

THE MICRO TECHNICAL JOURNAL

MICRO CORNUCOPIA

PC Graphics

This issue we're concentrating on graphics, from mathematically-generated fractals to human-generated acorns. Plus, we'll look at the newest hardware for displaying the finished product.

Drawing The Mandelbrot And Julia Sets page 6

It was so much fun creating Mandelbrot fractals that we couldn't resist using this one as our first full-color cover. Larry tells you how to do it.

Desktop Graphics page 10

Expanding the desktop world. Here's how an author and a graphic designer use desktop graphics.

Designing A PC Workstation Board Around The TMS34010 page 18

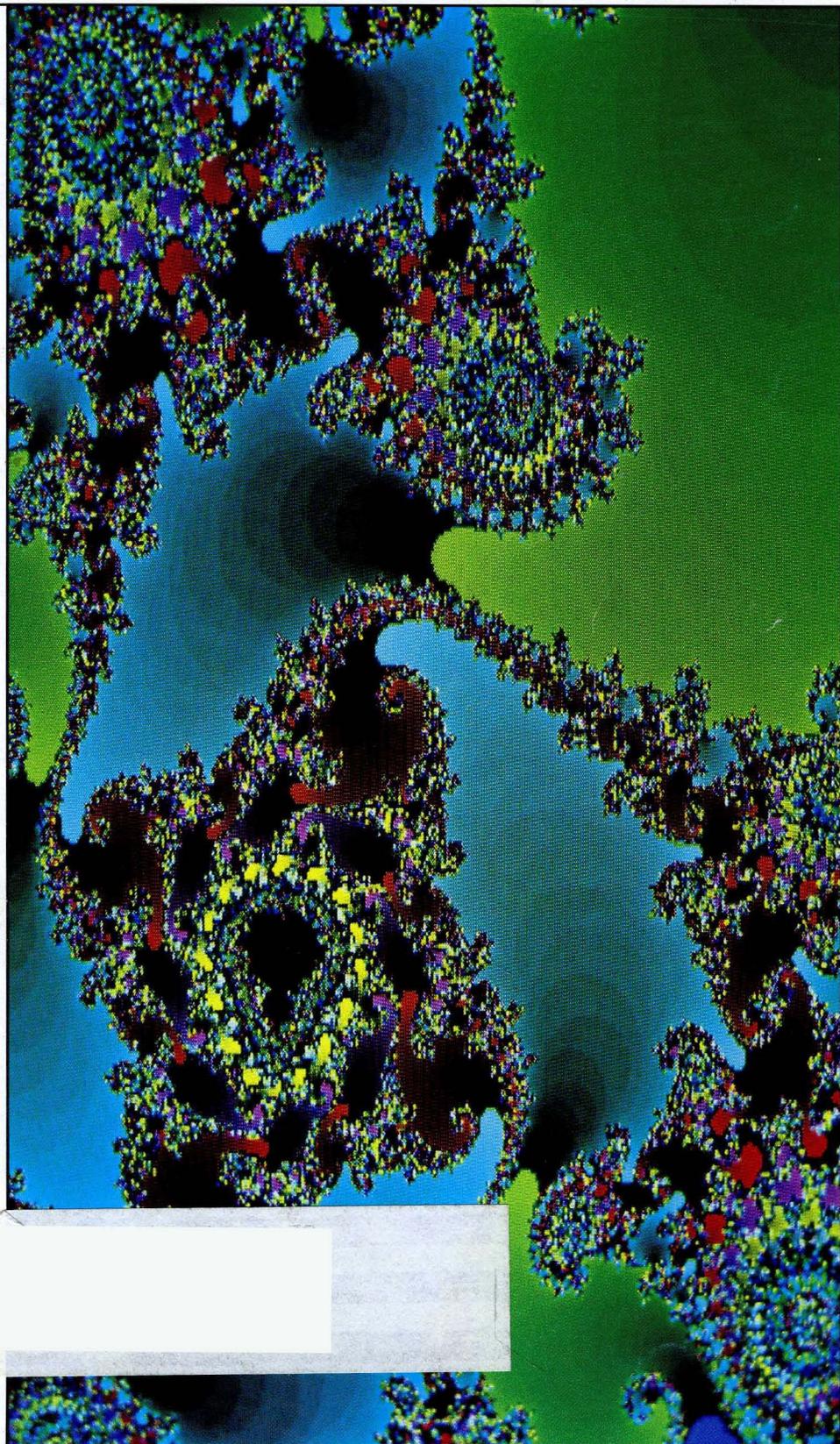
Dean Klein takes a close look at this super-powerful graphics processor.

A Hercules Primer page 24

The Hercules card is the only non-IBM standard. Here's how to write graphics routines for this interesting little card.

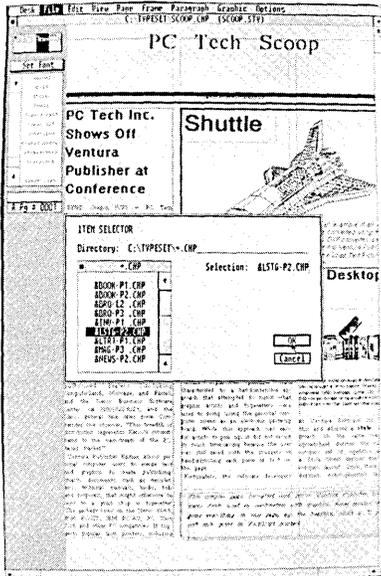
Controlling Synchronous Serial Chips With A Parallel Port page 30

Want some speed in your morning serial? Bruce Eckel clocks in with a primer on fast bits.

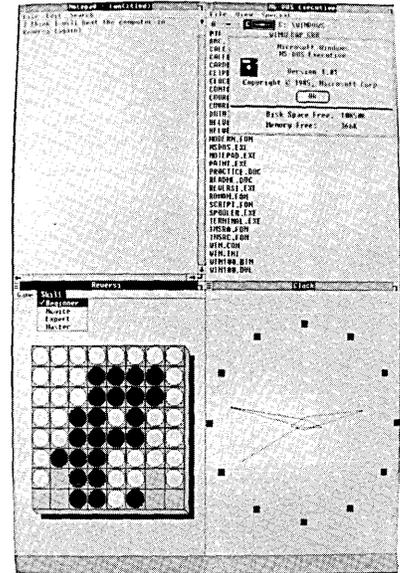


VERY HIGH RESOLUTION

The PC Tech Monochrome Video Board and Monitor



- Windowing, most desk top publishing packages, full page Lotus 123 and other applications.
- PC, XT, and AT Compatible.
- MDA (MONOCHROME) emulation.
- CGA (COLOR GRAPHICS) emulation.
- Full Screen Graphics.
- 6 MIPS TMS34010 32-bit graphics CPU. 736 by 1024 resolution. Both black on white and white on black text modes supported.
- Complete systems available.



***SPECIAL LIMITED TIME OFFER:** PC Tech is offering the complete video system (monitor, graphics adapter card, all necessary interface software and cables) for a special introductory price of \$995 plus shipping and handling. Bank card orders welcome.

SPECIAL FOR MICRO-C USERS: Ask for the Micro-C special which will include extensive development software at no charge!

Now also available: The PC Tech Color 34010 Video Board. This is the workstation class graphics engine featuring:

- 6 MIPS TMS34010 32-Bit graphics CPU.
- Up to 4 Megabytes non-display memory.
- Up to 1 Megabyte of display memory.
- Eight bits per pixel give 256 simultaneous colors.
- Color palette provides up 256K color choices.
- Hardware support for CGA/MDA emulation.
- 800 by 512 to 1024 by 800 resolution.

This board is designed to be the ultimate windowing engine with the PC Tech native mode MS-Windows driver. In addition to the MS-Windows support, CGA and MDA emulation software is provided as well as the PC Tech graphics development toolkit.

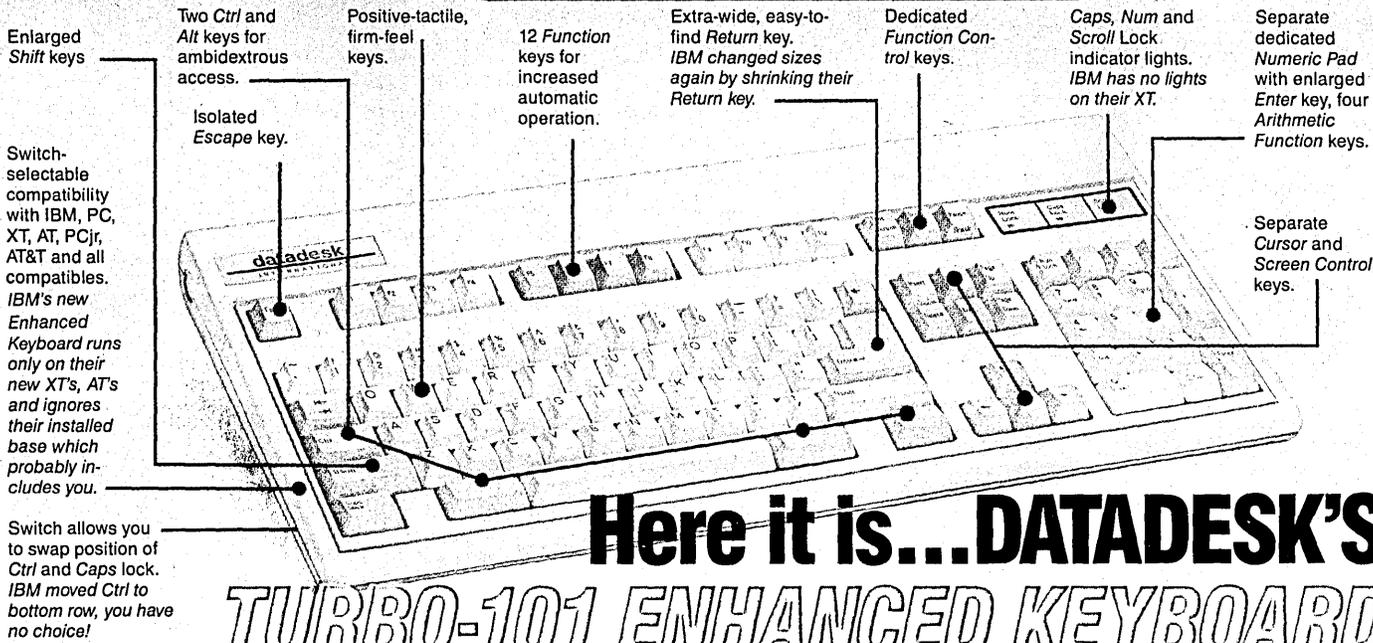
Introductory Special: \$995 (includes 1.5 megabytes memory, and software)

Designed, Sold and Serviced By:



904 N. 6th St.
 Lake City, MN 55041
 (612) 345-4555
 (612) 345-5514 (FAX)

Borland's Turbo Lightning™ FREE



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 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!

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

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 IBM AT are registered trademarks of International Business Machines, Inc.

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.



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

\$149.95* LIMITED OFFER

*Plus \$10 Shipping & Handling. California Residents Add \$9.75 Per Unit Sales Tax

NO. UNITS: _____

PAYMENT: VISA MC CHECK

AMOUNT ENCLOSED: \$ _____

Enter ←

NAME: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

PHONE: _____ COMPUTER TYPE: _____

CC NO.: _____ EXP: _____

Editor & Publisher
David J. Thompson

Associate Editors
Gary Entsminger
Cary Gatton

Technical Department
Larry Fogg

Director of Advertising
Laura Logan

Distribution Manager
Tracey Anthony

Accounting
Sandra Thompson

Order Department
Tammy Westfall

Graphic Design
Carol Steffy

MICRO CORNUCOPIA (ISSN 0747-587X) is published bi-monthly for \$18 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).....\$18.00
2 yr. (12 issues).....\$34.00
3 yr. (18 issues).....\$48.00
1 yr. (Canada & Mexico).....\$26.00
1 yr. (Other foreign).....\$36.00
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.

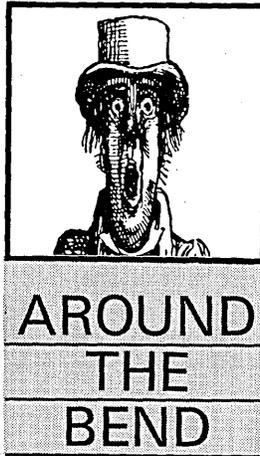
MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709

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
8 Bits, No Parity, 1 Stop Bit
503-382-7643

Copyright 1988 by Micro Cornucopia Inc.
All rights reserved
ISSN 0747-587X



By David Thompson

Computer Shows Revealed

Comdex is going to be bigger than ever. I know because I started calling around for motel reservations in August. It took four frustrating weeks to find a room, but it was worth the effort. Sandy and I will have a wonderful view of the Grand Canyon.

Of course there were still some slots in the by-the-hour rooms, but I'm not sure they'd let us stay there. We're married.

Secret New Products

The reason most people go to Comdex is to see (or announce) new products. I know there'll be the standard \$25 clones—the 800 meg, 5 ms, half-height hard drives—the one pound, 60 MHz, 80486 laptops—and so on. Of course all these have been rumored for years. It's gotten so folks are more interested in the products that aren't being shown.

"Hi, Zilog. Seen any good Z800s lately? You remember the Z800?"

"Hi, Microsoft. How about a demonstration of OS/2? You know, like the one you were supposed to give the developers last Spring."

I happen to know that the most interesting new products won't be announced at Comdex. These new "unrumored" products will be announced at the other "dex" shows. These shows aren't widely known, which makes them particularly fertile ground for hunters of unique technology.

You probably haven't heard about Rolladex (world's largest display of automatic bowlers), Index (data communications for the incarcerated), Spandex (work stations for rubber bridge designers), or Carddex (gambling software). And that's, of course, just a few.

Unrumored New Products

I thought you'd like a sampling of the new announcements, announcements which you won't find in any other magazine.

(1) *No Risk Poker*. This program is worth its weight in quarters if you're having friends over for a game. Program sets hands and takes bets. Selected player will lose steadily (amount is configurable) until the last half hour. Then he cleans out everyone. Great way to entertain old Navy buddies and relatives.

(2) *Insider*. Currently being tested by phone freaks,

(Continued on page 88)

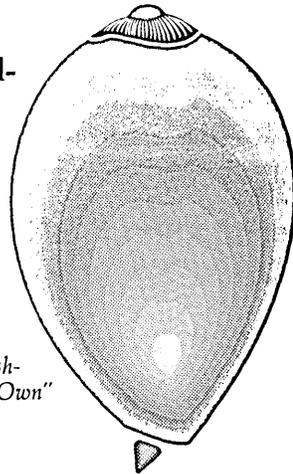
MICRO CORNUCOPIA

JAN./FEB. 1988 - ISSUE NO. 39

FEATURES

6 Larry Fogg
Drawing The Mandelbrot And Julia Sets

Math is really fun, especially if you're using it to explore the nether regions of Mandelbrot.



10 David Thompson
Desktop Graphics

Desktop graphics for desktop publishing? This is a deluxe "On Your Own" from those who are doing it.

18 Dean Klein
Designing A PC Workstation Board Around The TMS34010

The 34010 was the graphics processor of choice at Fall Comdex. Take a close look at this super chip with its general purpose instruction set.

24 Larry Fogg
A Hercules Primer

Nearly everyone has a Hercules (or Hercules compatible) graphics card. Here's how to write graphics routines.

30 Bruce Eckel
Controlling Synchronous Serial Chips With A Parallel Port

High speed communications. Bruce makes it easy.

48 Russ Eberhart
Taming The PC's Serial Port

Russ designs data transfer routines for his serial ports.

```
inc char ;next
cmp char,70h ;last
jnz txwait ;no,
lea dx,cr1f ;yes,
mov ah,9 ;prin
int 21h ;DOS
lea dx,oxgood ;load
mov ah,9 ;prin
int 21h ;DOS

mov ah,04ch ;term
int 21h ;DOS
```

54 Ramachandran Bharath
LISP Programming From The Bottom Up

Interested in true recursion? LISP is the prophet.

58 Sam Bogoch & Lain Bason
Parallon Message Passing

How to pass messages in a parallel system made up of standard processors.

COLUMNS

62 86 World

Laine runs afoul of the (local) law, then discusses the joys of working in foreign lands.

70 Shareware

Tony loads all his programs aboard the ARC. Thorough description of the rival archive packages.

78 Technical Tips

91 Pascal Column

The \$6.00 scanner part III. Fleshing out the software.

93 Culture Corner

CP/M CORNER

74 CP/M Notes

76 Kaypro Column

FUTURE TENSE

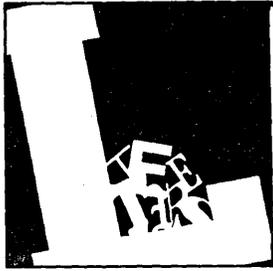
84 Tidbits

Programming an EGA.

96 Last Page

The Cover:

The Mandelbrot on this issue's cover was photographed for us by Dean Klein. He generated the image on PC-Tech's 34010 Hi-res color board. (Unfortunately we can't show you the animated version he was showing at SOG.)



Letters

Seagate And C

Thanks for a delightful magazine. Love the whole thing, especially the Culture Corner.

The Seagate ST 251 is a puzzle. We've had a rash of bad 251s here at Applied Management. Virtually every one has failed after several months on the job. I have one here in my home office, and price/performance-wise it's a gem. However, I expect a failure any time now. Meanwhile, Seagate is promoting them like they were movie stars. That would seem like a bad idea if they were flawed, wouldn't it?

I know Gary Entsminger printed those benchmarks for the C statistics program (issue #38 Tidbits) just to get reader feedback. I imagine that every fourth C programmer in the country sat down and ran the same program. Twice the time of Prolog?

Perhaps it's our supercooled machines out here, but my results are much better. Using Aztec C version 3.20e on a WiseTEK 10 MHz AT (no coprocessor) the program ran in about 5.25 seconds.

Guess I'm not much impressed with Turbo C. Aztec has a much more standard library. Not "improved" like the Turbo C library (see lseek ()). There are also typos and non-running examples in the Turbo documentation. In comparison with any other C compiler I've had experience with, Aztec C86 is much underrated. I started with Aztec on CP/M four years ago and see no reason to change.

Bob Worthy
Worthy Computer
1132 Breckenridge
Helena, MT 59601

Subhead Intentionally Left Blank

The blankety-blank Culture Corner in issue #38 left me speechless. The reason,

of course, has to do with the problems of obtaining a PAGE INTENTIONALLY LEFT BLANK, or PILB. Until now they were only available pre-printed in large quantities or as a custom-designed product from professional printers.

Imagine my frustration with the inability of any word processor to produce a PILB. I can't count the number of sleepless nights I've grappled with this problem until my mind has gone blank.

With the development of page make-up programs like Ventura, I was sure the problem had been solved. But alas! I was still unable to cajole my LaserJet into printing a PILB.

I finally concluded that I would have to write my own program to accomplish this. After many more sleepless nights (thank heavens for daytime!), I finally succeeded. As evidence of my success, I am writing this letter on the reverse side of a PILB. This would have been inconceivable before I developed this marvelous little program. And most wonderful of all, it operates with any word processor, under any operating system, and with any printer.

Of course my program has no name and it exists quite happily on a blank, unformatted disk. I am willing to share it with you and your readers. Anyone who would like a copy should send me a DISK INTENTIONALLY LEFT BLANK along with a CHECK INTENTIONALLY LEFT BLANK (except for the signature). Registered owners become eligible to receive documentation (a manual consisting of PILBs) and future updates. The first planned update will create a SCREEN INTENTIONALLY LEFT BLANK.

Joe Backus
P.O. Box 794
East Lansing, MI 48826
(000) 000-0000

Shareware/Public Domain Confusion

While speaking to you on the phone recently, I said I'd go over your "Best of Micro Cornucopia" poster and detail the problems I see with it. From the very start, the poster's subtitle, "A Catalog of Public Domain Software," is in error.

Much of the software displayed in the poster is shareware—quite distinct from public domain software. Shareware programs are copyrighted, public domain programs aren't. A shareware author retains all rights to the program, but allows it to be distributed freely. That's the only difference between a shareware program and a commercial program. If you give a friend a copy of PC File, you're not breaking the law; if you give her a copy of Paradox, you are.

You list PIANOMAN prominently—that's good. But you wrongly label it public domain—that's very bad. In theory, wide distribution is good for shareware programs. But if the people receiving them don't register, there's no benefit. I don't know why you chose to use that erroneous phrase on my program—I don't see it anywhere else. With all the rest of the programs (except two), you simply *imply* that they are in the public domain.

The two exceptions are PROCOMM and PC Write. You refer to PROCOMM as shareware and PC Write as "user supported software." That's nice, but will your readers understand how that's different from public domain?

In the same vein, you're giving poor Tony Barcellos a bad name. He writes great columns about *shareware* programs under the heading "In The Public Domain." Do yourselves, your readers, and Tony a favor. Get the heading right for that column.

I first contacted Micro C about the poster because a friend told me it listed PIANOMAN as public domain. I as-

(Continued on page 69)

Blaise puts the Accent on C with C TOOLS PLUS/5.0™

Enhance your Microsoft C programming environment with C TOOLS PLUS/5.0™—a new, quintessential library of C functions. C TOOLS PLUS/5.0 from Blaise Computing Inc. puts a prime accent on quickly building professional applications using the full power of Microsoft C Version 5.0 and QuickC. Now you can concentrate on program creativity by having full control over DOS, menus, interrupt service routines, memory resident programs, printer and keyboard control, and more!

C TOOLS PLUS/5.0 prebuilt libraries are ready to use with either QuickC or the Microsoft C Version 5.0 command line environment. Complete documented source code is included so that you can study and adapt it to your specific needs. Blaise Computing's attention to detail, like the use of full function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes C TOOLS PLUS/5.0 the choice for experienced developers as well as newcomers to C.

Continuous refinement of Blaise Computing's library products has produced a collection of tools that are unsurpassed for reliability, functionality and ease of use. Built upon the widely acclaimed C TOOLS PLUS, C TOOLS PLUS/5.0 includes such highly-developed features as:

WINDOWS

- Stackable, removable.
- Optional borders, cursor memory.
- Accept user input, formatted output.
- "print" window-oriented output. **NEW!**

◆ INTERRUPT SERVICE ROUTINES

- Capture DOS critical errors and keystrokes.
- Install hardware interrupt handlers.

◆ RESIDENT SOFTWARE SUPPORT

- Install, detect and remove memory resident programs.

◆ MENUS

- Horizontal and pulldown. **NEW!**
- Lotus-style support. **NEW!**

◆ INTERVENTION CODE

- Schedule C functions at specified times, intervals or with a "hot key." **NEW!**
- Take full advantage of DOS, even from memory resident programs. **NEW!**

◆ FAST DIRECT VIDEO ACCESS

- All monitors, even EGA 43-line mode.

◆ PRINTER CONTROL

- Access BIOS print functions. **NEW!**
- Control the DOS PRINT utility. **NEW!**

◆ UTILITIES AND MACROS

- Take advantage of DOS file structure.
- Manipulate data types, far & near pointers. **NEW!**
- Access any memory areas with fast "peek" and "poke" macros. **NEW!**

C TOOLS PLUS/5.0 supports the Microsoft C Version 5.0 and QuickC compilers, requires DOS 2.00 or later and is just **\$129.00**.

C ASYNCH MANAGER™ Version 2.0 IMPROVED!

C ASYNCH MANAGER is a library of functions designed to help you incorporate asynchronous communication capabilities into your application programs. Version 2.0 has been rewritten especially for Microsoft C Version 5.0 and Borland's Turbo C. Simultaneous buffered input and output to both COM ports at speeds up to 9600 baud, XON/XOFF protocol, modem control and XMODEM file transfer are among the many features supported and is priced at just **\$175.00**.

Blaise computing Inc. has a full line of support products for both Pascal and C. Call today for your free information packet.

BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

Reader Service Number 5

Now first \$129 and supports
Microsoft C 5.0 and QuickC

BLAISE N U

VIEW MANAGER \$275.00

Screen control; paint screens; block mode data entry or field-by-field control with instant screen access. For C or MS-Pascal.

LIBC TOOLS \$129.00

Windows; ISRs; intervention code; screen handling and EGA 43-line text mode support; direct screen access; DOS file handling and more. For Turbo C.

Turbo POWER SCREEN

COMING SOON! General screen management; paint screens; block mode data entry or field-by-field control with instant screen access. For Turbo Pascal.

Turbo POWER TOOLS PLUS \$99.95

Screen and window management including EGA support; DOS memory control; ISRs; scheduled intervention code; and much more. For Turbo Pascal.

Turbo ASYNCH PLUS \$99.95

Interrupt driven support for the COM ports. I/O buffers up to 64K; XON/XOFF; up to 9600 baud; modem and XMODEM control. For Turbo Pascal.

PASCAL TOOLS/TOOLS 2 \$175.00

Expanded string and screen handling; graphics routines; memory management; general program control; DOS file support and more. For MS-Pascal.

ASYNCH MANAGER \$175.00

Full featured interrupt driven support for the COM ports. I/O buffers up to 64K; XON/XOFF; up to 9600 baud; modem control and XMODEM. For MS-Pascal.

KeyPlayer \$49.95

"Super-batch" program. Create batch files which can invoke programs and provide input to them; run any program unattended; create demonstration programs; analyze keyboard usage.

EXEC \$95.00

NEW VERSION! Program chaining executive. Chain one program from another in different languages; specify common data areas; less than 2K of overhead.

RUNOFF \$49.95

Text formatter for all programmers; flexible printer control; user-defined variables; index generation; general macro facility. Crafted in Turbo Pascal.

LIGHT TOOLS \$99.95

Windows; ISRs; EGA 43-line text mode; direct screen access; DOS file handling and more. For the Datelight C compiler.

TO ORDER CALL TOLL FREE

800-533-5037

TELEX NUMBER-338139

Yes! Send me the prime accent!

Enclosed is \$_____ for _____ copies of _____
 Please send me more information on your products.
 CA residents add Sales Tax. Domestic orders add \$4.00 for
 UPS shipping, \$10.00 for Federal Express standard air.
 Name: _____ Phone: (____) _____
 Address: _____ State: _____ Zip: _____
 City: _____ Exp. Date: _____
 VISA or MC#: _____

Microsoft
is a registered trademark of
Microsoft Corporation. QuickC
is a trademark of Microsoft Corporation. Turbo C
is a registered trademark of Borland International.

Drawing The Mandelbrot And Julia Sets

Attractive Fractals

This article will deal with the Mandelbrot set and its close relative, the Julia set. Beware ye who read on. Fractals are addictive and incredibly greedy with your computer's resources. One fractal can tie it up for days.

Way back in issue #33, I wrote about geometric fractals—those infinitely detailed, wildly irregular, self-similar curves. Since then I've become more interested in the class of fractal which deals with domains of attraction. You may recognize the figure on this issue's cover. It's one of these non-geometric fractals.

Why Bother?

Who needs a program that gobbles up so much time? Actually, fractals would be worth the trouble if only for their beauty. One of the characteristics of fractals is that they look very familiar—very natural. They'll remind you of anything from trees, to galaxies, to landscapes, to eddies in a stream, to images you'll remember from grade school when you saw your own blood under a microscope.

Remember the Genesis world in Star Trek II? It was a fractal world. For a good look at the wide variety of fractals (and enough math to keep anyone happy), see B. B. Mandelbrot's *The Fractal Geometry of Nature* and *The Beauty of Fractals* by H.-O. Peitgen and P. H. Richter.

They aren't just pretty pictures, though. Fractals model complex systems very successfully. Scientists study the scattering of stars in galaxies, turbulent flow, stock market behavior... Douglas Adams even suggests in his latest book, *Dirk Gently's Holistic Detective Agency*, that fractal geometry can predict the movement of laughter through a crowd

of people. I got a chuckle out of that.

Domains Of Attraction

Dr. Mandelbrot (the father of fractals) has spent a great deal of time studying domains of attraction within the complex number plane.

The notion of domains of attraction is central to these fractals, so let's talk about it. Look at the function:

$$f(z) \leftarrow z^2$$

$$\text{where: } z = x + iy$$

This may be unfamiliar notation to some of you. Z is a complex number. That means it has two components—a real part and an imaginary part. Real numbers are the ones you're used to. They can be represented as a fraction involving integers.

Imaginary numbers have an i component also. i is the square root of -1 . How do you take the square root of -1 ? You can't. So we'll imagine that you can and call the result i . It helps to think of complex numbers as vectors—that is, a quantity with a direction. Complex numbers exist as vectors in the complex plane.

Complex number theory is a very powerful tool. For example, you can use it to predict the response of active circuits in electronics quite easily. Or you can work with differential equations—considerably more difficult. Everything we talk about from now on will take place in the complex number plane.

Back to $f(z) \leftarrow z^2$. Run this function through an iterative process.

That is:

$$z_{n+1} = z_n^2$$

If we start with some initial value for Z and iteratively plug the results of Z^2 back into the function, one of three

things happens. For an initial Z greater than one, Z^2 becomes larger and larger. It approaches infinity. So we can say that infinity is an attractor for Z^2 .

If the initial Z is less than one, Z^2 becomes smaller and smaller. It approaches zero, another attractor for Z^2 . If Z equals one, then Z^2 doesn't change. One squared will always be one.

When I say Z is one, I'm talking about its modulus or size. You find the modulus of a complex number by taking the square root of the sum of the squares of its real and imaginary parts—its distance from the origin of the complex plane.

So Z^2 has two attractors, infinity and zero, and a border between them consisting of a circle with radius one.

This is a simple function. What if we add a complex constant? The function becomes:

$$f(z) \leftarrow z^2 + c$$

$$\text{where: } c = p + iq$$

This function is the heart of both the Mandelbrot and Julia sets. What a surprise: that such a simple function could open up a world of infinite complexity and beauty.

The Mandelbrot Set

The Mandelbrot set is a connected set (see Figure 1). That means that every point in the set touches other points in the set. There aren't any islands disconnected from the main figure. Infinitely many smaller copies of the Mandelbrot set attach to the main set by filaments.

What are we looking at in Figure 1? Really just an extremely deformed version of the circle we saw for the function $f(z) \leftarrow z^2$. Extremely deformed is an understatement. The Mandelbrot set has been called the most complex object in mathematics.

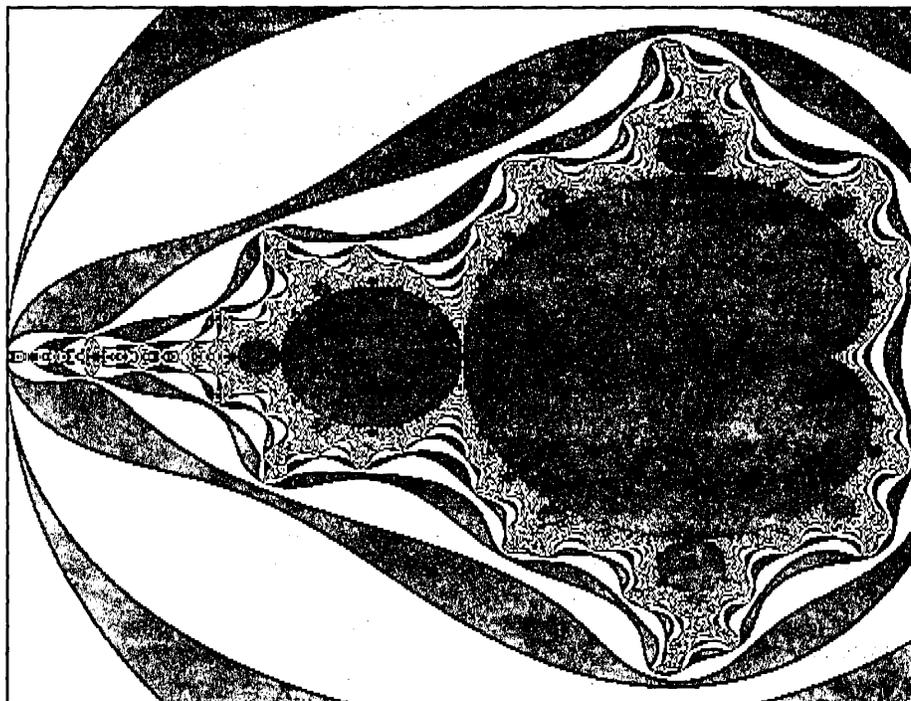


Figure 1 - The Mandelbrot Set

Definition time:

$$z_{n+1} = z_n^2 + C$$

Starting with $Z_0 = 0$, the Mandelbrot set is the set of all values of C such that Z remains finite after any number of iterations. Seems like that would be a hard test to deal with.

```
while (Z < infinity)
  do another iteration
```

This code would be slow even on a Cray. There are two things working for us though. First, complex number theory says that, for this function, if the size of Z ever exceeds 2, then Z will eventually blow up (become infinite). But it could still take a long time for Z to exceed 2, so we'll also limit the number of iterations to 1000.

It turns out that there are points which blow up after 1000 iterations, but not enough to worry about. Basically, you use this value, the 1000 iterations, to set the resolution of the picture. The more iterations, the better the resolution.

So all we really have to test for is:

```
while (size of Z < 2) and
      (# iterations < 1000)
  do another iteration
```

To generate the Mandelbrot set, look at each value of C on the screen (the screen is a map of C —the x-axis represents C 's real part and the y-axis represents its imaginary part). See what happens to $Z^2 + C$. The initial value of Z will always be zero. If after 1000 iterations the size of Z remains less than 2, then C is within the Mandelbrot set and it gets colored black.

The other bands of color show how fast the points outside of the Mandelbrot set blow up. A point close to the border may take 900 iterations to blow up while one farther away could blow up in one iteration. That makes sense. The larger initial values of points far away from the Mandelbrot set *should* make them blow up faster.

The most detailed and interesting pictures come from zooming in on the border region. That's why they take so long to draw—each of the points is close to the Mandelbrot set and therefore takes many iterations to blow up.

The Julia Set

The Mandelbrot set explores the C plane by looking at the behavior of $f(Z) = Z^2 + C$ for various values of C , and $Z_0 = 0$. Julia sets turn things around a bit. They set some value for C and explore the Z plane. Instead of a different value of C for each point plotted, we use a different value of Z_0 .

You'll find lots of room for exploration here. The Mandelbrot set is one well-defined object. You can zoom in on various portions of it and get wildly different pictures, but it's still just one object. On the other hand, there are infinitely many Julia sets—one for each value of C .

The relationship between the Mandelbrot and Julia sets allows us to predict, to a certain extent, what the Julia sets will look like. Remember, the Mandelbrot set is plotted in C space, while each value of C generates a different Julia set in Z space.

If we take C from within the Mandelbrot set, the corresponding Julia set will be connected. As C passes over the border of the Mandelbrot set, the Julia set becomes disconnected. When C is far away from the Mandelbrot set, the Julia set turns into a very diffuse, disconnected dust.

Graphics Tools And Hardware

Before we get into the coding, let me talk briefly about hardware. I spent a lot of time playing with CGA on a large screen monitor. But it was really unsatisfying. The resolution just isn't good enough. And detail is what fractals are all about. I settled on a Hercules card because its resolution is quite good and there are lots of them out there. If you want to play with Mandelbrot and Julia sets, I'd recommend that you have *at least* the resolution of a Hercules card (720 X 348). EGA (640 X 350) works nicely, too.

Earl Hinrichs from PC Tech was one of the stars of this year's SOG. He showed off PC Tech's new 34010 based video board by creating truly fantastic Mandelbrot and Julia sets. Earl had his own assembler routine running on the 34010. So he had excellent color resolution along with blazing speed. If you have an itchy pocketbook and a yearning for great beauty on your monitor, look into this board.

I'm using Borland's Turbo C to generate fractals these days. I had hoped that Turbo C Tools from Blaise Computing would supply the graphics routines I needed to control the Hercules card. Unfortunately, they implement only those graphics procedures supported by the BIOS. So hello CGA and EGA, goodbye Hercules. See *A Hercules Primer* in this issue for a library of graphics functions.

Coding For These Critters

For the Mandelbrot set, we only have to find out what region of the set the user wants to view, and then examine each point in that region to see if it's in the set. Very easy. And very slow.

Figure 2 shows the guts of a simple program to explore the Mandelbrot set. I'll leave it to you to add procedures to get parameters from the user and to save and show the different pictures you generate. Or you can be lazy and download the complete program from Micro C's RBBS at (503) 382-7643. Look in the CURRENT ISSUE area. It's also available on disk (along with the rest of the code from this issue) for you modemless folks. Give us a call at 1 (800) 888-8087 and ask for the Issue #39 disk. (Disks are \$6 for U.S. subscribers, \$8 for non-subscribers and foreign readers, postpaid.)

The function `mandel` receives the screen's boundaries as parameters. They define a portion of the C plane. The first two lines of code calculate the change in P between adjacent columns and the

Figure 2 - Mandelbrot Set Generator

```
#define sqr(x) (x*x)
const int maxcol = 719; /* set up for herc card */
const int maxrow = 347;
const int max_colors = 2;

int max_iterations; /* 1000 for best results */
int max_size; /* usually 4 */

void mandel (float Pmax, float Pmin, float Qmax, float Qmin)
{
    int color, row, col;
    float P, Q, modulus, deltaP, deltaQ,
        Xcur, Xlast, Ycur, Ylast;

    deltaP = (Pmax - Pmin)/(maxcol - 1); /* determine real axis increment */
    deltaQ = (Qmax - Qmin)/(maxrow - 1); /* determine imaginary axis increment */
    for (col = 0; col <= maxcol; col++) /* look at each point on screen */
        for (row = 0; row <= maxrow; row++)
        {
            P = Pmin + col*deltaP; /* determine coordinates of point in C plane */
            Q = Qmin + row*deltaQ;
            Xlast = Ylast = modulus = 0.0;
            color = 0;
            while ((modulus < max_size) && (color < max_iterations))
            {
                /* go around until function blows up or until max_iterations */
                Xcur = sqr (Xlast) - sqr (Ylast) + P; /* find components of next Z */
                Ycur = 2 * Xlast * Ylast + Q;
                color++;
                Xlast = Xcur; /* update last Z */
                Ylast = Ycur;
                modulus = sqr (Xcur) + sqr (Ycur); /* find size of Z */
            }
            draw_point (col, row, (color % max_colors));
        }
} /* mandel */
```

change in Q between adjacent rows. Once we know these step sizes, we use the nested for loops to look at each point (P, Q) on the screen.

When testing a point, the number of iterations of the while loop determines the color of the point. So call the loop counter "color." Xcur is the real part of X_{n+1} and Xlast is X_n 's real part. Expand $Z^2 + C$ yourself to verify the first two lines of the while loop. Remember, $i^2 = -1$.

Note that we've really calculated the square of the modulus in the last line of the while loop. No sense wasting the time to take the square root when it's just as easy to compare modulus² to 2².

Finally, draw the point using `(color % max_colors)` for its color. The modulo division lets you use the available palette of colors repeatedly throughout the range of `max_iterations`. Choose a multiple of `max_colors` for the value of `max_iterations`. This forces points within the Mandelbrot set to be drawn in color 0, typically black.

Julia Sets

Programming for Julia sets is very similar (see Figure 3). Pass values for P and Q to the function `julia`. These con-

stants determine which Julia set we'll see. Since we're looking at Z space rather than C space, `julia` receives screen boundaries in terms of X and Y instead of P and Q.

You'll see two differences in the body of `julia`. (I won't say it—it would be too easy.) First, we're calculating increments in X and Y. And second, the value for Z_0 comes from the screen location being tested. In `mandel` it was always zero.

Execution Speed

The best fractals I've done have each taken over a day to generate. Even on a 10 MHz 186. That's a real drawback. Especially in a place like Central Oregon. I enjoy our thunderstorms, and during the last few weeks we've had some doozies. But when they destroy a day's worth of work...

So we'd really like to speed up the process. Sinking some bucks into faster hardware is the best solution. The PC Tech board creates fractals in a matter of minutes. And I'll be cranking out fractals on a parallel processing board stuffed with V20s in the near future. Stay tuned.

Cheaper, though less effective, solutions exist in software. One speedup

technique takes advantage of symmetry. The Mandelbrot set is symmetric about the real axis. So, as long as the real axis is in the center of the screen, we only have to go through the iterative process for either the top or bottom of the screen. As each half column is generated, we draw its reflection on the opposite half of the screen.

This cuts the generation time in half but doesn't work when you zoom in on an area which doesn't contain the real axis. No more symmetry there.

The Julia set has a different symmetry. Instead of symmetry about an axis as in the Mandelbrot set, we have symmetry about a point—the origin. Examine only the points in the top of the screen. When a point (row, col) has been colored, fill in its companion at (max_row - row, max_col - col). Again, this technique won't work unless the origin is at the center of the screen (or, at least, on the screen).

Or you can reduce max_iterations to some value below 1000. The lower the value, the less time it takes to draw the set. But the resolution suffers. This method allows coarse exploration to take place without as much of a time commitment. When you find an interesting region, go back to the full 1000 iterations for higher resolution.

Earl came up with an algorithm which deals with large portions of the screen rather than points. Split the screen into sections. Now look at the center point of each section. How does it behave when put through the iterative process? Color the entire section according to the behavior of the center point.

Now split each section into more sections and repeat the process. You can see that we're starting at a very low resolution and improving that resolution with each new subdivision of sections. The advantage of this method is that you don't have to calculate every point on the screen. After only a few rounds, you can get an idea what the fractal will look like and decide whether it's worth continuing.

What's Next?

A year ago I finished the first fractal article by promising further exploration, including a dip into fractal music. I lied. I got sidetracked by Dr. Mandelbrot and a multitude of other projects. This time I'm not lying. Along with music, I'll be looking into applications of fractal mathematics in the real world. Till then.

Figure 3 - Julia Set Generator

```
void julia (float Xmax, float Xmin, float Ymax, float Ymin, float P, float Q)
{
  int color, row, col;
  float modulus, deltaX, deltaY,
        Xcur, Xlast, Ycur, Ylast;

  deltaX = (Xmax - Xmin)/(maxcol - 1); /* determine real axis increment */
  deltaY = (Ymax - Ymin)/(maxrow - 1); /* determine imaginary axis increment */
  for (col = 0; col <= maxcol; col++) /* look at each point on screen */
    for (row = 0; row <= maxrow; row++)
      {
        modulus = 0.0;
        color = 0;
        Xlast = Xmin + col * deltaX; /* find coordinates of Z [0] */
        Ylast = Ymin + row * deltaY;
        while ((modulus < max_size) && (color < max_iterations))
          { /* go around until function blows up or until max_iterations */
            Xcur = sqr (Xlast) - sqr (Ylast) + P; /* find components of next Z */
            Ycur = 2 * Xlast * Ylast + Q;
            color++;
            Xlast = Xcur; /* update last Z */
            Ylast = Ycur;
            modulus = sqr (Xcur) + sqr (Ycur); /* find size of Z */
          }
        draw_point (col, row, (color % max_colors));
      }
} /* julia */
```

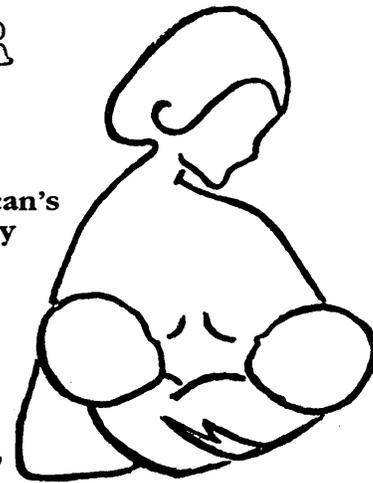
MULTIUSER DOS NOW!

You can develop DOS multi-user applications now with Pecan's new Poly Power System. On any XT, AT, PS/2 or compatible. Under any version of DOS 2.0 or later.

All you need is Pecan's Poly Power System, one or more of our compilers: UCSD Pascal, Modula-2, FORTRAN-77, BASIC, Assembler (C is coming soon), and inexpensive dumb terminals on serial ports.

You can even have multiple programmers develop on a single machine.

Poly Power System, \$399.95. Compilers, \$79.95 each. Call us for full details and to place your order.



PECAN
The UCSD Pascal Company
1-800-63-PECAN

Pecan Software Systems, Inc.
1410 39th Street
Brooklyn, NY 11218
718-851-3100

Pecan Software Europe, Ltd.
MGM House, Oakfield Grove
Clifton, Bristol, BS8 2BN, England
0272 733633

Reader Service Number 29

Desktop Graphics

I have a very carefully worked out method for finding something to write about. I hang around the office until something clubs me in the head. However, this time I gave myself an assignment, outlined where I was going with it, and began researching. For a while it looked like this procedure wouldn't work at all.

I'd intended to give you an overview of all the great and glorious ways to scan graphics, edit the scanned images, create graphics from scratch, and incorporate all the above into desktop packages like Ventura.

I managed most of that. But it was a struggle. I'd assumed that I could just get on the phone, line up some very graphic people, take notes, and pass along what I learned. However, finding the people wasn't all that easy.

After a lot of digging, I found two folks who sounded interesting and arranged to spend half a day with each, doing on-site interviews. What follows is a transcription of the discussions.

Two weeks ago Carol Steffy, our new graphics designer, and I headed for Oregon's city lights in our search for folks doing computer graphics. We started the day in Corvallis with Dr. David Sullivan, an associate professor of information science at Oregon State University.

David had written a computer textbook using Ventura Publisher, and I'd heard through Z-Soft that he was also using a graphics package called Publisher's Paintbrush. (And, even if we didn't learn anything about graphics, he should have a wealth of information about dealing with publishers.)

He was a technical person using an AT, Ventura, and a graphics package to produce books. Not only could Carol

and I get information for this article, we could try out what we learned on this article.

In The Beginning

On arriving at David's ranch-style house (packed into a suburban subdivision), we were shown into a large bedroom/family room mostly taken over by desks and computers. A patio door opened onto a small fenced back yard.

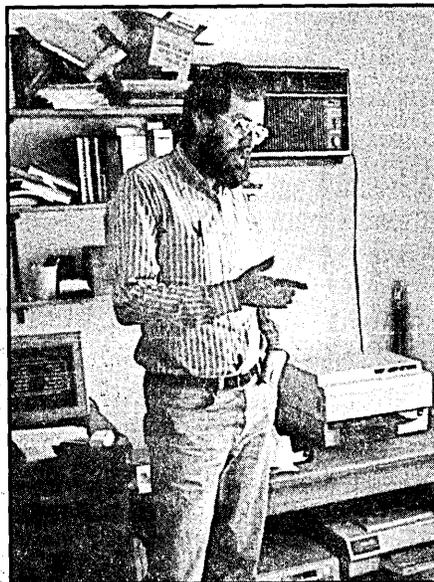


Figure 1 - David Sullivan.

David started the discussion with a description of the ways desktop publishing has changed the way books are designed. How he works with designers.

David Sullivan: It used to be that all the interaction between me and the publishers happened on the phone, things were outside of my control. It ended up that I'd get these design suggestions at the end of the project that said, "We need to have boxed inserts scattered throughout to match the design we created after you submitted the manuscript."

It's sort of like collecting all the materials for a house and then bringing in the architect. He sits down, takes an inventory of what you've got, and then designs the house around what's there.

Then he says he wants you to go out and get a bunch more materials. He couldn't find a spot for some of the material you already had, so he threw it away.

(Laugh)

That really gets to you in short order.

What you say is, "I want a design to start with and I'll paginate. I'll do all the work because then I'll be in control."

There are things you can do now that you really couldn't do before. If you're writing a boxed insert, you know how long the page is going to be. If you're two inches short of filling the page then you can go back and write a little more.

You have the designer, working at the beginning of the project, designing what the concept should be and the construction worker, the author, working within the envelope.

Micro C: How do you handle graphics in your books?

David Sullivan: It's easiest in Ventura to put all the artwork in a separate art chapter file—if you have art, piece after piece, page after page. I just open up holes, little grey patches, the same size as the art so the printer knows where to position it. After all, how long does it take someone to physically paste it in if the hole's there? The caption's there.

Micro C: How do you create the artwork?

David Sullivan: That depends. Some of it is line art and that's done by someone else. The text stuff is just samples of what output would look like: Lotus FreeLance, for instance, straight off the screen. I wind up with a wide variety of illustrations from a wide variety of programs. Many will have overlays in

color with labels and arrows pointing out what I'm discussing in the text.

You can really get a good feel of what a software package's output looks like when you see it on the screen (through screen capture).

My whole goal is to do such a good job with the artwork and the text that the competition (other writers and other publishers) can't compete in the category that I select.

After all, if I'm going to do something, I'm going to do the best I'm able.

Micro C: What got you started in this form of publishing?

David Sullivan: I took last year off, without pay, to spend full time going to conferences. If I hadn't done that, there's no way I could have switched over to desktop publishing. Jumping in early, the solutions have not really been there.

Ventura 1.0 was a pretty crude product. Took me weeks to work around some of the rough edges.

What the publisher sent me is all the stuff they would normally give a professional typesetter. I had to look at all that and try to figure out how it matched up to what I could do with Ventura Publisher. Then I created the style sheets.

I probably have two months of full time work in the style sheets. Style for chapters, boxed inserts, artwork chapters, glossary—and each is different. Unfortunately you can't print out what's in a style sheet. A tremendous shortcoming, and it wouldn't take long to reverse engineer the data inside a style sheet file and create a report writer, then sell it for a couple hundred dollars a copy.

Micro C: We'd certainly be interested in a product like that. What about your equipment? How did you select that?

David Sullivan: The budget has been whatever I've wanted to have. I got the Mac because Adobe's downloadable fonts wouldn't download from an IBM

to a PostScript printer. They claimed you could get the Mac versions, have someone to convert them over to IBM disk format, and then download them from the IBM. I went through the process. It didn't work, and they weren't willing to give me any technical support.

They just said, "We believe this can be done. We did it in our technical department, but we won't tell you how we did it."

(Editor's note: I later heard that if you initialize Adobe fonts on the Mac, save them as a text file (not Adobe font file), transfer them to the PC and add:

```
0 serverdict
begin exitserver
```

it's transferred, makes some sense.

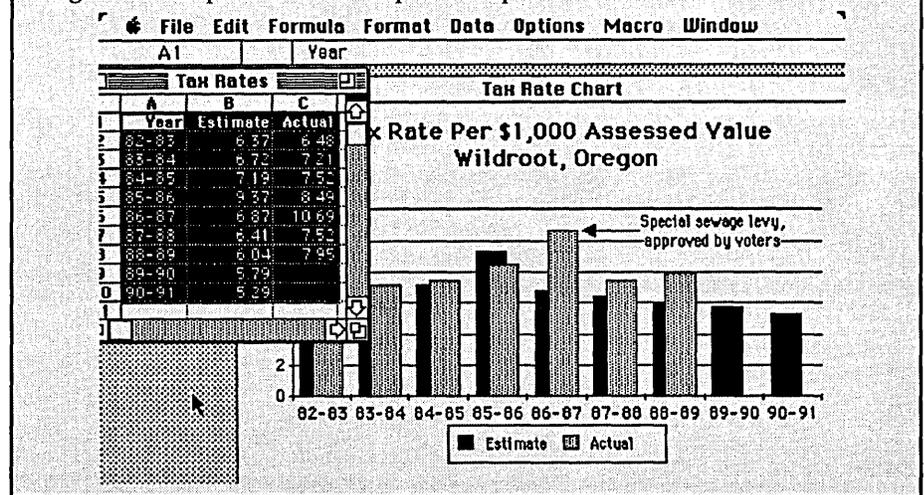
Micro C: How about converting graphics files?

David Sullivan: The thing I find amusing is that grabbing Macintosh screens and publishing them through a PC is often easier than it is to grab PC screens. That seems really strange, but it's true. Though it runs on the PC, Ventura Publisher has a box for Mac Paint-type stuff and you just click on it.

Built into the Mac operating system is a screen capture utility. They name their utilities flowers, open apples and closed apples. Cutesy stuff. I haven't figured it all out.

It's on the IBM side that you have the problem. You need lots of different screen captures because they don't all

Figure 2 - Sample of Screen Capture Graphics.



to the start of the text file, you have a PostScript font file which can be copied directly to the PostScript device.

Anyway, I got a Mac. Because of the Mac, I needed a network so the IBM and Mac could talk. I got the Centram Tops network. Then I needed to bone up on conversion utilities so the data, when

work with all the PC programs. There are so many different standards for different screens.

I spent \$100 to capture images from Microsoft windows because Windows freezes up if there are certain memory resident programs. Other screen capture programs are sensitive to some TSR

routines. None of these things are documented.

I called Microsoft and asked how they grabbed screens for their documentation. After being passed around for awhile, I reached someone who knew. He basically said they had this quick and dirty thing that sometimes works with Windows, but not reliably. They certainly wouldn't be willing to release it to anyone else.

I'm thinking, "Wait a minute. You're creating a window operating environment that doesn't have a built-in screen capture, and you can't even get one to run in-house?"

