

The Magazine For ALL Commodore Computer Users

• Exploring The C-128

• Extending The Cruise  
• More C-64 music

• Commodore 128  
• Memory Map

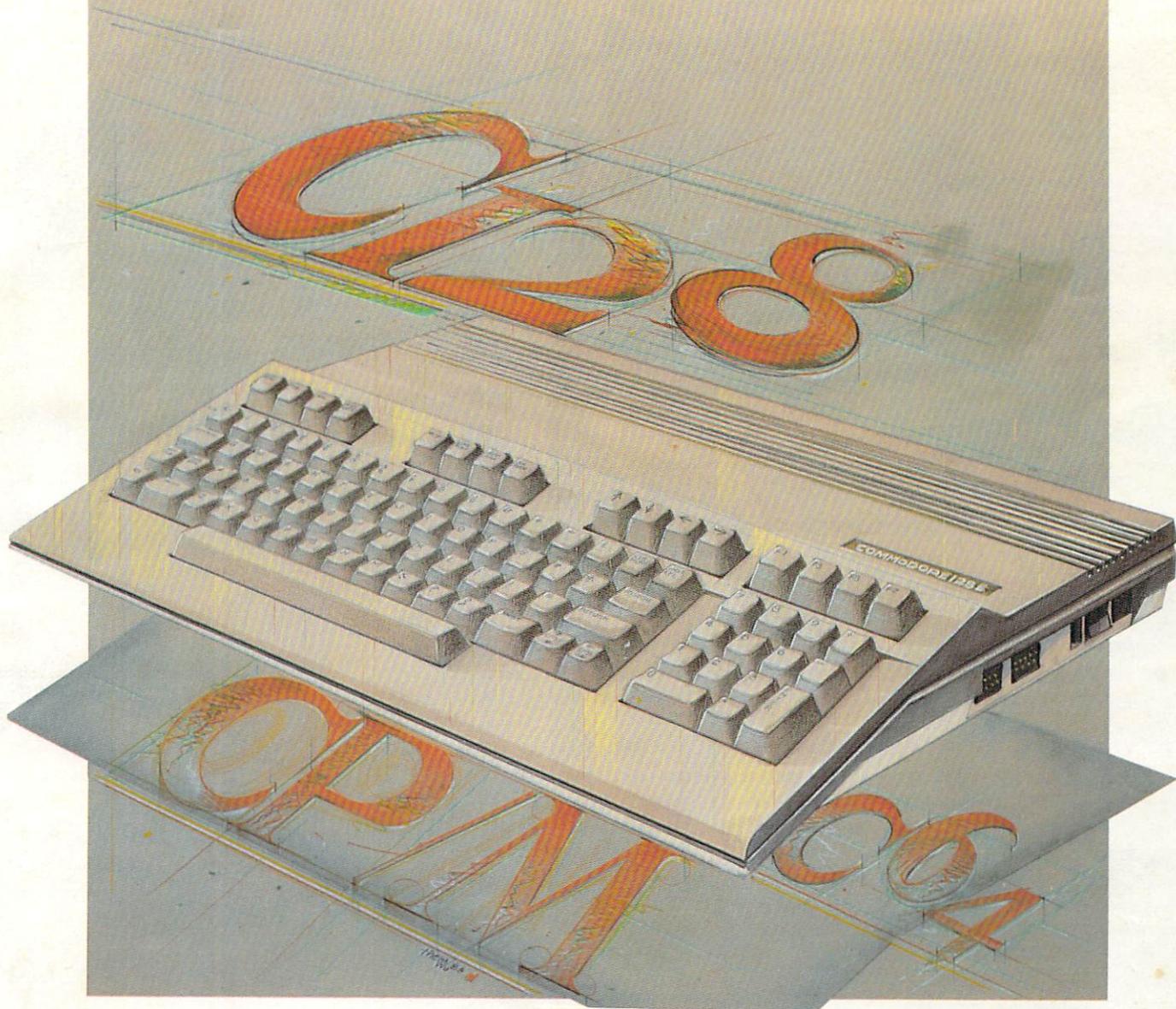
• 8-page pull out reference

• Getting Started  
• With CP/M

\$2.95

Issue No. 21

**TPUG** Magazine



Bulk rate  
U.S. postage paid  
Buffalo, N.Y.  
Permit # 1575



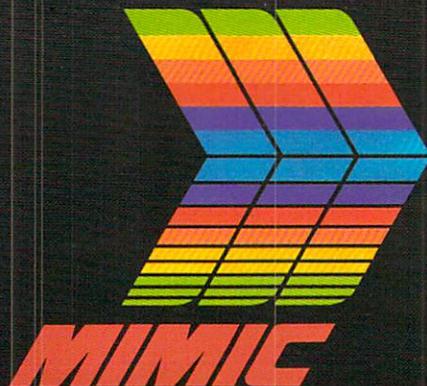
# HOW TO GET OVER \$2000 WORTH OF NEW CAPABILITIES FOR YOUR COMMODORE 64™



## FOR \$599\*

The Spartan™ is the Apple™ II+ emulator for your Commodore 64™ that will open up a whole new world of hardware and software for you! Imagine adding these features to your Commodore 64™ for the Spartan™ price of \$599:  Apple™ II+ hardware and software capabilities  64K RAM expansion  four software selectable Commodore 64™ cartridge slots  non-dedicated 8-bit parallel port  standard audio cassette deck capabilities for your C-64™.

The suggested retail value of comparable products offering only these capabilities is over \$2200.00\* — but the Spartan™ gives you much, much more! By building on your investment in your Commodore 64™ — an excellent introductory computer — you create a whole new system with both C-64™ and Apple™ II+ capabilities. There is a whole other world out there! The huge selection of Apple™ II+ hardware and software is now yours to explore! Call toll free for the Spartan™ dealer nearest you.



FOR INFORMATION WRITE:  
MIMIC SYSTEMS INC.  
1112 FORT ST., FL. 6L  
VICTORIA, B.C.  
CANADA V8V 4V2

TO ORDER CALL:  
**1-800-MODULAR**  
(663-8527)

\*All prices quoted are in U.S. funds, freight and taxes not included. Value of components equivalent to the Spartan™ system are quoted from Apple™ II+ CPU and Apple™ II+ single disk drive 1983 list prices, and from current suggested list prices and component specifications of other peripheral manufacturers. Commodore 64™ and Commodore logo are trademarks of Commodore Electronics Ltd. and/or Commodore Business Machines, Inc. Apple™ II+ is a trademark of Apple Computer Inc. Spartan™ is a trademark of Mimic Systems Inc. and has no association with Commodore Electronics or Apple Computer Inc. The Spartan™ is manufactured by Mimic Systems Inc. under license granted by ATG Electronics Inc. of Victoria, B.C. Canada.

# TPUG Magazine

*Publisher:* Bruce Hampson  
*Editor:* Nick Sullivan  
*Assistant Editor:* Tim Grantham  
*Production Manager:* Astrid Kumas  
*Editorial Assistant:* Iwona Sukiennik  
*Cover Painting:* Thom K. Wu  
*Creative Direction (Cover):* Steve MacDowall  
*Cover Scan and Assembly:* LK Graphics  
*Typesetting:* Noesis, Toronto  
*Printed in Canada by:* Delta Web Graphics  
Scarborough, Ontario

TPUG Magazine is published 10 times a year by Toronto Pet Users' Group (TPUG) Inc., the world's largest Commodore users' group. TPUG is a non-profit corporation dedicated to the service and support of owners and users of Commodore computers. All rights to material published in TPUG Magazine are reserved by TPUG Inc., and no material may be reprinted without written permission except where specifically stated.

*Correspondence:* Send change of address and subscription inquiries to: TPUG Inc., Address Changes, 101 Duncan Mill Road, Suite G7, Toronto ON, Canada M3B 1Z3. TPUG Magazine welcomes freelance contributions on all aspects of Commodore computing. Contributions should be sent on disk, though accompanying hardcopy is welcome. Be sure to include return postage if you wish materials returned. Please indicate on the disk label which Commodore disk format and word processing program you have used. Payment for articles published is \$30.00 per page if the author retains the copyright, and \$40.00 per page if the copyright is assigned to TPUG Magazine. Payment is made on publication. All contributions are subject to editing for length and readability. Address editorial contributions to: The Editors, TPUG Magazine, 101 Duncan Mill Road, Suite G7, Toronto ON, Canada M3B 1Z3.

#### *Circulation*

Subscription 12,000      Newsstand 10,000  
ISSN #0825-0367

VIC 20, Commodore 64 and SuperPET are trademarks of Commodore Electronics Ltd. PET is a registered trademark of Commodore Business Machines, Inc. CBM is a registered trademark of Commodore Electronics Ltd.

Subscriptions to TPUG Magazine may be obtained by joining the Toronto PET Users' Group (TPUG) Inc.

Regular member (attends meetings)	\$35.00 Cdn.
Student (full-time, attends meetings)	\$25.00 Cdn.
Associate (Canada)	\$25.00 Cdn.
Associate (U.S.)	\$25.00 U.S.
	\$30.00 Cdn.
Associate (Overseas - sea mail)	\$35.00 U.S.
Associate (Overseas - air mail)	\$45.00 U.S.

For further membership information please contact:

TPUG Inc., Membership Information	
101 Duncan Mill Road	1552 Hertel Ave.
Suite G7	Suite 144
Toronto ON	Buffalo, NY
Canada M3B 1Z3	USA 14216-2882

TPUG Telephone Numbers:  
Business Office (416)-445-4524  
Magazine Office (416)-445-9865  
Advertising Sales Representatives:

Hunter Nichols Inc.  
721 Progress Avenue  
Scarborough, Ontario M1H 2W7  
Telephone: (416)-439-4140

# DIRECTORY

86

TPUG Magazine

MAR

## Feature: Getting To Know The Commodore 128

- 8 Getting Started With CP/M by Adam Herst
- 9 Printing Help Files by M. Garamszeghy
- 11 Autobooting On The C-128 by M. Garamszeghy
- 12 A Scrounger's Guide To CP/M by M. Garamszeghy
- 14 C-128 Book Flawed But Useful by Jim Butterfield

## Articles

- 16 Amiga Dispatches by Tim Grantham
- 18 Extending the Cruise by Tim Grantham

## Reference Section

- 21 Commodore 128 Memory Map

## Micro Processes

- 32 Amiga/C-64 File Transfers by Chris Johnson
- 32 Formatting MFM Disks by M. Garamszeghy
- 33 A Secret In A DRAW Command by Elizabeth Deal
- 34 Printing Graphs From The Home Accountant by Karl Thurber
- 34 Organize Your Disk Directories by Karl Thurber
- 35 Inexpensive Disk Drive Fans by Karl Thurber
- 35 Saving Disk Directories by Karl Thurber

## Reviews

- 36 Movie Maker by Mike Martin
- 36 Printer Book For The C-64 by Ranjan Bose
- 37 Print Master by Mike Martin
- 38 Typing Tutor III by Malcolm O'Brien
- 39 Starting FORTH by R.C. Walker
- 40 Trolls and Tribulations by Robert Sodaro
- 40 Mapping The Commodore 64 by Ranjan Bose
- 40 Graphics Book For The C-64 by Ranjan Bose
- 41 Superman — The Game by Malcolm O'Brien

## Departments

- 2 Inside Information
- 4 Line Noise
- 6 The Answer Desk with Malcolm O'Brien
- 7 BBS Password for March
- 10 Marketplace
- 29 TPUG Software Order Form
- 30 Additions To The TPUG Software Library
- 42 Products Received with Astrid Kumas
- 44 Calendar of TPUG Events
- 45 Bulletin Board
- 45 Classifieds
- 48 TPUG Magazine Distributors
- 48 TPUG Contacts
- 48 Index of Advertisers

# Inside Information

## BUG and CBUG

We have received word from Norman Deltzke, president of the Chicago B-128 Users' Group (CBUG), that the national B-128 User Group (BUG) is no longer in operation, and that its director, Marlin Schwanke, is no longer attempting to operate the group or publish the newsletter that he had founded. Because of this, we must now advise you to *not* send away membership fees to the address we gave for BUG in our December 'Bulletin Board' section — there is apparently no longer anything to send away for.

CBUG, in contrast, is thriving, and looks like an important source of information and support for B-128 users. The club now claims about 2000 members in the US, Canada, Australia, New Zealand, England, Sweden, Germany, Italy and other countries, and a heavy publishing schedule. Activities include local meetings in the Chicago area, and support via telephone and newsletter for members elsewhere. CBUG's mailing address is:

### CBUG

c/o Norman Deltzke  
4102 N. Odell  
Norridge, Illinois 60634

Membership fees are \$10.00 (US) a year.

## This month

This month's feature on the Commodore 128, and particularly its CP/M mode, is largely the work of Adam Herst and Miklos Garamszeghy, two Toronto-area authors who see outstanding potential in this most versatile of 8-bit machines. Look for more articles on the 128 from Adam and Miklos in issues to come.

While we're talking about the C-128, owners of the 1571 disk drive will want to know of the following DOS bug, which we just saw reported in the latest copy of *Info* magazine. Look there for details but, briefly, the problem arises when you attempt to validate, in 1541 mode, a disk that was formatted double-sided in 1571 mode. After this operation, the 1571 mode will no longer recognize that your disk is double-sided, and unless you rectify the misunderstanding right away, you will in the course of subsequently saving and loading files become the proud owner of an irretrievably corrupted disk.

The fix, if you find yourself in this situation, is to use a direct access command to write \$80 (decimal 128) to byte 4 of track 18, sector 0. *Info* provides a short program to do this job for those who can't manage it on their own.

## Blank disks

Due to a hardware glitch in one of the 4040 disk drives, a number of blank disks were sent to members. The problem has since been corrected, but anyone who

received one of these disks can get a free replacement by mailing or telephoning their name, address, and membership number to the TPUG office.

## Amiga feedback problem

TPUG is looking for member feedback on the establishment of an Amiga chapter. Any person interested in becoming a coordinator for this chapter should contact Bruce Hampson, General Manager, at the TPUG Office.

The editors

## The state of TPUG

Much has happened at TPUG in the past three months. Perhaps most important has been the installation of a new computer system. This IBM-compatible system allows us much faster access to our 18,000 membership files and records, and allows us to keep better control over our orders. With the new system, we need less staff to give our members much better service with fewer problems.

Our finances are in great shape, and we are on a sound foundation with some innovative programs coming up. We have purchased some new office equipment and systems and are embarking on a new membership drive. Membership is growing again, and more and more associate clubs are joining us. We have also straightened out most of the complaints and problems.

One problem we are still having is with our software library. Major changes are being worked on in that area, including the acquisition of a high-speed disk duplicator, and a reworking of the disk cataloguing system. The Disk of the Month subscription series was even more successful than we had hoped, which led, ironically, to more problems. These are now almost completely solved.

One of our new staff members, Sally Murciano, is working almost exclusively on associate clubs. With more than 65 clubs and 3000 of our members included in this category, we have decided to spend even more time and resources on this section. If you are a member of a users' group not yet affiliated with us directly, write to Sally for more details. We can offer you a considerable savings on membership as well as some other great benefits.

We have had a number of problems lately with cheques and correspondence from our members. Please remember to include your membership number on *both* your letter *and* cheque. This is especially important for people who hand-write their letters instead of typing them. Sometimes signatures are hard to decipher, and without your membership number we have no way of getting back to you.

We would also appreciate it if you would not post-date cheques. These are very confusing and cause us great problems. If you must post-date, then please make sure that they are well marked (perhaps in a different colour) to show us. Some days we might process 500 cheques, and often these get through the system before the bank catches them.

We also appreciate the number of letters we have received expressing your support for us. We have just come through some difficult times with flying colours, and your understanding and patience have been most appreciated. Now that we have most things settled, we look forward to serving you even better.

Bruce Hampson  
TPUG General Manager

# TPUG library disk

Toronto PET Users Group Inc.  
101 Duncan Mill Road, Suite G7  
Don Mills, Ontario, Canada M3B 1Z3  
Tel: (416) 445-4524

# TPUG LIBRARY DISKS

10 months of the latest,  
fabulous public domain  
software only

**\$79.95** CANADIAN  
US orders US\$59.95

Subscribe now to start  
receiving the TPUG disk of  
the month

You **SAVE 20%** or more by using this offer as a TPUG member

**THIS OFFER VALID FOR PAID UP TPUG MEMBERS ONLY**

PLEASE PRINT

Name: \_\_\_\_\_ Membership # \_\_\_\_\_

Address: \_\_\_\_\_

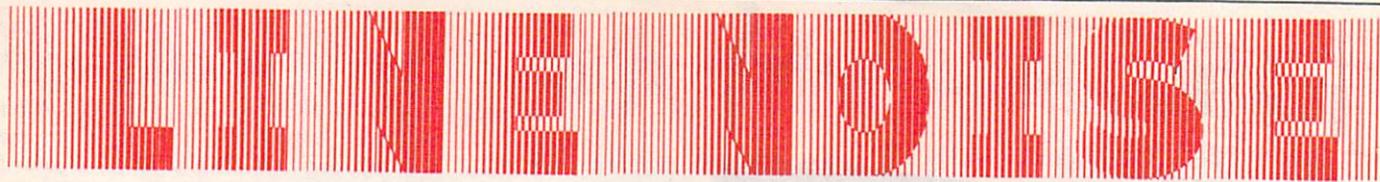
## Type of Payment

- Cheque                      Amount \$ \_\_\_\_\_  
 Mastercard                Currency  Can.  US  
 Visa                            Credit Card # \_\_\_\_\_  
 Money Order                Expiry Date \_\_\_\_\_

Signature \_\_\_\_\_

## Type of Computer

- C 64  
 VIC 20  
 PET  4040  8050  
 SUPERPET (10 disks)  4040  8050  
 C128 (1541 only)  
 MS/DOS  
 AMIGA (3½") (7 disks)



TPUG Magazine invites you to express your views on Commodore computing by writing to:

Line Noise  
TPUG Magazine  
101 Duncan Mill Road, Suite G7  
Don Mills, Ontario M3B 1Z3  
Canada

## Amiga skepticism

Your articles on the Amiga seemed particularly directed to people who like to simply fool around with computers: as one of those very people, I'd like to tell you about my first two weeks of Amiga ownership — its joy and its frustration.

The sound and colour are both substantially better than anything else I've played with and, in particular, the tremendous voice capability is worth hours of exploring its possibilities. Beyond that, my initial reactions have been negative. Most of my frustration results from not having any documentation on the machine — something that will, I suppose, be cured in time. Here I live in the shadow of the Company itself, know some of the employees, and get constant assurances that the documentation is available. The hard fact is that I can't get it, and have spent much of the past two weeks unsuccessfully trying to connect my printer, program the function keys, transfer files from one disk to another, or get into the fabled high-resolution mode — the kind of thing that was readily done on the C-64 because it came with a *Programmer's Reference Guide*. The absence of an equivalent volume makes the Amiga unready for marketing to programming buffs, and the total absence of software makes it unready for other users.

A big disappointment is the editor built into ABasiC. It is 1982 state-of-the-art, lacking the friendliness and power of even our present-day editors, let alone the bold new future promised by the advertising hype. Another disappointment is the slowness with which it starts up. Kick-starting is the way it was done in 1975 — there are better ways available now.

Much is made of the graphics capability of the Amiga. Its ABasiC commands have some new scope but they are not as powerful as I expected, not at all revolutionary, and not nearly so good as those of the MacIntosh. Mostly, new names have been given to old graphics commands, and the clumsiness of standard plotting routines has been retained. I was hoping for, and will probably get in the future, a substantially easier system for manipulating the screen display.

My advice to TPUG readers is, Don't Throw Away Your C-64! The Company may make more money changing models all the time, but hackers lose by it. We will get more return on our effort by staying with one model, continuing to build up our library of software and a warehouse of hardware attachments to fit that model. I strongly feel we users should do whatever is necessary to encourage Commodore to continue supporting the C-64, and do what we can to counter the trend among merchandisers to dump it in favour of these new models. The advantages claimed for the new models simply do not justify abandoning our C-64 software and hardware.

Deep in every hobbyist there always lurks a strong urge to get something newer and better, an urge that usually gets expressed as updating to some newer model. Loss of compatibility is a terrible price to pay — let me suggest that an alternative way to satisfy this craving is to go in for peripherals. Add on voice control and voice generation. Add on a hard disk drive. Keep your C-64 as the core of your system and connect up a video camera so it can see as well as hear and talk. If you don't already have a 1520 plotter, get one — it's marvellous (and also on the endangered list, like the C-64 itself).

Let's mount a campaign to promote enhancements for the C-64 rather than replacements. Save the C-64!

Walter Stauber  
Springfield, Pennsylvania

*You're right — Amiga documentation is hard to come by. But let's not be too nostalgic for the good old days of the Commodore 64 — the Programmer's Reference Guide did not come with the machine, but had to be purchased*

*separately, just like the documentation for the Amiga. Nor was it available simultaneously with the 64's release.*

*Another point on the same subject: the 64's Reference Guide was good, particularly by the Commodore standards of yesteryear, but it wasn't perfect. How much of what you know about the 64 came from magazine articles published over the last three or four years, how much from Jim Butterfield's memory maps, how much from other books? It will admittedly take a long time before we'll see that quantity of information for the Amiga, but this is true of any new machine. And though they are still in short supply, Commodore's programming manuals for the Amiga are both comprehensive and detailed. With such a complex machine, that means a lot of documentation — and some of it is even comprehensible.*

*ABasiC is a dead language now, of course, which no one is lamenting. Microsoft's AmigaBasic is vastly superior, and has a full screen editor too. Sure, you have to load it in from disk instead of having it pop up instantly when you switch on your machine, but there are some small compensations. For instance, try writing a 1200-baud terminal program in BASIC on the Commodore 64. You can do it in half a screen in AmigaBasic. How about a graphics package, with lines, circles, boxes and area fills, and with a screen resolution equal to the 64's hi-res mode, but with 16 colours instead of two? I've seen three or four such programs already on the Amiga, and they all have two things in common: they're fast, and they're short.*

*And here's something you didn't mention — multitasking. Until you've used the Amiga for a while, it's difficult to appreciate just how convenient the multitasking can be. The ability to skip back and forth between programs without having to shut any of them down is something you'll eventually take for granted — and you'll feel strangled when you have to go back to a single-task machine. The graphics and the sound are nice — spectacular even, by the standards of most computers — but I think it's the multitasking that will have the profoundest effect on hackers and users alike.*

*Throw away your Commodore 64? No, absolutely not. There are times when the*

*straightforwardness of the 64's user interface is just what you want and need — and of course, you've got all that software. But don't give up on the Amiga, either. It isn't perfect but, in a lot of ways, it's the best there is.*

Nick Sullivan

## On copy protection

I have been a Commodore computer user and a member of TPUG for several years now, and I have enjoyed both the computers and my memberships in TPUG. I have been quite disturbed by the past several issues of *TPUG Magazine*, and in particular by the November 1985 editorial.

The trend that has disturbed me is TPUG's apparent support of copy protection. While I am certainly not in favour of software theft (a term I prefer to piracy, which has certain romantic overtones), I am equally opposed to copy protection of software.

In particular, the last issue's editorial equated programs designed to copy or 'break' protected disks with 'piracy tools'. This implies, I suppose, that owners of these programs are in fact pirates, and magazines that advertise them are advocating piracy.

While it is true that these programs could be used for such illegal purposes, to call them 'piracy tools' is absurd. It is the same as calling people who own modems and terminal software 'computer criminals', since modems and terminal software are indeed necessary tools for anyone desiring to break into a system. Most owners of such equipment have probably never even tried to do such a thing, just as most owners of 'copy busting' programs are probably not pirates.

While I am not familiar with Canadian law, the copyright laws in the US give the original purchaser of a program the legal right to an archival copy. Copy protection attempts to deny this right to the legal purchasers, and copy-busting programs attempt to give it back. Consider someone who makes his living as a writer, and uses the computer as a tool of the trade. Only a fool would use a program that could not be backed up — deadlines could be missed, et cetera.

As personal computers grow more powerful — and Commodore computers, in particular, have grown enormously in power with the introduction of the C-128 and the Amiga — copy protection becomes increasingly less desirable. Many owners of Amigas and 128's will undoubtedly wish to purchase hard disks. I do not envy these individuals their future attempts to move copy-protected

software over to a hard disk. Amiga owners may wish to run concurrently two or three programs. Copy protection often involves replacing the DOS code with code from the protected programs — what is going to happen when the BCD word processor tries to change the code already installed by ACME spreadsheet? Several scenarios spring to mind, involving things from smoke to destroyed disks, lost data, and grey hairs.

The current trend is clearly anti-copy protection. A large number of software houses have announced that they have dropped copy protection on all of their products, due to the hardship forced upon their users. Publishers of software such as Borland (**Turbo Pascal**) have proven that copy protection is not necessary to a successful software business. Other publishers (such as Sirius Software) have demonstrated that copy protection does not ensure success by copy protecting their software and entering Chapter 11.

I would submit to you that, given the choice, most people would choose a non-protected program over a protected one. I would also suggest that there is strong evidence that the tide is turning against protection, both by legitimate users and legitimate software publishers. I would further suggest that without copy protection, 'copy-busting' programs would not exist. It seems to me the course of action is clear — if you really wish to stamp out these piracy tools, help your users, and remain in the forefront of the computer revolution (which TPUG always has), you should refuse to advertise any protected software, and advertise instead only the copy-busting programs.

Consider — if all magazines had this policy, in order to bring their products to the attention of the public, software houses would have to drop the protection on their products. This, in turn, would cause the Copy Buster programmers to go out of business, or turn to other products: what good is a copy-busting program when there is nothing to bust? People would be able to move programs to their hard disks without worry, Amiga owners would be able to multitask to their heart's content, and the software publishers would whistle all the way to the bank.

Scott Ballantyne  
New York, New York

*I agree that the trend among software publishers is away from copy protection, particularly on software directed at the new machines, and I too am glad that it is so. Nor do I personally oppose the making of archival backups of protected software, or utilities that make this possible.*

*But copy protection, with all its faults, is not a conspiracy on the part of software publishers to make life difficult for computerists, however common that side-effect may be. And it is a response to piracy (though perhaps, ultimately, a misguided and self-defeating one), not the cause.*

*Early, unprotected, commercial programs suffered widespread illegal duplication. It was not unnatural for the software manufacturers to seek methods of ensuring that the users of their products were also paying customers. Copy-protection is one of those methods; another one is the use of copious documentation, since information is far less conveniently copied when it comes in printed rather than magnetically-encoded form. The latter method is becoming more popular as the software itself becomes more complex: it is significant that most of those companies that do not use copy-protection are those whose products require heavy documentation; whereas games manufacturers, for instance, have been slower to abandon what for them is their only means of defence. It may be that it will one day be possible to abandon protection, along with such other inconveniences as locks on doors, border guards, and access codes for banking machines; meanwhile, we have to accept that we belong to a society that views all these impediments to civilized living as necessary and normal.*

*Let us set aside the question of legal rights, for surely there are conflicting rights here — ours to copy, the publishers' to protect — and agree that there is a case for copy-protection, whether or not it is finally convincing. How do the publishers of deprotection software fit into the scheme of things, and how do they present their products? Very often with an overt appeal to the romantic image of piracy you decry. Have you seen the full-page colour ads for a product called **The Shadow**? It is not directed at the freelance writer who needs a security backup of a word processor.*

*What amazes me about the magazine ads for deprotection utilities is not their existence, but their quantity. The fact that this rather arcane specialty is so prolific and so vigorous can't be unrelated to the huge amount of illegally-cloned software available in every schoolhouse and every corporate office-building. It may be that, by refusing ads for these utilities, TPUG has abstained from the battle against copy-protection; it may also be that magazines that accept such ads are putting the ability to pirate software within reach of the mass of consumers, rather than just the ingenious few.*

Nick Sullivan

# The Answer Desk

with Malcolm O'Brien

If you have a question for The Answer Desk, write to us at:

Answer Desk  
TPUG Magazine  
101 Duncan Mill Road, Suite G7  
Don Mills, Ontario M3B 1Z3  
Canada

## C-64 Meets Waterloo

Is Waterloo BASIC available for the C-64 and, if so, where can I get it?

Joseph Srdarev  
Etobicoke, ON

Yes, Joseph, it is. Waterloo Structured BASIC is in wide use on PETs, and is part of the curriculum in many schools and community colleges. It is favoured by educators, since its structured programming statements encourage the writing of more readable programs that are much easier to maintain (and to grade!). Subroutines take the form of named procedures that can be **CALL**ed by name. It also includes commands such as **AUTO**, **DELETE** and **RENUMBER**, which are very helpful editing tools. You can order it from:

WATCOM Products  
415 Phillip Street  
Waterloo, ON  
N2L 3X2  
(519) 886-3700

The software is on cartridge, and the package includes a comprehensive textbook containing both a primer and a reference manual. WATCOM also offers seminars on BASIC and WATCOM Pascal. Call or write for pricing information.

## Fade-in diagnosis

Sometimes when I switch on my C-64 system, the blue Commodore screen doesn't appear on my monitor — it's definitely on, but the screen is a blank dark grey. After several unsuccessful attempts, there will come a time when the regular blue Commodore screen actually fades on after a few moments. Sometimes it doesn't, and I just let my computer 'cool down' for a few hours. After that time, it switches on fine. I feel the problem is the

computer, not the monitor. No one I have talked to can tell me what's wrong (if anything). Your help would be appreciated.

M. Kanas  
Mount Albert, Ontario

Intermittent problems are the hardest to diagnose and to fix. My guess is that your problem is a faulty power supply. Try another power supply, if you have access to one, and see what happens. If that produces the same result, you'll have to investigate further. You can isolate the faulty component in your system if you proceed step by step.

First, disconnect your 64 from the monitor and hook it up to your TV, using the cable and RF modulator that came with the 64. If you still have the problem, it is definitely the computer itself, and you should have the VIC II chip checked. If the 64 works fine with the TV, reconnect the monitor, but this time try the inputs on the front of the monitor instead of the back (or vice versa if you've been using the front ones). If you still have the problem, try a different video cable. The last link in the chain is the monitor itself. Try a different monitor, or try using your monitor with a VCR. Somewhere in the course of all this, the offending item will be revealed. Although this process of replacing components and trying different things can be tedious and time-consuming, it is almost always effective for ferreting out 'gremlins'.

Back in the August/September issue, I gave the name and address of Cynthia Wood (HHB 32d AADCOM CMR 2122, APO, New York 09175) who has offered to be a resource for the Answer Desk. Cynthia's specialty is **Superbase**. She has since fielded several **Superbase** enquiries, which are reproduced here along with her answers. Thanks a lot, Cynthia!

## Railroad management

I have been using **Superbase 64** for a variety of address lists, and find it very valuable. However, I would like to use it to run my model railroad pass list on, but it would appear that you can't have a combination 'Key list' and 'Number list'. Or can you? If you can, how would it be arranged? I would like to have the pass numbers (001 on up) as the first item, followed by name, address and the like. I would also like to take the pass list and

use it to put the names and so on in alphabetical order. Is this possible?

