REMark®

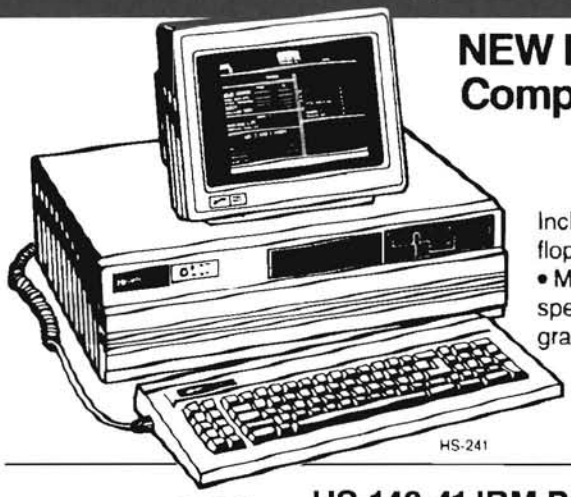
Volume 7, Issue 5 • May 1986
P/N 885-2076 Issue 76

Official magazine for users of HEATH ZENITH computer equipment.

ZPC Version 2 Is Here!
See Page 51

$2.50

IBM AT Compatible Personal Computer only $2899



HS-241

NEW IBM AT Compatible Advanced Personal Computer Kit

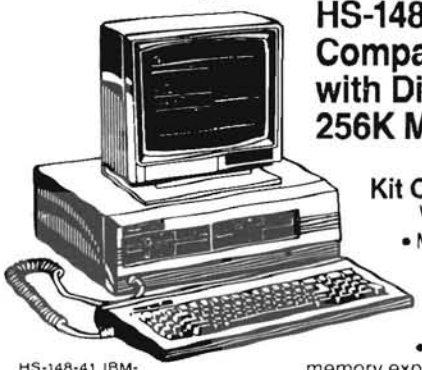
Priced Hundreds of Dollars Less Than the IBM AT!

Only $289900 Less HUG Discount

Includes: • 512K RAM expandable to over 15MB • single 1.2MB 5.25" floppy disk drive • controller card for two floppy and three hard disk drives • MS-DOS operating system • 8 expansion slots • 6 MHz clock for high speed processing • compatibility with almost any PC video card for color, graphics, and high resolution video options. **Monitors from $79.95**

See the catalog for SuperSaver discounts on accessories.

Take your HUG Discount on Heathkit and Zenith Data System Products



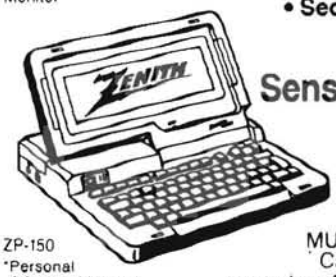
HS-148-41 IBM-PC Compatible Computer Kit with HVM-122A Monitor

HS-148-41 IBM PC Compatible Computer with Disk Drive and 256K Memory

Kit Only $89900
With HUG Discount

• MS-DOS operating system • accesses IBM software • optional 10 Meg, 20 Meg and 30 Meg Winchester hard disk drive available • 360K disk storage • 256K memory expandable to 640K • start-up diagnostic • famous Heath/Zenith quality

• Expand to 640K Memory for Just...$110.00
• Second Disk Drive Just $125.00



ZP-150 *Personal Advanced Laptop

Sensational price on this new PAL* LAPTOP COMPUTER

7½ lb. briefcase size unit has Microsoft WORKS, WORD, MULTIPLAN, and BASIC...Modem.. Calendar...Alarm...all built in. Much more, too. Act now for this unbelievable price.

Apply your HUG Discount $89900

LX-80

Epson LX-80 Printer

Just $23900

• Correspondence quality FRICTION FEED printer prints directly on your letterhead • Fast! 100 characters per second • Select typestyles at a touch. Over 150 combinations • Mix text and graphics on same line • One-Year warranty.

SYQUEST 10 MEG REMOVABLE WINCHESTER CARTRIDGE DRIVES
Includes drive, one 10 meg cartridge, controller board **$1195.00**

ZOOM/MODEM PC 1200 INTERNAL 1200/300 BPS Modem
For PC, XT, and AT computers. Hayes-compatible. Includes touch tone decoder, onboard RAM, battery backup **$299.00**

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
While quantities last **$449.00**

Smart Outlet Box with Increased Protection
Provides single switch control of up to seven electrical devices. UL approved with new, superior surge protection and noise filtering . . **Special $49.95**

NASHUA DSDD Diskettes . . . Per Box **$12.95**

ZENITH data systems
AUTHORIZED DEALER

LEASING AVAILABLE FOR YOUR BUSINESS NEEDS

Fine Electronics Since 1962

HZC-209

Prices good through June 15, 1986.
Call 616-982-3614 for store locations or consult your Heathkit Catalog.

EXPERTS: CAD NETWORKING COMMUNICATIONS

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

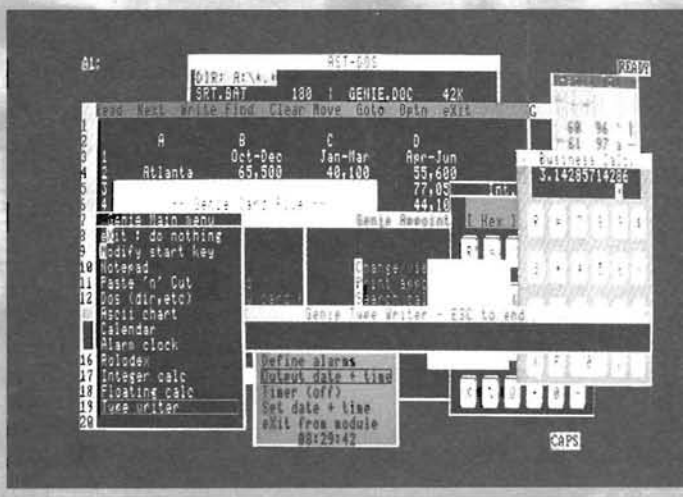
Heath® ZENITH®
Computers & Electronics

Units of Veritechnology Electronics Corporation

NOT PROTECTED

Genie™

Software Magic from Advanced Software Technologies



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

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

Here is what you get with Genie:

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

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

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

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

Only Genie Gives you so much for so little.

