



REMark®

Volume 7, Issue 4 • April 1986

P/N 885-2075 Issue 75

Official magazine for users of



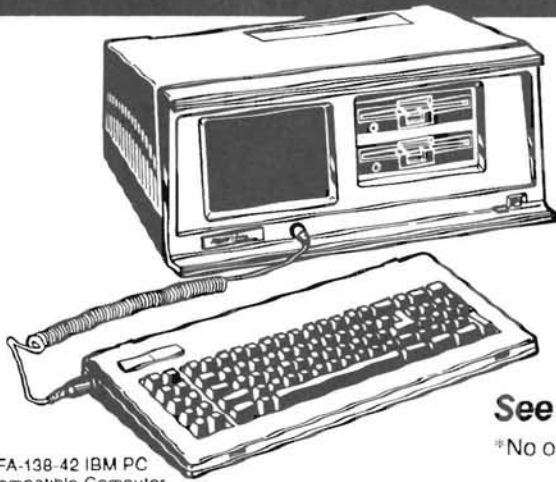
computer equipment.

Coming Soon!
A Z-100 PC Series by Joseph Katz

\$2.50



2 IBM PC Compatible Computers For Under \$1000 Each



ZFA-138-42 IBM PC Compatible Computer

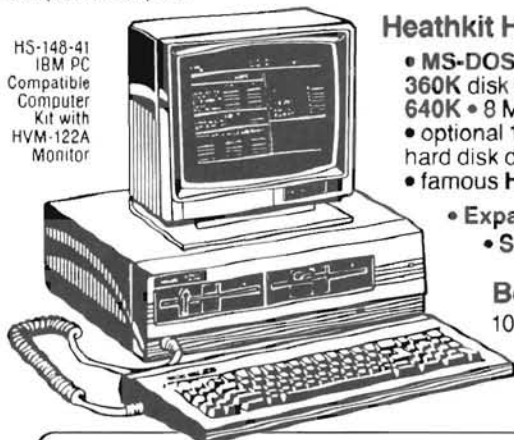
Zenith Data Systems Transportable Computer

• factory assembled • MS-DOS operating system • runs IBM software • dual 360K disk storage • 256K memory expandable to 640K • 8 MHz CPU • built-in monitor • both mono and color graphics • start-up diagnostic • famous Heath/Zenith quality

SPECIAL PRICE \$999⁰⁰*

50% off PC Savers available with purchase. See Heathkit catalog for details.

*No other discounts or special offers can be applied. Only items specified in this ad are included in the price.



HS-148-41 IBM PC Compatible Computer Kit with HVM-122A Monitor

Heathkit HS-148-41 Computer Kit

• MS-DOS operating system • runs IBM software • 360K disk storage • 256K memory expandable to 640K • 8 MHz CPU • both mono and color graphics • optional 10 Meg, 20 Meg and 30 Meg Winchester hard disk drive available (requires expansion card) • famous Heath/Zenith quality

- Expand to 640K Memory for just \$110.00
- Second Disk Drive just \$125.00

Bonus Buy

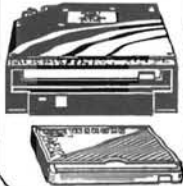
10 Meg Hard Disk Drive only \$499 with HS-148-41 (requires expansion card) or HS-151-21 purchase.

Installation included. Works with IBM PC and most compatibles. Hard Disk Drive \$599 if purchased separately.

KIT ONLY \$899⁰⁰*

Easy to Build. Assembles in just a few hours, without soldering. And backed by our promise "We won't let you fail."

SYQUEST 10 MEG Removable Winchester Cartridge Drives



Includes drive, one 10 Meg cartridge, controller board

SPECIAL \$1195.00
Reg \$1395.00

FREE Installation

ZOOM/MODEM PC 1200 INTERNAL 1200/300 BPS modem

for PC, XT and AT computers. The PC-132 is Hayes-compatible. Includes touch tone decoder, onboard RAM for programming, battery backup and more.

Reg \$489 **Special \$299.00**



IBM PC Compatible Computer Kit with 4 Expansion Slots

Has full IBM-PC compatibility, single 360K disk drive, expandability to 640K without an expansion card.

Second disk drive only \$125

\$1169.10* Includes free software worth \$720

HS-151-21 IBM-PC Compatible Computer Kit

NEW MASTERPIECE 4-Function Power Controller. Includes swivel base, 5 outlets, master switch, power surge protector, static ground **\$99.95**

Your Choice dBASE III or Framework II \$499

Monitor Kits from \$79.95

NASHUA DSDD Diskettes... 10 Per Box \$12.95

\$2500 CREDIT NO MONEY DOWN

We're giving qualified customers up to \$2500 of credit. Charge your purchase right on the spot. Ask for details.

ZENITH data systems
AUTHORIZED DEALER
Fine Electronics Since 1962



Call (616) 982-3614 for store locations nearest you.
Inquire about special corporate and government sales and service programs.



Prices good through May 15, 1986.
H2C-200

Your TOTAL SERVICE computer center
Service • Support • Software • Accessories • User Training • Competitive Prices

Heath® ZENITH®
Computers & Electronics

Units of Veritechnology Electronics Corporation

NOT PROTECTED

Genie™

Software Magic from Advanced Software Technologies



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.

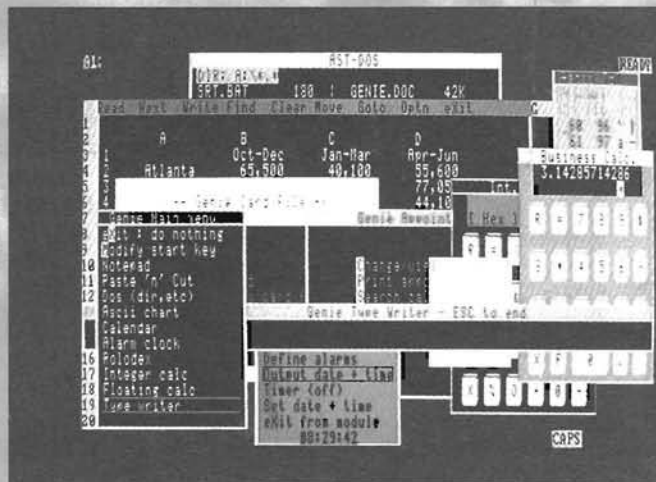
Here is what you get with Genie:

- **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)



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.

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 end <input type="checkbox"/>	shipping	4.00
M/C <input type="checkbox"/>	NY Tax (8.25%)	_____
VISA <input type="checkbox"/>	TOTAL	_____

Card Number _____
 Expires _____ Your Tel Num _____
 Name _____
 Address _____
 Company _____
 City/ST/Zip _____

Staff

Manager	Bob Ellerton (616) 982-3867
Software Engineer	Pat Swayne (616) 982-3463
Bulletin Board and Software Developer ...	Jim Buszkiewicz (616) 982-3463
Software Coordinator	Nancy Strunk (616) 982-3838
Secretary	Margaret Bacon (616) 982-3463
Editorial and Advertising Assistant	Lori Lerch (616) 982-3794
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

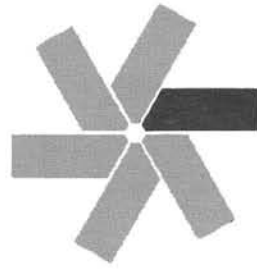
Although it is a policy to check material placed in REMark for accuracy, HUG offers no warranty, either expressed or implied, and is not responsible for any losses due to the use of any material in this magazine.

Articles submitted by users and published in REMark, which describe hardware modifications, are not supported by Heathkit Electronic Centers or Heath Technical Consultation.

HUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Heath equipment. As such, little or no evaluation of the programs or products advertised in REMark, the Software Catalog, or other HUG publications is performed by Heath Company, in general and HUG, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Heath Company, in general and HUG, in particular cannot be held responsible. The prospective user is, by virtue of obtaining and using these programs, assuming full risk for all consequences.

REMark is a registered trademark of the Heath/Zenith Users' Group, St. Joseph, Michigan.

Copyright (C) 1986, Heath/Zenith Users' Group



Buggin' HUG	7
COBOL Corner XVII	9
<i>H.W. Bauman</i>	
Midwest HUG Conference	13
Official International Heath/Zenith Registration Form	14
The Plight Of The Cockroach	15
<i>Jedariah (Jed) Borlowinski</i>	
ZPC Hardware Patch	17
<i>Pat Swayne</i>	
Heath/Zenith Related Products	23
<i>Jim Buszkiewicz</i>	
High Resolution Graphics With CP/M-85 The Final Chapter	25
<i>R.B. Owens</i>	
A Winchester For The '89 — Part Three	29
<i>Peter Ruber</i>	
Modifying The MS-DOS BIOS Of The H-160 To Use A 96-tpi Disk Drive	35
<i>Daniel Ng</i>	
Z-100 Drafting Program For Scientific Graphs	37
<i>Michael D. Chafetz, Ph.D.</i>	

HUG Price List	42
HUG New Products	43
CF	
A Configuration Utility For The H/Z-100	
<i>Jeff Kalis</i>	45
Patch Page	
<i>Pat Swayne</i>	51
Subdirectories And Command Pathing	
<i>William M. Adney</i>	53
A Report On The NEC V20 Microprocessor Chip For The "Z" Machine — Part I	
<i>Richard L. Mueller, Ph.D.</i>	63
Burn Baby, Burn! A Review Of The ID-4801 EPROM Programmer	
<i>Jim Buszkiewicz</i>	67
The WordStar Directory Changer	
<i>Joseph Katz</i>	69
G-PROM Review: Or How To Increase H-89 Graphic Resolution For Under \$20	
<i>R. Kenneth Strum</i>	71
PAINT.ASM — Part 2	
<i>Steven W. Vagts</i>	75

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
Analytical Products	44
Bromley Engineering	74
CCT	65
CDR Systems, Inc.	62
Computech	22
FBE Research Company	28
FINA Software	22
First Capitol Computer	50
Generic Computer Products	28
Heath Company	12,68
Intercontinental Microsystems	16
Intersecting Concepts	34
NewLine Software	73
NORCOM	74
Piiceon	82
S&K Technology	81
Secured Computer Systems	52
Software Toolworks	24
Software Wizardry	23,36
Studio Computers	66
UCI Corporation	6
Veritechnology Electronics Corp.	2
Barry Watzman	84
Woodrock	28

On The Cover: Pictured is a Heathkit ID-4801 EPROM Programmer. This reasonably priced kit will take about one evening to assemble. See article by Jim Buszkiewicz on Page 67.

Performance.

Introducing EasyPC™ from UCI. The most advanced IBM PC emulator you can buy for your Z-100. First to provide the compatibility, speed and audio support to make your Z-100 run like a PC. Since the EasyPC provides complete PC circuitry, you actually run PC programs on PC hardware. Quite simply, it's like installing a PC right inside your Z-100. In the PC mode, your Z-100 can run virtually the entire library of PC programs, including LOTUS 1-2-3, dBase III, FLIGHT SIMULATOR,

and most other copy-protected software. In the Zenith mode, your Z-100 is a Z-100, with no change in performance.

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

Quality.

You already know and appreciate the kind of quality and reliability that is built into your Z-100. UCI matches that commitment to quality from first step to last.

Sophisticated engineering and quality components combined with precision automatic assembly, component burn-in, and both in-circuit

and full functional testing, mean superior performance and long term reliability.

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

Availability.

EasyPC is ready when you are. If you've been waiting too long for PC compatibility, your wait is over. EasyPC is here now. Complete documentation is included for quick and easy installation,

so you'll be up and running fast. Your only problem will be which PC programs to get first.

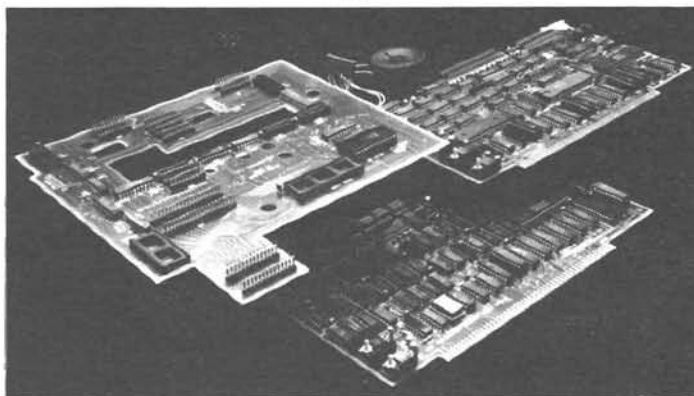
EasyPC is available through your independent Zenith dealer. Ask him for a demonstration.

Price.

At \$699, you'll pay a little more. But you'll get the best. Total compatibility. High speed performance. Unsurpassed quality and serviceability. In a word, value. You also get UCI's one year Solid

Service Guarantee and full customer support. If you have any questions or problems with any UCI product, or need the name of your nearest dealer, just call 800-UCI-COMPUTER.

Easy-PC™



Full IBM PC emulation for your Z-100.



CORPORATION
948 Cherry Street, Kent, Ohio 44230
(216) 673-5155 / 800-UCI-COMPUTER

EasyPC is a trademark of UCI, Inc. Zenith and Z-100 are trademarks of Zenith Electronics Corp. IBM and IBM PC are trademarks of International Business Machines. Lotus 1-2-3 is a trademark of Lotus Development Corp. dBase III is a trademark of Ashton-Tate. Flight Simulator is a trademark of Microsoft Corp.

BUGGIN' HUG

Speeding Up The 150 And Making Word Accept Escape Codes

Dear Mr. Ellerton:

As a relative newcomer to computers, I have found that simple little tasks become gargantuan, time-consuming, frustrating, hair-pulling, major projects for me.

In December, I bought a Zenith 150 and joined this users' club. I find your magazine very interesting and informative and am very impressed with the articles, but unfortunately, it seems that most of the articles and software is geared toward the Heath 89 or 100. It would be nice if more information was given about the 150, like "How to speed it up."

There is another, perhaps more important problem, however... After asking many people in the local users' club and not being able to solve this problem, I thought I would present it to you and maybe you could help. Here goes...

Microsoft Word Version 1 (the updated version has been ordered, but has not yet arrived) does not offer options to change the printer control codes (Escape codes) within the program. Therefore, everytime I want to compress or enlarge the type, I have to save the file, leave the program, get into BASIC, write the control codes, send them to the printer, THEN go back to the Word program, call up my file, and then send it to the printer. That is very time consuming and has to be done everytime I want to change the codes.

I think the problem could be solved by writing a BASIC program and somehow patching it into the Word program, but I don't even know how to write and save the BASIC program, let alone patch it into the Word program. And can a program that runs with BASIC run with MS-DOS?

HEI!!

Your prompt response to this will enhance my life. Thanks.

Sincerely,

Jackie Tobias
P.O. Box 696
Roosevelt, AZ 85545

Hardware involved: Zenith 150 and Citizen MSP 10 (Printer)
Software: Microsoft Word, Version 1

To H-89 Codonics Imaginator Users

Dear HUG,

I read with interest, Steven Sussman's letter in the February issue of REMark, concerning his inability to find a program which will dump the SigmaSoft graphics screen to his printer. Unfortunately, he is not alone with his problem, as Cleveland Codonics Imaginator users like me had the same dilemma.

I finally grew tired of looking for programs which ran under CP/M on my Codonics Imaginator equipped H-89, so I wrote them.

My programs are CP/M executable .COM files, originally written in assembly language, hence, are very compact (1 or 2k) and quite fast. I've decided to make them available, very inexpensively, to H-89 Codonics Imaginator users.

The first program dumps the graphics screen to an Epson (or compatible) printer with graftrax almost as fast as the printer can print. This is about 30 times as fast as an MBASIC program can do it.

The second program saves the full graphics screen to your disk. The screen is saved in binary, hence, it is about 1/10 the size of a comparable file written by MBASIC. It, too, is about 30 times as fast as an MBASIC program. These files can be read back from disk to repaint the graphics screen at will.

The third program is unique, and has proven to be invaluable to me. It enables the user to store up to 3 complete graphics screens in separate 'pages' of memory. This gives the user the ability to create, store and overlay graphics on the screen much like transparencies can be overlaid by an overhead projector. This resulting combination of graphic overlays can then be saved to your disk and/or dumped to the printer, as well.

I also compiled a character generator program which provides the user with the capacity to create alphanumeric characters on the graphic screen. The user provides simple x-y coordinates and the 'string' to be displayed, enabling him to label, number, or include text within the graphics display.

Although I originally wrote these programs for my own use, I believe simple, effective, and inexpensive software should be available to everyone. I will include these programs, a demo, and documentation on disk for \$19.00 to all readers who contact me. Specify hard- or soft-sector. Except for the public domain, you can't get more inexpensive than that.

Steve Leache
10 Cullinan Drive
Gaithersburg, MD 20878

Harper's June 1985 Article

Dear HUG:

I would like to provide some information and corrections related to Mr. Harper's June 1985 REMark article: Inside MBASIC Program Files. I've recently converted the cross-reference program (CRG.BAS) for use with ZBASIC, Rev. 1.0 (10/1/82) on the H/Z-100 computer. Please take note of the following items:

A. CGR.BAS Program Corrections

1. Line 10300: Change LD=FNUA!(AA\$) to LD=FNUA!(AA\$)
2. Line 10540: Change LF\$(HB\$)-LI\$+LE\$ to LF\$(HB)=LI\$+LE\$
3. Line 10380: Delete one of the three line 10440 numbers in the ON...GOTO instruction.

B. For use with ZBASIC

1. Set SA=2868
2. The screen control variables, i.e. CK\$, EA\$, CG\$, CI\$, and FNAB\$ must be deleted. Replace them with the appropriate LOCATE, SCREEN, and CLS commands.
3. To simulate the erase-to-end-of-page (EA\$) in line 10260, I used LINE(1,6*9)-(640,225),0,BF

To digress for a moment, I would like to inform your readers that ZBASIC and the new GW-BASIC do NOT support many of the

H/Z-100 screen control escape codes (found in Appendix B of the Z-100 User's Manual) such as: delete line, erase-to-the-end-of-page, etc. A previous article (Intro. To ZBASIC Part II — REMark #35) was in error when it stated some of the codes could be used. Only two codes are allowed, enable and disable the 25th line.

Overall, I prefer the output format of this cross-reference generator program to that of the one in the August 1984 issue of REMark. However, I believe the detailed explanation of how Mr. Dodgen's program accomplished its function was superior to the one provided for CRG.BAS. The intricate string manipulations performed by Mr. Harper's program makes necessary some step-wise discussion of how the program works. Otherwise, the article was informative and useful.

Yours very truly,

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

More Than The Usual Number Of Articles

Dear Pat:

I found the September issue of REMark contained more than the usual number of articles meaningful to my particular H-120 world, thanks particularly for the update on the 8MHz modification; I'm anticipating it for my older machine soon.

Fred Ormand's typewriter program filled a gap, but I couldn't make the ESC key work — presume a difference I haven't learned yet on my ZBASIC. I changed it to DELETE and it works fine. Also, made a few other changes; a copy is enclosed for H/Z-100s, if anybody's interested.

A question though on Vincent Alfieri's MS-DOS Amenuensis article. I, too, am lazy and have been writing one letter batch files for many things, but new to me in his article was writing such a file with DOS instead of EDLIN (which I find very easy), but I couldn't get it to work on my H-100. It does everything as Mr. Alfieri describes, but after CTRL-Z (nothing on F6) the H-100 prints the file lines on the screen and follows with the "files copied" message. Mr. Alfieri, of course, was writing for a PC machine, but is there a way of doing this on an H-100? I got lost in the manual searching this one out.

Thanks again.

Kindest regards,

Rodney E. Cavin, Director
P.O. Box 507
Altamonte Springs, FL 32701

```
10 REM "TYPE.BAS" by Rodney E. Cavin
20 REM in ZBASIC for H/Z-100s
30 REM From "TYPIST.BAS" by Fred T Ormand
40 REM in REMark, September 1985
50 LF$=CHR$(10)
60 CR$=CHR$(13)
70 ESS=CHR$(127)
80 CLS:LINE (140,5)-(450,30),,B
90 LOCATE 2,22:PRINT "The printer is now a typewriter"
100 LOCATE 3,26:PRINT "(Make sure it is ready)":PRINT:PRINT
110 X=CSRLIN
120 LOCATE 25,22:PRINT:CHR$(27)+"p",
" Press DELETE to exit program. ";CHR$(27)+"q"
130 LOCATE X,1,1
```

```
140 CH$=INPUT$(1)
150 IF CH$=ES$ THEN 220
160 PRINT CH$;
170 LPRINT CH$;
180 IF CH$<>CHR$ THEN 140
190 PRINT LF$;
200 LPRINT LF$;
210 GOTO 140
220 PRINT:PRINT:PRINT TAB(28);CHR$(27)+"p";
" END OF PROGRAM ";CHR$(27)+"q"
230 FOR P%=1 TO 4000:NEXT:CLS:SYSTEM
```

Note: If user desires to return to ZBASIC, line 230 should end with "END" instead of "SYSTEM", which returns the user to ZDOS.

A Bug In The MS-DOS Programmer's Utility Pack

Dear HUG:

I purchased the MS-DOS Programmer's Utility Pack several weeks ago and was quite pleased with the information and utilities provided. I have discovered a bug in the MDISK.ASM file, which occurs when creating a RAM disk of 512K or greater. The source index (SI) register is not compared to the proper value when determining the number of directory entries and the FAT sectors used with a given disk size. The bug occurs at the end of the MD__SIZE table in MDISK.ASM.

The current code in the MD__SIZE table is:

```
MD_SIZE DW 64
        DW 128
        DW 256
        DW 512
MD_SIZE_END = OFFSET $
```

This gives MD__SIZE__END the value of the offset \$36D, which is the location of the number of directories for a RAM disk of less than 64k. Since a RAM disk of 512k or greater causes the loop to continue until the source index register is equal to the value in the location \$36D, the values for the number of directory entries and the FAT sectors are in error. The maximum value for the source index register should be 8.

The following patch to MDISK.ASM will correct this problem:

```
MD_SIZE DW 64
        DW 128
        DW 256
        DW 512
MD_SIZE_END = OFFSET $ - OFFSET MD_SIZE
```

This now assigns the value of 8 to MD__SIZE__END and will jump out of the loop when the source index register reaches this value for RAM disks of 512k or greater. This will give the proper number of directory entries and the number of FAT sectors to use.

Also, the number of directory entries may be changed from 256 to 128 for RAM disks of 512k or greater in the MD__DIRS table which follows the MD__SIZE table. This will give you an additional 4k of usable memory on the RAM disk.

The documentation for MDISK.ASM in the Heath/Zenith manual is extremely lacking in information, but Pat Swayne's article "Using the MS-DOS Memory Disk" in the March 85 issue of REMark is very informative.

Sincerely Yours,

Ross A. Ziegenhorn
9059 Arborwood
San Antonio, TX 78250

Continued on Page 83

COBOL Corner XVII

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

Introduction

I hope that the "COBOL Corner" Users have carefully studied the last article. It has the information that is the background for what we will be doing over the next series of articles! In fact, you should have that article available as you read and use this article. In the "Real World" Tables are very important for most programs not only in COBOL, but in many other languages and software. They are sometimes called arrays. As you readers of "COBOL Corner" know, we believe in learning by doing! We will be using tables in most of the remaining programs we will be doing, and they will be getting quite complex. That complexity will not bother the User that works out each of the following programs and not just reads about them! We will start with a rather simple program in this article that will use a Serial Search with INDEXES and the SEARCH statement. The program will read a Personnel Record file from disk and print a Personnel Department Assignment Report. This report will require the program to use two "hard-coded" tables — Months and Department Names and Numbers — as references.

I am going to leave the Preliminary Phases for the COBOL User to do. We have done many of these together, so I would expect the User to know how to do them. PLEASE DO NOT try to do this program without these programming aids!! If you do not remember how to do any of the planning phases, refer back to some of your earlier projects. I will provide a portion of the coding to get the User started with the NEW items that this program will stress. Again, if you have been DOING the previous programs and not just reading about them, this coding will seem easy because you have LEARNED BY DOING! Use your previous project for reference if you need help.

Disk Input File (FILEL4)

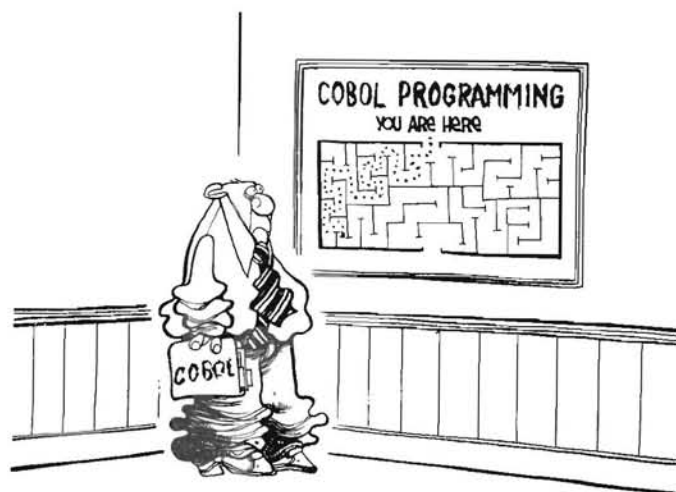
```

1--RECORD CODE (IF USED)-----2 CHARACTERS
2--FIELD NOT USED-----9 CHARACTERS
3--PERSONNEL NAMES FIELD-----18 CHARACTERS
4--FIELD NOT USED-----30 CHARACTERS
5--DEPARTMENT NUMBER FIELD-----4 CHARACTERS.
6--FIELD NOT USED-----17 CHARACTERS.

```

Hard-Coded Tables

I provided the formats for the MONTH and DEPARTMENT NUMBER and NAME Tables in the last "COBOL Corner" article. Refer back to this article as I will not redo them.



File Discussion

Remember your FILE SECTION must provide for the two files we will have. First, the Input Disk file and second, the Output Print Report file. The files will both be described in the WORKING-STORAGE SECTION. Also, the two "hard-coded" tables will be specified in the WORKING-STORAGE SECTION.

Special Serial Search Coding

WORKING-STORAGE SECTION

```

01 WS-SWITCHES
   05 WS-EOF-SWITCH          PIC X(03).      VALUE "YES"
      88 END-OF-FILE
   05 WS-ENTRY-FOUND-SW     PIC X(03)        VALUE "YES"
      88 TABLE-ENTRY-FOUND
01 ?S-REPORT-CONTROLS
   05 WS-PAGE-COUNT          PIC 9(03)        VALUE +1 COMP-3.
      88 FIRST-PAGE          VALUE +1.
   05 WS-LINES-USED          PIC 9(02)        VALUE ZERO
      88 FULL-PAGE           VALUE +40 THRU +43.
   05 WS-LINE-SPACING        PIC 9(02)
01 WS-DATE-AREA.
   (SEE DISCUSSION BELOW IN THIS ARTICLE)
01 WS-PROGRAM-TABLES
   05 MONTH-CONSTANTS
      (SEE LAST COBOL CORNER ARTICLE)
   05 DEPT-NBR-NAME-DATA.
      (SEE LAST COBOL CORNER ARTICLE)
01 PD-DEPT-NBR-PERSON-RECORD
   (SEE INPUT DISCUSSION ABOVE IN THIS ARTICLE)
01 HEADING-LINES.
   (USE ANY IDEAS YOU HAVE THAT WOULD MAKE A GOOD REPORT)
01 OL-OUTPUT-LINE (FOR 132 COLUMN REPORT)
   05 FILLER                  PIC X(39)      VALUE SPACES
   05 OL-PERSON-NAME           PIC X(18)
   05 FILLER                  PIC X(09)      VALUE SPACES
   05 OL-DEPT-NBR              PIC 9(04)
   05 FILLER                  PIC X(09)      VALUE SPACES

```

```

05 OL-DEPT-NAME      PIC X(17)
05 FILLER            PIC X(36) VALUE SPACES.

```

PROCEDURE DIVISION

200-PROCESS-DEPT-NAME-REPORT

```

IF FIRST-PAGE OR FULL-PAGE
  PERFORM 870-PRINT-REPORT-HEADING
MOVE PD-DEPT-NBR      TO OL-DEPT-NBR
MOVE PD-PERSON-NAME   TO OL-PERSON-NAME
PERFORM 210-LOOKUP-DEPT-TABLE.
IF TABLE-ENTRY-FOUND
  MOVE PD-DEPT-NAME(ST-INDEX) TO OL-DEPT-NAME
ELSE
  MOVE "DEPT NOT IN TABLE"    TO OL-DEPT-NAME.
MOVE OL-DEPT-NAME             TO DEPT-REPORT-LINE
MOVE 1                         TO WS-LINE-SPACING
PERFORM 890-PRINT-REPORT-LINE
PERFORM 800-READ-PERSONNEL-RECORD

```

210-LOOKUP-DEPT-TABLE

```

SET ST-INDEX          TO 1
SEARCH DEPT-TABLE-ENTRY
  AT END
    MOVE "NO "        TO TABLE-ENTRY-FOUND-SW
  WHEN
    TD-DEPT-NBR(ST-INDEX) IS GREATER THAN PD-DEPT-NBR
    MOVE "NO "        TO TABLE-ENTRY-FOUND-SW
  WHEN
    TD-DEPT-NBR(ST-INDEX) IS EQUAL TO PD-DEPT-NBR
    MOVE "YES"        TO TABLE-ENTRY-FOUND-SW

```

Table Lookups With INDEXES And SEARCH Statements

From the Coding that I have provided above, how can you tell if this is a Serial Search? How many Users know? A Serial Search is one where the table is stepped through, entry-by-entry, until a Match is found or End of File is reached. Also, the ST-INDEX must be SET to 1 so the serial search is started at the top of the table! With this HINT, can you find the key phrase(s) for a serial search? When the table has been established with an INDEXED BY clause and a SEARCH statement is used, index incrementation is handled automatically by the COBOL compiler; thus, a step-by-step serial search! (A Binary Search key phrase is SEARCH ALL!)

I used multiple WHEN phrases that cause the SEARCH to END whenever ANY WHEN condition is satisfied. This is similar to that of an OR operator. The multiple WHEN choice is used for what is usually called a Serial Search with an "early exit"! I am sure that the User can see that WHEN a condition is satisfied the SEARCH statement will be exited. Be sure you note the conditions used — "IS GREATER THAN" and "IS EQUAL TO"! Further, this example uses indexes rather than subscripts. We will be doing a table program using subscripts in a following article.

The INDEXED BY clause is usually written with a User-defined index-name; ST-INDEX in my example. The index-name is used like a variable subscript for identification of a specific occurrence. That sounds like a mouthful of words. If you have programmed with Arrays in BASIC, you will understand this better.

Let's get a little ahead of ourselves, as we need to know there are three important differences between subscripts and indexes:

1. When an index-name is specified in the INDEXED BY clause, the index is automatically provided by the COBOL compiler.

The programmer does NOT establish a data-name in the DATA DIVISION as would be required for a variable subscript.

2. Subscripts are usually used in storage of occurrence values; whereas, indexes contain "displacement" values. Displacement refers to the number of positions from the starting position of the table.
3. Index values are usually different than normal data values; therefore, initialization of the index cannot be done with a MOVE statement as can initialization of a subscript. Also, arithmetic statements — Add or Subtract — cannot be used with incrementation or decrementation of an index. Instead, the SET statement MUST be used for an index initialization, incrementation, decrementation, or other modifications. Do not let this last sentence scare you. You will not see any usage except the SET statement as a rule!

I do not expect you to understand these three differences at this time. We will do projects that will demonstrate these differences in following articles. Tables are so important that we will be doing many table programs in many of the following articles.

Another point needs clarification. Index data items may be specified in the DATA DIVISION to allow for storage of index displacement values for later use without their conversion to occurrence numbers. To specify an index data item, the USAGE IS INDEX clause is used. Notice that PICTURE and VALUE clauses are not allowed with INDEX usage. Index lengths are always uniform and assigned by the COBOL compiler. Recognize the difference between indexes and index data items. indexes are specified with the INDEXED BY phrase and are used, like a subscript, to refer to a certain occurrence of a data item containing an OCCURS clause. The only function of an index data item is to store, without conversion, the contents of an index. Therefore, index data items are only required occasionally and they are not in common use.

In the PROCEDURE DIVISION, index data items can only be used in a SET statement or in a relation condition with an index or another index data item. I will try to clarify all of this for the Users with projects illustrating these terms in following articles. We will be working with tables and their terms the rest of the way with "COBOL Corner"!!

By now, I would expect most Users to see that tables are defined in the WORKING-STORAGE SECTION of the DATA DIVISION! To establish a "hard-coded" storage table, the programmer uses VALUE clauses of equal length, padded with spaces for the shorter items. This was shown in Article XVI. After the table data has been specified with VALUE clauses, the table MUST be redefined with the OCCURS clause. The OCCURS clause tells the compiler how many times a field(s) is reported. It always requires an integer. The OCCURS clause cannot be used with an 01-level or 77-level and it cannot contain a VALUE clause. Whenever an OCCURS clause is used with a data item, either a subscript or an index MUST be used when referring to that data item in the PROCEDURE DIVISION. We are using indexes with this program and we will use subscripts in a future program.

Note! Each of the data items in the hard-coded tables were NOT given a User defined name, but they were named FILLER. This is usually the best way because the data is normally referred to only by the redefined data item within the OCCURS clause. User defined names can be used if desired, but they are seldom used.

Besides the two hard-coded tables, this program uses an Input-Loaded Table from records in a disk file. Each record contains the

data for one table entry. The program contains the program logic to read the table records and move the input table data to the table area of the WORKING-STORAGE SECTION. Can you see where this is done? Review the previous article and this article if you need help. The program logic MUST check the input records so that it can determine:

1. If table data is available.
2. When table data has come to an end and the processing of regular data is to begin.

When this table data is loaded from the disk file, ALL the additional definitions and processing statements for that input file must be provided. This includes the following:

1. SELECT statement.
2. FD statement.
3. OPEN statement.
4. READ statement.
5. CLOSE statement.

Be sure to check your coding for these! ONE WARNING for the Input-Loaded Table Data — There is a chance that through additions or deletions the table data will exceed the limits of the defined table. If changes to the data are expected, the number of occurrences specified in the OCCURS clause should be a value large enough to cover a reasonable number of additions.

Note: This program does NOT get into Input Data ERROR checking. Data validation and ERROR traps are very important to programmers. We will study and add these checks into future programs. Also, the processing is usually more efficient by using indexes versus subscripts; thus, the programmer should determine whether the INDEXED BY clause can or should be used. Additional articles with more programs using indexing and subscripts will enable the User to decide which to use. The serial search is best when the table data is limited to 25 or 30 records. If this is exceeded, good programmers will consider using a Binary Search. Again, future programs will clarify how this decision is best made.

Data Handling

I would expect that many Users will have internal clocks installed into their computer. We have been writing our code up to now assuming that there was no clock. If this is still true, use the same coding that we have used in prior programs for the Date coding. But, if you want to make use of your internal clock, the ACCEPT (Format 1) statement will be required. This affects your WORKING-STORAGE SECTION and the PROCEDURE DIVISION coding. Refer to your COBOL Manual about this ACCEPT statement. I have listed the main part of the Date coding below:

WORKING-STORAGE SECTION

```
01 WS-DATE-AREA.
   05 WS-DATE-WORK                PIC 9(06)
   05 WS-DATE-CONV REDEFINES WS-DATE-WORK
      10 WS-YEAR                  PIC 9(02)
      10 WS-MONTH                 PIC 9(02)
      10 WS-DAY                   PIC 9(02)
```

PROCEDURE DIVISION

```
100-INITIALIZE-VARIABLE-FIELDS
   ACCEPT WS-DATE-WORK           FROM DATE
   MOVE WS-MONTH                 TO H-MONTH.
   MOVE WS-DAY                   TO H-DAY
   MOVE WS-YEAR                  TO H-YEAR
```

Note: This Date handling could vary with different computers and clock set-ups. These Hints should get the User started.

COBOL Update

Back about March 1985, Heath made an offering to the owners of COBOL-86 (MS-463-3) that was used with the H/Z-100 computers, but NOT the H/Z-150 PC computers, a discount of \$300 towards the NEW Version of COBOL Compiler (MS-5063-3) that is priced at \$395. I hope that the Users that had the earlier version took advantage of this offer. This new version of COBOL provided the following changes:

1. Software works with both the H/Z-100 and the H/Z-150 PC type computers. This means that the owners of the H/Z-150 PC type of computers can NOW have a COBOL compiler.
2. A major enhancement is the INSTALL program. The function keys and the screen parameters are setup by this INSTALL program for either of the computer types. The foreground/background attributes and the function keys will be available for both types of computers. SO, let's get the owners of the PC type computer going with "COBOL Corner"!
3. MS-DOS Version 2.xx pathnames can now be used within COBOL programs and are allowed by the REBUILD utility.
4. The Compiler switches, /T and /C, now allow for drives A through Z to be specified, rather than the previous limit of A through D. This is important if the User is using a hard-disk drive and RAMdisks.
5. The limitation which existed on the size of an item subordinate to an item containing an OCCURS clause (2048 bytes maximum) has been removed.
6. Any COBOL programs that were used with versions prior to 1.10 MUST be run through the REBUILD utility and recompiled.
7. This new COBOL software will work with the MS-SORT (MS-5063-12) software. We have not reached the point where SORT will be required with our projects, but we are getting close. There are two schools of programmers. One group believes that the SORT should be done externally to the COBOL program and the other group believes that the SORT should be done internally within the COBOL program. Most COBOL programmers that use Mainframe computers use internal SORT. If you are learning COBOL for use in the Real World, you should know both ways. I will be teaching both ways in future programs and I would advise the readers to obtain this new version of COBOL and the MS-SORT software. There is an MS-SORT software package for use with COBOL-80, as well.

Conclusion

We have now reached the point where we will be building a stack of knowledge with one program as the foundation for the next program. I hope that the User will learn each program or you could get lost with the coming programs. I will hold your hands with a lot of help for the new material in each program, but I will not be repeating the coding from prior programs. Of course, you MUST always do the planning phases.

If you need help, write to me with a printout of your problem with an explanation of the problem. I will answer if you furnish a SASE (business size), otherwise, you will NOT receive an answer. I enjoy working with the readers, but I get too many letters to work without this!



HUG Members only!



HUGE HUGGER DISCOUNTS now available on these selected products...exclusively for members only!

That's right. We're offering HUG members these close-out, last call and limited quantity specials from Heath Company.

The items listed are not necessarily representative of complete systems. Please call your nearest Heath/Zenith Computers & Electronics Center or Heath Company toll-free at 1-800-253-7057 for more information before ordering.

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

REMark Discount List

Item	Last Adv'd Price	March Issue Price	HUG Member Price	Qty Left	Item	Last Adv'd Price	March Issue Price	HUG Member Price	Qty Left
H-19-3					PPA-210-2				
H-88-1					PPA-210-3				
H-88-5					PPA-215-1				
H-88-9					PPA-260-2				
H-89A					PPA-260-3				
HCA-9					PPA-260-4				
PA-100-1					PPA-260-5				
PC-121-50					PPA-260-6				
PC-131					PPA-260-8				
PC-140					PPA-260-9				
PC-200					PPA-260-20				
PC-310					PPA-270-1				
PCA-120-1					PPA-270-2				
PP-110					PPA-270-4				
PP-210					PPA-270-5				
PP-215					PPA-400-1				
PP-260					ZA-100-4				
PP-400					ZA-219				
PP-401					ZG-219				
PPA-210-1					ZW-219				

Heathkit®
Heath
Company

First Annual
MIDWEST HUGCON
Sponsored by the St. Louis Heath Users' Group
Official Conference Registration Form

HENRY VIII INN AND LODGE
Saint Louis, Missouri
May 2, 3, 4, 1986

Name: _____

Company: _____

Address: _____

City: _____ State: _____ Zip: _____

The registration fee is \$20.00 per registration and entitles you to all functions of the conference. A Visitor registration of \$5.00 allows you simply to attend the seminars and exhibits. These tickets do not include eligibility for the Grand Prize or banquet while attending the conference.

Amount Enclosed: \$ _____ Number Attending: _____

What brand and model of computer(s) do you operate? _____

Are you a Non-User Attendee? Yes _____ No _____

Are you a computer related manufacturer? Yes _____ No _____

If yes, would you like exhibitor information? Yes _____ No _____

Is anyone in your party interested in non-conference activities in the St. Louis area? Yes _____ No _____

Special Notice To Exhibitors

Exhibitor Information Packets are available on request from the St. Louis HUG. Those interested in exhibiting should contact us at the address below.

MAKE CHECKS PAYABLE TO: St. Louis HUG, Inc.
Mail To: MIDWEST HUGCON
Attn: Conference Registration
P.O. Box 482
Collinsville, IL 62234

Registration forms must be postmarked no later than April 12, 1986.
Cancellations will not be accepted after this date.

Sponsored by the St. Louis Heath Users' Group will be held May 2nd, 3rd, and 4th at HENRY VIII Inn Lodge in St. Louis County, Missouri.

HENRY VIII is an excellent hotel with 400 luxurious rooms and suites with rustic decor, reasonable rates, first-rate food, plenty of parking, and conveniently located only 2 miles from Lambert International Airport with 24-hour courtesy car services.

HENRY VIII conference rates are \$55.00 per room, single or double occupancy. A hotel reservation request will be returned to you upon receipt of MIDWEST HUGCON registration.

The three day conference is for Heath/Zenith and compatible people, including IBM Users!!! The conference will cost \$5.00 to get into the Conference Meetings, Vendor Area, and give you a chance for a Door prize.

Advanced Registration is required to be eligible for the Grand Prize drawing at the Saturday Night Banquet. The banquet tickets will cost \$20.00 for each individual attending. Included in this price is an outstanding meal and access to the Conference Meetings and Vendor Areas during the three day conference.

We plan to have a number of speakers for a wide variety of topics during the conference. I guarantee there will be something for everyone from novice to expert.

Hope to meet all of you at the First Annual MIDWEST HUGCON!!! Cheers . . . Rich, SLHUG SYSOP

**INTERNATIONAL
 HEATH/ZENITH USERS' GROUP
 CONFERENCE**

Official Conference Registration Form

**O'Hare Hyatt Regency
 Rosemont, Illinois
 August 15, 16, 17**

Name(s): _____

 Company: _____
 Address: _____
 City: _____ State: _____ Zip: _____

Enclosed is \$25.00 for each of the individuals listed above to attend the International HUG Conference being held the weekend of August 15, 16, and 17, 1986. Please send tickets along with information regarding hotel reservations and transportation.

Amt. Enclosed: _____ No. Attending: _____

For Our Information:

Which Heath/Zenith computer do you now operate? _____

- | | | |
|---|-----|----|
| Are you a Non-User-Attendee? | Yes | No |
| Are you a computer related manufacturer? | Yes | No |
| If yes, would you like exhibit information? | Yes | No |
| Are you, or anyone in your party, interested in activities in or around the Chicago area other than the Conference? | Yes | No |
| If yes, please indicate any suggestions you may have:
_____ | | |

Special Notice To Exhibitors:

Exhibitor Information Packages are available on request from the Heath/Zenith Users' Group. Those of you interested in exhibiting your products should contact us as early as possible to ensure a position at this year's event.

For Your Information:

The \$25.00 you are paying for your reservation to the International HUG Conference entitles you to all functions of the Conference. Visitor tickets, for those of you simply attending the seminars and exhibits, are available for \$10.00. Visitor tickets do not include eligibility for prizes or food while attending the Conference.

Please send your completed registration form or suitable copy to:

Heath/Zenith Users' Group
 Attention: International HUG Conference Registration
 Hilltop Road
 St. Joseph, Michigan 49085

The Plight Of The Cockroach

Jedaliah (Jed) Borlowinski

-by-

leaving it subject to attack by the standard warble. This warble squaring effect is called: 'Variations Effecting Cockroaches'. Now, for the first time, you can create all the necessary sounds with your IBM PC, or PC compatible computer, like the H/Z-150/160/200 which is manufactured by Heath/Zenith. This program was written in the highly efficient programming language, GW-BASIC. Its effective range is 700 feet, and should be quite sufficient for most households. To be most effective, the program should be allowed to run for 72 hours or more. This amount of time will assure effective warble penetration. The program itself is quite complex. The timing loops and subroutine locations are very critical, and must not be changed or modified in any way. Even though a part of the program may look ineffective, be assured that these parts are there to generate the inaudible parts of the warbling effect. Compiling this program will also destroy the proper high pitch frequency generation. It must be run by the GW-BASIC interpreter.

Next month, watch for my program called: "Speaker Saver". This program is a Terminate-and-Stay-Resident routine which prevents high frequency spikes from destroying the internal speaker in your computer. Due to the subliminal effects needed to be generated by this program, a Fortn Compiler will be required. On a final note, I would like to mention that this article should only be adhered to during the first day of the fourth month of any year.

