The Magazine For ALL Commodore Computer Users

TPUG Magazine

$2.95

December 1985

Butterfield explains:
64 and 128 Video

Punter imagines:
The Ultimate Word Processor

Wright explores:
WYSIWUG

Plus:
Amiga reference pullout
Our first Amiga program
Expanding the B-128

0 56698 72145 12

THE Premier Word Processing Package – CREATIVE COMPUTING

"PaperClip is easy to use, yet offers the advanced features of programs designed for the IBM PC These include: block move, copy, delete, macros, automatic page numbering, headers, footers, underlining, boldface, super and subscripts, variable character pitch, and custom character sets. The editing screen can be set up to 130 columns wide, & text can be scrolled in any direction. A preview mode displays formatted text exactly as it will appear on the printed page. You may further define your own formatting parameters, including margins, line lengths, page length and spacing.
PaperClip contains over 30 printer files for all the current major models. The documentation is excellent and the disk itself unprotected, though keyed through a joystick port. This means you can make as many back-up copies as you like, but can use the program only when the key is inserted."–CREATIVE COMPUTING

"PaperClip is one of the easiest of the professional word processors to use, with a sensible manual and plenty of aids for the accident-prone."–COMPUTING NOW

"a "must have" in an ideal software library"–ELECTRONIC LEARNING

"PaperClip is the Cadillac of word processors"–OMNI *"...best professional word processor available"*–RUN

"an excellent full-featured word processor"–THE BOOK OF COMMODORE SOFTWARE *"hard to beat"*–ACE

"So clearly superior, ...State-of-the-art word processing"–ANTIC *"...most powerful of packages"*–COMMODORE MAGAZINE

"So far as we are concerned, PaperClip is the top word processor running on a micro computer."
–HOME APPLICATIONS FOR THE C-64

"exceptional word processing"–INPUT *"many features...easy to use"*–ATARI EXPLORER

"You'll find yourself growing spoiled."–FAMILY COMPUTING *"PaperClip is a logical evolutionary step forward."*–AHOY

"A superb word processor,...the most sophisticated to date!"–COMPUTE *"the ultimate word processor..."*–ANALOG

"...does exactly what it was intended to do...and more"–COMMODORE MAGAZINE

"An excellent word processor,...well designed...many advanced features!"–INFOWORLD

"...facts attest to its excellence!"–FAMILY COMPUTING

"You will not find a word processing package superior to this one!" –CREATIVE COMPUTING

PaperClip™

*"The #1 Best Selling Word Processing Package"**

–BILLBOARD'S COMPUTER SOFTWARE CHARTS!

Available for Commodore 64, Pet, all Atari home computers, and new enhanced 128K versions for Apple II/e/c, Atari 130XE and Commodore 128. New in 1986: Paperclip Elite for IBM MS DOS, Amiga and Atari ST.

30 Mural Street
Richmond Hill, Ontario
L4B 1B5 Canada
(416) 881-9941
Telex: 06-986-266

17875 Sky Park North,
Suite P, Irvine, California
USA 92714
(416) 881-9816
Telex: 509-139

"The Energized Software Company!"

WRITE TO US FOR FULL COLOR CATALOG of our products for Commodore, Atari, Macintosh, Apple and IBM systems.

FOR TECHNICAL SUPPORT OR PRODUCT INFORMATION PLEASE PHONE (416) 881-9816

**as compiled from national retail store sales reports for week ending January 5, 1985*

TPUG Magazine

Publisher: Bruce Hampson
Editor: Nick Sullivan
Assistant Editor: Marya Miller
Production Manager: Astrid Kumas
Editorial Assistant: Iwona Sukiennik
Cover Painting: Don Ballanger
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.

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 16,000 Newsstand 10,000
ISSN #0825-0367

Subscriptions to TPUG Magazine may be obtained by joining the Toronto PET Users' Group (TPUG) Inc. Another benefit of TPUG membership is the right to order inexpensive disks and tapes from the club's extensive software library. Yearly membership fees are:

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
Suite G7
Toronto ON
Canada M3B 1Z3

1552 Hertel Ave.
Suite 144
Buffalo, NY
USA 14216-2882

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

DIRECTORY

85 TPUG Magazine DEC

Inside Information

TPUG management changes

This month we say goodbye to Louise Redgers, who has been our publisher and advertising director, as well as TPUG's general manager, since March '85. It feels strange not to have our own personal Tasmanian devil whipping in and out of the office, leaving a trail of papers in her wake as she attempts to do twelve million things at once. We at the magazine enjoyed working with Louise as publisher, and we'll miss her.

Louise's successor as general manager and publisher is Bruce Hampson. For the past three years, Bruce has been business manager of the Canadian Computer Dealers Association, a trade association for the microcomputer industry — a position he still holds. Bruce brings boundless enthusiasm to TPUG, as well as a varied business background that includes everything from car sales management to running a chain of pet stores for six years. TPUG wishes Bruce the best of luck as he takes on this complex new challenge.

Butterfield 1986 Diary

Jim Butterfield's new 1986 *Commodore Computer Diary* is available now. Besides being a handy day-to-day diary, this pocket-sized gem is packed with useful (some will find it essential) reference material for the whole range of Commodore computers. The diary is sensibly laid out and designed for practical use. Do yourself a favour, and order yours as soon as possible. (TPUG should have them in shortly).

This month's issue

Our cover this month is the creation of artists Don Ballanger and Leonardo da Vinci. (We think Leonardo would have done *his* part much better if he had had an Amiga).

In this issue, our feature is 'Computer Graphics', with articles written by Jim Butterfield and Chris Johnson. Jim has reworked his comprehensive description of the VIC II video chip from a series he did originally in *Compute!*, and also takes time to look at what you can do with the advanced graphics commands of the C-128. Chris gives us his ABasiC program **Drawmouse**, our first-ever program for the Amiga.

Also in this issue: **WordPro** author Steve Punter gives us his idea of 'the ultimate word processor', while Ian Wright looks at the advanced capabilities of non-Commodore word processors — features that will probably be incorporated in word processors written for the C-128 and the Amiga in the near future. Plus there's our usual broad selection of Micro Processes articles and product reviews.

Raising further issues...

Readers please note that our next *TPUG Magazine*, (January/February '86) will be the first combined issue of the year. (There will be, as always, ten in the year).

We will be starting out the year with the theme of 'Artificial Intelligence'. Something there is a lot of round 'ere, mate.

BBS downs and ups...

We are pleased to announce Sylvia Gallus has taken over as TPUG's new Sysop, and the BBS is up and running again. In addition, Steve Punter has handsomely taken time out of his busy schedule to sign on as Assistant Sysop. This month's password — UNICORN. New BBS Hotline Number: (416) 273-6300

VIC w-p breakthrough!

VIC co-ordinator Anne Gudz tells us that Advantage Computer Accessories is offering a word processor for the VIC 20 — **Textmaster**, available on cassette or disk. It's for all VICs, and all Commodore printers. Price: $19.95, plus provincial sales tax.

Textmaster from Advantage Computer Accessories, 1020 Meyerside Drive, Mississauga, Ontario L5T 1K7.

More on spooling

Liz Deal has provided us with a couple of factual corrections to her article on B-128 spooling (August/September 1985). Liz based her B-128 program on a PET program by the inventor of IEEE spooling, T.M. Peterson (not Patterson, as the article stated) that originally appeared in *The Transactor* (not *COMPUTE!*). Mr. Peterson also supplied Liz with the following additional references to his work:

1. *COMPUTE!*, volume 3, issue 1, January 1980, page 118.

2. A quotation in Raeto Collins West's *Programming the PET/CBM*, page 379.

Bradley into the breach

As we go to press, TPUG stalwart David Bradley is heroically manning the magazine booth at the World Of Commodore III show. WOC III came this year at the crazy time in our production cycle. Had it not been for David, our C-64 TOP 20 contest at the show would have ended before it began. This morning we left him to deal single-handedly with the madding crowd, having already imposed on his generosity for the same task all day yesterday, and in the preparations for the contest in the week previously. From all the magazine staff to David — thanks for the help.

Don't read this!

...Unless you are a rabid adventure game player.

Have you ever played a thoroughly enjoyable adventure game and got stuck three-quarters of the way through? I don't mean temporarily stuck, for a mere month or so: I mean hopelessly, irrevocably, gloriously *stuck!* With a sigh, you wistfully tuck your well-worn main disk back inside your marvellously-illustrated package of **The Dungeons Of Dword**, place the box on your highest unused 'dust-collection' shelf, and reformat for ever your equally well-worn 'Dword positions' disk...

If you have, take heart. *TPUG Magazine* plans to do something about this ghastly predicament.

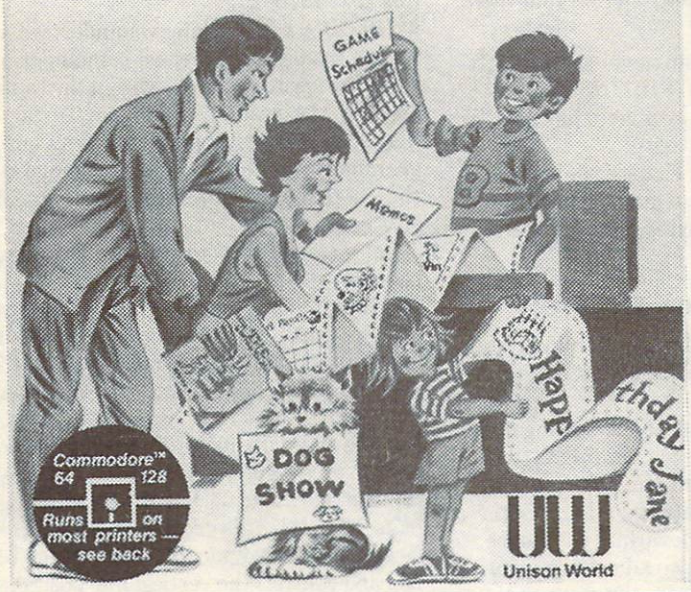
We ask those who have been stuck, and who have actually overcome seemingly insurmountable obstacles in adventure games, to send us their most valuable and hard-won hints from these favourite (or loathed) games. We will publish the best of them from time to time in a new 'Adventure Clinic' section.

Also, if you are still irrevocably stuck, and would like a friendly boost, write to 'Adventure Clinic' c/o *TPUG Magazine*, Suite G7, 101 Duncan Mill Road, Don Mills, Ontario M3B 1Z3. We will be happy to help you on your way again with a timely hint or two (providing *we're* not stuck at the same spot in that game...)□

The Editors

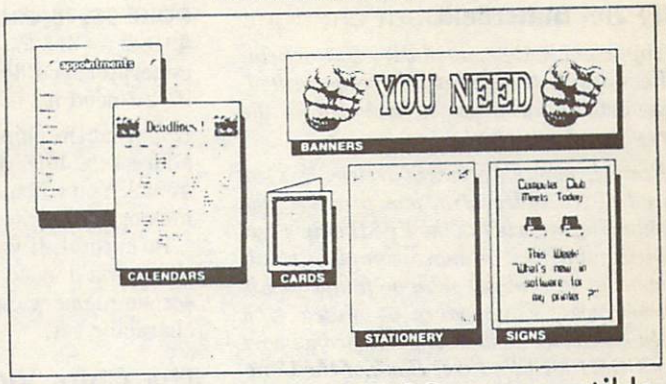
PrintMaster™

Print your own custom ■ Cards ■ Signs
Invitations ■ Stationery ■ Banners ■ Calendars



PrintMaster
AND
Art Gallery I

from Unison World Inc.



For your C-64/128, IBM PC or compatible

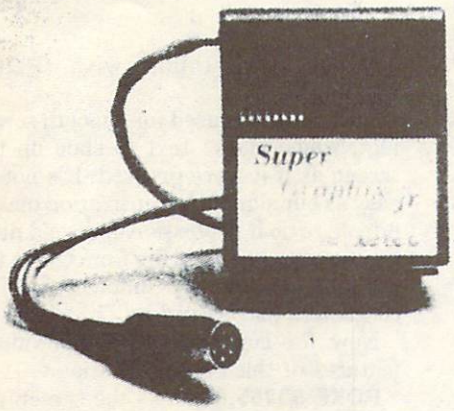
Features at a glance:

- library of 111 readymade graphics plus 11 background patterns
- 8 type fonts in a range of sizes, with Outline and 3-D text effects
- 11 border designs to frame your signs and cards
- Graphics Editor with powerful tools for easy creation, altering, and saving of graphics
- Text Editor for fast writing and placement of text. With choice of type fonts, size, and special effects
- Preview mode allows you to see your creation BEFORE you print it
- Design Archive. You can save your favorite designs to use later

PrintMaster supports the following printers: EPSON RX, FX, MX (with Graftrax) and LQ1500, Okidata 92, 93 (in IBM mode), IBM graphics printer, Tandy DMP-2100, DMP-430 (in IBM mode), Panasonic 1090, 1091, 1092, Citizen, and the Star Gemini.

Epson, Okidata, IBM, Tandy, Panasonic, Citizen, and Star are registered trademarks of Epson America, Inc., Okidata Corp., International Business Machines Corp., Tandy Corp., Panasonic Corp., Citizen Corp., and Star Micronics, Inc., respectively.

Super Graphix jr.



High Performance . . . . . Low Cost! ! !

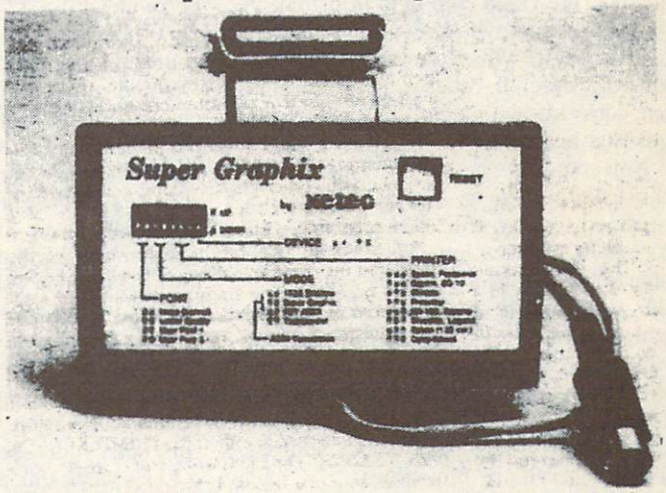
NOW - CORRESPONDENCE QUALITY and GRAPHICS are available for the Commodore Computers in one cost effective interface with the following features

- Micro Buffer
- Graphics/Normal Quality Printing
- Correspondence Quality
- 8 Active Switches with Changes Constantly Monitored
- 10 Printing Modes
- Centronics Compatible
- Supports All Major Printers
- 100% Compatible with Software for 1525
- User's Manual with Software Examples
- Compact Design Plugs Directly into Printer

ATARI VERSION AVAILABLE

The GRAPHIX AT printer interface for Atari computers supports most parallel printers and allows use of Atari graphics characters.

Super Graphix



You Asked For It . . . . . Here It Is! ! !

GRAPHICS and FONTS plus an 8K BUFFER for the ultimate in performance and speed. The XETEC Super Graphix Interface for Commodore Computers now offers a new high in technology with these features:

- 8K Buffer Standard
- 10 Printing Modes
- 3 Internal Screen Dumps
- Extensive Command Channel
- Reset Button to Halt Printing from Buffer
- Switch Settings on Label for Quick Reference
- Correspondence Quality Font Built-In
- Capable of Storing 2 Additional Fonts
- Correct Graphics/Text Aspect Ratio for all Major Printers
- 8 Active Switches with Changes Constantly Monitored
- Internal Fonts Support Super-script, Sub-script, Underlining, Bold-face, and Choice of 9 Pitches

QUALITY PRODUCTS AVAILABLE FROM

PHASE 4 DISTRIBUTORS

7157 Fisher Road, S.E., Calgary, Alberta T2H 0W4
(403) 252-0911

FOR YOUR NEAREST DEALER CALL

1-800-661-8358

C-64 Video from $41 to $5A

by Jim Butterfield

For this month's series of articles on C-64 and C-128 video, Jim has provided the following overview of the VIC II chip's versatile abilities. A more complete treatment of the subject may be found in his seven-part 'Commodore 64 Video — a Guided Tour', which is published as part of ***COMPUTE!'s First Book of the Commodore 64.***

The 6566 video chip digs out its display information from the computer's memory. From this memory, it gets: the characters to display (the video matrix), how to draw the characters (the character base) and sprite information. If it's been kicked into high-resolution mode, it doesn't worry about characters, but still uses the character base to tell which pixels to turn on or off, and the video matrix to tell what colours to use. Separate from all this, the video chip gets information from the colour nybbles — a separate type of memory to which the video chip has 'direct access'. In other words, it doesn't use conventional memory addressing to get this last data.

Not counting the colour nybbles, the chip looks at memory a great deal; but it's confined to only 16K of memory to get its video information. That's only one quarter of the RAM memory capacity of the Commodore 64. But we can pick which quarter — which 16K slice — we would like to use. Normally, it's the first slice, which goes from address zero to 16383, but we can change that. We change it by a poke to memory address 56576 — here are the values:

POKE 56576,7 (normal value): the chip sees RAM from 0 to 4095, and from 8192 to 16383. The ROM character generator is in the slot from 4096 to 8191.

POKE 56576,6: the chip sees RAM from 16384 to 32767. No character generator, so we should either draw our own characters (Greek? Hebrew? Italics?) or use the screen in high resolution.

POKE 56576,5: the chip sees RAM from 32768 to 36863 and from 40960 to 49151. The ROM character generator is in the slot from 36864 to 40959.

POKE 56576,4: the chip sees RAM from 49152 to 65535. There's no character generator; you'll have to make your own if you need it.

Note that the chip can never see RAM at addresses 4096 to 8191 or 36864 to 40959. You will not be able to put screen memory or sprites there.

Be careful. If you change the memory slice, you'll need to move everything: screen memory, sprites, and maybe the character set.

The Chip: Video Control

On to the 6566 chip itself. We'll go through the registers, but not in strict numeric order.

Since we've talked about setting the memory 16K slice, we might as well describe how to point the chip towards the screen (video matrix) and character set (character base) within that slice. First, we figure the two addresses according to their offset from the start of the slice. So, if we've chosen the slice beginning at 32768, an address of 33792 would be reduced (by 32768) to a net of 1024. Do this with both the screen and character base addresses. Next: screen locations must be a multiple of 1024, and character base locations must be a multiple of 2048. You can't start the screen and character base just anywhere; each slice has at most only 16 starting points. So... divide by 1024 to get the 'block number' for each of the two values. Each of these numbers should be from 0 to 15.

Now — take the screen number, multiply it by 16, and add it to the character base number. The resulting value, an even number from 0 to 254, should be poked to location 53272. The one poke sets both addresses.

Let's work this backwards. On a 'normal' Commodore 64, we may peek address 56576 and we'll usually see a value of 7, which says 'the slice from 0 to 16383'. That's what we expect. Let's track the video further: a peek of 53272 yields 21. We need an even number, so we may throw away the one and get 20. Now: 20 is 1 times 16 plus 4. So we read this as: video matrix at 1024 (remember that the screen can usually be poked at 1024?) and character matrix at 4096. Hmmm... we did say, above, that there is indeed a character generator located at 4096. Oh, and if you get 23 instead of 21, that's okay; we're just using a different part of the character generator (for text mode).

Just a forward-looking reminder: when we get to sprites, you must remember to place the sprite drawings within the same slice as the rest of the video. In fact, that's sometimes the reason why programmers switch to another memory slice — to make room for lots and lots of sprites.

Let's move on to general control registers.

Location 53265 (hex D011) is an important control location, responsible for many functions. Its normal value is 27 decimal. First, let's look at the 'low nybble'.

If we want to make the screen slide gently in a vertical direction, we may replace this value with new ones ranging from 16 to 23. This crops a line from the display (we now have 24 rows instead of 25), and controls the vertical fine positioning of the screen. Try this, and watch what happens:

```
for j=16 to 23:poke 5326
  5,j:next j
```

Restore everything with **POKE 53265,27**.

This feature is used for 'smooth scrolling', which allows text to slide up the screen as if it were greased. It's not as easy as our simple demonstration makes it look: to do it professionally, we'd need two screens, and we'd flip from one to the other whenever a complete set of characters had passed by.

Now for some of the higher-valued features of this control location.

POKE 53265,0 blanks the screen (by eliminating the 16 bit, which is the Screen Enable Control). The computer runs a bit faster with the screen shut off.

High Resolution

The next control bit — value 32 — switches the display to pure bits. No more characters: the screen will be purely pixels as we switch to high resolution mode. We'll use a lot of memory for this one: memory to feed the screen will be 8000 bytes.

We mentioned briefly before that the high resolution bits are taken from the character base location, not the screen (video matrix) area. An extra factor is

this: a high resolution screen can come only from block 0 or block 8 within a slice. Thus, there are only two places from which a high resolution screen can be drawn within each 16K slice. Indeed, practically speaking, there are only five locations within the whole computer that a high resolution screen can be located. These are: 8192, 16384, 24576, 40960 and 57344. (49152 is barely possible, but there are so many difficulties that it's hardly ever used.)

High resolution needs to be carefully set up, but let's plunge right into it, using an impractical address (0). Type **POKE 53265,59** and you'll see an intricate pattern on the screen. What you are looking at now is a bit map of RAM memory addresses 0 to 4095, plus the character generator area. The top of the screen will twinkle a little. Those are the page zero values changing: things like the real time clock and interrupt values are constantly in motion. In the bottom half of the screen, we'll see the character generator itself. That's one of several reasons why address 0 is not practical for high resolution bit mapping: we can't change the character generator, since it's in ROM.

While we're in this peculiar mode, try to clear the screen. It doesn't clear, but colours change. Screen memory, into which we are typing, is now used to hold colour information for the high resolution screen. If you're good at typing 'blind', try these:

```
for j=3200 to 3519:poke
  j,0:next j
for j=3204 to 3519 step
  8:poke j,255:next j
```

If you manage to type the lines in right, you'll see a blank area being set up in the high resolution screen, and then a line.

Reset everything with the power switch, and let's move on.

Extended Colour

If we add 64 to the contents of 53265, we'll invoke the extended colour mode. This will allow us to choose both background and foreground colours for each character. Normally, we may only choose the foreground: the background stays the same throughout the screen. With extended colour, you lose some of your character set, but get a variety of background colours across the screen.

Try **POKE 53265,91**. Nothing happens, except that the cursor disappears... or at least becomes less visible. Why? We've traded the screen reverse feature for a new background colour. Try typing characters in reverse font, and see what happens. Try typing some 'shifted' characters — you won't get them, but may be surprised at what you see. Extended colour is purely a screen display phenomenon. **POKE 53265,27** will bring all the characters you have typed back to their normal appearance.

Location 53270 (hex D016) doesn't use the two highest bits, so we'll poke values that range only from 0 to 63. If we peek 53270, we'll see a number that is 192 too big. To see the working value, type **PEEK(53270) AND 63**.

We saw a vertical fine scroll in location 53265. Location 53270 has a horizontal fine scroll that works exactly the same way. Type:

```
for j=0 to 7:poke 53270,
  j:next j
```

You'll see the screen characters slide over horizontally; and the screen is trimmed to 38 characters.

If you add 16 to the contents of 53270, you'll switch to multicolour mode. This is not the same as the extended colour mode that we discussed previously. Multicolour allows *selected* characters to be shown on the screen in a combination of colours. Extended colour, you may remember, allows screen background and foreground to be individually set on each character. Here's the trick: we invoke multicolour on an individual character by giving that character a colour value greater than 7. This way, the 'control-key' colours (red, blue, black) behave normally; but the 'logo-key' pastels (grey, puce) switch to multicolour mode.

You'll need to create a new character base to exploit the advantages of multicolour, since the old characters weren't drawn with multicolour in mind. However, we can get a quick idea of the feature by invoking it: **POKE 53270,24** sets up multicolour; the screen characters may turn a little muddy, but don't worry about them. Now set a primary colour such as cyan and type a line. Normal, right? Next, set up one of the alternate colours (hold down the 'Commodore' key and press a key from 1 to 8). Type some more: you'll get multicolour characters. They won't make much sense, since the character generator isn't building the colours suitably; but you can see that something new is going on.

