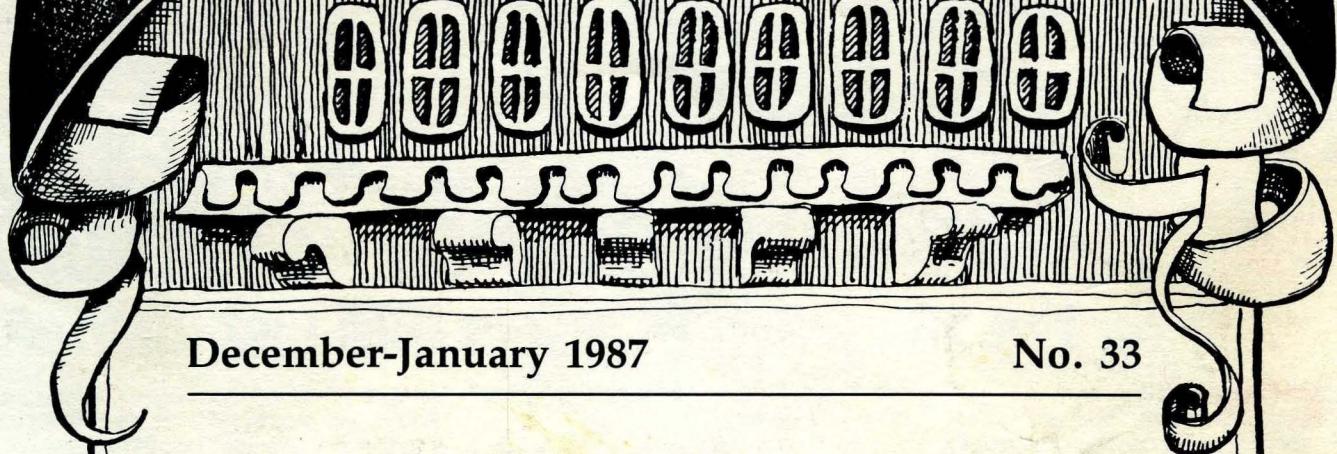


Down & Dirty

\$3.00

MICRO CORNUCOPIA

The
Micro Technical Journal



December-January 1987

No. 33

Low Level Projects

From Boots To Device Drivers: Secrets Of MS-DOS 6

A clean, in-depth look at making MS-DOS work for you.

Debugging A Disk 12

This companion to "Secrets of MS-DOS" uses DEBUG to poke around in MS-DOS.

Professionalize Your Programs Using Turbo Pascal's Built-in System Procedures 16

Turbo Pascal gives you assembly language style access to MS-DOS interrupts.

Controlling The Real World With Your PC: Intro To Electronics Step 2 22

This is the second part of Bruce's fine series. A great hardware primer.

PD-32: Messages From The Board 28

The Public Domain 32000 system is hot. The latest developments.

Selecting Subsets From Sets: The Formula For Combinations In Prolog 34

Real-life solutions to real-life problems.

Introduction To Fractals 36

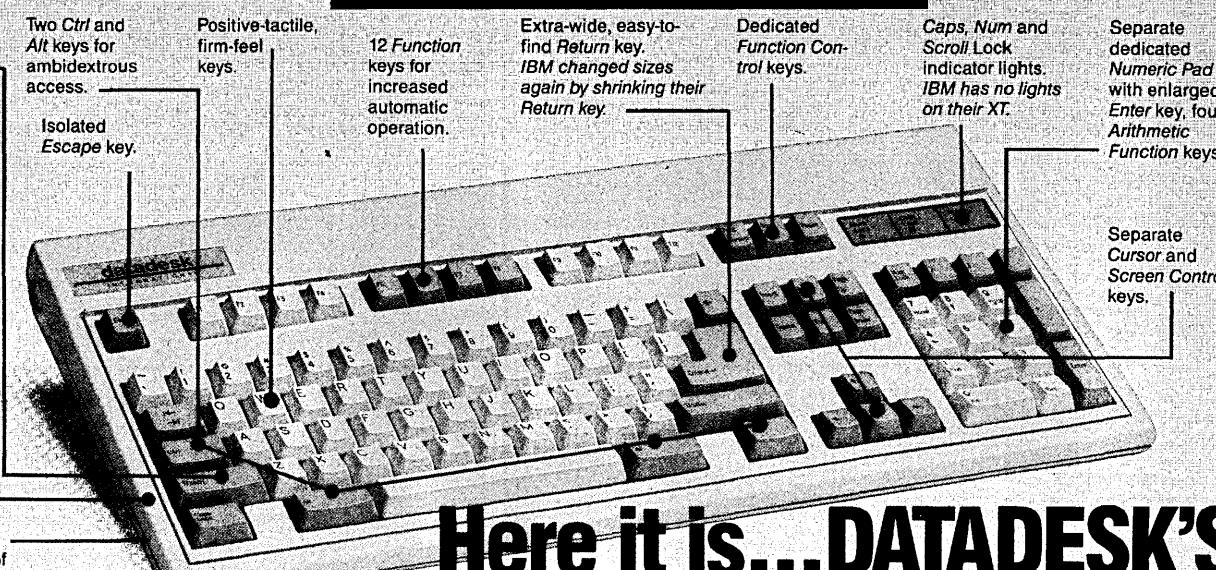
*If you haven't watched a fractal eat its way across
your screen, you've missed something.*



Borland's Turbo Lightning FREE

Shift keys
Switch-selectable compatibility with IBM, PC, XT, AT, PCjr, AT&T and all compatibles. IBM's new Enhanced Keyboard runs only on their new XT's, AT's and ignores their installed base which probably includes you.

Switch allows you to swap position of Ctrl and Caps lock. IBM moved Ctrl to bottom row, you have no choice!



Here it is...DATADESK'S **TURBO-101 ENHANCED KEYBOARD** for the 10 million PC users IBM just ignored!

IBM just announced their new redesigned "standard" keyboard for personal computers. There's only one problem: it won't work on your IBM computer if it was purchased prior to June 1986 or on any PC compatible purchased at any time!

Not to worry. Our new Turbo-101 Enhanced Keyboard gives you the layout and enhancements of the IBM with some logical improvements (see above photo). And it works on your existing PC, XT, AT, PCjr, AT&T, Epson and virtually all compatibles!

Get Borland's Turbo Lightning™ For FREE!

To really turbocharge your productivity, we are including, free-of-charge, Borland's red-hot Turbo Lightning software with each keyboard. Now, when using SideKick, WordPerfect, Microsoft Word, 1-2-3 or most popular programs, our Turbo-101 Keyboard will check your spelling as you type, gives you instant access to Random House's 80,000-word Concise Dictionary and 60,000-word Thesaurus and much, much more!

"Lightning's good enough to make programmers and users cheer, executives of other software companies weep," says Jim Seymour of PC Week. Sold separately, Turbo Lightning retails for \$99.95!

The Only Alternative

datadesk™
INTERNATIONAL

7650 Haskell Avenue
Van Nuys, California 91406 (818) 780-1673

Turbo-101 is a trademark of DataDesk International. Turbo Lightning is a trademark of Borland International. IBM and AT are registered trademarks of International Business Machines, Inc.

The Turbo-101 is the best data entry tool since the pencil!

For users of spreadsheets like 1-2-3, the Turbo-101's separate cursor controls and numeric keypad makes entering numeric data into cells and moving from cell to cell as natural as moving your fingers. And for word-processing, the 'Selectric' typewriter layout makes the Turbo-101 as easy to use as a pencil; and with the extra large Enter, Shift & Control Keys, you'll make so few mistakes, you won't even need an eraser!

SPECIAL OFFER!

ONLY \$149.95* FOR BOTH
KEYBOARD & SOFTWARE

Includes 30-day money back
guarantee and 2 year full warranty.

To prove that we don't ignore you or your pocketbook, you get our Turbo-101 Enhanced Keyboard and Borland's Turbo Lightning for an astounding \$149.95.* No, you didn't read it wrong. During this amazing Introductory Offer you get both keyboard and software for less than most software programs by themselves! Now, if you're still feeling ignored, you can always do what you-know-who wants you to do...and buy a new computer to get their keyboard!

credit card orders call
(800) 826-5398
in CA call
(800) 592-9602

*Price does not include adaptor cables required by certain compatibles. A limited offer—price subject to change without notification.



Up to now, DataDesk International may be one of the best kept secrets, but here's what's being said about our first end-user Keyboard/Borland software bundle:

"Who Can Pass Up a Deal? Department. Talk about an aggressive product!"

John C. Dvorak,
InfoWorld Mar 86

"It solves all of the problems exhibited by their regular PC/XT keyboard...it's a great bargain!"

PC Productivity Digest
May 86

"DataDesk Intl. has designed a sturdy and handsome keyboard that has tactile response...is the hardware bargain of the year" says Charles Humble, Oregonian Jan 86

"The best part of the keyboard is the way it feels. It's ideal! And fast. I've never worked on a keyboard with a nicer touch."

Business Computer Digest
Aug 86

"It's a good keyboard. Good feel: the keys have tactile feedback. No mush at all. This is about as good a keyboard deal as you're likely to find...I have absolutely no hesitation in recommending the Model PC8700."

Jerry Pournelle,
Byte Magazine Sept. 86

"This keyboard is neat to type on and feels solid. It has tactile feedback keys...I can type much faster on it." Test Drive Scorecard:
DataDesk-10 Key Tronics-9
Teleconnect Magazine
May 86

BOTH TURBO-101 ENHANCED™
KEYBOARD AND BORLAND'S TURBO
LIGHTNING™ SOFTWARE FOR ONLY:

\$149.95* LIMITED
OFFER

*Plus \$10 Shipping & Handling.
California Residents
Add \$2.75 Per Unit Sales Tax

NO. UNITS: _____
PAYMENT: VISA MC CHECK
AMOUNT ENCLOSED: \$_____

Enter ↗

NAME: _____
ADDRESS: _____
CITY: _____ STATE: _____ ZIP: _____
PHONE: _____ COMPUTER TYPE: _____
CC NO.: _____ EXP: _____

TAKE THE NEXT LOGICAL STEP

\$89 Price

- Separate Compilation
- Native Code Generation
- Large Memory Model Support
- Multitasking
- Powerful Debugging Tools
- Comprehensive Module Library
- Available for the PC and the VAX

Move up to LOGITECH MODULA-2/86. Whether

you're a single programmer or part of a team, with LOGITECH MODULA-2/86 you'll decrease your overall development cycle and produce more reliable, more maintainable code. Build your program using our extensive library modules, your own modules or those from a growing list of available third-party software vendors. If you're a Turbo Pascal user you can even take your existing code along with you with the help of our new Translator!

NEW & IMPROVED!

Turbo Pascal to Modula-2 Translator \$49

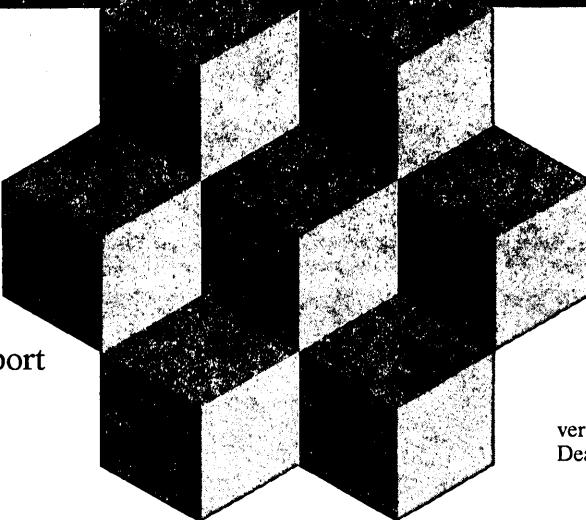
Now it's even easier for Turbo users to step up to Modula-2/86. Our improved Translator changes your Turbo source code into Modula-2/86 source, solving all the incompatibilities, and translating the function calls of Turbo into Modula-2/86 procedures. Implements the complete Turbo library!

LOGITECH MODULA-2/86 \$89

Complete with Editor, Run Time System, Linker, 8087 Software Emulation, Binary Coded Decimal (BCD) Module, Logitech's comprehensive library, Utility to generate standard .EXE files. AND more!

- LOGITECH MODULA-2/86 \$129
with 8087 Support
- LOGITECH MODULA 2/86 PLUS \$189
For machines with 512K of RAM. Takes advantage of larger memory to increase compilation speed by 50%.

Turbo Pascal is a registered trademark of Borland International.



**NEW,
improved Turbo
Pascal to Modula-2
Translator!**

Call for information about our VAX/VMS version, Site License, University Discounts, Dealer & Distributor pricing.

To place an order call our special toll free number:

800-231-7717

In California:

800-552-8885

MODULA-2/86

RUN TIME DEBUGGER \$69 (Source level!)

The ultimate professional's tool! Display source code, data, procedure call chain and raw memory. Set break points, assign values to variables, pinpoint bugs in your source.

UTILITIES PACKAGE \$49

Features a post-mortem debugger (PMD). If your program crashes at run time the PMD freezes the situation so you can pinpoint, in the source, the cause of the error and the status of the data. Also includes a disassembler, cross reference utility and version that allows conditional compilation.

LIBRARY SOURCES \$99

Source code for our major library modules is now available for customization or exemplification.

WINDOW PACKAGE \$49

Now you can build true windowing into your Modula-2 code. Powerful, though only 15K in size. Features virtual screens, color support, overlapping windows and a variety of borders.

MAKE UTILITY \$29

Automatically selects modules affected by code changes to minimize recompilation and relinking. Even figures out dependencies for you!

CROSS RUN TIME DEBUGGER AND ROM PACKAGE \$199

Now available at an introductory price!

YES

I want to move up to LOGITECH MODULA-2/86!

Here's the configuration I'd like:

<input type="checkbox"/> Logitech Modula-2/86	\$89
<input type="checkbox"/> with 8087 support	\$129
<input type="checkbox"/> Plus Package	\$189
<input type="checkbox"/> Turbo to Modula Translator	\$49
<input type="checkbox"/> Run Time Debugger	\$69
<input type="checkbox"/> Utilities Package	\$49
<input type="checkbox"/> Library Sources	\$99
<input type="checkbox"/> Window Package	\$49
<input type="checkbox"/> Make Utility	\$29
<input type="checkbox"/> ROM Package	\$199
Total Enclosed	\$_____

Visa Mastercard Check Enclosed

Card Number _____ Expiration Date _____

Signature _____

Name _____

Address _____

City, State _____

Zip _____

Phone _____



LOGITECH

Logitech, Inc.
805 Veterans Blvd.
Redwood City, CA 94063
Telephone 415-365-9852

For European pricing please contact:
Logitech S.A.
Box 32, CH-1143
Apples, Switzerland
Telephone 41-21-774545

MICRO CORNUCOPIA

The Micro Technical Journal

Editor & Publisher
David J. Thompson

Associate Editors
Gary Entsminger
Margret Rosenberg

Technical Department
Larry Fogg

Director of Advertising
Laura Logan

Director of Marketing & Graphics
Julie Pope

Accounting
Sandra Thompson

Circulation Manager
Tracey Anthony

Order Department
Cary Gatton Tammy Westfall

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
P.O. Box 223
Bend, Oregon 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.

RBBS - 24 hrs. 300-1200-2400 baud
503-382-7643

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

AROUND THE BEND

High Places

Up In The Air

I just got back from a quick trip to Florida, and the part I liked best was the plane ride.

It wasn't the scenery —

Sure, all those mountains, rivers, and deserts are fascinating and I haven't gotten tired of the plains with its checkerboard of 80-acre fields and 6-acre towns. I always ask for a window seat so I can make nose impressions on the plexiglass.

It was dusk as we departed Chicago so I watched as small towns were swallowed up in the deepening twilight. Soon, however, they reappeared below us, twinkling like reflected stars.

The travel was the best part because I had nothing to do but sit in my seat and nod as stewardesses came around with beverages, meals, magazines, and pillows.

There were no phone calls, no "Should we stay with 16,000 copies on this run?" or "Have you finished your talk for this evening?" or "Have you looked over the press release we just sent you?" (Someone's been hiring PR people away from the boiler rooms.)

It's nice to have an enforced rest once in a while.

Professional Travelers

If you spend much time in a plane you discover that professional travelers have an unwritten conduct code. "Pretend your neighbor isn't there."

All around me, the sports-coat and tie types wore the bored expressions of veterans. Great, it would be the dullest, quietest possible trip. A chance to put some thoughts together. After nodding for an hour (ginger ale, peanuts, and a pillow) some ideas started bubbling to the surface. I dug out my Kaypro 2000.

That was a mistake.

By the time I had extracted the 2000 from its case, a stewardess had asked me if it was a computer and the guy in the next seat was insisting that I tell him how a computer could help his business.

By the time I had the lid open and the system booted, I had the undivided attention of three attendants and the entire non-smoking section.

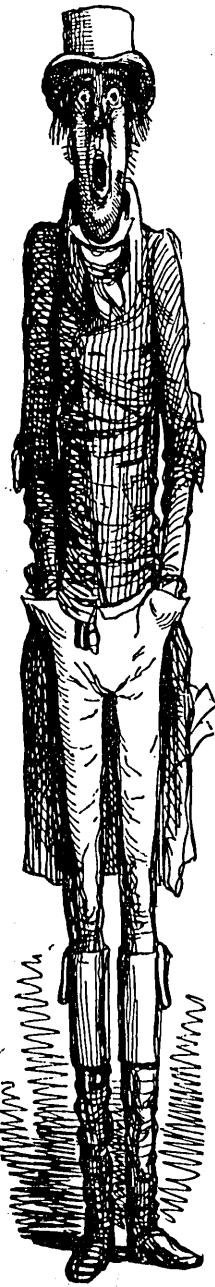
Doggedly I fired up the editor and tried to get my thoughts down before they were overwhelmed by audience participation.

It was impossible.

The fellow in the next seat was determined to read every word I entered (if you know anything about the 2000, you know where he had his head), and I'm sure he would have gladly read it aloud for everyone in the cabin.

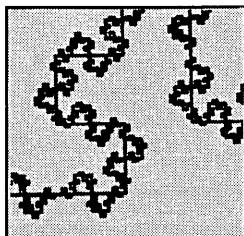
Hey folks, this ain't prose 'til I say it's prose.

(continued on page 91)



Features

- 6 Earl Hinrichs**
From Boots To Device Drivers: Secrets Of MS-DOS
From Earl's standing-room-only talk at SOG V. Poke around in the engine room with one of the best.
- 12 Earl Hinrichs**
Debugging A Disk
Uncomfortable with DEBUG? Read this companion article to "Secrets of MS-DOS."
- 16 Rick Ryall**
Professionalize Your Programs Using Turbo Pascal's Built-in System Procedures
Turbo gives you handles into the operating system that most folks never grab. Great fun.
- 22 Bruce Eckel**
Controlling the Real World With Your PC: Intro To Electronics Step 2
If schematics read like Martian poetry, prop up your feet, grab some chips, and let Bruce shed some light on hardware.
- 28 Larry Fogg**
PD-32: Messages From The Board
Last issue we published schematics on the world's cheapest UNIX system. This is some of the reaction.
- 34 Ramachandran Bharath**
Selecting Subsets From Sets: The Formula For Combinations In Prolog
Here's a look at combinations, recursion, and Prolog.
- 36 Larry Fogg**
Introduction To Fractals
What do you get when you cross graphics with mathematics? Graphematics? Nope. Fractals.
- 48 Larry Fogg**
The Bulletin Board Blues
Micro C's move to a clone-based bulletin board gave murphy a great chance to dabble. Join Larry as he arms himself with Pascal and does battle.
- 50 Gary Entsminger**
CONVERTing: Pattern Matching & List Processing For Turbo Pascal Programmers
CONVERT is an incredible language: incredibly powerful and incredibly opaque. This is Gary's gentle introduction.



- 64 David Thompson**
PC Pursuit: Cheap Communications
If those free boards aren't so free when you get your phone bill, you need the down-load on PC Pursuit.
- 74 Gary Entsminger**
Super Chips & Parallel Architectures: Designs For The Future
Afraid your shiny new turbo-clone is already outdated? Then you really shouldn't read this.
- 88 Andy Bakkers**
A 256K Upgrade For The Big Board II
You can't do just one 256K upgrade.

Columns

- 44 On Your Own**
Consulting from a consultant's point of view.
- 56 Kaypro**
7MHz upgrade for 83 Kaypros.
- 60 86 World**
Modula-2 and setting PC step rates.
- 66 C'ing Clearly**
Write your own printf().
- 70 Pascal Procedures**
John covers queues, stacks, and TSRs. Great stuff.
- 78 The S-100 Bus**
Controlling solid-state relays.
- 82 In The Public Domain**
The usual collection of exciting new software.
- 84 Technical Tips**
- 85 Culture Corner**

Future Tense

By Gary Entsminger

- 86 Tidbits**
Inside a supercomputer book.
- 96 The Last Page**
Version 1.1 of Turbo Prolog has many new features.

LETTERS

MS-DOS Isn't PC-DOS

Regarding "The Last Page" in issue #32: AARRGH!! (with a long, dwindling scream). Interrupt 16 (10H) DOES NOT EXIST in MS-DOS. The flavor known as "PC-DOS" for the IBM line of microcomputers has this interrupt. Let us remember that MS-DOS is not a synonym for the total IBM environment of microcomputer operating systems.

By using interrupt 16 you become hardware dependent on a machine which emulates PC-DOS and ROM BIOS (or is IBM).

Michael M. McDaniel
P.O. Box 1010
Beaverton, OR 97075

Program Corrections

I initially subscribed to your fine magazine on the recommendation of a friend. His statement was, "The programs work." I wish he had qualified that a little.

I direct your attention to Micro C, issue #32, page 66, "Recover A Directory By Reading & Writing Disk Sectors." The following bugs keep the program from working:

1. ERRMSG delimited with apostrophe and has apostrophe within the message. Neat trick. Bet it won't assemble correctly.

2. Probably typo ... WRITSEC routine has 2 consecutive JNC's ... second one probably should be JC rather than JNC.

And if I could be picky, I would like to quibble with these:

3. ASSUME directive for Code, Extra, and Data. Why Extra? Why NOT Stack?

4. MOV DH, MOV DL, MOV CH, MOV CL ... gads, doesn't he know how to MOV DX ... CX?

5. DISPLAY PROC ... ye gads. Has the man never heard of INT 21H, function 09?

To end this letter on a positive note, after correcting the apostrophe and JNC instruction, the program does what Gary says it will do. It copies a directory.

David E. Michener
Custom Computer Service
7466 S.E. 112th Ave.
Portland, OR 97266

New Hard Drive Products

Here is another hard drive manufacturer deserving of mention. And, a surprising new product you'll find closely related to that topic.

I've been using a Tulin TL-340 for several months now. I'm not exactly certain where this company came from. I was told Qume had something to do with it. It is on Qume Drive (cute, huh?) in San Jose. Somebody out there can check it out.

Anyway, Tulin has two note-worthy hard drives. The TL-326 (at 26.7 Meg unformatted) and the TL-340 (at 40 Meg unformatted). They both use a plated disk. RLL controllers generally require plated media.

Both drives utilize 640 cylinders, with 10 spare (640-649). Both recommend parking on cylinder 656. The TL-326 has 2 platters (4 heads), the TL-340 has three platters (6 heads).

Maximum access times are quoted at 95ms, about the same as the predecessor TL-226/TL-240. These drives are faster. The earlier ones would not work reliably on the AT. These do. Also, as before, average access times are quoted as 40ms. This time it may be true.

I've been using the Tulin in an 8MHz XT clone (rice rocket), running 1:1 interleave on an Omti 5510 controller. No trouble yet. Seems quick.

Tulin sells almost exclusively to the O.E.M. market. They seem reluctant to deal with little guys. My feelings on this score would make for 2 or 3 articles. Anyway, if you want to try one of their drives, and run into trouble, contact me and maybe we can work out something.

The TL-340 costs \$425 at 1ea. The TL-326 is around \$300.

Hard Disk Controller

There is a company in Tempe, AZ, you will be hearing more about. It is called the Konan Corporation. They seem like nice people. They have come up with a hard disk controller that is a little unusual. You may find several interesting features. Patent pending.

- * Compression algorithm — transparent within the controller, to squeeze data by as much as EIGHT TIMES! Typical is 1.5 to 2x.

- * Significantly greater error correc-

tion ability — 64K. Yes!

- * Disk caching — storing data that's accessed repeatedly in RAM.

- * Fragmentation control — storing as much data contiguously as possible.

- * Works with system clock speeds up to 16MHz. Yet fails at 1:1. Why???

When you get down to the specifics, things get fuzzy. You may or may not hit some bug. They are doing so much with the program in the EPROM. But maybe all the little details haven't been worked out yet.

But if you ever did a low level format (as in 'g=c800:5 or 6') this controller has more surprises for you. Menu driven, FDISK is built in, you only need to add FORMAT, from your DOS. Such Menus! You can configure everything.

There's more. When you first install this board, IT WILL BOOT! Not the hard drive. I mean the BOARD! It comes in as your B: drive, and has programs on it to initialize and set up the hard drive. You will then set a DOS partition to boot from. You will also set up one or more non-DOS partitions as 'E-Disks'. Useful in breaking up a large drive into sub-DOS-limited sizes. The 'E' disk (yes, can be 'd:', etc.) is the partition where compressed files get stored.

This controller is NOT an RLL. It works at the standard 5-Megabit rate, and does not require you to replace your present drive. Now, you can get 30 Meg out of a 10 to 20, and by the time you read this, probably exceed the DOS 32 Meg limit without having to install a driver. You need DOS 3.0 or later. Something coming out of that EPROM, no doubt.

You probably think 'this has got to be expensive'. Nope. The cost to dealers is currently \$125, one off. They think 10,000 controllers a month would be nice. I think they could sell ten times as many.

So who is Konan? Where'd they come from? What else do they make?

Konan was started in 1978 by Dave Baughman and Dave Evoy as a disk controller company. Initially, they made controllers for the S-100 bus. Among them, they built for ST-506 interface, SMD interface, and floppy controllers (5-1/4 & 8). A couple years ago, they designed a controller for the

ATTENTION MODEM OWNERS:

Introducing:



Easiest MODEM PROGRAM
in the world to use !

- Menu and F-key driven
- Cursor/first-letter selected
- Completely "help" lined
- STATUS graphic at all times

only \$49.95 + \$2 SH

See your software dealer or

Order Direct (805)497-8054

CalSOFT
3609 Thous. Oaks Blvd-Ste 121
Westlake Village, CA 91362



POWERFUL TIME-SAVING FEATURES

- "Batch" file transfer
- Most powerful Auto-Entry Strings (script macros) anywhere !
- Standard XMODEM and XMODEM(crc)
- Unattended file transfer
- 1200/300 baud
- IBM/Hayes compatibles



The SLR SuperLinker Plus is 3 - 10 times faster than any other linker, and look at these features:

- link a full 64K output (COM, HEX, SPR or PRL)
- works with Microsoft, Fortran, Basic, Cobol
- supports 32 character externals (SLR format)
- full drive/user support with alternate DU search
- supports 8 address spaces
- fill uninitialized spaces with 0 or FF
- global cross reference
- DSD80/SID compatible .SYM file
- manual overlays
- load map

requires Z80 CP/M 2.2 or greater 32K TPA

\$195

SLR Systems

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

PC bus. That was the KDC-230.

With what appear to be mostly minor outward changes to that board, they now also have the KXP-230. The boards are only 6 inches long, due to the extensive use of VLSI and surface mount technology. There is an 84 pin device, 256K and 64K EPROMs, an Omti 24 pin device, and a Z-80 processor — plus about 22 other TTL (mostly LS) chips. The board can handle two hard drives. Worth noting, there are no jumpers soldered on to correct 'mistakes'.

Removal of a jumper between pins on the board switches the base address from C800 to D800, in case you have more than one controller.

The older KDC-230 controller sells for \$70. With some minor foil cuts to correct the interrupt address and a Version 1.11 EPROM, that controller will work in the Tandy 1000.

Konan has done something else I can appreciate. They do not assemble their own boards. That means we are not paying for the equipment needed in that process. They send them out for stuffing and soldering.

If you install one of these boards, you have a piece built by DEC in your

machine. It is engineered by Konan, yes. Run off during otherwise idle time for the equipment. If it keeps my cost down, I like it.

For more information:

Tulin Corporation
2393 Qume Drive
San Jose, California 95131
(408)942-9025

Konan Corporation
Mike Lehrer, VP-Sales
4720 South Ash Avenue
Tempe, Arizona 85282
(602)345-1300

If you have questions for me, I can be reached, days, at:

Tucson Computer Products
Bill Hemmings, hacker interface
2465 North Alvernon
Tucson, AZ 85712
(602)881-8138

More About Newport Components

In issue #31 you commented adversely on the 1.2 meg drive from

Newport Components. I purchased one from Weltech about a month ago and, though I have not exercised it heavily, it seems to work OK.

I use it to interchange data between an IBM AT at the office and my XT clone at home. On several occasions I have used the full capacity of the disks. Upon reading your article, I rechecked some of those disks and found them to be readable.

Perhaps they have improved the drive since your review.

Charles F. Campen, Jr.
234 San Antonio Road
Arcadia, CA 91006

Editor's note:

It would be great if they have. But in the meantime, I think I'll try one of those little 720K 3 1/2" drives supported by MS-DOS 3.2. That is if Microsoft can get the bugs out of 3.2.

I just bought a copy of 3.2 and now I'm afraid to use it. DISKCOPY doesn't seem to want to work between two 360K drives when the target disk is unformatted. I like to copy disks.

(continued on page 80)

From Boots To Device Drivers

Secrets Of MS-DOS

The X16's BIOS has a reputation for being one of the fastest, cleanest BIOS's around, so we were pleased as punch to have Earl, the X16's BIOS writer, take us into the heart of MS-DOS at SOG V.

If you missed his talk (or the discussion of operating systems in the software forum), didn't make it to Bend, couldn't find the room, or didn't take notes, here's his written version — a blow by blow account of what's happening as MS-DOS gets ready to run your programs and peripherals. There's a lot here for beginners and professionals alike.

Let's take a close look at MS-DOS. Let's look at what Microsoft has implemented and, in the process, see if there isn't a hint of what it plans to do. For convenience, we'll divide the MS-DOS operating system into four parts:

Boot
BIOS
DOS
Command processor

On a bootable floppy, the boot code is stored in the first sector (track 0, head 0). But on a hard disk, the first sector is the partition sector (which has the info to establish the DOS partition location); the boot code resides on the first sector of the DOS partition.

The BIOS and DOS reside in the hidden files, IBMIO.COM (or IO.SYS) and IBMDOS.COM (or MSDOS.SYS). The command processor is in COMMAND.COM, which isn't hidden and will appear in directories.

BIOS

The BIOS is the link between the central processing part of the comput-

er and its peripherals. It can be subdivided into —

BIOS (resident)
BIOS (initialization)
system initialization

Microsoft (the maker of MS-DOS) supplies the system initialization in the file, SYSINIT. The resident and initialization BIOS's are usually written by the computer manufacturer.

SYSINIT is attached to the BIOS written by the manufacturer to make the BIOS file, IBMIO.COM. Shortly after boot, the initialization part evaporates leaving only the resident part of the BIOS.

The resident part (of the BIOS) is a collection of device drivers, used when DOS needs to talk to some piece of hardware. I'll talk more about device drivers later.

IBMDOS.COM & COMMAND.COM

IBMDOS.COM (or "DOS," for short) is the heart and soul of MS-DOS and is identical throughout all implementations of the same versions (Compaq's = IBM's = PCTech's = Zenith's). It's a buffer between the program and the hardware.

Your programs can run on different systems (maintaining portability) if you let DOS handle the basic operating details. Since the part of DOS the program sees doesn't change between installations, software gains machine independence, at least in theory.

But you can bypass the operating system (for speed or more direct control) and program the hardware. Many programmers do it, losing that independence.

The part of DOS a user sees is the command processor, which generates the DOS prompt, reads and executes commands from the keyboard, and executes batch files. The command processor can be changed with the

DOS shell command, but it's almost universally called COMMAND.COM.

DOS Initialization

On most PCs, on power up or after system reset, the program in the system ROM(s) is executed. This program will initialize and checkout the system and try to boot a disk.

On the IBM PC, it tries to read the first sector of the floppy in drive A. If it can't, and there's a readable hard disk, it reads the first sector of the hard disk. Either way, it loads a boot sector into memory at 0000:7C00.

The boot sector will know how to read the rest of the system files to continue the boot. This is true for all operating systems (for example, UNIX), not just MS-DOS. Some program disks (copy-protected games, for example) have their own loader, or miniature operating system, on the first sector.

The boot sector loads the first directory sector of the boot disk into memory between 0050:0000 and 0050:01FF. It checks the directory to ensure that the first two directory entries are IBMIO.COM and IBMDOS.COM. It determines the length of IBMIO.COM (the BIOS) and then loads it. (PC-DOS (the IBM version) loads the BIOS at 0070:0000; other implementations may put it elsewhere.)

The BIOS must occupy contiguous sectors starting at the first data sector (the first sector after the directory). Presumably this makes writing the boot sector easier, but it has some drawbacks. For example, the DOS "SYS" command, which puts the system files on a disk, may give a "no room" error when a disk has lots of free space, but has it in the wrong places.

The BIOS initializes itself, sets up a pointer to the end of its resident part, and then jumps to SYSINIT. (The BIOS can adjust the end of BIOS

By Earl Hinrichs

PC Tech
P.O. Box 128
904 North 6th St
Lake City, MN 55041

pointer to reserve space for tables or buffers.)

SYSINIT then moves itself to the physical top of memory and loads IBMDOS.COM at the end of the BIOS.

The remaining initialization is done with calls to DOS from SYSINIT.

The order in which items are initialized is a delicate matter. For example, the DOS function call "interrupt", interrupt 21, and the disk device driver must be initialized before CONFIG.SYS can be read. But installable device drivers, disk buffers, etc. can't be initialized until after CONFIG.SYS is read. So how did Microsoft get around this dilemma?

SYSINIT does the following — initializes DOS interrupt vectors (interrupts 20H - 3FH) and BIOS device drivers, reads CONFIG.SYS, loads and initializes installable device drivers, and creates internal buffers & tables. The exact order (the tricky part) isn't clear, but it works. Finally, SYSINIT loads and executes COMMAND.COM.

COMMAND.COM

COMMAND.COM also has three divisions: resident, initialization, and transient.

The resident part (or code) sits in low memory, just above DOS. The initialization part is loaded immediately above the resident. And the transient part resides in the highest free memory.

SYSINIT transfers control to the initialization code, which loads and executes AUTOEXEC.BAT if it's present. Otherwise it asks for time and date, ending the initialization (and the need for the initialization code). So COMMAND.COM can and will load programs immediately after its resident part, overwriting the initialization code.

Programs are given control of all high memory, including the memory which contains the transient portion of

COMMAND.COM. Most won't need so much and will return to the resident part of COMMAND.COM with the transient code still intact. When the resident part of COMMAND.COM regains control (from a program), it checks to see if the transient code is still intact. If it's not, the transient code is reloaded.

Device Drivers

The BIOS is a collection of resident device drivers. You can extend it by adding installable device drivers through the file CONFIG.SYS.

The BIOS must contain the CON, PRN, AUX, & CLOCK device drivers and at least one disk device driver.

PC-DOS comes with the installable device drivers ANSI.SYS (ANSI terminal emulation), VDISK.SYS (memory or RAM disk), and DRIVER.SYS (special disk formats). At PC Tech we provide the installable device drivers EMB.DEV (for extended memory), MEMDISK.DEV (memory disk which uses extended memory), COMX.DEV (for extra serial ports), and HARDISK.DEV (for more than two hard drives). Other manufacturers provide other drivers. Slicer has a driver which lets you generate IBM key codes on a serial terminal. Most specialized hardware add-ons for the PC come with installable device drivers.

All device drivers contain a short table called the device header (See Figure 1). The device header consists of link, attribute, strategy, interrupt, and name fields. The link field of each device driver points to the next driver. The last driver has -1 in its link field.

Figure 1 - Device Header Table

Device Header (18 bytes)	
LINK	4 bytes
ATTRIBUTE	2 bytes
STRATEGY	2 bytes
INTERRUPT	2 bytes
NAME	8 bytes

The device header must be the first item in an installable device driver. You can have more than one device driver per file, and multiple device drivers will be linked via the link field. But generally, installable device drivers come one to a file. This allows you to more easily pick and choose devices to install in CONFIG.SYS.

During initialization, DOS fills in the segment values in the link fields and links drivers from different disk files.

Attributes And Bits

Each attribute field defines specific characteristics of the device. (See Figure 2.)

Device drivers are either character or block. Character drivers drive things which handle I/O a byte or two at a time — the keyboard, video display, modem, printer. Block drivers drive things which process I/O in large blocks or sectors — floppy, hard, and RAM disks, for examples. Attribute bit 15 is set (to 1) to indicate a character driver; it's 0 to indicate a block.

A character device must have a name placed in the device header's name field. Names for block devices are assigned by DOS when they're installed. DOS assigns A: to the first block device, B: to the second, and so on. Disk names can be changed with the ASSIGN and SUBST commands.

Often one block device driver may drive several devices. For example, the same driver can be used for all floppies. In block devices the first byte of the name field holds the number of logical units controlled by the driver. (In PC-DOS 3.xx all floppy and hard disks are controlled by the same device driver.)

The strategy and interrupt fields in the device header point to two entry points in the device driver. DOS will call the strategy routine with a pointer, ES:BX, to a command description in a

(continued next page)

(continued from page 7)

block of data called the request packet. The strategy entry point simply stores a pointer to the request packet. DOS will then call the interrupt entry point, which fetches the request packet and performs the indicated function.

Sometimes you need to do more than simply read or write data to a device driver. For example, you might want to set the baud rate on a serial port or ask about the physical characteristics of a disk.

Device drivers which handle control information like this have attribute bit 14 set. The meaning of the control strings is up to the device driver. Programs can pass control information to or from a device driver, by using the DOS IOCTL function call (44H).

One character device is designated the clock device and always handles I/O six bytes at a time. The bytes are, in order — days since Jan 1, 1980 (word), minutes, hours, 1/100 seconds, seconds. The clock device is identified by attribute bit 3.

Some character devices are designated as standard input (attribute bit 0), standard output (bit 1) and NUL device (bit 2). Normally these devices are built into the BIOS, but an installable device driver with one of these bits set overrides the setting in the BIOS. If you install ANSI.SYS, for example, it replaces CON as the standard output device.

You can designate one character device for special output (attribute bit 4).

Three new block device driver commands were introduced in DOS 3.00. Attribute bit 11 — called "the removable media bit" — is set in block devices that support the new commands.

Device drivers offer flexibility, but

create overhead. So sometimes we can improve performance by bypassing a device driver. We can generate fast one-character output, for example, by loading the character in AL and calling interrupt 29H. On the IBM PC, and most other MS-DOS implementations, the CON device is capable of this special output.

Media Descriptor & BPB Table

Some block devices, in particular floppy disks, will have to support several formats. DOS has two ways to switch formats.

The first, a carry over from DOS 1.0, predefines several formats. The first byte of the second sector, called the FAT_ID, or media descriptor byte, selects one of the predefined formats. Support is limited to the predefined formats.

In the second, the boot sector contains a table, called the BPB (BIOS parameter block), which holds all the necessary format information. Any number of new formats can be read off the disk.

The first method is used by IBM, the second by most other MS-DOS implementors.

Attribute bit 13 is set if the driver is capable of supporting non-IBM formats. In the PC-DOS disk driver, this attribute bit is off, which means disk formats are determined by the media descriptor byte. However, if the driver doesn't recognize the media descriptor byte, it will try to read the BPB from the boot sector.

One more item, just to confuse things: Microsoft carefully distinguishes between media descriptor byte, and FAT_ID byte. The media descriptor byte is in the BPB table, and the FAT_ID byte is in the file allocation table. These two bytes don't have

to be the same, but in all IBM formats they are.

Request Packets

The first thirteen bytes of the request packet have the same meaning for all device driver commands. Only the definition and the number of remaining bytes vary from command to command.

The fixed fields are length, unit number, command, and status.

The length field is the length of the request packet in bytes. For block devices, where one driver may drive several devices, the unit field selects one of the devices.

Command Codes

The command code identifies the nature of the device driver request. Some commands apply to all devices; some only to character devices or only to block devices. Some other commands are only applicable if certain attribute bits are set. The status field is filled in by the device driver and returned to DOS, reporting the command's success or failure.

The done bit (#8) is used when the operation is complete. The busy bit (#9) is used by the non-destructive input command and the two status commands. All other commands clear the busy bit before returning.

The error bit (#15) is set when an error occurs, indicating an error in the lower 8 bits.

The BPB

The init command is DOS's first call to the device driver. It's called once. The variable portion of the init command has fields for number of units, ending address, BPB array pointer, and drive number.

For block devices, the drive number field has the name DOS will assign to the first unit in the device (0=A, 1=B, etc.). For an installable device driver the BPB array pointer points to the string following the "=" in CONFIG.SYS. This pointer can be used to pass parameters to the device driver.

The driver must handle any required initialization, then return a pointer to the end of the device driver in the ending address field. Typically the initialization code is put at the end of the device driver; then a pointer to the

Figure 2 - Attribute Fields

ATTRIBUTE

bit 15	1 = character device, 0 = block device (disk)
bit 14	1 = supports IOCTL, 0 = does not support IOCTL (block devices)
bit 13	1 = non-IBM format, 0 = IBM format
bit 11	1 = supports removable media
bits 5-10,12	not used, or secret meanings.
bit 4	(character device) 1 = special output (int 29)
bit 3	(character device) 1 = current clock device
bit 2	(character device) 1 = current NUL device
bit 1	(character device) 1 = current standard output
bit 0	(character device) 1 = current standard input

start of the initialization code is returned. This way the memory occupied by the initialization code can be freed after it's executed.

A block device driver must generate a BPB table for each device it controls. A BPB describes to DOS the format of the device, and includes, among other things, the sector size, device size in sectors, and directory size. The device driver also builds up a table of pointers to the BPB tables, one for each unit.

A pointer to this table of pointers is returned to DOS in the BPB array field. (Remembering that the BPB array pointer is a pointer to a table of pointers can save days of debugging; this is experience talking.) Block devices must also fill in the number of units field.

The BPB may change later, through the BUILD BPB command.

Details

DOS searches the initial tables to find the largest sector size. This becomes the size of the DOS sector buffers. A bug in DOS 2.x would only check the block devices in the BIOS for sector size; the installable devices were not checked. Thus, an installable device may not have a sector size larger than the largest built-in block device. (I don't know whether this has been fixed in 3.1.)

In PC-DOS the largest sector is 512 bytes. 16 bits are used to pass sector numbers to a device driver. So block devices are limited to 64K sectors, or to $1/2 \text{ K} * 64 \text{ K} = 32 \text{ Megabytes}$ overall.

Media Check

DOS uses the media check command to determine whether a disk has been changed. The variable portion of the request packet has fields for media descriptor byte, change code, and volume ID. The current media descriptor byte is passed to the driver (which may help determine whether the disk has been changed).

The driver returns disk change information in the change code field. -1 indicates the media has been changed; +1 indicates that it hasn't; 0 is returned when the driver isn't sure.

A driver for a hard disk (not the removable kind) or memory disk will

always return +1, no change. A driver for a typical floppy will usually return 0, not sure. Some floppy disk drives can detect an open door, and so would return -1 if the door has been opened, and +1 otherwise.

There are some tricks the device driver can use in the absence of a door switch. It may keep track of the media descriptor byte, volume ID, directory checksum, or other things to help detect disk changes. Often the floppy driver will keep track of the time since the last disk access. If it's been a short time, say less than two seconds, it returns +1, no change. If the driver detects that the media has been changed, and the removable media attribute bit is set, then the driver also returns a pointer to the previous volume name.

Media Check To BUILD BPB

If the media check command indicates media change, or if the media check command returns not sure and it's ok to change disks, then DOS will call the BUILD BPB function. (It's ok to change disks when there are no dirty buffers — altered buffers which have not yet been written.)

The BUILD BPB function must somehow determine the format of the disk, create a BPB table, and return a pointer to the table. The variable portion of the request packet consists of fields for media descriptor, transfer address, and BPB table pointer. If the device driver has the non-IBM format bit clear, then the transfer address will point to a buffer which holds the second disk sector.

Recall that the first byte of this sector is used to determine the disk format. If the non-IBM format bit is set then the transfer address pointer points to a one sector size scratch area. Generally this area is used to read in the boot sector of the disk, and extract the BPB table from the boot sector.

The distinction between IBM formats and non-IBM formats is no longer very important. Disks formatted with PC-DOS 3.xx have a copy of their BPB table in the boot sector. Thus disks formatted on an IBM can be used on non-IBM machines.

If PC-DOS 3.xx does not recognize the media descriptor byte on a disk, it will look in the boot sector for the BPB

table. Hence PC-DOS 3.xx can read disk formats from non-IBM machines. The variable part of the request packet for input commands consists of fields for media descriptor, transfer address, transfer count, sector number, and volume ID.

The media descriptor, sector number, and volume ID fields are not used for character devices. A character device will read the number of bytes set by the transfer count field into a buffer pointed to by the transfer address field.

Block devices also transfer data to the buffer pointed to by the transfer address field, but now the transfer count is a number of sectors, not bytes. The sector number field gives the starting sector.

A block device with its removable media attribute bit (11) set will know how many files are currently opened on the disk. If it detects a disk change while there are open files, it will abort the read and return an invalid disk change error. It will then set the volume ID field to point to the volume name of the disk that should be in the drive. DOS can then ask the user to put the right disk in the drive.

The IOCTL call is for reading control information.

Output Commands

The output commands are similar to the input command. For character devices, the transfer address points to a buffer of transfer count characters which are sent to the device. For a block device, the transfer count sectors, from the buffer at transfer address, are written starting at the sector number.

Block devices with attribute bit 11 set, the removable media bit, will check for illegal disk changes, and return the volume name of the removed disk. For block drivers, the output with verify command usually reads the data back to check for a successful write. For most character devices, the output with verify function is the same as the normal output function. The IOCTL call is for sending control information to the device.

(continued next page)

(continued from page 9)

Non-destructive Input

The non-destructive input function has one field in the variable portion of the request header — the character field. The non-destructive input function is used to peek at the input buffer. The next character in the input buffer is returned in the character field.

The character isn't removed from the input buffer. A call to the input function must be made to remove the character from the buffer. If the input buffer is empty, then the busy bit in the status word is set. The non-destructive input function does not wait for a character, but the busy bit indicates whether a call to the input function would wait.

DOS uses this function to check for ^C from the keyboard. There is no waiting if the keyboard buffer is empty, and as long as ^C is not the first character, the type ahead buffer is not disturbed. This function is also used to stop console output when ^S is typed.

Because this function does not remove characters from the input buffer, ^S will work only when the type ahead buffer is empty. If some other character is at the head of the buffer, DOS never sees the ^S.

The variable portion of the status calls request packet is empty. The status calls set the busy bit in the status word if a call to the input or output functions would wait.

The flush calls tell the device driver to clear the buffers. Some things are too important (Do you want to format your hard disk (Y/N)?), or too unexpected (Abort, Retry, Ignore) to trust to the type ahead buffer. In these and similar situations, DOS will call the input flush function before looking for a response.

Block devices which have the removable media attribute bit set must know how many files are open on the disk for the input and output commands. DOS will call device open or device close whenever a file is opened or closed. This way the driver can keep a running count of the number of open files. Character devices can use the device open and device close calls to perform special functions. For instance, form feeds may be sent to a

printer.

The driver sets the busy bit in the status word if the device has removable media. DOS uses this call to decide how strongly to word the warning message before formatting the disk.

Speculative Wrap Up

The double entry points, strategy and interrupt, in device drivers seem strange. Wouldn't just one call be more efficient? The recommended strategy routine is simply —

```
MOV REQUEST_PACKET, BX  
MOV REQUEST_PACKET[2], ES  
RET
```

The interrupt routine usually contains —

```
MOV BX, REQUEST_PACKET  
MOV ES, REQUEST_PACKET[2]
```

So, the double calls seem a waste. DOS guarantees that no other device driver calls will be made between the call to the strategy routine and the call to the interrupt routine. Therefore the reason for having two calls is not to get something done in between the calls.

What about the 'done' bit in the status word? This bit must be set whenever the interrupt routine returns. Doesn't a return from the interrupt routine imply that the command is completed? Also, what are those eight undocumented bytes doing in the request packet?

Let's speculate. Suppose that strategy and interrupt calls aren't tied together one for one. This would be reasonable in an operating system which wasn't so concerned about doing things sequentially.

But the strategy routine had better be re-entrant. Storing a pointer in a fixed memory location won't do.

Four of the eight unused bytes in the request packet could be used to make a nice linked chain of requests. Suppose the interrupt routine is called with several requests queued. The interrupt routine could look over the requests, and select one or several to start doing. Now we have a reason to use the done bit. Only the completed requests will have the done bit set on return.

Frequently the interrupt routine will have to wait for some I/O event. A good interrupt routine wouldn't wait.

It would start the task, then return with the busy bit set and the done bit clear.

Perhaps our speculative operating system could accomplish something useful while waiting for a disk drive to seek, modem to receive a character, or a human to type a key. Would it be too poetic, or too coincidental, for an interrupt to signal to the operating system when to return to the interrupt routine?

Is Microsoft preparing a multi-tasking version of MS-DOS, or am I dreaming?



TURBO COMMUNICATION WITH MULTI-TASKING

- Communicate with mainframe or micros
- Mult-tasking (run up to 32 programs, all at the same time.) Compatible with most DOS software, including LOTUS, dBASE, WORDSTAR, etc.
- INTERRUPT - DRIVEN buffered communication capabilities.
- X-MODEM file transfer
- ASCII file transfer
- Auto answer/Auto dial (Hayes compatible)
- Menu driven
- Communicate at speed up to 9600 bps.
- Written in TURBO PASCAL.
- Documented source code included for communication program.
- Manual (which teaches the fundamentals of telecommunications.)
- Manual (with source code documented and xreferenced.)
- For IBM-PC/XT and most compatibles.

