REMark®

Volume 7, Issue 8 • August 1986
P/N 885-2079 Issue 79

$2.50

Official magazine for users of HEATH ZENITH computer equipment.

Continued from Page 63

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
885-8006	Farm Accounting System	45.00	30
CP/M			
885-1219-[37]	CP/M Navigational Program	20.00	31
MSDOS			
885-6006-37	MSDOS NAVPROG	20.00	73
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 Ascomy	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
MSDOS			
885-8036-37	MSDOS RFCAD Ver. 3.50	30.00	73
COMMUNICATION			
HDOS			
885-1122-[37]	HDOS MicroNET Connection	16.00	37
885-8005	MAPLE (Modem Appl. Effector)	35.00	29
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
885-8012-[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
885-3033-37	MSDOS HUG MCP	40.00	71
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	
885-4006	REMark Vol. VI Issues 60-71	25.00	
885-4500	HUG Software Catalog	9.75	
885-4501	HUG Software Catalog Update #1	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
885-3036-37	MSDOS TREE-ID	20.00	77

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 82

What A Great Bunch

Dear HUG:

Last August you published my request for help with the reverse video problem I had on my H-89A. Boy, what a response I received. What a great bunch of people in HUG.

All of the replies I received zeroed in on the P101 connector and the BR. I had wired the BR direct to the transformer after reading the article in REMark, so I installed a new BR as recommended by many. In fact, Dr Dave Barbee, Pullman, WA sent me the proper BR to install. I also installed a ground wire from the video circuit board to a corner bolt of the transformer as recommended by Orlan Stone, Medford, OR (Zenith Field Service Bulletin FSB-Z89-070).

I went off-line and filled the CRT screen with reverse video — problem solved, no more "rolling bars". After about an hour with still no problem, I booted and ran a program that contained reverse video — the problem was still there. Back to off-line and no problem; back to the program and the problem.

The reverse video in the program was flickering as compared with the "rolling bars". The flickering was random rather than in a regular sequence of the rolling bars. I tried several programs.

If anyone has a suggestion for this flickering problem, I'd be very grateful to hear from them.

To all those who helped me solve the "rolling bar" problem, my sincere thanks.

Sincerely yours,

Bob Speidel
Box 95E, Route 1
Emmaus, PA 18049

*

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

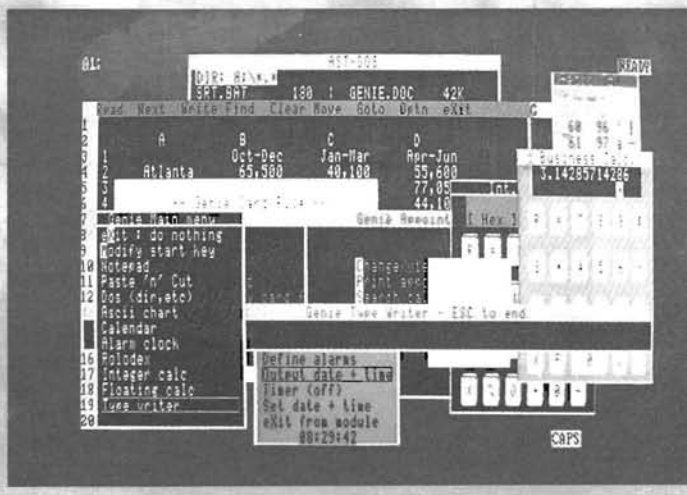
	NEW MEMBERSHIP RATES	RENEWAL RATES
U.S. DOMESTIC	$20 ☐	$17 ☐
FPO/APO & ALL OTHERS*	$35 ☐	$30 ☐ U.S. FUNDS

* Membership in France and Belgium is acquired through the local distributor at the prevailing rate.

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 functions 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

ORDERS ONLY CALL

1-800-437-4400 Ext 800

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

_____	Copies of Genie	@ $54.95	_____
_____	Spelling Helper(s)	29.95	_____
_____	Print Spuffer(s)	24.95	_____
_____	Scientific calc.	19.95	_____
_____	Genie pack (I want it all) Saving $40	$89	_____
		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 _____

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

Production Coordinator ..... Lori Lerch
(616) 982-3794

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

HUG Bulletin Board ..... (616) 982-3956

Contributing Editor ..... William Adney

Contributing Editor ..... Joseph Katz

Printer ..... 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®

Buggin' HUG 7

Mainstream Computing
Joseph Katz 11

Faster Disk Access For Heath/Zenith MS-DOS
Pat Swayne 17

Packet Radio Description And A Simulator — Part 2
Luis E. Suarez 21

P-SST Maestro: Providing The Sounds Of Music To Your Z-100
John Britton 25

Max Memory Mods Review Of Two Memory Modifications For The H/Z-100 PC Series Computers
Pat Swayne 31

SPREADSHEET Corner — Part 16
H. W. Bauman 35

ZPC Update #8
Pat Swayne 41

Official magazine for users of HEATH ZENITH computer equipment.

Volume 7, Issue 8 • August 1986

MACAT: A Disk Cataloging System
Louis M. St. Martin 45

On The Leading Edge
William M. Adney 53

Clock Watcher's Delight Revisited
Pat Swayne 59

HUG New Products
..... 63

HUG Price List
..... 64

SEEK.COM: A CP/M Assembly Language Learning Adventure Part 2
M.D. Zapolski, Sr. 65

EasyPC For The Z-100
LtCol. Myles P. Somers USMC 71

A Winchester For The '89 Part Seven
Peter Ruber 75

Heath/Zenith Related Products
Jim Buszkiewicz 81

Index of Advertisers

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

Advanced Software Technologies 3
American Cryptronics, Inc. 82
Analytical Products 40
Bersearch Information Services 80
ByteSize Concepts, Inc. 24
CDR Systems, Inc. 2
FBE Research Company, Inc. 62
First Capitol Computer 44
Gemini Technology 84
Graymatter Application Software 23
Heath Company 66
Hilgraeve, Inc. 26
HUG PBBS 51
Intersecting Concepts 70
Liberty International 34
KEA Systems Ltd. 81
Kalltronics 33
Micronics Technology 30,33
Paul F. Herman 20
Rex Service Company 20
S&K Technologies 78
Secured Computer 58
Software Toolworks 9
Software Wizardry 78
Western Regional HUG Conference ... 6
Veritechnology Electronics Corp. 72

On The Cover: Shown is an HD-4040 and HD-4040-1. These instruments are used in packet radio. Part 2 of "Packet Radio Description And A Simulator by Luis Suarez is on Page 21.

Western Regional

Heath/Zenith Users' Group

Conference

- LATEST TECHNOLOGY
- NEW SOFTWARE
- HEATH/ZENITH VENDORS
- GUEST SPEAKERS
- PANEL DISCUSSIONS
- DINNER/PRIZE DRAWING

NOVEMBER 14 - 16, 1986

AT... Disneyland® Hotel ANAHEIM, CA.

The Official Hotel At The Magic Kingdom

1986 WESTERN REGIONAL HEATH/ZENITH USERS' GROUP CONFERENCE

A CONFERENCE OF HEATH/ZENITH COMPUTER USERS AND LOCAL USER GROUPS.

SPECIAL GUEST SPEAKERS - PANEL DISCUSSIONS - HEATH/ZENITH COMPUTERS - ELECTRONICS - AND HEATH/ZENITH RELATED VENDORS.

CONFERENCE OPENS FRIDAY AFTERNOON ON NOVEMBER 14TH AND CLOSES AT NOON ON SUNDAY, NOVEMBER 16TH.

☆ [*Advance Registration Rates Good Thru October 15th*] ☆

NOVEMBER 14 - 16, 1986

DINNER/PRIZE DRAWING $25.00
WITHOUT DINNER $7.50

THE DISNEYLAND HOTEL IN ANAHEIM, CALIFORNIA

— WIN A COMPUTER —

THOSE WHO ATTEND THE DINNER* ON SATURDAY, NOVEMBER 16TH ARE ELIGIBLE FOR NUMEROUS PRIZES INCLUDING

(1) HS 241-81 (Extended Drive) With Video Card and Monitor

(2) ZP150 (1) Hero Jr.

— SPECIAL DISNEYLAND HOTEL RATES —

Single — $75.00
Twin/Double — $75.00
Triple/Quad — $86.00

*You must be present to WIN!!

PLEASE COMPLETE FOR EACH REGISTRANT:

MAKE CHECKS PAYABLE TO: **WESTERN REGIONAL HUG** CONFERENCE
Mail To: ● 1555 N. Orange Grove Avenue, Pomona, CA 91767

NAME _____
ADDRESS _____
CITY,STATE,ZIP _____

CHECK ONE
__ Admission includes Dinner/Prize Drawing $25.00 ($28 after 10/15)
__ Admission WITHOUT Dinner $7.50 ($10 after 10/15)

NAME _____
ADDRESS _____
CITY,STATE,ZIP _____

CHECK ONE
__ Admission includes Dinner/Prize Drawing $25.00 ($28 after 10/15)
__ Admission without Dinner $7.50 ($10 after 10/15)

NAME _____
ADDRESS _____
CITY,STATE,ZIP _____

CHECK ONE
__ Admission includes Dinner/Prize Drawing $25.00 ($28 after 10/15)
__ Admission WITHOUT Dinner $7.50 ($10 after 10/15)

NAME _____
ADDRESS _____
CITY,STATE,ZIP _____

CHECK ONE
__ Admission includes Dinner/Prize Drawing $25.00 ($28 after 10/15)
__ Admission WITHOUT Dinner $7.50 ($10 after 10/15)

BUGGIN' HUG

Oh How I Wished It Had Come Earlier

Dear HUG:

On Monday, January 13th I received the January issue of REMark. A quick skim of the index and I was studying your article on The Gemini Board. Oh how I wished that your article, i.e. the Magazine had come on the previous Saturday. Let me tell you the story.

I had recently bought a Gemini board and late Monday afternoon was starting to copy programs into the PC division of my Z-100 Winchester. The Zenith MS-DOS side worked fine. I wanted to print some directories of the subdirectories. It wouldn't print. I must have run thru the Configur program two dozen times, with all those questions about parity, stop bits, seven and eight bit words. There are lots of conceivable combinations. I restudied the manual on my trusty MX-100 printer with serial interface. Finally, it was late, too late I thought to call anyone, and besides dinner was probably getting cold at home. There was the latest issue of REMark.

A first reading of your article and it sounded like I'd have to have one printer cable for the Zenith MS-DOS and a separate one for the PC version. But wait . . . what's this about COM2 . . . I never would have guessed. It was so easy . . . once you know how. . .

Many thanks . . . and the rest of the article is also very helpful and informative. Thanks again.

Yours truly,

James R. Deephouse
19 Pelham Road
West Hartford, CT 06107

Graphix Plus II

Dear HUG:

I would like to install a Graphix Plus II (Rev. 0.3) board in my Z-150 HD Computer. We have tried it and the unit locks up; also we would like to add a second disk drive and are wondering if anyone has done so, and if you can offer any advice?

Sincerely,

Connie Williams
Broadway Trust
2654 Providence Street
Ft. Myers, FL 33901

Fixes For MagicWand?

Dear HUG:

I have been using MagicWand, Release 1.11, with CP/M on my H-8 since 1982, and prefer it to WordStar for many uses. I have a couple of problems with it, but I have not seen any fix for them in REMark or the other Heath-oriented magazines I read. I hope someone has solved them and can provide the information to the H-8 and H-89 users.

The problems relate to the use of proportional spacing on my Diablo 1640 (WH44) printer. My first problem was to locate plastic printwheels for the 1640, which will not work reliably with metallized printwheels. I have finally found suppliers for these.* The first one I located was a Camwil THEME 10pt PS B08501-01. The proportional spacing table in the MagicWand PRINT command appears to be well suited for this printwheel.

Can you supply a patch or other information to customize PRINT .COM for these or other printwheels?

Since adding additional disk drives to my H-8 system, I have encountered another problem with MagicWand. I can EDIT files from various disk drives, but can PRINT only from drives A: and B:. Any other drive in the file name gives the response "Invalid file name. Name of the primary text file:." Since my drives A: and B: are the original H-17 hard-sectored drives, and the additional drives use soft-sectoring, I am unable to take advantage of the greater storage capacity of the soft-sectored drives. The A: and B: drives use the Heath hard-sector controller board; the other drives use a Trionyx C-H8 controller. With this hardware, it is not feasible to configure the system with the soft-sectored drives as A: and B: without giving up other desirable features. I see nothing in the MagicWand documentation stating that PRINT works only with drives A: and B:.

Can anyone supply a patch or other information on how I can use PRINT with additional disk drives?

Yours truly,

Herold A. Treibs
Box 334
Richland, WA 99352

* Sources for proportional space printwheels for Diablo 1640 and similar printers:

The Drawing Board
P.O. Box 2995
Hartford, CT 06104-2995

Business Support Services, Inc.
705 Butternut Avenue
Royal Oak, MI 48073

WordPerfect And The MPI 99G

Dear HUG:

I need some help making WordPerfect print out on my MPI 99G using the software driver (Artisan). If I understand the problem correctly, the program outputs via the BIOS and not via a DOS function call, thus bypassing the software driver. I had the same problem using PeachText (but not with WordStar). Has anyone figured out a patch to use on either of these word processors so that they will be rerouted through the software driver? The companies were not helpful in solving this problem.

Thank you,

James Spinti
6051 S. Drexel
Chicago, IL 60637

A Unique Problem With "Graphics"

Dear HUG:

Recently, I purchased a copy of your wonderful program ZDOS/MSDOS Graphics, Heath P/N 885-3031-37. The program works extremely well only I have a unique problem:

I have an HP Laserjet Printer which works incredibly well with everything but graphics. Yes, I have a Z-100. Do you have any ideas about how I can make your wonderful package print to a laserjet printer?

Any thoughts you might have would be greatly appreciated.

Best regards,

R. Gillem Lucas
Gillem Lucas & Associates
3111 Camino Del Rio North, Suite 1155
San Diego, CA 92108

Pat Swayne's SCRNCLK

Dear HUG:

I'm beginning to tutor myself in 8088 Assembly Language. My first project was very modest but successful and useful.

I love the SCRNCLK but never could get it to turn off. I found out why: you must enter CLOCK O, that is capital O. To get it to go off for both upper and lower case o, I changed Listing 2 on Page 40, May 1984 REMark to:

```
SOS:    LODSB                   ;GET NEXT BYTE
        CMP     AL,' '          ,SPACE?
        JZ      SOS             ;IF SO, SKIP IT
        CMP     AL,'O'          ;OFF?
        JZ      CLKOFF          ;YES, KILL CLOCK
        CMP     AL,'o'          ;OFF?
        JZ      CLKOFF          ;YES, KILL CLOCK
CLKON:  MOV     BYTE PTR ES:[BX],1  ;ENABLE CLOCK
        INT     20H             ;AND EXIT
CLKOFF: MOV     BYTE PTR ES:[BX],0  ;KILL CLOCK
EXIT    INT     20H             ,AND EXIT
CLOCK   ENDS
        END     START
```

There may be (probably are) fancier ways to do it but it works!

Sincerely,

David K. Wheeler
306 Winslow Street
Watertown, NY 13601

SuperPATH By Martin Scot Development Corporation

Dear HUG:

I was recently asked to beta test a program called SuperPATH by Martin Scot Development Corporation on my Zenith 150. SuperPATH is a family of directory performance utilities for IBM PCs and compatibles, which not only works on the 150 PC, but is a solid, well-built product.

SuperPATH's utilities, created to be both an extension and improvement of MS-DOS, can be used with either a hard- or floppy disk system. They can be made memory-resident or used as executable files in your root directory.

The SPATH command replaces the limited DOS Path command; the benefit being that applications have no problem finding their supporting files in the directory maze. Programs like WordStar can be called from a document directory, a feat that is not possible using PATH alone.

Four other utilities on the disk help round this product out nicely. First, a program that lets you rename a directory - try that in DOS (a process that includes creating a new directory, copying the files to it, deleting the old files and then deleting the old directory).

The second is a program that allows you to create an entire directory path with one command. When you combine this with number three, directory listing utility, that among other things, allows you to output your file names into a batch file with DOS commands appended, you have the makings of a simplified back-up procedure.

The fourth utility is an enhanced "change directory" command. It allows you to run an "autoexec" type batch file upon entering a directory. Handy if you need to change a path specification or routinely run any other DOS command each time you change directories.

The final version of the product has nine utilities in all, with concise help screens, and a complete user's manual. It's well worth the $39.95 investment. For more information give Martin Scot Development a call in Seattle at (206) 527-9605. Or write to them at 4515 Purdue N.E., Seattle, WA 98105.

Regards,

David M.H. Butler
1619 E. Columbia Street
Seattle, WA 98122

Not Obvious/Easy Answers

Dear HUG:

As a quite constant user of the ZP-150 these last months, I've been delighted to keep discovering new features and applications (yes, Rick Lutowski — I'm very much an "Indian"!), but I've also run into some nuisance-value characteristics that haven't seemed to have obvious/easy answers. For letter writing, I normally want other-than-default formatting values and I detest having to enter these separately each and every time I begin. Finally, a simple solution dawned on me; set up the desired formatting under WORD, then use the COPY function to create a named file of the formatting (only) with a non-WORD file-extension; I've been using FMT.PLN for this purpose.

When I want to begin a new letter, I can COPY the file FMT.PLN to FILENAME.WRD and I'm ready to go; it's as easy as that! Attempting to RUN the file FMT.PLN (under PLAN) results in a "Cannot run application" error message; the file remains intact. A standard letter header, or several different ones, could be stored and recalled in a similar manner.

Irving G. Bouton
134 Camelot Drive
Rochester, NY 14623

UCSD Pascal Problem

Dear HUG:

I've run into a problem with UCSD Pascal on my Heath H-120 and I thought perhaps one of REMark's readers might have the solu-

tion. I doubt if it's a bug per se, but it has caused no end of trouble trying to figure out what's going wrong.

The problem relates to turtlegraphics on the Z-120. Both the EXAMPLE2.CODE and EXAMPLE4.CODE programs execute, suggesting that the graphics code is present and properly installed, but when I typed the turtlegraphics demonstration program on pages 8-42 and 8-43 of the Applications Development Guide, the compiler gives me the following error:

```
Unit not in library
Line 4
```

Line four of the demo program reads:

```
{$U SCREENOPS.CODE} screenops;
```

The preceding line reads:

```
USE turtlegraphics;
```

Which apparently compiles correctly. Using the LIBRARY utility to examine the file SYSTEM.PASCAL shows the following is present (among many others):

```
1 u TURTLEGR 4119
```

Using the same utility to examine SYSTEM.PASCAL shows the following is present (again, among many others):

```
17 u SCREENOP 582
```

The documentation on the compiler says that both the above files are searched at compile time in order to locate the required code called for in the demo program, but the compiler clearly isn't finding the SCREENOP code. Changing the name in the demo program to match the contents of SYSTEM.PASCAL produces the same results, and I can't find a way to extract SCREENOP from SYSTEM.PASCAL to make a separate element for inclusion in the demo. The format of the piece of code I've been able to extract is incompatible with that required by the compiler.

I've run out of things to try, yet I feel that it's just a question of leaving out a step or doing something wrong. Can you give me some pointers to help me get off dead center? I'll appreciate any help you can offer.

Sincerely,
D.C. Shoemaker
HQ USEUCOM Box 897
APO NY 09128

DSKLBL.BAS: January REMark

Dear HUG:

The article in January's REMark called DSKLBL.BAS was very timely. It came just as I was in the process of doing my yearly labeling of files at the store — a very boring chore. I recognized the possibilities right away. I changed the program as given in the following listing and ran it on my Z-100 with GW-BASIC.

I changed line 10 to include the information that was in REMark Jan. '86. I added a colon and Read C$ to line 80. This allows the program to automatically read the data given in line 110 and print the line, "Charge Sales", changing the month each time through the loop. Since most of the time I require one label for each month and have some six areas where I have to have labels, it is very easy to change line 90 from "Charge Sales" to "Credit Memos" and let the computer print the labels and change the month each time through the loop. Where necessary, the position of the month on the line can be changed by changing the Tab(15) on line 90 to whatever you need. I left out any lines that I did not need. I hope others find this program as useful as I did.

I also thought it was very good that the article included information on how to adapt the program to a different system than the one it was written on. Is it possible that REMark could adopt a

policy of giving some information on how to change BASIC programs that, for example, were written in MBASIC to run in HDOS or ZBASIC. Or devote an aritcle to translating from one BASIC to another. I have found many programs in REMark that would be exciting to put into my Z-100, but just don't know how to translate the MBASIC to something my GW-BASIC can deal with.

Sincerely,

Joe Nikolai
HQ USAFE Box 5876
APO NY 09012

```
10 'dsKLBL.BAS VER 1.4 07/09/85 R. BOEHLEIN REMARK JAN 86
20 PRINT CHR$(27)+"E"
30 INPUT "How many labels do you wish to print?";A
40 PRINT :PRINT
   "Make sure printer is ready and labels are in position "
50 PRINT "Hit any key to start printing.":X$=INPUT$(1)
60 'SET PRINTER TO 8LI
70 'LPRINT
80 FOR B=1 TO A:READ C$
90 LPRINT "Charge Sales";TAB(15) C$
100 LPRINT
110 DATA Jan 86,Feb 86,Mar 86,Apr 86,May 86,Jun 86,Jul 86,
       Aug 86,Sep 86,Oct 86,Nov 86,Dec 86
120 LPRINT :LPRINT :LPRINT:LPRINT
130 NEXT B
140 PRINT : PRINT :PRINT "MORE(Y/N) <N>";:X$=INPUT$(1)
150 IF X$="Y" OR X$="y" THEN 20 ELSE 210
210 PRINT CHR$(27)+"E":SYSTEM
```

Another Approach To Keyclick

Dear HUG:

I've found a useful trick that some of your readers may find of interest. Over the last couple of years, I've seen numerous methods and tricks for turning off keyclick, some reasonable and many cumbersome. Few have addressed the irritating problem of software packages which reset upon exit so that you're once again back with power-up conditions and have to reinitialize your environment back the way you want it.

For those of us with MS-DOS 2xx and above, a simple way to do this is via the PROMPT command, which can also be used to maintain fore- and background colors. Simply put, for my Z-100 I use the following PROMPT command in my AUTOEXEC.BAT file to ensure that keyclick stays off, and I keep a foreground color of cyan:

```
PROMPT $em50$ex2$n$g
```

The $e is used to enter an escape character, the m50 gives me the color scheme I want, the x2 turns off keyclick, the $n and $g establish the default drive and ">" sign which is the normal default prompt string. Using this has removed a source of some irritation for me and seems to have no adverse effects.

Cordially,

Neal A. Van Eck
417 Tower Drive
Port Washington, WI 53074

A Nice Company To Deal With

Dear HUG:

Like most people, I usually write letters about a company's product or service only when I'm unhappy with that company. I'm glad to say that this letter is for quite the opposite reason. I own an H-89A and decided it was time to upgrade the old warhorse rather than start from scratch with a whole new system. After scouring hundreds of ads and reading umpteen articles in REMark and SEXTANT on what I could add to the 89 to bring it up to snuff, I settled on a C.D.R. double-density controller board, a C.D.R. SuperRAM 89 ram disk and a 4 MHz speed module, all from ANALYTICAL PRODUCTS in Woodlake, CA. I placed the order over the phone with the owner, Peter Shkabara, and in just over 1 week everything was delivered. Very impressive service, thought I.

C.D.R's installation manual for the controller board leaves a bit to be desired, but thanks to a separate letter Peter enclosed to clarify things, installation went quite smoothly and was relatively quick. I won't bore you with the details, but problems developed. Some of which were my doing, some with the 89 itself, and some with a faulty file or two on the software distribution disk. To cut a long story short, Peter was quick to respond and solved all the problems.

The old 89 is humming along quite nicely now and I couldn't be more satisfied. C.D.R. deserves a pat on the back also. The ram disk is just great! All you huggies out there who want to upgrade your 89 won't find a better company to deal with than ANALYTICAL PRODUCTS. They have products for other H/Z computers also, including ZCPR3, so send for their latest catalog or better yet, place an order. You won't be disappointed!

Sincerely,

Bill Gahn
10030a Executive Drive North
St. Ann, MO 63074

Help!

Dear HUG:

I installed a second internal 5-1/4" disk drive in my Z-100 last year, but I've decided that two drives aren't enough. Can someone tell me how I can add two more external 5-1/4" disk drives to the 8" floppy port which comes with the computer so I can use them as drives C: and D:? I use Z-DOS at home and MS-DOS at work. Any information would be greatly appreciated.

Sincerely,

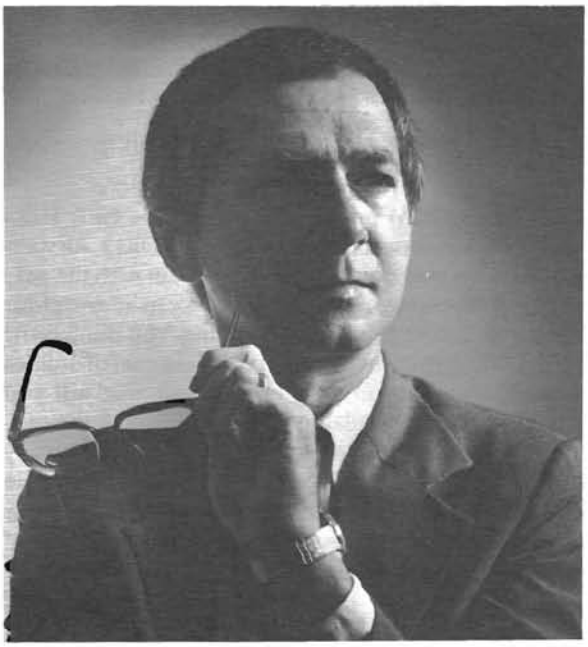
Glenn D. Faini
2525 Corning Avenue, Apt. 102
Fort Washington, MD 20744-3039

EasyPC Emulation

Dear HUG:

I enjoyed William Adney's review of the Gemini Emulator Board in the January issue of REMark. Like many in the Zenith community, I have been waiting for rumored IBM-emulation for the Z-100 to become a reality in the marketplace. I am in charge of microcomputer operations for a college and as such, have responsibility for over 50 Z-100 low profile computers. Over the past three years, we have found software support for the Z-100 to be adequate for our needs. However, we have found that much of the commercially available educational software is written specifically for the IBM PC. Consequently, to get the maximum use of our Z-100s, emulation was desirable.

Continued on Page 82



Mainstream Computing

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

I'm nearly finished playing catchup after the coincidence of Spring COMDEX with the end of Spring Semester. Enthusiasm as usual had outstripped reason, leaving me to deal with ambitious commitments that all fell due at about the same time. Those to the University took priority as always, then came the heap of others. This unsalaried few weeks before summer school starts has been exhausting. Which way to that Ivory Tower everyone talks about?

Of course it's been fun too, exploring new things for mainstream computers. But even the fun has involved work. For example, I've had to establish a reject pile for products that advertised better than they perform, and a hold pile for products that promise immediate revision to work properly on Heath/Zenith compatibles.

A few vendors seem not to understand the differences among reviewing, evaluating, and beta testing, which makes dealing with them incredibly — and unnecessarily — difficult. Of those, we will say little . . . and, eventually, nothing at all. But what to do in the case of a potential dazzler that just doesn't work properly for us right now, produced by a vendor who really seems to want to fix it for us? There's one package, for example, that works under Zenith's MS-DOS 2.10 but not Zenith's 3.10, even though it's supposed to work with both. I called the vendor's toll-free support number and the technician compared COMMAND.COM dates with me: they were different, so the problem seems attributable to a quiet revision of the operating system and the vendor welcomes the opportunity to make the program work with both versions. He promised a program revision for my MS-DOS 3.10, which I assume is the 3.10 others have too. How many software houses do you know that maintain toll-free support and are eager to do what it takes to serve the Heath/Zenith community? Before you read this column, the package should be working and ready for evaluation. It's cases like this for which I've established the hold pile. Obviously, I think a user's column has potential to do missionary work. We want to encourage vendors to address us, not blast them when they make a misstep when venturing on our turf. Or so I think.

Part of the fun to talk about this time has been two nice finds: the ZOOM/Modem PC1200 and a couple of Run/CPM packages. They confirm my sense of our being in a time of consolidation, perfection, and polishing for this generation of mainstream computers. Nevertheless, both of them surprised me. We'll also look in passing at the information you need to build your own modem cables for the H/Z-241. Make one yourself — no skill required — and you save the price of your next year's membership in the Heath/Zenith Users' Group. That's pretty good, don't you think? There's also an "undocumented feature" — the current euphemism for "bug" — in Crosstalk XVI that I'll tell you about. And I'll wind up with a glance at things to come.

Figure 1
The ZOOM/Modem PC 1200 for XT and AT compatibles.

The ZOOM/Modem May Become The New Modem Standard
ZOOM Telephonics' ZOOM/Modem PC1200 arrived a few weeks ago. It looks like we have something special here. I'll say it

straight out: I think the ZOOM/Modem may well become the new standard in modems for mainstream computers. So far it seems to have invalidated two of my longstanding rules.

Rule One was, Never bank on a so-called "Hayes-compatible" modem instead of the real thing. The simple reason is that I've never found a truly Hayes-compatible modem. If I can't move all my software from my Hayes and have it behave the same way on the Hayes-compatible, with no changes at all, it may be a very nice modem, but Hayes compatible it isn't. Oddly enough, I seem to have been more steadfast than Hayes Microcomputer Products. Its Smartmodem 2400 uses a superset of the "Hayes standard" commands and, according to the August 12, 1985, issue of InfoWorld, may vary in other ways from its Smartmodem 300, 1200, and 1200B — the very modems that set the standard and are used to define "Hayes compatibility."

Rule Two was, Always go with an external modem instead of an internal modem. Three of the reasons are that internal modems use an expansion slot without compensating the loss, internal modems are more difficult to install than external modems, internal modems must be so much smaller than external modems that they tend to be much less capable and reliable, and internal modems are difficult or impossible to move from one computer to another. Here, too, I've been much more dogmatic than Hayes, because its Smartmodem 1200B is an internal against which other internals are measured.

I bought those rules with my own time and money, so I have followed them out the window. Well, the ZOOM/Modem is a Hayes-compatible 300/1200 Baud internal modem and it's a knockout and the only exception to my rules so far. The reason is, it preserves Hayes compatibility while setting new standards for functionality. It does Hayes, then goes much further. Understand that I'm not in any way knocking Hayes modems. They are Quality. What I'm doing is praising ZOOM/Modem, which looks to me like equivalent quality with more features for less money.

The ZOOM/Modem is indeed Hayes compatible: Crosstalk XVI, HyperACCESS, and HUGMCP are the three communications programs I've been using steadily with my external Hayes, and they worked — with one change of port assignments because of the way I chose to install the ZOOM/Modem, not because of anything to do with Hayes compatibility — after I unplugged the Hayes and installed the ZOOM/Modem. What impressed me even more was that all my Crosstalk XVI and HyperACCESS scripts run with the ZOOM/Modem exactly as they do with the Hayes. That's really impressive because my scripts depend on timers, such as "WAit DElay 20 : REply "go PCS48 | " ("wait for two seconds before issuing the command 'go PCS48'," which logs onto CompuServe's HUG Special Interest Group). If the timing is off, as it is with other "Hayes compatibles" I'd tried in the past, an entire unattended session can go "Boom!" most expensively. I haven't had a failure yet with the ZOOM/Modem. Kerplunk goes my blanket objection to a Hayes-compatible modem.

There seems to be one basic ZOOM/Modem, offered in two models distinguished by features above and beyond those which make it Hayes compatible. The easiest way to explain the basic features, those in the ZOOM/Modem ST, is to suggest that you read the specifications of a Hayes Smartmodem 1200B, then add the following: support for four COM ports instead of the usual two; "Demon Dialing"; an audio output jack; a built-in speaker with an accessible volume control, and the ability to work reliably in an AT running at 8MHz. That's the basic ZOOM/Modem — the ST — at $299, significantly less than the Hayes Smartmodem 1200B at $549. The advanced ZOOM/Modem XL has all those features plus these: clock/calendar; touchtone decoder; touchtone password capabilities; and an 8KB buffer. It lists for $399 — close to the street prices I've seen for the Hayes.

Let's set price aside and take a brief tour through those features:

- **Support for four COM ports (both models)** — Early H/Z-150s came with two serial ports: COM 1 and COM 2. All standard compatibles now come with just one, but with the easy capability to address two. In fact, MS-DOS can address four COM ports but that requires programming. You don't do it: the ZOOM/Modem comes with as complete a set of utilities as I've seen, including installation routines that tell you which COM ports are available and let you install the modem as the one you want. You don't have to fiddle with any switches, or even know what you are doing, because ZOOM's software does the work. Kerplunk goes one of my objections to an internal modem: although the ZOOM/Modem occupies a physical port, it actually adds serial port addresses.

- **8MHz-AT compatibility (both models)** — Kerplunk goes one of my objections to an internal modem: the ZOOM/Modem is portable, at least among the compatibles we now have. Heath/Zenith's AT compatible, the 241, runs at only 6MHz so the ZOOM/Modem should be under no great strain when I get a chance to try it in my H-241. I haven't yet because the machine is down. When it's resuscitated, again, I'll move the ZOOM/Modem and see how it works in the 241. I anticipate no problems with the modem.

- **Demon Dialing (both models)** — ZOOM Telephonics invented the Demon Dialer, which is a telephone dialer that can alternate between dialing busy numbers very quickly, over and again, until one of them answers. The Demon Dialing feature on the ZOOM/Modem allows dialing two such numbers. I find it useful when I have to log onto bulletin boards that always seem busy: the persistent little devil works fast enough so I sneak in between the time one caller hangs up and the next caller tries dialing by conventional means.

- **Audio Output Jack (both models)** — You can connect an audio output device — microphone output, tape recorder, voice synthesizer, and the like — to the jack. Couple this capability with the ZOOM/Modem's ability to distinguish a human caller from another computer and you have a system that can respond appropriately to each kind of caller. Combine these features with the touchtone sensitivity of the enhanced model XL and you have spectacular possibilities.

- **Touchtone decoder (XL only)** — As I was saying, the ZOOM/Modem provides spectacular possibilities. Add the ability to decode touchtone signals to the audio jack and your single computer can serve as a central information system, for example. It answers a voice call, instructs the caller to tap in his code number on the phone's keypad, then delivers information intended for that caller. Perhaps the word is to call back by computer to download a document. The caller does. Interesting?

- **Touchtone password (XL only)** — Even more interesting is the security gained by requiring a password so strangers don't barge in. The phone rings. The computer answers. The caller hears nothing at all until he taps out the required password, at which point — and only then — does the system continue. Wrong password? Too long before the right one comes? Click: no connection.

- **Clock/calendar (XL only)** — Match the above features to software with callback capability, like Microstuf's Remote, so that your system responds to the correct password by calling back to the remote computer at its preassigned telephone number at a predetermined time or after a specified delay, and you have a pretty secure system without spending impressive money. (You also have the basis for a good spy thriller, don't you?) ZOOM Telephonics' background in voice-telephone devices makes me suspect that the company has some surprises up its corporate sleeve. At any rate, kerplunk goes one of my objections to an internal modem: I pulled my separate clock/calendar board and used its slot for the ZOOM/Modem with its own clock calendar, so I wound up with the same number of physical slots I had before installing it.

- **8KB buffer (XL only)** — You have to finish a report and the clock/calendar is ticking towards your deadline. Word processor, spreadsheet, database manager: that's the software on your mind, and you'll just have to close off the outside world. Right? Wrong. Your ZOOM/Modem can take incoming computer messages and store them until you can look at the incoming mail. The buffer serves — get this — as an answering machine for computer calls, right down to a "prerecorded" greeting.

Let me prepare you for the feature I found decisive, which I haven't yet mentioned. The ZOOM/Modem is modular. Major components are socketed, not soldered. The XL model, therefore, is really the ST model with the additional features plugged in. Should you buy the ST and later decide you want the additional features, ZOOM Telephonics will sell them to you separately: the clock/calendar module is $49; the touchtone decoder/password module is $29; the 8KB buffer is $29. The total upgrade cost is $107, only $7 more than if you had ordered the XL in the first place. (I don't count the cost of PFS: ACCESS, a communications program you can buy at a discount with the ZOOM/Modem. The reason is that I don't count PFS: ACCESS itself, because it is copy-protected. With so many good communications programs available for every level of need, I can't imagine considering a copy-protected communications program. I don't consider it.) ZOOM Telephonics evidently has kept the penalty for under-ordering quite reasonable. But all that is only preparation for what made me decide to look at the ZOOM/Modem in the first place.

The ZOOM/Modem was introduced with the announcement that in June, 1986, ZOOM Telephonics will provide registered owners with the opportunity to upgrade to 2400 Baud. Terry Manning, ZOOM's spokesman, tells me that the upgrade will be $199. I'm supposed to get one, so by the time you read this column I should be working with the upgrade. My Hayes Smartmodem 1200 started life with a maximum capability of 1200 Baud and will end its life no faster. Not so my ZOOM/Modem PC1200. Nice?

Do take a look at this modem, even if you already have a modem and are not actively in the market. If I'm right, now might be the time to do some horse trading on your existing modem — especially if it's a Hayes — and switch over to the ZOOM/Modem before it becomes the hottest thing around. Manning tells me that several of the Heath Electronic Centers stock the ZOOM/Modem, so you can see one there.

From now on I'll be using the ZOOM/Modem as a testing standard here. If anything for the "Hayes standard" does not work properly on it, I'll rerun the tests on my Hayes and let you know what happens both ways. ZOOM Telephonics is making that job easy.

An "Undocumented Feature" Of Crosstalk XVI

When I installed the ZOOM/Modem as COM 2, left the H-158 serial port as COM 1, and disconnected the Hayes Smartmodem 1200, of course I had to tell my communications software to use COM 2 instead of COM 1. That was the only necessary software modification in the changeover. Elementary.

Well, the experience taught me something curious: Crosstalk XVI itself seems not to check if anything really lives on an active serial port it has been told to address. What I mean will become clear when I tell you how I found it out.

You set Crosstalk XVI parameters — such as the port to use — from files with an "XTK" extension. I was sloppy and missed a few of those "XTK" scripts, including the one that calls CompuServe. It, therefore, sent everything to COM 1 when I began testing Crosstalk XVI on the ZOOM/Modem, but of course there was nothing connected to COM 1 because I had removed the Hayes. My omission would have been insignificant if Crosstalk XVI did any error checking. But it didn't. It dialed the phone number, told me to wait for a connection, then said it had made the connection and ran the "XTS" script that logs me onto CompuServe. And then it waited for the CompuServe "ID" prompt . . . forever. If I hadn't tested the ZOOM/Modem first with HUGMCP (a simple program that takes the port number in one place only), I'd have thought there was something wrong with the modem. There wasn't, of course: Crosstalk XVI was making the trouble. But the discovery took time because I assumed the existence of an error trap that doesn't seem to exist. Boo!

If you own Crosstalk XVI and have worked a way around this "feature," I'd be grateful to know. No matter which program you use, keep my experience in mind if you have problems after changing COM ports — or after changing communication programs.

Making Your Own AT Cable

One advantage to using an internal modem is that you don't need a cable from the computer to the modem. You do need one with an external modem or other device that runs from the computer's serial port. Maybe I can save you some time and money connecting any of those devices to the serial port on a 241 or other AT compatible. It's configured DTE (Data Terminal Equipment), of course, so you need a standard cable to hook up DCE (Data Communications Equipment), such as a modem.

Of course, we don't really mean "standard" when we say "standard" in this context: we're talking computerese here, not English. What we mean is only a straight-through cable, in which each pin on the computer output is connected to the equivalent connector on the modem input. We most certainly do not mean that the same cable can be used on PC compatibles and AT compatibles: AT compatibles, like the 241, need a DB-9 connector, while the PC compatibles take a DB-25 connector.

I needed a modem cable immediately after my 241 arrived, and the local Radio Shack Computer Center is only a few blocks from home, so on the way to the University one day I stopped and bought one: $24.95 for a six-foot serial cable for the Tandy 3000 (catalog number 26-1399).

That's all right for a one-shot deal, I suppose, but it's an unnecessarily-expensive proposition to buy several cables when cable-building is so easy and the makings so cheap: one DB-9 male connector and a hood for $2.00, one DB-25 male connector and hood for $2.00, and 9-conductor cable for 18 cents a foot come to $5.08. The saving is about $20 per cable and the work is minimal: 18 solder points or crimps, depending upon the kinds of pins in the connectors you buy. Even an English Professor can do construction on that level.

To build a cable you need to know the "pinouts," of course: in this case, the equivalent pins on each connector. Table 5.2 in the Z-200 PC Series Owner's Manual gives the pin definitions at the 241 end, but then you have to hunt elsewhere for the equivalences at the modem end. Or you can use the handy diagram accompanying this column and get right to work saving some money.

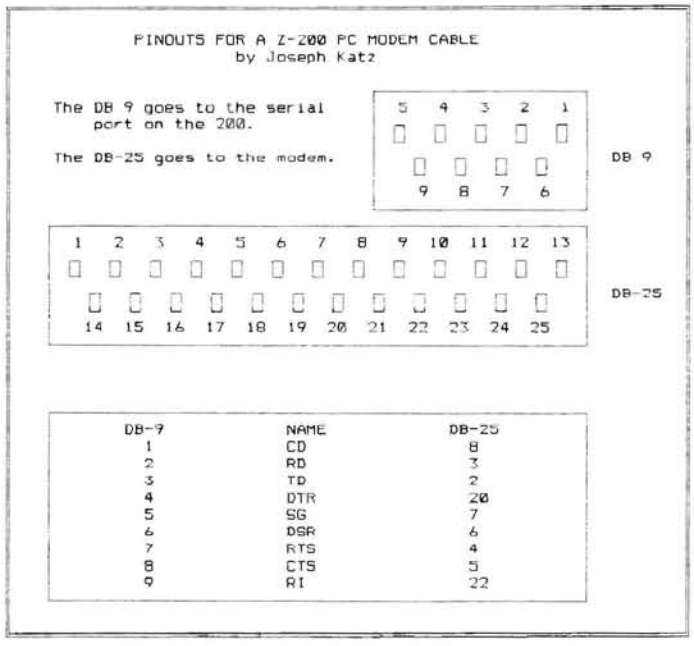


Figure 2
Pin connections for building an AT-compatible modem cable. Diagram made with MonoGrafx.

MonoGrafx

You might be interested in knowing that the handy little diagram showing the pinouts was done in about ten minutes with Monografx. MonoGrafx is the best thing I've seen so far for doing charts, forms, and line diagrams intended for printing. It's fast to load, easy to use, and what reviewers call "intuitive": which I think means that the program works just the way a reasonably intelligent person would expect it to work. What I like most is that I don't have to take a refresher course each time I use MonoGrafx.

If you have any of Heath's or Zenith's compatibles and an Epson or Epson-compatible printer, you needn't bother installing MonoGrafx. Use it right from the box. Installation for other printers takes a second: just choose from the list of supported printers in the installation program. If you own a Hewlitt-Packard Laserjet, there's support for it, too. MonoGrafx comes with a tutorial disk and a manual written in my native language, English. You really don't need either to use the program — it has sensible pulldown menus, rational prompts, and a help screen that does the necessary — but both are worth examining as models of the way such things ought to be done.

In fact, the only thing I don't understand about Monografx is the pricing. It's $69.95 for a copy-protected version, $99.95 for an uncopy-protected version. Maybe the $30 extra defines the term "a license to steal?" Borland International plays this same game with SideKick, but it's a tasteless game no matter who deals the cards. Otherwise, MonoGrafx has class. We'll talk more about it in the future.