Adding 32 to the contents of 53270 gives chip reset. You won't want to do this very often — it's done on your behalf when you turn the power on. If you do use it, remember that, to make it work, you must turn reset on, and then off again. **POKE 53270,32:POKE 53270,8** will clear you out of multicolour mode.

The Raster Register

Location 53266 (hex D012) and the high

Continued overleaf...

6566 Video Chip
C64 Control & Miscellaneous Registers

D011	Extended Clr. Mode	Bit Map	Display Enable	Row Select	Y-Scroll		53265
D012	Raster Register						53266
D013	Light Pen Input					X	53267
D014						Y	53268
D016	x	x	Reset	Multi Colour	Column Select	X-Scroll	53270
D018	Screen: VM13	VM12	VM11	VM10	Character Base: CB13	CB12	CB11	x	53272
D019	IRQ	Interrupt Sense:			Light Pen	Spr-Spr Collision	Spr-Back Collision	Raster	53273
D01A		Interrupt Enable:			Light Pen	Spr-Spr Collisions	Spr-Back Collisions	Raster	53274

Colour Registers

D020	X	Exterior Colour (Border)	53280
D021	X	Background Colour #0	53281
D022	X	Background Colour #1	53282
D023	X	Background Colour #2	53283
D024	X	Background Colour #3	53284
D025	X	Sprite MultiColour #0	53285
D026	X	Sprite MultiColour #1	53286

6566 Video Chip
C64 Sprite Registers

Sprite 0	Sprite 7		Sprite 0	Sprite 7
D000	D00E	X Position	53248	53262
D001	D00F	Y Position	53249	53263
D027	D02E	Sprite Colour	53287	53294

Bit For Sprite#: 7 6 5 4 3 2 1 0

D010	X-Position High	53264
D015	Sprite Enable Flags	53269
D017	Y-Expand	53271
D01B	Background Priority	53275
D01C	Sprite MultiColour Mode	53276
D01D	X-Expand	53277
D01E	Interrupt: Sprite Collision	53278
D01F	Interrupt: Background Collision	53279

Diagram courtesy of The Transactor

bit of the previous location are not much use to the BASIC programmer, but can be very valuable if you're using machine language. Here's the idea: by looking at these locations, you can tell exactly where the screen is being scanned at that moment. This allows you to change the screen as it's being scanned: half-way down, you could switch from characters to bit map, or change to multicolour, or move a sprite that has already been displayed.

If you're really hot on machine language, you may want to take an extra step: instead of watching where the screen is, you can leave a message: "Wake me when you get to scan line 100". ML types will recognize this as an interrupt request. How do you set the identity of the desired scan line? By placing it into the same locations, that's how. We have a dual function here: when we read, we recall the scan location; when we write, we store an interrupt value.

Light Pen

Locations 53267 and 53268 (hex D013 and D014) are the light pen registers. An Atari-style light pen can be plugged into joystick port number one; if it sees a suitable signal from the screen, the X and Y values will be latched into these registers. The light pen can be used on an interrupt basis: we can 'stop the music' and get immediate action if we choose to set things up that way.

This is the second time we've mentioned interrupts: let's discuss them.

Interrupts

Interrupts are for machine language maniacs — things happen too fast for BASIC to cope in this area. If you're a beginner, you might prefer to skip this section. There are four types of interrupt: raster, light pen, and two kinds of sprite collision. ("Daddy, what's a sprite?" "Shut up, kid, we'll talk about them later.") We may use all of them, or none; and even when these signals are not used for interrupt, we can check them.

Location 53273 (hex D019) tells us which of the four events have occurred. We don't need to make the interrupts 'live' — they will signal us any time the particular event happens. The weights are as follows:

1 (bit 0): the raster has matched the preset line value;

2 (bit 1): a sprite has collided with the screen background;

4 (bit 2): a sprite has collided with another sprite;

8 (bit 3): the light pen has sensed a signal;

128 (bit 7): one of the above has triggered a live interrupt.

Once any of the above takes place, the bit will remain stuck on until you turn it off. How do you turn it off? It sounds like backwards logic, but you turn an interrupt signal off by trying to turn it on. Suppose that we have both a raster and a light pen signal; we'll see a value of 9 (8+1) in the interrupt register. Now — suppose further that we are ready to handle the light pen, so we want to turn its signal off. We do this by storing 8 into location 53273. Huh? Wouldn't that turn it on? Nope: it turns it off, and leaves the other bit alone. So after storing 8, we look at the register again and (you guessed it) we see a value of 1 there. I swear.

Location 53274 (hex D01A) is the interrupt enable register: it sets the above signals for 'live interrupt'. Select bits 0 to 3 corresponding to the interrupts you want. Whatever you select will now trigger a processor interrupt when it happens, and also light up that high bit of 53273. Don't forget to shut the interrupt flag off when you service it, using the method indicated in the previous paragraph. Otherwise, when you finish the job and return from the interrupt (with RTI) it will re-interrupt you all over again.

A Little Colour

Some of the colours we have mentioned, and some we have yet to discuss, are neatly stored in addresses 53280 to 53286 (hex D020 to D026). We may store only values 0 to 15 here, for the sixteen Commodore 64 colours.

The arrangement of registers is: the exterior (border) colour, then four background colours (they may be selected as part of multicolour characters or bits); and finally, two colours reserved especially for sprites. ("Daddy, what's a sprite?" "Isn't it time you were in bed, kid? Okay, we'll talk about...")

Sprites

Sprites are completely separate from the 'conventional' video circuitry. You can lay a sprite on top of just about anything. But first, what's a sprite and how do we define it?

Sprites are sometimes called Movable Object Blocks (MOBs), and that's what they are — movable objects. The nice thing about them is that they appear on the screen independently of the main screen image, so that when our sprite airplane (for example) flies past a background object on the screen, the object reappears after the plane has passed by. This can save a lot of programming.

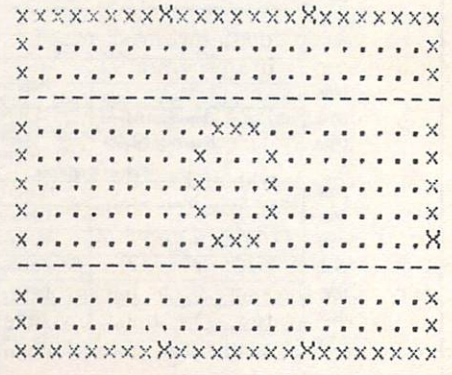
We noted earlier that the video chip can reach only 16K for its information. This includes three things: the screen memory (or video matrix), the character generator (or character base), and the sprite information. It all has to come out of the same 16K section.

Each sprite drawing occupies 63 bytes, and uses a 64 byte block. (Yup, we're gonna divide by 64 and use a sprite 'block number' later). So within 16K, we would have room to draw up to 256 sprites (if it weren't for that pesky screen and character set). We use only 8 drawings at a time, but we can switch a given sprite from one drawing to another as we wish, to make the legs move or the wings flap.

In the 'normal' Commodore 64 configuration, the video chip accesses the slice from address 0 to 16383. Forget about most of the first 1024 bytes; they are busy as a BASIC work area. Forget about addresses 1024 to 2047; that's the screen. Forget about everything above 2048, since that's where your BASIC program will go unless you move it. Even then, forget about the addresses from 4096 to 8192, since there's a character generator there, and it can't be changed. So where do we put our sprites?

To draw lots of sprites, we would need to relocate BASIC RAM to a higher location, or move video to another 16K slice that is not so busy. But we can find room for a few sprites in the existing space. The following sprite areas are available: sprite 11 at 704 to 766; sprite 13 at 832 to 894; sprite 14 at 896 to 958; and sprite 15 at 960 to 1022. These last three use the cassette tape buffer, so don't expect your sprite maps to survive tape access.

There are quite a few utility programs around that will help us draw sprites. You should use them; they will help make life easier. In the meantime, we can draw a sprite the 'hard' way by using a sheet of squared paper. Let's draw a diskette (battle of the floppies?). A quick sketch:



There are twenty-four pixels across (that takes three bytes of eight bits each), and twenty-one down. We may analyze the pixel pattern eight at a time, using a binary system to describe each byte. We end up with DATA statements something like:

```
10 data 255,255,255,128,
   0,1,128,0,1,128,0,1,1
   28,0,1,128,0,1,128,0,
   1,128,0,1
20 data 128,28,1,128,34,
   1,128,34,1,128,34,1,1
   28,28,1
30 data 128,0,1,128,0,1,
   128,0,1,128,0,1,128,0
   ,1,128,0,1,128,0,1,25
   5,255,255
```

Now we 'place' the sprite into slot 13 with:

```
40 for j=0 TO 62:read x:
   poke j+832,x:next j
```

Running the above program will place the sprite drawing into slot 13, but you won't see anything; nobody is using it yet.

Let's tell a sprite to use this drawing. We do it in an odd way: we don't use the video chip control registers at all. Instead, we use the video matrix, or 'screen memory'. You may recall that 1024 addresses are set aside for the video memory, but the screen only holds 1000 characters. What about the extras? At least some of them are used to designate which sprite picture to use for a given sprite. The last 'live' screen address is 2023. We could point sprite 0 to sprite drawing 13 (the one we have just done) with **POKE 2040,13**. Better yet, let's point *all* the sprites at this drawing:

```
50 for j=0 to 7:poke 204
   0+j,13:next j
```

We're almost ready to energize the sprite, but first, let's assign a position on the screen. For sprite 0, we set the position by poking to 53248 and 53249. Let's put a value of 99 in each, and then turn the sprite on. If you've run the above program, you may do this with a direct command, or give it a program line:

```
60 poke 53248,99:poke 53
   249,99:poke 53269,1
```

Either way, you should see the sprite on the screen. You can try changing the sprite colour as desired by poking a value from 0 to 15 into location 53287. One colour will be the same as the background, so that the sprite will be almost invisible — but not quite, since we can see when it covers part of the text.

You can move the sprite around at will by changing the values you have poked into 53248 and 53249. Try playing with the values: you may find that (vertically, at least) you can move the sprite partly or completely off the screen. If you like, try the following command:

```
for j=99 to 180:poke 532
  48,j:next j
```

Now substitute 53249 for 53248 and try it again.

A little experimenting will show that we can move the sprite vertically anywhere we like, including partly or completely off the screen. But we can't reach the full width of the screen with the range of values (0 to 255) that we can poke into 53248. We use a high bit to cover the extra distance. It's located at 53264: poking 53264 with a value of 1 causes sprite zero to be moved to the right — perhaps off screen.

For sprite position or colour, we use a particular address to control a particular sprite. In other words, there are eight addresses, one for each sprite. For the other controls — including the sprite enable and x-high, which we have already used — we must signal which sprites we wish to affect by a code, since all sprites are affected by a single control address. Here's how it works.

We use a bit map; the pattern is:

Sprite 0: value 1
Sprite 1: value 2
Sprite 2: value 4
Sprite 3: value 8
Sprite 4: value 16
Sprite 5: value 32
Sprite 6: value 64
Sprite 7: value 128

We use addition to signal a combination of sprites. If we wished to turn on sprites zero and two, we would **POKE 53269,9** (nine is the sum of eight and one). All other sprites would be turned off.

That's how the x-position high bit works: we set sprite zero to the right-hand sector of the screen with **POKE 53264,1**. All the other registers we will discuss work the same way.

We can make the sprite move behind the main screen, if you wish. Do this with location 53275. For example, **POKE 53275,1** will place the sprite behind the screen text.

The sprite that we have drawn isn't very big. We can make it larger in the X and Y directions with addresses 53277 and 53271 respectively. These addresses are often used together — when an object is drawn bigger, it looks closer, and we often want this effect in games and animations. Try, separately or individually, **POKE 53277,1** and **POKE 53271,1**.

Our sprite is one colour only — the colour we selected in 53287. The other colour is 'transparent', so it isn't really a colour at all. We may code our sprite in four colours (three plus transparent, to be exact), but we would need to draw it slightly differently. Instead of one bit representing either 'colour' or 'transparent', a grouping of two bits will be needed to describe four conditions: the sprite colour (as before), special colour #1, special colour #2 and transparent. These special colours, by the way, are kept at 53285 and 53286: they are the same for all sprites; only the designated sprite colour is individual.

The last two registers tell you about collisions. **PEEK(53279)** will tell you if any sprites have collided with the background since you last checked. They certainly have, of course, if you've been messing around with the screen as suggested. **PRINT PEEK(53279)** will yield a value of one: checking the bit table above tells us that sprite zero has hit the background. Now: checking this location clears it, but if the sprite is still touching some of the screen text, it will flip right back on again. Move the sprite to a clear part of the screen. Print the peek again — it will likely still say one, since the sprite has hit characters since it was last checked. If the sprite is safely in a clear screen area, the next peek will yield a zero.

We've only activated one sprite, so we won't see any signs of collision between sprites. You would see this in location 53278, but right now **PEEK(53278)** will yield zero. Again, when you get a signal here, you'll know which sprites have bumped; and testing the location clears it, so that only new 'touches' will be shown on the next test.

A small comment here: these two peek locations are marked 'Interrupt'. Yet when such collisions occur, they are logged — and that's all. The word 'interrupt' has a special meaning to machine language programmers, and no interrupts seem to be happening. The programmer who wants interrupts to happen must enable the interrupt by storing the appropriate value into address D01A hexadecimal... and then write the appropriate extra machine language coding to make it all work.

This completes our roster of registers, but the plain mechanical facts don't convey the remarkable things that you can do with the Commodore 64. You gotta see it to believe it. □

Graphs Without Math?

by Jim Butterfield

Some of the newer Commodore machines make graphics much easier. Unless you had help via Simons' BASIC, **Super Sketch**, or some comparable system aid, you wouldn't even think of doing a graphics output from your computer data.

Now it's all changed. The newer machines — from the Commodore 16 and the Plus/4 up to the Commodore 128 — have extensions to BASIC that make drawing geometric figures quite straightforward. The Amiga's a special case, since its BASICs (both of them) and its graphics capability are effected in a new style; I'll concentrate on the 'traditional' machines for now.

On these, you get commands like: **DRAW**, to plot a point, or to connect points with a line; **CIRCLE**, to draw circles, arcs and ellipses; **BOX**, to draw squares or rectangles; and **PAINT**, to fill enclosed areas with solid colours. There are also lots of functions that allow you to ask questions, such as **RGR** (What graphics mode am I in right now?) and **RDOT** (Where's my drawing cursor, and what colour is it drawing?).

So all that fancy math can be tossed out of the window, right? Well, not exactly. Much more is within the reach of the average programmer; but there's still good reason to be aware of the arithmetic — and to know some extra math.

An example: You're preparing a simple forecast — fifteen per cent growth over six years — and your program looks like this so far:

```
100 graphic 0
110 dim d(5)
120 d(0)=5000: rem first
    value
130 for j=1 to 5
140 d(j)=d(j-1)*1.15: re
    m 15% growth
150 next j
160 print "business data
    --"
170 for j=0 to 5
180 print using "##  ##,
    ###.##";j,d(j)
190 next j
200 input "want a bar ch
    art";x$
210 if asc(x$)<>asc("y")
    then end
```

So you get a nice table of numbers running from 5000 to slightly over 10,000, and want to graph them. On the C-64, you'd need help. On later machines, you'd need to take care, but you might do a simple graph with coding along the following lines:

```
300 graphic 3,1: rem mul
    ticolour, clr screen
310 for j=0 to 5
320 box ,j*20+20,150,j*2
    0+35,150-(d(j)/100)
330 next j
340 char ,12,20,"press a
    ny key"
350 getkey x$
360 graphic 0
```

We've chosen multicolour mode and then used only one colour — but you can dress that part up if you wish. The real effort comes in changing those values ranging from 5000 to 10,056.79 into a scale that fits in the range of 0 to 160; here we've solved it by dividing the values by 200. Yes, there's a **SCALE** command that would convert the figures for us, but it doesn't solve another problem: coordinates are measured (going down) from the top of the screen, and we want to plot (going up) from somewhere near the bottom. That's what line 320 does for us, with the business of subtracting from 150.

The new commands have made everything much easier; but you can't get away from arithmetic. And other jobs call for mathematical understanding.

```
100 a=0:t=0:graphic 0
110 dim d(5)
120 data 125,195,305,388
130 for j=1 to 4
140 read d(j):t=t+d(j)
150 next j
160 print "division sale
    s:"
170 for j=1 to 4
180 print using "## ###
    <space>###.#%";j,d(j
    ),d(j)*100/t
190 next j
200 input "want a pie ch
    art";x$
210 if asc(x$)<>asc("y")
    then end
```

We've brought in a different kind of data this time — sales from four different divisions — and have presented the percentage contribution of each division. It's convenient to show this kind of result as a pie chart. The **CIRCLE** command certainly helps us draw a nice circle for the pie, although, unless we use **SCALE**, we'd better remember to give its size as 30 by 45 or it will look distinctly egg-shaped. But how do we draw those lines to split up the pie? Unless you know a little about the trig functions **SIN** and **COS**, you'd better be prepared to do a good deal of rough cutting. Here's a simple start to the job; later, you can think of how to **PAINT** the various sections in distinctive colours, or how to label them with the **CHAR** statement.

```
300 graphic 3,1: rem mul
    ticolour, clr screen
310 circle ,80,80,30,45
320 for j=1 to 4
330 a=2*π*d(j)/t+a
```

(Remember that the angle, *A*, in — ugh! — radians, must be the sum of all angles plotted so far.)

```
340 x=30*sin(a): y=45*co
    s(a)
350 draw ,80,80 to 80+x,
    80+y
330 next j
340 char ,12,20,"press a
    ny key"
350 getkey x$
360 graphic 0
```

There are sneaky ways to get around the use of **SIN** and **COS**, but they're much more work.

Don't be scared of it all. Play around and see what happens. Graphics is (are?) fun — and if a little math adds a lot of pizazz, go for it. □

ELECTRONICS 2001

AN AUTHORIZED COMMODORE SYSTEM DEALER SINCE 1979

NOW AVAILABLE 128 SOFTWARE

DATA MANAGER	WORD WRITER	SWIFT CALC.	CONSULTANT	PAPERCLIP
				

FOR YOUR COMMODORE 64

THIS MONTH'S SPECIALS

SIDEWAYS FOR C-64	THE MANAGER	IMAGE MAKER	OUBLIETTE	SPRINT PRINT
$39.95	$19.95	MPS 802 HIGH RES. PRINTING $10.00	$14.95	FULL SPOOLING PRINTER BUFFER SALE PRICE $39.95

 The complete guide to the 1541 operating system. $21.95	 $36.95	PaperClip (the word processor)  $69.95 with SpellPack $89.95	Boss Joystick  $21.95	Kraft Joystick  $15.95	Compuserve Starter Kit $44.95

DISKETTES SS/DD BOX OF 10 FROM $9.95

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.

Drawing with the Amiga mouse

by Chris Johnson

I am still not convinced that a mouse is the easiest person-to-computer interface, either to learn or to use, for most applications. But it is one of the best implements for drawing. And since my Amiga came with the aforementioned rodent, I made a drawing program my first project.

The Amiga comes with ABasiC, a language vastly richer than the BASIC 2.0 and BASIC 4.0 that most Commodore users are familiar with. (I shall not attempt to compare it with the much enhanced BASIC 3.5 of the C-16 and Plus/4, or the C-128's BASIC 7.0.) Unlike all previous Commodore BASICs, ABasiC does not have a full-screen editor. The line editor's commands are simple, however, and it doesn't take long to learn them.

Let's start this program by clearing the screen and setting up the screen colours:

```
10 rem **** clear screen
   and set colours ****
20 scnclr
30 rgb 0, 0, 0, 0
40 rgb 1, 15, 15, 15
50 rgb 2, 9, 9, 9
60 rgb 15, 5, 15, 6
```

The 32 colour registers are adjusted with the **RGB** command. The first number is the number of the register (0 to 31); the next three take a value in the range of 0 to 15 and represent the amount of red, green and blue respectively in the register. If all three are set to the maximum of 15, the colour in that register will be white; if all three are zero then it will be black. The values in between will give you something in between — try it for yourself.

Line 30 sets register 0, the screen colour, to black. Line 40 makes the text colour (and most graphics) white. Not the most exciting pair of colours, but after almost 20 years in the newspaper business I tend to think visually in black and white. Besides, I couldn't afford an Amiga RGB monitor as well as the computer, so I am using an amber monochrome monitor. I tried using my 1701 colour monitor with it. It was fine for watching graphics demonstrations, but the letters were sometimes rather blurred and the flicker was terrible. Lines 50 and 60 set the cursor and border colours.

Now let's consider the main routine. The guts of it are simply:

```
ask mouse xposition%, y
  position%, button%
draw (xposition%, yposi
  tion%)
```

The mouse, when asked, tells you its horizontal and vertical co-ordinates. It also tells you whether or not the left button has been pressed. (ABasiC cannot read the right button.) **DRAW** sets the pixel at the specified location to the **PENA** colour. All we have to do to make a drawing program is to loop through those two lines. However, we'll dress them up a bit first.

Before going any further, I'd like to point out one of the delightful features of ABasiC: long variable names. They take a little longer to type (and I used to use single-letter variables for that reason), but more descriptive variable names make a program much easier to understand if you have never seen it before or haven't looked at it for a while. Variable names can be up to 255 characters long, though only the first 31 are significant.

Now for the loop. We'll use a feature common to many BASICs, but not BASIC 2.0 or 4.0 — **WHILE** and **WEND**.

```
400 rem ** main loop **
500 while a$ = ""
510 get a$
600 wend
```

That's the framework for the loop. Run it: it will keep on looping until *a$* is no longer a null string — that is, until a key has been pressed. Add the guts:

```
550 ask mouse xposition%
    ,yposition%,button%
570 if button% then draw
    (to xposition%,yposi
    tion%)
```

Now try it. *BUTTON%* is true if the button has been pressed. Move the mouse and a line will be drawn only when the left button is pressed. We could have left out the **TO**, but then the line would be broken if you moved the mouse quickly. **TO** tells the computer to **DRAW** a line from the last position to the new one. The only problem with this occurs when you release the button and move the pointer across the screen. When you press the button again a line will be drawn from the end of the last line to the current position.

To remove that annoyance, we use another new command, **ELSE**. Type **EDIT 570 <RETURN>**. Now **X <RETURN>** will put you in insert mode at the end of the line. Add:

```
else locate (xposition%,
  yposition%)
```

and hit **RETURN** twice to get out of edit mode.

If the button is not pressed, the (invisible) pixel cursor will be relocated to the current position without drawing a line. In effect the button emulates **PENUP** and **PENDOWN** in LOGO or COMAL.

That's it — the basis for a drawing program. A keypress will take the program out of the loop so that commands can be acted upon. Our first commands will be *Quit* and *Clear Screen*, represented by **Q** and **C**.

When a key is pressed, program execution continues with the statement after the **WEND**.

```
700 rem *** commands ***
900 if a$ = "c" then scn
    clr
920 if a$ <> "q" then a$
    = "": goto 500
930 scnclr
940 end
```

We have left room for other commands to be inserted between lines 700 and 900. Let's try a couple more, *Erase* and *Draw*.

```
710 if a$ = "e" then dra
    w% = 0
720 if a$ = "d" then dra
    w% = 1
```

Since register 0 is the screen colour and register 1 the text colour, we can assign the value of *DRAW%* to **PENA**, which does the drawing. We'll put it right at the end of the command list in case we want to change the colour in any of the other commands.

```
910 pena draw%
```

What we have so far is fairly minimal, but it would be easy to expand. We could add dozens of commands, perhaps forsaking the keyboard altogether, letting the mouse do the walking. Commands for changing colours, filling in areas, drawing circles, rectangles and so on could be readily implemented. On this occasion, though, we'll be content with just one more brief set of commands. We add:

```
520 oldx% = xposition%
530 oldy% = yposition%
560 b% = button%
```

. . .and we make a major change to 570:

```
570 if b% then for i = 0
    to brushwidth% step
    density: draw (oldx%
    + i * xslant, oldy%
    + i * yslant to xpos
    ition% + i * xslant,
    yposition% + i * ysl
    ant): next
```

Now that's a long line! And ABasiC lets us do it — in fact we can enter up to 255 characters.