Complete Package: \$29.95
CALL LARRY AT
(312) 778-3146

LARRY CHRYSIKOS
6500 S. WASHTENAW
CHICAGO, ILLINOIS 60629

Figure 3 - Device Driver Details

MEDIA DESCRIPTOR BYTE

byte	sides	sectors	tracks
FFH	2	8	40
FEH	1	8	40
FDH	2	9	40
FCH	1	9	40
F9H	2	15	80
F8H	Fixed disk		

REQUEST HEADER FORMAT

Length	1 byte
Unit number	1 byte
Command	1 byte
Status	2 bytes
Reserved	8 bytes
Command specific data	variable length

REQUEST HEADER DETAILS

Device Driver Command Codes

- 0 Init (all devices)
- 1 Media Check (block devices)
- 2 Build BPB (block devices)
- 3 IOCTL Input (if attribute bit 14 set)
- 4 Input (all devices)
- 5 Non-Destructive Input (character devices)
- 6 Input Status (character devices)
- 7 Input Flush (character devices)
- 8 Output (all devices)
- 9 Output with Verify
- 10 Output Status (character devices)
- 11 Output Flush (character devices)
- 12 IOCTL Output (if attribute bit 14 set)
- 13 Device Open (if attribute bit 11 set)
- 14 Device Close (if attribute bit 11 set)
- 15 Removable Media (if attribute bit 11 set)

Status word

bit 15 set if error occurred
 bit 9 set if device is busy
 bit 8 set if operation is complete
 bits 7-0 error code

Error codes

- 0 Write protected
- 1 Unknown unit
- 2 Not ready
- 3 Bad command
- 4 CRC error
- 5 Bad drive request structure length
- 6 Seek error
- 7 Unknown media
- 8 Sector not found
- 9 Out of paper
- A Write fault
- B Read fault
- C General Failure
- F Invalid disk change

Init (command #0, all devices)

HEADER 13 bytes
 NUM_UNITS 1 byte
 ENDING_ADDRESS 4 bytes
 BPB_ARRAY 4 bytes
 DRIVE_NUMBER 1 byte

Media Check (command #1, block devices only)

HEADER 13 bytes
 MEDIA_DESCRIPTOR 1 byte
 CHANGE_CODE 1 byte
 VOLUME_ID 4 bytes

Build BPB (command 2, block devices only)

HEADER 13 bytes
 MEDIA_DESCRIPTOR 1 byte
 TRANSFER_ADDRESS 4 bytes
 BPB_TABLE 4 bytes

IOCTL Input (command 3, only if attribute bit 14, IOCTL, set)
 Input (command 4, all devices)

HEADER 13 bytes
 MEDIA_DESCRIPTOR 1 byte
 TRANSFER_ADDRESS 4 bytes
 TRANSFER_COUNT 2 bytes
 SECTOR_NUMBER 2 bytes
 VOLUME_ID 4 bytes

Output (command 8, all devices)

Output with Verify (command 9, all devices)
 IOCTL Output (command 12, only if attribute bit 14, IOCTL, set)

HEADER 13 bytes
 MEDIA_DESCRIPTOR 1 byte
 TRANSFER_ADDRESS 4 bytes
 TRANSFER_COUNT 2 bytes
 SECTOR_NUMBER 2 bytes
 VOLUME_ID 4 bytes

Non-Destructive Input (command 5, character devices only)

HEADER 13 bytes
 CHARACTER 1 byte

Input Status (command 6, character devices only)
 Output Status (command 10, character devices only)

HEADER 13 bytes

Input Flush (command 7, character devices only)
 Output Flush (command 11, character devices only)

HEADER 13 bytes

Device Open (command 13, devices with attribute bit 11 set only)
 Device Close (command 14, devices with attribute bit 11 set only)

HEADER 13 bytes

Removable Media (command 15, block devices with bit 11 set only)

HEADER 13 bytes

Debugging A Disk

By Earl Hinrichs

PC Tech
P.O. Box 128
904 North 6th St
Lake City, MN 55041

This began as a sidebar (a short aside) for Earl's article on MS-DOS in this issue. However, it was too long to be a sidebar and too important to leave out.

It is important for two reasons. First, it seemed like everyone knew how to poke around on CP/M disks to find out what was going on 'inside' the disk. Not so many are comfortable with MS-DOS disks.

Second, Microsoft included DEBUG with MS-DOS, but you won't find it documented in the standard manual. I guess they figured most people wouldn't use it.

You don't have to believe everything you read about MS-DOS disks, and formats, and data, and allocations, and system tracks, and directories, because you can fire up your computer and find out for yourself.

Start off by formatting a floppy disk with the system option. I used drive B.

C>FORMAT B: /S

Insert new diskette for drive B:
and strike ENTER when ready

Format complete
System transferred

362496 bytes total disk space
69632 bytes used by system
292864 bytes available on disk

Format another (Y/N)?n

Notice 69632 bytes used by system.
Get a directory.

C>DIR B:

Volume in drive B has no label
Directory of B:\

COMMAND COM 23791 12-30-85 12:00p
1 File(s) 292864 bytes free

COMMAND.COM accounts for 23791 of the 69632 bytes used by the system. Where are the rest? DIR won't help us find the lost bytes. We need something stronger, so dig out DEBUG.

C>DEBUG

Use the DEBUG load command to read the directory. The syntax for the load command is 'L m d s n' where m = memory location, d = disk number, s = start sector, n = sector count. The directory starts on sector 5 on 360K disks.

-L 0 1 5 1

Use the dump command 'D' to see the directory in Figure 1. The number at the left, 128E, will be different when you try this. 128E is the first free segment on my system after I've loaded DOS, with a few installable device drivers and my favorite memory resident programs.

Two Hidden Files

Here we find two files not reported by DIR — IBMBIO.COM &

IBMDOS.COM. At offset 0B in each directory entry is the file attribute byte. 27 means read only, hidden, system file, archive. 20 means archive.

The archive bit is always set, unless some special backup program clears it. This disk was just created, so all files have the archive bit set.

Read only, hidden, and system bits set means hands off to DIR, ERASE, COPY, and most other commands. If you're curious about the rest of the directory, the 00 60 9E 0B is the time and date stamp. At offset 1A is the starting cluster, IBMBIO.COM starts at cluster 0002, IBMDOS at cluster 0012, and COMMAND.COM at 002E. For 360K disks a cluster is two sectors. The last four bytes are the file size. IBMBIO.COM is 00003FF1 bytes long, IBMDOS.COM is 00006F3D bytes long, and COMMAND.COM is 00005CEF bytes long.

If you add these sizes you'll come up with a number slightly smaller than the 69632 = 11000H reported by format. It takes 44H clusters to hold the three files. 44H clusters of 1K bytes each make 11000 bytes total.

Figure 1 - Directory Dump

-D 0

128E:0000	49 42 4D 42 49 4F 20 20-13 4F 4D 27 00 00 00 00	IBMBIO COM'....
128E:0010	00 00 00 00 00 00 00 00 60-9E 0B 02 00 F1 3F 00 00?..?..?
128E:0020	49 42 4D 44 4F 53 20 20-13 4F 4D 27 00 00 00 00	IBMDOS COM'....
128E:0030	00 00 00 00 00 00 00 00 60-9E 0B 12 00 3D 6F 00 00=0..
128E:0040	43 4F 4D 4D 41 4E 44 20-13 4F 4D 20 00 00 00 00	COMMAND COM
128E:0050	00 00 00 00 00 00 00 00 60-9E 0B 2E 00 EF 5C 00 00`....\..
128E:0060	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
128E:0070	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

Figure 2 - File Allocation Table

-L 0 1 1 1

-D 0

128E:0000	FD FF FF 03 40 00 05 60-00 07 80 00 09 A0 00 0Be..`.....
128E:0010	C0 00 0D E0 00 0F 00 01-11 F0 FF 13 40 01 15 60e..`..
128E:0020	01 17 80 01 19 A0 01 1B-C0 01 1D E0 01 1F 00 02
128E:0030	21 20 02 23 40 02 25 60-02 27 80 02 29 A0 02 2B	I ..#e..`...)+..
128E:0040	C0 02 2D F0 FF 2F 00 03-31 20 03 33 40 03 35 60	...-.1.36.5`
128E:0050	03 37 80 03 39 A0 03 3B-C0 03 3D E0 03 3F 00 04	.7..9..;=-=?.
128E:0060	41 20 04 43 40 04 45 F0-FF 00 00 00 00 00 00 00 00	A .CE.E.....
128E:0070	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

Going Further

Let's explore further. Read sector number 1. (The L command starts numbering sectors with 0, so sector number 1 is actually the second sector on the disk.) See Figure 2.

This is the file allocation table, the FAT. The first byte is the fat_id byte, 0FDH for 360K floppies. The rest of the table tells DOS which clusters hold which files. Now read the first sector, the boot sector, and dump the first few bytes. See Figure 3.

EB 34 90 at offset 0000 is a jump command. Bytes 0003 - 000B are used for OEM identification. They show that I used PC-DOS 3.2 to format the disk. The next 19 bytes are an extended BPB table. They show this disk having —

```
0200 bytes per sector
 02 sectors/alloc unit (cluster)
 0001 reserved sector (boot sector)
 02 FATs
 0070 directory entries in root
 02D0 total sectors on the disk
   FD media descriptor byte
 0002 sectors for each FAT
 0009 sectors per track
 0002 heads
 0000 hidden sectors
```

Figure 3 - Start of Boot Sector

```
-L 0 1 0 1
-D 0 2F
128E:0000 EB 34 90 49 42 4D 20 20-33 2E 32 00 02 02 01 00 .4.IBM 3.2.....
128E:0010 02 70 00 D0 02 FD 02 00-09 00 02 00 00 00 00 00 00 .p.....
128E:0020 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 0F .....
```

Figure 4 - Interrupt Vectors 20H - 3FH

```
-d 0:80
0000:0080 6C 13 91 02 8D 13 91 02-F5 02 35 0E 2E 03 35 0E 1.....5...5.
0000:0090 BD 02 35 0E 09 15 91 02-4C 15 91 02 FE 5D 91 02 ..5.....L....].
0000:00A0 72 13 91 02 7A 01 70 00-72 13 91 02 72 13 91 02 r...z.p.r...r...
0000:00B0 72 13 91 02 72 13 91 02-57 02 26 0D BD 1E 70 00 r...r...W.&...p.
0000:00C0 EA 73 13 91 02 13 91 02-72 13 91 02 72 13 91 02 .s.....r...r...
0000:00D0 72 13 91 02 72 13 91 02-72 13 91 02 72 13 91 02 r...r...r...r...
0000:00E0 72 13 91 02 72 13 91 02-72 13 91 02 72 13 91 02 r...r...r...r...
0000:00F0 72 13 91 02 72 13 91 02-72 13 91 02 72 13 91 02 r...r...r...r...
```

Figure 5 - NUL Device Header

```
-d 291:48 59
0291:0040 00 00 C8 09 04 80 C6 14 ..NUL .....
0291:0050 CC 14 4E 55 4C 20 20 20-20 20
```

Looking At Memory

Now let's leave the disk, and look at memory. The PC-DOS BIOS, IB-MIO.COM, is always at segment 0070. DOS proper is loaded at the end of the BIOS.

The size of the BIOS varies not only with DOS versions, but also on machine configuration. For instance, the BIOS throws out the drivers for the hard disk on machines which do not have hard disks. On an AT the BIOS includes some patches for bugs in the AT ROM BIOS.

One way to find the start of DOS is to look at the interrupt vectors in Figure 4.

These are the interrupt vectors for interrupts 20H - 3FH, which are reserved by DOS. Each interrupt vector consists of 4 bytes — a segment:offset pointer. Remember that the lower significant byte is at the lower address.

The first few vectors are 0291:136C, 0291:138D, 0E35:02F5, 0E35:032E. Most interrupts will point to DOS. Look over the segment values in these interrupts, and notice that most segment values are 0291. This is the DOS

segment.

(What are the others? BIOS is at 0070, DEBUG has been loaded at 0E35. Investigating 0D26, int 2E, and 9113, int 30 is left as an exercise, because I don't know.)

Finding NUL

The NUL device is in DOS. Now that we know where DOS is, we can use the debug search command to find it.

```
-s 291:0 fff0 "NUL "
```

```
0291:0052
0291:01B7
0291:9486
0291:9711
0291:EB5A
0291:EF68
0291:EF76
```

The first address is the one we want. The others are from various buffers in DOS and DEBUG which process the search command and so contain the string 'NUL'. The name of the device driver is at offset 0A in the header, so the NUL device header starts at 48 and is 11 bytes long. See Figure 5.

The first four bytes are the link to the next device driver. Remember that the lower part of the address is in the lower bytes. So the next device is at 09C8:0000. NUL's attribute bits are 8004, character device, NUL device. Strategy and interrupt entry points are at 14C6 and 14CC (see Figure 6). What does the NUL device do?

Not much, NUL merely sets the done bit in the status word and returns. Let's see what the next device is. Device headers are 11 bytes long, so the next device header is at 9C8:0 to 11. See Figure 7.

Lost?

Don't be alarmed if you find yourself at a different device driver. I installed this device with CONFIG.SYS. You should see whatever

(continued next page)

DEBUGGING A DISK

(continued from page 13)

devices you have installed through CONFIG.SYS. This one is a block device with non-IBM formats (attribute = 2000); it has only one unit. It's a memory disk, my drive D:. See Figure 8.

This is my installable device for expanded memory management. Notice the next device has segment 0070, and so is in the BIOS. See Figure 9.

Here we are at the BIOS CON driver. If you got lost along the way, do a quick search. S 70:0 FFF0 "CON" will find occurrences of the string 'CON' in the BIOS. Remember, this string will be 0A bytes past the start of the device driver. Attribute 8013 says this character device is standard input, standard output and can do output through int 29H.

Since we are now in the BIOS, your results should be similar to mine. See Figure 10 for the next few drivers.

This device driver is a block device which supports removable media calls, and can process control strings (Attribute = 0840). This device driver has three units, for the two floppies, and one hard disk on my system.

■ ■ ■

Figure 6 - Strategy And Interrupt Entry Points

-u 291:14c6 14cc

0291:14C6 26	ES:
0291:14C7 814F030001	OR
0291:14CC CB	RETF

Figure 7 - RAM Disk Device Header

-d 9c8:0 11

09C8:0000 00 00 75 09 00 20 35 01-D4 02 01 00 00 00 00 00	..u... 5.....
09C8:0010 00 00	..

Figure 8 - Expanded Memory Management Device Header

-d 975:0 11

0975:0000 B3 0B 70 00 00 80 4C 00-21 01 45 4D 4D 58 58 58	..p...L.I.EMMXXX
0975:0010 58 30	X0

Figure 9 - BIOS CON Driver

-d 70:bb3 bc4

0070:0BBO 68 0C 70 00 13-80 C6 00 D1 00 43 4F 4E	h.p.....CON
0070:0BC0 20 20 20 20 20	

Figure 10 - More BIOS Drivers

-d 70:c68 c79

0070:0C60 17 0D 70 00 00 80 C6 00	..p.....
0070:0C70 D7 00 41 55 58 20 20 20-20 20	..AUX

-d 70:d17 d28

0070:0D10 15-0E 70 00 40 A0 C6 00 E6	..p.0....
0070:0D20 00 50 52 4E 20 20 20 20-20	.PRN

-d 70:e15 e26

0070:0E10 E5 0E 70-00 08 80 C6 00 0C 01 43	..p.....C
0070:0E20 4C 4F 43 4B 24 20 20	LOCK\$

-d 70:ee5 ef6

0070:0EE0 7A 0C 70-00 40 08 C6 00 12 01 03	z.p.0.....
0070:0EF0 6A 13 70 00 80 00 00	j.p....

-d 70:c7a c8b

0070:0C70 29 0D 70 00 00 80).p...
0070:0C80 C6 00 D7 00 43 4F 4D 31-20 20 20 20COM1

-d 70:d29 d3a

0070:0D20 71 20 70 00 40 A0 C6	q p.0..
0070:0D30 00 EC 00 4C 50 54 31 20-20 20 20	...LPT1

-d 70:2071 2082

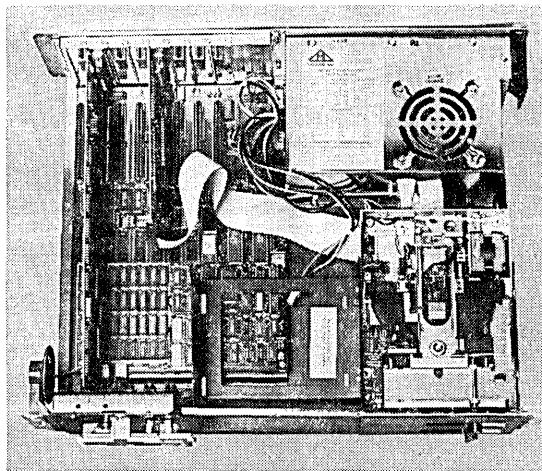
0070:2070 83 20 70 00 40 A0 C6-00 F4 00 4C 50 54 32 20	. p.0....LPT2
0070:2080 20 20 20	

-d 70:2095 20a6

0070:2090 FF FF 70-00 00 80 C6 00 DD 00 43	..p.....C
0070:20A0 4F 4D 32 20 20 20 20	OM2

Note: the FFFF in the link field indicates that this is the last device driver.

BUILD YOUR OWN PC XT/XT TURBO



EXPANDABILITY

8 IBM Compatible I/O Slot bus. Works with all Hardware for IBM PC/XT

VIDEO CARD

Monochrome graphic card or color graphic card

CUSTOM LANGUAGES

EPROM & ROM sockets, accepts 2764-27128 chips for custom BIOS applications.

RANDOM ACCESS MEMORY

640K RAM on board. Utilizes 41256-150 N.S. RAM chips. 250% higher reliability factor based on 2 5ths amount of memory chips required with IBM PC/XT for 640K RAM using 64K RAM chips. Memory versatility, single 256K or 640K RAM on board selectable jumper.

OPTIONAL PROCESSOR

INTEL 8087 Math CPU socket for volume numerical calculations.

ENCLOSURE

Convenient flip-top style or slide-in style metal case. Scratch resistant enamel surface.

POWER SUPPLY

150 Watt output, 110 or 220 volt operation. Quiet cooling fan. Ample current for high-wattage consumption accessories.

FLOPPY DRIVES

Model Teac-55BV 5 1/4" 1/2 Height, Double Sided, Double Density, 48 TPI, 360K Storage, Direct Drive, Quiet mechanism, low power consumption.

HARD DRIVE UPGRADES

Room for two optional half height accessories. Auxiliary cooling fan mounting capability.

PC/XT/AT COMPATIBLE ADD ON & ACCESSORIES

MONITORS

RGB Color/amber/green (TVM)	\$359.00
TTL Monochrome amber/soft white (Mctek)	109.00

SYSTEM COMPONENTS

XT Motherboard 0-640K (2 layers) 4.77/8MHZ	149.00
XT Motherboard 0-640K (4 layers) 4.77 MHZ	149.00
XT Turbo-motherboard (4 layers) 8 MHZ	179.00
XT Case (flip-top)	45.00
XT Case (slide-in)	49.00
XT 150W power supply	85.00
XT 5160 keyboard (AT style) FCC approved	79.00
Floppy controller w/cable	39.00
Multi I/O card	89.00
384K Multifunction card (OK)	85.00
I/O Plus II	65.00
Color graphic card	65.00
Monochrome graphic w/printer port I	85.00
Teac 55BV Floppy drive	109.00
Assembly & Test	25.00

HARD DISK SYSTEMS:

Internal SEAGATE ST-215 10MB hard disk drive with controller	360.00
Internal SEAGATE ST-225 20 MB hard disk drive with controller	399.00
Internal 30MB hard disk drive with controller	655.00
V20-8	25.00
EasyData 1200H (half size internal modem)	139.00
half size internal modem card for IBM Portable and PC/XT/AT and compatibles, same features as 1200L but without the additional serial port.	

XT Assembly Manual	9.00
--------------------	------

HARDWARE FOR IBM AT

a. AT compatible motherboard	\$650.00
8 slots 1 MB memory on board, expandable to 1 M	
b. Combined floppy/hard disk controller	239.00
c. Printer port/serial port card	99.00
d. 200 watt power supply	139.00
e. 1.2MB floppy disk drive	155.00
f. 2MB multifunction	199.00
OK expandable to 2M, parallel port, 2 serial ports, game port	
g. 3MB multifunction	229.00
OK expandable to 3M, parallel port, 2 serial ports, game port	
h. AT type case	89.00
i. AT compatible keyboard	89.00
j. 20MB hard disk, SEAGATE 4026	595.00
k. 30MB hard disk, SEAGATE 4038	739.00
l. 40MB hard disk, SEAGATE 4051	899.00

ACCESSORIES

Surge Suppressed outlet strip	9.50
Surge Suppressed outlet strip (UL approved)	12.95
Parallel printer cable/serial cable	9.99
Disk storage w/lock (holds 100 pcs)	14.50
Disk storage w/lock (holds 50 pcs)	10.95
Roll Top Disk Storage (holds 100 pcs)	29.50

McTEK SYSTEMS, INC.

2316 FOURTH STREET
BERKELEY, CA 94710
(415) 549-3472

DEALERS: Identify yourself as a dealer and we'll send you dealer information. We're reliable, efficient, and price competitive.

Professionalize Your Programs

Using Turbo Pascal's Built-in System Procedures

Turbo Pascal is a powerful development system that you can use to write professional programs. System hooks like the built-in procedures MsDos and INTR let you speed up your creations. If you already program in Pascal (or just want to know what the jabbering's about), then come along on Rick's guided tour into the depths of Turbo.

System level procedures like MsDOS and INTR are operating system-specific to MS and PC-DOS, and like the BIOS and BDOS procedures built-into the CP/M version of Turbo Pascal, they let you access the operating system.

Reading a disk sector, checking video type, determining disk density, and checking the printer (is it ready or disconnected?) before sending it a character are problems you can solve easily and efficiently using the MsDos or INTR procedures.

Defining A Free Union Variant Record

In order to use either MsDos or INTR, we must first define a variable type and variable. I'll call these "RegisterRecord" and "Registers."

The variable called "Registers" will represent the CPU's registers so we'll have an easy way to preset them before calling DOS.

For the variable type, we'll use a free union variant record (a record defined without a tag field, but with a tag field type). See Figure 1.

You can think of a free union as two or more definitions of the same space. We could do the same with absolute variables.

Editor's note: The more common kind of variant record is called a "discriminated union." In a discriminated union, the variants are actually different from each other, and a "tag field" is used to tell

which one is in force (to discriminate between them). See Figure 2.

In Figure 2, the tag field is "sport." After that, the record is completely different, depending on whether the sport is baseball or football.

Since a free union has no tag field, there is no way to discriminate between its variants. They are simply two (or more) different ways of looking at the same data (in this case, the first four registers).

"RegisterRecord" allows us to directly reference any of the first 4 registers as 16 bit registers (AX, BX, CX, DX) or as 8 bit registers (AL, AH, BL, BH, etc.), where H refers to the high byte of the register and L refers to the low byte. This breakdown into 8-bit registers makes some functions easier to write and easier to understand later.

Editor's note: If you create a non-variant record to hold register values, it's a little more work to set the high order bytes — you have to multiply the original value by 256 in order to shift the bits into the high byte.

System Level Routines

I like to break system level routines down into three groups:

Function calls

Interrupts

Direct memory and port manipulations

There are some unwritten rules about when and where to use each group, but you should use function calls if you can. Interrupts are second best, with direct memory and port manipulations last. Microsoft promises to support function calls for all future versions of DOS. Which means: programs that run on current versions of DOS should run on future versions.

Approximately 80 function calls can be accessed through the procedure MsDos. For their names and other useful information, you'll probably need more than the DOS reference manual (the enigmatic doc supplied with every legal purchase of PC or MS-DOS).

If your manual doesn't discuss function calls (the later DOS references

Figure 1 - RegisterRecord

```
type
  RegisterRecord =
    record case integer of
      1:(AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags: integer);
      2:(AL, AH, BL, BH, CL, CH, DL, DH: byte);
    end;
```

Figure 2 - A Discriminated Union

```
type
  game = record
    TeamName: string[30];
  case sport: (baseball, football) of
    baseball: (inning: integer;
               runs, hits, errors: integer;
               BaseballTactics: (bunt, slide, steal,
                                   badger, eject, homerun));
    football: (quarter: integer;
               points: integer;
               penalties: integer;
               FootballTactics: (kill, maim, sack, charge,
                                  trap, bomb, tackle, block));
  end; { game }
```

don't), borrow a Technical Reference for your specific computer and version of DOS. Although not complete, Peter Norton's *Programmer's Guide To The IBM PC* talks about most of them.

GetFreeSpace

Let's use the record declared earlier and the procedure MsDos to determine the space left on a disk.

After fumbling through the DOS manual for a while (am I the only one who thinks this manual is hard to read?) I found that function 36h (54 decimal), GetFreeSpace, does what I need.

Load the function number (36h) into register AH, and the drive number to test into DL (0 = logged drive, 1 = drive A, 2 = drive B, and so on). Nothing else needs to be set.

After making the proper assignments to AH and DL in the Registers record variable, we call MsDos:

```
MsDos( Registers );
```

On exit, the registers will contain the following information —

AH = Sectors per allocation unit, or FFFFh if and only if the drive indicated does not exist. (Obviously this can be used to test for the existence of disk drives.)

Figure 3 - Disk Space Function

```
FUNCTION DiskSpaceFree: integer;
var
  Registers : RegisterRecord;
  Tracks, Sectors, BytesPerSector : integer;
begin
  with Registers do
  begin
    fillchar( Registers, sizeof( Registers ), 0 );
    AH:= $36; { function number }
    DL:= 0; { choose LPT1 }
    MSDOS( Registers );
    Tracks:= BX;
    Sectors:= AX;
    BytesPerSector:= CX;
    if AX = $FFFF then DiskSpaceFree:= AX
    else DiskSpaceFree:= round( Sectors * BytesPerSector/1024.0* Tracks );
  end;
end; { DiskSpaceFree }
```

BX = number of free allocation units (clusters)

CX = bytes per sector

Whoa, there. I don't know about you, but I don't see any mention of free disk space. Assuming the manual hasn't lied (guaranteed, right?), there must be a way to find disk space from this information.

And there is — bytes per sector times sectors per allocation unit times the number of free allocation units. Or, in registers:

AX * BX * CX

to get the number of bytes remaining.

See Figure 3, a complete Turbo Pascal function to check disk space. It returns the number FFFFh if the drive was invalid and the number of kbytes remaining on the selected drive (which was passed as a value parameter) otherwise.

Interrupts

Interrupts are very similar to function calls. (Since functions are called through interrupt 21h, they are really just a special type of interrupt.)

Interrupts are more closely tied to a computer's hardware than function calls, and so may vary in their calling sequences and effects from make to

make or model to model. Therefore it's not as safe to use them. (The BIOS hides these differences when you make function calls.)

To try out an interrupt, let's check the printer port to see whether it's ready to receive a character. It's always good to do this before printing a file since it keeps the system from hanging if there's a printer problem.

Interrupt 17h is devoted specifically to the printer and will tell you, among other things, the current printer status.

According to the DOS manual, we need to put 2 in the AH register (for "get printer status"), and 0 (for LPT1) or 1 (for LPT2) in the DX register to select the printer port. Then use the built-in procedure, INTR, to call the interrupt:

```
intr( $17, Registers );
```

After the call, the status is returned in register AH in the form of an error code.

Test Run

"Great," I thought. "Now I can write a little function to get the printer port status. I'll test it by running my program with the printer cable plugged in and with it unplugged."

It didn't work. The error code returned was independent of the state of the printer. Not very useful.

After a lengthy trial and error process (sparing you the details), I discovered that we have to RESET the printer port before the status is correct. (Couldn't we all write books on what the manuals don't tell us!)

To reset the printer port, we just put 1 in the AH register (for "reset the printer"), and select the printer port in the DX register. Then we call interrupt 17h, and the port is reset.

For the complete PrinterReady function, see Figure 4. This works with my

(continued next page)

USING TURBO PASCAL

(continued from page 17)

computer (Tandy 1000) and printer (HP ThinkJet); it may not work with yours. *Editor's note: It doesn't work with mine.*

I used an interrupt instead of a function call to test the printer port status because I couldn't find a function call that fit my requirements and worked as easily. So the code might be sensitive to some computer/printer combinations. A little risky, perhaps, but probably not dangerous.

Altering Memory Directly

The last programming method is one I'd avoid unless there were no other choice — poking or reading values directly to or from system memory or ports. Software written this way will be most sensitive to differences in equipment.

I've found only one compelling reason for altering memory directly in the MS-DOS programs I've written so far: speeding up the screen (my CP/M Kaypro 10 has much faster screen I/O than my Tandy). In this case, the slowness of the standard function calls and interrupts and the knowledge that most hardware vendors place the video pages in the same memory locations make this choice reasonably sound.

Editor's note: This is one of the primary reasons why very very few PC programs run on the non-clones, and one of the reasons that the clones are such close copies of the original.

Screen Memory

Let's use Turbo's built-in Mem array to alter the screen memory.

There are two things we need to know before writing directly to screen memory — where the screen memory is, and its format. To simplify things, we'll write to the current video page (it's also possible to write to a video page other than the current one and then switch to it, making the screen "change instantly").

The base or starting location of the screen memory is usually in one of two places (in hexadecimal segment:offset notation): B000:0000 or B800:0000.

The former is used for monochrome video controller cards and the latter for

Figure 4 - Printer Ready Function

```
FUNCTION PrinterReady: boolean;
var
  Status : byte;
  Registers : RegisterRecord;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
  begin
    AH:= $01;                                { code to reset the printer }
    DL:= $00;                                { printer number, 0 = LPT1 }
    Intr( $17,Registers );                   { call printer interrupt }
    AH:= $02;                                { code for get the printer status }
    DL:= $00;                                { printer number, 0 = LPT1 }
    intr( $17,Registers );                  { call printer interrupt }
    Status:= AH;
  end;
  PrinterReady:= not Odd( Status shr 4 ); { test bit 4 }
end; { PrinterReady }
```

Figure 5 - Fast Procedure To Display A String

{ Note that there is no check in the procedure below to insure that the text of the line will not wrap around to the next line; it is assumed that the line will fit. Note also that the use of the Turbo whereX and whereY functions assumes that the entire screen is being used. If you wish to use this procedure with windows, it will be necessary to subtract the first column number of the window from whereX and the first line number from whereY in order to calculate the offset. }

```
type
  string255 = string[255];

PROCEDURE SpeedWrite( Line : string255 );
const
  ScreenSegment = $B800; { for color card, change to $B000 for monochrome }
var
  Offset, i : integer;
begin
  Offset:= pred( whereX )#2 + pred( whereY )#160; { calculate mem. location }
  for i:= 1 to length( Line ) do
  begin
    Mem[ ScreenSegment:Offset ]:= ord( Line[i] ); { set character byte }
    Offset:= Offset + 2;                         { skip attribute byte }
  end;
  gotoXY( whereX + length( Line ), whereY ); { move cursor to end of line }
end; { SpeedWrite }
```

Figure 6 - Reverse Video Toggle

{This procedure swaps the colors of the character and the background at the current cursor position, which effectively toggles reverse video on and off}

```
PROCEDURE InvertCharacter;
var
  Registers : RegisterRecord;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
  begin
    AH:= 8;          { code for read character and attribute at cursor location }
    BH:= 0;          { video page number, 0 = normally active page }
    intr( $10,Registers ); { call video interrupt }
    BL:= (AH shr 4) and $07 + (AH and $07) shl 4 + (AH and $08); { do invert }
    BH:= 0;          { video page number, as above }
    AH:= 9;          { code for write character and attribute }
    CX:= 1;          { number of characters to write }
    intr( $10,Registers ); { call video interrupt }
  end;
end; { InvertCharacter }
```

color (which I use).

The format goes like this — each character location in memory occupies two bytes, a character byte which is the ASCII (or extended ASCII) character and an attribute byte. The attribute

byte describes character color, background color, and whether or not the character is blinking. The two bytes are one after the other with the character byte first. Thus each screen line (80 characters) occupies 160 bytes. Check

Figure 7 - Procedure To Change The Data Transfer Area

```
{ The following procedure will set the DTA to the memory location defined by the values Segment and Offset. All subsequent disk read and write data will be buffered at the new DTA, where you can look at it and modify it if you like. Remember that the DTA must be at least as large as the size of one sector. Default DTA is located at 80h in the program segment prefix. }
```

```
PROCEDURE SetDataTransferArea( Segment, Offset : integer );
var
  Registers : RegisterRecord;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
  begin
    AH:= $1A;           { function code for set DTA }
    DS:= Segment;       { segment portion of address }
    DX:= Offset;        { offset portion of address }
    MsDos( Registers ); { make service call }
  end;
end; { SetDataTransferArea }
```

Figure 8 - Function To Check Disk Type

```
{ The following function returns the disk type of the drive number passed to it, where fixed disk = F8h, quad density = F9h, SS 9 sector = FCh, DS 9 sector = FDh, SS 8 sector = FEh, and DS 8 sector = FFh. }
```

```
PROCEDURE DiskType( Drive : byte );
var
  Registers : RegisterRecord;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
  begin
    AH:= $1C;           { function code for get FAT information }
    DL:= Drive;         { disk drive number, 0= Default, 1= A, etc. }
    MsDos( Registers ); { make service call }
    DiskType:= Mem[ DS:BX ];
  end;
end; { DiskType }
```

Figure 9 - Function To Find Logged Drive

```
{ The function below returns the number of the currently logged drive, where for consistency's sake (0 often refers to the default drive) 1 = A, 2 = B, and so on. }
```

```
FUNCTION CurrentDrive : byte;
var
  Registers : RegisterRecord;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
  begin
    AH:= $19;           { function code for get current drive number }
    MsDos( Registers ); { make service call }
    CurrentDrive:= succ( AL ); { 1 = A, 2 = B, etc. }
  end;
end; { CurrentDrive }
```

the DOS reference manual of your choice for a list of attributes.

If we assume that the attributes are fine as they are, we can use the procedure in Figure 5 to add a string beginning at the current cursor location.

If you use this procedure with a standard IBM video controller, you may see snow on the screen while the data's being written. You can stop the snow by poking data into screen memory only during horizontal or vertical retrace.

To further illustrate the use of function calls and interrupts, I have included several additional routines (see Figures 6-10). These are variations of procedures and functions that I use in my own programming.

Where To Go For Help

Now that you know something about system level routines, the real challenge rears its recurring head: where do you find the information you need to apply these system level hooks?

A natural response is — ask someone who already knows how to apply them, the "use an expert" ploy.

Good idea, but unfortunately many professional programmers don't know how to apply service calls. Besides, it's more fun sometimes to muck around yourself, and it won't take much mucking to become an expert. Which leads to question two, "Where does an expert turn for help?"

Public domain and shareware programs that include source code, and technical magazines like Micro C and PC Tech Journal are good sources.

Books

Another alternative is to buy aftermarket books that address the needs at hand.

The first book almost everyone buys is *Programmer's Guide to the IBM PC* by Peter Norton. It contains a great deal of information, is written in a style noticeably better than Norton's articles in PC Week or PC Magazine, and is reasonably well-organized. However, *Programmer's Guide* suffers from an attitude problem.

Too many times Norton pushes his
(continued next page)

(continued from page 19)

other books and programs or leaves the reader hanging with phrases like "since this book does not cover (subject goes here), you should look elsewhere for that information." I call the first irritation commercialism. The latter is nothing less than "whimping" in a book that proclaims to be the "ultimate reference guide to the entire family of IBM PCs." Perhaps Norton's "Utilities" (\$99) have destroyed his candor.

For example, when he discusses reading a hard disk sector via interrupt 13h, Norton writes that service 0Ah of this interrupt will perform a long read of an AT hard disk sector. Allegedly, the CH register should contain the cylinder number, but since this only allows cylinder numbers below 256, and we find elsewhere in the book that an AT hard disk contains 605 cylinders, we must conclude that this information is either incomplete or incorrect. To make matters worse, Norton adds:

"We'll outline the new services here, but we won't go into any great detail for the same reason we have passed lightly over many other model-dependent features: Our main concern

Figure 10 - Cursor Switch

```

{ The following procedure will turn the cursor on or off }

PROCEDURE TurnCursor( State : boolean );
const
  Visible = 0;
  Invisible = 1;
  StartLine : integer = $06; { start and end lines should be changed }
  EndLine : integer = $07; { for monochrome cards }
  CursorType : integer = $00;
begin
  fillchar( Registers, sizeof( Registers ), 0 );
  with Registers do
    begin
      case State of
        false : begin { blanks cursor }
          CursorType:= Invisible;
          CH:= CursorType shl 5 + StartLine;
          CL:= EndLine;
        end;
        true : begin { sets cursor on }
          CursorType:= Visible;
          CH:= CursorType shl 5 + StartLine;
          CL:= EndLine;
        end;
      end; { of case statement }
      AH:= $01; { code for set cursor }
      intr( $10, Registers ); { call video interrupt }
    end;
end; { TurnCursor }

```

in this book is to explore the entire PC family, not the peculiarities of one model or another."

Norton sometime forgets that he's titled his book, *Programmer's Guide to the IBM PC*, and that programmers are likely to want to know this type of information.

"Mapping The IBM PC & PCJr"

A book that doesn't share this attitude problem is *Compute's Mapping the IBM PC and PCjr* by Russ Davies.

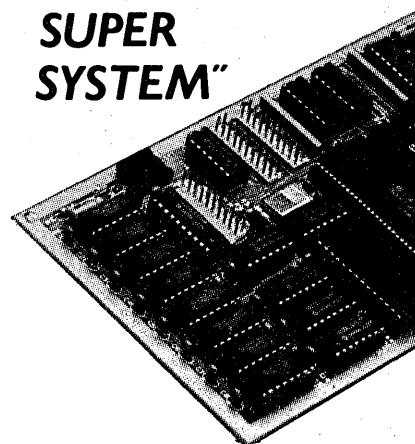
It thoroughly covers the memory and port organization of the PC, with many programming examples in BASIC and assembly, as well as interrupts and service calls. Its organization isn't quite as good as *Programmer's Guide*, so it's not as easy to use as a reference, but it doesn't leave the impression that the author purposely withheld information.

Finally

If you're really interested in improving your programming style, I heartily recommend you write a disk editor. After writing CP/M PROBE for Micro C's Pascal programming contest and working on an MS-DOS version, I'm convinced there's no better way to learn the quirks of an operating system.

Byte Magazine called it.

"CIARCA'S SUPER SYSTEM"



The SBI80 Computer/Controller

*Featured on the cover of Byte, Sept. 1985,
the SB180 lets CP/M users upgrade to a
fast, 4" x 7½" 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 3½", 5¼"
and 8" drives).
 - **Measures 4" x 7½", 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**

**For technical assistance or
to request a data sheet, call:
1-203-871-6170**



Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066

★ SPECIAL ★

DBASE BOOK OF BUSINESS APPLICATIONS

by Michael J. Clifford \$5.95

Reg. \$19.95 While they last ONLY \$7.95

KAYPRO EQUIPMENT

9" Amber CRT	\$39.00
9" Green Monitor	35.00
12" Green Monitor	45.00
81 Series ROM's	7.00
Keyboard (Kaypro)	75.00
Keyboard Siemens w/case	65.00

KAYPRO MAINBOARDS

K2 With Sockets	\$39.00
K2 Populated - Tested	129.00
K2 with PRO8-3 Mod.	239.00
<i>This one runs SS, DS, or Quad drives. With manual & software, tested and guaranteed 30 days.</i>	
K2 PRO8-3 Mod. EXCHANGE	149.00
WD1002-5 Hard Disk Controller	200.00
Host Interface Board	15.00

CABINETS

K2X	\$35.00
K10	35.00
<i>Bottom portion with front and back only.</i>	

CRT Bezel	12.00
-----------	-------

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

CPU/SUPPORT

MC68000-8 CPU	\$9.99
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

TEST EQUIPMENT

OSCILLOSCOPES

Tek 485 350 MHz	\$2995
H-P 1725A 275 MHz	2095
Tek 454 150 MHz	825
Phillips 3260E 120 MHz	995
Phillips 3214 25 MHz	575

SPECTRUM ANALYZERS

H-P 140T/8554B/8552A	\$4900
Tek 491 10 MHz-40 GHz	4600
H-P 851B/8551B 10 MHz-40 GHz	1500

HOURS: Mon. - Fri. 9-6 — Sat. 10-4

MINIMUM ORDER — \$15.00

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. For more info please call — DON'T WRITE! We don't have time to answer your letters.

IBM/PC COMPATIBLES

Mainboard, 8 Slot, Case,
Power Supply \$225

To make this a complete system, add

A) Memory B) Floppy Controller
C) Drive D) Keyboard E) Video Card
F) Video Monitor G) Multifunction Card

A) **MEMORY**

256K 150 NS	\$25
512K 150 NS	50
640K 150 NS	68

B) **FLOPPY DISK CONTROLLER**

Card for 2 Floppy Drives	\$36
Card for 4 Floppy Drives	42

C) **5 1/4" FLOPPY DISK DRIVES**

Mitsubishi M4851 DSDD 40 Tr	\$99
Mitsubishi M4853 DSDD 80 Tr	.99
Fujitsu M2551A DSDD 40 Tr	.99
Panasonic JU455 DSDD 40 Tr	105
Shugart 475 DS Quad 1.2Mb	159

D) **KEYBOARDS**

Cherry Keyboard (no case)	\$38
AT Style Keyboard	69

E) **VIDEO CARDS**

Tomcat with Parallel and Lightpen Port	\$53
Hercules compatible Video Board	75
Color Graphics Adapter	69
Enhanced Graphics Adptr-(EGA)	.275

F) **VIDEO MONITORS**

Roland MB-122G, 12" (no case)	\$39
Samsung MD-1254G, 12" Green	82
Samsung MD-1254A, 12" Amber	88
Goldstar MBM-2015, 12" Amber	106
Mitsi AT-1332A 13" RGB TTL	.295
Goldstar MCH-4015 13" RGB	.420
Goldstar MCL4333 12"	
Composite Color Monitor	159
<i>1 year parts & labor. Apple & Commodore compatible, with audio amp & volume control. Great for VCRs.</i>	

G) **MULTI FUNCTION CARD**

Parallel & Serial Port, Game Port Floppy Controller, Clock & Cal.	\$96
----------------------------------------------------------------------	------

EGA PACKAGE DEAL

Package consists of Intergraph + 4 EGA Card and the Autoseek 2000 EGA Monitor by Int'l Graphics. No software patches necessary. 1 yr. guar. . . . \$795

EPROMS

2716	\$3.50
2732	3.75
2764-25	3.50

PRINTERS

Centronics 703	\$395.00
Centronics 588 - As Is	125.00
NEC2000 w/Single Sheet Feed	.999.00

Sweda Cash Registers

Models 303, 2640 & 2650
From major supermarket . . . \$99 ea.

SYSTEM EXAMPLE #1 For the Hacker (Cheap)

Mainboard, Case, Power Supply	\$225
256K Memory 150 NS	25
Floppy Controller (2 Drives)	36
Floppy Drive 1/2 Ht DSDD	99
Keyboard Cherry (no case)	38
Video Board with Parallel and Lightpen Port	53
Roland MB-122G, 12" Green Monitor (no case)	39

\$515

SYSTEM EXAMPLE #2 FCC Approved (Not Cheap)

Mainboard, Case, Power Supply	\$225
640K Memory 150 NS	68
Multi I/O, Parallel, Serial, Floppy, Clock/Cal.	96
2 DSDD Floppy Drives (minimum)	198
EGA Package	795
AT Style Keyboard	69

\$1451

Oh, you wanted a turbo board . . . 40
and a 20M Hard Drive & Controller . . . 410

Now how much would you pay?

JUST \$1901

HARD DISK DRIVES

10M Seagate 212	200
10M Rodin RO-252	230
20M ST-225	385
20M Tandon TM252	350
20M Tulin (Oki)	345
20M Half Height with Controller	410
32M Half Height with Controller	667
40M Quantum Q540	650
60M with Controller	1150
70M Vertex V170	725

HARD DISK CONTROLLERS

WD-1002-WX2 with Cable	\$156
Omni-5510	150
Adaptec 2770A	
(Get 15M on 10M Drive)	156

POWER SUPPLIES

Elgar 400W Unit. Power Sup.	\$375.00
+5V/1A, -5V/.2A, +12V/1A,	
-12V/.2A, -24V/.05A	15.00
+24V/2.2A	8.00

SWITCHERS

5V/9.5A, 12V/3.8A, -12V/.8A	\$39.00
5V/3A, 12V/2A, -12V/.4A	29.00
5V/10A	25.00
5V/75A, +12V/8A, +12V/8A	.55.00
5V/20A	35.00

MISCELLANEOUS

Headset/Boom Microphone	\$3.95
Nicad Pack 12V/.5AH	6.50
5 Blade Muffin Fans	7.50
Joystick 4 Switches 1" Knob	5.50

Controlling The Real World With Your PC

Intro To Electronics Step 2

Bruce continues his easy, low-tech approach to explaining hi-tech. Once you realize how easy it is for a computer to control electro/mechanical devices, you'll probably have the computer controlling your whole house (teenagers and all). (If you're interested in controlling 120VAC with your system, check out the S-100 column in this issue.)

This article is the second in a series. In the first, I discussed setting up a lab and showed how to connect a computer (via the parallel port) to a breadboard using optocouplers to control non-TTL voltage and current levels. These are the voltages required by many "real world" gadgets, including stepper motors.

This time, I'll describe a stepper motor, hook it up to a breadboard, and program it to move.

Magnetism & Motors

Think of a motor as having two

main parts — a rotor (turning part) and a stator (stationary part). Each part contains magnets, which can be manipulated to turn and stop the rotor. The north pole of one magnet attracts the south pole of another, but north repels north (& south repels south) causing the rotor to turn.

You can make the stator a north or south pole by driving a current through it.

In an AC motor, a "rotating magnetic field" is produced in the stator by the AC line current. The rotor tries to catch up to this rotating field.

In a DC motor, mechanical or electronic switching changes the field every time the rotor gets close ("the poles are always more attractive in the next field?") so it keeps moving.

In a stepper motor, the rotor (instead of being continually teased forward, as in a DC motor) is allowed to rest in little pockets of stable magnetism. If you grasp the shaft of a stepper and twist, you'll find it takes quite a bit of force to move it out of one of these little pockets. But by changing the polarity of the pockets, you can

make the rotor jump from one magnetic pocket to the next.

The stator consists of many magnetic pockets. Some motors have more stators than others; you can determine how many if you know the "step angle" — the number of degrees the shaft turns when the rotor jumps from one pocket to the next.

7.5 and 15 degrees are common step angles, and the steppers would have 48 and 24 magnetic pockets, respectively (360 degrees — a full revolution — divided by 7.5 and 15, the step angles).

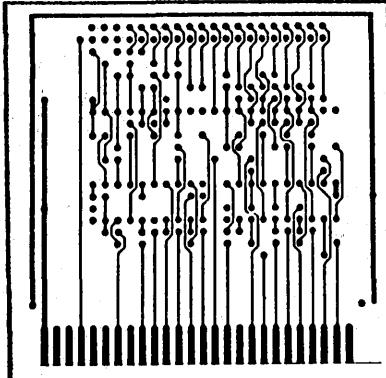
Your printer uses a stepper motor to position the print head. Turn the printer off and grasp the head (careful if it's been printing — it's probably hot!).

Slide it back and forth. Now turn the printer on (most printers will whack the print head against the left stop a few times — to find the head position in case somebody just moved it back and forth). Now move the head. Harder! What did you find out?

Most stepper motors have permanent magnets with windings around

PCB EDIT

- o CREATE DOUBLE SIDED BOARDS ON YOUR PC OR XT -- FAST --
- o MOUSE SUPPORT - CUT AND PASTE -- UNLIMITED TRACE WIDTHS -- 10 MIL TRACE SEPARATION -- BLOCK PAD OR TRACE DELETE.
- o OUTPUT TO DOT MATRIX PRINTER (PLOTTER DRIVER DUE SEPT 86).
- o MUCH MORE - CALL FOR INFO
- o ONLY \$89.95



DON FLETCHER
RT 7 BOX 120
COLUMBIA, MO
65202
(314) 443-7294

ANALOGIC

- o 32 CHANNEL LOGIC ANAL FOR THE PC/XT (STAND ALONE CPM-80 VERSION READY JANUARY 1987)
- o 16 BIT TRIGGER WORD, 4 CLOCK QUALIFIERS
- o TIMING DIAGRAM, HEX, OR DEBUG/DDT DISPLAY
- o BARE BOARD WITH SOFTWARE AND SCHEMATICS - \$75.00
- o ONLY \$389.95

them. The windings bump the rotor from one magnetic pocket to the next and hold the rotor in a particular pocket. Changing the polarity of one of the windings causes the rotor to move. Leaving the polarities alone holds the motor in place; but current will continue to flow through the coils, making it difficult to move when you push on it.

Where To Find Steppers?

I got my mine out of old 8" disk drives which had been sold for scrap. I highly recommend this route, since it's a tremendous education to take a drive apart.

If you can't find any scrap which isn't more expensive than a new motor, try a mail-order house (get a name from Micro C or Computer Shopper) like American Design Components, 62 Joseph St., Moonachie, NJ 07074. If you're really in a hurry, try Radio Shack. It might have them, and you shouldn't have to pay more than \$10 for one.

Fooling With A Motor

The best way I found to get a feel for one of these things requires a nine-volt battery and a willingness to play around with it.

Although most steppers say they require 24 Volts (this is indeed what the final circuit was designed for), you can usually get some kind of action out of them with a nine volt battery. They just won't have much twist on nine volts.

Some steppers have two coils; some have four. Sometimes they have a common ground; sometimes the two terminals of each coil are brought out on a separate wire.

In any case, by playing around with the battery, you should be able to figure out enough about the connection to hook the motor into our circuit from last issue.

Exploring A Stepper

Here's my approach — using a stepper from a scrap drive.

I had five wires coming out of the stepper motor. I took the ground wire of the nine-volt battery clip and connected it to what looked like it might be ground on the stepper motor (probably a black wire).

I didn't worry too much about hurting it; I didn't think I could do much damage with a nine-volt battery. The other four wires on the stepper were connected to the four coils. I poked them with the plus wire from the battery clip.

Here I noticed a strange thing — when the stepper is in some positions, poking some wires has no effect; poking some others will cause it to go backward, others will send it forward. The trick is to determine the sequence of wires that have to be pulsed to make the motor go forward and backward. We need to know these sequences in order to program it.

There are actually many ways to set a motor up to do its job. For example, if you need more "holding torque" (it resists more when it's at rest in a well and you try to move it), you can run current through two coils at once. It can get complicated. Fortunately, you can make it work usefully just by determining the step sequence and turning on one wire at a time.

Keep it stupid, simple (at least at first). If, for example, you appear to have two wires for each coil (the coils have no electrical contact with each other), just tie one end of each coil to ground.

Modifying The Circuit

Once you have the step sequence worked out, let's modify our circuit (from last issue) so the computer can directly control the stepper motor. That circuit drove an LED through a current-limiting resistor.

In the diagram (Figure 1), you'll see the LED and resistor have been replaced with an inductor (representing a single stepper motor coil) and a parallel diode. You should have one of these for each coil of your stepper motor.

Now, when the Darlington transistor is turned on, the stepper motor coil will be energized. Since each coil is controlled by one bit from the parallel port, we can write software to drive the motor.

What's the Diode For?

When any kind of inductor is energized, it creates a magnetic field and stores energy in it. When there's no voltage across the coil, there's no field.

The instant a voltage appears across the coil, current wants to flow, but the coil refuses to let it pass until it builds up its magnetic field. As the field builds, current starts to flow.

When the field is completely built up, current zooms through. (The coil is happy, having exacted its energy toll from the first electrons through in order to build its field, so it doesn't care about the rest.) In fact, at this point there is no voltage across the coil anymore — it just looks like a piece of wire.

The coil resists any change in the current flow through it. When the current tries to start, it resists. And when the current tries to stop, it resists as well.

If you suddenly stop driving current through the coil, it will continue to push current through itself, getting energy by collapsing the magnetic field it built up. It stores energy in the magnetic field when you try to start it, and draws energy from the field when you try to shut it down.

This energy is formidable. I've heard stories of attempts to develop a circuit breaker for use in DC power transmission. Apparently, the currents are so tremendous that even small inductances in the power lines can cause amazing destruction of the circuit breaker, its housing, transformers, etc. I guess it just blows everything apart.

Of course, we aren't working with anywhere near those voltages or currents in this case, but the principle is the same: if you try to abruptly shut off the current through an inductor, it won't let you at first because current continues to flow.

So if we were to shut the Darlington off abruptly (and, compared to the coil, the Darlington shuts off virtually instantaneously), the coil would force current through the transistor and punch a hole through it. Bad news.

