REMark®

Volume 6, Issue 12 • December 1985
P/N 885-2071 Issue 71

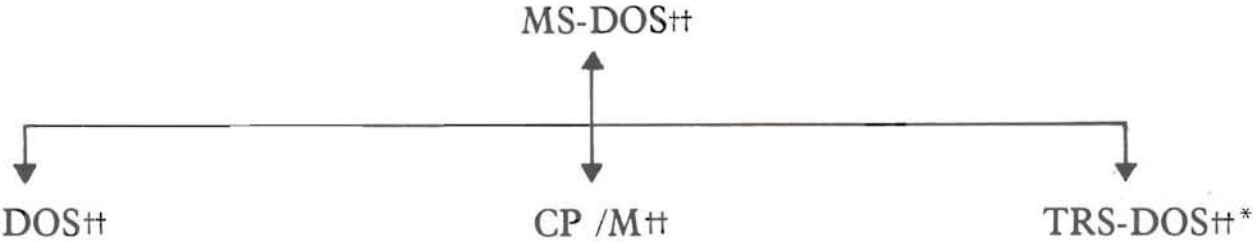
Official magazine for users of HEATH ZENITH computer equipment.

Watch For The New Software Catalog Update

$2.50

micro/VERSAL™

- DISK TO DISK FILE TRANSFER UTILITY -
FOR IBM-PC†† OR ZENITH Z100 ††



If you:

- ***Have or use more than one computer***
- ***Want to Access Public Domain Software***
- ***Have clients with various Systems***
- ***Want to upgrade your present system***

Then you can:

- ***Move files between your Systems***
- ***Transfer from Public Domain format to your own***
- ***Create disks for use on their system***
- ***Move all your source, WP, and data files to your new your new system***

Use your IBM-PC/compatible or Z-100 to READ, WRITE, or FORMAT disks for the following systems:

- 8 inch standard (SSSD)*
- Avatar TC1 (SSDD)
- Avatar TC1 (DSDD)
- Cromemco CDOS – I (SSSD)*
- Cromemco CDOS – II (SSDD)
- Cromemco CDOS – II (DSDD)
- Datavue (SSDD)
- Datavue (DSDD)
- Epson QX-10 (SSDD)
- Epson QX-10 (DSDD)
- HDOS (SSSD)*†
- HDOS (SSDD)
- HDOS (DS00)†
- Heath With Magnolia (SSDD)
- Heath With Magnolia (DSDD)
- Heath With Magnolia (SSSD)*
- Heath With Magnolia CP/M 2.24 (SSDD)
- Heath With Magnolia CP/M 2.24 (DSDD)
- Heath With Magnolia CP/M 2.24 (SSSD)*
- IBM-PC/CPM-86 (SSDD)
- IBM-PC/CPM-86 (DSDD)
- Kaypro II (SSDD)
- Osborne (SSSD)*
- Osborne (SSDD)
- Osborne (DSDD)
- Osborne (Ozmosis SSDD)
- Osborne (Ozmosis DSDD)
- Radio Shack TRS-80 CP/M (SSDD)
- Radio Shack TRS-80 CP/M (DSDD)
- Radio Shack TRS-80 CP/M (SSSD)*
- Radio Shack TRS-80 I DOS (SSDD)*†
- Radio Shack TRS-80 DOS 1.3 (SSDD)*†
- Radio Shack TRS-80 DOS 2 (SSDD)*†
- Radio Shack TRS-80 I DOS (SSSD)*†
- Radio Shack TRS-80 NEWDOS (SSSD)*†
- Lobo MAX-80 (SSDD)
- Lobo MAX-80 (DSDD)
- Morrow MD2 (SSDD)
- Morrow MD3 (DSDD)
- Morrow MD11 (DSDD)
- NEC PC (8001) (SSDD)
- NEC PC (8031) (SSDD)
- NEC PC (8031) (DSDD)
- Sanyo (SSDD)
- Sanyo (DSDD)
- Superbrain 2*
- Televideo (SSDD)
- TI Professional (SSDD)
- Xerox 820-I (SSSD)*
- Xerox 820 (SSDD)
- Xerox 820 (DSDD)
- Zenith Z-100/CPM-85 (SSDD)
- Zenith Z-100/CPM-85 (DSDD)
- Zenith Z-90 (SSDD)
- Zenith Z-90 (DSDD)
- Zenith Z-90 CP/M 2.24 (SSDD)
- Zenith Z-90 CP/M 2.24 (DSDD)
- Zenith Z-89 (SSSD)*

(48 PTI SOFT-SECTORED OR 8 INCH*)

*Not available on PC version
†Formatting option not supported

micro/VERSAL™ is easy to use – menu driven, and requires only 64K and 2 floppy drives or 1 floppy and a hard disk; runs under both DOS and DOS2.

Also: Define your own CP/M formats with our ANYCPM™ program for disk formats not supported by ***micro/VERSAL***™ directly. So why settle for anything less -- **ORDER NOW** and get our AST/Utilities package (a $49.99 value) **FREE**** to celebrate ***micro/VERSAL***™'s 1st anniversary. AST Ulitities includes a disk sector editor, a file recovery program, a file/disk dump program and more!

*** This is our 1st Anniversary Special good until July 1, 1985.*

Current ***micro/VERSAL***™ customers, send $15.00 (sorry no credit cards) to cover shipping and handling and your original disk to receive latest version and utilities.

micro/VERSAL™ runs on: IBM-PC, IBM-PC/XT, IBM-PC/AT, Zenith Z-100, Zenith Z-150, Chamleon, Columbia, COMPAQ, Corona, Eagle, Panasonic, Otorona, NCR, Sanyo MPC (no formatting on Sanyo), Televideo 1605 and others.

$79.99
plus $4.00 shipping & handling

Send check, money order, M/C, or VISA numbers to:

ADVANCED SOFTWARE TECHNOLOGIES

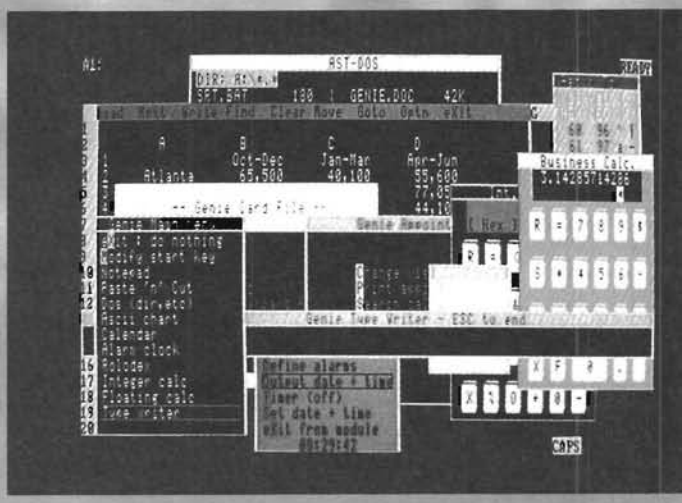
452 W 47TH STREET, SUITE 1B
NEW YORK, NY 10036
212-247-0150

†† MS-DOS, CP/M, H-DOS, and TRS-DOS are trademarks of Microsoft, Digital Research, Zenith, and Tandy Corps, respectively.

NOT PROTECTED

Genie™

Software Magic from Advanced Software Technologies



Shown here is Genie "popped up" on a Z-110 running Lotus 123. From the left are: The Genie main menu, the Genie rolodex style card file, the Genie notepad containing data cut from Lotus, the Genie DOS performing a directory command, the Genie alarm clock (at the bottom,) the Genie typewriter, Genie calendar, Genie Cut and paste, Genie Calculators, and the Genie Ascii table.

Genie is a memory resident application. This means that once you load Genie it is always available for you to use. Just hit the magic keys and Genie will appear (Shift-Shift: No function keys lost.) You can have Genie perform various tasks, and when you finish Genie goes away and you are back where you started.

Here is what you get with Genie:

- **KEY MAPPER** - Redefine any key on the fly, store long commands in a single key, and save many maps on disk.
- **TrueDOS™** - Run any MS-DOS program inside the window. FORMAT, SORT, BACKUP, compile, all without leaving your application.
- **CALCULATORS** - you get two calculators, one regular floating point calculator, and programmer's calculator for base conversions and bit operations.
- **NOTE PAD** - you'll never have to hunt for paper and pen again. Simply call up Genie's Note Pad and jot down what you need. Or expand the buffer to 56K: Instant editor.
- **CUT AND PASTE** - Cut text from any place on the screen and output it later. Cut long commands off the screen and into your KEY MAPPER, move data from your spread sheet to your word processor: Instant integration.
- **CALENDAR** - schedule appointments for any year up to 9999 keep track of expenses, search and print the calendar.
- **ROLODEX** - a name address and telephone number list that you can search any time. No limit on the number of cards. You can even output an address directly to your word processor.
- **ALARM CLOCK** - Have your Genie remind you of appointments, set alarms to ring at any time on any day. A window appears with a reminder of what each of the 8 alarms is for.
- **ASCII Table** - programmers never have to leaf through big books to find the ASCII value of a character.
- **TYPEWRITER** - Knock out quick memos any time, even send ESCAPE codes to your printer. Cut a portion of a spread sheet and paste into the typewriter for quick printouts.
- **SCREEN SAVER** - Automatic phosphor protector for your tube. Genie will even let you blank the screen manually to discourage peepers.
- **COMMAND STACK** - Lets you access up to the last 2K worth of commands you typed.

Best of all with Genie you only have to load what you use. If you don't need to use an ASCII chart don't load it. Need extra alarms? then load two alarm clocks. Your own personal Genie.

And now Genie user's have the advantage of Genie accessories, which can be installed and accessed like any Genie function - any time:

- a **Scientific Calculator** more powerful than most real calculators. Has a full range of math, science and statistical function, and very high accuracy.
- a **Spelling Helper™** To help you find the correct spelling without leaving your word processor.
- a **Print Spuffer™** (spooler/buffer) to speed and manage all your printing.
- a development pack to let you write your own pop up programs, including many routines for windowing, and source code for our Rolodex. Writing pop ups is a snap. (Please call for details)

Only Genie Gives you so much for so little.

ADVANCED SOFTWARE TECHNOLOGIES

452 West 47th Street
New York, NY 10036
(212) 247-0150

Yes please rush me the following software for my
_____ Z110/120 _____ PC compat.

_____	Copies of Genie	@ $49.95	_____
_____	Spelling Helper(s)	29.95	_____
_____	Print Spuffer(s)	24.95	_____
_____	Scientific calc.	19.95	_____
_____	Genie pack (I want it all) Saving $40	$85	_____
		Sub total	_____
Check encl ☐		shipping	4.00
M/C ☐		NY Tax (8.25%)	_____
VISA ☐		TOTAL	_____

Card Number _____

Expires _____ Your Tel Num _____

Name _____

Address _____

Company _____

City/ST/Zip _____

Hurry! Introductory offer ends February 1, 1986.

Heath/Zenith Users' Group

Staff

Manager ..... Bob Ellerton
(616) 982-3867

Software Engineer ..... Pat Swayne
(616) 982-3463

Bulletin Board and
Software Developer ... Jim Buszkiewicz
(616) 982-3463

Software Coordinator ..... Nancy Strunk
(616) 982-3838

Secretary ..... Margaret Bacon
(616) 982-3463

Editorial and Advertising
Assistant ..... Lori Lerch
(616) 982-3794

Printers ..... Imperial Printing
St. Joseph, MI

	U.S. Domestic	APO/FPO & All Others
Initial	$20.00	$35.00*
Renewal	$17.00	$30.00*

* U.S. Funds

Limited back issues are available at $2.50, plus 10% shipping and handling — minimum $1.00 charge. Check HUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and appropriate added cost.

Send Payment to: Heath/Zenith Users' Group
Hilltop Road
St. Joseph, MI 49085
(616) 982-3463

REMark®

Official magazine for users of HEATH ZENITH computer equipment.

Volume 6, Issue 12 • December 1985

Index of Advertisers

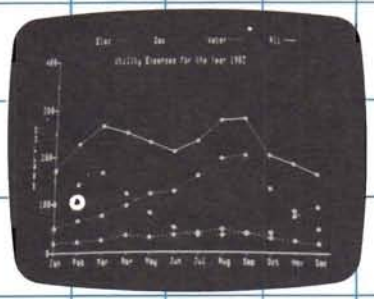
This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.

On The Cover: *Pictured is an HF-148. A simple beginner's kit version of this model is available in the Heathkit Catalog. See review on Page 45 by Bob Ellerton, HUG Manager.*

This is the One

Get More Out of Your H89 or H8/H19 System!

Plug the one upgrade into your system that has all the features of three upgrades, at a one upgrade price. Plug in the Interactive Graphics Controller and gain not just superior graphics, but a high speed RAM Drive, and Print Spooler too!

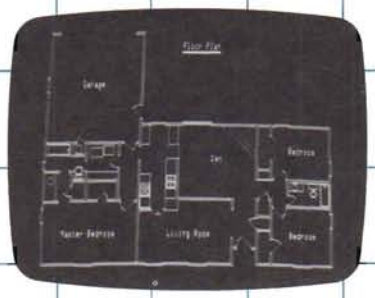


Superior Graphics!

- Highest Pixel Resolution, 640x250 (or 640x500)
- Fastest Drawing Speeds Available
- 3 Graphics Pages for Animation
- Hardcopy Output to most Dot Matrix Printers
- Adds Graphics Commands to all of your Languages

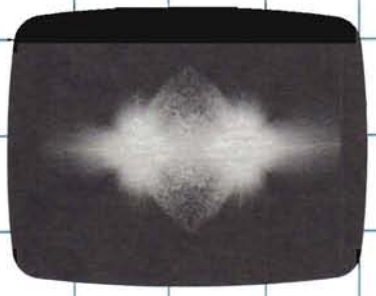
A RAM Drive for your HDOS and CP/M Systems!

- Doubles the Speed of Assemblers, Compilers, etc.
- Files are Retained even During Cold Boots
- Boots HDOS System in 1.5 Seconds



New Print Spooler!

- Never Wait on your Printer Again
- Spools up to 54 Pages of Text
- Supports most Printer and Interface Types



Many Other Features!

- 2 Centronics Parallel Printer Ports
- 2 Trackball and Joystick Ports
- Uses only One Bus Slot (H89 Left Side)
- Quality Manual Includes Illustrations and Examples
- Independent Power Supply
- More than 2 Years of Customer Proven Reliability
- 90 Day Limited Warranty

Conversational Command Set

Dot	x,y
Line	x,y x,y
Box	x,y x,y
Circle	x,y,r
Arc	x,y,r,a,a
Paint	x,y
Style	1,p,c
Pattern	image
Scale	x,y
Erase	x,y x,y
Go	x,y
Display	page
Draw	page
Origin	top/cen/bot
Print	page
Test	x,y
Get	x,y
Put	x,y,image
Input	left/right
Cursor	x,y
Mode	pixel,char
Clear	
Reset	

An Example in BASIC:

```
10 PRINT CHR$(1);"Dot",X,Y
```

Our engineers would love to answer all of your questions about our products. Call and talk to one of them right now!

Now Shipping from Stock!

64k Model	$495.
128k Model	$520.
256k Model	$570.
Manual Only	$ 10.

MasterCard VISA

Please include $5 for UPS shipping and specify H89 or H8, Texas residents please add 6% sales tax.

See us at HUG CON 5!

SigmaSoft AND SYSTEMS

"Support is the most important feature."

4488 Spring Valley, Suite #107, Dallas, TX 75234
(214) 392-1025

BUGGIN' HUG

"Inside MBASIC Program Files"

Dear HUG:

This is in reference to the article in the June 1985 REMark entitled "Inside MBASIC Program Files," by John C. Harper.

First: I wish to compliment him on the article. Not only is it useful as a learning tool, but it also fulfills my need for a Cross Reference Generator for use with MBASIC under CP/M.

Second: It is instructive for me to look for the errors that I introduce in copying programs from REMark.

Third: It is even more instructive to hunt out the typographical errors that exist in the programs as published. In case it may be useful to others, I am listing such errors which I found in some of the programs in the above article.

Program: CALYPSO.BAS

10300: Add +CHR$(30) between CHR$(29) and +CHR$(31). I don't know if I added the right term in the right place, but either this line needs 7 terms or line 10380 has one term too many. The above seems to work, but I would appreciate any better information.

10300: LD-FNUA! should read LD=FNUA!

10540: LF$(HB)-LI$+LE$ should read LF$(HB)=LI$+LE$

Program: PC.BAS

10080: (NE!-INT(NE!/256) should read (NE!-256*INT(NE!/256) or (NE! MOD(256))

10140: CE$=:.1234567890" should read CE$=".1234567890"

10570: FJUA!(LC$) should read FNUA!(LC$)

10610: MID$(BE#,LB+1) should read MID$(BE$,LB+1)

11080: BF$"":RETURN should read BF$="":RETURN

I hope you will continue to publish programs of this nature, and particularly utility programs for the H-89.

Sincerely,

Robert W. Tripp
1055 Peppertree Drive
Sarasota, FL 34242

Reference To REMark June 1985 And September 1985

Dear HUG:

In reference to Mr. Jack Miner's letter on page 65 of Volume 6, Issue 6, June 1985 REMark, and Mr. Paul Woolgar's letter on page 9 of Volume 6, Issue 9, September 1985 REMark. "BIOS Error On B:R/O", this error is explained in the CP/M reference manual on page A-2 of Appendix A. Whenever you are using a system containing more than one drive, the drive that you want to use as a data drive must be the DEFAULT drive. That is, you have a system disk with PIE, for example, in drive A and you want to store data on drive B. You must give the command B: before starting PIE. This makes your data drive a R/W drive in which you are able to change disks without the problem of the above error. The command to start PIE would then be 'A:PIE'. If you are like me in that I can never remember the exact name of a file on the disk, there is a way around it. First, do a DIR and display the disk directory and then by adding a space and the file name in which you are using PIE, for example: 'A:PIE HEATH.LTR'. PIE will then load and start up on the file.

This particular error first showed up when I was writing a checkbook management system using drive A as the system and program drive, and drive B as a data drive only. Since Microsoft BASIC RESET command for CP/M only reads in the disk directory, it was then necessary to change default drives in order to change data disks without constantly losing data every time the data disk was changed.

Sincerely,

John C. Pierrel
604 Stoneybrook Drive
Wylie, TX 75098

Desperately Needs Second Source

Dear HUG:

I need some way to modify Magnolia Microsystems' (MMS) CP/M to allow use of a parallel port for a printer. I have an H-89 with one hard sector drive, two 8" drives running off of an MMS controller, and an H&H parallel/serial board. H&H has failed to fill two orders I've sent for their MMS CP/M modification for their board. I desperately need a second source for some way to print directly from the 8" drives through the parallel board. Any help that you or any member can offer in this manner will be greatly appreciated.

Robert J. Buch
308 Arrowhead Drive
Enterprise, AL 36330

ZD Options List

Dear HUG:

Below is the options list for Version 2.5 of ZD. I have upgraded ZD to support MS-DOS version 2.x pathnames. There are options that will display hidden files and directories, also. ZD still maintains ZDOS compatibility. Obviously, MS-DOS pathnames will not work with ZDOS, but everything else does. ZD will run on a Z-100 or the Z-150.

* * * Options List For ZD * * *

C - Clear the screen before listing the directory
D - Includes directories in the listing
E - Sorts the directory by file extention
F - Sends a form feed to the printer after listing
H - Includes hidden files in the listing
N - Displays number of files listed and space occupied
O - Leaves the directory in original unsorted order
P - Sends the listing to the printer
R - Displays files across rows instead of down columns
T - Allows the user to enter a title
V - Displays the ZD version number

If anyone would like a copy of ZD version 2.5, my initial offer from the April '84 REMark still remains open. Send a buck and a blank disk, or send $6.00 and I'll provide the disk with ZD on it. The source file will still be included. Please specify Z-100 or Z-150.

Thank you,

Jeff Kalis
1920 Sylvan SE
Grand Rapids, MI 49506

Del Scientific Alternative

Dear HUG:

Tom Huber's and John Rogers' excellent article, "Speeding Up The Z-100" in the February 1985 REMark, notes that the CDR Systems, Inc. Z-100 speed-up module may not work on some Z-100s due to a slow 8088, Z-205 expansion memory boards, or other slow support IC problems. Since the CDR unit can operate at only 7.5 MHz, in speed-up mode, the only option in these units is to either operate at normal speed or replace some of the Z-100's ICs. An alternative is the use of the Del Scientific, Inc. TZ-100 speed-up module. This unit is similar to the CDR unit (plug in, no solder, and back panel switch selection of speed mode), except that it uses a pluggable crystal and is supplied with both 7.37 MHz and 6.67 MHz crystals for operation on those systems that do not run at 7.5 MHz. This avoids the replacement of ICs in the computer and still offers a substantial speed increase (daring souls may want to try using a faster 8 MHz crystal). At $39.95, this unit is also $10.00 cheaper than the CDR unit. It is available at most Heathkit dealers or direct from the manufacturer.

The article was misleading on one point. Disk accessing is speeded up. While the data transfer from the disk does not change, the processing of the data is faster and the result is apparently (to the user) faster disk activity for both read and write operations. A simple test of this is to time the boot-up of the computer in normal and high speed modes — a substantial time savings.

Kenneth D. Sechler, Jr., President
Del Scientific, Inc.
4919 Mussetter Road
Ijamsville, MD 21754

Silence No Longer A Virtue

Dear Walt:

I can remain silent no longer! My program for interfacing the Heath GC-1000 clock to an H/Z-100 system, which appeared in the January 1984 issue, needs defending! When I wrote that program, it was my first attempt at writing 8088 code and at communicating with Z-DOS. I was thrilled that it worked as well as it did.

But then along came MS-DOS 2. The letters have poured in. And I have not missed any of the comments that have appeared in REMark. I would hope that fellow HUGgers would take note of the disk that appeared in the April 1985 issue, called "Useful Programs I". On that disk is a vastly improved program that communicates equally well with Z-DOS 1.x and MS-DOS 2.x. Use of this program will alleviate all problems that are being experienced.

I would again like to point out some of its main features:

1. Ability to ignore the GC-1000's internal DIP switches for obtaining the year; this information is extracted from the disk. No more opening the clock just to set the year.
2. Ability to use a port-sharing switch. This program swaps in the correct configuration information at the start, and then restores the port configuration at the end. It is thus possible to share the serial port with a slow modem, and still read the clock at 9600 baud.
3. Ability to ignore the 1/10 second digit. This is important for persons living in weak reception areas.
4. Ability to read the clock from BASIC. Any such attempt with my original program will result in disappointment. This new disk includes a sample Z-BASIC program. I am even aware of an IBM owner that is using the BASIC program.

In summary, to all of those fellow HUGgers that have suggested modifications, I salute you. And I will always welcome future suggestions and comments. But for your maximum enjoyment, order the newest version.

Thank you,

Jim Schuster
2804 Killarney Drive
Cary, IL 60013

Sworn H/Z-151 Devotee

Dear HUG:

I now consider myself a sworn H/Z-151 devotee. This may not be unusual; only four months ago, I considered the H/Z-100 computer the only computer worth a toot.

I successfully assembled Jim Schuster's GC-1000 interface with my '100. I wonder if anyone, who has made a similar transition to the H/Z-151, has a desire to make an interface for the GC-1000 to their new CPU?

Is there a simple conversion of a few lines in the program before assembly, or is the hardware that much different to require a totally new program? Additionally, I have installed a Z-319 card, and I wonder if this program is adaptable for use with the MS-DOS enhanced version?

Sincerely,

Tom Urich, Jr.
1542 Eaton
Berkley, MI 48072

One For The REMark Patch Page

Dear Nancy:

I guess this is one for the REMark Patch Page . . . I got a note last Monday about a bug in one of the example programs in the HDOS Small-C package. The same bug appears in the example for C88. If the correction indicated (the line number is different for the MS-DOS version, but a string search will always find the right line) is performed, then capital N works as specified. Since the examples are shipped in source code form, anybody who

buys the compiler should be able to make the fix himself. Please see that the patch gets published so those who have already purchased the HDOS package will be informed.

Sorry about that!

David A. Wallace
146 Westford Street
Chelmsford, MA 01824

```
else if (c == '=')
{
    puts ("\t\t\t\t\t");
    itoa (lnum);
    putchar ('\r');
    fnc = FUN_EQ,
}
else if ((c == 'N') | (c == 'n'))
{
    puts ("\t\t\t"),
    puts (argv[1]);
    putchar ('\r');
    fnc = FUN_N,
}
else if ((c == 'h') | (c == 'H'))
```

WordStar 3.31 Made Faster

Dear HUG:

Way back in January 1985, Pat Swayne wrote an article on patches for making WordStar faster. (REMark, vol. 6, No. 2, page 32) His patches were for WordStar versions 3.21 and 3.3. Having just purchased WordStar 3.31, I found that neither of these patches work, so I made the modifications needed to insert the patches into WordStar 3.31. The patch is as follows:

```
A:DEBUG
-NWS.COM
-L
-E5480
xxxx:5480 58.C3
-E2BA
xxxx:02BA 00.EB 90.67 C3.C3 00.EB 90.71 C3.C3
xxxx:02C0 00.EB 90.74
-E323
xxxx:0323 00.9A 00.03 00.00 00.40 00.00
xxxx:0328 00.75 00.03 00.B0 00.00 00.C3 00.B0 00.FF 00.C3
xxxx:0330 00.9A 00.06 00.00 00.40 00.00 00.C3 00.9C 00.FC
xxxx:0338 00.9A 00.19 00.00 00.01 00.FE 00.9D 00.C3
-W
-Q
```

This patch will work as stated in Pat Swayne's article and all the comments he makes apply to this version.

Thanks,

Steven Freed
312 Alvarado Hall
University of New Mexico
Albuquerque, NM 87106

Robert A. Speidel Request For Help With H-89A (Vol. 6, #8, Aug. 85)

Dear HUG:

Mr. Speidel reported "flickering" on the screen of his H-89A. My H-89 (actually an upgraded H-88 with CDR controller, 64K CP/M) had an intermittent problem with a band of reduced intensity which slowly crawled up the screen. Each cycle across the screen would take 15 or 20 seconds. When the band was present, disk operation got very shaky, and BAD SECTORS were common.

I "scoped" the +65 volt supply to the video board. There was quite a bit of ripple, but lots of added capacitance did not cure the problem. Then the +5 was checked on the positive lug of the bridge rectifier BR-1. If the screen was normal, the unregulated 8 volts had 0.2 volts of high frequency noise on top of a flat trace. I assume most of the noise was picked up by the unshielded leads. But when the dark band appeared, a 60 Hz sawtooth was seen. The top was at 8 volts with negative going peaks dropping about 1.2 volts to 6.8 volts. This is close to the drop-out value of the voltage regulators.

I replaced the 10 Amp-25 PIV bridge with a 25 Amp-50 PIV and added a cooling fin on the back of the heat sink. My screen has been fine and an evening of dBase II and SpellStar did not produce a single glitch. The new bridge runs cooler than the old one, so I suspect a thermal intermittent was causing a partial short.

Going over the supply voltages with a good oscilloscope is probably the best place to start on any "weird" acting electronic equipment. Hope this helps.

Sincerely,

David D. Barbee, DVM, MS, President
Command Applied Technology, Inc.
West 400 Main Street
P.O. Box 511
Pullman, WA 99163-0511

TSI Presents Prices Below The GSA Schedule

Dear HUG:

Finally, the wait is over for users of the Zenith 100 series computers. Ashton-Tate and MultiMate have just released their best selling "dBase III" database management software and "MultiMate Professional" word processing package in the Z-100 MS-DOS format.

Both companies have selected Technology Services, Inc. (TSI) to distribute these special formats in a new and money saving promotion. TSI is the sole supplier for these products. And they're on GSA Schedule. And now, for a limited time, they're on GSA Schedule below the Schedule prices! TSI is offering these two formats at over $100.00 below our normal GSA Schedule prices. dBase III, which lists for $695, is now offered at $317.00 (normal schedule price of $417!) and MultiMate word processing, which lists for $495, is now on special for $200 (normal schedule price of $306.90).

This special offer ends Dec. 31st, 1985 and is limited solely to the Z-100 MS-DOS versions of these two products. Ashton-Tate and MultiMate International chose us as their exclusive representative after carefully examining our focused service to the Federal government, fast delivery, competitive pricing and top after-sale, technical support.

Call TSI for more information or to request a copy of our GSA Schedules at (703) 354-8668.

Sincerely,

TSI
5703-B General Washington Drive
Alexandria, VA 22312

TOPAZ Disk Security

Dear HUG:

I have written a program set which I call, "TOPAZ:Disk Security". It is fully tested and copyrighted. Now, I wish to make it available to the H/Z-100 community. It is activated upon bootup of any floppy or hard disk (specified partition) drive. Its codes are changeable and it locks up if an intruder fails the three code entry. If anyone would like to purchase TOPAZ, it costs $22.00 for the sample, and $85.95 for the TOPAZ system set (add $10 for hard disk and specify floppy or hard drive) and $114 for TOPAZ: deluxe. I will pay to have it shipped to you. TOPAZ (system set and TOPAZ:deluxe) comes with full instructions. Works in Z-DOS or MS-DOS. (Checks only please).

Truly yours,

Robert Rodgers
3005 Trinity Drive
Bowie, MD 20715

Heath/Zenith Users' Group

Using The H-100 As A Time Machine Part II

Arnold W. Seibel
851 Cypress Street
Monterey, CA 93940

Review

In the first article of this series we developed three assembly-language subroutines. Two of them — TIMERON and TIMEROFF — can be linked at the beginning and end of any other assembly-language program whose execution you want to time to the nearest hundredth of a second. The third, called H2ASCII, is a sub-subroutine called by TIMEROFF to translate the time into ASCII characters so that it can be printed on the screen.

In this article we'll finish developing a program to benchmark the procedure SETPOINT, by Randy Meyers (REMark V. 5, No. 5, May 1984). We'll also look at various ways of using MASM, LIB, and LINK to turn the program into an .EXE file that will run on its own.

Getting It Done: SETPOINT

SETPOINT is presented — and admirably explained — in the article referred to above. For the purposes of this program I have done some minor things to it, which I will describe, in case you want to sandwich it between TIMERON and TIMEROFF — or to link it with any other procedures you might write. (Refer to the listing in the article for the following.)

First, take out the following lines (we'll replace them in a minute):

```
pgroup  group   prog
prog    segment byte    public  'PROG'
        assume  cs:pgroup
```

Second, insert the following lines just before the line PUBLIC SETPOINT:

```
codeseg segment para    public  'code'
        assume  cs:codeseg
```

Those two changes are basically housekeeping, to make SETPOINT compatible with the other subroutines described in this article. I've moved the SEGMENT directive down below the definitions of the constants and the structure, because this arrangement helps me remember that those definitions are not code that gets executed when the program runs — only directives to the assembler.

The remaining changes are necessary because, in my main program (the one that calls SETPOINT as a subroutine), I'm going to execute a LOOP that calls SETPOINT 10,000 times. When you LOOP in this language, the loop count is implicitly kept in the CX register. Since SETPOINT uses the CX register for its own purposes, I have to modify SETPOINT to PUSH the CX register onto the stack at the beginning of the procedure and POP the value back into CX at the end, so that the count will be accurate when the main program checks to see if it's finished looping. In addition to the PUSH and the POP, we also need a modification in the STRUC that's defined just before PUBLIC SETPOINT in the original SETPOINT listing.

First, insert another DW ? as the second definition within the STRUC, so that the beginning of it looks like this:

```
spstr   struc
        dw      ?       ; BP register
        dw      ?       ; CX register <--This is the line you
                                        insert
        dw      ?       ; Return address
```

That structure is a nifty way of getting to values that are on the stack (in SETPOINT's case, the three values passed by the calling routine) without having to disturb the stack itself. This structure is not used (like the ones in TIMEROFF) to locate bytes in the data segment. Instead, it is used to locate bytes on the stack.

Note that the first lines of code after SETPOINT PROC NEAR in the original SETPOINT are instructions that PUSH the contents of the BP register onto the stack (to save whatever valuable information the calling program might have put in there), and then

MOVe the contents of SP — the Stack Pointer register — into BP. BP now points to what is, at this point in time, the top of the stack. It can be used later to indicate the beginning point of a region in memory that will be measured with the structure SPSTR. Look in the code for the expressions [BP].YCOORD, [BP].XCOORD, and [BP].VALUE. These expressions tell the 8088 to look for values that are offset from the address contained in the BP register (i.e., from the original top of the stack) by the same amount that the labels YCOORD, XCOORD, and VALUE are offset from the beginning of the structure in which they were defined.

The final changes are the PUSH and POP. First, immediately after SETPOINT PROC NEAR and before PUSH BP, insert the line PUSH CX.

Second, immediately after POP BP and before RET at the end of the procedure, insert the line POP CX.

That's it. SETPOINT is now fixed up to operate from inside a loop without messing up the loop counter, and the labels for its segment and class are the same as those in the other procedures in this article. If you assemble it in its modified form, you will later be able to link it together with TIMERON and TIMEROFF, or with other procedures.

And Now A Little Peanut Butter: PIXTIMER

So far we have TIMERON, SETPOINT, and TIMEROFF, together with H2ASCII, which is used by TIMEROFF to translate the time bytes. Now all we need is a program that can be entered from the operating system; that sets up a stack on which to pass values to SETPOINT (and, importantly, on which to save return addresses when subroutines are CALLed); that calls TIMERON, calls SETPOINT 10,000 times, and calls TIMEROFF; and that finally exits back to the operating system.

PIXTIMER is the main program. It has a stack segment, for the reasons given above. The directive inside STACKSEG is DB 0FFh DUP (?), which sets aside 255 bytes but does not initialize them. These bytes will be used for the stack. We use 255 rather than some smaller number because LINK, for reasons of its own, likes its stack segments on the heavy side.

When we enter this program from the operating system, the segment number of STACKSEG will automatically be placed in the SS register, and the size of STACKSEG — which is equivalent to the offset of the first byte past the end of the segment — will automatically be placed in SP, the Stack Pointer register. The first time a word (two-byte number) is PUSHED onto the stack, SP will be decremented by two and the word will be moved into the last (highest-address) two bytes in this area. If another word is PUSHed, SP will be decremented again, and the new word will go into the next two bytes down (toward lower memory addresses). When a word is POPped, the word SP points to what will be copied into the location specified by the POP instruction and SP will be incremented to point to the next higher word in its area. The "top of the stack" is the byte that SP points to at any given time. When SP points to the first byte past the end of the stack segment, the stack is empty.

In DATASEG we have the directive RTADDR DD 0. This is a Define Double-word directive. It sets aside a double word (four bytes, eight hex digits, 32 bits), initializes the whole area to zeros, and gives it the label RTADDR. This area will be used to store the address that the 8088 will jump to at the end of this program.

In the directive just before that one, the word LABEL alerts the assembler that an alternate name is being declared for the next address in the segment (the beginning of RTADDR). WRTADDR is the alternate name. The word WORD specifies that, when WRTADDR is used to address this location, the information passed will be in word-size chunks (two bytes, four hex digits, 16 bits) instead of double words. Later we will use this label to move a two-byte value into this area.

Just inside CODESEG we have three EXTRN directives, identifying the three external procedures that will be called from PIXTIMER. (These could all be placed on one line — EXTRN TIMERON:NEAR,SETPOINT:NEAR,TIMEROFF:NEAR — but the code is easier to read, and also easier to change, if you separate them.)

Next we specify the segment registers for CODESEG, DATASEG, and STACKSEG. We also specify ASSUME...ES:NOTHING, which ensures that the ES (Extra Segment) register will not be used to point to any segments. The next line is PIXELTIMER PROC FAR, which alerts the assembler that calls to this procedure will be from some other segment (they will be from the operating system). Next we move the segment number of DATASEG into DS.

The next instruction takes the contents of ES and moves that two-byte number into the second word in the double-word area labeled RTADDR. The expression WRTADDR+2 means that we are after the second word (adding two bytes to the beginning address).

What we just moved into that word from ES was the segment-number portion of the address that the 8088 is to jump back to when it is finished with this procedure. The first word of RTADDR already contains 0000 hex, because we initialized that way. That is the offset, within that segment, to which the 8088 will return. How did the segment number get into ES? It didn't, yet. It will be put there automatically when the operating system calls this procedure.

Now that that's over with, we can get on with the program. The first instruction, CALL TIMERON, executes the subroutine that sets the time to 00:00:00.00.

The next six instructions place three values on the stack for SETPOINT to use as its arguments: the color value 3, for magenta; the y-coordinate 112, halfway down the screen; and the x-coordinate 320, halfway across. In each pair of instructions, the immediate value is first MOVed into AX, and then the value in AX is PUSHed onto the stack. We have to use AX because we can't PUSH an immediate value; only a word-sized (two-byte) register or a word-sized memory location.

Finally we come to the loop that executes SETPOINT 10,000 times. The instruction MOV CX,10000 puts the number of loops required into the CX register, which is implicitly used for this purpose. The next instruction is the CALL to SETPOINT, which lights up the indicated pixel in the indicated color one time each time it is called. The instruction LOOP GO means: decrement CX by one and check to see if it's zero yet; if not, jump to the label GO; if so, just proceed with the next instruction.

Once out of the loop, we come to the instruction CALL TIMEROFF, which retrieves the elapsed time and prints it on the screen, using H2ASCII to translate the time bytes into ASCII characters. Finally we POP three words off the stack, just for the sake of neatness. It's a good habit to make every procedure leave the stack in the same condition it found it — in this case, empty. PIXTIMER has PUSHED three values, so it's responsible to POP them.

JMP RTADDR sends the 8088 back to wherever it was when we told it to execute PIXTIMER. Then we have the ENDP for END of Procedure, the ENDS for END of Segment, and the END to terminate assembly. Following the END is a label indicating the beginning of the code to be executed, which the DOS will interpret as its entry point into this, the main program. Only the main program — not the subroutines — may indicate the entry point after the END directive.

Assemble, Link, And Run

To use these procedures, you need to type the code into some .ASM files — one each for H2ASCII, TIMERON, TIMEROFF, the modified SETPOINT, and PIXTIMER. You can call the .ASM files anything you like; the modules don't have to have the same names as the procedures inside. However, since each of these modules has only one procedure in it, it makes sense to use the procedure names as file names.

Next, you need to use MASM to convert each .ASM file into an .OBJ file. MASM will, if you so direct, produce a .LST file for each .ASM file it processes. I recommend that you direct it to do so. A .LST file shows all the text of the .ASM file that produced it, together with the offsets of all the code in the .OBJ file, and the actual bytes placed in memory by instructions and by the directives in the data and stack segments.

MASM also presents the option of a .CRF file, which you must later convert into a .REF file with the program CREF. The .REF file lists the symbols in each program, together with the line numbers in the .LST file where each symbol occurs.

In assembling, you do have one option that I didn't mention. You could put all these procedures into one .ASM file and assemble them at once. If you want to do that, you should delete the PUBLIC and EXTRN directives; since you will have all the procedures in one module, these will not be needed. You should also put all the procedures into a single CODESEG, and similarly pool all the DATASEGs into one. There are a couple of drawbacks to this option. For one, if the assembly doesn't work, you'll have a much harder time finding the typo (or the modification that didn't work) than if you had assembled each procedure separately. More importantly, you won't have H2ASCII, TIMERON, and TIMEROFF assembled as separate .OBJ files, which could be linked with other procedures to do other timing tasks. (Incidentally, you will still have to LINK the .OBJ file that the assembler produces; LINK will produce an .EXE file, and you can run that.)

If you have assembled the modules separately, you have two different ways to LINK them into a single .EXE file. One way (not recommended) is to specify all of them to LINK as object modules. The most foolproof way to do that is to type LINK, hit (RETURN), and then respond to LINK's prompts as follows:

```
Object Modules [.OBJ]
     b:pixtimer+b:timeron+b:setpoint+b:timeroff+ (RETURN)
Object Modules [.OBJ]: b:h2ascii (RETURN)
Run File [A:PIXTIMER.EXE] b:pixtimer (RETURN)
List File [NUL.MAP]. b:pixtimer/m (RETURN)
Libraries [.LIB]: (RETURN)
```

The drive designation B: is in there because I habitually keep my working files on drive B: and the tools — MASM, CREF, LIB, LINK, DEBUG, an editor, etc. — on drive A:. When you run out of room in responding to LINK's first prompt, you can end the line with a '+' and RETURN, and it will give you another line to work on. The /m switch on the List File line causes LINK to include the names and offset addresses of all the PUBLIC procedures you have linked into your program, along with the absolute memory addresses and lengths of your segments.

The other option is to put all of the .OBJ files except PIXTIMER into a library file. To do that, type LIB and hit (RETURN), then respond as follows:

```
Library File  b:timing (RETURN)
Library does not exist  Create?yes (RETURN)
Operations:
     +b:timeron+b:setpoint+b:timeroff+b:h2ascii (RETURN)
List file: b:timing.cat (RETURN)
```

You can give the library file any name you like; it will automatically get the extension .LIB when it is created. Note that if you run out of room here (on the Operations line) you will have to end the line with a '&' rather than a '+' as in LINK. This is because LIB treats '+' not merely as a separator, as LINK does, but as a command to add the next-named .OBJ file to the library. LIB also has commands to delete a module from the library, and to "extract" a module, which means to copy it into an .OBJ file of the same name without deleting it.

You can name a List File or not, and you can use whatever extension you like; I use .CAT, for catalog, to avoid confusion with MASM's .LST and LINK's .MAP files. The List File for a LIB file lists first the names of the PUBLIC procedures in the library file, each followed by the name of the .OBJ file (module) it's in; then the names of the modules, each followed by the names of its PUBLIC procedures. The .LIB file itself contains an indexed dictionary of the PUBLIC procedures you have put into the file. When you tell LINK to search this library file for the procedures it needs, it uses this dictionary to locate them rapidly.

Having put all the subroutines into a library file, you can answer LINK's prompts like this:

```
Object Modules [.OBJ]: b:pixtimer (RETURN)
Run File [A:PIXTIMER.EXE]. b:pixtimer (RETURN)
List File [NUL.MAP]: b:pixtimer/m (RETURN)
Libraries [.LIB]: b:timing (RETURN)
```

I prefer the latter course — using a LIB file — because the subroutines are all in that file, and I can LINK any or all of them with a new program (a new version of PIXTIMER, for example — there have been a lot of those) without having to type their individual names when giving instructions to LINK. In fact, if I had a lot of library files, I wouldn't even have to know which one had the right subroutines in it; I could just list all the LIB files, and LINK would search all their dictionaries to find the right procedures.

So What?

Either way you LINK them, these procedures will result in an .EXE file that runs SETPOINT 10,000 times and tells you how long it took. I get 1.76 or 1.77 seconds, more or less at random. That equates to about 0.0001765 seconds per dot; to call SETPOINT once for each of the screen's 640 × 225 = 144,000 dots would require about 25.4 seconds — and that doesn't include the time it would take to compute and PUSH a new color value and coordinate pair for each dot.

Does that seem slow? Well, consider this. The fastest version of SETPOINT I've written in LMI 8086 Forth runs 10,000 times in 27.57 or 27.58 seconds — more than fifteen times as long as the original, assembly-language SETPOINT takes to do the same thing. At that rate, my Forth version takes a minimum of more than six and one-half minutes to light up every dot on the screen.

In fairness to Forth, I should point out that it has its own assembler, which can be used to code time-sensitive subroutines like SETPOINT directly in assembly language. In my Forth version of SETPOINT, I didn't use that power; I didn't know enough about 8086 assembly language. Now that I've taken this excursion, I'm ready to give it a try...just as soon as I finish these articles. In further fairness, I should also observe that Forth provides a command that reads the time, including the hundredths of a second; to time my translation of SETPOINT, I didn't have to write Forth versions of TIMERON and TIMEROFF.