Since we have added multiple statements after the **IF**, we cannot use **ELSE**, so we put the relocation of the pixel cursor on a new line:

```
580 if b% = 0 then locat
    e (xposition%, yposi
    tion%)
```

In the commands section we insert:

```
730 if a$ >= "0" and a$
    <= "9" then brushwid
    th% = val(a$)
```

To prevent any possible mishaps, we initialize a few variables at the beginning:

```
100 rem **** initialize
    variables ****
110 xslant = -.5
120 yslant = .5
130 brushwidth% = 1
140 draw% = 1
150 pena draw%
160 density = 1
```

By hitting a number from 0 to 9, your brushstroke can now be adjusted, both for drawing and erasing, to one of ten widths.

Save your program (if you haven't done so already), with:

```
save "Drawmouse"
```

If you have saved it previously, update the disk with:

```
replace "Drawmouse"
```

□

BASIC09 under Super-OS/9

by Avygdor Moise

I remarked in last month's issue that Super-OS/9 has the richest and most versatile implementation of an OS-9 Level-I operating system, putting the SuperPET in the forefront of all 6809-based microcomputers. To fully appreciate the newly-acquired computing capabilities of the SuperPET, I wanted to demonstrate that the SuperPET actually executes programs faster under the control of OS-9 than it does under the two resident operating systems (BASIC 4.0 and Waterloo microBASIC).

As a test, I ran a simple prime number generating benchmark program, given to me by TPUG SuperPET librarian Bill Dutfield, and compared the execution speeds on the PET model 8032 BASIC 4.0 interpreter, the SuperPET microBASIC interpreter and Super-OS/9's BASIC09 interpreter. In evaluating the significance of the performance tests, it is important to note some of the differences and similarities among the three BASIC interpreters (see box).

The following is a copy of the benchmark prime number generating program, written in BASIC. This program was run on all of the above operating systems without any modifications. For the sake of fairness, all of the calculations were carried out in floating point format (32-bit mantissa and 8-bit exponent), and all extensions to the dialects were avoided, in order to eliminate any potential system dependencies.

```
procedure prime

1 rem use the date$ func
  tion instead of the
2 rem time$ function whe
  n running basic09
3 rem
10 print "Start time ";t
   ime$
20 for n=1 to 1000
30 for K=2 to 500
40 m=n/K
50 l=int(m)
60 if l=0 then 110
70 if l=1 then 100
80 if m>1 then 100
90 if m=l then 120
100 next K
110 print n;
120 next n
130 print "End time ";ti
    me$
140 end
```

It can be clearly seen from the listed results (see box) that BASIC09, which ran on the SuperPET under Super-OS/9, won the competition. It is also important to mention (although it is not shown in this test) that the execution speed of BASIC09 will not increase as a function of the program length, as is the case with Commodore BASIC 4.0 and Waterloo microBASIC.

Although very powerful, BASIC09's editor is not a full screen editor. However, the OS-9 user may use any editor he/she chooses (**scred**, **dynastar**, **stylograph** or **edit**, for example) to modify his/her programs, since all BASIC programs (if not squeezed) are stored as simple ASCII text files.

What makes BASIC09 and OS-9 two of the best software implementations on any microcomputer? Why does BASIC09 (and any other program that runs under the control of OS-9) execute fast and efficiently? These are the questions I will attempt to answer in next month's article. □

Feature Comparison: Three BASIC Dialects

	CBM BASIC 4.0	Waterloo microBASIC	OS-9 BASIC09
Operating system	CBM BASIC	Waterloo BASIC	Super-OS/9
CPU type	6502	6809E	6809E
CPU speed	1 Mz	1 Mz	1 Mz
Usable memory	31K	30K	25-35K
Structured	no	yes	yes
Parameter passing	no	yes	yes
Modular	no	yes	yes
Save/load format	PRG-Tokens	ASCII/Tokens	ASCII/I-code
Variable types			
Real	yes	yes	yes
Integer	yes*	yes	yes
String	yes**	yes	yes
Boolean	no***	no***	yes
Byte	no****	no****	yes
User defined	no	no	yes
Formatted I/O (print using)	no	no	yes
Execution speed (relative to BASIC 4.0)	1	0.55	2.45

* BASIC 4.0 always converts integers to real numbers during intermediate calculations.

** BASIC 4.0 strings are limited to 255 characters in length.

*** BASIC 4.0 and microBASIC will allow integer variables to be used in boolean algebra.

**** A byte data type may be simulated by the use of the BASIC 4.0 **ASC** function and the microBASIC **ord** function.

Test Results

	Execution Time	Execution Speed Relative to BASIC 4.0
BASIC 4.0	**1123**	**1.00**
microBASIC	**2050**	**0.55**
BASIC09	**495**	**2.45**

The 3-board bug

by Avygdor Moise

TPUG has received an increasing number of telephone calls from 3-board SuperPET owners who report problems with their SuperPET computers. The initial complaints originated from OS-9 users who could not get their newly-acquired Super-OS/9 operating systems going. After a prolonged investigation, we have learned that the problem was not created by OS-9, but rather was brought to light by this new operating system.

For those among you who own a 3-board SuperPET and have so far not experienced any problems, let me describe the symptoms that have led to these complaints, and what causes them.

The symptoms

- After acquiring Super-OS/9 and following the MMU installation instructions, the SuperPET will not run OS-9. No other side effects are observed — the 8032 mode and the Waterloo software still work fine.
- After prolonged use of some software that accesses the SuperPET's extended memory (64K bank-switched RAM), it seems that the data gets corrupted and the software 'crashes' frequently.

The cause

The SuperPET has 96K of RAM, which is partitioned into two sections. The first section is the familiar 32K block, starting at hexadecimal 0000, and ending at hexadecimal address 7FFF (decimal 32767), providing the user with 32K of contiguous memory. The second section is a 64K block of memory that is subdivided into 16 segments, 4K each. Each of the 16 segments is referred to as a 'bank' of memory. The programmer may access any *bank* of the extended 64K memory by poking the bank number at hexadecimal address EFFC (decimal 61436). Any byte within a selected bank may be accessed by reading (peeking) or writing (poking) data from/to the hexadecimal address range 9000-9FFF (decimal 36864-40959).

If this sounds complicated, do not worry too much about it, since the Waterloo software, word processors (like **PaperClip**) and Super-OS/9 all make the bank switching completely invisible to you.

For the bank-switched memory to work (that is, to be able to store numbers and let you retrieve them at some later time), all of the 64K memory needs to be accessed periodically (refreshed). The hardware in the SuperPET was designed to cycle through each memory byte to ensure that there will be no memory loss due to insufficient refresh. Unfortunately, a design error was made by the creators of the memory section of the 3-board SuperPET. This error prohibits refresh to the 64K memory section when a program executes code from the bank-switched area. The hardware bug was fixed on the newer releases of the 2-board SuperPET, which became available shortly after the SuperPET was introduced.

The solution

The obvious solution to the problem is to contact Commodore International and ask them to take back the faulty SuperPET boards (two) and replace them with a working equivalent single board. Our initial experience with Commodore Canada indicates that they are unlikely to do this unless you really insist, and even then will charge you a couple of hundred dollars.

TPUG has paid a hardware engineer to come up with a fix, and one has been found. The only problem is that the two chips needed to correct the problem can only be obtained from an electronics ditributor in California, and they are in short supply. We anticipate having a first running prototype (fix) some time in January, if the chips are delivered as promised. □

TPUG Conference 1986

Planning for TPUG's Fifth Annual Conference is well under way. The dates to circle on your calendar are May 24 and 25, 1986. The conference will be held, as it was last year, at the Ontario Institute of Studies in Education at 252 Bloor St. W. (Bloor and St. George), in Toronto. As usual, the conference will feature a two-day program of speakers for beginners and experts on all aspects of Commodore computing, including seminars on the new machines like the Commodore 128, the PC10 and PC20, and the Amiga.

Among last year's speakers were Jim Butterfield, disk expert Gerry Neufeld (author of *Inside Commodore DOS*), Super-OS/9 guru Avy Moise, Frank Covitz (co-creator of **Sky Travel**) and well-known Commodore writer Elizabeth Deal. Next year's roster is not yet finalized, but is certain to be equally stellar. We'll have more details next issue — stay tuned.

Anyone For WYSIWYG?

by Ian A. Wright

There's a new 'buzzword' going around among users of word processors — it's WYSIWYG (pronounced 'wiz-ee-wug'), which stands for What You See Is What You Get. It means that the screen display accurately shows what will be printed out on paper — including superscripts and underlining. WYSIWYG programs are the latest in a long line of word processing improvements since the days of the earliest line editors.

Surveys have suggested that up to ninety per cent of home computer owners (other than gamers) use their equipment for word processing in one way or another. These people range from the occasional letter-writer to the professional screenwriter or magazine editor. The former wants ease of use; the latter needs access to a wide range of functions. Some users may want to have more than one word processor available — a simple one for memos and letters, and a full-featured one for producing major reports. How have we ended up with such a variety? Let's look back and see how the word processor has evolved. Then we can gaze into our cloudy crystal ball for some ideas of what we might expect from programs in the near future.

The first microcomputer word processors were line editors — like **Easy Edit**, **LED**, or the program **Teddy**, written in BASIC by Brian Morrow for the Commodore PET, in 1982. **Teddy** was a simple sentence-oriented word processor that allowed inputting a series of 'sentences' of up to 254 characters. This text could then be manipulated using insert, duplicate, delete, find and replace, change, caps lock, tabs and paging commands. The printing function allowed continuous page output, variable pagination, margins and tabs; and even features like underlining, centering and expanded printing. The program supported both tape and disk storage. All of these functions were part of a public domain program that allowed 800 characters to be manipulated in an 8K PET. And it was *free*!

In those early days, if you had around four hundred dollars to spend, you could buy **WordPro** for your PET/CBM and have the ultimate in word processing software for the home computer. It was marvellous. In 1981, I used **WordPro 3+** to write a small book that required seventeen separate text files — each of which was edited many times, then linked together to make the finished book. Without **WordPro 3+**, I would have packed it in on any number of occasions before the final draft was ready. Searching and replacing, changing the spacing and margins, footnoting... all these functions of **WordPro 3+** made the book possible. When it came time to show someone the fruits of my labour, however, I used a ream of paper to get the final printout, because small formatting errors were invisible until they were printed — and fixing one error often produced others further on. A neat finished product was difficult without any video or screen output. Using **WordPro 4+** or **PaperClip** solved that.

...Simpler is often better — and it's always cheaper...

All the commonly-used Commodore word processors, such **WordPro**, **Wordcraft 80**, **Superscript**, RTC's **Script 64** and **PaperClip**, offer excellent screen editing similar to that of the resident Commodore screen editors. You have taken for granted that you can zip around the screen and change words or format instantly. You are not tied to editing only one screen line at a time, as you might be with other microcomputers... and you can skip to the last paragraph, and change it as easily as the first one. However, you cannot see double spacing, centering, or subscripts on the editing screen, because these programs use 'post formatting' rather than WYSIWYG. In fact, there are two distinct types of screen available in present high-quality word processors — the editor and a video previewer. The video output screen shows you how the text will look on paper — double spacing the text, centering the titles, and using reversed characters to show some of the enhanced printing features — but it's not WYSIWYG.

The latest (C-64) versions of these word processors allow you to use alternate character sets, see the actual 80-column printout on your monitor, produce text printed in columns side-by-side, manipulate numeric charts, check for spelling errors, and so on. In fact, the current crop of word processors for home computers are so well-developed that they have been used for many business and professional word processing applications. But *this* is not WYSIWYG, either.

I have been using some of the MS-DOS word processors on a Commodore PC-10, lately, and there's a whole new world of features available on these 'new' machines. I am writing this article using **Wordperfect 4.0** by SSI Software, and its features are truly amazing.

Double spacing, underlining, boldfacing — all appear on screen as you write. The text is word-wrapped to match whatever margins and tabs you set — and you can reset these and other page formatting commands anywhere in the document, then watch the screen reformat before your eyes. You can write up to five text columns at a time. If you edit the columns on the screen, the column breaks alter accordingly. The screen does not show the format commands unless you request that they are displayed, so the only interruption on the screen is a half-line status prompt at the lower right. You have twenty-four unbroken text lines to edit. This *is* WYSIWYG, because you edit, revise and see the effect on the screen without invoking a separate video mode.

For the scholar, footnoting is a breeze, since **Wordperfect 4.0** calculates when to place your footnote on the page for you. You can select how your footnote is to appear, then go back later and edit both the footnote and the format. If you change footnotes (deleting or adding to them), they will automatically be renumbered.

For the businessman who has to prepare reports each quarter, the program can save a number of page and printing formats so that next month's report is called up with only one key press, and will only have to have the new data entered. High security files can be passworded easily. Columns of numbers can be totalled — and new columns can be generated, using the four built-in math functions. Print spooling will allow you to work on another file while the previous one is being printed, and the dual-document editing feature means that you can alternate between two different

pieces of writing at the press of a key. This will let you call in prepared or 'boilerplate' text from the other file.

For 'power users' like magazine editors and scriptwriters, there are macros. Macro-instructions have been part of better word processing software for some time. A phrase definition is tied to a short series of key strokes (for example, 'Toronto Pet Users Group' might become **ESC-T**; or a series of tabbing and indenting commands might become **ESC-***). In addition, **Wordperfect 4.0** extends the macros to execute commands, write text and chain the actions to occur recursively. For example, you can replace double spaces occurring after a period with single spaces, and do this repeatedly for a number of files.

Another of these 'super-macros' might be an opening format designed to use changes in underlining, spacing, printwheels, et cetera, called up using **ALT-F** each time you use the program. This is possible because your macros become a permanent part of your personal version of **Wordperfect 4.0**.

Automatic elimination of 'widows' and 'orphans' means that **Wordperfect 4.0** is the answer to a writer's dreams. No more checking to see if the first or last lines of a paragraph are left dangling on their own page — the program does this for you. You can access two separate files and pass material from one file to the other. This feature is much appreciated when composing written material at the keyboard. I have a point-form outline of this article as **Doc 2**, and I can refer to it, or cut-and-paste from it, easily. Some current word processing programs come with a spelling checker that allows you to make corrections without leaving the main program — this is a useful feature. **Wordperfect 4.0** comes with a built-in 100,000 word dictionary.

The current 'state of the art' word processing programs are not for everyone. If you use a word processor infrequently, and have no need for some of the more esoteric functions, then save your money, because full-function programs are expensive. Save yourself the time you would have to spend learning how to use these complicated programs. A simple menu-operated program may cover all your word processing needs. Save yourself the expense of having to replace your current hardware and peripherals unnecessarily. There is a wealth of uncomplicated, inexpensive and time-tested software available, and you can avoid a lot of grief — especially if you're a novice unsure of your word processing needs. You don't have to spend four hundred dollars or more on a full-featured word processor, when forty-nine will give you all the functions you need. Simpler is often better — and it's always cheaper.

Does this mean that word processors have — in the words of the song — 'gone about as fur as they can go?'

No way!

In the near future, inexpensive home computer word processors will be able to insert photographs and produce graphics — just like the Macintosh can now. The use of windows will become standard. You will be able to edit more than one text area and see all of them on the screen at the same time, and they will be linked so that changing one will make changes in the others. The Amiga's windowing routines are just one example of possible trends in this area.

. . . Automatic elimination of 'widows' and 'orphans' . . . is the answer to a writer's dreams . . .

Multitasking is going to change the face of word processors because you will have a number of programs running concurrently. Grammar checkers like **Punctuation and Style** and **Grammatik** will read over your material for common grammatical errors. These programs can help pinpoint consistent errors in language and, when run along with your word processor, you will have an English tutor online. Rather than importing a graph or picture, you will be able to flip to another program, select a set of statistics, and have that program draw the graph while you return to the word processor, which has been saving your current file after checking for grammar and spelling errors. While this has been going on, a series of form letters have been printing out from a concurrent mail-merge program. When the graph is ready, you place it on the screen and size it to fit the columns you're typing; then add the text and a footnote. Impossible? It's being done right now, using megabuck business systems. Soon it will be in your home.

Future word processing software will de-emphasize the keyboard, and replace many of the current command structures with dedicated function keys, mouse controllers, trackballs or other input devices. You will be able to select the device most suited to your present activity.

There is no reason for block editing commands to continue to be restricted to linear movements — down and across movements make little sense when using cut-and-paste operations on clip art or graphics. So-called 'intuitive' editing is an area that is lacking in most current home computer word processors.

The screen display of soon-to-be-available word processors will allow a variety of character fonts to be displayed so that, as you would with the Macintosh, you will be able to change both the size and shape of letters used within a document. This will allow you to create letterheads or pictorials as part of a text file. Graphic display, however, will not alter the need for true letter-quality printing for business letters. As laser printers drop in price, the distinction between business and data quality will be one of software decisions, rather than one of hardware restrictions.

Word processing has come a long way in a few short years, from line editors to WYSIWYG programs. The ability to write and revise without extensive rewriting is what makes word processing appear to be magic. What manufacturers should work towards next is a way to get rid of all this keyboarding! □

The Ultimate Word Processor

by Steve Punter

As the author of a well-known word processor (**WordPro**), I have often thought of what the 'ultimate' word processor would be like. Without becoming bogged down with such issues as available technology or cost, let us assume we are to have access to a machine with any capabilities we deem necessary. Let us also assume that this machine runs at *extremely* high speeds, and so is capable of doing the most complex operations in reasonable lengths of time.

I don't pretend to have the world's greatest imagination, so if you can think of more fanciful things than I, then all the more power to you. With that out of the way, let's start imagining the 'World's Ultimate Word Processor'...

The Display Area

The display area should be capable of displaying a *complete* printed page, which should be at least 160 characters in width, and 88 lines in length. 160 is the maximum width of most printers, and conceivably the widest you will ever practically need. Some may argue this point, but since we're only imagining this, it's hardly a major issue. 88 lines is the length chosen for the page, to allow for the use of 8 lines per inch, on an 8 1/2 by 11 inch sheet of paper. It must be capable of displaying exactly what will appear on the printed page, and that includes full proportional spacing.

Current daisy wheel printers have movement increments of 1/120th of an inch, yet there are other types, especially laser printers, which can move in much smaller increments. In order to facilitate the exact size of any given proportional character, our display screen resolution should be more than up to the task. To allow for this, each character should be at least 16 pixels wide. Since we are allowing up to 160 characters across the screen, that would require a horizontal resolution of 2,560 pixels (160 by 16).

We should also allow at least 16 pixels in the vertical plane of each character to allow for true descenders, super and subscripts, et cetera. Since our page must be at least 88 lines long, we would need a vertical resolution of 1,408 pixels (88 by 16).

Character Set/Keyboard

On a rudimentary level, the display screen should be capable of showing underlined, boldfaced, italicized, superscripted and subscripted characters as they would appear in print. It should also have a full complement of business and mathematical symbols, as well as all known accents. Even the ultimate word processor couldn't have *every* character in the world, so it should supply the operator with a convenient way of defining his or her own characters.

The keyboard should be laid out *specifically* for the word processor, making the entry of this vast array of characters as easy and as natural as possible. It should have keys specially set aside for the user-definable characters, and a convenient way of marking them.

Editing Capabilities

Let's face it, this word processor should be able to do almost *anything*. Not just simple 'cut and paste' operations, but every conceivable text modification you could think of. I could spend pages and pages harping on the features I consider the most useful, but that would be a bore. None the less, the program should be capable of simple tasks such as moving single words, sentences or paragraphs to new locations; and capable of restructuring text to suit the changes. Since the program will be page-oriented, it should be just as capable of moving text from one page to another while maintaining complete page structure throughout the entire document, regardless of length.

Text Size

Since the ultimate machine can have as much memory as we please, we would obviously like it to be capable of storing a complete document, regardless of size, in available memory. This would be unnecessary, though, if we had a *very* fast and capacious disk drive. The word processor need not have more than a single page of text in memory at one time. If it is capable of moving text on and off the disk at high speeds, we will not perceive the difference between a disk-based system and a RAM-based system.

To protect us against power surges or failures, the machine should have a power backup capable of running the entire system for the *full* duration of the power outage. The word processor should also come with a work light.

Spell Checking

Our dictionary should contain *every* known word in the English language: somewhere between 300,000 and 500,000 words. In Canada, it should also come equipped with a complete French dictionary. But, since we are talking 'ultimate', it should contain dictionaries for *every* language spoken on the face of the earth.

Of course, certain Oriental languages do pose a problem for the entire system, since their 'character sets' are too large to be accommodated by a useful keyboard. Experiments in other parts of the world have met with little success in this area.

In addition to checking your spelling

and supplying you with a list of *possible* correct spellings, the system should also be able to recognize proper word use. For instance, the sentence 'I left the disk over their on the desk' consists of legitimate English words only, but the use of 'their' is incorrect. The 'ultimate' word processor should be able to 'see' this.

Communications

Each separate word processor should be completely able to phone another, should its user opt to be part of a worldwide network. The 'ultimate' word processor could automatically call up other 'ultimate' word processors to send or obtain necessary files. The machine should be capable of multitasking so that these calls could be made to a word processor already in use without any detectable interruption. The manufacturer of the word processor might also opt to have a central computer to operate this network.

Printer Compatibility

There is an almost infinite number of printer code standards in the world, yet the 'ultimate' word processor must be capable of adapting itself to *all* of them. Having written more printer modules than I'd like to remember, I can attest to how difficult a task this would be unto itself, but after all, we are talking 'ultimate', aren't we?

Another thing that isn't as standard as the industry would like: printer bus structure. Although many printers use the 'Centronics Parallel' type, many do not. The 'ultimate' word processor should have ports for all known printer interfaces.

Odds and Ends

I could go on and on talking about specific things that this program *should* be able to do, but that would get time-consuming. By simple definition, the 'ultimate' word processor should be wanting for nothing. It should literally do anything we could possibly imagine, and then some. We don't yet have an 'ultimate' word processor, because we don't have the necessary hardware (at least, not at a price anyone could afford). Perhaps one day, when computer technology becomes infinitely more complex, we won't need a word processor at all — we'll just have our robot secretary (or robot housekeeper) print us up what we need by merely dictating to them. Who knows: the future is where we can all let our imaginations go wild. From what I can see, enjoy the present, *look forward* to the future. □

Computer Barn-Raising

by Mike Martin

Tightly-locked doors, bars on windows, guns in the bedroom, and general distrust seem the new order of the day. The world is changing, and those who remember the old ways are sad at the 'progress' our society is making. As school children, we read of the warmth and friendliness of pioneer society. The climate and elements were the enemy, not our neighbours. People banded together for survival. Locked doors were not needed. When a neighbour's barn burned, the community would hold a barn raising as a social event. If a family was in poor health, the neighbours would help with the planting or harvesting. In current times, social agencies have not replaced neighbours in solving these type of problems. People helping people is still the only answer.

New technology has been considered part of the problem. It is credited with bringing on the isolation and selfishness that prevail today. However, the new home computer technology is bringing back the past. The complexity of this new technology has made it once again important for people to help people. Users' groups are filling that need. They are necessary for this new kind of survival. We are reluctant to read instruction sheets packed with new clocks or coffee machines, much less the information by the pound packed with computers and software. With all the codes and differing systems included in software and hardware, the beginner finds a wide gap between reading and understanding the information.

To make matters worse, each software package uses different keys and commands for similar functions. Software is sold, more often than not, in a sealed package, with no opportunity to try out the product before buying. So having helpful neighbours is not a luxury, but a necessity.

Users' groups have brought back the town meeting, the barn-raising, and helping hands. The atmosphere at a meeting is warmer and more cordial than at political rallies or even church socials. In users' groups more than in any other social situation, the age barrier vanishes. Two members sixty years apart in age hold a discussion on joysticks. Couples can be overheard explaining concepts to each other. And as often as not, it is the wife who takes notes, while the husband stares at the ceiling. I have seen three generations of a family all asking questions and offering advice. Any problem brought up by a new user is wrestled to the ground by five other members who have already beaten it into submission.

The new version of barn-raising involves giving a new user a good wedge and disk utility, along with a dozen whiz-bang public domain programs to replace the lonely **READY** prompt that flashes on his blank screen. Each library disk becomes a treasure chest to be explored. The 'old timers' (anyone with a computer that is over six months old) delight in sharing software timesavers, as well as all the advice needed to run it.