Scientists, over the years, have studied various methods of ridding the average homeowner of pests of every sort: mice, rats, cockroaches, fleas, and termites just to name a few. New poisons are constantly being developed. Some of these include powders that you 'sprinkle' behind your appliances, sprays that seek and destroy in the smallest of crevices, and entrees that appear, taste, and smell like bait to a rodent, but are laced with the deadliest of poisons. Now only recently have scientists begun to explore the possibility of high tech electronic devices, specifically sound generation, to repulse those unwanted intruders.

In recent months, several companies have offered the public 'high tech' electronic audio type device which they claim will repel all types of household pests, including insects of all types and rodents. Some of these companies include: Poopel, Rumco, and Keetell. Unfortunately, these companies have not gone far enough with these studies, placing on the open market a product which, in simple language, does not work!

Now, due to most recent studies by the Institute for the Better Management of Pest Control (IBMPC, for short), scientists have uncovered startling new facts regarding the effects of sound generation on household pests. Because of heavy pressures applied by the poison industry, this information would still have been kept secret if not for the undercover efforts of my late brother, Fred Borlowinski! It is in Fred's memory that I release these startling facts.

It is indeed true that most household pests will respond in some way to the generation of constant audio frequencies, however, not in the way we'd like them to. As an example, most rodents get frisky at 16,479 Hz, cockroaches tend to multiply at an abnormal rate at 17,231 Hz, and fleas will bite more than normal at 18,712 Hz. As one can see, constant frequencies produced by existing electronic devices do not produce the desired effect. It has been shown that sound, 'warbled' between several different frequencies will have detrimental effects on all household pests, including rodents! Because of this new technology, this warbling effect is measured in new units called: 'Hertz Effectiveness Against The Household pests'. Applying five of these units to an infested area will not only drive away the pests, but will cause a fatal nervous disorder in those pests called: 'Zombitis Delayicus Syndrome'. Scientists have not yet discovered why this occurs, only that it does.

Two exceptions to the general pest world exist. First, are the flying insects. Because of the air disturbance, and subaudible frequencies generated by the flapping of their wings, special cancellation warbles must be generated to negate these effects. These negative effects are called: 'Beats Against Rapid Flyers'. The second exception is the prehistoric cockroach. These little devils seem to be impervious to everything. Scientists at IBMPC, however, have found their weakness. By generating the standard amount of warbles at the exact rate of a warble (warble squared), the defense mechanism of the common cockroach is destroyed

```

10 REM MULTI-SONIC INSECT REDUCTION PROGRAM
20 REM FOR PC-TYPE COMPUTER
30 REM BY FRED BORLOWINSKI
40 REM
50 REM
60 REM FOR BEST RESULTS, RUN ON REAL IBMPC
70 DEFINIT A-Z,CLS,KEY OFF
80 FOR I=1 TO 200:READ X:X=X+32-RND(2):PRINT CHR$(X):
90 NEXT I:PRINT
100 INPUT "ENTER ESTIMATED NO. OF COCKROACHES TO DESTROY":BUGS
110 RANDOMIZE BUGS:REM AREN'T BUGS ALWAYS RANDOM?
120 FOR I=1 TO 5:READ SN(I):NEXT I
130 WHILE BUGS > 0
140 FOR I=1 TO 5
150 SOUND SN(I) < 1 THEN PRINT "ERROR -- NO COCKROACHES FOUND!":STOP
160 SOUND SN(I)+INT(RND*500):DU(I)+INT(RND*3)
170 NEXT I
180 WEND
190 DATA 67,97,117,116,105,111,110,32,45,45,32,67
200 DATA 111,99,107,114,111,97,99,104,101,110,32,119
210 DATA 105,108,108,32,98,101,32,110,110,114,114,101
220 DATA 109,101,108,121,32,97,110,103,114,121,13,10
230 DATA 119,104,101,110,32,121,117,32,113,117,105
240 DATA 32,116,104,105,115,32,112,114,111,103,114,97
250 DATA 109,33,33,32,32,84,111,32,112,114,111,116,101
260 DATA 99,116,32,121,111,117,114,32,101,113,117,105
270 DATA 112,109,101,110,116,113,102,114,111,109,32
280 DATA 115,97,98,111,116,97,103,101,44,32,115,116
290 DATA 111,114,101,32,116,104,105,115,32,112,114,111
300 DATA 103,114,97,109,32,111,110,101,111,32,97,111,110
310 DATA 32,114,101,109,111,118,97,98,108,101,13,10
320 DATA 109,101,100,105,97,44,32,97,110,100,32,114
330 DATA 101,109,111,118,101,32,112,110,97,97,32,119,115
340 DATA 97,102,101,32,112,108,97,99,101,32,119,104
350 DATA 101,110,32,110,111,116,32,105,110,32,117,117,115
360 DATA 101,46,13,10
370 DATA 1000,1,1500,2,1000,2,1500,1,1000,1

```

*



NOVELL COMPATIBLE

ONE-STOP ARCnet™ LAN SOLUTIONS

Now you can get all the tools you need to build industry standard **TOKEN-PASSING** networks from one source.

InterContinental Micro.

Our networking packages let you configure any combination of nodes (PC s and compatibles, ATs, XTs, Jrs., Z-100 s) to a common Fileserver in just a few minutes with our menu-driven installation program.

Active and passive hubs allow you to integrate up to 255 users on a single network with *increased* efficiency as the network grows.

In addition, our stand-alone ARCnet™ Workstation/Fileserver and complete line of S-100 Bus products (single board computers, slaves and controllers) give you and your customers even more options.

ARCnet is a trademark of Datapoint Corporation.

We're also compatible with other industry-leading ARCnet PC™ and S-100 products, and our PC networking boards include an optional 256K of system RAM.

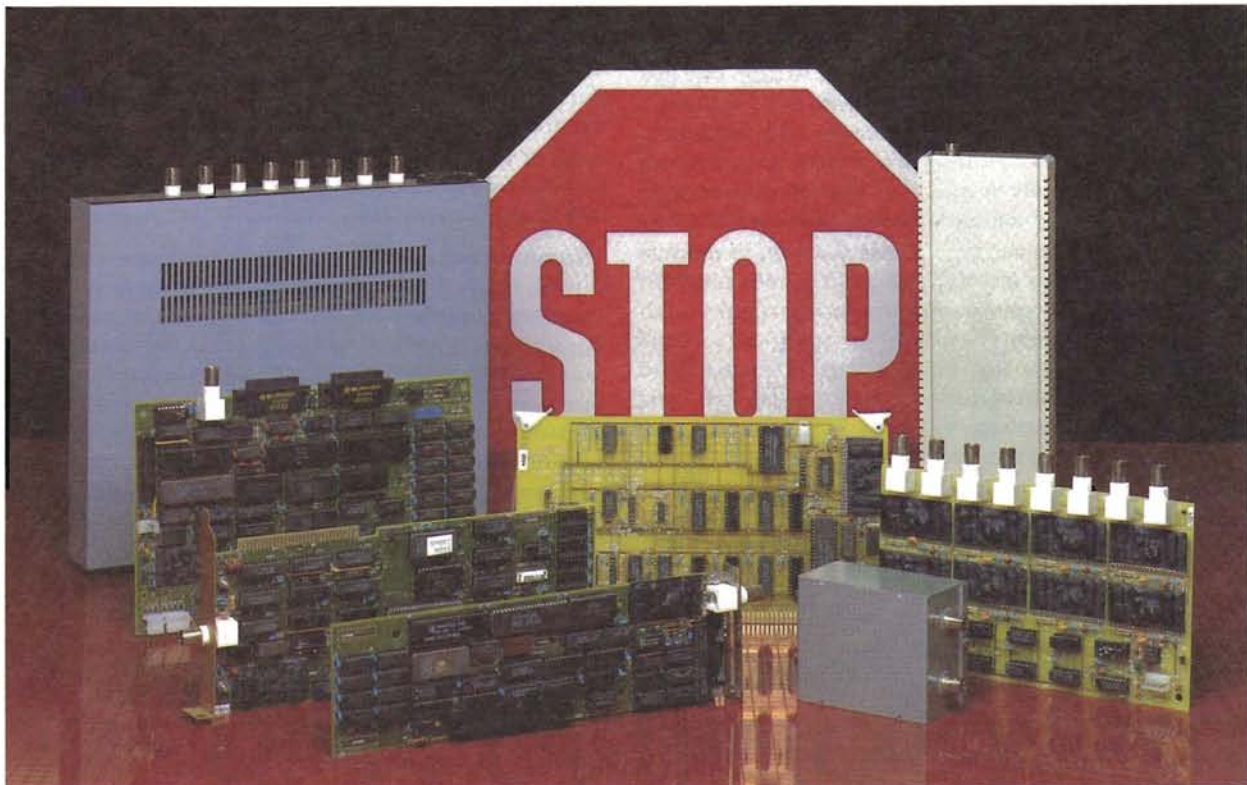
All with InterContinental Micro's reputation for outstanding reliability, exceptional service, and highly competitive pricing.

Call InterContinental Micro today, or circle the bingo number below for ARCnet™ LAN solutions.



4015 Leaverton Court, Anaheim, CA
92807; Phone: (714) 630-0964
Telex: 821375 SUPPORT UD; Easylink: 62562040.

Call for GSA pricing.



ZPC Hardware Patch



Jumpin' Jiminy!

Build this simple circuit and run many PC programs under ZPC without patching them!

Pat Swayne
HUG Software Engineer

CAUTION: In addition to building a circuit from "scratch", you must make a minor modification to the video board in your computer in order for the circuit to work. Although the modification is quite simple, it involves soldering to a trace on the board. Care should be taken when making the modification, and you should only attempt it if you feel comfortable doing such work.

ZPC is a HUG program (885-3030-37) that lets you run many IBM PC programs on your H/Z-100 computer. Unfortunately, many of the programs must be patched before they will run under ZPC. Although I have tried to provide patches for some of the more popular PC programs in REMark and on the ZPC Support Disk (885-3034-37), it has turned out to be a real hassle. The patches for each program are usually only good for a specific release of the program, and if there have been several updates, however minor, new patches may be necessary for each new update. For example, I have already released three different patches for SuperCalc3, and I am about to release two more, later in this article. In this article, I will describe a simple circuit that you can build and plug into your computer that will eliminate the need to patch SuperCalc3 and many other PC programs. The circuit can be built on an S-100 wire-wrap board for the H/Z-100, and a version of it can also be built on the bread board of an ET-100 computer. I will present schematics of both versions in this article.

Note: The circuits described in this article are not "kits". They are designed to be built up from scratch in "home brew" fashion. For this reason, only the schematics and theory of operation are

presented, but not actual construction details. It is highly possible that my method of construction is not the best, anyway.

Your computer must have 768k of memory in order to take advantage of ZPC and these circuits. See the July 1985 REMark for information on putting 768k in an H/Z-100, or the February 1986 REMark to see how to put 768k in an ET-100.

A Little Technical Background

The reason why a program must be patched is usually because it displays characters on the screen by writing directly to video memory. Before it writes to video memory, it tests a port to detect the proper time to perform the write operation. On an H/Z-100, the port the program accesses does not return the value the program is looking for, so it "hangs up". Sometimes, the offending program will also write to the PC video control ports, which conflict with the H/Z-100 video ports. This can cause the display to flash, change colors, or "go crazy".

A few PC programs intercept the keyboard interrupt, and during interrupt time they get the code for the key just pressed by inputting from a port, and/or clear the keyboard of the last code via another port. ZPC emulates the PC keyboard interrupt, but the ports for reading or clearing the keyboard do not exist on an H/Z-100.

The circuitry described here, which I call the ZPC Hardware Support circuitry (ZHS), provides a port that a PC program can test and get the value it is looking for when it attempts to write to video memory. It also provides a decoder for the keyboard ports of a PC that generate an interrupt whenever the ports are accessed. A software routine, supplied later in this article, uses that

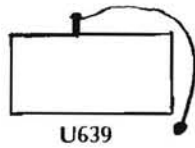
interrupt to provide the key code for the requesting program or clear the keyboard.

Video Board Modification

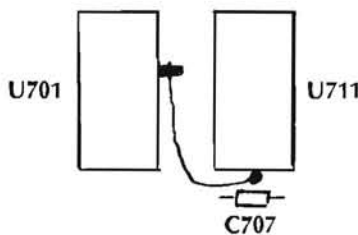
One thing the ZHS cannot do is prevent a write to a PC video control port from conflicting with an H/Z-100 video port. A simple modification to your video board is required to correct that problem. The modification is harmless to normal H/Z-100 operation, and can be performed even if you decide not to build the ZHS.

To perform the modification, first unplug your computer and remove all disk drives, printer, and any other cables attached to it. On an H/Z-100, remove the cover, the video/drive assembly (all-in-one) or the drive assembly (low profile), the cabinet base, the keyboard, and the video board. On an ET-100, you only need to open the computer up to expose the video board, but you do not need to remove any components.

Prepare a 16-pin IC socket (Heath part no. 434-299) by bending pin 13 outward 90 degrees and attaching a 2" (5 cm) length of 28 or 30 AWG wire-wrap wire to the bent-out pin. Remove the IC at U369 (H/Z-100) or U701 (ET-100) and place it in conductive foam. On a Z-100, install the prepared socket at U369 (you may have to bend the bent pin more than 90 degrees to clear RP303), and connect the wire from pin 13 to the feed-through hole by pin 8 of U369, as indicated in the drawing below.



You may have to scrape some of the solder resist (green paint) from the feed-through hole before you can solder the wire to it. On an ET-100, install the prepared socket at U701 (you may have to bend the bent pin more than 90 degrees to clear RP701), and connect the wire from pin 13 to the feed-through hole between U711 and C707, as shown below.



Replace the IC that you removed from U369 or U701, installing it into the modified socket.

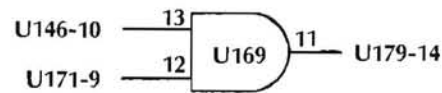
Optional Serial Port Modification

While you have your computer opened, there is an additional modification that you may want to perform that will increase the number of programs that you can run without patching them first. A program that writes to one of the serial ports on a PC-type computer may write to the memory control latch on an H/Z-100 or ET-100, causing the machine to lock up. You can perform a modification that will prevent that from happening. On an ET-100, the modification is very simple, but on an H/Z-100, it requires an additional IC gate. I will present the ET-100 modification in detail, since I did it to my ET-100, but I have not done the H/Z-100 modification, so I will just present a schematic of it.

These modifications are done to the main board of each computer, not to the video boards.

To perform the modification on an ET-100, you will need two 16-pin IC sockets and a 2" (5 cm) length of wire-wrap wire. Connect one end of the wire to pin 10 of one 16-pin socket. Do not bend the pin out, and connect the wire as close to the socket base as possible so that the socket can be plugged into another socket. On the other 16-pin socket, bend out pin 14 and connect the end of the wire attached to pin 10 of the first socket to the bent pin. Remove the ICs at U255 and U242. Plug the socket with the wire attached to pin 10 into U255, and the other socket into U242. Replace the two ICs removed, installing them into the modified sockets.

On an H/Z-100, an additional gate is required to prevent the PC serial port conflict, because of the way the 8085 processor accesses ports. Below is a diagram of a possible implementation of the modification that uses an unused gate in U169.



Pin 14 of U179 is normally connected to ground, and you would have to use a socket with a pin bent out to make the connection as was done with the previous modifications. The other connections can be made to the foils on the back of the board.

After you make the board modifications to your H/Z-100 or ET-100, reassemble it and test it before proceeding with construction of the ZHS circuitry.

The ZHS Circuits

Figure 1 is a schematic of the H/Z-100 version of the ZHS circuit. Photo 1 shows how I built the H/Z-100 version on an S-100 wire-wrap board. Although I used a wire-wrap board, I used ordinary IC sockets (not wire-wrap sockets), and soldered all connections so that the board would be "thin" enough to be placed in any S-100 slot of an H/Z-100. The numbering of the ICs on the H/Z-100 schematic refers to the order from left to right that I placed the ICs on the S-100 board (except for the voltage regulator, U7). If you have never built a circuit on an S-100 board before, you should know that edge connectors are numbered 1 to 50 from left to right on the component side of the board, with pins 51 thru 100 on the back side arranged so that pin 51 is at the same end as pin 1.

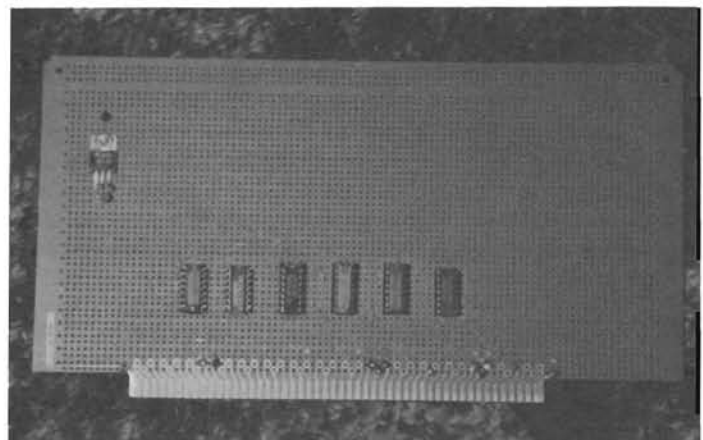


Photo 1
The H/Z-100 version of the ZPC hardware support circuit.

Photo 2 shows the ET-100 version of the ZHS circuit built on the ET-100's breadboard. Figure 2 is the schematic of the circuit. All connections were made on the breadboard, using the solderless connectors.

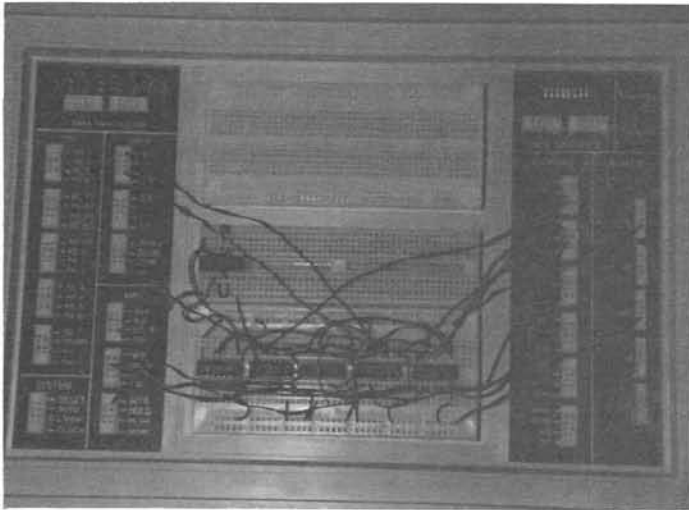


Photo 2

The ET-100 version of the ZPC hardware support circuit.

Theory Of Operation

On an H/Z-100 or ET-100, all of the ports used occupy address below 256 (100 hex), so only 8-bit circuits are used to decode the ports. This means that if you should run a "foreign" program that tries to access a port with an address equal to or greater than 256, the decoder will simply disregard the upper 8 bits of the address. If a program attempts to write to port 3DA (hex), which is one of the PC video ports, the H/Z-100 will see it as an attempt to access port DA. The jumper installed on the video board feeds address line A8 to a chip enable on the video port decoder, so that if the port address is 3DA instead of DA, the decoder will be disabled. The main board modification works similarly, except that address line A9 is used. The extra gate on the H/Z-100 main board modification is required because both the 8085 and the 8088 use the ports we are trying to protect, and the 8085 places a copy of the port number on the upper 8 address lines when it accesses an 8-bit port. Therefore, the address line going to the chip enable must be active only when the 8088 processor is active.

When a PC program is going to write to video memory, it reads port 3DA (hex) and waits until bit 0 goes low. Then it waits for bit 0 to go high again. Sometimes bit 3 is tested instead of bit 0. In the ZHS circuit, an 8-input NAND gate and some inverters are used to decode a read from port 3DA. Each access to the port flips a binary counter, whose output goes through buffers to data lines

0 and 3. Therefore, accesses to port 3DA will alternately see bits 0 and 3 go high and low. This action satisfies the tests made by the PC program.

When a program attempts to read ports 60 or 61 (hex), a second 8-input NAND gate decodes the attempt, and an interrupt is generated. The data sheet for the interrupt controller used in H/Z-100 and ET-100 computers specifies that the interrupt request line should be held active until the interrupt is acknowledged. On the H/Z-100 version of the ZHS circuit, I used an R-S flip-flop made from NAND gates to hold the interrupt line active until the interrupt acknowledge line on the S-100 bus goes active. I had to use a fast chip (74AL10) for the R-S flip-flop, because if I used a normal LS chip (with an 8-MHz Z-100), the interrupt was not serviced in time. For example, if the program contained:

```
IN      060
NOP
NOP
```

the interrupt would not be serviced until the 2nd NOP was reached.

On the ET-100, there is no interrupt acknowledge line on the breadboard, so I had to use a one-shot to hold the interrupt line active a sufficient amount of time. On the H/Z-100, I used vectored interrupt line 7 from the S-100 bus, which is set up

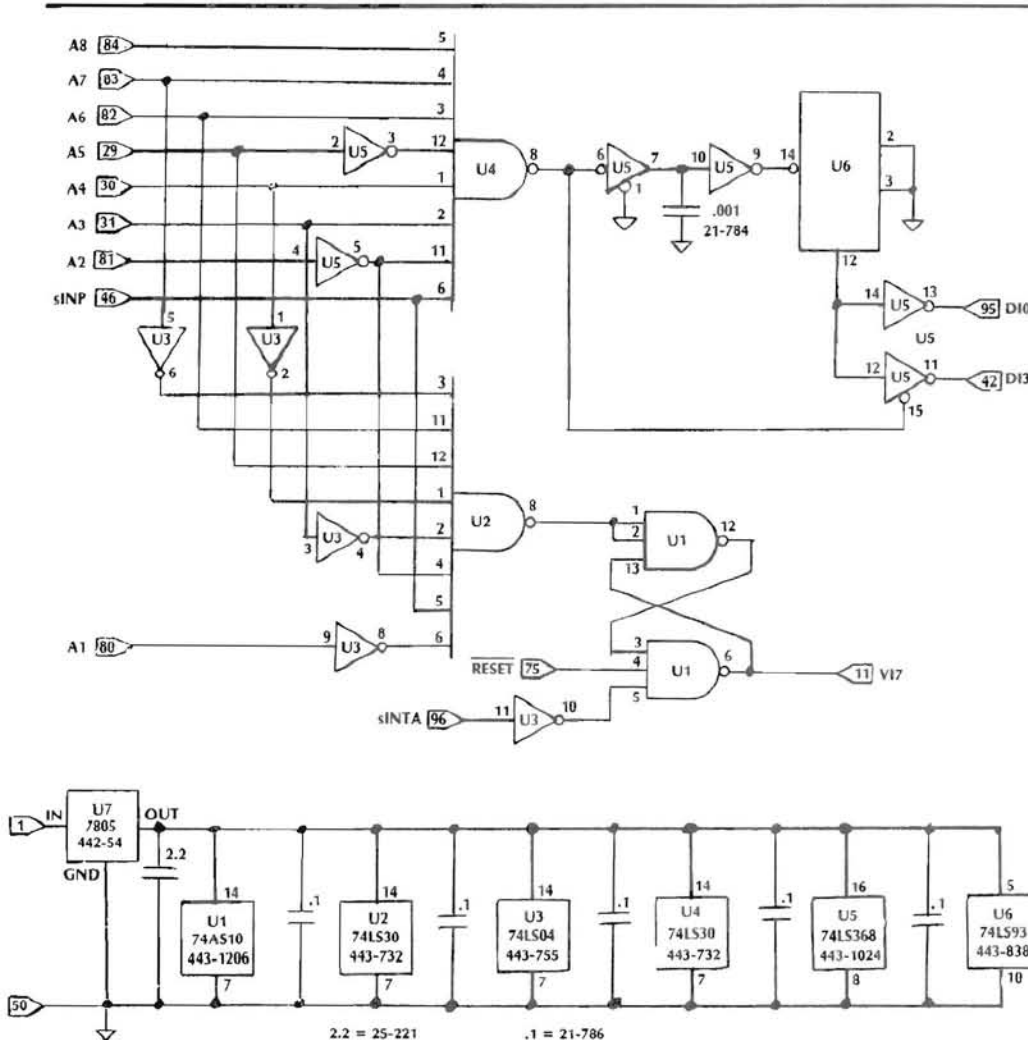


Figure 1

Schematic of the H/Z-100 version of the ZPC hardware support circuit.

to produce a type 4F (hex) interrupt when it is activated. On the ET-100, I used the user interrupt line provided on the breadboard, which produces a type 41 (hex) interrupt.

Those of you who are reading this who are planning to use one or the vectored interrupt lines on the S-100 bus for your own purposes should note that neither of the methods I used to hold the interrupt line active is probably the best method. Probably the best method would be to hold the line active until the software servicing the interrupt could deactivate it by sending a command to a port. Of course, on my circuits, that would have meant constructing another port, and I wanted to keep them as simple as possible. It is also possible to reprogram the 8359 (the chip that controls interrupts) so that it is activated by the edge of a pulse rather than a maintained level, and then you would not have to worry about keeping the line active until the interrupt is acknowledged. But the chip is programmed the way it is so that the vectored interrupt lines meet S-100 standards, and something you add to the computer later may require those standards.

When the interrupt is generated by the ZHS circuit, the software servicing it must find out which port was accessed. If the port was 60, the routine must return the value of the last key typed. If the port was 61, the routine must check to see if the keyboard has been cleared, and clear it if it has not.

Software Support

I have written a short program to provide the software support required by the keyboard section of the ZHS circuits. If you do not have an assembler, you can enter one of the two BASIC programs listed below and run it under ZBASIC or BASICA. The program will create a file on the currently logged disk called KEYINT

.COM. Use Program 1 if you have an H/Z-100, and Program 2 if you have an ET-100.

Program 1

```

10 REM THIS PROGRAM CREATES KEYINT.COM FOR AN H/Z-100
20 OPEN "0",1,"KEYINT.COM"
30 FOR I=1 TO 138
40 READ BYTE:PRINT #1,CHR$(BYTE);:NEXT I
50 CLOSE #1:SYSTEM
60 DATA &HEB,&H67,&H0,&H0,&H0,&H0,&H0,&H0
70 DATA &H0,&H0,&H2E,&HA3,&H2,&H1,&HB0,&H20
80 DATA &HE6,&HF2,&HE6,&HF0,&H2E,&H89,&H36,&H4
90 DATA &H1,&H2E,&H8C,&H1E,&H6,&H1,&H5E,&H1F
100 DATA &HFB,&H8A,&H44,&HFF,&H1E,&H56,&H2E,&H8E
110 DATA &H1E,&H6,&H1,&H2E,&H8B,&H36,&H4,&H1
120 DATA &H3C,&H60,&H74,&H27,&H3C,&H61,&H74,&HE
130 DATA &H3C,&HEC,&H75,&H1A,&H83,&HFA,&H60,&H74
140 DATA &H1A,&H83,&HFA,&H61,&H75,&H10,&H2E,&H80
150 DATA &H3E,&H9,&H1,&H1,&H75,&H8,&HCD,&H91
160 DATA &H2E,&HC6,&H6,&H9,&H1,&H0,&H2E,&HA1
170 DATA &H2,&H1,&HCF,&H2E,&HA1,&H2,&H1,&HCD
180 DATA &H90,&H2E,&HC6,&H6,&H9,&H1,&H1,&HCF
190 DATA &H0,&H33,&HC0,&H8E,&HD8,&HBE,&H3C,&H1
200 DATA &HC7,&H4,&HA,&H1,&H8C,&H4C,&H2,&HE
210 DATA &H1F,&HE4,&HF3,&H24,&HF7,&HE6,&HF3,&HE4
220 DATA &HF1,&H24,&H7F,&HE6,&HF1,&HBA,&H69,&H1
230 DATA &HCD,&H27

```

Program 2

```

10 REM THIS PROGRAM CREATES KEYINT.COM FOR AN ET-100
20 OPEN "0",1,"KEYINT.COM"
30 FOR I=1 TO 138
40 READ BYTE:PRINT #1,CHR$(BYTE);:NEXT I
50 CLOSE #1:SYSTEM
60 DATA &HEB,&H65,&H0,&H0,&H0,&H0,&H0,&H0
70 DATA &H0,&H0,&H2E,&HA3,&H2,&H1,&HB0,&H20
80 DATA &HE6,&HF2,&H2E,&H89,&H36,&H4,&H1,&H2E

```

```

90 DATA &H8C,&H1E,&H6,&H1,&H5E,&H1F,&HFB,&H8A
100 DATA &H44,&HFF,&H1E,&H56,&H2E,&H8E,&H1E,&H6
110 DATA &H1,&H2E,&H8B,&H36,&H4,&H1,&H3C,&H60
120 DATA &H74,&H27,&H3C,&H61,&H74,&HE,&H3C,&HEC
130 DATA &H75,&H1A,&H83,&HFA,&H60,&H74,&H1A,&H83
140 DATA &HFA,&H61,&H75,&H10,&H2E,&H80,&H3E,&H9
150 DATA &H1,&H1,&H75,&H8,&HCD,&H91,&H2E,&HC6
160 DATA &H6,&H9,&H1,&H0,&H2E,&HA1,&H2,&H1
170 DATA &HCF,&H2E,&HA1,&H2,&H1,&HCD,&H90,&H2E
180 DATA &HC6,&H6,&H9,&H1,&H1,&HCF,&H0,&H33
190 DATA &HC0,&H8E,&HD8,&HBE,&H4,&H1,&HC7,&H4
200 DATA &HA,&H1,&H8C,&H4C,&H2,&HE,&H1F,&HE4
210 DATA &HF3,&H24,&HFD,&HE6,&HF3,&HBA,&H67,&H1
220 DATA &HCD,&H27

```

If you run KEYINT after you run ZPC, it will load itself into memory, and process the interrupts generated by the ZHS circuitry. If you have an assembler, you can create KEYINT.COM by assembling the source code listed following this article. After you assemble and link the program, use EXE 2BIN to make it a .COM file.

ZPC Modification

To increase the number of programs that will run without patching using the ZHS

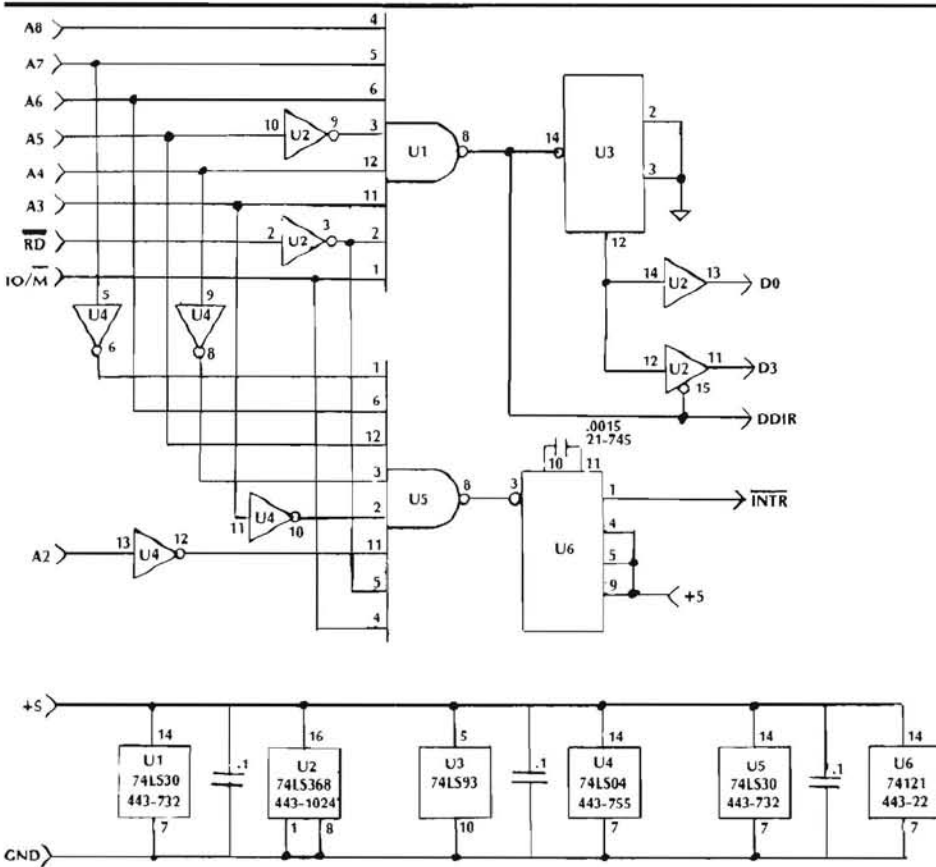


Figure 2
Schematic of the ET-100 version of the ZPC hardware support circuit.

circuitry, you need to patch ZPC so that it places the PC color/graphics card port at a particular memory location. The patch also changes two other values that are stored in memory. These are the address of the two standard COM ports on a PC, and I originally thought that they should be stored in memory at the same place that a PC stores them. As it turns out, it is better if non-conflicting port numbers are stored. To make the patch, create a file called MODZPC.DAT that contains these lines:

```
E16F
10 10
E1C9
D4 3 0 2B 0 CF
"
Q
```

Note that there are two spaces between the two 10s in the second line of the file. Copy MODZPC.DAT and DEBUG.COM to your ZPC system disk. It is assumed that the disk contains either ZPC3.COM or ZPC3A.COM, which has been renamed to ZPC.COM. Log on to the disk, and enter

```
DEBUG ZPC.COM <MODZPC.DAT
```

at the system prompt, and hit RETURN. ZPC.COM will be patched for you.

ZPC2 Is Coming!

I am working on a new version of ZPC that should be ready for release next month. If you have purchased the current version, you will be able to send in your disk and upgrade to the new version at a fraction of the list price. Watch the "New HUG Products" column of REMark for details. Here are some of the features of the new ZPC.

- Emulation of the PC BIOS disk interrupt, that will allow you to run some copy protected programs without altering them.
- Emulation of the PC monochrome video card in addition to the color/graphics card. In the monochrome mode, you will be able to run several programs without patching (and without hardware support) that require patches under the current ZPC.
- Built-in support for the circuitry described in this article. By using the new ZPC with the ZHS circuitry, you may be able to run up to 80 or 90 percent of all PC programs without patching them.
- Your choice of IBM or regular Z-100 fonts for alphanumeric characters while ZPC is in the PC mode.
- All fixes and improvements discovered since the original ZPC was first released will be incorporated in the new ZPC, along with several other improvements.

SuperCalc3 Patches

If you tried all of the patches I have published for SuperCalc3, release 2.0, and none of them worked, here is another one to try. Use the following lines as the text of your SC3PCH.DAT file, and follow the patch instructions given in previous ZPC Update articles.

```
F2FE3,2FED,90
F30BE,30C8,90
A30C5
AND BX,7FFF

FCC81,CC8B,90
FCCA1,CCAB,90
ACCA8
AND CL,7F
```

```
E3179
90
E318D
90
EC884
90
EC887
90
EC88A
90
EC88D
90
EC893
90
ED6AF
90
"
Q
```

If you have SuperCalc3, release 2.1, here is a patch that may make it run under ZPC. First, create a file called SC3PCH.DAT that contains these lines:

```
NSC3.COM
L
F381B,3825,90
FAE7B,AE85,90
AAE82
AND BX,7FFF

FDD7C,DD86,90
FDD9C,DDA6,90
ADDA3
AND CL,7F

E3957
90
E396B
90
ED9C1
90
ED9C4
90
ED9C7
90
ED9CA
90
ED9D0
90
EEDC1
90
"
NSCRIBI.DRV
L
E249C
90
E24A3
90
"
Q
```

Copy SC3PCH.DAT and DEBUG.COM to your SuperCalc3 Prod disk (you may have to delete SCRHERC.DRV first), log on to the disk, and enter

```
DEBUG <SC3PCH.DAT
```

at the system prompt, and hit RETURN. The patch will be applied automatically.

```
PAGE ,132
; KEYBOARD INTERRUPT PROCESSOR FOR ZPC
; THIS PROGRAM PROCESSES INTERRUPTS GENERATED BY THE
; ZHS BOARD IN THE EVENT OF INPUT FROM PORT 60 OR 61
;
; BY P. WAYNE, HUG SOFTWARE ENGINEER
;
; COPYRIGHT (C) 1986 BY HEATH/ZENITH USERS' GROUP
```

```

TRUE EQU 0FFFFH
FALSE EQU NOT TRUE
Z100 EQU TRUE ;ASSEMBLE FOR Z-100
ET100 EQU FALSE ;ASSEMBLE FOR ET-100
MASTER EQU 0F0H ;MASTER PIC BASE PORT
SLAVE EQU 0F0H ;SLAVE PIC BASE PORT
IF Z100
ZHSINT EQU 04FH ;Z-100 ZHS INTERRUPT
ENDIF
IF ET100
ZHSINT EQU 041H ;ET-100 ZHS INTERRUPT
ENDIF

JMPF MACRO
DB 0EAH ;DEFINE FAR JUMP
ENDM

DUMMY SEGMENT STACK
DUMMY ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE
ORC 100H

START JMP SHORT SETUP ;SET UP PROGRAM

AXSAV DW 0 ;AX SAVED HERE
SISAV DW 0 ;SI SAVED HERE
DSSAV DW 0 ;DS SAVED HERE
KEY DB 0 ;VALUE FROM KEY
INFLG DB 0 ;KEY INPUT FLAG

KEYINT: MOV CS:AXSAV,AX ;SAVE AX (FOR NOW)
MOV AL,20H
OUT MASTER,AL ;RESET MASTER PIC
IF Z100
OUT SLAVE,AL ;RESET SLAVE PIC
ENDIF
MOV CS:SISAV,SI ;SAVE SI
MOV CS:DSSAV,DS ;SAVE DS
POP SI ;GET USER'S IP
POP DS ;GET USER'S CS
STI ;ENABLE OTHER INTERRUPTS
MOV AL,-1|SI| ;GET PORT NUMBER
PUSH DS
PUSH SI
MOV DS,CS:DSSAV ;RESTORE DS
MOV SI,CS:SISAV ;RESTORE SI
CMP AL,60H ;READ KEY?
JZ KEYIN
CMP AL,61H ;RESET KEYBOARD?
JZ RESKEY
CMP AL,0ECH ;IN AL,DX?
JNZ NOKEY ;BAD INTERRUPT
CMP DX,60H ;READ KEY?
JZ KEYIN
CMP DX,61H ;RESET KEYBOARD?
JNZ NOKEY
RESKEY: CMP CS:INFLG,1 ;KEY TAKEN?
JNZ NOKEY ;NO
INT 91H ;RESET KEYBOARD
MOV CS:INFLG,0 ;RESET KEY FLAG
NOKEY: MOV AX,CS:AXSAV ;RESTORE AX
IRET ;RETURN FROM INTERRUPT
KEYIN: MOV AX,CS:AXSAV ;RESTORE AX
INT 90H ;GET KEY
MOV CS:INFLG,1 ;SET KEY FLAG
IRET

DB 0
ENDRES LABEL NEAR ;END OF RESIDENT CODE

SETUP XOR AX,AX
MOV DS,AX ;PUT DS AT 0
MOV SI,OFFSET ZHSINT*4 ;POINT TO ZHS INT VECTOR
MOV WORD PTR [SI],OFFSET KEYINT ;SET NEW VECTOR
MOV 2[SI],CS
PUSH CS
POP DS ;FIX DS

```

```

IN AL,MASTER+1 ;READ MASTER PIC MASK
IF Z100
AND AL,0FFH-8 ;UNMASK SLAVE PIC INPUT
ENDIF
IF ET100
AND AL,0FFH-2 ;UNMASK USER INTERRUPT
ENDIF
OUT MASTER+1,AL
IF Z100
IN AL,SLAVE+1 ;READ SLAVE PIC MASK
AND AL,7FH ;UNMASK VECTOR INTERRUPT 7
OUT SLAVE+1,AL
ENDIF
MOV DX,OFFSET ENDRES ;POINT TO END OF RES. CODE
INT 27H ;EXIT, PROGRAM RESIDENT

```

```

CODE ENDS
END START

```

H/Z-25 Super Chip Set

- 4 Chip Set
- 12 New Features
- 256 Characters
- Double Strike
- Super/Sub's
- Underline
- Italics
- more

\$84.95
+S&H

FINA SOFTWARE
16144 Sunset Bl. #3
Pacific Palisades
California 90272
213/454-6393

MS-DOS DISK MYSTERY REVEALED

\$24⁹⁵

*** DAE ***
Displays MS-DOS Disk Data in HEX and
ASCII and tells you what you are looking at!

YOU CAN SEE IT * * YOU CAN CHANGE IT easily

SEE DISK Format details SEE the BOOT code
SEE your FILES and DIRECTORIES SEE the FATs

Works on 5" disks, 8" disks, HARD disks, RAM disks

USE for Education USE for Changing File Names
USE for Changing Files USE for Changing Labels
USE for Patches USE for many other things
USE to change anything on the disk

AVAILABLE for (MS-DOS 2.0 or greater required)
Heath/Zenith 100 machines.....Order DAE-100
Heath/Zenith PC machines... ..Order DAE-151
IBM and Compatibles... ..Order DAE-IBM

COMPUTECH

55 Charles Parish Drive, Poquoson, VA 23662
(804) 868-8055 MC & VISA Welcome

Heath/Zenith Related Products

Jim Buszkiewicz
HUG Software Developer

KEA Systems Ltd. of Vancouver B.C., announces a new utility called DECKHAND. This utility brings more DEC-like power and syntax to the commonly used utilities: COPY, DELETE, DIR, RENAME, and TYPE. For example, some of the DIR features include sorting by name, type, size, and date. In addition, the output can be formatted and directed to a specific output device. Some of the features added to the COPY, DELETE, TYPE, and RENAME commands include confirmation, querying, and logging. DECKHAND sells for \$69.00 and will run on ANY MS-DOS computer.

KEA also has available a new PAL for the H/Z-158 computer system, allowing access to the full 768k of memory. This new PAL, ZPAL-158, also allows segmenting the memory so memory disks can be utilized. ZPAL-158 presently sells for \$36.00.

For more information regarding these two products, contact KEA Systems Ltd. 412-2150 West Broadway, Vancouver, B.C. Canada V6K 4L9. To place an order, call (800) 663-8702.

Newline Software of Tiverton, RI announces a new GW-BASIC compatible compiler, called BUSI__BASIC. Some of the features in this compiler include: up to 640k memory usage, 4 virtual screens plus normal console screen, indexed sequential access method files (ISAM), up to 64 data segments and 64 code segments, separately compiled modules, locally defined variables and modular programming, load and call assembly language programs, optional line numbers in source code, remarks and comments do not consume memory space, and the ability to create your source code with any text editor. BUSI__BASIC presently sells for \$159.00, but is being offered for the introductory price of \$99.00. For more information, write Newline Software, P.O. Box 289, Tiverton, RI 02878, or call (401) 624-3322.

C.D.R. Systems, Inc. is releasing a powerful new hard disk controller package for the Z-100 computer line. It looks just like the Z-217 controller to the Z-100, but has added features and is less expensive. Call (619) 560-1272 for more information. ✱

For the Z-151

**WILDFIRE
IS COMING!**



**Wildfire is the speed
mod for the Z-151 / 161**

IT'S NEW — it's more than just a
crystal change

IT'S FAST — actually increases
processor efficiency

IT'S FROM —

Software Wizardry
THE MAGIC TOUCH

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

DISCOVER WHAT 10 SOFTWARE INSIDERS ALREADY KNOW!

1. "...a little company that has succeeded over the last five years in giving its customers a lot of value for their money.." - **Richard O'Reilly, syndicated computer columnist**
2. "All the firm's programs are graced with features that the entire industry should be required to provide. Their manuals are the most honest in the industry." - **T. R. Reid, syndicated computer columnist**
3. "...the two most important attributes of Software Toolworks' programs: they're priced low, and they work well." - **The Washington Post**
4. "Unlike the software of many publishers these days, none of The Software Toolworks' programs are hobbled by copy protection, nor does the company try to tie up its customers with mumbo-jumbo licensing agreements.." - **The Los Angeles Times "Computer File"**
5. "I don't know how The Software Toolworks stays in business charging such low prices for such nifty products..." - **David Gerrold computer columnist and science fiction author**
6. The Software Toolworks C/80 compiler is "...the best software buy in America..." - **Microsystems**
7. "It would be the best of all possible worlds if some magic diskette could be made that would transform any hardware in such a way as to receive The Software Toolworks' numinous software!" - **Wm. F. Buckley, Jr.**
8. "The Software Toolworks has a whole raft of programs that work, and (the company) sells them for reasonable prices. I strongly recommend that you get their catalog." - **Dr. Jerry Pournelle computer columnist "Byte" and science fiction author**
9. "One of the original pioneers in low cost, high quality software, The Software Toolworks publishes over 50 solid products for MS-DOS and CP/M under \$60." - **Walt Bilofsky, chairman of The Software Toolworks**
10. "All products are shipped within 24 hours, come with a money back guarantee, and free technical support is available (though you'll hardly need it)!" - **Joseph Abrams, president of The Software Toolworks**

SEE US AT THE WEST COAST COMPUTER FAIRE!