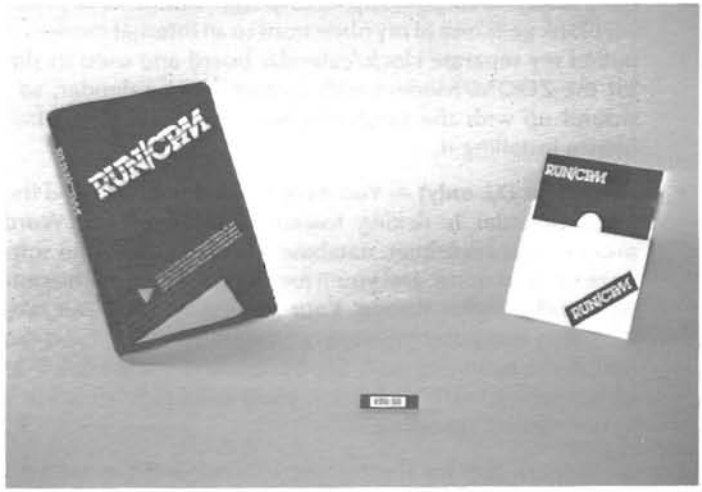


Figure 3
With Run/CPM, the Z-100 PC can run CP/M programs from the Z-100 and other 8080/8085 computers.

Run/CPM And Run/CPM-Z80

Chances are, some day I will cut my remaining ties to the CP/M operating system, but not yet. I still use a few CP/M programs, still have a couple of hundred CP/M disks with valuable, but infrequently-used data, and still use a Kaypro 4-84 computer with CP/M as my transportable. All together, it's too much of an investment to discard. Janet's investment in CP/M is even more considerable because, as a graphic designer, she must be prepared for clients with projects that include material for typesetting from CP/M disks. I'm, therefore, easily interested in products that let me use CP/M disks in my mainstream computers.

Most such are "floppy disk converters," which allow you to copy disk files from the CP/M disk to MS-DOS. RDCPM.COM, which Heath/Zenith supplies on all its MS-DOS distribution disks, is a simple example of the breed: it will read the directory or copy files from 5-1/4", 48-tpi ("tracks per inch") CP/M disks made on a Heath/Zenith computer or IBM PC CP/M-86 system. That's an awfully nice throw-in, but RDCPM isn't sophisticated enough to cope with the bewildering variety of CP/M disk formats. It can't do a thing with my Kaypro 4-84 disks, for example. Commercially-available disk converters present a menu — sometimes several menus — of supported CP/M formats and, if your disk is on it, allows you to copy from it. The question is what to do with a CP/M disk file that has been moved to MS-DOS.

Data files aren't much problem. If they've been produced by the CP/M version of an MS-DOS program, they can be used without fiddling. Even if the data came from a program not available on MS-DOS, if you know the present structure and the one you need, someone usually can do a filter to convert it. (My language of choice is C, which is perfect for such things.) Data files aren't much problem after they have been moved from CP/M to MS-DOS.

Programs used to be a big problem because you can't just move program files from CP/M to MS-DOS and run them. They won't, unless you run them under a CP/M emulator. This kind of software serves as a translator between the CP/M program and the MS-DOS environment in which you want it to run. Early CP/M emulators for MS-DOS, such as HUG's own CPEM and The Software Toolwork's ZP/EM, used software only. They do nicely in simulating a CP/M environment for many programs, but they cannot satisfy the requirements of many other programs.

It's a complex area made quite complicated by the family tree of microprocessors used in CP/M machines. You don't need to know all the details, but you do have to remember that there are two branches in the tree: the Intel 8080 and the Zilog Z-80. The Z-80 added a set of instructions to what the 8080 made available, so programs written to run on the 8080 will run on the Z-80, but the converse is not true. The H/Z-100, for example, has an 8085 microprocessor (in addition to an 8088 microprocessor) and, therefore, can use the CP/M operating system. But it will not run CP/M programs written for the Z-80, only those for the 8080. So it will run SuperCalc but not SuperCalc2.

Because the Intel 8088 microprocessor used in PC compatibles is a descendant of the 8080 line, early CP/M emulators could simulate much of the environment needed by programs for 8080 CP/M programs, but not for Z-80 CP/M programs. Even with the 8080 programs, the early CP/M emulators could not do a complete job: some programs use perfectly legal 8080 instructions that are hard or impossible to simulate in MS-DOS on an 8088.

Time droned on, at about the same pace as the past few paragraphs. Then, more than a year ago, NEC's V series of microprocessors introduced new possibilities. These chips speed up MS-DOS processing while making the full range of 8080 instructions available to software that can use them. Of course, a new generation of CP/M emulators was developed to use them. Pull the 8088 in your exact PC compatible (but not the dual-speed H/Z-158, which uses an 8088-2 microprocessor), load the emulator, and run the CP/M program you have transferred to the MS-DOS disk. More — still not all, but most — instructions will be supported and, therefore, many more programs can be run.

I keep implying that there are limitations to any CP/M emulator. The fundamental limitation that none of them overcomes arises from The Hardware that Isn't. Take, for example, a CP/M communications program that calls a specific port address. Since no one is home at that address, and since no one ever will be home there, nothing good can happen from the call. Whenever a CP/M program directly addresses hardware unique to a specific machine or, worse, to an odd configuration of a specific machine, nothing good will happen. As a general rule, forget about using CP/M graphics or communications programs on MS-DOS with an emulator.

All right. What we have so far is a generation of tools that requires a two-stage process to make CP/M files usable on a mainstream computer: first, copy files from the CP/M disk to an MS-DOS disk; second, run program files under a CP/M emulator. The second stage actually requires more from the user. Depending on the emulator, there are two ways to go. One involves renaming command files so they have a unique extension: a convention is to replace ".COM" with ".CPM" so "WS.COM" is renamed to "WS.CPM." Then the emulator is executed to load a "shell" — a translation program — which simulates the CP/M environment. Then you issue the command, "WS." The other way available in some emulators is to make the shell part of the CP/M program so it carries the environment whenever you run the program. Either way, you wind up with an additional, modified, copy of the program.

Micro Interfaces' Run/CPM series is a new generation of CP/M emulators, which extend the capabilities of mainstream computers to encompass CP/M software. You don't have to copy either data files or programs from CP/M to MS-DOS: the system is given a "smart" shell that knows which set of instructions apply in each situation. You can go through the two stages required by earlier emulators (although Run/CPM is not one that will install a shell onto the CP/M programs directly), but you don't have to.

You do have to modify your machine to use the Run/CPM series, but the modification is simple and reversible. I've been using Run/CPM on our Z-152 and Run/CPM-Z80 on our H-158: the former is the 8080 emulator; the latter is the Z-80 emulator. For Run/CPM you replace the 8088 with the supplied 8MHz NEC V20 microprocessor: pull the 8088 with the chip-puller included in the package, and plug in the V20. It's required for use of the Run/CPM software, and anyway it's a popular modification to speed up processing on exact PC compatibles, generally. (When I told Dan Robey of Micro Interfaces that many owners of Heath/Zenith compatibles already use a V20 and won't need another, he said he'll sell the software alone for a slight discount. Tell him I sent you.) For Run/CPM-Z80 you install the supplied Z-80 coprocessor board in a vacant slot on the backplane. You can use Run/CPM-Z80 on any of Heath/Zenith's mainstream computers — including the AT-compatible 241 — but don't use Run/CPM on the H/Z-158 or the PC-200 series: use it only on compatibles that have the 8088.

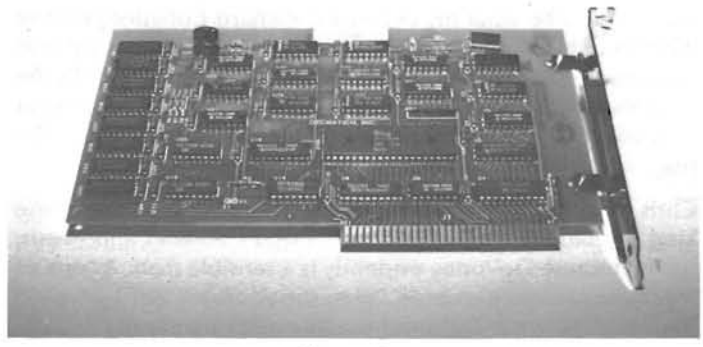


Figure 4
With Run/CPM-Z80, the Z-100 PC can run CP/M programs from most computers — including Z-80 machines.

Then install the software with the nearly-foolproof installation program. It does some trivial but essential things like set the environment in COMMAND.COM to prepare for execution of the emulator itself. Run it — the commands are "RUNCPM" for

Run/CPM and CPMBRD5 for Run/CPM-Z80 — when you are in a session that requires use of CP/M. Then when you want to use a CP/M program or data file, put the CP/M diskette in a disk drive and toggle the drive to emulate the CP/M format you need: ALT R pops up a menu with your choices. You can run programs, read files, write them, copy them, and — in general — do just about what you could do with them on the CP/M machine. You can even format CP/M disks. You now have CP/M capabilities on your computer.

Are there limitations other than the universal limitation posed by programs that address Hardware that Isn't? Yep. The major limitation is that everything goes slower. Oddly, Run/CPM moves faster than Run/CPM-Z80, I think. The 5MHz Run/CPM-Z80 is slow, but probably tolerable for occasional use or use with absolutely irreplaceable software. There's a 9MHz Run/CPM-Z80 that ought to process the CP/M side faster, but I don't know because I haven't seen it. Once a program is running, things seem to go about as fast as always, but my Run/CPM-Z80 takes its own sweet time getting to that point. Don't forget that I said everything goes slower: on the MS-DOS side too. Before I Run/CPM or Run/CPM-Z80, I load a program that marks the place in RAM where the emulator will begin. Then, when I'm finished with my CP/M session, I run another program that unloads the emulator — which brings the system back to speed.

Nevertheless, Run/CPM and Run/CPM-Z80 are little marvels. Incidentally, they include the only programs that can read many of the Kaypro CP/M disk formats directly. Kaypro, bless them, seems to have screwed up some of its disk formatters, a "feature" which has created problems for me in many endeavors. All the disk conversion software I have seen, outside the ones built into the Run/CPM series, note the problem and provide an inconvenient workaround: first you use the program's formatter to prepare a fresh diskette in Kaypro format — the way the Kaypro would have done it if it could do it the right way; then you return to the Kaypro and copy onto that new diskette the files you want to use; and at last you use the conversion program again to transfer the Kaypro files to MS-DOS. I've done it enough over the past couple of years so that I would buy Run/CPM as a disk converter alone. The Run/CPM series is the best I've seen yet.

Using PC DOS

I don't think much of most "user's guides" to operating systems: they tend to be what my old friend Richard Congdon, former director of Northern Illinois University Press, used to sneer at as "non books." Better you should read the manuals, especially the ones that Heath/Zenith provides. Sure, they're not as good as they used to be — but, then, what is? They're still a cut above the rest I've seen.

Chris DeVoney's "Using PC DOS" is exceptional. It seems the kind of guide that will grow with a user's abilities and needs, partly because DeVoney evidently is a sensible man. As one of my colleagues once erupted, "How can you press 'em if you can't find 'em?" So when DeVoney explains you can reboot the computer by pressing "CTRL ALT DEL," he shows you a sketch of the keyboard with arrows pointing to the pertinent keys on the PC and AT. Sketches like these resemble IBM compatibles more than Heath/Zenith's, but I suspect that most readers of our persuasion will get the relevant points anyway. As is true of compatibles themselves, the worth of DeVoney's book transcends insignificant differences among compatible machines. Experienced users should benefit from "Using PC DOS" too, both as a browser on MS-DOS Version 3 and as a practical reference on Versions 2 and 3. By "practical" I mean a reference containing solid facts, derived from investigation, such as a table of how SORT collates characters in the range 128-255. I didn't know that. "Using PC DOS" has a lot more I didn't know. Besides, DeVoney writes in the English language.

It's interesting how my status in the household has risen considerably since U.P.S. delivered a Summagraphics sketch pad and Summagraphics programmable optical mouse. "Summagraphics" is a Big Name in the world of graphics design, which is Janet's world. When she goes to shows of professional graphics equipment, manufacturers whose products employ a mouse or graphics tablet invariably use Summagraphics' in demonstrations. That's interesting because it means they literally are banking on Summagraphics. Janet is much impressed. So is Matthew: "At last! We can do some real graphics around here!" And just what does he think his mother has been doing all these years? And just what does he mean by "we?"

Products Discussed

ZOOM/Modem PC1200 ST $299
ZOOM/Modem PC1200 XL $399
ZOOM Telephonics, Inc.
207 South Street
Boston, MA 021111
800/631-3116

MonoGrafx Vers 3.0. Copy-protected $ 69.95
Uncopy-protected $ 99.95

Run/CPM Vers 6.1. With 8MHz NEC V20 $ 99.95
Without NEC V20 $ 89.95
Analytics International, Inc.
1365 Massachusetts Avenue
Arlington, MA 02174
800/992-0085

Run/CPM-Z80 $199.95
Micro Interfaces Corp.
6824 N.W. 169th Street
Miami, FL 3015
800/637-7226

Using PC DOS
by Chris DeVoney
Paperback. ISBN 0-88022-170-4.
$ 21.95
Que Corporation
7999 Knue Road
Indianapolis, IN 46250
800/428-5331

Faster Disk Access For Heath/Zenith MS-DOS

Pat Swayne
HUG Software Engineer

This article is the result of a letter I received from a HUG member who complained that when he ran IBM's PC-DOS on his Z-100 PC series computer, disk access was faster than when he ran Heath/Zenith MS-DOS. I began looking into ways to speed up MS-DOS disk access on PC-type computers, and when I had accomplished that, I also looked into speeding up access on Z-100 (dual processor) computers. The speed-up methods that are discussed in this article are the result of my efforts. The methods used for the two "families" of computers (PC and Z-100) are quite different, and will be explained in separate sections.

Faster Access For PC-type Computers

All PC-type computers, including the H/Z-100 PC and H/Z-200 PC series, use DMA (direct memory access) for disk I/O. This means that disk access should be quite fast, if things are done properly. The main things that slow down disk access are delays that are put in on purpose. One of the delays is added to give the disk drive heads time to "bounce" each time they are loaded, and another delay provides time for the disk motor to start. In the Heath/Zenith versions of MS-DOS, both of these delays are larger than they have to be. The head bounce delay is especially large. I suppose the delays were made large so that the operating system would work with several drives from different manufacturers. However, the drives supplied by Heath/Zenith do not require such long delays. There are two drive types supplied. One is the Shugart type (a lever is rotated to lock and release disks), and its heads remain loaded all the time, so a head bounce delay is not needed at all. The other drive type is from Mitsubishi (a rectangular plate is pressed to release disks), and works with little or no head bounce delay.

The head bounce delay slows access down more than it actually should, because it seems to be applied every time the system commands the drive to seek a track, regardless of whether the head is already loaded or not. As you may know, especially if you have Mitsubishi drives, the system signals the drive to keep the head loaded for a few seconds after each access, so that the head will not "bang" repeatedly during rapid multiple accesses.

The delays can be patched using DEBUG. You can patch them temporarily, to determine the effect on your system, or patch them permanently. To patch the delays temporarily, use DEBUG as shown in this example:

```
-E0:579
0000:0579  0D.00   04.01
-Q
```

This example is for a Z-100 PC series computer. The head bounce delay value is located at 0:579, and the default value is 0D. In the example, we have patched it to 0. The motor start delay value follows immediately, and we have patched it from 4 (which causes a 1 second delay) to 1 (1/4 second delay). Try copying files and formatting disks using the /V switch with FORMAT to test system performance. If you have Mitsubishi drives, and your system gives any indication of trouble, change the head bounce delay (at 0:579) to 1. If you have a Z-200 PC series computer, your delays are at the same location, but the default values are different, as shown in this example:

```
-E0:579
0000:0579  0F.00   08.02
-Q
```

The default head bounce value was 0F, and we patched it to 0. The motor start delay value was 08, and we patched it to 02.

A good test to compare performance with and without the delay changes is to format a disk with the /V switch and time the operation. Do this with the delays patched, and then re-boot (which will reset the delays to the old values) and time the formatting again. If you listen to your drive as it steps, you may be able to tell that the actual formatting of the disk is done at about the same speed each time, but the verification of the disk after formatting will be much faster with the delays reduced.

The delays can be patched permanently by patching the part of your system disk that supplies the values stored at 0:579. DEBUG can be used to make the patch, but the procedure is a little complicated, so I wrote a program called MAKFST that can patch the delays for you. The assembly source code for MAKFST is at the end of this article, but if you do not have an assembler, you can create MAKFST.COM by entering and running the following BASIC program:

```
10 REM THIS PROGRAM CREATES MAKFST.COM
20 DEFINT A-I:OPEN "O",1,"MAKFST.COM
30 S=0:S1 = 8095 .FOR I=1 TO 78
```

```
40 READ B:S=S+B:PRINT #1,CHR$(B);
50 NEXT I:IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 160,92,0,254,200,51,210,185,64,0
80 DATA 187,78,1,205,37,114,59,191,78,1
90 DATA 185,0,128,176,223,187,2,37,186,2
100 DATA 9,252,242,174,117,40,57,29,117,248
110 DATA 57,85,2,117,243,160,93,0,36,15
120 DATA 136,69,8,160,109,0,36,15,136,69
130 DATA 9,160,92,0,254,200,51,210,185,64
140 DATA 0,187,78,1,205,38,205,32
```

The MAKFST.COM generated by this program is for use on H/Z-100 PC series computers. For H/Z-200 PC series computers, change lines 30 and 100 to read as follows:

```
30 S=0:S1 = 8101 :FOR I=1 TO 78
100 DATA 15,252,242,174,117,40,57,29,117,248
```

After you make MAKFST.COM, copy it to your system disk, and run it to change delays using this syntax:

```
MAKFST d:n,n
```

where d: is a drive designation, and n,n are the values for the head bounce and motor on delays. For example, if you want delays of 0 and 1 on a disk in drive A:, you would enter

```
MAKFST A:0,1
```

It is probably a good idea to back up any disk before using MAKFST on it. You can alter the delays on a hard (Winchester) disk, as well as on a floppy, but only floppy disk access is affected in either case. MAKFST makes the changes to your disk only, not to memory, so you must reset and re-boot before the changes will take affect.

Faster Access For H/Z-100 Computers

Unlike PC-type computers, H/Z-100 series computers do not use DMA for disk I/O, and patching the head bounce and motor on delays has little, if any, affect on disk speed. The system actually works as fast as it can. There is, however, one thing that affects disk access speed that can be changed. When the drive head steps from one track to the next, the system used for accessing the disk is slow enough that the disk rotates beyond the first sector before the system is ready to look for it. The disk must, therefore, rotate one full turn to allow the first sector to come around again. If the sectors could be recorded in a skewed fashion on the disk surface, so that sector number 1 on the second track is displaced a couple of sectors from where sector 1 on the first track is, then the disk would not have to rotate a full turn. This "trick" was used successfully to increase disk access speed on the old H89 series computers under the HDOS operating system, so the idea is not new.

When a disk is formatted on an H/Z-100 computer under MS-DOS, the format program makes a call to the BIOS (part of the operating system) to perform actual formatting for each track on the disk. I have written a program called FASTFORM that remains resident in memory once it has been loaded, and intercepts the calls to the BIOS to format disk tracks. It rearranges the "track image" that is used to fill in the sectors on the disk, so the sector numbers are shifted by 2 whenever the head must change tracks, and by 1 whenever the head must change sides. Normally, the system is fast enough not to miss the first sector when the head changes sides, because there is always an index hole gap at this point. With the sectors skewed, however, there will not always be an index hole gap at each side change, so a skew of 1 sector is needed.

The source code for FASTFORM is listed at the end of this article, but if you do not have an assembler, you can create FASTFORM .COM by entering and running the following program:

```
10 REM THIS PROGRAM CREATES FASTFORM.COM
20 DEFINT A-I:OPEN "O",1,"FASTFORM.COM
30 S=0:S1 = 27176  FOR I=1 TO 318
40 READ B:S=S+B:PRINT #1,CHR$(B);
50 NEXT I:IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 233,225,0,0,0,0,0,0,0,1
80 DATA 0,0,64,0,70,70,251,156,60,5
90 DATA 116,31,60,6,116,18,46,198,6,9
100 DATA 1,1,46,198,6,6,1,0,46,198
110 DATA 6,7,1,0,157,46,255,30,10,1
120 DATA 202,2,0,80,81,86,87,30,83,6
130 DATA 14,31,128,62,6,1,0,117,34,38
140 DATA 138,7,162,3,1,80,176,8,255,30
150 DATA 10,1,88,50,228,209,224,3,216,38
160 DATA 139,31,38,139,71,20,163,4,1,7
170 DATA 91,83,6,128,62,3,1,4,115,108
180 DATA 252,38,138,71,1,8,6,7,1,162
190 DATA 8,1,38,196,127,5,138,38,9,1
200 DATA 138,62,6,1,10,255,116,2,254,199
210 DATA 50,219,139,14,4,1,176,245,242,174
220 DATA 117,30,38,129,61,245,245,117,245,38
230 DATA 128,125,2,254,117,238,38,136,101,5
240 DATA 254,196,58,231,117,2,180,1,254,195
250 DATA 235,220,136,30,6,1,160,9,1,254
260 DATA 200,117,2,138,195,128,62,7,1,0
270 DATA 116,7,128,62,8,1,0,116,6,254
280 DATA 200,117,2,138,195,162,9,1,7,91
290 DATA 31,95,94,89,88,233,72,255,30,51
300 DATA 192,142,216,190,72,2,196,60,38,129
310 DATA 125,254,70,70,117,2,205,32,199,4
320 DATA 16,1,140,76,2,176,8,154,72,0
330 DATA 64,0,184,128,0,6,31,139,63,137
340 DATA 69,26,139,127,2,137,69,26,139,127
350 DATA 4,137,69,26,139,127,6,137,69,26
360 DATA 190,73,0,139,60,129,199,75,0,78
370 DATA 199,4,205,146,198,68,2,203,31,137
380 DATA 62,10,1,186,228,1,205,39
```

Once you have made FASTFORM.COM, copy it to your system disk, and load it into memory by entering

```
FASTFORM
```

at the system prompt, and hitting RETURN. You will only get increased disk access speed on disks that are formatted after FASTFORM has been installed. Both FORMAT and DISKCOPY use the BIOS routines intercepted by FASTFORM to format disks, so you can DISKCOPY an existing disk to make a faster version of it.

You will save the most time during DISKCOPY operations when the source disk has been formatted with FASTFORM loaded, and DISKCOPY is used with it loaded. If you use your H/Z-100 to copy a lot of disks, as in manufacturing software products, FASTFORM can save you quite a bit of time when it is all added up. You will save more time when duplicating single-sided disks than when duplicating double-sided disks (as compared to duplicating without FASTFORM), because the drive head must step on every track change on a single-sided disk. If you use the HUG Screen Clock program or another background utility that puts any drain on processor time, you should disable it (turn it off) during DISKCOPY operations, or your time savings will be reduced.

There will be only a little savings in time during normal disk operations when you use FASTFORM formatted disks, and these disks will be read a little slower than normal ones on a PC-type computer on which the disk delays have been shortened, as described in the first part of this article. But they can be read on any

MS-DOS system, so you can use FASTFORM if you manufacture disk software on an H/Z-100, regardless of where it will be used.

MAKFST and FASTFORM Source Code Listings

```
        PAGE    ,132
;       MAKFST -- MAKE HEATH/ZENITH MS-DOS FAST
;
;       THIS PROGRAM PATCHES THE DISK DRIVE HEAD BOUNCE
;       AND MOTOR ON DELAYS IN MS-DOS.
;
;       TO USE THIS PROGRAM, ENTER
;
;         MAKFST d:n,n
;
;       WHERE d: IS A DRIVE CODE, AND n,n ARE THE VALUES
;       FOR THE HEAD BOUNCE AND MOTOR ON DELAYS.
;
;       BY P  SWAYNE, HUG SOFTWARE ENGINEER  11-JUN-86

DSKRD   EQU     25H             ;DISK READ INTERRUPT
DSKWRT  EQU     26H             ;DISK WRITE INTERRUPT

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     5CH
DFCB    LABEL   BYTE            ;DEFAULT FCB ADDRESS
        ORG     100H

START:  MOV     AL,DFCB         ;GET DRIVE CODE
        DEC     AL              ;CORRECT IT
        XOR     DX,DX           ;START WITH SECTOR 0
        MOV     CX,64           ;READ 64 SECTORS (32K)
        MOV     BX,OFFSET BUFFER;PUT IT HERE
        INT     DSKRD           ;READ FROM DISK
        JC      EXIT            ;BAD READ
        MOV     DI,OFFSET BUFFER;POINT TO BUFFER
        MOV     CX,8000H        ;SEARCH 32K
        MOV     AL,0DFH         ;LOOK FOR THIS
        MOV     BX,2502H        ;AND THIS
        MOV     DX,902H         ;AND THIS
;       **USE 0F02H ABOVE FOR Z-200**
        CLD                     ;SEARCH FORWARD
SCHLP   REPNZ   SCASB           ;LOOK FOR DISK PARMS
        JNZ     EXIT            ;NOT FOUND
        CMP     [DI],BX         ;IS THIS IT?
        JNZ     SCHLP           ;NO
        CMP     2[DI],DX        ;CHECK NEXT WORD
        JNZ     SCHLP           ;NOT IT
        MOV     AL,DFCB+1       ;GET FIRST DELAY VALUE
        AND     AL,0FH          ;REMOVE ASCII
        MOV     8[DI],AL        ;FIX DELAY
        MOV     AL,DFCB+17      ;GET SECOND DELAY VALUE
        AND     AL,0FH          ;REMOVE ASCII
        MOV     9[DI],AL        ;FIX DELAY
        MOV     AL,DFCB         ;GET DRIVE CODE
        DEC     AL              ;CORRECT IT
        XOR     DX,DX           ;START WITH SECTOR 0
        MOV     CX,64           ;WRITE 64 SECTORS (32K)
        MOV     BX,OFFSET BUFFER;DATA IS HERE
        INT     DSKWRT          ;WRITE TO DISK
EXIT:   INT     20H             ;AND EXIT
BUFFER:                         ;DISK READ/WRITE BUFFER

CODE    ENDS
        END     START
```

```
        PAGE    ,132
;       FASTFORM -- PROGRAM TO CAUSE FORMAT AND DISKCOPY
;       TO MAKE FAST DISKS
;
;       THIS PROGRAM INTERCEPTS FORMAT TRACK CALLS TO THE
;       BIOS, AND ALTERS THE TRACK IMAGE IF SIDE 1 IS
;       FORMATTED   THE RESULT IS THAT SECTOR NUMBERING IS
;       OFFSET BY 2 SECTORS WHEN THE DRIVE HEAD MUST STEP,
;       AND BY 1 SECTOR WHEN THE DRIVE CHANGES SIDES
;
;       BY P  SWAYNE, HUG SOFTWARE ENGINEER  25-APR-86

BIOS    SEGMENT AT 40H
        ORG     48H
DSKFUNC LABEL   FAR             ;BIOS DSKFUNC OFFSET
BIOS    ENDS

RETFD   MACRO
        DB      0CAH            ;FAR RETURN WITH DISPLACEMENT
        ENDM

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     100H

START:  JMP     SETUP           ;SET UP THIS PROGRAM

TRKSIZ  DW      0               ;TRACK SIZE
SPT     DB      0               ;SECTORS/TRACK
SIDFLG  DB      0               ;DOUBLE SIDE FLAG
SIDE    DB      0               ;CURRENT SIDE
FRSTNO  DB      1               ;FIRST SECTOR NUMBER

DFADR   DW      0,40H           ;ORIGINAL DSKFUNC ADDRESS

;       INT 92H PROCESSOR
;       BIOS DSKFUNC IS SENT HERE

        DB      'FF'            ;IDENTIFIER
INT92:  STI                     ;ENABLE OTHER INTERRUPTS
        PUSHF                   ;SAVE FLAGS
        CMP     AL,5            ;FORMAT TRACK?
        JZ      MYFORM          ;YES
        CMP     AL,6            ;STEP IN?
        JZ      GOBIOS          ;YES, DO IT
        MOV     CS:FRSTNO,1     ;ELSE, RESET FIRST SEC  NO
        MOV     CS:SPT,0        ;AND SECTORS/TRACK
        MOV     CS:SIDFLG,0     ;AND SIDE FLAG
GOBIOS: POPF                    ;RESTORE USER'S FLAGS
        CALL    CS:DWORD PTR DFADR      ;GO TO BIOS
        RETFD                   ;RETURN, FLAGS INTACT
        DW      2
MYFORM: PUSH    AX              ;SAVE SOME REGISTERS
        PUSH    CX
        PUSH    SI
        PUSH    DI
        PUSH    DS
        PUSH    BX
        PUSH    ES
        PUSH    CS
        POP     DS              ;PUT DS HERE
        CMP     SPT,0           ;SECTOR/TRACK SET?
        JNZ     NOTFT           ;YES
        MOV     AL,ES:[BX]      ;ELSE GET DRIVE CODE
        CMP     AL,2            ;DRIVES 0-1 ALLOWED
        JB      DRVOK           ;DRIVE OK
        JMP     NOFORM          ;AND EXIT
DRVOK:  PUSH    AX              ;SAVE DRIVE CODE
        MOV     AL,8
        CALL    DWORD PTR DFADR ;GET DISK PARAMETERS
        POP     AX
        XOR     AH,AH           ;AX = DRIVE
        SHL     AX,1
        ADD     BX,AX           ;FIX POINTER
        MOV     BX,ES:[BX]      ;GET TABLE FOR THIS DRIVE
        MOV     AX,ES:20[BX]    ;GET TRACK SIZE
        MOV     TRKSIZ,AX       ;SAVE IT
        POP     ES              ;FIX ES
        POP     BX              ;AND BX
        PUSH    BX
        PUSH    ES
NOTFT:  CLD                     ;FORWARD SEARCHES, ETC
        MOV     AL,ES:1[BX]     ;GET SIDE FLAG
        OR      SIDFLG,AL       ;SET OUR FLAG
        MOV     SIDE,AL         ;SAVE CURRENT SIDE
        LES     DI,ES:DWORD PTR 5[BX];GET TRACK IMAGE ADDRESS
```

```
        MOV     AH,FRSTNO       ;GET FIRST SECTOR NUMBER
        MOV     BH,SPT          ;AND SECTORS PER TRACK
        OR      BH,BH           ;SET YET?
        JZ      SPTNS           ;NO
        INC     BH              ;ELSE, ADD 1
SPTNS:  XOR     BL,BL           ;CLEAR A COUNTER
        MOV     CX,TRKSIZ       ;SEARCH THIS MUCH
SFHD.   MOV     AL,0F5H         ,SEARCH IMAGE FOR THIS
SFHD1:  REPNZ   SCASB           ;SEARCH FOR SECTOR HEAD
        JNZ     TRKDN           ;NOT FOUND, END OF TRACK
        CMP     ES:WORD PTR [DI],0F5F5H ;IS THIS IT?
        JNZ     SFHD1           ,NO, KEEP LOOKING
        CMP     ES:BYTE PTR 2[DI],0FEH
        JNZ     SFHD1
        MOV     ES:BYTE PTR 5[DI],AH ;FIX SECTOR NUMBER
        INC     AH              ,INCREMENT NUMBER
        CMP     AH,BH           ,PAST MAX?
        JNZ     SECOK           ;NO
        MOV     AH,1            ;ELSE, START OVER
SECOK:  INC     BL              ;COUNT THIS SECTOR
        JMP     SFHD            ,AND KEEP LOOKING
TRKDN:  MOV     SPT,BL          ;SAVE SECTORS/TRACK
        MOV     AL,FRSTNO       ;GET FIRST SECTOR NUMBER
        DEC     AL              ;SUBTRACT 1
        JNZ     SETF            ;OK TO USE
        MOV     AL,BL           ;ELSE, SET TO MAX
SETF.   CMP     SIDFLG,0        ;DOUBLE SIDED DISK?
        JZ      SNGSD           ;NO
        CMP     SIDE,0          ;SIDE 0 NOW?
        JZ      SETF1           ,IF SO, DONE WITH SEC  NO.
SNGSD.  DEC     AL              ;ELSE, SUBTRACT 1
        JNZ     SETF1           ;OK TO USE
        MOV     AL,BL           ;ELSE, SET TO MAX
SETF1:  MOV     FRSTNO,AL       ,NEW FIRST SECTOR NO
NOFORM  POP     ES              ,DONE, FIX REGISTERS
        POP     BX
        POP     DS
        POP     DI
        POP     SI
        POP     CX
        POP     AX
        JMP     GOBIOS          ;LET BIOS WORK

ENDRES                          ,END OF RESIDENT CODE

;       SET UP FOR FASTFORM BY INTERCEPTING DSKFUNC CALL

SETUP:  PUSH    DS              ;SAVE DS
        XOR     AX,AX
        MOV     DS,AX           ;POINT TO INTERRUPT SEGMENT
        MOV     SI,OFFSET 92H*4 ,VECTOR FOR INT 92
        LES     DI,DWORD PTR [SI]       ;GET WHAT'S THERE
        CMP     ES:WORD PTR -2[DI],'FF' ;CHECK FOR FASTFORM
        JNZ     NOTIN           ;NOT CURRENTLY INSTALLED
        INT     20H             ;ELSE, EXIT
NOTIN.  MOV     WORD PTR [SI],OFFSET INT92 ;SET VECTOR HERE
        MOV     2[SI],CS
        MOV     AL,8
        CALL    DSKFUNC         ;GET DISK TABLE ADDRS
        MOV     AX,400H         ,NEW DELAY VALUE
        PUSH    ES
        POP     DS              ;POINT DS TO BIOS
        MOV     DI,[BX]         ;GET DRIVE A  TABLE
        MOV     26[DI],AX       ,SET NEW DELAY
        MOV     DI,2[BX]        ;GET DRIVE B: TABLE
        MOV     26[DI],AX       ;SET NEW DELAY
        MOV     SI,OFFSET DSKFUNC+1
        MOV     DI,[SI]         ;GET DSKFUNC ADDRESS
        ADD     DI,OFFSET DSKFUNC+3 ;MAKE IT ABSOLUTE
        DEC     SI              ;BACK UP TO "JMP"
        MOV     WORD PTR [SI],92CDH ;REPLACE WITH "INT 92"
        MOV     BYTE PTR 2[SI],0CBH ,AND "RETF"
        POP     DS              ;FIX DS
        MOV     DFADR,DI        ;SAVE OLD DSKFUNC ADDR
        MOV     DX,OFFSET ENDRES;POINT TO END OF CODE
        INT     27H             ;EXIT, CODE RESIDENT

CODE    ENDS
        END     START                                    *
```

Packet Radio Description And A Simulator

Part 2

Luis E. Suarez, OA4KO/YV5
Apartado 66994
Caracas 1061-A
VENEZUELA

A Packet Simulator In BASIC

There are several TNCs on the market, including the Heath HD-4040 which is based on the TNC-1 developed by Tucson Amateur Packet Radio. The program PACKDEM.BAS, presented here, is based on the last release of TAPR, the TNC-2. See also the listing of equipment sources at the end of this article.

PACKDEM.BAS simulates a limited TNC-2 operation. There is no possibility to control a radio or to packetize information to be sent to a modem. To make the simulation more realistic, I have used, when necessary, a split screen operation to simulate the screen display of the correspondent station you are in contact with.

I have developed the program on my Heath H-89 under CP/M and MBASIC version 5.2. However, it may be easily converted to other BASIC versions. No more than 48K of memory RAM is necessary and makes no use of disk I/Os, as explained.

Not all the TNC-2 commands are implemented in this program. However, all commands may be called. Since the simulator cannot transmit packets, it is understandable that there is no reason to code an algorithm not needed, but you can execute, and hence change all default parameters, even if some of them are really not used (yet) in this program. In any case, if you like to experiment, just go ahead and develop the subroutine, according to the command you like to implement.

The program is coded in a structured layout. This approach certainly saves time, if you would ever like — or need — to modify it. Hence, all subroutines corresponding to the TNC-2 commands are located from line 600 to line 1480. One, or more, for each command available. Most routines are very simple and need no further explanation.

From line 1500 to line 1575 you find the Error Calls. Lines 1580 to 1700 are devoted to Message Calls. Not all the calls are implemented in the program. Hence, this is an area of further development, too. All Error Calls, as stored in the TNC-2 memory, are available in the program.

The Random Problem

At line 1960, I have used a trick to reseed the RND(n) function of my computer. The reason is, random generator numbers are prone to be repetitive and, hence, predictable. That's why I use, in this and all my programs, the statement RANDOMIZE PEEK(12). The problem is that each version of BASIC handles the random generator function in a different way. In my case, MBASIC Version 5.21, the situation is as follows:

```
RND(n)      n<Ø restart; n=Ø repeat; n>Ø next number
RANDOMIZE   reseed random number generator
```

In other words, using RND(-1) will return the same sequence each time the line is executed, since the seed number has not been changed. If the random number generator is not reseeded, the RND function returns the same sequence of random numbers every time the program is run. For MBASIC version 4.83, the behavior is different:

```
RND(n)      n<Ø reseed; n=Ø repeat; n>Ø next number
```

If you run RANDOMIZE, the interpreter returns: "(Random Number Seed? -32768 to 32767) and waits for the keyboard input. If you run RANDOMIZE(nn), where nn is any number within the specified range, then the number generator is reseeded without waiting for input. I use RANDOMIZE PEEK(12) to ensure a new and different number is used as a seed each time line 1960 is executed. Address 12 corresponds to the system clock.

Running The Simulator

Well, I cannot describe the function of all 95 commands available in the TNC-2 firmware. Instead, I will go with a sample run as described in the system manual TAPR sent along with the equipment.

Of course, you must load the BASIC interpreter before loading the program, then enter RUN and your screen will display the heading. The carriage return <CR>, will be shown at the beginning for clarity, but is omitted thereafter in this writing. Your input is shown in lowercase letters, and the TNC-2 output is shown in uppercase for clarity, too. In fact, you may use upper or lowercase at any time, as the real TNC-2. The limitation regarding this matter is that contrary to the TNC-2, the program accepts whole word commands only. The real thing accepts command attributes in either hexadecimal or decimal and are displayed in hexadecimal. The program accepts decimals only. Comments in the sample run are enclosed within brackets.

```
  !A

bbRAM loaded with defaults

Tucson Amateur Packet Radio TNC 2
AX.25 Level 2 Version 2.0
Release 1.1.1 - 16K RAM
Checksum $7F
cmd:<CR>
cmd:

[You are now in the command mode, as noted by the prompt
 'cmd:', and any time you hit return, another prompt will
 be displayed  Now, continue.]

cmd:monitor off
MONITOR was ON
cmd:mycall
MYCALL      NOCALL
cmd:mycall oa4ko
was         NOCALL
cmd:mycall
MYCALL      0A4KO
cmd:

[First, you have disabled the MONITOR option  Don't worry,
 nobody will be hurt and we'll go to MONITOR very shortly
 You have  entered your call  (of course, you may use your
 Amateur Radio call letters)  and the TNC-2 will use it as
 your station address.  From now on, all packets addressed
 to your station will be acknowledged by the TNC ]

cmd:awlen
AWLEN 7
cmd:parity
PARITY 3
cmd:

[You have verified that your system is set to 8 bit words
 and parity set to no parity bit.]

cmd:autolf
AUTOLF    ON
cmd:screenln
SCREENLN  80
cmd:nucr
NUCR      OFF
cmd:nucr on
NUCR was  OFF
cmd:

[Some terminals require a linefeed sent after a carriage
 return  If yours does not, you must change AUTOLF to OFF.
 If your terminal is 80 characters wide you are okay,
 otherwise, change it  Similarly, your terminal may need
 an extra null after <CR>, if so use NUCR ON.]

[Now let's try a basic Connecting and Disconnecting
 exercise as described in the system manual  Advice is
 in order  I have used a split screen operation, just
 to show you what is received by the correspondent
 station  In reality, your TNC-2 does not implement
 such operation  It is obvious that in real-life TNC-2,
 you don't need to see what the correspondent station is
 receiving.  Using the simulator, any time you connect
 with another station, the upper portion of your display
 will be erased and that portion will be used to print in
 what is supposed to be received at the other station  If
 connected to yourself, no split operation is performed.]

cmd:mycall
MYCALL 0A4KO
cmd:connect oa4ko
*** CONNECTED to 0A4KO

[The "*** CONNECTED to" message tells you that a successful
 connection was done  In real-life this kind of self-
 connection is done by placing a jumper in a header that
 will link the modem input with the modem output or between
 the digital circuit input and digital circuit output,
 for checking purposes  In the program this is just a trick
 Okay, let's type a message ending with a <CR>  You will
 see that anything you type will be retransmitted.]

Hello, there.
Hello, there

[The <CR> causes your message to be packetized and trans-
 mitted. Any time you hit <CR> while connected, it will
 cause a packetization and a transmit command to your
 radio  To return to command control just hit CTRL-D (hold
 down the CONTROL key while typing the letter D.) Your
 TNC-2 will return to command mode as shown by the cmd.
 prompt.]

<CTRL-D>
cmd:convers
Whatever I type in convers mode is transmitted
Whatever I type in convers mode is transmitted.
Even a carriage return
Even a carriage return
<CR>
<CR>
<CTRL-D>
cmd:DISCONNE
*** DISCONNECTED

[Well, another command was necessary to disconnect and is
 DISCONNE. The successful disconnection is show by the
 message "*** DISCONNECTED"  You have experienced the
 CONNECT, CONVERS and DISCONNE commands  As observed, you
 returned to command mode, but the simulator was still
 connected with yourself. Then, when you entered the
 CONVERS command, you returned to that mode to continue
 the QSO with yourself  Finally, you successfully discon-
 nected with the DISCONNE command.]

[-When the CONVERS mode is used in the simulation, the
 split screen is enabled.- If you go directly to the
 CONVERS mode, your packets will not be addressed to any
 station  Your clever TNC-2 will send them to the address
 "CQ" (default of UNPROTO command)  If you're an Amateur
 Radio operator, you know what CQ means  calling anyone.]

cmd:monitor on
MONITOR was OFF
cmd:convers
Hello world   This is Packet!!!
0A4KO>CQ:Hello world   This is Packet!!!
I'm looking for somebody in the parking lot
0A4KO>CQ:I'm looking for somebody in the parking lot
<CTRL-D>
cmd:

[As you see, CONVERS is a clean procedure to call CQ. You
 don't need to care about your call letters, for they are
```

```
 already stored in memory  Now you can go ahead and call
 your buddy ]
cmd:connect wa7gxd
*** CONNECTED to WA7GXD
Hello Lyle, I have joined the Packet Radio Revolution!!!
Well, almost ...
<CTRL-D>
cmd:connect nk6k
Link state is: CONNECTED to WA7GXD
cmd:convers
Bye Lyle. Will see you on oscar 10
<CTRL-D>
cmd:disconne
*** DISCONNECTED
```

Well, maybe you were not so lucky as I was. It may happen that Lyle is busy with another station and the message "WA7GXD busy" will show such possibility. "Link state is: CONNECTED to WA7GXD" indicates that you cannot connect to another station while connected with a previous one. To do it, you must use a different connect stream (from A to J) to perform a multiconnect with up to ten stations in the same radio channel. This is possible with the USERS command that allows you to select the number of multiple-connections. The procedure is similar to the one used to configure the CP/M system to allow several drives. In Packet Radio, each stream is selected with a switch character identifier, like| A. This feature is exclusive of TNC-2 version 1.1.1, a big step toward radio networking.

Oops!! Yes, your MONITOR is ON and you may receive messages from other stations on the frequency. There is a lot of activity in packet nowadays.

What Else Within The Program?