The meetings start with an hour or so of questions and answers, news, rumours and gossip about new products and stores. A break provides time to talk one-on-one with people who have that program you are unsure about buying. Then another hour of demonstrations, reviews and classes. BASIC, machine language and 'exotic' dialects are taught, but never as *expert* and class, or *teacher* and students. Everyone participates. A reluctantly-asked beginner's question might open up a wealth of tips benefitting everyone in attendance. The meetings are dependent on each member, and are shaped as much by who stayed home as by who attended.

The future seems to be bringing back the past: past values, old-time meetings and old-fashioned communication. People are talking to each other again — and even listening. And what will the computer future bring back to us next? Hayrides? Picnics? A computerized square dance? Bobbing for apples? □

Micro Processes

Bips And Bauds

by Ajay Jindal

In this world of ever-changing computer jargon, it is very easy to misuse terms. In an effort to seem fluent in computerese, many people will throw around words without knowing their proper uses. Of course, keeping on top of everything is a full-time job in itself, but all it takes is a few misinformed people to lead many astray. One such misused term is the word 'baud'.

Anyone who knows anything about modems and telecommunications knows that this term is a measure of how fast a computer can send and receive data. In essence, this is true — except that it is not a measure of actual data (like bits and bytes), but of communication elements. The proper term to use is bits per second (BPS).

Let's start by examining the bottom of the barrel. Most modems today are 300 baud. That is, they can receive and send data at a rate of 300 bits per second (BPS). In this case, BPS and baud can be used interchangeably. When the modem wants to transmit a 1, it sends out a certain frequency; and when it wants to send a 0, it sends out a different frequency. 'Baud' refers to the maximum rate that these frequencies can alternate in a second, and BPS refers to how much data is actually sent in that second.

On to 1200 BPS (more commonly, but incorrectly, referred to as 1200 baud). Here, there are four different frequencies. The first means 00, the second means 01, the third 10 and the fourth 11. The maximum rate at which these four frequencies can shift is 600 times per second, therefore it is 600 baud. But since each frequency transmits two bits, the data transfer rate is 600 times 2, or 1200 bits per second.

To clarify this, let's use a similar situation where we have two men walking: The first man takes 300 steps every minute, and each stride is one metre long. We can say this person walks at 300 steps per minute (baud). We can also say that he walks at 300 metres per minute (BPS). Now the second guy takes 600 steps per minute, but because he's quite tall, each of his strides are two metres in length. His 'baud-rate' is 600 steps per minute, but since each step is two metres long, his actual velocity is 1200 metres per minute. Modems work in frequencies, bits and seconds, instead of steps, metres and minutes.

The newest thing in modems is one that operates at 2400 BPS. Ordinary telephone lines can only handle about 900 baud, so how do they manage 2400 BPS? Well, the actual operating speed is 600 baud, but each frequency represents 4 bits. That means there are 16 different frequencies. One each for 0000, 0001, 0010, 0011, 0100, . . . , 1111. This guy's a giant that takes 600 steps in a minute — but each step is four metres long!

It is interesting to wonder why baud caught on. Of course, somebody said 'baud' when they meant 'BPS', and everyone else just followed. But my theory is that 'baud' is easier to say than BPS. 'Baud' is only one syllable, and doesn't require the intricate oral contortions that BPS requires. BIPS (BIts Per Second) is an easy word to use, so let's designate BIPS as the official term with which to measure data transfer rates. We could be the ones to enlighten everyone else, and bring them out of communications ignorance! □

Backup Safety Scheme

by Karl T. Thurber Jr.

The first time you lose a great deal of valuable data such as a large mailing list or your computer-based chequebook, you'll appreciate the value of backing up your data regularly. I, too, learned when I lost the data on a disk containing the complete listing of my disk collection.

Get in the habit of backing up valuable data disks every time you make a significant change to the disk, using an appropriate disk-copying program. Use a 'grandfather system', in which you maintain on hand *three generations* of data disks. In this system, you have disks #1, #2 and #3. Assuming that you're about to work with disk #1, first back up disk #2 to disk #3 (assuming also that when you first used disk #1, you created disk #2). After you finish working on #1, then copy it over to #2. The idea is to have three copies around, to avoid the total loss of data that could result if you fouled up in copying any two of the disks. When copying, be sure to keep a write protect tab on the source disk (the one to be copied from)! A piece of masking tape over the little notch in the *side* of the disk will do the job. It can easily be removed after copying, and won't damage your disk.

If your sessions result in large amounts of data being created, the loss of even the primary (#1) disk could represent a good deal of retyping. Thus, you may find it worthwhile to obtain a copy utility (such as **The Clone Machine** or similar software) that also allows you to reformat and copy *single tracks*, as opposed to only whole disks or files.

If your data disk develops a bad spot, such as from experiencing a disk error or power loss while entering data, and this bad spot prevents you from accessing the file, you can often copy over just the bad track from the backup disk to the original disk. If you're lucky, the new data you added in the most recent 'lost' session will be on another part of the disk, and the overall data loss won't be catastrophic. Owning and knowing how to use a good disk editor utility program is also excellent insurance against such data losses. □

The Underline Truth

by Paul Blair

Letters that I get voice an oft-repeated plea — how does one underline on a CBM 1526 printer? The writers all bemoan the fact that, while many other printers can do bells, whistles and sundry other clever things, the 1526 (which has a whole heap of nice functions *not* included in later CBM printers) can't set off reports with nice lines under headings, and so on.

Well, you *can* underline on the 1526. It takes a bit of work, but I had my hands on a 1526 for an afternoon, and I found I was able to make it do some useful things. My printer paper supply is a bit less than it was, but it was fun. Anyway, it was raining...

I started off in BASIC, because that gave me a wider range of features to play with.

The key to the underlining procedure lies with **CHR$(141)**. That little letter feller tells most printers to do a carriage return *without* a line feed. Cutting through the jargon, the print head is returned to the beginning of the line it is printing, without winding the paper up a line. On the 1526, this 'beginning' is usually just adjacent to the last character printed.

Can you imagine what that can do for you? ...Nothing?

Nothing, heck! Try overprinting on the same line, to create shades from white through to black, and you will be able to do some very impressive graphics. In fact, most students of computing are given an exercise of this kind at some time or another.

Back to underlining (where we came in). By printing some characters, then a **RETURN** without linefeed, we find ourselves poised to put the print head over the text just printed, and print an underline character.

This short BASIC routine demonstrates one technique. After opening a channel to the printer and defining what is to be printed, lines 40 and 50 print the string (one character at a time), print a **CHR$(141)** to stay on the same line, then pop an underline character (**CHR$(228)**) under the last letter printed.

```
10 open4,4
20 a$="underline test - cbm1526"
30 forx=1 tolen(a$)
40 print#4,spc(x)mid$(a$,x,1)chr$(141
   );
50 print#4,spc(x)chr$(228)chr$(141);
60 next:print#4:close4
```

The printer gets pretty busy during all this. Watch the printer in action — it will show you, better than I can tell it.

Now, let's tackle underlining from a word processor program. I will show you how with **EasyScript**, because most of you will probably have some familiarity with its use.

Section 8.2.11.2 of my **EasyScript** manual defines 'Special Characters'. Briefly, you may define ASCII codes that are not obtainable from the keyboard, and access them from within your text. To set up the characters, you press **f3**, then enter your definitions. This will show up on the screen like this:

```
*0=33:1-141←
```

where Key 1 is set for **CHR$(33)**, and Key 2 for **CHR$(141)**. To use them in the text where you want them, the sequence is simply **f1** and (in this case) 0 or 1. The screen will show the number 0 or 1 in reverse.

Try this little sample. The command line is first. Key 1 is the carriage return without linefeed, Key 2 is the underline, and Key 3 forces the lower case. I'll explain the last one later on.

The second line has been specially typed for clarity. The (1) and (222), et cetera, are each individual sequences of **f1** and the numeral 1 (or 2 or 3) from the top row of your keyboard. The left-arrow character is the **RETURN** symbol you usually get when you press **RETURN**.

```
<F3>1=141:2=228:3=17←
<12 spaces>TITLE(1)<12 spaces>(222223
  ) of your page←
←
The next line goes here.←
```

Output this to your 1526. It worked for me, so it should work for you.

The **CHR$(17)** is a forced return to upper case/lower case mode. Without it, the words 'of your page' would print in upper case.

That's the good part. The bad news is that if you only want to underline some word or words in the middle of a line, you are going to need to do some work. You will have to determine where each word to underline falls in the printed line, because you will have to space across (in theory, at least) from the left-hand margin of the line to where the underline must start. By using output to the screen to check how things get set out, however, you will be able to work it out quite easily. A minor price to pay for such a gem!

Well, that's it. The technique is easy to use, and becomes even easier with practice. The output does suffer a bit, because the underline sits directly below the normal character. It would look better with a one-dot spacing down from the line, but the character design in the 1526 won't allow it. □

1525 Ribbon Re-Inking

by Karl T. Thurber Jr.

The cartridge ribbon on the Commodore 1525 printer is becoming increasingly expensive and difficult to find. Tiring of frequently replacing the ribbon, I hit upon a fairly effective method of prolonging its life.

Actually, several approaches or combinations of approaches may be used. You can obtain a supply of ordinary liquid stamp pad ink (special printer ribbon ink would be even better, if you can obtain it), and load a small quantity into a syringe or tiny eyedropper. You then carefully pry open the cartridge with a

small screwdriver, and 'inject' the ink into the roller sponge in the left-hand ribbon cartridge with this fresh supply of ink.

You can also use a stamp pad ink refiller of the type that has a 'roller ball' on it, similar to the roller used on some brands of deodorant sticks. You can gently roll this refiller along the ribbon to gradually re-ink it while the ribbon is in motion, thereby indirectly transferring the ink to the roller sponge in the ribbon cartridge.

If the ribbon is dried out, you can often rejuvenate it by squirting a minute quantity of WD-40 or similar lubricant into the roller sponge. Alternately, you can place a small quantity of lubricant on your fingers, and rub it directly onto the ribbon while the printer is in use and the ribbon is in motion.

The procedures described here are messy, and some experimentation is needed to develop the right 'touch' in re-inking these troublesome ribbons. Naturally, you should avoid over-inking, or using too much WD-40, or you will clog up your printhead. If your best efforts fail, most Radio Shack stores carry the #26-1424 Line Printer Ribbon Cartridge ($8.95 US) — a dead ringer for the Commodore ribbon.

Needless to say, 'printer heaven' is owning a printer that uses a standard typewriter ribbon! □

B-128 Memory Expansion with the Calc Result Cartridge

by Elizabeth Deal

Want to put some memory in bank 15, but can't figure out how to do it in a quick way? Well, this article shows how to add 8K. In fact, we'll do better: we'll have a deluxe 8K package of protected memory.

It all began with the **Calc Result** cartridge, which Protecto sells under the generic name of **Spreadsheet**.

Snooping reveals that the **Calc Result** cartridge uses one chip (6000-8000 hex, an 8K EPROM). *Wouldn't it be neat*, I said to myself one day, *if I just extended their existing lines for plugging in my extra memory?* A phone call to Howard Harrison in Philadelphia (an expert in this sort of thing) got things moving.

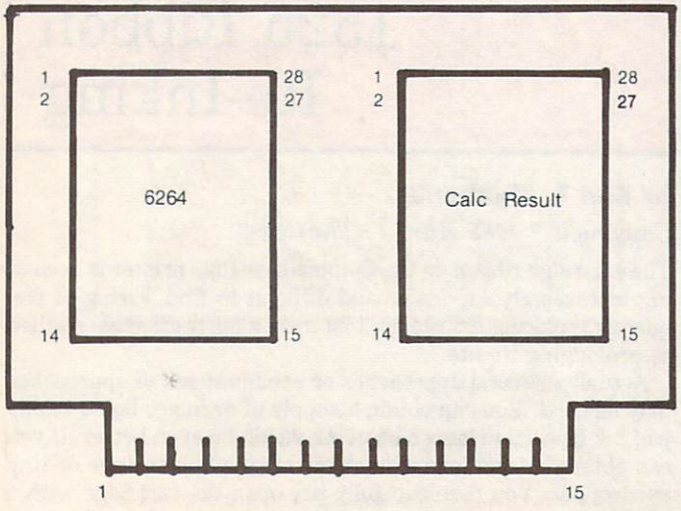
Having taken the cartridge apart, he found that the **Calc Result** people had thoughtfully provided just about everything: the wires and chip-select decoding are all built in! The only thing needed is a 28-pin, 8K static memory chip (#6264, for instance); and a bit of wire to enable the write-line. Here's how.

Be warned! Following these instructions will likely void your spreadsheet warranty. Proceed at your own risk. (This has been tried and tested by two people, but we cannot assume any responsibility for your mishaps.)

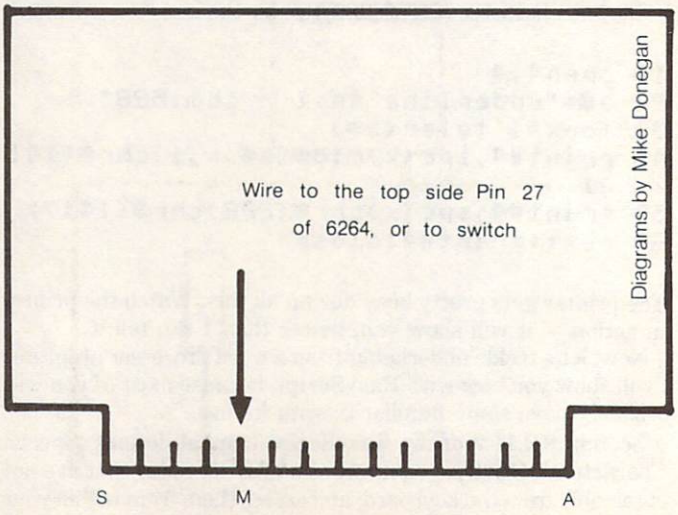
Instructions

- Open up the cartridge, using a kitchen knife, or similar. This has got to be the hardest part of the job. (The temptation to use a hammer is great, but the **Calc Result** inside is too valuable.)
- Remove the **Calc Result** chip by prying it gradually from two sides with a screwdriver, or some other blunt object. (We're doing this only for its safety.)
- Using a desoldering tool, carefully unplug fourteen pairs of holes, so that a socket can fit in the holes later. While you do this, you'll have plenty of time for thinking about why they are plugged up in the first place.
- Put in a low-profile 28-pin socket: for example, Radio Shack #276-1997. Solder all the pins very carefully to the circles surrounding them. Be careful not to mess up here — keep the solder very close to the pins.
- Take about four inches of a very thin, *flexible* stranded wire. Strip about 6 mm of it, twist, and bend into a 1.5 mm loop. Tin it. Squeeze it flat. You have to connect it to the read-write line (M) on the bottom of the cartridge expansion board. Put a tiny bit of solder on the cartridge finger, as far back into the board as you can (otherwise it won't plug in). Put the wire on it, and solder in place. It's tricky to connect to a flat surface, but it can be done. Make sure that the wire does not touch any adjacent 'fingers', or some other soldered place.
- Bend pin #27 (the read-write pin on the 6264 memory chip) flat out. Put the chip in the socket with the notch furthest away

Top View



Bottom View



from the computer. Pin 27 does not get plugged in (that's why you bent it out).

• If you do not want the deluxe package, connect the wire from M to this pin, plug the **Calc Result** chip back in, and you're in business. Otherwise, read on...

8K deluxe RAM-ROM package

Hmm, I mused: *If the read-write line is not used by the EPROM, then wouldn't it be possible to install a switch that would protect my new memory from being written over?* I could load **Supermon** and nothing — not even my own bad code! — will write over it. Time to call Howard again...

Howard finds the idea amusing. I assure him something like that was done on the PET. Okay. We need a tiny on-off switch and a 5K-ohm resistor. Howard explains that you can't just leave a line dangling in an unknown state. I thought that switches were either on or off (line open for read-only, or closed for reading and writing), but it turns out that there is a third position — unknown state. We can't have that. Hence the resistor.

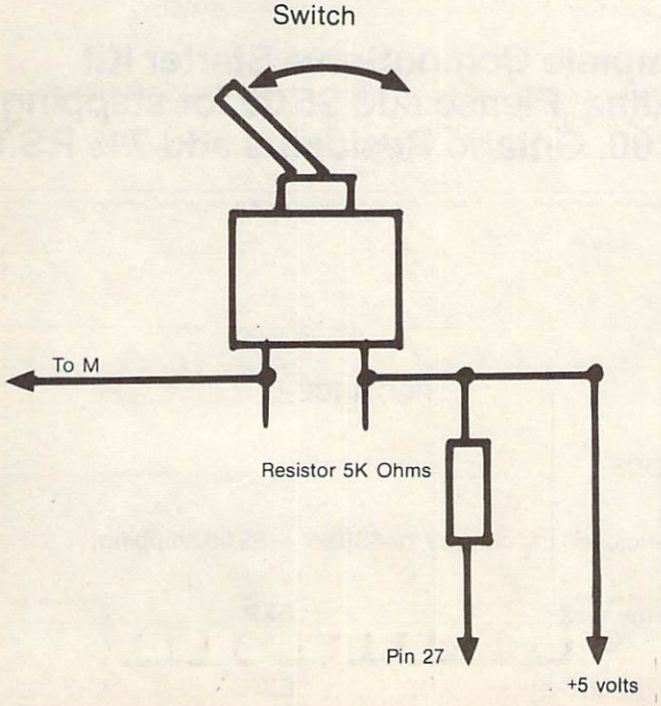
• Get a tiny on-off switch at Radio Shack, and don't tell them about the unknown state. Drill a hole on the side of the cartridge shell, somewhere between the first and the second bump of the shell, counting from the computer. Install the switch, making sure it can't twist and turn. Incidentally, you can use the extra washer as a stabilizer. Glued to the inside of the shell, it prevents the board from wobbling as it normally does.

• Attach the wire from connector M to one leg of the switch.

• Bundle together one end of a 5K resistor and a piece of thin wire about the same length as the first one. Solder the two to the other leg of the switch. Take the other end of the wire you just used, and solder it to pin 27 of the 6264 chip. You've now completed the read-write line circuit.

• Connect the other end of the resistor with a wire to any place that carries 5 volts — follow a path from the board's lines 14 and 15 (top). You have several wiring choices here: pins 28 or 26 on the chip, or any hole on the 5v line, will do.

• **Calc Result** goes back into place. You're all done. □



Sign on to CompuServe

For only $39.95* Cdn. get a complete CompuServe Starter Kit with detailed manuals and 1 hour of time. Please add $5.00 for shipping and handling. *U.S. send $28.95 + $5.00. Ontario Residents add 7% P.S.T.**

YES!, SEND ME THE KIT

SEND COUPON TO:
Thornhill Computers Inc.
7323 Yonge Street
(Just North of Steeles)
Thornhill, Ontario
Canada L3T 2B2
(416) 886-2494

NAME _____

ADDRESS _____

CITY _____ PROVINCE _____

POSTAL CODE _____

☐ I have enclosed my cheque for $39.95 + $5.00 shipping.

☐ Charge my Visa # _____ EXP. _____

☐ Charge my M/C # _____ EXP. _____

Toronto area members may shop in person at our store.

ED — The Amiga Screen Editor

by Roy Reddy

ED is the full-screen editor that comes with every Amiga as part of the Workbench diskette. This editor is not covered in the standard user documentation. Although there is no space in this issue to provide details, here is a quick reference to the **ED** commands. Complete documentation will be available in the *AmigaDOS User Manual*, which Commodore says will be available in early 1986 from your local Amiga dealer.

The template for the **ED** is:

ED "FROM/A, SIZE/K"

where the **FROM** argument is the file to be edited. **ED** allows for a file of 40,000 bytes as a default. If more memory is required, the **SIZE** switch may be used to allocate more memory.

ED has two categories of commands — 'immediate' and 'extended'. Immediate commands are a sequence of keystrokes performed while in the middle of a document. Some immediate commands are a combination of the **CTRL** key and another key. Extended commands are executed in a command mode. The 25th row of the screen is reserved for the extended command line and is entered by striking the **ESC** key. Some commands are available in both modes.

Immediate Commands

Cursor Movement

Cursor up	Move cursor up
Cursor down	Move cursor down
Cursor right	Move cursor right
Cursor left	Move cursor left
Backspace	Move cursor left and erases character
DEL	Delete character under cursor and moves remaining text left.
ESC	Enter the extended command mode at bottom of screen.
RETURN	Return cursor to left edge, breaking current line if in insert mode.
TAB	Move cursor to next TAB position right.

Commands performed in conjunction with the CTRL key (two keys held simultaneously).

CTRL-A	Insert a line after the current line.
CTRL-B	Delete the current line.
CTRL-D	Scrolls text down 12 lines (almost equivalent to cursor up).
CTRL-E	Move to opposite (top or bottom) corner of screen.
CTRL-F	Flip the case of the character under the cursor.
CTRL-G	Will repeat the last extended command.
CTRL-H	Equivalent to backspace.
CTRL-I	Equivalent to TAB
CTRL-M	Equivalent to RETURN
CTRL-O	Deletes spaces until next non-space character, or deletes character (like DEL) if not a space.
CTRL-R	Moves cursor to the space following the previous word.
CTRL-T	Moves cursor to the start of the next word (non-space).
CTRL-U	Almost equivalent to cursor down, scrolls text up 12 lines.
CTRL-V	Update or redraw the screen.
CTRL-[	Equivalent to ESC.
CTRL-]	Move cursor to opposite (left or right) end of current line.

Extended Commands

The extended mode is entered with **ESC** or **CTRL-[**. Some extended commands can contain additional parameters (string or numeric). Strings must be delimited by a character other than alphanumerics, brackets, a space or a semicolon. The slash character **/** is an acceptable delimiter, and is used in the examples in this article.

A /str/	Insert a line of text 'str' after current line.
B	Move cursor to the bottom of the file.
BE	Set the Block End at cursor position.
BF /str/	Search for 'str' backwards.
BS	Set the Block Start at cursor position.
CE	Move the cursor to the end of the line.
CL	Equivalent to cursor left in immediate mode.
CR	Equivalent to cursor right in immediate mode.
CS	Move the cursor to the start of the line.
D	Delete the current line.
DB	Delete the previously defined block.
DC	Equivalent to DEL in immediate mode.
E /str1/str2/	Exchange 'str2' for 'str1' throughout document.
EQ /str1/str2/	As above, but will query before exchange.
EX	Extend right margin ignoring defaults.
F /str/	Find 'str' in document.
I /str/	Insert a line of text 'str' before current line.
IB	Insert a copy of a previously defined block.
IF /fil/	Insert or merge the file 'fil' into the document at the current line.
J	Joins the current line with the next.
LC	Set ED to distinguish between upper and lower case.
M n	Moves the cursor to line n.
N	Moves the cursor to the start of the next line.
P	Moves the cursor to the start of the previous line.
Q	Quits ED without saving text
RP	Repeat the last extended command continuously.
S	Create a new line by splitting the current line.
SA	Save the document.
SB	Moves cursor to the start of the defined block.
SH	Show the status (margins, tab length, filename).
SL n	Set the left margin to the value specified in n.
SR n	Set the right margin to the value specified in n.
ST n	Set the distance between each TAB to the value specified in n.
T	Moves the cursor to the top of the document.
U	Undo the last change made (except delete line).
UC	Set ED to *not* distinguish between upper/lower case.
WB /fil/	Write previously defined block to file 'fil'.
X	Exit from ED and write document to file.

An AmigaDOS CLI Reference

by Roy Reddy

This article is meant as a quick reference card for those users without an AmigaDOS manual. At the time of writing Commodore states that they are supplying Amiga developers with the Amiga manual sets, and that they hope to have these manuals available to end users in early 1986. The manuals will be sold through the same retail outlets as the Amiga computers. Although it is not possible to cover the commands here in the same detail as the manuals do (they're not small), this information will at least get you started and give you a basis for further exploration.

The AmigaDOS environment is very helpful in that it has a sort of built-in 'help' feature. If you remember the command, but forget the pattern for the arguments, you can ask AmigaDOS to show you. The syntax for this kind of help is:

<Command> <space> ? <return>

AmigaDOS will respond by showing you the argument template. This template has three qualifiers, which are preceded by slashes as follows:
/A: this argument must be present and may *not* be omitted.
/K: the argument must use this keyword.
/S: this argument is used as a switch.
In future issues the command details and their applications could be further discussed, but I'll just give the commands themselves, along with an example or two for each one.

