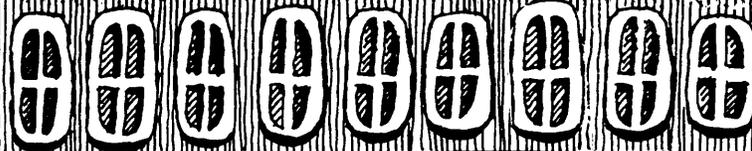


Pascal Winners

\$3.00

MICRO CORNUCOPIA

The
Micro Technical Journal



Feb-March 1986

No. 28

Hot Potpourri

- This Winter's Amiga**6
A software developer's first impressions of his Amiga.
- PC Main Lines — A System Primer**8
Want to know what's going on inside the PC and XT? Here's a circuit by circuit overview.
- Comdex (The Tale Of One Large City)**21
Nothing dramatically new, but many upgraded, cheaper versions of the old.
- Home Care For The Invalid Computer (Part 2)**34
We've dealt with shake and bake so it's time for chips and dips.
- Pascal Runoff — The Winners' Circle**50
What did 103 people write in Turbo Pascal? And who were the best?
- Rescuing Lost Text From Memory**52
A disk error doesn't have to spell disaster, especially when what you want still resides in memory.
- Drawing A Planter In 3-D**53
The best Kaypro 84 graphics we've seen (this one could grow on you).
- Turbo Pascal Inline Code From A .PRN File**64
No more hand assembly for CP/M Turbo.

SAVE OVER 30% ON OUR GIFT PACKS!
60-DAY MONEY-BACK GUARANTEE

How Borland's Three New Holiday Packs Will Fill Your Stocking Without Emptying Your Piggybank.

Three special packs with dazzling discounts that will help get you into a Holiday mood. You can get some of Turbo, most of Turbo, or all of Turbo—including the two newest members of the Turbo family, Turbo GameWorks™ and Turbo Editor Toolbox™. You also get our unmatched 60-day money-back guarantee, quality products that aren't copy-protected.

TURBO NEW PACK \$95.00.

You get the two exciting new members of the Turbo Pascal family,

- TURBO GAMEWORKS, Chess, Bridge, and Go-Moku, complete with source code and a 200-page manual.
- TURBO EDITOR TOOLBOX, all the building blocks to make your own editors and word processors, complete with source code and a 200-page manual.

TURBO HOLIDAY PACK \$125.00.

You get all three of the Turbo family classics for only \$125.00 (about a 30% discount). Turbo Pascal 3.0 and Turbo Tutor and Turbo DataBase Toolbox—all for just \$125.00.

- TURBO PASCAL combines the fastest Pascal compiler with an integrated development environment.
- TURBO TUTOR teaches you step-by-step how to use Turbo Pascal with commented source code for all program examples on diskette.
- TURBO DATABASE TOOLBOX offers three problem-solving modules for your Turbo Pascal programs: Turbo Access, Turbo Sort, and GINST, which generates a ready-to-run installation program that lets you forget about adapting your software to specific terminals.

TURBO HOLIDAY JUMBO PACK \$245.00.

This is it—the whole thing, the entire Turbo family including its two newest members. You get:

- Turbo Pascal
- Turbo Graphix Toolbox
- Turbo Tutor
- Turbo DataBase Toolbox
- Turbo GameWorks
- Turbo Editor Toolbox

and you pay only \$245.00 for all six! Which means that you're getting everything at only about \$40 a piece. Quite a holiday deal. (And if you already own one or several members of the Turbo family, be creative—nothing can stop you from buying the Jumbo Pack, picking out the ones you already have and giving the rest as holiday gifts to family and friends. At these prices you can afford to give to others and to yourself.) Speaking of Holidays, this offer lasts until March 31, 1986. (At Borland, we like to make the Holidays last.)



BORLAND
INTERNATIONAL

4585 SCOTTS VALLEY DRIVE, SCOTTS VALLEY,
CA 95066 PHONE (408) 435-8400 TELEX 172373

Copyright 1985 Borland International BI-1017B

Turbo Pascal and Turbo Tutor are registered trademarks and Turbo DataBase Toolbox, Turbo Graphix Toolbox, Turbo Editor Toolbox, Turbo GameWorks, and MicroStar are trademarks of Borland International, Inc. WordStar is a trademark of MicroPro International Corp. MultiMate is a trademark of MultiMate International Corp. Microsoft is a registered trademark and Word is a trademark of Microsoft Corp. WordPerfect is a trademark of Corellite Software International.



NEW!

TURBO GAMEWORKS \$69.95.

Our new Turbo GameWorks offers games you can play and replay without Turbo Pascal or revise and rewrite with Turbo Pascal 3.0. We give you the source code, the manual, the diskettes and the competitive edge. Chess, Bridge and Go-Moku. State-of-the-art games that let you be player, referee, and rules committee all at once because you have the Turbo Pascal source code. Learn exactly how the games are made—so you can go off and make your own. And Turbo GameWorks is the only quality game you can buy that is not copy-protected. Sold separately, only \$69.95. (Just \$47.50 if you buy the Turbo New Pack.)

NEW!

TURBO EDITOR TOOLBOX \$69.95.

Build your own word processor—for only \$69.95! You get ready-to-compile source code, a full-featured WordStar™-like word processor, and a 200-page manual that tells you how to integrate the editor procedures and functions into your programs. With Turbo Editor Toolbox, you can have the best of all word processors. You can make WordStar behave like Multi-Mate. Support windows just like Microsoft's Word. And do it as fast as WordPerfect does it. Incorporate your new "hybrids" into your programs to achieve incredible control and power. Sold separately, only \$69.95. (If you buy the Turbo New Pack, the price drops to just \$47.50.)

**Holiday Gift Packs,
Turbo GameWorks™
& Turbo Editor Toolbox™**

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by
Credit Card call (800) 255-8008, CA (800) 742-1133.

Make checks payable to:
Borland International.

The holiday packs include an upgrade coupon for both options so you get BCD and 8087 support for \$39.95 (regularly \$55.00).

Carefully describe your computer system!

My computer's name and model is: _____

The disk size I use is: 3 1/2" 5 1/4" 8"

Name: _____

Shipping Address: _____

City: _____ Zip: _____

State: _____ Telephone: _____

NOTE: Turbo Editor Toolbox and Turbo GameWorks are available for the IBM PC and true-compatibles using Turbo Pascal 3.0 ONLY.

***Gift Pack Offers Last Until March 31, 1986**

* Turbo Holiday Jumbo Pack	\$245.00	Quantity _____
* Turbo Holiday Pack	\$125.00	_____
* Turbo New Pack	\$95.00	_____
Pascal	\$69.95	_____
Pascal w/8087	\$109.90	_____
Pascal w/BCD	\$109.90	_____
Pascal w/8087 and BCD	\$124.95	_____
Turbo DataBase	\$54.95	_____
Turbo Graphix	\$54.95	_____
Turbo Tutor	\$34.95	_____
Turbo Editor	\$69.95	_____
Turbo GameWorks	\$69.95	_____

These prices include shipping to all U.S. cities. All foreign orders add \$10 per product ordered.

Amount: (CA add 6% tax) _____

Payment: VISA MC Check Bank Draft

Credit Card Expiration Date: _____

Card # _____

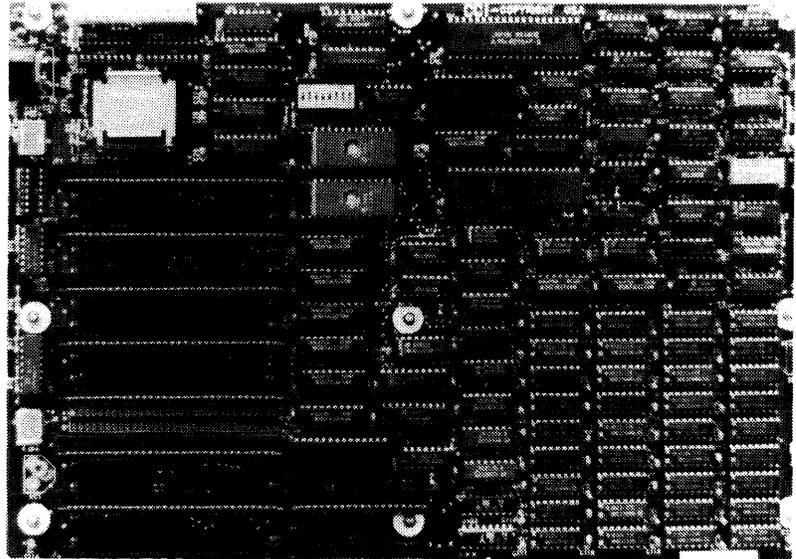
COD's and Purchase Orders will not be accepted by Borland International. California residents: add 6% sales tax. Outside USA: add \$10 and make payment by bank draft, payable in U.S. dollars drawn on a U.S. bank. H8

THE CHALLENGER XT-186

NOT JUST ANOTHER CLONE

FULLY IBM COMPATIBLE BUT FOUR TIMES FASTER

CHALLENGER XT-186 SINGLE BOARD COMPUTER



Standard Features :

- Intel 80186 CPU, 8MHz.
- On-board 640K, parity checked RAM memory
- 8 slots for I/O Interface Board
- 4 channel 8237 DMA
- 8 channel 8259 Interrupt
- 3 channel 8253 Timer
- IBM PC-AT-like Bus with true 16 bit data path

Optional Features:

- 10 MHz operation
- 8087 Numerical Processor support

The Challenger XT-186 offers full IBM PC-XT compatibility. It supports industry standard operating systems: MS/DOS, PC-DOS, 1.0 through 3.1, UNIX and CP/M and runs all existing software with higher performance. The XT-186 mother board can directly replace any existing PC or PC-XT mother board, as it has the same physical dimensions and mounting holes. The Intel 80186 and 640K on-board 16 bit path RAM makes the XT-186 the fastest PC currently available.

MOTHER BOARD, 8MHz, 640K, RAM, 8 I/O slots

\$795.00

Add-On Cards:

Color Graphic Board: RGB and composite. 320X200 Color/B&W, 640X200 B&W only, light pen input **\$120.00**

Multifunction Card: Real Time Calendar Clock with Battery Backup, 2 RS-232 ports built-in, 1 parallel port **\$120.00**

Floppy Diskette Controller Card: Double-sided, double density, can support 4 floppy diskette drives **\$70.00**

Monochrome Display Card: Supports 256 character codes on 4K ROM, 4K display buffer on static RAM with TTL monochrome video output **\$110.00**

Monochrome Graphic/Printer Card: Two graphic pages with a resolution of 720X348, software supplied to allow the use of the basic graphic command, centronic parallel printer interface **\$140.00**

Visa and Mastercard Accepted

Please Add \$3.00 Shipping

HOLLISTON COMPUTER

P.O. Box 615, Chepachet, R.I. 02814

401-568-0522

MICRO CORNUCOPIA

The Micro Technical
Journal

Editor & Publisher
David J. Thompson

Operations & Advertising
David Pogue

Associate Editors
Rebecca Ozrelic Gary Entsminger

Accounting
Sandra Thompson

Graphic Design
Craig Lannes

Office Manager
Tracey Anthony

Technical Department
Bruce Berryhill Larry Fogg

Staff Assistants
Dorcas Dsenis Cary Gatton
Laura Logan Renee Katter

MICRO CORNUCOPIA (ISSN 0747-587X) is published bi-monthly for \$16 per year by Micro Cornucopia Inc. 155 NW Hawthorne Bend, OR 97701. Second-class postage paid at Bend, OR and additional mailing offices. POSTMASTER: Send address changes to MICRO CORNUCOPIA, PO Box 223, Bend OR 97709.

SUBSCRIPTION RATES:

1 yr. (6 issues).....\$16.00
2 yr. (12 issues).....\$30.00
3 yr. (18 issues).....\$42.00
1 yr. (Canada & Mexico).....\$22.00
1 yr. (Other foreign).....\$30.00
Make all orders payable in U.S. funds on a U.S. bank, please.

CHANGE OF ADDRESS: Please send your old label and new address.

Micro Cornucopia
PO Box 223
Bend, OR 97709

CUSTOMER SERVICE: For orders & subscription problems call 503-382-5060, 9 am to 5 pm, Pacific time, M - F.

For technical help call 503-382-8048, 9 am to noon, Pacific time, M - F.

Copyright 1986 by Micro Cornucopia Inc.
All rights reserved
ISSN 0747-587X

EDITORIAL

By David J. Thompson

This Is Not An Editorial!



I've been asked by the graphics department to come up with a name for this column. Most people think of this as an editorial. Even I've thought of it as an editorial — but careful, diligent, and exhaustive research has indicated otherwise.

Editorials are short and dull. This column, on the other hand, is long.

I've heard that editors are often ex-writers who've been kicked upstairs so they can't do too much damage. They're allowed to write editorials because no one reads them. And since editorials are supposed to be opinion, editors don't even have to get their facts straight. (Perfect job for a seasoned newspaper reporter.)

This column may wind up named "Cornucopious Comments," "Editor's Column," "All That's Fit, We Print," or something equally inspirational. (If you have any ideas, get them in quickly!)

SOG V

The fifth running of the Semi-Official Get-together (SOG V) is now official and speakers are already champing at the bit. If you missed the gate last year, now is the time to get on track for 1986 (July 24-27).

We'll be announcing all the particulars in the April issue (no foolin') but it should be quite similar to SOG IV. We'll start off with white water rafting and end up in the wilderness. Sandwiched somewhere in between will be some truly great technical sessions.

We're looking for expertise on expert systems, Modula, the PC bus (and design), 68000, Amiga, 32000, C, Pascal, prolog, and anything else that sounds (graphics & sounds) interesting. If you want to volunteer someone (yourself?) get your card or letter in to:

Turn In A SOGgy Speaker
Micro Cornucopia
PO Box 223
Bend, OR 97709

Remember, fame, glory, prestige, and the thanks of all mankind (and women too) are only part of the benefits of speaking at the SOG. (If we think of any others, we'll let you know.) Also, if you have any article ideas on the above subjects, we're interested in that too.

Bring Spouses & Kidses

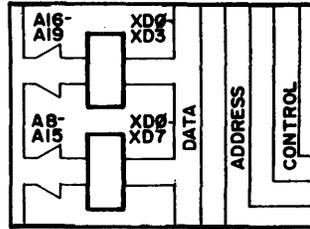
We'll be doing more of the very popular tours around the area for the other halves and quarters and by popular

(continued on page 86)

Features

8 PC Main Lines — A System Primer

Want to know what's going on inside the PC and XT? Here's a circuit by circuit overview.



6 This Winter's Amiga *A software developer's first impressions of his Amiga.*

18 WordStar Function Key Reassignment *Peek inside this venerable editor.*

21 Comdex (The Tale Of One Large City) *Nothing dramatically new, but many upgraded, cheaper versions of the old.*

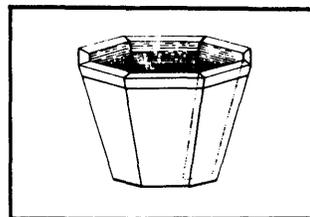
42 Automatic 8" Disk Drive Motor Control *Give your drives a break with this simple project.*

And Featuring

50 Pascal Runoff — Winners' Circle *What did 103 people write in Turbo Pascal? And, who were the best?*

52 Rescuing Lost Text From Memory *A disk error doesn't have to spell disaster, especially when what you want still resides in memory.*

53 Drawing A Planter In 3-D *The best Kaypro 84 graphics we've seen (this one could grow on you).*



63 Controlling A Kaypro & A PC Cursor *Want to know how to control your blinkin' cursor?*

64 Turbo Pascal Inline Code From A PRN File *No more hand assembly for CP/M Turbo.*

34 Home Care For The Invalid Computer (Part 2) *We've dealt with shake and bake so it's time for chips and dips. Now we're down to looking at pins.*

75 Designing A Communications Program For Data Access *Not all the problems are obvious when you need to access a major data base. This sample solution is written in FORTH.*

97 Having Your Pascal And Modula, Too *Modula II is not another foreign language, it's practically native tongue for Pascal fanatics.*

Columns

14 S-100 Bus S-100 memory discussion.

16 In The Public Domain *Steve Leon's warning of impending disaster (be careful what you run).*

25 C'ing Clearly *Writing readable code in C.*

30 Xerox 820 Column *How to talk to a Z80 SIO.*

37 86 World Potluck from Laine: *concurrent CP/M, speeding up Turbo Pascal, Turkey.*

48 Kaypro Column *A scientist looks at screen reflections. (And sees more than himself.)*

70 Pascal Procedures *Easy way to add inline object code.*

84 On Your Own *Which programs make good shareware, which don't.*

91 Culture Corner *Table talk.*

Future Tense

By Gary Entsminger

94 Tidbits *Amiga, cube rooters root for their favorite cube root, Z80 hard disk drive.*

104 The Last Page *Modula's miraculous modules.*

LETTERS

Suggestions For Micro C

Micro Cornucopia has broadened its focus this year, and I think that is good. The question is how much is good, and when do you start to lose enough depth to carry on? You have to be careful to remain distinctive — not just for the purpose of being distinctive — but to provide something special, something people will seek out! I'm not sure I can say what that is, but I have some suggestions.

I appreciate your efforts on hardware, but I don't think most construction is economically viable. First you have to find the parts, build the project, and then debug it, and end up spending more money than it would have cost assembled. It just doesn't pay! I like learning about the functions of the hardware, but I think the articles about hardware should cover what I might do. Like a modification to my computer or its peripherals. Or adding equipment/peripherals to my system. Or assembling sub-assemblies into an operational system. This is the limit in today's world.

The cost of a pc board, components, and test equipment makes other projects impossible for 95 percent of the population. More value would come from configuring hardware for different systems and limited trouble shooting experiences.

How about a list of repair facilities for different components and manufacturers. And maybe a watch dog service on bad software and hardware, and help for victims. How about fixes for software and hardware bugs? Or people to contact for information. Or more want ads at really cheap prices, so people can trade their stuff.

John L. Johnson, Jr.
707 Edge Hill Rd.
New Bern NC 28560

Editor's note:

Sol Libes told me the same thing a year ago, only he predicted the end of all hardware projects: construction, mods, everything. He predicted that computers would go the route of calculators. No one would fix them or modify them. They would be too small and too cheap to bother with.

We are looking for hardware projects

that can be done in a weekend. This means limiting the scope a bit but we are definitely looking. I still believe that where there's smoke there's a hot soldering iron.

As for keeping Micro C unique, I've spent many sleepless nights struggling with this (I'm open to suggestions). We're leaving our old, comfortable niche because we won't survive if we don't. In our case, I hope we're doing more than simply moving over to the next fad; I hope we'll be able to move right out to the front edge of technology in a way that will not only keep you up to date (and excited like we are) but also make it possible for those of you with limited budgets (by definition a budget is limited) to participate in the quantum leaps that technology is making.

Perfect Footnotes

I'm working on a large manuscript and have a problem with Perfect Writer whenever a footnote is too large or comes too near the bottom of a page of text being formatted. Rather than being carried over to the bottom of the next page, the footnote is either lost or jumbled into the text.

Do you know of a public domain or inexpensive program which will fix this bug? Aren't there the same types of patches for Perfect Writer that there are for WordStar? I've heard of a program called Bigfoot, but I think it's only for endnotes, not footnotes. Can you elucidate?

Now some questions about the Pro-884 MAX ROM: Does it work on an '85 2X? Is there a type-ahead buffer? Will it work with Xtrakey? Can you make it automatically date-stamp files (with a built-in clock)? Does it do the quick warm boots? Is a notepad function built in? What about the \$15 rebate for purchasers who send back their other ROM?

Peter Barub
313 Dufour St.
Santa Cruz CA 95060

Editor's note:

About Perfect, we don't have a clue. Perfect kind of conflicts with the personalities here at the office. As for the 884 Max, it's nearly Perfect. The Pro-884 will work with the latest Xtrakey, but it doesn't have a type-ahead buffer or date stamping.

However, Plu-Perfect's date stamper will date your files. It does warm boots out of ROM (no system disk required after initial system boot up).

Famous Last Words

I was so sorry to hear that Kaypro was following the masses into the 16-bit arena. But then, assurances were given that the 8-bit machines would always be supported by Kaypro, and the overwhelming availability of CP/M programming would keep the 8-bit Kaypros alive.

Well, true to my (and many others') expectations, it would seem that 8-bit support is dying. Matter of fact, for one major software supplier, it's already dead. In the December "Profiles" (Volume 3, No. 5) there is a two-page advertisement from Micropro on pages 64 and 65 extending only to Profiles readers the offer of WordStar compatible software. Calling them leads to the big disappointment that ALL of the 8-bit software included in the ad (which expires April 1986) is no longer available.

Anyone got a copy of Mailmerge (and manual) they want to sell for \$49 (that's Micropro's advertised price)?

Ronald A. Sanford
1336 Denbright Rd
Westview Park
Catonsville MD 21228

Preassembled Amiga Kits?

So you're frustrated? You want a machine that has great sound and graphics, but you don't like the Amiga because it doesn't come as a kit? Big Deal! A fair amount of your magazine is software, and with the Amiga there is a whole passel of hardware to control and loads of registers to dangle your programming toes in.

It's a 68000 based single board system that gives you access to its system busses via an 86-pin edge connector. It supports 32 colors at 640x200 and 16 colors at 640x400. With its custom VLSI to support animation, four sound channels, and a ROM that includes a text to speech algorithm, this multi-tasking machine can walk and talk.

Since when has not coming in a kit

LETTERS

stopped you from making hardware modifications. I'll bet some of your readers with solder behind their ears will find lots of "neat stuff" to add and modify on the Amiga. I'd love a 1 Meg RAM disk, or how about a battery backed-up clock? Show me how to add a 5.25" floppy. What about a voice and video digitizer? These are just the standards that should come first.

There is no technical magazine for the Amiga, and it would sure make a nice addition to the other systems you cover. Maybe you could consider it a kit that comes preassembled.

Bob Gobeille

805 Laporte Ave.
Fort Collins CO 80521

Editor's note:

Most people don't know that we're "wet behind the ears" because the flux hasn't dried yet. And, yes, I'm excited about the Amiga, too.

More On Amiga

Please consider including a column on the Commodore Amiga or, at least, a column on 68000 based machines limited to the Amiga, the Mac, and the Atari ST. I would enjoy articles on adding memory (perhaps exploiting some of the cheap memory boards for the PC) and devices to the Amiga's expansion port.

Peter B. Schroeder

11550 S.W. 108 Court
Miami FL 33176

Fixing Bugs With A RAM Disk

I appreciated your review of the Kaypro 2000 in the Issue 26 Kaypro Column. I'm writing this on my 768K 2000 at 37,000 feet over Alabama. Super productivity tool!

Two complaints you had about WordStar, and the problem of high power usage while the disk drive is running, are solved by a RAM disk. With the RAM upgrade on my machine, Kaypro sent a RAM disk program which I installed for 400K. On my WordStar program disk, I simply set up an AUTOEXEC.BAT batch file, which, on cold boot automatically copies the WordStar program files to the

RAM disk (phantom drive C:), and then logs to drive C:. Then, the disk drive practically never runs again until I finish writing.

The edited file must end up back on A:, since when the machine is shut down, the contents of the RAM disk dies. (Oops, final approach into Atlanta, please stand by. ^KX.)

OK, between planes, the RAM disk program is set up with the CONFIG.SYS program, as per Kaypro manual addendum instructions, so the RAM disk is set up before MS-DOS even looks for an AUTOEXEC.BAT file. Then when it does, that file has the following in it:

```
ECHO OFF
MODE LPT1: = COM1:
MODE COM1:24,E,7,1,P
(sends printer info to serial port)
COPY A:WS.COM C:
COPY A:WS.INS C:
COPY A:WSMSG.S.OVR C:
COPY A:WSOVLY1.OVR C:
COPY A:MAILMRGE.OVR C:
COPY A:D.COM C:
COPY A:COMMAND.COM C:
C:
```

From power-up to C> is 1:35, of which 1:10 is disk drive running time. Then, type WS A:MICRO-C.LET<CR>, and 4.2 sec. later, the Status Line shows up, ready to edit. All other functions in WS are much faster, from block moving to saving to whatever. I'm sure any other frequent drive access programs would benefit as much.

Lee D. Sundin

P.O. Box 392
Granville OH 43023

Private Domain

Some comments about your going into the private domain software market. I think it's the greatest idea since sliced bread. I'll be an enthusiastic participant. However, I have a suggestion. Increase the selling price of the package to \$24.95 and the author's slice of the cake to \$5. It seems that the public's perception is that if an item is less than \$25, then it can't be of much practical value. Besides, as an author, \$5 fits my mind (and pocket-

book) better than \$3.

H.D. Standard

P.O. Box 1925

Henderson TX 75653-1925

Editor's note:

Thanks for the sentiment. (I enjoy bread too, but I'm in a bit of a jam for something to put on it.) As for selling Private Domain software at \$25, I was a little against the idea at first. However, you definitely have advocates among the staff (and among other writers), so it looks like you're right on the money. The \$5 royalty would fit into \$25 just fine.

He Likes It!

MicroSphere RAM disk, MicroSphere RAM disk — where have you been all my computer life!

I've never written a product endorsement letter before, but the RAM disk demands it. It's truly given unbelievable speed and flexibility to my Kaypro 2-84. I bought the 512K model; it took less than an hour to set up. I use PluPerfect, and it works terrifically with the RAM disk.

John C. Brewer

Associated Press Chief of Bureau
1111 S. Hill St., Room 263
Los Angeles CA 90015

BG Micro's Xerox 820

I bought a Xerox 820 from BG Micro about eight months ago. Have there been any design corrections? I recall talking to someone who encountered an oversized resistor in the video display circuitry, but I don't have any details. Also, I'd like to convert the display output to composite video. Any ideas on these questions?

Jon Pollack

8110 Manitoba #210
Playa del Rey CA 90293

Editor's note:

Instructions for installing composite video are found in Micro C #15, page 45. As for design corrections and oversized resistors, you might try Mitch Mlinar, our Xerox columnist.

(continued on page 82)

This Winter's Amiga

By Martin Murray

INOVA

11311 Stemmons Freeway Suite 7
Dallas, TX 75229

A cursory reading of Amiga World Magazine would lead one to believe that all's well in Amiga land. Well, it is and it isn't. Read on for an inside look from a software developer's point of view.

The Amiga (a multitasking micro based on the MC68000) promises the most bang for the buck in 1986. Priced to sell at \$1295, it's impressive: main system unit with 256K RAM, an 880K 3.5" floppy drive, keyboard, mouse, operating system (and ROM!), and ABasiC.

Additional floppy drives are \$295 each, and another 256K of RAM is \$195. You can use your TV for display (cheap) or buy a fancy 4095 color Amiga RGB monitor for \$495. (Sony has a nice Trinitron monitor that works also.)

If you're planning to use the system for development work, your system should have two drives (one is external), 512K, and the Sony monitor (better quality image with less shake than the Amiga RGB). You should also get an Epson compatible dot matrix printer or one of the supported laser printers (Laser Jet or Laser Jet Plus).

Undocumented Features

The Amiga has great graphics, great sound, and at least one undocumented feature — an experimental operating system. A large portion of ROM (256K) is still on a floppy called KICK-START which you load before the Amiga can load AmigaDOS, the operating system.

After AmigaDOS loads, the fun really starts. For example, it doesn't always know when it's out of memory. And when it's out of memory it crashes. Even when it's not out of memory it crashes (but less often and less dependably).

In Commodore's defense, they "promise" to iron out the bugs, ROM the code, and get on with selling and supporting the Amiga. That's good news for third-party developers, who really need a solidified system in order to make software.

What's Really Exciting

Coming from a CP/M background

(Z80), we're quite impressed with the operating system. Especially the portion that runs the mouse, sizes the windows, and handles multitasking.

The multitasking looks like it's done very well (outside of the out-of-memory problem), specifically, the way tasks can be manipulated, the way you can move between foreground and background, the way you can run large numbers of tasks simultaneously. We've run 13 at once without problem; the limit appears to be 20.

I love the Amiga, even though I couldn't get excited about the Macintosh.

Running A Program

You get things done two ways on the Amiga. The first (seen by most users) is through INTUITION and WORKBENCH.

They display icons, move the mouse pointer, and provide powerful metaphors for disk and file manipulation. For example, to copy a file from disk to disk, you drag the icon for the file from its window on the source disk into the window of the destination disk.

WORKBENCH then takes over and copies the file, prompting you to swap disks at appropriate times if you have a single drive. To run a program, position the mouse pointer on the program's icon and click one of those cute mouse buttons twice.

For programmers, there's the Command Line Interface (CLI) — the equivalent of CP/M's CCP or MS-DOS's COMMAND.

You type the name of the program you want to execute (filenames can be up to 31 characters), add any parameters, and press return.

After a program's loaded, it calls the operating system to open a window. All of an individual program's I/O occurs in that window.

You can change the size and position of the window by pointing the mouse to particular gadgets in the window and dragging them. And you can make room for more windows!

Multiple programs can run concurrently in different windows on the screen. (The CLI is a program running in its own window, for example.)

INTUITION (and the mouse) lets you move between them easily.

Super Graphics

From low (320x200) to high (640x400) resolution, there's a wide range of display modes with between 16 and 32 colors displayable in each.

The graphics come from the custom chips (designed by Jay Miner). The chips know about eight hardware sprites and support the creation of "vsprites," or virtual sprites, which are sprite composites, for chip-controlled animation.

The Amiga has four-channel, stereo sound with right/left jacks connecting directly to your stereo amplifier. The music demo is amazing, and with the MIDI interface we should see a lot of musical applications making tunes soon. Additionally, a built-in speech synthesizer can be fully configured and controlled from something as simple as BASIC.

But, ABasiC (Amiga BASIC) isn't simple!

It's a rich, full interpretive BASIC implementation with graphics and sound commands and machine language and operating system calls.

Why fiddle around with BASIC, though, when you can write in nice, clean 68000 assembly language, right? Right — if you can get the \$450 development package, which contains (I hear) a macro assembler, linker, Lattice C, IBM PC cross-development tools, and seven manuals.

Editor's note: Hold the presses! Late word is that very soon, Amigas will be shipped with MicroSoft BASIC (shudders in the crowd), but not just any MicroSoft BASIC (slight looks of relief). This MicroSoft BASIC (do you have it memorized yet?) is supposed to have all the graphics support, sound support, plus full screen editor, and won't require line numbers (whew). I haven't seen it yet.

Real Work

We won't be working in BASIC; we're assembly language hackers at heart, and we'll be writing assembly language libraries and debuggers for starters. We're really glad to be skipping from the Z80 directly to the 68000

without going through the Intel confusion. The only advantage I can see of the 8086/8088 is the inherent relocatability of code on 64K boundaries. But that's not that big a deal, because you can write code to run anywhere you put it in the 68000.

The Amiga looks like a really good area for us, because there's almost no software available for it. In fact, most of the products that say they're shipping now are still vaporware. Deluxe Paint, however, is real, and it's really zingy. It's a full-featured graphics design program with animation written by Dan Silva of Electronic Arts.

The Future Now

Lots will be happening in Amiga land this year. I hear someone will offer MS-DOS so you can run PC software, and (with 8.5 megabytes of address) you can expect memory expansion boards, hard disks, modems, etc.

But no matter what happens (and despite the shortcomings now) the makers of the Amiga have made my day. I love it. (Now if I could just get my hands on that development package.)



FREE SOFTWARE FROM THE PUBLIC DOMAIN

User Group Software isn't copyrighted, so no fees to pay! 1000's of CP/M and IBM software programs in .COM and source code to copy yourself! Games, business, utilities! All FREE!

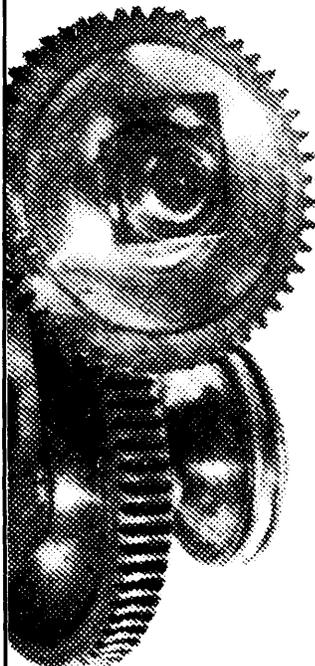
	Rent	Buy
IBMPC-SIG 1-390 Diskettes	410.	850.
IBMPC-BLUE 1-154 Diskettes	175.	435.
SIG/M UG 1-240 Diskettes	155.	650.
CP/M UG 1-92 Diskettes	45.	250.
PICO NET 1-34 Diskettes	25.	100.
KAYPRO UG 1-54 Diskettes	65.	200.
EPSON UG 1-52 Diskettes	65.	200.
COMMODORE CBM 1-28 Diskettes	25.	65.

Public Domain User Group Catalog Disk \$5 pp.
Send your check & specify computer format.
Rental is for 7 days after receipt, 3 days grace to return. Use credit card, no disk deposit.
Shipping, handling & insurance \$9.50 per library.
(619) 941-0925 Orders & Technical (9 to 5)
(619) 727-1015 24 Hr. 3 Min. Info Recording
Have your credit card ready!

National Public Domain Software



1533 Avohill Dr.
Vista, CA 92084
1-800-621-5640 wait
for tone dial 782542



BD Software, Inc., maker of the original
CP/M-80 C Language Development
System, knows

Time is precious

So the compilation, linkage and execution speeds of BDS C are the fastest available, even (especially!) on floppy-based systems. Just ask any user! With 15,000+ packages sold since 1979, there are *lots* of users . . .

New! Ed Ream's RED text editor has been integrated into the package, making BDS C a truly complete, self-contained C development system.

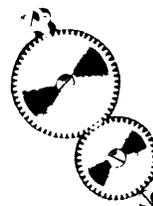
Powerful original features: CDB symbolic source-level debugger, fully customizable library and run-time package (for convenient ROM-ing of code), XMODEM-compatible telecommunications package, and other sample applications.

National C User's Group provides direct access to the wealth of public-domain software written in BDS C, including text editors and formatters, BBS's, assemblers, C compilers, games and much more.

Complete package price: \$150.
All soft-sectored disk formats, plus Apple CP/M, available off-the-shelf. Shipping: free, by UPS, within USA for *prepaid* orders. Canada: \$5.
Other: \$25. VISA, MC, COD, rush orders accepted.

BD Software, Inc.

BD Software, Inc.
P O Box 2368
Cambridge MA 02238
617 • 576 • 3828



PC Main Lines — A System Primer

By Gary Entsminger And Bruce Berryhill

Micro C Staff

"If you show an engine or a mechanical drawing to a romantic, it's likely he won't see much of interest. The surface reality is dull — lines, numbers. A classical person sees underlying form. Beauty's not what is seen, but what it means." Robert Persig, Zen And The Art Of Motorcycle Maintenance.

If you're new to computer hardware, or if you want to know more about the PC, read on. We're going to review the system. (Pass one Taiwanese XT-compatible main board, please, and don't scrimp on the layers.)

A PC system board consists of a microprocessor, ROM, RAM, DMA, I/O, system interrupts, timer, plus a system bus (address, control, and data lines) for communicating. Figure 1 shows the main lines between these subsystems, with the 8088 microprocessor at the heart of the system.

The microprocessor sends and receives information via busses.

The data bus is made up of 8 lines, each line carrying 1 bit, so data is sent and received 8 bits at a time.

The address bus is made up of 20 lines (20 bits of address can select one of over 1 million different memory locations). The processor uses the address bus to tell memory which memory location (byte) it wants to read from or write to.

The control bus is an informal collection of all the control signals needed to keep things coordinated (clocks, interrupts, ready lines, read, write...).

When the processor needs to send a message (write a byte) to memory, it selects a location in memory by putting that address on the address bus. Then it puts the message (byte) on the data bus and activates the write line on the control bus.

When the processor needs to receive a message (read a byte) from memory (or from an I/O device), it places a value on the address bus, activates the read line (on the control bus), and then receives the value on the data bus.

The processor also has under its command a DMA (direct memory access) controller. The DMA can also read from memory (or I/O) and write to memory (or I/O). Thus the processor can tell this controller to handle data transfers while the processor does other tasks.

For example, in order for you to continue typing while a disk is accessed, the microprocessor lets the DMA controller take over the data transfer between the disk and memory. The DMA transfers each byte of the disk sector from the floppy controller to memory, allowing the microprocessor to tend to other tasks, the keyboard, timer, etc.

Clocking And Latching

Of course, talking about putting a value on the data bus and actually doing it at the precise time that the reading device expects it are two different things.

Precise timing of data on the bus is the system's key to success. If the simple gates and the smart ICs are to

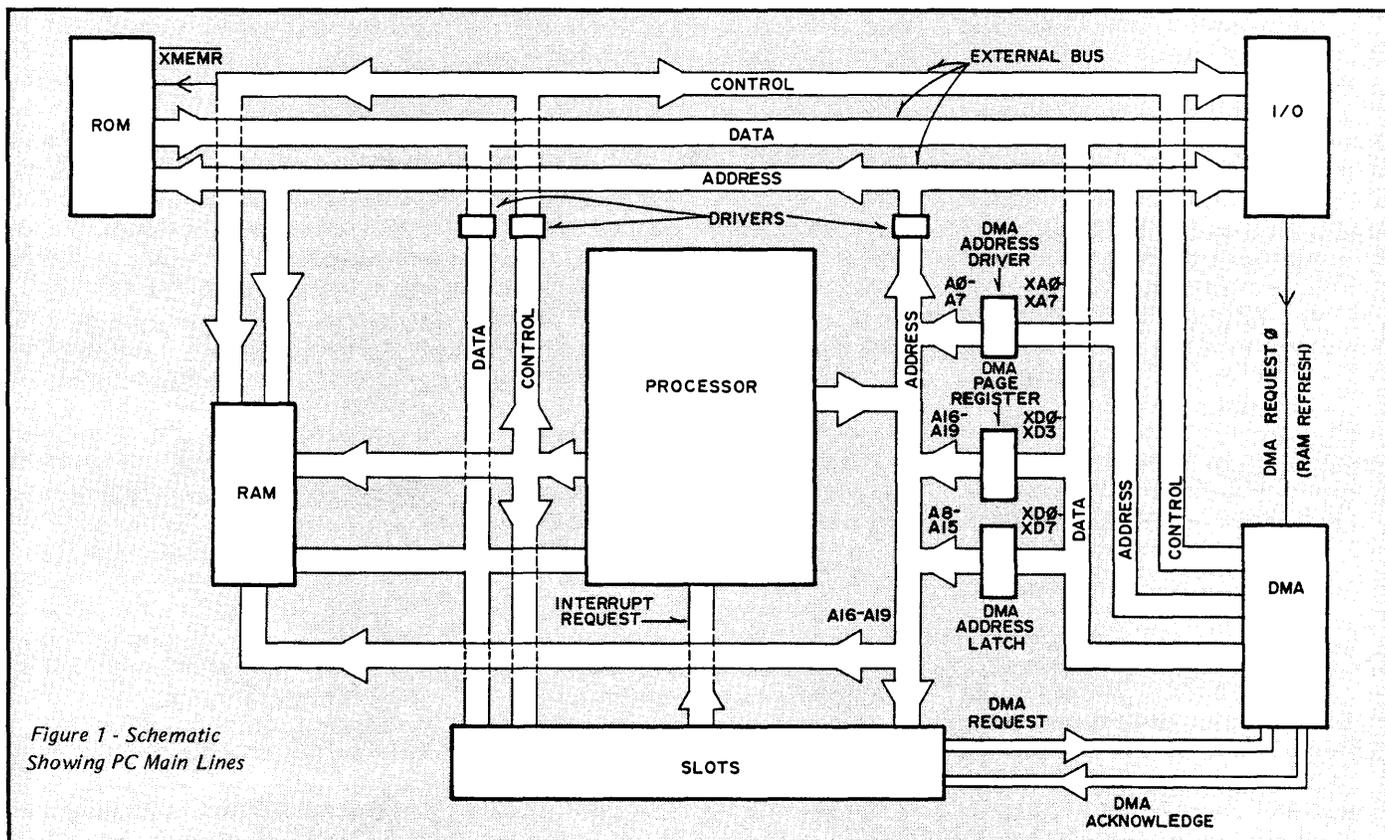


Figure 1 - Schematic Showing PC Main Lines

work together they have to dance to the same drummer. Otherwise voltages on the busses will simply flop around unpredictably.

The 8284A clock generator has the job of drummer, synchronizing all the signals within the computer.

The 8284A clock generator converts +5 volts into an asymmetric square wave to synchronize the logic gates.

A gate is a circuit which accepts an input state, deals with it logically, and outputs a corresponding state. One gate, a buffer, for example, outputs whatever it receives as an input. A "high" input remains a "high" output. "Low" outputs "low".

The first machine (or computer) cycle begins when a signal from the power supply indicates DC power has stabilized. Power is supplied through the 12-pin connector at one edge of the main board.

The 8284A sends its signal to the 8088 microprocessor's CLK pin (at 210 nanoseconds per period). The 8088 is divided into two separately functioning units — the Execution Unit (EU) and the Bus Interface Unit (BIU).

During one bus cycle the BIU talks to memory or I/O devices. One bus cycle requires four clock cycles or periods (T-states) to read or write a byte to/from memory or I/O. If the memory read is an instruction fetch, the byte is put in an instruction queue (first in, first out buffer). The queue holds 4 bytes on the 8088, 6 bytes on the 8086.

The EU gets its data and instructions from the instruction queue. The time required for the EU to execute an instruction depends on the complexity of the instruction.

40 Pins

The microprocessor communicates with the outside world via its 40 pins (Figure 2).

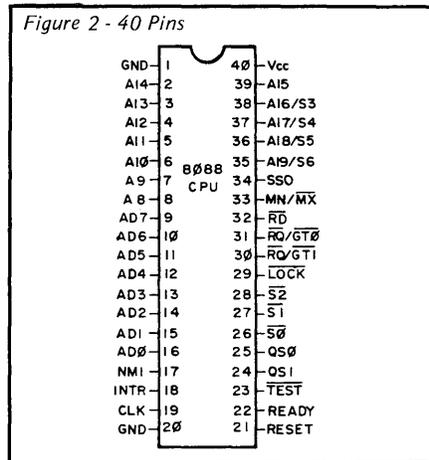
Twenty of these pins are multiplexed — address lines part of the time, and status lines or bi-directional data lines the rest of the time.

The 8088 uses pins AD0-AD7, A8-A15, and A16/S3-A19/S6 to address up to 1 megabyte of memory.

Eight of the address lines (AD0-AD7) become bi-directional data lines, and 4

of the address lines (A16/S3-A19/S6) become status lines after the first period of a bus cycle. By "time-multiplexing" 40 pins can accomplish what otherwise would take 52 pins.

To illustrate, let's step through a frequently used microprocessor cycle — fetching an instruction from RAM.



Reading From Memory

One of the 8088's registers, an instruction pointer (IP), always points to the next instruction in memory. That's the starting point.

The 8088 reads the IP and adds it to the code segment register.

There are several segment registers in the 8088. Since registers inside the 8088 (such as the IP) can hold only 16 bits, but an address is 20 bits, the 8088 must get the other four bits from somewhere. These come from a segment register, in this case, the code segment register. If the processor were reading a byte of data (rather than an instruction), then it would automatically add the data segment register to the register containing the address of the data byte.

It then puts the result on the address bus (A0-A19) and increments the IP during the first clock period. Again, because some of the address pins are used for other things, some of the address bits must be held by simple latches so that those address pins are free to become data pins or control pins. The Address Latch Enable signal controls these latches.

When the 8088 operates in maximum mode the 8288 Bus Controller gener-

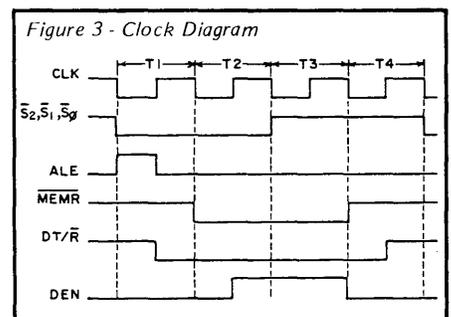
ates the Address Latch Enable (ALE) by decoding signals from the 8088's S0, S1, and S2 pins. ALE instructs the 74LS373 Octal Latches to hold the address. (In minimum mode, which doesn't allow co-processing, the 8088 generates its own ALE. The PCjr's the only machine we know of that runs in minimum mode.)

So, when ALE goes from high to low, the address from the address lines (A0-A7 and A12-A19) gets stored (latched) into two 74LS373s. A8-A11 are not multiplexed (so they don't need to be latched). They are simply buffered by a 74LS244.

Simultaneously, the Data Transmit/Receive line (DT/R) goes low, setting the direction of the 74LS245, allowing the 8088 to read the data lines.

At the end of the first period, the MEMR (memory read) line (from the 8288) goes low, in effect saying to memory, "The processor wants to read a byte." Memory doesn't put any information onto the bus until it sees this signal.

In the middle of period 2, the DEN (data enable) line goes high making the 74LS245's buffered output available on the data lines. At the beginning of period 4 the 8088 reads in the data. See Figure 3 — the clock diagram.



If memory (or an I/O device) isn't ready (or can't be interrupted), then it pulls the READY line low, and the 8088 simply marks time until READY goes high again. For example, if the 6845 video controller is accessing video memory when the 8088 wants to use it (video memory), the 6845 pulls the READY line low.

(continued next page)

Memory Write

A memory write cycle is similar to a memory read (or instruction fetch), except MEMW (memory write from the 8288) goes low instead of MEMR.

Main Board To Peripherals

Essentially, computing on the Main Board follows this simple pattern. The microprocessor, the heart (or brain) of the system, directs its accomplices via the control, address, and data busses. The signals on these busses are interpreted (via simple gates or software) by subsystems and peripherals into useful actions.

Three connectors allow the main board to communicate with the keyboard, reset button, and speaker. Eight slots allow it to communicate with the rest of the world.

On our PC (most clone boards look essentially identical to the IBM XT) power and the serial keyboard I/O

flank one corner; reset and speaker flank another; and the slots stand in a row — the eight biggest objects on the main board.

Each slot has 62 pins which are connected to the PC bus (address, control, and data lines), interrupt lines, DMA (request and acknowledge) lines, power, and any additional control lines. Thus these slots give board designers access to the very heart of the system.

I/O ports (and memory) respond to predetermined addresses. An I/O Read or Write happens much like a Memory Read or Write, except the IOR and IOW pins in the 8288 go low instead of MEMR and MEMW.

I/O devices (like memory) are responsible for monitoring the signals that let them know they're needed. These signals include addresses and control signals (such as read, write, I/O enable ...).

DMA, Co-processing, Etc., Etc.

One of the beauties of the 8088 is its almost presidential skill at delegating authority. To transfer data between I/O devices and memory, for example, it can call on the DMA (Direct Memory Access) chip, saying, "Take this much information from here and stick it there."

The DMA takes over the address, data, and some of the control lines and transfers information from memory address to memory address or between memory and I/O without microprocessor intervention.

Similarly, if a co-processor (like the math 8087) can handle a job, the 8088 passes control using status lines QS0 and QS1 or RQ/GT0 and RQ/GT1.

As the system becomes more complex, the 8088 farms out more tasks, and the "smart ICs" (the 8237 DMA, the 8259A interrupt controller, the 8255 parallel I/O circuit, etc., etc.) take

DISK SERVICE MANUAL II ■ PRINTER & PLOTTER ■ COMPUTER PHREAKING ■

Disk drives MUST be periodically cleaned and lubricated, and repaired as needed. Malfunctions can be devastating in lost programs, data and text; loss of business; upset customers; down time. YOU can maintain, troubleshoot, and repair drives WITHOUT EXPENSIVE OR DELICATE EQUIPMENT OR DIAGNOSTIC SOFTWARE - often in situ and in less time than it takes you to remove, pack, ship, receive, unpack, re-install, re-configure, and retest drives sent to drive repair shops!! Shipping drives is risky!! If you want the job done right, on time, and at minimal expense - DO IT YOURSELF! OVER 100 LABELED PHOTOS AND ILLUSTRATIONS OF STANDARD-BUS 5.25", 8" AND MICROFLOPPY DRIVES, AND SPECIAL DRIVES (APPLE, COMMODORE, SONY)!!

Chapter I: GENERAL. Chapter II: OPERATION ADVICE & TIPS. Chapter III: ERROR MESSAGES (and what they mean). Chapter IV: DIAGNOSTICS & TROUBLESHOOTING (how-to, step-by-step). Chapter V: MAINTENANCE. Chapter VI: SPEED ADJUSTMENT. Chapter VII: R-W HEAD ALIGNMENT (includes hysteresis and eccentricity). Chapter VIII: ELECTRONICS & REPAIRS (includes correct power/ground system wiring). Chapter IX: MISCELLANEOUS REPAIRS (T00 Sensor, T00 End Stop, Sector Index, Write-Protect, Head Loader, Compliance, Cone Assembly, Spindle Assembly, Module Assembly, Logic Boards, Spindle Motor, Door). Chapter X: DRIVE TEST STATION (professional shop plans). Chapter XI: REPAIR SHOP TECHNIQUES. Chapter XII: DRIVE ANALYSIS SOFTWARE CRITIQUE. Chapter XIII: DRIVE MODIFICATIONS. Appendix A: GLOSSARY. ONLY \$22.

DISK DRIVE TUTORIAL II ■

The detailed theory and practical facts of floppy drives, diskettes, FDCs, interfacing, formatting, and disk-stored software. A must for the Student, Programmer, and Computer Shopper (save \$\$\$)! Relates to drives of every manufacturer, and used in IBM, APPLE, TANDY, COMMODORE, KAYPRO, TI, HP, NORTH STAR, ATARI, DEC, etc. systems. DOZENS OF LABELED PHOTOS AND ILLUSTRATIONS:

Chapter I: GENERAL. Chapter II: DISK DRIVES. Chapter III: DISKETTES. Chapter IV: INTERFACING. Chapter V: FORMATTING. Chapter VI: SOFTWARE (compatibility and protection). Chapter VII: RECOMMENDATIONS. Appendix A: ADDRESSES. Appendix B: GLOSSARY. ONLY \$17.

STOCKPRO II ■

An effective, yet unique and unconventional method to select common stocks! Multiple applications demonstrated that stocks chosen by STOCKPRO subsequently increased in value by an average of 31% to 211% per year!! STOCKPRO II is upgraded with increased analytic powers. Use the shrewd statistical and intuitive approach of STOCKPRO II to outsmart the legions of Wall Street eggheads! Completely described and illustrated. Written in straightforward MICROSOFT BASIC.

ONLY \$15 for all Documentation (includes listing). ONLY \$50 for Diskette + all Documentation.

Consumertronics Co. ALAMOGORDO, NM 88310
2011 CRESCENT DR., P. O. DRAWER 537

PRINTERS & PLOTTERS ■

Printers and plotters are costly, with little relationship between cost, quality and capability! **PRINTER & PLOTTER MANUAL** describes:

- (1) The physical specifications, features, and control codes of popular printers and plotters.
- (2) Printer and plotter interfacing, problem areas and how to correct them. Some are hard to interface. Most can be memory-upgraded to save YOU much time and expense.
- (3) Many cost-saving tips and recommendations in the purchase and operation of printers and plotters.
- (4) Circuit diagrams of X-SWITCHERS. Independently switch one/two printers or plotters to one/two computers without disconnecting either. Schematics of Centronics Parallel, RS-232C and 20 ma Current Loop.
- (5) Additional sections devoted to PRINTER TYPES, PLOTTER TYPES, EXPANDING MEMORY, INTERFACING, ELECTRONIC TYPEWRITERS, RIBBON RE-INKING SECRETS, PAPER, PATCHING SOFTWARE, SPEED RATINGS, BUYING STRATEGY, and RS-232C PROTOCOLS.
- (6) BUFFERS, SPOOLERS, MEMORY UPGRADES, ELECTROGRAPHIC, TRACTOR FEED, DIP AND PANEL SWITCHES, DOT-MATRIX, DAISYWHEEL, INK JET, LASER, FLAT-BED, ROLLER-BED, DRUM, WORDPROCESSING, DATAPROCESSING, CAD/CAM, TIME-OUT PROBLEM, GRAPHICS TABLET, CALLIGRAPHY, PS TYPESETTING, ADVERTISING, PUBLICATION LAYOUTS, etc.
- (7) Over 100 printer- and plotter-related terms defined, including abbreviations, control codes, and interface terms.
- (8) Cleaning, lubricating, maintaining, and repairing printers and plotters, with actual examples.

Comprehensive and exhaustive - many illustrations and photographs - ONLY \$15.

COPIER MANUAL ■

The total cost of a modern photocopier can exceed that of a new car! Yet, copier prices and brand names have little to do with features and quality! Until **COPIER MANUAL**, almost nothing was available for you to compare copiers for features, quality, and price. Modern copiers are dropping in price, and have substantially increased features and quality. **COPIER MANUAL** describes in detail:

- (1) Evaluations of dozens of features, and prices of over 100 popular modern copiers manufactured and/or sold by A.B. DICK, CANON, GESTETNER, IBM, MINOLTA, MITA, MONROE, PANASONIC, PITNEY BOWES, RICOH, ROYAL, SANYO, SAVIN, SHARP, 3M, TOSHIBA, and XEROX!! Also includes evaluations of dozens of quality factors of many of these copiers exhaustively tested by us.
- (2) Our evaluations of each major copier manufacturer, which ones we recommend, and their addresses. An in-depth evaluation - advantages and disadvantages - of the copier we recommend as the best buy for an office copier.
- (3) Dozens of little known facts and tips on copier selection and operation. How to make substantial savings on purchases, maintenance, repairs and supplies. How to extend copier and drum life. Where to get cheaper supplies - what you can and cannot recycle. Copier myths debunked!
- (4) The optimum combination of paper and ink color to protect YOUR copyrighted publications and classified data from unauthorized use. How to copyright publications.

Frank, comprehensive and exhaustive. Dozens of illustrations, photographs, and tables. Designed to save YOU \$ Thousands. ONLY \$15.

COMPUTER PHREAKING ■

According to the FBI, less than 5% of all DISCOVERED computer crimes result in conviction! Computer crime, or "Phreaking" costs \$ Billions per year, and is clearly one of the most dangerous - yet most profitable and least risky - of all crimes! **COMPUTER PHREAKING** describes in detail:

- (1) Dozens of computer crime methods. Schemes include: Input Transaction Manipulation, File Alteration & Substitution, Unauthorized Software Modification, Code Busting, Wiretapping, Electronic Trespass, etc. Many actual examples - detailed case history based upon actual court records of a major group! Why/how Government, business and financial institutions are easily victimized by savvy Phreaks!
- (2) Numerous countermeasure, protection and security schemes - passwords to public key encryption methods. State-of-the-art techniques. Foil even the sharpest Phreaks!
- (3) Definitions of popular computer crime terms, including PIRACY, TROJAN HORSE, LOGIC BOMB, TRAPDOOR, GODFATHER, MUTANT, ZOMBIE, BODY SNATCHER, SILENT ALARM, CHEESEBOX, CANDYMAN, CODE 10, etc.

Learn how to become a computer crime fighter! Comprehensive, illustrated, frank. ONLY \$15.

COMPUTER SECURITY ■

Computer crime is rampant and increasing! Annual losses are \$ Billions - plus devastating losses in privacy, personal security, etc.! **DON'T BE VULNERABLE!**

We developed many versatile, simple-to-apply, virtually impossible-to-defeat, computer security techniques. These include, but go far beyond, passwords and ciphers. No hardware or ROM mods required.

Included are invulnerable BASIC cipher algorithms (encryption and decryption); BASIC program for computing, testing multikeys; cipher program design techniques; how to use zaps to protect disk files from unauthorized access; clever concealment techniques; secure operations procedures; review of security books - more!

\$1,000 CONTEST: WE WILL AWARD \$1,000 TO THE FIRST CONSTANT WHO CORRECTLY DECIPHERS OUR 25,000+ CHARACTER CIPHERTEXT! Now that we've polished off the lightweights, it's time for YOU to show us your stuff! Not a "public key", DES or "one-time pad" cipher. Included are details, contest rules, and many hints and clues. Complete descriptions and illustrations. ONLY \$25.

SUPER RE-INKING METHOD! ■

New printer/typewriter cartridge ribbons are costly, and yet may produce less than 5 hours of quality copy! And are an inconvenience to order - **WHEN YOU CAN FIND THEM!** Now, you can re-ink your own cloth ribbons to last about 10 hours of quality use for about 50 cents and 10 minutes of effort per ribbon. Not any ink will do! We developed the right combination of clay-free ink and carrier - both commonly and inexpensively available from stores in black and 4 colors. Includes complete plans for your own cheap motor-driven re-inker. Completely described and illustrated. **STOP WASTING MONEY ON RIBBONS! ONLY \$7.**

By John Williams, MSEE, former CS Professor, NMSU. We pay all U.S. shipping! (12%, \$3 min. foreign orders). Please allow 4-6 weeks for check-paid orders, else 2-3 weeks. Quantity buyers welcomed - Substantial quantity discounts! Custom editions for quantity orders! **PLEASE ORDER TODAY! FREE CATALOG for all orders over \$20 (else \$1).**

over significant portions of the process.

Bye

If we've whetted your appetite, and you want more immediate information about the PC's hardware, we recommend "The IBM Personal Computer From The Inside Out" by Sargent and Shoemaker and/or the "IBM XT Technical Reference." And of course, we'll continue to explore our personal clones and write about them on the pages of Micro C.

Amazingly, the microprocessor and its support chips stay coordinated, and everything works (usually). It's just a matter of timing.

There are reports, however, of programmers waking during near-sleepless nights from dreams of broken latches and spiraling bus lines.



THE BEST Z80 ASSEMBLER ON THE MARKET JUST GOT BETTER!

Z80ASM
NOW ONLY **\$49.95**

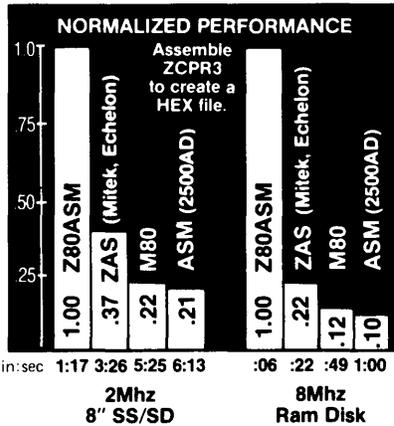
DON'T ASK HOW OURS CAN BE SO FAST ... ASK WHY THEIRS ARE SO SLOW!

"... a breath of fresh air ..."
Computer Language, Feb. 85

"... in two words, I'd say speed & flexibility",
Edward Joyce, User's Guide #15

Now fully compatible with M80 in .Z80 mode with many extensions. Time & date in listing, 16 char. externals, plus many other features.

To order, or to find out more about our complete family of development tools, call or write:



SLR Systems

1622 N. Main St., Butler, PA 16001
(800) 833-3061, (412) 282-0864
Telex 559215 SLR SYS

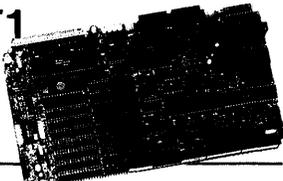


C.O.D., Check or Money Order Accepted

SHIPPING: USA/CANADA + \$3 • OTHER AREAS + \$10
Z80 CP/M compatibility required.

MSC-LAT1

\$649



KAYPRO™ users can share the advantage to LAT1. Just take off your main KAYPRO board and put LAT1-K into your cabinet.
All advantage of LAT1 is yours now! **\$674**



MSC-MTC

MSC-ICO

\$499



CP/M plus is a registered trademark of Digital Research Inc
Z80 is a registered trademark of Zilog Inc
Turbo Dos is a registered trademark of Software2000 Inc
Mountain Side Computer and ZENET are trademarks of Southern Pacific Limited

Distributors
England-Quanta systems 01-253-8423
Denmark-Danbit 03-662020
Finland-BB Soft 90-692-6297
India-Betamatix PVT Ltd. 0812-71989
Australia-LAMRON PTY. Ltd. 02-808-3666

Manufacturer and International distributor
SOUTHERN PACIFIC LIMITED
Sanwa Bldg., 2-16-20 Minamisaiwai, Nishi, Yokohama, JAPAN 220
Phone: 045-314-9514 Telex: 3822320 SPACIF J
Advanced single board computer technology company

ZENET NETWORK through twist pair

- 6Mhz HD64B180 (Z80 upward compatible) 512K byte on board (256K installed, 384K RAM/DISK)
- LAN: ZENET port 800K baud CSMA CD twist pair bus type upto 500 meters HDLC
- Floppy: 3.5, 5 and 8 inch, d/s density, d/s sided and d/s track automatic density/format checking
- Hard disk: SCSI interface on board
- Video: 80 X 24 characters (color) and 640 X 200 pixels color graphic 128K byte video RAM character set is downloaded from disk
- Timer: battery back up calendar

- Serial: RS232C X 2 and TTL X 1
- Parallel: Centronics type, 16 bit TTL, 78 bit keyboard port (32 characters FIFO)
- O.S.: Turbo Dos, MP/M (multiuser) banked CP/M plus (single user)
- Size: 10 X 6 inch 4 layered
- Assembled and tested
- BIOS source code available
- Completely faster than other Z80SBC

MSC-PCX
8088 expansion card for LAT1 soon available

WORLD SMALLEST COMPUTER

- Full personal CP/M system in palm 4mhz Z80 256K RAM (128K RAM DISK)
- Serial: RS232C X 2 automatic baud rate checking
- Parallel: Centronics type printer port
- Floppy: 3.5 inch micro floppy disk drive 800K byte (option 5, 3.5 inch drive d/s sided d/s track automatic density checking)

- O.S.: CP/M plus bank version
- BIOS source code available
- Completely faster than other Z80SBC

MSC-MTC/P

Full assembled pcb of MTC
Under \$189 in OEM quantity

\$299

Full featured CP/M plus system

- Z80 4mhz 128K Byte RAM Floppy: 3.5, 5 and 8 inch d/s density, d/s sided and d/s track upto 4 disk drives Automatic density/format check
- Serial: RS232C X 2
- Parallel: Centronics type, 16 bits 10, 78 bit keyboard port
- Timer: battery back up calendar
- Video: 80 X 24 high speed CRT controller
- O.S.: CP/M plus bank version included
- Size: 10 X 6 inch 4 layered

- BIOS source code available
- DRI CP/M plus manual \$50
- New word processor program for MSC-ICO ADD \$50
- Completely faster than other Z80SBC

MSC-HCS

Expansion card for ICO
RAM disk (upto 2M byte) and SCSI hard disk interface card for ICO with installation program

\$199

USA distributor
SOUTHERN PACIFIC COMPUTER PRODUCTS U.S.A., INC.
21 Altarinda Rd. Orinda, CA 94563
Phone: 415-253-1270
Dealer and distributor inquiries welcome



the alternative.

The SANYO complete word processing system

The alternative for WANG, Xerox or any dedicated word processing system.

At Micro Supply Organization we offer the lowest prices on Sanyo computers and software. With prices like these you can afford the convenience of owning and operating more than one computer. We also offer the User Support Hotline for questions concerning your computer or about software availability. Whether you need one or a dozen computers, Micro Supply Organization is the place to get them!

The MBC 1250 Accounting Software

We carry a complete line of ADS accounting business software. Buy G/L, A/R, A/P, Inventory and receive the Payroll module FREE. The current retail price is \$398. per module. **Now from MSO you pay only \$99 per module.**

COMPAT

Read, write, and format over 50 different CP/M and MS-DOS disks on your MBC 1250 CP/M computer. **\$79**
\$59 with purchase of MBC 1250

USER SUPPORT HOTLINE

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$1.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

Get more for your money!

SANYO MBC 1250

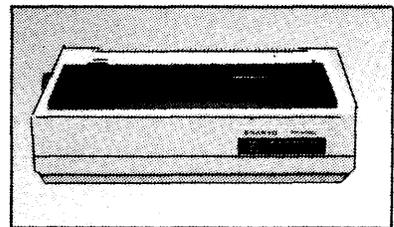


Including this FREE software:

- Wordstar • Spellstar
- Mailmerge • Calcstar
- Infostar • BASIC

8-Bit Integrated Computer with High-Resolution Display

- Two Z-80A CPUs (main and secondary) with no-wait mode and large 64KB RAM/4KB ROM memory capacity for fast execution.
- High-resolution 640 x 400-dot matrix display with full graphics functions.
- Choice of 33-line or 40-line text mode.
- CP/M operating system with editor, assembler and all standard utilities.
- Easy-to-use Sanyo Graphic BASIC.
- Two (MBC 1250) internal double-sided, double-density, double-track, 5 1/4" slim-type mini floppy disk drives with 640KB formatted capacity.
- Specially-designed detachable keyboard with coiled cable for easier operation.
- Interfaces for one Centronics printer one RS-232 port.



NEC PC-PR103A

Dot Matrix 110 cps

- 16 Wire print head, 40 cps NLQ
- 80 char/line or dot-image graphics
- Centronics parallel interface
- Tractors or Friction feed
- Pica, elite, condensed or proportional

Includes printer cable & full warranty

PR 5000

Letter quality Daisy Wheel Printer

- Letter-quality printer with 96-character daisy wheel printing element.
- Accepts paper up to 13 inches in width.
- 14 cps bidirectional printing.
- 10, 12 and 15 cpi printing pitches.
- Centronics parallel interface.
- Optional form tractor.

Includes printer cable & full warranty.

This complete package lists for \$3194.00

Computer & Printer
ONLY



\$899



the advantage.



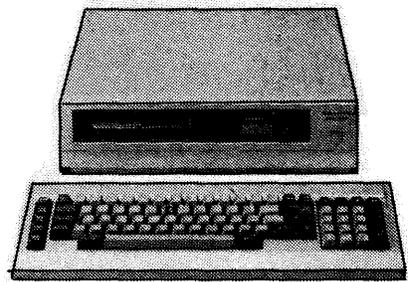
MBC 550 Series MS-DOS Computer

Complete with over \$1000 of FREE Software!

- MS-DOS 2.11 • MS-DOS 1.25
- Wordstar • Spellstar • Mailmerge • Calcstar
- Infostar • Easywriter I • BASIC

Dual 800K
Drive System
with Sanyo monitor
only **\$899**

ONLY \$599 (Driveless)
(without monitor)



OWL version 10 meg. internal + 800K Floppy **\$1399**

Disk Drive Options

A • (2) SS/DD 160K Drives
• Mounting Hardware
\$99

B • (2) DS/DD 360K Drives
• Mounting Hardware
\$199

C • (2) DS/DD 96tpi 800K Drives
• Mounting Hardware
• 800K Operating System
\$249

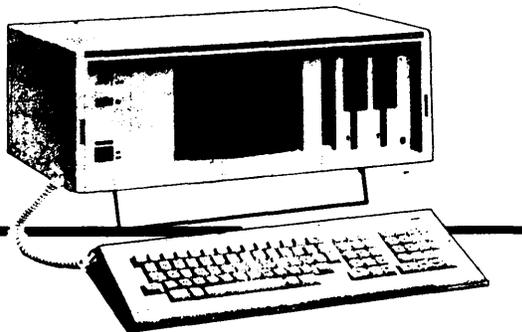
Single Floppy Disk Drives

TEAC FD-55A 160K \$59
TEAC FD-55B 360K \$109
TEAC FD-55F 800K \$119

FREE INSTALLATION of disk
drives with purchase of
SANYO 550.

MS-DOS & CP/M COMPUTER

MS-DOS 2.11 & CP/M 80 Computer
Two Fast Z80 A's & one 8088 - Processor



Portable Philips Computer

FREE Software Included
Wordstar (Wordprocessor) • Calcstar (Spreadsheet)

Data Plot + (Graphics Program)
MS-DOS - 2.11 CP/M 80 operating systems
BASIC - Microsoft BASIC
Dual 640K drives CP/M - 800K Drives MS-DOS 2.11
256K RAM expandable to 512K for MS-DOS
8087 Math Processor (optional)
64K RAM under CP/M Plus 32K RAM for video
Graphics - 512x252 pixels Alpha - 24x80 plus attributes
SASI interface built in (works with 10 meg OWL)
2-Serial ports bi directional ASYNC & BSYNC 60K bit/sec
9" Green screen or optional external monitor

\$699

The OWL **\$799**

SANYO internal 10MB Drive
Complete with all manuals, software, and cables.