ADVANCED SOFTWARE TECHNOLOGIES

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

ORDERS ONLY CALL

1-800-437-4400 Ext 800

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

_____	Copies of Genie	@ $54.95	_____
_____	Spelling Helper(s)	29.95	_____
_____	Print Spuffer(s)	24.95	_____
_____	Scientific calc.	19.95	_____
_____	Genie pack (I want it all) Saving $40	$89	_____
		Sub total	_____
Check encl ☐		shipping	4.00
M/C ☐		NY Tax (8.25%)	_____
VISA ☐		TOTAL	_____

Card Number _____

Expires _____ Your Tel Num _____

Name _____

Address _____

Company _____

City/ST/Zip _____

Heath/Zenith Users' Group

Staff

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

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

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

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

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

REMark®

Official magazine for users of HEATH ZENITH computer equipment.

Volume 7, Issue 5 • May 1986

Index of Advertisers

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

On The Cover: *Pictured are the Superdrives from CDR Systems, Inc. A Review can be found on Page 49 by Jim Buszkiewicz.*

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

New Z-148 Winchester Systems

- one floppy
- one half-height winchester drive
- 640 K RAM
- 8 MHz operation
- amber Zenith monitor
- open expansion slot
- serial and parallel ports

These Z-148 systems now available at special prices:
12-Megabyte formatted . . . only $1629
21-Megabyte formatted . . . only $1829
32-Megabyte formatted . . . only $1999

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

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

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) 724-2336

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

BUGGIN' HUG

Eliminate Error Message

Dear HUG:

Pat Swayne's "ZPC Update #4" contains patches for programs compiled with the QUICK BASIC compiler. The article implies that the patches apply to programs compiled with the /O switch and which do not use a runtime module. This is to let other readers know that the patch can also be applied to the runtime module (e.g. BRUN10.EXE) if the application program was not compiled with the /O switch. By applying Pat's patches to the BRUN10.EXE module, I eliminated the "Cannot run child of BASIC" error message. Most users will also need to use the PRNFIX.BAS program (or its assembler equivalent) if the application program uses a printer. By using these two fixes, I can now run programs compiled with QUICK BASIC with the ZPC2 program.

I don't know how Pat Swayne is able to turn out so many useful programs and fixes, but he is certainly an asset to the Z-100 community. Thanks for publishing his work!

Sincerely,

Ernest Bentley
Farm Computer Center, Inc.
8 Roanoke Street
Christiansburg, VA 24073

Software Wizardry Rebuttal

Dear HUG:

In reply to your letter from James Curtsinger, published in the March 1986 issue of REMark, we would appreciate it if you would publish this letter to present our side of a tricky situation involving a Techmar Graphics Master board.

At the time Mr. Curtsinger ordered his Techmar board, we informed him that there were two different ways it could be purchased — either under a Zenith part number or under a Techmar part number. The Techmar part was less expensive to purchase, and presumably would work in the system without modification (we later found out that this wasn't true, as Mr. Curtsinger indicated in his letter). His choice was to purchase the less expensive card.

Some time later, Mr. Curtsinger contacted us to inform us that this card wasn't working in his system. We then contacted our Zenith representative, and were told that "There should be no problem as, after all, Zenith is selling them." We also contacted Techmar who repeated the same argument.

After a number of additional phone calls to Techmar, we finally determined that the boards being sold via Zenith were modified by Techmar prior to being shipped — and that a replacement address decode ROM was necessary in order to resolve an addressing conflict. We immediately ordered this ROM, at our own cost, to resolve Mr. Curtsinger's problem.

After even more phone calls to Techmar, and an extended delay, we received the ROM, and used it to correct the board. Mr. Curtsinger received the card, modified to work in his system, in October of last year.

It would appear that this letter has been laying around for some 6 months or so prior to being published. We do agree that the Techmar board addressing problem is something users should be aware of, but we question whether a 6 month old letter should be published without checking, at least, to see whether the problem had been resolved in the meantime with the customer, out of fairness. As published, the letter makes it appear that Mr. Curtsinger has been waiting since June of last year to have his problem resolved, and this is certainly NOT the case.

We'd also like to point out that it is physically impossible for a computer retailer to check every add-on circuit card in existence on a given machine, and that we indicated to Mr. Curtsinger that we hadn't actually tried it ourselves in the Zenith computer. We simply indicated to him that in checking with Zenith (which we had, in advance of the sale), we were told that there was no identified problem with using the card in the Zenith systems. Indeed, it may well be that Zenith was unaware of the ROM change at that time, as the ROM was being changed by Techmar prior to shipment.

We can certainly appreciate Mr. Curtsinger's frustration with the board and with us. We share his frustration with Techmar, and to a lesser extent, with Zenith as to the compatibility problem with the card. However, we in no way represented the board other than as we knew it, and we did resolve the problem once identified.