File Utilities

; *Begins a comment.*
Format: **[<command>] ; [<comment>]**
Template: **"command" ; "comment"**
Examples:
copy DF0:c/list to ram: ; Copy list program to RAM drive
list DF1: ; display directory of DF1: drive

< and **>** *Directs command input and output respectively.*
Format: **<command> [<][>] [<arg>]**
Template: **"command" > "TO" < "FROM" "args"**
Examples:
DATE > date-file
recv-file < send-file

COPY *Copies one file to another or copies all the files from one directory to another.*
Format: **COPY [FROM] <name>] [TO <name>] [ALL] [QUIET]**
Template: **COPY "FROM,TO/A,ALL/S,QUIET/S"**
Examples:
COPY DF0: TO DF1: ALL QUIET
COPY print-file TO PRT:

DELETE *Deletes up to 10 files or directories.*
Format: **DELETE <name> [<name>] [ALL] [Q|QUIET]**
Template: **DELETE ",,,,,,,,,ALL/S,Q=QUIET/S**
Examples:
DELETE out-dated-file
DELETE temps/file1 temps/file1

DIR *Shows filenames in a directory.*
Format: **DIR [<name>] [OPT A|I|AI]**
Template: **DIR "DIR,OPT/K"**
Examples:
DIR
DIR DF1: OPT A

ED *Enters a screen editor for text files.*
Format: **ED [FROM] <name> [SIZE <n>]**
Template: **ED "FROM/A,SIZE"**
Examples:
ED temp/ed-file
ED large-file SIZE 55000

EDIT *Enters a line editor.*
Format: **EDIT [FROM] <name> [[TO] <name>] [WITH <name>] [VER <name>] [OPT <option>]**
Template: **EDIT "FROM/A,TO,WITH/K,VER/K,OPT/K"**
Examples:
EDIT ed-file WITH edits VER nil:
EDIT orig-file TO new-file

FILENOTE *Attaches a note with a maximum of 80 characters to a specified file.*
Format: **FILENOTE [FILE] <file> COMMENT <string>**
Template: **FILENOTE "FILE/A,COMMENT/K"**
Examples:
FILENOTE my-picture COMMENT "drawn in November"
FILENOTE src-file COMMENT "source for screen"

JOIN *Concatenates up to 15 files to form a new file.*
Format: **JOIN <name> <name> [<name>] AS <name>**
Template: **JOIN ",,,,,,,,,,,,,AS/A/K"**
Examples:
JOIN src-file1 src-file2 AS all-src
JOIN text data results AS experiment

LIST *Examines and displays detailed information about a file or directory.*
Format: **LIST [DIR] <dir> [P|PAT <pat>] [KEYS] [DATES] [NODATES] [TO <name>] [S <string>] [SINCE <date>] [UPTO <date>] [QUICK]**
Template: **LIST "DIR,P=PAT/K,KEYS/S,DATES/S,NODATES/S,TO/K, S/K,SINCE/K,UPTO/K,QUICK/S"**
Examples:
LIST DF1:
LIST P ??
LIST :I S handler

MAKEDIR *Creates a directory with a specified name.*
Format: **MAKEDIR <dir>**
Template: **MAKEDIR "/A"**
Examples:
MAKEDIR DF0:test
MAKEDIR RAM:temp/files

PROTECT *Sets a file's protection status.*
Format: **PROTECT [FILE] <name> [FLAGS <status>]**
Template: **PROTECT "FILE,FLAGS/K"**
Examples:
PROTECT file1 rwd
PROTECT temp/file FLAGS r

RENAME *Renames a file or directory.*
Format: **RENAME [FROM] <name> [TO|AS] <name>**
Template: **RENAME "FROM/A,TO=AS/A"**
Examples:
RENAME OLD-NAME NEW-NAME
RENAME RAM:C/FILE1 NEW-FILE

SEARCH *Looks for a specified text string in all the files of a directory.*
Format: **SEARCH [FROM] <name>|<pattern> [SEARCH] <string> [ALL]**
Template: **SEARCH "FROM,SEARCH/A,ALL/S"**
Examples:
SEARCH DF0: my-name
SEARCH DF1: source ALL

SORT *Sorts simple files.*
Format: **SORT [FROM] <name> [[TO] <name>] [COLSTART <n>]**
Template: **SORT "FROM/A,TO/A,COLSTART/K"**
Examples:
SORT file TO sorted-file
SORT list TO sort-list COLSTART 5

TYPE *Types a file to the screen that you can optionally specify as text or hex.*
Format: **TYPE [FROM] <name> [[TO <name>] [OPT N|H]**
Template: **TYPE "FROM/A,TO,OPT/K"**
Examples:
TYPE preferences OPT H
TYPE :s/startup-sequence

CLI Control

BREAK *Sets attention flags in a given process.*
Format: **BREAK <task> [ALL] [C] [D] [E] [F]**
Template: **BREAK "TASK/A,ALL/S,C/S,D/S,E/S,F/S"**
Examples:
BREAK 6
BREAK 4 D F

CD *Sets a current directory and/or drive.*
Format: **CD [<dir>]**
Template : **CD "DIR"**
Examples:
CD RAM:
CD DF1:temp/l

ENDCLI *Ends an interactive CLI process.*
Format: **ENDCLI**
Template: **ENDCLI**
Examples:
ENDCLI

NEWCLI *Creates a new interactive CLI process.*
Format: **NEWCLI [<window>]**
Template: **NEWCLI "WINDOW"**
Examples:
NEWCLI
NEWCLI CON:30/30/300/120/"NEWEST CLI"

PROMPT *Changes the prompt in the current CLI.*
Format: **PROMPT <prompt>**
Template: **PROMPT "PROMPT"**
Examples:
PROMPT
PROMPT "%n>"

RUN *Executes commands as a background process.*
Format: **RUN <command>**
Template: **RUN command +**
command
Examples:
RUN TYPE :s/startup-sequence
RUN COPY :c/list TO RAM: +
CD RAM: +
LIST

STACK *Displays or sets the stack size for commands.*
Format: **STACK [<n>]**
Template: **STACK "SIZE"**
Examples:
STACK
STACK 9000

STATUS *Displays information about the CLI processes currently in existence.*
Format: **STATUS [<process>] [FULL] [TCB] [SEGS] [CLI|ALL]**
Template: **STATUS "PROCESS, FULL/S, TCB/S, SEGS/S, CLI=ALL/S"**
Examples:
STATUS
STATUS 1 FULL

WHY *Explains why a previous command failed.*
Format: **WHY**
Template: **WHY**
Examples:
WHY

Command Sequence Control

ECHO *Displays the message specified in a command argument.*
Format: **ECHO <string>**
Template: **ECHO ""**
Examples:
ECHO "This string was echoed to the screen"

EXECUTE *Executes a file of commands.*
Format: **EXECUTE <commandfile> [arguments]**
Template: **EXECUTE "command-file","args"**
Examples:
EXECUTE :s/startup-sequence
EXECUTE my-command-file

FAILAT *Fails a command sequence if a program returns an error code greater than or equal to this number.*
Format: **FAILAT <n>**
Template: **FAILAT "rclim"**
Examples:
FAILAT
FAILAT 14

IF *Tests specified actions within a command sequence.*
Format: **IF [NOT] [WARN] [ERROR] [FAIL] [<string> EQ <string>] [EXISTS <name>]**
Template: **IF "NOT/S,WARN/S,ERROR/S,FAIL/S,,EQ/K,EXISTS/K"**
Examples:
IF EXISTS :c/cd
IF ERROR

LAB *Defines a label (see SKIP).*
Format: **LAB <string>**
Template: **LAB <text>**
Examples:
LAB error-location
LAB ok

QUIT *Exits from command sequence with a given error code.*
Format: **QUIT [<returncode>]**
Template: **QUIT "RC"**
Examples:
QUIT 14
QUIT

SKIP *Jumps forward to a LAB in a command sequence (see LAB).*
Format: **SKIP <label>**
Template: **SKIP "LABEL"**
Examples:
SKIP
SKIP error-location

WAIT *Waits for, or until, a specified time.*
Format: **WAIT <n> [SEC|SECS] [MIN|MINS] [UNTIL<time>]**
Template: **WAIT ",SEC=SECS/S,MIN=MINS/S,UNTIL/K"**
Examples:
WAIT
WAIT UNTIL 12:25

System and Storage Management

ASSIGN *Assigns a logical device name to a filing system directory.*
Format: **ASSIGN [[<name>] <dir>] [LIST]**
Template: **ASSIGN "NAME,DIR,LIST/S"**
Examples:
ASSIGN temp: DF0:source/files
ASSIGN LIST

DATE *Displays or sets the system date and time.*
Format: **DATE [<date>] [<time>] [TO|VER <name>]**
Template: **DATE "DATE,TIME,TO=VER/K"**
Examples:
DATE today
DATE 01-OCT-85 12:32

DISKCOPY *Copies the contents of one entire floppy disk to another.*
Format: **DISKCOPY [FROM] <disk> TO <disk> [NAME <name>]**
Template: **DISKCOPY "FROM/A,TO/A/K,NAME/K"**
Examples:
DISKCOPY FROM DF0: TO DF1:
DISKCOPY DF0: TO DF0: NAME "Copied Disk"

FAULT *Displays messages corresponding to supplied fault or error codes.*
Format: **FAULT [<n>]**
Template: **FAULT ",,,,,,,,,"**
Examples:
FAULT 123
FAULT 133 234 245

FORMAT *Formats and initializes a new 3 1/2 inch floppy disk.*
Format: **FORMAT DRIVE <drivename> NAME <string>**
Template: **FORMAT "DRIVE/A/K,NAME/A/K"**
Examples:
FORMAT DRIVE DF0: NAME "New Blank Disk"

INFO *Gives information about the filing system.*
Format: **INFO**
Template: **INFO**
Examples:
INFO

INSTALL *Makes a formatted disk bootable.*
Format: **INSTALL [DRIVE] <drive>**
Template: **INSTALL "DRIVE/A"**
Examples:
INSTALL DF0:

RELABEL *Changes the volume name of a disk.*
Format: **RELABEL [DRIVE] <drive> [NAME] <name>**
Template: **RELABEL "DRIVE/A,NAME/A"**
Examples:
RELABEL DF1: "Disk over there"

EDIT — The Amiga Line Editor

by Roy Reddy

EDIT is the line-oriented editor that comes with every Amiga as part of the Workbench diskette. This editor is not covered in the standard user documentation. Although there is no space in this issue to provide details, here is a quick reference to the **EDIT** commands. Complete documentation will be available in the *AmigaDOS User Manual*, which Commodore says will be available in early 1986 from your local Amiga dealer.

Commodore full screen editors have spoiled me, making it hard to get interested in studying a line editor. However, **EDIT** offers quite a bit of power in that it can modify files using commands from another file called an 'Edit Command File'. **EDIT**'s format and template are:

Format: **EDIT [FROM]<name> [[TO]<name>] [WITH<name>] [VER<name>] [OPT Pn:Wn:PnWn]**

Template: **EDIT "FROM/A,TO,WITH/K,VER/K,OPT/K"**

EDIT has some limitations, the first of which, you will discover, is that you cannot create a new file with it. For this reason the **FROM** argument must be present, even though the keyword **FROM** is theoretically optional.

The argument **TO** instructs **EDIT** what filename to give the destination file. If the **TO** argument is omitted, **EDIT** creates a file and will rename it with the **FROM** name when you quit **EDIT**. The original file is saved as **':T/EDIT-BACKUP** and will remain until the next **EDIT** session.

The **WITH** file is the 'Edit Command File' which gives **EDIT** its additional editing power. The **VER** file, if specified, will contain error messages that may have been generated during the **EDIT** session. If either the **WITH** or **VER** arguments is omitted, **EDIT** will use the keyboard and screen for input and output respectively.

EDIT has memory and line width defaults that can be adjusted using the **OPT** keyword and either/both the **P** and **W** arguments. With the **P** argument, you can adjust the amount of memory **EDIT** uses to retain previous lines. The **W** argument adjusts the maximum line length used by **EDIT**. The default setting of these parameters is **P40W120** — 40 previous lines retained, and 120 maximum line length.

The following abbreviations are used the command descriptions below:

/qs/	Qualified string
/t/	String
n	Line number, or. . .
<f>	File specifier
sw	+ or — (on or off)

Character Pointer Commands

<	Move character pointer left.
>	Move character pointer right.
#	Delete character at pointer.
$	Lower case character at pointer.
%	Upper case character at pointer.
<SPACE>	Turn character at pointer to space.
PA /qs/	Move character pointer to after /qs/.
PB /qs/	Move character pointer to before /qs/.
PR	Reset character pointer to start of line.

Current Line Positioning Commands

M n	Move to line n.
M +	Move to highest line in memory.
M -	Move to lowest line in memory.
N	Move to next line in memory.
P	Move to previous line in memory.
REWIND	Make current line line #1 of source file.

File Search Commands

F /qs/	Search for string /qs/.
BF /qs/	Search backward for string /qs/.
DF /qs/	Search and delete line with string /qs/.

Text Display and Verification Commands

?	Verify current line.
!	Verify with case indicators.
T	Type lines until end of file.
T n	Type n lines forward.
TL n	Type n lines with line numbers.
TP	Moves to lowest line then type lines.
V sw	Switch to turn on/off line display verification.

Commands That Operate On The Current Line

A /qs/t/	Insert string /t/ after string /qs/ on current line.
AP /qs/t/	Same as above but moves character pointer.
B /qs/t/	Insert string /t/ before string /qs/ on current line.
BP /qs/t/	Same as above but moves character pointer.
CL t	Concatenate current line + string /t/ + next line.
D	Delete current line.
DFA /qs/	Delete after string /qs/ to the end of the line.
DFB /qs/	Delete before string /qs/ to the end of the line.
DTA /qs/	Delete from start of line to after string /qs/.
DTB /qs/	Delete from start of line to before string /qs/.
E /qs/t/	Exchange string /qs/ for string /t/.
EP /qs/t/	Same as above but moves character pointer.
I	Insert characters from keyboard before characters in line.
I <f>	Insert characters from file **<f>** before characters in line.
R	Replace characters from keyboard.
R <f>	Replace characters from file **<f>**.
SA /qs/	Split current line After string /qs/.
SB /qs/	Split current line Before string /qs/.

Commands That Operate Globally

GA /qs/t/	Globally place string /t/ after string /qs/.
GAB /qs/t/	Globally place string /t/ before string /qs/.
GE /qs/t/	Globally exchange string /t/ for string /qs/.
CG [n]	Cancel global operation **[n]** (all operations if **[n]** omitted).
DG [n]	Disable global operation **[n]** (all operations if **[n]** omitted).
EG [n]	Enable global operation **[n]** (all operations if **[n]** omitted).
SHG	Show information about globals in use.

Input/Output Redirection

FROM	Read from source (original file).
FROM <f>	Read from file **<f>**.
TO	Return to original destination file.
TO <f>	Send output to file **<f>**.
CF <f>	Close file **<f>**.

Miscellaneous Commands

'	Repeat previous A, B, or E command.
= n	Set line number to n.
C <f>	Take commands from file **<f>**.
H n	Set halt flag at line n (if n=* then Halt and unset H).
Q	Quit input from command file, or windup if no command file.
SHD	Show data; last cmd, search string.
STOP	Stop; quit without changes made to existing file.
TR sw	Switch to suppress trailing spaces from lines.
W	Windup; continue through the remaining source file.
Z /t/	Change value of current input terminator string to /t/.

STOLEN PROPERTY!

DO YOU COPY © SOFTWARE?

THE SYMBOL © OR THE WORD "COPYRIGHT" MEANS THAT A PROGRAM OR WORK IS PROTECTED BY INTERNATIONAL COPYRIGHT LAWS!

"COPYRIGHT" OR © MEANS THAT SOMEONE HAS INVESTED TIME AND MONEY IN THE DEVELOPMENT OF THAT PROGRAM. QUITE SIMPLY, THAT PROGRAM IS SOMEONE ELSE'S PROPERTY.

IF YOU HAVE A "COPY" OF A © PROGRAM WHICH YOU DID NOT BUY, THEN YOU HAVE STOLEN PROPERTY!

DON'T STEAL SOFTWARE!
BUY IT AND SUPPORT THE INDUSTRY YOU'VE INVESTED IN!

BATTERIES  INCLUDED

"The Energized Software Company!"

30 Mural Street
Richmond Hill, Ontario
L4B 1B5 Canada
(416) 881-9941
Telex 06-21-8290

17875 Sky Park North
Suite P, Irvine, California
USA 92714
(416) 881-9816
Telex 509139

printed as a service to the software retailers of North America

Library Additions

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.

PET Disk (P)TN

Presented by Mike Donegan

The bulk of this month's disk is given over to **HotCan**, a program written by R.S Dumont, M.E. Lux and H.W. Orr of the National Research Council office in Saskatoon, Saskatchewan. File input modifications for PET/CBM equipment were performed by A.E. Krause of the Electrical Engineering Department of the University of Saskatchewan. **HotCan** is designed as an aid in estimating the space heating requirements of residences. Manuals are available for about 25 dollars (Cdn.) prepaid from: Publications Section of Buildings Research, National Research Council of Canada, Ottawa, Ontario, Canada K1A 0R6. Use of the manuals is recommended for proper operation of the program. The program is also available for non-Commodore computers — inquire at the address above.

The menu program in this package is called **Hello**. It is responsible for invoking the various modules, which are **Change Costs**, **Add Data**, **List Data**, **Instructions**, **HotCan**, **Exec Sub** and **Output Sub**. Data files include **String Save**, **Energy Costs**, and twelve files containing climatic data for various cities: **Vancouver**, **Edmonton**, **Suffield**, **Swift Current**, **Saskatoon**, **Winnipeg**, **Toronto**, **Ottawa**, **Montreal**, **Fredericton**, **Halifax** and **St. John's**. There are also four alternate files of city data that you can set up yourself with the aid of the manual (**First Alternate** through **Fourth Alternate**), and two example files from the manual, **Inp.dat** and **Manual.dat**. The first of these two does not contain comments; it is the default data file if no other is specified. **Manual.dat** is the commented version.

Another type of data file is specific to the house under investigation. Files of this type can also be set up by the user. The package includes a set of data files created using actual measurement from the house of A.E. Krause in Saskatoon. The names of these files are **Krause.fi.dat**, **Krause.sf.dat**, **Krause.dat** and **Krause.ni.dat**. Files for your own house can be generated with an editor. Two are included on the disk: **32K-Editor4.p** (for PET/CBM) and **Editor64.c** (for C-64).

When you run **HotCan** for the first time, use the data file input method and the default file, pressing **RETURN** when the dot ('.') appears on the screen. This will you show you what type of data is required for input. Good luck.

In addition to **HotCan**, this disk features a package of educational freeware for teachers, by R. Dray of Peterborough, Ontario. Information on the package is contained in the data file **Read-me**, which you can examine with the accompanying utility, **Read-me Reader**. The heart of the package is the **Mark Master 2.3c** program for recording student marks.

Miscellaneous programs on this month's disk are **Calculator.z** — a utility to add a calculator function to your programs; and three science tutorials — **Resistors.z**, **Motion.z** and **Electricity.z**.

```
0 pet disk (p)tn
29    "list-me ptn.l"      prg
9     "hello"              prg
5     "change costs"       prg
15    "add data"           prg
20    "list data"          prg
8     "instructions"       prg
89    "hotcan"             prg
13    "exec sub"           prg
47    "output sub"         prg
1     "citys"              seq
1     "string save"        seq
1     "energy costs"       seq
2     "vancouver"          seq
2     "edmonton"           seq
2     "suffield"           seq
2     "swift current"      seq
2     "saskatoon"          seq
2     "winnipeg"           seq
2     "toronto"            seq
2     "ottawa"             seq
2     "montreal"           seq
2     "fredericton"        seq
2     "halifax"            seq
2     "st. john's"         seq
2     "first alternate"    seq
2     "second alternate"   seq
2     "third alternate"    seq
2     "fourth alternate"   seq
2     "inp.dat"            seq
13    "krause.fi.dat"      seq
8     "manual.dat"         seq
13    "krause.sf.dat"      seq
13    "krause.dat"         seq
12    "krause.ni.dat"      seq
1     "copyhotcan"         prg
7     "32k-editor4.p"      prg
7     "editor64.c"         prg
57    "mark master2 3.p"   prg
1     "read-me reader.z"   prg
23    "sample marks"       seq
3     "read-me"            seq
6     "calculator.z"       prg
59    "resistors.z"        prg
33    "motion.z"           prg
51    "electricity.z"      prg
```

C-64 Disk (C)TN

by Derick Campbell

This month's Commodore 64 disk contains a useful graphics utility, **VMSup2.Rel.c**), and a program (**Bit SuperOSE.c**) that shows off its abilities by drawing patterns derived from your input.

We also have a Wild West railroad game, **Iron Horse.c**; and a computer version of a classic card game — **Cribbage**.

Those of you who are musically inclined will enjoy this month's jukebox of songs. If you also have a technical bent, you won't want to miss the tutorial that explains how to set up and use a digital/analog converter, with the programs **C64T-4VM.C** and **C64D-4VM.C** (tape and disk).

Also included on this disk is a set of disk utilities that will help you out with many 1541 operations, and the latest update of Jim Butterfield's **Supermon** machine language monitor.

```
0 c-64 disk (c)tn
9     "list-me (c)tn.l"    prg
73    "iron horse.c"       prg
26    "sid's jukebox.c"    prg
84    "sid.songs.d"        prg
7     "sid.vsmon.d"        prg
1     "sid.george.d"       prg
15    "super64run.c"       prg
8     "super64ins.c"       prg
30    "cribbage.c"         prg
12    "bit window.rel.c"   prg
37    "disk utilities.c"   prg
47    "tutor-music.c"      prg
15    "c64t-4vm.c"         prg
15    "c64d-4vm.c"         prg
13    "arioso in g.d"      prg
59    "ram-bow.c"          prg
34    "bal. checkbook.c"   prg
```

Continued overleaf...

Software order form

NAME _____

STREET ADDRESS _____

CITY/TOWN/P.O. _____

PROV/STATE _____ POSTAL/ZIP CODE _____

TELEPHONE _____ MEMBERSHIP NO. _____

TORONTO PET USERS GROUP, 101 Duncan Mill Road, Suite G7, Don Mills, Ontario M3B 1Z3 416-445-4524

disks

To order club disks by mail, send $10.00 for each 4040/2031/1540/1541 disk (4040 format), discount price 5-10 $9.00 each, 11 or more $8.00 each; and $12.00 for each 8050/8250 disk (8050 format). We do honour purchase orders from school boards.

These disks are for use with a _____ computer and a _____ disk drive.

Please send me the following:

3 Letter/No. Code	Description	4040 or 8050 Format	Price

Total $ _____ .00

tapes

To order VIC 20 or Commodore 64 library tapes, send $6.00 for each tape.
To order PET/CBM or Commodore Educational Software, send $10.00 for each tape.

These tapes are for use with a _____ computer and a datasette.

If for a PET computer, what model - _____ - BASIC - 1.0( ); 2.0( ); 4.0( )?