Accounting Software

G/L, A/R, A/P, for small businesses with under 20 employees. Designed
for the MBC 550 in color.
Entire package only **\$149**

Accessories

MODEM

- Volks Modem 1200/300
baud modem with cable - **\$159**
- PR-5000 15 cps LQP - **\$279**
- DM-3112 12" Sanyo Monitor - **\$89**
- DMC-6113 13" Hi-Res RGB Color Sanyo monitor **\$299**
- 16K Printer Buffer - **\$84.95**
- Best prices on cables anywhere

At these prices you can enjoy the convenience of owning
more than one computer!

CHOOSE THE SYSTEM THAT'S RIGHT FOR YOU!

805/393-2247



All systems carry full 90 day warranty. Check in advance. Add 3% for VISA/MC.
Shipping & handling charges will be added to each order.

CASH PRICE ONLY For our catalog with complete
details and prices, send \$1.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

The S-100 Bus

By Dave Hardy

736 Notre Dame
Grosse Pointe MI 48203

If you're having trouble remembering how to make holes in your memory, or if you can't keep your windows clean, it's clear that you need to read on. Dave definitely knows how to do windows.

One of the banes of the S-100 user is the utility board with on-board memory. Many EPROM programmers, video boards, CPU boards, disk controllers, and other S-100 components (especially older non-IEEE-696 cards) require a section of the system addressing space that isn't available, especially if the user has a "single-board" S-100 system.

Some of the new S-100 products use the IEEE-696 standard's 24-bit addressing capability to avoid making a hole in system RAM. However, a surprising number of utility boards don't use the extended addressing, and thus require the user to chop a hole (or window) into system memory.

Hardware Windows

To make a window in memory, you need to modify the system's memory board. Sometimes (but not usually) a manufacturer will let you turn off part of the memory.

At least one S-100 single board Z80 manufacturer offers a "field-mod" for windowing memory that adds four ICs and about 20 wires. The resulting board looks like it's been attacked by a chain saw.

If you prefer less drastic modifications (windows without pains), you might use one of the circuits shown in Figure 1 or 2.

Data Decoding

Figure 1 shows a simple addition that will work on any memory board using 8-bit wide static RAMs or EPROMs. Figure 2 shows a more general circuit that's good for disabling a 1K block of memory in just about any system.

Both of these circuits disable the board's INTERNAL data bus driver (that is, the chip that gates the memory IC's data onto the system data bus), but each circuit controls the driver for a different reason.

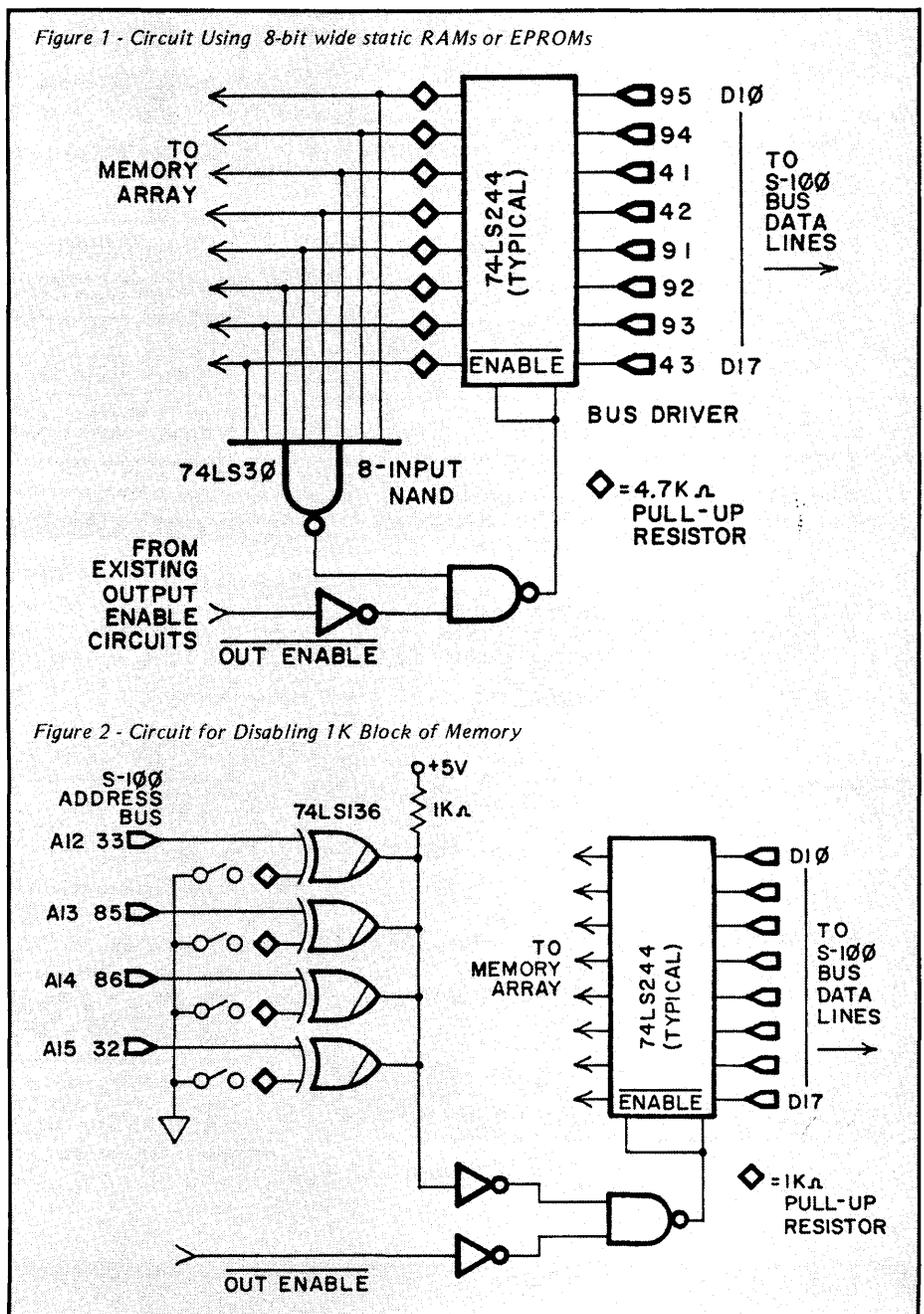
Figure 1's circuit assumes the user has removed the memory chip that would normally be located in the window's address area so the memory board would read all 1s in that area. Whenever all 1s are sensed on the INTERNAL memory data bus by the eight-input NAND gate, the gate prevents the bus driver from transferring the 1s to the main data bus.

In fact, this circuit will also turn off the memory board every time it reads

all 1s in memory (0FFH, to those of us who speak in HEX).

In other words, the circuit will work properly only in a system whose data lines are properly terminated (the lines are pulled up so they're all 1s while they're not being driven, so the processor sees a 0FFH anyway).

Most component-type S-100 machines (i.e., CPU on one board, memory on another) work fine with this circuit (including all of mine, strangely



enough), but most single board S-100 systems don't, usually because their designers saw no need to terminate an internal bus.

Address Decoding

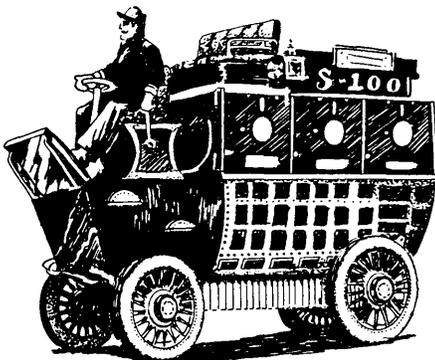
Unlike the circuit in Figure 1 which uses data content to disable system memory, Figure 2's circuit uses system address to disable memory, which is a more general, but more complex method.

Although this circuit probably couldn't be used in a single board system either, its general idea could be applied in any situation, even dynamic RAM, and over any range of addresses. Figure 2 is really just a simple address decoder that disables the memory's output driver whenever a specific memory area is addressed. This sample circuit is set up to make a window that starts at address 1000H and extends to address 1FFFH.

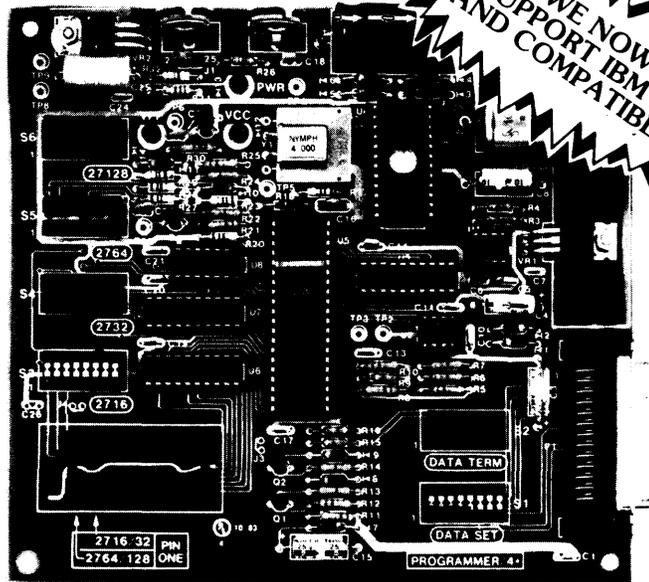
I have seen both of these circuits in S-100 boards, often, I suspect, as an afterthought of the manufacturer.

Next Time

Response from TurboDos-based multi-processor S-100 manufacturers has been rather slow, so I've had to delay our expedition into TD-land for a while. I hope to have some information and informal comparisons by the next issue.



PROGRAMMER/4+



A LOW COST ALTERNATIVE TO EPROM PROGRAMMING

- Reads and programs 2716, 2732, 2764, and 27128 EPROMS.
- Reads 2-16K ROMS.
- Direct connect to any RS232C terminal or computer.
- Plug selectable as either a data set or data terminal.
- All voltages made on board, (no power supplies needed).
- (User supplies power Xformer, 25.2 to 30 VAC C.T.1 Amp.).
- Power electronically switched, (can't damage EPROMS).
- Zero insertion force socket for EPROM.
- Programs, verifies, and dumps in both ASCII and hex.
- Edit buffer (like DDT).
- Saves hex and or image files to and from disk.
- Saves or loads all or partial buffer.
- Completely menu driven for ease of operation.
- Commands of Test, Read, Display, Save, Load, Program and more.
- Check sum calculation.
- All software on disk including well commented source code.
- Detailed owners manual including schematic
- All chips socketed.
- Not a kit! Completely built and tested.
- 48 hour dynamic burn-in and test before shipment.
- 90 day limited warranty on parts and workmanship.
- 24 hour return policy on repairs.
- Delivery from stock.
- PROGRAMMER 4+ WITH OWNERS MANUAL AND DISK. \$199.95**
- MS-DOS DISK ONLY \$15.00**

Order from



1659 Scott Blvd., Suite 1
Santa Clara, CA 95050
(408) 354-5084

VISA and MASTERCARD telephone orders welcome.

Please specify Disk format
CP/M 8" IBM format, KAYPRO II, XEROX 820, OSBORNE I, others.

Please specify method of shipment, UPS or Postal Service.
California residents add 6% Sales Tax. Dealer Inquiries invited.

In The Public Domain

By Stephen M. Leon

200 Winston Drive
Cliffside Park NJ 07010

This issue Steve covers text formatters and worms. Formatters are good, worms are bad. Actually, worms might be OK if they were a little less destructive. (Something that ate its way slowly across the screen without destroying any data or beeped the Stars and Stripes at random might be interesting.) Anyway, this column contains a good reference list of the goodies and the baddies.

While you and I sit at our terminals making the machine display what suits our whims, most of the rest of the world see only the printed page. Improving the looks of that page is a job done rather well by software in the public domain.

The definitive program to format text is Professor Ernest E. Bergmann's ROFF4. (SIG/M Volume 174 with partial update on SIG/M Volume 195.) While seemingly written for the special requirements of scientific manuscripts, ROFF allows the user to design scientific symbols or special characters, produce headers and footnotes, and produce a page that does justice to the text. The CP/M world is way behind the MAC and the PC in tabletop publishing, but ROFF is one of the gems that provides a means to close this gap.

High Resolution Graphics Printing

Another program to add printing power to a CP/M machine is the High Resolution Graphics Printing System from the Air Force Institute of Technology (SIG/M Volume 194). It comes configured for C. Itoh, Epson, and Okidata printers with driver source code for other printers.

Neither ROFF nor the Air Force program has the ease of use of the Macintosh. Moreover, you won't get output equal to the Apple Laser. What you will get with a little work is some pretty good looking copy with charts and graphs that'll make your copy distinctive.

If your needs are not quite as demanding and your skills a little weak, the libraries are still filled with programs that come in handy at printing time. If you have an Epson FX printer,

Paul Gans' FXDRV (SIG/M 225) allows a second character set in WordStar. (The same volume also contains a scientific font for the MX-80.) Your Epson or Okidata printer will produce two-column text with DBL211 (SIG/M 193).

Not satisfied with two columns, Harold McIntosh has provided the public with the ENCOL library (SIG/M 206). With it, you can do two, three, or four columns on a page, CP/M-80 or CP/M 86, and even have the ability to print two sides of a page.

On the same disk you will find TY. With it you can type any file — even binary, even squeezed. Even squeezed in a library! Professor McIntosh and his people at the Universidad Autonoma de Puebla in Puebla Mexico are probably the most productive educational assemblage in the world from the standpoint of contributing high quality software to the public domain. They deserve our thanks again and again. Printer setting programs are handy, and there are plenty of them in the library. SIG/M 172 contains CP/M 80 and 86 versions of printer setting programs for the Oki and the Epson. The code is there so you can adjust them for any printer. ANYCODE on 202 allows you to set codes too, but this time from WordStar.

The new SIG/M catalog (through Volume 246) is finally printed and a few minutes of browsing will turn up many more helpful printing programs.

The Worms Keep Popping Up

As if it isn't bad enough that the IBM PC brought forth a breed of software beggars and their demon shareware, it also brought out a new class of creep who puts worm programs on BBS systems.

Tom Neff (Compuserve 76556,2536) is trying to report on this type of software and suggests you watch out for the following programs:

DOSKNOWS.EXE — An apparent FAT killer, not to be confused with the harmless DOSKNOWS system-status utility. The *real* DOSKNOWS.EXE is 5376 bytes long. If you see something called DOSKNOWS that isn't close to

that size, sound the alarm.

EGABTR — Description says something like "improve your EGA display," but when run it deletes everything in sight and prints "Arf! Arf! Got you!"

FILER.EXE — Labeled "Great new filing system" — wiped out a 20 Meg hard disk. There may be a legit program with the same name, but be careful.

SECRET.BAS — This may pop up on a bulletin board with a note saying it doesn't seem to work, and would someone please try it; when you do, it formats your disks.

STRIPES.EXE — This one draws an American flag, but meanwhile it's busy copying the BBS password file to another file, STRIPES.BQS.

Tom's report also covers what he called "hacked" and "pirated" software. We just finished a three-day seminar with about 200 lawyers discussing computer law. It was an interesting group of sessions on everything from shrink wraps to why DRI gave in to Apple. I would guess that had we asked, the consensus of that group would have been that this nonsense about the "send me a donation crowd" thinking they own the exclusive right to the world by merely putting on a copyright notice, is just that — nonsense.

Here we have people who won't even spend their own money to commercially distribute their software trying to impose all sorts of conditions upon users. These are some of the same people who literally steal a public domain program, put a copyright notice on it after they change two lines of code, and then get upset when we just laugh at them. But don't get me wrong, there's a lot of good software that comes out as shareware.

Tom Rettig's dGENERATE contains some of the best dBASE routines around, and I gladly sent him \$15 to cover his costs. But Tom not only gives you a compiled version of a screen editor and code generator, he also gives you his code to incorporate into your own programs.

Tom Neff is providing a service with his list of worms. I think, however, he

is way off base in thinking that we have no right to alter shareware.

News From PC/BLUE

Hank Kee has been quite busy grinding out new PC/BLUE releases. Next issue we'll spend a little more time with some of the new PC/BLUE releases. For the moment, however, I just want to mention four of his new disks. Hidden on PC/BLUE 151 is CED — Command Line Editor. If you are running a PC without Super Key or one of the other commercial programs that allow you to edit the command line, CED is a must.

Chris Dunford, the author of CED, is the kind of programmer all of us should look up to. CED is a true contribution to the public domain. It provides complete command line editing, recall of previously issued commands, etc., etc. To top it off, you have a 25-page professional manual.

In the area of new language releases, we have Ron Cain's Small C for the PC on Volume 154 and Dave Betz's latest version of XLISP on Volume 146. Bob Morein's Prolog is on Volume 157. All three should be of interest to the non-appliance PC user. While we have not tried them on generic MS-DOS, they should work in that environment. CP/M users will find Small C on SIG/M Volume 224 (Z80 with Floating Point) and on SIG/M 149 (CP/M 86). XLISP for CP/M 86 is on SIG/M Volume 153 and for CP/M 80 on SIG/M 118. Gerald Edgar's EPRO (Small Prolog) is on SIG/M Volume 242 (Z80).

Software Distribution Points

I've had a number of requests for more detailed distribution information. The list runs about three pages and is included with the catalog. In addition, there are a number of commercial operations and rental libraries which make the software available at reasonable rates. Most of the commercial operations have a lot faster turnaround than the non-commercial distributors. However, the non-commercial operations usually have bulletin board systems that allow you to download. Most of the new software is on the CP/M SIG on Compuserve and very

shortly should be on the new General Electric GENIE network.

It usually takes about a month to get an order from either SIG/M or PC/BLUE. SIG/M software is distributed only in 8" format. For other formats you're probably better off with one of the commercial services. Dusch Computer Services, 405 East 6th Avenue, Roselle, NJ 07203, (201) 241-0719, promises a maximum shipping time of

two days for any 5" format. They charge \$7 per disk and donate \$1 to either SIG/M or PC/BLUE.

Apple format SIG/M software is available from the Philadelphia Area Computer Society, Apple II SIG/M Library, Box 477, Kulpville, PA 19442.

The following is a list of SIG/M (and in most cases, PC/BLUE) regional distributors:

SIG/M And PC/BLUE Regional Distributors

Dave Mitton
NECS New England Computer Society
13 Swan St
Arlington MA 02174
Voice (617) 646-3080
CBBS (617) 646-3610

Charles E. Lewis
Cleveland Digital Group
379 S. Hametown Rd
Copley OH 44321
Voice (216) 666-5982
RCIPM (216) 645-0827

William Earnest
Lehigh Valley Computer Group
RD#1 Box 830
Orefield PA 18069
Voice (215) 398-1634
CBBS (215) 398-3937

Stan Levine
CP/M Users of Washington DC
2053 N. Abingdon St
Arlington VA 22207
Voice (703) 522-1192

John Irwin
Miami Amateur Computer Group
9159 South West 77th Ave
Miami FL 33156
Voice (305) 271-4592

Charlie Wells
Atlanta Computer Society
465 Northgate Pass
Roswell GA 30075
Voice (404) 993-0366
CBBS (404) 636-6130
CBBS (404) 394-4220

Jim Mills
CACHE/CPMUG
Box C-176
323 South Franklin
Chicago IL 60606-7092
Voice (312) 469-2596
RCPM (312) 469-2597
RCPM (312) 545-8086

Dave Hardy
Technical CBBS
736 Notre Dame
Grosse Pointe MI 48230
Voice (313) 885-0506
Voice (313) 846-8000
CBBS (313) 846-6127

Fred Pfafman
Texans Computer Club
2320 Heather Hill Lane
Plano TX 75075
Voice (214) 596-5034

A.J. Lundquist
Denver CP/M SIG of Colorado
P.O. Box 633
Broomfield CO 80020-0633
Voice (303) 466-7938
CBBS (303) 465-1313

James Yuen
Aloha Computer Club
919 Luna Helu St
Kailua HA 96734
Voice (808) 262-2845

Jorge Blat
CPMUG of Puget Sound
8016 188th SW
Edmonds WA 98020
Voice (206) 771-1408
CBBS (206) 621-8665

Tim Linehan
Olympia RCPM
129 N. Decatur
Olympia WA 98502
Voice (206) 357-7400
Voice (206) 357-6300
CBBS (206) 352-7530

Sam Lay
South Orange County Computer Club
6771 Bonnie Dr
Huntington Beach CA 92647
Voice (714) 842-6588

Charlie Foster
JRT Pascal & Z Users Group
7962 Center Parkway
Sacramento CA 95823
Voice (916) 392-2789
CBBS (916) 483-8718

Trevor Marshall
Thousand Oaks RBBS
3423 Hill Canyon Ave
Thousand Oaks CA 91360
Voice (805) 492-3693
RBBS (805) 493-1495
RBBS (805) 492-5472

Fukuoka Toshio
Maeda Build. 8F
Kita-20, Higasi-1
Higasiku Sapporo Japan
Voice 01-44-36-2807

Bill Bolton
P.O. Box 357
Kenmore
QLD 4069 Australia
Voice 61-7-378-2338
RBBS 61-7-378-9530
Local (02) 378-9530

Dave Bowerman
West Coast Computer Society
P.O. Box 4031
Vancouver B.C. V6B 3Z4
Voice (604) 937-3148
CBBS (604) 937-0906

Judson Newell
4691 Dundas Street, W
Islington Ontario M9A 1A7
Voice (416) 239-2835
CBBS (416) 231-9538
(416) 231-1262

Peter Briggs
Transvaal Amateur Computer Club
P.O. Box 2513
Kempton Park 1620
South Africa

Derek Fordred
CP/M Users Group of UK
72 Mill Road
Hawley
Dartford, Kent, England DA2 7RZ
Voice (011-44) 0322-22669

Hans Stauffer
Caracus Computer Club
Apartado 66394
Caracus Venezuela 1061A
Voice (011-58) 2-987-3308

SIG/M volumes are available on 8" SSSD disks for \$6 each (\$9 foreign) directly from SIG/M, Box 97, Iselin, NJ 08830. Printed catalogs are \$3 each (\$4 foreign). Disks in a variety of formats may be obtained through the worldwide SIG/M distribution network. The distributor list is included with the printed catalog. A disk version of the catalog (Volume 00) is available for \$6. PC/BLUE volumes are \$7 each (\$10 foreign). The printed catalog is \$5. Both are available from the New York Amateur Computer Club, Box 100, Church Street Station, New York, NY 10008.



WordStar Function Key Reassignment

By Dr. Wayne Sugai

Bucknell University
Lewisburg PA 17837

The Televideo TPC-1 is currently one of the better values in transportable CP/M machines. Although it comes with the Telewrite editor, many WordStar fans will want to use their favorite poison, and with all those function keys, who can resist setting up some single keystrokes to replace the control combinations?

This simple patch routine lets you redefine 14 of the Televideo TPC-1's function key assignments within WordStar 3.0 and 3.3.

Installation

After completing the function key assignments described in FKPAT (see Figure 2), assemble with ASM or MAC. Use DDT for the patch insertion procedure.

Theory Of Operation

Using CP/M's BDOS function 9 (print string), WS30PAT (Figure 2) sets up the TPC's function (F1-F10) function keys for 14 WordStar operations (such as cursor movement, block manipulation, and disk access).

Combined with the shift and ctrl keys, the 10 function keys can generate 32 different characters.

WS30PAT modifies WordStar by placing function key assignment data in the MORPAT area (which is loaded when the program begins execution). Thus, the program simply uses MORPAT as a data-storage area for BDOS function 9, which is "directed" to MORPAT by a jump instruction at TRMINI.

This is a convenient way to redefine keys, but there is only room to redefine 14 keystrokes.

Although the routine was originally written for WordStar version 3.0, I've included the addresses for Version 3.3 in comments. You'll just need to change the TRMINI and MORPAT address locations.

Standalone

Finally, conversion of this patch routine to a standalone program requires only the code changes shown in Figure 1. This way you can redefine the full 32 characters. (The program must be run right before loading the editor.)



Figure 1 - Code Changes for Standalone Program

```
Eliminate:
    TRMINI    equ    [xxx]
    MORPAT    equ    [xxx]
                org    TRMINI
                jmp    TPCINIT
                org    MORPAT

Add:
    org      0100h    ;This becomes the first line of the
                    ; program

Change:
    tpcinit: mvi    c,9    =>    mvi    c,9

Add:
    jmp      0        ;This comes right after "call 5"
```

Figure 2 - Patch for WordStar 3.0

```
;
; WS30PAT.ASM--A WordStar 3.0 patch for redefining f1-f10 function keys
; on the Televideo TPC-1 portable computer. Since this patch is written
; for WordStar version 3.0, the addresses shown in the body of the code
; do not correspond to WordStar version 3.3. Where applicable, the proper
; addresses for WordStar version 3.3 are shown in comments.
;
; As noted below, WordStar's MORPAT auxiliary patch area has room to
; store the data strings for only 14 of the TPC-1's 32 possible function
; key assignments. Implementation of the full 32 possible reassignments
; is thus possible only "outside" WordStar, requiring the use of a stand-
; alone version of this patch, which would be executed from CP/M immedi-
; ately before WordStar
;
; -- Dr. Wayne Sugai / Bucknell University / Lewisburg, PA (8/23/85)
;
;-----
;
; A0>ddt ws.com        ; Use DDT for patch insertion
; DDT VERS. 2.2        ; DDT sign-on
; NEXT PC
; 3F00 0100            ; Address for WordStar Version 3.0
; -IWS30PAT.HEX       ; [I]nsert the .HEX file
; -R                   ; [R]ead it in
; NEXT PC
; 3F00 0100            ; [G]o to "0" (as in zero)
; -GO                  ; Save 69 for WordStar Version 3.3
; A0>save 62 wsnew.com
;
;-----
;
trmini equ    0292h    ; TRMINI is the terminal initialization area in
                    ; WordStar. When this patch is installed, the
                    ; code will be: C3E002 (jump to MORPAT). This is
                    ; for WordStar Version 3.0. Change this to 0275h
                    ; for Version 3.3.
;
;
; morpat equ    02e0h    ; MORPAT is the area in WordStar for additional
                    ; code such as this. This is for WordStar Version
                    ; 3.0. Change this to 02cbh for Version 3.3.
;
;
; org      trmini    ; Terminal initialization always comes first.
; jmp      tpcinit   ; Jump to the patch routine in MORPAT.
;
;
; org      morpat    ; Patch starts in the MORPAT area.
;
; tpcinit: mvi    c,9    ; CP/M BDOS function 9
;          lxi    d,fkpat ; Load function key assignments
;          call   5      ; BDOS entry: patch it all at once
;
;-----
;
;-----
```

```
; FKPAT contains the specific TPC-1 function key assignments and must
; follow this convention:
```

```
1bh,7ch,[x],00,[y1],[y2],19h
```

```
; where:
```

```
[x] => function key number (the following table of TPC-1 function
; key values is from the Televideo TPC-1 System Reference
; Manual)
```

Function Key Values

Key	Unshifted	Shifted	Ctl- Unshifted	Ctl- Shifted
f1	31h	30h	3bh	46h
f2	32h	3dh	47h	40h
f3	33h	3eh	48h	4dh
f4	34h	3fh	49h	4eh
f5	35h	40h	4ah	4fh
f6	36h	41h	4bh	50h
f7	37h	42h		
f8	38h	43h		
f9	39h	44h		
f10	3ah	45h		

```
[y1],[y2] => the specific key assignment.
```

```
NOTE--[y2] is needed only if the command string requires two
; keys (e.g., ^Q,R). Otherwise, only [y1] will be needed,
; and the blank [y2] entry should be deleted from the
; data string.
```

```
; Example:
```

```
db 1bh,7ch,3eh,00h,0bh,44h,19h ; <Shift>f3 => ^XD (Save and
; exit to main menu)
```

```
; where:
```

```
1bh,7ch => FK redefinition lead-in ESCape sequence
3eh => FK value--<Shift>f3
00h => Empty data byte (must be included)
0bh => "K" (first byte of two-key WordStar command string)
44h => "D" (second byte of two-key WordStar command string)
19h => FK redefinition termination byte
```

```
NOTE---There is room in MORPAT for only 14 function key assign-
; ments. The following table shows 32 assignments just for
; convenience. Don't exceed this limit of 14 or you will
; overrun the available MORPAT area!
```

```
The 14 assignments shown below are arbitrarily selected for use
; as an example of WordStar functions and should be redefined to
; suit the user.
```

```
fkpat db 1bh,7ch,31h,00h,01h,19h ; f1 => ^A (Word left)
db 1bh,7ch,32h,00h,06h,19h ; f2 => ^F (Word right)
db 1bh,7ch,33h,00h,11h,53h,19h ; f3 => ^QS (Line left)
db 1bh,7ch,34h,00h,11h,44h,19h ; f4 => ^QD (Line right)
db 1bh,7ch,35h,00h,1ah,19h ; f5 => ^Z (Line up)
db 1bh,7ch,36h,00h,17h,19h ; f6 => ^W (Line down)
db 1bh,7ch,37h,00h,03h,19h ; f7 => ^C (Page forward)
db 1bh,7ch,38h,00h,12h,19h ; f8 => ^R (Page backward)
db 1bh,7ch,39h,00h,11h,52h,19h ; f9 => ^QR (Beginning of file)
db 1bh,7ch,3ah,00h,11h,43h,19h ; f10 => ^QC (End of file)
db 1bh,7ch,3ch,00h,0bh,51h,19h ; <Shift>f1 => ^TQ (Abandon
; edit)
db 1bh,7ch,3dh,00h,0bh,53h,19h ; <Shift>f2 => ^TS (Save and
; return to editing)
db 1bh,7ch,3eh,00h,0bh,44h,19h ; <Shift>f3 => ^TD (Save and
; exit to main menu)
db 1bh,7ch,3fh,00h,0bh,58h,19h ; <Shift>f4 => ^TI (Save and
; exit to system)
```

```
This is the limit of 14 function key assignments. Proceed only if you
; have left some of the above assignments unused (be sure to leave the
; ";" intact on the unused assignments).
```

```
db 1bh,7ch,40h,00h, h, h,19h ; <Shift>f5 =>
db 1bh,7ch,41h,00h, h, h,19h ; <Shift>f6 =>
db 1bh,7ch,42h,00h, h, h,19h ; <Shift>f7 =>
db 1bh,7ch,43h,00h, h, h,19h ; <Shift>f8 =>
db 1bh,7ch,44h,00h, h, h,19h ; <Shift>f9 =>
db 1bh,7ch,45h,00h, h, h,19h ; <Shift>f10 =>
db 1bh,7ch,3bh,00h, h, h,19h ; <Ctl-Unshifted>f1 =>
db 1bh,7ch,47h,00h, h, h,19h ; <Ctl-Unshifted>f2 =>
db 1bh,7ch,48h,00h, h, h,19h ; <Ctl-Unshifted>f3 =>
db 1bh,7ch,49h,00h, h, h,19h ; <Ctl-Unshifted>f4 =>
db 1bh,7ch,4ah,00h, h, h,19h ; <Ctl-Unshifted>f5 =>
db 1bh,7ch,4bh,00h, h, h,19h ; <Ctl-Unshifted>f6 =>
db 1bh,7ch,4ch,00h, h, h,19h ; <Ctl-Shift>f1 =>
db 1bh,7ch,4cb,00h, h, h,19h ; <Ctl-Shift>f2 =>
db 1bh,7ch,4db,00h, h, h,19h ; <Ctl-Shift>f3 =>
db 1bh,7ch,4eh,00h, h, h,19h ; <Ctl-Shift>f4 =>
db 1bh,7ch,4fh,00h, h, h,19h ; <Ctl-Shift>f5 =>
db 1bh,7ch,50h,00h, h, h,19h ; <Ctl-Shift>f6 =>
```

```
ret ; This RET statement must be here!
```

```
end
```

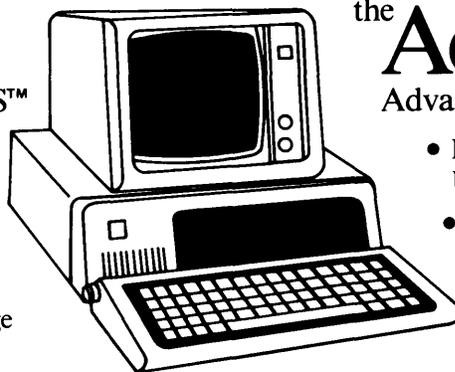
End of Listing

CP/EM™

CP/M™ 80 Emulation for MS-DOS™

- Z80 Emulation
- CP/M Version 2.2 Compatible
- Accelerator Compatible
- Serial Program Utilities Included
- FAST-Written in Assembly Language

\$79.95



the Accelerator™

Advanced CPU for the IBM PC

- Faster Program Execution
Up to 30%, 50% and more!
- Full 8088/6 Compatibility
- Extended Instruction Set
- 8 Bit and 16 Bit Bus Versions Available

\$49.95*

*5 MHZ for the IBM PC

The Intelligent Computer Users Group
P.O. Box 10118 • Rochester, NY 14610
(716) 425-2519

IBM is a trademark of International Business Machines.
MS-DOS is a trademark of Microsoft Corporation.
CP/M is a trademark of Digital Research Incorporated.



Dealer inquiries welcome.

EARTH LAUNCHES NEW STARS

EARTH COMPUTERS launches two of the "Hottest" new stars in the S-100 Universe. Both the **TURBOMASTER 8™** and **TURBOSLAVE I™** are Star performers, featuring high speed Z-80H CPUs.

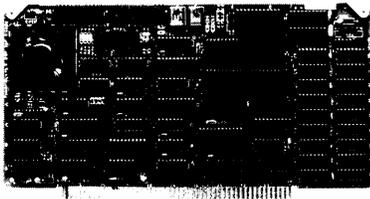
EARTH's new stars are fully compatible with the Multi-user TurboDOS operating system, and will operate in most S-100 systems, including pre-IEEE 696 systems.

TURBOMASTER 8

This outstanding new 8-Bit Single Board Computer offers features that are out of this world:

- On-board ST-506 Winchester Controller
- TurboDOS, CP/M, MP/M compatible
- 5-1/4" and 8" Floppy Controller
- Up to 256KB of memory
- 8 MHz, Z-80H CPU
- 2 Serial ports
- 1 Parallel port

TURBOSLAVE I

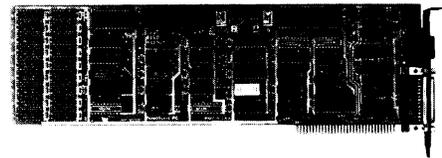


The perfect companion to the TURBOMASTER 8 or other 8/16-Bit Master processor. This high speed slave utilizes an 8 MHz Z-80H CPU and offers extensive on-board diagnostics... an industry exclusive.

- No paddle boards
- S-100, IEEE 696 compatible
- 128KB of RAM
- 2 RS-232 ports, 50-38.4K Baud
- FIFO communications
- Data transfers to 1 MB

EARTH COMPUTERS also manufactures a growing line of PC compatible stellar performers such as:

- **TURBOSLAVE PC™**—an 8 MHz



Z-80 single board slave processor that runs CP/M applications on a PC and is compatible with the TurboDOS multi-user operating system.

- **TURBOACCEL 286™**—a high performance 80286 accelerator that boosts PC performance up to five times.
- **EARTHNET PC™** and **EARTHNET S-100**, the low cost, ARCNET-compatible way to tie PC and S-100 systems together.

To put these stars to work for you, call or write EARTH COMPUTERS. **BE SURE TO ASK ABOUT HOW YOU CAN WIN A FREE Z-80 CO-PROCESSOR BOARD.**



EARTH COMPUTERS

"Building Blocks For The Super Micro"

P.O. Box 8067, Fountain Valley, CA 92728 • TELEX: 910 997 6120 EARTH FV • PHONE: (714)964-5784

Comdex (The Tale Of One Large City)

By David Thompson

Sandy and I took 100 copies of issue #27 to Comdex — two large boxes full of magazines. I returned with those same two boxes, both full of press releases, product announcements, and weeples (small fuzzy animals that frequent major computer shows).

The weeples were cute and innocent (Jennifer and Erin collect them); the hype in the releases and announcements wasn't nearly as cute or innocent. But then what did we expect from the Hard-Shaftner-&-Sharks set?

Las Vegas is an interesting place to hold a computer show. First, the hotels don't particularly like us computer folk because we don't gamble. But I don't think the hotels realize how much we computer types have in common with gamblers.

I noticed the people who were gambling displayed much the same intensity that I feel when I'm writing or programming. I can sit down at the computer after supper and in almost no time at all it's morning.

One of the attendees mentioned that the casinos have no windows because they don't want the gamblers to have reference to day or night. If they don't know it's morning then they won't know they've already been up all night. (Watch out for software houses that have windowless work areas.)

No CP/M-80

Comdex was more interesting this year. There weren't any announcements of new CP/M software but that guarantees there won't be any fewer announcements next year.

So, practically everything was PC-DOS and the theme this year seemed to be "more of the same." Fortunately, "the same" was faster, more powerful, cheaper, and there were even some new bells and whistles.

Text Editors

Those with text editors were adding features: spelling, thesaurus, math calculations, graphics, support for new printers (including the lasers), and even translation of abbreviations (wld drv me nts).

PC Write, the shareware folks, were

there selling their latest version at \$10 per disk — you can register for an additional \$75. With registration you get certificates for two updates, support, and a printed manual. Bob Wallace has new versions coming that should shame WordStar. If you want the latest version (with manual on disk) which you are welcome to share with everyone, send \$10 to:

Quicksoft
219 First N #224
Seattle WA 98109
206-282-0452

The most unusual text editor I saw was Multi-lingual Scribe. It lets you edit, display, and print English (including most European characters), Hebrew, Greek, Arabic, and Russian. And you can intermix all these languages at will. A single keystroke switches between them. Characters are written on the screen from left to right or from right to left depending on the language. This \$349.95 package even lets you create your own characters, and it includes keyboard templates. It's available from:

Gamma Productions, Inc
817 10th St, Suite 102
Santa Monica CA 90403
213-451-9507

Outline Processor

I didn't see this outline processor run, but Sandy did and she was very excited about it. Called Eazy, it appears to have more features than ThinkTank, Ready, and Maxthink, yet it's only \$49.95. It's supposed to be compatible with standard text editors, have built-in editing functions, on-disk tutorials, full outline display, and more. Contact them for more info:

Computer Personalities
303 Hegenberger Rd, Suite 405
Oakland CA 94621
415-568-4821

Graphics

Graphics packages (computer aided

design style) were really showing up in force. Though Auto-Cad has become the standard, its high price (around \$2500) has encouraged a whole slew of competitors. Some of them were little more than fancy painting programs, while others included libraries for architecture or electronics and provided ways to develop and store your own graphics symbols. Prices ran from \$250 to \$1000. I'll try to have a more thorough report for you soon.

Of all the reasons to get into the PC environment, CAD has to be one of the most important.

Plotters

There were a lot of plotters at the show, but most were busy showing off someone's CAD package. There was one plotter, however, that was busy showing itself off.

It's called Penman and it looks more like an errant rodent than a plotter. The paper is fastened down on a flat surface and the mouse-like plotter scurries about marking here and there. This little three-pen robot is accurate to one line width, emulates a standard HP plotter, and produces very large drawings for a very small price. There are two models.

Penman II-D does D size (24" by 36") plots and will retail for \$575! Penman II-E does E size (34" by 44") plots and will retail for \$725!

The units' one disadvantage that I could see is that they are slow. A standard plotter costs 10 times as much but will create plots in 1/10 the time. Expect to see the Penman on the market by May 1986. It will be available from:

Penman Products Corp
801 West Street Road
Feasterville PA 19047
215-357-8988

Database Packages

Ashton-Tate announced a new version of dBASE III that does faster indexing (rumor has it that they

(continued next page)

changed the structure of their index files). Both dBASE III and Clipper (a dBASE III compatible compiler) use SoftLok, a reportedly fussy copy protection scheme.

WordTech Systems now has a \$169 dBASE III look-alike (interpreter) in addition to their \$750 dBIII compiler. I understand that the WordTech software is no longer copy protected.

WordTech Systems
PO Box 1747
Orinda CA 94563
415-254-0900

Rumor has it that there's another dBASE III style interpreter under development. Currently under beta test, it'll probably sell for \$99. It should be available sometime this Spring. More information as I receive it. (If I owned Ashton-Tate stock right now, I think I'd dump it.)

There were also lots of other database packages. Some were cheap, others, like RBASE 5000 for multi-user PCs, were \$1500.

Scanners Make An Appearance

There's been a lot of activity during the past year in scanners. The problem has been getting all that stored up hard copy into a computer, and with compact disks on their way, the need to find a non-manual way to convert all this data is getting pretty acute (microfiche, look out).

The units I saw scanned at about 200 dots per inch and output the scan data to the computer. It was up to the computer to translate that information into vector information or text.

The computer needs graphics in vector form if it is to be manipulated by any of the CAD packages. It also needs to be able to translate characters into ASCII text.

The scanner designers are already working on vector outputs and ASCII recognition (the good ones are going to be using artificial intelligence in their character recognition schemes). I expect to see vector outputs by next year, but ASCII will probably take another year. (One outfit is doing a bit of both already, but the system is very expensive.)

Drivetek Strikes Again

Kodak was showing off its latest drive technology. I was interested until I saw the 2.6 meg floppy drive that they were featuring. Wow! I'll bet Kaypro would love to sell Kodak all the Driveteks they have sitting in the warehouse (both Kaypro and Kodak bought rights to manufacture them). Kaypro even bought a bunch of the

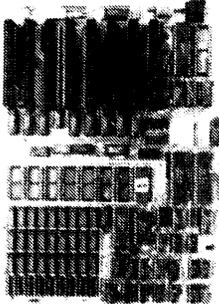
drives for its Robie. Kaypro has written off the whole lot.

Other than the fact that they couldn't find disks that worked reliably with the drives (the disk manufacturers saw things a bit differently), the drives were perfect.

Another Day, Another Drive

Anyway, if you wanted a new (pick

NOW AT THE SBC MART

<p style="text-align: center;">POWER SUPPLY</p>  <ul style="list-style-type: none"> - 135 Watts - side switch - top quality - one year warranty - standard cables for 4 disk drives • Model PS-135 power supply regularly \$129, now only \$95 	<p style="text-align: center;">HARDWARE SPECIALS</p> <ul style="list-style-type: none"> ★ keyboard 5150 style regularly \$125, Model KB-3 only \$90 ★ keyboard 5151 style regularly \$195, Model KB-3 only \$129 • memory nine 150ns DRAMS Set ID-64 64K reg \$8, now \$5; Set ID-264 264K reg \$55, now \$39 ★ no better DS/DD drives anywhere, Sanyo 48tpi F-4 \$99; F-9 96tpi \$129
<p style="text-align: center;">CASES</p>  <ul style="list-style-type: none"> - high quality - side-switch - flip-top • reg \$95, now \$69 	<p style="text-align: center;">MOTHERBOARDS</p>  <ul style="list-style-type: none"> - full IBM PC/XT compatibility - 8 I/O slots - runs IBM's PC-DOS 3.1 - BASIC interpreter available - great foundation for business or personal system - 1 year warranty
<p style="text-align: center;">HARD DISKS</p>  <ul style="list-style-type: none"> - complete 10MB Kit - special, only \$450 <ul style="list-style-type: none"> • complete internal 1/2-high 10MB kit, incl. controller and cables, special \$450 • complete 10MB kit with low-power 3-1/2" half-high drive, fits 5-1/4" slots, \$485 ★ 20MB half-high drive HD-2 now \$445 	<ul style="list-style-type: none"> ★ Model MB-1 256K motherboard with OK regularly \$295, now \$220 ★ Model MB-3 640K motherboard with OK regularly \$399 now \$290 • Model MB-2 640K motherboard, 4.77 and 6.67MHz clock, w/256K Reg \$449, now \$399; w/OK Reg \$409, now \$369, this is a fast board
<p>LAST MINUTE SPECIALS</p> <ul style="list-style-type: none"> ★ QMTI hard disk controller, up to 2 drives, with cables, super fast! \$175 - floppy disk controller, any mix 1 to 4 reg 48tpi or 96tpi quad drives \$74 ★ Hercules equiv monochrome graphics card, runs 123, etc w/printer port \$145 - multifunction 384K RAM, clk/cal & S, P, game ports w/OK \$145, w/384K \$170 	
<p>COMPLETE IBM PC COMPATIBLE SYSTEM</p> <ul style="list-style-type: none"> ★ 256K RAM, keyboard, 48/96 floppy controller, 360K floppy, mono or color card, case, power supply, w/IBM PC DOS 3.1 \$839 w/o IBM PC-DOS ONLY \$799 	
<p>Visa/MC/AmEx ORDERS: (619) 375-5744</p> <p>The SBC Mart, 821 Commercial St., P.O. Box 1296, Ridgcrest, CA 93555</p>	
<p>Shipping and handling: drives/motherboards/cases/power supplies \$4.50 ea, software \$2.50 ea, cards \$2 ea, keyboards \$4. CA residents add 6% sales tax.</p>	<p style="text-align: center;">SBC PLEDGE</p> <ul style="list-style-type: none"> - service after sale - technical support - low prices - one-year warranty

one) — hard, floppy, cartridge, laser, write-once, read only, read-write — drive, all you had to do was walk 50 feet.

There were 6 meg floppies, 80 meg 3 1/2" winchesters, drives on cards, encyclopaedias on disks — you name it, they were there. They had bits packed so tightly inside some of those little tiny things that it's a wonder they

could get them all in and out without bruising them.

Desktop Publishing

The decline in price of the 300 dot per inch laser printers has created a new market for printer driver software that justifies, paginates, columnizes, and holds graphics. When you put all this together you have desktop pub-

lishing. All you need is a computer, a page makeup package, some way to create graphics, and a printer — and you can put together very adequate newsletters (certainly much better than the standard letter quality).

According to the experts, we will need to reach 500 dots per inch characters before the laser outputs will be indistinguishable from the outputs of standard typesetters. (Digital typesetters run between 1000 and 1500 dots per inch but the printing process doesn't reproduce more than about 500 dots per inch.)

One of the problems with the higher resolutions is that the amount of data is the square of the resolution, so doubling the dots per inch quadruples the memory requirements (at least within the laser printer).

The Press

I heard a lot of voices of frustration from the writers and editors who attended Comdex. The frustration was aimed primarily at the show promoters (no terminals or special phones so they could file their stories). In addition, the promoters strongly discouraged companies from making such tools available to the press.

"We should protest!" "They're ripping us all on the hotels and now they're keeping us from filing our stories." "Maybe we could form a committee." "Let them know that we won't come to their show if they don't treat us fairly." "They used to be very helpful." "They needed us then."

The computer press has gotten so close to the industry (which includes the shows) that it's taken for granted. (Oh, we get in free, and we get a room to hide in when we're about to throw up from the: "Oh you're from the press, you have to look at...")

In a way the reduction of special goodies is good. Perks leave the perks beholden to perks. But the relationships that have developed between the writers and those they are writing about still bind the writers' hands. It happens in all of journalism. You get inside someone's skin and you start thinking the way they do. It's only logical.



Advanced Concepts "Mini-Winnie" HARD DISK SYSTEMS For your Z-80 Computer!

Our EXTERNAL systems install in seconds! Includes an attractive vertical cabinet with integral P/S and fan, 3 foot cables. Everything necessary to plug in and go!

Our INTERNAL systems do not displace a floppy drive, do not require a power supply upgrade, use the highest technology winchesters available. No compromises.

COMPLETE SUBSYSTEMS

Plug in and run! Includes software, documents, cables, brackets as req'd.

	\$665. 10mb External	\$790. 20mb External	\$790. 10mb Internal	\$845. 20mb Internal
Kaypro (1,2,4,10)	✓	✓	✓	✓
Big Board I,II	✓	✓		
Xerox 820-I,II	✓	✓		
Micro Mint SB-180	Add \$10.	Add \$10		
Gemini Zorba	✓	✓	Add \$20.	Add \$20.
Osborne I	✓	✓	✓	✓
Sanyo MBC-1150/1160	✓	✓	*	*
AVL Eagle	✓	✓	*	*
Alspa II	✓	✓		
	(* Under Development)			
SPECIALS . . . while stock lasts . . .				
5mb External \$465. TM602 or equiv. (less P/S enclosure \$395.)				
10mb External \$495. CM5616 or equiv. (less P/S enclosure \$425.)				
15mb External \$535. CM5619 or equiv. (less P/S enclosure \$465.)				
(call for current specials)				

Drives

20mb 3.5"	\$495.	Drive cable sets 1.5' - 10.95	3' (std.) - 11.95	6' - 14.95
10mb 3.5"	\$439.	Kaypro interface cables (externals) 16.67		
20mb 5.25" HH	\$439.	Host extension cable, 1 foot 18.95		
10mb 5.25" HH	\$279.	"Y" power cable 9.50		
10mb FH SPECIAL (call)	under \$200.	Document package (incl. with system) 8.00		
5mb FH SPECIAL (call)	under \$120.	Enclosure 5.25" FH, with P/S - 69.00		
		without P/S - 49.00		

Accessories

The key to the Mini-Winnie system is the INSTALL program, which links your drive and your computer, the way YOU want it. Extensive menu options allow selection of any ST-506 compatible drive, as 1 to 4 logical drives within CP/M at any drive letter. Diagnostic and format capability provided to match the installation. Distributed in SOURCE form.

We have club, user group, OEM/Dealer programs. Call for quantity schedule. If for any reason you are not satisfied, return for full refund. Special drives warranted for 30 days. Info and manual \$8.00 credited with system purchase. Specials subject to availability.

ORDER INFO: Call (305) 482-7302. Terms: UPS cash COD, check or M.O. Please allow 4 weeks for delivery. Fla. add 5% tax. Shipping extra. Please specify desired floppy format.



Advanced Concepts Engineering & Consulting
8926 S.W. 17 St., Boca Raton, FL 33433

\$49.95 FMT \$49.95

Text Formatting System

FMT provides most of the features of the high-priced Text Formatters at our inexpensive price – and it's easier to use, too! Note the features below:

- Easily configured to your printer. Configuration files for 20+ printer models are provided or generate your own.
- FMT gets the most from your printers by taking advantage of their special features, including condensed, double width, enhanced, double print, italics, elite, letter quality, multiple fonts, etc.
- Multiple modes and combinations of modes can be used on the same line or even in the same word.
- FMT works with your favorite editor!
- FMT uses meaningful mnemonic commands in the style of SCRIPT or ROFF (each command appears on its own input line), including commands for the various printing modes.
- No embedded control codes - you don't have to remember those strange escape/control sequences.
- FMT runs at the maximum speed your printer allows for each printing mode - graphics mode is not required.
- Standard formatting features provided, including headers and footers, automatic page numbering, text justification, tabs for table generation, and embedded files up to TEN deep.
- FMT automatically builds Table of Contents, List of Figures, and three level alphabetized Index.
- Detailed 100+ page manual profusely illustrated with examples.
- Works equally well with IBM-PC, TI-PC, IBM clones and look-alikes (PC-DOS/MS-DOS 128k). Also works with CP/M 8080 and Z80 systems with 64k.
- \$49.95 plus \$2.00 shipping and handling.

Specify system.

VISA and Master Card Accepted
Dealer Inquiries Welcome

TINY TEK, INC.

Route 1, Box 795
Quinlan, Texas 75474
(214) 447-3025



PRICE REDUCTION:
\$149.95

Eco-C Compiler

Release 3.4

We think Rel. 3.4 of the Eco-C Compiler is the fastest full C available for the Z80 environment. Consider the evidence:

Benchmarks*
(Seconds)

Benchmark	Eco-C	Aztec	Q/C
Seive	29	33	40
Fib	75	125	99
Deref	19	CNC	31
Matmult	42	115	N/A

*Times courtesy of Dr. David Clark
CNC - Could Not Compile
N/A - Does not support floating point

We've expanded the library to 120 functions, improved the manual and added new compile-time switches. The price is \$149.95 and includes the SLR Systems Assembler-Linker. A savings of \$145.00 over the old price!

Call today
1-800-952-0472 (orders only)
1-317-255-6476 (information)



ECOSOFT, INC. (317) 255-6476
6413 N. College Ave. • Indianapolis, Indiana 46220



C SOURCE CODE

XLT MACRO PROCESSOR w/source 95.00

XLT can be used as a preprocessor, text translator, or simple cross-assembler, depending on your macro definitions. Comes with manual, C source, macro files for Z80 assembler preprocessor, 8748 cross assembler. CP/M or PC versions.

PROGRAMMER'S TOOLS w/source 45.00

Full-blown programs you can use as-is or mine for programming gold - includes listing formatter, file encryption/decryption, easy-to-use windowing package, 12 programs in all, with manual. CP/M version now, PC version soon.

Q/C 4.0 CP/M C COMPILER w/source NOW 95.00

Will be \$195 after 3/31/86. Fast, full featured compiler from the Code Works includes longs, single-precision floats, excellent manual.

QC88 C COMPILER FOR PC's w/source NOW 125.00

Will be \$195 after 3/31/86. As above, but runs on PC clones.

QCX C CROSS-COMPILER w/source NOW 295.00

Will be \$495 after 3/31/86. Runs on PC, generates code fit to run on Z80 CP/M or ROM base controller system.

• Tony Ozrelic's L.A. Software is now KYSO Technology •

TO ORDER: Specify 8" SSSD, Kaypro CP/M, or PC version. Add \$5.00 to total for shipping and handling. Personal check or M.O. are OK. Sorry, no COD's, PO's, or plastic money. 30-day money-back guarantee. Most orders shipped 1st class within 5 working days.

KNOCK YOUR SOCKS OFF



• P.O. Box 5246 • Bend, Oregon 97708 • (503) 389-3452 •

C'ing Clearly

By Ron Miller

1157 Ellison Dr.
Pensacola FL 32503

There has always been a little animosity between Pascal fanatics and ardent C'ers. Ron takes a look at the trade-offs made by both languages so you can make up your own mind.

Just last week, while flipping through computer magazines to drool over prices of 30 Meg internal hard disks, I stumbled across three separate articles which rejected C as the language to learn because it is "notoriously cryptic, powerful but obscure, and not self-documenting like Pascal."

My defensive first reaction was to note that any language, natural or computer, defeats anyone who has not internalized its conventions. Gosh, I remember my experiences with Latin. Surely no one, I mused, ever really quarreled with his parents using the third conjugation. I had visions of Rome falling because somebody couldn't remember the dative plural of "Visigoth" in a moment of panic.

Only later, while shaking my head over foreign students struggling with the mysteries of English word order and the perversities of our prepositions, did I recognize that "easy" and "natural" describe the state of the user and not the language itself.

When Is C Obscure?

Still, I've slogged through enough of other folks' listings to know that the very power of C code can lead to a mare's nest of control structures and variables which even the best intentioned comments cannot untangle. Nothing as unspeakable as a long BASIC program full of gosubs and gotos, but something approximating the fine print in an insurance policy. You know something is being said, and at a rather high rate, but what, exactly?

Since I have written acres of C code in the last year or two, I decided to revisit some of my utilities and library functions to see what code could be followed right off the bat, what couldn't, and what could be learned from the stylistic differences between the two. I also pulled out some of my Pascal listings for comparison.

After suppressing the temptation to recode rather than go on with my speed reading, I came to the conclusion that C is no more or no less "self-documenting" than Pascal. Choose your variable names well and they will tell you what's going on.

If many implementations limit names to seven characters, while Pascal implementations often allow dozens, the fact is that limited line space and the sheer irritations of typing keep the programmer from using more than seven anyway, and the distinction between lower and upper cases in C is no less handy for labeling than a few extra spaces.

The greatest virtue in the area of variable-naming is not wordiness but the force of habit itself. I noticed that after my earliest efforts I settled down to the same old labels for the same old techniques. General index variables are — in order — 'i,' 'j,' and 'k.' General character storage is in 'c' and 'q.' All characters used as booleans end in capital 'Q.' Utility character pointers are "charptr." File handles are "fold," "fnew," "fin," "fout," and — if just one — "fdes." Nothing very creative here, but nowadays when I run into a "runQ," I can be pretty sure that somewhere I've tested whether something was run or not and have stored 1 or 0 in the variable.

Line- Vs. Parenthesis-Oriented Languages

The obvious difference between Pascal and C is that Pascal is a line-oriented language, while C is parenthesis-oriented. In grammatical terms, it's coordination versus subordination.

Reading Pascal is like slicing bread — one line, one operation, with the exception of the chaining that goes with multiple "if...then" syntax. This stratification leads to open, uncrowded, perhaps a bit "gassy" code. The convention of giving "BEGIN" and "END" their own lines further decompresses the code, yielding a definite clarity in the individual expressions and an attendant vagueness in the overall relationships between. The subroutines tend to dribble themselves down the page in a manner that

recalls BASIC at times.

C, on the other hand, works outward from nested elements within the line; it depends on a carefully defined order of operations to permit keeping fundamental routines together in single visual cluster.

Reading C is like opening Chinese boxes, one shell after another. A good C routine is like a good English sentence: sinewy, pointed, economical, and just a little surprising. A fine turn of C, like a fine turn of phrase, often takes time to unfold itself fully to the reader.

The gain over Pascal is one of elegance and efficiency. The price paid is a decrease in the immediacy of the individual line. Hemingway's sentences are easier to follow than Henry James's, but there's lots less going on in them.

To observe these trade offs in action, consider a routine that gets a single character from a file and lists it on the screen if the character is printable, and substitutes a dot if it's not. In Pascal, this routine would be:

```
read(_file,c);
if (c>=' ') and (c<='~')
then write(c)
else write('.');
```

First one step, then an "if" for the next. In C this might be written *a la* Niklaus Wirth:

```
c=getc(_file);
if( c>=' ' && c<='~' )
putchar(c); else putchar('.');
```

but a C aficionado is much more likely to nest the character acquisition and use the ternary operator:

```
putchar( (c=getc(_file)) >= ' '
&& c <= '~' ? c : '.');
```

Notice the inside-out operation: first evaluate the inmost parentheses, so that c is loaded. Then calculate the boolean and return the character from the appropriate alternative. Then send it to the screen. If the order of operations were not meticulously prescribed by the language, the whole thing would trip over itself.

(continued next page)

Though the C routine would undoubtedly execute faster, most of us will prefer the readability of the Pascal. If it runs just once, who cares if you gain a millisecond or less? But suppose you wanted to print out the file character by character until you reached an ASCII 4? In Pascal that would be:

```
repeat
  read(_file,c)
  if ord(c) <> 4 then
    if (c>=' ') and (c<='~')
      then write(c)
      else write(' ');
until ord(c) = 4;
```

The code is getting gangly. C improves it thus:

```
while( (c=getc(_file)) != 4 )
  putchar( c >= ' '
    && c <= '~' ? c : '.');
```

Here, I think, C syntax is ultimately more readable. Though the single line may not be completely clear at first glance, at least the whole operational unit is together.

Assuming fluency in C, the code reads the way we first conceived the process: Get the character and test it. If it passes the test, find out what sort

of character it is and list it in the appropriate format. Pascal syntax forces two tests and two write statements and must be traced out by some sort of mental flowchart.

Documentation Through Function Calls

The C loop is more readable if you use an appropriate name for the function. The standard C library contains the function "isprint()," of obvious meaning. Or you can write it yourself:

```
isprint(c)
char c;
{ return c>= ' ' && c<='~'; }
```

With that device at hand, you get:

```
while( (c=getc(_file)) != 4 )
  putchar( isprint(c) ? c : '.');
```

Now that's more like it. In fact, I submit that self-documentation in C is largely a matter of using descriptively-named function calls.

No doubt, the creation of new functions can spread code all over the place, decreasing readability. For that reason, I'd avoid writing function "char_dot()" which could be stuck inside the parentheses of putchar to handle both tasks. The reader would have to go find "char_dot" in the listing, because its operations would scarcely be intuitive.

Again, it's a trade off. If the subroutine is repeated often enough or if it's involved enough, then a function can probably be justified, if a good name can be found. However, if the next peruser of the code will have to page up or down through the code to find out what in the devil is going on, the simplification has been bought at too high a price.

This, of course, does not touch upon the larger matter of structuring the overall flow of the program itself. For anyone who bothers to read a C or a Pascal column, such issues have been settled long ago. And it helps to offer occasional comments, though I believe comments are far more valuable in labeling large program blocks than in unraveling the meaning of individual lines and routines.

Cleaning Things Up

Perhaps these issues can best be illustrated with a more practical example. The other day I was tinkering with routines that change individual bytes in a sector of memory as part of a larger disk/memory/file utility. As part of the design I had written two major screen functions, one to list a sector ddt-fashion, with hex codes on the left and print characters on the right; and one to edit this array by

Figure 1 - Two Major Screen Functions

```
#define XDISPL 1 /*displacements to use in gotoxy to overlay display */
#define YDISPL 4
#define CHARDISPL 54

struct SEC { /*sector mapped as 8X16 array */
  char byte[0x08][0x10];
};
/*****/
display(s) /*display sector, ddt-style */
{
  struct SEC *s;
  char i,j,q;

  for(i=0;i<0x08;i++){
    printf("\n%0: ",i); /* number rows */
    /* & char with 0xff to defeat sign extension on chars */
    for(j=0;j<0x10;j++) printf("%02x%c",s->byte[i][j] & 0xff,
      (j == 7) ? '-' : ' '); /* hyphen between bytes 7 & 8 */
    puts("- ");
    for(j=0;j<0x10;j++) putchar( isprint(q=s->byte[i][j]) ? q : '.');
  } /* dot if not printable */
  putchar('\n');
/*****/
change(s) /*move around and edit sector bytes */
{
  struct SEC *s; /* sector address passed to function */
  char q,l,item,charQ;

  display(s);
  for(charQ=item=0;){ /*start at upper lefthand character */
    if(charQ) gotoxy(item/0x10+XDISPL,item%0x10+CHARDISPL); /* which display? */
    else gotoxy(item/0x10+XDISPL,3*(item%0x10)+YDISPL);
    switch(q=grabchar()){
      case 0x1b: charQ = lcharQ; /* toggle cursor between hex and chars*/
        break;
      case 0xd: return; /* pop out on <CR> */
      case 0x8: item--;
        break;
      case 0xa: item += 0x10;
        break;
      case 0xb: item -= 0x10;
        break;
      case 0xc: item++;
        break;
      default: /* If char, change it. If hex, test for hex digits */
        if(charQ) s->byte[item/0x10][item%0x10] = q;
        else if(isxdigit(q) && isxdigit(l=grabchar()))
          s->byte[item/0x10][item%0x10] = hexit(q,l);
        /* now refresh the char indication, whether altered or not */
        gotoxy(item/0x10+XDISPL,item%0x10+CHARDISPL);
        putchar( isprint(q=s->byte[item/0x10][item%0x10]) ? q : '.');
        gotoxy(item/0x10+XDISPL,3*(item%0x10)+YDISPL);
        printf("%02x%c",s->byte[item/0x10][item%0x10] & 0xff,
          (item%0x10 == 7) ? '-' : ' ');
        item++;
        break;
    }
    item &= 0x7f; /* loop around if item > 0x7f or item < 0 */
  }
/*****/
hexit(hi,lo) /* convert two hex chars to integer */
char hi,lo; /*e.g.,'f','f' --> 255 */
{
  hi=toupper(hi);
  lo=toupper(lo);
  return 0x10*( (hi<'A') ? hi-'0' : 10+hi-'A') +
    ( (lo<'A') ? lo-'0' : 10+lo-'A');
}
/*****/
grabchar()
{
  return bdos(1,0); /* direct character fetch; no buffering */
}
/*****/
isxdigit(c) /* is digit possible as part of hex #? */
char c;
{
  return (c>='0' && c<='9') || (c>='A' && c<='F') || (c>='a' && c<='f') ;
}
/*****/
gotoxy(x,y)
int x,y;
{ printf("\33-%c%c",x,y);
  } /* For Kaypro. A more complex interrupt 10h call for IBM PC. */
```

moving about the screen with a cursor. I wrote the code in Figure 1.

The next day I glanced at the "change" function and knew that my "default" section would be opaque in two months. The array elements fill up the line so much that the logic is obscured. Moreover, the gotoxy() expressions compound the mess. Now take a look at Figure 2. By inventing two rather self-documenting functions and by employing a character pointer to access the array, I made things more legible.

I don't know about you, but I can read that, and can read it better without all those Chatty-Cathy remarks. If your code needs line-by-line commentary, recode it and find ways to make the language explain itself.

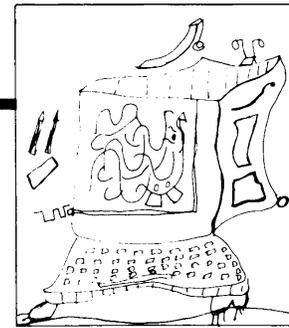
Pascal enthusiasts ought to try translating these two routines into their favorite tongue. My guess is that it will take twice as many lines and be so strung out that it will have to be parsed by a mental flowchart. Who says C is innately obscure?

Choosing a First Language

Which is not to say that I would advise anyone to take on C as his first structured language. As long as Turbo Pascal is around, the novice has almost interpreter-speed interaction with a compiler which will catch most of the silly errors that a C compiler will blissfully ignore.

In the end, I probably compose three times more Pascal than C code, because in meat-and-potatoes programs I can crank out operative Pascal in a fraction of the time. When I just sit down unshaven to drink black coffee and type for six hours, get up to go to the bathroom, and then remember to see if the damned thing compiles, I reach for my Turbo.

But if you know C, it's not any more obscure than Pascal. It just offers more temptations for those of us who think that beauty and efficiency are two sides of the same coin.



C-BUNDLE \$99

VIEW: CRT Based Disk Diagnostic

EZZAP: ROM Burning Utility
includes schematic

C-PACK: Utilities in C

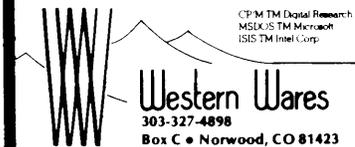
C-Games: User Modifiable Maze Game

All are written in C, include Source Code, and available separately.

ICX TOOLKIT

ICX: ISIS to CP/M or MS-DOS exchanger for MDS and iPDS each \$89

ISE: ISIS Emulator for CP/M \$89



CP/M™ Digital Research
MS-DOS™ Microsoft
ISIS™ Intel Corp.

Figure 2 - A More Readable Figure 1

```

/*****/
change(a)
struct SEC *a;
{
char q,l,item,charQ,*charptr;

display(a);
for(charQ=item=0;){
charptr = &a->byte[item%0x10][item%0x10];
if(charQ) charplace(item); else hexplace(item);
switch(q=grabchar()){
case 0x1b: charQ = lcharQ;
break;
case 0xd: return;
case 0x8: item--;
break;
case 0xa: item += 0x10;
break;
case 0xb: item -= 0x10;
break;
case 0xc: item++;
break;
default: if(charQ) *charptr = q;
else if(isxdigit(q) && isxdigit(l=grabchar())) *charptr = hexit(q,l);
charplace(item);
putchar( isprint(*charptr) ? *charptr : '.' );
hexplace(item);
printf("%02x%c",*charptr & 0xff, (item%0x10 == 7) ? '-' : ' ');
item++;
}
item &= 0x7f;
}
}
/*****/
hexplace(byte)
int byte;
{ /* three bytes per character in the hex mode */
gotoxy(byte/0x10+XDISPL,3*(byte%0x10)+YDISPL);
}
/*****/
charplace(byte)
int byte;
{
gotoxy(byte/0x10+XDISPL,byte%0x10+CHARDISPL);
}

```

Note: Code from Issue 27, Inside the Small C Compiler, should read as follows: Listing 2 - page 25 from 23rd line down (anchors away comment) to 77th line (CALL CBDOS), insert after 26th line from end (Call Main).