Well, as explained, I cannot make a description of all 95 commands of the TNC-2. The System Manual from TAPR has many, many pages devoted to describe them. I can only scratch the surface of Packet Radio with a limited simulator like this. If you'd like to display all the commands available in the TNC-2, just type DISPLAY, and be ready to use CTRL-S to hold the screen, otherwise, they will scroll up very fast. Already implemented within the program are the following commands:

AUTOLF	-Used to control the carriage return characters.
BTEXT	-Text automatically sent if Beacon is enabled.
CALIBRA	-This command is used to calibrate the modem. You must set first the parameter with the CALSET command and then CALIBRA. To exit from CALIBRA type Q without a carriage return.
COMMAND	-A CTRL-D is used to return to command mode from CONVERS mode. You may change this value. If you use decimal 2, a CTRL-B will be necessary.
CONNECT	-Is an immediate command to initiate a connect request.
CR	-This is the send packet character. Should match your terminal requirements and must be selected along with AUTOLF.
CTEXT	-Text transmitted after requested connection. For example: "I'm not here now. Please leave a message. Will call after 8:00 pm."
DISCONNE	-Disconnection procedure.
DISPLAY	-Shows commands and their attributes. The TNC-2 shows values in hexadecimal, but attribute may be entered in either decimals or hexadecimals. The simulator program uses decimals only.
MHEARD	-List all stations worked.
MHCLEAR	-Clears MHEARD list.

MONITOR	-You may read traffic in the frequency.
MYCALL	-Call sign
PACLEN	-Length of text in bytes to be packetized is transmitted.
RESET	-Resets the TNC-2 to default command attribute.
SCREENLN	-Sets your screen length.
SENDPAC	-ASCII value to send Packets. Default is 13 or carriage Return. If you change to 20, a CTRL-T will send.
TRANS	-Transparent mode of operation. Useful for bulletin board operations and binary file transferring. To exit, type three CTRL-D or your selected COMMAND value, if different.
UNPROTO	-Default is CQ for the address of unconnected mode. CQ means, calling all stations.

Should you like to list the commands to your printer, just run the program and on the command prompt type CTRL-D to stop execution. Then type:

```
FOR A=1 TO 82:LPRINT A;") ";CMD$(A),ATRIB$(A):NEXT
```

Each command will be listed with the default values. I'm sure you will be intrigued to know what those remaining commands are intended for. Finally, let me tell you that the TNC-2 is so smart that the front panel has only four LEDs. They are used to sign Power On/Off, Connection, Status and Data Carrier Detection. The first is obvious. The CONnection LED indicates that a successful connection is established. The STAtus LED, informs you that there are still one or more packets in the TNC's memory to be acknowledged. The DCD LED tells you that the carriers (string of data bits) are detected. This is not RF carriers, but digital information detected by the microprocessor. The RF carrier is handled differently by the TNC. In FM, the Squelch is used to . . . Well, that's another story. You will enjoy simulating the LED's operation, if your computer has a 25th line or can protect fields. Just "light" the PWR LED at RUN. "Light" the CON LED anytime you make a connection (hint: *** CONNECTED to:.) "Light" the DCD LED anytime you receive a message.

Not bad for a limited demo program, is it? If you'd like to know what all the other commands are used for, just ask any packeteer in your area. (I'm not sure if TAPR is selling the Systems Manual separately, as they did with the TNC-1.) If you are a licensed amateur, try the frequencies listed elsewhere in this article, the fellow hams in the nets will be glad to help you and to discuss the command uses. Maybe you'd like to call TAPR, they will be glad to address you to the nearest packeteer in your geographical location. A visit to your neighboring Heath store will help, too. There are always amateur radio operators on the staff, or shopping around. Well, that's up to you now!

What's TAPR?

Packet Radio is possible thanks to the efforts of many Amateur Radio operators in the USA and Canada, but the excellence achieved, is especially due to a selected group of professional members of Tucson Amateur Packet Radio Corporation, a non-profit research and development organization founded in November 1981 to introduce Amateur Packet Radio to southern Arizona. Now TAPR is an international organization with members all around the world and certainly the leader in the Packet Radio development. All major Amateur Radio equipment manufacturers, including Heath/Zenith, are selling the controller designed by TAPR. The product is of very high technology and excellent quality. Besides, TAPR's service and support is far above the standards of commercial enterprises. Believe it or not, all the work is done by volunteers.

As TAPR's president, Lyle Johnson, said in an editorial in TAPR's newsletter (Packet Status Register), they started selling the first TNC-1 in late 1983, and 1-1/2 years later more than 2200 kits were sold. When they announced the availability of the new TNC-2, on August 19, 1985, the demand was so astonishing that the Tucson telephone system crashed several times in one day. The Davis Markham Air Force Base in Tucson could not talk to the world and all lines including WATTS were jammed, too! In two days, they accepted 650 orders! They sold 1200 units and now the unit is marketed by commercial enterprises. TAPR wizards are also developing a high speed RF modem and a powerful digital controller for Level 3 and Level 4 applications.

Current TAPR officers are Lyle Johnson, WA7GXD, President; Pete Eaton, WB9FLW, Executive VP; Pat Snyder, WA0TTW, Secretary and Dan Morrison, KV7B, Treasurer. The design of the last TNC was done in Florida (software), New Jersey (hardware), Chicago (modem) and Beta tested all around the world. No one is paid for their efforts, but the office manager, Christina Kurtz.

That's all folks! Maybe you will order your TNC-2 immediately and while waiting for it, you will be glad to play around with the program, implementing those extra subroutines. If so, maybe you will be kind enough in sharing the work with the fellow computer hackers. Of course, I'm interested too, but please include a SASE and 2 IRC's if a reply is expected. If enough interest is developed, I'll be glad to write another article about Packet Radio hardware, with a description of TAPR's TNC-2.

If you are not a Radio Amateur, let me tell you that computerism and radio communications will converge in a field named Telematics. Don't doubt that we all meet there, sooner or later. So, it is time for you to consider joining the ranks of ham radio. See you on the radio waves!

P-SST Maestro

Providing The Sounds Of Music To Your Z-100

John Britton
1705 Bonnyville Boulevard
Louisville, KY 40216

Most personal computers on the market today have the capability of producing a variety of sounds. These sounds include multi-voice music, special gaming sound effects, and even speech. The Heath/Zenith Z-100 was a notable exception until Software Wizardry came to the rescue with the P-SST accessory board and its associated software. Although the board comes with an assortment of support software, more was needed to produce real music. Therefore, Software Wizardry created P-SST Maestro, a software package with a music editor and supporting utilities. Software Wizardry has been advertising P-SST Maestro for nearly as long as the board has been in production. Finally, it is on the shelf of many distributors. If you own the P-SST board and want to more fully explore its music potential, P-SST Maestro is for you. If you don't own a P-SST board, read on anyway. To completely review Maestro, the P-SST hardware must also be reviewed. Hence, I'll answer the question "What is P-SST?", with emphasis on the sound production features, and then use that information to answer the question, "What is P-SST Maestro?"

What Is P-SST?

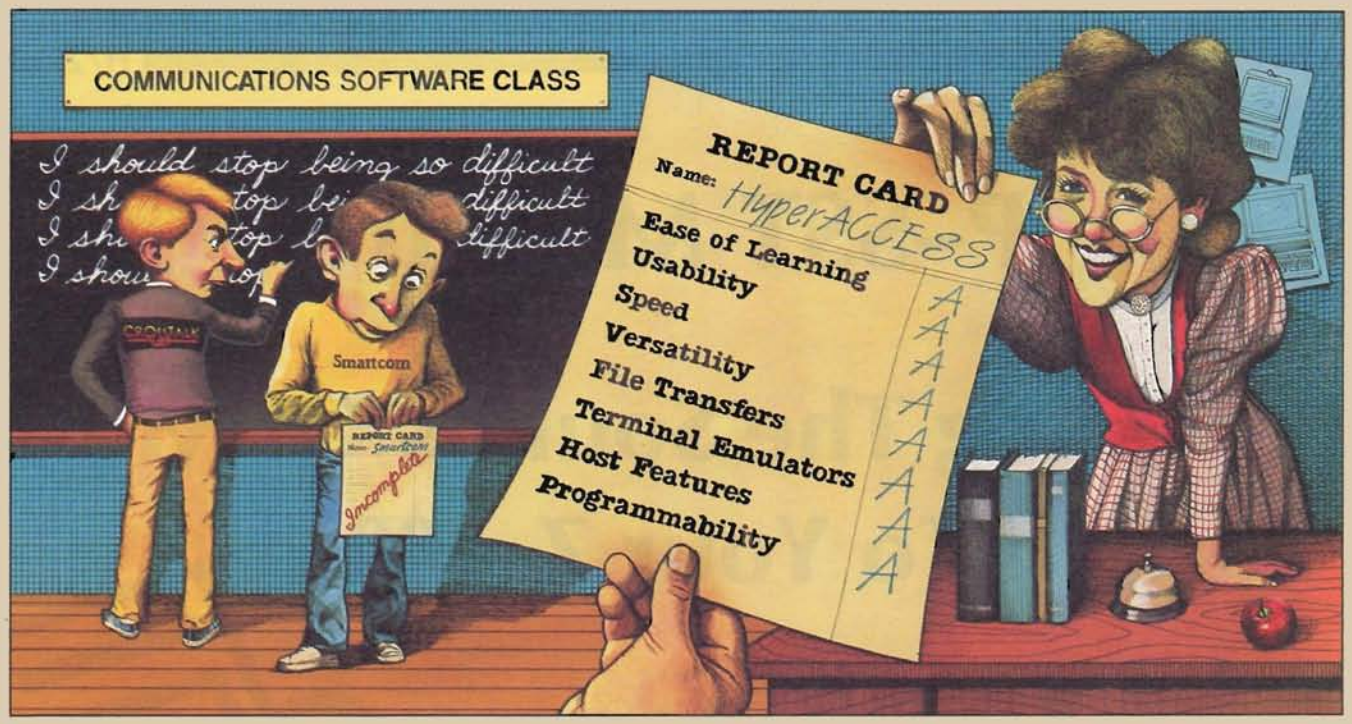
P-SST is a multifunction board for the Z-100 (S-100 bus) which offers Programmable Speech, Sound and Time (P-SST). It has a real time clock with battery back-up, a VOTRAX speech chip, two joy stick ports compatible with ATARI joy stick hook-up, and a three-channel sound synthesizer with both tone and noise generation. Your Z-100 must be equipped with MTR-100 ROM version 1.2 or higher and Z-DOS (MSDOS) version 1.10 or higher.

Since this review is designed to discuss what P-SST Maestro can and cannot accomplish, it will not discuss in detail the speech and time functions. However, a few words for those contemplating purchasing the board are in order. The speech function is generated by the VOTRAX speech chip, and is supported by several utilities using a simple text-to-speech algorithm. The algorithm has about 75% accuracy, but has features built-in to force it to provide good pronunciation of the entered text. The time function provides a real-time clock and a utility to access the date and time on boot-up. Also, there is an alarm register which generates an interrupt at the proper time. In addition, the alarm can generate a TTL output on one of the external ports. Using this port, it would not be difficult to have the computer turn on itself and the coffee pot, wake you with a rousing march, and then load your favorite program. Of course, to hear the march, a method of changing the signals generated by the board to audible sounds is required.

To hear the sounds played by the P-SST board, a 2-inch speaker may be connected directly to the board or, if preferred, a sound system may be connected to a separate port. I've tried both the small speaker, as well as connecting the board to my stereo system. The sound is much better through the stereo, as would be expected.

The manual for the P-SST board and its accompanying software is easy to read and logically laid out. It gives very good details of the inner workings of the board, plus schematic diagrams. Descriptions of all programmable registers and examples for using them are included. Enough information is provided to make it easy to develop custom software without a lot of pain. Incidentally, the manual makes extensive use of examples throughout, a valuable teaching aid lacking in many manuals.

The three-channel synthesizer has many methods of producing sound on any of its channels (voices). The foundation of making music entails changing frequency (pitch), amplitude (volume), and the duration of a particular sound. These controls are easy to program directly. However, the board also has a sound 'envelope' function. The sound envelope may be set to cause the sound to decay (loud-to-soft), or build (soft-to-loud), or cycle continuously loud-to-soft-to-loud . . . The rate of change is also



HyperACCESS Leads its Class in Every Subject

Whether you're just beginning to use your PC for communications or you've been frustrated by limitations of another program, HyperACCESS is the answer. Simple for novices. Exhilarating for experts. Ideal for businesses.

There's no initial hurdle. HyperACCESS **adapts itself** to your computer and modem (it dials, redials, and answers more than 40 models). No steep learning curve. You can use it just by looking at your screen. The manual is crystal clear. An amazingly detailed training program is included.

It comes ready to call CompuServe, Dow Jones, EasyLink, MCI Mail, NewsNet, The Source, and others (trial subscriptions included). Learns your passwords the first time you call and enters them for you from then on. You can list more than 100 computers (or people) for instant calling. HyperACCESS logs every call.

Astonishing versatility! Lets you link with systems that defy others. With mainframes, minis, micros, or RS-232C devices. It emulates VT-52, VT-100/102, TeleVideo 925/950, IBM 3101, H-19, and TTY terminals. It supports up to 19,200 baud and transfers files **FAST** using Kermit, Xmodem (CRC or Checksum), and ASCII Protocols.

HyperACCESS is an outstanding host, too. Even when you're not there, your PC can be accessed by callers who have passwords you've defined. You set whether each caller can read or type messages, transfer files, run programs*, or use your PC without restrictions.

HyperACCESS has the clearest, most complete script language, so you can create custom functions and menus, or automate any portion of a call. And we give you powerful scripts — transfer files between unattended PC's — pick up electronic mail at 2 am — act as a high-security dialback host, and more.

Even in the midst of communications, HyperACCESS lets you run your other DOS software or use your computer's DOS commands. And HyperACCESS is so well-behaved, it can also run under Windows or TopView, concurrently with other programs.

Conduct Your Own Examination

Call or write Today!

800-826-2760 or (313) 243-0576

I have a: _____ Z100
_____ PC Compatible

☐ Send me a demo disk for $2

☐ Send me HyperACCESS for

$149

30-Day Money-Back Guarantee
Not Copy Protected

Name: _____

Company Name: _____

Shipping Address: _____

City: _____

State: _____ Zip: _____

Payment: ☐ Check ☐ VISA ☐ MC Card Exp. Date: _____

Credit Card #: _____

Add $5 shipping, $20 overseas; Michigan residents $6 sales tax.

System Requirements: IBM PC, XT, AT, or compatibles. 192K RAM minimum. PC-DOS 2.0 or greater. (Separate Z-100 version.)

Hilgraeve Inc., P.O. Box 941 Monroe, MI 48161
Communications Software Since 1981.

variable, so the change in volume can be very fast, very slow, or anywhere in between. In addition to normal musical tones, each channel can also produce white 'noise', courtesy of the noise generator. The noise generator can be controlled directly or it too can be controlled by the envelope registers. Each channel can produce noise, or music, or both at the same time! This scheme allows a clever programmer to produce very complex sounds.

The sound quality is what you would expect from a relatively inexpensive "synthesizer". The tone generators produce square waves, as opposed to sine waves, and at the tone fundamentals only (no harmonic series). This produces a harsher sound than, say, a violin or some settings of a more expensive synthesizer. However, once your ear becomes accustomed to the sound, it is not offensive at all. One aspect of the sound quality that is very good: virtually no background noise! Even with the stereo amplifier volume very high there is no discernible hiss. When music is being played an occasional low frequency 'thud' or 'pop' may be heard as a note is initiated, but that is only a minor distraction.

The board comes with a disk full of utilities, games, and many single-voice music "scores". This disk provides machine level routines, Z-BASIC routines, and example programs demonstrating the many options available. See Figure 1 for a rundown on some of the programs and files that come with the P-SST board.

The MUSic Interrupt Service Routine (MUSISR.COM) mentioned in Figure 1 is the heart of the music support software. Unfortunately, it provides for only single voice music! To get the three-voice version, you must buy Maestro!

MUSISR.COM: MUSic Interrupt Service Routine. Allows playing music in background mode, similar to a print spooler. Using this routine the computer can play music on a single voice while you do other activities.

DEMO.BAT: Plays the available "scores" (*.SCR) in background mode while a separate program draws artwork on the screen. A very impressive file that is good to show your spouse when asked why you bought the board!

SPEECH.EXE: Uses a simple text-to-speech algorithm to change typed words to speech phonemes. The results are surprisingly good.

CLOCK.COM: Resident routine that can read and set the on-board clock. I put the command CLOCK READ into AUTOEXEC.BAT so that the time and date are automatically put into the system when booting.

PSGPONG.EXE: Example program of the classic game of breakout which demonstrates many sound effects.

***.BAS:** A library of BASIC routines to aid in using the joy stick, speech, clock, tone, and noise functions in BASIC.

***.SCR:** A collection of single-voice scores. Collection includes examples of many styles including pop, classical, jazz, and country.

MUSICA.EXE and MUSICK.EXE: Plays a score as background music while drawing "string art" or "kaleidoscopes" on the screen. It's an impressive demonstration of the capabilities of the Z-100 and compiled BASIC.

Figure 1

The P-SST multifunction board comes with a disk full of support software and example programs. Here is a brief description of many of the files and programs it provides.

What Is P-SST Maestro?

The software on the Maestro disk is much the same as that which comes with the P-SST board, except it supports three-voice music. That is not to say you must use Maestro to use all three channels, but to do so requires direct programming of the board. Maestro supplies an easy interface between the computer and the board for three-voice activities. By convention, any file with a 3 as its last character is for three voices. Therefore, MUSISR3.COM is the three-voice interrupt service routine and is NOT the same as MUSISR.COM! Figure 2 gives a brief description of the files on the Maestro disk.

MUSISR3.COM: Three-voice version of the the music interrupt service routine.

DEMO.BAT: Plays the supplied scores in background mode while MUSICA3, MUSICK3, and MUSICE (supplied with the disk) take turns drawing string art, kaleidoscopes, and staves of music, respectively.

MUSICE.EXE: MUSIC Editor program. This program is described in detail in the text.

PLAY.EXE: Plays any valid score and is invoked at the MS-DOS prompt (i.e. A> PLAY BACH3.SCR).

MUSIC3.BAS: Example program using interpreted Z-BASIC.

MUSIC3C.BAS: Same as MUSIC3.BAS, except for compiled Z-BASIC.

***.BLD:** Support files for interpreted Z-BASIC.

***.OBJ:** Support files for compiled Z-BASIC.

PSSTWP.OBJ: Z-BASIC compiler port initialization routine.

***.SCR:** This disk comes with over 20 prepared "scores" to be played by MUSICA3, MUSICK3, PLAY, MUSICE, Z-BASIC programs, or another compatible program. Although not artistically transcribed, they are for the most part, accurately transcribed. Also, there is a nice variety of music, including baroque, classical, pop, jazz, and country.

Figure 2

Brief description of many of the files included with the P-SST Maestro disk.

The interrupt service routine uses the Z-100 timer interrupt to play music in the "background". It works similar to a print spooler. This means that music can be played while doing other activities with the computer. Since the timer is independent of the system clock, any system clock speed may be used without effecting the music. I have tried operation at both 5 MHz and 8 MHz with no change in the speed of the music. Of course, if bypassing the service routine, your music speed will change with the speed of the clock.

One side-effect of the interrupt service routine is the required 33K of memory which cannot be recovered without resetting the system. This had been a problem for me. Watchword, for example, would not run on my 192K system after using the music program because of the lack of space. This problem was solved when I purchased a memory upgrade to increase my RAM to 768K. (With all of the available 'resident' utilities requiring memory, I expect nearly everyone will eventually put money into extra memory). To use any of the programs from the Maestro disk, MUSISR3 must be installed. If you forget, the follow-on pro-

grams will remind you. However, I recommend reading the manual to get the whole story.

Similar to the P-SST Manual, the manual for P-SST Maestro is very good. I recommend the first 49 pages be read prior to inserting the disk into the drive. However, if you MUST get started right away, put a copy of your P-SST disk in the drive and type DEMO. This will give you about an hour of demonstration music and graphics to entertain you while you read the section about the music editor (about 20 pages). If you are really anxious, dive right into the editor. Since it is menu driven it can be successfully "explored". Also, instead of starting off with your own creation, I recommend loading one of the many provided "scores". Pick a tune that you are especially familiar with and play around with it. You'll find the exercise worthwhile. The remainder of the manual provides some insight to music history plus a good section on how to read music and transcribe manuscripts to a form the editor and the other support routines can understand.

As with the single voice package, Maestro provides routines for easy use from .EXE programs or Z-BASIC programs (either interpreted or compiled). Several examples for using these routines are provided. I successfully developed Z-BASIC programs using 3-voice music on the first try merely by emulating the examples. If you do what the manual says, its a snap! The most important file for music development is the music editor, MUSICE.COM.

The Music Editor

The editor is used to develop new scores or edit already existing scores. To use it, merely type MUSICE at the MSDOS prompt. The program first asks whether your monitor is color or monochrome. My monitor is amber, but I find the color presentations are more pleasing with the "shades-of-gray."

The next screen presents a menu showing 8 options (Figure 3). Being predominately menu driven, the software is fairly user friendly and requires minimal use of the manual to get started. However, the manual offers many excellent techniques for using the editor and should be read.

P-SST Music Graphics Editor
MAIN MENU

F1 - Specify Program Options
F2 - Display Directory & Read File
F3 - Play Music Score (Manual Mode)
F4 - Play Music Score (Automatic Mode)
F5 - List Music Score Text
F6 - Edit Music Score Text
F7 - Write Music Score to File
F8 - Exit to the Operating System
Select the desired function:

File=* NONE* Lines=1/0 Bytes=0 Voice=1 Delay=1
Sound=ON 19:20

Figure 3
Main menu display of the music editor (MUSICE.COM)

To learn the editor's features, I'll look briefly at each of the main menu options:

Specify Program Options allows changing several parameters used in other functions. Parameters include setting the starting line number for editing, enabling/disabling graphic displays, enabling/disabling speech or sound generation, setting the voice to be edited, and other similar options.

Display Directory & Read File will, as the name suggests, display the directory and retrieve a score from disk.

Play Music Score (manual mode) is, in my opinion, the best mode to use to edit an existing score. This mode first displays a staff with sixteen notes of the selected voice, then plays that voice through the speaker, and then waits for instructions. If editing is elected, only the current text for that particular voice is used. Because all three voices are not used some of the editing problems in the Edit Music Score Text function described below are reduced in scope. Figure 4 shows a picture of the staff. The staff displayed on the screen has a vertical line (to separate measures) for each line of the score. In fact, the definition of a measure is one line of the input text. Although it is intended that one line equals one measure, sometimes it is not possible to put three voices on one line when there are a lot of notes in the measure. In this case, use multiple lines for one measure. By using both the visual staff and the aural sounds, it is easy to verify the notes were entered correctly. If a mistake is made, you can edit the line, then see and listen to it again.

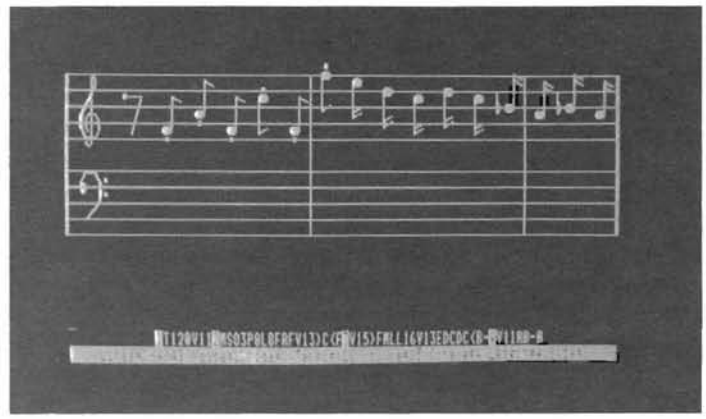


Figure 4
The staff as displayed by Maestro. Each line of the score represents one measure on the staff. Also, the staff displays only one voice at a time. There are no provisions for signature blocks at the beginning of a line, and no special manuscript techniques are supported.

Play Music Score (Automatic Mode) plays all three channels of the music from start to finish. If graphics are enabled, the notes for the selected voice are displayed on a staff. Playing the score with graphics has very little useful purpose because the rate the notes are put on the screen have nothing to do with the rate of the music. The notes on the screen go by at a constant rate set in the program options function. If the number of notes per minute is constant, the rate of change of the notes can be set to give a credible look of displaying the score in time with the music. Otherwise, you either run out of screen displays long before the music is finished, or you run out of music long before the screen displays are finished. In addition, the graphics display will interrupt the flow of the music. Every sixteen notes the screen needs to clear before displaying the next sixteen notes. This causes a little 'burp' while the interrupt service routine waits for this process to complete. If your machine is running at 8 MHz the burp is all but indiscernible, except for very fast music. Fortunately, the burp can be eliminated altogether by disabling the graphics display through the Specify Program Options command. (On the other hand, this graphics-with-music show can greatly impress your non-musician friends!) The function still has utility as it is the only function within MUSICE that plays all three voices. I use

it to verify that all three voices are synchronized and working together. It is also nice for a break to listen to the fruits of your labor.

List Music Score Text merely lists the notation of the score. A similar list can be obtained from the Edit Music Score function.

Edit Music Score Text is the primary input function of the editor. Unfortunately, this is the weakest part of the whole package! The editor is a line oriented editor and not very flexible. To edit an existing line, you first identify the line-by-line number. The prompt "old text" comes on. If you enter text it can find on the line, it then prompts "new text" to get the replacement text. If the line you are editing is long, the "old text" may require a lengthy string to uniquely identify the part to be changed, and then most of it will need to be reentered on the "new text" prompt to put everything back. For example, if the current line is "A B C A B C D" and it needed to be changed to "A B C A F C D" (changing the second "B" to an "F"), the exchange would go something like this:

```
old text: B C D
new text: F C D
```

Note that if you entered only "B" at "old text", it would find only the first B. And if you entered "B C" at "old text", it would only find the first "B C". There is no way to go directly to the second "B" to make the change without including extra characters. I think the whole operation would be better if this editing function were a screen oriented editor. In fact, the manual recommends using your own word processor to initially set up the score and then use Maestro to fine tune it. However, the product is advertised as an editor and should do a better job of that function. Although the editing, per se, is mediocre, the notation is simple and consistent.

All commands to control the sounds are easy to learn. The notes are the letters A through G (standard music notation). A suffix of '#' or '+' will raise the note a half-step (sharp), and a suffix of '-' will lower the note a half-step (flat). Unfortunately, the syntax rules will not allow proper use of notes according to music theory.

Maestro syntax says that a C-flat is not allowed. The contention is that it is the same as a B. Likewise, an E-sharp is not allowed because it is the same as an F. While on the piano keyboard this may be true, it is not true for the theorist. Any musician knows that a C-sharp major triad requires an E-sharp (not an F), and that an A-sharp major triad requires a C-double sharp! Important? Maybe not for this software. However, many of the scores entered will be taken from a printed manuscript, and that WILL follow the rules of music theory, not the logic of the layman. This means you may have to transcribe a note from a printed score to something the editor can understand.

Why do it that way? Once again, IBM compatibility raises its ugly head! The IBM-PC BASIC's PLAY command uses this same notation scheme. Nonetheless, Software Wizardry could have done better by allowing double (or even triple) sharps and flats. Allowing the more proper notation would not lose the IBM compatibility and still provide a more comprehensive notation for P-SST users. However, it appears that this "foresight" may pay dividends to Software Wizardry . . . The Gemini board (to make the Z-100 compatible with the IBM PC) will soon support the music functions of the IBM through the P-SST board! But, getting back to notation . . .

Since the board is capable of producing sounds in an 8 octave range, a method of changing octaves is required. This can be done two ways: explicitly with the "O" command, or relatively with a "<" or ">". For example, "O3" means play in the third octave. The "<" means go down one octave from the current octave, and the ">" means go up one octave. An octave is defined as the range of notes from a "C" up to a "B". To go from a "B" UP to the next "C" would be written B>C.

Although merely entering notes is the first step to making music, other parameters must be used to make it sound 'musical'.

Control of loudness, texture, and speed are very important and, with Maestro, easy to use and remember. Volume is controlled by using a V followed by a number from 0 to 15, where 15 is the loudest. Texture, or how slurred/staccato the notes are from one to the next, can be controlled by either of two methods: An S followed by a number 0 to 64 controls the slur/staccato, or the IBM method of an M followed by an S (staccato), an N (normal, or "detached"), or an L (legato).

Speed is changed by one of three methods. First, tempo is changed by using a T followed by the number of beats per minute. A quarter note takes one beat. T120 means play 120 quarter notes (or quarter note equivalents) per minute. The L command controls the number of notes per beat. L8 means play eighth notes, or two notes per beat. L16 means play sixteenth notes, etc. You are not required to use these multiples, however. By changing either the tempo (T) or the length (L) you can simulate triplets, quintuplets, or just about any combination you want!

To summarize, the command "V15 MN T120 L8 O3 A" means play very loud, with normal space between notes at 120 beats per minute using eighth notes, and play an A in octave 3.

To add a second voice to a measure (line), merely add a comma followed by the desired notes. For example, the command

```
L8 C D E F, L16 G F E D C D E F
```

tells voice one to play the series C D E F in eighth notes at the same time voice two is playing the series G F E D C D E F in sixteenth notes.

You can see it is not too difficult to enter notes and their lengths to create a "score". The first piece I transcribed into the editor was BACH's two-part invention in F. My first attempts sounded very mechanical. The key to making music enjoyable is to constantly change volume and texture. That takes a lot of work! I edited the score three times before producing a score I would call musical. Figure 5b shows an excerpt of that score. Notice that each of the two voices often have different volume levels. This is called "voicing". It allows the more important material to be more easily heard above the rest of the music. Compare it with the "plain vanilla" original score in Figure 5a. Notice the lack of directive commands in that version. There are virtually no changes in volume or note detachment. This is shown by no V, M, or S commands being present. If this had been a more romantic piece, changes in tempo would also have been required to make the music sound as romantic music should. My wife, a professional musician, was more impressed with what came of the extra effort than with the plain vanilla transfer of notes from manuscript to computer.

This brings us to the last two options from the main menu: **Write Music Score to File,** and **Exit to Operating System.** They are self-explanatory, but there is one very big danger . . . What if you are

planning to write to disk (F7) and your eye-hand coordination isn't all it should be and you hit F8 (exit to operating system)? Too bad! You've lost all your work! I did it once, and learned to be extremely careful. Also, saving your work frequently (a good habit anyway) will keep the amount of extra work to a minimum if you do make the mistake. Hopefully, future versions will question the action or require a control-key function before sending you to the operating system.

```
T120 V15 MS     T120 V15 MS
03 P8 L8 FAF >C<F,    P4 P4 P4
>F L16 ED CDC<B-,    P8 02 L8 FAF
AB-AG,     >C<F
L8 FA>C<A,     >F L16 EDCDC<B-
>FC,    AB-AG
L16 A14 >C<B->C <A>C<B->C,    L8 FA>C<A
<A>C<B->C,  >FC
<F14 AGA FAGA,    L16 A14 >C<B->C <A>C<B->C
FAGA     <A>C<B->C
D FEF DFEF DFEF,     <F14  AGA FAGA FAGA
L8 <BG >D<B >FD,     D FEF DFEF DFEF
L16 G AGF EFED     L8 <B G >C<G
CDC<B-     >EC
A8 >DC<B>C<BA GAGF,    L16 F GFEDEDC <B>C<BA
EFED C8 >C<B,    G8 >C<B ABAG
L8 >C<E,    FGFE
F>C <E>C <DB,     DEDC GFEF L8 G<G
>C4 P2     02 P8 L8 CECGC
```

Figure 5a

Author's first version of J. S. Bach's two-part invention in F. Note the lack of changes in volume and texture. Although easier to enter into the editor, the resulting music is somewhat bland.

In spite of the quality of music this software can support, I still have several disappointments:

The envelope generator is not supported! The envelope generator could add another dimension to the music by using varying sound decay rates. I have been successful simulating a plucked guitar string, or piano string by programming two different decay rates on two separate voices. One voice uses a fast decay rate, while the other uses a slow decay rate, while both are playing the same frequency. This produces a very convincing "plucked" sound. Unfortunately, the music interrupt service routine doesn't support this kind of complexity.

Also, the noise generator is not supported! Using the noise generator would provide a convincing percussion section in one or more of the channels. Using the envelope generator with tones and noise together (which the board CAN accomplish) could provide a very good foundation for pop, rock, jazz, and other forms of contemporary music.

For students of music, it is too bad that the screen display is capable of showing only one voice at a time, and even that is extremely limited. There is no provision for key signatures or a timing block. Also, notes above octave six will show ledger lines, but the display runs out of screen before it gets to the note. Another detractor involves successive notes. The usual method of writing successive eighth notes is to place a bar across them, but the graphic display shows each note individually. Although Maestro is good for composing, it is not as good as a piano, and is not capable of producing the manuscript for performing, or for your theory assignments. Oh well . . .

Do these disappointments mean don't buy? No! If you have not purchased the P-SST board for your Z-100, consider it. Few accessories for a computer can do so much for the investment. If you have purchased the P-SST board and have any inclination to do something dynamic with it, I highly recommend P-SST Maestro. Most pieces of software have room for improvement, but for $59.95, Maestro can offer hours of entertainment and enjoyment. It is especially good when setting up games, since the interrupt service routine will play the music in the 'background' and hardly effect the game. Mood music could be added to adventure games that would reflect the danger, or better yet, something to come. Children's games could be made more entertaining by using children's songs. The idea list is endless.

```
T120 V11,   T120 V11
MS 03 P8 L8 FAF V13 >C<F,     P4 P4 P4
V15 >F ML L16 V13 ED CDC<B-,    P8 V11 02 MS L8 FAF
V11 AB-AG,     V13 >C<F
L8 MS FA>C<A,     V15 >F ML L16 EDCDC<B-
V13 MS >FC,     ML V11 AB-AG
V14 S6 L16 A14 V13 ML >C<B->C <A>C<B->C,     L8 V13 MS FA>C<A
<A>C<B->C,    >FC
V14 S6 <F14 V11 ML AGA FAGA,
      V14 S4 L16 A14 ML V11 >C<B->C <A>C<B->C
FAGA,     <A>C<B->C
V14 MS D ML V11 FEF DFEF DFEF,    V13 S6 <F14 ML V11 AGA FAGA FAGA
V11 L8 MS <BG >D<B >FD,     V13 MS D ML V11 FEF DFEF V11 DFEF
V13 L16 ML G V11 AGF EFED,     MS V10 L8 <B V11 G >C<G
V10 CDC<B-,    V11 >EC
MN A8 ML >DC<B>C<BA V11 GAGF,   V13 L16 ML F V11 GFEDEDC <B>C<BA
V12 EFED MN C8 V13 ML >C<B,    MN G8 V12 ML >C<B V13 ABAG
MN V14 L8 >C<E,   V14 FGFE
V15 F>C <E>C <DB,    V15 ML DEDC GFEF MN L8 G<G
ML >C4 P2,    V13 MN 02 P8 L8 CECGC
```

Figure 5b

Author's final version of J. S. Bach's two-part invention in F. Of particular note is the changing of texture and volume to add interest to the music. Each voice treats these attributes differently, further adding to the style.

The support Software Wizardry gives to the Heath/Zenith community is commendable, and the P-SST board along with Maestro gives the Z-100 a capability that no other vendor has tried to give it — talking, time functions, joy sticks, and especially, three-voice music. I look forward to further support from Software Wizardry, and hope that other software developers will offer features that use the P-SST board.

Max Memory Mods

Review Of Two Memory Modifications For H/Z-100 PC Series Computers

Pat Swayne
HUG Software Engineer

With software getting more memory hungry all the time, and memory resident utilities "popping up" everywhere, the ability to put the most memory you can in your computer and use all of it is becoming more important than ever. In this article, I will describe two modifications that let you accomplish that. These modifications are the amazing GNS PALs and software from GraphNet Systems, Inc. for H/Z-150 and H/Z-160 computers, and the ZPAL-158 by KEA Systems, Ltd. for H/Z-158 computers.

The GNS PALs

The H/Z-150 and H/Z-160 series computers contain a memory board that has room for 5 banks of memory ICs. They were originally intended to be populated with 64k chips, so that the total memory on the board was 320k. Various independent manufacturers have provided replacement memory decode PALs (Programmable Array Logic) that allow 256k chips to be used in at least some of the banks, so that the board could hold 640k or even 704k of memory. Now GraphNet Systems has developed two new PALs that allow you to put 256k chips in all 5 banks, for a total of 1.2 megabytes of memory on the board. One of them, model GNS64MB, lets you use 640k of the memory as program memory, and 578k as "hidden" memory, while the other one, model GNS70MB, lets you use 704k as program memory and 512k as hidden memory. (Note: Some peripherals may not work if you have 704k of program memory.) Only 64k of the total memory in 5 256k banks is not used at all by the GNS system. The only thing that you can currently do with the hidden memory is to use it for a RAM disk (memory disk). However, I have been told that GraphNet Systems is working on a way to allow the use of the extra memory as Lotus/Intel expanded memory. Lotus/Intel expanded memory is a specification for using memory beyond 640k for certain spreadsheets or other programs.

The GNS PAL "kit" consists of a single memory decode PAL that is used to replace the one on your memory board, a disk containing RAM disk software, and a single sheet of documentation. When you install the GNS PAL, you must solder a jumper wire on the back of the board. This jumper apparently taps a port that selects the hidden memory (bank switching). After the GNS PAL and the jumper wire have been installed, along with the 256k chips, your computer will work as before (except that it will report 640k or 704k of RAM). Neither the MFM-150 ROM nor the CHKDSK program will "see" the hidden memory.

The RAM disk software provided with the GNS PAL works quite well, and you can actually make the memory disk larger than the 512k or 578k of hidden memory. You can use up to 767k for your RAM disk (why not 768k?), which means that a combination of hidden memory and program memory is used. The RAM disk software can be made to cause a beep every time the memory disk is accessed, but that is annoying, and it slows access of the memory disk down considerably. The beep "feature" is actually the default mode of operation, and you must include the word MUTE in the CONFIG.SYS command line that activates the RAM disk program, if you do not want beeping.

You can set up the RAM disk software to re-initialize the memory disk when you boot up (after resetting), to leave the data intact, or to prompt you to select re-initialization or not. You can also configure it to provide a 60 character keyboard buffer, if you find the normal 15 character buffer inadequate.

If you cannot afford to buy 5 banks of 256k memory chips at one time, you can install the GNS PAL and 3 banks, to give you 640k or 704k of program memory. Later, you can get the other two banks to allow use of the RAM disk software.

The GNS PALs represent the ultimate memory upgrade for the H/Z-150 and H/Z-160 series computers that does not require an additional board. The only bad thing about them is that you are locked into having 640k or 704k of program memory, but you can probably trade in your PAL for the other option if you change your mind after buying your kit.

The GNS PALs were developed by GraphNet Systems, Inc., P.O. Box 337, Reading, PA 19603. They are available directly from GraphNet, and are also available (at a slight discount, I believe)

from Graymatter Application Software, 1501 Township Line Road, Norristown, PA 19401, (215) 279-4460. Graymatter sells the PAL kits with or without RAM chips at the following prices:

PAL alone	$ 49
PAL and 27 chips (3 banks)	$129
PAL and 45 chips (5 banks)	$179

All kits come with RAM disk software. Be sure to specify GNS64MB or GNS70MB when you order.

The ZPAL-158

On the CPU/memory card of an H/Z-158 series computer, there is room for 3 banks of memory ICs. If all three banks are populated with 256k chips, you have a total of 768k of memory on the board, but the computer can use only 640k of it. The ZPAL-158 from KEA Systems allows you to use all 768k of your memory. Some or all of the extra 128k of memory can be used to expand your program memory to 704k, to add a memory disk, or both.

The ZPAL-158 "kit" consists of a single memory decode PAL chip that is used to replace the one on your CPU/memory card, and 16 pages of instructions. The instructions present two methods for installing ZPAL-158, called the "Physical Jumper Alternative" and the "Programmable Jumper Alternative". The simplest method is the Physical Jumper method, which consists of removing your old memory decode PAL, plugging in the new one, and setting some jumpers to configure the way you want to use your extra memory. You can configure ZPAL-158 in 4 ways, as explained in the following table:

Configuration	Memory Usage
1	704k of program memory and 64k unused.
2	704k of program memory and a 64k block at D0000-DFFFF.
3	640k of program memory and a 128k block at C0000-DFFFF.
4	640k of program memory and a 128k block at D0000-EFFFF.

The Programmable Jumper Alternative provides the same 4 configurations, but you do not have to change jumpers to change from one to another. Instead, you write to a port using the built-in MFM-150 debugging program to select a configuration. To install ZPAL-158 using the Programmable Jumper method, you must make some wiring changes (add jumper wires, etc.) to your CPU/memory card. I did not want to do the extra wiring, so I selected the Physical Jumper method, and used configuration 4.

After you install ZPAL-158, a little extra work is required to get your computer working when you turn it on, and in the case of configurations 1 and 2, each time you reset it. The reason for this is that the ROM monitor does not recognize the extra memory. If you select configurations 2, 3, or 4, the computer will not auto-boot when you turn it on, but instead display a memory parity error message. If you use configuration 1, you must press Ctrl-Alt-Ins to prevent your computer from auto-booting when you first turn it on. With all four configurations, you must initialize the extra memory somehow before you can use it. This can be done by entering some commands given in the ZPAL-158 instructions with MFM-150, but with configurations 3 and 4, it can also be done entirely in software, so that all you have to do after the parity error message appears is to boot your disk or winchester manually. I never liked auto-boot on power up anyhow, so I don't mind the way ZPAL-158 changes the system. With configurations 3 or 4, the computer will auto-boot if you press Ctrl-Alt-Del after you have booted up initially, because the initialization of the extra memory in those configurations only has to be done at power up.

To make good use of the extra memory enabled by ZPAL-158, especially in configurations 3 and 4, you need a memory disk program that can access the new memory at its non-contiguous location. KEA does not currently supply memory disk software, but they may in the future. The VDISK program supplied with MS-DOS version 3 will not work, but the MDISK program supplied with MS-DOS version 2 and also with the Programmer's Utility Pack works fine. You can use it as is, if you initialize the memory manually using MFM-150 when you turn on your computer and include a line in your CONFIG.SYS file like this

```
DEVICE=MDISK.DVD SIZE=128 START=D000
```

The above example is for memory configuration 4. If you have the Programmer's Utility Pack, you can modify the source code of MDISK and reassemble it so that it will initialize the extra memory for you, and also so that you can reset and reboot without destroying data in the memory disk. These modifications are presented below. The modification to allow resetting without destroying data was devised by John Stetson, and first printed in CHUG, the newsletter of the Capital Heath Users' Group in Fairfax, Virginia. The modifications are presented by permission from CHUG. They have been modified to work in the ZPAL memory environment.