S. Richard Knotts  
Wheaton, Maryland

If I understand your question correctly, **Superbase** will readily handle your problem. You may choose either the **Number** field (001 and so on) or the **Name** field to be your **Key** field. This will depend on the order in which you want the records to appear when you are working with the file on-screen (as opposed to outputting to printer or creating reports) — mainly entering and editing your file. If the **Number** field is the **Key** field then your records will be in numerical order. If your **Name** field is the **Key** field, they will be in alphabetical order. The **Key** field does not have to be the first field of the record.

If you choose the number field to be your **Key** field, follow this procedure for creating an alphabetical 'key list' for output and report uses (see section 10 in the Reference section of the User's Manual):

1) From Menu 2, press **f4** to choose the Sort option

2) Type: **all on [name] to "halphlist"**

Put the name of your **Name** field in the brackets and call the list (in quotes) anything you want. Start it with an **h** if you want to be able to view the list with the **Help** option, and end it with the word **list** to prevent accidentally writing over other files by choosing an identical name. This is a good habit. I use **-base**, **-list**, **-export** and similar file-type suffixes to prevent accidental erasures. Files within a database are the only things that I don't give an identifier.

3) Press **return**. **Superbase** will create an alphabetical list of your files. Viewing it with the **Help** option will display the numbers of each record. To see the records in the keylist, use the **Output** option:

1a) From Menu 1, press **f4**

2a) Type: **from "halphlist" [number] [name] [address] [state]**

The brackets enclose your field names. These may be omitted, in which case all fields will be listed, or any number of fields may be chosen for viewing. Also, the word **down** (as in **down from "halphlist"**) may be inserted to display the fields in a column instead of **across**. **Down** and **across** remain in effect until

the other is specified. The same goes for **display** and **print**, which select the screen and the printer respectively.

3a) Press **return** and watch your records turn up alphabetically on the screen

4a) Output to printer by using **print from...** in step 2a above.

There is another interesting feature of the **Sort** option that allows you to create a list in descending numerical or reverse alphabetical order. Simply insert a **D-** in the sort command (step 2 above becomes **all D- on...**). Your sorts can also be layered (alphabetical first names within alphabetical last names, for example) by specifying more than one field in the sort command.

### Output hangup

I have two questions:

1) When I wish to print out a list or report to my printer, I'll get a few lines of the data and then the printer stops. The disk will continue to spin, but nothing is output to the printer. I get out of it by disconnecting the printer. I end with an 'I/O 5' error. I am using a Cardco +G interface, with transparent mode on and auto LF off. I use 'dev 4, pdef 5'. Can you give me any hints as to what my problem may be and possible measures to correct it?

2. When loading a database, the screen shows the database catalog of files. When I wish to scratch a file, I enter the Main-tain menu, then use the other option to scratch the filename from the disk (which does happen). My problem is that the filename remains in the Database Catalog, even though the file itself is gone. How do I remove it?

Matthew Brown  
Chilliwack, BC

1) I tried to duplicate the problem and it refused to be a problem. I have a Gemini 10X printer and the same interface. I normally have the transparent mode off (interface dip switch seven on) but I turned it on (SW7 off) to match your set-up. Keeping auto LF off (my usual mode) I changed the pdef to 5 (I normally use 0), and still came up with no printing problems. You did not say which printer you have. I would suggest you try turning the transparent mode off and using pdef 0. If that doesn't work, I don't think I can help. By the way, the 'I/O 5' is a Kernal message meaning 'Device Not Present', which really tells you only what you already know: that there has been a catastrophic breakdown in communications between the computer and the printer.

2) To delete a file you must first delete all

records (with **d** on the **Select** menu, confirming each deleted record with **y**). Then from the main menu, type **select delete** to delete the file. The filename should now no longer be in the catalog, and you can delete the file from the disk.

### Sort with duplicate keys

I have an address list with an added numerical field, and I would like to have a printed list in numerical order. When I attempted to sort on the numeric field, I ran into a problem that I had experienced before when using the sort option: some numbers are printing out of order. When the print routine accesses the record, it does so by the key, and in this case looks for the first Smith that it has available. Because there are several Smiths, the one it grabs may not have the proper number. Do you know of any other way around this problem?

Bob Adler  
Kenilworth, New Jersey

Here is a program that will enable you to print your pass list numerically. It will probably be a very slow process but it works. The manual leads me to believe that there is another way: Section 4.1.4, page R-20, says to devise a program using **Select Next**, but I don't think it's possible to **Select Next** when outputting. We would have to be able to stop the output process to check each record to make sure it is, indeed, the Smith we want (using **if/then**), and selecting the next record if it isn't.

```

10 rem sort on number field
   with duplicate keys
30 rem for screen display a
   dd "01,n" before first f
   ield in line 230
100 ask 04,4"Enter total nu
   mber of records ";r
110 print 030"PASS LIST" 01
   "No." 06 "Name" 035 "Ad
   dress" 060 "City"
190 n=0:n1$=""
200 n=n+1:n1$=str$(n):if n<
   10 then n$="00"+right$(
   n1$,1)
210 if n>9 then n$="0"+righ
   t$(n1$,2)
215 if n>99 then n$=right$(
   n1$,3)
220 select match where [num
   ber] is n$
230 print [number] & [first
   names] & [lastname] 030
   & [address] 055 [town]
240 select last
    
```

```

250 if n=r then display:men
   u
300 goto 200
    
```

Lines 200 and 210 set up a variable based on the number **n**, which is increased by one each time through. If **n** is less than 10, it must be preceded by two 0s; if greater than 9, by only one. The **right\$** function eliminates the space reserved for the sign of a number variable so that **n\$** won't be "001". Line 240 insures that each search will begin at the beginning of the file. □

## TPUG BBS

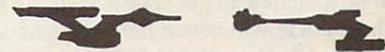
The NEW telephone number is:

### (416) 273-6300

Operating hours:  
24 hours per day  
7 days per week  
The password is...

## OUTER

THE ULTIMATE SIMULATION



## STAR TREKING

THE GAME

### Version II

For The 64 And 128

- Impressive Graphics With Hi-Res Features
- Realistic Sound
- Strategic Thought Involvement
- The Only Game That Parallels The T.V. Series And Movies



UFLAND  
SOFTWARE  
INC.

Dept. A  
P.O. Box 1324  
Meaford, Ont.  
N0H 1Y0, Canada  
(519) 538-1758

Not Available On Cassette

**\$21.95 Cdn.     \$19.95 U.S.**

First Class Postage Included  
Ont. Residents Add 7% Provincial Sales Tax

*Even Spock Would  
Find It Challenging*

# Getting Started With CP/M

by Adam Herst

Copyright © 1986 Adam Herst

In their advertising, Commodore compares the C-128's 'higher intelligence' with the fallen fruits of the Apple orchard. Unfortunately, Commodore forgot to include in their comparisons one of the C-128's most powerful features, a Z80 microprocessor and its accompanying operating system, CP/M. This feature makes the C-128 one of the lowest-priced entries into the ranks of dedicated business computers, and opens up a programming environment never before seriously implemented on Commodore computers.

In CP/M mode, you have access to CP/M Plus, the latest version of CP/M: the same operating system that controls countless other computers from a variety of manufacturers. Previous versions of CP/M are compatible (with restrictions) with CP/M Plus and a large base of programs exists: from languages to utilities, databases to wordprocessors, available both commercially and from public domain libraries. Using these programs requires little if any knowledge of CP/M other than how to get your system set up. This article will cover the basics of setting up CP/M on the C-128, and will tell you how to extract a forty page Help manual from your CP/M systems disk.

What is CP/M? You can think of CP/M as an extremely sophisticated and powerful DOS wedge. Its primary purpose is to manipulate, and maintain status information about, the physical devices that make up your system, the disks the system is operating on, and the files that are on those disks. A bonus is the inclusion of all the tools necessary for a complete assembly language programming environment.

## Physical devices

Most CP/M systems consist of a computer running CP/M and at least one disk drive. CP/M on the C-128 can support multiple disk drives (almost a necessity), and up to two printers. If you've used more than one disk drive on a CBM machine before, you are probably aware that they can support up to four disk drives assigned device numbers 8 through 11. CP/M mode mimics this setup and supports up to four

physical drives assigned device letters A through D. To set up a multiple-drive system, you must change drive device numbers through hardware, or through software from C-128 mode, *before* booting the CP/M system. After the system has booted, drive 8 becomes drive A, drive 9 becomes drive B, and so on.

Two drives are recommended since CP/M involves a lot of disk and file shuffling. If you do not have two or more drives, don't worry — CP/M on the C-128 is configured to provide a virtual drive available to everyone regardless of the number of physical drives attached to the system. This drive is designated as drive E. The use of drive E won't save you the disk switching, but it will make copying files and running multidisk programs possible.

## Booting the system

Unlike the BASIC operating system in 128 mode, which resides in ROM, the CP/M operating system must be loaded into memory from disk. There are two ways to boot the CP/M system on the C-128.

If you are using a single drive or multiple drives assigned different drive numbers through hardware, you can boot the system by inserting the systems disk in drive 8 with the computer off and the drive on. Once the disk is inserted, turn on the computer and CP/M will autoboot.

If you are using multiple drives that need their drive numbers changed through software, first change the drive numbers from 128 mode. After this has been done, put your systems disk in drive 8 and type **BOOT**. CP/M will boot and take control of the system. If you try to boot CP/M by using the reset button, you will only succeed in resetting the drives.

In both cases, a successful boot is indicated by the CP/M ready prompt, **A>**, and a flashing cursor.

As mentioned, CP/M's primary purpose is to provide status information about your system. One way it does this is through the prompt. The **A>** prompt tells you that your primary drive is drive A. This is the drive that will be accessed unless you tell the system otherwise. If you are using a multiple drive system, you can change the primary drive designation by typing the new primary drive letter followed by a colon at the prompt. To

change the primary drive to drive B, type **B: <RETURN>** at the prompt.

CP/M expects you to type in commands at the prompt. Commands are files that reside on disk with the file type **.COM**. To see a directory of the disk in your primary drive, type **DIR** at the prompt.

## 40 or 80 columns?

When CP/M boots on the C-128, the system expects an 80 column screen as the output device. Consequently, information is formatted for 80 columns. If you are using a 40 column screen, the C-128 allows you to scroll across 80 columns using the grey cursor keys. If you haven't tried this yet, do so. You'll understand why I say that, though it works, it is very annoying.

Unfortunately, most programs for CP/M expect an 80 column screen and format screen information accordingly, so you will have to use the screen scroll function keys. However, if you are using only the CP/M commands and utilities, you can tell the system to format output for a 40 column screen.

To do this, insert the utilities disk containing the **device.com** file in the primary drive, and at the prompt type **device console [page columns = 40] <RETURN>**. Until you change it again all subsequent CP/M output will be formatted for a 40-column screen.

## Copying disks

The first thing I do when I get new disks is make backups. (The reason for this is another story that I'm only just starting to be able to laugh about). There is a reason other than security to make copies of the systems disks provided with the C-128. To ensure compatibility with 1541 disk drives, the two sides of the distributed disks are recorded separately on a single-sided disk drive. To access the second side you must flip the disk over! If you are using a 1571 you will want to copy these two sides onto a double-sided disk so that disk flipping is not necessary.

Unlike other implementations of CP/M, a copy of the systems disk can be made using the **PIP** utility. **Copysys** is not implemented on the C-128. Files with the **.SYS** extension are treated in the same

*Continued overleaf...*

# A hard copy of Help

by M. Garamszeghy

Copyright 1985 Miklos Garamszeghy

CP/M Plus on the C-128 has a comprehensive on-line Help facility that allows you to call up data on the various features of CP/M, and the syntax of most of its commands. Sounds convenient, doesn't it?

Well, it is — except when you need it most: when you're stuck in the middle of a CP/M command or utility and you forget the command syntax. One solution is to exit the program or abort the command (assuming that you haven't forgotten how) and call up the Help routine. A better solution which I have found is to consult a hard copy of the Help file listing. The major portion of the **Help.HLP** file is an ASCII file (remember that CP/M works in true ASCII, not PETSCII) containing the text for the on-screen displays.

When printed out on paper, the 83K byte Help file is over 40 pages of single-spaced text. That is a formidable amount of documentation for an on-line Help file. The file contains some very useful data, not found in the C-128 users' guide, about the various CP/M commands and utility programs. This makes it a very handy reference tool for novice and experienced CP/M programmers alike.

There are several ways to create a hard copy of the **Help.HLP** file. Perhaps the easiest, though least elegant, method is to use the CP/M **PIP** command. When the CP/M prompt (**A>**) is displayed on the screen, type in **pip lst: = help.hlp <RETURN>**.

Some garbage will be printed on the printer. After several pages have been printed, the main text body of the file will start to print. If you don't mind a few pages of garbage first, then this will produce an acceptable printout of the **Help.HLP** file. The garbage consists of an electronic table of contents for the main body of the Help file.

If you are using a non-Commodore-compatible printer through an interface such as the **CARDCO/?**, the interface should be set for transparent mode for a standard ASCII print out. In ad-

dition, the printer can be set to skip over the perforations on fanfold paper, if this feature is implemented on your printer, which produces a nicely formatted output. A Commodore or compatible printer should be set for upper case/lower case mode. However, because the Help file is in standard ASCII, upper-case characters will appear as lower-case, and vice versa. This is a minor inconvenience only.

If you have a 1571 disk drive, the hard copy can be obtained in a slightly more elegant fashion. Although it is not documented in the C-128 System Guide or even in the on-line Help file, the CP/M Plus **Help** command has two very important options: **Extract** and **Create**. These options allow you to convert the **Help.HLP** file into a standard ASCII text file, and then back into the **Help.HLP** file, respectively. To use these options, you must have a disk with at least 172K bytes of space on it (83K for **Help.HLP**, 7K for **Help.COM**, and 82K for the **Help.DAT** file created by the process). Unfortunately, this exceeds the 168K of usable space on a single-sided C-128 CP/M disk.

The procedure is as follows. Format a disk as a double-sided CP/M disk on a 1571 drive. Next, copy the **Help.COM** and **Help.HLP** files from the system disk to the new disk with the command **pip e: = a:help.\* <RETURN>** (for a single drive copy); or **pip b: = a:help.\* <RETURN>** (for a dual drive copy).

After the files have been copied, put the new double-sided disk in the default drive, and type **help [extract] <RETURN>**. **Extract** can be abbreviated to **E**. The drive will go on for several minutes and the message 'Extracting data' will be displayed. The process creates a file named **Help.DAT**. This is a standard ASCII text file containing only the text part of the **Help.HLP** file. The file can then be printed with the command: **pip lst: = help.dat <RETURN>**. Of greater interest, though, is the fact that the file can also be edited with any ASCII text editor, such as **WordStar** (in non-document mode) or the CP/M

utility **Ed.COM**. You can revise the existing material and even add some new stuff of your own!

If you do add new data, each entry must be preceded by a header line in the following format: **///nheading name**, where **n** is the Help level from 1 to 9, and the **heading name** is the entry name for the Help index. The three slashes must begin in the leftmost column. A Help level of one indicates a major topic, while 2 to 9 are used for subtopics. The new material should be arranged in alphabetical order by primary heading and subtopics. It can be inserted between existing topics if the alphabetical ordering so dictates. An example of a set of new entries would be:

```
///1New Topic
Put text here.
///2Any Subtopic
More text.
///3Subsubtopic
More text.
///2BSubtopic
Text for second subtopic.
///1Second new topic
More text.
...and so on.
```

The text content can be anything that can be printed on the screen. To get a feel for the structure of the topic entries, it is best to examine a few of them with a text editor.

Once the file has been edited and any new records added, it is reassembled into the **Help.HLP** file with the command: **help [create] <RETURN>**. **Create** can be abbreviated to **C**.

By using the **Extract** and **Create** options for the Help command you can customize and update your Help file, and even add entries pertaining to your favourite programs. In addition, there is no law that says your Help file must contain data on CP/M commands. You can easily create a custom **Help.HLP** file containing names and addresses, or even recipes, which can be readily searched by a command such as: **help john doe** or **help fudge brownies**. This is in effect a free database program built into CP/M on the C-128! □

way as other files. Before making any copies, you must first format the destination disks. To do this, insert your utilities disk with the **format.com** file on it into the primary drive, and at the prompt type **format <RETURN>**. This will load and initiate the **format** command. Follow the prompts to format as many disks as you like.

When you've returned to the systems prompt, reinsert your systems disk with the **pip.com** file on it and at the prompt type: **pip <RETURN>**. This will load **pip**, the built in file manipulator, into memory, initiate execution and return with the **pip** prompt (\*). You now have to tell **pip** from which disk to copy, the files to be copied, and the disk to copy them to.

If you are using a single drive system, you will have to copy files from the primary drive (which I will assume to be A) to the virtual drive E. At the prompt type **e: = a:.\*.\* <RETURN>**. CP/M will prompt you when it is time to shuffle disks.

If you have more than one drive then you will be copying from the primary drive to another drive (which I will assume to be drive B). At the prompt type **b: = a:.\*.\* <RETURN>**.

Since we want to copy the whole disk we've used the wildcard **\*.\***, which will match every file and cause it to be copied. The first asterisk matches the filename, while the second asterisk matches the file type. As in Commodore DOS, the **?** can be used to match a single letter. After the disk is copied you will be returned to the **\*** prompt. Copy the rest of your disks in the same manner. If you're using a 1571, remember that two single-sided disks can fit onto one double-sided disk.

If you're curious about what you've been doing, one adequate source of information is the systems guide provided with the C-128 computer. A better source is also provided with the C-128 in a less obvious form. Built into the CP/M Plus operating system is a **help** command and extensive Help files. This information can be accessed by inserting the systems disk with the **help.com** and **help.hlp** files on it into your primary drive and typing **help** at the prompt. Follow the prompts to retrieve the information you desire.

While more comprehensive than the C-128 manual, using online Help files can be inconvenient. If you are in the middle of executing a program and discover that you need some relevant information, you must abort the program, shuffle your disks and then call up the necessary Help

file. Wouldn't it be nice if you could have all of this information available whenever you needed it, online or off? Guess what? Built into CP/M are provisions for printing the Help files into a forty page Help manual! For complete instructions, see Mike Garamszeghy's sidebar accompanying this article.

### Where to get software

A large number of commercial CP/M programs are available to fill practically every need. Many of these packages are 'vanilla' programs, designed to run on a variety of CP/M computers. If you are buying prepackaged CP/M programs, there are a couple of rules to remember:

- Make sure the program disk contains an **install** file to configure the program to suit your system.
- MP/M (multiple terminal CP/M) and CP/M-86 (CP/M for 16-bit computers) will not run under CP/M Plus.

Alternative sources for CP/M programs are the various users' groups and public domain libraries. CP/M hackers have refined many of these programs to commercial quality over CP/M's ten-year history. They can usually be obtained for nominal membership or downloading fees. The rules for using commercial programs apply here as well.

The original version of CP/M shipped with the C-128 did not include support for the RS232 port. Consequently, it was impossible to use a modem from within CP/M mode. Downloading CP/M files meant translating files downloaded in 128 or 64 mode. Commodore has since released an upgrade to the BIOS that offers a number of useful enhancements, including full support of the RS232 port.

As well as a 128-mode library, TPUG's C-128 chapter has formed a CP/M mode library. Programs are distributed on single-sided disks recorded using CBM's GCR technique. With the permission of Commodore Canada, the files necessary to upgrade the BIOS are being distributed on the CP/M library's first disk, (Z)AA.

As with other Chapter libraries, the C-128 library will only be as good as you make it. Please donate any *public domain* programs you may have found or written. □

*Adam Herst, in addition to being the co-coordinator for the C-128 chapter, is the TPUG CP/M librarian. He will continue to explore all aspects of CP/M on the C-128 both at the chapter meetings and in future articles.*

## Marketplace

### COMPUTER RENTALS

We buy, trade, sell and repair Commodore computers.

• 1541 alignment	\$ 29.00
• Box of 10 disks	\$ 14.95
• 4040 or 8050 Drive	\$995.00
• 2031 Drive	\$375.00
• Superpet	\$995.00
• Green Monitor	\$ 95.00
• Education software	\$ 3.00

### COMPUTER RENTALS

17 Cedarbank Crescent  
Don Mills, Ontario M3B 3A3  
Tel: (416) 446-1035

## T1 General (Ontario) Income Tax Templates for the C-64

Includes schedules 1 to 10  
Available for \$19.95 plus  
O.S.T. from  
R Link, Box 87, Beaverton,  
Ontario L0K 1A0  
Specify one of  
Calc Result™, Multiplan™,  
or Better Working™.

### Commodore Repairs

C-64 Computer \$69<sup>95</sup>  
1541 Disk Drive \$69<sup>95</sup>

Total price includes pick-up  
and delivery. Dealer dis-  
counts available.

### VOYAGEUR

Box 730 Almonte, Ontario  
K0A 1A0  
Tel. 613-256-2597

# Autobooting on the C-128

by M. Garamszeghy

Copyright © 1986 M. Garamszeghy

The C-128 has the ability to automatically load and run either a machine language or BASIC program from disk at power-up or when the reset button is pressed. In this process, known as 'autobooting', the computer loads the contents of track 1, sector 0 from the disk into memory at address 2186 in bank 0 (part of the BASIC stack area) and checks to see if that sector begins with a certain sequence of bytes. If the byte pattern is found, the machine language instructions in the remainder of the sector are executed. This feature works with both the 1571 and 1541 disk drives, as well as with most other compatible Commodore and non-Commodore drives.

The required byte sequence consists of the three character string "CBM", followed by four zero bytes. This can be followed by an optional message (up to sixteen characters) to be displayed during boot-up. The message is followed by two more zero bytes. This sequence is followed by some machine language instructions and a BASIC command string consisting of either **RUN <filename>** or **BOOT <filename>** for BASIC or machine language programs, respectively. The final byte is another zero. The machine language instructions are:

```
ldx #address low
ldy #address high
jmp $aea5
```

'Address low' and 'address high' are the low and high bytes of the beginning of the **RUN** or **BOOT** string that follows. This address may change depending on the length of the boot message string. The **JMP** instruction transfers execution to a routine in the BASIC ROM.

The program below, **Autoboot Maker**, allows you to write the required byte pattern to make a boot disk. One word of caution, however. Since the program uses the random block-write method to write the boot sector, the boot disk should not be subjected to the DOS **Validate** or BASIC 7.0 **COLLECT** commands. Either of these commands may cause the boot sector to be overwritten, and will in any case cause it to become deallocated. **Autoboot Maker** will warn you if it finds track 1, sector 0 already allocated. In that

case, you can either overwrite the existing sector or try another disk. If the disk already has a boot sector written, the old data will be replaced by the new. In any event, it is perhaps better to write a boot sector on a new disk immediately after formatting it. This ensures that the sector does not have any valid data on it.

Interestingly, the BASIC command string used in the boot sector is not limited to **BOOT <filename>** or **RUN <filename>**, but can be any valid immediate mode BASIC command. By changing lines 70 and 80 of the program to:

```
70 bo$=bo$+"directory"+
chr$(0)
```

for instance, you could have an automatic display of the disk directory on boot-up. Other useful applications include **DLOAD**ing a program (without running it); **BLOAD**ing a machine language utility, sprite data, function key definitions or a custom title screen; or perhaps merely changing the screen colours with a couple of **COLOR** commands. The only apparent restriction on the command string is that its total length must be less than one BASIC line (up to 160 characters) and must contain only one logical line (that is, no **CHR\$(13)**s). All statements must be immediate mode (**INPUT** and **GET** are not allowed).

Of course, most of the above applications (except for the **DLOAD**) can be performed by a **RUN <filename>** type command string. However, while these programs are easily accessible to prying eyes from the disk directory, a person needs a disk sector editor to view or change the contents of a custom boot sector command string.

```
10 bo$="cbm":c0$=chr$(0):fo
ri=1to4:bo$=bo$+c0$:next
:color0,7:color4,7:prin
t" "
20 print"boot message":inpu
t" max 16 chars";me$:me$
=left$(me$,16):sa=2831+
len(me$)
30 input" boot file name";b
f$:input" [b]asic or [m]
ach. lang.";ft$:print" "
40 dclose:print" boot mess
age: "me$:print" boot fi
```

```
le name: "bf$;: ifft$="m
"thenprint" mach. lang.
":elseprint" basic":
50 input" ok [y/n]";o$:ifo
$<"y"then10
60 bo$=bo$+me$+c0$+c0$+chr$
(162)+chr$(255andsa)+chr
$(160)+chr$(sa/256)+chr$
(76)+chr$(165)+chr$(175)
70 ifft$="m"thenbo$=bo$+"bo
ot":elsebo$=bo$+"run"
80 bo$=bo$+chr$(34)+bf$+chr
$(0):open15,8,15:open8,8
,8,"i":ifdsthendclose:go
to110
90 print#15,"u1:":8;0;1;0:p
rint#15,"b-a:":0;1;0:if
ds=65then120
100 print#15,"b-p:":8;0:pri
nt#8,bo$;:print#15,"u2:
":8;0;1;0:dclose: ifds
then110:elseend
110 print" disk error >> ";
ds$:goto40
120 print" boot sector alr
eady in use":input" ov
er-write [y/n]";ow$: if
ow$="y"then100:else40
```

## FREE SHIPPING BY MODEM

### SATELLITE ELECTRONIC SHOPPING MALL

Take advantage of this first FREE electronic shopping mall. Available to all computer and modem users.

Now you can shop for insurance, cars, computers, books, software, travel tickets & holidays, and much more in the comfort of your own home or office. The electronic stores are open 24 hours a day, 7 days a week for your convenience.

Stores in the Mall	Modem
Bayside Computerized	831-0666
Insurance Shopping Data Base	
Compu-Car	831-6678
Shop for new GM, Chrysler, and Ford cars	
Computer Shoppe	831-6907
Order computer hardware	
Alpha Books & Software	831-6908
Order computer books and software	
Electronic Travel Agency	Coming soon
Book flights, holidays & hotels	

#### And More Openings Coming

Check with any of the shops for more mall information.

**LINK UP WITH YOUR  
SATELLITE DEALER TODAY!!**

# A scrounger's guide to CP/M

## *How to beg, borrow and otherwise obtain CP/M software for the C-128*

by M. Garamszeghy

Copyright © 1986 M. Garamszeghy

CP/M opens a whole new world of adventure for both the casual and serious Commodore computer user. Although it is one of the oldest microcomputer operating systems (born *circa* 1975), and has been implemented to a limited degree on other Commodore machines, it is totally new to most owners of the C-128. Over the years, CP/M has become almost the de facto standard for 8-bit computer operating systems, especially business-oriented systems.

There are literally thousands of CP/M computer programs on the market and in public domain libraries. Many of these programs will, in theory, run on the C-128. The problem lies in the fact that practically none of the software is available in Commodore GCR disk format, the recording technique used exclusively by the 1541.

You may not realize it, but there was some CP/M software included with the C-128 computer. If your machine came with only one CP/M disk (marked 'System Disk') like mine did, turn it over! Even if you have one of the new double-sided 1571 drives, you must still turn the disk over to access the second side. The flip side (marked 'User Utilities Disk') contains most of the utilities referred to in the manual — **Ed.COM**, **Date.COM**, **Submit.COM** and other CP/M favourites. (Because I generally do not use 'flippies', it took me three months before I noticed this, quite by accident.)

Another valuable source of CP/M software and technical information is the mail-away card packed in the *C-128 System Guide* manual. If you are serious about CP/M programming, the two-inch thick manual and the three disks (one floppy and one single-sided) that you get for your 20 bucks (30 Canadian) are well worth the money. The manual contains

very detailed instructions on the CP/M resident commands, as well as the transient utilities. For the machine language programmers and hardware hackers, it also contains detailed information on the various BIOS and BDOS function calls, complete with examples of how to use them. The disks contain a whole slew of machine language-oriented utilities, such as the two main CP/M assemblers **Mac.COM** and **RMac.COM**, and the disassembler **Sid.COM**. Also included are demonstration assembly language programs and library files.

---

**... Commercial CP/M software is available in many different disk formats. Unfortunately, Commodore GCR is usually not one of them...**

---

The next major source of CP/M software is the public domain libraries. (Although their stocks are usually limited, there are a few public domain libraries that cater to Commodore CP/M users at modest cost. One that comes to mind is Poseidon Electronics, 103 Waverly Place, New York, New York, 10011.) Most of these libraries operate bulletin boards (remote CP/M — RCP/Ms, in the popular lingo), where the software can be uploaded or downloaded for a nominal charge. This, of course, assumes that you have a modem and a suitable terminal program for your 128 (usually a form of X-Modem, otherwise known as MODEM7, which is the standard CP/M program file transfer protocol).

Commercial CP/M software is available in many different disk formats. Unfortunately, Commodore GCR is usually not one of them. The problem of disk format

incompatibility was reduced with the introduction of the 1571 disk drive, which can read most of the popular CP/M disk formats (except Apple CP/M, which is perhaps the most common). However, that is no help to people like myself, who do not have a 1571 drive. Despite this, I now have a fairly extensive library of CP/M software (including such favourites as **WordStar**, **CalcStar**, **dBase II**, **MBASIC** and **Fortran-80**) on 1541 format disks, thanks to a bit of electronic jiggery-pokery. The trick lies in first obtaining the software in a format readable by a 1571 drive, then using the genius of the 1571 to translate for you. The catch is, of course, that you must have temporary access to a 1571. I used one in a local computer store, the manager of which didn't seem to mind me playing with his demo equipment for a couple of hours.

The 1571 is truly a remarkable machine. Using the CP/M **PIP.COM** utility ('PIP' stands for **Peripheral Interchange Program** — I never could figure out why they just didn't call it **Copy**), you can copy from a foreign disk format to 1541 GCR format with a single drive. The procedure is as follows. First format some blank disks as C-128 single-sided disks with the CP/M **Format** command. These are equivalent to 1541 format CP/M disks. Next, with the CP/M system disk (the boot disk that comes with the computer) in the drive, type **PIP <RETURN>**. When **PIP** is loaded and ready to run, it will respond with the prompt **\***.

Now you can copy the program files. Remove the system disk and replace it with the foreign disk containing the files you wish to copy. To copy an entire disk, enter the command **e:=a:.\* <RETURN>**. To copy a single file, use **e:=a:<filename.ext> <RETURN>**, where **filename.ext** is the filename and extension of the file you

wish to copy. The 1571 will buzz and click for a few moments. A reverse video message will appear at the bottom of the screen listing the suspected format of the foreign disk. The message 'IBM-SS', for example, indicates a single-sided IBM CP/M-86 format disk. If you know this to be the correct format, press **RETURN**. Do not use the **CTRL-RETURN** key combination to lock in the disk format.

The 1571 will log drive **a:** as that format. If you know for sure that the proposed format is incorrect, or if you wish to see what other choices you have for that disk format (some disk formats are similar enough that the 1571 cannot automatically distinguish between them), use the cursor-left and cursor-right keys to scroll through the choices. When you find the correct one, press **RETURN**. Single-drive copying is then performed by following the prompts on the screen for removing and inserting the disks at the correct times. Press **RETURN** to restart the copying each time you insert a disk. Disk **a:** is the foreign source disk, and disk **e:** is the C-128 destination disk. It may be a good idea to cover the write-protect notch on the source disk to prevent disaster from striking if you accidentally mix up the two disks. Drive **e:** is a 'phantom', or virtual, disk drive. It does not really exist except in the memory of the C-128. This concept is essential for single-drive copying.