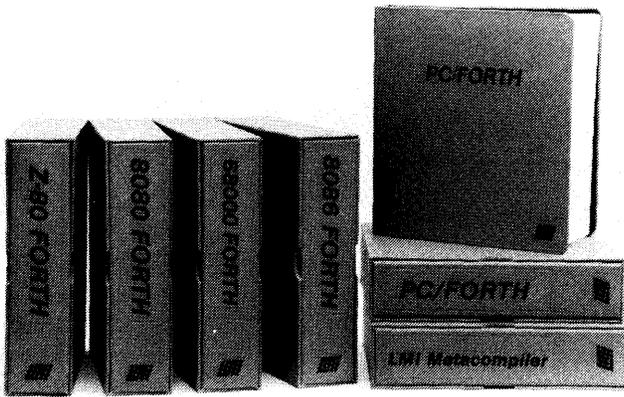
HiSpeed C Library

- **REFERENCE MANUAL** with examples, three indexes, module maps, impatient user's guide (5.5 x 8.5, 410 pp). More than 310 functions in five libraries for Software Toolworks C/80 in Microsoft M80 format (also SLR Systems). Twenty-seven utilities. Requires Digital Research CP/M 2.2 and 8080/Z80 CPU.
- **Develop portable applications** using Bell Labs Unix V7 conventions. Designed from Kernighan & Ritchie, Harbison & Steele, Banahan & Rutter, Sidebottom & Wortman and Donald Knuth's three-volume set.
- **Settable dynamic buffers.** Full stream I/O. Transcendentals. Strings. Math. Searching & sorting. Timing. Compact code (K&R HELLO.COM is 254 bytes). Up to 255 files. Direct-disk. Directory. Memory management. High-speed seek. Fastest block reads. System access. Speed and code size optimization. Automated compile, assemble and link. Ideal incremental development.
- **Useful C Utilities:** Xref, Beauty, Calls, BoldList, Archive, Portable Source Generator, Jump-on-error Monitor.
- **Sample Programs:** Learn or teach C style using source code from textbooks, scholarly periodicals and popular magazines.
- **No royalties.** No copy protection. No limitations on educational use. No extra charges.
- **Reference Manual with library object code on disk** \$29.95
- **Source code archives and utility sources on disk** \$19.95
- **Phone orders shipped immediately. Write or Call today for free brochure.**

Viking C Systems
2243 South Belaire Drive
Salt Lake City, UT 84109
1-801-466-6820/Grant



TOTAL CONTROL with LMI FORTH™



**For Programming Professionals:
an expanding family of
compatible, high-performance,
Forth-83 Standard compilers
for microcomputers**

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, and 6502
- No license fee or royalty for compiled applications

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

**Call or write for detailed product information
and prices. Consulting and Educational Services
available by special arrangement.**

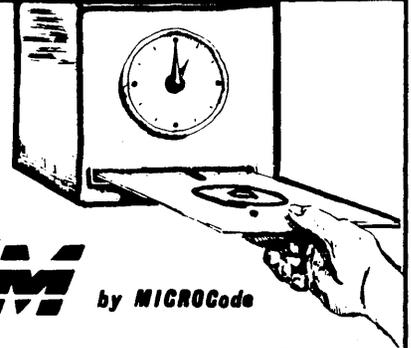
LMI Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, D-7820 Titisee-Neustadt
UK: System Science Ltd., London EC1A 9JX
France: Micro-Sigma S.A.R.L., 75008 Paris
Japan: Southern Pacific Ltd., Yokohama 220
Australia: Wave-onic Associates, 6107 Wilson, W.A.

**AUTOMATIC
TIME/DATE
STAMPING
OF FILES
WITH**

QP/M by MICROCode



ONE PACKAGE FOR ALL CP/M SYSTEMS!

- 100% compatible with CP/M 2.2 and resides in same space
- Uses your computer's hardware/software clock
- 10-15% faster disk read/write
- User-selectable default drive/user area
- Automatic disk re-logging
- Full Access to ALL 31 user areas
- Completely replaces BDOS and CCP
- Efficient backup utility
- Requires QBIOS or CP/M 2.x
- 9 new transient commands, 5 new system calls

AUTOMATIC INSTALLATION TO ANY CP/M SYSTEM

QP/M is automatically written to your system tracks, using QINSTALL and your SYSDISK program.

Kaypro/Xerox/BB-I BIOS Available

MICROCode offers a custom BIOS for Kaypro and Xerox/Big Board owners on a bootable QP/M disk.

COMPLETE QP/M PACKAGE with Documentation Includes

- D Sorted DiRectory program. Displays time/date, system files, and LBR directories.
- QBACKUP Copies and verifies only new or updated files
- QINSTALL Installs QP/M on system tracks.
- QP/P All PIP functions, plus copies files with current or existing time/date.
- QSTAT All STAT functions, plus shows or changes file date(s), archive bit, and MORE.
- QSUB Replaces SUBMIT and XSUB. Has nesting capability, internal command set (including conditionals), embedded XSUB, and more.
- TDCNFG Creates time/date initialization module (supports software clock, Z-Time or Optronic clock).

QBIOS: Available for the Kaypro, Xerox 820-II, Xerox 820-I and BigBoard-I with Emerald Microware or SWP Dual Density (820-I/BB-I). QP/M bootable disk will be shipped if QBIOS is ordered with QP/M. (No installation required!)

NEW PRODUCTS FROM MICROCODE

KEYMAP: Not just another keyboard configuration program! Written exclusively for the Xerox 820/820-II. Full ON-SCREEN keyboard image while editing. Create your own function keys! Multiple custom keyboards can be created, and any 2 can be instantly toggled.

SMARTROMS: Add 820-II features to your 820-I/BigBoard-I. Provides real-time clock, screen dump, screen/program pause, dumb terminal mode, reset with CTRL/ESC, CRT blanking after 11 minutes of idle time, 7 or 8-bit keyboard mode, 820-II (4.0x) compatible vectors, and MUCH MORE! Replaces your socketed ROMs (no soldering!)

PRICES

QP/M.....	\$60.00
QBIOS FOR QP/M.....	\$20.00
KEYMAP.....	\$20.00
SMARTROM SET..... (single density).....	\$30.00
For SWP..... \$40.00..... PLUS 2 (Emerald).....	\$50.00

When ordering, please specify your computer system and media preference (8" IBM 3740 or any popular 5.25" format).

Please add \$3.00 shipping/handling (U.S. & Canada) for UPS shipment, or \$8.00 for foreign air mail. CA residents please add local sales tax (6% or 6.5%). VISA and MasterCard are accepted (add 3%). **FREE information packet available upon request.**

MICROCode Consulting

Department M5
Box 9001
Torrance, CA 90508-9001
(213) MICROCD (24-hour recorder)



AT LAST: AN AFFORDABLE HARD DISK

HARD DISK SYSTEMS by Emerald Microware

Now is the time to take advantage of the low prices on hard disk drives. We have everything you need from the host board to a complete turn-key system for your computer, order only as much as you need. Runs on Xerox 820-1 or -2, Kaypro, or most ANY Z80 based system.

HDS Host board

- Plugs into Z80 socket, no wiring required (special 50 pin host is available for Kaypro 84 series computers)
- Interfaces directly with 40 pin Western Digital 1002 series controller boards
- Comes fully assembled and tested
- Switch selectable I/O addressing

HDS Software package by MICROCode Consulting

- Drivers locate above your existing operating system and BIOS, so it will not interfere with your other programs that install down below.
- Custom installs for your system without having to assemble any software
- Reserves system tracks and swaps so you can Warm Boot from hard drive
- Allows installation of up to three hard drives and two 5 1/4" floppies on one WD1002-05 controller board
- Runs diagnostic on system each time hard drive is installed
- Can be configured for 5 to 40 Meg hard drives, from menu or custom install
- Includes manual, FORMAT, TEST, PARK, INSTALL, and SWAP utilities

Western Digital 1002-05 Hard Disk Controller Board

- 5 1/4" profile with standard 5 1/4" power connection (only 5 VDC required)
- Has WD2797 floppy controller and interface on board
- Can control up to three hard drives and four 5 1/4" floppies.
- 40 pin host interface
- ST506 compatible hard drive interface

HDS TURN-KEY HARD DISK SUBSYSTEM by Emerald Microware

- External hard disk subsystem complete with all cables, controllers, hard drive, and cabinets with power supply, fully tested and burned in. Host adaptor and full installation kit supplied with software on disk format of your choice.
- Internal hard disk subsystem for '84 series Kaypro computers, complete with host, controller, cables, 3 1/2" hard drive, mounting hardware, and fan.
- Available as 10, 20, or 40 Meg, single or multi-drive systems
- External units available with both hard disk and 5 1/4" quad density floppy drives installed

10 Meg Single drive Subsystem	\$ 795.00
20 Meg Single drive Subsystem	\$ 995.00
Host board with software package	\$ 89.00
WD1002-05 Hard disk controller board	\$ 185.00
WD1002-05 controller with Host and software	\$ 250.00
20 Meg with dual 5 1/4" floppy subsystem	\$1295.00

XEROX BOARDS AND ACCESSORIES

We stock a limited quantity of Xerox parts and accessories for the 820 computers.

Xerox 820-2 Main Computer Board	\$ 95.00
Xerox 820-2 Floppy controller board	\$ 75.00
Xerox 820-2 Main board with floppy controller board	\$150.00
Xerox 5 1/4" disk drive cable	\$ 12.00
Xerox internal video harness with brightness control	\$ 9.00
Parallel ASCII keyboard (no numeric pad) with docs	\$ 25.00
PC mount power connector for 820 board	\$ 2.50
Dual half height, 5 1/4" disk drives (new), DSDD - 48 TPI, in cabinet with Xerox standard cable	\$265.00
Dual half height, 5 1/4" disk drives (new), DSQD - 96 TPI, in cabinet with Xerox standard cable	\$295.00

We guarantee satisfaction on all of our products or your money back. Include \$4.00 shipping and handling, \$6.00 for COD, add \$3.00 for Blue Label. Phone hours: 8:30am to 5:00pm weekdays; modem hours: 10:00pm to 8:00 am seven days a week.

WE SUPPORT OUR CUSTOMERS

Is single density making your Xerox 820-1 feel crowded?

DOUBLE DENSITY FOR THE XEROX 820-1 OR BIG BOARD I

Clearly the most versatile double density package available for the 820-1. Lets you run up to four disk drives at once, and mix 5 1/4" & 8". Runs almost all Kaypro and Xerox 820 software. Supports all standard printers and most add-ons such as the Ferguson ram board. Our ROM sets operate in banked mode to allow more space in the TPA for your programs. You get Mini-monitor functions, autoboot capability, automatic disk identification (up to 19 formats built in), IOBYTE enabled for device selections, and you can read and write 48 TPI disks on 96 TPI disk drives.

X120 Board - assembled and tested - with ROM set	\$160.00
X120 Board - full kit - with ROM set	\$140.00
X120 Bare board with ROM set	\$ 72.00
X120 Bare board only	\$ 26.00
ROM set only	\$ 50.00
Hard to find parts set (trimmers, caps, resistors)	\$ 12.00
WD2793 controller chip	\$ 22.00
Other packages available.	

Can't find CP/M for your Xerox? Now you don't need it.

QP/M by MICROCode Consulting

At last, an inexpensive replacement for CP/M, and you don't need CP/M to get started! Get full CP/M compatibility with many new options. Stamp your file with the time and date - transparent to your operation, automatically re-log your disks (no more control C), change your default drive search from the keyboard, use up to 31 USER areas, select your disk drives or user areas with semicolon or colon, maintains ARCHIVING bit for efficient backups (great for hard drives), and much, much more. Runs in native Z80 code, in the same space as CP/M 2.2. Has menu oriented install for use on ANY Z80 based system. Available as a bootable disk (with BIOS) for the Xerox 820-1, 820-2, 820-XPRO, 820-PLUS2, or Kaypro.

QP/M, bootable for the Xerox 820-1, 820-2, or Kaypro	\$ 80.00
QP/M, bootable for the Xerox 820-1 using XPRO or PLUS2	\$ 65.00
QP/M, without BIOS (not bootable)	\$ 60.00

Tired of carrying your computer back and forth? Why not just the diskette?

UNIFORM by Micro Solutions

The solution to the diskette incompatibility problem. This program allows files to be transferred back and forth between close to a hundred computer formats including between CP/M and MSDOS. Lists computers by name - no special technical knowledge required. Versions available from stock for the Kaypro (XPRO), Xerox 820-2, Morrow MD2&3, and IBM 100% compatibles. Other versions available on request.

Uniform	\$ 64.95
---------	----------

PANASONIC HALF HEIGHT DISK DRIVES

These are some of the finest new 5 1/4" disk drives. They are fast, reliable, and have about the lowest power consumption around.

National JA551 (Shugart 455) - DSDD, 48 TPI	\$114.00
National JA561 (Shugart 465) - DSQD, 96 TPI	\$129.00



P.O. BOX 6118 ALOHA, OR 97007 (503) 642-1860

Xerox 820 Column

By Mitchell Mlinar

1225 Fonthill Ave.
Torrance CA 90503

The Z80 SIO is the communications chip in nearly every Z80 based system. It's very powerful and very smart, but programming it is not a trivial process (just look at the manual). Mitch continues with the SIO in this column.

Last time, I discussed SIO software configuration. This issue, I'll finish off the topic with the baud rate and sample input/output programming.

Setting The Baud

After initializing the SIO for the desired bit-width, frame, mode, and parity, you need to set the baud rate. Baud rate is determined by the 8116 baud rate generator. This chip has a write-only register for baud (e.g., you cannot check the current baud rate by reading the register) which is located at I/O port address 00 for communications and 0CH (12 decimal) for the PRINTER. The value written to the port depends on the baud rate as shown below.

Baud Rate Values for COM 8116

Baud Rate (bps)	Value (hex)
50	00
75	01
110	02
134.5	03
150	04
300	05
600	06
1200	07
1800	08
2000	09
2400	0A
3600	0B
4800	0C
7200	0D
9600	0E
19200	0F

Due to the high speed of the Z80 as compared to the snail's pace of serial transmission, sending and receiving characters requires some handshake; this is provided through the SIO read registers 0 and 1 which are detailed in Figure 1 for asynchronous operation.

To access either read register, you'll need to disable interrupts, set the register number as described last issue, fetch the value, and restore interrupts. Even more important for detecting errors, if ANY status bit changes

Figure 1 - SIO Read Registers 0 and 1

```
Read Register 0
Bit 7 6 5 4 3 2 1 0
Name Brk x CTS x DTR Tx Empty Int. Pending Rx avail.

where "Brk" indicates a break has been received
"CTS" and "DTR" status of clear to send & data terminal ready lines
"Tx Empty" is high to indicate that the transmit is
empty and ready to accept another character
"Int. Pending" is used on channel A (COM) to indicate
an interrupt has occurred
"Rx avail." is high to indicate a character has been
received.
```

```
Read Register 1
Bit 7 6 5 4 3 2 1 0
Name x Frame Err Rx Overrun Err Parity Err x x x

where "... Err" is high if that type of error occurred
```

Figure 2

```
;
; Get read status: 00 if not ready, FF if char is ready
;
RXSTAT: DI ;since we are playing with SIO registers
XOR A ;force SIO to read/write register 0
OUT (6),A ; done via control port
IN A,(6) ;get read register 0
PUSH AF ;save read status
LD A,10H ;reset ext/status interrupts
OUT (6),A ; via write register 0
EI ;restore interrupts
POP AF ;restore read status
;
; ..... ;could have error checking here (CTS, DCD, ..)
RRCA ;put 'Rx avail' into carry flag
LD A,0 ;clear out A register w/o destroying carry
SBC A,A ; gives us 0 for no char, FF if char ready
RET ;exit with result
;
; Get character into A register from input
;
RX: CALL RCVSTAT ;loop until character is available
JR Z,RX ; is Z when no char yet
IN A,(4) ;get character
RET ; and exit with it
;
; Transmit status: 00 if ready, FF if still full
;
TXSTAT: DI ;
XOR A ;set read/write register 0
OUT (6),A ; is set
IN A,(6) ;get register
EI ;leave ext/status alone on transmit
AND 4 ;strip all but xmit status
XOR 4 ;flip status (so is zero if ready)
RET Z ;exit if ready
OR OFFH ;is full, so give all bits high
RET ;return to calling program
;
; Transmit character in C register
;
TX: CALL TXSTAT
JR NZ,TX ;loop until transmit buffer empty
LD A,C ;get char
OUT (4),A ; and send it
RET ;return to calling program
```

(except for "Tx Empty" and "Rx avail."), the read registers are not updated until a "reset ext/status interrupts" occurs. This is easily done by writing 10H to "write register 0" before restoring interrupts during read access.

Initialization

If I wanted to write my own modem module to attach to a general communications program, there are usually five modules required: initialization, receive status, transmit status, receive

character, and transmit character. Initialization was described last time (plus baud rate in this article). For the COM channel, port 04 is data, port 06 is control. See Figure 2.

Notice how handshake for incoming/outgoing characters is handled with only two bits. More advanced applications use interrupts so the processor can do other things besides being stuck in a loop between characters.



NOW FOR XEROX 820 BB & BBII

LOW COST DISK CONTROLLER

SAVE WEAR AND TEAR ON YOUR DISK DRIVES AND FLOPPIES WITH THE MODEL 3801 ALL SOLID STATE RELAY. SMALL ENOUGH TO FIT EASILY INSIDE YOUR DISK DRIVE, THE 3801 CAN BE INSTALLED IN MINUTES. YOUR BIG BOARD WILL THEN TURN YOUR DRIVES ON AND OFF AUTOMATICALLY AS THEY ARE NEEDED.

- FEATURES -

- * SMALL SIZE - 1.75 X 1.40 X 0.35
- * FAST INSTALLATION - DIAGRAM INCLUDED
- * ZERO CROSSING - ELIMINATES ALL SWITCHING NOISE
- * TRIAC OUTPUT - NO MECHANICAL PARTS
- * DVDT FILTER - INCLUDED
- * LOW COST - ONLY \$8.80 EACH
- * 1 YEAR WARRANTY
- * 30 DAY MONEY BACK GUARANTEE

PERNIX

COMPANY OF CALIFORNIA
1619 SOUTH MINNIE STREET
SANTA ANA, CALIFORNIA 92707

(714) 547-4316

CALIF. RES. ADD 6% SALES TAX
ADD \$1.00 EA. POSTAGE & HANDLING
PLEASE SPECIFY BBI, BBII OR XEROX

SINGLE BOARD HIGH PERFORMANCE ENCLOSURE

FOR 820, BIG BOARD OR OTHER SINGLE BOARD COMPUTER

High Res 12" Motorola Monitor
Big 200 Watt Power Supply
All Wired Up & Ready To Slide in Your SBC \$169.00

A BEST BUY

Beautiful - Keytronics full parallel ASCII (hooks right up to 820 or BB) Keyboard with numeric keypad/function keys.
A Best Buy \$ 39.00
Extra for Keyboard Enclosure \$ 15.00
MONITOR ONLY..... \$ 59.00
BIG POWER SUPPLY ONLY..... \$ 35.00
MATRIX KEYBOARDS..... \$ 15.00
8" Disk Drive Shugart 810 Half Height..... \$ 99.00
2 8" Disk Drives in Enclosure with power..... \$299.00
2 5 1/4" Disk Drives in Enclosure..... \$149.00

XEROX 820-1 COMPUTERS

ENTIRE 820-1 SYSTEM IN ENCLOSURE WITH KEYTRONICS KEYBOARD

with 2 8" Drives..... \$699.00
with 2 5 1/4" Drives..... \$599.00
with Diablo 630 Printer add..... \$899.00
(Diablo sold only with complete system)

DIGITAL RESEARCH CP/M WITH BIOS FOR 820-1 OR 820-11 INCLUDING LICENSE AND ALL MANUALS 100 AVAILABLE..... \$49.00
ENHANCED MICRO-CODE QPM BOOTABLE DISC SPECIAL WITH COMPUTER ONLY..... \$45.00
CENTRONICS PARALLEL PRINTER CABLE 820-1 or 820-11..... \$35.00
PARALLEL DRIVER(COM FILE)..... \$15.00

EVERYTHING WE SELL IS GUARANTEED TO WORK!

SPECIAL ON CP/M SOFTWARE CLOSEOUT

NEW LICENSED DISK

WORDSTAR 3.3 or 3.01..... \$99.00
SUPERSORT..... \$50.00
SPELLSTAR..... \$45.00
MAILMERGE..... \$50.00

★ New AUTOLETTER for Dbase II★
Write form letters or normal letters automatically to your clients, customers or anyone in your database without typing names and addresses!
Also does single or multiple envelopes or labels.
Add your own forms or letters..... \$99.00



**CRAIG'S
COMPUTERS**

All prices: plus shipping
and handling

(213) 379-1342

NEW... FLOPPY DISK CONTROLLER BOARD FOR THE XEROX 820-II

Direct replacement for the standard 820-II Floppy Daughter Board. Uses the FDC1793/9229B for reliable operation. Perfect for surplus 820-II boards.

Assembled/tested..... \$109.95
Complete Kit..... \$79.95
Bare Board w/9229B..... \$59.95
Bare Board w/DOC..... \$39.95

Include \$3.00 Shipping
VISA/MC COD Add \$1.50

REPAIR SERVICE
XEROX 820 I/II BIG BOARD I

THE DATA MILL
12 Crescent Court
Montgomery, IL 60538
312-892-6573

J.B. FERGUSON, INC.

817-640-0207

ONE MEGABYTE RAM DISK FOR 820-1 & BIGBOARD-1

- THIS IS THE **FASTEST** RAM DISK THAT YOU CAN BUY
- EASY INSTALLATION **WITHOUT** ETCH CUTS, CABLES, ETC.
- USES **NONE** OF YOUR PRECIOUS I/O OR THE CPU SOCKET
- CONVENIENT PLUG-IN DESIGN
- RESIDES IN THE MEMORY ADDRESS SPACE OF THE COMPUTER
- COMPATIBLE WITH SWP AND EMERALD DOUBLE DENSITY
- EXPANDABLE IN 256K INCREMENTS
- SMALL 4.5" x 6.5" x .75"
- PLUGS INTO THE SAME SOCKET AS THE 820-256K

This product comes with complete machine-drawn schematics, and a full theory of op. Assembly, installation and test procedures are thoroughly documented. PAL code is included, and extra PALs are available. A PAL is shipped with each system. RAM disk software is included with source on disk. Please specify 5 or 8 inch disk.

P.C. board, interconnect hardware, documentation, PAL and software \$ 75.00
 Kit w/o RAM (all TTL, Bypass capacitors, sockets, plus all of above)..... 125.00
 SPECIAL !!! 150 nS 256K dynamic RAMs when ordered with the kit..... ea 3.25
 Monitor ROM for the 820-or-BB1 with the initialization constants for RAM mapper..... 10.00

UNIVERSAL COMPOSITE VIDEO ADAPTER

This device was designed to interface any computer with TTL-compatible separate-sync video outputs (like the 820-2) to a composite video monitor. The polarity and duration of the horizontal and vertical sync pulses are individually adjustable. This allows the user to alter the framing of the display. The output is via RCA phono jack w/75Ω output impedance. Dot rise and fall times are < 40 nS for sharp well-defined characters.....\$25.00

820-2 FLOPPY DISK CONTROLLER

This package consists of the 820-2 floppy PWA bare PC board, the data separator ROM, a full set of schematics and a parts list. The PC board and the schematics are for the revision 5 Xerox PWA. The assembly drawings and the PC board are functionally identical to the Xerox product. This package will be available in mid-February.

\$35.00

820-2 B SIZE SCHEMATICS, ETCH 2	\$15.00
820-1 B SIZE SCHEMATICS	15.00
820 RESET SWITCH.....	3.00
820 VIDEO CABLE,CONNECTS 820 TO SEPARATE-SYNC MONITOR, W/PROVISIONS FOR BRIGHTNESS POT	10.00
820-1 CBIOS AND ROM SOURCE, 5 1/4" OR 8" DISK.....	35.00
820 5 1/4" DUAL DRIVE DISK CABLE W/POWER CONN.....	20.00
820 NINE PIN POWER CONNECTOR (PLUGS INTO 820).....	4.00
820 NINE PIN PC CONNECTOR (SOLDERS ONTO 820).....	4.00
820-2 FLOPPY DISK CONTROLLER SCHEMATIC.....	10.00
CP/M PRIMER (XEROX CP/M HANDBOOK)	7.50
XEROX WORDSTAR HANDBOOK	4.50

820-225.00

These computer boards have all the TTL, capacitors, resistors, connectors, Z80 peripheral devices, and the baud rate generator. Only the CPU, the ROMs and the disk controller is not included. Schematics included. 5.01 ROM set included on disk w/ char generators, all as hex files.

ASTEC POWER SUPPLY.... \$35.00

These supplies are brand-new 65 WATT off-line switchers. They have four independent outputs rated at: +5V @ 6A, +12V @ 1.2A, +12V @ 2.1A & -12V @ .25A. SIZE: 7.7"W x 2.0"H x 4.2"D

820-1 COMPOSITE VIDEO ADAPTER

Assembled and tested.
 Plugs into 820's video connector. **\$15.00**
 Produces high-quality video.
 RCA phono jack output.

J.B. Ferguson, Inc.
 P.O. Box 300085
 Arlington, Texas 76010
 (817) 640-0207

TERMS: MasterCard/VISA, money order, certified check, UPS cash COD. Allow 2 weeks for personal or company checks to clear. Add \$5 for shipping/handling. Texas add 5.125% state sales tax.

M.A.G.I.C.TM COMPUTERIZED INVESTING

FOR ASTUTE INVESTORS AND ALL THOSE AIMING TO LEARN

WARE ANNOUNCES UNIQUE \$79.95 M.A.G.I.C.TM INVESTMENT SOFTWARE

San Diego—Now investors of all skill and experience levels from novice to expert can completely develop and test their own strategies. Under flexible user control, M.A.G.I.C. becomes a computer of objective buy and sell signals, a consistent decision maker, an effective market timer, a profit optimizer, an investment strategy developer. This affordable, innovative investment software is available from **WARE Tools for Investors** for CP/M and MS-DOS systems with a Basic interpreter such as MBASIC, BASICA or GWBASIC.

M.A.G.I.C. gets its name from a descriptive acronym: **M**oving **A**verage **G**enerated **I**vestment **C**riteria. Using techniques explained below, users make retrospective trading runs on historic data to develop the optimum strategy for their chosen vehicle. Then, using this strategy with current prices, M.A.G.I.C. computes buy/sell signals to guide real-time trading in that vehicle.

Dr. R. B. Gordon, author and publisher of M.A.G.I.C. software, based the program logic on his more than 25 years of experience with moving average trading signals. In his words, "It would be difficult to find a more productive use for your computer than using it to improve your personal finances." Paper trading with computer-assisted signals before you invest hard earned money, is the rational way for inexperienced investors to gain confidence and for experienced investors to attain even greater skills. Put your computer to work! M.A.G.I.C. will compute the buy/sell signals and then tally the profits. Full details follow.

WHAT M.A.G.I.C.TM DOES

If you are a successful investor or aim to become one, M.A.G.I.C. will sharpen and enhance your skills. If you are a beginner or want to start now, M.A.G.I.C. teaches you to trade profitably before risking your capital. M.A.G.I.C. is a universal moving average trading system development tool, not a trading system. It is a means to an end—the end you decide is best for you—not an end in itself. M.A.G.I.C. software does its work thru three modules, each doing its specialized job in proper order.

(1) **DMAGIC** processes price data available on disk or entered via the keyboard and puts it into a form usable by **MAGIC**. Stock split and mutual fund dividend data are entered so that corrected prices can be used in all computations.

(2) **MAGIC** computes the moving averages and oscillators and produces extensive 132 column printouts, for groups of up to 15 securities, commodities or market indices. It first computes any 6 user-chosen moving averages and then goes on to compute 12 percentage differences between pairs of moving averages. These percentages are called oscillators because their values fluctuate above and below the zero point in waves that can duplicate those occurring in the price of the market or security in question. The term, often used to describe these oscillators, *overbought/oversold*, is very apt since high oscillator values denote price tops for selling and low values denote price bottoms for buying. Using the extensive historic price data on the distribution disc, the

MAGIC module gives users an immediate capability to develop optimum buy and sell strategies to suit their investment temperament, whether it means day trading in volatile media or conservative long term investing in blue chip securities or mutual funds.

(3) **SMAGIC** (pronounced DOLLARMAGIC) does computerized trading on oscillator values computed by **MAGIC**. It uses simple or sophisticated trading parameters, input by the user. It computes the date and price for each buy or sell (long only or long and short), the profit or loss of each trade and an overall summary of that trading campaign on the selected security and oscillator. Its extensive output goes to both screen and printer. See the sample printout below.

\$MAGIC 5 YEAR TRADING RUN

LISTING OF COMPLETED SMALL TRADES FOR HISTORIC DATA COMMON STOCK (SEEING CAPITAL) 11/00/80

TRADING LONG AND SHORT WITH M.A.G.I.C. OSCILLATOR FROM 800711 TO 808000

DATE ENTERED = 02/08/81

SECURITY	STRAT	LONG PRICES	SHORT PRICES	DURATION	NO. OF TRADES	PROFIT (LOSS)	MAX. UNREALIZED GAIN (LOSS)	NET PROFIT (LOSS)
800000	1	1.00	1.00	0	0	0.00	0.00	0.00
800001	1	1.00	1.00	0	0	0.00	0.00	0.00
800002	1	1.00	1.00	0	0	0.00	0.00	0.00
800003	1	1.00	1.00	0	0	0.00	0.00	0.00
800004	1	1.00	1.00	0	0	0.00	0.00	0.00
800005	1	1.00	1.00	0	0	0.00	0.00	0.00
800006	1	1.00	1.00	0	0	0.00	0.00	0.00
800007	1	1.00	1.00	0	0	0.00	0.00	0.00
800008	1	1.00	1.00	0	0	0.00	0.00	0.00
800009	1	1.00	1.00	0	0	0.00	0.00	0.00
800010	1	1.00	1.00	0	0	0.00	0.00	0.00
800011	1	1.00	1.00	0	0	0.00	0.00	0.00
800012	1	1.00	1.00	0	0	0.00	0.00	0.00
800013	1	1.00	1.00	0	0	0.00	0.00	0.00
800014	1	1.00	1.00	0	0	0.00	0.00	0.00
800015	1	1.00	1.00	0	0	0.00	0.00	0.00
800016	1	1.00	1.00	0	0	0.00	0.00	0.00
800017	1	1.00	1.00	0	0	0.00	0.00	0.00
800018	1	1.00	1.00	0	0	0.00	0.00	0.00
800019	1	1.00	1.00	0	0	0.00	0.00	0.00
800020	1	1.00	1.00	0	0	0.00	0.00	0.00
800021	1	1.00	1.00	0	0	0.00	0.00	0.00
800022	1	1.00	1.00	0	0	0.00	0.00	0.00
800023	1	1.00	1.00	0	0	0.00	0.00	0.00
800024	1	1.00	1.00	0	0	0.00	0.00	0.00
800025	1	1.00	1.00	0	0	0.00	0.00	0.00
800026	1	1.00	1.00	0	0	0.00	0.00	0.00
800027	1	1.00	1.00	0	0	0.00	0.00	0.00
800028	1	1.00	1.00	0	0	0.00	0.00	0.00
800029	1	1.00	1.00	0	0	0.00	0.00	0.00
800030	1	1.00	1.00	0	0	0.00	0.00	0.00
800031	1	1.00	1.00	0	0	0.00	0.00	0.00
800032	1	1.00	1.00	0	0	0.00	0.00	0.00
800033	1	1.00	1.00	0	0	0.00	0.00	0.00
800034	1	1.00	1.00	0	0	0.00	0.00	0.00
800035	1	1.00	1.00	0	0	0.00	0.00	0.00
800036	1	1.00	1.00	0	0	0.00	0.00	0.00
800037	1	1.00	1.00	0	0	0.00	0.00	0.00
800038	1	1.00	1.00	0	0	0.00	0.00	0.00
800039	1	1.00	1.00	0	0	0.00	0.00	0.00
800040	1	1.00	1.00	0	0	0.00	0.00	0.00
800041	1	1.00	1.00	0	0	0.00	0.00	0.00
800042	1	1.00	1.00	0	0	0.00	0.00	0.00
800043	1	1.00	1.00	0	0	0.00	0.00	0.00
800044	1	1.00	1.00	0	0	0.00	0.00	0.00
800045	1	1.00	1.00	0	0	0.00	0.00	0.00
800046	1	1.00	1.00	0	0	0.00	0.00	0.00
800047	1	1.00	1.00	0	0	0.00	0.00	0.00
800048	1	1.00	1.00	0	0	0.00	0.00	0.00
800049	1	1.00	1.00	0	0	0.00	0.00	0.00
800050	1	1.00	1.00	0	0	0.00	0.00	0.00
800051	1	1.00	1.00	0	0	0.00	0.00	0.00
800052	1	1.00	1.00	0	0	0.00	0.00	0.00
800053	1	1.00	1.00	0	0	0.00	0.00	0.00
800054	1	1.00	1.00	0	0	0.00	0.00	0.00
800055	1	1.00	1.00	0	0	0.00	0.00	0.00
800056	1	1.00	1.00	0	0	0.00	0.00	0.00
800057	1	1.00	1.00	0	0	0.00	0.00	0.00
800058	1	1.00	1.00	0	0	0.00	0.00	0.00
800059	1	1.00	1.00	0	0	0.00	0.00	0.00
800060	1	1.00	1.00	0	0	0.00	0.00	0.00
800061	1	1.00	1.00	0	0	0.00	0.00	0.00
800062	1	1.00	1.00	0	0	0.00	0.00	0.00
800063	1	1.00	1.00	0	0	0.00	0.00	0.00
800064	1	1.00	1.00	0	0	0.00	0.00	0.00
800065	1	1.00	1.00	0	0	0.00	0.00	0.00
800066	1	1.00	1.00	0	0	0.00	0.00	0.00
800067	1	1.00	1.00	0	0	0.00	0.00	0.00
800068	1	1.00	1.00	0	0	0.00	0.00	0.00
800069	1	1.00	1.00	0	0	0.00	0.00	0.00
800070	1	1.00	1.00	0	0	0.00	0.00	0.00
800071	1	1.00	1.00	0	0	0.00	0.00	0.00
800072	1	1.00	1.00	0	0	0.00	0.00	0.00
800073	1	1.00	1.00	0	0	0.00	0.00	0.00
800074	1	1.00	1.00	0	0	0.00	0.00	0.00
800075	1	1.00	1.00	0	0	0.00	0.00	0.00
800076	1	1.00	1.00	0	0	0.00	0.00	0.00
800077	1	1.00	1.00	0	0	0.00	0.00	0.00
800078	1	1.00	1.00	0	0	0.00	0.00	0.00
800079	1	1.00	1.00	0	0	0.00	0.00	0.00
800080	1	1.00	1.00	0	0	0.00	0.00	0.00
800081	1	1.00	1.00	0	0	0.00	0.00	0.00
800082	1	1.00	1.00	0	0	0.00	0.00	0.00
800083	1	1.00	1.00	0	0	0.00	0.00	0.00
800084	1	1.00	1.00	0	0	0.00	0.00	0.00
800085	1	1.00	1.00	0	0	0.00	0.00	0.00
800086	1	1.00	1.00	0	0	0.00	0.00	0.00
800087	1	1.00	1.00	0	0	0.00	0.00	0.00
800088	1	1.00	1.00	0	0	0.00	0.00	0.00
800089	1	1.00	1.00	0	0	0.00	0.00	0.00
800090	1	1.00	1.00	0	0	0.00	0.00	0.00
800091	1	1.00	1.00	0	0	0.00	0.00	0.00
800092	1	1.00	1.00	0	0	0.00	0.00	0.00
800093	1	1.00	1.00	0	0	0.00	0.00	0.00
800094	1	1.00	1.00	0	0	0.00	0.00	0.00
800095	1	1.00	1.00	0	0	0.00	0.00	0.00
800096	1	1.00	1.00	0	0	0.00	0.00	0.00
800097	1	1.00	1.00	0	0	0.00	0.00	0.00
800098	1	1.00	1.00	0	0	0.00	0.00	0.00
800099	1	1.00	1.00	0	0	0.00	0.00	0.00
800100	1	1.00	1.00	0	0	0.00	0.00	0.00

HOW M.A.G.I.C.TM WORKS

M.A.G.I.C. software develops its buy and sell signals thru an analysis of one or more of the 12 overbought/oversold oscillators. The oscillators detect significant changes in price trend and signal potential buy and sell points. For each security, the user makes a series of retrospective trading runs with historic data. Once the best trading method has been determined in this way, the program uses real time prices to compute current buy/sell actions.

Markets of all kinds rarely stand still. Security and commodity prices are continually going up and down. There are cycles measured in minutes, hours, days, weeks, months and years. Price swings are magnified by excesses in investor emotions. When rising for whatever reason, the trend continues above a rational price due to investor greed; when falling, the trend continues below a rational price due to investor fear. Cyclical price tops and bottoms are jagged, making it difficult to detect trend changes. So, moving averages are used to smooth the price fluctuations. A ten day, or ten week, moving average means simply the average of the past 10 daily or weekly prices. The value of this average changes or "moves" by being recalculated each day or week.

M.A.G.I.C.'s time-tested techniques permit its users to exploit recurring price cycles by buying lower and selling higher than the majority of traders. Use of moving averages to time buys and sells on the overall market or on individual securities is the oldest, most reliable method yet devised. And, by computerizing the well-developed moving average trading rules, M.A.G.I.C. develops systems that are optimized for each vehicle and investor need.

USING THE M.A.G.I.C.TM COMPUTERIZED INVESTING TOOLS

Effective use of M.A.G.I.C. takes no more time than your life style and investment needs dictate. A retiree aiming to maximize total return on invested capital might profitably devote an hour a day. A typically busy, working individual with long term objectives might spend as little as 15 minutes per month and, in that time, make the highest hourly rate imaginable. Another investor might track a group of 12 funds on a weekly basis in no more than 20 minutes per week. In contrast, a day trader could devote up to 10 hours a week in trading securities or commodities.

There are as many ways to apply M.A.G.I.C.'s capabilities as there are users. Since **WARE** does not sell a "system", but gives users the tools to develop their own best strategies, there is no concern about too many users "spoiling the system". M.A.G.I.C. provides a virtually unlimited number of buy and sell strategies to yield an optimized result for each vehicle studied. Yet, this vast output is readily managed by a few input parameters. M.A.G.I.C.'s flexibility enables the selected strategy to range from simple to very sophisticated.

Investors familiar with moving averages will put the M.A.G.I.C. oscillators to effective use immediately. Novices will quickly appreciate the potential of computerized investing and the opportunity to learn on paper before risking their capital. All users will enjoy watching capital accumulating on the screen during a profitable trading run. But M.A.G.I.C. is not a game; it is a professional tool to help investors at all levels of experience and skill improve the profitability of their investments.

M.A.G.I.C. is menu-driven and very user-friendly. Complete instructions are provided in a clearly written, 65 page illustrated Manual. And after-sale telephone consultation is available if needed. Some investment background is helpful but not necessary since the basic elements of technical trading systems are covered in the Manual. M.A.G.I.C. software is very affordable. Price of the 3 module package, is a tax-deductible \$79.95 (\$83.45 postpaid). A super, "working" demo, Little M.A.G.I.C., is just \$25.00 ppd.

LITTLE M.A.G.I.C.TM DEMO AND TUTOR

The Demo disc holds shortened versions of the **MAGIC** and **\$MAGIC** modules plus historic security data. Easy run via menus and help messages on screen, **LITTLE M.A.G.I.C.** produces typical output to both screen and printer and provides unlimited practice runs and many enjoyable, educational hours. The Demo package contains on-disc tutorials on technical analysis, moving averages and oscillators and a 13 page illustrated Manual.

Non-investors and investors unfamiliar with technical analysis will find that the Demo gives them an easy way to learn how to create and use moving average oscillator buy/sell signals. The \$25.00 cost of the Demo, less a \$10 service fee, is refundable on any subsequent purchase of M.A.G.I.C. When purchased in combination with the full program, the Demo is available for an extra \$6.50. Over 3/4 of our customers choose this combo for a quick introduction to the immense capabilities of the **MAGIC** and **\$MAGIC** modules.

SEND FOR FREE ILLUSTRATED BROCHURE/CATALOG

M.A.G.I.C. ORDER FORM (Mark Items Below)

M.A.G.I.C. 3 modules, 65 p. Manual, ppd. \$83.45
 LITTLE M

Home Care For The Invalid Computer (Part 2)

By David Thompson

Last issue we discussed a couple of simple diagnostic methods that just about any can handle. This time we're starting into a more sophisticated method. Fortunately it requires very little equipment, unfortunately it requires some real understanding of the system.

I remember the first time I saw a schematic for a digital device. I was dumbfounded. There were no resistors, capacitors, or transistors. For all I could tell the strange shapes on the paper didn't even receive 5V. (Truly low power devices.)

That was eight years ago, a time when you didn't get involved with a microcomputer if you couldn't read a schematic (at least not until the Apple II and TRS-80 showed up).

Apple and Radio Shack made their marks by providing systems that you could use without any inside information. Of course, if the system you got didn't work very well you wound up on a first name basis with service support. Now, of course, everything runs right out of the box (or in the box, or with no box at all).

How does this relate to servicing? It's hard to fix something you don't understand, and what we've gained in instant use, we've lost in knowledge of the system. When you build something or modify it, you gain a lot more than manual dexterity.

Computer dealers are in the same boat as the rest of us. They no longer put together systems. They unpack them, put them on tables, and then look for the on/off switches. (One person I know went into a computer store, sat down at their AT&T Unix demo, and changed the password. Last I heard, they still hadn't figured out how to get back into the system — but that's another story.)

Board Swapping and T&S

In issue #27 we covered two simple diagnostic procedures — board swapping and temperature & stress. Those are the procedures I use if at all possible. Board swapping is fast, but it may not be feasible unless you have a friend with a similar system (and even

after you've removed the defective board you might want to repair it rather than toss it). Temperature & stress only works on problems that are related to temperature sensitive ICs and intermittent connections.

Wiggling Lines

To an outsider, the inside of a computer no doubt looks like a maze of unrelated copper traces and IC pins. Output pins are connected to the traces which carry the information to input pins. The input information is scrambled by the ICs and output again. There is no apparent rhyme nor reason for anything that happens inside a computer — even when the computer is working correctly.

When the computer isn't working correctly, where do you look? How do you tell what's right and what's wrong?

Well, just as we made some assumptions when we tried temperature & stress (a chip that's sensitive to heating or cooling is probably defective), we're going to make some assumptions here. The primary assumption is that we have a "hard" failure, e.g., the defective part or line is receiving information but its output (or other end) is stuck. What do I mean "stuck"?

A stuck output is either high (around +4V) or low (around +0.7V) or somewhere in the undefined region inbetween (+2V to +3V). It doesn't move enough to change states (go from high to low or vice versa).

A stuck line is either open (bad socket, bad solder joint, break in the copper trace) or shorted to another line. (When two lines short together the problem may be very difficult to find unless one is ground or +5V.)

So we're looking for lines that are moving high to low, low to high. If they're moving, we'll assume they're OK. If they're not moving then we need to know if they should be. If they should be, all we need to figure out is why they aren't.

Tools

You can't just look at a line or a pin and tell if it's moving (at least I can't).

Even holding your finger on a line won't tell you much (unless it's, say, 110VAC). You need a sensitive little helper to tell you what's going on.

Voltmeter

A VoltOhmMilliammeter (VOM) will read a stuck voltage, but I haven't found one that will reliably tell you whether a line is wiggling.

You should have a meter just on principle (if only to give your workbench that professional look), and if you suspect you have shorts somewhere (we'll discuss longs another time) a simple resistance measurement will tell you very quickly.

Meters aren't particularly expensive (one of the newer digital models for \$50 to \$100 would be good). Stay with the name brands — even Fluke has had some really good deals on digital meters lately, and they are super quality.

You should use a meter to check the power supply outputs (down to a shocked gnat's eyelash), but you still need a logic probe or an oscilloscope to see whether the line is moving or not.

Logic Probes

There are lots of logic probes, and they all do about the same thing. Most have one light (usually green) which comes on when they detect a voltage below about +1.7V and another light (red) which comes on when they detect a voltage above +3V.

That's the easy part. They also have pulse stretchers. Say that the line you're monitoring stays high for 1 second, then goes low for 1 us. Without the pulse stretcher, you wouldn't see that short low. The lamp couldn't light in 1 millionth of a second, but even if it could, your eye wouldn't see it. But, no matter what the relative duration of the highs and lows, if they're both there, a logic probe will be merrily blinking red, green, red, green...

Probes cost between \$20 and \$100. The difference is brand name, construction quality, and speed. If you need to see very short pulses (50 ns or shorter) then plan to pay \$100. The

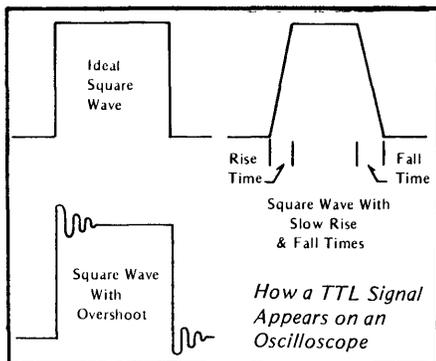
\$20 units from places like Radio Shack are usually quite adequate for most micros.

Scopes

Oscilloscopes are the only instruments which can tell you what the signal really looks like (logic analyzers give up some of the detailed "tell it like it is" information that scopes provide for a broader view of many signals).

Scopes are much more expensive than logic probes and more difficult to use and understand. The temptation is to purchase an old Heathkit 5 MHz scope (your system is only running 4 MHz right?) for \$50 or \$100 and then assume the information on the tube is correct. It ain't.

The slowest scope I'd use would have a 50 MHz bandwidth. Period. It would also have an X10 probe (10 meg Ohm 20 pF), and two channels. Part of the reason you use a scope is to see what the signal looks like. Is it rising quickly enough? Are there extra squiggles (called ringing)?



A slow scope will show you garbage that isn't there and will miss garbage that is. A low impedance probe (X1) will slow up the rise time and cause ringing just by its presence. So a signal that is otherwise fine will look horrible.

Expect to pay between \$800 and \$2000 for a good scope, new. If you can find an old Tektronix 540 through 547 series monster in good running order (with dual trace 50 MHz plug-in) you could have a very high quality unit for under \$300. At 50 lbs, it wouldn't be portable, however (but it will look VERY sophisticated).

Books

I'm not going to try to teach you everything about ICs and signals and so forth. If you're already comfortable reading schematics, using a logic probe and scope, great. If not, here's where you get to take a bit of a detour.

Trying to diagnose a system using wiggling lines without knowing anything about gates or signals or microprocessors would be like learning assembly language without understanding binary, octal, or hexadecimal.

For those of you near a technical college, the best option would be to find an evening course (or just a text book) on microcomputer theory. If you're not close to a college, Heathkit has some very good microprocessor courses. They're on the expensive side but quite well done. It's best (but not necessary) to get a course for the processor you'll be learning.

A few other books that'll get you started (but aren't as complete as I'd like) are *Inside the IBM PC* (available everywhere) and *IBM PC Troubleshooting and Repair* by Robert C. Brenner (Sams). The "Troubleshooting and Repair" book is aimed at getting you up and servicing immediately, but its use of service trees (replace U whats-its-name, if that doesn't cure the problem replace U whose-its-face...) can leave you more confused than enlightened if the suggestions don't work. Avoid that part of the manual if possible. If you have a Z80 system, you might pick up *Z80 Design Projects* by William Bardon Jr (Sams). This book comes recommended by Oron Lott.

By next issue you'll have a full complement of tools (at least a logic probe and meter), and you'll have picked up a BS in microprocessor theory. Then we'll be ready to poke around inside a system and see what is and isn't there. ■ ■ ■

Excellence

In your job, it depends on having the best tools available at your disposal. With such tools, your productivity increases and your work becomes easier.

Wisely, you keep a sharp eye open for products using the latest technology...Those truly representing the state of the art.

You have now located the source of advanced debugging technology for PC-DOS and CP/M-80. More powerful debugging software is not available anywhere...at any price. Yet the cost is affordable to even the smallest budget.

DSD-80...Absolutely the most powerful and easiest to use debugger for CP/M-80. Full screen symbolic design now includes a back tracing capability. Only 125.00

DSD-86...New and innovative design combines the most sophisticated user interface with the most flexible display to create a new generation of debugging technology for the IBM PC. Only 69.95

SoftAdvances

P.O. Box 49473 - Austin, Texas 78765 - (512) 478-4763

Visa & Mastercard Accepted. Please include 4.00 for shipping and handling.

Toll Free Order Line
Order Line: 1-800-223-EPIC
9AM-6PM CST

EPIC SALES

132 Walnut-Plano Center
Garland, Texas 75042

EPIC SALES WINCHESTER SUB SYSTEMS

Featuring Drives By Seagate

XEROX 820,820-II,BIGBOARD I,KAYPRO II/IV

- Supports CPM 2.2 operating system
- Host board plugs into Z-80 socket
- WD 1002 controller board external in cabinet
- Boot CPM from 8" or 5 1/4" floppy
- Easy installation

Complete documentation package is available for \$35.00.
 This package includes all manuals and software on a diskette (if required).
 The \$35.00 is refundable with the purchase of a Winchester sub-system
 from EPIC SALES.

S-100,Z-100

- Supports CPM 2.2 operating system
- Host board plugs into S-100 BUSS
- XEBEC S-1410 controller board external in cabinet
- Boot CPM from floppy (8")
- Easy installation

Fast & Efficient Service!

All orders are expedited as a matter of our regular policy.
 We are proud to have achieved our goal of shipping most orders
 within 24 hrs. of receipt.

Type of System	Seagate ST-212 1/2 High (Low Power)			Seagate ST-225 1/2 High		Erwin 110 Tape Backup	
	10 MEG Internal	10 MEG External	Dual 10 MEG External	20 MEG Internal	20 MEG External	Internal	External
IBM-PC & COMPATIBLES	\$599	\$799	\$1199	\$699	\$899	\$595	\$695
TI PC & COLUMBIA	\$699	\$899	\$1299	\$799	\$999	\$595	\$695
APPLE	N/A	\$949	\$1349	N/A	\$1049	N/A	N/A
XEROX 820& KAYPRO	N/A	\$899	\$1399	N/A	\$999	N/A	N/A
S-100	N/A	\$999	\$1599	N/A	N/A	N/A	N/A

SATISFACTION GUARANTEED: Order your EPIC Winchester System now - test it for 30 days, if not satisfied, return for immediate refund

1-YEAR FULL WARRANTY ● Every unit is thoroughly tested before shipping and our warranty includes both parts and labor.

TECH SERVICE HOTLINE ● If you have a question or a problem Call our service technicians.

8" FLOPPY DRIVES

Siemens FD200-8
 DSDD 8" Floppy Drives
 (Shugart 850 Compatible)

\$185 NEW

8" DRIVE SUB-SYSTEM

Includes: (2) FDD - 200-8
 DSDD Drives, Vertical
 Case and Power Supply

\$569 NEW

XEROX 820-I & II

Disk Drives Cable 5 1/4".....\$20
 Disk Drive Cable 8".....\$30
 8/5 Multiplexer Box.....\$149
 (Run 8" & 5 1/4" drives simultaneously)

EPIC SALES SPECIAL

5 1/4" SEAGATE

WINCHESTER DRIVES

ST-212 10 MEG 1/2 HIGH **\$399**
 ST-225 20 MEG 1/2 HIGH **\$499**

WINCHESTER CONTROLLER & HOST BOARDS

DTC 5150BX IBM Winchester Controller \$249
 ZOBEX TI PC Winchester Controller \$350
 XEBEC 1410 Controller..... \$249
 Software \$35
 Western Digital 1002-05HDO..... \$295

5 1/4" HALF HIGH DRIVES

For IBM PC, TI PC, Xerox 820,
 and others
 Epson SD-521
 (SA-455 Compatible)

40 Track, 48 TPI
 DSDD
\$119 NEW

Case & P/S for (2) 5 1/4"
 Half High Drives.

\$72

POWER SUPPLIES

IBM-PC Replacement 135 Watt

\$135

Boshert XL81-5630R
 81 Watts Maximum
 + 5/8A, +12/5A, -12/1A

\$39

Tech. Info. (214)272-5724 → **TOLL FREE ORDER LINE! 1-800-223-3742**

TERMS: Master Card, Visa, American Express add 3% service charge. Allow 10 days for checks. \$4.00 Minimum UPS charge. \$1.65 for COD. Texas Residents 6 1/8% Sales Tax.

Store Hours:

9-6 M-F...9-1 Sat.

Welcome again to Laine's adventures as Micro C's Turkish bureau chief (alias agent 0086). His investigations of memory theft are detailed in this, his latest *communiqué*. Also be with us next time when his faithful sidekick asks the hair raising question: "Why do the Taiwanese charge \$10 a copy for MS-DOS when haircuts are only \$5?"

So you think you know about culture shock, huh? You don't know a thing about culture shock until you have your morning tea sitting on a hand woven cushion on the floor of a tent in the middle of the Anatolian plain, miles from the nearest electricity or clean water supply, hosted by shepherds dressed in homespun clothing.

But that isn't what really makes you think; it's 10 hours later, after the 45 minute jet flight back to the city, walking down a busy street, passing by ladies dressed in the latest European fashions, with American rock and roll music blaring from the doorways of the shops. Wow.

If you think I've seen a lot of things in the 3 months I've been here, you're right. If you think I still have a lot to see, you're even more right.

I've already seen the centuries old walls of the city of Diyarbakir (at the north end of the Mesopotamian Valley, 100 miles from Syria on the Tigris River), the extreme poverty of Anatolian villages, the mosques and the "covered bazaar" of Istanbul, and the destruction caused by the 1976 earthquake in Lice (Lee-jay). I've eaten in fine restaurants in Istanbul, sat down for a few beers with the local working stiffs in a small bar just across the Bosphorus in Uskudar, and sung along with American folk tunes played by a Turk in a small cafe in Ankara. I haven't yet been to the shores of the Mediterranean or the Black Sea, or climbed to the hilltop castles scattered all across the countryside. There are too many things here for me ever to see all of them, especially when I have to work. Ah well, I guess all I can do is give it my best try.

Geez, fellas, I'm really sorry about the length again, but I've been finding

out so much great stuff about CCP/M that I just HAVE to tell someone (and God knows there's nobody over here I could tell it to). Anyway, I really think that multitasking operating systems need to be covered.

Concurrent CP/M (A Multitasking Operating System)

I've been studying CCP/M and want to talk about it a bit. In particular, I'll show you how to execute a program from within another program, using "shared code" (different processes having separate data, but sharing the same code in memory), how to patch some programs to work better with CCP/M, and also give some tips on hand translating Z80 code to 8086 code. Sorry all you hardware freaks, but I've hit a hardware lull of sorts until I find somebody who is selling a snappy little 80386 kit, or at least get some info about the 386 (unless, that is, you want to hear about all the high technology in IBM clones from Taiwan...).

Executive Sweet

I remember saying some time ago that the thing I missed most about CP/M-80 was my trusty old ZCPR2 with its multiple command lines, named directories, command search paths, etc. Even though I did find a ZCPR-like CCP replacement for CP/M-86 on SIG/M awhile back, it had some very serious (pronounced: crashable) bugs, and the guy didn't even have the guts to include the source code. At least he admitted all the bugs (well, almost all the bugs...).

Soon after I began using CCP/M daily and reading the manuals, I came across a system call with the name "P_CLI" (Process, Command Line Interpreter). It receives the address of a command line, just like what you would type at the CCP (TMP in CCP/M jargon) prompt, and creates a new process to execute the command.

Hey, this is pretty cool! I can write a sophisticated user interface in a high-level language and make one simple call to execute a program; none of this "load it yourself" and "parse it yourself" stuff. I plan to spend a little time

in the next few issues developing my great user interface (in Turbo Pascal, since everybody seems to have it) while explaining some of the more advanced features of CCP/M.

To those of you diehard assembly language freaks: bear with me. Sure, I know the thing would be much smaller and faster if I wrote it in assembly language, but this is supposed to show concepts. (Don't lose faith Earl, deep down I really DO love assembly.)

By the time we finish with this little jewel, I hope to have the following features implemented: multiple commands on one line, extended command line editing, recall and editing of previously typed commands, I/O redirection transparent to the application program being run, command piping, and named directories (named user areas, actually). As we go along, I hope to have the latest version of the shell available on the Micro C Bulletin board (in the Slicer directory).

Some of the above listed features are just "grunt work," but others (I/O redirection, for instance) require real insight into the internals of CCP/M. (Yes, it'll all be done through BDOS calls in a hardware independent fashion.) Before we get into all the real heady stuff, though, let me explain the basic elements of executing a program (easy) and returning to the calling program when you're finished (also easy, but not quite as obvious).

First things first. We need a name. Let's call this little fella SHELL.

Editor's note: SHELL.PAS, CCPM.INC, EDITSTR.INC, and SHORT.PAS are all found on the Micro C bulletin board. Only SHORT.PAS is printed in this issue of the magazine.

The Executionist

Second things second. Given a string of characters, how do we tell CCP/M to execute it? Like I said, this is the easy part. The P_CLI system call expects the DS:DX register pair to point to a command line "framed" in 0 bytes, e.g.:

```
0,'your command here',0
```

(continued next page)

Since Turbo Pascal keeps its strings in a slightly different format (a length byte followed by the string), this calls for a small conversion. First we'll use the length byte to put the 0 at the end of the string, then we'll zero out the length byte. So, our first pass at an "Exec" procedure looks like this:

```
PROCEDURE Exec (cmd : string128)
var reg : BDOSreg;

{ see Figure 1 }
{ for data types }
begin
cmd[length(cmd)+1] := chr(0);
cmd[0] := chr(0);
reg.CX := P_CLI;
reg.DS := seg(cmd);
reg.DX := ofs(cmd);
BDOS (reg); { predefined in Turbo }
end; { Exec }
```

And, strangely enough, it works. The only problem is that when the new program finishes executing you return not to the calling program, but to the TMP (Terminal Message Processor). Oops. Another look at the CCP/M Programmer's Reference reveals that in order to be "reconnected," Exec must be the highest priority process requesting the console when the console is released by the called program. So, let's up the priority and try attaching to the console. Instead of just "bdos(reg)," do this:

```
{ see Figure 1 }
{ for definitions of }
{ SetPriority and AttachDefVC }
SetPriority($C5); { we're important }
BDOS(reg);
AttachDefVC; { wait for console }
SetPriority($C8);
{ we're insignificant }
```

Since TMPs always have a priority of \$C6 (if you don't believe me, just do "systat[p]"), and since the TMP is always waiting to reattach to the console when a program is running, we have to set our priority one better (lower is better, like golf) and put in a request to reattach the default VC (virtual console). Now Exec gets the console back instead of the TMP. Oh, happy day.

Errors

One more problem though: what if the specified command doesn't exist?

What if there is an error loading the program? P_CLI kindly returns an error value in AX and BX. If AX is 0 then there was no error; if AX is not 0 then BX contains an error code as specified in table 6.5 of the Programmer's Reference. We could just look at the error within Exec and print out an error message right then, but it would be more flexible if we returned an error code to the caller. That way the calling routine can do something wild like automatically trying to execute ("submit "+cmd) or ("sim80 "+cmd). It's easiest to make Exec into a function that returns the error number (or 0 for "no error"). See Figure 1 — SHORT.PAS for the completed example.

Now all you have to do to execute another program from within your own application is make the following statement:

```
Errnum := Exec(commandline);
```

The command line could be hard-coded into the program, input from the terminal with a "readln," entered from a data screen similar to the database program included with Turbo Toolbox, or even chosen from an array of commands after getting a menu selection from the user. It's all up to you; do whatever your little heart desires.

Something Useful, Something New...

Now we have our first building block. To use it, let's just put it in a loop that prints a prompt, gets an input line with "readln," and executes the command, stopping when the command "EXIT" is given (sound familiar to all you lemming-DOS users?). While we're at it, we may as well parse through the input line looking for a separator ("!") to delimit multiple commands on the same line. Also, there's no harm in putting a nice little line input routine in (EDITSTR.INC on Micro C bulletin board) that allows inserting and deleting characters and moving around within the line. No harm in allowing recall of the last command we typed either. Figure 1 has the resulting "Phase 1" of our project.

Next time, we'll add named directo-

ries to this structure (simply by adding an array[0..15] for each drive) and hopefully implement I/O redirection by having SHELL intercept all BDOS interrupts of its "children." If you want to study ahead, see the hint on pages 6-153 of the CCP/M Programmer's Reference.

Polite Processes (or "Shared Code")

One very useful feature of CCP/M is the ability of multiple processes to share a common code segment (only if they're running the same program, of course!). For instance, this means if you have four users running the same 60K accounting program, you can save 180K of memory for other uses (or possibly "shoehorn" everything into a smaller machine).

It is extremely easy to implement shared code in an operating system running on the 86 family. Since code and data segments are specified by separate registers, all you need to do is set the same code address in the process descriptors for all the sharing processes while allocating different data segments for each.

And everyone keeps complaining about a segmented architecture! I'd like to see shared code implemented as easily on a base level 68000 system (one without external MMU units). The 68K's PC relative addressing is OK for making code relocatable, but it just doesn't cut it when you want to share code segments between processes (since the position of data relative to code is fixed).

After I read the section of the manual on shared code, I decided I would try it. I set the "shared" attribute of a little database program I wrote in Turbo and called it up on two screens at the same time, then did a "systat[m]" on a third console to look at memory allocations. Surprise, surprise! It showed four allocations for FILE (the program in question) and two of them were at the same address. I saved 40K of memory in about five seconds.

After my great success, I started thinking about what a shared code segment implies. First: any variables stored in the code segment are accessible to all users. Second: code overlays are strictly taboo, since it would be

possible for two processes to request different program code in the same overlay area at the same time, and somebody would be sure to execute the last half of somebody else's instruction and wind up wiping 9 months of accounting data off the disk in less time than it takes to say "illegal instruction trap."

With this sobering thought in my mind, I started up Turbo and took all the OVERLAY PROCEDURE defini-

tions out. Whew! That was close. (Of course, I had to try crashing it first, just for fun. Sure enough, it worked.)

The final word on running Turbo programs in shared code mode: DON'T use overlays, and be careful about using typed constants as initialized static variables (typed constants are stored in the code segment). Other than that, there seems to be no problem.

You shouldn't have any problem

with programs written in other high level languages. Just look out for the same problems as above. For assembly language programs however, don't even attempt to use it as shared code unless you wrote it yourself, or the vendor specifically says it will work.

Sluggo Pascal

One of the first things I noticed when Ron (my cellmate) and I started using the Slice as a 2-user system was that whenever we were both running Turbo, everything slowed down about 75 percent. I worried over the problem for awhile and finally (with a little cerebral stirring from Ron) came up with the solution. The whole problem occurs because whenever the Turbo editor (and any program compiled with Turbo) is waiting for terminal input, it's sitting (actually not sitting, but running around in circles) in a hardcoded loop checking the console status over and over until it successfully gets a character. That was fine for normal CP/M (it was the only way possible, matter of fact), but now the CPU DOES have something better to do when I'm just staring at the screen.

I knew that Turbo was getting input with the following routine:

```

INPUT:  MOV     CL,6
        MOV     DL,0FFh
        INT     22h
        JZ      INPUT
        RET

```

So I turned on good old DDT and sent it looking for all occurrences of "MOV DL,0FFh" in TURBO.COMD. There it was, sitting down in the runtime library all nicely modularized and done only once (the editor and the application program use the same runtime library). All I needed to do was change the 0FFh (check for key typed and return the key or 0 if no key ready) to 0FDh (suspend process until key typed). Simple:

```

A>ddt
-rturbo.cmd
-s290
FF fd
CD .
-wturbo.cmd
~^C

```

(continued next page)

Figure 1 - SHORT.PAS

```

TYPE
  string128 = string[128];

  BDOSRegisters = record
    AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : integer;
  end; { BDOSRegisters }

{-----}
FUNCTION GetVC : integer; { return this process's console # }

VAR reg : BDOSRegisters;

begin
  reg.CX := $99;
  BDOS (reg);
  GetVC := lo(reg.AX);
end; { GetVC }

{-----}
PROCEDURE AttachDefVC; { wait for default console & attach }

VAR reg : BDOSRegisters;

begin
  reg.CX := $92;
  BDOS (reg);
end; { AttachDefVC }

{-----}
PROCEDURE SetPriority (priority : integer);

VAR reg : BDOSRegisters;

begin
  reg.DX := priority;
  reg.CX := $91;
  BDOS (reg);
end; { SetPriority }

{-----}
FUNCTION Exec ( command : string128 ) : integer;

VAR reg : BDOSRegisters;

begin
  command[length(command)+1] := chr(0);
  command[0] := chr(0);
  reg.DS := seg(command);
  reg.DX := ofs(command);
  reg.CX := $96; { P_CLT system call }
  SetPriority($C5); { assure we return here instead of to TMP }
  BDOS (reg); { execute the process }
  AttachDefVC; { reattach the console while priority is high }
  SetPriority($C8);
  IF (reg.AX = 0) THEN
    Exec := 0
  ELSE
    Exec := reg.CX;
end; { Exec }

{-----}

```

(continued from page 39)

It worked! I could run four copies of Turbo (not with shared code, though, sorry to say) and it was just as fast as a single copy. Later I booted up standard CP/M to do something and by chance ran Turbo. BAM!! Where the heck did all those little triangles on the screen come from??? Guess I forgot that standard CP/M doesn't support subfunction FD of function 6 (raw console I/O). It thinks I want to output the character with value FDh. Back to square two (but at least not back to square one).

So, now I have to make a patch that

will install itself at runtime, but only after checking the CP/M version number (with function 12) to make sure the patch is really needed. Simple patch, just have to find some free memory. Looks like there's a bit at the beginning of the file just after the copyright notice. OK, here goes:

```
A>ddt
-rturbo.cmd
-a80
0080 jmp B0      ;was 'jmp 2700'
0082 .
-aB0
00B0 mov c1,0C
00B2 int E0
00B4 cmp al,31
00B6 jb C1
00B8 cs:mov byte[290],FD
00BE jmp 2700
00C1 cs:mov byte[290],FF
00C7 jmp 2700
00CA .
-wturbo.cmd
```

You may wonder why I bother to put the FF byte in if it's already there. Well, you remember I said that programs written in Turbo use the same runtime library? Now, think about what happens when you run Turbo under CCP/M to compile a program that may later be run with CP/M. Seems our little application program will come with a patched library, huh?

By the way, you may run into many programs that have this input problem. As long as they are well modularized you can use the same idea to patch them.

Cutting Turbo Down To Size

Ever see "Out of memory" on your screen? Kind of makes you want to bash the tube in with your forehead, doesn't it? Sometimes it happens because you don't have much memory, but when you have 512K it usually happens because you have a couple of real memory hogs running.

I hate to keep beating on Turbo, but it happens to be one of these hogs, even though it doesn't need to be. Turbo automatically tries to give itself 256K of data space (plus 35K of code) when it starts. Running two copies of Turbo doesn't even leave enough memory for a nose-picking program (we'll get into robotics later). The silly thing is that Turbo wouldn't know what to do with more than 128K of memory anyway.

It's easy to fix the problem. Just modify the header record (the first 128 bytes) of the .CMD file. It happens that byte 10 and 11 specify the "Maximum Paragraphs for Data." As delivered, it asks for 4000 paragraphs (256K). Just change byte 11 from 40 to 10 and it will only request 64K. I've been using this modified Turbo for over a month now and haven't yet had any problems compiling or running even my biggest programs.

Programs compiled with Turbo are hogs, too. This can be changed from the "Compiler Options" menu. I usually make the maximum 1000 paragraphs for large programs, and I put the minimum and maximum down to 100 for small utilities.



Z SETS YOU FREE!

Z Operating System, an 8-bit OS that flies! Optimized HD64180/Z80 assembly language code — full software development system with proven linkable libraries of productive subroutines — relocating (ROM and RAM) macro assembler, linker, librarian, cross-reference table generator, debuggers, translators and disassemblers — ready to free you!

High performance and flexibility! Productivity results from dynamically customized OS environments, matching operator, tasks and machine.