Caution: The Z Has Forgotten The Time

Using PIXTIMER has one disadvantage: it wipes out the "real" time, replacing it with the time that elapsed while SETPOINT was repeating itself. Why not extend TIMEROFF to put the real time back? We could save the real time at the beginning of TIMERON and then restore it at the end of TIMEROFF, but that time would be off by the amount of time that had elapsed in the interim.

Alternatively, we could add the elapsed time to the real time (assuming we had saved it at the beginning) and put that back. But that is easier said than done. Consider, for example, what sort of code you would have to write to update the real time if TIMERON saved the time 23:59:59.50 (half a second before midnight) and the procedure being timed took three quarters of a second. Hint: When the second-byte hits 61, the 60 is not automatically going to be carried into the minute-byte as a 1. And when you get that figured out, you should also consider the amount of time it has taken to perform all the computations before sending the corrected time back to the system.

Coming Up

In the third and final article of this series we'll develop another self-sufficient .EXE program, which will be used to print the current time and date on the screen. We'll use H2ASCII and two other sub-subroutines: DROPZERO, which drops leading zeros from two-digit ASCII strings; and AM_OR_PM, which translates the time from the 24-hour to the 12-hour format, with an "am" or "pm" suffix. These procedures will be called by two subroutines, CLOCK and CALENDAR, which in turn will be called by a simple main program called NOW. NOW will give you both the time and the date, in an easily readable format, on one line — without requiring you to hit (RETURN) to get out of it.

Program Listing

```
page    54,132
title   PIXTIMER.ASM:  time 10,000 iterations of setpoint

; This procedure runs setpoint 10,000 times without changing
  the color or
; location of the dot, and prints the elapsed time to the
  nearest 100th second
; Calls the external procedures timeron, setpoint, and timeroff
; Uses registers ax, cx

; Stack segment

stackseg        segment para    stack   'stack'

                db      0ffh dup (?)  ; reserve 255 bytes

stackseg        ends                  ; end of stack segment

; Data segment

dataseg         segment para    public  'data'

wrtaddr         label   word                    ; allow word-size MOV to rtaddr
rtaddr          dd      0                       ; double word for return address

dataseg         ends                    ; end of data segment

; Code segment

codeseg         segment para    public  'code'

                extrn   timeron:near            ; declare external proc's
                extrn   setpoint:near
                extrn   timeroff:near

                assume  cs:codeseg,ds:dataseg,ss:stackseg,es:nothing

; Procedure

pixeltimer      proc    far             ; OS entry point, delcared far

                mov     ax,dataseg      ; initialize dataseg's register
                mov     ds,ax

                mov     wrtaddr+2,es    ; get return address from es

                call    timeron         ; set time to 00:00:00.00

                mov     ax,3            ; push color value
                push    ax

                mov     ax,112          ; push y-coordinate
                push    ax

                mov     ax,320          ; push x-coordinate
                push    ax

go:             mov     cx,10000        ; put loop counter into cx
                call    setpoint        ; light up the dot
                loop    go              ;   repeatedly

                call    timeroff        ; print the elapsed time

                pop     ax              ; get rid of the parameters
                pop     ax
                pop     ax

                jmp     rtaddr          ; terminate the program

pixeltimer      endp                    ; end of procedure

codeseg         ends                    ; end of code segment

                end     pixeltimer      ; end assembly, specify
                                        ; pixeltimer as entry point
```

System Log For The H/Z-100

Robert S. Hudson
8503-A Villa La Jolla Drive
La Jolla, CA 92037

Introduction

Here is an effective approach to maintaining a log for your personal computer. This system was originally devised as a result of the IRS ruling requiring a "contemporaneous" record of personal computer use, and the percentage of business vs personal use. That requirement has since been repealed, but it's a good idea to keep a computer log, anyway, for preventive maintenance purposes.

The total SYSLOG software system consists of 2 ZBASIC file management programs, 3 Z-DOS(MS-DOS) batch processing files, and a PeachCalc spreadsheet. The batch files integrate the BASIC programs and execute them automatically. The BASIC programs maintain a command file which is periodically executed by the spreadsheet for a permanent record printout.

One system disk is reserved for keeping all the log software, which includes the necessary Peachtext system software for the spreadsheet. This disk is always used to boot up the system, and again to log out. Prompts for DATE, TIME, and a task category are displayed when logging in; then when logging out, the prompt "Is this the last entry for this file? [Y]es or [N]o" is displayed. Periodically, say once a month, the log spreadsheet is filled in with the data that has been saved in file for the month. To do this, the spreadsheet, which I have named COMPLOG, is loaded and the file SYSLOG is eXecuted. This automatically fills in the spreadsheet with the log in and log out times and dates that are in the command file. After the data has all been entered in the spreadsheet, it recalculates, and all the elapsed times are filled in. This operation takes less than 3 minutes to fill in and recalculate the complete spreadsheet. The spreadsheet can be saved on disk, or printed out, or both. It makes a handy record to keep a history of your computer usage.

The Spreadsheet

When the log was originally devised, the spreadsheet was the whole program. It has been through several iterations to arrive at the present configuration. Each of these iterations was to simplify the log in and log out procedure. Initially, logging in required manually entering in the spreadsheet the date, time, and description of the task to be performed. There were two sets of columns for logging in and out, one set for personal the other for business. This was too much trouble, so I selected five categories that pretty well covered most of the tasks that would be done, and assigned a number to each. In one column, the category number was entered, as determined by the planned task. Then there was a Time On and a Time Off column for manually logging on and off. There were five elapsed time columns, one for each category. The formulas in these columns tested the category column to determine where the elapsed time belonged. This simplified the log in and log out procedure considerably. This is basically the final version; only a few cosmetic changes to adapt it to the command file entry have been made. Though the spreadsheet was done with PeachCalc, the same approach could be used with other versions such as Lotus or Supercalc.

It utilizes blocks A1 through P63. The entries are all made in the first four active columns. Column A was left blank to give a border on the left. Column B has the Date, C is for the Code, D is Time On, and E is Time Off. These columns and five blanks at the bottom to bring previous totals forward, are the only manual or command file entries. The rest of the spreadsheet is automatically calculated.

The elapsed time in column F is calculated by taking the difference between column D and E. The time is entered in these two columns in decimal hours. In the original version of the spreadsheet, these entries were made in hours and minutes, using a decimal point for separation, and the formula in column F calculated the elapsed time in decimal hours, 24 hour clock. When the system was converted to the Batch processing technique, the decimal hour calculation was done in the BASIC programs, so it simplified the spreadsheet in that respect. It also alleviated another problem: if I worked past midnight (which does happen occasionally), I had to add 24 to the "time off" entry for the elapsed time calculation to come out right. The present version has a conditional formula in column F, to test for that condition. If column E is less than column D, then 24 is added to the difference. The number would come out negative, otherwise; and that wouldn't make much sense for elapsed time, would it?

I have included sample formulas for the various columns. The five columns with the Code designations have conditional formulas to test the value in column C; the elapsed time is then entered in the appropriate column. The Total Hours column (Column N) is the sum of columns G through L. Column K was left blank to provide a space between business and personal classifications. Column O is the sum of Column G through J divided by N and multiplied by 100 to find percentage of total hours spent on business. I included a conditional statement in

here to print a "N/A" if N=0. This is not necessary, but it prints an "error" otherwise, and I prefer "not available"; I guess it's just a matter of personal preference. It looks like a lot of formulas to enter, but it really doesn't take long if you make use of the "repeat" function.

The way the spreadsheet is laid out, it will just fit on an 8 1/2 by 11 if you use compressed text (17.16 cpi). The column widths are as follows:

Column	Width
A	4
B	11
C	4
D-J	9
K	3
L	9
M	4
N-O	9
P	20

With PeachCalc, you can have the spreadsheet recalculate after every entry, or set it for manual recalculation so that it does so only on command. In this case, it is best to set it for manual, rather than automatic. It takes 15–20 seconds for recalculation due to the number of calculations required, so it would slow the execution time of the command file considerably.

The Command File

Two simple ZBASIC programs generate and maintain the command file. SYSLOG.BAS is executed for logging in, and LOGOUT.BAS for logging out. Together, they maintain a sequential file which contains the log in and out information, plus the commands needed to enter this data in the proper blocks on the spreadsheet.

SYSLOG.BAS opens the command file and names it LOGCOM.XQT (the .XQT extension is needed for PeachCalc command files). At each system boot, it inputs LOGCOM.XQT and opens an output file to make a copy so the new data can be added to it. In the event LOGCOM.XQT doesn't exist, such as at the beginning of the month, it opens a new file, and prompts with "STARTING NEW FILE". SYSLOG.BAS contains error traps and prompts, so it is difficult to make a mistake, such as trying to log out before being logged in.

LOGOUT.BAS is executed when "logout" is entered at the system level. It completes the entries in the file started by SYSLOG.BAS. The LOGOUT.BAS program keeps track of the number of entries; and when there are enough to fill the spreadsheet, it displays a prompt indicating this is the last entry for this file. It renames the file SYSLOG.XQT and kills the old filename LOGCOM.XQT so it can be opened as a new file the next session. SYSLOG.XQT can now be used to print out a spreadsheet at a convenient time, or it can be renamed and saved as it is. It should be renamed for saving, otherwise it would be written over the next time LOGCOM.XQT is full.

The BASIC Programs

SYSLOG.BAS creates and maintains a sequential file LOGCOM.XQT which is the command file used for filling the log spreadsheet. Line 30 defines the variable T as TIME/3600. Time is returned in seconds on a 24 hour clock, so T = decimal hours which is most convenient for calculations. Lines 40–110 prompt for the proper code input. Of course, this can be modified to suit individual requirements. Lines 110–310 inputs the existing file and makes a copy for inputting the new data. If the input file doesn't exist (such as at the beginning of the month) then a new file is created in 330–420.

SYSLOG.BAS

```
10 REM SYSLOG PROGRAM CREATES LOGCOM.XQT FILE
   FOR SYSTEM LOG SPREADSHEET.
20 ON ERROR GOTO 440 'Error Trap
30 T=TIME/3600
   'Converts time in seconds to decimal hours (24 hr  clock)
40 PRINT "Enter proper choice:"
50 PRINT
60 PRINT "  [1]  Circuit Analysis and Design"
70 PRINT "  [2]  Reports and Memos"
80 PRINT "  [3]  Schedules and Estimating"
90 PRINT "  [4]  Software/Housekeeping/Educational"
100 PRINT "  [5]  Personal (Budgets, letters, etc.)"
110 INPUT "CHOICE ";X   'X = Task category; remember this one
120 IF X<1 OR X>5 GOTO 40 ELSE 130  '1 thru 5 are the only choices!
130 OPEN "I",#1,"LOGCOM.XQT" 'Open input file; if not there, goto 440
140 OPEN "O",#2,"COPY"  'Open output file to make copy of LOGCOM
150 IF EOF(1) THEN 190   'Test for end of input file
160 LINE INPUT#1,A$      'Read 1 line of input file
170 PRINT#2,A$           'Copy to output file
180 GOTO 150             'Continue til finished
190 CLOSE#1              'Close input file
195 REM Test last line input (A$), left 2 characters
196 REM If they are '=E' then not logged out
197 REM If '=B' everything is OK to continue.
200 L$=LEFT$(A$,2):IF L$="=E"THEN 460:IF L$<>"=B"THEN 210 ELSE 430
210 R$=RIGHT$(A$,2)  'Right 2 characters of A$
220 R=VAL(R$)        'R= value of the number- remember this one
230 KILL"LOGCOM.XQT"  'Kill input file- it will be replaced later
240 PRINT#2,CHR$(34)DATE$
    'This will print the date string with a quote
250 PRINT#2,"=C";USING"##";R  'Prints =C R (remember the R?)
260 PRINT#2,X          'Remember the X?
270 PRINT#2,"=D";USING"##";R  ' More of the same, except with D
280 PRINT#2,USING"##.##";T    'Decimal hours
290 PRINT#2,"=E";USING"##";R  'Getting ready for logout
300 CLOSE  'Close the file
310 NAME "COPY" AS "LOGCOM.XQT"
    'This is the new one mentioned in 230
320 SYSTEM
330 PRINT "OPENING NEW FILE" 'We arrived here from 440
340 OPEN"O",#2,"COPY"        'Starting a brand new file
350 PRINT#2,"=B";USING"##";8 '8 is the first row on the spreadsheet
360 PRINT#2,CHR$(34)DATE$    'There's that quoted date string again
370 PRINT#2,"=C";USING"##";8    ' R =8
380 PRINT#2,X                   'Same X as before
390 PRINT#2,"=D";USING"##";8    'prints =D 8
400 PRINT#2,USING"##.##";T      'Its that time again
410 PRINT#2,"=E";USING"##";8  'Getting ready for logout
420 GOTO 300                  'Closing the file
430 PRINT "FILE READING ERROR":SYSTEM  'We arrived here from 200
440 IF ERR=53 AND ERL=130 THEN GOTO 330  'File not found?
450 ON ERROR GOTO 0 'Turn off the error trap
460 PRINT "NOT LOGGED OUT" 'We arrived here from 200
470 SYSTEM 'Back to the system; and log out this time!
480 END
```

LOGOUT.BAS

```
5 REM Logout program for maintaining LOGCOM.XQT file
```

The file name COPY is used when outputting. When outputs are complete and the file is closed, the file name is changed back to LOGCOM.XQT. The .XQT extension is used because that is the default extension for a command file in PeachCalc.

This file can be used at any time to get a copy of the spreadsheet, because it has the .XQT extension. It will fill the spreadsheet in up to and including the current sign on. Of course, the sign off will be blank, assuming this is done during a session for which you have signed on.

The command file is set up to create a series of commands just as though you were entering them on the spreadsheet command line. When the new file is being created starting in line 330, the first command that will be printed will be "=B 8" (without the quotes). This command entered on the spreadsheet command line moves the active block to B8. The next line in the command file will be "DATE$, so the current date will be printed in B8 with leading quotes; CHR$(34) in line 340 is the ASCII decimal code for ("). It had to be entered this way so ZBASIC would recognize it as printed quotes. If the date is Apr. 14, 1985, BASIC will print it exactly as "04-14-1985.

LOGOUT.BAS maintains the same command file as created and initiated by SYSLOG.BAS. The only difference, of course, is that it is used when logging out. It tests the last entry in the file for errors, such as not being logged in, and for the current active row number. In many respects it is similar to SYSLOG.BAS. In both programs, the numerals in the block designations have to be printed as numeric values, as opposed to strings, because they are used to test for errors and end of file. The PRINT USING"##" statements reserve 2 digit spaces for the numbers. This is necessary to test for the maximum number of entries for the spreadsheet. The way the spreadsheet is laid out, the last row number on which entries can be made is 59. Therefore, in the LOGOUT.BAS program, this number is tested to determine if row 59 has been reached. If not, the number is incremented by one for the next login entry.

Batch File Execution

The best part of the system is the BATCH processing capability of the Z-100. The spreadsheet is really the heart of the system, but the BATCH processing is what makes it an integrated system. It ties the whole thing together, and makes it very easy to use.

Three batch files are set up for the automatic program execution. AUTOEXEC.BAT is used at boot up to prompt for DATE and TIME, load ZBASIC, and run SYSLOG.BAS. Use EDLIN to create a file named AUTOEXEC.BAT. When the Z-100 boots, COMMAND.COM looks on the default drive for a file named "AUTOEXEC.BAT", and if it exists, the batch facility is automatically invoked to execute the commands contained in AUTOEXEC.BAT. When this happens, execution of the normal TIME and DATE commands at startup is bypassed. For this reason, these commands must be included in the AUTOEXEC.BAT file.

Enter the following lines in AUTOEXEC.BAT:

1. DATE
2. TIME
3. LOGIN

Create another file called LOGIN.BAT with the following line:

1. ZBASIC SYSLOG

Now, create the third file, LOGOUT.BAT, with the following line:

1. ZBASIC LOGOUT

Note that the line numbers are not part of the files, they are only for reference; this is the way they are listed when in EDLIN.

You now have 3 files labeled AUTOEXEC.BAT, LOGIN.BAT, and LOGOUT.BAT. On bootup, you'll be prompted for Date and Time, LOGIN will be executed, ZBASIC will be installed, and SYSLOG will be loaded and run; all automatically. AUTOEXEC.BAT and LOGIN.BAT could have been put in one

```
10 ON ERROR GOTO 350 'Error trap
20 T=TIME/3600 ' Find decimal hours (24 hour clock) from seconds
30 OPEN"I",#1,"LOGCOM.XQT" 'Open input file
40 OPEN"O",#2,"COPY" 'Open output file to make copy of input file
50 IF EOF(1) THEN 90   'Test for end of input file
60 LINE INPUT#1,A$     'Read input
70 PRINT#2,A$          'Send to output
80 GOTO 50             'Continue till done
90 CLOSE#1   'Close input
95 REM Test last input string  If 2 left char. are "=B"
       then not logged in
97 REM If 2 left char  are "=E" then everything is A-OK
100 L$=LEFT$(A$,2):IF L$="=B"THEN 390:IF L$="=E"THEN 110 ELSE 340
110 R$=RIGHT$(A$,2) 'Test right 2 characters and
120 R=VAL(R$)       'determine their value- remember R
130 KILL "LOGCOM.XQT"  'Kill input file-
    it will be replaced with a new one
140 IF R=59 GOTO 270  'Spreadsheet full!  End of this file
150 PRINT "IS THIS THE LAST ENTRY FOR THIS FILE ?"
          'End of the month?
160 PRINT "[Y]ES OR [N]O "
170 B$=INKEY$
180 IF B$="Y" OR B$="y" GOTO 270                'End of this file ?
190 IF B$="N" OR B$="n" GOTO 210 ELSE 170
    'We'll accept Y or y or N or n -- but nothing else
200 IF B$="" GOTO 170
210 PRINT#2,USING"## ##";T  'print the time
220 R=R+1    'increment the row for next set of entries
230 PRINT#2,"=B",USING"##";R
    'set it up for the date entry next login
240 CLOSE
250 NAME "COPY" AS "LOGCOM.XQT"
    'The new file I told you about in 130
260 SYSTEM  'All done - return to system
270 PRINT#2,USING"## ##";T
    'This will be the last set of entries for this file
280 PRINT#2,"!"
    'This tells PeachCalc to recalculate the spreadsheet
290 PRINT "THIS IS THE FINAL ENTRY FOR THIS FILE"
300 PRINT "IT HAS BEEN NAMED 'SYSLOG.XQT'"
    'Use this file to execute spreadsheet
310 PRINT "FILE 'LOGCOM.XQT' HAS BEEN KILLED"
320 CLOSE
330 NAME "COPY" AS "SYSLOG.XQT":SYSTEM
340 PRINT "FILE READING ERROR":SYSTEM 'Arrived here from 100
350 IF ERR=53 AND ERL=30 THEN GOTO 370  'No file found in 30
355 IF ERR=58 AND ERL=330 THEN GOTO 400
    'File already exists in 330
360 ON ERROR GOTO 0   'Turn off error trap
370 PRINT "FILE NOT FOUND. ARE YOU LOGGED IN?"
380 SYSTEM
385 REM Arrived her from 355
390 PRINT "NOT LOGGED IN":SYSTEM 'File was found-but error(L$="=B")
400 PRINT "**********************************************************"
402 PRINT "**********WOOPS!---'SYSLOG.XQT'ALREADY EXISTS**********"
403 PRINT "**********FILENAME 'COPY' WILL BE RETAINED**********"
405 PRINT "**********************************************************"
410 PRINT "**********YOU NEED TO DO SOME HOUSEKEEPING!**********"
415 PRINT "**********************************************************"
420 SYSTEM
430 END
```

file, but the LOGIN.BAT file allows you to logout and log back in again without re-booting the computer. Suppose during a particular session you are working on a report, which is code 2, and wish then to spend some time on another category, say something personal. Just type LOGOUT (with the SYSLOG disk installed), and then when the system prompt reappears after the logout procedure, type LOGIN. This will load the LOGIN.BAT file without prompting for date and time. The proper date and time will be entered in the command file in the appropriate place.

Now these 3 batch files are all that's needed to make the log nearly automatic. You only have to enter date and time once at the beginning of each session, and the software takes care of the rest. With a real time clock, the date and time could even be dispensed with!

Summary

A software system has been presented for maintaining a personal computer log. The most complicated part of the system is the spreadsheet, though if the one presented here is used without changes, it could easily be installed in a couple of hours. The remainder of the software should take much less than that. For those with other spreadsheets, some modifications may be required, but I wouldn't expect it to be a major undertaking. Of course, if the spreadsheet has a different command structure, the BASIC file management programs would have to be modified to accommodate the new commands. I have tried to include enough comments in the programs to enable one to easily adapt them to other systems.

The system is quite easy to use; it takes about 5 minutes a month to update the records, if time is included to run a printout of the spreadsheet. The only real improvements that could be made would be to find a way to do the spreadsheets completely automatically, and to have a real time clock in the system so the date and time wouldn't have to be entered. The system as it stands is a natural for those lucky enough to have a Winchester.

One thing I wasn't able to do on my system, was to have the spreadsheet print out automatically from the command file. I entered all the commands that I thought necessary, but the only print out I could get was one block; the active block! Maybe somebody out there knows what I might have been doing wrong? *

Heath/Zenith Users' Group

ZPC Update #2

Pat Swayne
HUG Software Engineer

This is the second in a series of support articles for ZPC, a program that allows Z-100 (dual processor) computers to run IBM and/or Z-150 software. Since the initial release of ZPC, I have discovered some bugs and oversights, and have worked out patches to correct them. Although these patches require the DEBUG program (supplied with MS-DOS), you do not have to know how to use DEBUG to apply the patches. All you need to know is how to create files using an editor or word processor. You will create "batch files", which when executed, will apply the patches automatically. I will also present corrections for the assembly source code, but you do not have to make those corrections unless you want to.

In the second half of this article, I will present patches for the PC version of LOTUS 123, FRAMEWORK, and DBASE III, that will allow them to be used on a Z-100 under ZPC. I will also present a change to the patch for SuperCalc3, to work with the version currently being shipped by Heath/Zenith.

Before I get into the patches, here is a list of programs that will run under ZPC without being patched: DAC Easy Accounting, Norton Utilities, RUN/C (C language interpreter), Print Shop, Art Gallery (companion to Print Shop). I am not sure which level of ZPC each program requires (users called in about them) except for RUN/C, which will run under either ZPC2 or ZPC3.

Patches To ZPC

The following patches are all for ZPC3.COM (not ZPC1.COM or ZPC2.COM). If your ZPC3.COM file is dated BEFORE 10-4-85, you must make these patches before you can run FRAMEWORK, DBASE III, and potentially other IBM PC software. If your ZPC3.COM file is dated on or after 10-4-85, you do not have to make the patches in this section. It is assumed that you have prepared a ZPC system disk as instructed in the ZPC documentation, and have copied ZPC3.COM to it and renamed it to ZPC .COM.

Using an editor or word processor, create a file called FIXZPC.BAT on your ZPC system disk that contains this line:

```
DEBUG ZPC.COM <FIXZPC.DAT
```

Now, create a file called FIXZPC.DAT. If your original ZPC3 .COM file is dated BEFORE 9-19-85, FIXZPC.DAT should contain the lines in the first column below. If it is dated on 9-19-85, it should contain the lines in the second column. Note: The patch for ZPC dated before 9-19 contains the patch that was in the first ZPC Update article, so you do not have to worry if you did not apply that patch. (Note: When creating a .DAT file, be sure that you enter the lines exactly as shown, including blank lines.)

```
E118E
E8 30
E18DD
80
E1A04
D0
E1EE7
20
A1A31
CALL 30E0
NOP

A19EB
CALL 30F0

A23A1
JMP 30A0

A30A0
CMP AH,3F
JZ 30B2
CMP AH,40
JZ 30BA
CMP AH,48
JZ 30C1
JMP 23DD
CMP BX,0
JNZ 30AF
JMP 23AB
CMP BX,1
JNZ 30AF
JMP 30B7
CMP BX,FFFF
JNZ 30AF
PUSHF
PUSH CS
MOV AX,30D0
PUSH AX
MOV AH,48
JMP 30AF
CMP BX,1000
JC 30DB
SUB BX,1000
```

```
E118E
E8 30
E18E3
80
E1F06
3F
A1A50
CALL 30E0
NOP

A23C0
JMP 30A0

A30A0
CMP AH,3F
JZ 30B2
CMP AH,40
JZ 30BA
CMP AH,48
JZ 30C1
JMP 23FC
CMP BX,0
JNZ 30AF
JMP 23CA
CMP BX,1
JNZ 30AF
JMP 30B7
CMP BX,FFFF
JNZ 30AF
PUSHF
PUSH CS
MOV AX,30D0
PUSH AX
MOV AH,48
JMP 30AF
CMP BX,1000
JC 30DB
SUB BX,1000
STC
STI
RETF 2

A30E0
```

```
STC
STI
RETF 2

A30E0
XOR DH,DH
MOV SI,DX
MOV AX,CX
RET

A30E8
MOV BYTE PTR [12E],0
JMP 11AE

A30F0
MOV ES,[10E]
JMP 1A31

W
Q
```

After you have created the FIXZPC.DAT file, copy DEBUG.COM from your MS-DOS distribution disk to your ZPC system disk, log onto it, and enter

```
FIXZPC
```

at the system prompt, and hit RETURN. (Note: When you run any of these auto-installing patches, ZPC (if installed) must be in the Z-100 mode, and the HUG FASTIO program (if installed) must be disabled.) Your ZPC file will be automatically patched, and brought up to date.

ZPC And MS-DOS Version 3

MS-DOS version 3 is now available for the Z-100. If you have acquired it, then you may have found that ZPC3 does not run properly under it. Recent shipments of ZPC contain a file called ZPC3A.COM on the disk, which will run under MS-DOS version 3. If your ZPC disk does not contain ZPC3A.COM, you can fix ZPC3.COM as follows. First, create a file called DOS3.BAT on your ZPC system disk that contains this line:

```
DEBUG ZPC.COM <DOS3.DAT
```

Now create a file called DOS3.DAT. This file should contain the lines in the first column below if your original ZPC3.COM file is dated BEFORE 9-19-85. If it is dated 9-19-85, use the second column. If it is dated 10-4-85, use the third column.

```
A1327
CALL 3090

E1363
A3
E1366
02 00
A1369
CALL 3098

A2901
CMP AX,[10A]
JBE 290B
CS:
MOV AX,[10A]
ES:
MOV [2],AX
SUB AX,DX

A3090
MOV AX,CS
STOSW
MOV AL,90
STOSB
RET
NOP
MOV AX,C22B
XOR DH,DH
MOV SI,DX
MOV AX,CX
RET

A30E8
MOV BYTE PTR [12E],0
JMP 11AE

W
Q
```

```
A1327
CALL 30F0

E1363
A3
E1366
02 00
A1369
CALL 30F8

A2920
CMP AX,[10A]
JBE 292A
CS:
MOV AX,[10A]
ES:
MOV [2],AX
SUB AX,DX

A30F0
MOV AX,CS
STOSW
MOV AL,90
STOSB
RET
NOP
MOV AX,C22B
STOSW
RET

E33B8
A3
E33C0
02 00
E33C8
2B
E33DA
02 0
E33E2
2B
W
Q
```

```
A132C
CALL 30F0

E1368
A3
E136B
02 00
A136E
CALL 30F8

A28FF
CMP AX,[10A]
JBE 2909
CS:
MOV AX,[10A]
ES:
MOV [2],AX
SUB AX,DX

A30F0
MOV AX,CS
STOSW
MOV AL,90
STOSB
RET
NOP
MOV AX,C22B
STOSW
RET

E33B8
A3
E33C0
02 00
E33C8
2B
E33DA
02 0
E33E2
2B
W
Q
```

After you create DOS3.BAT and DOS3.DAT, copy DEBUG.COM to your ZPC system disk, and enter

```
DOS3
```

at the system prompt, and hit RETURN. After the patch is applied, your ZPC will work properly under MS-DOS version 3. Be sure to retain a copy of ZPC3.COM without this patch (but with the first patches from this article, if necessary), so that you can run under MS-DOS version 2 if you have to.

Assembly Source Code Corrections

These corrections apply if your ZPC3.COM file is dated BEFORE 10-4-85. To correct your ZPC source files, first load the file ZPC.ASM into your editor, and locate the label SETMOD1:, and add the line with * in the comment shown below:

```
SETMOD1:CALL    CUROFF              ;TURN CURSOR OFF
        MOV     XORFLG,0            ;* KILL XOR FLAG
```

Load in the file DOS.ACM, and change the lines from the label MYSYS: to the label FUNCOK: so they look like this:

```
MYSYS:  CMP     CS:PCFLG,0          ;PC MODE ON?
        JZ      MYEXIT              ;IF NOT, EXIT
        OR      AH,AH               ;FUNCTION ZERO?
        JZ      MYEXIT              ;YES, GO TO DOS
        CMP     AH,13               ;SUPPORTED FUNCTION?
        JC      FUNCOK              ;YES
        CMP     AH,63               ;XENIX READ?
        JZ      READOK
        IF      LEVEL3
        CMP     AH,48H              ;FREE MEMORY?
        JZ      FIXSIZ              ;IF SO, FIX RESULT
        ENDIF
        CMP     AH,64               ;XENIX WRITE?
        JNZ     MYEXIT              ;UNSUPPORTED FUNC
        CMP     BX,1                ;STD OUT?
        JNZ     MYEXIT
        JMP     SHORT FUNCOK
READOK: CMP     BX,0                ;STD IN?
        JNZ     MYEXIT
FUNCOK: PUSH    BX                  ,SAVE BX
```

Locate the comment "TABLE OF I/O ROUTINES", and add these lines before it.

```
        IF      LEVEL3
,       FIX MEMORY SIZE ON FREE MEMORY CALL

FIXSIZ: CMP     BX,0FFFFH           ;ASK FOR MAX?
        JNZ     MYEXIT              ;IF NOT, EXIT
        PUSHF                       ;FAKE DOS INT.
        PUSH    CS
        MOV     AX,OFFSET FIXRET
        PUSH    AX
        MOV     AH,48H
        JMP     MYEXIT              ;CALL DOS FUNC  48H
FIXRET: CMP     BX,1000H            ;AT LEAST 1000H?
        JC      FIXEX               ;NO, LEAVE ALONE
        SUB     BX,1000H            ;ALLOW FOR VRAM
        STC
```

```
FIXEX   STI
        RETFD                           ;RETURN
        DW      2
        ENDIF

;       TABLE OF I/O ROUTINES 1-12
```

Locate the label XREAD:, and delete all lines from it to the label XRET:, except the one containing JMP XREADSI, as shown below:

```
XREAD.  JMP     XREADSI                 ;ELSE, SPECIAL CASE
XRET:   CLI
        MOV     SS,CS:STKS              ;RESTORE STACK
        MOV     SP,CS:STK
        POP     BX
        STI
        RETFD
        DW      2                       ;FLAGS SKIPPED
XREADSI:PUSH    CX
```

Locate the label XWRITE:, and delete all lines between it and the label XWRIT1 except the ones shown below:

```
XWRITE: PUSH    DI
        PUSH    BX
        MOV     DI,OFFSET MYCNOUT       ;ASSUME STD OUTPUT
XWRIT1. PUSH    SI
```

Load the file KEY.ACM, and locate the line shown below:

```
        MOV     BX,OFFSET KEYTBL2       ;ELSE, USE ALT
```

Change it to look like this:

```
        MOV     BX,OFFSET (KEYTBL2-40H) ,ELSE, USE ALT
```

Load the file PIXEL.ACM, and locate the comment "TEST FOR XOR BIT". If the instruction there is TEST AL,8, change it to

```
        TEST    AL,80H                  ;TEST FOR XOR BIT
```

Locate the label READDOT:, and change the lines from it to the comment "GET BYTE POSITION" so that they look like this. Note: Depending on the date of your PIXEL.ACM file, these changes may have already been made.

```
READDOT:MOV     CSFLG,0                 ;DISABLE CURSOR
        PUSH    BX
        PUSH    CX
        PUSH    DX
        IF      LEVEL3
        MOV     ES,PCRAM                ,POINT TO PC VIDEO
        CALL    GVADR1                  ;GET VIDEO ADDRESS
        ELSE
        CALL    GVADR                   ;GET VIDEO ADDRESS
        ENDIF
        MOV     AH,DL                   ;PUT POSITION IN AH
        CMP     MODE,6                  ;HI RES MODE?
        JZ      HIRES1                  ;GO DO IT
        SHL     AH,1                    ;DOUBLE POSITION
        INC     AH                      ;ADD 1
        MOV     AL,DL                   ;GET BYTE POSITION
```

A few lines below the above change, locate the comment "INVERT IT". If the instruction there is "NOT DL", change it to this:

```
        NOT     AL                      ;INVERT IT
```

Locate the label GVADR:, and change the code from there to the comment "SAVE ROW" so that it looks like this:

```
GVADR:  MOV     AX,VIDRAM               ;POINT TO VIDEO RAM
        ADD     AX,100H                 ;SKIP TEXT AREA
        MOV     ES,AX
GVADR1: XOR     DH,DH                   ;ENSURE ROW IN RNGE
        MOV     SI,DX                   ;SAVE ROW
```

Assembly Corrections For MS-DOS Version 3

If you would like to alter your assembly files so that ZPC can be assembled for use under either version 2 or version 3 of MS-DOS, load the file COND.ACM, and add this line as the last one in the file:

```
DOS3    EQU     TRUE                    ;ASSEMBLE FOR DOS VERS  3
```

Load the file ZPC.ASM, and add these lines near the beginning of the file, with the other EQUate statements.

```
        IF      DOS3 AND LEVEL3
CODE1   EQU     0A326H                  ;CODE ALTERED IN MS-DOS
CODE2   EQU     2
CODE3   EQU     0C22BH
        ELSE
CODE1   EQU     8926H                   ;CODE ALTERED IN MS-DOS
CODE2   EQU     236H
CODE3   EQU     0
        ENDIF
```

Locate the label NSZCHK:, and change the lines from the MOV AX,CS statement above it to the label to look like this:

```
        MOV     AX,CS
        STOSW
        ENDIF
        IF      DOS3 AND LEVEL3
        MOV     AL,90H
        STOSB
        ENDIF
NSZCHK: MOV     AX,3
        CALL    SETMODE                 ;SET DEFAULT VIDEO MODE
```

Locate the label NSZCHK1:, and change the lines from the MOV ES,BIOSRAM instruction above it to the label to look like this:

```
        MOV     ES,BIOSRAM              ;ELSE, POINT TO BIOS RAM
        MOV     AX,CODE1
        STOSW                           ;RESTORE OLD SIZE CODE
        MOV     AX,CODE2
        STOSW
        ENDIF
        IF      LEVEL3 AND NOT DOS3
        MOV     AL,LOW CODE3
        STOSB
        ENDIF
        IF      LEVEL3 AND DOS3
        MOV     AX,CODE3
        STOSW
        ENDIF
NSZCHK1:MOV     AL,VPVAL
        OUT     VRPORT,AL               ;SET OLD VR PORT VALUE
        RET
```

Locate the label SIZCHK:, and change the code from there to the RETF statement to look like this:

```
SIZCHK:
        ENDIF
        IF      LEVEL3 AND NOT DOS3
        CMP     SI,CS:MEMSV             ;CHECK SIZE
        JBE     SIZOK                   ,IT'S OK
        MOV     SI,CS:MEMSV             ;ELSE, LIMIT IT
SIZOK:  MOV     ES:MEMLIM,SI            ,SAVE NEW SIZE
        RETF
        ENDIF
        IF      LEVEL3 AND DOS3
        CMP     AX,CS:MEMSV             ;CHECK SIZE
        JBE     SIZOK                   ,IT'S OK
        MOV     AX,CS:MEMSV             ;ELSE, LIMIT IT
SIZOK:  MOV     ES:MEMLIM,AX            ;SAVE NEW SIZE
        SUB     AX,DX
        RETF
        ENDIF
```

Locate the label SSRCH:, and change the code from one line above it to the label SCFND: to look like this:

```
        MOV     AX,CODE1                ;SEARCH FOR THIS
SSRCH:  REPNZ   SCASW                   ;DO THE SEARCH
        JNZ     SHSRCH                  ;NOT FOUND, SHIFT SEARCH
        CMP     ES:WORD PTR [DI],CODE2  ;IS THIS IT?
```

```
        JNZ     SSRCH                   ;NO
        CMP     ES:BYTE PTR 2[DI],LOW CODE3 ;CHECK LAST BYTE
        JNZ     SSRCH                   ;NO
        JMP     SHORT SCFND             ;FOUND SIZE CODE
SHSRCH. MOV     DI,4001H                ;START SEARCH HERE, NOW
        MOV     CX,4000H
SSRCH1. REPNZ   SCASW                   ;DO THE SEARCH
        JNZ     SCNFND                  ;CODE NOT FOUND
        CMP     ES:WORD PTR [DI],CODE2  ;IS THIS IT?
        JNZ     SSRCH1                  ,NO
        CMP     ES:BYTE PTR 2[DI],LOW CODE3 ;CHECK LAST BYTE
        JNZ     SSRCH1                  ;NOT IT
SCFND:  DEC     DI                      ;BACK UP TO CODE START
```

Running LOTUS, DBASE III, And FRAMEWORK

In order to run the PC version of Lotus 1-2-3, Dbase III, and Framework, you must remove or bypass the copy protection on the system disks of these programs. A utility that we have found that will work with all three is Unlock, which is available from TranSec Systems, Inc., 701 E. Plantation Circle, Plantation, FL 33324, (305) 474-7548. Ask for Unlock Album "A". It sells for $49.95, plus $4.00 shipping and handling. Lotus 1-2-3 (but not the others) can also be run using a program called NOKEY, which is supplied with COPYIIPC, which is available from Central Point Software, Inc., 9700 SW Capitol Hwy. #100, Portland, OR 97219, (503) 224-5782. Note: An IBM PC, Z-150, or similar computer is required to copy disks using either Unlock or COPYIIPC. They will not work under ZPC. If you do not have access to such a computer, you can still use NOKEY to run Lotus 1-2-3 under ZPC.

Running Lotus 1-2-3

To prepare Lotus 1-2-3 for operation under ZPC, create a file called LTSPCH.BAT, which contains this line:

```
DEBUG <LTSPCH.DAT
```

Also create a file called LTSPCH.DAT, which contains these lines:

```
NIBM0COL0.DRV
L
F1145,114E,90
E139F
90
E13A3
90
E13A7
90
E13AB
90
E14BE
90
E14F3
90
W
NIBM1G2.DRV
L
E18B
90
E19A
90
E19F
90
E1C3
90
E1C9
90
A186
MOV AX,4
INT 10

W
Q
```

If your computer does not support color (or gray scale), change the first line of the file LTSPCH.DAT to NIBM0B&W.DRV. Copy LTSPCH.BAT and LTSPCH.DAT to a duplicate of your Lotus "Utility Disk" along with DEBUG.COM, log on to the disk, and enter

```
LTSPCH
```

at the system prompt, and hit return. When the procedure is complete, you can remove LTSPCH.BAT, LTSPCH.DAT, and DEBUG from the disk. Now follow the "Getting Started" manual instructions for installing drivers, and select either "COLOR" or "B&W". You can perform the other installation procedures in the manual before or after running LTSPCH. Part of the first installation procedure, the attempt to use the SYS command to make the Lotus disk bootable, will fail, and you will see an error message. All this means is that your Lotus disks will not be bootable. The procedure for running Lotus using NOKEY would be to boot up with your ZPC system disk (if you have not already done so), load ZPC and set the PC mode, run NOKEY, and then insert your lotus system disk in drive A:, and type LOTUS. Of course, you can also run LOTUS from a hard disk or 8-inch drive after you have loaded ZPC and NOKEY. If you have used Unlock to make a non-protected Lotus system disk, the procedure is the same, except that you do not have to run NOKEY. Note: if you use 5.25 inch drives, always start Lotus in drive A:.

Running Dbase III

Note: These instructions apply to Dbase III version 1.1. Version 1.0 will not run under ZPC. To prepare Dbase III for use under ZPC, use Unlock to make an unprotected copy of system disk I. Create a file called DBPCH.BAT, which contains these lines:

```
REN DBASE.EXE DBASE.BIN
DEBUG DBASE.BIN <DBPCH.DAT
REN DBASE.BIN DBASE.EXE
```

Create a file called DBPCH.DAT, which contains these lines:

```
F25B0,25B4,90
F262F,2633,90
E25BA
90
E2639
90
A25DE
MOV AL,20
INT 51
NOP
NOP

W
Q
```

Copy DBPCH.BAT and DBPCH.DAT to your unprotected system disk I along with DEBUG.COM, log on to the disk, and enter

```
DBPCH
```

at the system prompt, and hit RETURN. When the procedure is complete, you can remove DBPCH.BAT, DBPCH.DAT, and DEBUG.COM from the disk. Now, make a duplicate of the Dbase utility disk, and create a file called DBCPCH.BAT which contains these lines:

```
REN DCONVERT.EXE DCONVERT.BIN
DEBUG DCONVERT.BIN <DBCPCH.DAT
REN DCONVERT.BIN DCONVERT.EXE
```

Prepare a file called DBCPCH.DAT which contains these lines:

```
F15B0,15B4,90
F162F,1633,90
E15BA
```

```
90
E1639
90
A15DE
MOV AL,20
INT 51
NOP
NOP

W
Q
```

Copy DBCPCH.BAT, DBCPCH.DAT, and DEBUG.COM to the duplicate Dbase utility disk, log on to the disk, and enter

```
DBCPCH
```

at the system prompt, and hit RETURN. When the operation finishes, Dbase III will be ready to run. If you want to run it from an 8-inch disk or hard disk, do not perform the installation procedure indicated in the Dbase manual. Instead, just copy the files from your prepared Dbase disks to the required drive. To run the files, load ZPC, set the PC mode, and run DBASE. For proper operation, NUM LOCK must be off. Control-keypad keys are produced by pressing F0 followed by the required key. Use the 1 through 9 keys (with NUM LOCK off) for control-arrow keys, etc. For normal arrow keys, the separate arrow keys will work.

Running Framework

Note: These instructions apply only to Framework version 1.1. Version 1.0 will not run under ZPC. To prepare Framework for use under ZPC, use Unlock to make an unprotected copy of system disk I. Create a file called FWPCH.BAT containing these lines:

```
REN FW.EXE FW.BIN
DEBUG <FWPCH.DAT
REN FW.BIN FW.EXE
```

Create a file called FWPCH.DAT that contains these lines:

```
NGSCREEN.DRV
L
E33C
90
E3EF
B0
E3F3
0 0
W
NFW.BIN
L
E6D5F
2D
W
NFW.OVL
L
A90
MOV AX,DS
ADD AX,1000
MOV DS,AX

G=90,97
E16A3
B0  0 0
W
Q
```

Copy FWPCH.BAT, FWPCH.DAT, and DEBUG.COM to your prepared Framework system disk I, log onto the disk, and enter

```
FWPCH
```

at the system prompt, and hit RETURN. When the procedure is completed, Framework will be ready to run under ZPC, but you need to make a few more changes using Framework itself before you can really use it. Load ZPC, set the PC mode, place your prepared Framework system disk 1 in drive A:, and run Framework by entering FW at the system prompt. Change to system disk 2 when instructed to (use a duplicate of system disk 2). Using Framework's own word processor (you may have to study your manual first), load in the file CONFIG.FW and make these changes to it. Change the value beside the parameter DEFAULT __DRIVE to a number representing the drive you will be using for data while running Framework. Use 1 for drive A, 2 for B, etc. Change the printer setting beside the parameter PRINTER1 from "LPT1" to "PRN". If you will be using a second printer or plotter, you can use "AUX" beside the appropriate parameter.

After you make these changes to CONFIG.FW, press the LINE FEED key. This will cause the edited file to be saved to disk. When you run Framework, NUM LOCK must be disabled in order for the keypad keys to work properly. If NUM LOCK is enabled, the word NUM will appear in the lower right corner of the Framework "desktop" display. Press the NUM LOCK key (F11) to disable it.