Is there a problem here?

They finally recommended that I buy Hot Shot, which works if you have the particular type of monitor that Hot Shot captures. EGA is one. The reason I can't use a really high-resolution screen is I need to take snapshots, and EGA is the highest up that list for the screen capture utilities.

You look at those problems on the IBM side and all of a sudden, even though I use the IBM 98% of the time, there's a lot of logic in that little box (Mac) over there.

Sure, you're single-sourced to one hardware manufacturer, but the real question is: "Are you single sourced to one software philosophy?" If we all stick to the IBM then I think we're single-sourced to a bankrupt philosophy.

With IBM you get a weird command line style, hierarchical menu bars in spreadsheet programs, pull-down menu bars in Reflex, function-key interfaces in Word Perfect, control-key interfaces in WordStar, and you've got users who'd rather get things done than learn eight different interfaces. Someone who buys a Mac will use more software because they aren't so intimidated.

If it hadn't been for the Mac, I doubt we'd be seeing OS/2.

Micro C: If you were writing an article on creating graphics for your desktop program, what would you tell the *Micro C* audience?

David Sullivan: I'd be cautious about suggesting that tools for graphics are the best thing since sliced bread. Illustrators can work with pen and ink more successfully and more quickly than they could do a one-shot thing on the computer. Do you want to create a sketch of a floppy disk on the computer? As a one-shot thing? That's assuming, of course, you're not doing the standard graph or pie chart, but a technical illustration.

I've looked at Adobe Illustrator (for the Mac) and other programs like that, but by the time you've learned to use all the tools and get everything right, the average person with pen and ink could have done more illustrations, more quickly.

If what you're documenting is what's going on in the PC, then there are good canned approaches like Frieze and Hot Shot for capturing screen menu displays. SideKick's good for capturing screen text.

My conclusion is that graphics are coming along more slowly than the text.

Plus, there's another aspect. Giving someone an electronic paint program is like giving someone a brush and a buck-

I'm thinking, "Wait a minute. You're creating a window operating environment that doesn't have a built-in screen capture, and you can't even get one to run in-house?"

et of paint. The skill isn't in the package, the skill is in the artist. What does the artist want to do? If there's an artist who really wants to use graphics programs, that's fine.

Micro C: What about scanning and modifying?

David Sullivan: The only scanning I've played with has been photographs. Even then it doesn't seem practical, because the halftone process is so quick and easy (at print shops).

The majority of the illustrations I deal with are either example screen displays or extremely involved illustrations.

Micro C: What about the possibility of scanning line art? Then tracing it? Modifying it? Things that wouldn't be as easy to do with traditional tools. And doing it very precisely. And why limit yourself to trying to duplicate the media

that's already available? Why not synthesize graphics using electronic tools like we synthesize music?

David Sullivan: You need to talk to a graphics artist to get any enthusiasm. I'm really clear in my mind that people who aren't artists shouldn't be dealing with these sorts of packages. Unless your job is graphic art then these tools will only distract you from what you should be doing.

Micro C: However, take the synthesizer analogy one step further. A very powerful synthesizer doesn't make you a musician, but there will be people who do beautiful things with a synthesizer who would have done nothing if they'd had only a piano.

David Sullivan: I have almost no artistic ability. My stick figures are still back in the third grade. So given that, I make sure that when I need art, someone else does it. Like everyone else, I generalize well; if I can't do it, then nobody else will.

Micro C: There's something much more mysterious about creating an image from scratch, an image that reflects what you really want to say. I think the thing that happens when people scan images is that they begin with someone else's work. They want to use it as their own, but it isn't.

David Sullivan: Plus the original creator may be upset because you've ruined his work. But I've heard of no one who's been taken to court because they stole all the visual elements of a publication. That's why I think so many people can successfully do desktop publishing.

Later That Day

Wes Wait is a completely different kind of person. Not concerned with the text, almost anti-techie, he's the *complete* graphic designer and illustrator. The architect in David's analogy.

Wes comes into a project at the end. The text is finished, signed off, and delivered on disk. A Macintosh disk.

As Carol and I entered his home/studio high in the wooded hills, west of Portland, we heard a low, quiet chanting. We left our shoes at the door.

Micro C: A cursory glance indicates you're strictly Macintosh-based, is this a problem when you get materials from clients? Can they all generate Mac disks?

Wes Wait: Either that or send the text over the phone. I've been fortunate. I tell my clients that if they have an IBM to just have a secretary re-key it into a Mac, or we get the PC disk translated.

I'm not a typist; I have no desire to be a typist.

I bought a scanner and within two months they were supposed to bring out an OCR (optical character reader) program for it. It's now nine months later.

Micro C: How long did this first project take you?

Wes Wait: I don't remember exactly, it was two years ago. I was trying to learn how to use the software and design something real at the same time. It was an exercise to work out a relationship with the software. I suspect the project took me a couple of weeks, including a lot of phone calls to Aldus. They've always been notorious. I've never been able to call them and get someone on the phone immediately. They always call back two hours later. By then you've figured it out or you're dead in the water until they call.

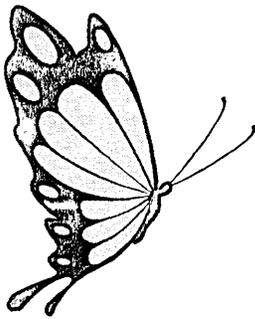
With all this technology, there are ways to get around all that.

Micro C: Do you have Adobe Illustrator?

Wes Wait: Yes. (He digs out a butterfly.)

Micro C: Looks like it works beautifully.

Figure 3 - Wes' Adobe Butterfly.



Wes Wait: It's an amazing tool.

Micro C: You went through the Linotronic L300 for your output?

Wes Wait: Yes.

Micro C: Did you have anything to do with the accompanying text?

Wes Wait: They gave me the text signed-off as error free and then I formatted it—made all the typographical decisions. They proofed it. Of course, clients never proof it like they should. Even when the technology lets them see it at a very early stage (300 dpi Apple-Writer output which looks amazingly similar to the finished piece), they still find errors at the end.

There's really no difference between

the first-round proof out of the computer and last-round proof off the typesetter. But they don't focus like they should. I don't read the text anymore. It's their responsibility.

I'm just cake decorating.

Micro C: So there's a market for cake decorators.

Wes Wait: Oh yes, there always has been. There was a time when I was involved in projects from the start. I had control over what was written, total

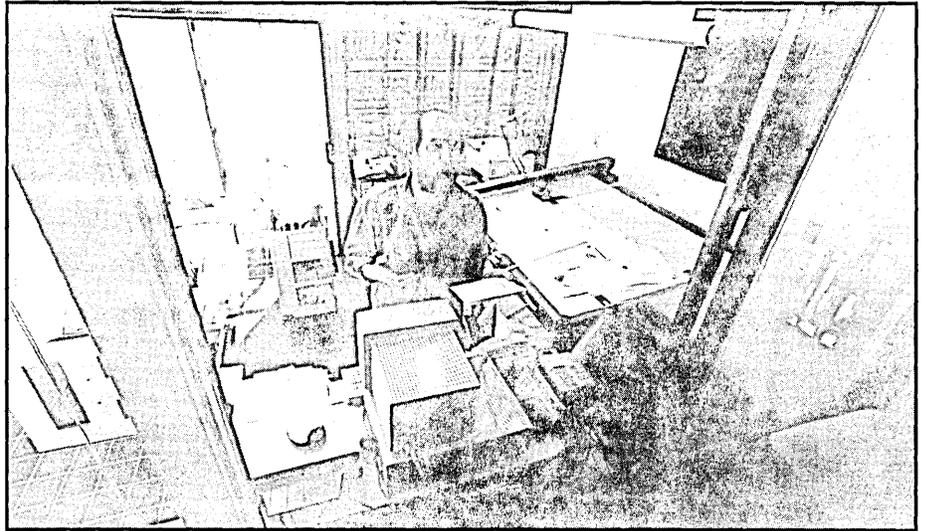


Figure 4 - Wes Wait in his Portland studio.

creative control. I'm not interested in that anymore. Now I just design the layout, proof the piece on the Laser-Writer, do final output through the (Linotronic) L300, paste it down on the board, lay a tissue over it and I'm finished.

Micro C: What about photos?

Wes Wait: I scan those and put them in the original layouts, so clients see sized halftone photos in the presentation proof I hand them.

Micro C: Of course the print shop uses the original photo when producing the final piece. For quality.

Wes Wait: Right. But the scanned image still gives them the cropping, the placement. It's a lot tighter looking than a tissue layout. The nicest thing about presentations like this is that they get approved immediately. There's no imagination required.

No more saying, "This is going to look like this, and that's going to look like that..." And then the client isn't happy when it's done. It isn't the way he imagined it.

Micro C: How many proofs do you give people?

Wes Wait: One.

Micro C: How many sample layouts?

Wes Wait: One. If I feel that the con-

cept is complex, then maybe I'll do a couple. Usually I'm dealing with just one person. It used to be that I'd paste up the presentation, then rip it all up and paste it again for the printer. With the computer, I just re-output it.

Micro C: Now that you're using the computer, how's business? Are you swamped?

Wes Wait: I don't sell the computer as a way to get projects, it's just my choice of a way to work. Actually a lot

of projects lately have come in because of the computer. Customers specifically want it. I just did a thing for Tektronix. They wanted me to develop a whole format for a communications system.

They were going to produce material on their own desktop equipment so I created the look, the shell which they would fill in each month. The City of Portland hired me a couple of months ago on a project that had to be done this way because of time constraints.

At the moment I'm very comfortably doing nothing. It doesn't bother me in the least. I was incredibly busy until I left for vacation, and everyone was incredibly upset that I was going away for a month. Now I'm back but the projects are waiting for approvals or whatever. I should have stayed away longer.

Micro C: What packages do you use?

Wes Wait: Cricket Draw, Adobe Illustrator, Pagemaker, Microsoft Word, Macpaint, Macdraw, Superpaint. Right now Illustrator is taking the place of (just about) everything else for me.

Micro C: What percentage of your work is still traditional pen and ink?

Wes Wait: Two years ago I decided that I couldn't use anything else if I was going to learn how to use the computer. That's exactly what I've done.

Micro C: You can't go into it 50-50 while you learn?

Wes Wait: You can't use it as aspirin. A lot of people have talked to me about getting a computer, and then if a project warrants it they'll use it. But it won't be cost effective because you'll always be calling up the manufacturer asking how to use the software. Unless you commit to using it, you won't master it.

Micro C: How long did it take you...

Wes Wait: I'm still working on it because I'm still getting new software. I was constantly referencing the manuals for the first year. I don't suggest you use the technology unless you really use it, otherwise it will always be a toy.

Micro C: Now that you have reasonable mastery, how has it changed the way you work? The process? What are you doing now that you wouldn't have before?

and see what happens. Or you grab the curve itself. You can just play with these curves until infinity.

This is probably the only piece of software created for the Mac which is not intuitively obvious. You have to know where to put the points to create the curves, so you have to have a good mental image of what you're going for. It's not a freehand tool. Though you can use it sort of free, it's intended to be very powerful for tracing.

You can scan a photo in and then draw on top of the photo and do a drawing in nothing flat. This piece took an hour and a half. I output it on a Linotronic at 1270 dots per inch. It's a very fast tool.

Micro C: There's a real difference between 300 dots per inch and 1270, especially when you look at characters described identically by PostScript, but

mapped. It had a totally digitized look. I called the company and they said it shouldn't. I'll tackle it again.

Micro C: Any scanner generates bit-mapped images. There's software coming out which translates bitmapped images into vectors. Of course the image will change somewhat in the translation. Hopefully it'll be compatible with something that can edit vector images. Then it could be output on a Linotype at 1200 dpi.

Wes Wait: That would be really far out. Right now I'm using scanned images as templates for drawing in Adobe Illustrator. Illustrator is so quick to work with. Tracing over a template, it only took me two hours to create this image.

Or I can take a letter, make it black, reduce it slightly and make it white, then reduce it again in black and lay all three layers over each other, creating a

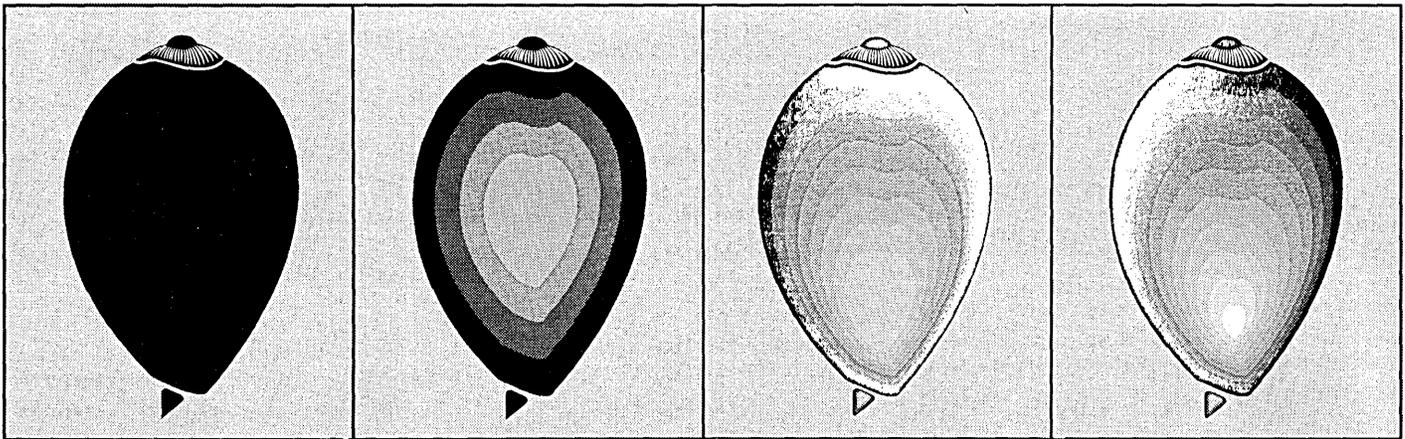


Figure 5 - Four Progressions of Wes' Computer Artwork.

Wes Wait: Mainly, the time frame to do the project is much shorter. Using the computer is second nature. I'm not spinning my wheels. I have a sense now that I could push the creativity much farther than I have.

Micro C: If I asked you for a complex piece of line art, would you do it yourself or pass it on to someone else?

Wes Wait: I don't consider myself an illustrator, and never have. But if it's something I can reduce to a symbolic level then I'll do it.

I can scan something, crop it, re-scale it, and lay it on the screen. That way I can work from an exact ratio in terms of shapes, weight of the line, everything.

Micro C: How do you work with the curves on Illustrator?

Wes Wait: It's pretty far out. Very hard for a lot of people. Rather than drawing like you're used to, you create a curve by giving reference points. If the curve isn't right, you grab the handles

the hairlines, the serifs, are so thick, so crude on the 300 dot machine. It feels heavy, loses its crispness.

Wes Wait: I did one piece that was all 300 dpi. The client wanted to save some money. I told them if they found the laser proof satisfactory, they could have it. It's just part of my creative process.

Dynagraphics printed the final piece, and they have a trick. They screen 300 dpi type. Nowadays, with the laser printer papers they're coming out with, there's no need to screen. The laser output is so crisp, screening no longer helps. I'm using Hammermill Laser Plus paper, it feels like it's clay coated.

Micro C: But usually you output on the Linotronic. How about outputting scanned images? Is 300 dpi scan resolution a problem?

Wes Wait: I haven't tried one recently. Around January (1987), I tried to output a scanned image and it came out bit-

very sharp version of the letter. Illustrator has some incredible typographical things.

Each program has its own strength, like Cricket Draw can set type in a circle easily. In Illustrator it's very hard. Generally, though, I use Illustrator.

Micro C: How did you do the acorn with the eye on top?

Wes Wait: I did the tiny moon shapes on top and made them white. Then duplicated them and duplicated them. Then stretched each individually to give it a unique character. When I reduced the first one, they filled in too much, so I took every-other one out.

This was an interesting project. I saw this ad. I didn't know why, but all of a sudden I started looking through metaphysical publications, had a feeling I was supposed to do something. When I saw the ad, I felt I knew this person and she was going to do a book.

So I went and saw her, and she's

doing a book. And I'm probably going to do a book. Her book. I've just started on the logo.

There's a trap to the computer, it's so easy to explore you can get lost in the process. It's so easy to make another change you lose half a day on a project that should have taken an hour.

I think of myself as kind of a spiritual plagiarizer. I get a strong sense of something floating through the air. I just copy it and put it down on paper and it always works. If it doesn't work for the client, it's because he's so uptight he can't see it.

Micro C: Have you tried a PC-based system?

Wes Wait: From what I hear, it's a real pain in the butt working with PC-based Pagemaker. The approach to the technology is so different between Apple and IBM.

the operating system provides a single standard. On the PC there are many standards, so you need to know which programs can share files and which can't. Many do, but it's something you need to check ahead of time.

Wes Wait: You need to do a lot more homework. Another example is Microsoft Word; any idiot could go in and run it. With WordStar, you have to learn so much stuff. IBM will switch over because the Mac has proven the viability of this format, but there's so much stuff which is still geared to the traditional person-computer relationship that it's going to take a while for OS/2 and the PC to get here. Meanwhile Apple will be light-years ahead.

Apple has kept all this stuff to itself so no one will be able to clone it, and so they'll push it that much farther ahead.

My conclusion is that desktop technology is coming along very quickly. The illustrating part of the process, however, is still learning to crawl. The software and hardware is just getting solid, and a few folk are beginning to feel their way around.

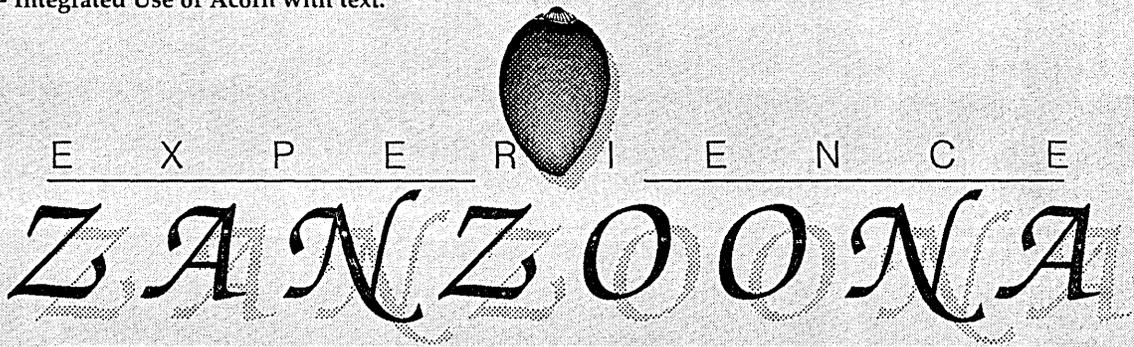
So, people are choosing to manipulate text on the computer, leaving holes for the art. Then they're letting print shops insert hand-drawn art and photographs.

Graphics are only slightly beyond the quill and ink era.

Meanwhile Back At Micro C

I'm back to writing at night because it's been bedlam around the office. We've starting collecting hardware and software so we can do our own graphics, everything. (Well, nearly everything.) No more text and holes.

Figure 6 - Integrated Use of Acorn with text.



I played with Pagemaker on the PC down at ComputerLand. There's no comparison.

Micro C: How's it different?

Wes Wait: I spent 30 seconds, after working with the Mac version for two years, and I knew it'd be a pain in the butt. I just walked away from it. It's designed to work in the IBM environment, so the interaction with the piece is entirely different. All I can tell you is my feeling about it. I just clicked a few boxes and moved a few things and, oh boy, I'm glad I have a Mac.

Micro C: There are some philosophical differences between the Mac and the PC operating systems, one of the main ones is that if you output something to a file, a text file, or graphics file, the next application may need to understand that file. There's no single type of text or graphics file on a PC.

On the Macintosh you can move information from one application to another without too much worry since

Post Script

The two interviews were interesting, partly for the kinds of graphics David and Wes were doing, but even more for the graphics they weren't doing. David was grabbing example screen displays for his books. That's fine, but it's a very limited form of graphics.

Wes showed us some very, very nice graphics pieces: pieces which would have been quite difficult to render with other media. However, he too, after two years, was still tentative, still feeling his way in this new technology.

After leaving Wes, Carol and I headed into downtown Portland to talk to one of the major printing houses. We'd heard that the printer was kind of a center for desktop publishers in the area.

They'd just had a meeting attended by the ten top designers in the area. When we asked them who was generating interesting computer graphics, they said there was only one, Wes Wait.

(Well, not so many holes.)

When we started desktopping, we had an XT, a Herc monitor, a dot matrix printer, and Ventura. After one issue, we moved over to an AT and added a Xerox full-page monitor. (I'd buy one of PC Tech's full-page monitors if I had it to do it over because it's a lot more versatile and supported by a lot more software.)

Just lately (since putting out issue #38), we've borrowed an HP scanner (the Scanjet) and purchased an NEC 890 laser printer.

The printer has 3 meg of RAM, PostScript, LED 300 dpi NEC engine, HP Laserjet+ emulation, 35 faces (scalable from 2 to 200+ points), bells, whistles, fans, a tiny little screen, serial, parallel, and Appletalk interfaces, everything. It was Christmas (the kid variety) when the printer arrived. (Okay, laugh. Tell me I can't follow my own advice and wait for the prices to really tumble. A guy can only wait so long.)

We'll still be doing the final output on a Linotronic. But we'll be proofing on something we can read! With the real typeset! You have to hold the two outputs side by side to see the difference.

We also recently acquired Pagemaker for the PC. Unlike Wes, Carol has had no trouble moving from the Mac version of Pagemaker to the PC version. But then she's flying with Ventura, too.

Creating Graphics

I've been reading everything I can find about graphics: drawing, creating, manipulating.... We have purchased and used PC Paint and PC Paintbrush but have always run into some kind of critical limitation. We've just received version 3.0 of Publisher's Paintbrush. It works particularly well for editing scanned images.

But that wasn't a surprise. All my publisher friends who knew (and knew they knew) thought Publisher's was the best PC graphics editor. It's easy to learn, and lets you edit (add and remove dots) right down to the pixel level. (Fortunately the package is easy to learn because the documentation isn't very complete.)

Though Publisher's can control the HP scanner directly, I found it much easier and faster to use HP's software then convert the files to PCX format for Publisher's.

HP's program lets you use your mouse to open a window around the image. Then it scans only the area in the window. (Very, very important.) With Publisher's, you type in the coordinates of the scanned area, a much more cumbersome process. HP's program also includes some significant editing commands, but none of them work. The manual says they're there, the menus say they're there, but HP hasn't written the code yet.

But once you've got the image in Publisher's, it's great. Just today, Carol and I were scanning the bugs which we use to distinguish the columns. (For instance, the place setting at the top of the "On Your Own" page.) Then we started cleaning up the images and outputting them on the NEC. Hey, the original, hand-drawn art isn't this good. It's not even close.

In a matter of days, Publisher's Paintbrush and our scanner have become key parts of our graphics arsenal.

However, I'm waiting for one more piece of software. Scanners and bitmapped graphics programs like Publisher's have one big limitation:

resolution and graphics file size (actually that's two big limitations, but...).

These limitations are really the result of one problem. Bitmapped graphics force you to store and manipulate greater and greater quantities of data as images get larger and resolutions get higher. Add grey scales or color, and you have multi-meg postage stamps. (Anyone insisting that an 8088 with 640K is overkill hasn't played with graphics.)

Vector Graphics

Anyway, most of the recent excitement in the graphics world has come from the Adobe Illustrator camp. Otherwise sane people are purchasing Macs just to run this hot little program. Illustrator generates vector graphics (like Autocad).

Though you can't directly manipulate scanned images, you can put them on the screen to trace over. I guess that makes you an intelligent bitmap-to-vector translator.

Though the process is a bit slow, the results are spectacular, especially when the images are output at 1200 or 2500 dpi.

Adobe is reportedly working on a PC version of Illustrator. Aldus (and no doubt, others) is also working on vector programs. Perhaps their packages will have a more intuitive feel, but infinite resolution and compact files are more than enough to recommend them.

Scanners

I'm really happy with the HP scanner and would probably purchase one immediately, except for one thing. With software, it's about \$2,000. For another \$500, there are some scanners which are also OCRs (optical character readers).

With the price so close, it's tempting to go for the extra power, though I'm not sure how worthwhile it'll be. At least 90% of the material we get by mail is in machine readable form (on the RBBS the percentage is even higher). And, I understand that OCRs are pretty fussy about what they'll read. I'll let you know when I find out.

Pagemaker

We finally have Pagemaker, thanks to a care package from Aldus. (They couldn't understand how I could write rationally about their product without trying it. Well, irrational writing is my style, so they may be disappointed.)

After half a day of putzing around, I'm certain that Pagemaker isn't the

product for laying out *Micro C* articles. But, I'm equally certain that it will be well-used. Take the Micro Ads, for instance. Carol's already doing those in Pagemaker. She just makes the box, types in the text, sets the type face and style, slides things around, pushes here, nudges there. Very smooth. Very obvious.

If you're doing a newsletter, it would be a tossup. An ad? Use Pagemaker. A magazine? Use both Ventura and Pagemaker. (Need money to purchase all this fancy stuff? I'm still working on that. Meanwhile, the care packages help.)

Ventura Fixes

While attending the Seybold Conference this fall I connected up with the Ventura folks. They said they'd just sent Xerox a fix for my screen problems (unreadable 9-point text and a cursor that doesn't know its place).

I called Xerox and after mere minutes got through (call between 3 and 4 p.m., Texas time, you'll get through, too). They sent me the fix (four disks).

It upgraded version 1.1 to version 1.11 and included a three-page list of bugs it fixes. The screen and cursor problems weren't on the list, but I crossed my fingers and did the installation. It didn't help.

Then I got a great letter from Steve Terry with the fix. A great fix! It took five minutes total! See his letter in this issue's Techtips.



**Rafting
Barbeque
Computer tech talk
SOG VII
July 14 - July 16
Bend, Oregon
Leave city life behind!**

SOFTWARE ENGINEERING COMES OF AGE.

ANNOUNCING LOGITECH MODULA-2 VERSION 3.0

Modula-2 is the language of choice for modern software engineering, and LOGITECH Modula-2 is the most powerful implementation available for the PC. The right language and the right tools have come together in one superior product. Whether you're working on a small program or a complex project, with LOGITECH Modula-2 Version 3.0 you can write more reliable, maintainable, better documented code in a fraction of the time at a fraction of the cost.

**FREE TURBO PASCAL
TO LOGITECH MODULA-2
TRANSLATOR**

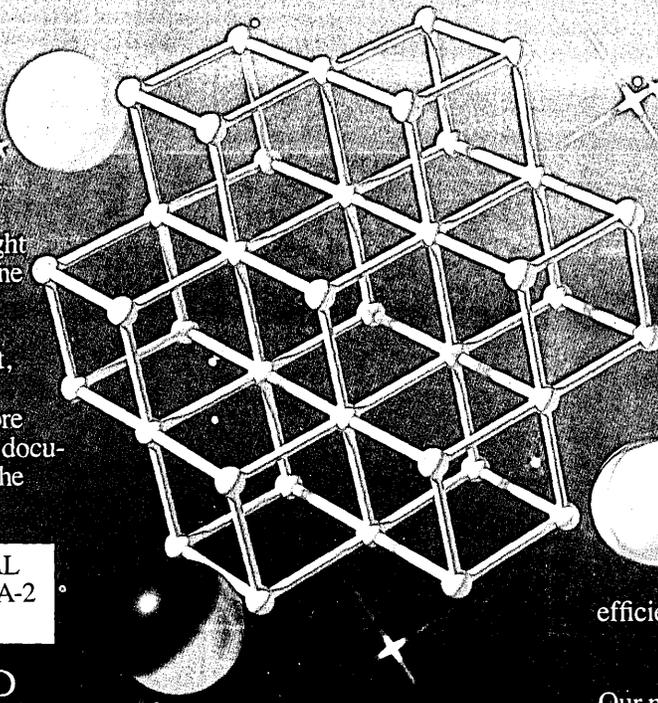
NEW, IMPROVED DEBUGGERS

Time gained with a fast compiler can be lost at debug time without the right debugging tools. With the powerful Logitech Modula-2 Debuggers you can debug your code *fast*, and dramatically improve your overall project throughput. The Post Mortem Debugger analyzes the status of a program after it has terminated while the dynamic, Run Time Debugger monitors the execution of a program with user-defined break points. With their new, mouse based, multiple-window user interface these powerful debugging tools are a pleasure to use.



NEW, INTELLIGENT LINKER

Links only those routines from a particular module that you need, so you eliminate unreferenced routines and produce smaller, more compact executable files.



NEW, IMPROVED COMPILER

Faster and more flexible. Now its DOS linker compatible object files (.OBJ) can be linked with existing libraries in C, PASCAL, FORTRAN and ASSEMBLER—so you can build on previous development and put the power of LOGITECH Modula-2 to work for you right now. Fully supports Wirth's latest language definition, including LONGINT and LONGSET, which provides large set support including SET, of CHAR. Provides optimization for tighter, more efficient code generation.

NEW EDITOR

Our new, mouse based editor is fully integrated, easy to learn, fast and easy to use, and very customizable. Its multiple, overlapping windows and color support make it easy to manage parts of one file or several files on the screen at one time. You'll love using it—with or without a mouse.

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

To place an order call toll-free:
800-231-7717
In California:
800-552-8885

- LOGITECH Modula-2 V. 3.0 Compiler Pack **\$99**
Compiler in overlay and fully linked form, Linkable Library, Post Mortem Debugger, Point Editor
- LOGITECH Modula-2 V. 3.0 Toolkit **\$169**
Library sources, Linker, Run Time Debugger, MAKE, Decoder, Version, XRef, Formatter
- LOGITECH Modula-2 V. 3.0 Development System **\$249**
Compiler Pack plus Toolkit
- Turbo Pascal to Modula-2 Translator **FREE**
With Compiler Pack or Development System
- Window Package **\$49**
Build true windowing into your Modula-2 code.
- Upgrade Package
Call LOGITECH for information or to receive an order form.

Add \$6.50 for shipping and handling. California residents add applicable sales tax. Prices valid in U.S. only. Total Enclosed \$ _____

VISA MasterCard Check Enclosed

Card Number _____ Expiration Date _____

Signature _____

Name _____

Address _____

City _____ State _____

Zip _____ Phone _____



LOGITECH, Inc.
6505 Kaiser Drive, Fremont, CA 94555
Tel: 415-795-8500

In Europe: LOGITECH, Switzerland
Tel: 41-21-87-9656 Telex 458 217 Tech Ch

In the United Kingdom: LOGITECH, U.K.
Tel: 44908-368071 Fax: 44908-71751

Designing A PC Workstation Board Around The TMS34010

Graphics Never Looked Better (Or Faster)

I was talking to an outfit which designed and manufactured graphics controllers, and just offhand I asked if they were using TI's new 34010. "Nope." Would they be? "Nope." What kind of graphics logic were they using? "PALs." How would they adapt to new graphics interfaces? "New PALs."

Fortunately, you don't have to make new PALs (friends are hard enough to come by as it is) to meet new standards if your video controller is smart, 34010 smart.

Most of you are information users. You use computers, and computers manage information. They massage it, display it, print it, reduce it, expand it, and occasionally lose it.

Fewer than ten years ago, most of us were content to store information on floppy disks and view it on a Teletype, or (if we were especially lucky), on a CRT terminal with an 80 character by 24 line display. In those days 64K bytes of memory was a luxury.

Needless to say, the picture has changed. Hard drives are cheap. Now 16 megabytes of solid state memory fit on a single memory expansion board. It's a sign, friends, of the Information Age and represents great advances in technology. Yet the tools to display all this information have been lagging. Until very recently.

To expand the amount of displayable information, while simultaneously condensing it into a form which can be more easily assimilated by the computer user, programmers have turned to graphics.

After all, isn't a picture worth a thousand words? Certainly many companies have made fortunes selling programs which display data as pies and bars.

Currently the amount of information

which we can display is limited by resolution, colors, and speed.

Resolution is the number of picture elements, or pixels, displayed on the CRT and is usually expressed in pixels, horizontally and vertically. The number of colors we can display is a function of how deep the memory is or how many bits there are per pixel.

Speed is a factor because, as the amount of information increases so does the processing time. As the number of pixels increases so does the time it takes the CPU to draw them.

Existing PC Graphics Controllers

There are currently three basic types of graphics controllers used on most PCs: MDA/Hercules, CGA, and EGA/VGA. These boards have much in common.

All are memory mapped; that is, in order for the CPU to display something on the CRT, it simply writes to the appropriate place in memory.

When these boards are in text mode, the data written shows up as characters. When they're in graphics mode, the data shows up as a series of pixels on the CRT. Simple enough.

This works fine until the resolution increases. Even with an EGA system most users will see a noticeable slowdown in character display operations. Try graphics mode. Run the following short test if you have a color monitor and feel the need for speed:

```
MODE CO40<cr>  
DIR<cr>
```

How slowly the display scrolls! Yet, this mode (CO40) is only 320 by 200 pixels, mediocre resolution! Can you imagine 1024 by 800 pixels, where each pixel is eight bits instead of two?

Unfortunately, this slow speed has led the programmers of most good

graphics programs to skip the BIOS and operating system so they can access the hardware directly.

All of these boards have a 6845 CRT controller chip or 6845 emulation for performing the display refresh and cursor functions.

Workstation Graphics

Workstation manufacturers have taken a different approach by making the workstation graphics controller intelligent with the addition of a microprocessor dedicated to graphics tasks. Typically this has been a 68000, and lately has been enhanced with the addition of special hardware to speed the graphics operations.

There are several advantages to using a second processor for graphics operations. First, the graphics processor can do some of the image calculations for the system processor.

Second, and perhaps more important, the graphics interface for applications programs can be made device-independent. Thus, workstation vendors have been able to offer several graphics options (with a range of resolutions, black and white or color...) without affecting application programs.

New Generation Graphics Controllers

Two types of graphics controllers have been developed by semiconductor companies during the last two years: display list processors and graphics processors.

The display list processor chips transform lists of commands into raster data for the display. These can be both character and graphics commands. Display list processor manufacturers include AMD, Hitachi, Intel, NEC, and National Semiconductor. Some of these devices even include windows and zoom-in hardware. However, in order to be most effective in a high-level graphics

task, the display list processors must have a microprocessor controlling them (to interpret the high-level commands).

The Graphics CPU

Combining the graphics drawing capabilities of the display list processor with a high performance microprocessor on a single chip is an approach taken by Texas Instruments and National Semiconductor.

The Texas Instruments TMS34010 is a 5-6 million instruction per second 32-bit CPU, optimized for graphics operations. For a basic overview of this part see the April-May 1987 issue of *Micro C* (#35, p. 86, "Changing The Picture").

Designing A Graphics Board

I think a great video board for a PC is one with all the capabilities of a workstation but without the cost. That's why we chose the TMS34010 when we faced designing a graphics card, which not only had to support high resolution

without flinching but also emulate common PC video cards.

Let's dig into the details (see Figure 1 for a block diagram of the system).

The 34010 supports two types of memory. One type holds programs and fonts, the other holds data for display. High-speed shift registers pump two bits per pixel out to the monitor at up to 100 million pixels per second.

The TMS34010 is housed in a 68-pin plastic chip carrier. This package allows the 34010 to be either surface mounted or socketed. The package is identical to that used by Intel for its 80186 and 80188 CPUs, with one major difference. Intel starts counting pins at the corner of the package while Texas Instruments and the rest of the world count pins from the center of the package. Figure 2 shows the TMS34010 pin layout. The package is approximately one inch square.

Memory Interface

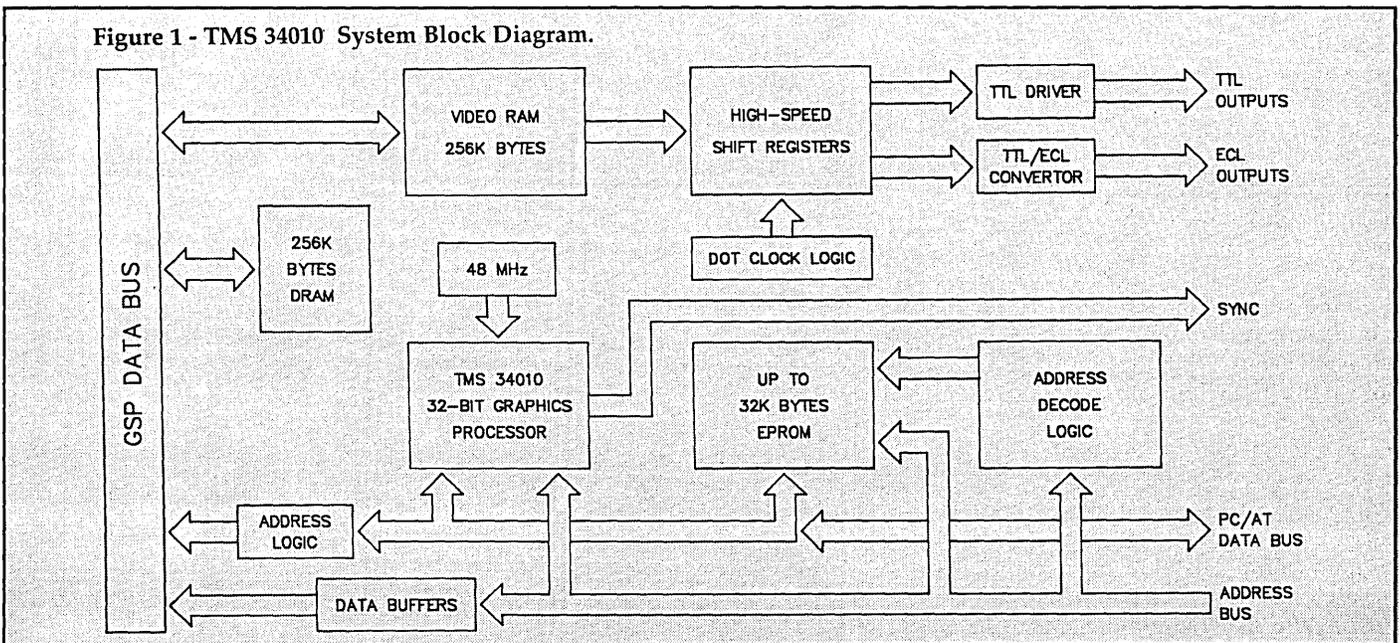
The TMS34010 has a multiplexed address and data bus. The address bus is designed to interface to most dynamic type memories which require a multiplexed address bus. So the designers at TI have triply multiplexed the bus, first providing the memory row address, then the column address, and finally the data (see Figure 3). The column address must be latched while the bus is used for data.

A simple transparent latch such as a 74LS373 does the job just fine.

The TMS34010 controls this latch with the signal LAL. For decoding banks of memory, the upper address must be latched and decoded during the first part of the memory access cycle. This is the row address. Again, a simple transparent latch works just fine, this time being controlled by the 34010's RAS signal.

In designs having multiple banks of memory, the RAS or CAS signal is

Figure 1 - TMS 34010 System Block Diagram.



directed to the active bank. This is done by sending the latched upper addresses, as well as the TMS34010 RAS or CAS signal, to a decoder. In this case, the decoder might be a PAL (Programmable Array Logic) or PROM.

Special Memory Cycles

The TMS34010 performs two types of refresh cycles: memory refresh and screen refresh.

The first is quite simple and well understood. Since each and every row of the dynamic RAM must be refreshed at least once every four milliseconds, the chip refreshes RAM downright regularly.

During a refresh cycle the designer must be sure that every bank of memory receives the refresh address and a RAS signal. Thus, if the bank selection decoder is selecting banks of memory using the RAS signal, all banks must receive RAS in this special case. The TMS34010 announces refresh on its LAD15 pin during the start of the memory cycle. This signal must also be latched.

Screen Refresh Cycles

The screen refresh cycle requires a little background. The TMS34010 must use a special type of dynamic RAM for its screen buffer. The RAM is often called Video RAM (VRAM) or multi-port dynamic RAM. (A hint of what makes this RAM so special.)

The multi-port dynamic RAM has two ports for accessing data. One port is nearly identical to a standard DRAM interface, having address pins, a RAS pin, a CAS pin and an OE (output enable) pin. This is the port used by the TMS34010 for writing or reading screen memory. The second port is a serial access port, where data can be shifted in or out at rates up to 25 MHz (for a 120 nsec part). This is the side of the DRAM closest to the display.

During the screen refresh cycle, data in the multi-port dynamic RAM is transferred, one row at a time, to the serial access port (to be displayed). Each row is 256 bits. The multi-port dynamic RAM assumes it will be this type of memory operation if its OE pin is active at the start of the memory cycle. The 34010's TR/QE pin is connected directly to the RAM's OE pins.

Meanwhile, all video RAMs must receive RAS and CAS during a screen refresh cycle. The video controller provides a signal on LAD14 during the column address time of a screen refresh

Figure 2-TMS 34010 Pinout.

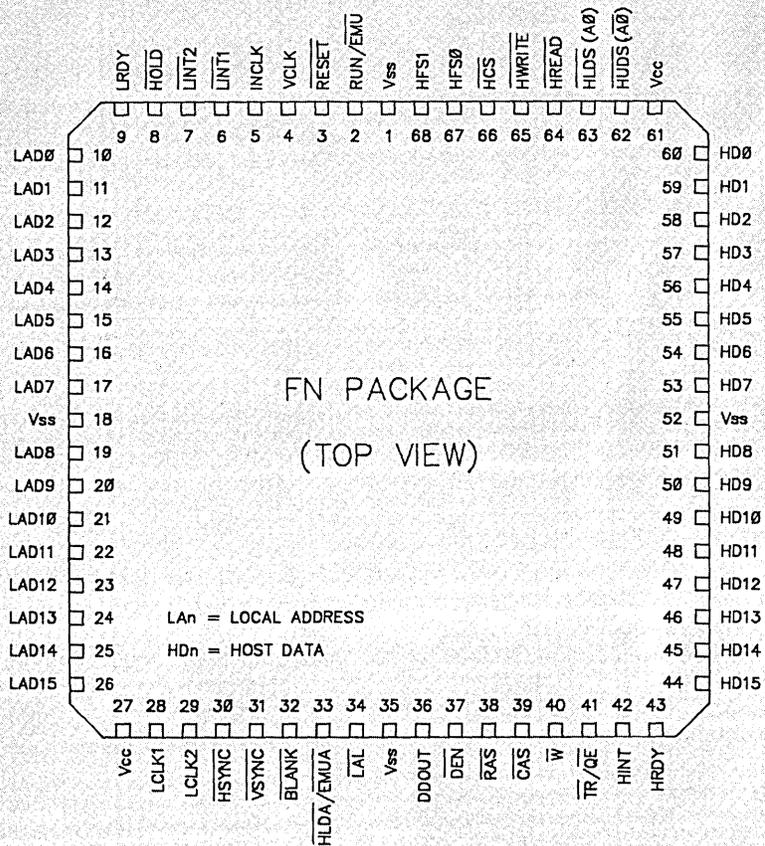
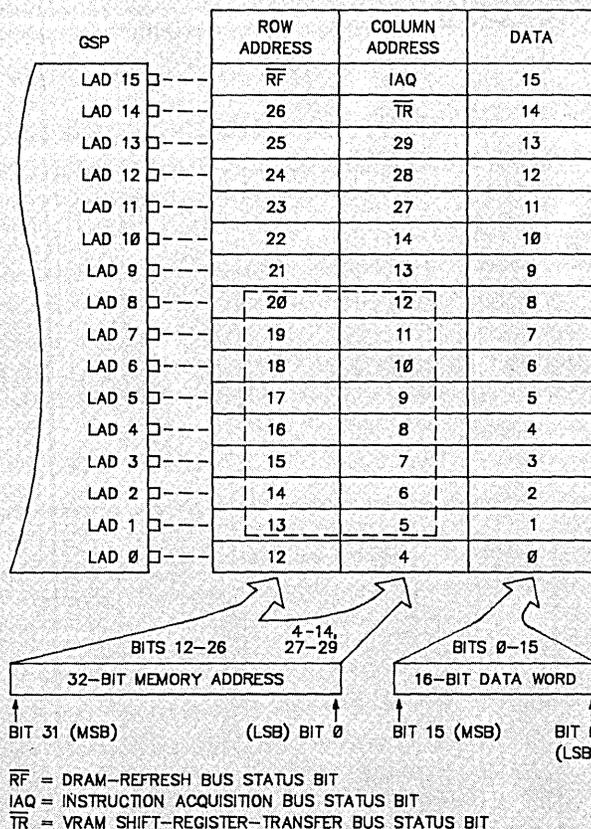


Figure 3-Triple Multiplexing of Addresses and Data.



cycle.

I/O Interface

The host processor (the PC) communicates with the TMS34010 through a host data bus and several other control pins. This interface allows the host to access all of the 34010's memory as well as its I/O registers.

The host data bus is 16 bits wide but can be used with an eight bit bus quite simply. Two input signals, UDS and LDS, select the upper and lower halves of the bus.

By connecting the upper half of the host data bus (HD8-HD15) to the lower half of the host data bus (HD0-HD7), and by connecting LDS to the host's lowest address pin (A0) and UDS to inverted A0, the eight bit interface is complete. The TMS34010 also has conventional chip select, read and write input pins. Two address inputs select any one of the four internal registers (see Figure 4).

CRT Interface

I mentioned earlier that the screen buffer is a special type of memory. This memory has a second access port through which data may be shifted out at up to 25 MHz. These shift registers are 256 bits long.

Figure 4 - Register Select Logic.

HFS1	HFS0	Selected Register
0	0	HSTADRL
0	1	HSTADRH
1	0	HSTDATA
1	1	HSTCTL

Since a screen refresh cycle can occur only during the horizontal blanking time (once per scan line), it's obvious that something else is needed to accomplish state-of-the-art video output with over 1000 pixels horizontally and data rates of over 100 Mpixels per second. The solution is to parallel several multi-port dynamic RAMs with their outputs going into a high-speed shift register. The register then sends the data to the display.

On our 34010 board we set the rate at 75 MHz, though we tested the logic at up to 110 MHz. We used 74F393 high-

speed shift registers rated at 125 MHz.

At these speeds timing is very critical, and you soon learn that every nanosecond counts. The shift registers must be reloaded the instant they shift out their last pixel. A high-speed counter keeps track of load cycle timing and clocks in the data from the multi-port dynamic RAMs.

We provide two bits per pixel to the CRT, giving four levels of pixel intensity. We simply duplicated the high-speed shift registers to output the second data bit.

On the TMS34010, a pixel (in our case, two bits) is placed in adjacent memory locations. This is in contrast to many other systems where a pixel's bits are stored in several planes of memory. The adjacent bit organization provides much simpler control of pixel color (or intensity) than the multiple plane organization.

ECL Outputs

In addition to having TTL outputs to a monitor, the board also has ECL outputs. The advantages of the ECL outputs are higher speed and fewer problems with noise.

Both true and complemented ECL data is sent to the CRT. Since both signals run in close proximity to each other through a cable to the CRT, the effect is the radiation from one line cancels the radiation from the other. This means your radio is less likely to go to lunch and the FCC stays happy.

The disadvantage with ECL is that it's really a power hog, consuming large amounts of current from the PC's wimpy -5V supply. We use only one ECL chip, a 10124 TTL to ECL converter, to keep power consumption to a tolerable level.

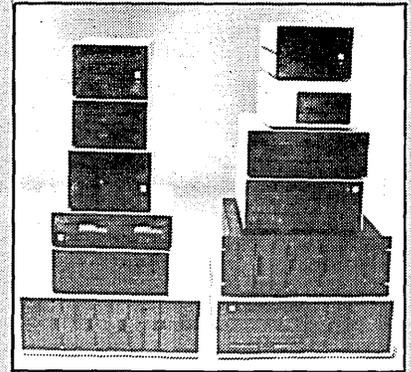
CRT Timing Registers

Most CRTs need horizontal and vertical sync signals to time the start of the beam across the face of the CRT. The controller provides these signals and allows the timing of them to be programmed (via eight internal registers) to match the characteristics of most CRTs. To the programmer, these registers appear in TMS34010 memory space starting at address C000000H (see Figure 5).

PC Specific Support Hardware

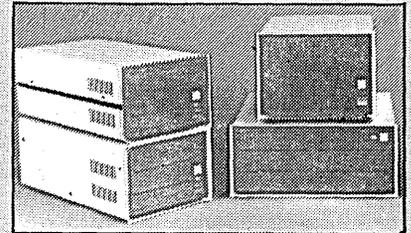
In order for a video board to be compatible with most IBM PC applications, the PC must see display memory at either B0000H (for MDA) or B8000H (for

ENCLOSURES & POWER SUPPLIES



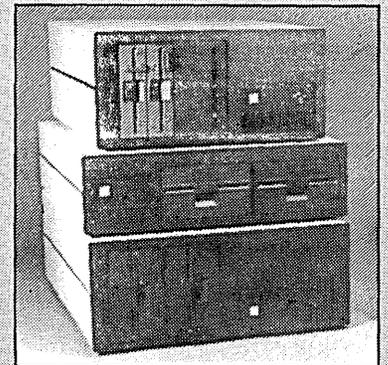
for FLOPPY WINCHESTER TAPE DRIVES

from \$100



for SINGLE BOARD COMPUTERS

from \$125



for IBM & S-100 SYSTEMS

from \$200

*1 piece, prices lower in quantity.
(Disk drives not included)

Call or write for free catalog

INTEGRAND
RESEARCH CORP.

8620 Roosevelt Ave. • Visalia, CA 93291

209/651-1203

TELEX 5106012830 (INTEGRAND UD)

IBM™ International Business Machines

Reader Service Number 22

Announcing

ImageWise™

Serial Digital Imaging System



ImageWise functions as a standalone video digitizer or a complete tele-imaging and video capture system.

ImageWise's serially bit mapped digitized pictures are universally compatible with any computer or modem. It is ideally suited for CAD/CAM, Desktop Publishing, Tele-Imaging, and Security.

System Specifications ■

- *NOT bus dependent
- *Captures an image in 1/60th second
- *Accepts any B/W or color NTSC video
- *Resolution of transmitted image is 256x244x64 gray scale. Resolution selectable: High - Medium - Low



unretouched photos

- *Video Input: 75 Ohm, 1V peak-to-peak.
- *Video Output: 75 Ohm, NTSC, 1.5V peak-to-peak.
- *Serial Input/Output: RS-232 - 8 bit, one stop bit, no parity - 300 bps to 57.6K bps selectable data rate - Xon/Xoff handshaking - switch selectable data compression (on/off).
- *Modem compatible: Functions as a video or a remote surveillance camera.
- *Video processing: PC/MS-DOS picture upload/download and conversion utilities to popular Paint & Desktop Publishing programs.

Optional PC Utilities Disk converts ImageWise files for use with popular Desktop and Paint programs.

DT01 Digitizer/Transmitter \$349.00

DR01 Receiver/Display \$349.00

To order call

1-800-635-3355

TELEX: 643331

For technical information call
1-(203)-871-6170

MICROMINT, INC.
4 Park St., Vernon, CT 06066



Reader Service Number 23

Figure 5 - CRT Timing Registers

REGISTER	ADDRESS		DESCRIPTION
DPYADR	C000 01E0	Display Address	Counts the number of scan lines output between successive screen refresh cycles and contains the source of the row and column addresses output during a screen refresh cycle: LNCNT (bits 0-1) - Scan line counter SRFADR (bits 2-15) - Screen refresh address
DPYCTL	C000 0080	Display Control	Contains several parameters that control video timing signals: HSD (bit 0) - Horizontal sync direction DUDATE (bits 2-9) - Display address update ORG (bit 10) - Screen origin select SRT (bit 11) - Shift register xfer enable SRE (bit 12) - Screen refresh enable DXV (bit 13) - Disable external video NIL (bit 14) - Noninterlaced video enable ENV (bit 15) - Enable video Bit 1 is reserved.
DPYINT	C000 00A0	Display Interrupt	Specifies next scan line that will cause a display interrupt request.
DPYSTRT	C000 0090	Display Start Address	Provides ctrl of the automatic mem-to-shift-register cycles necessary to refr a screen: LCSTRT (bits 0-1) - Specifies number of scan lines displayed between screen refreshes. SRSTRT (bits 2-15) - Starting scrn-refr. addr
DPYTAP	C000 01B0	Disp Tap Pt. Addr	Contains a VRAM tap point addr output during shift register transfer cycles.
HCOUNT	C000 01C0	Horiz. Count	Counts the number of VCLK periods per horiz scan line.
HEBLNK	C000 0010	Horiz. End Blank	Designates the endpoint for horiz blanking.
HESYNC	C000 0000	Horiz. End Sync	Specifies the endpoint of the horiz sync interval.
HSBLNK	C000 0020	Horiz. Strt Blnk	Specifies the starting point of the horiz blanking interval.
HTOTAL	C000 0030	Horiz. Total	Specifies the total number of VCLK periods per horizontal scan line.
VCOUNT	C000 01D0	Vertical Count	Counts the horiz scan lines in video display.
VEBLNK	C000 0050	Vertical End Blank	Specifies the endpoint of the vert blanking interval.
VESYNC	C000 0040	Vertical End Sync	Specifies the endpoint of the vert sync pulse.
VSBLNK	C000 0060	Vertical Strt Blnk	Specifies the starting point of the vert blanking interval.
VTOTAL	C000 0070	Vertical Total	Specifies the value of VCOUNT at which the vertical sync pulse begins.