Real-time control kernel option allows quick software development for industrial control applications, other tools and utilities for office desk-top personal computing functions, local area networks to Ethernet, AppleTalk, Omninet, ArcNet, PC-Net (Sytek) — from micro to mainframe command, control and communications. Distributed processing application programs are easily developed.

- Extreme organizational flexibility, each directory another environment
- Multiple Commands per line
- Aliases (complex series of commands known by simple names) with variable passing
- Named Directories with absolute password security
- Full-screen command line editing with previous command recall and execution
- Shells and Menu Generators, with shell variables
- Command-file search Paths, dynamically alterable
- Screen-oriented file manipulation and automatic archiving and backup
- 512 megabyte file sizes, 8 gigabyte disks handled
- Auto disk reset when changing floppies
- TCAP database handles characteristics of over 50 computers and terminals, more easily added
- Tree-structured online help and documentation subsystem
- 76 syntax-compatible support utilities

Your missing link has been found — Z! Now fly with eagles! Fast response, efficient resource utilization, link to rest of computing world — shop floor to executive suite, micro to corporate mainframe. Call 415/948-3820 for literature.



Echelon, Inc. 101 First Street • Suite 427 • Los Altos, CA 94022 • 415/948-3820



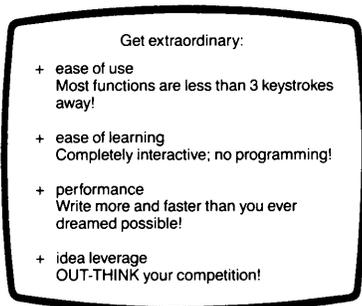
Idea Processing — Pure and Simple.

Announcing the all NEW

OUT-THINK™ outline processor.

See your ideas like you've never seen them before!

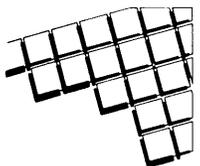
Capture your fleeting ideas. And keep 'em organized with OUT-THINK, the incredible, new outline processor from KAMASOFT. OUT-THINK unleashes the power in your workhorse CP/M computer and turns it into an Idea Processor.



OUT-THINK supports your thinking process and keeps you in touch with your ideas. That's because OUT-THINK works the way your mind works—naturally.



Begin by brainstorming. OUT-THINK lets you jot down ideas quickly, as you think of them. If you want to elaborate, you can add text with full screen editing.



Then develop your ideas using a familiar outline format. Change the structure as easily as you change your mind. Move an idea and all attached text moves with it.

You can keep track of your main line of thought by collapsing the details from view. Then, expand portions of the outline to develop the specifics when you need to. By collapsing and expanding, you can maintain an overview and literally see how your ideas fit together.

And OUT-THINK is fast. Your ideas remain at your fingertips. OUT-THINK can locate and retrieve by keyword—even a misspelled one—in less than a second per outline file.



Once you've entered your ideas, you can get them out with a full complement of formatting, printing and file conversion options (including Wordstar compatibility). From a two-page report to the Great American Novel, OUT-THINK will rotate your headers and footers with page numbers and give you a Table of Contents to boot.



OUT-THINK is brought to you by KAMASOFT. We pioneered outline processing and brought this amazing new technology to CP/M computers. For over 2 years, our

KAMAS programmable outline processor has been widely acclaimed:

"Outline processing might be the next wave in software, and KAMAS is a good place to start."
- Ezra Shapiro, BYTE

"The outline processor is simple, easy to use, and will appeal to all users."
- Mark Renne, INFOWORLD

"If you're a writer using a CP/M based system, get KAMAS. You'll love it. It's that simple."
- Marybeth Highton, FOGHORN

"Considering the quality of the program itself, the documentation, and newsletter, it has to be rated an outstanding value."
- Thomas Tucker, PROFILES

Our new OUT-THINK is simpler and easier to use than KAMAS. Yet, it offers more printing and formatting options, better handling and control, and faster operations. And at \$49.95 your ideas have never had it so good. OUT-THINK is simply spectacular. Try it for 60 days. If you're not completely satisfied, return it for a full refund.

If you create and organize text in the course of your work, you owe it to yourself to get OUT-THINK.*

*But for some people, enough is never enough. Some like the challenge: others need the power. For you, we have the KAMAS Combo Pack. You get OUT-THINK's ease of use coupled with the extra horsepower of the KAMAS programming environment. And both use the same outline files.

Rush me my choices now:



OUT-THINK
the simple and easy to use outline processor:

special introductory \$49⁹⁵*



KAMAS
the programmable powerhouse:

\$99⁹⁵*



KAMAS COMBO PACK
the power and the glory:

\$129⁹⁵*

*Includes S/H U.S. & Canada. Overseas add: \$10 for OUT-THINK; \$30 for KAMAS or COMBO PACK.

KAMAS owners: Send us your master disk and \$25 for OUT-THINK.

VISA MasterCard Payment Enclosed

Card No. Exp. Date

Signature

Name

Address

City State Zip

Telephone

Our software is available for most CP/M computers. Tell us about yours:

Computer Make/Model

Disk Size: 5 1/4" SSDD 5 1/4" DSDD 8" SSDD

Requirements: CP/M 2.2 or above (48K TPA). Minimum 200K of disk storage. 80 column x 24 line screen. (Note: KAMAS requires 56K TPA and Z80 processor.)

Terms: All orders prepaid. PO's accepted when accompanied by payment. Foreign checks must be drawn on US banks in US funds.

MAIL ORDERS TO:

KAMASOFT

2525 SW 224th Ave., Dept. 111
P.O. Box 5549
Aloha, OR 97007

**CALL IN YOUR ORDER:
503-649-3765**

Wordstar is a registered trademark of MicroPro, Inc. Z80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc. KAMAS, KAMASOFT, and OUT-THINK are trademarks of KAMASOFT, Inc.
© 1985 KAMASOFT, Inc.

Automatic 8" Disk Drive Motor Control

By Jim Doolittle

Microtrol Devices
18611 Pebble Place
Nevada City CA 95959

Shutting down 8" drive motors between accesses has long been a thorn for system owners. You are dealing with 110VAC, motors, and timers. It turns out that you face the same problems controlling any 110VAC device from a computer. So, this isn't a bad place to begin.

After five years of listening to the whine of the 8" drive motors (sometimes as long as 15 or 20 minutes between accesses) I decided it was time to do something about it. What resulted was a simple unit to automatically turn these little monsters on and off.

Theory Of Operation

Not knowing how long I might have my present system, I decided to make the unit completely computer independent. Since the 50 conductor disk drive bus is standard for all drives and computers, this was the logical place to look for a signal to control the unit. The only signals that would selectively control each 8" drive are the drive select lines #0 thru 3 located on signal pins #26, #28, #30, and #32 of the ANSI 50 conductor bus.

Next I had to figure out how to use this signal. Some disk drive controllers constantly poll each drive and report status back to the computer. This

means that every millisecond or so, each drive select line will be pulsed to its active low condition. If we used drive select lines to directly control the motor switch, the motor would be turned on and off every millisecond when inactive. Figure 1 shows how, by inverting the drive select signal (U2A) and utilizing a 1.5 ms missing pulse detector (U1A), the output (U1A-#5) will remain high (inactive) during the period when the drive select line is inactive and also during the polling pulse period.

T1, by the way, discharges C1, the timing capacitor, when the drive select line is inactive. During the polling pulse period, T1 is turned off and C1 begins to charge through R2. But the polling pulse is a shorter duration than the C1,R2 time constant, preventing the timer from resetting, and changing the output on U1A-#5 to low (active). When the drive select line is active (low), the U1A times out in 1.5 ms and U1A-#5 goes low.

The next section is a 60-second resettable timer. When the input (U1B-#8) goes low (active), output U1B-#9 goes high (active). At the same time T2 is turned on preventing timer capacitor C4 from charging. When U1B-#8 goes high (inactive) T2 is turned off and C4 starts to charge through R3. Approximately 60 seconds later, U1B-

#9 goes low. Therefore, the disk drive motor turns on within 1.5 ms of the drive select line becoming active, and turns off 60 seconds after the last disk access.

One of the most important functions of this unit is to isolate the 115 volt AC line from the low voltage power supplies and the computer. To accomplish this, I used an optically isolated triac driver (Q1) to drive the motor switching triac (Q2).

Inductive loads (the drive motor, in this case) present a problem both for the triac and the triac driver because the voltage and current are not in phase with each other. Since the triac turns off at zero current, it may be trying to turn off when the applied current is zero but the applied voltage is high. This appears to the triac as a sudden rise in applied voltage, which turns on the triac if the rate of rise exceeds the commutating dv/dt of the triac or the static dv/dt of the triac driver.

The solution to this problem is provided by using "snubber" networks (R7,C5 and R8,C6) to reduce the rate of voltage rise seen by the device. The triac snubber (R8,C6) in this case may be overkill, but in most instances the snubber will adequately protect the triac. Since the triac snubber is dependent on the triac used and the load,

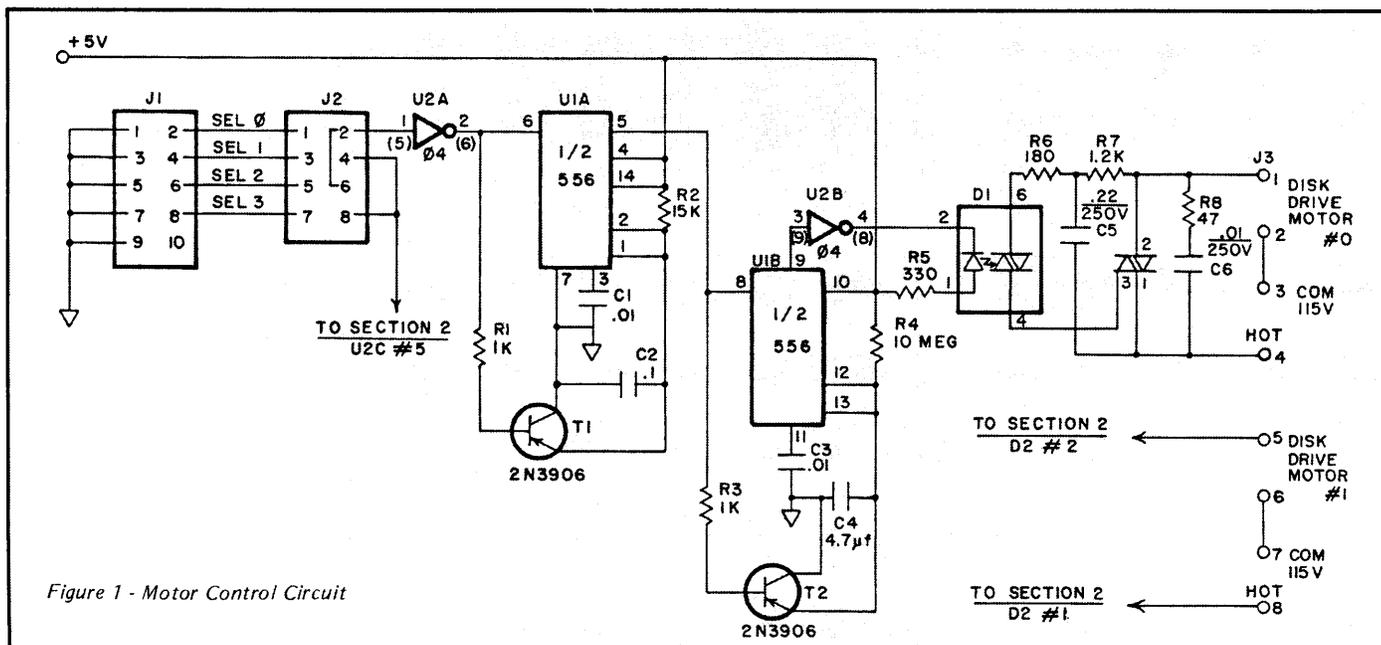


Figure 1 - Motor Control Circuit

R8,C6 was used as a precaution.

Construction

The Radio Shack IC/LSI PC board #276-162 can hold two controllers. I soldered the DIP sockets and components directly to the board, connecting pins with wire-wrap wire.

You can use a protected 10 position header (#R202ND and #R8300ND — Digi-Key) or a 16-pin DIP socket and matching DIP header (#276-1980 — Radio Shack) for the input jack and matching socket connector termination (J1).

J2 can be an 8-contact straight dual row male header and shorting jumpers (#929836-01-36 and #929950-00 — Digi-Key). This will allow you to select any two of the four available drive select lines.

For the output (J3), use a dual row 8 position barrier strip (#274-670 — Radio Shack) or a multi-conductor nylon Quick Disconnect plug and socket (#274-239 and 274-229 — Radio Shack).

The rest of the construction is straightforward. Component placement isn't critical, but try to allow as much space as possible between low voltage and high voltage components and wiring.

Usually there's enough room behind the drives and next to the power supply to mount the PC board. If not,

use a separate enclosure. If the enclosure is metal, be sure the 115V line ground is connected to the enclosure. This is usually a green wire.

The input from the 50-conductor bus is accomplished with a 50-pin male insulation displacement connector and socket (#M50 and #S50 — Jameco) and a 9-conductor piece of 28 gauge flat stranded ribbon cable (#171-9 — Jameco). Make sure that the #1 conductor goes to pin #25 on the 50 contact male connector and pin #1 on the 10 contact input plug. Install the 50-contact socket at any convenient location on the 50-conductor flat cable running between the computer and the 8" drives.

Be careful hooking up the AC power lines and motor lines. First of all, keep in mind that 115 volts can be just as deadly as 115 kilovolts (though not as messy). The old "one hand in the pocket" routine isn't such a bad idea.

First of all, figure out which of the two motor leads is connected to the hot side of the AC line. If you can't determine this visually, a voltmeter test from the incoming ground lead (usually green and tied to the chassis) to both of the other AC leads (ahead of the on-off switch, if any) will give the answer. The hot lead is the one that reads about 115V. The other will read about 0V and is the common.

The hot lead should be disconnected

from the AC hot lead and connected to the motor control line on the unit. Connect the AC hot and common leads to their appropriate terminals on the unit. During the connecting process all units must be disconnected from the AC power source.

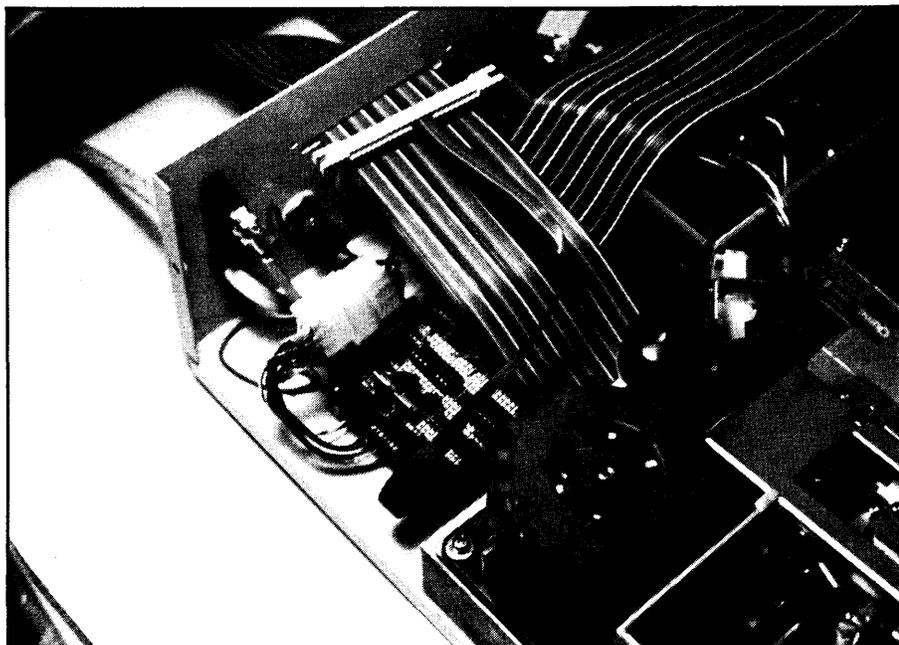
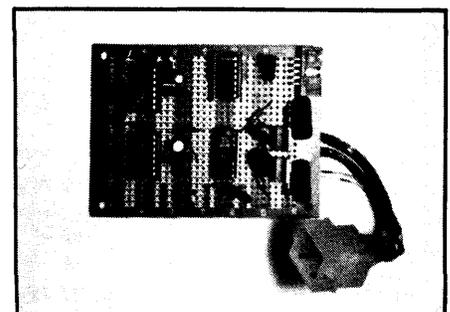
Before installing the EICs in the DIP sockets, check across the +5 and -5 volt terminals with an ohmmeter; it should read greater than 1000 ohms. If not, look for shorts in the 5V wiring. The 5V leads can now be connected to the disk drive power supply. Before connecting the leads, determine which is the +V and which is the -5V (ground) connection. Now install the ICs in their sockets.

Testing The Controller

Turn on the 5V power supply (not the 115V AC!). Apply 5 volts in series with a 4700 ohm resistor to U2A-#1 (and leave it connected). U1A-#5 should read high (5V) and U2B-#4 should read high. This is the normally inactive condition. By grounding (-5V) U2A-#1 and U2B-#4 should read low (<1V). This is the active or "motor on" condition.

If everything checks out up to this point, it's time to connect the 115V AC line to the terminals hot and common. Connect a 100 watt light bulb across the motor terminals (J3-#1 and #2). Apply both the 5V and 115V AC power. When U2A-#1 is high the light bulb should be off, and when U2A-#1 is grounded (low) the light bulb should be on. It should stay on for approximately 60 seconds after U2A-#1 goes high.

Now you can connect the signal input from the 50 conductor cable, as previously described, power up, and give it a try.



CPM-80 LIVES ON YOUR PC

CP/Mulator puts a 4mhz 8 bit CPM-80 emulator in your IBM-PC for only \$99.

- Execute 8 bit CP/M programs at 4 mhz. Stop waiting around for those slow software emulators at twice the price of this speed demon!
- CP/Mulator makes a fantastic CP/M development environment and allows you to use all your CPM-80 tools, and the IBM-PC tools too!
- Why write off that expensive package that you bought for CPM-80 just because you upgrade to a IBM-PC? Apply the \$99 solution and keep it working. No retraining or conversion blues!
- CP/Mulator includes the NEC uPD70108 processor which increases the speed of your IBM-PC by at least 10%. 100% compatible with existing IBM-PC software and 8087 compatible.

PURCHASE THE SOFTWARE ALONE FOR ONLY \$69!!

Source Information
P.O. Box 2974
Warminster, PA 18974
Phone (215) 628-4719

NEW!!

SPOOLER II 512 K Buffer/Converter

Serial/Parallel and Parallel/Serial Conversion

Protocol Conversion & Data Switch

Buffer Outputs to Printers and Plotters

Buffer Inputs from Instruments for Laboratory Applications

Two digit LED display monitors status

New SPOOLER II is an enhanced Print Buffer and Interface/Protocol Converter based on the Hitachi HD64180, capable of data transfers up to 38.4 KBaud serial and 75 KBaud parallel. It provides 520K of buffer space — perfect for buffering giant printed documents, large files for plotters, and data gathering for laboratory applications. Compatible with PC's, MAC's, and all other computers and peripherals with serial or parallel interfaces.

I/O connectors mount directly to 5¼ by 7½ inch board. Optional enclosure available. Includes complete documentation and application notes.

SPOOLER II board, with Power Supply, Wired & Tested \$249.00

SPOOLER II board, with Power Supply, Complete Kit 229.00

Optional Enclosure 49.00

Prepaid units shipped via UPS Blue at no charge. COD orders also accepted.

No credit cards at this time. CA orders add 6½% Sales Tax.

To order or for further information call or write:

L-BAND SYSTEMS

1037 E. Lemon Ave. • Monrovia, CA 91016
(818) 357-0566

LEAP INTO A NEW DIMENSION



WITH AZTEC CII

Full implementation of "C" with standard floating point, library, and I/O subroutines. UNIX VER 7 compatible. Produces relocatable 8080 (optional 0Z80) assembler code. Relocating assembler and linker supplied with package or use Microsoft M80 and L80, SID/ZSID debugger interface. FAST COMPILATION AND EXECUTION.

AZTEC CII FOR CP/M \$199

(Special price for Micro C subscribers \$149)

MANX

software systems

Box 55, Shrewsbury, N.J. 07701
(201) 780-4004



Also available for Apple DOS, HDOS, CP/M-86, PC-DOS

uniforth™

**ANNOUNCING OUR
NEW PRODUCTS**

Sampler (\$35) and/or

Professional Systems (from \$175) for...

Apple ProDos

Amiga

Micro VAX

Macintosh

IBM PC/AT

DEC Falcon

UNIFORTH is a full-featured, floating-point, FORTH-83 Standard language available for most micro and mini computers.

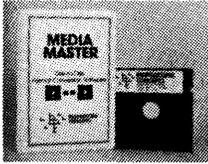
Call or write for our *free catalog* or use our Bulletin Board (614-459-7736, 300/1200 Baud) for more detailed information

UNIFIED SOFTWARE SYSTEMS
P.O. Box 21294, Columbus, OH 43221-0294
(614) 459-7735

"MAKE YOUR IBM-PC KAYPRO COMPATIBLE"

Intersecting Concepts Announces
3 Solutions To Solve Your
Computer Incompatibility

But will it work on my computer? Yes! Finally, there are three easy ways to exchange information, transfer files, and run CP/M software on MS-DOS machines.



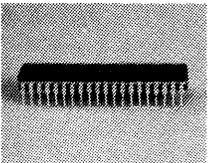
1. **MEDIA MASTER** is our direct disk-to-disk format conversion program. Already an accepted industry standard, this \$39.95 program uses simple screen prompts that lets you

read, write and format up to 70 different 5¼" diskettes from CP/M, MS-DOS and PC-DOS operating systems. So if you work on a IBM PCompatible at the office, but use a CP/M computer at home, now you can easily transfer files that would otherwise be "foreign" to your computer's operating system.



2. **MEDIA MASTER PLUS** goes one step further by converting 8-bit CP/M software to run on 16-bit MS-DOS and PC-DOS machines. This newly released \$59.95 product

combines our IBM-PC version of Media Master with ZP/EM, a powerful new emulation program. The results are amazing: CP/M programs using 8080 instructions and data can be transferred from popular computers like Osborne, Kaypro and Zenith to run on MS-DOS and PC-DOS machines!



3. **ACCELER-8/16** is also new and dramatically improves the performance of Media Master Plus by tailoring the CP/M emulation around a NEC V20 microchip. This chip simply

replaces the 8088 processor in your MS-DOS computer. Once installed, it'll run your CP/M and MS-DOS software much faster. (Speed improvements are roughly 15% faster in MS-DOS and 350% faster in CP/M!) With ACCELER-8/16 your MS-DOS computer is now CP/M compatible for only \$99.95! All three solutions save you money by eliminating expensive modems and communications software.

To Order

To order Media Master, Media Master Plus, or Acceler-8/16, call 800-824-7888, and ask for operator 251. For additional product and upgrade information contact:



Intersecting Concepts, Inc.,
4573 Heatherglan Ct., Moorpark, CA 93021
or call 805-529-5073.

COMBINATION 1MB RAM DISK AND 64K PRINTER BUFFER \$445

ELECTRONIC RAM DISK SPECIAL MEMORY
OFFER, TWO BANKS OF MEMORY FOR THE
PRICE OF ONE!!! THIS IS A LIMITED OFFER.

Electronic RAM Disk and Printer Buffer Combination

— the ultimate add-on for Kaypro I, II, 4, 4-84, 2-84 and 2X Computers. Incredible speed and efficiency are offered using MicroSphere's dual operation Electronic RAM disk and printer buffer. No longer will you need to wait for your Kaypro to slowly finish mundane tasks such as running a printer or waiting for floppy drives to turn when you have better things to do.

Features:

- ★ Can be powered externally ★ 255 directory entries
- ★ Fully buffered data transfer using Z80 PIO
- ★ Compatible with Pro 8 and Plu-Perfect or 5 Mhz speed up kits
- ★ RAM disk memory not lost on cold boot or reset of Kaypro
- ★ Printer buffer sizes: 4K, 32K, 64K ★ 10 Mhz microprocessor speed
- ★ Printer buffer runs independently of RAM disk operation
- ★ Micro Cornucopia Pro 8 Version 3 ROM includes RAM Disk Code
- ★ Installation Kit for Kaypro 1, 2-84 and 2X without clock or modem included at no charge if required ★ Choice of drive A or E
- ★ On 4-84 the standard installation disables either internal modem or real time clock unless you install the adapter board
- ★ All installation materials included in listed price expect for Kaypro 10 Adapter Board

NEW: Kaypro 10 RAM Disk Adapter Board

For Kaypro 10 without internal modem \$ 50.00
Adapter board will restore either internal clock or modem in Kaypro 4-84 or can add additional decoder parallel ports on all Kaypros.

Any RAM Disk listed below can be expanded to 1mb by adding memory chips

Tested RAM Disk without RAM Chips	\$295.00
256 RAM Disk	\$370.00
512 RAM Disk	\$370.00
768 RAM Disk	\$445.00
1 MB RAM Disk	\$445.00

Reduced
RAM Disk
Prices!

Kaypro II to IV Upgrade

... includes 2.5Mhz to 5Mhz speedup, two 390K disk drives (double sided drives) to replace the single sided drives, price does not include shipping \$395.00

TLC LOGO

Easy and exciting language for all ages. TLC LOGO is an exceptionally complete logo with vectors, multiple turtles, full floating point decimals and extremely fast program execution.

Two Versions Offered:

Standard version using only internal graphics of Kaypro 1, 2-84, 4-84, 2X and 10; no hardware additions or modifications required; 16,000 pixel resolution \$99.95

Deluxe version for all Kaypro Computers that have the MicroSphere Color Graphics board. Features 16 colors, 32 sprites, 49,000 pixel resolution and utilities such as screen dump to printer or disk \$129.95

Special: Color Board & Deluxe TLC Logo \$199.95

Improved External Monitor Adapters for Kaypro

All boards offer dual screen operation using unmodified industry standard monitors. External monitor displays original character set; internal screen unaffected by installation or operation.

1. New Kaypro 1, 2-84, 4-84, 2X, 10, requires IBM type external monitor \$49.95
2. Older Kaypro 2 & 4, standard composite video \$49.95

Printer Muffler —

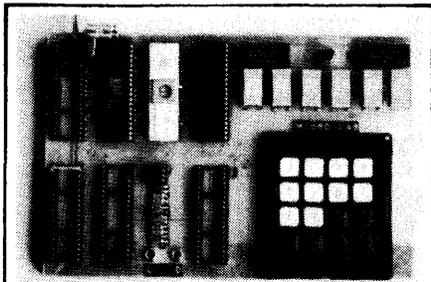
Capable of handling a top feed printer 24" wide, 18" deep and 11" high. Specially designed to accommodate both the tractor feed and single sheet feeder of the Kaypro or Juki 6100 printer. ★ foam lined for maximum sound reduction ★ can be shipped UPS. And your secretary will love it! Price does not include shipping \$149.95

Infocom Adventure Games Call us last, best prices around.

EXPANDED CATALOG NOW AVAILABLE
100's OF SOFTWARE LISTINGS FOR CP/M NOW AVAILABLE

MicroSphere, Inc.
P.O. Box 1221
Bend, Oregon 97709
503-388-1194





***** HOBBYIST *****
MICRO-LAB: A SINGLE BOARD COMPUTER

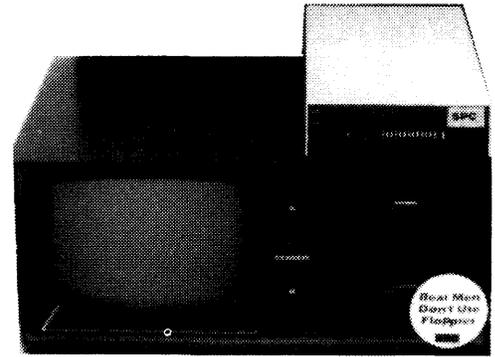
 * CPU-8085-3MHZ RAM-8156 EPROM-8755
 * MEMORY- 4K EPROM/256 BYTES RAM/EXPAN
 * I/O PORTS- 32 PARALLEL LINES/SID-SDD
 * INTERRUPTS- 4 VECTORED INTERRUPTS
 * HEX SYSTEM- KEYBOARD AND DISPLAY
 * EPROM PROGRAMMER- PROGRAMS 8755
 * CASSETTE LOAD AND DUMP
 * INTERFACE-(40 PIN) BUS AND PORTS
 * SUBROUTINES- KEYS/DISP/INT/DEBUG
 * MANUAL WITH DRAWING

 MICRO-LAB A/T-\$179 KIT-\$139

 ORR CORP.
 P.O. BOX 9433, CINCINNATI, OH 45209
 CHECKS, VISA/MC# EXP. DATE
 ADD #3 SHIPPING, OH RES. ADD 5.5% TAX

FOR THE SERIOUS KAYPRO® USER

THE NEW **PRO/FILE**
SERIES OF WINCHESTER
DISK DRIVES FROM SPC



Whether you have a **KAYPRO 2, 4, 10,** or even a **K-16,** the new low **PRO/FILE** hard disk drives from **SPC** are cost-effective and easy to install. A screwdriver is all you need. With up to 34 Megabytes available in a small 2 inch-high cabinet, you can say goodbye to floppy problems forever.

SPC provides next day delivery on all products, a one year warranty, and a friendly approach that says we appreciate your business. Customer references are provided upon request. Larger system capacities up to **120Mb.** are also available, including combination fixed/removable cartridge systems.

Some of the prices above are limited to current stock on hand and are subject to change, so call **SPC** now and let's discuss your application. Ask about our trade-up policy and controller exchange program. Call now... ask for Shelly or Bob.

SYSTEMS PERIPHERALS CONSULTANTS

9747 Business Park Avenue
 San Diego, CA 92131
 (619) 693-8611

*If you own a Pro-8
monitor, this is the product
you've been waiting for!*

MULTICOPY

Save hours of time on your Kaypro with this powerful utility!

- Format, copy and verify 96 tpi disks in under 4 minutes (exact speed depends on type of drives and year of Kaypro)
- Format, copy and verify Kaypro double and single-sided disks in approximately 1 minute and 2 minutes respectively
- Format, copy and verify **any** soft-sector, single or double-sided 5¼" disk format on your Kaypro at comparable speeds (e.g. Osborne SSDD in under a minute). Large library of formats supplied
- Use any combination of Drives A, B, C or D as source or destination drives

(Not currently available for the Kaypro 10)

Please send **MULTICOPY** to:

Name _____

Address _____

Enclosed is **\$39** plus **\$3** shipping
(California residents add 6% sales tax)

Check or money order VISA MasterCard

Number _____ Exp. _____

Signature _____

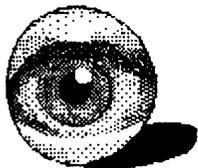
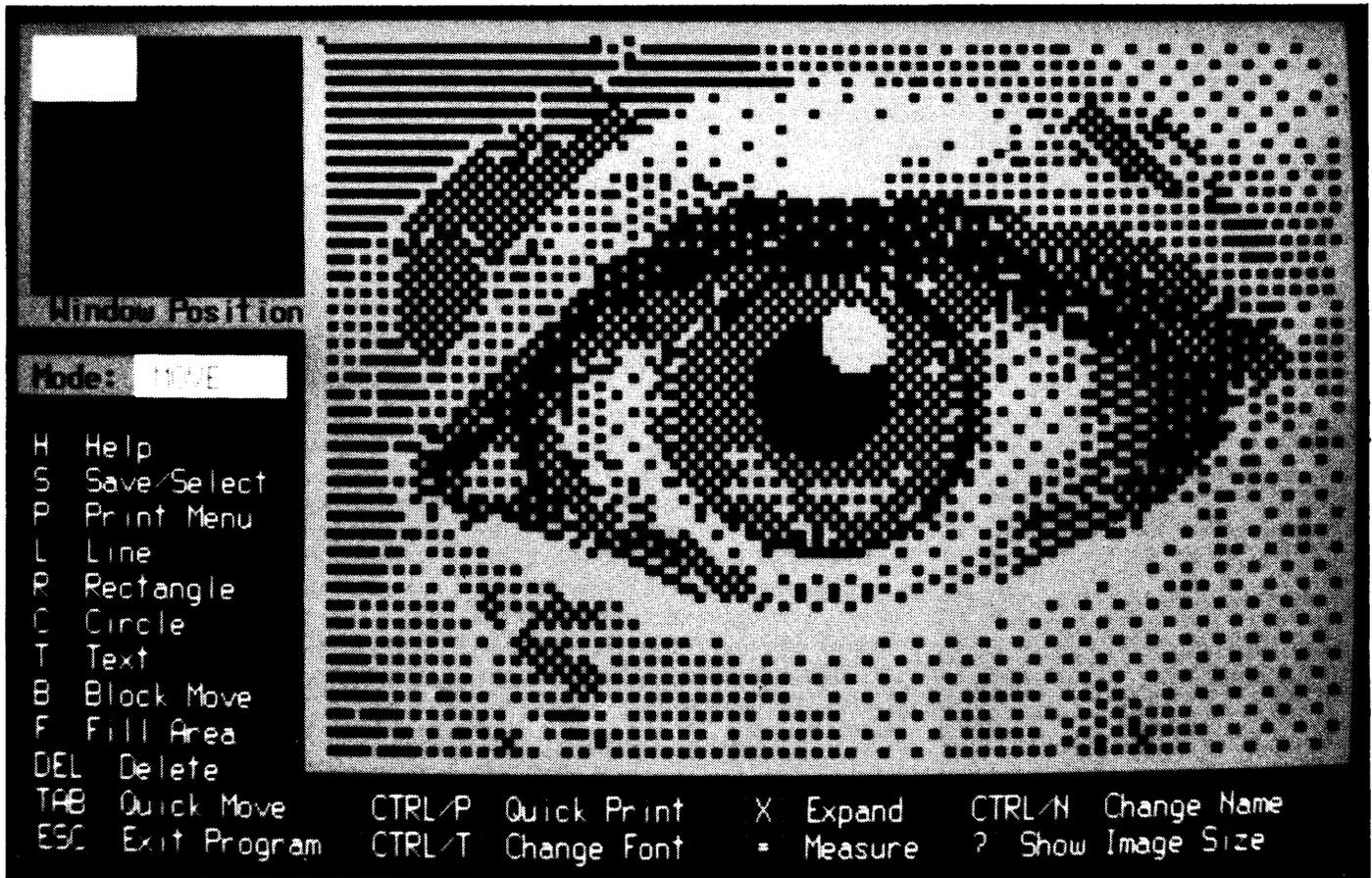
(\$5 shipping charge for orders outside N. America)

CP/M is a registered trademark of Digital Research, Inc.

For further information on **MULTICOPY** and other Kaypro and CP/M software, please write or call:

Plu*Perfect Systems

BOX 1494 • IDYLLWILD, CA 92349 • (714) 659-4432

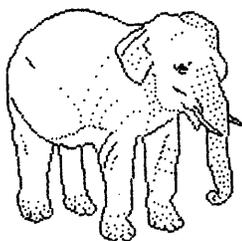


"Kaypro makes a nice computer- too bad it doesn't do graphics"

Make that didn't do graphics. **SCS-Draw** is here, bringing Macintosh-style graphics to the Kaypro computer for only \$59.95.

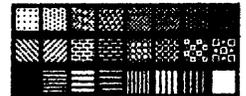
The first true drawing program for Kaypro computers, **SCS-Draw** turns your screen into a sketchpad on which you can draw detailed images like those shown here. These images can be saved on disk and printed on most dot-matrix printers and many letter quality printers.

There are many applications of **SCS-Draw**. For example, you could design and print your business card, company logo, party invitations, banners, or technical illustrations.



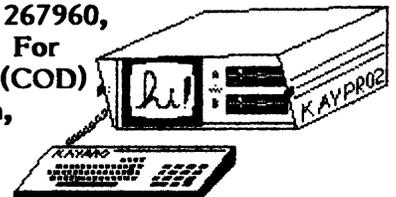
And the best thing about **SCS-Draw** is that it's fun to use -- when was the last time you had some fun with your Kaypro?

SCS-Draw gives you a variety of useful drawing tools to choose from. You can draw a detailed image dot-by-dot, or do a rough sketch with straight lines. Need a compass? **SCS-Draw** can draw circles of any size, wherever you need them.



More advanced features include block moves, pattern filling, and four built-in fonts. And **SCS-Draw's** windowing capability lets you easily draw images many times larger than the Kaypro's screen.

SCS-Draw is available now for all 1984 and 1985 Kaypro CP/M computers. To order your copy, send \$59.95 (check or M.O.) to Second City Software, Box 267960, Chicago, IL 60626. For immediate delivery (COD) or more information, call 312-577-7680.



Kaypro Column

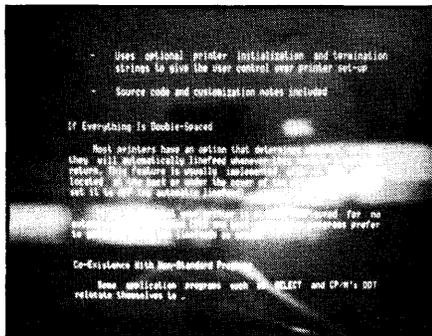
By Charles H. Chandler

80 Boundary Rd
Malden MA 02148

Want a simple Kaypro mod? Something non-technical like a non-glare screen? This is a non-glare screen but it's still pretty technical. Sorry, maybe next time.

Who hasn't complained about unwanted light falling on the computer or monitor screen and washing out contrast, as in Figure 1? Mesh filters, spray-on liquid, and green filters have all been offered as solutions, but often cause reflections of their own.

Figure 1 - Before



An Optical View

My solution uses an optical principle that is not original with me, nor even particularly new. The principle is illustrated in Figure 2.

Picture your eye, at right, looking toward a computer screen (in dashed outline, at left). Next, imagine a curved mirror or reflector (solid line) placed in front of the screen. Finally, suppose that the reflector's curve is such that all light rays coming to your eye must come from a point below and in front of the screen — as shown in the figure. If these conditions are met, then the only thing you can see will be that point. Make that point a black surface (light trap), and black is all you'll see.

This curve is an ellipse, with your eye at one focus and the black point at the other. The mirror can be expanded to an elliptical trough or cylinder, and the point can be expanded to a surface (parallel to the axis of the cylinder). Same result.

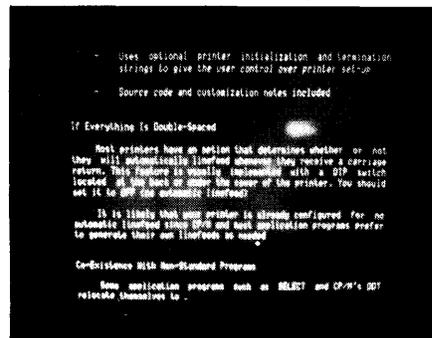
Suppose we replace the mirror with a sheet of transparent color-filter material? We can see the screen through that — and the only thing we can see reflected from it is the light trap, the

black surface. There are no distracting reflections off the filter sheet, because reflected light from any other source just doesn't go to your eye. The black surface doesn't give off any light. As a result, the filter sheet (assuming it is smooth and clean) actually seems to disappear.

Figure 3 shows the desired arrangement. The "light shade" is a box-like structure with the front and back open; you look right through it at the screen. The sides contain grooves or tracks into which you slide your sheet of filter material. These tracks, in turn, serve to shape the filter into the desired elliptical form. The inside of the box is painted flat black — and there's your light shade and light trap, all in one.

The result is shown in Figure 4: greatly reduced stray illumination on the screen, and at the same time absolutely no reflections from the filter sheet.

Figure 4 - After



Construction Notes

The structure must be light and stiff. I used mostly corrugated cardboard, with a few pieces of 3/4" wood cove molding in the upper corners and across the front at the bottom for extra stiffening. Cardboard can be remarkably strong, especially if the directions of the corrugations are crossed in alternate layers.

To lay out the tracks for the ellipse, I drew the CRT screen and the measured location of my eye to full scale on an easel pad. Then I got my ellipse by cut-and-try. There are two constraints in drawing the ellipse: (1) its major axis (A—A in Figure 3) must pass through one focus (F1) and through the eye position (F2); and (2) the ellipse must clear the screen by a

small but comfortable margin.

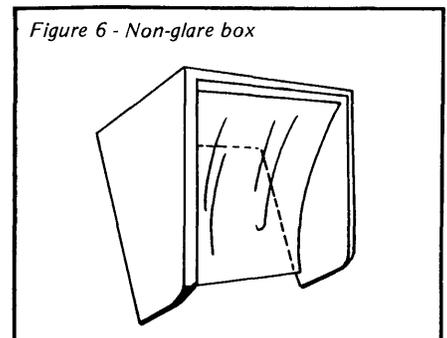
The solid part of the ellipse (in Figure 3) is all that is used for the filter. I have drawn the unused part with dashed lines, just to show how it all goes together. The light trap, of course, must be placed at F1; it is formed by the bottom of the box.

Figure 5 illustrates the classical method of constructing an ellipse, using drawing-pins and a piece of string. The cut-and-try part comes in getting just the right amount of slack in the string so that the ellipse falls in the desired place relative to the screen, as shown in Figure 3.

Having gotten my curve and sketched a suitable shape for the light shield, I pricked the outlines of the shield and curve through the original sheet of paper onto cardboard "raw material" with a scribe. Then I connected the points with pencil lines and cut the material with a smooth, unscratched knife blade in a sabre saw.

The inner sides of the box (containing the elliptical tracks) came in two pieces, fore and aft, for each side. (They were three plies thick, so the tracks were about 3/8" deep.) I established the track width by separating these pieces with 1/32" cardboard shims, and glued them inside the outer sides of the box to form the tracks.

Figure 6 shows the box in an early



stage of construction (without the bottom, but with a piece of filter material temporarily in place). Finishing touches included spray-painting the inside of the box flat black, and the outside with aluminum paint. I sanded the aluminum paint lightly to remove raised grain, burnished it with the

back of the sandpaper, and gave the outside a couple of coats of clear spray lacquer to keep aluminum flakes from getting loose.

Filter Material

Here again we have two requirements: (1) the color of the filter material must roughly match the color of the CRT phosphor (i.e., green or amber); and (2) the material must be of reasonably good optical quality — that is, clear and smooth. I discovered two unlikely but very effective filter sheets. One was the material in a transparent plastic report binder (Ful-Vu E-Z SLIDE, available in stationery stores),

and the other was a material intended for making overhead projector transparencies with a tinted background.

The binder material's color was rather on the blue side, but it worked well in severe cases; it was used in Figure 4. The transparency material was a perfect color match for my green screen.

For computers with color displays, of course, a color filter is out, but a neutral-density (gray) material will be helpful. Whatever material you use, one thickness must be enough. Using two layers will add back-and-forth reflections between them. It is a good idea to wipe the filter sheet with anti-

static treatment before final installation.

Concluding Notes

A word is in order about attaching the light shield and filter holder. My computer is a Kaypro II with a 9" screen. If the front of your monitor is smooth and featureless, you may have to use something like double-sided sticky foam tape for attachment — in which case ultra-lightweight construction is essential. Some support from the bottom will help in situations like this.

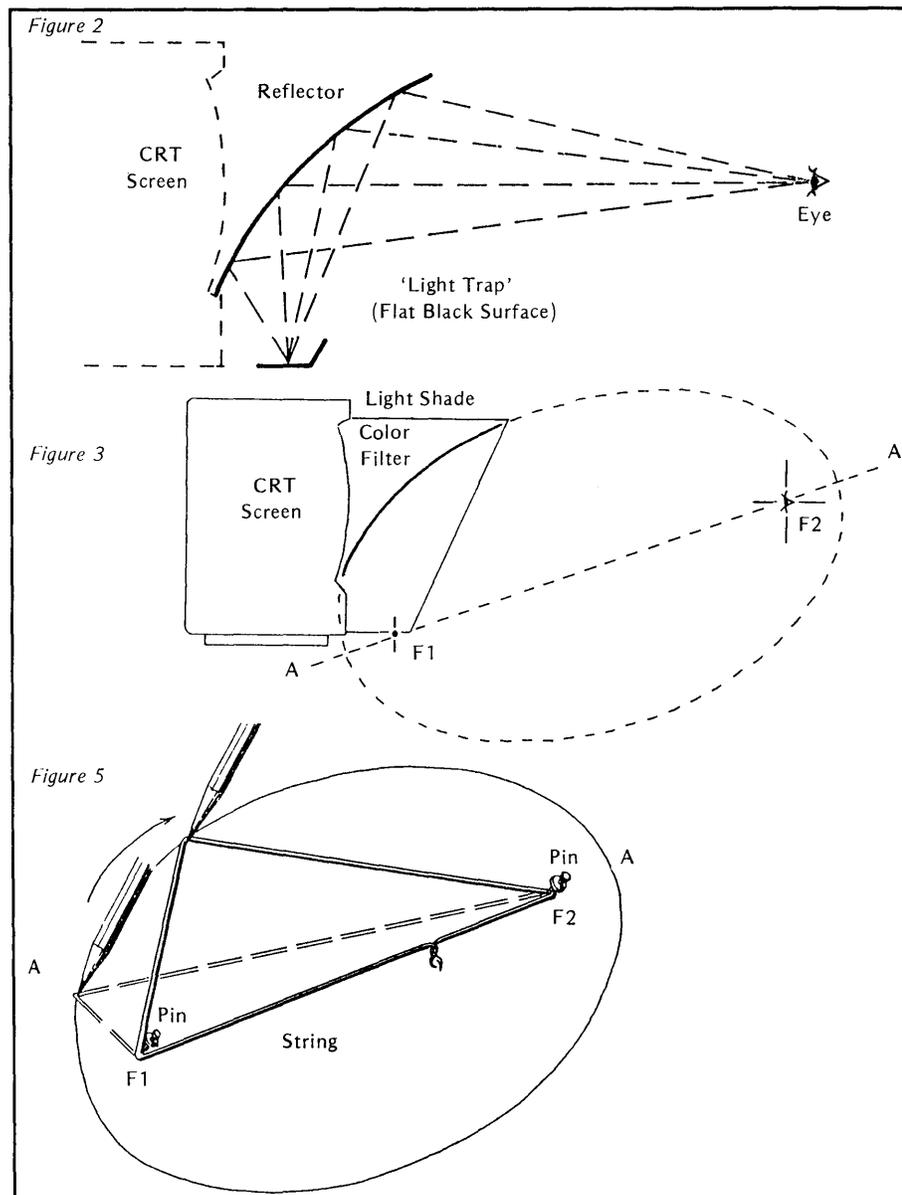
With larger monitors, the box might get a bit bulky and cumbersome. However, if you can arrange the setup so your eye is slightly above the center of the screen, the resulting ellipse will be tilted upward and back, and will not extend as far in front of the screen. You may have to try several layouts (on paper!) in order to achieve the best compromise. It may be helpful to duplicate your ellipse on a separate piece of tracing paper, place it over your layout, and experiment.

Finally, the surprise. Before you do anything else, cut a light-shield out of black paper, cardboard, whatever is handy. Hold it or tape it in front of your screen. You may find that it shades the screen so well you don't need the filter material at all! In that case, just build the box and bless your luck.

EDITOR'S UPDATE: No Garbage On 84 Video

Michael Stocker called recently to say that he had finally rid his screen of occasional garbage characters (even when using the Pro-884 Max in fast video mode). He simply replaced the 6116 static RAM chips in the video section with faster (150ns) parts.

He said he had gotten his new 6116s from Microprocessors Unlimited for \$1.00 each (the video section holds two). The chips are located at U15 and U23 on the 84 Kaypro 1, 2, and 4. See the back cover of this issue for the address of Microprocessors Unlimited.



Pascal Runoff — The Winners' Circle

Well, this is the big issue for all those of you who are wearing Pascal Runoff T-shirts. So cross your fingers and read on.

One hundred and three of you entered your programming projects in our Pascal Runoff. Thanks for the enthusiastic response. It was Christmas around here as we opened package after package of surprises.

Before We Announce The Winners

For those who joined late — the Pascal Runoff was open for entries from June 1 through November 1. We accepted any program written in Turbo Pascal during 1985. We didn't limit program length or subject, and consequently we received a healthy hodgepodge — adventures, sideways printer setter-uppers, 2- and 3-D graphic displays, an overtime scheduler, an integral solver, a pi calculator, a write-your-congressman utility, a screen generator, a poem generator, a bicycle gearing analyzer, a solar intensity measurer, and oodles of useful and not so useful micro-accessories.

So picking the winners was formidable, but fun. Fortunately, lots of folks helped out.

In particular, thanks to Mike Weisert (and his helpers at Borland International), Larry Fogg and Renee Katter (at Micro C), and the rest of the Micro C staff for cataloging, reviewing, testing, and, finally, selecting the winning programs.

The Envelope, Please

The Grand Prize (a 1 megabyte MicroSphere RAM disk or a Definicon 32032-based board) goes to —

Richard Ryall
2323 McBride Lane #42
Santa Rosa CA 95401

for PROBE, a slick screen-oriented disk editor for the Kaypro that works on floppies and hard disks.

PROBE is easy to use and has very readable code and documentation. Works with graphic and non-graphic Kaypros. Great job, Richard.

Second through sixth place winners

each receive their choice of two products from Borland and a \$100 gift certificate from Micro C.

Second Prize to —

Steve Mitton
7662 Fallswood Way
Lorton VA 22079

for RESCUE, a handy utility that locates lost text in memory and writes it to a file. (See Steve's article this issue.)

Third Prize to —

Dennis Sprague
912 4th Ave NW
Great Falls MT 59404

for PLANTER, a bizarre program that calculates the dimensions of a multi-sided wooden planter box and plots it in 3-D. (See Dennis's article this issue.)

Fourth Prize to —

Frances M. Coniglio
1225 Martha Custis Dr
Alexandria VA 22302

for VOCABULARY BUILDER, an educational program to help students increase their foreign language vocabularies.

Fifth Prize to —

Steve Wilcox
1215 South Osceola
Denver CO 80219

for PAMPHLET, which takes a WordStar text file and rearranges the pages in the proper order for a folded pamphlet. Very nice output (this is really a tricky operation, folks).

And Sixth Prize to —

Ernest W. Adams
254 College Ave #F
Palo Alto CA 94306

for BOTTICELLI, an intelligent, educational guessing game. Congratulations!

Honorable Mention

We started with 103 entries and whittled them to 80 and 70, then on down until we reached the "dirty dozen."

Two programs we liked a lot just missed prizes — DRAW, a Kaypro drawing package written by Randy Everton, and ORBIT, a comet/planet plotter written by George Ferguson (see his article this issue).

Wrap Up

Thanks again for your submissions. Everyone who entered by November 1 received a Pascal Runoff T-shirt of Pascal and Bug; if you entered on time and didn't receive one, let us know.

See Figure 1 for an alphabetical listing of programmers and their entries. All Pascal Runoff winning entries are available on Micro C users disks. Call for more information.

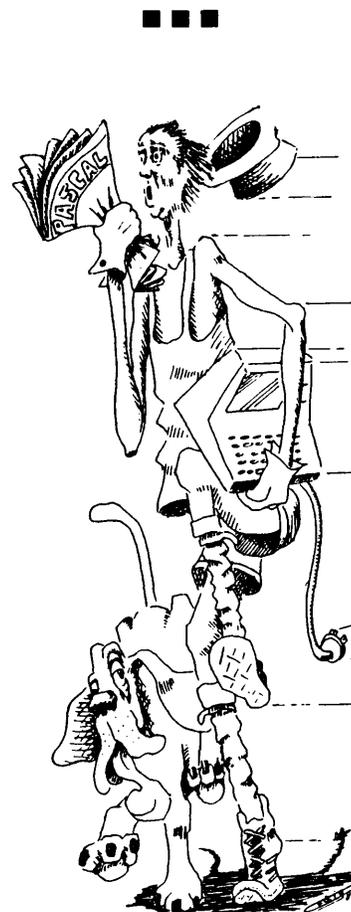


Figure 1 - Pascal Runoff Entries

Abrams	NOTEBOX	hierarchical DB	Knowles	CRYPTO	game
Adams	BOTTI	game	Lee	ADAPTINT	solve integrals
Alford	OKIUTL	printer utility - character set editor	Losch	COMPAXIT	removes spaces in .BAS file
Arden	RECIPBOX	DB	Marks	EASYASPI	calculate pi
Barcus	BIKEGEAR	analyze bicycle gearing	Marsh	MENUMAKE	menu generator
Beachy	QUEST	adventure	McCoy	RADIX	base conversion
Berry	LANDER	game	McKinnon	REMIND	appointment calendar
Bloom	SPELLOO	game	Meacham	PHONE	phone number DB
Bohn	CROSSPRINT	print sideways on printer	Meacham	SLITF	file splitter
Bowden	WYC	write your congressman	Mierau	INPUT	string input routines
Brooke	CRYPTOR	file encryption	Mierau	SCREEN	screen generator
Brown	CALENDAR	18 month calendar	Mitton	RESCUE	recover Turbo source from memory
Brown	FKEY	function key redefinition	Murphy	FX100	printer utility
Buro	PSYUTIL	utilities for use in psychological testing	Naleway	POSTER	prints posters
Calwas	QUASILAND	adventure game	Oshel	NEON	84 Kaypro graphic display
Candle	AMORT	calculate loan payments	Oualline	CATALOG	disk catalog system
Cebula	MANDEL	fractal graphics	Oualline	MODEM	communications
Cedarblom	LSTCTRL	printer utility	Oxborrow	TD	directory
Coleman	SW	print sideways on printer	Palmer	DISPLAY	improved CP/M TYPE
Conforti	QW	text editor	Penney	DUMP	dump/modify utility
Coniglio	VB	vocabulary builder	Pisarchuk	SPLIT	split Hex files
Connell	WHISTLES	Kaypro bells and whistles	Reed	TWOSIDES	print both sides of paper
Despain	ARCHIVE	combines files into one archive file	Regan	PRINT	file printer
Dingacci	TRUN	run time package for .CHN files	Rock	FANFOLD	print both sides of paper (14 yr old)
Dodrill	OT	show MSDOS directory tree	Rogers	MAP	word game
Duncan	CASSTIME	times cassette tapes	Rogers	CNS	multi-base calculator
Erdmann	BAFFLES	game	Rosenberg	DBLJOTTO	game
Everton	DRAW	graphics editor for 84 Kaypros	Roussi	CALEND	calendar printer
Fankbonner	CARDCAT	DB	Rowley	DIAMOND	84 Kaypro graphic display
Fergusson	ORBIT	display orbits	Rudd	CLOCK	access Kaypro clock
Fields	TBLCTENT	table of contents for WordStar	Runcie	PRINTER	printer utility
Fleming	ENCRYPT	file encryption	Ryall	PROBE	disk editor
Frick	CYFER	file encryption	Sailors	GENCHARFILE	create DUNGEONS characters
Frick	MDC	polar/cartesian coordinate transform	Salvador	MORTGAGE	calculate mortgage payments
Gilbert	MSWEEP	CP/M to 8" MSDOS file transfer	Schmid	PRINTIT	configure printer
Gimlin	ARTIST	printer graphics	Schmittroth	CLINK	transfer spreadsheet to spreadsheet
Gjerde	INV	theater inventory	Schrum	OT	overtime scheduler
Grabar	SORTT	sort utility	Shiflett	DECODE	decrypt MBASIC file
Gray	CHECK	will Turbo .COM run on this size system?	Shiflett	LONGMATH	long integer math
Griffith	DBAR	math teacher	Shyne	XREFFP	cross reference generator for text files
Griffith	DDRAW	graphics editor for 84 Kaypro	Skirvin	VSORT	sort utility
Groos	DICE	game	Smith	FASTFONT	printer utility
Hamblen	TGAME	typing game	Snyder	TRADE	game
Hays	LIST	file lister	Sosman	SRT	sorting routine
Heiss	TOOLBOX	library of procedures for 84 Kaypros	Sprague	PLANTER	design planter boxes
Hicken	GRADER	school grades spreadsheet	Torney	SUNPOS	determine solar intensity
Hyder	CURSOR	modify cursor	Toso	KEYCH	reconfigure keyboard
Johnston	ALT2	calculate maximum altitude of model rocket	Wakefield	INCLUDE	runtime package for .CHN file
Jones	LE-MENU	CP/M shell	Weinberger	SAVER	recover Turbo source from memory
Kerber	PRINTIT	printer utility	Wilcox	PAMPHLET	print pamphlets from WordStar file
Kircher	INLINE	generates inline code from .PRN file	Witte	DRAW1	83 Kaypro drawing program
			Woolley	MAILLIST	mailing labels

103 ENTRIES

GENERAL UTILITIES PACKAGE

Rolland Management Systems Utilities Package offers maximum access to files for CP/M* users

Sort System - Organize and reorganize any file

- ★ fixed or variable length records
- ★ unrestricted sort keys
- ★ skip or include records or blocks

\$39.95

Menu System - No need to "chain" back to menu

- ★ unrestricted construction of menus
- ★ restrict user modification of menus
- ★ enhanced batch processor under the menu

\$29.95

Batch List Utility - List many files with one command

- ★ optional page heading, page number or date
- ★ output to printer or another file
- ★ configure for any printer

\$19.95

Dump Utility - Decode and display entire file contents

- ★ show hexadecimal, integer or RAD50 value
- ★ dump to screen, printer or disk file
- ★ begin at any block and dump any number

\$19.95

FREE with purchase of any Utility - **Base Conversion Utility** - decimal to binary to hexadecimal to RAD50

All 4 Utilities: \$79.95

Mastercard or Visa call (919)276-6070

Rolland Management Systems, Inc.

Rt 5, Box 135

Laurinburg, N.C. 28352

CP/M is a registered trademark of Digital Research, Inc.

Rescuing Lost Text From Memory

By Steve Mitton

7662 Fallswood Way
Lorton VA 22039

Steve's RESCUE.PAS snared second prize in the Pascal Runoff. In addition to being useful, RESCUE is a good example of programming with procedures. Look closely at the embedded function and procedure within a procedure.

You're programming in Turbo and run a program before you've saved your source code. The program, OOPS.PAS, doesn't do exactly what you expected, but instead loops infinitely, forcing you to warm boot to regain control of your computer. Down to zero in a word. You've lost your program, and many minutes of precious programming time fly out the window.

But Don't Panic

You can recover your program (or any text, for that matter) in CP/M with DDT —

1. Locate the lost text in memory, noting the address where the text begins and ends.
2. Calculate the size of the text with DDT's 'h' command, convert this value to decimal, and divide by 256 (decimal) to obtain the number of pages to save (in decimal).

3. Move the text to address 100 Hex using the DDT 'm' command.

4. Save text using CP/M's SAVE command.

Or you could rescue your text with Turbo Pascal.

RESCUE.PAS, my Pascal Runoff entry, dumps memory and then writes the block you specify to a file, RESCUE.TXT.

Operation RESCUE

Q — To QUIT RESCUE.

B — To BEGIN the search through memory. The default address to begin the scroll is 16000, since RESCUE.COM is 16K.

A — To specify an ADDRESS to start/restart scrolling.

After you've marked the beginning and ending of the text you want rescued, RESCUE writes your block to a file. That's all there is to it.

It's unlikely RESCUE will overwrite

(continued on page 59)

Figure 1 - Predefined Array MEM

```
function NextChar : char;

begin
  NextChar := chr (mem [address]);
  address := succ(address);
  If address = 0 then
    writeln (^G, crlf, 'Warning *** End of 64K memory ***', ^G)
end; { NextChar }
```

Figure 2 - Program RESCUE

```
program rescue;

const start = 16000.0;
  CharSet : set of char = [' '..'^'];
  cr = ^M;
  lf = ^J;
  crlf = ^M^J;
  maxint = 32767;
  AddrConv = 65536.0;
  linesize = 72;
  conin = 1;
  resetDOS = 13;
  closefile = 16;
  delfile = 19;
  writesector = 21;
  makefile = 22;
  setdma = 26;

type FcbType = record
  dr : byte;
  name : array [1..11] of char;
  data : array [12..35] of byte
end;

var FileOut : FcbType;
  LocSt, LocEnd, Loc : real;
  drive, Achar, Command, response : char;
  Done, goodnum, StartSelected : boolean;

procedure readchar(var anychar: char);

{
  Allows a response to be entered without a carriage
  return and translates all inputs to uppercase.
}

var areg : byte;
begin
  repeat
    until keypressed;
    areg := Bdos(conin);
    anychar := UPCASE(chr(areg))
end; { readchar }

procedure readnum ( var areal : real; var goodnum : boolean);
{
```

(listing continued on page 59)

Drawing A Planter In 3-D

By Dennis Sprague

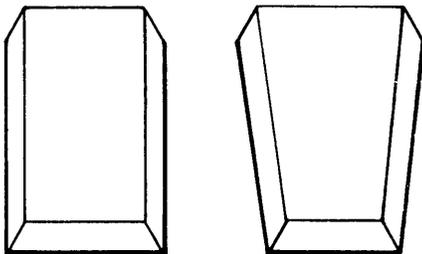
912 4th Ave. NW
Great Falls MN 59404

Even if you're not in the mood to build a planter, you'll find PLANTER's 3-D graphics eye-catching (on an 84 Kaypro, yet). Apply them to your own programs. Dennis's entry calculated and drew its way into our hearts (and third place in the Pascal Runoff).

You can build a wooden planter by cutting and fitting a number of similarly shaped boards together. The result resembles an oak barrel or old fashioned wooden bucket. Answer a few of PLANTER's questions and it creates a three dimensional image.

Carpenter's Details

To build a planter with top and bottom equal, you need to know only one angle — the bevel on two edges of each board.



To create a planter with larger top or bottom you must taper each board at the top and bottom and bevel the tapered edges as well.

The angles in the taper and bevel are determined by the number of sides in the planter and the relationship between height and the top and bottom diameters.

Each board will be an isosceles trapezoid with the nonparallel sides beveled toward the same surface of the board. The taper and bevel of the edges allow the sides to fit together to form the planter.

Program Notes

PLANTER is interactive. You describe your planter — number of sides, top and bottom diameters, and height — and PLANTER draws it, and reports the shape of the boards needed to build it.

Editor's note: This program works fine as is. We've added the TypeKaypro function to allow it to run on 83 non-graphics Kaypros. As this is al-

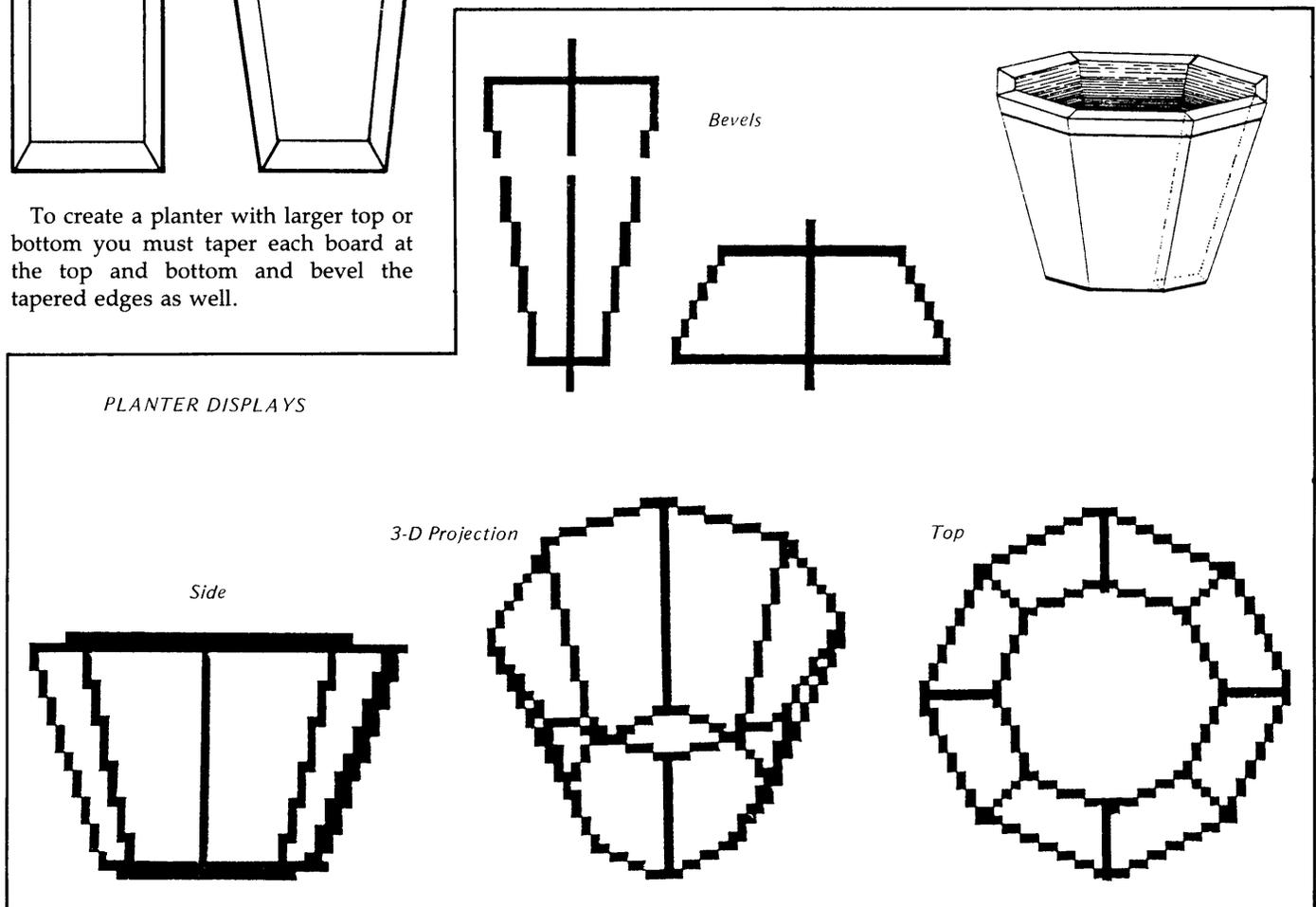
ways a concern with us (e.g., we must ship .COM files which will run on a variety of different size and type systems), we compile our Turbo programs on a 60K system. In the case of PLANTER, this meant that we had to split the source file.

3 Dimensions

To display a three dimensional image on a surface, combine two of the coordinates to produce a two dimensional display. The y and z coordinate values are combined using a rotational value so the image appears to tilt forward or backward. See procedure draw3Dline in the program listing (Figure 1). It's a procedure embedded within procedures containing several embedded procedures.