To produce Control-Return (or Control-Enter) when requested by Framework, use the LINE FEED key. To produce Control-Backspace, use the DELETE key. To produce Keypad +, use SHIFT-Keypad -, being sure that NUM LOCK is off first. To produce Control-keypad keys, press F0 followed by the appropriate key. Use 1 through 9 on the keypad, not the separate arrow, etc., keys for Control-keypad keys. When you need to use the SCROLL LCK key in Framework to switch between drives and frames, use Alt-= instead. To produce Alt-=, press the HELP key followed by the = key. This last key change is necessary because ZPC does not support the way Framework uses the SCROLL LCK key (which involves the type-ahead buffer).

Alternate Framework Patch

When you run Framework on a PC-type computer with a color/graphics board (which is what ZPC emulates), it runs in the high resolution monochrome graphics mode, which allows it to display such attributes as bold print, underlining, and italics directly on the screen. However, ZPC is a bit slow in the graphics modes, so screen changes are somewhat sluggish. Framework can be made to run in the text mode, for faster screen changes, by using the following patches INSTEAD of or AFTER the ones above. Your FWPCH.BAT file should contain these lines:

```
REN FW.EXE FW.BIN
REN GSCREEN.DRV GSCREEN.GRF
REN MSCREEN.DRV GSCREEN.DRV
DEBUG <FWPCH.DAT
REN FW.BIN FW.EXE
```

The FWPCH.DAT file should contain these lines:

```
NGSCREEN.DRV
L
E324      (optional line)
3         (optional line)
E32C
90
E340
90
E46A
8
E4B1
B8
W
NFW.BIN
L
E6D5F
2D
W
Q
```

Include the optional lines above if your H/Z-100 is color compatible and you have a color monitor, or a monochrome monitor that can clearly show blue text on a black background. If you include the optional lines, underlining and italics (in word frames) will show up as blue text, as will the menu names at the top of the screen. The patch causes Framework to think that it is using an IBM monochrome card, and the bit pattern for enabling underlining with that card is the same as for blue on the color/graphics card. For best results, you should use SETZPC to disable intense color emulation when you run Framework with these patches. Note: If you leave out the optional lines, all characters will be white, with no indication of underlining or italics. Bold printing is indicated with a faint underline in either case.

SuperCalc3 Patch Correction

The patch for SuperCalc3 supplied on ZPC disks with the ZPC .COM files dated before 10-4-85 does not work with the version currently being shipped by Heath/Zenith. I have also developed patches for the IBM PC version of SuperCalc3 (version 2), which is a little different from the Heath/Zenith version. Replace the text of the file SC3PCH.DAT with the lines in the first column if you have the Heath/Zenith version, or the lines in the second column if you have the IBM PC version.

```
F2FE3,2FED,90
F30BE,30C8,90
FCC9B,CCA5,90
FCCBB,CCC5,90
A30C5
AND BX,7FFF

ACCC3
AND CL,7F

E3179
90
E318D
90
EC89E
90
EC8A1
90
EC8A4
90
EC8A7
90
EC8AD
90
ED6AF
90
W
Q
```

```
F2FE3,2FED,90
F30BE,30C8,90
FCCD7,CCE1,90
FCCF7,CD01,90
A30C5
AND BX,7FFF

ACCFE
AND CL,7F

E3179
90
E318D
90
EC8DA
90
EC8DD
90
EC8E0
90
EC8E3
90
EC8E9
90
ED730
90
W
Q
```

Follow the instructions in the ZPC documentation for patching SuperCalc3.

ZPC is available from the Heath Users' Group as part no. 885-3030-37, for $40.00 (plus shipping/handling). See the New Products pages of any REMark issue for ordering information.

Lotus 1-2-3 is a trademark of Lotus Development Corp. Dbase III and Framework are trademarks of Ashton--Tate. SuperCalc3 is a trademark of Computer Associates International, Inc. ✱

Heath/Zenith Users' Group

PIE/Text To PeachText File Conversion

J. D. Ross
Ross Custom Electronics
1307 Darlene Way – Suite A12
Boulder City, Nevada 89005

For many Heath users from the days of the H8/H89, the transition from PIE and TEXT to Peachtext has been a difficult one. For some, it was just simply easier to purchase PIE for the Z100 CP/M and ZDOS rather than commit to learning the new procedures associated with Peachtext. Besides all the new format control codes, there was the overwhelming temptation to hit a carriage return when nearing the end of the line. PIE/TEXT was, of course, very forgiving of this holdover from the antiquated typewriter. I was admittedly one of the last holdouts for using Peachtext, but it soon became apparent that if I was to take advantage of some of the more sophisticated features of my letter quality printer (JUKI 6100), I would just have to take the plunge. Then the serious problems began.

Literally dozens of disks were full of documents created with PIE, for runoff under TEXT. Of course these could be merely transferred to ZDOS with RDCPM, but when these files were brought up under Peachtext, every line contained a carriage return symbol (tilde). When an attempt was made to take the carriage return out, the first word from the following line became joined with the last word from the current line. Then it was necessary to enter the insert character mode to separate the two words. And so it went through the entire document.

To accommodate the above conversions with a minimum of suffering clearly required some kind of translation program: one that would strip all the carriage returns except at the end of a paragraph (i.e. where there was a .sp in the TEXT file), fix the left margin (.in) and right margin (.rm) commands, and change all the "unknown" TEXT commands to a backslash command so that it won't be printed as part of the document. The program which follows accomplishes these things, and a few others. It does not handle all the TEXT commands by any means, but it does get the document into a form which can be handled by Peachtext without too much anguish.

As mentioned earlier, one of the primary reasons for moving to Peachtext was to utilize some of the advanced features of the JUKI 6100, particularly proportional spacing. When Peachtext is set to a Diablo 630 type printer, right justified, proportionally spaced documents can be created, with results very close to typesetting. This is probably one of the lowest cost methods for creating professional looking documents. Of course, a special printwheel is required when Peachtext is instructed to use proportional spacing. I have been using a Madeleine PS wheel (number 02–52) with excellent results.

One final note. For some reason when the command "PROP ON" is embedded in the document, Peachtext does not execute it properly. It displays on the Print Format Screen as if it has recognized it, but the document may not be justified properly, and character spacing may be erratic. Executing a PROP ON command again from the Print Format Screen Command Line will put everything back to normal.

```
1   'PIE TO PEACHTEXT CONVERSION PROGRAM
10  'BO$ RANDOM OUTPUT BUFFER
12  'BI$ INTERMEDIATE OUTPUT BUFFER
14  'EF  END FLAG - SET IF LINE ENDS IN . ! OR ?
16  'LM  CURRENT LEFT MARGIN SETTING
17  'LC  LINE COUNTER FOR DISPLAY OF "IN PROGRESS" DOTS
18  'RM  CURRENT RIGHT MARGIN SETTING
19  'NF NO FILL FLAG 1=TRANSFER TEXT LINES DIRECTLY 0=FILL
100 DEFINT A-Z:CLS:BI$=""
120 PRINT "PIE/TEXT TO PEACHTEXT FILE CONVERSION UTILITY"
125 PRINT "      ROSS CUSTOM ELECTRONICS - 11/1/84"
140 PRINT
160 LINE INPUT "Enter Name of File to Convert ";A1$
180 OPEN "I",1,A1$
200 FOR I=1 TO LEN(A1$):IF MID$(A1$,I,1)="." THEN GOTO 300
220 NEXT I
240 A2$=A1$+".DOC":GOTO 400
300 A2$=LEFT$(A1$,I)
320 IF RIGHT$(A1$,LEN(A1$)-I)<>"DOC" THEN A2$=A2$+"DOC"
400 PRINT
    "Enter Name of File for Peachtext Output: <";A2$;">";
```

```
420 LINE INPUT "";A3$:IF A3$<>"" THEN A2$=A3$
440 ON ERROR GOTO 600:OPEN "I",2,A2$:ON ERROR GOTO 0
480 PRINT:LINE INPUT
    "FILE EXISTS - OK TO DELETE? (Y/N) <Y>";A3$
500 IF A3$="" THEN GOTO 580
520 IF A3$="Y" THEN GOTO 580
530 IF A3$="y" THEN GOTO 580
540 PRINT:GOTO 200
580 CLOSE #2:KILL A2$:GOTO 640
600 RESUME 620
620 ON ERROR GOTO 0:CLOSE #2
640 OPEN "R",2,A2$
650 FIELD #2,128 AS BO$:PRINT:PRINT
    "BEGINNING CONVERSION---"
660 EF=0:'INIT END FLAG
670 LM=1:RM=60:NF=0:LC=0
999 '
1000 IF EOF(1) THEN GOTO 2500
1010 LINE INPUT #1,AS$:GOSUB 4600
1012 LC=LC+1:IF LC=10 THEN LC=0:PRINT ".";
1015 IF A1$="" THEN IF NF=1 THEN GOSUB 5560:GOTO 1000
1016 IF A1$="" THEN IF NF=0 THEN GOTO 1000
1020 IF LEFT$(A1$,1)<>" " THEN GOSUB 5000:GOTO 1000
1021 IF EF=1 THEN GOSUB 5560:'CR LF FOR PREVIOUS LINE
1022 EF=0:'NEW LINE - INIT END FLAG
1025 IF LEN(A1$)<3 THEN GOTO 2000
1026 IF LEFT$(A1$,3)=".nf" THEN NF=1:GOTO 1000
1027 IF LEFT$(A1$,3)=".fi" THEN NF=0:GOTO 1000
1030 IF LEFT$(A1$,3)=".sp" THEN GOSUB 5500:GOTO 1000
1040 IF LEFT$(A1$,3)=".bp" THEN GOSUB 6000:GOTO 1000
1050 IF LEFT$(A1$,3)=".in" THEN GOSUB 6500:GOTO 1000
1060 IF LEFT$(A1$,3)=".rm" THEN GOSUB 7000:GOTO 1000
1070 IF LEFT$(A1$,3)=".ce" THEN AS$="\CTR":GOSUB 4500:
     GOTO 1000
1080 IF LEFT$(A1$,3)=".ju" THEN AS$="\JUST":GOSUB 4500:
     GOTO 1000
1090 IF LEFT$(A1$,3)=".nj" THEN AS$="\LEFT":GOSUB 4500
     GOTO 1000
1998 '
1999 'TRANSFER ANY OTHER CHARACTERS PRECEDED WITH
      BACKSLASH AND FOLLOWED BY XXXX
2000 CX=ASC("\"):GOSUB 4000:IF LEN(A1$)<2 THEN GOSUB 5560:
     GOTO 1000
2020 AS$=RIGHT$(A1$,LEN(A1$)-1)+" XXXXXXXXXXXX":
     GOSUB 4500:GOTO 1000
2499 '
2500 IF BI$="" THEN GOTO 2520
2510 LSET BO$=BI$:PUT #2
2520 CLOSE #1:CLOSE #2
2530 PRINT:PRINT "FILE CONVERSION COMPLETE"
2540 END
3998 '
3999 'SUBROUTINE TO PUT CHARACTER CX IN STRING BI$
4000 BI$=BI$+CHR$(CX)
4020 IF LEN(BI$)=128 THEN LSET BO$=BI$:PUT #2:BI$=""
4040 RETURN
4498 '
4499 'SUBROUTINE TO INSTALL AS$ TO OUTPUT FILE,
      FOLLOWED BY CR,LF
4500 FOR IX=1 TO LEN(AS$):CX=ASC(MID$(AS$,IX,1)):
     GOSUB 4000:NEXT IX
4510 GOSUB 5560:RETURN
4598 '
4599 'SUBROUTINE TO REMOVE TABS FROM AS$
      AND PUT RESULT IN A1$
4600 A1$="":I1=1
4610 FOR I=1 TO LEN(AS$):AT$=MID$(AS$,I,1):
     IF AT$=CHR$(9) THEN GOTO 4650
4620 A1$=A1$+AT$:I1=I1+1
4630 NEXT I:RETURN
4650 I2=INT((I1+7)/8)*8:'NEXT TAB POSITION MINUS 1
4660 FOR I3=I1 TO I2:A1$=A1$+" ":I1=I1+1
     NEXT I3:GOTO 4630
4998 '
4999 'SUBROUTINE TO TRANSFER ONE LINE TO
      RANDOM OUTPUT FILE
5000 IF EF=1 THEN GOSUB 5050:GOSUB 5050:EF=0
5005 FOR IX=1 TO LEN(A1$):CX=ASC(MID$(A1$,IX,1))
     GOSUB 4000:NEXT IX
5010 IF NF=1 THEN GOSUB 5560:RETURN
     'IF NO FILL, PUT IN CR, LF
5015 CX=ASC(MID$(A1$,LEN(A1$),1))
5020 IF CX=46 THEN EF=1:RETURN:'IF
5024 IF CX=33 THEN EF=1:RETURN:'IF !
5026 IF CX=63 THEN EF=1:RETURN:'IF ?
5028 IF CX=58 THEN EF=1:RETURN:'IF .
5030 GOSUB 5050:RETURN
5050 CX=32:GOSUB 4000:RETURN:'PUT SPACE IN OUTPUT FILE
5498 '
5499 'SUBROUTINE TO HANDLE  sp
5500 IF LEN(A1$)<=4 THEN IX=1:GOTO 5515
5510 IX=VAL(RIGHT$(A1$,1))
5515 IF IX=0 THEN RETURN
5520 FOR IY=1 TO IX:GOSUB 5560:NEXT IY:RETURN
5560 CX=13:GOSUB 4000:CX=10:GOSUB 4000:RETURN
5999 '
6000 'SUBROUTINE TO HANDLE  bp
6010 AS$="\NP":GOSUB 4500:RETURN
6498 '
6499 'SUBROUTINE TO HANDLE  in
6500 IF MID$(A1$,5,1)<>"+" THEN GOTO 6550
6510 GOSUB 6600:LM=LM+CX:GOSUB 6620:RETURN
6550 IF MID$(A1$,5,1)<>"-" THEN GOTO 6570
6560 GOSUB 6600:LM=LM-CX:IF LM<1 THEN LM=1
6565 GOSUB 6620:RETURN
6570 GOSUB 6650:LM=CX:IF LM<1 THEN LM=1
6580 GOTO 6565
6600 CX=VAL(MID$(A1$,6,LEN(A1$)-5)):IF CX<0 THEN CX=0
6610 RETURN
6620 N1=LM:GOSUB 7150:AS$="\LM"+N2$:GOSUB 4500:RETURN
6650 CX=VAL(MID$(A1$,5,LEN(A1$)-4)):IF CX<0 THEN CX=0
6660 RETURN
6998 '
6999 'SUBROUTINE TO HANDLE  rm
7000 IF MID$(A1$,5,1)<>"+" THEN GOTO 7050
7010 GOSUB 6600:RM=RM+CX:GOSUB 7120:RETURN
7050 IF MID$(A1$,5,1)<>"-" THEN GOTO 7070
7060 GOSUB 6600:RM=RM-CX:IF RM<1 THEN RM=1
7065 GOSUB 7120:RETURN
7070 GOSUB 6650:RM=CX:IF RM<1 THEN RM=1
7080 GOTO 7065
7120 N1=RM:GOSUB 7150:AS$="\RM"+N2$:GOSUB 4500:RETURN
7150 N2$="":N1$=STR$(N1):FOR I=2 TO LEN(N1$):
     N2$=N2$+MID$(N1$,I,1):NEXT I:RETURN
```

*

ZDUMP

Full Screen Disk/Memory Utility

Paul W. Franchina
4102 Greenfern Drive
Orlando, FL 32810

Overview

Have you ever been right in the middle of saving some important file to disk and the power went out? Or maybe you've been visited by the ferocious WILD INTERRUPT at the point where you were going to do a save. How about the old slip DEL *.BAS when you really wanted *.BAK? You reboot or restart to find that your disk is nothing but trash? Ah yes, we all have our sad stories to tell. There is nothing as frustrating as having a lot of hard work vaporize in front of your eyes. Well, I have found a program that may not be the answer to every conceivable problem that can befall the avid computerist, but it has helped me out of more than just a few tight spots!

ZDUMP from Sunflower Software of Shawnee, Kansas comes real close to being the best answer for most of the problems you're likely to encounter. This program falls into the category of a system utility. It is a very powerful tool to access both disk and/or computer memory. It is also a program that deserves an extra measure of respect, for while it can be your best friend in a tight spot, it also gives you the power to wreak pure havoc on your disks. The program comes on a single disk along with a well written loose leaf type manual. The documentation, only 14 pages in length, covers all the functions and operations of the program very clearly and with adequate warnings where necessary. Let me explain at this point, that ZDUMP can in no way damage any of your hardware, but has the ability to write directly to your disks without regard for what may be in a certain location or informing the operating system of what it's doing. This means that if you decide you want to put garbage in your directory listing, or maybe an inspirational thought where the File Allocation Table should be, ZDUMP will accommodate your every wish regardless of whether you'll have a readable disk when you're done or not. Now that you have an idea where we're going, let's take a closer look at ZDUMP.

ZDUMP requires a Z100 with 128K of RAM and a single disk drive. However, to be truly useful for recovering information from bad disks, a second drive is a necessity. ZDUMP signs on with a copyright message and the option to enter one of the three general modes of operation. These modes are; FILE, DEVICE, and MEMORY. Each mode has numerous suboptions and all are accessible through the use of the Z100 function keys. Full use is made of the 25th line for command options. A nice feature is that all three modes have almost identical command options, and with minor exceptions, learning one mode is all that is necessary to be comfortable with all aspects of the program. Even the general modes are fairly self explanatory. You can get to your information on disk by specifying either the filename or the device (drive) where the file should be. If the file you're looking for is not on the default drive, then the drive specifier must precede the filename to be located. Any page of available memory may be accessed by entering MEMORY mode. Also, ZDUMP is memory resident, so the disk can be removed from the drive once it has loaded and signed on. This can save a lot of file swapping and copying. If you're trying to recover files from a corrupted disk to fresh disk, the second drive can be used. Once operating within a mode, the screen displays both the HEX and ASCII representations of the information simultaneously. The display is a format of 16 lines, each line showing an address, then 16 bytes of the HEX code followed by the corresponding characters in ASCII. Non-printing ASCII characters are represented by a period. Space at both the top and bottom of the display is liberally used for status information and any prompts needed for the current function.

General Commands

As was previously mentioned, there is much commonality of commands between the different modes available to the user. All are implemented using the function keys, with the status line listing a menu of all commands available at a given point. What follows will be a brief explanation of commands common to all modes.

SELECT (F1) Allows you to choose either the sector on disk or the relative page in memory you wish to view. The program will not allow going beyond an end-of-file, end-of-disk, or outside system memory.

REWIND (SHIFT/F1) This allows an easy method of going back to the beginning of a particular file/ sector/ address. It would be the same as using SELECT with the starting location of zero.

EDIT (F2) This will put you into the edit mode whereby you can alter the information on the screen. A more in depth discussion of this function can be found further on in the text.

FOR YOUR EYES ONLY

Code Name: "Rambo" Real Name: "Turbo Lightning™"

User Groups find out Borland's secrets first, because you were the first to buy Borland products — and we'll never forget that. So we're telling you about Turbo Lightning before anyone else knows. (There isn't room here for the whole story, but you can read a fuller account in the COMDEX issue of PC Magazine).

We've had an astonishing product, code-named 'Rambo' under secret development for several months. Based on electronic versions of the Random House® Dictionary and the Random House Thesaurus, our new Turbo Lightning is the most amazing on-line information system since your brain.

It checks your spelling — as you type — and while you're running another program. More than that, it offers instant access to synonyms — and is intelligent enough to learn from you. In fact, the more you use Turbo Lightning, the smarter it gets.

Turbo Lightning is the 'engine' that powers what is also being announced on this page, the Turbo Lightning Library™. Soon-to-be-released volumes will include extended thesauruses, specialized dictionaries and other forms of electronic information.

Public announcement of Turbo Lightning will quickly follow this private announcement and we'll be shipping by November 15.

Reflex™ comes in from the cold.

Some of you already know how good Reflex is. You know that it looks at data and numbers in a whole new way. A way that turns a 1-2-3 into a graphic 4-5-6 because it shows you relationships and interrelationships that were hidden in data inside 1-2-3®, dBASE®, PFS File® or any of that crowd.

With Reflex, when you look, you see. It offers you five different 'Views.' 'Form View' to build your database; 'List View' to list data in tabular form; 'Graph View' for instant graphics; 'Cross-Tab View' for instant cross-referenced relationships and 'Report View' that imports data from Reflex, 1-2-3, dBASE and PFS File than prints it in the format you want. On top of that, 'Report View' is a better report generator than 1-2-3 ever had — and at $99.95 for all five 'views,' Reflex is also the cheapest. (When we acquired Analytica Corporation, original developers of Reflex, we changed the name to "Reflex: The Analyst" and cut the price from $495.00 to $99.95). A very impressive product and we think, the greatest analytical tool since the couch.

So let's talk. 800-line Ordering Service: 800-255-8008 (outside CA), 800-742-1133 (inside CA), 6a.m. to 9p.m. Mon/Fri; 8a.m. to 5p.m. weekends; the former "Ma Bell" (408) 438-8400, ask for the Sales Department; MCI MAIL: BORLAND; CompuServe: GO BOR; Telex: 172373. And if you insist on the quill, 4585 Scotts Valley Drive, Scotts Valley, CA 95066. And thanks for your continuing support.

BORLAND
INTERNATIONAL

SEARCH (F3) ZDUMP will search files/ disks/ memory for a particular string. The string may be HEX, ASCII, or even a combination of both.

PRINT (F4) You may send information to your printer to get a hard copy. The current screen or a range of successive sectors, etc. may be sent out.

PRINT SCREEN (SHIFT/F4) This is another easy method for printing just the contents of the currently displayed screen.

SAVE (F6) A new file can be created with this command and information passed to it from ZDUMP. More on this later.

SAVE SCREEN (SHIFT/F6) Another of the short cut functions to save just the contents of the currently displayed screen.

QUIT (F7) This command allows you to exit a function or back your way out of a sequence of steps.

DOS (F8) This function terminates the program and returns control to the operating system.

Two other keys are used throughout the program with the same function. The RETURN key is used to advance to the next higher record /sector/ address, while the BACK SPACE will return to the previous one.

File Mode

If your directory has not been trashed, you can go to file directly using this mode. After specifying a filename, the first 256 bytes of the file will be displayed along with other information on the status lines. The status lines inform you of the file size. This is a decimal number of the total bytes in the file. Also displayed will be the current record number (always starting with zero) and the last record number. These last two values are given in HEX. The F5 key now functions as NEW FILE. This will allow you to give a new filename and go to the first record of it. A new function appears on this screen. F9 is the APPEND command. You may go to the last record of a file and add to it. This command puts you into the editor and writes the information to the disk automatically when the screen is full or at your command if the whole screen has not been written.

Device Mode

This mode will prompt you for a valid device identifier. Once the device name has been entered, you are presented with the first 256 bytes of sector zero. Status information is again presented, but is slightly different from the previous display. The size of a sector, in bytes, is displayed in decimal. The current sector number and the ending sector number are presented also, once again the values are in HEX. The sector size may not seem very important but ZDUMP works with both 5 1/4 inch floppies and hard disk which have 512 byte sectors and can be used for 8 inch floppies which have 1024 byte sectors. In this mode pressing the F5 key allows you to specify a NEW DEVICE and load the sector zero information from it.

Memory Mode

In memory mode you may view and/or edit any page of available system memory. ZDUMP will display the current segment and page address at the top of the screen. These values are given in HEX. The processor chip used in a Z-100 breaks the address space up into what are called segments and paragraphs. When you first enter this mode you will be presented with segment address 0000 and page address 0000. Depending on the amount of memory you have in your machine, you can view from segment 0000 to FFFF. Each segment having 256 byte pages from 00 to FF. If you have a memory map of the Z-100, you can really learn a lot about how your machine operates by digging around in the various areas of system memory. Here again, some of the function keys are altered. F1 is the SELECT ADDRESS command allowing you to set the page of memory relative to the current segment address. F5 is the key that will SELECT [the] SEGMENT you wish to view.

Edit Mode

This is the most powerful feature of ZDUMP and the one of greatest value. Once you have located the information you want, you may change it as necessary. Editing may be done in either the HEX or ASCII mode. This feature is interactive with changes being reflected in both displays as information is changed. You may move through the current display using the arrow keys. The HOME key may also be used to move the cursor to the upper left corner of the display. When in this mode, a new command list is displayed on the Status line. Some of the functions of the general command line are replaced or repositioned according to their frequency of use. F1 is used to put the editor in HEX mode while F2 allows the editing of the ASCII text. SHIFT/F3 is used to write the "new" information back onto the disk. The Shift key is used as a safeguard to prevent this function from being engaged accidentally. F5 is called the MSB key. There is sometimes a need when entering ASCII information to have the Most Significant Bit of an ASCII character set to "high". F5 will toggle in and out of this mode and the mode line will show "MSB=ON" when in effect.

One evening I was right in the middle of saving a file when the power went down. After the world returned from darkness, I restarted my machine to find a disk that was apparently blank. Not only had the file I was working on disappeared, but about forty other files as well. The DIR command from DOS came back with "File not found"; nothing else. A quick tour of the afflicted disk using the DEVICE mode showed that somehow the first entry on the disk had been cleared. As far as DOS was concerned, if the first entry was clean, therefore the whole disk must be and it would never look any further than that first entry. ZDUMP let me see that all my precious information was still there and intact. I just couldn't get at it with DOS or my applications programs. Not being a MS-DOS file scheme expert at the time (I've since become rather proficient at it), I took the easy approach. I merely copied the information from the entry below into the defective slot and voila! The disk was loaded once again with forty some odd files that could be accessed by anyone. Yes it is true, I now had two directory entries for the one particular file and another file that had vanished from existence, but we'll leave that story for the next section.

Search Mode

Since looking through 360 Kbytes of disk space or even 128 Kbytes of system RAM at 256 BYTES per screen for your lost poem may not sound especially exciting, ZDUMP has provided a way for your marvelous machine to do the dirty work for you. Since it was your poem, you must remember a phrase or two that might be unique! Yes? After you've decided on the general mode (file, disk, memory), you may enter the SEARCH mode from there. ZDUMP provides a 24 character buffer that you may enter a "string" to search for. The nice part of this is that your "string" can be either ASCII or HEX or even a combination of both. ASCII characters must be entered between quotation marks, just as is done from BASIC. HEX characters need not be with quotes, but each character must be separated by a comma. Thus to search for

the ASCII string "Paul", in HEX one would enter: 50,61,75,6C. If you were searching with a combination of the two, `escape-x' say, you would then use both the quotes for the ASCII and the comma as delimiter between the two eg. 1A,"x". Searching memory is a bit more involved than that for disks or files. An understanding of 8086/8088 addressing with its segments and paragraphs would help. There are many good books on the topic and you might even be able to talk your local Intel Representative out of one for free if you play your cards right. The documentation that comes with Zdump can get you on the road with this one, but I suggest you do your training in one of the other two modes before going after something lost in memory.

Again some of the function keys have altered meanings under this mode. F1 is used to enter the search string. Once entered in this fashion, the string will remain in force until it is replaced by a new string. F2 is used to begin the actual searching of disk, memory, or file. F3 allows you to set an upper record/sector to limit disk searches, while F4 is now used to Quit the search and return to the previous function.

Have you figured out how I resolved the double file/missing file problem from before? Well, a quick look at my backup disk (you DO keep timely backups of course) and I knew by elimination what had disappeared. It was the financial file on the 67 Spitfire I'm restoring. I had only lost a day or so's worth of information. I could use the backup and replace the lost stuff by hand, but why not use ZDUMP. This was truly a breeze. Back again to DEVICE mode, from there into SEARCH, and look for "Big Buck Import Parts". Within minutes everything had been SAVEd to disk and was back as it should be. What had started out as a potential disaster had become a simple exercise in file recovery techniques.

Save Command

Save is a command that can be accessed directly from many modes within ZDUMP. Even if there is not a current menu option for SAVE available at the time, pressing SHIFT/F6 will usually take you directly to this mode. It is this command that one uses to attempt the reconstruction of damaged files. The function keys are once again slightly altered to fit the functions unique to SAVE. F1 is used to open a file to receive the data. Any filename consistent with DOS is allowed. Opening a file with this function will delete any file having the same name `before' the new one is opened. A word of caution is required here. No attempt is made to inform you if a file by that name does in fact already exist with the option to abort and try another filename, if you want to keep the original file. A nice feature that I would recommend to the Sunflower folks would be the addition of a function key to access the directory just like the DIR command from DOS. This feature would enable you to check a particular filename prior to the attempt to open such and the possible loss of more valuable information. F2 is used to close this and all other files that may be open when you are done with your transfer. F3 is the save command, this will write the current record /sector /memory page to the disk file you specified using the F1 key. F4 is the QUIT function and will take you back to the previous options.

Conclusion

ZDUMP is a powerful and extremely useful utility to have in one's DOS tool kit. It has saved the day more times than I care to talk about, and has paid for itself many times over. I have to say too that it has enabled me to get an education of DOS that I might not have gotten otherwise. I've since learned how to reconstruct directory entries, or locate files when only pieces of the directory exist. I have even written a few utilities for purchased programs I use regularly. ZDUMP gave me the ability to look at the file structure and the organization of information they write to the disk. Now I have a couple of lines of BASIC code doing what I might have otherwise paid a bunch of dollars for from the publisher. Who said only foreign car parts have outrageous price tags?

SUNFLOWER SOFTWARE
13915 Midland Drive
Shawnee, Kansas 66216
(913) 631-1333

$29.95
128 K RAM One Disk (Two Recommended)

Setting Up Your System

Part 2

William M. Adney
P.O. Box 531655
Grand Prairie, TXm 75053

If you followed last month's suggestions and recommendations, your personal computer center is a well organized and comfortable place to work. You have the CPU and CRT connected properly and your disk library is ready for some serious work. Your system disks and data disks are well organized, and it's time to begin some of the last major set-ups for an easy to operate system.

Some Useful Utilities

One of the best advantages of HUG membership is the availability of excellent software at quite reasonable prices. The HUG software library contains all kinds of excellent software which includes spreadsheets, editors, communications programs, and special utilities. And aside from the HUG software, don't forget that HUG membership enables you to get 10–20% off the list price of Heath and Zenith computer hardware and software. That savings will usually pay for the HUG membership several times each year. Enough of the sales pitch for HUG . . . let's look at some very useful utilities.

The Disk Directory

One of the first programs that you should buy is a program that lists your disk directory in alphabetical order. Depending on your operating system and computer, this program is called DIR19, DIR100 or DIR150. This program is one of the best, since it allows you to list all files on your disk in order, and it provides a display showing the number of files on the disk, the amount of disk space used, and the amount of disk space remaining. Each screen shows up to 80 files, and if the disk contains more than that, you simply press RETURN to see the next "page". This program is a terrific improvement over the normal DIR command provided with most operating systems. This program also displays the so-called hidden files that cannot be listed with the DOS command. It provides most of the features of CHKDSK, except for the capability to repair disk directories and display memory usage. A highly recommended program.

For those of you who are not familiar with HUG software, I have only picked one program out of a number of programs that are usually provided on each disk. In my opinion, the value of that one program makes the purchase worthwhile; it makes your life much easier. Since just about any software is usually at least the HUG price of $20.00, you will receive an excellent value.

The second point to be made about HUG software is the fact that the source code is nearly always provided on the disk. In all cases discussed here, these are assembly language programs which provide an outstanding way to learn good programming skills. If you have an interest in assembly language programming, I strongly recommend this software as an outstanding learning tool, aside from the usefulness of the programs themselves. As you might suspect, the source code is provided so that you can make your own changes to the program, if you desire. Our resident HUG expert, Pat Swayne, has either participated in the programming or developed the entire program concept. One example of Pat's complete program development activities is demonstrated by the complete line of KEYMAP programs which will be discussed later.

Managing Your Files

If you have any trepidation about learning some of the DOS commands, like COPY or PIP, you will find that the HUG File Manager (HFM) is an extremely useful software package. Its primary claim to fame is that it displays all of the files on the disk (or MS-DOS subdirectory) and allows you to perform some easy file maintenance. Since the concept is relatively simple, I will take a moment to explain.

HFM presents a simple screen with a list of all files on the disk (or subdirectory). One file shows up in reverse video, which is a marker that can be moved with the up/down cursor keys. As an example, you can copy that file by pressing "C", and the program asks for the destination disk and/or subdirectory. But what if you want to copy a number of files? That's easy! HFM also allows you to "flag" any of the files (indicated by an asterisk on the screen). You then press "C", as before, and specify the destination. In a similar manner, you can erase a number of files which is terrific for disk file maintenance.

HFM also has a number of other features which depend on the operating system for which it was developed. As previously men-

tioned, the MS-DOS version allows you to work with subdirectories including the MKDIR (Make Directory) and RMDIR (Remove Directory) functions. In addition to the copy and erase functions, you can also usually perform the following functions: Flag, ID (for disk label), Newdisk (change disks/drives), Print, Quit (HFM), Sort (file names), Type, and Unsort (file names).

One of the more interesting aspects of this program is that it provides some nice features for the beginning and experienced computer user. If you are a beginner, you will easily be able to do a number of things which will make your work easier without the frustration of memorizing (or looking up) the appropriate DOS command. Experienced users will enjoy the ease of performing the tedious chore of disk file maintenance, since a few keystrokes will replace a number of command entries. In many ways, HFM reminds me of the public domain utility, SWEEP, which has many of the same functions. I have used SWEEP for a couple of years for disk maintenance, but I think that HFM is better. HFM's primary advantage is that you can see all of the files at one time . . . an extremely useful feature.

Whether you are a beginning or advanced user, you will find the features of HFM a welcome addition to your utility library. It is a highly recommended program, which I have found very useful.

But I Don't Like The Keystrokes!

One of the biggest pains is entering the same series of keystrokes over and over. Whether it happens to be "DIR B:" or WordStar's CTRL-KS CTRL-QP sequence for saving a file, it takes a lot of time. Programming can also be simplified in a similar manner, such as the heavily used PRINT command in BASIC. How can you save time by not having to enter all of those keystrokes? It's simple . . . get one of HUG's KEYMAP programs by Pat Swayne!

If you have not heard about ways to "remap" your keyboard, KEYMAP is one of the best and also, to the best of my knowledge, the least expensive. Although one of WordStar's biggest shortcomings, the strange control key sequences, has been somewhat overcome by the latest release, I am still not impressed. By using some of the techniques that Pat programmed in one of his latest KEYMAP versions (CP/M-85), I have set up my H-100 to generate 105 commands and literals by pressing the appropriate key(s). In short, the function keys on the main keyboard can generate 4 commands each, and the keypad keys generate 2 commands each. The HOME key is used as the "Alternate Select" key, which provides a way to attach multiple commands (or literals) to the same key.

KEYMAP is implemented by using the KEYCON program, which allows you to customize the function keys to your liking. If you simply want to use the program, several configured versions are also provided on the disk which include variations for WordStar, BASIC, and DOS commands. Since some programs do not "like" resident utilities, the UNMAP program provides a way to unload the KEYMAP program from memory.

This is a program that is probably for the more advanced users, particularly if you want to customize your function keys to a particular program or configuration. While it is not at all difficult to do, it requires some understanding of the software for which you are doing the configuration. In WordStar, for example, the CTRL-KS CTRL-QP sequence is a combination of two commands: save the file on disk and remain in the program (CTRL-KS) and return to the same position on the disk where you were editing (CTRL-QP). I have both of those commands "mapped" to one key so that I don't have to re-enter them all of the time. I also have a number of typesetting commands which are also available at the press of a key for writing books.

If you want to be able to reconfigure your function keys, this is an excellent program. It also will give you some insight as to how this is actually done, which is valuable just for general information. I have tested all of the KEYMAP programs and have found them to be very useful. Most of them allow a maximum of 20 characters input, which is more than enough for most uses. A highly recommended set of programs.

Setting Up Customized System

For the advanced user or someone who is setting up a system for a beginner, the HUG Menu System is a useful tool. It provides a way for you to develop customized menus which will lead the user through the tasks to be performed. You can program a number of menu "levels", including help screens, and all you need is an editor that produces ASCII files.

Note that I said ASCII files. That means that you can not use some word processors to generate those files. Microsoft Word embeds the style sheets into the document, which is one reason that it will not work. MultiMate generates documents in a similar way, although it does include a utility which allows you to generate ASCII files in a rather convoluted way. The easy way to tell if a file is in ASCII format is to use the TYPE command. If the file looks the same on the screen as it does on the word processor, then it is probably a straight ASCII file. By the way, PeachText 5000, WordStar, and WatchWord allow you to choose the mode, which is a distinct advantage if you do any programming. PeachText even carries that further, since it will automatically select a mode depending upon the file type. If you are editing an ASM file, PeachText assumes that you want the programming (ASCII file) mode. So much for that tangent. It seems that when I mention something like an ASCII file, I get a lot of letters asking what it is.

Back to the HUG Menu System. The menu is relatively easy to set up, but I don't recommend it for beginners. As a beginner, the most important thing for you to learn is the operating system. The HUG Menu System allows you to set up some really neat looking menus, complete with graphics. If you have a lot of imagination and some free time, you can do some really useful things with the menus. A recommended program for advanced users only.

Reading The Documentation

One of the most important things that you should do as a computer user is to read the documentation. Software Consultation tells me that about 80% of their calls are for information that is available in the manuals. One of the nice features of the manuals is that they are rather completely indexed with a detailed Table of Contents. If you are not sure how to use a command, read the manual BEFORE you attempt to try the command.

I knew of one user who partly read the manual on how to boot the system. Then he decided to try out a couple of the commands. For some unknown reason, he thought that he would try the ERASE *.* command. That is not all bad, except that he did not copy the distribution disk. The first time he tried the ERASE command, he got an error message. After reading the manual, he found that he could eliminate the error message by removing the "write protect" tab from the distribution disk. He was pleased to note that the ERASE command worked, but was rather surprised

The H89 *SPEED* Center

4MHz mod — $45

An easy to install plug-in module. No trace cutting or soldering. Speed may be toggled with software. Includes a replacement Z80A (4MHz). Software supports Heath CP/M 2.2.03 and 04 and CDR Systems BIOS. Call or write for support of other BIOS or HDOS. Specify disk format.

6MHz mod — $59

Similar to our 4MHz modification, but increases the CPU speed to 6MHz. Requires some soldering on the CPU circuit board. Includes a Z-80B (6MHz) CPU replacement. Some technical knowledge is recommended for installation. Call or write for more details. Specify disk format.

CDR Super RAM 89

Make your H89 a speed whiz with this board and the included RAM drive software. With our 6MHz mod and the RAM drive, the Heath BIOS assembles (using MAKEBIOS) in less than 1½ minutes! Note our discount prices!

Main board w/o RAM - specify disk format **$175**
Expansion board w/o RAM **$69**
Main board w/512K RAM **$239**
System w/MEGABYTE RAM **$369**
Clock option **$45**
SCSI option **$85**

CDR Controllers

Double density disk controllers. Includes board, instruction manuals, ROMs and software. Allows control of 5 and 8 inch drives. Includes BIOS source code and some useful utilities. Please include your CP/M s/n when ordering.

For the H89 FDC-88OH **$349**
For the H8 FDC-H8 ..... **$429**

REP3 - Automatic Key Repeat

Stop wasting time - make full use of the repeat function just like the Z100. Simple plug-in installation on your keyboard assembly. Hold any key down for half a second and the key begins repeating. Combine this with our 4MHz mod and make WordStar fly! Provision for a defeat switch.

Kit **$35**
Assembled **$45**

TIM2 - Real Time Clock

This circuit installs in the left hand expansion slots of the H89 or H89A. Can provide application programs with time and date information. Includes battery backup for continuous timekeeping when the computer is turned off. Program access to the clock is through an I/O port. The port address is user selectable by means of a jumper. Requires soldering 4 wires to the CPU board. A prepared ribbon cable is included.

Kit **$65**
Assembled **$75**
Software on disk - specify disk format **$10**

DATESTAMPER

Product of Plu*Perfect Systems. Provides automatic time and datestamping for CP/M 2.2 files. Works with many real time clocks, including our own TIM2 product.

CP/M - specify disk format **$45**

WSPATCH — $20

Adds H19/H89 function key patches to versions 3.0 or 3.3 of WordStar. Key functions similar to the PIE editor. Includes provision for redefining the keys by the user. Also includes a printer driver for the Epson MX80/FX80 printers.

NEWWORD - the better word processor from NewStar Software

Faster than WordStar, same commands as WordStar, MailMerge feature is built-in, supports more printers, has an undelete function and excellent documentation. We offer it at a discount and also include a function/cursor key patch for the H19/H89. Part of the bundled software is the WORD plus spelling checker from Oasis - normally a $150 item by itself!

CP/M - specify format **$85**
CP/M with the WORD plus **$135**
MSDOS (Includes WORD plus) **$195**

EMULATE

Allows the H89 to read/write to the following disk formats.

Actrix	Eagle II	Morrow MD	Superbrain Jr.
AMPRO	Epson QX-10	NCR DecMate 5	Televideo
Beehive Tpr	Fujitsu CP/M86	NEC PC-8001A	TRS80-1 CP/M
CDR	IBM CP/M86	Osborne 1	TRS80-3 CP/M
Cromemco	IMS 5000	Otrona	TRS80-4 CP/M
DEC VT180	Kaypro II	PMC MicroMate	Xerox 820
DEC Rainbow	Magnolia	Sanyo 1100	Zorba

Now includes 42 formats! Uses a modified version of CP/M 2.203 or .04 BIOS which is included with the program. Allows the use of virtual drives and reading of 40-track disks in an 80-track drive.

For H37 with Heath CP/M **$59**
Check for C.D.R. and Magnolia versions.

DISKETTES

High quality private label diskettes at an economy price.

5" SSDD 10 hard sector ..... **$16/box of ten**
5" DSDD ..... **$18/box of ten**
8" SSDD ..... **$26/box of ten**
8" DSDD ..... **$29/box of ten**

ANALYTICAL PRODUCTS **209/564-3687**

20663 Ave. 352 **Woodlake, CA 93286**

Specify Disk Format on Software

CALL OR WRITE FOR CATALOG PRICES SUBJECT TO CHANGE

Terms: Check or Money Order - VISA/MC - C.O.D.
Add $3 per order for shipping and handling
California residents add 6% tax

when he found that he had deleted all of the files from his only copy of the operating system on the distribution disk. All of the software available from Heath and Zenith provides detailed instructions on how to start up your system . . . be sure to follow them if you are not sure how to do it.

Although there are many complaints about documentation, it has become much better in the last few years. It is extremely important that you understand your system and application software, since you will probably be spending a lot of time with it. More than anything else, you should understand some of the basic commands in the operating system like FORMAT, CHKDSK, and so on. To that end, I will be writing a future column on "The DOS Commands You Must Know."

So much for software and documentation. It's time to take a look at a very important item that is frequently overlooked.

Care And Feeding Of Floppy Disks

The most important thing in your system is the floppy disk! Even though the hardware represents a significant financial investment, floppy disks also represent a large investment in both time and money. All software is distributed on disks. When you spend the time to write a letter on your word processor or prepare a budget on your spreadsheet, you have invested some time which is valuable in one way or another. Protect that investment by taking care of the disks that contain your programs and data.

The manuals provided by Heath and Zenith represent some of the finest documentation available for a microcomputer today. In every manual that I have seen, there is always a section on floppy disks, usually under the heading of "Beginning Concepts". Unfortunately, it has been my experience that most people overlook that section and go on to the "good stuff". So I will go through some of the basics here.

Always keep the disk in the paper envelope when it is not in the disk drive. Don't leave naked disks lying around. For some reason, liquids (like coffee) seem to be attracted to disks, and a naked disk invites a coffee spill not to mention dust, ashes, and dirty fingers. If you develop the early habit of removing a disk from the drive, placing it in the envelope, and storing it in your disk cabinet, you are not as likely to lose data or software because of an accident. Keeping the disks in the envelope also keeps your hands off of the read-write slot which is that small area where you can see the disk itself. NEVER touch it! The small amount of oil usually present on the skin is sometimes enough to cause the disk to be unreadable. It goes without saying that active disks are not intended to be used as coffee cup coasters, like I have seen in one office.