Our fault was apparently in not communicating our knowledge of the board adequately at the time of the sale, at least not sufficiently for Mr. Curtsinger to recall at a later date. The communication problem is something we always do our best to avoid, by anticipating customer responses before they happen, but realistically is something we have limited control over.

Unfortunately, problems such as occurred with Techmar and our Zenith contact have existed since man invented fire — and realistically, this sort of thing will happen no matter what we, as a high volume dealer, try to do to resolve it. Distributors and manufacturers are far harder to deal with than dealers, generally less knowledgeable, and certainly less helpful when it comes to resolving problems.

Largely as a result of this precise transaction, we have almost completely eliminated sales of Techmar products from our product line, and replaced them with more supportive manufacturers.

Although it obviously didn't appear so from Mr. Curtsinger's vantage point, we did everything in our power to resolve the problem he faced. Indeed, we DID resolve the problem. We never knowingly leave a customer hanging. We simply could not get Techmar to honor their promise to sell us the replacement ROM in a timely fashion — nor was anyone at Zenith able to help us (indeed THEY didn't appear to be aware of the ROM change).

Our work is predominantly in the more complex applications. We configure custom systems for specific customers and work with more complex arrangements of peripherals and upgrades. We pride ourselves on our success rate in this regard. For every

letter like Mr. Curtsinger's we receive 50–100 of praise for our work and support. Sadly, the trouble cases are usually those that find their way into magazines.

For Mr. Curtsinger's aggravation with us, I apologize to both him and the community — and guarantee that we will do everything in our power to prevent these situations in the future — even to the point of anticipating customer needs or requirements that they don't themselves recognize. In the small percentage of unavoidable problems that do arise, I can assure you that we have no intention of leaving a customer at the mercy of unusable products, nor have we ever. If you will check with Mr. Curtsinger, you will find the problem is long gone.

Sincerely,

Tom Jorgenson, President
Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, MO 63301

Tecmar Captain Multifunction Card

Dear HUG:

In your March 1986 issue of REMark, a letter from James Curtsinger of the University of Minnesota made reference to a Tecmar Captain Multifunction card which he purchased from an independent Zenith dealer, as not working properly in Zenith PC compatible computers such as the Z-151. He went on to say that the Heath/Zenith supplied model of the Tecmar Captain board has a different PAL chip to make it work with Zenith computers due to the 320K RAM standard on these machines.

I would like to point out that we have been selling the standard Tecmar Captain board for over two years now without difficulty. The only 'trick' that has to be done is to remove 9 chips from the 5th bank of RAM on the Zenith's RAM card so that there is only 256K on it. Then install those 9 chips along with any other chips you might want to add, onto the Tecmar card. Now the machine has not lost any RAM, and still conforms to the Tecmar wanting to only see 256K on the original computer's RAM card. Since the Tecmar card holds up to 384K of RAM, you still have the ability for a total of 640K of total RAM in the machine. There is really nothing complicated about it, and works fine. I just thought your readers would like to know, since a Zenith supplied Tecmar card is really non-standard in the sense that its starting address is now incorrect for any PC compatible that holds only 256K of RAM on the motherboard as it would start at 320K and leave a 64K hole.

Please note that we are authorized Zenith dealers ourselves, and are not trying to sway anyone from buying the product through any Heath store or from the factory, but thought they should be aware that there really is no problem at all with the standard Tecmar card.

Sincerely,

Ray Massa, President
Studio Computers, Inc.
999 South Adams
Birmingham, MI 48011

A Minor Correction To Halley's Comet

Dear HUG:

A minor correction needs to be made to the Halley article (my fault). The 4th Order Runge-Kutta method should be:

```
x(T+dT) ~= x(T)+(1/6)*(slope1 + 2*slope2 + 2*slope3
  + slope4)*dT
```

The x(T) was left out in the text I sent you and I apologize for this oversight. However, since this was not the method employed by the program, the omission should not impact the readability of the article to any great extent.

Sincerely,

James Tursa
3489-C Lake Austin Boulevard
Austin, TX 78703

Paul Woolgar Letter In REMark For September 1985

Dear HUG:

After replying to the avalanche of letters I received after my letter of October 6, 1984 appeared in the June issue of REMark, I vowed to refrain from any more questions, particularly since I had had no further problem for nine months.

However, after reading Mr. Woolgar's letter, I felt that I should put in my bit in defense of the H-89 and the Software Toolworks PIE program.

My next PIE failure occurred, strangely enough, when I wrote to thank Mr. George Stephenson (1253 Park Place, Quincy, IL 62301) for the detailed information he sent me regarding procedures for using DEBUG to locate and recover a file still in RAM. I used PIP LST:=B: to print my letter. It worked OK and then when I tried to make a file copy, I got the message "BDOS Error on B: Bad Sector". The keyboard was still OK this time. I used Mr. Stephenson's procedure to locate the letter and to SAVE it to a new disk, where everything ran OK.

Otherwise, I have had none of the problems delineated by Mr. Woolgar. I have a surge suppressor in the line. I use the switch in the suppressor to turn on the H-89, two additional hard-sector drives, and the printer. The only time I turn off the printer with the system energized is when I want to return all the printer controls to the default values. My system disk has CP/M 2.2.03 modified with a Lindley Systems driver to permit sending control signals to my Epson LQ1500 printer. I put the 6k PIE program on all my correspondence disks so there is no question re: soft boots. My PIE is modified to permit the use of TABs.

With two failures over a period of some three years, I think that I have no complaints. My first letter wouldn't have been necessary if I had realized that a RESET did not erase the RAM memory. I did have some minor problems before I discovered, the hard way, that the only safe way to remove a file from the screen when no change is to be made is to use the Control-E SAVE command.