CGA), as well as a 6845 type video controller at a certain I/O address, depending on video type.

We chose to do neither on the PC Tech card, yet we remain compatible through clever deception. (Editor's note: Now I suppose you'll be running for higher office.)

When graphics information is written to the IBM CGA screen, it's displayed

from left to right starting with the most significant bit to the least significant bit. This is opposite from the way the TMS34010 displays data. To accommodate this difference, the host CPU reverses the data when it reads from or writes to TMS34010 memory.

This is easy to do since the two data buses are separated by buffers. This little trick gave Earl (Hinrichs) fits when writ-

ing the TMS34010 code for emulating a text display since the ASCII codes are also reversed.

Emulating the 6845 register set is a bit more difficult since the 6845 has an index register which points to one of many data registers. We accommodated this by providing a hardware index register to override the lower address bits during a read or write to a 6845 data register. The data is actually in a special region of TMS34010 dynamic memory. This trick fools even the best programs.

We also generate a display enable signal which changes with each horizontal sync. Many application programs check this signal to see if they can write to the display memory without causing screen disruption.

And we added software control of video emulation. (We just write to a 34010 register to select emulations.)

Emulation modes include-

MDA compatible - 80 characters by 25 lines

Enhanced MDA - 80 characters by 66 lines

CGA compatible - 80 characters by 25 lines

- 320 by 200 graphics
- 640 by 200 graphics

Enhanced CGA - 80 characters by 66 lines

- 320 by 200 graphics
- 640 by 200 graphics
- 736 by 1024 graphics

Wrap Up

Earl will detail several of these in his article next issue. I would just like to add that the CGA emulation modes are good enough to fly Flight Simulator and Jet. The high resolution 736 by 1024 graphics mode is compatible with another popular video board, allowing most any desktop publishing application or other graphics application to work without modification.

In the future, emulating other video boards will become less important as higher-level graphics standards are adopted on the PC. This may be a standard such as Microsoft Windows or Presentation Manager, or possibly CGI,

DGIS, Metawindows or HALO. Whatever the standard, the TMS34010 will really shine by off-loading most of the graphics tasks from the host CPU.

What else does the future hold? Certainly higher resolutions. I've already modified a board to provide 1600 by 1024 resolution at 1 bit per pixel. The data rate has been run as high as 160 million pixels per second in this mode. And 2048 by 1500 resolution at 200 million pixels per second isn't far away.

Texas Instruments has let it be known that a successor to the TMS34010 is on its way. The TMS34020 promises a full 32-bit data path and probably some new instructions. Higher speeds and a larger cache are not unlikely, either.

Personally, I'd like to see floating point capability, 10-15 MIPS sustained performance, parallel processor support, etc., etc...

Editor's note: Dean and Earl are marketing their 34010 graphics board through their company, PC Tech, 904 North 6th St., P.O. Box 128, Lake City, MN 55041.



Come to
SOG VII!
July 14 - July 16
Bend, Oregon

And Bring Your
Friends!

OMNIVID™

WINDOWS! SPRITES! ROW TABLE SCREENS!
FOR THE '83 KAYPRO II OR 4, XEROX 820-1, AND BIG BOARD-1

THE OMNIVID DEVICE DRIVERS

Blazing fast assembly language device drivers make impressive screens quick and easy! No complicated code to write. Just send simple escape sequences to your Console Output routine. Multiple screens, each with its own windows and sprites, may be held in memory for instant use! OMNIVID™ makes it simple. And, with FLEXUTIL.COM (provided) it's quick and easy to create an executable .COM file which loads your selection of the OMNIVID™ device drivers in protected upper TPA, above the BIOS, or at any absolute address. Information on how to write and include your own device drivers is also provided!

Easy to use with most High Level Languages. Toolboxes provided for Sbasic® and Turbo Pascal®. TURTLE.BOX (also included for use with Turbo Pascal®) brings TURTLE GRAPHICS ability.

ALSO INCLUDED

- * A device driver which supports a MICROSOFT compatible serial (RS-232) mouse. (Can be configured to support the popular LOGITECH C7 serial mouse.) Now you can write mouse driven programs with pop-ups and pull-downs for your 286 computer!!
- * A device driver for dumping the screen to your printer.
- * SLOTS.COM - A realistic slot machine game demonstrates ROW TABLE DRIVEN SCREENS.
- * JAWS - A game to demonstrate the ability to animate using the OMNIVID™ device drivers.

Available now for the '83 Kaypro II, 4, Xerox 820-1, Big Board-1. Soon for the '84 series Kaypros and PC compatibles.

ONLY 49.95 (includes shipping)
KY residents add 5% Sales Tax
VISA, MC, or COD accepted

Call or write for info. Or order documentation on disk for only \$8.00 (includes shipping).

ALSO AVAILABLE

A GRAPHICS CHARACTER ROM for your 83 Kaypro II or 4, Xerox 820-1, or Big Board-1 supplements the OMNIVID™ device driver package with a thin-line and pixel graphic (160 x 48) character set. As well as providing better looking window borders and an improved alphanumeric character set, the pixel graphic character set is highly desirable for use with TURTLE.BOX (above). ROM PRICE 19.95 (includes shipping).

TEL (606) 325-3736 9AM-6PM EST

FLEXISOFT • 3987 VALLEY VIEW DRIVE • ASHLAND, KY 41101

Reader Service Number 25

A Hercules Primer

Black Sheep Graphics

Need to increase your New Year's resolution? Here's nearly everything you ever wanted to know about writing graphics routines for the Hercules card.

Probably should call this monochrome sheep graphics. The Hercules graphics card stands out as the only "nonstandard" graphics card to become successful in the PC market.

CGA and EGA cards both have the support of ROM BIOS routines to control the graphics mode and set individual pixels. Not so with the Hercules card. Pascal graphics libraries do exist for the Herc. But in the switch to Turbo C for coding fractal programs, I lost them.

I can't do without the Herc's 720 X 348 resolution, it's still the best mix of low cost and high resolution going. So let's build a library of functions to tame the Hercules card.

Is It There?

First things first. Is there a Hercules card installed in the system? I guarantee some weird results if you use these functions to draw in a non-Hercules system.

Interrupt 11h checks for the presence of a monochrome card. If the AX register has bits four and five set on return, then some kind of monochrome card is out there. It could, however, be the lowly IBM Monochrome Display Adapter (MDA).

To further discriminate between monochrome cards, we'll have to dig a little deeper. At the heart of most PC video boards lies Motorola's 6845 CRT controller. Two I/O ports to the Hercules card give access to the 6845's registers, while another two ports provide status information and mode control.

On the Hercules card, bit seven of the status port contains information on vertical retrace. During horizontal scans, bit seven remains high. After the bottom scan line has finished, the electron beam returns to the top scan line. During this return, or retrace, the Hercules card resets bit seven to zero.

The MDA always leaves bit 7 low, so during retrace the Hercules looks just like the MDA. `herc_here` samples bit 7 long enough to be sure vertical retrace has finished. Then, if bit 7 ever goes high, it's a Herc. (See Figure 1 for all graphics functions.)

Why use 800h for the number of times to sample bit 7? Strictly because it works on each of the mongrel machines in the Micro C office. If you want to be absolutely sure of correctly identifying the video card, use 1000h.

Configuration

The Hercules card has two video pages, each 32 Kbytes. The first, page 0, starts at address 0xb0000. This is the location of the monochrome text screen. Page 1 starts at 0xb8000. The configuration port controls access to the 6845's various modes.

You can set up the 6845 three different ways. With bits zero and one at zero, the card displays only text. Setting bit zero to one allows the graphics mode. Finally, a zero in bit one masks the use of graphics page 1. Or, a one in bit one enables page 1.

Note that, since we can mask off page 1, both the Hercules card and a CGA card could be installed in the same system. Both page 1 and the CGA card start at 0xb8000. But with page 1 masked, there's no contention.

Graphics Mode

Before setting any graphics mode, send a byte to the configuration port to allow graphics on one or both of the

video pages. (See `set_graphics` in Figure 1.) Next, turn off the screen so the user won't see any nastiness during the mode change. This involves the mode control port.

Only four bits of the mode control port have any effect. Bit one sets text mode (zero) or graphics mode (one). Bit three either blanks the screen (zero) or activates it (one). Bit five disables text blink (zero) or enables it (one). And bit seven determines the active display—zero for page 0 and one for page 1.

So we'll send 00000010b to the mode control port, turning off the screen and setting graphics mode. Now program the gaggle of registers in the 6845. Of its 18 registers, only the first 12 have any importance in setting video modes. Figure 2 gives a description of each of them.

The final lines of the `set_graphics` function clear both pages and turn on the video display of page 0. More about `clear_gr_scr` later. For now, the important aspect of this function is that it sends 00001010b to the mode control port, displaying page 0.

Text Mode

`set_text` looks a lot like `set_graphics`. A zero to the configuration port locks out the graphics modes. Another zero to the mode control port sets text mode and turns off the screen. A new slough of parameters goes out to the 6845 registers. Then we clear the screen and send 00101000b to the mode control port. This enables the text blink and turns on the screen.

I've read about timing problems when programming the 6845 from high level languages. My Hercules clone's documentation warns against the practice due to the undefined frequencies output during a mode change. But I've been using Turbo C to control the 6845 for some time now with no smoke. I've

even successfully programmed the 6845 with Turbo Pascal. If you're worried, you can always rewrite the mode changes in assembler.

What's The Point?

In order to do anything with graphics, you'll need to be able to perform the most fundamental graphics function—draw a point. Our display is bit-mapped, with each pixel on the screen corresponding to a bit in memory. So setting the appropriate bit turns on a pixel. The only trick lies in finding the memory address of a given pixel.

We have to deal with a factor-of-four interleave in memory addressing of rows. The first (0th) row starts at offset 0 within the page. Row one starts at offset 2000h, row two at 4000h, row three at 6000h, and row four at 90d (90 bytes/row * 8 bits/byte = 720 bits/row).

So where's the point (X, Y)? (Y MOD 4) tells us which quarter of the page's memory it's in. 2000h * (Y MOD 4) gives the offset of that quarter page. The line containing (X, Y) is (Y DIV 4) lines down within the quarter page. Chalk up another 90 * (Y DIV 4) bytes to add to the offset. Within that row there will be an offset of (X DIV 8) bytes to the byte containing (X, Y). And the bit we're after is bit (7 - (X MOD 8)). Messy indeed. (See `draw_point` in Figure 1.)

To draw the point, set up a mask byte by shifting a one into the corresponding bit. Then, ORing the memory location with the mask turns on the pixel. ANDing with the one's complement of the mask turns the pixel off.

Line Drawing

Lines are just collections of points, and we already have the `draw_point` function. But how do you decide which points should be included in the line? I

lifted Bresenham's algorithm from *Computer Graphics—A Programming Approach*, a graphics text by Steven Harrington.

Bresenham's algorithm works with integer arguments and uses only integer math. It also avoids division and multiplication so it zips right along. Very briefly, this algorithm breaks the problem into two major cases—steep lines and shallow lines.

In the shallow line case, it draws one endpoint, then steps through the columns toward the other endpoint. At each column, it tests to see if the line has moved into a new row. If so, the row number gets incremented (for positive slope lines) or decremented (for negative slopes). The steep line case reverses the situation. We step through the rows and test for a change of column.

I might mention that although the variable name `dX` stands for "Delta X," this code is guaranteed not to crash.

Miscellaneous Functions

`init` takes care of initializing the two global variables, `page_drawn` and `page_viewed`. `page_drawn` holds the segment address of the active drawing page and `page_viewed` holds the segment address of the displayed page. `set_page_viewed` and `set_page_drawn` alter these two variables. `set_page_viewed` also programs the mode port to display the specified page. You can use these functions to write to one page while viewing the other.

No mysteries in `clear_gr_scr`. I turn off the screen when clearing the displayed page, but that's just a matter of taste. It's not necessary. `clear_txt_scr` needs to fill all attribute bytes with sevens—the normal attribute.

Save any screen to disk using `save_screen`. It copies the specified page directly to disk under the file name passed to the function. I fooled with this

one for quite a while. The pointer to video memory just wasn't working out. Tony Ozrelic put me on the right track, and while the code I came up with ain't portable, it does work.

`get_screen` restores a saved screen. If you omit the `screen_off` in `get_screen`, you'll get to see a graphic demonstration of the video memory interleave. To save a screen in text mode, use `save_screen` and pass it `page0` as a parameter.

Direct Screen Writes

Sure it's not graphics, but foregoing the video services of interrupt 10h is what this article is all about. And you can very easily write characters directly to video memory with a Hercules card.

Characters live in the even bytes of video memory while their attributes (blinking, underline, etc.) are found in the succeeding odd locations. Each character/attribute pair takes up two bytes and each row of 80 characters takes up 160 bytes. So it's easy to find the memory offset of a particular screen location.

$$\text{offset} = 160 * \text{row} + 2 * \text{column}$$

`scr_write` just pokes away at memory until it reaches the end of the string being written. No sweat. If you use this function on a CGA system, you'll need to consider the following.

Video memory is dual ported. That means the processor can write to memory at the same time that the video card reads it for a trace on the screen. On color systems this creates snow during direct screen writes. Inexcusable. A program has only to poll the vertical retrace bit of the status port and restrict screen writes to the retrace period. No more snow.

The Herc doesn't care, though. It

must be doing arbitration on its own because you can alter video memory any old time you please. Of course, this applies to the graphics mode as well as the text mode.

Turbo C Tirade

I have to bitch about Turbo C for a bit. Version 1.0 has some major problems. The floating point library does peculiar things. And the classic style of function definition doesn't work with floats. The following function will print a value for var1, but it won't be the value you passed.

```
void param_pass (var1)
float var1;
{
    printf ("value = %f", var1);
}
```

Changing the function to:

```
void param_pass (float var1)
{
    printf ("value = %f", var1);
}
```

solves the problem. There shouldn't have been a problem. Things get worse when you use inline assembly language. Now, anytime you use a float in an assignment statement, the compiler does some funny casting and MASM gags. (Hence the popular expression, "That's enough to gag a MASM.")

But I love Turbo C in spite of these annoyances. The integrated environment makes program development much easier. And compilation and execution times are impressive. I wish Borland had had its act together for version 1.0, but so far I've been able to work around the problems.

Play Time

I've covered only the bare essentials, but they'll be enough to get you started. If you're interested, I'll go over arc drawing, shape filling, and the like in a future article. So do some fractals or spiff up your latest program. Enjoy yourself. Graphics needn't be useful, just fun.



Figure 1 - Graphics Routines For The Hercules Card

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#define boolean char
#define yes 1
#define no 0
#define index 0x3b4 /* 6845 ports */
#define data 0x3b5
#define mode 0x3b8 /* Herc ports */
#define status 0x3ba
#define config 0x3bf

const char normal = 0x07; /* char attributes for direct memory screen writes */
const char reverse = 0x70;
const char under_line = 0x01;
const char normal_blink = 0x87;
const char reverse_blink = 0xf0;
const char under_blink = 0x81;
const int page0 = 0xb000; /* base address of video pages */
const int page1 = 0xb800;
const int text_base = 0xb000; /* base address of text page */
const int max_col = 719; /* Herc parameters */
const int max_row = 347;
const int screen_size = 0x800; /* bytes per page */

int page_drawn, page_viewed; /* current drawing and viewing pages */

void init () /* set initial values for global variables */
{
    page_drawn = page0;
    page_viewed = page0;
} /* init */

boolean herc_there () /* test for presence of Hercules card */
{
    int i;

    geninterrupt (0x11);
    if ((_AX & 0x30) == 0x30) /* check for monochrome card presence */
        for (i = 0; i < 0x800; i++) /* is it a Herc? */
            if (inportb (status) & 0x80) /* watch bit 7 */
                return yes;
    return no;
} /* herc_there */

void screen_off ()
{
    outportb (mode, 2);
} /* screen_off */

void set_page_viewed (int page)
{
    if (page == page0)
        outportb (mode, 0x0a);
    else
        outportb (mode, 0x8a);
    page_viewed = page;
} /* set_page_viewed */

void set_page_drawn (int page)
{
    page_drawn = page;
} /* set_page_drawn */

void clear_gr_scr (int page)
{
    int i;

    if (page_viewed == page)
        screen_off (); /* turn off screen if page is displayed */
    for (i = 0; i < 0x3FFF; i++) /* reset all bits in the page */
        poke (page, 2 * i, 0);
    if (page_viewed == page) /* turn screen on again */
        set_page_viewed (page);
} /* clear_gr_scr */

void clear_txt_scr ()
{
    int i;

    for (i = 0; i <= 0x3fff; i++) /* fill text pg with nuls & attribute 7 */
        poke (text_base, 2 * i, 0x0700);
} /* clear_txt_scr */
```

```

void scr_write (int X, int Y, char str [81], char attrib)
{
    /* direct memory write of text */
    int i, offset;

    i = 0;
    offset = 2 * (80 * Y + X); /* find memory offset of first char in str */
    while (str [i] != '\0') /* poke each char/attrib pair */
    {
        pokeb (text_base, offset, str [i]);
        pokeb (text_base, ++offset, attrib);
        i++;
        offset++;
    }
} /* scr_write */

void set_graphics ()
{
    char i;
    char params [12] = {0x35, 0x2d, 0x2e, 0x07, 0x5b, 0x02,
                      0x57, 0x57, 0x02, 0x03, 0x00, 0x00};

    outportb (config, 3); /* allows both graphics pages */
    screen_off ();
    for (i = 0; i < 12; i++) /* program 6845 registers for graphics */
    {
        outportb (index, i);
        outportb (data, params [i]);
    }
    clear_gr_scr (page0); /* clear and turn on screen */
    clear_gr_scr (page1);
} /* set_graphics */

void set_text ()
{
    char i;
    char params [12] = {0x61, 0x50, 0x52, 0x0f, 0x19, 0x06,
                      0x19, 0x19, 0x02, 0x0d, 0x0b, 0x0c};

    outportb (config, 0); /* lock out graphics modes */
    outportb (mode, 0); /* set text mode and blank screen */
    for (i = 0; i < 12; i++) /* program 6845 registers for text */
    {
        outportb (index, i);
        outportb (data, params [i]);
    }
    clear_txt_scr ();
    outportb (mode, 0x28); /* enable blink and turn on screen */
} /* set_text */

void save_screen (char fname [13], int page) /* write to disk */
{
    FILE *f;
    char far *screen;

    screen = page * 0x10000; /* wierd, but it works */
    f = fopen (fname, "w");
    fwrite (screen, screen_size, 1, f);
    fclose (f);
} /* save_screen */

void get_screen (char fname [13], int page) /* read from disk */
{
    FILE *f;
    char far *screen;

    screen = page * 0x10000;
    f = fopen (fname, "r");
    screen_off ();
    fread (screen, screen_size, 1, f);
    set_page_viewed (page); /* turn the screen back on */
    fclose (f);
} /* get_screen */

void draw_point (int col, int row, boolean fill)
{
    int byte_ofs; /* offset within page for byte containing the point */
    char mask; /* locates point within the byte */

    mask = 1 << (7 - (col % 8));
    byte_ofs = 0x2000 * (row % 4) + 90 * (row / 4) + (col / 8);
    if (fill) /* draw the point */
        pokeb (page_drawn, byte_ofs, peekb (page_drawn, byte_ofs) | mask);
    else /* erase the point */
        pokeb (page_drawn, byte_ofs, peekb (page_drawn, byte_ofs) & ~mask);
} /* draw_point */

```

FORTHkit

5 Mips computer kit

\$400

Includes:

Novix NC4000 micro
160x100mm Fk3 board
Press-fit sockets
2 4K PROMs

Instructions:

Easy assembly
cmFORTH listing
shadows
Application Notes
Brodie on NC4000

You provide:

6 Static RAMs
4 or 5 MHz oscillator
Misc. parts
250mA @ 5V
Serial line to host

Supports:

8 Pin/socket slots
Eurocard connector
Floppy, printer,
video I/O
272K on-board memory
Maxim RS-232 chip

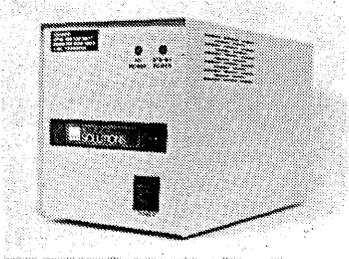
Inquire:

**Chuck Moore's
Computer Cowboys**

410 Star Hill Road
Woodside, CA 94062
(415) 851-4362

Reader Service Number 21

UNINTERRUPTABLE POWER SOURCE



MICRO

SOLUTIONS protects your equipment and your data from **power outages** and **brownouts**. Our power systems provide the **fastest switching speed** in the industry (2 ms ± 1).

EMI/RFI filtering and surge/spike protection all in one affordable unit. 1 year warranty on all units. Available in a size to suit your needs -

200 watts	\$290.00
350 watts	\$360.00
550 watts	\$410.00
800 watts	\$610.00
1000 watts	\$710.00

Includes shipping to your door in the continental U.S.. As specialists in overseas systems, we can supply 220 volt units. Call or write for details.

SOFTWARE SPECIAL

BROOKLYN BRIDGE

The **BROOKLYN BRIDGE** supplies the link between the new PS/2 IBM computers or laptops that use 3.5" diskettes and the rest of the MS-DOS world still using 5¼" drives. The cable supplied will allow you to transfer files and software between the two computers **FAST**. Simple to use and reliable. Get it now for only **\$99.00**



ASHER TECHNOLOGY PUTS FAX ON YOUR DESKTOP OR YOU CAN TAKE IT WITH YOU ON YOUR LAPTOP FOR LESS THAN \$500. We have found the answer to every small business's need for FAX at a price they can afford, and that works. Get in on this exciting new technology. Call for free demo disk!



P.O. Box 166 Riner, VA 24149
1-800-323-4829 (703) 382-6624
call 24 hours - 7 days a week

Visa MasterCard C.O.D.

We Ship Worldwide Dealers Supported

Reader Service Number 24

```

void draw_line (int X1, int Y1, int X2, int Y2, boolean on)
{
    int dX, dY;                /* use Bresenham's algorithm to draw a line */
                                /* vector components */
    int row, col,
        final,                /* final row or column number */
        G,                    /* used to test for new row or column */
        incl,                /* G increment when row or column doesn't change */
        inc2;                /* G increment when row or column changes */
    boolean pos_slope;

    dX = X2 - X1;                /* find vector components */
    dY = Y2 - Y1;
    pos_slope = (dX > 0);        /* is slope positive? */
    if (dY < 0)
        pos_slope = !pos_slope;
    if (abs (dX) > abs (dY))    /* shallow line case */
    {
        if (dX > 0)            /* determine start point and last column */
        {
            col = X1;
            row = Y1;
            final = X2;
        }
        else
        {
            col = X2;
            row = Y2;
            final = X1;
        }
        incl = 2 * abs (dY);    /* determine increments and initial G */
        G = incl - abs (dX);
        inc2 = 2 * (abs (dY) - abs (dX));
        if (pos_slope)
            while (col <= final) /* step thru cols. checking for new row */
            {
                draw_point (col, row, on);
                col++;
                if (G >= 0)        /* it's time to change rows */
                {
                    row++; /* positive slope, so increment through the rows */
                    G += inc2;
                }
                else                /* stay at the same row */
                    G += incl;
            }
        else
            while (col <= final) /* step thru cols. checking for new row */
            {
                draw_point (col, row, on);
                col++;
                if (G > 0)        /* it's time to change rows */
                {
                    row--; /* negative slope, so decrement through the rows */
                    G += inc2;
                }
                else                /* stay at the same row */
                    G += incl;
            }
    } /* if |dX| > |dY| */
    else                            /* steep line case */
    {
        if (dY > 0)            /* determine start point and last row */
        {
            col = X1;
            row = Y1;
            final = Y2;
        }
        else
        {
            col = X2;
            row = Y2;
            final = Y1;
        }
        incl = 2 * abs (dX);    /* determine increments and initial G */
        G = incl - abs (dY);
        inc2 = 2 * (abs (dX) - abs (dY));
        if (pos_slope)
            while (row <= final) /* step thru rows - check for new column */
            {
                draw_point (col, row, on);
                row++;
                if (G >= 0)        /* it's time to change columns */
                {
                    col++; /* pos. slope, so increment through the columns */
                }
            }
    }
}

```

```

    G += inc2;
  }
  else
    G += incl;
}
else
  while (row <= final) /* step thru rows checking for new column */
  {
    draw_point (col, row, on);
    row++;
    if (G > 0)
      /* it's time to change columns */
      {
        col--; /* neg slope, so decrement through the columns */
        G += inc2;
      }
    else
      /* stay at the same column */
      G += incl;
  }
} /* draw_line */

```

Figure 2 - 6845 Registers

Register	Function	Settings (hex)	
		Text	Graphics
0	Total horizontal characters	61	35
1	Displayed horizontal characters	50	2d
2	Horizontal sync position	52	2e
3	Sync width	0f	07
4	Total vertical rows	19	5b
5	Vertical total adjust	06	02
6	Vertical rows displayed	19	57
7	Vertical sync position	19	57
8	Interlace mode and skew	02	02
9	Maximum scan line address	0d	03
10	Cursor's starting scan line	0b	00
11	Cursor's ending scan line	0c	00

ANNOUNCING A CALCULATED BREAKTHROUGH IN COMPUTING

If you use a
SCIENTIFIC OR FINANCIAL
CALCULATOR and a
PERSONAL COMPUTER,
you need the
PC HYPERCALCULATOR.

YOU NEED: A pop-up scientific/financial calculator for the IBM PC integrated with every program you use.

YOU NEED: A programmable calculator with 100 registers, 1000 program steps, and alphanumeric prompts.

YOU NEED: A faithful emulation of the Hewlett-Packard HP-11C and HP-12C that runs 20-40 times faster.

YOU NEED: Only \$49.95 plus \$3 for shipping (includes free 8087 version and utility programs).

Stop copying from calculator
to computer now!

Call toll-free:

(800) 628-2828, ext. 502

Sunderland Software Associates
Post Office Box 7000-64
Redondo Beach, CA 90277

HP-11C, HP-12C and IBM PC are trademarks of Hewlett-Packard Co. and International Business Machines Corp., respectively.

Reader Service Number 41

ADD TO THE POWER OF YOUR PROGRAMS WHILE YOU SAVE TIME AND MONEY!

CBTREE does it all! Your best value in a B+tree source!

Save programming time and effort.

You can develop exciting file access programs quickly and easily because CBTREE provides a simple but powerful program interface to all B+tree operations. Every aspect of CBTREE is covered thoroughly in the 70 page Users Manual with complete examples. Sample programs are provided on disk.

Gain flexibility in designing your applications.

CBTREE lets you use multiple keys, variable key lengths, concatenated keys, and any data record size and record length. You can customize the B+tree parameters using utilities provided.

Your programs will be using the most efficient searching techniques.

CBTREE provides the fastest keyed file access performance, with multiple indexes in a single file and crash recovery utilities. CBTREE is a full function implementation of the industry standard B+tree access method and is proven in applications since 1984.

Access any record or group of records by:

- Get first
- Get previous
- Get less than
- Get greater than
- Get sequential block
- Get all partial matches
- Insert key and record
- Delete key and record
- Change record location
- Get last
- Get next
- Get less than or equal
- Get greater than or equal
- Get partial key match
- Get all keys and locations
- Insert key
- Delete key

Increase your implementation productivity.

CBTREE is over 6,000 lines of tightly written, commented C source code. The driver module is only 20K and links into your programs.

Port your applications to other machine environments.

The C source code that you receive can be compiled on all popular C compilers for the IBM PC and also under Unix, Xenix, and AmigaDos! No royalties on your applications that use CBTREE. CBTREE supports multi-user and network applications.

CBTREE IS TROUBLE-FREE, BUT IF YOU NEED HELP WE PROVIDE FREE PHONE SUPPORT.

ONE CALL GETS YOU THE ANSWER TO ANY QUESTION!

CBTREE compares favorably with other software selling at 2,3 and 4 times our price.

Sold on unconditional money-back guarantee.

YOU PAY ONLY \$99.00 - A MONEY-SAVING PRICE!

TO ORDER OR FOR ADDITIONAL INFORMATION

CALL (703) 356-7029 or (703) 847-1743

OR WRITE



Peacock Systems, Inc., 2108-C Gallows Road, Vienna, VA 22180

Reader Service Number 20

Controlling Synchronous Serial Chips With A Parallel Port

Long Distance Data Transfers At Incredible Speeds

This month I'm going to show you how to use your parallel printer port to control serial TTL chips. The output chip will drive a seven segment display, and the input chip will read setpoint values from the outside world. The Turbo C code is a background task triggered by the serial clock interrupt, so you can run all your normal programs while monitoring and controlling an external process. Real Multitasking! But first, a little deviation.

My friend Daniel Will-Harris has written his first book, called "Desktop Publishing With Style." It's filled with great examples and is **MUST** reading if you're into serious desktop publishing on the PC.

Daniel and I have competed in everything since junior high school. There was even a bet about who was going to be taller (for which, I might mention, I've never been paid. Ahem). So when I found out he was writing a book, I of course got busy outlining MY book.

While researching it, I discovered a problem. Desktop publishing is a new field, and Daniel's book is the first which isn't just a rehash of a manual for people who've nipped the program. There are, however, many good electronics books. And as I read through them, I found much of what I was going to talk about already on paper. Could I really make a worthwhile contribution when the subject has been so well covered?

Chapter Two

Everyone's working on chapter two of a book, right? So, until I come up with a really smashing idea, I'll give you a list of electronics books which I've found clear and valuable:

(1) *The Art Of Electronics*, by Horowitz

and Hill. Sophisticated enough for designers, but with a nuts-n-bolts feel to it.

(2) *The Texas Instruments TTL Data Book* and the *National Semiconductor Linear Data Book*. A CMOS data book is also handy. Page through these and look at the diagrams. After a while, your brain will decide things aren't intolerably complicated.

(3) Forrest Mim's *Circuit Scrapbook II* (I don't have "I", but I assume it's as good). This guy has been designing small, clever circuits and writing clearly about them for as long as I can remember. I expect to get some article inspirations from this book.

(4) Don Lancaster's TTL and CMOS cookbooks are good introductions with many simple circuits. You could probably use these in lieu of data books for a while.

Books which deal with the innards of, and electronic interfacing to, PC's—

(5) *The IBM PC From the Inside Out*, by Sargent and Shoemaker. A great book: it explains electronics AND interfacing AND PC hardware. Buy it first.

(6) *Interfacing to the IBM Personal Computer*, by Eggebrecht. This is a condensation and clarification of the technical manuals. This book has more than its share of errors, but it's good for PC bus interfacing.

(7) The IBM technical manuals. If there's one thing Big Blue can do, it's document. The Rom BIOS listings are very useful as examples.

(8) *Programmer's Problem Solver for the IBM PC, XT & AT*, by Jourdain. Bite-size examples about each and every internal organ of the PC.

(9) Joseph Carr's books seem to cover the same topics, but he presents the subject well. I have *Interfacing Your Computer to Virtually Anything* from TAB (how's that for a one-line movie plot?).

There. That ought to keep you busy.

Synchronous Communications

And now down to business. In the last issue, I went into all the details of the \$21 Microsphere parallel printer card (which is identical to most others), so you know how to use it to control hardware. Now we'll do a rather odd thing: use the parallel port to control serial hardware.

This isn't serial hardware like you're used to thinking of: the serial port on your computer. Those are asynchronous serial ports, which means you give the UART (Universal Asynchronous Receiver-Transmitter) chip a byte, and it sends it out on a single wire, without any other wire containing synchronizing information (i.e., an additional clock signal to tell the receiving unit when the bits are valid).

Asynchronous transmission is tricky in that each bit transmits during a certain precise period of time within the byte transmission. And each byte is surrounded by start and stop bits to mark the beginning and the end. The timing information is, in essence, built into the signal.

A synchronous interface has separate data and clock lines. To send information, you put a bit on the data line, cycle the clock, put the next bit on the data line, etc. To receive information, you read the data line, cycle the clock, etc.

This system has two big advantages: (1) speed, and (2) the elimination of critical timing requirements. The fastest UART usually runs at 19.2 Kbaud (19,200 bits per second; the Zilog SCC—Serial Communications Controller on the Macintosh can run at 2 Mbaud, but it's not exactly the cheap, common UART used in RS-232).

In a synchronous system, we're only limited by the speed of our logic, which can be significant since any sort of serial communication is generally slower than parallel transfers. In this case, we're

limited by the speed of the PC since we're "bit banging."

Since we control the clock there are no critical timing requirements. If an interrupt occurs in the middle of a byte, fine: we can pick up whenever it's finished without corrupting anything. This is very convenient for software control. It's possible to make a software UART, but it must never be interrupted: so it won't work on a PC.

Just Two Little Chips...

Figure 1 shows a circuit for eight bits of synchronous serial input and output. I tried to be clever and minimize the number of communication lines. Originally I used three: one for clock, one for control, and one for data (you'll recall from issue #38 that four lines on the parallel port can be read AND written). But I found the single data I/O line was getting corrupted by the LS165, so I used separate data input and output lines.

I discovered the LS165 was STILL messing with the LS164. I gave it a separate load/select line to pin one. Things started working right, so I rearranged the proto-board. I accidentally pulled an important wire, but it kept on working anyway. I couldn't figure out why. I finally threw up my hands and stopped trying to save wires. You can probably find ways to combine them, but I had a deadline.

So much for clever. The PC turned out to be very useful for debugging hardware. One generally has to fetch an oscilloscope or logic analyzer to figure out what's happening. When driving the logic with a PC, however, you simply put in printf()'s and getch()'s. The latter causes the program to pause until you hit a key, to slow the thing down to a crawl. You can watch what's happening and tweak the lines any way you want. I prefer it to an oscilloscope, and it's MUCH cheaper.

Figure 1A - Converting Serial To Parallel.

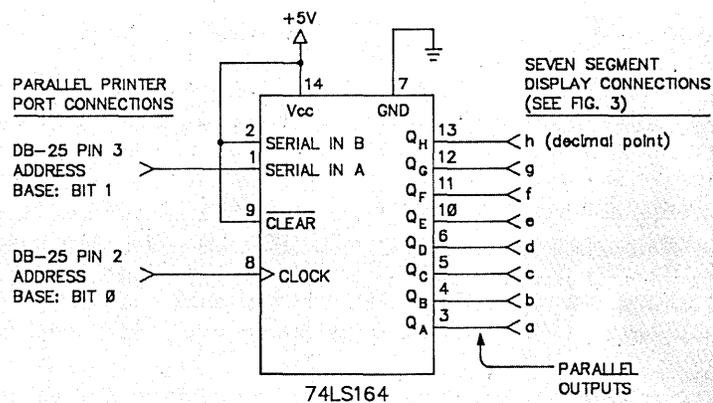


Figure 1B - Converting Parallel To Serial.

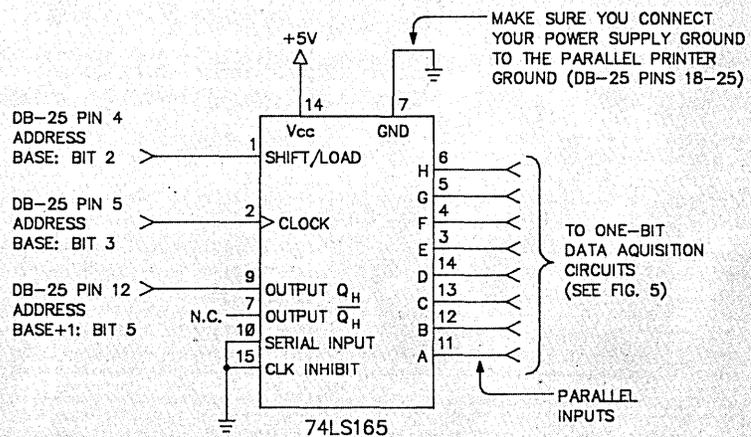


Figure 1C - Counting Pins From The Top Side Of An IC.

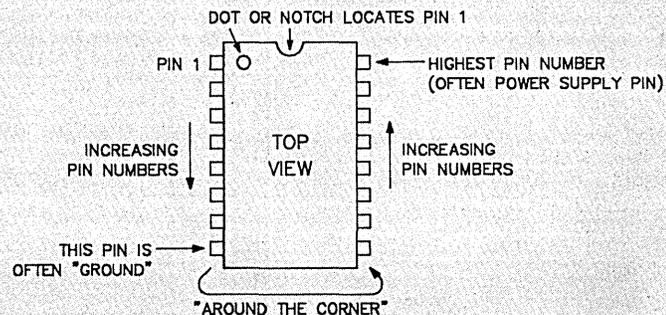
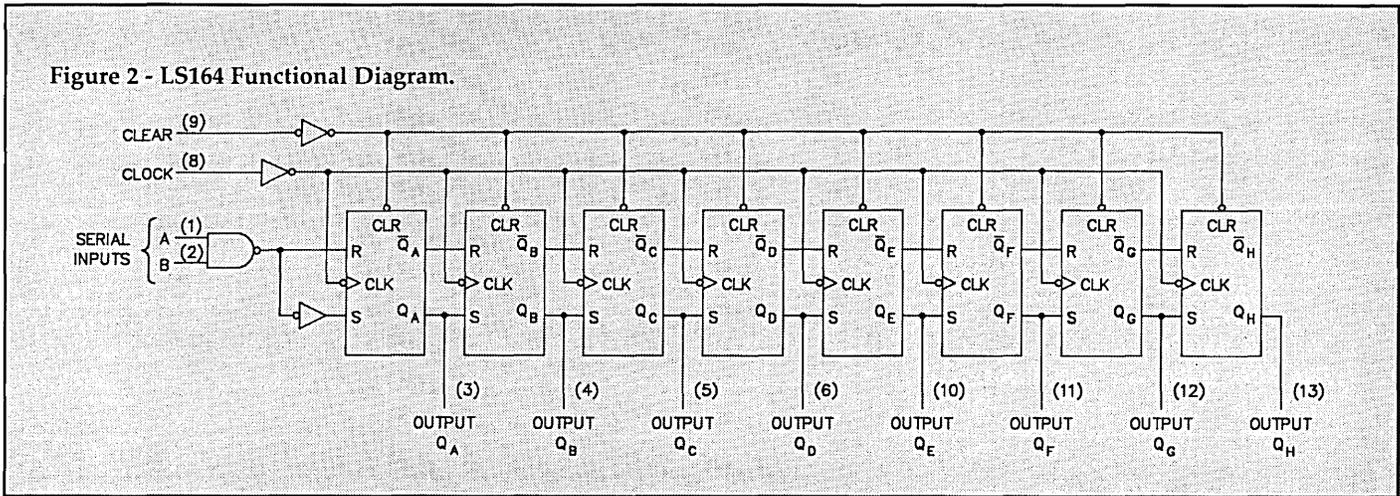


Figure 2 - LS164 Functional Diagram.



Serial In, Parallel Out

The LS164 is a "serial-in, parallel-out shift register." You shift eight bits of data in through the serial pin (using a serial clock), and it's presented at the eight parallel output pins. Simple as that. Figure 2 shows the internal logic of the chip (taken from the TI TTL data book); this is useful when you're trying to figure out how to connect it.

This chip can be a problem because its outputs show the data as it shifts into the chip, so they flicker. In this application I'm just driving a seven-segment display, so it doesn't matter. If you're driving relays or switches, you should put an octal latch on the outputs and gate the outputs with another control line when the shifting is complete.

The seven-segment display (Figure 3) is the "common anode" (C.A.) type, which means all the anodes of the LEDs in the display are connected. We use a C.A. type because, as is typical for TTL, the outputs will "sink" (draw in) much more current than they will "source" (put out). Thus we tie the common anode of the LED to +5V and each of the cathodes to an LS164 output pin and we can control each LED segment.

The specs in the data book say the chip will only sink 8 mA. Since the typical LED will handle over 20 mA, a direct connection should be okay. I tried this and it didn't burn the LEDs out, but the chip specs are usually conservative and the LEDs seemed awfully bright, so (I'm told) I probably shortened their life.

You're supposed to put one resistor in series with each LED, but I was lazy and just put a single resistor from the +5V supply. This looked fine to me, but if you use several displays you'll probably see the difference, so don't cheat on "production" designs. If you want to know more about LEDs, check

out my articles in past issues of *Micro C.*

Different seven-segment displays use different pin-outs. To discover whether you have a common-anode or common-cathode type, and which pin is common, use an ohmmeter. You can also use a 5V

DC power supply, but make sure you have a 300-ohm or so resistor in series or you'll fry a segment.

Parallel In, Serial Out

This chip (the LS165) was a

Figure 3 - Common - Anode Seven Digit Display.

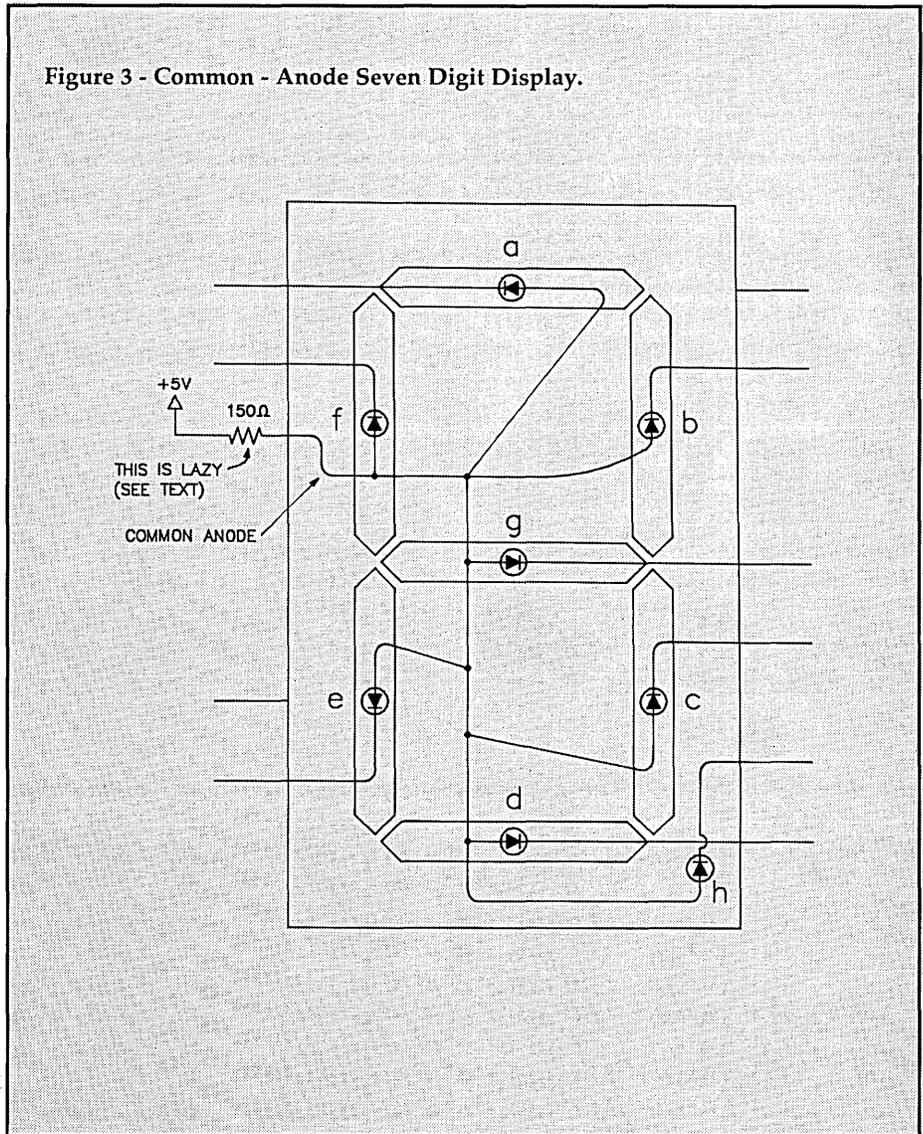
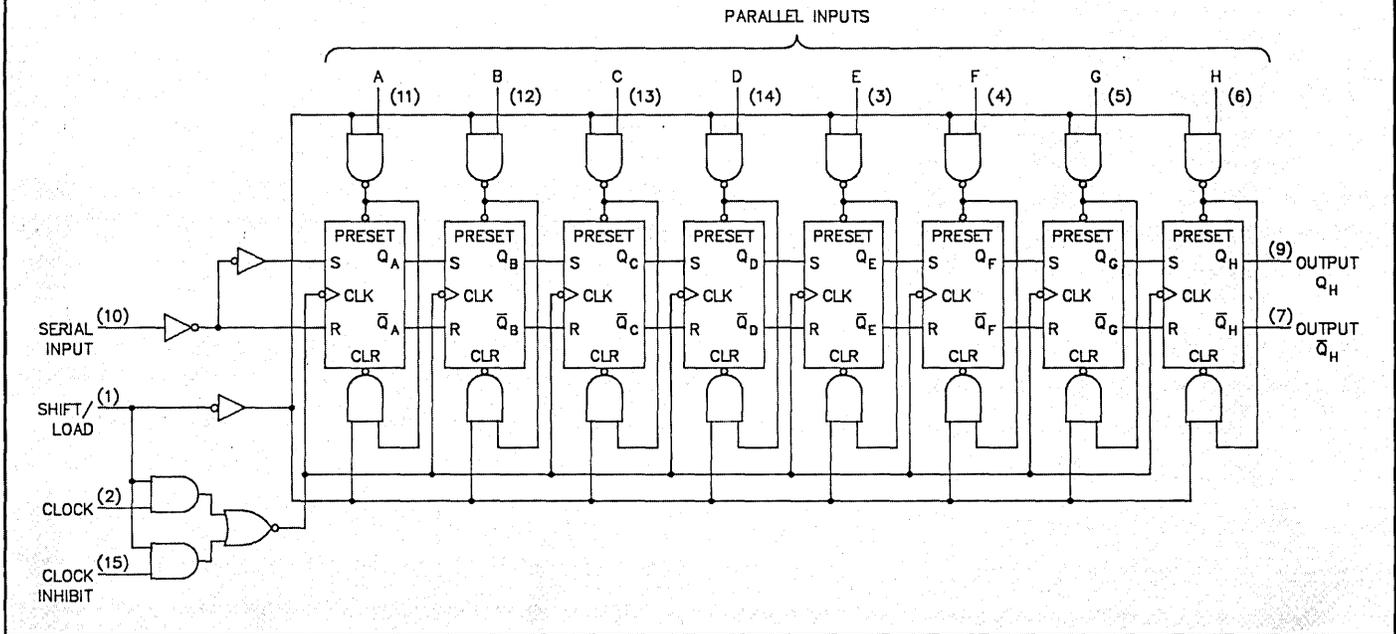


Figure 4 - LS165 Functional Diagram.



BOOGER. I suppose it's my own fault, really, because I've gotten casual about social amenities between chips. Tri-state logic politely disconnects itself from a data bus when you de-select it so other chips can use the bus. This is the way life should be in civilized society. There are, unfortunately, neanderthal chips with atrocious table manners.

When pin 1 (the shift/load pin) is low, the flip-flops will load the data from the eight parallel inputs (see Figure 4, the schematic for the chip). The clock line shifts this data out through the serial pin. Notice, though, that parallel input "H" (pin 6) is immediately present at the serial outputs when the chip loads. It just flows right through. This prevents the connection of the serial output to a common serial bus: the output is always "on," consequently it won't let anyone else use the bus. I had to find this out the hard way.

So What Good Is An Input?

Buttons and switches make good simple detectors and alarms. For measurements, use an A/D converter (some converters even have synchronous serial outputs). Sometimes, you just want to know if an analog value is above or below some arbitrary threshold. To accomplish this we use a comparator.

A comparator is just an op-amp (see issues #35 & #36) with some of the internal damping circuitry taken out. This increases the speed of the amplifier and

makes it more unstable, but a comparator just zips from one rail to the other and never spends time in the middle (where stability is a problem).

Figure 5 shows how to connect a switch or a comparator to the inputs of the LS165. When the "+" input of the comparator is higher than the "-" input, the output will go to the positive rail; when it's lower than the "-" input, the output will go to the negative rail. The output of the LM339 comparator is "open-collector:" it doesn't pull itself up when the output is high. You have to help it with a pull-up resistor.

The LM339 is a brother to the LM324 op-amp (which I've discussed before), and as common. You can also use the LM324 in this application because speed isn't a problem.

Whatever you connect to the "+" input of the comparator should be connected to the same supply voltages as the comparator (in this case, 0V and 5V) to keep the input voltage inside the range of the comparator.

There are tons of simple, useful things you can do with these cheap workhorses. Figure 5 shows two applications: a CdS cell (available at Radio Shack) for light threshold sensing, and a thermistor for temperature threshold sensing.

You simply adjust the potentiometer (variable resistor) until you reach the desired switching threshold. Since the inputs of the comparator are very high impedance (they draw almost no cur-

rent), you can connect the output of a sensor to several comparator inputs and be notified as it crosses different thresholds (see Figure 5).

Connections

I modified my \$21 Microsphere parallel card to I/O address 0x238 (see issue #38). The I/O map in the IBM technical reference manual says this is unused.

I use a ribbon cable and a DB-25 clamp-on ribbon cable connector for the computer end, and a row of header pins to plug into my proto-board at the other end. You can get most of the parts from:

JDR
110 Knowles Drive
Los Gatos, CA 95030
(800) 538-5000
or (408) 866-6200.

I checked with my friend Brinkley Barr about cabling distances. As the system stands, he wouldn't trust it farther than 20 or 30 feet. For long distances, he suggests the following: pull the 74LS374 from the printer card and replace it with a CMOS 74HCT374. At the remote end, use CMOS 74HC14 Schmitt trigger inverters as buffers and CMOS chips everywhere else. (This has other advantages: in his project, he's using the 74HC594, which is like our LS165 but with built-in octal latches on the parallel outputs. Necessary for driving relays.)

The noise immunity gained from

CMOS should allow cabling distances in the hundreds of feet (watch your grounds, though!). If you want further design details, send me mail.

Software

When reading articles in magazines, I find it painful to flip back and forth from the article to the code. I've tried to pack everything into the comments.

There are three pieces of code, and one header file. The file CLOCKINT.C installs an interrupt service routine (ISR) in the clock interrupt (which calls the former clock ISR when it's finished). The file MYFUNC.C is the function called by the ISR; you can modify it to make your own clock-based background process. SERIAL.C contains the functions which write to the serial chip hardware. By changing the "#undef test" at the beginning of this file to "#define test," you can create a stand-alone test program.

The header file COLORS.H contains definitions to control the CGA, but you could easily add Hercules compatibility and make the system portable to both systems.



68000

and 6800/2/8/9

SOFTWARE

SK-DOS - a powerful DOS for the 6809 (\$75) or the 68000 (\$140, incl. an editor, assembler, Basic, utilities, code for a boot ROM, etc.)

HUMBUG - a monitor/boot ROM, \$50 - \$75.

OTHER SYSTEM SOFTWARE including assemblers, text formatters, editors, spell checkers, languages, etc., all very reasonable.

HARDWARE

A wide selection of single-board computers and systems, from \$275.

COMBINATIONS

Package deals of fast and powerful computer plus DOS and more, from \$350.



SOFTWARE SYSTEMS CORP.

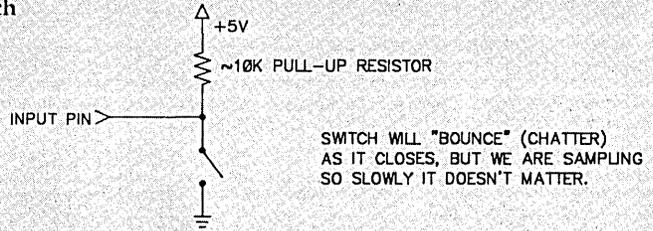
BOX 209 • MT. KISCO, NY 10549

914/241-0287

Reader Service Number 40

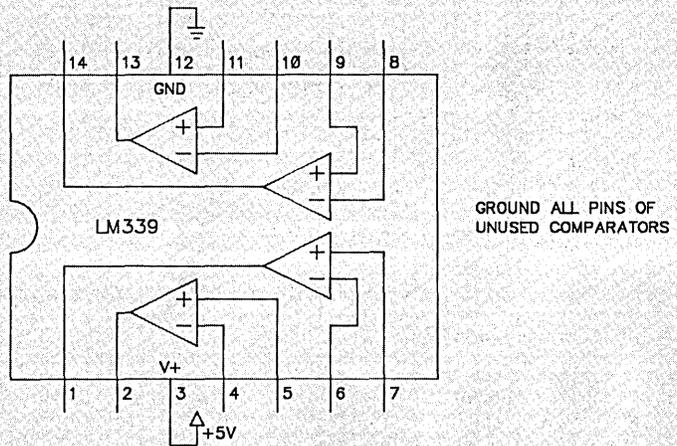
Figure 5 - One - Bit Data - Acquisition Circuits.

Figure 5A - Switch



SWITCH WILL "BOUNCE" (CHATTER) AS IT CLOSSES, BUT WE ARE SAMPLING SO SLOWLY IT DOESN'T MATTER.

Figure 5B - LM339 Quad Comparator Pinout



GROUND ALL PINS OF UNUSED COMPARATORS

Figure 5C - One Bit A/D Converter

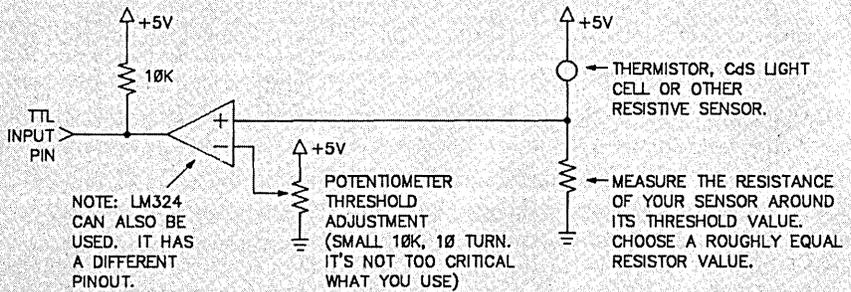
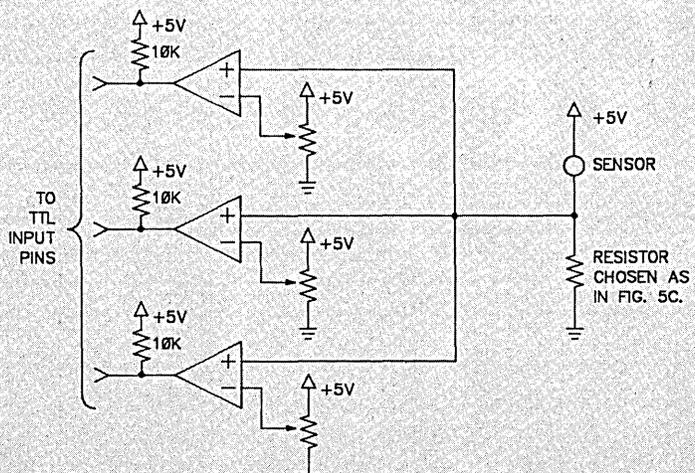
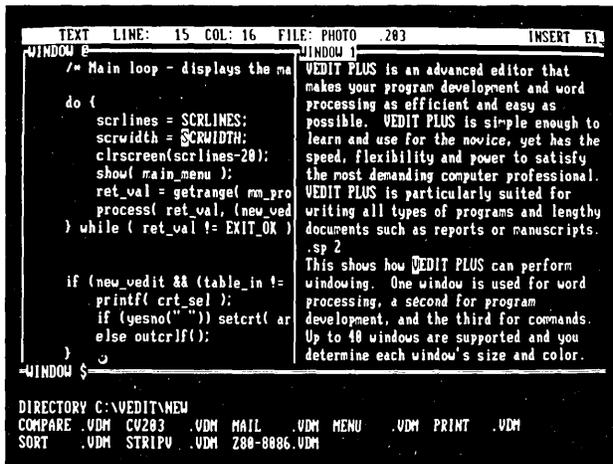


Figure 5D - Monitoring Several Thresholds on the Same Sensor



New, Lower Prices for CP/M

- VEDIT Version 1.40 \$49 (Single file, no windows)
- VEDIT PLUS Version 2.32 \$79 (Multiple file, no windows)
- VEDIT PLUS Version 2.33 \$95 (Current version with windows)



#1 PROGRAMMABLE EDITOR

FREE Fully Functional Demo Disk *

Stunning speed. Unmatched performance. Total flexibility. Simple and intuitive operation. The newest VEDIT PLUS defies comparison.

Try A Dazzling Demo Yourself.

The free demo disk is fully functional - you can try all features yourself. Best, the demo includes a dazzling menu-driven tutorial - you experiment in one window while another gives instructions.

The powerful 'macro' programming language helps you eliminate repetitive editing tasks. The impressive demo/tutorial is written entirely as a 'macro' - it shows that no other editor's 'macro' language even comes close.

Go ahead. Call for your free demo today. You'll see why VEDIT PLUS has been the #1 choice of programmers, writers and engineers since 1980.

Available for IBM PC, Tandy 2000, DEC Rainbow, MS-DOS, CP/M-86 and CP/M-80. (Yes! We support windows on most CRT terminals, including CRT's connected to an IBM PC.) Order direct or from your dealer. \$185.

Compare features and speed

	BRIEF	Norton Editor	PMATE	VEDIT PLUS
'Off the cuff' macros	No	No	Yes	Yes
Built-in macros	Yes	No	Yes	Yes
Keystroke macros	Only 1	No	No	100+
Multiple file editing	20+	2	No	20+
Windows	20+	2	No	20+
Macro execution window	No	No	No	Yes
Trace & Breakpoint macros	No	No	Yes	Yes
Execute DOS commands	Yes	Yes	Yes	Yes
Configurable keyboard layout	Hard	No	Hard	Easy
'Cut and paste' buffers	1	1	1	36
Undo line changes	Yes	No	No	Yes
Paragraph justification	No	No	No	Yes
On-line calculator	No	No	No	Yes
Manual size / index	250/No	42/No	469/Yes	380/Yes

Benchmarks in 120K File:

2000 replacements	1:15 min 34 sec		1:07 min 6 sec	
Pattern matching search	20 sec	Cannot	Cannot	2 sec
Pattern matching replace	2:40 min	Cannot	Cannot	11 sec

VEDIT and CompuView are registered trademarks of CompuView Products, Inc. BRIEF is a trademark of UnderWare, Inc. PMATE is a trademark of Phoenix Technologies Ltd. Norton Editor is a trademark of Peter Norton Computing Inc.

* Demo Disk is fully functional, but does not readily write large files.

Call for 286 / XENIX Version Fully Network Compatible

- 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 support.
- Horizontal scrolling - edit long lines.
- Flexible 'cut and paste' with 36 'scratch-pad' buffers.
- Customization - determine your own keyboard layout, create your own editing functions, support any screen size.
- Optimized for IBM PC/XT/AT. Color windows. 43 line EGA.

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/Undent for 'C', PL/I, PASCAL, etc.
- 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 to/from WordStar and mainframe files.
- Print any portion of file; selectable printer margins.

MACRO PROGRAMMING LANGUAGE

- 'If-then-else', looping, testing, branching, user prompts, keyboard input, 17 bit algebraic expressions, variables.
- Flexible windowing - forms entry, select size, color, etc.
- Simplifies complex text processing, formatting, conversions and translations.
- Complete TECO capability.
- Free macros: • Full screen file compare/merge • Sort mailing lists • Print Formatter • Menu-driven tutorial

CompuView

Figure 6 - Turbo C source of Synchronous Interface.