3 Letter/No. Code	Description	Price

Total $ _____ .00

The prices indicated include postage and handling as well as Ontario Provincial Sales Tax (if applicable)

☐ Cheque/money order enclosed (payable to TPUG)

☐Visa/Mastercard # _____

Signature _____

Expiry date _____

Library Additions

```
10    "basic fun.c"        prg
75    "address book 2.c"   prg
23    "adbk delete.c"      prg
7     "vmtest plots.c"     prg
5     "bit superose.c"     prg
4     "-vmsupp.6000.d"     prg
22    "vmsup2.rel.c"       prg
```

```
0 c-64 music disk (c)sh
5     "list-me (c)sh.l"    prg
20    "kplay"              prg
35    "sidnews#1"          prg
38    "sidcat50"           prg
4     "albumleaf.mus"      prg
3     "etal.mus"           prg
2     "osprey.mus"         prg
8     "axelf-1.mus"        prg
3     "feast.mus"          prg
6     "pastoral.mus"       prg
5     "axelf-r.mus"        prg
6     "fsonatina.mus"      prg
4     "peanuts.mus"        prg
2     "bistro.mus"         prg
5     "gigue.mus"          prg
2     "pipers.mus"         prg
29    "brand1-3.mus"       prg
3     "gsonatina.mus"      prg
3     "presto.mus"         prg
2     "byebye.mus"         prg
8     "happysong.mus"      prg
8     "scarlatti.mus"      prg
2     "calliope.mus"       prg
1     "harmonica.mus"      prg
4     "scipio.mus"         prg
5     "canon.mus"          prg
12    "hindemith.mus"      prg
8     "sheworks.mus"       prg
16    "cantina.mus"        prg
2     "holst.mus"          prg
7     "splash.mus"         prg
7     "commodore.mus"      prg
11    "homecoming.mus"     prg
4     "t&v.mus"            prg
3     "courante.mus"       prg
6     "glad.mus"           prg
7     "tpi#14.mus"         prg
2     "crab.mus"           prg
10    "gothos.mus"         prg
18    "vconcerto.mus"      prg
3     "dref.mus"           prg
2     "k.c.o..mus"         prg
3     "venite.mus"         prg
12    "duckie.mus"         prg
12    "liberty.mus"        prg
12    "victors.mus"        prg
16    "duwahrer.mus"       prg
17    "little.mus"         prg
6     "wsoldier.mus"       prg
1     "eggs.mus"           prg
6     "longest.mus"        prg
1     "yoy.mus"            prg
4     "enola.mus"          prg
7     "mule.mus"           prg
4     "zorro.mus"          prg
8     "songcopy"           prg
14    "printer"            prg
83    "promo"              prg
```

```
0 "mem.test+os9.rel" o2 2a
462   "OS9 DRIVE A"        rel
23    "test.os9.tests"     seq
2     "test.os9.mmu"       seq
8     "print.error.asm"    seq
3     "print.error.b09"    seq
1     "test.main.cmd"      seq
11    "test.main.asm"      seq
6     "test.main.b09"      seq
4     "test.main"          prg
1     "test.banks.cmd"     seq
22    "test.banks.asm"     seq
9     "test.banks.b09"     seq
5     "test.banks"         prg
1     "test.os9.cmd"       seq
6     "test.os9m.asm"      seq
9     "test.os9b.asm"      seq
5     "test.os9m.b09"      seq
1     "test.os9b.lst"      seq
16    "test.os9b.b09"      seq
4     "3.board.chip.map"   seq
14    "README"             seq
1     "test.os9.map"       seq
10    "test.os9"           prg
1     "test.os9.exp"       seq
```

Unclassified

NEC Spinwriter 3530 Printer: 35 cps Letter Quality, with Batteries Included ADA 1600 interface. Connects to CBM 8032 or PET. $800 or best offer. Also: 4040 Disk Drive, $600 or best offer. (613)-345-5521 any time, ask for John Childs. All equipment excellent condition.

For Sale: Commodore 4032 32K with 2031 disk drive. Includes TNW-2000 serial-parallel interface, 300 baud modem, toolkit and macroassembler. Also assorted equipment and software. Nice condition. Near Detroit, MI. $650.00 US. (313) 461-1787

WANTED: 8032, 8050 or 8250. Call Maria (416) 364-9020.

For Sale:: CBM 8032 with cassette drive. Also included: ROM Rabbit, PaperClip (on disk and tape) plus hundreds of other programs. $600.00. Call Ken (519) 759-7397

For Sale:: CBM SuperPET with 4040 disk drive, MX-80 printer, interface and lots of software. Asking $1395.00 or best offer. Call 844-4060 or 844-6001

For Sale:: 4040 dual disk drive, IEEE cable, C-Link interface. $500.00 or b/o. (604) 365-8192

For Sale:: SuperPET, 4040 drive, Visicalc, Wordpro 4+, KRAM Database and lots of other software. $2100.00 or best offer. (416) 446-1035

For Sale:: B-128 computer, 8050 disk drive, 4023 printer w/spare cartridge, Zenith monitor. Complete with all cables, instruction manuals, programmer's reference guide and Superscript II. Only used about ten hours by a beginning programming student. $750.00. Call (406) 822-4989 after 5pm. MST.

For Sale: Commodore 8032 computer, 8050 Drives, MX-80F/T Printer, WP4+, Basic Compiler, Manager and Kram. $1700.00/Best Offer. Call Mike (416) 431-1597.

For Sale: 4016 PET (upgraded to 32K) and 2031 disk. Has 9-inch screen. All for $750.00. Write: Guy Leger, 23 Des Oblats, Lasalle PQ, Canada H8R 3K9.

Reviews

Five Commodore 64 Word Processors

Review by Michael Quigley

The following five word processors are all for the Commodore 64. Although none of them is good enough to take the place of **PaperClip**, one of the most popular word processors ever created — or even Cardco's **Write Now!**, which I use extensively — some of them have interesting capabilities, as well as a few oddities.

All of them are disk-based, and all but one are in machine language. **TexEd** is the only one not protected, while **Textomat 64** has some fancy DOS routines that make it impossible to use with the Epyx Fastload cartridge. The others contain the now-archaic, 1541-destroying errors that cause a knocking exactly like disk-formatting sounds.

Textomat 64

Textomat 64 comes from Abacus Software, the same people who brought us all those nifty books like *The Anatomy of the 1541 Disk Drive*, wretchedly translated from the German. **Textomat** seems to have come from the same factory, but its 105-page, three-ring binder manual is logically organized and almost completely free of spelling, grammatical and translation problems.

Unfortunately, the manual's ease of use is not duplicated in the program. While the overlying design concept is a good one, there are a lot of annoying little idiosyncrasies. For one thing, you have to hit a shifted **RETURN** at the end of a paragraph. And the delete key doesn't work normally — you have to put it on the character you want to delete, instead of deleting the character *before* the cursor. The manual says this is "more convenient, since most of the time the cursor is positioned at beginning (sic) of the unwanted text so you don't have to move to the end to delete it." Uh, sure...

The program performs most of its functions with menus and prompts, many of which appear at the bottom of the screen. These are accessed with the cursor keys and the **f1** key, which is used along with the **f2** key to jump back and forth to the main text area ('Write Mode'). The **CTRL** key is used to enter the 'Command Mode', in which one can edit and move through the document. One problem with many of these menus is that you really have to think in advance about what you are doing. For example, when you want to search and replace, you cannot see your text; and when you want to rename a file, you cannot see the directory.

One part of this program I really disliked was customizing **Textomat** for my particular printer. (The program is written for use with a 1525, 1526 or MPS 801, for which no such modification is necessary.) This involves two things: converting the *entire alphabet* (upper and lower case) to the corresponding ASCII codes of the printer, and defining control characters to produce things like subscripts, superscripts and bold type — in hexadecimal! I tried to do this, with little success on printout. Considering **PaperClip** includes setup files for just about every printer on the market, I don't see why **Textomat** couldn't at least include some for the popular makes of printers. (Two of the other programs reviewed here — **The Whole Bit** and **Word Commander 64** — do just that).

Textomat does have some interesting features, like the ability to make calculations in the text and print foreign language characters. However, in light of the printer hassles I encountered, I find it very hard to recommend.

***Textomat 64** ($39.95), from Abacus Software, PO Box 7211, Grand Rapids, Michigan 49510.*

The Whole Bit

The Whole Bit is not all that bad, though it's not without its share of little peculiarities.

Editing is literally of the full-screen variety — if you want to go beyond the boundaries of the screen, you have to use the function keys to scroll. The function keys have different... er... functions, depending on whether you are in 40-column or 80-column mode (the latter scrolling text from left to right). Pushing **RETURN** while typing in text makes a double space, while the 'up arrow' is used for a single return at the end of a paragraph.

The speed of the cursor has been increased, which makes manipulating it easy when editing text. However, in other parts of the program where you are supposed to place the cursor on something (like a file name, to load it in the directory), this can be a disadvantage — especially when coupled with the fact that the cursor only moves down in this kind of situation.

One of the major characteristics of **The Whole Bit** that is either annoying or reassuring depending on your viewpoint is that most of the prompts are double-checked — the *Are you sure?* syndrome, which is no guarantee that mistakes will not be made. One nice touch is that a shifted **RUN/STOP** will get you out of virtually any situation, and back to the main menu.

DOS commands are limited to 'new', 'scratch', 'resave', 'rename', 'save' and 'load'. Each file is saved along with various parameters, such as margins, tab settings, lines per page and justification, as a 'master file'. In order to merge one file into another, you have to save the parts to be merged as a 'non-master file' — that is, without the auxiliary information. You can also copy or chain together up to four files, and save them under a new file name.

Eight printer functions are supported — bold face, double strike, italics, compressed, underlining, wide type, superscripts and subscripts; and there are another six user-definable keys, which have to be entered as hex numbers.

The Whole Bit is generally well-designed and relatively easy to use. (One major oversight is that if you hit **RUN/STOP** and **RESTORE**, it's bye-bye to everything.) Aside from a few grammatical atrocities, the manual for the program is very good.

***The Whole Bit** ($39.95), Applied Technologies, Inc., Computer Products Div., Kittery, Me. 03904.*

Bank Street Writer

Bank Street Writer is a limited word processor that seems to have been designed with novices in mind. Text is created in the 'Write Mode'. If you want to edit it, you have to enter the 'Edit Mode', which allows you to move the cursor around. Then you have to return to the 'Write Mode' to actually make the corrections. The 'Transfer Mode' allows you to do various DOS operations, such as 'new', 'rename', 'scratch', 'load' and 'save'. File names can be a maximum of eight characters long, since the remaining let-

ters are for an optional 'password', to prevent others from accessing the material. (I'm sure that examining the material would be a relatively easy task for someone who really wanted to see it).

Compared to many other word processors, **Bank Street Writer** is strictly 'meat and potatoes'. Block manipulation is limited to fifteen lines of text at a time, and no printer tricks (like italics, condensed print, and so forth) are supported. There isn't even a character to indicate where paragraphs end!

Although prior to printing you can establish such parameters as the number of characters per line, pagination, spaces between lines, page heading, pause between pages, and ejecting the last page, there are other things that can only be changed in the 'Utility Program'. These include lines per page, the top and bottom margin, printer device number and secondary address, screen colours, line and form feeds, and an audible clicking when keys are pressed. You cannot make use of the 'Utility Program' when **Bank Street Writer** is running — instead, you have to access it while the main program is loading by pushing the 'left arrow' key.

Personally, I don't like **Bank Street Writer**; and especially not the business of jumping back and forth between the 'Write' and 'Edit' modes constantly. I can see, however, where it would be well-suited to educational applications. The program is extensively menu-driven and crash-proof; it comes with a lengthy tutorial on the disk, instructing the user in the fundamentals of word processing (according to **Bank Street Writer**); and the manual for the program is excellent.

***Bank Street Writer** ($49.95), Broderbund Software, 17 Paul Drive, San Rafael, CA 94903. The package contains two copies of the program disk.*

Word Commander 64

One of the first things I try to do with a new word processor is to 'make it screw up'. One of the nice things about **Word Commander 64** is that it makes it easy to access many printer features, like underlining and boldface, by using the Commodore key with a single letter — for example, 'U' for Underline — before and after the section you want to change. So I threw in everything but the kitchen sink, and **Word Commander** performed without a moment's hesitation — even when mixing double-wide and normal print in justified columns!

While there are many aspects of **Word Commander 64** like this one that I like (including the ability to move the cursor all over the place), there are quite a few that are disappointing — starting with the knock-knock noise while the program loads.

While there is full-screen editing, the cursor is in the 'eternal insert' mode, and there is no way to correct words by typing over them. The cursor colour is the same as the border and, while you are allowed a wide range of choices, I found it difficult to get satisfactory colour combinations.

DOS commands (aside from 'save' and 'load') are limited to 'new', 'scratch' and 'directory'; and if you save a program under a file name that already exists, the previous one is replaced without warning, which may not be what you want. File names are limited to ten characters. You can't merge a file on disk with one in memory — instead you have to chain them together while printing.

I wrote to the manufacturer about these problems, and they sent me back a reply (at least give them points for customer support) saying that most of the complaints were 'design features'. To this, I can only say 'too bad', since in **Word Commander 64** there are the makings of a really first-class word processor.

***Word Commander 64** ($49.95), NMG Micro Software, P.O. Box 131, Marlboro, NJ 07746.*

TexED

In its ads, **TexED** is described as a "powerful text editor for document processing and program design" that allows you to "create, modify, and save cassette data files and disk SEQuential files", as well as "convert program files to and from SEQuential files". Among its features are a "line image editor using simple commands", "full screen editor (uses cursor control keys)" and a "print command with indentation and margins (which) supports COMMODORE printers".

Sounds pretty good, eh? Well, I got the shock of my life when I received **TexED** for review. The program is written in BASIC! This means that it's slower than molasses. I tried compiling it, which improved matters slightly, but in doing so, I found an error — an **ON...GOTO** a line that didn't exist!

To make things worse, the documentation for the program is unusually bad, written in a style reminiscent of bibliographical footnotes, and full of peculiar phraseology the program's author seems to have invented to make life more complicated. Consider the following: "A file is a bounded text which is written on tape or disk"; and "By default, TexED uses the line-image editing mode. Line-image commands are instructions to TexEd which direct the program to carry out some operation, such as LOADing some file into a buffer."

Terrible! Terrible! And then there are commands like ".+n addresses the nth line after the current line", ".-n addresses the nth line before the current line" and "$-n addresses the nth line before the last line". As Charlie Brown would say: "Good Grief!"

I can't see much use for **TexEd**, except for someone who wants to study how a word processor works prior to writing their own. At least it comes on a high-quality Maxell disk.

***TexEd** ($19.95), APCAD, PO Box 83, Saline MI 48176. VIC 20 or C-64 versions on disk or tape.* □

The PX-80 printer
from TEO
Epson-compatible dot matrix printer

Review by Doug Chisholm

Last spring, just before school ended, I purchased a Commodore 64. Not long after that, I discovered a sophisticated word processor called **PaperClip** (from Batteries Included), which introduced me to the joys of word processing. I quickly became an addict, and my neighbours soon tired of my frequent use of their printers. It soon became apparent I needed a printer of my own: school would be starting soon, and I couldn't bear the thought of going back to rough copies and pens.

I looked at various styles of printer, and opted for dot matrix, so I could do graphics. I also needed one with high-quality print, to use when writing essays. In short, I wanted the best of both worlds — letter quality printing, and graphics too. I'd heard some horror stories about the Commodore 1526, and it seemed to lack the advanced features of other printers (such as double strike, subscripts, superscripts and underlining capabilities). I looked at the Epson line of printers, which were very nice, but very expensive. Then, much to my great delight, I discovered the TEO PX-80!

The TEO PX-80 is an Epson clone that has all the features of the Epson MX-80

to 'Epson' mode, and I use the MX80 printer file with **Paperclip** without any problems. I purchased my PX-80 for 500 dollars with the interface — a Cardco ?/+G (see *TPUG Magazine*, January 1985) from Comspec, in Toronto.

The PX-80 is a square-pin dot matrix printer — the square pins provide superior print formation, compared to the traditional round pins. The pins are not vertically aligned, but rather offset, to allow rounder characters. One can't see spaces between the dots, even in the normal mode! The letters look good without double strike, and even on an old ribbon.

The PX-80 prints bi-directionally and is logic-seeking. This speeds things up by allowing it to print from both right-to-left, and left-to-right, as well as skipping any unnecessary spaces. The speed is rated at 80 CPS (characters per second), but the interface slows things down considerably.

Subscripts and superscripts are supported, as are italics, underlining, emphasized type, and programmable character spacing. You can select these features from software, or by setting DIP switches in the back of the printer. You can choose between 40, 80, 71 or 142 characters per line, and you can even choose between a slashed or unslashed zero.

The printer accepts both tractor or friction feed paper. This allows you to use personal letterhead, or forms without the little perforations on the side, if you wish, as well as as the more typical perforated tractor-feed computer paper. You can adjust the printer to accept paper of any width from 4 to 10 inches.

As I have already mentioned, the printer will work nicely with the Card ?/+G. In the emulation mode, the TEO PX-80 emulates the 1525 exactly. I have run several programs designed for use with the 1525, without having to make modifications because of my printer. I have also successfully used **Doodle!**, **PaperClip** and **The Print Shop**.

This printer has very few faults. However, the hard plastic dust covers seem to break easily (mine did, and so did those of two friends). Another weak point is the manual. It was obviously translated from the Japanese very poorly (but it is good for a laugh!) Here's how they describe removing the shipping screws from the bottom of the printer: "Carefully lift from of printer unit and make it stands as the bottom of case be vartically face to you and hold the untit by the one of you hand on the soft surface... After it is removed, gently back the unit to lay flat on a firm surface, position the printer front be face to you". Needless to say, the manual did make setting up and using the advanced functions of the printer a little difficult.

However, I would rate this as extremely good value for the money. The bottom line: I have yet to see a dot matrix printer that I would rather have. □

Dr. Seuss Fix-up The Mix-up Puzzler

from CBS Software

Pre-school level game for the C-64

Review by Gerry Gold

This one came to me through Jeff, age four. Anything that can keep him busy while *I'm* busy is worth looking at again. His initial reaction upon encountering it by accident at the TPUG office was total fascination so, with the hope that it would hold the children for hours, I took it home to review. I had to re-align my 1541 disk drive first, so that it would load, but at last **Dr. Seuss** was ready to be user-tested.

Two-year-old Karen, brother Jeff and sister Laura crowded around the screen. In a few minutes, they had the game worked out.

Dr. Seuss presents three of his renowned animals — the Cat in the Hat, the Grinch and Sneetch. When you press the fire button, the screen scrambles the heads, torsos and legs of the three cartoon characters so that none of them have the right parts. Your job is to put them back together again, using the joystick. A blank 'puzzle piece' to the right gives you a place to store things, while you move the puzzle pieces around.

You remember those dime puzzles? The squares slide upward and sideways until you make the right picture? That's basically what we have here. Do it right, however, and a part of the cartoon figure will wiggle back at you in approval. With five levels of complexity, it is supposed to appeal to a wide range of ages, from four to whatever.

For my kids, 'whatever' is about age seven. Only the 2-year-old did not get bored quickly. But that's not too bad, because games for 2-year-olds are hard to find.

What seems to be true about this type of game is that kids write their own rules. Little kids do not want simply to reassemble the Grinch and friends. The real fun is in creating new animals — like a Grinch with the Cat's belly and Sneetch feet. They did not even miss the encouraging wiggle-'rewards' for doing things right.

The verdict? Definitely for 2 to 6 years in age. Some parental encouragement may be initially required, because the joystick does not always slide the panels as fast as the kids would like to see them move. Attention span is about twenty minutes, and replay is assured for at least a week or two. □

Microshare Multi User System from COMSPEC

for C-64 and Commodore PET/CBM

Review by Ronald Fredericks

The Microshare Multi User System allows up to sixteen PET/CBM or eight model 64 computers to access up to two disk drives and five printers. The unit is well suited for educational classroom settings, or other environments where several computers desire to share disk drives and/or printers. The PET/CBM model comes in 4, 8, 12 or 16 channel units. The C-64 version comes in 8 channel. Each unit comes with one 5, 8, 12 and 16 foot cable per four channel unit. The C-64 version has both IEEE and C-64 serial ports. There is also a built-in 16K print buffer in the C-64 version.

The basic Microshare unit for the PET/CBM comes in a four or eight channel version. There is a Microshare expansion unit available to add an additional four or eight channels. Basic units can easily be expanded by adding the expansion unit and connecting it to the main Microshare unit with the cable supplied. The remaining installation is very simple, and anyone with a basic knowledge of computer connections can easily install an entire system with the aid of the very well-written instruction manual. The unit comes with all the necessary cables for a typical installation, except the PET-IEEE and IEEE-IEEE cables required to hook up the peripheral devices. For installations requiring longer than normal cable runs, additional length cables and extension cables are available at an additional charge.

Testing of the Microshare

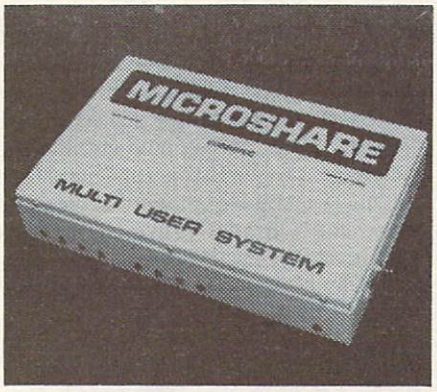
The Microshare is actually an IEEE switch. It polls all computers connected to it, several times each second, to see if a computer is requesting use of the disk or printer. If a request is received by Microshare, it locks out all other computers requesting use of devices until the present computer is finished. Multiple requests are stacked in the order received, and each computer in turn is connected to the disk or printer when the present operations are completed.

The easiest way to test the Microshare is to put a disk in the disk drive, go to each computer, and press **SHIFT** and **RUN/STOP** as quickly as possible. Each computer should load the program in order, as requested. The LED light on the front should come on as each computer is given access to the disk or printer. All computers should load the program with no 'crashes'. Any problems can usually be traced to faulty connections, since the equipment has been thoroughly tested by the manufacturer.

Optional settings

There are two dip switches on the bottom of the Microshare unit. These are preset at the factory, but can be changed. Switch #1 is a timed wait setting. The Microshare will switch to the next unit in line when the present computer stops communicating with the device for a period of time. This time period is two seconds, with switch #1 set in the 'off' position. If the switch is set to the 'on' position, the wait time is reduced to only one second.

The Microshare will also switch to the next computer in line if the data transfer



stops in the middle of an i/o operation, or if the device itself holds up the operations. If a user 'crashes' his computer, or is accidentally powered off while communicating, the unit will switch to the next computer in line after about sixteen seconds. If switch #1 is set to the 'on' position, the time is reduced to eight seconds.

Switch #2 is a channel-switching mode selection. If, for some reason, a user stops communicating with a device, the Microshare will release to the next computer in line (time specified by switch #1 setting). This has the advantage of keeping a 'hung' program or an inactive user from tying up the system. The disadvantage is that a user could have open files closed, if there is too much delay between operations.

If switch #2 is set to the 'on' position, the Microshare will *not* switch to the next channel waiting in line when a user is in the **CMD** mode, or has not closed the files. This allows any amount of time to pass between operations. However, a single user could tie up the entire system until the files are closed. Programs containing file commands (**OPEN**, **INPUT#**, **PRINT#1**, **CLOSE**, and so on) should always be well written, to prevent accidental loss of data. Files should be opened, the data transferred, and the files closed as quickly as possible. Good programming techniques make this essential, and even more so in a multi-user environment. Microshare is totally transparent to the user, and does not keep track of what files are open, or what data is being sent. It acts merely as an IEEE switch that connects users, in turn, to the disk or printer.

First-hand experience

In the computer lab at our high school, we have twenty-one PET/CBM 8032 computers with two 8050 disk drives, one 9090 hard disk drive, and three printers. We are running these in two network configurations.

We *were* using a combination of a software and hardware network. The software part had to be booted into each computer in order to be operational. The access to the devices was controlled by means of a small control board plugged into the user port on the rear of the computer. The boards were then chained together with three wires connected from computer to computer. IEEE-IEEE cables were used to chain the computers to the disk drives and the printers.

This system had several disadvantages, but was very good for teaching BASIC programming, because of all the built-in features, such as password protection of a student's files. However, the system would not provide bus protection unless all computers were loaded with the software program. It was not possible to run combinations of different types of software at the same time. The worst feature of this system was that it often locked up the entire system because of hardware problems. I point this out only to compare it to Microshare.

After reading about Microshare, units were ordered and installed. Everything performed as designed. We can now run anything we want on any computer without crashing our system. Turning a computer on and off does not affect the system, even while another computer is using the disk or printer. We can still use the good features of our old software network for programming class management with the Microshare. The system provides our lab with greater flexibility in what we can do in our classes.

I think the most positive feature is the

trouble-free environment it provides for both the students and the teacher.

Conclusion

Based on my experience as a BASIC programming teacher, and having to keep twenty-one computers on line all day, I can strongly recommend the Microshare multi-user system and the 64K print buffer. I found the product easy to install, and it has been one hundred per cent reliable. The version for the Commodore 64 ought to be just as reliable.