Very truly yours,

Jack Miner
316 So. Rosedale Avenue
Lima, OH 45805

Don Keller's Letter Struck A Chord

Dear HUG:

Don Keller's letter in the June 1985 REMark struck a chord in me. I still have a cassette connected to my H-8, although I went to HDOS a long time ago for the obvious advantages of disk storage. And, while I now use an H-150 in my everyday work, I'm not

about to junk my H-8 (only a couple of months ago, I tore it down to install gold pins on the motherboard.

Most of my learning about microcomputing was on the H-8, and a lot of it was with cassette software. Today, we are so insulated from the machine with disk software and "user-friendly" emphasis, that it's hard to understand what is really happening inside. Granted, not everyone using computers wants to know about bits and bytes and I/O. But, I'll add my voice to Don's that a user group ought to support all levels of users.

If there is little interest voiced anymore for cassette or H-11 support, then it shouldn't involve a lot of time on the part of the staff to handle the occasional article on those subjects. At least, could those who are interested be kept informed of articles submitted (if not published), so they can write to the authors for copies?

Another idea is to form special-interest groups for those "antique" machines. If the few technological stragglers could band together (guess it's called networking in the jargon of the day), they might be able to preserve the spirit of adventure that used to be personal computing.

The newest developments are exciting for those who are up there in front. But there's still a lot for the individual to discover at whatever technical level he or she happens to be, and with whatever system is available to them. Remember, most of us were there once.

Don Skiff
339 South Ashley, #2
Ann Arbor, MI 48103

All I Get Are Horizontal Lines

Dear HUG:

I have a Z-100 with an Okidata 93. I cannot get PSCOKI.COM to dump to the printer properly. All I get are horizontal lines across the page. PSC.COM works fine. Has anyone used PSCOKI.COM successfully with a Z-100 and an Okidata 93? Also, does anyone know if MicroSoft's Word has a non-document (program writing) mode similar to WordStar?

Thank you,

Steve Munger
1970 O'Shea Lane
Marietta, GA 30062

Okidata 84 And The Z-158

Dear HUG:

I would appreciate any information that your staff or readers could provide in order to interface a serial Okidata 84 printer operating at 4800 baud rate to a Zenith 158 computer. We have been unable to configure at that baud rate using the cable pin connections provided by Heath and operating without parity.

The printer is equipped with the R5CH serial board and functioned quite nicely with the Z-120.

Your help would be appreciated in providing answers for proper pin connections and switch settings.

Sincerely,

Ervin T. MacDonald
P.O. Box 190
Houlton, ME 04730

Problems You Might Want To Avoid

Dear HUG:

I recently responded to an ad in REMark magazine to buy two upgrade kits for my Z-100: a speed upgrade to 8 MHz, and a memory expansion to 768K. I ran into some problems which others in my position might want to avoid.

I first installed the speed upgrade (instructions by UCI Corp., 948 Cherry St., Kent, OH 44240) and it worked fine, once I replaced the Intel 8088 with the faster 8088-2 (cost: approx. $13), switched for either 5 MHz or 8 MHz.

Several weeks later, I installed the memory upgrade (instructions by Various Ware, P.O. Box 21070, El Cajon, CA 92021). This particular kit requires piggybacking two 74LS257s and bussing all the pin 1s of the new 256K RAM chips to a new address line on the piggybacked IC. When it was done, I discovered that it worked fine at 5 MHz, but wouldn't work at all at 8 MHz.

The distributor of these kits (Tex-Matics Micro Systems, 3059 W. 15th St., Suite 100, Plano, TX 75075), who had previously assured me that they would work together, was pleasant, but of no help in solving the problem. Various Ware would not return my phone calls at all.

UCI, however, proved to have the answer all the time. The tech who took my call asked if I had changed a half dozen various ICs, which he named. I hadn't, of course, since nowhere was it specified that with a memory upgrade, these other changes were necessary. A simple sentence in UCI's instructions warning of this would have been decent.

The additional changes are mostly faster versions:

At U149 use Zenith #41-19 (150ns delay)
At U200 use 8T98
At U153 use 74S280
At U221 use 74ALS32
At U234 use 74S74
At U155 use 74AS00
At U146 use 74F257 (2 req'd. since this is the piggybacked chip)

With these additional changes (cost: approx. $14), the machine runs all 768K at the 8 MHz speed perfectly. I hope alerting others to this hazard will save them some grief.

Very truly yours,

Richard P. Rudy
19846 Beekman Place
Cupertino, CA 95014

Gemini Doesn't Put Slash Through Zero

Dear HUG:

I have had a Gemini-10X for several years and it doesn't put the slash mark through the zero. This has bugged me for several years, as I type in programs that don't have the slashed zero in the listing, this became very evident to me when I decided to put the CALC.BAS from March 85 REMark in the computer. In the listing, the "OH" and the "ZERO" are just opposite from the way the Gemini prints them and I had to go back through and change most all of them. This got me to thinking that there has to be a better way.

The GEMINI has dot graphics mode, so I designed a BASIC program for the H-89 using CP/M and Microsoft BASIC. I also used a compiler so I can run it from a COM file. It works out very well and any programs that I list, there is no doubt which one is which.

Here is the listing:

```
10 PRINT CHR$(7)
20 INPUT
   "*** Is the printer turned on ? (Y OR N) ***";A$
30 IF A$ <> "Y" THEN 10
40 ESC$=CHR$(27)
50 LPRINT ESC$;CHR$(42);CHR$(0);
60 LPRINT ESC$;CHR$(42);CHR$(1);CHR$(48);CHR$(0),
   CHR$(28);CHR$(66);CHR$(32);CHR$(81);CHR$(8);CHR$(69),
   CHR$(2);CHR$(33);CHR$(28);
70 LPRINT ESC$;CHR$(36);CHR$(1),
```

Hope this might help others.