```

/* Complete listing (1) -- CLOCKINT.C, MYFUNC.C, SERIAL.C, COLORS.H */

/* CLOCKINT.C : Turbo C program to install your own interrupt
function in the clock interrupt, while still preserving everyone
else's. Bruce Eckel, Eisys Consulting 1987. To make the control
program, you must create a project file which contains the
following, and make it's file the "project name": colors.h is a
header which the C file includes; if the header gets changed,
that C file will be re-compiled even if it wasn't modified
itself. */

#define CLOCKINT 8 /* IRQ0 on the 8259 is interrupt 8 in the
vector table */

#define PROG_SIZE 0x87F /* Run the Turbo C compiler with the
options:linker:mapfile set to "segments." Look at the mapfile
generated for this program. The "stop" address for the stack is
the highest adress used -- set PROG_SIZE to this value for use
with the "keep()" command */

void interrupt (*old_int_handler)();

/* This, believe it or not, is a variable declaration. I've put
it way out here (outside of any function declarations) so it
doesn't go away when main exits (i.e. it's global). It holds the
address of the old interrupt handler function so the NEW
interrupt handler can call the OLD one when it's through. To read
it, start in the middle and work your way out: old_int_handler is
a POINTER ("*") to a FUNCTION ("()") of type "interrupt"; this
function doesn't return a value ("void"). */
main()
    /* all this does is install the interrupt handler. I
don't use any "printf()s" because it increases the
resident program size so much. */
{

/* first: function prototypes, right out of the book. These
don't generate any code, but they tell the compiler (and us)
exactly how the functions should be called so it can catch
mistakes. The same effect can be achieved with "#include
<dos.h>", but this is more educational. */

extern void interrupt (*getvect(int intr_num))();
/* The above declaration is tricky to read: getvect returns a
pointer to a function of type interrupt (and an "interrupt"
function never returns a value, thus the "void"). It's deceiving
because the function "getvect" is actually returning something,
but the first word you see is "void" */

extern void setvect(int intr_num, void interrupt (*isr)());
/* This function really DOESN'T return anything. It must
be passed an integer and a pointer to a function of type
interrupt which has no return value. The ability to pass
function addresses around allows us to almost entirely eliminate
the need for assembly language when using C. */

void interrupt my_int_handler();
/* Prototype for the interrupt we are inserting in the clock
chain (a lot like inserting an item in a linked list). Notice
the word "extern" doesn't appear in this declaration because the
function definition is contained in this file. */

/* The code is trivial: get the old clock interrupt vector (for
our routine to pass control to when it's done), install our
routine as the new interrupt service routine, and call "terminate
but stay ready." */

old_int_handler = getvect(CLOCKINT); /* store the address of the old
handler */
setvect(CLOCKINT, my_int_handler);
keep(0,PROG_SIZE); /* first parameter is exit status */
}

```

(Listing continues on page 38.)

DEBUGGING SWAT TEAM

Order Eco-C88 Rel. 4.0 New Modeling Compiler
and get C-more at no extra charge!

Seek and Correct

You already know that fast compilation does not mean fast program development. Backing up for bogus error messages and removing the bugs takes time. Eco-C88's "Seek and Correct" three-way error checking finds even the most elusive bugs, clearing the path for swift program development.

Double Barrel Error Checking

Eco-C88 nails syntax errors cold and tells you about the error in plain English. And there's no avalanche of false error messages, either. Other compilers can generate up to four times the number of error messages actually present; they leave it up to you to guess which ones are real. You'll be more productive with Eco-C88 because there is no guess work.

Eco-C88 provides ten levels of semantic error checking. You can select from almost no checking to the fussiest you've ever seen. Eco-C88's "picky flag" finds subtle errors that slip by other compilers.

Eco-C88 also features:

- All data types, plus ANSI Enhancements
- Robust library, including many new ANSI functions
- CED editor with online function help, split windows, compile-edit-link capability
- New, expanded manual with sample programs for the library functions

C-more Source Code Debugger

Finally, if a really nasty bug persists, put C-more, our source code debugger, to work. With C-more you can watch your program as it executes, single-step it, set simple or conditional breakpoints, test complex expressions, use variables as indexes into other variables, initialize and trace variables, examine CPU registers, display results with printf()-type options and much more. C-more can help you track down bugs in minutes rather than days.

The price for Eco-C88 is \$99.95. And, for a limited time, we'll give you our C-more debugger at no extra charge.

Ecosoft Inc.

6413 N. College Ave.
Indianapolis, IN 46220

(317) 255-6476 (Tech Info)
(800) 952-0472 (Orders)

Memory Options

Model	Code Size	Data Size
Small	64K	64K
Compact	64K	1 Meg
Medium	1 Meg	64K
Large	1 Meg	1 Meg

(Listing continued from page 36.)

```
void interrupt my_int_handler()
{
    extern void my_function(); /* i.e.: this is defined in another file */
    const int clock_count = 9; /* wait this many ints between invoking our
        function; there are about 18 interrupts per second. */
    static int i = 0; /* The interrupt counter doesn't go away
        between interrupts */

    if(i++ == clock_count) { my_function(); i = 0;} /* Do whatever we want... */
    /* (with the usual caveat that we can't make any DOS or BIOS calls) */
    (*old_int_handler)(); /* ... and continue the chain by de-referencing the
        pointer to the old clock service routine. */
}

/* This was just TOO easy !! */

/* A caveat: I haven't done extensive testing, so this could slow
down the time-of-day clock on the PC. If it does, one might add
a function to restore the correct time from the battery-backed
clock-calendar. */

/* MYFUNC.C: Turbo C function installed as a TSR by CLOCKINT.C.
Bruce Eckel, Eisys Consulting 1987. This integrates the serial
input and output functions, and some screen display code from
issue #38 (which must be used since an interrupt routine can't
make DOS calls; i.e. DOS isn't re-entrant). Each time
my_function() is called, it outputs a new character from the
led_table[] to the serial-to-parallel chip, gets a byte from the
parallel-to-serial chip, and displays the byte as ones and zeroes
in the lower right corner of the screen.

You can replace the code in my_function() with anything else
(as long as your code or any C library functions you call don't
make DOS or ROM BIOS calls -- printf() does) to create your own
clock interrupt background process. */

#include "colors.h" /* #defines for the CGA */

/* Now we tell the compiler that "led_table[]" is in another
file. We still have to show what it looks like so the compiler
can generate the right code when we reference elements of the
structure. The actual address of led_table[] is figured out
("resolved") by the linker. This type of declaration is often
contained in a header file as a "typedef." The header file is
then included in the file where the table is defined and any
other files where it is declared. Note, however, that a header
file with anything other than definitions in it (i.e. code) is
asking for trouble. */

extern const struct {
    char character;
    unsigned char code;
} led_table[];

/* Here's the function which is called by my_int_handler() */
void my_function()
{
    void display_byte(unsigned char byte);
    extern void outbyte(unsigned char ch);
    extern unsigned char inbyte();

    static int i = 0; /* statics don't go away between function
        calls, so we don't output the first character every time. It's
        only initialized to zero when the program is loaded. */

    display_byte(inbyte()); /* Read the LS165 and display it on the screen. */
    outbyte(led_table[i].code); /* send a character to the seven-segment
        display.*/
    if(led_table[++i].code == 0) /* increment the table index and see if
        we're at the end of the table; */
        i = 0; /* if so, start over. */
}

```

(Listing continues on page 40.)

HALTED

SERVING NORTHERN CALIFORNIA
SINCE 1963!

SPECIALTIES

HARD DRIVES + CONTROLLERS

RACAL/VADIC STAND ALONE MODEM

- MODEL 212LP
- 1200/300 BAUD
 - BELL 212/103 COMPATIBLE
 - AUTO DIAL/AUTO ANSWER
 - OEM VERSION P.S. & CASE NOT INCLUDED.

\$19⁹⁵

SWITCHING POWER SUPPLY—
IDEAL FOR DISK DRIVES— 50 Watt +
5V 4 AMP +12V 3 AMP •4" x 6" x 1-3/4"



\$ 29.95

SHUGART SA801R 8" FLOPPY
DRIVE— 600K SSDD • Still Sealed in
Factory Boxes

\$ 149.00

—IC PROGRAMMERS— XT/AT COMPATIBLE

ALL USE SINGLE SLOT IN YOUR XT/AT AND INCLUDE
EXTERNAL PODS WITH ZERO INSERTION FORCE
SOCKET(S)

PAL PROGRAMMER *269.95

- PROGRAMS MOST 20 & 24 PIN PALS
 - VERIFY, PROGRAM & BURN SECURITY FUSE LINK
- 8748/49 PROGRAMMER *199.85

—EPROM PROGRAMMERS—

PROGRAMS 27XX to 27512
12.5, 21, 25V DEVICES INCLUDES SOFTWARE FOR
STANDARD & INTEL HEX FORMATS

- | | |
|-----------------------|---------|
| SINGLE GANG | *109.95 |
| FOUR GANG | *169.95 |
| TEN GANG | *309.95 |

**CORVUS HARD DRIVES + SOFTWARE
AVAILABLE NOW!! PLEASE CALL!!**

IF YOU DON'T SEE WHAT YOU WANT, CALL US!



HALF HEIGHT EXTERNAL DRIVE ENCLOSURE

- Attractive Low Profile Chassis 19x15x2 3/4
- Fits nicely directly under PC
- Standard IBM Colors
- Bezel for 5 1/4" and 3 1/2" Drive
- 60 Watt P.S. w/DC Fan and Drive Cables

\$89⁹⁵/\$29⁹⁵
complete chassis only

HALTED SPECIALTIES offers its customers
Unique "SUPERMARKET STYLE" shopping
for ANY and ALL ELECTRONIC NEEDS. We
stock literally THOUSANDS of parts— from the
newest IC's to some of the first transistors.
ALSO, for the electronic enthusiast, we carry a
full line of computer accessories, test
equipment, tools, R+D supplies and much
more! PLEASE CALL OR VISIT ONE OF OUR
THREE RETAIL STORES TODAY!

IBM COMPATIBLE FLOPPY DRIVES

Double Sided-Double Density
360K

- | | |
|--------------------------------------|--------|
| Tandon Full Ht. w/IBM Logo | *59.95 |
| MPI Half Height | 69.95 |

QBUS SASI HOST ADAPTER.

•Made by Xebec •Brand New!!
\$35.00

YOUR COMPLETE ELECTRONIC SUPPLY STORE!

LITTON TRACKBALL

- 250 CYCLES PER REVOLUTION
- 5 VDC OPERATION
- SPEC. SHEET INCLUDED

\$49⁹⁵

- NEW!! TI 787 Terminal —
 - 120 CPS Thermal Printer
- 110 to 9600 BAUD MODEM: Bell
212, 103 + Racal 3400 Compatible.
• Acoustic/Direct Connect
• RS-232 Interface, Includes Manu-
als, Cables & Paper.

\$ 249.00

ANDERSON/JACOBSEN Stand-
Alone MODEM— • 1200 Baud Racal
3400 Compatible • Acoustic/direct Con-
nect • • Originate Only • INCLUDES
MANUAL + CABLES.

\$ 29.95

3 CONVENIENT LOCATIONS:

HSC Electronic Supply of Santa Rosa
6819 S. Santa Rosa Ave.
Cotati, CA
(707) 792-2357

HSC Electronic Supply
5549 Hemlock St.
Sacramento, CA
(916) 338-2545



827 EVELYN AVE., SUNNYVALE, CA 94086

WE SHIP C.O.D.

MAIL ORDERS

Store Hours:
Mon-Fri 8:00-7:00
Saturday 9:00-5:00

Call NOW! (408)732-1573

TERMS: Minimum order \$10. California residents add 7% sales tax. Prepaid orders send freight COD or call for charge. Shipping will be added to credit card and C.O.D. orders. Prepaid orders over \$100 use money order or certified check. Please do not send cash. Some items limited to stock on hand. Prices subject to change.

Reader Service Number 11

(Listing continued from page 38.)

```
/* This function displays a byte as ones and zeroes in the lower
right corner of the CGA screen, without calling printf(). */

void display_byte(unsigned char byte)
{
    void putc_at_location(char, int, int, unsigned char);

/* you don't HAVE to give variable names in a declaration, but it's not
as clear if you don't. */

    int i, column = 64; /* "line" and "column" are the starting locations */
    const int line = 24; /* The compiler barfs if we try to change a constant.*/
    for( i = 7; i >= 0; i--){ /* start with the high bit and work down */
        putc_at_location(byte & (1 << i) ? '1' : '0', column++, line,
            LIGHT_GREEN_CHAR | RED_BACK);
        /* See SERIAL.C for a description of the "?" ternary expression. */
        putc_at_location(' ', column++, line, BLACK_CHAR | RED_BACK);
    }
}

/* Puts a character and its attribute anywhere on the screen. */
void putc_at_location(char ch, int x, int y, unsigned char attribute)
{
    pokeb(SCREEN_BASE, ((y * SCREEN_WIDTH) + x) * 2, ch);
    pokeb (SCREEN_BASE, ((y * SCREEN_WIDTH) + x) * 2) + 1, attribute);
}

/* SERIAL.C: Turbo C program to write to a TTL LS164 and read
from an LS165 through a parallel printer card. Bruce Eckel,
Eisys Consulting 1987. You can create either a stand-alone
program for testing, or link the compiled file into the
"CLOCKINT.C" project using CLOCKINT.PRJ. */

#undef test /* #define test creates a stand-alone test program
from this file */
#define BASE 0x238 /* I modified my $21 Microsphere card so it's in an
unused address space (no collisions with other
printer cards) */

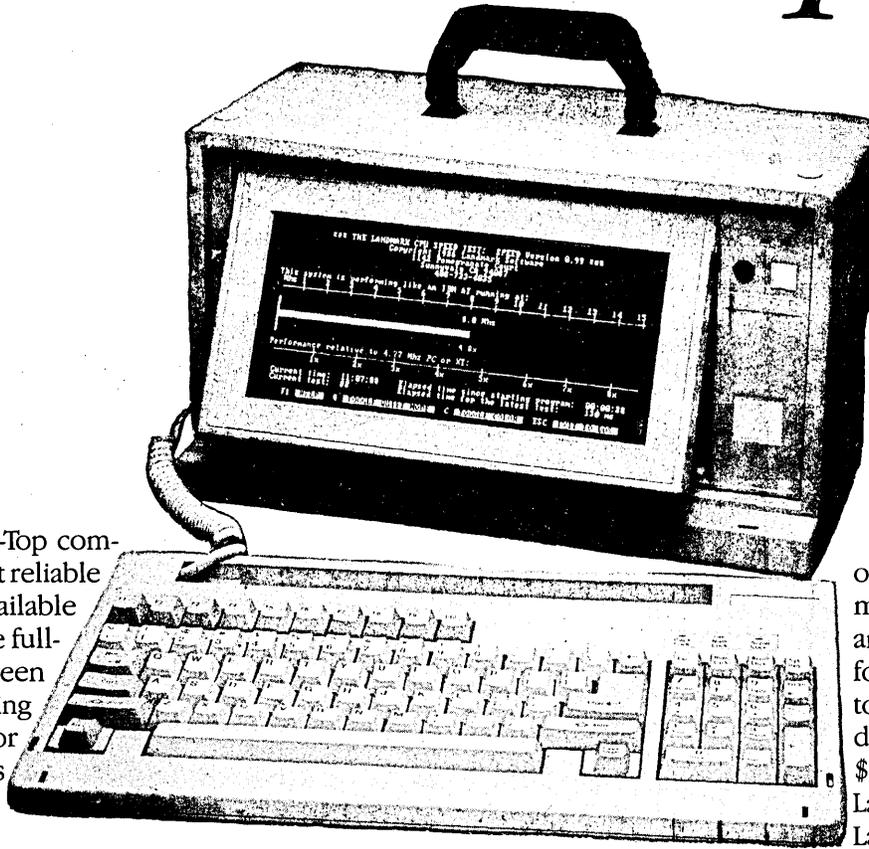
#define BIT(I) (unsigned char)(1 << I) /* this should be evaluated
at compile time so there won't be any overhead at
run time. The "(unsigned char)" is called a "cast"
(like casting into a mold); it forces the result to
be that data type (in this case, one byte long and
don't mess with the high bit). */

/* These definitions make the code much easier to follow (and
thus less prone to error). "ORing" with a bit sets that bit to
"1", while "ANDing" with the complement of a bit sets that bit to
zero. I'm being sort of tricky here: BASE is the printer data
address and can be read from as well as written to, so I read it,
change a single bit, and write it out again so the other bits
aren't changed and can be used for other applications. You can't
do this with the other printer port locations (BASE+1 and
BASE+2). It would probably be more proper to use global memory
locations to preserve the state of all three. */

/* The output clock for the LS164: */
#define LOWER_CLOCK_O output(BASE, inport(BASE) & ~BIT(0)) /* pin 2 on DB-25 */
#define RAISE_CLOCK_O output(BASE, inport(BASE) | BIT(0))
/* Changing the output data bit for the LS164: */
#define DATA_0 (inport(BASE) & ~BIT(1)) /* pin 3 on DB-25 */
#define DATA_1 (inport(BASE) | BIT(1)) /* Load and shift control for the LS165 */
#define LOAD_LS165 output(BASE, inport(BASE) & ~BIT(2)) /* pin 4 on DB-25 */
#define SHIFT_LS165 output(BASE, inport(BASE) | BIT(2))
/* The input clock for the LS165: */
#define LOWER_CLOCK_I output(BASE, inport(BASE) & ~BIT(3)) /* pin 5 on DB-25 */
#define RAISE_CLOCK_I output(BASE, inport(BASE) | BIT(3))
```

(Listing continues on page 42.)

The Ultimate Lap-Top



The McTek 286B Lap-Top combines the fastest, most reliable AT motherboard available with the most visible full-size LCD lap-top screen on the market. Running at a switchable 8 or 10 MHz, it includes a 20MB hard disk, 3½" floppy drive, parallel & serial ports, Award 3.01 bios, 640k, turbo indicator LCD & mouse interface. The screen is a fantastically readable,

electroluminescently backlit, 80-column by 25-line LCD with adjustable intensity and screen-angle. It's as readable as a CRT. You can also plug in a digital or analog

color monitor or a digital or composite monochrome monitor. Included also is an external 5¼" floppy port for reading and converting to 3½" disks (5¼" external drive w/case, power supply: \$179 when purchased with Lap-Top). The McTek 286B Lap-Top comes fully assembled with our one-year parts & labor guarantee, and sells for an amazing, complete price of only **\$1899!**

3 MB On-Board AT!

Our McTek 286A is the most integrated AT-compatible to date. It utilizes the highly regarded Chips & Technology chip set, and includes memory upgradable *on board* to 3 megabytes. No more worries about speed compatibility with expanded memory cards! The 8/10MHz, 0-wait state McTek 286A runs at 11.5 Norton SI, and an effective 13.2MHz on the Landmark test. Serial, parallel & game ports are all standard *on board*. With Award 3.01 bios, 640k, 200W power supply, Samsung amber monitor with Hercules-compatible controller, locking case, AT-style keyboard, 1.2MB drive, 20MB Seagate. Assembled & fully tested, with a full one-year warranty. Get in on the most advanced AT-compatible on the market, at the lowest price ever offered! **\$1399!!**

XT Turbos & Supers

640k 4.77/8MHz and 4.77/10 switchable XT turboboards; two 360k floppy-disk drives with controller; one parallel, one serial and one game port; AT-style keyboard; clock, FCC-approved slide-case; eight slots; Hercules-compatible graphics card; amber monitor w/base; fully assembled and tested; one-year parts *and* labor warranty.

\$649 XT Turbo 4.77/8MHz Complete
\$699 Superturbo 4.77/10MHz Complete

McTek Systems, Inc. • 2316 4TH Street • Berkeley, CA 94710 • 415-843-0714



DISK DRIVES	PRINTERS	MONITORS	PC/XT	PC/AT	MISC.
Fujitsu 360k\$75	Citizen CD 120 \$159	Samsung amber\$ 79	640k TurboMothrbrd....\$ 85	McTek286 6/8/10/12MHV. 299	Kingtech Portable Computer
Fujitsu 1.2MB.....\$99.	Citizen CD 180 \$189	TVM EGA color 399	10MHz TurboMothrbrd.... 95	Baby McTek 286B-AT	Kits: XT/AT (power supply,
Teac\$79	HPLASAR Serial2..... \$1799	TVM RGB color 289	Multi I/O w/disk contrir...59	8/10 O-wait339	case, keyboard, monitor)
Teac 1.2MB \$105	Epson LX-800..... \$219	NEC Multisync 559	640K RAM card 39	McTek 286A O-wait 3MB \$380/410
Toshiba 3½" 720k \$119	Toshiba 321 XL..... \$559	Sony Multiscan 650	2MB Expansion card.....115	4 ports on board 449	Eprom burner 4-socket ... 139
Floppy controller\$22	Call for prices of other brands	HGC-compat.mono card .55	RS232 2-port card 35	3MB Multifunction card ..125	Lap-Top Kits899
20MB Hard Disk Kit.... \$289		Color graphic card49	4-serial port card 95	2MB Expansion card125	AC power center25
30MB Hard Disk Kit.... \$319		EGA Paradise 159	Game I/O card 15	Multi I/O card 59	AC power strips 15
ST-225\$239			384k Multifunction card... 69	Locking slide case..... 65	Diskette file box 9
ST-238\$269			FCC-app. slide XT case...28	200W power supply 79	Printer or serial cable 8
ST-4038\$529			150W power supply 59	AT-style keyboard55	Ventura Publishing
ST-251 40MB..... \$429			XT keyboard 52	WD HD/floppy controller.149	Software \$489
	MODEMS	MOUSE			
	Easydata Int. 300/1200 ...\$85	Logimouse C7.....\$ 75			
	Taiheho external 3/12... \$105				
	Everex 2400 external... \$195				

If You Don't Have WindowDOS 2.0, You're Wasting Time!!

"When Baba Ram Dass said "Be here now, remember," designers of hard disk utilities should have paid heed. A powerful manager like XTREE can track files and subdirectories and execute DOS commands, but it isn't memory resident. Handy pop-up DOS commanders like PopDOS may be here now, but they lack the power of a full-fledged disk manager. After much meditation, the developers of WindowDOS 2.0 have come up with the best answer yet to the guru's paradox.

Until now, the closest thing to a real RAM-resident disk manager was version 1.0 of WindowDOS. It offered a full-screen pop-up menu and could rename, copy, and delete files. But it couldn't move files, format disks, or rename subdirectories—which XTREE can. Now version 2.0 is here, and it's a winner. Its RAM resident (using less than 50K) but offers all the power of a nonresident disk manager."

—Patrick Marshall, WindowDOS 2.0 Product Review, PC World, May, 1987

Once you've experienced the convenience of instant access to DOS commands, you'll never be satisfied with returning to DOS to list files, format disks, or copy, rename, or erase files. Nor will you be happy with a DOS shell, because shell programs are just as inaccessible as DOS when you are using an application program. **Only one program combines memory-residency with the power of a full-featured disk manager: WindowDOS Version 2.0.**

Features Not Found In DOS

- ◆ Sort directories in 8 ways--or not at all
- ◆ Copy, erase, and move groups of files
- ◆ Find any file in seconds
- ◆ Display default directory of any drive with a single keystroke
- ◆ Display graphic tree
- ◆ Global copy & erase commands
- ◆ Copy function prompts you to insert another disk when necessary
- ◆ Display hidden files and subdirectories
- ◆ Display file contents in various formats and page forward/backward
- ◆ Display Wordstar files in readable format
- ◆ Unique RAM Environment function shows name, size, location, and interrupts of every program in memory
- ◆ Rename subdirectories for instant reorganization
- ◆ Hide and unhide subdirectories
- ◆ See and change file attributes
- ◆ Send control codes to printer
- ◆ Switch default printer
- ◆ Password "lock" your system
- ◆ Set AT Real-Time Clock
- ◆ 5-minute screen-blanking function
- ◆ Input response macros

Enhances These Functions

- ◆ Format disks (faster than DOS)
- ◆ Make and erase subdirectories
- ◆ Copy, rename, and erase files
- ◆ Copy files to printer or COM ports
- ◆ Display disk free space and other media information
- ◆ Check and set the time and date

Benefits

- ◆ **Saves Time**—No waiting to exit or reload programs. Instant access to DOS functions whatever your current task. Easily saves 10 or more minutes a day.
- ◆ **Comprehensive**—Broad range of commands, including many not supplied by DOS. Satisfies the needs of both new and advanced users.
- ◆ **Simplifies DOS**—No need to remember exact DOS commands. Intuitive interface and "point and shoot" design saves keystrokes and prevents mistakes. Group file "tagging" avoids the drudgery of repetitive commands.
- ◆ **Security**—Capability to hide/unhide subdirectories, password "lock" a computer, and check for unwanted programs in RAM helps secure data and prevent unauthorized access.

WindowDOS 2.0 Addresses "RAM Cram" Like No Other Program!!!

1. Designed specifically to be loaded first, unlike most memory-resident programs that insist on being loaded last.
2. Uses a hot key combination that does not have an associated ASCII value—prevents key conflicts with other programs.
3. Unique RAM Environment function lets you monitor the locations, memory costs, and interactions of all programs in memory, including the currently running program. Great for power users/developers.

Other Information

- ◆ Not copy protected
- ◆ Uses only 51K of memory
- ◆ Supports EGA & Hercules
- ◆ Runs memory-resident or as a stand alone program
- ◆ Uninstall command
- ◆ PC/XT/AT/100% Compatibles
- ◆ **Order Today--Only \$49.95**

WindowDOS Associates • Box 300488-C • Arlington, Tx 76010 • 817-467-4103

(Listing continued from page 42.)

```
void print_byte(unsigned char byte)
{
    int i;
    for(i = 7; i >=0; i--)
        byte & (1 << i) ? printf("1 ") : printf("0 ");
    /* See below for description of the ":" ternary expression. */
    printf("\n");
}

#endif defined(test) /* You can put things after an "#endif" as notes to
                      yourself; the compiler ignores them. */

/* And now, the function definitions: */
void outbyte(unsigned char byte)
/* Send a byte to the serial-to-parallel chip (LS164) */
{
    int i;
    RAISE_CLOCK_O; /* This is just for the benefit of the initial call. */
    for(i = 7; i >= 0; i--){ /* I start with 7 because that bit goes out first */
        LOWER_CLOCK_O;
        /* Now present the data at pin 1 of the DB-25 */
        outport(BASE, byte & (1 << i) ? DATA_0 : DATA_1); /* Outputs are
                                                             inverted */
        /* The "ternary expression" returns the value to the left of the ":"
        of the question is true (i.e. non-zero) and the value to the right
        of the ":" if the question is false (i.e. zero). It's very nifty
        and compact, since it not only performs an "if-then-else," but it
        also returns a value. */

        RAISE_CLOCK_O; /* clock the data in */
    }
}

unsigned char inbyte()
/* read a value from the parallel-to-serial chip (LS165) */
{
    int i;
    unsigned char result = 0;
    RAISE_CLOCK_I; /* Clock should be high before loading data into the chip */
    LOAD_LS165; /* Move the data at the inputs into the flip-flops */
    SHIFT_LS165; /* Enable shifting out of the data */
    for(i = 7; i >= 0; i--) { /* bit 7 comes out of the LS165 first */
        /* Bit 7 is present at the output immediately when the chip is loaded,
        so we have to pluck it off (into pin 12 of the DB-25) before we do
        any clocking. */
        result |= ((inport(BASE+1) & BIT(5)) ? 1 : 0) << i;
        /* form result: The ternary expression returns 1 if bit 5 of i/o
        address BASE+1 is a "1" and 0 if it's a zero. This value is
        shifted into the appropriate bit position by the "<< i".
        result |= is the same as saying: result = result |, so the bit is
        ORed into the result. This way, we poke each serial bit into the
        correct place in the parallel result. */
        LOWER_CLOCK_I;
        RAISE_CLOCK_I; /* The next bit appears on the rising edge of
                       the clock. */
    }
    return result;
}

/* COLORS.H: definitions for CGA screen characteristics and colors */
#define SCREEN_BASE 0xb800 /* base address of color graphics card (and EGA
                           in color graphics mode */

#define SCREEN_HEIGHT 25
#define SCREEN_WIDTH 80
#define SCREEN_CHARS (SCREEN_WIDTH * SCREEN_HEIGHT * 2)
/* number of chars and attributes in a screen */

#define BIT(I) (unsigned char)(1 << I) /* A bit mask macro. The compiler
                                       evaluates it, so it doesn't cost anything. */

/* Make a complete attribute by ORing a CHARACTER type with a
   BACKGROUND type */
```

(Listing continues on page 46.)

ZRP/M™ creates

Z280®

CP/M®2.2 compatible

IBM PC

ZRP/M is an operating system combined with a Z280 emulator. Either standalone or with DOS present, ZRP/M provides the solid base of a genuine operating system reliably distinct from the facade created by an MSDOS interface. All 2.2 system and CBIOS calls are supported, 56.5k TPA, file date and time stamping, fast virtual disk, iobyte redirection, terminal emulation, color console display, auto relog, COM path, SAVE anywhere, single key phrase recall, built-in access to DOS drives. SETDISK redefines a drive to any of over 80 CP/M formats. System disk with manual \$129. Shipping \$5 (\$10 nonUS)  



118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

TURBO C QUICK C LET'S C DESMET C DATALIGHT C ECO-C
LATTICE C MICROSOFT C AZTEC C COMPUTER INNOVATIONS C

NEW --- Limited time offer.

Peacock System's CBTREE

Object library for only \$29!

Our FULL COMMERCIAL VERSION of CBTREE in object library format is being offered for the amazingly low price of \$29.

CBTREE provides you with easy to use functions that maintain key indexes on your data records. These indexes provide you with fast, keyed access, using the industry standard B+tree access method.

Everything you need to fully utilize CBTREE in your applications is included. The CBTREE source code can be purchased later at any time for the \$70 difference. Example source programs and utilities are included FREE.

CBTREE source library	\$99
Object library only	\$29

This limited time offer is simply too good to refuse. Peacock's standard ROYALTY FREE, UNCONDITIONAL MONEY-BACK GUARANTEE, AND FREE TECHNICAL SUPPORT applies to this offer.

To order or for additional information
call (703) 356-7029 or (703) 847-1743 or write:



PEACOCK SYSTEMS, INC.
2108 GALLOWS ROAD, SUITE C
VIENNA, VA 22180

Trademarks: Turbo C (Borland); Quick C (Microsoft); Let's C (Mark Williams); DeSmet Software); Datalight (Datalight); Lattice C (Lattice); Microsoft C (Microsoft); Aztec C (Manx Software); Computer Innovations C (Computer Innovations); Eco-C (Ecosoft, Inc).

Reader Service Number 20

For Sale! T-SHIRTS from

THE MICRO TECHNICAL JOURNAL

MICRO CORNUCOPIA

Bend, Oregon



S-M-L-ExL \$6.95

Phone 1-800-888-8087

Do It Yourself Micro Ad Only \$99

THE MICRO TECHNICAL JOURNAL
MICRO CORNUCOPIA

See Page 94.

(Listing continued from page 44.)

```
#define BLUE_CHAR BIT(0)
#define GREEN_CHAR BIT(1)
#define RED_CHAR BIT(2)
#define INTENSE BIT(3)
#define BLUE_BACK BIT(4)
#define GREEN_BACK BIT(5)
#define RED_BACK BIT(6)
#define BLINKING BIT(7)

#define BLACK_CHAR 0
#define CYAN_CHAR (GREEN_CHAR | BLUE_CHAR)
#define MAGENTA_CHAR (RED_CHAR | BLUE_CHAR)
#define BROWN_CHAR (RED_CHAR | GREEN_CHAR)
#define WHITE_CHAR (RED_CHAR | GREEN_CHAR | BLUE_CHAR)
#define GRAY_CHAR (INTENSE | BLACK_CHAR)
#define LIGHT_BLUE_CHAR (INTENSE | BLUE_CHAR)
#define LIGHT_GREEN_CHAR (INTENSE | GREEN_CHAR)
#define LIGHT_CYAN_CHAR (INTENSE | CYAN_CHAR)
#define LIGHT_RED_CHAR (INTENSE | RED_CHAR)
#define LIGHT_MAGENTA_CHAR (INTENSE | MAGENTA_CHAR)
#define YELLOW_CHAR (INTENSE | BROWN_CHAR)
#define BRIGHT_WHITE_CHAR ( INTENSE | WHITE_CHAR)

#define BLACK_BACK 0
#define CYAN_BACK (GREEN_BACK | BLUE_BACK)
#define MAGENTA_BACK (RED_BACK | BLUE_BACK)
#define BROWN_BACK (RED_BACK | GREEN_BACK)
#define WHITE_BACK (RED_BACK | GREEN_BACK | BLUE_BACK)
#define GRAY_BACK (INTENSE | BLACK_BACK)
#define LIGHT_BLUE_BACK (INTENSE | BLUE_BACK)
#define LIGHT_GREEN_BACK (INTENSE | GREEN_BACK)
#define LIGHT_CYAN_BACK (INTENSE | CYAN_BACK)
#define LIGHT_RED_BACK (INTENSE | RED_CHAR)
#define LIGHT_MAGENTA_BACK (INTENSE | MAGENTA_CHAR)
#define YELLOW_BACK (INTENSE | BROWN_BACK)
#define BRIGHT_WHITE_BACK ( INTENSE | WHITE_BACK)
```

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)
QUANTITY ONE PRICES SHOWN for OCT. 25, 1987

OUTSIDE OKLAHOMA: NO SALES TAX

640K MOTHERBD UPGRADE: Zenith 150,
IBM PC/XT, Compaq Portable & Plus; hp Vectra

DYNAMIC RAM			
1Mbit	1048Kx1	100 ns	\$29.50
1Mbit	256Kx4	120 ns	32.00
51258	* 256Kx1	100 ns	6.95
4464	64Kx4	150 ns	3.60
41256	256Kx1	80 ns	5.95
41256	256Kx1	100 ns	5.15
41256	256Kx1	120 ns	3.95
41256	256Kx1	150 ns	3.50
41264	+ 64Kx4	120 ns	5.25
EPROM			
27C1000	128Kx8	150 ns	\$37.95
27C512	64Kx8	200 ns	15.50
27256	32Kx8	250 ns	5.75
27128	16Kx8	250 ns	5.50
STATIC RAM			
43256L-12	32Kx8	120 ns	\$11.50
5565PL-15	8Kx8	150 ns	3.30

* 2-PORT VIDEO RAM
* STATIC DRAM FOR 386 COMPAT
8087-2 80287-8 80387-16
\$160.00 \$245.00 \$510.00

68000 NOW!

TinyGiant 68000 Single Board Computer

The HT68K TinyGiant is a great little 68000 single board computer. It uses only +5 and +12 volts and has all of these features: Two Serial Ports, 1 Parallel Port, Expansion Bus, 128K RAM - Expandable on Board to 512K. Floppy Disk Controller, uses MS-DOS Disk Format. 5.75" x 8.0", Fits on a 5 1/4" Drive. K-OS ONE Operating System and Software.

K-OS ONE 68000 Operating System Package

Get the K-OS ONE operating system for your 68000 hardware. With it you can read and write MS-DOS format diskettes on your 68000 system. Included in the package are: K-OS ONE Operating System Source Code, Editor, Assembler, HTPL Compiler, Sample BIOS Code.

HT68K TinyGiant with K-OS ONE . . .	\$395.00
K-OS ONE Operating System Package	\$50.00
HT - Forth Language	\$100.00
Edit Toolkit - HTPL Source & Manual	
Line Editor, Screen Editor	
and Text Formatter	\$50.00
Lizard Land - An HTPL Adventure Program	
with Source Code	\$15.00

Write or call today for more information and a free 68-KNEWS letter.

Order Now: **Hawthorne Technology**
VISA/ MC /COD 8836 Southeast Stark
(503) 254-2005 Portland, OR 97216

OPEN 6 1/2 DAYS, 7:30 AM-10 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 \$4/1 lb Fr: P-1 \$10.50/2 lb
MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts μ P ∞
MICROPROCESSORS UNLIMITED, INC.
24,000 S. Peoria Ave., (918) 267-4961
BEGGS, OK. 74421

No minimum order. Please note that prices are subject to change. Shipping & insurance extra, & up to \$1 for packing materials. Orders received by 9 PM CST can usually be delivered the next morning, via Federal Express Standard Air @ \$4.00, or guaranteed next day Priority One @ \$10.50!

Reader Service Number 37

Reader Service Number 34

C CODE FOR THE PC

source code, of course

C Source Code

Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	\$400
CQL Query System (SQL retrievals plus windows)	\$325
Greenleaf Data Windows (windows, menus, data entry, interactive form design)	\$315
Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	\$300
GraphiC 4.0 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	\$275
Vitamin C (MacWindows)	\$200
resident C (TSRify C programs)	\$165
Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	\$160
Greenleaf Functions (296 useful C functions, all DOS services)	\$160
Essential C Utility Library (400 useful C functions)	\$160
Essential Communications Library (C functions for RS-232-based communication systems)	\$160
PC/IP (CMU/MIT TCP/IP implementation for PCs)	\$100
B-Tree Library & ISAM Driver (file system utilities by Softfocus)	\$100
The Profiler (program execution profile tool)	\$100
Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	\$100
Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	\$100
QC88 C compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	\$90
CBTree (B+tree ISAM driver, multiple variable-length keys)	\$80
ME (programmer's editor with C-like macro language by Magma Software)	\$75
Wendin PCNX Operating System Shell	\$75
Wendin PCVMS Operating System Shell	\$75
Wendin Operating System Construction Kit	\$75
EZ_ASM (assembly language macros bridging C and MASM)	\$60
Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	\$50
Heap Expander (dynamic memory manager for expanded memory)	\$50
Make (macros, all languages, built-in rules)	\$50
Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	\$50
Coder's Prolog (inference engine for use with C programs)	\$45
PC/MPX (light-weight process manager; includes preemption and coroutine packages)	\$45
Biggerstaff's System Tools (multi-tasking window manager kit)	\$40
TELE Kernel (Ken Berry's multi-tasking kernel)	\$30
TELE Windows (Ken Berry's window package)	\$30
Clisp (Lisp interpreter with extensive internals documentation)	\$30
Translate Rules to C (YACC-like function generator for rule-based systems)	\$30
6-Pack of Editors (six public domain editors for use, study & hacking)	\$30
ICON (string and list processing language, Version 6 and update)	\$25
LEX (lexical analyzer generator)	\$25
Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	\$25
C Compiler Torture Test (checks a C compiler against K & R)	\$20
Benchmark Package (C compiler, PC hardware, and Unix system)	\$20
PKG (task-to-task protocol package)	\$20
A68 (68000 cross-assembler)	\$20
Small-C (C subset compiler for 8080 and 8088)	\$20
tiny-c (C subset subset interpreter including the tiny-c shell)	\$20
Xlisp 1.5a (Lisp interpreter including tiny-Prolog in Lisp)	\$20
List-Pac (C functions for lists, stacks, and queues)	\$20
XLT Macro Processor (general purpose text translator)	\$20
C Tools (exception macros, wc, pp, roff, grep, printf, hash, declare, banner, Pascal-to-C)	\$15

Data

DNA Sequences (GenBank 48.0 of 10,913 sequences with fast similarity search program)	\$150
Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	\$60
Webster's Second Dictionary (234,932 words)	\$60
U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	\$35
The World Digitized (100,000 longitude/latitude of world country boundaries)	\$30
KST Fonts (13,200 characters in 139 mixed fonts: specify T _p X or bitmap format)	\$30
USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	\$20
NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	\$15
U. S. Map (15,701 points of state boundaries)	\$15

The Austin Code Works

11100 Leafwood Lane

Austin, Texas USA 78750-3409

Free surface shipping on prepaid orders

Voice: (512) 258-0785

BBS: (512) 258-8831

Email: FidoNet 1:382/12

MasterCard/VISA

Reader Service Number 4

Taming The PC's Serial Port

Not Just Any Port In A Storm

Serial ports are not trivial pursuits. Here Russ tries to talk directly to his serial ports. How he does it and why are the subjects. His serial port test routine is dynamite - we've already used it here at Micro C to verify that one of our cards has a bad serial controller.

As a result of recent experiences with the serial port on an IBM PC AT, a number of descriptive words for the port have come to mind. Most of them aren't printable.

I encountered this curious beast while working on the Physiologic Acquisition and Telemetry System (PATS) at the JHU Applied Physics Lab. PATS is part of the Large Primate Facility, a joint U.S./French mission to be flown on NASA's Space Shuttle in 1992.

We plan to use two IBM 80286 machines (such as the PS/2 Model 50) as PATS Flight Support Stations aboard the Shuttle. We'll select a machine next year, and it might be a portable (battery-powered) 80C286 machine if a workable one's available.

The job of the computers will include system control and data acquisition via a command/telemetry link. Part of the interface work we are doing involves communicating with the command/telemetry link at 4800 baud via one of the AT's serial ports.

In the process of developing the system, I've learned a lot about writing serial port software.

Speed Limitations & Coding Methods

Please note that speed depends on lots of things: the type processor, its clock speed, etc. Therefore, nothing I'm about to say is absolute.

That said, my first conclusion is that there is an upper limit of 1200-2400 baud when using BASIC. The exact upper

limit depends upon how you are carrying out the communication, and how much other processing, calculating, displaying or file storage, you have to do.

If absolutely all you're doing is outputting or receiving data to or from RAM, you might beat 2400 baud. But probably not.

Even assembly language is limited to 4800 baud unless your program is talking directly to the hardware. DOS's serial port services— Function Interrupt (21H) or the ROM BIOS Serial Port Interrupt (14H)—are just too slow for 9600 or 19,200.

Fairly Fast Serial Functions

Of course, the highest level assembly language interface is DOS Function Interrupt 21H. With this interrupt, you'll use either the auxiliary (serial) input and output services (03 and 04, respectively), or the default auxiliary file handle (03).

Interrupt 21H is a pain for two reasons. First, it starts losing incoming characters at 3600 baud on 4.77 MHz machines. Second, there is no way to check the status of the serial port or to detect I/O errors (incorrect or lost characters).

Despite all its drawbacks, 21H is the only serial interface that's valid for all compatibles. Machines made by manufacturers other than IBM may have ROM BIOS structures and Port Address arrangements different than IBM's.

(Editor's note: Actually, compatibility isn't much of a problem, only the incompatibles like the Sanyo 555 or other early kluges have non-clone BIOSs. The reason that Award, Phoenix, and others have put so much work into their ROMs is to make them look precisely like IBM's.)

Fairly Faster Serial Functions

You can shortcut some of the delays in Interrupt 21H by directly calling ROM BIOS Serial Port Interrupt 14H.

This gives you two big advantages.

You can initialize the port with interrupt 14H. And, it's faster.

It isn't fast enough, however, to communicate reliably at 4800 baud (on a 4.77 MHz machine) if the machine also has to do some file management. (Part of the problem is that the BIOS's serial driver isn't interrupt driven.)

Furthermore, we discovered that this second level doesn't work with PATS. A quick look at the ROM BIOS listing shows that it expects to be talking to a modem or printer. In the absence of handshaking, 14H throws up.

So we decided we'd have to write code that talked directly to the 8250. The software is thus hardware specific and, in our case, will probably work only on IBM hardware.

Not only will it work only on IBM hardware, it turns out that it works only on *some* IBM hardware. And this leads me to the subject of apparent incompatibilities among various versions of IBM equipment.

(Editor's note: If anything, all the 8088 and V20 based clones we've tried have appeared electrically identical to the XT. In fact, as alike as the ROMs are, the hardware appears even more so. We've tried Russ's test routine on a number of systems, including one running an 8 MHz 80186. The routine didn't flinch.)

The serial controller chip used in AT serial cards is not totally compatible with the XT's 8250, nor is the AT environment identical to the XT.

This means that chip level software written for the PC or XT may not work on the AT. And, vice versa, XTs will probably have trouble with AT specific code.

We wrote the first version of the PATS software on a standard PC, then tried to run it on an AT with the 8250-based serial board from the PC.

We got a number of errors - some

fairly consistently, some only occasionally. As soon as we substituted a real AT serial port board, and made a few changes to the software, the errors went away.

I can't explain it all, but I came to one conclusion. The 8 MHz AT is significantly faster than the PC, so you cannot reliably do two consecutive OUTs to a register.

For example, the following sequence won't work:

```
mov dx,register_address
mov al,something
out dx,al
mov al,something_else
out dx,al
```

But the following will:

```
mov dx,register_address
mov al,something
out dx,al
jmp $+2
mov al,something_else
out dx,al
```

We tried using a NOP, but it didn't do the job so we used the JMP. The NOP takes 3 clock cycles, the near JMP takes 15.

In case you haven't run into the "jmp \$+2" command, it simply means "jump two bytes from the beginning (\$) of this instruction." It is, therefore, functionally the same as an NOP. (It is not documented in any of my assembly language books, not even Intel's *iAPX 86/88, 186/188 User's Manual*.)

You'll find quite a few of these JMP \$+2 commands sprinkled around in the AT's BIOS. My theory is that many, if not most, of them are there because of timing problems. It seems a shame to have to slow the 80286 machine down. Oh, well.

Anyway, we have written our first

version of communications software for PATS at the hardware level. For the second version, we'll probably write a special device driver.

Both versions could be subjects for future articles, but next is a piece of diagnostic software that came about as we tried to figure what in tarnation was going on with our serial ports.

You'll find quite a few of these JMP \$+2 commands sprinkled around in the AT's BIOS. My theory is that many, if not most, of them are there because of timing problems.

Serial Port Diagnostic

Our development of this elementary serial port diagnostic was definitely a case of necessity. We had blown out six, count 'em, six, serial port boards. And in the brief periods when one would work, we would get errors that didn't make any sense.

We had no idea where the problem was. There were four main candidates: 1) the external interface circuitry to which the serial port was connected; 2) the cable that connected the two, including sub-possibilities of incorrect wiring and open/shorted connections; 3) the

serial port board itself; and 4) the communications software.

It turned out there were problems in three of the four areas. If we hadn't had a simple diagnostic, we'd probably still be troubleshooting.

SEPOD

The routine, named SEPOD, for Serial Port Diagnostic, is far from complete and far from ideal. So before discussing it in detail, I'll lay out my idea of an ideal diagnostic.

The ideal routine should check absolutely everything: all baud rates, control lines, parity settings, and interrupts. In other words, it should verify each bit of each of the ten registers in the 8250. And, it should check the line transmitter and receiver chips.

It should also be easy to run and require a minimum of external hardware (ideally, it would run itself and require no external hardware).

If your serial port hardware is functioning properly, it should tell you so. If it isn't, it should tell you exactly what the problem is - in plain English.

Oh yes, and it should run on any machine that runs MS-DOS or PC-DOS, whether true blue or completely clone.

Since this is the real world, SEPOD is not ideal. It is easy to run and, if it's included in your autoexec.bat file, it will run automatically. Also, it tells you in plain English where your trouble may lie.

It does, however, require an external connector (with three pairs of pins jumpered) plugged into the serial port. It doesn't check all baud rates, and it doesn't test interrupt operation. As they say in the textbooks, the rest of the baud rates and the interrupts are left to the reader.

Confidence

Overall, I'd estimate that the successful running of SEPOD probably gives you a 95-97 percent degree of con-

fidence that the serial port hardware is okay. If it doesn't succeed, chances are probably 97-99 percent that you have a hardware problem.

The routine runs at 4800 baud, which is faster than most applications software. In particular, it should be more than adequate for checking serial ports used with 2400 baud modems.

The routine is written at two of the three levels discussed earlier. The portion which checks general port operation uses BIOS level calls, so it should run on all really compatible machines.

The internal-loop part of the routine addresses the hardware registers directly. You'll need to substitute your serial port hardware register addresses for those in the program (which are for an IBM PC/XT) if yours are different.

Figure 1 contains the assembly language source. Please read the notes before you assemble and link it.

(Editor's note: All the code for this issue, including SEPOD.ASM and SEPOD.EXE, is available on the Micro C RBBS (503) 382-7643, and on the Issue #39 disk. Order the Issue #39 disk from Micro C for \$6 (U.S. subscribers) and \$8 (non-subscribers and foreign). Ask for 5 1/4" MS-DOS or Kaypro, or 3 1/2" MS-DOS format. Price includes postage.)

When running the program, the external check is performed first. You must have a DB-25 connector with three pairs of pins jumpered for this test to succeed. Jumper pins 2 and 3, 4 and 5, and 6 and 20. (For serial ports with non DB-25 connectors, jumper the TX-RX, CTS-RTS, and DTR-DSR lines.)

The second part of the test, the internal loopback, runs if there's an error in the external check. If you don't know your hardware register addresses, you can disable this part of the program by changing the "jna intlp" statement to "jmp exit."

So, for the cost of a DB-25 connector (about \$1.50) and a few minutes soldering (three jumpers), you can check out your serial port.

Post-Mortem

So what was causing our problems? First, we had some minor problems in CMOS-to-TTL level conversion that were contributing to the early demise of chips on our interface circuit. Second, we needed to slow down some of our code by adding "jmp \$+2" commands.

The third, and most insidious, problem was found only after running the test repeatedly while jiggling, pushing and pulling cables. We had an inter-

Figure 1 - Sepod Assembly Listing, Tests COM1 at 4800,8,1,N

```

; NOTE: For this test to be successful, the following pins on the
; serial port connector MUST be jumpered together: Jumper
; pins 2 and 3, jumper pins 4 and 5, and jumper pins 6 and 20.
;
; With pins jumpered as indicated above, the program will first
; perform an external looping test that checks the overall
; performance of the serial port by transmitting and receiving a
; series of ASCII characters. If they are received correctly
; they will be displayed on the screen, followed by a message that
; the port checks out completely. If, and only if, an error is
; detected, an error message is displayed, and an internal loop
; test is performed that checks the 8250 UART by looping back
; internally. If the first (external) part fails, and the second
; (internal) part runs, the trouble may be in the transmitter
; and/or receiver chips, or with one of the handshaking lines
; (CTS/RTS or DSR/DTR). If the second part also fails, the
; 8250 UART may be bad.
;
; NOTE: The overall external loop check will work on all PC's and
; clones, since it uses BIOS interrupts. The internal loop
; check, however, addresses hardware registers directly, and
; therefore must have the correct register addresses. Make
; sure that the five statements with asterisks in the comment
; fields all have the correct register addresses for your
; machines. If you cannot determine these addresses, you
; may want to disable the internal check part of the program
; by changing the statement "jna intlp" to "jmp exit" on the
; line with "#####". The register addresses are valid for
; the IBM PC, XT, AT and PS/2 Model 50.
;
data segment public 'data'
signon db 0dh,0ah,'SEPOD Serial Port Diagnostic for COM1:',0dh,0ah
db 'Copyright 1987, Russell C. Eberhart',0dh,0ah,0dh,0ah
db 'You MUST have the following pins jumpered on the serial
port',0dh,0ah
db 'connector for this test to be successful:',0dh,0ah
db 'Jumper pins 2 & 3, 4 & 5, and 6 & 20.',0dh,0ah,0dh,0ah
db 'Testing overall port performance (external loop):',0dh,0ah,'$'
char db '0'
errcnt db 0h
crlf db 0dh,0ah,0dh,0ah,'$'
exgood db 'The COM1 serial port checks out completely OK.',0dh,0ah,'$'
ingood db 'The internal looping check of the 8250 is OK.',0dh,0ah
db 'You may have a problem with the transmitter',0dh,0ah
db 'and/or receiver chips, or with the CTS/RTS',0dh,0ah
db 'and/or DTR/DSR handshaking lines.',0dh,0ah,0dh,0ah,'$'
exerr3 db 'ERROR: Transmitted character not received.',0dh,0ah,'$'
exerr1 db 'ERROR associated with character received.',0dh,0ah,'$'
exerr2 db 'ERROR: Character received not same as sent.',0dh,0ah,'$'
intchk db 0dh,0ah,'Testing internal loopback performance:',0dh,0ah,'$'
bomb db 0dh,0ah,'The internal looping check fails. You may have a bad
8250 UART.',0dh,0ah,'$'

data ends
;
; dgroup group data ;define data group, loaded last
;
; stack segment para stack 'stack'
; db 256 dup(0) ;reserve 256 bytes for stack
; stack ends
;
; code segment byte public 'code'
; assume cs:code, ss:stack, ds:dgroup
;
; main proc far
;
; ; Set up data segment
start: mov ax,data ;set up data segment
; mov ds,ax ;into DS
;
; ; Initialize communications port
; mov ah,00 ;function 0 = initialize port
; mov al,0c3h ;4800,8,N,1 (0a3h for 2400)
; mov dx,0 ;use COM1
; int 14h ;serial port service
;
; ; Print sign-on message
; lea dx,signon ;load address of sign-on message
; mov ah,9 ;print string service
; int 21h ;DOS function
;
; ; Clean out receiver data and shift registers
; mov cx,2 ;set up to loop twice
; lean: mov ah,2 ;set up read character
; mov dx,0 ;use COM1

```

```

; int 14h ;serial port service
; loop clean ;do twice
;
; Check external loop, transmit & receive 80 characters
txwait: mov ah,03h ;serial port status request
        mov dx,0 ;use COM1
        int 14h ;serial port service
        test ah,20h ;transmitter hold register empty?
        jz txwait ;not empty
;
        mov al,char ;start with "0" character, then increment
        mov dx,0 ;use COM1
        mov ah,01h ;write character
        int 14h ;serial port service
;
wait: mov cx,0ffff ;set up 1-second loop
      mov ah,3 ;status request
      mov dx,0 ;use COM1
      int 14h ;serial port service
      and ah,01h ;character ready yet?
      jnz rcv ;yes, receive it
      loop wait ;no, try again (for 1 second)
      jmp error3 ;if no success, receive char error
;
rcv: mov ah,2 ;yes, read char function
     mov dx,0 ;use COM1
     int 14h ;serial port service
     and ah,08eh ;error check (09eh? break also?)
     jc error1 ;handle external receive error
;
     cmp al,char ;compare sent with received char
     jnz error2 ;handle garbled transmission error
     mov bx,0 ;page 0
     mov ah,0eh ;TTY display service
     int 10h ;video I/O
;
     inc char ;next character
     cmp char,7eh ;last character?
     jnz txwait ;no, send another char
     lea dx,crlf ;yes, load cr-lf string
     mov ah,9 ;print string service
     int 21h ;DOS function
     lea dx,exgood ;load "success" message
     mov ah,9 ;print string service
     int 21h ;DOS function
;
exit: mov ah,04ch ;terminate process service
      int 21h ;DOS function (END program)
;
; External receive error reflected in AH
error1: lea dx,exerr1 ;load line status error after rcv message
        jmp errprn ;jump to print error message
;
; Received character not same as that sent
error2: lea dx,exerr2 ;load sent received error message
        jmp errprn ;jump to print error message
;
; Received character not available (bit 0 not 1)
error3: lea dx,exerr3 ;load xmttd char not rcd error message
errprn: mov ah,9 ;print string service
        int 21h ;DOS function interrupt
        inc errcnt ;add one to the error count
        cmp errcnt,1 ;test to see if previous error
        jna intlp ;if not, check internal loop #####
;
; If internal loop test done with error, print message and exit
        lea dx,bomb ;load total failure message
        mov ah,9 ;print string service
        int 21h ;DOS function
        jmp exit ;go to exit
;
; Set up internal loopback and test
intlp: mov dx,3fch ;modem control register address *****
       mov al,13h ;set INTERNAL loopback
       out dx,al ;do it
;
       mov char,'0' ;re-start with character 0
;
       lea dx,intchk ;load address of internal check message
       mov ah,9 ;print string service
       int 21h ;DOS function
;
intxwt: mov dx,3fdh ;line status register address *****
        in al,dx ;put line status in AL
        test al,20h

```

(Listing continues next page.)

mittent short in the 25-conductor RS-232 connector. *That* is what destroyed six boards. I now have that connector at home, in my Museum of Significant Failures.

I also found out that you can't mix the three levels of serial port programming. For example, setting the internal loopback by setting the modem control register directly and then using BIOS interrupt 14H to send and receive characters won't work. It hangs. Every time. Pick your level and stick with it.

In the process of going through the problems with the serial port, then writing the SEPOD diagnostic, and, finally, writing this article, I've learned more about the serial port than I really cared to.

But, if you would like to explore beyond this article, there's a good chapter on the serial port in the book *8088 Assembler Language Programming: The IBM PC, Second Edition*, by David C. Willen and Jeffrey I. Krantz, published in 1985 by Howard W. Sams, ISBN: 0-672-22400-3. Ray Duncan's *Advanced MS-DOS*, published in 1986 by Microsoft Press, ISBN: 0-914845-77-2, also has good material on the serial port.

■■■

Come to
SOG VII!
 July 14 - July 16
 Bend, Oregon

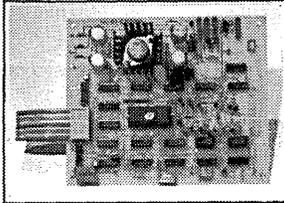
Rafting

Barbeque

Computer tech talk

MS-DOS, CP/M E/EEPROM PROGRAMMING SYSTEM

2708
2758
2716
2516
2532*
2564*
68764*
2816A
2732
2732A



2764
2764A
27128
27128A
27256
27512
27CXXX
2864A
8751*

*SOCKET ADAPTER
REQUIRED -
DIAGRAMS INCLUDED

A FULL FEATURED HARDWARE/SOFTWARE PACKAGE

- FAST PROGRAMMING ALGORITHM
- NO PERSONALITY MODULES REQUIRED
- INSTALL PROGRAM FOR SOFTWARE
- ALL SUPPLIES ON BOARD
- PROGRAMS 26, 25, 21 & 12.5V E/EEPROMS
- LARGE COMPREHENSIVE MANUAL
- USES NO SYSTEM POWER OR CHASSIS SLOT
- STAND-ALONE BOARD
- HIGH SPEED PARALLEL OPERATION
- FIVE LED STATUS/ACTIVITY INDICATORS
- HIGH QUALITY "TEXTTOOL" ZIF SOCKET
- REQUIRES 24 OR 25 VOLT XFMR FOR POWER

PARALLEL PRINTER INTERFACE

CONNECTS TO ANY PARALLEL PRINTER INTERFACE
USES 8 OUTPUT DATA BITS AND THE PRINTER BUSY LINE FOR DATA INPUT

CONTROL PROGRAM COMMANDS

- PROGRAM EPROM(S) FROM DISK FILE
- READ DISK FILE INTO BUFFER
- READ EPROM(S) INTO BUFFER
- VERIFY EPROM IS ERASED
- CHANGE EPROM TYPE
- SAVE EPROM(S)/BUFFER TO DISK
- PROGRAM EPROM(S) FROM BUFFER
- COMPARE EPROM(S) WITH BUFFER
- COPY EPROM(S)
- BUFFER MONITOR MODE (SEE BELOW)

THE BUFFER MONITOR MODE HAS 17 SUB-COMMANDS FOR DETAILED OPERATIONS. THESE INCLUDE: FILL, DUMP, TRANSFER, PROGRAM, READ, VERIFY, EXAMINE, MODIFY, CHECKSUM, BIAS, INSPECT, SINGLE BYTE BURN, LOGICAL OPERATIONS(AND/OR/XOR), SET BUFFER BIAS, HEX ARITHMETIC, ETC.

ASSEMBLED AND TESTED UNIT WITH COMPLETE DOCUMENTATION AND SOFTWARE ON DISKETTE

\$199

PARTS KIT WITH SOFTWARE AND DOC. \$179 BARE BOARD, SOFTWARE & DOC. \$69
SOFTWARE AVAILABLE ON 5 1/4" OR 8" DISK FOR IBM, KAYPRO, & OTHER FORMATS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL:

VISA

ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

MASTER
CARD

CALL OR WRITE FOR MORE INFORMATION -- ADD \$4.00 FOR SHIPPING - \$3.00 COD

Reader Service Number 30

```

;transmitter hold reg. empty?
    jz intxwt        ;not empty, so loop
;
    mov al,char      ;start with "0" char, then incr
    mov dx,3f8h     ;xmitter holding reg. addr *****
    out dx,al        ;send the character
;
    mov cx,0ffh     ;set up 1-second loop
inwait: mov dx,3fdh ;line status register addr *****
        in al,dx    ;put line status in AL
        test al,leh  ;receive error check
        jnz error1  ;handle external receive error
;
        test al,01h ;character ready yet?
        jnz incrv   ;yes, receive it
        loop inwait ;no, loop for up to 1 second
        jmp error3  ;if no success, receive char error
;
incrv:  mov dx,3f8h ;receiver data register addr *****
        in al,dx    ;put received char in AL
        and al,7fh  ;keep only valid lower 7 bits
;
        cmp al,char ;compare sent with received char
        jnz error2  ;handle garbled transmission error
        mov bx,0    ;page 0
        mov ah,0eh  ;TTY display service
        int 10h     ;video I/O
;
        inc char    ;next character
        cmp char,7eh ;last character?
        jnz intxwt  ;no, send another char
        lea dx,crlf ;yes, load cr-lf string
        mov ah,9    ;print string service
        int 21h     ;DOS function
        lea dx,ingood ;load internal "success" message
        mov ah,9    ;print string service
        int 21h     ;DOS function
;
        jmp exit    ;all done, exit
;
main    endp
code    ends
        end start
    
```

SCIENTIFIC SOFTWARE

- SCI-GRAF** \$99.95 Create huge hi-res plots with log or linear scaling. Screen and printer output. Automatic legends and labels. Flexible ASCII input. Works with CGA, EGA, Hercules, and mono cards.
- SCI-GRAF MODULES** \$250.00 Create custom hi-res graphs from within your own programs by linking to our object code. Supports all *SCI-GRAF* features plus plotter output, and more! Microsoft C, Turbo C, and Aztec C versions. No royalties.
- FONTEEDIT** \$49.95 Create custom Greek, math, or other symbols for use with *SCI-GRAF* or *SCI-GRAF MODULES*. Requires IBM compatibility and CGA or EGA.
- SCI-DATA** \$59.95 Perform least squares fits (linear, parabolic, and exponential) and normal curve approximations. Convert between polar and rectangular coordinates. Also supports a variety of scaling transformations. Great complement to *SCI-GRAF*!
- SCI-EVAL** \$49.95 Pop-up scientific expression evaluator, more powerful than other pop-up calculators. Complete expression editing facility. Full range of functions: scientific, statistical, logic. Requires IBM compatibility.

Free shipping on prepaid orders. No credit cards.

MSC Microcomputer
Systems
Consultants

P.O. Box 747, Santa Barbara, CA 93102
(805) 963-3412

Reader Service Number 36

Make this Your Ad

In our March - April Issue.

\$232

6 x Rate

Reserve space by Dec. 15
Ads due by Dec. 20

EMERALD MICROWARE - Your CP/M and MS-DOS Connection

Hard Disks for your Z80 CP/M computer from Emerald Microware and MICROCode Consulting

No other upgrade improves your computer's productivity like a hard disk. We have all the hardware and software to install a hard drive on your Xerox 820, Kaypro, Zorba, or almost any Z80 CP/M 2.2 system.

- HDS Host Board with Winchester Connection software . . . \$ 89.00
- HDS Board with software and WD1002-05 board \$250.00
- WD1002-05 Hard Drive Controller Board \$185.00
- Rodime, LaPine, & Miniscribe hard drives, and XT controller cards - call for prices

The KayPLUS ROM Package by MICROCode Consulting

Get the performance of a Kaypro 10 and more, even on your Kaypro 2. Lets you install up to four floppies and two hard drives, with no software assembly required. Adds features such as automatic screen blanking, type-ahead buffer, boot from hard drive, and quad density support. Includes manual, standard utilities, AND hard disk utilities

- KayPLUS ROM Set \$ 69.95
- KayPLUS ROM Set with QP/M ★★ SPECIAL ★★ \$115.00

QP/M by MICROCode Consulting, CP/M 2.2 compatibility with outstanding performance.

QP/M adds features such as automatic disk relogging, drive/user selection from colon, 31 user areas, drive search path, and transparent time/date stamping; all in the same space as CP/M. Installs from a convenient customization menu, no software assembly required. Bootable disks available with CBIOS for Kaypro, Xerox, & BBI.

- QP/M Operating System, complete ready to boot \$ 80.00
- QP/M without CBIOS (installs on any Z80 system) \$ 60.00

MicroSolutions - Software and hardware to link CP/M and MS-DOS

UniForm-PC by MicroSolutions \$ 64.95
This program allows you to read, write, copy, and format diskettes for over a hundred CP/M and MSDOS computers on your PC, XT, or AT, including 8", 96 TPI, high density, and 3 1/2" formats (with optional hardware). Once installed, UniForm stays memory resident so you can use your standard DOS commands and other programs directly on your original diskettes.

- Uniform for Kaypro and other machines \$ 64.95

UniDOS by MicroSolutions \$ 64.95
UniDOS program uses the NEC V20 CPU chips to actually RUN your favorite 8080 CP/M programs on your PC. Use UniDOS with UniForm-PC, and you can run them directly from your CP/M format diskettes. All standard CP/M BDOS calls are supported.

- UniDOS w/UniForm and V20 chip \$135.00

★★★ NEW ★★★ **UniDOS Z80 Coprocessor Board** . . . \$179.95
This 8Mhz. Z80H half-card runs your Z80 and 8080 code programs at LIGHTNING speed on your PC or AT. Functions just like the UniDOS program, except NO V20 or emulation mode is required to run your programs.

The CompatiCard by MicroSolutions \$169.95
This half-card floppy controller allows you to run up to four 8", 5 1/4"(standard, 96 TPI, or high density), or 3 1/2" disk drives on your PC/XT. With the CompatiCard and the UniForm-PC program you can format, read, and write almost all CP/M and MSDOS disk formats.

- CompatiCard with UniFORM-PC ★★★ SPECIAL ★★★ \$225.00

MatchPoint-PC by MicroSolutions \$169.95
This half-card allows you to read and write to NorthStar hard sector, Apple DOS, PRODOS, and Apple CP/M diskettes on your PC. INCLUDES a copy of the UniForm-PC program, as well as utilities to format disks, copy, delete, and view files.

★★★★★★ SUPER SPECIALS!! ★★★★★★★

PC-Mastercard by Magnum Computer

Up to 1.5 Megabytes of RAMDISK and PRINT SPOOLER (or boost your system to 640k), with serial, parallel, game ports, and real time clock! This is one of the BEST multi-function cards on the market. Can use mixed banks of 64K and 256K chips. Comes with memory manager, ramdisk, spooler, and diagnostic software.

- PC-MASTERCARD (0k installed) \$ 69.95
- call for pricing on 384k and 1.5M boards

Turbo Editor Toolbox by Borland \$ 19.95
Ever wanted to add text editing to your Turbo Pascal application, or write a word processor that does things the way that YOU want? Comes with source for two sample editors, modules for windowing, multi-tasking, and many other options. Requires PC with Turbo Pascal 3.0.

COPY II PC by Central Point Software \$ 19.95
Stop worrying about your copy protected disks. COPY II PC lets you back them up, so you can keep going when your master disk can't.

PC Tools Deluxe
by Central Point Software \$19.95

InfoStar Manual Set \$ 18.00
Genuine DataStar and ReportStar manuals in MicroPro three vol. set

- 8" Generic Diskettes - Ten Pack \$ 7.50
- 8" Generic Diskettes - Hundred Pack \$ 65.00
- Single sided, SD or DD, with Tyvec sleeves

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

Four Device Printer/Data Switch \$ 49.95
Quality with economy. These four port boxes can be used with either RS232, or IBM parallel (DB25) printer cables.

- IBM style Parallel Printer Cable \$ 12.00

300/1200 Baud Internal Modem \$117.00
Half-card, Hayes compatible, auto-answer, auto dial

CT-6260 MULTI I/O - FLOPPY DISK CONTROLLER \$ 69.95
Half-card parallel, COM1, optional COM2, game port, floppy disk controller, real time clock, with manual and cables.

TWO DRIVE FLOPPY DISK CONTROLLER \$ 29.95
Half-card floppy controller with cable.

Double Density for the Xerox 820-1 by Emerald Microware and MICROCode Consulting

Run up to four 5 1/4" (48 or 96 TPI) and 8" drives at once. Get support for all standard printers, mini-monitor functions, autoboot capability, 19 built in disk formats, and banked ROM-BIOS for more TPA. Software compatible with Kaypro and Xerox 820.

- 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
- X120 Bare Board only ★★★ CLOSE-OUT SPECIAL ★★★ \$ 15.00
- or two for \$ 25.00, five for \$ 50.00



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

Call or write for our complete catalog of parts and accessories for the Kaypro, Xerox 820, and PC/AT's. Full repair services available for Kaypro, Morrow, Xerox, disk drives, and most clones.

Prices subject to change without notice. Include \$4.00 shipping and handling, \$7.00 for COD, call for Blue Label charges. VISA and Mastercard accepted. 30 day money back guarantee on all products.

LISP Programming

From The Bottom Up

An Introduction To Recursion, An Introduction To....

Your recursions have been thrown for a loop, your expert software's playing dumb, and C's no longer a challenge? You've got problems. Big problems. It may be time for you to stand up before your fellow engineers and LISP.

For over 20 years, LISP has been THE LANGUAGE of Artificial Intelligence, and until recently (and the sudden popularity of Prolog) it had this arena to itself. But there are signs that the two will influence each other and both will evolve and prosper as a result.

Recently (and only recently) LISP has become readily available on microcomputers, and even now isn't all that well-understood by most PC programmers. This article is intended to introduce LISP to those of you who haven't gotten around to experimenting with it yourselves.

LISP programs are famous for their complexity and subtlety. At the same time, one of LISP's great attractions is that a beginner can start programming after grasping only a handful of concepts.

Key Concepts—Lists & Functions

The emphasis in AI applications isn't so much on arithmetic or numeric processing as on manipulating information. The inventor of LISP, John McCarthy, recognized that putting information into lists is very handy for processing.

"LISP" stands for LIST Processing language, and the first concept is that data is typically structured as lists of items.

Programs in LISP consist essentially of lists of items enclosed in parentheses. The first item in the list is a symbol denoting an operation to be carried out

on the remaining items.

We usually describe this by saying that the first item is a FUNCTION which is APPLIED to the other items (called ARGUMENTS). Alternately, you might say the function MAPs the arguments into a result. This is the reason we sometimes see LISP described as a "functional programming language."

Let me illustrate. Consider what happens when we boot up a LISP interpreter (for example, the public domain version XLISP, which is available on Micro C disk #MS19). We get the prompt sign ">" on the screen. If we enter the following items after the prompt—

```
( + 3 4 )
```

The function + is applied to the arguments 3 and 4 to produce the result, 7.

We can describe LISP as working continually in a READ-EVAL-PRINT loop. It reads what we type in, evaluates it, and prints out the result of evaluation. Then it goes back to waiting for us to give it something else to evaluate and print.

Let's look at a couple more examples of functions mapping arguments into results. For historical reasons, the function for extracting the first element of a list is called "CAR" (standing for Contents of Address Register). It's standard in most LISP implementations (along with other standard functions like +, -, * and /).

Suppose we give XLISP a list with "CAR" as the first item:

```
> ( car ' ( A B C D E ) )
```

Then, "CAR" is treated as a function to be applied to the arguments which follow it. In this case, there's only one argument, the list of five items A through E. So, applying the function, the

system produces the response A on the screen.

Why put the quote in front of the list? When LISP looks at any list, it will assume the first item is a function to be applied to the following items.

In this case, A is not a function to be applied to B, C, D, and E. The quote mark tells the system to treat the whole list as data. To put it another way, we're telling the system NOT TO EVALUATE the list (A B C D E), but leave it as it is.

(Note: I've used lower case characters for function names because some versions will accept only lower case.)

You can see the difference between evaluation and preventing evaluation by using a quote in the following example.

If we type in (+ 3 4), LISP responds with a 7. But if we type in '(+ 3 4), LISP responds with (+ 3 4); that is, it returns the list unevaluated. We can force an evaluation with the built-in function "EVAL":

```
(eval '(+ 3 4) ) produces a 7.
```

CDR And CONS

Two other built-in functions which have a central role in LISP are CDR and CONS.

(1) "CDR"—(again, an historical name), drops the first item of the list leaving the rest of the list as the result. For example, asking for (cdr '(P Q R S)) will produce (Q R S).

Remember, the CDR of a list is always a list, too. So, what is (cdr '(F))? Clearly, the result here is an empty list, and in most LISP implementations, the result would appear on the screen as "nil."

(2) "CONS"—accepts two arguments, placing the first argument in the first item position in the second argument.

For example, (cons 'A '(B C D)) will produce the result (A B C D). Again, if

we ask for the result of (cons '(A B) '(C D)), the result we get will be: ((A B) C D).

The point to note here is that in this case we've asked the system to put the list (A B) as the first item of a list which has two elements C and D. So the result is a list with three items, in which the first item is itself a list. We can have lists within lists within lists...

Defining Our Own Functions

A powerful feature of LISP is that it makes it easy to define our own functions, to supplement the built-in functions like + and CDR.

I'll illustrate: Suppose we don't like the name CAR for extracting the first item of a list, and would prefer to call the first item by a more natural name of HEAD of the list. Here's how we would teach LISP to recognize the new name. We type in:

```
( defun   head(whatever) (car
  whatever) )
```

defun stands for DEfine FUNction. We're telling the LISP system that when we want it to apply the function "head" to any argument "whatever," we mean the same thing as "CAR" of that argument.

The system acknowledges this with a message, which varies with the particular LISP implementation.

Now we can type: (head '(L M N)), and the system will respond with L. If we'd tried this without first giving the definition using DEFUN, the system would have responded with an error message.

Similarly, we can define whatever we want. For example:

```
(defun tail(L) (cdr L) )
```

After providing the system with this definition, we can ask for (tail '(X Y Z)) and get back (Y Z) as the result.

In the two examples above, I've used only one argument in the definition. However, I could have used any number of arguments to define a new function. We'll see some examples after looking at the next key concept.

Branching

Let's say we want to input a number which represents the temperature in degrees Fahrenheit, and we want the program to CLASSIFY this input using the following criteria: If the temperature is more than 90 degrees, it should be classified as HOT. Otherwise, if it's 90 degrees or less, but more than 70 degrees, it should be classified as WARM. Otherwise, if it's 70 degrees or below, but above 40 degrees, it should be classified as OK. Finally, any temperature of 40 degrees or below should be classified as COLD.

I've coded this in Figure 1.

Figure 1 - Hot & Cold LISP

```
(defun   tempclass(x)
  ( cond
    ( ( > x 90) 'HOT )
    ( ( > x 70) 'WARM )
    ( ( > x 40) 'OK )
    ( t        'COLD ) ))
```

(1) I use "DEFUN" to DEfine a FUNCTION "tempclass," and this function will operate on one argument x as input, to produce a result.

(2) The keyword "COND" is used to indicate that this is a function which produces different results depending on tests of conditions.

(3) Following the keyword COND, there is a series of lists consisting of two items each. In each of these pairs, the first item is a logical condition; if the condition is true, the action is taken. Otherwise, the program tests the next pair.

So, if the condition (> x 90) is true,

(i.e., x is greater than 90), the program will output the message HOT and end. Otherwise, it will go and test the next condition (> x 70) ... and so on. So we see that by typing in (tempclass 92) we would get the message HOT; (tempclass 62) would produce the message OK and so on.

(4) The last condition-action pair has just the condition "t" which stands for TRUE. In other words, if the tests of all the conditions have not produced a true result, the program reaches this last pair and takes the final action. If, for instance, we type (tempclass 30) the program will display COLD.

(5) The parentheses have got to match, as we have put lists within lists; but these can be indented or even written on separate lines to increase readability.

LISP has other conditional instructions, but COND is a good starter.

Recursion—A Power Tool In LISP

Recursion is a key concept in AI programming, in LISP as well as in Prolog.

A recursive program is one which invokes or calls itself, but with some change in the arguments. To put in another way, part of a recursive program has a pattern similar to the whole program.

The reason a recursive description of a structure or a recursive program doesn't become a circular definition is this: we describe a pattern as having a subpattern similar to the whole. This means this subpattern has a part which is similar, and this in turn has a similar subpattern. As we go on, we finally get to a part which is extremely basic, the simplest possible example of the recurring pattern. A list is a good example.

A list is a pattern which has a part, the CDR or tail, which is also a list. This tail has a subpattern, with its CDR also a list, and this in turn has a CDR which is a list ... and finally, when the list is

down to one item, it has a CDR which is an empty list, the simplest list.

Similarly, to see how a recursive program "recurses" to a simple case, let's look at a classical example: a recursive program for joining two lists to produce a combined list. We want a function called, say, "JOIN," so that if we say:

```
( join ' ( a b c ) ' ( d e f g ) )  
we get the result:  
  
( a b c d e f g )
```

We can define this operation recursively as follows:

To join list#1 and list#2:

set aside the CAR of list#1,

join the CDR of list#1 to list#2
to produce list#3,

Then CONS the CAR of list#1 to
list#3 to produce the result we
want.

We see the typical recursive structure. Joining list#1 and list#2 requires the actions for joining, but with some change in the arguments. We're using the CDR of list#1, not the whole of list#1.

We can see that joining, using CDR of list#1, will in turn involve using its CDR ... until we reach the stage where the first argument is an empty list. And this brings us to the simple case, joining an empty list to a second list. The result is just the second list. We can express this in LISP; see Figure 2.

This uses some of the concepts we've already seen. The first condition tested is whether the first list is "null" (empty). If it is, the result has to be the second list. "null" is a built-in function in most versions of LISP, including

Figure 2 - Joining Two Lists

```
(defun join (first second)  
  