If you are using two drives, **a:** and **b:** (either two 1571s — one set to device number 8, and the other set to device 9 via the DIP switches at the back — or a 1571 set to device 8 and a 1541 set to device 9 via the hardware method), you can use the command **b: = a:.\* <RETURN>** or **b: = a: <filename.ext> <RETURN>**, where the source disk is in drive **a:** and the destination disk is in drive **b:**. With two drives it is not necessary to swap disks during the copying process. A similar procedure to that described above can also be used for copying files from C-128 format to a foreign format, and between two foreign formats.

Once the programs have been transferred to 1541 format, the easiest way to back up your software with a 1541 is to start up the C-128 in C-64 mode, and use one of the many '3 minute' disk copying programs available. This, of course, can only be used to copy whole disks. You will still need the CP/M **PIP** command to copy individual files.

If you own a 1571 drive, you can purchase CP/M software in a wide variety of popular disk formats. Of course, these

will be 5 1/4-inch disks. Much CP/M software, including all of the stuff I started with, is available on 8-inch disks. (This was how all CP/M software was originally used.) How, do you ask, did I get these 8-inch disks into the 1571 drive to copy them? The answer is, obviously, I didn't. What I did use was a bit of creative hacking.

My software started life on 8-inch CP/M disks used on an ancient (circa 1981!) TRS-80 Model II computer. The programs were transferred via a null modem cable to a nearby IBM-PC. (For the benefit of techies, I used an X-Modem program on the TRS, and **CrossTalk** in X-Modem mode on the PC.) The result was a set of CP/M files on 5 1/4-inch disks, but in PC-DOS format. Despite all its talents, the 1571 cannot decode the directory data on PC-DOS disks. (CP/M directories are of the same structure in any disk format and are very simple to decipher). To overcome this, I wrote a short hybrid machine language-BASIC program for the IBM-PC which converts PC-DOS files into the CP/M-86 files that can be read by the 1571, thus readying them for translation into 1541 format. The point to this paragraph is not to show a roundabout method to get CP/M software for the C-128, but to emphasize that CP/M software from one machine and disk format can be easily made (relatively speaking) to run with a totally different machine and disk format.

With some CP/M software, it is necessary to 'configure' the program for the specific hardware features of your system. Popular CP/M software, such as **WordStar**, often comes with an installation program allowing you to set up the software for running on a wide variety of computer systems. Unfortunately, I very much doubt that you will find a configuration program that has the C-128 in its list of supported systems. This may change as the C-128 gains popularity. However, the engineers at Commodore had the good sense to make the C-128 emulate one of the more popular computer terminals, the Lear Siegler ADM-31. (It also emulates the less sophisticated ADM-3A.) The LS-ADM-31 is found on nearly all CP/M configuration program menus. Thus, almost all CP/M software can be configured to run on the C-128.

Incidentally, this article was written with **WordStar** configured for the ADM-31, using a 1541 disk drive. Not the fastest word processor combination around (for obvious reasons), but still quite manageable. □

## Missing Letter Puzzle

- 1) Insert your COMAL disk in drive\*.
- 2) Type **LOAD "C64 COMAL\*\*",8**
- 3) Type **RUN** (starts COMAL)
- 4) Type **AUTO** (turn on auto line#'s)
- 5) Enter the program lines shown below (COMAL indents lines for you)
- 6) Hit **RETURN** key twice when done
- 7) Type **RUN**  
Pr-gr-mm-r's P-r-d-s- P-ck-g-  
Programmer's Par?

```
0010 dim text$ of 39, disk$ of 2
0020 open file 2,"missing.dat",read
0030 disk$:=status$: count:=0
0040 if disk$="00" then
0050   count'text
0060 else
0070   close // no data file found
0080   create'text
0090 endif
0100 play'game
0110 //
0120 proc count'text
0130   while not eof(2) do
0140     read file 2: text$
0150     count:+1
0160   endwhile
0170   close
0180 endproc count'text
0190 //
0200 proc create'text
0210   open file 2,"missing.dat",write
0220   print "input text (or blank):"
0230   repeat
0240     input text$
0250     if text$>"" then
0260       write file 2: text$
0270       count:+1
0280     endif
0290   until text$=""
0300   close
0310 endproc create'text
0320 //
0330 proc play'game
0340   open file 2,"missing.dat",read
0350   for x:=1 to rnd(1,count) do
0360     read file 2: text$
0370   endfor x
0380   close
0390   for letter:=1 to len(text$) do
0400     if text$(letter) in "aeiou" then
0410       print "-",
0420     else
0430       print text$(letter),
0440     endif
0450   endfor letter
0460   print
0470   for letter:=1 to len(text$) do
0480     while key$<>text$(letter) do
0490       print "?"+chr$(157), //left
0500     endwhile
0510     print text$(letter),
0520   endfor letter
0530 endproc play'game
```

\* If you don't have COMAL yet, order a **Programmer's Paradise Package—\$19.95**. It includes the complete COMAL system plus over 400 pages of information. Add \$5 more to get our 20 interactive lesson Tutorial Disk. Add \$2 shipping. Visa/MC or US funds check accepted. Send to:

**COMAL Users Group USA**  
6041 Monona Drive, Room 109  
Madison, WI 53716  
phone 608-222-4432

# C-128 book flawed but useful

## Commodore 128 Internals

by K. Gerits,  
J. Schieb and F. Thrun  
from Abacus Software

### Review by Jim Butterfield

Copyright © 1986 Jim Butterfield. Permission to reprint is hereby granted, provided this notice is included in the reprinted material.

This book contains a great deal of information about the Commodore 128 that is not yet generally available. It contains a rich fund of technical data on the 128, together with comments on the C-64 mode as it fits together with the machine. The information isn't complete (perhaps it's too early to hope for that yet). There are some inaccuracies. And the material is heavily technical... it's not suitable for a beginning level programmer or user. Considerable machine language experience is called for in working through the book.

### Hacker's heaven

The book has a heavy hardware/hacker/machine language content. You know that if a chip is mentioned, you'll get detailed pinout diagrams and register descriptions. If an interface is described, you will get a picture of the edge connector and possibly information on voltage levels. Good for service departments, but probably overkill for the programmer and user who would never think of diving in with a soldering iron. This kind of detail is featured prominently, and takes up a fair amount of space in the book; the reader who doesn't need it will end up browsing through to areas of more interest.

The book also spends a good deal of time and space on 'known' hardware and techniques. Even though the 40-column chip has only two new registers (at \$D02F and \$D030), the whole chip is

discussed at length in Chapter 2. You'll get the story on sprites, split screens, character sets, screen colours, high-resolution, multicolour and extended colour modes. None of this is new, and in this area the book cautions, "all of the following programs must be entered in the 64 mode". The two new registers are not discussed at this point. You'll find similar detailed discussions of 'known' chips in Chapters 3 and 4, which deal with the CIAs and with the SID chip.

If you're moving up from the Commodore 64, you might feel that you know all this stuff, and that you'd rather see the book focus specifically on new material. On the other hand, you might be pleased to see the book attempting to cover the entire range of the architecture.

Chapters 5 and 6 do break considerable new ground. The VDC — the Visual Display Chip that drives the 80-column display — is outlined in detail and several of its idiosyncrasies are discussed. And the MMU — the Memory Management Unit that controls the architecture of the computer — is also outlined. Until the Commodore Programmer's Reference Guide becomes available, this book may be one of the best sources of information on these two chips.

Chapter 7 is titled 'Assembly Language Programming'. It largely discusses the Kernal routines, and how to create a 'boot' or autostart disk. It is the last 'narrative' chapter of the book... and yet the book is less than half finished.

Chapter 8 is the bulkiest part of the book. It contains a detailed disassembly of Kernal ROM, with each line commented. I miss the BASIC ROMs, but they are 32K in size, and including them would have more than doubled the size of this book, which is already massive. To read such ROM listings, a user needs to have some expertise. The pieces of code fit together jigsaw fashion, and to understand one part the reader will need to know how other parts interrelate. The disassembly code contains some 'European' ROM programming that is not found in North American models of the Commodore 128.

### The verdict

The book is relatively good, but is not completely free of errors. Part of the disassembly on page 387 is wrongly per-

formed (instructions at F934, F936 and F94B are not shown correctly). The memory configuration table on page 145 has copied a typographical error from some early Commodore documentation; configuration 13 should reference RAM 0, not RAM 1 as shown.

This book is not an easy read. There's a lot of good information here, but you'll have to dig for it. There's an index, but it's of limited use. The writing style is often stuffy ("Those who think it would be boring to take a close look at this chip would be deceiving themselves.")

But there's good stuff in here. The book is out early, and contains a great deal of data not easily available from other sources. For the serious student — and you'd better be a serious student — it's a goldmine of good information.

*Commodore 128 Internals, Abacus Software, PO Box 7211, Grand Rapids, Michigan 49510.* □

**NOW GET TRUE DESCENDERS  
FOR 801, 1525, 803, GP-100,  
HUSH 80 & SIMILAR PRINTERS**

*From the Descender ROM People*



**FEATURING:**

- True descenders
- A pleasing alternative
- Uniform character formation
- No change in graphic capability
- No change in software compatibility
- No change in printer operation
- North American standard font
- Complete instructions
- Easy installation
- No soldering

**\$39.95 Cdn      \$29.95 U.S.**

Ont. Residents add 7% Prov. Sales Tax  
Cheque, Money Order, Visa or MasterCard



**WILANTA ARTS**

6943 Barrisdale Dr.  
Mississauga Ont.  
L5N 2H5



**1-416-858-9298**

*Get a Wilanta Descender ROM Today  
and make your old nightmare  
Just a font memory of yesterday!*

# ATTENTION

## ALL COMMODORE 64, VIC 20, COMMODORE 16, AND PET OWNERS

A complete self-tutoring BASIC programming course is now available. This course starts with turning your computer on, to programming just about anything you want! This course is currently used in both High School and Adult Evening Education classes and has also formed the basis of teacher literacy programs. Written by a teacher, who after having taught the course several times, has put together one of the finest programming courses available today. This complete 13 lesson course of over 220 pages is now available for the COMMODORE 64, VIC 20, COMMODORE 16 and PET computers and takes you step by step through a discovery approach to programming and you can do it all in your leisure time! The lessons are filled with examples and easy to understand explanations as well as many programs for you to make up. At the end of each lesson is a test of the information presented. Furthermore, ALL answers are supplied to all the questions and programs, including the answers to the tests. Follow this course step by step, lesson by lesson, and turn yourself into a real programmer! You won't be disappointed!

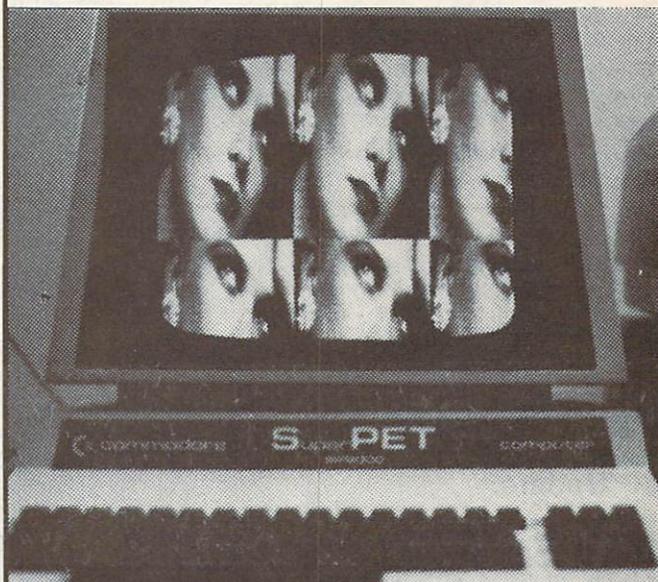
We will send this COMPLETE course to you at once for just \$19.95 plus \$3.00 for shipping and handling (U.S. residents, please pay in U.S. funds). If you are not COMPLETELY satisfied, then simply return the course within 10 days of receipt for a FULL refund.

**FOLLOW UP COURSE** Now available! a 200 page course exclusively on sequential and relative files using a unique approach for those with very limited file programming experience - set up your own personal and business records! — disk drive a must — same author — same guarantee — same cost— this course for all computers except Vic 20.

Fill in the coupon or send a facsimile.

NAME: _____		TP
ADDRESS: _____		
CITY: _____		
PROV./STATE: _____		
POSTAL/ZIP CODE: _____		
I desire the BASIC programming course for:	Any complete course: \$19.95	
Commodore 64 <input type="checkbox"/> Vic <input type="checkbox"/>	Postage and handling: \$3.00	
Pet <input type="checkbox"/> Commodore 16 <input type="checkbox"/>	Total: \$22.95	
I desire the follow up course on relative and sequential files (for all above computers but Vic 20).	Send Cheque or Money Order to:	
	Brantford Educational Services	
	6 Pioneer Place,	
	Brantford, Ontario,	
	Canada N3R 7G7	

# INTRODUCING THE HRT SUPER-RES GRAPHICS BOARD



This is a graphics board which is universally compatible with all models of PET's. With Superpets (both 2 and 3 boards) It works equally well on both the 6502 and 6809 side as well as with OS-9. The graphics board is the result of two years of R&D which began in Nov. 83 with the first prototype. Then starting in Mar. 84 it was test marketed for four months through TPUG Magazine. The original model only worked on the 2001 with a resolution of 320 x 200. At the 84 TPUG Conference the 4016 and 4032 models were unveiled. Then in Dec. 84 at the Superpet chapter meeting the 8032, 8096 and 9000 models were introduced. Over the last year these different models were field tested in schools, universities, business and homes. The HRT SUPER-RES GRAPHICS BOARD was then designed with the results from these tests and input from expert users.

The window size is 700 hor. by 300 ver. for the 8032's, 8096's and 9000's and 640 x 200 for all other models. However the card has an even larger resolution of 1024 x 512 pixels. The most that can be displayed on the monitor is determined by the window size but the card supports scrolling left, right, down and up to view the entire graphics area. This window can be defined to any size, can be located anywhere on the monitor and can scroll about the 1024 x 512 pixels. The graphics can even be overlaid on top of the text without interfering (even if both text and graphics are scrolling).

To accomplish this high resolution the board has its own 64K of ram. Another attractive feature is that this extra memory can alternatively be used as a ram disk.

The above picture has a window size of 700 x 300 and can be scrolled about to reveal the entire 1024 x 512 pixels. The picture of the girl is 280 x 192 and only one image can be shown on an Apple II Computer. (the original source)

A large software package is included with the purchase of the super-res board. It includes plotting utilities, sketching routines, turtle graphics, algebraic graphs, animation routines etc.

Ease of installation is one of the important design features. The circuit board simply plugs into the mother board under the monitor with no soldering required.

Please send cheque or M.O. for \$200.00 U.S./\$253.00 Cdn. plus \$5 shipping and handling (Ontario residents please add 7% Prov. Sales Tax) To HIGH RES TECHNOLOGIES, 16 ENGLISHIVYWAY, TORONTO, ONTARIO M2H 3M4. Note: Please specify computer and disk drive model numbers.

(416) 497-6493

# Amiga Dispatches

by Tim Grantham

On a sea of wild rumours about CBM's imminent demise, the Amiga has sailed on, propelled by waves of new software, buoyed by continuing excitement over its extraordinary capabilities. No other computer has had so much software available so soon after its debut. Yet it's not an easy computer to market — there's just nothing else like it. Of the one million (!) computers Commodore sold in the last quarter of 1985, only 10,000 were Amigas. But chin up, Amigos! We believe that given time, the Amiga will boldly go where no computer has gone before. Meanwhile, more news from the frontiers.

## CP Update

The controversy over copy protecting software for a machine designed from the bottom up for multitasking rages on. Electronic Arts has continued to press for innovative approaches. For a time, they were considering a kind of optical dongle used in England: a piece of coloured plastic that the user looks through when booting up the program. Cheap to manufacture (but difficult to synthesize at home), they enable the user to read a series of encrypted characters on the boot-up screen. Once these are entered, the program continues.

It's not only end-users who are fighting EA on the CP issue — witness the following from one software developer:

#: 6188 S1/Software  
25-Jan-86 02:53:39  
Sb: #6132-electronic arts  
Fm: ben blish 72667,1341  
To: Electronic Arts 76004,237

... The package my company is creating is priced in the \$1000.00 range... Eminently Stealable... and it is NOT copy protected. If it is stolen, among other things, I will lose my home, and probably my wife. However, human nature is generally good, and I will take the chance, and I will not take the chance of aggravating my bona fide customers. Anyway, I am not a thief, lots of others aren't either, and YOU can afford to take the chance. Please do so. And that's my Opinion. — Ben Blish, SoftCircuits Inc; 305-721-2707 anytime —

EA responded thusly:

#: 7244 S1/Software  
04-Feb-86 19:32:56

Sb: #EA Copy Protection  
Fm: Electronic Arts 76004,237  
To: all

Okay, I have a proposal: What is your opinion if we: a) continue to sell our utilities protected, although the protection is of the type that lets you make backups. When booting, it requests that you insert the master disk, to verify the signature. However, b) enclosed in the package is a card that lets you, with proof of purchase, buy a completely unprotected version for say \$10-\$20. I think that most users will not be upset at just using the first version, but for those who really need it, the unprotected version will be available. To purchase the unprotected version, we will probably have on the card some sort of site license/non-piracy agreement to sign, and/or we might encrypt (but not display) your name, address, etc. in the disk we send you. Again, this is just a proposal, so I want your feedback. Thanks.

Jeff J. — EA

Response to this has been generally favourable. EA is to be commended on their continuing dialogue in this matter. Drop a line to EA if you have any constructive ideas on this very important subject.

As a final note, one company already has a copy program in beta-test that has proved capable of backing-up Deluxe Paint, and all other EA software.

## Software Bulletins

These just in... Micro-Systems Software continues to lead the way in productivity software with its **Online!** terminal software, for which two (free) upgrades have already been issued. File transfers with CompuServe's B protocol are included in the latest version (v1.26). **Analyze!**, a spreadsheet, was released in January, and **Scribble!**, a word processor, is currently in beta-test and should be available shortly... By the time you read this, TDI software will have released its **Modula-2** compiler (not copy-protected). Borland, famous for its **Turbo Pascal**, should have the Amiga version ready by the middle of this year... Aegis' **Pro Draw** CAD software will share **AutoCAD**'s program-mability. Prices on latest Aegis software are as follows: **Animator/Images** combo. \$139.95, **Images** \$79.95, **Impact** \$199.95, **Draw** \$199.95 (all prices US)...

Michael Reichmann of Batteries

Included has announced that BI are developing a version of the **Degas** drawing software, considerably enhanced from the Atari ST version. The author is Tom Hudson. Availability — some time this summer. BI has also announced that no forthcoming Amiga programs from them will use any CP... Paul Higginbottom, Amiga Product Manager, and well-known to TPUG members, has announced the imminent arrival of **AmigaTERM**, C-A's own terminal software... Ben Blish, of Soft Systems, has sent out copies of his **PCLO** printed circuit board CAD program, which makes full use of Amiga's graphics, mouse, and multitasking environment, for beta-test. He has promised that there will be no copy-protection on it...

Tom Casey, co-author of **Deluxe Video Construction Set**, appeared on the Amigaforum on CompuServe to explain the delay on release of the program. It is currently in beta-test, but he wants to get all the kinks out of this extremely powerful and innovative software. With fingers crossed, he hopes it will be ready in March. **DVCS** can import any image, sound or music in the IFF format jointly developed by EA and C-A... The new ROM Kernel 1.1 manuals are being shipped to developers and include the necessary info on the printer drivers, so all the people crying out for a way to get their printer into Preferences shall soon be quieted... Bob Perez has handed over the porting of his **VMCO** terminal program for the Mac to the legendary Captain Crunch, aka John Draper...

Ashton-Tate is soliciting requests for an Amiga version of **dBaseIII**. If you would like to see this happen, leave Email to the Sysop of the Ashton-Tate forum on CIS, Steve Silverwood [76703,3035]... C-A is rumoured to be preparing a disk full of PD software, including Andrew Finkle's version of **MicroEMACS**... Finally, there are apparently different versions of the IBM emulator in beta-test, resulting in conflicting reports of its ability to run various MS-DOS programs. Some can get Borland's **Sidekick** program to run, while others get only crashes. One person claims to have been able to run CP/M software on the Amiga, using an MS-DOS CP/M emulator running under the IBM emulator!

## Hardware News

Unlike those of us in the frozen north,

user groups in the US are getting demonstrations of new peripherals. Larry Miller of FAUG (First Amiga Users Group) gave a lengthy report on CIS of a demo of a frame grabber called **LIVE**, from A-Squared Systems. The unit plugs into the bus expander on the right side of the computer and, in combination with the software, digitizes a standard composite colour video signal into RAM in IFF format (so the **Deluxe** series can use it), for about 300 dollars. It should be available in the second quarter, providing the FCC doesn't hang it up. It does tie up the computer so that no other tasks can be performed while it is in operation, and it is also a 'bus end': it does not pass on the other signals on the bus. This means you would not be able to use it in combination with other 'bus end' units, such as the hardware IBM emulator is rumoured to be. A-Squared is working on a 'Professional' model (at much higher cost) for those who want to process Land-Sat photos, for infrared and X-ray image analysis, and so on.

While we are on the subject of video, the Genlock interface is ready for release. William Volk, of Aegis Development told me that he had used a production (not a prototype) unit to overlay a rotating 3-D NBC logo created with **Aegis Draw** on a television picture from a VCR, on the Amiga's monitor.

Two reps from Tecmar gave a short demo of their Hard Disk and Expansion Module at the Cleveland Amiga Users Group. Apparently, only FCC approval is required, though Tecmar says there is no way they can meet the current demand; they just can't build them fast enough. Microforge is also producing hard disks in various capacities, but Paul Higginbottom reports that both Microforge and Tecmar's products make the mistake of stopping other processes when they access the drive. Carl Sassenrath, author of the **EXEC** on the Amiga, has assured everyone that this is *not* the fault of the Amiga OS.

A-Time is a clock-card that plugs into the parallel port. It has a parallel port on the back, and sells for \$49.95. It's available from Akron Systems Development, P.O. Box 6408, Beaumont, TX 77705.

It looks like CBM's encounter with Andy Warhol has not proved too rich for their blood. They are continuing their promotion of the Amiga as a tool for artists by announcing their sponsorship of the *Amiga Studio Theatre* in the Canadian Pavilion at Expo 86. CBM will provide 20 Amiga computers for use by performing artists across Canada, to create

new works or to integrate special effects into their performances.

Some users are exploring the Amiga's multi-user capabilities. Ron Williams has reported that after entering **newcli ser:** at the CLI, a message was output to the serial port saying **new cli**, and a prompt appeared. But it did not seem to be active because no corresponding window appeared on the Amiga's monitor. John Eric Hinkley cleared up that little mystery by saying that the Amiga serial port does not appear to respond to delimiters like **CR**. It seems you have to send 200 characters to get the serial port to give the line to the **new cli**. He wrote a small 9600-baud terminal program for his IBM PC that buffered what was entered, and when **CR** was pressed, it sent the line padded with extra spaces to bring it up to 200 bytes. He was able to get the machine to respond to such commands as **dir**, **list**, **info**, and so on. Running programs from the serial port proved problematic because they 'attach' themselves to the Amiga's keyboard, and their input/output cannot be reset for the serial port without encountering an error stating that **ser:** has already been attached by another process. I'm sure we haven't heard the last word on this subject, though.

## Other News

A new Amiga magazine has debuted called *Amazing Computers*. Word is that it is packed entirely with Amiga-specific articles and that it is off to a good start. Subscriptions are \$24.00 (in the US) and can be had by writing to PIM Publications, P.O. Box 869, Fall River, MA 02722, or calling (617) 679-3109.

Charles Brannon, the **COMPUTE!** Amiga columnist and editor, has a regular presence on the Amigaforum. You can send ideas and comments to him care of PPN 76004,73 on CIS.

Lastly, the following message should give you some idea of the changes in CBM's attitude towards its customers:

```
#: 7214 S9/SoftwareDevelopment
   04-Feb-86 08:40:47
Sb: REV. 1.1 MANUALS
Fm: FRANK SCHWAB 73137,3142
To: BELA 76703,3015
```

Bela — Took your advice and called Commodore Software Support today. Lisa there told me that yes, even we non-developer scum who bought only the developers' manuals set, will get the revision 1.1 manuals. Who at Commodore is responsible for this outrageous behavior, supporting users after they buy their machine, and providing software and manual updates? Absolutely unheard of! BUT I LIKE IT!!!! □

## The Walker

- 1) Insert your COMAL disk in drive\*.
- 2) Type **LOAD "C64 COMAL\*",8**
- 3) Type **RUN** (starts COMAL)
- 4) Type **AUTO**  
(COMAL provides the line numbers)
- 5) Enter the program lines shown below  
(COMAL indents lines for you)
- 6) Hit **RETURN** key twice when done
- 7) Type **RUN**  
Watch an animated sprite hobble across the screen. Change the (99) in line 450 for really fast walking

```
0010 setup
0020 repeat
0030 walking
0040 until key$="q" //Q to Quit
0050 //
0060 proc setup
0070 blue:=14; pink:=10
0080 white:=1; black:=0
0090 define'images
0100 repeat
0110 input "speed (1-10): ": speed
0120 until speed>=1 and speed<=10
0130 background black
0140 setgraphic 0
0150 spriteback blue,pink
0160 spritcolor 1,white
0170 spritesize 1,false,false
0180 plottext 1,1,"press q to quit"
0190 endproc setup
0200 //
0210 proc define'images closed
0220 dim shape$ of 64, c$ of 1
0230 shape$(1:64):=""
0240 shape$(64):=chr$(1)//multicolor
0250 c$:=chr$(0)
0260 for x=22 to 63 do shape$(x):=c$
0270 c$:=chr$(170)
0280 for x=1 to 21 do shape$(x):=c$
0290 define 0,shape$
0300 c$:=chr$(20)
0310 for x=22 to 42 do shape$(x):=c$
0320 define 1,shape$
0330 define 3,shape$
0340 c$:=chr$(60)
0350 for x=43 to 63 do shape$(x):=c$
0360 define 2,shape$
0370 endproc define'images
0380 //
0390 proc walking
0400 for walk:=1 to 319 div speed do
0410 x:=walk*speed
0420 y:=100+walk mod 4
0430 spritepos 1,x,y
0440 identify 1,walk mod 4
0450 pause(99)
0460 endfor walk
0470 endproc walking
0480 //
0490 proc pause(delay) closed
0500 for wait:=1 to delay do null
0510 endproc pause
```

\* If you don't have COMAL yet, order a **Programmer's Paradise Package**—\$19.95. It includes the complete COMAL system plus over 400 pages of information. Add \$5 more to get our 20 interactive lesson Tutorial Disk. Add \$2 shipping. Visa/MC or US funds check accepted. Send to:

**COMAL Users Group USA**  
6041 Monona Drive, Room 109  
Madison, WI 53716  
phone 608-222-4432

# Extending the cruise

## More hardware and software for the musical Commodore 64

by Tim Grantham

Since my last article about music software and hardware for the 64 ('Cruising the Sea of 64 Music', November 1985), I have had the opportunity to examine some new products, and to take a closer look at some mentioned only briefly in that article.

### Melodian keyboard

Melodian is primarily a manufacturer of keyboards, and their expertise shows. The Melodian Musical Keyboard looks good and feels good, the keys having quite the best action of all the add-on keyboards for the C-64 that I've tried. It also has a three-and-a-half octave range, which means much less switching of octaves from within the program. The keyboard has two connectors to the C-64: a large plug that is inserted into the user port, and a smaller one that plugs into the rear joystick port.

The same straight-forward craftsmanship has been put into the accompanying software. The **ConcertMaster** program is completely menu-driven, and essentially emulates a three-track tape deck and polyphonic synthesizer. Each voice of the SID chip is assigned its own track, and each track can be recorded to, played back, monitored, muted, and so on, completely independently. The usual procedure is to record one track first, then record a second while listening to the first play back. A third track is recorded in the same fashion. This is deliberately reminiscent of the way a piece is assembled at a professional recording studio, track by track, until all sixteen or thirty-two tracks have been recorded. You can also create your own sounds by changing any of the instruments provided by the program. The major features of the SID chip are all under your control, including the filters, synchronization, and ring modulation.

I found the keyboard and software combination ideal for musical 'sketching'. Anything longer becomes a problem, because of the lack of editing functions.

There is no way to change just one or two notes in what you've previously recorded. But if you have enough digital dexterity (no pun intended), it's very easy to lay down a melodic line, then experiment with different harmonies while listening to the melody. Once I'm happy with my three-part arrangement, I store it to disk for later transcription to a full-featured music editor like **Music Assembler 64** or **Master Composer**.

**Concert Master** has a colourful, though simplified, score display that scrolls rapidly from right to left as the music is played. It's not of much use except to reinforce the relationship between the pitch of the notes and their position on the musical staff.

Melodian also has an instruction program called **Rhythm Composer**, that makes a game out of learning musical notation. The player is required to match a musical phrase played by the computer. The program is a little *too* precise, perhaps: a novice is likely to be frustrated with the rigidity of the timing. Human beings are not metronomes, and time values in music are not always exact. Nonetheless, this program can be valuable simply for the discipline it enforces.

Melodian has also released *Programmer's Tool Kit*, a book containing BASIC source listings of programs that read the musical keyboard and read and create music files. Another recent addition is the Melodian **ScoreMaster**, which prints out **ConcertMaster** files in musical notation. Any printer that can emulate the 1525, 1515, or the MPS 803 in graphics mode can be used.

### MacMusic

A fine program for the beginner is Passport Design's **MacMusic**, so named because it emulates the user interface of the Apple Macintosh, complete with drop-down menus and icons. You can do everything by just moving the pointer with the joystick to the appropriate spot on the screen and pressing the fire button. It's a very smooth implementation,

and the only frustration I encountered was with the slow speed of the pointer, as I moved it around the screen.

You literally draw the music onto the screen (the pointer becomes a paintbrush). Each voice is drawn as a line in a different colour. The length of the line determines the duration of the note, and its vertical position on the screen defines its pitch. You can choose from a number of fixed sounds, and several different musical scales. You can have a lot of fun drawing in a blues scale, or the pentatonic scale, because 'wrong' notes are less apparent in these scales.

Two other solid features of this program are the ability to change either the meter (number of beats per measure) or the tempo for any or all measures in the music, allowing complex rhythms to be played. You can also change the instrument being played in any bar, but there is no facility for creating your own sounds.