The MDISK source file, MDISK.ASM, is on the disk marked "UTILITIES" if you have the MS-DOS Version 2 Pack, or on the disk marked "Z-100 BIOS SOURCES DISK #2" if you have the MS-DOS Version 3 Pack. To modify MDISK.ASM, load it into an editor and locate this line:

```
;                SIZE=num-Kb and START=start-paragraph
```

Add some comments after the line as follows:

```
;                SIZE=num-Kb and START=start-paragraph
;       Modifications for data protection on reset by
;             John Stetson, as presented in CHUG, Aug  1985
;       ZPAL-158 memory initialization mod by P  Swayne, HUG
```

Next, locate the following line:

```
        LDS     DX,ES:DWORD PTR CIN_KADDR[BX]
```

Add a label EXITOK: to the line as follows:

```
EXITOK: LDS     DX,ES:DWORD PTR CIN_KADDR[BX]
```

Next, locate the following line:

```
DRIVE           DB      ?, ":", CC_CR, CC_LF, "$"
```

Modify the line, and add lines following it, to look like this:

```
DRIVE           DB      ?, ":"
CRLF            DB      CC_CR, CC_LF, "$"

;       Message for user reformat prompt
PROMPT_MSG      DB      CC_CR, CC_LF
        DB      "Re-initialize the Memory Disk (Y/N)? $"
```

The next modification has to be done only if you got MDISK .ASM from the MS-DOS Version 2 Programmer's Utility Pack. If you got it from the MS-DOS Version 3 Pack, the modification has already been done. Locate the following line:

```
MD_SIZE END     =       OFFSET $
```

Change it to look like this:

```
MD_SIZE_END     =       OFFSET $ - OFFSET MD_SIZE
```

The remaining modification is for all versions of MDISK.ASM. For this modification, locate the following two lines:

```
        INT     21H
        JMP     MD_INIT4        ; Finish init. overwrite tra
```

Insert lines between them, so that you have this:

```
        INT     21H

;       Allow the user to skip disk initialization

        MOV     DX,OFFSET PROMPT_MSG
        MOV     AH,DOSF_OUTSTR
        INT     21H             ; PROMPT FOR RE-FORMAT
        MOV     AH,DOSF_CONIN
        INT     21H             ; GET REPLY
        PUSH    AX              ; SAVE IT
        MOV     DX,OFFSET CRLF
        MOV     AH,DOSF_OUTSTR
        INT     21H             ; PRINT CRLF
        POP     AX              ; GET REPLY
        AND     AL,5FH          ; CAPITALIZE
        CMP     AL,'Y'          ; YES?
        JZ      REINIT          ; IF SO, RE-INITIALIZE
        JMP     EXITOK          ; ELSE, SKIP
REINIT:

;       Clear ZPAL'S extra memory (configuration 4) [PWS]

        PUSH    ES
        CLD
        MOV     AX,0D000H       ; FIRST EXTRA BANK
        MOV     ES,AX
        MOV     CX,8000H
        XOR     AX,AX
        REP     STOSW           ; CLEAR FIRST EXTRA BANK
        MOV     AX,0E000H       ; SECOND EXTRA BANK
        MOV     ES,AX
        MOV     CX,8000H
        XOR     AX,AX
        REP     STOSW           ; CLEAR SECOND EXTRA BANK
        POP     ES

        JMP     MD_INIT4        ; Finish init. overwrite tra
```

After you have made the modifications to MDISK.ASM, copy it and the following files to the same disk: PARMS.ASM, MACLIB .ASM, DEFASCII.ASM, DEFMS.ASM, and DEFDEV.ASM. These files are available on the disk marked "Z-100 BIOS SOURCES" if you have the MS-DOS 2 Pack, or the disk marked "Z-100 BIOS SOURCES DISK #2" if you have the MS-DOS 3 Pack. Boot up on a system disk containing MASM, LINK, and EXE2BIN, and enter these commands to assemble your new MDISK:

```
MASM B:MDISK,B:MDISK,
LINK B:MDISK,B:MDISK;  (ignore error message)
ERASE B:MDISK.OBJ
EXE2BIN B:MDISK B:MDISK.DVD
```

The above example assumes that MDISK.ASM and the supporting files are on drive B:. An error message may be generated by LINK, but ignore it. When the assembly is finished, copy the new MDISK.DVD to your system disk, set up your CONFIG.SYS, and reset and reboot. Your memory disk should now be operational. You should be able to boot right after turning your computer on, without clearing the extra ZPAL memory first. Each time you boot, you will be prompted "Re-Initialize the Memory Disk (Y/N)?". You must answer Y the first time you boot after turning on the power, but every time you boot after a reset, you may type N or just hit RETURN, and the data in your memory disk will be left intact.

When the H/Z-158 series was first introduced, some people were disapointed that it was not as versatile in its memory expansion capabilities as the H/Z-150 series (using the original memory boards). I guess that is still true, but at least with the ZPAL-158 you can use all of the memory that you can put on the board. To make the ZPAL-158 more "user friendly", perhaps KEA should look into modifying the monitor ROM so that it recognizes and clears the extra memory.

The ZPAL-158 kit is available for $36 (US) from KEA Systems, Ltd., #412-2150 West Broadway, Vancouver, B.C., Canada V6K 4L9, (604) 732-7411. There is also a ZPAL-148 for H/Z-148 computers. Quantity discounts are available.

*

SPECIAL OFFER

GEMINI EMULATOR BOARDS

$349.*

40% BELOW REGULAR RETAIL

The Gemini Emulator Board delivers IBM PC compatibility to your H/Z100 computer.

As a result of a volume purchase direct from the manufacturer, Liberty International has acquired a limited number of Revision #1 Gemini Emulator Boards that have been **factory upgraded** from 5MHz to 8MHz. All of these Boards have the **latest version of software** installed and come with the same **manufacturer's warranty** and technical support as new Gemini Emulator Boards.

This is a manufacturer-authorized clearance. Just phone us or fill in and mail the coupon below and we'll ship you a Gemini H/Z100 Emulator Board at more than **40% below regular retail price.**

* Only while quantities last; available only through Liberty International; no charge for using credit cards; shipping included; Liberty International reserves the right to withdraw or revise this offer at any time without notice.

PLEASE SEND ME:

_____ Gemini Emulator Boards at $349. each.

☐ Payment Enclosed Total $ _____ **

☐ Please charge my Credit Card

☐ Visa # _____ exp__ /__

☐ MC # _____ exp__ /__

SHIP TO: R

Name: _____

Phone: ( ) _____

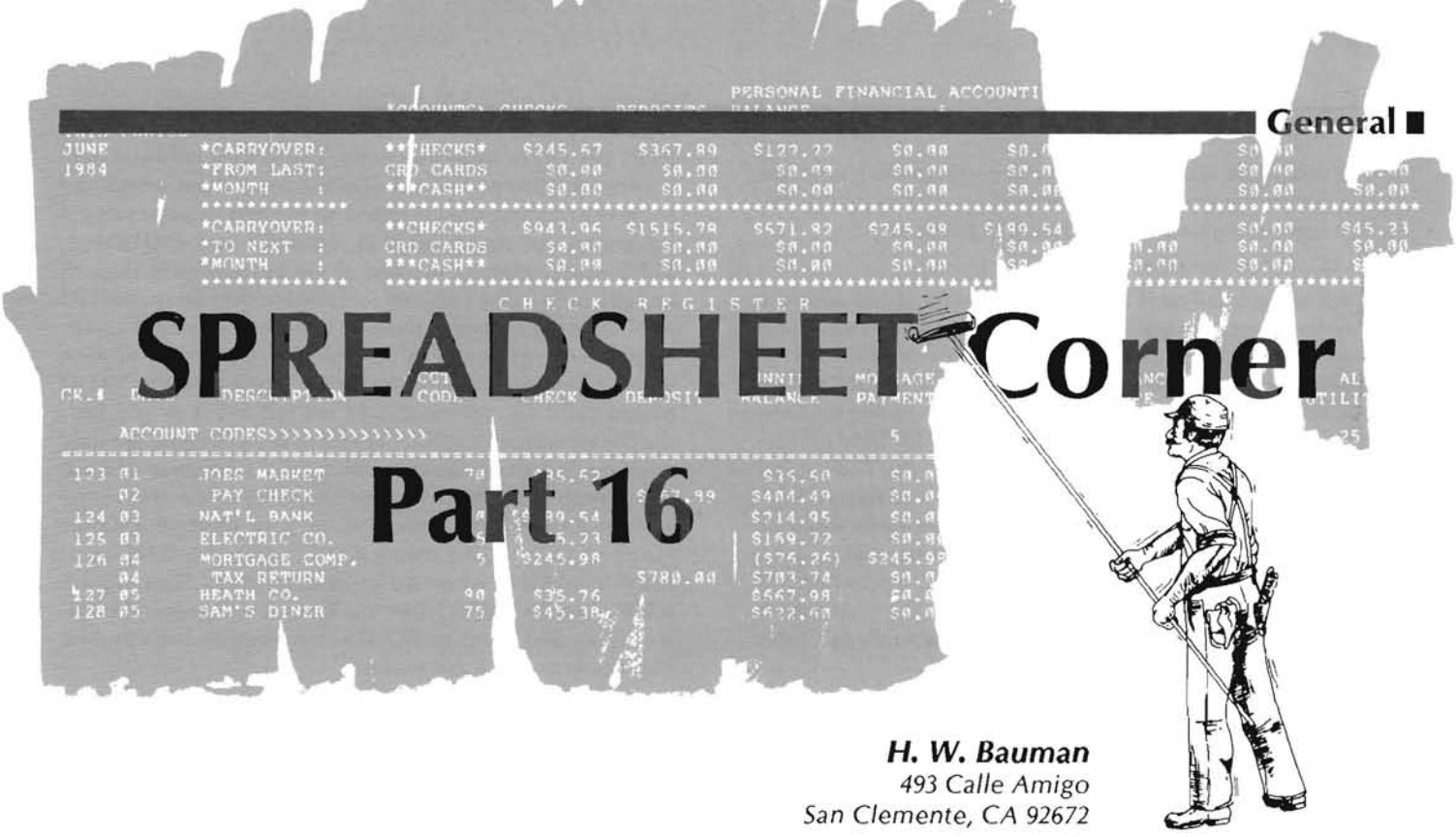
Address: _____

_____

State: _____ Zip _____

** Virginia residents add 4% Sales Tax; mail coupon to address below or call (804) 482-5768.

LIBERTY INTERNATIONAL
101 SCARLETT DRIVE
CHESAPEAKE, VA 23320
(804) 482-5768



SPREADSHEET Corner Part 16

H. W. Bauman
493 Calle Amigo
San Clemente, CA 92672

With the introduction of "SPREADSHEET Corner — Part 15", we designed a short, simple database application. We added a macro, ALT-A, to Sort our data with a Primary-Key, YTD, and a Secondary-Key, NAME. We also provided a macro, ALT-B, to Sort the second-quarter sales in descending order. By using Range Names in place of cell labels, the macro we created would be immune to column changes. Finally, we were left with the problem of creating a macro that would be flexible enough and intelligent enough to distinguish between the four quarters. Our goal for this project is to end up with a menu operated database! Use the worksheet from Part 15 (See Figure 1) for reference.

Before I carry on with our project, I should discuss the LOTUS 1-2-3 "/X commands". These "X" commands are one of the principle items that put the LOTUS 1-2-3 AHEAD of most of the other spreadsheet software packages! Once you readers have learned how to use the "X" commands, I am sure you will find that working with 1-2-3 will increase your worksheet's power. It will also make your work look professional and add to your enjoyment. They are MORE than a typing alternative; they make LOTUS 1-2-3 programming possible and easy to do. The other computer languages are not as simple to learn. As always with "SPREADSHEET Corner", we will learn by doing! As we proceed, I will define the "X" commands for you and we will start using them!

1. /XI command directs 1-2-3 to perform a task only if a specified condition has been met. The format for the command is:

   ```
   /XI condition formula~.  keystrokes  .
   ```

   The /XI command means that "IF the result of the condition formula is True, the keystrokes that follow the tilde (~) in this cell will be used. Otherwise, skip to the next cell in this macro and continue the execution of the macro. Did you readers notice that the I stands for IF and the Tilde "~" represents THEN. If any readers have programmed with BASIC or some other computer language, their IF-THEN-ELSE will look familiar to this /XI command. Whatever comes between the I of the /XI command and the following Tilde (~) is the condition formula. If you will check back and look at the ALT-B macro, we could revise it as follows:

```
18-Nov-85          (SALES IN THOUSANDS OF DOLLARS)

                   SPREADSHEET Corner COMPANY
                   *OFFICE SALES RANKING PLAN*

                                                    YTD
OFFICE       NAME      1STQTR 2NDQTR 3RDQTR 4THQTR YTD RANK
-------------------------------------------------------------
ATLANTA      ABBOTT,T   178    188                  366   1
BOSTON       WILSON,E.  188    173                  361   2
LOS ANGELES  RICE,L     195    161                  356   3
PORTLAND     NEWMAN,D   169    178                  347   4
OMAHA        JOHNSON,E. 178    163                  341   5
CHICAGO      MILNER,H   162    171                  333   6
```

\A	\B
/DSPYTD~D~	/DSP2NDQTR~D~
SNAME~A~G	SNAME~A~G
/DF~1~~~	

Figure 1

```
/XIQTR=2~/DSP2NDQTR~
```

The condition is QTR=2 and if this condition is True, then the macro will execute the specified keystrokes in this cell. The manual command would look as follows:

```
/Data Sort Primary-Key 2NDQTR Return
```

Thus, we are saying if the current quarter is the second-quarter, the macro will set the Primary-Key to 2NDQTR and then continue with the rest of the macro. If QTR is NOT equal to 2, the macro program will skip the /DS2NDQTR~ and skip to the next macro label without executing the keystrokes that were jumped over. Note that the Tilde (~) following the condition stands for THEN! But, the Tilde at the end of the Data command still represents the Return keystroke; so, whatever follows the macro's first Tilde, whether it is a single command or several commands, will only be performed if QTR=2 is True. If not, the macro program will proceed with the next keystrokes given in the next macro cell. (This will be further clarified when we actually use these macros in our Project.)

2. /XG command is a very useful command. The G stands for Goto and it instructs 1-2-3 to go directly to a different cell in the macro to get the next macro command. A simple example would be:

```
/XGF1~
```

This command tells 1-2-3 to stop the current procedure, go to cell F1 (a macro cell), and execute the keystrokes found there. The designation of a /XG command can be an active cell or a cell name. The Tilde that terminates the command is Essential! If the Tilde were not there, the macro would NOT work. This /XG command should remind you of the BASIC language GOTO command.

3. Together, the /XI and /XG commands will permit the User to translate many keystrokes into a language of macros and the X commands. Let's use QTRN to represent the four quarters, where N could stand for the 1ST, 2ND, 3RD, or 4TH quarter. Assume that QTRN already contains the value of the current quarter. We will begin entering the revised \B macro with the following:

   A. Name the QTRN cell by moving the cursor to cell J1 and type /RNCQTRN Return Return. Cell J1 now has the name QTRN.
   B. Move cursor to \B macro (cell L6).

   The macro's first task will be to evaluate the contents of QTRN to find what quarter should be Sorted. If it is the second-quarter sales, we want it to issue the correct Sort command. The macro label would be this:

```
/XI(QTRN=2)~/DSP2NDQTR~D~SNAME~A~G
```

I enclosed the condition in parentheses so it would be easier to see! The keystrokes to the right of the first Tilde would be executed for the second-quarter, which translates to:

```
/Data Sort Primary-Key 2NDQTR Return Descending Return
Secondary-Key NAME Return Ascending Return Go
```

I will keep our labels short; so, all keystrokes that are part of the /XI condition will be in the same label cell as the /XI command. Note, to make the example easier to follow I am going to start with the current quarter as the first-quarter. We will be replacing the contents of the L6 cell with a new entry:

   C. Type '/XI(QTRN=1)~/DSP1STQTR~D~SNAME~A~G

   DO NOT press Return yet! If you did press Return, press F2 (Edit) key to continue the entry. Up to now we have only entered the first part of our planned procedure, which had the task of Sorting the file based on the first-quarter sales. If we ended our macro label at this point and if QTRN=1, 1-2-3 would execute the rest of the keystrokes in the cell and proceed to the next cell included in the macro. We DO NOT want to proceed to the next label, but we want 1-2-3 to stop executing the macro. We could stop macro execution with a blank cell because only non-blank cells are VALID macro cells. So, where should we tell 1-2-3 to go? Let's decide we will use a cell at the end of the macro we are now creating, but we do not know where that is because we have not completed the macro. We DO NOT know where the end location will be. Let's plan for it by giving it a Range Name — CONTX. We create CONTX with the /RNC command a little later when we know its location. Now, we will tell the macro that QTRN=1 is True. It will assign the Primary-Key and jump to CONTX. We need to see this. The /XG command will do this task:

   D. Type /XGCONTX~ and press Return.

   The first \B macro cell L6 should have the following revised label:

```
/XI(QTRN=1)~/DSP1STQTR~D~SNAME~A~G/XGCONTX~
```

Can you follow what we have here? The next cell will represent the following step of the procedure. Try to see if you can see what it will be! It is almost the same as the step we just completed. QTRN=1 will be replaced with QTRN=2 and 1STQTR will be changed to 2NDQTR. To save typing and thus reduce the chance for a typing error, I like to use the Copy command:

   E. Put the cursor on cell L6, type /C (COPY), and press Return.
   F. Press Down Arrow key once to cell L7 and press Return.
   G. With the cursor on L7, press F2 (Edit) key.
   H. Change QTRN=1 to QTRN=2 and 1STQTR to 2NDQTR and press Return.
   I. Repeat the above procedure, but use QTRN=3 for QTRN=2 and 3RDQTR for 2NDQTR in cell L8 and press Return.
   J. Again, repeat the procedure using QTRN=4 for QTRN =3 and 4THQTR for 3RDQTR in cell L9 and press Return.
   K. Cell L10 is the Blank cell that ends this macro. This is where CONTX will be located by typing '/RNCCONTX and pressing Return Twice.

The new version of the \B macro should look like the following:

```
      K
5   /XI(QTRN=1)~/DSP1STQTR~D~SNAME~A~G/XGCONTX~
6   /XI(QTRN=2)~/DSP2NDQTR~D~SNAME~A~G/XGCONTX~
7   /XI(QTRN=3)~/DSP3RDQTR~D~SNAME~A~G/XGCONTX~
8   /XI(QTRN=4)~/DSP4THQTR~D~SNAME~A~G/XGCONTX~
```

To test the ALT-B macro, load QTRN with a value for the second-quarter. Be SURE to SAVE the worksheet before testing the new macro (No ZAP!).

   L. GoTo QTRN (cell J1), type in a 2. Hold the ALT key down and press the B key.

What happened? The results should be the same as the results of our previous manual second-quarter Sort. Did yours? If not, use

the error tracking methods we discussed earlier. Do you remember how? A macro (ALT-B) can easily have logical or typo errors. If you heard a "beep" or if your results are in error, then you most likely have a typo error. When the User executes this macro, it operates so rapidly that you cannot watch it. If a visual check cannot find the error, use the Single-Step Mode of macro execution. It is important to know this method, because we will get into greater complexity with some macros where this method is a must! Here is the method for your review:

1. Hold down ALT key and press F1 (Help) key. You will see the word "STEP" appear in the lower-right bottom panel. A step is a single keystroke or a single /X command. You can press any key (I use the spacebar) and let it up, to perform a step.
2. Hold down ALT key and press the B key. Note that an SST (single-step) indicator will appear in the upper-right control panel which is your signal to press any key! Follow your macro as you step through it for errors. Because we set QTRN to 2, the condition will be True. The macro should end after it completes the second-quarter Sort. You should NOT see any keystrokes for the third or fourth quarters.
3. To disable the Single-Step Mode, hold down the ALT key and press the F1 key. This is a "toggle" function. The "STEP" indicator will disappear from the lower control panel.

Did you find your error? (Also, the Single-Step Mode will not work with the LOTUS 1-2-3 Version 1.0.) I would expect that all the readers are using Version 1A or Version 2.0 (I have not received my Version 2.0 yet.)

We have not completed the improvements for our project by a long shot. Compare your worksheet with Figure 2. Next, we will automate the quarterly Sort even more! As we have it now, the User must give QTRN a value of 1, 2, 3, or 4. Without this value, the macro will not work and the User must give QTRN a value of 1, 2, 3, or 4. Without this value, the macro will not work. Here we can use another /X command, called /XN. This command prompts the user for a number, retrieves the number, and stores the response in a specified cell. The format for the command is (no letter characters will be accepted):

```
/XNmessage~location~
```

The "message" is the prompt the User will find displayed in the upper control panel, and "location" is the cell address or Range Name where the User wants the \B macro to begin with the following label:

```
/XNEnter Desired Quarter(1,2,3,or4):~QTRN~
```

The prompt should be clear to the User. Of course, you might think of a better one. Here is the procedure we will use:

1. Press F5 (GoTo) key, type \B, and press Return to move the cursor to the cell Named \B by pressing the Up Arrow key.
2. Type /Move (Move command), press the END key, press the Down Arrow key twice to highlight the entire macro including the Blank cell Named CONTX, and press Return. The cursor will return to its original cell and 1-2-3 will request the destination.
3. Move the cursor with the Down Arrow key to cell L7 and press Return.

The macro has been moved down and the cursor returns to the blank cell above the macro.

4. Type '/XNEnter Desired Quarter(1,2,3,or4):~QTRN~ and press Return.

Let's check the location of \B with the F5 (GoTo) key. Press the F5 key, type \B, and press Return. You should find \B is still attached to its old location that is now the second cell of the macro; so, we must Range Name it again.

5. Type /RNC\B and press Return. 1-2-3 will suggest cell L7, which is the current location of \B macro. Move the cursor to cell L6 and press Return.

This should make the \B macro functional again. Test it by holding down the ALT key and pressing the B key. The upper control panel should prompt you with the message "Enter Desired Quarter(1,2,3,or4):":

6. Type in a "1" to Sort the first-quarter sales and press Return. Check the results. Do they look correct? If not, you now should know how to find errors! (Try 2 also)

So far, we have created macros to Sort and Rank by YTD Sales and to Sort the four quarterly sales. Our worksheet has a lot of other valuable data. So, we will create a couple of more macros to Sort Data by name, Sort by office, and one of the most important, a Print macro which might seem the hardest.

First, we will design the Sort by name macro — ALT-C. When our Database increases by adding more names, it will become necessary to Sort the NAME field in alphabetical order. This will be easy for you now with the alphabetical order. This will be easy for us to do now with the experience we have attained from this project. All we need to do is point the Primary-Key to the NAME field column. The manual commands to do this task would look like this:

```
Data Sort Primary-Key NAME Return Ascending Return Go
```

When we translate this set of commands into the macro which we will call \C, the procedure would be as follows:

1. Move the cursor to cell M6, type /RNC\C, and press Return Twice.
2. Type the label '/DSPNAME~A~G and press Return.

Again, test the macro by holding down the ALT key and press the C key. Did your macro work? Look at the NAME column! Let's do

18-Nov-85 (SALES IN THOUSANDS OF DOLLARS) | 2

SPREADSHEET Corner COMPANY
OFFICE SALES RANKING PLAN

OFFICE	NAME	1STQTR	2NDQTR	3RDQTR	4THQTR	YTD	YTD RANK
ATLANTA	ABBOTT,T	178	188			366	1
BOSTON	WILSON,E	188	173			361	2
LOS ANGELES	RICE,L	195	161			356	3
PORTLAND	NEWMAN,D.	169	178			347	4
OMAHA	JOHNSON,E.	178	163			341	5
CHICAGO	MILNER,H	162	171			333	6

\A	\B
/DSPYTD~D~	/XI(QTRN=1)~/DSP1STQTR~D~SNAME~A~G/XGCONTX~
SNAME~A~G	/XI(QTRN=2)~/DSP2NDQTR~D~SNAME~A~G/XGCONTX~
/DF~1~~~	/XI(QTRN=3)~/DSP3RDQTR~D~SNAME~A~G/XGCONTX~
	/XI(QTRN=4)~/DSP4THQTR~D~SNAME~A~G/XGCONTX~

Figure 2

another test to rank the records on YTD Sales and then put the NAME fields in an alphabetical order. This is done by executing the \A macro followed by the \C macro. How do you like your display now? Is it easier to use the program now? I want to cover a few loose ends with the readers at this point. You might want to know why I am choosing the macro name letters in kind of a senseless fashion. At this time, I can tell you it will show up when we get to the menu feature of this project! Some of the experienced readers can see if they can figure it out. I have NOT told you when to put the label-prefix (apostrophe) in many of the places where it is needed. Did you remember to put them in? If I told you every time, I do not think you would learn this necessity. I am not going to put them in very often any longer either. You should be well aware of their purpose and need. Do you agree? Also, I have stopped telling you when to make the columns wider or narrower. This is a matter of preference and the need to be able to print your worksheet on your paper size. Did you pay attention to the column widths? Did you find anyplace where it was a MUST to widen the column? I feel that these details are obvious and that you will learn the importance of your Preparation Form. You have not forgotten these preliminary steps, have you? If you have, I bet that you will learn to remember in the future how much faster you could have finished the project! I do not think that you should have to be told about this form. It should now be automatic! WATCH OUT! Anyway, what I am telling you is that this is a good time to think about these items and that you will find places where you will need these details to make your work easier and professional looking! Also, have you SAVED your Worksheet before testing new or revised macros? If not, expect to be Zapped some time when you have an error in your macro under test!

Another macro we need is one to Sort by OFFICE and within OFFICE to Sort by NAME. Our current database does not need both, but we should design the worksheet as flexible as possible and it is not hard to do. Here is the manual method:

```
Data Sort Primary-Key OFFICE Return Ascending Return
Secondary-Key NAME Return Ascending Return Go
```

Now translate that set of commands into the macro we will call \D.

1. Move cursor to cell N6, type /RNC\D, and press Return Twice.
2. Type /DSPOFFICE~A~SNAME~A~G and press Return.

Test the \D macro as always. You know the method. By the way, are you finding macros interesting, easy, and fun? I hope so, because I would suggest as your time permits that you go back over some old worksheets and you start to see how macros could be beneficial. They are what makes a project look and work professionally. We have now created macros to really Sort our database in many ways, BUT it is usually very necessary to PRINT our results!

Remember when designing a macro, the first step is to determine how to do the task manually without a macro. Perform the task and write down EVERY KEYSTROKE and COMMAND that you used on a piece of paper. When your procedure produces the correct results, simply translate the keystrokes into a macro. I CANNOT tell you this too many times. You will NOT be able to write complicated macros without this procedure! This next macro to PRINT the RESULTS is the hardest we have attempted so far. Let's think about our printed report, it will usually start from the top-left corner of the worksheet, which is where we will locate the cursor to get started:

1. Press the HOME key.
2. Enter the 1-2-3 Print Menu by typing /Print, select Printer Option, clear previous ranges by selecting Clear range, select Range option so we can define the desired print range, type a . (period) to anchor the Range.

Now, we need the lower-right corner of the Range we want to print. Usually it would be done by using the Down Arrow and Right Arrow keys to move the cursor to this corner. I want to make this macro a learning experience for the User where this could be a rather large worksheet, so it would take a lot of keystrokes using just Arrow keys. The macro would not know where this lower-right corner is located. It must be supplied with the intelligence and logic to find the corner! However, if we combine the END key with Arrow keys I think that we can get the macro to understand the task.

3. Press the End key, followed by pressing the Down key. A range A1..A6 will be suggested and highlighted. This is NOT the complete Print Range.
4. Press the End key again, followed by the Right Arrow key and the range A1..i6 will be highlighted which is still incomplete.
5. Press the End key and then the Down Arrow key. NOW, the entire range A1..I13 will be highlighted and displayed in the upper control panel!
6. Press Return, because we have the desired Print Range painted.
7. Add your column widths (if you have a Preparation Form, it will have this information.) within the Print Range. I find that I have 59 characters which will print on my 8-1/2 inch paper. The 1-2-3 defaults are 4 for the left Margin and 76 for the right Margin; so, we must reset the Margins!
8. Type Options Margins Left 10, and press Return.
9. Type Margins Right 69, and press Return.
10. Select Quit to return to the Print Menu. Make sure your paper is adjusted (aligned) and that your printer is turned on (If you forget to turn the printer on, it can cause the computer to "hang-up" and the only way to get out of this problem is a Re-Boot!) Thus, I always save my worksheet before I get to the printing!
11. Type Align, type Go and the Report should be typed.
12. Select Page (Advance the paper to next page.) and select Quit (Exits the Print Menu.).

Here is the manual commands I used to print the report:

```
Home /Print Printer Clear Range
Range   (period) End Down End Right End Down Return
Options Margins Left 10 Return Margins Right 69 Return Quit
Align Go Page Quit
```

Again, as always we will translate these commands into keystrokes to design the print macro. We have some special keys to translate into the macro. They are HOME, END, Down Arrow, Right Arrow, and again the Return key. We have done some of these before; so, this should be fairly easy. Our macro, called \E, which will print the report, looks like this (I split the macro into two label cells to make it visually easier)! Type the following starting at cell O6:

```
{HOME}/PPCRR.{END}{DOWN}{END}{DOWN}{END}{DOWN}
OML10~MR69~AGPQ
```

Can you follow the sequence? If not, enter it with the keyboard and watch the required strokes. It will become clear to you! Now, as always, we MUST name the macro:

Move the cursor to cell O6, type /RNC\E, and press Return twice.

Check the paper alignment in the printer and verify that the printer is ON! Test the macro by holding ALT key down and pressing the E key. How did you do?

This completes the work on our project for this time. With the next article we will finally reach our goal of a menu operated program that any User could work. Before I sign off, I would like to touch on a few items. A few months ago, I did a short review on the MyCalc spreadsheet and since that time a few bugs have shown up. The Software Toolworks, who published the program, reworked the software and they have offered the Users who registered their program a FREE UPDATE. All that they required was that you return the original disk to them! Besides the free update, they offered to sell any program that they have at a 10% discount and they also waived the $3.00 shipping charge! How many software suppliers are doing this? I call this an outstanding supplier that goes out of their way with Customer Service and they should get our business! I ordered the MYWRITE software with the 10% off and no shipping charge. I have been looking for a simple-to-operate wordprocessor to use where I do not want to use my WordStar software for a simple note or letter! I hope that the readers will have taken advantage of this offer before they read this article. I am writing this in November 1985.

I am sure that the readers know that I like the LOTUS 1-2-3 software by now! I have tested the LOTUS SYMPHONY software on my son's IBM and found that this is too much bother for me at this time. However, I did like the idea of the integrated wordprocessor. I was talking to a friend of mine about this need and he supplied a macro for 1-2-3 that provides a simple answer to my need. I want to pass it on to my readers, but I do not know who originated the macro so that I could give them the credit for it.

Wordprocessing Macro For LOTUS 1-2-3

```
\W    {?}~
      /RJ~
      {END}{DOWN}
      /XG\W~
```

1. Set the column width A to 6, or whatever your left margin will be.
2. Set B column width to 60 or whatever line width you desire.
3. Put the macro in some out-of-the-way place on the worksheet; such as, AA1.
4. Name the macro \W with the /RNC command.
5. Put the cursor in column B.
6. Type ALT-W.
7. If you type over 240 characters, 1-2-3 will beep, so hit Return and press F2 Edit key.
8. Press the Down key 1 or 2 times for paragraph spacing.
9. Have you ever wanted a two column text? This macro will do that, also!
10. Set the B column to 25 or 30 characters wide. When you have completed your entry, use /Move to get one-half the text into column C that must be set to the same width as B column.

You will need to experiment a little to get use to it; BUT this is a really useful macro. Here is the macro revised a little. If you use the first \W macro for only one line of text, it will drop you to the bottom of the 1-2-3 spreadsheet. This revised macro will solve that problem with the /XI command:

```
\W    {?}~
      /RJ~
      /RNCtemp~{BS}{DOWN}~
      /XI@COUNT(temp)>=2~
      {END}{DOWN}
      /XG\W~
```

This is a good test for the readers that are working on their article project. It gives you another place to use the /XI and /XG commands. Be sure to try both of the macros! Even if you do not want to put text into your worksheet reports, it is a very interesting use of macros! Do any of the readers have any interesting macros? If so, send them to me so they can be shared with the other readers.

That is all for this article. Happy SPREADSHEETING! If you have anything I can help you with, please send it to me with a self-addressed and stamped you know what! I still get letters without and I have stopped answering these letters. If you do not get an answer, you know you did not put in an SASE.

Figure 3A

18-Nov-85 (SALES IN THOUSANDS OF DOLLARS)

SPREADSHEET Corner COMPANY
OFFICE SALES RANKING PLAN

OFFICE	NAME	1STQTR	2NDQTR	3RDQTR	4THQTR	YTD	YTD RANK
ATLANTA	ABBOTT,T.	178	188			366	1
BOSTON	WILSON,E	188	173			361	2
CHICAGO	MILNER,H	162	171			333	6
LOS ANGELES	RICE,L	195	161			356	3
OMAHA	JOHNSON,E.	178	163			341	5
PORTLAND	NEWMAN,D	169	178			347	4

Figure 3B

2

```
    \A                                \B
/DSPYTD~D~ /XNEnter Desired Quarter(1,2,3,or4):~QTRN~
                                      /DSPNAME~A~G
SNAME~A~G  /XI(QTRN=1)~/DSP1STQTR~D~SNAME~A~G/XGCONTX~
/DF~1~~~   /XI(QTRN=2)~/DSP2NDQTR~D~SNAME~A~G/XGCONTX~
           /XI(QTRN=3)~/DSP3RDQTR~D~SNAME~A~G/XGCONTX~
           /XI(QTRN=4)~/DSP4THQTR~D~SNAME~A~G/XGCONTX~
```

Figure 3C

```
/DSPNAME~A~G /DSPOFFICE~A~SNAME~A~G
```

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). Includes CP/M software support for Heath, CDR Systems and Magnolia. Call or write for info on HDOS support. Specify disk format.

6MHz mod — $59

Similar to our 4MHz modification, but increases the CPU speed to 6MHz. Requires some soldering on the CPU board. Includes a replacement Z80B (6MHz). May require replacing additional parts. Some technical knowledge is recommended for installation. CP/M support only. 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

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. Includes WORD Plus spelling checker by Oasis - normally a $150 item by itself! We offer this package at discount and also include a function/cursor key patch for the H19/H89.

CP/M - specify format $115
MSDOS - specify computer $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	Visual 1050
DEC Rainbow	Magnolia	Royal/Adler	Xerox 820
		Sanyo 1100	Zorba

Now includes 44 formats! Uses a modified BIOS which is included with the program. Allows reading of 40-track disks in an 80-track drive.

For H37 with Heath CP/M $59
For CDR BIOS 2.91 . . . . . $49
Check for Magnolia version.

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

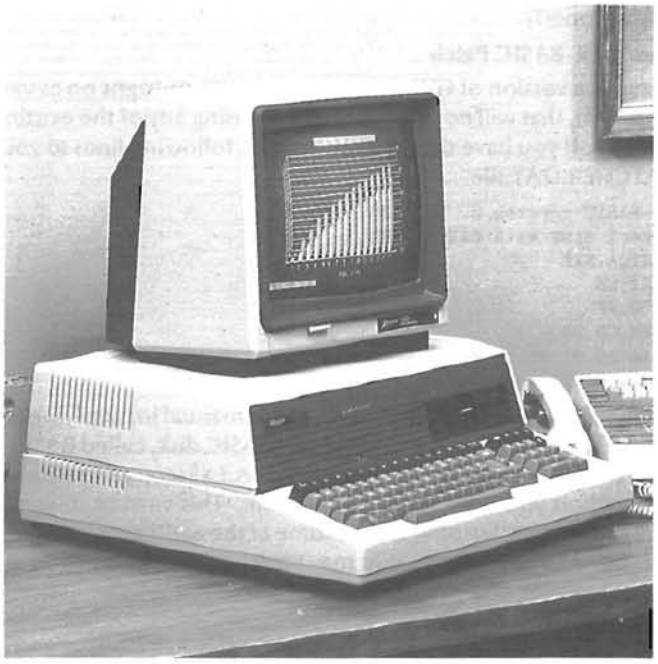
20663 Ave. 352 Woodlake, CA 93286

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

ZPC Update #8



Pat Swayne
HUG Software Engineer

This is the eighth in a series of articles in support of ZPC, a program that allows you to run IBM PC software on H/Z-100 (dual processor) computers. This is the second article in the series that has delt with version 2 of ZPC.

In this ZPC Update, I will present a patch for ZPC itself (version 2 only) that alters the way it emulates the Control-Break key sequence. I will also discuss how to run Lotus 123 release 2, and present a correction to the DBASE III Plus patch, and patches for a new release of Heath/Zenith GW-BASIC and some more versions of MultiMate not covered by previous patches.

Note: A patch to correct a bug in PATCHER.COM that could affect some patches was presented in the last ZPC Update article. If you missed that article, send me a self-addressed, stamped envelope (c/o HUG) and indicate that you want a copy of ZPC Update #7.

ZPC Control-Break Patch

When you type Control-Break on a real PC-type computer, it does two things. It generates an interrupt, and it returns a keyboard scan code of 0. When ZPC emulates Control-Break (when you type F0-F12), it generates the interrupt, but does not return a scan code at all. Most programs use the interrupt to "see" Control-Break, but there are a few that look for the 0 scan code. Among the programs that look for the scan code are Lotus 1-2-3 release 2 and GW-BASIC version 1. I have worked out a patch for ZPC Version 2 (but not ZPCSM.COM) that causes it to generate the proper scan code, as well as the interrupt when you type F0-F12. To make the patch using DEBUG, create a file called FIXBRK.DAT that contains these lines:

```
A236B
JMP 38D8
JNZ 2375
MOV AX,1A1B
JMP 23A4

A38D8
CMP AL,2
JNZ 38EC
MOV BYTE PTR [20DC],0
POP ES
POP DS
POP BX
POP AX
INC AX
INC AX
PUSH AX
MOV AX,0
RET
CMP AL,5B
JMP 236E

W
Q
```

Copy FIXBRK.DAT and DEBUG.COM to your ZPC system disk, log on to the disk, and enter

```
DEBUG ZPC.COM <FIXBRK.DAT
```

at the system prompt, and hit RETURN (you must be in the Z-100 mode). The patch will be applied for you. If you would rather use PATCHER to apply the patch, add these lines to the end of your PATCHER.DAT file:

```
ZPC Break key fix
Insert a disk containing ZPC.COM
ZPC.COM
```

```
226B,E9,6A,15,75,05,B8,1B,1A,EB,2F
37D8,3C,A2,75,10,C6,06,DC,20,0,7,1F,5B,58,40,40,50
37E8,B8,0,0,C3,3C,5B,E9,7D,EA
z
```

Follow the instructions in your ZPC manual for using PATCHER to apply the patch.

Running LOTUS 1-2-3 Release 2

Lotus 1-2-3 release 2 contains a program called INSTALL that is used to set it up for your screen and printer. If you do not alter the default screen settings when you run INSTALL, you can run Lotus 2 under ZPC in video mode 7 without patching, but Lotus will not draw graphs in this mode. (You must use UNLOCK to remove copy protection first.) If you install the screen as an "IBM color card, single color monitor", LOTUS will work in video mode 2 or 3 with the ZHS circuit installed, without patches. It can be patched to work without ZHS, but the patches are not in a fixed location, and so PATCHER cannot be used to do the patching. A special patch utility must be used, as in the case with a few other programs. I have written a patcher, called FIXLTS.COM, that can make the patches. If you type in and run the following BASIC program, it will create FIXLTS.COM. Or, if you prefer, use the source code at the end of this article.

```
10 REM THIS PROGRAM CREATES FIXLTS.COM
20 DEFINT A-I:OPEN "O",1,"FIXLTS.COM
30 S=0:S1 = 23556 :FOR I=1 TO 220
40 READ B:S=S+B:PRINT #1,CHR$(B);
50 NEXT I:IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 190,209,1,191,93,0,185,11,0,252
80 DATA 243,164,191,124,0,51,192,185,3,0
90 DATA 243,171,186,92,0,180,15,205,33,254
100 DATA 192,117,4,205,32,0,0,199,6,106
110 DATA 0,1,0,186,220,1,180,26,205,33
120 DATA 139,14,6,0,129,233,220,1,186,92
130 DATA 0,180,39,205,33,60,2,115,220,137
140 DATA 14,35,1,191,220,1,176,186,187,218
150 DATA 3,242,174,117,19,38,57,29,117,247
160 DATA 131,199,2,81,185,10,0,176,144,243
170 DATA 170,89,235,228,139,14,35,1,191,220
180 DATA 1,176,186,187,217,3,242,174,117,11
190 DATA 38,57,29,117,247,198,69,2,144,235
200 DATA 241,139,14,35,1,191,220,1,176,186
210 DATA 187,216,3,242,174,117,11,38,57,29
220 DATA 117,247,198,69,2,144,235,241,199,6
230 DATA 125,0,0,0,139,14,35,1,186,92
240 DATA 0,180,40,205,33,10,192,116,2,205
250 DATA 32,186,92,0,180,16,205,33,254,192
260 DATA 116,243,186,201,1,180,9,205,33,205
270 DATA 32,68,111,110,101,46,13,10,36,49
280 DATA 50,51,32,32,32,32,32,83,69,84
```

Copy FIXLTS.COM to your system disk, and enter

```
FIXLTS d:
```

where d: is the designation of the drive containing the Lotus disk that has 123.SET on it. Be sure you run INSTALL before you make the patch, and if you ever run INSTALL again to change something, you will probably have to redo the patch.

Note: I do not know why Lotus 1-2-3 release 2 will not work if you try to install it for a color monitor. If I find out, I will include the information in a future ZPC article.

Dbase III Plus Patch Correction

When Dbase III Plus is patched with the patch supplied with ZPC Version 2, it will not send output to a printer properly. To correct the problem, modify the Dbase III Plus section in PATCHER.DAT so that it looks like this:

```
DBASE III + vers  1.0
Insert unlocked system disk 1, containing DBASE.LD1.
DBASE.LD1
3274,90,90,90,90,90
327E,90
36AB,90,90,90,90,90
36B5,90
1CFA9,EB    <-- add this line
z
```

Then use PATCHER to patch Dbase III Plus again, and it will print properly.

New GW-BASIC Patch

There is a version of GW-BASIC, dated 1-29-86 (signs on as version 2.15), that will not work under ZPC using any of the existing patches. If you have this version, add the following lines to your PATCHER.DAT file.

```
GW-BASIC version 2. 1-29-86
Insert disk with BASICA.EXE.
BASICA.EXE
6A21,EB
6B7D,FF
7FE0,CD,90
7AAA,B8,07,0E,CD,10,90,90
z
```

Then follow the instructions in your ZPC manual for applying the patch. There are two files on the GW-BASIC disk, called BASICA .EXE and BASICA.COM. Use only BASICA.EXE (copy it to another disk) when you want to run GW-BASIC. This version does not work as efficiently under ZPC as some of the earlier versions, so if you have an earlier one, you may want to use it instead.

More MultiMate Patches

Two patches for MultiMate version 3.3 have already been released, but I have found a release that does not work with either (WP.EXE is dated 10-24-84). If your MultiMate 3.3 does not work with the patches released, add these lines to your PATCHER .DAT file:

```
MULTIMATE version 3.3, 10-24-84
Insert the System disk containing WP.EXE
WP.EXE
F7A5,90,90,90,90,90
F7E9,90,90,90,90,90,90,90,90,90,90
F8CC,90,90,90,90,90
F8FA,90,90,90,90,90
F969,90,90,90,90,90,90,90,90,90,90
F9DC,90
F9EF,90
z
```

Use PATCHER to apply the patch, as instructed in your ZPC manual. Here is a patch for an earlier version of MultiMate that a user wanted to use under ZPC:

```
MULTIMATE version 3.22 (IBM)
Insert the System disk containing WP.EXE.
WP.EXE
D449,90,90,90,90,90
D465,90,90,90,90,90,90,90,90,90,90
D4A5,90,90,90,90,90,90,90,90,90,90
D4E3,90,90,90,90,90
D4F2,90,90,90,90,90
D514,90,90,90,90,90
D523,90,90,90,90,90
D568,90,90,90,90,90,90,90,90,90,90
D5D3,90
D5E6,90
z
```

Upgrade Your ZPC

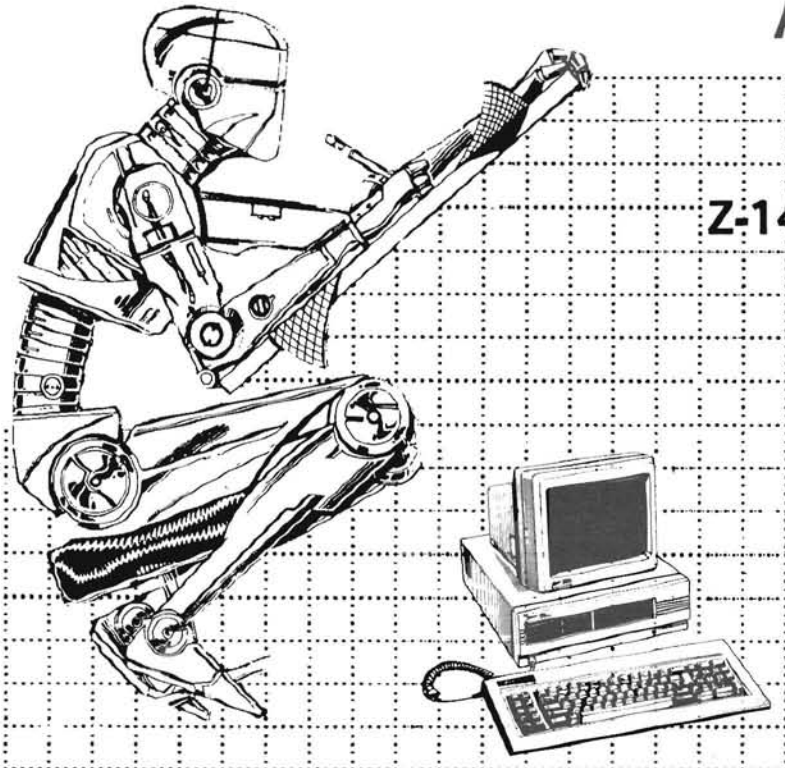
If you are still using ZPC version 1, you're missing the power and versatility of ZPC Version 2. You can upgrade by sending your

original HUG ZPC disk and $20.00 to Heath User's Group, Attn: Nancy Strunk, Hilltop Road, St. Joseph, MI 49085. If you have both the old ZPC and the ZPC Support Disk, send both in and $15.00 for the upgrade. Make checks payable to: Heath Users' Group.