MEDIA MASTER PLUS™ - Now you can run most CP/M programs on your IBM PC or compatible *without* additional hardware. *Media Master™* lets you read, write and format over 140 different double density CP/M formats. *ZP/EM™* (included) emulates an 8080 processor to execute CP/M software at machine language level giving you the equivalent of a 1 MHz 8-bit computer. Includes terminal emulation for Osborne, Kaypro, Heath/Zenith and DEC VT52 displays. Requires 192K RAM. For IBM PC and compatibles. - **\$59.95**

TOOLWORKS C™ - Ideal for learning this popular programming language, the full featured *TOOLWORKS C* compiler is powerful enough for professional program development. 8-bit CP/M version proven in over 5 years of use to be among most efficient and reliable. MS-DOS version produces relocatable object files (.OBJ), includes complete library source code, supports all version of DOS. An outstanding value. - **\$49.95**

TOOLWORKS C MATHPAK™ - Add true 32-bit floating point and signed integer data types to the *TOOLWORKS C* compiler. Speed programming tasks with a complete library of mathematical functions (source) included. MS-DOS version supports the 8087 for true number-crunching power. - **\$29.95**

AIRPORT™ - You are an air traffic controller with radar screen filled with aircraft under your guidance. See how long you can last without cracking. Tougher than the real thing! CP/M and MS-DOS. - **\$19.95**

LISP/80™ - Experiment with artificial intelligence programming! Over 75 LISP functions. Includes source for an editor written in LISP and several demonstration programs. CP/M and MS-DOS. - **\$39.95**

READING PROFESSOR™ - Boost your reading speed while increasing comprehension and retention. Self paced lessons show quick improvement with immediate feedback. Includes 48 practice reading passages and other exercises. Developed by skilled educators, this program tracks your progress at reading rates up to 2,500 wpm. Additional libraries available for Junior High, College/Adult, and Professional levels at \$15.95 each. For IBM PC, most CP/M-80 computers. - **\$59.95**

COMPUTER CHEF® - Original computerized cookbook and recipe file. Data base of over 70 recipes with space to add your own. Computer selects recipes by ingredient, keyword, or title then displays or prints your selection. Scale recipes by portions or ingredient quantities. CP/M and MS-DOS. - **\$29.95**
WOK TALK™ (100 Chinese recipes) - **\$29.95**
CHOCOLATE BYTES™ (100 chocolate recipes) - **\$29.95**

WHAT'S FOR DINNER™ (200 recipes for use with above programs) - **\$19.95**

THE ORIGINAL ADVENTURE™ - Explore the dangerous Colossal Cave for treasures. Endure mystical spells, overcome adversaries. The only official version, uncut, endorsed by its creators, Crowther and Woods. CP/M and MS-DOS - **\$19.95**

MYCHESS® - Award-winning world class chess program, featuring nine skill levels and over 850 opening moves. Prints moves and saves games in progress. Outstanding graphics display board on IBM PC and most CP/M computers. - **\$34.95**

ELIZA - The pioneering artificial intelligence program, *Eliza* is a "psychiatrist" that carries on a conversation in plain English. Use authentic script, expand it or develop your own. CP/M and MS-DOS. - **\$24.95**

The Software Toolworks®

14478 Glorietta Drive Sherman Oaks, CA 91423



TO ORDER CALL:

1-800-223-8665; in CA 1-800-228-8665

U.S. and Canada add \$2.00 per order s/h. Overseas orders add \$5.00 per order. CA residents add 6.5% sales tax. Dealer inquiries invited.

High Resolution Graphics With CP/M-85

The Final Chapter

R. B. Owens

1052 Felix Avenue
Windsor, Ontario
Canada N9C-3L5

Back in February of 1985, I published my first article on high resolution graphics with CP/M-85 on the Z-100/H-100. I hope this article stirred the imaginations of some of you. I thank those who sent me the kind letters. I think that The Heath Users' Group are special people and the letters I received have proven it.

After months of hacking, I have come up with the final version of the PLOT program. It represents hundreds of hours of refining and modifying, and I am satisfied with this final version. It is a significant improvement over the one in the February issue and now supports full color operation, as well as the C80 compiler. I have a tried and true method of loading the plot routine from your MBASIC programs which you can also use to load any other assembler subroutines you might have.

This program will only work with CP/M-85 version 2.2.103. The reason is that this version has two subroutines in the bios which earlier versions do not have. These are specifically the PEEK and POKE functions which allow the 8 bit microprocessor (8085) to have access to memory outside the usual 64 kilobyte area allotted to it. The Z-100 video ram is one such unaccessible area and without the aid of the PEEK and POKE functions, high resolution graphics displays would be impossible when running 8 bit CP/M programs.

You may have noticed the large table of Y offset values at the end of the listing. This table eliminates some really time consuming divide and multiply steps which would otherwise be necessary. The X offset calculation has been revamped for maximum speed. In Monochrome mode, this entire program takes approximately the same amount of time to execute as either the PEEK function or the POKE function, so further refinement is both difficult and unnecessary. In Color mode, the routine slows significantly because of the 3 PEEK and 3 POKE calls necessary to access all banks of video ram.

If you will be using this subroutine with MBASIC, then follow these steps first. First, place ASM.COM, DDT.COM, PLOT.ASM and your favorite editor on a disk.

1. Using editor, set MBASIC Flag to TRUE, C80 Flag to FALSE.
2. Assemble Plot Program. Ex: ASM PLOT.AAZ.
3. Load DDT Program.
4. -IPLOT.HEX<ret>.
5. -R1B00<ret>.
6. -G0<ret>.
7. A>SAVE 3 PLOT.LOD<ret>.

You now have a small, easy-to-load file on your disk named 'PLOT.LOD'. Use this small routine to load this file to memory from disk, remembering first to load MBASIC with the upper memory limit switch /M:&HE600.

```
10 PLOT%=&HE600
20 OPEN"R",1,"PLOT.LOD"
30 FIELD#1,128 AS X$
40 GET#1
50 IF EOF(1)THEN 110
60 FOR X%=0 TO 127
70     POKE PLOT%+B%+X%,PEEK(VARPTR(#1)+X%)
80 NEXT
90 B%=B%+128
100 GOTO 40
110 CLOSE
120 '           Start Your Program Here
```

If you are using the PLOT routine with C80, then it will really shine, especially in Monochrome mode.

1. Using Editor, set C80 To TRUE, MBASIC to FALSE.
2. Assemble with M80.

3. Add to your C library or Link with your C program with L80.

```

*****
*                               *
*           PLOT SUBROUTINE     *
*                               *
*           R.B.Owens          *
*                               *
*****

```

An 8080 Assembler Color Graphics Driver For The Z/H100 Computer Using CP/M 85 2.2.103

This Program Represents Hundreds Of Hours Of Combing And Refining. I Hope It Can Be Of Use To Fellow Heath/Zenith Users.

Purpose: Plot a Point on the Screen at X(Hor), Y(Ver), in C(Color)

Hardware: Z100 Computer, any Configuration
Software: CP/M 85 2.2.103, The Software Toolworks C80 C Compiler

Or Microsoft Interpreter Basic

Assemble: With M80 Microsoft Macro Assembler. If Calling From MBASIC Interpreter Than Assemble With Cp/m ASM. We Will Need The HEX File And DDT To Get Usable PLOT.LOD File

Link: With L80 Microsoft Linker Or Place Into Your Library

Options: Color Or Monochrome Operation (Faster In Monochrome) Call From C80 or MBASIC.

Use: From C80, plot(x,y,c);
From Mbasic, CALL PLOT(X,Y,C),

Miscellaneous Equates

```

FALSE EQU 0
TRUE EQU NOT FALSE
COLOR EQU FALSE
C80 EQU FALSE
MBASIC EQU TRUE

```

Color Mapping

Color Mode Monochrome mode

```

0 - Black      0 - Black
1 - Blue       1 - White
2 - Red        2 - White
3 - Magenta    3 - White
4 - Green      4 - White
5 - Cyan       5 - White
6 - Yellow     6 - White
7 - White      7 - White

```

Z100 Video Ram Equates

```

BRAM EQU 0C000H ;Blue Video Ram Segment
RRAM EQU 0D000H ;Red Video Ram Segment
GRAM EQU 0E000H ;Green Video Ram Segment
BBIT EQU 00000001B ;Blue Enable Bit
RBIT EQU 00000010B ;Red Enable Bit
GBIT EQU 00000100B ;Green Enable Bit
VRPORT EQU 0D8H ;Video Ram Control Port
VRCOL EQU 01111000B ;Enable Vram, Color Mode
VRDEF EQU 00001000B ;Vram Default Mode

```

CP/M 85 - 2.2.103 Equates

Note - CP/M 2.2.100 Does Not Have The Bpeek And Bpoke Functions Required For Operation Of This Plot Subroutine. Other Versions May or May Not, Check Your Bios85.asm Listing If You Want To Be Sure.

Because CP/M-85 Is So Machine Specific And Because I Can't Imagine Someone With A 48K Environment I Have Pre-Determined The Addresses Of The Bpeek & Bpoke Functions For Speed These Addresses Here Are Only Valid For Version 2.2.103. Other Versions May Have A Different Warm Boot Address Check 1 & 2 Of Memory For Your Warm Boot Address

```

WBOOT EQU 0F703H ;Warm Boot Address CP/M-85 2.2.103
BPEEK EQU WBOOT+3FH ;Bpeek Address
BPOKE EQU WBOOT+42H ;Bpoke Address

```

Note Bpeek and Bpoke functions do not disturb (BC), (HL) registers Found This out by Trial and Error

Begin Plot Program

Get Passed Color, Y And X Values
We Want (BC) = X Value, (L) = Y Value, Variable COLR = Color Value

IF C80

C80 Caller Places Data On Stack Using (HL) Registers

```

<SP> DW Return Address
      DW Color Value (0 - 7) ; (HL) Also Has Color Value
      DW Y Value (0 - 224)
      DW X Value (0 - 639)

```

PLOT:

```

MOV A,L ;Place Passed Color Value Into Local Area
STA COLR
LXI H,7 ;Point At X Value High Byte
DAD SP
MOV B,M ;(BC) = X Value
DCX H
MOV C,M
DCX H ;Point At Y Value Low Byte (Always < 256)
DCX H
MOV L,M ;(L) = Y Value

```

ENDIF

IF MBASIC

Microsoft Caller Places Data Addresses In Registers

```

(BC) Color Value Address
(DE) Y Value Address
(HL) X Value Address

```

ORG 0E600H ;Place Program In Highest Possible Memory
;Remember To Load MBASIC /M:&HE600

DDTBIAS EQU 100H - \$;Use This Value As The Bias To Load Program
;Under DDT

```

LDAX B ;Get Color
STA COLR
MOV C,M ;(BC) = X Value
INX H
MOV B,M
LDAX D ;Get Y Value
MOV L,A ;(L) = Y Value

```

ENDIF

Get Y Offset Value From Table

```

MVI H,0 ;Ready (HL) For Double Add
DAD H ;Y * 2 To Point Into Y Offset Address Table

```

```

LXI D,YOFTBL ;(DE) = Y Offset Table
DAD D ;Point At Y Offset
MOV E,M
INX H
MOV D,M ;(DE) = Y Offset
; Calculate X Offset From X Value In (BC)
; X Offset = X / 8
; Quick Divide By 8 (3 Right Rotates / 2 / 4 / 8)
MOV A,C ;Move X Value Low Byte To A
ANI 00001111B ;Isolate Remainder Bits
MOV L,A ;(L) = Remainder
MOV A,B ;/ 2
RAR
MOV B,A
MOV A,C
RAR
MOV C,A
ANA A ;Clear Carry
MOV A,B ;/ 4
RAR ;(B) Is Now Empty
MOV A,C
RAR
ANA A ;Clear Carry
RAR ;/ 8
; Total Vram Segment Offset = Y Offset + X Offset
ORA E ;Or In Y Offset Low Byte
MOV E,A ;(DE) = Offset For Bpeek And
; Bpoke
; Get Pixel Offset From Table Using X Offset
; Remainder In (L)
LXI B,POFTBL ;(BC) = Pixel Offset Table
; Address
MVI H,0 ;Ready (HL) For Double Add
DAD B ;Have Bit Offset Loc In (HL)
MOV B,M ;(B) = Desired Pixel
IF COLOR
IN VRPORT ;(A) = Current Vrport Byte
PUSH PSW ;Save Current Setting
MVI A,VRCOL ;Enable Vram, Color Mode
OUT VRPORT
; Blue Driver
; On Entry: (B) = Bit to Poke
; (DE) = Offset in Segment
LXI H,BRAM ;(HL) = BlueRam Segment
PUSH D ;Save Offset
CALL BPEEK ;(A) = Byte Now There
POP D ;Restore Offset
MOV C,A ;(C) = Byte There Now
LDA COLR ;(A) = Desired Color
ANI BBIT ;Shall We Set Blue Color Bit?
MOV A,B ;(A) = Desired Bit
JNZ BBITON ;Yes - Set Bit On Else
CMA ;Negate Desired Bit And
ANA C ;Mask Bit Out Of Byte There Now
JMP BDONE
BBITON ORA C ;Or It With Byte There
BDONE MOV C,A ;(C) = New Byte To Poke
PUSH D ;Save Offset
CALL BPOKE ;Display New Byte Of Bits
POP D ;Restore Offset
; Red Driver
; On Entry: (B) = Bit to Poke
; (DE) = Offset in Segment
LXI H,RRAM ;(HL) = Red Ram Segment
PUSH D ;Save Offset
CALL BPEEK ;(A) = Byte Now There
POP D ;Restore Offset
MOV C,A ;(C) = Byte There Now
LDA COLR ;(A) = Desired Color
ANI BBIT ;Shall We Set Red Bit?
MOV A,B ;(A) = Desired Bit
JNZ RBITON ;Yes - Set Bit On Else
CMA ;Negate Desired Bit And
ANA C ;Mask Bit Out Of Byte There Now
JMP RDONE
RBITON ORA C ;Or It With Byte There
RDONE MOV C,A ;(C) = New Byte To Poke
PUSH D ;Save Offset
CALL BPOKE ;Display New Byte Of Bits
POP D ;Restore Offset
; Green Driver
; On Entry: (B) = Bit to Poke
; (DE) = Offset in Segment
LXI H,GRAM ;(HL) = Green Ram Segment
PUSH D ;Save Offset
CALL BPEEK ;(A) = Byte Now There
POP D ;Restore Offset
MOV C,A ;(C) = Byte There Now
LDA COLR ;(A) = Desired Color
ANI CBIT ;Shall We Set Green Bit?
MOV A,B ;(A) = Desired Bit
JNZ GBITON ;Yes - Set Bit On Else
CMA ;Negate Desired Bit And
ANA C ;Mask Bit Out Of Byte There Now
JMP GDONE
GBITON ORA C ;Or It With Byte There
GDONE MOV C,A ;(C) = New Byte To Poke
CALL BPOKE ;Display New Byte Of Bits
POP PSW ;Get Old Vrport Setting
OUT VRPORT ;Restore
RET ;Return To C Main
COLR DB 0 ;COLR Variable
ENDIF
IF NOT COLOR
MVI A,VRDEF ;Vram Default Mode
OUT VRPORT
; Monochrome Video Driver - Uses Green Ram Only
LXI H,GRAM ;(HL) = Green Ram Segment
PUSH D ;Save Offset
CALL BPEEK ;(A) = Byte Now There
POP D ;Restore Offset
MOV C,A ;(C) = Byte There Now
DB 3EH ;(MVI A,Immediate) INSTR
COLR: DB 0 ;COLR Variable
ANA A ;Is Black?
MOV A,B ;Get Desired Bit Before
; Decision
JZ BITOFF ;Yes - Turn Bit Off
ORA C ;Else Turn Bit On
MOV C,A ;(C) = New Byte To Poke
JMP BPOKE ;Display New Byte And Implicit
; Return
BITOFF: CMA ;Complement Pixel Mask
ANA C ;Mask Bit Out Of Pixel Byte
; Now There
MOV C,A ;(C) = New Byte To Poke
JMP BPOKE ;Display New Byte And Implicit
; Return
ENDIF
; Pixel Offset Table
POFTBL: DB 10000000B
DB 01000000B
DB 00100000B
DB 00010000B
DB 00001000B
DB 00000100B

```

DB 00000010B
DB 00000001B

Y Offset Table

Instead Of Repeatedly Calculating Y Offset, I
Chose To Use This Table It Sure Speeds Up The
Plot Subroutine.

YOFTBL:

DW 0000H,0080H,0100H,0180H,0200H,0280H,0300H,0380H,0400H
DW 0800H,0880H,0900H,0980H,0A00H,0A80H,0B00H,0B80H,0C00H
DW 1000H,1080H,1100H,1180H,1200H,1280H,1300H,1380H,1400H
DW 1800H,1880H,1900H,1980H,1A00H,1A80H,1B00H,1B80H,1C00H
DW 2000H,2080H,2100H,2180H,2200H,2280H,2300H,2380H,2400H
DW 2800H,2880H,2900H,2980H,2A00H,2A80H,2B00H,2B80H,2C00H
DW 3000H,3080H,3100H,3180H,3200H,3280H,3300H,3380H,3400H
DW 3800H,3880H,3900H,3980H,3A00H,3A80H,3B00H,3B80H,3C00H
DW 4000H,4080H,4100H,4180H,4200H,4280H,4300H,4380H,4400H
DW 4800H,4880H,4900H,4980H,4A00H,4A80H,4B00H,4B80H,4C00H
DW 5000H,5080H,5100H,5180H,5200H,5280H,5300H,5380H,5400H
DW 5800H,5880H,5900H,5980H,5A00H,5A80H,5B00H,5B80H,5C00H
DW 6000H,6080H,6100H,6180H,6200H,6280H,6300H,6380H,6400H
DW 6800H,6880H,6900H,6980H,6A00H,6A80H,6B00H,6B80H,6C00H
DW 7000H,7080H,7100H,7180H,7200H,7280H,7300H,7380H,7400H
DW 7800H,7880H,7900H,7980H,7A00H,7A80H,7B00H,7B80H,7C00H
DW 8000H,8080H,8100H,8180H,8200H,8280H,8300H,8380H,8400H
DW 8800H,8880H,8900H,8980H,8A00H,8A80H,8B00H,8B80H,8C00H
DW 9000H,9080H,9100H,9180H,9200H,9280H,9300H,9380H,9400H
DW 9800H,9880H,9900H,9980H,9A00H,9A80H,9B00H,9B80H,9C00H
DW 0A00H,0A80H,0A100H,0A180H,0A200H,0A280H,0A300H,0A380H,
0A400H
DW 0A800H,0A880H,0A900H,0A980H,0AA00H,0AA80H,0AB00H,0AB80H,
0AC00H
DW 0B000H,0B080H,0B100H,0B180H,0B200H,0B280H,0B300H,0B380H,
0B400H
DW 0B800H,0B880H,0B900H,0B980H,0BA00H,0BA80H,0BB00H,0BB80H,
0BC00H
DW 0C000H,0C080H,0C100H,0C180H,0C200H,0C280H,0C300H,0C380H,
0C400H

END



FREE MEMORY & SOFTWARE FOR YOUR ZENITH PC COMPUTER

That's right! You get 128K of RAM plus RAM disk and Print Spooler Software absolutely **FREE!** when you purchase the TECMAR Captain PC Enhancement Board. This board works with any PC compatible system including the Z150, Z160, and Z158. Also included are a Serial Port, Parallel Port and Clock/Calendar.

List Price of Board with 256K Ram
\$399.00

GENERIC SPECIAL PRICE with 384K RAM
\$249.00

Call or write for more information
or a

FREE Product Catalog

For Faster Delivery
Call 906-249-9801



Add 2% of total cost of order for shipping and handling (minimum \$4.00) and 4% sales tax for Michigan residents. C.O.D. shipment available for additional \$2.00 charge.

SEND TO:

GENERIC COMPUTER PRODUCTS, INC.

POB 790, DEPT. 46R
MARQUETTE, MICHIGAN 49855

768K Z-100™ WITH OUR ZMF100™ PACKAGE

**\$65
EACH**

Our ZMF100™ modification package lets you plug 256K RAM chips into your "old" motherboard Z-100™ computer. Up to 768K (27 RAM's) on the part number 85-2653-1 motherboard.

Simple, mostly plug-in installation requires no trace cutting or funny assembly tricks. Fully reversible. No permanent changes to the motherboard.

Requires disassembly of the computer and some easy soldering to connect the pin-1's of the memory sockets together. All materials (except memory chips and solder) are supplied along with complete, detailed instructions.

Price: \$65 each.

Quantity Discounts. UPS Shipping Free.
Order Direct or See Your Local Heath/Zenith Dealer.
Literature Available. VISA or Mastercard Accepted.

FBE

FBE Research Company, Inc.
P.O. Box 68234
Seattle, Washington 98168
(206) 246-9815

FILE MANAGER UTILITY

- Easily executes any MS-DOS command against a list of files (including batch commands and piping).
- Displays and sorts files by name, extension, size or date.
- Applies MS-DOS commands against a file list in a full-screen editor fashion.
- One of the most useful utilities you can buy. You may find that you will never use the "dir" or "zdir" command again.

**A Must if your system has
numerous files.**

Requirements

MS-DOS Version 2 or greater. Zenith Z-100 computer with or without color memory.

\$44.95

Woodrock Software
c/o Paul Joselow
Spruce Lane & Furnace Dock Road
Peekskill, NY 10566

Phone (914) 739-0192

A Winchester For The '89

Part Three

Peter Ruber

P.O. Box 502

Oakdale, NY 11769

Quick Watson, The QuikStor

Quikstor isn't so much a hard disk subsystem as it is a unique Winchester operating system for the H-89 and H-8 computers, boasting both CP/M and HDOS software. In fact, it is, as of this writing, the only program that supports HDOS. I've noticed in the Annual BUSS Directory that several firms offer HDOS drivers for the Z-67 board, but all requests for information have fallen on deaf ears. This leads me to the conclusion that these are not viable products and certainly not worth investigating further.

Quikstor is the brainchild of Henry Fale of Quikdata Computer Services, Inc., which is also the parent company of H-SCOOP, the popular monthly newsletter for the Heath/Zenith community. Although Quikdata has been a Zenith Data Systems distributor for many years, and a distributor for Magnolia Microsystems, Inc., they were concerned about the limited amount of Winchester software support and the lack of good documentation available to H-89 (and H-8) owners. They also knew that there was a sizable block of H-89 and H-8 computer owners who would welcome a Winchester software package that included HDOS, and set out to fill this void in 1982.

After many months of discussions to establish a framework for the scope of the software, Quikdata contracted with Dean Gibson of Ultimeth Corp. to write the HDOS partition and Ray Livingston of Livingston Logic Labs to write the CP/M partition software. Both men were among the very early independent hardware, firmware and software supporters of the H-89 and H-8, and understood the potentials and limitations, of these machines.

The primary ground rule was that the Quikstor Winchester Software would support only the Heath/Zenith implementation of CP/M and HDOS, including the MTR90 Monitor ROM and the equivalent ROMs from Magnolia, because the Quikstor package used both the Heath/Zenith Z-67 and Magnolia 77320 SASI host adaptors. The Quikstor software package was bundled as a complete subsystem with the Xebec S1410 hard disk controller, and Tandon TM 503 Winchester drive in a heavy-duty case with a switching power supply, fan and built-in line filter, and demonstrated for the first time at HUGCON II, in 1983.

It was an appropriate time for launching such a product because the prices of Winchester drives had been steadily dropping to the point where the serious computerist, as well as the hobbyist could afford to purchase a high-speed, high storage hard disk system. But a number of H-89/H-8/H-19 owners encountered problems using the Quikstor. They had acquired various enhancements in terms of upgraded Terminal Logic ROMs and enhanced Monitor ROMs in conjunction with assorted hardware from various vendors, and Quikstor would not work with many of them. Some of these TLB ROMs stripped out the ANSI code which was detrimental to the successful operation of the Quikstor hardware/software package.

The original authors and Quikdata issued several minor revisions to the software over the next year to compensate for some of the different firmware on the market. Finally, in the fall of 1984, Quikdata initiated a complete rewriting of the software to incorporate the following features:

- Support of multiple Winchesters and multiple sizes
- Continued support for HDOS 2.0
- Continued 2/4 MHz support
- CP/M 2.2.03 and 2.2.04 with and without ZCPR for all standard H/Z drives
- New improved partition and test utilities
- Up to four active CP/M partitions on line at one time
- Support for some TLB ROMs without ANSI code (some still don't work and Quikdata should be consulted on the compatibility of the substitute TLB ROMs you may have).
- Support for the H/Z-29 terminal (which needs all the help it can get because it promised H-19 compatibility, but didn't fulfill this obligation and required software handshaking which the H-19 did not).
- Allow user selection of drive sizes and types

Considering that the Quikstor has followed a three-year enhancement evolution, it is safe to believe that the final version

(Revision 5) published in May of 1985, will be the definitive execution of this software. One of the considerations in rewriting the software was to allow the user to select hard disk drives of different sizes and types because many Quikdata customers expressed the expertise and interest in being their own integrators.

The current revision became a nightmare of coordinating and testing because both Dean Gibson and Ray Livingston had sold their H-89 and H-8 equipment. Dean went IBM and Ray acquired a Zenith 150. Thus, the burden of testing and debugging fell on Henry Fale and the staff at Quikdata. Chief obstacles in rewriting the code centered on the inclusion of the Z-29 as an alternate terminal, and making the software compatible with some popular TLB ROMs, even though the ANSI code had been stripped out or modified. Based on some of the background information supplied to me, I suspect that the current revision was probably more expensive to produce than the original version. Some debugging was so extensive that equipment had to be supplied to both Ray and Dean, and then tested and checked out during marathon long-distance phone sessions using an H-8 front panel for debugging.

The incompatibility among many TLB ROMs stifled a fair amount of new hardware and software development for the H-89 because the respective vendors could not (or would not) cooperate among themselves to adhere to a common standard. I recall reading in either H-SCOOP or BUSS a few years ago that one major Heath/Zenith support vendor offered to act as a clearing-house for the establishment of hardware and software standards so that related products would work in all systems. Had some of these firms banded together they might have had greater success with their products or possibly avoided going under from the costly expense of technical support.

Compatible ROMs

In order to use the Quikstor Winchester software, you must have the Heath/Zenith MTR90 Monitor ROM (444-84 or 444-142) at U518 and the secondary address decoder (444-83) at U516. The only equivalent Monitor ROMs supported by Quikdata are the -

Ultimeth	MTR-4K
Magnolia	444-84B
Kres	KMR-100

because they are fully functional with all Heath/Zenith CP/M and HDOS software. If you purchased the Z-37 double-density controller, the upgraded ROMs will have been included in your kit.

The Terminal Logic Board requires the installation of the original Heath/Zenith ROMs. On the older H-19/H-89, these are located at U422 and U430. On the newer H-19A/H-89A, they're to be found U437 and U445. Sorry, the Watzman/HUG ROM set will not work with Quikstor, as I found out when I tried to boot the Quikstor subsystem and software for the first time. Early releases of Quikstor would not, as I mentioned earlier, support any replacement TLB ROM that either stripped out the ANSI code or modified it. The current version supports both ANSI and Heath codes and will "automatically switch to the code that the terminal can handle by polling the terminal upon activating and sensing its needs." This means that the UltraROM and the SUPER19 ROM may work with Quikstor, but Quikdata will not support any ROMs other than Heath's.

On the related hardware side, the software will function with the D-G Super 89 replacement CPU board. It will also work with an

H-8 computer system that has a Z-80 CPU card and any Trionyx or D-G boards. If your '89 has been modified for 2MHz/4MHz operation, the software will Boot at 4MHz. But you must have a Monitor ROM capable of booting at 4MHz, such as the Kres KMR-100 or the Ultimeth MTR-4K. Based on testing and experience, Quikdata insists that the only "technically sound" 2/4MHz hardware modification is the Kres module. There are some reliable 8MHz modifications available for the H-8, and the Quikstor software will boot at this speed without changing the software. However, Quikdata will not provide technical assistance for any speed modifications not specifically listed in their manual.

Partitioning

Two standard subsystems were recently offered by Quikdata: a 15 MB system using the Tandon TM503 19.14 MB (unformatted) drive; and a 31 MB system with the Rodime 203E 40 MB (unformatted) drive. While this series was in preparation, I learned that Quikdata was phasing out the 15 MB Tandon system because the manufacturer had discontinued production of this drive. It has been around for a few years and is one of the full-height 5.25" drives that requires a larger case and a hefty power supply. It is being replaced by Seagate ST225 half-height drive offering 20 MB of formatted storage for the same price as the original 15 MB subsystem, and in a smaller cabinet.

The nice feature of the Quikstor software is that it was designed to have both CP/M and HDOS resident on the same drive. Each partition can be FORMatted or INITialized and then SYSGENed so that you can Boot from any partition. You are allowed a total of 15 user-selected partitions. The default partition assignments as shipped by Quikdata, unless the purchaser specifies otherwise, are:

For the QS15 System —

PARTITION #	NAME	SIZE MEGS
0	CP/M 0	5.0 MB
1	CP/M 1	5.0 MB
2	HDOS 0	1.2 MB
3	HDOS 1	1.2 MB
4	HDOS 2	1.2 MB
5	HDOS 3	1.2 MB

For the QS31 System —

0	CPM0	8.0 MB
1	CPM1	8.0 MB
2	CPM2	4.0 MB
3	CPM3	4.0 MB
4	HDOS0	1.2 MB
5	HDOS1	1.2 MB
6	HDOS2	1.2 MB
7	HDOS3	1.2 MB
8	HDOS4	1.2 MB
9	HDOS5	1.2 MB

For the QS20 System —

0	CPM0	8.0 MB
1	CPM1	8.0 MB
2	HDOS0	1.2 MB
3	HDOS1	1.2 MB
4	HDOS2	1.2 MB
5	HDOS3	1.0 MB

In order to get their customers on line with a minimum of effort, Quikdata PARTITIONS and FORMATS all drives, so that the user merely has to SYSGEN each respective partition. Under HDOS this procedure takes about 5 minutes. The setting up of CP/M BIOS will take about 30 minutes. In a few moments, we will devote some space to the partitioning procedure and walk you through the Menu Screens. The reason for this is to give you an overview and some familiarity of how Quikstor works should you ever decide to repartition the drive. If you don't, all you will ever work with is the Main Menu Screen.

The QS15 can hold a staggering amount of data, when you consider that a 1.2 MB HDOS partition is the equivalent of two 96-tpi quad-density disks. During the 90-day period, I worked with the QS15 subsystem, I threw every working file onto the drive and didn't succeed in even filling one HDOS partition. The remaining storage on my CP/M partition paled the amount of data I put on it. It is conceivable that, under strenuous use, your floppy drives could be idle for months on end.

Partitioning is accomplished by the SASIX utility. There are CP/M and HDOS versions (they function identically) and separate versions for CP/M 2.2.03 and 2.2.04. As I explained in the first part of this series, a partitioning utility takes the available hard disk storage and divides it into manageable chunks of space that are practical to work with. Anywhere from 20-25% of space is lost in this process because of ID headers that are used to identify the various cylinders on the drive.

Depending on the size of the partitions, you must be cognizant of the "minimum size" that a file will occupy. Under HDOS this is known as a "cluster". A cluster is a group of sectors which is the smallest size available for a file. On an SSD 100k disk, the cluster size is 2k. On a large partition, the cluster size could be more than 20 sectors. This means, if you have a one sector file that physically occupies only one sector on the partition, the directory will log it as 20 sectors. This will explain, in part, why the HDOS partitions on the foregoing charts were prepared for 1.2 MB each, rather than a larger size. A 2.5 MB HDOS partition has a cluster size of 38 sectors, while a 5.0 MB HDOS partition's cluster size is a whopping 76 sectors. Thus, a 1.2 MB HDOS partition makes reasonably efficient use of directory space without being too wasteful.

CP/M also has limitations on the maximum number of directory entries which depend on the size of the partition. Quikstor allows for a maximum of 8 MB per partition, which is the CP/M limit (not true for CP/M+). The cluster sizes for CP/M are as follows:

0 to 256k = 1k minimum per file
256k to 2 MB = 2k minimum per file
2 MB to 8 MB = 4k minimum per file

0 to 256k partition = 64 maximum directory entries
256k to 512k partition = 128 max. directory entries
512k to 2 MB partition = 256 max. directory entries
2 MB to 8 MB partition = 512 max. directory entries

If you want to calculate the partition size from the number of allocated cylinders, note that in the case of the 15 MB Tandon, 1 cylinder = 6 tracks. Each track can hold 8k of storage, or 48k for 1 cylinder. The default CP/M partitions of 100 cylinders yield 48000k storage, or 4.8 Megabytes.

For the default HDOS partitions of 25 cylinders, you get 1200k of storage, or 1.2 MB. For estimating purposes, 20 cylinders is equal to 1 MB. Calculating the sector/cluster size of HDOS partitions is accomplished by the following formula:

$$((\# \text{ CYLINDERS}) * (\# \text{ HEADS})) / 8$$

Take this value, round it up to the next highest even integer value, and that is the number of sectors taken by each cluster, which is the minimum sector size for any file. Quikstor's standard partitions are set up with a size of 25 cylinders, which results in a "cluster" size of 20, per the following example based on the above formula:

$$((25) * (6)) / 8 = 150 / 8 = 18.75, \text{ rounded up to } 20.$$

Setting Up The Quikstor Partition

There are two sections in the Quikstor Manual that may cause temporary hyperventilation when you first encounter them. The first is the running of the Partition Utility and then setting up the Quikstor BIOS on your CP/M partitions. The primary rule to remember, if you purchase the Quikstor software for installation on your own Winchester subsystem, is that you can't expect a lot of hand-holding from Quikdata, as you are no longer paying them to be the system integrator. Simply put, you are on your own.

For all the fear that I may have conveyed by the previous paragraph, I assure you that it's not as dreaded as it sounds, because Dean Gibson of Ultimeth Corp. has created an ingenious series of menu-driven screens to walk you through the entire Partitioning program. Furthermore, so much care has gone into the preparation of the manual, that you'll feel like an expert when you're through.

Take any bootable CP/M system disk and transfer the SASIX.COM file from the Quikstor CP/M distribution disk and type SASIX.

The first screen comes up on the CRT and you are requested to provide data on your I/O Port, the Drive, the Controller card, and a couple of other irritating details the software finds quite important. You shift from field to field using an on-screen menu that explains how to use your cursor and editing keys to move around the screen. When you have entered the appropriate data, you can check that the SASI Controller card is working properly by pressing the #5 key. Then press the #6 key to proceed to the next screen.

This is considered to be the Main Screen, because it is here that you will not only establish the size of each of the partitions, but also enter some technical data about the hard disk drive you are using, such as:

Number of Heads
Number of Cylinders
Write Precompensation starting cylinder number
Write Reduce current starting cylinder number
Seek Type
Error Len

Quikdata provides all the necessary data on the hard disk drives they offer in their packaged system. If you have your own drive, make certain that you obtain a manual and enter the correct data.

When you're finished setting up this screen, you can press key #4 and the software can now check to see if your drive is functioning properly by reading one sector on each track of the hard disk drive. This Check Drive feature also has a hidden benefit in that it allows you to park the read/write heads at the inside track if you plan to relocate or ship your drive.

When you are in this screen, you are also able to generate the Magnolia Microsystem's partition table from the Ultimeth partition table. The MMS partition table is different from Ultimeth's, and invoking this function will cause a different set of data to be recorded on the boot track. This section must be used with great care if you are planning to install both the Heath/Zenith and MMS CP/M implementations on different partitions. This is not supported by Quikdata.

Screen Three shows the formatting options and an action code to format specific partitions or the entire disk. Obviously, you will select to format specific partitions which you established while in Screen Two. You can also elect to format an alternate boot track (either track 1, 2 or 3) in the event track 0 ever becomes defective, so that you will be able to boot successfully from any partition. It is worth mentioning that this feature is unique. No other hard disk system integration offers an alternate boot track (including those from the Blue Plague of Boca Raton). If Track 0 goes bad with other systems, the Winchester is unusable.

To set up your HDOS partitions is somewhat simpler, because this is accomplished through the SASIX.ABS program and two drivers: DVDDKGEN.ABS (which is used to set the number of partitions or logical drives on your Winchester hard disk) and DKSASXV3.DVD (which will become the actual device driver to access the partitions. This is a breeze, because it follows all HDOS conventions in establishing and setting up a device through the SET option. You set your number of drives DK0: to DK7: (yes, you can have up to eight HDOS partitions with Quikstor), then SHIFT/RESFT the computer so that when you boot up again, the system will recognize the existence of the DK: devices.

If your HDOS system volume has a DK.DVD driver already on it, you must delete it. Then, copy the DKSASXV.3.DVD driver from the Quikstor HDOS distribution disk to your system volume and rename it DK.DVD. Your new DK.DVD device driver must now be set by typing BYE and rebooting the system. Now, you invoke your SET DK: HELP command and SET your Port Address, Drive Number (if it's the first Winchester on your system, this is 0), the Controller number and, finally, the Partition Category (meaning the number of HDOS partitions you have allowed on this drive). This will allow you in the future to SET additional drives and controllers if you plan on paving the road to Wallet Depletion by daisy chaining multiple drives and controllers from the host adaptor card.

Now you can INIT and SYSGEN each partition separately and load any application programs you plan to work with. I should point out, before I get too far afield, that the SASIX.COM utility under CP/M and SASIX.ABS under HDOS are not only exactly alike, but can be used interchangeably.

In contrast to the simplicity of HDOS, the CP/M section takes on an ominous glow, because you must create a new BIOS and you must exercise care by following the instructions very carefully. The Quikstor CP/M Winchester software distribution disk is chock-full of files with BIOS.SYS files covering all the possible floppy drive combinations that will be used in combination with the Winchester (H-17, Z-37, H-47) under both the 2.2.03 and 2.2.04 version.

Take a blank disk. Format and Sysgen it from a CP/M system volume, making certain that the BIOS.SYS is on the disk. Run MOVCPMxx, which you will have PIPed from your particular version of the original CP/M distribution software. This will become your QSBIOS System Disk. Now boot from this disk and

transfer the appropriate INSTLxx.SUB file from the Quikstor distribution disk. Also, copy the QSPUTSYS.GEN, EX14.COM and CFGPATCH.COM.

Quikdata and Ray Livingston were exceedingly foresighted by including several important Public Domain utilities on the Quikstor CP/M distribution disk:

XDIR.COM — An enhanced directory listing program which displays an alphabetically sorted directory listing complete with the size of each file.

FBAD.COM — A non-destructive disk test and bad blocks lock-out utility.

ZCPR — A CCP (Console Common Processor) replacement which provides several new and useful system commands, as well as enhancing the user interface.

EX14.COM — Replacement for the Digital Research SUBMIT and XSUB batch processors, providing useful enhancements.

So, if you want ZCPR automatically installed on your QSBIOS Disk, copy the ZCPR.COD and ZCPR.REL files to this disk at this time.

Also, copy the DDT.COM, MOVCPM37.COM and CONFIGUR.COM files from the original CP/M distribution software. Now, we will run the installation program on the QSBIOS system disk by typing EX14 INSTL0x (where x is 3 or 4, depending on your CP/M version. When this function has been fully executed by the computer, QSPUTSYS.GEN, MOVCPM37.COM, CFGPATCH.COM, CONFIGURE.COM, ZCPR.COD and ZCPR.REL will be gone. In their place, you will now find QSPUTSYS.COM and QSCONFIG.COM.

Now, you will copy the appropriate BIOS0xxx.SYS to the QSBIOS system disk. For example, the Z-37 version under CP/M 2.2.04 would be BIOS0437.SYS, which you will then rename as BIOS.SYS. Then, you will run QSPUTSYS by typing QSPUTSYS A:BIOS.SYS. No specifier is used. When this program has finished, you will have a bootable Quikstor CP/M disk.

RESET and reBOOT your '89. Configure your system by running QSCONFIG, and make the proper changes to represent your system floppies, printers, terminals, etc. Reset and reboot your system, and place any other important CP/M utilities on your new Quikstor CP/M system volume that you normally use. Now, you can FORMAT and SYSGEN each Winchester partition you set up earlier.

The balance of the Quikstor CP/M documentation provides an explanation of all Error Messages, as well as an explanation of the utility programs on the distribution disk.

When all is said and done, you have a fully operational monster capable of devouring (storing) vast chunks of your work, which, with a little bit of luck on your part, will keep you away from the family for even longer periods of time.

Now — when you first turn on your '89 and you see the familiar H: or MMS: prompts, you type B for Boot and a boot menu appears on the screen. You will stare at this with great fascination, while the blinking cursor awaits your command to enter the partition of your choice. For the first few times it's kind of like an "Open Sesame" experience.

Backing Up Your Winchester Drive

One thing you will have to be concerned with on a periodic basis is the backing up of your Quikstor Winchester files. This can be

an onerous chore at best, especially if you wait until your partitions are at the bulging point.

HDOS partitions do not present too much of a problem because their individual size are the equivalent of two 96-tpi quad-density drives. It's the 4 MB or 8 MB CP/M partitions that can cause grief. Quikdata offers an interesting support package from Stock Software, Inc. called BACKREST, which can do the dirty work for you.

While there isn't sufficient space in this article to give you a full-blown rundown of the BACKREST features, it is worth every penny in terms of time saved. It is an intelligent hard disk to floppy backup and restore utility that can be modified to the user's needs. (Maybe we'll get to a full evaluation later in this series.)

It has the ability to only backup those files that were created or modified since the previous backup. This greatly reduces the number of floppy disks required for each scheduled backup. If you have a multi-user system or a single system that is used by several people, you can code your files in a manner to designate each user and backup a single user's files.

You can instruct BACKREST to ignore certain delineators so that it skips them in a backup procedure, such as *.\$\$\$ files, *.TXT files, etc. Very long files that are too large to fit on a destination disk are split and automatically merged when restored. It can create a management report of what it has done for a permanent record. BACKREST will also serialize all floppy disks that it uses for backup. So, it knows where it put what.

It is a clever utility that does your thinking for you, as well as forgives you when you goof. The documentation is excellent, and I do recommend that you invest in this utility if you use your Winchester extensively.

Conclusion

I cannot equate the Quikstor against the Magnolia CP/M Winchester software on an apples-to-apples basis, because the former utilizes the Heath/Zenith version of CP/M. There are numerous differences that are a matter of personal preference. I

had hoped to include some observations on the Z-67 Winchester software in this series, but Heath's marketing division told me the package was dead and buried. I am hopeful, however, that some kindly soul at Heath will read this and will make an evaluation package available so that I can report on it before this series is concluded.

I liked Quikstor for many reasons. The Operation Manual is well laid out and it includes a smattering of technical info for those who prefer to put together their own subsystem. It also includes technical information on all the drives supported by Quikdata, as well as "questions" and "answers" that the uninitiated person is likely to ask. I didn't try the system at 4MHz because I haven't bothered to speed up my old '89. I'm not as impatient as I used to be.

I liked Quikstor because it supported HDOS. As clumsy as HDOS is, I am fond of it, and it will do things that CP/M doesn't. Unfortunately, all the professional software is in CP/M.

Both Dean Gibson and Ray Livingston have done an outstanding job with the software. Their final salvo to the life of the '89 (and the H-8) is probably their best. Henry Fale and the staff at Quikdata must have sweated countless hours in laying out and writing the documentation. The Quikstor manual is superb. All set-up procedures are presented in a logical sequence that is easily understood. It is probably the only manual I have ever encountered that didn't talk down to me.

Basic pricing information at the time this was prepared is: The QS20 (or QS15 if available) system of Winchester drive, Xebec controller, case with power supply and all cables, is \$995. A second "slave" drive without the Xebec is \$795. The QS31 system is \$1695, and a "slave" drive is \$1495. The Magnolia 77320 SASI host adaptor card and ROMS is \$225. The Quikstor CP/M and HDOS hard disk driver software is \$149 (\$195 if you buy it without the system). BACKREST is \$95. For more information, write to:

QUIKDATA COMPUTER SYSTEMS, INC.
2618 Penn Circle
Sheboygan, WI 53081
(414) 452-4172



QUERY!³ IS YOUR KEY...



TO A WORLD OF INFORMATION!

Organize information your way... easily, affordably and safely. Create the reports, forms, invoices, letters and tables of your dreams; and your requirements. Just sit back and let your computer do all the work. Query³ and support software is, indeed, your key to success. Includes mail list option, MENU, easy-to-follow letter-quality instructions, help screens & 11 programs; Create, Add, View, Purge, Recover, Printer, Search, Sort, Writer, Redesign and Autopro. **And all for only \$99.95.**

Call 919/378-1050 with Visa, MasterCard or Choice number, or send \$99.95 check or money order. COD's & PO's accepted. Please add \$3.00 shipping & handling, and NC residents add 4 1/2% sales tax.