Planter Program Listing
(begins on page 54)



```

program planter; { 1985 djs }
{
  written by Dennis Sprague

  TypeKaypro function added by Larry Fogg at Micro Cornucopia - 12/85
}
{$U+}
type
  stringType = string[20];
{
  description of the planter.
}
  description = record
    numberOfSides : integer;
    topDiameter, bottomDiameter, height, largestDimension : real;
  end;
{
  point in a three dimensional coordinate system.
}
  point = record
    x, y, z : real;
  end;
{
  line in three dimensional system
}
  line = record
    pA, pB : point;
  end;
{
  two dimensional point and line
}
  twoDPoint = record
    x, y : integer;
  end;
  twoDLine = record
    p1, p2 : twoDPoint;
  end;
{
  figure of four points able to define the corners of a single
  side of the planter in a three dimensional coordinate system.
}
  figureOf4 = array [1..4] of point;
{
  a figureOf4 plus top and bottom center points of the planter.
}
  boardImage = record
    figure : figureOf4;
    topCenter,
    bottomCenter : point;
  end;

var
  planterVar : description;

{
  output to CON: or LST:
}
  outputDevice : text;
  usingGraphics : boolean;

{$I PLANTER1.INC}

FUNCTION TypeKaypro : Boolean; { tests for 83 vs 84 Kaypro }
VAR
  OriginalVal,
  NewVal,
  NewValStored : Integer;

BEGIN
  OriginalVal := Port [30]; { port 1E is unused in 83 - doesn't exist in 84 }
  NewVal := OriginalVal - 2;
  Port [30] := NewVal; { store new value in port 1E }
  NewValStored := Port [30]; { read value stored in port 1E }
  IF NewVal <> NewValStored { was value actually stored? }
  THEN TypeKaypro := true { 84 model }
  ELSE TypeKaypro := false; { 83 model }
  Port [30] := OriginalVal { restore original value }
END; { TypeKaypro }

procedure intro;
begin
  clrscr;
  writeln(' .... Planter program .... ');
  writeln;
  writeln('a wooden planter may be described by its number of sides,');
  writeln('the height and the top and bottom diameters. ');
  writeln;
  writeln('supply the overall dimensions and the number of sides of');
  writeln('the planter, this program will return the dimensions of');
  writeln('each board required to build the planter. ');
  writeln;
end;

procedure calculateOneSide(planter : description);

procedure setUpBoard(var board : boardImage; planter : description);

procedure setUpImage(var workBoard : boardImage; planter : description);

procedure setUpCenterLineOfPlanter(var board : boardImage;
  planter : description);
begin
{
  create the center points of the top and bottom of the planter
}
  with board do
  begin
    topCenter.x := 0;
    topCenter.y := 0;
    topCenter.z := planter.height / 2;
    bottomCenter.x := 0;
    bottomCenter.y := 0;
    bottomCenter.z := - planter.height / 2;
  end;
end;

```

```

begin { setUpImage }
{
  set up figure representing a side of the planter
  along with the top and bottom center points of the planter.
}
  setUpSide(workBoard.figure, planter, 1);
  setUpCenterLineOfPlanter(workBoard, planter);
end;

procedure moveCornerOneToOrigin(var board : boardImage);
var
  corner : integer;

procedure translate(var vpoint : point; offset : point);
begin
  with vpoint do
  begin
    alter the coordinates of 'vpoint' by the value of the
    coordinates of 'offset'. 'offset' is usually the new origin.
  }
    x := x - offset.x;
    y := y - offset.y;
    z := z - offset.z;
  end;
end;

begin { moveCornerOneToOrigin }
{
  adjust all points so that corner one of the figure is at the
  origin.
}
  for corner := 2 to 4 do
    translate(board.figure[corner], board.figure[1]);
    translate(board.topCenter, board.figure[1]);
    translate(board.bottomCenter, board.figure[1]);
    translate(board.figure[1], board.figure[1]);
  end;

procedure tiltBoard(var board : boardImage; planter : description);
var
  corner : integer;
  angleOfEdgeToPlanterCenterLine : real;

procedure rotateXZ(var vpoint : point; rotation : real);
var
  scratchX : real;
begin
{
  the image may be rotated in the XZ plane by using rotateX
  for X coordinate values and rotateY for Z values.
}
  with vpoint do
  begin
    scratchX := x;
    x := rotateX(x, z, rotation);
    z := rotateY(scratchX, z, rotation);
  end;
end;

begin { tiltBoard }
{
  the image has to have an egde along the Z axis to measure the
  egde bevel.

  if the top and bottom diameters are equal the edge is already
  on the Z axis.
}
  if planter.topDiameter <> planter.bottomDiameter then
  begin
{
  figure the angle an edge of the planter will make with
  the center line of the planter.
}
    with planter do
    if height = 0 then
      angleOfEdgeToPlanterCenterLine := 1.5708 { 90 degrees in radians }
    else
      angleOfEdgeToPlanterCenterLine :=
        arctan( ( (topDiameter / 2) - (bottomDiameter / 2) )
          / height);
{
  rotate each point in the figure and the top and
  bottom center points so that corner three of the side
  of the planter is on the Z axis.
}
    for corner := 1 to 4 do
      rotateXZ(board.figure[corner], - angleOfEdgeToPlanterCenterLine);
      rotateXZ(board.topCenter, - angleOfEdgeToPlanterCenterLine);
      rotateXZ(board.bottomCenter, - angleOfEdgeToPlanterCenterLine);
    end;
  end;

procedure drawIt(figure : figureOf4;
  sizeFactor : real;
  message : stringType);
var
  screenOrigin : twoDPoint; { location of origin on screen }
  imageSize : real; { largest size image displayed }
begin
{
  display the side during the creation, translation and tilting
  of the figure. used as visual check during calculations.
}
  imagesize := 80 / sizeFactor;
  screenOrigin.Y := 50;
  screenOrigin.X := 80;
  clrscr;
  writeln(message);
  write(' |', char(82), char(33), char(82), char(192));
  write(' |', char(33), char(112), char(132), char(112));
  drawSide(figure, 0, screenOrigin, imageSize);
  delay(2000);
end;

const
  testing = true;
begin { setUpBoard }

```

```

{
to see the board image used to calculate dimensions
set 'usingGraphics' and 'testing' to true.
}
setUpImage(board, planter);
if usingGraphics and testing then
drawIt(board.figure, planter.largestDimension, ' set up ');

moveCornerOneToOrigin(board);
if usingGraphics and testing then
drawIt(board.figure, planter.largestDimension, ' move to origin ');

tiltBoard(board, planter);
if usingGraphics and testing then
drawIt(board.figure, planter.largestDimension, ' tilt ');
end;

procedure boardDimensions(board : boardImage;
planter : description);

function distanceBetweenPoints(a, b : point) : real;
begin
{
units of distance between two points in a three dimensional
coordinate system.
}
distanceBetweenPoints :=
sqrt(sqrt(a.x - b.x) + sqrt(a.y - b.y) + sqrt(a.z - b.z));
end;

procedure midPoint(var varPoint : point; a,b : point);
begin
{
returns 'varPoint' as the point midway between two points
in a three dimensional coordinate system.
}
varPoint.x := (a.x + b.x) / 2;
varPoint.y := (a.y + b.y) / 2;
varPoint.z := (a.z + b.z) / 2;
end;

function topWidth(board : boardImage) : real;
begin
{
length of top edge of side of the planter
}
with board do
topWidth := distanceBetweenPoints(figure[1], figure[2]);
end;

function bottomWidth(board : boardImage) : real;
begin
{
length of bottom edge of side of planter
}
with board do
bottomWidth := distanceBetweenPoints(figure[3], figure[4]);
end;

function centerLineLength(board : boardImage) : real;
var
midTop, midBottom : point;
begin
{
length along centerline of board
}
with board do
begin
midPoint(midTop, figure[1], figure[2]);
midPoint(midBottom, figure[3], figure[4]);
end;
centerLineLength := distanceBetweenPoints(midTop, midBottom);
end;

function centerLineToEdgeAngle(board : boardImage) : real;
begin
{
taper of the edge of the side
}
centerLineToEdgeAngle := 57.2958 *
abs(arctan((topWidth(board) - bottomWidth(board)) / 2)
/ centerLineLength(board));
end;

function edgeAngle(board : boardImage) : real;
var
workMidPoint : point;
begin
{
the tapered edges of the sides of the planter must be beveled
to this angle for the sides to fit smoothly.

corners one and three are on the Z axis by now. use the
X and Y coordinates of the midpoint of the opposite side
( the line between corners two and four ) to calculate

the angle in relation to the X axis. this is the required
edge bevel.
}
with board do
midPoint(workMidPoint, figure[2], figure[4]);
edgeAngle := 57.2958 * abs(arctan(workMidPoint.y / workMidPoint.x))
end;

function angleOfSideToVertical(board : boardImage;
planter : description) : real;
var
midTop,
midBottom : point;
topDistance,
bottomDistance : real;
begin
{
this is the angle of a side of the planter measured from the
centerline of the side to the center line of the planter.
}
if planter.height = 0 then
{
figure is flat
}
angleOfSideToVertical := 90
else
if planter.topDiameter = planter.bottomDiameter then
{
figure is straight sided
}
angleOfSideToVertical := 0
else
with board do
begin
{
calculate angle
}
midPoint(midTop, figure[1], figure[2]);
midPoint(midBottom, figure[3], figure[4]);
topDistance := distanceBetweenPoints(midtop, topCenter);
bottomDistance := distanceBetweenPoints(midbottom, bottomCenter);
angleOfSideToVertical := 57.2958 *
abs(arctan( ( topDistance - bottomDistance )
/ planter.height ));
end;
end;

procedure drawFlatView(top, bottom, center : real);

procedure setPoint(var p : point; px,py,pz : real);
begin
with p do
begin
x := px;
y := py;
z := pz;
end;
end;

var
figure : figureOf4;
screenOrigin : twoDPoint; { location of origin on screen }
imageSize : real; { largest size image displayed }
begin { drawFlatView }
{
draw flat view of board from dimensions
}
screenOrigin.x := 110;
screenOrigin.y := 70;

if top >= bottom then
begin
if top >= center then
imageSize := 50 / top
else
imageSize := 50 / center;
end
else
begin
if bottom >= center then
imageSize := 50 / bottom
else
imageSize := 50 / center;
end;
end;

set up the board
}
setPoint(figure[1], - top / 2, center / 2, 0);
setPoint(figure[2], top / 2, center / 2, 0);
setPoint(figure[3], - bottom / 2, - center / 2, 0);
setPoint(figure[4], bottom / 2, - center / 2, 0);

gotoxy(48, 15);
write('flat view of board ');
write('L', 'L', char(73), char(142), char(132), char(142));
drawSide(figure, 0, screenOrigin, imageSize);

{
change it to end of board
}
imageSize := 50 / top;
screenOrigin.x := 50;
screenOrigin.y := 80;
figure[1].y := 0;
figure[2].y := 0;

set up edge angle
}
pointOnCircle(figure[3],
top / 2,
- edgeAngle(board) * 0.017453);

with figure[3] do
begin
{
move it over
}
x := x + figure[1].x;
y := - y;
end;
with figure[4] do
begin
{
mirror image of point three
}
x := - figure[3].x;
y := figure[3].y;
end;

gotoxy(13,22);
write(' view of end of board ');
gotoxy(13,23);
write(' (thickness not to scale)');

write('L', 'L', char(90), char(82), char(132), char(82));
drawSide(figure, 0, screenOrigin, imageSize);
delay(2000);
gotoxy(1,10);
end;

```

(listing continued on page 56)

```

begin { boardDimensions }
{
display the results of the calculations
}
clrscr;
writeln(outputDevice);
with planter do
  writeln(outputDevice,numberOfSides:3,
    ' sided figure with top of ',topDiameter:2:1,
    ', bottom of ',bottomDiameter:2:1,' height of ',height:2:1);
writeln(outputDevice,
  '----- each board dimensions ----- ');
writeln(outputDevice,
  '          angle of          angle of ');
writeln(outputDevice,
  '          center edge to          one side ');
writeln(outputDevice,
  '          top bottom line center edge of planter');
writeln(outputDevice,
  '          width width length line angle to vertical');
writeln(outputDevice,
  '-----');
writeln(outputDevice,
  topWidth(board):8:2,
  bottomWidth(board):10:2,
  centerLineLength(board):10:2,
  centerLineToEdgeAngle(board):8:2,
  edgeAngle(board):9:2,
  angleOfSideToVertical(board, planter):15:2);
if usingGraphics then
  drawFlatView(topWidth(board), bottomWidth(board), centerLineLength(board));
end;

var
workBoard : boardImage;
begin { calculateOneSide }
setUpBoard(workBoard, planter);
boardDimensions(workBoard, planter);
end;

begin { main }
intro;
usingGraphics := TypeKaypro; { true if 84 Kaypro with graphics }
assign(outputDevice, 'CON:');
{
text output can go to LST: or CON:
}
reset(outputDevice);
{
prime the pump
}
acceptDimensions(planterVar);
{
if the number of sides is not valid quit.
}
while planterVar.numberOfSides in [3..40] do
begin
if usingGraphics then
threeViews(planterVar);
calculateOneSide(planterVar);
acceptDimensions(planterVar);
end;
end.
{ included file PLANTER1.INC }

procedure acceptDimensions(var planter : description);
type
OkSet = set of 0..40;

function sourceInteger(message : stringType; accept : OkSet) : integer;
label
redo;
var
work : integer;
begin
redo:
work := 0;
write(' enter ',message,' .. ');
{$I-}
readln(work);
{$I+}
if (not (IOResult = 0)) or
(not (work in accept)) then
begin
writeln('*** error ***');
goto redo;
end;
sourceInteger := work;
end;

function sourceReal(message : stringType; lowLimit, highLimit : real) : real;
label
redo;
var
work : real;
begin
redo:
work := 0.0; { in case of just RETURN entered }
write(' enter ',message,' .. ');
{$I-}
readln(work);
{$I+}
if (not (IOResult = 0))
or (work < lowLimit)
or (work > highLimit) then
begin
writeln('*** error ***');
goto redo;
end;
sourceReal := work;
end;

procedure acceptNumberOfSides(var workPlanter : description);
begin
{

```

```

get number of sides in planter
}
workPlanter.numberOfSides := sourceInteger('number of sides',[0,3..40]);
end;

procedure acceptOtherDimensions(var workPlanter : description);
begin
{
get top and bottom diameters and height of planter
}
with workPlanter do
begin
topDiameter := sourceReal('top diameter',0.0,999.9);
bottomDiameter := sourceReal('bottom diameter',0.0,999.9);
height := sourceReal('height',0.0,999.9);
end;
end;

procedure setLargestDimension(var workPlanter : description);
begin
{
store largest dimension in the planter description.
}
with workPlanter do
begin
if topDiameter >= bottomDiameter then
largestDimension := topDiameter
else
largestDimension := bottomDiameter;
if height > largestDimension then
largestDimension := height;
end;
end;

label
redo;
begin { acceptDimensions }
writeln('enter 0 sides to exit ');
redo:
acceptNumberOfSides(planter);
with planter do
begin
if numberOfSides in [3..40] then
begin
{
number of sides is acceptable
}
acceptOtherDimensions(planter);
if ((topDiameter = 0) and (bottomDiameter = 0))
or ((height = 0) and (topDiameter = bottomDiameter)) then
begin
writeln(' *** error in planter dimensions *** ');
goto redo;
end;
setLargestDimension(planter);
end
else
begin
{
number of sides not acceptable, zero everything
}
topDiameter := 0;
bottomDiameter := 0;
height := 0;
largestDimension := 0;
end;
end;
end;

function rotateX(oldx, oldy, angletorotate : real) : real;
{
calculations for rotating x coordinate of oldx, oldy pair thru
'angletorotate' ( measured in radians ).
}
begin
rotateX := (oldx * cos(angletorotate)) + (oldy * sin(angletorotate));
end;
function rotateY(oldx, oldy, angletorotate : real) : real;
{
calculations for rotating y coordinate of oldx, oldy pair thru
'angletorotate' ( measured in radians ).
}
begin
rotateY := (oldy * cos(angletorotate)) - (oldx * sin(angletorotate));
end;

procedure pointOnCircle(var pointparm : point; diameter, arc : real);
{
return point 'pointparm' at rotation 'arc' ( measured in radians )
from positive X axis on circle with diameter of 'diameter'.
the circle is centered on the origin.
}
begin
pointparm.x := (diameter / 2) * cos(arc);
pointparm.y := (diameter / 2) * sin(arc);
end;

procedure setUpSide(var figure : figureOf4;
planter:description;
side:integer);

procedure setUpTopPoints(var workFigure : figureOf4;
workPlanter : description;
side : integer);

var
rotation : real;
begin
{
points 1 and 2 in figure will be on the top rim of the planter.
the points will be separated by the angle determined by the
number of sides in the planter and which side we are creating.
}
with workPlanter do
begin
{
corner one

```

```

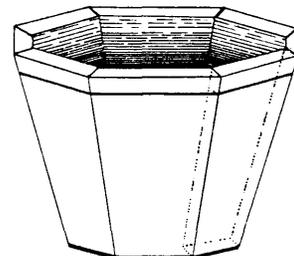
}
rotation := 6.28319 / numberOfSides * ( side - 1 );
pointOnCircle(workFigure[1], topDiameter, rotation);
workFigure[1].z := height / 2;
{
corner two
}
rotation := 6.28319 / numberOfSides * side;
pointOnCircle(workFigure[2], topDiameter, rotation);
workFigure[2].z := height / 2;
end;
end;
procedure setUpBottomPoints(var workFigure : figureOfN;
workPlanter : description;
side:integer);
var
rotation : real;
begin
{
points 3 and 4 in figure will be on the bottom rim of the planter.
the points will be separated by the angle determined by the
number of sides in the planter and which side we are creating.
}
with workPlanter do
begin
{
corner three
}
rotation := 6.28319 / numberOfSides * ( side - 1 );
pointOnCircle(workFigure[3], bottomDiameter, rotation);
workFigure[3].z := - height / 2;
{
corner four
}
rotation := 6.28319 / numberOfSides * side;
pointOnCircle(workFigure[4], bottomDiameter, rotation);
workFigure[4].z := - height / 2;
end;
end;
begin { setUpSide }
{
set up figure that will define a side of a planter.
two points on both the top and bottom rims of the planter
are created for each side of the planter.
}
setUpTopPoints(figure, planter, side);
setUpBottomPoints(figure, planter, side);
end;
procedure drawSide(figure : figureOfN;
rotation : real;
origin : twoDPoint;
scale : real);
procedure draw3DLine(work : line; origin : twoDPoint; scale, rotation : real);
procedure combineYZ(var work : line; rotation : real);
{
to display a three dimensional image on a two dimensional
surface you can combine two of the coordinates to produce a
two dimensional display.
}
the Y and Z coordinate values are combined using a rotational
value so the image appears to tilt forward or backward on the
display.
}
with work do
begin
pA.y := rotateY(pA.z, pA.y, rotation);
pB.y := rotateY(pB.z, pB.y, rotation);
end;
end;
procedure lineToScreen(work : line; origin : twoDPoint; scale : real);
var
scratch : twoDLine;
procedure screenScale(var twoD : twoDLine; work : line; scaleValue : real);
begin
{
scale the working values for the display size and convert
into integer values.
}
with twoD do
begin
p1.x := round(work.pA.x * scaleValue);
p1.y := round(work.pA.y * scaleValue);
p2.x := round(work.pB.x * scaleValue);
p2.y := round(work.pB.y * scaleValue);
end;
end;
procedure drawKayproLine(work : twoDLine; origin : twoDPoint);
begin
{
the Kaypro '84 CON: device allows an escape sequence to create
lines using the block graphics available.
}
escape followed by ascii 'L' uses the next four characters
sent to the CON: as the y and x coordinates of the endpoints
of the desired line ( 'yxxy' order).
each coordinate value needs binary 32 added to it.
}
the Y axis on Kaypro goes down screen so the scaled y values
are subtracted from the screen origin to allow positive values
to go up on screen.
}
binary 32 must be added to the X and Y coordinates sent to screen.
}
with work do
write(^[L',char(32 + origin.y - p1.y), [ endpoint 1 ]
char(32 + origin.x + p1.x), [ ' ' 1 ]
char(32 + origin.y - p2.y), [ endpoint 2 ]
char(32 + origin.x + p2.x)); [ ' ' 2 ]
end;
end;

```

```

begin { lineToScreen }
screenScale(scratch, work, scale);
drawKayproLine(scratch, origin);
end;
begin { draw3DLine }
combineYZ(work, rotation);
lineToScreen(work, origin, scale);
end;
var
workLine : line;
begin { drawSide }
{
draw edges of a single side of a planter
}
top
}
workLine.pA := figure[1];
workLine.pB := figure[2];
draw3DLine(workLine, origin, scale, rotation);
{
side
}
workLine.pA := figure[2];
workLine.pB := figure[4];
draw3DLine(workLine, origin, scale, rotation);
{
bottom
}
workLine.pA := figure[4];
workLine.pB := figure[3];
draw3DLine(workLine, origin, scale, rotation);
{
side
}
workLine.pA := figure[1];
workLine.pB := figure[3];
draw3DLine(workLine, origin, scale, rotation);
end;
procedure threeViews(planter : description);
var
currentSide,
step : integer; { view loop counter }
workFigure : figureOfN;
screenOrigin : twoDPoint; { location of origin on screen }
imageSize : real; { largest size image displayed }
const
radiansIn45Degrees = 0.785385;
{
Kaypro screen size.
}
horizontalPixels = 160;
verticalPixels = 100;
begin { threeViews }
clrscr;
write(' three views of planter ');
{
draw three views of the planter image. the first is from
the top looking down into the planter, the second at 45 degrees
tilt and then a side view.
}
an image size of one fourth of the horizontal screen dimension
will allow three images of the same size with a border around
each image.
}
the Kaypro has 160 horizontal pixels so the image size is
160 / 4 or 40 pixels.
}
imageSize := horizontalPixels / 4;
{
vertical center of images will be slightly below vertical center
of screen.
}
the Kaypro has 100 vertical pixels and counts from the top of
the screen for vertical coordinates. the screen origin vertical
setting will be 60% of 100 or 60 pixels.
}
screenOrigin.y := round(verticalPixels * 0.60); { for Kaypro }
for currentSide := 1 to planter.numberOfSides do
begin
setUpSide(workFigure, planter, currentSide);
for step := 0 to 2 do { number of 45 degree steps to take }
begin
{
the horizontal screen origin is dependent on which of the
three images is being drawn.
}
each image is one fourth of the entire screen but we're only
drawing three images. the first image is centered two thirds
of the image size from the left side of the screen. the next
two images follow at intervals of four thirds of the image size.
}
screenOrigin.x := round(( imageSize * 0.66 )
+ ( step * imageSize * 1.33 ));
drawSide(workFigure,
- radiansIn45Degrees * step,
screenOrigin,
imageSize / planter.largestDimension);
end;
end;
delay(5000);
end;

```



Path Searching
 Archiver
 Aliases
 8Mb Print Spooler
 Disk Variables
 Screen Paging
 Multiple Commands
 Pipes
 Memory Management

CP/M™

+

ConIX™

=

I/O Redirection
 Function Keys
 Argument Handling
 Virtual Disk
 Overlay Searching
 Memory Files
 New SysCalls
 Perfected User Areas

If you've tried some of the earlier CP/M-80 add-ons such as Microshell™ and ZCPR™, then you know it's possible to add new features to CP/M. Now you can have much more without all the trade-offs when you use ConIX - **the latest and greatest upgrade for CP/M!**

"(ConIX) goes a great deal farther in flexibility than any other CP/M shells. . . this system has all of its competitors beat." *Computer Language, 6/85.*

ConIX Operating System: List \$99.95 ~~Special~~

A complete upgrade for 48K+ CP/M 2.2/3.0 and equivalent systems. Provides professional capabilities with blinding speed, as often found on more expensive MS-DOS™ and UNIX™ machines. Installs easily in 1 minute to add over 100 new commands and features. Eliminates all points of user frustration with CP/M. Uses only 1/2K TPA, 0-27K disk minimum.

ConIX is the greatest, most powerful 8-bit upgrade, with speed and capabilities that are so incredible it's bringing users *back* to CP/M!

ConIX Programming System: List \$99.95 ~~SALE~~

A full programming language for ConIX extends CP/M SUBMIT capability. Features conditionals, loops, subroutines, labels, nesting, interrupt processing, error traps, and debugging facilities. Design intricate menu systems. Includes a special "compiler" that provides string and numeric variables, an integer math package, a relocatable assembly code generator, and much more. A *must* for CP/M power-users and developers!

ConIX Pull-Down Menu: List \$34.95 ~~Special~~

A user-friendly interface to ConIX that is fully menu-driven, with helpful prompts and easily understood explanations of all menu functions. Loads with a single keystroke at the prompt level. Source code included!

ConIX Library I - XCC Utilities: List \$49.95 ~~SALE~~

Over 25 utilities written in the ConIX shell language, including hierarchical directories with overlay - adds pathname capability to existing software, interactive debugger, move/copy/link multiple files, print files with pagination, review disk files for deletion, unerase disk with stats, full-screen TYPE, and more. Source code included! Does not require Programming System.

NEW YEAR SPECIALS - SAVE 30% TO 100% OFF!

- ConIX O.S. or Programming System: **\$49.95 each**
- ConIX O.S. and Programming System: **\$79.95 for both**
- SUPER SAVINGS WHEN YOU BUY 'EM ALL: ConIX O.S., Programming System, Menu System, Library I - **Only \$99.95!!**
- ConIX Menu System: **FREE with ConIX!**
- ConIX Library I: **\$34.95 with any ConIX product**

Now that we've knocked the bottom out of our prices, you should have no reason to suffer along with plain-old CP/M or incompatible half-baked add-ons any more! Get started with ConIX for \$49.95, or steal us blind and get everything for only \$99.95. Either way, you get the Menu System absolutely FREE! Get on the bandwagon with a company that's committed to *you* with the ongoing development of products for CP/M and ConIX.

Prices include manual, 8" disk, and end-user support. 5¼" disk conversions: 48TPI \$6, other \$10. Shipping: \$4.50 UPS, \$10 Canada, \$15 overseas. NY residents add sales tax. Discount incentives for Dealers, OEMs, User Groups - CALL!



Computer Helper Industries Inc. • P.O. Box 680 • Parkchester Station, NY 10462
 800-628-2828 X513 24hr. order hotline - COD only • 212-652-1786 M-F:10-6 sales/technical staff

"We're helping your computer work better for you!"

Trademarks - CP/M: Digital Research, ConIX: Computer Helper Ind., Microshell: New Generation Systems, ZCPR: Richard Conn/Echelon, MS-DOS: Microsoft, UNIX: AT&T Bell Labs

RESCUING LOST TEXT FROM MEMORY

(continued from page 52)

your text when it's loaded. But if you think your text might be in the first 16K of memory, use the CP/M SAVE command —

```
A> save 63 rescue.16k
```

You can call up the file with DDT, move it to higher memory, and then rescue with RESCUE —

```
A> ddt rescue.16k
```

```
DDT VER 2.2      (see this)
NEXT PC        (watch for '_' prompt)
4000 100
m100,4000,4100 (type this)
^C
```

```
A> RESCUE
```

The Guts Of RESCUE.PAS

RESCUE accesses memory via one of Turbo Pascal's predefined arrays: MEM. See Figure 1.

Procedure Scroll (Figure 2) scrolls through memory beginning at StartLoc. It prints only the characters in CharSet (the printable ASCIIs) and the <cr> and <lf> characters. All other values are represented by periods. For readability a <cr> followed by <lf> will constitute an EOL. Otherwise, the length of the string "line" will determine the EOL. Before exiting, StartLoc is reset so the scroll can resume at the last address scanned.

Procedure Scroll contains an embedded function (NextChar) and an embedded procedure (PrintLine).

Since Scroll alone needs to use NextChar and PrintLine, they're inaccessible to the remainder of program RESCUE.

Limitations, Notes, & Wrap Up

RESCUE was written for a 64K CP/M system. Many of the input/output calls are made directly to the CP/M operating system. But with minor modifications RESCUE.PAS can be adapted for MS-DOS systems.

RESCUE will recover only text actually in memory. For a 64K system using WordStar, RESCUE can recover a file up to about 25K long.

May RESCUE allow you to recover from at least some of your mistakes.

■ ■ ■

RESCUE PROGRAM LISTING (continued from page 52)

```
By reading numbers in as a string, the VAL function built into TURBO
provides an easy way to ensure a number has been input.
)

var code : integer;
    astring : string[5];
begin
    readln(astring);
    VAL (astring, areal, code);
    if code = 0 then
        begin
            goodnum := true;
            areal := int(areal)
        end
    else goodnum := false
end; { readnum }

procedure initialize;
begin
    LocSt := 0;
    LocEnd := 0;
    drive := chr($40);
    StartSelected := false;
    done := false
end; { initialize }

procedure init_msg;
begin clrscr;
    writeln('          RESCUE.COM ver 2.0');
    writeln('    Copyright (C) 1985 by Steve Mitton (tel 703 339-5842).');
    writeln('    Released to the PUBLIC DOMAIN for non-commercial use. ');
    writeln;
    writeln('    The purpose of RESCUE.COM is:');
    writeln('    (1) to SEARCH the TPA for a lost program text');
    writeln('    and once found,');
    writeln('    (2) to WRITE it to disk as RESCUE.TXT. ');
    writeln('    In principal, a lost text can be recovered after a WARM BOOT');
    writeln('    (control-c), and sometimes after a COLD BOOT (restart button).');
    writeln('    When a program crashes, load RESCUE.COM immediately. Be sure');
    writeln('    sure to insert a disk that has enough empty space to accept the');
    writeln('    recovered file. The drive you select to write the file RESCUE.TXT');
    writeln('    on is automatically reset by RESCUE.COM to a R/W status. ');
    writeln('    Learn how to use RESCUE.COM before you really need it. ');
    writeln('    In an emergency, you want to be right -- the first time! ');
    writeln('    Note: In a 64k computer, its memory is deployed like this:');
    writeln('          0 -- 100     reserved for CP/M');
    writeln('          100 -- 16,000 occupied by RESCUE.COM itself');
    writeln('          16,000 -- 60,000 Transient Program Area');
    writeln('          60,000 -- 64,000 CP/M Operating System');
    writeln;
    write('          ===> M -- GO TO MENU <=== ');
    repeat
        until keypressed;
    clrscr
end; { init_msg }

function MemoryMarked : boolean;
begin
    if StartSelected and (LocSt < LocEnd) then MemoryMarked := true
    else MemoryMarked := false
end; { MemoryMarked }

procedure WriteFile;

label EXIT;
var NumSectors, I, Areg, Try: byte;
    DMA : integer;
    WroteSector : boolean;

begin
    {
    Verify addresses and drive and initialize FCB
    }

    write(crlf, 'START ADDRESS = ', LocSt:1:0,
        crlf, 'END ADDRESS = ', LocEnd:1:0,
        crlf, 'Write file RESCUE.TXT to');
    case drive of
        '@' : writeln(' default drive? <Y/N>');      { '@' is ASCII 40H }
        'A'..'P' : writeln(' drive ', drive, '? <Y/N>');
    end; {case}
    readchar(response);
    if response <> 'Y' then goto EXIT;
    FileOut.dr := ord(drive) - $40;
    FileOut.name := 'RESCUE.TXT';
    fillchar (FileOut.data[12], 23, 0);
    BDOS(resetDOS);

    {
    Open file RESCUE.TXT
    If a file RESCUE.TXT exists and it will be deleted first
    }

    Areg := BDOS(delfile, addr(FileOut));
    Areg := BDOS(makefile, addr(FileOut));
    if Areg in [0..3] then writeln(crlf, 'Writing file . . .')
    else
        begin
            writeln('Cannot open file');
            goto EXIT
        end;

    {
    Have to be careful since max integer 32767.
    The initial 128 subtracted will be incremented below.
    }

    if LocSt < maxint then DMA := round(LocSt) - 128
    else begin
        DMA := round(LocSt - AddrConv) - 128;
        end;
    NumSectors := round((LocEnd - LocSt)/128) + 1;
    writeln('Start address = ', LocSt:1:0);
    writeln('End address = ', LocEnd:1:0);
    writeln('Number of sectors = ', NumSectors:1, ' ');
    writeln(' . . . ');
    for I := 1 to NumSectors do
        begin
            DMA := DMA + 128;      { increment DMA }
            BDOS (setdma, DMA);
            Try := 0; WroteSector := false;
            repeat
                Areg := BDOS (writesector, addr(FileOut));
                if Areg <> 0 then Try := succ(Try)
                else WroteSector := true
            until (Try = 5) or (WroteSector);
            if Try = 5 then writeln('g, *** Warning ** error writing sector ', I)
            end; { do }
        end;
end;
```

(listing continued on page 60)

RESCUE PROGRAM LISTING
(continued from page 59)

```

    { Close file }
    Areg := BDOS (closefile, addr(FileOut));
    if Areg in [0..3] then
        begin
            writeln ('RESCUE.TXT written');
            done := true
        end
    else writeln (^G, '** error ** file write error ??');
EXIT:
end; { Write_File }

procedure Scroll (var StartLoc : real);
{
This procedure will scroll through memory beginning at StartLoc. It will
print only the characters in CharSet ( the printable ASCII's ) as well as
cr and lf characters. All other ASCII's are represented by a '.'.
For readability a cr followed lf will constitute an EOL, otherwise the
length of the string 'line' will determine the EOL. Before exiting,
StartLoc is reset so that the scroll can resume at the last address
scanned if desired.
}

label CONTINUE, EXIT;
var address : integer;
    line : string[linesize];
    Areg : byte;
    achar : char;

function NextChar : char;
{
This function returns the character at the current 'address' and
increments 'address' for the next read.
It also does a check to ensure 'address, is within 64K.
}
begin
    NextChar := chr (mem [address]);
    address := succ(address);
    if address = 0 then
        writeln (^G, crlf, 'Warning *** End of 64K memory ***', ^G)
    end; { NextChar }

procedure PrintLine;
{
This procedure will print a string, the global 'line' along with the
line number for the following line.
}

var LineNum : real;
begin
    if address < 0 then LineNum := address + AddrConv
    else LineNum := address;
    write (' ', line, crlf, LineNum:5:0);
    line := '';
end; { PrintLine }

begin { Scroll }
    if StartLoc < maxint then address := round(StartLoc)
    else address := round(StartLoc - AddrConv);
    line := '';
    write (StartLoc:5:0);

    repeat
        achar := NextChar;
    CONTINUE:
        if length(line) > linesize then PrintLine;
        if achar in CharSet then line := line + achar
        else
            if achar = cr then
                begin
                    achar := NextChar;
                    if achar = lf then
                        begin
                            PrintLine;
                            achar := NextChar;
                            goto CONTINUE
                        end
                    else
                        begin
                            line := line + '.';
                            { concats cr to line as a '.' }
                            { no need to advance read to next char }
                            goto CONTINUE
                        end { else if lf }
                    end { if cr }
                else line := line + '.';
            until keypressed;

    Areg := BDOS (conin);
    command := Ucase(chr(Areg));
    if Command = ^S then
        begin
            repeat
                until keypressed;
            Areg := BDOS(conin);
            Command := Ucase(chr(Areg));
            if Command = ^S then goto CONTINUE
            end;
        EXIT:
            if address < 0 then StartLoc := address + AddrConv
            else StartLoc := address
        end; { Scroll }

procedure DisplayHelp;

```

```

begin
    writeln('          ==> MENU <==');
    writeln;
    writeln('          Q -- to QUIT rescue.com at any time');
    writeln;
    writeln('          ==> TO SEARCH THROUGH MEMORY <== ');
    writeln;
    writeln(' B -- to BEGIN the search [at 16,000, the default address] ');
    writeln(' A -- [at any time] to specify a start/restart ADDRESS ');
    writeln(' ^S -- temporary pause/resume search scroll');
    writeln(' M -- pause display to go to MENU');
    writeln(' C -- to CONTINUE the scroll from the last address shown');
    writeln;
    writeln;
    writeln('          ==> TO RECLAIM AN AREA OF MEMORY <==');
    writeln;
    writeln(' S -- to specify a STARTING address for rescue.txt');
    writeln(' E -- to specify an ENDING address [important: be sure this address]');
    writeln('       is at least two lines below the end of your program');
    writeln('       so that a proper End-of-File will be included. ');
    writeln(' D -- to specify the DRIVE that rescue.txt is to be sent to. ');
    writeln(' W -- to WRITE rescue.txt to the disk in the drive specified');
    write('Command selected ->          GOOD LUCK ');
    gotoxy(21, 28);
    readchar(command);
    writeln
end; { DisplayHelp }

{
main program
}

begin
    initialize;
    init_msg;
    clrscr;
    writeln;
    DisplayHelp;

    repeat
        case command of
            'A' : begin
                write(crlf, 'Address to resume scroll',
                    crlf, '<cr> will resume scroll at last address -> ');
                readnum (Loc, goodnum);
                if not goodnum then
                    begin
                        writeln(^g, '** Input error ** bad number ??');
                        DisplayHelp
                    end
                else command := ^C;
                writeln;
            end;
            'B' : begin
                Loc := start;
                scroll(Loc)
            end;
            'C' : scroll(Loc);
            'D' : begin
                writeln(crlf, 'Select the drive to write RESCUE.TXT.',
                    crlf, '<cr> selects default drive, or select ^A thru ^P');
                readchar(drive);
                if not (drive in ['A'..'P']) then
                    begin
                        drive := chr($40);
                        writeln(crlf, 'Default drive selected')
                    end
                else writeln(crlf, 'Drive ', drive, ' selected');
                DisplayHelp
            end;
            'E' : if StartSelected then
                begin
                    write (crlf, 'Select RESCUE.TXT end address -> ');
                    readnum(LocEnd, goodnum);
                    if not goodnum then
                        begin
                            writeln(crlf, ^g, '** Input error ** bad number ??');
                            DisplayHelp
                        end
                    else
                        if MemoryMarked then
                            begin
                                WriteFile;
                                if not done then DisplayHelp
                            end
                        { note DONE set to true by WriteFile if write good }
                        else
                            begin
                                write(crlf, ^g, '** error ** end address too small',
                                    ' START ADDRESS = ', LocSt:1:0,
                                    ' END ADDRESS = ', LocEnd:1:0);
                                DisplayHelp
                            end
                        end
                    end
                else
                    begin
                        writeln (^G, crlf, 'Start address has not been selected');
                        DisplayHelp;
                    end;
            end;
            'M' : begin
                writeln;
                DisplayHelp
            end;
        end;
    end;
end;

```

```

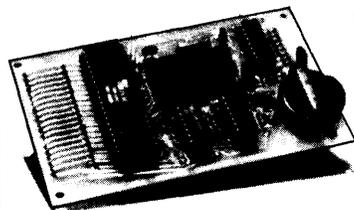
'Q' : begin
  writeln(^G, crlf, 'Abort rescue operation? <Y/N>');
  readchar(response);
  if response <> 'Y' then
    begin
      writeln(^G, crlf, '*** Continue RESCUE ** ');
      DisplayHelp
    end
  else
    begin
      writeln(^G, crlf, 'Abort at user request');
      done := true
    end
  end;

'S' : begin
  write (crlf, 'Select RESCUE.TXT starting address -> ');
  readnum(LocSt, goodnum);
  if goodnum then
    begin
      StartSelected := true;
      command := 'A'
    end { if goodnum }
  else
    begin
      writeln(crlf, ^g, '*** Input error ** bad number ??');
      DisplayHelp
    end;
  end;

'W' : if MemoryMarked then
  begin
    WriteFile;
    if not done then DisplayHelp
  end
  { note DONE set to true by WriteFile if write good }
  else
  begin
    writeln(crlf, ^g, '*** error ** memory not marked ??');
    DisplayHelp
  end;
  else
  begin
    writeln( ^G, crlf, '*** error ** bad command ??');
    DisplayHelp
  end
  end; {case}
until done
END.

```

End of Listing



Ztime-I

CALENDAR/CLOCK

\$69 KIT

NOW WITH FILE
DATE STAMPING!

- Works with any Z-80 based computer.
- Currently being used in Ampro, Kaypro 2, 4 & 10, Morrow, Northstar, Osborne, Xerox, Zorba and many other computers.
- Piggybacks in Z80 socket.
- Uses National MM58167 clock chip, as featured in May '82 Byte.
- Battery backup keeps time with CPU power off!
- Optional software is available for file date stamping, screen time displays, etc.
- Specify computer type when ordering.
- Packages available:

Fully assembled and tested	\$99.
Complete kit	\$69.
Bare board and software	\$29.
UPS ground shipping	\$ 3.

MASTERCARD, VISA, PERSONAL CHECKS,
MONEY ORDERS & C.O.D.'S ACCEPTED.

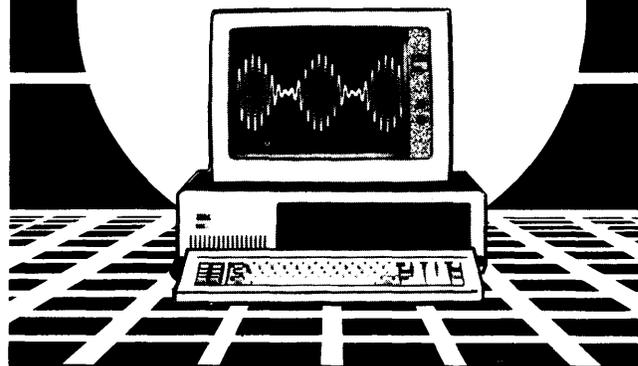
N.Y. STATE RESIDENTS ADD 8% SALES TAX



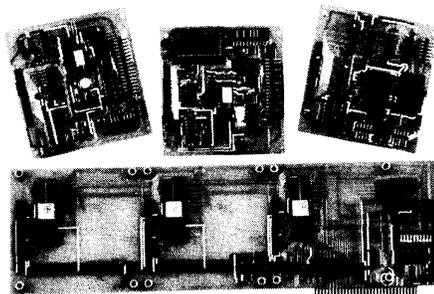
KENMORE
COMPUTER
TECHNOLOGIES

P.O. Box 635, Kenmore, New York 14217 (716) 877-0617

WE BRING
ENGINEERS, SCIENTISTS
AND THE IBM-PC
TOGETHER

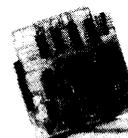


Modular Data Acquisition System



- | | |
|--------------------------|--------------------------|
| • 12-bit resolution | • 8-bit resolution |
| • 24-KHz throughput rate | • 30 KHz throughput rate |
| • 16 channel SE, 8 DI | • 16 channel SE, 8 DI |
| • LABSTAR Software | • LABSTAR Software |
| \$690.00 | \$490.00 |

IEEE-488 GPIB Multifunction Board

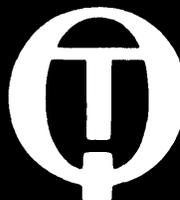


- Full GPIB controller capability
- Three 16-bit interval timers
- 24 digital I/O lines
- Software included **\$495.00**

Waveform Synthesizer Board



- Generates user-definable signal
- Up to 2000 points per envelop
- 200ns maximum output rate per point **\$795.00**



QUA TECH, INC.

478 E. Exchange St.
Akron, OH 44304
(216) 434-3154

ERAC CO.

8280 Clairemont Mesa Blvd., Suite 117
San Diego, California 92111
(619) 569-1864

KAYPRO EQUIPMENT

9" Amber Monitor	\$39.00
9" Green Monitor	35.00
12" Green Monitor	45.00
81 Series ROM's	7.00
Keyboard (new Kaypro)	75.00
Keyboard Siemens w/case	65.00
Wiring Harness	7.50

KAYPRO MAINBOARDS

K2 With Sockets	\$39.00
K2 Populated - Tested	129.00
K2 with PRO8-3 Mod.	239.00

This one runs at 2.4 & 5 MHz. up to 4 drives SS, DS, or Quad drives. New ROM has screen dump and step speed select. With manual & software, tested and guaranteed 30 days.

K2 PRO8-2 Mod. EXCHANGE	149.00
-------------------------	--------

We will update your K3 or K4 as above. Mail working Main board.

We can install double or quad drives and check complete package CALL
K10 Populated - Tested 250.00
WD1002-5 Hard Disk Controller 200.00
Host Interface Board 5.00

CABINETS

K10 or K2X	\$45.00
------------	---------

Bottom portion with front and back only.
CRT Bezel 12.00

SUBSTITUTE CABINET

This cabinet has a 12" green monitor on top that swivels left to right with a 3-position vertical movement. The base is 16"x18"x5" and contains a switching power supply. It has room for the main board, two drives, a modem and whatever else you want to stuff in it. It comes with a Siemens keyboard with case and is programmed like the Kaypro keyboard with many extra Wordstar and CP/M commands. The whole cabinet is color-coordinated and sharp looking. Nobody will ever suspect this is a Kaypro. Now you can build the classiest Kaypro computer ever for less than \$550.00. The complete cabinet is a steal at \$179.00

MISC. CABLES

9" 50 Pin - 50 Pin Header	\$2.50
14" 40 Pin - 40 Pin Header	2.25
9" Hard Disk - 20 Pin Header	2.00
14" Dual Inline 16 Pin Male (2)	2.00
I/O Connector for Keyboard	2.25
Coiled Keyboard Cable	1.50

Pioneer LD700 Laser Video Disk Player \$299.00
Wireless Remote Control 55.00

ERAC CO.

8280 CLAIREMONT MESA BLVD., SUITE 117
SAN DIEGO, CA 92111 (619) 569-1864

STORE HOURS

Mon. - Fri. 9:00-6:00 — Sat. 10:00-4:00
TERMS: VISA, MasterCard, Certified Checks, Money Order, NO COD. Visa and MasterCard add 3%. *Personal checks must clear BEFORE we ship. Include shipping charges. California residents add 6% Sales Tax.*

Call for our Test Equipment Mailer!

★ SPECIALS ★

★★★★★★★★★★★★★★★★★★★★
Bahco Wire Cutter "Swedish" 6.95
★★★★★★★★★★★★★★★★★★★★

CPU/SUPPORT

MC68000-8 CPU	\$12.50
C80186-3	27.00
C80186-6	9.00
Z80 CPU	1.00
Z80A CPU	2.50
Z80A PIO	2.50
Z80A SIO	5.00
Z80A DMA	10.00
8088	10.00
8089-3	29.00
D8742 CPU/EEPROM 12 MHz	20.00
D8284A	3.00
D8288	9.00
4164-15	.90
41256-15	2.50
41256-12	2.75

MODEM

TMS 99532 (Modem)	\$12.00
-------------------	---------

FLOPPY CONTROLLER

WD 1791	\$9.00
WD 1793	5.00
WD 1797	7.00
WD 2797	9.50

BAUD GENERATOR

WD 8116	\$4.50
---------	--------

MISCELLANEOUS

UC3524 Sw. Reg.	\$.85
WD1010	20.00
WD1014	28.00
WD1015	28.00
MM58167A Clock Chip	5.50
CA301 Op Amp	.35
CA083 Op Amp	.45
2N3055	.70
2N6852	1.00
Voltage Regulators	
LM323K + 5V - 3A	4.50
LM78H12 + 12V - 5A	5.00
6116 2Kx8 Static Ram	1.50

VIDEO CONTROLLER

6545	\$7.00
6545A-1	8.00
6845	6.00

EPROMS

2716	\$3.50
2732	3.75
2764	2.25
2764-25	3.50

74LS SERIES

HC00	\$.35
HCU04	.40
LS 125	.40
LS 373	.60
LS 375	.60
4073	.35
7406	.40

SOLAR CELLS

2½" x 5/16", .5V .5A	\$2.25
----------------------	--------

LOW PROFILE IC SOCKETS

8 Pin 50 per Tube	\$2.50
14 Pin 25 per Tube	1.50
16 Pin 25 per Tube	2.25
18 Pin 25 per Tube	2.50
20 Pin 20 per Tube	2.00
24 Pin 20 per Tube	2.50
28 Pin 10 per Tube	2.00
40 Pin 10 per Tube	1.60

PRINTERS

Centronics 703	\$395.00
Centronics 588	225.00
Centronics P1	105.00
Teletype 43KSR	300.00
NEC 2000 with Single Bin Cut Sheet Feeder	1500.00
Transtar 315 Color	375.00

\$100

CCS 2200 12 slot MF	\$375.00
CCS 2810A CPU	210.00
2065 64K RAM	250.00
2422 FDC 5¼" & 8"	325.00
2718 I/O 2 Ser., 2 Par.	225.00

POWER SUPPLIES

— WALL PLUG-IN TYPE —

13.5VDC - .5A Regulated	\$4.25
9VDC - .3A	2.15
6VAC - .95A	1.35

— LINEAR —

Condor Model CP731 DC Power Supply + 14V-4A, + 5V-7A, ± 15V-.5A	\$16.00
Power Supply with Nicad Pack 12V .5AH	6.50
14KV DC/DC TV Power Supply + 5V/1A, - 5V/.2A, + 12V/1A, - 12V/.2A, - 24V/.05A	15.00

SWITCHES

5V-9.5A, 12V-3.8A, -12V-.8A	\$39.00
5V - 10A	25.00

FLOPPY DISK DRIVES

— ½ HEIGHT —

Mitsubishi M4851 DSDD	\$99.00
Mitsubishi M4853 DS Quad	139.00
Qume Trak 142 5¼" DSDD	99.00
Tandon TM 848-1 8" SSDD	149.00
Tandon TM848-2 8" DSDD	229.00

— STANDARD HEIGHT —

Shugart SA460 DS Quad - As is	\$36.00
Shugart SA850 8" DSDD	199.00
Remex RFD4000	199.00
Perci 277 2x8"	399.00
All Instruction Manuals	10.00

TERMINALS

InterTube II Smart (B&W)	\$225.00
InteColor 3602 Smart (Color)	329.00

MISCELLANEOUS

Headset/Boom Microphone	\$3.95
Lead Acid D Cell 2V - 2.5A	1.50
Nicad Pack 12V .5AH	6.50
5 Blade Muffin Fans	7.50
Joystick 4 Switches 1" Knob	5.50
Elgar 400W Unint. Power Sup.	425.00
16-Pin Component Carrier, Gold	25.00

Controlling A Kaypro & A PC Cursor

By Paul Hyder

28900 SW Parkway Ave #9
Wilsonville OR 97070

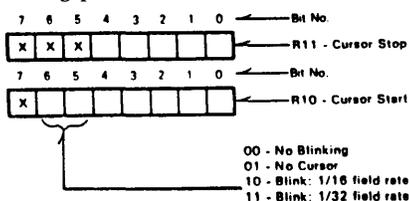
One important IC the Kaypro and PC have in common is the 6845 CRT controller. Paul's Pascal Runoff entry wasn't a finalist, but it does peek inside the 6845 and solve the very practical problem of controlling your cursor.

Turbo Pascal and the 6845 CRT controller make it easy to control the Kaypro's (and in a limited way) the PC's cursor. Here's how you do it.

Inside The 6845 CRT Controller

The 6845 has 19 internal registers. Zero through 3 establish the horizontal format and timing parameters. Four through 9 determine vertical format and timing characteristics. Ten through 17 handle the cursor attributes, screen memory addressing, and the light pen interface. We'll use 10 and 11 to control the cursor.

Register 10 is the cursor start register. Register 11 is the cursor stop register. Bits 5 and 6 of the cursor start register determine whether the cursor will blink or not. Bits 0 through 3 in register 10 determine the scan line starting point. Bits 0 through 3 in register 11 determine the scan line ending point.



To get a full block cursor on a Kaypro 84, for example, we turn on all 16 scan lines by assigning 0 to register 10 and 15 to register 11.

Ports

Most Z80 and 8086/8088 peripheral devices are I/O Mapped. That means that the processor makes the I/O line active and then puts the port number on the address bus. Turbo Pascal provides predefined port arrays to access the ports.

On the Kaypro 84 the CRT controller is at port 28. Data to be put on the screen go to port 29.

Note: The Kaypro 83 doesn't use a video controller; the cursor is set by the way the gates are cobbled together. So this discussion of the 6845 isn't applicable to 83s.

To control the cursor we simply send a value to select register 10 —

```
Port[Select] := Start;
```

and a cursor value which sets bits 5 and 6 —

```
Port[Data] := StartValue;
```

Refer again to the register drawing above.

From Z80 To 8088

Since the PC also uses the 6845 CRT controller we can control its cursor as well, up to a point. The PC hardware automatically makes the cursor blink,

therefore a no-blink assignment (00) to bits 5 and 6 has no effect. (You can make a non-blinking cursor by turning the cursor off and writing a graphic cursor, but that's another story.) We can, however, alter the cursor shape by changing the scan line values in registers 10 and 11, as we did on the Kaypro.

First, we need to obtain the port addresses for the PC. You can sleuth it out of the BIOS, if you're diligent. It makes a difference whether you're using a color or monochrome card. 0304H is the address of the color card. 0B4H is the address of the monochrome card.

The PC uses 8 scan lines to define characters (the Kaypro 84 uses 16) with no space between scan lines. So in our program we need to redefine several constants to reflect PC data — Select, Data, Top, and Bottom.

```
Select = 980; (* Select port *)  
Data = 981; (* Data port *)  
Start = 10; (* Register 10 *)  
Stop = 11; (* Register 11 *)  
Top = 0; (* Top of cursor *)  
Bottom = 7; (* Bottom of cursor *)
```

You won't be able to stop the PC's blinkin' cursor this way, but you can have a block, semi-block, or underline, thanks to Turbo. See Figure 1 for the entire SETCURSOR program.



Figure 1 - SETCURSOR Program for the Kaypro 84

```
program setcursor (Input,Output);
```

```
const
```

```
    Select = 28; (* Select port for 6845 *)  
    Data = 29; (* Data port for 6845 *)  
    Start = 10; (* Register 10 of 6845 *)  
    Stop = 11; (* Register 11 of 6845 *)  
    Top = 0; (* Top of block cursor *)  
    Bottom = 15; (* Bottom of block cursor - underline value also *)
```

```
    Bblink = 64; (* Bit to set for blinking cursor *)  
    Brate = 32; (* Bit to set for slow blink: *)
```

```
var
```

```
    blink : boolean;
```

```
    startval,  
    stopval : byte;
```

```
    response : char;
```

```
begin
```

```
    startval := Top; (* default is Block noblink*)  
    stopval := Bottom;
```

```
    write('Cursor blink:(y,n) ');
```

```
    repeat  
        read(kbd, response);  
        response := Ucase (response);
```

```
    until response in ['Y','N'];  
    writeln (response);
```

```
    blink := (response = 'Y');
```

```
    if blink then
```

```
        begin
```

```
            startval := startval + Bblink;
```

```
            write('Cursor blink speed fast or slow:(f,s) ');
```

```
            repeat
```

```
                read(Kbd, response);
```

```
                response := Ucase (response);
```

```
                until response in ['F','S'];
```

```
                writeln(response);
```

```
                if response = 'S' then
```

```
                    startval := startval + Brate;
```

```
                end; (* set blink speed *)
```

```
    write('Block or Underline Cursor:(b,u) ');
```

```
    repeat
```

```
        read(Kbd, response);
```

```
        response := Ucase (response);
```

```
    until response in ['B','U'];
```

```
    writeln(response);
```

```
    if response = 'U' then
```

```
        startval := startval + Bottom;
```

```
    Port[Select] := Start;
```

```
    Port[Data] := startval;
```

```
    Port[Select] := Stop;
```

```
    Port[Data] := stopval;
```

```
end.
```

Turbo Pascal Inline Code From A .PRN File

By Jim Kircher

17915 NE 19th Pl.
Bellevue WA 98008

Jim's Pascal Runoff entry wasn't a prize winner, but its usefulness for low-level work made it one of our favorites. In particular, check out Jim's use of Turbo Pascal's IORESULT function. It might keep one of your programs from crashing. (Note, there's more than one way to skin assembly language. See the Pascal Column for another way of assembling Turbo inline code.)

Turbo Pascal, unlike Standard Pascal, allows you to insert low-level machine code into your high-level Pascal source via its INLINE feature. By writing some machine instructions yourself (instead of letting Turbo translate all the Pascal source to machine language) you can make your programs run almost as fast as assembled code.

I favor using Pascal source for everything but the tight loops and very low level byte/bit manipulations. Here, very small relocatable subprograms can be assembled and included as Inline procedures.

Unfortunately, entering machine code from the keyboard (hand-assembling, as it were) is tedious and prone to errors. INLINE.COM, my Pascal Runoff entry, will create this code from a .PRN file.

Theory Of Operation

To use, enter —

```
A>INLINE filename [file type .PRN assumed]
```

The .PRN file is opened and text is processed and written to a file created with filename.INL.

The original program was designed to work on .PRN files generated by Z80MR. But it turns out that some assemblers, ASM for example, insert blank lines at the beginning of the .PRN file. To generalize the program to handle other assemblers I substituted the line —

```
If (Length (Linebuffer) > 0 Then
```

for the line:

```
If linebuffer[2] <> ' ' Then
```

in PROCEDURE ProcessFile. (See Figure 1).

To have readable .INL files, all tabs in the .PRN file must first be expanded to spaces with PIP.

```
A> PIP fname.prn = fname.prn[T8]
```

The generated file starts with the INLINE statement and left parenthesis

and ends with the ');' as required by Turbo. All text that isn't executable code is surrounded by braces. Dollar signs and slashes are included where Turbo requires (delimiting each byte of code).

The program stops processing the .PRN file when it encounters the END pseudo-op code. This eliminates the Symbol table, etc., from the inline file.

Besides being relocatable, several conventions must be followed in the Z80MR source code. An object module must be produced. Macros must be expanded. Conditional code must not be listed for false conditionals. The switches for the preceding are all set properly (on) by default by Z80MR; no pseudo-ops are needed.

If you use DEFM, DEFB, etc., the LIST A (normally off by default) pseudo-op must be placed in the Z80MR source code file before assembly so all bytes will be listed in the .PRN file.

IORESULT

When you're working with disk files you need to know about a potential troublemaker — I/O error — the runtime code Turbo Pascal adds to every program it compiles to guard against runtime errors. For example, if you try to open a file that doesn't exist, Turbo generates an I/O error, and your pro-

Figure 1 - Inline Routine

```
TYPE
  Workstring = STRING[126];
  Filename = STRING[12];
VAR
  PrnFileName : Filename;
  InlFileName : Filename;
  SourceFile : TEXT;
  DestFile : TEXT;
  I,J : INTEGER;
  OK : BOOLEAN;
  linebuffer : Workstring;

FUNCTION Open(VAR fp:TEXT; name: filename): BOOLEAN;
BEGIN
  ASSIGN(fp,Name);
  { $I- } { so program does not terminate on bad filename }
  RESET (fp);
  { $I+ } { I/O checking active again }
  IF IOresult <> 0 THEN
    BEGIN
      Open := FALSE;
      CLOSE(fp);
    END
  ELSE Open := TRUE;
  END (open);

FUNCTION CopyTo(ch : char; Str1 : filename) : filename ;
VAR
  S : filename;
BEGIN
  S := '';
  Str1 := Str1 + ch ; { so while loop will exit }
  { the next line copies Str1 to S char by char until '.' }
  WHILE Str1[LENGTH(S)+1] <> ch DO S:=S+Str1[LENGTH(S)+1];
  CopyTo := S;
END;

PROCEDURE OpenMain;
VAR
  Str1 : filename;
BEGIN
  REPEAT
    IF NOT Open(SourceFile,PrnFileName) THEN
      BEGIN
        OK := FALSE;
        IF PrnFileName <> '' THEN WRITE(PrnFileName,' not found,');
        WRITELN(' Enter file name or <RETURN> to end. ');
        READLN(Str1);
        IF Str1 = '' THEN HALT;
        Str1 := CopyTo('.',Str1); { strip file type, if entered }
        InlFileName := Str1 + '.INL';
        PrnFileName := Str1 + '.PRN';
      END
    ELSE OK := TRUE;
    UNTIL OK;
  END (open main);

FUNCTION Trim( S : workstring): workstring; {trim spaces fore and aft}
BEGIN
  WHILE (S[1] = ' ') and (LENGTH(S) > 0) DO delete(S,1,1);
  WHILE (S[LENGTH(S)] = ' ') and (LENGTH(S) > 0) DO delete (S,LENGTH(S),1);
  Trim := S;
END (Trim);

PROCEDURE OpenDest;
BEGIN
  ASSIGN(destFile,InlFileName);
  REWRITE(destFile);
END (open dest);

PROCEDURE ProcessFile;
LABEL 10;
VAR
  destBuff : workstring;
CONST
```

gram dies.

Usually, however, you don't want your program to die. You want it to (at least) do something, such as create the file or look elsewhere for the old one. The IORESULT function in Turbo leads to the solution.

After each I/O statement is executed, the IORESULT function receives an integer which it checks to see if the I/O statement was successful. A 0 indicates success; anything else results in an error code (and a dead program).

The error will crash the program even if you use IORESULT. So, to keep your program running, you must disable the error trap with the compiler command \$I.

In my program, error trapping is disabled in the function "Open" so it won't terminate when it gets a bad filename. Procedure OpenMain then writes a "not found" message to the user and solicits a new file name. See Figure 1.



QUALITY SOFTWARE YOU CAN AFFORD!

NEW
Z80 Symbolic Debugger

- Only \$49.95 plus shipping.
- Screen oriented with a simultaneous display of instruction mnemonics, register, stack, and memory values.
- Breakpoints may be set on any combination of fixed memory address, register values and/or memory values.
- Uses Digital Research compatible SYM files.
- Supports Hitachi HD64180.

Relocatable Z80 Macro Assembler

- Only \$49.95 plus shipping.
- 8080 to Z80 Source Code Converter.
- Generates Microsoft compatible REL files or INTEL compatible hex files.
- Compatible with Digital Research macro assemblers MAC & RMAC.
- Generates Digital Research compatible SYM files.
- Conditional assembly.
- Phase/dephase.
- Cross-reference generation.
- Full Zilog mnemonics.
- INCLUDE and MACLIB FILES.
- Separate data, program, common, and absolute program spaces.
- Supports Hitachi HD64180.
- Z80 Linker and Library Manager for Microsoft compatible REL files available as an add-on to Assembler.

ATTENTION
Turbo Pascal
Assemblers will generate
in-line machine code
include files.

TO ORDER, CALL TOLL FREE:
1-800-367-5134, ext. 804
For information or technical assistance:
(808) 623-6361

Specify desired 5 1/4" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.

MITEK

Z80 is a trademark of Zilog, Inc. MAC and RMAC are trademarks of Digital Research, Inc. Turbo Pascal is a trademark of Borland International, Inc.

PRICE LIST

Z80 Macro Assembler: \$49.95
Assembler, Linker, and Library Manager: \$95.00
Manual Only: \$15.00
Z80 Symbolic Debugger: \$49.95
Manual Only: \$15.00
Assembler, Linker, Library Manager, and Debugger: \$134.95

Include \$5 for shipping and handling.

**P. O. Box 2151
Honolulu, HI 96805**

ATTENTION
Turbo Pascal
Assemblers will generate
in-line machine code
include files.

```

code1 = 7;          {Z80MR .PRN files, 1st byte of Z80 code in column 7}
mmonon = 25;       { " " " " , column where mnemonic starts}
BEGIN
WHILE NOT EOF(SourceFile) DO
BEGIN
DestBuff := ''; I := code1;
READLN(SourceFile, Linebuffer);          { get 1 line of text }
IF Length (Linebuffer) > 0
THEN
BEGIN
IF Linebuffer[2] <> ' ' THEN              { if address bytes, }
BEGIN                                    { then code follows. }
WHILE Linebuffer[I] IN ['0'..'9', 'A'..'F'] DO { get all hex bytes }
BEGIN
IF I = code1 THEN destBuff := '$';        { $ delimit 1st byte }
IF (Odd(I)) and (I > code1) THEN destBuff := destBuff + '/$';
destbuff := destbuff + Linebuffer[I];
I := SUCC(I);
END;
IF I > code1 THEN DestBuff := DestBuff + '/'; {delimit last byte of code}
END;
IF LENGTH(linebuffer) > I-1 THEN          { still some text left ? }
BEGIN
IF trim(COPY(linebuffer, mmonon, 5)) = 'END' THEN GOTO 10;
For J := I-1 TO 14 DO
DestBuff := DestBuff + ' ';              {Add spaces to line up columns}
destBuff := DestBuff + CHR(123);        { beginning comment brace }
destBuff := DestBuff + COPY(linebuffer, I, 100); {copy rest as comments}
destBuff := DestBuff + CHR(125);        { ending brace }
END;
WRITELN(DestFile, DestBuff);            { processed line to file}
END; { IF Length... }
END;
10:END; {ProcessFile}

PROCEDURE CommandLine(VAR S : filename);
{ get 1st argument from command line. $5D is hex address of 1st character }
{ in command line. The end of the 1st argument was marked with 0 by CP/M. }
BEGIN

```

```

S := '';
{ next line copies memory from hex 5D into string S until 0 encountered }
WHILE MEM[LENGTH(S)+$5D] <> 0 DO S:=S+CHR(MEM[LENGTH(S)+$5D]);
S := CopyTo('.', S);                      { dont include any file type}
S := CopyTo(' ', S);                      { dont include spaces }
END; {CommandLine}

{ ----- Main Program Body ----- }

BEGIN
CommandLine(PrrFileName);                 {get file name from command line }
IF PrrFileName <> ' ' THEN
BEGIN
InFileName := PrrFileName + '.INL'; {dest file always has type of .INL }
PrrFileName := PrrFileName + '.PRN'; {main file always has type of .PRN }
END;
OpenMain;
OpenDest;
WRITELN;
WRITELN(' -----writing to file ', InFileName, '--');
WRITELN(destFile, 'Inline '); {write reserved word 'Inline' to file}
WRITE(destFile, '(');          { " the left parentheses " " }
ProcessFile;                  { " the code & comments " " }
WRITELN(destFile, '#00');      { last byte must not be delimited }
RESET(destFile);              { now read the file }
WRITELN;                       { and write to the console }
WHILE NOT EOF(destFile) DO
BEGIN
READLN(destFile, Linebuffer);
WRITELN(Linebuffer);
END;
CLOSE(SourceFile);
CLOSE(DestFile);
END.

```

Plotting Orbits Using Kepler's Equation

By George S. Ferguson

Box 1212, RR 1
N Whitefield ME 04353

ORBIT.PAS was an honorable mention in the Pascal Runoff and a Micro C favorite. It should be a perfect addition to a skywatcher's library.

George's Pascal programming style is very unorthodox. He's put all of his functions and procedures in an Include file (like a library), and program flow jumps through the large main program from label to label with GOTOs.

My Pascal Runoff entry will bring Halley's comet into your living room. While it's plotting the orbits of planets, moons, asteroids, and even comets, it'll graphically teach you the principles of orbital motion. Figure 4 contains the program.

A Little Background

Two values determine the shape of an ellipse —

1. Eccentricity (e) — degree of flattening,
2. Semi-Major Axis (a) — 1/2 the smallest diameter, measured in A.U.

Two additional values are required for the program to function —

3. Step or M — the constant angular rotation interval. In effect, time,

4. Scale — properly chosen to fit the screen.

To get you started I've set up several orbits, including Halley's comet's, to run from a menu. When you tire of these, you can easily create your own, keeping in mind —

1. Values of 'e' must be in the range [0... 0.999],
2. Values of 'a' can assume any positive number, usually in [1..40],
3. Values of 'step' are integers and must be in the range [1..360],
4. Values of 'scale' are integers; experiment!

How It Works

ORBIT is always in one of two phases.

When it's computing, it calculates and stores screen coordinates in an array. Pressing any key during this process aborts the operation and returns you to the Top Menu. See Figure 1.

When it's displaying, it reads the screen coordinates from the array and plots the new positions.

Procedure GetCoords is in effect a polar/rectangular conversion routine, another of Turbo's missing scientific

functions. It takes V and R (angle and distance) and computes Latitude and Departure (differential screen coordinates) and sums them into Focusy and Focusx to generate true screen coordinates. These are then stored in the ARRAY screencoord, in the RECORD coord. See Figure 2.

Procedure Plotit (Figure 3) plots and erases coordinates.

Note that while the "step" is constant, the true positions of the orbiting body on the ellipse aren't evenly spaced due to the changing velocity. This is an effect of gravity and is noticeable with orbits of high eccentricity.

As the object approaches Perihelion (when it's closest to the sun), it speeds up. At Aphelion (when it's farthest from the sun) it's slowest.

A straight line connecting the orbiting body to the sun "sweeps" out equal areas in the orbital plane in equal intervals of time. This phenomenon is known as "Kepler's Second Law."