Shunt

This is where the shunt diode comes in.

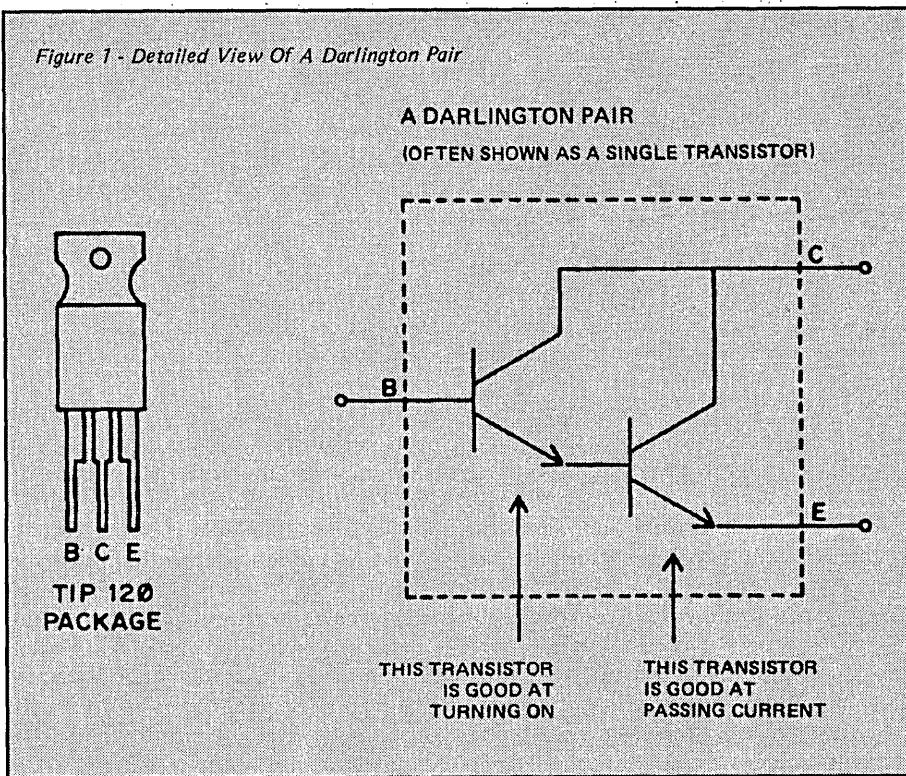
When we turn the coil on, the voltage appears from the emitter of the Darlington to ground (you can see this in Figure 1) and attempts to pass current through the little bar part of

(continued next page)

CONTROLLING THE REAL WORLD

(continued from page 23)

Figure 1 - Detailed View Of A Darlington Pair



the diode, in the opposite direction of the arrow. But it's no go. It's as if the diode isn't there.

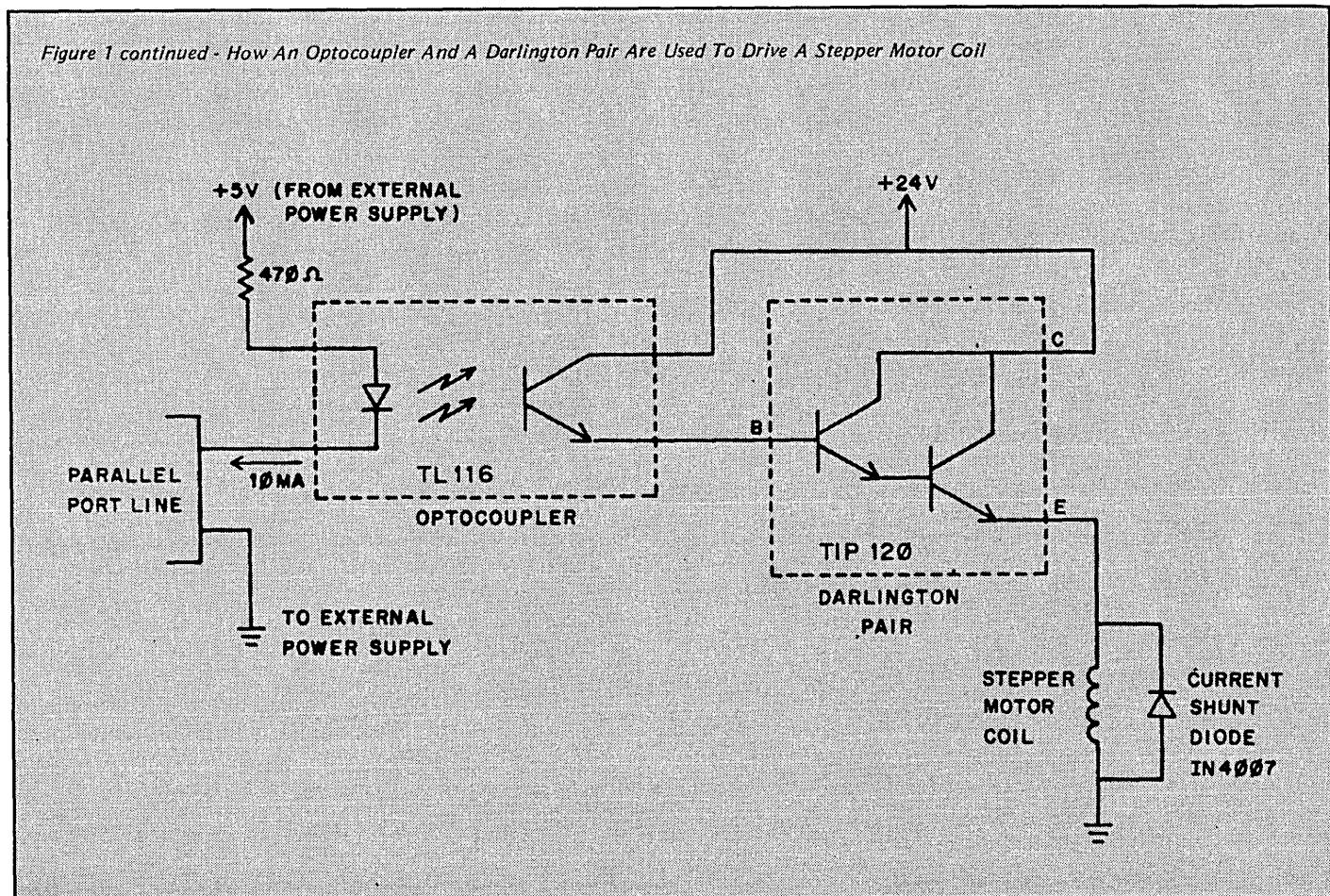
If we suddenly shut the transistor off, the current wants to continue flowing from the emitter through the coil through ground. It can't pull any more out of the emitter (without damaging it), but it CAN easily circulate current around through the shunt diode, since the direction the coil is pushing the current is in the direction of the diode's arrow. Thus, the current just runs around in a circle until the magnetic field dissipates.

If you were using a relay to switch even higher voltages/currents (110 volts, for instance), you would use the same circuit to prevent damage to the transistor by the relay coil.

Hook Up

The best way I've found to connect the stepper is to arrange the coils in their "forward" step order; that is, the first coil to be energized should be connected to bit zero of the parallel

Figure 1 continued - How An Optocoupler And A Darlington Pair Are Used To Drive A Stepper Motor Coil



port, the second should be connected to bit one, etc.. This makes programming a lot easier.

Programming can also be simplified by the following — if you want to set a bit, the byte to put to the output port is 2^{bit} . Thus, bit 0 is set by outputting 2^0 , bit one is set with 2^1 , etc.. So, if you write the BASIC code —

```
100 FOR I = 0 TO 3 : OUT &H19,2^I :  
NEXT
```

you'll cause the motor to move through four of its steps (the port number works for my Kaypro 2X; yours will differ). This could be made into a subroutine or function and called as many times as you need to move the motor to the desired position.

If you follow this line of development, you'll discover something rather fascinating — the overhead of the function or subroutine call will cause the motor to move inconsistently. For example, if you write the code —

```
10 FOR J = 1 TO 50 : GOSUB 100 :  
NEXT
```

and add

```
110 RETURN
```

you'll discover an uneven sound. While statement 100 is executing, the steps come out fairly evenly, but when it RETURNS and then GOSUBs again, there's extra time inserted.

In this case, BASIC is the culprit. My version is interpreted, so a fundamental piece of code is running all the time which looks at the statements, decodes them, decides what to do with them, and then does it. A slow approach. In addition, the exponential calculation 2^{BIT} is usually floating point, so while it's easy to program, it's lousy during execution.

But it gives you an idea of some of the problems involved in real-time programming (hooray for BASIC!). You're not just concerned with getting the job done; you also have to worry about the details — timing, speed, etc. Everything counts when you're affecting things in the real world.

The solution to this problem is fairly

simple, since even the BASIC interpreter is fairly fast in comparison with the maximum speed at which stepper motors usually operate. All we need do is get the time between writes to the port to be fairly consistent. As a subroutine —

```
200 FOR J = 1 TO K  
210 OUT &H19,1:'Set bit0,Clear rest  
220 OUT &H19,2:'Set bit1,Clear rest  
230 OUT &H19,4:'Set bit2,Clear rest  
240 OUT &H19,8:'Set bit3,Clear rest  
250 NEXT  
260 RETURN
```

Which can be called with —

```
20 K = 50 : GOSUB 200
```

I also eliminated the floating-point exponentiation. It's not as elegant, but it runs faster. So fast, in fact, that the motor may begin to oscillate because its coils are being energized faster than it can physically respond.

There are a number of ways to fix this (but be careful, one fix leads to another). If you're using assembly language you have enough control to

"ramp up" the stepper motor — slowly move it up to the speed you want so it will be turning fast enough not to oscillate. In BASIC we can create a dummy loop —

```
FOR Q = 1 TO 10 : NEXT
```

after each "OUT" statement to slow the motor down.

You can tweak the delay time by changing the number in the "FOR" loop and experiment to determine when it starts oscillating.

You can also get a little more control of the timing by using a compiler — which generates assembly code that runs much faster than interpreted BASIC.

Since everything runs so much faster, you can write more elegant code (procedure calls in Turbo Pascal are much quicker than subroutine calls in BASIC), add longer delays, and have a lot more resolution when tweaking times.

(continued next page)

Figure 2 - Moving The Stepper Motor

```
5 ' RANDOM STEP ANGLE GENERATOR FOR STEPPER MOTOR OUT PARALLEL PORT  
10 FOR I = 1 TO 13 : GOSUB 1000 : NEXT : ST = 10  
20 AMOUNT = INT(RND * 20) -10  
25 PRINT AMOUNT, ST, AMOUNT + ST  
30 IF ST + AMOUNT < 0 THEN AMOUNT = ST : FOR I = 1 TO AMOUNT : GOSUB 2000 :  
NEXT : GOTO 20  
40 IF ST + AMOUNT > 20 THEN AMOUNT = 20 - ST : FOR I = 1 TO AMOUNT :  
GOSUB 1000 :  
NEXT : GOTO 20  
50 IF ST + AMOUNT < ST THEN ST = ST + AMOUNT : FOR I = 1 TO AMOUNT :  
GOSUB 2000 : NEXT : GOTO 20  
60 IF ST + AMOUNT > ST THEN ST = ST + AMOUNT : FOR I = 1 TO AMOUNT :  
GOSUB 2000 : NEXT : GOTO 20  
65 IF AMOUNT = 0 THEN 20  
70 PRINT "ERROR"  
80 END  
1000 ' GO FORWARD SUBROUTINE  
1010 OUT &H18, 1  
1020 X = X+1:Y = Y+1 : These kill some time  
1030 OUT &H18, 2  
1040 X = X+1:Y = Y+1  
1050 OUT &H18, 4  
1060 X = X+1:Y=Y+1  
1070 OUT &H18, 8  
1080 X = X+1:Y=Y+1  
1090 RETURN  
2000 ' GO BACK SUBROUTINE  
2010 OUT &H18,4  
2020 X = X+1: Y = Y+1  
2030 OUT &H18,2  
2040 X = X+1:Y=Y+1  
2050 OUT &H18,1  
2060 X=X+1:Y=Y+1  
2070 OUT &H18,8  
2080 X=X+1:Y=Y+1  
2110 RETURN
```

CONTROLLING

(continued from page 25)

I decided to make the motor move a random number of steps, forward or backward, wait a random period of time, then repeat the loop. The BASIC's in Figure 2.

You Have The Power

Now you can drive two stepper motors with your PC (4 lines per motor). This gives you the power to do some kind of X-Y positioning, or linear positioning of two independent objects. Lots of possibilities here.

You can also use the eight lines as independent switches to control 24-volt, low current (depending on the ability of the Darlintons you use, a few amps) devices, or to drive relays to control higher voltages.

Unfortunately, most parallel ports are unable to read their lines, so you can't discover whether switches in the outside world are open or closed, or hook up an analog-to-digital (A/D) converter to read variable voltages. A digital-to-analog converter (DAC) could be connected to give a variable output voltage.

■ ■ ■



MAG TAPE CONVERSION

800/1600 BPI ASCII/EBCDIC

½"-9 Track Data Transfer to Microcomputer Magnetic Media

• PC DOS Diskettes

Raw Data Dump to one Diskette \$50⁰⁰

Batch Service

Setup Fee: \$40⁰⁰

Each Diskette: \$18⁰⁰

• Alternative Media

8" or 5 ¼" Diskette, Hard

Disk or Cassette Tape: Inquire

Informational Brochure Available

MICRO LOGIC SYSTEMS

207 Kent Avenue #1, Kentfield, CA 94904

(415) 461-8077

XT CLONE SYSTEMS

Turbo Mother Board 4.77 and 8 MHz.

640 K Ram installed on board

Serial, Parallel, Game Ports

Clock/Calendar

Color Video Board (CGA)

150 Watt Power Supply

Flip Top Case

AT Style Keyboard

ABOVE WITH 2 FLOPPY DISK DRIVES \$ 699.00

WITH 1 FLOPPY AND 20 MEG \$1049.00

WITH 1 FLOPPY AND 30 MEG \$1199.00

Assembled and Tested for 24 Hours

Color Monitor RGB (CGA)	\$ 310.00
Color Monitor RGB (EGA)	\$ 490.00
Monochrome TTL (Green)	\$ 125.00
Monochrome TTL (Amber)	\$ 135.00
EGA Color Video Card	\$ 195.00
MS DOS 3.1	\$ 50.00

CLONE PARTS

Motherboard Standard 4.77 MHz	\$ 110.00
Motherboard Turbo 4.77 & 8 MHz	\$ 135.00
Color Video Card (CGA)	\$ 65.00
Monochrome Video TTL	\$ 75.00
Floppy Disk Controller	\$ 45.00
Hard Disk Controller	\$ 100.00
Disk I/O	\$ 95.00
RS 232 Serial Card	\$ 45.00
Power Supply 150 Watt	\$ 75.00
Flip Top Case	\$ 45.00
AT Style Keyboard	\$ 65.00

HARD DISK DRIVES

20 Meg Rodime 202E	\$ 249.00
20 Meg Seagate ST4026 (for AT)	\$ 399.00
30 Meg Seagate ST4038 (for AT)	\$ 479.00
30 Meg CDC Wren 1 (for AT)	\$ 479.00

Does NOT include controller

SA800-2 Disk Drive	\$ 39.00
SA850 Disk Drive	\$ 69.00
Mitsubishi M2894 (850 Comp)	\$ 69.00
Call for Shugart Parts	
Cabinet and Power Supply for	
Two Disk Drives	\$ 100.00
Cabinet, PS, and TWO SA850s	\$ 200.00

CASCADE ELECTRONICS, INC.

ROUTE 1 BOX 8

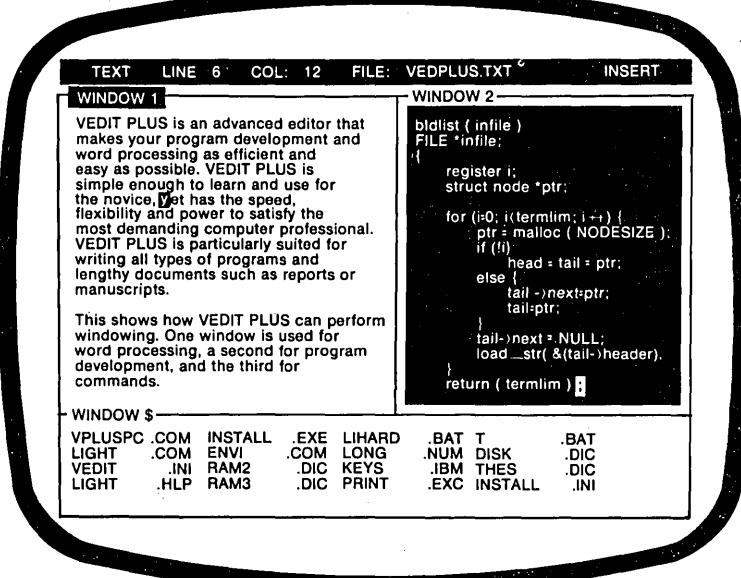
RANDOLPH, MN 55065

507-645-7997

Please ADD Shipping on all Orders

COD Add \$3.00 Credit Cards ADD 5%

Limited to Stock on Hand Subject to change



VEDIT PLUS

#1 CHOICE IN PROGRAMMABLE EDITORS

VEDIT PLUS has been the #1 choice of professionals since 1980. Our latest release is even better - you can open windows to simultaneously edit several files, access many editing functions with pop-up menus, use keystroke macros to speed editing, and run other programs or DOS commands from within VEDIT PLUS.

Whether your needs are program development, technical writing or word processing, VEDIT PLUS is your answer. With over 40,000 users you can depend on VEDIT PLUS to perform consistently and reliably. It is simple enough to learn for the novice, yet has the speed, flexibility and power to satisfy the most demanding professional.

Compare. No other editor is as powerful - unlimited keystroke macros, instant 'off-the-cuff' command macros utilizing a complete programming language, single command file comparison, special word processing and programming features. No other editor is as easy to use - on-line help, pop-up menus, 75 page tutorial, 380 page manual, and VEDIT PLUS is completely customizable.

Fully supports color windows on IBM CGA & EGA, and even windows on most CRT terminals (including CRT's connected to an IBM PC). Available for IBM PC, TI Professional, Tandy 2000, DEC Rainbow, Wang PC, MS-DOS, CP/M-86 and CP/M-80. Order direct or from your dealer. \$185

"To sum things up, VEDIT PLUS is a small, fast, sophisticated editor with a wealth of features and a good macro language. It offers many rewards for the dedicated programmer."

Computer Language, Chris Wolf, Scott Lewis, Mark Gayman 6/86

"VEDIT PLUS is a wholly remarkable program: blindingly fast, extremely powerful, and highly flexible."

Profiles Magazine, Robert Lavenda 4/86

VEDIT PLUS FEATURES

- Simultaneously edit up to 37 files of unlimited size.
- Split the screen into variable sized windows.
- 'Virtual' disk buffering simplifies editing of large files.
- Memory management supports up to 640K.
- Execute DOS commands or other programs.
- MS-DOS pathname and CP/M user number support.
- Horizontal scrolling - edit long lines.
- Flexible 'cut and paste' with 36 text registers.
- Customization - determine your own keyboard layout, create your own editing functions, support any screen size, any CRT.
- Optimized for IBM PC/XT/AT. Also 132 column & up to 70 lines.

EASY TO USE

- Interactive on-line help is user changeable and expandable.
- On-line integer calculator (also algebraic expressions).
- Single key search and global or selective replace.
- Pop-up menus for easy access to many editing functions.
- Keystroke macros speed editing, 'hot keys' for menu functions.

FOR PROGRAMMERS

- Automatic Indent/Indent for 'C', PL/I or PASCAL.
- Match/check nested parentheses, i.e. '{' and '}' for 'C'.
- Automatic conversion to upper case for assembly language labels, opcodes, operands with comments unchanged.
- Optional 8080 to 8086 source code translator.

FOR WRITERS

- Word Wrap and paragraph formatting at adjustable margins.
- Right margin justification.
- Support foreign, graphic and special characters.
- Convert WordStar and mainframe files.
- Print any portion of file; separate printer margins.

MACRO PROGRAMMING LANGUAGE

- 'If-then-else', looping, testing, branching, user prompts keyboard input, 17 bit algebraic expressions, variables.
- CRT emulation within windows, Forms entry.
- Simplifies complex text processing, formatting, conversions and translations.
- Complete TECO capability.
- Free macros: • Full screen file compare/merge • Sort mailing lists • Print Formatter • Main menu

VEDIT and CompuView are registered trademarks of CompuView Products, Inc. MS-DOS is a registered trademark of Microsoft. CP/M is a registered trademark of Digital Research. WordStar is a registered trademark of MicroPro.

CompuView

1955 Pauline Blvd., Ann Arbor, MI 48103 (313) 996-1299, TELEX 701821

Messages From The Board

We've received a lot of calls lately that begin with, "I'm really excited about the PD-32, is it really for real?" Yes, it's for real but you might not know it if you haven't logged into the Micro C RBBS. In fact, the PD-32 has more than doubled the activity on the board. Join the fun; the PD-32 is being shipped right now!

The hottest topic in these parts is the PD32 project introduced last issue in Micro C. Let's take a cruise through the more interesting of the PD32 messages on the Micro C RBBS's PD32 conference.

Once you've logged on (see "The Bulletin Board Blues" in this issue for general information on bulletin board use), type "j 1" to join the PD32 conference. Then an "r 1" reads the first message. Notice that a response of "+" to the "More" prompt will allow reading of successive messages simply by pressing carriage return. Here we go.

CONNECT

Micro Cornucopia Bulletin Board
PCBoard Software - Ver. 10.0 - 01/08/86

System up at 20:37 on 10-21-1986 at 1200 baud.

Please enter C/R [C/R] - no?

Micro Cornucopia Magazine RBBS
300/1200/2400 Baud - 24 Hours/day

Larry Fogg << Sysops >> Margret Rosenberg

What is your first name? jim
What is your last name? davis
Checking user file. Please wait ...
Password (Dots Will Echo)? ...

Good evening Jim. Check your mail again [C/R] - yes? n

(40 min. left) Main Board Command?
j 1

Welcome to the PD32 Conference Jim!
Conference file directories are now active.

View other conference members [C/R] - no?

Check your conference mail [C/R] - yes? n

(40 min. left) PD32 Conference Command? r 1

Date: 09-26-86 (10:13)

To: ALL

From: SYSOP

Subj: DEFINICON BUILDS PD-32S

I received a call from Trevor Marshall that Definicon will be handling commercial manufacture of the PD-32 board. The user group will still be handling kit and board sales to individual hackers.

To contact the user group, leave a message for Dan Efron here.

To contact Definicon, call 818-889-1646, or see their ad in Micro C.

More: [Y]es, [N]o, [NS]non-stop,
[T]hread, [#], [RE]ply? +

Date: 10-06-86 (17:52)

To: ALL

From: DAVID THOMPSON

Subj: EFRON LIVES

I talked to Dan Efron today (10-6-86) and George Scolaro, and I have good news. Dan will be logging onto this board beginning the end of this week. He loaned out his modem but it will be back in a couple of days.

Also, George has boards running. Definicon has several as well. Definicon will be selling the kits and the assembled boards. The agreement Definicon

had with George and Dave is that they will sell the kits for less than what a hobbyist could purchase parts for. That will be \$1,000 for a 10 MHz 2 megabyte kit with UNIX. They expect to be shipping within a month.

Definicon will not be supporting the kit. Dan Efron will handle that via boards such as this one. He will also be talking directly with Definicon.

Dan likes this option because it means he doesn't have to deal with boards and parts. He's really interested in the system and will enjoy the supporting role.

Definicon will sell assembled and tested boards for \$1500. They will provide support for those boards.

The CP/M interface will be available (hopefully) when George gets back from Australia which should be around January 1 1987. Dan will work on the interface boards and George is doing the software.

Even if you are planning to build a CP/M board, you will start with the same board from Definicon. However, I think I'd wait a month or so to see how much other interest there is in a CP/M base, after all, you'll need a hard drive, 20 meg or more, if you're planning to run UNIX.

You can also download the 432.lbr file from this board. It's a tentative 32000 operating system and software that might well be the ticket for you floppy bound souls. (Get in and get your feet wet.) (They're the two greatest feet you got.) David Thompson -- (I'm a friend of the sysops)

Date: 10-07-86 (04:52)

To: DAN EFRON

From: JAMES ULLREY

Subj: PD32 INTEREST ON BMUG

Hello Dan, There is a message on the BMUG RBBS (Berkeley Macintosh Users Group) expressing interest in the 32016. I referred them to the Micro C RBBS, but since you live in Berkeley you might give them a ring. It's 415 849 2684 (data).

Date: 10-10-86 (09:22)
To: ALL
From: SIGI KLUGER
Subj: PD32

To answer some of the questions, the PD32 kit MAY be available without UNIX on a case-by-case basis. We should be able to ship the first kits in about a week or two. All kits will be available in the 10 MHz version only. Expect the price for a kit including UNIX and 1 MB of RAM to be around \$1000. George is currently working on updating the boards so that some of the required jumpers can either be eliminated or shorted.

I will be calling here once in a while to answer questions or check to see if someone needs help, but since I find this system to be very cumbersome to use, don't expect me on too much. Maybe the sysop could make the huge banner at the beginning abortable or clean it away altogether (It's gone. Ed.), or better yet, switch to FIDO.

Date: 10-10-86 (20:11)
To: DAN EFRON
From: JEFF FORBES
Subj: PD32

What is the licensing on the version of UNIX for the PD32? Are educational institution group licenses available or is it a single user-single computer license? Essentially, can one copy of Unix be shared among several machines at the same site?

Date: 10-14-86 (00:38)
To: JEFF FORBES
From: DANIEL EFRON
Subj: PD32

The single-user version licensing agreement of the UNIX we're using absolutely prohibits use on more than one machine. If you need some sort of educational or site license I can talk with Dave Rand about feasibility.

Date: 10-14-86 (00:49)
To: ALL
From: DANIEL EFRON
Subj: PD32 QUESTIONS

I'll be on here regularly about every week (more often if I see the messages piling up). The board looks like it's available now from Definicon, but I want to make sure that the terms are agreeable. I'd like to get some feedback on what type of boards people want to build. 10 Mhz, 2MB or 6Mhz, 1MB ...

Also, I'd like to know who is interested in building one right now. I think Definicon does not want to deal with end users, they want me to handle orders and help. What does everyone think about this and about pricing (i.e. is \$1200 reasonable for 10Mhz, 2MB boards with UNIX).

Date: 10-14-86 (11:00)
To: DANIEL EFRON
From: DANIEL POIROT
Subj: PD32 PC SUPPORT

So it looks like things are getting off the ground on the PD32 project! I got my Oct/Nov issue of Micro C and was thrilled at the PD32 concept. I do have a couple of questions, though.

I went to a National seminar here in Houston a while back and got a complete set of 32000 chips (6MHz I bet) and some documentation. At the time, the National guys didn't have anything to use for a monitor or know of any IBM based software support. Do you know where I can get some help? I would like to get a bare board system up, but without a PROM I really don't know what to do. I would like an assembler that would run on my IBM-PC and some way to download software to the PD32. Is there any more documentation or software for the PD32?

I would be glad to contribute to the project if there is anything more to be done.

Date: 10-15-86 (03:36)
To: DAN EFRON
From: PAT MASON
Subj: PD32

Scott Baker and I have a proto board nearly ready to try. We would like to know if you could leave the PROM hex files and the PAL JEDEC files here?

I am also curious about the version of UNIX; does it include vi? csh? cc, make, sccs, etc.? What about manuals? Are they included or can they be purchased?

How do we go ahead and order UNIX? I'm ready. We should have all the parts and the hardware ready some time next week I hope.

Date: 10-15-86 (21:56)
To: DAN EFRON
From: GEORGE NEUSSE
Subj: PD32

IF I WERE TO BUILD A 6MHZ BOARD WOULD IT SAVE ME ANY CASH? ALSO I WOULD LIKE TO KNOW WHO TO SEND A CHECK TO FOR THE BOARD. DO I SEND IT TO YOU? AND WHEN CAN I SEND THE CHECK? ALSO HOW MUCH SHOULD I WRITE IT FOR?

IT ALSO LOOKS LIKE THE PRICE IS GOING UP A LITTLE. I HAVE A PC CLONE RUNNING AT 8MHZ. CAN THE PD32 RUN IN MY CLONE? IS THERE ANY INTERFACE SOFTWARE WITH THE PD32 PROJECT FOR THE PC. ALL I HAVE BEEN HEARING IS ABOUT CPM AND THE PD32. IF ONLY I COULD GET ONE IN THE MAIL TODAY!!!

Date: 10-18-86 (10:59)
To: ALL
From: JIM WHEATER
Subj: PD32 EAST COAST

Msg # 14 got garbled with line noise, so let's try this. I am the SysOp of The Computer Nookery in New Jersey (12/2400 Baud 24 Hrs 7 Days PCBoard rel 10.0).

I am a programmer (16 Yrs) and am VERY exited about the PD32 project. As soon as I read issue 32 of Micro C, I put a Conference called PD32 in my board. I suspect the East Coast people

(continued next page)

(continued from page 29)

will want to know what's going on, so I want to know if it's OK for me to support the PD32 at this end. If so, I will be calling in here at least once a week (probably would anyway).

My BBS # is 1-201-423-4258 for those who wish to join me on this info exchange.

Date: 10-18-86 (20:17)
 To: DAN EFRON
 From: SAM WARD
 Subj: PD32

I WOULD BE VERY INTERESTED IN PUTTING TOGETHER A BOARD WITH YOUR HELP. THAT WAY I COULD GET THE BOARD UP AND RUNNING, THEN BUY UNIX AT A LATER DATE. KEEP IN TOUCH.

Date: 10-21-86 (21:49)
 To: JIM WHEATER
 From: DANIEL EFRON
 Subj: PD32 EAST COAST

I think it's great that you want to help support the PD32, I look forward to the information your BBS comes up with. Why don't you give me a call here at Berkeley. The number is (415) 643-2270. You can also send me mail at one of the addresses that I will list in a following message.

Date: 10-21-86 (21:52)
 To: SAM WARD
 From: DANIEL EFRON
 Subj: PD32

Right now we're looking into providing bare PCB's with the PROMs and the PALs and maybe memory, but we're waiting to see if there is significant interest. I can provide you with artwork, schematics, PAL and PROM listings if you desire. See a following message for more info on full kit purchases.

Date: 10-21-86 (22:11)
 To: ALL
 From: DANIEL EFRON
 Subj: PD32

Here's the latest scoop on the PD32. I will be sending out more information to people who have written to me (and are therefore on my mailing list).

Definicon has come through with a full kit that can be purchased immediately. There are a few things that one should consider before purchasing. 1) The PD32 purchased from Definicon will only be supported by the PD32 users group. 2) All the hardware is sold basically on an as is basis (although you should expect working parts). 3) Do not buy this system if you are not capable of building it or of having someone else build it for you. 4) To run Unix you'll need at least 20 megs of disk space.

With that said, here are the details: You purchase the board direct from Definicon. You can either call them or write them. Their address and phone number are:

Save BUILD YOUR OWN IBM XT & IBM AT COMPATIBLE SYSTEMS Save

-Why Pay More - Build Your Own With Ease-Have Fun-Save a Fortune-
 Introducing McKnight XT Self-Assembly Kit

Assembled in Less Than 1 Hour with Screw Driver at SUPER LOW COST

- Including 1 Mega Byte PC2 CPU Mother Board w/ 256K, Color Graphic Card or TTL Monochrome Card, Floppy Disk Controller Card, One ½-height DS/DD Drive, Flip-Top Case, 135W Power Supply, Keyboard, Clock Calendar and Game Port.
- AT Mother Board for above \$555
- AT-32 Self-Assembly Kit \$Call
- IBM PC/AT fully compatible

ONLY
\$775.00

HARD DISK

- 20 meg kit \$455
 - 30 meg kit \$555
 - 88 meg kit \$1250
 - Flip-Top For Easy & Quick Access to Inside
- ONLY \$50.00

XT, AT POWER SUPPLY

XT-135W \$75.00 AT-200W \$CALL

XT, AT KEYBOARD

XT-LED for Cap Lock & Num. Lock
 84 Keys
 \$75.00

AT-Same Layout as IBM PC/AT \$75

PC/AT ADD-ON CARD

- AT 1MB memory card \$CALL
- AT 1.5MB Multifunction Card \$CALL
- AT 1.2MB Floppy Disk Controller \$CALL

PC/XT ADD-ON CARD

- PCP-128 Eprom Programmer \$129
- 384 Multifunction Card - Serial, Parallel, Clock, Game, w/Cables & Software - OK \$99
- Mono/Graphic w/Printer Card - Version II Hi-Res (Lotus 1 2 3) \$109
- Color Graphic Card w/Manual \$69
- Monochrome TTL Card 2 left \$50
- 512K RAM Card OK w/Manual \$75
- Floppy Disk Controller w/Cable (handle 4 drives) \$45
- RS232C Card w/Manual \$45
- Parallel Printer Card \$35
- Networks \$CALL

XT, AT MOTHER BOARD

- XT-16 CPU Mother Board - \$199-229
 - IBM PC/XT Fully Compatible, Run all Popular IBM Softwares
 - 8088 Microprocessor w/8087 Optional
 - 8 I/O Slots
 - Up to 1 Mega Byte Memory on Mother Board
 - Fully Assembled & Tested AT-32 CPU MOTHER BOARD
 - IBM PC/AT Fully Compatible
 - 80286 Microprocessor w/80287 Optional
 - 640K Standard, Upgrade to 1 MB on Board
 - On Board Clock Calendar
 - 8 I/O Slots
- \$795**

All Cards Fully Tested, Assembled & Warrantied / School & Institutional P.O. Accepted
 OEM Dealers Welcome - Please call for our Special Dealer Prices

A-OK COMPUTERS

2005 S E 82nd Ste 2 • Portland, Oregon 97216 • 503/771-6758 503/771-6862
 IBM is a trademark of International Business Machines Corporation

Definicon Systems
31324 Via Colinas #108/9
Westlake Village, CA 91362
(818) 889-1646

You must ask for the LOW COST UNIX board. The prices are as follows:

6 MHz, 1 Meg = \$287
6 MHz, 2 Meg = 425
10 MHz, 1 Meg = 473
10 MHz, 2 Meg = 610

That's a little misleading though, everyone MUST purchase Unix from Definicon when they order a board through them. The cost of Unix Sys V.2 is \$500. These prices do not include shipping. You can get second day shipping for about \$20.

The best deal here seems to be the 6Mhz, 2 Meg board. Anyway order to your heart's content. We will probably be providing bare boards with PALs and PROMs. Any interest? You will be able to buy Unix for these separately through the users group.

Please send me your name for more info. We'll be supplying a lot of nice little extras through the users group.

(36 min. left) PD32 Conference Command? g

Time Logged: 7 minutes
Time Used: 7 minutes

Thanks for calling Jim ...



NEW INTEGRATED BIOS FOR BIGBOARD II

This BIOS adds special features to your Bigboard II floppydisk and/or Winchester system.

Read, write and format any 5" and/or 8" disk formats. 40 formats included. Use any type of 5" and/or 8" drive. New SYSGEN works between 5" and 8".

INCLUDES:

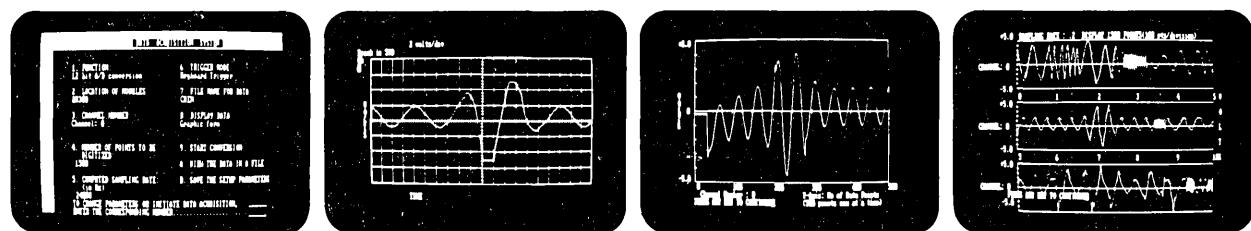
- CONFIguration program that lets you install any new floppy disk format INTERACTIVELY.
- FORMATTER allows you to format almost any diskformat.
- PC-COPY reads and writes PC diskettes on your Bigboard II;
- MONITOR EPROM with serial keyboard and translate table.
- 300 page ZCPR2 manual.
- 60 page BIGBOARD II tech manual.
- BOOTABLE DISK contains free ZCPR2 and P2DOS system. With TIME and DATE stamping. BIOS also has provisions for 256K RAM disk, Centronics, system in ROM.
- WINCHESTER FORMATTER and SYSGEN Supports XEBEC, W-D and ADAPTEC type controllers. Subdivides into any specified number of drives.

Price: \$99.95 (specify disk)

ANDY BAKKERS

de Gervelink 12 • 7591 DT Denekamp
The Netherlands • Tel: 31-5413-2488

Please pay with US\$ Money Order.
MC or VISA welcome.

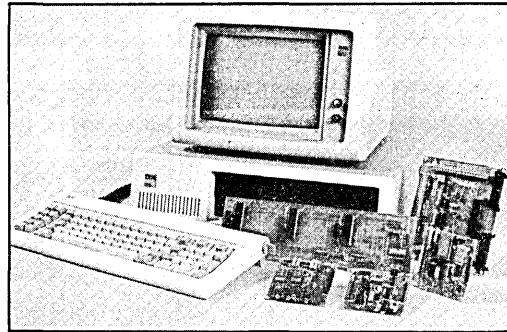


MODULAR DATA ACQUISITION SYSTEM

We Bring Engineers, Scientists And The IBM-PC Together. Our unique high-performance modular data acquisition system allows you to purchase the configuration that exactly meets your requirements. All of Qua Tech's data acquisition products include LABSTAR, powerful software designed to give BASIC programmers total control of their application.

**8 BIT A/D
SYSTEM**

**12 BIT A/D
SYSTEM**



QUA TECH, INC.

478 E. Exchange St. Akron, OH 44304

CALL NOW

(216) 434-3154

FULL POWER DEBUGGING

DSD86, The PC-DOS Debugger 69.95
 DSD87, The PC-DOS Debugger with 8087 Support. 99.95
 DSD80, The CP/M Debugger 125.00



SoftAdvances

P.O. Box 49473 • Austin, Texas 78765 • (512) 478-4763
 1-800-232-8088

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)
 QUANTITY ONE PRICES SHOWN for OCT. 26, 1986

OUTSIDE OKLAHOMA: NO SALES TAX

DYNAMIC RAM

	1000Kx1	100 ns	\$40.00	V20	8MHz	\$10.00
1Mbit	64Kx4	150 ns	4.10	V30	8MHz	\$12.00
4464	256Kx1	100 ns	4.85			
41256	256Kx1	120 ns	3.30			
41256	256Kx1	150 ns	2.75			
41128	128Kx1	150 ns	4.99			
4164	64Kx1	150 ns	1.35			
	EPROM					
27512	64Kx8	250 ns	\$22.00			
27C256	32Kx8	250 ns	6.50			
27256	32Kx8	250 ns	5.25			
27128	16Kx8	250 ns	4.25			
27C64	8Kx8	200 ns	4.75			
2764	8Kx8	250 ns	3.95			
	STATIC RAM					
43256L-12	32Kx8	120 ns	\$23.00	8087	5 MHz	\$125.00
6264LP-15	8Kx8	150 ns	3.05	80287-8	8MHz	\$325.00

OPEN 6½ DAYS, 7 AM-9 PM. SHIP VIA FED-EX ON SAT.

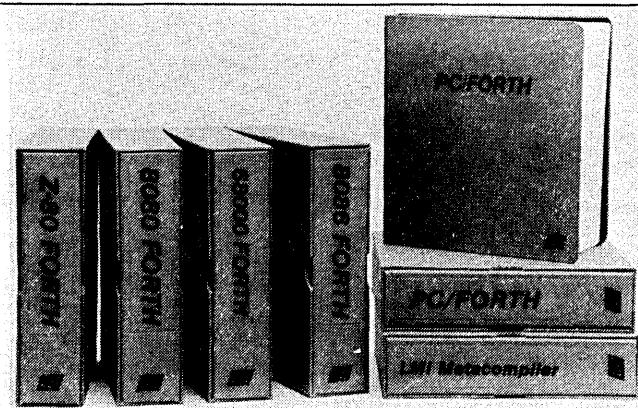
SUNDAYS & HOLIDAYS: SHIPMENT OR DELIVERY, VIA U.S. EXPRESS MAIL

SAT DELIVERY INCLUDED ON FED-EX ORDERS RECEIVED BY:
 Th: Std Air \$6/4 lbs
 Fr: P-One \$13/2 lbs

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts μPoo
 MICROPROCESSORS UNLIMITED, INC.
 24,000 S. Peoria Ave., (918) 267-4961
 BEGGS, OK. 74421
 No minimum order

Please call for current prices because prices are subject to change. Shipping & insurance extra.
 Cash discount prices shown. Orders received by 9 PM CST can usually be delivered to you the next morning, via Federal Express Standard Air (\approx \$6.00, or Priority One (\approx \$13.00)

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: CForth Application Compiler

- 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, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

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, Titisee-Neustadt, 7651-1665
 UK: System Science Ltd., London, 01-248 0962
 France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16
 Japan: Southern Pacific Ltd., Yokohama, 045-314-9514
 Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

IT'S EMERALD MICROWARE FOR EXPANSION AND SUPPORT FOR YOUR CP/M COMPUTER

HARD DISKS FOR YOUR Z80 SYSTEM

by Emerald Microware

If you have been thinking about moving up to a hard drive, why wait any longer? We have everything you need to complete your system, hardware and software, for the Xerox 820, Kaypro, or almost ANY Z80 system.

HDS Host Board — If your computer does not have a hard disk interface, this is the hardware that you need to connect your computer to the WD1002 hard disk controller board.

- Plugs into the Z80 socket, no other wiring required
- Interfaces to the WD1002 controller board
- Switch selectable I/O addressing
- Comes fully assembled and tested

THE WINCHESTER CONNECTION

by MICROCode Consulting

The most comprehensive hard disk software package that you can buy.

- Designed for use with the WD1002 controller board
- Works with one or two hard disks — 5 to 64 meg
- Menu installation, no software to assemble
- Complete testing and error handling
- Automatic swap, warm boot from hard drive
- BIOS drivers install above or below CP/M, your option
- Allows custom partitioning and mixed drives types
- Includes manual, format, test, park, and swap utilities

WD1002-05 HARD DISK CONTROLLER BOARD

by Western Digital

- Standard ST506 drive interface
- Same size as standard 5 1/4 " drive
- 40 pin host interface
- WD2797 floppy disk controller interface on board
- Can control up to three hard drives
- Direct replacement for Kaypro 10 controller

HDS Board with Winchester Connection Software \$ 89.00

WD1002-05 Controller Board. \$185.00

HDS Board, WD1002 Board, and software \$250.00

ACCESSORIES FOR THE KAYPRO AND THE XEROX 820

Xerox 820-2 Main Computer Board.....	\$ 75.00
Xerox 820-2 Floppy Controller board	\$ 65.00
Xerox 820-2 Main board w/Floppy Controller....	\$125.00
Xerox High Profile Keyboard — bare	\$ 25.00
Xerox internal video cable	\$ 8.00
Board mount power connector	\$ 2.50
Parallel ASCII keyboard (not standard Xerox)	\$ 15.00
Dual 5 1/4 " Disk Drives — DSDD 48 TPI, in cabinet with standard Xerox cable	\$265.00
Panasonic DSDD 48 TPI Disk Drives	\$114.00
Panasonic DSQD 96 TPI Disk Drives	\$129.00
Rodime RO252 — 10 Meg — 3 1/2 " Hard Drive	\$275.00
Kaypro 2X Real-time Clock parts kit	\$ 29.00
Kaypro 2X Hard Disk interface parts kit	\$ 16.00
UniForm by Micro Solutions, in stock for Xerox 820, Kaypro, Morrow, Bondwell, and IBM-PC	\$ 64.95

Prices subject to change without notice. Include \$4.00 shipping and handling, \$7.00 for COD. Phone hours: 8:00 am to 5:00 pm weekdays; check our bulletin board — RQP/M, 5:30 pm to 8:00 am seven days a week, for our latest stock and prices. 30 day money back guarantee on all products.

THE KayPLUS ROM PACKAGE

by MICROCode Consulting

The most important element in the performance of your Kaypro computer is its monitor rom. With KayPLUS you get all of the advantages of a Kaypro 4 or 10, even on your Kaypro 2.

- No software assembly required
- Install up to four floppies and two hard drives
- Boots from floppy or hard disk
- Supports 96 TPI, 48 TPI, and 3 1/2 " disk drives
- 32 character keyboard buffer
- Automatic screen blanking (not avail. on 83 series)
- 12 disk formats built-in
- Full automatic disk relogging with QP/M
- Internal real-time clock and ram card support
- Includes manual, format, configuration, diagnostics, sys-gen, AND hard disk utilities
- Available for '83 and '84 series Kaypros

KayPLUS ROM Set. \$ 69.95

KayPLUS ROM Set with QP/M *** SPECIAL *** \$115.00

QP/M by MICROCode Consulting

Full CP/M 2.2 compatibility with many more features. You've seen the replacements that eat up memory and need auxiliary programs to run. Not QP/M. Fifteen internal commands, automatic disk relogging (no more control C), user area selection from colon, 31 user areas, drive search path, archive bit maintenance, and transparent time/date stamping, all in the same space as CP/M 2.2. Installs from a convenient customization menu, without any software to assemble. Bootable systems available for Kaypro, Xerox 820 and Big Board.

QP/M Operating System, complete bootable copy \$ 80.00

QP/M without BIOS. \$ 60.00

PLUS2 ROM by MICROCode Consulting

X120 DOUBLE DENSITY BOARD by Emerald Microware

Clearly the most versatile double density package for the Xerox 820-1. Run up to four floppy disk drives at once, both 8" and 5 1/4 " at the same time. Software compatible with Kaypro and Xerox 820. Supports all standard printers, and most add ons like the Ferguson Ram Board. You get mini-monitor functions, autoboot capability, 19 built in disk formats, and bank mode ROM operation for more space in your TPA. Lets you run 48 TPI disks on 96 track drives. Works with Uniform and QP/M.

PLUS2 ROM Set and X120 Board A&T \$ 135.00

PLUS2 ROM Set and X120 Bare Board \$ 62.00

PLUS2 ROM Set only \$ 49.95

120 Bare Board only *** SPECIAL *** \$ 15.00

Other kits, parts, and packages available

Service and parts available for Kaypro, Xerox, BBI, and others. Call for information.



P.O. Box 1726, Beaverton, OR 97075 • (503) 641-0347

Selecting Subsets From Sets

The Formula For Combinations In PROLOG

In this short paper, Ramachandran takes on sets, statistics, combinations, recursion, and PROLOG, all (and more) in just 6K of text — a pretty fair combination in itself. Problem lovers, gather 'round.

In many decision-making situations, we need to know how many smaller groups or subsets of a given size can be selected from a larger group of objects.

In statistics, we draw samples of a particular size from the bigger group we want to study. We then use statistical formulas to make realistic estimates about, for example, the average income of the entire population of a country by determining the average income of a sample. These formulas are based, among other things, on knowing how many possible samples could have been selected.

Math Review

$N!$ is the factorial function ($N * (N-1) * (N-2) * \dots * 1$). (For example, $6! = 6 * 5 * 4 * 3 * 2 * 1 = 720$.)

And a standard formula to find the number of combinations is:

$$n! / (r! * ((n-r)!))$$

where n is the number of objects in the large group from which the selection is made, and r is the number of objects to be selected.

For example, the number of subsets of size 3 that can be selected from a group of 10 is:

$$10! / (3! * ((10-3)!))$$

or 120.

The standard notation for expressing the number of ways of selecting r objects from a set of n is:

$$C(n,r).$$

read — "Number of COMBINATIONS of r from n ".

A Recursive Algorithm

Consider the number of ways in which a subgroup of r objects can be selected from a larger group of n objects. If one of the objects is A, then all selections fall into one of two distinct categories — those which include A, and those which don't.

All the selections which include A consist of A along with $r-1$ objects selected from the remaining $n-1$ objects of the bigger group. So, the number of such selections must be:

$$C(n-1,r-1).$$

And all the selections which do not include A must consist of r objects selected from the $n-1$ objects left when A is excluded. So the number of such selections must be $C(n-1,r)$.

We arrive at the equation:

$$C(n,r) = C(n-1,r-1) + C(n-1,r)$$

a definition of something in terms of a simpler version of itself.

One problem now equals two simpler problems. The first tells us to consider choosing one less object from a group with one less in it, and the second tells us to select the same number from a group with one less in it.

By Ramachandran Bharath
Dept Mgmt, Mktg & Computer Information
Northern Michigan University
Marquette, MI 49855

Each of these subproblems can be changed to two even simpler problems by the same technique, so we now have a recursive algorithm. But it can't go on forever.

In one of our subproblems, we'll sooner or later subdivide until we're considering choosing 0 objects. Obviously, the number of ways of choosing 0 objects from a group is just one — leave everything out!

In the other subproblem, by repeating the process, we'll come to a stage where the size of the group shrinks to the same number as the number of objects to be selected. So there's obviously only one way of selecting — select all the objects!

Recursion Boundaries

The conditions of a recursive definition when the solution becomes obvious are called the BOUNDARY CONDITIONS of the problem. In this example, the boundary conditions are:

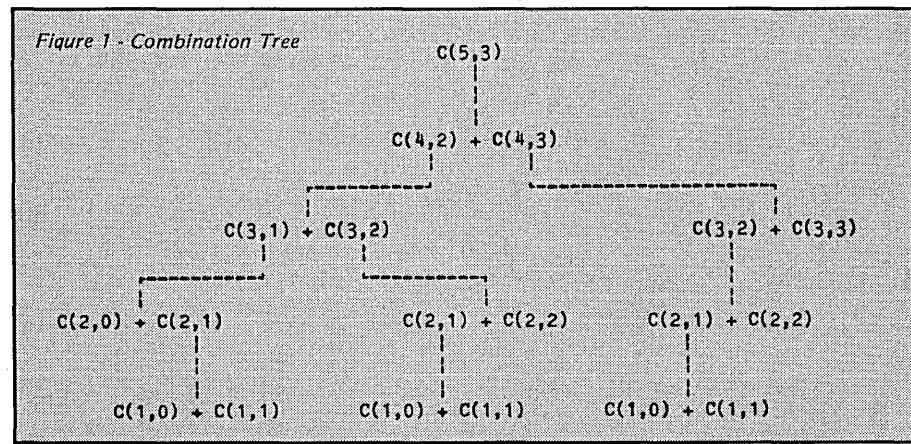
$$C(\text{Anynumber},0) = 1$$

$$C(\text{Same},\text{Same}) = 1$$

For example, if we want to know $C(5,3)$, we can logically say:

$$C(5,3) = C(4,2) + C(4,3).$$

But that raises the question — What is $C(4,2)$?



By the same argument, $C(4,2) = C(3,1) + C(3,2)$. Similarly, we can apply this argument to $C(4,3)$ as well. The result of this recursive process is shown diagrammatically in Figure 1. We see that as we keep "recurring", we reach one or the other of the boundary conditions:

$C(\text{Anynumber}, 0)$
 $C(\text{Same}, \text{Same})$

$$So C(2,1) = C(1,0) + C(1,1) = 1 + 1 = 2.$$

And moving back up the tree to $C(3,1)$:

$$C(3,1) = C(2,0) + C(2,1) = 1 + 2 = 3.$$

Similarly, we can see that $C(3,2)$ is 3, and $C(4,2)$ is $3 + 3 = 6$.