Thanks,

Hubert Peery
306 E. 45th
Shawnee, OK 74801

"For Those Times When You Really Need A Typewriter"

Dear HUG:

This is in regard to Fred Ormand's article "For Those Times When You Really Need A Typewriter." For some time now, I have been doing just that without using any programs other than the operating system (HDOS) itself. I have an H-8 with H-17 and the H-19 terminal and use HDOS exclusively, but this should work with the H-89 also.

All you have to do is type:

```
COPY AT:=TT (or LP:=TT: if you are using LP:)
```

Enter your information here. If you make a mistake, you can backspace to correct until you enter a 'CR' for a new line. You can also use control codes — although they won't be printed on the screen — to control your printer. One word of caution, make sure you end the last line with a 'CR' or you might miss some of the information.

When you finish typing, all you do is type 'CNTRL D' and all the information on the screen will be printed.

Using this method is like having one of the new electronic correctable typewriters, and you don't have to load anything.

I hope this information will be of use to some fellow HUGGIES, and if there are any of you in the London, Ontario region, I'd like to hear from you. (Phone: 519-451-7565)

Sincerely,

Tom Few
R.R. #1
Arva, Ontario
CANADA N0M 1C0

Preventing Prowriter Printing Problems

Dear HUG:

A note that may be helpful to users of the C. Itoh Prowriter printer:

Bill Locke (REMark, April 1985) gave a patch to make the PSC.COM screen dump utility send the printer a Return before a Line Feed, to fix the problem of the Prowriter's printing the whole screen on one line, if the Line Feed is sent first. You can make the same patch to the source listing PSC.ASM, if you have it (although it is not mentioned in my manual, PSC.ASM is on my Z-DOS version 1.25 Distribution Disk I).

To make the patch, load PSC.ASM into EDLIN (or other editor) and locate the following section of code;

```
IF GENLF
     MOV     AL,0AH
     CALL    FAR PTR BIOS_PRINT       ; do LF
ENDIF

IF GENCR
     MOV     AL,0DH
     CALL    FAR PTR BIOS_PRINT       ; do CR
ENDIF
```

Change it to read:

```
IF GENCR
     MOV     AL,0DH
     CALL    FAR PTR BIOS_PRINT       ; do CR
ENDIF

IF GENLF
     MOV     AL,0AH
     CALL    FAR PTR BIOS_PRINT       ; do LF
ENDIF
```

Then assemble and link according to the instructions in the PSC.ASM listing.

Mr. Locke's letter helped me understand why I had the same problem when using my Prowriter 8510A with the Z-DOS PRINT utility, which apparently also sends a Line Feed before Return. I haven't figured out how to patch PRINT.COM, but it really doesn't matter, because a REALLY easy way to fix the problem is to set DIP switch 1-7 in the Prowriter to the "Closed" position (the factory setting is "Open"). I am indebted to one of the helpful people at the Sacramento Heath/Zenith center for finding this fix; the Prowriter manual was unclear to me on this point. With DIP switch 1-7 closed, my Prowriter behaves properly with PSC.COM (patched OR unpatched) and PRINT.COM, as well as with Lotus 1-2-3, Z-BASIC, and PeachText 5000.

Best Regards,

Les Landers
2841 Latham Drive
Sacramento, CA

Additional HA-108 Upgrade Information In Regards To Older H/Z-100 CPU Boards

Dear HUG:

I was very happy to see your article in the July '85 issue entitled, "Installing The HA-108 Upgrade Kit On Older H/Z-100 CPU Boards". I read this with much interest, because I have a 2-1/2 year old Zenith Z-100 that has served me well over the years; but lacks the newer 27128 monitor ROM and hardware updates that the newer machines have. As I carefully studied the article, I began to feel comfortable with the detail that Pat Swayne had put into fool-proofing the procedure. I also thought about the large WARNING! notice if something went wrong! I could ruin my $4,000 computer beyond repair! Continued on Page 83

Serial Ports
Part 1

David E. Warnick
RD#2 Box 2484
Spring Grove, PA 17362

This month, we'll begin looking at your computer's serial communications port. We'll see what it is and what it does. We'll see just what we can change to make it perform the tasks we have for it. Then, we'll write some programs to set up and use the serial port. This work applies to the Heath H-8, H-89, HS-151, and HS-158 series computers. It also applies to the portables, and to any computer utilizing the 8250 Asynchronous Communications Element (ACE) to drive its serial port.

With the information in this series, you'll be able to write applications for:

Amateur Radio RTTY and CW (Morse Code)
Amateur Radio AMTOR and Packet Transmission
Telephone Communications for the Hearing Impaired
Modem Communications
Serial Printer and Plotter Drivers

and many more, limited only by your imagination. This is an article I've planned for over a year now, as the serial port is a very powerful tool, and a thorough programming presentation is hard to find.

In Serial Communications, each byte of data is sent by sending its bits, one at a time, across a single conductor. To do this, we must take a byte of data (one character) and hold it until all bits are sent, then get the next byte. Receiving data is the reverse of this process. In order that the sending and receiving circuits can work together, they must work at the same speed. This speed is measured in bits per second or "baud". The interface must know how many bits are used to make a byte. Then there are things called stop bits and handshake signals. Somehow, the receiving circuit must be able to indicate whether it is or is not able to receive data, and the transmitting circuit must be able to recognize this indication and transfer it to the CPU (the computer's microprocessor).