This program is excellent if you like a 'sandbox' approach to creating music. You can paint great swathes of polyphonic sound quickly and easily, with over a dozen different timbres. When you want to start entering individual notes, carefully timed and placed, however, the approach becomes awkward and time-consuming. A display of a piano keyboard runs up one side of the composing screen, so you know exactly where to draw the note. But it is no faster than it would be to write out what you want in musical notation. This is mitigated by the fact that, once the music has been entered, it is easy to change. You can edit your music note by note, or you can use the cut-and-paste functions to move around chunks of the score. This is especially useful when you want to repeat sections of the music. Once they have been cut out, they can be copied as many times as necessary. The immediate playback that the computer offers, combined with the ease of editing, make this a worthwhile program in many respects. However, I think that someone who already has musical training might prefer a program

with more a more utilitarian approach.

Passport Design's also publishes **Computer Hitware**, arrangements of popular contemporary music on disk for the C-64. Current disks include selections by the Police, Van Halen, Duran Duran, and others. Also available is **Computer Sheet Music**, which teaches keyboard technique and musical notation by displaying songs on the screen in large-note format. The program works in conjunction with Passport Design's MUSIC 64 keyboard, or any MIDI keyboard and interface.

## FastFingers

In **FastFingers**, author Dan Laskowski has taken an approach entirely different from any I've seen before. He has utilized all the resources of the SID chip to create a performance lead synthesizer and sequencer. By combining all three voices into one, he has created 256 remarkable sounds — 'presets' in synthesist's parlance. You can also create your own sounds, using the programs extensive control functions and built-in special effects, by overwriting the presets. These new presets can be saved to another disk.

The list of special sound effects is impressive. It includes:

- **Flange** detunes each of the oscillators with respect to the others.
- **Glide** causes the pitch of each note to slide to the pitch of the next one.
- **Wah** adds a wah-wah effect to the notes.
- **Tremelo** causes a rapid oscillation in the filter cut-off frequency, giving the impression of a rapidly-varying loudness.
- **Weow** makes an upward or downward slew of the pitch as the key is released.
- **Shftr** results in a phase shift of the sound by detuning only one of the voices.

The sequencer in **FastFingers** has also been modelled on those used by professionals. These work very much like super-efficient tape recorders, but without any tape. To use the sequencer in **FastFingers**, you press the **Rcord** key, after setting the speed, and play the sequence. Press **Stop** when done. Now you can **Play** it (backwards or forwards), speed it up or down, punch in (change individual notes), **Cut**, **Splice**, and so on. You can also record arpeggios (series of notes taken from a single chord) that can be triggered from the keyboard, in much the same fashion. Laskowski claims you can store up to *forty minutes* worth of music in the sequencer.

Yet another unusual aspect of the program is that it uses the piano keyboard

made by Mattel for its Intellivision home computer. The 64's typewriter keyboard can also be used, but the Intellivision keyboard works very well. In one respect, it is even superior to the Melodian keyboard: it has forty-eight keys to the Melodian's forty. Laskowski has created an adaptor for the keyboard, which is included with the software. If you can't find one of these keyboards locally, Laskowski himself has a limited number for sale (see address below).

All of the above capabilities of **FastFingers** are standard features on today's electronic instruments, and Laskowski is to be congratulated for successfully programming them into the C-64.

Yet I have some doubts about the practicality of the approach. The software has been designed very much with the performing musician in mind. It is possible to control the program without using a monitor, but lugging a C-64 and a disk drive from gig to gig may not be worth the trouble even for those musicians who are interested in the initial investment in the system. An SX-64 would solve the problem handily, but they are hard to get these days.

Secondly, despite the great sounds Laskowski has coaxed out of the SID chip, the quality may just not be good enough for serious musicians. The SID has a signal-to-noise ratio of only 48 decibels, and notoriously fickle filters. Laskowski does claim to have significantly improved the dynamic range of the SID by processing the sound through an add-on DBX expander.

Laskowski should consider adapting his program for the Melodian keyboard, now that Melodian has released programming documentation for it. The Intellivision keyboards are going to be hard to get.

The organization of the controls in **FastFingers** is dense, but quite logical. The same cannot be said of the documentation, which suffers from a lack of clarity and friendliness.

## SID Editor

As a sysop of the Commodore Music and Graphics forum on CompuServe, I can get some idea of the success of a music creation program from the number of files submitted. For a long time, **Master Composer** was the favourite, not only because of its ease of use and its editing features, but because it was one of the first to create 'stand-alone' files (files that can be transmitted and run as independent programs). **Master Composer** has been toppled from pre-eminence, though, by the appearance of many pieces created with Harry Bratt's **SID Editor**. This very

powerful program can be found in *All About the Commodore 64, volume 2*, by Craig Chamberlain. Almost all the programs in this book are available from COMPUTE! Publications on a disk.

**SID Editor** gives you note-by-note control over every aspect of the sound. This program pushes the SID chip to its limits. Almost every imaginable musical effect can be created: tricky time signatures, syncopated rhythms, tempo variations, portamento, vibrato, ties and slurs. The same goes for the creation of the sounds themselves, with the availability of such features as filter sweeps and pulse-width sweeps, detuning, transposition, and so on.

But what really sets this program apart is that, in combination with the book, it can be used by either a total novice or a trained musician. The book provides a very good introduction to all the basics of musical notation and synthesizer technique. Almost every feature of the program can be accessed with a joystick. More experienced users can enter commands and note values through the keyboard.

Each note is displayed in three ways: as one of the keys on a keyboard, as a note on a staff, and as a letter. You can change any aspect of any note anywhere in the piece. One very powerful feature is the ability to define up to 16 phrases of music that can be called or redefined at any time. They can even consist entirely of commands, so that you could, for instance, call one phrase of notes with another phrase of commands, ringing the notes through a series of changes in key, articulation, tempo, and so on. In the hands of an expert, large pieces can be constructed very efficiently, conserving memory.

Like **Master Composer**, files created with **SID Editor** can be added to, and controlled by, other programs. Individual voices can be turned on or off; the music can be stopped and restarted from where it left off; and individual sections of the music can be flagged, so that the calling program can synchronize its actions with the music.

Four utility programs provided with **SID Editor** enable you to list the music, extract sections, append one or more files to another, and cross-merge files — use one of the voices of one file for the voice of another. This last is useful for rearranging the voices in one file.

## Music Assembler 64

**SID Editor** is a very complete program, but even it does not have the power of **MA64** by Nick Sullivan, so-called because

the music is written in a source code that has much in common with BASIC, and then assembled by the program into a playable 'object code'. **MA64** is a commercial version, rewritten and greatly expanded, of **Music Assembler 64**, a TPUG freeware program available on disk (C)\$1.

In addition to every music control feature of **SID Editor**, **MA64** provides such arcane features as waveform scheduling, where each note can be set up to switch between different waveforms while it plays; pulse width vibrato, a controllable oscillation of the pulse width; and duration patterns, which work like a sequencer, enabling a series of pitches to be played in the same repeating rhythm. If you can think of some new tricks to teach the SID chip, **MA64** even gives you a way to poke the SID directly from within the object code!

One feature I particularly like is a variable delay on the vibrato. When vibrato is used on regular instruments — a flute, say — the vibrato is not added by the player until a second or two after the note is started. The only traditional instrument that comes readily to mind where this is *not* done, is the pipe organ.

**MA64** is superior to **SID Editor** in its implementation of vibrato and glissando, because it defines the depth of the vibrato and the sweep rate of the glissando in terms of intervals, rather than frequencies. The depth of the vibrato in **SID Editor** must be adjusted as the base pitch of the note changes, because as the note gets higher, the musical interval defined by the difference in the upper and lower frequencies becomes smaller. To the ear, the depth of the vibrato becomes shallower, even though the frequency variation is the same. **MA64**, on the other hand, sets the depth of the vibrato as a fraction of an interval, which is independent of the base pitch of the note. To the ear, the vibrato depth stays the same, no matter how high or low the base frequency.

This program is not for the novice: some knowledge of music is required. **MA64** is a mini-language, and as such, can be a little daunting even to those who can read music, but who have little understanding of programming. Those who are familiar with BASIC will find themselves in friendly territory though, for **MA64** uses many of the statements found in BASIC. Below, I have listed a short sample program in **MA64**:

```
10 sys 592
20 [0]:tf=0::initialize
   transposition factor
```

```
30 for ia=1 to 5
40 gosub motif
50 tf=tf+1: .t tf
60 next ia
70 end
80 /motif
90 c3h:e:a:g:c4:b3q:a:gh
100 return
```

This can be entered just like a BASIC program, using the built-in screen editor; you can load or save it, and you can even use editing utilities like **POWER**. The **SYS** in line 10 invokes the 'assembler', which generates music 'object code' from the succeeding statements. The program, when assembled, will play the note values in line 90 five times, transposing it up one semitone each time.

Sullivan provides a tutorial program on the disk, and on-line help files, as well as **SID Monitor**, which displays what's happening in all the SID's registers as the music is being played.

### It's been fun...

All of the above programs, with the exception of **FastFingers**, provide excellent arrangements of classical and contemporary music as demonstrations, and complete, readable documentation. It's been a pleasure exploring *all* of these programs. "Music hath charms..." so they say, and that seems to be reflected in the generally high quality of music software for the Commodore 64.

**ConcertMaster**, **RhythmMaster**, **ScoreMaster**, **Programmer's Tool Kit**, **Musical Keyboard**, by Melodian, Inc., 970 W. McNab Road, Fort Lauderdale, Florida 33309. (\$29.95 US each for the software, \$99.95 US for the keyboard).

**MacMusic**, by Passport Designs, 625 Miramontes Street, Half Moon Bay, California 94019 (\$49.95 US).

**FastFingers** by Dan Laskowski, See-Thru Enterprises, 10382 Shenandoah, Windsor, Ontario N8R 1B5. (519) 735-2995 (\$119.85 US for keyboard, adaptor, and software; \$59.90 US minus keyboard; \$24.95 US for software alone).

**All About the Commodore 64, Vol. 2**, by Craig Chamberlain. Published by Compute! Books, Inc., P.O. Box 5406, Greensboro, North Carolina 27403 (\$16.95 US for the book, \$12.95 US plus shipping for the disk).

**MA64.v2**, by Nick Sullivan. Distributed by Wilanta Arts, 6943 Barrisdale Drive, Mississauga, Ontario L5N 2H5 (\$29.95 Cdn., \$19.95 US). □

# GRAND OPENING OF RICH-HILL TELECOM INTERNATIONAL

servicing all of RTC's products



## OPENING SPECIALS GALORE

SCRIPT 64 WORD PROCESSOR  
~~\$59.95~~ ONLY \$29.95

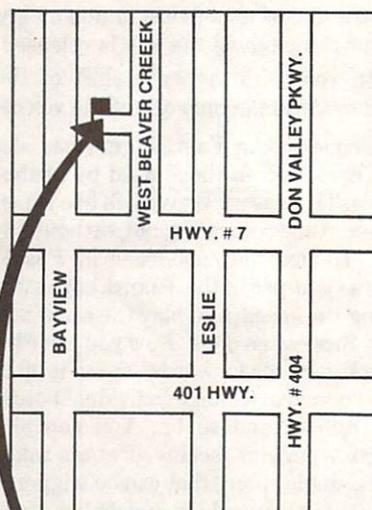
BUY & SELL SECTION  
TRADE UP!

FREE PROGRAM UPDATES  
BRING YOUR SCRIPT 64 DISK

FOR V2.7 UPDATE  
INTELTERM 64

~~\$49.00~~ \$39.00

★★ FREE BUFFET ★★



**RICH-HILL TELECOM  
INTERNATIONAL**  
200 West Beaver Creek Road  
Unit 15, Richmond Hill, Ont.  
Phone: (416) 731-1621

# Commodore 128 Memory Map

by Paul Blair  
and Jim Butterfield

By this time, memory maps for the Commodore 128 are sprouting up all over the place. But on these pages, Australian writer Paul Blair gives us a memory map with a difference. Paul has cross-referenced many locations on the 128 with their C-64 equivalents, so programmers who already have some familiarity with the layout of the 64 will more readily be able to get their bearings on the new machine. Many thanks to Paul for sharing this work with us.

\* \* \*

This memory map has been prepared to assist programmers understand more about the structure of BASIC 7.0 as implemented in the Commodore 128 computer. As the machine also has a C-64 operating mode, with BASIC 2.0, the map reflects common entry points as far as possible.

I have drawn on work done by Jim Butterfield. Jim had access to early releases of the C-128, and was generous in supplying information to friends. Without his assistance, this map would have been a long time coming. My thanks to him.

The information here represents our ideas of what is in Release 0 of the C-128 BASIC and Kernal ROMs. Consider this Version 1 of the map: it was completed in December, 1985, and it is possible that ROM revisions after that date will invalidate some of the data. It has not been the policy of Commodore to release memory maps to the general public, so we offer this map to help you get more from your Commodore computer.

## BANK 0

Label	C-128	C-64	Description
D6510	0000-1	0000-1	I/O port, similar to C-64
BANK	0002	NA	
PC-HI	0003	NA	Monitor use: Program counter MSB
PC-LO	0004	NA	Program counter LSB
S-REG	0005	NA	Processor status
A-REG	0006	NA	Accumulator (A) store
X-REG	0007	NA	X-register (X) store
Y-REG	0008	NA	Y-register (Y) store
STKPTR	0009	NA	Stack pointer (SP) store
INTEGR	0009		
CHARAC*	0009		Search character
ENDCHR	000A		
		Flag: scan for quote at end of string	
TRMPOS	000B	0009	Screen column from last tab
VERCK	000C	000A	Flag: 0=LOAD, 1=VERIFY
COUNT	000D	000B	Input buffer ptr/# of subscripts
DIMFLG	000E	000C	Flag: default array dimension
VALTYP	000F	000D	Variable type: \$FF=string, \$00=numeric
INTFLG	0010	000E	Var type: \$80=integer, \$00=floating point (FLP)
GARBFL	0011	000F	Flag: DATA scan/LIST quote/garbage collection
SUBFLG	0012	0010	Flag: subscript ref/user function call
INPFLG	0013	0011	Flag: \$00=INPUT, \$40=GET, \$98=READ
TANSGN	0014	0012	Flag: TAN sign/comparison result
POKER	0015		Current I/O device #
LINNUM	0016-17	0014-15	Integer value (line #, GOTO, POKE etc)
TEMPPT	0018	0016	Pointer: next string stack entry
LASTPT	0019-1A	0017-18	Pointer: current string stack entry
TEMPST	001B-23	0019-21	Stack: 9 bytes for 3 string pointers
INDEX1	0024-25	0022-23	General purpose pointer
INDEX2	0026-27	0024-25	General purpose pointer
RESHO	0028-2C	0026-2A	Workspace used by multiply and divide
TXTTAB	002D-2E	002B-2C	Pointer: start of BASIC (for Bank 0)
VARTAB	002F-30	002D-2E	Pointer: start of variables (Bank 1)
ARYTAB	0031-32	002F-30	Pointer: start of arrays
STREND	0033-34	0031-32	Pointer: end of arrays + 1
FRETO	0035-36	0033-34	Pointer: string storage (moving down)
FRESPC	0037-38	0035-36	Utility string pointer
MAXMM1	0039-3A	0037-38	Pointer: limit of memory (Bank 1 in C-128)
CURLIN	003B-3C	0039-3A	Current BASIC line number
TXTPTR	003D-3E	007A-7B	Pointer: BASIC work pointer (CHRGET etc)
FORM	003F	NA	Used by PRINT USING
FNDPTR*	003F-4A	NA	Pointer to item found by search
DATLIN	0041-42	003F-40	Current DATA line number
DATPTR	0043-44	0041-42	Current DATA address
INPPTR	0045-46	0043-44	Vector: INPUT routine
VARNAM	0047-48	0045-46	Current BASIC variable name
VARPNT	0049-4A	0047-48	Current BASIC variable address
FDECPT*	0049-4A		
LSTPNT	004B-4C		
ANDMSK*			Mask used for comparisons
FORPNT*		0049-4A	Pointer: index variable for FOR/NEXT
EORMSK*			Mask used for comparison
VARTXT	004D-4E	004B-4C	Pointer to operator table
OPPTR*			
OPMASK	004F	004D	Mask used for comparison
GRBPNT	0050-51	004E-4F	Pointer: used for function definition
TEMPF3*			
DEFPNT*			
DSCPNT	0052-0054	0050-51	Pointer: used for string operations
HELPER	0055		Flag: HELP or LIST
JMPER	0056-58	0054-56	\$4C (JMP) + address of function
TEMPF1	0059		
PTARG1	TEMPF1+2		Multiple defined for INSTR
PTARG2	TEMPF1+2		
STR1	TEMPF1+4		
STR2	TEMPF1+7		
POSITN	TEMPF1+10		
MATCH	TEMPF1+11		
ARYPNT	005A	0058-59	Used to define arrays (DIM)
HIGHDS*	005A-5B	0058-59	Pointer: used for block transfer
HIGHTR	005C-005D	005A-5B	Pointer: also used for block transfer
TEMPF2	005E	005C-0060	Temporary FLP Accumulator (Accum)
DECCNT	005F-60	005D-5E	# decimal point digits in conversions
GRBTOP	0061	005F	Flag: used to test for decimal pt in strings
DPTFLG*			
LOWTR*			
EXPNGN	0062	0060	Sign of exponent
FACEX	0063	0061	FLP Accum #1: exponent
FACHO	0064-67	0062-65	FLP Accum #1: mantissa
FACSGN	0068	0066	FLP Accum #1: sign
ARGEXP	006A	0069	FLP Accum #2: repeat as per FLPA #1
ARGHO	006B-6E	006A-006D	
ARGSGN	006F	006E	
ARISGN	0070	006F	Sign comparison, Accum #1:#2
FACOV	0071	0070	FLP Accum #1 rounding
FBUFPNT	0072-73	0071-72	Pointer: cassette buffer
AUTINC	0074	NA	Increment value for AUTO (\$00=off)
MVDFLG	0076	NA	Flag: 10K hi-res allocated
KEYNUM	0077	NA	Key chosen
NOZE*		NA	Leading zero counter
SPRNUM*		NA	Move sprite and sprite temporary Counter
HULP	0078	NA	Counter
KEYSIZ*	0078	NA	Key command length
SYNTMP	0079	NA	Temp for indirect loads
DSEDESC	007A-7C	NA	Descriptor for disk error (DSS)
TOS	007D-7E	NA	BASIC pseudo stack pointer
RUNMOD	007F	(see 9D)	Flag: RUN/direct mode
PARSTS	0080	NA	Status word for DOS parser
POINT*	0080	NA	Pointer to decimal point
PARSTX	0081	NA	
OLDSTK	0082	NA	
<b>Graphic zero page storage</b>			
COLSEL	0083	NA	Current colour selected
MULTC1	0084	NA	
MULTC2	0085	NA	
FG	0086	NA	
SCALEX	0087	NA	Scale factor in X
SCALEY	0089	NA	Scale factor in Y
STOPNB	008B	NA	Stop PAINT if not same colour
GRAPNT	008C-8D	NA	Utility pointers
VTEMP1	008E	NA	
VTEMP2	008F	NA	
STATUS	0090	0090	ST = status

STKEY	0091	0091	Flag: STOP and RVS keys	COLUMN	00EE	NA	Max number of screen columns
SVXT	0092	0092	Tape timing constant	DATA	00EF	00D7	Current char to print
VERCK	0093	0093	Flag: \$00 = LOAD, \$01 = VERIFY	LSTCHR	00F0	NA	Previous char printed (for ESC text)
C3PO	0094	0094	Flag: serial character awaiting output	COLOR	00F1	0286	Current colour to print
BSOUR	0095	0095	Actual serial character waiting	TCOLOR	00F2	NA	Saved attribute to print (INST/DEL)
SYNO	0096	0096	Flag: end of data block from tape	RVS	00F3	00C7	Flag: reverse mode on/off
XSAV	0097	0097	Temporary store during input (tape/RS232)	QTSW	00F4	00D4	Flag: quote mode
			# open files/index into file table	INSRT	00F5	00D8	Flag: insert mode (>0=# inserts)
LDTND	0098	0098	Default input device (0=keyboard)	INSFLG	00F6	NA	Flag: auto-insert mode (\$00=off)
DFTLN	0099	0099	Default output (CMD) device (3=screen)	LOCKS	00F7	NA	Disable SHIFT/C=, CTRL-S
DFTLO	009A	009A	Cassette parity during write	SCROLL	00F8	NA	Disable screen scroll, line linker
PRTY	009B	009B	Flag: byte read complete	BEEPER	00F9	NA	Disable CTRL-G (bell)
DPSW	009C	009C	I/O messages: \$00=nil, \$40=errors, \$80=all		00F9-FA		UNUSED
MSGFLG	009D	009D	Cassette error pass 1				

## PAGE 1

PTR1	009E	009E	Temp workspace 1	FBUFR	0100-FF		Processor stack area
T1*	009E		Cassette error pass 2	XCNT	0110		DOS loop counter
PTR2	009F	009F	Temp workspace 2	DOSF1L	0111		DOS filename 1 length
T2*	009F		Jiffy clock	DOSDS1	0112		DOS disk drive 1
TIME	00A0-A2	00A0-A2	Cassette/serial temp	DOSF1A	0113-14		DOS filename 1 address
PCNTR	00A3		Serial routine temp store	DOSF2L	0115		DOS filename 2 length
BSOUR1	00A4		Serial/cassette temp store	DOSDS2	0116		DOS disk drive 2
COUNT	00A5	00A5	Cassette buffer pointer	DOSF2A	0117-18		DOS filename 2 address
CNTDN*			RS232 receiver input bit storage	DOSOF2	0119-1A		BLOAD/BSAVE start address (SA)
BUFPT	00A6	00A6	RS232 receiver bit count in	DOSOFH	011B-1C		BSAVE ending address (EA)
INBIT	00A7	00A7	RS232 receiver flag for start bit check	DOSLA	011D		DOS logical address
BITCI	00A8	00A8	RS232 receiver byte buffer	DOSFA	011E		DOS physical address
RINONE	00A9	00A9	RS232 receiver party storage	DOSSA	011F		DOS secondary address
RIDATA	00AA	00AA	Pointer to tape buffer	DOSRCL	0120		DOS record length (REL files)
RIPITY	00AB	00AB	Tape end address/end of program	DOSBNK	0121		DOS Bank #
SAL	00AC-AD	00AC-AD	Tape timing constant	DOSDID	0122-23		DOS disk ID
EAL	00AE-AF	00AE-AF	Address of tape buffer	DIDCHK	0124		Disk ID check
CMPO	00B0	00B0	RS232 transmit bit count				
TEMP	00B1		RS232 transmit next bit to send				
TAPE1	00B2		RS232 transmit byte buffer				
BITTS	00B4	00B4	Number of characters in file name				
NXTBIT	00B5	00B5	Current logical file				
RODATA	00B6	00B6	Current secondary address				
FNLEN	00B7	00B7	Current device				
LA	00B8	00B8	Pointer to file name				
SA	00B9	00B9	RS232 transmit priority				
FA	00BA	00BA	Output character				
FNADR	00BB-BC	00BB-BC	# blocks left to read/write				
ROPRTY	00BD	00BD	Temp to collect byte				
OCCHAR*	00BD	00BD	Cassette motor control flag				
FSBLK	00BE	00BE	I/O start address				
MYCH	00BF	00BF	Pointer for general use				
DRIVE*			Tape read/write data				
CAS1	00C0	00C0	Bank for current LOAD/SAVE/VERIFY				
STAL	00C1-C2	00C1-C2	Bank where current file name (for FNADR)				
MEMUSS	00C3-C4	00C3-C4	RS232 input buffer addresses				
TMP2	00C5	NA	RS232 output buffer addresses				
BA	00C6	NA					
FNANK	00C7	NA					

### PRINT USING work area (C-128 only)

BNR	0125		Pointer: beginning number
ENR	0126		Pointer: end number
DOLR	0127		Flag: dollar
FLAG	0128		Flag: comma
SWE	0129		Counter
USGN	012A		Sign exponent
UEXP	012B		Pointer: exponent
VN	012C		# digits before decimal point
CHSN	012D		Flag: justify
VF	012E		# of positions before decimal point
NF	012F		# of positions after decimal point
POSP	0130		Flag: plus or minus
FESP	0131		Flag: exponent
ETOF	0132		Switch
CFORM	0133		Character counter
SNO	0134		Sign number
BLFD	0135		Flag: blank/asterisk
BEGFD	0136		Pointer: beginning of field
LFOR	0137		Length of format
ENDFD	0138		Pointer: end of field

## PAGE 2

BUF	0200-A0	0200-58	BASIC input buffer
FETCH	02A2-AE	NA	Bank PEEK subroutine
FETVEC	02AA	NA	
STASH	02AF-BD	NA	Bank POKE subroutine
STAVEC	02B9	NA	
CMPARE	02BE-CC	NA	Bank compare subroutine
CMPVEC	02C8	NA	
JSRFAR	02CD-E2	NA	JSR to another Bank
JMPFAR	02E3-FB	NA	JMP to another Bank
	02FC-FD	NA	Function execute hook

## PAGE 3

IERROR	0300-01	0300-01	Vector to \$4D3F/\$E38B: output error message
MAIN	0302-03	0302-03	Vector to \$4DC6/\$A483: main system loop
ICRNCH	0304-05	0304-05	Vector to \$430D/\$A57C: tokenize routine
IQPLOP	0306-07	0306-07	Vector to \$5151/\$A71A: LIST routine
IGONE	0308-09	0308-09	Vector to \$4AA2/\$A7E4: RUN routine
IEVAL	030A-0B	030A-0B	Vector to \$78DA/\$AE86: eval routine
IESCLK	030C-0D	NA	Vector to \$4321/NA: escape token crunch

### Global screen editor variables

KEYTAB	00CC-CD	00F5-00F6	Keyboard decode pointer (Bank 15 in C-128)
IMPARM	00CE-CF	NA	PRIMM utility string pointer
NDX	00D0	00C6	Number of characters in keyboard buffer
KYNDX	00D1	NA	Number of programmed chars waiting
KEYIDX	00D2	NA	Flag: index to function key string
SHFLAG	00D3	028D	Flag: shift key: \$00=no shift
SFDX	00D4	00CB	Copy of LSTX for rollover
LSTX	00D5	00C5	Last key pressed
CRSW	00D6	00D0	Input from screen/keyboard
MODE	00D7	NA	Flag: 40/80 columns (\$00=40 columns)
GRAPHM	00D8	NA	Flag: text/graphic mode
CHAREN	00D9	NA	Character base: \$00=ROM, \$04=RAM
SEDSA	00DA-DD	NA	Pointers: MOVLIN routine (2 addresses)
SED12	00DE-DF	NA	Pointers: SAVPOS
KEYVAR	00DA-DF	NA	Programmable key variables

### LOCAL screen editor variables (40/80 change maps out)

PNT	00E0-E1	00D1-D2	Pointer to screen line/cursor
USER	00E2-E3	00F3-F4	Colour line pointer
SCBOT	00E4-E7	NA	Window lower limit
SCTO	00E5	NA	Window upper limit
SCLF	00E6	NA	Window left margin
SCRT	00E7	NA	Window right margin
LXSP	00E8-E9	00C9-CA	Input cursor log (row, column)
INDX	00EA	00C8	Current input line end
TBLX	00EB	00D6	Position of cursor on screen line
PNTR	00EC	00D3	Row where cursor lives
LINES	00ED	NA	Max number of screen lines

IESCP	030E-0F	NA	Vector to \$51CD/NA: escape token print	BITNUM	0A15	0298	RS232 # bits to send
IESCEX	0310-11	NA	Vector to \$4BA9/NA: execute vector	BAUDOF	0A16-17	0299-9A	RS232 baud rate full bit time
.....	0312-13	NA	Unused (\$FFFF)	RIDBE	0A18	029B	RS232 receive pointer
CINV	0314-15	SAME	Vector to \$FA65/\$EA31: IRQ	RIDBS	0A19	029C	RS232 input pointer
CBINV	0316-17	FOR	Vector to \$B003/\$FE66: Break interrupt	RODBS	0A1A	029D	RS232 transmit pointer
NMINV	0318-19	C64	Vector to \$FA40/\$FE47: NMI interrupt	RODBE	0A1B	029E	RS232 send pointer
IOPEN	031A-1B	...	From \$FFC0 to \$EFBD/\$F34A: OPEN routine	SERIAL	0A1C	NA	Flag: fast serial internal/external
ICLOSE	031C-1D	...	From \$FFC3 to \$F188/\$F291: CLOSE routine	TIMER	0A1D-1F	NA	Decrementing jiffy register
ICHKIN	031E-1F	...	From \$FFC6 to \$F106/\$F20E: CHKIN routine	XMA	0A20	0289	Keyboard buffer size
ICKOUT	0320-21	...	From \$FFC9 to \$F14C/\$F250: CHKOUT routine	PAUSE	0A21	NA	CTRL-S flag
ICLRCH	0322-23	...	From \$FFC6 to \$F226/\$F333: CLRCHN routine	RPTFLG	0A22	028A	Flag: key repeat (\$80 = all, \$40 = none)
IBASIN	0324-25	...	From \$FFCF to \$EF06/\$F157: CHRIN routine	KOUNT	0A23	028B	Delay between key repeats
IBSOUT	0326-27	...	From \$FFD2 to \$EF79/\$F1CA: CHROUT routine	DELAY	0A24	028C	Delay before key repeats
ISTOP	0328-29	...	From \$FFE1 to \$F66E/\$F6ED: STOP routine	LSTSHF	0A25	028E	Delay between SHIFT/C = toggles
IGETIN	032A-2B	...	From \$FFE4 to \$EEEE/\$F13E: GETIN routine	BLNON	0A26	00CF	Cursor mode: \$00 = blink, \$FF = steady
ICLALL	032C-2D	...	From \$FFE7 to \$F222/\$F32F: CLALL routine	BLNSW	0A27	00CC	Cursor blink enable: \$00 = flash
EXMON	032E-2F	NA	Vector to \$B006/NA: Monitor	BLNCT	0A28	00CD	Cursor blink counter
ILOAD	0330-31	...	Vector to \$F26C/\$F4A5: LOAD	GDBLN	0A29	00CE	Character under cursor
ISAVE	0332-33	...	Vector to \$F54E/\$F5ED: SAVE	GDCOL	0A2A	0287	Cursor colour before blink
CTLVEC	0334-35	NA	Vector to \$C7B9/NA: Print CTRL chars	CURMOD	0A2B	NA	VDC cursor mode (when enabled)
SHFVEC	0336-37	NA	Vector to \$C805/NA: Print SHIFT chars	VM1	0A2C	NA	VIC text screen/char base pointer
ESCVEC	0338-39	NA	Vector to \$C9C1/NA: Print ESC chars	VM2	0A2D	NA	VIC bit map base
KEYVEC	033A-3B	NA	Vector to \$C5E1/NA: keyscan logic	VM3	0A2E	NA	VDC text screen base
KEYCHK	033C-3D	NA	Vector to \$C6AD/NA: store key	VM4	0A2F	NA	VDC attributes base
DECODE	033E-49	NA	Vector to \$FA80/NA: keyboard matrix tables	LINTMP	0A30	NA	
KEYD	034A-53	0277-80	IRQ Keyboard buffer	SAV80A	0A31	NA	80 col routines work area
TABMAP	0354-5D	NA	Tab stop bits	SAV80B	0A32	NA	
BITABL	035E-61	NA	Line wrap bits	SAV80C	0A33	NA	
LAT	0362-6B	0259-62	Logical file table	SAV80D	0A34	NA	
FAT	036C-75	0263-6C	Device number table	CURCOL	0A35	NA	VDC cursor colour before blink
SAT	0376-7F	026D-76	Secondary address table	SPLIT	0A36	NA	Split screen raster value
CHRGET	0380-9E	0073-8A	CHRGET subroutine	FNADRX	0A37	NA	Save X during Bank operations
CHRGOT	0386	0079	CHRGOT entry	PALCNT	0A38	NA	Jiffy adjustment for PAL systems
QNUM	0390	007C	Check for number				