Hoyle & Hoyle

Software at Soft Prices.

716 S. Elam Avenue
Greensboro, NC 27403
919/378-1050

Available in CP/M, HDOS, ZDOS, MSDOS & PCDOS.

"MAKE YOUR Z-150 CP/M COMPATIBLE"

Intersecting Concepts Announces 3 Solutions To Solve Your Computer Incompatibility!

"But will it work on my computer?"

Yes! Finally, there are three easy ways to exchange information, transfer files, and run CP/M software on MS-DOS machines.



1. **MEDIA MASTER™** is our direct disk-to-disk format conversion program. Already an accepted industry standard, this \$39.95* program uses simple screen prompts that lets you read, write and format up to 150 different 5 1/4" diskettes from CP/M, MS-DOS and PC-DOS operating systems. So if you work on a IBM-PC compatible at the office, but use a CP/M computer at home, now you can easily transfer files that would otherwise be "foreign" to your computer's operating system.



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



3. **ACCELERATE8/16™** is also new and dramatically improves the performance of *Media Master Plus* by tailoring the CP/M emulation around a NEC V20 microchip. This user installable plug-in chip simply replaces the 8088 processor in your MS-DOS computer. Once installed, it'll run your CP/M and MS-DOS software much faster. (Speed improvements are roughly 15% faster in MS-DOS and 350% faster in CP/M!) *Accelerate8/16* includes *Media Master Plus*, V20 CP/M Emulation Software, and the NEC chip for only \$99.95!

SEE US AT THE WEST COAST COMPUTER FAIRE — BOOTH 1833!

THE CHOICE IS YOURS!

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

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

And if the execution speed of your CP/M programs is important, there's *Accelerate8/16* for \$99.95, which provides a simple hardware *and* software solution to making your MS-DOS computer CP/M compatible.

TO ORDER

To order *Media Master*, *Media Master Plus*, or *Accelerate8/16*, call 800-628-2828, ext. 629.

All three solutions save you money by eliminating expensive modems and communications software, and none of our products are copy protected.

For additional product and upgrade information contact:



**INTERSECTING
CONCEPTS**

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



Dealer inquiries invited.
* \$99.95 for Dec Rainbow

Modifying The MS-DOS BIOS Of The H-160 To Use A 96-tpi Disk Drive

Daniel Ng
P.O. Box 18070
Pittsburgh, PA 15236

If you don't have a 10 megabyte hard disk with your H-160 or H-150, and you happen to have a double-sided 80 track disk drive like I do, you can have a poor man's hard disk with 0.7 megabyte by patching IO.SYS of Zenith Data Systems' MS-DOS version 2.11.

When I built the H-160, I ordered only one disk drive. I had in mind using the double-sided 80 track disk drive (which is completely plug compatible with the 40 track drive) that I have been using in my H-89. This way, I would have made use of a disk drive that I already had and would have 720 kbyte of storage on a diskette. I thought my experience in doing so might be useful to some members of the Heath Users' Group who would like to do the same, and I would like to hear from someone who might have done it a better way.

After I got the one-drive H-160 working, I plugged the 80 track drive into it, and it worked. However, just as I expected, the H-160 was using only 40 tracks of that drive, i.e. half of the disk was not used. The research I had done up to that point indicated that in order to use the full 80 tracks (720 kbytes) of that disk drive, I just needed to add a device driver to the MS-DOS CONFIG.SYS file.

Enquiries at the local Heath store and the Users' Group about buying such a device driver yielded no result. I have since spent some time studying articles about device drivers. As of this moment, I have learned a little more about device drivers, MS-DOS and the IBM PC than when I began, but I am still not knowledgeable enough to write that device driver.

I am thus forced to look for some quick and dirty methods. The H-160 (like the IBM PC) supports four disk formats (all 40 tracks): 1 side, 8 sectors (S-8); 1 side, 9 sectors (S-9); 2 sides, 8 sectors (D-8); and 2 sides, 9 sectors (D-9). My reading on disk formats indicated that there will be a BIOS parameter block for each disk format in the BIOS (file IO.SYS) to permit each of the disk formats

to be read. It appears that if I am willing to sacrifice one of these formats and substitute the parameters of that format with those corresponding to 80 tracks, 2 sides, 9 sectors (QD-9), I would have achieved my goal. The format I am willing to give up is the S-9, others may have reason to give up a different one instead.

It turned out that the parameters in the blocks are arranged together in a certain order and have well defined values. To be exact, they represent the number of bytes per sector (00 02 for 512 decimal in hexadecimal backward order), the number of sectors per cluster (01 or 02), the number of reserved sectors at the beginning of diskette (01), the number of FAT (02), the number of root directory entries (70 00 or 40 00 in hexadecimal for decimal 112 or 64), the total number of sectors on the disk (D0 02 for decimal 720 for 2 sides, 9 sector diskette or some other values for other disk formats), the disk format ID (FC, FD, FE, FF for S-9, D-9, S-8 and D-8, respectively), number of sectors per FAT (01 or 02), number of heads (sides) (01 or 02), and number of special reserved sectors (00 in all cases). Thus, for the S-9 format, the consecutive bytes of the parameter block would be 00 02 01 01 00 02 40 00 68 01 FC 02 00 09 00 01 00.

Examining the BIOS system file IO.SYS (version 1.07) using DEBUG (this is done by issuing the command DEBUG IO.SYS), I issued the s(earch) command to search for the above string between location 100 and 2000, (S100 2000 00 02 01 01 00 02 40 00 68 01 FC 02 00 09 00 01 00) sure enough I found it and the other parameter blocks beginning at location 017A. I then changed the values corresponding to the FC disk ID to 00 02 01 01 00 02 70 00 A0 05 FC 05 00 09 00 02 00 to signify the QD-9 disk format using the e(nter) command and then saved the changes with the w(rite) command.

You might have noticed that the changes I made to the BIOS corresponds to cluster size 1 sector, and 5 sectors per FAT. In order

to have enough entries in the FAT for all the sectors in the QD-9 format disk, the FAT has to be larger than the 1 or 2 sectors used by Heath and IBM. Three sectors per FAT would have had enough entries to accommodate the more than 700 clusters, if each cluster contained 2 sectors. Instead, I decided to assign 5 sectors to the FAT and 1 sector per cluster to make more efficient use of the disk spaces.

Of course, in order to be able to modify IO.SYS with DEBUG, you must change it from a system file to a non-system (archive) file. This can be done by using the FLAGS utility program or by doing so inside DEBUG by locating the IO.SYS entry in the directory sectors beginning at sector number 5 using the l(oad) command (L 0,0,5,1 for drive A or L 0,1,5,1 for drive B), locate the attribute byte of file IO.SYS (byte number 11, count the first byte as byte 0) and change it with the e(nte)r command to hexadecimal 20 and write it back to disk with the w(rite) command (W 0,0,5,1 or W 0,1,5,1). If you have access to the NORTON utility SM (sector modifying program), you can patch in the changes without having to change the file type.

This is only half the job. Now that the BIOS has been patched, it still remains to modify the disk formatting program FORMAT, so that I can format 80 tracks on a disk with it. Starting with a copy of FORMAT.COM changed to FORMAT80.COM, I again used DEBUG to examine it. I issued the search command to look for the same BIOS parameters of the parameter blocks. I found them at locations beginning at 2AE5. Once I found them, I patched the

parameters corresponding to the format FD (i.e. 2 sides, 9 sectors) beginning at 2AF9 to 00 02 01 01 00 02 70 00 A0 05 FC 05 00 09 00 02 00. The reason for this is, when FORMAT.COM (now FORMAT80.COM) is invoked from the command level, the formatted disk will have the disk ID and disk format parameters at the boot sector and at the two FATs correct for the disk format I specified. The Heath/Zenith supplied FORMAT program has a counter to format 40 tracks. That value is loaded into location 2A96 with the instruction

```
MOV WORD PTR [2A96], 0028
```

beginning at location 2CBF. I changed the 28 at 2CC3 to 50 so that it would format 80 tracks and wrote the changes in FORMAT80.COM back to disk.

I, then, reboot and issue the command FORMAT80, (or FORMAT80/v, or FORMAT80/s) to format, (or to format with verification, or to format with system) on drive B. After copying a small file into the disk in that drive and issuing the directory command, I was rewarded with seeing that there were 711 kbytes in that disk. I then set the IO.SYS file back as a system file. I have not encountered any problem so far.

One of these days, I would like to be able to accomplish the same thing with an intelligent device driver. That way, I should be able to keep the existing disk formats while adding the (QD-9) format. In addition, the 80 track disk drive would behave like a normal second drive in the sense that it can at least read all the other 40 track disk formats, though it may not write to them. *

FOR THE Z-100

Conforms to
IEEE -696 (S-100)

A low-cost high-performance
memory board from
Software Wizardry, Inc.

- Up to 1 megabyte on a single card with parity error checking
- On-board clock/calendar with software
- The only memory card not manufactured by ZDS that is specifically designed for the Zenith Z-100
- Emulates the Z-205 registers for complete present and future Z-100 compatibility
- Runs at 8 MHz with no wait states (when loaded with 150 nanosecond chips)
- Ramdrive software included
- Supports 24 bit addressing

List price: \$395 (unpopulated)
\$520 (with 1 megabyte)



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



Z-100 Drafting Program For Scientific Graphs

Michael D Chafetz, Ph.D.
2525 Dublin
New Orleans, LA 70118

If you act as I used to, you would normally use an analytic graphing program to look at your data. Then, you might send a mock-up to your draftsman for production, or if no such person exists, you might spend the time necessary to draft your own graphs. Well, why waste the time drafting when you can get the benefits of analysis and drafting all in one program?

There are many good analytic graphing programs that will allow you to represent your data quickly and easily. But, there are not many graph drafting programs based on the behaviors you normally employ while drafting for publication or professional presentation. A good graph drafting program will allow you to "size up" your X and Y axes, and to change the representation of data so that you can produce the most aesthetically pleasing graph.

The purpose of this graphing program is to eliminate the drafting necessary to produce professional quality slides or journal quality graphs. The drafting functions permit you to alter the relative lengths of ordinate (Y-axis) and abscissa (X-axis), and to plot points using five different plotting symbols, two connectors (lines or dots), and seven different colors. Options are provided for error bars, histogram bars, and for a regression line with corresponding statistics. The graph may be labeled horizontally or vertically with any alphanumeric character. The final production may be saved in file storage for subsequent use.

The graphing functions were written for use on a Zenith Z-100 with a Star Gemini-10X printer, but portability can be achieved by transcribing the Microsoft BASIC for use on any compatible machine. The Z-100 is advantageous because of the simultaneous screen presentation of color high resolution graphics and alphanumeric labeling.

The program is written in modular form to avoid the space taken up by program remarks. You can readily see the actions of any portion of the program by looking at the line numbers corresponding to the menu selection (x100). For example, the program introduction is contained on lines 0-99; the (1)Axes portion

is contained on lines 100-199; and the (2)Label section is contained on lines 200-299, and so forth. The program listing in Microsoft BASIC (ZBASIC) is available in Listing 1.

The actual graphing portion of the program takes up 1-2K of memory, but the program itself takes up about 7.5K. The increase gives you not only the flexibility necessary for drafting, but also the error trapping to keep the program from bombing if you make a mistake.

Description Of Functions

You initially view a framed plotting arena with a main menu on the 25th line (menu line). The choices in the menu are arranged in an order that reasonably approximates the order of choices made in drafting graphs: (1)Axes (2)Label (3)Data (4)Color (5)Photo/Print (6)File (7)Clear (8)End. You may select functions in any order, and may, for example, wish to change the color of an application before drawing it on the screen. The choice to quit (type Q) is an option available at any time.

(1)Axes

In order to achieve flexibility in graphics and labeling, a few questions are asked about the axes before they are drawn on the plotting arena. The following questions occur in sequence on the menu line before the X-axis is drawn:

Highest X-axis value? ____
Starting X value: ____
Mark every ____ value(s)?
X-length? ____

The first answer provides the X-axis limit, but you also have the option of initializing the X-axis with a value other than 0 (second answer). The answer to the third question provides the spacing for the hash marks, while the answer to the fourth question permits a change in the overall length of the X-axis. The typical

(default) length provides a column's worth of space for each X unit. Error-trapping has been provided so that usual response mistakes do not disrupt the program or the graph. For example, the axis will shorten itself if the values provided would normally lengthen it beyond the plotting arena.

After the X-axis is drawn, the following questions are received:

Highest Y-axis value? ____
Starting Y value: ____
Mark every ____ value(s)?
Y/X ratio?

Answers to these questions produce similar effects on the Y-axis. The last answer permits a change in the length of the Y-axis relative to the length of the X-axis. For example, for some data sets you may wish to draw a Y-axis half the length of the X-axis. The same kind of error-trapping has been provided for typical Y-axis response mistakes.

After both axes are drawn with the correct relative lengths and hashing the main menu appears again on the menu line.

(2)Label

You are now asked whether you want automatic axis numbering: Automatic numbering? <n> ____ . If you respond yes (y), the correct labeling of the hash marks appears on the screen. After hash labeling, or by default (n), a secondary labeling menu appears: Move: <<(4)up (1)dn (3)rt (2)lf>> (5)hlabel (6)vlabel (7)legend (8)Quit. The cursor has appeared in the middle of the plotting arena.

The first four choices are used to move the cursor and are best accessed by using the Zenith keypad. The choices take advantage of the spatial relationship of the numbers on the keypad, with 4 to move up and 1 to move down, and 3 to move right and 2 to move left. Fast positioning of the cursor can be achieved by holding down the fast repeat key (far left lower keyboard) in conjunction with a cursor movement key. The program will not permit movement beyond the frame of the plotting arena.

The (5)hlabel choice is used for horizontal labeling, while the (6)vlabel is used for vertical labeling. After either choice you may type in any alphanumeric character, which will appear on the screen at the cursor location. A carriage return is used to stop labeling. Any of these labels may be erased by positioning the cursor over the mistake and using hlabel or vlabel to type in spaces. The program will not permit labeling beyond the plotting arena.

The (7)legend choice is used to select the plotting symbol(s) that will be used for the data plot. You have a choice of five plotting symbols: (1)O-diam (2)F-diam (3)O-box (4)F-box (5)points. These are (in order): open diamond, filled diamond, open box, filled box, and points. The plotting symbol is placed wherever the cursor has been positioned. Each symbol may be labeled using the hlabel (or vlabel) selection.

The (8)Quit choice is used to return to the main menu.

(3)Data

A short menu is provided to ask whether regular or irregular X-axis spacing is desired: (1)Reg (2)Irreg (3)Qt. The first choice (regular) permits speed in entering data; only the plotted values need be supplied. The second choice (irregular) permits flexibility; any X value may be entered. The third choice returns you to the main menu.

For regular plotting (choice 1), you supply the spacing of the X values and the starting value:

Plot every ____ X value(s)?
...starting with? ____

These choices permit you to plot along the available hashing or to use whatever regularly spaced X values are suitable. A choice of five plotting symbols (open diamond, filled diamond, open box, filled box, or points) or two types of histogram bars (open or filled) is then provided. You are then asked to provide the corresponding Y values.

For irregular plotting (choice 2), you are asked to supply each X-value with a carriage return to enter the value. You may supply up to 250 values. The X-value mode is terminated by entering a Q (quit). The choice of plotting symbols is then provided, after which you are asked to provide corresponding Y values.

As each Y value is entered, the chosen plotting symbol appears at the correct coordinate. This plotting mode automatically shuts off at the end of the X-axis (in regular spacing mode) or after the number of plotted values match the number of X values supplied (irregular mode).

A short menu now appears to ask whether the plotted symbols should be connected by lines or dots: (1)no (2)lines (3)dots (4)qt.

After connecting the symbols, or by default (response #1), you are asked whether error bars are desired. In many fields of research, it is desirable to provide a graphic index of the variability associated with each plotted coordinate. A short question appears:

Error bars? <n> ____

A yes response (y) will elicit a prompt on the menu line:

+/- each point: ____

The program remembers each plotting coordinate in the X value order supplied by the user. You simply supply the standard deviation, standard error, or confidence interval calculated for each point. The error bar appears at each plotted point after each entry. This mode automatically shuts off after the number of error bars corresponds to the number of plotted points.

The program now asks whether a regression line is desired:

Regression line? <n> ____

A yes response (y) will produce the appropriate regression line on the data plot. The correlation coefficient, intercept, and slope appear on the menu line and will remain in view until you strike any key. The program now returns to the main menu.

(4)Color

This function first asks whether a color change is desired:

Change color? <n> ____

The no (default) response will return control to the main menu. A yes (y) response will evoke a menu containing a list of 8 colors: (0)Black (1)Blue (2)Green (3)Cyan (4)Red (5)Magenta (6)Yellow (7)White. All drafting is done in white unless this option is chosen to change the color. Any drafting performed after a selection will be done in the selected color. Only colors 1-7 appear on the screen; the black is the background color. Mistakes can be erased by drafting over the mistake after selecting the (0)Black option. Different color choices appear on a monochrome mon-

itor as slight differences in shade. After a color option is chosen, the program announces that the color has changed and then returns control to the main menu. Sorry, but the printer cannot handle the color (though it will print the graphics you have drawn in any color).

(5)Photo/Print

The first two questions ask whether the arena frame is desired:

Erase frame? <n> ____
 New frame? <n> ____

A photo session mode is now in progress. This simply means that the menu line is erased until you strike any key. The screen is now available for 35mm slide photography. This is best performed in a dark room with the camera affixed (and leveled) atop a tripod. Crisp slides can be produced under these conditions, because there is a considerable amount of light emitted by the Zenith monitor. Low ASA films and high shutter speeds are recommended to prevent the drafted lines from bleeding. Aperture or shutter priority cameras may be used as long as the aperture is wide and the shutter speed is fast. Good results have been obtained using Kodachrome 64 film, but you should remember that the E-6 processing available by using other films can be arranged overnight.

After the photo session (strike any key), the program asks whether a hard copy print is desired. Currently, all values have been set for the Star Gemini 10X printer. If this selection is chosen (default returns to main menu), the program sends the image on the screen directly to the printer. Examples of drafted plots designed and printed with this program are provided in Figure 1. The program now returns control to the main menu.

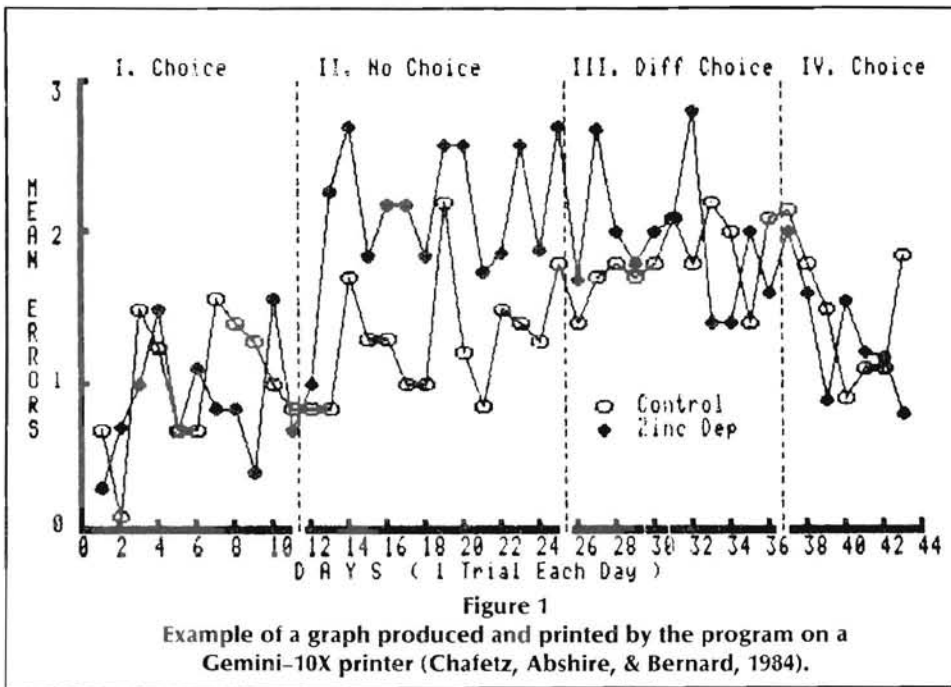


Figure 1
 Example of a graph produced and printed by the program on a Gemini-10X printer (Chafetz, Abshire, & Bernard, 1984).

(6)File

Any amount of drafting (finished or unfinished) may be saved in a file for future use. For example, it may be desirable to photograph a graph, save it in a file, and then come back later to print the graph (perhaps after additional labeling). A short menu is first provided: (1)Save (2)Restore (3)Files (4)Quit.

Selection of the first choice leads to a question:

Save a file? <n> ____

A no response will return you to the short menu. A yes (y) response will elicit another question:

File name: ____

The program then saves the screen image in the file whose name is supplied by the user. The file storage requires about 49K of disk space. On the conventional 5-1/4 inch DS/DD disk (320K) used in the Zenith, approximately 5-6 graphs can be stored comfortably. The number of files depends, of course, on the available disk space. The image is stored with the appropriate color values used in drafting.

Selection of the second option (Restore) allows you to place the image of a previously stored file back on the screen:

Restore a file? <n> ____

A no response returns the control back to the short menu. A yes response elicits another question:

Which file? ____

You must supply the name of a previously stored file. The image contained in the file will then appear on the screen. Caution: Any image presently on the graphing arena will be wiped out in favor of the newly restored image. This new image is now subject to any additional labeling. If additional data plots are required, you should keep in mind that the coordinate values are not stored along with the image. The axis values should be supplied by returning to (1)Axes in the main menu. Data may then be added to the restored image. Any newly restored image (just like any other image) can be printed using the (5)Photo/Print option from the main menu.

Selection of the third option (Files) provides a listing of files from either drive A or B. A question first appears to caution you that the file listing wipes out the current image and to ask whether you really wish to see the files. A no response returns control back to the short menu. A yes response elicits:

Drive A or B? ____

The single response denoting either drive returns the appropriate listing of files. You may now select the correct file to restore.

The (4)Quit option returns control to the main menu.

(7)Clear

This option wipes out the image on the graphing arena, sets all values to 0, and provides a clean graphing arena.

(8)End

This is used to end the program.

Current Limitations

The primary limitations reside in the hard copy printing. The printing is available only in black print. Although color may be desirable in some applications, most of the graphs printed in

journals are done in black ink, so this limitation is not particularly restrictive.

It takes over thirty minutes to print a graph. This is not a printer problem given that the Gemini 10X operates at 120 cps. It takes a considerable amount of time to read and interpret the screen, and this is the fastest algorithm out of three that have been tried. This is, however, a short amount of time when compared to actual drafting, and you should keep in mind that you are free to take a coffee break during printing.

There is one minor glitch in the printing. The Gemini 10X interprets the binary summed value of 9 to be a horizontal tab. In order to prevent unsightly tabbing, the bit image value of 9 is changed to a similar 8 (leaving out 1 bit). This occurs infrequently in an image, and it is not easy to detect the missing bit unless you are looking for it. You are invited to find the missing bits in the figure to determine whether the image has been compromised.

Although you are permitted flexibility in the initiation and extension of X and Y axes, the program provides for only positive-going axes with an origin in the lower left quadrant. You are free, however, to transpose the data by addition to the values so that the program recognizes the positive values. You can then choose to label the coordinates with negative values.

References

Chafetz, M.D., Abshire, F.M., & Bernard, D.L. Zinc deficiency in adult rats alters foraging in a radial arm maze. In C.J. Frederickson, G.A. Howell, & E.J. Kasarskis (Eds.) *The Neurobiology of Zinc, Part B: Deficiency, Toxicity, and Pathology*, New York: Alan Liss, Inc., 1984, pp. 109-120.

Listing 1 Program Listing for Scientific Graph Drafting

```

1 REM Scientific graphing program written and developed by
2 REM Michael D. Chafetz, Psychology, Tulane Univ., New
  Orleans, LA 70118
3 CLS:LINE(0,0)-(639,117),4,BF:GOSUB 70
5 ON ERROR GOTO 1000
10 CLS:LINE(0,0)-(639,208),.B:DG=0:EN=0:
  DIM SX(250),RX(250),RY(250):NC=250
15 S$="12345678":CS$="01234567":F=0:YV=0:CF=7:P=12:Q=40
  DIM SP(250):GOSUB 40
20 ON VAL(X$) COSUB 100,200,300,400,500,600,700,800
30 GOTO 20
40 GOSUB 999:LOCATE,,0:PRINT"Choose:
  (1)Axes (2)Label (3)Data (4)Color (5)Photo/Print (6)File
  (7)Clr (8)End":BEEP
50 X$=INPUT(1):IF INSTR(S$,X$)=0 THEN 50 ELSE RETURN 30
70 BEEP:LOCATE 5,27:PRINT"SCIENTIFIC GRAPH DRAFTING"
71 LOCATE 7,39:PRINT"by"
72 LOCATE 9,27:PRINT"Michael D. Chafetz, Ph.D.":
  FOR X=1 TO 1400:NEXT X
73 CLS:LINE(0,0)-(639,208),.B
74 LOCATE 3,7:PRINT"This program is designed to help you
  draft graphs for professional"
75 LOCATE 4,2:PRINT"
  presentation or manuscript submission to journals."
76 LOCATE 6,7:PRINT"You'll be drawing on a plotting arena --
  the frame bordering these words."
78 LOCATE 9,7:PRINT"All you need to do is to respond to a
  few questions from each menu"
79 LOCATE 10,2:PRINT"selection. Type Q at any time to quit.
80 LOCATE 14,7:PRINT"Ready to go when you are....
  Type any key to start...."
81 IF INKEY$="" THEN 81 ELSE RETURN
100 GOSUB 999:EN=1:INPUT"Highest X axis value? ",V$:
  GOSUB 160:XV=VAL(V$)

```

```

102 DEF FNY=XL*XV*PR*3.25/YV:DEF FNX=8*XL
103 GOSUB 999:EN=2:INPUT"Starting X value? ",V$:GOSUB 160:
  XS=VAL(V$):XV=XV-XS
105 GOSUB 999:EN=3:INPUT"Mark every ___ value(s)? ",V$:
  GOSUB 160:XM=VAL(V$)
107 IF XM=0 THEN XM=XV/5:BEEP:GOSUB 169
110 GOSUB 999:EN=4:INPUT"X length? ",V$:GOSUB 160:
  XL=VAL(V$):IF XL=13 THEN 110
115 IF XV*XL+6<=79 THEN LINE(50,186)-(50+XV*FNX,188),.BF
  ELSE GOTO 170
120 FOR N=0 TO XV STEP XM:
  LINE(50+N*FNX,188)-(52+N*FNX,183),.BF:NEXT N
125 GOSUB 999:EN=5:INPUT"Highest Y axis value? ",V$:
  GOSUB 160:YV=VAL(V$)
128 GOSUB 999:EN=6:INPUT"Starting Y value? ",V$:GOSUB 160
  YS=VAL(V$):YV=YV-YS
130 GOSUB 999:EN=7:INPUT"Mark every ___ value(s)? ",V$:
  GOSUB 160:YM=VAL(V$)
133 IF YM=0 THEN YM=YV/5:BEEP:GOSUB 169
135 GOSUB 999:EN=8:INPUT"Y/X ratio? ",V$:GOSUB 160
  PR=VAL(V$)
143 IF FNY*YV<179 THEN LINE(50,188)-(52,188-FNY*YV),.BF
  ELSE GOTO 170
145 FOR N=0 TO YV STEP YM:
  LINE(50,188-N*FNY)-(56,187-N*FNY),.BF:NEXT N
148 IF DG=1 THEN 220
150 DG=0:GOTO 40
160 V=ASC(V$):IF V=81 OR V=113 THEN 40 ELSE
  IF V<58 AND V>47 OR V=46 THEN RETURN ELSE BEEP:GOSUB 999
165 PRINT "Use numbers!":FOR X=1 TO 600:NEXT X
167 ON EN GOTO 100,103,105,110,125,128,130,135
169 GOSUB 999:PRINT"No 0's here! ...making 5 marks."
  FOR X=1 TO 600:NEXT:RETURN
170 BEEP:GOSUB 999:PRINT"Too long! Making shorter units. "
  FOR X=1 TO 600:NEXT
175 IF YV=0 THEN XL=69/XV:GOTO 115
180 PR=170/XL/XV/3.25:GOTO 143
200 GOSUB 999:LOCATE,,0:PRINT"Automatic numbering? <n>":
  A$=INPUT$(1)
205 IF A$="Q" OR A$="q" THEN 40 ELSE IF A$<>"Y" AND A$<>"y"
  THEN 220
208 IF XV=0 THEN GOSUB 999:PRINT"No X values! ":BEEP:
  FOR X=1 TO 600:NEXT X:GOTO 40
210 FOR N=0 TO XV STEP XM:LOCATE 22,(6+N*XL):PRINT N+XS:
  NEXT N
215 FOR N=0 TO YV STEP YM:LOCATE (192-N*FNY)/9,3:
  PRINT N+YS:NEXT N
218 IF LEN(STR$(YV+YS))>3 OR LEN(STR$(YM))>3 THEN DG=1
  GOTO 143
220 999:PRINT"Move:<<Up(4) Dn(1) Rt(3)
  Lf(2)>> hLabel(5) vLabel(6) Legend(7) Quit(8)"
225 LOCATE P,Q:C$=INPUT$(1):IF INSTR(S$,C$)=0 THEN 225
228 ON VAL(C$) GOTO 230,231,232,233,250,255,265,40
230 P=P+1:IF P>23 THEN ERROR 5 ELSE 225
231 Q=Q-1:GOTO 225
232 Q=Q+1:GOTO 225
233 P=P-1:GOTO 225
250 L$=INPUT$(1):IF ASC(L$)=13 THEN 220 ELSE PRINT L$:Q=Q+1
253 IF Q>79 THEN ERROR 5 ELSE LOCATE P,Q:GOTO 250
255 L$=INPUT$(1):IF ASC(L$)=13 THEN 220 ELSE PRINT L$:P=P+1
260 IF P>23 THEN ERROR 5 ELSE LOCATE P,Q:GOTO 255
265 GOSUB 999:LOCATE,,0
  PRINT"1=0-diam 2=F-diam 3=0-box 4=F box 5=points"
270 X=Q*8-4:Y=P*9-4:K$=INPUT$(1)
  ON VAL(K$) GOSUB 950,955,960,965,970:GOTO 220
300 GOSUB 999:LOCATE,,0:PRINT"X values:
  (1)Reg (2)Irreg (3)Qt":X$=INPUT$(1):C=0
301 S=0:IF INSTR("123",X$)=0 THEN 300
  ELSE ON VAL(X$) GOTO 304,302,40
302 GOSUB 999:EN=1:INPUT"
  Enter each X value and hit return (Q to qui ) ",V$
303 IF V$="Q" OR V$="q" THEN 309 ELSE GOSUB 380:XH=VAL(V$):
  IF XH<XS THEN 390 ELSE IF XH>(XV+XS) THEN 395
  ELSE SX(C)=51+FNX*(XH-XS):RX(C)=XH:C=C+1:NC=C:GOTO 302
304 GOSUB 999:EN=2:INPUT"Plot every ___ X unit(s) ",V$:
  COSUB 380:XP=VAL(V$)
307 GOSUB 999:EN=3:INPUT"...starting with: ",V$:GOSUB 380:
  S=VAL(V$)-XS:IF S<0 THEN 390 ELSE IF S>XV THEN 395

```



```

309 GOSUB 999:LOCATE ,0:PRINT"Plot
1=0-diam 2=F-diam 3=0-box 4=F-box 5=Points 6=HIST-0
7=HIST F":CH$=INPUT$(1):C=0:CH=VAL(CH$):
IF CH=0 THEN CH=1
310 GOSUB 999:EN=4:INPUT
"Enter each Y value and hit return (Q to quit) ",V$
315 IF V$="Q" OR V$="q" THEN 335 ELSE GOSUB 380:YP=VAL(V$):
IF YP<YS THEN 390
318 IF YP>(YV+YS) THEN 395
319 IF X$="1" THEN SX(C)=51+FNX*(S+C*XP):RX(C)=(C*XP+S)+XS
320 Y=188-FNY*(YP-YS):X=SX(C):SP(C)=Y:RY(C)=YP:C=C+1
325 ON CH GOSUB 950,955,960,965,970,975,980:IF C>=NC
THEN 335 ELSE IF (C*XP+S)<=XV THEN 310
335 BEEP:F=0:GOSUB 999:LOCATE ,0:PRINT
"Connect? (1)no (2)lines (3)dots (4)Qt":CP$=INPUT$(1)
IF INSTR("1234",CP$)=0 THEN 335 ELSE ON VAL(CP$)
GOTO 355, 338,348,40
338 FOR N=0 TO C-2:X1=SX(N):X2=SX(N+1):Y1=SP(N):Y2=SP(N+1)
343 LINE(X1,Y1)-(X2,Y2),CF:NEXT N:GOTO 355
348 FOR N=0 TO C-2:X1=SX(N):X2=SX(N+1):Y1=SP(N):Y2=SP(N+1):
B=(Y2 Y1)/(X2 X1)
349 Z=SQR(8.09*(Y2-Y1)^2+(X2-X1)^2:IF X1>X2 THEN S=AP X1,X2,
SWAP Y1,Y2
350 FOR T=X1 TO X2 STEP (X2 X1)/(Z/9):
PSET(T,Y1-B*X1+B*T),CF:NEXT T:NEXT N
355 BEEP:GOSUB 999:LOCATE ,0:PRINT"Error bars? <n>":
EB$=INPUT$(1)
360 IF EB$="Q" OR EB$="q" THEN 40 ELSE
IF EB$<>"Y" AND EB$<>"y" THEN 370 ELSE IF C=0 THEN 40
363 FOR N=0 TO C-1:X=SX(N):Y=SP(N)
365 GOSUB 999:EN=5:INPUT"+/- (Each point) ",PM
368 LINE(X,Y)-(X,Y+PM*FNY),CF:LINE -(X,Y-PM*FNY),CF:NEXT N
370 BEEP:GOSUB 999:LOCATE ,0:PRINT"Regression line? <n>":
RC$=INPUT$(1)
371 MX=0:MY=0:DX=0:DY=0:RZ=0:IF C=0 THEN 40
372 SC=640/225:IF RG$<>"Y" AND RG$<>"y" THEN 40
374 FOR N=0 TO C-1:MX=MX+RX(N):MY=MY+RY(N):NEXT N:MX=MX/C:
MY=MY/C
375 FOR N=0 TO C-1:DX=DX+(RX(N)-MX)^2:DY=DY+(RY(N)-MY)^2:
NEXT N:DX=SQR(DX/C):DY=SQR(DY/C)
376 FOR N=0 TO C-1:RZ=RZ+((RX(N)-MX)/DX)*((RY(N)-MY)/DY):
NEXT N:RZ=RZ/C
377 RB=RZ*DY/DX:RA=MY-RB*MX:XF=51+XV*FNX:RI=188 FNY*RA.
YI=188-FNY*(RB*XS+RA-YS):YF=188-FNY*(RB*(XV+XS)+RA-YS)
378 LINE(51,YI)-(XF,YF),CF:GOSUB 999.
PRINT"r = ",RZ,"Intercept = ",RA,"Slope = ",RB
379 X$=INKEY$:IF X$="" THEN 379 ELSE 40
380 V=ASC(V$):IF V=81 OR V=113 THEN 40 ELSE
IF V<58 AND V>47 OR V=46 THEN RETURN ELSE BEEP:GOSUB 999
385 PRINT"Use numbers:FOR X=1 TO 600:NEXT
387 ON EN GOTO 302,304,307,310,365
390 BEEP:GOSUB 999:PRINT"Value too low!":FOR X=1 TO 600
NEXT:GOTO 387
395 BEEP:GOSUB 999:PRINT"Value too high!":FOR X=1 TO 600:
NEXT:GOTO 387
400 GOSUB 999:LOCATE ,0:PRINT"Change color? <n>":
C$=INPUT$(1)
405 IF C$<>"Y" AND C$<>"y" THEN 40
410 GOSUB 999:PRINT"Pick color (7 to keep white): ":
GOSUB 435:CF$=INPUT$(1)
415 IF INSTR(CS$,CF$)=0 THEN 410 ELSE CF=VAL(CF$)
430 GOSUB 999:PRINT"Color changed!":BEEP:FOR C=1 TO 600:
NEXT C:GOTO 40
435 LOCATE 25,32,0:PRINT
"0=Blk 1=Blu 2=Crn 3=Cy 4=Rd 5=Mag 6=Ylw 7=Wht":RETURN
500 GOSUB 999:LOCATE ,0:PRINT"Eraser frame? <n>":E$=INPUT$(1)
503 IF E$="Q" OR E$="q" THEN 40
505 IF E$="Y" OR E$="y" THEN LINE(0,0)-(639,208),0,B
510 GOSUB 999:LOCATE ,0:PRINT"New frame? <n>":F$=INPUT$(1)
513 IF F$="Q" OR F$="q" THEN 40
515 IF F$="Y" OR F$="y" THEN LINE(0,0)-(639,208),CF,B
520 GOSUB 999:BEEP:LOCATE ,0:PRINT"Photo session.":
FOR N=1 TO 600:NEXT N
525 GOSUB 999:X$=INKEY$:IF X$="" THEN 525
530 GOSUB 999:LOCATE ,0:PRINT"Print screen? <n>":
G$=INPUT$(1)
532 IF G$<>"Y" AND G$<>"y" THEN 40
533 GOSUB 999:LOCATE 25,1:PRINT

```

```

"Take a break. Printing takes time!"
535 DIM P(639):D=2
540 LPRINT CHR$(27) CHR$(64)
542 LPRINT CHR$(27) CHR$(49)
545 FOR X=1 TO 637:V=0:FOR Y=0 TO 7.
V=V+1*(0<>POINT(X,Y+D))*2*(7-Y):NEXT Y
550 IF V=9 THEN LET V=8
553 P(X)=V:NEXT X
555 LPRINT CHR$(27) CHR$(76) CHR$(125) CHR$(2),
560 FOR N=1 TO 637:LPRINT CHR$(P(N)),NEXT N
565 LPRINT CHR$(13)
570 D=D+7:IF D=212 THEN 40 ELSE 545
600 GOSUB 999:LOCATE ,0:
PRINT"(1)Save (2)Restore (3)Files (4)Quit"
605 X$=INPUT$(1):IF INSTR("1234",X$)=0 THEN 605
610 ON VAL(X$) GOTO 615,635,660,40
615 GOSUB 999:LOCATE ,0:PRINT"Save a file? <n>":X$=INPUT$(1)
620 IF X$<>"Y" AND X$<>"y" THEN 600
625 GOSUB 999:INPUT"Which file? ",FI$
630 GOSUB 999:DEF SEG=&HE000:BSAVE FI$,0,&HB999:GOTO 600
635 GOSUB 999:LOCATE ,0:PRINT"Restore a file? <n>":
X$=INPUT$(1)
640 IF X$<>"Y" AND X$<>"y" THEN 600
645 GOSUB 999:INPUT"Which file? ",FI$
650 DEF SEG=&HE000:BLOAD FI$,0:DEF SEG=&HD000:BLOAD FI$,0
655 DEF SEG=&HC000:BLOAD FI$,0:GOTO 600
660 GOSUB 999:LOCATE ,0:
PRINT"Files overprint screen - you sure? <n>":
665 X$=INPUT$(1):IF X$<>"Y" AND X$<>"y" THEN 600
670 GOSUB 999:PRINT"Drive A or B? ":LOCATE 25,15.
D$=INPUT$(1):IF INSTR("ABab",D$)=0 THEN 670
675 FOR X=2 TO 23:LOCATE X,1:PRINT SPC(78):NEXT:LOCATE 2,1:
FILES D$+":* *":GOTO 600
700 X$="":CLEAR:CLS:GOTO 5
800 LOCATE ,1:CLS:END
950 CIRCLE(X,Y),4,CF:RETURN
955 CIRCLE(X,Y),4,CF:PAINT(X+2,Y),CF,CF:RETURN
960 LINE(X 4,Y-2) (X+4,Y+2),CF,B:RETURN
965 LINE(X-4,Y-2)-(X+4,Y+2),CF,BF:RETURN
970 PSET(X,Y),CF:RETURN
975 LINE(X-10,187)-(X+10,Y),CF,B:RETURN
980 LINE(X-10,187) (X+10,Y),CF,BF:RETURN
999 LOCATE 25,1:PRINT SPC(79):LOCATE 25,1,1:RETURN
1000 IF ERL=160 OR ERL=380 THEN BEEP:GOSUB 999.
PRINT"Give a value!":FOR X=1 TO 800:NEXT:
IF ERL=160 THEN RESUME 167 ELSE RESUME 387
1001 IF ERL>600 THEN BEEP:GOSUB 999:PRINT"Give a filename!":
FOR X=1 TO 800:NEXT:
IF ERL<635 THEN RESUME 615 ELSE RESUME 635
1003 GOSUB 999:PRINT"Off the screen - moving you back one ":
BEEP:FOR N=1 TO 600:NEXT N
1005 IF P>23 THEN P=23 ELSE IF Q<1THEN Q=1 ELSE IF Q>80 ELSE
IF P<1 THEN P=1
1010 RESUME 220

```

About The Author

Michael D. Chafetz is a Statistical Consultant in New Orleans, Louisiana.