When you run commercial software to drive serial printers or to communicate via a modem, these things are done for you. That's what questions like '300 or 1200 baud?' are for. They take your information and use it to program the serial port to do the job you want. Now you'll learn what these programs do, and what you need to do to use the serial port for your own applications. Then, you'll learn to write the program lines which do this for you.

The following brief description of the function of a serial port and what's in the computer to drive it is from information contained in the Heathkit H-89 and HS-151 manuals. One thing you must become aware of is that Heath/Zenith offers the finest computer documentation available. All the answers are there. Sometimes you've got to dig a little to find them, but they are there. It's a good idea, when you have a bit of free time, to pick up your manual and look through it a page at a time. You don't necessarily have to read anything. Just note what's there. Then when you have a question, you'll remember where to look. Don't worry if this stuff seems too technical. Read through it and get the drift. Even if it's too much, you'll learn to program this thing, and you'll be glad you did.

When the computer talks to a serial device (Modem, serial printer, etc.), it talks through an Asynchronous Communications Element (ACE) and RS-232 compatible line drivers and receivers. The RS-232 merely means that certain voltage levels are standard for the ones and zeros it will send and receive. Line drivers are transmitters. They send the data signals or certain control voltages from the computer. The receivers do the opposite. They get the data and control signals from the outside world and pass them to the ACE. The line drivers and receivers are circuits included in the microchips on the computer's serial interface board.

The ACE is also a chip. It's one of the large 40-pin models, and is the part we really want to look at in this article. The ACE converts parallel ASCII data to serial data and drives the communications lines through the line drivers. The ACE also receives incoming serial data from the serial port via the receivers and converts that to parallel data for use within the computer. The ACE divides the frequency of the master clock (an oscillator) to control the baud

rate which is the speed of transmission. In addition, the ACE provides management of the serial control signals (word length, start and stop bits, and parity). When the ACE has received a complete byte of serial data, it signals the microprocessor that there is data available.

So, you can see that the ACE is a very powerful chip. It's even more powerful because it can change to meet a variety of communications needs. You can control these changes by your programming and it's very easy when you understand how it works.

The ACE chip used in Heath/Zenith computers is the model number 8250. Also, the clock which is an oscillator located on the same circuit board as the ACE runs at 1.8432 Mhz (1,843,200 Hz). This information is needed to set the communications baud rate. (Do not confuse the CPU clock with the ACE clock. They're two different things.) The last piece of information we'll need is the address to which we must send the programming instructions, and where we send outgoing data and receive incoming data.

If you have an H-89, there are three ACEs to support the three serial ports. The addresses are:

320 to 327 Octal DCE
330 to 337 Octal DTE
340 to 347 Octal DCE

If you have an HS-151 or HS-158, two serial ports are possible, each with its own ACE (8250 microchip). Their addresses are:

03F8 to 03FF Hex COM1
02D0 to 02D7 Hex COM2

If you have an H-100, please check your manual for addresses. I'm sorry I don't have one of those machines. Perhaps a reader could drop a line to me or Buggin' HUG with the information.

For the purpose of this series of articles, we'll look at the port i called DTE on the H-89 and COM1 on the PC compatibles. I'll explain the DTE/DCE thing in a future article on the RS-232C interface. The important thing to learn and remember now is that what we're doing here applies to any 8250 ACE in any computer. All you have to do is change from the address we're using here to the correct address of the port you want to control on your computer.

Why are there eight addresses? I thought an address for anything connected to the computer's data buss was just like a memory address, that is, it could handle an eight-bit byte of data. Why would a single port need eight bytes?

Each address represents a separate register within the ACE chip. (The eighth is unused.) Each register can handle eight bits, one byte, of data at a time. What we put into each register, whether a given bit is a one or a zero, determines how the ACE operates, the speed of transmission, how many bits are in a byte of data, etc.

The seven registers we'll be concerned with are numbered 0 through 6. What they're called and what they do is covered in detail in a table in your owners manual. I'll include a summary here and will cover in detail what you'll need to know for our programming efforts. A copy of the table is included at the end of this article for your reference.

Register 0
Receive Buffer Register (when receiving)
Transmit Buffer Register (when transmitting)

This register is used to receive and send data through the ACE. Under a special condition, when setting up the ACE to a specific baud rate, we use this register to receive one of two numbers required to set that speed. Exactly how this is done will be explained later.

Register 1
Interrupt Enable Register

The 8250 ACE chip is capable of sending any of four different interrupts to the CPU. These are the:

Receive Data Available
Transmitter Holding Register Empty
Receiver Line Status
Modem Status

interrupts. By setting the appropriate bit to a logic one, we enable an interrupt By setting it to logic zero, we disable the interrupt. We'll see how to do this shortly. Like Register 0, this register has an extra function. Under a special condition, it receives the second number required to set the baud rate at which the ACE will operate.

Register 2
Interrupt Identification Register

This register is used to save the system CPU time and cycles, thus making your system faster. The four interrupts we can enable in Register 1 are assigned priorities. From the highest to the lowest, they are:

Receive Line Status
Received Data Ready
Transmitter Holding Register Empty
Modem Status

If any interrupt is pending, bit zero is logic zero. The CPU can thus look at this register. If bit zero is logic one, no interrupts are pending, so nothing needs to be done here. If bit zero is logic zero, an interrupt is pending and the ACE needs service. Bits two, three, and four identify the highest priority pending interrupt only, no matter how many interrupts are pending. This way, the system will take care of the most important item first, and no CPU time was required to figure out which one it was.

