personal computing hardware and software to produce "publication quality" text and graphics. The judgement as to what exactly is publication quality, and what is not, in reality depends upon your audience and the amount of time and money you can afford to expend. The undisputed king of the desktop publishing realm has been and still remains the Apple Macintosh, although the rest of microcomputerdom is catching up quickly. The Macintosh's ability to quickly and easily integrate attractive character sets of various sizes and styles with high quality computer illustrations is now being emulated via software on a variety of computer systems including our own C-128s.

Perhaps the best known of these software "Mac-a-likes" in the Commodore world is GEOS from Berkeley Softworks. As regular readers of these pages know, GEOS was originally marketed for the C-64, but now, Berkeley is now marketing a greatly enhanced version of GEOS for the C-128. We have been fortunate to be a part of the pre-testing process for this product and we have come to rely on it for a great deal of our graphics page make-up needs. Although the C-64 version of GEOS brought us many of the user interfaces and powerful text/graphics integration, it could not deliver the speed and flexibility of the Macintosh due to the memory, speed, and display limitations of the C-64. The C-128 version comes a lot closer to the "near-mac" goal. By utilizing the C-128's 80 column display, the user can easily manipulate text and graphics which span the entire width of the document, whereas on the C-64 version of GEOS one was forced to scroll over the width of a page in rather small horizontal increments. While this difference might appear trivial to the casual user, power users (such as myself) immediately realize the increased pagemaking speed this potentially adds, especially when you use photo (graphics) scraps in conjunction with the photo manager.

Besides the added screen width of GEOS 128, other aspects of its operation are making the Macintosh comparision a valid one. By using a 1750 RAM expansion unit in conjunction with a 1571 or 1581 disk drive you will find that the sluggish nature of GEOS on the C-64 is just a bad memory. In fact, as an illustration of just how much like a Macintosh our little C-128's are becoming, you might take note of the pocketwatch and sparrow illustrations on the front cover of this issue of Twin Cities 128. Both are high quality graphic "scraps" converted from Macpaint files for the Macintosh using a little public domain utility you can download from Genie or Quantum Link. There are also many public domain and commercial conversion utilities for transforming graphics from Commodore 64 programs such as The Print Shop, Printmaster, The Newsroom, Doodle and Koala paint. Through the utilization of conversion programs with GEOS 128 one can have an immense amount of high quality pre-drawn graphics at their disposal.

Although I am quite pleased with GEOS 128, it is too bad that Berkeley chose to package the "standard" version of GEOwrite (albeit with the addition of 80 column screen support) instead of the enhanced version of GEOwrite that will be packaged with Writer's Workshop 128. The enhanced version of GEOwrite (version 2.1) in the Writer's Workshop 128 package truly rounds out GEOS 128 as a pagemaking system, because it supports better margin and spacing control, allows for much more accurate positioning of GEOpaint graphics within your text as well as much greater text editing capabilities. In addition, also included with Writer's Workshop 128 is Text Grabber 128, which makes the importation of text created with other word processors a snap, as well as GEOlaser 128 which allows the output of your text and graphics to an Apple laserwriter should you be fortunate enough to beg, borrow, or steal the use of one.

Another fabulous pagemaking package is Fontmaster 128 by Xetec. This program like GEOS 128, allows the use of multiple fonts as well as graphics integration. Fontmaster's approach to pagemaking is from a word processing standpoint, and as such,

C128 Graphics Continued

incorporates all the usual features of a good word processing program. But unlike most word processors Fontmaster 128 allows the user to integrate Print Shop, Printmaster, and Doodle graphics with text. The graphics integration features of Fontmaster are so strong that they allow text to be formatted around the graphic image, set apart from the image, or even the text to be overlaid on top of the image. Fontmaster also supports multiple column output and comes with a font editor so you can design your own fonts. This font editing ability can also be used to create custom graphic logos or symbols for easy integration into your text. Also, like GEOS 128 with Writer's Workshop you can convert text created with other word processing programs to Fontmaster 128 format.

Video graphics

The other side of computer graphics is screen graphics, or as it is referred to in the television industry, video graphics. Video graphics are by nature more flexible than page graphics, since paper is a static media while video displays are dynamic (i.e. they are constantly being changed or refreshed). However, dealing with video graphics is also more complex. When creating video graphics one might be concerned with items such as color, color resolution, screen scrolling and animation, and appearance on various monitors and situations.

Since its introduction in mid-1985, Commodore's own Amiga computer has been the undisputed champion of microcomputer video graphics. The reason for the Amiga's dominance in this field has been twofold. First, the Amiga's hardware is made up of several custom VSLI controllers which facilitate very high resolution graphics, very high color resolution, and fluid animation. The other reason lies in the fact that several third party software producers have decided to exploit the Amiga's hardware potential by developing incredibly sophisticated video graphics applications. But as powerful and sophisticated as the Amiga's video graphics abilities are, its little brother in the Commodore line, our own C-128 is now making great strides in the Amiga's direction. This is not to suggest that the C-128 has any chance of eclipsing the Amiga's supremacy in the video graphics field, but recent C-128 hardware and especially software developments in this area will certainly have the potential to shock many who don't believe our eight-bits can "cut it".

The software development which has revolutionized C-128 video graphics is BASIC 8 by PATECH software, an incredible three dimensional graphics programming language which pushes the C-128's graphics abilities far beyond what most people believed was possible. The hardware development is the increasing acceptance of the (until recently esoteric) addition of 80 column video RAM to your C-128, and the advent of the C-128D which comes with a standard complement of 64K VDC RAM. This procedure involves the replacement of the C-128's 16k VDC RAMs with 64K RAMs and is detailed complete with pictures in issue #9 of Twin Cities 128. I strongly recommend that anyone who is interested in graphics on the C-128 have this modification done to their machine(s).

As I wrote in a preview of this product in issue #15 of Twin Cities 128, it is very difficult to describe in words what can be done with BASIC 8 without the benefit of a C-128 and RGBI color monitor at your side. Perhaps the two most Amiga-like features of BASIC 8 are its color resolution and virtual screen abilities with 64K of VDC RAM. With the extra video RAM installed, BASIC 8 can create virtual screens with a horizontal size of up to 2040 pixels and a vertical size of up to 819 pixels.

Using the virtual screen facility and the incredibly fluid 255 speed scroll command you can create spectacular animation effects. These effects range from standard cartoon-like page flipping to precise movement over very large bitmaps. Also aiding BASIC 8's animation abilities is a very sophisticated and fairly fast buffering system provided for the movement, manipulation and duplication of screen data and the ability to rotate screen data around a user defined origin in all three dimensions.

And of course all the above can be done in full color. BASIC 8 supports a variety of different color resolutions to allow the C-128 graphics artist an immense amount of flexibility. Under BASIC 8 you can mix foreground and background colors in the following color matrices: 8 x 16 pixels, 8 x 8 pixels, 8 x 4 pixels, and 8 x 2 pixels. In addition, through a sophisticated technique called dithering which involves the mixing of specific colors with various background data you can achieve 128 different "shades" of color. Frankly, I cannot describe how stunning the resulting combination of BASIC 8's array of colors used with the right artistic insight in the 8 x 2 color mode, you will just have to experience it yourself.

Conclusions?

So after reading this two page diatribe what thoughts would I like to leave with you? Simply this: The graphics potential of the C-128 is just now being fully understood and exploited. In many respects these new found abilities can compare favorably with the results found on much larger, and much more expensive (especially when you consider the price of both hardware and software) computers. Personally I gain a certain satisfaction from the exploitation of the C-128 in a domain where it is most certainly an underdog. In the near future, it is my belief that we are going to see even better applications of the aforementioned hardware and software, both unilaterally, as well as together as talented programmers devise conversion utilities for the transfer of pictures, fonts, and fill patterns between various software formats. And rest assured, Twin Cities 128 will remain at the forefront of these developments, helping to nurture them into reality and letting you know about them first. Anyway, I'm off to practice my jumpshot and...maybe its not too late after all.

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

Modify CP/M for 1581 Usage

by: Miklos Garamszeghy

Older versions of C-128 CP/M will not fully support the new 1581 disk drive. This is a pity because the large capacity of the 1581 combined with its high speed make it an ideal CP/M drive. Of course, you can mail in the coupon which comes with the 1581 along with some of your hard earned cash and get yet another "upgrade" version of CP/M which does support the 1581. I already have three different versions of CP/M for my C-128 (the original plus two "upgrades") so why do I need another one just to use the 1581? The short answer is that I do not.

The first problem involves creating a 1581 CP/M format disk. This is easiest to do with the C-128 in native mode using the burst mode format command since older versions of CP/M's FORMAT.COM utility do not work properly with the 1581. The following command string can be used, assuming your 1581 is device 9:

```
open15,9,15
print#15,"u0"+chr$(134)+chr$(2)+chr$(79)+chr$(10)+
chr$(0)+chr$(229)+chr$(1)
close 15
```

This will format the disk identical to a 1581 DOS disk physical structure (i.e. 512 byte/sector, 10 sectors track) but without the directory, BAM, etc. and fill it with CP/M blank disk bytes ($e5 or dec 229).

Next, with a few simple modifications to the CPM+.SYS file, you can take full advantage of the capacity of the 1581 without forking out for the latest upgrade disk. You can still use the disk formatted by the method above with CP/M if you do not make the following mods, but you will only be able to fill it half full. This mod will not allow you to boot CP/M from the 1581, but it will allow you use the 1581 once CP/M has been booted from a 1571 or 1541. (In my present set up, the 1571 is device 8 and the 1581 is device 9. CP/M can only be booted from device 8 anyway). A note in Commodore's favor is that the CP/M upgrade will allow you to boot from the 1581 if it is connected as device 8.

The procedure involves changing a few bytes in part of the CPM+.SYS file known as the "disk parameter block table". This table, which is described in detail in the "CP/M 3 System Guide" under section 3.3 BIOS Data Structures on pages 40 to 44, contains the data for the physical characteristics of the MFM disk formats supported by C-128 CP/M. The location of the table in the CPM+.SYS file depends on the version of CP/M that you are using. The instructions for all three current versions are outlined below. The locations are summarized in Table 1 and referenced in the text.

Before continuing, you will need a formatted CP/M disk (C-128 1541 single or 1571 double sided format) containing the CP/M system files (CPM+.SYS and CCP.COM) and the utility SID.COM (or the older DDT.COM or equivalent debugger utility). You can also include a copy of SHOW.COM to check the results afterwards. The SID.COM program resides on the CP/M additional utilities disk that comes with the Digital Research CP/M plus documentation. Several other public domain debuggers are also available if you do not have SID.

NOTE: USE A BACK UP WORK DISK. DO NOT DO THIS WITH YOUR ORIGINAL SYSTEM DISK BECAUSE IT WILL MAKE PERMANENT CHANGES TO THE OPERATING SYSTEM.