(cond  
  ( (null first) second )  
  ( t ( cons (car first)  
            (join (cdr first) second) ) )  
))
```

XLISP.

Recurring a list always ends with an empty list, so a test for "null" is really

handy. If the "null" test fails, we see the next condition action pair has "t" as the condition, because there's nothing else to test. And the action to be taken is to CONS the CAR of the first list to the join of the CDR-of-the-first-list with the second list, as we've done above.

This function also illustrates a general rule for writing recursive programs. We should test first to see if we have reached the simplest case (or BOUNDARY CASE) to see whether we need more recursion. We then put in the code for handling the non-boundary cases to get them down to the simple case.

This operation of joining two lists to produce a combined list is a very useful programming tool, so most versions of LISP provide this as a built-in function, usually with the name "append." I've used it as an example because it illustrates the principles of recursion nicely. You might like to experiment with it and other more complex recursive ideas (like Quicksort) in order to experience the power of recursion.

In Closing

I'd like to summarize a few distinctive features of LISP:

(1) It's a FUNCTIONAL programming language. If programs are written as functions which are evaluated and return a value, then we can write functions in which one of the arguments is a function, and the value of this second function is used by the first function, and so on "recursively" to any depth. Thus our programs can be tested as small modules—which are part of larger modules—which are part of larger modules—and so on.

(2) Recursion and the flexible structure of lists work together when processing symbolic information.

(3) Because programs are functions which are lists of items, and data can also be lists of items, a LISP program can use a LISP program as its data!

Smart Programs

Let's take this one more step. How about letting programs create or modify other programs? Modify themselves? Ultimately, when we say a person learns, don't we mean he modifies his own program?

In other words, LISP has the structure to learn and change. This is a major reason it has become a key tool in Artificial Intelligence work. The breakthrough expert systems programs in the 1960's were written in LISP.

As I said earlier, LISP has many

powerful features, and many complex programs have been written in this "old" language. I haven't said anything about the many similarities between LISP and Pascal (or BASIC)—operations like assigning a value to a variable, writing a loop, and so on. For a more in-depth discussion of LISP and LISP programming, check out one or more of the following books:

Abelson, H. and others: *The Structure and Interpretation of Computer Programs* (MIT Press, 1985).

Betz, David: *An XLISP Tutorial* (pages 221 ff in BYTE, March 1985).

Fladung, B.J.: *The XLISP Primer* (Prentice-Hall, 1987).

Friedman, D.P. & Felleisen, M.: *The Little LISP* (Science Research Associates, 1986).

Narayanan, A. & Sharkey, N.E.: *An Introduction to LISP* (Halsted Press/John Wiley, 1985).

Touretzky, David S.: *LISP: A Gentle Introduction to Symbolic Computation* (Harper & Row, 1984).



CIRCUIT BOARDS

PCB-Edit... creates multi-layered PCB's with ease. Included are solder mask and legend ink support, plotter -- printer drivers
MUCH MORE \$99.95

PCB-Shop... will build double sided, plated thru holes, circuit boards from your artwork or PCB-Edit files for only \$1.00 per square inch (single quantities) with no set up charges.

ANALOGIC ... the 32 channel logic analyzer for the IBM PC. Has 16 bit trigger word, 80 nano second sample time, for only: FULL UNIT BARE BRD

\$399.95 \$99.95

MOVER... the two axis stepper motor driver controlled by your printer port.

only: FULL UNIT BARE BRD
\$179.95 \$49.95

ANALOGIC

**PHONE (602) 458-4065
BOX 3228, Sierra Vista, AZ 85636**

Reader Service Number 39

Now it's
easier than
ever to order
from Micro
Cornucopia!

Use your
Visa or
Mastercard
and
call:

1-800-888-8087
ORDERS ONLY
9 AM - 5 PM, Monday - Friday - Pacific Time

XT CLONE SYSTEMS

(One YEAR guarantee on system)

Turbo Mother Board 4.77 and 10 MHz
640 K Ram installed on board
Serial, Parallel, Game Ports
Clock/Calendar AT Style Keyboard
Color Video Board (CGA) Monochrome Opt
150 Watt Power Supply Flip Top Case
ABOVE WITH 2 FLOPPY DISK DRIVES \$ 649.00
WITH 1 FLOPPY AND 20 MEG \$ 899.00
WITH 2 FLOPPY AND 20 MEG \$ 979.00

Assembled and Tested for 24 Hours

AT TURBO SYSTEM

AT COMPATIBLE MOTHER BOARD WITH BIOS
8 MEG AND 12 MEG SWITCHABLE SPEED
512K RAM INSTALLED UP TO 1024 ON BOARD
WA2 HARD DISK/FLOPPY DISK CONTROLLER
MONOCHROME GRAPHICS VIDEO WITH PRINTER
1.2 MEG OR 360 K FLOPPY
220 WATT POWER SUPPLY AT CASE
AT KEYBOARD SET UP DISK
ONE YEAR WARRANTEE ON SYSTEM \$1095.00
EGA UPGRADE FOR ABOVE \$ 100.00
512K UPGRADE (1024 INSTALLED) \$ 50.00
5339 KEYBOARD UPGRADE \$ 30.00

HARD DISK DRIVES

20 Meg Seagate ST4026 (for AT) \$ 495.00
30 Meg Seagate ST4038 (for AT) \$ 595.00

Does NOT include controller

Color Monitor RGB (CGA) \$ 275.00
Color Monitor RGB (EGA) \$ 385.00
Monochrome TTL (Green) \$ 110.00
Monochrome TTL (Amber) \$ 120.00
EGA Color Video Card \$ 150.00
MS DOS 3.1 \$ 50.00

CITIZEN PRINTERS

MODEL 120D 120 CPS 9" \$ 200.00
MODEL MSP-10 160 CPS 9" \$ 300.00
MODEL MSP-15 160 CPS 15" \$ 400.00
MODEL MSP-20 200 CPS 9" \$ 350.00
MODEL MSP-25 200 CPS 15" \$ 500.00
MODEL 35 35 CPS LETTER QUALITY \$ 500.00
ALL PRINTERS COME WITH CABLE

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

Parallon Message Passing:

Using MS-DOS And C To Communicate In Parallel

We took a close look at the Transputer in issue #38. This article covers the software side of a parallel system based on a very different processor, the 8088-compatible V20. In fact, this is a real, expandable parallel processor system based on boards (each with 8 V20s) which can simply be plugged into a PC/XT/AT or clone.

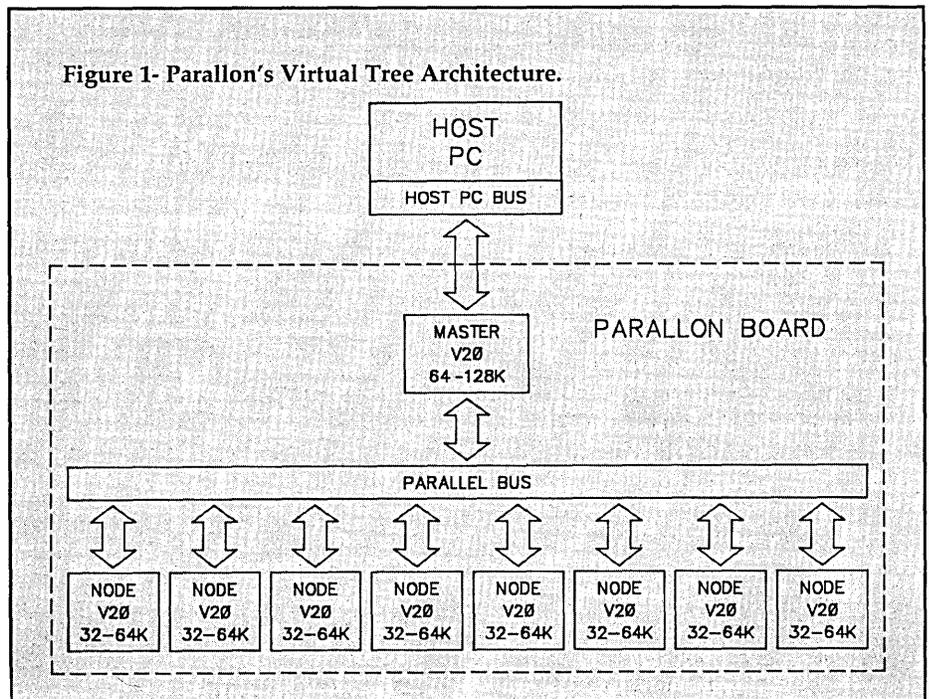
The future of parallel processing depends, more than anything else, on the development of good software. So far, manufacturers of parallel hardware have tried to solve this problem for their users by building high-level parallel language environments which shield users from the underlying parallelism of the hardware.

Unfortunately, this approach places the burden of parallel systems research on the shoulders of a few people, leading to relatively unsophisticated solutions. By contrast, we believe parallel software can best be developed in the same way as today's microcomputer software—through the creativity of independent developers.

Our Parallon 1 board contains 8 node processors, each consisting of an NEC V20 Intel-compatible CPU and 32K or 64K of local no-wait state RAM, a master processor with an NEC V20 and either 32K or 128K of no-wait state RAM, a DMA controller, timers, and I/O ports. The board can be plugged into the slot of a PC, AT, or (optimally) 386 compatible. You can use it alone or add additional boards, each under control of the host.

We've achieved this flexibility by integrating the Intel CPU and IBM bus architectures into a Virtual Tree architecture, a hierarchy of processors shaped like an inverted tree. See Figure 1.

Each processor in the tree can control



up to 16 processors below it. (For instance, the host computer can control up to 16 boards, each board containing up to 16 nodes.)

In the PC-based Parallon system, the PC's built-in processor (the host), is at the top of the inverted tree. At the next level down are the master processors (one on each Parallon board), all accessible to the host over the PC bus. At the bottom are the node processors, accessible to the master over an on-card bus.

Communications

Okay, you're about to ask—how do all these processors communicate?

First, processors at the lowest level of the tree (nodes) do the computing, while all higher levels (masters and host) handle message passing. Second, all tree control functions are top-down between adjacent layers and use memory-mapped registers (leaving the I/O map

free for I/O).

Part of the memory map of each master, for instance, is a window into the full memory map of a node, switchable (using memory-mapped registers) to reach different nodes. Because this is true at every level of the tree, it's possible for the host to access the RAM not only of the masters (i.e., its branches) but also the nodes (its branches' branches)—a very useful feature for debugging.

One implication of this is that since processors (particularly nodes) can't directly access the RAM of other processors at the same layer of the tree, this isn't a "shared-memory" system in the usual sense. Rather, it's a message-passing system like a local area network in which information is passed between processors explicitly as messages—but much faster, over parallel busses instead of serial lines.

Because bus control between layers is top-down, sending messages down the tree is a matter of writing to the next layer down, while messages aren't sent up the tree so much as retrieved by the next upper layer.

Message Passing

The basic message-passing procedure goes like this: when a node has a message to send, it alerts its master (running MPASS.EXE, its messaging program); the master copies the message into its own RAM, checks a destination header, and sends the message down to the destination if that node is also on its board, or alerts the host if it isn't. At that point, the host (running HPASS.EXE) would send the message down to the appropriate master, which in turn would relay it down to the appropriate node or nodes.

Each message automatically takes the shortest possible path from source to destination—messages between nodes on the same board never reach the PC bus, for instance. Because all interconnects are achieved via parallel busses instead of serial ports, interprocessor communications are direct and fast, limited only by bus bandwidth.

To address applications which require the widespread distribution of common data, we've supported a variety of hardware logical-address or "group" modes. While messages of this type occupy the same bandwidth as single physical-address messages going up the tree, when sent down they are broadcast to a group—any or all processors at a given level—simultaneously.

The challenge of designing the system software for the Parallon 1 was to make the system as flexible as possible, without sacrificing ease of use or speed. The system should encourage experimentation, freeing users from the details of inter-processor communication without restricting them. Finally, programmers should be able to write

code once which can be reconfigured for any Parallon system.

Experienced
MS-DOS
programmers can
therefore
immediately start
programming for
the Parallon,
without learning a
new language or
environment.

Tools

Since each node on a Parallon 1 board can execute Intel machine instructions, the Parallon systems software allows programmers to choose from a wide variety of off-the-shelf MS-DOS development tools. Most programmers have a favorite high-level language and environment "tool kit." The Parallon systems software can work with just about any of these because it avoids implementation-dependent ".OBJ" files and uses only standard ".EXE" files, which any MS-DOS development system can generate.

Experienced MS-DOS programmers can therefore immediately start programming for the Parallon, without learning a new language or environment.

Inter-Node Communication

Inter-node communication on the Parallon 1 is available to programmers at the assembly-code level or through function calls which support message-passing. With just a few C-compatible function calls, the programmer can send and receive messages between processing nodes and perform I/O. These function calls are OUP, WAIT, INP, RELEASE, and DISCARD.

The system calls for output are OUP and WAIT. Most operating systems copy output to a system buffer before passing it on to its ultimate destination. OUP allows a program to keep running without copying the message into a buffer, because the master will soon copy the message into its own memory space anyway. However, between the time when a program calls OUP and the master picks up the message, the program must avoid clobbering the message.

When a program wants to reuse the memory it has recently used to send a message, it must WAIT. The program cannot continue until the master has picked up the message. A cautious programmer might call WAIT after every OUP call, but with care, occasional calls will suffice.

The system calls for input are INP, RELEASE, and DISCARD. A call to INP returns to the address of the current incoming message. If the next message isn't in the node's memory yet, it's fetched from the master's memory. Since the nodes have limited memory, a node can only store a few messages at a time.

DISCARD tells the system that the program has finished reading the message. The memory space used by the message is freed, and the next call to INP gets a new message. RELEASE tells the system that the program will not be using the message for a moment. This allows the system to move the message to another memory location if it needs

to create a large block of contiguous memory.

Each message created by OUP has a header containing an address field which can be either a "physical" address, which has four-bit fields to specify one of 16 boards and one of 16 nodes, or a "logical" address, which points to a procedural list of physical addresses.

The address field is very inconvenient to use directly, since it requires the programmer to code for a specific hardware configuration. Instead, we've implemented a system of symbolic addressing. This refers to labels in the source code which are assigned logical and physical addresses at load time, allowing the programmer to experiment with different communication configurations without recompiling.

A user-specified netlist of connections tells the loader how to assign the symbolic addresses. We assume each node is running a separate program module even if the code is identical.

The netlist provided to the loader specifies a series of connections between the symbolic addresses of different modules. For instance, module A's symbolic address "out_right" could be connected to module B's symbolic address "in_left" with the line (A,out_right)-(B,in_left).

In addition to the translation of symbolic addresses at load time, the Parallon system software maintains runtime tables to translate from logical addresses to physical addresses, allowing messages to be broadcast or to be rerouted on the fly. The logical-to-physical tables can be omitted to save space if the programmer doesn't need them.

Optimizing Message Passing

We designed the message-passing system to accommodate different types of algorithms. Some algorithms use many short messages, while others use fewer, larger communications. Sometimes data is sent continuously, other times in bursts.

Short messages aren't as efficient since each message requires some overhead. The overhead can become significant if the message contains only a few bytes.

To minimize the overhead involved in sending short messages, MPASS.EXE running on the master concatenates them into larger messages if the communications load is heavy. When only a few messages are being sent, short messages are sent individually.

If a node receives a continuous stream of messages, its memory space can become fragmented. Programmers can choose a version of INP which compacts memory when necessary, eliminating fragmentation.

The intelligent memory compaction algorithm can usually free the required space with the smallest possible overhead. Memory compaction can move messages unpredictably, but the protocol for locking a message in place is very simple.

The familiar MS-DOS development tools provide a large library of sophisticated DOS I/O calls, but programmers must remember that the code they write will not be running on a PC.

When INP is called, the received message is automatically locked, preventing it from being moved until the module explicitly RELEASEs or DISCARDs the message. The programmer has the option of RELEASEing messages frequently, infrequently, or not at all. Furthermore, fragmentation can be essentially eliminated by specifying a larger message-passing space at load time.

I/O

The message-passing system is very useful for simplifying parallel programming, but by itself it doesn't allow any communication with the outside world. The familiar MS-DOS development tools provide a large library of sophisticated DOS I/O calls, but programmers must remember that the code they write will not be running on a PC. As a result,

DOS system calls and input/output procedures won't work the Parallon.

The Parallon system software provides a simple and flexible alternative to DOS system calls. The I/O facilities rest on top of the message-passing system. Messages representing I/O are sent to and received from device-driver processes running under HPASS on the PC host processor. The links between Parallon modules and device-driver processes can be reassigned, just like the links between Parallon modules.

For example, the Parallon 1 comes with a color version of Conway's game of Life. Each module calculates a subsection of the display screen, and must communicate with the modules which calculate neighboring sections. When the modules are loaded into the Parallon, the programmer can specify that module A sends its right edge to module B's left edge, and vice versa.

The modules' input channels for their initial states can be tied to a device-driver process which reads a file, or to one which accepts keyboard input. Likewise, the modules' output channels can be tied to a display-driver program which relocates or perhaps rotates the subscreen they represent. Neither the modules nor the device-driver processes need know where the messages they send actually go, since HPASS takes care of the details of mapping between symbolic and real addresses.

Wrap Up

The modular and flexible nature of Parallon systems software provides a convenient base for developing applications. As parallel software developers become more familiar with the tricks of parallel programming, they can write higher-level tools which build upon the Parallon systems software.

Tasks which could be automated include assignment of code modules to nodes, decisions about how best to interconnect modules, serializing parallel code (thus eliminating message passing overhead between two small modules), and parallelizing serial code.

Each of these tasks requires a non-trivial level of analysis, but the system certainly can support such complex software.



Software Developers

We need your program!

Do you have a program that's good enough to sell, but don't want the problems or financial risk of producing, typesetting, printing, packaging, warehousing, marketing, distributing and supporting a product?

Why start your own software house?
We've done it for you!

Merlin Publishing Group
is now accepting submissions of micro-computer software for publication.

You get:

- initial cash payments
- generous royalties
- to spend your time programming

But we can't help you if you don't submit. Call or write today for our submission guideline kit.



PUBLISHING GROUP™

1240 Johnson Ferry Place, Suite A10
Marietta, GA 30068
(404) 977-6034

P. S. See us in Atlanta at COMDEX-Spring '88
Reader Service Number 35

**DISK FORMAT
CONVERSION**

XenoCopy-PC

PC-DOS program
lets your PC

\$79.95 + \$5.00 S/H
Sales Tax if CA.



UPS
COD

READ / WRITE /

FORMAT / DUPLICATE

Disks from over 300 other micros

Upgrades available from previous versions
for only \$25.00 Call for Authorization

To Order Contact:

XENOSOFT™

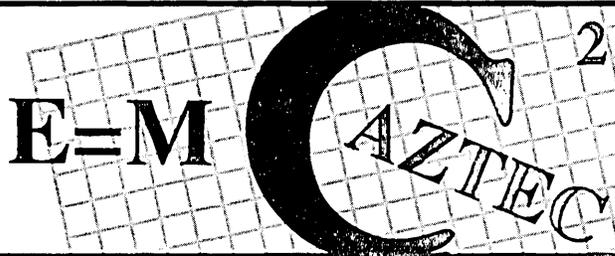
1454 Sixth Street, Berkeley, CA 94710

(415) 525-3113

this was on purpose.

good software is it's own reward

Reader Service Number 38



Genius Begins With A Great Idea...

FREE
2 DAY
DELIVERY

Aztec C86 4.1
New PC/MS-DOS
CP/M-86 • ROM

Aztec ROM Systems
6502/65C02 • 8080/Z80
8086/80x86 • 680x0

Superior performance, a powerful new array of features and utilities, and pricing that is unmatched make the new Aztec C86 the first choice of serious software developers.

An IBM or Macintosh is not only a less expensive way to develop ROM code, it's better. Targets include the 6502/65C02, 8080/Z80, 8086/80x86, and 680x0.

Aztec C has an excellent reputation for producing compact high performance code. Our systems for under \$1,000 outperform systems priced at over \$10,000.

Aztec C86-p.....\$199

• optimized C with near, far, huge, small, and large memory - Inline assembler - Inline 8087/80287 - ANSI support - Fast Float (32 bit) - optimization options • Manx Aztec 8086/80x86 macro assembler • Aztec overlay linker (large/small model) • source level debugger • object librarian • 3.x file sharing & locking • comprehensive libraries of UNIX, DOS, Screen, Graphics, and special run time routines.

Initial Host Plus Target...\$ 750
Additional Targets.....\$ 500
ROM Support Package....\$ 500

Vax, Sun, PDP-11 ROM HOSTS

Call for information on Vax, PDP-11, Sun and other host environments.

Aztec C86-d.....\$299

• includes all of Aztec C86-p • Unix utilities make, diff, grep • vi editor • 6+ memory models • Profiler.

Cross Development

Most Aztec C systems are available as cross development systems. Hosts include: PC/MS-DOS, Macintosh, CP/M, Vax, PDP-11, Sun, and others. Call for information and pricing.

Aztec C86-c.....\$499

• includes all of Aztec C86-d • Source for library routines • ROM Support • CP/M-86 support • One year of updates.

CP/M • 8080/Z80 ROM

C compiler, 8080/Z80 assembler, linker, librarian, UNIX libraries, and specialized utilities.

Aztec C II-c CP/M & ROM...\$349

Aztec C II-d CP/M.....\$199

Third Party Software

A large array of support software is available for Aztec C86. Essential Graphics • C Essentials • C Utility Library • Greenleaf Com. • Greenleaf General • Halo • Panel • PC-lint • PforCe • Pre-C • Windows for C • Windows for Data • C terp • db_Vista • Phact • Plink86Plus • C-tree.

How To Become A User

To become an Aztec C user call 800-221-0440. From NJ or international locations call 201-542-2121. Telex: 4995812 or FAX: 201-542-8386. C.O.D., VISA, MasterCard, American Express, wire (domestic and international), and terms are available. One and two day delivery available for all domestic and most international destinations.

C' Prime

**PC/MS-DOS • Macintosh
Apple II • TRS-80 • CP/M**

These C development systems are unbeatable for the price. They are earlier versions of Aztec C that originally sold for as much as \$500. Each system includes C compiler, assembler, linker, librarian, UNIX routines, and more.. Special discounts are available for use as course material.

C' Prime\$75

Aztec Systems bought directly from Manx have a 30 day satisfaction guarantee. Most systems are upgradable by paying the difference in price plus \$10. Site licenses, OEM, educational, and multiple copy discounts are available.

MANX

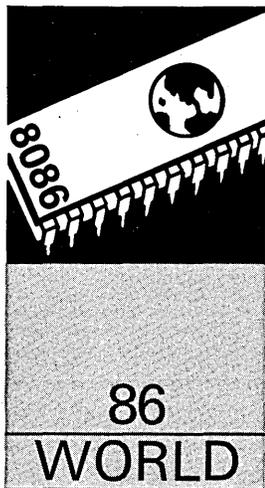
Manx Software Systems
One Industrial Way
Eatontown, NJ 07724

To order or for information call today.

1-800-221-0440

In NJ or international call (201) 542-2121
TELEX 4995812

Reader Service Number 17



The Wrong Side Of The Law

By Laine Stump

Redhouse Press
Merkez PK 142
34432 Sirkeci
Istanbul, Turkey

Want to add a little fun to your dull and dreary existence? Thought of casting off the old routines and trying something new? How about packing up your computer expertise and taking it overseas for a year or two? Laine's been in Turkey for over two years and he's having a....

Hosh Geldiniz! ("You Came With Happiness!"), said the man in the black coat, white shirt, black tie, and twenties style slouch hat as he reached through the jeep's window to shake Sarge's hand.

"Gunaydin!" ("Good morning!"), Sarge replied, obviously too shaken to realize that it was late afternoon. Our friend's backup man stood nervously behind the tractor parked in the middle of the road, shifting from one foot to the other as he tried to make the WWI vintage rifle in his hands seem a bit less obvious.

We were near the top of Hasan Dag, one of the tallest mountains in North Central Anatolia. Our rafting trip on the Kizilirmak River had ended a day early after losing three of our four paddles. So we decided to try and find the dirt road up over Hasan Dag that showed on our aeronautical charts.

"What are you doing up here?" our new friend now demanded. "Why didn't you stop in the village? I'm the mayor and you're supposed to ask me for permission before you come up here! You shouldn't be here!" His manner had changed immensely in the few seconds since he had given us the customary Turkish welcome. He was now visibly shaking, partly from nervousness and partly from the near freezing cold of the late autumn afternoon at 8,000 feet. I began to worry that the nervousness would spread to his henchman hiding behind the tractor.

"Wait a minute," we said. "There weren't any signs. Why were we supposed to stop in your village? We were just driving on the road."

"You just are!" he replied. "Now give me your ID cards. We're going back down to the village and I'm going to call the Jandarma (military police) and see what they have to say. You start down and we'll be right behind you."

Jason started to say something back to the mayor about this being a public place and we could be here if we wanted, but Sarge and I managed to shut him up. We pulled past the tractor and the shivering young moustached man with the rifle twirler's gun and started bumping down the sheep track that had led us into this mess in the first place.

On the way down we discussed our strategy. "Look," said Sarge to everybody, but obviously pointed towards Jason, the resident hothead, "he may be right and he may be wrong, but out here HE'S the law. We'd better be damn polite and do just what he says. I'd appreciate if nobody talked back to him or questioned his authority. I'm sure that he's just as nervous as we are. Let's just tell him we're sorry and we didn't know we had to ask permission. And DON'T show him our maps!"

Back down in the village, we were motioned into a parking space that looked to be the flat roof of somebody's mud and rock dwelling. As the four of us unfolded ourselves from the little Mercedes Jeep, the usual crowd of villagers started to gather.

"Where are you from?"

"We're all Americans."

"Why are you in Turkey? Are you tourists?"

"No, we're working."

"Embassy?"

"No. Well, Sarge works at the embassy, but I work for the Development Foundation of Turkey, Jason teaches theater at the Government Conservatory, and Rich is a freelance journalist. He just came to Turkey a month ago and we were out showing him around."

"You live in Ankara?"

"Yes."

"Why did you come out here?"

"Just to look around."

"How much money do you make???"

Mr. Mayor was still in trying to contact the Jandarma from the only phone in town.

Meanwhile, the conversation had lulled and I decided to start it back up again.

"Is that a Turkish-made rifle?" I asked our moustached, parka clad guard.

"Yes."

"Made in Kirikkale then I guess..."

"Yeah." I didn't feel like asking where the bullets were made or where he had been trained in handling guns.

A man of about 50 years or so, wearing a knitted cap that showed he had made a pilgrimage to Mecca, and obviously of some importance in the village, stepped up to talk to me. "What were you doing up there?"

"We heard that Hasan Dag was the highest mountain around and we wanted to see it up close."

"Well there's a road there, but knowing the correct turns is very critical," he said, showing off his city vocabulary. "You know, there's gold in these mountains. We thought you were up here trying to steal some of it. There were some Germans up here awhile back trying to dig some up and now we're in charge of protecting the mountain."

"Oh. So that's why you were worried! Well, we didn't even know there was gold up here. You don't have to worry about us. See, this jeep is new and we're looking for high roads because we wanted to, uh" ("Hey Sarge! What's Turkish for 'check it out'?" A shrug of the shoulders.) "We just wanted to see if it's a good car or a bad car." Sometimes I don't say things very eloquently, but most of the time I end up making myself clear somehow.

The mayor had come back from his office. "Nobody answered down at the station, so I'm going to take down your names and your license plate number and send it in to them."

"Okay. Look, we're really sorry about the trouble. We just didn't know we were supposed to ask permission."

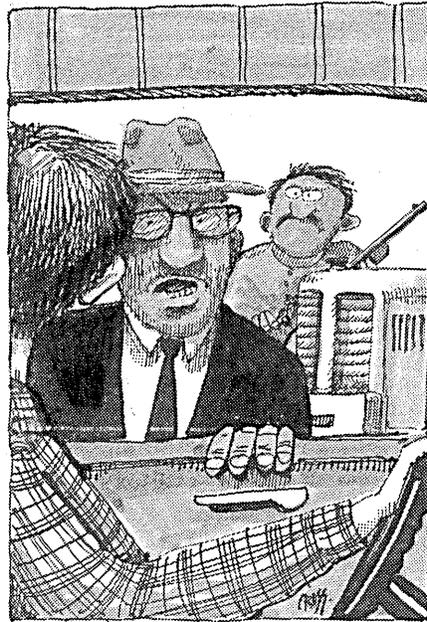
"Don't worry about it. If you come back, just stop in and see me. You can have dinner with us and then we'll go together and I'll show you the mountain."

"That would be really nice. Well, we'd better be going now, it's getting late. Nice to meet you all." We shook hands with every adult male in the crowd of 20 or so that had now surrounded the jeep, climbed back in and started down the rest of the trail leading back to the main highway, waving our goodbyes to the mayor, the 78-year-old retired railway worker, some of the cutest little girls in the world, and our

moustached friend with the antique rifle.

The "Other Side" Of Working Abroad

This is just one of many experiences I've had since I left the U.S. to come to Turkey just over two years ago. Most people who think of working overseas think either of volunteer work—living in a mud hut in a remote village teaching the art of latrine digging to a bunch of illiterate pygmies. Or they think of oil companies—living in a foreigners' compound somewhere in the desert of northern Chad, making \$150,000 a year by working long hours in an inhospitable place with boring scenery and nothing to do for recreation but



throw darts at fading pictures of the Ayatollah.

There is a middle ground, though, and I think that the middle is where the most fun is. It's no fun to be isolated totally from challenging high tech work, but it's also no fun to completely separate yourself from the local population. Why go to a foreign country if you're just going to pretend you're still in America?

The work you do while you are overseas may or may not be satisfying or educational, but the real satisfaction and education comes when you step out the door of your office into the streets of a city where all the signs are written in Martian, the people speak in unintelligible staccato bursts of emotion, and the traffic is reminiscent of the demolition derby at the Deschutes County Fair.

Since I have just ended my two year contract with the Development Founda-

tion of Turkey, I thought it would be kind of fun to reminisce about my two years living "in the middle" and maybe let you in on a bit of the fun to expect if you do the same.

I've been trying for days to find a way of organizing all of this and have had absolutely no success. I guess the best form of organization then is "stream of consciousness." Here goes...

Curiosity

In any country outside of Western Europe, practically any local you meet is going to be extremely curious about you, where you're from, why you're here, if you're married, how you can stand being separated from your family, how much you make. The normal questions.

The first words of Turkish that I learned were in response to these questions. Within weeks I could field the stock series of inquiries. Back in Ankara after the fiasco on Hasan Dag, I popped into a cab, in a hurry to get over to the American Research Institute to install the new winchester in their Multitech PC:

"Say, you're not Turkish, are you? Are you German? Sprechen Sie Deutsch?"

"No, I'm an American. I don't know German."

"Oh. Well you speak Turkish very well. How long have you been here?"

"Two years."

"Very good for two years. Are you a student? Or in the military?"

"No. I work for the Development Foundation of Turkey."

"Ah, yes, the Development Bank of Turkey."

"No, no, the Development FOUNDATION of Turkey."

"Oh, yes. Tell me, is there Development in Turkey??"

Small laugh. "Of course. Slow, but it's there."

"What part of America are you from?"

"I'm from Wyoming."

"Oh, yeah, Miami! I have an uncle there!"

"No, no! Not Miami! WYOMING!"

"Yeah. Nice beaches there. And all those women!! Do you like it here in Turkey?"

"Yes I like it very much."

"What about the food? Are you accustomed to the food yet?"

"Yes. I love the food. Whenever I leave the country I miss it."

"Are you married??"

"No."

"Well, find yourself a nice plump Turkish girl, get married and stay here."

"Nah, I don't want to get married."

"WHAT??? Is your family here?"

"No. They're in America."

"Where in Ankara do you live?"

"Gaziosmanpasa."

"Oh. How much money do you make?"

This is the stock line of questioning. No matter where you are, you can expect the same, with a few variations, like "Which is better, America? Or Turkey?" Or how about, "What do you think of Turkish girls? They're beautiful, aren't they?? Look at that one there! Wow!!!" The cabbie nearly throws us into the oncoming traffic as he stabs his finger at a bulging heap of fat wrapped in a black tent with a face that would make Stephen Spielberg chuck his cookies. "Right, right. Just beautiful." And then there's always this one: "What about this AIDS thing? Aren't you scared?????"

Some days I get tired of answering the stock list. But then, just about when I'm ready to start ignoring them, someone will throw in a new one: "Are American girls really as loose as they say in the papers?????" Or even better: "How long does it take to come to Turkey from America by bus?"

Foreigner

At times it is unamusing to be treated as a foreigner. Growing up in a town with a big tourist industry (Cody, Wyoming), I learned to hate "pilgrims" and "flatlanders" (both are local terms for tourists, mostly of the Eastern and Mid-west variety) nosing around with their pink bermudas, stupid wide-brimmed hats, and big telephoto lenses hanging off the end of their Pentaxes.

Now the tables have turned; now it's ME who's the outsider. I'm very conscious of that Canon hanging around my neck. Many days I take it off and leave it home. I like blending in with the crowds, being unnoticed. I see much more of the "real" Turkey that way.

In Turkey, being a foreigner is sometimes a big advantage. If you can call it an advantage. Several times I have been with a group of foreigners going into a crowded restaurant and watched in embarrassment as the head waiter hustled a table full of Turks into the street in order to make room for us. If a foreign-looking person is wandering around looking lost, it is not uncommon for a Turk who knows a few words of English to come

up and befriend him, doing everything possible to help him find his way. Sometimes they will even accompany the person all the way to his destination, or even invite him to their house for tea.

On the other side of the coin, many merchants consider a tourist an easy mark. I have been offered carpets for \$1,000 that weren't worth more than \$300. One carpet dealer in the Covered Bazaar in Istanbul told me that he charges tourists as much as he thinks he can get out of them. He showed me a flat-weave carpet worth about \$400. "I sold one just like this to a Swedish girl for \$1,500!" he said, quite pleased with himself. Remember: that's not cheating, it's just good business.

Speaking The Language

I enjoy speaking Turkish because it makes me more of a local. No matter

Yes, please,
kind sir! I sell to
you flying carpet!
Yes, please!

how hard I try, though, I still can't dress enough like a Turk to disguise myself completely. This leads to a situation that annoys me more than anything else; shopkeepers who insist on speaking to me in English, German, French, or even Serbo Croatian (Yugoslavian), ignoring the fact that I am speaking back to them in Turkish.

It's fine if they actually CAN speak English, but nothing turns me off from going into a shop faster than some black-moustached young guy yelling after me as I walk past his door, "Yes, please, kind sir! I sell to you flying carpet! Yes, please!" If only I knew a few more slang words that wouldn't start a fight...

One of the most satisfying moments of my entire stay in Turkey was when, stranded in the Konya bus station in the middle of a cold, snowing, January night, I struck up a conversation with the young man next to me. Due to his clean-shaved head and lack of moustache, I could tell he was a soldier on home leave.

"Got a light, brother?"

"Nope. Sorry, I don't smoke."

"So where you going?"

"I'm on my way back to Ankara. I just came from Aksehir and they told me there were busses to Ankara from here every hour all night. But now I'm here and they say there aren't any more busses until 4:30 a.m."

"Boy, that's pretty bad luck. I'm going back to duty, myself."

We talked for at least 30 seconds before he said, "Hey! You're not a Turk, are you?"

I guess the black jacket and the darkness of the bus station helped some, but I was really proud of my ability to blend in then. For the next week it was my favorite story—"How Laine was Mistaken for a Turk for 30 Seconds!"

The biggest barrier to having a really educational and fun experience in any foreign country is learning the language. Once you have at least a functional knowledge of the local tongue, things really open up to you. You are no longer just "one of those crazy Germans." You become a source of information about the outside world. And while you're giving out information, you might even get a bit back in return.

Gathering Information

We had given Jason's passport to the gateman before we entered the construction sight of a new dam on the Kizilirmak River. While we were wandering around we got curious about just how deep this dam was going to be. How much of the good rafting water upstream was it going to destroy? On the way out we decided to ask the gateman.

"Excuse me, but do you know when this dam will be completed?"

"Well, they've been working on it now for awhile and, one of these years, yeah, one of these years, not this year maybe, but they've been working on it and, uh, well maybe 1989, uh 76, 90..., maybe in 1988 they might finish it, but we're not really sure right now. There's some Rumanians here though, and they've been working on it..."

"How deep is it going to be?"

"Well, that's not really too clear at this point just how deep it's going to be. Because it's not finished yet, you know, we can't really say just exactly how deep it will be. But they've been working on it, these Rumanians, you know, and, well, when they finish it then it'll start to fill up and then maybe we'll be able to get some idea of just how deep it will be. But at this point it's just not really too clear."

"Oh. Uh, thanks."

But just try asking the same guy the maiden name of the mayor's older brother's second son's wife's mother...

Sometimes it's kind of tough to get the kind of details you want out of a discussion. The problem is that the local people aren't necessarily all that interested in what you are interested in. They also sometimes have different points of view.

"What's the river like downstream from here?"

"It's very dangerous! All kinds of rapids and things! You aren't thinking of going down it in THAT thing, are you? You'll die if you do!"

"Is this the road to Ermenek?"

"Are you kidding?!? There isn't any road to Ermenek from here! You can't get there from here!"

At another stop on the same road. "Is this the road to Ermenek?"

"Yes, but you'll never make it in that thing, you'd need a jeep or something."

"Ever been to Ermenek yourself?"

"No, but there are some truck drivers who go there all the time. They told me."

Work

Unless you happen to own majority stock in Microsoft, you're going to have to work while you're overseas to avoid starving. In many cases that may mean readjusting your idea of what an acceptable job is. It's not that there aren't computer jobs, it's just that they usually aren't quite as high tech as you'd be used to, especially if you're into systems software, firmware, or hardware design.

Most of the requests I get from people here in Turkey are not at the level of, "I need you to design an optical chocolate chip counter for my cookie making machine." They are more like, "I've got this friend who owns a printing company. Why don't you help me convince him that he should switch to desktop publishing and then we can share the consulting fee to set up his system." Or maybe, "Laine! We're in a real bad spot! We can't figure out how to make Word Perfect do a flush right margin! You've got to get over here right away!!!" The level of computer literacy just isn't as high here.

You shouldn't expect to come to a developing nation and find a job writing ROM BIOSs for PC clones or designing control boards for automatic chicken feed mixers. All of those kinds of things are done in Western Europe, the U.S. and Japan and then sent here as finished

products. Computer work over here is at a more grassroots level.

Just to give you an idea, here are some examples of jobs that I have discussed with various people (leaving out the ones which were just daydreams):

- Set up direct communications between a typesetting machine and an IBM and write software to translate from one data format to another.
- Design a database for a Turkish-English dictionary and make reports that output directly to typesetting.
- Translate several software products into Turkish, including menus, manuals, hyphenation, sorting, etc.
- Assist in the selection and installation of a computer system and software for a small archaeological library.
- Modify the firmware of various output devices to allow printing all characters in the Turkish alphabet.
- Assist in getting a distributorship for a certain brand of laser printer for a Turkish computer company.
- Set up a database of the archives for one of the largest museums in the world.

Of course, most of these jobs only show themselves after you are already in a country. You could just buy a plane ticket and hope that you found something once you got there, but I wouldn't recommend it. I would suggest calling the consulate of the country or countries you are interested in and asking for information; a few copies of the local English-speaking newspapers would help. If you have a friend who is already living overseas, that can be a big help; they might know of an opening that you could fill, at least temporarily.

Another way to "get in" to the job market of a foreign country is as an English teacher. Native English speakers are always in high demand. If you are lucky enough to have a teaching certificate, you can try your luck at the international teacher's hiring conference held every year at Iowa State University in Cedar Falls.

If you're really anxious but can't seem to find the right connections from stateside, just take a vacation and hop on a plane. When you get there, pick up all the local newspapers. Search in the phone directory for any company dealing with computers. Try to find some expatriates and get yourself invited to happy hour at the British or Canadian Embassy. You might even try talking to

IS NOTHING SACRED?

Now the FULL source code for TURBO Pascal is available for the IBM-PC! WHAT, you are still trying to debug without source code? But why? Source Code Generators (SCG's) provide completely commented and labeled ASCII source files which can be edited and assembled and UNDERSTOOD!

SCG's are available for the following products:

— TURBO Pascal (IBM-PC)*	\$ 67.50
— TURBO Pascal (Z-80)*	\$ 45.00
— CP/M 2.2	\$ 45.00
— CP/M 3	\$ 75.00
* A fast assembler is included free!	

"The darndest thing I ever did see..."
Pournelle, BYTE

"I have seen the original source and yours is much better!"
Anonymous, SOG VI

The following are general purpose disassemblers:

— Masterful Disassembler (Z-80) ..	\$ 45.00
— UNREL (relocatable files) (8080)	\$ 45.00

VISA/MC/check card#	Shipping/Handling	\$ 1.50
expires	Tax	\$
	Total	\$



All products are fully guaranteed. Disk format, 8" (), 5" (type).

C.C. SOFTWARE, 1907 ALVARADO AVE., WALNUT CREEK, CA. 94596, (415) 939-8153

CP/M and TURBO Pascal are trademarks of Digital Research & Borland Int.

Reader Service Number 31

the U.S. Embassy, but don't expect too much.

If you work at making yourself, your qualifications, and your intentions known, you might just find a job. If you do, cancel your return ticket and call home to have your stuff shipped over. If not, at least you got a vacation out of it.

Wages

The biggest commodity you have to sell is advice. Don't sell it short, either. I have the habit of feeling guilty about charging for words and thoughts. That's bad. If I keep it up, I could starve.

While it is true that you shouldn't let yourself go for free, you should also keep in mind that pay is all relative. For instance, I found it quite easy to survive in Ankara on \$200 a month (plus housing). Of course, I didn't live like most foreigners did, but I was comfortable; and when vacation time came around, I had plenty of money saved up to buy stupid playthings like computers and river rafts.

One thing to be careful of: in some countries (e.g., Nigeria) it is illegal to take more than a certain amount of local currency out of the country. Make sure you check into the local laws and insist on having everything over the limit paid into a foreign account. If the local economy is unstable, try to keep as little money in local banks as possible. If you get a local account, try to get a dollar account or a sterling account.

My own Turkish bank account is in the top drawer of my dresser (or sometimes in the front pocket of my suitcase). At times I let it dwindle to nothing. If you are in a country with a political situation not as stable as Turkey, though, you might consider always keeping enough cash on hand to buy a ticket out of the country just in case things get uptight. Or at least have a credit card with a high credit limit.

Contracts

If you're being hired from the U.S., you'll probably be required to sign a contract guaranteeing that you'll work for a minimum amount of time, usually one or two years. This is to protect the company's investment in you. Usually they will pay your airfare and moving costs from the U.S. and, if you finish the contract, they will pay the costs back as well.

There is another expense that is often overlooked—adjustment time. Unless you are working in an office full of Americans, you will be completely

worthless for the first couple months. You'll be freaked out from the strange food, strange people, strange music, strange language. In the meantime, you're still getting paid.

As long as you're writing up a contract anyway, make sure of a few other things:

(1) Full medical insurance that covers you anywhere in the world.

(2) Housing of a guaranteed standard, with telephone, heating if necessary, constant supply of water (if possible), washing machine, etc.

(3) Adequate vacation time each year (i.e., at least one month). If you're going to be working in some strange place, you may as well take the time to see it.

(4) Double check that your moving expenses and airfare will be paid both ways.

Never assume anything, always have it written and signed.

Good Luck, Jim

I started out writing this just to let you all in on some of the freaked out

things that have happened to me lately, just wanting to be a storyteller. But it looks as if the whole thing was designed to convince you to give a try at working overseas. Maybe, in some unconscious way, it was. I know that I wouldn't have missed the last two years for anything. Matter of fact, I'm looking forward to more of the same.

If any of you are currently working in some wild, uncharted country, or if you end up doing so in the future, I'd love to hear about it. (*Editor's note: So would Micro C.*) I'm always on the lookout for new opportunities.

Meanwhile, if there are any college seniors out there who are interested in having a couple years of excitement before getting a "real" job, drop me a line (in care of *Micro C*) telling me what you're interested in and what kind of qualifications you have. Maybe I can rustle something up for you here.

A Program For Internationals

Since the topic of the month is working in foreign countries, I thought I

Figure 1 - Printer Output Translation Program

```
*****
** EPSPRN - a program to translate some IBM foreign characters to **
** Epson foreign characters. **
** To use, include the line 'EPSPRN' in AUTOEXEC.BAT **
** CAUTION - does not work correctly with graphics programs!!! **
**
** To assemble: MASM EPSPRN; **
** LINK EPSPRN; **
** EXE2BIN EPSPRN **
** REN EPSPRN.BIN EPSPRN.COM **
** ERASE EPSPRN.EXE **
**
** Laine Stump, October 11, 1987 **
**
** Permission granted to do whatever you damn well please with **
** this program. **
*****
;
CODE segment 'CODE'
assume cs:code
FIRSTBYTE equ this byte ;between here & LASTBYTE remains res.

ORG 100h
START: JMP INIT ;init code is at end so we can get rid of it.

*****
XLATTABLE equ this byte
XLTTLEN equ 8 ;number of bytes in each xlation string
; structure is:
; IBMCHAR, up to 8 char xlation string terminated w/OFFh
; currently set up to do characters important to Turkish alphabet
; unfortunately there is no way to properly do an undotted small "i"
; change to fit your requirements

DB 80h,'C',8,'',OFFh,0,0,0 ;C
DB 81h,1Bh,'R',2,'}',1Bh,'R',0,OFFh ;u
DB 83h,'a',8,'^',OFFh,0,0,0 ;a
DB 87h,'c',8,'',OFFh,0,0,0 ;c
DB 8Dh,1Bh,'R',6,'~',1Bh,'R',0,OFFh ;i
DB 8Eh,'A',8,'^',OFFh,0,0,0 ;A
DB 94h,1Bh,'R',2,'|',1Bh,'R',0,OFFh ;o
DB 98h,'I',8,'''',OFFh,0,0,0 ;I
DB 99h,1Bh,'R',2,'\'',1Bh,'R',0,OFFh ;O
DB 9Ah,1Bh,'R',2,']',1Bh,'R',0,OFFh ;U
DB 9Eh,'S',8,'',OFFh,0,0,0 ;S
DB 9Fh,'s',8,'',OFFh,0,0,0 ;s
```

would include a little assembly language program that catches all printer output and translates from IBM standard foreign characters into Epson's "alternate character set" foreign characters. See Figure 1.

The program is a TSR that installs itself over interrupt 17 (hex, of course) and searches through a table to look for a match for every character that is output. If a match is found, it sends a string of 1 to 8 characters to the printer in place of the original character. Usually, a command is sent to switch to an international character set, print the character, and switch back.

You can use it in other ways, however; notice how I do Ç (C with cedilla) by printing "C^H,". You may even want to switch to graphics mode and output a bit pattern (although that could cause problems if your program switches to italics, subscript, or NLQ). If you need more than 8 characters in the translation strings, just change the number XLTLLEN (and make sure you add the correct number of filler bytes to every entry in the table!!).

You'll notice that at certain times I ignore the translation. This was necessary because I found that sometimes the program was translating things like horizontal position parameters and graphic byte counts. My method of telling when to not translate is by no means thorough. What is needed is a state machine that exactly mimics the Epson printer command set. If anybody is ambitious enough to do that, send me a copy!

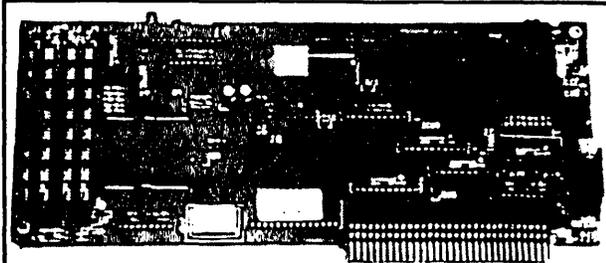
Next issue I'll show you a simple method of adding international keyboard input to your PC. In the meantime you can hold down the alt key while using the keypad to type the decimal number associated with the character you want. For example, to get ä, just push down and hold alt, type "128" on the keypad, then let up on alt. Unless you have a cheap, rotten Taiwanese BIOS ROM, you should get a ä! If not, just hang around and I'll show you a few tricks.

P.S.

The River Trip down the mighty Goksu was truly incredible. I can't describe it properly with words, though. You'll just have to come see the slides at the next SOG.



A Reliable PC/XT Compatible For The Corner Stone of Your Products



Announcing The SLY40-XT

The SLY40-XT is a small (4-1/4" by 9-1/4"), four layer card featuring all of the PC/XT mother board functions. The board simply plugs into a passive back plane or SLICER'S 10 slot bus board.

- High Integration — Composed of just 17 Low Power CMOS ICS
- NEC's 8 MHZ V40
- One Megabyte of Zero Wait State RAM
- Up to 64K of ROM
- 8087 Co-Processor Socket
- Customized Serial Port
- Standard Keyboard Connector
- Slicer's Own Bios, Source Code Included
- Ideal For Tough Industrial, OEM and Portable Applications
- American Made and Fully Supported by Slicer

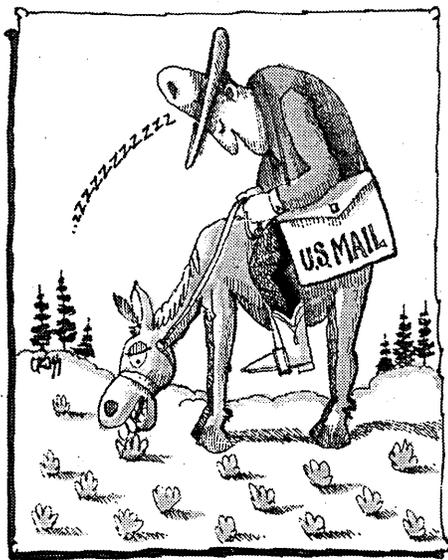
Ask About Our Complete Line of Computer
Products and Accessories!

MasterCard, Visa, Check, Money Order, or C.O.D.
Allow four weeks for delivery.
Prices subject to change without notice.
NOTE NEW ADDRESS & PHONE NO.

 **SLICER**™ Slicer Computers Inc.
3450 Snelling Ave. So.
Minneapolis, MN 55406
612/724-2710
Telex 501357
SLICER UD

PC and XT Are Trademarks of International Business Machines

Reader Service Number 19



U.S. Postal Service Statement of ownership, management and circulation (Required by 39 U.S.C. 3685) 1A. Title of Publication: MICRO CORNUCOPIA 1B. Publication Number: 0747-587X 2. Date of Filing: 9-30-87 3. Frequency of Issue: Bi-monthly 3A. Number of Issues Published Annually: 6 3B. Annual Subscription Price: \$18.00 4. Location of Known Office: 155 NW Hawthorne, Bend, Oregon 97701-2917 5. Location of the Headquarters or General Business Offices of the Publishers: 155 NW Hawthorne, Bend, Oregon 97701-2917 6. Name and Complete Address of the Publisher, Editor, and Managing Editor: Publisher: David J. Thompson 155 NW Hawthorne, Bend, Oregon 97701-2917; Editor: David J. Thompson 155 NW Hawthorne, Bend, Oregon 97701-2917; Managing Editor: David J. Thompson 155 NW Hawthorne, Bend, Oregon 97701-2917 7. Owner: Micro Cornucopia, Inc. PO Box 223, Bend, Oregon 97709-0223; David J. Thompson 1259 NW Iowa, Bend, Oregon 97701-1001; Sandra S. Thompson 1259 NW Iowa, Bend, Oregon 97701-1001 8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1% or More of Total Amount of Bonds, Mortgages or Other Securities: None 10. Extent and Nature of Circulation: Average Number of Copies Each Issue During Preceding 12 Months A. Total Number of Copies Printed: 20,536 B. Paid Circulation: 1) Sales Through Dealers and Carriers, Street Vendors, and Counter Sales: 6,556 2) Mail Subscription: 8,675 C. Total Paid Circulation: 15,231 D. Free Distribution by Mail, Carrier, or Other Means, Samples, Complimentary, and Other Free Copies: 3,495 E. Total Distribution: 18,726 F. Copies not Distributed: 1) Office Use, Left Over, Unaccounted, Spoiled after Printing: 860 2) Returns from News Agents: 950 G. Total (Sum of E, F1 and 2)-Should Equal Net Press Shown in A: 20,536; Actual Number of Copies of Single Issue Published Nearest to Filing Date: A. Total Number of Copies Printed: 42,093 B. Paid Circulation: 1) Sales Through Dealers and Carriers, Street Vendors, and Counter Sales: 12,615 2) Mail Subscription: 8,300 C. Total Paid Circulation: 20,915 D. Free Distribution by Mail, Carrier, or Other Means, Samples, Complimentary, and Other Free Copies: 20,140 E. Total Distribution: 41,055 F. Copies not Distributed: 1) Office Use, Left Over, Unaccounted, Spoiled after Printing: 1,038 2) Returns from News Agents: 0 G. Total (Sum of E, F1 and 2)-Should Equal Net Press Run Shown in A: 42,093

```

DB 0A6h,'G',8,'~',OFFh,0,0,0,0 ;G
DB 0A7h,'g',8,'~',OFFh,0,0,0,0 ;g
DB 0 ;end of table sentinel
;*****
INT17h equ this dword
INT17ofs DW ?
INT17Seg DW ?

PREVCHAR DB 0
;*****
INTSERV:
OR AH,AH ;IF function 0 (output char)
JNZ INTSERV2
CMP PREVCHAR,1Bh ;AND not midway in a printer command
MOV PREVCHAR,AL
JNE INTSERV0
CMP AL,'$' ;only checking for these commands now
JZ INTSERV00
CMP AL,'C' ;because these are used by MS-WORD
JZ INTSERV00
CMP AL,'J'
JZ INTSERV00
CMP AL,'K'
JNZ INTSERV01

INTSERV00:
MOV PREVCHAR,1Bh
INTSERV01:
JMP short INTSERV2

INTSERV0:
PUSH SI ;THEN translate
MOV SI,offset XLATTABLE

INTSERV1:
CMP AL,CS:[SI] ;look for match in table
JE INTSERV3
ADD SI,XLTLEN+1
CMP byte ptr CS:[SI],0 ;end of table?
JNE INTSERV1
POP SI

INTSERV2:
JMP INT17h ;output original character

INTSERV3:
;output a string terminated w/FF
;string is at CS:[SI+1]

INC SI
CMP byte ptr CS:[SI],OFFh
JE INTSERV9
MOV AL,CS:[SI]
MOV AH,0
PUSH SI
PUSH DX
PUSHF
CALL INT17h
POP DX
POP SI
JMP INTSERV3

INTSERV9:
POP SI
IRET

LASTBYTE equ this byte
;*****
; Below this point only used during initialization, then discarded
DOS equ 21h
;
assume ds:code
INIT: MOV AL,17h ;get the original INT 17h vector
MOV AH,35h
INT DOS
MOV INT17ofs,BX ;save to call later
MOV INT17Seg,ES

MOV AL,17h
MOV DX,offset INTSERV
MOV AH,25h
INT DOS

MOV DX,(LASTBYTE-FIRSTBYTE+15)/16
MOV AL,0
MOV AH,31h
INT DOS

CODE ends
end START
< end of Figure 1 >

```

LETTERS

(Continued from page 4)

sumed the worst—that many other shareware programs were so listed. In fact, it's not as bad as I thought. But I hope your next catalog will make the difference very clear. Thanks for your attention.

Neil J. Rubenking
Member, Board of Directors
Assn. of Shareware Professionals
300 Page St.
San Francisco, CA 94102

Editor's note: Points well taken. I attended the Shareware Conference last winter and so I'm well aware of the problems faced by the shareware industry. We're changing the public domain column to "Shareware" and we've corrected the catalog. Sorry for the mistake.

Computers In Bend?

I recently received issue #37 of your magazine as a sample copy. You should have solicited sooner. After living in Puyallup, Washington, and being raised in Yakima, it's difficult to understand how Bend, Oregon, (and I've been there) would become the home of a decent computer publication.

Maybe your readers already know the history of the magazine and its staff, but I would certainly be interested to see it in print. How does one make a living programming in Bend or anyplace close to it? Is there anyplace close to Bend?

It's been a long time since I've spent so much time with a single issue of a magazine. I'm currently considering Desktop Publishing for another business venture and found your article very informative. Having been an assembly language programmer for over 21 years with strong Pascal skills, "Taking The Plunge Into C" almost makes me want to look at the Lets-C compiler I won.

"Intro To Database Programming" confirmed the methodology I've used in dBase applications. The Culture Corner was great and the EMS article was very informative. The shareware articles were of great interest as I'm considering distribution of several of my financial applications as shareware.

A word of warning to those considering programming the DMA controller (see "DMA Control On The PC"). It's very easy for a data buffer in memory to cross a 64K page boundary. Therefore, the beginning and ending addresses of

the data buffer should be checked to be within the same memory page. Very interesting things happen when they are not.

Ken Zaremba
Zaremba Enterprises Ltd.
12215 Irwin Way
Boulder Creek, CA 95006

Editor's note: Yes, Ken, there are computers in Bend, we brought them with us. No, Bend isn't near anywhere, that's why we're here. As for how we got here? We're here because we like it here. Finally, you're right, we should do a history of Micro C. Perhaps for next issue's Culture Corner.

DMA Feedback

I found the complementary copy of your issue #37 absolutely wonderful. I delighted not only in its avoidance of such pressing issues as printing multi-colored directories and securing your recipes from prying eyes, but also in its inclusion of some down-to-earth hardware control. At last, a magazine that helps the electronics novice do something substantial with that hunk of metal and silicon that sits mutely on his desk.

Having spent the last few months figuring out how to get a digitizer to write directly to the memory of an IBM PC, it was with special interest that I discovered the article by Larry Fogg, "DMA Control On The PC." He is grossly understating the case when he says that this subject can be confusing. It can be downright mystical, and his was one of the most concise and exacting descriptions of this Great Mystery of the PC that I have seen.

Being naturally nit-picky, I can't resist finding one small fault with the article as it stands. In its discussion of registers, the article states that the DMA page registers for channels 0-3 are at addresses 80h to 83h, respectively.

Actually, addresses 81h, 82h, and 83h map to channels 2, 3, and 1. Channel 0 is at address 87h. I guess this is just another of IBM's attempts to keep us from turning to more boring pursuits, like skydiving.

Eric J. Pilger
Institute for Astronomy
2680 Woodlawn Dr.
Honolulu, HI 96822

More DMA Comments

I was very surprised to read in *Micro*

C issue #37 that IBM had taken the liberty of violating the 2 msec refresh spec of the dynamic RAM chips. It just didn't sound like the usual conservative IBM engineering.

It didn't occur to me until the next day that you had said, "We know that it takes 512 DREQs to refresh all of memory..." Not quite. I recall working with 4116 16K X 1 chips that had 128 rows requiring refresh. When 4164s came along, most manufacturers gimmicked them to also require only 128 refresh cycles.

I remember doing an upgrade of an Atari 400 from 16K to 64K, replacing the 4116s with Micron Technology 4164s. The poor Atari would work for a moment, then lose its mind. It turns out that the Atari controller cycled only A0 through A6 for 128 refresh cycles and the Micron SpuDRAMs were among the few 4164s requiring 256 cycles. Other 4164s worked fine.

To get to the point, beginning with the first popular 4116s (and even into the 1 Meg chips), manufacturers have maintained the refresh requirement of 128 rows per 2 msec. As the number of rows requiring refresh has doubled with each generation, the total time to refresh all rows has also doubled. The seven plus msec that you figured for 512 cycles actually turns out to be, surprise, a bit less than two msec for each 128 cycles.

The good news is that I think it's a great idea to adjust the refresh timer to squeeze out a few more milliMIPS. It hasn't seemed to make my clone the least bit nervous. I enjoyed the rest of the DMA article very much, and learned a great deal about the subject.

I have one question. I know that the IBM manuals are the definitive source for technical information, but I don't have the hundreds of bucks for a complete set. The schematics that came with my clone are microscopic. Is there some other source?

Gary Crowell
2014 W. La Palma
Anaheim, CA 92801

Editor's note: We don't know of another source for the main board schematics. Sams may have a book out now. Several folks wrote in pointing out the error in the refresh discussion. Thanks.





All Aboard The ARC

By Anthony Barcellos
P.O. Box 2249
Davis, CA 95617-2249
(916) 756-4866

Tony Barcellos teaches mathematics at American River College in Sacramento and has a T-shirt that says "ARC" on it. He is also Sacramento PC's software librarian and editor of Sacra Blue, its monthly newsletter.

You have a nifty public domain or shareware utility you want to send to a friend. Since both of you have modems, this should be easy, right?

In the olden days, the program you wanted to share was most likely a tiny public domain utility in a single disk file. Today the programs we exchange are dominated by shareware, and most of these comprise collections of companion files. Thus the delivery of the nifty program becomes a tedious transmission of file1, file2, file3 ..., until the job is done.

In Single File

Computer users didn't put up with that for very long. The "library utility" LU came along as a means of lumping multiple files into a single "library" file. Now you could send everything in a single transmission.

Of course, this single transmission might take a *very* long time, since the one lumped-together file could be a pretty large lump. "Squeeze" programs were used to compress the files before they were merged by LU into a library.

Thus there developed the routine known so well to software librarians and bulletin board fans: squeeze the files, library the files, upload the library, download the library, break up the library, "unsqueeze" the files. All of us had a suite of programs like SQZ, NUSQZ, LU, LU86, USQ, ALUSQ, etc., to squeeze, library, and unsqueeze program files. You won't be surprised to learn that the proliferation of utilities for these purposes led to numerous improvements—and, of course, incompatibilities.

All Aboard The ARC

Then Thom Henderson of System Enhance-

ment Associates (SEA) had a better idea. Why not create a single utility program that would take care of the entire process. In a break with the past, Henderson called his libraries "archives" and SEA's program was released in March, 1985, under the name ARC.

ARC took the shareware world by storm. Electronic bulletin boards across the country embraced the new standard. LU headed for the shelf. The .ARC extension blossomed on the BBS download directories. While I still keep copies of LU and LU86 in my utility collection for handling old files with the outdated library extension (.LBR), today's utility of choice is ARC.

Of course, soon there would be other ARC-typal programs with their numerous improvements. (And—uh oh!—incompatibilities.)

Everybody Into The Pool

ARC's sudden primacy did not remain unchallenged for long. SEA had unknowingly launched a revolution, and those that rushed to its standard brought along a few new ideas of their own.

The first skirmishes of the ARC revolution were fought on the bulletin boards where SYSOPs who clung to LU criticized ARC's performance. Indeed, as the LU partisans fell back, ARC failed to move in quickly enough and some of the abandoned territory was seized by swiftly moving clones.

Phil Katz of PKWARE is the author of PKARC and PKXARC, archiving and archive extraction utilities, respectively. PKXARC was his initial release, intended to speed the process of pulling files out of an archive. Written in assembly language, PKXARC left ARC in the dust. SYSOPs and software librarians appreciated Katz's five-fold speed advantage and PKXARC became well-established.

SEA Changes

ARC evolved by stages into version 5.20, released late in 1986 with improved performance and enhanced compression algorithms. The program currently comes in the form of an executable file that is itself an archive. It can be

found in software libraries and on BBSs as ARC520.COM. When executed (just enter "ARC520" at the DOS prompt), the program performs a self-extraction, generating the files ARC.EXE (the actual ARC program) and ARC.DOC (the user's manual), ARC.TXT (a word from our sponsor), and ARCE.COM (a special archive extraction utility).

The documentation for ARC is quite good, detailing the program's numerous options. ARC can, of course, add to or delete files from an archive. It can update archive files (adding missing files or overwriting existing files with newer versions), freshen them (updating only files that are already archived, adding no new ones), display text files (without extracting them), execute archived programs (without extracting them), and list archive contents. Other options

you're transmitting a file long-distance, and size is time!)

The "stowage factor" shows the degree to which a file was reduced. In the given example, overall file size fell by 28%.

Raising The Standard

Not content to confine its efforts to high-performance archive extraction, PKWARE introduced its first version of PKARC in August, 1986. Yet another assembly language speed demon, PKARC was offered (accompanied by PKXARC) as a complete alternative to ARC.

Both PKARC and PKXARC are now at version 3.5 and are distributed as a self-extracting executable archive file called PKX35A35.EXE, released in April, 1987. Upon entering "PKX35A35" at the DOS prompt, you are presented with the

disk for the performance tests.

The archive illustrated in Figure 1 was created by ARC in 9 minutes and 15 seconds on a standard IBM PC with a hard disk and a clock speed of 4.77 MHz. On the same system, the identical files were archived by PKARC in 1 minute and 27 seconds. PKARC also achieved a 30% reduction in overall file size versus only 28% for ARC. (See Figure 2.)

PKXARC beats the pants off ARC on the file-extraction side. While ARC disassembled its Baker's Dozen archive in 5 minutes and 30 seconds, PKXARC took only 1 minute and 2 seconds to dismantle the similar archive created by PKARC. (PKXARC required an additional 5 seconds when used on the archive created by ARC.)

Upward, Ever Upward

SEA and PKWARE have both been careful to maintain compatibility with earlier releases of their utilities. For its part, PKWARE has consistently produced programs that adhere to the archiving standards set by ARC. In release 3.5 of PKARC, however, PKWARE for the first time goes beyond ARC by introducing a compression algorithm not yet supported by SEA. Like LU and its variants of olden days (that is, two years ago), archives now suffer from incompatibility.

The "verbose" archive listing produced by PKARC for the ButtonWare programs (Figure 2) shows the stowage method that PKWARE calls "squashing." ARC chokes on archives that contain squashed files and issues a message suggesting that perhaps you need a new version of ARC. (ARC responds this way any time it sees an unfamiliar archive format.)

PKARC also permits the user to annotate archives with brief descriptions of their files. While this feature does not directly conflict with ARC, the comments are invisible to the SEA program, Phil Katz warns that PKARC-created comments will be lost if an archive is altered in any way by ARC.

If PKWARE and SEA continue to diverge in their implementations the user is in for trouble. Fortunately, PKWARE has provided an "old compatibility" option to disable file-squashing and preserve ARC-compatible archive formats. If you forget to invoke this option, however, you can send someone an archived file that he can't extract (just as I did when I transmitted my first *Micro C* column).

Figure 1 - ARC Generated Archive, Time: 9 Min. 15 Sec

Name	Length	Stowage	SF	Size now	Date	Time	CRC
BAKERS12.EXE	15091	Crunched	17%	12621	1 Jun 87	1:00a	5CC6
BTTNCALC.EXE	119463	Crunched	26%	89499	1 Jun 87	1:00a	3231
CAL.DAT	1792	Crunched	39%	1097	1 Jun 87	1:00a	CD2B
CALENDAR.COM	4520	Packed	4%	4360	1 Jun 87	1:00a	829B
DISKUTIL.EXE	114695	Crunched	27%	84343	1 Jun 87	1:00a	1BC2
DOZEN.BAT	128	Packed	82%	24	1 Jun 87	1:00a	408A
DOZEN.TBL	896	Crunched	26%	667	1 Jun 87	1:00a	6D9B
FILECOMP.EXE	14573	Crunched	28%	10493	1 Jun 87	1:00a	CD43
GKEY.EXE	5120	Crunched	34%	3399	1 Jun 87	1:00a	B582
LOCATE.EXE	6144	Crunched	32%	4187	1 Jun 87	1:00a	A861
ORDER	2048	Crunched	51%	1004	1 Jun 87	1:00a	F648
P90.EXE	22433	Crunched	27%	16474	1 Jun 87	1:00a	3455
PC_SORT.EXE	10752	Crunched	35%	7091	1 Jun 87	1:00a	D91C
PRN_FILE.EXE	3584	Crunched	41%	2135	1 Jun 87	1:00a	095F
RDIR.EXE	5120	Crunched	44%	2906	1 Jun 87	1:00a	C2E1
READ.ME	13056	Crunched	51%	6503	1 Jun 87	1:00a	856B
RESPONSE	1024	Crunched	32%	701	1 Jun 87	1:00a	7963
SETSCRN.COM	70	Packed	5%	67	1 Jun 87	1:00a	3D03
SET_SCRN.EXE	6415	Crunched	20%	5176	1 Jun 87	1:00a	5ADF
SNAPSHOT.COM	5444	Crunched	23%	4209	1 Jun 87	1:00a	F3A1
SWCOML2.COM	340	Crunched	17%	283	1 Jun 87	1:00a	156C
SWLPT12.COM	349	Crunched	18%	289	1 Jun 87	1:00a	AB9A
Total	22		28%	257528			

govern the file storage format, the way files are archived ("added" with the original file remaining or "moved" with nothing left behind), and archive integrity.

The "verbose" file-listing option produces a detailed list of archive contents. (See Figure 1.) The "stowage" column identifies the packing routine used by ARC in adding a file to the archive. Files can be "stored" (added without compression), "packed," "squeezed" (no longer supported in the most recent version of ARC), or "crunched." As a file is examined before archiving, ARC determines which compression algorithm will produce the smallest archive. (Time is money when

files PKARC.COM, PKARC.DOC, PKXARC.COM, PKXARC.DOC, PKXARCJR.COM, MAKESFX.COM, PKSFX.DOC, and README.DOC. The .DOC files are the expected user's manuals. The MAKESFX utility is used to create self-extracting archives and is explained in PKSFX.DOC.

PKARC offers enhanced file compression as well as a drastic performance edge over ARC. A collection of ButtonWare shareware utilities called Baker's Dozen provided guinea pigs for a performance comparison of ARC versus Phil Katz's alternatives. The ButtonWare programs totaled 353,057 bytes and completely filled a standard 360K floppy disk. The files were copied to a hard

(Editor's note: Tony's first piece was a lot more interesting after we finally decompressed it.)

Buerg To The Rescue

Just in case squashing becomes popular, SEA now distributes Vernon Buerg's ARCE extraction utility with ARC. Better known for his excellent LIST program, Buerg is a craftsman whose tools should be welcome in anyone's utility room. He's included "unsquashing" in ARCE so that

and customized versions of its programs. Contact:

PKWARE, Inc.
7032 Ardara Avenue
Glendale, WI 53209

PKWARE requests a \$20 contribution for use of either PKARC or PKXARC. A \$45 registration fee entitles you to a diskette with the next release of both programs and their companion files. Remember to report the version you're

into the Sacramento PC software library. In a way, Baker's Dozen harks back to the small, single-purpose utilities that used to pepper the BBS landscape. Although ButtonWare concentrates on full-fledged application packages like database managers and word processors, the Baker's Dozen grab-bag of utilities adds a nice dash of spice to the company's offerings.

On the Menu

Baker's Dozen comes with a menu-driven front-end that the steady user will soon abandon. As the documentation admits, the menu is "only intended to get you started. Normally you should run Baker's Dozen programs by themselves." However, it's a good way to get a first look at what the ButtonWare utilities can do for you.

Bttncalc. Is your application too small to justify Lotus 1-2-3 or SuperCalc? Then try the ButtonCalc one-page spreadsheet from ButtonWare. You can save and reload your work, of course.

Calendar. Yet another calendar program. But this isn't an appointments calendar. It's a programmable date calendar that pops a display into the corner of your screen. Use the cursor keys to page through the various months. Make the program resident (one of its options) and find out any date at any time from within any program.

DiskUtil. Need I say more? Nibble on bytes to your heart's content. Hop from file to file or directory to directory. Render your FAT. Raise files from the dead. Not for children or first-timers.

FileComp. Fed-up with the wimpy DOS COMP command? Button's program will compare two text files and display the line numbers of unique lines, the unique lines themselves, or a map of matching line numbers or lines; it can ignore spaces in its comparisons.

GKey. Get all the keyboard information you want: key number, scan code, hexadecimal and decimal ASCII values.

Locate. Another descriptive name. Find files wherever they may be lurking or even find all files containing a specified text string. You can make the text search insensitive to case.

P90. This is a "sideways" printing program for Epson-type printers. Where did that name come from? "Print 90 degrees."

PC_Sort. Use command-line options or let the program prompt you for up to four fields by which your file should be sorted. Ascending, descending, case sen-

Figure 2 - PKARC Generated Archive In 1 Minute, 27 Sec.

Filename	Length	Method	Size	Ratio	Date	Time	CRC
BAKERS12.EXE	15091	Crunched	12414	18%	06-01-87	01:00:00	5CC6
BTTNCALC.EXE	119463	Squashed	85411	29%	06-01-87	01:00:00	3231
CAL.DAT	1792	Crunched	1097	39%	06-01-87	01:00:00	CD2B
CALENDAR.COM	4520	Packed	4375	4%	06-01-87	01:00:00	829B
DISKUTIL.EXE	114695	Squashed	80206	31%	06-01-87	01:00:00	1BC2
DOZEN.BAT	128	Packed	24	82%	06-01-87	01:00:00	408A
DOZEN.TBL	896	Crunched	666	26%	06-01-87	01:00:00	6D9B
FILECOMP.EXE	14573	Crunched	10240	30%	06-01-87	01:00:00	CD43
GKEY.EXE	5120	Crunched	3285	36%	06-01-87	01:00:00	B582
LOCATE.EXE	6144	Crunched	4006	35%	06-01-87	01:00:00	A861
ORDER	2048	Crunched	1005	51%	06-01-87	01:00:00	F648
P90.EXE	22433	Crunched	16111	29%	06-01-87	01:00:00	3455
PC_SORT.EXE	10752	Crunched	6965	36%	06-01-87	01:00:00	D91C
PRN_FILE.EXE	3584	Crunched	2135	41%	06-01-87	01:00:00	095F
RDIR.EXE	5120	Crunched	2913	44%	06-01-87	01:00:00	C2E1
READ.ME	13056	Squashed	6564	50%	06-01-87	01:00:00	856B
RESPONSE	1024	Crunched	701	32%	06-01-87	01:00:00	7963
SETSCRN.COM	70	Packed	67	5%	06-01-87	01:00:00	3D03
SET_SCRN.EXE	6415	Crunched	5165	20%	06-01-87	01:00:00	5ADF
SNAPSHOT.COM	5444	Crunched	4185	24%	06-01-87	01:00:00	F3A1
SWCOM12.COM	340	Crunched	283	17%	06-01-87	01:00:00	156C
SWLPT12.COM	349	Crunched	289	18%	06-01-87	01:00:00	AB9A
-----	-----	-----	-----	-----	-----	-----	-----
0022	353057		248107	30%			

PKWARE archives can be imported into SEA environments.

Whether this portends new features in ARC isn't clear. We'll just have to wait for version 5.3. In the meantime, devoted ARC users will find that ARCE is also a performance demon and much faster at file extraction than ARC itself.

The Passengers On The ARC

ARC can be obtained directly from:

System Enhancement Associates, Inc.
21 New Street
Wayne, NJ 07470

for \$50, which includes a program disk and printed documentation. If you obtain ARC from a software library or BBS, you must remit a license fee of \$35 before using the product in a commercial or government environment. Contact SEA for details on site licenses and commercial distribution.

PKWARE also offers site licenses, volume discounts, commercial licenses,

using when sending in your registration fee.

Vernon Buerg can be reached in Daly City, California, or contacted via his Fido BBS.

Vernon Buerg
456 Lakeshire Drive
Daly City, CA 94015
BBS: (415) 994-2944, FidoNet 125/4

References

The actual compression algorithms used by ARC and PKARC are quite complicated. The SEA documentation refers to articles which contain the details. Also, the March, 1987, issue of *Dr. Dobb's Journal* contains extensive benchmarks on the performance of ARC and its competitors.

Button, Button, Button . . .

I picked Jim Button's Baker's Dozen as the test subject for my archiving experiments because I just recently put it

sitivity, column offsets, and field lengths are all under your control. (Vernon Bueg has a wonderful sorting program called SORTF that you might also be interested in. It makes the DOS SORT filter look sick.)

Prn_File. Redirect printer output to a file. (How did this get left out of DOS? That's what I'd like to know.) Make a program that demands a printer feel right at home on a printerless PC—or capture information you'd rather have on disk than on paper.

RDir. Remove a subdirectory and all of its contents, including lower subdirectories.

Set_Scrn. Choose your screen foreground, background, and border colors. Requires ANSI.SYS.

Snapshot. A screen-grabbing utility that can be made resident. Works only with text screens.

SWCOM12. Swap COM ports 1 and 2. (What else is there to say?)

SWLPT12. Yet another utility to swap LPT1 and LPT2. (If you're one of the few people with *three* active parallel ports, then you might prefer LPT3SWAP.COM from HullSoft, which moves all three ports in a cycle. It's available for downloading from various BBSs in Sacramento and San Francisco.)

As usual, ButtonWare has another solid product that will benefit users. While some entries in the Baker's Dozen are old hat, the collection as a whole is very nice. You can also trust ButtonWare products to be free of major bugs and easy to use. Baker's Dozen fits firmly into this ButtonWare tradition. The registration fee for the Baker's Dozen is \$59.95. Add \$5 for shipping and Washington state residents must include 8.1% sales tax.

Jim Button
P.O. Box 5786
Bellevue, WA 98006
Toll free: 1-800-JBUTTON



MAILBASE SYSTEM™

Productivity Software

for dBase/Wordstar/Ventura Publisher

• Letters/forms/contracts • Production • Record keeping • Grouped/repeated work, variations • Secretarial or professional use • Meeting management • Desktop input • dBASE file organizer • Develop your own specialized system with no programming • Constant or on-the-fly formatting • Stackware with standard programs

Painless construction of general letters, customized contracts, tabbed tables for Ventura Publishers, etc; from dBASE II or III files. Use any version dBASE & Wordstar/MM. Track meeting participants; contracts; business letters; automatically make action summaries. Over 5 years of practical development. Never again type anything twice. MS/PC-DOS, but also an Apple II CPM-Softcard version (not 7), 5 1/4 disks.

FEATURES

1. Use any dBASE file & fields up to 214 (characters or numeric);
2. Automatically track outgoing multi-copy letters & variants;
3. Select any fields at run time for letter integration, adjacent fields for block text;
4. Branch to alternate letters in a single mailmerge pass then summarize regional actions;
5. Copy any fields to subsequent records, either old or newly appended, for letter/contract production;
6. Make consistent "structure extended" data dictionaries in dB format for transparent systems management;
7. Produce correctly tabbed 2, 3, or 5 col. tables for Ventura Publisher from any dB file;
8. Other dB file management & production utilities; 2 col. Harvard Publisher tables, (other wp's on request);
9. On-disk documentation: manual; tutorial; examples; letter/contract skeletons. Hardcopy manual \$12 extra;
10. Use to customize invoicing systems, meeting management operations without programming.

- NOT COPY PROTECTED
- Mail order only, \$10 secondary sales rebate
- Money order or personal check (allow ten days to clear).

SEND TO:

HARGER I.N.T.
P.O. Box 20, Grand Central Station
JKT Pouch
New York, New York 10163

ONLY
\$45

dBASE II & dBASE III are trademarks of Ashton Tate, Wordstar & Mailmerge are trademarks of MicroPro, Apple II is a trademark of Apple Computer Inc. Softcard, MS-DOS & Microsoft are trademarks of Microsoft Corporation, CP/M is a trademark of Digital Research Inc. PC DOS is a trademark of International Business Machines Corporation. Ventura Publisher is a trademark of Univation Inc. Harvard Publisher is a trademark of Software Publishing Corporation.

Reader Service Number 26



Why are serious PC software developers demanding ...

Show Me!

Because *Show Me!* is the NEW "must have" tool for today's PC programmer. With this memory-resident, file-windowing utility, see your productivity soar as you view as many as four files at once instantly (in ASCII, EBCDIC or hexadecimal) at the touch of a key.

- escape that caged-in feeling while programming in Turbo Pascal, Turbo Prolog, Turbo C, Turbo BASIC, QuickBASIC 4.0, dBASE, BASICA, and similar programming environments, and work with up to four additional program files at once -- complete with copy & paste, print, and search capabilities
- view multiple source and listing files in up to four windows while using CodeView, DEBUG, SYMDEB, and other debuggers -- great for assembly programming!
- peek at any ASCII file including Wordstar document files and easily paste all or part of a file into virtually any program that accepts keyboard input
- visually compare files side-by-side in simultaneously scrolling windows
- find your files in the built-in directory window

Order *Show Me!* today for only \$39 (+\$5 S/H)!

To order call toll-free 800-634-3122

Visa and MasterCard accepted • 30-day satisfaction guarantee • Not copy protected



Ask about *So Help Me!*, a flexible, context sensitive help screen driver. Add full-color help screens to all your programs — royalty free — for only \$79!

Serengeti Software • P.O. Box 27254 • Austin, Texas 78755 • 512-345-2211

Show Me! & *So Help Me!* trademarks Serengeti Software; Borland Int'l, MicroPro, Ashton-Tate & Microsoft trademarks acknowledged.

Reader Service Number 27



CP/M Notes

Speedy CP/M

Now that there are several super fast CPUs running the Z-80 instruction set, I would like to see a thorough comparative review of the systems that use them (SemiDisk's DT-42, Micromint's SB180FX, and High Tech Research's Ultraboard).

As a Kaypro owner, I welcome these developments in the CP/M arena and hope that they don't get overshadowed by the lemming-like rush to the MS-DOS ocean.

Douglas C. Campbell
17 McMaster Ave.
Toronto, Ontario
Canada M4V 1A8

Editor's note: We're interested in recent CP/M developments, also. Read on for more information.

Ultraboard Status

I spoke with Bill Nesting of High Tech Research recently to see how they were progressing on their Z280 board for the Kaypro. They've solved some problems they had been experiencing with the NCR video chips. However, the NCR chips aren't quite up to the performance Bill wants. He's talking with Hitachi and will make a decision soon. A change to the Hitachi chips would add \$50 to the Ultraboard's price.

An interesting addition to the Ultraboard is Advanced Concepts Engineering's hard disk controller. This option adds another \$99 to the price.

Production quantities of the new "bug free" Z280 are expected by the first of the year. If High Tech goes with the NCR chips, they hope to be producing boards in November. A switch to the Hitachi chips would delay production until January 1. In that case, all boards should contain the new Z280. If boards are produced before the updated CPU is available, High Tech Research will do upgrades for \$40.

High Tech Research
1135 Pine Street #107
Redding, CA 96001

800-446-3220
800-446-3223 (CA)

Micro C Staff

Kaypro Keyboard Fix

During a very humid time of the summer, my Kaypro II-83 keyboard began to misbehave. Every other key on the third row (beginning with TAB) went crazy. At first I thought it must be the excess humidity. After running a dehumidifier for a while things improved—but only temporarily. A visual check of the keyboard circuit board showed nothing amiss, so I began to think about shelling out for a new one. (That was before Kaypro announced the \$20 deal on its surplus keyboards.)

But before sending for a new keyboard, I gave it one more try. I disassembled the keyboard, removing the dozens of little brass screws which hold the printed circuit board to the keys. I gave the circuit board a thorough cleaning with a spray can of cleaner-degreaser and inspected it closely. On that suspect third row of keys, I found the culprit. Little green crystals of corrosion were shorting traces together and causing the keyboard to issue bursts of garbage.

To restore the keyboard to operating condition, I used a tiny jeweler's screwdriver to scrape away the corrosion—very carefully, of course. An orange manicuring stick would work well, too.

When I was sure all the corrosion was gone, I reassembled the keyboard. On powering up the old Kaypro, everything worked properly and I have had no further problems. A local computer repair technician told me that most flaky keyboards can be fixed this simply.

Joseph I. Mortensen
4214 Chelsea Ct.
Midland, MI 48640



CP/M™ Software: Saving The Best for Last

Plus: The Greatest ConIX™ Giveaway

CP/M: Some people love it, others love to hate it, but most still use it. Its users complain that most software companies have abandoned it. Very true, yet we haven't! We've been selling the ConIX software line for many years; we developed it, we market it, and we support it - completely! *What?! You haven't tried it?* Saving the best for last, eh? *Don't wait!* Support your CP/M software company - try ConIX for as low as \$10! What's more, you could even get lucky and receive your entire order **FREE!** See details below.

ConIX™ Operating System

An extensive upgrade for 48K+ CP/M 2.2/3.0 and equivalent systems. Provides professional capabilities with blinding speed, as often found on high-end UNIX™ machines. Installs easily in just minutes to add over 100 new built-in commands and features while maintaining 100% compatibility with all your existing software! Includes I/O redirection, aliases, improved user area access, auto-searching, PF Keys, Screen Paging, Print Spooler, Archiver, New SysCalls, ... Eliminates many points of user frustration with CP/M. Uses only 1/2K TPA, 0-27K disk minimum.

Included **FREE** with commented source is the Pull-Down Menu System, a user-friendly interface to ConIX. Loads with a single keystroke!

ConIX is the greatest, most powerful 8-bit upgrade, with speed and capabilities that are so incredible it's bringing users *back* to CP/M!

ConIX™ Programming System

A structured programming language for ConIX extends CP/M SUBMIT capability. Adds conditionals, loops, subroutines, labels, nesting, interrupt processing, error traps, and debugging facilities. Design intricate menu systems and command-automation shells. Also includes a special source-code "compiler" that provides string and numeric variables. An absolute *must* for CP/M power-users and developers!

ConIX™ Library Vol. I XCC Utilities

Over 25 utilities for ConIX written in the 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!

ConIX™ Shareware Version

A new Shareware version of the ConIX O.S. includes 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. Register by purchasing regular ConIX.

ConIX™ Disk Manual Version

To reduce the cost for those who want to purchase only the ConIX O.S., we are offering the complete software package with documentation provided on disk. The disk manual has each chapter stored in individual files, excluding the Chapter Summary, Chapter Reference, and Index sections that come standard in our regular typeset manuals.

The Great Giveaway: Every 100th ConIX Order **FREE!**

That's right! Every 100th order processed by our computer will be shipped with a Credit Certificate for the total purchase price or \$100, whichever is lower. This credit may be used toward a future purchase from CHI, or may be redeemed for cash within ninety (90) days of receipt. Your odds are an incredible *1 in 100!*

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. To enter, certify eligibility by signing order form.

Name: _____
Company: _____
Address: _____

Tel. #, Hours: _____

End-User Software Licenses:

ConIX Operating System \$ 29.95 \$ _____
Disk Manual Version \$ 19.95 \$ _____
Limited Shareware Version \$ 0.00 \$ _____
Printed Manual Only \$ 9.95 \$ _____

ConIX Programming System \$ 29.95 \$ _____
Printed Manual Only \$ 9.95 \$ _____

ConIX Library | XCC Utilities \$ 24.95 \$ _____
Printed Manual Only \$ 9.95 \$ _____

All ConIX Packages Above \$ 69.95 \$ _____

Computer Brand:

Software Distribution Disk Format: *

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" S_D_TPI Sector \$ 10.00 \$ _____

* Add only one format charge per order.

Format Brand:

Shipping Information:

UPS Ground USA \$ 4.50 \$ _____
US Mail USA (For P.O. Boxes) \$ 6.50 \$ _____
Air Mail Canada \$ 9.50 \$ _____
Air Mail Foreign \$ 12.50 \$ _____

Subtotal: \$ _____
Sales Tax (N.Y. Residents Only): x _____% \$ _____
Total (Thank You!): \$ _____

I Certify ConIX Giveaway Eligibility:

X _____

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 \$. Delivery in 2-4 weeks.

Computer Helper Industries Inc.
Post Office Box 680
Parkchester Station, N.Y. 10462
(212) 652-1786 9AM-5PM M-F



The Kaypro 4-84 Internal Modem

G. R. Blowers

We were somewhat hard pressed for a Kaypro column this issue when, lo and behold, this terse treatise on the Kaypro's modem came in on our RBBS. The original file was dated April, 1984, but the information is timeless.

For those of you doing interesting things with your Kaypros: think about submitting an article to Micro C. We'd like to see what you're up to.

This article documents the internal modem in the Kaypro 4-84. I got the information by decoding the relevant parts of ST.COM (provided on the CP/M disk). Some of the code comes straight from ST and some is code that I used to test the various modem functions.

The 4-84's modem is not a smart type (not Hayes compatible). It consists primarily of two TI chips—an FSK modem, and a dialer. All programming and filtering must be done by the user. The modem makes use of both a parallel and a serial port. Each of these ports must be programmed.

Port Locations

The parallel port lives at 21h, with its command port at 23h. You'll find the serial port at 0Dh, and its command/status port at 0Fh. The modem appears to use the normal modem CTC port of 00H, which needs to be set up for 300 baud (outp(0,5)). I'll divide this discussion into four parts: port bit usage, initialization, modem, and dialing.

I don't intend to decode ST any further than I already have. All I want, or need, is enough to put the internal modem through its paces. Although I can find no copyright notice in ST.COM, I presume Kaypro considers it proprietary. Therefore: No commercial use may be made of this information without prior written consent from Kaypro Corporation.

Parallel Port Bit Usage

The low nibble (bits 0-3) of the parallel port (21h) is used for dialing. Place the number (not

the ASCII) here when dialing. During modem use, send this nibble a 0Ah.

Bit 4 sets the dial mode, reset (zero) for tone and set (one) for pulse. It must be set during modem operation. Bit 6 specifies off/on hook. A reset bit 6 means off-hook and set means on-hook. Setting bit 7 commands the dialer to do its thing.

Serial Port Bit Usage

On to the serial status/command port at 0Fh. Read Register 0 (RR#0) behaves normally except for bit 5. Bit 5 seems to be used for ring detection. It looks at DCD (Data Carrier Detect).

See Figure 2 for initialization of write registers 0-4 (WR#0-4).

In WR#5, RTS (Request To Send) and DTR (Data Terminal Ready) are important. Setting RTS enables the modem. With RTS set and DTR reset, the modem goes to originate mode. The combination of RTS set and DTR set gives the answer mode.

The serial data port lies at 0Dh.

Initialization

The CPU must be initialized to IM2 and interrupts must be enabled. Then load the interrupt vector, and finally, enter the interrupt service routine. Although the code could be anywhere, I chose to use the page 0 restart area. See Figure 1 for the CPU initialization code and Figure 2 for port initialization.

Modem Use

Figure 3 contains code for operating the modem. The dialing routine (Figure 4) comes from ST.COM. I found that a delay was needed between numbers. The code assumes that register A contains a valid number. Remember, this is not the ASCII representation of the number, but the actual binary number, zero through nine.

All filtering and selection of tone or pulse mode dialing must be done prior to entering the dialing loop. I won't go into the detection of carrier or switching to the modem mode as they are straightforward. Normal conventions

Figure 1 - CPU Initialization

;this routine (until RET) must be called during initialization
;of the main program. The whole thing must be loaded prior to
;being called.