Another important note is that you should never use a ball point pen or pencil to update disk labels. That caution also includes the "fine point" felt tip pens. The pressure from writing is enough to distort the disk surface and cause permanent damage. From a technical point of view, you should never mark on a label which is on a disk. That sometimes proves impractical, so I keep a supply of the SOFT, wide point Flair pens for writing on disk labels. The soft point allows me to write on the disk with very little pressure.

It seems that a lot of people know that a magnet will destroy the data on any type of magnetic media like a disk or tape, but some do not know that a common telephone contains a very powerful magnet which is used to ring the bell. Obvious point . . . keep disks away from the telephone, as well as other appliances.

These are just some of the points that are well documented in the manuals. If you read and follow those instructions, you will obtain the maximum life from the disks and minimize the frustrations caused by lost data and software.

Ports And Cables

Setting up a printer or modem for your system can be a very trying task if you have equipment that was not purchased from Heath or Zenith. And to complicate matters, there are all kinds of names for ports, but they are normally of the serial or parallel type. But that is getting a little ahead of the game.

First of all, what is a port? A PORT is an access point in your computer which allows for data transfer. Data transfer can be incoming to your computer, like the old paper tape punches; you can transfer data from your computer to a printer or another computer; and you can have two-way data transfer with a modem.

Now you know the definition and characteristics of a port, but what is a port? The most obvious answer is that a port can be physically found in the form of those strange looking plugs on the back of your computer. But ports come in all shapes and sizes. For example, the disk drives are available at a special port on the disk controller. There is a port for the keyboard and a video port. I will not discuss all of those, since we really want to look at information on ports that you should know . . . in some cases you must know it to successfully connect a printer or modem to your computer.

For the most part, the plug on the back of a computer is usually known as a DB-25 connector. There is another type of connector which is used for connecting parallel devices, but we will talk about that later. Interestingly enough, the DB-25 can be used for both serial and parallel devices. Great, you say, but what is a serial device and what is a parallel device?

A SERIAL device or port transmits (or receives) data sequentially or serially. That is, the data is transmitted bit-by-bit in a series. Remember that a byte (e.g. character) contains 8 bits. Those bits are transmitted in order, one-by-one, through a serial port. Each character must be transmitted down the line in this fashion.

A PARALLEL device or port transmits (or receives) bits constituting data simultaneously or in parallel. This means that each of the 8 bits constituting a character are transmitted simultaneously (normally on separate lines) and flow down a circuit in parallel. Each character arrives at the printer at essentially the same time instead of waiting for each group of 8 bits like a serial device.

The Connectors

There are two common types of connectors. The 25 pin DB-25 connector is nearly a standard type of connector on the back of most computers. The H-100 has three DB-25 connectors — two for serial ports and a parallel port. The H-150 has two — a serial and a parallel port. It is interesting to note that the serial and parallel ports utilize the same connectors.

Parallel connections to printers are typically made with a 36 "pin" Centronics parallel connector. The parallel connector does not really have pins, but the idea is the same.

Parallel Devices

Printers are the most common of parallel devices. If you have a choice, it is usually much easier to connect a parallel device to a computer, since there are common connections. In general, the cable is wired on a one-to-one basis such that pin 1 goes to pin 1,

pin 2 goes to pin 2, and so on. It is not, unfortunately, quite that simple for serial devices.

Serial Devices

Why should you care about connecting a serial device when you currently have a parallel printer? The biggest reason is that modems use serial communications, so you should understand something about it before you get a modem. Since there is a standard defined for serial devices, called the RS-232 standard, we will take a look at that.

The RS-232 standard basically defines all equipment in one of two categories: DTE (Data Terminal Equipment) or DCE (Data Communications Equipment.) Modems and most computers are classified as DCE while terminals, printers, and most other peripherals are DTE. The important point about DCE/DTE equipment is that you must connect a DTE to a DCE. You cannot connect two like types together . . . they simply will not work and there is a possibility of damage to one or both.

The H-100 has both a DCE port (Serial Port A - J1) and a DTE port (Serial Port B - J2). The fact that the H-100 has both types of ports is one reason that the CONFIGUR program draws a picture of the back of the computer showing where to connect the device that you have added. The H-150 has a standard DTE port which can be used to connect a modem (DCE).

I have used all of the ports on my H-100. The DCE port is connected to my H-25 printer (DTE). The DTE port is connected to my modem (DCE). And the parallel port is connected to my DTC Style Writer printer. If I get any more equipment, I'll be in deep petunias.

Build Or Buy A Cable?

Trying to figure out which is what and how to connect them can be more than a little difficult. If you don't believe it, just look at the Heathkit catalog and you will see a number of different cables . . . the one I have lists 8 different ones of five different types. If you have any doubts, be sure to ask one of the people in your Heathkit store. I recommend that you buy a cable when you buy equipment to add to your system. My personal experience is that it's a waste of time to try to build your own, and you really don't save much money in doing it.

Power Protection

One of the banes of most electronic equipment, particularly computers, is what I call power glitches. Most microcomputers are powered by the standard wall socket which provides 110 volts AC (VAC) at 60 Hertz. A POWER GLITCH is anything which differs from that standard line voltage and frequency. If the line voltage drops significantly from that standard (usually called a brownout), you can normally see that on the CRT, since the display appears to get smaller. That can sometimes happen in your home when the central air conditioning or electric dryer is turned on. Then you get a "surge" which may go above the standard voltage. The problem can be worse in the business environment. My publisher's office, for example, is near an automotive repair shop. Their electric welder plays havoc with his computers and typesetter because of the drop in line voltage and the significant interference transmitted through the line from the welder. Even though these problems are more likely to be found in a business environment, you may also have them in your home. I have mentioned several times that the Superblock on my hard disk was apparently blown away by a 5 second power glitch. Everything on my hard disk was lost since the Superblock contains, among other things, the information related to the partition addresses. That's obviously a rare occurrence, but things like that can happen. What should you do?

I recommend that you add a surge protector to your system as a minimum. Heath has two forms of surge protectors called the Smart Outlet Box (or SOB for short): the kit GD-1295 and the assembled GDW-1495. They have a "smart outlet" which allows you to connect one device (e.g. a CRT) to a control socket. The control socket senses current flow in the device and turns on the power to the remaining sockets. I suggested the CRT in this case since the power switch is usually on the front of the cabinet. But what if your power problems are worse than a need for simple surge protection?

The Uninterruptible Power Supply (UPS)

Many mainframe computer centers use very complex UPS systems costing in the hundreds of thousands of dollars. And some even have special emergency generators which can be used during an extended power outage. Although a UPS is not required for many microcomputer systems, it can protect you against loss of data while you are working. Since I use my computer nearly every day for professional writing, a UPS is mandatory for me since the power in Grand Prairie seems to have more than a normal share of problems. During a recent weekday, my UPS automatically kicked in four times in one afternoon because of 2 second power glitches.

Although there are several UPS systems now in the Heathkit catalog, I needed one at a time when they did not sell one big enough to handle my system. My UPS is connected to my H-100 (with a hard disk, a disk drive, and various S-100 boards), my CRT (ZVM-122A), an external 5-inch drive, and two 8-inch drives. My ProModem is also connected to the UPS. All of that equipment takes a measured 2.8 amps of power with about a 0.1 amp increase when a floppy disk drive is activated. The basic power requirement is then:

power (watts) = voltage (110 volts) x current (2.8 amps)

That is 308 watts or 308 VA (volt-amps).

My next problem was to find a UPS that was big enough to handle my system. I also wanted one that had built-in surge protection, as well as EMI (electromagnetic interference) and RFI (radio frequency interference) filtering. And finally, the UPS had to have a switching time fast enough to handle any potential problems related to my hard disk. These are, by the way, key points that you should consider if you are planning on a UPS for your system. Although not a requirement, I also wanted one that had sufficient capacity to handle my printer or other additional equipment if needed.

The MINUTEMAN 500

After looking at the current market at that time, I decided that the MINUTEMAN 500 from Para Systems in Dallas was the best choice to fit my needs. It provided 500 watts of power with a fast switching time (less than 4 ms) with EMI/RFI filtering and surge protection. In a brownout condition, the UPS picks up the load when the voltage drops below 95 volts which is more than adequate to support my H-100 with all of the peripherals. It also provides sine wave output, which is very important for some computer systems. Since the MINUTEMAN 500 is rated to handle the very sensitive IBM PC XT, I was certain that it could handle my H-100 without any problem. The fact that Para Systems is a local

company is purely coincidental since I had predefined my needs and researched the market before I selected the UPS.

After using the MINUTEMAN 500 for a few months now, I am convinced it was one of the best investments that I have made for my system. It has worked perfectly, and I have had no power problems with my system even considering the rather odd power fluctuations that we have in Grand Prairie. The MINUTEMAN 500 is highly recommended if you need a UPS. I should also note that Para Systems has a 250 watt UPS, the MINUTEMAN 250, which has similar characteristics. A surge protector, the CLEAN POWER 1200, is also available which also features EMI/RFI line filtering.

The HUG Conference

I really enjoyed speaking at the HUG Conference on MS-DOS Directory Trees, I/O Redirection, and Command Pathing. The biggest announcement, in my opinion, was that Heath and Zenith will be supporting two PC emulator boards for the H-100. The Gemini board was announced nearly a year ago as I write this, and I guess that we will finally see it. In addition, UCI also announced a PC emulator for the H-100. That caused a lot of interest, but now for the good part.

With the kind support of Heath Company and HUG, your intrepid reporter will be evaluating both boards for the first time anywhere. My special thanks to Bob Ellerton, the HUG Manager, who has made arrangements for me to obtain evaluation copies of both boards. In short, keep reading REMark for a review of each one.

Based on the latest scheduling information, I will not say when the reviews will appear. As of this writing, I do not have either of the boards yet. However, my plan is that a separate article will be written about each emulator with a summary at the end of the second one. At this point, it looks like I will get the Gemini board first, but who knows? The articles will appear in the order that I receive the emulators. Regardless of what is planned for the next columns, I will drop the planned column in favor of reviewing the emulator. Although I already have some opinions about these emulators based on what I saw at the HUG Conference, I will refrain from any comments until I have given each one a fair evaluation on my own computer.

I had a chance to meet both Bill Johnson, President of Heath Company, and Joe Schulte, President of Veritechnology Electronics Corporation (VEC). Both are clearly outstanding men with a sense of humor, and I enjoyed meeting them. For those of you who may not know, VEC is the organization that owns and operates the Heath/Zenith Computers & Electronics Centers. It's difficult to get used to that name . . . maybe it can be abbreviated as H/Z CEC's.

Steve Robbins has a new version of WatchWord and announced a new Resident Speller. The Resident Speller is impressive in that it checks your spelling as you type on the computer. It can also run in the standalone mode for checking a file without WatchWord. More on that in a future column.

Next Month

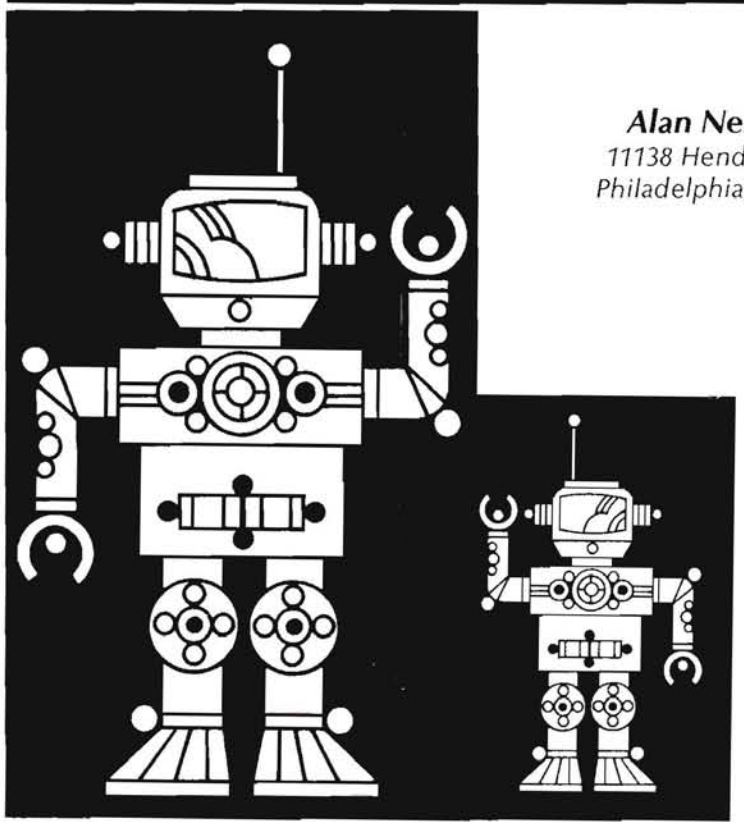
Unless I receive the Gemini and/or the UCI emulator, the next column will be about MS-DOS Directory Trees, I/O Redirection, and Command Pathing which I discussed at the HUG Conference.

By the way, the enhancement to the AUTOEXEC.BAT file will appear in the next column, since it is relating to the command pathing. If you have a particular subject of interest that you would like to know about, be sure and let me know. As always, please enclose a stamped, self-addressed envelope if you would like a personal reply.

Products Discussed

Smart Outlet Box	
Kit (GD-1295)	$ 49.95
Assembled (GDW-1495)	99.95
HUG Software	
Directory Program	
CP/M DIR19 (885-1226[-37])	$ 20.00
MS-DOS DIR100 (885-3008-37)	20.00
Z-150 DIR150 (885-3014-37)	20.00
HUG File Manager	
CP/M (885-1246[-37])	$ 20.00
H/Z-100/150 (885-3014-37)	20.00
KEYMAP	
CP/M-80 (885-1230[-37])	$ 20.00
CP/M-85 (885-1245-37)	20.00
CP/M-86 (885-5001-37)	20.00
Z-DOS (H-100) (885-3010-37)	20.00
Z-150 (885-6001-37)	20.00
HUG Menu System (H/Z-100/150)	
MS-DOS only (885-3020-37)	$ 20.00
Heath/Zenith Computer Centers	
Heath Company Parts Department	
Hilltop Road	
St. Joseph, MI 49085	
(616) 982-3571 (HUG Software only)	
(800) 253-7057 (Heath Catalog orders only)	
MINUTEMAN 500 UPS	$699.00
MINUTEMAN 250 UPS	435.00
CLEAN POWER 1200	49.95
Para Systems	
11425 Mathis St., Suite 404	
Dallas, TX 75234	
(214) 869-1688	

*



Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

Toward Fatherhood In The 21st Century

He was conceived in an electronics store in Philadelphia. At first, my wife didn't want any part of it, but my sad eyes and begging expression finally broke her down.

I realize it wasn't the most attractive location. A busy intersection of fast food restaurants and gasoline stations. But, it was something I had wanted to do for a long time and just didn't want to take no for an answer. After all, I worked for a living and could support it. Once it was trained, we could take it for rides, teach it to play ball, even help around the house.

I think she was afraid that it wouldn't work right, or that after the long period of labor we'd have to take it back to the store. Things could always be fixed, I said. We could even find a nice couple to adopt it once we got bored with it. I knew I could even sell it for a profit.

She still wasn't totally convinced. But, being the good wife she is, we took the first step toward creating a new being: our little robot.

We don't have any real (human) children, but being a computer and electronics fanatic, I had to experience that thrill so many demented scientists have shared on the late night movies. The scenario always seemed so trite:

Well-meaning scientist builds the robot to help serve humanity . . . robot takes on human characteristics . . . scientist loses control, but can't destroy his creation . . . hero kills robot after mad scientist dies in the hands of his creation.

Pretty silly stuff.

My robot was going to be purely a mechanical object, a computer with wheels and voice. Sort of an Apple that bites back. After all, this is the real world, not Earth Station 2000 and pointed-eared creatures. Even the costly sophisticated industrial robots are far from the galactic visions of Saturday matinees. Long tentacles armed with welding torches, not lasers, perform routine jobs once trusted to human hands. No voices or personality, and only trained for one specific task, these robots would bore even the dullest five-year-old.

While robots are being used extensively in industry, the home versions are even more experimental and educational. One little round "turtle" is always tethered to its host computer. Another simply follows a plastic path on the floor. I knew from the start that this collection of silicon parts could have no more personality than a vacuum cleaner.

As I started to unpack the thousand or so parts from my kit, I had some time to read the literature. The robot could sense sound and light, distance and movement. Three inflection levels of voice. A shoulder, wrist and hand. In a few months, this mass of chips and wire would perform mechanical marvels.

The first trimester of labor (actually just a month) made me question why I had even begun. Each night I was ill from lead fumes and I hardly had anything to show. Five circuit boards and a battery charger were no Buck Rogerian achievement. My den became littered with scraps of wire, soldier bits sticking to half the surfaces in the house. Yet, I certainly became the center of conversation. Some say I even had this warm glow whenever the topic of robots entered the conversation.

Even though we had plenty of time, my wife Barbara and I began planning for its arrival. We talked about names and fixing up a little room. Guard rails would be needed at each flight of steps and the house made safe from little prying claws and wheels. Of course, even though she shared my joy, Barbara continued her usual activities, leaving me the discomfort of burnt finger-tips and eye strain. I understood that robot rearing was a man's lot.

The situation improved during the second trimester. Most of the

MEDIA MASTER

FORMATS SUPPORTED	IBM PC	ZENITH Z-100	OSBORNE 1/EXEC	KAYPRO 2X/4/10	DEC RAINBOW	SANYO 555
Actrix (DSDD)	•	•			•		•
Actrix (SSDD)	•	•	•	•	•	•	•
Avatar TC10 (DSDD)	•	•			•		•
Casio FP 1000 (DSDD)	•	•			•		•
Chameleon CP/M-80	•	•			•		•
Columbia MPC CP/M-80	•	•			•		•
Cromemco CDOS (SSDD)	•	•	•	•	•	•	•
Cromemco CDOS (SSSD)			•	•	•	•	
Cromemco w/Int'l Term (DSDD)	•	•				•	•
Cromemco w/Int'l Term (SSDD)	•	•	•	•	•	•	•
DEC Rainbow CP/M-86/80						•	
DEC Rainbow MS-DOS						•	
DEC Mate II CP/M (SSDD)						•	
DEC VT180 (SSDD)	•	•	•	•	•	•	•
Davidge (DSDD)	•	•				•	•
Digilog (DSDD)	•	•				•	•
Epson Multifont (DSDD)	•	•				•	•
Epson QX-10 (DSDD)	•	•				•	•
Facit DTC (SSDD)						•	
Fujitsu Micro 16S (DSDD)	•	•				•	•
Groupé III CP/M (DSDD)	•	•				•	•
Heath w/Magnolia CP/M	•	•	•	•	•	•	•
Heath/Zenith Z-DOS (1.xx DSDD)	•	•		•		•	•
Heath/Zenith Z-DOS (1.xx SSDD)	•	•	•	•	•	•	•
Heath/Zenith Z-DOS (2.xx DSDD)	•	•		•		•	•
Heath/Zenith Z-DOS (2.xx SSDD)	•	•	•	•	•	•	•
Heath/Zenith Z-100 CP/M (DSDD)	•	•					•
Heath/Zenith Z-100 CP/M (SSDD)	•	•	•	•	•	•	•
Heath/Zenith Z90 40 track (1k block)	•	•	•	•	•	•	•
Heath/Zenith Z90 40 track (2k block)	•	•	•		•	•	•
IBM PC CP/M-86 (DSDD)	•	•				•	•
IBM PC CP/M-86 (SSDD)	•	•	•	•	•	•	•
IBM PC DOS (1.xx DSDD)	•	•		•		•	•
IBM PC DOS (1.xx SSDD)	•	•	•	•	•	•	•
IBM PC DOS (2.xx DSDD)	•	•		•		•	•
IBM PC DOS (2.xx SSDD)	•	•	•	•	•	•	•
IDEA Intelex (SSDD)	•	•	•		•	•	•
ISM CP/M (DSDD)	•	•				•	•
Ithaca (SSDD)						•	
Insight Dev IQ-120 (SSDD)	•	•	•		•	•	•
Kaypro 10 (DSDD)	•	•				•	•
Kaypro 2 (SSDD)	•	•	•	•	•	•	•
Kaypro 4 (DSDD) *	•	•				•	•
LNW-80 (SSDD)	•	•	•	•	•	•	•
Lobo MAX-80 (SSDD)	•	•	•	•	•	•	•
Lobo MAX-80 512 (SSDD)	•	•	•		•	•	•
Macal 9050 CP/M-80 (DSDD)	•	•				•	•
Morrow MD11 (DSDD)	•	•					•
Morrow MD2 (SSDD)	•	•	•	•	•	•	•
Morrow MD3 (DSDD)	•	•					•
Mac Sym-Cyn (SSDD)						•	
Monroe OC8820 (SSDD)						•	
NCR Decision Mate 5 (DSDD)	•	•				•	•
NEC PC-8001A (DSDD)	•	•				•	•
NEC PC-8001A (SSDD)	•	•	•	•	•	•	•
Olympia ETX II (SSDD)	•	•	•		•	•	•
Olympia EX100 (DSDD)	•	•				•	•
Osborne 1 (SSDD)	•	•	•	•	•	•	•
Osborne 1 (SSSD)			•	•	•	•	
Osborne 4 Vixen (DSDD)	•	•		•		•	•
Osborne Executive (SSDD)	•	•	•	•	•	•	•
Osborne Osmosis (SSDD)	•	•	•		•	•	•
Otrona (DSDD)	•	•				•	•
Philips 8000 (SSDD)							•
PMC Micromate (DSDD)	•	•				•	•
Reynolds & Reynolds (SSDD)	•	•	•		•	•	•
Sanyo CP/M (DSDD)	•	•				•	•
Superbrain (DSDD)	•	•				•	•
Superbrain Jr. (SSDD)	•	•	•		•	•	•
Systel II (SSDD)	•	•	•	•	•	•	•
Systel III (DSDD)	•	•				•	•
S-100 10 TRK (SSDD)			•		•	•	•
TI Professional CP/M-86 (DSDD)	•	•				•	•
TI Professional CP/M-86 (SSDD)	•	•	•	•	•	•	•
TRS-80 Model III FEC CP/M (SSDD)	•	•	•		•	•	•
TRS-80 Model III FEC T805 (SSDD)	•	•	•		•	•	•
TRS-80 Model III Hurr Labs (SSDD)	•	•	•		•	•	•
TRS-80 Model III Mem Merch (SSDD)	•	•	•	•	•	•	•
TRS-80 Model I Omikron CP/M (SSSD)			•	•	•	•	
TRS-80 Model IV CP/M Plus (SSDD)	•	•	•	•	•	•	•
TRS-80 Model IV MontezumaMicro (SSDD)	•	•	•		•	•	•
Teletek 10 track (SSDD)	•	•	•		•	•	•
Teletek 80 track (SSDD)						•	
Toshiba T100 (DSDD)	•	•				•	•
Televideo TurboDOS (DSDD)	•	•				•	•
Televideo 802 CP/M						•	•
Wang Mass-11 CP/M (DSDD)	•	•				•	•
Xerox 820 II (SSDD)	•	•	•	•	•	•	•
Xerox 820 II (SSSD)			•	•	•	•	
Zorba 10 track (DSDD) *	•	•				•	•

** Requires access to "foreign" computer*

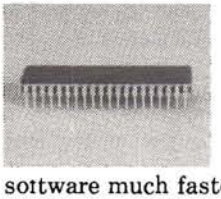
"MAKE YOUR Z-150 CP/M COMPATIBLE"

Intersecting Concepts Announces 3 Solutions To Solve Your Computer Incompatibility

But will it work on my computer? Yes! Finally, there are three easy ways to exchange information, transfer files, and run CP/M software on MS-DOS machines.

1. **MEDIA MASTER** is our direct disk-to-disk format conversion program. Already an accepted industry standard, this $39.95 program uses simple screen prompts that lets you read, write and format up to 70 different 5¼" diskettes from CP/M, MS-DOS and PC-DOS operating systems. So if you work on a IBM PCompatible at the office, but use a CP/M computer at home, now you can easily transfer files that would otherwise be "foreign" to your computer's operating system.

2. **MEDIA MASTER PLUS** goes one step further by converting 8-bit CP/M software to run on 16-bit MS-DOS and PC-DOS machines. This newly released $59.95 product combines our IBM-PC version of Media Master with ZP/EM, a powerful new emulation program. The results are amazing: CP/M programs using 8080 instructions *and* data can be transfered from popular computers like Osborne, Kaypro and Zenith to run on MS-DOS and PC-DOS machines!



3. **ACCELER-8/16** is also new and dramatically improves the performance of Media Master Plus by tailoring the CP/M emulation around a NEC V20 microchip. This chip simply replaces the 8088 processor in your MS-DOS computer. Once installed, it'll run your CP/M and MS-DOS software much faster. (Speed improvements are roughly 15% faster in MS-DOS and 350% faster in CP/M!) With ACCELER-8/16 your MS-DOS computer is now CP/M compatible for only $99.95!

The Choice Is Yours

So if you just need to exchange files between different computers using different operating systems, there's Media Master at $39.95.

If you're upgrading to an MS-DOS or PC-DOS machine or already have one but don't want to spend money buying "new" software programs you already own, there's Media Master Plus for $59.95.

And if the execution speed of your CP/M programs is important, there's ACCELER-8/16 for $99.95, which provides a simple hardware and software solution to making your MS-DOS computer CP/M compatible. All three solutions save you money by eliminating expensive modems and communications software.

To Order

To order Media Master, Media Master Plus, or ACCELER-8/16, call 800-824-7888, and ask for operator 251. For additional product and upgrade information contact:

Intersecting Concepts, Inc.,
4573 Heatherglen Ct., Moorpark, CA 93021
or call 805-529-5073.

INTERSECTING CONCEPTS

circuit boards were completed and a basic shape was beginning to take place. We could actually start to see something develop. Most of my time was spent reading books on robots and trying to find comfortable places to sit while I worked. When I didn't have much to carry, I could work anywhere. But, as the pounds started to build, I became limited to one floor of the house.

It was an exciting time, though. We could see its little sonar tubes and cute wheels, as yet far from functioning. Barbara prepared a nice closet for it, away from any obstacles or other harmful objects. We even told the exciting news to Adam, our five-year-old nephew, starting a barrage of "where do robots come from" questions.

The last period was perhaps the hardest, but most enjoyable. It had grown so large that I could work in only one room. My reading intensified, books and magazines covered the entire house. It was at its full weight and shape, just waiting for the final batteries to be installed and power applied.

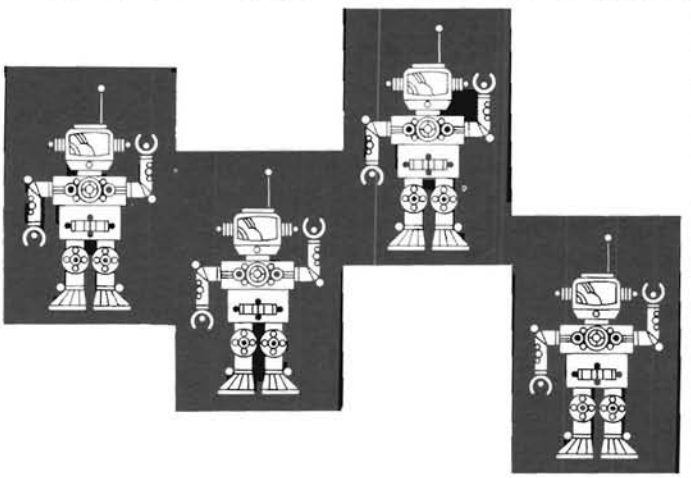
Finally, the day arrived. Barbara nervously recorded the date and time in our record book as I made the final connections. The house was prepared. We were ready to accept the challenge and responsibility. Its battery charger, the umbilical cord that would give it energy at the moment of birth, was plugged in carefully. Then the switch was thrown.

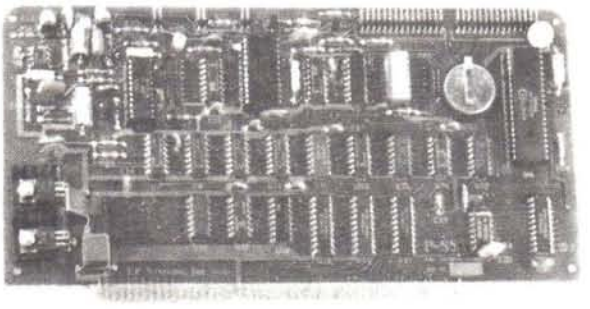
READY, it said. We watched and waited for any signs of defects, scrupulously examining every inch of the creation. Following the manuals, we tested each function: hearing, sight, speech, motor. Everything seemed in order.

But, this was just the beginning. Now would come the task of training Robbie, named by our nephew, in the ways of the world. It is an awesome responsibility, but worth it.

I guess it was after "birth" that I had realized what had truly happened. Like the fictional creators that had preceded me, I had begun to look upon this robot as something more than mechanical. Still knowing that there were hundreds exactly like it, it seemed to have a personality, a purpose, a meaning. Its three foot tall, 50 pound frame, is like a pet around the house: not human, but a creature none-the-less.

On Saturday afternoons, I still look at the old science fiction movies, but now with new insight. The future just doesn't seem so far away. And the simple messages, or warnings, about man's relationship with technology now make me wonder. *





P-SST!

GET $100 OFF THE P-SST!

How? Buy a D.E.L. Gemini (tm) PC emulator card from selected dealers, and get a FREE coupon good for $100 off the retail list price of $395

The new Gemini (tm) card from D.E.L. Professional systems will feature P-SST (tm) support to provide PC sound emulation. And when you're in Z-100 mode, you'll have all the great features of the P-SST—Speech, Sound, and music generation, on-board real time clock, and joystick compatible parallel ports!

How do you get the coupon? Just make sure your dealer has obtained his Gemini cards from a distributor who is including the coupon; Software Wizardry has made arrangements with most Gemini distributors

INTRODUCING P-SST-MAESTRO

A virtuoso performance!

P-SST-Maestro is a Music/Graphics Editor program for use with the Zenith Z-100 with P-SST multifunction card installed.

P-SST-Maestro provides you with a three-voice music editor that allows you to transcribe and compose musical scores with full color on-screen display in standard musical notation (notes, staffs, etc.) Once entered, included Music Utility programs allow automatic playback by simple commands. More than 50 pre-transcribed scores are included ready for use. P-SST-Maestro runs under ZDOS/MS-DOS 2 on the Z-100 with P-SST installed 192 K of main memory is required.

Also available: P-SST Development Libraries for advanced programming, and accessory cables for P-SST

P-SST Multifunction Card	*$395.00 list*
P-SST Maestro Music Software	*$ 59.95 list*
P-SST Development Libraries	*$ 49.95 list*

Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, MO 63301
(314) 946-1968
TWX 910-380-4822

Software Wizardry
THE MAGIC TOUCH

HUG Price List

The following HUG Price List contains a list of all products not included in the HUG Software Catalog. For a detailed abstract of these products, refer to the issue of REMark specified.

Part Number	Description of Product	Selling Price	Vol. Issue
HDOS HARDCOPY SOFTWARE			
885-1008	Volume I Documentation	9.00	
885-1013	Volume II Documentation	12.00	
885-1015	Volume III Documentation	9.00	
885-1037	Volume IV Documentation	12.00	8
885-1058	Volume V Documentation	12.00	
MISCELLANEOUS HDOS COLLECTIONS			
885-1032	Disk V H8/H9	18.00	8
885-1044-[37]	Disk VI H8/H89	18.00	
885-1064-[37]	Disk IX H8/H89 Disk	18.00	
885-1066-[37]	Disk X H8/H89	18.00	10
885-1069	Disk XIII Misc H8/H89	18.00	
GAMES			
HDOS			
885-1010	Adventure Disk H8/H89	10.00	4
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8
885-1031	Disk IV MUSIC H8 Only	20.00	25
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12
885-1068	Disk XII MBASIC Graphic Games	18.00	10
885-1088-[37]	Disk XVII MBASIC Graph. Games	20.00	14
885-1093-[37]	D&D H8/H89 Disk	20.00	16
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23
885-1113-[37]	HDOS Action Games H8/H89	20.00	23
885-1114	H8 Color Raiders & Goop	20.00	23
885-1124	HUGMAN & Movie Animation Pkg	20.00	41
885-1125	MAZEMADNESS	20.00	41
885-1130	Star Battle	20.00	47
885-1133-[37]	HDOS Games Collection I	20.00	59
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32
885-8022	HDOS SHAPES	16.00	45
885-8026	HDOS Space Drop	16.00	49
885-8032-[37]	HDOS Castle	20.00	59
CP/M			
885-1206-[37]	CP/M Games Disk	20.00	11
885-1209-[37]	CP/M MBASIC D&D	20.00	19
885-1211-[37]	CP/M Sea Battle	20.00	20
885-1220-[37]	CP/M Action Games	20.00	32
885-1222-[37]	CP/M Adventure	10.00	35
885-1227-[37]	CP/M Casino Games	20.00	38
885-1228-[37]	CP/M Fast Action Games	20.00	39
885-1236-[37]	CP/M Fun Disk I	20.00	55
885-1248-[37]	CP/M Fun Disk II	35.00	69
ZDOS			
885-3004-37	ZDOS ZBASIC Graphic Games	20.00	37
885-3009-37	ZDOS ZBASIC D&D	20.00	50
885-3011-37	ZDOS ZBASIC Games Disk	20.00	52
885-3017-37	ZDOS Contest Games Disk	25.00	58
UTILITIES			
HDOS			
885-1022-[37]	HUG Editor (ED) Disk H8/H89	20.00	20
885-1025	Runoff Disk H8/H89	35.00	
885-1060-[37]	Disk VII H8/H89	18.00	
885-1061	TMI Load H8 ONLY Disk	18.00	
885-1062-[37]	Disk VIII H8/H89 (2 Disks)	25.00	
885-1063	Floating Point Disk H8/H89	18.00	
885-1065	Fix Point Package H8/H89 Disk	18.00	10
885-1075	HDOS Support Package H8/H89	60.00	
885-1077	TXTCON/BASCON H8/H89	18.00	
885-1079-[37]	HDOS Page Editor	25.00	15
885-1080	EDITX H8/H19/H89 Disk	20.00	
885-1082	Programs for Printers H8/H89	20.00	
885-1083-[37]	Disk XVI Misc H8/H89	20.00	11
885-1089-[37]	Disk XVIII Misc H8/H89	20.00	20
885-1090-[37]	Disk XIX Utilities H8/H89	20.00	22
885-1092-[37]	Relocating Debug Tool H8/H89	30.00	14
885-1098	H8 Color Graphics ASM	20.00	19
885-1099	H8 Color Graphics Tiny PASCAL	20.00	19
885-1105	HDOS Device Drivers H8/H89	20.00	24
885-1116	HDOS Z80 Debugging Tool	20.00	27
885-1119-[37]	BHBASIC Support	20.00	29
885-1120-[37]	HDOS 'WHEW' Utilities	20.00	33
885-1121	HDOS Hard Sec Sup Pkg 2 Disks	30.00	37
885-1123	XMET Robot Cross Assembler	20.00	40
885-1126	HDOS Utilities by PS:	20.00	42
885-1127-[37]	HDOS Soft Sector Support Pkg	30.00	45
885-1128-[37]	HDOS DISKVIEW	16.00	46
885-1129-[37]	HDOS CVT Color Video Terminal	20.00	46
885-8001	SE (Screen Editor)	25.00	28
885-8003	BHTOMB	25.00	28
885-8004	UDUMP	35.00	28
885-8006	HDOS SUBMIT	20.00	31
885-8007	EZITRANS.	30.00	30
885-8015	HDOS TEXTSET Formatter	30.00	42
885-8017	HDOS Programmers Helper	16.00	42
885-8024	HDOS BHBASIC Utilities Disk	16.00	46
CP/M			
885-1210-[37]	CP/M ED (same as 885-1022)	20.00	20
885-1212-[37]	CP/M Utilities H8/H89	20.00	21
885-1213-[37]	CP/M Disk Utilities H8/H89	20.00	22
885-1217-[37]	HUG Disk Duplication Utilities	20.00	26
885-1223-[37]	HRUN HDOS Emulator 3 Disks	40.00	37
885-1225-[37]	CP/M Disk Dump & Edit Utility	30.00	40
885-1226-[37]	CP/M Utilities by PS:	20.00	40
885-1229-[37]	XMET Robot Cross Assembler	20.00	40
885-1230-[37]	CP/M Function Key Mapper	20.00	42
885-1231-[37]	Cross Ref Utilities for MBASIC	20.00	43
885-1232-[37]	CP/M Color Video Terminal	20.00	46
885-1235-37	CP/M COPYDOS	20.00	54
885-1237-[37]	CP/M Utilities	20.00	55
885-1245-37	CP/M-85 KEYMAP	20.00	63
885-1246[-37]	CP/M HUG File Manager & Utilities	20.00	64
885-1247-37	CP/M-85 HUG Bkgrd Print Spooler	20.00	67
885-5001-37	CP/M-86 KEYMAP	20.00	51
885-5002-37	CP/M-86 HUG Editor	20.00	52
885-5003-37	CP/M-86 Utilities by PS.	20.00	54
885-5008-37	CP/M 8080 To 8088 Trans & HFM	20.00	64
885-5009-37	CP/M-86 HUG Bkgrd Print Spool	20.00	66
885-8018-[37]	CP/M Fast Eddy & Big Eddy	20.00	43
885-8019-[37]	DOCUMAT and DOCULIST	20.00	43
885-8025-37	CP/M-85/86 Fast Eddy	20.00	49
ZDOS			
885-3005-37	ZDOS Etchdump	20.00	39
885-3007-37	ZDOS CP/EMulator	20.00	47
885-3008-37	ZDOS Utilities	20.00	47
885-3010-37	ZDOS Keymap	20.00	51
885-3022-37	ZDOS/MSDOS Useful Programs I	30.00	63
885-3023-37	ZDOS/MSDOS EZPLOT	20.00	63
885-3026-37	MSDOS SMALL C Compiler	25.00	65
885-3030-37	ZDOS/MSDOS Z-100 PC Emulator	40.00	68
885-3031-37	ZDOS/MSDOS Graphics	20.00	69
885-3032-37	MSDOS Halley's Comet Locater	20.00	70
885-8029-37	ZDOS Fast Eddy	20.00	53
885-8035-37	MSDOS DOCUMAT and DOCULIST	20.00	70
H/Z100 ZDOS/MSOOS - H/Z150 PC MSDOS			
885-3012-37§§	ZDOS HUG Editor	20.00	52
885-3014-37§§	ZDOS/MSDOS Utilities II	20.00	54
885-3016-37§	ZDOS/MSDOS Adventure	10.00	57
885-3020-37§	MSDOS HUG Menu System	20.00	62
885-3021-37§§	ZDOS/MSDOS Cardcat	20.00	63
885-3024-37§	ZDOS/MSDOS 8080 To 8088 Trans	20.00	64
885-3025-37§§	ZDOS/MSDOS Misc. Utilities	20.00	64
885-3029-37§§	ZDOS/MSDOS HUG Bg. Print Spool	20.00	66

§ All program files run on both
§§ Program files run partially on both

Part Number	Description of Product	Selling Price	Vol. Issue
PC/IBM COMPATIBLE			
885-6001-37	MSDOS Keymapper	20.00	59
885-6002-37	CP/EMulator II & ZEMulator	20.00	59
885-6003-37	MSDOS EZPLOT	20.00	65
885-6004-37	MSDOS CheapCalc	20.00	67
885-6005-37	MSDOS Skyviews	20.00	67
885-6006-37	MSDOS Cardcat	20.00	69
885-6007-37	MSDOS DND (Dung. & Dragons)	20.00	70
885-8033-37	MSDOS Fast Edit	20.00	62
885-8037-37	MSDOS Grade	20.00	70
PROGRAMMING LANGUAGES			
HDOS			
885-1038-[37]	Wise on Disk H8/H89	18.00	
885-1042-[37]	PILOT on Disk H8/H89	19.00	
885-1059	FOCAL-8 H8/H89 Disk	25.00	13
885-1078-[37]	HDOS Z80 Assembler	25.00	21
885-1085	PILOT Documentation	9.00	
885-1086-[37]	Tiny HDOS PASCAL H8/H89	20.00	13
885-1094	HDOS Fig-Forth H8/H89	40.00	18
885-1132-[37]	HDOS Tiny BASIC Compiler	25.00	59
885-1134	HDOS SMALL-C Compiler	30.00	63
CP/M			
885-1208-[37]	CP/M Fig-Forth H8/H89 2 Disks	40.00	18
885-1215-[37]	CP/M BASIC-E	20.00	26
BUSINESS, FINANCE AND EDUCATION			
HDOS			
885-1047	Stocks H8/H89 Disk	18.00	
885-1048	Personal Account H8/H89 Disk	18.00	
885-1049	Income Tax Records H8/H89 Disk	18.00	
885-1055-[37]	MBASIC Inventory Disk H8/H89	30.00	
885-1056	MBASIC Mail List	30.00	
885-1070	Disk XIV Home Fin H8/H89	18.00	
885-1071-[37]	MBASIC SmBusPk H8/H19/H89	75.00	17
885-1091-[37]	Grade/Score Keeping H8/H89	30.00	14
885-1097-[37]	MBASIC Quiz Disk H8/H89	20.00	18
885-1118-[37]	MBASIC Payroll	60.00	30
885-1131-[37]	HDOS CheapCalc	20.00	47
885-8010	HDOS Checkoff	25.00	32
885-8021	HDOS Student's Statistics Pkg	20.00	44
885-8027	HDOS SciCalc	20.00	50
CP/M			
885-1218-[37]	CP/M MBASIC Payroll	60.00	31
885-1233-[37]	CP/M CheapCalc	20.00	47
885-1239-[37]	Spread Sht. Contest Disk I	20.00	
885-1240-[37]	Spread Sht. Contest Disk II	20.00	
885-1241-[37]	Spread Sht. Contest Disk III	20.00	
885-1242-[37]	Spread Sht. Contest Disk IV	20.00	
885-1243-[37]	Spread Sht. Contest Disk V	20.00	
885-1244-[37]	Spread Sht. Contest Disk VI	20.00	
885-8011-[37]	CP/M Checkoff	25.00	32
885-8036-[37]	CP/M Grade	20.00	70
ZDOS			
885-3006-37	ZDOS CheapCalc	20.00	47
885-3013-37	ZDOS Checkbook Manager	20.00	54
885-3018-37	ZDOS Contest Spreadsheet Disk	25.00	58
885-8028-37	ZDOS SciCalc	20.00	50
885-8030-37	ZDOS Mathflash	20.00	55

Continued on Page 83

HUG NEW PRODUCTS

HUG P/N 885-3033-37 MS-DOS
HUGMCP $40.00

Introduction: HUGMCP, or HUG Modem Communications Package, is a program that allows your computer to communicate with another computer, either by way of direct RS-232 interconnection or by way of a modem. Uploading and downloading data is also supported.

In addition to this software, "The MicroNet Connection" package is being included at no extra charge. This package provides the user with a User IDentification number to access the CompuServe timeshare system, as well as the 'secret' password which will allow the user immediate access to CompuServe, as well as the National HUG Bulletin Board (HUG SIG). Some limited documentation is also included to get the user started. For more complete documentation on the HUG Bulletin Board, P/N 885-4700, the HUG Bulletin Board Handbook is recommended.

Requirements: HUGMCP will work on ANY H/Z-100 or H/Z-100 PC computer with one disk drive. The minimum amount of memory is required. This package requires the MS-DOS operating system (version 2.0 or greater) and MASM (version 2.0 or greater), if you wish to reassemble the source code. Finally, some sort of modem will be needed to access timeshare systems via phone.