After booting up CP/M, note the version date printed on the screen. This will be used as a reference for the addresses in Table 1. Insert your work disk and type in:

```
sid cpm+.sys <return>
```

where <return> is the key marked "return" at the right of the keyboard. Note that for clarity, I will use lowercase letters to indicate items that you type in and UPPERCASE for prompts made by the computer.

After a few moments, the following status message will be displayed:

```
CP/M 3 SID - Version x.x
NEXT MSZE  PC  END
zzzz zzzz 0100 CEFF
#
```

where zzzz is a hexadecimal number based on the version date, as listed in Table 1. Jot this down for future reference as it will be needed when saving the changes. The "#" symbol is SID's command prompt.

Now we are ready to make the changes. The physical format of a 1581 disk is identical to that of the EPSON QX-10 (10 sectors per track, 512 bytes per sector, 2 sides). The only difference is in the number of tracks per side (80 for the 1581 compared to 40 for the EPSON). It makes things easier to start with the EPSON parameters and change them a bit rather than create a whole new entry. (The modifications will still permit full read compatibility with the EPSON disks and nearly full write compatibility on the 1571. The incompatibility in writing is only that the operating system may attempt to write more to the EPSON disk than it can hold, thus causing a disk error if you try to write to an almost full disk).

Next type in:

```
dyyyy <return>
```

where yyyy is taken from Table 1. This is SID's d or display memory command. Note that in this, and all other SID commands, there is no space between the command letter and the command parameters. For example, d yyyy would produce an error prompt.

SID will respond with a display similar to:

```
yyyy: 50 00 04 0F 01 BD 00 7F 00 C0 00 20 00 02 00
02 P.......... ....
yymm: 03 0A 45 70 73 6F 6E 20 51 58 31 30 49 A5 00
81 ..Epson QX10I...
```

plus some more rows of similar hexadecimal numbers followed by the "#" prompt. The next step is to

CP/M for the 1581 mod Continued

change some of the bytes. The ones that are changed are referred to in Digital Research's documentation as EXM (extent mask - 1 byte), and DSM (total drive storage - 2 byte word). These are the two parameters which tell the system how much data it can store on a disk. The change is done with SID's s or set memory command. Type in:

```
sxxxx <return>
```

where xxxx is taken from Table 1. SID will respond with:

```
xxxx 01
```

The 01 is the current value of the byte at this location. Change it to the desired value by typing in 00 followed by return. SID will then prompt for the next byte:

```
xxx1 BD
```

Type in 86 followed by return. SID will then prompt for the next byte:

```
xxx2 00
```

Change this by typing in a 01 followed by return. That completes the major changes. At the next prompt

```
xxx3 7F
```

Enter a period "." followed by return. This should bring back the main SID prompt "#". If you want to change the disk type name from Epson QX10 to say 1581, use the s command again:

```
snnnn
```

When SID prompts with:

```
nnnn 45
```

Type in a quote (") followed by 3 spaces then 1581 and 3 more spaces and a return. At the next prompt, type in a period to return to the main SID prompt. Type in: dyyyy <return> again to check your changes and the following should be displayed:

```
yyyy: 50 00 04 0F 00 8b 01 7F 00 C0 00 20 00 02 00
02 P.......... ....
yymm: 03 0A 20 20 20 31 35 38 31 20 20 20 49 A5 00
81 ..  1581   I...
```

The final step is to save the changes. This is done with SID's w or write command:

```
wcpm+.sys 0100 zzzz <return>
```

Once this has been done, your new CP/M system is ready for action. Re-boot the computer (you need to boot the modified CP/M system) and turn on your 1581 as drive b: (device 9). You can now PIP some files to a formatted 1581 disk (as outlined above) and use it at will for all file storage. You can check the capacity of the drive by using the SHOW.COM utility:

```
show b:
```

will display the amout of space currently available on the 1581. If it does not give something like 778k read write space free with a formatted blank disk, then your changes may have gone awry. Double check the changes outlined above and try again.

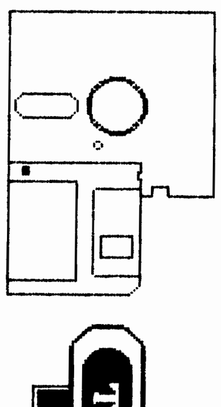
The procedure outlined above can also be done using the RAM disk (drive m:) as the working drive if you have a 1750 RAM expander. In this case, you would PIP the required files to the RAM disk, do the modifications and PIP them back again to your work disk. The same method can be used to alter the disk parameter table to support other CP/M disk formats on the 1571, such as Televideo, Xerox, DEC, etc which are not normally supported by C-128 CP/M. In this case, I would suggest that you consult the explanation of the table parameters in the CP/M System Guide. You also need to have a thorough understanding of the disk format that you wish to implement.

Table 1: Summary of CPM+.SYS file addresses

	Values by CP/M Version Date		
Parameter*	1 Aug 85	6 Dec 85	8 Dec 85
zzzz	5d00	6400	6400
yyyy	1400	2161	2161
xxxx	1404	2165	2165
nnnn	1412	2173	2173

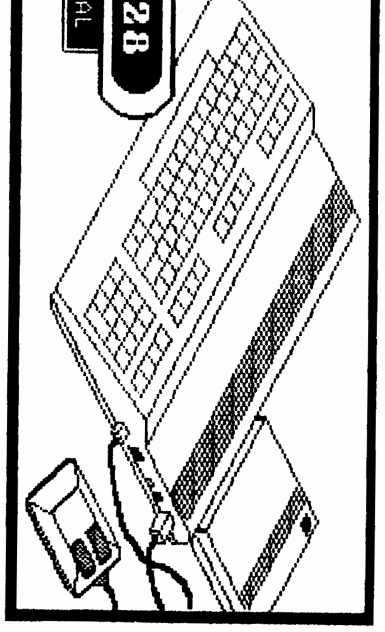
Note: Refer to text for explanation of parameters

The Second Bi-Annual C-128 Product Compendium



TWIN CITIES 128

THE COMMODORE 128 JOURNAL



This page and the following six pages comprise what we feel is the most complete listing of commercial products for the Commodore 128 personal computer (in native mode) you will find anywhere. Although our staff has made a painstaking effort to make sure this listing is accurate, it is possible that errors have occurred. If you find an error, or discover an omission, please bring it to our attention by contacting us via the means listed on the inside front cover.

For your convenience we have included comments and support codes on many of the software products. The comments are those of our managing editor and should be taken "cum grannis salis". Or as he puts it, "what good is running a magazine, if you can't shoot your mouth off every once in a while?"

Software support codes: 41 - supports 1541, 71 - supports 1571, 81 - supports 1581, xx - 1581 support is suspected but not tested, REU - support RAM expansion unit, 40 - supports forty column display, 80 - supports 80 column display, MSE - supports the 1351 mouse, JOY - supports a joystick or the 1350 mouse, PTR - supports most types of printers, DOT - supports only dot matrix printers, MOD - supports a modem, RGB - requires an RGBI monitor. Note: Items in parenthesis are required.

C-128 HARDWARE AND SOFTWARE PRODUCERS
(LISTED SIDEWAYS)

Abacus Software
2201 Kalamazoo St. SE
Grand Rapids MI 49510
616-241-5510

Access Software
2561 S. 1560 West
Woods Cross UT 84087
801-298-9077

Activision
2350 Bayshore Frontage Road
Mountain View CA 94043
201-838-9027

Adam Enterprises, Inc.
P.O. Box 1792
Springdale AR 72765

B.E.S.T.
Box 852
Mc Minnville OR 97128
503-472-9512

Basement Boys Software
P.O. Box 30901
Portland OR 97230
503-781-1114

Batteries Included
30 Mural Street
Richmond Hill ON L4B 1B5
416-881-9941

Berkeley Softworks
2150 Shattuck Avenue
Berkeley CA 94704
415-644-0883

Bouncing Dog Software
P.O. Box 6753
Minneapolis MN 55406
612-729-7682

Briley Software
Box 2913
Livermore CA 94550
415-455-9139

Brivall
P.O. Box 129
Kutztown PA 19530
215-683-5433

Byteware
906 West 6th Avenue
Monmouth IL 61462
309-734-7096

California Computing Spec.
Dept. A 5102 Neptune
Newport Beach CA 92663
714-548-4144

Cappco
P.O. Box 7652
Chula Vista CA 92021
619-477-5970

Cardinal Software
14840 Build America Drive
Woodbridge VA 22191
703-491-6494

Central Point
9700 SW Capitol Hwy. #100
Portland OR 97219
503-224-5782

Clockwork Computers
4612 Holly Ridge Road
Rockville MD 20853
301-924-5509

CMS Software Systems Inc.
2204 Camp David
Mesquite TX 75149

COMAL Users Group USA Limited
6041 Monona Drive
Madison WI 53716
608-222-4432

Commodore
1200 Wilson Drive
West Chester PA 19380
415-431-9100

Compumed
P.O. Box 6939
Salinas CA 93912
408-758-2436

Computer Enterprises
P.O. Box 171206
Arlington TX 76017

Cosmi
415 N Figueroa St
Wilmington CA 90744
213-835-9687

Country Road
70284 C.R. 143
Ligonier IN 46767
219-894-7278

CW Data-Labs
1632 Mapfle Ave.
Philadelphia PA 19111
800-537-LABS

Data Foundations
2206 Meloy Rd.
Kent OH 44240

Digital Solutions
P.O. Box 345 STN. A
Willowdale ON M2N 5S9
416-731-8775

Electronic Arts
1820 Gateway Dr.
San Mateo CA 94404
415-571-7171

Emerald Components Intl.
Dept. R P.O. Box 1441
Eugene OR 97440
800-356-5178

Epyx
1043 Kiel Ct.
Sunnyvale CA 94089
408-745-0700

Family Software
3164 Surrey Lane
Aston PA 19014
215-497-5561

Firebird
74 N. Central Avenue
Ramsey NJ 07446
201-934-7373

Free Spirit Software Inc.
538 S. Edgewood
La Grange IL 60525

Genealogy Software
Dept. MS, P.O. Box 1151
Port Huron MI 48061
519-542-4424

HES
150 N. Hill Drive
Brisbane CA 94005

ICT
P.O. Box 863
Middletown MD 21769
301-371-4000

Infocom
125 Cambridge Park Drive
Cambridge MA 02140
617-492-1031

Jameco Electronics
1355 Shoreway Road
Belmont CA 94002
415-592-8097

KFS Software, Inc.
1301 Seminole Blvd. #153A
Largo FL 33540
813-584-2355

King Microware
Suite 210 5950 CotedesNeiges
Montreal Canada PQ H3S 1Z6
514-737-9335

Kobetek Systems
1007 Commercial Street
New Minas NS B4N-3C4
902-678-1541

Kracker Jax
P.O. Box 6216
Vancouver WA 98668
206-696-4956

Kyan Software
1850 Union Street #183
San Francisco CA 94123
415-626-2080

Michaelsoft
4821 Harvest Ct.
Colorado Spring CO 80917
303-596-4243

Micro Aided Designs
215 E. Orangethrope Ave. #345
Fullerton CA 92632
714-680-5179