```
0008 XOR A ;set A to 0
0009 LD I,A ;hi byte of interrupt vector
000B IM2 ;PIO set up to use IM2 only
000D EI ;enable them
000E RET
000F DB 00 ;skip to make next hit even
0010 DW 0012H ;vector to next word on interrupt
0012 EI ;do nothing interrupt service
0013 RETI ;go back where we were
```

Figure 2 - Port Initialization

```
outp(0X23,0X0F); /* set port to output mode */
outp(0X23,0X87); /* enable interrupts */

outp(0X23,0X10); /* set low byte of interrupt vector */
outp(0X21,0X4A); /* initialize port */

outp(0X0F,0x18); /* I used the "normal" SIO initialization */
outp(0X0F,1); /* rather than the one in ST.COM except */
outp(0X0F,0); /* for WR#5 */
outp(0X0F,3);
outp(0X0F,0xC1);
outp(0X0F,4);
outp(0X0F,0x44);
outp(0X0F,5);
outp(0X0F,0x68); /* this is abnormal, but from ST.COM */
call(0x08,0,0,0,0); /* this calls the CPU initialization */
/* I didn't use it during trial runs - */
/* an oversight on my part and it */
/* appeared to work without it. But to */
/* be on the safe side it should be */
/* included */
```

Figure 3 - Modem Operation Code

```
outp(0X21,0X0A); /* the following take the modem off-hook */
sleep(1); /* delays a bit */
outp(0X21,0X1A); /* makes parallel port ready for modem work */
outp(0X0F,0X05); /* selects WR#5 */
outp(0X0F,0X6A); /* selects modem in originate mode outp(0X0F,0XEA) */
term(); /* this function would set answer mode and finally */
/* go to terminal mode of YAM */
outp(0X21,0X4A); /* code to go back on-hook (hangup) */
```

Figure 4 - Dialing Code

```
LD C,A ;put it in C for later
IN A,(021H) ;get present value of port
AND 0F0H ;strip low nibble
OR C ;put number to be dialed in proper place
RES 7,A ;lower dial command
OUT (021H),A ;put it out
SET 7,A ;raise dial command
OUT (021H),A ;put it out
RES 7,A ;lower dial command
OUT (021H),A ;put it out
```

for Rx character available and Tx buffer empty apply. Be sure to include timeouts for no answer or carrier.

I hope that this helps you write the necessary patches to the standard MODEM programs. I don't intend to carry it any further than possibly customizing a version of YAM (Yet Another Modem program).

□□□



Come to
SOG VII!
July 14 - July 16
Bend, Oregon

Improve Your Manual Dexterity

New Desktop Services From Micro C

You've no doubt followed our trials and tribulations as we've come to grips with desktop publishing.

Now that we have the magazine under control we're branching out, sharing our expertise with those of you struggling with manuals, ads, newsletters, whatever.

We'll design a style sheet for you, then take your (ASCII, WordStar, WordPerfect, or other) text, illustrations, and listings, from disk, paper, or our RBBS and put them together. What you get in return are pages ready for the print shop. And you'll get them for less: usually 1/5th the price of the old-fashioned graphics services. (Of course when it's time to revise the piece, the old-fashioned methods aren't even worth considering.)

Services Include:

- Graphic Design
- Graphic Production
- 300 dpi Proofing
- 300, 1270, 2540 dpi Final Art
- Illustrating
- Scanning
- Schematic Drafting
- Technical Editing
- Proofreading

For more information, contact

Carol Steffy or
David Thompson
Micro Cornucopia
PO Box 223
Bend Oregon 97709
503-382-8048



Technical Tips

Ventura Fix

As both a programmer and a desktop publishing specialist, I have been following with great interest the continuing saga of *Micro Cornucopia's* adventures with Xerox Ventura Publisher. I think your descriptions of *Micro C's* implementation of Ventura are among the most informative and best-written articles that I have read on desktop publishing (and as research director of a desktop publishing training center, I read a lot of articles on the subject).

In the November/December issue (#38), your article "Laser Printers, Typesetters, and Page Description Languages" covered some problems you were having with screen fonts in version 1.1 of Ventura. These problems can be fixed quite easily.

Ventura provides screen fonts only for the "common" text sizes (i.e., 10, 12, 14, 18, 24, and 36 point). Text in other point sizes may look rough and may create problems in positioning the cursor, as you have found. New screen fonts can be created using DOS's DEBUG utility. The procedure is as follows.

First, locate a screen font file close to the type size of the new screen font you wish to create. For example, to create an 11 point screen font, you might use the existing 10 point screen font. Font files will probably be found in the directory \VENTURA unless you have extensively modified Ventura's default directory structure.

Screen font files are usually named IBMETxxI.EGA, where xx is the point size. The extension will be EGA in most cases, even if you are using a monochrome graphics adapter.

Copy the existing screen font file to a file with the new size in its name. To create an 11 point screen font you would copy IBMET10I.EGA to IBMET11I.EGA.

Now use DEBUG to modify the new file as follows:

```
DEBUG IBMET11I.EGA
-d
-e 0102
Debug will display .0A—change it to 0B
-w
-q
```

You can use the above procedure to create as many new screen fonts as you need for the point sizes which you normally use. Be aware, however, that each screen font in the system will reduce the memory available for text and graphics. And more screen fonts could slow down the program if the text and graphics have to be swapped in and out of the smaller memory space.

I hope this has been of some help and keep up the good work.

Steven H. Terry
Director, Research Center
Electronic Directions Group
21 East Fourth St.
New York, NY 10003

32 Bit Bugs

Some vendors are shipping machines which contain an Intel 80386 CPU that is reliable only with 16-bit software. Last spring Intel reported a defect in many of their already manufactured 32-bit 80386 CPUs. The defect affects only 32-bit multiplication. Software which uses 32-bit multiplication instructions may give incorrect results. However, all software written for the 16-bit processors will run flawlessly on the 386 CPUs in question.

Today very little software uses the 386 instruction set, so not many users will be affected. Anyone who buys a 386 machine expecting to run 386 software

on it may want to check their CPU before the warranty expires. The restricted chips are marked, "16 BIT S/W only." An Intel rep told me the machine vendor should make good on it. My vendor knew of the problem and agreed to send a replacement. I wonder how many won't discover the problem until 32-bit software hits the market?

Richard Navas
Computer Software Creation
P.O. Box 4305
Bellingham, WA 98227

Low Level Hard Disk Format

A number of folks (ourselves included) have been stymied lately by rude computer behavior during low level hard disk formatting. When using DEBUG to execute the ROM-based formatter on the hard disk controller, the system goes bye-bye. The OMTI controller, and possibly others, exhibit this unfortunate behavior.

The solution to this problem lies in the fact that the formatter's code begins at C800:6 rather than the customary C800:5. So, if your system hangs when you enter G=C800:5, try the following:

- Invoke DEBUG
- Enter "G=C800:6"
- If the format program comes up, you're on your way.

Micro C Staff



ORDERFORM

POSTAGE-PAID SELF-MAILER

Tear out, fold, and staple both ends if check is enclosed.

YES, I WANT TO SUBSCRIBE!

NEW

RENEWAL

	U.S.	CAN./MEX.	FOREIGN
1 yr. 6 issues	<input type="checkbox"/> \$18	<input type="checkbox"/> \$26	<input type="checkbox"/> \$36
2 yrs. 12 issues	<input type="checkbox"/> \$34	<input type="checkbox"/> \$50	<input type="checkbox"/> \$68
3 yrs. 18 issues	<input type="checkbox"/> \$48	<input type="checkbox"/> \$72	<input type="checkbox"/> \$99



Save
24% Off
the
newstand
price

DISKS MS DOS 5¼" MS DOS 3½"
 Other
Specify Disk # and size

DISK TOTAL

OTHER PRODUCTS
Back Issues, T-shirts... specify size

OTHER TOTALS

GRAND TOTAL

CHECK ENCLOSED
U.S. funds drawn on a U.S. bank, please

VISA MASTERCARD

- - -

Are you a current Micro C subscriber? Yes No

To Place Your Order Immediately
CALL: 1-800-888-8087
9-5, M-F, Pacific Time

NAME

COMPANY

ADDRESS

CITY STATE ZIP

TEAR HERE

TEAR HERE

Expires

MICRO CORNUCOPIA

READER SERVICE CARD JAN./FEB. 1988 ISSUE NO. 39

Write in the reader service numbers of any advertisers from whom you would like to receive free information.

NAME _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

FOLD HERE



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

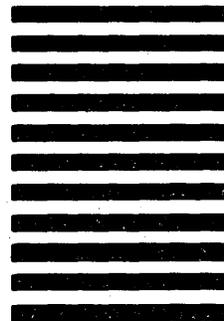
BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 19 BEND, OR

POSTAGE WILL BE PAID BY ADDRESSEE

THE MICRO TECHNICAL JOURNAL

MICRO CORNUCOPIA

P.O. Box 223
Bend, OR 97709-0223



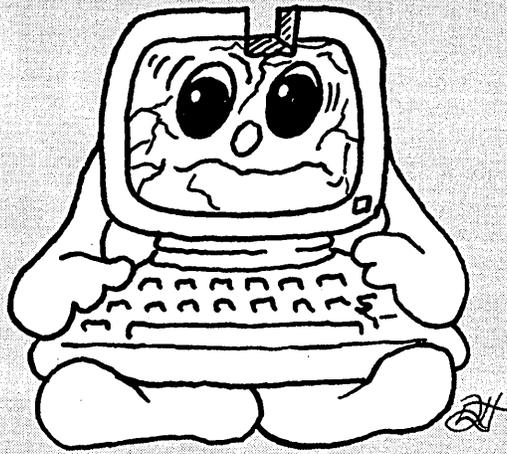
FOLD HERE

**NEXT
ISSUE**

C Language Issue

- C Shootout: A Review Of The Newest C Packages
- Turbo C/Turbo Pascal Comparison
- Source Level Debugger For Turbo C
- Programming the TMS 34010 Graphics Processor: Tricks And Deceptions
- Programming With Microsoft Windows
- More Parallel Processing

STAPLE TO CLOSE



Is There A Gap In Your Info?

Fill in your back issues
of Micro C today

ISSUE #1 (8/81)
Power Supply
RAM Protection
Video Wiggle
½ PFM.PRN
16 pages

ISSUE #2 (10/81)
Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
½ 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 S10s
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
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)
SOLD OUT

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)
SOLD OUT

Issue #11 (4/83)
SOLD OUT

ISSUE #12 (6/83)
256K for BBI
Bringing Up BBI
dBase II
Look at Wordstar
Double Sided Drives for BBI
Packet Radio
5MHz 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 (2/84)
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 BBI
6 MHz On The BBI
Bulletin Boards
Track Buffering On Slicer
4 MHz For The 820-I
64 pages

ISSUE #20 (10/84)
HSC 68000 Co-Processor
DynaDisk For The BBI
Serial Printer On BBI Sans S10
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 DSI-52
SOG IV
The Future Of Computing
MS-DOS In The Public Domain
Graphics In Turbo Pascal
104 pages

Issue #27 (12/85)
SOLD OUT

ISSUE #28 (2/86)
Pascal Runoff Winners
Rescuing Lost Text From Memory
Introduction To Modula-2
First Look At Amiga
Inside The PC
104 pages

ISSUE #29 (4/86)
Speeding Up Your XT
Importing Systems From Taiwan
Prototyping In C
C Interpreters Reviewed
Benchmarking The PCs
104 pages

ISSUE #30 (6/86)
PROLOG On The PC
Expert Systems
Logic Programming
Building Your Own Logic Analyzer
256 K RAM For Your 83 Kaypro
PC-DOS For Non-Clones
104 pages

ISSUE #31 (8/86)
RAM Resident PC Speedup
Practical Programming In Modula-2
Unlinking The PC's Blinkin' Cursor
Game Theory In PROLOG and C
104 pages

ISSUE #32 (10/86)
Public Domain 32000:
Hardware and Software
Writing A Printer Driver For MS-DOS
Recover A Directory By
Reading & Writing Disk Sectors
96 pages

ISSUE #33 (12/86)
Controlling Stepper Motors
From Your PC
Introduction To Fractals
The Secrets Of MS-DOS, From
Boots To Device Drivers
Poking About In The System
With Turbo Pascal
96 pages

ISSUE #34 (2/87)
Designing With The 80386
Build A Simple Oscilloscope
A Cheap 68000 Operating System
A Concurrent Operating System
Recovering Directories And FATs
96 pages

ISSUE #35 (4/87)
Building An 8-channel Temperature
scanner
Designing an Expert System
Teaching Your PC To Beep
Who's Making Great Hard Drives?
Learning Assembly Language
96 pages

ISSUE #36 (6/87)
Build A Midi Interface For Your PC
Designing A Database, Part 2
Interrupts On The PC
Hacker's View Of MS-DOS Vs 3.X
Digital To Analog Conversion, A
Designer's View
96 pages

ISSUE #37 (9/87)
Desktop Publishing On A PC
Build Your Own Hi-Res Graphics
Scanner For \$6.00, Part 1
Designing A Database, Part 3
Controlling AC Power From Your PC
Expanded Memory On The PC/XT/AT
Uninterruptible Power Supply For
RAMdisks
96 pages

ISSUE #38 (11/87)
Parallel Processing
Laser Printers, Typesetters
And Page Definition Languages
Magic in the Real World
Build A Graphics Scanner
for \$6.00, Part 2
Writing a resident program
extractor in C.
96 pages

BACK ISSUES OF MICRO C

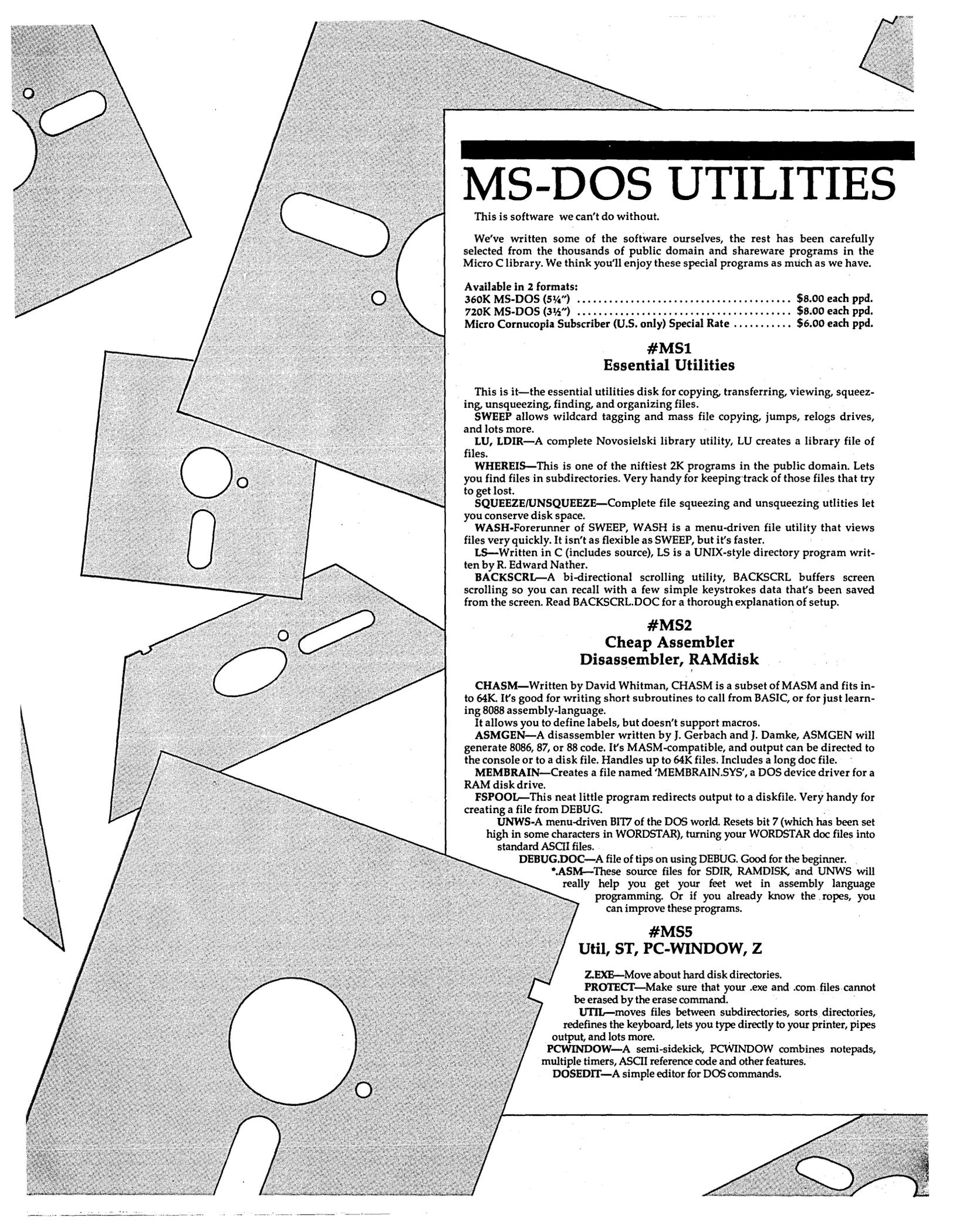
U.S., Canada & Mexico

Issues #1-34 \$3.00 each ppd.

Issues #35 through current issue \$3.95 each ppd.

Foreign (air mail)

Issues #1 through current issue \$5.00 each ppd.



MS-DOS UTILITIES

This is software we can't do without.

We've written some of the software ourselves, the rest has been carefully selected from the thousands of public domain and shareware programs in the Micro C library. We think you'll enjoy these special programs as much as we have.

Available in 2 formats:

360K MS-DOS (5¼")	\$8.00 each ppd.
720K MS-DOS (3½")	\$8.00 each ppd.
Micro Cornucopia Subscriber (U.S. only) Special Rate	\$6.00 each ppd.

#MS1 Essential Utilities

This is it—the essential utilities disk for copying, transferring, viewing, squeezing, unsqueezing, finding, and organizing files.

SWEEP allows wildcard tagging and mass file copying, jumps, relogs drives, and lots more.

LU, LDIR—A complete Novosielski library utility, LU creates a library file of files.

WHEREIS—This is one of the niftiest 2K programs in the public domain. Lets you find files in subdirectories. Very handy for keeping track of those files that try to get lost.

SQUEEZE/UNSQUEEZE—Complete file squeezing and unsqueezing utilities let you conserve disk space.

WASH—Forerunner of SWEEP, WASH is a menu-driven file utility that views files very quickly. It isn't as flexible as SWEEP, but it's faster.

LS—Written in C (includes source), LS is a UNIX-style directory program written by R. Edward Nather.

BACKSCRL—A bi-directional scrolling utility, BACKSCRL buffers screen scrolling so you can recall with a few simple keystrokes data that's been saved from the screen. Read BACKSCRL.DOC for a thorough explanation of setup.

#MS2 Cheap Assembler Disassembler, RAMdisk

CHASM—Written by David Whitman, CHASM is a subset of MASM and fits into 64K. It's good for writing short subroutines to call from BASIC, or for just learning 8088 assembly-language.

It allows you to define labels, but doesn't support macros.

ASMGEN—A disassembler written by J. Gerbach and J. Damke, ASMGEN will generate 8086, 87, or 88 code. It's MASM-compatible, and output can be directed to the console or to a disk file. Handles up to 64K files. Includes a long doc file.

MEMBRAIN—Creates a file named 'MEMBRAIN.SYS', a DOS device driver for a RAM disk drive.

FSPOOL—This neat little program redirects output to a diskfile. Very handy for creating a file from DEBUG.

UNWS—A menu-driven BIT7 of the DOS world. Resets bit 7 (which has been set high in some characters in WORDSTAR), turning your WORDSTAR doc files into standard ASCII files.

DEBUG.DOC—A file of tips on using DEBUG. Good for the beginner.

***ASM**—These source files for SDIR, RAMDISK, and UNWS will really help you get your feet wet in assembly language programming. Or if you already know the ropes, you can improve these programs.

#MS5 Util, ST, PC-WINDOW, Z

Z.EXE—Move about hard disk directories.

PROTECT—Make sure that your .exe and .com files cannot be erased by the erase command.

UTIL—moves files between subdirectories, sorts directories, redefines the keyboard, lets you type directly to your printer, pipes output, and lots more.

PCWINDOW—A semi-sidekick, PCWINDOW combines notepads, multiple timers, ASCII reference code and other features.

DOSEEDIT—A simple editor for DOS commands.

#MS25 Ultra Utilities

The three Ultra Utilities programs will allow you to map disks, unerase files, format non-standard disk, interrogate sectors, and much, much more.

This is a very useful set of utilities (a poor man's NORTON). Many hours of work and frustration can be avoided by learning them, so have at it.

#MS27 System Primer

We think this disk will satisfy at least some of your curiosity about MS-DOS systems programming. We've included lots of assembler source code, so dig in.

SCAV finds and marks bad blocks on both floppies and hard disks.

WHEREIS finds files anywhere within a directory structure.

DIAGS—Special serial, parallel, and video diagnostics for the PC. Use this excellent program to explore your system.

ASYNCR—Loadable asynchronous device driver for MS-DOS.

LPTX intercepts BIOS interrupt 17, the line printer interrupt. It redirects the output of LPT1, LPT2, or LPT3 to a disk file. All three may be active at the same time.

DOS1, ROLLDOS1 & 2, DRIVER & DRIVER1—Stop wondering how device drivers really work and explore these tutorials. Good examples of character device drivers and de-bugging techniques.

STUFIT stuffs your least used files into the inner tracks of the disk. This frees the outer tracks for work space and speeds access times considerably.

#MS36 General Utilities

BATMAKER helps create .BAT files. Perfect when using FIND on all .TXT files, for instance. Very handy.

BWVID lets you see what is happening on the screen when you have a color graphics card (CGA) and a monochrome monitor.

CED is called a Command line EDitor but it's far, far more than that. Includes macro definitions, control of DEBUG, repeating and editing of previous commands, etc.

DEBUG.DOC is a simple but very handy quick reference guide to DEBUG.

EXPAND and **SHRINK** are detab and entab utilities.

PC-STAT—Reports system information—memory available, drive status, etc.

PC-TEST is similar to Norton's speed test, but its test takes longer and it doesn't report such wildly optimistic speed figures.

POPALARM is really neat. It's a memory resident alarm clock that reminds you to do what you'd otherwise prefer to forget.

RECALL remembers the last 50 DOS commands. Commands may be edited and/or reexecuted.

REMINDR—This is a daily black-book that stores its data on disk.

SCR*—Utilities for creating batch files which incorporate screen images. This is a great extension to MS-DOS batch capabilities.

FILTERS—The remaining files are classics from the Software Tools book. One of the real attractions of these filters is that they come with assembly language source.

#MS37 Disk Utilities

COVER prints out directories in compressed format to be pasted on floppies.

CRC67—Finally, an MS-DOS cyclic redundancy checker (CRC) that works (Fast!). Checks CRC values for files against a previously recorded list of CRC's.

DISKORAY checks floppy rotation speed and allows stepping of the head.

DISKPARK parks the heads of all hard disks in your system using the innermost track.

DISKWIPE—Be careful. This completely erases a disk, including the formatting.

FDATE allows editing of the time and date stamp on DOS files.

FILES—A very complete directory program.

MOVE2—Intelligent COPY routine.

REFRESH rerecords data on a disk. It does 12 retries on reads and 2 on writes so it may be able to recover those "bad" sectors.

SDIR—Version 5.0 of the super directory program.

SST—Just what every busy hard disk needs. SST reorganizes files into contiguous sectors on the disk. This really speeds up disk accesses.

TIMEPARK parks the heads on a running hard disk after a user specified amount of time without accesses.

WD*—Everything you always wanted to know about Western Digital's WD 1002S-WX2 hard disk controller. Also information on optimizing its performance with the Seagate ST225 drive.

WHEREII searches for one or more files through all the directories on a specified drive. Supports wild-cards.

New! #MS39 Modula-2 Compiler

This disk contains an excellent shareware implementation of Modula-2 from Fitted Software Tools. It sports an integrated editor, linker, and make utility. Requires 512K of RAM.

New! #MS40 More Utilities

BOOTDUMP—Gives information on the currently logged drive—sector size, sectors/track, # of heads, etc.

BOX—Boxes text displayed in a batch file. Includes C source.

KBBUFF—Extends type ahead buffer by 160 characters.

KOLOR—Sets foreground, background, and border colors from the DOS prompt.

LURUN—Executes programs within an LBR format library file.

NJRAMD15—Sets up RAMdisk in EMS or EEMS memory.

OV143—Overview is a great file manipulation program. Somewhat like XTREE but much faster since it doesn't read in the entire directory structure unless necessary. Includes source.

REBEEP—Beeps until a key is pressed. Quite useful for batch files.

SIMCGA—Excellent utility which allows CGA programs to run on Hercules systems.

WAITN—Use in batch files to pause a specified amount of time.

For Technical Help:
(503) 382-8048 (9 a.m.—noon, PST, M-F).

Order today from:

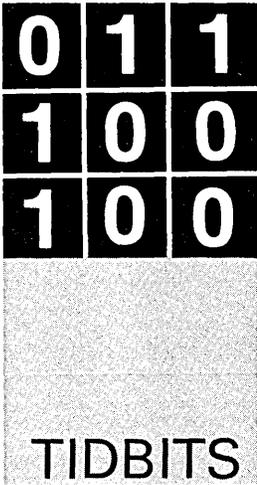
MICRO CORNUCOPIA



P.O. Box 223 • Bend, Oregon 97709

Phone orders: (503) 382-5060





Dots, Lines, And Fills: Programming The EGA

By Gary Entsminger
1912 Haussler Dr.
Davis, CA 95616

Gary's been wishing for EGA graphics ever since he got his X16. Now that he's got it, what can he do with it? Use it, what else? Herein you'll find out how he writes EGA graphics routines.

Although the EGA (enhanced graphics adapter) has been coloring the monitors of some PC users for over three years now, I've just gotten 'round to learning how to use it. For those of you coming into the game late, the EGA is a sweet step up from CGA, with most of the Hercules adapter's resolution. In enhanced mode (16), its resolution is 640 by 350 pixels, 64 colors, 8 by 14 cells (or characters), and 256K RAM which will support up to 8 graphic pages).

Figure 1 - Modes

EGA Mode	Txt/Graph	Adapter	Colors	Resolution
0	T	C	16	320 By 200
1	T	E	16/64	320 By 350
2	T	C	16	640 By 200
3	T	E	16/64	640 By 350
4,5	G	C,E	4	320 By 200
6	G	C,E	2	640 By 200
7	T	M	4	720 By 350
13	G	C,E	16	320 By 200
14	G	C,E	16	640 By 200
15	G	M	4	640 By 350
16	G	E	16/64	640 By 350

The colors are fine, the text is fine—but cheap, the EGA system isn't. Mine (Daetech adapter and TVM color monitor) ran about \$560 at McTek, the best price I could find in the Bay Area. (But I expect you folks traveling even further west could make a better deal.)

Not The First Thing, But...

Not first, but early on, I decided to learn how to program the adapter. I've been itching to add some colorful bells and whistles to my programs for ages, so I began to delve.

The first problem (as usual) was lack of information. The Daetech manual is a surprising-

ly interesting 80 pages of details, but a tutorial it isn't. It references the IBM Personal Computer Seminar Proceedings, the EGA Chip Set Data Sheet, and a BYTE article from 1985 written by FORTHER Richard Wilson.

The Wilson article, "Programming the Enhanced Graphics Adapter," is excellent, and filled with gems like, "Unfortunately, programming the EGA is not a trivial task. The hardware has a complex, sophisticated, and somewhat idiosyncratic architecture, but it is rarely obvious from the technical documentation just what all that complex hardware does."

Naturally, Wilson had me interested, and got me as far as reading and writing a pixel in assembly language. Not bad, but I wanted something a little gentler.

So I headed to the UC library, the Walden bookstore, and a few of my other favorite hangouts to rustle up a little more presentable information.

From these outlets and Osbourne/McGraw-Hill publishing, I netted several good sources, including *Advanced Turbo C*, by Herbert Schildt, and *Advanced Graphics in C*, by Nelson Johnson. "Advanced" anything that I can read is flattering.

Modes, Masks, And Programming

In order to make sense of this discussion, you need to know at least a little about modes, colors (that's easy), addresses, registers, interrupts, lines, ports, masks, pointers, offsets, and pixels.

Modes are built into PC-DOS in 16 flavors (see Figure 1). A few CGA and Monochrome modes work only with those respective adapters; but the EGA, in general, can emulate most CGA and monochrome adapters in "their" modes, making it an all-purpose adapter.

You can use the PC-DOS MODE command to change some modes, but not the ones created especially for the EGA. To change those you'll need a programming language equipped for the low levels. I used C, changing the mode by calling BIOS interrupt 10 hex (it's intended for this and similar video purposes). Set the AH &

AL registers and call it:

```
void set_mode (mode)
int mode;
{
    union REGS r;

    r.h.al = mode;
    r.h.ah = 0;
    int86 (0x10, &r, &r);
}
```

Once we've set the mode, we can draw by turning pixels on and off. A point is a pixel (actually the nearest pixel to the coordinates of the point) turned on. A line consists of points. A box consists of lines. A bar is a filled box.

We can tell the EGA which pixels to turn on and off by calling the BIOS or by talking to the EGA directly. The first method is easier; the second is more efficient.

In the first, we write a function much like the one we just wrote (above). Once again we call interrupt 10 hex (using the C system function `int86`) after setting register values:

```
void bios_dot(x,y,color)
int x,y,color;
{
    union REGS r;

    r.h.ah = 12;
    r.h.al = color;
    r.x.dx = y;
    r.x.cx = x;
    int86(0x10, &r, &r);
}
```

And that simply is all the low-level details you have to know—call the BIOS to set the mode; call the BIOS to write the pixels. Every other drawing algorithm (including the ones that plot for days those elegant fractals) depends on dots (or points or pixels).

Which brings up a point. How fast is a function which calls the BIOS? In this case—not very. Whatever it takes at the system's level to save us the trouble of handling a task directly, takes time. When we use the BIOS we gain simplicity (or even capability) and lose efficiency. (If you want to skip ahead to the results—Figure 2—you can see just how the trade works.)

A more efficient method of drawing points bypasses the BIOS and calls the EGA directly by sending a function number to the EGA's INDEX REGISTER, which is a port.

The EGA's display buffer begins at address A0000000L. Specific memory

Figure 2 - Times

	Mode	Screen Size	Fill 1/2 Screen
BIOS	14	640 By 200	47 sec.
BIOS	16	640 By 350	80 sec.
DIRECT	14	640 By 200	15 sec.
DIRECT	16	640 By 350	25 sec.

System -- 80186-based PC (8 MHz)

locations are offsets from this address. (For those of you who can relate to the CGA or Hercules adapters—the CGA's display buffer can be found at B8000000, and the Hercules at B0000000.

The display resolution can vary (see Figure 1) up to a maximum of 640 by 350 pixels. We address video memory, though, in bytes. See the base equation in Figure 3 to write a point directly to EGA memory.

The EGA display buffer corresponds to a page, or screen. The display buffer

Figure 3 - Write a Point to Memory

```
#include <stdio.h>
#include <dos.h>

#define TRUE -1
#define ENABLE 0x0F
#define INDEXREG 0x3CE
#define VALREG 0x3CF
#define OUTINDEX(index, val)
    {outp(INDEXREG, index); \
    outp(VALREG, val);}

#define EGA_BASE 0xA0000000L
#define WIDTH 80L
#define XMAX 639
#define YMAX 349
/* for mode 16; 199 for mode 14 */
#define XMIN 0
#define YMIN 0
#define XORIT 0x80

point_mem(x, y, color, orxor)
int x,y, color,orxor;
{
    unsigned char mask = 0x80,
    exist_color;
    char far *base;

    if (x < XMIN || x > XMAX
        || y < YMIN || y > YMAX) return;

    base = (char far *) (EGA_BASE +
        ((long)y * WIDTH + ((long)x /
        8L)));

    mask >>= x / 8;

    exist_color = *base;

    OUTINDEX(0, color);
    OUTINDEX(1, ENABLE);
    OUTINDEX(8, mask);
    OUTINDEX(3, 0x18);

    *base ^= TRUE;

    OUTINDEX(0, 0);
    OUTINDEX(1, 0);
    OUTINDEX(8, TRUE);
    OUTINDEX(3, 0);
}
```

stores values for the screen beginning at the upper left-hand corner, the point (0,0). A byte offset is equal to the HEIGHT (the y coordinate) times the WIDTH of the screen plus the x-coordinate divided by 8 (bits). Base is a far pointer to this location.

We set and reset the INDEXREG and VALREG registers with the macro OUTINDEX. The code in Figure 3, adapted from *Advanced Graphics in C*, will draw points very quickly in modes 14 and 16. A little more modification (moving the reset out of the function and into `main()`) will speed it up considerably if you're drawing and filling complex shapes and such.

Lines And Such

The code in Figure 4 is a version of Bresenham's integer-based algorithm for drawing lines. I think the code (adapted from *Advanced Turbo C*) is self-ex-

Figure 4 - Line Using Mem Writes

```
void line(start_x, start_y,
end_x, end_y, color)
int start_x, start_y, end_x,
end_y, color;
{
    int t, distance;
    int xerr = 0, yerr = 0,
    int delta_x, delta_y;
    int incx, incy;

    delta_x = end_x - start_x; /*
    Get the change in X */
    delta_y = end_y - start_y; /*
    Change in y */

    if (delta_x > 0) incx = 1; /*
    If a change, inc or */
    else if (delta_x == 0) incx =
    0; /* decrement X. */
    else incx = -1;

    if (delta_y > 0) incy = 1; /*
    Same for y */
    else if (delta_y == 0) incy = 0;
    else incy = -1;

    delta_x = abs(delta_x); /*
    X or Y changing faster? */
    delta_y = abs(delta_y);
    if (delta_x > delta_y) distance
    = delta_x;
    else distance = delta_y;

    for (t = 0; t <= distance + 1;
    t++) {

        mem_point(start_x, start_y, color, 0x1
        8); /* Plot point. */
        xerr += delta_x;
        yerr += delta_y;
        if (xerr > distance) {
            xerr -= distance;
            start_x += incx;
        }
        if (yerr > distance) {
            yerr -= distance;
            start_y += incy;
        }
    }
}
```

Figure 5 - Draw Line Using BIOS

```

void bios_line(start_x, start_y,
              end_x, end_y, color)
int start_x, start_y, end_x, end_y,
  color;
{
  /* register */ int t, distance;
  int xerr = 0, yerr = 0, delta_x,
    delta_y;
  int incx, incy;

  delta_x = end_x - start_x;
  delta_y = end_y - start_y;

  if (delta_x > 0) incx = 1;
  else if (delta_x == 0) incx = 0;
  else incx = -1;

  if (delta_y > 0) incy = 1;
  else if (delta_y == 0) incy = 0;
  else incy = -1;

  delta_x = abs(delta_x);
  delta_y = abs(delta_y);
  if (delta_x > delta_y) distance =
    delta_x;
  else distance = delta_y;

  for (t = 0; t <= distance + 1;
       t++) {

    bios_dot(start_x, start_y, color);
    xerr += delta_x;
    yerr += delta_y;
    if (xerr > distance) {
      xerr -= distance;
      startx += incx;
    }
    if (yerr > distance) {
      yerr -= distance;
      starty += incy;
    }
  }
}