#### MONITOR stores (C-128 only)

XCNT	0A80		Compare buffer
HULP	0AA0		
FORMAT	0AAA		
LENGTH	0AAB		
MSAL	0AAC		
SXREG	0AAF		
SYREG	0AB0		
WRAP	0AB1		
XSAVE	0AB2		
DIRCTN	0AB3		
TEMPM	0AB4		
CURBNK	0AC0	NA	Current function key ROM bank
PAT	0AC1	NA	Physical address table
TBUFFR	0B00-BF	033C-FB	Cassette buffer
RS232I	0C00-0DFF		RS232 I/O buffers
RS232O	0E00-0FFF		System sprites
PHKBUF	1000-09	NA	Programmed key lengths
PHKDEF	100A-FF	NA	Programmed key lengths and definitions
DOSSTR	1100-1130		DOS output buffer

#### Graphics variables

1131-1138

#### Line drawing variables

1139-1148

#### Angle routine variables

1149-114F

#### Circle drawing variables or general use variables or shape and move shape or graphics variables

ADRAY1	117A-117B	0003-4	Vector to FLP-fixed routine (\$84B4)
ADRAY2	117C-117D	0005-6	Vector to Fixed-FLP routine (\$793C)
LPEN	11E9-11EA	NA	Light pen X,Y values
OLDLIN	1200-01	003B-3C	Previous BASIC line number
OLDTXT	1202-03	003D-3E	Pointer: BASIC statement for CONT
PUCHRS	1204-1207	NA	PRINT USING chars ( , , \$)
ERRNUM	1208	NA	ER = Error type
ERRLIN	1209-120A	NA	EL = Error line number (FFFF No error)
TRAPNO	120B-120C	NA	Line to go to on error (FFxx none)
TMPTRP	120D	NA	Hold trap # temp
ERRTXT	120E		
TEXTTP	1210-11		End of BASIC (Bank 0)
MAXMMO	1212-13		BASIC program limit \$FF00
TMPTXT	1214-15		Used by DO loop
USRPOK	1218-1A	0310-12	USR program jump
RNDX	121B-1F	008B-8F	RND seed value

#### Music Stores

1220-1271

#### Interrupt Stores

1276-1280

## BANK 0

VICSCR	0400-07E7		40-column screen memory
	07F8-07FF		Sprite identity area
	0800-09FF	NA	BASIC pseudo stack

#### Absolute Kernel Variables

SYSVEC	0A00-0A01	NA	Vector: BASIC restart
DEJAVU	0A02		Kernel warm/cold start status vector
PALNTS	0A03		PAL/NTSC system flag
INSTAT	0A04		
MEMSTR	0A05-06	0281-82	Bottom of memory pointer
MEMSIZ	0A07-08	0283-84	Top of memory pointer
IRQTMP	0A09-0A	029F-A0	Save for IRQ during tape
CASTON	0A0B	02A2	TOD sense during tape
KIKA26	0A0C	02A3	Tape read temp
STUPID	0A0D	02A4	Tape read IRQ indicator
TIMOUT	0A0E	NA	Fast serial timeout flag
ENABL	0A0F	02A1	RS232 enables
M51CTR	0A10	0293	RS232 control register
M51CDR	0A11	0294	RS232 command register
M51AJB	0A12-13	0295-96	RS232 user baud rate
RSSTAT	0A14	0297	RS232 status register

### Sound Stores

START	1281-12FF		
BASIC	1C01	0801	Normal start of BASIC text
	2000-3FFF		Screen memory (hi-res)
	4000-FBFF		RAM memory (hi-res)

## BANK 1

0400-FBFF BASIC variables, arrays, strings

## BANK 14

Same as Bank 15, except:

D000-DFFF Character generator ROM (same as C-64)

## BANK 15

4000-CFFF	A000-C000	BASIC ROM
D000-D030	....	40-col video chip 8564 (Same as 64)
D400-D41C	....	SID sound chip 6581 (Same as 64)
D500-D50A	NA	MMU 8722 memory setup registers
D600-D601	NA	80-column CRTC
D800-D8E7	....	Colour nybbles (Same as 64)
DC00-DC0F	....	CIA1 6526 (IRQ) (Same as 64)
DD00-DD0F	....	CIA2 6526 (NMI) (Same as 64)
DF00-DF0A	NA	DMA controller
E000-FE00	E000-FFFF	Kernal ROM
FF05-FFFF	FF81-FFFF	ROM: transfer, jump table

## BASIC 7.0/2.0 ROM Routines

Label	B7.0	B2.0	Description
BCOLD	4000	A000	JMP COLD START (\$4023) Start of Basic ROMs
	4003	A003	JMP WARM START (\$4009)
	4006		JMP \$A84D IRQ S/R
BASSFT	4009	E37B	Warm start routine
INIT	4023	E394	Cold start routine
INITCZ	4045	E3BF	Initialize BASIC
	4112	NA	Bell
	417A	NA	Set preconfiguration registers
	418D	NA	Set up sprite movement tables
INITMS	419B	E422	Print "COMMODORE Basic 7.0..."
INITV	4251	E453	Set BASIC links initialize vectors
INITAT	4279	E3A2	CHRGET routine to move to \$0380
	42CE	NA	Indirect fetches: from (\$50) Bank 0
	42D3		from (\$3F) Bank 1
	42D8		from (\$52) Bank 1
	42DD		from (\$5C) Bank 0
	42E2		from (\$5C) Bank 1
	42E7		from (\$66) Bank 1
	42EC		from (\$61) Bank 0
	42F1		from (\$70) Bank 0
	42F6		from (\$70) Bank 1
	42FB		from (\$50) Bank 1
	4300		from (\$61) Bank 1
	4305		from (\$24) Bank 0
CRUNCH	430A		Crunch tokens (\$AF8A)
CRUNCH	430D	A57C	Crunch tokens (\$0304)
ESCLK	4321		Escape token crunch (\$030C)
	43E2		Check if keyword found
RESLST	4417	A093	Table of Keywords
OPLIST	46FD	A140	Keyword vectors
OPTAB	4828		Operator vectors (\$AF6F)
ERRTAB	484B	A19E	Table of error messages
ERROR	4A82	A437	Find message (.A) start of error msg
GONE	4AA2	A7E4	Read and execute next statement (\$0308)
NEWSTT	4AF6		Set up statement for execution (\$AF90)
CONT	4B34	A857	Update CONT pointer
	4B3F	NA	Execute/trace statement
ESCEX	4BA9		Skip statement (\$0310)
STOP	4BCB	A82C	Perform STOP
END	4BCD	A82F	Perform END
GETFMN	4BF7	B3E1	Set up FN reference
			4C86
			4C89
			4CB6
			4D2A
			4D3F
			4DAF
			4DC6
		A480	4DCD
			4DE2
		A49C	4F4F
		A533	4F93
		A560	4FAA
			4FFE
			5017
			5047
			5050
			5059
			5064
		A613	FNDLIN
		A96B	LINGET
		A69C	LIST
		A6C9	5151
			51CD
		A642	NEW
		A659	RUNC
		A65E	51F8
		A677	LDCLR
		A68E	5238
		A68E	STXPT
		A8D2	5254
		A8F8	5262
		A93B	RETURN
		A906	DATA
		A909	528F
		A928	REM
		NA	529D
		NA	DATAN
		NA	52A2
		NA	52A5
		NA	52C5
		NA	IF
		NA	5320
		NA	537C
		NA	5391
		A94B	ONGOTO
		A9A5	53A3
			53C6
		AA80	54F6
		AA86	PRINTN
		AAA0	553A
			5540
			555A
			5600
		AB7B	GET
		ABA5	5612
		ABBF	INPUTN
		ABBF	5648
		ABBF	INPUT
		AC06	5662
		AD1E	QINLIN
		B081	569C
		E12A	READ
		NA	56A9
		NA	57F4
		NA	587B
		NA	5885
		NA	58B4
		NA	58B7
		NA	58BD
		NA	5975
		NA	5986
		NA	59AC
		A883	GOSUB
		A8A0	59CF
			59DB
			5A1D
			5A3D
		A857	CONT
		A871	5A60
			5A9B
			5AA6
			5ACA
		A81D	RESTRE
			5AE1
		NA	5AF8
		NA	5BAE
		NA	5BFB
		NA	5D19
		NA	5D68
		NA	5D75
		MA	5D89
		NA	5D99
		NA	5DA7
		NA	5DC6
		NA	5DDF
		A742	FOR
		NA	5DF9
		NA	5E87
		NA	5EFB
		NA	5F34
		NA	5F4D
		NA	5F62
		NA	5FB7
		NA	5FD8
		NA	5FDB
		NA	5FE0
		NA	6039
		NA	608A
		NA	60B4
		NA	60B7
			Perform OR
			Perform AND
			Perform COMPARE
			Print "READY"
			Output error (\$0300)
			BREAK
			Ready for BASIC (\$0302)
			Ready (\$AF85)
			Handle new line
			Rechain lines (\$AF87)
			Get BASIC line
			Search BASIC stack
			Insert in stack
			Check if room for more
			Copy stack pointer
			Set stack pointer
			Delete from stack
			Find BASIC line (\$AF8D)
			Get FLP number (\$AF9F)
			Perform LIST
			LIST subroutine (\$0306)
			Escape token print (\$030E)
			Perform NEW (\$AF84)
			Get ready to RUN (\$AF7E)
			Perform CLR (\$AF81)
			Reset stack
			Back up text pointer
			Perform RETURN
			Perform DATA/BEND
			Perform REM
			Scan to next statement
			Scan to next line
			Perform IF
			Search/skip BEGIN/BEND
			Skip string constant
			Perform ELSE
			Perform ON
			Perform LET
			Check string location
			Perform PRINT#
			Perform CMD
			Perform PRINT
			Print format character
			Perform GET
			Perform INPUT#
			Perform INPUT
			Print "?" and take input
			Perform READ
			Perform NEXT
			Perform DIM
			Perform SYS
			Perform TRON
			Perform TROFF
			Perform RREG
			Perform AUTO
			Perform HELP
			Place help marker
			Perform GOSUB
			Perform GOTO
			Push return address to stack
			Perform GO
			Perform CONT
			Perform RUN (\$AF7B)
			Get next statement (\$AF99)
			Perform RESTORE (\$AF9C)
			Perform RENUMBER
			RENUMBER (continued)
			Scan program
			Change line number
			Find RENUMBER start
			Line count
			RENUMBER increment S/R
			Scan forwards
			Block move
			Move block down
			Move block up
			Perform FOR
			Perform DELETE
			Get line range
			Perform PUDEF
			Perform TRAP
			Perform RESUME
			Restore TRAP pointer
			Bad syntax — exit
			"CAN'T RESUME"
			Perform DO
			Perform EXIT
			Perform LOOP
			"LOOP NOT FOUND"
			"LOOP WITHOUT DO"

	60E1	NA	Define programmable key		83E1	NA	Perform XOR
	610A	NA	Perform KEY		8407	NA	Perform RWINDOW
	61A8	NA	Perform PAINT	RND	8437	E097	Perform RND
	62B7	NA	Perform BOX	RNDO		E094	
	642B	NA	Perform SSHAPE	QSETNR		E0BE	
	658D	NA	Perform GSHAPE	RND1		E0D3	
	668E	NA	Perform CIRCLE	RNDRNG		E0E3	
CIRSUB	6750	NA	Draw circle (\$AF78)		84A7		Evaluate fixed number
	6797	NA	Perform DRAW	AYINT	84B4		FLP-fixed signed (\$AF00)
	67D7	NA	Perform CHAR	POS	84C9	B39E	Perform POS
	6955	NA	Perform LOCATE	ERRDIR	84D9	B3A6	Check if direct mode
	6960	NA	Perform SCALE		84D0		"ILLEGAL DIRECT"
	69E2	NA	Perform COLOR		84E0		"UNDEF'D FUNCTION"
	6A5C	NA	Store current colours		84E5		Set up 16-bit fixed-FLP
	6A79	NA	Perform SCNCLR		84F5		"DIRECT MODE ONLY"
	6B06	NA	Fill memory page	DEF	84FA	B3B3	Perform DEF
	6B17	NA	Set screen colour	GETFNM	8528	B3E1	Check FN syntax
	6B30	NA	Clear hi-res screen	FNDORF	853B	B3F4	Perform FN
	6B5A	NA	Perform GRAPHIC	STRD	85AE	B465	Perform STR\$
	6BC9	NA	Perform BANK	CHRD	85BF	B6EC	Perform CHR\$
	6BD7	NA	Perform SLEEP	LEFD	85D6	B700	Perform LEFT\$
	6C2D	NA	Perform WAIT	RIGHTD	860A	B72D	Perform RIGHT\$
	6C4F	NA	Perform SPRITE	MIDD	861C	B737	Perform MID\$
	6CC6	NA	Perform MOVSPR	PREAM	864D	B761	Set up string parameters
	6DE1	NA	Perform PLAY	LEN	8668	B77C	Perform LEN
	6E02	NA	PLAY subroutine	LEN1	866E	B782	Exit from string mode
	6EB2	NA	Set SID sound	ASC	8677	B78B	Perform ASC
	6EFD	NA	PLAY error		8688		Calculate string vector
	6F03	NA	Dotted note		869A		Set up string
	6F1E	NA	Note A-G	MOVINS	874E	B67A	Move string to store
	6F69	NA	Sharp	FRESTR	877B	B6A3	Discard unwanted string
	6F6C	NA	Flat	FRETMS	87E0	B6DB	Clean descriptor stack
	6F78	NA	Rest		87F1		Get integer byte
	6FD7	NA	Perform TEMPO		8803		Get parameters for POKE/WAIT
				WAIT		B82D	
	7046	NA	Perform FILTER	GETADR	8815	B7F7	Convert FLP to an address (\$AF0C)
	70C1	NA	Perform ENVELOPE	FSUB	882E	B853	Perform SUBTRACT (\$AF12)
	7164	NA	Perform COLLISION	FSUBT	8831		Arg-FLP Accum #1 (\$AF15)
	7190	NA	Perform SPRCOLOR	FADD	8845	B867	Add memory (\$AF1B)
	71B6	NA	Perform WIDTH	FADDT	8848	B86F	Perform ADD (\$AF1B)
	71C5	NA	Perform VOL		8917		Adjust FLP Accum #1
	71E1	NA	Perform SOUND		894E		Round up FLP Accum #1
	72CC	NA	Perform WINDOW	OVERR	895D	B97E	"OVERFLOW"
	7335	NA	Perform BOOT	LOG	89CA	B9EA	Perform LOG (\$AF2A)
	7372	NA	Perform SPRDEF		8A0E	B849	Add 0.5
	76EC	NA	Perform SPRSAV	FMULT	8A24	BA28	Multiply by memory (\$AF1E)
	77B3	NA	Perform FAST	FMULTT	8A27	BA30	Perform MULTIPLY (\$AF21)
	77C4	NA	Perform SLOW	ROMUPK	8A89	BA59	Unpack ROM to FLP Accum #2 (\$AF5C)
FRMNUM	77D7	AD8A	Type match check				Unpack RAM to FLP Accum #2 (\$AF5A)
	77E7		"TYPE MISMATCH MISMATCH"	CONUPK	8AB4	BA8C	Adjust FLP Accum #1/#2 (\$AF5A)
	77EA		"FORMULA TOO COMPLEX"				Multiply by 10
FRMEVL	77EF	AE83	Evaluate expression (\$AF96)	MULDIV	8AE3	BAB7	"DIVISION BY ZERO"
EVAL	78D7		Evaluate item (\$AF93)	MUL10	8B17	BAE2	Divide by 10
EVAL1	78DA		Evaluate item (\$030A)		8B33		Divide into memory (\$AF24)
GIVAYF	793C		Convert fixed-FLP (\$AF03)	DIV10	8B38	BAFE	Perform DIVIDE (\$AF27)
PARCHK	7950	AEF1	Eval within parentheses ()	FDIV	8B49	BB07	Unpack ROM to FLP Accum #1 (\$AF63)
CHKCOM	795C	AEFD	Check for comma	FDIVT	8B4C	BB12	Pack FLP Accum #1 to \$5E
SYNERR	796C	AF08	Syntax error	MOVFM	8BD4	BBA2	Pack FLP Accum #1 to \$59
ISVAR	7978	AF28	Search for variable				Pack FLP Accum #1 to RAM (\$AF66)
MOVFRM	7A85		Unpack RAM to FLP Accum #1 (\$AF60)	MOV2F	8BF9	BB27	FLP Accum #2 to FLP Accum #1 (\$AF69)
			Locate variable	MOV1F	8BFC	BBCA	FLP Accum #1 to FLP Accum #2 (\$AF6C)
			Check variable	MOVMF	8C00	BBD0	Round FLP Accum #1 (\$AF4B)
DOREL		B016	Check if numeric	MOVFA	8C28	BBFC	Get sign (\$AF51)
NUMREL		B01B	Check if string				Perform SGN
STREL	7B3C	B02E	Create new variable	MOVAF	8C38	BC0C	
NOTFNS	7B46	B11D	Set up array				Convert fixed to FLP (\$AF0F)
NOTFDD	7CAB	B261	"BAD SUBSCRIPT"	ROUND	8C47	BC1B	Perform ABS (\$AF4E)
BSERR	7D25	B245	"ILLEGAL QUANTITY"	SIGN	8C57	BC39	Compare FLP Accum #1 to memory (\$AF54)
	7D28		Calculate array size	SGN	8C65	BC39	Convert FLP to fixed
UMULT	7E3E	B34C	Array pointer S/R	ACTOFC		BC3C	Perform INT (\$AF2D)
	7E71			INTOFC		BC44	Move string to FLP Accum #1
				FLOATC	8C75		
FRE	8000	B37D	Perform FRE	ABS	8C84	BC58	Get ASCII digit
VAL	804A	B7AD	Perform VAL	FCOMP	8C87	BC5B	"IN"
VAL1	8052		Convert ASCII string to FLP (\$AF09)				Print integer
	8076	NA	Perform DEC				Convert FLP to ASCII (\$AF06)
PEEK	80C5	B80D	Perform PEEK	QINT	8CC7	BC9B	
POKE	80E5	B824	Perform POKE	INT	8CFB	BCCC	
	80F6	NA	Perform ERR\$	FIN	8D22	BCF3	
	8139	NA	Swap .X with .Y	AADD		BD7E	
	8142	NA	Perform HEX\$	STCONS		BDB3	
	816B	NA	Convert byte to hex		8DB0		
	8182	NA	Perform RGR	INPRT	8E26	BDC2	
	818C	NA	Get graphics mode	LINPRT	8E32	BDCD	
	819B	NA	Perform RCLR	FOUT	8E42	BDDD	
	8203	NA	Perform JOY	FOUTIM		BE68	
	824D	NA	Perform POT	TICONS		BF11	
	82AE	NA	Perform PEN	SQR	8FB7	BF71	Perform SQR (\$AF30)
	82FA	NA	Perform POINTER	FPWR	8FBE	BF7B	Raise to power (\$AF36)
	831E	NA	Perform RSPRITE	NEGOT		BFB4	
	837C	NA	Perform BUMP	EXCONS		BFBF	
	8397	NA	Perform RSPOS	FPWRT	8FC1		Perform exponent (\$AF39)
				NEGOP	8FFA	BFB4	Perform MINUS/SUBTRACT (\$AF33)



KBINP		E5CA		SCNDK	E4E0	EDB9	From \$FF96: send SA after LISTEN
KSINP		E632		CIOUTK	E503	EDDD	From \$FFA8: output byte on serial bus
SCINP	C29B	E63A	Input from screen (\$C009)	UNLTKK	E515	EFEF	From \$FFAB: send UNTALK on serial bus
TGLQT	C2FF	E684	Check for quotes				From \$FFAE: send UNLISTEN on serial bus
SCPUT		E691		UNLSNK	E526	EDFE	
SCPNT		E716					
	C30C		Wrap up screen print	ACPTRK		EE13	
IRQK	C55D	EA87	Scan keyboard (\$C012)	NMICNT		EEBB	
KEYVEC	C5E1		Keyboard read (\$033A)		E5C3	NA	Turn fast disk off
REPEAT	C651		Get key and repeat (\$C015)		E5D6	NA	Turn fast disk on
KEYCHK	C6A0		Store key (\$033C)		E5FB	NA	From \$FF47: flip fast disk on/off
KBDTBL	C6AD	EB81	Keyboard decode tables	RSWRT	E5FF	EF06	Transmit RS232
PRINT	C72D		Print to screen (\$C00C)	RSBLD	E69D	EF59	RS232 receive
	C77D	NA	ESC-O (escape)		E795	F04D	Set up RS232 input
	C7B9		Print CTRL character (\$0334)		E7CE	F086	Get from RS232
	C805		Print SHIFT character (\$0336)		E7EC		Interlock RS232/serial
	C854		CHRS(29): cursor right		E805		RS232 control I/O
	C85A		CHRS(17): cursor down		E878		RS232 receive timing
	C880		CHRS(14): text		E8A9		RS232 transmit timing
	C8A6		CHRS(11): lock	THEADR	E8D0	F72C	Find any tape header
	C8AC		CHRS(12): unlock		E919	F76A	Write tape header
	C8B3		CHRS(19): home		E980	F70D	Get buffer address
	C8C2		CHRS(18): rvs		E987	F7D7	Get SA, EA of tape buffer
	C8C7	NA	CHRS(2): underline on		E99A	F7EA	Find specific header
	C8CE	NA	CHRS(130): underline off		E9BE	F80D	Advance tape pointer
	C8D5	NA	CHRS(15): flash on		E9C8	F817	"PRESS PLAY..."
	C8DC	NA	CHRS(143): flash off		E9DF	F82E	Check tape state
	C8E3		Open screen line		E9E9		"PRESS RECORD..."
	C91B		CHRS(20): delete		E9F2	F841	Initiate tape read
	C94F	NA	CHRS(9): tab		EA18	F864	Initiate tape write
	C961	NA	CHRS(24): tab switch		EA26	F875	Tape code
	C980	NA	ESC Z: clear all tabs		EA8F	F8D0	Check tape stop
	C983	NA	ESC Y: set default tabs		EEA1	F8E2	Set read timing
	C98E	NA	CHRS(7): bell		EAEB		Read tape bits
	C9B1		CHRS(10): linefeed		EC1F		Store tape characters
ESCAPE	C9C1	NA	Test ESC sequence (\$C01E,\$0338)		ED51		Reset pointer
	CA14	NA	ESC T: set window top		ED5A		New character setup
	CA16	NA	ESC B: set window bottom		ED69		Send transition to tape
WINDOW	CA1B	NA	Set window (\$C02D)		ED8B		Write data to tape
	CA3D	NA	ESC I: insert line		ED90		Tape write
	CA52	NA	ESC D: delete line		EE2E		Tape leader
	CA76	NA	ESC Q: erase to line end		EE9B		Switch IRQ vector
	CA8B	NA	ESC P: erase to line start		EEA8		IRQ vectors
	CA9F	NA	ESC (at): clear rest of screen		EEB0		Tape motor off
	CABC	NA	ESC V: scroll up		EED0		Control tape motor
	CACA	NA	ESC W: scroll down	NGETIN	EEEE	F13E	From \$FFE4,\$032A: get a char (usually keybd)
	CAE2	NA	ESC L: scroll on				
	CAE5	NA	ESC M: scroll off	NBASIN	EF06	F157	From \$FFCF,\$0324: get a char from input device
	CAEA	NA	ESC C: cancel auto-insert				
	CAED	NA	ESC A: auto insert		EF48		Get char from tape
	CAF2	NA	ESC S: block cursor	NBSOUT	EF79	F1CA	From \$FFD2,\$0326: send char to output device
	CAFE	NA	ESC U: underline cursor				
	CB0B	NA	ESC E: non-flashing cursor	NOPEN	EFBD	F34A	From \$FFC0,\$031A: open file for read/write
	CB21	NA	ESC F: flashing cursor				
	CB37	NA	ESC G: bell enable				
	CB3A	NA	ESC H: bell disable				
	CB3F	NA	ESC R: screen reverse	NCHKIN	F106	F20E	From \$FFC6,\$031E: prepare a file for input
	CB48	NA	ESC N: screen normal				
	CB52	NA	ESC K: go to end of line	NCKOUT	F14C	F250	From \$FFC9,\$0320: prepare a file for output
	CB81	NA	ESC J: go to start of line				
	CC34	E716	Send character to screen (\$C003)	NCLOSE	F188	F291	From \$FFC3,\$031C: close a file
SCRORG	CC5B	E505	Read row/column (\$C00F)		F202		Search for file
PLOT	CC6A	E50A	Read/set cursor (\$C018)		F212		Set file parameters
KEYSET	CCA2	NA	Define function key (\$C021)	NCLALL	F222	F32F	From \$FFE7,\$032C: abort I/O, close files
SWAPPR	CD2E	NA	ESC X: switch 40/80 screen (\$C02A)	NCLRCH	F226	F333	From \$FFCC,\$0322: restore default I/O devices
CURSORS	CD57	NA	Position 80-col cursor (\$C01B)				
	CDCA	NA	Set Register 31	CLRIO	F23D	NA	From \$FF4A: clear I/O
	CDCC	NA	Set screen register	LOADSP	F265	F49E	From \$FFD5: load memory from disk or tape
	CDD8	NA	Read Register 31				
	CDDA	NA	Read screen register	LOAD	F26C	F4A5	Load (\$0330)
	CDE6	NA	Set CRT to screen address		F27B	F4BF	Serial load
	CDF9	NA	Set CRT to colour address		F32A	F539	Tape load
INIT80	CE0C	NA	Initialize 80 columns (\$C027)		F3A1		Disk load
					F4BA		Get serial byte
	E000	Reset S/R			F4C5		Receive serial byte
RESTOR	E056	FD15	From \$FF8A: reset standard I/O devices		F503		Toggle clock line
					F50F		"SEARCHING"
VECTOR	E05B	FD1A	From \$FF8D: set/store I/O vectors		F533		"LOADING"
RAMTAS	E093	FD50	From \$FF87: test RAM	SAVEP	F53E	F5DD	From \$FFD8: save memory to disk or tape
	E0CD	NA	Transfer all Banks				
IOINIT	E109	FDA3	From \$FF84: initialize I/O	SAVE	F54E	F5F4	Save (\$0332)
	E1F0	NA	Search for Boot ROM		F5B5	F642	Terminate serial input
	E242	NA	Reset to 64/128		F5BC		"SAVING"
	E24B	NA	From \$FF4D: GO64		F5C8	F65F	Save to tape
	E263	NA	Code to move to \$02		F5F8	F69B	From \$FFE4: update clock (+1)
TALKK	E33B	ED09	From \$FFB4: send TALK on serial bus	UDTIMK	F65E	F6DD	From \$FFDE: read TI clock
LSTNK	E33E	ED0C	From \$FFB1: send LISTEN on serial bus	RDTIMK	F665	F6E4	From \$FFDB: set TI clock
				SETMK	F66E	F6ED	From \$FFE1,\$0328: scan STOP key
	E43E	ED40	From \$FFA5: send byte on serial bus	NSTOP	F67C	F6FB	"TOO MANY FILES"
	EDAD		Error: Device not present		F67F	F6FE	"FILE OPEN"
TKSAK	E4D2	EDC7	From \$FF93: send secondary address (SA)		F682	F701	"FILE NOT OPEN"

	F685	F704	"FILE NOT FOUND"	FFBA	Set LA FA SA (\$F738)
	F688	F707	"DEVICE NOT PRESENT"	FFBD	Set length, FN address (\$F731)
	F68B	F70A	"NOT INPUT FILE"	FFC0	Open logical file (\$031A,\$EFBD)
	F68E	F70D	"NOT OUTPUT FILE"	FFC3	Close logical file (\$031C,\$F188)
	F691	F710	"MISSING FILE NAME"	FFC6	Open channel in (\$031E,\$F106)
	F694	F713	"ILLEGAL DEVICE NO"	FFC9	Open channel out (\$0320,\$F14C)
	F71E	F71F	Print if direct	FFCC	Close I/O channels (\$0322,\$F226)
SETNAM	F722		Print I/O message	FFCF	Input from channel (\$0324,\$EF06)
SETLFS	F731	FD99	From \$FFBD: set filename	FFD2	Output to channel (\$0326,\$EF79)
	F738	FE00	From \$FFBA: set file#, device#, set adr	FFD5	Load from file (\$F265)
			From \$FF68: set load/save Bank	FFD8	Save to file (\$F53E)
READST	F744	FE07	From \$FFB7: read ST byte into .A	FFDE	Set internal clock (\$F665)
SETMSG	F75C	FE18	From \$FF90: error messages on/off	FFE1	Read internal clock (\$F65E)
MEMTOP	F763	FE25	From \$FF99: read/set upper limit of BASIC	FFE4	Scan STOP key (\$0328,\$F66E)
			From \$FF9C: read/set lower limit of BASIC	FFE7	Get char from queue (\$032A,\$EEEB)
MEMBOT	F772	FE34	From \$FF9C: read/set lower limit of BASIC	FFEA	Close all files (\$032C,\$F222)
			From \$FFF3: get start of keyboard registers	FFED	Increment clock (\$F5FB)
IOBASK	F781	E500	From \$FF5C: search for SA	FFF0	Read screen size (\$C00F)
			From \$FF59: search/setup file	FFF3	Read/set X,Y on screen (\$C018)
			From \$FF50: trigger DMA	FFFA	Return I/O base (\$F781)
			Get character from memory	FFFC	\$FF05 = NMI
			Store loaded byte	FFFE	\$FF3D = reset
			Read byte to be saved		\$FF17 = IRQ
		NA	From \$FF74: get char from Bank	NMI	
		NA	From \$FF77: store char to Bank	RESET	
		NA	From \$FF7A: compare char with Bank	IRQ	
		NA	From \$FF6B: get memory control mask		
		NA	S/R to go to \$02A2-\$02FB		
		NA	DMA code to go to \$03F0		
		NA	From \$FF56: check autostart ROM		
		NA	From \$FF53: check for boot disk		
		NA	Read next boot sector		
		NA	Convert to 2 digit decimal		
PRIMMS	FA17	NA	From \$FF7D: Print message		
	FA40		NMI interrupt entry (\$0318)		
	FA65		Normal IRQ entry (\$0314)		
	FF05		NMI transfer entry		
	FF17		IRQ transfer entry		
	FF33		Return from interrupt		

## Kernal Table

For space economy, this table gives C-128 vectors only.