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	Issue	Part Number	Description of Product	Selling Price	Issue	Part Number	Description of Product	Selling Price	Issue				
HDOS HARDCOPY SOFTWARE															
885-1008	Volume I Documentation	9.00		885-1083-[37]	Disk XVI Misc H8/H89	20.00	11	885-3029-37§§	ZDOS/MSDOS HUG Bg. Print Spool	20.00	66				
885-1013	Volume II Documentation	12.00		885-1089-[37]	Disk XVII Misc H8/H89	20.00	20	885-3032-37§	MSDOS Halley's Comet Locator	20.00	70				
885-1015	Volume III Documentation	9.00		885-1090-[37]	Disk XIX Utilities H8/H89	20.00	22	885-3035-37§§	MSDOS SPELL5 & SPELL5F	20.00	72				
885-1037	Volume IV Documentation	12.00	8	885-1092-[37]	Relocating Debug Tool H8/H89	30.00	14	885-8039-37§§	MSDOS DPATH	20.00	74				
885-1058	Volume V Documentation	12.00		885-1098	H8 Color Graphics ASCII	20.00	19	885-8040-37§§	MSDOS HELP Programs	20.00	74				
MISCELLANEOUS HDOS COLLECTIONS															
885-1032	Disk V H8/H89	18.00	8	885-1099	H8 Color Graphics Tiny PASCAL	20.00	19	§ All program files run on both §§ Program files run partially on both							
885-1044-[37]	Disk VI H8/H89	18.00		885-1105	HDOS Device Drivers H8/H89	20.00	24	PC/IBM COMPATIBLE							
885-1064-[37]	Disk IX H8/H89 Disk	18.00		885-1116	HDOS Z80 Debugging Tool	20.00	27	885-6001-37	MSDOS Keymapper	20.00	59				
885-1066-[37]	Disk X H8/H89	18.00	10	885-1119-[37]	BHBASIC Support	20.00	29	885-6002-37	CP/EMulator II & ZEMulator	20.00	59				
885-1069	Disk XIII Misc H8/H89	18.00		885-1120-[37]	HDOS 'WHEW' Utilities	20.00	33	885-6003-37	MSDOS EZPLOT	20.00	65				
GAMES															
885-1010	Adventure Disk H8/H89	10.00	4	885-1121	HDOS Hard Sec Sup Pkg 2 Disks	30.00	37	885-6004-37	MSDOS CheapCalc	20.00	67				
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8	885-1123	XMET Robot Cross Assembler	20.00	40	885-6005-37	MSDOS Skyviews	20.00	67				
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8	885-1126	HDOS Utilities by PS:	20.00	42	885-6006-37	MSDOS Cardcat	20.00	69				
885-1031	Disk IV MUSIC H8 Only	20.00	25	885-1127-[37]	HDOS Soft Sector Support Pkg	30.00	45	885-6007-37	MSDOS DND (Dung. & Dragons)	20.00	70				
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12	885-1129-[37]	HDOS DISKVIEW	16.00	46	885-8033-37	MSDOS Fast Edit	20.00	62				
885-1068	Disk XII MBASIC Graphic Games	18.00	10	885-1129-[37]	HDOS CVT Color Video Terminal	20.00	46	885-8037-37	MSDOS Grade	20.00	70				
885-1089-[37]	Disk XVII MBASIC Graph Games	20.00	14	885-8001	St (Screen Editor)	25.00	28	PROGRAMMING LANGUAGES							
885-1093-[37]	D&D H8/H89 Disk	20.00	16	885-8003	BHTOMB	25.00	28	HDOS							
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18	885-8004	UDU:IP	35.00	28	885-1038-[37]	Wise on Disk H8/H89	18.00					
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20	885-8006	HDOS SUBMIT	20.00	31	885-1042-[37]	PILDT on Disk H8/H89	19.00					
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23	885-8007	EZITRANS	30.00	30	885-1059	FOCAL-8 H8/H89 Disk	25.00	13				
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23	885-8015	HDOS TEXTSET Formatter	30.00	42	885-1078-[37]	HDOS Z80 Assembler	25.00	21				
885-1113-[37]	HDOS Action Games H8/H89	20.00	23	885-8017	HDOS Programmers Helper	16.00	42	885-1085	PILDT Documentation	9.00					
885-1114	H8 Color Raiders & Goop	20.00	23	885-8024	HDOS BHBASIC Utilities Disk	16.00	46	885-1086-[37]	Tiny HDOS PASCAL H8/H89	20.00	13				
885-1124	HUGMAN & Movie Animation Pkg	20.00	41	CP/M											
885-1125	MAZEMADNESS	20.00	41	885-1210-[37]	CP/M: ED (same as 885-1022)	20.00	20	885-1094	HDOS Fig-Forth H8/H89	40.00	18				
885-1130	Star Battle	20.00	45	885-1212-[37]	CP/M Utilities H8/H89	20.00	21	885-1132-[37]	HDOS Tiny BASIC Compiler	25.00	59				
885-1133-[37]	HDOS Games Collection I	20.00	59	885-1213-[37]	CP/M Disk Utilities H8/H89	20.00	22	885-1134	HDOS SMALL-C Compiler	30.00	63				
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32	885-1217-[37]	HUG Disk Duplication Utilities	20.00	26	CP/M							
885-8022	HDOS SHAPES	16.00	45	885-1223-[37]	HRUN HDOS Emulator 3 Disks	40.00	37	885-1208-[37]	CP/M Fig-Forth H8/H89 2 Disks	40.00	18				
885-8026	HDOS Space Drop	16.00	49	885-1225-[37]	CP/M Disk Dump & Edit Utility	30.00	40	885-1215-[37]	CP/M BASIC-E	20.00	26				
885-8032-[37]	HDOS Castle	20.00	59	885-1226-[37]	CP/M Utilities by PS:	20.00	40	BUSINESS, FINANCE AND EDUCATION							
CP/M															
885-1206-[37]	CP/M Games Disk	20.00	11	885-1229-[37]	XMET Robot Cross Assembler	20.00	40	HDOS							
885-1209-[37]	CP/M MBASIC D&D	20.00	19	885-1230-[37]	CP/M Function Key Mapper	20.00	42	885-1047	Stocks H8/H89 Disk	18.00					
885-1211-[37]	CP/M Sea Battle	20.00	20	885-1231-[37]	Cross Ref Utilities for MBASIC	20.00	43	885-1048	Personal Account H8/H89 Disk	18.00					
885-1220-[37]	CP/M Action Games	20.00	32	885-1232-[37]	CP/M Color Video Terminal	20.00	46	885-1049	Income Tax Records H8/H89 Disk	18.00					
885-1222-[37]	CP/M Adventure	10.00	35	885-1235-37	CP/M COPYDOS	20.00	54	885-1055-[37]	MBASIC Inventory Disk H8/H89	30.00					
885-1227-[37]	CP/M Casino Games	20.00	38	885-1237-[37]	CP/M Utilities	20.00	50	885-1056	MBASIC Mail List	30.00					
885-1228-[37]	CP/M Fast Action Games	20.00	39	885-1245-37	CP/M-85 KEYMAP	20.00	63	885-1070	Disk XIV Home Fin H8/H89	18.00					
885-1236-[37]	CP/M Fun Disk I	20.00	55	885-1246-[37]	CP/M HUG File Manager & Utilities	20.00	64	885-1071-[37]	MBASIC SmbusPk H8/H19/H89	75.00	17				
885-1248-[37]	CP/M Fun Disk II	35.00	69	885-1247-37	CP/M-85 HUG Bkgrd Print Spooler	20.00	67	885-1091-[37]	Grade/Score Keeping H8/H89	30.00	14				
ZDOS															
885-3004-37	ZDOS ZBASIC Graphic Games	20.00	37	885-5001-37	CP/M-86 KEYMAP	20.00	51	885-1097-[37]	MBASIC Quiz Disk H8/H89	20.00	18				
885-3009-37	ZDOS ZBASIC D&D	20.00	50	885-5002-37	CP/M-86 HUG Editor	20.00	52	885-1118-[37]	MBASIC Payroll	60.00	30				
885-3011-37	ZDOS ZBASIC Games Disk	20.00	52	885-5003-37	CP/M-86 Utilities by PS:	20.00	54	885-1131-[37]	HDOS CheapCalc	20.00	47				
885-3017-37	ZDOS Contest Games Disk	25.00	58	885-5008-37	CP/M 8080 To 8088 Trans. & HFI:	20.00	64	885-8010	HDOS Checkoff	25.00	32				
UTILITIES															
885-1022-[37]	HUG Editor (ED) Disk H8/H89	20.00	20	885-5009-37	CP/M: 86 HUG Bkgrd Print Spool	20.00	66	885-8021	HDOS Student's Statistics Pkg	20.00	44				
885-1025	Runoff Disk H8/H89	35.00		885-8018-[37]	CP/M: Fast Eddy & Big Eddy	20.00	43	885-8027	HDOS SciCalc	20.00	50				
885-1060-[37]	Disk VII H8/H89	18.00		885-8019-[37]	DOCUMAT and DOCULIST	20.00	43	CP/M							
885-1061	TMI Load H8 ONLY Disk	18.00		885-8025-37	CP/M-85/86 Fast Eddy	20.00	49	885-1218-[37]	CP/M MBASIC Payroll	60.00	31				
885-1062-[37]	Disk VIII H8/H89 (2 Disks)	25.00		H/Z100 ZDOS/MSDOS - H/Z150 PC MSDOS											
885-1063	Floating Point Disk H8/H89	18.00		885-3005-37	ZDOS Etchdump	20.00	39	885-1233-[37]	CP/M CheapCalc	20.00	47				
885-1065	Fix Point Package H8/H89 Disk	18.00	10	885-3007-37	ZDOS CP/EMulator	20.00	47	885-1239-[37]	Spread Sht. Contest Disk I	20.00					
885-1070	HDOS Support Package H8/H89	60.00		885-3008-37	ZDOS Utilities	20.00	47	885-1240-[37]	Spread Sht. Contest Disk II	20.00					
885-1077	TXTCON/BASCON H8/H89	18.00		885-3010-37	ZDOS Keymap	20.00	51	885-1241-[37]	Spread Sht. Contest Disk III	20.00					
885-1079-[37]	HDOS Page Editor	25.00	15	885-3022-37	ZDOS/MSDOS Useful Programs I	30.00	63	885-1242-[37]	Spread Sht. Contest Disk IV	20.00					
885-1080	EDITX H8/H19/H89 Disk	20.00		885-3023-37	ZDOS/MSDOS EZPLOT	20.00	63	885-1243-[37]	Spread Sht. Contest Disk V	20.00					
885-1082	Programs for Printers H8/H89	20.00		885-3026-37	MSDOS SMALL C Compiler	25.00	65	885-1244-[37]	Spread Sht. Contest Disk VI	20.00					
CP/M															
885-1083	Disk XVI Misc H8/H89	20.00	11	885-3030-37	ZDOS/MSDOS Z-100 PC Emulator	40.00	68	885-8011-[37]	CP/M Checkoff	25.00	32				
885-1089	Disk XVII Misc H8/H89	20.00	20	885-3031-37	ZDOS/MSDOS Graphics	20.00	69	885-8036-[37]	CP/M Grade	20.00	70				
885-1092	Relocating Debug Tool H8/H89	30.00	14	885-3034-37	ZDOS/MSDOS ZPC Support Pkg	10.00	72	ZDOS							
885-1098	H8 Color Graphics ASCII	20.00	19	885-8029-37	ZDOS Fast Eddy	20.00	53	885-3006-37	ZDOS CheapCalc	20.00	47				
885-1099	H8 Color Graphics Tiny PASCAL	20.00	19	885-8035-37	MSDOS DOCUMAT and DOCULIST	20.00	70	885-3013-37	ZDOS Checkbook Manager	20.00	54				
885-1105	HDOS Device Drivers H8/H89	20.00	24	Continued on Page 83											

HUG NEW PRODUCTS

printer will print the data tables, but an MPI 99 or compatible printer is required to use the included printscreen function.

The following programs are included on the HUG P/N 885-8041-37 ORBITS disks:

Disk A

ORBITS	.DOC	.ORMEC	EXE
ORSYS	.EXE	SETUP	.EXE
SETUP	.FIL	ORBEND.	.COM
README	.DOC		

Disk B

AMOR	.ORB	APOLLO	.ORB
DEMO10	.OSY	DEMO3	.OSY
DEMO4	.OSY	DEMO5	.OSY
DEMO6	.OSY	EARTH	.ORB
FAYE	.ORB	HALLEY	.ORB
HIDALGO	.ORB	JUPITER	.ORB
MARS	.ORB	MERCURY	.ORB
NEPTUNE	.ORB	ORBEND	.ASM
ORMEC	.BAS	ORSYS	.BAS
PLUTO	.ORB	SATURN	.ORB
SETLIP	.BAS	URANUS	.ORB
VENUS	.ORB		

Program Author: Larry MacNeil

Program Content: ORMEC was written to help people visualize the mathematical concepts presented in Kepler's Laws. Given any 2 out of 7 elliptical parameters that describe an orbital path (semi-major axis, semi-minor axis, linear eccentricity, semi-latus rectum, eccentricity, radius at perigee, radius at apogee), the program solves for the other 5 parameters. If the central mass is also given, the program will solve for the velocities of a satellite at apogee and perigee, and for the period of the orbit. Then, colorful graphic displays, in high resolution, 640 X 480 interlace mode



HUG P/N 885-8041-37

ORBITS \$25.00

Introduction: This software package, called "Orbits", is a set of simulation programs which show what possible orbital paths look like and how satellites move while orbiting in these paths. Each colorful ellipse pattern that is created, with its moving satellites, has a beauty all its own. Now we can see the harmony of the spheres! This package was written (over a years period of time) to help students visualize the basic mathematical concepts involved in orbital mechanics.

Requirements: This package will run on an H/Z-100 with ZDOS/MSDOS version 1.25 or higher. The hardware also requires 192k of memory, as well as 64k color RAM chips installed on the video board. The monitor can be monochrome, but color displays are much easier to interpret. More than one disk drive will be useful for data storage, but is not required. Printing is optional; any

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.

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 assemblies (using MAKEBIOS) in less than 1 1/2 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-880H \$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

ZCPR3

We are licensed by ECHELON, Inc. to distribute the Z-System for the Heath market. This includes the ZCPR3 system with supporting software utilities, the ZRDOS-plus BDOS replacement and applicable documentation. Comes already installed and includes a bootable disk ready to run.

ZCPR Gives you many features found in UNIX and MSDOS. The ZRDOS is not in the public domain and gives you automatic logon of disks. No more "BDOS ERROR ON B:" messages just because you forgot to do a ^ C after changing disks.

Available for Heath BIOS 2.2.03 and .04 and for CDR BIOS. Check for availability of other BIOS configurations. Call or write for a brochure on this excellent product.

ZCPR3 - specify format and hardware \$98

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 OX-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

CF

A Configuration Utility For The H/Z-100

Jeff Kalis

1920 Sylvan, SE
Grand Rapids, MI 49506

Where I work, we have about a dozen Z-100 computers doing all sorts of things. Most of all, they are used by engineers in the development of programs to control our processor-based products. These programs are assembled using cross assemblers and burned into PROMs or used to have a dedicated processor made. Needless to say, these Z-100's get connected and disconnected from different equipment all day long. The only problem with this is much of the equipment does not use the same I/O configuration. This means the CONFIGUR.COM program needs to be run for each different piece of equipment. No real serious problem, but it's time consuming, and you can't make the changes in a batch file. It is also inconvenient not to be able to see what the current configuration is. Hence the need for CF.

CF has four modes of operation. The first mode is just a short help message and is invoked by entering 'CF ?' at the command prompt. A message will come up explaining the four modes.

The second mode will allow you to see what the current configuration is. Entering 'CF' by itself at the command prompt will result in the configuration of both the PRN and AUX devices being displayed on the CRT. If the device is configured parallel, this is reported, as well as which connector it should be hooked to. When the device is serially configured, the baud rate, word length, parity, number of stop bits, and hand shaking protocol, as well as the port and connector is displayed.

Entering 'CF S <d:filename.ext>' will save the current PRN and AUX device configuration on drive d: in the file filename.ext. If filename.ext exists, it will be replaced. If different configurations are to be saved on disk, use the CONFIGUR.COM program to set up the desired parameters, then save them to different files on disk. This way many different configurations can be saved.

To reconfigure the computer to one of the saved configurations, just enter 'CF <d:filename.ext>' where filename.ext is the name of a file saved with the S option discussed above, and d: is the drive where this file is located. This is the fourth and last mode of operation for CF.

CF will work for ZDOS version 1.x and MS-DOS version 2.x on any H/Z-100 computer. The changes made by CF are only made to the computers memory, and the next time the system is booted up, these changes are reset. Because CF calls a couple BIOS routines directly, this program will not run on any other machine. Such is the nature of direct BIOS calls. Speaking of BIOS calls, anyone who picks up the MS-DOS version 2 PROGRAMMER'S UTILITY PACK from ZDS should take note. The BIOS entry points as listed on page 8.20 in table 8.3 are wrong. If you need to use a direct BIOS call, be sure to get the information from the BIOS SOURCES disk that comes with the PACK, instead of the manual.

Let me take a moment here to talk about BSE100.EXE. This is the Basic Screen Editor that comes in the UTILITY PACK. This article was written with BSE100 and I rather like it. It's not a WordStar, but it sure beats the pants off EDLIN. I like it because it's pretty fast and only takes up about 26K on the disk, unlike WordStar which needs about 100K. At the time I wrote this, the PACK was selling for \$74.00 and I would recommend you buy one. There are many more utilities that come with the package, as well as all the information you would probably need about MS-DOS version 2.

The listing for CF that follows contains a liberal amount of comments, so following the code should not be too difficult. When typing in the source code, feel free to eliminate any or all comments. This will cut down on the number of keystrokes needed to enter the code.

Type in the source code using EDLIN or some other text editor and save it under CF.ASM. Assemble it using MASM. Fix any typos so MASM reports no errors or warnings when it finishes. Next, link the program with LINK. Ignore the stack segment error and convert the CF.EXE file to a .COM file using EXE2BIN. This can be done in one step by entering 'EXE2BIN CF.COM'.

Those of you who do not wish to dive into the inner workings of CF can skip ahead, because next is a discussion of how CF works.

And, for those who don't want to wear their fingers to the bone, send me a blank disk along with a buck and I'll return your disk with a copy of CF and a copy of the source code. Sending me just \$6.00 will get you a new DSDD disk with the files on it.

Let's discuss the configuration process first. I would like to direct you to the ZDOS manual page 6.22 for a general description on configuration and the need for logical devices. What CF does is take the group of information in the configuration tables and either save it, display it or change it. The operating system uses this information to develop the correct signals and protocol to talk to whatever is hooked up to the I/O ports. These tables are stored somewhere in IO.SYS, however, there happens to be a pointer in the BIOS entry area that tells us where these tables are. The first EQUate in the program is the offset into the BIOS segment where this pointer is located. The next two EQUates are the offsets for the PRN and AUX configuration routine entry points. A few more EQUates follow defining the function calls used along with a couple of ASCII characters.

The segment registers are set up for a COM file and the FCB's and program start location are ORGed. The first instruction CF will execute will be a jump to skip over our data. Next, we save storage for the table pointers, then store the configuration routine addresses to be used later in an indirect subroutine call. CF's messages come next. From the label PT1 to the label RUN are text strings that will be used to display the configuration information. These strings must have the exact number of spaces in them or you'll get some strange messages when the program is run. This is because the data in the tables are coded and these codes are used to create offsets into the text strings. If the text lengths are not right, the message will not start printing in the right spot.

The next thing CF will do is get the location of the configuration vector table. This address is loaded into the SI register. The first pointer in the table is for the disk vector. We want the second and the third pointers for the PRN and AUX tables, respectively. These addresses are stored in the area we reserved earlier.

The input command that invoked CF is copied to location 80 hex by ZDOS, so now we can check that input and determine what function to perform. If nothing other than 'CF' was entered, the program jumps to display the current configuration. If a '?' was entered, the help message will be displayed. When both of these checks fail, CF looks for the first non-space character. If it is an 'S', then CF looks for a space to follow it. If a space is there, the program goes to the save portion of the program. It goes to the load portion if any other input was entered.

The save function will save the current configuration in a disk file. The filename of this file will be in the secondary FCB at 6C hex, because the primary FCB at 5C hex contains the 'S'. CF moves the filename into the primary FCB while at the same time it checks for wildcard specifiers. CF will not let you name an output file that has a wildcard in it. Next, this file is created, the information retrieved and written to disk. The file is then closed and the program terminated. The subroutine to retrieve the configuration data simply moves this data to the Data Transfer Area (DTA) at location 80 hex. The record size is set to 70 bytes, because that is the length of the two combined data tables.

The load information function is just the reverse of the save function. The filename is in the primary FCB when CF is invoked, so this file is opened and the first 70 bytes are copied to the DTA. This data is then moved to the configuration tables. This does not fully accomplish the task of changing the configuration. Two BIOS routines have to be called upon to finish the job. The calls are made to the indirect locations that were set up earlier with the locations of the BIOS entry points. One call for each device

and that takes care of that. The BIOS calls are described on page 1.38 in the ZDOS manual.

Last, but not least, is the display function. The data is moved into the DTA and then it is analyzed, decoded and printed on the CRT. Pages 1.39 through 1.41 in the ZDOS manual lists the definitions for the PRN and AUX devices. If the device is configured for parallel operation, then only the port address and plug connection need to be reported. On the other hand, if it's serial, there is a whole bunch of stuff to display.

If the first byte in the table is other than a 1 or a 2, a message will be displayed stating the device has an unknown configuration. CF only understands a 1 or a 2 as the first byte in the configuration table.

The way CF decodes most of the configuration data is to get the code for a particular parameter and generate an offset into the list of strings for that parameter. Take for instance, a baud rate code of 6 (see page 1.40 in the ZDOS manual). This represents a baud rate of 300. CF will take the code, in this case 6, and multiply it by the length of the baud rate strings stored in the message area. The baud rate strings are also 6 characters long. After the multiplication, the offset is 36 characters from the start of the baud rate string storage. CF will point to this location and then call the print string routine function call, hence the reason for the '\$' character at the end of each string. Remember the '\$' itself must be counted as part of the string when figuring the offset, because it does take up a byte in memory.

The port where the device is connected could be any port in the system, so a provision is made to display the port number if it is not one of the three standard port addresses. The plug number on the back of the H/Z-100 is also displayed if the port is one of the three standard ports.

CF has been well tested and I don't believe there are any bugs in it, but if you find one or have any other problems with CF, feel free to contact me and I'll try to help you out. Send me a note in the mail or leave me a message through COMPUSERVE or the HUG SIG. I don't get on there as often as I would like to, but eventually I'll see your message. My COMPUSERVE ID is 75216,2023.

About The Author

Jeff Kalis is a Quality Control Engineer for Rowe International Inc. in Grand Rapids, Michigan. Through his job he has become well versed in assembly language programming using several different processors including the F8/3870, 6502, 8748, 8085, and the 8088. He has a Z-100 at home, as well as one at work. His interests include computers, tropical fish, baseball and most of all his wife and two kids.

Listing 1

```
PAGE      60,132
;
; 'CF' - CONFIGURATION UTILITY
;      Version 2.0
;
;      Written by Jeff Kalis
;      Nov. 1984
;
; CF will report the current system configuration for
; the PRN and AUX devices by entering 'CF'
; When 'CF S filename.ext' is entered, the current
; configuration is saved on disk in the named file
; If 'CF filename.ext' is entered, the system configur-
; uration will be changed to the data from the named file
```

```

, 'CF ?' will cause a short help message to be displayed.
;
; When the need to change configuration is frequent, the
; CONFIGUR utility is bothersome and cannot be included
; in a batch file CF solves these drawbacks by saving
; the configuration tables for the PRN and AUX devices
; in a file on the disk so configuration is easily changed
; CONFIGUR must be used to initially set the system then
; use CF to save it Any number of different configura-
; tions can be saved.
;

```

*** PROGRAM EQUATES ***

```

CTADDR EQU 0061H ; BIOS location with pointer to
                vector tables
PRNFUNC EQU 004BH ; BIOS entry point to configure
                the PRN device
AUXFUNC EQU 004EH ; BIOS entry point to configure
                the AUX device
BASE_SEG EQU 0040H ; BIOS base segment address
REC_LEN EQU 70 ; Length of configuration tables
                for PRN and AUX
DTA EQU 0080H ; Default data transfer address

LF EQU 10 ; ASCII line feed
CR EQU 13 ; ASCII return
PSTRNG EQU 9 ; function call to print a string
OPEN EQU 0FH ; function call to open a file
CLOSE EQU 10H ; function call to close a file
READ EQU 14H ; function call to read a sequen-
                tial file record
WRITE EQU 15H ; function call to write a sequen-
                tial file record
CFILE EQU 16H ; function call to create a file
ESC EQU 1BH ; ASCII escape
B2DOS EQU 20H ; system interrupt to return to dos
SYS_CALL EQU 21H ; system interrupt to execute a
                function call

```

*** SET UP FOR COM FILE ***

```

CODE SEGMENT
ASSUME CS:CODE,DS:CODE,ES:NOTHING

ORC 005CH
FCB1 DB 0 ; Location of primary FCB
ORC 006CH
FCB2 DB 0 ; Location of secondary FCB
ORC 0100H ; Program start location

START JMP RUN ; Jump over the data area

```

*** DATA STORAGE ***

```

PRN_VEC DW 0 ; storage for address
                of PRN table
AUX_VEC DW 0 ; storage for address
                of AUX table
PRNF DW PRNFUNC,BASE_SEG; indirect call to PRN
                configure routine
AUXF DW AUXFUNC,BASE_SEG; indirect call to AUX
                configure routine

```

*** MESSAGE STORAGE ***

```

ERR1 DB CR,LF,'Illegal filename',CR,LF,'$'
ERR2 DB CR,LF,'Unable to open input file',CR,LF,'$'
ERR3 DB CR,LF,'Error writing file to disk',CR,LF,'$'
ERR4 DB CR,LF,'Unable to open output file',CR,LF,'$'
OK1 DB CR,LF,'Function complete'
CRLF DB CR,LF,'$'

```

*** CONFIGURATION MESSAGE STORAGE ***

```

PT1 DB CR,LF,ESC,'p PRN: ',ESC,'q configuration '$'
PT2 DB CR,LF,LF,ESC,'p AUX: ',ESC,'q configuration '$'

```

```

UNKL DB 'has an unknown device assignment',CR,LF,'$'
DEVI DB 'is '$'
PARI DB ESC,'p PARALLEL ',ESC,'q on Port '$'
SER1 DB ESC,'p SERIAL ',ESC,'q',CR,LF,LF
DB ' WORD STOP',CR,LF
DB 'BAUD LENGTH BITS PARITY PORT'
BAUD DB ' ASSIGNMENT HANDSHAKE PROTOCOL',CR,LF,'$'
DB '45.5 '$' ; Baud rate strings ( 6 bytes ea)
DB '50 '$'
DB '75 '$'
DB '110 '$'
DB '134.5'$'
DB '150 '$'
DB '300 '$'
DB '600 '$'
DB '1200 '$'
DB '1800 '$'
DB '2000 '$'
DB '2400 '$'
DB '4800 '$'
DB '9600 '$'
DB '19200'$'
DB '38400'$'
WRDLEN DB ' 5 '$' ; word length strings
                (14 bytes ea)
DB ' 6 '$'
DB ' 7 '$'
DB ' 8 '$'
SBITS DB ' 1 '$' ; stop bit length
                strings (8 bytes ea)
DB '1.5 '$'
DB ' 2 '$'
ODD DB ' odd '$' ; parity strings
                (13 bytes ea)
EVN DB ' even '$'
DISA DB 'disabled '$'
PJ1 DB 'A (J1) (E8H) '$' ; port strings
                (18 bytes ea)
PJ2 DB 'B (J2) (E8H) '$'
PJ3 DB ' (J3) (E8H) '$'
PXX DB ' (' '$'
PXH DB '? '$'
PXL DB '?H) '$'
HNDTBL DB ' ** NONE ** '$' ; handshaking
                strings
DB ' EXT/ACK '$'
DB ' DC3/DC '$'
DB 'RTS positive (pin 4) '$'
DB 'RTS negative (pin 4) '$'
DB 'DTR positive (pin 20)$'
DB 'DTR negative (pin 20)$'

```

*** HELP MESSAGE STORAGE ***

```

HELP DB CR,LF,' CF - Configuration Utility'
DB ' Ver. 2.0 By Jeff Kalis',CR,LF,LF
DB 'CF ?<return>',14 DUP (' '),
DB '- displays this help message'
DB CR,LF,LF
DB 'CF<return>',16 DUP (' '),
DB '- displays current PRN and AUX: '
DB 'configuration',CR,LF,LF
DB 'CF S <d:filename><return> - saves the
DB 'current PRN and'
DB 'AUX: configuration',CR,LF
DB 28 DUP (' '),
DB 'in the file <filename> on'
DB 'drive <d:>',CR,LF,LF
DB 'CF <d:filename><return> - configures
DB 'the PRN: and AUX:'
DB 'devices to the ',CR,LF
DB 28 DUP (' '),
DB 'data previously saved'
DB 'in <filename> on drive <d:>',CR,LF,'$'

```

*** BEGIN PROGRAM CODE HERE ***

```

RUN: PUSH ES
MOV AX,BASE_SEG
MOV ES,AX

```

```

MOV SI,WORD PTR ES:CTADDR ; First thing to do is
MOV AX,WORD PTR ES:[SI+2] ; get the configuration
MOV WORD PTR PRN_VEC,AX ; vector table locations
MOV AX,WORD PTR ES:[SI+4]
MOV WORD PTR AUX_VEC,AX
POP ES

MOV BX,DTA
CMP BYTE PTR [BX],0 ; If there is nothing in
; the input buffer,
CFH: JNZ CFH ; jmp to display the current
JMP DISINFO ; configuration information
INC BX
INC BX
CMP BYTE PTR [BX],'?'
JNZ CF2
MOV DX,OFFSET HELP
MOV AH,PSTRNG ; Print the help message
; if ? was entered

INT SYS_CALL
INT B2DOS

CF1: INC BX
CF2: CMP BYTE PTR [BX],' ' ; Skip over spaces
JZ CF1
MOV AL,[BX] ; Convert the first byte
; to lower case

OR AL,' '
CMP AL,'s' ; If the first byte is
; not a 's' then
JNZ LINFO ; go to the load informa-
; tion routine.

CF3: INC BX ; The next byte after
; the 'S' must be blank
CMP BYTE PTR [BX],' ' ; to stay in the save
; info routine else
JNZ LINFO ; go to the load info
; routine

SINFO: ; Save the configuration data in the
; filename specified.
; 'S' was entered as the first filespec
; and was
; placed in FCBI The filename we want
; would be
; in FCBI2

MOV CX,12
MOV DI,OFFSET FCBI ; Move the filename into
; the default FCBI
MOV SI,OFFSET FCBI2 ; while checking for
; wildcard characters

FMOV: LODSB
CMP AL,'*' ; Is char '*'
JNZ SIN1

OUT1: MOV DX,OFFSET ERR1 ; Print illegal name
; message if a
EXIT: MOV AH,PSTRNG ; wildcard was found
INT SYS_CALL
INT B2DOS ; Return to DOS

SIN1: CMP AL,'?' ; Is char '?'
JZ OUT1
STOSB
LOOP FMOV ; Loop until all 12 bytes
; are moved

MOV DX,OFFSET FCBI
MOV AH,CFILE ; Create a new file
INT SYS_CALL

OR AL,AL ; Check for successful open
JZ SIN2 ; branch if it was OK

MOV DX,OFFSET ERR4 ; Point to 'unable to
; open' message
JMP SHORT EXIT ; then back to DOS

```

```

SIN2: MOV FCBI+32,0 ; Record #0
MOV FCBI+14,REC_LEN ; Set record size
CALL MVINFO ; Move the configuration
; data to the DTA

MOV DX,OFFSET FCBI
MOV AH,WRITE ; Write the record
INT SYS_CALL

OR AL,AL ; Branch if successful
JZ SIN3

MOV DX,OFFSET ERR3 ; Write error message if not
JMP EXIT

SIN3: MOV DX,OFFSET FCBI
MOV AH,CLOSE ; Close the file
INT SYS_CALL
MOV DX,OFFSET OK1 ; Print the function
; complete message
JMP EXIT ; and return to DOS

MVINFO: ; Subroutine to move configuration
; tables to DTA

MOV CX,REC_LEN ; bytes to move
MOV DI,DTA ; Destination
MOV SI,WORD PTR PRN_VEC ; Source index
MOV DX,BASE_SEG ; Segment start

SLP1: PUSH DS ; save old segment on
; stack
MOV DS,DX ; new segment to DS
LODSB ; Get byte from 40:[SI]
POP DS ; restore old DS
STOSB ; Store byte DS:[DI]
LOOP SLP1 ; loop till done
RET

LINFO: ; Load the specified file and move the
; configuration
; data to the tables then call the
; configuration routines.

MOV DX,OFFSET FCBI ; Open the file
MOV AH,OPEN
INT SYS_CALL

OR AL,AL ; Check for proper open
JZ LIN1

MOV DX,OFFSET ERR2 ; Error if not
JMP EXIT

LIN1: MOV FCBI+32,0 ; Record #0
MOV FCBI+14,REC_LEN ; Set record size
MOV DX,OFFSET FCBI
MOV AH,READ ; Read the first record
INT SYS_CALL

MOV CX,REC_LEN ; Number of bytes to
; move
MOV BX,WORD PTR PRN_VEC ; Destination offset
MOV AX,BASE_SEG ; Destination segment
MOV ES,AX ; into ES
MOV SI,DTA ; Source location

LLP1: LODSB ; Get a byte from DTA
MOV ES:[BX],AL ; Store it in table
INC BX
LOOP LLP1 ; loop until done

MOV AX,0300H
MOV BX,WORD PTR PRN_VEC
CALL D:WORD PTR PRNF ; Call PRN config-
; uration routine

MOV AX,0300H
MOV BX,WORD PTR AUX_VEC
CALL D:WORD PTR AUXF ; Call AUX: configura-

```



```

tion routine
JMP SHORT SIN3 , DONE
DISINFO: ; Get the configuration data from the tables and
; display the analysis on the CRT
CALL MVINFO , Move the configuration
data to DTA
MOV DX,OFFSET PT1
MOV AH,PSTRNG ; Print PRN header
INT SYS_CALL
MOV BX,DTA , Point to PRN: data
CALL ANALYZE ; Analyze the data
MOV DX,OFFSET PT2
MOV AH,PSTRNG ; Print AUX header
INT SYS_CALL
MOV BX,DTA+35 ; Point to AUX: data
CALL ANALYZE , Analyze the data
JMP SIN3 , DONE
ANALYZE: ; Analyze the data from the table and print the
; appropriate message on the CRT
MOV AL,[BX]
CMP AL,3 ; If the first byte is
0, 1 or 2, continue
JB CONT1
UNKNOWN:MOV DX,OFFSET UNK1 ; else print 'unknown
device class'
MOV AH,PSTRNG ; message then return
INT SYS_CALL
RET
CONT1: MOV DX,OFFSET DEV1
MOV AH,PSTRNG , Print 'is' message
INT SYS_CALL
MOV AL,[BX]
DEC AL
JZ SERIAL
PARALL: DEC AL ; Back if class = 0
JNZ UNKNOWN
MOV DX,OFFSET PAR1
MOV AH,PSTRNG , Print 'parallel' message
INT SYS_CALL
ADD BX,2
CALL CHKPRT
MOV DX,OFFSET CRLF
MOV AH,PSTRNG
INT SYS_CALL
RET
SERIAL: MOV DX,OFFSET SER1
MOV AH,PSTRNG ; Print 'serial' message
INT SYS_CALL
ADD BX,4 , Move BX pointer to baud
rate byte
MOV AL,[BX]
AND AL,0FH
MOV AH,6
CALL TIMES ; Multiply AL times AH
- return in AX
MOV DX,OFFSET BAUD
ADD DX,AX ; print baud rate
MOV AH,PSTRNG
INT SYS_CALL
ADD BX,2 ; Point BX to word and
parity byte
MOV DX,OFFSET WRDLEN
MOV AL,[BX]
AND AL,0FH ; Isolate bit data
SHR AL,1
SHR AL,1
MOV AH,14
CALL TIMES
ADD DX,AX , AX contains offset
MOV AH,PSTRNG
INT SYS_CALL
MOV AL,[BX]
MOV CL,6 ; Isolate stop bit data
tion routine
SHR AL,CL
SUB AL,1
MOV DL,8
MUL DL
MOV DX,OFFSET SBITS ; Point DX to correct message
ADD DX,AX
MOV AH,PSTRNG , Print it
INT SYS_CALL
MOV AL,[BX]
AND AL,010H , Check parity enable bit
JNZ PENA
MOV DX,OFFSET DISA
JMP SHORT PPAR
PENA: MOV DX,OFFSET EVN
MOV AL,[BX]
AND AL,020H
JNZ PPAR
MOV DX,OFFSET ODD
PPAR: INT SYS_CALL ; Print it
SUB BX,4 , Point BX to port byte
CALL CHKPRT ; Print port assignment
ADD BX,3 , Point to handshake bit
MOV AL,[BX]
AND AL,7 ; Isolate bits
MOV CL,22
MUL CL
MOV DX,OFFSET HNDTBL
ADD DX,AX
MOV AH,PSTRNG
INT SYS_CALL ; Print handshaking
MOV DX,OFFSET CRLF
INT SYS_CALL
RET
CHKPRT: MOV AL,[BX]
CMP AL,0E8H
JNZ NOT_E8
MOV DX,OFFSET PJ1 , If the port is E8 print
this message
JMP SHORT CONT2 ; else
NOT_E8: CMP AL,0ECH
JNZ NOT_EC
MOV DX,OFFSET PJ2 ; If the port is EC print
this message
JMP SHORT CONT2 ; else
NOT_EC: CMP AL,0E0H
JNZ NOT_E0
MOV DX,OFFSET PJ3 , If the port is E0 print
this message
JMP SHORT CONT2 ; else
NOT_E0: MOV CL,4
SHR AL,CL
CALL H2ASC ; convert the port to
ASCII and print it
MOV PXH,AL
MOV AL,[BX]
AND AL,0FH
CALL H2ASC
MOV PXL,AL
MOV DX,OFFSET PXX
CONT2: MOV AH,PSTRNG , Print the port assignment
INT SYS_CALL
RET
H2ASC: AND AL,00FH
OR AL,030H ; Converts the right half
of AL to
an ascii byte in AL
CMP AL,03AH
JB H2RET
ADD AL,7
H2RET: RET
TIMES: MOV CL,AH ;Multiply AL by AH
XOR AH,AH
MUL CL
RET
CODE ENDS
END START

```

Call *First Capitol* for the best prices and custom configurations



Z-158 "SuperPak" — our newest supersystem

- one floppy
- one 10 meg removable cartridge winchester drive, built-in back up and data security . . . the same drive used in the Tempest high security system.
- one 21 meg (formatted) fixed winchester drive
- 640 K RAM
- Monochrome or color monitor

Internal and external tape backup systems available
30 meg and larger high speed drives available

FREE Hayes compatible 300/1200 external modem with 20 and 30 meg systems, through May 31 only!



Z-200 "SuperServer" — ideal as LAN fileserver or super CAD station

- 62 meg (formatted), 28ms access drive
- Network hardware included
- Optional high resolution video boards/monitor available
- 69 meg high speed tape backup
- Backup power supply

Call for winchester Z-148 configurations and prices

Authorized AutoCAD, Novell, and Zenith Data Systems dealer

For the Z-151/161

Complete upgrade to 640K, RamPal from Software Wizardry plus 18 41256 RAM chips . . . only \$85.00

New! Fastcard Superboard for Z-151/161/158, the only board you'll ever need! Up to 2 Mb expanded memory (EMS), serial, parallel, and game ports, clock/calendar, ram disks, disk caching, print buffering, security password and diagnostics. Call for info and price.

For the Z-110/120

Complete 768K motherboard upgrade, RamPal-100 from Software Wizardry plus 27 41256 RAM chips . . . only \$145.00

Complete memory/speed upgrade, new switchable 5/8 MHz Software Wizardry speed upgrade, RamPal-100 from Software Wizardry plus 27 41256 RAM chips . . . only \$275

**First
Capitol
Computer**

ZENITH | **data
systems**

AUTHORIZED SALES AND SERVICE

First Capitol Computer
1106 First Capitol Dr.
St. Charles, MO 63301
(314) 946-1968

Note: Due to magazine production delays before appearance of this ad, we ask that you call us for our current and best prices.

Patch Page

Pat Swayne
HUG Software Engineer

This article presents patches for the HUG Background Print Spooler (PC version) from HUG disk 885-3029-37, and the Z-100 KEYMAP program from HUG disk 885-3010-37.

HBPS Patch

The PC version of the HUG Background Print Spooler that was initially released in the December 1985 REMark contains a problem that may affect operation with some programs. If you attempt to print with the spooler enabled, and your program issues a "printer not ready" or other printer error message; and if the program works properly with the spooler disabled, the following patch can be used to correct HBPS. This patch applies only to the files HBPS.PC2 dated 10-02-85, and HBPS.PC1 dated 10-15-85. It is assumed that you have copied one of these files to your system disk, and renamed it to HBPS.COM.

To install the patch, first use an editor to create a file called FIX-HBPS.BAT that contains this line:

```
DEBUG HBPS.COM <FIXHBPS.DAT
```

Now create a file called FIXHBPS.DAT that contains the lines shown below. Use the lines from the first column if you use HBPS.PC1, and the lines from the second column if you use HBPS.PC2.

E19A	E1AA
1F	1F
E19F	E1AF
1A	1A
E1A4	E1B4
1A	1A
E1AE	E1BE
0B	0B
A1B8	A1C8
JNZ 1DD	JNZ 1ED
JMP 0:0	JMP 0:0
CS:	CS:
MOV [103],SP	MOV [103],SP
CS	CS:
MOV [105],SS	MOV [105],SS
PUSH CS	PUSH CS
POP SS	POP SS
MOV SP,103	MOV SP,103
STI	STI

CALL 1E1	CALL 1F1
CLI	CLI
CS	CS:
MOV SS,[105]	MOV SS,[105]
CS:	CS:
MOV SP,[103]	MOV SP,[103]
MOV AH,D0	MOV AH,D0
IRET	IRET
A3DC	A47F
MOV [1BB],DI	MOV [1CB],DI
MOV [1BD],ES	MOV [1CD],ES
W	W
Q	Q

Copy the files FIXHBPS.BAT, FIXHBPS.DAT, and DEBUG.COM (supplied with MS-DOS) to the system disk containing HBPS.COM, log on to the disk, and enter

```
FIXHBPS
```

at the system prompt, and hit RETURN. The patch will be installed automatically. **Note:** You must use the MS-DOS version 2 (or higher) release of DEBUG.COM to make the patch. If you are using HBPS.PC1 under MS-DOS version 1, copy everything over to an MS-DOS version 2 disk, make the patch, and then copy HBPS.COM back to your MD-DOS version 1 disk.

If you would like to fix the source code for HBPS, load the file HBPSPC.ASM into an editor, and locate these lines (comments on the lines shown are "chopped off" to fit in the column):

JZ	MYEXIT	;EMPTY. OK TO CHECK
MOV	AH,0D0H	;ELSE, JUST RETURN G
IRET		
MYEXIT:	JMPF	;JUMP TO INT 17 PROC
I17ADR	DD	0
		;INT 17 ADDRESS
PRNOUT:	MOV	CS:SYSSTK,SP
	MOV	CS:SYSSTKS,SS
	PUSH	CS
	POP	SS
	MOV	SP,OFFSET LSTAK
	STI	
	CALL	MYPROUT
		;PRINT THE CHARACTER
MYRET:	CLI	
	MOV	SS,CS:SYSSTKS
	MOV	SP,CS:SYSSTK
	IRET	;RETURN, FUNCTION CO

Change the lines so that they look like this:

```

JNZ COMRET ,NOT EMPTY, RET GOOD
MYEXIT: JMPF ;JUMP TO INT 17 PROC
I17ADR DD 0 ;INT 17 ADDRESS
PRNOUT: MOV CS:SYSSTK,SP ,SAVE STACK
MOV CS:SYSSTKS,SS ,AND STACK SEGMENT
PUSH CS
POP SS ;PUT STACK SEGMENT H
MOV SP,OFFSET LSTAK ;SET LOCAL STACK
STI
CALL MYPROUT ;PRINT THE CHARACTER
MYRET: CLI
MOV SS,CS:SYSSTKS ;RESTORE SYSTEM STAC
MOV SP,CS:SYSSTK
COMRET: MOV AH,0D0H ;RETURN GOOD STATUS
IRET ;RETURN, FUNCTION CO
    
```

Now save the edited file, and reassemble it if you wish.

KEYMAP Patch

If you purchased MS-DOS version 3 for the Z-100, and if you attempt to run the MS-DOS 2 version of the HUG KEYMAP program under it, you will get a "wrong version of MS-DOS" message. That is because the program tests explicitly for MS-DOS version 2 rather than testing for version 2 or above. The problem is easily fixed in any configured or unconfigured KEYMAP program that you may be using. Just use an editor to create a file called KEYFIX.DAT that contains these lines:

```

AA3E
JNB A47

W
Q
    
```

Copy KEYFIX.DAT to the disk containing the KEYMAP file you want to fix along with DEBUG.COM, log on to the disk, and enter

```
DEBUG KEYMAP2.COM <KEYFIX.DAT
```

at the system prompt, and hit RETURN. Substitute the name of the actual KEYMAP file you are patching (KEYWS2.COM, etc.) for KEYMAP2.COM in the above line. The patch will be installed automatically for you.

If you would like to patch the source code, load KEYMAP2.ASM into your editor, and locate these lines:

```

SETUP MOV AH,48
INT 21H ;GET DOS VERSION
CMP AL,2 ;DOS 2 0?
JZ DOSOK ;YES
    
```

Change the fourth line to read

```
JNB DOSOK ;YES
```

Then save the edited file, and reassemble it if you wish.





Secured Computer Systems

HEATH/ZENITH 88, 89, 90 PERIPHERALS

16K RAM EXPANSION CARD

Expand your H/Z 88, 89 RAM Memory to a **FULL 64K** and begin using larger and more powerful programs with our **16K RAM card**

Fully compatible with: Magnolia Microsystems and COR CP/M and disk I/O interface cards

Featuring: Complete installation instructions • Mounting bracket • 90 day warranty

Field reliability record exceeding 3 years

Only \$65.00
Shipping & Handling \$5.00



PORT SERIAL CARD I/O

2 PORT PARALLEL

"... not your typical vanilla-flavored serial and parallel interface"

Your H/Z 88, 89, 90 can now directly connect and operate EPSON, IDS, ANADEX, GEMINI, SILVER REED, NEC, SUPER 5, PROWRITER, OKIDATA, and many more line printers using CENTRONICS style parallel interface with our 2/3rds, 2 port serial, 3 port parallel interface card. Or you may use all 24 digital lines in various configurations of input, output, or bidirectional modes for industrial control or data sampling.

Features:

- 2 Serial Ports Supporting Ring Input, and External Clock
- 3 Parallel Ports, 24 Total Digital Input/Output Lines
- Fully Compatible with All Models of H/Z 88, 89, 90 using Heath/Zenith CP/M or HDOS
- Now Supporting CP/M Version 2.2.04
- Choice of Centronics Line Printer Support Software for the CP/M or HDOS Operating Systems
- Reduced Computer Bus Loading and Chip Independent Design

Complete with installation instructions, documentation, 90 Day Warranty, two Serial Cables and a Parallel Cable Internal to the Computer

Price \$199.00 **Second Operating System Driver \$25.00**
Shipping & Handling \$10.00



REAL TIME CLOCK

You will be able to perform time and date stamping for point of sales software, and bulletin board software or perform time studies as well as real time data sampling with our **REAL TIME CLOCK**. This peripheral card is a perfect companion to our 2/3rds card for industrial control and data sampling. STOP WATCH time study and alarm demonstration software is included for the CP/M operating systems. You will be able to view the current date and time on screen continuously or simply listen to an audible beep every fifteen minutes and the hour chimed or disable the clock entirely at your option.