Register 3
Line Control Register

This is where we program the number of stop bits, parity, and break control. Bit seven of this register has a special use. if we make bit seven a logic one, we tell the ACE that we're going to put special numbers into Register 0 and Register 1 to set the baud rate. We then reset bit seven to logic zero. You'll be amazed at how easy all this is when we do it.

Register 4
Modem Control Register

This register controls the interface with the modem or other device connected to the serial port. Data Terminal Ready (DTR) and Request To Send (RTS) are turned on and off by this register. It also provides a signal loop-back capability for diagnosis and programming.

Register 5
Line Status Register

This register provides data to the CPU concerning data transfer through the ACE.

Register 6
Modem Status Register

This register is a partner to Register 4. It provides information to the system about the control lines from the modem or other

peripheral device connected to the serial port, and reports such things as Clear To Send (CTS) and Data Set Ready (DSR).

There they are. The seven registers of the ACE. I know all this sounded complicated, but please stay with me and I'll make it all very clear. We'll see how to address a register and to either put data into it or read data from it. Then, we'll see how to put these new-found programming skills to work.

The address of Register 0 is the same as the lowest address of the chip. In the case of the H-89, the address of Register 0 is 330 octal. Each successive register is one address higher. Register 1 is at 331 octal, Register 2 is at 332 octal, and so forth. If we write a program with these values in it, the program will run fine on the H-89. A reader with an H-100 or a PC clone would then have to go through the entire program, recognize our addresses, and change them to the correct number for his machine.

If you've read my articles in the past, you know of my penchant for standardization of those things which make programming easier and programs more readable (understandable). Here's one idea which other authors have used, and which makes a lot of sense. When we talk about an ACE and its address, we're really talking about the serial port. So, lets assign a variable, PORT, the value of the lowest, or Register 0 address. For the H-89 that's:

```
PORT=&0330
```

The &O (letter O) tells BASIC we're dealing with an octal number. For the HS-151, HS-158, and other PC clones we'd use:

```
PORT=&H3F8
```

This is a hexadecimal (base 16) number.

Now, if we want to send something to Register 0, we use the variable PORT as the address. To send a byte of data to Register 3 the address would be PORT+3. This way we can adapt our program to other machines by changing just one line, the one which sets the value of PORT. We can also direct our programming and data to different serial ports within a computer by reassigning the value of PORT.

We'll address one more bit of information we need to get this thing going this month, then we'll do some programming. I told you that if we set bit seven of Register 3 to a logic one, we could put special numbers into Register 0 and Register 1 to set the baud rate for the transmission and reception of data. So what are those special numbers? They must equal a number which when divided into the clock frequency (our clock is 1,843,200 Hz) yields a number 16 times the desired baud rate. To find the number, multiply the desired baud rate by 16. Then divide the result into 1843200. If we want to set the ACE to 1200 baud, we do it like this.

```
1200 * 16 = 19200
1843200 / 19200 = 96
```

If we write the decimal number 96 in binary, using two bytes (the numbers for some baud rates are greater than 255 decimal and thus need two bytes) it becomes:

```
00000000  01100000
```

If we convert each byte into octal numbers we get:

```
000  140
```

The Most Significant Byte, MSB, (000) goes into Register 1, so we'd send an octal zero to PORT+1. The Least Significant Byte, LSB, (140) goes to Register 0, so we'd send octal 140 to PORT. Let's see how this is done.

A check of our manual shows that the following steps must be taken to program the 8250 ACE:

1. Disable all UART interrupts
2. Set ACE in loop-back mode
3. Program the ACE as you want it
4. Read a character
5. Wait at least two character times
6. Read a second character
7. Take the ACE out of loop-back mode

With these simple steps, our programming will be done.

To disable the UART interrupts, we must set bits 0, 1, 2, and 3 of Register 1 to logic zero (see the table at the end of the article). Bits 4 through 7 are always zero, so we must pot the binary byte 00000000 into Register 1 (PORT+1). We'll use the BASIC command OUT. Checking your BASIC (MBASIC, ZBASIC, or GWBASIC) manual, you find that the OUT command takes a port address and a byte of information. We'll set our variable, PORT, and disable the interrupts.

```
1000 PORT=&0330      'FOR PC CLONES IT'S PORT=&H3F8
1010 OUT PORT+1,0    'SEND ZERO TO REGISTER 1
```

The 0 decimal is the same as 00000000 binary or 000 octal. The next step is to set the ACE into loop-back mode. Check the table at the end of the article. You'll see that bit 4 of Register 4 is called loop. We must set this to logic one by sending binary 00010000 to Register 4. (Remember to count the bits from right to left, starting with zero.) That's the same as decimal 16 or octal 20. Send it this way:

```
1020 OUT PORT+4,16  'WE COULD HAVE USED OUT PORT+4,&0020
```

Now, we'll program the ACE for 1200 baud operation. First, we'll set DLAB on (bit 7 of Register 3) with 10000000 binary, or 128 decimal, or 200 octal. Next, we'll put the numbers we calculated earlier into Register 0 and Register 1. Then we'll turn DLAB off.

```
1030 OUT PORT+3, 128      'TURN DLAB ON
1040 OUT PORT, &0140      'LSB TO REGISTER 0
1050 OUT PORT+1, 0        'MSB TO REGISTER 1
1060 OUT PORT+3, 0        'TURN DLAB OFF
```

I keep pointing out the decimal and the octal values of a binary number to make a point. Many programmers prefer one or the other for whatever reason they may have. Either number works as long as you tell your system what base it is. The important thing is to get the correct ones and zeroes in the correct positions in the binary byte before you make your conversion.