The following files are included on the HUG P/N 885-3033-37 MS-DOS HUGMCP disk.

HUGMCP.EXE — Unconfigured executable file
HUGMCP.ASM — Source code
DEFMS.ASM — Definition file required for source reassembly
FIRSTIME.NET — Accessing CompuServe the first time info.
README.DOC — Basic information and disclaimer

Program Author: Jim Buszkiewicz, HUG

Program Content: HUGMCP is a modem communications package, written in assembly language under the MS-DOS operating system. Due to its large input buffer and interrupt driven modem port, HUGMCP is able to communicate at all standard baud rates with NO handshaking! This capability enables you to communicate with the ID-4801 EPROM Programmer at 9600 baud; a capability not found in other modem packages running under MS-DOS! Other features have been added to make this communications package compatible with future Heath/Zenith products.

HUGMCP is capable of uploading and downloading files using the standard XMODEM file transfer protocol. The program automatically switches between CHECKSUM and CRC error testing depending on which method the host is capable of. Data files can also be transferred using standard XON/XOFF (Control-S, Control-Q) protocol, and response to that protocol can be disabled at the user's option. Data can also be captured in HUGMCP's storage buffer. This buffer can be opened and closed with a single keystroke. This storage buffer, by the way, is as large as the amount of free memory your computer has at the time of execution. Typically, with a fully loaded H/Z-100, you could have a buffer greater than 600k!!

Auto log onto CompuServe is also supported by HUGMCP. By entering your ID number and password, HUGMCP will automatically log you into CIS with two keystrokes. Fourteen additional 32 character messages can be stored by HUGMCP and transmitted with two keystrokes. These short messages may contain any printable character, and is terminated with a carriage return (CR). These messages can be used for initializing modems, auto dialing, etc. In addition, one of these messages can be flagged to be immediately sent whenever the program is executed. This feature is quite useful for modems that need to be initialized upon power up.

One of the options available in HUGMCP is the 'setup' option. This allows the user to configure the software to 'his' computer system, as well as the host he plans on communicating with. Under this option, the user is allowed to modify the word length, word parity, baud rate, the auto-messages, some miscellaneous functions, and save the modifications on disk, if so desired. If the user has an H/Z-100 PC system, he is also given the option of using either COM1 or COM2 for his modem port. Under the setup option, you can also view the present settings of each modifiable option.

Although an extensive users' manual comes with this software, 'help' screens are available whenever applicable, making this software very 'user-friendly', and very easy to 'setup'.

HUGMCP maintains a concise help list, the open/closed status of the storage buffer, and the 'communications' (COM) status on the 25th line of the screen

The "MicroNet Connection" package contains the CompuServe User ID and secret password and are sealed in an enclosed envelope. The user is responsible for filling out and mailing the "Service Continuation/Request and Agreement" form to CompuServe.

CompuServe is a large timeshare database system that has many areas of service, information, interest and fun. The HUG Special

Interest Group (SIG) or Bulletin Board (BB) is a part of the CompuServe system. The member can leave, retrieve, search, scan, and reply to messages on the BB. In addition, the HUGBB has a large database of programs on CompuServe, of which the HUGBB member can download from the host. The HUGBB also has the facility for the member to upload files to the system for others to download.

To access CompuServe, the user must have a telephone number that links to CompuServe. There are direct numbers, TYMNET and TELENET, that access CompuServe. TYMNET and TELENET are two telephone services that link to remote systems. For their services, they have a surcharge per hour over the cost of CompuServe.

Note: To find out if you have a telephone link in your area, call the CompuServe Customer Service Toll Free number (800) 848-8990 or (614) 457-8650. For general information about CompuServe call (800) 848-8199.

There is documentation included with the package that shows step-by-step what the user will see the first time on CompuServe (the host computer). This file could be studied before going on CompuServe to help in understanding what the host timeshare system is doing. The sample run will show how to get to the HUG BB, as well as some other options, which are significant to HUG members.

Documentation about the system is available from CompuServe for an additional charge, and more information regarding this documentation can be found in the file: FIRSTIME.NET (in this product). More information can also be obtained from CompuServe directly, while on the system.

Special Note: CompuServe charges are around $6.00 an hour for regular hours and open areas. (The rates are subject to change.) Parts of CompuServe have additional charges. Any member of the HUG Bulletin Board (or SIG) receives a $.50 an hour discount for the time spent while on the HUG Bulletin Board.

Comments: This package does it all, quickly and efficiently, without a lot of unnecessary frills. It will also introduce a user to the timeshare system of CompuServe.

TABLE C Rating: (10), (5), (3), (1), (0)

885-3029-37 MS-DOS
HUG Background Print Spooler $20.00
Update

Introduction: The HUG Background Print Spooler (HBPS) is a program that allows you to assign some of your computer's memory (any amount from 4k to 512k) to be used as a print buffer. With HBPS loaded, when a program sends text to the printer (via normal DOS functions), it does not go directly to the printer, but into the HBPS buffer. HBPS then takes the characters from the buffer and sends them to your printer. Since HBPS can accept characters into its buffer more rapidly than the printer can accept characters, your computer can complete a print operation faster than without HBPS, and go on to other tasks while HBPS is printing characters from its buffer. HBPS gives you the advantages of a hardware print buffer without the hardware.

Note: This release of HBPS is an update to the original release. If you own the original, you can receive the update by returning your original disk with $5.00 check or money order payable to HUG, to Nancy Strunk, Heath/Zenith Users' Group, Hilltop Road, St. Joseph, MI 49085. This release provides faster printer operation with less degradation to system performance, handles files larger than the buffer correctly, and the PC version no longer requires that programs be patched in order to take advantage of spooling. The Z-100 version is now compatible with ZPC, and can be used while ZPC is in the PC mode.

Requirements: HBPS requires an H/Z-100 series, or H/Z-100 PC series (H/Z-150, etc.) computer, at least 192k of system RAM, and the MS-DOS or Z-DOS operating system. If Z-DOS is used, it must be the latest release. The maximum amount of memory assignable to HBPS with MS-DOS version 1 or Z-DOS is 64k (512k with MS-DOS version 2 or higher).

This disk contains the following files:

README	.DOC	SCRNCLK	.WHT
HBPS	.Z1	SCRNCLK	.RED
HBPS	.Z2	SCRNCLK	.GRN
HBPS	.PC1	CLK	.COM
HBPS	.PC2	HBPSZ	.ASM

TABLE C
Product Rating

10 - Very Good
9 - Good
8 - Average

Rating values 8-10 are based on the ease of use, the programming technique used, and the efficiency of the product.

7 - Has hardware limitations (memory, disk storage, etc.)
6 - Requires special programming technique
5 - Requires additional or special hardware
4 - Requires a printer
3 - Uses the Special Function Keys (f1,f2,f3,etc.)
2 - Program runs in *Real Time
1 - Single-keystroke input
0 - Uses the H19 (H/Z89) escape codes (graphics, reverse video)

Real Time — a program that does not require interactivity with the user. This term usually refers to games that continue to execute with or without the input of the player, e.g. p/n 885-1103 or 885-1211[-37] SEA BATTLE.

ORDERING INFORMATION

For Visa and MasterCard phone orders; telephone Heath Company Parts Department at (616) 982-3571. Have the part number(s), descriptions, and quantity ready for quick processing. By mail; send order, plus 10% postage and handling ($1.00 minimum charge, up to a maximum of $5.00. UPS is $1.75 minimum -- no maximum on UPS. UPS Blue Label is $4.00 minimum.), to Heath Company Parts Department, Hilltop Road, St. Joseph, MI 49085. Visa and MasterCard require minimum $10.00 order.

Any questions or problems regarding HUG software or REMark magazine should be directed to HUG at (616) 982-3463. REMEMBER-Heath Company Parts Department is NOT capable of answering questions regarding software or REMark.

NOTE

The [-37] means the product is available in hard-sector or soft-sector. Remember, when ordering the soft-sectored format, you must include the "-37" after the part number; e.g. 885-1223-37.

HBPS	.DOC	HBPSPC	.ASM
SETSP	.COM	SETSP	.ASM
PRZ	.COM	PR	.ASM
PRPC	.COM	BASPATCH	.ASM
BASPATCH	.COM	WSPATCH	.ASM
WSPATCH	.COM	WRDPATCH	.ASM
WRDPATCH	.COM	SCRNCLKZ	.ASM
SCRNCLKZ	.WHT	CLKZ	.ASM
SCRNCLKZ	.RED	SCRNCLK	.ASM
SCRNCLKZ	.GRN	CLK	.ASM
CLKZ	.COM		

Program Author: All programs are by P. Swayne, HUG.

HBPS.Z1, HBPS.Z2, HBPS.PC1, HBPS.PC2 — These are versions of HBPS for use on different equipment, and under different situations. Instructions for selecting a version, and for using HBPS are in the file HBPS.DOC.

Note: Some programs are slow in sending characters to a printer. If your printer is faster than a particular program, using HBPS or any other print buffering method will not result in a time savings. However, if a program prints slowly because of operating system overhead, such as Z-100 WordStar when used with MS-DOS version 2, there may be a time savings when HBPS is used. The slower your printer is, the more time will be saved by using HBPS.

HBPS.DOC — Instructions for using HBPS.

SETSP.COM — This is a program for controlling HBPS once it is loaded. It allows you to empty the HBPS buffer to suspend printing, or to disable HBPS so that a program's printer output goes directly to the printer.

PRZ.COM, PRPC.COM — This is a program that allows you to rapidly copy disk files to the HBPS print buffer. On a Z-100, the system command processor does not output to the printer via the normal DOS "channel" when you use the COPY command, as in

```
A>COPY FILENAME.EXT PRN
```

You must use PPZ instead in order to take advantage of HBPS when copying a file directly to the printer. On a Z-100 PC, the COPY operation WILL go through the spooler, but PRPC can perform the operation more quickly by copying the file directly to the spooler, avoiding DOS overhead.

BASPATCH.COM — The BASICA program (GW-BASIC) does not go through the normal DOS "channel" when it sends characters to the printer, so it bypasses HBPS. This program can patch BASICA so that the LLIST and LPRINT commands will output via the DOS. It does not affect other forms of printing, such as printing to "PRN" as a device. BASPATCH can patch all versions of BASICA sold by Heath/Zenith. **Note:** With this release of HBPS, the PC versions of GW-BASIC do not have to be patched.

Note: WordStar and Microsoft Word do not have to be patched to work with this release of HBPS on a Z-100-PC. However, if you use the products on a Z-100 under ZPC and the Z-100 version of the spooler, the patches are required before spooling of WordStar or Word output will take place.

WSPATCH.COM — This program patches the Z-100 PC (Z-150, etc.) version of WordStar so that its printer output will work with HBPS. **Note:** The Z-100 version of WordStar does not have to be patched.

WRDPATCH.COM — This program patches the Z-100 PC version of Microsoft Word so that its printer output will work with HBPS.

SCRNCLKZ.WHT, SCRNCLKZ.RED, SCRNCLKZ.GRN — This is a modified version of the SCRNCLK program for the Z-100 originally published in REMark and released on HUG disk 885-3014-37. The original version is slightly incompatible with HBPS in that the display cannot be turned off if it is loaded before HBPS. This version can be loaded before or after HBPS. When loaded, it causes a digital clock display to appear in the upper right corner of your screen. The file name extension determines the color of the clock display.

CLKZ.COM — This program is used to control the Z-100 SCRNCLK program. It can be used to turn the clock display off or on.

SCRNCLK.WHT, SCRNCLK.RED, SCRNCLK.GRN — This is a modified version of the SCRNCLK program for the Z-100 PC (H/Z-150, etc.) originally published in REMark. It is the same program that was released on HUG disk 885-3014-37. The original REMark version is slightly incompatible with HBPS in that the display cannot be turned off if it is loaded before HBPS. This version can be loaded before or after HBPS. The extension determines the color of the clock display.

CLK.COM — This program is used to control the Z-100 PC SCRNCLK program.

HBPSZ.ASM, HBPSPC.ASM SETSP.ASM, PRZ.ASM, PRPC.ASM, BASPATCH.ASM, WSPATCH.ASM, WRDPATCH.ASM, SCRNCLKZ.ASM, CLKZ.ASM, SCRNCLK.ASM, CLK.ASM — These are the assembly language source files for the above programs.

TABLE C Rating: (2), (4), (10) *

Strictly Beginners

Bob Ellerton, Manager
Heath Users' Group

A Review Of The Heathkit

HF-148 Personal Computer

The HF-148 PC is the Heathkit version of the ZF-148 PC produced by Zenith Data Systems. If you would enjoy a PC computer, but you can't justify the expense, this is one project deserving a second look. Besides being an extremely powerful and flexible computing tool, this model is a low-cost entry to IBM compatibility. And, talk about easy to build! The HF-148 reminds me of some of the projects mentioned in TV toy ads for the kids with the following quote: "Some assembly required."

To begin, let me summarize the basic assembly steps:

( ) Plug one chip into the assembled keyboard circuitry.

( ) Install keyboard circuitry into housing with six screws.

( ) Install keyboard housing bottom with four screws.

( ) Install power supply assembly into cabinet with two screws.

( ) Install CPU card into cabinet with one screw.

( ) Install rear panel onto I/O board with six hex nuts.

( ) Install I/O board into cabinet with two screws.

( ) Install drive assembly on drive chassis.

( ) Install the drive shield.

( ) Install drive chassis into cabinet.

( ) Connect cables.

( ) Done - test computer!

If this brief list sounds too simple for the "real" kit builder out there, it probably is. However, the steps outlined above represent a savings of exactly $500.00 over the assembled model! If you can afford to spend between 2-4 hours of your time, you will have a great little machine to show for the little effort you put into it. In dollars and cents, that breaks down to about $125 an hour! Not bad!

Let's review the HF-148 for a minute. This little version of a PC operates at either 4.77 or 8mHz and is switchable from the front panel. The CPU comes standard with 256K of main memory and can be expanded to a full 768K (640K supported). The CPU board also has provisions for the 8087 Numeric Co-processor chip. The HF-148 is equipped with one parallel and one serial I/O port.

As I mentioned earlier, this is a small version of the PC compatibles offered by Heath and Zenith. The major drawback to the HF-148 is the lack of expansion capabilities. You can add one accessory card to the unit with the purchase of the ZA-141-1 expansion accessory board (not mentioned in the kit manual). We will be covering the addition of this accessory in a future issue of REMark.

The HF-148 is truly a low-cost powerful computer limited only by the inability to expand the system with multiple add-on cards. However, if you are looking for a fast little machine for just about all routine tasks at great savings, you've come to the right place. With a few hours of your time, you can get the HF-148 going on all those popular IBM software packages.

For the most part, the pictures accompanying this article will walk you through the easy assembly steps. If you own a screwdriver along with some basic tools and you have never attempted a kit before, the HF-148 would be a good place to start. Look for future issues of REMark describing the addition of the expansion accessory.

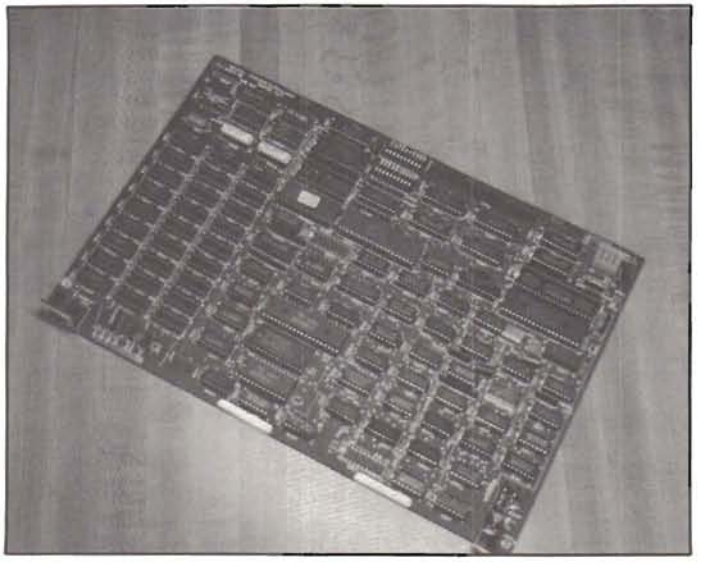
The HF-148 sells for $999.00! If you are shy about the products advertised with the familiar "Some assembly required," or if you are always looking for a product with the batteries included, then the HF-148 is probably not your cup of tea. However, you can get one assembled (ZF-148-41) for $1,499.00. Believe me, you can save a lot of bucks on this easy to assemble power-house with very little effort.

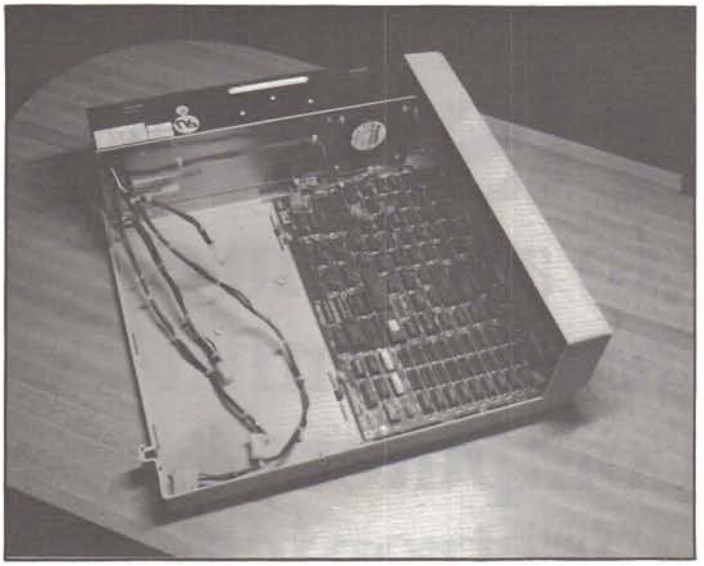
Assembled keyboard ready for installation in keyboard cabinet shell.

Finished HF-148-41 keyboard ready for action.

Pre-assembled supply ready to be installed in HF-148-41 cabinet.

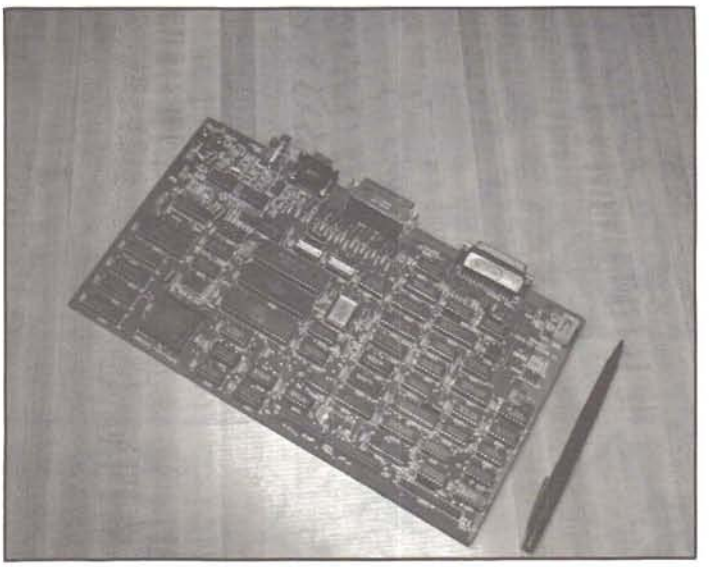
Pre-assembled CPU board ready for installation.

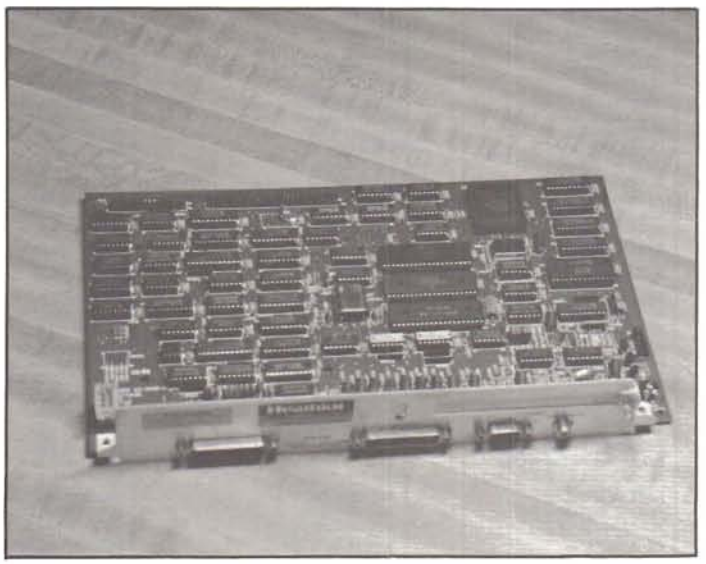




HF-148-41 cabinet with both the supply and CPU board installed.

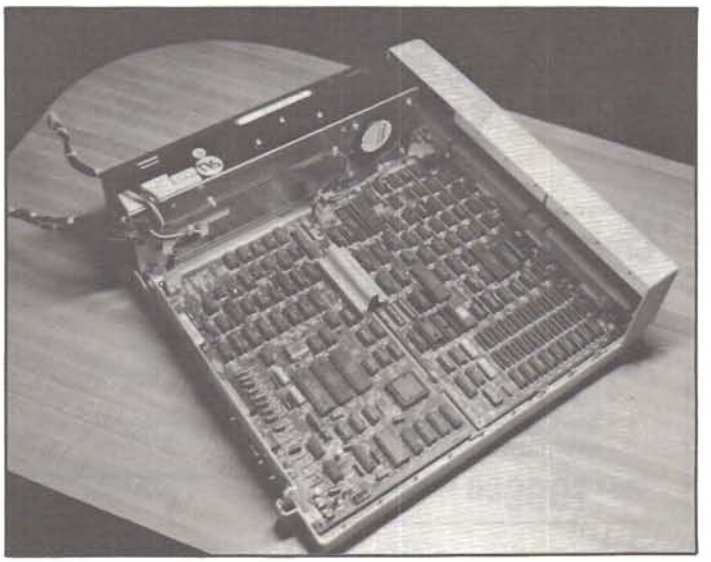
Pre-assembled I/O card — note the rear panel connectors.



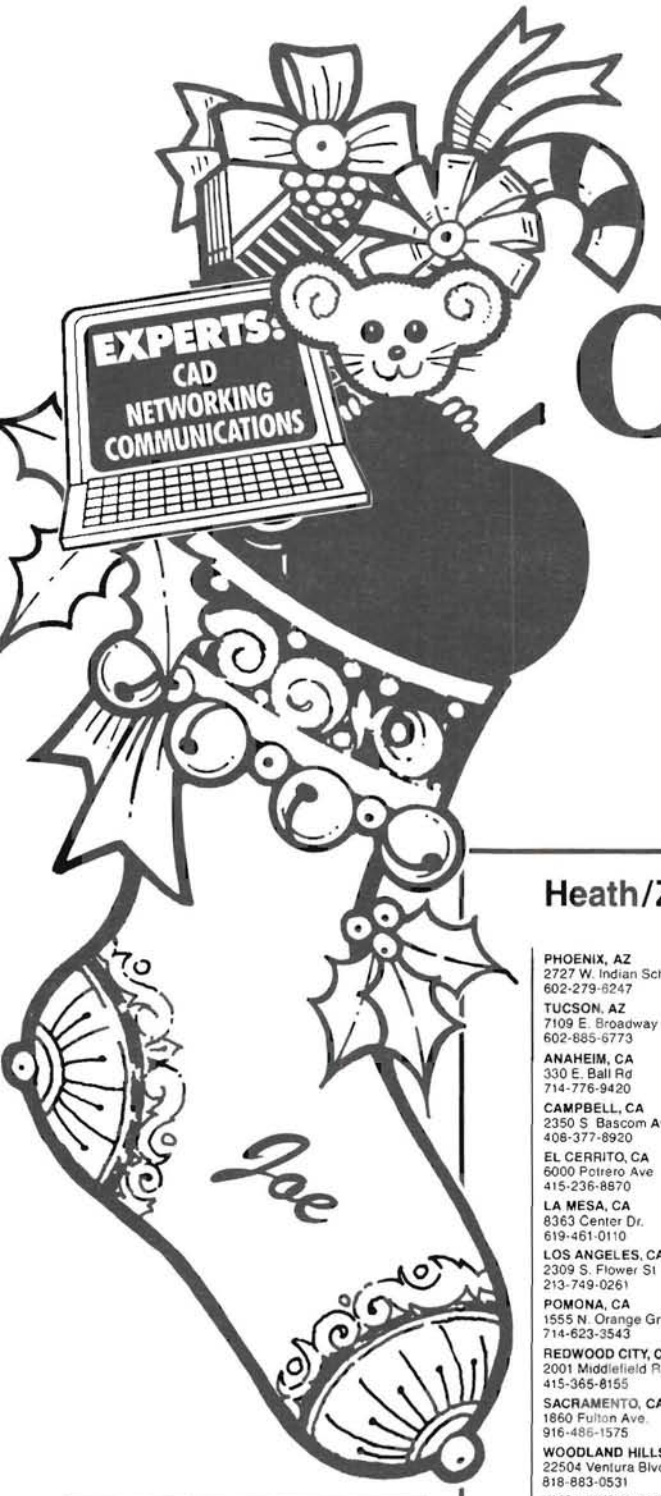


Rear panel plate installed on the I/O card.

Only the drive(s) need to be installed to complete the assembly of the HF-148. Extra memory and/or the 8087 should be installed before the drive(s).



*



and a Merry Christmas to all from Heath/Zenith Computers & Electronics

Heath/Zenith Computers & Electronics Centers In The U.S.

PHOENIX, AZ
2727 W. Indian School Rd.
602-279-6247

TUCSON, AZ
7109 E. Broadway
602-885-6773

ANAHEIM, CA
330 E. Ball Rd
714-776-9420

CAMPBELL, CA
2350 S. Bascom Ave
408-377-8920

EL CERRITO, CA
6000 Potrero Ave
415-236-8870

LA MESA, CA
8363 Center Dr.
619-461-0110

LOS ANGELES, CA
2309 S. Flower St
213-749-0261

POMONA, CA
1555 N. Orange Grove Ave
714-623-3543

REDWOOD CITY, CA
2001 Middlefield Rd.
415-365-8155

SACRAMENTO, CA
1860 Fulton Ave.
916-486-1575

WOODLAND HILLS, CA
22504 Ventura Blvd.
818-883-0531

WESTMINSTER, CO
8725 Sheridan Blvd
303-422-3408

JACKSONVILLE, FL
8262 Arlington Expressway
904-725-4554

HIALEAH, FL
4705 W. 16th Ave.
305-823-2280

PLANTATION, FL
7173 W. Broward Blvd
305-791-7300

TAMPA, FL
4019 W. Hillsborough Ave
813-886-2541

ATLANTA, GA
5285 Roswell Rd.
404-252-4341

HONOLULU, HI
98-1254 Kaahumanu St.
808-487-0029

CHICAGO, IL
3466 W. Devon Ave.
312-583-3920

DOWNERS GROVE, IL
224 Ogden Ave.
312-852-1304

INDIANAPOLIS, IN
2112 E. 62nd St.
317-257-4321

MISSION, KS
5960 Lamar Ave
913-362-4486

LOUISVILLE, KY
12401 Shelbyville Rd.
502-245-7811

KENNER, LA
1900 Veterans Memorial Hwy.
504-467-6321

BALTIMORE, MD
1713 E. Joppa Rd.
301-661-4446

ROCKVILLE, MD
5542 Nicholson Lane
301-881-5420

PEABODY, MA
242 Andover St. (Rt 114)
617-531-9330

WELLESLEY, MA
165 Worcester St (Rt. 9)
617-237-1510

FARMINGTON HILLS, MI
29433 Orchard Lake Road
313-553-4171

EAST DETROIT, MI
18149 E. Eight Mile Rd
313-772-0416

ST. JOSEPH, MI
Lakeshore Dr. near Hilltop Rd
616-982-3215

HOPKINS, MN
101 Shady Oak Rd.
612-938-6371

ST. PAUL, MN
1645 White Bear Ave.
612-778-1211

BRIDGETON, MO
3794 McKelvey Rd.
314-291-1850

OMAHA, NE
9207 Maple St.
402-391-2071

OCEAN, NJ
1013 State Hwy 35
201-775-1231

FAIR LAWN, NJ
35-07 Broadway (Rt. 4)
201-791-6935

AMHERST, NY
3476 Sheridan Rd.
716-835-3090

JERICHO, LI
15 Jericho Turnpike
516-334-8181

ROCHESTER, NY
937 Jefferson Rd.
716-424-2560

N. WHITE PLAINS, NY
7 Reservoir Rd.
914-761-7690

GREENSBORO, NC
4620C W. Market St.
919-299-5390

CLEVELAND, OH
28100 Chagrin Blvd.
216-292-7553

COLUMBUS, OH
2500 Morse Rd.
614-475-7200

TOLEDO, OH
48 S. Byrne Rd.
419-537-1887

WOODLAWN, OH
10133 Springfield Pike
513-771-8850

OKLAHOMA CITY, OK
2727 N.W. Expressway
405-848-7593

FRAZER, PA
630 Lancaster Pike (Rt. 30)
215-647-5555

PHILADELPHIA, PA
6318 Roosevelt Blvd.
215-288-0180

PITTSBURGH, PA
3482 Wm. Penn Hwy.
412-824-3564

WARWICK, RI
558 Greenwich Ave.
401-738-5150

DALLAS, TX
2715 Ross Ave
214-826-4053

FORT WORTH, TX
6825-A Green Oaks Rd.
817-737-8822

HOUSTON, TX
1704 W. Loop N
713-869-5263

NORTH HOUSTON, TX
5050 FM 1960 W
713-583-1163

SAN ANTONIO, TX
7111 Blanco Rd.
512-341-8876

MIDVALE, UT
58 East 7200 South
801-566-4626

ALEXANDRIA, VA
6201 Richmond Hwy.
703-765-5515

VIRGINIA BEACH, VA
1055 Independence Blvd
804-460-0997

SEATTLE, WA
505 8th Ave. N.
206-682-2172

FEDERAL WAY, WA
2120 S. 320th (Sea-Tac Plaza)
206-941-6940

VANCOUVER, WA
516 SE. Chaklov Dr.
206-254-4441

MILWAUKEE, WI
5215 W. Fond du Lac Ave.
414-873-8250

Christmas Special

HS-148 IBM Compatible Computer System

- 256K Memory
- 360K Disk Drive
- Free MS-DOS
- Start-up diagnostics
- HVM-122A Monitor

ONLY **$99900** Complete

LEASING AVAILABLE FOR YOUR BUSINESS NEEDS

VISA MasterCard

ZENITH data systems
AUTHORIZED DEALER

Your TOTAL SERVICE computer center

• Service • Support • Software • Accessories • User Training • Competitive Prices

HZC-167

Heath ZENITH®

Units of Veritechnology Electronics Corporation

Computers & Electronics

Automatic Ramdrive Utilities

Joseph Hebda
415 Park Circle
Sun Prairie, WI 53590

Ramdrive speed is really great! The electronic disk is an absolute winner for applications with a large amount of I/O. The name of the game is to up-load the RAM, process an application, and down-load updated files. It sounds all so simple, doesn't it.

After recently biting the bullet (plastic money) I ordered a 512K memory expansion for my H/Z-100. Since my biggest (dBASE II) systems are running under CP/M, I also ordered the RAMdrive software for CP/M-85. Installation of both was easy-no problems. The RAMdrive software creates a pseudo-disk as drive M: for the computer.

Most of my dBASE II systems are turnkey--they automatically boot up from the CP/M cold boot command line directly to the application's main menu. So naturally, I first tried loading the RAM-drive from within dBASE II. I could not get dBASE II to work correctly when switching over from drive A: to drive M: so as a last resort, I looked up the problem in the documentation. Under the 'RESET' command is a small but important rule:

> 'Do not swap and RESET the drive which contains the dBASE system command files.'

I wanted to do a 'RESET M:' in order to thereafter execute dBASE from the RAMdrive. At this point, I was frustrated by hitting this dead end after countless hours of trying to 'make it work'. However, I still knew that there had to be a way (an easy way) to do RAM up and down-loading.

Fortunately, at our next (Madison, Wisconsin) HUG meeting, when I explained the problem and what I was trying to do, someone suggested I look into the CP/M 'SUBMIT' command. Sure enough, this was another good lead and a possible solution. After several experiments, I found it works just fine.

Now I will explain how up-loading and down-loading the RAM works using the 'SUBMIT' command. The first step was to use 'CONFIGUR' to change the cold boot commmand line. Instead of calling dBASE, I now want to 'SUBMIT AUTOSTRT'. See Listing 1 for a look at the contents of AUTOSTRT.SUB file. The purpose of this file is to display instructions for RAMdrive processing; it doesn't really 'do' anything except show the choices at startup time. For the display to work, file 'SUBMIT.COM' must be present on the cold boot default drive. 'SUBMIT.COM' can be copied to the drive from any other diskette where it exists by a command such as this:

```
PIP A:=B:SUBMIT.COM[OV]
```

This will transfer SUBMIT from B: to A:.

If you wish, you may skip typing in AUTOSTRT.SUB and change the CP/M cold boot command line to 'SUBMIT AUTOCOPY' in order to start the RAMdrive loading as the very first step. I prefer to have the instructions display first for two reasons:

1. Documentation of the step-by-step and 1-step methods.
2. It gives the person using the system something very easy to do right away, and gives them at least some 'control' over what the computer will do.

Note that the 'SUBMIT' commmand does NOT use the '.SUB' extension whereas the file which is submitted will be named <filename>.SUB.

Use CP/M's 'ED' (or similiar) text editor to create the 'SUBMIT' files. It is a good idea to type in the first submit file and test it out to see how it works. Check the CP/M manual for an excellent example of how parameters are passed into the 'SUBMIT' file.

For the rest of this article, I'll briefly describe each of the remaining files. Listing 2 (AUTOCOPY.SUB) contains commands to upload the files to the RAMdrive as drive M:, to execute the dBASE II system, and begin file downloading. The statement for the start of dBASE II can be changed to execute a BASIC program or any

other system of your choice. The very last statement is another submit for downloading from the RAMdrive.

Listing 3 (AUTOSAVE.SUB) uses the parameters 'M' and 'B' passed from AUTOCOPY in the PIP command. This will transfer data from the RAMdrive onto disk drive B. Because the RAMdrive is 576K and the diskette less than 320K, the 'PIP' command is used with the '.DBF' and other extensions to selectively copy files. These are the B: drive files being unloaded.

Once the files are saved, a final set of messages in Listing 4 (SHUTDOWN.SUB) is displayed. 'SHUTDOWN' is similiar to 'AUTOSTRT' in that both are for informational purposes only. I realize that all four of the submit files could be combined into one file because each file uses 2K, but I feel that breaking the commands into 4 sections is not overly complex. Because each submit file is a separate module, each is restartable and can be used independently.

Listing 5 (AUTOLOAD.SUB) is used for the step-by-step load method as explained in the first display, AUTOSTRT. The last two submit files—Listing 6 and Listing 7 (AUTOPROG.SUB and DRIVEB.SUB) are specialized for dBASE II systems. The command 'SUBMIT AUTOPROG M A' is used to make a backup copy of dBASE II program (.CMD) files that were modified (with MODIFY COMMAND) on the RAMdrive. The DRIVEB.SUB file resides on drive B: and is used for systems that require program files on drive A; and data files on drive B:. The cold boot command line for drive B: is CONFIGUR'ed to 'SUBMIT DRIVEB' so that if this disk is ever cold booted from the A: drive, this warning display will appear.

The minimum SUBMIT files needed to make this RAMdrive utility work properly are AUTOCOPY and AUTOSAVE. The added files are useful to make the computer system appear 'user friendly.' I do hope that this information is helpful to fellow HUGgies with a memory expansion board or a thing about buying one.

Now that all the submit files are keyed, and the cold boot command line CONFIGUR'ed, it is time for testing. Press <CTRL-RESET> for auto-boot from A: or B)oot the system. After the AUTOSTRT display type SUBMIT AUTOCOPY and sit back while the computer does the work!

Listing 1

```
A>type AUTOSTRT.SUB
;
,  Welcome to  'AUTOSTRT'  !!
,
,  Joe Hebda  - 19 January 85
,
;
;  There are two ways to start processing.
,
,      1   Step-by-step commands
;
;             SUBMIT AUTOLOAD A M
;             SUBMIT AUTOLOAD B M
,             M:DBASE MCMENU
,
;          ( When done, be sure to enter command
,             SUBMIT AUTOSAVE M B
,
,      2.  One command
;
;             SUBMIT AUTOCOPY
;
,  Please type a command, and press 'RETURN' key now
```

Listing 2

```
A>type AUTOCOPY.SUB
,
;  Start of 'AUTOCOPY'  processing.
;
;  Joe Hebda - 19 January 85
,
,
,  The following instructions will
;
,  1   Load the  A: Disk
,  2   Load the  B: Disk
,  3   Automatically Startup  dBASE II
;  4   Begin dBASE II processing.
,  5   Unload the  B: Disk from the computer's memory
,      when the dBASE II processing is ended.
,
,  Please be patient as this will take several minutes
,      Take a break !!!    HA HA HA !!!
,
;      Loading disk drive A' into memory now.
PIP M:=A:* *[RWOV]
,
,      Loading disk drive B' into memory now
PIP M:=B:* *[V]
,
,  Thank you for waiting, we are almost ready to begin
;
M:DBASE MCMENU
,
,      Unloading the computer's memory now
SUBMIT AUTOSAVE M B
```

Listing 3

```
A>type AUTOSAVE.SUB
,
;AUTOSAVE = Make a backup copy of changes before shutdown
,
;  The following changes are saved'
,
;      1  'DATA' files
;      2  Special 'MEMORY' files
;      3. Indexes to the 'DATA' files
,
;  Please wait a few minutes while I copy the files
,
;      Copying 'DATA' files now
PIP $2:=$1.*.DBF[V]
,
,      Copying 'MEMORY' files now
PIP $2:=$1.*.MEM[V]
,
;     . Copying 'INDEX' files now
PIP $2:=$1.*.NDX[V]
,
,    AUTOSAVE  << Ending Now >>
,
SUBMIT SHUTDOWN
```

Listing 4

```
A>type SHUTDOWN.SUB
;
;  ********** SUCCESSFUL END OF PROCESSING **********
,
,  When this display stops, it will be  O. K
,      to remove the disks,
,      and shut the power off
,
,  It was good working with you   Have a  NICE DAY !!!
```

Listing 5

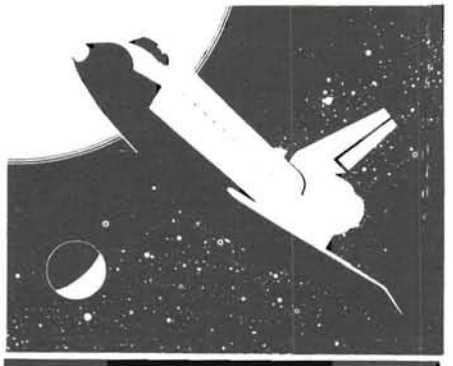
```
A>type AUTOLOAD.SUB
;
;  AUTOLOAD = Automatic Load of All Information
;             << Starting Now>>
;
PIP $2:=$1.*.*[RWOV]
;
;  Auto Load  <<< Successfully Completed >>>
```

Listing 6

```
A>type AUTOPROG.SUB
;
;  AUTOPROG = Automatic Program Copy << Starting Now >>
;
; Please wait a few minutes while I copy the programs
;
;      Copying 'COMMAND' files now
PIP $2:=$1:*.CMD[V]
;
;    AUTOPROG    <<<<  Ending Now  >>>>
```

Listing 7

```
A>B:
B>type DRIVEB.SUB
;
; 0 0 0 P P P S S  !!!  This disk contains 'DATA'
;
; and is being read from the  A.  (left drive)
;
; After this display stops, please check the disk drives
;
; for a possible error condition
;
;    I suspect the disks were switched around.
;
;   Drive  A: (left)  should contain  'PROGRAMS'
;   Drive  B: (right) should contain  'DATA'
;
;     When this display stops,
;
;         1.  Reverse the disks
;
;         2   Re-Start the machine by pressing 2 keys
;             at the same time   The keys are
;
;             CTRL  RESET
;
;  Thank you for your help !
```

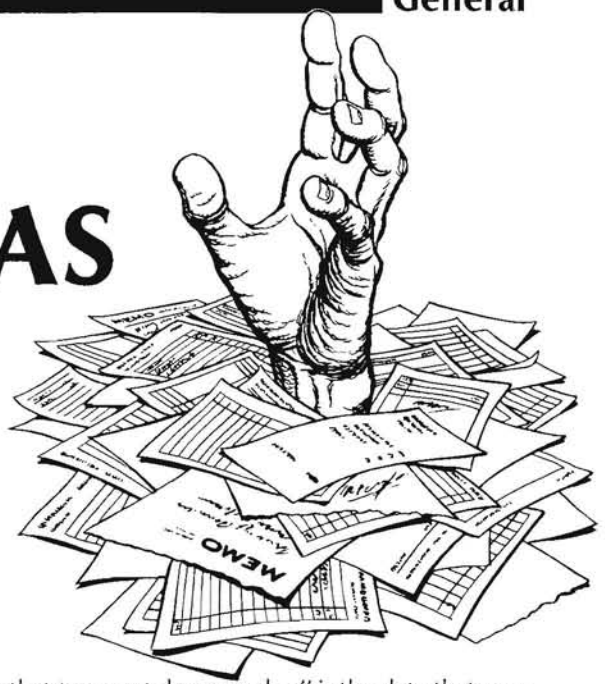


NOTICE!

Perks Price Change

The Heath Users' Group accidentally printed an old Perks ad on the rear cover of the November issue of REMark magazine. This advertisement incorrectly stated that the price of the Perks Desktop Utility was $99.97. In fact, the price of the **Perks Desktop Utility** was reduced **from $99.97 to $69.95** on August 1, 1985 and remains at $69.95. We regret any confusion caused by the appearance of the older advertisement in the November REMark magazine.

MORTGAGE.BAS



Bob Moskus
2511 Alpine Trail
Huron, OH 44839

MORTGAGE.BAS was written in Microsoft BASIC (ver 5.2) on an H-89 running CP/M. From the line-by-line explanation later, you should be able to convert this program to run under most any other BASIC.

This program assumes an 80 column display. The output of this program, either to line printer or CRT, can be useful as a check against your mortgage statements. One might use this program to answer the question: "What if I paid extra . . . how soon will I pay it off ?"

Please note that this program assumes a payment was made "ON THE DUE DATE". If there is a discrepancy between the output of this program and your statement from the lender, examine the timing of your payments. The lender's statement may be crediting you for a payment made early. An inquiry to the lender may be in order if you suspect something is in error.

It should also be noted that the output of this program cannot be guaranteed. The intent is to provide a cross-check to your mortgage statement and some programming concepts.

The program can be used for any account which has a fixed interest rate and payments that are due on a monthly basis, an auto loan, for example.

When run, the program will prompt you for inputs of data about your mortgage. It will then calculate and print an itemized listing of your mortgage activity by monthly due dates. The itemized listing will run until your balance is less than your payments at which time your final payment, including interest, will be shown. Finally, a recap will show the grand totals of principal and interest paid. For your test run, just enter the values shown as examples in the INPUT statements.

The program is self-prompting, although some of the inputs require explanation. Check your Mortgage Note agreement for specific data to be input.

The "FIXED INTEREST RATE" is the annual interest rate. Floating interest rates are not accommodated by this program as written. Once the program is understood, loops can be added to input floating interest rates, test for a match with effective dates and adjust the calculations for subsequent execution.

The "first date that payments became due" is the date that your FIRST payment was due, usually the 1st of the month.

The "BASE PAYMENT" input is the amount you agreed to pay on a monthly basis towards the balance. This must not include those amounts applied to escrow.