FIXLTS Source Listing

```
        PAGE    ,132
;       FIXLTS -- FIX LOTUS 1-2-3 RELEASE 2 TO WORK UNDER
;       ZPC WITHOUT ZHS
;
;       BY P  SWAYNE, HUG  13-JUN-86

;       DEFINITIONS

OPENF   EQU     0FH             ;OPEN FILE
CLOSEF  EQU     10H             ;CLOSE FILE
SETDTA  EQU     1AH             ;SET DTA ADDRESS
READF   EQU     27H             ;READ FILE (RANDOM BLK.)
WRITEF  EQU     28H             ;WRITE FILE (RANDOM BLOCK)

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     6
BUFSIZ  LABEL   WORD
        ORG     5CH
FCB     LABEL   BYTE            ;DEFAULT FCB ADDRESS
        ORG     100H

START   MOV     SI,OFFSET LTSNAM;POINT TO "123.SET"
        MOV     DI,OFFSET FCB+1 ;PUT IT HERE
        MOV     CX,11           ;11 CHARS
        CLD
        REP     MOVSB           ;MOVE IT IN
        MOV     DI,OFFSET FCB+32;POINT TO RECORD COUNTERS
        XOR     AX,AX
        MOV     CX,3
        REP     STOSW           ;CLEAR RECORD COUNTERS
        MOV     DX,OFFSET FCB
        MOV     AH,OPENF
        INT     21H             ;OPEN FILE
        INC     AL
        JNZ     OPENED          ;OPEN OK
FILERR  INT     20H             ;ELSE, EXIT
PCHSIZ  DW      0               ;STORE PATCH SIZE HERE
OPENED: MOV     WORD PTR FCB+14,1 ;SET RECORD SIZE TO 1
        MOV     DX,OFFSET BUFFER
        MOV     AH,SETDTA
        INT     21H             ;SET DTA TO BUFFER
        MOV     CX,BUFSIZ       ;GET BUFFER SIZE
        SUB     CX,OFFSET BUFFER;REMOVE PROGRAM SPACE
        MOV     DX,OFFSET FCB
        MOV     AH,READF
        INT     21H             ;READ FROM FILE
        CMP     AL,2
        JNC     FILERR          ;FILE ERROR
        MOV     PCHSIZ,CX       ;STORE FILE HERE
        MOV     DI,OFFSET BUFFER
SCODLP0:MOV     AL,0BAH         ;SEARCH FOR THIS
        MOV     BX,3DAH         ;AND PORT 3DAH
SCODLP  REPNZ   SCASB           ;SEARCH FOR IT
        JNZ     NXPCH           ;NOT FOUND, CHECK NEXT
        CMP     ES:[DI],BX      ;IS THIS IT?
        JNZ     SCODLP          ;NO, KEEP LOOKING
        ADD     DI,2            ;MOVE TO PATCH AREA
        PUSH    CX
        MOV     CX,10           ;PATCH 10 BYTES
        MOV     AL,90H          ;WITH "NOP"
        REP     STOSB           ;MAKE THE PATCH
        POP     CX
        JMP     SCODLP0         ;LOOK FOR MORE
NXPCH:  MOV     CX,PCHSIZ       ;GET SIZE AGAIN
        MOV     DI,OFFSET BUFFER
        MOV     AL,0BAH         ;SEARCH FOR THIS
        MOV     BX,3D9H         ;AND PORT 3D9H
SCODLP1:REPNZ   SCASB           ;SEARCH FOR IT
        JNZ     NXPCH1          ;NOT FOUND, CHECK NEXT
        CMP     ES:[DI],BX      ;IS THIS IT?
        JNZ     SCODLP1         ;NO, KEEP LOOKING
        MOV     BYTE PTR 2[DI],90H      ;MAKE THE PATCH
        JMP     SCODLP1         ;LOOK FOR MORE
NXPCH1: MOV     CX,PCHSIZ       ;GET SIZE AGAIN
        MOV     DI,OFFSET BUFFER
        MOV     AL,0BAH         ;SEARCH FOR THIS
        MOV     BX,3D8H         ;AND PORT 3D8H
SCODLP2:REPNZ   SCASB           ;SEARCH FOR IT
        JNZ     PCHDN           ;NOT FOUND, DONE
        CMP     ES:[DI],BX      ;IS THIS IT?
        JNZ     SCODLP2         ;NO, KEEP LOOKING
        MOV     BYTE PTR 2[DI],90H ;MAKE THE PATCH
        JMP     SCODLP2         ;LOOK FOR MORE
PCHDN:  MOV     WORD PTR FCB+33,0 ;RESET RECORD ADDRESS
        MOV     CX,PCHSIZ       ;COUNT OF BYTES TO WRITE
        MOV     DX,OFFSET FCB
        MOV     AH,WRITEF
        INT     21H             ;WRITE TO FILE
        OR      AL,AL
        JZ      PCHOK           ;WRITE OK
WRERR:  INT     20H             ;ELSE, EXIT
PCHOK:  MOV     DX,OFFSET FCB
        MOV     AH,CLOSEF
        INT     21H             ;CLOSE FILE
        INC     AL
        JZ      WRERR           ;ERROR IN CLOSING
        MOV     DX,OFFSET DONE
        MOV     AH,9
        INT     21H             ;ELSE, SAY "DONE"
        INT     20H             ;AND EXIT

;       DATA AREA

DONE    DB      'Done.',13,10,'$'

LTSNAM  DB      '123     SET'

BUFFER  LABEL   BYTE            ;BUFFER STARTS HERE

CODE    ENDS
        END     START
```

FIRST CAPITOL DESIGNS THE BEST VALUES IN SYSTEMS AND SOFTWARE



Z-148 38 Mb System

$1999

- ✲ blazingly fast
- ✲ 8 mHz with no wait states
- ✲ new high speed disk controller
- ✲ 640 K RAM
- ✲ amber graphics monitor
- ✲ 38 Mb disk and 1 floppy
- ✲ serial and parallel ports
- ✲ color graphics adaptor

CAD Package Special

Get into CAD for a low cost ... our CAD package special includes Generic CADD, a program amazingly similar to AUTOCAD (TM), and the Microsoft mouse. Special price only $199, through September 30, 1986. If purchased with Z-148 38 Mb system, only $149.

Options: DotPlot — will allow you to output Generic CADD plot files to your dot matrix printer, if you don't have a plotter, only $24.95.

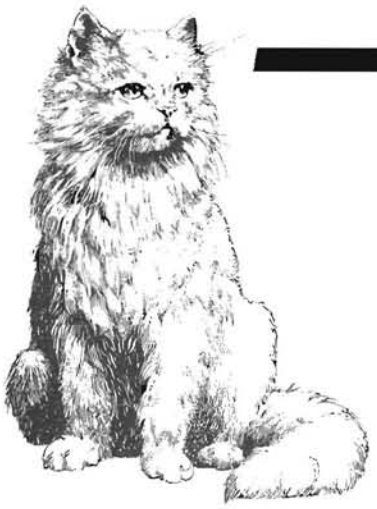
AutoConvert — convert your Generic CADD files to AUTOCAD DFX format and vice versa, only $24.95.

ZENITH | data systems

AUTHORIZED SALES AND SERVICE

First Capitol Computer

First Capitol Computer
1106 First Capitol Dr.
St. Charles, MO 63301
(314) 724-2336



MACAT

A Disk Cataloging System

Louis M. St. Martin
860 Hillcrest Drive
Pomona, CA 91768

Prologue

The idea for an MS-DOS disk catalog system began with discovery of a file, "REPRINT.COM" on a disk. A quick search through the notebook where copies of disk directory listings and DOC, README, TXT, etc., are kept, turned up nothing. Nor could I find it on any other disk by reading directories. A catalog system by which a file could be traced and identified was no longer merely desirable, it had become a necessity. In my H-89, CP/M days, I used MCAT to manage my library. Perusal of BBS and COMPUSERVE files failed to reveal a ready-made MS-DOS equivalent. I did find a program 'DISKCAT-4' on the Pomona HUG BBS, but it's for IBM and Z-150s. An attempt to adapt it to my Z-100 using ZPC failed. I decided to develop my own disk catalog system.

Introduction To MACAT

MACAT is a system for compiling a catalog of disk filenames in two easily read indices for locating and identifying files. The system provides for updating the catalog as files are added, moved, and deleted. The system has the capability to append comments to the filename to aid the user in identifying files whose mnemonic titles seldom reveal their contents and/or purpose. The catalog is designed to permit maintenance by use of any popular full screen editor; with some precautions to avoid potentially destructive inclusions of text formatting codes.

MACAT consists of three modules; a batch file, MACAT.BAT, and two BASIC programs, WHAT.BAS and WHERE.BAS. To run the system, MS-DOS Version 2.0 and GW-BASIC or ZBASIC are required. A full screen editor is desirable. I use BSE, the Basic Screen Editor supplied in the MS-DOS Programmer's Utility Pack. The rationale for my choice of BSE is discussed later.

Disks to be cataloged must have a unique filename on them identifying the particular disk. The filename can be any combination of three characters, alpha and/or numeric, but it must be preceded by a hyphen, eg., "-103". WHAT.BAS looks for that distinctive filename and uses it to link the filenames on that disk to that number in the catalog. This disk filename must be the only one on the disk preceded by a hyphen or minus sign lest WHAT .BAS be confused. This filename can be added to the disk by a) opening a directory of that name using the "MKDIR" or "MD" command, or b) creating a blank file using an editor. There is an advantage in using an editor to add this filename; it will be displayed by VFILER or HFM, whereas a directory name will not. The disadvantage is that some editors allocate a minimum disk space for any file opened.

Operation of MACAT is achieved by placing the MACAT system disk in drive A:, entering the command "MACAT [d:]", where "d" is the drive containing the disk to be included in the catalog. The batch file will control the program without attention from the user until that part of WHAT.BAS is reached where comments are to be added, if desired, to the catalog. At the conclusion of each transit of the complete batch file, two indices are available, (see Figures 1 and 2). WHAT is a listing arranged in alphabetical order of filenames (leftmost field) within the disk (middle field), together with the comments appended by the user (rightmost field). WHERE is an alphabetically ordered index of filenames (leftmost field) each followed by the number(s) identifying the disk(s) on which each file is located (rightmost field). Each time a disk is processed, files scratched since the previous processing are deleted from both WHAT and WHERE indices. Likewise, files added since the last processing are added to the two indices.

The batch file, MACAT.BAT, must be tailored for the individual user's resources. The batch file presented in Figure 3 includes provisions for taking a look at the disk directory by means of VFILER to read any ASCII files to identify them and make a decision as to the desirability of keeping or deleting those files no longer needed on the disk about to be cataloged. Then MACAT .BAT calls COVER so a copy of the modified directory of the disk can be printed. If the user does not have either of these public domain programs, they should delete those lines from the batch file or replace them with others that accomplish the same tasks.

The two lines calling BSE WHAT and BSE WHERE may also be deleted or modified to call the user's own editor.

```
 2281
-000.UTL      000 MACAT disk catalog system disk
COVER.COM     000 Directory printout
CPMDIR.BAS    000 CM/P disk catalog module of MACAT
CRTSAVER.COM  000
DISCAT.BAT    000 An early version of MACAT.BAT
DISCAT.DTA    000 Disk catalog data a la PC-FILE
DISCAT.HDR    000
DISCAT.INX    000
FLAGS.COM     000 MS-DOS flag manager
LAST.BSE      000 Last file called by BSE
LS.COM        000 Disk status review
MACAT.BAT     000 Disk catalog system batch file-single disk
MACAT.RAM     000 MACAT.BAT for RAM-needs updating
MACPM.BAT     000 MACAT.BAT for CP/M disks
MORE.COM      000
PC.COM        000 Part of ZPC
PC-DEF.EXE    000 Part of PC-FILE
PC-EXPOR.EXE  000
PC-FILE.EXE   000
PC-FILE.PRO   000
PC-FIX.EXE    000
PC-IMPOR.EXE  000
PC-LABEL.EXE  000
PC-OVL.EXE    000
PC-PRINT.EXE  000
PC-SETUP.EXE  000
PC-SORT.EXE   000
PC-UTIL.EXE   000
PCP.COM       000 Printer manager
RAMDISK.DVD   000
RDCPM.COM     000
RMCOL.COM     000
SAMPLE.BAS    000 Sample data base for PC-FILE
SCREEN.RPT    000 PC-FILE ?
SCRNCLK.COM   000
SETZPC.COM    000 Part of ZPC
SORT.EXE      000
THISDISK.BAT  000 MACAT cut down to a catalog for each disk
TWODISK.BAT   000 MACAT for two 5" disks
VFILER.COM    000
WHAT.         000 Disk catalog-filename by disk
WHAT.BAS      000 MACAT catalog program
WHAT.BKP      000 WHAT backup
WHERE         000 Merged catalog file
WHERE.BAS     000 Merging program for MACAT
X.            000
Z100.COM      000 Part of ZPC
ZDIR.COM      000
ZPC2.COM      000 Part of ZPC
-001.DOS      001 MS-DOS working disk
ALTCHAR.SYS   001
AUTOEXEC.BAT  001
CHKDSK.COM    001
```

Figure 1

Two BASIC programs perform the principal tasks of generating the two indices comprising the catalog. The first of these, WHAT.BAS, strips the sorted disk directory of its header and supplemental information associated with individual filenames; links each filename in the directory with the disk identifying number; compares the disk number with the previously generated catalog; and makes a decision as to which filenames from the previously generated catalog should be preserved, which should be deleted, and adds new filenames and descriptive comments. The output of this program is the completed file, WHAT. The second program, WHERE.BAS, uses the output of WHAT.BAS to merge the filenames by seeking all disks having common filenames and listing them together.

```
YACHT2        104
YAS00.DSP     113
YAS00.SCR     113
YAS002.SCR    113
YAS003.SCR    113
YCLEE11.85    102
YEARS         111
YEARSR        111
Z-ARTIST.BAS  105, D05
Z-ARTIST.DOC  105, D05
Z-ARTIST.EXE  105, 113, D05
Z-CHART       D45
Z100          D11
Z100.ASM      D27
Z100.COM      000, 149, 150, 151, 155, 156, D27
Z150          D11
Z80.LIB       109
ZBASIC        D44
ZBASIC.COM    105, 106, 113, 118, 157, D26
ZBCAT.BAT     157
ZCONVERT.COM  D08
ZD.COM        113
ZDIR.COM      000, 001, 116, 118, 136, 155
ZDOSX.OBJ     134
ZEROCLK.COM   108
ZPC.ASM       D27
ZPC.COM       151, 156, 158
ZPC.EVL       128
ZPC.TXT       104
ZPC1.COM      D27
ZPC2.COM      000, 149, 150, 151, 155, D27
ZPC3.COM      D27
ZSID.COM      109
```

Figure 2

The catalog can be interrogated with an editor and printed after compilation.

Be careful not to destroy the backup catalog file, WHAT.BKP, after you've invested some effort and time in it. The batch file provides this and it should not be edited under any circumstances. It is assurance against accidental erasure or mutilation of the working file. ATTRIB.COM, a program that affords some protection from accidental erasure, is available on HUG part number 885-3025-37, ZDOS/MSDOS Utilities. An alternative is FLAGS in the MS-DOS Programmer's Utility Pack. The number at the top of WHAT and WHAT.BKP must not be altered or deleted else the next processing of MACAT will result in eventual loss of filenames from the catalog.

A double-sided five inch 48 tpi disk accommodates a catalog of approximately 1000 entries before the BSE editor gags for lack of disk space; an average of 25 files each on 40 disks. The WHAT catalog will have grown to about 31k when this occurs. Switching the catalog operation to an 8 inch disk eliminates this problem. An alternate solution is to put ancillary programs like the editor on separate disks. Another is to divide the catalog at about 1000 lines, storing each section on separate disks. Later, we discuss other options for those with more files than can be accommodated by a single five inch disk.

Detailed Discussion Of The System

This system could have been written as a monster BASIC program, but since certain necessary parts of the process were readily available from the HUG software or as supplied with MS-DOS 2.0, it was decided to utilize them in batch files and reduce the effort involved in developing the system.

The prototype batch file is shown in Figure 3. The first line clears the screen of clutter left over from previous activity. TYPER is a

handy program found on the same disk as ATTRIB.COM and other goodies useable by MACAT. TYPER's function is to allow printing of a label, with the disk name and number. ATTRIB clears the Read/Only flag of the WHAT.BKP file so the last valid output of a cataloging session is preserved. The third line checks for WHAT on the catalog disk. If it does not yet exist, the sequence jumps to avoid confusion on the screen with messages saying files could not be found. VFILER (or HFM, also found on the disk with ATTRIB.COM) is called. After completion of perusing the disk's contents, the batch file calls on COVER to do its thing. Next, the directory of the selected disk is copied into a file labeled "TEMP". TEMP is then sorted by HSORT (or the MS-DOS program SORT.EXE) into a file called "TEMP.$$$". HSORT.COM is another of those priceless gems on the HUG MSDOS/ZDOS Utilities disk. In the interest of conserving disk space, TEMP is deleted now. BASICA or ZBASIC is called and WHAT.BAS begins.

```
CLS
TYPER
IF NOT EXIST WHAT GOTO HERE
ATTRIB WHAT.BKP
COPY WHAT WHAT.BKP
ATTRIB WHAT.BKP R
 HERE
VFILER %1
A:
COVER
A
ECHO ON
DIR %1 >TEMP
HSORT TEMP TEMP.$$$
DEL TEMP
BASICA WHAT
IF EXIST !!! GOTO QUIT
BSE WHAT
HSORT TEMP SORT.CAT
BASICA WHERE
DEL TEMP* *
DEL SORT.CAT
:QUIT
IF EXIST !!! DEL !!!
```

Figure 3

Referring to Listing 1, at lines 150 to 220, WHAT.BAS opens the file, TEMP.$$$, generated by MACAT.BAT to count the number of files on the disk to be cataloged. At line 260 the program attempts to open the file, WHAT, generated by previous excursions through MACAT. If it does not find it, FLAG is set to -1. If WHAT is found, the program falls through to determine the number of filenames in the existing catalog and dimensions an array, OTHER$(), to the sum of filenames in both TEMP.$$$ and WHAT. At line 310 another array, NEWNAME$(), is dimensioned to accommodate the filenames in TEMP.$$$. The program tucks the filenames from TEMP.$$$ into this latter array at lines 340 to 420, meanwhile looking for a disk number, ie., a directory or filename preceded by a "-". If it does not find such a disk number before End-of-File at line 330, the program jumps to line 1050 where a message is displayed telling the user the program could not find a disk number and since nothing of meaning can be accomplished without it, the program opens a null file; "!!!", to alert MACAT and terminates itself. MACAT checks for "!!!", and if it exists, jumps to QUIT and exits to the operating system. This graceful exit is taken to eliminate frantic key punching to keep MACAT from proceeding merrily ahead.

```
10 '                    WHAT.BAS
20 '
30 ' STRIPS GARBAGE FROM TEMPORARY DIRECTORY FILES,
   LABELS EACH FILENAME WITH ITS SOURCE DISK IDENTI-
   FICATION, AND CREATES A MASTER CATALOG
40 '
50 '                Louis M. St.Martin
60 '                860 Hillcrest Drive
70 '                 Pomona, CA  91768
80 '                   714/622-3248
90 '
100 '
110 '                  15 DECEMBER 1985
120 '
130 LOCATE ,,0
140 '********************************* COUNT DISK FILES
150     OPEN "I",#1,"TEMP.$$$"
160 N = 0   NU.COUNT = 0
170 FLAG = 0
180     IF EOF(1) THEN 220
190             INPUT#1, FILENAME$
200 N = N + 1
210 GOTO 180
220 CLOSE#1
230 '********************************* GET DISK NUMBER
240 OPEN "I",#1,"TEMP.$$$"
250     ON ERROR GOTO 300
260 OPEN "I",#2,"WHAT"
270     ON ERROR GOTO 0
280     INPUT#2,COUNT
290 DIM OTHER$(COUNT+N)
300     IF ERL = 260 THEN FLAG = -1
310 DIM NEWNAME$(N+1), FILENAME$(N), NEWC$(N)
320  I = 1
330     IF EOF(1) THEN CLOSE#1 : IF DISC$ = ""
        THEN 1050 ELSE 450
340  INPUT#1, FILENAME$
350     IF LEFT$(FILENAME$,1) = "-"
        THEN DISC$ = MID$(FILENAME$,2,3)
360 '********************************* GET DISK FILENAMES
370     IF FILENAME$ = "" THEN 410
380             PLACE = INSTR(FILENAME$,CHR$(32))
                        IF PLACE > 8 THEN PLACE = 8
                        ELSE PLACE = PLACE - 1
390     NEWNAME$(I) = LEFT$(FILENAME$,PLACE) + "."
                        + MID$(FILENAME$,10,4)
                        + SPACE$(8-PLACE)
410     I = I + 1
420 GOTO 330
430 '
440 '****************************** CREATE MASTER CATALOG
450 OPEN "O",#3,"TEMP.CAT"
460     IF FLAG = 0 THEN 480 ELSE 610
470    IF EOF(2) THEN 610
480     INPUT#2,MASTER$
490             IF MASTER$ = "" THEN 470
500             IF ASC(MASTER$) < 33 THEN 470
                        'SKIP GARBAGE
510             IF INSTR(MASTER$,DISC$) > 0 THEN 570
520 ' ************** PRESERVE FILENAMES ON OTHER DISKS
530  LOCATE 13,1 : COLOR 3,0 . PRINT "Transferring disc #" ,
                    MID$(MASTER$,14,3); " files"
540                 NU.COUNT = NU.COUNT + 1
550                 PRINT#3,MASTER$
560             GOTO 470
570 M = M + 1
580             OTHER$(M) = MASTER$
590                 GOTO 470
600 CLOSE#2
610 FOR J = 1 TO M
620              IF OTHER$(J) = "" THEN 710
                        ' SKIP BLANKS IN OLD DIRECTORY
630     FOR K = 4 TO N
640              IF NEWNAME$(K) = "" THEN 700
                        ' SKIP BLANKS IN NEWNAMES
650 ' ********************** PRESERVE CURRENT FILENAMES
660         IF INSTR(OTHER$(J),NEWNAME$(K)) > 0
```

```
                        THEN 670 ELSE 700
670             PRINT#3,OTHER$(J)
680             NEWNAME$(K) = ""
690             NU.COUNT = NU.COUNT + 1
700     NEXT K
710 NEXT J
720 '
730 ' ******************************** ADD NEW FILENAMES
740 FOR K = 4 TO N
750     IF NEWNAME$(K) = "" THEN 870
                                ' SKIP BLANKS IN NEWNAMES
760     IF LEFT$(NEWNAME$(K),4) = "TEMP" THEN 870
                                ' SKIP TEMPORARY FILENAMES
770 CLS : BEEP : COLOR 4,7 : LOCATE 13,1
780 PRINT " " ; NEWNAME$(K) ;" " ; : COLOR 7,0 :
      PRINT " is a new filename on this disc  You may"
790 PRINT TAB(5) , "a) add a descriptive comment,"
800 PRINT TAB(5) , "b) hit RETURN to skip commenting, or"
810 PRINT TAB(5) ; "c) enter '\' to leave " ; : COLOR 6,0 :
    PRINT NEWNAME$(K) ; : COLOR 7,0 : PRINT " off the list "
820  INPUT , NEWC$(K)
830     IF NEWC$(K) = "\" THEN 860
840             NU.COUNT = NU.COUNT + 1
850             PRINT#3,NEWNAME$(K) + DISC$ + " " + NEWC$(K)
860             NEWNAME$(K) = ""
870 NEXT K
880 CLOSE#2 , CLOSE#3
890 '
900 CLS
910 OPEN "I",#3,"TEMP.CAT"
920 OPEN "O",#2,"WHAT"
930 OPEN "O",#4,"TEMP"
940             PRINT#2,NU.COUNT
950             PRINT#4,NU.COUNT
960     IF EOF(3) THEN 1020
970             INPUT#3,MASTER$
980             PRINT#2,MASTER$
990             PRINT#4,LEFT$(MASTER$,17)
1000 LOCATE 13,1,0 : COLOR 0,3   PRINT " Writing disc #",
       MID$(MASTER$,14,3) , " filenames to output file. "
1010             GOTO 960
1020 CLOSE #2 : CLOSE#3 : CLOSE#4
1030  SYSTEM
1040 ' ********************** NO DISC ID NUMBER MESSAGE
1050 LOCATE 13,1 : COLOR 4,7 : BEEP
1060 PRINT "YOU FAILED TO ASSIGN A NUMBER TO THIS DISC
    RE-ENTER DOS AND 'MKDIR -NNN' THEN BEGIN AGAIN "
1070 OPEN "O",#5, "!!!"   PRINT#5, A   CLOSE#5
1080 SYSTEM
```

Listing 1

MS-DOS and CP/M directories fill out filenames to occupy twelve places by adding spaces between the end of the prefix and the period ahead of the extension. Lines 380 and 390 of WHAT.BAS eliminate those spaces so the finished product in WHAT is identical to what one would type to search for a filename with an editor.

If WHAT.BAS finds the needed disk number in TEMP.$$$, the program moves to line 450 where a new file, "TEMP.CAT", is opened for temporary storage of filenames from both TEMP.$$$ and WHAT after processing by WHAT.BAS. At line 510 the program examines each filename in WHAT to see if it contains a string matching the current disk number. If the match is not made, the program falls through and the filename is printed immediately in TEMP.CAT at line 550. If a match is found, the program jumps to 570-580 and the string containing the matching filename is entered into the array, OTHER$(). When End-of-File of WHAT is reached, the program moves on to line 610 where treatment of filenames from NEWNAME$(), ie., TEMP.$$$, and the strings in the array, OTHER$(), begins.

Remember, the filenames in OTHER$() are those from the existing catalog containing a string identical to the current disk number. In lines 610 to 710, the program compares the filenames in OTHER$() with those in NEWNAME$(). If it finds the NEWNAME$() filename in a string from OTHER$() at line 660, the string from OTHER$() is printed in TEMP.CAT. The reason for printing the OTHER$() string instead of NEWNAME$() is that the former is comprised of the filename, the disk number, and any comment added when it was processed during a previous cycle through MACAT; otherwise, if the filename from TEMP.$$$ was preserved, the comment would be lost. Meanwhile, as WHAT .BAS matches old filenames with new, it nulls both to clear them from the arrays. The residue remaining when the J and K FOR-NEXT loops exhaust themselves are either filenames in NEWNAME$() which have been added since the last cycle of the current disk, or strings in OTHER$() associated with files deleted from the current disk since its last cycle. The latter are ignored from here on, and are left out of the updated catalog being compiled in TEMP.CAT.

Filenames remaining in the array NEWNAME$(), are processed through lines 740 to 870 where the user is prompted to decide whether to add a comment, skip the comment, or tell WHAT.BAS to leave the filename out of the catalog. The filename, disk number, and comment, if any, are combined and printed in TEMP .CAT.

All the while TEMP.CAT is being updated with strings, a tally is being kept of the entries at lines 540, 690, and 840.

In lines 910 to 1010 the program transfers the data in TEMP.CAT to WHAT. But first, the count of entries in TEMP.CAT is entered into WHAT for use at line 280 when it is opened on the next cycle of MACAT. Lines 970 to 1010 create the updated version of the catalog. At lines 950 and 990 another new file, "TEMP" is printed with the count, filenames, and disk numbers, less comments. TEMP is the input to WHERE.BAS.

When the End-of-File of TEMP.CAT is obtained, WHAT.BAS closes all files and exits to MS-DOS.

This description has skipped some features of WHAT.BAS for sake of continuity and clarity. Error traps at 250, 270, and 300 and the flag test at 460 avoid catastrophic results if WHAT does not yet exist. The garbage filter at line 500 eliminates nonsense, such as non-ASCII characters in the directories. To keep the user from worry, messages at lines 530 and 1000 furnish assurance the program has not gone astray in endless loops.

The system now returns to MACAT.BAT where preparation of the second part of the catalog, WHERE, begins. However, in my batch file I call BSE so I can examine WHAT for anomalies and possible editing. For example, maybe I goofed and left a filename in that is utterly redundant to carry in the catalog. This is the last opportunity before WHERE is prepared to delete such filenames as COMMAND.COM, ALTCHAR.SYS, etc., from the catalog. These are files common to most disks in my library and there is no need to list them over and over again. So, I take advantage of the opportunity to delete them. Also, I like to have things tidy, and since WHAT.BAS appends the current disk to the bottom end of WHAT, I use BSE to move it to its proper place in the numerical sequence of disks listed in the catalog. Or, perhaps I've had second thoughts about the comment I added after some filename. And, I've even added filenames erroneously deleted while WHAT.BAS was running.

Moving on — TEMP, the file prepared along with WHAT, is sorted into SORT.CAT which is in turn used by WHERE.BAS to

prepare the merged index of filenames listed alphabetically with the disk numbers on which they appear. WHERE.BAS is fairly straight forward and works because the file it operates on is sorted. The filename count is input and used to dimension an array, FILENAME$(). At line 190, a garbage filter eliminates the occasional anomalous data.

Line 210 compares two adjacent filenames and their corresponding disk numbers. If the filenames match but the disk numbers do not, then the disk number associated with the second filename is appended to the filename of the first. If no match of filenames occurs or if both the filenames and the disk numbers are identical, no change is made in either filename and the program jumps to line 260 where another test is performed. If the two filenames are equal — remember, the disk numbers are also equal — the program jumps again, this time to return and thus skip printing anything out to the catalog WHERE. On the other hand, if the filenames are not the same, the first of the two filenames is printed to WHERE, it is cleared to a null at line 270, and the program returns to pick up the next filename. Filenames not printed at line 260 remain in the array until they are printed and cleared to a null, thus they accumulate disk numbers until they are finally printed. Line 230 sees that the last remaining filename is printed; if it was not previously matched and printed. Another of our reassuring messages is displayed at line 200.

The two indices, WHAT and WHERE, are now complete, so MACAT cleans up the residue of temporary files and exits back to DOS.

```
10 '                        WHERE.BAS
20 '
30 '      MERGES CATALOG FILES GENERATED BY WHAT.BAS
40 '
50 '                  LOUIS. M. ST.MARTIN
60 '                    (714) 622-3248
70 '                   16 NOVEMBER 1985
80 '
90 CLEAR
100 OPEN "I",#1,"SORT.CAT"
110     INPUT#1, COUNT
120 '
130 DIM FILENAME$(COUNT)
140 '
150 OPEN "O",#2,"WHERE"
160     FOR I = 1 TO COUNT
170     IF EOF(1) THEN 230
180     INPUT#1, FILENAME$ : FILENAME$(I) = FILENAME$
190             IF LEN(FILENAME$) < 4 THEN 170
200   LOCATE 13,1,1 : COLOR 0,6 :
                PRINT " Now merging filename " ,
                  PRINT LEFT$(FILENAME$,13)
210       IF LEFT$(FILENAME$(I),12) = LEFT$(FILENAME$(I-1),12)
             AND INSTR(RIGHT$(FILENAME$(I-1),3),
             RIGHT$(FILENAME$(I),3)) = 0
             THEN FILENAME$(I) = FILENAME$(I-1) + ", "
             + RIGHT$(FILENAME$(I),3) ELSE GOSUB 260
220     NEXT I
230 PRINT#2, FILENAME$(I-1)
240 CLOSE#1 : CLOSE#2
250 SYSTEM
260 IF FILENAME$(I-1) = FILENAME$(I)
             THEN 280 ELSE PRINT#2,FILENAME$(I-1)
270 FILENAME$(I-1) = ""         ' CLEAR STRING FILE SPACE
280 RETURN
```

Listing 2

Variations Of MACAT

Earlier we mentioned alternate solutions to the disk capacity problem. As the overload problem occurs while MACAT is running, one answer is limit the files on the working disk to those actually involved in compiling the catalog. Splitting the files between two disks is another option. Figure 4 is a system that accomplishes that. Another solution (see Figure 5) is to copy everything to a RAM disk, perform the compilation, and copy the finished catalog to floppy disk.

```
CLS
ECHO OFF
TYPER
IF NOT EXIST WHAT.BKP GOTO HERE
ECHO PUT DISK "TWO" IN DRIVE B:
PAUSE
COPY WHAT.BKP B:WHAT
:HERE
ECHO PUT DISK TO BE CATALOGED IN ITS ASSIGNED DRIVE
PAUSE
HFM
A:
DIR %1 >TEMP
ECHO PUT DISK "TWO" IN DRIVE B:
PAUSE
HSORT TEMP B:TEMP.$$$
B:
BASICA WHAT
IF EXIST !!! GOTO QUIT
DEL TEMP.$$$
A:
HSORT B:TEMP B:SORT.CAT
DEL B:TEMP* *
B
ECHO ON
BASICA WHERE
DEL SORT.CAT
A:
COPY B:WHAT A:WHAT.BKP
:QUIT
```

NOTE:

Files on Disk "ONE" -
- ATTRIB.COM
- HFM.COM
- HSORT.COM
- TWODISK.BAT
- TYPER.COM
- WHAT
- WHAT.BKP

Files on Disk "TWO" -
- BASICA.EXE
- WHAT.BAS
- WHERE.BAS

Figure 4

How about CP/M files? There is an excellent public domain program around; MCAT and XCAT are the version I'm familiar with. But I wanted a catalog system that compiled all my directories into a single file, using MS-DOS. The system shown in Figure 6 provides capability to mix MS-DOS and CP/M disks in a single system. To use it, put your regular MS-DOS catalog disk in drive A: and the CP/M disk in drive B:, enter "MACPM B:", and stand back. A special BASIC program is needed to allow the directory generated by RDCPM to be read by WHAT.BAS. To see why, one need only to compare the format of a directory read from an MS-DOS disk by DIR.COM with that of a CP/M disk directory provided by RDCPM. I'll not go into a line-by-line explanation of CPMDIR.BAS, Listing 3. Believe me, it does nothing more glamorous than picking the filenames out of the clutter and making sense of them.

Another format of MACAT, a variation that does not produce a catalog of all the disks in one's library, but does print an indi-

vidual catalog on the disk being cataloged, is presented in Figure 7. It's for folk who can remember which disk has what on it, but can't remember why they have each file, where they got it, or some other vital bit of information about the files on the disk. WHAT.BAS is needed, but WHERE.BAS is not used with this system.

```
CLS
TYPER
IF NOT EXIST WHAT GOTO HERE
ATTRIB WHAT.BKP
COPY WHAT WHAT BKP
ATTRIB WHAT.BKP R
COPY WHAT I:
:HERE
COPY BASICA.EXE I·
COPY HSORT.COM I
COPY WHAT.BAS I.
A:
VFILER %1
A·
COVER
ECHO ON
DIR %1 >I:TEMP
I:
HSORT TEMP TEMP.$$$
BASICA WHAT
IF EXIST !!! GOTO QUIT
DEL TEMP.$$$
A:
BSE I:WHAT
I.
DEL WHAT.BAS
HSORT TEMP SORT.CAT
DEL TEMP* *
A
COPY WHERE.BAS I:
I:
BASICA WHERE
DEL SORT.CAT
COPY WHAT A.
COPY WHERE A:
:QUIT
IF EXIST !!! DEL !!!
A.
```

Figure 5

Getting Along With MACAT

MACAT is slow. The first half dozen disks are cycled rather quickly. But after that prepare to spend a while processing using the GW-BASIC or ZBASIC interpreter, it takes about six and a half minutes to process a 1200 line catalog. With eight inch disks, the system consumes about six minutes. Compiled programs on my Z-100 with the CDR speed module installed reduces the time to five and a half minutes. The RAM disk version of MACAT does it in five minutes. Before installing the speed module and expanding my memory, I scheduled catalog update for times when I was nearby, but occupied with some other task I could perform while MACAT ran unattended such reconciling my checkbook, catching up on articles in REMark and SEXTANT, cleaning up paper files, reading the mail, or just walking away to let my Z-100 grind away.

Conscientious use of MACAT provokes tidiness. Searching for documentation for use in developing meaningful comments prompts cleaning up paper files. Be ready to embark on a mission of faithfully printing and religiously storing all the DOC, TXT, README, etc., files on your disks.

After getting started with MACAT, I began wondering if it's worthwhile to run all one's disks through the system. Some disks are not appropriate to catalog, such as disks currently in use for program development and single purpose disks containing files unique to the tasks performed. MACAT is most useful to keep track of seldom used important files, etc.

Files common to many disks and generally indispensable to system operation need not be cataloged for each and every disk on which they appear. That is the rationale behind providing the option to leave some files out of the catalog. There is the annoyance that these files will be presented for comment each time the disk is processed, but it is a small penalty compared to taking up space with redundant information. It is suggested that the location of the backup copy of such files be entered in the catalog so if one needs to return to a reliable source for a copy, it can be found quickly.

```
CLS
TYPER
IF NOT EXIST WHAT GOTO HERE
ATTRIB WHAT.BKP
COPY WHAT WHAT.BKP
ATTRIB WHAT.BKP R
:HERE
RDCPM DIR %1 >TEMP.CPM
BASICA CPMDIR
HSORT TEMP TEMP $$$
DEL TEMP
BASICA WHAT
IF EXIST !!! GOTO QUIT
BSE WHAT
HSORT TEMP SORT.CAT
BASICA WHERE
DEL TEMP* *
DEL SORT.CAT
:QUIT
IF EXIST !!! DEL !!!
```

Figure 6

If you edit WHAT (never edit WHAT.BKP), avoid introduction of TABS. Some editors use horizontal tabs (Hex 09) to eliminate white space in disk files and WHERE.BAS has difficulty digesting them.

About editors — I prefer BSE, the Basic Screen Editor supplied in the MS-DOS Programmer's Utility Pack. Two others I've tried, Software Toolworks "PIE" and "PeachText" introduce anomalies that garble the message displayed by line 530 in WHAT.BAS. PIE does something more serious; when WHERE.BAS is reading a TEMP file generated by a WHAT file processed through PIE, the string is shifted left one character and the output file, WHERE, is sprinkled with odds and ends of characters from the comments. These problems can be resolved by modifying the appropriate lines in the BASIC programs and are left as exercises for those waiting to tackle them. PIE can be patched by the vendor's instructions to eliminate the anomalies introduced by horizontal tabs.

The most important reasons for my preferring BSE are a) editing, particularly block movement, and b) its "shell" capability. Block movement is much more readily accomplished with BSE than other full screen editors I've tried. This is important if you want hard copy of your WHAT file neatly arranged in disk number order for quick visual access. MACAT puts the filenames of a disk cycled through the system at the end of WHAT, regardless of disk number. Putting the list in order afterward is a simple step with BSE. PIE, requires the writing of a macro to move, ie. — copy, move, and delete — a block. And PIE is limited to moving blocks of about twenty lines or less. Text manipulation with PeachText is cumbersome; too many menus to peruse and key through.


"Have Your Computer Talk To Ours!"
"HUGPBBS is on line, 24 hours a day, with over 10 megabytes of free software available for downloading. There's software for every Heath/Zenith operating system, with the majority being for MSDOS, and specifically the Heath/Zenith PC compatible computer systems. Also included is software for HDOS, CP/M, and MSDOS for the H/Z-100 computer system. In addition to this software is a message base through which you can exchange information with other HUG members. Have your computer call (616) 982-3956, 'The Heath Users' Group Personal Bulletin Board System', and make connection at 300, 1200, or 2400 baud. Type a carriage return several times to get my attention. Registration requires that you supply your human's first name, last name, HUG ID number, and some sort of secret password (up to 16 characters). Alternatively, your human can call Jim Buszkiewicz at HUG, and register via voice connection at (616) 982-3837. Call today! All it takes is a computer, modem, and a phone call for your computer to talk to ours!"
MaC

But it's the shell function of BSE that I find of inestimable value in developing a catalog. When one reaches the point in WHAT.BAS where comments are to be added, no matter how carefully one prepares before starting MACAT, and no matter how carefully one contemplates the directory displayed by VFILER (or HFM), some file or more, long ignored, causes one to pause during the comment step. This is why my batch file affords the opportunity to look at the WHAT file immediately after it is compiled. BSE allows me to stop, call up any program on the disk being cataloged and run it, without exiting to DOS! So, when I run into a file I don't recognize or one for which I've cooked up a hokey comment, I have the option of exercising the file, exiting back to BSE, then adding or modifying comments while my mind is still fresh with what I've just learned. No relying on memory or pencil notes. A note of caution: the end-of-file character, ^Z, can be moved along with other bits of text. This happened to me once after my catalog was half complete. Subsequent processing with WHAT with ^Z midway through the current catalog resulted in loss of half the file. That is when I added provisions for automatic production of a backup file. BSE displays all control characters. I check very carefully for the location of the end-of-file character after editing.

```
CLS
VFILER %1
A:
COVER
A
ECHO ON
DIR %1 >TEMP
SORT <TEMP >TEMP.$$$
BASICA WHAT
BSE WHAT
IF EXITS !!! GOTO QUIT
DEL TEMP*.*
:QUIT
IF EXIST !!! DEL !!!
```

Figure 7

I use VFILER instead of HFM for three reasons; I can designate the drive containing the disk in the command line invoking VFILER, it takes up less disk space — 10k, as opposed for HFM (15k), and it requires fewer keystrokes to manage. HFM does have a feature not found in VFILER; capability to open a directory.

Purchase of HUG part number 885-3025-37 is recommended, if for no other reason, than to get HSORT.COM. The sorting program provided with MS-DOS 2.0 is much too slow for our purpose here. A file of a thousand lines takes about seven minutes for sorting by SORT.EXE, whereas HSORT.COM does it in less than a half minute. The capability of SORT.EXE to sort on any character location in a string is of no advantage to us here. Besides, with HSORT.COM you get all those other useful programs, including REPRINT.COM which I hoped (after I located its documentation with MACAT) could be used to dump a disk directory to a file. With that capability, MACAT could run under ZDOS!

Epilogue

This project began in earnest when I opened my package of eight inch disks looking for one to use, digging into the reasons Pat Swayne's ZPC did not work for me. I like lots of disk space when I debug so I can keep copies of changes that did or did not work. No more double-sided eight inch disks, only a couple of single-sided — and who needs a 250k disk! I poked around and found in my "active" library, a double-sided disk with what appeared to be largely junk. It was then I decided I'd had it, I wanted some means of listing all my files. I've liberated two eight inch and about a half dozen five inch disks by cataloging with MACAT, examining the indices for redundant and obsolete files, and purging the extras and junk. And I'm only half way through my library!