Micro R&D
3333 South Wadsworth A104
Lakewood CO 80227
303-985-1473

Micro-W
1342-B Route 23
Butler NJ 07405
201-838-9027

Micro-W Distributing, Inc.
1342B Route 23
Butler NJ 07405
201-838-9027

Microsphere
521 Plymouth St.
Greensburg PA 15601
412-838-1166

Mid-Kansas Computers
204 W. 6th P.O. Box 506
Newton KS 67114
316-283-0208

Midwest Software
Box 214
Farmington MI 48024
313-477-0897

Nth Digit Solutions
3243 Arlington Avenue #195
Riverside CA 92506

Nu-Age
P.O. Box 1042
Indian Rocks Be FL 33535
813-323-8389

Patech Software
P.O. Box 5208
Somerset NJ 08873
201-545-1571

PMG Software
22371B 119 Avenue
Maple Ridge BC V2X 2Z2
Canada

Prism Software
401 Lake Air Dr. Suite O
Waco TX 76710
817-751-0200

Pro-Line
755 the Queensway E. Unit 9
Mississauga ON L4Y-4C5
416-273-6350

Professional Software
51 Fremont Street
Nedham MA 02194
617-444-5224

Progressive Peripherals
464 Kalamath St.
Denver CO 80204
303-825-4144

Quantum Leap
4214 Arden Way
San Diego CA 92103
619-297-7078

Quantum Software
P.O. Box 12716
Lake Park FL 33403
305-840-0249

R & R Mimsa
9500 S.W. 51 Terr.
Miami FL 33165

Roger Wagner Publishing
10761 Woodside Ave Suite
Santee CA 92071
619-562-3221

S.O.G.W.A.P. Software
611 Boccaccio Avenue
Venice CA 90291

ScanAm
10 Garber Square
Ridgewood NJ 07450
201-445-5280

Schnedler Systems
1501 N. Ivanhoe Street
Arlington VA 22205
703-237-4796

Skyles Electric Works
231 E S Whisman Road
Mountain View CA 94041
415-965-1735

SoftSync
162 Madison Avenue
New York NY 10016
212-685-2080

Solid State Software
1125 E. Hillsdale Blvd. #
Foster City CA 94404
415-341-5606

Sonus Corporation
21430 Strathern #H
Canoga Park CA 91304
818-702-0992

Spinnaker
One Kendall St.
Cambridge MA 02139
617-494-1200

Taxaid Software
606 Second Avenue SE
Two Harbors MN 55616
218-834-5012

Timeworks Inc.
444 Lake Cook Rd.
Deerfield IL 60015
312-948-9200

Visionary Software
25882 Orchard Lake Rd Sui
Farmington Hill MI 48018
313-443-0414

Wayne Levine
535 17th St. SW
Owatonna MN 55060
507-451-5720

WOODSoftware
P.O. Box 16193
Wichita KS 67216
316-529-1861

Xetec, Inc.
2804 Arnold Road
Salinas KS 67401
913-827-0685

DATABASE SOFTWARE

Title: Data Manager 128
Company: Timeworks Inc.
Price: 59.95
Support/Comments:
Be sure you get the latest upgrade

Title: Data Master 128
Company: Bouncing Dog Software
Price: 29.95
Support/Comments:
41-71-xx-(80)-PTR

Title: Dfile 128
Company: Michaelsoft
Price: 24.95
Support/Comments:
41-71-81-40-80-PTR

Title: Family Tree
Company: Genealogy Software
Price: 49.95
Support/Comments:
41-71-xx-80-PRT
Genealogy database

Title: FGS
Company: Byteware
Price: 21.95
Support/Comments:
Genealogy database

Title: Filer's Choice
Company: Activision Inc.
Price: 39.95

Title: Flex File 128
Company: Cardinal Software
Price: 49.95

Title: Flexfile 128
Company: Mid-Kansas Computers
Price: 49.95

Title: LeagueBowl-40
Company: Briley Software
Price: 210.00

Title: PED C
Company: Byteware
Price: 21.95

Title: Pocket Filer 128
Company: Digital Solutions
Price: 39.95
Support/Comments:
41-71-81-40-80-PTR

Title: Pocket Filer 2
Company: Digital Solutions
Price: 59.95
Support/Comments:
41-71-81-REU-40-80-MSE-JOY-PTR
Very fast with RAM expansion unit
Impressive math language and time
and date support

Title: Profile 128
Company: Pro-Line
Price: 69.95

Title: Programmer's Notebook
Company: Free Spirit Software Inc.
Price: 19.95

Title: Record Master 128
Company: WOODSoftware
Price: 49.95

Title: Superbase 128
Company: Progressive Peripherals
Price: 89.95
Support/Comments:
41-71-40-80-PTR
The finest database software
currently available for the C-128.
Fully programable, fast, but does
not support RAM Expansions or the
1581 in current version.

Title: The Consultant
Company: Batteries Included
Price: 59.95

EDUCATION

Title: BASICally Simple 128
Company: Free Spirit Software Inc.
Price: 19.95

Title: CRIMP
Company: Wayne Levine
Price: 35.00

Title: Geography World Edition
Company: CA Computing Specialists
Price: 39.95
Support/Comments:
41-71-(40)-(JOY)

Title: I am the C128
Company: Activision
Price: 34.95

Title: McGuffy's Grader
Company: Midwest Software
Price: 49.50
Support/Comments:
Nice grading program for teachers

Title: Mr. Quizzer
Company: Free Spirit Software Inc.
Price: 19.95

BUSINESS & FINANCE

Title: Accountant Inc.
Company: SoftSync
Price: 99.95

Title: B.E.S.T. Accounting C128
Company: B.E.S.T.
Price: 89.95

Title: Cash In - Cash Out
Company: Microsphere Corp.
Price: 69.95

Title: CCI Bottom Liner Serial/IEEE
Company: Clockwork Computers
Price: 134.95

Title: CCI Integrated Merchandiser
Company: Clockwork Computers
Price: 499.00

Title: CCI Merchandiser & Accounting
Company: Clockwork Computers
Price: 400.00

Title: CCI Mortgage
Company: Clockwork Computers
Price: 199.95

Title: CCI Patient Tracking
Company: Clockwork Computers
Price: 199.95
Support/Comments:
Requires two 1571 drives

Title: CCI Property Rental
Company: Clockwork Computers
Price: 134.95

Title: Checkbook 128
Company: Nu-Age
Price: 19.95

Title: CMS Accounting System
Company: CMS software
Price: 179.95
Support/Comments:
Regarded by many as the finest
dedicated accounting packages
for the C-128

Title: Finance and Statistics
Company: Cardinal Software
Price: 19.95

Title: Future Tax
Company: Taxaid
Price: 19.95

Title: Money Master
Company: PRG software
Price: 29.90

Title: Personal Accountant
Company: SoftSync
Price: 34.95

C-128 Products Continued

Title: Personal Portfolio Manager
Company: Abacus Software
Price: 39.96

Title: Small Business Bookkeeping
Company: Adam Enterprises, Inc.
Price: 24.95

Title: SuperClerk
Company: Kobetek Systems
Price: 59.95

Title: Swiftax 128
Company: Timeworks Inc.
Price: 59.95

Title: Sylvia Porter's Investment
Company: Timeworks Inc.
Price: 69.95
Support/Comments:
41-71-xx-REU-PTR

Title: Sylvia Porter's
Personal Finance
Company: Timeworks Inc.
Price: 69.95

Title: TAS 128
Company: Abacus Software
Price: 59.95

Title: Taxware
Company: Skyles Electric Works
Price: 49.95

Title: The Accountant
Company: KFS Software, Inc.
Price: 149.95
Support/Comments:
Highly regarded, more than adequate for most small business purposes

Title: TISAR 128
Company: CW Data-Labs
Price: 49.95

GAMES

Title: A Mind Forever Voyaging
Company: Infocom
Price: 26.95

Title: Battlefront
Company: Electronic Arts
Price: 39.95

Title: Bureaucracy
Company: Infocom
Price: 39.95
Support/Comments:
Fantastic game and concept

Title: Cardinal Game Disk
Company: Cardinal Software
Price: 19.95

Title: Spies Are US
Company: Quantum Leap
Price: 19.00

Title: The Experiment
Company: Quantum Leap
Price: 20.00

Title: The Great War
Company: Free Spirit Software Inc.
Price: 29.95

Title: The Mirror
Company: Compumed
Price: 24.95

Title: The Pawn
Company: Firebird
Price: 9.95

Title: The Scoop
Company: Telarium
Price: 32.95

Title: Trinity
Company: Infocom
Price: 39.95
Support/Comments:
A nice change of pace from the typical adventure game

GRAPHICS

Title: CADPak C128
Company: Abacus Software
Price: 59.95
Support/Comments:
41-71-xx-(40)-MSE-JOY-(DOT)

Title: Chartpak-128
Company: Abacus Software
Price: 39.95

Title: Color-Res
Company: Briwall
Price: 14.95
Support/Comments:
41-71-81-(80)
Graphics conversion program can convert 8K bitmaps to BASIC 8

Title: Geos 128
Company: Berkeley Softworks
Price: 69.95
Support/Comments:
41-71-81-REU-40-80-(MSE/JOY)-DOT
Finally. This is a true winner and a must have for anybody with even a slight interest in page based graphics.

Title: Home Designer 128
Company: Briwall
Price: 45.00
Support/Comments:
A nice design application with powerful features. Don't let the name fool you into thinking this is a toy, it is a powerful design tool.

Title: T.H.I.S.
Company: Micro Aided Designs
Price: 59.95
Support/Comments:
41-71-xx-(REU)-(40)-MSE-

Title: The Ruler 128
Company: Bone Frontier Software
Price: 49.95

Title: Three-D Canvas
Company: Cappco
Price: 49.95

Title: Writer's Workshop 128/GEOS
Company: Berkeley Softworks
Price: 39.95
Support/Comments:
41-71-81-REU-40-80-(MSE/JOY)-DOT
Upgraded version of GEOwrite 128 and text grabber 128 make this a must have for any power GEOS user.

INTEGRATED PACKAGES

Title: Desk Manager
Company: SoftSync
Price: 39.95

Title: HomePak 128
Company: Batteries Included
Price: 49.95

Title: Jane
Company: Commodore
Price: 69.95
Support/Comments:
41-71-(40)-(JOY)-PTR
This package is aweful.

Title: Rhapsody 128
Company: King Microware
Price: 74.95

Title: Trio 128
Company: SoftSync
Price: 69.95
Support/Comments:
Not much better than Jane.

C-128 Products Continued

MISCELLANEOUS SOFTWARE

Title: Deluxe Circuit Analysis C128
Company: Nth Digit Solutions
Price: 59.95

Title: Golf Handicapper
Company: Computer Enterprises
Price: 29.50

Title: Land Surveyor C-128
Company: R & R Hissa
Price: 20.00

Title: Micro Doctor 128
Company: Micro R & D
Price: 30.00

Title: Swimming Pool Chemistry
Company: Free Spirit Software inc.
Price: 19.95

Title: Technical Analysis System
Company: Abacus Software
Price: 59.96

MUSIC SOFTWARE

Title: Enhanced Sidplayer
Company: Compute Books
Price: 24.95
Support/Comments:
41-71-81-(40)-JOY
Excellent music editing system. A must for both casual or serious music interests. Many public domain music files in Enhanced sidplayer format.