	FF47	NA	Fast disk toggle (\$E5FB)	4189	Registers for \$D501
	FF4A	NA	Clear I/O (\$F23D)	4267	BASIC links
	FF4D	NA	GO64 (\$E24B)	46FC	Action vectors
	FF50	NA	Trigger DMA (\$F7A5)	47D8	Function vectors
BTCALL	FF53	NA	Boot load program from disk (\$F890)	5250	PUDEF characters
PHENIX	FF56	NA	Call card cold start rtnes (\$F867)	5AF0	Keywords/RENUMBER
LKUPLA	FF59	NA	Search tables for LA (\$F79D)	6A4C	Colour codes
LKUPSA	FF5C	NA	Search tables for SA (\$F786)	6CB3	Bit masks
SWAPPER	FF5F	NA	Swap to alt display device (\$C02A)	6FE4	Voice parameters
DLCHR	FF62	NA	Initialize 80 col character RAM (\$C027)	6FE7	Length characters
PFKEY	FF65	NA	Program function key (\$C021)	6FEC	Command characters
SETBNK	FF68	NA	Set Bank for LOAD/SAVE/VERIFY (\$F73F)	702F	Bell sequence
			Convert Bank to MMU specification (\$F7EC)	7039	SID voice steps
GETCFG	FF6B	NA	JSR to any Bank, RTS to calling Bank (\$02CD)	7691	Sprite vectors
JSRFAR	FF6E	NA	JMP to any Bank (\$02E3)	8490	RND multiplier
JMPFAR	FF71	NA	LDA (FETVEC),Y from any Bank (\$F7D0)	849A	Value 32768
INDFET	FF74	NA	STA (STAVEC),Y to any Bank (\$F7DA)	8F76	+0.5
			CMP (CMPVEC),Y to any Bank (\$F7E3)	8F7B	Decimal constants
INDSTA	FF77	NA	Print immediate (\$FA17-always JSR here)	8F9F	TI constants
INDCMP	FF7A	NA	Release # of KERNAL ROM	9005	Exponent series
PRIMM	FF7D	NA	Initialize screen editor (\$C000)	9485	Trigonometry series
			Initialize input/output (\$E109)	9CCA	Video matrix lines hi
RELEASE	FF80		RAM test (\$E093)	9D1C	Bit masks
	FF81		Reset vectors to default values (\$E056)	A627	DOS command masks
	FF84		Change vectors for user (\$E05B)	B6C3	Mode tables for MONITOR
	FF87		Control OS messages (\$F75C)	B715	Mode characters for MONITOR
	FF8A		Send SA after LISTEN (\$E4D2)	B721	Compacted mnemonics for MONITOR
	FF8D		Send SA after TALK (\$E4E0)	C033	Screen address low
	FF90		Set/read top of memory (\$F763)	C04C	Screen address high
	FF93		Set/read bottom of memory (\$F772)	C065	I/O link vectors
	FF96		Scan keyboard (\$C012)	C06F	Keyboard shift vectors
	FF99		Not used in C128	C6DD	Programmed keys
	FF9C		Handshake byte in (\$E43E)	C79A	Vectors
	FF9F		Handshake byte out (\$E503)	CEA8	Programmed key lengths
	FFA2		Send UNTALK (\$E515)	E073	Vectors to \$0314
	FFA5		Send UNLISTEN (\$E526)	E105	Bank transfer masks
	FFA8		Send LISTEN (\$E33E)	E2C7	VIC 8564 setup
	FFAB		Send TALK (\$E33B)	E2F8	CRT 8563 setup pairs
	FFAE		Get I/O status ST (\$F744)	F50C	"U0" disk reset
	FFB1			FAB0	Keyboard matrix unshifted (\$033E,\$0346)
	FFB4			FAD9	Keyboard matrix shifted (\$0340)
	FFB7			FB32	Keyboard matrix C = key (\$0342)
				FB8B	Keyboard matrix CONTROL key (\$0344)
				FBE4	Keyboard matrix caps lock (\$0348)
				FF00	MMU controls

## Workspace In BANK 1

BASIC 7.0 uses variable workspace in Bank 1 for its own requirements. A short list of recognized areas is given below. The list is not exhaustive, and care should be taken if you plan to use this space for your own programming.

\* \* \*



# Library Additions

TPUG's library of public domain software grows month by month. Hundreds of disks containing thousands of programs are available to TPUG members at the nominal cost of ten dollars per disk. Considering that each disk is packed with good programs, at today's software prices, this is a fantastic value.

In order for the library to keep growing, our librarians need a constant supply of new programs. If you have written a program or a collection of programs that you think might be an asset to the library, please send it to: TPUG Program Library, 101 Duncan Mill Road, Suite G7, Don Mills, Ontario M3B 1Z3, Canada. If your contribution is accepted, you will be sent the library disk of your choice. If, for some reason, your contribution is not needed, your original disk will be returned to you.

On these pages our librarians describe recent additions to the library.

## VIC 20 Disk (V)TQ

### Presented by Richard Best

The first disk/tape of 1986 starts out with a hectic game called **Obliteration**. This is a cross between **Centipede** and **Space Invaders**, in which you try to stay alive while continuously shooting at approaching aliens. In **Starbase**, another space game, you protect your space station by firing rockets at the approaching asteroids. The popular game **Berzerk** has appeared on other VIC disks. This update is noisier, and also runs with a SuperExpander.

Youngsters will find **Timer/Add** challenging — you have to solve as many addition problems as you can before the timer runs out. **Cannibal** is a simulation in which you set the ratio of cannibals to explorers. The explorers may not survive surrounded by hungry cannibals. Beginning programmers are not the only ones who will enjoy **Dumb Prg**, a clever demo of the **CHR\$( )** function.

If you're anxious to press your VIC into service as a day planner, **Chimes** is just the thing. It allows you to set up to nine alarms, each with a message. Or how about a simple filing system? **DBMS** will store page after page of text and then save it all to tape.

'Power users' will get a lot of use out of **5K Reset**, **8K Off**, and **8K On**, which will help to clean up memory pointers, or

enable and disable an 8K expander. And disk users have been screaming for **Fast Format**, a utility that formats a diskette in only 40 seconds. Once you've formatted the disk, examine its surface with **View BAM**.

The two remaining programs on this month's disk are truly unique. **Men-Flex** is a test for the cryptically inclined. It presents you with 20 number riddles derived from common phrases. **Write +** is a tool to be used by someone who is handicapped, and unable to communicate by normal means. It turns the VIC screen into a menu of words and letters, and the joystick into a pointer. Sentences can then be written on the screen by selecting their parts from the menu.

Many thanks to all those who contributed programs throughout 1985. With your continued efforts we should be able to keep the VIC alive for at least as long as the venerable PET. □

```
0 vic 20 disk (v) tq
11 "list-me (v) t-q/1" prg
10 "list-me (v) t-q/2" prg
15 "oblit ins.v5k" prg
15 "oblit ins.cass" prg
15 "obliteration" prg
6 "timer/add.v" prg
8 "starbase ins.v5k" prg
8 "starbase main" prg
2 "5k reset.v5k" prg
5 "cannibal.v" prg
12 "chimes.v" prg
5 "dumb prg.v" prg
11 "write+.v" prg
25 "write+.seq" seq
11 "berzerk.v5k" prg
2 "8k off.v12k" prg
1 "8k on.v12k" prg
30 "men-flex.v12k" prg
8 "dbms.v" prg
8 "fast format.v5k" prg
6 "view bam.v5k" prg
```

## Commodore 64 Disk (C)TQ

### Presented by Derick Campbell

This month's C-64 disk has great programs for everyone.

Telecommunicators won't want to miss the latest version of **Xmobuf**, a terminal program that is great for using on Delphi, CompuServe, and on TPUG's own

bulletin board system (though you will not be able to use the program for uploading and downloading on the TPUG BBS).

For programmers, we have the ultimate super-utility: a program with all the commands you'd ever want, and all in one package: the **DOS Wedge**, **Tiny Aid**, **Super DOS**, **Xtramon**, and many more.

If it's educational computing that interests you, we have a computer quiz — just right for students; while for rest and recreation there are two games, and a song — **Karma Chameleon**.

Who could ask for more? Don't miss it! □

```
0 c-64 disk (c) tq
8 "list-me (c) tq" prg
28 "tank combat.c" prg
1 "pinball" prg
160 "pinball.pb" prg
3 "quiz.inst.c" prg
57 "quiz.c" prg
18 "prquiz.c" prg
18 "super.aid.c" prg
14 "wedge.inst.d" prg
14 "tap.inst.d" prg
21 "super.dos.inst.d" prg
42 "s.a.comments.d" prg
36 "xmon.inst.d" prg
5 "tap.9b5b-9fff.d" prg
10 "n&w.c700-cfff.d" prg
12 "xmon.74cc-801b.d" prg
9 "culture club.c" prg
36 "chameleon.d" prg
78 "xmobuf 7.4" prg
55 "xmobuf.doc" seq
```

## C-128 Disk (Y)AA

### Presented by James Kokkinen

This is the first TPUG library disk for the C-128 computer. Due to the relative 'newness' of this computer, submissions to the library are just beginning to dribble in, so there are some unused blocks at the end of this disk. All contributions to the C-128 library will be gratefully accepted for review and possible inclusion in future monthly diskettes.

What we *do* have this month is an extensive demonstration of some of the features of the C-128 computer, as well as some communications terminal programs.

**Invention 13.mus** is Commodore's demonstration of the C-128's SID chip playing two voices simultaneously. **Simple Key.mus** and **Music Board** are short routines that allow users to create their own little tunes. **Soundmaker** allows easy creation of sounds that can be used in games, or as audible prompts in other programs.

There are four communications programs on this month's disk. **Small-term.128** is an X-Modem protocol terminal that, as well as **CP/Term.128**, can download to single-sided C-128 CP/M diskettes. The **List Me (T)T1**, as well as the sequential documentation files on this diskette, explain the above files in more detail, as well as **Tacoterm.128**, **CP/M term2** and **JHF Miniterm**.

In the recreational vein, there are two games included this month. **Litter Patrol** is a one-player game that has the participant collecting garbage strewn over the highway by non-environmentally-minded automobile passengers. **Deepsix.128** has the single participant diving for gems and dodging various underwater hazards. Both games are enjoyable examples of what one can create with the C-128's simplified graphics commands.

**Sprites on 128** is the text of Sean Rooney's well-received presentation to the February C-128 meeting. The balance of the sprite programs on this diskette were created and/or adapted by Sean for his presentation, and are included here for your enjoyment.

The utilities included are mostly taken from the 1571 Demo Diskette, with a few exceptions, some of which are noted here. **Uni-copy** is Jim Butterfield's (it works in 128 mode as well!). **Keypad 64.C-128** is actually a C-64 program that permits the use of the C-128's keypad and cursor keys while in C-64 mode. **2 1571 keys** allows the use of DOS commands for those fortunate enough to own two 1571 disk drives.

As noted by Bill Graves in the **List Me** program, several of these programs were downloaded from various bulletin board systems and, to the best of our knowledge, are all in the public domain.

*Editor's note: There will be no PET disk this month but Mike Donegan informs us that there will be an Amiga disk by the time you read this.* □

0 c-128 disk (y)aa  
 17 "list me (t)t1" prg  
 1 ".....music....." prg  
 7 "invention 13.mus" prg  
 3 "simple key.mus" prg  
 3 "music board" prg

2	"soundmaker"	prg	3	"car race"	prg
1	"..communications"	prg	1	"...utilities..."	prg
37	"smallterm.128"	prg	14	"uni-copy"	prg
51	"cp/m term.128"	prg	8	"dir print"	prg
16	"cp/mterm.doc"	seq	9	"dir squeeze"	prg
69	"tacoterm.128"	prg	14	"disk-dump"	prg
18	"cp/m term2"	prg	8	"display bam"	prg
5	"jhf miniterm"	prg	6	"change unit"	prg
13	"jhf miniterm doc"	seq	8	"unscratch"	prg
1	"..games/demos..."	prg	8	"load address"	prg
13	"litter patrol"	prg	5	"seq lister"	prg
14	"deepsix.128"	prg	5	"header change"	prg
18	"sprites on 128"	seq	9	"printer test"	prg
7	"sprsav demo"	prg	4	"format disk"	prg
3	"bent space"	prg	8	"autoboot maker"	prg
2	"black hole"	prg	4	"keypad 64.c128"	prg
3	"wavy spokes"	prg	2	"2 1571 keys"	prg

## We'll Send You Money!

That's right — TPUG Inc. will send you — with NO obligation (on our part) — hard cash that you can spend as you please. You don't even have to be a TPUG member! Hard to believe? It's true!

Here's how to take advantage of this special offer: Just send us a self-addressed stamped envelope, along with an original unpublished article suitable for TPUG Magazine. The topic doesn't matter — just make sure that it relates to Commodore computers, and that it's brilliant. We do the rest.

Your article should be on a Commodore disk, preferably with accompanying hard copy. Sequential PETASCII files are best, with no embedded word processor commands, but other formats are acceptable. Just be sure to state both disk format and word processor used on the label of the disk, along with your name and the filename of the article. We'll get back to you as soon as possible, usually within three weeks of receiving your submission.

By the way, if you have an idea for an article, but you're not sure whether it's suitable for TPUG Magazine, check it out with us first. Just call (416) 445-4524 (9 to 5 EST weekdays), or drop us a line at: Submissions Department, TPUG Magazine, 101 Duncan Mill Road, Suite G-7, Don Mills, Ontario, Canada M3B 1Z3.

And remember, if we print your article — we'll send you money!

# Micro Processes

## Amiga/C-64 File Transfers

by Chris Johnson

I am writing this article with **Ed**, the full-screen editor that comes as part of AmigaDOS. When I write an article for *TPUG Magazine*, I usually send it by modem. Unfortunately, I have neither a modem nor a terminal program for the Amiga; I do have a Commodore 1650 modem and several terminal programs for my C-64.

I also have a null-modem cable that I use to connect the two computers (with a VIC-1011A interface on the C-64); and the AmigaDOS has commands that enable the transfer of material. (These would work just as well if there was a modem connected.)

On the C-64 I load a terminal program with a buffer. I am using **FirstTerm**, which is available on a TPUG disk. I open the buffer with **logo-b**. I am using the default communications speed of 300 BIPS, though I could increase it to 1200 BIPS. (The C-64 does not receive reliably at higher speeds than that, though I regularly *send* material from a C-64 to another computer at 2400 BIPS.)

On the Amiga, I have used **Preferences** to set the baud rate at 300. I open a CLI window and enter the following command: **type <filename> to ser:**. This command will be familiar to users of CP/M or MS-DOS, but new to most Commodore users. If I had left off the **to ser:**, the file would have been printed on the screen. As it is, the file is sent to the serial port, through the null-modem cable, and into the buffer of my C-64. When the file is received, I leave terminal mode on the C-64, save the buffer to disk, and ship it later to *TPUG Magazine* by modem. The same command can be used to receive files from another computer. This time, the command is:

**TYPE SER: to df0:<filename>**

This takes the bytes coming in through the serial port and stores them in the specified file. To exit and close the file, type CTRL-C on the Amiga; this will not take effect until the incoming buffer is full, so transmit a string of characters (spaces will do) from the other computer until the **\*\*BREAK** appears on the Amiga.

There is one small problem, however, with sending **Ed** files from the Amiga, which is that **Ed** stores data line by line, with a linefeed character (**chr\$(10)**) at the end of each line. Most word processors on Commodore computers use the carriage return character (**chr\$(13)**) as the terminator, and then only at the end of each paragraph, not each line.

But it is not hard to write a program to strip the unwanted linefeeds from the file. The only requirement, when preparing the **Ed** file, is that a 'hard' line ending — one that will not be translated to a space when the file is formatted for printing or typesetting — must be entered with two carriage returns. This is normally done at the end of a paragraph, for instance, and at the end of each line of such data as program listings.

The stripping of the unwanted characters can be done at either

end. I use the following ABasiC program on the Amiga to send the files and strip the line endings at the same time.

```
10 cnl$ = chr$(155) + "E"
20 un$ = chr$(10)
30 header$ = home$ + cnl$ + " Transmit file to serial port."
  + cnl$ + chr$(10) + string$(37," ")
40 print header$; cnl$
50 input "Filename"; filename$
60 on error goto 180
70 open "i", #1, filename$
80 open "o", #2, "ser:"
90 while not eof(1)
100 line input #1, a$
110 if lastline% and len(a$) then print #2, " ";
120 print #2, a$;
130 if len(a$) = 0 then print #2, chr$(13); chr$(13);
140 lastline% = len(a$)
150 wend
160 close
170 end
180 print err$(err); cnl$
190 print "Press any key to continue."
200 getkey i$: scnclr: resume 40
```

## Formatting MFM Disks on the 1571 Disk Drive

by M. Garamszeghy

Copyright © 1986 M. Garamszeghy

The 1571 disk drive is capable of reading and writing a number of standard CP/M disk formats when used with a C-128 in CP/M mode, including (among others):

- IBM CP/M-86 single-sided
- IBM CP/M-86 double-sided
- Kaypro II single-sided
- Kaypro IV double-sided
- Osborne single-sided
- Epson QX-10 double-sided
- Slicer 9

These are all referred to as MFM (maximum frequency modulation) formats, based on the technique used to record the data on the disk. Normal Commodore DOS (used by the 1541, for example) uses a method called GCR (group coded recording), which is totally different from MFM. There are several other differences between MFM and GCR disks. MFM disks generally have the same number of sectors on each track, while Commodore GCR disks usually have different numbers of sectors for different ranges of track numbers. MFM tracks are numbered from zero, while the sectors on each track are numbered from one. This is the opposite of Commodore GCR

disks, where tracks are numbered from one and sectors start at zero.

Although Commodore CP/M can read and write in MFM formats with a 1571 drive, the current version of the `format.com` program supplied on the CP/M system disk cannot create or format new disks in these extended disk formats. It is, however, quite simple to format disks in these or any other format you wish with a 1571 drive.

In addition to the standard Commodore DOS commands, the 1571 has a few new ones. These commands, collectively defined as 'burst mode' in the 1571 manual, allow you to read, write and format disks in virtually any format you choose. Most of the commands, such as those for reading from and writing to the disk, need fairly sophisticated machine language programming to access, because the required high-speed data transfer is under totally manual control. The extended `format` command is the simplest to use. It can be quite easily invoked from BASIC because high-speed data transmission is not required.

The description of the burst mode commands in the 1571 manual is cryptic at best. The syntax for the MFM `format` command can be simplified to:

```
open15,8,15,"u0"+chr$(b1)+chr$(129)+chr$(0)
+chr$(b4)+chr$(39)+chr$(b6):dclose
```

In this instruction, `b1` must be set to 70 for a single-sided disk, or 102 for a double-sided disk; `b4` is a number from 0 to 3, representing 128, 256, 512 and 1024 bytes per sector respectively; and `b6` is the number of sectors per track.

The actual bit patterns in the characters sent after the `U0` (which selects the burst mode) are explained in detail in the 1571 manual. I have simplified them here to make the command easier to use. There are other optional parameters that allow you to partially format a disk, format different areas of the disk in a different manner, change the sector slew rate, change the fill byte, and so on. These can be the basis of, for example, a simple yet very effective copy protection scheme that you can use on your own disks and programs. Try reformatting track 35 on a standard 1541 DOS disk with the instruction:

```
open15,8,15,"u0"+chr$(198)+chr$(129)+chr$(0)
+chr$(0)+chr$(35)+chr$(10)+chr$(34)+chr$(34):dclose
```

Now see what happens when you try to copy the disk, even with a whole-disk nibble-type copier for protected disks. If the program you wish to protect tries a random access block-read or block-write to the reformatted track, an error will result. If the drive (either a 1541 or 1571) does not return an appropriate read error code, the disk has been formatted in GCR and is obviously not an original. The reformatting of track 35 should be done before you copy your program to the disk, in order to ensure that no program data are lost by wiping out the track. In addition, you should use the DOS `block-allocate` command to allocate all sectors on the track, to prevent DOS from accidentally trying to write there.

The following short BASIC 7.0 program, `MFMFormat`, is for use in the C-128 native mode. It allows you to effortlessly format disks in a number of different MFM formats on the 1571 drive. The `DATA` statements at the end of the program contain the required parameters (sectors per track, bytes per sector and number of sides) for each type of disk. These parameters can all be changed to create custom disk formats if you wish. However, most of the possible formats will not be supported by CP/M on the C-128. In addition, the line numbers for the `DATA` statements should not be changed, due to the use of a

calculated line number reference in line 30 for reading the selected data.

```
10 print"          mfm disk formatter":print
   "          by m. garamszeghy"
20 print" options: ":fori=1to5:reada,a,a,a
   $:printi": ";a$:next
30 input" select a format";f:restore(f*10
   +100):read b6,bs,sd,f$
40 print" format . >> ";f$:print" press <r
   eturn> to continue"
50 print" or press any other key to abort"
   :getkeya$:ifa$<>chr$(13)thenrun
60 b5=39:s(0)=128:s(1)=256:s(2)=512:s(3)=1
   024:si=0:ifsd=2thensi=32
70 b1=70ors:fori=0to3:ifbs=s(i)thenb4=i:el
   senext
80 open15,8,15,"u0"+chr$(b1)+chr$(129)+chr
   $(0)+chr$(b4)+chr$(39)+chr$(b6)
90 input#15,a:dclose:input" format anothe
   r <y/n>";fa$:iffa$="y"then40
100 data8,512,1,"ibm cp/m-86 single-sided
110 data8,512,2,"ibm cp/m-86 double-sided
120 data10,512,2,"kaypro iv double-sided
130 data10,512,1,"kaypro ii single-sided
140 data5,1024,1,"osborne single-sided
```

## A Secret in a DRAW Command

by Elizabeth Deal

There is an undocumented polar-coordinate drawing command in the BASICs for the newer Commodore machines, the Plus/4, the C-16 and the C-128. If you disassemble Plus/4 ROMs, you will see that the `DRAW` command begins at `$C4D9`. It uses several subroutines, of which one — at `$C3F7` — is of interest: it fans out to do various kinds of jobs depending on which of many permitted syntaxes are in the program. The system tries to get X and Y coordinates but, if it finds them separated by a semicolon (the test for this is at `$C400-C405`), what would be X is taken to mean radius length, and what would be Y is taken to mean an angle (North is 0 degrees, East is 90, and so on. Notice I said 'degrees' — not 'radians'!). A value over 360 degrees wraps around back to zero, but there is a two-byte limit on its size.

The variety of permitted syntaxes seems almost unlimited. The following two programs will allow you to study this useful variant of the `DRAW` command in more detail.

In the `Rays` program, line 150 just plots a point 25 pixels away from the center, placed at  $x=y=100$ , at an angle varying with `J`, which in this case is 12 points on a circle, every 30 degrees (like the hour markers on the clock). If you add, one by one, the remaining lines (listed here with `REM` in front of the code) you should be able to see how `DRAW` behaves.

Pressing any key cancels the graphic mode, so you can play with the program.

In the `Tiny Turtle` program, the drawing angle is a sum of previously used angles, so that you can draw in the same fashion as in the LOGO language. 'Distance' is LOGO's `forward` command, and 'angle' is LOGO's `turn` command. A bunch of

## Micro Processes

60-degree turns with the same distance value will close up a hexagon.

I am not testing *DA* for a 360-degree limit; you may wish to do that. To get out of the input loop, just enter negative values. **SCALE 1** allows fairly large values to be used — see your manual for details.

100 rem rays

120 scale 0:graphic 1,1

130 n = 12:a = 360/n

140 for j = 0 to n:da = j\*a

150 draw 1,100,100,25;da

160 rem draw 1 to 35;da

170 rem draw 0 to 4;da

180 rem draw 1 to 10;da

190 next j

200 get key i\$:graphic 0

100 rem tiny turtle path

110 scale 1:graphic 2,1

120 locate 100,100:da = 0

130 input "distance, angle";d,a

140 if d < 0 or a < 0 then graphic 0:end

150 da = data

160 draw 1 to d;da

170 goto 130

A project: A spiral is a straight line in the polar coordinate system. Try writing a tiny routine to draw at least five swings of a spiral. □

set of **Home Accountant** programs, and is *not* the **GRAPH** option accessed from the **MAIN MENU**. Use this program to **LOAD** in your saved graph.

- When your saved graph finishes painting itself on the screen, and the disk drive light goes out, hit the **F1** key. Your graph should be dumped to your printer.

The graphs that result are somewhat elongated from how they appear on the screen, but nevertheless look quite professional. Other screen dump utility programs can be used, but the **Cardco** program is inexpensive and simple to use, and it is widely available.

If you have trouble with the printouts, experiment with the DIP switch settings on your interface and/or printer, or with the memory location you selected for the screen dump routine from the **Cardco Printer Utility** program (four default locations are available). I found that my 10X gave good results with my Tymac Connection printer interface set to the 'transparent' mode with line feeds on. □

## Organize Your Disk Directories

by Karl Thurber

How often have you forgotten which directory entries belong with which set of programs? Which program is the one to load? Which program is the instruction file?

If you work with programs that involve several sub-programs or files, it's well worth your trouble to organize their placement on the disk. A typical scenario would be to have an "empty" filename that shows what follows, such as **"- - -Blaster Game"**. The first file should be the one you load, and may be called the program's name, or perhaps **"Boot"**. The main program files, loaded by the boot (if any is required), may be indented or otherwise labeled so that you don't try to load them directly. After the last sub-program directory entry, either enter a blank line, **"- - - - -"**, or start another empty filename for the next series of programs on the disk.

Leave room at the top of the list for a menu, DOS wedge, or your favourite utility. For example, if your first entry is a menu for the whole disk, you can 'autorun' it by typing **LOAD "\*\*\*\*",8:**, then hitting **SHIFT/RUN-STOP**. This adds a professional touch to your programming.

A word of caution: when repeatedly scratching and re-saving programs, you may find that your 1541's DOS has moved the relative position of your files within the directory. Thus on first inspection, your files may have apparently 'disappeared'. You can reorganize the disk if this happens, using the Copy command of the **DOS Wedge**; better yet (at least less confusing), use a file-by-file copier to transfer the files to a new disk. This will also let you reorganize the order of the files on the disk.

Also, don't run your directory listings through a program that alphabetizes your entries if you follow this technique! Your directory entries will then be in *alphabetical* order, but you want them in *program* order! □

## Printing Graphs from the Home Accountant

by Karl Thurber

As furnished by Continental Software, the popular **Home Accountant** program for the Commodore 64 can generate and save a variety of graphs to disk, but you're on your own in figuring how to print out those saved graphs — at least in Version "E" of the program (which I have), and previous versions. Here's one way to print out those fancy household graphs, if you have a printer utility program such as is contained on the **Cardco Printer Utility Disk**:

- Generate the desired graph and **SAVE** it to disk, using the **GRAPHS** option of the **Home Accountant MAIN MENU**.

- Using the **Cardco** disk, load the **Printer Utility** program and select the **Keyed Screen Dump** option from the menu. Choose a memory location from the menu for the dump (Choice 1, \$C800, works well). Indicate your type of printer (I use a Gemini 10X), and indicate whether you want a single or double density print-out (the former is easier on the ribbon). When the **READY** prompt appears, type in the required **SYS** command. Now, you are ready to dump a screen by hitting the **F1** function key on your keyboard.

- Now, load the **Home Accountant GRAPH** module. Note that this is an entirely *separate* program on your disk from the main

## Inexpensive Disk Drive Fans

by Karl Thurber

Would you like to cool your disk drive for under \$10? For over a year I have cooled my 1541s with 'new surplus' muffin fans. Flat and square-shaped, muffin fans are used to reduce the internal temperature of much electronic equipment — just what's needed to make a 1541 run cool.

Muffin fans are often sold used for \$4-6 and as new surplus for \$7-9 through electronics parts houses and surplus vendors, who advertise in the pages of such electronics publications as *Radio-Electronics*. Many of the amateur radio publications, such as *73*, *QST*, *CQ* and *Ham Radio*, also carry ads from such suppliers. Look for the magazines, and send for the catalogs of appropriate vendors. The fans are in plentiful supply; you just need to find out which firms stock them.

Even better, attend a local 'hamfest' or 'flea market' at which used and surplus electronic equipment is sold. Almost always, there are dealers and individuals in attendance who sell such fans. Consult the magazines listed above for news about these events, or, if you know a local ham radio operator, ask him when the next hamfest is scheduled in your area.

Some cautions: be sure you obtain a 110-130 VAC 50/60 cycle fan (other fans look the same); and try to obtain the special little connector plug that mates with the one on the fan. Obtain

a plastic or metal grill when you buy your fan, if available, to prevent finger-damage! Finally, install four small rubber mounting feet (the sticky-back kind is best) on the fan to keep it from marring the top of your disk drive when it rests on the drive. Arrange the airflow to blow in or out — it's your choice!

## Saving Disk Directories

by Karl Thurber

If you don't own a printer, but would like to have a record of early versions of a particular disk's directory, you can actually load in the directory and immediately save it to disk, where it is stored just like any BASIC program.

For example, if you have a particular disk that you're about to rearrange, and would like to be able to recall the particular programs that were on it, and their order in the directory, before working on the disk, do this: **LOAD "\$",8** to bring the disk's directory into the computer's memory. Add any comments you'd like to have saved, such as the date (don't worry much about syntax, since this is a non-runnable program). Save what you have to another disk, giving the saved directory a convenient filename. The saved directory may be recalled at a future date, should you wish to know what the directory looked like at a certain time.

In experimenting with this procedure, I found that I could not save the directory under the reserved filename "\$". However, "DIRECTORY" or "CATALOG", or filenames with these words in them, would work okay. □

# OS/9 Software NOW AVAILABLE!

## Word Processor — Stylograph III

Mail merge, Spell checker. Menu driven, fast. Extensive features. Formats to screen as you work. . . . . US \$149 Cdn \$210

## C Compiler

A full implementation of K & R standard C, with double precision variables. . . . . US \$138 Cdn \$170

## BASIC09

A fully structured, sophisticated semi-compiled (I code) BASIC. See Byte magazine April 1984. . US \$110 Cdn \$155

## FORTRAN 77

Full FORTRAN compiler...coming soon.  
. . . . . US \$120 Cdn \$166

## DynaStar, DynaForm, DynaSpell

A full screen (similar to microEDIT) editor, text formatter and spell checker. . . . . US \$90 Cdn \$127

Other: Pascal, DynaCalc, Database etc., available on demand.

To Order: Send a cheque or money order (add \$10 for shipping/handling and 7% PST where applicable) to:

TPUG,  
101 Duncan Mill Rd., Suite G-7,  
Don Mills, Ontario, Canada M3B 1Z3

# Reviews

## Movie Maker from Electronic Arts Animation program for Commodore 64

### Review by Mike Martin

Electronic Arts has just rereleased a program called **Movie Maker**. This animation program came out several years ago, at much higher cost — almost double that of this version. The new incarnation includes more material, in the form of some animation by the bizarre cartoonist Gahan Wilson.