Our school has a lot of money tied up in our lab, and we expect to use it for several more years to teach BASIC programming, word processing, spreadsheets and database programs, as well as how to operate other types of software. The Microshare will enable us to make the most efficient use of the disk drives and printers at a minimum cost. We also installed two COMSPEC 64K print buffers with our system, which provides a minimum of on-line time for printing operations.

Information about Microshare can be obtained from the manufacturer, COMSPEC Communications, Inc., 153 Bridgeland Avenue, Unit 5, Toronto, Ontario M6A 2Y6 (416) 787-0617. The cost of an eight channel unit for PET/CBM is 1,195 dollars, and a 16 channel system is 2,095 dollars. The 64K print buffer cost is 499 dollars each. The Microshare for the C-64 costs 995 dollars for an 8 channel system. All prices are in Canadian dollars. The price is reasonable, considering that this is a product that performs so well. □

COMSPEC
MCS 6400
from COMSPEC
64K IEEE print buffer
for PET/CBM computers

Review by Ronald Fredericks

The COMSPEC MCS 6400 is a transparent, easy-to-use 64K print buffer for Commodore PET/CBM computers and printers. The buffer may be used with Commodore IEEE printers, or with other brands of printers — it has an optional Centronics parallel port. The buffer may be used with or without COMSPEC's Microshare multi-user networking system.

about 2 x 10 x 6 inches. The unit has an 'on/off' switch, a **PAUSE** button, and a **RESET** button. It comes with its own plug-in power supply. Connections are made to the IEEE printer by means of a PET-IEEE cable (not supplied). Connection to Centronics-type printers is made with a Centronics-type cable. The MCS 6400 is designed and manufactured by COMSPEC Communications, Inc., 153 Bridgeland Ave., Unit 5, Toronto, Ontario M6A 2Y6.

Advantages of a print buffer

With a print buffer, an entire print file can be transferred to the buffer in a very short period of time — the computer can send the data much faster than the printer can print it. After the data has been sent to the buffer, it sends the data to the printer. Once the data has been transferred to the buffer, the computer is free to use the disk drive or do other operations. Thirty minutes of printing can be dumped into the buffer in only ninety seconds. A typical three-page program listing can be dumped to the buffer in about four seconds. In applications such as word processing and computer labs in schools, the buffer provides more efficient use of both equipment and the individual's time. This is especially true in a multi user environment. Other users hardly notice any delays in gaining access to the disk drive while the printer is running. The price of the MCS 6400 is 499 dollars (Cdn.).

Installation

The installation of the MCS 6400 is very simple. The buffer has one IEEE input connection port on the back of the unit. There are two output connection ports. One is the IEEE, and the other is a Centronics parallel port. A regular IEEE cable is used to connect the buffer to the printer. If the connection is made from the computer, it is done with an IEEE input and an IEEE cable for the output to the printer. The buffer may also be used with the Microshare multi-user system.

After connecting the cables to the units, the computers and the printer are powered up. The buffer power supply is plugged in, and the switch is turned on. The LED light flashes once and remains on, if the unit passes all internal diagnostic tests. The buffer is now ready for operation.

Switches

There are two push-button-type switches on the front of the buffer unit. The button on the right side is the **PAUSE**. Printing can be halted by pushing the button once. Pushing the button a second time will cause printing to resume. The button on the left side is for clearing the buffer memory, if desired. There are also four dip switches on the bottom of the unit, which provide settings for Commodore printers and Centronics printers; convert Commodore ASCII to true ASCII; and set the unit device number.

Applications

I installed two buffers in our high school computer lab. We have two separate networks using COMSPEC's Microshare multi-user system. Prior to the installation of the buffers, students were always wanting to use the disk drive, but the network was often tied up because of students listing their programs.

Sometimes, the network was tied up as much as one half the class period. During the time the printer was in operation, students could not use the disk drive. As a result, they got very impatient while waiting. It seemed like fifty times each day that I heard: "Who is printing?" or "Hurry up!". With the installation of the buffers, life has become much easier for both students and teacher.

Now it is just like having the printer not connected to the system. There are no obvious delays while the computer sends data to the buffer. Students can print and do disk operations at the same time. It take no longer to send a file to the buffer than it does to load it from the disk. The buffer provides greater productivity in the lab.

Conclusion

If one does a lot of printing, such as in a computer lab setting or a lot of word processing applications, the MCS 6400 print buffer is a necessary piece of equipment. The unit is designed very well, operates beautifully, and is completely transparent to the user. I would highly recommend the MCS 6400 print buffer. □

Hitchhiker
from Infocom
Text adventure game
for the Commodore 64

Review by Shafqat Khan

March 3, 1978, saw the introduction of the most popular radio series of all time over BBC's airwaves: Douglas Adams' *Hitchhiker's Guide To The Galaxy*. For those of you who are not familiar with the

series, *Hitchhiker* is a bizarre story about an average Englishman named Arthur Dent who traverses the universe discovering strange people, places and things. The strange people, places and things also discover Arthur Dent.

After Adams made piles of money converting *Hitchhiker* into more radio programs, four books (which he calls 'the Hitchhiker trilogy'), and a television series, he converted it into an interactive computer adventure game, as well. He approached Infocom and was immediately teamed up with Steven Meretzky (author of **Planetfall** and **Sorcerer**). This duo's sense of humour meshed so well that, in many instances, the computer game is even more fun than the book.

Although Infocom rates the game as standard level, it should be tackled as an advanced game — it is definitely not suitable for beginners. If you cannot get through the first few moves, remember to read and examine *everything* carefully. The first chapter of the book is helpful, but after that, the book and game branch out in different directions (you really do not have to buy the book to solve the game, although it gives you a good frame of reference for dealing with certain actions in it).

And now, for all you prospective **Hitchhiker** players — some hints:

- Do not think logically.
- The game does lie to you at certain points.
- You must have the babel fish to solve certain parts of the game.
- Read all descriptions carefully.
- Consult the guide when you need help.
- Enjoy the Vogon poetry.
- Don't panic!

The adventure hints that a second part might be released in the future. Considering the number of *Hitchhiker* books, this series is likely to extend further than the **Zork** series did. I will gladly answer any questions about the game, if frustrated players wish to write to me c/o *TPUG Magazine*. □

Quink
from CBS Software
Trivia quiz game
for Commodore 64

Review by Ian A. Wright

Quink is a *Trivial Pursuit*-type game that fills eight boxes with facts. Some are 'matches' and others are 'loners', but all the facts are based on information in one subject area. The computer generates a three-by-three grid, and you are given a fixed time limit in which you must select the item(s) that do not match the others — *and* press the correct key on the C-64 keyboard.

Quink can be played as a two-player (or two-team) game; or one-on-one against the computer. You can choose from six categories: fame, pop culture, imagination, science and nature, general knowledge and mixed bag. Other than a bias toward American content, the information bank of 4500 items is fine. There are five levels of difficulty, and you can choose to play each of these at a more advanced level, where you are responsible for indicating a completed screen — plus bonus rounds. There are enough built-in options to make **Quink** last for a long time.

Quink is a fun game, but it's not for kids. A friend's two sons (eight and ten) would watch but not play **Quink** — it was too difficult for them. The time limit is not adjustable down to their playing speed, and they couldn't work the keyboard accurately enough.

Neither can I! I would like **Quink** to have a keyboard overlay, because using the T, Y, U, G, H, J, B, N and M keys is awkward. I still find myself pressing the wrong key, even when I know the correct answer.

Being locked to the keyboard is unavoidable — but it is one of the great failings of all computer-based trivia games. I believe adults would enjoy sprawling around the four sides of a game-board more than huddling around a computer keyboard and monitor. For the 'closet trivia freak', however, **Quink** is just the ticket. There is no sound to disturb the rest of the family at 2:00 am, the game is fast, the factual information is extensive and informative — and no one will know how many mistakes you made.

Quink comes in an attractive library case with a sixteen-page game manual and a disk. This very nice packaging is unfortunately spoilt by having a disk read error, which bangs the drive head on my 1541 (not very healthy for the faithful beast). I wonder if having a keyboard overlay would not have been more effective than copy protecting the disk?

***Quink**, from CBS Software, CBS Publishing Group, 1 Fawcett Place, Greenwich, CT 06836. In Canada: Holt, Rinehart and Winston.* □

Forecast!
from CBS Software
Weather forecasting
program for the C-64

Review by Dave Neale

How many times have you made some important plans, only to have them ruined because of poor weather? Now that can all change with a new program from CBS Software, called **Forecast!**. You'll be able to turn your C-64 into a weather station that can log, detail and forecast the weather on a daily basis.

With five different subprograms to work with, you will be able to predict the weather for this afternoon, evening or even for tomorrow, using the Weather Forecaster section. A 'Weather Calculator' can make it easier to understand the types of measurement used in forecasting. While doing all this, you can store each day's weather in the 'Weather Keeper' section, for an accurate record of the previous weather. The 'Weather Traveller' section allows you the opportunity to see the weather for almost anywhere in the USA. The last section, called 'Weather Tracker', lets you follow the patterns of various hurricanes.

The basic program is quite good: however, there are a few limitations that should be mentioned — the most important for Canadian readers being that it is an American publication, and so Canada doesn't play a role in any of the stored infomation or weather maps.

The graphics are pretty good: however, the biggest complaint I had about it was the seemingly unreasonable time it took to load various subsections. I know the 1541 is a slow drive, but **Forecast!** exaggerates this drawback.

Included with the program comes an excellent manual to help the budding forecaster become acquainted with weather jargon. There are all sorts of well-detailed drawings and photographs to help explain weather patterns and concepts. Also, the manual includes a list of American and Canadian colleges and universites where Atmospheric Sciences are studied, as well as a glossary and a bibliography.

Despite some failings, **Forecast!** is a good program for those who are interested in the weather, and want to know if that picnic they're planning will be rained on or not. □

Jance Hard Wire Microcomputer Security System

from Jance Associates

Software, cartridge and hardware package for C-64 and VIC 20

Review by Robert J. Sodaro

Jance Associates Inc., PO Box 234, East Texas, PA 18046, (215) 398-0434.

Jance Associates has developed a cartridge interface that will allow you to use your computer as a burglar alarm system. The program is available on both disk and tape, and will work with either the VIC 20 or C-64. The hardware for the security system retails for $195.00 (US), thus placing it well below comparable home security systems.

The hardware that comes with the Jance system is the VIC Rel Cartridge (its computer interface), 13 magnetic switches (to mount on doors and windows), 2 deactivate buttons, 1 panic button (for remote activation of alarms), 2 alarm bells (one for inside and outside your house), one 12 volt DC power supply for bells, 5 window warning decals, and 200 feet of 22 gauge, stranded, twin-conductor wire.

To many, wiring the house with a burglar alarm system might seem an impossible challenge. Not so with the Jance system. This system is so simple that even an individual who knows very little about electricity (like this reviewer) should be able to do it unaided. Wiring up the Jance System is so simple, it is child's play, even for someone who has trouble remembering which way a lightbulb is screwed in.

With the aid of very explicit schematics, diagrams and instructions, even a novice is capable of ensuring the security of his or her home. Step by step, you are guided through each phase of the installation process. First, you must determine a central location for the computer. It must be placed where it will neither be in the way, nor difficult to access. (Unless you have a second unit, you must be able to get to your computer for normal use when the alarm system is not operating.)

The Jance Hardwire Alarm System requires that wires be run from each door or window to the computer. Jance also offers a 'wireless' system, which uses your home's own electrical system. With the wireless system, leads run from windows and doors directly to the nearest electrical outlet. Once the computer is plugged in and turned on, the circuit is completed, and the system is operating.

When running wires around your house, care must be taken to tuck loose wires under edging, or run through ducts, so they will not become a hazard. This information is pointed out by the system's instructions, which seem to anticipate nearly every possible problem. In addition to the printed instructions, there is also a program, complete with troubleshooting hints contained on the disk or tape itself. The user is able to list and/or 'back-up' the security program, as there are no copy protection measures taken (at least there were none on the copy tested), possibly so the user will be able to customize the program. Detailed program documentation is also included.

The system allows for various user options. Windows can be set up in either parallel or serial wiring, so the windows can be left part-way open. This allows for ventilation without having to worry about setting off the alarm.

When you are ready to leave your house, you push one of the deactivate buttons, disabling the system for five minutes. You then exit the house, and close the front door. Re-entry triggers a 10-second warning beeper that sounds prior to the unleashing of both inner and outer alarms. This allows you time to reach the deactivate button and kill the alarm. The function keys on the VIC 20 and C-64 are used for activating, deactivating and testing the system.

In addition to a simple aural alarm, multiple functions are available as add-ons to the Jance system. These include the ability to install more than the minimum number of switches to the system (allowing for the possibility of virtually all windows and doors to be protected). Another option is a program that will automatically dial a phone number, ring an alarm over the phone, hang up, and dial another number. Up to 100 numbers can be called in this fashion. Motion and vibration detectors can be added to the system, as well as programs that will turn lights on and off, and raise and lower room and water temperatures. Many other functions are also available. None of these features were included with the model tested, and will obviously add to the $195.00 price tag.

There are one or two drawbacks to utilizing this alarm system in conjunction with your computer. First, while the alarm is in use, the computer cannot be used for any other function. Also, while it is in use, your transformer is continually on. The people at Jance have informed us that this will cause no harm to the transformer (it will get quite warm, though). Still, leaving it on all the time just might cause the transformer to burn up faster than it normally would. In either case, the total life span of the transformer is well into the thousands of hours, and it really should not make much of a difference in the long run.

In the final analysis, the Jance Security System gets a very high rating, and those C-64 and VIC 20 owners who are interested in having an alarm system installed should check into it. Not only is this system priced well below comparable home security systems (under 500 dollars, counting the cost of a VIC 20, datasette and extra magnetic switches), but you still have use of a very versatile home computer system, as well. □

Cassette Book for the Commodore 64 (and VIC 20)

by D. Paulissen

from Abacus Software

212 pages, soft cover

Review by Anne E. Gudz

Abacus Software, Inc., PO Box 7211, Grand Rapids, MI 49510. ISBN 0-916439-04-6.

This book is a 'must' for those of us still using a tape system, if only for the directions on how to hook a loudspeaker to the datasette, so one may position the tape at the beginning of the program; or, more importantly, to aid in aligning the sound head, so no more **?LOAD ERROR** messages are received.

The author also explains how to save programs after **?LOAD ERROR** does occur.

The intricacies of tape use are described in some detail, as well as how the datasette actually works. The author explains program handling in both direct mode and program mode. Self-starting programs and program protection are covered, along with storage of machine language programs. A section on adapting other cassette recorders for data storage is also included, together with the program **FastTape**, which the author

claims is ten times faster in loading and saving programs. (Data can be loaded and stored up to twenty times faster than with the normal operating system).

Other useful programs included are catalogues of the cassette contents, in regular save or **FastTape** save modes. □

Okimate 10 Colour Printer
from Okidata
Colour printer for the Commodore 64

by Malcolm O'Brien

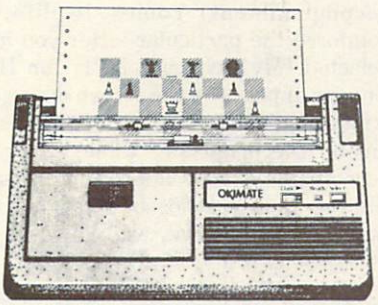
The businessman with bar charts of financial data... The video artist with stacks of disks, full of **Doodle** pictures... The child who enjoys colouring in pictures with a KoalaPad... What do they all have in common? Colour graphics! They all need a way to get the pictures off the screen and onto paper. Well, the Okimate 10 can do it!

Rarely has so much excitement been generated by the appearance of a printer. I'll bet you were as intrigued as I was when we first started seeing the ads for the Okimate 10. Gorgeous colour pictures, a (direct connect) serial bus interface and an attractive price. It was almost too good to be true! Make no mistake — it is true, and it's everything that you thought it would be.

The first thing that struck me about the printer when I opened the package was its small size. It's a little smaller than a 3-ring binder (although it's heavier than it looks), so transporting it is a snap. Don't let its size fool you into thinking that it's a toy, however. This is really a first-class piece of machinery — it just has more brains than mass. What I mean to say is that the Okimate 10 is very well-designed, and the more I used it, the more I marvelled at it.

It's a thermal transfer printer. This means that the print head's heat causes the colour on the ribbon to melt off onto the paper. Two ribbons are supplied, by the way — one black, one colour. You also get a few sheets of special paper. The waxy colour from the ribbon will adhere more evenly to this very smooth paper than it will to regular paper. The more expensive grades of paper tend to have a rough surface, and this will tend to result in spotty printing. The special paper is available through Okidata (an order form is included), but you can also use Savin photocopier paper and get good results. Acetate sheets will produce excellent copy, too — a must for presentations using overhead projectors. Perhaps you remember the 'World of TPUG' graphic by Terry Coviello on the cover of our May issue. That was done on acetate (it is backed with white paper in the photograph, of course), and the result is strikingly vivid and clear.

The Okimate 10 offers both tractor feed and friction feed, and both work very well. I detected no slippage in the friction feed, and I was particularly impressed with the tractor feed. I use 'sheer cut' paper (the kind that leaves a smooth edge when you tear the pin-holes off), and I have often had the problem of the pinhole strips tearing off and winding back into the tractor mechanism. The Okimate handled my finicky paper perfectly, thus relieving me of a major annoyance.



There's another option open to you that's worth mentioning too. If your application doesn't require colour, you can print without a ribbon, if you use thermal paper. There are different grades of thermal paper, though. Get the stuff that's made for the IBM PCjr printer. This is just the ticket for those 'colourless' applications — word processing, spreadsheets, program listings, and so on. While I'm on the subject of spreadsheets, I should also say that the Okimate 10's bag of tricks includes condensed print — something that's always been absent in Commodore printers. It's often a very handy feature!

Thermal printers generally offer quiet operation, and the Okimate 10 is no exception. You will not have to suffer the slings and arrows of outraged neighbors or family members. Printing with the Okimate 10 is about as noisy as ironing.

Installation is a piece of cake. Slide in the Plug 'n' Print module and snap the cover over it. Don't worry, it's goof-proof. Even if you have eleven thumbs and don't read the instructions. But read the instructions. They're not only instructive, they're very attractively printed and well laid out. Installing the serial cable and printer ribbon is just as simple. In about two minutes, you're all ready to go!

Once you're all hooked up, load the introductory software and follow the instructions on the screen. Soon you'll have some text samples and beautiful colour pictures! In fact, the Okimate can print more colours than the C-64 can display! If you've ever made a hi-res picture and printed it, only to find your picture looked squashed on paper, you'll certainly appreciate the way the Okimate 10 does justice to your works of art. The software will faithfully reproduce pictures made with the KoalaPad, **Doodle**, and a number of other popular packages.

As if all this wasn't enough, it's 1525-compatible. And forget about the dip switches because there aren't any. Just print, and it works! What a novel idea! Why can't more printers be this friendly?

It's a shame to have to talk about drawbacks when writing about such a fine piece of hardware. And really, it isn't so bad when weighed against its other attributes but... it takes two feet of ribbon to print one *line* of colour. That's a lot of ribbon. It means that you'll get about eight to ten colour pictures per ribbon cartridge. Printing in colour is definitely more expensive than printing in black! Still, to the three types of user described at the start of this review, the price differential will be of little consequence.

All in all, if you're in the market for a printer, don't pass this one by. You may find that it suits your needs admirably. If you *need* to print colour graphics, then you need the Okimate 10! □

Snoopy To The Rescue
from Random House
Educational game for Commodore 64
$39.95

Review by Jim Grubbs

It's arcade time! **Snoopy to the Rescue** is appropriate for ages 8 to adult. The premise of the game is that Snoopy's best buddy Woodstock has been bird-napped by an evil cat named Morehairy! Woodstock is being held captive in a castle, and Snoopy must make it through the six levels of the castle to rescue his buddy.

The opening graphics provide a Commodore 64 Snoopy lover's delight. Perched atop his doghouse, Snoopy starts typing: *"It was a dark and stormy night..."*. As he types, the letters appear in the balloon on the screen. Then there is a sequence of pictures that shows Snoopy receiving a ransom note and, finally, a picture of the castle where Woodstock is being held captive.

The game is not overly complicated. As you manoeuvre Snoopy through the levels, you must stop at each of five blocks and collect a number. In order to successfully leave the level, the total of the five numbers you have collected must match Morehairy's secret number that appears at the top of the screen. So, even while seemingly doing nothing but having fun, mathematical skills are being reinforced.

There are some red robots to watch out for, and Snoopy can slip off the rocks, but I didn't find the program to be overly picky.

I would rate the difficulty of play appropriate for the lower end of the age scale and — for people like myself who really aren't very good at arcade games! — **Snoopy to the Rescue** offers an attainable goal. It is on a two-sided disk, with the second side offering more challenging play.

One nice feature is the the ability to practise any level you wish — unlike many games, which offer only one way to get to each level (through the preceding one!). The 'sound off' option is included, as is the scoreboard. All in all, it is a very enjoyable game.

Although the graphics are excellent, and the Snoopy character charming, this is not the program for you, if you're mainly interested in seeing the Peanuts characters come to life. If that's what appeals to you, one of the other programs in the series will do a better job. (There are six altogether). □

Charlie
Brown's ABC's
from Random House
Educational game
for the Commodore 64
$39.95

Review by Jim Grubbs

As an adult, it's difficult to think of computer software for three-year-olds. **Charlie Brown's ABC's** is aimed at just such an audience, with a suggested upper age limit of seven. The concept is a simple one, and not vastly different from other such learning packages.

When you load the program the first time, a 'bookplate' appears on the screen, and you are prompted to enter your name as owner of the disk. It's a very nice touch, and really makes the child feel as if the program belongs to him or her. Instructions for **Charlie Brown's ABC's** are simple, but a 28-page small colouring book-style manual is included.

There are two ways to put the program into action. You can either press any letter key, or use the cursor key to view the letters in alphabetical order. Either way, you are presented with a screen containing the letter in both upper and lower case, and a picture of an object that begins with that letter.

While viewing the screen, you must press the letter again. When you do, an animated scene (with musical accompaniment) comes to life, to reinforce the particular letter you have selected. My favourite is 'H for Hat'. Snoopy appears on the screen dressed in a magician's hat and cape. He places his hat on the magician's table, all to the accompaniment of appropriate music. Snoopy then reaches into the hat and pulls out Woodstock, who is wearing a bunny suit.

The animation is not quite cartoon quality, but still amazing, coming out of a Commmodore 64.

The program is very attractively packaged in a reusable box, which contains not only the instructions and a double-sided disk (the alphabet is split into two parts), but also some activity cards that can be used for additional learning games.

The only problem I found with the program was that, each time the child presses a letter and a screen changes, a 'zipping' sound occurs. It seems that this is intended as a hook, to keep the child's attention. As such, it seemed to work well — but it seemed to have a very irritating effect on almost all adults who were exposed to it for more than a few minutes!

Charlie Brown's ABC's not only teaches the alphabet, but familiarizes youngsters with the keys on the computer. Chances are that older children and adults will develop a strong desire to relearn the alphabet, too, when this program comes into the house!

***Charlie Brown's ABC's**, from Random House Electronic Publishing, 201 East 50th Street, New York, NY 10022.* □

Peanuts
Picture Puzzlers
from Random House
Educational game
for Commodore 64
$39.95

Review by Jim Grubbs

Recommended for ages 4 to 8, **Peanuts Picture Puzzlers** is somewhat more challenging than the first in this series, **Charlie Brown's ABC's**. That, of course, is as it should be. The same clear, concise comic book-type instructions are included, though they have been reduced to a mere twelve pages.

Sixty-one different picture combinations are possible. You select a character for each side of the puzzle, from a cast of thirteen characters. Normally, a puzzle containing eight pieces is created, but you can select either four or sixteen, to adjust the level of play to the child involved. Play can either be against the clock with options of 30 seconds, 2 minutes or 5 minutes, a limit of your choice, or no time limit at all.

You begin by viewing a picture. When you are ready to start, you press **RETURN**. A single piece of the puzzle appears on the right-hand side of the screen. An outline of the puzzle appears on the left. By either cycling through the picture pieces, or by moving the cursor to the position that matches the piece being displayed, you start solving the puzzle. When you are correct, the piece pops into place. If you are wrong, a tone is sounded.