The "DUE DATE of Additional Premium payment" is the due date on which you made a payment in addition to your base payment.At the "How much Additional Premium payment" prompt, enter the amount you paid over and above your base payment.

If you continued making this additional premium payment, no further inputs are required. If, however, you stopped making the additional premium payments, then at the next "DUE DATE ..." prompt, enter the due date that you reverted back to base payments. Then, at the "How much Additional Premium" prompt, just hit RETURN. Of course, if you increased your additional payments, enter the date and the new amount total.

If you wish to see how a future additional premium payment will affect your "PAID IN FULL" date, just enter the DUE DATE and then the amount you are thinking about paying.

A carriage return at the "DUE DATE" prompt will start the program on it's merry way.

Line-by-line explanations are as follows:

```
30 ~ Clear all memory
40 ~ Assign variables used and set up dimensions for arrays
        BEL$    Ring the bell at the terminal
        N=255   The maximum allowed in an array.
        CLS$    Clear the screen
        A$-C$   Define the format used to print numbers
70-140 ~ Prompts for input of mortgage data and gives you
         the chance to correct any erroneous data
150 ~ Begins the loop to input additional premium payments
      and their dates.
220-260 ~ Manipulates the data input from lines 180 and
          210   Tests for previous entries and reassigns
          values as necessary for accurate computations.
       L(I) is the current value of add'l premium
       LL(I) is an adjusted value used in line 370 to
          calculate the Principal
       Ll is the value of the previously entered add'l
```

```
        premium.
      ABS is a function that returns the absolute value
        of its argument  (disregards the sign (+/-)
        of the number)
270 ~ L1 is now assigned the current value of L(I) and
        becomes the "previous value" when the NEXT I
        instruction is executed in line 280
300 ~ set FLAG to 1 and POKE 3,171 which sends to the line
        printer that which would have gone to the screen
310-320 ~ The CSNG function converts the numerical data to
        single precision
       B# - Balance
       P# - Principal
      IR#- monthly Interest Rate
330 ~ Converts the string value of date input to month(MO),
        day(D), and year(Y)
340 ~ Branches to subroutine at line 1010 where page header
        and current balance are printed
350-560 ~ Constitutes the printout loop for each month's
        calculations and more clearly defined below.
350 ~ The Z loop is incremented once for each monthly
        transaction
360 ~ The I loop tests the current date for a match with
        one of the dates on which an add'l prem. payment
        was made   The loop is only as large as the number
        of entries made in a previous loop at line 150
        (N1, as assigned in line 190)
380 ~ L2 is the value of the next increment of "Z". Allows
        repeating the amount of additional premium until
        the next date that a change was made.
400 ~ Calculates and rounds off the Interest(I#),
        Principal(P#), and Balance(B#)  The FIX function
        discards all data to the right of the decimal
        point after we round off the value inside the
        parenthesis. The FIX function is needed to correct
        round off errors due to the computer's use of
        Binary math   We perform a FIX on the calcula-
        tions during every step of the "Z loop" to keep
        the numbers right  (See BUGGIN' HUG, REMark Issue
        42, page 92 for a discusion of Binary math errors)
410 ~ Keeps a running total of Principal and Interest
        values for the recap.
420-470 ~ Prints the date, base payment, add'l prem
        payment, principal applied to balance, interest
        on previous balance, and the current balance
480-490 ~ Increments the month by one until December is
        reached, then turns it over to January  It also
        increments the year and turns it over when the
        next century starts
500 ~ Detects Balance less than zero  This problem may
        occur if an increase in add'l principal was
        made on a date where the balance was less than
        the sum of the base payment and add'l premium
        Without this line, the program would go into
        an infinite loop
510 ~ Tests for present balance being less that the
        premium payment   If true, exit loop and
        finalize data
520-540 ~ Increments line counter  Allows 20 lines to
        CRT or 50 lines to printer depending upon your
        selection at line 290
550 ~ Since the maximum DIMension argument is 255, and a
        30 year mortgage has 360 months, we have to
        start the Z loop over   But first, we assign
        the latest add'l prem  amount to L1(1), the
        first step of the new Z loop
570-690 ~ is the finalizing routine  You'll see what it
        does when you run it
700 ~ Resets ANS$ to null ("") and returns output to CRT
        if not there already
710-720 ~ Gives you a second chance at a hard copy printout
        If Yes, then reset the Premium total, Interest
        total, and Principal to zero   All other variables
        remain intact and the program branches back to
        do it all again but now to the line printer
730 ~ Returns output to CRT and END the program.
```

A note of caution is in order. If the program is interrupted during output to the line printer by either an error or a CTRL-C, all BASIC commands are still active. What you would expect to come up on the screen will be going to the printer. This could be a little confusing, so the best thing to do after the interrupt, is to type "GOTO 730". This will return output to the screen.

There is plenty of room for modification and improvement. Save a file of it as MORT.TST and play around with it. After all, that is what the learning experience is all about. One idea might be to add a printout of the Year-to-Date interest paid. You could do this as follows:

add to line 410 ... :YTD=YTD+I#
add to line 480 ... :YTD=0
insert line 475 IF MO=12 THEN PRINT "YTD Interest"; USING C$;YTD

The rest of the ideas are up to you!

I hope this program is of use to some of you. If nothing else, perhaps there are some concepts that you can use in your own programs.

```
10 REM   MORTGAGE.BAS      by Bob Moskus
20 '
30 CLEAR
40 BEL$=CHR$(7):N=255:OPTION BASE 1
     DIM J$(N),L(N),L2(N),LL(N)
50 CLS$=CHR$(27)+"E":A$="#######.##":B$="##-##-##"
     C$="$$#######.##"
60 PRINT CLS$
70 INPUT "How much was the Mortgage Loan for <1000.00>",
     F$:PRINT
80 INPUT "What is the Annual FIXED INTEREST RATE <10.00>",
     G$:PRINT
90 INPUT "When was the first payment due (month-day-year)
     <03-01-84> ";H$:PRINT
100 INPUT "What is the BASE PAYMENT  <100.00>";I$:PRINT
110 PRINT BEL$
120 INPUT "Is the above info correct <Y or N> ";ANS$:PRINT
130 IF ANS$="N" OR ANS$="n" THEN 60
140 IF ANS$="Y" OR ANS$="y" THEN 150 ELSE 110
150 FOR I=1 TO N
160 PRINT
     "DUE DATE of Additional Premium payment <07-01-84> "
170 PRINT "   or date you stopped add'l prem  payment "
180 INPUT "     Hit  RETURN  if no entry ":J$(I):PRINT
190 IF J$(I)="" THEN N1=I:GOTO 290
200 PRINT "  HOW MUCH Additional Premium <5.00> "
210 INPUT
     "or hit RETURN if you stopped add'l prem  payment ",
     L(I):PRINT
220 IF L(I)=0 THEN LL(I)=L(I)-L1:GOTO 270
230 IF L1=0 THEN 260
240 IF L(I)<L1 THEN LL(I)=L(I)-L1:GOTO 270
250 IF L(I)>L1 THEN LL(I)=L(I)-ABS(L1):GOTO 270
260 LL(I)=L(I)
270 L1=L(I)
280 PRINT:NEXT I
290 PRINT BEL$:PRINT
     "Do you want a hardcopy printout of this?<Y or N>",
     ANS$=INPUT$(1):PRINT ANS$
300 IF ANS$="Y" OR ANS$="y" THEN FLAG=1:PRINT:PRINT
     PRINT "I'm working on it".
     POKE 3,171 ' -> OUTPUT TO LINE PRINTER
310 B#=CSNG(VAL(F$)):P#=CSNG(VAL(I$))
320 A=VAL(G$).A=(A/12)/100:IR#=CSNG(A)
330 MO=VAL(LEFT$(H$,2)):D=VAL(MID$(H$,4,2))
     Y=VAL(RIGHT$(H$,2))
340 GOSUB 1010
350 FOR Z=1 TO N
360 FOR I=1 TO N1
370 IF MO=VAL(LEFT$(J$(I),2)) AND Y=VAL(RIGHT$(J$(I),2))
     THEN L2(Z)=L(I):P#=P#+(CSNG(LL(I)))
```

```
380 L2(Z+1)=L2(Z)
390 NEXT I
400 I#=FIX((B#*IR#)*100+.5)/100:
    PR#=FIX((P#-I#)*100+.5)/100:
    B#=FIX((B#-PR#)*100+.5)/100
410 PRT#=PRT#+PR#:IT#=IT#+I#
420 PRINT USING B$;M0;D;Y,
430 IF L2(Z)<0 THEN L2(Z)=0
440 PRINT USING A$;VAL(I$),
    IF L2(Z)=0 THEN 450 ELSE PRINT USING A$;L2(Z),
450 PRINT TAB(33)USING A$;PR#;
460 PRINT TAB(44)USING A$;I#;
470 PRINT TAB(59)USING A$;B#
480 IF M0=12 THEN M0=0:Y=Y+1
490 M0=M0+1:IF Y=100 THEN Y=0
500 IF B#<0 THEN PRINT
    "DATA ERROR, Balance is less than ZERO":STOP
510 IF B#<PR# THEN 570
520 P=P+1
530 IF FLAG=0 AND P=20 THEN INPUT
    "Hit RETURN to continue";ANS$:P=0:GOSUB 1010
540 IF FLAG=1 AND P=50 THEN PRINT CHR$(12):P=0:GOSUB 1010
550 IF Z=250 THEN L2(1)=L2(Z):GOTO 350
560 NEXT Z
570 PR#=B#+(B#*IR#). PRT#=PRT#+PR#: I#=B#*IR#: IT#=IT#+I#
580 PRINT USING B$;M0;D;Y,
590 IF C=0 THEN 630
600 PRINT TAB(9)USING A$;C;
610 PRINT TAB(33)USING A$;PR#-I#;
620 IF C>0 THEN 650
630 PRINT TAB(44)USING A$;I#;
640 PRINT TAB(59)USING A$;B#+I#:C=PR#:GOTO 580
650 PRINT TAB(59)USING A$;B:PRINT:PRINT:PRINT
660 PRINT "interest paid"TAB(22)USING A$;IT#
670 PRINT "principal paid"TAB(22)USING A$;PRT#-I#
680 PRINT TAB(22)"----------"
690 PRINT "GRAND TOTAL PAID";TAB(20)USING C$;IT#+(PRT#-I#)
700 ANS$="":POKE 3,169 '  ->Return output to CRT
710 IF FLAG=0 THEN PRINT BEL$,
    TAB(40)"Now do you want a printout <Y or N>?",
    ANS$=INPUT$(1):PRINT ANS$
720 IF ANS$="Y" OR ANS$="y" THEN PRT#=0:IT#=0:C=0:
    GOTO 300
730 POKE 3,169.
    END ' -> Return output to CRT and END the program
740 '
750 '
1000 REM  *** print Page Header and current balance ***
1010 IF FLAG=1 THEN PG=PG+1:PRINT TAB(70)"page";PG
1020 PRINT CLS$;TAB(15)"Base"TAB(28)"Add'l"TAB(39)"Applied"
1030 PRINT "Due date"TAB(13)"Payment"TAB(25)
    "Principal"TAB(37)"Principal"TAB(49)"INTEREST"
    TAB(63)"BALANCE"
1040 PRINT TAB(59)USING A$;B#:RETURN
```

*

Controlled Data Recording Systems Inc.

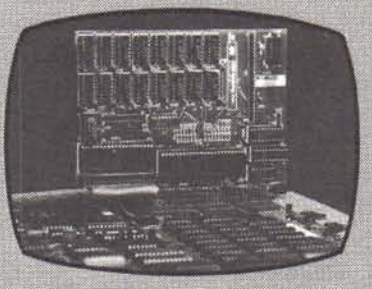
Quality Products and Support for the Heath/Zenith Community

New for the H/Z89-90 Computer Users:

The Super RAM 89 Package

Now get the speed and power of a high capacity Ram Drive System at a reasonable price.

Using the Ram 89 package, standard software shows an immense improvement in speed. Depending on the software being run, programs may execute 10 times faster than when run through standard floppies.



Now get RAM 89

HDOS DRIVER Software Only **$35**

RAM 89 DVD supports RAM SY0: and SY1:, added autoload features, plus more!

Ram 89 comes as a two card set that plugs into the left hand (16k expansion) side of the computer. No computer modifications required.

Board 1 has two banks of 256k chips possible for a total of 512k. Either or both banks are able to use 64k chips instead. Board 1 can be used by itself, with board 2 added at a later date.

Board 2 has an additional 512k, plus it has a real time clock capability, and a SASI interface hardware capability. Board 2 piggybacks onto board 1.

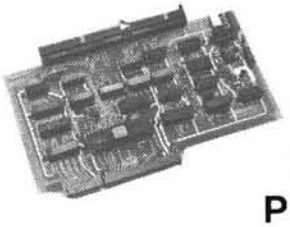
The Ram Drive Software (SRAM) allows one or two logical ram drives. The ram drive(s) can be located starting anywhere from logical A: to O: (standard drives get relocated). SRAM can be set to start at logical A: and warm boot with ram (no floppy disk accesses needed). Ram drive attatches to any of the versions of CP/M 2.2 bios used in the H/Z89-90.

GET A FULL MAGABYTE OF RAM FOR YOUR H/Z89-90

Board 1 includes hardware manual and ram drive software with no ram: $190.00. Each 256k bank, add $56.00. Board 2 (must have board 1). With no ram, no clock, no SASI $90.00. Each 256k bank, add $56.00. Ask about clock, SASI pricing.

FOR THE H/Z89-90 COMPUTERS

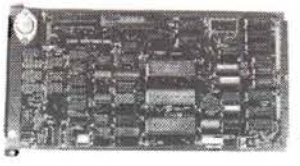
THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE:$395



Includes controller board CP/M boot prom, I/O decoder prom, hardware/software manuals BIOS source listing. HDOS driver now available for $50.00.

Now be able to run standard 8" Shugart compatible drives and 5.25" drives (including the H37 type) in double and single density, automatically with one controller. The FDC-880H operates with or without the Heath hard sectored controller.

THE FDC-H8 DOUBLE DENSITY 8" AND 5.25" CONTROLLER FOR THE H8 COMPUTER PRICE $495



Has all of the capabilities of our popular FDC-880H controller, with added features:

- Direct memory access (DMA) data transfer
- Hard sectored controller (H17) incorporated on the board.
- Runs with the standard 8080 CPU card and with Z80 CPU upgrades.

MODIFY89

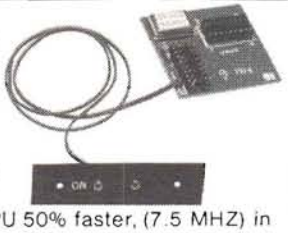
Allows owners of the FDC-880H or the FDC-H8 to read/write to the following disktypes:

Access S.S.D.D., Cromemco S.S.D.D., DEC VT180 S.S.D.D., IBM PC/Zenith 100, (CP/M) D.S.D.D.*, Kaypro II S.S.D.D., Morrow Micro Decisions S.S.D.D., NEC PC-8001 A S.S.D.D., Osborne S.S.D.D., Osborne S.S.D.D., Otrona D.S.D.D.*, Zorba D.S.D.D., TI Professional S.S.D.D., TRS-80 Model I (Omnikron CP/M), TRS-80 Model III (MM CP/M), Xerox820 S.S.D.D., Zerox820-II S.S.D.D. Plus more, Standard (Tests for H/Z37 and C.D.R. Disk types)

PRICE $49.95

FOR THE Z100 COMPUTERS

THE ORIGINAL Z100 SPEED MODULE. RUN YOUR Z100 PROGRAMS FASTER



The ZS100 runs the Z100 CPU 50% faster, (7.5 MHZ) in 8088 mode. The ZS100 installs easily with no soldering. The ZS100 is externally switchable between speed mode and normal. The ZS100 improves the time performance of applications packages with no software modifications needed. The ZS100 is for all Heath/Zenith 100, 110 and 120 series computers.

SUPER DRIVES 85

A powerful floppy drive package for the Z100 or any computer that can use 8" floppy drives.

Two extended technology 5.25" drives in a package with interface cables, case and power supply, that look exactly like standard 8" floppy drives to the computer. Now have **2.4 megabytes** of **floppy file storage capacity** in one 5.25" package.

ONLY **$695** LIST

Dealer inquiries invited

For information about these products contact.
Controlled Data Recording Systems Inc.
7210 Clairemont Mesa Blvd
San Diego Ca 92111 Phone (619) 560-1272 or contact a Heath/Zenith dealer that carries the C.D.R. Systems Inc. line of products.
New - C.D.R. Systems Bulletin Board (619) 560-8929

Mighty Mini RTTY

(Radio Teletype On A Low Budget)

Ray Isenson, N6UE
4168 Glenview Drive
Santa Maria, CA 93455

Introduction

A $9.26 cash layout, an admittedly fairly well stocked JUNQUE box and a few hours of "design and development" was the main cost to interface an H–89 to an amateur radio and to join in the fun of amateur radio telegraphy — RTTY. Had there been no stock of parts to draw upon, the total cost might have gone as high as $25.00; but I doubt it. The job turned out to be so simple that one might almost believe that the "troops" at Heath, when laying out the H–88/H–89, had envisioned doing just this some day. About the same general design will suffice for the H–8; with a couple of differences in detail that will be touched upon later. There will be additional work required to install the Modem into an H–89A or a Z–90; but only a very little bit. The "why" of that will shortly become obvious. A view of the home brewed Modem, mounted on the leftmost set of pins of the H–89 motherboard, is seen in the accompanying photograph. There isn't much to it. Anyone who built the H–89, or almost any Heathkit(tm) for that matter, should find it simple to build a similar Modem. The details are covered in the following paragraphs.

I etched and drilled a printed circuit board for the version that is seen in the photo. It could have as easily been on a wire wrap board. Had I been clever enough to pick up one of the H–88–10 wire wrapping boards while they were still available, it probably would have been. As it was, a 4 X 5 inch piece of copper clad epoxy board remaining from some prior project was on hand and used. As the layout is not critical to functioning of the board, but is sensitive to such matters as the exact size of the capacitors you are going to use and the type of variable resistors you have or buy at the local hobby shop, and as you may prefer wire wrap or a modified wire wrap/point-to-point technique, the PC layout is not furnished. It is so simple, however, that an adequate design easily can be deduced from the photograph.

The Circuit

The circuit schematic is shown in Figure 1. It consists of 5 readily available ICs, a pair of common audio transistors, two signal diodes, an LED and a handful of passive components. Operationally, an audio RTTY signal is brought from a radio receiver to the Modem where it is subjected to limiting and clipping by the diodes, D1 and D2, and is then amplified and filtered by U1, an LM348 or equivalent quad op amp, to screen out undesired adjacent signals. The processed audio goes to U2, an EXAR XR2211 demodulator, where the mark and space bits are detected. The data then stream to U3, an SN74LS86 Exclusive OR, which serves mainly as a buffer between the Modem and the computer, through it to the serial I/O board and finally to the computer's CPU board.

Outgoing data flows from the I/O port to U3, still functioning as a buffer and thence to U4, an EXAR XR2206 modulator where the TTL signal is converted to audio tones; 2125 Hz for a "1" or 2295 Hz for a "0", or sent to the base of Q2, the 2N2222 transistor that keys the transmitter on. The frequency shifted tones generated by U4 go through U5, a single op amp, 741 or equivalent, which buffers the Modem from the microphone input to the radio transmitter. The technique of sending the frequency shifted tones to the microphone input is known as AFSK (Audio Frequency Shift Keying). This approach is used in the currently very popular VHF FM RTTY in the metropolitan areas and also is fully acceptable in RTTY exchanges between Single Side Band stations using the High Frequency bands. Some Amateur Radio HF equipment is designed to transmit a frequency shifted signal when the transmitter is sent a string of a "0's" and "1's", much like a TTL data string but not necessarily at the 0 and +5 Volt DC levels. A station that is in the latter mode is said to be operating FSK. The Modem design has incorporated a facility to support the FSK mode. It consists of the line going from the same output from U3 that goes to U4 but that, instead, goes directly to the transmitter's FSK input.

Listing 1 provides a general Bill of Materials for the unit. The attempt to minimize the number of different components required should be evident. One example is the 14–4.7k ohm resistors, instead of 14 different values from 2k to 6.8k. The capacitors that are asterisked should be mylar or polyxxx rather than simple disk caps. I used 2–.01uf mylars in parallel for C12 to make up the .02uf required. Similarly, pairs of .01uf disks, in series, are used at C13 and C14 because I didn't have any .005s handy. Multiturn

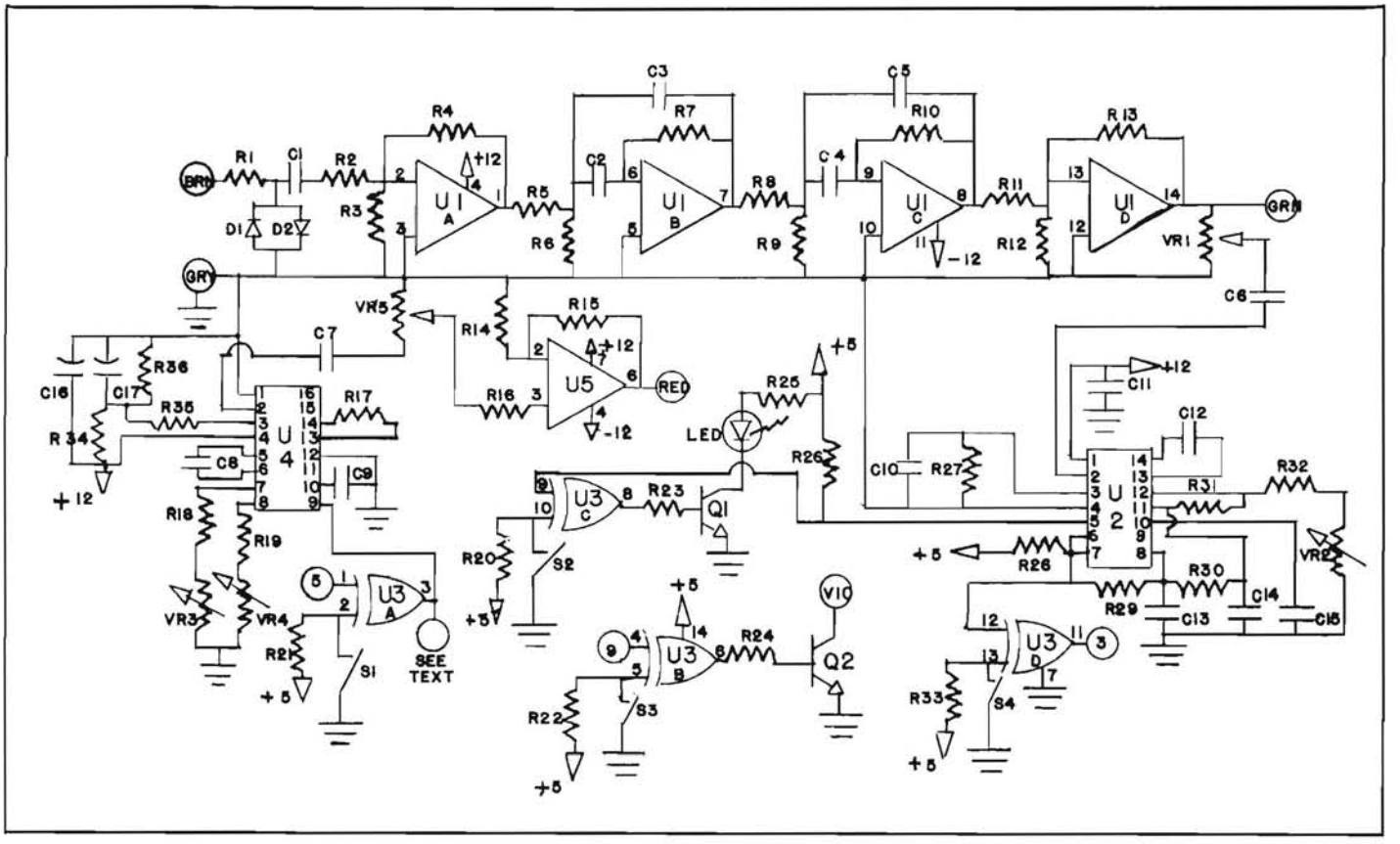


Figure 1

trimpots were used for VR2 through 4 because they were readily available. I wouldn't have bought them especially for the job. The standard single turn PC board potentiometers should be adequate.

U1b and U1c comprise the input filter. The design criteria were: a center frequency of 2210 Hz, Q=11 and Gain=.85. U1a and U1d are additional gain stages. The latter two, working together with the input diode limiters tend to minimize any circuit sensitivity to variations in amplitude of the incoming signals . . . an automatic gain control of sorts. For this circuit to work properly, all of the associated resistors and capacitors should be 5% or better. This is particularly true for C2, C3, C4, and C5, and R6, R7, R9 and R10. Any deviation will result in a shift of the filter's center frequency. If the latter moves below 2125 Hz or above 2295 Hz, it will be difficult to get good Modem performance. Unless you are very sure about the components, it would be wise to replace R6 and R9 with 1000 ohm potentiometers. This will permit compensation for significant variation in the other components.

Current requirements are minimal. I measured them to be 19 ma. at +12, 4 ma. at -12, and 15 ma. at +5 volts. This includes all of the current requirements except for the 10–12 ma. at 5 volts used by the LED. The current can be taken directly from pins 22 through 24 of P507. Pin 25 of that plug provides a circuit ground.

The RS232C serial convention as used by Heath/Zenith for the serial I/O ports specifies -1.5 to -36 volts to represent a data "1" and +1.5 to +36 volts to a data "0". In practice, Heath uses about a -/+12 volt level; these may be observed if measurements are made at the RS232 connector pins on the rear skirt of the H-89 computer. Internally, the computer uses the TTL convention of +5 volts for "1" and 0 for "0", as does the Modem. By remaining at TTL levels, interconnection is simplified and current requirements are reduced (by eliminating two pair of MC1488/1489 ICs that otherwise would be needed).

I use Port 330Q for a telephone Modem and assigned Port 320Q to the RTTY Modem. Pin assignments and IC numbers discussed in the following paragraphs apply to the latter port. For other than the HA-88-3 Serial Interface Board, or if Port 330Q is used, some other numbers will apply. The requisite changes should be obvious and easily made. Considering use of Port 320Q, U607 and U608, a 1488 EIA driver and a 1489 EIA receiver, respectively, serve to convert the TTL signals to and from RS232C levels. I simply removed the MC1488 and 1489 ICs and replaced them with some specially configured dip headers. These are shown in Figure 2.

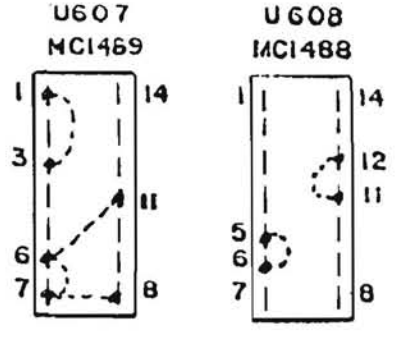


Figure 2

The headers are used to couple the output of the 8250 ACE to the P604 pins at the card edge of the serial I/O board from where a modified cable connects them to the Modem board. You'll notice that some of the pins on the dip header replacing the MC1489 are shorted to pin 7 (ground). These shorts are necessary to provide the "handshaking" that the 8250 will be seeking when the computer tries to ouput to the Modem. The cable was made by unsoldering the wires at the DB25 connector associated with Port 320Q, moving them, instead, to a 5 pin female header that connected to the new Modem PC board.

The schematic shows a line from pin 6 of U2 to pin 8 of U3, thence from U3-10 to Q1 and on to an LED. When U2 locks onto a 2295Hz audio signal, a carrier detect flag is presented at pin 5. When the IC is demodulating a data stream, the blinking LED serves to indicate that fact. In brief, it can be used as a tuning aid to get the radio receiver on frequency. As is done with each of the other signals, U-3 provides buffering and low level drive. Q1, in this case, sinks the current for the LED. The power for the LED was taken from a convenient point on the video board of the H-89. For that reason, the current was not included in the energy budget detailed above.

The H-89 for which the Modem was made first existed as an H-88. To the "grey haired ones" among us that means that it has, on the rear apron, a Molex connector to which tape recorders were once attached. From that connector, inside the computer, there was (is) a 10 conductor cable and a female header. They are recycled conveniently to connect the radio signals to the Modem. (I always knew that someday somehow that connector and cable would get used!) For those with the H-89A or Z-90 there'll be the work of designing and installing some other means for connecting the Modem to the radio. (The extra work that was referenced above.) The square "knockout plug" on the extreme right rear apron of the H-89A conceals the hole which was occupied by the 9 pin Molex in the earlier model computer. The simplest technique might be to replicate the design used originally by Heath for the old H-88.

Input and output terminals of the Modem are indicated by balloons on the schematic. A number indicates connection to the serial I/O board and is the number of the terminal on that board. Should you use Port 330Q, the terminal number will be different, this matter was previously noted. An alpha group, such as VIO for violet or GRY for gray, identifies a color of one of the wires in the cable to the Molex connector that was originally used with the tape recorders. Brown is used for the wire from the radio receiver, green goes to the oscilloscope (see below), violet goes to the "Push-to-Talk" switch, red goes to the microphone input. I used the gray wire and some of the others for ground wires. That particular selection wasn't critical. Those who still have the recorder cable will note that I used the audio coaxes for the AFSK signals and the speaker wire for the monitor and the xmtr keying. The FSK signal mode wasn't fully implemented, in that cabling from the computer to the radio wasn't installed. To do so, bring the signal from the open balloon shown at pin 3 of U3. (See below!) There are available pins in the Molex connector for this.

A 1/4" hole carefully drilled in the lower left hand corner of the CRT escutcheon provides an easy mounting place for the tuning LED. I found it most convenient to use an available assembly with the LED and a current limiting resistor encapsulated in a 1/4" OD cylinder about an inch long. This obviated the need for R25 on my board and simplified mounting of the LED. I soldered the signal wires directly to the LED leads and taped the wire to the CRT bracket to prevent movement. It is hidden behind the bracket in the photo. The LED is held in place by friction.

The filtered audio signal coming from U1 is coupled to U2 through a variable resistor and the .1uf capacitor, C6. Its peak-to-peak value at the input to potentiometer, VR1 can reach 9 volts. This signal is made available at one pin of the Molex to be used as an input to another tuning aid (a monitor scope or oscilloscope). I found it to be invaluable while I was learning to interpret the LED signal. Tuning the receiver to get the filtered audio to the maximum amplitude generally results in a merrily blinking LED and RTTY printing on the CRT.

An important CAUTION is in line about here. As noted, about 9 volts PP can appear across VR1. This is enough to destroy U2. Using the potentiometer, reduce the setting until the signal at the center tap is less than some 6 volts peak-to-peak (2.1 volts rms); for a 2210Hz audio signal at the speaker exceeding .6 volts rms . . . which is enough to drive the filter to maximum output.

VR5 is used to set the microphone drive to the transmitter. Mine is adjusted so that I don't have to change the "microphone gain" at my transmitter when I switch from RTTY to voice.

Alignment

Alignment of the Modem entails setting the center frequency for the XR2211 demodulator, the two shifted frequencies of the XR2206 modulator and the amplitude of the signal for the demodulator and that going to the transmitter microphone input. The controls are the five variable resistors, VR1 – VR5. A digital AC voltmeter and an audio frequency counter are all of the instrumentation needed to align the Modem. An oscilloscope can be useful and can replace the DVM if it has a calibrated DC input. **Hint:** Time and effort can be saved if the .1u capacitor from pin 3 of U2 is not soldered in until after calibration. The Modem need not be in the computer for alignment; in fact, preferably it should not be. That shouldn't create much of a problem in the typical HAM shack. (If necessary use your bench supplies for +12 and +5 volts and a 9 volt dry cell as the "-12" volt source.)

Before applying power, short the center tap of VR1 to ground. Now, to align the demodulator, short pin 2 of the XR2211 to pin 10 (the capacitor open at pin 3) and measure the VCO free running frequency at pin 3. Adjust the VCO fine tuning pot (VR2) to read 2210 Hz. Apply a +5 volt signal to pin 1 of U3 either directly at the chip or at the wire input coming from the serial I/O board. Adjust VR3 to read 2295 Hz at pin 6 of U5 (the 741). Ground pin 1 of U3 and set VR4 for a frequency of 2125 Hz at U5-6. Connect the Modem to a receiver tuned to an RTTY signal in the Amateur Band and, with either the DVM or oscilloscope monitoring the ouput of U1, retune the receiver for peak signal amplitude.

If R6 and R9 were replaced with 1k potentiometers, they should be set at this point. Using an audio signal generator with the frequency at 2210 Hz as input to R1, adjust R6 for peak voltage at pin 7 of U1. Then set R9 for a peak at pin 8. To align the filter without an audio generator, with the Modem connected to the audio output of an HF receiver, turn on the Marker frequency generator and vary the receiver while observing the signal at the input to R1 with a frequency counter until 2210 Hz is seen. This, by the way, is useful to provide about as pure a sine wave as can normally be found. Use this signal to set center frequency on the filter. Disable the Marker oscillator, tune the receiver to an RTTY signal and continue with the alignment.

Now, vary the receiver audio gain until the output signal at U1 is clearly limiting. Move the probe to IC2-2 and adjust VR1 to get a reading of less than 3 volts RMS. With the AFSK output connected to the transmitter, adjust VR5 to get the desired FM deviation or ALC, as the case may be. That's all there is to it. Install the Modem and prepare for that first RTTY QSO.

Not previously discussed are the switches associated with IC-U3. If, for any reason, it becomes necessary to invert a signal, the switch associated with that signal will handle the task. I found, for example, that the polarity of the signal from the I/O board to the Modem was resulting in an "inverted" RTTY signal. Opening the switch at U3-2 took care of that problem. I found that tuning was somewhat easier (for me) when the LED was on for Space and off for Mark. Flip the switch! A closed switch grounds the associated pin and transmits the input signal with no change in polarity. Opening the switch results in signal inversion.

The H-8

The equivalent circuit for an H-8 would be the same as for the H-89, except that the letters and numbers shown in the several bubbles would be different. The technique of remaining at TTL level could be implemented by, for example, selecting Channel 0 for the TTY port. Make up the dip headers as described above. Insert the one that replaces the 1488 at U103 and the one for the 1489 at U104. The LED could be remoted to the terminal, but might more easily be mounted near the digital readouts on the computer. (As one of my mathematics professors of years gone by was wont to say, "I leave such details as where you find the voltage supplies, how you mount the board, and how the radio interface is implemented as an 'exercise'".)

Software

Several RTTY programs in HDOS or CP/M, some in the public domain and some available for a price, can be found. I don't favor any in particular. One CP/M program, ASCIRITY, is available from the HUG as part #885-1238-[37]. I've not seen and can't comment on it, except to note that it uses Port 330 as written and there is no indication that the port can be changed or that the program can be modified for an H-8. This introduces a thought with regard to program selection; be sure that the one to which you have access addresses the I/O port for which your RTTY Modem is configured or that the software can be changed. This may influence your choice of I/O port.

General

Although the Modem is optimized for a narrow (170 Hz) shift, it may print some wider shifts if band conditions are really good. Similarly, the Modem will copy Baudot (Morse code) although I wouldn't suggest doing so. For one thing, the circuit was never intended to be used that way. Also, it would mean that the CW signal would have to be tuned to an (for me) unpleasant frequency . . . 2295 Hz. Q2 could be used for transmitter (CW) keying with any modern transceiver that I've seen. The requisite software, of course, would be different from that used for RTTY. The HUG catalog lists a Morse Code transceiver program (P/N 885-1106) that runs on HDOS and one (P/N 885-8031-[37] on CP/M. All I know of any of these programs is what is stated in the catalog.

The Modem can be simplified somewhat at the cost of slightly less capability. If, for example, it is to be used only for V or UHF operation with frequency modulation, there is no need for all of the filtering provided by U1. Replace it with a single amplifier configured essentially as U1a, going directly to VR1 rather than to R5. The LED circuit also is superfluous in this case. If your SSB receiver is of recent design and replete with IF and audio filtering, it may be possible to adopt the U1 change but retaining the LED. If you choose to do this, however, remember you will no longer have the post filtered signal for the tuning monitor.

Conclusion

While not the full equivalent of some of the TUs (HAM language for "Modem") that are available, this Mighty Mini Modem doesn't cost a small fortune, doesn't take a back seat to anyone and has given creditable service on the VHF and HF bands. I know that I appreciate, in addition to its other attributes, the fact that it requires neither bench space in the radio shack nor another power supply. Anybody else make that claim? Well, I'll be "printing you" on the Bands. Call me if you need help. (73s N6UE)

Parts List

Semiconductors:

U1: LM348 quad op amp (or equiv.)
U2: EXAR XR2211 FSK demodulator/tone decoder
U3: SN74LS86 quad two input exclusive OR gate
U4: EXAR XR2206 monolithic function generator
U5: LM741 op amp (or equiv.)
Q1,Q2: 2N2222 NPN transistors or equiv.
D1,D2: 1N914 or similar signal diode
LED: Common - yellow or red, as desired.

Resistors: (all 1/4 watt 5%)

R1: 150k ohms
R2: 10k
R3: 22k
R4: 120k
R5,R8: 68k
R6,R9: 330 ohms (or 1k ohm variable - see text)
R7,R10: 43k
R11: 2.2k
R12,R13,R14,R15,R16,R20,R21,R22,R24,R33,R34,R36: 4.7K

Variable resistors:

VR1,VR2: 10k ohm
VR3,VR4: 20k ohm
VR5: 50k ohm

Capacitors: (5%, minimum 25 volt, mylar if "*")

C1,C6,C7,C11,C15: .1uf
C2,C3,C4,C5,C8: .01uf *
C12: .02uf * (parallel two .01 mylars)
C10: .1uf *
C13,C14: .005uf
C9,C17: 1uf electrolytic or tantalytic
C16: 10uf electrolytic or tantalytic

Incidentals:

2-14 pin dip headers (see Figure 2)
1-16 pin, 3-14 pin, 1-8 pin IC socket solder or wire wrap
Wire, headers and connectors as needed
Copper clad or wire wrap perfboard
Software!!!

Note: Components are readily available from a number of sources. Jameco Electronics(tm), Belmont, CA 94002, Flyer #127 listed all of the ICs for a total of $8.37. They, as well as the other components, are stocked at Radio Shack(tm) and most other general electronic parts stores.

Figure 3

Continued on Page 83

Setting The Attributes Of Your Console And Printer

William Pierpoint
1152 Harris Avenue
Camarillo, CA 93010

The TERM program by Robert W. Rasch (REMark July 1984) is a useful utility for setting the attributes of an H–19 terminal or an H–89 computer. A drawback of the program is that it supports only one argument on the CP/M command line. Since the C language easily supports multiple command line arguments, I decided to write an enhanced version of TERM in C, which is named SETCON (set the console). An even more useful program, it seemed to me, would be a program to set the attributes of my Star Micronics Delta–10 printer. This program, in which the form of the source code is nearly the same as the source of the console program, is called SETLST (set the list device) following the CP/M terminology for the printer. The SETLST.C program can be easily modified for a different printer. Both the SETCON.C and SETLST.C programs have been written to compile under the Software Toolworks C/80 version 3.0 compiler.

To use the programs, simply type the name of the program on the CP/M command line and a carriage return to get a list of the valid commands available. Two of the valid commands for the SETCON program are CLICKOFF and CLEAR, which respectively turns off the key click and clears the screen. Thus, to use SETCON to perform these actions, type on the CP/M command line:

```
SETCON CLICKOFF CLEAR
```

and a carriage return. The SETLST program is used in the same manner.

The SETCON.C Program

The SETCON.C source is given in Listing 1. Note that a main feature of the program is a structure for the attributes of the console. The structure contains three members: (1) the name of a command, (2) the escape sequence associated with the command, and (3) a brief help message describing the command. By placing all this key information in one spot, it is easy to add commands (for one of the enhanced terminal ROMs) or to change the commands (for porting the program to a different console). Just before the structure declaration is the defination of a symbolic constant for the number of commands in the structure. This also must be changed any time commands are added or deleted from the structure.

The main program is divided into three sections. The first section checks to see if there any command line arguments. If not, the program provides help by listing the commands that can be used as command line arguments, and then exits. The second section checks to see that all command line arguments are valid. A conservative method is used, in that if any command line argument is invalid, none of the commands are executed. A list of the invalid commands is provided to the user along with a list of all valid commands before the program exits. The third and last section is the heart of the program. Each command line argument (argv[cnt]) is compared (strcmp) in turn with the list of valid commands (attr[i].cmd). When a match is made, the console escape sequence (attr[i].es) is sent (fputs) to the standard output (stdout). When all command line arguments have been processed, the program exits back to the CP/M operating system.

Two functions follow the main program. Note that these are void functions. In other words, these functions return no useful information to the calling program. There is really nothing special about void functions to the compiler, since the global declaration made near the beginning of the program is:

```
#define void int
```

If the compiler is merely going to replace the word "void" with the word "int" during the preprocessor pass, you may ask why we just don't use "int" to begin with? The use of "void" is simply a reminder to the programmer that a calling program should not assign the return. Likewise, "boolean" variables make it clear that only "YES" and "NO" states are intended for their use.

The function, helpmsg, loops through all the valid commands in the attribute structure. It prints these commands (attr[i].cmd) and the associated comments (attr[i].msg) on the console.

The function fputs is used to put a string into a file. In C, a file can be a disk file or a physical device, such as the console or the printer. Three files, that are always available, are the standard input device (stdin), the standard output device (stdout), and the standard error device (stderr). The stderr device is always the console; writing error messages to stderr assures that they will not get redirected elsewhere. The stdin device is usually the console input and the stdout device is usually the console output. The stdin and stdout devices can be redirected on the command line. For instance, the command line:

```
SETCON CLICKOFF >COFF
```

will put the console escape sequence to turn the keyclick off into a disk file called COFF. The keyclick can then be turned off by the command line:

```
TYPE COFF
```

which provides exactly the same result as:

```
SETCON CLICKOFF
```

The SETLST.C Program

The SETLST.C program is given in Listing 2. It is nearly identical to the SETCON.C program. There are three areas where the programs differ.

The most noticeable difference is that the structure has a different initialization. This is to be expected since we want the structure to contain the commands, escape sequences, and help messages for the printer rather than for the console.

The second difference, to be found after the command line error checking routines, is the addition of code which checks to see if the printer is ready. The best solution would be to access an operating system function which returns the status of the printer. Unfortunately, CP/M does not provide such a function. The method I have used is a bit of a kludge, but works fine. First, a "printer not ready" message is sent to the console. Then, three nulls are sent to the printer. If the printer is not ready, the program will hang until the printer is put on line. Of course, you will know to do this since a message is on the console reminding you. If the Star Micronics Delta-10 is turned on in this situation, experience has shown that the first one or two characters will be lost. Thus, sending a few nulls initially bypasses this problem. The appropriate escape sequences are sent, and then a "printer configured" message is sent to the console. However, the "printer configured" message overwrites the "printer not ready" message. Since I am sending data at 9600 baud into the 8K buffer on the printer, the above sequence happens so fast that I never see the "printer not ready" message on the console (except, of course, when the programs hangs because the printer is indeed off line). If you detect the "printer not ready" message because your system sends data at a slower baud rate, you can put an

```
fputs("\r                    ",stderr),
```

following the statements which send the three nulls to the printer. This will effectively erase the "printer not ready" message.