/* FILL boxes */

void fill_box(start_x, start_y,
              end_x, end_y, color)
int start_x, start_y,
  end_x, end_y, color;
{
  /* register */ int i, begin, end;
  begin = start_x < end_x ? start_x :
    end_x;
  end = start_x > end_x ? start_x :
    end_x;

  for(i=begin; i<=end; i++)
    line(i, start_y, i, end_y, color);
}

/* FILL using the BIOS */

void bios_fill(start_x, start_y,
               end_x, end_y, color)
int start_x, start_y,
  end_x, end_y, color;
{
  /* register */ int i, begin, end;
  begin = start_x < end_x ? start_x :
    end_x;
  end = start_x > end_x ? start_x :
    end_x;

  for(i=begin; i<=end; i++)
    bios_line(i, start_y, i, end_y, color);
}

```

planatory, but you can delve deeper into Bresenham's method by perusing either of these fine books.

Figure 5 brings Tidbits toward a conclusion by using points and lines to fill boxes. The only difference between the two fill functions is how they draw dots (indirectly via the BIOS or straight to the adapter). Fills take time, of course, even on ATs and the like, and gave me a chance to test the functions.

Out Of Here

Figure 6 is the main testing function. I filled BIOS boxes and direct write

boxes in modes 14 (640 by 200) and 16 (640 by 350). The results, as I said, are in Figure 2.

In sum (on my system), direct writes to the EGA display buffer (bypassing the BIOS) are a little better than three times faster than BIOS writes. The 640 by 200 display (which isn't quite as fancy) fills up a little less than twice as fast as the 640 by 350. The slow-mo BIOS was a lot more noticeable.

I'm out there, it's 4 a.m., and that's Tidbits.



Figure 6 - Main Test

```

main ()
{ int x,y;

  set_mode (14); /* or 16 */
  printf ("Tidbits 39 test - hit any key to start");
  getch();
  for (x = 0; x < XMAX; ++x)
    bios_dot(x,100,3); /* to cross the screen */
  for(y = 0; y < YMAX; ++y)
    bios_dot(320,y,4); /* " " " " */
  getch();
  for (x = 0; x < XMAX; ++x)
    mem_point(x,105,7);
  for(y = 0; y < YMAX; ++y)
    mem_point(325,y,9);
  getch();
  printf("start");
  fill_box(0,0,649,99,4);
  printf("stop");
  getch();
  printf("start");
  bios_fill(0,100,649,199,3);
  printf("stop");
  getch();
  set mode(3); /* reset text mode */
}

```

See Micro C At The West Coast Computer Faire

We'll Enjoy Seeing You

Come and meet some of the Micro C crew at the West Coast Computer Faire, April 7 - 10. We'll have magazines, T-shirts, and ads for sale. Plus we'll be glad to talk about article ideas or anything else. Our booth is (tentatively) #951.

You're Looking At The Best Advertising Bargain In The Industry



When You Advertise In Micro C You Get:

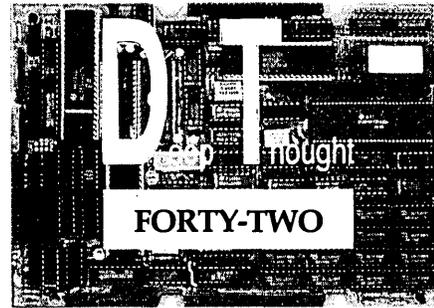
- **A knowledgeable audience.** Advertisers say they can tell immediately when they get an order from a Micro C reader. Our readers are better informed than the readers of any other publication, including Byte.
- **An audience that misses nothing.** No matter what size ad you pick, your message won't get lost in Micro C. Over 90% read every word and every ad in every issue.
- **An active audience.** Many advertisers report more sales from their Micro C ad than from their ad in any other magazine, regardless of price, regardless of audience size.
- **The lowest cost per thousand readers of any journal.** Compare our rates with Byte, Dr Dobbs, Computer Language... with any other magazine. Then compare total readership. Micro C is the one place where you can do serious advertising on a budget.

For rates and deadlines, and other fun particulars, call Laura Logan at:

503-382-8048

Micro C — Serious Advertising On A Budget

THE MICRO TECHNICAL JOURNAL
Micro Cornucopia



The "thoughtful" alternative from *SemiDisk*.

Designed around the 64180 microprocessor, the DT-42 is loaded with *more* of all the features you need: More speed, more memory, more ports and more TPA!

How did we fit all these features on one 5.75" by 8" single-board computer?

- 9.216MHz 64180 Microprocessor (runs Z80 programs)
- 512K DRAM, Zero wait states, fully populated.
- Three RS232C serial ports (Standard baud rates to 38,400)
- One Centronics parallel printer port
- WD2793 disk controller (up to 8 drives, SD, DD or High Density, 3½", 5¼", and 8" drives)
- SASI channel for hard disk controller (software provided)
- Many popular disk formats supported
- Requires only +5V @ 1 amp.
- ZRDOS/ZCPR3 with exclusive "Hyperspace" operating system, offering 57.5K TPA (NOT 48K like some others. *No 8 bit is bigger!!*) Richard Conn's ZCPR3, The Manual included free!
- Provisions for real-time clock and on-board terminal options.
- Socket for 28-pin EPROM.

Compare! You won't settle for less. Or slower. Or smaller.

DT-42 Computer	\$ 499
TMP (on-board terminal)	\$ 100
SmartWatch	\$ 50
Z-system software	\$ 50
ZAS & Debuggers	\$ 25
8MB disk emulator w/ SCSI	\$ 2049
Battery backup for above	\$ 150

Call or write for more
information or to place an order.

SemiDisk

11080 SW Allen Blvd., Beaverton, OR 97005
(503) 626-3104

Reader Service Number 18

CALL FOR FREE CATALOG

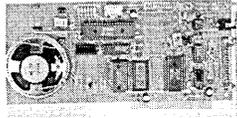
TEXT TO SPEECH BOARD!

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!

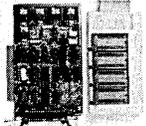
PRICE CUT!

\$6995
ASSEMBLED & TESTED

NEW! IC TESTER! \$149.00

SIMILAR TO BELOW EPROM PROGRAMMER. PLUGS IN TO YOUR PC OR XT. TESTS ALMOST ALL 14, 16, AND 20 PIN 74XX SERIES. INCLUDES STANDARD POWER, "S" AND "LS" DEVICES. ALSO TESTS CD4000 SERIES CMOS. SOFTWARE INCLUDED CAN EVEN DETERMINE PART NUMBERS OF MOST UNMARKED AND HOUSENUMBERED DEVICES WITH SIMPLE MOD. THIS UNIT CAN ALSO TEST 6.4K AND 256K DRAMS! WITH MANUAL AND SOFTWARE: \$149. PERFECT FOR SCHOOLS.

PC/XT EPROM PROGRAMMER \$169



ASK ABOUT OUR NEW PAL PROGRAMMER!

* 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 TEXTOL 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. \$169 WITH SOFTWARE.

JUST RECEIVED. SAME AS ABOVE PROGRAMMER, BUT PROGRAMS 8 UNITS AT ONE TIME - \$299.

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

TERMS: Add \$3.00 postage. We pay balance. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Texas Res. add 6-1/4% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

Reader Service Number 32

XEROX 820-1 AND 820-2 ITEMS

Reconditioned, Assembled and Tested

820-1 8" COMPUTER SYSTEM.....	\$330.00
5 1/4" COMPUTER SYSTEM.....	\$350.00
820-2 8" COMPUTER SYSTEM.....	\$395.00
5 1/4" COMPUTER SYSTEM.....	\$415.00
820-1 COMPUTER MONITOR (COMPLETE).....	\$125.00
820-2 COMPUTER MONITOR (COMPLETE W/ CONTROLLER).....	\$195.00
820 COMPUTER MONITOR (NO MAIN BOARD).....	\$ 85.00
HIGH PROFILE KEYBOARD (COMPLETE).....	\$ 45.00
820-1 MAIN COMPUTER BOARD.....	\$ 50.00
FULLY POPULATED BOARDS, AS IS (NEED REPAIR).....	\$ 20.00
820-2 MAIN COMPUTER BOARD.....	\$ 70.00
FULLY POPULATED BOARDS, AS IS (NEED REPAIR).....	\$ 30.00
820-2 FLOPPY CONTROLLER BOARD.....	\$ 95.00
DUAL 8" SSDD DISK DRIVES/ENCLOSURE (COMPLETE).....	\$175.00
DUAL 8" DISK DRIVE CABINET (NO DRIVES).....	\$ 75.00
5 1/4" DUAL DISK DRIVE CABLE.....	\$ 20.00
8" DUAL DISK DRIVE CABLE.....	\$ 35.00
RS-232 CABLES.....	\$ 10.00

LINE CORDS.....ea.	\$3.00
Z80-B 6MHz.....ea.	\$3.00
Z80-H 8MHz.....ea.	\$9.50
5 1/4" DSDD DISKETTES.....ea.	\$.60
8" SSDD DISKETTES.....ea.	\$1.25
DC300A DATA CART..USED...2/	\$5.00

E2I COMPUTER PRODUCTS
2273 AMERICAN AVE. #8
HAYWARD, CA 94545
(415) 786-9203

TERMS: Pre-payment, COD, Visa/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. All products are assembled and tested and have a 30 day warranty unless otherwise stated. Call or write for current product and price listing. Xerox is a trademark of Xerox Corporation. CP/M is a trademark of Digital Research.

Reader Service Number 33

AROUND THE BEND

(Continued from page 2)

this data communications program automatically routes calls via free phone lines into the most interesting systems in government and industry. If the intrusion is discovered, the program identifies itself as a CIA operative, leaves George Bush's home number, and signs off. An especially good gift for someone doing time.

(3) *The Disputer*. This computer was designed by VAI (Very Artificially Intelligent) Corporation. The system takes the opposing side of any argument. If two sides are already taken, the system creates a third. The latest model is powerful enough to destroy any and all opposing ideas, especially new ones. After producing the first working model, the company has been unable to decide how to manufacture it.

On The Desktop Front

Local Apple dealers are insisting that Macs are the only way to do desktop publishing. They have some interesting sales points:

- IBMs are too hard to use. (Hey, these guys should know, they were selling heavy iron until a local used car lot folded. But they have a point and will continue to have one until OS/2 or some other windows environment catches on. Plus, the salesmen aren't interested in learning DIR and COPY.)
- Apple has better desktop software. (Not true, Ventura Publisher and PC Pagemaker are certainly equal to, if not ahead of, anything available for the Mac. But, the Apple's DP stuff is very good and getting better.)
- Mac has more powerful packages for creating, scanning, editing, type, and graphics. (That's true and will probably continue to be true—at least until Adobe comes out with a PC version of Illustrator.)

See the Desktop Graphics article in this issue for a look at how we're dealing with graphics at Micro C.

Micro C's Seventh Annual Technical Forum

If you're tired of the same old technical forums with their high prices, crushing traffic, crime, and three-piece-suits, then plan to be in Bend July 14 through 16 for SOG VII. (Our Seventh Annual Semi-Official Get-together.)

As usual, we'll have white water rafting (Thursday) followed by two days of technical talks at Central Oregon Community College. No dorms this year, they'll be closed for remodeling, but we'll be glad to send you information on motels and camping.

Want to speak? Call, write, or log onto the Micro C RBBS (503-382-7643). Include a short outline and a two or three paragraph description of your subject, your name, address, day and evening phone. Special things await speakers (T-shirt, afternoon raft trip...).

Many SOG VI attendees said they got more real information than they received at any of the high-priced (\$300+) conferences. Ask someone who's attended. They'll tell you.

Looking For Groups

Are you a member of a computer group whose members should know about *Micro C*? Send us the name of the group along with the name, address, and phone number of the librarian or president and an estimate of the number of attendees. We'll send the group some copies of *Micro C* for their library along with some subscription forms and posters to pass out.

ALL SALES SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. FREE COPY UPON REQUEST.

And, for your help, we'll gladly send you any Micro C user disk. Free. Along with our thanks.

Sinclair QL

I've received four letters after my request for QL feedback. One of the letters contained two issues of the "Detroit Area Times/Sinclair News" (DATSN). The newsletter covered: vanishing newsletters, shrinking user groups, and disappearing commercial outlets. It was a requiem mass quite familiar to CP/M users.

DATSN reprinted a letter written to the Vancouver Sinclair Users Group (hey, when letters get that hard to come by...).

"This man (Clive Sinclair) took the most sophisticated microprocessor (68008) and completely ruined it by putting it in a QL. The only machine in the world that uses not hard drives, not floppies, but mini 8-tracks. The display is weird, the keyboard is weird, and the operating system is weird."

(Meanwhile, I understand that Quantum is closing out its stock of QLs at \$99 each. If nothing else, they're cheap. Awfully cheap.)

The Active Core

Most systems are fortunate enough to have two types of users. The first type is a hacker, innovator, fanatic. These are the people who grabbed the first Apple IIs, the S-100 systems, the Big Boards, even the early PCs.

They upgrade hardware, write utilities, and poke about in the operating system. These are also the people who start users groups, write newsletters, teach classes, run bulletin boards, and put together software libraries. They are the founders and officers, volunteers and instructors who make up the hearts of local groups.

The second user is the appliance operator. He, too, owns an Apple, an S-100 system, a Big Board (in the form of a Kaypro or Xerox), or a PC: but he's a consumer, not a mover. He'll pay the dues, take the classes, log onto the bulletin board, and attend the meetings. Occasionally, he'll even buy the doughnuts.

If a system is to become a financial success, the appliance operators will have to outnumber the shakers 100 to 1. The operators purchase most of the hardware, and add-ons, and software. They're the ones who make manufacturers happy.

As a system matures, however, the shakers get itchy. Their interests turn to new systems: just as they were once interested in Apple IIs and Big Boards, they continue to be interested in new systems.

The user group leaders, who spread the word, teach the floundering, chair the meetings, and repair the systems, begin moving on. Holes show up in the officer corps. For a while, there will be new shakers to step in, but not forever.

Sure, there are names on the rolls, lots of names, but the organization has lost its heart. Eventually, attendance begins to decline and the topic of discussion shifts from exploring an exciting new system to survival of the group.

If it's going to survive, it will have to change its focus: either tackle an exciting new system, or become a food SIG. Either way is food for thought.

On The Same Note

Morrow Owners Review (MOR), the bimonthly Morrow magazine, is calling it quits. They're busy notifying all readers that the December issue will be the last.

MOR had 4,000 readers in June, 1987. By September, the subscriber base had dropped to 2,400. Meanwhile, the Morrow

The Dream-88

30 Mb XT Compatible

XT Turbo, 4.77-8 MHz motherboard
640 K of on board RAM
1 5.25" 360K Fujitsu floppy drive
30 Mb Seagate ST-238 Hard drive
Floppy controller card (controls 2)
Hard disk controller card
12" amber Samsung monitor (tilt/swivel)
Monographics card with parallel port
AT style keyboard
XT slide case (UL and FCC approved)

Complete! \$795⁰⁰

The Dream-286

40 Mb AT Compatible

AT 6-10 MHz 0 wait state motherboard
1 Megabyte of on board RAM
1 5.25" TEAC 1.2M floppy drive
40 Mb Seagate ST-251 Hard drive
Hard/floppy disk controller card
12" amber Samsung monitor (tilt/swivel)
Monographics card with parallel port
AT style keyboard
AT case (UL and FCC approved)

Complete! \$1495⁰⁰

All systems can be customized
to your exact specifications.

Everex 1200 Baud Modem	\$ 95
Everex 2400 Baud Modem	\$ 195
Magnavox Multimode EGA Monitor	\$ 567
Genoa Super EGA High Res (800x600)	\$ 275
Micropolis 71 Mb Hard Disk	\$ 849
Mountain 40 Mb Tape Backup Unit	\$ 455
Toshiba 720K 3.5" Floppy Drive	\$ 135
Ventura Desktop Publisher	\$ 465
Word Perfect Version 4.2	\$ 239
Norton Utilities (Advanced)	\$ 89
Turbo Basic	\$ 65
Turbo Pascal	\$ 65
Turbo C	\$ 65
Citizen 120D / 180D Printer	\$ 189 / 215
Citizen MSP40 / MSP45 Printer	\$ 379 / 495
Citizen Overture 10 page/min laser (laser printer includes toner)	\$ 1595

DreamTech (408) 996-2373
5175 Moorpark Avenue San Jose, CA 95129

Reader Service Number 16

user group (Bay Area Micro Decision Users Association) will merge with the Bay Area Kaypro Users Group.

Scanner Parts

Some of you have had trouble locating the OPB708 photosensor for John Jones' \$6 scanner. The part is \$2.40 and is available from:

Optron, Inc.
1201 Tappan Circle
Carrollton, TX 75006
(214) 323-2200

Since their minimum order is \$1,000, you might want to get it from one of their distributors: Arrow, Hamilton Avenet, Halmark, Pace Setter Electronics, Pioneer, and Bell Industries. If you have trouble locating one of these, call Optron and they'll put you in touch with someone in your area.

John mentioned that Texas Instruments has an equivalent part. It's part numbered TIL 139.

PC Network

Over the past year I've had several conversations with PC Network. Each time I've called them because a *Micro C* reader was having problems getting PCN to ship products. (No, the Network hasn't advertised in *Micro C*, helping readers is just a hobby.)

Each time PC Network's response was cordial and each time they insisted that slow deliveries (four months and counting) resulted from problems with suppliers. However, they didn't seem worried enough about these problems to stop advertising the scarce products, nor were they shy about cashing our readers' checks.

Recently, PC Network signed a consent decree with the Federal Trade Commission agreeing to give customers the right to cancel orders and agreeing to honor warranties. According to the FTC, PC Network paid \$61,000 in civil penalties as part of the deal.

However, PC Network is also under Chapter 11 protection from creditors (as of mid-September). Creditors include customers who haven't received what they've ordered.

On the other hand, if you're having trouble with an order or with warranty service from PC Network, you're not necessarily stuck. Contact the Chicago branch of the FTC at (312) 353-4423. They might be able to help.

More Hard Drives, More Hard Problems

I just received a call from Dusty Johnson at Rotating Memory Service, a large winchester drive repair shop. He had read my hard drive piece in issue #35.

"Absolutely true, every word," he said. (Obviously a man of culture and distinction.)

"We're seeing lots and lots of 225s and 251s, but that's not all. There's a design problem with the Seagate 4000 series. Sit one of those full-height monsters out on the bench, apply power, and then remove it. The head retracts so fast the drive moves. Actually jumps.

"The head assembly bangs so hard during the auto-retraction that the heads bounce against the platters. Not the best way to treat heads or platters."

He mentioned to me that the stepper motor shaft on the 225s isn't swedged into place—they epoxy it in and eventually you get stepper shaft wobble. Finally, the head doesn't line up with the track. At that point you can't recover your data. Period.

"On the other hand, Seagate has done some very good things, they almost never have head crashes, and they use very good media. If they'd put in a good motor and a track 0 sensor, they'd have a great drive."

He said he'd seen a lot of Microscience 725s come in for repair but none of the newer 825s. Maybe that bodes well for the 825 we've got on the RBBS. He also confirmed that Miniscribe is building great drives. Even their cheap ones.

ERAC

ERAC Company, supplier of parts and boards for Kaypros and such is not running its ad this issue. They've been having trouble getting orders shipped.

Our Slick, Glossy Cover

No excuses this time. No one made me do it. I'm guilty.

When I saw those beautiful fractals on PC Tech's color graphics board at SOG VI, I knew, eventually, I was going to have to print color. Well, here it is. The cover photo was photographed by Dean Klein directly off a PC Tech monitor. He sent 36 fractal shots. I wish you could see them all. (Maybe someday when we have two pages of color.)

Thanks Dean, I guess I owe you one.

(Actually, I owe him two. He's just finished up a 286 system which matches most of the specs I gave him for a 386 system. 16 MHz, works with ALL plug-in boards, SCSI interface, etc. etc.)

On Your Own

The graphics article in this issue started out as a desktop piece, changed into an On Your Own piece, then shifted back into a desktop piece. It's both, and long enough to include the C column too. If you want to read "On Your Own," read "Desktop Graphics." (Which reminds me. Next issue will be a C issue.)

As I write these very words, Scott Ladd is finishing up a comparison of all the actively marketed MS-DOS C compilers. (Including Borland's 1.5 and Microsoft's 5.0.)

He's benchmarking compile time and run time, checking compatibility with standards, trying out user interfaces, debuggers, utilities, and he's looking (line for line) at the code each compiler generates. (Whew!) I've had a peek at some preliminary results and I'm surprised. Very surprised.

Until next issue, when I answer the burning question: "What was the most anticipated new product which didn't make it to Comdex," that's all from greater Bend.

David Thompson
Editor & (Ventura) Publisher





PASCAL PROCEDURES

By John Paul Jones

6245 Columbia Ave.
St. Louis, MO 63139
(314) 645-1596

Logitech's Modula-2 Version 3.0 & Scanner - III

John mentions two new products, Logitech's new Modula, which he has—and Borland's new version of Pascal, which he doesn't have. And he's down to the nitty gritty in the scanner code. It's getting interesting folks.

Logitech has released a new version of their Modula-2 development system, and at first look it appears to be a winner. The compiler alone sells for \$99, compiler and tool kit for \$249. For those who have version 2, an upgrade package is available. For \$89 and your V2 system disk, you get the complete development system, compiler and tool kit.

I'll report in more detail in a later column, but for now I should mention that the new compiler generates DOS LINK compatible .OBJ files, which means that the assembly language interface is much easier. The libraries have been upgraded and expanded to include bit operations on bytes, words and blocks, block operations, expanded calendar and time functions, graphics support with windows, long integers (32 bits), large sets, timing functions, random number generation, improved overlay management, and more.

The tool kit includes an intelligent linker (that can exclude the unused portions of a module), symbolic run time debugger, source formatter, a make utility (generates a batch file to recompile all client modules when a DEFINITION module is changed), a version utility, a disassembler (converts .OBJ into assembly), cross referencer, and source checker in addition to the complete sources for all of the libraries. Their Turbo Pascal to Modula-2 source code translator is also included.

If you get and install the entire development package and translator, be prepared to use between 4 and 5 meg of your hard drive.

Turbo Pascal V4.0

I don't have it in hand yet, but Turbo Pascal version 4 has been announced. It generates

LINK compatible .OBJ files and uses the same (very good) user interface as their Prolog, C, and BASIC compilers. The price is reported to be \$99. More later.

Scanner Part III

My goal for this installment is to get to the point of putting captured pixels on the screen. As you will see in the code, there have been some changes made in the definition module presented last time, some out of necessity, others for convenience. The altered portions of the .DEF module are indicated in comments in the source.

All BYTE variables have been changed to CHAR. This is because of the limitations some compilers place on BYTE variables. Overall, it was cleaner to use type CHAR and do type conversions where needed.

The goal of having all adapter-specific code in module Config cannot be met since absolute variables (at a user-defined address) must have their addresses specified in a pure constant, not a constant expression. Unfortunately, a named constant counts as an expression. You will have to put an adapter-specific address in the screen variable declaration.

The Data Capture Routine

For speed efficiency and to provide compatibility with as many Modula-2 compilers as possible, I decided to code the data capture routine in assembler, make it resident, and access it through a software interrupt.

This means I have to load the capture program BEFORE running the high level driver program. So that we can clean up afterward, you should load one of the TSR maintenance programs like the public domain MARK before the resident capture routine.

The assembly code in Figure 1 is written for the shareware assembler A86 (see the Public Domain column, issue #38 of *Micro C*). Once you get used to it, it's a pretty good assembler.

Editor's note: John's assembler code, along with his high level code, is available on the Micro C RBBS. Or give us a call at 1-800-888-8087 and order the Issue #39 disk.

The program first installs its internal function dispatcher as interrupt 60H. It then sets up to intercept the real time clock interrupt (but does not change its frequency) and exits to the operating system. The functions supported through INT 60H are:

(1) Re-program the clock hardware for a faster clock tick. At appropriate intervals the original clock tick routine is called.

(2) Capture a line of scan data, one data point per clock tick. The target address of the captured data, and the number of data points required, are passed in CPU registers.

(3) Restore the original clock vector.

High Level Interface

The implementation module ScrnStuff provides the interface between the resident capture routine and the image processing software (to be written). It also provides printer control routines (which step the scan head across the paper and advance the paper).

I decided to use direct DOS calls to the printer for two reasons. First, the data output is limited and repetitive and does not need the versatility of a full function printer interface. Second, I didn't want to use up the space that the file system module would require. (We

will need it later for image save/restore.)

For each scan line, the software needs to:

(1) Send a line of null graphic data to the printer.

(2) Delay a short while to allow the print head to come up to speed.

(3) Capture pixel data using the TSR.

(4) Send the code for a one-pixel line feed to the printer.

You may need to change the appropriate character constants for your printer. The codes in the module work for EPSON or STAR printers.

ClrScr clears the screen; it should be called before putting the screen into graphics mode with GraphMode. GraphMode uses the interleave factor defined in Config to decide which adapter you're using. Then it either calls a BIOS function (for CGA or EGA) or directly re-programs the video hardware. TextMode, which should be called before the main program exits to DOS, sets up the hardware to again display text.

PixAddress, ClrBit and InvertBit will be used later in image processing. The last two of these use inline machine code for speed—they could have been coded in Modula as operations on bitsets (with the proper data conversions).

Do You Get The Picture?

TestScan is the module for testing this interface and should allow you to get a scanned image on the screen. Don't be too concerned about the quality of this first image, later software (and possibly hardware) mods will fix it up.

Several people have asked me about PC Paintbrush picture file formats. I'm sorry to say that I have no inside information about them, but we will still be able to get the captured images into a form that the program can use. Paintbrush has a companion resident utility called FRIEZE that prints Paintbrush images. One of its other functions is to capture graphic images to disk files. I've successfully captured and manipulated images from Generic CADD and several Turbo Pascal programs using FRIEZE. I see no reason it shouldn't work for this application also.

Next Time

Look for the final installment of the scanner project where we'll get into the basics of image processing, disk save/restore, and printing (unless you've got two printers, obviously a separate operation).

Full Featured AT Motherboard fits XT or AT case! \$399

(6/8MHz, \$489 for 6/10)

Upgrade your XT to a real AT for about the price of an "accelerator" card

OR

build a space-saving AT from scratch.

Features:

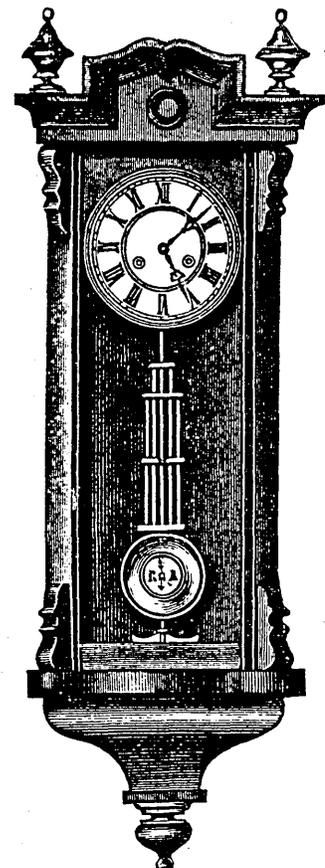
Phoenix BIOS, 1MB memory (0 K installed), VLSI technology, 8 expansion slots, Clock/Calendar.

Add \$95 for 1MB memory installed and tested.

Other selected components available include

- Toshiba 1.2MB floppy drive \$118
- Floppy/Hard drive controller \$194
- XT size case w/LEDS, lock, reset \$ 55
- Everex Magic I/O Par/Ser card \$ 69
- Everex EGA card \$195
- Mono/Graph/Printer card \$ 69
- Famous Datadesk Keyboard \$119
- High density floppies (10) \$ 10

Call SoftSide Systems at (503) 591-0870



1-Hz System Clock



CULTURE CORNER

By David Thompson

Customer Relations

Cementing The Ties

The relationship between the customer and the repair facility has never been close. But lately the problem turned ugly when Horace Goetz shot two repairmen in a crowded computer store. The most shocking part of all was the way the judge let him off with a slap on the hand.

So, in light of the problem, we at Micro C suggest you forge closer ties with your customers. Foster a climate of trust. And, most of all, make them feel guilty. We highly recommend you begin using this letter immediately.

Sample Customer Letter

Dear Ms. Mr. Customer Complainer,

Hi, I'm number , the service woman man _____ who's been personally assigned to take care of your computer. I'm sure over the next 6 to 12 months we'll get to know each other very well.

Enough of the social stuff. I've recently received the twisted remains of something your dealer thinks we manufactured. We also noticed a name etched into the case. And it's not yours. (In lieu of reporting this slight discrepancy to the police, we're giving you our special non-owner repair rate. It's an offer you can't refuse.)

Anyway, the last time I saw something resembling this was Halloween night when my boss set fire to the dumpster. But, in an attempt to appear more than fair, I've poked about it a bit and found:

- Silkworm missile damage.
- Your cat was pretty desperate.
- Parts of salami sandwiches.
- A Micro C 2-to-4 upgrade.
- A loose wire.
- Fragments of anchor rope.
- The welding torch used to install the RAM.

Even though we've never seen anything like this system before, our manufacturing people tell us we have a warehouse full of spare parts,

some of which may work. We will be glad to:

- Replace the motherboard.
- Replace the daughterboard.
- Replace the fatherboard.
- Replace the otherboard.
- Replace the thingy-thingy (a technical term for a dingus).
- Replace the keyboard.
- Replace the keyboard operator.
- Replace your old flame.
- Replace the wires:
 - All
 - Some (Specify color) _____
- Replace the batteries.
- Replace the detonator.
- Replace your cat.
- Replace your Silkworm missile (express).

Now, Ms. Mr. Customer Complainer, it's your turn. Fill in the portion below. Then separate out the green, green/yellow, grey, black, cyan, maroon, and lavender copies. Return all other copies, plus a picture of your first born to: (Company Rubber Stamp)

Dear Repairperson ,

It would please me deeply if you would be so gracious as to fix the aforementioned hopeless case:

- For better.
- For worse.
- For good.
- Don't fix it, I'll pay off the national debt.

I'm including my VISA or Master Card number, [_____]. The credit limit is [_____]. I'm sure you can make more use of this card than I can.

And, again, thank you for letting me impose.

Your loving customer,

[_____] (Must be signed in blood)

MICRO ADS

A Micro Ad is the inexpensive way to reach over 22,000 technical folks like yourself. To place a Micro Ad, just print out your message (make it short and sweet) and mail it to Micro C. We'll typeset your ad (no charge) and run it in the next available issue. You can also send camera-ready copy. Rates: \$99 for 1 time, \$267 for three times, \$474 for 6 times (a best buy at only \$79 per insertion). Full payment must accompany ad. Each ad space is 2 1/4 inches by 1 3/4 inches.

\$200 MULTI-USER VOICE MAIL SYSTEM

FOR IBM COMPATIBLE INCLUDES:

- Multi-User Auto Telephone Answering
- Dictation W/True Editing-Listen While Using Text Editor
- Forwards Calls or Calendar Entries
- Includes On Board 1200 Baud Hayes Compatible Modem
- Remote Controllable From Any Touch Tone Phone
- Multi User-Full Password Security
- Configurable For Telemarketing-D Base Data Retrieval
- Programmers Interface Available

UNIQUE PRODUCTS 301-655-0550
8226 BRATTLE RD BALTIMORE, MARYLAND 21208

Reader Service Number 43

★ UNIQUE TAPE BACK-UP \$300

Use your video tape player and our board as a 100 meg tape backup for hard drive 1 meg/minute — quad redundancy IBM/PC or compatibles-\$300+ \$9 S & A 30 day money back guarantee

MAKE THOUSANDS WITH YOUR HOME IBM COMPATIBLE

Let your home computer work for you all day fully unattended. With our Telemarket System, take polls, market surveys and provide Telemarketing Service for local clients automatically. Includes details to start your own business & all features in our voice mail ad on this page.

UNIQUE PRODUCTS 301-655-0550
8226 BRATTLE RD BALTIMORE, MARYLAND 21208

FREE BUYER'S GUIDE

Call or write for our FREE comprehensive "Buyer's Guide" containing hundreds of languages, utilities and books specially for IBM personal computers and compatibles. We're the world's leading independent dealer of programmer's development tools because we provide sound advice, low discount prices, fast delivery, FREE domestic shipping and no hidden charges.

Programmer's Connection
800-336-1166 USA
7249 Whipple Ave. NW
North Canton, OH 44720
800-225-1166 Canada
9102406879 Telex
216-494-3781 International

Reader Service Number 51

QL Computer \$99.95 pp

128K Ram, expandable to 896K, 68008 Chip, 32 Bit processor, 2 Built-in microdrives. Write for catalog and information.



Sharp's Inc.
Rt. 10 Box 459
Mechanicsville, VA 23111
(804) 746-1664

Advanced C Source Code Analysis With C:LINES/C: TREE

- Outlined listings reflect flow of control *within* procedures.
 - Tree diagrammer shows flow of control *between* procedures.
 - Source code reformatter.
- \$59.95, shipping & handling included. Maryland residents add 5% sales tax. Requires PC Compatible, MS-DOS 256K

SOFTREX • 4807 Bethesda Av., Suite 287
Bethesda, Md 20814 • 301-881-8274

Reader Service Number 56

MSPACE

MAJOR

SPECIALIZING IN
**HARD DISK - TAPE BACKUP
EXPANDED RAM**

• **\$700.00 SPECIAL** •
**KONAN KXP CONTROLLER
MAXTOR 1065 DISK DRIVE
FORMATS TO 58-110 MEG.**

Call 602/230-4213
9029 N. 43rd Avenue • Suite 106
Phoenix, Arizona 85051

Reader Service Number 50

The GAUSS Mathematical and Statistical System

- Publication Quality 2D and 3D Graphics
 - Interactive and Large-Scale Matrix Programming
 - Link Fortran, C, Assembler Subroutines
 - Statistics, Optimization, Database Management, Simulation, Linear Programming, Non-Linear Equation Solution, signal Processing, and More!
- \$350

For the IBM PC-XT-AT-System/2 and Compatibles.

Aptech Systems, Inc.
1914 N. 34th St., Suite 301
Seattle, WA 98103
(206) 547-1733

Reader Service Number 61

GA/3 VIDEO CARD

Use Up to 3 Monitors From One PC
Use All 3 Monitors Up to 1000 Feet From Your PC

- Supports CGA/HGC/MDA in any combination
- Works with standard PC software also
- XT/AT example software included
- Long Distance Option \$39.95 per monitor
- Quantity/OEM discounts

GA/3 Video Card \$699

IHI Advanced Products Division
6280 East Progress Lane, East Wing
Parker, Colorado 80134

303/841-4772 (Colorado) 800/237-2842 (Outside CO)

Reader Service Number 81

HOW COMPLEX IS YOUR SOFTWARE?

FIND OUT WITH PC-METRIC™!

- computes popular complexity metrics
- checks program against standards
- predicts bugs and effort
- versions for many popular languages
- includes 100 page tutorial/manual

SET Laboratories, Inc.
Dept. MC, Box 03627, Portland, OR 97203

Reader Service Number 58

Programmer's Paradise

The microcomputer software source that caters to your programming needs.

*Gives You Superb Selection,
Personal Service, Unbeatable Prices!*

CALL OR WRITE FOR THE LATEST CATALOG

1-800-445-7899
In NY: 914-332-4548

Programmer's
Paradise™



A Division of Hudson Technologies, Inc. 42 River Street, Tarrytown, NY 10591

Reader Service Number 68

IBM Style Monochrome Card
with parallel port. Made in Japan.
\$35.00

XT Style Keyboard without case
mfg. by Cherry \$25.00

The dBase Book of Business Applications, 335 p. \$3.95

Koala Computer, Inc.
4306 Torrance Blvd.
Torrance, CA 90503
(213) 316-5866

Reader Service Number 88

Macro Language/1 (ML/1)

The fast, language independent macro processor

Extend your current programming language or define your own.

- Many useful utility ML/1 definitions included.
- Speed source program conversions or level upgrades.
- Unlimited macro definition passes. Supports 30K macro definitions per pass.
- Define 4GL extension to your production language/environment.
- Define new levels of programming abstraction and reduce software development time, effort, and tedium.
- Define object-oriented language extensions.
- Define complex text search and replacement macros.

Requires: IBM PC, XT or AT and 100% compatibles, PC-DOS (MS-DOS) 3.1 or later, 256K. Now only \$199.95, residents of New Jersey add 6% sales tax. (609) 448-3876 **CACS Software, 34 Brooklawn Drive, East Windsor, New Jersey 08520.**

ADVERTISERS INDEX

Issue 39

RS#	PG#
39	Analogic Company 56
30	Andratech 52
4	Austin Code Works 47
5	Blaise Computing 5
1	Borland International Back Cover
31	CC Software 65
15	Cascade Electronics 57
21	Computer Cowboys 27
6	Computer Helper Ind. Inc. 75
7	CompuView 35
8	Datadesk International 1
32	Digital Research Computers 88
16	Dreamtech 89
33	E2I Computer 88
9	Ecosoft Inc. 37
10	Emerald Microware 53
25	Flexisoft 23
11	Halted Specialties 39
26	Harger I.N.T. 73
34	Hawthorne Technology 46
22	Integrand 21
12	LOGITECH Inc. 17
17	Manx Software Systems 61
35	Merlyn Productions 61
	* Micro Cornucopia 45, 81, 82, 83
	* Micro Methods 45
36	Microcomputer Systems 52
23	Micomint 22
37	Microprocessors Unlimited 46
2	MicroSphere Inside Back Cover
24	microSOLUTIONS 28
42	McTEK Systems 41
3	PC Tech Inside Front Cover
20	Peacock Systems 29, 45
29	Pecan Software 9
18	SemiDisk Systems 87
27	Serengeti Software 73
19	Slicer Computers 67
40	Star-K Software Systems 34
28	Softside Systems 92
41	Sunderland Software Assoc. 29
14	WindowDOS Assoc. 43
38	XenoSoft 61

*Advertiser wishes to be contacted directly.

VIDEO TAPES
Understanding C \$49.95
Power Programming in C \$69.95

- 4 hours of instruction
- diskette with source
- companion manual

Add your sales tax, plus \$3 shipping
Applied Logic
 2309 Royce Dr., Arlington, Texas 76016

Reader Service Number 66

FastCopy 3.0

Turns your PC into a diskette duplicating machine!

for ONLY \$69!

SYSTEMS, SOFTWARE, SUPPORT
 PO Box 751022
 Houston, Tx 77275-1022
 713-941-3100

Reader Service Number 89

For PCs ...
 ... \$1250

(212) 580-0257

Parallon™ parallel processors
 Human Devices, Inc. 322 W 71 St NY NY 10023

Reader Service Number 45

Computer Industry Almanac

Fact-filled guide about the computer industry. Includes people, companies, products, trends, natl users groups, associations, headlines of the year, book clubs, periodicals, forecasts, financial facts, benchmarks, fun trivia. 780 pgs. \$29.95 + \$2 shipping.

Computer Industry Almanac, Inc., 8111 LBJ Frwy, 13th Floor MC, Dallas, TX 75251-1313. Telephone (214)231-8735.

Reader Service Number 46

RAM DISK
S-100
2 Meg, Port I/O
New, Warranted
\$725

S. Lugert
 439 Peck Slip or call:
 NY, NY 10272 718-622-0654

Reader Service Number 52

Azatar DOS Toolkit

Turbo Pascal Tools for DOS 3.XX

NEW! Featuring:

- 80 Pascal procedures and functions
- Professionally bound manual
- Example programs
- No licensing for compiled code

Only \$95!!

Azatar MicroSystems Inc. 3300 Monroe Ave.
 Rochester, NY 14618 (716)385-9780 Hrs: 9-5 Eastern

Reader Service Number 47

EFFECTIVE
LOW-COST,
DISPLAY
ADVERTISING

16 Megabytes EMS and/or Extended Memory

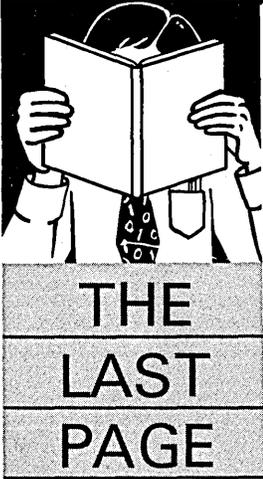
- Works on 8 or 16 bit bus
- 16 bit transfer on AT bus
- Single board design
- Includes RAM disk and extensive diagnostics
- Quantity/OEM discounts

XT and AT Compatible

Designed, Manufactured, Sold and Serviced by **Ptech**

904 North 6th St. Lake City, MN 55041 (612) 345-4555

Reader Service Number 54



The Idea Generator

By Gary Entsminger
1912 Haussler Dr.
Davis, CA 95616

Gary looks at a very different program which helps solve a very big problem. Generating and evaluating ideas. (For the other side of this topic, check out the "Disputer" in "Around The Bend.")

I'm always looking for ideas. So naturally, the idea of an idea generator grabbed me. "The Idea Generator," in fact, is already more than an idea, it's a program distributed by Experience in Software in Berkeley. From the offset, I'll admit I didn't know what to expect from an idea generator, and I hadn't even glanced at the manual. But I did expect it to be smart enough to let me use it anyway.

So I turned it on, hoping to generate an idea.

In Session

The Idea Generator is dialogue and menu driven. A session consists of queries and suggestions from the I.G. based on input from you. The idea generator doesn't generate ideas for you but helps YOU structure yours. If you think the way I do, a little structure comes in handy.

In the beginning, after assuming you have no idea what's going on, the main menu offers some choices—

PROBLEM STATEMENT
IDEA GENERATION
EVALUATION
QUIT

And under PROBLEM STATEMENT, the first of many submenus:

Describe the situation
List your goals
People involved
To sum up

Other dialogues let you choose ideas to evaluate, to rate according to goals, costs, benefits, and their effects on others (ideas and people). Several techniques for idea generation get you started:

Similar situations,
Metaphors for your situation,
Other perspectives,
Focus on goals one by one,
Reverse goals,
Focus on the people involved,
Make the most of your ideas,
Sum up.

The help screen and files are very useful. One function traces your progress and asks if you'd like to see any of the screens you've already visited or completed. Another lets you view the big picture.

Printing and redirection to text, report, and Thinktank files make it easy to return to any condition state in a session.

When you exit the idea generator in unreal time, it asks—Your computer says it's 1/1/80; what's the real date?

My second session began with—

Welcome back to The Idea Generator(tm).

If you thought of any new ideas while you were away, go to the IDEA SCREEN now by pressing the F2 key.

To add to or revise any of your information, you need only go back to the appropriate screen.

YOUR SITUATION: I need to finish The Last Page.

Do you want a quick review of what you've done so far?

Details

I chatted with Larry Moss in Experience in Software's tech support. He said that many of his customers were using the idea generator to brainstorm before writing programs (prototyping) and to solve small business problems. At the very least, I had fun with this program; it's innovative, instructive, and so far has helped me find an idea I thought I'd misplaced.

Order from:

Experience In Software
2039 Shattuck Ave., Suite 401
Berkeley, CA 94704
(415) 644-0694 — Price \$189



IT'S WHAT'S UNDER THE HOOD THAT COUNTS!

XT KIT W/ 2 Floppy Drives.

Includes: 640 K RAM Serial, parallel and game ports, clock/calendar, AT-Style keyboard, cabinet, power supply, mono graphics card and amber or green monitor. Keyboard switchable turbo.

8 mhz with standard
slide cabinet..... 649.00
10mhz with lock, LED, Reset
& Turboswitch..... 699.00

XT KIT W/20MB Hard Drive.

Includes: 640 K RAM Serial, parallel and game ports, clock/calendar, AT-Style keyboard, cabinet, power supply, mono graphics card and amber or green monitor. Keyboard switchable turbo.

8 mhz with standard
slide cabinet..... 949.00*
10mhz with lock, LED, Reset
& Turboswitch..... 995.00*
*(For 30MB Miniscribe add \$50.00)

CASES & POWER SUPPLY

150 Watt Power Supply (XT) 59.00
200 Watt Power Supply (AT) 99.00
XT Slide Case 32.00
XT Flip Top or XT Slide
with Lock & LED 49.00
AT with Lock & LED 65.00

MONITORS

EGA/CGA (Auto Switch) 495.00
VGA/EGA/CGA Color 650.00
CGA Color 339.00
Samsung Amber 12" TTL 89.00
Samsung Green 12" TTL 89.00

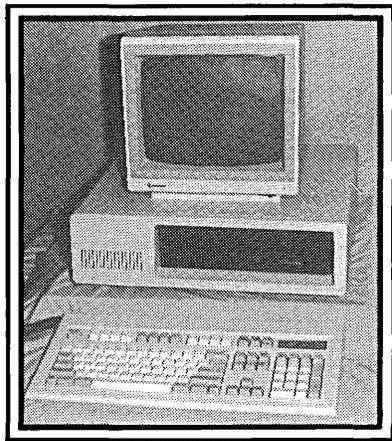
VIDEO CARDS

Color/Graphics 55.00
Color/Graphics/Parallel 65.00
256K EGA Graphics 155.00
Mono/Graphics/Parallel 69.00
ATI Graphics Solution—
Mono, Herc. Color Emulation on
Mono CGA (List 299) 150.00
ATI Wonder Auto Switch Mono, Herc,
CGA, EGA, VGA (List 499) 299.00
Any monitor, Any software, Auto conversion

EXPANSION CARDS

Clock Card 25.00
Dual Floppy Disk Controller 25.00
Joystick 25.00
Game Port 19.00
Multi-Function, 1 ser/par/clk/game/
2 floppy 79.00
Parallel (printer) 19.00
Serial Port (RS232) 1 port 29.00
640K RAM (ØK installed) 35.00
XT/AT RS232
(4 port/2 installed) 59.00

Prices are subject to change without notice.
Shipping CHARGES will be added.



— Pictured keyboard is 5339 —

KIT OPTIONS

*MS DOS 3.21 w/
GW Basic 95.00
*5339 Keyboard Sub 24.00
*Color Options:
(Includes video card & monitor)
CGA Color 200.00
CGA/EGA Color 410.00
CGA/EGA/VGA Color... 590.00
ASSEMBLY AND TESTING
XT Systems 60.00
AT/80386 Systems 80.00

MOTHERBOARDS

XT/Turbo 4.77/10mhz 119.00
AT 6/10 mhz (5 layer) Choice of
Phoenix or DTK Bios 385.00
XT/Turbo 4.77/8 mhz
(2 layer) 109.00
XT/Turbo 4.77/8 mhz
(4 layer) 119.00
80386 8/16 mhz/Award
Bios & 1MB RAM 1595.00
For XT/AT: (120ns chips)
640K memory 85.00
For AT: 1MB memory 135.00

FLOPPY DISK DRIVES

Fujitsu 360K 97.00
Toshiba 360K 99.00
Toshiba 1.2 MB 145.00
3½" Drive Kit 145.00

KEYBOARDS

5339 Professional XT-AT
w/12 function key 79.00
5060 Keyboard AT Style 55.00

FROM THE INSIDE OUT: TECH TIP —

The interleave determines the number of sectors a hard drive skips between disk writes.
For Hi Speed Hard Drives (under 40 MS) on 10 Mhz ATs we use 3. For Standard Speed Hard Drives (over 40 MS) on XTs we use 4 or 5.
To optimize a hard drive, use a utility program such as Mace Utilities for interleave testing and recommendations.

NEW!

80386 KIT —

Includes: 8/16 mhz, 1MB RAM, 1 360K floppy drive, 1 1.2 MB HD, 1 40MB HD, Award bios, switchable keyboard, monochrome monitor, monographics. Serial/parallel ports, case, power supply, game port, clock/calendar. 2875.00

80286 - AT KIT

Includes: 640K RAM, Serial parallel and game ports, clock/calendar, AT-style keyboard, cabinet, power supply, monographics card and amber or green monitor keyboard switchable turbo.
1 360K Floppy drive and 40 MB
Seagate St 251 Hard Drive,
6/10mhz 1895.00

HARD DRIVES & CONTROLLERS

AT 40 MB Seagate #251 495.00
AT Hard Drive &
floppy controller (WD) 185.00
20 MB Miniscribe HD
with controller 349.00
30 MB Miniscribe HD
with controller 399.00
AT 40MB NEC HD
with controller 695.00
20 MB NEC HD
with controller 449.00
30 MB NEC HD
with controller 499.00

SOFTWARE

The Twin Spreadsheet 49.00
Leading Edge Word
Processor 49.00
Ventura Desktop
Publisher by Xerox 525.00
Turbo C by Borland 89.00

ACCESSORIES

1200 Baud Modem — Internal
(Leading Edge Model L)
Hayes compatible 99.00
2400 Baud Modem — Internal
(Leading Edge Model L)
Hayes compatible 219.00
1200 Baud Modem — External
Hayes compatible 119.00
V20-8mhz 14.00
Memory Chips (call for prices)

BUILDING YOUR OWN CLONE

****FREE BOOKLET****
*90-day warranty/30-day money back
(subject to restrictions)

Free Instructions with Each System



MicroSphere, Inc.
P.O. Box 1221
Bend, Oregon 97709
(503) 388-1194

Hours: Monday-Friday
9:00-5:30 Pacific Time



Borland's new Turbo C: The most powerful optimizing compiler ever

Our new Turbo C[®] generates fast, tight, production-quality code at compilation speeds of more than 13,000* lines a minute!

It's the full-featured optimizing compiler everyone has been waiting for.

Switching to Turbo C, or starting with Turbo C, you win both ways

If you're already programming in C, switching to Turbo C will make you feel like you're riding a rocket instead of pedaling a bike.

If you've never programmed in C, starting with Turbo C gives you an instant edge. It's easy to learn, easy to use, and the most efficient C compiler at any price.

“ Turbo C does look like What We've All Been Waiting For: a full-featured compiler that produces excellent code in an unbelievable hurry . . . moves into a class all its own among full-featured C compilers . . . Turbo C is indeed for the serious developer . . . One heck of a buy—at any price.

Michael Abrash,
Programmer's Journal ”

Join more than 100,000 Turbo C enthusiasts. Get your copy of Turbo C today!

Technical Specifications

- ☑ *Compiler: One-pass optimizing compiler generating linkable object modules. Included is Borland's high-performance Turbo Linker.™ The object module is compatible with the PC-DOS linker. Supports tiny, small, compact, medium, large, and huge memory model libraries. Can mix models with near and far pointers. Includes floating point emulator (utilizes 8087/80287 if installed).*
- ☑ *Interactive Editor: The system includes a powerful, interactive full-screen text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code.*
- ☑ *Development Environment: A powerful "Make" is included so that managing Turbo C program development is highly efficient. Also includes pull-down menus and windows.*
- ☑ *Links with relocatable object modules created using Borland's Turbo Prolog™ into a single program.*
- ☑ *Inline assembly code.*
- ☑ *Loop optimizations.*
- ☑ *Register variables.*
- ☑ *ANSI C compatible.*
- ☑ *Start-up routine source code included.*
- ☑ *Both command line and integrated environment versions included.*
- ☑ *License to the source code for Runtime Library available.*

Sieve benchmark

	Turbo C	Microsoft® C
Compile time	2.4	13.51
Compile and link time	4.1	18.13
Execution time	3.95	5.93
Object code size	239	249
Execution size	5748	7136
Price	\$99.95	\$450.00

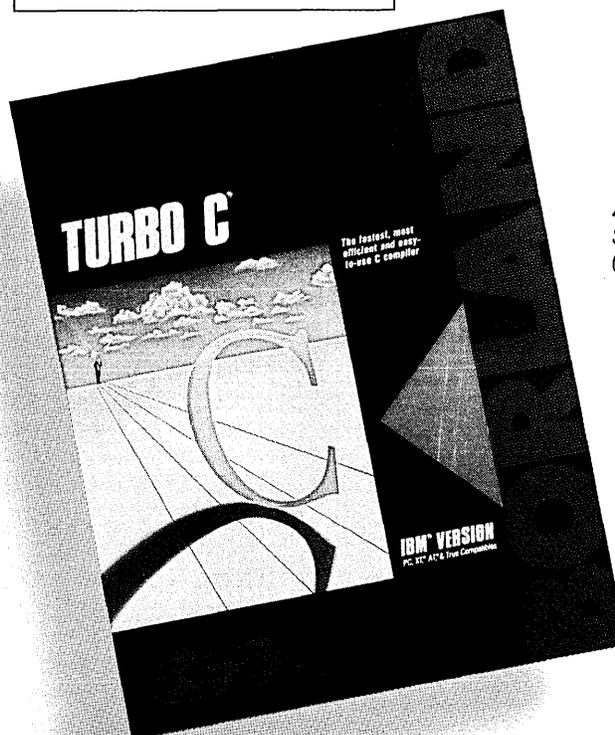
*Benchmark run on an IBM PS/2 Model 60 using Turbo C version 1.0 and the Turbo Linker version 1.0, Microsoft C version 4.0 and the MS overlay linker version 3.51.

Minimum system requirements: IBM PC, XT, AT, PS/2 and true compatibles. PC-DOS (MS-DOS) 2.0 or later. 384K.

For the dealer nearest you or to order by phone call

(800) 255-8008

in CA (800) 742-1133 in Canada (800) 237-1136



4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

Only \$99.95!

All Borland products are trademarks or registered trademarks of Borland International, Inc., or Borland/Analytica, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.
Copyright 1987 Borland International

BI-1136

Reader Service Number 1