Figure 1 - Two Phases of Orbit

```
while (M<=360) do
begin
  M:=toradians(M);    { degrees to radians }
  compute_it;        { compute the position }
  getcoords; plotit;  { compute the screen coords and plot them }
  M:=radiansto(M); M:=M+step; i:=i+1; {increment M and array index}
  if keypressed then { abort }
  begin
    switch_statusline(off); goto top; {restart program}
  end;
end;
```

Figure 2 - Procedure GetCoords

```
v:=radiansto(v); temp:=v;
if (v>=0) then v:=90+v
else
v:=(360-(360+v))+90;
v:=toradians(v);

Latitude:=round(scale*(cos(v)*r));
if (temp>=0) then Latitude:=abs(Latitude);
Departure:= round(scale*(sin(v)*r));

{store the coords!}
screencoord[i].x:=focusx+Departure;
screencoord[i].y:=focusy+Latitude;
end;
```

Figure 3 - Procedure Plotit

```
{kaypro ESC sequence to write a pixel}
write(#27, '*', char(screencoord[i].y+32),char(screencoord[i].x+32));

if not orbiton then {we want to erase the previous pixel}
{kaypro ESC sequence to un-write a pixel}
write(#27, ' ', char(screencoord[i-1].y+32),char(screencoord[i-1].x+32));
```

Figure 4 - Program Orbit

```
program orbit;

label top,run,start,cont; {program segment labels for (gasp!) goto's}

const focusy: integer=50; {the middle of the screen..}
      speed: integer=75; {initial orbit speed; 75 ms. delay.}
```

```
type
  coords= record
    x:integer;
    y:integer;
  end;
  coordset= 1..360; {more than we'll need}
  switch= (on,off);

var
  r, a, M, e, v : real; {the major orbital elements and their derivations}
  Latitude, Departure, focusx, code, max, step, i, scale: integer;
  firsttime, info, orbiton: boolean; {control variables}
  ch: char;
  screencoord: array[coordset] of coords; {an array of orbital positions}
  input: string[5]; {used to bullet-proof data entry}

{$I orbit.inc}
{$I orbithlp.inc} — Text messages (not included in magazine listing)

begin {program}

{this section provides menu options of select orbits and help.}

top: {jump here to start over...}

  clrscr; i:=1; firsttime:=true; switch_cursor(on);
  orbiton:=false; info:=true; switch_statusline(on);
  writeln(' (c) ORBIT ver 7.0');
  writeln(' for the Kaypro 4-84');
  writeln;
  writeln('Pick an orbit or <SPACE> to roll your own!');
  writeln;
  writeln('A. Halley''s Comet');
  writeln('B. Earth');
  writeln('C. Comet Encke');
  writeln('D. an orbit of HIGH eccentricity');
  writeln('E. an orbit of NO eccentricity (a circle)');
  writeln('F. My favorite');
  writeln('H. HELP!');
  writeln('Q. quit program'); writeln;
  write('Type the letter of your choice: ');
  repeat read(kbd,ch) until upcase(ch) in ['F',' ', 'A','B','C','D','E','Q','H'];
```

ALL SALES SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. FREE COPY UPON REQUEST.

Z80A FOR CP/M 2.2

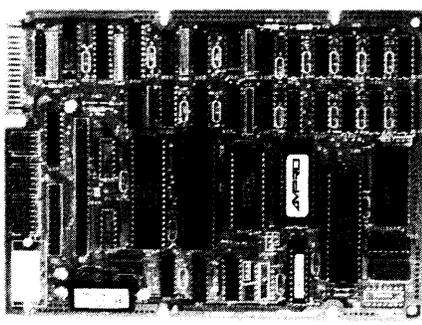
THE LITTLE BOARD™

INTEL 80186 FOR PC-DOS

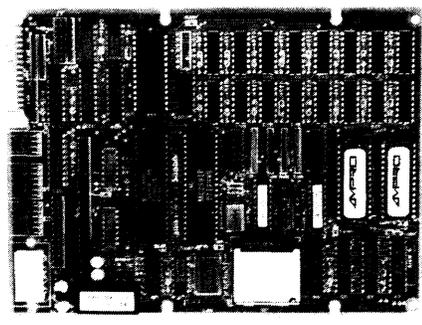
LITTLE BOARD™/Plus
8 BIT

NEW! TWO BOARDS TO CHOOSE FROM!
THIRD GENERATION SINGLE BOARD COMPUTERS!

LITTLE BOARD™/186
16 BIT



- COMMON FEATURES OF BOTH BOARDS:**
- * MOUNTS DIRECTLY TO 5 1/4 IN. DISK DRIVE
 - * DD FLOPPY CONTROLLER ON BOARD
 - * TWO RS232 SERIAL PORTS
 - * CENTRONICS PRINTER PORT
 - * SCSI HARD DISK INTERFACE.
 - * ASSEMBLED, TESTED, READY TO USE



FEATURES:

- * 4 MHZ Z80A CPU
- * 64K RAM
- * COMES WITH CP/M 2.2 DISKETTE + ZCPR3
- * READS OR WRITES TO DOZENS OF POPULAR 5 1/4 IN. FLOPPY FORMATS
- * 2732 BOOT EPROM
- * USE +5VDC @ .95A AND +12VDC 50MA
- * WORKS WITH 40 OR 80 TRACK DRIVES

\$299⁰⁰
(A & T WITH CP M 2.2)

BY AMPRO!

NOTE: BOTH OF THESE NEW LITTLE BOARDS™ WORK FINE WITH OUR POPULAR ZRT-80 TERMINAL BOARD KIT FOR \$89.95.

DISK DRIVE SPECIAL!
1/2 HT. DS-DD 40 TRACKS
PERFECT FOR LITTLE BOARD™
\$79.95 ea. 2 FOR \$139.00

FEATURES:

- * 8 MHZ INTEL 80186 FOR TRUE 16 BIT
- * 128K OR 512K RAM
- * WILL BOOT STANDARD PC DOS
- * 3 TIMES COMPUTING POWER OF A P.C.
- * RUNS THOUSANDS OF "MS-DOS" GENERIC PROGRAMS
- * COMES WITH MS-DOS COMPATIBLE ROM-BIOS AND UTILITIES
- * PC-SUPPORT SOFTWARE AND SOURCE AVAILABLE FOR \$89.00
- * MULTI-LAYER PC BOARD FOR LOW NOISE

\$419⁰⁰ (A & T, 128K) **\$499⁰⁰** (A & T, 512K)

Digital Research Computers

(OF TEXAS)

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

TERMS: Shipments will be made approximately two weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's with a \$75 deposit. Balance UPS COD. Add \$4.00 shipping. USA AND CANADA ONLY

Z80 TM OF ZILOG. CP/M TM OF D.R.I. LITTLE BOARD TM OF AMPRO. PC-DOS TM OF IBM. MS-DOS TM OF MICROSOFT.

```

case upcase(ch) of
'A': begin { Halley }
      e:=0.9673; a:=17.94; step:=10; scale:=4; statusline;
      goto run;
      end;
'B': begin { earth }
      e:=0.0167; a:=1; step:=10; scale:=35; statusline;
      goto run;
      end;
'C': begin { comet Encke }
      e:=0.847; a:=2.209; step:=10; scale:=25; statusline;
      goto run;
      end;
'D': begin { high eccentricity }
      e:=0.998; a:=50.00; step:=5; scale:=4; statusline;
      goto run;
      end;
'E': begin { a circle }
      e:=0.0; a:=1; step:=10; scale:=40; statusline;
      goto run;
      end;
'F': begin { my favorite }
      e:=0.875; a:=18; step:=10; scale:=4; statusline;
      goto run;
      end;
'H': begin { do help- info and commands }
      clrscr; help; goto top;
      end;
' ': clrscr; { to start }
'Q': begin { quit }
      switch_statusline(off); switch_cursor(on); halt;
      end;
end; {case}
{this section gets user input values for "custom" orbits}
start: {jump here to get user input}

i:=1; info:=false; switch_cursor(on);
repeat
gotoxy(1,1); clreol;
write('Enter eccentricity: '); read(input);
if not firsttime then {we already have an arrayfull}
begin
if (input=('R')) or (input=('r')) then {no change; use stored coords}
begin
gotoxy(1,1); clreol; switch_cursor(off); goto cont;
end;
if no change

```

```

end; {if not firsttime}
val(input,e,code); if (code>0) or (e>=1) or (e<0) then write('^G');
until (code=0) and (e<1) and (e>=0);
gotoxy(1,25); write('e= ',e:4:4); {we have good e}

repeat
gotoxy(1,1); clreol;
write('Enter semi-major axis: '); readln(input);
if not firsttime then
begin
if (input=('R')) or (input=('r')) then
begin
gotoxy(1,1); clreol; switch_cursor(off); goto run;
end;
end;
end; {if not firsttime}

val(input,a,code); if (code>0) or (a<0) then write('^G');
until (code=0) and (a>0);
gotoxy(15,25); write('a= ',a:4:4); {we have good a}

repeat
gotoxy(1,1); clreol;
write('Enter step: '); readln(input);
if not firsttime then
begin
if (input=('R')) or (input=('r')) then
begin
gotoxy(1,1); clreol; switch_cursor(off); goto run;
end;
end;
end; {if not firsttime}

val(input,step,code); if (code>0) or (step<1) then write('^G');
until (code=0) and (step>=1);
gotoxy(30,25); write('M= ',step:4, ' deg'); {we have good step}

repeat
gotoxy(1,1); clreol;
write('Enter scale: '); readln(input);
if not firsttime then
begin
if (input=('R')) or (input=('r')) then
begin
gotoxy(1,1); clreol; switch_cursor(off); goto run;
end;
end;
end; {if not firsttime}

```

(listing continued on page 69)

Do it yourself...

DESKTOP PUBLISHING™

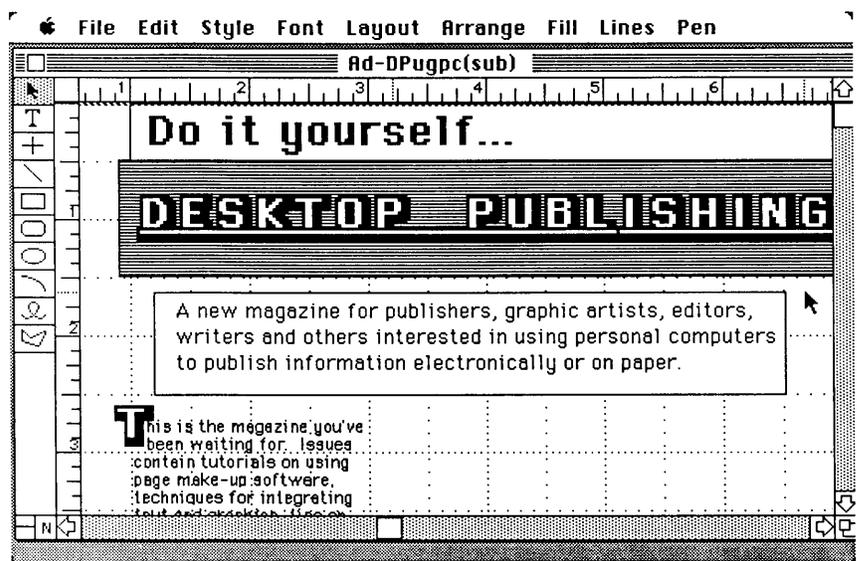
A new magazine for publishers, graphic artists, editors, writers and others interested in using personal computers to publish information electronically or on paper.

This is the magazine you've been waiting for. Issues contain tutorials on using page make-up software, techniques for integrating text and graphics, tips on producing more appealing graphics and typography, and helpful information about data communications and electronic publishing.

The bi-monthly magazine covers the latest products for the newly-emerging desktop publishing market, including laser printers, scanning equipment, page make-up software, archival storage devices, and topics of interest to people who do their own electronic or paper publishing.

The cover price is \$5, but you can save \$11 by ordering a subscription for \$24 and paying with your order. You'll get seven issues for the price of six! And there's no risk -- you can cancel after the first issue for a full refund!

We also publish *User's Guide to CP/M*, devoted entirely to CP/M computer users, and *PC USER*, a new magazine for users of PC-compatible computers and MS-DOS software. Each magazine is \$22 for a six-issue subscription. Send payment now and get seven issues for the price of six!



- DESKTOP PUBLISHING \$24
- USER'S GUIDE TO CP/M \$22
- PCUSER \$22

Name _____

Address _____

City _____

State, Zip _____

MC/Visa _____

Exp. Date _____ MC

2055 Woodside Rd. #180,
Redwood City, CA 94061
415-364-0108 BBS: 415-367-1029

MCI Mail: Tony Bove
CompuServe: 70105,722
The Well: t-c

DESKTOP PUBLISHING: for anyone who publishes information using computers.

Pascal Procedures

By John P. Jones

6245 Columbia Ave.
St. Louis MO 63139
(314) 343-8422

John called right after issue #27 had gone to bed (to typesetting) really excited about his idea for this column (he was very early but the thought couldn't wait). His struggles with inline code in Turbo were so bothersome that he had begun looking for alternatives. He called because he had found a solution. The result is this really close look at the relationship between a popular compiler and a popular assembler.

As fast as Turbo Pascal is, there are times when it can't keep up with time critical jobs. At these times, the only answer is to use a machine language (ML) routine. Borland offers two mechanisms which allow you to interface between Pascal and ML — EXTERNAL procedures and INLINE machine code.

In 16-bit systems (CP/M-86 and PC/MS-DOS), the Pascal runtime system loads EXTERNAL subprograms from the disk file specified in the external procedure statement. The only restriction is that the executable code in the file must be relocatable. The interface is thus fairly easy and painless.

For the CP/M-80 version of Turbo, EXTERNAL routines must be specified with their address. By using Turbo's End subcommand, an area above the program can be reserved for machine code to be loaded at runtime. See Pascal Procedures in Micro Cornucopia issue #17 for an example of this method. Although a bit cumbersome, this is probably the best method for large machine code files.

Low Level Input, High Level Language

My topic this time is the other technique, INLINE machine code for CP/M-80. The syntax for the INLINE statement is:

```
inline ( $10/ 1234/ count/ name+3);
```

The reserved word INLINE is followed by a series of code elements, enclosed in parentheses, separated by slashes. Code elements can be numeric constants, symbolic names or expressions involving constants, symbols, and the current location counter (indi-

cated by '*'). Only addition and subtraction operators are allowed in expressions.

For short routines, hand assembling and entering the hex values is not too painful. It would be nice, however, to automate both the generation of machine code and its inclusion in a Pascal program.

Assembly language (AL) is just one level above hand coding of ML. Every ML instruction has a mnemonic AL representation. These are converted into ML by an assembler, which also allows symbolic references to addresses and constants.

Problem: When you write an AL routine for Turbo, you have no idea where in memory the compiler will want to put it. Conversely, you don't know where to tell the assembler to find any variables or constants defined by the Pascal compiler.

Solution: Use a relocating assembler. Rather than generate machine code that resides at a specific address, a relocating assembler outputs what is called a relocatable object module (a REL file). Information in the REL file allows another program, the linker, to modify the object code for location anywhere in memory. In addition, relocating assemblers allow you to specify symbols as external, e.g. defined outside the current module.

So far, we haven't gained much, since REL files are not easy to analyze, and we don't really want to write a full linker for Turbo.

If You Can Read It, Turbo Can

As it turns out, the listing file created by Microsoft's M80 contains ALL the information we need to generate a valid Inline statement for Turbo Pascal. (I looked at four relocating assemblers; the other three all had one or more key pieces of information missing from the listing file.) Also, since the listing is a text file, the program to convert to an includable inline procedure is easier.

Listing 1 is the M80 listing file for a routine to read data from a hypothetical 12-bit analog to digital converter.

Editor's note: Listings 1,2,3, and 4 are all found on the Micro C bulletin board, 503-

382-7643. Only Listing 2 is printed in its entirety in this article.

Note that all code bytes generated are in the listing — some assemblers truncate the listing of things like quoted strings to what will fit on one line. All code is represented by either two hex digits with a following space, or four hex digits, high byte first. This is how a 16-bit hex value is normally written, but not how it is stored in memory. All 16-bit values defined externally are flagged with an '*' suffix, and those that are relative (to the start of the module) are suffixed with an apostrophe.

What you can't SEE in the listing is that the AL source has been copied verbatim from the input, that each page is preceded with a form feed (ctrl-L), and that the assembler output (everything up to the source part of each line) uses spaces (not tabs) as separators. This last means that we can count on fixed field positions on each line for the analysis.

Restrictions

By using a few simple rules in the AL and Pascal programs, the abilities of both the assembler and compiler can be used to complement one another and simplify the Pascal source code generator program.

Assembler:

1. There should be no code-generating statements before the first comment, and the first comment should be the Pascal source for the procedure/function heading. Subsequent comments should contain any Pascal declarations for constants and variables. The first non-comment line will terminate this section. There can be no Pascal code-generating statements in this section. This requirement greatly simplifies the conversion program.

2. All data areas and variables should be declared in the Pascal and made known to M80 with EXTRN statements.

3. You can't use directives to M80 that generate non-relocatable code (like .PHASE) or non code-relative code (DSEG).

4. Expressions evaluated by M80

must generate either correct absolute values or relocatable values.

5. There must be an "end" statement — it is used by the conversion program as well as the assembler.

6. NO ASSEMBLER ERRORS!

Pascal:

1. All declarations passed to M80 via EXTRN statements should be unique within the first 6 characters.

2. Symbols passed in EXTRN statements must be defined before the INLINE code is included.

3. Constants passed as external references MUST evaluate to 16 bits, or Turbo may not generate the correct INLINE code. Also, M80 understands only 16-bit externals.

Critical Z80 Instructions

Most of the hex data can be copied to the Pascal almost verbatim — all op codes, quoted strings, and absolute addresses or constants. Relocatable and external references need some additional work. The Z80 assembler instructions that can generate these references are shown in Figure 1.

the label enclosed in parentheses — we can copy that label directly to the inline code. The second group has the label or data following a comma — we can copy the word following the comma to the output. This leaves just the unconditional CALL and JP instructions. The program skips any label, skips the operator, then copies the next word to the output.

Inline Code Generator

The program (Listing 2) accepts an M80 listing file as input, and outputs a Pascal source file (see Listing 3 on bulletin board) which can be \$Included or read in with the Turbo editor. References external to the AL routine are supported, and all internal labels are resolved for the compiler. Assuming that all external references are known to Turbo, the resulting code will compile correctly without any manual changes. The output is formatted for readability, and the assembler source is included as Pascal comments.

There are several assumptions which are critical to the program's operation. First, the restrictions outlined above

not necessarily universal. If you take the tab expansion out, the "end" test will have to be modified. On first entry, the prefix string is set to "\$" as the first byte needs no '/' separator. The prefix is then changed to '/\$' for all later hex values.

PROCESS_EXT is very sensitive to proper syntax. The ':' is used to locate the end of a label (M80 requires the colon), and normal separators like '(' and ',' are also used as markers.

Finally, for earlier versions of Turbo, you'll have to replace the two EXIT statements in function OPEN_FILES with GOTOS.

It's Not Perfect Yet...

Global references are the only other useful facility the assembler-linker combination provides. Individual labels within the AL can be declared global and thus callable from other modules. It is the complement to the EXTRN declaration; an external label is declared global in another module.

I haven't yet come up with a clean way to let Turbo know about labels defined in an AL module. If you do, please let me know. One not so clean method is used in Listing 4 (available on the Micro C bulletin board). It involves setting up a JP table in the AL module with the requested routine number passed as a parameter from Turbo.

That's All, Folks

This project has been fun, and the result extends the usefulness of Turbo Pascal to projects that otherwise would be unsuitable.

(See Listing 2 on page 72)



Correction

Issue 27's Pascal Procedures Listing 1 was not pasted up correctly. On page 62, the center third of the listing beginning with "At this point..." and ending at "end." should be inserted just before "End of Listing" on page 63.

Figure 1

ld	a,(label)	;load accumulator from memory
ld	(label),a	;load memory from accumulator
ld	rp, data	;load register pair immediate
ld	rp,(label)	;load register pair from memory
ld	(label),rp	;load memory from register pair
jp	label	;unconditional jump
jp	cc,label	;jump if condition code
call	label	;unconditional call
call	cc,label	;call if condition code
ld	r,data	;load register immediate

M80 flags the last of these with a relocation error if "data" evaluates to an external or relative value.

If "label" or "data" evaluates to a relocatable value, conversion for Turbo is easy. The address in the listing is really an offset from the start of the module, and Turbo can be instructed to add this value to the absolute address of the start of the module.

External references are only a little more difficult. Look again at the list of instructions. They fall into three distinct syntax groups. One group has

must be followed. If you don't, the output will not compile correctly. Second, the program assumes that the AL "end" statement is in lower case. Change the program if you use upper or mixed case. Also, proper Z80 syntax for the AL is critical.

In addition to the comments there are a couple of other points, especially for those who want to modify the program for another assembler.

In procedure PROCESS_LINE, a page header is detected by a form feed (ctrl-L) at the beginning of the line,

```

{$v- Allow any length string as procedure parameter}
program include_from_m80;

[ Written for CP/M-80 Turbo Pascal Version 3

  Author: John P. Jones

  October 1985 ]

[ This program takes as input a listing file generated by
Microsoft's M80 assembler and outputs a Pascal source
file. The Pascal procedure/function heading and declarations
are assumed to be coded as comments in the beginning of the
assembler source. Identifiers defined in Pascal are passed
to M80 with an 'extrn' statement and all internal labels
are resolved for Pascal. ]

const
  code_start = 11; { M80 listing column where code starts }
  asm_start = 33; { M80 listing column where asm source starts }
  maxinp = 132; { length of max input line }
type
  anystr = string[255];
var
  m80_lst, pas_include : text;
  continue : char;

function open_files : boolean;

[ Prompt for input listing file name. If null input, exit. If no
extension specified, add the default extension '.PRN'. Prompt
for output file name. If null input, use input file name with
file extension '.INC'. If input with no extension, add default
extension '.INC'. Check for existence of input file, if does
not exist, abort. Open both input and output files.
Returns TRUE if everything OK, else returns FALSE. ]

var
  m80_name, include_name : string[16];
begin
  writeln;
  write('M80 listing file name: ');
  readln(m80_name);
  if length(m80_name) = 0 then
    begin
      open_files := false;
      exit;
    end;
  if pos('.',m80_name) = 0 then m80_name := m80_name + '.prn';
  writeln;
  write('Pascal output name: ');
  readln(include_name);
  if length(include_name) = 0 then
    include_name := copy(m80_name,1,pos('.',m80_name)-1) + '.inc';

  {$i- disable runtime I/O errors while check for input }
  assign(m80_lst,m80_name);
  reset(m80_lst);
  {$i+ re-enable runtime I/O check }

  if ioresult <> 0 then { reset of file OK? }
    begin
      writeln('^g,No input file!');
      open_files := false;
      exit;
    end;
  assign(pas_include,include_name);
  rewrite(pas_include);
  open_files := true;
end; { function open_files }

procedure process_files;
const
  hex : set of char = ['0'..'9','A'..'F'];
  whitespace : set of char = ['I',' '];
  prefix : string[3] = ' *';
type
  procname = string[20];
var
  proclist : array of procname;
  procstart : procname;
  inpline : string[maxinp];
  outpline : string[127];
  chptr : integer;
  inheader, done : boolean;

procedure scan(var st,len : integer; str : anystr);

[ Scan str for next 'word' starting at position st. Skip
leading whitespace, then count chars til next whitespace.
Returns pointer to beginning of word and length of word. ]

var
  wordend, eos : integer;
begin
  eos := length(str);
  while (str[st] in whitespace) and (st < eos) do st := succ(st);
  len := 0;
  wordend := st;
  while (not(str[wordend] in whitespace)) and (wordend < eos) do
    begin
      wordend := succ(wordend);
      len := succ(len);
    end;
end; { procedure scan }

procedure get_proc_name ( var pn : procname);

[ Extract procedure name for included procedure from first
assembly language comment line. Assumes normal syntax for
procedure header. Skips keyword procedure/function then
returns a string representing the procedure name ]

var
  st,len : integer;
begin
  st := asm_start + 1;
  scan(st,len,inpline); { skip procedure/function }
  st := st + len;
  scan(st,len,inpline);
  pn := copy(inpline,st,len);
end; { procedure get_proc_name }

function process_line : boolean;
[ process one line of input ]

const
  codelen = 12; { length of M80 hex code field }
  srclen = 99; { max length of assembler source }
  relmarker = '!!!';
  extrnmarker = '##';
var
  code_copy : string[24];
  src_copy : string[srclen];
  lastword : integer;
  endpos : integer;

procedure process_ext;

[ Process external reference. Make local copy of asm source,
then add '/' delimiter to output line. Strip comment from
source. If external ref in parentheses, just copy its name
to output. If not in parens, check for conditional JP or
CALL by check for comma. If comma, copy name from comma
to end of source line. If no paren or comma, is unconditional
JP or CALL so skip potential label, skip operator (JP or CALL) then
copy next word as external label. ]

var
  srcline : string[srclen];
  st,len,left : integer;
begin
  srcline := src_copy + ' ';
  outpline := outpline + '/' + ' ';
  left := pos(':',srcline);
  if left <> 0 then srcline := copy(srcline,1,left-1);
  left := pos('(',srcline);
  if left > 0 then
    begin
      len := pos(')',srcline) - (left + 1);
      outpline := outpline + copy(srcline,left+1,len);
    end
  else begin
    left := pos(':',srcline);
    if left > 0 then
      begin
        st := left+1;
        scan(st,len,srcline);
      end
    else begin
      st := pos(':',srcline)+1; { skip potential label }
      scan(st,len,srcline); { get & skip operator }
      st := st + len;
      scan(st,len,srcline);
    end;
    outpline := outpline + copy(srcline,st,len);
  end;
end; { procedure process_ext }

procedure process_code;

[ Process code portion of input line. Make local copy
of code portion of input line then sequentially scan
code for hex 'words'. If word is marked as relocatable
with apostrophe suffix, copy word to output and append
'+' and procedure name so Turbo can calculate absolute
address. If external reference, call process_ext to
extract and append proper label. Finally, if is an
absolute value, copy to output line. ]

var
  st,len : integer;
begin
  code_copy := copy(inpline,code_start,codelen);
  st := 1;
  len := 0;
  while st < length(code_copy) do
    begin
      scan(st,len,code_copy);
      if len > 0 then
        begin
          if code_copy[st+len-1] = relmarker
            then outpline := outpline +
              prefix+copy(code_copy,st,len-1)+'+'+procstart
            else if code_copy[st+len-1] = extrnmarker
            then process_ext
            else outpline := outpline + prefix + copy(code_copy,st,len);
          st := st + len;
        end;
      end;
    end;
end; { procedure process_code }

```

```

begin { procedure process_line }

{ First check for no assembler source, or page header line.
If is valid line, make local copy of source portion and
expand tabs so output more readable. Check for assembler
'end' statement, if is, processing of file is done. Next
check if is hex code present and if is, call process_code
to generate proper output. Finally, do some output
formatting and copy assembler source to output as a Pascal
comment. }

src_copy := '';
outline := '';
if (length(inline) >= asm_start) and { source present ? }
not (inpline[1] = 'L') then { listing header ? }
begin
src_copy := copy(inpline,asm_start,srclen);
while pos(' ',src_copy) < 0 do
begin
while pos(' ',src_copy) mod 8 < 0 do
insert(' ',src_copy,pos(' ',src_copy));
delete(src_copy,pos(' ',src_copy),1);
end;
{ terminate source with space so following search will work }
src_copy := src_copy + ' ';
process_line := pos('end ',src_copy) < 0;
end;
if inpline[code_start] in hex { hex code present ? }
then begin
process_code;
prefix := '/ $'; { need different prefix for all succeeding bytes }
end;
while length(outline) < 30 do outline := outline + ' ';
if (src_copy < ' ') and (length(src_copy) > 2)
then outline := outline + '[' + src_copy + '];
{ split output line if wider than screen }
if length(outline) > 79 then
begin
lastword := 79;
while not (outline[lastword] in whitespace) do
lastword := pred(lastword);
writeln(pas_include,copy(outpline,1,lastword));
writeln(pas_include,copy(outpline,lastword,255):lastword);
end
else
writeln(pas_include,outline);
end;
{ procedure process_line }

begin { procedure process_files }

{ First read and copy the Pascal source header coded as comments
in beginning of assembler source after skipping all leading non-
comment lines. Copy required 'begin' and 'inline (' to output
then process input line by line until 'end' statement detected.
Finally write closing parenthesis and 'end;' to output. }

inheader := false;
while not inheader do
begin
readln(m80_lst,inpline);
if length(inpline) >= asm_start
then if inpline[asm_start] = ';' then inheader := true;
end;
get_proc_name(procstart);
while inheader do
begin
writeln(pas_include,copy(inpline,asm_start+1,255));
readln(m80_lst,inpline);
inheader := (length(inpline) >= asm_start)
and (inpline[asm_start] = ';');
end;
writeln(pas_include,'begin');
writeln(pas_include,' inline(');

repeat
done := process_line;
readln(m80_lst,inpline);
until done;

writeln(pas_include,')');
writeln(pas_include,'end;');
end;
{ process_files }

procedure close_files;
begin
close(pas_include);
close(m80_lst);
end;

begin { main program }

repeat
if open_files then { get filenames and open if input present }
begin
process_files;
close_files;
end;
writeln;
write('Convert another? '); { continue with another? }
read(kbd,continue);
writeln(continue);
until not (continue in ['Y','y']);

end. { main program }

```

End of Listing

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)

OUTSIDE OKLAHOMA: NO SALES TAX

V20 CPU @ \$16.00 V30 CPU @ \$17.50
8087-2 Math Coprocessors 140.00

DYNAMIC RAM

256K	64Kx4	150 ns	\$4.75
256K	256Kx1	120 ns	3.25
256K	256Kx1	150 ns	2.47
128K	128Kx1	150 ns	3.50
64K	16Kx4	150 ns	2.75
64K	64Kx1	150 ns	1.49

EPROM

27C256	32Kx8	250 ns	\$7.50
27256	32Kx8	250 ns	4.75
27C64	8Kx8	250 ns	3.75
2764	8Kx8	250 ns	2.50

STATIC RAM

6264LP-15	8Kx8	150 ns	\$2.99
-----------	------	--------	--------

640 Kbyte MOTHERBOARD KITS: IBM PC/XT
Zenith 150, Compaq Portable & Plus: \$64.46

QUANTITY ONE PRICES SHOWN
100 ns 256Kx1 D-RAM @ \$5.15

OPEN 7 DAYS: WE CAN SHIP VIA FED-EX ON SAT.

NO EXTRA
COST FOR
F-EX SAT
DELIVERY
ON ORDERS
RECEIVED BY
Th: Std Air
Fr: P-One

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts $\mu P\infty$
MICROPROCESSORS UNLIMITED, INC.
24,000 S. Peoria Ave., (918) 267-4961
BEGGS, OK. 74421

Prices shown above are for Jan. 27, 1986

Please call for current prices. Prices subject to change. Please expect higher or lower prices on some parts due to supply & demand and our changing costs. Shipping & insurance extra. Cash discount prices shown. Orders received by 6 PM CST can usually be delivered to you by the next morning, via Federal Express Standard Air @ \$6.00, or Priority One @ \$13.00!

INEXPENSIVE CP/M PRODUCTS

- Featuring:
- New warranted equipment only
 - Full product support
 - Fix for CP/M 8MB barrier
 - Special dealer and volume pricing

HARD DISK UPGRADE KITS

Internally installed

10MB	\$ 995	40MB	\$2,550
20MB	\$1,550	70MB	\$4,800

MAJOR SYSTEM UPGRADE FOR Z80 COMPUTERS — CP/M 3.0

CP/M 3.0 is a powerful upgrade system to CP/M 1.0 - 2.2 for those who require a fast and sophisticated programming, data processing and file serving system in stand alone or multiuser environments.

Features include Hitachi 64180 processor with NCR 5380 SCSI interface.

QUAD DENSITY FLOPPY DRIVE SUPPORT

Drives and PROM's available for any configuration including above listed hard disk kits.

FORBES ENGINEERING

4023 N. 38th Ave.

Phoenix, AZ 85019

CALL (602) 269-9514

ED/ASM-86

The first truly integrated Editor, Assembler, Debugger, and Linker is now available from Oliver Computing Company.

* The Editor supports block move, block copy, partial save and partial load insertion. All text is tokenized for both efficiency of storage and very fast assembly time. FAST screen I/O and separate editor/debug windows.

* The Assembler can be immediately invoked after any editing command. Output can be directed into memory, to a .COM file, .EXE, or to ED/ASM-86's own efficient .LNK file format. 8087/186/286 instructions supported. Macros and most standard pseudo-ops are supported.

UNIQUE to ED/ASM is its structured assembly language capability. No more wasted labels and time! Multiple segmented programs can be sent to .COM files.

* The built in debugger supports symbolic disassembly, integration with the assembler, as well as disassembly of 8087/186/286 instructions. Display 8087 registers in scientific decimal format. SPLIT SCREEN debugging with a dynamic register window is easily invoked. Disassemble programs to disk file for easier analysis. "Single-step" data editing supported for bytes, words, double words, and 8087 floating constants in memory. You can "patch" assemble a line at a time with symbols from previous assembly (even define symbols as you go!)

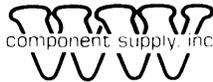
* The linker supports ED/ASM-86 link files, which can be created from virtually any valid ED/ASM-86 source file, even lines with multiple external symbols in expressions. (Note: not compatible with usual link format)

* And many more features too extensive to be listed here.

All of these features exist in one integrated program. A typical development cycle with ED/ASM-86 consists of entering a program with the editor, or editing an existing program, typing "A:,-IM" to assemble into memory, going into the debugger to immediately test the program. Then you go back to the editor for the next cycle.

ED/ASM was developed for Oliver Computing's own use, and has been used for over a year to enhance itself!

ED/ASM-86 is only \$95. Send check or M.O. to:
Oliver Computing Company, P.O. Box 90140, Indianapolis, IN 46290 { (317) 849-4450 }
for immediate delivery via UPS.



1771 Junction Ave. • San Jose, CA 95112 • (408) 295-7171

Hours: 10:00 am - 5:00 pm / MON-FRI

Shipping charged on all orders. Minimum order: \$15.00. Please call on all orders as items are limited to quantities on hand. Prices subject to change without notice. **C.O.D., Prepaid or Master Card or Visa orders only.**
NO OPEN ACCOUNTS / NO FOREIGN ORDERS, PLEASE!

POWER SUPPLIES:

ACDC OEM 5N17-1, 5V@17A, 10"x5"x5", 14lbs.	\$45.00
Trygon LBS6-24-OV, 4.8-6.8V@24A, 6"x17"x8", 36 lbs.	\$60.00
Lambda LXS-D-5-OV, 5V@27.5A 7 1/2"x7 1/2"x9 1/4", 22 lbs.	\$60.00
Sorenson SSD9-50, 9V@50A switcher, 7 1/2"x5 1/2"x12 1/4", 13 lbs.	\$175.00
Standard Power 200 B 12, 12V@5A unregulated 4 1/2"x8 1/4"x5 1/2", 12 lbs.	\$20.00
ACDC 15N5-1, 15V@5A, 10"x4"x5", 6lbs.	\$75.00
Standard Power 200 B 16, 16V@12.5A unregulated 4 1/2"x8 1/4"x5 1/2", 12 lbs.	\$15.00
Sorenson SLC20-5, 20V@5A, 5 1/4"x3 1/2"x8 1/2", 4 lbs.	\$40.00
Sorenson SSD24-8.5, 24V@8.5A 5 1/4"x3 1/2"x10", 5 lbs.	\$90.00
Astec AA11261, 5V@6A 28V@1A, unregulated switcher, 7 1/2"x4 1/2"x2"	\$50.00
130W power supply, IBM PC compatible	\$140.00
HB5-3, 5V@3A Linear, 4"x5"x2"	\$15.00
CP411, 5V@8A, -5V@3A, +12V@1.2A, -12V@1A, +24V@2.5A, 5A(surge)	\$35.00
16 1/2"x5"x3" Linear, (Good for 8" Hard Disks, and 5 1/4" Floppies)	\$35.00
CP417, 5V@4A, 12V@2.5A(Avg.) 7.5A(Surge) Linear, 8 1/2"x5"x3" (Good for 5 1/4" Hard Disks)	\$50.00
ACDC, 551-111, 5V@8A, +12V@1.7A Linear, 11"x5"x3" (Good for 5 1/4" Floppies, or Low Current Hard Disks)	\$35.00
Osborne Power Supply +5V@2.5A, +12V@2.02A, -12V@1A Switcher, 6 1/4"x4"x2" (Good for 5 1/4" Floppies)	\$18.00

CASES WITH POWER SUPPLIES:

For one full ht., or two half ht. 5 1/4" floppies.	\$96.00
For one full ht. or two low pwr. half ht. hard disks.	\$200.00

DISK DRIVE BONANZA:

DISK DRIVES:

SURPLUS Tandon TM501-full ht., 5 megabyte formatted. One platter, two heads, access: 85 msec., 612 tracks, 306 cylinders, +12V@1.5 A typical, 5 max., 5V@.8A	\$119.00
SURPLUS Tandon TM502-full ht., 10 megabyte formatted. Two platters, four heads, access: 85 msec., 1,224 tracks, 306 cylinders	\$225.00
SURPLUS TM 503-full ht., 15 megabyte formatted. Three platters, six heads, access: 85 msec., 1,836 tracks, 306 cylinders	\$340.00
SURPLUS TM 602S-full ht., 5 megabyte formatted. Two platter, access: 99 msec. 612 tracks, 1 cylinder	\$95.00
SURPLUS Tandon TM 603SE-full ht., 11 megabyte formatted. Three platter, six head, access: 137 msec., 1,380 tracks, 230 cylinders	\$225.00
NEW Tulin TL226-half ht. 20 megabyte	\$500.00
NEW Tulin TL240-half ht., 32 megabyte	\$800.00
NEW Microscience HH725-hlf. ht., 20 megabyte	\$512.00
NEW Microscience HH612-hlf. ht., 10 megabyte	\$380.00
NEW TEAC SD510, half ht., 10 megabyte, requires 130W power supply for P.C.	\$380.00
NEW Shugart SA455/Panasonic JA55112N-5 1/4", half height, 48TPI, 40 track, 6msec. double sided	\$99.00
NEW TEAC FD55BV-48 TPI, 40 track, 6msec., half height, double sided	\$109.00
NEW TEAC FD55FV-5 1/4", half height, 96TPI, 80 track, 3msec., double sided	\$149.00
NEW TEAC FD55 GFV-AT-5 1/4", 1.2 megabyte floppy for IBM PC-AT	\$175.00
NEW TEAC FD35F-3 1/2", 80 track, 135 TPI, double sided	\$180.00

DISK DRIVES: (continued)

NEW TEAC FD35E-3 1/2", 80 track, 135 TPI, single sided,	\$168.00
SURPLUS Shugart SA450-5 1/4", full height, 48 TPI, 40 track, 25 msec. track to track, double sided	\$75.00

HARD DISK CONTROLLER CARDS:

NEW DTC 5150 BX, for IBM PC- Without Data Cables	\$185.00
With Data Cables	\$195.00
NEW DTC 5250-will run two 5 1/4" floppies and two 5 1/4" hard disks. For example two TEAC FD55B, one Microscience HH725 20 megabyte fixed hard disk, and one Syquest SQ306R 5 megabyte removable hard disk.	\$360.00
NEW DTC 5290-For IBM PC-AT, will run 5 1/4" floppies and hard disks	\$325.00
NEW Western Digital WD1002-WX2-for IBM PC Without Data Cables	\$170.00
With Data Cables	\$180.00
NEW WD W1002-WA2-for IBM PC-AT Will run 5 1/4" floppies and hard disks	\$299.00

* All drives with asterisk are compatible with the IBM PCTM as 360K or 380K drives. Half height drives may require mounting brackets or filler plates not supplied with the drive.

When ordering controller cards please indicate the model of drive you will be using, and the number of heads, cylinders and formatted capacity of the drive.

All drives carry a 90-day guarantee.

GAVILAN PARTS:

32K RAM Capsules	20.00
64K RAM Capsules (no plastic)	40.00
	3/100.00
Thermal Paper	3.50
Boot Rom Board with 56K Ram Drive	65.00
Printer Battery Packs	13.00

Designing A Communications Program For Data Access

By Michael Bazaraal and Teri Tatham

2608 E 121st St.
Cleveland OH 44120

No matter what language you use, if you've ever considered writing your own version of a modem program you should take a peek at the following. Michael and Teri have provided an interesting start to just about anything you'd need to do.

We designed the following data communications program in FORTH to run on a number of different micros. The program accesses BRS (on UNINET), the lowest-cost source of on-line scientific literature which includes, for example, MEDLINE, a database compiled by the National Library of Medicine.

How The Connection Works

In many cases, you aren't talking directly to a computer when talking to a value-added network (VAN); you're really talking to a communications node called a PAD (packet assembly and disassembly facility).

The PAD takes characters from a terminal (your micro) and puts them together into a packet of perhaps 20 characters to send to the mainframe. The mainframe returns information as packets — each with error detection

information and an address. The PAD disassembles the packets and sends the information to you as individual characters.

This packet/character process is an efficient way to transfer data, but causes one problem. When you tell the PAD to stop sending characters (you send it an X-off) it passes that command onto the mainframe, but in the meantime the mainframe may have already sent you another packet. So you may receive information after you've said "stop."

Software Overview

Protocol requirements are simple. Data is sent or received at 300 or 1200 baud with an asynchronous protocol consisting of 7 data bits, 1 stop bit and even parity, or 8 data bits, 1 stop bit and no parity.

Your system must, of course, be able to send an X-on (11 hex) and an X-off (13 hex). X-off (ctrl-S) stops the other system, and X-on (ctrl-Q) restarts it.

You also need to be able to send a "break," which tells the computer at BRS to stop its current task and get ready to receive new commands. This is handy when a long listing is turning

out to be a waste of time.

A break is sent by keeping the outgoing signal line in the "mark" condition (low) for about 10 bit times. The communications chip in your computer — generally, a universal synchronous or asynchronous receiver and transmitter (USART) — must know how to generate a break.

Saving Text

The safest way to save incoming data is to use a RAM buffer. Once you have the data (or the buffer is full) then you transfer it to disk.

A reasonable size for the buffer is 30K. Longer listings may be accommodated by using an X-off command when the buffer is full.

Null Of That

Your computer also must be able to filter out incoming nulls (unless it already ignores them). UNINET appends 21 nulls (00h) to each line of data (Figure 1). Originally this was done to give a printer time to do a carriage return and a line feed. Nowadays nulls aren't usually needed (though some systems still need one

(continued next page)

Figure 1 - Hex Dump of a Sample of Text with Nulls Removed

```

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
7A80 00 00 00 00 00 54 59 50 45 20 49 4E 20 53 45 41 .....TYPE IN SEA
7A90 52 43 48 20 54 45 52 4D 53 20 4F 52 20 45 4E 54 .....RCH TERMS OR ENT
7AA0 45 52 20 43 4F 4D 4D 41 4E 44 0D 00 00 00 00 00 .....ER COMMAND.....
7AB0 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
7AC0 00 53 31 20 2D 2D 3E 20 20 20 50 52 4F 54 41 .....S1 --> PROT A
7AD0 4D 49 4E 45 0D 00 00 00 00 00 00 0A 00 00 00 00 .....MINE.....
7AE0 00 00 00 0D 00 00 00 00 00 00 00 00 00 00 00 00 .....
7AF0 00 00 00 00 7F .....
7B00 7F 7F 7F 41 31 20 20 20 20 20 37 33 33 20 20 .....A1 733
7B10 44 4F 43 55 4D 45 4E 54 53 20 46 4F 55 4E 44 0D .....DOCUMENTS FOUND.
7B20 00 00 00 00 00 00 0A 00 00 00 00 00 00 00 00 00 .....
7B30 00 00 00 00 00 00 54 59 50 45 20 49 4E 20 53 45 .....TYPE IN SE

```

About 40% of the characters in average text received from BRS via Uninet are nulls (00). Uninet inserts six nulls after each carriage return (0D) and 6 nulls after each line feed (0A). As of May 1985 logging on to BRS caused 9 more nulls to appear after each line feed, although this has recently changed. Thus, a total of 21 nulls were sent before every new line. The rebouts (7F) are also unnecessary, but do not form a large proportion of the material an average session.

Figure 2 - Terminal Benchmarks

	Milliseconds per Pass			
	VSX 8MHz 8086	CPM-86 MSDOS	V-4 5MHz 8088 2-80 CPM-86 CPM-80	IBM-PC 5MHz 8088 PCDOS MSDOS
0. Empty loop.	.01 .01	.03 .05	.03 .03	.03 .03
1. Move byte.	.04 .04	.10 .14	.09 .09	.09 .09
2. "Null" to CRT (EMIT).	.42 .50	1.50 2.64	.65 1.93	.65 1.93
3. "Null" to CRT via OS.	.35 .45	.87 2.32	1.92 1.83	1.92 1.83
4. "A" to CRT (EMIT).	.63 .71	1.77 2.85	.65 1.93	.65 1.93
5. Line feed (EMIT).	9.31 9.34	9.56 19.41	18.57 21.92	18.57 21.92

These computers have integral terminals. The Vector Graphics VSX was tested using LM-FORTH for MSDOS and LM-FORTH for CPM-86; the V-4 was tested using LM-FORTH for CPM-86 and LM-FORTH for CPM-80. The IBM PC was tested using LM-FORTH for PCDOS and LM-FORTH for MSDOS, both under PCDOS.

The benchmarks consisted of 65520 passes.

```

HEX
: TEST-0 FFF0 0 DO LOOP ;
: TEST-1 FFF0 0 DO 41 C000 C1 LOOP ;
: TEST-2 FFF0 0 DO 00 EMIT LOOP ;
: TEST-3 FFF0 0 DO 00 06 SMAP FDOS 2DROP LOOP ;
: TEST-4 FFF0 0 DO 41 EMIT LOOP ;
: TEST-5 FFF0 0 DO 0A EMIT LOOP ; ;S

```

TEST-0 represents the overhead of the benchmark itself.

TEST-1 should take about the same time as sending the character via a serial port (using no handshake or status checks) to an external terminal.

TEST-2 represents the time it takes to invoke the terminal function via EMIT, but display no character. However, the IBM PC treats a "null" as a "space", and takes as long to process a "null" as to process a printable character (see TEST-4 below).

TEST-3 invokes the terminal via an OS call (06 char FDOS). Result status bytes are dropped. Usually the direct OS call is faster than "EMIT" because the need for output redirection is not checked and counters are not incremented. For the IBM-PC, LM-FORTH (PC-DOS version) EMIT uses the ROM BIOS; thus for PC-DOS FORTH EMIT is faster than the OS call.

TEST-4 gives the average time to display one printable character in a string of continuous text. These computers insert a line feed (see TEST-5) and carriage return when the right margin of the CRT is reached. In the Vector Graphics computers TEST-3 is faster than TEST-4, since "nulls" are not sent to the video RAM and thus line feeds are not generated.

TEST-5 was run starting with the cursor at the bottom of the screen, so for every pass all characters in video RAM were shifted by 80. The IBM PC is slow in this benchmark since it must move 2K of attribute information in addition to moving 2K of characters with each linefeed.

These benchmarks illustrate that the execution speed of terminal functions in machines with integral terminals is dependent on not only the processor and clock speed, but also on the operating system and the specific FORTH version.

or two). However, unless properly supervised, nulls can mess up the way data is displayed.

Data Rates

Characters arrive at up to 120 per second (1200 baud) or one character every 8.3 ms. This isn't usually a problem since most CPUs can read a character from an input port in a microsecond or two. So it would appear there is plenty of time to read a character from the serial port, check to see if it's a null, store it, display it, scan the keyboard buffer, and then get the next character from the serial port.

In assembly language the whole process might require 200 microseconds; FORTH might take 10 times as long, or 2 milliseconds. Even so, we still have plenty of time.

These estimates are reasonable for computers using a separate terminal. But in some micros, such as Apple or IBM PC, the CPU may have to do CRT functions as well.

The line feed is the most time-consuming terminal function, requiring 9 or more milliseconds (Figure 2). So the distracted processor may lose some characters unless the serial port can interrupt it.

If the software supports interrupts, then an incoming character can interrupt the terminal routine, forcing the CPU to grab it and save it before finishing the terminal function.

Interrupt-driven input thus allows a slow computer to dependably capture the character following, for example, a line-feed. However, programs which use interrupts are often hardware specific, so we'll not go that route.

CUA Portable Program

The PAD gives you an easy alternative to interrupts. You can tell it to send one or two nulls after every line feed so your computer doesn't miss any important information. (The PAD can also accept any user defined character as a break so you don't have to

convince your hardware to output that long low.)

Multi-System FORTH

FORTH runs on a variety of operating systems. Words used to write disk files in the format of a specific operating system are not part of the FORTH standard, but implementations of FORTH from one vendor generally employ the same syntax for disk operations regardless of the operating system.

We like FORTH-83 from Laboratory Microsystems (LM-FORTH, see Ref. 1). It comes in versions for CPM-86, CPM-80, CPM-68000, MS-DOS, and PC-DOS. Under these operating systems, identical LM-FORTH words create and write disk files. You can distribute programs written in LM-FORTH if you "seal" the program so only the application may be run.

If only the FORTH nucleus and the words required for a communications program are included, there is room

Figure 3 - FORTH Screens

```
Screen 16
0 ( Communications program )          HEX
1 04 CONSTANT DATA-PORT ( Value is machine-dependant. )
2 05 CONSTANT STATUS-PORT ( " " )
3 02 CONSTANT MASK-BYTE ( " " )
4 : DATA-OUT DATA-PORT PCI ; ( PCI - use P1 for Z80 LM-FORTH. )
5 : DATA-IN DATA-PORT PC@ ; ( P@ for Z80 LM-FORTH. )
6 : STAT-IN STATUS-PORT PC@ ; ( " " )
7 : TOG-VID ASCII # EMIT ; ( Using "#" to demarcate message. )
8 ( Reverse video may be used eg. 14 EMIT for Vector Graphic. )
9 : STOP-REMOTE-COMPUTER 13 DATA-OUT CR
10 ." * Incoming data has been stopped with a ^S. CR ;
11 6000 CONSTANT #CHAR ( Size of capture buffer. )
12 CREATE CAPTURE-BUFFER #CHAR ALLOT
13 VARIABLE CHAR# ( Current offset within CAPTURE-BUFFER. )
14 VARIABLE CAPTURE-FLAG ( True if capturing data. )
15 VARIABLE K-COUNT ( Kilobytes of CAPTURE-BUFFER filled. ) -->
```

```
Screen 17
0 ( USART set-up. Usually can be done by operating system. ) -->
1 : STAT-OUT STATUS-PORT PCI
2 : SET-CLOCK 76 13 PCI 68 11 PCI 00 11 PCI ; ( 0068=1200 BAUD )
3 : CLEAR-USART 00 STAT-OUT 00 STAT-OUT 00 STAT-OUT 00 STAT-OUT ;
4 : USART-MODE [ BINARY 01101110 HEX ] LITERAL STAT-OUT ;
5 ( 1 stop bit, 8 data bits, no parity, BAUD = CLOCK / 16 )
6 : READY [ BINARY 00110111 HEX ] LITERAL STAT-OUT ;
7 : INITIALIZE SET-CLOCK CLEAR-USART USART-MODE READY ;
8
9 : SEND-BREAK [ BINARY 00111111 HEX ] LITERAL STAT-OUT ;
10 : DELAY 3FFF 1 DO LOOP ; ( approximately 1/2 second )
11 : BREAK SEND-BREAK DELAY READY ; -->
12
13 Values are specific for the Vector-4 computer with an Intel
14 8253 baud rate generator (13h control, 11h instruction port)
15 and an 8251 USART (04h data, 05h status/control port).
```

```
Screen 18
0 ( Capture words )
1 : FULL? CHAR# @ #CHAR FF - > ; ( CHARACTER-BUFF almost full )
2
3 : CAPTURE-TOGGLE STOP-REMOTE-COMPUTER FULL?
4 IF ." * Buffer full." CR ( Avoids reopening if almost full )
5 CAPTURE-FLAG ON ( Toggle to false below. )
6 THEN
7 CAPTURE-FLAG @ 1+ 01 AND DUP CAPTURE-FLAG ! ( Toggle flag. )
8 ." * " CHAR# @ . ." Characters in buffer."
9 IF ." Storing all displayed characters."
10 ELSE ." Ceased storing displayed characters."
11 THEN CR ." * Type ^Q to resume communications." CR ;
12
13 : WIPE CAPTURE-BUFFER #CHAR 1A FILL ( Set all bytes to 1Ah. )
14 00 K-COUNT ! 00 CHAR# ! ( Zero the counters. )
15 CR ." * Buffer emptied." CR ; -->
```

```
Screen 19
0 : CAPTURE ( b -- , Send character b to CAPTURE-BUFFER. )
1
2 : CHAR# @ 3FF AND 0= ( True if CHAR# is a multiple of 1K. )
3 IF K-COUNT @ DUP TOG-VID . TOG-VID ( Display K-COUNT. )
4 1+ K-COUNT ! ( Increment K-COUNT. )
5 THEN
6
7 CHAR# @ #CHAR 1- = ( Only one space left in CAPTURE-BUFFER. )
8 IF
9 CAPTURE-TOGGLE ( To off, see preceding screen. )
10 ." * Or type ^M to see buffer operation menu." CR
11 ELSE ( CAPTURE-BUFFER is not full. )
12 CAPTURE-BUFFER CHAR# @ + C! ( Store character. )
13 1 CHAR# +! ( Increment CHAR#. )
14 THEN ; -->
15
```

```
Screen 20
0 : RECEIVE
1 STAT-IN MASK-BYTE AND ( True if character waiting at port. )
2 IF DATA-IN ( If true read character from port. )
3 7F AND DUP ( Truncate to 7 bits, then DUP. )
4 CASE
5 00 OF DROP ENDOF ( Filter nulls. )
6 13 OF DROP ENDOF ( Do not capture ^S. )
7 ( 00 OF DROP ENDOF ) ( Spare filter. )
8 7F OF DROP ENDOF ( Filter rubout. )
9 ( Continues as default case if byte passed filter. )
10 CAPTURE-FLAG @ ( Check flag. )
11 IF DUP CAPTURE
12 THEN
13 EMIT ( Regardless of capture, display passed data. )
14 ENDCASE
15 THEN ; -->
```

```
Screen 21
0 ( Make disk file. Uses words specific to LM-FORTH ) FCB FILE1
1
2 : #80h-BLOCKS CHAR# @ 80 /MOD SWAP IF 1+ THEN ;
3 ( # of 80h byte blocks of disk space needed. Round upwards. )
4
5 : STORE CR ." ** FILE1 INPUT-FILENAME CR ( Prompt user. )
6 FILE1 MAKE-FILE
7 #80h-BLOCKS 00
8 DO
9 80 I # CAPTURE-BUFFER + ( Current address in buffer. )
10 FILE1 ?BUFFER-ADDR 80 CMOVE ( Move 80h bytes to disk buf )
11 FILE1 I WRITE-RANDOM DROP ( Write. Drop result status. )
12 LOOP
13 FILE1 CLOSE-FILE DROP ( Drop result status. )
14 ." ** CHAR# @ . ." Characters written to file." CR
15 WIPE ; -->
```

for a 40K text buffer in a computer having a 64K workspace. Larger buffers can be created in 8086 machines since the 64K version of LM-FORTH allows manipulation of the code segment register, and thus the use of whatever RAM is available.

The Communications Program

Screens 16, 20, 25, and 26 (Figure 3) make up a complete, but minimal, terminal program, and will run if the references to other functions are eliminated.

Screen 26 contains the main loop, which first reads the keyboard and then the serial port. Typing ESC terminates the program.

In Screen 25, KEY gets keyboard input when ?TERMINAL returns a TRUE flag (there is a character waiting). A case statement checks for special characters (like ESC); the rest are sent to the output port.

While characters entered at the keyboard seem to appear directly on the

screen, they actually go out the serial port, are echoed back by the modem or by the PAD, and then enter via the RECEIVE word in the same way as data originating at the remote computer.

Receiving

The RECEIVE word (screen 20) checks for an incoming character at the serial port by performing a logical AND between the contents of the port status register and the mask byte; a non-zero (TRUE) result signifies the presence of a received character.

A case statement removes unwanted characters. Characters can also be translated. For example, changing line 5 to "00 = IF DROP ASCII n" will cause incoming nulls to be displayed as n's. Characters which make it through the filter are displayed and may optionally be captured in a buffer.

Machine-dependent constants are located in screen 16. Port addresses and mask byte vary among brands of com-

puters, so you must supply these. If you want reverse video add the escape sequence or code at Line 7, SCR 16.

Serial port status is not checked before transmission since outgoing characters originate as keyboard entries — slow enough that the output port won't be overrun.

Of Buffers And Disks

Screens 18, 19, 21, and 24 handle buffer and disk operations.

Before screen 21 can be loaded, the LM-FORTH words for disk operations must be loaded from utility screens supplied with the particular FORTH version.

As the capture buffer fills, it checks the character count by performing a logical AND with 3FFh (screen 19). A 00 result means a multiple of 1K. This procedure is much faster than division. Every 1,000 bytes increments a counter which is displayed on the

(continued next page)

```
Screen 22
0 ( Sends line "n". )      VARIABLE PASS# VARIABLE ECHO
1 : .AUTO ( n ---; ) NO *   ( Computes offset into block. )
2 [ DECIMAL 27 HEX ] LITERAL ( Screen # of LOGON messages. )
3 BLOCK + ( Address of screen in RAM, add offset. )
4 BEGIN DUP ( addr ) C@ ASCII # = 0= WHILE ( Quit at "#". )
5   DUP ( addr ) C@ DATA-OUT ( Send char ) 0 PASS# ! 0 ECHO !
6   BEGIN STAT-IN MASK-BYTE AND ( True if echo received. )
7   IF RECEIVE 0! ECHO ! ( Finish with current char. )
8   ELSE PASS# @ 1+ DUP PASS# ! 200 = ( test )
9   IF CR TOG-VID ." Communication failed. Check Modem. "
10  CR ." Type SERIAL to restart. " TOG-VID QUIT
11   THEN
12   THEN ECHO @ ( Try 200h times or until echo. )
13   UNTIL 1+ ( increment addr )
14 REPEAT ( Send next character until "#" encountered. )
15 DROP ( address ) 0D DATA-OUT ( Send carriage return. ) ; -->
```

```
Screen 23
0 ( LOG-ON WORDS - steps through lines of text from SCR 27 )
1
2 VARIABLE STEP 0! STEP ! ( Index to logon sequence. )
3
4 : NEXTSTEP STEP @ DUP 9 < ( Test for last line. )
5   IF ( true if 8th line or less. )
6     DUP .AUTO 1+ STEP ! ( Send a line, increment index )
7   ELSE
8     CR ." Error - there are no more Log-On messages." CR
9   THEN ; -->
```

```
Screen 24
0 : MENU ( Menu for file operations. )
1 STOP-REMOTE-COMPUTER
2
3 ." *Press the indicated key to select function." CR CR
4 ." * A --> Copy capture buffer into a disk file." CR
5 ." * B --> Erase data which was captured in buffer." CR CR
6 ." *** Press any other key to exit from this menu. *** " CR
7
8 KEY
9 CASE
10 41 ( A ) OF STORE ENDOF
11 42 ( B ) OF WIFE ENDOF
12 ." * Neither A nor B - Menu aborts. " CR ( Default case. )
13 ENDCASE
14 ." *Type ^Q to restart the remote computer." CR
15 ." *Or type ^W to return to file operations menu. " CR ; -->
```

```
Screen 25
0 ( Read keyboard ) VARIABLE ESC ( True if "ESC" key pressed. )
1
2 : KEY? ?TERMINAL ( Is a character present at keyboard? )
3 IF KEY DUP ( If there is, place on stack and DUP. )
4 CASE ( Select proper disposition of the char. )
5 1B OF ( Esc ) DROP ESC ON ENDOF ( Set esc. flag. )
6 01 OF ( ^A ) DROP CAPTURE-TOGGLE ENDOF
7 17 OF ( ^W ) DROP MENU ENDOF ( List menu, SCR 24. )
8 11 OF ( ^Q ) DATA-OUT TOG-VID ." ^Q " TOG-VID ENDOF
9 0A OF ( LF ) DROP NEXTSTEP ENDOF ( Next logon step )
10 DATA-OUT ( If none of the above, transmit character. )
11 ENDCASE
12 THEN ; -->
```

```
Screen 26
0 ( Main loop of communications program )
1
2 : SERIAL CR ( Starts terminal program. )
3 ( INITIALIZE ) ( Needed if SCR 17 is used. )
4 ESC OFF ( Set flag to FALSE. )
5 WIPE ( Clear CAPTURE-BUFFER. )
6 CAPTURE-FLAG OFF
7 ." *Ready to transmit and receive." CR
8 ." *Not capturing characters. " CR CR
9 0! STEP ! ( Initialize auto-logon index. )
10 BEGIN
11 KEY? ( Read keyboard and transmit. )
12 ESC @ ( Read escape flag. )
13 0= WHILE ( While escape flag is false. )
14 RECEIVE ( Read the port. )
15 REPEAT ; ( Back to just after BEGIN. ) DECIMAL ;S
```

```
Screen 27
0 ( Log-On values for BRS via Uninet ) -->
1 AT DT 2565248# ( Type [ CR . CR ] manually when connected. )
2 SET7200:2# ( Sets break to "B. )
3 SET79:1# ( Sets one null after CR. )
4 SET714:1# ( Sets one null after LF. )
5 OWL:A# ( Selects service. )
6 314159# ( Substitute your account #. )
7 YOURS# ( Substitute your security #. )
8 1;1;MESH;N;N# ( This line instructs BRS to skip menus. )
9
10 Use manual CR when single CR is needed. BRS will default
11 to providing 80 character lines and continuous scrolling
12 if you answer with a CR to the screen width and length
13 inquiries from BRS. If your messages end with a line other
14 than #B change line # in SCR 23 to reflect last line used.
15 SET200:2,9;1,14:1# ( Alternative to lines 2,3,and 4. )
```


this program has the advantage of having all the required functions and no options. Availability of the source code in FORTH permits extensions for special requirements.

Ruminations

After using BRS for three years, we have some advice. Initial search words are only a keyhole into the data; browsing and several iterations are required to obtain good results. BRS searches not only the keywords but also the title, author, source fields, and the abstracts. Consequently, you'll find more citations for a given search word using BRS than other on-line sources for the MEDLINE data (Ref. 3).

In some searches, the only way to exclude irrelevant references will be to read through 100 or more titles on-line. Potentially useful references can be selected for downloading in the "long" form, which usually includes a 250-word abstract. Downloaded data should be written to disk and the session promptly ended — detailed reading and all editing and printing should be done off-line. Abstracts may be all that are required, but when we need the full text we use a library only after the citations have been identified.

Acknowledgement

Daniel Amiot supplied terminal benchmark data and the program adaptation for the IBM PC.

References

1. Tello, Ernie. "PolyFORTH and PC/FORTH." BYTE, November 1984, page 303.
2. Taylor, Robert III. "Send and RCV — A FORTH Implementation of the X-MODEM Protocol." Dr. Dobb's Journal, September 1983, page 66.
3. Cortesi, Dave E. "Dr. Dobb's Clinic." Dr. Dobb's Journal, September 1984, page 12.
4. Friedman, Bernard. "Pharmaceutical Riches On-Line." PC Magazine, February 5, 1985, page 291.



OKIDATA PRINTER MICROLINE 80

7 X 9 IMPACT DOT MATRIX, NEW RIBBON
80 CPS 10 CPI 6 OR 8 LINES PER INCH
3 TO 12 INCH FORM FEED TRACTOR & FRICTION
BUILT IN SELF TEST REMOVED FROM SERVICE
PARALLEL INTERFACE (CENTRONICS COMP) \$ 99.00

WINCHESTER 20 MEG HARD DRIVE HALF HEIGHT

LOW POWER (SAME AS FLOPPY) NEW
MODEL CMI 3426 \$369.00
CONTROLLER FOR IBM PC-XT \$125.00

KEYBOARD (MICROSWITCH) 113 KEYS 8 BIT ASCII

PARALLEL OUTPUT WITH CASE
AND COMPLETE DOCUMENTATION \$40.00

SA800-2 DISK DRIVE CLEAN - USED \$59.00

SA850 DISK DRIVE CLEAN - USED \$99.00

STANDARD MOUNTING - GUARANTEED 60 DAYS

CABINET FOR ABOVE HOLDS TWO DRIVES

POWER WIRING COMPLETED (AC AND DC)
FAN AND INPUT AC WIRING DONE
20 X 9 1/2 X 17 DEEP \$60.00

POWER SUPPLY FOR TWO DRIVES (SWITCHING)

WILL POWER DRIVES AND SBC 6 X 8 X 2 1/2
5VDC @ 8 AMPS, +12VDC, -12VDC, -5VDC @
.5 AMPS EA AND 24VDC @ 2 AMPS \$50.00

CABINET AND POWER SUPPLY TOGETHER \$100.00

SHUGART DISK DRIVE PARTS

SA800 HEAD \$15.00
SA800 STEPPER \$10.00
SA800-2 LOGIC BRD NON-WORKING \$10.00
SA800-2 LOGIC BRD WORKING \$35.00
SA850 HEAD AND STEPPER ASSY (BI OR TRI) \$50.00
SA850 LOGIC BRD NON-WORKING (BI OR TRI) \$20.00
SA800 OR SA850 SPINDLE BEARINGS (SET) \$ 3.00
SA800 OR SA850 INDEX LED (SET) \$ 3.00
OTHER PARTS ON REQUEST

SA800 ALIGNMENT AND CLEANING \$20.00

SA850 ALIGNMENT AND CLEANING \$30.00

Z80A PARTS CPU, CTC, SIO/1, 1771, 2716 \$ 2.00

6845 CRT CONTROLLER \$ 2.00

DIABLO PRINT WHEELS FITS 630 AND HYTYPE

COURIER 10, ELITE 12, COURIER 72 \$ 3.00
6 FOR \$15.00

CASCADE ELECTRONICS

ROUTE 1 BOX 8
RANDOLPH MN 55065
507-645-7997

FREE
SHIPPING
ON OVER \$40.00

COD ADD \$2 CREDIT CARD ADD 5% MN ADD 6%
LIMITED TO STOCK ON HAND EVENING CALL OK

TECHNICAL TIPS

Unreliable K10 Winchesters

We've been receiving a lot of calls lately from Kaypro 10 owners who are seeing progressive deterioration of their winchester drives. The problem begins with occasional bad sectors on the drives and then progresses to the point where the drive is unusable and the data is lost.

Owners can stop this deterioration by reformatting their drives every 6 months. This means copying all the information from the winchester to floppies using mufbak, then reformatting the winchester, booting off a floppy, and then copying the system tracks and files back onto the winchester.

Unfortunately there are some bugs in mufbak (files aren't always correct after they're restored) so I recommend using PIP or SWEEP to transfer all files that are smaller than 390K. For larger files, you can take your chances with mufbak, or manually break the

files into 390K chunks (if possible). Kaypro is supposed to be working on a new version of mufbak.

Editor's note, we haven't heard of any problems using BIGBURST on disk K15. Also, SQ.COM (squeezes files) does a very good job of reducing the size of large files (USQ.COM unsqueezes them). See disk K1.

Allyn Franklin
DriveMasters
489 S Robertson Blvd
Beverly Hills CA 90211

The T <fileref> Command

I bought an Epson Geneva with 128K this summer while I was in the States, but I didn't have the cash to buy Epson's disk drive. I figured I'd get along with the cassettes and my printer until my financial status improved. When I got home, I discovered that my C. Itoh (F-40) printer had

only a parallel connection.

I had also bought disk K28 (modem730) so I could transfer files between the Geneva and my Kaypro.

The Geneva has a serial transfer program in its ROM, but I wasn't able to get anything to transfer in either direction using the R (receive) or S (send) commands (except "multiple errors").

What good is a computer if you can't get out what you put in? (Although it does look impressive on the desk.)

I finally stumbled onto the T command. So far, I have transferred a file of 9K into the memory buffer of the Kaypro and everything went smoothly.

Editor's note: Terminal mode works because there are no checking or handshaking (start-stop) characters inserted into the data or expected from the other system. Terminal mode transfers are usually limited to moving ASCII text between two systems which are directly connected. The size of the transferred file is limited to the amount of RAM buffer space available within the modem program.

Set the following on both the Kaypro and the Geneva:

baud rate (I use 1200)
 data bits = 8
 stop bits = 1
 parity = none

To set the Kaypro, you have to use an editor to change the MKP4-10.ASM file. I used WordStar. Instructions are in the .ASM file. To set the Geneva, use its CONFIG program. Everything is explained very well in the documentation.

1. Bring up your own modified and assembled version of the MKP4-10.ASM file on the Kaypro (I named mine EPSON.COM) and type in T <fileref> with a <CR>. Now that you're in terminal mode with "*** Memory buffer available ***", simply hit ^Y and received characters will be stored in the memory buffer. ^R will turn off the collection.