And on the right branch, $C(4,3) = 4$, and (finally) $C(5,3) = 6 + 4 = 10$.

PROLOG

This kind of recursive calculation can be expressed very concisely in PROLOG:

OG. In the three PROLOG clauses in Figure 2, notice that the first two clauses are the boundary conditions and the third is the recursive reduction to simpler conditions. Try it with questions like:

?-combs(5,5,Howmany).
?-combs(7,0,Howmany).
?-combs(8,3,Howmany).

Figure 2 - Recursive Solution

```
combs(Anynumber,0,1).
combs(Same,Same,1).

combs(Group,Wanted,Result) :-  

Groupless is Group - 1,  

Wantless is Wanted - 1,  

combs(Groupless,Wantless,Leftbranch),  

combs(Groupless,Wanted,Rightbranch),  

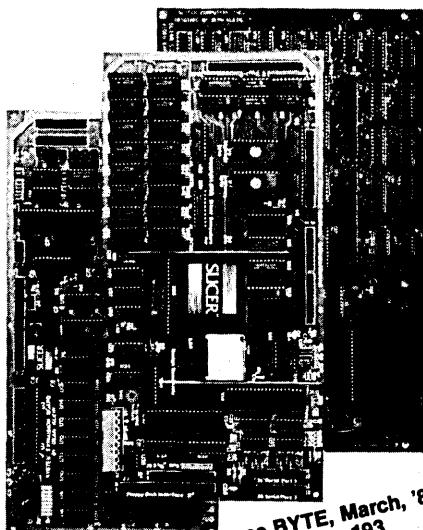
Result is Leftbranch + Rightbranch.
```

Wrap Up

I tested this program in PDPROLOG (written by Robert Morein at A.D.A. and available on Micro C MS-Disk 19).



SLICER... THE TRUE LOW COST MULTI-USER MICRO



see BYTE, March, '85
page 193

THE SLICER Real 16 Bit Power on a Single Board— Featuring the Intel 80186 (C) Step CPU

- Complete 8 MHz 16-bit microprocessor on a 6" x 12" board
 - 256K RAM, plus up to 64K EPROM
 - SASI port for hard disk controller
 - Two full function RS232C serial ports with individually programmed transmission rates—50 to 38.4K baud
 - Software compatibility with the 8086 and 8088.
 - 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
 - Software supports most types and sizes of disk drives
 - Source for monitor included on disk
 - Bios supports Xebec 1410 and Western Digital WD 1002 SHD controller for hard disks
- Fully assembled and tested only \$445

THE SLICER 1-MByte EXPANSION BOARD NEW

For expanded memory, additional ports, and real time clock

- 1-MByte additional dynamic RAM
 - 2 RS232C asynchronous ports with baud rates to 38.4K for serial communication
 - 2 additional serial ports for asynch (RS 232) or synch (Zilog 8530 SCC) communication
 - Real Time Clock with battery backup
 - Centronics type parallel printer port
- Fully assembled and tested only \$450

SLICER/1MByte EXPANSION COMBO The Slicer (without RAM or RAM controllers) with Full 1-MByte NEW Memory Expansion

CCP/M (Digital Research) \$950

THE SLICER PC EXPANSION BOARD Gives your Slicer high performance video capability

- IBM compatible monochrome video
- Video memory provides 8 pages of text or special graphics capability
- 2 IBM type card slots for color video, I/O expansion, etc.
- IBM type keyboard port

Fully assembled and tested only \$495

All boards available in kit forms
The SLICER Bulletin Board (300/1200 baud)
612/788-5909

Runs MS DOS generic software; PC DOS program operation not guaranteed

Also available: The μSlicer 188 \$400;
8087 Math Co-Processor Bd. (call); 10 MB Hard Disk \$465; W.D. 1002-SHD H.D.C. Bd. \$200; Enclosures, Power Supply, and Support Hardware.

CP/M 86 \$85, CCP/M \$250 (Digital Research, Inc.); MS DOS \$175 (Microsoft Corp.)

MasterCard, Visa, Check, Money Order, or C.O.D.
Allow four weeks for delivery. Prices subject to change without notice.

QUALIFIED DEALER INQUIRIES INVITED

SLICER Computers Inc.
2543 Marshall St. N.E.
Minneapolis, MN 55418
612/788-9481
Telex 501357
SLICER UD

Introduction To Fractals

By Larry Fogg
Micro C Staff

Math Can Be Beautiful

Larry was staring intently at his screen. At first glance it appeared that something was working its way haphazardly across his screen. On closer look I could see that it was leaving a trail so detailed that when I stepped back it lost all trailness, becoming a solid design with holes.

"Fractals," he said.

That was my first introduction to the beasties. I understand that before computers, mathefracticians (or whatever they're called) used to create them with paper and pencil. Drawing and erasing over and over as they worked their way towards tinier and tinier detail (these same folks discovered Life's gliders with pencil and Pink Pearl eraser). Who says computers aren't essential?

A few years back I got my first computer. It was an Apple II that had been run over by a fork lift. After much fixing and cursing I got it up and running. One of the first programs that resurrected Apple ran was a Hilbert curve generator.

Even on such a primitive computer, I enjoyed watching the curve slowly trace its way around the screen. This was my first taste of fractals.

What's A Fractal Anyway?

Consider a sheet of paper. Herr Entsminger's "Last Page" perhaps. Surely a flatter object does not exist. But after reading the page, if we begin to crumple it up in a fit of disgust, the two dimensional surface becomes more and more irregular. It eventually becomes a space filling wad and looks more like a three dimensional object. It now has a fractal dimension between two and three.

A one dimensional line shows the same behavior as it becomes more irregular. A line can be made to pass through every point on a surface and

so become two dimensional. The Hilbert curve (see Figure 1) is an example of this kind of plane filling curve.

Defining fractals is an iffy proposition. Even the father of fractals, Dr. Benoit Mandelbrot, has said that it would be best not to bother. It makes more sense to try to get an intuitive feel for fractals and state some of their properties.

Think about the common garden variety circle. Looking at successively smaller segments of the circle you'll end up with a straight line. There is no new detail at the smaller scales.

Unlike the circle, fractals are irregular geometric objects which show greater and greater detail no matter how small the scale. Many fractals also exhibit self similarity. That is, the detail at small scales looks just like the detail at larger scales.

The Length Of A Fractal Curve

The coastline of England provides the classic example of a fractal curve in nature. How would you measure the length of the coast? You could get a first approximation by stepping around a map with a pair of calipers, but think of all the details you'd miss. A much better (and larger) answer would come from strapping on a pedometer and walking the coastline.

As you're enjoying the walking tour, look at a one-step section of the coast. Aha! More detail. The point is that as you consider ever smaller details of the coastline, there will always be more details within the details and the length will grow without bound. (If it sounds to you as though some recursion is lurking around the corner, you're right.)

A Swipe At Mathematics

It's time to offend the mathematicians out there. The English coastline problem illustrates perfectly the trouble you can get into by applying mathematics to the real world.

Mathematics is not reality. Mathematics models reality. It is purely a construct: an elegant system devised by people for their own enjoyment and to help understand certain aspects of physical reality.

To say that the coastline becomes infinite ignores the fact that there is no infinitely small piece of matter and therefore no infinitely small step along the coast. Matter at the smallest scale becomes energy and is next to impossible to measure with a ruler. So it doesn't make sense physically to talk about an infinite coastline.

However, fractal geometry does remain one of the most fascinating and useful areas of mathematics. Fractals are used to model natural phenomena such as turbulent flow, interactions of plant communities, Brownian motion, and the distribution of stars. Fractals can even describe the activity of the stock market (a very unnatural phenomenon). And remember the new world created in Star Trek II? Fractals again.

Figure 1 - Hilbert Curve

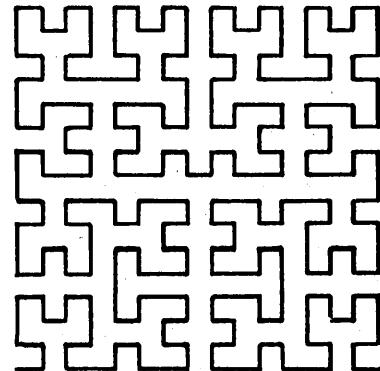
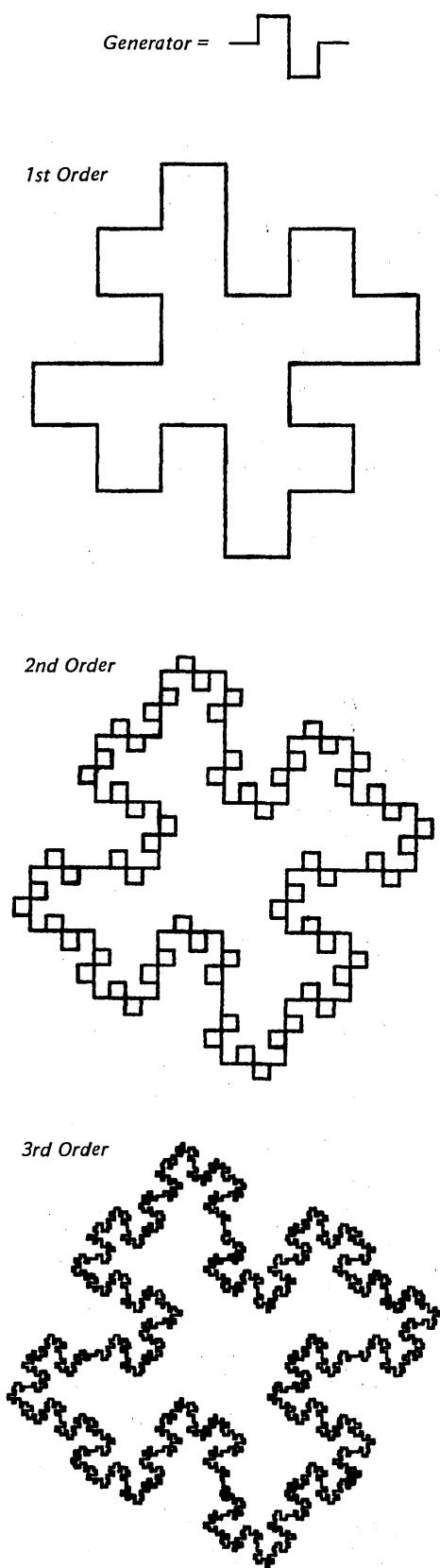


Figure 2 - Mandelbrot's Square Snowflakes



Geometric Fractals

This class of fractals involves repetition of exactly the same pattern in progressively smaller scale. Mandelbrot's square snowflake (see Figure 2) is a good example. A generator replaces each side of the square initiator. Each segment of the new shape is then replaced by a scaled down copy of the generator. And each segment of that shape is replaced by still smaller generators and on and on ...

An interesting property of this generator is that each iteration removes exactly the same area as is added. The area bounded by the snowflake remains constant while the perimeter continues to increase. So we end up with a finite area bounded by an infinitely long perimeter.

Figure 3 shows recursion coming to the rescue. This partial listing forms the basis for FRACTAL.PAS, which resides on the Micro C bulletin board in the CURRENT ISSUE LISTINGS area.

The heart of the program is the procedure Generate. This routine calls

itself to overlay the generator on successively smaller line segments until the segments are shorter than a given resolution. Then we draw the last set of generators. Nothin' to it.

More To Come

We've barely scratched the surface of fractal geometry. Exploration between integer dimensions is virtually unlimited. For example, just as objects exist with fractal dimensions between two and three, they also exist with dimensions greater than three. Three dimensional "slices" of these beasties create truly bizarre pictures.

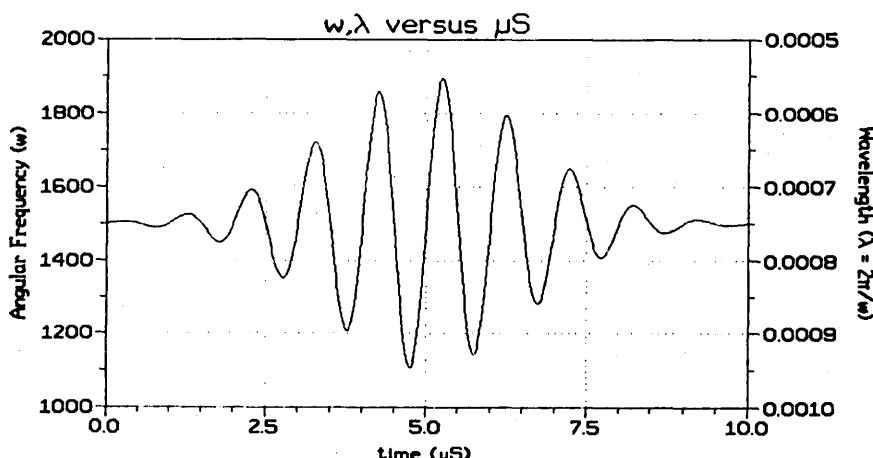
I'll be looking into some of the more interesting applications of fractals including fractal music. Stay tuned for the results. Until then, grab yourself a copy of FRACTAL and have some fun.



(Figure 3 begins on page 38)

GRAPHICS

starting at \$49⁹⁵



SCI-GRAF: Produces line, scatter, and high/low plots on Epson or IBM compatible dot-matrix graphics printers. Supports images up to 1680x1712 dots (over 3 million pixels), log & linear scales, graphs in 7 colors (on a JX-80), batch processing. Requires MS-DOS 2 or 3, 256k ... \$99^s

FONTEdit: Easy-to-use full screen font editor allows you to create and modify fonts for our SCI-GRAF program. Create Greek, math, and custom symbols in 3 sizes. Supports SCI-GRAF's proportional spacing. Requires IBM compatible 320x200 pixel on-screen graphics & IBM compatible keyboard \$39^s

GRAF 3.0: Produces bar, pie, line, scatter, and high/low plots on Epson, IBM, C.Itoh, or NEC compatible dot-matrix graphics printers. Supports 14 fill-in patterns, 8 point-plotting symbols, automatic scaling, labeling, and legend creation. MS-DOS 2 or 3, 192k \$69^s CP/M-80: Requires 64k (54k TPA) \$49^s

MSC

Microcomputer
Systems
Consultants

32 West Anapamu Suite 190 Santa Barbara CA 93101 (805) 963-3412

Figure 3 - Fractals Listing

PROGRAM Fractal; { generates geometric fractal curves - }

The pattern below is used to define the generator. The endpoints of the line segment being overlayed are 11 and 15.

```
* 1  
* 6   * 2  
* 11  * 7   * 3  
* 16  * 12  * 8   * 4  
* 21  * 17  * 13  * 9   * 5  
* 22  * 18  * 14  * 10  
* 23  * 19  * 15  
* 24  * 20  
* 25
```

This program was written using Turbo Graphics Toolbox from Borland. If you don't have this software, all you need to write is a procedure to draw a line }

```
{$I TYPEDEF.SYS}           { Turbo Graphics Toolbox files }  
{$I GRAPHIX.SYS}  
{$I KERNEL.SYS}  
  
CONST  
  MaxGenSize = 15; { generator limit }  
  MaxInitSize = 10; { initiator limit }  
  
TYPE  
  LineRec = RECORD  
    X1, Y1, X2, Y2: Real;  
  END;  
  PointRec = RECORD  
    X, Y: Real;  
  END;  
  LineAry = ARRAY [1..MaxInitSize] OF LineRec;  
  PositionAry = ARRAY [0..MaxGenSize] OF Integer;  
  PointAry = ARRAY [1..25] OF PointRec;  
  
VAR  
  I,  
  InitiatorSize,  
  GeneratorSize,  
  Resolution: Integer;  
  Initiator: LineAry;  
  Generator: PositionAry;  
  
PROCEDURE GetInitiator (VAR Init: LineAry; VAR InitSize: Integer);  
{ returns an array of line records defining the initiator }  
END; { GetInitiator }  
  
PROCEDURE GetGenerator (VAR Gen: PositionAry; VAR GenSize: Integer);  
{ returns an array of integers defining the generator according to the }
```

```
pattern shown at the head of this file. }  
END; { GetGenerator }
```

```
PROCEDURE GetResolution (VAR Res: Integer);  
{ get Resolution from user - allow program exit with entry of 0 }  
END; { GetResolution }
```

```
FUNCTION Length (Lion: LineRec): Real; { determines line length }  
BEGIN  
  WITH Lion DO  
    Length := Sqr (Sqr (X2 - X1) + Sqr (Y2 - Y1));  
END; { Length }
```

```
PROCEDURE FindMidPoint (FirstX, FirstY, LastX, LastY: Real;  
                        VAR MidX, MidY: Real);  
{ finds coordinates of midpoint of line segment }  
BEGIN  
  IF (LastX >= FirstX)  
    THEN  
      MidX := FirstX + (LastX - FirstX)/2  
    ELSE  
      MidX := FirstX - (FirstX - LastX)/2;  
  IF (LastY >= FirstY)  
    THEN  
      MidY := FirstY + (LastY - FirstY)/2  
    ELSE  
      MidY := FirstY - (FirstY - LastY)/2;  
END; { FindMidPoint }
```

```
PROCEDURE FindPoints (Lne: LineRec; VAR Pts: PointArry);  
{ finds screen coordinates of only those points used by the generator }  
VAR  
  I: Integer;  
  a,b: Real; { used to find points not directly on Lne }  
BEGIN  
  WITH Lne DO  
    BEGIN  
      FindMidPoint (X1, Y1, X2, Y2, Pts [13].X, Pts [13].Y);  
      FindMidPoint (X1, Y1, Pts [13].X, Pts [13].Y, Pts [12].X, Pts [12].Y);  
      FindMidPoint (Pts [13].X, Pts [13].Y, X2, Y2, Pts [14].X, Pts [14].Y);  
      a := Pts [12].Y - Y1;  
      b := Pts [12].X - X1;  
      FOR I := 1 TO GeneratorSize DO  
        CASE Generator [I] OF  
          1: BEGIN  
            Pts [Generator [I]].X := X1 + 2*a;  
            Pts [Generator [I]].Y := Y1 - 2*b;  
          END;  
          2: BEGIN  
            Pts [Generator [I]].X := Pts [12].X + 2*a;  
            Pts [Generator [I]].Y := Pts [12].Y - 2*b;  
          END;  
          { you get the idea ... }  
          25: BEGIN  
            Pts [Generator [I]].X := X2 - 2*a;  
            Pts [Generator [I]].Y := Y2 + 2*b;  
          END;  
        END; { CASE .. Finally! }  
    END; { WITH }  
END; { FindPoints }
```

(Fractals Listing continued next page)

Fractals Listing

(continued from page 39)

```
PROCEDURE DrawCurve (Pts: PointArry; Lne: LineRec);
VAR
  I: Integer;
BEGIN
  FOR I := 1 TO (GeneratorSize - 1) DO
    DrawLine (Pts [Generator [I]].X, Pts [Generator [I]].Y,
              Pts [Generator [I + 1]].X, Pts [Generator [I + 1]].Y);
END; { DrawCurve }

PROCEDURE Generate (Lyne: LineRec);
{ recursive procedure to generate the curve }
VAR
  J: Integer;
  Segment: LineRec;
  Points: PointArry;
BEGIN
  FindPoints (Lyne, Points); { find coords of generator overlayed on Lyne }
  IF Length (Lyne) > Resolution { this condition limits the recursion }
    THEN
      BEGIN
        FOR J := 1 TO (GeneratorSize - 1) DO { for each segment of the }
          BEGIN { generator, overlay another }
            WITH Segment DO { smaller generator }
              BEGIN
                X1 := Points [Generator [J]].X;
                Y1 := Points [Generator [J]].Y;
                X2 := Points [Generator [J + 1]].X;
                Y2 := Points [Generator [J + 1]].Y;
              END; { WITH }
            { DrawCurve here too if you want to see all orders of the curve }
            Generate (Segment);
          END; { FOR }
      END { THEN }
    ELSE
      DrawCurve (Points, Lyne);
  END; { Generate }

BEGIN { MAIN }
  InitGraphic; { these 4 lines set up Turbo Graphics }
  DefineWorld (1, 0, 0, 1000, 1000); { play with 1000,1000 to set aspect ratio }
  SelectWorld (1);
  SelectWindow (1);
  GetInitiator (Initiator, InitiatorSize); { enter 0 to exit program }
  WHILE InitiatorSize > 0 DO
    BEGIN
      GetGenerator (Generator, GeneratorSize);
      GetResolution (Resolution); { enter 0 to back out }
      WHILE Resolution <> 0 DO
        BEGIN
          FOR I := 1 TO InitiatorSize DO
            Generate (Initiator [I]);
          REPEAT UNTIL KeyPressed; { keep screen until a key is pressed }
          GetResolution (Resolution);
        END; { WHILE Resolution <> 0 }
      GetInitiator (Initiator, InitiatorSize);
    END; { WHILE InitiatorSize <> 0 }
  LeaveGraphic; { restore text mode }
END.
```

FORTRAN a structured language? Here's How!

New RF-77...a Ratfor-like preprocessor for PC FORTRAN

- WHILE, FOR and REPEAT loops.
- Free form input. No column 1, 6, & 7 to worry about.
- DEFINED constants and INCLUDED files.
- Outputs standard ASCII source code. May be compiled by any Fortran 77 compiler.
- Runs on any MS-DOS/PC-DOS equipped personal computer.

\$65.

Postage paid.
Texas residents add \$3.98 tax.



**Logical
Developments**

P.O. Box 55798, Houston, Tx 77255

For VISA/MasterCard orders call:
1-800-835-2246, ext. 41

C CODE FOR THE PC

source code, of course

GraphiC 3.0 hi-res color plots	\$300
Panache C Program Generator	\$125
QC88 C Compiler	\$90
Concurrent C	\$45
Coder's Prolog in C	\$45
Translate Rules to C	\$30
LEX	\$25
YACC & PREP	\$25
tiny-c interpreter & shell	\$20
Xlisp 1.5a & tiny-Prolog	\$20
C Tools	\$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409
(512) 258-0785

Free shipping on prepaid orders

No credit cards

EMERALD MICROWARE - YOUR IBM DISK SPECIALISTS

UniForm-PC by Micro Solutions

This program allows you to read, write, and format diskettes for over a hundred CP/M and MSDOS computers, on your PC, XT, or AT. Once installed, UniForm lives in the background while you use your standard DOS commands and other programs to operate on the files on your disk. Copy files between CP/M, TRSDOS, and MSDOS, or work on the files directly on original diskettes.

UniForm-PC \$ 64.95

UniDOS by Micro Solutions

UniDOS allows you to actually RUN your favorite 8080 code CP/M programs on your IBM. Use UniDOS with UniForm-PC, and you can run them directly from your CP/M format diskettes. UniDOS uses the 8080 mode of the NEC V20 and V30 CPU chips, or can run in emulation mode for Z80 programs, and machines still using the 8088. All standard CP/M system calls are supported.

UniDOS \$ 64.95

UniDOS w/UniForm & V20 chip \$129.95

CompatiCard by Micro Solutions

Finally a card that allows you to run 8", 5 1/4", and 3 1/2" disk drives on your IBM or compatible, all at the same time. This board works with the UniForm-PC program to format, read, and write literally hundreds of CP/M and MSDOS disk formats on your PC. Use this card to replace, or in addition to your existing floppy controller card.

CompatiCard \$169.95

MatchPoint-PC by Micro Solutions

This half-size card allows you to read and write to Apple II and NorthStar diskettes on your IBM. INCLUDES a copy of the UniForm-PC program, as well as utilities to format disks, copy, delete, and view files on Apple DOS and Apple CP/M diskettes.

MatchPoint-PC \$169.95

COPY II PC and the COPY II PC OPTION BOARD

by Central Point Software

Have you ever accidentally destroyed one of your important copy-protected disks, or worried that you might? Now you can back up those important disks using COPY II PC and forget about those worries. The COPY II PC program can back up most copy protected disks without any additional hardware. The COPY II PC Option Board can back up almost ALL copy protected diskettes including the more complicated mechanical schemes.

COPY II PC OPTION BOARD \$ 89.95

COPY II PC program only \$ 34.95

PC TOOLS utility program \$ 34.95

Adaptec 2070 - RLL Hard Disk Controller Card

Adaptec is the name for speed and quality in hard disk controller boards. The Adaptec 2070 controller board uses the new Run Length Limited technology to squeeze an additional 50% more capacity on a conventional hard disk drive. For example, a 20 Meg MicroScience HH325 can operate with a capacity of 30 Megabytes in RLL mode. Works in both RLL and standard encoding modes. Please call or write for compatibility list, as RLL mode works with selected brands of hard disk drives.

Adaptec 2070 \$136.00

Hard Disk Controller Cards by National Computer Ltd.

This line represents high quality with low cost in standard HALF CARD hard disk controller boards. Versions available for XT and AT, some with floppy controllers installed.

HDC 5127 for XT \$105.00

HDC 5126 for AT \$169.00

HDC 5125 for AT w/Floppy Controller \$195.00

Prices subject to change without notice. Include \$4.00 shipping and handling, \$7.00 for COD. Phone hours: 8:00 am to 5:00 pm weekdays; check our bulletin board — RQP/M, 5:30 pm to 8:00 am seven days a week, for our latest stock and prices. 30 day money back guarantee on all products.

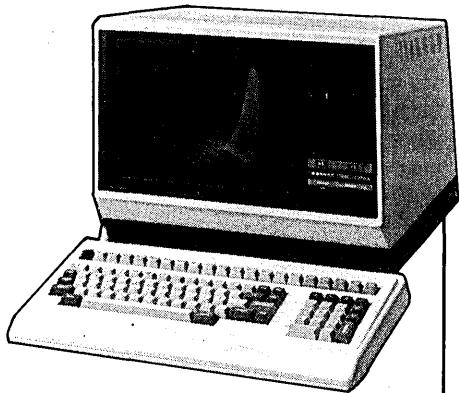
**EMERALD
MICROWARE**

P.O. Box 1726, Beaverton, OR 97075 • (503) 641-0347



SANYO

Prices don't get any lower.



SANYO

MBC 1200

FREE SOFTWARE FROM MICROPRO

- Thousands of FREE public domain software available
- COMPAT disk utility for over 50 CP/M formats—only \$39
- Professional accounting software available
- 20 meg harddrive optional

\$399

Special pricing for printers

MBC 1200 SPECIFICATIONS:

- Two Z-80A (main and subsidiary) CPUs with no-wait mode for fast execution, substantial memory capacity (RAM 64KB, ROM 4KB).
- High-resolution full graphic function with 640 x 400 dot matrix display.
- Choice of 33 or 40 line text mode.
- CP/M operating system complete with assembler, editor and all utilities.
- Easy-to-use Sanyo graphic BASIC.
- One (MBC 1200) or two (MBC 1250) internal double-sided, double-density, double-track, 5 1/4" slim-type 640KB formatted mini floppy disk drives.
- Keyboard with 15 user-programmable function keys.
- Interfaces for Centronics printer and one RS-232C port provided.
- Free software from Micropro: Wordstar, Mailmerge, Spellstar, Infostar, Calcstar & Basic.

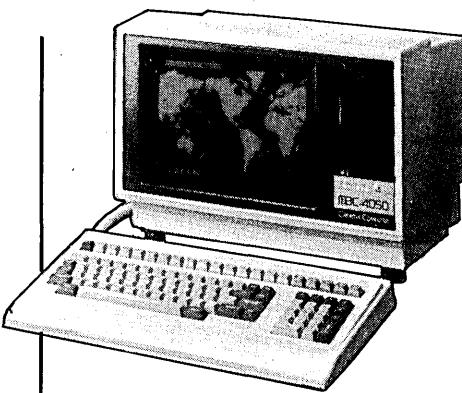
MBC 4000 SPECIFICATIONS:

- MS DOS - CP/M 86 operating system with editor, assembler and all standard utilities.
- 128KB RAM memory capacity, expandable to 640KB.
- Interfaces for one Centronics printer and one RS-232C port.
- 12" no-glare green monitor display screen for easy viewing.
- One (MBC 4000) or two (MBC 4050) internal double-sided, double-density, double-track, 5 1/4" slim-type mini floppy disk drives with 640KB formatted capacity.
- Keyboard with 15 user-programmable function keys.
- Free software from Micropro: Wordstar, Mailmerge, Spellstar, Infostar, Calcstar & Basic.

EDUCATIONAL DISCOUNTS

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.



SANYO

MBC 4000

FREE SOFTWARE FROM MICROPRO

- 8086 Microprocessor
- Runs MS DOS & CP/M 86
- Floppy disk drive with 640K capacity
- RAM expansion to 640K available
- Utility pack including BIOS & schematics option

\$499

Special pricing for printers

MSO
MICRO
SUPPLY
ORGANIZATION INC.

For our catalog with complete
details and prices, send \$2 to:

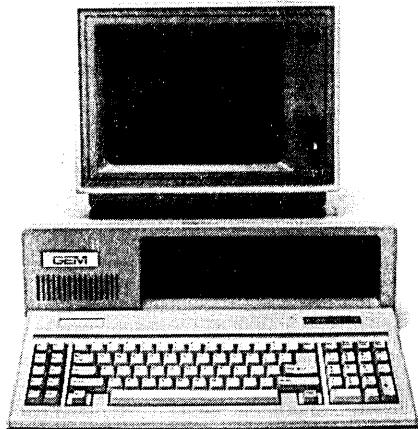
Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

15% Restocking on Returned Orders

805/393-2247



MAXIMUM PERFORMANCE.



GEM-ST
Don't be fooled.
No hidden cost!

In keeping with industry trends MSO is bringing our customers high performance P.C. compatibles and accessories. The GEM-ST™ features the V20-8 chip which runs at three times the speed of the IBM PC XT* and also runs CPM 8080 software.

**FREE 5 module
integrated software**

\$1199

**22 meg sub-system
with tape back-up
\$599 w/purchase**

GEM-ST Specifications:

CPU - V20-8 Chip & Run CPM
Software
RAM - 640K
Clock Rate - 4.77 or 8 MHZ
Disk Interface and Drives -
 (2) - 5-1/4" floppy drives 360K
ROM BIOS - Legal BIOS
Keyboard - AT Style Keyboard
Expansion Slots - 8 IBM* compatible slots
Power Supply - DC Voltage and Max Current
 + 5V @ 15A - 5V @ .5A
 + 12V @ 6A - 12V @ .5A
 Input Voltage 115/230V
 Meets UL/FCC Standards
Standard Interface - (1) - Serial RS-232-Port
 (1) - Game Port
 (1) - Parallel Printer Port
Clock - Clock calendar battery backed up
Video - TTL Hi-Res Monographics Card
 and TTL Monitor
Operating System - MS-DOS
Warranty: 1 year all parts.
 90 day labor.

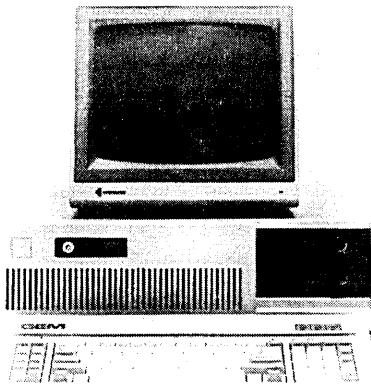
GEM-AT Specifications:

CPU - 80286
RAM - 1 Meg
Clock Rate - 8 MHZ
Disk Interface and Drives -
 (1) - 5-1/4" floppy drive 1.2 Meg
ROM BIOS - Legal BIOS
Keyboard - AT Style Keyboard
Expansion Slots - 8 IBM* compatible slots
Power Supply - 200 Watt
 Meets UL/FCC Standards
Standard Interface - (1) - Serial RS-232-Port
 (1) - Parallel Printer Port
Clock - Clock calendar battery backed up
Video - TTL Hi-Res Monographics Card
 and TTL Monitor
Operating System - MS-DOS
Warranty: 1 year all parts.
 90 day labor.

**EDUCATIONAL
DISCOUNTS**

At Micro Supply Organization we offer the lowest prices on GEM computers. 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 GEM computers. Whether you need one or a dozen computers, Micro Supply Organization is the place to get them.



GEM-AT
**Performance for
the power user.**

The GEM-AT™ runs eight times as fast as an IBM-XT™, for your multi-tasking and multi-user needs.

Including MS-DOS 3.2 with G.W. Basic. The GEM-AT™ is a complete turnkey system with everything to plug and go!

**FREE 5 module
integrated software**

\$1899

**32 meg sub system
with tape backup
\$799 w/purchase**



For our catalog with complete
details and prices, send \$2 to:

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

15% Restocking on Returned Orders



805/393-2247

By Peter McGovern

9304 SE Boise
Portland, OR 97266

Consulting With Writers

I was thinking the other day about the differences between us computer folks and writers. But then, we're all the same, really. We just write different things: they write romances and science fiction — we write technical articles, manuals, Pascal routines, schematics, and science fiction.

Writers are artists. They paint images and concepts using letters, words, and sentences much as an Italian-American chef creates a pizza using tomato sauce, cheese, and spices.

Writers want to be loved, respected, and published. They are superstitious, often certain that they write better when wearing a favorite shirt. Writers have been known for secreting manuscripts next to the frozen peas — for safekeeping. In spite of their image as eccentric and reclusive ne'er do wells, many writers are actually nice people. I should know; I am a writer.

Until recently, the typewriter has been a writer's basic tool. The bond between a mother and her newborn comes close in intensity to how writers feel about their typewriters. They know intimately the feel of its keys, the sound of its bell, and the smell of its platen.

Consulting

In addition to writing I've taken on another job: helping writers replace their typewriters. For the past three years I have consulted with, instructed, consoled, answered questions for, advised, and accepted payment from writers for this privilege.

I now spend 25% of my time consulting, the other 75% writing. I don't charge for most of my consulting time because it turns out to be a few minutes here and a few minutes there answering questions on the phone. If a call goes more than 1/2 hour I charge for it, but most calls are short. If I

have to travel, I charge a two-hour minimum (at \$25 per hour).

It's especially hard to charge writers. They don't think in terms of paying for a consultant on top of paying for a computer. Fortunately, I enjoy working with them, helping them.

Since I spend most of my time writing, an occasional phone call isn't a problem. In fact, it gives me a break. Sometimes, however, people call me 6 or 7 times a day, so if I'm concentrating on a writing project I let my recorder take the calls.

Frustrations

The biggest frustration of consulting? I guess that it's seeing people being overwhelmed by the process of acquiring and learning to use a computer. They want to make the switch painlessly, quickly, and they find that that's not the way it goes.

When I first talk with them, most writers know little more about computers than "I've heard I must have one." They both under and overestimate what a computer can do.

They are dismayed, for example, that a spelling checker won't automatically correct and exchange every misspelled word in their document. They are delighted when they discover they can recall a deleted paragraph. What they REALLY want to know, however, is "Where in the computer is my document, anyway?"

Learning

The first problem confronting writers is where to get information. Computer magazines and books aren't much help; writers want to get on with writing as simply and painlessly as possible. They don't want to be confronted with page after page of jargon or be advised to purchase an Altair or a Commodore Pet.

There are, fortunately, several books that have been written by writers who've made the change themselves. Poet and humorist Peter McWilliams

has written several books that are informative, opinionated, and entertaining. The most popular of these, *The Personal Computer Book*, was self-published by McWilliams before being purchased commercially.

Another author, William Zinsser, who wrote the style book *On Writing Well*, has also written a book on the process of moving from typewriter to word processor. *Writing With A Word Processor* is more an autobiographical description of his own adaptation than a 'how do you format a floppy disk' presentation. It has probably comforted and encouraged many more technophobic writers than any number of computer dealers saying "Don't worry; this Epson QX10 is totally user friendly."

Dealers

And speaking of dealers. Why IS it so hard to find one who is genuinely helpful? It is, in my opinion, because of the very nature of the computer industry. For a short while I worked as a salesperson at a store which advertised and sold computers for the lowest price in the New York Metropolitan area.

If you were to buy a computer there you would have gotten a great deal. You probably still would. Unless, of course, you bought a surge protector at the same time. That's because although the surge protectors cost the store only ten dollars apiece, their price was \$120!

Ideally, a salesperson would first assess your needs and budget and then recommend the appropriate system. But it doesn't always work that way. Other factors sometimes influence a salesperson's advice. When I was selling computers, the Panasonic company offered me a VCR free for every ten Senior Partners I sold. Not only that, but at one point our store got stuck with a surplus of about 25 Epson QX10s. Epson had abandoned that particular model and wouldn't

buy them back. To aid us in our sales effort our manager dropped the price by \$500 and offered us a \$300 bonus for every unit we sold.

This was great for me. But what did it mean for the customer entering our store? With so much financial incentive for us to sell particular machines it was difficult to be objective.

Support

And what kind of support can writers expect from a dealer? I've found that the first week with a computer is the most important for writers. They want to know everything from "What is a macro?" to "What is going to happen if I type 'format' and hit the return key?"

Naturally, they want to know the answers as soon as they discover the questions. But because computer salespeople usually have a sales quota to meet, they rarely have the time to provide the quality of support a writer needs. This is especially frustrating since computer stores justify their high prices by claiming to provide superior support and service!

Another obstacle writers must face when considering a transition to computers is that of deciphering technical jargon. Fortunately, most writers are proficient at learning new words. Words, after all, are what writers are all about. Unfortunately, many of the words writers know and love have totally different meanings when found in hardware and software manuals.

Sex

Most writers are women but the computing world is very technical, very male oriented. Women have a harder time learning computers partly because examples used in manuals tie in with male experiences. Also, women have more trouble with technical information because they are afraid they won't understand it.

If I present the information in terms they understand then they can handle

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 Winchester's available. No compromises.

COMPLETE SUBSYSTEMS

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

	\$465. 5mb External	\$665. 10mb External	\$790. 20mb External	\$740. 10mb Internal	\$845. 20mb Internal
Kaypro (1,2,4,10)	✓	✓	✓	✓	✓
Big Board I,II	✓	✓	✓	✓	
Xerox 820-I,II	✓	✓	✓	✓	
Micro Mint SB-180	✓	✓	✓	✓	
Gemini Zorba	✓	✓	✓	✓	
Osborne	✓	✓	✓	✓	
Sanyo MBC-1150/ 1160/1200	✓	✓	✓		
AVL Eagle (with mods)	✓	✓	✓		
Alspa II	✓	✓	✓		
Epson QX-10	✓	✓	✓		
Morrow MD-2,3	✓	✓	✓		Add \$6.00 for extra long cables.)
Z-80 S-100 Computers	✓	✓	✓		

You may buy our controller, software, cables & P/S enclosure for \$395. Add your own ST-506 family disk drive (up to 26mb) to complete your system!

Compatible with most other add-ons, Including Microsphere & Advent RAM disks, SWP CoPower/RAM disk, Pro-8 Family, Advent Turbo ROM, all speedup and double/quad density upgrades ZCPR3, Microshell, Plu* Perfect 2.2E.

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 with SOURCE files. No programming required.

We have club, user group, OEM/Dealer programs. Call for quantity schedule. If for any reason you are not satisfied, return for full refund within 30 days. All systems warranted for 90 days. Info and manual \$8.00 credited with your system purchase.

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. 17th Street
Boca Raton, Florida 33433

it. If they are familiar with home stereo systems, or cars, or cooking, or whatever, then I use these to help them understand how to use a computer.

More Than Just Word Processing

As a consultant, I can also suggest additional uses for the new tool.

Writers can use Databases to store everything from article ideas to editor and publisher mailing lists. I use, for example, a product called Squarenotes

to keep track of all my clippings and where they are filed. I can search through a list of hundreds of my clipped articles by any one of a couple of hundred keywords. I use dBASE III+ to keep track of the queries I send out. I also have databases filled with information on hundreds of potential markets.

My first use of WordPerfect's mail merge was to send off 127 letters to

(continued next page)

ON YOUR OWN

(continued from page 45)

127 magazines requesting writer's guidelines. I sent them in 127 addressed envelopes with 127 stamped and addressed return envelopes. With a typewriter it would have taken over a week. I probably wouldn't have even considered doing it. With my computer it took a day.

Next to a computer, a modem is perhaps the most useful and under-used writer's tool. In addition to allowing electronic manuscript submissions, a modem can connect a writer to the exciting world of information and people. Online services such as Compuserve and The Source provide all kinds of special interest groups and a host of other services potentially useful to authors. Databases such as Dialog or its less costly cousin, Knowledge Index, can save hours of library research — scanning millions of records in seconds and providing pages of bibliographic citations. A writer without a modem is like a sleeper without any dreams.

When I sold computers and worked as a consultant in New York I made a lot more money than I do now. But I

enjoy writing and working with writers more. I've been a voracious reader as long as I can remember. I owe a debt to the literary world. It's exciting and gratifying to be able to repay some of my loan by sharing with writers what I know about computers. And besides, I don't think it would be much fun being a typewriter consultant.

Editor's note: Peter is starting a special interest group (SIG) associated with the Willamette Writers Club. The group's focus will be on writing with a computer.

When I asked him if he really wasn't giving away his expertise he said "Sure, but I'm really a writer first. My payoff is seeing writers get into a tool that magnifies their productivity."



C UTILITIES

COMPLETE SOURCE, OF COURSE!

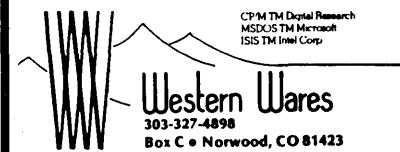
DOS-PACK: A disk full of useful MS-DOS programs, including a fancy C listing utility, disk sector editor and many others! (\$19)

TELED PLUS: Inter-system communications program, with Hayes / Zoom modem support, text capture w/ editing, MODEM protocol wildcard file transfers. Also available for MS-DOS, CP/M and ISIS-II. (\$89)

VIEW: The ultimate disk utility for CP/M systems! Recovers erased files, even if your directory is crashed. Displays or modifies every sector. (\$59)

ACCELER 8/16: Best of the CP/M emulators for MS-DOS. Enables PCs to run most CP/M programs, even Z-80 code! Also includes the Media Master disk conversion program. (no source code, V-20 chip included) (\$89)

Request a catalog of our products!



SINGLE BOARD MAIN/FRAMES FROM \$125

(Including power supply)



Write or call for complete catalog and prices.

8620 Roosevelt Ave./Visalia, CA 93291 (209) 651-1203
TELEX 5106012830 (INTEGRAND UD)

INTEGRAND
RESEARCH CORP.

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

TEXT TO SPEECH BOARD!

PC/XT COMPATIBLE. MAKE YOUR COMPUTER TALK!
A VERY POWERFUL AND AMAZING SPEECH CARD. USES THE NEW GENERAL INSTRUMENTS SPO256-AL2 SPEECH CHIP AND THE CTS256A-AL2 TEXT TO SPEECH CONVERTER.

THIS BOARD USES ONE SLOT ON THE MOTHERBOARD AND REQUIRES A COM SERIAL PORT. BOARD MAY ALSO BE USED IN A STAND ALONE ENVIRONMENT WITH ALMOST ANY COMPUTER THAT HAS A RS232 SERIAL PORT. FEATURES ON BOARD AUDIO AMP OR MAY BE USED WITH EXTERNAL AMPS.

DEMONSTRATION SOFTWARE AND A LIBRARY BUILDING PROGRAM ARE INCLUDED ON A 5 1/4 INCH PC/XT DISKETTE. FULL DOCUMENTATION AND SCHEMATICS ARE ALSO INCLUDED.

NEW!

69.95
COMPLETE KIT

89.95
ASSEMBLED & TESTED

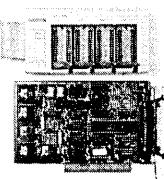
HARD DISK CONTROLLER CARD
THE FANTASTIC KONAN KDC-230. WITH MANUAL. SUPER EASY
TO CONFIGURE AND USE.
HALF SIZE PCB. \$99.00 — ADD \$8.00 FOR CABLES

**MEX-PC MODEM
SOFTWARE PACKAGE \$49.95**

A FANTASTIC COMMUNICATIONS PACKAGE WITH FEATURES TOO NUMEROUS TO LIST. SEE MAY/JUNE '86 MICRO SYSTEMS JOURNAL FOR FULL REVIEW OR CALL FOR BROCHURE. SUPPORTS COLOR, KERMIT PROTOCOL, Y MODEM BATCH, VT100 EMULATION, REMOTE OPERATION, ON LINE HELP, ETC. RUNS UNDER PC/MSDOS. 180 PAGE MANUAL & BINDER.

MEX-PACK LIST IS \$99.95 — SPECIAL **\$49.95**

NEW!
**PC/XT EPROM
PROGRAMMER**
\$199



* LATEST DESIGN * PROGRAMS UP TO 4 DEVICES AT ONE TIME * FEATURES EASY TO USE MENU DRIVEN SOFTWARE THAT RUNS UNDER PC OR MS-DOS. * USES AN INTELLIGENT PROGRAMMING ALGORITHM FOR SUPER FAST (8X) EPROM BURNING. * THIS PLUG-IN BOARD ATTACHES TO AN EXTERNAL MINI CHASSIS CONTAINING 4 TEXT TOOL Z.I.F. SOCKETS. * NO PERSONALITY MODULES REQUIRED. * AUTOMATIC VPP SELECTION: 12.5V, 21V, OR 25V. * EPROM DATA CAN ALSO BE LOADED FROM OR SAVED TO A DISKETTE. * PROGRAMMING SOFTWARE SUPPORTS: 2716, 2732, 2732A, 2764, 2764A, 27128, 27128A, 27256, 27256A, 27512, AND 27512A. * ASSEMBLED AND TESTED, BURNED IN WITH MANUAL. \$199 WITH SOFTWARE.

MICROTEK 2 M.B. EXPANDED MEMORY BOARD \$195 (O. K.)
MEETS LOTUS/INTEL/MICROSOFT SPEC. FOR EXPANDED MEMORY. FOR USE AS EXPANDED MEMORY (EMS), ELECTRONIC DISK, OR PRINT SPOOLER. WITH A SUPERB MANUAL AND INSTALLATION SOFTWARE. FOR PC/XT OR MOST COMPATIBLES. FOR 2 MEGABYTES OF 150 NS RAM (72 DEVICES) INSTALLED AND BURNED IN ADD \$225.

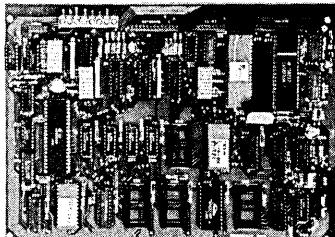
ZRT-80 CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- * Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- * RS232 at 16 BAUD Rates from 75 to 19,200.
- * 24 x 80 standard format (60 Hz).
- * Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- * Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- * Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- * 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- * Composite or Split Video.
- * Any polarity of video or sync.
- * Inverse Video Capability.
- * Small Size: 6.5 x 9 inches.
- * Upper & lower case with descenders.
- * 7 x 9 Character Matrix.
- * Requires Par. ASCII keyboard.

FOR 8 IN. SOURCE DISK
(CP/M COMPATIBLE)
ADD \$10



\$89.95
A&T
ADD
#ZRT-80
\$50

(COMPLETE KIT, 2K VIDEO RAM)

BLANK PCB WITH 2716
CHAR. ROM. 2732 MON. ROM

\$49.95

SOURCE DISKETTE - ADD \$10

SET OF 2 CRYSTALS - ADD \$7.50

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

HOT COMPUTER BOOKS!

GET THE MOST OUT OF THE INVESTMENT YOU MADE IN YOUR COMPUTER HARDWARE AND SOFTWARE WITH THE NEW WORDWARE SERIES OF COMPUTER REFERENCE AND TRAINING MANUALS. LEARN HOW TO MAKE ALL THAT EXPENSIVE SOFTWARE YOU BOUGHT REALLY COOK! BUY ANY 3: TAKE AN EXTRA 10% DISCOUNT!

ILLUSTRATED TURBO PASCAL	\$13.50
ILLUSTRATED AUTOCAD	\$18.50
ILLUSTRATED WORDSTAR	\$18.50
ILLUSTRATED dBASE II	\$13.95
ILLUSTRATED dBASE III	\$18.50
ILLUSTRATED LOTUS 1-2-3	\$15.50
ILLUSTRATED SYMPHONY	\$17.50
ILLUSTRATED C PROGRAMMING	\$17.95
ILLUSTRATED MS/PC DOS	\$12.95
ILLUSTRATED MULTIMATE	\$17.50
WORDSTAR CUSTOMIZING	\$18.50
COMMUNICATIONS HANDBOOK	\$12.50
ADVANCED TURBO PASCAL PROGRAMMING TECHNIQUES	\$19.95

NEW!

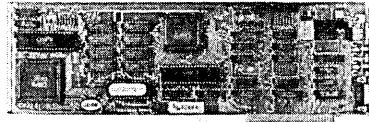
\$100 BUSS LIVES!
WE STILL SELL
\$100 PRODUCTS
AND CP/M
SINGLE BOARDS.
FREE CATALOG.

SCHEMA: Schematic Capture and Drawing Package

WORKS WITH PC'S AND COMPATIBLES. EASY TO USE WITH OFF THE SHELF PRINTERS, MOUSES, VIDEO CARDS, ETC. COMES COMPLETE WITH A HUGE COMPONENT DATABASE. SUPPORTS MONOCHROME, COLOR, OR EVEN EGA. FREE BROCHURE AND DEMO DISKETTE ON REQUEST. LIST \$500 — SPECIAL **\$396.15**

PC/XT AT

ENHANCED GRAPHICS ADAPTER



\$249

- * 3 CARDS IN ONE!
- * 100% IBM COMPATIBLE.
- * COMPATIBLE WITH IBM EGA, COLOR GRAPHICS ADAPTER, OR MONOCHROME ADAPTER.
- * 256K VIDEO RAM (4 TIMES MORE THAN IBM)
- * DUAL FREQUENCY OUTPUT FOR EITHER EGA OR STANDARD RGB COLOR MONITORS.
- * PERFECT MATE FOR NEC MULTISYNC COLOR MONITOR!
- * LIGHT PEN INPUT.
- * FULL 16 COLORS.
- * PARALLEL PRINTER PORT.
- * A SUPERIOR BOARD AT ABOUT 1/3 THE COST OF IBM EGAI
- * ASSEMBLED, TESTED, AND BURNED IN. WITH MANUAL.
- * USES CHIPS & TECH. VLSI 4 CHIP SET.
- * FIELD PROVEN BIOS.

THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

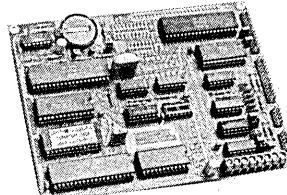
* FROM LINGER ENTERPRISES *

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. This highly versatile board can be used as a stand alone video terminal, or without a keyboard, as a video console. VT100, VT52 Compatible.

MICRO SIZE!

FEATURES:

- * Uses the new CRT9128 Video Controller driven by a 6502A CPU
- * On-Screen Non-Volatile Configuration
- * 10 Terminal Modes: ANSI, H19, ADM-5, WYSE 50, TVI-920, KT-7, HAZ-1500, ADDS 60, QUME-101, and Datapoint 8200
- * Supports IBM PC/XT, and Parallel ASCII Keyboards
- * Supports standard 15.75 kHz (Horiz.)
- * Composite or Split Video (50/60 Hz)
- * 25 X 80 Format with Non-Scrolling User Row
- * Jump or Smooth Scroll
- * RS-232 at 16 Baud Rates from 50 to 19,200
- * On Board Printer Port
- * Wide and Thin Line Graphics
- * Normal and Reverse Attributes
- * Cumulative Character Attributes: De-Inten, Reverse, Underline and Blank
- * 10 Programmable Function Keys and Answerback message
- * 5 X 8 Character Matrix or 7 X 9 for IBM Monitors
- * Mini Size: 6.5 X 5 Inches
- * Low Power: 5VDC @ .7A, ± 12VDC @ 20mA.