**Movie Maker** comes on two double-sided disks with a manual written for Commodore, Apple, and Atari versions. The manual begins with a tour of the 'studio' and descriptions of production terms and commands. At first glance, the program seems *very* complicated, but each section involves just a few commands, and is easily mastered.

The program allows the creation of 'movies' with fairly detailed animation, but with many restrictions. Three hundred 'frames' of animation are permitted. Standard theatrical movies run at 24 frames, or individual pictures per second. At that rate, these 'movies' would be little longer than 13 seconds. **Movie Maker** permits the selection of nine speeds, so the production can run longer, depending on the detail desired. Each film consists of up to six video tracks, with a maximum of 16 sequences of animation, and three sound tracks.

You start by making your drawings, in the Composing Room. You have two screens resident. One is the Background; the other contains all the drawings you plan to use. You are permitted four, and *only* four colours. This is limiting, as the same four colours (including the background) must be used all the way through. You may choose from all the usual 16 colours. If you draw a blue sky, and green grass, you then have two colours left for people or buildings. The graphics editor is limited and slow. After drawing your first figure, you 'outline' it, or define it so the computer knows where it is. You then duplicate it right next to

the first drawing, and change the position of an arm or leg. The process continues until you have all the positions you need. The editor allows change of pen colour, fill, erase, duplicate, and mirror, but not 'plot' or 'shapes'. By loading an 'Xtra Function File', you may add text or change the scale (size) of a drawing or text. While drawing is difficult, most figures only require three or four positions to animate walking or other movements. The figures are on the rough side.

After completing the drawings, you then define the sequences. For my first project, I drew a monitor with 'TPUG' in large letters, and 'READY' in small letters with a flashing cursor. The cursor turns into a butterfly and flutters off the screen. My first sequence was composed of a small square alternating with a blank frame. I set the speed at four, and let it cycle between those two drawings. I followed this with a four-frame sequence of the butterfly transformation, followed by a four-frame sequence of the wings fully extended, then half-folded, and fully-folded, then back to half-folded. This sequence cycled over and over. After positioning the 'cursor', I recorded a few seconds of it flashing. Then when the transformation sequence finished, I used the joystick to define the movement of the butterfly around the screen. One command allows the choice of how many pixels each figure moves at a time.

After recording the animation music, an 'Xtra' file is loaded for sound. One file is just sound effects. Number keys 1 through 9 are used to call up individual effects, and 0 silences them. The other 'Xtra' file is for music, but has only nine notes, with no provisions for volume, or defining the notes in any way. 'Limited' may be too kind a word for the musical capabilities of **Movie Maker**.

With the **background, shape, animation, and sound** files in memory, you can choose the **smooth** function, which converts the files into a completed production. It evens out the animation, and creates a final file. The **smooth** function permits the addition of credits at the head and tail of your feature. They scroll neatly by.

Text may also be used as a sequence within the film, and manipulated on screen along with your drawings.

Several files can be transferred to your

data disk to run your movie without **Movie Maker** loaded into your computer. These files include a number countdown, and several frames of credits for the company that made the program.

The manual presents several ideas for the use of your movies, but the best one involves making titles and credits for your videotapes. Also suggested is the creation of animated 'greeting cards' for your friends. The program is fun to use, and will provide hours of entertainment for children. (My 'Butterfly' classic took three hours to complete.) It might even inspire a few new-generation Walt Disneys. It won't, however, put the National Film Board out of business. □

## Printer Book For The Commodore 64

by R. Bruckmann,  
K. Gerits and T. Wiens  
Abacus Software, 1984  
350 pages, \$19.95 (US)

### Review by Ranjan Bose

I have a 1526 printer, and after buying this book my first reaction was — why did I bother? It contains only a few pages about my printer, and not all the information is accurate. Having purchased the book, though, I read it anyway, and discovered that it is, all told, a fairly good source of general information about different printers and interfaces. It also shows how to hook up an electronic typewriter to your 64.

The 1525/MPS 801 printers and the 1520 plotter get the lion's share of the attention. For owners of the latter machine, this book is the only source of vital information and utility programs that I know of. Useful hardcopy and high resolution screen dump routines for several printers are provided (including the odd 1526). A useful routine for printing formatted-tabular printouts is there as well. A complete description and memory map of the MPS-801 operating system is included. The optional accompanying disk (\$14.95 US) has all the programs and subroutines from the book and is a worthwhile purchase. □



# Print Master

## Review by

### Mike Martin



#### Review by Mike Martin

As I moved towards the back of the store, it caught my eye. I approached cautiously, and glanced at the box out of the corner of my eye. I picked it up, sniffed, and then shook it. After reading the box five or ten times more, I bought it.

After all, **Print Shop** by Broderbund was my first love. It has faithfully served me for over a year, and even provided more disks of graphics, with the promise of a companion disk of more borders, fonts, and a graphics editor with class. Could I be unfaithful to **Print Shop** after all this?

The similarities between the two packages are striking. **Print Shop** was first released for the Apple computers, then translated for Commodore, Atari, and finally IBM. **Print Master** by Unison World was written for the IBM, then translated for Commodore, Epson, and a CP/M version for Kaypro and compatibles. **Print Master** appears to be heavily inspired by **Print Shop**. I wasn't expecting much other than a slightly different version of **Print Shop**. But its pale yellow box contained some large surprises.

I will have to rely heavily upon comparisons between the two programs. If you are one of the two people in the universe who do not own **Print Shop**, the decision won't be all that difficult. If you *do* own **Print Shop**, you may agonize for hours before adding **Print Master** to your graphics collection.

The package contains your disk, and a manual that is functional, but not flashy. Also enclosed are your registration card, and several offers including an additional graphics disk, a book on graphics design,

and a backup disk for five dollars (US). At five dollars, I consider a backup disk to be a much needed service to the customer; anything over ten dollars I consider to be an exploitive rip-off.

**Print Master's** heavy-handed copy protection will teach your disk drive to dance and sing. That steady 'load' light I've come to expect, quickly became a flashing strobe light. My 1571 danced and flashed for a few minutes, and I wonder what the 1541 would have done.

At first glance, the program resembled **Print Shop**, although the first display was rather obviously designed for an 80 column monitor. The title screen was not readable, but the following menus were just fine. The screen display is black and white and offers the choices of Greeting Card, Sign, Stationery, Calendar, Banner, and Graphics Editor. Printer Setup and Exit to BASIC are also included.

Operation of the program is almost identical to **Print Shop**. A border is selected from the menu by highlighting the choices. Each choice displays a small section of the border, rather than the whole thing. The borders are very different than those on **Print Shop**. They include thick and thin lines, a simulated wood frame, a paper with a 'drop shadow' 3-D effect, nouveau-art deco, cars, footsteps, 'bang', lace, ants, and 'christmas'.

A graphic is selected from the screen listing of twelve titles, plus 'next' and 'previous' pages. But in addition to the list of names, you are given the option of *seeing* each of the graphics listed on the page. A chunk of twelve are loaded at one clip; then by moving the cursor 'highlight' up and down, each art cut is displayed on the screen. I consider that a big plus.

After selection of an art cut, you are given your choice of three sizes, and various positions to place them, including staggered and 'tiled' as a background pattern. The cursor is used to move up or down on custom placement, and each position is marked by a dotted line that revolves much like the flashing lights on a movie marquee.

Next, the type style is chosen. Eight fonts are offered, and I like them better than those offered by **Print Shop**. I find that I only use two or three of the fonts on **Print Shop**, as some are too large or too ornate. The **Print Master** type fonts are more functional, and include both upper and lower case letters. That alone makes the package worth purchasing, as far as I'm concerned. It means that you can fit in longer lines, but provide an all capital headline if desired. The spacing is proportional, allowing less space for an

'i' than an 'o'. In most formats, the fonts are available in two sizes, and also in your choice of solid, outline, or 3-D forms.

When you start to print your creation, the next couple of surprises hit. The program moves the commands to a small window, then opens another window and draws the completed page on your screen. You are then given the chance to go back and make corrections if needed, and are offered the option of saving the page to disk. What a *nifty* idea! On the greeting card section, it draws the front of the card, then moves that window to the corner, and draws another window to create the inside of the card. You may edit and correct a design without having to reload graphics, as in the **Print Shop** program.

The program supports Okidata 82A, 92, and 192; Okimate 10; Epson RX, FX, and MX-Graftrax; Star Micronics 10X/15X; C. Itoh 8510; Commodore VIC-1525, and MPS-801. The box lists Cardeo and TurboPrint/GT interfaces, but my Xetec graphics Sr. worked as well. Print quality does not seem to be quite as high as on **Print Shop**, because the occasional white lines between printhead passes are a bit more noticeable on my Gemini 10X.

The Calendar option prints either monthly or weekly versions. You select month and year, and you may include a graphic, alter the heading a bit, and put some text inside an individual day (but only three lines of eight characters). The Stationery option is similar to **Print Shop**, but allows two lines of text or one large line, and no address lines.

The Graphics editor is similar also, and just as limited and frustrating. Draw with a joystick, and swap the black and white sections, plus flip the picture left to right, or flip it upside down, and save it to disk. The file created is three blocks long, and will load into **Print Shop**, but is read differently, so it won't be recognizable as anything. It resembles the sideways picture on a TV set caused by misadjusting the vertical hold control. But don't lose hope. Some programmer will soon come up with a translation program to swap artwork back and forth. In the meantime, just load up one of the programs and print just the graphics, then use a second pass through the printer with the other program for your borders and text.

Do you need both programs? Probably not. If you can afford just one — which one? Probably **Print Master**.

Will I give up either program? Hell no.

**Print Master**, by Unison World, 2150 Shattuck Ave. #902, Berkeley, CA 94704.



## Typing Tutor III with Letter Invaders

from Simon & Shuster

Typing instruction  
program and game  
for Commodore 64

### Review by Malcolm O'Brien

The first thing you'll appreciate about **Typing Tutor III** is the documentation — a lovely hard cover (!) manual with nice typeset sheets and a *metal* spiral binding. Full marks to Simon & Shuster's Electronic Publishing Division for going the extra mile to provide would-be typists with high quality materials. The disk is inside a plastic 'page' bound into the back of the manual. This is preceded by the End User License Agreement, which is quite similar to the ones you may have seen on the slipcases of IBM software packages. The Agreement gives you permission to "make one copy for back-up purposes provided you reproduce and include the complete copyright notice on the back-up copy". This copy is, of course, for your own use on your own C-64. Do read the Agreement before opening the package.

If you think that I'm talking about serious, professional software here, you're absolutely right! Simon & Shuster (and Borland International, among others) are proving that good software *sells*. And 'good' means, in part, that you *must* have a back-up. This has been taken into account and provided for in the Agreement. Hats off to Simon & Shuster.

Having said that, let's have a look at the programs. **Typing Tutor III** and the **Letter Invaders** game are the creations of Kriya Systems, Inc. The Kriya people have incorporated a very clever idea into **Typing Tutor III**. So clever, in fact, that they have trademarked its name, Time Response Monitoring, and its derived acronym, TRM. According to the manual, TRM "notes the time it takes you to type each letter and then uses this information to create . . . lessons and practice tests that encourage you to respond automatically and without error, because they are specifically tailored to your own, unique requirements".

This facility sets **Typing Tutor III** apart from other typing programs that you may have tried. Since there are many typing programs in the public domain, you may have one or two of them and you may have found (as I did) that they tend to be

boring and do little to relieve the drudgery of typing drills. But **Typing Tutor III** is different. Time Response Monitoring affords the user a higher degree of interactivity which, in turn, leads to a higher degree of user involvement. Your progress is tracked minutely, and your results can be displayed on a bar graph so that you can examine the speed and accuracy of your responses for each and every key. Consequently, you become more acutely aware of your own particular strengths and weaknesses at the keyboard.

The net result of this is great playability, for want of a better word. Every time I run this program I wind up typing for hours. And when I stop, it's not because of boredom or frustration but simply because other tasks require my time and attention. Or because I should have been in bed an hour ago! Kriya is a Sanskrit word whose root meaning is 'do' and this program certainly keeps me 'doing'!

I wanted to find out if others would be affected in the same way by the program, so I corralled a friend into my chair while **Typing Tutor III** was on the screen. She's a competent typist but tries to steer clear of computers. Nevertheless, after a very few words from me (more of encouragement than of instruction), she began to work with the program, following the on-screen instructions. About an hour later, I concluded that **Typing Tutor III** really was as engaging as it seemed to me to be and I told her that she could stop. "Just let me do a few more", she said.

The programs include on-line help and instruction screens that are memory-resident and therefore don't require disk accesses. Of course, this means that they are loaded in every time that you use the program. This is a feature that I really dislike in a program, but I think that I am in the minority in this regard and that most users will appreciate having all the instructions immediately available. Still, I wish that we had a choice about loading the help. **Typing Tutor III** and **Letter Invaders** are long programs (191 and 134 blocks respectively). Fortunately both of them work with **Fast Load**. On the positive side, the information on the screens is clear, concise and complete. You may not need that wonderful manual.

From the main menu you have seven choices: Alphabet Keys, Number Keys, Words Test, Full Keyboard Test, Numbers Test, Standard Speed Test and Status Report. Alphabet Keys is the place to start if you're a neophyte. Number Keys is the place to go if you're one of

those people who still have to look at the keyboard when confronted with digits. Initiates may elect to go directly to Word Test or Numbers Test, and the adepts among you may opt for the Full Keyboard Test or Standard Speed Test (which always shows the same text unfortunately). If you select the word or number tests after completing a lesson, only the keys you've been working on are included.

The status report is another excellent feature of this program. From the status report you can change the program variables. Specifically, you can turn the sound on or off, change the threshold at which a key is considered fast, change the number of lessons that are presented before a Progress Report is displayed, or alter the number of lines in a test. This is great for adjusting the program to adapt to the differing needs of users — from the hunt-and-peck crowd to the speed demons.

The program gives very good reports on your performance, showing which keys have been typed 'fast', which keys have been mistyped, your accuracy (expressed as a percentage), your actual speed and your corrected speed (using the standard method of subtracting five words per minute for each error).

The game, **Letter Invaders**, is more fun than **Typing Tutor III** and offers almost as much performance information. The idea for the game has been used before but this is a first-class implementation. Characters fall from the sky endlessly. Evidently they are not at all intimidated by your land-based lasers, which obliterate the attackers when the correct characters are typed. Any missed characters will decimate the landscape. This is a great challenge and a tremendous responsibility rests on your shoulders. If its safety depended on me, frankly, the earth would be doomed. I'm not terribly fast (50+ wpm) but I have very good accuracy. Still, I was no match for the invaders; although I did manage to pick up a lot of bonus points for triads (three successive keystrokes for one finger). Where I really got creamed was on the special symbols. I use a PC/XT at work all day and then come home and flub all the keys that are in different locations on the 64. Certainly the game taxes your skills, and fairly *demand*s that you become a better typist. You'll do it too. And you'll have fun in the process.

So for your next keyboard meditation, concentrate your mind on **Typing Tutor III** and let Kriya be your guru. It could be a (third) eye-opening experience!

*Typing Tutor III with Letter Invaders, from Simon & Shuster.* □

## Starting FORTH

by Leo Brodie

Prentice Hall  
Softcover, 360 pages  
Price \$20.00 (US)

### Review by R.C. Walker

Written in the same matter-of-fact style as his *Thinking Forth*, Leo Brodie's *Starting FORTH* is an invaluable reference for FORTH programmers at all levels of expertise. Since it is also a tutorial, there are several exercises, with answers, included for each chapter. If you want FORTH, you will also want this book.

Let's look at each chapter in turn:

**Fundamental FORTH:** This chapter examines the properties of the language, and the design of FORTH words. FORTH programming consists of developing new words (commands) from previously defined words. Although FORTH is compiled, it is also interactive. Words can be executed simply by typing them and hitting the **RETURN** key. Application source code is normally saved on disk in blocks of 1024 characters each ('screens'), which are compiled into the system before executing. Alternatively, a completed application can be saved and loaded as a precompiled file. FORTH's parameter stack and postfix mathematics, the sources of much of the language's speed and power, are introduced here.

**How To Get Results:** In this chapter, FORTH mathematics is dealt with more deeply. Floating point words, though sometimes provided as an option, are not dealt with, since integer mathematics are standard (for reasons of speed), and there are methods for overcoming integer shortcomings (dealt with in a later chapter). Stack manipulation words are studied and used.

**The Editor (And Staff):** This chapter provides a description of a simple editor that allows one to create source code. It is written in FORTH, and takes the form of a separate vocabulary, since many words may otherwise be identical to some used in FORTH's main vocabulary. Comments may be freely added, since they are not compiled into the dictionary and, therefore, do not slow down execution time, as in interpreted BASIC.

**Decisions, Decisions:** FORTH, like any other language, uses logical operators for

decision making. These, and FORTH's **IF ELSE THEN** constructs are detailed.

**The Philosophy Of Fixed Point:** The use of integer mathematics is expanded upon in this chapter. In addition to the basic mathematics operators, other handy words are provided. These include the often-used fast operators (to add or subtract 1 or 2, and to multiply or divide by 2), miscellaneous operators, and words to manipulate the computer's return stack. Technical and philosophical aspects of fixed and floating point mathematics are touched upon. FORTH can often overcome the problem of not having floating point through the use of scaling operators and complementary rational approximations, as explained in this chapter.

**Throw It For A Loop:** FORTH's many loop structures are explored. The variations in the **DO LOOP** construct allow for some sophisticated applications.

**A Number Of Kinds Of Numbers:** This chapter explains the various types of numbers (bytes, single-length, double-length, signed, unsigned, ASCII) used by computers, and by FORTH in particular. FORTH's normal single-length numbers are 16 bits in length (allowing a range of over 65,000), while double-length numbers use 32 bits (for a range of over 4 billion). This should allow for more than ample precision, using scaling where appropriate. A useful property of FORTH is that practically any number base may be used without conversion. Binary, decimal, hexadecimal or a non-standard base may be used where appropriate. Logical operators and a relatively simple number formatter are also touched upon in this chapter.

**Variables, Constants and Arrays:** These items are used in FORTH in much the same way as in other languages. The use of constants, in particular, is encouraged, since there is no speed penalty, yet they aid in documenting the source code. Single dimension arrays and their uses are examined. Factoring, a programming development concept especially suitable for FORTH, is introduced here.

**Under The Hood:** This chapter examines the technical aspects of the FORTH language — what makes it run. First up is the interpreter, followed by vectored execution, a very useful programming technique. The dictionary structure shown is not that of all FORTH implementations, although the principles will be similar. The author shows how new definitions are added to the vocabulary. Also, FORTH's use of buf-

fers, the scratch pad, variables and vocabularies is examined.

Most FORTH implementations include an assembler vocabulary but, since each one is patterned after the particular processor for which the system is set up, it is not detailed in this book.

**I/O And You:** Discussed in this chapter is how FORTH handles input and output (mostly character strings to and from disk block buffers), and other data movement. BASIC programmers who have used **FOR NEXT** loops to transfer a block of data or fill an array will appreciate the speed of FORTH's move commands. Manual commands for transferring information between computer and disk drive, often handled automatically as a virtual memory system, are also presented.

**Extending The Compiler: Defining Words And Compiling Words:** FORTH is a compiled language. The compiler modules can easily be added to, as simply as defining a short line of source code. 'Defining words' specify the behaviour of words when they are compiled and when they are executed (interpreted). For example, variables and constants use different compile code. This results in different behaviours when executed. How many languages allow the programmer such powerful methods of extension?

A non-traditional flowcharting method is also briefly discussed in this chapter, in order to more deeply examine two important FORTH words, the interpreter and the colon compiler.

**Three Examples:** Three programming examples are presented, primarily to teach FORTH style. Good style ensures a superior product, no matter what one's endeavour. By putting into practice such elements as simplicity, structure, appropriateness and following FORTH conventions, one will help to ensure that their finished product is elegant.

The first example is a word game that prints randomly-chosen phrases. The program is cleverly set up so that the output seems almost coherent. The second example is a simple database implementation. The third example demonstrates how integer FORTH can handle problems that normally must depend upon floating point mathematics.

Whether your pleasure is learning FORTH, or just examining some aspect of the language, you may be hard pressed to find a better book. It does not matter whether you are a novice or an advanced programmer, as this book is far more than an introduction to FORTH. In my library, it is a valued reference. □

## Trolls and Tribulations

from Creative Software  
Arcade-style game  
for Commodore 64

### Review by Robert Sodaro

*You are a troll. All your life you have lived with darkness, dank mustiness, and the overpowering stench you could cleave with a dull blade. Danger lurks everywhere: carnivorous 'cretins', low-flying buzzards, troll-crushing 'skulls' and poisonous spiders. There are also the ever-present rising waters, rank with raw sewage from the underworld...*

This is the real world of Creative Software's **Trolls and Tribulations**. Games today tend to combine elements of previously-established hits into new patterns, and Creative Software has certainly done this with **Trolls**, producing a unique and humorous blend of two arcade classics: **Tutankhamen** and **Mario Bros.**

From **Mario Bros.**, **Trolls** lifts the single-screen, multilevel maze, where deadly beasties drop from above to hunt you down. In **Trolls**, the troll (equipped with an eight-shot pistol) must first shoot the attacking cretin, then knock the resulting 'egg' into the murky waters below. Coupled with this are a series of **Tutankhamen**-like mazes, where trolls must gather up treasures by traversing narrow corridors and leaping across great chasms.

**Trolls** forces the player to plan each move with care and precision. A light touch and a measure of forethought are required, or your hapless troll will wind up overplaying his hand and tumbling to his doom.

At the outset of the game, you are required to pick the level of play (novice, intermediate or expert) by opening the appropriate door. Your expedition of eight then enters the first set of mazes. The entire maze is actually a multilevel, horizontally-scrolling screen, sectioned off into levels, each fraught with its own particular set of dangers. Cretins (both green and red) can be killed, but skulls, buzzards and spiders cannot.

Locked doors appear at each end of all maze sections (the maze scrolls off to the right), and can only be opened by acquiring the proper keys. Keys (and treasures) are easily obtainable, by either leaping from ledge to ledge or dropping to shelves below your position. And fear not — long

falls will not harm you. (Unless, of course, you land in the water and drown). Strategically-placed trampolines are situated to assist in reaching higher ledges. Leaping skulls and buzzards in the confines of the low tunnels may inhibit you, at first, but no harm will befall you, since you soon discover that you can't hit your head.

The lower levels are easy enough to master but, as you progress deeper in the maze, you will need a greater degree of skill. Higher levels of play at the deepest sections will prove to be a true challenge for hardcore gamers. The progression from the first sections to the deeper and more difficult ones, however, is gradual enough to ensure continued interest as ability increases.

**Trolls** is a delightful game, but I do have two complaints. First, there is no way to bypass higher sections and begin deeper in the maze — you must always start in the first maze of each level, progressing through old territory before entering new ground. Second, a high score is provided, but there is no way to save the information to disk (the instructions speak of 'memory limitations'). Who cares that I can keep track of the high score in a single sitting — I want the *next* fellow booting up the game to know how good I am!

Still, **Trolls and Tribulations** is well worth its weight in mystic amulets.

*Creative Software, 230 East Caribbean Drive, Sunnyvale, California 94089.* □

## Mapping The Commodore 64

by Sheldon Leemon  
COMPUTE! Books, 1984  
268 pages, \$14.95 (US)

### Review by Ranjan Bose

Without denigrating the excellence of the Commodore 64 in a large number of serious applications, as a reasonably-priced yet powerful microcomputer, it is also ideal for learning how to program. The C-64 has excellent graphics and sound capabilities, but if you really want to make it run for the money, you have to learn about its logical organization, and you have to dabble in machine language.

Over the years, a lot of literature has accumulated concerning the various aspects of Commodore computers, especially the C-64. *Mapping The Commodore 64* is very well organized, and is

one of the most glitch-free reference manuals available. It describes all the memory locations in the C-64 operating system in reasonable detail, and also includes subroutines for performing many different programming feats. The book is a must for understanding what each register does, and how the operating system utilizes the different memory addresses and banks to perform programming magic.

One tends to shy away from reading about memory maps, because most of them make rather dry reading, with funny names for addresses and a presentation that is unavoidably disjointed and terse. *Mapping The Commodore 64* is different. It is very lucid, and actually makes rather interesting reading. I just wore out my first copy after several months of extensive handling, and am rushing to get another one before my programming comes to a standstill. □

## Graphics Book For The Commodore 64

by Axel Plenge

Abacus Software, 1984  
280 pages, \$19.95 (US)

### Review by Ranjan Bose

Sooner or later, you have to grow out of conservative BASIC programming, and start venturing into the other avenues the C-64 makes available. Sound and graphics programming are available from BASIC, but the slowness of BASIC routines soon brings you to the menu of options. These are: stay back if you are a chicken; buy an extension package, but forget about exchanging or selling your programs; or learn the nuts and bolts (machine language routines, handling VIC/SID routines, registers and interrupts); and lastly, buy a C-128 or Amiga.

For the nuts and bolts handyman, this book from Abacus is a great help. By the time you finish reading it, you will understand how to move screens, design character sets and sprites, and the mechanics of interrupt programming. Using the programs provided (sprite editor, character set editor, high-res routines, and so on), which are also available on an optional accompanying disk (\$14.95 US), and after studying the book several times, programming graphics will hold no terrors for you. From then on, your dedication and creative imagination will determine how far you go. Highly recommended. □

## Superman — The Game

from First Star Software

Arcade game for  
one or two players  
for Commodore 64

### Review by Malcolm O'Brien

I started reading before I went to school, and *Superman* was probably the first publication that I read regularly. Over the years, wherever Superman went, I followed — the comics, the television show and the movies. I've been a Superman fan for over thirty years. Since Superman is not bound by time or space, it seems perfectly natural that he should now appear on my monitor screen as well.

My children, Grace (10) and Sean (9), are big Superman fans too. Thanks to the miracle of videocassettes, they have seen the movies numerous times — occasionally watching the same film three times in a weekend! They both got very excited when I told them I would be reviewing the game for *TPUG Magazine*. So we all gathered around the C-64 on a Saturday afternoon (and evening!) to put the Man of Steel through his paces.

**Superman — The Game** is the first in a new series of programs from First Star Software. The series is called the 'Super Powers Software Collection', and I expect that it will be well received by game players at large. This is an arcade-style action game, although there are strategic aspects as well.

As with other offerings from First Star Software, the program's greatest strength is in its graphics. The opening screens are superb. When the program appears, Superman flies over the Metropolis skyline and pauses to wave to you. Then an excellent portrait of Supes 'dissolves' onto the screen, accompanied by suitable music. This gives way to a portrait of the evil villain, Darkseid, and the music becomes sinister. Real scary, kids! The last screen in the opening has the famous Superman logo and the program credits. The program was written by Jim Nangano and Fernando Herrera.

Mr. Herrera has made quite a name for himself as a creator of microcomputer graphics. A few years ago, Atari ran a contest for user-written programs, which was won by Mr. Herrera. His winning program, **My First Alphabet**, was written for his young son. Although it has no

educational value for anyone older than the Sesame Street set, the excellent graphics and the overall quality of the program were probably what caught the eye of the Atari judges. Mr. Herrera's efforts won him the first (only?) Atari Star award, and he then started First Star Software. He's been keeping the VIC II chip (not to mention the Pokey chip in the Atari) busy ever since.

**Superman** is no exception. Game action is very fast and very complex, graphically and strategically. You have to keep track of a number of game parameters to tip the scales in your favour. Here's the setting:

There are six Metropolis game screens. Three of them represent the streets of Metropolis, and the other three represent Darkseid's network of caves beneath Metropolis. When moving between screens, the two combatants do battle on the ground, under the ground, in the air, and in outer space. These transitional screens, called 'Combat Zones', can be very challenging exercises in themselves — the documentation describes them as "games within The Game". The Combat Zones are, once again, very appealing graphically, and a little bit easier to get a handle on than the Metropolis screens. At the beginning of the game, you can elect to play with or without Combat Zones. When you're new to The Game you may find that including the Combat Zones is the only way you can exercise a measure of control over the game! Although some points are won or lost in the Zones, it's the Metropolis screens that will make or break you.

Effectively, the six Metropolis screens constitute one large maze, which is populated by the citizens of Metropolis who will be captured or rescued during the course of the game. The citizens wander zombie-like through the streets and caves. They will blithely walk into Darkseid's prison without a second thought. The citizens are none too bright, and really *need* Superman's help! Each person rescued adds to Superman's score, and each one captured adds to Darkseid's. Actually, the score has three components — points, energy remaining, and number of people rescued or captured. Two of these, energy and people, can decide the contest. Whoever runs out of energy first loses. In the beginning, this will probably be you! The game can also conclude when all the citizens have been captured or rescued. In this case, the player with the highest people score is the winner. With time, as you improve, you may win on energy; later on, by conserving energy and playing strategical-

ly, you'll start to win games on people.

Playing the game effectively consists chiefly of knowing which of the four main game activities to perform at any given time. The four main actions are: assaulting your opponent directly, transporting citizens, rotating deflectors, and obtaining dimensionizers (used for travelling from screen to screen). Assaulting your opponent can be done bodily or by means of beams (Superman's heat vision or Darkseid's Omega beam). The deflectors serve the dual purpose of redirecting both the beams and the citizens of Metropolis.

At first I felt that the game was too hard. Darkseid seemed to be too smart and too fast. Strangely, he seems to be able to walk faster than Superman can fly! Sometimes, when first entering a new screen, Darkseid and Superman appear in the same spot. This 'freezes' you, which results in a serious energy loss, and you are helpless until Darkseid decides to move away. To be fair, you can sometimes use this same technique against your opponent: either to 'buy time' when the citizens are moving your way, or simply to deplete your opponent's energy reserves. While this ploy enhances the game strategically, it also diminishes the action. Although different players will react to this in different ways, there were times when I found it downright frustrating!

Aside from this minor complaint, the program has one serious flaw that I would like to see corrected. At the end of the game, when playing against the computer, the winner is seen flying triumphantly over the Metropolis skyline. In the two-player mode, however, the game ends abruptly and returns you to the 'Selections' board with no indication of who has won the game! David Bradley and I played the game for several hours one evening, and this happened to us several times. On one occasion we were returned to the 'Selections' board before either of had bothered to check our scores for the first time! On another occasion the Metropolis map showed that there were still unaccounted-for citizens; but we couldn't find anyone on any of the six screens!

In spite of these problems, which mar an otherwise excellent game, I would still rate the program highly. It will provide users with many hours of fun and entertainment.

*Superman — The Game*, from First Star Software, 18 East 41st Street, New York, New York 10017. □

# Products Received

## Presented by Astrid Kumas

The following products have been received by TPUG Magazine in recent weeks. Please note that these descriptions are based on the manufacturers' own announcements, and are not the result of evaluation by TPUG Magazine.

## Programs for the C-64/128

COMPUTE!'s Commodore 64/128 Collection from COMPUTE! Publications, Inc. 324 W. Wendover Avenue, Suite

200, Greensboro, NC 27408, (919) 275-9809. Price: \$12.95 (US).

Hard on the heels of the C-128 release, COMPUTE! Publications Inc. came up with a new title, *COMPUTE!'s Commodore 64/128 Collection*. The book, on sale since June 1985, contains ready-to-type-in programs for the C-64 and C-128 in its 64 mode.