Features:

- True I/O Addressing, Not Memory Mapped
- User Selectable Address
- Rechargeable Battery Backup Using Commonly Available Batteries
- Installable on the Left or Right Side of the Computer (Left side operation requires our I/O expansion module)
- Month, Day, Year, Hour, Minutes
- Seconds, 1/10's, 1/100's and 1/1000's of Second of Accuracy
- Interrupt Capability Based on Tenths of Seconds, Seconds, Minutes, Hour, Day, Week or a Specific Date and time
- CP/M Operating System Software Driver and Demonstration Programs
- HDOS is not Supported

Price \$130.00 with Batteries **\$114.00 without Batteries**
Shipping & Handling \$5.00



HDOS is a reg. trademark of the Heath Co CP/M is a reg. trademark of Digital Research
 PRICES ARE LESS SHIPPING AND TAX IF RESIDENT OF CALIFORNIA
 MAIL ORDER: 12011 ACLARE ST., CERRITOS, CA 90701 (213) 924-6741
 TECHNICAL INFO/HELP: 8575 KNOTT AVE., SUITE D, BUENA PARK, CA 90620 (714) 952-3930



TERMS AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE — VISA AND MASTER CARD GLADLY ACCEPTED

Subdirectories And Command Pathing

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

Before we get too involved in MS-DOS subdirectories, I thought I would tell you about the current status of the PC emulation articles, specifically the UCI Easy PC. Based on various reports, it appears that, as of the date of this writing, the Easy PC is not as production ready as we thought it would be. Although UCI is shipping the Easy PC to some vendors, Henry Fale reports some problems with an Easy PC in the January H-SCOOP (#70). The first problem was relatively minor — a speaker wire had been crimped to a terminal with the insulation intact. The second, and perhaps more serious, problem was an apparent problem in booting up Barry Watzman's MP/M-86 implementation on the Z-100. It just would not complete the booting process. Although I do not have Barry's MP/M-86, I do have his CP/M Plus, and I have found no problems with booting that on my Gemini-equipped '100.

Since it is obvious that the series of three articles on PC emulation will be delayed, the article on the Easy PC and the summary article will still appear in REMark as soon as it is available from Heath.

Beating A Dead Horse (Copy Protection)

In the same issue, Henry also reports a problem with running the Sub-Logic Pinball game program with the Easy PC using the CDR Z-100 Speed-Up kit running at 7.5 mhz. It is interesting to note that he reports that Pinball seems to run just fine at the standard 4.77 mhz clock speed. Although I do not know the specific details on the Pinball program, this report is symptomatic of another problem that all users should be aware of. The problem is simply that some copy-protected software will just not run at anything faster than 4.77 mhz. I will not go into the technical details of why this happens, but it is directly related to the software, not the hardware. The point is that some of the copy-protected software

will not run at the 8 mhz clock speed on any hardware, period. That is not a compatibility problem with the Gemini, the Easy PC, the H/Z-100 or the H/Z-158s. It is just a software "feature".

The cure for that is relatively simple — just reduce the speed of your equipment to the standard 4.77 mhz (usually referred to as 5 mhz). For H/Z-100 owners, that will require a switchable clock speed using the CDR Speed-Up kit or the UCI Speed-Up kit. Owners of the H/Z-158s are lucky since Heath and Zenith wisely included that switchable feature in these machines. And if you still don't know why I am personally opposed to copy protected software, this is just another reason. Copy protection has turned into a support headache for vendors and a gigantic pain in the neck for users.

There is another form of copy protection that can cause some really strange problems. This form of copy protection relies on the fact that the disk drive rotational speed must be nearly perfect or the software will not run. Henry reported that a user had problems in running the GATO software on a Z-150. The end result of all of this is that, although Software Consultation was not aware of any Z-150/GATO "compatibility" problems, the vendor of the software was.

GATO is copy protected and requires that the disk drive rotational speed must be essentially at the standard of 300 rpm for 5.25" drives. Apparently, the GATO vendors know of some problems with running GATO on the Heath '150s which is reported to be disk drive related. GATO is sensitive to the rotational speed, and in addition, apparently some of the '150 kit builders have not properly adjusted the disk controller bias setting.

Well, OK. The point is that, if you are building a computer kit, I suggest that you have the disk controller adjusted at a Heathkit

store, if at all possible. If that is not possible, you should make sure that you have the proper equipment for the adjustment and follow all Heath instructions exactly. As reported last month, I built a '150 kit for a friend, and ended up buying a good DVM (SM-77) at the Dallas store when I took the controller in for adjustment.

The bottom line is that some copy protected software is hardware sensitive. One of the assumptions that Software Consultation must make is that your hardware is in good working order and adjusted to specifications. If the hardware has not been adjusted properly or the performance characteristics have changed because of mechanical wear, all bets are off. That also applies to the IBM PC. GATO will not run on a PC if the rotational speed is off due to mechanical wear.

A good way for you to check your disk drives is to use the Zenith Disk Diagnostics software which provides a rotational speed test for floppy disk drives. I use it on a regular basis to check my system hardware. It's very useful for monitoring potential trouble areas like disk drive rotation speed. I keep my system adjusted so that I am within ± 1 rpm of the 300 rpm standard on the 5.25" drives and 360 rpm for the eight inch drives.

And a final note on the subject... if you are having problems running software on your '100 or '150 series computer, make sure that it isn't a hardware problem. Then, you should check with Zenith Software Consultation. Most of these software problems are generally known. And now, onward to MS-DOS subdirectories...

Using MS-DOS Subdirectories —The Good News

There are a number of good reasons for using subdirectories. Perhaps the best reason to use subdirectories is that you can organize disk files which can be particularly difficult on a hard disk because of the high capacity. Another reason is that you can simply add more files to a disk through the use of subdirectories. That may be particularly important in the case of a high capacity hard disk. More on that in a minute.

Subdirectories can be used to organize programs by function such as system, word processing, and spreadsheet. Subdirectories can be used to organize data by application such as outside letters, inside memos, payroll, personnel or whatever. I use different subdirectories which are related to articles, books, and various business functions. For example, I have joined the faculty at the University of Texas at Arlington, so I have a UTA subdirectory.

Another advantage of subdirectories is that, with proper organization, they can be used to shorten the listing of files on the screen when the DIR command is used. In many cases, the use of subdirectories can make it much easier to perform regular file backups on both floppy disks and hard disks. For example, there is no reason to continuously backup program disks (and associated subdirectories) unless you have added software since the last backup was made. The same is true for data disks and subdirectories. Some data may change daily, some weekly, and some on a monthly basis. Daily changes may include such things as memos created by a word processor or file updates to an accounting package. Monthly changes may include such things as closing the books at month end and sending invoices.

Some Facts About Subdirectories

All MS-DOS based systems, including Z-DOS and PC-DOS, have always had a directory which was called a ROOT directory. In-

deed, one of the functions of the FORMAT program is to initialize the disk by creating the basic information such as the boot loader, File Allocation Table (FAT), and the disk directory that we discussed last month. It is important to recognize that the physical size of the disk directory (in bytes) is a function of the physical characteristics of the disk. That is, the FORMAT program creates a specific number of sectors for the disk directory depending on the physical disk type (i.e. 5.25", 8" or hard disk) and the physical characteristics of the disk (single/double sided and single/double/quad density). The number of sectors created for the various disk types and characteristics is documented in the Zenith MS-DOS documentation under the FORMAT command. If you know that each directory entry is 32 bytes long and that a sector is 512 bytes for 5.25" formats, you can quickly calculate that each sector contains 16 directory entries. And since both standard DS/DD formats (8 and 9 sector) have seven directory sectors, it is also easy to calculate that a total of 112 ROOT directory entries are available on a standard disk.

Aha! That is the source of the limitation of the number of files that can be added to a disk — the physical size of the disk directory. Since there is some variation in the disk formats, I have shown the maximum number of root directory entries available for various disk formats as Listing 1. You will probably want to remember the maximum number of root directory entries for your normal disk format since that is the largest number of files that can be added to the root directory. And, if you want to know how many of the root directory entries have been used, you can use the DIR command. The number of files listed by the DIR command is basically the same as the number of directory entries. Why did I say "basically the same?" The answer is that some files on a bootable disk are hidden files. That will normally include the BIOS (IO.SYS), the system kernel (MSDOS.SYS), and the label file if one has been defined. Even though these files are hidden from the DIR command, they may take as many as three directory entries if a label file has been defined. Other system-related files, such as COMMAND.COM and ALTCHAR.SYS (H/Z-100s only), are displayed by the DIR command.

For example, let's say that the DIR command reports that there are 42 files on a 5.25" DS/DD disk. If the disk is bootable, there will be at least two additional files, IO.SYS and MS-DOS.SYS. So that is 44 used directory entries. If we have defined a label for the disk, we have used an additional directory entry for a total of 45 used directory entries. Subtracting that from the maximum number of available directory entries (112 from Listing 1), we find that we have room for 67 more files.

Why is that important? Let's assume that the disk contains the 42 normal files and the three hidden files which consume a total of 100 kilobytes of disk space. Furthermore, we'll assume that we're using a 5.25", 9 sector, DS/DD floppy disk. That means that the first 12 sectors (6 clusters) are reserved for the boot loader, FAT1, FAT2, and the root directory in that order. From a disk space perspective, we'll assume that our 45 files use a total of 100 sectors out of the maximum 354 available. We have discussed the idea that a cluster is 2 sectors (1 kilobyte) in previous articles. The bottom line is that we currently have 67 directory entries and 254 clusters (kilobytes) available for our use.

Let's begin to add a number of small files (less than 1KB in size) to our disk. Regardless of the size of the file, each file requires a directory entry. We will exhaust the remaining 67 root directory entries with these small files long before we use up the remaining 254K of disk space. And that is the point! You can add more small

files to any disk without the absolute limitations imposed by the number of root directory entries.

In our bootable disk example with the three hidden files, we still have a maximum of 109 directory entries assuming the use of the three hidden files. That means that we could still define 109 subdirectories and still more below that. Now the effective limitation is just the space available on the disk. And that's another advantage of using subdirectories. The number of files is essentially limited by the available disk space, not the number of root directory entries.

One other interesting fact about subdirectories is that a subdirectory is nothing more than a special file name in the root directory. As you might expect, it has a special file attribute byte (10H to be specific) that identifies it as a subdirectory.

Disadvantages Of Subdirectories

The bad news in using MS-DOS subdirectories is that you will observe that system performance is significantly slower in a floppy disk system. On a hard disk, it's hardly noticeable because of the faster disk I/O. But you will find that it's difficult to get something for nothing.

For the same reason, you will find that the creation of a large number of subdirectory levels is not recommended since it will cause even slower performance. That is particularly important to the discussion of the PATH command which we'll talk about later.

MS-DOS Subdirectories

One of the features that is available is MS-DOS Version 2 and later is the use of the subdirectory structure. Subdirectories are organized in a hierarchical directory structure that provides an enhanced method of file access and multi-level directories. As previously mentioned, this simplifies grouping of files by project or function (i.e. word processing, spreadsheets, etc.) and improves overall disk organization, particularly on a high capacity hard disk with lots of files.

Subdirectories are organized in a tree structure — with the main (root) directory at the root and other directories branching from the main trunk or other limbs. This treelike structure brings about the common name of a tree-structured directory or simply, a tree directory. As a tree has a root that serves as its foundation, the tree directory also has a root that serves as its foundation for all files and subdirectories on the disk drive. This first directory, called the ROOT directory, can contain file names, as well as entries for other subdirectories. Remember that the subdirectory name is a standard file name (directory entry) with a special attribute byte.

Like the root directory, a subdirectory can also contain file names and entries for other subdirectories. In version 1 of Z-DOS and PC-DOS, all directory entries were contained in the root directory. In version 2 of MS-DOS and PC-DOS, directory entries may be contained in the root directory or in any defined subdirectory.

Subdirectory Terminology And Restrictions

A sample of this hierarchical structure is shown in Figure 1. As you have noticed, the main directory is referred to as the ROOT directory. Subdirectories can be defined in the root directory which are one level below it. In this case, the root directory is the PARENT directory, and each subdirectory below it is a CHILD.

Figure 1 shows that the PROGRAMS and DATA subdirectories are children of the parent ROOT.

Similarly, we can define children for each of those subdirectories. The PROGRAMS subdirectory has two child subdirectories: SYSTEM and APPLICAT. The DATA subdirectory also has two child subdirectories: WORD-PRO and SPREADSH.

Now, if we wanted to get to the SYSTEM subdirectory, we would start at the root directory, and follow the PATH through the PROGRAMS subdirectory to the SYSTEM subdirectory. In this example, the path, as used in a command would be constructed as: \PROGRAMS\SYSTEM. Assuming that our CURRENT WORKING DIRECTORY was the root directory, we could execute a program, such as FORMAT, that was contained in the SYSTEM subdirectory, by entering the following command line: \PROGRAMS\SYSTEM\FORMAT B:. Since that is more than a little awkward, we will look at a better way when we discuss the path command later.

Some documentation reports that the maximum number of subdirectory levels is 255; other documentation says 32. Whichever is true, I haven't tested it simply because anything much more than three levels leads to significant system degradation. Plus, I don't consider it worthwhile to construct complex subdirectories since it is just too difficult to remember how everything fits together. My recommendation is that you only use the first subdirectory level, since that provides most of the benefits with the least impact on your system performance. You may, on occasion, find it beneficial to have one additional subdirectory level, but that will be rare.

There are, of course, some restrictions on the definition of subdirectories. Perhaps the most obvious is the fact that a subdirectory is nothing more than a file name (with a special attribute), and like file names, the following characters cannot be used in subdirectory names:

< > , ; : = + ? * [] | / \ space tab

You will probably recognize that most of these characters can be used in command lines in one way or another.

Some of the documentation on MS-DOS indicates that subdirectory names are limited to eight characters. One individual at last year's HUG Conference told me that that was not a specific restriction in MS-DOS. That is true, since a subdirectory name is a file name, and there is no technical reason that it could not be in the form of filename.typ. Since MS-DOS documentation I have seen generally does not suggest that subdirectory names can have a file type, I advise that you only use the first 8 characters. Documentation for all application programs that I have seen, like word processors, shows a standard subdirectory name with only eight characters. If you try to get too fancy and have subdirectory names in the form of filename.typ, you may find that some application software will not function properly with that long a subdirectory name. On a more practical level, it is to your benefit to keep subdirectory names as short as possible.

The final restriction is that, when used in a command, subdirectory names and file names must be separated by a backslash (\).

Setting Up Your Own Subdirectories

Since it is not immediately obvious how to efficiently set up some subdirectories, I have shown a sample set-up as Figure 2. Note that the root directory contains the most often used commands

in addition to some other files that are necessary for system booting. I suggest that you keep any frequently used commands in the root directory. ZDIR.COM is a sorted directory program provided with the Programmer's Utility Pack that I use quite frequently. I also keep the DIR100 HUG Program in that same directory since I use that a lot. The reason for keeping frequently used commands in the root directory will be more obvious when we look at pathing.

System files, such as COMMAND.COM, are required in the root directory for a successful boot. For the H/Z-100, ALTCHAR.SYS should also be in the root directory. AUTOEXEC.BAT must also be in the root directory if you use that feature. And the optional CONFIG.SYS file must also be in the root directory when used for system configuration. Although you can use path names in the CONFIG.SYS file for the DEVICES parameter, I recommend that all device drivers (if implemented) be contained in the root directory just because they are easier to keep track of.

The SYSTEM (some people prefer DOS or SYS) subdirectory contains all of the basic MS-DOS system-related commands such as FORMAT, DISKCOPY, etc. The HARDDISK subdirectory contains all of the commands related to my hard disk such as PART, SHIP, and so on. The PROG subdirectory contains all of the programming-related commands such as MASM, CREF, LINK, and EXE2BIN. UTIL contains most of the other programs from the Programmer's Utility Pack. WWORD contains WatchWord and the Resident Speller. ACCESS contains the HyperAccess program and related telecommunications information. PERKS contains all of Barry Watzman's Perks programs. And FONTS contains the Z-100 FONT command plus some other things that I was testing.

Note that I have specifically arranged these subdirectories in the order that I most frequently access them. That's an important concept when we discuss the PATH command. Even though I usually load Perks through an AUTOEXEC.BAT file, it is toward the end of the structure since that is only done once each session.

That's a discussion of the MS-DOS subdirectories from a conceptual point of view, now let's look at some of the specific commands for working with subdirectories.

Creating A Subdirectory — The MKDIR Command

Subdirectories are easy to create with the Make Directory (MKDIR) command. Refer back to Figure 1. Assuming that the current working directory is the root directory, we can create the first subdirectory, PROGRAMS, with the following command:

```
MKDIR PROGRAMS
```

Once that directory is created, we can then create the SYSTEM and APPLICAT subdirectories with the following commands:

```
MKDIR \PROGRAMS\SYSTEM
MKDIR \PROGRAMS\APPLICAT
```

The DATA subdirectory and its children are created in the same way. And if you want to reduce the number of keystrokes for this command, you can simply use MD (instead of MKDIR) which will produce the same results.

One of the features of all commands which allow a path specification is that a beginning backslash (\) is optional. If the beginning backslash is not specified (e.g. MKDIR PROGRAMS), the path begins at the current working directory. Since we started at the root directory anyway, the MKDIR PROGRAMS and MKDIR

\PROGRAMS are equivalent. When the beginning backslash is specified in a command, the path begins at the root directory. That will be more important when you change the working directory with the CHDIR command which we will look at in a minute.

Since it is rather easy to get confused during this process, I recommend that you always start in the root directory. More importantly, you should take the time to write down your directory structure before you enter any commands. In addition to making it easier to construct the command, you will also have complete documentation on your subdirectories.

Deleting A Subdirectory — The RMDIR Command

Now that you have created some subdirectories, we will find out how to delete them. That's done with the Remove Directory command or RMDIR. If we wanted to delete the SYSTEM subdirectory that we just created, we can use the command:

```
RMDIR \PROGRAMS\SYSTEM
```

The only additional requirement for the use of the RMDIR command is that the subdirectory must be totally empty. Like the MKDIR command, you can reduce the number of keystrokes for this command by using the alias RD.

Moving Around In Subdirectories — The CHDIR Command

Until this point, we have been forced to stay in the root directory. In this case, the root directory has always been the current working directory. The Change Directory command, or CHDIR, allows you to change to a different working directory or subdirectory. In order to get to the SYSTEM subdirectory, we can enter the command:

```
CHDIR \PROGRAMS\SYSTEM
```

The current working directory is SYSTEM which is the last directory in the path displayed by the DIR command. And if you think that CHDIR is too many keystrokes, you can use the alias CD.

Although I have shown two levels of subdirectories, I have suggested that only one level is very useful on a practical basis. The primary reason for that is that there are a couple of ways to shorten some of the commands.

For example, the command "CD \" will always return you to the root directory. And you can always go to the parent of the current subdirectory by entering the command "CD ..". Although the current subdirectory can also be indicated by a ".", I haven't found much use for that (e.g. CD). When you have changed to a subdirectory, however, you will see the "." and ".." at the beginning of the DIR listing. That indicates the current working directory and its parent, respectively. And if you forget which is your current subdirectory, just enter the CD command by itself. That will display the current disk drive and working directory.

And finally, there are a couple of ways to keep track of and document the total subdirectory structure. The first way is to use the Zenith SEARCH command as follows:

```
SEARCH * */T > DIRDOC
```

This version of the command will search all directories and direct the output listing of the directory tree structure (/D parameter) of the current drive to the file name DIRDOC. Then you can edit and/or print the listing to provide permanent final documentation of the subdirectories and files on a disk drive. A useful feature.

Some MS-DOS (and all PC-DOS) systems provide the TREE command which can be used in the same way. I personally prefer the SEARCH command because it has more options. And by the way, the SEARCH command can also be used to locate a file in a subdirectory in case you forget where a file is located.

Some Thoughts On Subdirectories

As you consider the creation of subdirectories, I recommend that you keep the subdirectory names and structure simple. Complex subdirectory structures are difficult to remember and will degrade system performance to some extent. Those are good reasons for using a single subdirectory level.

Perhaps the best way to create your own directory structure is to create subdirectories when you would normally change to a different floppy disk. That applies primarily to a hard disk system, of course, although many users may find it advantageous to create subdirectories on a system disk on a floppy system.

The structure you choose for your subdirectories will, of course, depend on your own specific applications and your method of keeping track of your files. Keep in mind, however, that the simpler the structure, the better.

MS-DOS keeps track of the current working directory for each drive just as it keeps track of the default drive. Whenever you boot your system or change floppy disks in a drive, the current directory is the root directory.

A directory path (or simply path) is a list of directories, in sequence, that the DOS must follow to get to a named file or subdirectory. Each subdirectory name, and the file name if used, must be separated by a backslash. If the path begins with a backslash, the DOS will start at the root directory regardless of the current working directory. If the path does not begin with a backslash, DOS will begin the path at the current directory.

A path name may also be preceded by a disk drive such as in the following: A:\SYSTEM\FORMAT. That is a valid command line which says: go to drive A, the SYSTEM subdirectory (under the root), and execute the FORMAT program. I will note in passing that I have found several versions of PC-DOS that do not correctly execute this command line. They change to the specified drive (in this case, A), and stop with no execution of the specified command.

Now that you understand something about subdirectories, it's time to take a look at another command — PATH. Even if you decide not to use subdirectories, you may find that including a PATH command in an AUTOEXEC.BAT file may be very useful.

The PATH Command

The PATH command provides the capability for the DOS to search for disk resident commands which includes batch files and application programs that are not located in the current directory. The general form of the PATH command is:

```
PATH [d:]path-1;[d:]path-2;[d:]path-3.
```

Note that each path name is separated by a semicolon (;) and each may be preceded by a drive letter. Refer again to Figure 2. Since I use a hard disk, each of my path names are preceded by E. I have included a PATH command in my AUTOEXEC.BAT file which is like the following:

```
PATH E:\;E:\SYSTEM;E:\HARDDISK;E:\PROG;E:\UTIL;E:\WORD...
```

Remember that I said that it is important to determine the order of the directory searches and place the most frequently used

commands in the root directory. Note that the first path specification is E:\. Regardless of the current working directory, that command tells the DOS to look at the root directory first if the command is not found in the current working directory. Since I only use one level of subdirectories, that is the most efficient way to use this feature. That's the reason that all of my most frequently used programs which provide directory lists (i.e. HUG's DIR100 and Zenith's ZDIR) are located in the root directory. Each of the subdirectory names is preceded by the drive so that, no matter which floppy disk drive or hard disk partition I am using, the DOS will always search the specified path if it does NOT find the program or batch file in the current working directory and/or disk drive.

All versions of MS-DOS search for commands in a specific order. First, DOS will search for built-in commands like DIR. Then, DOS will look for a COM file (e.g. FORMAT.COM), then an EXE file (e.g. MASM.EXE), and finally at batch (BAT) files. For example, if you have FORMAT.COM and FORMAT.BAT on the same drive in the same directory, DOS will never "see" (or find) FORMAT.BAT. DOS will find and execute FORMAT.COM since that is the defined DOS search order.

When a PATH command defines a path, DOS will first examine the command, and if it is a built-in command, DOS will execute it. Then, DOS will attempt to execute the command from the current working directory for a disk resident command. Failing that, DOS will continue its attempts to execute the command following the paths (subdirectories) specified in the PATH command. Command execution will be attempted in each specified path in the standard order of COM files, EXE files, and BAT files. If the command is not found in the first path, DOS will go to the next specified path and attempt command execution in that standard order. And so on.

Even if you do not have a hard disk system or use subdirectories, you may find it extremely useful to add a path command in the following form:

```
PATH A:\
```

That command, included in the AUTOEXEC.BAT file, will cause a search for a command in the root directory on drive A even if your default drive is displayed as drive B. If you use the system disk concept that I have mentioned in previous articles, you will find that this PATH command is very useful.

If you use the PATH command with subdirectories, you will get the best system response by planning ahead as I have mentioned. You should plan the search paths, such that you search the most frequently used subdirectories first and the least used last.

There is, of course, at least one significant disadvantage to the PATH command — directory searches are performed in the standard DOS order. Unfortunately, DOS will not search for overlay files (required with WordStar, for example) or help files (e.g. *.HLP) that are included with many of today's application programs. That's the bad news.

The good news is that MS-DOS Version 3, currently available from Heath and Zenith, provides a couple of new commands: JOIN and SUBST which can be used to overcome that search problem. They also can be used with programs that do not "understand" how to work with subdirectories, so you may find it possible to use some programs that do not have the capability to work with subdirectories. More on these new commands in a future article.

Hints And Kinks For Working With Subdirectories

It is obviously very easy to forget which subdirectory that you are currently using, particularly if you like to experiment as I do. Fortunately, there is an easy way to always display the current working directory with the PROMPT command. My personal preference is:

```
PROMPT $P $Q$Q$G      (displays E:\SYSTEM ==>_)
```

I have written a complete explanation on "How to Use the MS-DOS PROMPT Command" which appears in the March-April issue of Sextant. You may want to refer to it for some additional things that you can do with the PROMPT command.

If you want to see the current settings of the PATH and PROMPT command, just enter SET. You can also display the current path setting by entering PATH. And if you want to know the current working subdirectory, just enter CHDIR or CD.

When you create subdirectories, it is generally easier to create more subdirectories since it is easier to consolidate subdirectories than it is to split them up. The reason for that is that you cannot rename files to a different subdirectory. That would be like trying to rename a file to a different disk drive. Remember that a subdirectory entry can be treated just like a disk drive. Since the RENAME command does not work in this case, I wrote a simple batch file, called MOVEIT.BAT, which contains the following lines:

```
COPY %1 \%2\  
ERASE %1
```

My procedure is to first get into a subdirectory with the CHDIR command. Let's say that I wanted to move DEBUG.COM from the PROG subdirectory to the SYSTEM subdirectory as shown in Figure 2. I would then enter the command: MOVEIT DEBUG.COM SYSTEM. As you can see, the %1 in the batch file represents the file name in the current working directory. The %2 represents the subdirectory name where the file will be copied. Since I use single level subdirectory names, I have enclosed the %2 parameter with backslashes.

Now that you know about subdirectories, you will be able to organize files more efficiently. The PATH command should be useful to almost every user of an MS-DOS (or PC-DOS) system. Additional information on all of the commands discussed here is available in my Zenith/Heath MS-DOS (Z-DOS) FlipFast book and in the standard documentation provided with the operating system.

More On The Software Registration Process

It appears that there is still quite a bit of misunderstanding about how the Zenith Software Registration works particularly with regard to software updates. That was specifically brought to my attention in the January 1986 issue of H-SCOOP when a user complained about not receiving a software update. Part of that problem was probably my fault since I have mentioned in several articles that I received an MS-DOS update to version 2.21 from version 2.11. I never explained exactly why I received that update, and I have received a few letters over the past months about users not receiving a 2.21 update. Because of the number of letters that I received on the subject, I had already written a detailed discussion on the update process. Part of that information was provided to H-SCOOP to answer that user's complaint. Here is the complete story.

There is a lot of misunderstanding about Zenith software updates and how they work. I have mentioned in this column that I received an update to my version 2 MS-DOS (as well as Z-DOS), and since virtually everyone seems to have a Zenith MS-DOS operating system for either the Z-100 or the Z-150, I'll use that as an example.

Pull out your Zenith MS-DOS distribution disks and look at the last two numbers in the serial number: typically 01, 02, 03 or 04. That is basically the REVISION NUMBER of that software. My original version 2 MS-DOS for the Z-100 had a suffix of 01 — the first production run of that software. When I received an update, the serial number was the same except that the serial number had the suffix of 04. These updates may be the result of a change to IO.SYS (the BIOS), MSDOS.SYS, and/or COMMAND.COM plus any affected programs supplied on the distribution disk. Version changes for the system can easily be seen with the VER command which displays the version number for the BIOS and MS-DOS. Changes to COMMAND.COM are easily recognized because of the date displayed in the disk directory (DIR command).

The point is that people are notified of updates based on the revision number of their software. Everyone who had a "01" version of the Z-100 MS-DOS should have received an update. Obviously, people who had the "04" revision of MS-DOS did not. I suspect that these particular corrections may have also been made prior to the release of the "02" and "03" revisions, but I don't know that for a fact. And I don't recall what the exact revision break was for people to receive the free update. For example, the later revisions of MS-DOS corrected a rather obscure bug that caused an occasional "loss" of a character at a serial port under very special (and not normally encountered) conditions. In general, I have the impression that Zenith will provide a free update to software that has bugs that could result in a loss of data.

The point is that you shouldn't EXPECT to receive free software updates unless you know what the "cutoff" revision number is. And if you have a question about whether you SHOULD get an update, I suggest that you call or write to Zenith Software Consultation. The phone number and the address are contained in all of the software manuals. Because that phone number seems to be busy most of the time, I suggest that you write a letter with the name, model number, and serial number of the software.

On the other hand, you can always obtain the most recent version of any Zenith software for a specific fee, usually less than \$50.00. Again, check with Software Consultation about the latest version and cost for the update. Registered users are normally notified of major enhancements to Zenith software. For example, I received an update notice for Z-100 Multiplan at a cost of \$35.00. In addition to new distribution disks, I expect that I will also get updated documentation which is the reason for the higher cost than you might expect. I remember a couple of years ago when the H-89 CP/M was upgraded to version 2.2.04. A number of users complained that the \$35.00 update cost did not include new 3-ring binders. That did not occur to me at the time since I removed the old documentation from the existing binder and inserted the new. Alas, I guess it is just not possible to please everyone.

In that same issue of H-SCOOP, this user reported he called Zenith Software Registration and was told that they are not using computers. He must have misunderstood the comment. Indeed, Zenith does use an IBM mainframe computer, physically located

in Chicago for software registrations. Unfortunately, it is a rather archaic transaction driven batch system (in my opinion) which does not have interactive on-line capabilities.

Listing 1
Maximum ROOT Directory Entries For Disk Formats

Disk Type	Format	Maximum ROOT Directory Entries
5.25"	SS/DD	64
	DS/DD	112
	DS/QD	144
8"	SS/SD	68 (Standard SB-86 Format)
	SS/SD	104 (Standard Zenith Format)
	DS/DD	192
Hard Disk	n/a	480 per partition

Zenith processes on the order of 30,000 plus software registrations a month. License information in the mainframe computer database is keyed by the model number and serial number of the programs as described in my July 1985 REMark column. As the license agreements are received, they are physically sorted this way to facilitate database updates. However, the database is not actually updated unless it is known that an "Update Notice" (like Multiplan or Word) or a disk update will be sent to appropriate users (based on the revision number) as previously described. Based on my 20 years of mainframe experience, I can tell you that the cost of input, storage, and maintenance of this kind of information is very expensive. My guess is that's the reason that information is added to the system on an "as needed" basis. This particular process may be the basis for the user's misunderstanding that "all of the software registration information is on file cards".

Based on the way this process works, my interpretation is that this user was simply not "scheduled" to receive an update. That is probably the real reason that he was told that the information was on file cards (the license agreement forms actually). What he should have done is called Software Consultation to find out if, based on the serial number of his software, he was actually scheduled to receive an update.

When the software registration database is to be updated, the information is sent to a "keypunch" operation (not an IBM 029 I hope). Computer generated output (update cards, disk labels, and/or mailing labels) are used for the mailing to REGISTERED owners. For those of you who missed the information in last July's article, I have included "Your Part In The Software Registration Process" in this column.

And that's how the software registration process works from a procedural point of view. License agreements are physically sorted by model and serial number when received in the mail. Some are "filed" if there is no scheduled update for that particular model number software. Others are batched, sent to keypunch, and used to update the software database on the mainframe computer. For that software which has been filed, and an update is determined to be required, it is sent to keypunch and used to update the database so that the appropriate labels can be printed. Records are maintained in the form of the source document (i.e. the license agreement form returned by the user), and output reports from the computer update which may be paper or microfiche.

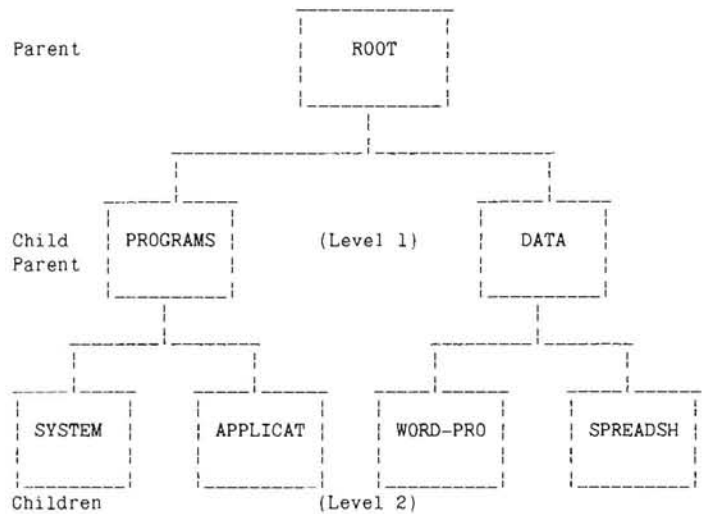
As I was wrapping up this article, I received the February 1986 issue of H-SCOOP (#71) which contains the epilogue to this

story. Part of this epilogue is that I wrote a lengthy letter to H-SCOOP because of the original user's problem.

H-SCOOP believes that the user in question should have received the update. Fine, I won't argue that point. I will stress again, however, that if you believe that you should have received an update, and have not received it yet, write a short letter to Software Consultation. They can advise you of the current status of software updates. Be sure to include the name, model number, and serial number of your distribution disk.

Lest you believe that the Software Registration process is totally a Zenith responsibility, I will repeat some of the information on software registration that was part of last July's article.

Figure 1
MS-DOS Directory Trees



Your Part In The Software Registration Process

Complaints about software updates are one of the business problems that all vendors have. I have talked to the people at Zenith Software Registration, and they have their share of problems, too. Many update-related complaints, when investigated, can be traced directly back to something that we, as users, did incorrectly or omitted.

One common problem is that a user may simply not have sent in the license agreement form. And if you haven't, there is obviously no way that you will receive any update. I've also been told that some forms are received without the name and address information. Same problem — you obviously will not get updates if they don't know where to send them. And for some of the forms that are filled in, the name and address is not legible, so it's impossible to determine the owner of the software.

Since Zenith processes thousands of software registrations per month, it's imperative that you do as much as possible to keep up your end of the bargain. I use this process for all of my license agreements, and you may find it useful, too.

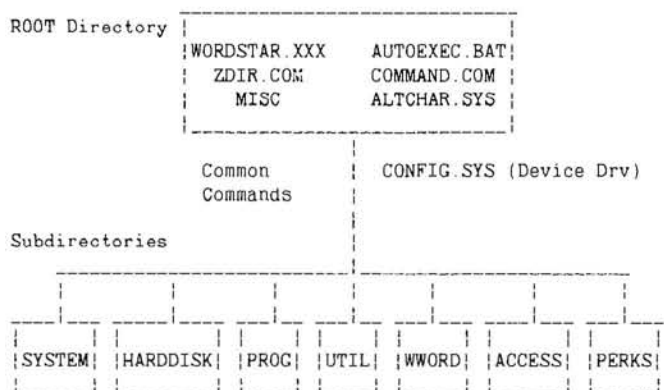
When I open the package, I keep all of the software license information together. That includes both copies of the license agreement (white and yellow) plus the envelope. I also remove the labels from the outside of the box so that I can use it directly on my copy of the license agreement.

Since my writing tends to be illegible, I always PRINT my name and address on both license forms along with the other required

information. Regardless of who you are — individual, business or military —, be SURE that you complete the name and address information legibly. After I've completed both forms, I take one of the labels that I previously removed from the box and tape it to my copy (the yellow one) of the license agreement. That's essentially an exact copy of the one on the Zenith copy (white) of the agreement which contains the name, model and serial numbers of the software. I've found that it's faster, and more accurate, to use the extra labels than to try to copy the information by hand. The white copy (and only the white one) gets mailed to Zenith in the supplied envelope, and the yellow copy goes in my "License Agreements" file for future reference.

Those of you who have purchased software recently will note that Zenith is making an effort to improve the software registration process by making it easier for us. In the newest software, only a registration postcard is provided which must be mailed to Zenith. The two copies of the license agreement are no longer provided. Since I still like to have a complete record of my software in a file folder, I use a pair of scissors to cut around the edge of the software envelope AFTER I have removed the distribution disks. I can then file the top part of the envelope with the software name, model, and serial number for future reference. But what can you do if you have a change of address like I did?

Figure 2
Setting Up Your Own Subdirectories



Software Registration — Change Of Address

The database for software registration is keyed by the model and serial numbers of your programs. If you have to contact Zenith about your software, for any reason, you must be prepared to furnish that information. And that includes a change of address. All you have to do is send the model and serial numbers of your software to Zenith along with your new address. I suggest that it's also a good idea to include your old address for reference if required. Don't just send in a "Change of Address" card . . . that isn't enough. You must include the model and serial numbers for each piece of software. For those of you who need to send in a change of address, I've included the Software Registration address at the end of the column.

There will inevitably be some people who think that this is too much work. Their argument will go along the lines that they should only have to send in a change of address card, and Zenith should do the rest of the work. Although there is some merit to that argument, you should already have a complete inventory of

your hardware and software, complete with vendor, model and serial numbers. If you haven't got a COMPLETE hardware and software inventory, lots of luck if you ever have to file an insurance claim . . . let alone the idea of getting an insurance policy to cover your computing assets in the first place. How much insurance should you get? As a quick test, write down everything you can think of that either is part of or supports your computer system. Did you remember to include all of those connecting cables for printers? What about those additional memory boards in your system?

Speeding Up Your System

My word count program just told me that I have already exceeded 7,000 words in this article, so I will make this short.

I have found a rather interesting way to improve system performance which is also quite inexpensive and easy to install. The product is the NEC V20 CPU chip which is pin-to-pin compatible with the 8088 CPU that is used in the H/Z-100s, the H/Z-150s, and the IBM PC. The V20 comes in two versions: the V20-5 for a 5 mhz clock speed, and the V20-8 for the 8 mhz clock speed. The V20-5 is so compatible with the 8088 that I have replaced it on the Gemini board with no problems. I still have the 5 mhz Gemini, so I can't change the clock speed. But, the performance improvement is quite noticeable in the video display and spreadsheet recalculation time. Although the V20 will not improve disk I/O, I expect that there will be a performance improvement in the 17-40% range based on what I have read. At this point, I am planning to do some benchmark tests to nail down the improvement in CPU throughput. More on that in a future article.

Although I can't tell you exactly what the performance improvement is in quantitative terms, you might find it advantageous to consider the V20. I have made arrangements with Don Murray at the local Heathkit store in Dallas to provide V20s to HUG members at reasonable prices.

There is a 5 mhz V20 (70105) which is available at \$19.95. If you have a 5 mhz system, you may want to order the V20-5 at \$19.95. But I suggest that you consider getting the 8 mhz V20 (70108) at \$24.95 since initial tests indicate that it runs fine at both the 5 and 8 mhz clock speeds. Owners of "combo" systems, like the '158s or '100s with speed-up kits, will probably be interested in these.

One other advantage for '150 owners . . . the V20 runs 8080 code. What that means to you is that you can, in some cases, run 8-bit programs intended for CP/M using some public domain "emulators". Some of the public domain software does not include a terminal configuration program which is the reason that I qualified that statement.

At this point, the V20-5 works perfectly, and I haven't found any problems. Even CP/M-85 performance is improved, probably because of the fact that video I/O is normally handled by the 8088. You can order either V20 from the Dallas Heathkit store by calling or writing to them at the address shown at the end of this article. Prices shown include \$1.75 for shipping and handling. Texas residents should add 5.125% sales tax, and if you live in Dallas county, it is 6.125% sales tax.

Next Month

Another one of the interesting features of MS-DOS version 2 and later is the capability for I/O redirection. We'll talk about the basics and how you can use I/O redirection next time unless I receive the UCI Easy PC in time for a review.

Time to wrap it up. I hope that the information presented in these articles has helped you, and I appreciate your letters of support for this kind of information. Although I have been writing articles for REMark for over two years now, I have received more letters in the last six months about this "general information series" than I have for all of the other articles combined. Thanks, I'm glad to hear what you like. I also appreciate your ideas for future articles, too.

I will be glad to answer questions about anything in this article, but be sure to include a self-addressed, stamped envelope if you wish to have a personal reply.

Products Discussed

H-Scoop (newsletter) 2618 Penn Circle Sheboygan, WI 53081	\$ 24.00 per year (12 issues)
MS-DOS Vers2 Programmers Utility Pack (CB-5063-16)	\$150.00
Easy PC Emulator (PC-250)	\$699.00
Z150 MS-DOS (OS-63-31)	\$ 90.00 (when purchased with PC-250 only)
Gemini Emulator Board (PC-251)	\$599.00
Z150 MS-DOS (OS-63-31)	\$ 90.00 (when purchased with PC-251 only)
MS-DOS Version 2 Z-100 only (OS-61-8)	\$150.00
Z-150 only (OS-63-50)	\$150.00
MS-DOS Version 3.1 Z-100 only (OS-63-30)	\$150.00
Z-150 only (OS-63-31)	\$150.00
Z-100 CP/M-85 (OS-53-2)	\$150.00
Disk Diagnostics Z-100 only (CB-463-13)	\$ 79.00
Z-150 only (CB-5063-28)	\$ 79.00
SM-77 Digital Voltmeter	\$119.95
Heath/Zenith Computer Centers Heath Company Parts Department Hilltop Road St. Joseph, MI 49085 (616) 982-3571 (HUG Software only) (800) 253-7057 (Heath Catalog orders only)	
Software Registration Heath/Zenith Data Systems Corporation P.O. Box 1000 St. Joseph, MI 49085	
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)	
FlipFast Command Guides CP/M-80/85	\$ 12.95
MS-DOS (Z-DOS)	\$ 19.95
Heath/Zenith Computer Centers S-A Design Books 515 W. Lambert, Bldg. E Brea, CA 92621-3991 (714) 529-7999	

Perks MS-DOS (Z-100 only)	\$ 69.96
CP/M Plus (Z-100 only)	\$250.00
Heathkit Stores (Perks only) Barry Watzman 560 Sunset Road Benton Harbor, MI 49022 (616) 925-3136	
Sextant (Magazine - 6 issues)	\$ 14.97/year
BUSS (Newsletter - 20 issues)	\$ 28.00/year
Sextant Publishing Company 716 E Street S.E. Washington, DC 20003 (202) 544-0900	
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	
Z100 Speed Module (Z-100 Only)	\$ 49.95
Heath/Zenith Computer Centers CDR Systems, Inc. 7210 Clairemont Mesa Blvd. San Diego, CA 92111 (619) 560-1272	
8.0 Mhz Speed-Up Kit (Z-100 Only)	\$ 49.95
Heath/Zenith Computer Centers UCI Corporation 948 Cherry Street Kent, OH 44240 (800) 824-2667 (Outside Ohio) (216) 673-5155 (In Ohio)	
V20-5 (70105) CPU (5 mhz)	\$ 21.70 (\$19.95 + 1.75 S&H)
V20-8 (70108) CPU (8 mhz)	\$ 26.70 (\$24.95 + 1.75 S&H)
Mr. Don Murray Heath/Zenith Computer Center 2715 Ross Avenue Dallas, TX 75201 (214) 826-4053 (VISA and Master Card accepted)	



OOPS!

The following is a correction to step #3 in the header of the 'Change Speed' program listing found on page 50 of the March 1986 issue of REMark.

3. Conver EXE file to COM. (EXE2BIN CS.EXE CS.COM)

If the extents are not specified as in the original instruction, a BIN file will be created instead of a COM file.



Controlled Data Recording Systems Inc.

Quality Products and Support for the Heath/Zenith Community

New for the H/Z89-90 Computer Users: The Super RAM 89 Package

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

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

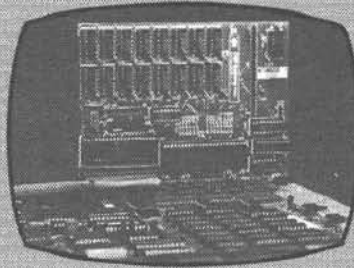
Now get RAM 89 **HDOS DRIVER** Software Only **\$35**

RAM 89 DVD supports RAM SY0: and SY1:, added auto-load features, plus more!

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

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

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



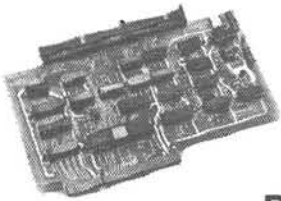
Ram 89 comes as a two card set that plugs into the left hand (16k expansion) side of the computer. No computer modifications required.