\$99.95

ADD \$40 FOR A&T
(Full Kit w/100 Page Manual)

SOURCE DISKETTE:
PC/XT FORMAT
5 1/4 IN. \$15

NEW OPTION

PC/XT keyboard ROM. Allows use of IBM style keyboard. \$15

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P. & H. Orders over \$50 add 85¢ for insurance. Prices subject to change without notice.

The Bulletin Board Blues

By Larry Fogg
Micro C Staff

Moving the Micro C RBBS onto a clone was a project not unlike digging a well. Every time Larry thought he had it licked, something caved in.

I caused most of the trouble. I really didn't want to lose all those users and all those messages (some really good ones).

What he won't tell you in the following article are the words he was muttering as text translation after text translation was rejected by that new RBBS with nothing more than the same, inscrutable error message. (BASIC strikes again.)

It was 1:30 Sunday morning. Dave had left an hour earlier (the slack-er) and my only companions were the softly glowing screens scattered around the World Headquarters of Micro Cornucopia. (Surely having branch offices in such exotic places as Davis and Turkey entitles Bend to be a World Headquarters.) I thought I'd just bang my head against the machine a few more times, then bicycle home.

Why was I doing this?

Mostly because the middle of the night is the only time I can concentrate. To put it mildly, Micro C is a madhouse in the daylight hours. But what can you expect from a World Headquarters? That morning I was putting the finishing touches on Micro C's new bulletin board system.

Our 5MHz Kaypro II with 3 quad density drives had performed admirably as a host for the ROS BBS, but we were feeling the crunch of limited drive capacity. Retiring the venerable Kaypro in favor of a Personal Clone with a 20 Meg hard disk gave us room to grow.

In Search Of The Perfect BBS

With the decision made to switch to an MS-DOS BBS, I began the search

for software. I especially wanted support for subdirectories and conferences. PCBoard, a public domain offering, filled the bill very nicely. It even supported Ymodem (1K block) file transfers.

Many thanks to Steve Sanders for alerting me to PCBoard's existence and supplying the code.

Translating The Files

The ROS system had collected a number of interesting messages and over 600 individual user records. We didn't want to lose these in the change over. (Actually, DAVE didn't want to lose them. I just wanted to go home to bed.)

At first glance I expected that the conversion of user and message files from ROS format to PCBoard format would be a disaster. Sort of like shooting in the dark and hoping my head wasn't in the way. I didn't have the source code for PCBoard, but luckily the documentation contained a fairly complete description of the file structures. ROS's documentation didn't cover the files at all, but I had the source code.

Strings And Things

PCBoard's files consist of a mixture of text and integers. No problem, I thought. ROS is written in Turbo Pascal, my favorite language. I'll just break out Turbo and borrow the necessary routines in ROS to read the files. Then I'll add a routine to write the information out to new files that PCBoard will understand.

It wasn't that simple in practice. PCBoard is written in BASIC and handles strings very differently from Pascal. In PCBoard's files, strings always take up as much space as the maximum allowable size of the string. This can waste a fair amount of disk space since all strings are padded to their maximum length with blanks. It also means that each string must start at exactly the right offset in the file to

be read correctly.

On the other hand, Pascal always knows how long a string is. The first byte of each string (StringName [0] in Turbo) is an integer representing the string length. When Turbo is told to read a string from a file, it looks at the string length, reads that many bytes, and is automatically in position to read the next field. Much simpler.

My solution was to write the strings out one character at a time and carefully pad each string with blanks to its proper length. I then added a CR/LF combination to the end of each line of the message file. PCBoard didn't understand the CR/LF, and printed each message as a single line. So I finally got around to what I should have done in the first place. Out came the disk editor to compare my translated file to a valid PCBoard file. How strange. They didn't look at all alike.

All lines of the valid file were terminated with E3h, which is the pi character on the PC. Since I try to know as little as possible about BASIC, I can't say if this is the standard end of line character or is peculiar to PCBoard. In any case, it was simple to convert all CR/LF combinations to pi.

Slipped Digits

Another problem was conversion of integers. I created a text file with the conversion program and therefore could not simply write an integer. Instead I wrote the two ASCII characters corresponding to the two bytes of the integer.

For example, the integer 42 is represented by hexadecimal bytes 2A 00. (Note that the least significant byte comes first.) These in turn correspond to an asterisk and a null. Turbo made things easy with the Chr() function. To convert the integer 42 I included the line, Write (FileName, Chr(42), Chr(0)).

Most of the other problems I encountered had to do with getting each field in its proper place. A disk editor

was invaluable for comparing my first attempts with files actually generated by PCBoard.

Use Of PCBoard

PCBoard wants to know all about you. As a new user you'll be asked where you're calling from, your phone number, what default file transfer protocol you'd like, and your choice of password. (Sysops browse through passwords on Friday nights for entertainment, so be inventive.)

Choose the W)rite User Info selection to update your record if you're an old user.

As of this writing there are four separate locations for messages: the main board, comments to the Sysop,

and two conferences. These conferences (Public Domain 32016 and Tech Help) are open to all users. Please leave messages in the appropriate area.

When responding to a message, use the RE option. Then we'll all be able to read related messages with the T)hread command. T)hread allows both forward and backward browsing through messages with the same subject heading.

Give Us A Call

The new board has been up for several weeks now with a minimum of grief. We do have a problem with line noise occasionally (the board signs on and then, unceremoniously, dumps

you). Calling back at a slower baud rate usually helps.

Anyone considering a similar project can find the altered ROS files (for message conversion) and CONUSR.PAS (for user file conversion) in PCBoard's ISSUE directory. Look for ROS-PCB.ARC. We are also distributing PCBoard as a two-disk public domain package (\$8.00 for the two-disk set).

We'd love to hear your reactions to Micro C's newest toy. Let us know if you have any problems or suggestions. The board is available 24 hrs/day and talks quite happily at 300, 1200, and 2400 baud. Call us at (503)382-7643.

■ ■ ■

MODEMS

INTERNAL MODEM

Features:

- IBM® Compatible
- Internal Half Card
- Bit Com™ Modem Software
- Fully Hayes® Compatible

— Four Year Warranty

1200 Baud
2400 Baud

\$145.00
\$275.00

EXTERNAL MODEM

Features:

- Requires Standard Serial Port
- RS 232C Cable
- Mite® Software
- Hayes® Compatible (no speaker but has all 8 LED's)
- Two Year Warranty

1200 Baud
2400 Baud

\$150.00
\$290.00

NEW FALL/WINTER '86 CATALOG NOW AVAILABLE
WITH CP/M & MS DOS HARDWARE AND SOFTWARE



MicroSphere, Inc.
P.O. Box 1221
Bend, Oregon 97709
503-388-1194
9-5 Pacific Time



Bulletin Board for Orders/Messages

503-388-9944

Hours: 5:30 p.m. - 8:30 a.m. Weekdays
Available 24 hrs. on Weekends

Hayes is a registered trademark of Hayes Microcomputer Products, Inc.
Mite is a registered trademark of Microf Laboratories.
Bit Com is a trademark of Bit Software, Inc.
IBM is a registered trademark of International Business Machines Corp.

CONVERTing

Pattern-Matching & List Processing For Turbo Pascal Programmers

Harold McIntosh's work (*the compilers REC and CONVERT*) is well-known to a small part of the hacker community (he received a computer hacker of the year award in 1984 for his contributions to the compiler arts), but not well-known to the general public. He sent Micro C the new MS-DOS versions a while back, with compiler source and manual-on-disk, and I've had a go at understanding this intriguing pair of compilers. What follows is an introduction to CONVERT, a pattern-matching language with a simple and elegant structure, from a Turbo Pascal perspective.

CONVERT is a pattern-matching language designed to translate literary or formal text (the source code of programming languages, for example). It's been around (on mainframes) since the mid-60s, originally as an interpreter within LISP (one of the two leading AI languages in the world).

The version available for MS-DOS and CP/M was written in REC, a language that has also been evolving from LISP, and has been used to translate Intel 8080 to 8086 assembler. In theory, CONVERT could be used for just about any kind of translation — natural language to computer language, for example.

How To CONVERT

A CONVERT program is a list of symbols, enclosed by parentheses. The CONVERT compiler looks at the list, translates it symbol by symbol, and acts on what it sees. Like Pascal, really, except — the structure's different.

For example, the emptiest program (a do nothing affair used exclusively for demonstrations) looks like this in Turbo Pascal —

```
begin  
end.
```

Not much is happening, and note that everything is optional (program heading and declarations).

The same do-nothing program in CONVERT is —

```
(())()
```

Which could also be written —

```
(  
()()()  
)
```

to illustrate its "block" structure.

The first paren signals the CONVERT compiler to begin matching symbols. The middle parens are for declarations (patterns, functions, and variables) and the main (or execution) body. The end paren says we're done.

Let's consider a slightly more complex example, a program that reads

Figure 1 - In Pascal

```
program Vowel;  
  
var  
    X : char;  
  
begin  
    repeat  
        Read(KBD,X);  
        IF Upcase(X) in ['A','E','I','O','U'] THEN  
            writeln('vowel')  
        ELSE  
            IF X in ['a','e','i','o','u'] THEN  
                write('vowel')  
            ELSE write('other')  
    until X = #27;  
    writeln('goodbye');  
end.
```

Figure 2 - In CONVERT

```
[VOWEL]  
(  
()()()  
(  
    (stop,goodbye!);  
    ((or,a,e,i,o,u),(%W,,vowel)(%R,TTY:));  
    ((or,A,E,I,O,U),(%T,VOWEL)(%R));  
    ((%W,,other)(%R));  
)  
)  
[program starts here]  
[declarations go here]  
[main body starts here]  
[main body ends here]  
[program ends here]
```

the keyboard, and decides whether a vowel has been pressed. After it decides, it writes a message — "vowel" or "other".

We'll also have it look out for an expression or special character (an escape) which will tell it to exit gracefully.

See Figures 1 and 2 for the Turbo Pascal and CONVERT versions.

The programs work identically, except that Turbo Pascal looks for an escape to tell it it's finished; CONVERT looks for "stop".

Let's compare the two line by line.

And Part By Part

Each program begins with its title. Notice that "Vowel" is enclosed by brackets in CONVERT's version. The brackets "[]" are comment designators

equivalent to Pascal's "{}". The program title is ignored by the CONVERT compiler. (The Turbo Pascal compiler also ignores the program title, but doesn't require it to be separated by brackets. It's understood. So a Turbo Pascal program will compile without the program header.)

In Turbo Pascal, we need to declare a variable to hold the character pressed at the keyboard. Our CONVERT program doesn't need a variable in this case because it doesn't save the character, it just processes it.

In general, you don't need variables in CONVERT when all the rules of a set are of the form "pattern to recognize, response." A program which makes substitutions from a table, for example, doesn't need variables. (Later, we'll look at a program that does.)

In our Turbo Pascal version, we don't need any types or constants, so we leave that part of our program blank and signal the compiler that we're ready to begin by using "begin," which seems appropriate.

In CONVERT, our declaration area (designated by the second line of parentheses) —

(())
is empty. The parenthesis on line 3 —

(

signals the beginning of the main body, or execution part of the program.

The important distinction is that CONVERT requires you to represent the empty parts of the program. Turbo Pascal doesn't. (CONVERT is really

structured!) Otherwise, they're basically very similar until they begin to execute.

Pascal has complex control structures — IF,THEN; REPEAT,UNTIL; etc., which must be specified. CONVERT has complex structures (WHILEs, etc.), but assumes an IF,THEN process order, if no execution flow is specified. The CONVERT compiler by default assumes it's to READ A PATTERN, and THEN DO SOMETHING.

So the CONVERT line —

(stop,goodbye!);

means: If "stop" at the keyboard (the default for input), write "goodbye!" to the CRT (the default) and exit. Otherwise, continue to the next parenthesis and do what it tells you.

Next, the CONVERT line —

((or,a,e,i,o,u),(%W,,vowel)(%R,TTY:));

says, "IF you see an a, e, i, o, OR u", THEN write (%W) "vowel", THEN read the keyboard. (In this case, "TTY:" represents the keyboard, but if we had left it out, as we did before, the keyboard would have been assumed).

THEN, repeat ("::"). The colon says, "let's start again at the beginning" or, "let's read another character and see what it is".

The next line handles uppercase —

((or,A,E,I,O,U),(%t,VOWEL)(%R));

says, "IF you see an uppercase A,E,I,O, OR U, writeLN (%t) "VOWEL", THEN read the keyboard (%R), and REPEAT (:) from the beginning.

The next line handles the general case (everything else) —

((%W,,other)(%R));

says, "IF you see anything else (,,

(continued next page)

Eco-C C Compiler

"This is the only package we reviewed that we would be willing to call a professional tool."

Computer Language, Feb., 1985

When the review mentioned above was written, the Eco-C C Compiler was priced at \$250.00. Now you can have the same compiler for a mere \$59.95. And that price is complete, including a library of 120 functions, all operators (except bit fields), structures, unions, long, floats, doubles, plus user's manual. We've even included a special version of the SLR Systems assembler and linker.

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

Eco-C requires 56K of free memory, 240K disk space (one or two disk drives or hard disk), Z80 CPU and CP/M 2.2 or later. We also have an MSDOS version at the same low price. Call today!

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



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



CONVERT

(continued from page 51)

write (%W) "other" and THEN read (%R) another character and REPEAT(:).

The next to closing paren —

)

says, "this is the end of the execution part" of the program.

And the closing paren —

)

says, "this is the end of the program."

Program Flow & Rules

CONVERT works by isolating parts of a text which satisfy some requirement (letters between a pair of parentheses, for example).

The isolated parts are then combined into new text, with the help of general or programmer-defined functions or subroutines.

The main body (the execution part) is a set of rules for processing the incoming patterns. In our example, our rule was — compare the character input from the keyboard, see if it matches our definition of a vowel, and report the results of the attempted match.

Its pattern-matching flow pattern makes CONVERT well-suited to prob-

lems whose solutions may conveniently be expressed in terms of transformation rules.

In order to use VOWEL, we have to compile it. Figure 3 shows the compilation process.

Reading & Writing Files

Let's consider a slightly more complex CONVERT program (see Figure 4) which uses variables and functions to copy files.

It's a bit more complicated, so we'll look closely at each line.

The line after [COPY.CNV] and before [main program] defines a function, "a", which has a variable (0), declared in the declaration part —

((0)(0)((^Z),; (<0>,(W)(R)):)) a

I've written the function on one line for convenience, but if you count the parens, you'll come up with the function's main part beginning with "((^Z)".

It says, IF the character (represented by the variable 0) read is a ^Z (or end of file), THEN end (;), ELSE write the character (0), and THEN read a character, and REPEAT (:).

The variable is declared in the variable part —

((0)(0)

The "a" at the end names the

Figure 4 - File Copy Utility

```
[COPY.CNV]
((0)(0)( (^Z),; (<0>,(W)(R)): )) a
[main program]
(
()
(
((%R,<8>:<9>.OLD)) R
((%W,<8>:<9>.NEW,<0>(^MJ))) W
(<8> 9)
((<8>:<9>(or, .,)<<
>>(%Or,<8>:<9>.OLD)<<
>>(%W,<8>:<9>.NEW)<<
>>(a,(R))<<
>>);
(<9>,<8>:<9>):
)
```

function, so it can be called from the main program. Note that this function is global (i.e. declared outside the main program, so it can be used by any function in the main body).

In the main body, two functions, R and W, are declared in the function section of the declaration part of the main program —

```
[main program]
(
()
(
((%R,<8>:<9>.OLD)) R
```

The R function (for read) —

((%R,<8>:<9>.OLD)) R

reads from a file (9.OLD) on drive 8. And W writes —

((%W,<8>:<9>.NEW,<0>(^MJ))) W

the character represented by the variable "0" and a carriage return/line feed (^MJ) to a file (9.NEW) on drive 8.

The numbers 8 and 9 are declared in the variable section and represent the drive (8) and the file name (9).

In the next line —

((<8>:<9>(OR, .,)<<

we use OR to indicate that the filename can terminate with either a space (), a period(.), or nothing(<>).

"<<" is a continuation marker, needed because CONVERT reads everything (spaces, tabs, etc.) literally. So, the pair "<<...>>" can be used to format programs into lines and columns. For example —

```
(and,<[8]>,<<  
>><0>)
```

is identical to

```
(and,<[8]>,<0>)
```

This program uses only one filename, but distinguishes the input file from the output file by adding the extensions "OLD" and "NEW" in the next two lines. The first opens a file for reading (%Or); the second opens a file for writing (%Ow) —

```
>>(%Or,<8>:<9>.OLD)<<  
>>(%Ow,<8>:<9>.NEW)<<
```

Function (or function) "a" which we declared globally gets called in line 7 to process the file —

```
>>(a,(R))<<
```

The function a says to read the old file (until we find a ^Z) and write the new file (by calling function W). Function R defines the read — read from a file.

CONVERT takes file names from the command line when we run the program. So, the variables for drive and file name will be bound at runtime.

The line —

```
(<9>,<8>:<9>):
```

sends us back to the beginning, and —

```
)  
)
```

wraps it up.

A Few Details

Variables are numbers in the range 0 to 30 (a restriction imposed by the REC implementation) separated by spaces. There are ways around this, but we won't go into them here.

The rule set (or main program) is a sequence of pairs. Each pair must be followed by a colon or semicolon.

Semicolon if it's a terminal rule (i.e. "quit"); colon if it's repetitive. The pair consists of a pattern (something to look for) and a skeleton (a function or functions).

A successful pattern match is followed by substitution in the skeleton; an unsuccessful match results in passing to the next rule. If no rules remain, the process ends.

Several functions, as we've seen, can work in sequence.

Patterns are either simple or composite. Simple patterns are variables or constants. Composite patterns are either boolean combinations of simpler patterns, pattern definitions, or recursive combinations of the two.

Skeletons are either constants, variables, or functions. Anything which isn't explicitly defined is implicitly a constant, which saves a great deal of quotation (and allows CONVERT to be more of a bare bones language).

Functions are defined either externally or internally as we've seen. Internal functions include input and

output routines, and conditional and iterative constructs (such as IFs and WHILEs).

Wrap Up

CONVERT programs have been used to manipulate algebraic expressions, translate assembler, compile and decompile FORTRAN, C, and PL/M, and parse complex natural language expressions.

If you're interested in getting to know this intriguing list processing language, it's available from SIG/M (for CP/M) and from Micro C (for MS-DOS). I'm not familiar with the CP/M version, but the MS-DOS version includes REC, CONVERT, a Manual-on-Disk, sample programs, and source for the CONVERT and REC compilers. In other words, everything you need to get CONVERTed.



**Written the
world's greatest
software? Then
tell the world!
Advertise in
Micro C.**



Whether your product is software, hardware, parts, PC compatibles, compilers, applications programs, or tools, you need to advertise it. There is no better place than Micro Cornucopia and no better price. This 1/3 page ad, run once, is only \$235.00 (multiple insertions are as low as \$156.04 per insertion). Either way, you'll be reaching over 14,000 active users for well under 2 cents per user! That's an opportunity you can't afford to pass up. Call Laura Logan or Julie Pope at (503) 382-5060, they'll give you the inside scoop on Advertising in Micro C, the world's greatest advertising bargain.

**Micro
Cornucopia** | (503) 382-5060

Path Searching
 Archiver
 8Mb Print Spooler
 Aliases
 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!*

"This system has all of its competitors beat."
Computer Language, June 1985.

"A superb product... ConIX is a bargain."
Micro/Systems Journal, May 1986.

ConIX Operating System: List \$49.95 ~~X~~ **Low As \$19.95!**

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 many 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 Pull-Down Menu: List \$~~49.95~~ ~~OFFER!~~

A user-friendly interface that is fully menu-driven, with helpful prompts and easily understood explanations of menu functions. Loads with a single keystroke at the prompt level. Source code included!

ConIX Programming System: List \$49.95 ~~X~~ **Now Only \$10.00!**

A full programming language for ConIX except SUBMIT capability. Features conditional loops, subroutines, labels, nesting, interrupt processing, error traps, and debugging facilities. Describes intricate menu systems. Includes a special "C" that provides string and numeric variables, integer math package, a relocatable assembler, generator, and much more. A must for C users and developers!

ConIX Library I - XCC Utilities: List \$49.95 ~~X~~ **Low As \$10.00!**

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.

Special Distribution Versions of the ConIX Operating System

NEW! ConIX Shareware Version:

Introducing a new Shareware version of our best seller, the ConIX Operating System. Included is our regular distribution software less the Archiver, On-Line Manual, Menu source code, and some satellite utilities.

ConIX Shareware is available through CHI for just the cost of the diskette and shipping, or on-line via many popular bulletin board systems. Users who wish to register their copy pay regular advertised prices and receive the complete software package as well as full-service customer support with upgrade notifications.

NEW! ConIX Disk Manual Version:

To reduce the cost for those who want to purchase only the ConIX Operating System, we are offering the complete software package without the printed manual. Instead, complete documentation is provided on disk with each chapter stored in individual files.

The disk manual does not include the Chapter Summary, Chapter Reference, and Index sections that come standard in our regular typeset manuals. Copies of the printed manual are available separately should you wish to obtain one at a later date.



ConIX™ Liberty Centennial FREE-Order Give-Away, New Price Reductions, Specials, and Free Offers

CHI is celebrating the 100th birthday of one of our nation's most beloved landmarks, the *Statue of Liberty*, with our *Free-Order Give-Away*. We're simply saying thank you for being our customer by giving you a chance to receive your order absolutely *FREE!*

Every 100th order processed by our computer will be shipped with a Credit Certificate for the total purchase price or \$100.00, which ever is lower. This credit may be used towards payment of a future order from CHI, or may be redeemed within ninety (90) days of receipt for a check for the full credit amount. Your odds of receiving a free order are an incredible *1 in 100!* To apply, certify eligibility by enclosing the completed and signed form below with your order. (Photocopies accepted.)

Offer applies only to private individuals and non-profit institutions ordering directly from CHI. Orders placed by PO or purchased for commercial use are not eligible. This offer fully replaces Sweepstakes previously advertised.

FREE! Pull-Down Menu System:

As a special offer, the ConIX Pull-Down Menu System is included *FREE* with all Operating System purchases. That's a savings of \$24.95 off the regular list price!

SPECIAL! Buy 'Em All Discount Offer:

As a special incentive, when you buy all three ConIX packages, you get the Library for only \$10. That's right, for \$69.95 you'll get the ConIX Operating System, Programming System, Library I, and three instructional manuals. In addition, we'll throw in the Pull-Down Menu at no extra cost. Act now and secure your copy of the complete line of ConIX products at our lowest price, ever!

FREE-Order Give-Away

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Dear Computer Helper Industries:

Please enter my name in your Give-Away offer. I understand that every 100th qualified order will be shipped with a Credit Certificate for the full price of the order or \$100.00, which ever is lower, and I will have up to ninety (90) days to exchange the credit for a check in full. I also understand that I am only eligible to enter if the order is for private, non-commercial, or non-profit institution use.

Signature: _____

Void where prohibited by law. Offer may be cancelled without prior notice.

DT: _____ O #: _____ C #: _____ PP \$: _____

Name: _____

Company (if any): _____

Address: _____

City, State, Zip: _____

Tel. #, Hours: _____

— ConIX Operating System:	\$ 29.95	\$ _____
— Disk Manual Version	\$ 19.95	\$ _____
— Limited Shareware Version	\$ 0.00	\$ _____
— Printed Manual Only	\$ 9.95	\$ _____
— ConIX Pull-Down Menu System	\$ 0.00	\$ _____
— ConIX Programming System:	\$ 29.95	\$ _____
— Printed Manual Only	\$ 9.95	\$ _____
— ConIX Library I XCC Utilities:	\$ 24.95	\$ _____
— Printed Manual Only	\$ 9.95	\$ _____
— All ConIX Packages Above	\$ 69.95	\$ _____

Computer Brand: _____

Disk Format Information: *

— 8" SSSD Standard	\$ 5.00	\$ _____
— 5 1/4" DSDD 48 TPI Soft Sector	\$ 5.00	\$ _____
— 5 1/4" SSDD 48 TPI Soft Sector	\$ 6.00	\$ _____
— 5 1/4" <u>S</u> <u>D</u> <u>TPI</u> <u>Sector</u>	\$ 10.00	\$ _____

* Add one format charge for each package ordered (complete system counts as one).

Format Brand: _____

Shipping Information:

<input type="checkbox"/> UPS Ground USA	\$ 4.50	\$ _____
<input type="checkbox"/> Air Mail Canada	\$ 9.50	\$ _____
<input type="checkbox"/> Air Mail Foreign	\$ 12.50	\$ _____

Subtotal: _____ \$ _____

Tax (N.Y. Residents Only): _____ \$ _____

Total (Thank You!): _____ \$ _____

POs and UPS CODs accepted by phone only. Sorry, credit card payment unavailable. Personal checks require 10 days to clear. Non-USA orders must be prepaid by bank draft in US dollars.

Computer Helper Industries Inc.

Post Office Box 680
Parkchester Station, N.Y. 10462
(212) 652-1786 9AM-5PM M-F

By Howard Saltzman
38 Main St Ct
Randallstown, MD 21133

7MHz On An 83 Kaypro

Every time we do another speedup, memory expansion, or video mod to a system, there's a flurry of hardware activity and a flurry of questions left on the Micro C RBBS. (Now there's a Tech Help conference just for such occasions.)

So, this 7MHz Upgrade for 83s ought to fill up the board (503-3827, 7643, 24 hrs, 300-1200-2400). I figure it will be at least as popular as 5MHz — you should see the data flash up on the screen. (On second thought, maybe you shouldn't.)

If you have an '83 series Kaypro II or 4 and are still running at 2.5MHz, then you're long overdue for a speed-up. If you've already done the 5MHz then you already have an idea what 7MHz will do.

The modification is really simple. A Z80H, rated at 8MHz, "B" parts for the PIO's and SIO, a 74LS04, and two 74F157s are all you'll need. If you can't get a Z80H, a Z80B will probably also run at 7MHz. The RAM and ROM need only be 200ns parts. (But faster is, of course, fine.) Many Kaypro's have 200ns RAM (a -2 or -20 following the device number). Check yours to see if it needs to be replaced.

Addressing The RAM

The RAM chips receive the 16-bit address put out by the Z80 in two eight-bit chunks. U33 and U34 decide which 8 bits are sent to the RAM depending on whether the MUX signal is high or low.

The first 8 bits are read by the RAM when the RAS (Row Address Strobe) goes low. Subsequently, the MUX signal goes high causing U33 and U34 to switch to the second 8 bits of the address. The CAS (Column Address Strobe) goes low next and the RAM reads in the second 8 bits.

On the 83 Kaypro, U66 generates

CAS and MUX signals, and it works fine at 5MHz. However, at 7MHz, the CAS signal is too far behind the RAS. So the RAM doesn't have time to put data on the bus before the Z80 reads it.

For 150ns RAMs the row

signal is received. Also, the CAS signal should not be generated until at least 25ns after the RAS signal. 200ns RAMs need to have 5ns added to each of these times.

Generating CAS & MUX

My solution is to remove U66 and replace it with a hex inverter (74LS04).

U39 pin 11 supplies RAS to the new 74LS04. The new chip inverts RAS three times and then sends it to U33 and U34 as a MUX signal.

The signal is then passed through two more inverters and becomes the CAS signal. The propagation delay through a 74LS04 inverter averages 8ns (that is, it takes about 8ns for a signal to get through each gate).

So RAS goes valid and 24ns (3 gates) later, the MUX line changes. Then 24ns later (2 gates in the new U66 and 1 gate in U48 which is part of the original circuit) CAS goes valid.

Wiring The New U66

The 74LS04 replacement for U66 should have pin 2 connected to pin 13 and pin 12 connected to pin 3. The output of this inverter is pin 4 from which the MUX signal is sent to U33 and U34.

Pin 4 is then connected to pin 11 and pin 10 should be connected to pin 9. The output of this inverter is pin 8 which should be connected to pin 5 which is connected on the board to U48 where it is inverted one last time to become the CAS signal. (i.e. The cheekbone's connected to the jawbone, the jawbone's connected to the funnybone, and the ...)

Pins 1 and 2 on U66 should be bent

up since these pins were originally tied high for U66. Similarly, pins 8 and 9, the clock and CLR inputs, should be bent up. The other pins should be bent into the socket but not connected to any

Another consideration is that the address multiplexers U33 and U34 must be able to change output in less than the 24ns between the MUX and the CAS signals. 74LS157 parts are not fast enough, so faster parts will be needed here. 74F157's will do the job nicely.

Editor's note: See Figures 1 and 2 for a wiring diagram of the entire 7MHz mod. Also, 74S157s worked fine in our 83 Kaypro, and they were a lot easier to get than 74F157s. The 250ns ROMs we tried also worked fine at 7MHz.

Clock

Now all we need is a 7MHz signal and we'll be off and flying. U86 receives the 20MHz master clock signal and divides it by two to produce a 10MHz clock. This 10MHz clock is fed back into the chip at pin 11 and divided by 2 and 4 to produce 5 and 2.5MHz signals respectively.

A 13.9776MHz clock is available from the video section of the board on pin 8 of U2. Connecting this signal to the bent out pin 11 on U86 gives a 7 and 3.5MHz output where the 5 and 2.5MHz signals were before.

This completes the 7MHz mod.

Editor's note: I've added the following to this article because I figured you'd have some questions. I hope this helps.

For Original IIs

Those of you who have an original Kaypro 2 with "81-149" marked on the paper stuck to the top of your monitor ROM are going to have to replace that ROM.

ROMs marked that way are 2716s

(continued on page 58)

Figure 1 - Using 74LS04 To Delay The MUX And CAS Signals

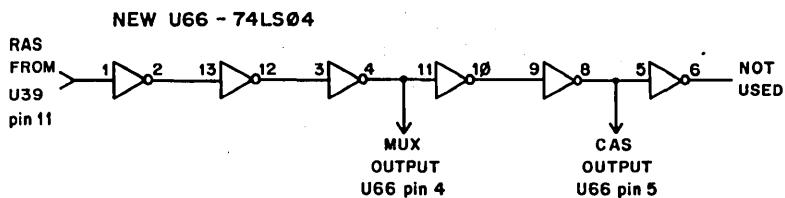
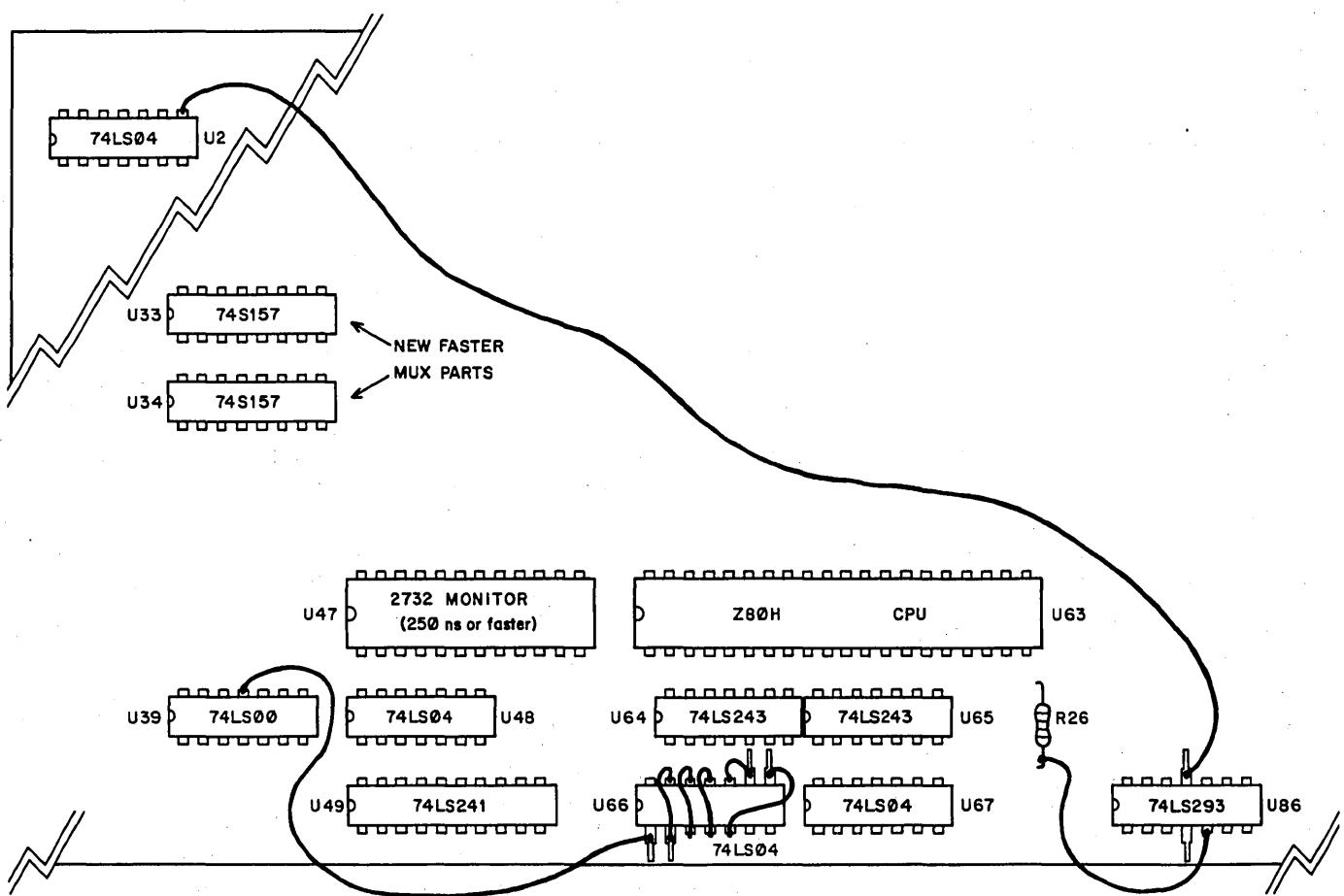


Figure 2 - 7MHz Speedup Wiring Diagram. (Pin 1 is immediately counterclockwise from the notch.)



(continued from page 56)

(2K bytes) and they are very slow. The fast 2716s are marked 2716-1 but they are 350ns parts (barely fast enough for 4MHz). The ones marked just 2716 were 450ns parts (barely fast enough for 2.5MHz).

If you want a cheap way to speed up your system, you can do one of two things. You can copy the data from your 2716 into the upper half of a fast 2732. Or, you can do the II to 4 upgrade (see Micro C issue #15 or #21 or #24) and then plug in a Kaypro 4 monitor ROM. (You can use your original single-sided drives with the 4 monitor if you wish.)

Our monitor ROMs are 250ns parts and the ones we've tested have worked at 7MHz. We've had trouble finding faster 2732s, and so far we haven't needed them. Anyway, if you have a Micro C Pro-Monitor ROM in an '83 machine, then you should be able to run the upgrade without replacing the ROM.

5MHz Speedsters

Those of you who did the U66 mod when you speeded up your system to 5MHz may have to undo that mod when you replace the 74164 (current U66) with the 74LS04.

If you did the modification by pulling out pins on the 74164 (MUX output on pin 3, CAS output on pin 4) and then jumpered from the chip to pads underneath the board, then all you have to do is remove the jumpers when you remove the 74164.

If you cut traces and added jumpers under the board, then you need to repair the traces and remove the jumpers (just the ones at U66) before you plug in the modified 74LS04.

84 Kaypros

This 7MHz mod is for 83 Kaypros. There is a reason. First, the clock circuit for the latest 84s is pretty much hidden inside a large custom LSI chip. Second, 84s have trouble enough working at 4MHz much less 7MHz.

Advent and a few other manufactur-

ers came out with 8MHz speedups just after the 84s were introduced. In fact, they came out with those products right after Kaypro called me, wondering why a lot of their machines were very sensitive to ROM chips and auto-dialers and 6545 video chips. The reason was that the poor Z80 was trying to drive too many things on the data and address busses.

Needless to say, the 8MHz speedups didn't last very long.

So, if you have an 84 and it's working, then don't fuss with the clock speed. After all, there's no reason to slow it down while it's still working.



MORE... VALUE and PERFORMANCE with Mitek's Relocatable Z80 Macro Assembler and Z80 Symbolic Debugger

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.

- 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 Users:

Assembler will generate Turbo Pascal in-line machine code include files.

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.

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: (808) 623-6361

Specify desired 5½" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.

P. O. Box 2151
Honolulu, HI 96805

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.

NOW AT THE SBC MART COMPUTING SALE-A-THON

1.2MB Floppy on your PC or XT

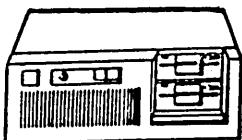
★ Now you can have a 1.2MB-and-360K AT-compatible floppy disk drive on your IBM (or compatible) PC or XT. Our 1.2MB controller replaces your floppy disk controller. The controller is \$149, a TEAC 1.2MB/360K disk drive is \$135. No software driver required. This is a super reliable subsystem. You can boot from the 1.2MB drive.

SPECIAL BONUS Norton Utilities Ver 3.0 Reg \$99 only \$60 w/purchase of any disk drive. IBM PC-DOS 3.2 (the real thing) Reg \$95 only \$70, w/purchase of motherboard.

ADD-ON CARDS

- multifunction card 384K, clk/cal w/battery, serial, parallel, and game ports, with OK save \$50 Model MFC-4 \$89
- monochrome graphics card runs 1-2-3 graphics, w/printer port, 720x348, MGC-1 now \$99 (Hercules equiv.)
- color adapter w/light pen port, RGB and composite outputs Reg \$140 Model CC-1 now just \$79
- ★ better than the Super 7: floppy disk controller, clk/calendar, serial, parallel & game ports, spooler & RAMDISK s/w Sale \$99 ideal mate for 640K motherboards
- clock/calendar card Reg \$58, CL-1 \$49
- ★ Turbo RAM AT 2MB RAM card. Its size and speed enable it to be used in all 6 to 10MHz 80286 computers. EMS, works with 1-2-3, etc. Reg. \$249 Model CSS-EMS now \$199 w/OK
- ★ Turbo RAM XT same as above but for PC/XT systems Model PCC-EMS \$179 w/OK

AT COMPATIBLES



IBM AT compatible computers. Made in the USA, complete systems, FCC approved. Components:

AT motherboard w/512K \$625
200 Watt power supply \$125
AT hard & floppy disk controller w/cables \$279
AT case \$99
AT keyboard \$99

Shipping and handling: drives/motherboards/cases/power supplies \$4.50 ea., cards \$2 ea., software \$2.50 ea., keyboards/modems \$4 ea., speedup kits \$2 ea., memory \$1/set, COMPUTERFACTS \$3 first set then \$1/set. CA residents add 6% sales tax.

Visa/MC/AmEx ORDERS:(619) 375-5744

The SBC MART, P.O. BOX 1296, Ridgecrest, CA 93555



HARD DISKS



- complete 30 MB kit
- special only \$499

- complete internal ½-high 20MB kit, incl. controller and cables, special \$439
- complete 30 MB kit with ½-high drive, uses new Adaptec 2070-A controller, \$499

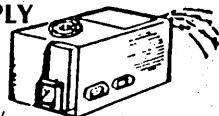
★ Generalized hard-disk controller works with almost any drive. Just tell the controller your drive specs. Fast OMTI 5510-7 hard disk controller with cables and OMTIDISK utilities HDC-4 only \$159

PRINTER BUFFER

Very easy to install 8K buffer for Citizen MSP 10 & 15 printers. List \$45, reg \$34, on sale \$21

- "Assembling an IBM PC/XT Compatible Computer System" A novice can put an IBM PC compatible together with this new detailed manual. \$19
- Microsoft's new QuickBASIC full MS BASIC compiler, Reg \$99, only \$89

ALMOST SILENT POWER SUPPLY



- 180 Watts
- side switch
- top quality
- one year warranty
- standard cables for 4 disk drives
- very low noise
- Model PS-135 power supply Reg \$129, now only \$75

TURBO PROGRAM

★ This program will set virtually all turbo motherboards to the turbo, or high-speed, mode. Can be called from the keyboard or from your AUTOTEXE file only. \$19

BEST DOS MANUAL

- ★ Microsoft Press' "Running MS DOS". For the less experienced and the most knowledgeable MS/PC DOS user. 423 pages. We regularly get calls from our customers telling us how great this book is.
- \$21.95

ABOUT OUR MOTHERBOARDS

- the most IBM PC compatible motherboards available
- each runs all commercially available software
- each works with all commercially available add-on cards

These FOUNDATION motherboards are the most compatible you can buy. All work with V20 chips, have parity checked memory, an 8087 socket, 8 adapter slots, four empty ROM sockets. Super manual includes complete data on how to put together a complete PC system. It even includes schematics.

Logitech Modula VS Turbo Pascal

By Laine Stump

Development Foundation of Turkey
Tunali Hilmi Cad. 22
Ankara Turkey

There comes a time in every adventure when things get a little (or a lot) crazy. I don't know if it's culture shock or computer shock that Laine's suffering, maybe it's a bit of both. Anyone who's been in the Peace Corps or Vista, no doubt, understands all too well.

Slouched on a chair in the living room listening to Jimmy Cliff on the blaster ('Stop That Train, I Wanna Get Off'). I arrived in 'Stamboul (Istanbul, to those of you who have never heard a local pronounce it) yesterday afternoon after a 6 hour drive through the mountains, forests, and villages of Western Anatolia.

The little orange Opel performed flawlessly, transporting myself and two computers swiftly and smoothly over the windblown passes, past the fume-belching lorries, and through the deathtrap maze of traffic jams that start in Izmit at the east end of the Sea of Marmara and continue for 130 kilometers all the way to the Bosporus Bridge. The jams don't ease up even when the freeway grows from four lanes to six.

Now I am sitting in Russ and Gloria's apartment, back bent into an unnatural and spine damaging position, using the dull light coming through the balcony door from the wispy, "morning after the rain" sunshine to illuminate the LCD screen on my little Toshiba.

I briefly considered moving outside to the balcony to let the view of the European side of the Straits of the Bosporus inspire my writing.

It is a typical 'Stamboul morning in the city, Topkapi Palace, the Blue Mosque, and Dolmabahce Palace faintly glimmering through the mist, minarets poking through the haze every hundred meters or so, a constant stream of ferry boats struggling from

Europe to Asia against the relentless current into the Sea of Marmara. There's an occasional tanker or cargo ship chugging sluggishly north towards the Black Sea, reminding me of Jason and his Argonauts, and the similar journey that they made through the Bosporus to the Black Sea. They returned with tales of giant sea monsters, beckoning sirens, and treacherous currents that would toss any but the most gallant and determined man to his death in foaming whirlpools that descended into the depths of Hell itself.

It is late October — a bit too cold to sit out on the balcony, especially the morning after a rain storm. Besides, I'm supposed to be writing about computers and listings and test runs and utilities and bugs, not heroic adventures and adventuresome heroes of centuries long past.

Project? What Project?

Usually I attempt to have a small programming project in every column. Some little utility routine that can be useful in many programs, or maybe act as a catalyst to help other people discover new ideas and new uses for old ideas. This month I am noticeably lacking the normal 'Assembly listing' and 'Pascal listing' that everyone has become accustomed to.

There is a reason for this. Usually I spend at least one fourth of my work time solving small problems and writing utilities and interfaces to make life easier. I haven't been doing much of this lately. I haven't had time.

Oh, sure, I've done some interesting things recently, but how many of you are really interested in changing the character set on your Hercules card to add the seven Turkish characters missing from the IBM set? How many people really want a program that gets Turkish characters from a keyboard? How many of you really give a damn about fixing non-working Turbo Clone boards by hacking up the BIOS ROM?

By the time you get them, either they work or you send them back to the dealer for a replacement.

(Actually, a few people might be interested in some of this. Maybe I should put together a disk of all the little utilities I've written for editing character sets, translating input keys, and everything else. Someone may get some use out of it someday...)

Besides, I've been so busy trying to make our worthless Clone boards run that I haven't had time to do anything really FUN in the last 3 months. My time has been spent trying tricks with resistors, modified ROMs, swapping boards to try and consolidate pieces from non-working systems into as many working units as possible.

Clone Casualties

More and more little problems keep showing up on the clone boards, and I realize more and more that they are not what they were represented to be. Most noticeable is the fact that almost none of them will run unmodified at 8MHz, as we were told they would.

This problem stems from two causes — 1) some floppy disk controller cards will not function at 8MHz reliably, and 2) if the system clock is changed back to 4.77MHz for every floppy access, the motherboard will eventually hiccup on a clock glitch and crash, sometimes rebooting, sometimes just dying in its tracks.

This is really a Catch 22 problem. If I leave in the ROM that slows down the clock for every disk access then I am 98% sure that the system will crash within the first 5 minutes of floppy activity; if I remove the piece of code in the ROM that switches speeds then the floppy controller will immediately begin getting 'read fault' errors. What to do, what to do...

I have found almost NO boards that can handle the continual switching of the system clock speed, but about 2/3 of the floppy controllers can handle 8MHz continuously (for everything

EXCEPT formatting). From this data, I hit on the compromise solution of using modified (non-switching) ROMs and running full out all the time. When I need to format a disk, I use a short program to switch to slow mode, format the disk, then switch back to fast mode.

That still means that about 1/3 of the machines will not work at 8MHz. Ah well, still better than a Commodore 64, I guess.

Other than that, I found two or three other motherboards that couldn't run at 8MHz (even when I didn't switch speeds all the time), four speakers that were connected improperly (they forgot to take the insulation off the wires before they crimped on the connectors), several more screws with stripped threads, and, of course, the ten or so floppy controllers that wouldn't work at 8MHz.

I am somehow reminded of a caption I saw on a cartoon in Fortune Magazine: "Gee, now I see it! If you design the quality into the products and manufacturing process to begin with, you don't have to worry so much about quality control and high failure rates!" Something like that anyway.

It's Only Logical

When I was in the States this summer, I got something I've been wanting for a long time - a compiler for Modula II. The compiler I got was the one from Logitech; when I called The Programmer's Shop they said that it had been the most popular Modula among their customers, so I thought I'd go along with the crowd for once.

It's already been three months since I left the U.S. and I still haven't had the time to try out anything major with it to see just how wonderful a module is. Not that I need to write something in Modula to discover that - I've been writing my Pascal programs that way for years.

The nice (??) thing about Modula is

that it FORCES you to follow the rules of modularization and data hiding. You can do the same things in Pascal if you like, but when it comes down to the deadline and things aren't finished, you can be lured into breaking a few rules to "save time." If you are working in a team with other programmers and you are using Pascal, you can't TRULY hide any data types or static data from them. If you are using Modula, you don't have to trust your teammates, you only have to trust Modula.

Logitech Modula

Logitech's Modula is a full implementation of the Modula described in the 2nd edition of Wirth's book on Modula. It also contains most (but not all) of the extensions and modifications given in the 3rd edition of the same book. It is a 4-pass compiler that receives source code and generates a .LNK and a .SYM file to later be linked with Logitech's own object linker program. The linker produces a .LOD file which can then be combined with the runtime support package with the LOD2EXE utility to create an .EXE file.

A text editor (called MOD) is included with the package, and it is "kind of" integrated with the compiler, but not to the extent of Turbo's editor. I think that I actually like the Logitech approach to integration better.

The editor calls the compiler and the compiler parses through the entire program, finding ALL syntax errors and placing them in a file. After compilation, the editor works through the error list, showing the error message on the screen along with the section of the program that generated the error.

This can be very nice if, for example, you have 4 errors at the end of a program that is several thousand lines long. If you were using Turbo, it would find the first error, abort the compilation, let you fix it, compile

everything again, find another error and abort, let you fix it, compile everything again, etc. With Logitech Modula, you only compile twice - once to find the errors and once to generate the object code.

Besides, I think the interactive feature of Turbo is being stressed much too heavily. Good programmers have their entire program designed, written, and even partially debugged before they send it to the compiler. The only thing an integrated editor should be used for, other than learning the language, is to correct stray syntax errors that creep in.

The majority of time spent on a project should not be spent recompiling and tinkering to try and make a program work. Time should be spent in the initial design, making it complete, clean, and compact (the three C's ??) so that you won't spend so much time guessing.

More Comparisons

I really hate to stoop to the level of comparing Logitech Modula with Turbo Pascal, because they aren't even in the same league; they both have their place on my system disk and I plan to continue using both of them. But I just got done reading a review of another Modula compiler and the entire review was geared toward showing how superior Turbo was to Modula. I feel the need to disagree and officially post a rebuttal.

Program Size

One of the first complaints that I heard about Logitech Modula (and many other Modula compilers for MSDOS) was that it didn't generate tight code. While it may be true that the .EXE files generated by Modula are anywhere from 1.5 to 2 times the size of a similar Turbo program, the problem is not that the code generation is sloppy, it is that when one procedure

(continued next page)

(continued from page 61)

of a utility module is used, the code for the entire module must be included in the .EXE file. There is also the overhead for the part of the runtime package that handles processes and concurrency. Besides, allowing a full 640K of code means that all procedure calls must be 'far' calls (5 bytes) instead of 'near' calls (only 3 bytes).

Of course, since Logitech does not limit programs to only 64K of code, and since you can rewrite your own I/O modules to contain only the procedures you need, this is not really a problem. If you are that concerned about code size, you should be programming in assembly language or FORTH.

If you write new I/O libraries for Turbo, you must include them, in source form, in every program you write. And even then, you are still stuck with the original 10K I/O and string library. If you want new libraries for Modula, you can completely throw away the originals if you like. Logitech even includes the assembly language source code to their runtime support package (process handling, program initialization) which you can change if you like.

Modula In ROM

They also have a package of libraries that allows you to compile ROMable code. While I usually prefer assembly for anything that will go into ROM, this might be an interesting idea if you were designing a complicated process control or data acquisition system and didn't want to have the extra expense (initial and maintenance) of a disk drive.

Converting From Pascal

In order to aid new converts to Modula, Logitech has developed a program (PAS2MOD) that will automatically translate a Turbo Pascal source file into a Modula source file. There are some instructions that can't be done by machine, but most of the drudge work of conversion is removed. Detailed instructions and examples are given in the PAS2MOD manual describing how to take care of the few "non-solvable" problems.

The translation program itself can be

quite an education for people trying to shift their thinking from Pascal syntax into Modula syntax. As the program is running, it can optionally display the original Pascal source code on the left side of the screen and the Modula code on the right.

A trap that I fell into when I first started reading the PAS2MOD manual was to judge the relative strengths of Pascal and Modula based on the conversion process.

"Look at how awkward this is!" I silently screamed. "Why do I have to make a procedure call just to empty a set???"

This is not a fair way to judge Modula, though. The problem is that the PAS2MOD manual explains all the areas where Modula is (apparently) inferior to Pascal. Unfortunately, it doesn't point out Modula's relative strengths.