When the puzzle is solved, you are rewarded by seeing the whole picture again, along with some animation. The graphics are superb, and the animation very cute. Music accompanies the picture.

There is even a scoreboard for the Top Ten Players, just like in an arcade game. It can be erased by performing a simple procedure. Somewhere between **Charlie Brown's ABC's** and **Picture Puzzlers**, the programmers decided that being able to turn off the sound might just be a good feature, and that option is included in this program.

Intended to develop eye/hand coordination, **Peanuts Picture Puzzlers** does an excellent job.

***Peanuts Picture Puzzlers**, from Random House Electronic Publishing, 201 East 50th Street, New York, NY 10022.* □

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 14 lesson course of over 230 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.

Fill in the coupon or send a facsimile.

TP

NAME:_____

ADDRESS:_____

CITY:_____

PROV./STATE:_____

POSTAL/ZIP CODE:_____

Check desired course:

COMMODORE 64 ☐
VIC ☐ PET ☐
COMMODORE 16 ☐

Send Cheque or Money Order to:
Brantford Educational Services
6 Pioneer Place,
Brantford, Ontario,
Canada N3R 7G7

Complete course:	$19.95
Postage and hand.:	$3.00
Total:	$22.95

NEW

PRO-TERM 64

INTELLIGENT TERMINAL PACKAGE

Now with VT100 emulations
A full screen editor in the buffer
40/80 columns in software
X modem protocol
Steve Punter's new C1 protocol
40/80 column display
Informative status line
Extensive Auto-Dial/Auto-logon commands
Upload/download:
(Punter) (X/ON X/OFF) (.IMG FILES)
Complete Disk support, DOS commands, dual drive, or two single drives
Complete Printer support
Transparent stream to Printer
Programmable Function Keys
Supports VT/52 and Televideo 910/920 terminal emulations

On disk ONLY $46.95

KING MICROWARE Ltd.

Suite 210
5950 Côtes des neiges
Montreal, Quebec H3S 1Z6
Canada

C.O.D., money order or cheque

Dealer and distributor inquiries invited.

Write for our FREE catalog.
Send us your programs. We publish software.

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.

CADPAK-64

CADPAK from Abacus Software, PO Box 7211, Grand Rapids, MI 45910. Price: $39.95 (US).

CADPAK-64 is a computer-aided design package for the C-64 and C-128 computers. It enables the user to draw pictures and graphic designs with a resolution of 320 points wide by 200 points high. Once completed, an exactly-scaled printer output can be obtained in either small or large size, on virtually any dot matrix printer (Epson, Okidata, Prowriter, Okimate-10 colour, et cetera).

There are two versions of **CADPAK-64** in one package. In the light pen version, selection of functions, options and the actual drawing is done using a light pen. This method should be chosen when a lot of freehand drawing will be done, and/or widely spaced points on the screen are going to be utilized. The keyboard version uses the cursor control in the drawing process. In both versions, all the functions of the program are selected from the master menu.

The screen can be split, creating twin 'screens' to work with. Areas of one screen can be copied to the other. Rotation in 90 degree steps, upsizing and reflection of the image are allowed, during copying. Text in four sizes may also be placed anywhere on the picture. **CADPAK-64** offers also an object management system. The system enables the user to store on disk up to 104 different objects, sized 16 by 16 pixels, which the user can recall for use later.

Still Sizzling...

Superscript for the C-64

PractiCorp International, the company which introduced **Practicalc 64** and **Practicalc II** spreadsheet programs, are distributing two new Precision Software products for C-64 computers. **SuperScript** is an enhanced and remodelled version of Precision's top-selling **EasyScript** word processor. **SuperScript** is compatible with existing **EasyScript** and **EasySpell** files, and integrates with **SuperBase**, as well as with PractiCorp's **Practicalc II**. The second product, **SuperType**, is a professional keyboard trainer. **SuperScript** is available for $49.95 (US), and **SuperType** at $29.95 (US).

For more information, contact PractiCorp, The Silk Mill, 44 Oak Street, Newton Upper Falls, MA 02164, (617) 965-9870.

For Ham Operators

QSKY Publishing announces the release of *Command Post reprint booklet*, by Jim Grubbs, who is also the author of *The Commodore Ham's Companion*, another QSKY publication.

Command Post brings all of the material from the popular *Command Post* column (*Commander Magazine*, September 1983 to June 1984) together in one convenient reference source. It includes actual program listings for morse code send and receive, radioteletype send and receive, duplicate checking and program exchange over the air. More than just a group of programs, *Command Post* is a tutorial in basic interfacing techniques that can be applied to amateur radio applications, as well as other control situations.

The booklet is available for $9.95 (US). In the United States and Canada, there is a $2.50 shipping and handling charge.

Grade Manager III

Grade Manager III (for the C-64) is a comprehensive grade calculation, management and reporting system for teachers in all areas of education, from elementary to university.

The system maintains a detailed assignment history for each student. Each history supports up to 95 assignments, with grades covering as much as six terms per school year, plus two special grades (such as 'mid-year' and 'final exams'). It also records times absent, times tardy, and bonus points for each student. Each file (class or subject) may have up to 100 students, with an unlimited number of files. Up to 30 files (or 600 student records) may be merged for the Report Card Summary. The program sorts students alphabetically by any field, or in any other order specified. Eighteen printer reports allow access to information in over fifty different ways. Most reports include bar charts. Four reports may also be reviewed on the video display.

The package contains two system diskettes (work copy and backup), user's manual, and quick reference card. Free GM3 User Group Membership is also offered. **Grade Manager III** is available for $69.95 (US) from Smoky Mountain Software, PO Box 1710, Brevard NC 28712, (704) 885-2516.

More from Phase 4...

More new products are available through Phase 4 Distributors, and we would like to mention just a few of them. **XMODEM** for the C-64, comes with dial/auto answer and **VIP Term**, and retails for $89.95 (Cdn). **Printmaster**, a program that enables the user to mix font styles on the same output, as well as rotate graphics, costs $47.95 (Cdn.).

There is also a whole range of Precision Software products for the C-128: **Fleet 128**, **Write & Spell**, **WordPro 128**, **SuperScript 128** and **SuperBase 128**.

For the new Amiga owners, it may be useful to know that Phase 4 will be supporting the machine fully. At the moment, the following Amiga products are in stock: **Maxicomm**, **Amiga Programming Library** and **Curta Graphic Tablets**.

For more information, contact Phase 4 Distributors Inc., 7157 Fisher Road S.E., Calgary, Alberta T2H 0W4. □

When a picture needs to be filled in, seven different patterns are available, plus a pattern editor, to create 8 by 8 custom patterns.

Sky Travel

Sky Travel from Deltron, Ltd., c/o Dr. F.H. Covitz, 155 Deer Hill Road, Lebanon, NJ 08833. Price $35.00 (US), plus $2.00 for shipping and handling.

TPUG Magazine has received several calls since **Sky Travel** was reviewed favourably in the June/July '85 issue. We are pleased to be able to tell you at last how to buy this product. Deltron has developed and is distributing a high-quality Plus/4 version of **Sky Travel**, as well as the C-64 version. The package includes a 1541 disk containing the planetarium program, and also a manual complete with charts, tables and diagrams.

Diskorganizer

Diskorganizer from The G.A.S.S. Company, 970 Copeland, North Bay, ON P1B 3E4 (705) 474-9602. Price: $29.95 (Cdn).

This product is a directory editor for the C-64 and 1541 disk drive.

With the **Diskorganizer** in hand, the user can rewrite all of a particular directory from scratch, changing places of files, grouping them by 'fencing' for ease of reading, scratch-protecting programs and files, and printing the directory from one disk to another.

There are eighteen command keys, and the function of each key is printed on the disk jacket, as well as in the manual.

Normally we provide information in this column based on the manufacturers' press releases, but I have seen this program in action, and I was impressed with its performance and potential usefulness.

Leroy's Cheatsheet

Leroy's Cheatsheet from Cheatsheet Products Inc., PO Box 111368 PGH., PA 15238 (412) 781-1551.

Leroy's Cheatsheet, a keyboard reference overlay for **Easy Script**, was announced in our November '85 issue. We have just received a new press release confirming that keyboard overlays for a whole variety of C-64 and VIC 20 software programs are available from the same company, including ones for numerous word processors, spreadsheets, databases, printers, language and utility programs, terminals and other popular programs such as **Doodle** and **Sky Travel**. The price for each 'cheatsheet' is $3.95 (US).

VideoFile

VideoFile from Bob Claster, Box 480210, Los Angeles, CA 90048 (213) 655-6795. Price: $49.95 (US), including postage and handling.

Claiming to be the first of its kind, **VideoFile** is a database program that will help VCR users keep track of the names of up to 200 video programs recorded on fifty different videotapes. It will allow three categories, listings either by category or all-inclusively, in order of length or alphabetically. The user can also make a printout of all his video programs.

The creator of **VideoFile**, Bob Claster, states that his product is well-suited for the home VCR user accustomed to 'time-shifting' (recording for viewing later). It searches the tapes for the best space to record new material, and immediately locates any show on the tapes. **VideoFile** can be used by both Beta and VHS owners.

Another feature — **VideoFile** will translate VCR counter numbers into an easily-understood hours/minutes format, so that the user can search for a space of two hours and fifteen minutes long on the tapes, instead of specifying a corresponding counter value. □

Calendar of TPUG Events

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 third Tuesday of the month, at 7:30 pm.

Central Chapter: Leaside High School, Bayview & Eglinton Aves. on the second Wednesday of the month, at 7:30 pm in the auditorium. For 'advanced' computerists.

COMAL Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the last Thursday of the month, at 7:30 pm in the Story Hour Room (adjacent to the auditorium). Note: location may change in October — for latest information call 445-9040.

Commodore 128 Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the third 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 cafetorium.

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

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

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.

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. □

JANUARY

MON	TUES	WED	THURS
		1	2
6	7 VIC 20	8 Central	9 Brampton
13	14 Hardware	15 SuperPET	16 Westside
20 New Users	21 Business	22	23 COMAL
27 Commodore 64	28	29 Communications	30

FEBRUARY

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

Bulletin Board

Fast Tracks competition

How can your computer help you to win a trip to the Indy 500? Simple. Design your own racetrack (complete with oil slicks, sound effects and whatever else you think is indyspensable), and submit your entry, on disk, to Activision, together with the sticker from their specially-marked **Fast Tracks** packages. (You will have guessed by now that **Fast Tracks** is their hottest new racetrack simulation game). It is distributed in Canada by Beamscope, the sponsor of the Dream Track competition.

Entries will be judged by professional race car driver Bobby Rahal, and you have until February '86 to send them in to Activision at PO Box 7286, Mountain View, CA 94039. See the specially-marked **Fast Tracks** packages for further details. Approximate price: $35.00 (Cdn).

B-128 group discussion

There has been some discussion concerning possible B-128 users meetings taking place in the Southern Ontario area (the Hamilton area has been suggested). We are expecting to hold the first meeting in mid-February, and Jim Butterfield has agreed to attend occasionally. The meetings may occur on a twice-yearly basis. If you are interested, contact B-128 librarian Paul Aitchison by BBS, CompuServe, Delphi or correspondence.

And from America...

So far, the only support for the B-128 appears to be from users' groups. There is one based in Illinois (CBUG) that we know about, and a 'national' one. The National B128 Users Group (BUG) publishes a monthly newsletter covering "important and invaluable information re the B128 family", whereas the local CBUG group provides "a more personalized forum".

A substantial quantity of invaluable utility programs is available through these two groups. Membership in BUG costs $20.00, and gets you the newsletter. Add $5.00 for first class. Contact National B128 Users Group, c/o Marlin Schwanke, 701 E. North Avenue, #C, Lopoc, CA 93436.

Killer whales join TPUG?

We hauled this little item aboard from the September issue of *The Ship's Log*, the newsletter of the Silver State Computer Users Group:

"Researchers at Marineland in Palos Verdes, CA, are now using PCs to gather information about Killer Whales. The eventual goal? Making the giants computer literate!!" A noble porpoise indeed.

New VIC catalogue

The new VIC 20 library catalogue is now available. TPUG would like to thank the following people for spending literally hours on its assembly. In alphabetical order: Richard Best, Doris Bradley, Ruth Bradley, Anne Gudz and Louise Redgers. Contact the TPUG Office, 101 Duncan Mill Road, Suite G7, Don Mills, Ontario M3B 1Z3, for your incredibly fat, fact-filled and exciting new catalogue.□

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

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 $195 (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.

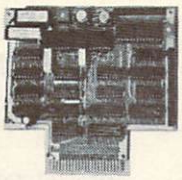
For further information call TPUG Inc. at (416) 445-4524, ask for Alan.

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.

DIADACS 1 «» TAKE CONTROL WITH YOUR COMMODORE 64

16 CHANNEL 12 BIT
A/D CONVERTER
D/A CONVERTER 12 BIT
DIGITAL OUTPUT 10 LINE
DIGITAL INPUT 12 LINE
REAL TIME CLOCK



* 12 BIT RESOLUTION - A/D CONVERTER 30 Conversions per second. 0 to 10 volt input. Controlled from Basic or Machine Language. * 12 BIT DIGITAL TO ANALOG CONVERTER - 0 to 10 volt output. * C-mos REAL TIME CLOCK CALENDAR - Year, day, month, day of month, day of week, hour, minute, second. Battery Backup - Set it once and forget it. * DIGITAL INPUTS 12 TTL compatible digital inputs for monitoring switches, relay closures, burglar alarms, etc. * DIGITAL OUTPUTS 10 TTL compatible outputs provide for controlling relays, digital displays, turning on alarms. On board +/-15 volts DC @ 20mA generated. Connectors for connection to external equipment. DIADACS 1 comes complete with user's manual and software driver.
Assembled and tested .....$279.00 US Dollars.
Add $3.00 for shipping U.S.A. Send check, money order, or MC/VISA.

MINI DIADACS 1
8 CHANNEL, 8 BITS A/D CONVERTER with 12 DIGITAL OUT and 12 DIGITAL IN.....$99.95 US DOLLARS

FAST DIADACS 1
SAME AS DIADACS 1 BUT HAS ON BOARD ROM SPACE and 100 MICROSEC CONVERSION.....$325.00 US DOLLARS

CONTROL MODE

New Software Package by TJN for the Commodore C64 and DIADACS 1. FEATURES:* DATA BASE MODIFICATIONS * ANALOG DEVICE DISPLAY *CONTACT INPUT DEVICE DISPLAY * CONTACT OUTPUT DEVICE DISPLAY * CONTACT OUTPUT CONTROL ** FUTURE SOFTWARE ENHANCEMENTS: * SEQUENTIAL CONTROL PACKAGE Will allow the user to build complex control sequences for home security, energy management and many other applications * ARCHIVE DATA STORAGE * REPORT GENERATION log of device values over several selectable time intervals using archival data. * TRENDING A plot of device value versus a selectable time interval.
CONTROL MODE I Without future enhancements.....$125.00 US DOLLARS
CONTROL MODE II With all enhancements.....$250.00 US DOLLARS

KMMM PASCAL

KMMM PASCAL for Commodore by WILSERV provides an EDITOR,EDITOR/COMPILER, and a TRANSLATOR. KMMM PASCAL is virtually a complete implimentation of Jensen and Wirth PASCAL and provides all of the string functions supplied by USCD PASCAL.
KMMM PASCAL on 1541 disk.....$99.95 US DOLLARS

CBM MICROTECH
P.O. Box 102 • LANGHORNE, PA 19047
• (215) 757-0284

C-64™ • C-128™ • VIC-20™ • SX-64™ • Plus 4™ • C-16™ • B-128™ • PET/CBM™

One disk. 25 business programs, $29.95

Now for the good part:

The **Intelligent Software** integrated home/business/educational package comes on disk or tape at the ridiculous price of **$29.95** (plus five cents for postage + handling). It is not public domain or home-brew software; totaling over 50 pages of source code, it was designed and tested as a package to take care of all your data processing needs.

Customers write: *". . . accolades for the authors. This is as slick a deal as I have seen and more than adequate for all except fancy presentations. The best thing is the ease of use . . ."*

"I have come to consider these programs among the most valuable pieces of software I own."

The package includes:

Database: A complete multi-keyed fixed-record-length data base manager. Sort or select (using all relational operators: =, >, <, AND, OR, NOT, wild card) on any field, perform computations on numeric fields. Any operation can be performed on all, or only selected records. All fields completely user-definable. Can be used for any number of tasks, including accounting, mailing lists, inventory control, record, tape, or book cataloging, expense account maintenance, or as an electronic rolodex. One customer writes: *"I am especially impressed with Database, and have used it to replace a half-dozen other 'database' type programs I had been using."* Even if you use your Commodore for nothing else, this program alone might justify its expense.

Word Processor: A full-featured menu-driven word processor including: very fast file commands, screen editing, text locating and full control over margins, spacing, paging, indentation, and justification. *". . . . well done and highly functional . . . Provides an excellent alternative to the high priced word processors . . . this is an excellent buy. Highly recommended."* — Midnite Software Gazette. *"Provides good basic features."* — Compute's Gazette.

Copycalc: An electronic spreadsheet. Turns your Commodore into a visible balance sheet; includes screen editor. *"Excellent program for budgeting, estimating, or any math-oriented use . . . well worth the money. Highly recommended."* — Midnite Software Gazette.

Also included: **ReportGen, ReportMerge** (these interface W/P with Database to create form letters, statements, invoices, mailing labels, other reports.); **Baseball Statistician** (compiles batting statistics for a baseball league); several W/P utilities, including **Index** (indexes W/P's text files); several Database utilities, including **DBmerge** (facilitates multi-file database applications.), and **DBStat** (analyzes D/B files); a programming utility, **ASCII,** which converts text files (program listings) into program files; also **Checkbook**; **Inventory**; **Paper Route**; **Loan Analysis**; **Breakeven Analysis**; **Depreciation**; **Labeler**; **File Copier**; more.

Versions of the package are available for every Commodore computer having a minimum of 10k RAM (including the C-128 in 128 mode). All programs will support tape, disk, and printer. Price includes documentation; Calif. residents add 6%. Add $3 for credit card, COD, 8050 disk, or cassette orders (cassette not available for Plus4™ and 16™.) No personal checks from outside USA. This ad is the catalog; send $1 (deductible from order) for a sampling of program output.

Intelligent Software
Quality Software since 1982

Box A Dept. T-7
San Anselmo, CA 94960
(415) 457-6153

C-64 & C-128* OWNERS:

Stop playing "Hide-and-Seek" with your Disk software! Let your Commodore 64™ "memory" do all the work for you with **MASTERDISK©**. Easy Manual and Program Menu-Screens make everyone a "pro" in 1 day!

Here's a small "PEEK" at SOME of the FULLY-INTEGRATED POWER of PRODUCTIVITY at your fingertips. . .

- a SORTED Master Catalog of ALL your Disks (even those un-listable "protected" Directories: Names, ID's, File info, etc.)
- Search + List any File/Disk/ID in seconds, automatically, in any of SEVEN WAYS.
- Transfer ML-Programs or Files from Disk to Disk (CATEGORIZE your software)
- Stop disaster before it strikes: change duplicate ID's fast, without total re-copy.
- Restore accidentally-scratched files in a flash, with VERIFICATION that they're OK.
- Works with 1 or 2 Single Disk Drives (1541, MSD, etc.) with or without Printer (Commodore or ASCII).
- Handy-Reference Card for HEX/ASCII/BINARY-TO-DECIMAL VALUE TRANSLATION + FORMULAS, included FREE.
- All this + MUCH MORE, for just $34.95 (U.S.A.), $40.95 (Canada)

Now YOU too can get the "Software Edge"!

MASTERDUAL©, "Son of MASTERDISK©" . . .

same power, but with Dual Drive Speed!

- Same Catalog Features as MASTERDISK©
- Same Disk repair functions, too
- For all C-64 & C-128* systems with Dual Disk Drives (4040, MSD, etc.)
- Utilizes all Dual Drive functions for maximum speed and ease of use
- FREE Handy Reference Card for HEX/ASCII/Binary to Decimal Translation + Formulas
- Experience FULLY-INTEGRATED POWER for just $29.95 (U.S.A.), $39.95 (Canada)

NOW!! THERE'S SUPERMASTERDISK©

- Combines features of MASTERDISK©, MASTERDUAL©
  For $39.95 (U.S.A.) $44.95 (Canada)

***SUPERMASTERDISK©, MASTERDISK© and MASTERDUAL©** are also compatible with the NEW C-128 (in C-64 Mode)!

We ship within 24 hours! Send Money Order or Check to:

INTEGRATED-SOFTWARE SYSTEMS
P.O. Box 1801, Ames, IA 50010
VISA/M.C./C.O.D., Call **(515) 233-2992**
9 a.m. - 9 p.m. (Central Time) Mon.-Fri.
Add $3 Shipping/plus $2 Foreign, C.O.D.
IA Residents add 4% Sales Tax.

IN CANADA: Cricket Distribution Co.,
P.O. Box 953, Cobourg,
Ontario, Canada K9A 4W4
VISA/M.C./C.O.D., Call **(416) 372-3692**
9 a.m. - 6 p.m. (Eastern Time) Mon.-Fri.
Add $3 Shipping/plus $2 C.O.D.
Ontario Residents add 7% Sales Tax

ASK ABOUT THE SPECIAL PRICE FOR TPUG MEMBERS

Whether you have tens or hundreds of disks, now YOU can be the MASTER of your TOTAL Software Library!

Dealer Inquiries Welcome

advertisers' index

TPUG Contacts

TPUG OFFICE 416/445-4524

TPUG MEETINGS INFO 416/445-9040

Board of Directors		
President	Chris Bennett	c/o416/445-4524
Vice-President	Gerry Gold	416/225-8760
Vice-President	Carl Epstein	416/492-0222
Recording Sec.	Doris Bradley	416/782-7320
	David Bradley	c/o416/445-4524
	Richard Bradley	c/o416/445-4524
	Gary Croft	416/727-8795
	Mike Donegan	416/639-0329
	John Easton	416/251-1511
	Keith Falkner	416/481-0678
	Louise Redgers	416/447-4811
General Manager	Bruce Hampson	416/445-4524
TPUG Magazine		
Publisher	Bruce Hampson	416/445-4524
Editor	Nick Sullivan	416/445-9865
Assistant Editor	Marya Miller	416/445-9865
Production Manager	Astrid Kumas	416/445-9865
Ad Director	Bruce Hampson	416/445-4524
Meeting Co-ordinators		
Brampton Chapter	Jackie Bingley	c/o416/445-4524
Central Chapter	Chris Bennett	416/782-8402
C-64 Chapter	Louise Redgers	416/447-4811
COMAL Chapter	Donald Dalley	416/742-3790
	Victor Gough	416/677-8840
Communications	Darrell Grainger	c/o416/445-4524
Eastside Chapter	Judith Willans	c/o416/445-4524
Hardware Chapter		c/o416/445-4524
SuperPET Chapter	Gerry Gold	416/225-8760
VIC 20 Chapter	Anne Gudz	c/o416/445-4524
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	Bill East	c/o416/445-4524
Librarians		
COMAL	Victor Gough	416/677-8840
PET	Mike Donegan	416/639-0329
SuperPET	Bill Dutfield	416/224-0642
VIC 20	Richard Best	c/o416/445-4524
Commodore 64	Derick Campbell	416/492-9518
French	Baudouin St-Cyr	c/o416/445-4524
B-128	Paul Aitchison	c/o416/445-4524
Amiga	Mike Donegan	416/639-0329
TPUG Bulletin Board		
Sysop (voice, weekdays)	Sylvia Gallus	c/o416/896-1446
Assistant Sysop	Steve Punter	c/o416/896-1446

TPUG
LIBRARY DISKS

10 months of the latest, fabulous public domain software

ONLY

$79.95* CAN

SAVE 20% or more by using this offer

Subscribe now to start receiving the

TPUG disk of the month

*US orders US$59.95

Name _____ Membership # _____

Address _____

_____

☐ Cheque ☐ Money Order ☐ VISA ☐ MASTERCARD Amount $_____ Currency ☐ Can. ☐US

Credit Card # _____ Expiry date _____ Signature _____

Type of disk: ☐ C-64 ☐VIC 20 ☐ PET ☐ SuperPET (10 disk set)

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.

*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.

MIMIC

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)