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

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

FOR THE H/Z89-90 COMPUTERS

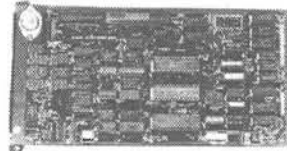
THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE: \$395



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

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

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



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

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

MODIFY 89

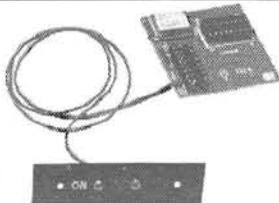
Allows owners of the FDC-880H or the FDC-H8 to read/write to the following disk types:

Access S.S.D.D., Cromemco S.S.D.D., DEC VT180 S.S.D.D., IBM PC/Zenith 100, (CP/M) D.S.D.D.*, Kaypro II S.S.D.D., Morrow Micro Decisions S.S.D.D., NEC PC-8001 A S.S.D.D., Osborne S.S.D.D., Osborne S.S.D.D., Otrona D.S.D.D.*, Zorba D.S.D.D., TI Professional S.S.D.D., TRS-80 Model I (Omnikron CP/M), TRS-80 Model III (MM CP/M), Xerox 820 S.S.D.D., Xerox 820-II S.S.D.D. Plus more, Standard (Tests for H/Z37 and C.D.R. Disk types)

PRICE \$49.95

FOR THE Z100 COMPUTERS

THE ORIGINAL Z100 SPEED MODULE. RUN YOUR Z100 PROGRAMS FASTER



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

SUPER DRIVES 85

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

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

ONLY \$695 LIST
Dealer inquiries invited.



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

A Report On The NEC V20 Microprocessor Chip For The "Z" Machines Part I

Richard L. Mueller, Ph.D.

11890-65th Avenue North
Maple Grove, MN 55369

Introduction

Some time last Fall, articles in magazines and notes on some of the Bulletin Boards started to talk about the replacement microprocessor chip for the 8088 chip from NEC. This chip is the V20, one in a series of advanced microprocessor chips in the NEC "V" series. For those micros that may be using the 8086 chip, the V30 is the replacement for that one.

In this article, I would like to relate my experiences in the short time that I have had the V20 chip and in a second article give a brief overview/background/description of the V20 itself with a mention of other "V" chips coming in the future. The information that I have seen on the V20 state that one can get anywhere from 5% to 100% or more improvement in execution of programs. Well, needless to say, I decided to get one this past December and test it out on both my Z-160 and H-100.

The V20 microprocessor chip comes in either a 5 MHz (uPD70108-5) version or an 8 MHz (uPD70108-8) version. In one of the notes that I saw recently, it talked about a 10 MHz version. However, I did not see that in the preliminary information that I received from NEC. Since I have not upgraded my H-100 to run at the 8 MHz speed, which means both of my machines run at 4.77 MHz, I ordered the 5 MHz version.

Benchmark Background

While waiting for my V20 chip to arrive (I ordered it from a hardware/software mail-order house), I decided to write a test program (my "benchmark") in assembly language and use MS-DOS to time a variety of instruction sequences. First, I would run my benchmark program on both machines using the 8088 chip that came with my machines, then rerun the program after installing the V20 chip. My intention was not to get an absolute pre-

cise timing of each instruction (although one could certainly do that with a lot of work), but basically to compare the timings of the various instruction sequences or operations of the two chips. I wanted to get a feeling on where the V20 chip was faster and where it was about the same as the 8088 microprocessor chip.

Before discussing the actual instruction sequences or operations that were used, I first want to describe how the timing was done. Anytime one wants to 'time' a particular operation, a time stamp must be taken immediately before the operation to be tested, and again a time stamp must be taken immediately upon completion of the operation. If precise timing is a requirement of your benchmark, then using the Operating System for getting the "times" may not be accurate enough. Some "overhead" would be introduced into your results.

In my case, I was not concerned with precise timings. I just wanted to get a "ballpark" comparison of several operations using the two microprocessor chips. Using the MS-DOS "GET-TIME" function was just fine for me. One must remember, this MS-DOS time function only returns the time to the nearest hundredth of a second, which was fine for my testing.

For curiosity's sake, I called the MS-DOS "GETTIME" function twice in succession, saved the results from the first reading and compared them with the second reading. To my surprise, the readings were the same which tells me that the Operating System takes less than one hundredth of a second to process the time request. This does not mean there is no overhead involved with getting the time, it just means that the overhead is less than one hundredth of a second. Anyway, reading the time via the MS-DOS Operating System was negligible for my benchmark.

Just as the MS-DOS time function took what appeared to be "no time at all", since the time returned from the function itself is

only carried to the nearest hundredth of a second, I expected my instruction sequences or operations to also take "no time" to execute. That's exactly what happened. To get around this situation, an instruction sequence or operation must be repeated a very large number of times to make sure the elapsed time is much greater than one hundredth of a second. I just used 1,000 or 1,000,000 depending on the operation; more on the operations in the next section.

List Of Operations Tested

The following is a list of the various instruction sequences or operations that were used in this comparison study. The results of the testing are contained in the next section. Although each instruction sequence or operation is intended to test out a particular instruction as its primary purpose, it takes many other instructions to accomplish the task. For example, a loop is needed to control the number of times a particular operation is executed. This means setting up counters, decrementing them, branching depending on the status of the counters, etc. So each instruction sequence or operation is more than just testing one instruction. But that's okay since the purpose of this exercise was just to compare various operations, not individual instructions.

1. 1000-Byte-Clear Test (1000 times). This sequence/operation clears (sets to zero) 1000 "Bytes" of memory using the 'REP STOSB' instruction. This 1000-byte buffer clearing operation is repeated 1000 times. Look at the results in the next section.
2. 1000-Word-Clear Test (1000 times). This sequence/operation is the same as the preceding test except that it clears 1000 "Words" of memory using the 'REP STOSW' instruction. This operation is also executed 1000 times to get a meaningful result.
3. 1000-Byte-Load Test (1000 times). This sequence/operation uses the 'REP LODSB' instruction. This sequence is not very meaningful as far as a production program is concerned, since it loads the same byte into the AL register 1000 times. However, from a benchmarking point of view, the operation makes sense. The 1000-Byte-Load test is executed 1000 times.
4. 1000-Word-Load Test (1000 times). This is the same as the preceding test except that a "Word" is loaded into the AX register instead of a "Byte".
5. 1000-Byte-Move Test (1000 times). Just as the name implies; 1000 "Bytes" are moved from one section of memory to another section of memory using the 'REP MOVSB' instruction. This Move operation is repeated 1000 times.
6. 1000-Word-Move Test (1000 times). This test is the same as the Byte Test above except "Words" are moved using the 'REP MOVSW' instruction.
7. Byte-Add Test (1,000,000 adds). Although this test would not be meaningful in a production program, it does, however, provide a good test for the ADD instruction. This test simply executes the following ADD instruction 1,000,000 times along with the accompanying loop instructions: 'ADD AL,BL'.
8. Word-Add Test (1,000,000 adds). Same as the previous test except that "Words" are added together instead of "Bytes": 'ADD AX,BX'.
9. Byte-Subtract Test (1,000,000 subtractions). This test simply subtracts one register from another 1,000,000 times using the following: 'SUB AL,BL'. Similar to the ADD tests.
10. Word-Subtract Test (1,000,000 subtractions). Same as above except "Words" are subtracted instead of "Bytes": 'SUB AX,BX'.
11. Byte-Divide Test (1,000,000 divides). Basically the same as the other arithmetic tests above: one word register is divided by a byte register. This operation is repeated 1,000,000 times using the following sequence: 'MOV AX,n' 'DIV BL' where n is any unsigned integer, and likewise, BL contains an unsigned integer. 'DIV' performs "unsigned" divisions.
12. Word-Divide Test (1,000,000 divides). Same as the preceding test except that this operation involves dividing a word register pair by a word register: 'MOV AX,n' 'MOV DX,m' 'DIV BX', where m is the most significant portion of a 32-bit unsigned integer and n is the least significant portion of the 32-bit unsigned integer. BX contains a 16-bit unsigned integer. Again, 'DIV' performs "unsigned" divisions.
13. Byte-Integer-Divide Test (1,000,000 divides). This is the same as the 'DIV' tests except that signed integers are used by the 'IDIV' instruction (i.e., signed divisions).
14. Word-Integer-Divide Test (1,000,000 divides). Same as the preceding tests except that a 32-bit signed integer is divided by a 16-bit signed integer. The DX AX register pair is divided by the BX register.
15. Byte-Multiply Test (1,000,000 multiplications). In this operation, the 8-bit unsigned integer in the AL register is multiplied by the unsigned integer in the BL register 1,000,000 times. This operation uses the 'MUL' instruction along with the accompanying loop instructions.
16. Word-Multiply Test (1,000,000 multiplications). Same as the Byte-Multiply test above except that the 16-bit unsigned integer in the AX register is multiplied by the 16-bit unsigned integer in the BX register.
17. Byte-Integer-Multiply Test (1,000,000 multiplications). This is the same as the Byte-Multiply test above except that the 8-bit signed integer in the AL register is multiplied by the signed integer in the BL register (i.e., signed multiplication operations). The 'IMUL' instruction was used here.
18. Word-Integer-Multiply Test (1,000,000 multiplications). Same as the preceding Byte-Integer test except that the 16-bit signed integer in the AX register is multiplied by the signed integer in the BX register.
19. Write 640 512-Byte Blocks Test. This operation writes 640 blocks of 512-bytes long on a floppy disk that was formatted as 9-sector per track disk. In other words, 320K of information was written on a 360K capacity disk. The purpose of this test was to see if the NEC V20 chip had any affect on floppy disk operations. See the results in the next section.
20. Read 640 512-Byte Blocks Test. This operation is similar to the preceding test except the blocks that were written above are now read. Nothing is done with the data that is read; each block read overwrites the previous block read into memory. Again, the purpose of this operation is just to see if the V20 chip has any affect on floppy disk reads.

Benchmark Results

Test	Z-160 (secs)		H-100 (secs)	
	8088	V20	8088	V20
1. Byte-Clear	2.14	.88	2.16	.90
2. Word-Clear	3.07	1.87	3.06	1.79
3. Byte-Load	2.80	1.93	2.87	1.95
4. Word-Load	3.79	2.91	3.70	2.86
5. Byte-Move	3.79	1.81	3.69	1.79
6. Word-Move	5.49	3.62	5.48	3.55
7. Byte-Add	6.76	6.26	6.56	6.11
8. Word-Add	6.70	6.26	6.56	6.11
9. Byte-Subtract	6.75	6.26	6.56	6.11
10. Word-Subtract	6.70	6.26	6.56	6.10
11. Byte-Divide (unsigned)	25.71	11.37	25.74	11.26
12. Word-Divide (unsigned)	42.02	15.10	42.31	15.19
13. Byte-Divide (signed)	30.26	15.16	30.50	15.09
14. Word-Divide (signed)	45.43	19.22	46.68	19.20
15. Byte-Multiply (unsigned)	22.69	10.82	22.94	10.79
16. Word-Multiply (unsigned)	34.05	13.24	34.24	13.19
17. Byte-Multiply (signed)	25.21	12.97	25.13	12.93
18. Word-Multiply (signed)	36.03	15.10	36.46	15.35
19. Floppy Disk Write	142.48	142.47	135.06	135.06
20. Floppy Disk Read	142.70	142.69	135.23	135.24

Conclusions

Except for the floppy disk operation tests, all the other tests showed an improvement, some very significantly, when the NEC V20 microprocessor chip was used. The reason why the floppy disk operations remained the same is that the operations were very dependent on the physical disk drives themselves, rather than on any specific instruction or set of instructions. The differences between the Z-160 runs and the H-100 runs are due to different physical disk drives, and different BIOS/Disk Drivers being used.

As you can see from the results, the Divide and Multiply operations showed tremendous improvement with the NEC V20 chip,

while other operations, such as the Add and Subtract operations, showed only very little improvement. The Load, Store, and Move operations also showed very significant improvement with the NEC V20 chip.

What all this means is that depending on the application that is being used, you could see little or no improvement over the 8088 chip or see a significant improvement. Those applications which depend heavily on disk activity will most likely show no improvement. However, those applications which have very little disk activity and have very high usage of the Multiply and Divide instructions, along with Loads, Stores, and Moves, will show a significant improvement. If your spreadsheet application has many formulas that perform multiplications and divisions, you will notice an improvement.

The application being used with its unique activity will determine whether there is any improvement or not with the NEC V20 chip. In most cases, you won't notice an improvement but it will be there. As I said earlier, I used the 5 MHz version of the V20. If you have upgraded your Z-Machine to run at 8 MHz, you have undoubtedly seen an improvement just running your 8088 chip at that speed. Adding the 8 MHz V20 chip should give you yet another improvement.

The purpose of this article was not to promote or sell you the NEC V20 microprocessor chip, but to give you some idea of some of the differences between it and the 8088 chip in terms of execution speed. The choice is yours. However, I can tell you that I am sold and I went out and purchased a second V20 chip so I could have a V20 chip in each of my Z-Machines. In my second article on the NEC V20 chip, I will discuss the additional instructions, the enhanced instructions, and the 8080 emulation mode that are all part of the NEC V20 (or V30) chip. I did write a test program that does switch between 8088 native mode and the 8080 emulation mode to perform a task, and I will discuss this in my next article. *

Serious Business Applications?

CCT,* the Nation's leading S-100 OEM, now offers products and support for the Z100 system. We have compatible 256K and 512K static memory boards, and the CCT Printerfacer 1™ The Printerfacer 1 is an intelligent I/O board with 256K of on-board memory, expandable to 1 megabyte, for printer/plotter intensive application

*Increases system speed and throughput tremendously!
Runs two devices simultaneously!
Fast!*

buffering. The board comes standard with 1 serial and

1 parallel port, residing at the same physical address, which are software and keyboard selectable. Increases system speed and throughput tremendously! Runs two devices simultaneously! Fast! Printerfacer 1 with cable set and CCT 12 month warranty.

Liberal Dealer Package Available—Inquire with CCT Marketing.

Zenith Data Systems

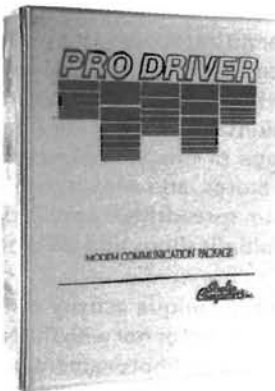
Z-138 PC Transportable CALL Z-148 PC Desktop CALL Z-158 PC Desktop CALL
Z-171 Lap Top Portable..... CALL Z-200 PC-AT System..... CALL

Hard disk upgrade kits are available for the Zenith System

CCT Implements Tomorrow's Technology Today™

CCT and Printerfacer 1 are trademarks of CCT.

CCT / 1 CCT Plaza • Box 4160 / West Sedona, AZ 86340 / 800-222-8686 / In AZ: 602-282-6299



SOURCE CODE

Studio Computer's popular communication program *Pro Driver*, is now available in source code form!

Pro Driver is a cleanly designed menu-driven MS-DOS modem communications program. It supports 3 popular transfer protocols as well as VT-52 & VT-100

emulation. Up to 32 auto-logon sequences and a usage reporter option lets you track all calls made. Full color selection modes as well as baud rate and all other pertinent options are settable through *Pro Driver*'s easy to use menus.

Written in the popular 'C' programming language, our source code package consists of two standard 5-1/4" soft-sectored floppy disks that contain almost a half megabyte of commented 'C' code.

We provide the complete source including our I/O device drivers for the Z100, Z100 PC and IBM PC compatible computers.

Registered owners of any release of *Pro Driver* may purchase the source for only **\$99** complete! If you are not a current owner of *Pro Driver*, the entire package of both the executable version along with a complete 200 page manual and the source code is just **\$148**. Of course we still offer *Pro Driver* in the standard 'executable only' version for just **\$49** (specify Z100 or Z100 PC).

Requirements needed to properly compile our source without modifications, are a disk drive with 1 megabyte capacity, the Computer Innovations C86 'C' compiler, Microsoft MASM and an overlay linker such as Phoenix's Plink-86.

For your *Pro Driver* source license, send check, money order or use your Visa/Mastercard. Please include \$3 for shipping and handling.

* Studio Computers cannot offer support on any modifications made to the source code. Compiled code generated from the source code may be used for non-commercial use only.

SPECIAL LIMITED TIME OFFER!

Need a good solid dependable 300/1200 baud modem? We've got them for only **\$169** from Practical Peripherals. This half-card IBM PC compatible cards plug directly into your Zenith computer. It even comes with its own software. And, it's backed by an incredible 5 year warranty.

We also have the popular Dr. Halo graphics software package. Use it to create mechanical sketches or draw in full color using one of its many icon driven commands. Then print your creations on your graphics printer. Our special price this month for the package is just **\$89**.



We have a super value this month on our Famous Maker 48 track double-sided, double-density soft-sectored diskettes. For only **\$9.88** per box of ten, you'll receive 100% certified floppy disks that are backed by a money back warranty and a free box of name brand disks if

not satisfied. Each box includes sleeves, labels and write-protect tabs. You just can't go wrong! Plus... for every 10 boxes you order, we'll throw in an extra box of ten disks *free!*

Please add 2% of the total order (\$3 minimum) for shipping and handling.



999 South Adams
Birmingham, AL 35201
Phone (313)-645-5365



Send for our complete 1986 catalog with super prices on Zenith floppy and hard disk computer systems and accessories.





Burn Baby, Burn!

A Review Of The ID-4801 EPROM Programmer

Jim Buszkiewicz
HUG Software Developer

A short while ago, Heath introduced a new kit which, I feel, had been long overdue. The kit I'm referring to is the ID-4801 EPROM Programmer. With this device, the digital enthusiast has the ability to explore every aspect of Erasable Programmable Read Only Memory ICs.

The heart of this instrument is a single large circuit board containing several dozen ICs, including an equal number of resistors, capacitors, switches, etc. Performing all the controlling functions on this board is a Z-80 microprocessor, running at 4 MHz. The kit can be completed in a single evening (even if Heath doesn't think so), especially if your evenings last between 3 and 5 hours. Most of the construction consists of inserting and soldering parts onto that large circuit board. More time consuming and tedious, is the construction of the 'personality' modules. These little modules determine which EPROM can be programmed by the unit. The programmer initially comes with 4k of RAM, but can be expanded to a total of 16k. The unit must contain as much or more RAM than the capacity of the EPROM being worked with. Therefore, in order to read or program a 2764 EPROM (8k X 8), the programmer must have at least 8k of RAM. EPROMs as large as 27128 can be read and programmed with the ID-4801.

The ID-4801 programmer is capable of 10 separate and distinct functions, some of which require prewired personality modules, as mentioned previously. These functions are described as follows:

Keypad mode allows you to enter or examine data in any RAM memory location. Memory addresses can be incremented or decremented. Data can also be entered repeatedly for duplicating entire blocks of data.

EPROM to RAM mode transfers data from a preprogrammed EPROM into RAM. Using this feature and the 'Program EPROM' mode, EPROMs can easily be copied.

RS-232 transmit and RS-232 receive modes allow a user to transfer data either to or from RAM. This is accomplished by way of an RS-232 link between the ID-4801 programmer and host computer.

Data transfers are done in ASCII Intel Hex format at 9600 baud. No type of handshaking is done, so the modem program being used must be capable of accepting data at that speed. Most modem communication packages will not work at that speed without any handshaking due to small input buffers. Presently, HUG is selling a package that will work properly with this programmer at full speed. This package is called HUGMCP (HUG P/N 885-3033-37), and sells for \$40.00.

Simulate mode allows the programmer to act as the EPROM itself. This allows the user to actually test and debug firmware before the EPROM is programmed.

The insert and delete RAM data modes allow the programmer to insert or delete data bytes from RAM. Doing so, causes the rest of the RAM data to move up or down in memory depending whether a byte was added or removed.

The byte search mode is used to locate data bytes in RAM and display both the data and the memory address. This mode can be used to locate either one or two data bytes.

The program mode does the actual programming of the EPROM. Programming data is taken from the RAM, starting at any address, and programmed into the EPROM, again, starting with any address, for any specified length. Before doing the actual programming, the programmer verifies that the EPROM is erased, and during programming verifies that each memory location is correctly programmed.

The final mode is the verify mode. This mode allows you to load a program into the programmer's RAM and compare it to a programmed EPROM. The programmer's RAM can be loaded from the Keypad, RS-232 port, or another EPROM.

Only 8 years ago did such an instrument cost several thousand dollars (I know, I priced them!). But today, Heath has made it possible for almost every computer enthusiast to own their own microprocessor based EPROM programmer. The present selling price, only \$369.95.



the fourth law of robotics

A robot shall make learning fun for man and thereby improve the quality of life for mankind.



A robot is a robot is a robot... was a robot. Until HERO 2000.

HERO 2000 is much more than a robot. It's a walking, talking 16-bit computer. With 64K ROM and 24K RAM expandable to more than half a megabyte. And a fully articulated arm with five axes of motion. Yours to program. Command. Modify and expand. Total system access and solderless experimenter boards provide almost limitless possibilities. Its remote RF console with ASCII keyboard gives total control. Available with three self-study courses. Backed by Heath Company, world leader in electronic kits.

Build your own HERO 2000. Or buy it assembled. Have fun learning skills that translate directly to the world of work.



HERO[®] 2000

the
knowledge
builder

FREE. Send today for latest Heathkit Catalog.



Mail coupon today to receive a **FREE** Heathkit Catalog featuring HERO 2000.

Mail to: Heath Company
Dept. 063-404
Benton Harbor, Michigan 49022

Name _____

Address _____

City _____ State _____

Zip _____

RO-141

Heathkit[®]
Heath
Company

A subsidiary of Zenith Electronics Corporation

The WordStar Directory Changer

Joseph Katz

*103 South Edisto Avenue
Columbia, SC 29205*

A notable limitation of WordStar and other MS-DOS translations from CP/M programs is that they do not recognize subdirectories. This wonderful addition to MS-DOS appeared first in version 2.0 and remains a part of version 3.0. It's a wonderful feature because subdirectories are the way to bring order to a big hard disk. They let you organize the hard disk into a treelike structure, with the root directory branching into a subdirectory for each kind of work you do on the computer. If you're a writer, for example, you could make a subdirectory for each project and one for correspondence with each publisher of your work. Then, when you want to turn from work on a chapter in *The Great American Novel* to a letter explaining why you can't meet the deadline for it, all you do is switch subdirectories. You get the idea.

Unfortunately, WordStar — and programs like it — doesn't, no matter what you try.

Some people try WordStar's "change logged disk" option, from either the main menu or within a document, figuring they can respond to the prompt with a full pathname instead of just the new drive identifier. They learn that the buffer here can hold only two characters: a legal drive letter followed by the trailing colon. You can't change subdirectories that way.

Other people try WordStar's "Run a Program" option on the main menu to execute the MS-DOS resident commands "CD" or "CHDIR." After all, either of those two commands will change directories from the MS-DOS command line. But neither will work when called from the "R" option on WordStar's main menu. Its restriction comes not from a limited buffer: the buffer will accept almost any combination of alphanumeric up to sixty-five characters long. The trouble is that only an "EXE" or "COM" program can be run from the main menu, so all MS-DOS integral

commands are inaccessible. It's, therefore, impossible to access them from a "BAT" file, because those are ignored, too. You can't change directories that way either.

The reason for WordStar's deficiency is that all MS-DOS incarnations of this once-dominant word processing program are really reincarnations of its CP/M versions. They, in turn, descended from ED.COM, the line editor that also is ancestor of MS-DOS's EDLIN.COM. All these programs were written to be generic, to strive for precisely the same behavior and appearance across the widest possible range of computer systems. Since CP/M has no equivalent of subdirectories, WordStar has no code to deal with them.

History, philosophy, and genealogy aside, it is a nuisance to use WordStar on a Winchester with more than one directory containing files that need editing in the same session. One must leave the program, change paths with the resident MS-DOS resident CD (or CHDIR) from the MS-DOS command line, then rerun WordStar. Of course, any special setups — spacing, margins, or tabs, for example — must then be reestablished, so a long session with WordStar often turns into a series of discontinuous editorial bursts. They are unnecessarily tedious.

And that's one big reason why WordStar has declined in popularity, giving way to programs that take better advantage of MS-DOS's capabilities.

Now that we know why we can't change subdirectories from WordStar, it's relatively easy to figure the way to do so. The trick is to think deviously. WordStar will run only a transient program — a "COM" or "EXE" program — not a resident program or a batch file. All one need do, therefore, is create a small transient program that feeds the new pathname to Function 3BH of INT 21H —

which is called by the resident CD — and Zoom: there's the new directory.

Meet CD.EXE, an assembly language program that allows changing directories from WordStar's main menu. Choose the main menu's "R" option to run a program. Then run CD, without a pathname: CD.EXE prompts for the "NEW PATH." Give it any pathname up to forty-nine characters long and press RETURN. (To use longer pathnames, adjust the first and last equates at the label "BUFFER" in CD.ASM.) If path and pathname both were valid, the directory is changed. If one, the other, or both were invalid, there is no directory change. In that respect, CD.EXE behaves just like the MS-DOS residents.

Where it differs is in its civilized responses to the user. Since resident commands, by definition, always occupy RAM space, systems programmers tend to keep them slim and functional. That way the residents occupy less space.

So resident commands expect complete, and completely accurate, commands on the command line: the MS-DOS resident CD requires the new pathname when it is invoked. It does not prompt for what it needs. But CD.EXE does. It will not accept a pathname as part of the command: it prompts for one instead.

And the MS-DOS resident CD stays mute after the command, whether or not the change succeeded. Many users, therefore, develop the good habit of issuing the resident CD command twice: once to command the change, once again to check if it was made. But CD.EXE reports either a successful change or a failure.

For that reason, you may like CD.EXE better than the resident commands and want to use it on the command line, as well as from WordStar. It can be done, if two minor adjustments are made. First, give CD.EXE some other name to avoid conflict with the resident command. There is no conflict when using CD.EXE from within WordStar for just the reason that a transient program is required in the first place: WordStar does not access resident commands, so the command "CD" from within WordStar will always use CD.EXE. But residents have priority in the MS-DOS hierarchy of command execution, so you will have to change the name of the transient to something like "CDD.EXE." Otherwise, when you issue the command "CD" from the command line, you always will be using the resident.

Second, make sure to put the renamed CD.EXE in a directory from which MS-DOS can find it through the resident Path command. On a well-ordered system, the likeliest best places are the root directory of the boot partition or the same directory as WordStar itself.

CD.EXE will run equally well on any computer that uses MS-DOS 2 or 3 — the generic operating system, as well as PC-DOS. It uses only standard INT 21H functions: 09H to display the prompt and other messages; 0AH to get the pathname; and 3BH to change directories. There is nothing fancy about this program. It merely works.

```

LINK WILL GIVE A 'NO STACK SEGMENT' ERROR   IGNORE IT
#

DATA SEGMENT
CR      EQU      0DH
LF      EQU      0AH
CRLF    DB      CR,LF,'$'
BUFFER  DB      49          , MAXIMUM PATHNAME,
                          , WHICH MAY BE ENLARGED
                          , BYTES ACTUALLY USED
                          , IN NAME
        DB      ?          , BUFFER NEEDED--ONE BYTE
                          , BIGGER THAN THE
                          , MAXIMUM PATHNAME
PROMPT  DB      'NEW PATH: $' , PATHNAME PROMPT
NO      DB      'PATH UNCHANGED$' , ERROR MESSAGE
YES     DB      'PATH CHANGED$' , SUCCESS MESSAGE
DATA ENDS

CODE SEGMENT
MAIN    PROC     FAR
        ASSUME  CS:CODE, DS:DATA
START:
        PUSH   DS          , SAVE FOR EXIT
        SUB    AX,AX       , ZERO IT
        PUSH   AX          , PUT IT ON THE STACK
        MOV    AX,DATA     , GET BUFFER
        MOV    DS,AX       , SHIFT IT

WORK:
        MOV    DX,OFFSET PROMPT , LOAD PROMPT
        MOV    AH,09H      , PRINT STRING FUNCTION
        INT    21H        , DO IT
        MOV    DX,OFFSET BUFFER , LOAD BUFFER ADDRESS
        MOV    AH,0AH      , KEYBOARD INPUT FUNCTION
        INT    21H        , DO IT
        ;COSMETIC PREPARATION FOR RESULT MESSAGE
        MOV    DX,OFFSET CRLF , LOAD IT
        MOV    AH,09H      , PRINT STRING FUNCTION
        INT    21H        , DO IT
        ;MAKE ASCIIZ STRING OF PATHNAME
        MOV    BX,BUFFER+1 , HOW MANY BYTES TO READ
        MOV    BH,0        , PUT IN BX
        MOV    [BUFFER+BX+2],0 , APPEND THE NULL FOR ASCIIZ
        ;REQUEST CHANGE
        MOV    DX,OFFSET BUFFER+2 , ADDRESS OF PATHNAME
        MOV    AL,0        , ACCESS CODE
        MOV    AH,3BH      , CHANGE DIRECTORY FUNCTION
        INT    21H        , DO IT
        JC     ERROR      , JUMP TO ERROR MESSAGE
        ;CHANGE SUCCEEDED
        MOV    DX,OFFSET YES , LOAD SUCCESS MESSAGE
        MOV    AH,09H      , PRINT STRING FUNCTION
        INT    21H        , DO IT
        RET             , BACK TO CALLER
        ;CHANGE FAILED

ERROR:
        MOV    DX,OFFSET NO , LOAD ERROR MESSAGE
        MOV    AH,09H      , PRINT STRING FUNCTION
        INT    21H        , DO IT
        RET             , DIE

MAIN    ENDP
CODE    ENDS
        END START
        END OF CD.ASM

```

Listing: CD.ASM

```

PAGE      55.80
TITLE     CD.EXE -- CHANGES MS-DOS 2 OR 3 DIRECTORIES
SUBTTTL   COPYRIGHT (C), 1986, BY JOSEPH KATZ
COMMENT #
PROGRAMMER: JOSEPH KATZ
          TO MAKE CD.EXE:
          1 ASSEMBLE WITH MASM "MASM CD;"
          2. LINK WITH LINK: "LINK CD;"

```

**Are you reading
a borrowed copy of REMark?
Subscribe now!**

G-PROM Review

(Or How To Increase H-89 Graphic Resolution For Under \$20!)

R. Kenneth Strum

*P.O. Box 35502
Brooks AFB, TX 78235
Copyright 1985*

Introduction

I made some improvements recently in my H-89A, so it would continue to meet my computer needs. I'll explain how I got higher resolution graphics out of my old war horse. Any modification I try has to really improve performance, be reliable and inexpensive. It has to be easy and quick to install.

My H-89A, prior to these modifications, was stock Heath. It was set up with 64K RAM, H-88-3 serial ports, H-17 disk drive controller, and three H-17 (hard sectored, 100K, old style) disk drives. My mechanical and electronic ability are somewhat limited, so "if I can do it, so can you" really does apply.

Evaluation Of G-PROM

I improved screen resolution by replacing the Heath character generator with G-PROM (\$19.95, including shipping, handling and complete documentation, available from NORCOM, 9630 Hayes, Overland Park, KS 66212). G-PROM is a character generator replacement for both the H-89 and H-19. Physically, it is a single ROM chip which replaces the Heath 24 pin DIP ROM at position U473 (H-19/H-89) or U420 (H-19A/H-89A) on the Terminal Logic Board (TLB).

G-PROM produces a new character set on the H-19/H-89 display. G-PROM will work under either/both the CP/M 2.2 and HDOS 2.0 operating systems without modification to any software. The new character set includes 23 revised ASCII and 24 graphic symbols, plus 81 unrevised characters. Photo 1 shows how the new graphic characters are more rectangular than the old Heath graphics. Note that in Photo 2 the shapes are square. This was a diagonal shape before. The new graphics have a two and one half times higher vertical resolution than the old set.

NORCOM has included an MBASIC program that demonstrates the new and old resolutions.

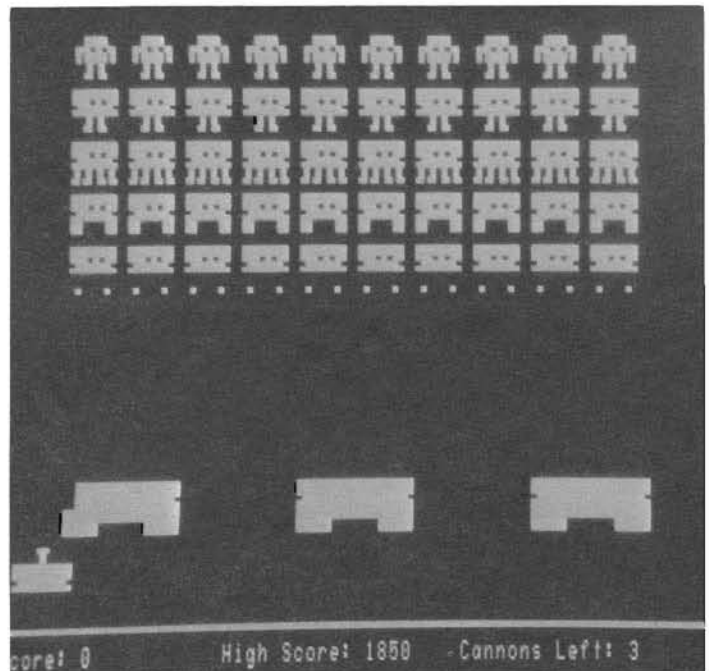


Photo 1
Demonstrates rectangular shapes to old "Invaders" game.

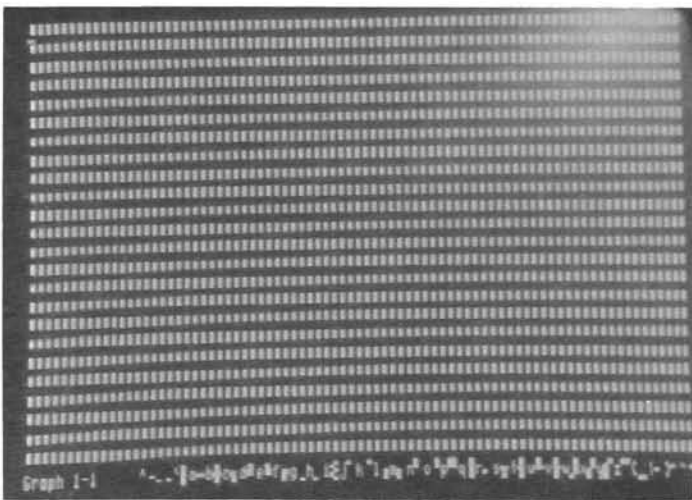


Photo 2
New rectangular shapes instead of diagonal shapes.

I was able to install G-PROM easily by following the directions supplied. Time was one hour, including the time for pulling the CPU and terminal logic boards out twice (I failed to label one connection since "I knew how it was oriented"). I inverted the plug on position P513 on the CPU board. I got only one "beep" when I powered up, and could not get the "H" prompt. However, the screen worked fine in the "off line" mode. If I had followed the directions supplied with G-PROM, which boldly stated to label each plug, I wouldn't have had a problem. I should add that it has been over a year since I "popped the top" on my H-89A. So, I was a little rusty, as you might guess. Photo 3 shows the G-PROM installed on the Terminal Logic Board.

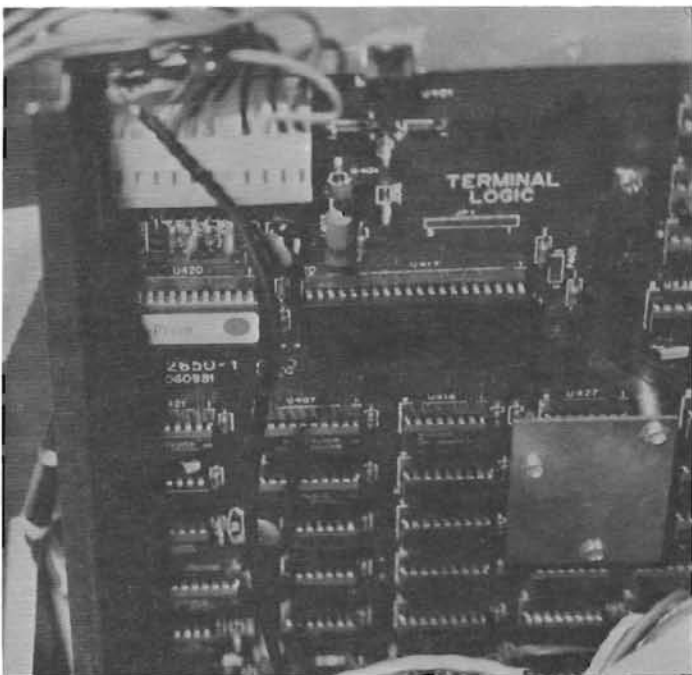


Photo 3
G-PROM installed on Terminal Logic Board of an H-89A.

Speaking of following directions, please pay special attention to the instructions for making sure the ROM's DIP pins are straight before you try to install the G-PROM ROM. Every DIP I've ever worked with has needed a little work to ensure the pins are straight up and down (perpendicular to the socket). Just roll the ROM's "legs" over a table top to straighten them. Be careful of static electricity. You can "zap" the ROM and ruin it if you don't ground yourself before picking it up.

Once the G-PROM was installed, I ran every program I had to see how each looked with the G-PROM character set. Most displayed just as before, only better. The only troublesome exception was Galactic Warrior. It looked "a little funny". It uses the "crossed lightning" graphic character which was replaced by a rectangular symbol. Shuttle Lander, marketed by Hoyle & Hoyle, and Invaders, marketed by The Software Toolworks, were affected to a lesser degree. See photo number two for an idea of the differences. All games played fine, though. Other programs that used graphics looked great. For example, if you use Pat Swayne's neat DIR19 program (HUG P/N 885-1126 title: HDOS Utilities by PS and the CP/M version, HUG P/N 885-1226), you will really like the full dividing lines. His SEE program (also on P/N 885-1126 and in CP/M, on a different HUG disk, P/N 885-1237), which uses graphics in the 25th line, looks fine. See Photo 3.

ED-A-SKETCH (a graphics editor), by the Software Toolworks, worked fine with G-PROM. Even the 25th line displayed the correct graphic characters with the corresponding keyboard elements. I enjoyed the increased resolution of the pictures I created with ED-A-SKETCH. For example, I sometimes make colored transparencies of my screen displays for briefings. The G-PROM graphics are much better than the original Heath graphics. This is my primary use of G-PROM's capabilities.

I tried to dump the pictures I created with ED-A-SKETCH and G-PROM characters to my Zenith MX-80A printer (same as Epson MX-80, type III) using FBE's SCREEN DUMPER program. Unfortunately, the old graphic characters were printed. See Figure 1 and Photo 4.

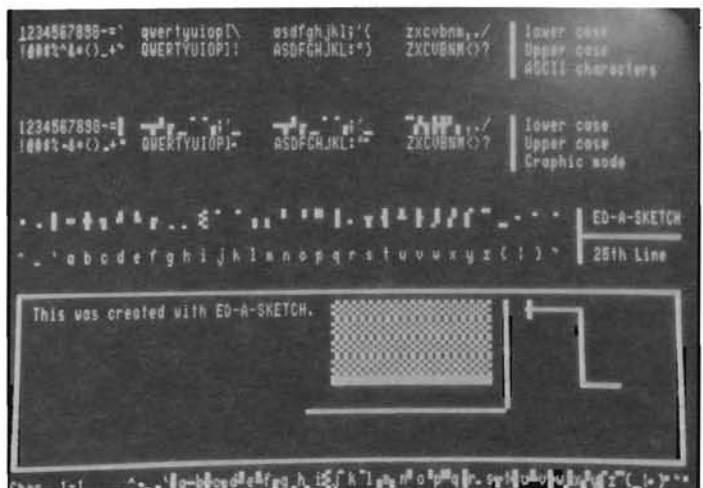


Photo 4
This is a picture of the screen that produced Figure 1.

Photo 1 shows new characters which Figure 1 shows old characters.

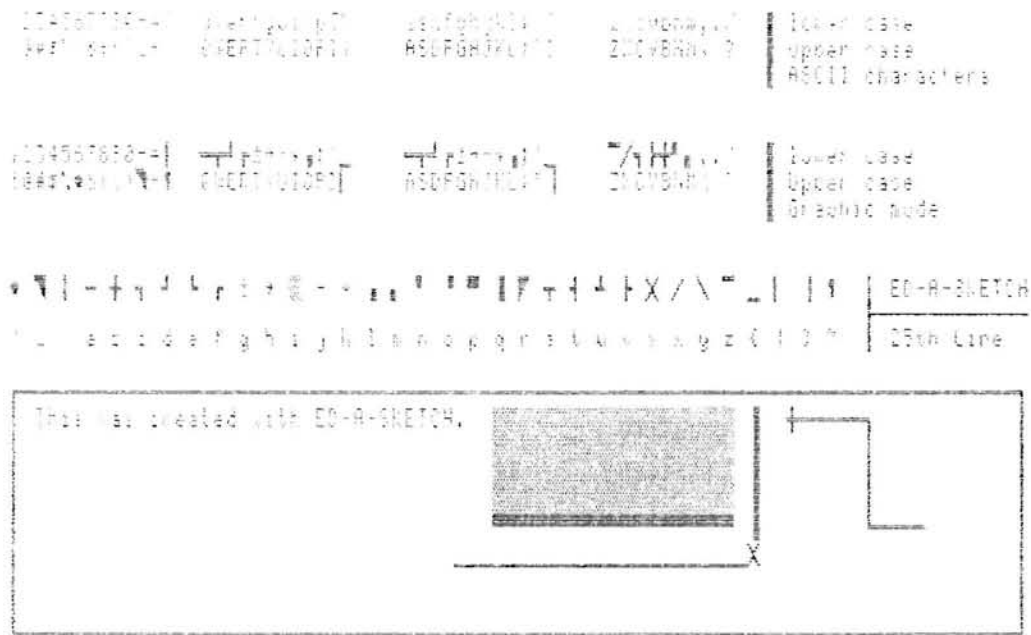


Figure 1

I asked NORCOM if there was some way to turn the old graphics "on and off" since they publish an MBASIC program to demonstrate the new and old resolutions. They told me that there was no way to do it. If they ever bring out a revision, I recommend they include a way to use the old graphics for games and SCREEN DUMPER.

Are there any other products which increase screen resolution for the H-89 or H-19? Yes, there are. Two of the most prominent are SigmaSoft and System's, Interactive Graphics Controller/Pseudo Disk (4488 Spring Valley #107, Dallas, TX 75234, 214-392-1025) and Cleveland Codonics Inc's, Imaginator (18001 Englewood Dr., Cleveland, OH 44130, 216-243-1198). Both start at more than \$400. However, they do more than increase screen resolution.

SigmaSoft and System's product offers resolution of 640 X 250 pixels. It also has a 64K (optionally expandable to 256K) pseudo disk. It supports ULTRAROM and SUPER19 ROM's. Expect to pay about another \$100-\$200 for options such as increased pseudo disk memory, cables, ROMs, and printer drivers. This product does produce a very nice printout on an MX-80 (and other printers as well), which I understand, is the same on screen as on paper.

The Imaginator is a screen graphics hardware system (504 X 247 pixels) with an instruction set that allows you to draw graphics from high level languages. Wizard Software House (79 Marshall Street, Providence, RI 02909, 401-331-5034) offers a program called "Illustrator" for about \$90, which is a screen and printer editor for the Imaginator. Newline Software (P.O. Box 289, Tiverton, RI 02878, 401-624-3322) used to offer the same product at the same price.

Can I recommend the G-PROM? Yes. It works well, with the exception of distortion in some graphic intensive games and the inability to dump the higher resolution graphics to a printer. However, it has absolutely superior screen graphics when compared to Heath's original set. G-PROM excels at screen oriented graphics. As far as I can tell, it is totally compatible with ED-A-SKETCH. Other graphic editors, such as Video Artist by Newline,

might also be compatible. G-PROM is inexpensive, less than \$20 including shipping, and support is there from NORCOM. If my experience is typical, you will like the increased screen resolution with G-PROM. My impression, after working the terminal for over 300 hours with the programs listed in Figure 1 is that G-PROM is one great buy for the Heath/Zenith computer community.

Free Software

to Balance Your Checkbook

It's **Check Processor** from Newline Software, and it's yours -- absolutely free. A \$50 value. Assign checks to 31 user-defined expense categories. Print out reports. Even use it for credit card applications! Easy to use. All documentation on disk. Available for IBM-PC and Zenith Z-100.

In addition, you'll get free demos of more software from Newline! It's our way of getting acquainted. We know that once you try our software, you'll be back for more.

Call or write for **Check Processor plus free software demos** today. Please specify hardware and operating system, and include \$4.99 for shipping, handling and materials. Ask for free catalog.