Title: MIDI Processor
Company: Sonus
Price: 169.95

Title: Rhythm King 128
Company: Skyles
Price: 89.95

Title: Super Sequencer 64/128
Company: Sonus
Price: 40.00

PROGRAMMING LANGUAGES

Title: BASIC 8
Company: PATECH software
Price: 39.95
Support/Comments:
41-71-81-REU-(80)-MSE-JOY-DOT-(RGB)
One of the most important pieces of C-128 software ever written. This BASIC extension opens graphic potent on the 128 that was previously not even dreamed of.

Title: BASIC 8 On a Chip
Company: PATECH software
Price: 19.95
Support/Comments:
41-71-81-REU-(80)-MSE-JOY-DOT-(RGB)
ROM version of BASIC 8 for the C-128' empty socket. You must buy the disk to order the ROM.

Title: C Power 128
Company: Pro-Line
Price: 89.95
Support/Comments:
I have heard that this is an excellent C implementation, but since I hate C with a passion my opinion does not count for much.

Title: COBOL 128
Company: Abacus Software
Price: 59.95

Title: COMAL 128 version 2.02
Company: COMAL Users Group USA
Price: 185.00
Support/Comments:
41-71-81-40-80-JOY-PTR
COMAL is the finest programming language available for the C-128. This 128 mode cartridge is pricey pricey, but I love it, and if you are programmer you will too.

Title: COMAL Superchip
Company: COMAL Users Group USA
Price: 29.95
Support/Comments:
41-71-81-40-80-JOY-PTR-(COMAL 2.01)

Title: Kyan Pascal 128
Company: Kyan Software
Price: 69.95
Support/Comments:
Nice implementation of Pascal

Title: Oxford Pascal 128
Company: Progressive Peripherals
Price: 49.95
Support/Comments:
Another nice 128 mode Pascal

Title: Super C Language
Company: Abacus Software
Price: 79.95
Support/Comments:
Here we go again another C package..yawn

Title: Super Pascal
Company: Abacus Software
Price: 59.95
Support/Comments:
Yet another decent 128 mode Pascal

Title: VS128COBOL
Company: Visionary Software
Price: 49.95

SPREADSHEETS

Title: Calc Result Advanced
Company: ScanAm
Price: 99.00

Title: Multiplan C128
Company: Epyx
Price: 45.00
Support/Comments:
Excellent full featured spreadsheet

Title: Planner's Choice
Company: Activision Inc.
Price: 39.95

Title: Pocket Planner 128
Company: Digital Solutions
Price: 39.95
Support/Comments:
41-71-81-(80)-PTR

Title: Pocket Planner 2
Company: Digital Solutions
Price: 59.95
Support/Comments:
41-71-81-REU-(80)-MSE-JOY-PTR
Excellent easy to use yet powerful spreadsheet with sophiscated graphics and my personal choice.

Title: Swift Calc 128
Company: Timeworks Inc.
Price: 59.95
Support/Comments:
41-71-xx-(80)

Title: Swiftsheet 128
Company: Cosmi
Price: 24.00

Title: Vizastar 128
Company: Solid State Software
Price: 119.97
Support/Comments:
41-71-xx-(80)-PTR
The king of C-128 spreadsheets. This is essentially Lotus 1-2-3 for the C-128.

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

C-128 Products Continued

TELECOMMUNICATIONS

Title: Bobsterm Pro 128
Company: Progressive Peripherals
Price: 79.95
Support/Comments:
41-71-81-(80)-PTR-(MOD)
Simply the best telecommunications package available.

Title: Cnet 128
Company: Perspective Software
Price: 69.95
Support/Comments:
41-71-81-REU-(80)-PTR

Title: EBBS
Company: Ed Parry
Price: 49.95
Support/Comments:
41-71-81-(80)-PTR

Title: ProtoTerm 128
Company: Briwall
Price: 21.95
Support/Comments:
41-71-81-(80)

Title: Sixth Sense 128
Company: Prism
Price: 49.95
Support/Comments:
41-71-xx-REU-40-80-(MOD)
Great C-128 telecommunications package, too bad Prism got a hold of it.

Title: SpeedTerm 128
Company: Abacus Software
Price: 39.95

UTILITIES

Title: 1541/1571 Drive Alignment
Company: Free Spirit Software Inc.
Price: 34.95

Title: 1571 Clone
Company: Micro-W Distributing, Inc.
Price: 49.95

Title: BASIC Compiler 128
Company: Abacus Software
Price: 59.95
Support/Comments:
The fastest, and most sophistciated BASIC compiler for the C-128.

Title: BASIC Program Cross-Referencer
Company: Data Foundations
Price: 24.95

Title: Blitz 128
Company: Skyles Electric Works
Price: 99.95
Support/Comments:
Good, easy to use BASIC compiler

Title: Buddy 128
Company: Pro-Line
Price: 69.95

Title: C-128 Micro Dr.
Company: Micro R&D
Price: 150.00

Title: Copy II 64/128
Company: Central Point
Price: 39.95

Title: Fasthack'em
Company: Basement Boys Software
Price: 29.95
Support/Comments:
41-71-(40)

Title: Gnome Kit
Company: Briwall
Price: 39.95

Title: Gnome Speed
Company: Briwall
Price: 59.95
Support/Comments:
Another easy to use BASIC 7.0 compiler

Title: Hack Pack 128
Company: Progressive Peripherals
Price: 49.95

Title: Mach 128
Company: Access Software
Price: 49.95
Support/Comments:
41-71-xx-40-80-PTR
Good speedup for 1541 owners..does little for the 1571 however.

Title: Matrix
Company: Progressive Peripherals
Price: 59.95
Support/Comments:
You can get just as good or better in the public domain, save your $$$.

Title: Merlin 128 Assembler
Company: Roger Wagner Publishing
Price: 69.95
Support/Comments:
41-71-81-(80)-PTR
Highly recommended

Title: Partner 128
Company: Timeworks Inc.
Price: 69.95
Support/Comments:
41-71-81-(80)-PTR
Many people swear by this one but abandoned mine long ago.

Title: Peek A Byte 128
Company: Quantum Software
Price: 29.95

Title: Petspeed 128
Company: Progressive Peripherals
Price: 49.95
Support/Comments:
41-71-81-40-80
Nice full featured BASIC compiler

Title: Physical Exam
Company: Cardinal Software
Price: 39.95
Support/Comments:
41-71-PRT

Title: RAMDOS-128
Company: Progressive Peripherals
Price: 34.95
Support/Comments:
RAM disk software for the 1700/1750 RAM expanders...excellent tho not transparent enough for most commeric applications.

Title: Rebel Assembler
Company: Nu-Age
Price: 29.95
Support/Comments:
41-71-81-40-80-PTR

Title: Sideways 128
Company: Timeworks Inc.
Price: 29.95

Title: Source Linker
Company: PRG Software
Price: 29.90

Title: Super Clone Machine
Company: Micro-W
Price: 49.95

Title: Super Disk Librarian
Company: Free Spirit Software Inc.
Price: 29.95

Title: Super Disk Utilities
Company: Free Spirit Software Inc.
Price: 39.95

Title: Symbol Master
Company: Schnedler Systems
Price: 49.95
Support/Comments:
Heard great things about this, but have yet to see it in person.

Title: The 128 Cannon
Company: Kracker Jax
Price: 34.95

C-128 Products Continued

Title: The Big Blue Reader
Company: S.O.G.W.A.P. Software
Price: 29.95
Support/Comments:
(71)-(80)-PTR

Title: The Big Blue Reader + CP/M
Company: S.O.G.W.A.P. Software
Price: 29.95
Support/Comments:
(71)-REU-(80)-PTR

Title: The Development System
Company: Access Software
Price: 79.95

Title: Time-DOS
Company: Family Software
Price: 19.95

Title: XREF-128
Company: Abacus Software
Price: 17.95
Support/Comments:
41-71-xx-
Works, but you can find just as good in the public domain

Title: Zoom! 128
Company: Skyles Electric Works
Price: 29.95

WORD PROCESSING

Title: Braintrust 128
Company: Country Road
Price: 21.95
Support/Comments:
Word Processor & outline processor combo

Title: Fleet System 3
Company: Professional Software
Price: 69.95
Support/Comments:
Great dictionary/thesaurus spell checker but somewhat cumbersome.

Title: Fleet System 4
Company: Professional Software
Price: 79.95
Support/Comments:
Same as Fleet 3 plus database

Title: Fontmaster 128
Company: Xetec, Inc.
Price: 59.95
Support/Comments:
41-71-xx-
Great font and graphics integration

Title: Ghost Writer 128
Company: HES
Price: 39.95
Support/Comments:
41-71-xx-(80)-PTR
Nice basic wordprocessor

Title: Legal-ease II
Company: Quantum Leap
Price: 49.00

Title: Paperclip 128
Company: Batteries Included
Price: 39.95
Support/Comments:
41-71-81-40-80-PTR

Title: Paperclip II
Company: Batteries Included
Price: 79.95
Support/Comments:
41-71-81-REU-40-80-PTR
Extremely full featured post formatted word processor. This one does it all. Has built in telecommunications package.

Title: Pocket Writer 128
Company: Digital Solutions
Price: 39.95
Support/Comments:
41-71-81-(80)-PTR

Title: Pocket Writer 2
Company: Digital Solutions
Price: 59.95
Support/Comments:
41-71-81-REU-(80)-MSE-JOY-PTR
Incredibly powerful and easy to use WYSIWYG word processor. Our staff swears by Pocket Writer 2.

Title: Superscript 128
Company: Progressive Peripherals
Price: 79.95
Support/Comments:
41-71-81-40-80-PTR
Excellent programmable word processor. Very powerful. Can be co-resident with Superbase 128

Title: Term Paper Writer
Company: Activision
Price: 59.95

Title: The Critic 128
Company: Quantum Leap
Price: 29.00

Title: Vizawrite Classic 128
Company: Solid State Software
Price: 89.97
Support/Comments:
Supports proportional spacing, and automatic multiple column layouts.

Title: Word Writer 128
Company: Timeworks Inc.
Price: 69.95

Title: Wordfile 128
Company: Michaelsoft
Price: 24.95

Title: Wordpro 128
Company: Spinnaker
Price: 69.95
Support/Comments:
I know people who have used Wordpro since the PET days, and would not useanything else, but I can't stand its cumbersome command structure.