The author will dump the BASIC listings, compiled programs, and batch files to five or eight inch disks on receipt of a blank disk and five dollars to cover handling and postage.

```
10 '                              CPMDIR.BAS
20 '
30 ' CONVERTS A CPM DIRECTORY TO MS-DOS FORMAT FOR
   PROCESSING BY WHAT.BAS
40 '
50 '                              Louis M. St.Martin
60 '                              860 Hillcrest Drive
70 '                               Pomona, CA  91768
80 '                                 714/522-3248
90 '
100 ' ************************ Count lines in directory
110 '
120 OPEN "I",#1,"TEMP.CPM"
130     IF EOF(1) THEN CLOSE#1 : GOTO 200
140             INPUT#1, LLINE$
150     COUNT = COUNT + 1
160     GOTO 130
170 '
180 ' ************** Transfer directory to temporary file
190 '
200 DIM FILENAME$( 4*COUNT + 1)
210 I = 1
220 OPEN "I",#1,"TEMP.CPM"
230     IF EOF(1) THEN CLOSE#1 . GOTO 450
240        INPUT#1,LLINE$
250             IF LEFT$(LLINE$,1) = " " THEN 260 ELSE 350
260     FILENAME$(I) = MID$(LLINE$,2,13)
270          FILENAME$(I) = LEFT$(FILENAME$(I),9) +
             RIGHT$(FILENAME$(I),3)
280     FILENAME$(I+1) = MID$(LLINE$,18,13)
290          FILENAME$(I+1) = LEFT$(FILENAME$(I+1),9)
             + RIGHT$(FILENAME$(I+1),3)
300     FILENAME$(I+2) = MID$(LLINE$,34,13)
310          FILENAME$(I+2) = LEFT$(FILENAME$(I+2),9)
             + RIGHT$(FILENAME$(I+2),3)
320     FILENAME$(I+3) = MID$(LLINE$,50,13)
330          FILENAME$(I+3) = LEFT$(FILENAME$(I+3),9)
             + RIGHT$(FILENAME$(I+3),3)
340 GOTO 430
350     FILENAME$(I) = MID$(LLINE$,1,13)
360           FILENAME$(I) = LEFT$(FILENAME$(I),9)
              + RIGHT$(FILENAME$(I),3)
370     FILENAME$(I+1) = MID$(LLINE$,17,13)
380          FILENAME$(I+1) = LEFT$(FILENAME$(I+1),9)
             + RIGHT$(FILENAME$(I+1),3)
390     FILENAME$(I+2) = MID$(LLINE$,33,13)
400          FILENAME$(I+2) = LEFT$(FILENAME$(I+2),9)
             + RIGHT$(FILENAME$(I+2),3)
410     FILENAME$(I+3) = MID$(LLINE$,49,13)
420          FILENAME$(I+3) = LEFT$(FILENAME$(I+3),9)
             + RIGHT$(FILENAME$(I+3),3)
430 I = I + 4
440 GOTO 230
450 OPEN "O",#2,"TEMP"
460     FOR K = 1 TO 5   PRINT#2, " " . NEXT K
470     FOR J = 13 TO I
480             IF FILENAME$(J) = "" THEN 520
490             L = LEN(FILENAME$(J))   IF L = 13 THEN 510
500             FILENAME$(J) = FILENAME$(J) + STRING$(13-L,46)
510             PRINT#2, FILENAME$(J)
520     NEXT J
530 CLOSE#2
540 SYSTEM
```

Listing 3

On The Leading Edge

A Look At MS-DOS Version 3.1

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

One of the letters that I received recently asks about the differences between MS-DOS version 2 and 3. In particular, he points out that I have been mentioning both in my column and wonders why.

The key reason for mentioning both is that I have found that most Heath/Zenith users still use some release of version 2 MS-DOS for both the '100 and the PC series. In addition, most of the items related to operating systems apply to both version 2 and version 3. This article is a specific exception to that since I will talk about some of the new commands and features that are now available in version 3 for both the H/Z-100 and the PC series computers.

As you might guess, I will also be speaking on this topic at this year's HUG Conference. This article is particularly for those of you who are unable to attend the Conference. For those of you able to attend the Conference, there may be a short "quiz" about this article at the beginning of each presentation.

An Overview Of MS-DOS 3.1

Unlike the major changes that were obvious in version 2, version 3.1 has many subtle changes plus a limited number of new commands and some modest changes to existing commands. Perhaps the most subtle and least obvious change to most users is the support of networking and file sharing. But I'm getting ahead of the discussion. Let's take a quick look at some of the new commands and features that are of interest to most users.

New commands in version 3.1 are: ATTRIB, JOIN, LABEL, SHARE, and SUBST. MS-DOS 3.1 for the PC series also includes GRAFTABL, KEYBXXXX, and SELECT. We will take a look at the uses of each command later on in this article.

The Configuration File (CONFIG.SYS) also has some new features: COUNTRY, FCBS, and LASTDRIVE. The SWITCHAR and AVAILDEV commands, which were added in version 2, are not available in version 3.1.

General command enhancements include changes to FC, FORMAT, LIB, LINK, and PRINT. Zenith has also modified some existing commands to make them closer to their PC-DOS counterparts. The biggest change here is that the ASSIGN command, previously used to assign hard disk partitions, has been changed to the ASGNPART command. That has been done so that the "new" ASSIGN (the old MAP) command can perform the same function as the IBM PC-DOS ASSIGN command.

For the same reason, the old MDISK has been renamed to VDISK for use in creating a virtual (i.e. memory) disk with the CONFIG.SYS file. And finally, the PC series BACKUP and RESTORE commands have been changed to provide compatibility with their PC-DOS counterparts.

The Most Needed Commands

I have always wondered why Microsoft did not provide some of the "obvious" commands in the initial release of MS-DOS. I say obvious because old-time CP/Mers will recognize that MS-DOS has not previously had any real equivalent to the CP/M STAT command. For those of you unfamiliar with CP/M, the STAT command provides a number of quite powerful features. The capability to dynamically change logical system devices (like CTTY but better) and the capability to "mark" (and unmark) files as Read-Only are specific features that I miss in MS-DOS. By the time version 2 came out, I had already written a program to change the file attributes.

I was also more than puzzled why Microsoft did not provide a command to allow the user to add/delete/change the disk volume label that was added to the FORMAT program in version 2. That seemed to be such an obvious program. Because of the experimenting that I do, I also wrote a simple program to do that.

The ATTRIB and LABEL commands provide these capabilities in MS-DOS version 3.1.

The ATTRIB Command

The ATTRIB command allows you to change the file attributes, but in a very limited way. Normal MS-DOS files have the Read/Write attribute so that you can add, change or delete a file. If you have ever mistakenly erased an important file with a DEL *.*, you will recognize that it might be very useful to mark a file as Read-

Only. In that case, the DEL *.* command will NOT erase the file because of the Read-Only attribute. In order to erase the file, you must first change it back to Read/Write.

In general terms, the command syntax for ATTRIB is:

```
ATTRIB [+R] [d:][path]afn
```

The ambiguous file name (afn) may include a specific file name or include the use of the asterisk (*) and question mark (?) wildcard characters in the file name specification. If the command is input with no optional parameters, the status of each file matching the afn description will be displayed.

The normal file attribute is Read/Write or minus the Read-Only attribute (i.e. -R). You can change one or more files to Read-Only by adding (+R) the attribute.

While all of this may seem nice, I still don't understand why Microsoft did not provide the capability to change other file attributes like STAT. It is sometimes nice to be able to have a hidden file (from the DIR command). MS-DOS also has something called a system file. The BIOS (IO.SYS) and the System Kernel (MSDOS.SYS) are defined as system and hidden files. If you want a command to "unhide" these files, you need to use the FLAGS command provided in the Programmer's Utility Pack. While that is a nice utility provided by Zenith, I still think that Microsoft should have provided that with the operating system.

The LABEL Command

The LABEL command provides a useful feature if you ever want to change the disk volume label created with FORMAT. If you use the volume label, and FORMAT a box of disks at one time, you never know just what label you will want or need. And so, many disks do not have a volume label. I wrote a simple program to modify the label several years ago; now a similar utility is part of MS-DOS.

The general form of the LABEL command is:

```
LABEL [d:][volume-label]
```

The volume-label is the standard 11 character name that may be added to the disk. If the optional volume-label is not input in the command line, the command will display the prompt: "Volume Label (11 characters, ENTER for none)?".

A label can be added or changed by simply typing in the new label. If no label is input, the current label (if one exists) is deleted.

The SHARE Command

The purpose of the SHARE command is to load operating system support for file sharing on a network. The general command form is:

```
SHARE [/F:file-space][/L.locks]
```

In addition to loading file sharing, SHARE also allows the network manager to modify the file-space and locks from the default values. Since most users will not need this command, I will not go into any detail as to the file-space and locks functions.

The Really New Commands

I think that the most interesting of the new MS-DOS 3.1 features are the JOIN and SUBST commands. Each provides a significant new capability that can be used to provide some new and exciting possibilities for users. And in the case of the Z-100 WordStar version 3.30, I couldn't get the program to run satisfactorily without using the SUBST command.

The JOIN Command

The JOIN command effectively replaces (joins) a specified disk drive name with a single subdirectory name on the current or specified disk drive. Fine, but what does THAT mean? In a practical sense, it means that you can effectively create a "single" disk drive through which all of a DIFFERENT disk drive's subdirectories are available. Terrific — that's as clear as mud!

Well, let's take a look at the command syntax, and then we'll take a look at an example. The general form for the JOIN command is:

```
JOIN new-d. [old-d:]\new-dir
```

The new drive (new-d) is required input and is sometimes called the joined drive. It must be a valid and active drive in your system. The new directory (new-dir) is usually created by the JOIN command although any directory can be used. But enough of the mud, let's take a look at an example.

Let's assume that we are looking at a PC series system with two floppy drives and a hard disk. Boot the system from the hard disk, and you will get the standard C> prompt. Assume that your hard disk contains ALL of your software (both system and application). Your floppy disks are used only for data. Insert data disks into both drives and enter the following commands:

```
JOIN A: \DRIVEA
JOIN B  \DRIVEB
```

The JOIN command will create the subdirectory entries, DRIVEA and DRIVEB, on the current drive C. In order to "change" from the current drive C, you MUST now use the CHDIR (CD) command to change directories (in this case, disk drives). For example, if you enter CHDIR \DRIVEA, MS-DOS will then "log-on" to drive A, and you can access any of the files and/or subdirectories on drive A.

It is really interesting to play with this command and see the different drive lights turn on when a CHDIR command is entered. Since my H-100 system has two 5.25 inch drives, two 8 inch drives, and a hard disk; I have joined all of the drives except the boot drive (E) to watch the results.

That sounds like a nice toy, but what good is it? Well, the most practical use that I have found so far is trying to find a file on a floppy disk and/or hard disk partition using the Zenith SEARCH command. I want to look through my programming set of 8 inch disks to find the program FF.ASM which sends a formfeed to my printer. I use the JOIN command to "connect" my 8 inch drives to the hard disk E drive. Then I use the following command: SEARCH FF.ASM. If the file doesn't show up, I replace the disks in the 8 inch drives and rerun the command. The process is easy since all I really have to do is hit F3 followed by a RETURN after I change the disks. There are obviously many possibilities for practical use of this command. If you think up some interesting uses of the JOIN command, let me know. You might see your name in one of my next columns as the originator of the idea.

Once the drive(s) are JOINed, there must be a way to "disconnect" the joined drives from the system. And there is. Following our example, we can disconnect drives A and B with the following commands:

```
JOIN A: /D
JOIN B  /D
```

Note that the space between the drive name and the /D parameter is required.

The SUBST Command

The SUBST command provides some really powerful capabilities, and allows you to substitute a disk drive name for a complete directory path. While that may not sound like much, it allows you to use "older" software that does not understand how to deal with subdirectories -- WordStar for example. Since WordStar does understand how to work with disk drives, the SUBST command effectively "renames" the path to a disk drive name. Then you can "change" directories by simply entering a drive name — almost the direct opposite of the JOIN command. The general form of the command is:

```
SUBST new-d  [old-d.]path
```

The new drive (new-d) is required input and is sometimes called the substituted or SUBSTed drive. The path is the directory path that will be "renamed" to the drive name. And you can include an optional drive name (old-d) if the path is not on the current drive.

When I started working with MS-DOS 3.1 on my H-100, I found that WordStar 3.30 does not like this new DOS version. Since WordStar does not understand subdirectories, I used to keep the two overlay files in every subdirectory that required editing features more advanced than EDLIN. That was obviously a lot of wasted space because of all the duplication.

When I tried to use that configuration with the Z-100 MS-DOS version 3.1, I found that all kinds of strange things would happen. Sometimes the system would freeze, sometimes I would get color, and one time I even got the WILD INTERRUPT message. Although the Z-100 version 3.30 of WordStar (and MailMerge) is known to be buggy, this was a totally new problem.

The SUBST command helps correct the problem. Let's say that I want to do some word processing in a subdirectory called \Z-MSDOS3. The first thing I did was "rename" the subdirectory with: SUBST A: E:\Z-MSDOS3. Then I found that I had to move all of the WordStar files to the root directory on my hard disk for everything to work properly. I fired up WordStar on drive E, and then used the "L" command to log-on to drive A. After that, I was able to use WordStar with no new problems.

One of the facts of life for WordStar fans is that all 16-bit versions of WordStar are nothing more than "translated" versions of the 8-bit software. I've known that for a number of years since it was obvious that there was no 64K memory limit (which required overlay files) in the MS-DOS systems like there is in CP/M. Why is that obvious? The biggest reason is that there is absolutely no reason why the WS command can't be a SINGLE file in MS-DOS. The presence of the two overlay files plus some fooling around with DEBUG convinced me that the code was virtually identical. Since then, I have talked to an expert who knows about the WordStar source code, and he confirmed my suspicions.

I have only used WordStar as an example since it was the obvious program that does not deal with subdirectories. As a personal matter, I have given up on the extensive use of the Z-100 WordStar because of its terrible performance and bugs. Oddly enough, the performance problems and bugs do not exist in the CP/M version 3.30 that I use quite a bit. As a result, I use WatchWord and Resident Speller for any extensive work under the Z-100 MS-DOS. Maybe one of these days we'll be able to talk Steve Robbins into writing a "degraded" version of WatchWord that will run on the PC series. In case you came in late, you can find additional information about WatchWord in the July 1985 REMark under the title of "Watching WatchWord Work Words and Other Goodies". And by the way, the latest versions of WatchWord and Resident Speller continue to be highly recommended.

The GRAFTABL Command (PC Series Only)

In order to continue to provide nearly total PC compatibility, some new commands have been added to the MS-DOS for the PC series computers. If you have read my articles before, I have usually referred to that as the Z-150 MS-DOS. Because the Z-150 MS-DOS is also used on the Z-200 computers, I will refer to it as the PC series MS-DOS in this and future articles.

The GRAFTABL command is virtually identical to its IBM PC-DOS 3.0 counterpart of the same name. It loads a table of additional character data (ASCII characters 128-255) into memory for use in the graphics mode with the color/graphics adapter.

The command syntax is just: GRAFTABL. No complicated options or parameters to remember.

The KEYBXXXX Command (PC Series Only)

The KEYBXXXX program set loads a keyboard layout program that overrides the one resident in the ROM BIOS. These programs are used to support the keyboard layout of various characters for foreign languages. The general form of the command is just the KEYBXXXX where the value XXXX is dependent on which country and language required for the keyboard.

It is important to note that there is a relationship between the KEYBXXXX programs and the GRAFTABL command. KEYBXXXX allows easy keyboard access to the foreign characters and GRAFTABL allows you to DISPLAY them.

The SELECT Command (PC Series Only)

The SELECT command is the easiest way to implement a foreign language keyboard. It automatically creates an AUTOEXEC.BAT file with a KEYBXXXX command in it, and it also creates a CONFIG.SYS file with the specified COUNTRY= parameter. Unlike the IBM PC-DOS documentation, the Zenith documentation specifically states that you should use this command to create backup copies of the distribution disks for the selected foreign language implementation. The PC-DOS documentation says nothing. Why is that important? Because the SELECT command creates (NOT updates) the AUTOEXEC.BAT and CONFIG.SYS files. Even if those files already exist on the disk, they will be OVERWRITTEN with the "new" files. Although I suppose that Microsoft had some rationale for writing the program to do that, I think that the program should update either or both files if they exist. Admittedly, that's a little more difficult to program, but I still think it should have been done that way.

In any case, the general form of the SELECT command is:

```
SELECT country-code keyboard-code
```

Although there are a number of international readers of REMark, I won't list the combinations for the country-code and the keyboard-code since Zenith has 19 in the latest MS-DOS version.

CONFIG.SYS Commands

Perhaps the most significant change in the CONFIG.SYS commands is that Microsoft has deleted the SWITCHAR and AVAILDEV commands that were only available in Version 2. No big loss — I never found a real use for either of those commands.

The COUNTRY command allows you to specify a numeric value which is assigned to each supported country for the appropriate

TIME and DATE format. For example, some countries use a DATE format of dd/mm/yyyy instead of the default US format which is mm/dd/yyyy. But my testing of this and other Version 3 commands for my new FlipFast book has revealed a significant documentation discrepancy at this point. Every bit of documentation that I have found says that the COUNTRY command changes the date format, time format, currency symbol, and the decimal separator. Both IBM and Zenith documentation are consistent in that respect. The interesting part of this is that I have never been able to get the currency symbol ($ for US currency of course) to change on the keyboard as a result of the COUNTRY command.

I have been able to change the date and time format items as a result of the COUNTRY command, but they are not the same changes indicated in the documentation. Moreover, the currency symbol NEVER changes as a result of the COUNTRY command — that requires a KEYBUK command. But let's look at what the COUNTRY command really does change.

It does change the DATE format from the default mm/dd/yyyy to the dd/mm/yyyy which is used in the United Kingdom (044). It also changes the DATE separator from the default slash (/) to a hyphen (-) or a period (.). Germany (049) and Switzerland (041) use a DATE format of dd.mm.yyyy for example.

The TIME format also changes, but not exactly in the way specified in the documentation. The default is hh:mm:ss.cc in the hours, minutes, seconds, and hundredths of a second (cc) based on a 24-hour clock. If you use the COUNTRY code for Denmark (045), you will see the time displayed as hh.mm.ss,cc. The default colon (:) has been changed to a period and the decimal separator has been changed to a comma (,), but my testing indicates that the format always remains as hh:mm:ss.cc regardless of the actual separators used in a country.

I have concluded that both IBM and Zenith documentation are wrong as to the items changed by the COUNTRY command. My guess is that the basic Microsoft documentation provided to Original Equipment Manufacturers (OEM's) is either misleading or wrong also. In any case, here is a summary of the items I found that were actually changed by the COUNTRY command:

- The DATE format (e.g. mm/dd/yyyy to dd/mm/yyyy)
- The DATE separator (e.g. slash (/) to period (.))
- The TIME separator (e.g. colon (:) to hyphen (-))
- The TIME decimal separator (e.g. period (.) to comma (,))

The exact details of each DATE and TIME format for each country are contained in my new set of FlipFast books.

The FCBS command specifies the number of File Control Blocks that can be open at any given time. The general command form is:

```
FCBS=number-open,number-protected
```

The number-open parameter is the number of FCBs that can be open at any given time. The number-protected is the number of FCBs that are protected from being automatically closed if a program tries to open more files than specified in number-open. I will not discuss the reasons for all of this except to say that, if you need this parameter, you should find the required information in your applications documentation.

The LASTDRIVE command sets the alphabetic drive identifier of all drives (including virtual drives set up with VDISK.SYS) that can be accessed by the system. The general form of the command is:

```
LASTDRIVE=d
```

Valid drive identifiers (d) can be in the range from A-Z in an MS-DOS system. The default depends on the block device drivers set up in the BIOS. This command is not normally needed unless you are using a number of virtual disks in your system or decide to ASSIGN/SUBST a nonexistent physical disk drive. On my H-100 for example, I have two 5.25" drives, two 8" drives, and a hard disk with four partitions for drives A-H. If I wanted to use the drive letter I for the ASSIGN or SUBST command, I would need to include the LASTDRIVE=I in the CONFIG.SYS file.

Enhanced MS-DOS Commands

The FC command for both Z-100 and PC series MS-DOS has been significantly enhanced to add a number of new parameters plus a change in some of the formats of those parameters.

The FORMAT command for the PC series has been changed to allow for formatting of high density disks for the Z-200 series plus some other parameter additions.

The FORMAT command for the Z-100 has been changed to default to 9 sectors per track formatting instead of 8.

The LIB command has a new PAGESIZE (/P) parameter and also allows appending a new library to an existing library.

The LINK command has six new parameters, supports memory overlays, and allows up to 16 directory paths for the library file. LINK has also been expanded to allow the linking of up to one megabyte (900K in version 2) files.

The PRINT command has six new parameters that are primarily associated with background printing from the spooler. All parameters that allowed numeric input (e.g. /C:n for the number of copies to be printed) now require a colon (:) to separate the parameter from the numeric value.

The memory disk that can be configured in the CONFIG.SYS file is now called VDISK.SYS in both systems. It is fully documented and can be configured in extended memory (1 megabyte or more) or the standard low memory (less than 1 megabyte).

The MAP command available in version 2 has been renamed ASSIGN with a command syntax that is identical to the PC-DOS ASSIGN command. The version 2 ASSIGN command that was used to assign hard disk partitions has been renamed the ASGNPART command with no change in command syntax or parameters.

The CONFIGUR command for the PC series MS-DOS version 3.1 has the capability to "add" user defined devices to the CONFIGUR menu plus the capability to perform automatic hard disk partition assignment when more than one hard disk is available.

And finally, PC series BACKUP and RESTORE commands have been modified to make them compatible with their PC-DOS counterparts.

I have not tried to provide all of the detailed changes to every command since it is included in the Zenith documentation, as well as my new set of FlipFast books.

Correcting Some Wrong Reports About MS-DOS 3.1

As I write this, I have consistently seen a significant amount of "misinformation" that has been reported by one major business user about MS-DOS 3.1 and hard disk support for the Z-200 system. In some cases, the problem has been the lack of user under-

standing of MS-DOS (including PC-DOS) hard disk support; in others, the reports are simply wrong since they state that some "problems" are bugs when the fact is that they are not.

Perhaps the most disturbing thing about these reports is the fact that current hard disk backups were not available. Although most hard disk users seem to "know" that they should take frequent and periodic hard disk backups, many fail to do so. This user failed to have a current backup and apparently lost quite a bit of business data. Even worse, it appears that Zenith MS-DOS is being "blamed" for this data loss problem because of what has been reported as a bug. I find it quite remarkable that an experienced business user of ANY hard disk computer system could fail to have current backups of hard disk data. Since he also sells computer hardware, including tape backup systems, it is difficult to imagine why he didn't use one on his business system.

Lest you think that I am not sympathetic to the loss of data on a hard disk system, you may recall that I also had a recent experience with a complete Winchester wipeout due to an early ROM version in the Easy PC (See June 1986 REMark). As I've said before in this column, it is a critical fact of using a hard disk that frequent and periodic (even daily) backups be taken.

As a result of these misunderstandings, user errors, and just plain wrong information, I am currently in the process of researching an article on "MS-DOS and High-Capacity Hard Disks". While this article will specifically talk about the Zenith MS-DOS and its use on the Z-200 series, we will also explore the whys and wherefores of the historic Microsoft 32 megabyte hard disk limitation that has always applied to all versions of MS-DOS and PC-DOS. Then we will look at how Zenith, by using some really clever programming techniques and working with Microsoft, was able to enhance their MS-DOS to overcome the physical 32 megabyte limit. Current versions of MS-DOS now allow you to use 32 megabytes in a single partition or a whopping 128 megabytes for each physical drive if all four partitions are used.

Before everyone starts looking for a 127.999 megabyte drive however, there are a few words of caution. You MUST use MS-DOS 3.10 with IO.SYS version 3.04 or later. Use the VER command to determine your IO.SYS version. If it's 3.03 or below, call Zenith Software Consultation to get an update. The reason is that the Zenith enhancement for high-capacity drives was not in the earlier versions of the BIOS (i.e. IO.SYS), aside from the fact that Heath and Zenith did not sell drives larger than 32 megabytes until after the updated IO.SYS was being shipped with the Z-200s. Note that this ONLY applies if you intend to use a high-capacity drive larger than 32 megabytes.

More Wrong Information On The SHIP Command

There is at least one incorrect report floating around that says that the IBM SHIPDISK and Zenith SHIP commands can destroy data on a hard disk. This report also states that both Zenith and IBM have confirmed this. I don't know about the IBM command, but I have checked with Zenith, and they don't know anything about the problem or the alleged "confirmation". Indeed, one highly placed and totally reliable Zenith source tells me that Zenith doesn't know any possible way that SHIP could destroy data, let alone the fact that Zenith supposedly "confirmed" this. If you have any personal experience with this problem, please write or call Software Consultation immediately with the details of the problem.

What I find particularly puzzling about this report is that I don't see any normal way that SHIP could possibly cause loss of data. And you don't need to be an MS-DOS expert to understand how SHIP works. Running the Z-100 PREP command basically establishes the parking cylinder (Winchester head landing zone) on the center of the hard disk. Parking cylinders for the PC series computers are basically part of the ROM table like they are on the IBM computers. If the hard disk (or computer) is accidentally bumped, the head will "bounce" on a portion of the hard disk that does not contain any data.

From a programming perspective, the SHIP program is quite simple to describe. First, SHIP interrogates the system to determine the location of the parking cylinder. Then SHIP tells the hard disk controller to move the Winchester heads to that specific parking cylinder. If you are interested in the technical details, the standard disk controller SEEK command is used to position the disk heads. Since the controller SEEK command does nothing but move the Winchester heads, it is more than a little difficult to imagine how data loss could possibly occur during execution of the Zenith SHIP command. When the system is subsequently powered-off, the Winchester heads "float" down to land on the parking cylinder.

As you can see, it doesn't appear likely that it is technically possible for the Zenith SHIP command to cause a loss of hard disk data. SHIP does not write to the hard disk ever. It only moves the heads. But I have been in data processing and programming for too many years to say that anything is impossible.

I can, for example, envision a situation that could cause a loss of data during (but not the result of) the use of the SHIP command. Power surges and transients can cause strange problems. If a power surge happened at EXACTLY the right time, say during the execution of the SHIP command (or any command for that matter), it is conceivable that the power surge could "ripple" through the system and cause the disk controller to issue a write command. Data might be destroyed (overwritten with garbage) as a result. Although I believe that the power supplies for the Heath and Zenith computers are extremely well designed to cope with normally expected power supply fluctuations, they are not designed to handle one thousand percent changes. It is important to note that this is NOT a hardware or a software problem. But what can you do if you want to protect your system against this kind of problem?

My solution was to buy an Uninterruptible Power Supply (UPS) for my system. I use the Minuteman 500 UPS that is available from Para Systems that I discussed in my December 1985 column. It provides 500 watts of standby power that will take care of my system for about 15 minutes — more than enough time to save data and power-off the system in the event of a long power outage. In addition, the Minuteman 500 also provides surge protection, Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) filtering of the power source for even more protection. Surge protectors are a less expensive way to cope with power surges, if you don't have to worry about the complete loss of power.

The HUG Conference

For those of you able to attend this year's HUG Conference in Chicago, I will cover all of these new commands and changes in more detail with some information on Windows. In addition, I will talk about some of the new surprises in a couple of areas. Hope to see you there.

In The International Mail . . .

I guess I always "knew" that HUG was an international organization, but I have recently received an unusual amount of inter-

national mail. Keith Watson (South Australia) wrote to ask me if I knew of any REASONABLY priced backup systems and/or software for the Z-100. I haven't found any yet. HEELLLPPPP! Does anyone sell or know of a reasonably priced tape backup system (or special backup software) for the Z-100? If so, let me know about it. Although several tape backup systems are available for the PC series (including the H/Z-200s and the Zenith compatible IBM AT) for under $600, similar systems for the Z-100 cost in excess of $2,000! Totally ridiculous — that's more than the original cost of the computer! If you know of any vendor that has a good tape backup system for the Z-100, let me know who it is and how much. If you are a vendor, write to me so that we can arrange for an evaluation of your backup system and/or software in REMark. And by the way, my estimate of a reasonable price for tape backup hardware is on the order of $1,000 or less.

I checked with HUG and found that our international membership is rapidly approaching 1,400. It is perhaps not surprising that I have received two letters from England and one from Germany.

In The Future

I haven't forgotten that we need to finish Part 2 of the article on MS-DOS Command Piping and I/O Redirection. That's for next month. Part 1 on that topic appeared in the May issue.

If things go as planned, we will take a look at MS-DOS and High-Capacity Hard Disks in the October issue. We'll take a look at some of the history of hard disks and some of the details of hard disk use. I hope that the article will dispel some unfounded rumors of "bugs" in MS-DOS. And we will completely "blow away" some suggestions that the '241 does not provide the same type of hard disk support as the IBM AT by looking at a list of the supported hard disks for each machine. In my attempt to provide HUG members with the latest and most accurate information about hard disks, MS-DOS, and the Z-200 computers, I will also be doing considerable testing on all of this on my own HS-241 equipped with a ZD-400 40MB hard disk.

Although I have a couple of other articles in mind for future issues, I am always interested in your suggestions. If you have a particular subject that you think would be an interesting article, please let me know. I would particularly like to invite suggestions from our international members since I suspect that information may be more difficult to find. As always, please enclose a stamped, self-addressed envelope if you would like a personal reply.

Products Discussed

Software

MS-DOS Version 3	
Z-100 only (OS-63-30)	$ 150.00
PC only (OS-63-31)	150.00
Programmer's Utility Pack (CB-3163-30)	150.00
Microsoft Windows	
PC only (MS-5063-30)	99.00

Hardware

Advanced Personal Computer (HS-241)	$2899.00
Monochrome/Color Video Card (Z-409)	239.00
20MB Winchester (ZD-200)	1499.00
40MB Winchester (ZD-400)	2499.00
H-100 Desktop Computer (HS-1108-41)	1599.00
Uninterruptible Power Supply	
200 watt (PP-111)	349.95
400 watt (PP-115)	549.95
1000 watt (PP-120)	1195.00

Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-7057 (Heath Catalog orders only)

Software Consultation
Zenith Data Systems Corporation
Hilltop Road
St. Joseph, MI 49085
(616) 982-3884 (Application Software only)
(616) 982-3860 (Operating Systems/Languages/Utilities 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

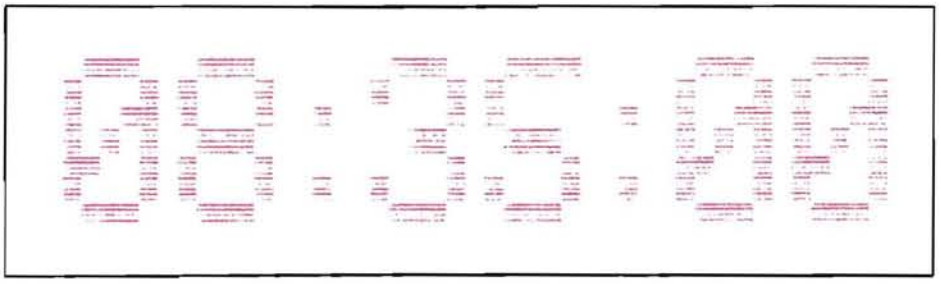
WatchWord	$ 100.00
Resident Speller (Z-100 MS-DOS)	100.00
Resident Speller (Z-150/PC)	99.00

S & K Technology
4610 Spotted Oak Woods
San Antonio, TX 78249
(512) 492-3384

*

Clock Watcher's Delight Revisited

Pat Swayne
HUG Software Engineer



Way back in the August 1984 issue of REMark, I described a program that caused the time to be displayed in the upper right corner of your screen, even while you ran other programs. In this article, I will present a new screen clock program for H/Z-100 PC and H/Z-200 PC series computers that is an improvement over the original in two important ways.

As I stated in the original article, one of the problems involved in writing a screen clock program for a PC type computer is that the system time cannot be easily accessed during the timer interrupt that runs the program. Because of this, the original program kept time internally, so it was possible for the time on the screen to be different from the system time (shown by the MS-DOS TIME command). Also, the original program updated the display every 18 timer interrupts, and since there are actually about 18.2 interrupts per second, the seconds were ticked off slightly faster than normal, and the clock had to halt for two seconds every so often to correct itself.

I have written a new screen clock program for PC type computers that derives the time displayed on the screen from the system time, and that ticks off the seconds accurately. The assembly source code for the new program, called SCLK.COM, follows the text of this article. If you do not have an assembler, you can type in and run the following BASIC program to create SCLK .COM.

```
10 REM THIS PROGRAM CREATES SCLK.COM
20 DEFINT A-I:OPEN "O",1,"SCLK.COM
30 S=0:S1 = 36088 :FOR I=1 TO 417
40 READ B:S=S+B:PRINT #1,CHR$(B);
50 NEXT I:IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 233,89,1,1,0,0,0,0,0
80 DATA 0,0,0,0,0,0,1,2,3,1
90 DATA 0,2,2,32,0,0,58,0,0,58
100 DATA 0,0,67,75,80,46,128,54,4,1
110 DATA 1,117,8,46,128,62,3,1,0,117
120 DATA 6,88,234,0,0,0,0,46,198,6
130 DATA 3,1,0,46,137,38,5,1,46,140
140 DATA 22,7,1,250,140,200,142,208,188,0
150 DATA 1,251,83,81,82,86,87,30,6,184
160 DATA 64,0,142,216,161,108,0,139,22,110
170 DATA 0,139,216,139,202,209,224,209,210,209
180 DATA 224,209,210,3,195,19,209,14,31,187
190 DATA 5,0,185,35,0,190,255,255,70,43
200 DATA 193,27,211,115,249,3,193,19,211,137
210 DATA 54,9,1,185,86,21,247,241,162,10
220 DATA 1,139,194,177,91,246,241,58,6,11
230 DATA 1,116,114,252,162,11,1,60,60,117
240 DATA 2,254,200,232,129,0,163,30,1,160
250 DATA 10,1,232,120,0,163,27,1,160,9
260 DATA 1,232,111,0,163,24,1,180,15,205
270 DATA 16,83,187,15,1,215,162,14,1,91
280 DATA 180,3,205,16,137,22,12,1,182,0
290 DATA 178,31,128,62,14,1,2,114,2,178
300 DATA 71,180,2,205,16,185,9,0,179,7
310 DATA 128,62,14,1,0,116,9,128,62,14
320 DATA 1,2,116,2,179,2,190,23,1,172
330 DATA 81,86,232,50,0,94,89,226,246,180
340 DATA 2,139,22,12,1,205,16,7,31,95
350 DATA 94,90,89,91,250,46,142,22,7,1
360 DATA 46,139,38,5,1,251,46,198,6,3
370 DATA 1,1,233,2,255,180,0,183,10,246
380 DATA 247,5,48,48,195,180,9,185,1,0
390 DATA 205,16,180,3,205,16,254,194,180,2
400 DATA 205,16,195,128,252,242,116,5,234,0
410 DATA 0,0,0,46,162,3,1,207,30,51
420 DATA 192,142,216,190,112,0,196,60,38,129
430 DATA 125,254,67,75,116,49,250,199,4,34
440 DATA 1,140,76,2,31,137,62,53,1,140
450 DATA 6,55,1,251,30,51,192,142,216,190
460 DATA 132,0,196,60,199,4,77,2,140,76
470 DATA 2,31,137,62,83,2,140,6,85,2
480 DATA 186,92,2,205,39,205,32
```

After you create SCLK.COM, place a copy of it on your system disk and enter

```
SCLK
```

at the system prompt, and hit RETURN. The current time will be displayed in the upper right corner of your screen, and will be updated each second. The program will remain resident in memory until you reboot your computer. Since you may want to turn off the clock display without rebooting, I have written a CLK .COM that can turn the display off and back on. Use the assembly source code at the end of this article, or the following BASIC program to create CLK.COM.

```
10 REM THIS PROGRAM CREATES CLK.COM
20 DEFINT A-I:OPEN "O",1,"CLK.COM
```

```
30 S=0:S1 = 2143  FOR I=1 TO 22
40 READ B:S=S+B:PRINT #1,CHR$(B),
50 NEXT I.IF S<>S1 THEN PRINT "TYPING ERROR!":END
60 CLOSE #1:LOCATE 23,1:PRINT "DONE!":SYSTEM
70 DATA 187,1,242,160,93,0,60,32,116,6
80 DATA 60,79,117,2,179,0,139,195,205,33
90 DATA 205,32
```

Place CLK.COM on your system disk, and enter

```
CLK OFF
```

at the system prompt, and hit RETURN to turn the clock off, or enter

```
CLK
```

and hit RETURN to turn it back on.

How It Works

On a PC type computer, a counter is maintained at a fixed location in memory that is incremented once for each timer interrupt. These interrupts occur approximately 18.2 times per second, or, more accurately, 65543 times per hour. The counter of these interrupts, sometimes called the tick counter, is preset by MS-DOS whenever you set the system time with the TIME command so that it always contains the number of interrupts, or ticks, since midnight (regardless of when you turned on your computer). It is, therefore, possible to calculate the time of day from the value in the tick counter, and that is what MS-DOS does when you use the TIME command to view the time.

One way to calculate the time from the tick counter value is to first divide that value by 65543, to obtain the current hour. Then divide the remainder from the first division by 1092.4 (one 60th of 65543) to obtain the current minute. Finally, divide the remainder of the second division by 18.207 (one 60th of 1092.4) to obtain the current second. That's easy to do on a calculator, but difficult when you are working the basic instruction set of a microprocessor, and you don't want to add a 20k floating point package to a .5k clock program. The basic instruction set of the microprocessor used in PC computers cannot work with decimal fractions, and it cannot even divide by a number more than 65535.

One way to make the above calculations easier is to multiply the tick counter value by 5 and then use 5 times the above constants in the divisions. That gets rid of the decimal fractions (if you assume 18.2 ticks per second), but you still have the problem of dividing by a number larger than 65535. To those of us who came over from 8-bit processors, that is no problem, because the 8-bit machines had no divide instruction at all, and we had to learn to divide by subtracting. That technique works on the new machines just as well, and can be used any time the built-in divide instruction cannot do the job.

The SCLK program gets the tick counter value every other timer interrupt (to reduce the drain on processor time) and multiplies it by 5. It uses rotation and addition to accomplish the multiplication (you cannot multiply by more than 65535 either), and then divides by the constants described above to calculate the current hour, minute, and second. My method for calculating the time from the tick counter value is not the same method used by MS-DOS (they go through a lot of rigmarole), but it is close, and the time on the screen clock is never more than about a half second off from the actual system time.

SCLK displays the time on the screen using only standard PC BIOS functions, and therefore, should work on any PC compatible machine. It will work during all modes of the color/graphics adapter, and also with a monochrome adapter. For more discussion of the display technique, see the original August 1984 "Clock Watcher's Delight" article.

The SCLK program produced by the BASIC program above produces a green display in the color text modes and a red or magenta display in the color graphics modes. For a white display, change lines 30 and 320 of the BASIC program to

```
30 S=0:S1 = 36093 .FOR I=1 TO 417

320 DATA 1,2,116,2,179,7,190,23,1,172
```

In the assembly source file, you can just change the color in the COLOR EQU GREEN statement to another color.