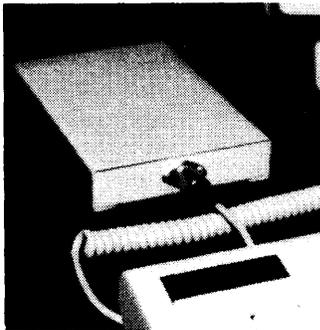
Corvatek Presents: **KEY-UP** THE KEYBOARD INTERFACE

USE AN IBM STYLE KEYBOARD ON ANY* COMPUTER!

Plug an IBM style keyboard into **CORVATEK'S KEY-UP** interface, plug the interface into your computer, and you are ready to type.

FEATURES

- Key Click Option
- Beeper
- Type Ahead Buffer
- Reprogrammable Keys (EEPROM)
- Dvorak option



KEY-UP PRICES

- DM-1 for Bigboards \$129.00
- DM-2 for Xerox 820 \$129.00
- DM-3 for Kaypro \$129.00
- DM-4 for Franklin \$129.00
- DM-5 ASCII Universal† \$129.00

KEYBOARDS

- KEYTRONICS KEYBOARDS
- KB 5150‡ \$112.00
- KB 5151§ \$164.00

  +S&H

(503) 752-4833

*** SAVE \$10.00 ***

Buy both Key-up interface and Keyboard and save \$10.00 on the total price

Order Now From:

CORVATEK 1100 N.W. VAN BUREN ST. (503) 752-4833
 CORVALLIS, OR 97330

* KEY-UP is used on any computer with a parallel input ASCII keyboard port.

‡ Same key placement as IBM keyboard.

§ Familiar typewriter key placement with separate cursor pad.

† The UNIVERSAL has all of the signals necessary for the user to adapt to any ASCII keyboard port. TTL signals on a DB-25 connector. Requires the user to build his own cable.

Key-up is a trademark of CORVATEK. Bigboard I, Bigboard II, Xerox, Keytronic, and IBM are trademarks of Digital Research Computers, Cal-Tex Computers inc., Xerox, Keytronics Corp. and International Business Machines.

TECHNICAL TIPS

2. Bring up the TERM program on the Geneva as described in the manual (very clear documentation).

3. After the file has been transferred, type ^E on the Kaypro keyboard to get back to command mode and then enter "WRT" to write the saved data to disk.

The reverse works as well. Put the Geneva into terminal mode first, then select the same mode for the Kaypro (a simple "T" from the COMMAND prompt). Once you're in Kaypro's terminal mode, hit ^T and specify the filename. (I select "Y" on the delay prompt.)

The file will appear on both screens while it is being transferred. On the Geneva each new line will write over the previous one, but it is still being saved.

Floyd Schneider
Berliner Ring 6
8047 Graz Austria

This might work for 4-84, but on my 4-83 it should be:

Procedure Motor Off

```
Begin
port[28] = port[28] OR 64
End;
```

Lewis Sternberg
535 NW 15th
Corvallis OR 97330-5809

Initializing K2000 Serial Port

There are two methods we have used to initialize the serial port. The first uses the mode statement to redirect output from the default port (Parallel or LPT1:) to the serial port, followed by another mode statement to set the parameters on the serial port.

```
MODE LPT1: = COM1:
MODE COM1:12,E,7,1,P
```

Where 12 = 1200 Baud, E = Even Parity, 7 bits/char, 1 Stop Bit, P = Infinite Retry on Busy

Note: This is only a sample; you may want to select a different set of parameters to suit the requirements of your external device. For more information on options consult your MS-DOS User's Manual.

The second method uses GW-BASIC. Write a program similar to the sample in Figure 1.

```
10 OPEN "COM1:1200,E,7,1,RS,CS1,DS1,CD1" AS #1
20 PRINT #1,"";this will print a null
30 SYSTEM";this will return to system
```

Note: This is only a sample; you may want to select a different set of parameters to suit your external device. Consult your GW-BASIC Manual for more information.

The Kaypro Corporation
Solana Beach CA

Nuts And Bolts

I had a hard time removing the screws on the monitor side of the disk drive enclosure. I tried a stubby screwdriver, but it was too long. Finally, I bought a no. 1 Phillips head drill bit with a hexagonal shaft. Costs about \$1, is only 2" long, and I can hold it in place with a finger and fit a small wrench on the hexagonal shaft. Screws come right out.

Christopher Conly
430 12th Ave E, Apt 301
Seattle WA 98102

Procedure Motor Off

Sometimes Turbo Pascal 2.0 will leave the disk drive motors running on my Kaypro 4-83. The folks at Borland gave me this procedure to turn the motors off.

Procedure Motor Off

```
const
system = $14
```

```
Begin
port[system] = port[system] and $EF;
End;
```

GRAF 3.0

\$49⁹⁵

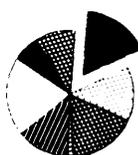
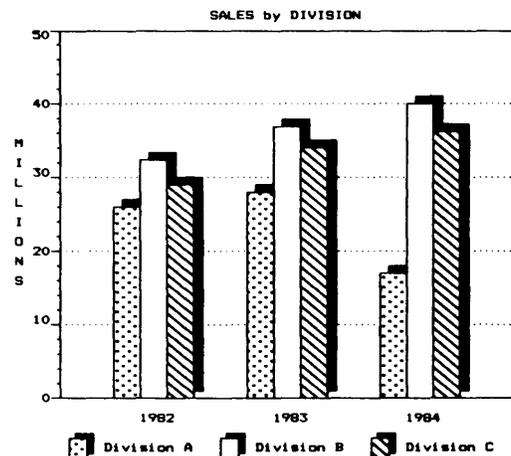
CP/M-80

\$69⁹⁵

MS-DOS / PC-DOS

the complete
BUSINESS and SCIENTIFIC
printer graphics program

- display floating point data directly from spreadsheets, data bases, word processors, and programming languages (or the keyboard) in a wide variety of bar, pie, line, and scatter plots
- plot and group up to 6 different variables on a single graph, distinguished by up to 14 different fill-in patterns and 8 different point-plotting symbols
- menu driven operation supports automatic graph scaling, labeling, and legend creation
- add up to 5 different-density grid lines, and choose from a wide variety of numerical labeling options



GRAF 2.0 Update Policy: Returning your original GRAF 2.0 disk to MSC entitles you to \$20.00 off the above prices.

TERMS: We ship via first class mail. The above prices include domestic shipping and handling. Orders outside USA require additional \$5.00 for postage. N.Y. residents add state sales tax. When ordering you MUST state your computer and printer make and model. We support MS-DOS (PC-DOS) version 2.0 or later on computers with at least 192k RAM, and CP/M-80 version 2.2 or later on 280 computers (other than modified Apples) supporting a TPA of at least 54k (requires 64k of RAM). Most soft-sector disk formats are available. (If you can read several formats, please send us a list.) GRAF 3.0 works with any printer fully compatible with one of the following: Epson FX, RX, LX, MX (with GRAFTRAX), or LQ-1500; C. Itoh Prowriter; NEC 8023A; Star Micronics Gemini 10X, 15X, SG-10, SG-15; IBM Graphics Printer; Okidata 192, and earlier Okidata models equipped with the "IBM Plug 'n Play" chips. (If you have an Okidata printer, other than the 192, the Plug 'n Play chips are required!)

MSC Microcomputer
Systems
Consultants

27 Forest Avenue

Port Jefferson Station

New York 11776-1820

© P. V. ORL. MS-DOS Version

LETTERS

Problems Down Under

I have been receiving Micro C for some months now and have found it very informative and useful. Unfortunately, the price to get it to Australia has become astronomical—almost \$45 Australian against \$16 in the U.S. Is there any chance of a special rate for the Sydney Kaypro User Group? We might get more of our members to subscribe if a reduced rate were available. If it makes a difference, the copies could be sent to a single address to reduce mailing costs.

I have a Kaypro 4 late '83 or early '84 model, 81-240A board with some chips soldered. The machine has worked very well and given no trouble whatsoever. I am having problems with the speedup and the video board I have tried to install.

The speedup kit was installed locally.

It can be switched off and some programs run at the higher speed. However, it has done strange things to some disks, corrupting files with chunks of other files. Apparently, it changed the file allocation blocks in the directory tracks. Could it be that the Monitor ROM (81-232) is causing the problem? For the moment, I am not running the computer at the higher speed.

The second problem is with the AT1 Video Attribute board from JFN Industries, Los Angeles. When installed—at normal speed—the machine will not boot up, and all I get is a flashing screen. I took the AT1 to a friend with an older (socketed) Kaypro 2 with the 81-110 board, and it worked perfectly on the first try. From the Kaypro Technical Manual, the only change I can identify is the different monitor ROM (81-149), but there is no reference to changes in the circuitry. A friend had a look with a scope and thinks there is a timing problem. Any connection with the speedup problem? I have written to JFN and expect to hear from them soon. Obviously, there is nothing wrong with the board, except that it does not work on my 4.

Hans J. Schneider
School of Geography
U.N.S.W.
Kensington NSW Australia 2033

Editor's note:

I do not know how the JFN board works. However, your drive data problem may be a slow system PIO. Try swapping the two Z80 PIOs in your system and see if the problem doesn't change.

Your monitor ROM (81-232) should be fast enough to run at 5MHz. I hope you did the CAS-MUX mod when you speeded up the system. If you didn't (many people don't), I'd do that immediately (see the Kaypro column in this issue, or see Issue #12). If your system runs OK for a while at 5MHz and starts making trouble, then it's probably a heat sensitive chip.

Fire up the system and wait until it is messing up. Next, remove the cabinet top and use ice cubes in a plastic bag to cool off groups of chips until the cooling makes the system run properly. Then heat it up again and isolate the chip that is causing the trouble.

Now let's take a look at the cost of the magazine. I agree that our foreign rates are too high, but we are barely breaking even on the Australian subscriptions as it is. Australia is one of the primary reasons we went to \$30 for foreign subscriptions. The postal service here wants nearly \$3 per copy for airmail to your area. The other option is to send the copies via surface mail (slow boat to you-know-where). You'll hear from us.

CROSS ASSEMBLERS

6502, 6800/6801, 6805, 8048, 8051
8080/8085, 8086/8088, 9900/9995

Features: Include Files
Listing Control
Cross Referencing
Nested Conditionals
Sorted Symbol Table

Disk & Manual\$100.00

Write or call for information on these and other products.

HAWTHORNE TECHNOLOGY
8836 S.E. Stark
Portland, OR 97216
(503) 254-2005

Xerox 820-1 & II

EPROM PROGRAMMER
only \$89⁹⁵

- ✓ Programs 2716, 32, 32A, 64, 64A, 128, 256, and more.
- ✓ Connects to parallel port & uses system's power supply.
- ✓ Read, copy, verify, display, edit, checksum & save or load promfiles from disk.
- ✓ Completely built and tested.
- ✓ Includes software on 5 1/4" or 8", user's manual, 40 pin ribbon cable ready to plug into the parallel port of your system.

SUPERIOR INTERNATIONAL
11663 Denton Dr. # 104
Dallas, Texas 75229
(214)484-5072

NEW

FREE CATALOG

CP/M
MSDOS

AFFORDABLE ENGINEERING SOFTWARE

TRSDOS
PCDOS

- **LOCIPRO** Root Locus — \$69.95
- **ACTFIL** Active Filter Design/Analysis — \$69.95
- **STAP** Static Thermal Analysis — \$69.95
- **MATRIX MAGIC** Matrix Manipulation — \$69.95
- **RIGHTWRITER** Proofreader & Writing Style Analyzer — \$74.95
- **ACNAP2** AC Circuit Analysis — \$69.95
- **DCNAP** DC Circuit Analysis — \$69.95
- **SPP** Signal/System Analysis — \$69.95
- **PLOTPRO** Scientific Graph Printing — \$69.95
- **PCPLOT2** High Resolution Graphics — \$69.95



Engineering
Professional Software

2200 Business Way, Suite 207 • Riverside, CA 92501 • (714) 781-0252



Shake, Rattle, And Crawl

I have a problem with vertical video wiggle, or crawl, on my Xerox 820 at about a 1Hz rate. It's not really terrible but does get irritating. Do you folks know of any fixes?

At the 1984 SOG there was talk of a ham net to discuss/implement a packet system, to be held on either 80 or 40 meters. Is that happening? If not, how about Sunday nights, 7 p.m., at 3870KHz and down? Or has the Tucson packet system ended the discussion?

Lee Chambers, WB7UEU
 1111 Archwood Drive #298
 Olympia WA 98502

Editor's note:

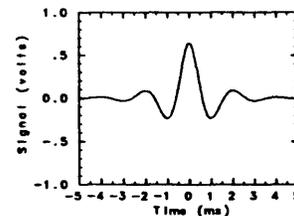
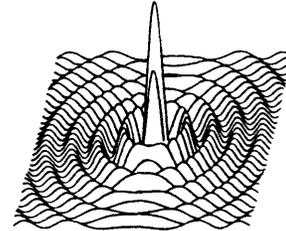
The usual reason for the video crawl is that the vertical output isn't exactly 60Hz so it beats against the 60Hz mains. If the vertical output is slightly too fast, you can usually reduce the frequency by putting a small variable capacitor (3-30pF) across the video crystal (in parallel with the leads) and then adjusting the capacitor until there is no wiggle. If the frequency is too low, then the same capacitor in series with the crystal might help.



FORTRAN GRAPHICS

MicroGlyph/SciPlottm graphics library provides FORTRAN routines for scientific and engineering applications.

- Presentation quality high resolution graphics with 240 by 216 dots/inch.
- Software generated text fonts, data centered symbols, uniform or proportional spacing, string plotting with rotation and size scaling capability, 14 sets.
- Linear or Log scaled axes.
- Vector graphics, point to point plotting with user selectable origin and data scaling.
- Multiple plot frame processing.



SYSTEM REQUIREMENTS:
 IBM PC/XT/AT 256k RAM,
 DOS 2.0, MS FORTRAN-77
 V3.2 or IBM Professional
 FORTRAN, and graphics
 printer or color card.

**TERMS: \$59.95 plus
 \$3.00 shipping.
 Mass. residents add
 5% sales tax.**

Trademarks: IBM of IBM,
 MS of Microsoft.



MicroGlyph Systems
 P.O. Box 1066
 East Arlington, MA 02174

Disk Sale

DySan

CORPORATION

TYPE	BOX OF 10
5"-SS/DD-48 TPI	19.50
5"-DS/DD-48 TPI	25.50
5"-SS/DD-96 TPI	29.50
5"-DS/DD-96 TPI	37.50
5"-DS/DD-IBM/AT	52.95
8"-SS/SD-48 TPI	23.95
8"-SS/DD-48 TPI	25.50
8"-DS/DD-48 TPI	29.95
3.5"-SS/DS	32.95

Available Soft or Hard Sector
 For Plastic Case Add 1.25/Box
 Plus Tax & Shipping

- Cash, Visa, Mastercard, COD -

Integral Systems Corp.

2900-H Longmire Drive
 College Station, TX 77840
(409) 764-8017

to host adapter

Shugart 1610 controller

50 pin

20 pin

to drive

34 pin

SASI to 5T506/412

5 1/4" HARD DISK CONTROLLERS

These new factory sealed Shugart controllers come with all documentation. They will control up to two 5 1/4" hard disk drives with up to 8 heads each. These unique controllers will mount directly on the drive. By the change of an EPROM they change their instruction set to emulate other popular controllers.

*Shugart 1610-3 emulates **Xebec** S1410
 1610-1 emulates **DTC** 510
 1610-4 SCSI version similar to **ADAPTEC** 4000

Using standard host adapters they work with:

- * **Apple** II, II+, IIE
- * **MAC** (see Sept. Dr. Dobbs)
- * **TRS 80** model III, IV
- * **ISI 5160** (PC clone)

- * **Wavemate** Bullet
- * All **AMPRO** boards (little board, etc.)
- * And other systems with SASI Ports or host adapters

1 for \$99. ea. 2 for \$90. ea. 100 for \$75. ea. Quantity pricing available.

• Computer Surplus Store (408) 248-0134 •

Other controllers also available for 8" and 14" drives including Shugart SA1000, 4000 SERIES and Quantum 2000 SERIES (DTC 1404, 1404D, 1403, 1403D, 1408I, 1420-1).

On Your Own

By David Thompson

I've heard lots of tales about shareware (or freeware). Some people say it must be great. All those pitches for money, money, money must be making a lot of software writers rich. They just put a program on their local bulletin board and almost by magic hundreds of letters bearing thousands of dollars come flooding in. It's just what chain letters promised they'd be.

The following "On Your Own" is a discussion with Bob Wallace, author of PC-Write, one of the most successful (if not the most successful) shareware programs available. I think you'll enjoy his insights.

Shareware is a unique area for software. In the past, people purchased commercial software from the author and then gave it away to friends. The shareware concept has changed that. Now people get software free, give it to their friends, and then their friends purchase it from the author. At least that's the way it's supposed to work.

I've talked to a number of shareware authors (I'd use the term freeware but it's copyrighted) and all have reported discouraging responses to their pleas for money.

The author of Castle has received several hundred letters requesting information on playing the game (a graphic adventure game for the PC) but only about \$500 in contributions. The author of QMODEM (an MS-DOS menu-driven version of MODEM7) reported that he has received only about \$1000. Both authors indicated they were sure that there were many thousands of copies in use.

I was quite surprised when I found Bob Wallace distributing his PC-Write at a Comdex booth. Here was shareware going head-to-head with the commercial guys in one of the most expensive shows there is. Quicksoft (Bob's company) was obviously more successful than the average shareware distributor.

How Does He Do It?

One key, he said, is the type of software. Productivity software (editors, spreadsheets, accounting pack-

ages, and databases) are good shareware products because they are constantly used, and owners are always on the lookout for bug fixes or new features.

Bob is continually updating his editor, adding support for new printers, new editing features, and new formatting options. He's also continually upgrading the manual, adding a quick reference card, and improving support. None of these would be very important, however, if the software were not used on a continuing basis, or if it were basically complete.

A game, a training program, or a basic utility would not work well as shareware, he felt, because the user is satisfied using it as is. Or, because he uses it and then moves on to something else. There is no ongoing need for updates.

How It Works

Bob advertises by attending shows like Comdex and advertised in many card decks. He makes most of his money selling disks of the latest version (with manual on the disk) for \$10 each. For every 100 disk orders, he'll also receive seven or eight orders for the \$75 registered-owner status. (Later, some of the \$10 purchasers will also register.)

Registered owners get: support by phone, a printed and bound version of the manual, two coupons good for updates, and a \$25 rebate for each new registration they are responsible for. 6,800 folks have registered.

Bob is not entirely sure why people register. It might be that they really

like the product; it might be the updates, or the manuals, or the \$25 kickback (though he doubts it because only one in seven claim the reward). People may also pony up because they like the idea of shareware (no heavy sales trips here). Anyway, he's planning to poll the registrees to see what they say.

Other Products

I asked Bob what other kinds of products he was working on. None. He said he intended to concentrate his effort on the editor, making it the very best he could. He thought this was better than trying to write a group of mediocre pieces.

But despite limiting himself to a single product, he isn't working alone. Quicksoft has ten employees (plus Bob): one writer, three in phone support, one in marketing, and the rest handling orders.

Marketing

Card decks have been the best method of advertising for PC-Write. If a deck costs \$1500 he expects it to bring in 100 \$10 orders and 8 or 10 \$75 registrations. Decks aimed at IBM users have been best, followed by those aimed at micro users, and then high tech managers. But even within these categories different decks will generate wildly different responses and even a successful deck may not generate a response forever.

Booths and word of mouth (word of disk) have also generated many sales. He noted that the computer shows (especially those aimed at the end

THE COMPUTER JOURNAL

Practical Programming & Hardware Projects

The Computer Journal is published bimonthly for those interested in programming their computers, interfacing to peripherals, and hardware construction.

Now expanded with in-depth articles covering Turbo Pascal, "C", Assembly Language, Kaypro, Ampro, Interfacing, plus CP/M and other operating systems.

**6 Issues (1 year) \$14 in US — VISA & MasterCard accepted
190 Sullivan Crd., Columbia Falls, MT 59912 (406) 257-9119**

user) have been a lot less productive lately. The crowds are down significantly, and those who do show up aren't buying much. Bob thinks people are waiting for the new PC-2 and the Amiga. Once these machines are out he thinks buying will turn up again.

Marketing is important, but he feels that most importantly the product has to be good. In conjunction with this he's planning to add spelling checking and support proportional spacing in the near future.

Some Suggestions

"If you decide you want to be a programmer and a businessman, then go ahead and do the package and get it out. Make sure you're doing something unique — all successful software companies are different."

He went on to note that there is a lot of room for new products and improvements to the old ones.

"Look at Lotus, for instance. It's slow and could really be improved."

Again

Before marketing a shareware product, decide whether you want to program, run a software company, or sleep. He says he enjoys programming and running a company, but he works 12 hours a day, 6 or 7 days a week.

"I take off a day every three weeks and took four days at Thanksgiving."



EPROM PROGRAMMER



APROTEK 1000
ONLY
\$265.00

COMPLETE WITH
PERSONALITY
MODULE

117 VOLT AC POWER-RS232
-6 BAUD RATES - HANDSHAKE TO HOST
ALLOWS READ, WRITE, VERIFY & COPY

Comes complete with a CPM, IBM-PC or Apple
Driver Program on Disc.

Programs the following 5 Volt 24 or 28 pin
devices: 2716 series through 27256, 25xx series,
68764 plus others. Please Specify Personality
Module desired with order. Additional Personality
Modules only \$15.00 ea. **Full 1 year warranty.**

TO ORDER CALL 1 800 962 5800 OR WRITE
APROTEK
1071 A AVENIDA ACASO Adtl
CAMARILLO CA 93010 \$4.00 Shipping USA
Info: (805) 987 2454 VISA or MC Adtl 3.00

NEW! FROM TCI

SPEED — POWER — VERSATILITY

EXPRESS 2.0

FULL SCREEN EDITOR

HIGH PERFORMANCE AT A FAIR PRICE

ONLY \$29.95

WHY WASTE YOUR VALUABLE TIME with a slow, clumsy, limited editor when you could **TAKE THE EXPRESS!**

With EXPRESS 2.0 you can. . .

- Fully Access **ALL 32 USER AREAS** with any editor command!
(Hey Winchester owners, check this out!)
- Execute **CP/M COMMANDS** without leaving the editor!
(Rename, Copy, Erase, Type, Directory, Log)
- Set up, use, and save on disk an **UNLIMITED NUMBER** of **POWERFUL KEYBOARD MACRO'S!**
- **TOTALLY RECONFIGURE THE KEYBOARD** to look like another editor if you desire so you don't have to learn new control keys!
- Use one of the **FASTEST SEARCHES EVER WRITTEN**. We take advantage of the powerful Z-80 string handling instructions. (Sorry 8080 owners, this one's too fast for you!)
- **INSTANTLY** configure **EXPRESS** for your terminal or computer using the included terminal data base. (Over 50 predefined terminals.)
- Edit **FILES LARGER THAN MEMORY** with **EASE!**
- Easily enter and edit **"CONTROL"** and **"HIGH BIT" CHARACTERS!**

Other EXPRESS 2.0 features. . .

- **DYNAMIC** word **WRAP/UNWRAP** • Full Cursor control (and then some!)
- **EASY** to **SET TABS** • **GLOBAL/SELECTIVE/LITERAL/IGNORE CASE REPLACE**
- **FULL BLOCK INSTRUCTIONS** including **BLOCK PRINT, BLOCK SAVE, & INCLUDE**
- **VARIABLE SPEED (FAST!) BI-DIRECTIONAL AUTO-SCROLL** • **GOTO PAGEN / LINE N**
- **COMPACT** - Only 24K on disk, 19K in RAM for program

EXPRESS VERSION 1.0 SAMPLER

- Version 1.0 is a sample copy available **FREE** on your local Bulletin board or for only \$10.00 from us. Version 1.0 was a very preliminary version of **EXPRESS** and did not yet have all functions implemented. However, we feel it is a good way to allow you to see for yourself the power of **EXPRESS** before you buy. Version 1.0 has the following limitations:

Only 16 user areas
Limited Keyboard MACRO's
Literal Search mode only
Manual not included

Copy, Rename, Erase not available
Dynamic word wrap unavailable
Definable "Block" size limited to 1K
File size limited to memory size

Gentlemen. . . Please rush me the following:

_____ Copies of **EXPRESS 2.0** at \$29.95 + \$5.00 postage and handling (includes manual)

_____ Copies of **EXPRESS 2.0** manual only at \$12.00 + \$3.00 postage and handling

_____ Copies of **EXPRESS** sampler (Version 1.0) at \$10.00 + \$2.00 postage and handling

I have enclosed my check or money order for \$ _____ payable to:

TCI
17733 205th Ave. NE
Woodinville, Washington 98072

Note: Washington residents must include
7.8% sales tax
Note: For C.O.D. add \$3.50

My shipping address is:

Name _____

Address _____

City _____ State _____ ZIP _____

Phone _____

My Z-80 based computer running CP/M-80 v2.2 is: _____

Disk size (5"/8") _____ Disk Density (S/D) _____ Single/Double sided _____

TOMORROW'S COMPUTING INNOVATIONS

XEROX 820 - I & II Assembled & Tested

820-I	\$50.00	☆ Z80-B 6MHz	\$ 8.00
CP/M (8")	35.00		
820 w/CP/M	70.00	☆	
820-II	70.00	☆ 5 1/4 DSDD	100.00
CP/M (8")	35.00	80 Track 96TPI	
820-II w/CP/M	90.00	☆ Phillips Drives	

XEROX is a Trademark of XEROX Corp. CP/M is a Trademark of Digital Research.

OEM HOBBYIST

Sips	Inductors	Pots	7400
Dips	Regulators	Caps	74LS
Chips	Resistors	Linears	some ECL

Tell us your needs - Send self addressed stamped envelope for our complete list.

Terms: All orders are F.O.B. Hayward Ca. Pre Payment or C.O.D. Ca Residents add sales tax. Prices subject to change without notice. Quantities limited to stock.

E² I COMPUTER PRODUCTS

2273 American Ave #8 Hayward CA 94545

[415]786-9203

OUTSTANDING PROGRAMS

For Your CP/M 2.2 System. . .

AutoSpool/80

\$39.95

Bring the power of a mainframe spooler to your CP/M 2.2 system. AutoSpool/80 features:

- Nearly unlimited storage capacity
- Multiple copies
- Printer controls, including pause for next page, resume print from middle of job, and presettable configuration of pitch and lines per inch.
- Smart file handling, including control of printing order, wild cards in file names, and individual cancelling

CHAIN2

\$15.00

At last! Run *all* your .COM files from *menus*, arranged any way you like, nested to any level! No modifications to .COM files required. CHAIN2 chains to your .COM file, then causes it to chain back to the menu when it exits. Menu program supplied.

DUMPREL

\$15.00

Displays the contents of your .REL relocatable files. These files are the output of many assemblers and compilers, used for linking into .COM files.

BANNER

\$15.00

Prints your message sideways in big block letters. Variable size characters. Requires no special printer. Composes characters with a second message.

ALL OF THE ABOVE: \$59.95

30-DAY MONEY BACK GUARANTEE

ANAHEIM SOFTWARE

13814 Hawes St., Dept. M

Whittier, CA 90605

(714) 863-9044



EDITORIAL

(continued from page 2)

request we're also planning some introductory sessions on computers so everyone can gain something technical at SOG V.

Pascal Runoff

This issue is full of Pascal. We're announcing the Pascal Runoff winners and couldn't resist including a few appetizers. If you haven't tried Pascal yet, get a copy of Turbo and try it. It's neat. Really, it's the best language and compiler for just sitting and knocking out something quickly (see the C column for verification).

End Of The Minis?

Trevor Marshall called last night to let me know what was happening at Definicon Systems. He mentioned that mini folks are making a big mistake by ignoring what's going on in the micro world. (That's pretty small of them.)

It's not just a story of MIPSy or FLOPSy (Cottontail or Peter). Look at the text editors that run on the DEC - VI is a fur-piece from WordStar, Vedit, or even Perfect. And EX (a line editor) predated Beatrix Potter (assuming she dated at all).

I used both VI and EX during my stint at Tektronix because we had DEC's for software development and documentation. I remember how I enjoyed going home to my CP/M system. Everything worked so much faster (one of the advantages of a single user system) and the tools were a lot easier to use.

Of course, I couldn't compile really huge programs at home, or run gigantic applications, or share files with several users. But I didn't expect to. Only minis did all that.

Well, no longer, MIPSy. We've got micros that out-mini many minis (you heard it here first). If the mini users don't stop and look down pretty quickly they won't be able to. They'll be looking up. Micros are within a "hare's breath" of standing whisker to whisker with the fastest DEC's (8600 class).

There will always be a market for minis. After all, look at all the software. (It'll still be expensive and hard to use, but that hasn't been a problem so far.)

For the price of a dozen micros a year (and Peter's jacket) you get a service contract that'll guarantee your mini a dose of chamomile tea every time it goes picking lettuce in Mr. McGregor's garden.

How much time do the minis have left? Good question. The 32032 isn't exactly sitting on its hands right now. Trevor has 4 MBytes stuffed onto the board, and George is just finishing up the virtual memory code which effectively raises the total to 15 MBytes (the winchester looks like part of memory). Trevor says this scheme is easier and faster than using overlays.

Also, National has first silicon on the 32332. At 15 MHz it runs identical object code three times as fast as the 32032. National is also working on tying a new floating point processor to the 332. That processor will do a 64-bit

math operation in 250 ns (down from 8 us), or a transcendental (sine, cosine) in 2 us.

How Fast Is Fast Enough?

Every time I talk about fast, someone wonders how much faster all this fasting is at getting something done. Well, this power play doesn't necessarily do what you're currently doing any faster than you're currently doing it. (Now you see why faster isn't always better. Come back here, you blinkin' cursor!)

However, the power trip opens up a whole new arena to people doing mainframe work on a micro budget. But we're only plugging a mere \$995 or \$1495 into a PC clone. What does that buy us?

Very good question. So good, in fact, that we'll be doing a series of articles on how universities, communities, individuals, and corporations are using the DSI board to solve really substantial problems. In fact, considering DSI's order rate, there should also be a pretty good market for contract programmers who have the system.

Getting A DSI Kit

Most of the sales have been to institutions (colleges, towns, government agencies, funny farms...), so the orders have primarily been for full-blown assembled and tested units (\$3,000+ a pop).

DSI likes shipping A&T units because they can test them thoroughly beforehand, and they make more money on them. However, Trevor assured me that the kits are still available to hackers. To get a kit (\$995 for a 6 MHz 256K unit, \$1495 for a 10 MHz 1 Meg of RAM) you need to have a VISA or MC in hand when you call, and then insist that you want a kit. (Also, you must be purchasing it from inside the U.S.)

If you're using anything other than plastic (e.g. purchase order) then resign yourself to getting an assembled unit.

Trevor noted that support costs (two engineers on the phones plus parts, shipping, repairing, etc.) for the kits have been substantial, so they're really encouraging people (especially non-hardware types) to go for the finished systems.

I know very well the costs of supporting user manipulated hardware. It seems like every budding (and unbudding) hardware type has done the II to 4 upgrade or the Kaypro speed-up as his first project (get out the blowtorch, honey). A lot of innocent Kaypros have suffered needless brain damage.

Limited Pascal

I know why C is the language of choice for Unix systems. The Greenhills Pascal compiler that's available for the DSI-32 is absolutely standard Pascal, just like Unix Pascal. That means no string functions and no random

(continued next page)

Digital dynamics

- **NEW** - SEETIME has alarms! You can set up to 4 alarms, each containing date, time & message.
 - **NEW** - Screen saver feature turns off the screen after a set time with no keyboard activity.
 - **NEW** - Configuration program to set SEETIME parameters:
 - time/date location on screen
 - military or civilian display format
 - cursor blinking or steady
 - screen saver timeout
 - **SEETIME** can input the time or date into any program, print the screen and display alarms.
 - **SEETIME** is part of TIMEPRO:
 - APPT personal schedule manager
 - TD file time/date stamping & backup
 - LOG computer usage logging
- TIMEPRO runs on '84 & later Kaypros with Kaypro, Kenmore or Advent clocks.....\$74.95
SEETIME upgrade for TIMEPRO owners....\$14.95

Shipping and Handling - \$2.50

(206) 772-0291 Visa, MasterCharge
12448 83rd Ave S., Seattle WA 98178

Write-Hand Man

"Almost a Sidekick for CP/M"

Ted Silveira—Computer Currents, Aug. 27, 1985

"WHM is ingenious and works as intended"

Jerry Pournelle, BYTE Magazine, Sept. 1985 (c) McGraw-Hill

Now available for CP/M 2.2, CP/M 3.0 and ZRDOS!

The convenience of *Sidekick* on your CP/M machine! Trigger **Write-Hand-Man** with a single keystroke and a window pops open to run desk accessories. Exit **Write-Hand-Man** and both the screen and program are restored. Use with any CP/M program and most any CP/M machine. Takes only 5K of memory.

FEATURES

Notepad for quick notes	File and Directory viewer
Appointment calendar	Quick access phonebook
HEX calculator	14 digit decimal calculator

BONUS

Add applications written by you or others! No other *"Sidekick"* lets you add applications. Dump screens, setup printers, communicate with other computers, display the date and time. Let your imagination run wild!

\$49.95 (California residents add tax), shipping included. COD add \$2. Sorry, no credit cards or purchase orders. 30 day guarantee. Formats: 8 inch IBM, Northstar and most 5 inch (please specify).

Write-Hand-Man only works with CP/M 2.2, ZRDOS and CP/M 3.0 (please specify). Simple terminal configuration required. Not available for TurboDOS. Compatible with keyboard extenders, hard disks, and other accessories.

Poor Person Software

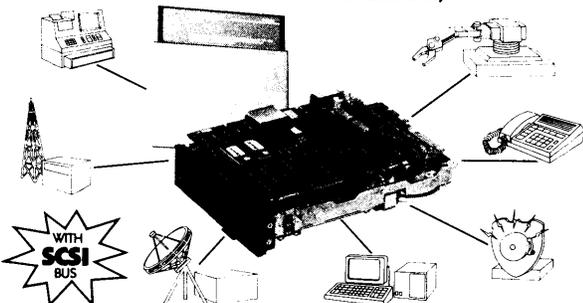
3721 Starr King Circle
Palo Alto, CA 94306
415-493-3735

Trademarks: **Write-Hand-Man** — **Poor Person Software**, CP/M—Digital Research, *Sidekick*—Borland International

Little Board™/186.... \$495

High Performance, Low Cost PC-DOS Engine

Boots IBM PC-DOS
(not included)

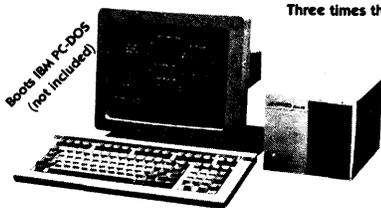


- Three times the COMPUTING POWER of a PC
- Data and File Compatible with IBM PC, runs "MS-DOS generic" programs
- 8 MHz 80186 CPU, DMA, Counter/Timers, 128/512K RAM zero wait states, 16-128K EPROM
- Mini/Micro Floppy Controller (1-4 Drives, Single/Double Density, 1-2 sided, 40/80 track)
- 2 RS232C Serial Ports (50-38,400 baud), 1 Centronics Printer Port
- Only 5.75 x 7.75 inches, mounts directly to a 5-1/4" disk drive
- Power Requirement: +5VDC at 1.25A; +12VDC at .05A; On board -12V converter
- SCSI/PLUS™ multi-master I/O expansion bus
- Software Included:
 - PC-DOS compatible ROM-BIOS boots DOS 2.x and 3.x
 - Hard Disk support
- OPTIONS:
 - Expansion board with:
 - 128 or 512K additional RAM
 - 2 Sync/Async RS232/422 serial ports
 - Battery backed Real Time Clock
 - 8087 Math Co-Processor
 - Buffered I/O Bus
 - STD Bus Adapter
 - Utilities source code
 - TurboDOS / Networking

BOOKSHELF™ Series 200

Fast, compact, high quality, versatile PC-DOS system

Three times the COMPUTING POWER of a PC



Priced from
\$1295.00
10MB System
Only **\$1945.00**

- Data and File compatible with IBM PC-DOS 2.x and 3.x
- Runs "MS-DOS generic" programs (Dbase II, Multiplan, Wordstar, Supercalc 2, Turbo Pascal, Fortran 77, Microsoft C, Lattice C, IBM Macro Assembler, Intel compilers & tools, GW Basic, etc.....)
- Works with any RS232C ASCII terminal (not included)
- Compact 7.3 x 6.5 x 10.5 inches, 12.5 pounds, all metal construction
- Based on Little Board/186
 - 512K RAM, no wait states
 - Two RS232 serial ports
 - One Centronics printer port
- One or two 360 Kb floppy drives
- 10MB internal hard disk drive option
- Software Included:
 - PC-DOS Compatible ROM-BIOS boots DOS 2.x and 3.x
 - Hard Disk Support
 - T/Maker III — Word processing, spreadsheet, relational database, spelling checker, and data encrypt/decrypt
- Expandable:
 - Floppy expansion to four drives
 - Hard disk and tape expansion
 - SCSI/PLUS™ multi-master I/O expansion bus

DISTRIBUTORS

ARGENTINA: FACTORIAL, S.A., (1) 41-0018, TLX 22408
BELGIUM: CENTRE ELECTRONIQUE LEMPEREUR, (041) 23-4541, TLX 42621
CANADA: DYNACOMP COMPUTER SYSTEMS LTD., (604) 872-7737
ENGLAND: QUANT SYSTEMS, (01) 253-8423, TLX 946240 REF:19003131
FRANCE: EGAL+, (1) 502-1800, TLX 620893
SPAIN: XENIOS INFORMATICA, 593-0822, TLX 50364
AUSTRALIA: ASP

MICROCOMPUTERS, (613) 500-0628
BRAZIL: CNC-DATA LEADER LTDA., (41) 262-9262, TLX 041-6364
DENMARK: DANBIT, (03) 66-90-90, TLX 43558
FINLAND: SYMMETRIC OY, (0) 585-322, TLX 121394
ISRAEL: ALPHA TERMINALS, LTD., (3) 49-16-95, TLX 341667
SWEDEN: AB AKTA, (08) 54-20-20, TLX 13702
USA: CONTACT AMPRO COMPUTERS INC., TEL: (415) 962-0230 TELEX: 4940302

AMPRO
COMPUTERS, INCORPORATED

67 East Evelyn Ave. • Mountain View, CA 94041 • (415) 962-0230 • TELEX 4940302

EDITORIAL

(continued from page 87)

access files (to name a few). (Not Wirthwhile.)

When I first tried doing things in Greenhills Pascal I felt like I was learning the language all over again (Turbo has really spoiled me). Fortunately, you can use the C I/O library that's included with the Pascal compiler to get random file access, and it's possible to write your own string functions using arrays of characters.

In fact, I understand that some folks are already writing fancy libraries for the compiler, so most problems with this implementation should go away soon. I'll keep you posted.

White Out

I'm working at home today because it would be very difficult to get to the office. It wouldn't be impossible — I could strap on the skis and get there in half an hour. But the staff thinks I'm stranded, so what the heck. (What a snow job.)

This is the most snow I've ever seen in Bend, and it's beautiful. There's no wind, so it forms powdery frosting peaks on branches and fences and vehicles (I know they're under there somewhere!).

Officially it's two feet deep (and still snowing), but I prefer to measure alongside the walk I've shoveled. There it's three feet. Easily. Those of you who see Bend during the SOG see only a small part of the beauty of this area. (On second thought, what am I doing in here? Where'd I put the wax?)

X16

Every once in a while we get word back that our readers have gone bonkers over something we've covered. The latest example of such response is the X16. We'd been hoping to try out one for ourselves, but after mentioning it at SOG IV we heard very little from Dean and Earl (they designed the hardware and software for the Slicer and then split off to do the X16).

It turns out that they were inundated by orders. It seems that everyone wants a very compatible system that's priced like a Kaypro but outruns an AT.

Anyway, I'm getting more and more interested in the system, but I know that Dean and Earl have been reduced to sharing a cobbled-together prototype. (It always happens to a small operation when there's a flood of anxious customers bearing cash.)

Next time I get really excited about something I'll try to be a little less effusive, at least until I've got my hands on one.

8086 Benefits

I almost didn't go back to Comdex this year. After all, last year was pretty much a waste of time. Only half-a-dozen booths had anything at all for CP/M, and very little of that was new.

This year was different. Oh, CP/M had disappeared entirely as far as anyone could see (which is a long, long way in the Nevada desert), but this year I was looking

for bargains in new 8088 software to run on our new clones.

See the Comdex article in this issue if you're at all interested in hearing about the gems I found in the drivel.

By the way, Sandy and I attended one party while we were there, given by the Canadian Export group. There was no buy-Canadian hype, crushing handshakes, or speeches from the Main Mounty (or whatever they call their top man).

I like that. In fact, I hope they manage to make something out of Canada. (Maybe I should leave my skis on and go see how they are doing.)

Stamping Out Letters

You should see the mail that wanders in here. Letters, boxes, large envelopes full of fancy printing and appeals for editorial space. The contents of our daily mail are fun but the stamps that bring in all that mail can be truly beautiful. (Sandy has quite a collection.) Thanks.

Issue 27

If you thought that issue #27 was too pretty, too prim and proper, relax. We won't let it happen again. The printer accidentally used the wrong paper (an extra \$1800 worth). The magazines also weighed 1 1/2 ounces more than before, which cost us almost \$2,000 in additional postage. Fortunately, the printer ate the paper (no mean feat) and also reimbursed us for the additional postage.

By the way, we have over 10,000 paid subscribers and sold about 1,500 copies of #27 on newsstands.

Amiga Report

We get a lot of calls at the office asking how we like the Commodore Amiga.

"Fine, we think, but we're not sure it's real."

But they're advertising it on TV; beautiful full color shots of an impressive looking figure walking towards an altar. On the altar is an Amiga! (I mean, who could ask for a better guarantee than that?)

Anyway, if it's on television then it must be real. Right? Well, they're selling all right, but I'm still not sure what they're really selling.

The Missing Company

Commodore had reserved (and I understand, paid for) a booth at Comdex. However, they weren't there. In fact, there were many no-shows at Comdex, but that's another story.

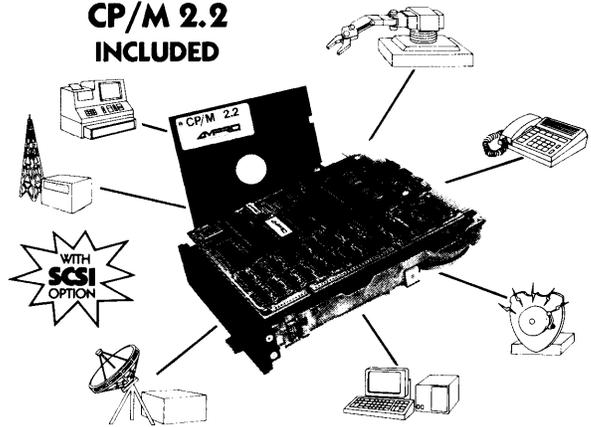
Commodore reportedly spread the rumor that they weren't there because they didn't want any more dealers. That, of course, doesn't wash. There can't be too much software available for the Amiga, and Comdex is an excellent place to meet software writers, to put together cooperative packaging deals, and so forth. If anything,

(continued next page)

Little Board™ \$249

The World's Least Expensive CP/M Engine

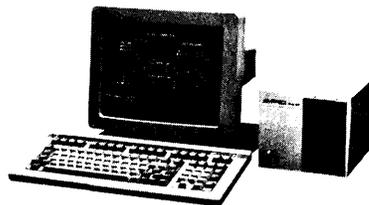
CP/M 2.2 INCLUDED



- 4 MHz Z80A CPU, 64K RAM, Z80A CTC, 4-32K EPROM
- Mini/Micro Floppy Controller (1-4 Drives, Single/Double Density, 1-2 sided 40/80 track)
- 2 RS232C Serial Ports (75-9600 baud & 75-38, 400 baud), 1 Centronics Printer Port
- Power Requirement: +5VDC @ .75A; +12VDC @ .05A / On board -12V converter
- Only 5.75 x 7.75 inches, mounts directly to a 5-1/4" disk drive
- Comprehensive Software Included:
 - Enhanced CP/M 2.2 operating system with ZCPR3
 - Read/write/format dozens of floppy formats (IBM PC-DOS, KAYPRO, OSBORNE, MORROW...)
 - Menu-based system customization
 - Operator-friendly MENU shell
- OPTIONS:
 - Source Code
 - TurboDOS
 - ZRDOS
 - Hard disk expansion to 60 megabytes
 - SCSI/PLUS™ multi-master I/O expansion bus
 - Local Area Network
 - STD Bus Adapter

BOOKSHELF™ Series 100

Fast, Compact, High Quality, Easy-to-use CP/M System



Priced from
\$895.00
10MB System
Only \$1645.00

- Ready-to-use professional CP/M computer system
- Works with any RS232C ASCII terminal (not included)
- Network available
- Compact 7.3 x 6.5 x 10.5 inches, 12.5 pounds, all-metal construction
- Powerful and Versatile:
 - Based on Little Board single-board computer
 - One or two 400 or 800 KB floppy drives
 - 10-MB internal hard disk drive option
- Comprehensive Software Included:
 - Enhanced CP/M operating system with ZCPR3
 - Word processing, spreadsheet, relational database, spelling checker, and data encrypt/decrypt (T/MAKER III™)
 - Operator-friendly shells; Menu, Friendly™
 - Read/write and format dozens of floppy formats (IBM PC-DOS, KAYPRO, OSBORNE, MORROW...)
 - Menu-based system customization

DISTRIBUTORS

ARGENTINA: FACTORIAL, S.A., (1) 41-0018, TLX 22408 **BELGIUM:** CENTRE ELECTRONIQUE LEMPEREUR, (041) 23-4541, TLX 42621 **CANADA:** DYNACOMP COMPUTER SYSTEMS LTD., (604) 872-7737 **ENGLAND:** QUANT SYSTEMS, (01) 253-8493, TLX 946240 REF:19003131 **FRANCE:** EGAL+, (1) 502-1800, TLX 620893 **SPAIN:** XENIOS INFORMATICA, 593-0822, TLX 50364 **AUSTRALIA:** ASP MICROCOMPUTERS, (613) 500-0628 **BRAZIL:** CNC-DATA LEADER LTDA., (41) 262-2262, TLX 041-6364 **DENMARK:** DANBIT, (03) 66-20-20, TLX 43558 **FINLAND:** SYMMETRIC OY, (0) 585-392, TLX 121394 **ISRAEL:** ALPHA TERMINALS, LTD., (3) 49-16-95, TLX 341667 **SWEDEN:** AB AKTA, (08) 54-20-90, TLX 13702 **USA:** CONTACT AMPRO COMPUTERS INC., TEL: (415) 962-0230 TELEX: 4940302

IBM®, IBM Corp.; Z80A®, Zilog, Inc.; CP/M®, Digital Research; ZCPR3™ & ZRDOS™, Echelon, Inc.; Turbo DOS®, Software 2000, Inc.; T/MAKER III™, T/Maker Co.

AMPRO
COMPUTERS INCORPORATED

67 East Evelyn Ave. • Mountain View, CA 94041 • (415) 962-0230 • TELEX 4940302

until they get their operating system a bit settled I don't think they'll be faced with too much of anything (except maybe questions about ads and vaporware and bugs and...).

They're selling systems, however, and we're all certainly interested. I'd wait six months before I bought one, though.

At about \$2000 a pop with two drives, 512K, and a color monitor, Amigas are probably too expensive to make much of a dent in the home user market. But a Commodore rep told me that they aren't expecting to do well in the business market (that belongs to IBM and company). That just leaves the Mac market, which Apple is trying to bail out of.

If I were writing software for the Amiga, I'd write tools for those writing games, and games for the home users. I might even write some graphics or music software, but beyond that, I don't know. Business software may not make sense.

Private Domain

Wow! We've received a dozen submissions (underwater maneuvers) for Private Domain, though issue #27 is just beginning to show up in your mailboxes.

This is really exciting. Larry has been dashing about the office shouting, "Look at this, look at this." Then Bruce has to go look, which piques Gary's curiosity, and then Dave Pogue gets drawn in (he loves crowds). Finally, of course, I can't stand it any longer so I wander over. (By the way, there's some really great stuff here.) All we have to do is check out the code, the documentation, the user interface — by SOG, guaranteed.

Meanwhile, keep those cards and letters and ideas and programs coming. Being in the middle of this (even if I'm the last man in) is the best antidote to burnout I know of. (But then, who's going to burn anything with all that snow out there.)

We've received a number of letters suggesting different ways of figuring the royalties and pricing the products. Some really good ideas (see the Letters for one suggestion) have already shown up but if you have an idea or opinion, we're still interested.

Dealers Exposed (Streets Safer)

The "Turn In A Dealer Department" here at Micro C has already received many hot tips on dealers who should have Micro C on their newsstands and book racks.

If you want to participate in this motherhood and apple pie campaign, please jot down names and addresses of aforementioned establishments and drop them to:

Turn In A Dealer Department
Micro Cornucopia
PO Box 223
Bend OR 97709

We send the dealers information about selling Micro C, plus you receive a certificate good for one disk — any public domain disk — for each of your dealers that decides to stock us.

Micro C Gets Second Class Rating!

Only 10 months after we made application to mail at 2nd class rates we got confirmation from the post office that we are indeed a magazine (and that we can mail 2nd class).

This is supposed to mean that it'll take a week for you East Coasters to receive your magazines (rather than 3 or 4 weeks). It also means that our postage rates drop a bit. (I wonder if life would be cheaper if I became a second class citizen.) Anyway, thanks USPS. I take back half of those unprintable things I thought about you.

C Books

After years of almost no C books at all, now it seems that the C section of the book store (where else?) has more selection than the Apple box.

I don't know if the plethora is a result of C's popularity, its difficulty, or simply that publishers don't have anything better to do. Anyway, after purchasing nearly \$100 worth of books, I've found a real gem. It's called:

C Primer Plus

By Waite, Prata, Martin

Published by Howard W Sams

ISBN 0-672-22090-3

\$21.95 — 531 pages

If you want to move swiftly into the heavy-duty stuff, this may not be your book. If you want to gently and painlessly ease your way in with a book that's just as comfortable on the nightstand (it's fun to read) as next to the computer, then this is for you. Kudos for the group.

The April issue will be a sort of C spectacular (it's not cast in C-ment yet, but it looks pretty C-cure). We've got something exciting to announce, so dust off Kernighan and Ritchie or prop open a copy of C Primer Plus, and you'll be ready when we get there.

Finally

Enjoy Yourself. (If you don't, who will?)

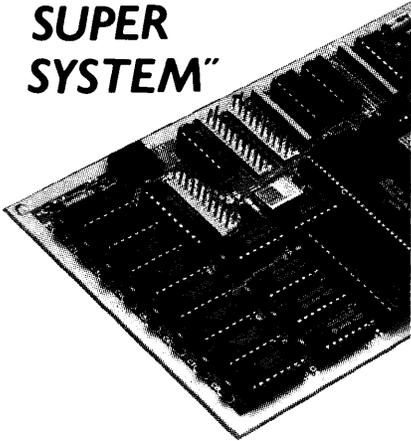


David Thompson
Editor & Publisher



Byte Magazine called it. ■

"CIARCIA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985, the SB180 lets CP/M users upgrade to a fast, 4" x 7 1/2" single board system.

- **6MHz 64180 CPU**
(Z80 instruction superset), 256K RAM, 8K Monitor ROM with device test, disk format, read/write.
- **Mini/Micro Floppy Controller**
(1-4 drives, Single/Double Density, 1-2 sided, 40/77/80 track 3 1/2", 5 1/4" and 8" drives).
- **Measures 4" x 7 1/2"**, with mounting holes
- **One Centronics Printer Port**
- **Two RS232C Serial Ports**
(75-19,200 baud with console port auto-baud rate select).
- **Power Supply Requirements**
+5V +/-5% @500 mA
+12V +/- 20% @40mA
- **ZCPR3 (CP/M 2.2/3 compatible)**
- **Multiple disk formats supported**
- **Menu-based system customization**

SB180-1
SB180 computer board w/256K bytes RAM and ROM monitor **\$369.00**

SB180-1-20
same as above w/ZCPR3, ZRDOS and BIOS source..... **\$499.00**

•Quantity discounts available.

NEW

COMM180-M-S
optional peripheral board adds 1200 bps modem and SCSI hard disk interface.

TO ORDER CALL TOLL FREE 1-800-635-3355 **TELEX 643331**

For technical assistance or to request a data sheet, call:
1-203-871-6170



Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066

THE CULTURE

C
O
R
N
E
R

Of Magazines And Modems

By Julianna Snow

Berliner Ring 6/11
8047 Graz Austria

He gets a lot of mail nowadays. Byte, Profiles, Micro Cornucopia, Personal Computing, and newsletters from organizations with unpronounceable abbreviations like NWKUG and IBMPC.

I wouldn't complain, except that he reads it during supper. I cut up the children's meat to the sound of tearing envelopes. Pages turn and forks clink in counterpoint. The steam rising from his mashed potatoes crinkles the last sheet of a newsletter.

The kids and I discuss that day's events at school in hushed tones. Who knows when their father might have an important bit of computer information to share with us. Suddenly he laughs out loud. (Why are the writers of computer magazines so funny? Sometimes I can't help but picture them with green, no-glare, monitor-shaped glasses.)

"Listen to this," he chuckles. "What a turn-out for the last meeting! Sixty-three computers brought their owners."

I smile. Our youngest drops a spoonful of peas into his lap.

He turns a page. "Hey!" he exclaims happily. "I can turn the computer on without touching the on/off switch. You just type TURNON."

"I don't get it," says our eldest. "How can you type on it when it's off?"

He doesn't answer. The potatoes have stopped steaming. The peas are starting to shrivel.

"If I had a modem, then I could leave messages at a computerized bulletin board, and if John had a modem, then we could both leave messages and talk to each other by computer."

I wonder if it's cheaper than a long-distance phone call, but I don't dare ask. Explanations have a tendency to cause arguments, or at best become boring.

He continues reading. "Modems come in many different flavors..."

"Chocolate and vanilla?" asks our daughter.

He peers at her over the newsletter and then patiently explains that a computer can change telephone sound into visual symbols and back again. That's a modem.

He returns to his reading. "There are two different types of couplers. Acoustic couplers transmit and receive data at a rate of 300 bauds."

"Three hundred bods?" I ask, astonished.

He glances up quickly. "That's *bauds*— bits per second."

I nod, knowing that my face registers about as much understanding as an unplugged monitor. The children ask to be excused from the table.

"Twelve hundred baud couplers are more expensive," he muses.

I shrug. "Why scrimp?"

He sighs and notices his cold food, our children's empty plates and chairs. He begins to eat. I guess I should be grateful he never asks me to reheat it in a warm boot.



MAKE YOUR KAYPRO SPECIAL!

The Kaypro Disk Library

37 Disks of Selected Public Domain Software

KAYPRO DISK K1
Modem software
KAYPRO DISK K2
Utilities
KAYPRO DISK K3
Games
KAYPRO DISK K4
Adventure
KAYPRO DISK K5
MX-80 Graphics
KAYPRO DISK K6
Word Processing Utilities
KAYPRO DISK K7
Small C Version 2 Compiler
KAYPRO DISK K8
Small C Version 2 Source
KAYPRO DISK K9
ZCPR
KAYPRO DISK K10
Assemblers
KAYPRO DISK K11
Library & Checkbook Programs
KAYPRO DISK K12
FORTH
KAYPRO DISK K13
Source of fig-FORTH
KAYPRO DISK K14
Smartmodem Program
KAYPRO DISK K15
Hard Disk Utilities
KAYPRO DISK K16
Pascal Compiler
KAYPRO DISK K17
Z80 Tools
KAYPRO DISK K18
System Diagnosis
KAYPRO DISK K19
Prowriter Graphics
KAYPRO DISK K20
Color Graphics Routines
KAYPRO DISK K21
SBASIC Routines & Screen Dump
KAYPRO DISK K22
ZCPR (Again)
KAYPRO DISK K23
Fast Terminal Software & New BYE
KAYPRO DISK K24
MBASIC Games & Keyboard Translator
KAYPRO DISK K25
Z80 Macro Assembler
KAYPRO DISK K26
EPROM Programmer & Character Editor
KAYPRO DISK K27
Typing Tutor
KAYPRO DISK K28
Modem 730
KAYPRO DISK K29
Turbo Pascal Games 1 With Source
KAYPRO DISK K30
Turbo Pascal Games II With Source
KAYPRO DISK K31
Turbo Bulletin Board
KAYPRO DISK K32
Forth-83 Much Fancier Forth
KAYPRO DISK K33
A super utilities disk
KAYPRO DISK K34
Five games plus source (mostly Turbo)
KAYPRO DISK K35
Small C Compiler & Source - Vers 2.1
KAYPRO DISK K36
Small C Library of 100 functions

Disks \$12.00 each

ROMS

Pro-8 Version 3 for Kaypro 4-83

Features include:

1. Screen dump with selectable dump character.
2. Select slow or fast step rate for each drive.
3. Automatically figures out what type of drive you are using.

Plus: User selectable cursor (blinking or not), ignores nulls, and your choice of 1-4 drives of the 191K, 390K, and 784K variety. (Use of 3 or 4 drives requires decoder.)

Installation requires no cuts or jumpers. The ROM simply plugs into a Kaypro 4-83 (or II-83 with a Kaypro 4 processor board). If you have a Kaypro II with the original II processor board then you must do the II to 4 upgrade. See issue #21 for details. Your Kaypro II has the original II board if the monitor ROM (a 20-pin chip with paper stuck to its top) is marked 81-149. The 4 ROM is marked 81-232.



for 84 KayPros

Pro-884

This ROM lets you run any mix of quad density (784K) drives and double-sided double density drives as A: and B: on any 84 Kaypro. Plus, if you plug-in the decoder board, you can run up to four drives.

Pro-884 MAX

The MAX gives you all the advantages of the Pro-884, plus faster screen scrolling, ZCPR in ROM (you 'warm boot' on any disk, whether it has system tracks or not), screen dump, ctrl-p type printer output, and VT52H/19 terminal emulation in addition to the original ADM3, to name a few. (We worked on this hummer for nearly a year.) If you need more information on the Pro-884s or our '83 compatible ROMS, call, write, or yell for our KayPro Users Catalog

Pro-Monitor ROMS

Pro-8 Ver 3 \$49.95
Pro-884 \$59.95
Pro-884 MAX \$79.95
Decoder Board \$39.95

Kaypro Add-Ons

Eight Inch Adaptor Board

OK, you asked for it: the 8 inch adapter board for the Kaypro. Now you can have: 1. 4 drives 2. Up to 1.3 meg of storage per disk. 3. Access to SIGM and CPMug software. 4. Super fast data access. The circuit board, ROM, and disk in this package will let your Kaypro run up to four drives. A and B have to be 5" drives, C and D can be either 8" or 5". The eight inch drives can read and write single density, double density, or double-sided double density. Plus, if you have an 84 2 or 4 you get all the Pro-884 MAX features.

The 8" adapter board plugs right in (no cuts, jumpers, or soldering — unless you have a Kaypro II, then you have to do a II to 4 upgrade.) You supply the drives, cabinet, and power.

8" Adaptor Board \$190.00

Schematic Packages

Finally, a schematic of your processor board, logically laid out on a single 24" by 36" sheet, plus a very complete, illustrated, Theory of Operation that's keyed to the schematic. You'll get information that's available nowhere else.

KayPro Schematic Packages

Kaypro II & 4 (pre-84) \$20
Kaypro 10 (pre-84) \$20
Kaypro 84 series (II, 4 & 10) \$20

SPECIAL PRO-884 NOTE:

(And for 8" Adaptor Board)

The Pro-884s are sensitive to the version of CP/M you are running.

1. Neither the Pro-884 nor the Pro-884 Max will run on CP/M 2.2U. However, if you can locate a CP/M 2.2F or 2.2G system disk (your dealer should have a copy) you should be able to run our 884 monitors. (Don't try to boot F or G before you change monitors.)

2. There are two distinct versions of CP/M 2.2G. Only the Pro-884 Max is sensitive to the version of 2.2G you have - it's the ZCPR in ROM that's the problem. (If you have CP/M 2.2F then you have a Normal CP/M.) So, before ordering the Max, boot up your original system disk and read the sign-on. If it's CP/M 2.2G then we need to know whether it is the high (normal) version or the low (minus) version.

To determine your G version (you'll become a G Whiz!):

```
A>DDT <cr>
L5 <cr>
(ddt's response)
```

The first line of the response will be a JMP D600 or a JMP D800. The JMP D600 means that you have a low (minus) version, and the JMP D800 means that it's a normal version. When you order your Pro-884 Max, be sure to specify whether you want the normal Max or the minus Max. Otherwise, we'll just guess that you need the normal Max.

MICRO CORNUCOPIA

P.O. Box 223 • Bend, Oregon 97709

Order Number 9-5 PST (503) 382-5060 • Technical Information 9-Noon PST Only (503) 382-8048

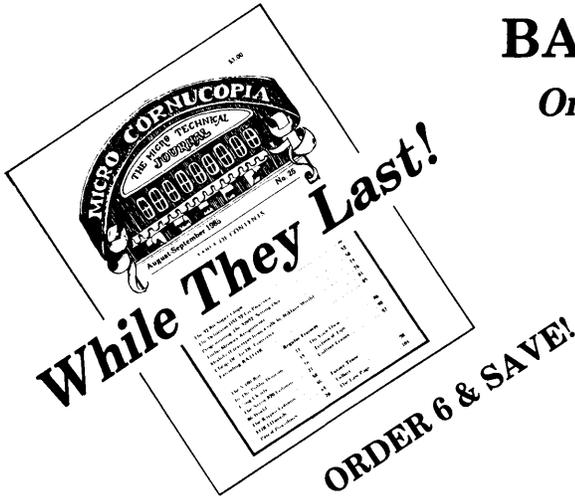
WE'RE CLEARING THEM OUT!

BACK ISSUES OF MICRO C

Only \$1.50 each when you order 6 or more

All Foreign \$2.00 each for 6 or more

(includes surface postage)



ISSUE NO. 1 (8/81)

Power Supply
RAM Protection
Video Wiggle
1/2 PFM, PRN
16 pages

ISSUE NO. 2 (10/81)

Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
1/2 PFM, PRN
16 pages

ISSUE NO. 3 (12/81)

4MHz Mods
Configuring Modem 7
Safer Formatter
Reverse Video Cursor
FORTHwords Begins
16 pages

ISSUE NO. 4 (2/82)

Keyboard Translation
More 4MHz Mods
Modems, Lync, and SIOS
Undoing CP/M ERASE
Keyboard Encoder
20 pages

ISSUE NO. 5 (4/82)

Word Processing
Two Great Spells
Two Text Editors
Double Density Review
Scrubble, A Formatter
20 pages

ISSUE NO. 6 (6/82)

BBI EPROM Programmer
Customize Your Chars
Double Density Update
Self-Loading ROM
Terminal In FORTH
24 pages

ISSUE NO. 7 (8/82)

6 Reviews Of C
Adding 6K of RAM
Viewing 50 Hz
On Your Own Begins
24 pages

ISSUE NO. 8 (10/82)

Drive Maintenance
Interfacing Drives
Installing A New BIOS
Flippy Floppies
C'ing Clearly Begins
Xerox 820 Begins
28 pages

ISSUE NO. 9 (12/82)

BBII EPROM Program
Relocating Your CP/M
Serial Print Driver
Big Board I Fixes
Bringing Up WordStar
Cheap RAM Disk
32 pages

ISSUE NO. 10 (2/83)

Saving A Flakey Disk
Hooking Win'g BBII
The Disk Color
JRT Fix
Serial Board Interface
Procedures Begins
36 pages

ISSUE NO. 11 (4/83)

BBI Expansion
BBII Details
Dyna, RAM Disk Review
Easier Base Video Cursor
Planetic Review
Keyboard Column Begins
36 pages

ISSUE NO. 12 (6/83)

256K For BBI
Bringing Up BBII
dBASE II
Look At WordStar
Double Sided Drives For BBI
Packet Radio
5MHz For Kaypro
40 pages

ISSUE NO. 13 (8/83)

CP/M Disk Directory
More 256K For BBI
Mini Front Panel
Cheap Fast Modem
Nevada COBOL Review
BBI Printer Interface
Kaypro Reverse Video Mod
44 pages

ISSUE NO. 14 (10/83)

BBII Installation
The Perfect Terminal
Interface To Electronic Typewriter
BBI Video Size
Video Jitter Fix
Slicer Column Begins
Kaypro Color Graphics Review
48 pages

ISSUE NO. 15 (12/83)

Screen Dump Listing
Fixing Serial Ports
Playing Adventure
SBASIC Columns Begins
Upgrading Kaypro II To 4
Upgrading Kaypro 4 To 8
48 pages

ISSUE NO. 16 (12/84)

Xerox 820 Column Restarts
BBI Double Density
BBII 5'78" Interface Fix
Kaypro ZCPR Patch
Adding Joystick To Color Graphics
Recovering Text From Memory
52 pages

ISSUE NO. 17 (4/84)

Voice Synthesizer
820 RAM Disk
Kaypro Morse Code Interface
68000-Based System Review
Inside CP/M 86
56 pages

ISSUE NO. 18 (6/84)

Kaypro EPROM Programmer
I/O Byte: A Primer
Kaypro Joystick
Serial To Parallel Interface
Business COBOL
60 pages

ISSUE NO. 19 (8/84)

Adding Winchester To BBII
6MHz On The BBI
Bulletin Boards
Track Buffering On Slicer
4MHz For The 820-I
64 pages

ISSUE NO. 20 (10/84)

HSC 68000 Co-Processor
DynaDisk For The BBII
Serial Printer On BBI Sans SIO
Cheap & Dirty Talker For Kaypro
Extended 8" Single Density
72 pages

ISSUE NO. 21 (12/84)

Analog To Digital Interface
Installing Turbo Pascal
Low Intensity BBI Video
Turbo Pascal, The Early Days
80 pages

ISSUE NO. 22 (2/85)

Xerox 820-II To A Kaypro-8
Sound Generator For The STD Bus
Reviews Of 256K RAM Expansion
In The Public Domain Begins
88 pages

ISSUE NO. 23 (4/85)

Automatic Disk Relogging
Interrupt Driven Serial Printer
Low Cost EPROM Eraser
Smart Video Controller
Review: MicroSphere RAM Disk
Future Tense Begins
88 pages

ISSUE NO. 24 (6/85)

C'ing Into Turbo Pascal
8" Drives On The Kaypro
48 Lines On A BBI
68000 Vs. 80X86
Soldering: The First Steps
88 pages

ISSUE NO. 25 (8/85)

Why I Wrote A Debugger
The 32-Bit Super Chips
Programming The 32032
Modula II
RS-232C: The Interface
104 pages

ISSUE NO. 26 (10/85)

Inside ZCPR3
Two Megabytes On DSI-32
SOG IV
The Future Of Computing
MS-DOS In The Public Domain
Graphics In Turbo Pascal
104 pages

BACK ISSUES

U.S. Regular Price \$3.00 each
6 or more 1/2 price \$1.50 each

Regular Foreign Price (air mail) \$5.00 each
6 or more (including Canada) \$2.00 each

Surface Postage Included

If possible, please indicate alternate issue in case we are out of your choice.

MICRO CORNUCOPIA

P.O. Box 223 • Bend, Oregon 97709

Order Number 9-5 PST (503) 382-5060 • Technical Information 9-Noon PST Only (503) 382-8048



Amiga Hot Lines

Amiga folk, the North American Amiga Users Group wants you to get excited about its favorite computer.

For a free copy of the NAAUG newsletter (their first) write —

North American Amiga Users Group
Box 376
Lemont PA 16851

Also, an Amiga news network is currently active on the ARPA net.

A few Software designers (at least) like what they've seen and heard the Amiga do, despite AmigaDOS crashes. They're confident.