Title: Wordpro 128/S
Company: Spinnaker
Price: 89.95

Title: Writer's Choice
Company: Activision
Price: 39.95

HARDWARE

Title: Command Center - C-128 Cabinet
Company: Ketek
Price: 149.95
Support/Comments:
Nice but overpriced

Title: E-link - IEEE Interface
Company: Progressive Peripherals
Price: 49.00

Title: Link-2 - C-128 IEEE Interface
Company: Rich Hill
Price: 139.00

Title: Quicksilver 128 - IEEE
Company: Skyles Electric Works
Price: 119.95

Title: Commodore MPS 1000 Printer
Company: Commodore
Price: 229.95

Title: Commodore MPS 1200 Printer
Company: Commodore
Price: 259.95

Title: SFD 1001 1 MB Disk Drive
Company: Commodore
Price: 179.95

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

C-128 Products Continued

Title: Mini-Chief 10 MB Hard Disk
Company: ICT Inc.
Price: 595.00
Support/Comments:
Comes in a 1571 enclosure with a 1571!

Title: Mini-Chief 20 MB Hard Disk
Company: ICT Inc.
Price: 695.00

Title: C-128 Upgrade ROMs
Company: Commodore
Price: 21.95
Support/Comments:
Corrects nagging bugs in the C-128 firmware.

Title: 1571 Upgrade ROMs
Company: Commodore
Price: 6.65
Support/Comments:
The 1571 upgrade ROM is a must purchase, improves drive performance 1000%

Title: JCT 1005 10 MB Hard Disk Drive
Company: JCT
Price: 829.00

Title: Excel-71 Disk Drive
Company: Emerald Components
Price: 209.00

Title: Lt. Kernal 20 MB Hard Disk
Company: Xetec Inc.
Price: 949.95

Title: RS232 Adapter
Company: Jameco Electronics
Price: 39.95

Title: CPS 128 External Power Supply
Company: Jameco Electronics
Price: 59.95

Title: RS232 Interface
Company: Omnitronix, Inc.
Price: 49.95

BOOKS

Title: Commodore 128 Programming Secrets
Company: Osborne McGraw-Hill
Price: 15.95

Title: COMPUTE!'s 128 Programmer's Guide
Company: COMPUTE! Publications Inc.
Price: 16.95

Title: Commodore 128 Assembly Language Programming
Company: Howard W. Sams and Co.
Price: 15.95

Title: 1001 Things To Do With Your Commodore 128
Company: Tab Books Inc.
Price: 12.95

Title: Commodore 128 Programmer's Reference Guide
Company: Bantam Books
Price: 21.95

Title: COMPUTE!'s First Book of Commodore 128
Company: COMPUTE! Publications Inc.
Price: 14.95

Title: 128 Machine Language For Beginnners
Company: COMPUTE! Publications Inc.
Price: 16.95

Title: The Essential Commodore 128 User's Guide
Company: HP Books Inc.
Price: 12.95

Title: Commodore 128 BASIC Programming Techniques
Company: Tab Books Inc.
Price: 12.95

Title: The Commodore 128 Subroutine Library
Company: Bantam Books
Price: 12.95

Title: COMPUTE!'s Kids and The Commodore 128
Company: COMPUTE! Publications Inc.
Price: 14.95

Title: Commodore 128 Tricks and Tips
Company: Abacus Software
Price: 19.95

Title: Commodore 128 Internals
Company: Abacus Software
Price: 19.95

Title: 1571 Internals
Company: Abacus Software
Price: 19.95

Title: C-128 BASIC 7.0 Internals
Company: Abacus Software
Price: 24.95

Title: CP/M On The C-128
Company: Abacus Software
Price: 19.95

Title: Peeks and Pokes
Company: Abacus Software
Price: 16.95

Title: C-128 BASIC Training Guide
Company: Abacus Software
Price: 16.95

Title: 128/64 Computer Aided Design
Company: Abacus Software
Price: 19.95

Title: Superbase: The Book
Company: Precision Books
Price: 15.95

Title: Mapping The Commodore 128
Company: COMPUTE! Publications Inc.
Price: 19.95
Support/Comments:
This is a must for serious programmer

Title: COMPUTE!'s Second Book of the Commodore 128
Company: COMPUTE! Publications Inc.
Price: 14.95

Title: Commodore 128 Datafile Program
Company: Tab Books Inc.
Price: 16.95

Title: Mastering Disk Operations on the Commodore 128
Company: Sybex Inc.
Price: 16.95

Title: 35 Amazing Games For Your C-12
Company: HP Books Inc.
Price: 14.95

Title: The Black Book of C128
Company: Value-Soft
Price: 15.95

Title: The Commodore 128 Mode
Company: Microcomscribe
Price: 14.95

Title: M. L. Routines for the C-128
Company: COMPUTE! Publications Inc.
Price: 12.95

Title: Your Commodore 128
Company: Osborne McGraw-Hill
Price: 14.95
Support/Comments:

Title: M.L. for the 128 & Other C= Computers by Butterfield
Company: Simon and Schuster, Inc.
Price: 16.95

Using USR

By: Miklos Garamszeghy

When mixing BASIC and machine language, most people automatically think of BASIC 7.0's SYS and RREG commands. These are great, as you can pass parameters back and forth, do all of your number crunching in machine code for maximum speed, etc. The only problem is that the parameters passed must be integers and less than 256: that is they must be 8 bit values. What do you do if you want to work with larger numbers? Well you can break them down into low and high bytes and pass the data that way. Fine so far. What happens if you want to work with fractions? How do you express a fraction as a few integers of less than 256? Well it ain't easy!

Fortunately, BASIC also provides a method for passing floating point numbers (i.e. ones that can include fractional parts as well as integer parts) between BASIC and machine language. This is done with the often misunderstood USR(X) function. The function has the general syntax:

```
     A=USR(X)
```

Now lets look at what this actually does. The value represented by the variable X is passed to a machine language routine via the computer's floating point accumulator #1 (called FAC1 for short). Your machine code then takes over and does something to it and returns a value to FAC1. BASIC then assigns the new value in FAC1 to the variable A. X can be either a variable or a constant. USR(X) is perhaps best described as a machine language version of BASIC's DEF FN(X) command which allows you to define your own customized BASIC function.

So far so good. The USR function will act like a SYS to the address pointed at by a vector at $1219-$121a (decimal 4633-4634) on the C-128. The execution of USR actually begins at $1218 which contains the byte $4c, the 8502 opcode for the JMP instruction. Before calling the USR routine, you must POKE the low and high bytes of the address of your machine code into these locations, respectively. The only restriction on machine code location is that it must **start in BANK 14** free RAM, that is, equivalent to BANK 0 below address $4000. The BANK 14 location is noteworthy because it contains the character set ROMs instead of the I/O block. If you wish your USR routines to perform direct I/O even if just to the screen or from the keyboard, you must switch to BANK 15. This is done with a:

```
     LDA #0
     STA $FF00
```

instruction sequence. In this case, you do not need to switch back to BANK 14 before returning to BASIC.

Convenient locations for the machine code include the cassette and RS-232 buffers from $0b00 to $0dff and the free RAM from $1300 to $1bff. Of course, most of your custom code can be located elsewhere if you do the correct BANK switching to access it and jump back to the low RAM area again and switch back to either BANK 14 or 15 before returning to BASIC. You can also access built-in ROM routines if you so desire. In fact, you can do anything with the machine code part that you feel like. You can even completely ignore the value in FAC1 if you wish. In this case, USR merely acts like a SYS to BANK 14 via the $1219-$121a vector, with no passing of parameters. Once the vector at $1219-$121a has been set, it remains active in both program and direct mode until either a hard re-set is performed or it is changed to something else. The vector can be changed as often as you wish within your BASIC program.

The prime advantage of USR is that you can specify floating point numbers to transfer to machine language. Floating point notation is a five byte representation of a number which consists of a 1 byte exponent followed by a 4 byte mantissa. Floating point numbers can contain both a whole and a fractional part, such as "123.456". Both the exponent and mantissa are in binary form. In simple terms, if you could do a literal translation of a floating point number it would look something like the following in binary notation:

```
     1001010.01011
```

The bits to the left of the "decimal point" represent increasing powers of 2: 2, 2, 2, etc.; while those to the right represent decreasing powers of 2: 2Δ-1, 2Δ-2, 2Δ-3 etc. These are equivalent to 1/2, 1/4, 1/8 etc. This produces a rather complex bit pattern in the five bytes which bears no real resemblance to "normal" number. Because binary floating point may not be the most accurate way to represent certain numbers (you only have 31 bits to represent both the whole and fractional part of the number combined), small round off errors are sometimes introduced.

The actual floating point representation of a number stored as a variable is less complex than would appear from the above paragraph if it is broken down logically. Take the following example. The number 12.75 is represented by the five byte string $84 $4c $00 $00 $00. This can be seen more clearly in the following explanation.

1. Take the highest power of 2 that will divide evenly into the number. In this case it is 8 or 2. Add the power (3) to $81 to get the exponent byte of $84. Exponent byte values of $81 or greater represent numbers of 1 or greater, while values of less than $81 represent fractions only (i.e negative powers of 2).

2. The sign of the number is positive (i.e +12.75). Therefore bit 7 of the first mantissa byte is 0. If the number was negative (i.e -12.75), bit 7 would be a 1.

3. The remainder of the whole part of the number is 12 - 8 or 4. In binary, this is represented by the bit pattern 1 0 0. Therefore the next three bits of the first mantissa byte are 1 0 0. Note that the number of bits used here is the same as the power of two used to calculate the exponent byte.

4. The rest of the bits in all of the mantissa bytes represent the fractional part of the number. In this case, .75 = 1/2 + 1/4. This produces the bit pattern 1 1. The remainder of the bits are all zero, since no higher fractional powers of 2 are required.

TWIN CITIES 128
THE COMMODORE 128 JOURNAL

Using USR Continued

See Figure 1 for the final representation of the mantissa bytes.

With USR(X), floating point numbers have a slightly different representation in the FAC's, depending on where they came from (i.e a constant or a variable). The bytes associated with FAC1 are outlined in the following example for the values of +12.75 and -12.75 represented both as a constant and as a variable.

From this table in Figure 2, it is apparent that USR(X) sets several flags in the FAC depending on where the data originated. Specifically, bit 7 of mantissa 1 is always set. The sign bit is transferred to bit 7 of the sign flag. If the sign flag is either 0 or $ff (for positive or negative, respectively), then the number came as a constant. If the value of the sign flag is any other, the number was transferred as a variable from BANK 1, again with bit 7 set for negative numbers.

Fortunately, you need not know too much about floating point notation in order to use USR. BASIC 7.0 contains several routines (listed in Figure 3) which perform conversions to and from various other formats such as integer and ASCII representation. These routines are most useful in USR routines because they allow you to do most work in simpler to use and faster integer format then convert back to floating point at the end.

On the C-128, FAC1 is located at RAM address $62 to $68 (including flags for the sign (+ or -) of the number and the exponent). A second FAC, FAC2 is located at $6a to $6f. The C-128 BASIC ROM contains a number of pre-programmed routines for manipulating the contents of the two FACs. The most important of these are outlined in Table 1. These can be called be your USR routine with a JSR statement. Remember that the addresses are all in BANK 15. In addition to the simple math routines listed, there are many more complex routines. Consult a C-128 memory map to find others if you need them.