Modula Can't

An example would help here, I think. One limitation of the Logitech implementation of Modula is that a function cannot return a complex type. For instance, you cannot have the following function:

```
PROCEDURE
  UpCaseStr(source:ARRAY OF CHAR)
            :ARRAY OF CHAR;
```

as you can in Pascal. (Notice that Modula removes the reserved word FUNCTION and replaces it with PROCEDURE.) Instead, you must turn the function into a procedure:

```
PROCEDURE
  UpCaseStr (source : ARRAY OF CHAR;
             dest : ARRAY OF CHAR);
```

At first this may not seem like any hindrance at all. Instead of calling it with:

str2 := UpCaseStr(str1)

you would use:

UpCaseStr(str1, str2);

It does get to be bothersome, though, when you realize that you can't do:

str2 := Trim(UpCaseStr(str1));

or

```
IF (UpCaseStr(str1) = 'ENGLISH')
THEN
```

Instead, you must call the procedure, placing the result into a temporary variable, then call the second procedure (or test the condition). I agree, this is a bit of a bother. It takes an extra local variable and a few extra key taps to get things done.

Turbo Pascal Can't

On the other hand, let's consider a feature of Modula that isn't in Turbo Pascal. Modula allows procedures to be passed as parameters to procedures. This makes Figure 1 possible.

Now, let's see you do THAT in an understandable manner with Turbo.

So, you get my point. By its very nature, the PAS2MOD manual points out weaknesses of Modula while not saying much about its strengths. Keep that in mind as you are reading it.

Overall Impression

To sum up my feelings and thoughts about Logitech Modula II — I must say I'm looking forward to translating all my Turbo I/O utilities into Modula. Chances are, I'll be using Modula on my next large project.

(Many of the extra utilities I wrote for Turbo — like the Exec procedure a few issues ago — are already included in one of the Logitech library modules.)

Logitech has put together a very complete program development system here. It includes a compiler, linker, and editor, as well as a post-mortem debugger and a run-time debugger (how many times have you wished you had THAT for Turbo?). You also get an extensive set of library modules including such necessities as file system management and mouse control (rat poison?).

Reservations

I do have a few small complaints. (Never satisfied, am I??) First, a few of the features of Modula as reported in the 3rd edition of Wirth's report are not yet implemented. Also, since sets larger than 16 elements are implemented with a special library instead of being built into the language, it's quite

inconvenient to use one of my favorites - "SET OF CHAR."

Another weakness is that there is no straightforward way to link in a module written in assembly language. Standard procedures for modifying register contents and an inline type of statement help alleviate the need for this facility, but it would be much nicer to have a utility program that would convert an .OBJ file from MASM into a .LNK and a .SYM file for Modula.

Final Score - 9.5 (make it fully optimizing, linking to assembly, and clairvoyant and maybe I'll raise it to 9.75).

Now Some Hardware

I hate disk drives that sound like chainsaws. My Bigboard was so frightful that I finally resorted to handling the stepping in software with timing loops to speed it up (the 1771 would only go up to 6ms and I needed about 1.6ms). I still cringe every time I hear the old familiar grrraaakk grrraakkk, just imagining aluminum shavings fall-

ing into the diskette and collecting right here on the very tracks where my Micro C columns reside.

Well, when I got my Toshiba it had mildly noisy drives. I lived with it for awhile, knowing there must be SOME way to change the step rate, but unfortunately I didn't have any documents on the 765 disk controller. Imagine my relief when I got a care package from Cecil containing a new mechanical pencil and *Inside the IBM PC* by that guy Norton. Finally I figured out what the 'disk parameter table' pointed to by the address contained in the INT 1Eh location. The high nibble of the first byte keeps the step rate. Just change its value from D (6 msec) to F (2 msec) and things are magically quiet. F will work great for most 80 track drives; you'll probably use E (4 msec) for 40 track drives.

On the Toshiba, I only changed the value in the disk parameter table in the boot sector; on other systems you may have to change it somewhere else (in the BIOS maybe). There is a program on Micro C MSDOS disk 22 that

Figure 1 - Passing A Procedure To A Procedure In Modula-2

```
TYPE FunProc = PROCEDURE (REAL) : REAL;
PROCEDURE GraphFun (lo, hi, step : REAL; Fx : FunProc);
  VAR x : REAL;
  BEGIN
    x := lo;
    WHILE (x <= hi) DO
      GraphPoint(x,Fx(val));
      x := x + step;
    END; { while x <= hi }
  END; { GraphFun }

{ Call it with: }

GraphFun (1,10,1,Sqr);
GraphFun (2,30,-3,Sin);
GraphFun (1,1.3,.002,MyFun);
```

Figure 2 - A Short Program To Change The Step Rate To 2 msec (or 4 msec)

A short program to change the step rate to 2 msec (or 4 msec).

```
MOV AX,0
MOV DS,AX
LDS BX,[1Eh*4]
MOV [BX],0FFh ; 0EPh for 4msec
MOV AX,0 ;reset disks from ROM BIOS to
INT 13h ;force reread of disk parameters
MOV AX,4C00h
INT 21h
```

does the fix by patching in the value after the system has booted. This program works fine (for 4 msec) even though the documentation is incorrect. The documentation notes that changing the value from D to E "lengthens the head settle from 26 to 28ms." This isn't correct. Oh well, it works anyway. (See Figure 2.)

Ok, back to work now. I didn't come to 'Stamboul just to ride back and forth across the Bosporus on the ferryboats and stare at mosques. I've got things to do!

Next time: "How to learn that you can't know that it is impossible to learn that you are insane if you think that you know it is impossible to learn that you think that you can't know you are insane."



16 BIT SINGLE BOARD COMPUTER

The Micro-86 SBC runs PC DOS versions 2.1 or 3.1, and hosts these features:

- 8 MHz 8086 16 bit CPU, can be replaced with V30 processor.
- 128k/512k on board RAM
- 2 asynchronous serial I/O ports, 1 for system console, 1 general purpose.
- Centronics printer port.
- Floppy disk controller will handle up to 4 drives, 5 or 8 inch, SD or DD.
- Standard EPROM contains BIOS, disk formatting program and setup program for disk drive and serial terminal set up.
- Small size, 5.75 x 8.5 in. Mounts on a 5 1/4 inch drive.
**\$349⁰⁰ + 3⁰⁰ shipping
(128k)**
- California residents please add sales tax.

Micro Technical Designs
7584 W. Azusa Rd.
Dos Palos, CA 93620
(209) 387-4235

PC Pursuit

By David Thompson

Cheap Communications

I first heard about PC Pursuit at a meeting of a computer club in Eugene. It was during descention (the time when everyone descends on the speaker) that Richard Bear mentioned unlimited long distance RBBS access for \$25 per month. After that, no one else had a chance. I asked Richard to get me the info. He did. And the following is the result.

Think for a moment: Why aren't you spending 12 hours a day exploring all those free boards across the nation?

Because you wouldn't get any sleep? Not a chance, you can get plenty of sleep at work.

Because you have a family?

Not good enough. You carry pictures don't you? Besides, the little tykes'll be a lot more interesting after they've graduated from college.

Because you can't afford \$4,500 a month in long distance charges?

Right!

But hold on. If you're one of the multitudes who's near a node (any of 17,000 Telenet nodes) there's something new that will let you feed your information habit for nearly nothing. (Better take the spare cot down to work right away.)

For \$25 per month (that's all, \$25 per month) PC Pursuit (a part of GTE Telenet) will give you unlimited calling into 14 cities (and any other burgs that are within free calling distance of these cities).

The cities and their area codes are:

- (404) Atlanta
- (213) Dallas
- (313) Detroit
- (214) Los Angeles
- (212) New York
- (415) San Francisco
- (202) Washington DC
- (617) Boston
- (303) Denver

- (713) Houston
- (210) Newark
- (215) Philadelphia
- (206) Seattle
- (312) Chicago

(There'll be six more cities by the first of the year.)

Unlimited But Limited

Your unlimited access doesn't include every RBBS within the specified area code, just the ones within the free calling distance of the specified city.

Your unlimited access is limited to the hours between 6 p.m. and 7 a.m. on weekdays. During business hours you pay an additional \$10.50 to \$14.00 per hour. But, that's significantly less than the \$60.00 per hour for standard long distance during prime time.

Your unlimited access is limited to devices which answer the phone with carriers. Telenet is an X.25 (packet switching) network, so you aren't talking directly to the board. You're talking to a node — which is talking to another node — which is talking to the board.

You'll be talking to your node at 300 or 1200 baud, but your data is put into packets (up to 128 bytes each) and passed along to the destination node at 56K baud. There, it's unpacketed and sent along to the destination system at 300 or 1200 baud. When the destination responds, of course, the whole process reverses itself.

With all this listening and packeting, there are some delays, and Richard Bear (the Springfield, Oregon fellow who turned me on to this new service) noted that the effective rate on his 1200 baud modem was about 600 baud.

Richard mentioned some bugs he'd found in the system.

"Occasionally Telenet just seems to drop me. This doesn't happen very often, but it does happen. Also, after connecting and releasing three areas, I get dropped automatically. If I stay on

for a long time (over 2 1/2 hours), I can also get dropped. Of course, I can get right back in."

"Despite these problems, PC pursuit does provide a very affordable way to access the country."

PC Pursuit Newsletter.

Richard sent along a PC Pursuit newsletter, and one of the most significant bits I picked up is that the network is set up for 7 bits per character, even parity, one stop bit.

The newsletter says that you can switch to 8 bits, no parity, for data transfers. And it adds that you can transfer information using XMODEM or any of the other 8-bit protocols that the board at the other end supports. (Which makes sense; the CRC or checksum looks like ordinary data to the network.)

Telenet

I called the Telenet folks. I talked first to customer support (the technical types). There a young fellow told me that the system could not support 8 bits, no parity.

"Why?" I asked.

"Because that's the way the software was written," he said, and then paused to let the full gravity of what he had said sink in.

"Then it can be changed," I said, waiting to see how he would handle a backspin lob.

He ducked.

I spent the next hour warming my phone's earpiece as I was passed from extension to extension. I could just hear them whispering in the background. "Watch out for line 5, he thinks we can change software."

Finally, I wound up in marketing.

Marketing is usually made up of engineers who have lost touch with technology, but in this case I was lucky.

"You want to log on in 8-bits, no parity?"

"Yes."

"When you log on, you enter:"

```
<cr>H<cr>
terminal = (you enter)U1<cr>
@C <area code>/3(or)/
    12,<userID><cr>
password (enter password)<cr>
connected (you enter)ATZ
ATDT, <7 digit phone number of
board>
```

Special note:

The notes in parentheses are my comments. The characters inside the <>'s are carriage returns <cr> or your own password, etc. /3 tells the system you want 300 baud, /12 is 1200 baud.

PC Pursuit will support 2400 baud to 80 of the dialup nodes (out of thousands) by the first of the year.

I mentioned Richard's problems when I talked to the Pursuit folks. They indicated that they had been having some startup problems but said that most of them had been fixed.

Whether they are fixed yet or not, this is quite a service. How else could you totally withdraw from society for only \$25 per month? How else could you call all over the U.S. for almost nothing without having to change phone booths every hour?

The only disadvantage I see to this service is that we don't have a node in Bend. On the other hand, with the magazine and all, it might just be an advantage. A big advantage.

PC Pursuit
1-800-368-4215 Voice
1-800-835-3001 FIDO — RBBS

■ ■ ■

Ever Wondered What Makes **TURBO PASCAL** 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 Code Busters"

The SCG-TP program produces
fully commented and labeled
source code for your TURBO-
Pascal system. To modify,
just edit and assemble. Version 3.0xA (280) is \$45.
SCG's available for CP/M 2.2 (\$45) and CP/M+ (\$75).
Please include \$1.50 postage (in Calif add 6.5%).

C. C. Software, 1907 Alvarado Ave.
Walnut Creek, CA 94596 (415) 939-8153

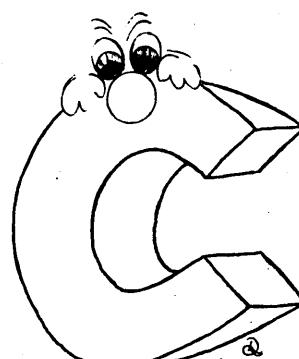
CP/M is a registered trademark of Digital Research, Inc.
TURBO Pascal is a trademark of Borland International

Don't miss it!



— Issue #34

for



contest

results!

Ron Miller
1157 Ellison Dr.
Pensacola, FL 32503

Text Displays, Fast And Fancy

At the heart of every C there's a printf(). In many ways printf() is indispensable. However, it's slow and its characters have very few good attributes. Ron has a fast fix.

As readers of my last column might have noted, I've been using C more and more often for practical purposes. That worries me a little. As an occasional English professor I've taken great comfort in the fact that my course on (say) *Paradise Lost* offers students no practical benefit whatsoever. The best things in life are not the means to some end but the ends in themselves. "Relevant?!" I love to bellow. "You're lucky if you turn out to be relevant to John Milton." *Editor's note: Who?*

Ever since I discovered the beauty of C I have thought of it as one of these goods-in-themselves, not quite like *Paradise Lost*, maybe, but not like dental floss or box wrenches, either.

Consider the elegance of the Kernighan and Ritchie version of the standard strcpy() function:

```
strcpy(t,s)
    char *t,*s;
{
    while(*s++ = *t++);
}
```

Such a wedding of form and function is a joy to concoct and a joy to compile. Where else in this fallen, messy world does one get that feeling, that "Well done thou good and faithful servant?" (Editor's note: I think he means it compiled without errors.)

Every software house can tell you, however, that C does workaday jobs quite well. And even I find myself reaching for C to write programs intended not just for myself but for others. With C I get smaller programs, clearer structures, and more explicit control over the operation of my utilities.

I also get some pretty lousy screen displays.

The Sins Of printf()

Though printf() allows extraordinary flexibility in formatting columns and displaying numbers — right justification, left justification, zero fill, signed and unsigned integers, hexadecimal representation — it does have its costs:

1. It is huge. Using printf() is like joining a record club. Order the song you want and you'll have to put up with all the extra records.

2. It is slow. Functions with lots of options also have numerous tests and jumps during a simple function call. Have you ever looked at the code of a full implementation of printf()?

3. It doesn't take advantage of direct writes to hardware. Since printf() does all its screen I/O through the operating system (for generality) it is slow and lacks lots of the goodies a CRT showman dreams of.

4. printf() uses strings. Even if you dump printf() and go to plain vanilla puts(), C insists upon displaying text one string at a time and that string really has to fit on one line. You'd have to do all kinds of unnatural acts to get a string to continue on to the next line, and the result would be more of a stunt than a practical option.

If all you are doing is writing:

```
puts("Save file? Answer: Y/N\n");
```

No problem. Anyone who has ever tried to get the right margin smooth on a whole stack of printf("%s",xx)'s in a row knows by painful experience, however, that printf() is not the tool for typing out lengthy passages of explanation or discussion.

Using A Word Processor Instead

My solution for this last or "text" problem in C programs is to create a text file with a word processor, which is more than willing to trim the mar-

gins and set the necessary number of carriage returns and free the programmer from the puzzle of "What will this look like on the screen?" That of course creates another problem, "How do I get this text file into my program?" This second puzzle, however, is easier to solve and leads to ways to get around the other printf() problems.

I've tried three ways to get word processor text (in the non-document mode, of course: no control characters unless you want smiley faces on the screen) into a C program.

1. Read the file sequentially with fgetc() and display the characters as they come up, stopping either at EOF or some other, predetermined, character. If you stop at a special character, the file pointer can save your place so the program will be ready to write the next screen.

2. Use read() to load the entire text file into an array and then scan the array with a character pointer. Let the character pointer (a static, of course) keep the place.

3. Create an .EXE file with blank space in it and use another C program to read the .EXE file into memory, find the blank space, fill it with text, and rewrite the .EXE file to disk to create a program with one giant initialized array.

Each of these strategies has limitations:

1. Character by character stream i/o is about as slow as printf(). Without a bunch of seek() games, which don't work too well on text files anyway, random access is impossible.

2. The load using low level i/o is fast, and access is easily made random with a character pointer, but you'll still have to haul two files around to run the program. The fewer the files, the fewer that get lost.

3. The best solution — if you can force your compiler to insert long patches of blank space into the .EXE file. I've used "DB NN dup (?)"

Figure 1 - Fast Display Utility

successfully with compilers that employ the Microsoft linker. It is a bother, however, to run the assembler, link the files, and create and remember to use a loader program.

Beating The Operating System

Once you've stuffed the word processor text into memory, the bottleneck in MS/PC-DOS becomes the operating system itself. A single putchar() calls interrupt 21h, which calls interrupt 10h, which performs an equipment check, pops the registers a handful of times, and then writes the character. (And we haven't even written the attribute byte.) If we knew the number of times DOS long jumps to get one darned character plugged into screen memory, we'd probably all write nasty letters to Redmond, Washington.

All that rigmarole buys you machine independence. But, if you are willing to be limited to clones, you can write directly to the screen.

If you're writing directly to the screen you'll need to know where the screen is so you know where to poke characters (and attributes).

Take a look at the first two lines of code in Figure 1 to see how I used interrupt 17 (11h) to ferret out the video RAM address. On return, the interrupt reports the machine's equipment in the AX register. Bits 4 and 5 indicate the type of video board (you set these bits with the system DIP switch). If bits 4 and 5 are high, then you have a monochrome card; if 4 is low and 5 is high, you have an 80-column color card, otherwise you should get your money back — your system thinks it's a jr. with a 40 column monitor. A monochrome card's memory lies at B0000, color memory lies at B8000.

If you are using a large memory model of C, a 32-bit pointer will do the job nicely. All you have to do is treat the screen area as an 80X25 integer array, with the high bytes of the integers as attributes and the low as characters. Poking integers is by no means a time-consuming operation.

If you want to fiddle with characters and attributes from the cozy confines of a small model C, you can dedicate a

(continued next page)

```
#include <regs.h>                                /* or however you generate interrupts */
#define TEXTSPACE ???                                /* however long the total text is */
#define SCREENCOUNT ???
unsigned scrseg;
char *charptr,*place[SCREENCOUNT], /* place[] = pointers to screens */
      buff[4000],texthold[TEXTSPACE];
main()
{
    struct regs rr;
    int i,fhand;

    interrupt(0x11,rr);
    scrseg=((rr.ax >> 4) & 3 == 3)? 0xb000:0xb800; /* which CRT adapter? */
    fhand=open("text.txt",0); /* load text area from word processor file */
    read(fhand,texthold,TEXTSPACE);
    close(fhand);
    place[0]=charptr=texthold;
    for(i=1;i<SCREENCOUNT;i++){ /* setting addresses of screens */
        while(*charptr++ != '~'); /* tilde divides screens */
        place[i] = charptr;
    }
    * * * * * /* this is the area where the program */
    text(QQ); /* would do something useful. */

    text(RR);
    * * * * *

}
/*********/
text(nn)
int nn;
{
    char att,*wptr;
    int y,i,*iptr;

    for(i=0,iptr=buf; i<2000;i++) 0x0720; /* blank screen buffer */
    for(charptr = place[nn]; *charptr != 0xa;) if(*charptr++ == 26) return;
    /* to end of line: this gives me an area for comments */
    for(att=0x7,y=0,wptr=buf;;charptr++) switch(*charptr){
        case 0xa: wptr = &buf[++y*160];
            /* point pointer to beginning of new line */
            if(y<25) break; /* fall through if screen too large */
        case '~': longmove(scrseg,0,buf,4000);
            /* End of screen. You could use port calls to the 6845 to
               blank the screen and avoid flicker. If you do, remember
               that the control port is 0x3d8 for color/graphics and
               0x3b8 for monochrome. More hardware dependencies! */
            return;
        case 0xd: break; /* some WP's don't include 0xd's */
        case '^': att ^= 8; /* toggling intensity bit: 0xf<->0x7*/
            break;
        default: *wptr++ = *charptr; /*insert character */
            *wptr++ = att; /*and attribute */
            break;
    }
}
/*********/
longmove(segment,offset,buffer,count)
unsigned seg,offset,ct;
char *buffer;
{
/* This one depends on
   a. How your version of C handles the stack.
   b. What sort of assembler you are using.

   I'll just give assembly pseudocode. You'd probably be
   using [BP+?] addressing */

#asm
    mov    es,segment
    mov    di,offset
    mov    si,buffer      ;ds is already set
    mov    cx,count
    cld
    rep    movsb          ;increment si & di with each loop
#endasm
}
```

(continued from page 67)

"screen" array down in the data segment. Then you can toss the whole array up to screen memory with an intersegmental move function based on the speedy "rep movsb" utility of the Intel chips.

If you read the original text from memory with one character pointer and write characters and attributes into "screen" memory with another, a screen chock full of text can be processed and displayed in the blink of an eye. Plus, you have COMPLETE control over highlighting, color, underlining, reverse video, blinking, and whatever else your monitor adapter allows and your conscience permits.

Then, you only have to transfer the "screen" to real screen memory to see it on the monitor.

When I'm creating text screens I use some of the less common characters to flag attributes for the display utility. For example, in Figure 1, the caret (^) toggles between highlighted and nor-

mal video and the tilde separates pages. You could use the vertical bar (!) to toggle reverse video, and the backwards single quote ('') to toggle the blink. If you're curious about which attributes you have, just whip out your tech manual and look in the index under "Video."

Noble Code, Vulgar Display

See Figure 1 for the fast display utility. Notice that in this sample, I load the word processor text into memory at the beginning.

My screens are much, much faster than they were when I struggled with printf(). You cannot imagine what unspeakably flashy, gaudy, and eye catching screens I can stuff into a piece of otherwise practical software. It sure is fun to generate my vulgarities with a most elegant run of C. One can't let oneself be too much of this world, you know.



HiSpeed C Library

The HiSpeed replacement library for CP/M Toolworks C/80 uses either Microsoft Macro-80/Link-80 or SLR Systems SLRMAC/SLRNK.

Compile HELLO.C into a 254-byte HELLO.COM in 10 seconds with a single command line.

Features:

- Fast K&R incremental development.
- Full STDIO stream library: fseek, fwrite, etc.
- Function calls match major MS-DOS, CP/M-68K and UNIX compilers. Painless portable code.
- Full math, strings, searching, sorting, timing.
- One function per page documentation with examples that work (410 pages, 310+ functions).
- Utilities to archive, bold-print, beautify, cross-reference, function map, . . . (27 in all).

Printed reference manual and object code \$29.95

One megabyte of source code on disk \$19.95

Free brochure. Phone orders shipped immediately.
All Media Master formats and 8" ssdd.

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



VIKING SOFTWARE

BARGAIN

68000 OPERATING SYSTEM

\$ 50

Now you can put K-OS ONE, a 'simply' great operating system, on your 68000. It is simple to use and has all of the features you would expect from an operating system in the class of MS-DOS or CP/M. It's easy to customize because you get source code for the operating system and the command processor.

They are written in HTPL, a powerful, high level, language hybrid. You don't have HTPL? No problem, the HTPL compiler is included. The K-OS ONE operating system is small. You can edit and re-compile the whole operating system in only 64K of RAM.

The package also contains a line editor, a 68000 assembler and a manual covering the whole package.

With K-OS ONE, you can read and write ASCII files on MS-DOS format diskettes. This gives you the tools you will need to port K-OS ONE to your hardware.

CP/M is a Trademark of Digital Research
MS-DOS is a Trademark of Microsoft Corporation
K-OS ONE and HTPL are Trademarks of Hawthorne Technology

To order, call: (503) 254-2005

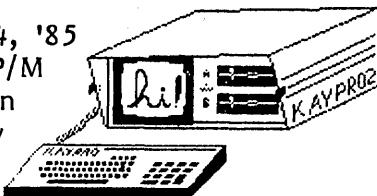


Hawthorne Technology

8836 SE Stark
Portland, Or 97216

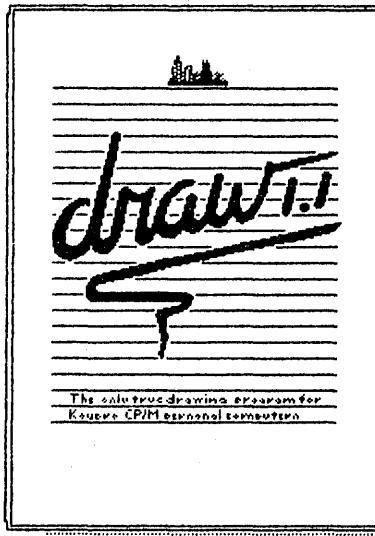
Graphics for the Rest of Us

If you own an '84, '85 or '86 Kaypro CP/M computer, you can quickly and easily draw diagrams and illustrations like those shown here. All you need is the right software: SCS-Draw.



SCS-Draw turns your mild-mannered Kaypro into a powerful drawing machine, with features like:

- * True pixel-by-pixel drawing and editing capability.
- * Multi-screen drawings, for a total resolution of over 120,000 pixels. (336x362)
- * Powerful printing options like rotation, enlargement, and mirror image.
- * Block move and file merge.
- * Context-sensitive help, pop-up menus and a 96-page user's guide. (Illustrated, of course.)
- * NEW in Draw 1.1: X/Y coordinate display, vertical text for graph axis labels, print to disk capability, and much more.



"A fabulous program... SCS-Draw permits you to draw to the best of your abilities -- and your computer's."

- Profiles, March 1986

"A great way to fly... the power of this program is unlimited."

- Cornerstone Magazine, June 1986

"Extremely easy to use... excellent for creating logos, bulletins and newsletters."

- Clergy KUG, July 1986

"The documentation is among the best I've seen."

- MicroTimes, August 1986

"Best use I have seen of the screen graphics."

- SCS-Draw user, Australia

"The best piece of commercial software I have bought... the quality is 4.0."

- SCS-Draw user, Idaho

"A pleasure to use... user-friendly, fun, well put together."

- SCS-Draw user, Michigan

"SCS-Draw is the first program for the Kaypro that I have truly enjoyed."

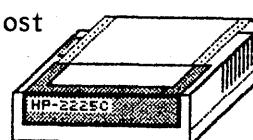
- SCS-Draw user, Kentucky

Get in on the excitement — get your copy of SCS-Draw today.

- [] Send me SCS-Draw immediately by UPS 2nd-Day Air. Payment of \$59.95 is enclosed.
- [] I have to see it to believe it -- send me a copy of the SCS-Draw Demo Program. I have enclosed \$5.00 for shipping and handling.

(Specify Kaypro model when ordering.)

SCS-Draw supports most dot-matrix printers, including Okidata, Epson, Star Gemini, C. Itoh, and IBM.



Daisywheel printers from Juki, Diablo and Brother are also supported, and a custom installation program is provided for printers not listed here.

**Second City Software 312-577-7680
P.O. Box 442
Mount Prospect, IL 60056**

This page was created and printed using only SCS-Draw and WordStar.

PASCAL PROCEDURES

By John P. Jones

6245 Columbia Ave.
St. Louis, MO 63139
(314)645-1596

Queues, Stacks, And Residents

Queues are good for more than just spelling contests. Here John takes another look at queues (and stacks). Then he looks at writing TSRs (terminate but stay resident routines), those neat memory resident programs that pop up when you least expect them. (Now if you auto-installed one that did something different every time it saw a carriage return...)

In issue 25 we talked about dynamic variables and pointers, particularly their use in list and tree structures. For queues and stacks of indeterminate length, this is the method of choice. But when you know the maximum size, you can implement a stack or queue more easily with an array and one or two indexes.

First let's review just what a queue is. There are two major types of queue — the FIFO (for First In First Out), and the LIFO (Last In First Out). A FIFO is a list, designed so that data can be added at one end, and later retrieved and deleted from the other end. The last time you were at the bank on a Friday evening you were in a FIFO queue.

In a LIFO queue, data is added to one end, and later retrieved from the same end. LIFOs are often referred to as stacks.

Either structure has three possible states: empty, not empty, and full.

LIFOs

To set up a LIFO, you need a data array and one index. Since the next item to be retrieved is always the last item added, the same index can be used for both adds and removes. Figure 1 shows an implementation of a simple LIFO.

The boolean functions PutOnQ and GetFromQ add data to and remove data from the queue. PutOnQ first checks to see if the queue is full, and

Figure 1 - Last In, First Out Queue

```
Program LIFOdemo;
const
  qsize = 20;

var
  queue : array [0..qsize] of char; { must be global vars since can be }
  qptr : 0..qsize;                 { asynchronously updated }
  ch2, ch3 : char;

function PutOnQ (ch:char):boolean;
begin
  if succ(qptr) > qsize then { queue full? }
    PutOnQ := false           { yes, signal failure }
  else begin
    qptr := succ(qptr);       { not full, update pointer & add data }
    queue[qptr] := ch;
    PutOnQ := true;
  end;
end;

function GetFromQ (var ch:char):boolean;
begin
  if qptr = 0 then            { queue empty? }
    GetFromQ := false          { yes, signal failure }
  else begin
    ch := queue[qptr];         { else get data & update pointer }
    qptr := pred(qptr);
    GetFromQ := true;
  end;
end;

procedure ProcessInput (ch:char); { not really much processing here! }
begin
  write(ch);
end;

begin
  qptr := 0;
  ch2 := 'E';                  { pred('A') }

  repeat
    ch2 := succ(ch2);

{ the following statement is to demonstrate the effects of async removal }
{   if random(10) > 3 then
      if GetFromQ(ch3) then ProcessInput(ch3); }

    until not(PutOnQ(ch2));

  while GetFromQ(ch2) do
    ProcessInput(ch2);
end.
```

Figure 2 - First In, First Out Queue

```
Program FIFOdemo;
const
  qsize = 20;

var
  queue : array [0..qsize] of char; { must be global vars }
  q_top_ptr, q_bot_ptr : 0..qsize;
  ch2, ch3 : char;

function PutOnQ (ch:char):boolean;
begin
  if succ(q_bot_ptr) mod (qsize + 1) = q_top_ptr then { queue full? }
    PutOnQ := false { if yes, can't add }
  else begin
    q_bot_ptr := succ(q_bot_ptr) mod (qsize + 1); { update bottom pointer}
    queue[q_bot_ptr] := ch; { store the data }
    PutOnQ := true; { signal success }
  end;
end;

function GetFromQ (var ch:char):boolean;
begin
  if q_top_ptr = q_bot_ptr then { queue empty? }
    GetFromQ := false { if yes, then nothing to get }
  else begin
    q_top_ptr := succ(q_top_ptr) mod (qsize + 1); { update top pointer }
    ch := queue[q_top_ptr]; { get the data }
    GetFromQ := true; { signal success }
  end;
end;

procedure ProcessInput (ch:char); { not a lot of processing here! }
begin
  write(ch);
end;

begin { Program FIFO }
  q_top_ptr := 0; { initialize pointers }
  q_bot_ptr := 0; { pred ('A') for demo }

  repeat
    ch2 := succ(ch2); { bump input data }

{ the following statement is to show asynchronous removal }
{   if random(10) > 3 then
      if GetFromQ(ch3) then ProcessInput(ch3); }

    until not(PutOnQ(ch2));

    while GetFromQ(ch2) do
      ProcessInput(ch2);
end.
```

exits with no action if it is. If space is available, the array index is updated, and the data stored. GetFromQ is, as you would expect, the reverse of PutOnQ. It first checks for the empty state, and if false, returns the last datum added to the stack. You may notice that the first array element (at [0]) is unused because the index is updated before data addition, and after data removal. If the reverse is done, the last array element will be unused.

I've included an asynchronous removal statement, which randomly removes about 1/3 of the data items right after they are added. Try the program both with and without it. As you will see, without async removal, a LIFO can be used to reverse the list without using recursion. However, when elements are being removed at random times during input, the output becomes jumbled.

FIFOs

A FIFO is what we generally mean when we talk about a queue. To implement a FIFO with an array, we need the array itself, and two index variables for adding and retrieving the data. A FIFO is most efficiently organized as a circular queue — a new item is stored in the first element of the array after the last element has been filled (if the queue is not full).

Figure 2 is a demonstration of a simple FIFO. Data is always added at the bottom of the queue with Q_BOT_PTR and removed from the top with Q_TOP_PTR. Each time we increment an index, we divide it by qsize + 1 and keep the remainder (modulo arithmetic). Thus we reset the index to zero whenever its value reaches qsize + 1.

The queue is full when the current bottom position is 1 less than the current top position. (So any new data

(continued next page)

PASCAL PROCEDURES

(continued from page 71)

would have to be written over the data in the top position.) When the current bottom position equals the current top position the queue is empty.

When you run this program with and without the async removal statement you will see that there is no difference in the sequence of the output, only in its length. The data will always be available in exactly the same order it was queued.

FIFOs are especially useful for data transfers between asynchronous processes. For example, you might have an interrupt-driven data capture routine which cannot process the data immediately. If the data capture routine queues the data in a FIFO, the processing routine can retrieve it in proper sequence at a later time.

TSRs

MS-DOS was designed to be extended on the fly using the "terminate but stay resident" function. Put simply, after initialization, a resident program will hook into one or more interrupt vectors, then invoke the terminate but stay resident DOS interrupt. The mechanism was designed for programs like MODE, one of whose functions is to add a serial printer driver to the operating system.

Unless you've been living in a closet for the last 3 or 4 years, you'll be aware that many software publishers have used this mechanism for pop-up productivity enhancement programs. (Borland's Sidekick, Superkey, and Turbo Lightning are all examples.) These programs wait quietly in memory until invoked by a particular keystroke or event, then take over from the currently running program until exited again. If the utility is implemented correctly, you should get back into your original program precisely where you left off, with no loss of data.

You can do the same thing yourself with a Turbo Pascal program! There is a set of routines available for downloading from the Borland SIG on CompuServe (and I assume, other BB's) which you can incorporate into your program to allow it to be a TSR. Within about 2 hours after downloading the files, I had a resident Hercules graphics screen dump to printer pro-

gram running. (See Issue 30 for the screen print procedure.)

The package includes several files within a single .ARC (archive) file:

STAYRES.340 - the 'main' program that sets up residency; your routine will be activated by it when the proper key is pressed.

STAYSAVE.340 STAYRSTR.340 - save and restore stack & registers when the program is activated or exited. Needed because DOS is not re-entrant and I/O would be impossible without this environment save.

STAYXIT.340 - removes the resident program from memory. (Only to be used if yours is the LAST resident program!!! DOS will go bananas if you remove a TSR from the MIDDLE of a group.)

STAYWND.340 - routines to create & use a window, then restore original screen before exit.

STAYSUBS.340 - routines to retrieve current directory & display it. Used by the demo program below.

STAYDEM.340 - demo of system — pops-up a directory window.

There are no documentation files for the routines, but all are liberally commented with explanations and instructions. Understandably, the core routines are mostly INLINE machine code. Although not recommended, it is more than possible to cookbook a TSR program with these routines without understanding what is really going on. My graphics dump was done this way, but I did go back later to learn more about the process.

I recommend that you test any TSRs you write very thoroughly, in the environment where they'll be used, before you become dependent on them. When another particular TSR is resident with the screen dump, I can crash the system about half the time if I run Generic CADD.

You should be able to learn quite a bit about DOS if you play around with this set of routines and a good book on DOS. (I just came back to this after a session on CompuServe — there is apparently a newer version of the resident routines; they are contained in the file STAY42.ARC in the DOS Turbo Pascal data library of the Borland SIG. You may want to check it out.)

STAY42.ARC is also available on the

Micro C board (503-382-7643) in the CURRENT ISSUE LISTINGS.

Things To Come..

Although Pascal is a very capable programming language, and Turbo Pascal is an especially efficient implementation of Pascal, neither is perfect. One of the more serious limitations is the lack of facilities for separate compilation of routines and routine libraries. It is NOT the language of choice for a large programming project (>4000 lines of code or so), and definitely not suitable if it's a team effort.

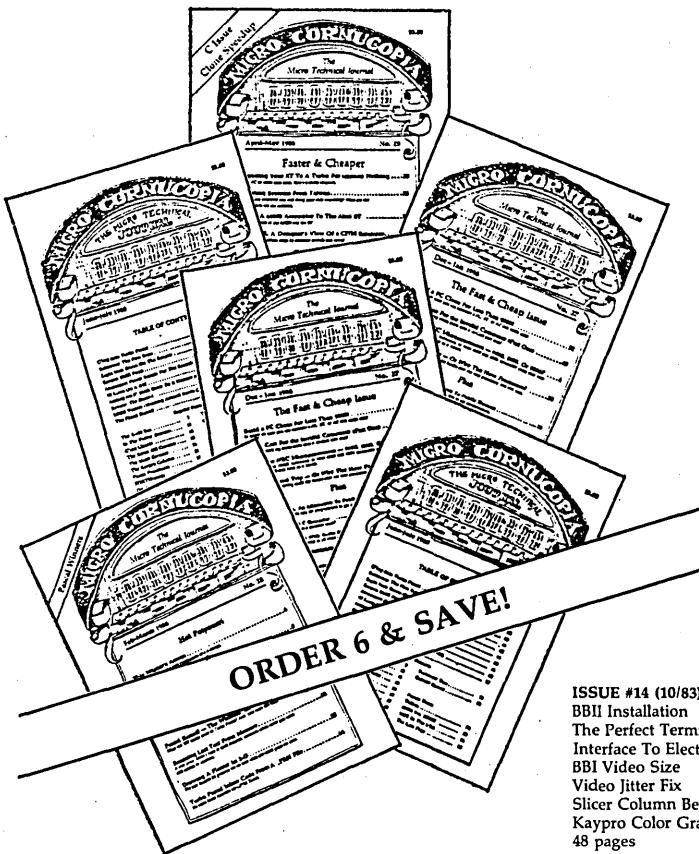
Modula-2 was developed to be the successor to Pascal and was designed to be a true general purpose language for everything from operating systems to applications.

I have begun to learn Modula-2, and plan to include some of my experiences in this column. Next time, we'll take a look at how Modula-2 compares with Pascal and we'll check out a couple of Modula-2 compilers.



Complete Your Education

If you're not working with a full set,
fill out your collection of Micro C today.


ISSUE #1 (8/81)

Power Supply
RAM Protection
Video Wiggle
1/2 PFM.PRN
16 pages

ISSUE #2 (10/81)

Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
1/2 PFM.PRN
16 pages

ISSUE #3 (12/81)

4 MHz Mods
Configuring Modem 7
Safer Formatter
Reverse Video Cursor
FORTHwords Begins
16 pages

ISSUE #4 (2/82)

Keyboard Translation
More 4 MHz Mods
Modems, Lync, and SIOS
Undoing CP/M ERASE
Keyboard Encoder
20 pages

ISSUE #5 (4/82)

Word Processing
Two Great Spells
Two Text Editors
Double Density Review
Scribble, A Formatter
20 pages

ISSUE #6 (6/82)

BBI EPROM Programmer
Customize Your Chars
Double Density Update
Self-Loading ROM
Terminal In FORTH
24 pages

ISSUE #7 (8/82)

6 Reviews Of C
Adding 6K of RAM
Viewing 50 Hz
On Your Own Begins
24 pages

ISSUE #8 (10/82)

Drive Maintenance
Interfacing Drives
Installing A New BIOS
Flippy Floppies
C'ing Clearly Begins
Xerox 820 Begins
28 pages

ISSUE #9 (12/82)

BBI EPROM Program
Relocating Your CP/M
Serial Print Driver
Big Board I Fixes
Bringing Up WordStar
Cheap RAM Disk
32 pages

ISSUE #10 (2/83)

Saving A Flakey Disk
Hooking Up BBI
The Disk I-O Doctor
JRT Fix
Serial Board Interface
Pas. Procedures Begins
36 pages

ISSUE #11 (4/83)

BBI Expansions
BBI Details
Dyna, RAM Disk Review
Easier Reverse Video Cursor
Planning Review
Kaypro Column Begins
36 pages

ISSUE #12 (6/83)

256K For BBI
Bringing UP BBII
dBase II
Look At WordStar
Double Sided Drives For BBI
Packet Radio
5 MHz For Kaypro
40 pages

ISSUE #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 #14 (10/83)
BBI Installation
The Perfect Terminal
Interface To Electronic Typewriter
BBI Video Size
Video Jitter Fix
Slicer Column Begins
Kaypro Color Graphics Review
48 pages

ISSUE #15 (12/83)
Screen Dump Listing
Fixing Serial Ports
Playing Adventure
SBASIC Column Begins
Upgrading Kaypro II to 4
Upgrading Kaypro 4 to 8
48 pages

ISSUE #16 (12/84)
Xerox 820 Column Restarts
BBI Double Density
BBI 5'/8' Interface Fix
Kaypro ZCPR Patch
Adding Joystick To Color Graphics
Recovering Text From Memory
52 pages

ISSUE #17 (4/84)
Voice Synthesizer
820 RAM Disk
Kaypro Morse Code Interface
68000-Based System Review
Inside CP/M 86
56 pages

ISSUE #18 (6/84)
Kaypro EPROM Programmer
I/O Byte: A Primer
Kaypro Joystick
Serial To Parallel Interface
Business COBOL
60 pages

ISSUE #19 (8/84)
Adding Winchester To BBII
6 MHz On The BBI
Bulletin Boards
Track Buffering On Slicer
4 MHz For The 820-1
64 pages

ISSUE #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 #21 (12/84)
Analog To Digital Interface
Installing Turbo Pascal
Low Intensity BBI Video
Turbo Pascal, The Early Days
80 pages

ISSUE #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 #23 (4/85)
Automatic Disk Relogging
Interrupt Driven Serial Printer
Low Cost EPROM Eraser
Smart Video Controller
Review: MicroSphere RAM Disk
Future Tense Begins
86 pages

ISSUE #24 (6/85)
C'ing Into Turbo Pascal
8' Drives On The Kaypro
48 Lines On A BBI
68000 Versus 80X86
Soldering: The First Steps
88 pages

ISSUE #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 #26 (10/85)
Inside ZCPR3
Two Megabytes On DS-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)	

Super Chips & Parallel Architectures

By Gary Entsminger
1912 Haussler Dr
Davis, CA 95616

Designs For The Future

Hypocubes, doughnuts, wafers, Josephson junctions, fuzzy inferers, transputers, LISP and PROLOG chips, distributed networks, and stream parallelism are just a few of the new names in computer engineering.

If "fuzzy" is a word you associate with stuffed animals, and "doughnut" implies something sweet to eat, or you're reading Micro Cornucopia, then you'll love this story.

The computer hasn't changed much (seriously, folks) since von Neumann designed the IAS Machine (built at Princeton's Institute for Advanced Study about 1947). That is in one sense, computers — IAS Machine, the IBM 7094, the IBM 360, the Cray-1 right on through to the "Macintosh" and "Amiga" — have processed instructions sequentially.

A CPU fetches a pair of numbers from memory, adds them, and returns the result to memory. How fast the CPU can fetch is very important, but processing time is determined not only by CPU speed, but also by the narrow path between the CPU and memory.

Bottleneck

The high speed shuttle of data and results between CPU and memory ultimately leads to a jam that would make the air traffic controllers at O'Hare International take notice. It's known as von Neumann's Bottleneck, and it can't be avoided.

You can speed up the processor, and you can widen the path, but eventually something (memory speed or the like) will jam the traffic. It's a problem, and it's been inherent in computers since the beginning.

Until recently, the big players (the IBMs, the Control Datas, the Crays) have compromised with the bottleneck, refining computers on the basis

of a fundamental principle — "Squeeze larger and larger numbers of smaller and smaller components onto a piece of silicon." Whatever computer you use (micro, mini, or mainframe) incorporates this principle.

Unfortunately, there are limits to how long you can apply the principle. Researchers working in the submicron range have reported quantum fluctuations, unacceptable roughness on the etched surfaces of chips. And submicron transistors are extremely sensitive to low-level radiation. A stray ray can switch a transistor from on to off. Not acceptable.

Also, as wire diameter decreases, resistance to current flow increases. To overcome resistance, the voltage is increased, generating more (and more) heat. Eventually, you spend a lot more money to get a little more speed.

Alternative Architectures

In order to produce faster, more cost-efficient computers, the big players (the Seymour Crays, the Lawrence Snyders) must either find new materials for components or explore alternative architectures.

In the last five years, work has begun or continued on exotics like — gallium arsenide semiconductors, Josephson junctions, organic transistors, transphasors, and wafer scale integration.

Research and development of new technology has become so expensive that the big players now aren't just companies, they're countries.

Let's consider the major efforts to refine and redefine computing in America, Europe, and Japan.

In America

As usual, and perhaps counter to what you've heard, the good ol' USA is still pacing laps for Europe and Japan, but Europe and Japan are catching up.

Important attempts to produce fast-

er, conceptually different, computers are in the works (or working) at Cray, Control Data, Goodyear Aerospace Corporation, Denelcor, Symbolics, Texas Instruments, Apollo, Sun, IBM, AT&T, Tectronix, MIT, Cal Tech, NYU, Columbia, the Universities of California and Washington, and others.

Most designs incorporate some form of parallel (or multi-CPU) architecture. (The Cray 1, for example, has four processors and Control Data's Cyber-plus will eventually be expandable to 64 processors.)

Parallel Developments

In 1964, Seymour R. Cray, the leading designer at Control Data, introduced the first computer to use a parallel strategy within the CPU — Control Data's CDC 6600.

It consisted of ten separate parts, each of which specialized in one kind of logic or arithmetic, and a queue which looked ahead and redirected operations that could be best handled by a specialized function. Its memory was partitioned into 32 banks, and, in effect, executed individual programs in separate memories.

This original idea took the computing world by storm and dominated the mainframe industry until 1967 when IBM introduced the 360/91 which used another new concept, pipelining.

Pipelining

In a pipeline, a number of functions are going on at the same time. Imagine a data assembly-line where two numbers are fetched from memory while at the same time two other numbers are being summed and the results from a previous math operation are being sent back to memory.

In 1969, Control Data again passed IBM, using the same pipelining strategy, but shortening the clock cycle (to 27.5 nanoseconds).

Pipelining has been around ever

since, and is still used in almost all supercomputers.

In 1976, Cray introduced the first commercially successful vector processing computer, the Cray-1. With this technique, strings of related 64-bit numbers (vectors) can be processed in parallel. Each clock cycle processes one vector.

In addition, the Cray-1 was more compact, and the Cray-2 (now being tested) is even smaller. Processors have been reduced to minuscule size and packed tightly into layout boards stacked in banks. Communication lines (busses or wires) have been shortened. A combination which while producing the world's speediest computers, has dramatically pointed out the limits of the concept.

The Cray-1's 300,000 integrated circuits produce enough heat to fry a gigabyte omelet. To keep it cool enough to work, the refrigerant, Freon, is circulated through each board.

The Cray-2 (several Cray-1's combined in one system) has a 4.5 nanosecond clock. The individual CPUs function independently, but share data. And their instruction streams can be synchronized.

An up and running Cray-2 is expected to peak at about 1000 megaflops, compared to the Cray-1's 160.

Cosmic Cubes

Of course, a supercomputer isn't cheap. The Cray-1 sells for \$10 million plus, and fewer than 50 have been sold.

Another approach to parallel is cheaper. The Cosmic (or Hyper) Cube (designed at Cal Tech & licensed by Intel) uses 64 microcomputers, each with 8086s, 8087s, and 128K of memory, to create a network of nodes.

Each node equals one single-board computer. Data is processed independently and information is passed in packets between nodes. This distribution of memory is particularly useful in large-scale symbolic processes.

The current Cube is, at best, about a third as fast as the Cray-1, but anywhere from 10-15 times faster than a DEC VAX 11/780.

The speed of the system is very much dependent on the software. For example, how do we avoid idling

when one node finishes its task before another? For this or any other network system to work efficiently, programs have to be written to take advantage of the network. Which means more cooperation between hardware designers and software writers. Not an easy task.

LISP & Fuzzy

One attempt at intimacy is the dedicated LISP processor. At Symbolics, Texas Instruments, and others, a non-number crunching strategy is leading to a new age in language chips.

LISP, a list-processing language used in AI applications, is incorporated into the hardware to speed up the time-consuming searches inherent in AI applications. A similar strategy heavily influences computer design in Japan's well-publicized *Fifth Generation Project*.

Along somewhat similar lines, AT&T has announced plans to implement a non-commercial fuzzy inference processor on a chip in the near future.

Rules in fuzzy logic are based on concepts like "not very tall" which are directly implemented in VLSI hardware with tree structured parallel sets. Each rule has its own parallel execution path, so all rules are evaluated simultaneously.

AT&T has apparently already produced a chip capable of running at 20MHz and processing 80,000 fuzzy logical inferences per second. Fast for fuzzy.

In Japan

The Japanese have set up two major projects to produce super computers — the "Fifth Generation" and the lesser known, "Superspeed Computer."

The Fifth Generation Project is oriented toward artificial intelligence refinement. AI applications — expert systems, pattern-matching databases, natural language interfaces — employ time-consuming searching techniques that use up a lot of memory. PROLOG machines (with PROLOG operating systems) are now up, incorporating high-level PROLOG in the hardware design. In effect, teaching the hardware a new language. Making it speak software.

The lesser known Supercomputer project is aimed at producing a supercomputer capable of 10 million megaf-

lops (10 trillion floating point operations per second). It will have a billion bytes of primary memory — a memory path (bandwidth) of 1.5 billion bytes per second — and a distributed parallel-processing architecture. In short — a supercomputer to match the best in the US.

In Europe

A new kind of computer (the IMS T414) is already on sale in England. It's based on the transputer, a 32-bit RISC (reduced instruction set chip) processor. It includes 4 10-megabit-per-second full-duplex serial channels ("links") and 2K bytes of 50-ns static RAM on a single chip. The links run concurrently with the processor.

In the near future, transputers will have on-chip hardware floating point and will be capable of 1 megaflop at the low, low price of \$50 per chip.

Production versions of the transputer are about 4 times faster than the 68020 and 10 times faster than the 80286 when processing high level languages like LISP and PROLOG.

Future supercomputers on this front will be a box of a hundred or more parallel-processing transputers.

Wrap Up

The rush to create faster, more powerful computers is a rush for gold. The horizon is checkered with amazing machines. By the time you read this, even more ingenious ideas will be on the way to becoming computers. Maybe the architecture that will dominate computing in the 1990s is still a notion.

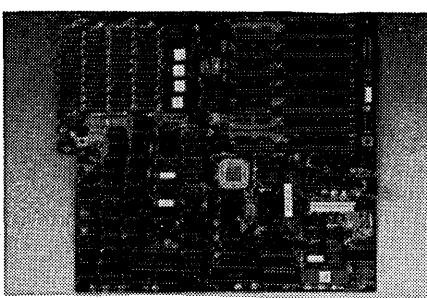
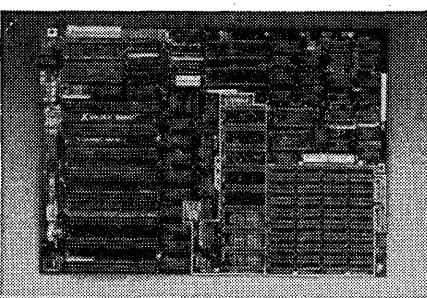
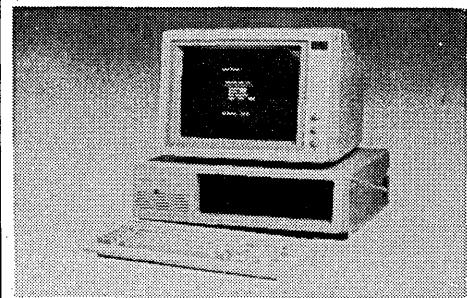
If you have a guess about the future, drop a line to Micro C's "Future Box", P.O. Box 223, Bend, OR 97709 or write me in Davis.

Although the big players will reap the benefits of the next wave first, you and I will definitely be affected.