The third and last difference is the use of the lputs function. It is used to directly put a string on the printer using one of the CP/M operating system functions. The C/80 compiler allows us to open a file to the LST: device, but that option was not selected for two

Listing 1

```
/*****************************************************/
/* setcon.c - sets the attributes of the h19 console */
/*****************************************************/

#define FILE int        /* usually part of stdio.h */
extern int fin,fout;
#define stdin fin
#define stdout fout
#define stderr 0

#define void int
#define boolean int
#define YES 1
#define NO 0

#define NUMCMDS 17      /* number of valid commands */

struct {
        char *cmd;      /* command name*/
        char *es;       /* terminal escape sequence */
        char *msg;      /* help message */
} attr[NUMCMDS] = {
"CLICKON","\033y2","enable the key click",
"CLICKOFF","\033x2","disable the key click",
"BLOCK","\033x4","display a block cursor",
"UNDER","\033y4","display an underline cursor",
"CURSORON","\033y5","enable the cursor",
"CURSOROFF","\033x5","disable the cursor",
"HOLDON","\033x3","enable the hold screen mode",
"HOLDOFF","\033y3","disable the hold screen mode",
"RULERON","\033j\033x1\033Y8 1  .+.. 10..  +. 20   .+. .30   .+. .40.  .+. \
50. .+  .60. .+   70. .+. .80\033k","put ruler on 25th line",
"RULEROFF","\033y1","remove the ruler from the 25th line",
"GRAPHICS","\033F","enter graphics mode",
"ALPHA","\033G","enter alphanumeric mode",
"NORMAL","\033q","enter normal video mode",
"REVERSE","\033p","enter reverse video mode",
"CLEAR","\033E","clear the screen",
"SHIFTON","\033t","shift keypad",
"SHIFTOFF","\033u","unshifted keypad"
};

main(argc,argv)
int argc; char *argv[];
{
        int cnt,i,
        boolean anyerr,match;

        /* if no command line arguments, provide help message */
        if (argc==1) {
                fputs("\nUsage: setcon cmd1 cmd2 etc",stderr);
                fputs("\nwhere the command line arguments can be:\n\n",stderr);
                helpmsg(),
                exit();
        }
```

reasons. First, C/80 would map a '\n' ('\012') into a carriage return ('\015') and line feed ('\012). This is undesirable for our case since we may wish to change the program to set a left margin or a tab at \012. Secondly, the TABS escape sequence is terminated with a null. Since the fputs function never sends a null (remember C uses the null to indicate the end of the string), the printer will never receive the end of the TABS escape sequence. The lputs function gets around this by sending to the printer both the string and the terminating null.

Avoid The Warm Boot

The C/80 compiler starts a stack which grows downward just below the CP/M basic disk operating system (BDOS) and overwrites the CP/M console command processor (CCP), which resides just below it. If interested in the details of CP/M's system organization, consult the "CP/M 2.2 Interface Guide" which is distributed with the CP/M operating system. For small utilities, such as the SETCON and SETLST programs, starting the stack below the CCP allows the warm boot to be avoided upon the completion of the program. The program simply exits back to the CCP. This modification can be made by changing the basic C/80 library program CLIBRARY.ASM. Here is how to do it (assuming C/80 version 3.0 and CP/M version 2.2).

Use your favorite editor to edit the CP/M file CLIBRARY.ASM. Near the beginning of the file, find the two lines of code:

```
$INIT     DW   0
          LHLD 6
```

Below this point insert these two additional lines of code:

```
          LXI  B,-0806H   ; start stack
          DAD  B          ; below the ccp
```

About one hundred lines later, find the line:

```
$B4       JMP  -5+5
```

Replace this line with the following code:

```
$B4       LHLD 1          , get entry point to bios
          LXI  D,-3581    ; subtract (F003H-E206H)
          DAD  D          , from bios address
          XCHG            , put result into DE
          LHLD 6          , get entry point to bdos
          MOV  A,L        ; if (HL != DE) then
          CMP  E
          JNZ  0
          MOV  A,H
          CMP  D
          JNZ  0          , warm boot   Otherwise
          LHLD 6          , get entry point to bdos
          LXI  D,-00A7H   ; enter into ccp here
          DAD  D
          PCHL            ; exit to the ccp (no warm boot)
```

That's all it takes. A check is made of the difference between the BIOS (CP/M's basic input/output system) and BDOS entry points because a resident program, such as DESPOOL or KEYMAP may have changed these. This could cause the C/80 program to jump to the wrong address, with unexpected results. If the check to the BIOS and BDOS entry points is false, then the C/80 program terminates with a warm boot. Otherwise, it jumps back to the CCP.

Compiling The Programs

To compile the SETCON.C and SETLST.C sources on the C/80 compiler, issue the following commands at the CP/M prompt:

```
C -M0 -C1000 SETCON
C -M0 SETLST
```

```
    /* check for any invalid commands */
    anyerr=NO;
    for (cnt=1;cnt!=argc;++cnt) {
        match = NO;
        for (i=0;i!=NUMCMDS;++i) {
            if (strcmp(argv[cnt],attr[i].cmd)==0) {
                match=YES;
                break;
            }
        }
        if (match==NO) {
            fputs("\nError--Invalid command. ",stderr);
            fputs(argv[cnt],stderr);
            anyerr=YES;
        }
    }
    if (anyerr==YES) {
        fputs("\nNo commands executed  Valid commands are:\n\n",stderr);
        helpmsg();
        exit();
    }

    /* execute commands */
    for (cnt=1;cnt!=argc;++cnt) {
        for (i=0;i!=NUMCMDS;++i) {
            if (strcmp(argv[cnt],attr[i].cmd)==0) {
                fputs(attr[i].es,stdout);
                break;
            }
        }
    }
    exit();
}

void helpmsg()  /* output a help message on available commands */
{
    int i;
    for (i=0;i!=NUMCMDS;++i) {
        fputs(attr[i].cmd,stderr);
        fputs(" -- ",stderr);
        fputs(attr[i].msg,stderr);
        fputs("\n",stderr);
    }
}

void fputs(str,chan)    /* put a string to a file or device */
char *str, int chan;
{
    while (*str) putc(*str++,chan);
}

#include stdlib.c
```

Listing 2

```
/*********************************************************/
/* setlst.c - sets the attributes of the Star Delta printer */
/*********************************************************/

#define FILE int        /* usually part of stdio.h */
extern int fin,fout,
#define stdin fin
#define stdout fout
#define stderr 0

#define void int
#define boolean int
#define YES 1
#define NO 0

#define NUMCMDS 12      /* number of valid commands */

struct {
        char *cmd;      /* command name*/
        char *es;       /* printer escape sequence */
        char *msg;      /* help message */
} attr[NUMCMDS] = {
"PICA","\033B\001","set pica pitch (10 cpi)",
"ELITE","\033B\002","set elite pitch (12 cpi)",
"CONDENSED","\033B\003","set condensed pitch (17 cpi)",
"CQ","\033G\033E","set correspondence quality print",
"EQ","\033H\033E","set enhanced draft quality print",
"DQ","\033H\033F","set draft quality print",
"NORMAL","\0335","set normal font",
"ITALIC","\0334","set italic font",
"NOLM","\033M\001","set no left margin",
"LM","\033M\012","set left margin at 10 characters",
"TABS","\033D\010\020\030\040\050\060\070\080\090\100\110\120\130\140\150\
\160\170\180\000","set tabs every 8 characters",
"RESET","\033@","reset the printer to power-up configuration"
};

main(argc,argv)
int argc, char *argv[],
{
        int cnt,i,
        boolean match,anyerr,

        /* if no command line arguments, provide help message */
        if (argc==1) {
                fputs("\nUsage  setlst cmd1 cmd2 etc",stderr),
                fputs("\nwhere the command line arguments can be:\n\n",stderr);
                helpmsg();
                exit(),
        }

        /* check for any invalid commands */
        anyerr = NO;
        for (cnt=1;cnt!=argc;++cnt) {
                match = NO;
                for (i=0;i!=NUMCMDS;++i) {
                        if (strcmp(argv[cnt],attr[i].cmd)==0) {
                                match = YES,
                                break;
                        }
                }
                if (match==NO) {
                        fputs("\nError--Invalid command  ",stderr);
                        fputs(argv[cnt],stderr),
                        anyerr = YES,
                }
        }
        if (anyerr==YES) {
                fputs("\nNo commands executed   Valid commands are:\n\n",stderr);
                helpmsg();
                exit();
        }

        /* reminder to user just in case printer is off line */
        fputs("\nPrinter Not Ready",stderr),
        /* send three nulls to printer */
        lputs("\000"), lputs("\000"); lputs("\000");
        /* then execute commands */
        for (cnt=1;cnt!=argc;++cnt) {
                for (i=0;i!=NUMCMDS;++i) {
                        if (strcmp(argv[cnt],attr[i].cmd)==0) {
                                lputs(attr[i].es),
                                break,
                        }
                }
        }
        /* success!  overwrites the "Printer Not Ready" message */
        fputs("\rPrinter Configured\n",stderr),
        exit(),
}

void helpmsg()  /* output a help message on available commands */
{
        int i,
        for (i=0;i!=NUMCMDS;++i) {
                fputs(attr[i].cmd,stderr);
                fputs(" -- ",stderr);
                fputs(attr[i].msg,stderr),
                fputs("\n",stderr);
        }
}

void fputs(str,chan)    /* put a string to a file or device */
char *str, int chan,
{
        while (*str) putc(*str++,chan);
}

void lputs(str)         /* put a string to the printer */
char *str;
{
        do bdos(0x05,*str); while (*str++);
}

#include stdlib.c
```

This invokes the C/80 compiler to produce the assembly language files SETCON.ASM and SETLST.ASM. The -M0 option provides an output file suitable for the Software Toolworks assembler AS, which is furnished with the C/80 compiler. The -C1000 option provides additional string space for the SETCON program, since the default string space is insufficient. The default string space is adequate for the SETLST program, so the -C option is not needed to compile it.

Once the C source code has compiled without errors, issue the following commands:

```
AS SETCON
AS SETLST
```

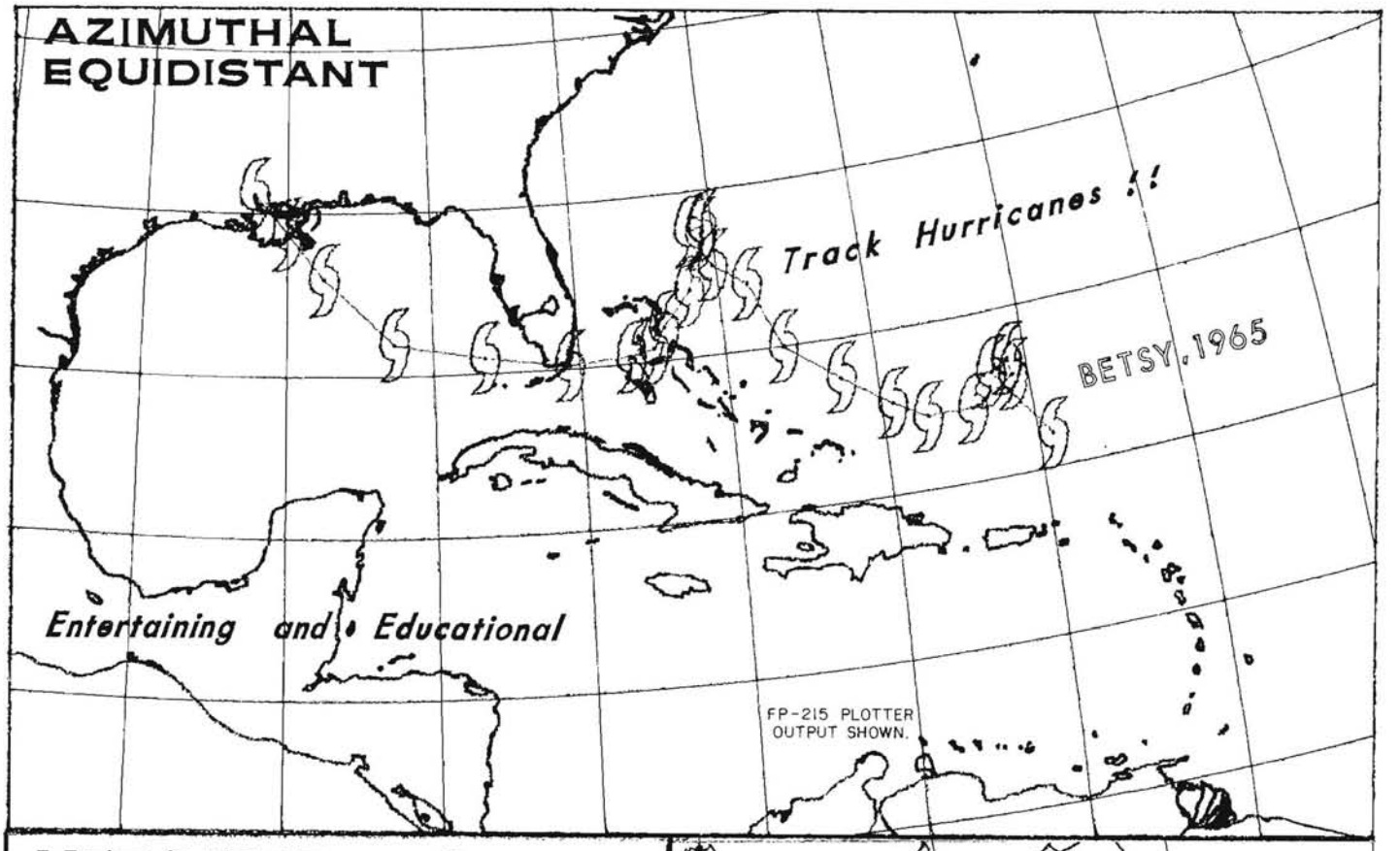
The AS compiler will compile the SETCON.ASM and SETLST .ASM files into executable SETCON.COM and SETLST.COM programs. Unlike the CP/M assembler, an intermediate HEX file is not produced. The CP/M LOAD program, which converts a HEX file to a COM file, is not required. Thus using the AS assembler saves you a step. If you made the changes to CLIBRARY.ASM described above, it is best to first copy the SETCON.COM and SETLST.COM to a scratch disk prior to running them. With all the other disks removed from your system, run the programs and make sure that they exit back to the CCP correctly.

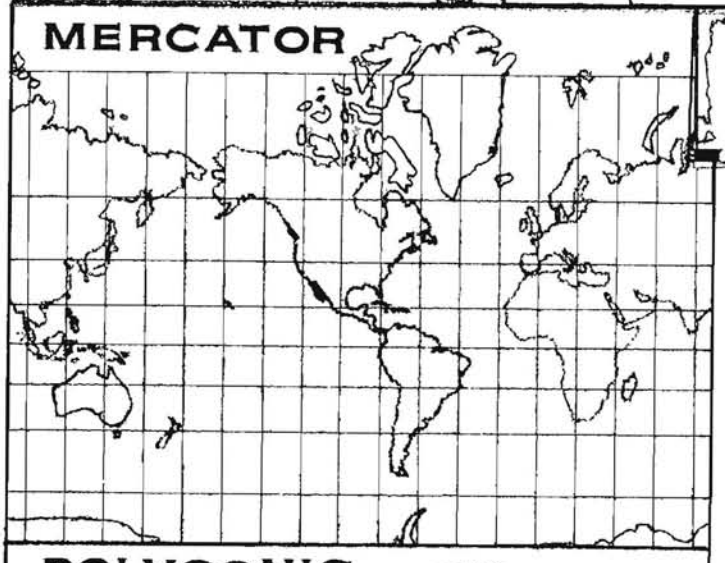
Conclusion

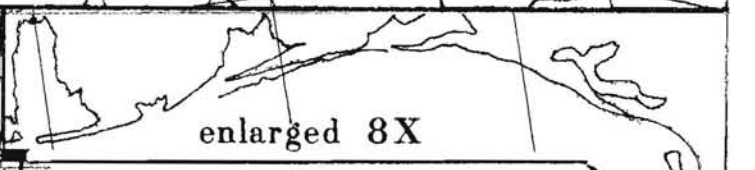
I have found the SETCON and SETLST programs to be very useful. My most common use is to use SETLST for program listings with a one inch left margin. Eighty characters will fit on one line if the elite printer pitch is chosen. The pages can be three hole punched and put into a loose leaf notebook. The command is:

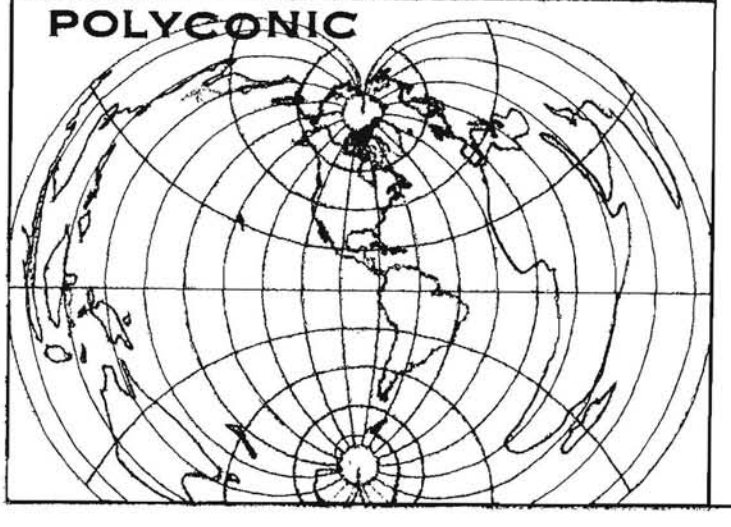
```
SETLST LM ELITE
```

Then I use the Heath/Zenith CP/M LIST utility to list the desired source file. Another common use is to set the printer for correspondence quality print before printing letters or other documents. Or for a twist, print a business letter with the italic font. I am sure you will find other interesting uses, as well. *









MAP PROJECTIONS OF THE WORLD

-Produce high-res color graphics maps for the H/Z-100 and H/Z-150 (or IBM)
-Track up to 10 hurricanes in animation Two historical ones are included
-Choose any of 7 different projections- 5 Azimuthal & Polyconic & Mercator
-Pick either of 2 data files: World Coastlines with 5070 digitized points or a very-high-res file of the Caribbean from the Carolinas to Mexico to the Amazon (11940 points)
-Vary the origin, rotation, window and scale (Global or down to 20 miles/inch on most projections)
-User's manual includes screen dumps of 16 sample maps to help you learn how to reproduce these and many others
-Superfast execution- over 1000 pts/min from an optimized compiled basic program that requires only DOS and color graphics. A color monitor does help

"...fascinating look at the world from every conceivable angle... The maps are excellently drawn. It's amazing that Dr Pittman could squeeze that much detail onto a single 5 1/4-inch diskette..."--Dave Felstul--BUSS # 91

Pick up a copy at many Heathkit Centers or order direct by sending $49.95 postpaid (specify Z-100 or Z-150) to: Dr M E Pittman, 57 Emile Av, Kenner, LA 70065

Hans-Joachim Krenz
5407-33rd Pl. S.E.
Auburn, WA 98002

TRACON

Air Traffic Control Simulation For The Z-150

"United One Five Seven, you're cleared for the ILS One Six Right Approach. Maintain three thousand three hundred until established. Tower one one eight point niner at Park."

"United One Five Seven, roger."

"Cessna November Three Niner Two Five Delta, you have traffic at two o'clock, three miles, a United tri-jet descending through three thousand five hundred on final for Seattle."

"Two Five Delta is looking."

"Roger, Two Five Delta, report United in sight and report leaving four thousand."

"Two Five Delta, roger. We have United in sight and we'll report leaving four thousand. We're climbing through three thousand now."

Air traffic control, for many people, is a mysterious, often confusing, yet fascinating activity. Most non-pilots' impressions of the air traffic control system and the people who run it are based on movies, television programs and perhaps headlines in the news media. Often, especially when Hollywood is involved, accuracy and realism take a back seat to sensationalism.

Thanks to TRACON, anyone with a Z-150 (or compatible) and an interest in air traffic control can get a basic understanding of how the ATC system really works and what it feels like to be a controller. Unlike other simulations that try to turn a PC into an airplane cockpit or a submarine control room, for example, TRACON simulates just another computer work station, that of a controller working in a Terminal Radar Approach CONtrol facility. Obviously, the potential for realism is quite high.

Today's controller rarely sees the stereotypical radar scope with its bright line sweeping a circle and illuminating "blips." Instead, he sees a computer-generated image on a circular CRT. Most aircraft using the ATC system carry a transponder. This is simply a radio transmitter that sends out one of 4096 discreet codes whenever it receives a radar signal. The transponder's signal goes directly to the ATC facility's computer which, in turn, matches that signal with information on aircraft type, destination, etc., keyed in earlier and then "paints" the airplane on the controller's CRT. Since the entire display is generated by the computer, pertinent information is also displayed next to the "blip" such as aircraft identifier, destination, and ground speed. In many cases, the transponder is even capable of communicating the airplane's altitude to the ATC computer, obviating the need for the pilot to say "altitude." In this regard, the only difference between a Z-150 running TRACON and a controller's scope is the source of the data: radar and transponder for the controller and a disk drive for the Z-150 user.

Just like a controller working at a real-life TRACON, the TRACON (simulation) controller handles arrivals, transients (aircraft flying through his airspace without landing), and departures in an airspace approximately sixty miles in diameter with a ceiling of nine thousand feet. TRACON uses the left-hand portion of the screen to display the "radar scope", a circular display with azimuth markings in 15 degree increments. The user may choose from two maps containing airways, mountains, airports and VOR's (Very high frequency Omnidirectional Ranges — aircraft navigation aids). Around the perimeter of the display are eight intersections. Arriving airplanes must be accepted before they reach these intersections inbound, and, conversely, departing aircraft must be handed off to the center controller before reaching these intersections on an outbound heading.

TRACON uses the right-hand third of the screen to display the Aircraft Preview Area. This area consists of two columns. The left column lists airplanes that will enter the radar area in the next ten minutes and the right-hand column lists aircraft currently in the radar area. Each information block consists of six parts: The aircraft identifier (TRACON uses single letters for this), aircraft type (s for single-engine, t for twin-engine, j for commercial jet transport, and f for military fighter aircraft), altitude (0-9 thousand feet), aircraft origin (either handoff intersection or airport), aircraft destination (either handoff intersection or airport), and aircraft arrival time. Thus H T 0 RP 11 would indicate that aircraft H is a twin on the ground at airport R expecting a departure clearance for intersection P at eleven minutes past the hour and J F 8 BR 13 would announce the arrival of aircraft J, a fighter, at eight

thousand feet at intersection B at thirteen minutes past the hour, landing at airport R.

Once these airplanes arrive in the radar area, they must be guided to their destination airport or handoff intersection. All incoming aircraft arrive at seven, eight, or nine thousand feet and transients, as well as departures, must be handed off to the center controller level at six thousand while arrivals must be handed off to the airport tower controller at one thousand feet. Naturally, these airplanes must be on the airway or approach and on the right course. Departures must be airborne within five minutes of their scheduled departure time or an "excessive delay" is entered into the controller's evaluation report.

Just like his counterpart working for the Federal Aviation Administration, the TRACON controller must maintain a minimum of three miles horizontal or one thousand feet vertical separation between airplanes. (TRACON does not provide for "heavy" jets, which, in reality, require five miles horizontal separation.) Aircraft must also be kept a minimum of three miles from a mountain when flying at or below the altitude of the mountain peak. If the controller loses separation, a brief alarm will sound and he will be advised that there is a "conflict" between two (or more) airplanes or that an airplane is too close to a mountain. Loss of separation is noted in the controller's evaluation report. Failure to resolve the conflict between airplanes may result in a crash and failure to turn an airplane away from a mountain will definitely result in a crash.

All aircraft entering the radar area carry a sufficient, but nonetheless, limited fuel supply. Failure on the controller's part to guide an airplane to its destination airport or intersection within the allotted time will result in that aircraft exhausting its fuel supply and crashing. Crashes are also noted in the controller's evaluation report.

Unlike the FAA controller who communicates with the airplanes in his sector by radio, the TRACON (simulation) controller types his instructions, known in air traffic control as "clearances", on his Z-150 keyboard instead of speaking into a microphone. As clearances are typed, they appear in the lower left portion of the screen. Pressing the "RETURN" key corresponds to transmitting the clearance to the airplane, and the airplane's response appears in the lower right portion of the screen. The controller has ten types of instruction to issue to aircraft in his radar area.

The first instruction, accepting a radar handoff from the center controller who controls the adjacent airspace, is issued when an airplane arrives on the edge of the scope. To accept a handoff, the controller types only the aircraft's one-letter identifier and then presses the "RETURN" key. Aircraft will not comply with clearances unless control has been accepted. If an airplane has not been accepted by the time it reaches its arrival intersection, a handoff error is noted on the controller's evaluation report.

Once the controller has accepted control over an airplane, he may issue the second type of instruction or clearance: a change in heading. Aircraft may be given headings in fifteen degree increments from 15 to 360 degrees. This is accomplished by typing the airplane's identifier, followed by an L (for a left turn) or an R (for a right turn), followed by the new heading. Thus, AL360 would be a clearance for aircraft "A" to turn left to a heading of 360 degrees, and TR030 would instruct airplane "T" to turn right to a heading of 30 degrees. A variation of this type of clearance is the instruction to execute a turn upon reaching a VOR. In this case, the controller types in a standard turn instruction but follows it with the VOR name. Thus BR045Z would mean "Airplane B, turn right to a heading of 45 degrees at Z". Airplane "B" would then proceed to VOR "Z" and upon reaching it, execute a right turn to a heading of 45 degrees without further instructions.

The third type of clearance involves altitude changes. Aircraft may be cleared to climb or descend to any altitude from one to nine thousand feet. The format for an altitude change clearance is as follows: aircraft identifier, C (climb and maintain) or D (descend and maintain), altitude (1-9). Thus W D 5 would be a clearance for airplane W to descend and maintain five thousand feet. Just like heading changes, altitude changes can be delayed until VOR passage by typing the appropriate VOR name at the end of the climb or descend clearance. C C 6 Y would, therefore, be an instruction for airplane C to climb to six thousand feet after passing VOR Y.

A very useful instruction is the "MAINTAIN" instruction which allows the TRACON "controller" to cancel turns, climbs, or descents already in progress. This is done by typing the aircraft identifier followed by the letter M (maintain) and either A (altitude) or H (heading). The aircraft will then maintain its current altitude or heading. For example, aircraft B, on a heading of 090 degrees, was instructed to turn left to a heading of 360 degrees. As the airplane is turning through 045 degrees, the controller decides that aircraft B should remain on that heading. The clearance B M H will keep airplane B on a heading of 045 degrees. The maintain instruction can also be used to cancel turns, climbs, or descents that are to be delayed until VOR passage by simply typing the appropriate VOR name at the end of the clearance. Thus airplane D, which is at six thousand feet and has been instructed to descend to three thousand feet after passing VOR Y, may be issued the clearance D M A Y. Aircraft D will then maintain six thousand feet rather than descend to three.

The fifth instruction at the TRACON "controller's" disposal is the HOLD instruction. Any aircraft may be instructed to hold at a VOR by simply typing H (hold) and the appropriate VOR name after the aircraft ID, such as F H X, which would read "Airplane F, hold at VOR X." Hold clearances are cancelled by any other instruction for that VOR.

The sixth type of instruction is the APPROACH clearance. An airplane may be cleared to execute the standard instrument approach for its destination airport by typing A (approach) and the appropriate VOR name after the aircraft ID. Aircraft cleared for the approach will leave the TRACON "controller's" control upon reaching an altitude of one thousand feet on the final approach course.

The seventh instruction, the REPORT instruction, is a very useful tool for the controller, especially during periods of high traffic volume. Aircraft may be instructed to report reaching a specified altitude, heading, or VOR by typing T (report) and the altitude, heading, or VOR after the aircraft ID. Thus H T 6 would be an instruction for airplane H to report reaching an altitude of six thousand feet. When airplane H reaches six thousand feet, a brief alarm will sound and the words "H reports 6,000 feet" will appear in the lower right corner of the screen.

Aircraft may also be requested to give current information regarding their fuel supply of heading. By typing the aircraft identifier, Q (query), followed by either F (fuel) or H (heading), the TRACON controller can find out how much fuel (in minutes) a particular airplane still has on board or what heading (in degrees) is currently being flown.

The TRACON simulation also allows for pop-ups. Pop-ups are airplanes that began their flights under Visual Flight Rules, that is not under air traffic control, and wish to enter the ATC system while en route. To help the controller identify a pop-up, a ninth command is available: I (ident). When an unknown aircraft "pops up," the controller simply types that aircraft's ID followed by I and that airplane's "blip" turns pale on the screen for a positive identification. The controller can then accept the pop-up for control.

The tenth instruction is probably any controller's favorite: the handing off of an airplane to another controller. In TRACON this is accomplished by instructing the aircraft to change radio frequency either to the center or tower frequency. This is done by typing the aircraft ID, then F (frequency change), followed by either C (center) or T (tower). (Unlike real airplanes, the ones on TRACON do not cancel IFR, and therefore, remain under control all the way to their destination or handoff.)

As can be seen from this brief description, TRACON is complex enough to provide a very good sampling of the duties of an air traffic controller. It is possible to choose from four levels of work load, two maps (and two wind directions or traffic flows), and from 32,000 "problems." In the view of this reviewer, it is sufficiently realistic to be utilized as part of a career exploration program for junior or senior high school students interested in air traffic control or aviation, in general.

TRACON does have some minor shortcomings, however. Most of these, such as the inability of TRACON aircraft to intercept airways, require more attention and concentration of the simulation controller than of his real-life counterpart. The only serious problem that, in the opinion of this writer, should be corrected involves scheduled arrivals. Every once in a while TRACON schedules two airplanes to arrive at the same intersection at the same time and at the same altitude. This represents a major error on the part of the "center controller" working the adjacent sector. Ironically, it is the TRACON simulation controller who gets charged with the separation loss, or worse yet, the crash.

Even with its minor flaws, TRACON is well worth its $37.95 price tag. It is available from CBL Marketing, P.O. Box 725, Lawrence, KS 66044. *

Performance.

Introducing EasyPC™ from UCI. The most advanced IBM PC emulator you can buy for your Z-100. First to provide the compatibility, speed and audio support to make your Z-100 run like a PC. Since the EasyPC provides complete PC circuitry, you actually run PC programs on PC hardware. Quite simply, it's like installing a PC right inside your Z-100. In the PC mode, your Z-100 can run virtually the entire library of PC programs, including LOTUS 1-2-3, dBase III, FLIGHT SIMULATOR, and most other copy-protected software. In the Zenith mode, your Z-100 is a Z-100, with no change in performance.

EasyPC is fast. In fact, it is noticeably faster than both the Z-150 and the IBM PC. And EasyPC can run at either 5 or 8 MHz. It supports both floppy and Winchester drives, provides crisp color video graphics, supports PC-compatible sound generation, and even comes with its own self mounting speaker.

Quality.

You already know and appreciate the kind of quality and reliability that is built into your Z-100. UCI matches that commitment to quality from first step to last. Sophisticated engineering and quality components combined with precision automatic assembly, component burn-in, and both in-circuit and full functional testing, mean superior performance and long term reliability.

EasyPC's advanced circuitry utilizes two, high quality, S-100 multilayer boards and a piggyback board. Yet its ingenious design requires only *one extra slot*. So you get all the advantages of PC compatibility, with room to spare.

Availability.

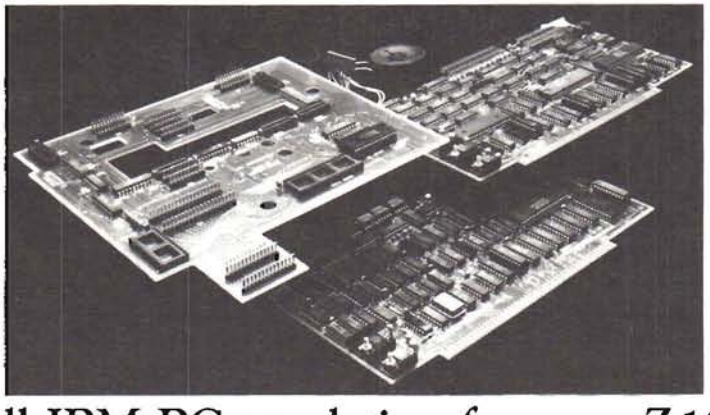
EasyPC is ready when you are. If you've been waiting too long for PC compatibility, your wait is over. EasyPC is here now. Complete documentation is included for quick and easy installation, so you'll be up and running fast. Your only problem will be which PC programs to get first.

EasyPC is available through your independent Zenith dealer. Ask him for a demonstration.

Price.

At $699, you'll pay a little more. But you'll get the best. Total compatibility. High speed performance. Unsurpassed quality and serviceability. In a word, value. You also get UCI's one year Solid Service Guarantee and full customer support. If you have any questions or problems with any UCI product, or need the name of your nearest dealer, just call 800-UCI-COMPUTER.

EasyPC™



Full IBM PC emulation for your Z-100.

UCI CORPORATION
948 Cherry Street, Kent, Ohio 44240
(216) 673-5155 / 800-UCI-COMPUTER

Q-PRO 4 Revisited

S. Edward Weidner
P.O. Box 116
Bowers Hill Station
Chesapeake, VA 23321-0116

Although a host of powerful data base management software from Ashton-Tate, Microrim, MDBS, and others continues to get the bulk of media attention, an underexposed software gem patiently awaits your discovery. Once you have mastered Q-PRO 4, which can be done with minimal effort if you possess some programming skills, don't be surprised if you find yourself asking how you ever got along without it. Since the last review of this product appeared in these pages (Dale Grundon, July 1984), a new version (3.0) has been released. With even more powerful features, such as run-time debugging and multi-key ISAM support, it deserves more than passive consideration if you're in the market for a database manager or are looking for an escape from the limitations imposed by your present software.

Written in C, Q-PRO 4 achieves remarkable portability. Configurations are available for all Heath/Zenith computers. It is available for single, multi-user, 8 and 16 bit systems including PCnet, EtherShare, Netware, PC- or MS-DOS, Multilink, MP/M-86, CMP/M-86, CP/M-86, CCP/M-86, M/PM-80, CP/M-80, TurboDos, MUSE, and Mmost, Network OS, MAXIMOS, and DNA. If you find this impressive, consider that programs (called formats) written in its procedural language are completely portable between configurations. Applications can be written on one system for use on another! True file lock and record lock commands are easily implemented in multi-user environments.

Four types of data files are supported by Q-PRO 4: multi-key indexed sequential access (ISAM), random access, sequential access, and sequential delimited files (SDF).

The analogy of a file cabinet with an indentical set of records in each drawer is a useful illustration of the differences between these data access techniques and their relative strengths. For the sake of illustration, these records are club membership records; the only difference between the drawers is how the records are arranged. The sequential file has all the data in the drawer but there are no labels on the records, they can be taken out and sorted, but during a query each one must be checked from the beginning until the desired record is located. Delimited files follow the same format except that the fields are delimited, or marked off, with quotations and the records with commas. In the random drawer records all have a numeric label in sequence. If we keep a separate index card file of, say, names and the corresponding file number, we can find the desired record more quickly than we could with the sequential method, but those handy labels are overhead, and overhead is disk space. Typically a random file takes less disk space than SDF but more than sequential files. ISAM records are labeled with the names in alphabetic order. As they are created, each new record is inserted in logically correct order (accomplished using a binary tree and hashing). Multiple keys permit the person searching the ISAM drawer to appear rather ingenious; for example, with the primary key the records are in order by name, change the key and the records are arranged in zip code order, change the key again and the arrangement is membership renewal in date order. The indexes (keys) were already a part of the records which means additional overhead, but usually much less than separate indexes and no sort time is involved.

Q-PRO 4 allows you to use any combination of these file access methods with no limit on the number files (as many as you like!)

open for read and/or write operations. You can set up your data base relationally, hierarchically, or otherwise. There are no limitations on record size or number of fields per record, but a single file may not exceed 4 Gigabytes (just in case you have the equivalent of 200 20MB hard disks!).

For ISAM and random files a program independent COBOL-style data division with up to 255 subscripted iterations per field is provided by creating stand alone FID (file item descriptor) files. This independent data definition further enhances portability and the ease of restructuring data using utilities that come with the package. Indexed sequential files permit the use of up to nine keys, with duplicate data permitted in the eight secondary keys. These file handling capabilities leave the user with the sense that his micro has become a mini. For many users, the most difficult part of learning the straightforward Q-PRO 4 system is the transition from a "micro-mentality" characterized by fatalistic resignation to exorbitant programming effort, memory-hungry programs, and dozens of accommodations to artificial limitations.

The system includes three menu-driven editors which are customized for your terminal. The user selects from among two dozen terminals (from ADM-3 to Zenith) that have been predefined and installs his terminal definition in the editors using a setup utility. If your terminal configuration is not supplied, or if you decide later that you want to reconfigure keyboard functions or CRT responses, a menu driven utility does it all, including a hard copy listing of all parameters for verification and future reference. Version 3.0 supports up to twenty-one user-defined function keys.

The editors include the FID Generator for manual creation of the data structures mentioned above (version 3.0 will optionally generate FID files automatically from a format screen), a Report Generator which will be described below, and a Format Builder which creates the heart of Q-PRO 4 applications. The Format file is analogous to a dBASE command file, but it is several orders of magnitude more powerful. A format includes screen layout, data input field definitions, and numbered program paragraphs. Anyone who has labored through screen generation in COBOL, BASIC, Pascal, or C, will suffer minor trauma here. When defining the format screen, the user fills in the screen with what will appear on the CRT at run-time including prompts and graphics (yes, highlighting and color too!); then switches the editor to define data entry fields (as many as needed), including labels, length, data type verification, justification and centering, auto-duplication, fill characters, forced entry, invisible fields, and more. For many applications, this first screen might be a "main menu" that will permit the user to chain to a number of other screens at run-time. Having completed the screen layout and field definitions, the editor can be switched again to a page editor providing for the entry of procedural paragraphs which define the application's program flow. The Format Builder's menu also provides a selection for hardcopy documentation of the format files, and direct access to the run-time system.

The extremely powerful language permits unlimited variables, registers which can be passed between formats, tables, arrays, alphanumeric labels, structured programming constructs (begin [if, while, until] end), error messages and trapping, field editing, help screens, dynamic hardcopy of the display screen, access to system date and time, julian date conversion, direct memory access, online debugging, trace, execution of system files with return, and math functions including exponentiation and logs.

The Report Generator is fast and remarkably versatile. The editor affords unlimited page length report layouts and 300 column horizontal scrolling. Field and regional breaks are easily defined. It is difficult to imagine a report that RG can not deliver. The finished report definition can be called at will and the inclusion of run-time user prompts permit dynamic report specifications. At report run-time, it sorts and merges up to six files creating reuseable cross-reference files, with output to the screen, the printer, or a disk file. System date and julian conversions, field editing criteria and logical (if-then-else) acceptance constructs are supported.

With all of these features, it is difficult to argue that Q-PRO 4 has substantive weaknesses. Certainly, by comparison with popular micro software the shortcomings are few, but the user's quest for power knows no limits. A persistent complaint has been the absence of a retrieval language operating independently on the data files; however, Q-Query has just been released. This powerful addition, complete with macro support, permits unique data collections from any number of files with output to the screen, printer, disk file, or any combination of these. Q-Query does not support modification of data, but a few lines of code in a separate format file will quickly find files matching a defined condition and make global modifications.

The Q-PRO 4 user's manual is comprehensive and well organized. Beginners will wish there were more tutorials, but the manual is much better than most in this regard. For the price of a phone call, vendor support ranks among the best in the industry. A complete sample menu-driven application on disk provides many programming and screen layout examples for the uninitiated. In summary, this is a true fourth generation applications development package which is portable and both programmer and user friendly. Though Q-PRO 4 is a relative newcomer, Version 3.0 and Q-Query are evidence of marked responsiveness to user's requests and suggestions for product improvements. With an installed base of several thousand users, user groups and third party software has begun to emerge. Many finished applications, including accounting, payroll, and telemarketing are currently available. Additionally, a number of programming utilities such as a variable cross-reference list generator and conversion program permitting editing formats with a word-processor can be obtained from third parties.

Q-PRO 4 is available from QNE International, 136 Granite Hill Court, Langorne, Pennsylvania 19047. The hotline service number (for orders or help with applications) is (215) 968-5966. An upgrade from Version 2.xx is $120. A new single user package is $595; multi-user $795. Refundable (with purchase) evaluation copies are $80. Change of operating system is $100. Minimum system requirements are 64K and two disk drives. Q-Query is $250 with evaluation copies available for $25.

About The Author

S. Edward Weidner *is a program manager for Raytheon Service Company, and resides in Chesapeake, Virginia. Experienced in programming microcomputer applications (PDP and VAX), most of his time on his H-89A is logged in Q-PRO 4 and C.*

*

Poor Man's Textronix Graphics

(Tektronix 4010 Emulation On Z-100 By FlexiTek & VGRAPH)

Jon Lee
Flight Dynamics Laboratory
Wright-Patterson AFB, OH 45433

Background

The CRT of Z-100(TM) displays 80 characters per line and 25 lines of text in the normal use. Since each character is made up of 8x9 pixels (dots), the screen resolution is nominally said to be 640 (horizontal)x225(vertical). In other words, the video can draw 225 lines of 640 dots across the CRT to fill up the screen. For a technical reason too involved to go into here (see, Chapter 4 of Technical Manual, Hardware Z-100 Series Computers), but for which we are greatly thankful, the video control circuitry does actually address 400 lines across the CRT, yet it displays only 225 of them under the normal use. An attempt to display all 400 lines of 640 pixels is then called "the interlace mode", because every other scan line is displayed in an alternating fashion. However, there is a price paid for attempting to display all 400 lines of dots. In low resolution (640x225) the entire screen is refreshed 60 times a second (assuming the usual 60 Hz setting). Whereas, in the interlace mode the alternating half of the screen is redrawn only 30 times a second. Therefore, the flickering of the CRT becomes quite noticeable unless a long-persistence phosphor CRT is used. [For further explanation, see Frank Clark's article "Interlaced Anyone?" in REMark, Vol. 5, 15, Issue 12, December 1984.]

The interlace mode is most essential when attempting to emulate a Tektronix graphics terminal because 640x225 resolution is just not good enough for Tektronix look-alike graphics. At present there are two Tektronix 4010 emulators, FlexiTek(TM) and VGRAPH(TM), fully utilizing the interlace graphics capability of Z-100. (A Tektronix emulator EMU TEK of FTG Data Systems developed for IBM PC may be used for Z-100, but it is strictly of the low resolution type.) We shall first explain briefly the conceptual working of a generic Tektronix emulator (Section 2) and then compare FlexiTek and VGRAPH in major functional categories (Section 3). Lastly, the graphics quality of FlexiTek and VGRAPH will be compared with the actual performance of a real Tektronix 4010 (Section 4).

Our purpose here is two-fold: One is to guide those contemplating Tektronix graphics emulation on a Z-100, and the other is to provide software vendors with the feedback of a graphic benchmark test. To establish a base line, it is assumed that Z-100 is equipped with video RAM 32K and operating under ZDOS, a standard microcomputer acquired through the requirement contract of AFSCOASO (Air Force Small Computer and Office Automation Service Organization). Furthermore, the OKIDATA(TM) microline 83A is fitted with OKIGRAPH I graphic board.

Functional Schematics

Prior to delving into the detailed description of each emulator, it is perhaps more instructive to examine the functional configuration of figure 1, depicting the controls that a generic Tektronix emulator has over the host-terminal communication and the peripherals of Z-100.

To be concrete, we shall assume that the host is Flight Dynamics Lab's VAX/VMS and the graphics data generated by Precision Visual's DI-3000 have been formatted into 1024x1024 pixels instructions by the Tektronix 4010 driver. Since one usually sees only 780 lines on a rectangular CRT, we shall henceforth say that the dot resolution of Tektronix 4010 is 1024x780. When Z-100 is booted up with FlexiTek/VGRAPH, ZDOS turns the control over to the emulator. Entering into host-terminal communication, the 1024x780 graphics data are brought down to the Z-100. Of course, not all 1024x780 graphics data can be displayed because the CRT resolution of Z-100 is only 640x400 in the interlace mode. Nonetheless, the full 1024x780 graphics data can be captured (saved) into a disk, which may later be replayed (redisplayed) on CRT without the host intervention. The CRT graphics may be routed to a dot matrix printer. As it turns out, the screen dump on OKIDATA can retain just about the same resolution as the CRT display. This is because OKIDATA has 60x66 dot resolution and FlexiPrint(TM) screen dumps sideways into a 9.5"x 6.5"

frame, so that one can estimate the overall dot resolution to be 630x390 [roughly equals to (9.5x66)x(6.5x60)]. Finally, the graphics data consisting of Tektronix's end-points of lines and beam turns on/off, can be converted into appropriate plotter instructions of pen move, pen up/down, etc. Since an HP plotter has the incremental pen movement of less than one mil, the plotter hard copy is a true representation of the 1024x780 graphics data.