Now let's take a look at the simple example in Figure 4.

The machine code in this example will divide any number by 256 and keep the remainder. This is useful for finding low bytes of memory addresses to pass to other machine code routines. Unlike other techniques for finding low bytes (such as X AND 255), this routine is not limited to values of less than 32768. (You can even use fractions or negative numbers if you wish, but the answer might not mean what you expect it to).

The next example in Figure 5 also provides a new BASIC 7.0 function: it will create a number that is the mirror image of the input. That is: 123.4 will become 4.321.

The program works by first converting the floating point number to an ASCII string, then reversing the characters in the string and converting the string back to floating point notation. The routine only works with positive numbers.

These two examples may seem trivial, but the USR(X) function can be used to speed up many of the C-128's slow math functions and also to add many more specialized ones for numerical number crunching type programs. Its chief advantage in these cases is speed over performing similar calculations in BASIC.

Figure 1:

```
0 1 0 0 1 1 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0
Δ Δ     Δ
Δ Δ     Fractional part starts here (1/2, 1/4, 1/8, 1/16 etc)
Δ Δ
Δ Remainder of whole part starts here
Δ
Mantissa sign bit 0 = positive, 1=negative

giving values of $4c 00 00 00.
```

Figure 2:

Address	Descrip.	Constant USR(12.75)	Constant USR(-12.75)	Variable USR(X)	Variable USR(-X)
$62	exponent sign	00	00	04	04
$63	exponent	$84	$84	$84	$84
$64	mantissa 1	$cc	$cc	$cc	$cc
$65	mantissa 2	00	00	00	00
$66	mantissa 3	00	00	00	00
$67	mantissa 4	00	00	00	00
$68	mantissa sign	00	$ff	$4c	$cc

Using USR Continued

Figure 3: Summary of C-128 Floating Point Routines

Address	Routine
Conversions:	
84b4	Convert FAC1 to two byte integer at $66, $67 with most significant byte first (i.e high byte)
84c9	Convert two byte integer to FAC1, LDA high byte and LDY low byte before calling
8c28	Copy FAC2 to FAC1
8c38	Copy FAC1 to FAC2
8c47	Round off FAC1 to integer (still in FP notation)
8cc7	Convert FAC1 to 4 byte integer at $64 to $67, most significant byte first
8d22	Convert ASCII to FAC1, ASCII string of numerical characters terminated by 0 byte. If string is in BANK 0, then set $3d-$3e with address of string and LDX #0 before calling. If string is in BANK 1, then set $24-$25 with address and LDX with value >0. In both cases, LDA with first digit in string before calling.
8e42	Convert FAC1 to ASCII string beginning at $010, ending with a 0 byte. This is required to PRINT the value of a number.
Math Routines:	
8831	FAC1 = FAC2 - FAC1
88dd	FAC1 = FAC2 + FAC1
8a55	FAC1 = FAC2 * FAC1
8b4c	FAC1 = FAC2 / FAC1

Figure 4: USR demo

```
  1 rem usr routine will give remainder of x/256
  2 rem for all integer values of x up to over 9 digits long
  3 :
  4 rem machine code part is:
  5 rem          jsr $8cc7  ; convert facl to 4 byte integer
  6 rem          lda #$00   ; null out high byte
  7 rem          ldy $67    ; keep least significant byte
  8 rem         jsr $84c9  ; convert 2 byte integer to fp
  9 rem         rts        ; go back to basic
 10 :
100 for i=2816 to 2826:read x:poke i,x:next
110 poke 4633,0:poke 4634,11
120 input x
130 print usr(x):goto 120
140 data 32, 199, 140, 169, 0, 164, 103, 32, 201, 132, 96
```

Figure 5: USR Demo 2

```
  1 rem usr routine will reverse digits of positive number
  2 rem eg. 123.45 will become 54.321
  3 :
  5 rem         jsr $8e42    ; convert fp to ascii string in buffer at $0100
 12 rem         ldx #$ff     ; set pointer #1
 13 rem         ldy #$00     ; set pointer #2
 14 rem         lda $0100,y  ; get a character from ascii string
 15 rem         beq $0b13    ; check for end of string 0 byte
 16 rem         sta $0b00,x  ; store string in buffer in reverse order
 17 rem         iny
 18 rem         dex
 19 rem         bne $0b07    ; get next character
 20 rem         inx          ; set to first digit
 22 rem         stx $3d      ; save it for charget
 23 rem         lda #$0b     ; high byte of address pointer
 25 rem         sta $3e      ; save it for charget
 26 rem         ldy #$00     ; reset index pointer
 27 rem         lda ($3d),y  ; get first digit
 28 rem         ldx #$00     ; digit stored in bank 0
 29 rem         stx $0c00    ; mark end of string with 0 byte
 30 rem         clc          ; first character is a number
 31 rem         jmp $8d22    ; jump to convert ascii to fp routine
 32 :
100 FOR I = 2816 TO 2855:READ X:POKE I,X:NEXT
110 POKE 4633,0:POKE 4634,11
120 INPUT X:PRINT USR(X)
200 DATA 32, 66, 142, 162, 255, 160, 0, 185, 0, 1, 240, 7, 157, 0, 11, 200
210 DATA 202, 208, 244, 232, 134, 61, 169, 11, 133, 62, 160, 0, 177, 61, 162, 0
220 DATA 142, 0, 12, 24, 76, 34, 141, 0
```

The Assembly Line

by: John Kress

One of the nice features about the 8563 Video Display Controller (VDC) is the fact that it is totally programmable. This means that with a little knowledge of how the system works, you have an immense amount of control over 80 column displays.

Many different configurations of the VDC display are possible, but in this session we'll concentrate on one mode: adding extra lines to the screen display. In testing, with a bit of monitor fine tuning, I've been able to display more than 30 lines to the 80 column screen. But most users probably won't want to make any adjustments to their monitor, so the following program will allow only 28 lines.

Just adding these lines, though, doesn't make them useable. Putting information on the added lines is not supported by the ROM screen editor, so to utilize these added lines you need to make up your own routines to print information there. (See the basic example that follows the source listing.)

How do you program extra lines to the screen?

Register number 6 of the 8563 chip controls the number of lines displayed. This register normally contains the value 25/$19 which sets up the 25 line display. By changing the value here, you can increase or decrease the number of screen lines displayed. In the example, we'll be increasing this value, and thereby adding another 80 bytes of character information and 80 bytes of attribute information (color memory), for each line added. But adding lines causes a problem. The default screen layout is as follows. The first character displayed on the screen (home position) represents the value stored in VDC memory location 0/$0000. The last character position of the normal screen is 1999/$07CF. The attribute memory, which handles color, flash, underline and case information, begins at 2048/$0800. If we add lines to the active screen, the memory for the added lines will over-run into the attribute memory. But we can use another set of registers to reposition the attribute start address, and free up a larger block of ram for the character information. If we do this we will also have to inform the 128 screen editor of the new location of the attribute memory location, using the starting page pointer at 2607/$0A2F. This pointer holds the high byte value of the beginning address for VDC attributes. As a point of interest, there is also a screen memory page pointer, for the screen memory at 2606/$0A2E.

Explaining the Assembly Listing.

In the following listing, using Merlin 128, I'll further explain some of the routines that are used to produce the new lines on the 80 column screen. This short program will expand the display to 28 lines on the screen. I've found that more than 28 lines will not be visible on most monitors, without altering the vertical size adjustment.

The program begins with line 25, where the attribute start address is relocated to $1000 hex. Also this same information is placed into the attribute page pointer that is used by the screen editor (line 28). Next VDC register six is loaded with the new number of lines that are going to be displayed on the screen (28), followed by a screen clear by printing a Clear/Home character. The next operation may need some adjustment for your particular monitor. Register 7 is a fine tuning of the vertical character sync position. The default value here is 32/$1D,and controls the position of the first displayed line on the screen relative to the border area. If the active display area needs to be moved upwards, increase the value in line 35, or decrease the value to move the screen down.

The following operation, CLRLIN, is a good example of the VDC chips ability to fill a block of memory with a common value. To accomplish a block fill a series of preparations needs to be completed. The following is a brief outline of these steps.

1) Bit 7 of the Vertical Smooth Scroll register controls whether the next block operation will be a block fill or copy. Setting this bit to %0 indicates a block fill, and %1 indicates a copy. By using this method a block of memory from 2 bytes up to 255 bytes can be either copied or filled.

2) The next step is to load registers 18-19/$12-$13 with the destination of the fill. In this example the area to be filled is located at 2000/$07D0, or the first memory location of the 26th line on the screen.

3) Then the data register, 31/$1F, is loaded with the data to place into the block of memory, starting at the address chosen above.

4) The final step is to load the Word Count register, 30/$1E, with the number of bytes to be filled, minus one. Because of the way that the count is handled, the VDC chip will always cycle through the word count value one additional cycle than the value programmed here. So, if you wish to fill 128 bytes, you'd program 127 to the word count register.

The routine labeled CLRATT is basically a carbon copy of the above, adjusted to clear out and set the color memory for the added lines. The value used in line 64 of the assembly listing of $82, can be changed to suit your color preference. One point that should be noted: the upper nibble, or higher four bits of the value contain information about the type of character displayed. The chart below indicates what the attribute byte means in a bit by bit representation.

ALT	RVS	UNDL	FLASH	R	G	B	I
7	6	5	4	3	2	1	0

ALT controls the character set chosen. A %0 here displays uppercase/graphics and a %1 displays the lower and uppercase character set.

RVS designates whether the character displayed will be normal or reversed video display. A %0 bit indicates the normal mode and %1 displays reversed video mode.

Assembly Line Continued

UNDL bit will display the character with underlining. As before, a %0 bit is normal mode and %1 bit indicates underlining on.

FLASH when set to %1 will cause the character displayed to flash, much like the cursor flashing, a %0 here is the normal mode display.

The lower nibble or lesser four bits contain the color assignments for the character, R being red, G being green, B is blue and I equals the intensity bit. A set bit here turns on the color assignment for the corresponding byte. Intensity when set to %1 will make the lighter shades of the color designated, such as Blue and Light Blue. The following is a chart of the color and their RGBI nibbles.

Color	R	G	B	I	HEX
BLACK	0	0	0	0	0
DK GRAY	0	0	0	1	1
BLUE	0	0	1	0	2
LT BLUE	0	0	1	1	3
GREEN	0	1	0	0	4
LT GREEN	0	1	0	1	5
CYAN	0	1	1	0	6
LT CYAN	0	1	1	1	7
RED	1	0	0	0	8
LT RED	1	0	0	1	9
PURPLE	1	0	1	0	A
LT PURP.	1	0	1	1	B
BROWN	1	1	0	0	C
YELLOW	1	1	0	1	D
LT GRAY	1	1	1	0	E
WHITE	1	1	1	1	F

That ends the setting up of the 28 line screen display. Now on to placing some useful information onto the new lines.