SCLK Source Listing

```
        TITLE   SCREEN CLOCK FOR MS-DOS
        PAGE    ,132
;       SCRNCLK V 2 -- SCREEN CLOCK FOR MS-DOS (Z-100 PC)
;
;       THIS PROGRAM PROVIDES A SCREEN CLOCK FOR MS-DOS THAT
;       IS DISPLAYED IN THE UPPER RIGHT CORNER OF THE SCREEN
;
;       THE COLOR OF THE DISPLAYED CLOCK IS DETERMINED BY THE
;       LABEL "COLOR", BELOW   IF YOU ARE IN A B/W MODE, THE
;       COLOR IS AUTOMATICALLY SET TO WHITE
;
;       BY P  SWAYNE, HUG SOFTWARE ENGINEER  23-MAY-86

;       DEFINITIONS

BLUE    EQU     001B            ;COLOR CONTROL BITS
GREEN   EQU     010B
CYAN    EQU     011B
RED     EQU     100B
MAGENTA EQU     101B
YELLOW  EQU     110B            ;(IBM CALLS IT BROWN)
WHITE   EQU     111B

COLOR   EQU     GREEN           ;CLOCK COLOR

ROW     EQU     0               ;PUT CLOCK ON TOP ROW

;       WE ASSUME 65543 TICKS PER HOUR   EVERYTHING IS
;       MULTIPLIED BY 5 FOR GREATER ACCURACY WITHIN THE
;       LIMITS OF INTEGER ARITHMETIC

TPHH    EQU     5               ;TICKS/HOUR * 5 HIGH
TPHL    EQU     23H             ;TICKS/HOUR * 5 LOW
TPM     EQU     5462            ;TICKS/MINUTE * 5
TPS     EQU     91              ;TICKS/SECOND * 5

SCRINT  EQU     10H             ;SCREEN CONTROL INTERRUPT
SCP     EQU     2               ;SET CURSOR POSITION
GCP     EQU     3               ;GET CURSOR POSITION
WCHAR   EQU     9               ;WRITE CHARACTER
GVS     EQU     15              ;GET VIDEO STATE
TIMEINT EQU     1CH*4           ;Z-150 TIMER INT  VECTOR
SYSINT  EQU     21H*4           ;MS-DOS SYSTEM INTERRUPT

JMPF    MACRO
        DB      0EAH            ;DEFINE FAR JUMP
        ENDM

SCLK    SEGMENT
        ASSUME  CS:SCLK,DS:SCLK,ES:SCLK,SS:SCLK
        ORG     6CH
TIMERL  LABEL   WORD            ;DEFINE TIMER LOW
        ORG     6EH
TIMERH  LABEL   WORD            ;DEFINE TIMER HIGH

        ORG     100H

START   JMP     SETUP           ;SET UP CLOCK INT , ETC
```

```
;       CLOCK DATA AREA

CLKFLG  DB      1               ,CLOCK ON/OFF FLAG
SKCNT   DB      0               ;SKIP COUNTER
SYSSTK  DW      0               ;SYSTEM STACK
SYSSTKS DW      0               ;SYSTEM STACK SEGMENT
HOU     DB      0               ;BINARY HOUR
MIN     DB      0               ,BINARY MINUTE
OLDSEC  DB      0               ;OLD SECOND
CPOS    DW      0               ;CURSOR POSITION
CMODE   DB      0               ;CURRENT VIDEO MODE
MODES   DB      0,1,2,3,1,0,2,2 ,DECODED MODES
TIMSTR  DB      ' '             ,SPACE BEFORE TIME
HOUR    DW      0               ;HOUR (ASCII)
        DB      ':'
MINUTE  DW      0               ;MINUTE
        DB      ' '
SECOND  DW      0               ;SECOND

;       PROCESS CLOCK INTERRUPTS HERE

        DB      'CK'            ,IDENTIFIER
MYTIME: PUSH    AX

;       FOR 1 CHECK PER 4 TICKS, USE THESE LINES

;       MOV     AL,CS:SKCNT     ;GET SKIP COUNT
;       INC     AL              ;ADD 1
;       AND     AL,3            ;MOD 4
;       MOV     CS:SKCNT,AL     ,SAVE NEW COUNT

;       FOR 1 CHECK PER 2 TICKS, USE THIS LINE

        XOR     CS:SKCNT,1      ;TIME TO CHECK?

        JNZ     TIMEXIT         ;NOT TIME TO CHECK
        CMP     CS:CLKFLG,0     ;CLOCK ENABLED?
        JNZ     CHTIME          ;IF SO, CHECK TIME
TIMEXIT:POP     AX
TIMEX   JMPF                    ,FAR JUMP INSTRUCTION
TIMADR  DW      0,0             ;SYSTEM TIMER ADDRESS

;       CHECK TIME, SEE IF A SECOND HAS PASSED

CHTIME: MOV     CS:CLKFLG,0     ;DISABLE OTHER CLOCK INTS.
        MOV     CS:SYSSTK,SP    ;ELSE, SAVE SYSTEM STACK
        MOV     CS:SYSSTKS,SS   ;SAVE SYSTEM STACK SEGMENT
        CLI                     ;KILL INTS  (ON AT ENTRY)
        MOV     AX,CS
        MOV     SS,AX           ;PUT STACK SEGMENT HERE
        MOV     SP,OFFSET START ;SET LOCAL STACK
        STI
        PUSH    BX              ,SAVE REGISTERS
        PUSH    CX
        PUSH    DX
        PUSH    SI
        PUSH    DI
        PUSH    DS
        PUSH    ES
        MOV     AX,40H          ;BIOS RAM SEGMENT
        MOV     DS,AX
        MOV     AX,TIMERL       ;GET TIMER LOW
        MOV     DX,TIMERH       ;AND TIMER HIGH
        MOV     BX,AX
        MOV     CX,DX           ;TIMER IN CX,BX
        SHL     AX,1            ;MPY TIMER * 5
        RCL     DX,1            ,* 2
        SHL     AX,1
        RCL     DX,1            ;* 4
        ADD     AX,BX
        ADC     DX,CX           ,* 5
        PUSH    CS
        POP     DS              ;PUT DS HERE
        MOV     BX,TPHH         ;GET TICKS/HOUR * 5 HIGH
        MOV     CX,TPHL         ,AND TICKS/HOUR * 5 LOW
        MOV     SI,-1           ;SET A COUNTER
DIVLP:  INC     SI              ;DIVIDE TICKS BY TICKS/HOUR
        SUB     AX,CX
        SBB     DX,BX
        JNB     DIVLP
        ADD     AX,CX
        ADC     DX,BX           ;DX,AX = REMAINDER
        MOV     WORD PTR HOU,SI ,SAVE HOUR
        MOV     CX,TPM
        DIV     CX              ;DIV BY TICKS/MINUTE * 5
        MOV     MIN,AL          ;SAVE RESULT
        MOV     AX,DX           ;GET REMAINDER
        MOV     CL,TPS
        DIV     CL              ;DIV BY TICKS/SECOND * 5
        CMP     AL,OLDSEC       ;HAS A SECOND PASSED?
        JZ      NOUPD           ,IF NOT, EXIT

;       A SECOND HAS PASSED, PRINT TIME ON THE SCREEN

UPDATE: CLD                     ;ENSURE FORWARD DIRECTION
        MOV     OLDSEC,AL       ;UPDATE OLD SECOND
        CMP     AL,60           ;INTEGER MATH ERROR?
        JNZ     NERR            ,NO
        DEC     AL              ;ELSE, CORRECT
NERR:   CALL    CONASC          ;CONVERT SECONDS TO ASCII
        MOV     SECOND,AX       ;RESULT TO BUFFER
        MOV     AL,MIN          ;GET MINUTES
        CALL    CONASC          ;CONVERT TO ASCII
        MOV     MINUTE,AX       ;RESULT TO BUFFER
        MOV     AL,HOU          ;GET HOURS
        CALL    CONASC          ;CONVERT TO ASCII
        MOV     HOUR,AX         ;RESULT TO BUFFER
        MOV     AH,GVS
        INT     SCRINT          ;GET VIDEO STATE
        PUSH    BX              ;SAVE PAGE
        MOV     BX,OFFSET MODES ,POINT TO VIDEO MODES
        XLAT                    ;GET MODE CODE
        MOV     CMODE,AL        ;SAVE DECODED MODE
        POP     BX              ;GET PAGE
        MOV     AH,GCP
        INT     SCRINT          ,GET CURSOR POSITION
        MOV     CPOS,DX         ;SAVE IT
        MOV     DH,ROW          ;GET ROW FOR CLOCK
        MOV     DL,31           ;ASSUME COLUMN 31
        CMP     BYTE PTR CMODE,2;80 COLUMN MODE
        JB      PTIME0          ;NO
        MOV     DL,71           ;ELSE, SET COLUMN 71
PTIME0: MOV     AH,SCP
        INT     SCRINT          ;SET CURSOR TO TIME AREA
        MOV     CX,9            ;SET A COUNTER
        MOV     BL,WHITE        ;ASSUME WHITE COLOR
        CMP     BYTE PTR CMODE,0;40 COL B/W?
        JZ      PTIME1          ,YES
        CMP     BYTE PTR CMODE,2;80 COL B/W?
        JZ      PTIME1
        MOV     BL,COLOR        ;ELSE, USE COLOR
PTIME1: MOV     SI,OFFSET TIMSTR;POINT TO TIME STRING
PTMLP   LODSB                   ;GET A DIGIT
        PUSH    CX              ,SAVE COUNTER
        PUSH    SI              ;SAVE POINTER
        CALL    PDIGIT          ;PRINT DIGIT
        POP     SI
        POP     CX
        LOOP    PTMLP           ;LOOP UNTIL DONE
PDONE:  MOV     AH,SCP
        MOV     DX,CPOS
        INT     SCRINT          ;RESTORE CURSOR TO
                                 USER'S POS
NOUPD:  POP     ES
        POP     DS              ;RESTORE REGISTERS
        POP     DI
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        CLI
        MOV     SS,CS:SYSSTKS   ;RESTORE SYSTEM STACK
        MOV     SP,CS:SYSSTK
        STI
```

```
        MOV     CS:CLKFLG,1     ;RE-ENABLE CLOCK
        JMP     TIMEXIT         ,AND EXIT

;       CONVERT NUMBER IN AL TO ASCII DIGITS IN AL, AH

CONASC: MOV     AH,0            ;CLEAR AH
        MOV     BH,10           ;GET RADIX
        DIV     BH              ;DIVIDE BY IT
        ADD     AX,'00'         ;ADD ASCII
        RET

,       PRINT A DIGIT ON THE SCREEN

PDIGIT: MOV     AH,WCHAR
        MOV     CX,1
        INT     SCRINT          ;WRITE CHARACTER
        MOV     AH,GCP
        INT     SCRINT          ;GET CURSOR POSITION
        INC     DL              ;MOV IT OVER
        MOV     AH,SCP
        INT     SCRINT          ;SET NEW CURSOR POSITION
        RET

.       LOCAL SYSTEM INTERRUPT PROCESSOR

MYSYS:  CMP     AH,0F2H         ,CLOCK PROCESS CODE?
        JZ      CLKSYS          ;YES
        JMPF                    ;ELSE, EXIT
SYSADR  DW      0,0
CLKSYS: MOV     CS:CLKFLG,AL    ,SET CLOCK CONDITION
        IRET

ENDRES:                         ;END OF RESIDENT CODE

,       SET UP CLOCK VECTOR AND INITIALIZE TIME
;       THEN EXIT WITH PROGRAM RESIDENT

SETUP:  PUSH    DS              ;SAVE DS
        XOR     AX,AX
        MOV     DS,AX           ;DS AT 0
        MOV     SI,OFFSET TIMEINT  ;POINT TO TIMER INTERRUPT
        LES     DI,DWORD PTR [SI]  ;GET VECTOR IN ES:DI
        CMP     WORD PTR ES:-2[DI],'KC' ;CLOCK ALREADY IN?
        JZ      ITSIN           ;YES
        CLI                     ;ELSE, TURN OFF INTERRUPTS
        MOV     WORD PTR [SI],OFFSET MYTIME ;PUT IN MY VECTOR
        MOV     2[SI],CS        ,AND THIS SEGMENT
        POP     DS              ;RESTORE DS
        MOV     WORD PTR TIMADR,DI ;PUT OLD VEC. AT OUR EXIT
        MOV     WORD PTR TIMADR+2,ES
        STI
        PUSH    DS
        XOR     AX,AX
        MOV     DS,AX           ;DS AT 0
        MOV     SI,OFFSET SYSINT;POINT TO SYSTEM INTERRUPT
        LES     DI,DWORD PTR [SI]       ;GET VECTOR
        MOV     WORD PTR [SI],OFFSET MYSYS ;PUT IN MY VECTOR
        MOV     2[SI],CS
        POP     DS
        MOV     WORD PTR SYSADR,DI      ;PUT OLD VECTOR HERE
        MOV     WORD PTR SYSADR+2,ES
        MOV     DX,OFFSET ENDRES;POINT TO END OF RES. CODE
        INT     27H             ;EXIT, LEAVE PROGRAM HERE
ITSIN:  INT     20H             ;EXIT, NOTHING DONE
SCLK    ENDS
        END     START
```

CLK Source Listing

```
        TITLE   SCREEN CLOCK CONTROL PROGRAM
        PAGE    ,132
;       THIS PROGRAM ALLOWS YOU TO TURN OFF OR ON THE SCREEN
;       CLOCK PRODUCED BY THE SCRNCLK PROGRAM.  TO USE IT,
;       ENTER
,
,       A>CLK OFF               TO TURN THE CLOCK OFF
;       A>CLK                   TO TURN ON CLOCK
;
;       BY P  SWAYNE, HUG   23-MAY-86

,       DEFINITIONS

CLK     SEGMENT
        ASSUME  CS:CLK,DS:CLK,ES:CLK,SS:CLK
        ORG     5CH
DFCB    LABEL   BYTE            ;DEFAULT FCB
        ORG     100H

START:  MOV     BX,0F201H       ,ASSUME CLOCK ON
        MOV     AL,DFCB+1       ;GET FIRST FCB CHAR
        CMP     AL,' '          ;SPACE?
        JZ      CLKON           ;NO ARGUMENT, CLOCK ON
        CMP     AL,'O'          ;OFF?
        JNZ     CLKON           ;NO, TURN CLOCK ON
        MOV     BL,0            ;CLOCK OFF CODE
CLKON:  MOV     AX,BX
        INT     21H             ;UPDATE CLOCK CONDITION
EXIT:   INT     20H             ;AND EXIT

CLK     ENDS
        END     START
```

*

HUG NEW PRODUCTS

HUG P/N 885-8044-37 MS-DOS
TCSpell $20.00

Introduction: TCSpell is a spelling checker designed to be quick and easy to use. Dictionary size is only limited by disk space while the document size is limited only by the memory if the number of unique words exceeds the room left in memory.

Requirements: TCSpell will run on systems with as little as 64k of memory but the number of unique words in a document is limited to approximately 600 words. It runs on the H/Z 150-PC compatible computer series, either monochrome or color monitor, with MS-DOS (any version).

This disk contains the following files:

TCSPELL	.COM	TCSPELL	.TUT
TCSPELL	.DOC	MASTER	.DIC
UNCOMP	.COM	COMPRESS	.COM
TCMERGE	.COM	README	.DOC

Author: Tim Schultze

Program Content: The TCSpell program is a spelling checker program that uses up to 10 disk based dictionaries. The dictionaries may exist in compressed format or as a sorted list of lowercase words separated by a carriage return and linefeed. The program reads all the unique words in from the file to be spell checked and then checks each specified dictionary in order to eliminate all valid words. The user is then prompted for action on each unknown word. Available actions include: Add to dictionary, store in CR/LF file name SAVE.$$$, Correct in file, Show word in context, and Ignore the word.

TCSpell operates on standard ASCII (CR/LF) files and Wordstar format files. Soft hyphens and hyphens across line boundaries are handled correctly. Hard hyphens are included as part of a word, whereas soft hyphens are ignored.

The TCSpell program was designed to be easy to use, fast, and as complete as each user requires. A dictionary, containing approximately 20,000 words, is included along with utilities to compress, expand (uncompress), and merge existing dictionary files. The user then has the option of expanding the supplied dictionary, editing it with any editor, merging an existing dictionary, or compressing an existing dictionary for faster access and less storage space.

Included on the disk are a tutorial/manual (TCSPELL.DOC) and an example file (TCSPELL.TUT) to be checked in parallel with the tutorial.

Comments: None

TABLE C Rating: (9) *

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.

Note: All special update offers announced in REMark (i.e. ZPC II update) must be paid by check or money order, payable to the Heath Users' Group. **NO CREDIT CARDS ACCEPTED.** ZPC II contains only one disk. It is a combination of ZPC I and the ZPC Support disk plus added improvements. Thank you.

HUG Price List

The following HUG Price List contains a list of all products 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	
885-1135-[37]	HDOS Variety Pkg	20.00	76

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	45
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
885-8042-37	ZDOS/MSDOS Poker Party	20.00	77

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/MSDOS

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
885-3034-37	ZDOS/MSDOS ZPC Support Pkg	10.00	72
885-3037-37	MSDOS ZPC II	60.00	76
885-8029-37	ZDOS Fast Eddy	20.00	53
885-8035-37	MSDOS DOCUMAT and DOCULIST	20.00	70
885-8041-37	ZDOS/MSDOS Orbits	25.00	75

H/Z100 ZDOS/MSDOS - 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
885-3032-37§	MSDOS Halley's Comet Locater	20.00	70
885-3035-37§§	MSDOS SPELL5 & SPELL5F	20.00	72
885-3038-37§	ZDOS/MSDOS DEBUG Support Util	20.00	77
885-8039-37§§	MSDOS DPATH	20.00	74
885-8040-37§§	MSDOS HELP Programs	20.00	74

§ All program files run on both

§§ Program files run partially on both

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
885-6009-37	MSDOS Screen Saver Plus	20.00	76
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

SEEK.COM

A CP/M Assembly Language Learning Adventure

Part 2

Introduction

In Part 1, we took an idea and explained how to convert it into an ASM program. Moreover, details related to CP/M's assembler and interface functions were described. Now we'll use that information, and get into the details of loading a disk file to RAM. But first, refer to the Basic Flow Chart (Chart #1). Note how it flows — load the file, count the lines, etc. This suggests the use of subroutines — go perform a task and return, and so on. Consequently, we can write program instructions to accomplish these tasks without knowing the details of the routines. In the program instruction area of SEEK.ASM, we can write:

```
BEGIN   CALL    OPNRD       ; Open file & read into RAM.
        CALL    LINES       ; Count lines
        CALL    WORDS       , Count words.
        CALL    SNTCS       ; Count sentences
ENPGM   JMP     00000       ; End program - warm boot
```

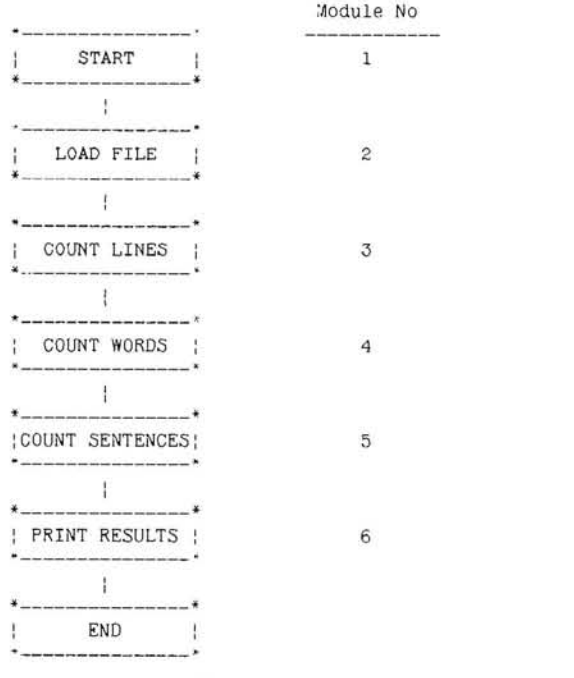


Chart 1
SEEK.COM Basic Flow Chart
As Shown in Part 1 of this Series

M. D. Zapolski, Sr.
226 N. West Avenue
Bridgeton, NJ 08302

The CALL instruction is like the BASIC GOSUB. Accordingly, this series of steps invokes 4 separate subroutines before ending the program. As with most anything, there are other ways to end a program. This method simply jumps (JMP — like a BASIC GOTO) to RAM location 0000H where it finds a vector (another JMP). And it sends the program to a routine which performs a CP/M "warm boot." The "warm boot" restarts CP/M by copying the CCP and BDOS from the system disk into RAM. The disks are also reset at this time. If you recall from Part 1, I implied that some programs overwrite the CCP. Performing a "warm boot" is how the programmer restores the CCP.

Step 2 — Module 2 (Load File)

Loading a disk file into RAM consists of three general steps: (1) open the file, (2) set the DMA, and (3) read the file into RAM (128 bytes at a time). As noted in Part 1, human error and machine limitations will make this task somewhat more complicated. But, programmers should learn to anticipate these types of problems and "trap" them, such that the program doesn't halt or go into "never-never land." To this end, our program will include some of these traps. In some instances, they provide the user with the capability to reenter the program. Our traps will only result in a displayed message (Msg) before the program's end.

To start with, we will use CP/M functions 15, 20, and 26. So, in the equate area of SEEK.ASM, enter the additional equates shown in Figure 5 below BDOS EQU 05H. Feel free to use decimal values where I used hex. But, make sure you use the proper base identifier [ref 2, pg. 2–10]. Did you notice the FCB EQU 05CH? FCB stands for File Control Block. For sequential files, it is a 33 byte area (starting at 05CH) that CP/M uses to store the file name entered on the command line. For example, at the A> prompt you enter SEEK TEST.DOC. The CCP searches the default disk for the file SEEK.COM. If found, the CCP places the file name TEST.DOC into the default FCB at 05CH. Then, when an OPEN FILE request is made via function 15, CP/M scans the disk direc-

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. Alaska and Michigan residents call: 616-982-3411.

Watch for additional MEMBER ONLY discounts in upcoming issues of REMark®!

REMark Discount List

	Item	Last Adv'd Price	HUG Member Price	Qty Left
H-19-3	H-19 to H-88 Mod Kit	220.00	199.95	11
H-88-1	Floppy Disk Accessory	150.00	99.95	42
H-88-9	Drive Mounting & Inst	25.00	14.95	3
H-89A	W/O CRT, Ser I/O, Disk Cont.	----	399.95	4
ADP-1501	Stereo Synthesizer	99.95	74.99	69
GDP-1433	Digital Golf Trainer	199.95	149.95	40
HCA-9	Print Stand For H-125	99.00	59.95	13
NEP-1210	Assem. Set-back Thermostat	79.95	39.99	86
PA-100-1	USR Telpak (PC&ZDOS)	99.00	69.95	2
PC-131	USR PCM-5 1200B Modem (PC)	349.95	239.95	37
PC-200	USR 1200B Modem Card (S100)	399.00	279.00	40
PCA-120-1	6-PAK Plus Game Acc'y.	39.95	19.95	16
PM-111	Stat-Attack Anti-Static Kit	24.95	9.95	73
PP-215	EPSON JX-80 Color Printer	699.95	449.95	25
PP-240	EPSON LQ-1500	1299.00	999.00	10
PP-400	Koalapad Touch Tablet	99.95	69.95	127
PPA-240-3	LQ-1500 Interface: Serial	149.95	129.95	8
PPA-240-4	LQ-1500 Interface: Parallel	94.95	79.95	30
PPA-400-1	Koala Speedkey Software	59.95	29.95	146
ZA-100-4	S-100 Extender Board	59.00	19.95	225
ZA-219	Amber CRT/H-121 (Tube only)	75.00	14.95	537
ZG-219	Green CRT/H-121 (Tube only)	75.00	14.95	84
ZW-219	White CRT/H-121 (Tube only)	75.00	9.95	142

* New additions this month.

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

tory for a match between it and the FCB. The FCB contains other information besides the file name. I suggest you review ref. 3a, pg. 93, and add the FCB equate to your ASM file.

```
. 9/19/85                          SEEK.ASM
;
; SEEK loads a text file into memory, counts the # of
; words, lines, and sentences in the file, the displays
. the results
,        -------- By: M.D  Zapolski, Sr. ---------
;
               BDOS    EQU     05H
               FCB     EQU     05CH
               OPEN    EQU     0FH
               STDMA   EQU     01AH
               RDSEQ   EQU     014H
;
               ORG     0100H
BEGIN   CALL    OPNRD          ; Open file & read into RAM
        CALL    LINES          ; Count lines
        CALL    WORDS          ; Count words
        CALL    SNTCS          , Count sentances
ENPGM   JMP     0000H          , End program - warm boot
;
               END     BEGIN
```

Figure 5
SEEK.ASM — Stage 2

Now it's time to get some paper and pencil, and a BIG eraser. Try to rough-out a flow chart starting with the fundamental requirements of the Open File routine. It should look something like Chart #2. After entry to the routine and opening the file, we set the DMA, and . . . What's a DMA? Let me clear the air. The Direct Memory Address (DMA) is used by the READ SEQUENTIAL function. It is the starting address for the 128 byte record being read into RAM. Remember in Part 1, I mentioned we had to tell the computer where to put the information it reads from the disk. This is the purpose of the DMA. So, if we have to specify the DMA for the 1st record, it will have to be specified for all successive records as well. Assuming the DMA is set, then read the record, and check to see if we've reached the EOF. If not, compute the next DMA, and read the next record. If at the EOF, then return to the CALLing point.

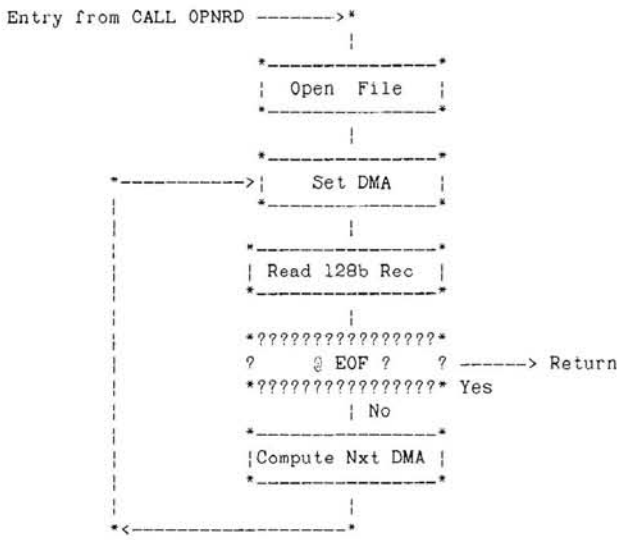


Chart 2
Open File And Read Sequential Routine
(Stage 1)

Don't be too anxious to start converting the flow chart into ASM opcodes. There are more details to incorporate into the flow chart. But, for a moment, I'd like to discuss my flow charts. Although, they don't necessarily use conventional symbols. They function as would conventional charts. And, from this point onward, each chart will serve the purpose of two. The first being to represent a first cut (rough) flow chart. These first cut activities will be those shown in the rectangular boxes like those in Chart #2. The second will show additional activities which result from afterthought. Error trapping, messages, and program refinements are examples of these activities, and are represented by the pointed-end boxes. Or, no box at all.

Getting back to the Open File routine, take Chart #2 and try to sketch in error trap boxes to:

1. Make sure a file name was specified.
2. Check the status of the last "read" operation (ok or not).
3. Ensure the next DMA doesn't exceed the RAM limit (bottom of the CCP).

And, since we will want to know why the program ended, insert a corresponding message for each trap. Finally, provide a message that tells the user when file loading starts. When you're finished, compare your work to Chart #3. If your chart doesn't closely match, then a review of the CP/M functions may be in order.

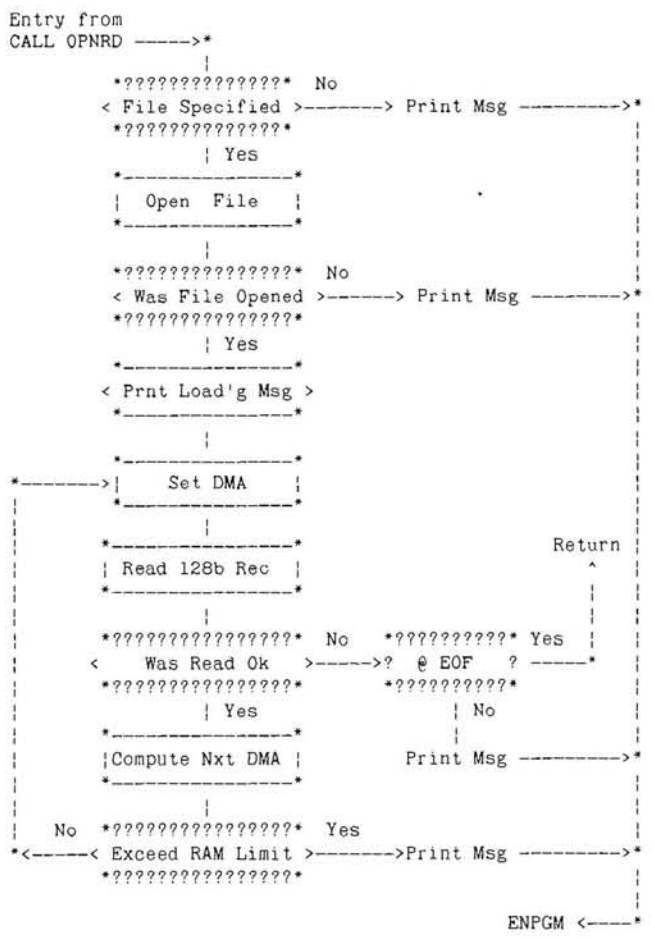


Chart 3
Open File And Read Sequential Routine
(Stage 2)

The final objective is to translate the flow chart into ASM opcodes. First, the file load routine is entered by CALL OPNRD. As such, its label will be OPNRD. Then, to check for a file name, we need to inspect the FCB at 05DH. If a name is specified, then the first character position of the name in the FCB will be something other than a blank space. Add these lines to SEEK.ASM to accomplish the check:

```
OPNRD   LXI     H,5DH
        MOV     A,M
        CPI     20H
        JZ      NFILE
```

The HL register pair is loaded with the value 5DH. Since the HL pair acts as a RAM pointer for data movement, the data at this address can be MOVed to the A register. Whereupon it's CoMpared to 20H (ASCII for a space). If a space is found, the Zero Flag is set TRUE (i.e, =1), and a JuMP is made to NFILE. The associated error message, and those to follow, will be described later. Now assuming a file name was entered, we can open the file as follows:

```
        LXI     D,FCB
        MVI     C,OPEN
        CALL    BDOS
        CPI     0FFH
        JNZ     RDFILE
```

As required by CP/M function 15, the FCB address is placed in the DE register pair, C is loaded with the function #, and a call is made to BDOS. After exiting the operation, the A register contains a status (directory) code. If this code is NOT 0FFH, the file was successfully opened. The program then jumps to RDFILE. Our chart now indicates a "loading" message should be displayed. But I don't want to change your train of thought. So, insert 2 NOPs (No Operation) as position savers, and we'll continue.

Take a moment to study Chart #3. Besides the CP/M functions, what else is needed to start the file read operation? Somehow we have to determine the value of the first DMA. As both SEEK.COM and the text file will reside in the TPA, the DMA becomes a function of SEEK.COM's file size (bytes). Right now, and until the end of program development, that information is unknown. One answer could be to estimate SEEK.COM's size.

A more efficient method is to use the assembler directive "$". When the assembler encounters the "$" in the operand field of an instruction, it associates the current value of its "origin counter" with the corresponding label. This counter keeps track of the addresses used during the assembly process. If the line TXTB EQU $ were entered in the DB/DW Area, it would assign the current address in the assembler's internal counter to TXTB. Our problem is solved. Add this new line as the last instruction line, but before the END directive. This example should also point out that EQU, and the other directives, may appear most anywhere in the ASM file. Coupling the RDFILE routine to the Set DMA and Read Sequential functions results in:

```
RDFILE  NOP
        NOP
        LXI     D,TXTB
        MVI     C,STDMA
        CALL    BDOS
        LXI     D,FCB
        MVI     C,RDSEQ
        CALL    BDOS
        CPI     0H
        JZ      NXBLK
        CPI     01H
        RZ
```

Consequently, DE is loaded with the value of TXTB, the DMA is set, DE is then loaded with the value of FCB, and we promptly loose the value of TXTB. This may not seem to be a problem. But, how then can we compute new DMA values? It turns out that by storing the current value of the DMA, the next DMA can be easily found. TXTB's value is a 16 bit address. Therefore, a Data Word (DW) is required to save TXTB. Take note, DBs store 8 bit values, whereas DWs store 16 bit values. In the DB/DW area, enter the line SAV1 EQU 0. This means that the label SAV1 represents a 16 bit value (DW) whose initial value is 0. In this program, the actual address of this storage area is not important. To transfer the contents of SAV1 to, and from, the program, we'll use the SHLD and LHLD instructions. Now change LXI D,TXTB to LXI H,TXTB and follow it with the line SHLD SAV1.

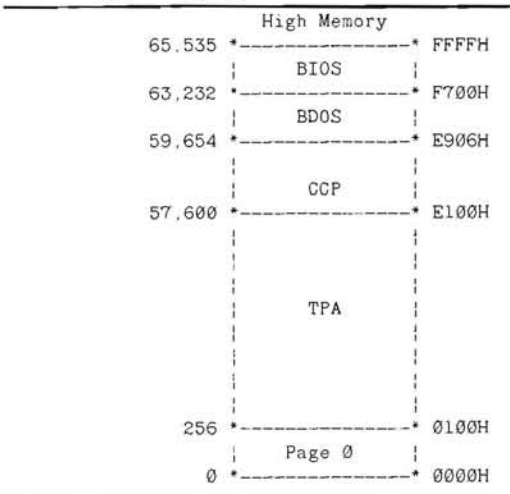
After reading the first record, a read status check is performed. If the A register contains 00H, the read operation was successful. This means 128 bytes (80H) of the text file is now in RAM starting at TXTB. If the A register doesn't equal 00H, then the EOF may have been reached. At the EOF, the status code = 01H. The CPI 01H will provide the necessary check and set the Zero Flag for the RZ instruction. If not set, the routine "falls into" an error trap.

The routine at NXBLK will compute the next DMA:

```
NXBLK   LHLD    SAV1
        LXI     D,80H
        DAD     D
        SHLD    SAV1
```

Since the previous routine may modify the HL registers, an LHLD will restore the correct value of TXTB. Then load DE with 80H, add 80H to HL (DAD D), and store the new DMA in SAV1. Afterwards, check the new DMA to see if it exceeds our RAM limit. And that limit is . . . Yes, the bottom address of the CCP (E100H — Figure 2, Part 1). This is the CKSIZ routine and looks like this:

```
CKSIZ   MVI     A,TOP
        CMP     H
        JNZ     RDBLK
```



Basic Input/Output System (BIOS) — This area contains the customized routines that operate the computer's peripherals.

Basic Disk Operating System (BDOS) — This area contains the general programs for the operation of the peripherals.

Console Command Processor (CCP) — This area contains resident CP/M programs like DIR, ERA, REN, SAVE, TYPE, and USER.

Transient Program Area (TPA) — This is the working area of memory where your executable programs & data are stored.

System Parameter Area (Page 0) — This area contains important information such as the BIOS and BDOS jump vectors, default File Control Block (FCB), current drive and user #, IOBYTE, and the restart area.

Figure 2
H/Z-100, CP/M-85 (Ver 2.2.103) Memory Map
As Shown in Part 1 of this Series

The RAM limit is a 2 byte value. Its Most Significant Byte (MSB) is E1H. The LSB is 00H and doesn't concern us. Upon entry to CKSIZ, the MSB of the new DMA is in the H register. If this value is less than E1, the RAM limit has not been exceeded. In this case, the CMP H instruction will NOT set the Zero Flag. Note the new equate, TOP EQU 0E1H. An important aspect of CKSIZ is it performs its check before the next read operation. As a result, it prevents writing beyond the RAM limit. Knowing the new DMA, the next record can be read. So, where does the RDBLK label go? The new DMA is in the HL pair, but will be needed in DE for the Set DMA function. If we insert the line RDBLK XCHG after the first SHLD SAV1 instruction, the dilemma disappears.

The last part of this module involves displaying messages. The method I use is just one of several available routines. We'll start by selecting CP/M function #2 — Console Output. So, add the equates CNOUT EQU 02H, CR EQU 0DH, and LF EQU 0AH to your file. Initially, the PRNSEQ routine looks a little tricky. But once explained, you'll understand its operation:

```
PRNSEQ  XRA     A
        ADD     M
        RZ
        CALL    CHROUT
        INX     H
        JMP     PRNSEQ
CHROUT  PUSH B!PUSH D!PUSH H
        MVI     C,CNOUT
        MOV     E,A
        CALL    BDOS
        POP H!POP D!POP B
        RET
```

This routine is implemented by two lines:

```
        LXI     H,MSG4
        CALL    PRNSEQ
```

Now, assume this message exists in the DB/DW area:

```
MSG4    DB      'Loading File   ',CR,LF,0
```

When the LXI H,MSG4 instruction is executed, the address of MSG4 is loaded into the HL pair. The subsequent CALL to PRNSEQ then causes the message to be displayed on the CRT screen. PRNSEQ first zeros the A register (exclusive OR A with itself. Next, the contents of the first RAM location (pointed to by HL) are added to the A register. The use of ADI vs MOV provides the ability to test for the end of the message. ADI is a math operation. And, the result of a math operation affects the status of the condition flags. Suppose a 0 were added (as a delimiter) to the end of a character string. Adding 0 to a 0 in the A register would then SET the Zero Flag TRUE (i.e, =1). This zero could be used as a condition to return (RZ) from the original CALL. In contrast to this, a letter would not set the flag. Instead, it allows the CALL to CHROUT, which outputs the character to the console device (the CRT). The process continues until the 0 delimiter is detected.

What's all that PUSHing and POPing about, you ask? It seems CP/M is not kind to programmers when they CALL to BDOS. The contents of registers are not preserved [ref 3a, pg. 92]. For protection, values in the BC, DE, and HL registers are PUSHed onto the stack before the CALL, and POPed off afterwards in reverse order. A word to the wise!

I've saved MSG1 for last because it uses H-19 control codes. As a general rule, when my programs start they clear the screen and home the cursor. Further, I use reverse video to highlight error messages. First, add these lines prior to the CALL OPNRD instruction: LXI H,MSG1, CALL PRNSEQ. Second, move the label BEGIN to the LXI H,MSG1 instruction. Third, enter the MSG1 DB shown on Figure 6. When MSG1 is displayed, it outputs the ESC 'E', ESC 'p', and ESC 'q' code sequences. In this way, the H-19 escape codes can be used in ASM.

To complete this module, add the PRNSEQ and CHROUT routines to SEEK.ASM, and replace the NOPs with the LXI and CALL PRNSEQ instructions. Then, using Figure 6 as a guide for wording, add the appropriate messages. With these revisions, Figure 6 now lists the current composite SEEK.ASM program. It includes routines for:

1. Displaying messages with H-19 escape codes.
2. Opening and loading a text file.
3. Determining DMA values.
4. Ensuring self-imposed RAM limits are not exceeded.
5. Error trapping.

```
; 9/19/85                SEEK.ASM
;
; SEEK loads a text file into memory, counts the # of
; words, lines, and sentences in the file, then displays
; the results
;          ---- By: M.D. Zapolski, Sr  ----
;
                BDOS    EQU     05H
                FCB     EQU     05CH
                OPEN    EQU     0FH
                STDMA   EQU     01AH
                RDSEQ   EQU     014H
                CNOUT   EQU     02H
                CR      EQU     0DH
                LF      EQU     0AH
                TOP     EQU     0E1H
;
        ORG     0100H
;
BEGIN   LXI     H,MSG1    , Clr scrn, hm cursor, intro msg
        CALL    PRNSEQ
        CALL    OPNRD     ; Open file & read into memory
        CALL    LINES     ; Count lines
        CALL    WORDS     ; Count words.
        CALL    SNTCS     ; Count sentaces
EPGM    JMP     0000H     ; End program - warm boot
;
OPNRD   LXI     H,5DH     , Look for 1st. char  of Fname
        MOV     A,M
        CPI     20H
        JZ      NFILE
        LXI     D,FCB
        MVI     C,OPEN    , Open file- CP/M func.# 15
        CALL    BDOS
        CPI     0FFH
        JNZ     RDFILE
        LXI     H,MSG3
        CALL    PRNSEQ    ; File not found
        JMP     EPGM
NFILE   LXI     H,MSG2
        CALL    PRNSEQ    , No Fname specified
        JMP     EPGM
RDFILE  LXI     H,MSG4
        CALL    PRNSEQ    , Loading file
        LXI     H,TXTB
        SHLD    SAV1
RDBLK   XCHG              ; DMA addr  --> (DE)
        MVI     C,STDMA   , Set DMA- CP/M func.# 26
        CALL    BDOS
        LXI     D,FCB
        MVI     C,RDSEQ   ; Read seq.- CP/M func.# 20
        CALL    BDOS
        CPI     0H        , Read ok ?
        JZ      NXBLK     , Yes, if (A)=0
        CPI     1H        ; 1 EOF ?
        RZ                ; Yes, return-file in memry
        LXI     H,MSG5
        CALL    PRNSEQ    , File read error
        JMP     0000H
NXBLK   LHLD    SAV1
        LXI     D,80H
```

```
        DAD     D           , Add 80h to HL
        SHLD    SAV1
CKSIZ   MVI     A,TOP
        CMP     H           ; At RAM limit?
        JNZ     RDBLK
        LXI     H,MSG
        CALL    PRNSEQ      ; Oops, too big
        JMP     EPGM
        RET
PRNSEQ  XRA     A           , Zero (A)
        ADD     M           ; (A)+<M> -->(A)
        RZ                  ; Return if (A)=0 (seq. complete)
        CALL    CHROUT      ; Send character to CRT (console)
        INX     H           , Point (HL) to next character
        JMP     PRNSEQ      ; loop until done
CHROUT  PUSH B!PUSH D!PUSH H
        MVI     C,CNOUT
        MOV     E,A
        CALL    BDOS
        POP H!POP D!POP B
        RET
                            , DB/DW, Message Area
MSG1    DB   1BH,'E',CR,LF,1BH,'pSEEK.COM',1BH,'q
             Ver 1.2 (10/21/85)'
        DB   CR,LF,'Coded by. M.D. Zapolski, Sr ',
             CR,LF,LF,LF,0
MSG2    DB   'No filename specified. . Syntax
             SEEK Fname.ext',CR,LF,0
MSG3    DB   1BH,'pERROR:',1BH,'q File not found   ',
             CR,LF,0
MSG4    DB   'Loading file . .',CR,LF,0
MSG5    DB   1BH,'pERROR:',1BH,'q Can''t read file. .',
             CR,LF,0
MSG6    DB   'File size exceeds memory availability   ',
             CR,LF,0
MSG7    DB   LF,'Searching File ...',CR,LF,0
SAV1    DW   0
CTR     DW   0,0,0
TXTB    EQU  $
        END  BEGIN
```

Figure 6
SEEK.ASM — Stage 3

If you temporarily change the CALL instructions below CALL OPNRD to comments, you can assemble, load, and run SEEK .COM. This will be an excellent exercise in the assembly process. Not to mention testing your ASM file for typographical errors. Moreover, you can use DDT.COM on SEEK.COM to examine the text file in RAM, and to monitor how the program steps through its instructions.

Summary

Up to this point, several aspects of ASM and CP/M have been covered. And, yet, we haven't reached the "heart of the program. Next month, we'll finally work on the text counting routines. Until then "walk" through the File Load routine. Is there another way to write its routines? Hmmm . . .

EasyPC For The Z-100

LtCol. Myles P. Somers USMC
Computer Course Director
Armed Forces Staff College
Norfolk, VA 23511-1216

The new IBM compatibility board from UCI is the best on the market. Not only does it display all colors that the IBM displays, but it has sound and highlighting, too. Since this board is a hardware enhancement to the Z-100, it is very fast; as a matter of fact, EasyPC is faster in the IBM mode than it is in the Z-100 mode. Another nice feature is that the screen paints just like an IBM, only much faster.

Description

UCI's EasyPC is an IBM-PC emulator for the Z-100 computer that gives full emulation of the IBM-PC on the Z-100. EasyPC is a 3-board set (one piggyback board that fits onto the Z-100 motherboard, and two S-100 boards — a Video Master board for IBM-PC video display and a Floppy Master board which is a replacement for the Z-100 floppy disk controller board). The system also contains a speaker and all required wiring assemblies to connect the system to the Z-100. Only one extra S-100 slot in the Z-100 is required if no communication port is desired (more on this later).

Installation

I was going to write a section on how to install the EasyPC boards and the procedure used in setting up the hard disk. But the UCI instruction manual does an excellent job at this, so there is only a little I can add of interest. UCI supplies some replacement chips (ICs) with EasyPC that must be used on the Z-100 motherboard, and these replacement chips are different (faster) from the ones used in the Z-100. However, don't be alarmed if the chip you are replacing is the same as the replacement chip from UCI, as Zenith has used some of the faster chips in recent months. Installation takes about 30 minutes from start to finish.

From Start To Present

I received my EasyPC board around the 1st of December 1985; it was one of the first off the line. As a result, it had a few problems, but this was something I expected, and I found the solution to all of them over the next couple of months.

My system is a Z-118, 8Mhz, 768k, with a Winchester disk. After successful installation of EasyPC, I spent the next several days trying all the IBM programs I could find, and all ran with no problem except those that required "BASRUN.EXE," which is used with some compiled BASIC programs. If BASRUN was used, EasyPC would lock up and the Z-100 would have to be rebooted. UCI has fixed this with ROM release (1.60), and now programs that require BASRUN operate as on an IBM-PC, only twice as fast. I had ROM 1.60 installed in my Z-100, and it worked fine; it also speeded up the IBM mode, so it was faster than before (9.65Mhz effective clock speed). The newest ROM released by UCI is ROM 1.63, which allows use of the new V-20 processor from NEC. (This is discussed later in this article.)

Highlighting

The next area I had to work on was highlighting. I called Randy Matus (UCI Technician), and he told me which pin on the EasyPC video board was the signal (pin 6 on plug P4). If you have a ZVM-135 color monitor, you can have highlighting; however, the cable supplied with the ZVM-135 for the Z-100 is not wired for it. This is very simple to correct.

1. Run a 20-gauge wire from EasyPC video board to pin hole 7 on the Z-100 RGB connector (not used by Zenith); wire needs a CRIMP CONTACT on one end and a female pin on the other (solder both). Next, fit the crimp contact into the UCI P4 plug pin hole 6 and the female pin into pin hole 7 on the Z-100 RGB connector.
2. Cut a piece of 20-gauge wire as long as the Z-100 video cable, solder a male pin on one end, disassemble plugs, solder one end of wire to pin 7 in Z-100 RGB cable plug, place the other end in ZVM-135 plug connector pin hole 15, and reassemble plugs. Now highlighting works just like an IBM-PC when using the IBM mode.

Volume Control

The EasyPC comes with its own speaker, but it has no volume control, so it is impossible to turn up or down. However, the

Heath/Zenith Computers & Electronics welcomes you to the 5th International

HEATH USERS GROUP CONFERENCE

HYATT REGENCY O'HARE, CHICAGO, AUGUST 15-16-17

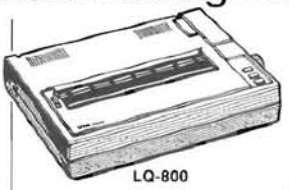
We're celebrating with SPECIAL SAVINGS like these* at the Conference and in all stores during August.


H-100

EASY PC from UCI Corporation

Only $499.00 Was $699.00 SAVE $200.00

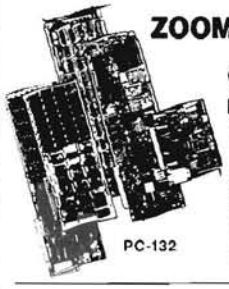
An emulator with three separate boards that lets you run IBM PC versions of many popular business programs on your H-100.


LQ-800

EPSON LQ-800 Letter Quality Printer

Only $649.00 Reg. $799.00 SAVE $150.00

Features bi-directional printing and special Quiet Mode operation.


PC-132

ZOOM/MODEM PC 1200

Only $249.95

Reg. $349.95 SAVE $100.00

Hayes compatible. Features almost totally automatic operation, plus faster and more reliable dialing than most modems.


CM-100

C.ITOH RGB COLOR MONITOR

Only $399.00

Super color monitor. Works with the IBM PC and compatibles.


PP-402

FLIGHT SIMULATOR Software and Joystick

Only $79.95 SAVE!

Hard Disk Upgrade with One-Year Exchange Warranty — 10 Meg for $499.00, 20 Meg for $599.00

20% OFF ALL COMPUTER ACCESSORIES**

**Diskettes, printer paper, print wheels, cables, ribbons, printer stands.

and at the show you'll find just UNBELIEVABLE BUYS!

Prices good through September 6, 1986. Call 616-982-3614 for store locations or consult your Heathkit Catalog.