Cube Root Retakes

Many of you have offered simpler solutions to Cheung Kwan's Cube Root Function (Future Tense, issue 27).

The simplest uses Turbo Pascal's built-in LN and EXP functions to evaluate a Number (Num) to a power (Expnt) —

```
root := Exp(Ln(Num) / Expnt);  
power := Exp(Ln(Num) * Expnt);
```

So, a FUNCTION cbrt can be written —

```
FUNCTION cbrt (Num: real) : real;  
BEGIN  
  IF x = 0  
  THEN cbrt := 0  
  ELSE cbrt := exp(ln(Num) / 3);  
END;
```

Thanks bunches to Richard Barnhart, Mike Bordelon, David Pickett, Eric Torney, Charles Kusmirek, Jim Kirchner, Jeff Wells, and Mark Bohn for their contributions.

Hard Disk For Z80s

If you're aching to control a hard disk with your Kaypro or Big Board, Advanced Concepts in Boca Raton, Florida has a reasonably priced solution. \$395 gets you a 5 megabyte full-height drive, MWC controller, software, documents, two 1.5' drive cables, and support in a neatly packaged, speedy bundle. (It's a steal.)

The MWC controller works with 5,

10, and 20 meg internal and external drives.

I asked Grant Newland (Advanced Concepts designer) how they designed the controller.

"Normally, the WD1010 is mated to a number of WD specialty LSIs and an extraordinary number of other parts. It's a sophisticated design, but it's usually underutilized.

"Passing data to and from the drive is simpler if you know which functions won't be needed — as in a CP/M BIOS application, for example.

"We used a relatively new (and expensive) National Semiconductor hybrid part for data separator. National didn't design the part for use with a 1010, but it works if it's told to ignore certain things in the data stream. The 1010 handles those.

"Finally, we spent a lot of attention optimizing format interleave for each host computer."

Thanks, Grant.

For more info —

Advanced Concepts
8926 SW 17th St
Boca Raton FL 33433
305-482-7302

TP Note

In CP/M 2.x (but not in CP/M 3.0), Turbo Pascal will always change the user area to zero on exit. The following patch maintains the current user area —

Change instruction at 2A44:

```
From LDA 43DB (3A DB 43)  
To   JMP 0
```

83 or 84? (The Fogg Solution)

Since port 1EH doesn't exist on the 84 Kaypros you'd think writing to it would have no effect. Not so. Rather odd graphic characters appear on the screen in most cases.

So the idea is to see if port 1EH is out there. If it is, you've got an 83 non-graphic Kaypro.

The solution (using "OriginalVal - 2" which worked well on our assortment of mongrels) follows —

```
FUNCTION TypeKaypro : Boolean;  
  {tests for 83/84 Kaypro}  
VAR  
  OriginalVal,  
  NewVal,  
  NewValStored : Integer;  
BEGIN  
  OriginalVal := Port [30];  
  {test for existence of port 1E}  
  NewVal := OriginalVal - 2;  
  Port [30] := NewVal;  
  {store new value in port 1E}  
  NewValStored := Port [30];  
  {read value stored in port}  
  IF NewVal <> NewValStored  
  {was value actually stored?}  
  THEN TypeKaypro := true  
  {it's an 84}  
  ELSE TypeKaypro := false;  
  {it's an 83}  
  Port [30] := OriginalVal  
  {restore original value}  
END; {TypeKaypro}
```

Gene Wolfe

And now for something really entertaining — an excerpt from "Alien Stones" by Gene Wolfe.

"You asked him to find out —." She hesitated.

"I asked him to find out what the numbers in the operating registers of the ship's computer were. To put it another way, I asked him to find out the answer — in raw form at least — of the last computation they performed."

"Is that possible? I would think their numbers would be all different — like Roman numbers or something, or worse ... whatever he found would just be ones and zeros —"

"Binary notation."

"Yes, because it isn't really numbers; you can't have real numbers inside a machine because they're not physical, but just things turned on and off; but I don't see what good knowing it — just one, one, zero, zero, zero, like that — will do you if you don't know how they'd be used in another machine.

"Captain, I know you must think I don't know what I'm talking about, but I did have to take some mathematics ... even if I wasn't very good at it."

The translation ended in a whisper of despair.

The stories in "The Island Of Doctor Death And Other Stories And Other Stories" are strangely good.

And if you really need to be taken away by a book, the less-morbid-than-it-sounds "The Shadow Of The Torturer" by Nebula Award winner Gene Wolfe will suck you into four volumes of hypnotic mystery.

Build A Cheap PC (The Saga Continues)

All month I've heard rumors (and rumors of rumors) of ultra cheap XT-compatibles and just cheap AT-compatible system boards. By now everyone in Taiwan must be earning a living copying IBM (or spreading rumors).

Generally, prices on XT and AT-compatible main boards are dropping, but aren't near "rumor cheap" levels yet.

Best buy on an XT board without memory is still about \$165 (if you're buying just one) and \$240 or so with 640K. An AT board (80286 CPU) costs

\$900 or so with 640K of memory. (Buy your memory quickly — it's going up in price.)

You can still build a good personal clone for less than \$800, but an AT clone's going to cost you another \$700 or so.

The prices of drives, monitors, printers, and various plug-in cards are no longer dropping, and good drives (Mitsubishis and TEACs) are much harder to find than they were a month ago. Who's got the parts? We'll let you know as we find out. You can buy TEACs for around \$100 each (up from \$79 3 weeks ago) and 20 meg Seagate half-heights with PC controller for \$499.

Drives from —
 OWL-Services
 P.O. Box 116
 Mertztown PA 19539
 215-682-6855

Seagates from —
 Supreme USA, Inc.
 6481 Orangethorpe Ave #9
 Buena Park CA 90620
 714-739-5250

AT components from —
 MicroPeach Computers
 4768 Lincoln Ave
 Cypress CA 90630
 714-995-3600

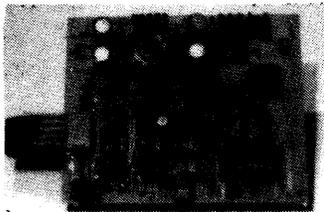
PC components from —
 Sky High Co
 1461 University Ave.
 Berkeley CA 94702
 415-549-3472

Next issue we'll look closer at C. And that's Tidbits!



CP/M, MS-DOS EPROM PROGRAMMING SYSTEM

2708
 2758
 2516
 2716
 2732
 2764



2732A
 2764A
 27128
 27128A
 27256
 27CXX

- SOFTWARE AVAILABLE FOR CP/M-80, CP/M-86 AND MS-DOS SYSTEMS
- STAND ALONE BOARD - ELECTRONIC SWITCHING OF EPROM TYPES
- USES 24 VOLT XFMR FOR POWER - ALL SUPPLIES/TIMING ON BOARD
- NO PERSONALITY MODULES TO BUY - LARGE COMPREHENSIVE MANUAL

**** PARALLEL PRINTER INTERFACE ****

CONNECTS TO ANY PARALLEL PRINTER INTERFACE - USES 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE). BUSY LINE IS A HIGH SPEED SERIAL INPUT. FULL EPROM READING AND PROGRAMMING. UNIT MAY ALSO BE CONNECTED TO ONE 8 BIT INPUT/OUTPUT PORT.

**** CONTROL PROGRAM COMMANDS ****

- PROGRAM EPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- READ EPROM(S) INTO RAM
- VERIFY EPROM IS ERASED
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 11 SUB COMMANDS
- FILL-DUMP-XFER-EXAMINE-MODIFY-BIAS-PROGRAM-VERIFY, ETC.)
- SAVE EPROM(S) TO DISK
- PROGRAM EPROM(S) FROM RAM
- COMPARE EPROM WITH RAM
- COPY EPROM

ASSEMBLED & TESTED UNIT WITH COMPLETE DOCUMENTATION AND SOFTWARE ON DISKETTE PARTS KIT WITH SOFTWARE AND DOC-->\$179 PCB SOFTWARE & DOC-->\$69 SOFTWARE ON 8 & 5 1/4 DISK FOR KAYPRO, OSBORNE, IBM & OTHER FMTS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL

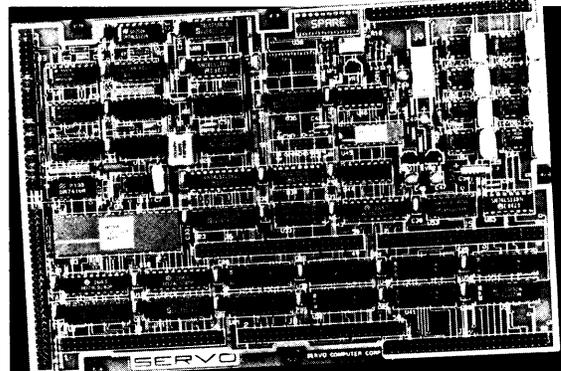
ANDRATECH
 P.O. BOX 222
 MILFORD, OHIO 45150
 (513) 752-7218

CALL OR WRITE FOR MORE INFORMATION -- ADD \$4.00 FOR SHIPPING OHIO RES. ADD 5.5% TAX -- VISA/MC. ACCEPTED -- \$3.00 FOR COD

SERVO INDUSTRIAL PROCESS CONTROL BOARD

14 Bit Lab Accuracy Industrial Durability

The Features of 5 Boards in One.



The Servo Process Control Board is a general purpose multi-channel analog and digital input/output controller which can communicate with and control most medium-scale industrial processes. Designed to work with the Servo Industrial Microcomputer, (or other computers which support the SASI interface), it offers great accuracy at prices starting as low as \$495.

- American made
- MILSPEC components
- 1 year warranty
- Maximum configuration includes:
 - 192 analog inputs
 - 64 analog outputs
 - 128 digital inputs
 - 512 digital outputs
- Based on the INTERSIL 14 Bit A to D converter
- Communicates with any micro or mainframe computer
- Easily programmed in high level languages (Turbo Pascal, Microsoft Basic, Forth)
- PID, FFT, Thermocouple linearization and laboratory DAS software supplied.
- Custom software and hardware engineering assistance available.



360-B N. Ellensburg St.
 Box 566
 Gold Beach, OR. 97444
 (503) 247-2021

SERVO COMPUTER CORPORATION



Z SETS YOU FREE!

Free to create computer environments right for you . . . free to automate repetitive tasks . . . free to increase your productivity. **Z-System**, the high-performance 8-bit operating system that flies! Optimized assembly language code — full software development system with linkable libraries of often needed subroutines —relocating (ROM and RAM) macro assembler, linker, librarian, cross-reference table generator, debuggers, translators, disassembler — ready to free you!

TERM III New generation communications package provides levels of flexibility, functionality, performance not available until now. Replaces BYE and XMODEM . . . master/server local area network capability . . . public or private bulletin board and electronic message handling are integral features . . . auto-dial/answer, menu install . . . XMODEM (CRC/Checksum), MODEM7 Batch, Kermit, CIS, and XON/XOFF protocols . . . 100-page manual **\$99.00**

Z-MSG Rolls Royce of message handling systems . . . mates with TERM III or BYE for most advanced overall electronic mail/file transfer capabilities . . . menu installed . . . extreme configurability . . . many levels of access and security . . . word, phrase editor, field search . . . complete message manipulation and database maintenance **\$99.95**

DISCAT Elegant, menu and command-line driven file and disk catalog manager. Generates and controls multiple master catalogs, working catalog used for update quickness. Nine flexible modules easily altered by user for custom requirements. Works with Z shells (VMENU, VFILER, MENU), aliases, and multiple commands per line **\$39.99**

ZCPR3: The Manual Bound, 350 pages, typeset book describes features of ZCPR3 command processor, how it works, how to install, and detailed command usage. Bible to understand Z-System **\$19.95**

ZCPR3 and I/OPS Loose-leaf book, 50 pages, 8-1/2" by 11", describes ins-and-outs of input/output processing using Z-System. Shows how to modify your BIOS to include I/O redirection . . . complements **The Manual** **\$9.95**

More missing links found — Z Application Programs! Fly with eagles! Our programs promote high performance through flexibility! Productivity results from dynamically changeable work environments, matching operator to tasks and machines.

Above programs require 48K-byte memory, ZCPR3, Z-Com, or Z-System, and Z80/NSC800/HD64180-based computer. Shipping from stock. State desired disk format, plus two acceptable alternatives. As payment, we accept Visa, Mastercard, personal checks, money orders, and purchase orders from established companies. We also ship UPS COD.

Call or write to place order or to obtain literature.



Echelon, Inc. 101 First Street • Suite 427 • Los Altos, CA 94022 • 415/948-3820

Having Your Pascal And Modula, Too

Excerpts From A Talk At SOG IV By Mike Weisert

Borland International
4585 Scotts Valley Dr.
Scotts Valley CA 95066

If you missed SOG IV then you missed this talk by Mike Weisert, one of the software writers at Borland assigned to create the Modula Tool Kit. If anyone knows Modula, Mike should.

Modula-2, a Pascal-like programming language, provides facilities missing from standard Pascal. Although it's distantly related to Algol, its direct precursors are Pascal and Modula-1.

Modula-2 originated when Niklaus Wirth, Swiss computer scientist, decided to use a single language for all his programming. Rather than modify an existing one, or embellish Pascal, he designed a new language.

Modula-2 (or just Modula) is equally suitable for expressing high and low-level algorithms and allows very large programs to be written in a clear, manageable form. It's a high-level language which can replace assembly language in all but the most time critical situations.

Modules

Wirth modules are similar to Pascal programs, but more flexible, thanks to nested modules and libraries.

Pascal allows nested procedures, but Modula-2 can separate visibility from existence. See Figure 1. You create a library of subroutines by splitting a module into definition and implementation parts. The definition is an overview of a module; the implementation provides the details. See Figure 2.

The definition acts as an interface to the implementation. This allows —

1. The user to avoid looking at actual code.
2. The compiler to check data types between modules.
3. Changing a module without changing users of that module.

A module can be a program, an instrument for hiding information, and a library unit.

Low-level Facilities

Pascal and Modula approach low-level facilities differently. Although subject to implementation dependencies, the basic set of low-level facilities

is defined in the Modula language (and is in a relatively standard form in all implementations). Not true for Pascal.

The best way to use these low-level hooks and still maintain some degree of portability is to encapsulate system dependent features in a module. Thus, to move a program to a new system you rewrite one low-level module. (If a low-level module already exists for the new system, you might just need to recompile.)

Quasi-concurrent Processing

Co-routines look like procedures that never end and are entered and exited with an explicit TRANSFER statement. They're easily implemented using the type Process imported from the SYSTEM module.

Co-routines "stay alive" when not being executed. They can be exited at any point and later resumed from the point of exit.

All variables remain intact because each co-routine has its own stack for local variables and activation records (procedure calls, for example).

You might use a co-routine to handle interrupts. See Figure 3. Modula-2 has a special construct — the procedure IOTRANSFER.

(continued next page)

Figure 2 -
Definition and Implementation Modules

```
DEFINITION MODULE ScreenIO;

EXPORT QUALIFIED
  ClearScreen, ClearToEOL, GotoXY;

PROCEDURE ClearScreen;

PROCEDURE ClearToEOL;

PROCEDURE GotoXY(X,Y: CARDINAL);

END ScreenIO.

IMPLEMENTATION MODULE ScreenIO;
FROM InOut IMPORT WriteInt;
FROM Terminal IMPORT WriteString, Writes;
FROM Strings IMPORT Length;

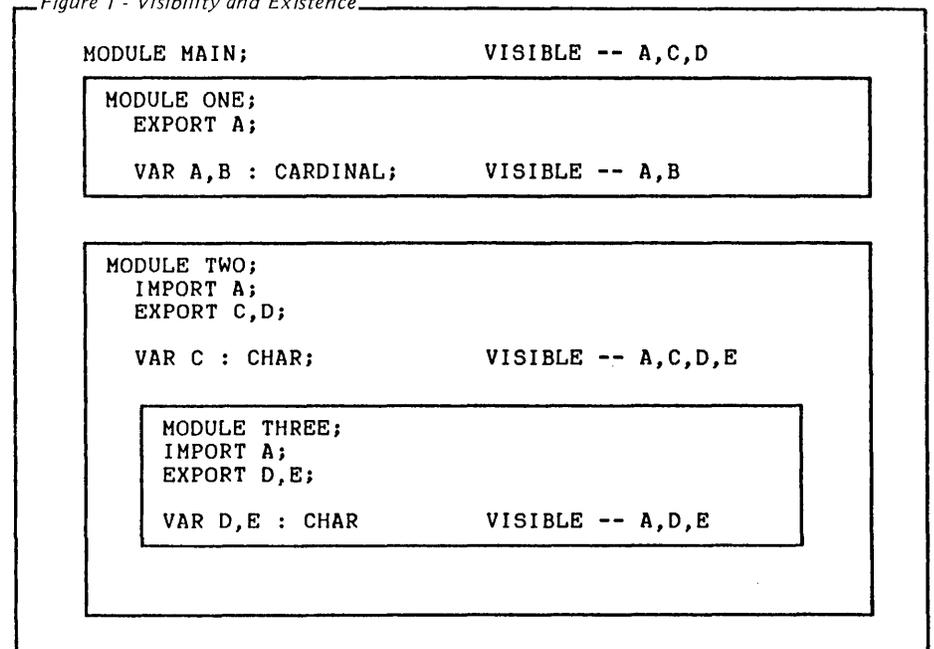
PROCEDURE ClearScreen;
BEGIN
  Command('2J');
END ClearScreen;

PROCEDURE ClearToEOL;
BEGIN
  Command('K');
END ClearToEOL;

PROCEDURE GotoXY(X,Y: CARDINAL);
BEGIN
  Command('');
  WriteInt(Y,1);
  Write(' ');
  WriteInt(X,1);
  Write('H');
END GotoXY;

PROCEDURE Command(s: ARRAY OF CHAR);
CONST
  Esc = 33C;
BEGIN
  Write(Esc);
  Write('');
  IF Length(s) > 0 THEN
    WriteString(s);
  END;
END Command;
END ScreenIO.
```

Figure 1 - Visibility and Existence



IOTRANSFER goes in the interrupt handler and specifies which interrupt vector will cause an unscheduled transfer back (to the interrupt handler). So, when an interrupt occurs the handler gets control, processes the interrupt, and then executes IOTRANSFER which returns control to the suspended program.

Modula-2 also provides a mechanism for assigning module priorities. Lower modules, of course, can't interrupt higher modules.

Absolute Variables

Absolute variable declaration is nice for machine specific code. For example, you can declare the variable "screen" at the address of a memory-map screen buffer by placing the address in square brackets after the name in the variable declaration. This allows you to place any type of variable in any memory location.

Unlike Pascal

Modula-2 allows procedures to be passed as parameters to other procedures. You can't do that in Turbo Pascal (although you can in standard Pascal).

The difference in this case between Modula and Pascal is that Modula allows procedures to be declared as variable types, and variables to be declared as a procedure types.

Type Transfer Functions

As in C and Turbo Pascal, Modula-2 provides a way of explicitly corrupting types. By prefixing a variable with a type identifier, you can cause the compiler to temporarily view a variable as something other than its defined type. However, this works only with types which use the same amount of memory. No code is generated by a type transfer function — it's just relaxed compiler type checking.

Thus, the following transfers are permitted:

```
CardVar := CARDINAL(PtrVar);
PtrVar := ADDRESS(CardVar);
```

whereas these are not permitted:

```
CardVar := CARDINAL(RealVar);
RealVar := REAL(BoolVar);
```

This explicit type corruption is powerful when you need to do things like pointer arithmetic or masking the bits of a character.

Sets

Modula-2 and Pascal look at sets differently. In Modula, each element of a set is a bit rather than a byte. And the number of elements allowed in a set has been restricted to the computer's word length — in most cases, 16 bits.

In Modula four operations on sets are allowed —

1. Union (logical OR)
2. Difference
3. Intersection (logical AND)
4. Symmetrical Intersection (logical XOR)

Operators are included for assignment, inclusion, and exclusion of elements, as well as the test for membership (IN).

Some Pascal programmers may miss having large sets (e.g., SET OF CHAR). However, these constructs can easily be implemented with procedures, whereas the bit-level logical operation provided in Modula-2 had to be implemented as extensions to Pascal.

The Borland Implementation

Borland's Turbo Modula-2 is a complete Modula-2 development system consisting of an environment shell which provides necessary development tools, including:

- A WordStar-like editor
- File management utilities
- A native code, overlay linker
- A module librarian
- Changeable default compiler switches

Turbo Modula-2 is also an incremental compiler, allowing you to edit as you compile. The language includes an M-Code interpreter based on the Lil-

```
Figure 3 - Interrupt Handling

MODULE InterruptDemo;

CONST
    keyint = OFOH; (* keyboard interrupt vector number *)

VAR
    main, inthandler, background : PROCESS;

PROCEDURE backgroundjob; (* a coroutine *)
BEGIN
    LOOP
        (* some processing *)
    END
END backgroundjob;

PROCEDURE keyboardjob; (* a coroutine *)

BEGIN
    (* initialize keyboard interface chip, set its
       interrupt vector to keyint
    *)
    LOOP
        IOTRANSFER(inthandler,background,keyint); (* wait for char *)
        (* Interrupt has now arrived. Fetch char and process it *)
    END
END keyboardjob;

BEGIN (* main program *)
    (* Allocate workspaces for the coroutines. Create
       coroutines by calls to NEWPROCESS.
    *)
    TRANSFER(main,inthandler);
END interruptdemo.
```

ith's (Wirth's computer) instruction set. Therefore, no linking is required to run a program.

The language provides complete support for Modula-2, and three extensions have been included as well:

1. String assignment and comparison
2. Pascal-like IO
3. Ada-like exception handlers

Why Not Just Use Turbo Pascal?

A lot of people have asked me what they would gain by switching to Modula-2. Good question.

For starters, Modula-2 is better suited for large programs. And it provides better support for libraries of routines.

On the other hand, Pascal (Turbo, in particular) is ideal for testing ideas which can then be easily included in a Modula library.

(Editor's note: Borland hasn't yet released Turbo Modula-2 in other than Beta-test form, but other Modula-2 compilers are available for CP/M and MS-DOS.)



IBMPC/MSDOS/CPM SOFTWARE

MBACOUNT/\$89.00: Small business double entry accounting. Up to 40 asset, 40 liabilities, 40 income, 80 expense & 20 capital accounts user assignable. Prints checks, P & L, balance sheet, chart of accounts, single account details, check ledger, etc. Generates a printable detail journal for audit trail.

A/R, A/P, PAYROLL INVENTORY: These programs are available as either stand alone software or MBACOUNT integrated modules. Stand alone **\$79.00 ea.** Integrated **\$50.00 ea.**

PROFBILL/\$149.00: Time keeping and billing for all professional people who bill at various hourly rates. Handles 400 clients, 10 partners/employees. Prints billings, statements, aged billings, partner/employee hours, etc. May be used stand alone or integrated with MBACOUNT.

SBACOUNT/\$79.00: Small business single entry system. Up to 99 income and 99 expense categories, user assignable. Prints checks, P & L (month, quarter, YTD), single account details, check ledger. Generates a printable income/expense journal for audit trail.

MAILLIST/\$89.00: Mailing list/filing program. 10 fields per record and up to 900 records per file, files limited only by disk space. Sort/print all or partial list by most fields on 1, 2, or 3 across labels on 80 col. printer. Search, add to, delete or edit by most fields.

MBADATA/\$89.00: A data/base-filing system usable without learning any special language or control codes. Up to 253 characters, 24 fields, 900 records per file. Sort, print, display any fields in any location in any order desired. Menu driven with step by step prompts!

STANFORM/\$79.00: Need to fill out pre-printed forms? Then this program was made for you! Generates programs for printing data in the proper location on any pre-printed form that will fit into your printer. Each form need be generated only once as it is saved on disk.

\$FINANCE/\$79.00: Personal finance, prints checks, posts to as many as 120 user assignable expense categories, posts checks & deposits, reconciles multiple checking accounts, etc. Similar program for up to 9 charge cards. Extracts/prints listings for various categories.

TOUCHTYP/\$49.00: Parallels high school/college level touchtyping course. Displays typos, final score. Timed speed drill mode. Electronic typewriter mode converts your printer to a one line memory electronic typewriter with settable margins, tabs, line spacing, etc.

MSBCHART/\$49.00: Prints alphabetized list of all variables & line numbers referenced in any program written in basic cross referenced to the line number where used. A godsend when writing or modifying basic programs.

Place your order or request **FREE** catalog by mail or if using COD or credit card, telephone anytime of the night or day. Add \$3.00 shipping & \$3.00 additional for COD. Shipment will be no later than next working day by 1st class mail. Californians must add 6% tax to prices.

MAP

Micro-Art Programmers

173 Birch Avenue, Cayucos, California 93430 Phone: (805) 995-2329

DRIVE ENCLOSURES

FLOPPY
8" & 5"

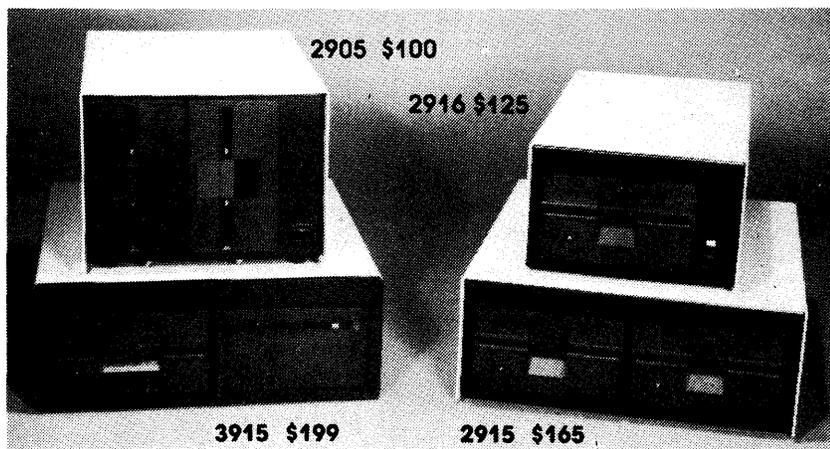
TAPE
5" & 8"

FLOPPY-WINCHESTER-TAPE

FROM \$80⁰⁰

(Includes Power Supply)

WINCHESTER
5" & 8"



- Desktop & Rack
- Heavy Duty All Metal Cabinet
- Fan & Dust Filter*
- Hefty Power Supplies
- Full or Slim Drives
- Power Harness From Supply To Drives
- Line Fuse, EMI Filter*, Detachable Line Cord
- Cabinets & Supplies Available Separately

* = Most Models (Disk drives not included)

INTEGRAND

RESEARCH CORPORATION

8620 Roosevelt Ave./Visalia, CA 93291

209/651-1203

Write or call for our brochure which includes our application note:
"Making micros, better than any ol' box computer"

32 Page Free Fakt Pakt Catalog

WANT ADS

The following folks are reaching you for only 20 cents per word. If you would like to reach the same audience, send your words and 20 cents for each to Micro Cornucopia.

Drive Cleaning Kit Sale! 8" only, floppyclene with chemicals and carriers. Reg. \$69.95, now \$20 each + \$2 postage. Datalife Head Cleaning System, Reg. \$11.95 and extra head cleaning pack (10) Reg. \$20 - now just \$6 + \$2 postage. National Public Domain, 1533 Avohill Dr., Vista, CA 92084. 33

Your Xerox 820-1 becomes Kaypro for \$105. Operate your drives in single or double density, use RS232 terminal, 820-1 board and power supply. Simple 2 cut 4 jumper mod, EPROM, double density disk driver board. Satisfaction guaranteed or money refunded. Mail \$105 to Barry Cole, 12770 Matteson Ave. #1, Mar Vista, CA 90066 today! Other versions available. (213) 390-3851. 28

Big Board II, keyboard, monitor, printer, software, 5 floppy drives, extras, \$600. (503) 640-5367. 28MU

\$25 Keyboards for computer builders - 83 keys, full ASCII; upper/lower case, all control characters, numeric pad, caps-lock, repeat, self-test! Brand new, hundreds sold to builders of Apples, Big Boards, Xerox 820s. Parallel output, positive TTL logic, strobe. Uses only 106mA of +5 volts. Custom case available. 90 day warranty unmodified. Keyboard \$25. Documentation (21 pgs./cable package \$5. Spare custom CPU/ROM \$4. All 3 (\$34 value) \$30. UPS additional, 5 lbs. Call/SASE for detailed spec sheet. Electrovalue Industrial Inc., Box 376-MC, Morris Plains, NJ 07950. (201) 267-1117. TF

Public Domain UG Software Rental: CP/M UG Vol 1-92 on 46 8" floppies, \$45, SIG/M UG Vol 1-240 on 108 8" floppies, \$155. PICONET Vol 1-34 on 17 8" floppies, \$25, Pascal-Z UG Vol 1-25 13 8" floppies, \$25, KUG (Charlottesville) 39 disks, \$25, IBM-PC SIG 1-390 PC-DOS, \$410, PC-Blue 1-175 MS-DOS, \$154, directory disks \$5 PP. Also available on 209 5.25" formats. Rental is for 7 days after receipt with 3 more days grace for return. Credit cards accepted (preferred). Downloading-disk format conversions. Call. User Group Software Automatic Update Service, \$7.50 per 2 volume set PP. 619-727-1015 info. 24 hrs. 619-941-0925 9-5. National Public Domain Software Center, 1533 Avohill, Vista, CA 92083. TF

TBKUG/DataCOM Super Systems (tm) proudly announces the arrival of the new SUPER System!! The remote multi-user database for the novice, the semi-pro, or seasoned hacker. Let your modem be your guide to the most current public domain for CP/M-80, MS/PC-DOS, TurboDOS, Turbo Pascal, Big Board, MEX, ZCPR3, CP/M-86, BASIC, Assembler, Kaypro, WordPro, and many other special interests. We are open 24 hours a day and accept 300, 1200, and 2400 baudrates. This is a multi-user TurboDOS-based S-100 system featuring state of the art equipment and dedicated to serving all your telecommunications needs. Membership fees are: \$35 per year for standard access or \$125 per year gets you a VIP membership that includes UNLIMITED system usage and a monthly update disk of CP/M software in Kaypro format. Join over 700 other users who have made the "right" choice. Fees are refundable if you're not satisfied within the first 30 days! Contact Steve Sanders at: TBKUG/DataCOM, 2643 Cedarview Court, Clearwater, FL 33519, or call (813) 791-1454/55 by modem and download an application. 30

Disk Drive Sales & Service. For Sale — Shugart SA 800-2, \$79; SA 850, \$129; SA 850/851R, \$139; MPI 52S 5.25" DSDD, \$55; New TEC FB 501 5.25" SSDD, \$59; Case w/power supply for two half-hgt. drives, \$59. SERVICE — SA 800/801, \$25; SA 850/851, \$35; 5.25" SS, \$30; 5.25" DS, \$40; 8" SS, \$40; 8" DS, \$50. All drives and service guaranteed for 60 days. Prices do not include parts or shipping. LDL ELECTRONICS, 13392 158th St. N., Jupiter, FL 33458 (305) 747-7384. 28LY

Hardware for the Kaypro Computer voice synthesizer unlimited speech \$70. EPROM programmers 2516, 2716, 2532, 2732, 2764 \$110. General purpose power supply used with EPROM programmers, \$30. All hardware comes in kit or bareboard form with software and schematics. Call or write Busch Computer, 395 Totoket Rd., Northford, CT 06472, (203) 484-0320. 29BU

"Getting Started With CBASIC" - Workbook, tutorial style for beginners/intermediates in this powerful language. Fundamentals through file handling, user functions, etc. Ends confusion, frustrations! Only \$25/U.S. Four Corners Press, Hanover, MA 02339. 30LI

For Sale: Slicer Computer 8 meg, two 8" DSDD drives, Okidata 92 printer, Wyse-50 terminal. \$1500. Contact Ken Horton, 324 Heritage Oaks, Eugene, OR 97405, (503) 485-0375. 28

5.25" disk drives. Reconditioned full height Tandon and CDC DSDD/SSDD. All are in excellent condition and have a new appearance. Each is tested, aligned and guaranteed. Price each: DSDD \$49.95, SSDD \$34.95 + \$5 shipping. Quantity discounts available. JANICK DATA, 1869 Riverbirch Dr., Sumter SC 29150, (803) 481-9205. 28

Computers and Equipment for sale: Kaypro IV \$700. Imsai based homebrew \$600. Houston Instruments Hi-Pad digitizer \$650. Details? SASE. D. Anderson, 1704 Virginia Way, Arcata, CA 95521. 28

Perfect Calc overlays for CP/M. TAX 1040: \$36.50. Schedules: ABCDEGW, 4562, SE. Six associated spreadsheets. Depreciations. Income averaging. Updates. Particularly fast and practical with hard or RAM disk. (Users Guide #14)// PAYROLL: \$74.50/ calculates your payroll. 941, 940 forms. Paychecks. FITC, FUTA, State PIT, DI, ETT, UI, SS. Eight employees. Limits. Exemptions. Allowances.// Manuals, support. Check, money order: Cal. add 6.5% sales tax. Format: Kaypro II, others. F. Wilhelm, 1298 Windermere Way, Concord, CA 94521, (415) 671-7309. 28

Big Board I enclosed with two 8" drives, keyboard, monitor, modem. Software: Userdisks, word processing, Turbo Pascal and 40 other disks. Wayne Naleway, 2836 N. Harper Ter., Peoria, IL 61604. (309) 686-1304. \$650. 28

For Sale: Xerox 820-1, two 5.25" drives, two monitors, keyboard, software, 8" drive, \$350. Call Dick Barney eves (607) 776-3935. 28

Kaypro II with software & carrying case \$500. Okidata 92 printer \$250. New Amdek 300A amber monitor \$100. Jill Wollins, 1204 17th Ave. NW, #6, Rochester, MN 55901. 28

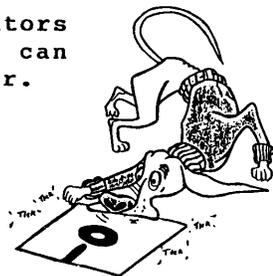
Interested in 32-bit Technology

*Systems & applications level programmer wanted to join the Definicon team.
Call Trevor Marshall (805) 492-3693 or (818) 889-1646.*

Ever Wondered What Makes CP/M® Tick?

Source Code Generators
by C. C. Software can
give you the answer.

"The darndest thing
I ever did see..."
"... if you're at
all interested in
what's going on in
your system, it's
worth it."
Jerry Pournelle,
BYTE, Sept '83



The S.C.G. programs produce
fully commented and labeled
source code for your CP/M
system (the CCP and BDOS
areas). To modify the system to your liking,
just edit and assemble with ASM. CP/M 2.2 \$45,
CP/M+ \$75, + \$1.50 postage (in Calif add 6.5%).

C. C. Software, 1907 Alvarado Ave.
Walnut Creek, CA 94596 (415) 939-8153

CP/M is a registered trademark of Digital Research, Inc.

8" CP/M 80 Disks from Micro Cornucopia

USERS DISK #2

- 1-Two single disk drive copy programs, both with source
- 2-Crowe Z80 Assembler source
- 3-New Crowe.COM file, debugged version
- 4-New CBIOS with parallel print driver & other extensions for CP/M 1.4 & 2.2
- 5-Disk mapper with source

USERS DISK #3

- 1-EPROM burning software for BB 1
- 2-Reset bit 7 (unWordStar a file)
- 3-Disk file CRC checker
- 4-New fast copy program & source
- 5-DU77, disk inspector/editor
- 6-FINDBAD, isolates bad disk sectors
- 7-Print fancy page headings

USERS DISK #4

- 1-CBIOS, custom bios for Tandon drives
- 2-ZCPR, dynamite CCP checks drive A for missing .COM files; improved commands
- 3-ZCPRBLOC, identifies CCP location

USERS DISK #5

- 1-CAT, disk cataloging routines
- 2-Modem 7 for Port A
- 3-Modem 7 for Port B
- 4-PACMAN, the arcade game
- 5-FAST, buffers the disk to speed up assemblies
- 6-NOLOCK, removes BB 1 shift lock
- 7-VERIFY, cleanup & verify a flaky disk
- 8-DUMPIX, enhanced for BB 1
- 9-UNLOAD, create .HEX file from .COM file

USERS DISK #6

- 1-REZ, 8080/Z80 disassembler, TDL mnemonics
- 2-PRINTPRN, prints Crowe listings
- 3-RUNPAC, run-time utility package for 8080 assembly language programs. Has 51 functions. Includes source which assembles under ASM

USERS DISK #7

- 1-CHNGPFM, PFM monitor mods
- 2-TERM, terminal routines let you set up BB as simple terminal, as a file receiver, or as a file sender
- 3-Checkbook balancing package
- 4-Disk Utilities - copy to memory, from memory, and dump

USERS DISK #8

- 1-BDSCIO, custom BDSC I/O for BB 1 (both .h and .c)
- 2-YAM, Yet Another Modem program in source & COM form. Turns BB into paging intelligent terminal, complete with printer interface, baud rates to 9600
- 3-ROFF, text formatter
- 4-SIGNS, prints large block letters

USERS DISK #9

- 1-ADVENTURE, expanded 550 pt version
- 2-Keybord translation program
- 3-CBIOS, serial & parallel printer interface
- 4-EPROM programming package for BB II, for 2732s only

USERS DISK #10 - Lots of Disk Utilities

- 1-REBOOT, sets up the CP/M auto load
- 2-SWEEP, directory/file transfer routine
- 3-A, Lets BB I recognize a double sided drive as one drive with 494K of usable space
- 4-FIX, super disk utility, does everything, much easier to use than DU77
- 5-Compare files routine
- 6-UNERA, retrieve erased files
- 7-FIND, check all drives on system for a file
- 8-MENU, menu program for CP/M
- 9-NEWCAT, enhanced disk catalog program
- 10-Single drive copy program that does track by track copies rather than file by file

USERS DISK #11 - Printer Utilities

- 1-Microline 92 printer routine
- 2-Graphics display package for MX-80 with Grafrax, very fancy
- 3-Epson MX80 setup for BB I with 59.5K CP/M

USERS DISK #12 - Games for BB I

- 1-ALIENS, a fast, exciting arcade game
- 2-ZCHESS, chess with a 1-6 level look ahead
- 3-MASTERMIND, match wits with the computer
- 4-BIO, Biorhythm charts complete with graphics on the BB I
- 5-LIFE, so fast it's real animation!
- 6-CRAP, see how much you'd lose in Vegas
- 7-WUMPUS, a caver's delight, kill the Wumpus or be killed
- 8-PRESSUP, similar to Othello
- 9-Games, 7 games in one program, includes blackjack, maze and animal

USERS DISK #13

- 1-ZZSOURCE, disassembles to real Zilog mnemonics
- 2-EX14, superset of submit or supersub
- 3-MOVPATCH, lets you use MOVECPM on other copies of CP/M
- 4-XMON, 3K expanded BB I monitor, use in ROM or as overlay
- 5-CURSOR, prompts you for cursor char you want
- 6-UMPIRE, very fancy RAM test
- 7-ZSIDFIX, display improvement for ZSID
- 8-PIPPAT, modify PIP so you can reset system from within PIP
- 9-@, Lets you use the BB as a calculator, including HEX
- 10-SORT, sort package written in C80

USERS DISK #14 - BB II Software

- 1-PRO32, latest 2732 reader & programmer
- 2-SMODEM2, lets BB II talk to Hayes smartmodem
- 3-GRAFDEMO, demonstrates BB II graphics (in BASIC)
- 4-ATTRTEST, demonstrates BB II graphics (in JRT Pascal)
- 5-INITISIO, initializes port B for 300 or 1200 baud
- 6-MENU, displays menu of .COM files, enter number to run file
- 7-SETCLK, sets realtime clock built into BB II
- 8-PRINT2, modified print which accesses BB II clock
- 9-BOX, draws a thin line box on screen
- 10-ALIENS, space invaders arcade game
- 11-LISTSET, printer interface, auto-enables RTS, ignores DCD

USERS DISK #15 - Word Processing

- 1-EDIT, very fancy line editor similar to EX (Unix)
- 2-TED, simple minded line editor
- 3-TTYPE, typing training program written in BASIC
- 4-TINYPLAN, very simple-minded spreadsheet.
- 5-C80 Text Utilities
- 6-CHOP, cuts off file after N bytes
- 7-ENTAB, replace spaces with tabs where possible
- 8-MS, double or triple spaces a file to output
- 9-RTW, removes trailing spaces from file
- 10-TRUNC, truncates each line to specified length
- 11-WRAP, wraps at column 80, plus pretty pretty printing, page #s...

USERS DISK #16 - BB I Modem Software

- 1-RCPM27, list of U.S. bulletin boards
- 2-SMODEM, interfaces BB I with Hayes Smartmodem
- 3-PLINK66, easy to use with non-CP/M host
- 4-BBPAT, menu selection of BAUD rate, bits/char, parity, & stop bits
- 5-MODEM 7+, Modem 7 plus BBPAT, lets you talk to anything from port A

8" Users Disks....\$12.00 each

USERS DISK #17 - Small C Version 2

SMALLC2, this substantially expanded version of Small C now includes for, goto, label, switch (case); external declarations; new preprocessor commands; expanded I/O includes redirection; initializers; plus 12 new expressions. The I/O and runtime libraries have been greatly expanded (including printf). Source & documentation on one disk.

USERS DISK #18 - FORTH

IFORTH, this is Idaho FORTH which can be burned into ROM or loaded from disk. It replaces the PFM monitor & handles all the monitor functions.

USERS DISK #19 - BB I Double Density

New BB I Monitor, BIOS, character ROM, Winchester Interface, ZCPR, and formatter from Trevor Marshall. See BB I expansion article in Issue #11.

USERS DISK #20 - Assemblers

CROWEASM: This is the Crowe assembler modified so that it runs on any CP/M system (including the BB I, BB II, Xerox...). LASM: This assembler is similar to the ASM that comes with CP/M except that it can link files at assembly time. PRINTPRN: Print routine for CROWEASM.PRN files. LIBRARY: Utilities which let you combine many files into one, then you can run, type, or extract any file within the larger system.

USERS DISK #21 - Winchester Utilities

BACKUP: Helps you back up the winchester onto multiple floppies. FLOPCOPY: Lets you make floppy copies (with only one floppy drive). BIGBURST: Backs up a very large winchester file onto multiple floppies. MULTCOPY: Use this like PIP but it prompts you to change disks. Accepts ambiguous file names. MDIR: Displays files in all user areas on selected drive.

USERS DISK #22 - Pascal Compiler

This is a real Pascal compiler. It supports only a subset of the language (no records, pointers, booleans, reals or complex) but it generates a real .COM file. Everything is on this disk: the compiler, its source, example programs and documentation.

USERS DISK #23 - Xerox Utilities

This disk contains Xerox specific utilities including a screen dump from Wayne Suga (with source); modifications for the SWP package including ZCPR, a new monitor, and a clock/calendar from Mitch Mlinar; and Jim Mayhugh's new monitor (see issue 19). A very special disk for Xeroxers.

USERS DISK #24 - Prowriter Graphics

This is a complete Prowriter printer graphics package written by the same Micro C subscriber who wrote the MX-80 graphics package. Plot points, lines, circles, boxes, and more. Examples, documentation.

USERS DISK #25 - Z80 Macro Assembler

This is a real Z80 macro assembler! Syntax closely follows RMAC and MAC. Also includes pseudo-ops to support conditional assembly etc. No phase or relocatable code.

USERS DISK #26 - BB II CP/M 3.0 Banked BIOS/ Winchester

Support CP/M 3.0 Banked BIOS implementation for the BB I. Roy Epperson's software to support the Adaptec ACB-4000 SCSI and the Rodime R204 5" Winchester on the BB II (see issue #19). Plus more Winchester programs.

USERS DISK #27 - BYE Remote CP/M System

BYE programs to run your BB I, BB II, OR XEROX 820-I as a remote CP/M system using a Hayes Smartmodem compatible modem. Includes programs to allow restricted access.

USERS DISK #28 - VFILER and Extended Single Density

VFILER is a screen-oriented file manipulation utility, similar to SWEEP, CLEAN, and DISK. Also, Larry Blunk's documentation and software for implementing extended single density (334K) on eight inch disks.

USERS DISK #29 - MODEM740

The latest incarnation of the immortal MODEM7 program. Too many features to list. Versions for both ports of BB I (Xerox 820) & BB II.

We Also Have 8" Disks For CP/M 86

Ask For Catalog

MICRO CORNUCOPIA

P.O. Box 223 • Bend, Oregon 97709

Order Number 95 PST (503) 382-5060 • Technical Information 9-Noon PST Only (503) 382-8048

BOOKS, MAGAZINES, AND T-SHIRTS FROM MICRO C



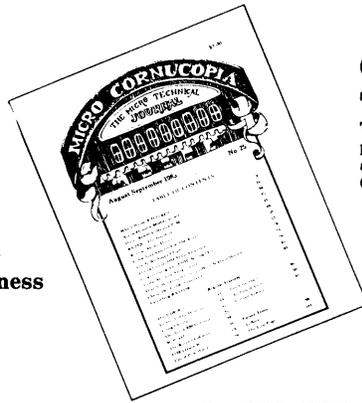
Inside CP/M
 By David Cortesi
 \$27.95 US, Can, Mex
 \$37.95 (Other Foreign)
 This is one of the best books on CP/M for everyone, users and programmers. It's our reference here at Micro C.



Your Fortune in the Micro computer Business
 By Victor Wild
 \$26.45 (US, Can, Mex)
 \$36.45 (Other Foreign)
 This is the best, most complete collection of working for yourself information We've found.



The Small C Handbook
 By James Hendrix
 \$17.95 (US, Can, Mex)
 \$22.95 (Other Foreign)
 If you really want to solve the mysteries of the Small C compiler (or compilers in general) you'll want this book.



ORDER YOUR MICRO C T-SHIRT TODAY!

These two-color tan shirts are the perfect software. Sizes S, M, L, and XL.
 Only \$6.95 ppd. (\$8.50 all foreign).



MICRO CORNUCOPIA MAGAZINE

Subscription Rates

1 yr US \$16.00
 2 yrs US \$30.00
 3 yrs US \$42.00
 1 yr Can & Mex \$22.00
 1 yr Other Foreign ... \$30.00

Inside information on the latest processors and languages.

MICRO CORNUCOPIA ORDER FORM

QUANT.	DESCRIPTION	PRICE EACH			TOTAL
		U.S.	CAN & MEX	OTHER	
	SUBSCRIPTION (6 issues per year) <input type="checkbox"/> New <input type="checkbox"/> Renewal	1 yr. \$16 <input type="checkbox"/> 2 yr. \$30 <input type="checkbox"/> 3 yr. \$42 <input type="checkbox"/>	<input type="checkbox"/> \$22	<input type="checkbox"/> \$30	
	USERS DISKS: <i>please specify #s</i>	\$12	\$12	\$12	
	KAYPRO SCHEMATIC PACKAGES <i>please specify #s</i>	\$20	\$20	\$20	
	BACK ISSUES: Check ad for available issues <i>#s</i>	\$3	\$3	\$5	
	T-SHIRT (<i>specify S,M,L,XL</i>)	\$6.95	\$8.50	\$8.50	
	FREE CATALOG				
	OTHER ITEMS:				
TOTAL ENCLOSED					

NAME _____
 COMPANY _____
 ADDRESS _____
 CITY _____
 STATE _____ ZIP _____

Are you a current subscriber to Micro Cornucopia? yes no

Check or money order enclosed
 (U.S. funds only drawn on a U.S. bank)
 Prices include postage (Air Mail for Other foreign)

Make checks payable to:
MICRO CORNUCOPIA

What kind of computer do you own/operate? _____

Card No. _____ Exp. _____ Signature _____
 Visa MasterCard

MICRO CORNUCOPIA

P.O. Box 223 • Bend, Oregon 97709

Order Number 9-5 PST (503) 382-5060 • Technical Information 9-Noon PST Only (503) 382-8048



The Micro C MS-DOS Library

MS-1 Essential Utilities

SWEEP, LU86403, LDIR103, WHEREIS, SQUEEZE, UNSQUEEZE, WASH

MS-2 Cheap Assembler & Disassembler

CHASM, ASMGEN, MEMBRAIN, UNWORDSTAR, FSPool, RAMDISK, SDIR, SDIR SOURCE

MS-3 Adventure

The original and most cussed adventure in C, with source.

MS-4 Single User RBBS

A decent bulletin board with source.

MS-5 File & Directory Utilities

UTIL, Z, VTYPE, ST, PROTECT, UNPROTECT

MS-6 fig FORTH

We combined fig FORTH source (MS-7) with this disk so you get both on one disk for \$12.00.

MS-8 Super Games

CASTLE, a graphic and conversational adventure (note: there's a bug that changes the game once it's been saved). TREKRUN, Star Trek Command adventure, and BOUNCING BABY, a finger exercise.

MS-9 PC Musician, Pascal Games, Calculator

You might call this an educational disk. PC MUSICIAN teaches you music while you're writing songs. (They play right on the PC.) 15 PASCAL GAMES with source. The games aren't memorable but they are good examples of Turbo code (we include the source) PROGRAMMER'S CALCULATOR shows the internal workings of a stack, memory, and registers.

MS-10 Modems

QMODEM and KERMIT

MS-11 Multi-User RBBS 12.3

Multi-user Remote Bulletin Board System with source disk (MS-12) included free.

MS-13 Editor & Formatter

ROFF4, PC-WRITE, and manuals on disk. Very popular freeware.

MS-14 FORTH 83

MS-15 Deskmate, Zap, Unerase

A sidekick lookalike, disk inspector, and more utilities.

MS-16 Piano Man & Other Tunes

Turn your keyboard into a piano.

MS-17 Arcade Games

PANGO, PYRAMID, BRICKS, and more. Good audio and graphics.

MS-18 Typing Tutor

A nice touch-typing practice tool. Keeps track of your keystrokes, errors, and practice time. Calculates accuracy and words/minute.

MS-19 Prolog, XLisp

Introductory interpreters to improve your artificial intelligence.

All disks are PC Standard DS/DD 360K — \$12.00 each, ppd.

MICRO CORNUCOPIA

P.O. Box 223 • Bend, Oregon 97709

Order Number 9-5 PST (503) 382-5060 • Technical Information 9-Noon PST Only (503) 382-8048

ADVERTISER'S INDEX

\$Ware Tools For Investors	33
Advanced Concepts Engineering	23
Ampro Computers	88, 89
Anaheim Software	86
Andratech	95
Aprotek	85
BD Software	7
BV Engineering	82
Borland International ... Inside Front Cover	
Cascade Electronics	79
C.C. Software	100
Computer Helper Industries	58
Computer Journal	84
Computer Surplus Store	83
Computing Technology	22
Consumertronics	10
Corvatek	80
Craig's Computers	31
The Data Mill	31
Desktop Publishing	68
Digital Dynamics	87
Digital Research	67
Earth Computers	20
Echelon, Inc.	40, 96
EcoSoft	24
Emerald Microware	29
Epic Sales	36
ERAC Co.	62
E21 Computer	86
Ferguson Engineering	32
Forbes Engineering	73
Hawthorne Technology	82
Holliston Computer	1
The ICU Group	19
Integral Systems	83
Integrand	99
Intersecting Concepts	45
Kamasoft	41
Kenmore Computer Technologies ...	61
KYSO Technology	24
Laboratory Microsystems	28
L-Band Systems	44
Manx Software	44
Micro-Art Programmers	99
MicroCode	28
Microcomputer Systems	81
Micro Cornucopia . 92, 93, 101, 102, 103	
MicroGlyph Systems	83
Micromint	91
Microprocessors Unlimited	73
MicroSphere	45
Micro Supply	12, 13
Mitek	65
National Public Domain Software	7
Oliver Computers	74
Orr Corp.	46
PC Tech	Back Cover
Periphco	15
Phenix Co.	31
PluPerfect Systems	46
Poor Person Software	87
Qua Tech	61
Rolland Management	51
SLR Systems	11
Second City Software	47
Servo Computers	95
Soft Advances	35
Source Information	44
Southern Pacific USA	11
Superior Electronics	82
Systems Peripherals Consultants	46
Tiny Tek	24
Tomorrow's Computing	85
Unified Software	44
United Products	Inside Back Cover
Viking C Systems	27
Western Wares	27
WW Components	74

The Last Page

By Gary Entsminger

A logical assembly of subprograms called a module distinguishes Modula-2 (or more colloquially, "Modula") from Pascal.

If you need to use a subprogram or procedure that's in a module, you import it.

Modules For Modula I/O

Like C, the Modula compiler doesn't include any statements for input and output; I/O is handled in the module library that's included with the compiler.

Let's assume (for example) your Modula compiler's I/O procedures have been assembled (collected) in a module called "InOut", and in the module you're writing you need to output a linefeed and carriage return. How can you manage it?

First, tell the compiler where to find the desired procedure (WriteLn), import it, and then call it —

```
MODULE CurrentAct;
  From InOut IMPORT
    WriteLn;

  BEGIN
    WriteLn;
  END CurrentAct.
```

Defining And Implementing

Now, let's assume you have lots to keep track of; you use many modules (Mathlibrary, FileSystem, Windows, etc., etc.) which contain many procedures (Put, Get, Sum, Power, WriteBlock, etc., etc.). Usually you'll want the results of a procedure, but won't care how the procedure generates them.

Herein lies a strength of Modula — it lets you conceal the unwanted details by separating modules into two parts: a definition and an implementation.

The definition part of a module contains the declarations of exported identifiers: CONSTANT, TYPE, VARIABLE, and PROCEDURE. It "defines" a module's interface to the outside world (to other modules), providing enough information for the compiler to check type consistency and for the programmer to use the module.

The implementation part contains the code that does the work.

For example, you might define MODULE Buffer which puts and gets

numbers to and from a last-in, first-out queue (a LIFO or stack) —

```
DEFINITION MODULE Buffer;
  VAR notempty, notfull: BOOLEAN;
  PROCEDURE put(x: CARDINAL);
  PROCEDURE get(VAR x: CARDINAL);
END Buffer.
```

Then implement it —

```
IMPLEMENTATION MODULE Buffer;
CONST N= 100 {stack's maximum size}
VAR in,out:[0..N-1];
    n:[0..N];
    buf: ARRAY[0..N-1] OF Cardinal;
PROCEDURE put(x: CARDINAL);
BEGIN
  IF n < N THEN
    buf[n]:= x;
    n:= n + 1;
    notfull:= n < N;
    notempty:= TRUE
  END
END put;

PROCEDURE get(VAR x: CARDINAL);
BEGIN
  IF n > 0 THEN
    n:= n-1;
    x:= buf[n];
    notempty:= n > 0;
    notfull:= TRUE
  END
END get;

BEGIN
  n:=0;
  notempty:= FALSE;
  notfull:= TRUE;
END Buffer.
```

Separate Compiling

The definition and implementation parts of the module are compiled separately. So you can change code in the implementation without recompiling the definition or any of the other modules which are already using it.

If you want to change the buffer in the above listing to a first-in, first-out queue (to be used as a keyboard buffer), alter the implementation part and leave the definition part alone.

Since the definition of the module hasn't changed, and the definition is all the outside world sees, those modules using the buffer won't know about the change in implementation from LIFO to FIFO (hope they don't need to). Consequently, the importing modules won't need to be recompiled. (Of course, everything has to be re-linked after you recompile the implementation part of the module.)

The new implementation (first-in, first-out) looks like this —

```
IMPLEMENTATION MODULE Buffer;
CONST N= 100;
VAR in,out:[0..N-1];
    n:[0..N];
    buf: ARRAY[0..N-1] OF CARDINAL;
PROCEDURE put(x: CARDINAL);
BEGIN
  IF n < N THEN
    buf[in]:= x;
    in:=(in + 1) MOD N;
    n:= n + 1;
    notfull:= n < N;
    notempty:= TRUE
  END
END put;

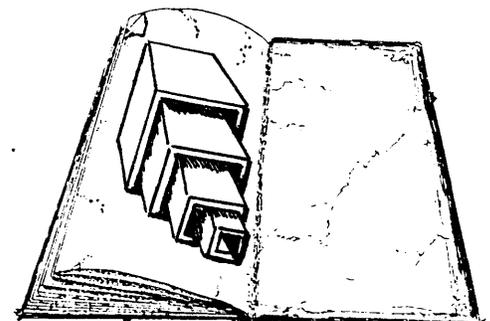
PROCEDURE get(VAR x: CARDINAL);
BEGIN
  IF n > 0 THEN
    x:= buf[out];
    out:=(out + 1) MOD N;
    n:= n-1;
    notempty:= n > 0;
    notfull:= TRUE
  END
END get;

BEGIN
  n:=0; in:= 0; out:= 0;
  notempty:= FALSE;
  notfull:= TRUE
END Buffer.
```

Big Programming: Trust Or Consequences

Assuming you don't write all the implementation code yourself (and in big programming projects, where Modula is most useful, you won't!), there's more than a few quanta of faith at stake. Your implementation cronies had better be implementing the module you're expecting from the definition, or everyone's in big trouble.

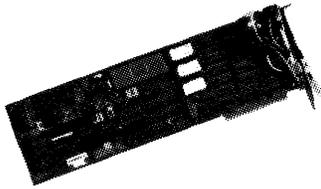
■ ■ ■



PLUG INTO SAVINGS! FOR YOUR I.B.M.

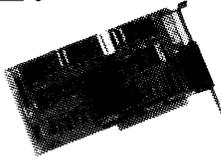
DataMac 5.25" hard disk controller card

\$99⁹⁵



DataMac Dual Async Card (card only) comes with one active port, you add the second 8250 and connector, with documentation

\$299⁹⁵



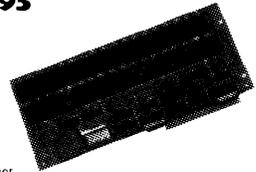
DataMac Dual Async card with 64K of memory expandable to 256K, you add the second 8250 and connector

\$599⁹⁵



DataMac 256K memory expansion (card only)

\$199⁹⁵



with purchase of card get sets of nine 4164-15 for only **\$69⁹⁵** set

purchase memory only for **\$199⁹⁵** set of nine

RODIME 20 MEG 5.25" HARD DISK

These drives are brand new in factory cartons, full size with documentation

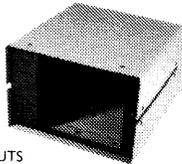
\$299⁹⁵

(similar to photo)



THE SENSIBLE 5 1/4" DISK DRIVE CABINET KIT

- FUSED POWER SUPPLY
5A @ 5V 4.8A @ 12V
- E.M.I. FILTER, 6 AMP
- SURGE PROTECTION
- POWER SWITCH
- POWER SUPPLY SHIELD
- CONNECTOR AND FAN CUTOUTS



CONSTRUCTED OF HEAVY GAUGE ALUMINUM, THIS UTILITY DISK DRIVE CASE WILL ACCOMMODATE: 5 1/4" HARD DISKS, FLOPPY DRIVES, HALF HEIGHTS, FULL HEIGHTS, 2/3 HEIGHTS, CONTROLLER CARD AND SINGLE BOARD COMPUTER. OVERALL DIMENSIONS 10 1/4" x 10" x 7" COLOR IS TAN

ONLY **\$119⁹⁵**

LESS POWER SUPPLY **\$79⁹⁵**

BUSINESS TELEPHONES

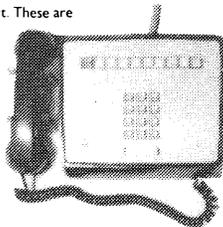
STOP PAYING MONTHLY PHONE CHARGES

These phones are not a cheap import. These are the same reliable I.T.T. phones that have served us both for many years.

Standard 10 button business telephones serviced by Bay Telephone Co.

ONLY **\$79⁹⁵**

limited to color selection on hand



BRAND NEW 6 BUTTON BUSINESS TELEPHONES ALL THE STANDARD FEATURES INCLUDING BUZZERS AND RINGERS. LOOKS THE SAME AS ABOVE PICTURE. COLOR, CHOC. BROWN.

\$59⁹⁵

ONLY (LIMITED QUANTITY)

GO ON LINE WITH UNITED PRODUCTS

Take advantage of the many items that space does not allow us to advertise. Use your computer to download our weekly specials and hard to find items.

On line 24 hours a day, 7 days a week.

300 or 1200 baud, even parity, 8 bits with one stop bit.

The phone number is: (206) 682-5593

New items are added weekly. Don't miss out, call each week.

CONTROLLER CARDS



D.T.C. 1403-D (shown with optional software and adaptor) control two 8" 851 DSDD and two 8" 1000 type

\$249⁹⁵

D.T.C. 520-1 control four 5.25" DSDD and two ST-506 type hard disks

\$99⁹⁵

HOST ADAPTER FOR ABOVE CONTROLLERS (for most popular Z-80 computers) The Winif Adapter and software

\$599⁹⁵

BUILDING A ROBOT? WE CAN HELP!

12 VDC STEPPER MOTORS

STANDARD 6 LEAD 4 PHASE
STEP ANGLE: 7.5°
RESISTANCE PER PHASE: 36 OHM
SHAFT: 1/4" x 3/8"
O.D. 1 1/4" x 2 1/4" (3/8" FLANGE)

REG PRICE \$32.00

\$99⁹⁵ EA.

\$69⁹⁵ EA. WITH PURCHASE OF ANY STEPPER DRIVER KIT

UNIVERSAL STEPPER MOTOR DRIVER KITS



FOR DIRECT CONTROL FROM YOUR COMPUTER

- Control almost any stepper motor, forward/REVERSE and step sequence.
- For stepper motors rated 1.7 VDC — 12.0 VDC
- Uses a power supply with an output of between 3.0 VDC and 12.0 VDC
- Modifications included for motors rated above 12.0 VDC
- Inputs/outputs are on a standard 22 pin edge card connector (.156 spacing) 4 1/2 x 6 1/2 card, G-10 epoxy.

TECHNA-KIT
cat# U.S.M.D.-C

\$299⁹⁵

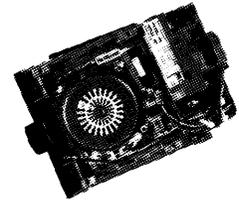
SAME SPECS. AS U.S.M.D.-C

- except a stand-alone driver
- On board clock (user optional)
- No edge connector

TECHNA-KIT
cat.# U.S.M.D.

\$199⁹⁵

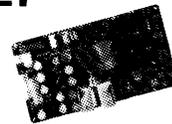
3 1/2" FLOPPY DRIVES SHUGART SA300



\$99⁹⁵ EA

- BRAND NEW
- 500K
- 80 TRACK
- 135 T.P.I.
- THESE DRIVES GIVE YOU THE SUPER RELIABILITY OF A HARD DISK. THEY ARE COMPATIBLE WITH ANY COMPUTER WHICH USES A 1770, 1771, 1791 OR 1793 CONTROLLER CHIP
- THEY WILL FORMAT SINGLE, DOUBLE OR QUAD DENSITY, DEPENDING ON YOUR SOFTWARE
- XEROX 820 (LITTLE AND BIG BOARD), KAPRO, AMPRO AND LOOK ALIKES AND H.P. SINGLE SIDED ARE JUST A FEW OF THE COMPUTERS THIS DRIVE CAN BE COMPATIBLE WITH

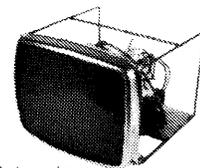
SWITCHING POWER SUPPLY



Real Power From a Famous Manufacturer delivers: 10 amps at 5V, 8 amps at 12V, measures only 8.5" by 4.5" and is only 2.25" thick

with pinouts **\$39⁹⁵** (limited supply on hand)

MONITORS



Ball Brothers TV-120 (shown above) 16mhz, Xerox 820 compatible, P-4 phosphor B.W. 12"

\$59⁹⁵

Ball Brothers TX-7 Xerox 820 compatible, 7", flat green screen w antiglare for photography

\$299⁹⁵

AudioTronics 964-02 Xerox 820 compatible, TTL X-Y input, 9" green screen

\$399⁹⁵



united products, inc.
serving the west coast since 1968

1123 Valley
Seattle, WA 98109

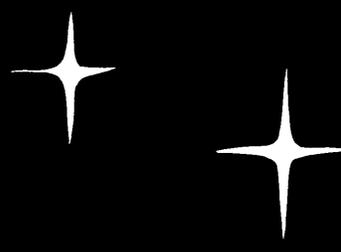
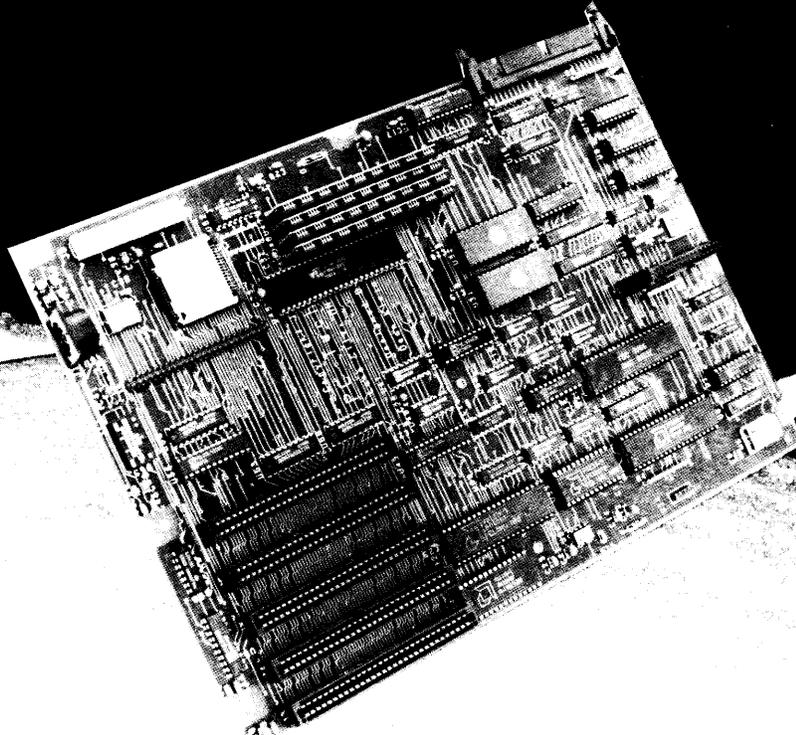
Hours: 9-6
Sat: 9-5

PHONE (206) 682-5025

Terms: All orders are shipped F.O.B. Seattle, Wash.
C.O.D.'s are OK with cash or certified check
MasterCard and Visa accepted
On orders over \$50.00 in the continental U.S. we pay the freight.
All advertised items are subject to prior sale

**NEW
LOWER \$**

the X-16A gives you Out of This World Performance



PLUS PC COMPATABILITY!

Features:

- 8 MHz 80186 microprocessor with true 16-bit data bus.
- True PC compatibility with our own custom ROM BIOS and PC cardslots for the video of your choice.
- Standard 512K zero wait DRAM, 640K or 1M options.
- 8087 numeric coprocessor port.
- On-board floppy disk controller for up to four 5¼" drives.
- On-board SCSI hard disk controller port.

NEW

Designed, Manufactured and Serviced by:

PCTECH

P.O. Box 128
904 North 6th Street
Lake City, MN 55041
(612) 345-4555

COD, VISA, MASTER CARD, CERTIFIED FUNDS

Kits

**NEW LOWER
PRICES**

- Not So Bare Board \$ 250.00
- Quick Kit \$ 625.00
- Full Kit \$ 750.00
- Assembled and Tested \$ 850.00
- 128K Add-On \$ 80.00
- ½ MEG Add-On \$ 200.00
- XT Style, Flip Top Enclosure \$ 85.00
- 150 Watt Power Supply \$ 135.00
- 5¼" 48 TPI DS
½ Height Floppy Disk Drive \$ 125.00
- 8087-2 Coprocessor Board \$ Call

NEW

Total Solutions:

Assembled and Tested Systems

- Mono-Chrome System \$1,865.00
Includes: 640K, Power Supply, Enclosure, Keyboard, Monitor, Two Drives, and Printer Port.
- Color System Call
- Quantity Discounts Upon Request
- Call For Info on Add-Ons, Hard Disks, Etc.