I opted to divide the new display into three 80 column display areas (lines), so that information can be placed onto any one of the three lines without over-writing information on one or more lines. In order to place information, such as a string, onto the new lines you must then use the SYS instruction from basic, passing the chosen line in the .A register, such as SYS DEC("B68"),LN. LN needs to be a value from 0 thru 2, 0 being the first added line, 1 the second, and 2 the third. The first operation on line 82 of the assembly listing tests the .A register for a value of 3 or more, and if so, returns to basic. Then the .A register is transferred to the .X register and the address for the beginning of the corresponding line is loaded into the program, from a table of addresses. This is done twice, the first time to blank out or clear the line, and the second time to print out the string. The message will be passed to the ML routine from basic by pokeing the ASCII value for each letter, in the MESG storage area.

What can you do with these added lines?

As most of us have seen, the CP/M side of the C-128 uses this type of display to show the status of the disk drive. One could use a basic subroutine to read the disk status (DS$) and place this information onto the added lines before and after a disk operation. Other things like the filename of the current file in use on the disk, or as I've found handy, a simple form of a scratch pad, when I want to keep track of an important address of a ML routine, for instance. This information will not be cleared when the computer performs a SCREEN CLEAR. Try the sample program that goes along with the object code and do some experimenting on your own.

Enter and save the source and save the ML code with the file name of 'PRINT ROUTINE.O'.
Those of you who don't have the use of an assembler can enter the data listed on the left side of the listing, using the C-128's built-in monitor.

```
1    ********************************
2    *                              *
3    * Merlin 128 Assembly Listing  *
4    ********************************
5    SETREG   EQU   $CDCC
6    READREG  EQU   $CDDA
7    BSOUT    EQU   $FFD2
8    SCRPAG   EQU   $A2E
9    ATTPAG   EQU   $A2F
10   PNT      EQU   $E0       ;pointer pair to screen memory
11   USER     EQU   $E2       ;pointer pair to attrib memory
12   SCBOT    EQU   $E4       ;window bottom
13   SCTOP    EQU   $E5       ;window top
14   SCLF     EQU   $E6       ;window left
15   SCRT     EQU   $E7       ;window right
16   WRITE80  EQU   $CDCA     ;write to memory pointed to
17                            ;by registers 18 - 19
18   READ80   EQU   $CDD8     ;read from 18 - 19
19
20
```

```
                *******************************
                *    START OF PROGRAM         *
                *******************************
                        ORG   $B00
0B00: A2 14     BEG     LDX   #20      ;attribute start address reg
0B02: A9 10             LDA   #$10     ;moved to $1000
0B04: 20 CC CD          JSR   SETREG
0B07: 8D 2F 0A          STA   ATTPAG   ;pointer for attribute page
0B0A: A2 06             LDX   #6       ;lines on screen
0B0C: A9 1C             LDA   #28      ;now 28
0B0E: 20 CC CD          JSR   SETREG
0B11: A9 93             LDA   #$93
0B13: 20 D2 FF          JSR   BSOUT    ;Print clr/home
0B16: A2 07             LDX   #7       ;vertical sync register
0B18: A9 1E             LDA   #$1E     ;set vertical sync position
0B1A: 20 CC CD          JSR   SETREG
0B1D: A2 18     CLRLIN  LDX   #24      ;copy/fill bit in
0B1F: 20 DA CD          JSR   READREG  ;this register
0B22: 29 7F             AND   #$7F     ;clear for a fill
0B24: 20 CC CD          JSR   SETREG   ;operation
0B27: A2 12             LDX   #18      ;set destination
0B29: A9 07             LDA   #$07     ;of the fill
0B2B: 20 CC CD          JSR   SETREG   ;to $07D0
0B2E: E8                INX            ;in registers
0B2F: A9 D0             LDA   #$D0     ;18 - 19
0B31: 20 CC CD          JSR   SETREG
0B34: A2 1F             LDX   #31      ;choose data register
0B36: A9 20             LDA   #$20     ;and fill area with
0B38: 20 CC CD          JSR   SETREG   ; space character
0B3B: A2 1E             LDX   #30      ;choose word count reg.
0B3D: A9 EF             LDA   #239     ; and fill 240 bytes
0B3F: 20 CC CD          JSR   SETREG   ;with the space ($20).
0B42: A2 18     CLRATT  LDX   #24      ;getcopy of copy bit
0B44: 20 DA CD          JSR   READREG  ;register and clear
0B47: 29 7F             AND   #$7F     ;bit 7 for a fill
0B49: 20 CC CD          JSR   SETREG   ;operation.
0B4C: A2 12             LDX   #18      ;set destination
0B4E: A9 17             LDA   #$17     ;of the fill in
0B50: 20 CC CD          JSR   SETREG   ;registers 18-19
0B53: E8                INX
0B54: A9 D0             LDA   #$D0
0B56: 20 CC CD          JSR   SETREG
0B59: A2 1F             LDX   #31
0B5B: A9 82             LDA   #$82     ;attribute char value
0B5D: 20 CC CD          JSR   SETREG   ;$8=lowercase $2=blue
0B60: A2 1E             LDX   #30      ;
0B62: A9 F0             LDA   #240
0B64: 20 CC CD          JSR   SETREG
0B67: 60        NOVALID RTS

                *******************************
                * The main routine to print out
                * a line to the extra lines.
                * use SYS PRINT,n where n is the
                * added line to print to in the
                * range 0 through 2.
                * 0 = the 26th line
                * 1 = the 27th line
                * 2 = the 28th line
                *******************************

0B68: C9 03     PRINT   CMP  #3        ;is the entry line number
0B6A: B0 FB             BCS  NOVALID   ;a valid line? if not RTS.
0B6C: AA                TAX            ;transfer .A to .X for
                                       ;setting the correct line
                                       ;address to work with.
0B6D: BD BF 0B          LDA  HIBYTE,X  ;get the hibyte of line
0B70: 8D 8C 0B          STA  ADDHI+1   ;to write to and store
0B73: 8D A7 0B          STA  HIADD+1   ;for clear and write routines
0B76: BD C2 0B          LDA  LOBYTE,X  ;get lobyte of line to
0B79: 8D 92 0B          STA  ADDLO+1   ;write and store for
0B7C: 8D AD 0B          STA  LOADD+1   ;clear and write routines.
0B7F: A2 18             LDX  #24       ;set-up for a block fill
0B81: 20 DA CD          JSR  READREG   ;to the line that's chosen
0B84: 29 7F             AND  #%01111111;clear bit seven for block fill
0B86: 20 CC CD          JSR  SETREG    ;and store value
0B89: A2 12             LDX  #18       ;register 18= update addr. hi
0B8B: A9 07     ADDHI   LDA  #$07      ;this value will change for the
0B8D: 20 CC CD          JSR  SETREG    ;correct line address.
0B90: E8                INX            ;select register 19
0B91: A9 D0     ADDLO   LDA  #$D0      ;this value also will change
0B93: 20 CC CD          JSR  SETREG    ;to address the right line.
0B96: A2 1F             LDX  #31       ;load the data into register 31
0B98: A9 20             LDA  #$20      ;fill character (space)
0B9A: 20 CC CD          JSR  SETREG
0B9D: A2 1E             LDX  #30       ;word count register #30
0B9F: A9 4F             LDA  #79       ;fill line with 80 spaces
0BA1: 20 CC CD          JSR  SETREG    ;to fill the line
0BA4: A2 12             LDX  #18
0BA6: A9 07     HIADD   LDA  #$07      ;this is the address pointer
0BA8: 20 CC CD          JSR  SETREG    ;that changes within the program
0BAB: E8                INX            ;register 19 update low byte
0BAC: A9 D0     LOADD   LDA  #$D0      ;low byte pointer to line
0BAE: 20 CC CD          JSR  SETREG    ;now printing to the line is set
0BB1: A0 00             LDY  #0        ;pointer to the text now set
0BB3: B9 C5 0B  PMSG    LDA  MESG,Y    ;message gets loaded
0BB6: F0 AF             BEQ  NOVALID   ;found the 0 byte delimiter
0BB8: 20 CA CD          JSR  WRITE80   ;put the data in memory
0BBB: C8                INY            ;increment text pointer
0BBC: D0 F5             BNE  PMSG      ;jump back for more text
0BBE: 60                RTS            ;a safety return for those who
                                       ;forget this
0BBF: 07 08 08
0BC2: D0 20 70
0BC5: 00
```

Try the following basic program to test out this routine.

```
10 IF PEEK(2816)<> 162 THEN BLOAD "PRINT ROUTINE.O": ELSE GOTO 30
20 SYS 2816: REM * SET-UP 28 LINE SCREEN
25 A$=D$: LN=1: GOTO 70: PRINT THE DISK STATUS ON LINE 26
30 INPUT "[CLR/HOME] ENTER A STRING TO PRINT ON ADDED LINES";A$
40 INPUT " LINE TO PUT STRING ON ( 1 - 3 )";LN
50 LN = LN -1:REM * CONVERT TO 0-2 VALUE FOR ML ROUTINE
60 IF LN < 0 OR LN > 2 THEN PRINT "[DOWN] INCORRECT VALUE": GOTO 40
70 AL = LEN(A$)
80 FOR A = 1 TO AL
90 POKE 3012+A, ASC(MID$(A$,A,1)): REM * STORE THE STRING AT MESG AREA
100 NEXT AL
110 POKE 3013 + AL,0: REM * POKE IN THE DELIMITER BYTE
120 SYS 2920,LN : REM * PRINT THE LINE, LINE NUMBER PASSED IN .A REG
130 PRINT " ENTER ANOTHER MESSAGE? ( Y/N )" : GETKEY DA$
140 IF DA$ = "Y" THEN GOTO 30
150 END
```

TWIN CITIES 128
THE COMMODORE 128 JOURNAL
Issue 16

Sparrow's Slick Tips

00111110: DIRECTORY by a shorter name
From Ray Bryan, Saint Paul, MN

A little known fact about BASIC 7.0 is that it actually contains two commands for obtaining a disk directory, the standard DIRECTORY command, as well as the CATALOG command. The CATALOG command functions identically to the DIRECTORY command, but has one often overlooked advantage, namely its keyword abreviation is shorter as demonstrated below:

Command	Abreviation
DIRECTORY	diR (di [shift] r)
CATALOG	cA (c [shift] a)

Keep in mind that both the CATALOG and DIRECTORY command default to unit 8, drive 0, so in order to access any other drive or device you must specify them with the ON extension. (example: cA on u9 retrieves a directory from device 9)

00111111: Creating and using partitions and subdirectories with the 1581
From: Loren Lovhaug, on a bus somewhere in Minneapolis

One of the nicest thing about the 1581, besides its speed and storage capacity, is its ability to create partitions and subdirectories. This ability makes the organization of files on a high capacity drive like the 1581 much more managable. Partition's on the 1581 are created through the use of the command channel. Partitions show up on your directories as "CBM" files, a new filetype implemented in the 1581's upgraded version of Commodore DOS. Here is how you create a partition:

```
Open 15,8,15,"/0:par-name,"+chr$(starting track)+chr$(starting sector)+chr$(low byte of # blocks)+chr$(high
byte of # blocks)+",C":Close 15
```

Partitions are nothing more than protected areas of disk space. Once a partition has been created the specifed area on the disk is protected from BAM allocation changes and validates (COLLECT). One must be very careful not to create a partition that includes the 1581's directory track, track 40. Doing so will make your disk somewhat useless although I am convinced that something along these lines is going to be used as a copy protection device if commercial software vendors ever decide to market software in the 3.5 inch format. In addition, you must be careful if you have already saved some data on your disk that you do not overwrite it with a partition. The SHOW BAM utility on the 1581 test/utilities disk can show you what areas of your disk have not been previously allocated for data storage. Partitions are handy things because it makes it possible for foreign formats and bizarre stuff to "live" on the same disk as normal fomatted data. For instance you can have a disk with both CP/M and native mode data on the same diskette (in addition it is looking quite possible that many really foreign MFM formats..like Atari ST may be able to live and even be read with the 1581.) In addition, provided your partition is created in such a way that it meets the following criteria you can format the partitioned area and thereby create a "sub-directory" (actually a directory of files that lives inside of the main or ROOT directory).