DON'T BUY UNLESS IT IS FCC APPROVED

WE HAVE FCC APPROVED



Call us if you need specifications on add-on boards & accessories

BUILD YOUR OWN OR WE ASSEMBLE & TEST FOR YOU

PC/XT COMPATIBLE ADD ON & ACCESSORIES

MONITORS

RGB Color	\$350.00
TTL Monochrome amber/w/swivel base	105.00

SYSTEM COMPONENTS

XT Motherboard 0-640K (2 layers) 4.77 mhz

HARDWARE FOR IBM AT

AT compatible motherboard \$599.00

XT Motherboard 4.77/8 MHZ

8 slots 1 MB memory on boaqrd, expandable to 1 M

XT Case (flip-top)

Combined floppy/hard disk controller 225.00

XT Case (slide in)

Printer port/serial port card/game part 105.00

XT 150W power supply

200 watt power supply 130.00

XT 5160 keyboard (AT style)

1.2MB floppy disk drive 150.00

Floppy controller w/cable

3MB multifunction 240.00

Multi I/O card w/controller

OK expandable to 3M, parallel port, 2 serial ports,

384K Multifunction card (OK)

game port 80.00

I/O Plus II

AT type case 85.00

Color graphic card

AT compatible keyboard 85.00

Monochrome graphic w/printer port I

20MB hard disk, SEAGATE 4026 590.00

Enhanced graphic card

30MB hard disk, SEAGATE 4038 720.00

Serial Card

40MB hard disk, SEAGATE 4051 880.00

Parallel Card

ACCESSORIES

Surge Suppressed outlet strip 9.50

Ram Card

Surge Suppressed outlet strip (UL approved) 14.00

Color graphic w/printer card

Parallel printer cable/serial cable 11.00

Teac 55BV Floppy drive

Disk storage w/lock (holds 100 pcs) 14.50

Fujitsu Drive

A/B RS232 switch box 45.00

Assembly & Test

A/B Centronics switch box 45.00

joystick 26.90

HARD DISK SYSTEMS:

MODEM

300/1200 board 1/2 card modem Hayes compatible 140.00

Internal SEAGATE ST-225 20 MB hard disk drive

RAM

64k 150 NS 1.30

with controller

256k 150 NS 2.90

Internal 30MB hard disk drive with controller

DOS

DOS 3.1 40.00

Hard disk controller

DOS 3.2 w/GW BASIC 80.00

EPROM COPIER

DISKETTE (Bulk)

Eeprom copier, burns 2716, 2732, 2764, 27128, 27256

Double side double density 30.00/50

burns 4 eproms at a time

FOR DEALER & OEM INFORMATION, CALL

WONDARL, INC.

741 E. 14th St.

Oakland, CA 94606

TEL: (415) 839-1718

ORDER INFORMATION:

WE ACCEPT VISA, MASTER CARD,

COD, PREPAID

COD NEED 30% DEPOSIT

• Z Best Sellers •

Z-COM (7 disks) \$119.00
 Easy auto-installation complete Z-System for virtually any Z80 computer presently running CP/M 2.2. In minutes you can be running ZCPR3 and ZRDOS on your machine, enjoying the vast benefits. Includes 70+ utility programs and ZCPR3: The Manual.

Z-Tools (4 disks) \$150.00
 A bundle of software tools individually priced at \$260 total. Includes the ZAS Macro Assembler, ZDM debuggers, REVAS4 disassembler, and ITOZ/ZTOI source code converters. HD64180 support.

PUBLIC ZRDOS (1 disk) \$59.50*
 If you have acquired ZCPR3 for your Z80-compatible system and want to upgrade to full Z-System, all you need is ZRDOS. ZRDOS features elimination of control-C after disk change, public directories, faster execution than CP/M, archive status for easy backup, and more!

TERM III (6 disks) \$99.00*
 TERM III has been called "a cosmos". Why? Because it's 29 programs, featuring T3MASTER and T3SERVER, are galactic in scope. KERMIT, XMODEM, 1k XMODEM, and batch YMODEM protocols supported.

DSD (1 disk) \$129.95
 The premier debugger for your 8080, Z80, or HD64180 systems. Full screen, with windows for RAM, code listing, registers, and stack. We feature ZCPR3 versions of this professional debugger.

Quick Task (3 disks) \$249.00
 Z80/HD64180 multitasking realtime executive for embedded computer applications. Full source code, no run time fees, site license for development. Comparable to systems from \$2000 to \$40,000! Request our free Q-T Demonstration Program.

*ZCPR3 required.

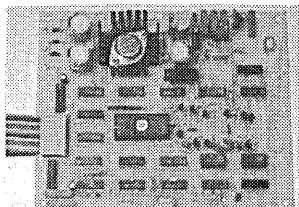
Z-System OEM inquiries invited.
 Visa/Mastercard accepted. Add \$4.00
 shipping/handling in North America,
 actual cost elsewhere.
 Specify disk format.



885 N. San Antonio Road • Los Altos, CA 94022
 415/948-3820 (Order line and tech support)

CP/M, MS-DOS EPROM PROGRAMMING SYSTEM

2708
 2758
 2716
 2516
 2532*
 2564*
 68764*
 2816A



2732
 2732A
 2764
 27128
 27256
 27512
 27CXX
 2864A

*ADAPTER SOCKET REQUIRED

- STAND ALONE BOARD USES NO BACKPLANE SLOT - FAST ALGORITHM
- NO PERSONALITY MODULES - INSTALL PROGRAM FOR SOFTWARE
- USES 24 VOLT XPMR FOR POWER - LARGE COMPREHENSIVE MANUAL
- PROGRAMS 26,25,21 & 12.5V 8/EPROMS - ALL SUPPLIES ON BOARD
- NOT A SERIAL PROGRAMMER, NO FILE DOWNLOADING REQUIRED

** PARALLEL PRINTER INTERFACE **

CONNECTS TO ANY PARALLEL PRINTER INTERFACE - USES 8 OUTPUT DATA BITS AND ONE BIT (THE BUSY LINE) FOR DATA INPUT.

** CONTROL PROGRAM COMMANDS **

- PROGRAM EPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- READ EPROM(S) INTO RAM
- VERIFY EPROM IS BRASED
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 11 SUB COMMANDS
- FILL-DUMP-XPER-EXAM-MODIFY-BIAS-PRGM-VERIFY-CKSUM, 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 --> \$199
 PARTS KIT WITH SOFTWARE AND DOC-->\$179 PCB, SOFTWARE & DOC-->\$69
 SOFTWARE ON 8 & 5 1/4 DISK FOR KAYPRO, AMPRO, IBM & OTHER FMT'S

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% TAX -- VISA/M.C. ACCEPTED -- \$3.00 FOR COD

LEAP INTO A NEW DIMENSION



WITH AZTEC CII

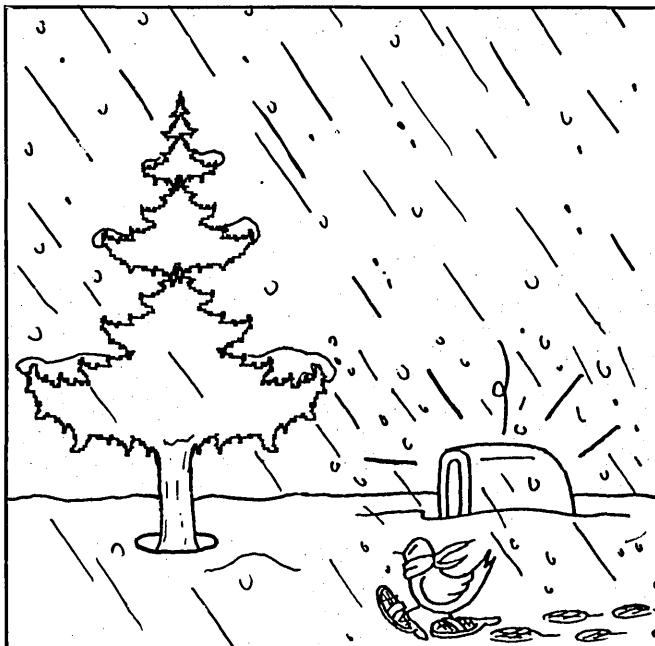
Full implementation of "C" with standard floating point, library, and I/O subroutines. UNIX VER 7 compatible. Produces relocatable 8080 (optional 0Z80) assembler code. Relocating assembler and linker supplied with package or use Microsoft M80 and L80, SID/ZSID debugger interface. FAST COMPILATION AND EXECUTION.

AZTEC CII FOR CP/M \$199

(Special price for Micro C subscribers \$149)

MANX
 software systems
 Box 55, Shrewsbury, NJ. 07701
 (201) 780-4004

Also available for Apple DOS, HDOS, CP/M-86, PC-DOS



*Happy Holidays
 from
 Micro C.*

(Fractal by Fogg)

By Dave Hardy

736 Notre Dame
Grosse Pointe, MI 48203

Driving Solid State Relays

As Bruce Eckel teaches us electronics by showing how to control stepper motors, Dave Hardy gives us a look at using solid-state relays to control high voltage, even 110VAC. Both give us an inside look at outside control.

We haven't talked very much about how to interface an S-100 machine to the real world. Of course, almost every S-100 talks to the real world via its system console or modem or printer, but I'm talking about the kind of interface that lets the S-100 do more than conventional I/O.

Many S-100 machines, for example, are used as laboratory or industrial controllers, or instrumentation. In fact, the S-100's capability for this kind of application is just about limitless.

In the last few years, I've received hundreds of real world applications from readers, including some which are so bizarre they defy description. One reader uses an S-100 frame to monitor and control building temperatures on his duck farm (or is it ranch?), while another, in a slightly different business, uses an S-100 machine to measure blood flow in laboratory animals, and to monitor strain gauges. S-100 frames have been used to control instrumentation in airplanes and ships, and have even been used in yacht racing and sail manufacturing. The list of applications seems endless.

Fortunately, I've pared the list down a bit (those of you with duck farms will have to wing it). The circuits shown here will be enough to give a reasonable idea of what other S-100 users have been up to all these years.

Notice that these circuits can be attached directly to a simple parallel I/O port. Actually, almost any computer could use these circuits, not just an S-100 machine, and the interface (shown in Figure 1) is remarkably simple. If you have an Apple, you're on your own...

Caveat Hacker

Before we discuss the general interface circuits shown, the reader should be aware that outside of the relatively safe 8 and 16 volt world of the S-100 frame, there are potentially deadly voltages available in many electric devices, especially those that connect to the power line. *Editor's note: 35 volts is a kind of break point between voltages that will harm humans and voltages that won't.*

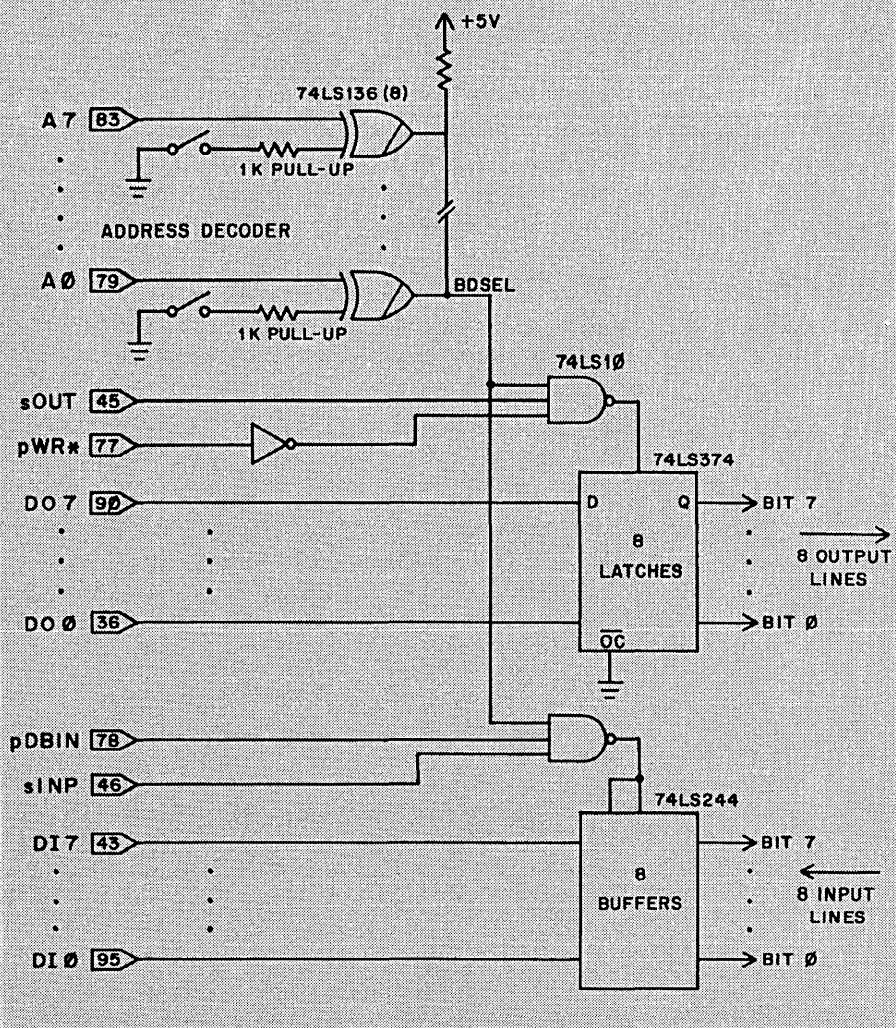
As a matter of fact, there are even

hazardous voltages available INSIDE most S-100 frames — in the power supply. Obvious safety precautions should be taken to prevent the reader (and computer) from being unable to read future issues of Micro C.

The Simple Interface

Figure 1, as mentioned above, shows a simple parallel I/O circuit. This interface takes up only a single parallel port, which can be any address you set in the switches.

Figure 1 - Common Parallel Interface



This circuit assumes that only 8-bit I/O is used in the S-100 machine. (This is the case in most S-100's, although the IEEE-696 specification for the S-100 bus allows 8 or 16-bit I/O addressing.) This simple parallel I/O port also uses no interrupts, no DMA or TMA, and no exotic parts.

If you don't like soldering irons or wire-wrap tools, there are many S-100 manufacturers who offer boards with simple parallel I/O ports. (Many S-100 parallel printer driver boards are actually just simple parallel I/O ports in disguise — or not so simple 8255 PIOs, which can also be used, if you know how to program them — or even less simple Z80 PIOs which aren't so hard to program but are very smart chips manufactured by Zilog and designed by the original team that moved over from Intel.) If you want to do it yourself, you'll need to add a 5-volt regulator to your home-made board to power these circuits.

Editor's note: Most multi-I/O and Herc graphics cards contain parallel ports — often called parallel printer ports. There are also two unused 8-bit parallel ports inside every 83 Kaypro.

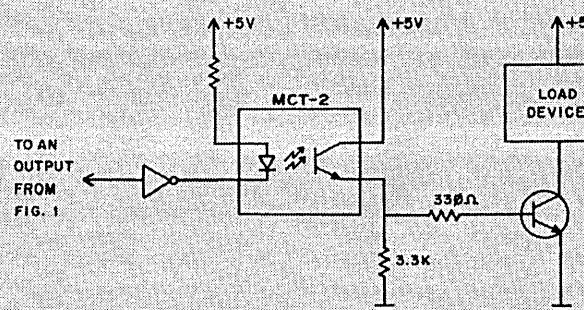
Seasoned "S-100 Bus" readers will instantly recognize Figure 1, because it has probably appeared here a dozen times in the past few years.

Controlled Output

Figure 2 shows the first step in connecting an S-100 machine to the hostile outside world. Using a simple opto-isolator (also known as an optocoupler), it is possible to use the parallel port in Figure 1 to control up to eight separate low-voltage devices. Depending on the opto-isolator used, higher voltage devices could also be controlled.

Optocoupler circuits like this are often used to connect one computer to

Figure 2 - Using Opto-Isolators



another without having any wires connected between them. Because there is no electrical connection between the two machines, each is safe from electrical malfunctions in the other. With opto-isolators, the only connection between the controller and the controlled is a beam of light. (For more info see Micro C issue #32, page 30.)

The circuit in Figure 2 works fine for machine to machine connections and for controlling low power DC devices. However, sometimes we need to switch high-voltage AC.

We could use a mechanical relay, but they're subject to failure. Plus, sparks across the relay's points cause problems for the equipment under control and make these relays dangerous in explosive atmospheres. (Like family life with teenagers.)

A solid-state relay is a better solution. Like an optocoupler, a solid-state relay isolates the control circuit from the controlled circuit by using a beam of light to turn on a phototransistor. Even better, a solid-state relay is designed to switch both DC and AC, and, depending on the unit, can switch up to several thousand watts.

Editor's note: Solid state relays are particularly noted for turning things on and off "softly." That is, there is a short time period when the power is coming on or going off. Solid state devices also draw very little power and they have no points to pit. However, they can be expensive, their voltage and current ratings are not unlimited, and they can be destroyed by the same power line spikes which take out computer supplies. In these situations, mechanical relays still make sense.

In Figure 3 we use a solid-state relay to switch 110 VAC (in this case a lamp or motor).

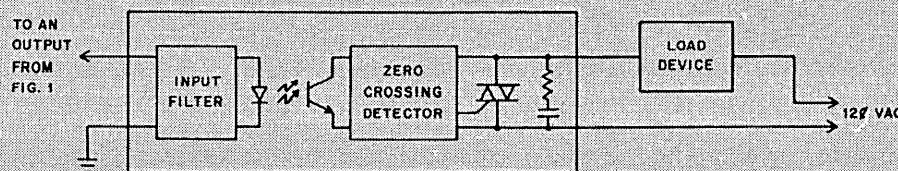
The computer could also control the speed of many motors by turning the power on and off very quickly and juggling the ratio of "on" time versus "off" time.

Controlled Input

The parallel port shown in Figure 1 also has eight INPUT lines, and there are just as many applications for inputs as there are for outputs. Obviously, simple switches could be connected to these lines to force a "1" or a "0"

(continued next page)

Figure 3 - Switching 110VAC With a Solid-State Relay

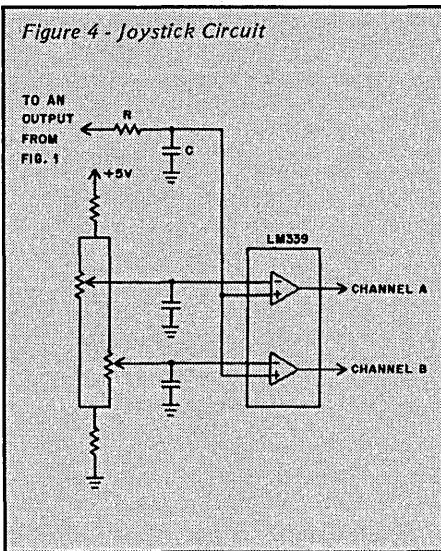


(continued from page 79)

onto each of them, as desired. Optocouplers could be used to provide additional isolation and allow another machine to control these lines.

An interesting joystick circuit submitted by a reader (it also appears in the book *Interfacing to S-100/IEEE-696 Microcomputers*) uses one output bit and two input bits to provide a simple two-channel position sensor, as shown in Figure 4.

The computer sends a "1" to the circuit's input to start capacitor C charging. Then it starts counting and waits for the output of the channel (either A or B) to go to a logical "1." By remembering the counter value



LETTERS

(continued from page 5)

MicroSphere reports that the serial drivers (MODE...) don't work in 3.2. I like to use serial printers.

Doesn't look like I've had a whole lot of luck putting large floppies on any of the office's clones.

Backwater Operating System?

I recently came across issue #29 of Micro Cornucopia (alas, too late to make plans for the SOG — maybe next year) and was pleased to discover the community of fellow computer enthusiasts you serve and represent.

I would like to address your comment that OS-9 is "a backwater operating system."

I would guess that there are well over half a million OS-9 based systems in current use, with a very high growth rate in the 16 and 32-bit versions — OS9-68K or OSK. Lack of software is hardly an issue because all standard CP/M software can run as a concurrent task on Z-80 coprocessor boards, under OS-9.

OS-9 is the second most popular operating system in Japan, behind MS-DOS. OSK has also been chosen as the standard for the next generation of European educational computing, running on hardware from such companies as Olivetti, Phillips, and Thompson. Phillips and Sony have chosen OSK (to be called CD-RTOS — Compact Disk Real Time Operating Sys-

tem) for the new Compact Disk/Interactive technology. OSK is now available for the IBM PC and clones, and the Apple IIe on coprocessor boards. It is available for the Atari STs and will soon be out for the Amiga and Macintosh.

The reasons behind the rapid adoption of OSK also make it extremely attractive to hobbyists. It is easily customized to specific hardware and software applications.

N. Bruce Nelson
2133 W. 1st Place
Mesa, AZ 85201

Editor's Note:

Thanks Bruce, I certainly didn't realize the kind of following OS-9K had. However, your letter does raise a few questions.

If the operating system is so popular, why is it so expensive? I receive their price list fairly regularly and it looks like I'd spend about \$1,000 for the operating system and two compilers. I'd get that and nearly 10 megs of utilities for \$500 by purchasing UNIX version 5.

Also, I thought MSX was the most popular operating system in Japan. At least that's what the MSX people told me.

How To Kill Your Machine

Here's a handy-dandy tip to remind my fellow hardware tinkerers of an important basic: any time you put two power supplies together on one machine, make sure you hook the supplies' grounds together. Not connect-

where each channel output changed to "1," the computer can determine the position of each of the joystick controls.

Since the counter value corresponds with the voltage at the variable tap of each potentiometer, the LM339 IC in this circuit is actually being used as a simple Analog to Digital Converter (ADC). If we replace one of the potentiometers in Figure 4 with a photocell, we've made a simple light sensor. Similarly, we could use a thermistor to make a temperature sensor, or a strain gauge to measure weight or stress. The interface possibilities are endless.



ing the grounds was a dynamite way to blow up a MicroSphere RAM disk and a Kaypro II at the same time.

Fortunately, the folks at MicroSphere helped me locate a bargain-priced replacement machine. But it would be nice to put a transplant in the dead machine. So what can I do with a Kaypro that has a fried board? The obvious answer is to buy a new Kaypro board from Erac, but are there other exciting alternatives? Little Board? Xerox 820? Can a PC-type board be used with the Kaypro keyboard?

Charles H. Kuttner, M.D.
213 Water Ave., NW, Suite 300
Albany, OR 97321

Editor's Note:

Get a new board from Erac. Take a look inside the Kaypro 16 to see what you'd be getting into going the PC route. The Little Board is a neat product but it needs a terminal, not just a monitor. The Xerox 820 needs a parallel keyboard and the ports don't come out in the right spots.

Interfacing A Synthesizer

My stepson recently acquired a Yamaha DX7 Synthesizer, and I am interested in interfacing it with either our MS-DOS clone or my older CP/M machine. I'd like to know where to get (or how to make?) the cheapest possible interface between the MIDI in/out on the keyboard and the parallel(?) port on the computers. Also, is there

anything in the public domain yet for editing the information once I get it to the computer?

I don't know much about music, just a little about the physics of sound, and not much more about computer hardware, but this MIDI business sounds like too much fun to leave to rich, computer-illiterate musicians. Any information you can give me or direct me to would be most appreciated.

Thatcher Deane
8626-25th NE
Seattle, WA 98115

Editor's Note:

Excellent question Thatcher. I too have a DX7 to which I'd like to add clone control. Any Walter Midis out there with information on using a clone as a controller/sequencer?

Turbo Pascal Bug On BBI

Recently I spoke with Larry Fogg about a problem I'd discovered with my BBI. Since then I've torn most of my hair out, so I decided I'd better write it down and hope that some Micro C reader can figure out what's going on.

When compiling Turbo Pascal programs to disk (both .COM and .CHN), Turbo writes all hex FFs to the directory entry where it should be putting the file info. Everything else seems to work perfectly, including programs run from within Turbo which write data to disk. I've tried making new

copies of Turbo from my master, compiling to the other drive, new RAM, and even a new floppy controller, but to no avail. C and BASIC programs compile just fine.

Does anyone have any ideas? I'd sure like to hear them.

Tom Casey
437 Margaret St. #75
Pittsburgh, NY 12901

except 0B, 0A, 08, and 0C are substituted for 05, 18, 13, and 04.

In WordStar 3.3 for my machine, the routines are duplicated at 4600, where the initial routine is accessed at beginning; INISUB is not needed, but UNISUB is necessary for exit because the duplicate routines are overwritten during the run.

The keypad table immediately follows the cursor table, so these routines are easily extended to modify it also. If the entire keypad is to be changed, the Kaypro routine will be shorter, but for the cursor keys only, my version is shorter.

Jack H. Wyatt
578 Garden St.
Sacramento, CA 95815



Patching WordStar And NewWord

Mark J. Boyd's article (issue #32, page 52) tells how he modified an editor on a Kaypro to change the cursor and keypad responses. This can be done much more easily in WordStar or NewWord, and probably in others as well, by installing the desired routines in MORPAT and calling them from INISUB and UNISUB. There are at least two simple routines for this, one of which is used by Kaypro. (See Figure 1.)

EA35 is the beginning of the cursor table in my K10 (first line of my routine in Figure 1). To find it in another machine, use this MBASIC program:

```
10 WBOOT = PEEK(1) + 256 *  
PEEK(2)  
20 CURSORADDR$ = HEX$(WBOOT  
+ 50)  
30 PRINT CURSORADDR$
```

The reset routines are the same,

Figure 1 - Initialization Routines For WordStar And NewWord

Initialization routines:

Kaypro's version:

		hex
LD	HL,(01)	2A0100
LD	DE,32H	112000
ADD	HL,DE	19
EX	DE,HL	EB
LD	HL,table	21#555
LD	BC,4	010400
LDIR		E8B0
RET		C9
table DB	05,18,13,04	

My version:

		hex
LD	HL,0EA35H	2135EA
LD	(HL),05H	3605
INC	HL	23
LD	(HL),18H	3618
INC	HL	23
LD	(HL),13H	3613
INC	HL	23
LD	(HL),04H	3604
RET		C9

AUTOMATE YOUR PROJECT

AT A COST YOU CAN AFFORD
with the SS-00A single board computer from IncRo

•TIL31

A ROM resident language for 8031 processor.
Faster than BASIC, easy to learn and extend.

•I/O

DATA bus, most address lines, RD & WR,
3 interrupt lines, eight 'bit' lines, etc.
RS232, shielded cable included.

•DEBUG

Interrupt driven, assembly debugger.
Directly down load Hex code.

•DOCUMENTATION

HARDWARE and SOFTWARE examples.
TUTORIAL.

•POWER

Any DC supply from 8 through 20 Volts.



INCREMENTAL ROBOTICS

BOX 560 SHERWOOD OR. 97140

PHONE (503) 625-7674

The SS-00A is subject to FCC Rules
and shall comply with them prior to delivery.

Building A Faster Clone

By Stephen Leon
200 Winston Drive
Cliffside Park, NJ 07010

Steve deals with his addiction to speed (and wins). I'm waiting with bated breath for the revelation: "Who sold him the speed" in the next episode of Public Domain.

Steve also gives us a peek inside his little black book with his list of the names and numbers of the cutest little Amiga boards.

I've never been one who thought the slogan "speed kills" meant anything more than a warning against using drugs. But it's true! Once you run a faster computer you'll never go back to a slow one.

In the last issue I mentioned building a Turbo clone. We finally got it running and it made the old IBM PC look like a Model T.

We've also been getting enough calls on the BBS to justify adding a second line. Quarterdeck's DESQview will handle a two-line BBS with ease. But a 4.77MHz PC is just too slow to support two lines.

The logical solution was to switch the Turbo to the board. Then we went back to using the PC and nearly died of boredom while it chugged along on 4.77 cylinders. So we ordered a Turbo AT clone.

Charlie Strom, the PC SYSOP on GENIE, had bought two of them from a company in California. Since Charlie is good on hardware, we went along

with his recommendation. For \$1,500 (the price we once paid for a single CompuPro memory board) we got an 80286 6-8MHz, no wait state machine with one meg of 120 ns RAM, a 360K floppy, a 1.2M floppy, a disk controller, two serial ports, one parallel port, keyboard, a Phoenix BIOS, etc. More on the AT clone next time.

PD Software As A Holiday Gift

This year has gone by quickly, and the holidays are getting close. Last year the bulletin boards were filled with greeting card programs, and the same will probably happen this year. Since the recipient of a computer greeting card has to have a computer, may we suggest that a useful gift would be a couple disks of public domain software?

SIG/M is still distributing the two sample Commodore 128 CP/M disks, and if you send me two disks and 73 cents in stamps, I will send them back to you. (Except for that one set that I put in the trunk of my car and somehow lost!) You can also order directly from SIG/M volumes 998 and 999. These are sampler volumes; 999 contains CP/M 80 software and 998 has CP/M 86 programs.

Amiga Bulletin Boards

The new ACGNJ BBS, at (201)886-8041, has lots of Amiga public domain software as well as an Amiga conference. We can't ship that software to

you because we can't write an Amiga disk — but if you have an Amiga you may want to download some of the material and give it to fellow Amiga users. Speaking of the Amiga, we've been getting Amiga material from AMuse, the New York Amiga user's group. They have a BBS at (212)269-4879. Other good Amiga boards around the country (according to Rudolf Musselman of 'The Duck Pond Board') are shown in Figure 1.

Reliance Mailing List For CP/M

We previously mentioned that Bill Meacham had sent us a copy of his Reliance Mailing List. We didn't have to ponder long about whether or not to release this excellent mailing list because it contained the big "no-no" — a request for a contribution. Bill made the decision easy; he dropped us a note authorizing removal of the request. It is now available on SIG/M 282.

SIG/M 283 contains Turbo Pascal I/O routines and math routines in BASIC. Bill Meacham is the author of the Pascal I/O routines, and the math routines are from Mike Finerty in Tucson. He has used Nevada BASIC and S-BASIC, but his routines can be adapted to work under almost any other version.

On the subject of Turbo Pascal, thanks to Clarence Rudd the ACGNJ BBS has a good Turbo Pascal section for both CP/M and MS-DOS. We may yet release it in the SIG/M library, but until we do, you might want to check the board.

More ACGNJ Mailing List

Last issue we mentioned that SIG/M 284 and 285 were PC-DOS versions of the ACGNJ mailing list. We also noted that 285 included Harry Van Tassell's BROWSE for Clipper. This mailing list is one that we use on a daily basis in my office. Subsequent to the release of the volumes, someone suggested that

Figure 1 - Amiga Bulletin Boards

AmigaLine!	California	(714) 772-4097
AmigaBoard	California	(213) 478-9788
BBS-PC	California	(415) 961-7250
Amiga Board	Colorado	(303) 693-4735
Casa Mi Amiga	Florida	(904) 733-4515
Micro-Systems	Florida	(305) 737-1590
Cyber Systems	Hawaii	(808) 521-3306
The Wind Dragon	Tennessee	(901) 755-5330
The Duck Pond	Utah	(801) 277-3200
Home Grown		

we save the expense of a word processing program for the letters we send out using these programs. As a result, using the Clipper MEMOEDIT command we built a word processor into the program. You should be able to modify the program with ease, but if not, drop me a disk and I will send you a copy. Helpful hint — Clipper mishandles the SET MARGIN TO command. You should already know that your screen will be messed up if you do not set your margin to 0 after printing with margins.

To print letters, you have to type the memo field. You can work around it by using a single field database to preformat the letter with the proper margins. In any event, if you can't solve it yourself, drop me a disk and return postage.

One last word about Harry. He popped in yesterday with a new version of the Clipper library. This time he wrote BROWSE as a function. Thus, all you need do is include a command BROWSE ("filename") and you have a full blown browse.

I complained that his last version did not allow you to add records. The new version does just that. He also has a QUERY which resembles the dBASE ASSIST format. This is also due to go into the library. Unfortunately, the Clipper library contains proprietary material, so we can't distribute it in the SIG/M library. Hopefully, Nantucket should be able to add these features to their product. We look forward to the next update.

TurboDOS Utilities From Sigi

Sigi Kluger is both a contributor to this magazine (see his "Unblinking the PC Cursor" in issue 31) and a prolific contributor to the public domain. We released a volume of his CP/M utilities (SIG/M 226) with 56 files on it. He has now sent us two disks of TurboDOS utilities which are available as volumes 286 and 287.

For those of you who run TurboDOS, this is a major release. The first volume has 62 items in an .LBR file. The second has 40. (Items include both object and source files.)

Included on these disks are SWEEP for TurboDOS, a RAM disk emulator, a V20 and V30 8080 emulator, a command line editor, a remote access package, a function key package, and a wildcard expansion module.

New PC/BLUE Releases

PC/BLUE 232 contains the well-received PC-Outline version 1.06. Volume 233 is the MCBS General Ledger version 3.2. The PC-Magazine Benchmark Tests (release 3) are on volume 234. This disk also has some good EGA tests.

Volume 235 contains Finance Manager II from Hooper International. A good macro assembler is on 236, and a somewhat confused automobile accounts receivable program is on 237. Statistical process control for manufacturing is the subject of Volume 238. PC/BLUE 239 is for Mandelbrot fans.

The CalTech Utilities And The AT&T 6300

Volume 240 contains the CalTech Utilities. These include: resident cut and paste, printer commands for IBM graphics printers, pop-up tables, dual monitor utilities, graphics characters, a command to exit a program caught in a loop, a bad sector fix, a hex file editor, directory utilities, an ANSI.SYS replacement, an ECHO replacement, and more.

Volume 241 contains "Managing Money with Your IBM PC" by Amihai Glazer and "The Stock Trader" from DSF Associates. A disk file manager and a program start-up menu shell are also on the disk.

Volume 242 is required reading for AT&T 6300 owners. It contains technical notes on the 6300, clock/calendar utilities for that machine, and a pro-

gram to set 80 x 50 video.

QModem 2.2 is on 243 and a teacher's gradebook program is on 244. Dennis Baer has written a "Structured Programming Language," which is on 245. Kermit version 2.29 is on 246. Bill Meacham's Turbo Pascal I/O routines are on 247 as is a spelling checker, a programmable RPN calculator, and a version of Othello. The Micro Development Software BASIC utilities, an advanced LOTUS tutorial, and Symphony Command language are on 248.

Winding up the list is a two-volume update to RBBS (14.1c) on 249 and 250. This is not a significant update from the last release, but if you are running a BBS you certainly should update.

How To Order

All of the recent SIG/M and PC/BLUE releases are available on the ACGNJ BBS at (201)886-1658. No pre-registration is required to download or upload. SIG/M volumes are stored as SIGxx.LBR. PC/BLUE volumes are PCxxx.ARC. (Except for SIG/M 284 and 285 which are .ARC files as they are in PC-DOS.) Or you can order disks from SIG/M and PC/BLUE.

SIG/M Volumes are available from SIG/M, Box 97, Iselin, NJ 08830.

PC/BLUE Volumes are available from the New York Amateur Computer Club, Box 100, Church Street Station, New York, NY 10008.



TECHNICAL TIPS

Unblinking The Kaypro 16 Cursor

A reader (who asked to remain anonymous) has "unblinking" his Kaypro 16 cursor by pulling out pin 12 of U3 (the blink output).

The Kaypros shipped with the full-size multi-video card, have their blink on pin 9 of U32. Pulling out this pin stops the blink. Unfortunately, there is no exposed blink signal on the new half-card multi-video boards Kaypro is now shipping.

After this mod, the cursor can be set for blink or not blink by programming the 6845 video controller. See Micro C issue #31, page 22.

Restoring Adventure

To restore a game in the MS-DOS version of Adventure (found on Micro Cornucopia's disk MS-3), invoke the game by typing:

ADVENT -R

Adventure will then ask you for the name of the saved file.

To get into the debug mode of Adventure, invoke it by typing

ADVENT -D -D -D

Mesh Feigenbaum
Parker Hannifin Corp.
17325 Euclid Ave.
Cleveland, OH 44112

Trouble With Floppy Disk Controllers

Within a few days after upgrading four Kaypros to 5MHz I started having disk access problems with two of them. It turned out that the 1793 disk controllers had failed (on the two oldest machines — serial #'s around 30,000).

The 1793 that died on Kaypro #1 was marked WDC'79 FD1793B-02 8312 16 on top, and AE2111B-00 17612418308 MALAYSIA on the bottom. The 1793 that died on Kaypro #2 was marked WDC'79 FD1793PL-02 8327 16 on top, and CA2111PL-0002 14698618326 MA-

LAYSIA on the bottom. The FDC 9216 data separator chips (U88) seem to be fine. After replacing the 1793 chips, the two older Kaypros run very solidly.

Frank Gadek

RD #1, Box 221-I
East Mill Hill Road
East Greenville, PA 18041

Using BACKUP & RESTORE

BACKUP and RESTORE are the much-maligned data-saving and recovery programs that come with MS-DOS. Their well-publicized problems have led to a hoard of commercial replacements. However, I've had good luck with them, since I figured out the undocumented problem.

The trick to making BACKUP and RESTORE work properly is to make sure the directory structure is the same during restore as it was during backup.

In other words, if you use BACKUP to copy your whole winnie onto floppies, you'd better write down the names and relationships of all the directories. That way, if you ever have to restore the files to the original

winnie (or onto a new winnie) you can first recreate the exact directory structure.

If you are just backing-up one directory, then you only need that one directory (with the same path) when you restore.

The version of MS-DOS doesn't seem to be important (for instance, you can restore 2.11 files onto 3.1 without problem), but incompatible directories will cause recovery errors that are unrecoverable.

Don Thompson
MicroSphere
PO Box 1221
Bend, Oregon 97709

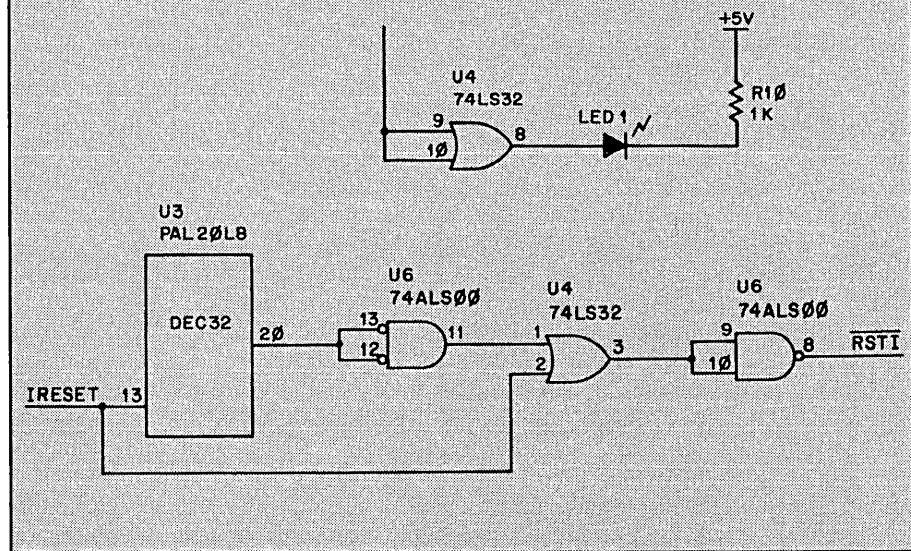
PD-32 Corrections

Figure 1 contains a pair of corrections for "PD-32: The Hardware." (See page 10 in issue #32.)

George Scolaro
22 Clotilde Street
Mount Lawley
Western Australia, 6050
Phone: 011619 2719342



Figure 1 - PD-32 Corrections



"Trademarks"

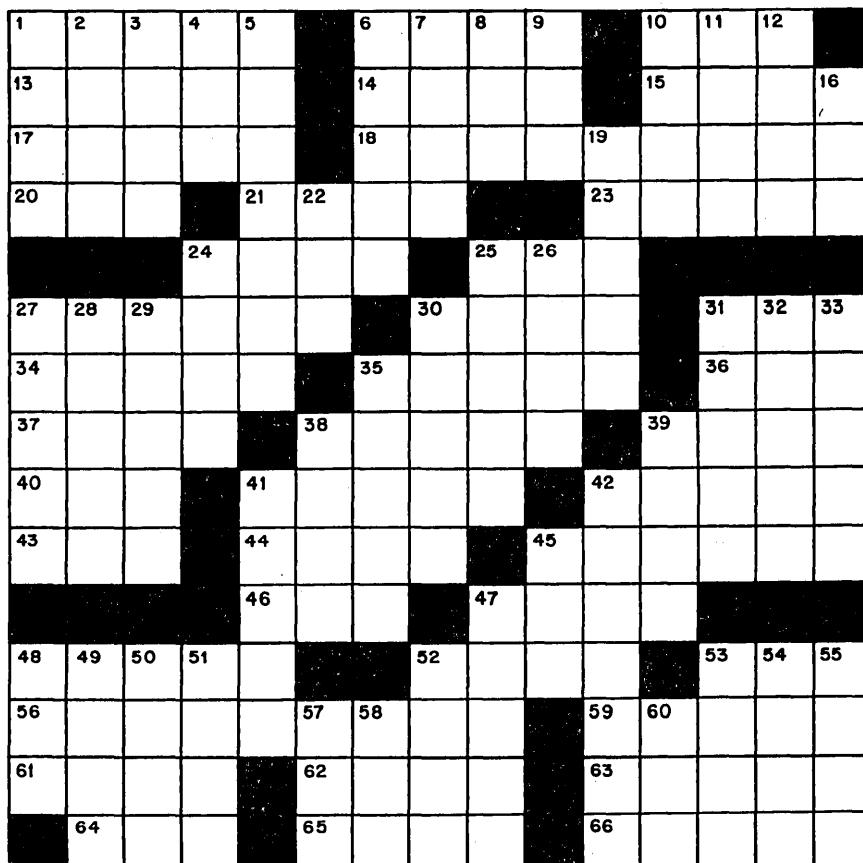
By Wayne Stoffel
 526 Wilson Road
 Annapolis, MD 21401

Across

- 1 Big or single
- 6 Computer instructions
- 10 Hgt.
- 13 Unplanned
- 14 Should be one for every close call
- 15 Modern music maker
- 17 It may not be questioned
- 18 (tm) bigamy?
- 20 Delete, to Kildall
- 21 The seven
- 23 Computer link to 24 Down
- 24 Welt
- 25 Cipher protocol (abbrev.)
- 27 (tm) star?
- 30 Stiff bristle
- 31 Chest part
- 34 South American animal
- 35 Computer
- 36 After after or before over
- 37 Globes
- 38 Keaton
- 39 Small group
- 40 MS format
- 41 OS moves
- 42 What one does with havoc
- 43 Direction
- 44 Complement
- 45 (tm) origin?
- 46 Console light
- 47 Lose yourself in the game
- 48 African capitol
- 52 Middle East capitol
- 53 Full deck in Rome
- 56 (tm) math wiz?
- 59 Desktop items
- 61 Language
- 62 Computer design
- 63 Language
- 64 K-O connector
- 65 Arquebus support
- 66 Copenhagen currency

Down

- 1 Asked
- 2 Aroma
- 3 Water plant
- 4 Periphery



- 5 (tm) cheapen also?
- 6 (tm) everybody welcome?
- 7 Work
- 8 Wipe out, for short
- 9 Follows cons and int
- 10 Shells, for short
- 11 Bring from memory
- 12 Carry
- 16 (tm) jewel?
- 19 Early PC
- 22 Attention
- 24 800
- 25 School VIPs
- 26 To be in Toulon
- 27 (tm) tree beginning?
- 28 Sophia
- 29 Look up there
- 30 The numbers, for short
- 31 Harder to find
- 32 Early computer
- 33 Chap
- 35 Let it out
- 38 Finishing word
- 39 If - then . . .
- 41 Coordinate
- 42 PC pioneer
- 45 Oil initials
- 47 CM twice
- 48 Cleo's problem
- 49 Select out
- 50 (tm) call at #10?
- 51 Control
- 52 && in C
- 53 Terrapin terrain
- 54 Informed about
- 55 Capri
- 57 File test
- 58 Tune
- 60 Unobstructed, for short

By Gary Entsminger

1912 Haussler Dr
Davis, Ca 95616

Super Books And PC-WRITE

Gary looks at a supercomputer book (OK, OK, the book isn't really that fast), the 68030, and a new version of PC-WRITE.

While researching my "Supercomputer" article (in this issue), I happened on Richard Jenkins' book *Supercomputers of Today And Tomorrow*, and couldn't resist the provocative Cray-2 on the cover.

Jenkins' book and the Cray-2 are recent developments (just been published & still being tested), and I was surprised to find a book about computers, especially high-speed computers, so up-to-date.

Its account of current and curious hardware — von Neumann (or serial) processing, parallel processing, gallium arsenide semiconductors, fine-grain architectures, array processors, reduction machines, optical computers, wafers, Josephson Junctions, and so on — is fascinating.

Jenkins sketches a history of high-speed computing in light, informative detail, throwing in tidbits like —

"ENIAC made its initial test run in November 1945, and the researchers immediately discovered two problems that have bedeviled computer designers ever since: heat dissipation and device failure. The heat thrown off by ENIAC's 17,468 tubes sent the temperature in the room soaring to 120

degrees and the tubes to premature deaths."

He assumes you don't know about the strange creatures he's describing, and he brings you up, without wearing out his welcome. For example —

"In their efforts to develop parallel computers, researchers have opened a technological Pandora's box. Dozens of different designs have flooded the research journals and just sorting them out is a job for a parallel supercomputer.

"For though there is essentially only one way to arrange the elements of a von Neumann computer, the possibilities inherent in parallelism are virtually unlimited. They hinge on three elements: the number of processors,



POOR MAN'S NETWORK

Now you can implement networking on your own CP/M computers and share resources whenever you want. Each user can access files and printers on the other computer directly, without expensive hardware or switches, and without a communications program. Share floppies, RAM-disk, hard disks, and printers between two users. Works with most standard CP/M programs, like Wordstar, PIP, dBase II, etc.

Poor Man's Network works best with bidirectional parallel ports, or standard serial ports at 9600 baud or greater. Present version supports two computers only, and requires CP/M 2.2 on each computer. Uses only 6K of memory. Does not require a hard disk; runs on 8080, 8085, Z-80, HD-64180.

Poor Man's Network comes ready to run on BigBoard I and II, Xerox 820, NorthStar Horizon, NorthStar Advantage, Televideo TPC I, Kaypro (not 2000), Apple II (Super Serial required), S-100 with Compupro Interfacer 4. Can be installed on other computers by altering one of the assembler overlay files provided. Each disk contains drivers for all the specified computers.

Best of all
is the price: only **\$69!**

Specify disk format: 8" SSD, NorthStar, Kaypro, or Apple CP/M. Phone orders accepted; sorry, no CODs or Purchase Orders. Payment may be by certified cheque, money order, Visa, or Mastercard. Personal cheques from outside Canada require up to 6 weeks to clear. Canadian orders please pay in Canadian dollars; others in US dollars. Price includes diskette, manual and postage. Ontario residents please add provincial sales tax.

Note: If you use another operating system, such as CP/M 3 or PC-DOS, send us your name and address, but no money, and we will notify you when it is available for your system.

ANDERSON TECHNO-PRODUCTS INC.
947 Richmond Road, Dept C
Ottawa, Ontario K2B 6R1, Canada

Telephone 613-722-0690 for more information or to order.

CP/M is a trademark of Digital Research; Z-80 is a trademark of Zilog; Wordstar is a trademark of MicroPro; dBase II is a trademark of Ashton-Tate. Poor Man's Network is a trademark of Doug Anderson Techno-Products Inc.

XEROX® 820-1 & 2 ITEMS

820-1 board, A&T.....	\$50.00
with 8" CP/M package.....	70.00
820-2 board, A&T.....	70.00
with 8" CP/M package.....	90.00
FLOPPY CONTROLLER card, A&T.....	95.00
if purchased with 820-2.....	65.00
CO-PROCESSOR board.....	65.00
8" CP/M package (820-1).....	35.00
8" CP/M package (820-2).....	35.00
5 1/4" dual disk drive CABLE.....	20.00

Xerox is a trademark of Xerox Corporation. CP/M is a trademark of Digital Research.

Capacitors
Diodes
ECL
Eproms
Fuses
LED'S
Linears
Memory Chips
Pots
Regulators
Resistors
Sips
Sockets
Switches
Transistors
7400
74LS
and more

Z80-B 6MHz
\$4.50
10 or more **\$3.00**

PHILIPS DRIVES
5 1/4" DSDD
96 TPI - 300 RPM
\$50.00 2/\$90.00

12" CRT MONITOR
w/power supply (no case)
\$20.00

5 1/4" DSDD DISKS .60

SCOTCH DC300A DATA CARTRIDGES
(USED).....\$5.00

LINE CORDS
10A 125V 1250W
\$3.00

E2I COMPUTER PRODUCTS

2273 American Ave. #8 - Hayward, CA 94545

(415) 786-9203

TERMS: Pre-payment, COD, Visa and Mastercard. California residents add sales tax. Orders are FOB Hayward, CA. Shipments by UPS ground unless otherwise requested. Prices and availability are subject to change without notice. Call or write for current product and price listing.

the size of each processor's memory, if any, and the way in which processors and memory are connected."

I liked this book, and you might, if you want to read a breezy yet intelligent introduction to high-speed computing.

\$14.95, in paper from TAB BOOKS

Newsbits

Motorola and Motorola Japan have announced the MC68030, a high performance version of the 68020. Motorola claims it "packs the power" of 300,000 transistors on a single semiconductor, will double the speed of the 68020, allow parallel data-bus input, and be available for sampling by next summer.

The 68030 will find its way into the Macintosh Plus (or Plus Plus, or whatever they'll call it) and the Motorola-Intel and Apple-IBM feuds will heat up all over again.

If I were a bettor, and I'm not, I still wouldn't bet on the winner.

Also in Japan, IBM (I don't know what IBM America said) has announced an integrated system based on its token ring network. (In short — traffic in rings generally flows in only one direction. So each ring station receives messages from one of its neighbors and passes it on through the "circle".)

The network's software ("Network/PC") will allow IBM and non-IBM hardware to run on the same system.

PC-WRITE

By now, almost everyone who computes on micros for a living has heard a little about the Shareware word processor — PC-WRITE, Bob Wallace, or Quicksoft.

This trio is, as far as I know, the best success story in the Share-soft-ware domain, and worth reme-ntioning.

Wallace began working for Microsoft when it was a young company of only 12 employees. From 1979 to 1983 he

wrote the compiler, front end, and much of the runtime for MS-Pascal. But when Microsoft grew to 300+ employees, he moved on and started Quicksoft.

PC-WRITE, Quicksoft's first product, is a slick, quick, and very complete word processing system which Wallace has released into the public domain. You use it, and if you like, you send Bob \$75, and he registers you and sends a manual and free updates.

It's a deal if I've ever heard one, and I hope folks continue to support it. For public domain PC-WRITE, unlike many other shareware products, isn't a demo or come-on; it's a fully functional program. Only the printed manual's been held back to protect Quicksoft's investment.

We've had PC-WRITE on a Micro C disk (#13) for almost a year now, but until recently I hadn't taken the time to use it, since I'm usually pleased enough with my standbys — VEDIT, EXPRESS, and WORDSTAR.

But this month I wanted to use Turbo Lightning's Thesaurus while I was editing, which meant VEDIT was out (it uses a graphic block cursor which LIGHTNING doesn't recognize), so I tried PC-WRITE.

I've opened and closed, edited, compiled, and printed about 200K of text files for this issue, and I'm tickled with the results. An excellent program worth the \$75, if you register and get the manual.

I used version 2.6, which includes excellent on-line help menus, multiple file and split-screen editing, extensive formatting, printing, file merging, searching, replacing, block and file moving, and loads of miscellanea like: word counting, WORDSTAR file conversion, and laser printer fonts. There are more, easier-to-get-at functions on PC-WRITE than on any other editor I've seen at anywhere near the price.