Besides the programs, the book includes information on the Commodore 128. The opening chapter, entitled *Inside the 128*, describes new features of the

C-128: CP/M, BASIC 7.0, 128 mode and peripheral ports.

There is also a chapter on programming: it offers hints on debugging BASIC programs and suggestions for writing text-adventure games.

As for the programs presented in the book, they range from games, sound and graphics applications to programming utilities, and have been grouped accordingly in three sections: *Recreation and Education, Sound and Graphics, Utilities*.

## Still Sizzling...

### News from Cardco

At the January 1986 CES in Las Vegas, Cardco exhibited its two **StealthTec** utilities, described as the first across-the-board transparent program interrupt technology for Commodore computers.

A transparent interrupt mechanism is one which can be called upon during the execution of another program. It will then 'disappear' when not needed, leaving the previous program intact, exactly as it was when the transparent interrupt was called upon.

**Freeze Frame**, the first **StealthTec** program, is a totally transparent screen dump utility. With a couple of keystrokes, it will immediately send whatever is on the screen right to the computer's printer. **Freeze Frame** is unique for two reasons: it works on virtually *all* programs and *all* programming languages, and it supports *any* printer or printer interface which emulates the Commodore 1525, as well as *any* Epson-compatible or *any* Okidata-compatible printer.

Second program in the **StealthTec** series is a business/productivity utility, yet unnamed. It will provide instant access to a calculator, an appointment calendar, a telephone directory/database, a memo writer, a screen dump utility, access to all DOS functions, and more.

Cardco will produce the initial **StealthTec** programs for the Com-

modore 64. Versions for the Commodore 128 will follow soon. For more information on **StealthTec** programs contact Cardco, Inc. 300 S. Topeka, Wichita, KS 67202.

Cardco announces also an affordable hard disk drive for the Commodore 64/128 computers. The company claims that its Hard Disk Drive is virtually transparent to all programs, and offers the following features:

- choice of three memory capacities: 5 megabytes, 10 megabytes, or 20 megabytes;
- high-speed operation (50 to 70 times the speed of the Commodore 1541 disk drive);
- expanded Commodore DOS commands for greater programming strength;
- total transfer of copy-protected programs.

Cardco's projected retail prices for its Hard Disk Drive will be \$599.95 (US) for the 5-megabyte model, \$899.95 (US) for the 10-megabyte, and \$1,299.95 (US) for the 20-megabyte version. Cardco plans that deliveries of the C-64 models will begin late in the first quarter, with the C-128 models to follow shortly.

### New Amiga Products

Brown-Wagh Publishing is now shipping three new productivity tools including a spreadsheet, telecommunication and bulletin board program for

the Amiga personal computer.

The spreadsheet program, called **Analyze!**, can be used for financial analysis and planning, bookkeeping, home budgets, check registers and more. Loaded with features similar to **Lotus 1-2-3**, **Analyze!** can produce professional-sized spreadsheets (256 columns by 8,156 rows).

**OnLine!** is a telecommunication program that transforms the Amiga into a terminal that is capable of interacting with micros and mainframes. The program can also link up with commercial information services, as well as send telex messages and electronic mail worldwide.

Last program, **BBS-PC**, is a versatile electronic bulletin board system that transforms any Amiga into an online information network. It interfaces to a hard disk or keeps up with a 2400bps modem.

All three programs were written by Micro-Systems Software Inc., and are being distributed by Softeam (1-800-847-6383), National Software Distributors (800-821-3612) and Computer Software Services (800-422-4912) in the U.S., and by Phase 4 Distributors (403-252-0911) in Canada.

**Analyze!** and **BBS-PC** retail for \$99.95 (US), while **OnLine!** retails for \$69.95 (US). For additional product information contact: Brown-Wagh Publishing, 100 Verona Court, Los Gatos, CA 95030 or call 408-395-3838. □

To make program entry easier and error-free, the publishers included in the *Appendices* general guidelines on typing in program listings, an error-checking program called **Automatic Proofreader**, and the machine language entry program **MLX**. However, some readers will certainly appreciate the convenience of buying with all the programs found in the book. The cost of the disk is \$12.95 (US), and the toll free number to call and order it, is 800-334-0868.

**Assembly Language:  
C-128 Update**

*Assembly Language for Kids: Commodore 64 with Commodore 128 Update* by William B. Sanders, published by Microcomscribe, 8982 Stimson Ct., San Diego, CA 92129. Price: \$14.95 (US).

William Sanders' book was reviewed in the August/September 85' issue of *TPUG Magazine*, and was highly recommended by our reviewer Michael Quigley. Since then, Microcomscribe has sent us an updated version of its publication — *Assembly Language for Kids: Commodore 64 with Commodore 128 Update*.

The *Update* comes as a separate 29-page booklet, neatly fitting inside the main book. It contains the book's programs all reconfigured for programming in the Commodore 128 Mode using the built-in mini-assembler. In addition, a C-128 memory map, new BASIC 7.0 tokens, C-128 Sprite Assembler with the new addresses for sprite storage, information on switching memory banks, as well as instructions in using the monitor and mini-assembler have all been included.

It is also worth mentioning that the *Update* lists in the new *Errata* a number of errors which slipped into the original printing of the book.

**C-64/128 Portfolio Manager**

**Personal Portfolio Manager for the Commodore 64 and 128** from Abacus Software, Inc., P.O. Box 7211, Grand Rapids, MI 49510. Price: \$39.95 (US).

This new product from Abacus Software has been designed for investors who need a portfolio management and analysis system. The program runs on the Commodore 64, SX-64 and Commodore 128 in its 64 mode.

**Personal Portfolio Manager** can be used as a tool in three main areas of personal investment:

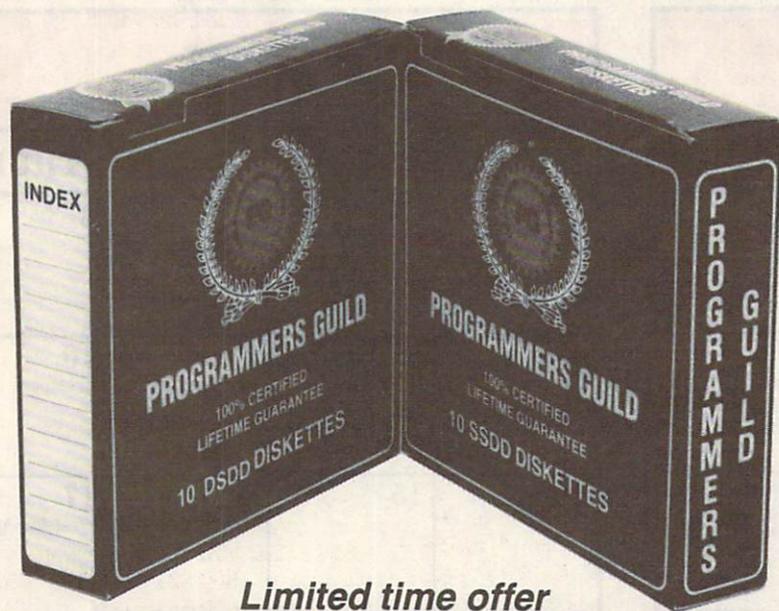
- Portfolio Management and Analysis — to compile various transactions in a portfolio and to report its performance;
- Fundamental Analysis — to select successful investments according to the users' particular criteria;
- Technical Analysis — to study past performance of a security in order to predict its future.

The program will handle up to one thousand open transactions on a single diskette.

An interesting feature of the **Personal Portfolio Manager** is that the quotes can be entered manually or automatically through Dow Jones or Warner Computer Systems. For this purpose a Vicmodem, Commodore 1650, or a 1660 modem can be used.

The user has also an Autorun feature which enables the user to set time for the system to log on, update quotes, log off, and print reports.

**Personal Portfolio Manager** comes with a manual divided into the following sections: *Getting Started, Tutorial, Reference* and *Appendices*. □



**Limited time offer**

**\$11.00**  
**\$14.00**

(Minimum purchase - 5. Delivery extra.)

# Switch

**to the highest  
quality, lowest  
priced brand name  
diskettes available  
in Canada.**

- Q.** Who has switched?  
**A.** Professional and educational institutions, national accounting firms, major engineering firms, software houses, computer retailers and the Toronto Pet Users Group.

**In Bulk - 3 1/2 inch discs**  
**SSDD \$3.50 each / DSDD \$4.00 each**



**Programmers Guild Products Ltd.**, 255 ManuLife Place, 10180 - 101 St.,  
Edmonton, Alberta T5J 3S4 Phone: (403) 428-6229



# Calendar of TPUG Events

## Meeting Places

**Amiga Chapter:** The new Amiga chapter meetings will be held in conjunction with the Westside chapter meetings.

**Brampton Chapter:** Brampton Public Library, Four Corners Branch, 65 Queen St., on the second Thursday of the month, at 7:30 pm.

**Business Chapter:** TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the second Wednesday of the month, at 7:30 pm.

**Central Chapter:** The Central Chapter will no longer be meeting.

**COMAL Chapter:** York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the fourth Thursday of the month, at 7:30 pm in the Story Hour Room (adjacent to the auditorium).

**Commodore 128 Chapter:** TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the first Wednesday of the month, at 7:30 pm.

**Commodore 64 Chapter:** York Mills CI, 490 York Mills Rd. (east of Bayview) on the last Monday of the month, at 7:30 pm in the cafeteria.

**Communications Chapter:** TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the fourth Wednesday of the month, at 7:30 pm.

**Eastside Chapter:** Dunbarton High School (go north on Whites Rd. from the traffic lights at Highway 2 and Whites Rd. to next traffic lights; turn left to parking lots) on the first Monday of

the month, at 7:30 pm.

**Hardware Chapter:** TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the second Tuesday of the month, at 7:30 pm.

**New Users Chapter:** TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the third Monday of the month, at 7 pm.

**SuperPET Chapter:** York University, Petrie Science Building (check in room 340). Use north door of Petrie to access building. On the third Wednesday of the month, at 7:30 pm.

**VIC 20 Chapter:** York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin), on the first Tuesday of the month, at 7:30 pm in the auditorium.

**Westside Chapter:** Clarkson Secondary School, Bromsgrove just east of Winston Churchill Blvd., on the third Thursday of the month, at 7:30 pm.

**68000 Chapter:** The meetings of the new 68000 chapter will be held in conjunction with the meetings of the SuperPET chapter.

*TPUG makes every effort to ensure that meetings take place when and where scheduled. However, unforeseen problems may occasionally arise that lead to a particular meeting being changed or cancelled. The TPUG meetings line (445-9040) is the best source of fully up-to-date information on meeting times, and should be consulted.*

*Are you interested in organizing some other interest group in the Greater Toronto area? Please let the club office know, by mail, phone or TPUG bulletin board.*

MARCH			
MON	TUES	WED	THURS
3 Eastside	4 VIC 20	5 Commodore 128	6
10	11 Hardware	12 Business	13 Brampton
17 New Users	18	19 SuperPET/68000	20 Westside/Amiga
24 Commodore 64	25	26 Communications	27 COMAL
31			

APRIL			
MON	TUES	WED	THURS
	1 VIC 20	2 Commodore 128	3
7 Eastside	8 Hardware	9 Business	10 Brampton
14	15	16 SuperPET/68000	17 Westside/Amiga
21 New Users	22	23 Communications	24 COMAL
28 Commodore 64	29	30	

# Bulletin Board

## Richvale reborn

**Richvale Telecommunications**, was one of the pioneer developers of Commodore software and hardware in the Toronto area. It bore the very individualistic stamp of its owner, Peter Smith, and was locally famous for both its creative ferment and its chaotic approach to business — which may, after all, have been two sides of the same coin. Among their better-known products were a word processor, **Script 64**; a copying utility, **Supercopy**; and an IEEE interface for the 64, the C-Link.

Richvale went out of business a while ago, after a long, unsuccessful struggle to stay afloat, and their passing seemed like the end of an era (eras being shorter these days than they used to be).

But a few weeks ago, Smith's wife, Barbara, announced the formation of a new firm, Rich-Hill Telecom International, to take over where Richvale left off. Rich-Hill will be marketing the same hardware and software line as Richvale.

Rich-Hill will be kicking things off with an open-house and sale on March 14th (evening) and 15th, which we hope is some time *after* you read this. Peter and Barbara invite old customers, well-wishers and other interested parties to join them for the event in their headquarters at 200 West Beaver Creek Road, Richmond Hill, in the Beaver Park industrial park (near highways 7 and 404). For additional info, call 731-1621.

## Racter replicated

**Our prayers have been heard**, it seems, for **Racter** has added a new persona to its particular form of Artificial Insanity. **Racter's** legal guardian, Mindscape Inc., has allowed the silicon scribe to enter the mind of the Amiga. Our human brains boggle at what this prolix program will do with multitasking.

## Distressed? Call this number . . .

**David Bradley** of the Bradley Brothers' Bulletin Board System has announced a new 'freeware' service. The Distressed Commodore Users' Hotline is for those having difficulties with their new machines. Business hours are 2:00 pm to 10:00 pm, Monday to Friday at (416) 488-4776. For further information about the service, write to Distressed Commodore Users' Hotline, 147 Roe Avenue, Toronto, Ontario, Canada M5M 2H8. Those with modems can also call any of the Bradley Brothers' Bulletin Board Systems at (416) 487-5833, 481-8661, 481-9047, 277-9991, 24 hours a day, 7 days a week.

## A mention . . .

**Robert Adler**, of the C-64 Users Group of Canada, is mad at us because we keep forgetting to mention the existence of his club. The club produces an excellent newsletter, *Hardcopy*, and operates both a Help Hotline, at (514) 739-3046, 10 am to 10 pm daily; and a BBS called Micropolis, at (514) 739-3446. The address is Snowdon P.O. Box 1205, Montreal, Quebec, H3X 3Y3. Peace is restored. ☐

shoebox BUSINESS SOFTWARE



DIRECT FROM THE AUTHORS WITH A 30 DAY SATISFACTION GUARANTEE PLUS CONTINUING SOFTWARE SUPPORT

## shoebox II ACCOUNTING PACKAGE

FAST, USER FRIENDLY WITH HELP KEYS, SOUND CUES AND AN EXTENSIVE MANUAL • USE WITH C-64 OR C-128 AND COMPATIBLE DISK DRIVE AND PRINTER • 97 GENERAL ACCOUNTS • FULL FINANCIAL STATEMENTS • INVOICING • ADDRESS DIRECTORY • CALCULATES TAX, DISCOUNTS, CCA, SHARED EXPENSES AND PREPAYMENTS.



Qty. \_\_\_\_\_ x \$89.95 \$ \_\_\_\_\_  
Shipping & Handling \$ \_\_\_\_\_ 5.00  
7% Sales Tax  
(Ont. residents only) \$ \_\_\_\_\_  
**TOTAL \$ \_\_\_\_\_**

DEALER INQUIRIES WELCOME

Prepaid Orders only

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ Prov. \_\_\_\_\_ Code \_\_\_\_\_

JOHN DUNLOP & ASSOCIATES LTD.  
R.R. #5 ORANGEVILLE, ONTARIO L9W 2Z2 (519) 941-9572



CHARGE MY VISA OR MASTERCARD NUMBER

EXPIRATION DATE

\_\_\_\_\_

# Classifieds

This space is for the ads of TPUG members. Wanted or for sale items only. Cost is 25 cents per word. No dealer ads accepted.

## Unclassifieds

**For Sale:** 8032, 4040, ADA 1450, Wordpro 4+, InfoPro, all for \$500.00 (US) or best offer. Call John (201) 263-4538 (evenings).

**For Sale:** SuperPET, 8050 drive, modem, with software — OS/9, Stylograph with Spell Check and Mail Merge, C Language, 7 Waterloo interpreters, Datamanager, BPI General Ledger and Job Costing Module, VisiCalc, WordPro 4 Plus. \$2,000.00 or best offer. Call Neil Stone, (416) 440-0368, business hours.

**For Sale:** 4032-N CPU \$300.00, 4040 Drive \$650.00, 4023 Printer \$200.00. \$850.00 for whole system. All units with documentation. R. Hyam, 225 Cambie Road, Kelowna, British Columbia, V1X 3H9, (604) 765-5215.

**For Sale:** SuperPET system, including 8250 disk drive, 2023 printer, WordPro 4 Plus, VisiCalc, other software and manuals. \$2000.00 complete, or best offer on system of component(s). Gordon Ritchie, (416) 626-5442.

## Ask Someone Who Knows

If you enjoy **Jim Strasma's** many books, and his articles in this and other magazines, you'll be glad he also edits his own highly-acclaimed computer magazine, now in its sixth year of continuous publication. Written just for owners of Commodore's many computers, each **Midnite Software Gazette** contains hundreds of brief, honest reviews.

**Midnite** also features timely Commodore news, hints and articles, all organized for instant reference, and never a wasted word. Whether you are just beginning or a long-time hobbyist, each issue will help you and your computer to work together effectively.

A six issue annual subscription is \$23. To subscribe, or request a sample issue, just write:

**MIDNITE SOFTWARE GAZETTE**  
P.O. Box 1747  
Champaign, IL 61820

**You'll be glad you did!**

## DON'T FEAR THE TAX FORMS!

If you live in Ontario, TAX-85 is one of the most comprehensive, PERSONAL Income Tax programs ever written for Commodore systems. If you are confused by: T4s, T5s and the many other slips, tax rules or tax forms, then TAX-85 is just for you!

Tax-85 asks VERY SIMPLE QUESTIONS and then makes all the decisions to give your MAXIMUM REFUND. It produces completed tax forms, on screen or printer, that resemble the original. You simply copy the numbers to the pink government forms.

Send \$23.75 (includes tax and postage) to:  
-Specify CBM or C64-

R.W. Dray & Co  
P.O. BOX 1025  
PETERBOROUGH, ONT.  
K9J 7A5

# Super-OS/9\* Is Here

TPUG has implemented the popular 6809 operating system OS-9\* on the SuperPET. Super-OS/9 greatly expands the software availability and the hardware capability of the SuperPET while preserving access to the Waterloo languages and programs.

The cost of Super-OS/9 to club members is \$210 (Cdn) (plus \$10 shipment/handling Ontario residents add 7% PST), which includes the cost of a hardware modification that will not affect the normal operation of your SuperPET, installation instructions and the operating system disks.

To obtain your copy please send your cheque or money order to:

TPUG  
101 Duncan Mill Rd., Suite G-7  
Don Mills, Ontario  
Canada M3B 1Z3

### What does Super-OS/9 offer?

- A true disk operating with features found on UNIX\* systems and on the AMIGA.
- Multi-tasking and multi-user capability.
- Hierarchical directory structure with time/date stamps.
- Programmable file access privileges for increased security.

Extensive software is available for OS-9, most of which runs on Super-OS/9.

Super-OS/9 V1.1 includes an assembler, editor, symbolic debugger, communication software and terminal emulation package. Available languages include BASIC09, CIS COBOL, Fortran 77, Pascal, Lisp, C and others. Application software include wordprocessors, spelling checkers, data bases and spread sheet programs.

TPUG has acquired public domain software and will assist users in the conversion of commercial software to Commodore format.

### Portability and Expandability

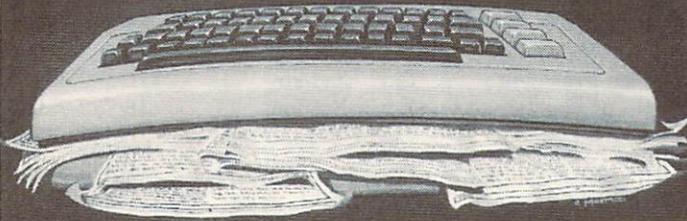
- Super-OS/9 programs will run on all OS-9 based computers (like the CoCo).
- Super-OS/9 will support standard disk drives (IBM format) and the Hi-res graphics board (700 x 300 dots).
- Super-OS/9 software is C compatible with OS-9 68k and AT&T Unix system V.

For further information call TPUG Inc. at (416) 445-4524, ask for Bruce.

NOTE: If you own a 3 board SuperPET and wish to acquire Super-OS/9, please call TPUG before, ordering Super-OS/9, for info about a hardware fix to a design error in your SuperPET computer.

Super-OS/9 is a trade mark of TPUG and Avygdor Moise.  
OS-9 is a trademark of Microware and Motorola.  
UNIX is a trade mark of Bell Laboratories.

# Expand Past Maximum Capacity!



## The Transactor

The Tech/News Journal For Commodore Computers

At better book stores everywhere! Or 6 issues delivered to your door for just \$15.00 (Overseas \$21 U.S. Air Mail \$40 U.S.)  
The Transactor, 500 Steeles Ave. Milton, Ontario, L9T 3P7.  
416 878-8438

Also check out The Transactor Disk and The Complete Commodore Inner Space Anthology - to us, expansion knows no limits!

# C64 + C128 Provincial Payroll

A complete Canadian Payroll System for Small Business.

- 50 employees per disk (1541)
- Calculate and Print Journals
- Print Cheques
- Calculate submissions summary for Revenue Canada
- Accumulates data and prints T4s
- Also available for 4032, 8032, PC-10 Commodore computers Fully Supported

Available from your Commodore dealer

Distributed by:

**M**ICROCOMPUTER SOLUTIONS  
170 The Donway West, Suite 401  
Don Mills, Ontario M3B 2G3  
Tel: (416) 447-4811

# COMSPEC

AUTHORIZED COMMODORE DEALER SINCE 1978.

HARDWARE • SOFTWARE • BOOKS • ACCESSORIES • SERVICE • RENTAL

866 Wilson Ave., Downsview, Ont. (between Dufferin & Keele) (416) 633-5605 or call our 24 hour B.B.S. 633-0185

**KO-REC-TYPE**  
MPS 802/1526/CPA80  
RIBBON.  
\$14.95

Buy 3 Get 1 Free  
(Limited Quantities)

**Font  
master  
II**  
WORDPROCESSING  
With 30+ Fonts  
including Hebrew & Greek  
\$69.95

**KRAFT  
JOYSTICK  
MAZE MASTER**  
4 or 8-Way Selection  
\$16.95



**MODEMS FROM  
\$69.95**

**COMSPEC RENTALS  
COMPUTER EQUIPMENT  
TO TRY BEFORE  
YOU BUY  
CALL FOR RATES!**

**PRECISION DISKS BY XIDEX**  
ss/dd \$14.95  
ds/dd \$17.95

**10% DISCOUNT  
TO TPUG MEMBERS**  
ON SOFTWARE, BOOKS AND ACCESSORIES  
(EXCLUDING SALES ITEMS)  
PHONE ORDERS ACCEPTED  
VISA/MASTERCARD/AMEX  
(416) 633-5605

# advertisers' index

Ahoy! Magazine	BC
Bayside	11
Brantford Educational Services	15
COMAL Users Group, USA	13,17
Comspec Communications	47
Computer Rentals	10
R.W. Dray & Co.	46
John Dunlop & Associates	45
Electronics 2001	IBC
High Res Technologies	15
Micro Solutions	47
Midnite Software Gazette	46
Mimic Systems Inc.	IFC
Programmers Guild Products	43
Rich-Hill Telecom International	20
R Link	10
TPUG (Disk Subscriptions)	3
TPUG (OS/9)	46
TPUG (OS/9 Software)	35
The Transactor	47
Ufland Software Inc.	7
Voyageur	10
Wilanta Arts	14

## TPUG Magazine Distributors

*Dealers: If you would like to carry TPUG Magazine in your store, you may order from any one of the following distributors:*

### CANADA

Compulit Distributors, Port Coquitlam, BC 604-464-1221

### USA

Prairie News, Chicago, IL	312-384-5350
Levity Distributors, North Hollywood, CA	818-506-7958
Whole Life Distributors, Englewood, CO	303-761-2435
M-6 Distribution, Houston, TX	713-778-3002
The Homing Pigeon, Elgin, TX	512-276-7962
Northeast News Distributors, Kingston, NY	914-382-2000
Fred Bay News Co., Portland, OR	503-228-0251
Alonso Book & Periodical, Alexandria, VA	703-765-1211
Cornucopia Distribution, Seattle, WA	206-323-6247
Guild News, Atlanta, GA	404-252-4166
Micro-PACE, Champaign, IL	800-362-9653
Nelson News 4651 F Street, Omaha, NE	68127
Summit City News, Ft. Wayne, IN	219/484-0571
Total Circulation, South Hackensack, NJ	201/342-6334

# TPUG Contacts

**TPUG OFFICE 416/445-4524**

**TPUG BBS 416/273-6300**

**TPUG MEETINGS INFO 416/445-9040**

### Board of Directors

President	Chris Bennett	c/o 416/445-4524
Vice-President	Gerry Gold	416/225-8760
Vice-President	Carl Epstein	416/492-0222
Recording Sec.	Doris Bradley	416/782-7320
	David Bradley	c/o 416/445-4524
	Richard Bradley	c/o 416/445-4524
	Gary Croft	416/727-8795
	Mike Donegan	416/639-0329
	John Easton	416/251-1511
	Keith Falkner	416/481-0678
	Anne Gudz	c/o 416/445-4524
General Manager	Bruce Hampson	416/445-4524

### TPUG Magazine

Publisher	Bruce Hampson	416/445-4524
Editor	Nick Sullivan	416/445-4524
Assistant Editor	Tim Grantham	416/445-4524
Production Manager	Astrid Kumas	416/445-4524
Ad Sales Reps	Hunter Nichols Inc.	416/439-4140

### Meeting Co-ordinators

Brampton Chapter	Jackie Bingley	c/o 416/445-4524
C-64 Chapter	Louise Redgers	416/447-4811
COMAL Chapter	Donald Dalley	416/742-3790
	Victor Gough	416/677-8840
Communications	Darrell Grainger	c/o 416/445-4524
Eastside Chapter	Judith Willans	c/o 416/445-4524
	Nina Nanan	c/o 416/445-4524
Hardware Chapter	Frank Hutchings	c/o 416/445-4524
SuperPET Chapter	Gerry Gold	416/225-8760
VIC 20 Chapter	Anne Gudz	c/o 416/445-4524
Westside Chapter	John Easton	416/251-1511
	Al Farquharson	519/442-7000
Business Chapter	Louise Redgers	416/447-4811
New Users Chapter	Louise Redgers	416/447-4811
C-128 Chapter	Adam Herst	c/o 416/445-4524

### Librarians

COMAL	Victor Gough	416/677-8840
PET	Mike Donegan	416/639-0329
SuperPET	Bill Dufield	416/224-0642
VIC 20	Richard Best	c/o 416/445-4524
Commodore 64	Derick Campbell	416/492-9518
B-128	Paul Aitchison	c/o 416/445-4524
Amiga	Mike Donegan	416/639-0329
C-128	Adam Herst (CP/M)	c/o 416/445-4524
	James Kokkinen (C-128)	" "

### TPUG Bulletin Board

Sysop (voice, weekdays)	Sylvia Gallus	c/o 416/896-1446
Assistant Sysop	Steve Punter	c/o 416/896-1446

# ELECTRONICS 2001

AN AUTHORIZED COMMODORE SYSTEM DEALER SINCE 1979

**FRIENDLY SERVICE FROM FRIENDLY FACES.  
THIS MONTH ONLY — 15% OFF ALL  
REGULAR PRICED SOFTWARE, ACCESSORIES  
AND MAGAZINES TO TPUG MEMBERS.**

**NEW IN STOCK NOW**

**1670 MODEM**

**1200 BAUD**

**FOR YOUR COMMODORE 64 - THIS MONTH'S SPECIALS**

<p><b>SIDEWAYS FOR C-64</b></p> <p><b>\$39.95</b></p>	<p><b>THE MANAGER C-64</b></p> <p><b>\$19.95</b></p>	<p><b>IMAGE MAKER MPS 802 HIGH RES. PRINTING</b></p> <p><b>\$10.00</b></p>	<p><b>OUBLIETTE</b></p> <p><i>Oubliette</i></p> <p><b>\$14.95</b></p>	<p><b>SPRINT PRINT</b></p> <p>FULL SPOOLING PRINTER BUFFER</p> <p><b>SALE PRICE \$29.95</b></p>	
 <p>The complete guide to the 1541 operating system.</p> <p><b>\$21.95</b></p>	 <p><b>\$36.95</b></p>	<p><b>PaperClip</b> (the word processor)</p>  <p><b>\$69.95</b> with SpellPack \$89.95</p>	<p><b>Boss Joystick</b></p>  <p><b>\$21.95</b></p>	<p><b>Kraft Joystick</b></p>  <p><b>\$15.95</b></p>	<p><b>Compuserve Starter Kit</b></p> <p><b>\$39.95</b></p>

**PRICES IN EFFECT UNTIL APRIL 30, 1986**

**ELECTRONICS 2001 LTD.**

Name \_\_\_\_\_  
Address \_\_\_\_\_  
\_\_\_\_\_

Check or money order enclosed  Visa  MasterCard  Amex

Acct.# \_\_\_\_\_ Exp. Date \_\_\_\_\_ Signature \_\_\_\_\_

**ELECTRONICS 2001 LTD.**

5529 Yonge Street, Willowdale, Ontario M2N 5S3  
Tel: (416) 229-2700

Note: All prices in Canadian Funds. Phone and mail orders welcome.  
Ontario residents add 7% sales tax.  
Add 5% for shipping (minimum charge \$2.00)

   10% discount for TPUG members on regular priced software, accessories and magazines.

# Ahoy!

## Number One in Commodore Coverage! \*

### Ahoy!

Ion International

\$2.50 ★★★★★-

Monthly publication devoted to Commodore computing. *Ahoy!* seems to fluctuate from issue to issue, but seems recently to be pulling ahead of *Run* and *Computel's Gazette* in content and style. Of the 3 'big names,' I would recommend this one most. (—B.D.)

### Computel's Gazette

Compute Publications

\$2.95 ★★★★★+

for: C-64

Slick monthly publication with coverage of Commodore's VIC-20, 64, Plus/4, and newer computers. Emphasis is on type-in games and occasional utilities. Reviews are generally candy-coated but reasonably current. Well-illustrated and laid-out. Goes with the flow, avoiding controversy and originality. Features a good selection of advertising. (—B.D.)

### RUN

CW Communications

\$2.95 ★★★★★+

for: C-64, C-128

This slick monthly Commodore magazine is chock-full of ads, lots of color and photos, occasional decent articles. Reviews are often out of date and, worse, not to be trusted. What can you say about a supposedly knowledgeable publication that prints shameless gushing praise for a turkey like the PLUS/4 month after month. Best features: MAGIC column, and STRASMA's 'Commodore Clinic.' (—B.D.)

In its September-October '85 issue, *INFO* (formerly *INFO 64*) Magazine reviewed all the major Commodore-specific publications. None received a higher rating than *Ahoy!*

Our review is reprinted here. In the interest of fairness, we've also reprinted the reviews of the other two members of the 'Big Three'—all three reviews word-for-word, exactly as they appeared in *INFO* Magazine.

We thank *INFO* Publisher Benn Dunnington for covering our segment of the marketplace in his usual hard-hitting style. And we'll try to stop fluctuating!

\*Source: *INFO* Magazine, September-October '85