HZC-229 *Products featured are at sale price. HUG discounts do not apply

Heath® ZENITH® Computers & Electronics

Units of Veritechnology Electronics Corporation

ZVM-135 does have an excellent speaker system with volume control which I wanted to use. I worked on this a while and I did get it to work. Plug SP (speaker plug) on the EasyPC video board has two pins — pin 1 is the signal and pin 2 is +5 volts. If you leave the UCI-supplied speaker connected and run a wire from PS pin 1 to the audio input on the ZVM-135, the monitor's speaker system works. The result is sound with volume control, and it is at least 3 times clearer on the larger speaker. What is really needed is a resistor of the same resistance as the speaker (8 Ohms) supplied by UCI. If the UCI-supplied speaker is left connected, you cannot adjust volume below that of the UCI speaker as it is still working. (Note: you must have the speaker or an 8-Ohm resistor between PS pin 1 and 2 for the monitor's speaker to work). Audio input can also be run through the VIDEO cable, as the ZVM-135 has an audio input on its RGB plug. Audio Hookup through the video cable is the same as for highlighting above, only use pin 18 on the cable's monitor connector and pin 6 on the cable's RGB connector. The line from the EasyPC board to the Z-100 RGB connector is very easy to make.

1. Cut a piece of 20-gauge wire 12 inches long.
2. Solder a crimp contact on one end and a female pin on the other end.
3. Cut a piece of 20-gauge wire 4 inches long and solder a crimp contact on one end.
4. On the other end of the 4-inch wire, solder an 8 or 10 Ohm resistor.
5. Place the crimp contact on the 12-inch wire into pin hole 1 of a 2 position crimp-type connector (same type of connector as used by UCI).
6. Place the 4-inch wire crimp contact into pin hole 2 of the same connector.
7. Strip a spot on 12-inch wire about 5 inches from crimp connector and solder the other end of the resistor to the 12-inch wire.
8. Cover the bare wire around the resistor with tape or heat shrink.
9. Insert the female pin on 12-inch wire into the Z-100 RGB connector pin hole 6 until seated.
10. Connect new PS plug to audio output on the EasyPC Video board. (Note: I use a 1/4 watt 10-Ohm resistor.)
11. If you want to hide the wires running along the video cable, use a piece of spiral wrap (same type as the kids use on bicycle brake lines).

Mouse

UCI has developed a driver for a MOUSE through the serial ports (J1 and J2). The driver software is priced at $10.00 and supports PC-MOUSE of Mouse Systems, MICROSOFT's serial MOUSE, and MANAGER MOUSE of Torrington Company. The UCI drivers are interrupt driven for maximum speed; as a result, the number 15 pin of the ICs that fit into U242 and U243 on the Z-100 motherboard must be bent out so they are not connected into the sockets. This will not affect normal operations of the computer. However, if you are using ACCESS for your communications program in the Z-100 mode, it will not run with these pins out of the socket. The good news is that PROCOM ver 2.1 will run in the terminal mode with these pins out; as a result, the Z-100 can receive color and graphics through the modem just like the IBM-PC. Some soldering may be needed to reconnect the mouse's cable to proper pins on the mouse's serial plug in order to work through the J1 port on the Z-100, (pin 3 of PC-Mouse must go to pin 2 of J1 on the Z-100). I have received my driver from UCI, and PC-MOUSE from Mouse Systems. The mouse works with LOTUS 123, SPELLBINDER, and several other programs. This will be a great enhancement to the Z-100 if you like working with graphics.

Communications

There is a problem with the interface of IBM communication programs and the Z-100 serial COM ports (J1&2). The problem has to do with the handshaking requirements of the IBM programs. As a result, most, if not all, of the IBM communications programs will not run. It may be possible to patch many of these programs so they will run on EasyPC. UCI has worked this problem out, however, and has an additional S-100 communication board (EasyIO) that solves this problem; it should be ready during June this year. There is an advantage to the EasyIO board in that it will have two additional COM ports fully IBM-PC compatible, one IBM-PC game port, and a real-time clock that will work in both Z-100 and IBM modes. Cost should be around $100, which is less than the cost of a real-time clock (ZCLK Board priced at $119.95) for the Z-100 without EasyPC.

The serial ports (J1&J2) work normally in the Z-100 mode, and all programs and hardware operate as if EasyPC were not installed. I have a Novation Professional 2400 modem connected to J2 and have run my Daisywriter 2000 through J1 with no problems in the Z-100 mode. The advantage to this board is that I will gain two more COM ports, which I need with a serial mouse installed.

V20 Processor

The new V-20 processor from NEC will run on the new releases of EasyPC; ROM 1.62 has a fix which allows this new high-speed processor to operate. Using the V-20 in the EasyPC gives a very fast, effective clock speed to the Z-100 in both modes — (25.51) in IBM mode and (24.09) in Z-100 mode of operation. (Note: this is using a math looping program that takes 10 seconds to complete on an 8088 at 4.77 Mhz; this program takes 1.87 seconds on EasyPC in the IBM mode with the V-20 installed. The same program takes 2.42 seconds on an IBM-AT. I also tried a large worksheet in LOTUS 123; it ran through the calculations at least 3 times faster than a normal 8088-2.) If you have an early release of the EasyPC board, you will need a 68 pf capacitor across pins 20 and 25 on the processor (installed at UCI on all new boards). The best spot to place this 68 pf capacitor is on the same side of the EasyPC System board where all the other components are located.

1. Set the EasyPC system board on a table with the component side up and facing you as if it were in the Z-100 with you sitting at the keyboard (all writing on the board will be right side up).
2. Locate the two rows of 20 foil trace through holes between S2 and S3 (these look like two rows of 20 solder-filled pin holes); pin 20 is located at the left end of 1st row closest to S2 (left end is close to RP1), and pin 25 is in the 2nd row of through holes, 5 solder dots in from the left side.
3. Solder a 68 pf capacitor across these points.
4. Install new ROM 1.62, and reinstall the EasyPC system board in the Z-100.

UCI will also update the system board if you ship it to them; turnaround is very fast.

As you may know, the V-20 processor uses CMOS technology which has very low power consumption; therefore, the V-20 generates less than 1/3 the heat of the 8088 while operating. Although the V-20 operates at the same 5Mhz or 8Mhz as the 8088, the V-20 is faster than the 8088 because the V-20 takes fewer clock cycles to do the same operation; therefore, its processing throughput is faster. The V-50 should also run in the EasyPC board when it arrives in the States from Japan. Rumor is that the V-50 is a combination of the V-20 and the 8087, all in one chip that plugs into the 8088 socket. The Z-100, EasyPC, and the V-50 (if this rumor is true) should prove to be a very interesting combination with programs such as Lotus 123 Ver 2.0 and other programs that use the super-high-speed math processor.

Other Boards

The UCI S-100 expandable RAM memory board (Easy Memory board) will operate with the EasyPC at 5Mhz or 8Mhz. This would push a 768k Z-100 to around 2.75 Mb of RAM, priced at $395 with no memory chips installed, it also includes RAM drivers for both Zenith and IBM operation. If speed is what you need, this is the way to go, as it is much faster than either a floppy or a hard drive. UCI also has a battery backup (RAM saver) for their Easy Memory board so you will not lose data stored in the RAM drive when power to the computer is off or interrupted. This upgrade to the Easy Memory board is $189.

For those of you who need 8087 support, UCI has released Easy87, which is the 8087 upgrade to EasyPC. This is a very neat package that connects to the 8088 socket (U211) on the EasyPC system board after removal of the 8088 or V-20 processor; removed processor is reinstalled on the Easy87 board. Easy87 is a small board with all the requirements for 8087 support that connects to the EasyPC system board via ribbon cable. The Easy87 board mounts on the side of the power supply box. The advantage to this is that it is a cooler place and keeps the heat, generated by the 8088 and the 8087, away from Z-100 motherboard and the EasyPC system board. Easy87 board works fine with either the V-20 or 8088 processor installed. This board is about $80.00 without the 8087 math processor installed.

EasyPC will function with any of the S-100 boards that normally run in the Z-100, as long as the S-100 board's port address is not in conflict with the port addresses used by EasyPC. An example of this is the U.S. Robotics S-100 modem. I have not been able to get this internal modem to work in the Z-100 or IBM mode with EasyPC installed in the Z-100. Although I have tried several different port addresses, none of them work; all I get for my effort is a wild interrupt, and Z-100 locks up. I plan on sending the modem and software (ACCESS) to UCI and letting them look into this problem; more than likely, the solution is a simple one.

Other boards that plug into the Z-100 motherboard are usually in conflict with the EasyPC System board (ZCLK and the various 8087 piggyback boards that plug into the 8088 socket on the Z-100 motherboard are examples) and cannot be installed in conjunction with EasyPC. Some speed-up kits are producing problems because the chips on the motherboard are not fast enough (check with UCI if you have any questions).

Operating The EasyPC

Once you have the EasyPC installed in your Z-100 and you learn the new keys for the IBM mode (key markers supplied), it is hard to tell the difference between the Z-100 and the IBM-PC, except for the speed (EasyPC is much faster). All operations are just like an IBM-PC when in the IBM mode, including disk operations. EasyPC will read all IBM program protection formats, including ones that use the tracks outside of the Z-100 range. Try this with a software or firmware emulator. If you have a hard disk installed, you will not notice anything different in its operation once it is set up in accordance with the instructions supplied from UCI or Zenith. The drives in the IBM mode are "A & B" for floppy and "C" for hard drive; this makes it very easy to use installation programs written for the IBM-PC hard disk.

The Z-100 mode operates as a normal Z-100 without EasyPC installed. All drive designations remain the same, as does the processing speed (unless a V-20 is installed). I have not been able to access the IBM partition from the Z-100 mode or vice versa when in the IBM mode, but I did not put a lot of effort into this. Access across partitions may be possible; however, I fail to see any real advantage except for data files, as most programs stored in these partitions will only cause problems if run in the wrong mode. If data files are kept on floppy disks and a RAM drive is used when fast access speed to data is required, access across partitions is not a problem, and inadvertent calls to the wrong programs are avoided.

All memory resident programs that I have tried in the IBM mode have worked fine, including SideKick, although SideKick must be loaded last, in most cases, to avoid conflicts with other memory resident programs, the same as on the IBM-PC.

Conclusion

I have not found an IBM program (other than one game written in BASIC) that will not run on the EasyPC, and these programs also run faster than they will on an IBM PC. I guess the question is, how do I compare EasyPC to the Gemini board? Well, in my opinion, there is no comparison; EasyPC is the winner by a large margin. Comparing the Gemini board to EasyPC is like comparing a Volkswagon to a Porsche. If you want full color, access to all the disk tracks of some IBM copy protection methods, and speed, EasyPC is the only way to go.

The Future

I plan to upgrade my Z-118 with two 20 MB or 30 MB half-height, hard-disk drives soon, as the Z-217 Winchester controller board and the EasyPC board are said to handle up to 4 hard drives. I plan on installing both half-height, hard-disk drives internally. Another area I plan on looking into is installing a second floppy disk drive (Drive B) above the present half-height floppy drive (Drive A) in my Z-118, as I find it very inconvenient having to copy from Drive A to hard disk and then back to a blank floppy diskette in Drive A when making back-up copies of software or data not on my hard disk. I expect this upgrade to cost around $100.00 for floppy drive and cables, and about $350.00 for each hard drive (20 MB) and cables. If there is any interest on this subject from fellow REMark readers, I will be happy to write up my findings. I also solicit any constructive comments on this article or any new findings that other HUG members may have on the EasyPC board.

*

A Winchester For The '89 Part Seven

Peter Ruber
P.O. Box 502
Oakdale, NY 11769

The Ampro/Heath CP/M Computer

Interesting things sometimes come in small packages. And if you're looking for alternative solutions to upgrading your H/Z-89 computer in a very economical manner, the next two articles will offer enhancements that may surprise you.

The person responsible for these articles is Henry Fale of Quikdata Computer Services. Most readers of this series will recall my evaluation of Quikdata's QUIKSTOR Winchester software in Part Three. Several months ago, I received a call from William Dollar, president of Ampro Computers. He had been speaking with Henry Fale earlier that day, and Henry suggested that I would probably be interested in looking at the new CP/M and PCDOS single board computers with the SCSI hard disk interface.

After a brief exchange of letters, the boards arrived. A quick review of the documentation convinced me that I had some impressive hardware that had to be written about in this series. For those of you who aren't familiar with Ampro Computers, they began to market a product several years ago called the "Little Board". This was a complete CP/M computer on a board no larger than the standard 5.25" floppy drive. In fact, it was designed to mount directly on a floppy drive, from which it derived its power, and included the necessary circuitry to control up to four drives of various configurations.

Recently, the original "Little Board" computer came out in an enhanced version for only $289, which included an SCSI hard disk interface. And a short while later, Ampro issued a PCDOS version using an 80186 8-MHz CPU, with SCSI interface and 128k RAM (expandable to 1-Megabyte), for $495. Because of the diversified software and hardware complexity of both boards, we will devote a separate article to each system.

Connecting The Hardware

If you're wondering how all of this relates to the H/Z-89, bear in mind that the CP/M Little Board/PLUS computer was designed to operate with any RS-232C terminal. Since the '89 was an outgrowth of the H-19 terminal, and because it communicates serially with the '89's CPU Logic Board, it is a simple matter to convert the '89 into a full-featured terminal by disconnecting the Molex cable plugged into P404 on the back of the TLB. In order to complete the external communications ability of the TLB, you then take the 134-1070 DTE connector from P605 of your H-88-3 Serial I/O card and plug it into P404 on the TLB. While you're at it, you might as well disconnect the power supply plug at P516 on the top right of the CPU board in order to reduce the drain on the power supply.

Doing this will not affect the proper operation of the TLB because the required voltages and sync pulses used by the TLB are derived from P514 and P515 on the left side of the CPU board. This is to say that the signal and power lines from the transformer and the video board are supplied to P514 and then fed to P515, which in turn transports them to P401 on the TLB.

Mating the '89's Terminal Logic Board to the Ampro CP/M-PLUS computer board requires a six-wire cable between the DTE connector and J3 on the Little Board, which is nicely detailed in the Ampro Technical Manual. While the board itself uses keyed Molex shell connectors, you can cut down any long connector and make do with it. Additionally, all you need is a six-wire ribbon cable, six miniature shoe clips that will have to be soldered to one end of the cable, and one DB-25 female connector. I used a Radio Shack D Subminiature Flat Cable Connector (#276-1565), which had metal teeth that would ordinarily grip a 25-wire ribbon cable and simply soldered the other end of the six-wire cable to it.

Here is a diagram of the terminal connector:

Ampro Pins	Signal Name	Function	In/Out	H-89/ DTE Pins
1	Ground	Protective Ground	--	1
2	Ground	Signal Ground	--	7
3	TxD	Data Output	Out	3
4	HSO	Handshake Out (RTS)	Out	5
5	RxD	Data Input	In	2
6	HSI	Handshake In (CTS)	In	20

If you're planning to mount the CP/M board inside your H-37 drive cabinet, you will need to clamp on a 34-pin edge card connector to the drive cable and a standard floppy power "Y" connector to provide the +5 and +12 volt power to the Ampro computer.

Since I don't have any internal floppy drives, I mounted the Ampro board inside the cavity where the drive used to be, using right angle brackets secured to the inside of the cover plate. (See Photo 1) I connected the 4-wire floppy drive plug to the board and then the edge card connector from my hard sector controller. That enabled me to connect either 48tpi or 96tpi drives to the back panel of the '89.

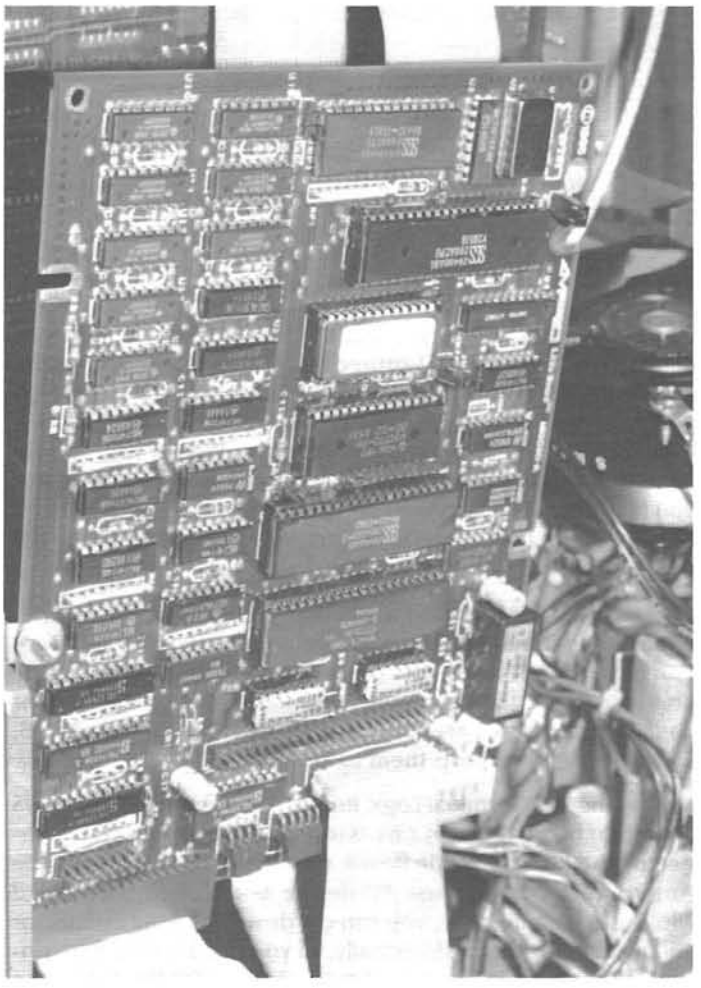


Photo 1

This is a quick and dirty installation of the Ampro CP/M Little Board/PLUS computer mounted inside the '89 where the internal disk drive used to be. See text for interfacing procedures.

Additional connectors on the Ampro board include a second serial port for hooking up a modem or a serial printer; a parallel printer port; an SCSI hard disk interface connector; a user ID connector, if you plan to hook a group of Little Boards together around your house or office in a networking arrangement; and a dual purpose 4-pin that will allow you to connect an LED to monitor the board and install a Reset switch.

Computing Power

When you first look at the Ampro CP/M Little Board/PLUS, you wonder where the ICs are, because it only contains 27 chips, not counting the 8/4164 DRAMS. The effect is unimpressive until you begin to study the specification pages in the Technical Manual.

But it is the board level equivalent in terms of functions and features to:

- The H/Z-89 CPU Board
- The H-88-3 Serial I/O, together with a Parallel Card (or the Z-89-11 Multi I/O card)
- The Magnolia 77320 SASI hard disk host adaptor card
- The H-88-1 hard sector disk controller
- The Z-37 soft sector disk controller

plus other features not found on the '89 — all on a card measuring 5.75" × 7.75".

Fast And Powerful

The Ampro CP/M Little Board/PLUS is controlled by a 4-MHz Z-80A CPU. All system functions are based on a single 16-MHz master clock. A Z-80 Counter Timer Circuit (CTC) provides four programmable counter or timer channels. Two of the CTC channels provide the baud rate clocks used by the two serial I/O ports. The other two CTC channels are available for use as programmable timers in applications programs or for real-time clock functions.

A Z-80 Dual Asynchronous Receiver/Transmitter (DART) provides two fully programmable serial I/O ports. Each channel has four of the standard RS-232C signals: TxD, RxD, RTS and CTS. These signals are sufficient for interfacing most serial printers, modems and terminals. In those cases where other interface signals are required for one of the serial ports, handshaking signals can be borrowed from the second port (if not needed by that port).

Programmable baud rate clocks are supplied by the CTC for baud rates up to 9600. Additional circuitry provides baud rates of 19.2K and 38.4K baud for Port A only. Using the Watzman or other TLB ROMS supporting these higher baud rates should provide you with a highly responsive terminal/computer relationship. If you plan to use the '89's Terminal Logic Board at baud rates higher than 9600, the video tearing you see when text scrolls up the screen becomes more pronounced. Technical Micro System, Inc. (P.O. Box 7227, Ann Arbor, MI 48102) has an inexpensive "Flicker-Free" modification kit that eliminates this problem.

The parallel printer port provides the 10 essential signals of a Centronics-type printer interface: Data Bits 1-8, Data Strobe and Busy.

A Western Digital 1772 floppy disk controller provides all of the functions required to interface with standard 5.25" drives and most 3-4" micro drives. The 1772 includes within a single LSI device digital phase locked loop, digital write precompensation, motor on start/stop delay and software controlled step rates. It will support 4 floppy drives of any configuration, including 40 and 80 track single-sided and 40 and 80 track double-sided drives. It will also support sector sizes of 128, 256, 512 or 1024 bytes.

The SCSI hard disk interface is implemented through the NCR 5380 SCSI bus controller IC (the same as used on C.D.R.'s Super RAM 89). It fully supports the ANSC X3T9.2 standards, but will

additionally allow the connection of up to 64 target devices instead of the standard 8 devices.

The Ampro CP/M SCSI bus can control up to 8 hard disk drive controllers, and a total of 11 Winchester drives. The capacity of each drive can be of any size from 5-MB to 88-MB. That's nearly 1-Gigabyte of on-line hard disk storage.

How The Software Works

If you find the hardware specifications impressive not only in terms of the $289 price, but the power you get with just a handful of parts, the software is equally impressive and is easily worth more than the price of the board.

To make your Ampro/Heath 89 computer operational, you will need one or more double-sided floppy drives (either 48 or 96-tpi) capable of a 6 mS step rate.

When you first turn on your terminal and disk drives, you will notice that the LED in drive A: goes on. This is a signal that everything is ready. You load the Ampro CP/M distribution disk and close the drive's latch or door. The Boot ROM does everything else without intervention: it mounts the disk, loads the operating system into memory, and presents you with a Menu Screen of various functions.

Our first order of business is to make several backup copies of the distribution disk and create a working system volume. This is easily accomplished with either a single or double drive system, although the former will require the usual disk switching routine. If you booted from a 96-tpi DS/DD drive (the Ampro CP/M 2.2 media comes on a 48-tpi DS/DD disk), you will only be able to read from, but not write to, the 48-tpi format. So, if you have both types of drives, it might be advisable to create separate system volume for each format.

The format command is called AMPRODSK, which is accessed from the main Menu Screen (as are a dozen other often used commands, such as SYSGEN, CONFIG, DIR, ERA, STAT, TYPE) by typing the number 1 and following the instructions. The screen displays the track number and disk side as it formats. The same applies to SYSGEN. To then backup your Ampro CP/M system volume, you then type the letter "C" from the Menu, and then the wild card designation of *.*. This procedure takes a few minutes because each file that is being copied is also verified, which provides you with a means of testing the media integrity of your disk.

You also receive a Public Domain program for terminal communications called MDM740 which has been enhanced by Ampro. Typing the letter "X" from the Main Menu activates the program. Like everything else, it is completely menu driven using mostly one-letter (or sometimes a three-letter) commands. It supports full up and downloading and saving to disk and printer, plus everything else you expect from an intelligent terminal program.

The Ampro CP/M distribution disk is packed with nearly five-dozen files. You receive the standard CP/M 2.2 utilities, the Ampro BIOS, the ZCPR3 enhancement package, plus a host of special utilities written by Ampro to make life easier for you. As one who has had his ups and downs with CP/M over the years, this is, by far, the finest implementation I have ever used. In addition, you can create your own Menu Screens with one of the ZCPR3 "shell" programs to assign one-letter commands for any frequently used programs or utilities you wish to invoke from the Main Menu rather than entering the Manual CP/M command and typing the filename.

The Ampro User Manual is only a small step below the excellent standards created by Heath/Zenith over the years, and certainly a model of clarity that some H/Z support vendors should emulate. All CP/M, ZCPR3 and Ampro utilities are covered with as much detail as is required for you to understand their purpose. It is intelligently written so beginners will feel comfortable using CP/M. Old-timers of CP/M won't be left out because there are enough technical details to keep them happy for many weeks.

The important feature of the Ampro CP/M software that I was anxious to check out was the MULTIDSK Version 2 utility, because it concerns the compatibility factor between this system integration, the expensive software sitting on your shelves, and the data files you have built up over the years. If this compatible relationship did not exist, there would not have been much purpose in reviewing this product.

Fortunately, Ampro designed a great deal of CP/M compatibility into their software implementation. Among the drive formats it will support are those of the Actrix, DEC VT180, IBM (CP/M-86), Kaypro II, Lobo Max80, Morrow MD2, NEC PC8001A, Osborne 1 and 2, TI Pro (CP/M-86), TRS80-1-3-4, Xerox 820 (I & II), Kaypro 4/10, Morrow MD3, PMC-101, Sanyo MBC 1000/1100, Televideo 802/803, DEC Rainbow (CP/M), Eagle IIE-2 — and every Heath/Zenith format you may have, including the reading and formatting of Z-100 48-tpi SS and DS CP/M-85 disks. You cannot use CP/M-85 on the Little Board, nor the IBM CP/M-86 operating system, but all data files created on those system disks can be transferred to Ampro's CP/M and worked with if you have the appropriate program software — i.e. word processing, database, spreadsheet, etc.

The following lists the Heath/Zenith specific formats that are readable with the MULTIDSK utility:

SS 48-tpi Menu:	H/Z 100
	H/Z 89 SD
	H/Z 89 DD
	H/Z 89 XD
	H/Z w/Magnolia
DS 48-tpi Menu:	H/Z 100
	H/Z 89 DD
	H/Z 89 XD
96-tpi Menu:	H/Z 89 SSDD
	H/Z 89 DSDD
	H/Z 89 SSXD
	H/Z 89 DSXD

A utility called MULTIFMT will allow you to format disks in a variety of Heath/Zenith '89 and '100 formats. This creates mobility in transferring your Text or Data files between the Ampro CP/M computer and the H/Z-89 and '100. If you have an oddball CP/M computer at the office, the RESET utility will allow you to establish disk track and sector parameters for the target drive, so you can use data disks from that system on the Ampro. This program is quite complicated. A thorough reading of the documentation and a detailed knowledge of the foreign disk format is necessary before you attempt to use this program.

Not wanting the user to be restricted to merely using files from other CP/M computers, the DOS and DOSMFT programs will allow you to read, copy and format MSDOS compatible 8 or 9 sector/track disks (either single or double-sided) if you have a Z-100 or Z-150 PC series computer (or an IBM clone). While I haven't used this feature a great deal, I did a directory check of

for the Z-151/161

WILDFIRE

from Software Wizardry

Speed The ultimate speed-up solution for the Z-151 and Z-161.* With WILDFIRE your Z-151/161 will go faster than the 8 MHz Z-158.

Compatibility WILDFIRE suspasses processor chip additions for compatibility with the IBM PC and costs up to $600 less.

Featuring A daughter board, higher speed replacement chips for the processor board, a high/low speed switch, an LED indicator to mount on the front panel, and hardware reset switch.

Price Software Wizardry brings you WILDFIRE, with attractive front panel, complete installation instructions, and reliable daughter-board design for only $249.

Contact your dealer, or order direct from Software Wizardry by calling (314) 724-1738.

WILDFIRE

Software Wizardry

Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, MO 63301
(314) 724-1738.

IBM PC is a registered trademark of International Business Machines.
*WILDFIRE only works on Zenith Z-151 and Z-161.

S & K Technology Inc.
Quality Software for Heath/Zenith Microcomputers

WatchWord™
for the Z100

The ultimate in word processing with speed and power. See the reviews in Remark (Jul 1985) and Sextant (Jan-Feb 1985; Sep-Oct 1985). Requires 192K RAM.

Price: $100

Demo Disk: $ 3
(Includes the Resident Speller Demo.)

The Resident Speller™
for the Z100

A spelling checker for use with WatchWord.

Check your spelling as you enter text from WatchWord. Includes a stand-alone spelling checker for ASCII files.

50,000 word expandable dictionary, requires 192K RAM (300K for use with WatchWord).

Price: $100

(Demo included on WatchWord Demo Disk.)

The Resident Speller™
PC Version for the Z150

Check your spelling as you type from most word processors.

Includes a stand-alone spelling checker for ASCII files.

50,000 word expandable dictionary. Requires 90K in addition to memory required by your word processor.

Price: $99

Demo Disk: $ 2

Texas Residents add state sales tax

S & K Technology, Inc., 4610 Spotted Oak Woods, San Antonio, TX 78249, (512) 492-3384

about 30 MSDOS program disks and then verified them on a Z-151 PC. When you read the directory of an MSDOS disk, you only see the actual program and data files. MSDOS or PCDOS .COM files will not show up on the screen.

One of the interesting features I found on the CP/M distribution disk is that the command format when you log onto a specific drive is A0-15 or B0-15, etc. These are referred to as user specific partitions. Thus, if you are using your computer in an office situation where several people will be sharing the system, or even at home if several members share your enhanced Ampro/Heath computer, a numerical designation can be assigned either to each user, or the numerical designation can represent a type of data file.

For example, if you're in the habit of using correspondence disks in a numerical sequence, rather than maintaining a separate disk for each person you write to, you can prepare a list of how you wish to assign these partitions. To wit:

A0: Heath/Zenith Users' Group
A1: FBE Research Co.
A2: Quikdata Computer Services
A3: C.D.R.
A4: SigmaSoft & Systems
A5: Technical Micro Systems

and so forth. Generally, I use PIE for daily correspondence. It's quick and simple and allows me to use the screen as a typewriter. Most short articles are prepared on PIE, too, because magazines require unformatted ASCII text files for computerized typesetting. Articles longer than the 1000-line limit imposed by PIE are either broken up into separate files or I use the SPELLBINDER Office Management System.

So, when I produce my daily correspondence, I type PIE A0: HUG1.LTR to enter the HUG letter file partition so that the letter will be saved accordingly. Since I don't keep copies of letters, it used to be that I had to sift through a directory listing in excess of a hundred or more files on a 96-tpi disk in order to locate the numerical designation of the last letter written to a specific person. This involved a time consuming use of CTRL S/Q to stop and start the scrolling director screen.

Now I simply type DIR A0: or DIR B9: and receive a directory listing of only those files I need be concerned with. This is also ideal if you use your '89 as a billing system with separate disks for each month. A disk partition can be assigned to each account. The size of the partition is flexible according to how you use it. You aren't required to maintain equal size partitions for each numerical designation. It's just that you have the flexibility to use this feature as it is appropriate to your needs. Or ignore it entirely.

Low-Cost Upgrade

The greatest savings in considering the Ampro CP/M board will be realized by H/Z-89 owners who currently have only a single hard-sector drive. For an investment of $289 for the Little Board, plus another $250 for double-sided drives, you can put together (software included) an up-to-date system with a hard disk drive interface that will not force you to give up your current software.

If you own an H-19 terminal in conjunction with an H-8 computer, the addition of an Ampro CP/M Little Board/PLUS will give you H/Z-89 compatibility, plus all the other features. Going the conventional route of spending $300 for a soft-sector controller, plus $250 for a hard disk host adaptor card, $100 for a parallel card, and $250 for drives is nearly twice the cost.

Or, if you want H/Z-89 compatibility with the features and excellent keyboards offered by the newer Z-29, Z-39 and Z-49 terminals, you can put together your own system quickly.

Then, you can add the Ampro PCDOS 80186 CPU single board computer for some real fun. But you'll have to wait to learn about this board.

The SCSI Winchester Software

The hard disk partition and driver software is supplied on a separate disk. Because of the overhead required by the size of the Winchester drive you will be using, you must create a CP/M size through MOVCPM.COM (or ZMOVCPM.COM if you are planning to use the enhanced Console Command Processor software).

Unlike the Magnolia and Quikdata Winchester software which is hardware specific to the Xebec S1410 or S1410a controllers (or the C.D.R. SASISOFT partitioning software which is available for either the Xebec or the Adaptec ACB-4000 controllers), the Ampro SCSI bus interface can be used with all the above, as well as the Adaptec ACB-5000 and the Shugart 1610-4 controller, and the Xebec OWL, which is a relatively inexpensive integration of a 10-MB hard disk with an S1410a controller in a single unit.

The Ampro Z-80 hard disk software diskette includes the new SCSI BOOT EPROM code in Intel HEX format (SBT-1-0.HEX), which permits auto-booting directly from a hard disk drive if you are using the Xebec OWL, Adaptec or Shugart controllers.

The Xebec S1410 and S1410a controllers (and the DTC 510A and 510B, as well) are not self-initializing. They require the addition of installation-specific commands in either the SCSI BOOT EPROM or in the boot strap loader contained in the HGEN utility, in order to initialize the controller prior to drive access. Ampro will provide the source code for its BIOS and the SCSI BOOT EPROM for a nominal charge. This will allow you to select a controller at the best price advantage. The Shugart 1610-4 can often be purchased from Ampro for $125, while the Xebec and Adaptec units sell for $250-$290.

The simplistic usage I enjoyed with Ampro's CP/M has been carried over to the hard disk software implementation. First step is to make a system volume labeled "Hard Disk System", and construct a CP/M system size based on the supplied chart as it relates to the size of the hard disk drive you will be attaching to your system. Then, use SYSGEN to write this to your newly formatted diskette. Next, you use the CONFIG utility to set your port configuration and other initialization parameters. Test the new disk by booting on it.

After transferring all the hard disk utility software to your Hard Disk System Volume disk, and the standard utilities for transferring files, directory listings, etc., run the HFORMAT program to format the hard disk drive. Enter the specific data on your controller (brand and model number), the information on your hard disk drive (number of cylinders, number of heads, step rate, write precomp data, etc.), and the number and size of the partitions you wish to write on to the hard disk. You can create partitions of 1 to 8-MB in size based on your requirements, which you accomplish by assigning an alphabetical letter for each partition.

One of the utility programs you will appreciate is a Public Domain program called FINDBAD which, when you run it, will perform a read of every sector of each CP/M drive letter partition you specify, logging all errors encountered during the check. It

will log and group all bad sectors in a file called [UNUSED].BAD and lock them out. The drive will then be unable to read or write to these sectors, thus preserving the integrity of your data. If you erase any partition and then SYSGEN it at any future date, you must run FINDBAD again and re-identify the bad sectors on the partition.

If you choose to boot directly from a hard disk partition shown on a Menu screen, you will have to create a hard disk initialization alias with the ZCPR3 ALIAS utility, so that in effect the hard disk partitions become the primary boot drives and the floppy drives the secondary boot drives. During this process the drive ID's are swapped. For example, if you have installed a 20-MB Winchester with four 5-MB partitions identified as drives F: to I:, they will become drives A: to D:, and your actual and logical number of floppy drives will become F: to I:.

Documentation

A user's success in installing a new product is closely related to the quality of the documentation, and there is nothing skimpy about what you receive.

You get a 98-page System Software User's Manual, a 46-page Hard Disk Software User's Manual and a 68-page Little Board/PLUS Technical Manual, to which is appended 76 pages of component specification data on all the master ICs that control the operations of the board.

The first two manuals are presented in a HOW-TO (setting up and using the software, how to use the commands in creating specific system volumes) and WHAT'S-WHAT (a thorough explanation of all the related utility programs). For users who are being exposed to CP/M for the first time, text references are supplied and sources for expanded ZCPR3 utilities are given.

As stated earlier, the CP/M BIOS can be modified if you are an experienced programmer, and Ampro will provide the source code for a nominal charge. Ampro also maintains a technical hotline for those who may have difficulty with any software or hardware aspects of the system. They would also like to hear from anyone who may have discovered a "bug" of any sort.

There are also several Ampro User Group Bulletin Board Systems operating 24-hours a day in California, which can provide assistance, Public Domain software, and a general exchange of ideas relating to the Little Board series. One board is owned and operated by Ampro to disseminate information.

My favorite part of the documentation package was the Technical Manual. This presents full schematics, parts lists, theory of operation, block diagrams, details for interfacing all connectors, manufacturer part numbers for all connectors, plus the IC specification sheets.

My criteria in reviewing any product is that I must first like it. If I don't, I won't bother to work with it because I consider it an intrusion on my time. It is difficult to find fault with the Ampro CP/M Little Board, except that the manufacturer should include the connectors and shoe clips for the two serial ports and the LED/Reset post headers. These parts are not generally found in your local Radio Shack store and must be ordered by mail. And if you don't have a healthy supply of odd parts in your hobby box, you will experience a delay of 3-4 weeks before you can get your Little Board operational.

* * *

In addition to offering the CP/M Little Board/PLUS by itself, Ampro Computers has complete systems available with assorted drives and cases with all connectors terminating on the back plane for easy interfacing with your modified '89 or Zenith terminal.

The package I received also contained some product literature from several support firms who can supply assorted cases and hardware, LAN devices to network the Little Boards for office usage. The Little Boards are also available from a growing network of dealers, including some Zenith Data Systems Distributors who are packaging them as low-cost systems with Zenith terminals.

The Ampro CP/M Little Board/PLUS is warranted for 90 days. Thereafter, servicing is provided on a "board exchange" basis for $99 on a 24-hour turn-around basis. Although the boards are stamped out like cookie cutters in a Singapore factory, I have been informed that the failure rate after 90 days is less than 1%. Only the master ICs are socketed (CPU, SCSI Controller, DART, ROM and Disk Controller); RAM, TTL chips and passive components are soldered directly to the board, making it difficult for the user to troubleshoot his own board. Nor will it be offered as a kit by Digital Research Computers, Inc. as was the original board, because the NCR SCSI controller and Western Digital floppy controller chips are expensive to replace if damaged by static electricity.

For additional information and current pricing of Little Boards, please contact:

AMPRO COMPUTERS, INC.
67 East Evelyn Avenue
Mountain View, CA 94041
(415) 962-0230

Related Products

Jim Buszkiewicz
HUG Software Developer

NOTE: *The following information was gathered from vendors' material. The products have not been tested nor are they endorsed by HUG. We are not responsible for errors in descriptions or prices.*

ZPAY Payroll Systems announces the release of ZPAYII, a new and simple to use payroll system, complete with an on-line help system.

ZPAYII is for those who don't really need the complications of a payroll package that must or can be integrated with a general ledger system. ZPAYII is a "stand alone system" with easy to read reports, simple to use menu driven functions, and on-line help facilities to guide you along the way. It is designed for those who would like to have control over their own payroll. No accounting experience or specialized training is necessary to use this package.

ZPAYII will handle different pay periods by your selection. It will handle salaried, hourly, commissioned, and salary plus commission employees, as well as 1099 employees. ZPAYII will allow you to have two overtime rates, handle bonuses, six additional pay deductions in addition to the normal federal tax, FICA, state tax, local tax, and an SDI type tax. ZPAYII will also record and track your vacation pay, as well as cost accounting and allows title assignment for up to ten different department accounts. ZPAYII computes the payroll and prints many different types of reports. It remembers when each month or quarter starts and automatically posts the necessary information. ZPAYII uses an encripted data file format and has three levels of password protection to keep prying eyes and fingers from confidential information. The number of employees ZPAYII can handle is only limited by the disk size. It has a built-in tax table editor so you may change your own tax tables when necessary. With this feature, there will be no costly updates or programmers required. ZPAYII will run on any of the H/Z-100 PC/200 PC computer systems, and requires at least 2, 5-1/4" drives, 192k of RAM, and any type of printer. ZPAYII is available for $39.95 from: ZPAY Payroll Systems, 3516 Ruby Street, Franklin Park, IL 60131, (312) 671-3364.

Blue Chip Software announces, the "American Dream", management simulation software. This 'game' is intended to familiarize the user with the broad goals of corporate management. It focuses on the coordination of interdepartmental operations of a large company. The player starts the simulation as the C.E.O. (corporate executive officer) of a manufacturing company and must make decisions regarding the performance of several operating departments. Those departments include sales & marketing, finance, research & development, manufacturing engineering and production. The goal of the simulation is to build the company to a $100 million plus business through sound financial, marketing and manufacturing decision making. Alternative strategies, such as low cost, high volume or high cost, low volume may be appropriate given market conditions and corporate capabilities. The simulation allows the user to adjust interest rates, inflation rates and other significant business factors in order to develop an understanding of impact of these factors on business operations.

This product is targeted toward middle managers and supervisors. The company simulated in "American Dream", is a mature company in a mature marketplace. It is this unique perspective that makes this product ideal for those who are involved in middle management of large companies. "American Dream" is a product of Blue Chip Software, 6744 Eton Ave., Canoga Park, CA 91303, (818) 346-0730, and retails for $124.95.

Continued from Page 10

After examining and testing the two products now on the market, we chose to install the EasyPC emulation package from UCI Corporation. To date, I have installed 27 of the units and would like to report my observations. Installation has been in Z-100s ranging in age from 3 months to over 2 years. All are dual floppy drive machines with memory configurations ranging from 192K to 768K, operating at 5 and 8 mhz.

First of all, we have yet to find any software designed for the IBM PC that will not run on the EasyPC. Additionally, there seems to be no loss of performance in speed or graphics when compared with the same software running on a Zenith PC. Of even greater importance is that staff and students using the computers don't have to do anything except indicate the mode which matches their software. In fact, the EasyPC allows a hardware switch setting which specifies a default mode. In our case, if no mode is chosen, the Z-100 boots as a Z-100.

The boards are easy to install. Instructions are well written and well illustrated. Of the 27 units I have installed, I have had some performance difficulty with about five. In each case, the support staff at UCI Corp. was extremely helpful in understanding the symptoms, pinpointing potential problems, and in ALL cases, identifying an adjustment or weak component in the Z-100. For instance, one Z-100 required a new 8088 processor when, after operating for over an hour, the computer experienced difficulty in the IBM mode. With some older Z-100s and monitors, video signal strength problems in the Z-100 mode were corrected with adjustments to the Z-100 video board contrast and signal ports.

Because of the number of EasyPC units I have in stock, I have had the luxury of swapping EasyPC components when a problem arose. So what may have appeared on the surface to be a problem with the EasyPC, did not leave the installation as I tried other identical boards. Of the 27 units installed (3 boards each), I have 2 boards which appear faulty and are being replaced by UCI.

Calls for assistance to the UCI toll free number can only be characterized as courteous, interested, and (most important) resulting in problem resolution. I am impressed with the company and the product.

We think the Z-100 is a wonderful computer. Its performance, reliability, and ease of service have been critical to its successful use as the foundation of our computer labs. I estimate that the addition of the EasyPC emulator has lengthened its software support life by at least two years — making this an extremely good investment.

Sincerely,

Timothy J. Donovan
9 George Street
Montpelier, UT 05602

Continued on Page 83

Continued from Page 63

DATA BASE MANAGEMENT SYSTEMS

HDOS

Part Number	Description of Product	Selling Price	Vol. Issue
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

885-6008-37	MSDOS NAVPROG	20.00	73
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
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

MSDOS

885-8038-37	MSDOS RFCAD Ver. 3.50	30.00	73

COMMUNICATION

HDOS

885-1122-[37]	HDOS MicroNET Connection	16.00	37
885-8005	MAPLE (Modem Appl. Effector)	35.00	29

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
885-8012-[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
885-3033-37	MSDOS HUG MCP	40.00	71

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	
885-4006	REMark Vol. VI Issues 60-71	25.00	
885-4500	HUG Software Catalog	9.75	
885-4501	HUG Software Catalog Update #1	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
885-3036-37	MSDOS TREE-ID	20.00	77

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 82

What A Great Bunch

Dear HUG:

Last August you published my request for help with the reverse video problem I had on my H-89A. Boy, what a response I received. What a great bunch of people in HUG.

All of the replies I received zeroed in on the P101 connector and the BR. I had wired the BR direct to the transformer after reading the article in REMark, so I installed a new BR as recommended by many. In fact, Dr Dave Barbee, Pullman, WA sent me the proper BR to install. I also installed a ground wire from the video circuit board to a corner bolt of the transformer as recommended by Orlan Stone, Medford, OR (Zenith Field Service Bulletin FSB-Z89-070).

I went off-line and filled the CRT screen with reverse video — problem solved, no more "rolling bars". After about an hour with still no problem, I booted and ran a program that contained reverse video — the problem was still there. Back to off-line and no problem; back to the program and the problem.

The reverse video in the program was flickering as compared with the "rolling bars". The flickering was random rather than in a regular sequence of the rolling bars. I tried several programs.

If anyone has a suggestion for this flickering problem, I'd be very grateful to hear from them.

To all those who helped me solve the "rolling bar" problem, my sincere thanks.

Sincerely yours,

Bob Speidel
Box 95E, Route 1
Emmaus, PA 18049

*

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 885-2079