It's fast (faster than VEDIT and WORDSTAR, as fast as EXPRESS) and easy to learn (even without the man-

ual). If you're looking for a powerful word processor at a not-so staggering price, try PC-WRITE, from Quicksoft

—
219 First N. #224
Seattle, WA 98109
(206)282-0452

(If you want to sample first, without the manual, version 2.6 is available from Micro C for 8 bucks.)

And may all your indices, footnotes, endnotes, and margins be perfect ones and zeros.

And that's enough tidbits for this issue, I'm out of here.



OVERSTOCK & DEMOS COMPUTERS (including software)

KAYPRO 2	\$200
KAYPRO 2X & 4	300
KAYPRO 10	695
KAYPRO 16	1295

Printers	89.199
dBASE Software	39

(with tutorials)
PROJECT DATA SYSTEMS, INC.
1050 Northgate Dr. #200
San Rafael, CA 94903
(415) 492-1840

256K Upgrade For The Big Board II

By Andy Bakkers

de Gervelink 12
7591 DT Dendkamp
The Netherlands

I owe a lot to Andy Bakkers. He shows up at every SOG (the whole staff looks forward to his arrival), and he always brings goodies from Europe. This modification by Frank Baak, written up by Andy, is one of those goodies.

History repeats itself. We did it in the past — changed the 16K chips on the Big Board I to 64K chips. So there had to come a time when someone would take the Big Board II and replace the 64K chips with 256K chips. The person who did it was Frank Baak of the Netherlands.

The Circuitry

The schematic of the required selection logic is given in Figure 1. Because it's copied from Jim Ferguson's 256K RAM card, the software is the same as for the 256K RAM card. This software is implemented in the integrated BIOS. The RAM select decoder has been replaced by a multiplexer that switches simultaneously with the RAS/CAS multiplexer on the Big Board II.

Figure 2 shows the schematic diagram for the refresh counter extension. You should use this if your RAM requires 256 refresh cycles. So far I have seen only 256K RAM chips with 256 refresh cycles. This circuit adds one bit to the 7-bit refresh counter of the Z80 and requires two additional TTL chips.

Parts List

The following parts are required for this modification:

- 2 x 74LS30
- 1 x 74LS74
- 1 x 74LS139
- 2 x 74LS157
- 1 x 74LS670
- 4 x 16-pin socket 3 x 14-pin socket
- 8 x 41256
- wire wrap wire

The Operation

The total circuit consists of seven TTL chips plus eight RAM chips (41256). First you have to cut a number of traces. The traces are given below. I have indicated the pin closest to which you have to cut the trace.

U68 pin 13

U68 pin 10

U68 pin 14 (for the 256 refresh cycle change)

Pin 1 of every RAM chip has to be isolated.

If your RAM is soldered in, just cut the trace going to pin 1 and then replace the 64K chips with sockets. Connect all the pin 1s together underneath the board.

If your RAM is socketed, just remove the 64K parts and when you install the 256K RAM be sure to leave the pin 1s out of the sockets. Then connect all the pin 1s together.

Finally, connect the pin 1 line to pin 4 of D1.

The numbering of the ICs is as follows: the letters A,B,C,D are along the long side of the breadboard area, and the figures 1 and 2 along the short side.

Solder the wire wrap sockets of the ICs only with the power and ground pins to the board. Note that power and ground are available on the holes at the edge of the circuit board.

The power and ground pins are:

for A1,B1,C1,D1 : pin 8 grnd, pin 16 +5V

for A2,B2,C2 : pin 7 grnd, pin 14 +5V

Connect the power and ground with a good quality wire to the power and ground pins.

Next you can make the other connections. Note that the signals marked with an * (asterisk) are NOT on the pin side of the cut trace, and connect to the other side of the cut.

After the modification the circuit works exactly the same as the 256K RAM card, so the integrated BIOS software will allow you to have a 192K RAM disk, or you can implement CP/M 3.0 and use the RAM area for the buffers.

Frank Baak
Maasdal 35
2904 CN Capelle a/d IJssel
The Netherlands.

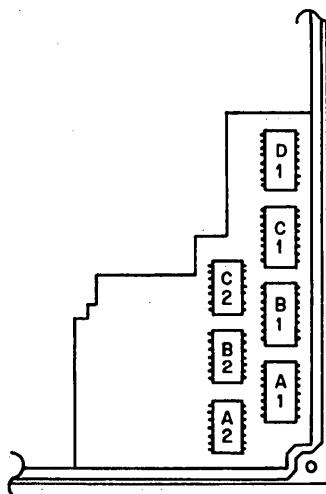
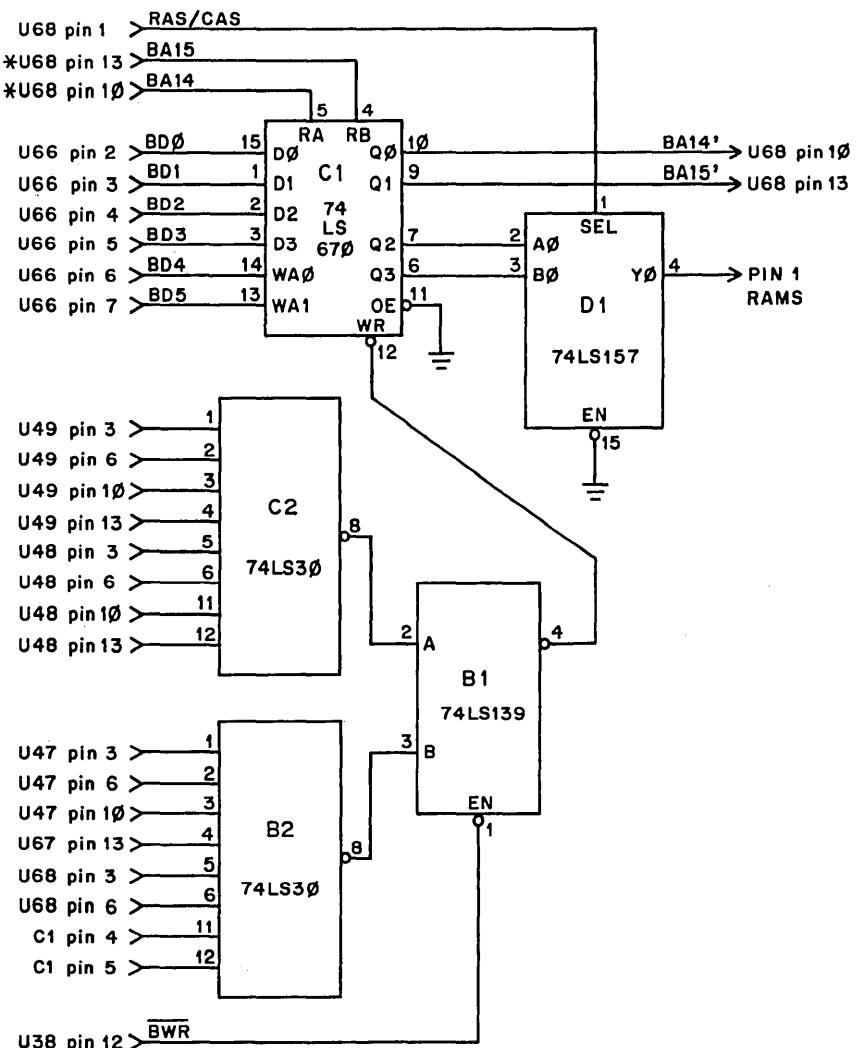
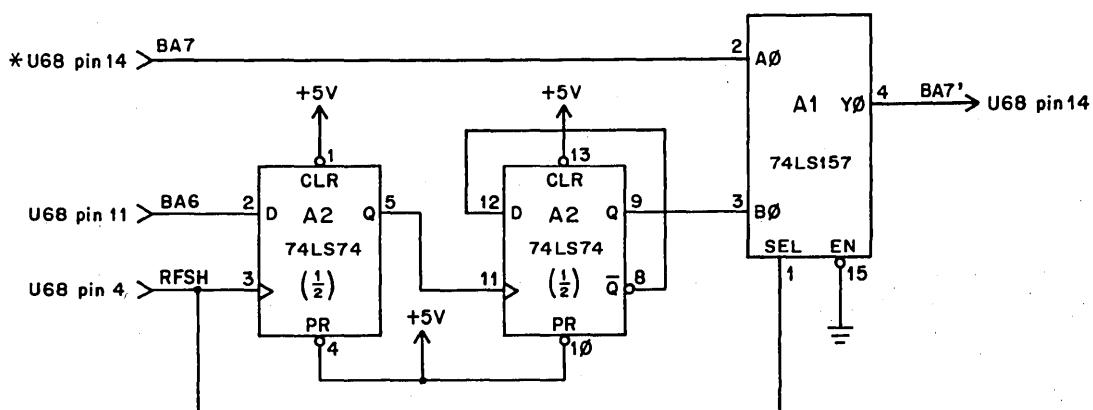


Figure 1 - 256K Select Logic



* INDICATES THE CONNECTION TO THE DISCONNECTED TRACE

Figure 2 - Refresh Circuit For 256 Refresh Cycles



* INDICATES THE CONNECTION TO THE DISCONNECTED TRACE

— MONITORS —

15" WELLS GARDENER.....\$20.00
—24 VDC
—3 LINE TTL INPUT
—GREEN PHOSPHOR
—OPEN FRAME

9" AUDIOTRONICS.....\$20.00
—3 LINE TTL INPUT
—OPEN FRAME

5" ZENITH.....\$23.60
—3 LINE TTL INPUT
—GREEN PHOSPHOR
—OPEN FRAME

5" MOTOROLA.....\$23.60
—3 LINE TTL INPUT
—BLACK & WHITE
—OPEN FRAME

SYNC-SEPARATOR KIT

USE WITH ANY 3 LINE TTL MONITOR
ALLOWS YOU TO RUN 3 LINE MONITOR
FROM A COMPOSITE VIDEO SOURCE!
\$12.95

14" LUXOR RGB.....\$299.95

—IBM COMPATIBLE
—RGB INPUTS
—16 COLORS
—RESOLUTION: 640 x 262

CABLE & POWER CORD INCLUDED!

LUXOR MONITOR + IBM COMPAT.
COLOR GRAPHICS CARD.....\$369.95

HAZELTINE 1500
TERMINAL
USED, TESTED GOOD
\$89⁹⁵

TELEVIDEO TS 816/40
\$995

TS 816/40 W/40M EXP.....1,500
TS 816/20.....\$600

HALTED SPECIALTIES offers unique supermarket style shopping for your electronic needs. We stock thousands of parts, from the newest IC's to some of the first transistors. We also have computer accessories, test equipment, tools, R&D supplies and much more. Please call or visit one of our retail stores.

— HARD DISK DRIVES —

3.5" MMI 10 MB.....\$229.95
—10 MB HALF HEIGHT
—ST 212/412 COMPATIBLE
—SHOCK MOUNTED IN 5 1/4" FRAME
—1 YEAR WARRANTY
—GREAT FOR PORTABLE!

SEAGATE ST 225.....\$349.95
—20 M BYTE
—1/2 HEIGHT
—1 YEAR WARRANTY

ST 225 + CONTROLLER.....\$459.95
—WITH CABLES
—READY TO INSTALL

MMI 10 MB + CONTROLLER.....\$329.95
—WITH CABLES
—READY TO INSTALL

HIGHER CAPACITY DRIVES AVAILABLE
CALL!!

— HARD DISK CONTROLLERS —

OMTI 5510 PC/XT CONTROLLER.....\$139.95
—SHORT SLOT, LOW POWER CARD
—COMES WITH CONFIGURABLE BIOS AND
SOFTWARE TO HANDLE ANY 5 1/4" HARD
DISK
—COMES WITH CABLES

WESTERN DIGITAL.....\$119.95
—FOR IBM PC/XT
—WITH CABLES

**WE STOCK A
FULL LINE
OF IBM
COMPATIBLE
PRODUCTS
PLEASE
CALL FOR
LATEST
PRICES**

SPECIAL
OF THE
MONTH!

DUAL FLOPPY CONTROLLER CARD FOR IBM

(Without cable &
mounting bracket)

\$19.95

— SHUGART 8" DISKETTES —

—SA150
—DOUBLE SIDED
—NEW
—BOX OF 10.....\$12.50

— FLOPPY DISK DRIVES —



5 1/4" TEC FB503.....\$99.00
—DOUBLE SIDED 48TPI
—IBM PC/XT COMPATIBLE
—QUIET!



SHUGART SA801R.....\$179.00
—SINGLE SIDED, DOUBLED DENSITY
—600 KB FORMATTED CAPACITY
—NEW, NOT RECONDITIONED!

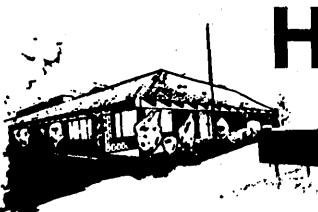
J FORMAT SOFTWARE PACKAGE.....\$43.95
ALLOWS USE OF 96TPI DRIVE ON
IBM PC/XT 720 KB CAPACITY

— POWER SUPPLIES —

VARIABLE 0-10 VOLTS DC.....\$375.00
—0-200 AMPS
—VOLTAGE AND CURRENT METERS
—SINGLE PHASE AC INPUT
—MADE BY SCR

EPROM PROGRAMMER
FOR IBM PC
\$109.00

—INTERNAL CARD WITH EXTERNAL ZIF POD
—PROGRAMS 2716 UP TO 27512
—SOFTWARE INCLUDED



HALTED SPECIALTIES CO., INC.

827 E. EVELYN AVE., SUNNYVALE, CA 94086

MAIL ORDERS CALL: (408) 732-1573

3 CONVENIENT LOCATIONS

HSC Electronic Supply of Santa Rosa
6819 S. Santa Rosa Ave.
Cotati, CA
(707) 792-2357

HSC Electronic Supply
5549 Hemlock Street
Sacramento, CA
(916) 338-2545

Store Hours:
Mon.-Fri. 8:00-7:00
Saturday 9:00-5:00

**WE SHIP
C.O.D.**

TERMS: Minimum order \$10.00. California residents add 7% sales tax. Prepaid orders sent freight C.O.D. or call for charges, shipping will be added to credit card and C.O.D. orders. Prepaid orders over \$100.00 use money order or certified check. Please do not send cash. Some items limited to stock on hand. Prices subject to change.



AROUND THE BEND

(continued from page 2)

I finally admitted defeat, closed up the 2000, wrestled it back into its bag, and slid it under the seat. The stewardesses went back to hawking their wares and the passengers once more nodded into their copies of *Mainliner*. All the passengers, that is, except the turkey who hovered just over my left shoulder.

He still wasn't sure that a computer wouldn't help him.

Sneaky

I don't know if any of you know the story of Post-It notes (those little yellow sticky-back pads that breed in offices). Well, the 3M team that designed the pad (and the adhesive) sent samples around to the secretaries in all the other 3M divisions. Within weeks the research group couldn't produce enough pads to keep up with internal orders.

Then the group issued an ultimatum: if 3M wouldn't fund production of the pads, they weren't going to supply any more. If they hadn't funded production there would have been mutiny in the ranks.

Anyway, when I heard the story I kind of chuckled. It's interesting to see someone make a good idea indispensable.

Meanwhile, that was happening to me.

I've always liked Borland products; they're good and they're cheap. They send fun things to the editorial department and we do our best to find time to review them.

In this case, the first copy of Reflex disappeared before us editorial types got our hands on it. Then an update arrived and disappeared. But I was too busy to worry very much about missing database packages (Turbo Prolog was getting prime time.)

I still wouldn't have noticed Reflex if Sandy hadn't mentioned that she really liked it. (And she is one of these database eclectics.) Then I started noticing that the office's computer cowards were spending significant portions of their days smiling at monitors. (Were they playing Adventure? Nope. Reflex.)

I made this discovery the day after an auto dealer called to ask how he could put together a mailing list. If he had waited a day I would have recommended Reflex. (They just have to have your answer right away, those car salesmen.) As it was, I recommended he contact a programmer.

Now, of course, I'm curious about Reflex. All those computer beginners are creating databases, reports, graphs... everything. In minutes. By themselves. And they love it.

But the Reflex manuals are almost impossible to nab for more than a few minutes at a time and if Borland wanted its Reflex back, there'd be mutiny in the ranks.

Things That Don't Work

Generally we do our best to make sure things work before we put them in the magazine. (There have been times when we've run science fiction as science fact, but

we've tried to make those the exception.)

In the April 87 issue we'd like to do the opposite. We'd like to include things that don't, didn't, or couldn't work: ideas, systems, software, companies, training packages, anything. Real failures and pseudo failures.

For Example

Remember the text editor that let you enter the first three characters of a word and then it filled in the rest of the word? (If it guessed wrong you could go back and correct it.) They called me one day to give me a chance to get on the bandwagon. They had incredible stats on how much faster a typist could go using their package (their typist, using their text). I wonder if they're still in business.

Remember all the mods you could do to an IBM Selectric typewriter to interface it with a computer? You could either build solenoids into the unit or mount solenoids over the keys. Either way, the typewriter would print slowly (and unevenly) for about two weeks and then die. (Several members of the Beaverton CP/M Users Group were better stocked with Selectric parts than the IBM service center.)

Remember Osborne's IBM clone? It was famous for being the only clone priced higher than an IBM.

Remember JRT Pascal v. 3.0? (Famous for its many release dates.) The Zorba? Remember when PC magazines were trying to predict whether MS-DOS or CP/M

(continued next page)

RP/M2™ creates

Z80®

CP/M®2.2 compatible

IBM PC

Now available for Decimation's Blue Thunder softcard, PC RP/M2 is an operating system. Either standalone or with DOS present, PC RP/M2 provides the solid base of a genuine operating system reliably distinct from the facade created by an MSDOS interface. All 2.2 system and BIOS calls are supported, with 56.5k TPA, file date and time stamping, fast virtual disk, iobyte redirection, terminal emulation, color console display, auto relog, COM path, addressable SAVE, single key phrase recall. SETDISK redefines a drive to any of over 80 CP/M formats. DOSDISK invokes built-in access to DOS drives. System disk with manual \$129. Blue Thunder™ softcard \$149. Shipping \$5 (\$10 non-US) 



118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

(continued from page 91)

86 would be the winner? How about the Grid Compass portable? (\$8150 each.)

If you have inside information on any of the above, or have been asked to attempt some off-the-wall things by some off-the-wall people, write to us, call us, or use the Micro C RBBS (503-382-7643, 24 hrs, 300-1200-2400). Leave a note, message, or file. Or, failing that (hey, we take failure seriously), drop me a note.

If you have an idea for a whole article, definitely let us know (somehow).

Don't think that this is going to be a frivolous issue. It's not. Some of the best people are still trying to come up with a replacement for the round wheel. (You have to learn to roll with it.)

PD-32 Support

You can contact Dan Efron on the PD-32 conference (Micro C RBBS 503-382-7643, 24 hrs, 300-1200-2400). See this issue for "PD-32: Messages From The Board." He will also provide support via his Arpa-net address: defron@violet.berkeley.edu.

Prolog

I used to worry that Gary was working too hard. Mornings would find him a bit on the haggard side, and by afternoon he'd be home, working alone. Out of those long afternoons and evenings came a couple of really good articles on Prolog.

He'd gotten into the language in little over a month (reading mostly the Turbo Prolog manual), and he had begun acting a bit like a guru. (It's a lot more fun being a guru than living with one.) Anyway, Gary made the mistake of leaving one of the Turbo Prolog manuals here when he headed south.

I picked it up.

Hey, it's a tutorial. Not a bad one either. Very readable, even in bed, the bathroom, and at the dinner table — leading to interesting conversations:

"What do you mean I ignore the kids?"

"What kids?"

You don't have to be anywhere near a computer to learn Prolog. (And you don't have to write expert systems to know you've been missing something at home.)

Now Borland has a new version (a few fixes and a lot of new features), and they are updating all current owners free. But Gary'll probably get the update. Shucks.

PD-68000

Joe Bartell called the other day to say that there's been a lot of interest in the 68000 system. He's been looking for a very cheap, very powerful, easy to build 68000 system on which to place his very cheap operating system. That's what we're missing for a full-blown 68000 issue.

It could be a stand-alone system or a plug in card. If you have any ideas, contact Joe at Hawthorne Technolo-

gy 503-254-2005. (Joe also has a technical bookstore in the other half of the building, so if you're looking for technical info...) Neat, eclectic (that means books and computers) guy.

Tech Help

We've been providing tech help on the phone ever since we started Micro C. It's been fun and it's been a hassle. It's fun when we can suggest a solution and the recipient calls back to say it worked. It's frustrating when there just doesn't seem to be an answer.

It's also frustrating to have to limit tech support to 9 a.m. to noon, but we have to do it. Even with the limited hours, Larry and I have to work evenings and weekends to get the articles written, hardware mods checked, software routines run, disks documented, catalogs written, and the RBBS properly brought up.

Speaking of the RBBS, we have a new one. It's a clone with 20 megs of winnie. If you see this issue's article on bringing up the board you'll know what Larry has been doing with his weekends. We have a tech help section on that board and we check it 5 days a week. If you need help, feel free to log on, (Join the tech help conference, and leave us a message.

When you call back in a day or two you'll get our answer, if we have one, plus you might even get someone else's answer. And if it's a particularly interesting problem, your question and the answers might even show up in Micro C.

Of course if you have a dynamite answer, by all means, log on. Who knows when someone will come up with a question for your answer.

Hard Times With Hard Drives

I'm working especially late tonight because I spent the daylight part of today working on the clone that holds the entire Micro C database.

For no apparent reason the machine picked today to refuse to boot off its internal winnie. Oh, the system's been grumbling for a while, but it's been nothing a reset or two couldn't handle as the machine was coming up.

Anyway, today it got right ornery. Half an hour worth of resets didn't make an impression. It even refused to boot off the floppy. Finally I removed the Western Digital controller board, disconnected the 20-meg Seagate's power connector, and tried booting from the floppy one more time.

It worked.

Then, just on a lark, I reinstalled the controller board and reconnected the 20-meg Seagate. Then, probably out of spite, the system booted off the Seagate when I hit the power.

So I double checked that I had everything backed up onto an external tape and winchester unit (an ancient but very dependable Tallgrass system) and then reformatted the internal Seagate.

I didn't just do an operating system reformat; this was a real hard core, low level, controller card format — which I followed by an FDISK partition and a FORMAT C:/s. I wanted to rewrite all those sector headers that

don't get rewritten any other way.

Boy did the system boot quickly after that (I think I got its attention)! I checked the winnie for bad sectors with Norton's DISKTEST. No errors. Then copied back all the data.

It turned out that John Jones (of Pascal fame) called while I was in the middle of the reformat. He mentioned that a number of his friends had been forced to do hardware reformats of their winnies after similar boot problems.

That made me feel better, but I'll feel especially better when the system boots properly tomorrow morning.

FTL Modula-2 Source

If you're wondering where you to get FTL Modula-2 (discussed in Issue #32's Kaypro Column), wonder no longer. It's published by:

Workman and Associates
1925 East Mountain St
Pasadena CA 91104
818-791-7979

The compiler is available for CP/M and MS-DOS and the price is \$49.95 for CP/M and I believe it's the same for MS-DOS (complete with editor). See John Jones' Pascal column in Issue #34 for a close look at this compiler.

REC & CONVERT

Harold McIntosh is not particularly well known in the U.S., certainly not as well known as Nicholas Wirth. But McIntosh and Wirth share a great deal. Both are professors of computer science at foreign universities and both have created their own languages.

There is a significant difference, however. We've all heard about Pascal, but very few of us are familiar with REC or CONVERT. Pascal was created as a teaching tool. So, it's very easy to learn. REC is short for Regular Expression Compiler. It was created as a tool for creating other tools, such as translators, compilers, and assemblers. It's not so easy to learn.

When Harold called last Spring and asked if I wanted to run an article on REC, he apologized with:

"REC hasn't really been very popular, I'm not sure why."

"But I've got an MS-DOS version of it now and I'd be tickled to have Micro C release it."

Well, I've always been fascinated by computer programs that made magic. In my book, assemblers are small magic, compilers are medium size magic. Compiler compilers are very big magic. However, it sometimes takes big magic to understand them. REC's problem is that Harold McIntosh understands it very very well.

Gary Entsminger spent many, many hours (including lots of 5 a.m. hours) editing the article, trying to make sense out of a paper written for graduate CS students. Then Gary wrote some sample programs for CONVERT.

When I talked to him last week he was ecstatic — in a sleepy sort of way.

"They've got to try them. They are just like Prolog, right on the front edge, they just need to be discovered. What a powerful pair of programs."

Despite Gary's work, and his enthusiasm, understanding CONVERT is still no trivial pursuit — but it's easier. And who knows, the task might well be worth it. After all, one of the big areas of AI research is pattern matching, and CONVERT was created to match and translate patterns, any patterns.

Move over Nicholas, here comes Harold (and Gary).



Editor & Publisher

The Ultimate In Software....



Order
Your Micro C
T-Shirt Today!

Why not collar one of our user-friendly, two-color shirts? This is top coverage that's available in: S, M, L, and XL.

Micro C T-Shirt \$6.95 ppd (US)
(specify size) \$8.50 (foreign)

WANT ADS

The following folks are reaching you for only 30 cents per word. If you would like to reach the same audience, send your words and 30 cents for each to Micro Cornucopia.

"Getting Started With CBASIC" - Tutorial workbook for beginners/intermediates in this powerful language. Fundamentals through file handling, user functions, etc. Gets you started and takes you up to advanced features and functions. Ends confusion, frustrations! \$25/U.S. Fourcorners Press, 222 Washington St., Hanover, MA 02339.

Affordable Engineering Software for CP/M-80, PC-DOS, MS-DOS, and TRS-DOS. Free 52-page catalog. Low cost pen plotter driver, transfer function analysis, scientific calculator, communications design spreadsheet, graph printing, circuit analysis, signal processing, mathematics, root locus, active filter design, thermal analysis, and report proofreader programs. BV Engineering, 2200 Business Way, Suite #207, Riverside, CA 92501. (714) 781-0252. VISA and MasterCard accepted.

DataCOM Super Systems (tm) is your one-stop headquarters for all the latest public domain and shareware programs. We support CP/M, CP/M+, CP/M86, MS-DOS, PC-DOS, and TurboDOS users with over 100MB of online files. DataCOM is a remote multi-user TurboDOS database with 2 incoming phone lines and Courier 2400 modems. We have an international reputation as the Tampa Bay Kaypro Users Group and have been online since July 1982. Call (813) 791-1454 by modem (300/1200/2400 baud) and download an application or send \$35 to: DataCOM Systems, 2643 Cedarview Ct., Clearwater FL 33519. DataCOM also has over 120 disks of CP/M and 150+ disks of IBM software available by mail for \$10 each. Catalogs available if you send a disk in a reusable mailer and include appropriate return postage. Specify Kaypro SSDD, DSDD, or IBM DSDD format.

Version Control System - TLIB (tm) stores ALL versions of your source code in ONE compact library file, even with hundreds of revisions. Updates (deltas) 5 - 7 times faster than UNIX SCCS. Date and comments with each version. Easy retrieval, LAN-shared libraries. Synchronized control of related source files. Free public domain MAKE (with Lattice C source) by Landon Dyer. MS/PC-DOS 2.x/3.x \$99.95 + \$3 s/h. VISA/MC. Burton Systems Software, POB 4156, Cary NC 27511-4156. (919) 469-3068.

640K MotherBoard Upgrade 640K memory without using an expansion slot! Allows easy insertion of 256K chips on the system board. 100% compatible. 30 day \$\$ back guarantee, 2 year limited warranty. COMPAQ Portable & Plus - \$29; \$99 w/memory. IBM XT, 3270 & Portable - \$39; \$109 w.m. IBM PC - \$50; \$129 w.m. \$4 s/h. Site discount. Dealers welcome. Product of Innovations, Inc. Aristo, 16811 El Camino #213-S, Houston TX 77058, (713) 480-6288, 800/3ARISTO.

Science Software for the Commodore 64 and C-128 (in C-64 mode). Includes hi-res color graphics and capability to save to most popular printers. Programs for amateur astronomy, earth satellites and aeronautics. \$19.95 - \$34.95. Demo disk and catalog available for \$5. Science Software, 7952 W. Quarto Dr., Littleton CO 80123.

For Sale: 2 SA604 5MB Winnies \$75 each, Western Digital 1000-05 Winchester controller \$85, Shugart 1610-3 Winchester controller \$75, JMR-1H5 IBM style external enclosure with power supply and cables \$125, Winif Z80 host adapter for 1610-3 (includes software) \$35. Any 2 items 10% off. Phil Mattison, 1701 E. Todd, Tempe, AZ 85283 (602)897-0293.

TURBO CHEQUES! and **TURBO BUDGET!**: TURBO Cheques! is the most powerful personal Checkbook management program available for the IBM-PC. TURBO Budget! provides Budget Management tracking capabilities. Both programs are professionally developed, easy to use, support color, hard disks and contain no copy-protection. 160+ page Users Manual. Written in fast Turbo Pascal. FREE Demo disk available. For the IBM-PC/XT/AT and true compatibles (Requires DOS 2.0 or higher, 2 DS/DD floppy disk drives, 128K RAM.) \$49.95 for both programs. CompuTech Box 7000-309 Redondo Beach, CA 90277. 213-377-7198

Explore Digital Signal Processing at a reasonable cost. Our Model 10 board for the IBM PC/etc. has the speed needed for a variety of real time applications. For example, a 1024 point FFT takes 70 milliseconds. The Model 10 is based on the Texas Instruments TMS32010 DSP chip. Included are a debugger, FFT and screen display software, and more. A/D and D/A option also available. Prices start at \$650. For more information or a brochure, write/call Dalanco Spry, Suite 241, 2900 Connecticut Ave NW, Washington, DC 20008 (202) 232-7999

CP/M-80 Emulates PC-DOS! Innovative utilities let CP/M reassign drives, autofind files (even overlays!), transform user areas into "disk drives," create smart batch-jobs, "window" the BDOS, format text columns, and much more. Inexpensive, copy-enabled, 30-day trial. LOGIC ASSOCIATES, 1433 Thome, Chicago, IL 60660, (312) 274-0531. Ask for free catalog, reviews.

■ ■ ■

**NEW LOWER 16 BIT PRICE
SPEED — POWER — VERSATILITY**

EXPRESS 2.2 FULL SCREEN EDITOR

ONLY \$29.95

Most EXPRESS users are converts from the GIANTS of WORD PROCESSING
.....and you couldn't pay them to go back...HERE'S WHY

- FULL ACCESS TO CP/M USERS AREAS (up to 32) with any editor command!
- NAMED DIRECTORIES supported on MSDOS VERSION!
- MEMORY MAPPED VIDEO AVAILABLE FOR IBM PC and PC clones!
- BUILT-IN CP/M LIKE COMMANDS (RENAME, COPY, ERASE, TYPE, DIR, LOG)!
- POWERFUL KEYBOARD MACRO'S (as many as you want)!
- FULLY RECONFIGURABLE COMMAND KEYS (emulate any other editor if you like)!
- FAST, FAST, FAST SEARCH!
- TERMINAL DATA BASE...INSTANTLY configure for over 50 predefined terminals!
- FILES LARGER THAN MEMORY handled with ease!
- CONTROL and HIGH BIT CHARACTERS may be entered and edited!
- DYNAMIC WORD WRAP/UNWRAP • FULL CURSOR CONTROL (and then some)
- EASY to SET TAB STOPS • GLOBAL/SELECTIVE/LITERAL/IGNORE CASE REPLACE
- FULL BLOCK INSTRUCTIONS including PRINT, SAVE, INCLUDE, MOVE, COPY, DELETE
- VARIABLE SPEED (FAST!) BI-DIRECTIONAL AUTO SCROLL • GOTO PAGE N / LINE N
- COMPACT on disk and in RAM. (even the 16-bit versions are only 25k bytes)!
- NOT COPY PROTECTED!
- AFFORDABLE...High performance at a fair price is our motto
- EXPRESS 1.0 SAMPLER available FREE on your Local Bulletin Board or for \$10.00 from TCI

GENTLEMEN ... I'm ready to step up to EXPRESS... please send:

____ copies EXPRESS 2.2 at \$29.95 + \$5.00 shipping and handling
____ copies EXPRESS 1.0 at \$10.00 + \$2.00 shipping and handling

My computer uses: PCDOS — MSDOS — CPM-86 — CCPM-86 — CPM-80 (Circle One)

Disk format: Kaypro II — Kaypro 2X,4,10 — Osborne 1 SSDD — EPSON QX-10

(Circle One) Morrow MD2 — Morrow MD3 — MS-PCDOS STD 5" DSDD — XEROX 5" DSDD
IBM CPM-86 5" SSDD — Zenith Z90 — 8" SSDD — Apple 35 TRK CPM
Other 5" Computer _____ Sides _____ Den _____

Send Check or Money Order to: Name _____

TCI

17733 205th Ave. NE

Woodinville, WA 98072

Washington residents must

add 8.1% Sales Tax

Address _____

City _____ State _____ Zip _____

Phone Day _____ Night _____

Mastercard/Visa _____ Exp _____

MASTER
CARD

PHONE ORDERS 1-206-644-3133

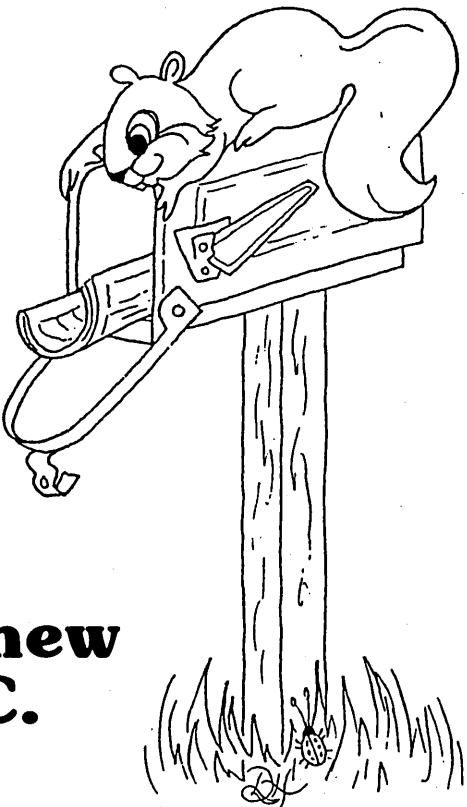
VISA

TOMORROW'S

COMPUTING

INNOVATIONS

**Peek
inside
your
mailbox
and
see...**



**...what's new
at Micro C.**

Sample of disks in the Fall catalog

MS-DOS Disks-5½" or 3½"

MS-24 Superior Games
Core Wars, SPACE WAR, Dungeons & Dragons, and (TRONLC includes Source.)

MS-25 Ultra Utilities
U-ZAP, U-FORMAT, and U-FILE; Map diskettes, unerase files, format non-standard disks, interrogate sectors & much more!

MS-26 Expert System Generator & Tutorial
Introductory expert system builder (ESIE), example expert systems, and detailed tutorial.

MS-27 System Primer
Device drivers and utility tutorials with source and DIAGS-the super system snooper.

All Disks \$8.00 each

**Micro
Cornucopia**



9-5 Weekdays
Order 503-382-5060



P.O. Box 223
Bend, OR 97709

Advertisers Index

Issue 33

A-OK Computers	30
Advanced Concepts Engineering	45
Anderson Techno-Products, Inc.	86
Andratech	77
Austin Codeworks	41

Bakkers, Andy	31
Orland International	Back Cover

CalSOFT	5
Cascade Electronics	26
C.C. Software	65
Chrysikos, Larry	10
Computer Helper Industries	54,55
Computing Technology	59
CompuView	27

Data Desk	Inside Front Cover
Digital Research	47

Echelon, Inc.	77
EcoSoft	51
Emerald Microware	33,41
ERAC Co.	21
E2I Computer	86

Fletcher, Don	22
---------------------	----

Halted Specialties	90
Hawthorne Technology	68

Incremental Robotics	81
Integrand	46

Laboratory Microsystems	32
Logical Development	41
Logitech, Inc.	1

McTek	15
Manx Software	77
Microcomputer Systems Consultants	37
Micro Cornucopia	53,73,95
Micro Logic Systems	26
Micro Methods	91
Micromint	20
Microprocessors Unlimited	32
MicroSphere	49
Micro Supply	42,43
Micro Technical Designs	63
Mitek	58

PC Tech	Inside Back Cover
Project Data Systems	87

Qua Tech	31
----------------	----

SBC Mart	59
Second City Software	69
Slicer	35
SLR Systems	5
Soft Advances	32

Tomorrow's Computing	94
----------------------------	----

Viking Software	68
-----------------------	----

Western Wares	46
Wondral	76

THE LAST PAGE

By Gary Entsminger

1912 Haussler Dr.
Davis, CA 95616

I've had an unusually fun time approaching this issue's deadline, thanks in part to the weather (gorgeous in northern California), and in part to the recent arrival of an improved version of Turbo PROLOG.

Besides being faster than its predecessors (v. 0.5 and v. 1.0 are the ones I've used), 1.1 has many new features — a built-in linker (hooray!), an improved trace mode, a size-of-window toggle, and a compile project option — to mention a few.

The new linker and compile project option make it very easy to build modular programs, particularly if you're using a hard disk. But you can build large programs even on floppies, once you get the hang of it.

For an example of modular programming, step with me through an innovative variation of "hello world" that lets us edit an overused message.

Developing A Project

In order to create a project, we first need a project file. Call it anything you want, but make sure it has a ".prj" extension. Let's call this one, "hello.prj".

"hello.prj" will contain the module list for the project, and just to keep it interesting, let's say our project has 3 modules — main.pro, sub1.pro, and sub2.pro. The contents of hello.prj will then look like this —

main + sub1 + sub2

Borland suggests you make all global declarations in a separate file. So let's call it — "globdef.pro". It looks like this —

```
global domains
  name = string
global predicates
  welcome(name) - (i)
  variation(name) - (i)
/* (i) specifies the flow pattern
(i.e. ""name" will be input).
A flow pattern is required
by all global predicates */
```

Our main module will clear a window, ask for a name, and call a predicate from a submodule. Like this

```
project "hello"
include "globdef.pro"
```

```
predicates
  start
goal
  start.
clauses
  start :- clearwindow,
    write("your name, please"),
    nl, nl, nl,
    read(Newname),
    welcome(Newname).
```

The predicate "welcome" is in another module (sub1.pro) which looks like this —

```
project "hello"
include "globdef.pro"
clauses
  welcome(Anyname) :-
    write("Hello ", Anyname),
    nl,
    write("This is the world"),
    variation(Anyname).
```

And the predicate, "variation", is in another module (sub2.pro) —

```
project "hello"
include "globdef"
clauses
  variation(Anyname) :-
    nl,
    write("you can edit hello world"),
    write(" if you like by pressing"),
    write(" any character"),
    edit("Hello, world", _).
```

Obviously, we don't need all these modules for our little program, but this extravagance will help you catch the drift of this kind of programming.

Compiling And Linking

The Turbo PROLOG system consists of a number of files — PROLOG.EXE, PROLOG.OVL, PROLOG.SYS, PROLOG.ERR, PROLOG.LIB, and INIT.OBJ. In order to load Turbo PROLOG, we need PROLOG.EXE and PROLOG.OVL on one disk (let's put it in A).

After we load it, PROLOG.EXE is in memory so we don't need it taking up diskspace (it takes about 217K, space we could use for programs and large databases), so let's figure out how to organize our floppies.

We do need PROLOG.OVL (and several other files) to generate an .EXE file from our project files, so let's create 3 disks — a loader with PROLOG.EXE, PROLOG.OVL, PROLOG.ERR, and PROLOG.SYS; a library disk with PROLOG.LIB, INIT.OBJ,

and HELLO.PRJ (our project file); and a source disk with PROLOG.OVL, PROLOG.ERR, and our modules (GLOBDEF.PRO, MAIN.PRO, SUB1.PRO, and SUB2.PRO).

Put the library disk in drive B, the loader in A, and from drive A enter — PROLOG

to get started.

Then replace the loader with the source disk, pull down "Options" from the menu, specify the project, and compile.

That's it (if we haven't made any errors).

Running

After the modules are compiled and linked, Turbo PROLOG asks if you want to execute the .EXE file. It's up to you.

The program (hello.exe) politely asks for your name, writes a version of "hello world", and then calls a full screen editor which allows you to mangle the message (something I've always wanted to do with "hello world"), and much more.

You can read, edit, and write auxiliary files, move and copy blocks between files, search and replace, and use many basic WORDSTAR-like cursor commands. And, an online help file for the editor is included. Not bad for one line of code!

Out Of Here

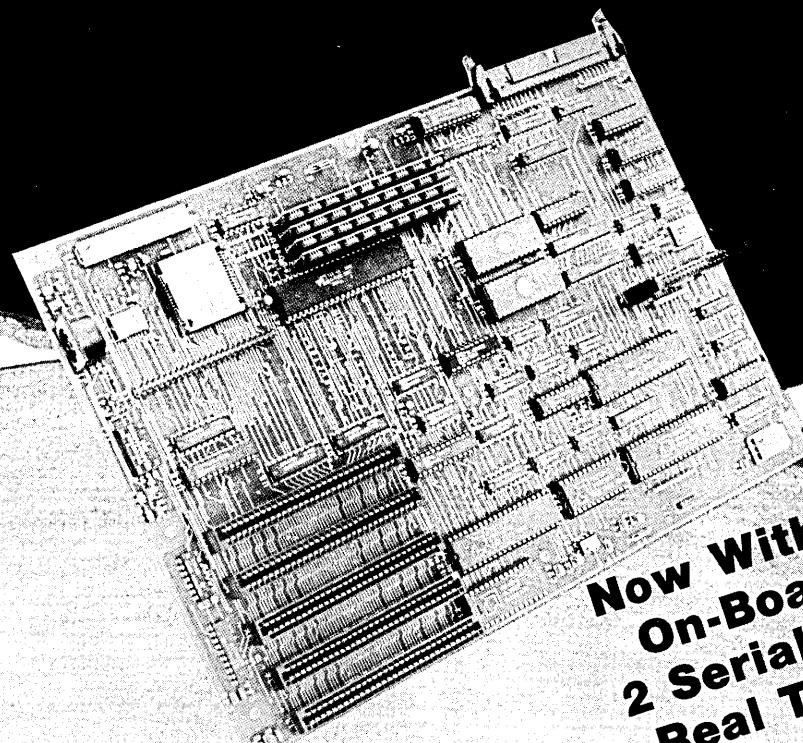
A minimal project (main, globdef, and 1 submodule without the editor) takes up about 35K. Adding the editor adds another 25K. Compile time for "hello.exe" (to 60K) on the X16 (an 8 MHz 80186) was 45 seconds, much of which was spent reading and writing to disk. Development on a hard disk is much faster.

So far, I'm impressed with Borland's style of PROLOG. It's friendly, fast, useful, and particularly warm to Turbo Pascal programmers.



**NEW
LOWER \$**

the X-16B gives you Out of This World Performance



*Now With Optibasic
On-Board 8087-2
2 Serial Ports, a
Real Time Clock.*

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, or 1M options.
- 8087 numeric coprocessor support on board.
- On-board floppy disk controller for up to four 5 1/4" drives.
- On-board SCSI hard disk controller port. Supports WD1002 SHD, WD1003 ACS, OMTI 5100, 5300 ADAPTEC ACB 4000.
- 2 On-Board Serial Ports.
- Battery Back-Up Real Time Clock.

NEW

NEW

Designed, Manufactured and Serviced by:

PCTECH

P.O. Box 128
904 North 6th Street
Lake City, MN 55041
(612) 345-4555

NEW LOWER PRICES

Kits - All Wave Soldered!!

- Full Kit w/o SCSI, serial ports, clock . . . \$ 550.00
- Assembled and Tested
 W/O SCSI, serial ports, clock . . . 1-4 \$ 599.00
 5-9 \$ 525.00
 SCSI Port add \$ 20.00
 Serial Ports each - add \$ 12.50
 Real Time Clock and Battery add \$ 12.50
- 1/2 MEG Add-On \$ 105.00
- XT Style, Enclosure \$ 75.00
- 150 Watt Power Supply, Tested . . . \$ 105.00
- 8087-2 Coprocessor \$ Call

NEW

Total Solutions:

Assembled and Tested Systems

- Mono-Chrome System \$1,499.00
 Includes: 512K, Power Supply,
 Enclosure, Keyboard, Monitor,
 Two Drives, and Printer Port,
 1 Serial Port, Real Time Clock.
- Quantity Discounts Upon Request
- Call For Info on Add-Ons, Hard Disks, Etc.

COD, VISA, MASTER CARD, CERTIFIED FUNDS

Why more than half a million people are using Turbo Pascal

Because Turbo Pascal is faster than any other Pascal compiler—it's become the industry standard. And because Turbo Pascal is backed by a complete range of "tool boxes" that give you most of the programming tools you'll ever need.

The Turbo Pascal family is never static, but is continuously expanding, with new products like Turbo Editor Toolbox and Turbo GameWorks.

"Language deal of the century... Turbo Pascal"

Jeff Duntemann, PC Magazine

Turbo Pascal has got to be the best value in languages on the market today

Jerry Pournelle, BYTE Magazine

This compiler, produced by Borland International, is one of the best programming tools presently available for the PC

**Michael Covington,
PC Tech Journal '95**

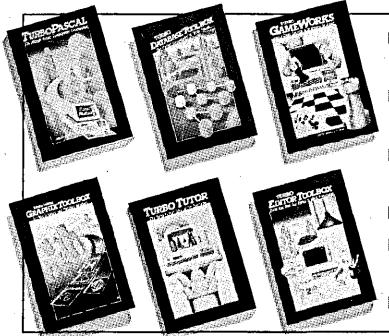
The secret of software success is not merely low price, but top quality allied with complete documentation, like our 400-page reference manual.

All of which are some of the reasons why Turbo Pascal is clearly the leader, and the recipient of awards like *PC Week's* "Product of the Year" and *PC Magazine's* "Award for Technical Excellence."

NEW! TNT* A DYNAMITE DEAL!

*Turbo 'N Tutor

Turbo Pascal and Turbo Tutor for only \$125.00. You save nearly \$15.00, and have everything you need to get up and go with Turbo Pascal!



A whole family of tools

Success breeds success, so the Turbo Pascal family has flourished. Your choices now include:

- Turbo Pascal 3.0 combines the fastest Pascal compiler with an integrated development environment, and 8087 math co-processor support and Binary Coded Decimals.

- Turbo Database Toolbox a perfect complement to Turbo Pascal. It includes a complete library of Pascal procedures that allows you to search and sort data and build powerful database applications.

- Turbo Graphix Toolbox includes a library of graphics routines for Turbo Pascal programs. Lets even beginning programmers create high-resolution graphics with an IBM®

- Turbo Pascal® 3.0 with 8087 and BCD
- Turbo Database Toolbox™
- Turbo Graphix Toolbox™
- Turbo Tutor® 2.0
- Turbo Editor Toolbox™
- Turbo GameWorks®

Hercules™ or compatible graphics adapter. Does complex business graphics, easy windowing, and stores screen images to memory.

- Turbo Tutor 2.0 teaches you step-by-step how to use Turbo Pascal, with commented source code for all program examples on diskette.

An offer you can't refuse

You can save \$119.75 when you choose the Turbo Jumbo Pack. Six different Turbo Pascal products for only \$299.95.

You get Turbo Pascal 3.0 and Turbo Editor Toolbox and Turbo Tutor 2.0 and Turbo Graphix Toolbox and Turbo GameWorks and Turbo Database Toolbox!

So act now—rush to your dealer or call us today!

NEW! Amazing value! Turbo Editor Toolbox includes MicroStar, a full-blown editor that also does windows! You get ready-to-compile source code and a 200-page manual that tells you how to integrate the editor procedures and functions into your programs.

Also includes MicroStar, a complete editor with full windowing capabilities. (You could pay \$100.00 or more for a program like MicroStar, but you get it free as part of our Turbo Editor Toolbox.) You can also use Turbo Editor (which of course integrates with Turbo Lightning™) to build your own word processor!

NEW! Turbo GameWorks gives you the games you can play, write, rewrite, bend and amend! Turbo GameWorks reveals the secrets and the strategies of game design. You're given source code, a 200-page manual, and the insight

needed to write and customize your own irresistible games.

Also includes ready-to-play Chess, Bridge, and Go-Moku—an ancient Japanese game that can divert you from reality for hours on end.

YES! I want the best!

To order by phone,
or for a dealer nearest you,

in CA call (800) 742-1133

Copies	Product	Price	Totals
—	Turbo Pascal 3.0 w/8087 & BCD**	\$ 99.95	\$ _____
—	Turbo Pascal for CP/M-80	69.95	\$ _____
—	Turbo Database Toolbox	69.95	\$ _____
—	Turbo Graphix Toolbox*	69.95	\$ _____
—	Turbo Tutor 2.0	39.95	\$ _____
—	Turbo Editor Toolbox*	69.95	\$ _____
—	Turbo GameWorks*	69.95	\$ _____
—	Turbo Pascal & Turbo Tutor	125.00	\$ _____
—	Turbo Jumbo Pack*	299.95	\$ _____
—	Outside USA add \$10 per copy CA and MA res. add sales tax	\$ _____	\$ _____
—	Amount enclosed	\$ _____	\$ _____

Prices include shipping to all US cities.

Carefully describe your computer system:

My is: 8-bit 16-bit

I use: PC-DOS MS-DOS CP/M-80 CP/M-86

My computer's name and model is:

The disk size I use is: 3½" 5¼" 8"

Payment: VISA MC Money Order Check

Credit card expiration date _____

Card #

Name: _____

Shipping Address: _____

City: _____

State: _____ Zip: _____

Telephone: _____

CODS and purchase orders WILL NOT be accepted by Borland.
Outside USA make payment by bank draft, payable in US dollars
drawn on a US bank.

NOT COPY PROTECTED 60-DAY MONEY-BACK GUARANTEE

If within 60 days of purchase this product does not perform in accordance with our claims, call our customer service department and we will gladly arrange a refund.

All prices are suggested list prices and are subject to change without notice.

Minimum System Requirements:

Turbo GameWorks, Turbo Graphix Toolbox, Turbo Tutor, & Turbo Editor Toolbox—192K. All other products, 128K.

*IBM PC, PCjr, AT, and true compatibles.

SU8

**16-bit only.

BORLAND
INTERNATIONAL

Vive la différence

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX 172373

Borland products include Turbo Prolog, Turbo Pascal, Turbo Pascal for the Mac, Turbo Tutor, Turbo Editor Toolbox, Turbo Database Toolbox, Turbo Graphix Toolbox, Turbo GameWorks, Turbo Lightning, Lightning Word Wizard, Reflex, The Analyst, Reflex for the Mac, Reflex Workshop, Sidekick, Sidekick for the Mac, Traveling Sidekick, and SuperKey—all of which are trademarks or registered trademarks of Borland International, Inc. or Borland/Analytica, Inc. Traveling Sidekick is not in any way associated with Traveling Software, Inc. of Seattle, Washington.

CP/M-80 is a registered trademark of Digital Research, Inc. MS-DOS is a registered trademark of Microsoft Corp. IBM is a registered trademark of International Business Machines Corp. Hercules is a trademark of Hercules Computer Technology. Copyright 1986 Borland International. BI-1082BW

TURBOPASCAL®

© Copyright 1983. Licensed Material. Program property of BORLAND International, Inc., 4585 Scotts Valley Drive, Scotts Valley, CA 95066. Unauthorized use, duplication or distribution is strictly prohibited by Federal Law.