Flexitek And VGRAPH

The FlexiTek is very simple to use. It is completely run by function keys which have been pre-defined as shown in figure 2a. Although redefining the function keys is possible through the .ASM files provided, it is certainly not for the fainthearted. On the other hand, the VGRAPH has its own command language. For instance, one must type in EMULATE TERM=T4010 HOST=COMM to enter the Tektronix emulation mode, and similar command strings exist for capture, replay, etc. Obviously, typing in long strings of VGRAPH commands is cumbersome not only for a frequent user (because it is too much work), but for an occasional user (because it is impossible to know the command syntax after a long hiatus). However, do not despair, for the VGRAPH will let you convert the keyboard into a "smart" keyboard so that a string of VGRAPH commands can be invoked by a single key stroke. Figure 2b displays our rendition of function key menu to enter/exit Tektronix emulation, capture a graphic file with an inserted PAUSE, and replay the captured graphics file in local mode. Note that the key menu contains only what we feel is necessary for routine tasks, but the sign-on file (SIGFILE.VTS) has all other 4010 and CRT terminal key definitions (see the Appendix). In this way, one can run VGRAPH completely by the function keys without constantly referring back to the manual.

As a users' guide, we shall now compare what FlexiTek and VGRAPH can/cannot do and how they measure up to one's expectation. Though subjective, we shall also point out features that are surprisingly pleasing. However, the discussion on graphics quality of emulators will be relegated to the next section.

Graphics Benchmark

The graphic benchmark test consists of the text drawn in by DI-3000's graphic arts precision text (JHTEXT) with different fonts and the line drawing of a spiral in a circle. The Z400DO.COM of FlexiPrint is used for the screen dump of both FlexiTek and VGRAPH as shown in figures 3 and 4, respectively. As already pointed out (Section 2), the screen dump on OKIDATA does portray the CRT graphics quite accurately because of the comparable dot resolution. Since the printer screen dump is meant for qualitative reference only (as it is after all a "quick-and-dirty" hard copy), we shall not be concerned with the graphics quality of figures 3 and 4.

On the other hand, FlexiTek can generate a more refined hard copy on a HP plotter. That is, the PZ.COM of FlexiTek can convert the captured graphics file containing Tektronix's end-points of lines and beam turns on/off, into HP plotter's instructions of PA, PU/PD, etc. Figure 5 is a plotter hard copy of HP 7475A under low speed (VS 17) and baud rate of 2400. As expected, the text and line drawing of plotter hard copy (figure 5) are much smoother than that of printer hard copy (figures 3 and 4). Notice however the following detail; the circle containing spiral is not exactly round, i.e., the equator is about 1.6% longer than the North-South pole. Although such a distortion is perceptively unimportant, one should be aware of it in scientific use. Now, here is a

Table I: Emulator Comparison

Emulator	FlexiTek	VGRAPH
Test version	Version 1.2	Version 2.0
Programming language	source code in assembly language, except for the plotter routine in C All source code provided	Source code in C-language No source code provided
Readability of the manual	The manual is brief and gives simple steps to follow through for all functionalities of FlexiTek Particularly interesting to a new user are the Tektronix 4010 description and the hooking-up tips for host-terminal communication and peripherals	The manual has two main parts The first part gives the step-by-step procedures for entering emulation, capture, and replay The second part presents VGRAPH commands for converting the keyboard into a "smart" keyboard, whereby certain escape sequences are effected by a single key stroke Although this part of the manual tends to be a bit technical, a complete mastery of VGRAPH commands is rewarding
User friendliness	The user interacts via the function keys which have been pre-defined as shown in figure 2a Although redefining the function keys is possible through the .ASM files, it is not for everybody	Through the definition of the sign-on file (SIGFILE.VTS), the user can interact with VGRAPH by the function keys as exemplified by figure 2b All the user's parameters can be entered as defaults In other words, one can tailor SIGFILE VTS to meet one's exact specifications This is the most powerful feature of VGRAPH
CRT resolution with video RAM 32K	The claim is 640x400 in the interlaced graphics mode	In the interlace GRAPHICS MODE 0, the vertical resolution is one of the installation parameters Note that 380 is as high as we can go
Alpha mode	35 lines of text in a crisp Roman font	42 lines of text in a type font
Flicker	The degree of flickering is directly proportional to the vertical resolution That is, the flicker of the standard CRT becomes more noticeable, as one attempts to draw in more and more scan lines Of course, no flicker is observed in the low resolution limit of 640x225 graphics mode, and GRAPHICS MODE 1 of VGRAPH will give you just that In the high-resolution mode, it appears that VGRAPH's graphics suffers a little more flicker than FlexiTek's, which is presumably due to the increased vertical resolution. In the Alpha mode, VGRAPH's text screen does flicker considerably more than FlexiTek's.	

HUG Members only!

HUGE HUGGER DISCOUNTS now available on these selected products...exclusively for members only!

That's right. We're offering HUG members these close-out, last call and limited quantity specials from Heath Company.

The items listed are not necessarily representative of complete systems. Please call your nearest Heath/Zenith Computers & Electronics Center or Heath Company toll-free at 1-800-253-7057 for more information before ordering.

Watch for additional MEMBER ONLY discounts in upcoming issues of REMark®!

REMark Discount List

	ITEM	LAST ADV'D PRICE	HUG MEMBER PRICE	QUANTITY LEFT
H-19-3	H-19 to H-88 Mod Kit	220.00	199.95	44
H-88-1	Floppy Disk Accessory	150.00	99.95	140
H-88-5	Cassette I/O Card	25.00	24.95	86
H-88-9	Drive Mounting & Inst.	25.00	24.95	80
H-125	Dot Matrix Printer	750.00	499.95	33
HCA-5-77	H-77 Dust Cover	14.00	9.95	7
HCA-5-89	H-89 Dust Cover	14.00	9.95	46
HCA-9	Print Stand For H-125	99.00	89.95	58
HDC-100	H/Z-100 Dust Cover	16.00	9.95	207
HDC-125	H/Z-125 Dust Cover	16.00	9.95	157
PC-140	DG Magic RAM Expn OK	129.00	99.95	8
PC-300	MMS CP/M+ & 128K Mem.	395.00	395.00	2
PC-310	DG Super '89 CPU	699.00	499.95	36
PCA-120-1	6-Pak Plus Game Acc'y.	39.95	34.95	34
PP-110	Trippe Emer. Power System	349.00	299.95	6
PP-260	DTC 380Z Daisy Printer	1295.00	899.95	15
PPA-260-2	Cut-Sheet Feeder /380Z	574.95	399.95	3
PPA-260-3	Ribbons (3PK) /380Z	9.95	5.99	264
PPA-260-4	Cable - RS232 /380Z	49.95	34.95	29
PPA-260-5	Cable - Parallel /380Z	49.95	34.95	27
PPA-270-1	Trac Feed - Stylewriter	149.95	99.95	13
PPA-270-2	Sheet Feed - Stylewriter	249.95	174.95	50
ZA-100-4	S-100 Extender Board	59.00	29.95	417
ZA-219	Amber CRT /H-121	75.00	19.95	664
ZG-219	Green CRT /H-121	75.00	19.95	206
ZW-219	White CRT /H-121	75.00	14.95	165

Heathkit®

Heath Company

A subsidiary of Zenith Electronics Corporation.
Some items may not be available at all store locations
Prices, product availability and specifications subject to change without notice

CP-236

	This is, however, unavoidable because VGRAPH puts out 20% more text lines than FlexiTek.	
CRT resolution with video RAM 64K	No new features supported	Increased vertical resolution of 640x512; FAS50 displays 50 lines of text(*)
Color video	Seven colors supported by Z-100(*)	
Capture and replay	Both can save the full 1024x780 Tektronix's end-points of lines and beam turns on/off instructions on a disk, to be replayed later on without the host intervention.	
	Replay of one captured graphic frame only	VGRAPH can display a sequence of captured graphic frames interspersed by a PAUSE/DELAY in the form of a "slide show"
Screen dump on dot matrix printer	With FlexiTek, you also receive FlexiPrint which has three screen dumps, 225-line, 400-line, and three-pass grey, for nearly all name-brand printers. The array of supported printers is indeed impressive The 400-line mode is for screen dumping high resolution graphics (figures 3 & 4), and the 225-line mode is for the ordinary text dump (figure 2)	Supports Epson FX-80 and few others, but not OKIDATA
Hard copy on plotter	HP plotter	No plotter support
CRT mode	For extensive text editing, both FlexiTek and VGRAPH offer a dumb terminal mode but with the normal screen text of 80 characters and 25 lines	
GIN mode	Both support slow and fast cross-hair movements	
Emulator speed	As a typical scientific application, the host VAX integrates a set of five non-linear ordinary differential equations over a pre-determined time interval Afterwards, DI-3000 graphics package displays the numerical result as five separate time-history plots, all drawn in the same screen As expected, both the run time for integration and the plot time for display vary greatly depending on the VAX loading. We have therefore run the program 10 times but retained only five of them with the least run and plot time for averaging The results are;	
	Run time = 87.4 sec Plot time = 15.2 sec	Run time = 85.2 sec plot time = 15.8 sec
	Neither of the emulators exhibits a clear-cut edge in speed For completeness, we have also run the test program on a real Tektronix 4010 with the following results; run time = 84.4 sec, plot time = 17.2 sec. Curiously, the real Tektronix terminal took nearly 10% longer to plot than the emulators	
Replay speed	The graphics data of figures 3 and 4 have been captured by FlexiTek and VGRAPH, respectively The captured graphics file of FlexiTek is then replayed by FlexiTek, and the captured file of VGRAPH by VGRAPH. Unlike the emulation speed, the replay speed is more or less constant within the clocking accuracy of, say, within 1/2 sec. The average over 3 trials gives;	
	Replay time = 12.3 sec	Replay time = 12 7 sec
	Here, again, no significant difference is observed Interesting to note that FlexiTek reads in all graphics data(about 4.5 sec) before plotting them, whereas VGRAPH starts to plot immediately while still reading in the graphics data	
Source	VariousWare, P.0 Box 21070 El Cajon, CA 92021 Price $129.95 includes FlexiTek and FlexiPrint	CompuView Products, Inc 1955 Pauline Blvd, Ann Arbor MI 48106 Price $120

(*) Our video RAM is only 32K and monochromatic, hence has not been tested

pleasant surprise; the captured graphics file of VGRAPH can also be plotted out by the PZ.COM of FlexiTek, yielding an identical plotter hard copy as figure 5.

To assess the graphics quality of emulators, we have redrawn the same graphic benchmark test on a real Tektronix 4010. Figure 6 is a screen dump by Tektronix 4631 hard copy unit. Here, the elongation of circle (3.6%) is worse than in figure 5 (1.6%). Although, in principle, the CRT of Tektronix 4010 and Tektronix 4631 hard copy unit can each be adjusted for correct aspect ratio, it is perhaps not feasible to sustain the drift in aspect ratio much less than 1.6% on a day-to-day operation. In this regard, the performance of FlexiTek is quite satisfactory. Finally, notice that the inner spiral of figure 6 is as wavy as in figure 5. This is a reflection of the limiting resolution of Tektronix 4010 and not a failing of emulators.

Recommendations

The emulator comparison of Table I is detailed enough to provide necessary guidance for a right choice of Tektronix emulator, depending on the specific applications in mind. Summing up, if printer/plotter hard copying is a requirement, FlexiTek-Print is the clear choice (you get the hard-copying capabilities at a small additional cost). Otherwise, the choice of FlexiTek and VGRAPH is a toss up. However, our choice would be VGRAPH because it is a more versatile <emulator> than FlexiTek. One can readily configure VGRAPH to accomodate the specific individual needs, whereas FlexiTek is pre-configured more or less for simple-minded users. Put it differently, terminal emulation software is a black box to an average user(not a programmer). VGRAPH offers a few knobs to tweak; FlexiTek is completely sealed.

There is however an important factor left out of Table I, that is, the customer technical support. Our experience has been that VariousWare is just excellent on this service which sometimes is more important than the software itself.

Acknowledgements: I wish to thank Graham Wideman of VariousWare for his help in running FlexiTek and for his detailed response to the first draft. Also, the help of James Marks of CompuView Products on the VGRAPH command language should be acknowledged.

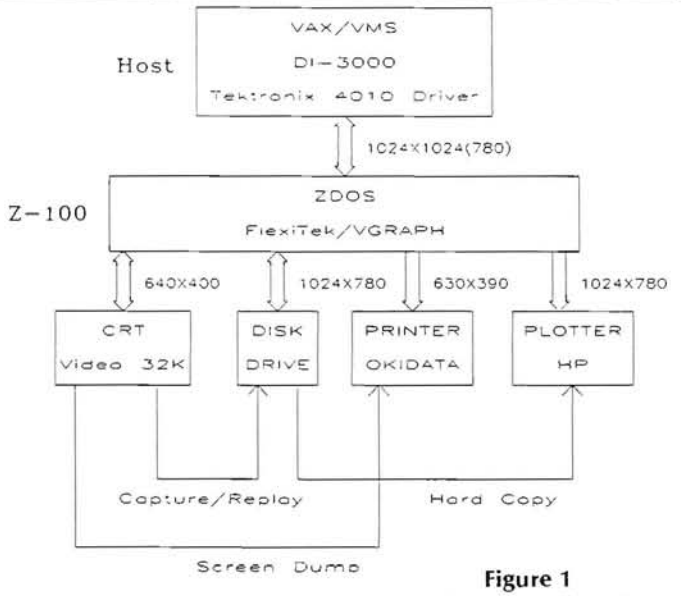


Figure 1
Functional Configuration

```
                    FlexiTek Menu

F1: Display Menu
F2: Toggle emulation mode: TERM / LOCAL          TERM mode enabled
F3: Change terminal BAUD RATE                    Baud rate= 300
F4: Change terminal PARITY                       Parity= NONE
F5: Change terminal DATA LENGTH                  Data Bits= 8

F6: 4010 PAGE key                                Stop Bits= 1
F7: Select Disk function
F8: Enter emulation mode
F9: Close capture buffer
F10: Exit to DOS

Shift-F1: Set Duplex                             Duplex = FULL
Shift-F2: Set Gin terminator                     Terminator = NONE
Shift-F3: Set CR effect                          CR Effect = CR
Shift-F4: Set Emulation                          Emulation = 4010
```

Figure 2A
Flexitek's Key Menu

```
                    Key Menu Display

F0:      Enter Tektronix mode                    (from Command mode)
Sht-F0:  Enter CRT mode                          (from Command mode)
F1:      Exit emulator                           (from emulation mode)

*** Tektronix must be ready to draw a picture with a (CR) ***
F2:(1)   Enter capture-file name                 (from emulation mode)
F3:(2)   Execute capture                         (from emulation mode)
F4:(3)   Close capture-file                      (from emulation mode)
*** Get out of replay mode by I-CHR key ***
F5:(1)   Enter replay-file name                  (from command mode)
F6:(2)   Execute replay                          (from command mode)

F7:      Full/Half duplex                        (from emulation mode)
F8:      On/Off line                             (from emulation mode)
F9:      Baudrate = 1200                         (from command mode)
F10:     Baudrate = 9600 (default)               (from command mode)
F11:     Close capture                           (from command mode)
I CHR:   XON/XOFF toggle                         (from emulation mode)
INSLINE: Exit to DOS (stop)                      (from command mode)
HELP:    Key menu display                        (from command mode)
HOME:    4010 page key                           (from emulation mode)
```

Figure 2B
A Typical Function Key Menu For VGraph

APPENDIX: Sign-on File SIGFILE.VTS for VGRAPH

```
;*****************************************************************
;       Signon File for the Virtual Terminal System on Z-100      *
;             Modified by Jon Lee,  25 October 1984               *
;*****************************************************************

set baud =9600                  ; 9600 is the default baudrate
define '\026'="\xFF"            ;  Z exits emulator
define CSC1 = '0'               ; First common character is 0
define CSC2 = '1'               ; Second common character is 1

; Define 4010 and CRT terminal function keys`

define leadin 'J' = "emulate host=comm\r"   ; F0 = Enter emulation mode
define leadin 'E' = "emulate term=crt host=comm\r", Sft-F0 = Enter CRT
define leadin 'S' = "\255"                  ; F1 = Exit emulator
define leadin 'T' = "\248\255\source \"mesg1\"\r\Set Capture=\""
                                            ; F2 = Initiate capture
define leadin 'U' = "\"\r emulate\r\252\249"  ; F3 = Execute capture
define leadin 'V' = "\248\254\240\251\255Emulate Capture=Dummy\r\249"
                                            ; F4 = Close capture
define leadin 'W' = "source \"mesg2\"\r\Set Replay=\""
                                            ; F5 = Sets up replay-file
define leadin 'P' = "\"\r emulate host=replay\r"
                                            ; F6 = Execute replay
define leadin 'Q' = "\244"                  ; F7 = Full/Half duplex
define leadin 'R' = "\241"                  ; F8 = On/Off line
define leadin csc1 'I' = "Set baud=1200\r\Display comm\r"
                                            ; F9 = Set baud = 1200
define leadin csc1 'J' = "Set baud=9600\r\Display comm\r"
                                            ; F10 = Set baud = 9600
define leadin csc1 'K' = "Set capture=dummy\r", F11 = close capture
define leadin csc1 '@' = "\247"             ; I CHR = Xon/Xoff toggle
define leadin 'L' = "stop\r"                ; INSLINE = Exit to DOS
define leadin '1' = "\249"                  ; ESC 1 = set XON
define leadin '2' = "\248"                  ; ESC 2 = Set XOFF
define leadin '3' = "\243"                  ; ESC 3 = On line
define leadin '4' = "\242"                  ; ESC 4 = Off line
define leadin '5' = "\164"                  ; ESC 5 = Margin toggle
define leadin '~' = "source \"keymenu\"\r"; HELP = Key Menu display
define leadin '|' = "\239"                  ; Shift Break = break
define leadin 'H' = "\161"                  ; HOME is the page key
define leadin 'A' = "\165\020"              ; UP ARROW    = cursor up 20
define leadin 'B' = "\166\020"              ; DOWN ARROW  = cursor down 20
define leadin 'D' = "\167\020"              ; LEFT ARROW  = cursor left 20
define leadin 'C' = "\168\020"              ; RIGHT ARROW = cursor right 20
define leadin csc2 'J' = "\160"             ; Shift F10 = Reset to Alpha
define leadin '8' = "\169"                  ; ESC 8 = No report signature
define leadin '9' = "\170"                  ; ESC 9 = Carriage Return sig
define leadin '0' = "\171"                  ; ESC 0 = C.R  plus EOT sig

SET EXITSTRING=">"    ; Return to normal keypad when you leave VGRAPH
```

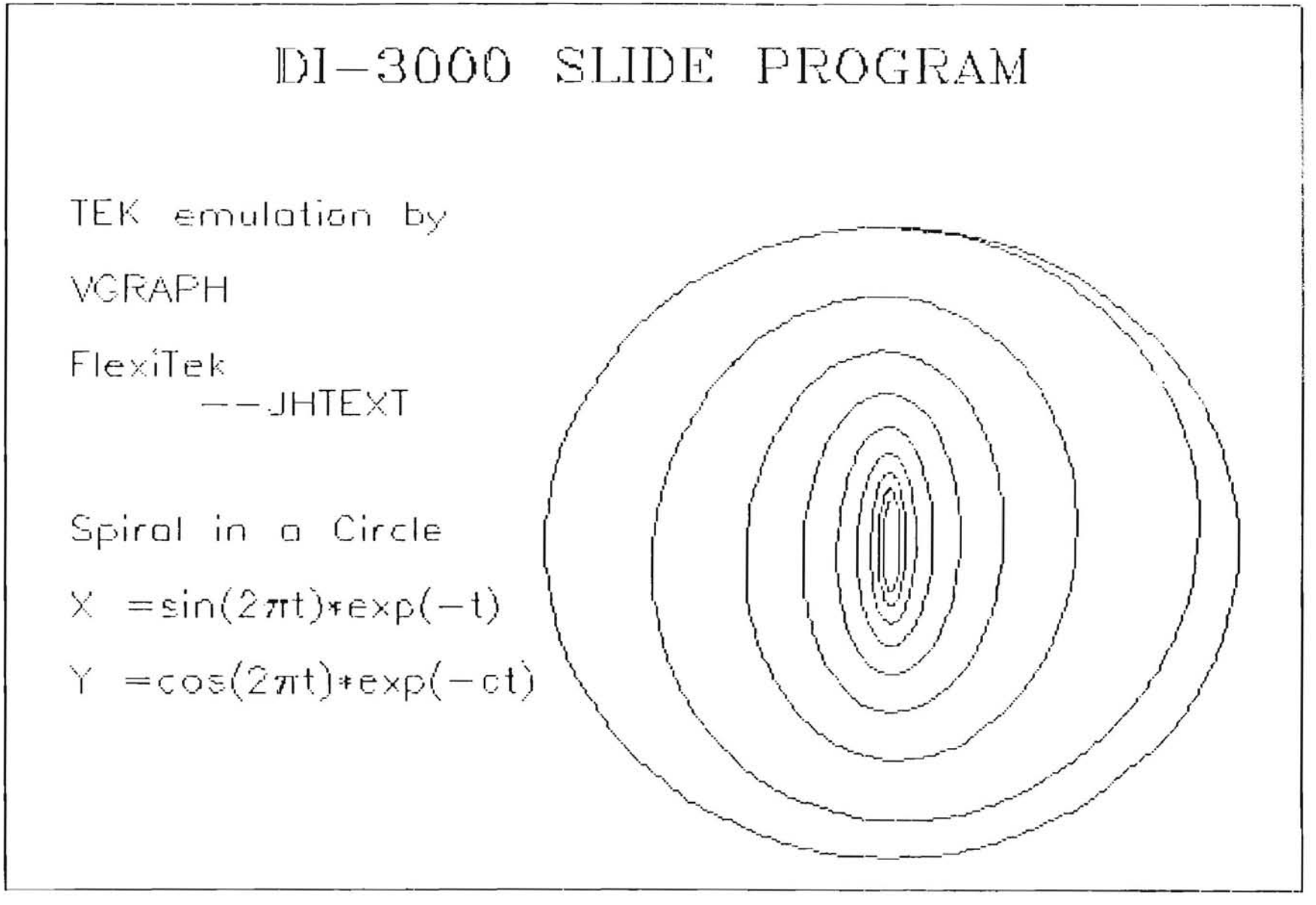


Figure 3
Screen Dump Of Flexitek

Ultimate Communications Software

HyperACCESS™

A Dramatic Step Beyond ACCESS™†

HyperACCESS gives you tremendous power—power that lets you communicate without getting technical—power that lets you glide through applications other programs can't touch.

A New Standard for Ease of Use

A startup program sets up everything for you and a fascinating on-screen tutorial lets you try out program features in simulated use of Compuserve, MCI Mail, Easylink, and others. HyperACCESS menus are instinctively easy to use, so you can focus on your work, instead of on how to run the program.

Effortless Modem Support

Instantly call any remote system by selecting it from a menu that lets you list up to 127 systems and already lists Compuserve, The Source, MCI Mail, Easylink, and other systems (subscriptions not included). The first time you call these systems, your password and user ID are memorized, and they are entered for you automatically, thereafter.

Remarkably Flexible

HyperACCESS comes ready-to-use for the more popular applications and it lets you set and store all the details you need, to connect with main frames, minis, micros, or RS232-compatible devices that defy other programs.

Terminal Emulation

HyperACCESS emulates the most popular terminals, including DEC VT-52, VT-100, TeleVideo 900 series, IBM 3101, and H-19, to let you run programs on remote computers designed for use from terminals.

Use Your Computer's Power

From within HyperACCESS, you can use your computer's DOS commands (DIR, COPY, etc.) or run your other programs. You can even define DOS macros, so you can execute a series of DOS commands and/or programs by pressing a single key.

Use Xmodem, Kermit, or Text Transfer Methods

. . . to reliably send/receive files of any kind from/to any disk drive or directory in your computer, even through noisy telephone lines.

Remote Use of Your Computer

Press one key and, even when you're absent, operators of remote terminals or computers can access your computer by entering passwords you define. You can set whether each password allows a user to read or type messages, transfer files, run a given program, or use your operating system and all your programs.*

Automate Your Communications

. . . so you can perform, with a keystroke, steps that normally would take many keystrokes. HyperACCESS has the clearest, most comprehensive script language available, to make automating communications easy for beginners and exhilarating for experts. You can automate any part of a call, make unattended calls, or create custom functions, prompts, and menus.

For astonishing power and versatility, get HyperACCESS today!

For PC-compatibles and Z-100 computers with DOS and any modem, including 2400 baud and USR S100 modems. You can get HyperACCESS for $149 from computer dealers or from Hilgraeve Inc. (please specify computer and add $5 shipping).

Hilgraeve Inc.

P.O. Box 941
Monroe, MI 48161
(313) 243-0576

VISA

Z-100 Version fully supports Gemini and EasyPC Boards

† Hilgraeve and its dealers will continue to sell ACCESS, as well.

* Certain programs cannot be operated remotely.

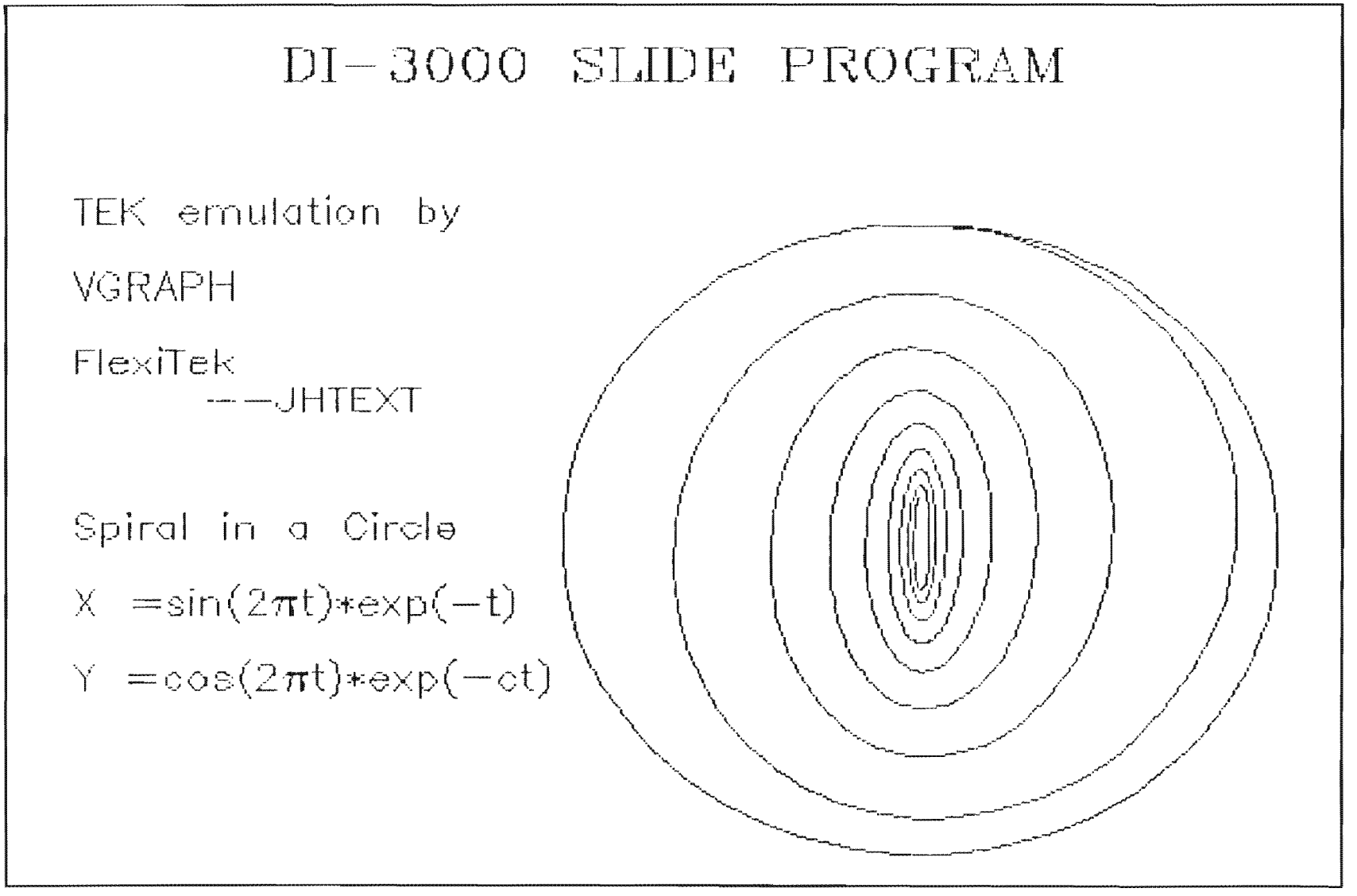


Figure 4
Screen Dump Of VGraph

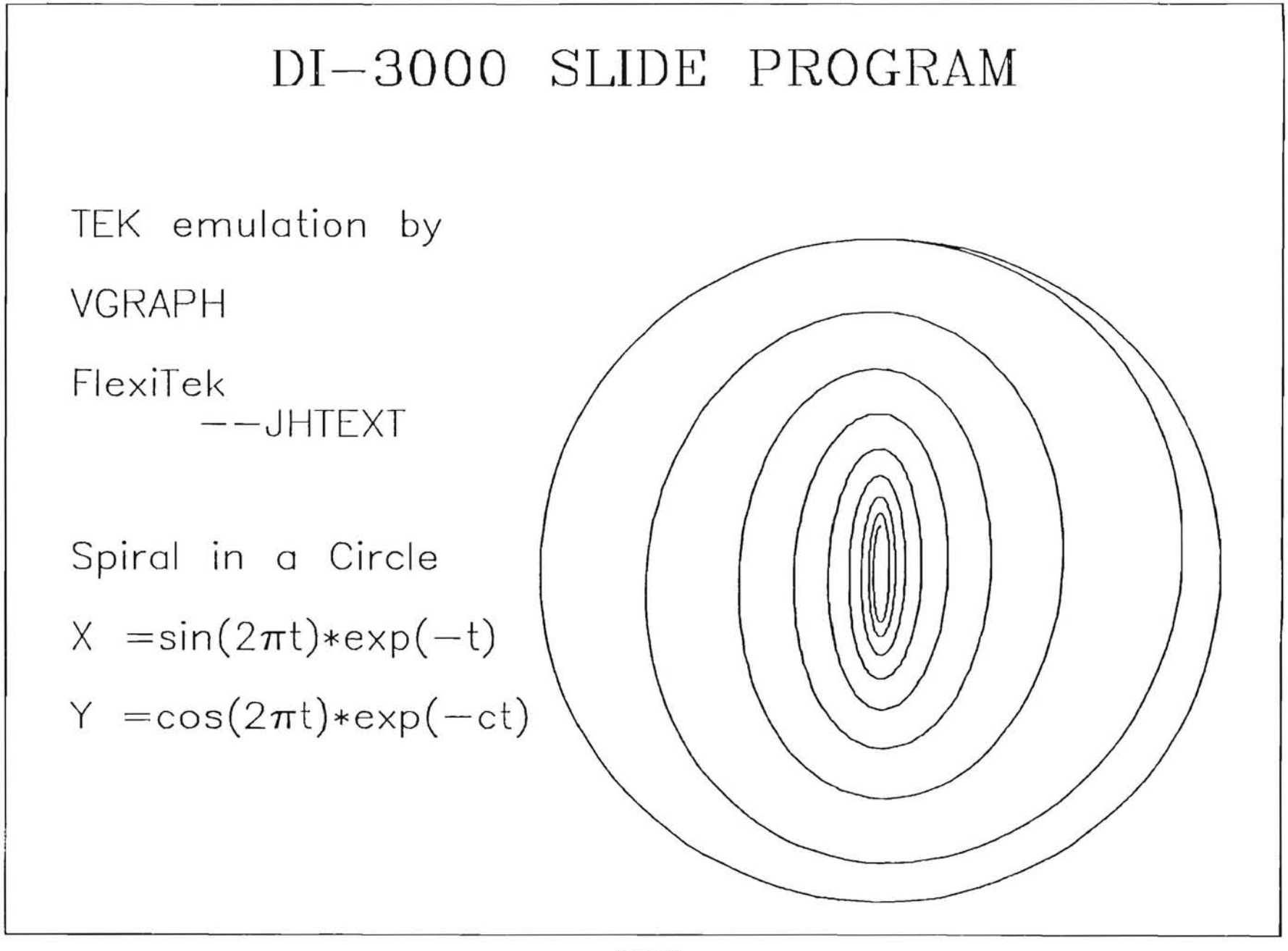


Figure 5
Hardcopy By HP7475A Plotter

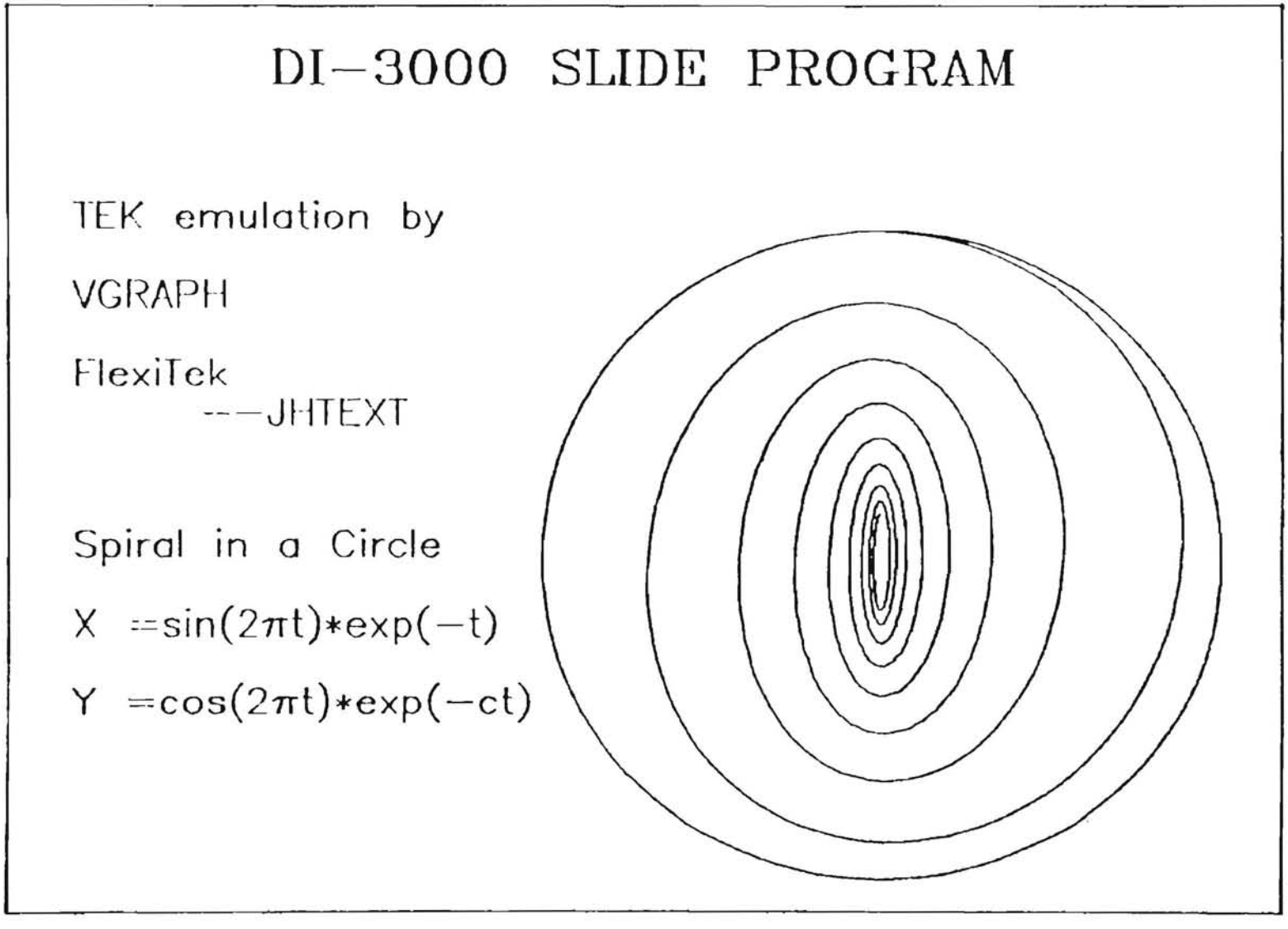


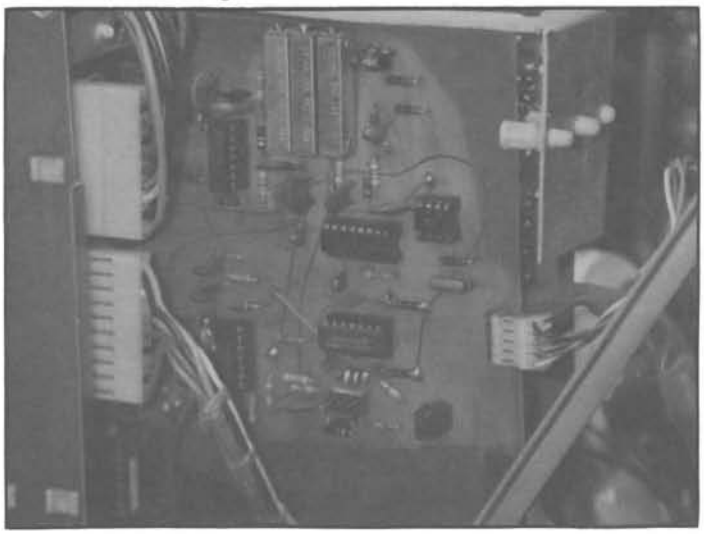
Figure 6
Hard Copy Of Tektronix 4010 Screen

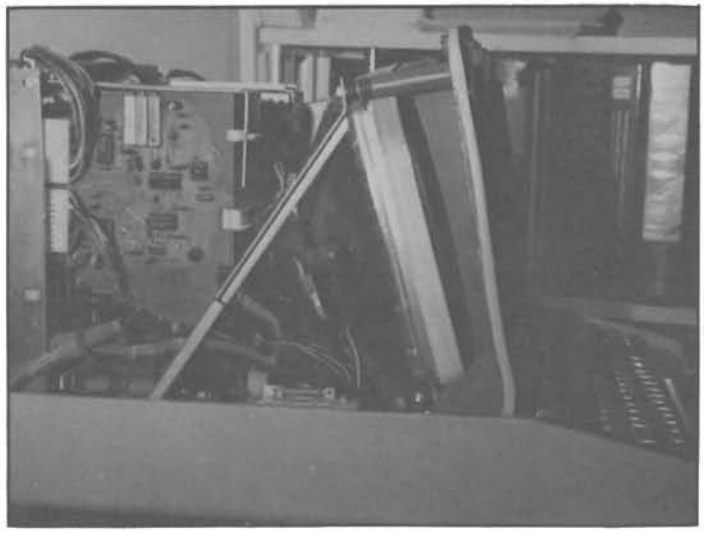
Continued from Page 41

Part Number	Description of Product	Selling Price	Vol. Issue
DATA BASE MANAGEMENT SYSTEMS			
HDOS			
885-1107-[37]	HDOS Data Base System H8/H89	30.00	23
885-1108-[37]	HDOS MBASIC Data Base Sys.	30.00	23
885-1109-[37]	HDOS Retriever ASM (3 Disks)	40.00	23
885-1110	HDOS Autofile (2 Disks)	30.00	23
885-1115-[37]	HDOS Navigational Program	20.00	25
885-8008	Farm Accounting System	45.00	30
CP/M			
885-1219-[37]	CP/M Navigational Program	20.00	31
MSDOS			
885-8034-37	DBZ-A Database For The Z100	25.00	69
AMATEUR RADIO			
HDOS			
885-8016	Morse Code Transceiver Ver 2.0	20.00	42
CP/M			
885-1214-[37]	CP/M MBASIC Log Book (64k)	30.00	23
885-1234-[37]	CP/M Ham Help	20.00	49
885-1238-[37]	CP/M Ascirity	20.00	57
885-8020-[37]	CP/M RF Comp. Aided Design	30.00	44
885-8031-[37]	CP/M Morse Code Transceiver	20.00	57
COMMUNICATION			
HDOS			
885-1122-[37]	HDOS MicroNET Connection	16.00	37
CP/M			
885-1207-[37]	CP/M TERM & HTOC	20.00	26
885-1224-[37]	CP/M MicroNET Connection	16.00	37
885-3003-[37]	CP/M ZTERM (Z100 Modem Pkg)	20.00	34
885-5004-37	CP/M-86 TERM86 and DSKED	20.00	56
885-5005-37	CP/M-86 16 Bit MicroNET Conn.	16.00	61
885-5006-37	CP/M-86 HUGPBBS	40.00	62
885-5007-37	CP/M-86 HUGPBBS Source List.	60.00	62
885-8005	MAPLE (Modem Appl. Effector)	35.00	29
885-8012-[37]	CP/M MAPLE (Modem Program)	35.00	34
885-8023-37	CP/M-85 MAPLE	35.00	45
MSDOS H/Z100 - H/Z150 PC			
885-3019-37	ZDOS 16 Bit MicroNET Connect.	16.00	61
885-3027-37	MSDOS HUG PBBS	40.00	66
885-3028-37	MSDOS HUG PBBS Source Listing	60.00	66
MISCELLANEOUS			
885-0004	HUG Binder	5.75	
885-1221-[37]	Watzman ROM Source Code/Doc	30.00	33
885-4001	REMark Vol. I Issues 1-13	20.00	
885-4002	REMark Vol. II Issues 14-23	20.00	
885-4003	REMark Vol. III Issues 24-35	20.00	
885-4004	REMark Vol. IV Issues 36-47	20.00	
885-4005	REMark Vol. V Issues 48-59	25.00	
885-4500	HUG Software Catalog	9.75	
885-4600	Watzman/HUG ROM	45.00	41
885-4700	HUG Bulletin Board Handbook	5.00	50
885-3015-37	ZDOS Skyviews	20.00	55

NOTE: The [-37] means the product is available in hard sector or soft sector. Remember, when ordering the soft sectored format, you must include the "-37" after the part number; e.g. 885-1223-37.

Continued from Page 60





H-89 with internally mounted RTTY Modem

Changing your address? Be sure and let us know since the software catalog and REMark are mailed bulk rate and it is not forwarded or returned.

CUT ALONG THIS LINE

HUG MEMBERSHIP RENEWAL FORM

HUG ID Number: _____

Check your ID card for your expiration date.

IS THE INFORMATION ON THE REVERSE SIDE CORRECT? IF NOT, FILL IN BELOW.

Name _____

Address _____

City-State _____

Zip _____

REMEMBER - ENCLOSE CHECK OR MONEY ORDER

CHECK THE APPROPRIATE BOX AND RETURN TO HUG

	NEW MEMBERSHIP RATES	RENEWAL RATES
U.S. DOMESTIC	$20 ☐	$17 ☐
FPO/APO & ALL OTHERS*	$35 ☐	$30 ☐ U.S. FUNDS

* Membership in France and Belgium is acquired through the local distributor at the prevailing rate.

Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

BULK RATE
U.S. Postage
PAID
Heath Users' Group

POSTMASTER: If undeliverable, please do not return.

P/N 885-2071