In order to create a sub-directory inside a partition the partition must be created as follows:

1. The partition area must be at least 120 blocks in size.
2. The starting sector must be 0.
3. The ending sector must be a multiple of 40 (and thus the total number of blocks are as well).

If your partition qualifies you can create a subdirectory as follows:

1. Select the partition via the command channel using: Open 15,8,15,"/0:partition name"
2. Format the area using the normal HEADER command or OPEN 15,8,15,"N0:name,id".

Now you have a nifty directory inside of your main directory. You can return to the main directory by either issuing a DCLEAR or OPEN 15,8,15,"I0" or selecting it via the command channel with Open 15,8,15,"/"
Note that DOS will chew up 40 blocks in your sub-directory for BAM, directory storage, etc.

You can access subdirectories from within any application program that allows you to send a command via the command channel. Most of the best packages allow this, such as Bobsterm, Pocket Writer, Paperclip, etc.

Sparrow's Slick Tips

01000000: Advanced pattern matching on the 1581
From Fred Bowen, Commodore Engineering, via USENET

The latest version of Commodore DOS, version 3D, found in the new 1581 3.5 inch disk drive has some useful additions to the powerful filename pattern matching features. The new routines inside the 1581 now accept filename extensions delimited with a period as valid matching criteria for use with the wildcard character (the asterisk). Here are some examples:

DIRECTORY"*.work"	Displays a directory of all files with the extension '.work'
DIRECTORY"*.work=p"	Displays a directory of all PRG type files with the extension '.work'
DIRECTORY"*.work=s"	Displays a directory of all SEQ type files with the extension '.work'
DIRECTORY"*.work=c"	Displays a directory of all CBM type files (partitions) with the extension '.work'
DIRECTORY"*.work=u"	Displays a directory of all USR type files with the extension '.work'
DIRECTORY"*.work=r"	Displays a directory of all REL type files with the extension '.work'
DIRECTORY"*.dir,*.work"	Displays a directory of all files with the extension '.dir' and '.work'

These pattern matching techniques will work on the DIRECTORY, CATALOG, and SCRATCH commands. Note that unlike some disk operating systems like CP/M and MS-DOS, Commodore DOS does not reserve any file extensions so you are free to create files with any extensions you like. Also note that the above file extension pattern matching abilities are only found in the 1581's version of DOS thus far, so you should not rely upon them if you are writing a program that might be used on any earlier Commodore drive such as the 1571 or the 1541.

01000001: BASICPAINT on monochrome monitors
From Louis Wallace, BASIC 8 co-author

Some of you may have noticed some problems using BASICPAINT on monochrome monitors. David Darus and I developed BASIC 8 specifically for use with RGBI monitors, and you are missing a lot if you are doing any BASIC 8 hacking on a green screen, but there is no reason why you can't use a monochrome monitor, provided you make a few alterations to BASICPAINT. You see, since David and I felt that this package was meant to be used with RGBI monitors we forgot to test it with monochrome monitors. As you may (or may not) have noticed the black and dark grey colors of BASICPAINT are impossible to read on a monochrome monitor, but fortunately this is a small oversight that is easily fixed. Here is what you must do:

Line #	Fix
72	Change the definition of the variables FC, BC, and OC to: FC=15:BC=0:OC=0
160	Change the parameters in the @CLEAR and @COLOR commands to: @CLEAR,0,0,15:@COLOR,0,15,0
1573	Change quote mode colors to CTRL-C, CTRL-white (the 2 key), CTRL-black (the 1 key). This will appear as a reverse CEP.
7096	Same change as in line 1573
9006	Same change as in line 1573
9255	Change the parameters in the @CLEAR and @COLOR commands to: @COLOR,0,15,0:@CLEAR,0,0,15
9510	Change the parameters in the @CLEAR and @COLOR commands to: @COLOR,0,15,0:@CLEAR,0,0,15
9520	Same change as in line 1573
10037	Same change as in line 1573

These nine lines will change Basic Paint from the default grey and black to white and black, providing the contrast necessary for using monochrome monitors.

01000010: Easy documentation saver
From Sparrow James, Sitting in the midst of too many manuals

If you are like me, in the midst of programming or using a complex piece of application software you are constantly finding yourself consulting the program's or language's documentation. Perhaps the most time consuming and frustrating aspect of this procedure is the "search syndrome", especially in manuals that do not provide indexes and tables of contents. In addition, the search syndrome also has a detrimental effect on the pages of your manual as they become, worn, torn, and dog-eared. Some have suggested as a solution to the "search syndrome" the use of handy command summary reference cards. But for me this solution is not a useful one, because given the state of my office I would be forever searching for the reference card itself. Also often in complex operations a short command summary is just not a replacement for detailed explanations. But recently I have come upon a solution that promises salvation for my Superbase, BASIC 8, and COMAL manuals. This solution involves the placing of colored and marked index tabs (the kind you can get from your local stationary store) at various vertical positions on the outside edges of your manual pages making it much easier to locate key sections of your documentation. Of course, none of us really has the time or desire to go through the drudgery of doing this to each of our manuals ourselves, so go and enlist the help of your spouse, children, or friend the librarian down the street. Note: When dealing with your spouse or children, your selling and description of this project is crucial. Be sure to use terms like, "family togetherness", "interesting diversion from the tube". If all else fails, I have found bribery works well. As for librarians, don't worry, I have found that librarians don't seem to mind doing this kind of "busy work" since they do it all day long anyway. In fact some of them seem to enjoy it.

Advanced VIC Color techniques

by: Miklos Garamszeghy

The BASIC 7.0 command "GRAPHIC 1" is normally used to turn on the 320 x 200 two color graphic display. Unknown to most people, it can also be used to create a graphics screen in 16 colors simultaneously! How can this be when the manual says that only two colors, foreground and background, are supported in this mode? Just to keep the non-believers at bay, try this short program:

```
1 rem 16 color graphics demo
2 rem by m. garamszeghy
3 :
10 COLOR 0,1:COLOR 4,1:GRAPHIC 1,1
20 T$="16 color hi-res"
30 FOR I=1 to 15:COLOR 1,(i+1):CHAR,12+i,0,
   MID$(T$,I,1):NEXT
40 FOR I=0 to 2:FOR J=1 to 5:COLOR 1,(i*5+j+1)
50 CIRCLE,j*70-50,i*50+60,10,10,,,,(j+i)*20
60 PAINT,j*70-50,i*50+60:NEXT J,I
```

As you can see, there are a variety of geometric shapes on the screen, so the computer is obviously in graphics mode. There are also 16 colors displayed (including, of course, the background color of black).

Let's take a closer look at what two color graphics actually means. The 320 by 200 graphics screen is really divided up into 1000 cells. Each cell has an area of 8 pixels by 8 pixels and corresponds to one of the 40 column by 25 row character positions in text mode. Two color graphics does NOT mean that you can only have two colors on the screen at any one time. It DOES mean that you can only have two colors in any one of the cells at any given time. The trick to displaying more than two colors on the screen at once is that the two colors in different cells need not be the same as those in their neighbors. Thus, each time you change either the foreground or background colors with the COLOR 1 or COLOR 0 command, respectively, you are only affecting the cells which you change by writing to (i.e. storing new graphic information, whether text or a bit pattern by one of the graphics commands) until the next COLOR command is issued. Cells which have already been colored are not changed each time you give a new COLOR command. Thus, by carefully selecting the areas to put your graphics in, it is quite possible to display up to 16 colors on the 320 x 200 screen.

The COLOR command is used to alter the screen colors in any of the C-128's text or graphics modes. The color data for the bit map graphics screen are stored in the C-128's RAM from $1c00 to $1fff in column major order, if the graphics screen has been allocated. The low nibble ($0x) holds a number from 0 to 15 representing the background color for a given cell, while the high nibble ($x0) is the foreground color, also 0 to 15. The color codes used in the nibbles are all one less than the corresponding codes used in the COLOR command. In other words, a hex color byte of $16 for a given cell would mean that the foreground color was white (color code 2) and the background was blue (color code 7).

If you wish to have more than two colors in one cell, then you can overlay the cell with a stationary sprite. Sprite colors are independent of the main screen colors and can thus be used to add color to the screen if they are not needed for other purposes.

Incidentally, the graphics color RAM area is at the bottom of the normal BASIC text work space when the graphics screen is not allocated. When you allocate the graphics screen without clearing it with a BASIC program in memory, you will see a random series of colors on the screen. This is caused by the presence of BASIC text bytes in the color memory area. (When the graphics screen is allocated, the BASIC program is not MOVED to a higher RAM location. It is merely COPIED. The original remains intact until it is overwritten by a screen clear or other graphics command thus producing random color patterns.)

This next example can be used to demonstrate random colors. It will fill random areas of the color map with random bytes from 0 to 15, representing the 16 possible background colors.

```
1 rem 16 color random demo
2 rem by m. garamszeghy
3 :
10 COLOR 0,1:COLOR 4,1:GRAPHIC 1,1
20 X=RND(0)*16:Y=RND(0)*1023+7168+RND(0)*12
30 POKE Y,X:GOTO 20
```

After running the program for a while, hit the <run/stop>-<restore> key combination and enter the MONITOR. You can then display all of the color codes, even change them if you wish, with the MONITOR command: M 1c00. If you have separate 80 column and 40 column video monitors, you can display the graphics screen on the 40 column display and the MONITOR dump on the 80 column screen. This will allow you to directly observe the effect of changing the bytes in the color RAM area.

Now what use is all of this? Well, for one thing, the method can be used to create multi-color backgrounds for games, etc. without having to resort to the lower resolution of the 160 x 200 pixel "multi color" graphics mode. This same technique can also be used in multi color graphics mode to display 16 colors on the screen, but with 4 colors in each cell.