 **NEWLINE SOFTWARE** P.O. Box 289
 Tiverton, RI 02878
 M/C & VISA welcome (401)-624-3322

Figure 2
G-PROM/Software Compatibility Chart

Program *	Compatibility with G-PROM
PIE	Appears totally compatible
TEXT	Appears totally compatible
SPELL	Appears totally compatible
TEXT PROCESSOR	Appears totally compatible
EDIT19	Appears totally compatible
HOME FINANCIAL SYSTEM	Appears totally compatible
MYCALC	Appears totally compatible
ZENCALC	Appears totally compatible
QUERY!2 Series of programs	Appears totally compatible
AUTOFILE	Appears totally compatible
MBASIC CAI	Appears totally compatible
MBASIC QUIZ (HUG P/N 885-1097)	Appears totally compatible
MBASIC DATABASE SYSTEMS (HUG P/N 885-1108)	Appears totally compatible
ZLYNK I	Appears totally compatible
TYPING TUTOR	Appears totally compatible
MBASIC 4.82	Appears totally compatible
ED-A-SKETCH	Appears totally compatible
PACK & CRYPT	Appears totally compatible
SCREEN DUMPER	Doesn't print the new characters. What you see on the screen is not what is printed
SHUTTLE LANDER	Slight distortion
SPACE INVADERS	Slight distortion
GALACTIC WARRIOR	Moderate distortion
HUG'S HARD SECTORED SUPPORT PACKAGE	Appears totally compatible
HUG'S HDOS UTILITIES BY PS	Appears totally compatible
SPEEDUP	Appears totally compatible
MX.DVD	Appears totally compatible
SUPERSYSMOD 2	Appears totally compatible
WORDSTAR (CP/M)	Appears totally compatible
HUG'S CP/M UTILITIES BY PS	Appears totally compatible
TEXT PROCESSOR (CP/M)	Appears totally compatible
CPS (CP/M)	Appears totally compatible
GALACTIC WARRIOR (CP/M)	Appears totally compatible

* All programs under HDOS unless otherwise noted.

About The Author

R. Kenneth Strum (Kent) is an officer in the USAF. He is Directory of Administration at the Aerospace Medical Division, Brooks AFB Texas. He's working on his Master's degree in Computer Resource Management through Webster University, and built his H-89A through one of his micro-computer courses in 1983. He published an article in REMark in November of 1984 on printing graphics with the H-89 and MX-80A. He uses his computer for church membership databases, home finance, writing, and at the office. *

Bold, Stylish Text for H-19 and H-89!

T-Prom provides an entirely new look

for your CRT display. All of the text characters are enhanced for greater style, clarity and boldness. Now you have a choice between "stick" letters and clear, stylish screen text.

T-Prom is a plug-in replacement for the original character generator ROM, and is priced at only \$19.95 including shipping, documentation, and installation instructions. (Another new product, **GT-Prom** offers the combined benefits of T-Prom and G-Prom for only \$24.95. Write for information on G-Prom.)

Send your order to: **NORCOM**
9630 Hayes
Overland Park, KS 66212

Bromley Engineering Internally Mounted Hard Disk Drive Kits for Z-100

BS-Z100-11	11Meg formatted	\$1095
BS-Z100-22	22Meg formatted	\$1195
BS-Z100-35	35Meg formatted	\$1395

Call for quote: Can use ANY ST506 Winchester
Includes: Winchester, ribbon cables, controller,
boot rom, software and instructions.

Features: Bootable, 16 partitions, plug in and go
operation, one year limited warranty.

System requirements: 192K RAM, DOS 2 or
greater, Current Zenith PAL (on motherboard).
Uses any power supply.

Bromley Engineering

883A Cambridge St., Cambridge, MA 02141
Call (617) 661-3144, Ask for Brian.

PAINT.ASM

Part 2

Steven W. Vagts
9509 Gray Mouse Way
Columbia, MD 21046

As you recall, last month we discussed the screen dump program, "PAINT.ASM". It demonstrated the use of CP/M system calls, H/Z graphics, simple printer operations and disk file manipulations in a simple, but lengthy program. We also discussed a few possible uses, especially for possibly presenting graphics in games.

Assuming the reader has loaded and tested this program, we will now add the enclosed routines to enable loading H/Z graphics fonts to a capable printer and allow the sending of PAINT's screens directly to this printer. This greatly enhances the utility of the program allowing the development of personalized letterhead, cards and graphic prints.

Add the enclosed routines to your program. Location doesn't especially matter. Notice that the CHFONT routine replaces the one line routine previously placed in the program and the PRNTR routine is a direct substitution for the previous PRNTR routine (for the Heath H-14 printer).

The reader desiring to meld both the H-14 and new printer routines together will find it an easy modification, especially if they ask a question at the program start to determine which printer will be used and save this information as a flag. Simply check this flag in the PRNTR routine to determine which subroutine to use. I have information on H/Z CP/M BIOS and CDR BIOS patches to enable changing printers (between serial and parallel) on each cold boot.

Panasonic KX-P1092 Printer

Before describing the additional routines, I need to briefly describe the capabilities of the Panasonic KX-P1092 dot matrix, near letter quality, multi-mode printer. The two other printers in the Panasonic series, KX-P1090 and 1091, are similar.

The KX-P1092 is a very reasonably priced, near letter quality printer with some of the finest print I've seen in its price class.

The printer's nine pin print head forms a 9X9 dot matrix character in draft mode and 12X18 in the NLQ mode. It is capable of being programmed with user defined font/graphics characters. Additionally, it has several of its own character sets, including IBM graphics, selectable through the use of dip switches.

It is capable of five modes of text printing: draft font, proportional font, near letter quality (NLQ) font, draft italics and near letter quality italics. These fonts and graphics sets are stored in ROM (Read Only Memory).

User defined fonts must be downloaded into the printer's rather limited buffer, one of the few complaints I have. This printer buffer totals 7.5k (kilobytes). Another 4k is provided on a rather expensive, optional board. Switches divide this buffer between that portion needed for the downloaded font and that available for printing purposes.

Downloading The H/Z Font

Using the FONT (f5) key will display several font choices in a menu. Choices include the three ROM fonts and the downloadable draft font we will develop. Downloadable proportional and NLQ fonts can also be included easily by the reader.

On selection of the draft font, the program loads a disk file, "HZFONT.DFT", which contains the graphics characters and reverse video characters we develop and sends them directly to the download area of the printer's buffer. Examples of downloadable draft font are shown in Figure 1. Similar characters can be loaded for the proportional and NLQ fonts, but this is left to the reader as an exercise. The proportional printing has its limitations (my second minor complaint — I can't get it to print only 80 characters per line) so I can't recommend it, though the characters are much more clearly defined without taking as much memory as the NLQ font. Examples of NLQ programmable font are included in Figure 2.

Downloading NLQ font also has a limitation, which gets me back to my first complaint. Due to the extensive memory required to define each character (31 bytes of memory per character), reverse video cannot be loaded for any of the alphabet, because it would require that all the printer's buffer space be devoted to the downloaded font, leaving none as a printer buffer. I have not done this to determine its effect on computer/printer operations and speed.

The draft font easily fits in the required buffer space and is very successful, though the characters are only roughly defined. In

draft font, this printer cannot fire a single print head pin twice in succession. Note the difference between Figures 1 and 2.

Therefore, using draft font, all graphics (normal and reverse video) and reverse video of all, but small alphabet letters, can be loaded and accessed. The limiting factor becomes the number of accessible downloaded character positions that can be accessed at any particular time.

As shown in Figure 3, the printer's character set is defined as a matrix. Each element of the matrix is defined in hexadecimal (Hex) code.

For example, if the printer were told to print character 04BH, standard Hex code for the letter K, the printer refers to the matrix, finds column 4 and row B, and locates the letter K to print. Using eight bits, only 000H to 0FFH, or 0 to 255 decimal, elements can be addressed.

After loading the new character font, we must inform the printer to refer to the download buffer area rather than its own character set to determine the character to print. This is accomplished through MSG41 of the enclosed program. MSG41 is the hexadecimal representation of `ESC+%+1+0'`.

`ESC+%'` informs the printer to use the font described by the following two numbers. The first, a `1` or a `0` represents the downloaded buffer area or the ROM area, respectively. The second number indicates the font:

- 0 = Draft font
- 1 = Proportional font
- 2 = Near Letter Quality font

These are represented by lines `MSG41'` thru `MSG46'` of the program.

Composing The "HZFONT.DFT" Font Disk File

The "HZFONT.DFT" file is easily composed using a word processor, such as "MAGIC WAND".

For ease in loading the font from separate disk files and for easily identifying changes to be made to the font in the disk file, the font disk file is composed of binary code. This differs from the procedures given in the printer's operating manual, which seem to be overly difficult.

As you can see in Figures 1 and 2, the pattern of `0`'s and `1`'s enables easy visualization of the character being formed. `1`'s represent when a print head pin would fire. `0`'s represent blank areas.

Additionally, the `0`'s and `1`'s being read from the disk file convert easily in the program. The carry flag is used to transfer each bit of information provided by the `0` or `1` to form a byte of character information in the GETFONT routine. The disadvantage is the amount of disk space dedicated to downloadable font files.

In Figures 1 and 2, notice that the orientation of the examples are sideways. This is because the dot matrix printer uses a print head with a vertical row of pins. It then prints each character from left to right, all appropriate pins firing at the same time, to form each vertical segment of the character. Each of these vertical segments require one byte of data in the draft and proportional fonts and two bytes in the near letter quality font. Each line of data is, therefore, read into memory to form each of the vertical segments of the character.

DRAFT CHARACTER SAMPLES

00001010,ONE	00001010,B	00001010,d
11111111,	01111110,	00011000,
00000000,	00000000,	00000000,
01111001,	00000000,	00011000,
00000000,	01111110,	00000000,
00000000,	00000000,	00011111,
00000000,	01101110,	00000000,
01111111,	00000000,	00011111,
00000000,	00110100,	00000000,
11111111,	10000001,	00000000,
00000000,	00000000,	00000000,
11111111,	11111111,	00000000,
00001010,2	00001010,C	00001010,e
11111111,	11111111,	00000000,
00000000,	00000000,	00000000,
00111001,	10000001,	00000000,
00000000,	00000000,	00000000,
01011110,	01111110,	00011111,
00000000,	00000000,	00000000,
01101110,	01111110,	00011111,
00000000,	00000000,	00000000,
01110001,	00111100,	00011000,
00000000,	00000000,	00000000,
11111111,	11111111,	00011000,
00001010,3	00001010,D	00001010,f
11111111,	01111110,	00000000,
00000000,	00000000,	00000000,
00111100,	00000000,	00000000,
00000000,	01111110,	00000000,
01101110,	00000000,	11111000,
00000000,	01111110,	00000000,
00110100,	00000000,	11111000,
00000000,	00111100,	00000000,
10001001,	10000001,	00011000,
00000000,	00000000,	00000000,
11111111,	11111111,	00011000,

REVERSE VIDEO GRAPHICS

Figure 1

The printer is sent some preliminary bytes of information to inform it that the data it will receive are downloaded font characters. In the KX-P1092, the command defining the download of draft font is `ESC+&+0+n+m+a+P1+P2+...+P11'`.

This command is broken down easily. `ESC+&+0'` tells the printer that it is to expect a character or more to download.

The letters `n'` and `m'` define the start and stop areas on the character matrix, Figure 3, that we discussed. For example, if we specified `041H,05AH'`, then the printer would expect 26 new characters to be downloaded and used in place of the capital letters of the alphabet.

The previous information is contained in MSG35 of the enclosed program. Here, we are downloading characters from 080H to 0FFH, the entire second half of Figure 3; a total of 128 characters.

The `a'` tells the printer the "attribute" of each forthcoming character. It specifies whether the upper or lower eight print head pins will be used to print the character and other information to be used if proportional spacing is used. Looking at Figure 1 again, the first line of each example is the "attribute". It designates the use of the lower eight pins and the start and stop print-

ing position of each character. For our purposes, we don't need to worry about this last information.

The `P1', `P2', etc. are the lines of character information following the attribute line. In draft font, there are 11 lines of character information, in the proportional font, 18 lines, and in the NLQ font, 15 lines of two bytes each.

Following every line of eight `0' and `1' combinations in Figure 1, a comma must be inserted. This informs the program that it has received enough information to form a byte of character information and it needs to be sent to the printer. The temporary storage buffer, FTWORD, is then cleared and prepared for another eight bits of information.

Notice that in Figure 2 a semicolon is used following the "attribute". Because two bytes must be sent for each vertical segment of a character, in reverse order than that shown by the way I differentiate between the two kinds of information.

NLQ CHARACTER SAMPLES

```
00001111;ONE      00001111;2
11111111,11111111, 11111111,11111111,
11111111,11111111, 00000111,11100011,
11111111,11111111, 00000001,11110001,
11111111,11111111, 00010001,11111000,
00111111,11100111, 00011000,11111000,
00111111,11100001, 00011000,11111000,
00000000,00000000, 00011000,11111000,
00000000,00000000, 00011000,11111000,
00000000,00000000, 00011000,11111000,
00111111,11111111, 00011100,00111000,
00111111,11111111, 00011110,00000001,
11111111,11111111, 00000111,00000011,
11111111,11111111, 00000111,10000111,
11111111,11111111, 11111111,11111111,
11111111,11111111, 11111111,11111111,
```

REVERSE VIDEO

```
00001111;b      00001111;c
00000011,11000000, 00000011,11000000,
00000011,11000000, 00000011,11000000,
00000011,11000000, 00000011,11000000,
00000011,11000000, 00000011,11000000,
00000011,11000000, 00000011,11000000,
11111111,11111111, 11111111,11000000,
11111111,11111111, 11111111,11000000,
11111111,11111111, 11111111,11000000,
00000011,11000000, 00000000,00000000,
00000011,11000000, 00000000,00000000,
00000011,11000000, 00000000,00000000,
00000011,11000000, 00000000,00000000,
00000011,11000000, 00000000,00000000,
00000011,11000000, 00000000,00000000,
```

GRAPHICS

Figure 2

At a semicolon, the program knows to send the current byte of information directly to the printer. At the first comma, it knows to temporarily store the current byte of information. At the second comma, it sends the new byte of data, then recalls the first byte and sends that.

The end of file is marked with a `#' so it isn't confused with the CTRL-Z that marks the normal end of file. This enables you to load and stop loading at different parts of the character matrix of Figure 3 without having to develop separate disk files.

Programmable Line Spacing

Finally, to make effective use of the program, the printer's line spacing must be changed. The line spacing default is 1/6 inch; standard for most printers. Using the printer's internal ROM graphics characters, this is fine because its own graphics characters are taller than those that are downloaded. The printer actually takes two passes to first print the top half, then the bottom half of those in ROM.

Since our characters are limited to what can be printed by 8 pins, we need to be able to reduce the line spacing so no space is printed between lines.

The KX-P1092 has several different commands available for this, so depending upon your needs these can be changed. Program lines, MSG24 thru MSG28, give the most common selections. This printer can be programmed up to a line spacing of just over an inch, if necessary.

Using The Program

Last month, I explained a use of the program to add graphics screens to game programs. Once the graphics screen was displayed, it could be changed in the same manner action graphics are produced — through overwriting certain areas of the screen to give a sense of motion.

This month, I'd like to describe the use of this program to produce unique, custom-designed letterhead.

After entering the program and clearing the screen, design what you would feel to be appropriate for your letters, in the same manner the game screens were developed. After a screen is completed, save the finished product to the disk.

Several slightly different versions can be developed and saved for later comparison by saving each version to disk, making changes, and saving the new version under a different name. I use the same name, but place a different number for each at the end of the name.

While designing the letterhead screen, send versions to the printer to get a feel for the differences between the on-screen version and the printed version. Depending upon how accurately your graphics characters match those on the screen, the printed result may be considerably different. That's one of the advantages of the separate data file, however, characters can easily be changed to fit the occasion without having to reassemble the program. These characters don't even have to be H/Z graphics — design your own to do what you want them to.

Also, be alert to the difference in character height. My design of a round space station on the screen, turned into a rather squat oval on paper because of the different character height. Remember, we had to adjust the printer's line spacing for the same reason.

Our program does not operate in conjunction with whatever word processor you will use for your letter, so the letterhead screen and the letter will require careful planning so as not to interfere on the paper, and must be printed independently. Print one first, then reinsert the paper in the printer and print the other.

Similar graphics could be placed at the bottom of the paper for truly individualized mail.

Figure 4 is a brief MBASIC-80 routine that will read the disk file created with PAINT.ASM and display it on the screen. Line 110

PANASONIC KX-P1092 DOT MATRIX CHARACTER SET

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NULL		SP	0	@	P	'	p	■	■	■	■	■	■	■	■
1		DC1	!	1	A	Q	a	q	■	■	■	■	■	■	■	■
2		DC2	"	2	B	R	b	r	■	■	■	■	■	■	■	■
3		DC3	#	3	C	S	c	s	■	■	■	■	■	■	■	■
4		DC4	\$	4	D	T	d	t	■	■	■	■	■	■	■	■
5			%	5	E	U	e	u	■	■	■	■	■	■	■	■
6			&	6	F	V	f	v	■	■	■	■	■	■	■	■
7	BELL		'	7	G	W	g	w	■	■	■	■	■	■	■	■
8	BS	CAN	()	8	H	X	h	x	■	■	■	■	■	■	■	■
9	HT)	9	I	Y	i	y	■	■	■	■	■	■	■	■
A	LF		*	A	J	Z	j	z	■	■	■	■	■	■	■	■
B	VT	ESC	+	B	K	[k	[■	■	■	■	■	■	■	■
C	FF		,	C	L	\	l	\	■	■	■	■	■	■	■	■
D	CR		-	D	M]	m]	■	■	■	■	■	■	■	■
E	SD		.	E	N	>	n	>	■	■	■	■	■	■	■	■
F	SI		/	F	O	_	o	DEL	■	■	■	■	■	■	■	■

Figure 3

requests the file name, but A\$ may be given any legal file name by the host program. Place lines 40 and 120 through 180 in your game programs to see how easy it is to enhance, otherwise, somewhat dull programs. I've added several graphics screens to a STAR TREK game and it makes a tremendous difference.

Closing

That completes the review of the high points of the program. However, programmers desiring to make modifications should be alert to another difficulty I encountered.

The KX-P1092 has a switch that allows the zero to be printed slashed or unslashed. The printer apparently places the slashed zero at position 07FH on the character matrix in Figure 3. As illogical as it is, however, it has an effect on other places in the matrix.

Selection of the slashed zero will not permit a character to be downloaded at position 0B0H! Instead, that new character appears at position 0FFH, overwriting whatever character you were trying to download there! Strange? Don't select the slashed zero if you want to download to those locations and you will be fine.

The KX-P1092 printer can be summed up as a very capable printer for the money, with an excellent near letter quality type. Programming it does seem to be overly complicated, however; mostly due to its numerous capabilities. Certain capabilities are available in one or two of the ROM fonts, but not in the third. The proportional font, for example, can't be limited to an 80 character line, at least not that I've been able to find.

There are several ways that a person would think the printer could be forced to print only 80 characters per line during proportional printing. There is a command to set the right margin, but not applicable to this font. How about selecting Pica Pitch (10 characters per inch)? No, that forces you to the draft font.

Programming a line feed or carriage return causes a capital 'X' to be printed in the first column, overwriting what should be printed there, but otherwise works fine. All efforts to eliminate the 'X' failed.

```

10 REM -----
20 REM * * * DISPLAY PAINT SCREENS ROUTINE * * *
30 REM -----

40 E$=CHR$(27): E1$=E$+"E"

100 REM GET NAME OF FILE WITH STORED DATA (OPTIONAL)
110 INPUT "What PAINT data file do you wish displayed", A$
120 PRINT E1$: REM CLEAR SCREEN
130 OPEN "I",#1,A$
140 FOR X=1 TO 24
150 IF EOF(1) THEN 180
160 LINE INPUT#1,Z$
170 PRINT Z$
180 NEXT X
190 STOP
    
```

Figure 4

Getting back to the program itself; it was well worth the programming experience. Months of effort were expended on eliminating minor bugs. A fix to one, always seemed to create a whole new set of them.

Would I do it again? You bet. I haven't had this much fun since I put reverse voltage to an electrolytic capacitor. It makes an excellent firecracker, by the way — and smoke was everywhere.

I hope everyone has as much fun developing this program that I had, and I hope these two articles have helped to some extent.

As I've said before, I'm interested in any improvements readers feel may be appropriate and I'd be happy to address questions if self-addressed, stamped envelopes are included.

As I mentioned last month, I have developed much more of a program than presented here, but I am limited to H/Z and CDR CP/M formats. I would be happy to send the source code for everything except the data files for \$3.00. I'll also send the finished product on disk, in the format discussed above for \$7.00, if you include a preformatted disk. Write or call for additional information. Please include a phone number in case I have problems.

Good luck in all your future programming endeavors.

Equipment And Information Sources Mentioned

In addition to REMark Magazine & Heath/Zenith Computers:

- Panasonic KX-P1092 Multi-Mode Printer
- Panasonic Industrial Company
- One Panasonic Way
- Secaucus, NJ 07094
- "MAGIC WAND" Word Processing System.
- Peachtree Software, Inc

Listing

```

; VAGTS PAINT PROGRAM - CONTINUED
;
CHFONTE LXI   D,MSG3   ;DUMP SCREEN TO MEMORY
CALL    MSG50
CALL    SAVE
LXI     D,MSG9   ;DISABLE LINE 25
CALL    PMSG
QFONTE  LXI   D,MSG40  ;GET PRINTER FONT
CALL    PMSG
MVI     C,CONIN
CALL    BDOS     ;A=CHAR
CPI     031H
JZ      DRAFTD  ;DRAFT DNLOAD
CPI     032H
JZ      DRAFTR  ;ROM DRAFT
CPI     033H
JZ      PROPR   ;ROM PROPORTIONAL
CPI     034H
JZ      NLQR    ;ROM NEAR LETTER QUALITY
CPI     035H
JZ      SETROM  ;RESET ALL ROM FONTS
CPI     036H
JZ      LPLS    ;SET LP LINE SPACING
CPI     037H
JZ      DSPLY   ;NO CHANGE
JMP     QFONTE

DRAFTD  MVI   A,1
STA     FONTEG ;STORE FONT
LDA     DRAFTF
ORA     A
JNZ     DFTSET ;FONT ALREADY SET
CALL    CLRFCB
MVI     B,11
LXI     D,DFCB+1 ;BEGIN OF FCB FNAME
LXI     H,HZDRAF ;BEGIN OF HZFONT
CALL    LDFCB
MVI     B,0
AGAIN   PUSH  B
CALL    OPNFIL
CPI     255    ;A=255 IF ERROR
POP     B
JNZ     LDFFONTE
INR     B
LXI     D,DFCB ;DRIVE BYTE OF DFCB
MOV     A,B
STAX    D
CPI     003H   ;# OF ACTIVE DRIVES+1
JZ      NOFILE ;LAST DRIVE CHECKED
JMP     AGAIN  ;CHECK ANOTHER DRIVE

NOFILE  LXI   D,MSG14  ;NO FILE IN DIRECTORY
CALL    PMSG
CALL    DELAY
JMP     QFONTE

LDFFONTE MOV  A,M ;GET LETTER OF FNAME
STAX    D ;STORE IT IN FCB
INX     D
INX     H
DCR     B ;A CHAR WAS STORED

JNZ     LDFFONTE ;LOAD ANOTHER CHAR
RET

LDFFONTE LXI   D,MSG20  ;LOADING FONT FROM DISK
CALL    PMSG
CALL    LDROM  ;LOAD ROM FONTS
LXI     H,MSG35 ;DLOAD DRAFT
CALL    LPMSG
CALL    GETFFONTE
ORA     A
JZ      SETROM
CALL    DELAY
MVI     A,1
STA     DRAFTF
DFTSET  LXI   H,MSG41  ;SET DRAFT PRINT
CALL    LPMSG
JMP     DSPLY

GETFFONTE MVI   A,0
STA     FTWORD ;CLEAR INPUT FONT WORD
RDFONTE  CALL  RDRECD
ORA     A
RZ
RDLOOP  MOV   A,M ;GET A CHAR
CPI     030H   ;IS CHAR A 0?
CZ      BIT0
CPI     031H   ;IS CHAR A 1?
CZ      BIT1
CPI     02CH   ;IS CHAR A COMMA?
CZ      LPWORD ;MUST HAVE A FULL WORD
CPI     023H   ;END OF FILE "#"
JZ      FONTEDN ;DONE
NEXTCH  INX   H
DCR     B
JNZ     RDLOOP ;LOOP UNTIL 128 CHARS
JMP     RDFONTE ;GET ANOTHER RECORD

FONTEDN CALL  CLOSDK
MVI     A,1
RET

RDRECD  LXI   D,DFCB ;FCB ADDRESS
MVI     C,READ
CALL    BDOS ;TRY TO READ A RECORD
ORA     A ;READ OK?
JNZ     UNSUC1
LXI     H,DMA ;POINT TO DMA AREA
MVI     B,128 ;PRINT 128 CHARS
MVI     A,1 ;GOOD READ
RET

BIT0    MVI   A,0
RAR     ;ROTATE BIT0 OF A TO CARRY
JMP     BITSET
BIT1    MVI   A,1
RAR     ;ROTATE BIT0 OF A TO CARRY
LDA     FTWORD ;LOAD PARTIAL WORD
RAR     ;ROTATE CARRY TO BIT7 OF A
STA     FTWORD ;STORE PARTIAL WORD
MVI     A,0
RET

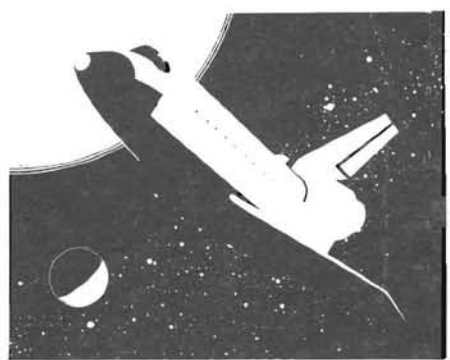
```

LPWORD	PUSH	B			
	PUSH	H			
	LDA	FT#WORD	:A=CHARACTER		
	MOV	E,A			
	CALL	PRNTCH			
	MVI	A,0			
	STA	FT#WORD			
	POP	H			
	POP	B			
	RET				
SETROM	MVI	A,0			
	STA	FONTFG	:STORE FONT		
	CALL	LDR0M			
	LXI	D,MSG19	:ROM FONT SET		
	CALL	PMSG			
	CALL	DELAY			
	JMP	QF0NT			
LDR0M	MVI	A,0			
	STA	DRAFTF	:ZERO DRAFT FONT FLAG		
	LXI	H,MSG34	:DLOAD ROM CG		
	CALL	LPMSG			
	RET				
DRAFTR	MVI	A,0			
	STA	FONTFG	:STORE FONT		
	LXI	H,MSG44	:SET ROM DRAFT PRINT MSG		
	CALL	LPMSG			
	MVI	A,0			
	STA	DRAFTF	:ZERO DRAFT FONT FLAG		
	JMP	DSPLY			
PROPR	MVI	A,0			
	STA	FONTFG	:STORE FONT		
	LXI	H,MSG45	:SET ROM PROP PRINT		
	CALL	LPMSG			
	JMP	DSPLY			
NLQR	MVI	A,0			
	STA	FONTFG	:STORE FONT		
	LXI	H,MSG46	:SET ROM NLQ PRINT		
	CALL	LPMSG			
	JMP	DSPLY			
LPLS	LXI	D,MSG24	:LPT LINE SPACING		
	CALL	PMSG			
	MVI	C,CONIN			
	CALL	BDOS	:A=CHAR		
	CPI	031H			
	JZ	LS7	:SET LPT LS		
	CPI	032H			
	JZ	LS8	:SET LPT LS		
	CPI	033H			
	JZ	LS9	:SET LPT LS		
	CPI	034H			
	JZ	LS12	:SET LPT LS (DEFAULT)		
	CPI	035H			
	JZ	DSPLY	:NO CHANGE		
	JMP	LPLS			
LS7	LXI	H,MSG25			
	JMP	SETLS			
LS8	LXI	H,MSG26			
	JMP	SETLS			
LS9	LXI	H,MSG27			
	JMP	SETLS			
LS12	LXI	H,MSG28			
SETLS	CALL	LPMSG			
	JMP	DSPLY			
PRNTR	LXI	D,MSG5	:PRINTING SCREEN MSG		
	CALL	MSG50			
	CALL	SAVE			
PRLP	LXI	H,BUFR	:GOTO START OF BUFR		
	MOV	A,M	:GET CHAR		
	CPI	15Q	:CR? MEANS END OF SCREEN		
	JZ	LPDONE			
	MOV	E,A			
	CPI	ESCAPE			
	JZ	ADD1	:ALLOW FOR ESCAPE CODES		
	LDA	COUNT			
	DCR	A			
	STA	COUNT			
	CZ	SETCT	:RESET THE COUNT IF 0		
	LDA	LPRVFG			
	ORA	A			
	JNZ	ADD96			
	LDA	LPGFFG			
	ORA	A			
	CNZ	ADD129			
PRNTIT	CALL	PRNTCH			
INCH	INX	H			
	JMP	PRLP			
ADD1	INX	H			
	MOV	A,M			
	CPI	070H	:IS IT "p"?		
	JZ	LPRVID			
	CPI	071H	:IS IT "q"?		
	JZ	ELPRVID			
	CPI	046H	:IS IT "F"?		
	JZ	LPGRAF			
	CPI	047H	:IS IT "G"?		
	JZ	ELPGRAF			
	JMP	INCH			
LPRVID	MVI	A,1			
	STA	LPRVFG	:RV FLAG		
	JMP	INCH			
ELPRVID	MVI	A,0			
	STA	LPRVFG			
	JMP	INCH			
LPGRAF	MVI	A,1			
	STA	LPGFFG	:GRAPHICS FLAG		
	JMP	INCH			
ELPGRAF	MVI	A,0			
	STA	LPGFFG			
	JMP	INCH			
ADD96	MOV	A,E			
	ACI	96	:ADD 96 FOR RVIDEO		
	MOV	E,A			
	JMP	PRNTIT			
ADD129	MOV	A,E			
	ACI	129	:ADD 129 FOR GRAPHICS		
	MOV	E,A			
	RET				
LPMSG	MOV	A,M	:GET MSG CHAR		
	CPI	'\$'	:END OF MSG?		
	RZ				
	MOV	E,A			
	CALL	PRNTCH			
	INX	H			
	JMP	LPMSG			
EXIT	CALL	DELAY			
EXIT1	LXI	H,MSG33	:RESET PRINTER		
	CALL	LPMSG			
	LXI	D,MSG8			
	CALL	PMSG			
	POP	PSW	:GET CP/M FLAGS		
	POP	H	:GET CP/M STACK		
	SPHL		:SET IT		
	RET		:RETURN TO CP/M		


```

13.10.10,' ROM Fonts have been loaded ',13.10,'$'
13.10.10,' Loading Font File from disk ',13.10,'$'
01BH,045H,'SELECTION OF PRINTER CHARACTERISTICS ',13.10
' 1 = LPT Line Spacing of 7. ',13.10
' 2 = LPT Line Spacing of 8. ',13.10
' 3 = LPT Line Spacing of 9. ',13.10
' 4 = LPT Default LS of 12 (value of 1/6 inch). ',13.10
' 5 = No Change. ',13.10
'DE What is the number of your selection <1 - 5>? $'
01BH,041H,007H,01BH,032H,'$' ;LS=7
01BH,041H,008H,01BH,032H,'$' ;LS=8
01BH,030H,'$' ;LS=9
01BH,041H,00CH,01BH,032H,'$' ;LS=12
01BH,041H,00CH,01BH,04GH,01BH,048H,'$' ;DEFAULT LPT EXIT
01BH,03AH,000H,000H,000H,'$' ;LOAD ROM CHARS
01BH,026H,000H,000H,0FFH,'$' ;DNL0AD DRAFT
01BH,045H,'Select Printer Font: ',13.10
' 1 = Draft Font ',13.10
' 2 = ROM Draft ',13.10
' 3 = ROM Proportional Printing ',13.10
' 4 = ROM Near Letter Quality Printing ',13.10
' 5 = Reset all ROM Font. ',13.10
' 6 = Change printer characteristics. ',13.10
' 7 = No Change. ',13.10
'DE What is the number of your selection <1 - 7>? $'
01BH,025H,001H,000H,'$' ;DRAFT DNL0AD
01BH,025H,000H,000H,'$' ;DRAFT ROM PRINT
01BH,025H,000H,001H,'$' ;PROP ROM PRINT
01BH,025H,000H,002H,'$' ;NLQ ROM PRINT
MSG19 DB 'HZFONT DFT'
MSG20 DB 'PARTIAL FONT WORD BUFFER'
MSG24 DB '1=DRAFT FONT DNL0ADED'
DB '0=ROM, 1=DRAFT, 2=PROP, 3=NLQ FLAG'
DB '0=OFF, 1=ON'
DB '0=OFF, 1=ON'
DB '12 IS DEFAULT LINE SPACING'
MSG25 DB
MSG26 DB
MSG27 DB
MSG28 DB
MSG33 DB
MSG34 DB
MSG35 DB
MSG40 DB
MSG41 DB
MSG44 DB
MSG45 DB
MSG46 DB
HZDRAF DB
FTWORD DB
DRAFTF DB
FONTFG DB
LPRVFG DB
LPGFFG DB
LSFLAG DB

```



**EXPLORE
NEW WORLDS
WITH
HUG
GAME
SOFTWARE**

S & K Technology Inc.

Quality Software for Heath/Zenith Microcomputers

<p style="text-align: center;">WatchWord™ for the Z100</p> <p><i>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.</i></p> <p>Price: \$100</p> <p>Demo Disk: \$ 3 (Includes the Resident Speller Demo.)</p>	<p style="text-align: center;">The Resident Speller™ for the Z100</p> <p><i>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.</i></p> <p><i>50,000 word expandable dictionary, requires 192K RAM (300K for use with WatchWord).</i></p> <p>Price: \$100</p> <p style="text-align: center;">(Demo Included on WatchWord Demo Disk.)</p>	<p style="text-align: center;">The Resident Speller™ PC Version for the Z150</p> <p><i>Check your spelling as you type from most word processors. Includes a stand-alone spelling checker for ASCII files.</i></p> <p><i>50,000 word expandable dictionary. Requires 90K in addition to memory required by your word processor.</i></p> <p>Price: \$99</p> <p>Demo Disk: \$ 2</p>
--	--	--

Texas Residents add state sales tax

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

Continued from Page 43

using Z-GRAPH-100 subroutines, show the constructed orbital path ellipse, the area of the ellipse divided into equal areas with an integration routine, and a satellite orbiting with a velocity directly proportional to its true velocity. The parameters can be saved in a file for later use in developing an orbital system with ORSYS.

ORSYS will combine the files created in ORMEC to construct a system of up to 10 satellites orbiting around a common central mass. The angle of perigee and the starting point may be specified for each orbit. A system can also be saved as a file, so reviewing a system is easy. The data tables can be viewed on the screen and printed out to a draft printer. The program uses a number of matrix operations to scale and rotate the orbital paths. The graphic displays, again in interlace mode, show dashed and colored ellipses for each orbital path, so the paths may be distinguished on either a color or monochrome monitor. A zooming feature allows the user to choose the orbit that is shown, full scale. The satellites can be viewed with or without the ellipses on the screen, and the overall speed of the satellites can be varied for different effects. The velocity variations (accelerations) in each orbit and the relative velocities of different orbits can easily be seen. Some sample orbit and system files are included for initial demonstration purposes.

Both programs are compiled ZBASIC programs, so they run fairly quickly. The displays are of the ecliptic plane and do not take into account orbits that go out of the ecliptic. It is assumed that the central mass is spherical and enormously larger than the mass of the satellites, and that the satellites themselves are small enough so they do not collide or affect each other's orbital paths. Thus, it is assumed that the gravitational field of the central mass exerts

the only force in effect and has an inverse square relationship. The orbits are, therefore, re-entrant (stable). This is close enough to reality to simulate many known orbits and allow the user to ask "What if..." questions. Extensive error handling and correction techniques are employed.

Comments: The author of this package is a senior physics major at San Jose State University, and has spent many hours in developing this software. This software is intended for the serious minded individual or student involved with the concepts of Astronomy and Kepler's Laws.

TABLE C Rating: (10)

HUG P/N 885-3033-37 Update

HUGMCP, HUG P/N 885-3033-37, has been updated to version 1.1. This newer version now has the ability to set the colors of the text display, and if you're running an H/Z-100 PC/200 PC, the screen border, background, 25th status line text, 25th status line background, all independently. Additionally, HUGMCP can now concurrently print the data in the capture buffer while communicating with the host computer. Printer 'busy' will not cause the program to 'hang' or issue error messages as most other software will. Current owners of HUGMCP can obtain this update by sending their original 885-3033-37 disk along with \$5.00 to: Nancy Strunk, Heath Users' Group, Hilltop Road, St. Joseph, MI 49085. *

PIICEON ANNOUNCES AVAILABILITY OF ITS NEW 1-MEGABYTE DYNAMIC RAM MEMORY BOARD, SUPERAM™ MODEL SR1024/ZD FOR ZENITH Z-100 & HEATH H-100 MICROCOMPUTERS

This new 1MB SUPERAM Memory Board operates with both 8085 and 8088 CPU's with fast access times. Piiceon offers the same standard 1-year warranty for this board as for its time-proven 256KB memory board.

And, check these new low, low suggested retail prices...

SR50A/ZD, 256KB — \$395.00

SR1024/ZD, 1MB — \$985.00

Contact your Heath or Zenith Dealer for full details on these and other Piiceon SUPERAM Memory products — call today!

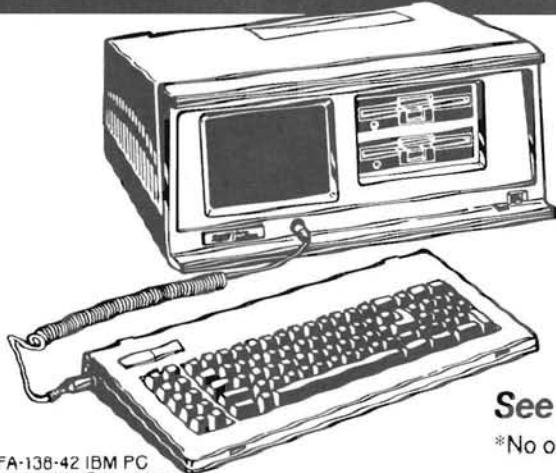
All you need to remember is... Piiceon.

PIICEON

2114 Ringwood Avenue
San Jose, California
408/946-8030

SUPERAM is a Trademark of Piiceon, Inc.

2 IBM PC Compatible Computers For Under \$1000 Each



ZFA-138-42 IBM PC Compatible Computer

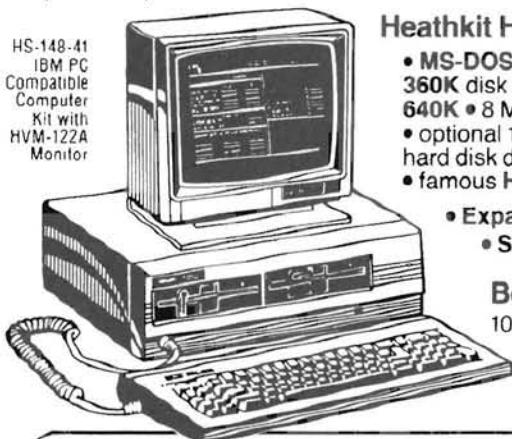
Zenith Data Systems Transportable Computer

- factory assembled • MS-DOS operating system • runs IBM software • dual 360K disk storage • 256K memory expandable to 640K • 8 MHz CPU • built-in monitor • both mono and color graphics • start-up diagnostic • famous Heath/Zenith quality

SPECIAL PRICE \$999⁰⁰*

50% off PC Savers available with purchase. See Heathkit catalog for details.

*No other discounts or special offers can be applied. Only items specified in this ad are included in the price.



HS-148-41 IBM PC Compatible Computer Kit with HVM-122A Monitor

Heathkit HS-148-41 Computer Kit

- MS-DOS operating system • runs IBM software • 360K disk storage • 256K memory expandable to 640K • 8 MHz CPU • both mono and color graphics
- optional 10 Meg, 20 Meg and 30 Meg Winchester hard disk drive available (requires expansion card)
- famous Heath/Zenith quality

- Expand to 640K Memory for just ... \$110.00
- Second Disk Drive just \$125.00

Bonus Buy

10 Meg Hard Disk Drive only \$499 with HS-148-41 (requires expansion card) or HS-151-21 purchase.

Installation included. Works with IBM PC and most compatibles. Hard Disk Drive \$599 if purchased separately.

KIT ONLY \$899⁰⁰*

Easy to Build. Assembles in just a few hours, without soldering. And backed by our promise "We won't let you fail!"

SYQUEST 10 MEG Removable Winchester Cartridge Drives



Includes drive, one 10 Meg cartridge controller board

SPECIAL \$1195.00
Reg. \$1395.00

FREE Installation

ZOOM/MODEM PC 1200 INTERNAL 1200/300 BPS modem

for PC, XT and AT computers. The PC-132 is Hayes-compatible. Includes touch tone decoder, onboard RAM for programming, battery backup, and more

Reg. \$489 **Special \$299.00**



IBM PC Compatible Computer Kit with 4 Expansion Slots

Has full IBM-PC compatibility, single 360K disk drive, expandability to 640K without an expansion card.

Second disk drive only \$125

\$1169.10* Includes free software worth \$720

HS-151-21 IBM-PC Compatible Computer Kit

NEW MASTERPIECE 4-Function Power Controller. Includes swivel base, 5 outlets, master switch, power surge protector, static ground **\$99.95**

Your Choice dBASE III or Framework II \$499

Monitor Kits from \$79.95

NASHUA DSDD Diskettes... 10 Per Box \$12.95

\$2500 CREDIT NO MONEY DOWN

We're giving qualified customers up to \$2500 of credit. Charge your purchase right on the spot. Ask for details.

ZENITH data systems

AUTHORIZED DEALER

Fine Electronics Since 1962

Prices good through May 15, 1986.

HZC-200



Call (616) 982-3614 for store locations nearest you.

Inquire about special corporate and government sales and service programs



Your **TOTAL SERVICE** computer center

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

Heath ZENITH

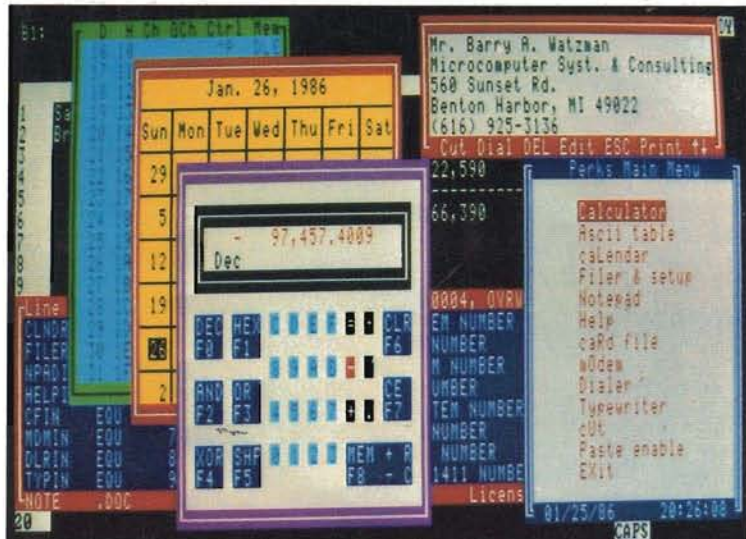
Units of Veritechnology Electronics Corporation

Computers & Electronics

Announcing
PERKS™

Version 2!!!

- New!** Cut/Paste Module!
- New!** Typewriter Module!
- New!** Variable Size Notepad Buffer (up to 64K)!
- New!** Modem Module!
- New!** Dialer Capability!
- New!** Card File Module!
- New!** Removable under *both* MS-DOS and Z-DOS!
- New!** Additional Features in Existing Modules!
- Old!** Still works with *both* MS-DOS and Z-DOS!
- Old!** Still small in size (79K)!
- Old!** Still only \$69.95!



"Shown above is an actual photo of Perks in operation, with Lotus 1-2-3 in the background. The main menu and windows for the notepad, calculator, ASCII table, calendar and card file modules are visible."

Now the leading Z-100 Desktop Utility is even better! Version 2 of Perks adds the most asked for additional features while retaining Perks' superior user friendliness, better documentation and compatability, ease of use, small size and ability to run under both Z-DOS and MS-DOS. If you have a Z-100, this is one program you can't afford to be without. And at it's low price of \$69.95*, you don't have to!

Perks is available at all Heath/Zenith Computers & Electronics Centers, many independent Zenith Data Systems dealers or directly* from:

BARRY A. WATZMAN Microcomputer Systems & Consulting
 560 Sunset Road • Benton Harbor, Michigan 49022-7142 • (616) 925-3136

Perks is a trademark of Barry Watzman.

*Plus \$4.00 for S&H, Michigan Residents add 4% sales tax also.



Hilltop Road
 Saint Joseph, Michigan 49085

BULK RATE
 U.S. Postage
 PAID
 Heath Users' Group