Thus far, we've programmed the speed. Now, we could also set word length, stop bits, parity, etc. When we do that in future articles, we'll have to take into account everything in the register we're writing to. To finish now, we've got to do the reading and delays. Let's finish our program to set the baud rate to 1200.

```
1070 X=INP(PORT)     'READ A CHARACTER FROM THE ACE
1080 FOR X=1 TO 20  NEXT X     'DELAY
1090 X=INP(PORT)     'READ ANOTHER CHARACTER
1100 OUT PORT+4,0    'OUT OF LOOP-BACK MODE
```

Next month, we'll see how to set some of the other things we can program into the ACE. We'll also look at the difference between programming, and setting a status within the ACE. We're getting close to some really exciting applications.

We've been through a lot of technical talk and data about the 8250 ACE. I know that makes for slow, boring, and sometimes difficult reading, but stick with me. You'll be glad you did.

See you next month.

	Register Address									
	0 DLAB = 0	0 DLAB = 0	1 DLAB = 0	2	3	4	5	6	0 DLAB = 1	1 DLAB = 1
Bit No	Receiver Buffer Register (Read Only)	Transmitter Holding Register (Write Only)	Interrupt Enable Register	Interrupt Identification Register	Line Control Register	MODEM Control Register	Line Status Register	MODEM Status Register	Divisor Latch (LS)	Divisor Latch (MS)
0	Data Bit 0*	Data Bit 0	Enable Received Data Available Interrupt (ERBFI)	"0" if Interrupt Pending	Word Length Select Bit 0 (WLS0)	Data Terminal Ready (DTR)	Data Ready (DR)	Delta Clear to Send (DCTS)	Bit 0	Bit 8
1	Data Bit 1	Data Bit 1	Enable Transmitter Holding Register Empty Interrupt (ETBEI)	Interrupt ID Bit (0)	Word Length Select Bit 1 (WLS1)	Request to Send (RTS)	Overrun Error (OR)	Delta Data Set Ready (DDSR)	Bit 1	Bit 9
2	Data Bit 2	Data Bit 2	Enable Receiver Line Status Interrupt (ELSI)	Interrupt ID Bit (1)	Number of Stop Bits (STB)	Out 1	Parity Error (PE)	Trailing Edge Ring Indicator (TERI)	Bit 2	Bit 10
3	Data Bit 3	Data Bit 3	Enable MODEM Status Interrupt (EDSSI)	0	Parity Enable (PEN)	Out 2	Framing Error (FE)	Delta Receive Line Signal Detect (DRLSD)	Bit 3	Bit 11
4	Data Bit 4	Data Bit 4	0	0	Even Parity Select (EPS)	Loop	Break Interrupt (BI)	Clear to Send (CTS)	Bit 4	Bit 12
5	Data Bit 5	Data Bit 5	0	0	Stick Parity	0	Transmitter Holding Register Empty (THRE)	Data Set Ready (DSR)	Bit 5	Bit 13
6	Data Bit 6	Data Bit 6	0	0	Set Break	0	Transmitter Shift Register Empty (TSRE)	Ring Indicator (RI)	Bit 6	Bit 14
7	Data Bit 7	Data Bit 7	0	0	Divisor Latch Access Bit (DLAB)	0	0	Received Line Signal Detect (RLSD)	Bit 7	Bit 15

* Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

Table I
8250 Registers

Word-Search Puzzle Generator

Leonard J. Oswald
115 N. Oak Street
Mt. Angel, OR 97362-0025

This article and the accompanying program came into being as a result of an original program and article appearing in the June 1981 issue of Softside magazine starting on page 92. This is a copyrighted program written by David W. Durkee for the Apple computer with translations for the Radio Shack TRS-80 and Atari computers by Jon Voskuil.

For those of you who have been around since the September 1983 issue of REMark, the above paragraph may look familiar. You are right. That was the exact same first paragraph of an article that created a lot of interest for the H/Z-89 using CP/M and MBASIC. I thought maybe it would be of interest to give you a version for use with the H/Z-110/120 and H/Z-150/160 machines.

First off, for those who had trouble with the H/Z-89 version, let me apologize for a small bug in line 10. After the words "PRINT CL$" — that should have been a semicolon and not a colon. How that gremlin got into the listing probably will never be explained, but this will give us a lesson to be used on all programs that have a syntax error . . . do not believe the listing in the magazine can not have an error. Thanks to a letter dated July 2, 1985 from Ray Hanna of Baltimore MD, I found another bug in the text on page 43 . . . line 120 should have read as follows:

```
L=INT(RND(1)*40)+1   U=INT(RND(1)*40)+1
```

This version will run on both of the newer Heath/Zenith computers. The main listing will be for the H/Z-110/120 and any lines with slight changes for the H/Z-150/160 will be included in the text of this article.

The many letters that we received from happy users and some with questions on how to get it running (most about the one error in the printed listing), were most gratifying. I found that many of the people who had problems were beginners. For those of us who have been around computing for a few years, remember that we should not lose sight of the fact that many people have had little experience using BASIC or that they have been using a BASIC written for some other machine. I have experience in using BASIC on ten different computers, and on one of these machines I have used 6 different versions of BASIC. I am sure that many people type in these programs they see in a magazine and give up when it does not work and especially when they type it in correctly as printed. I, for one, will answer all letters that I receive and will give as much help as I am able, but please send a self-addressed envelope with postage attached.

I have hundreds of programs that I have written, copied from magazines and books, been given by other hobbyists, some given to me by vendors for debugging and some that I have bought. I have the fun of continually debugging these programs to make them more user friendly and eye appealing when looking at the screen. When I do this for vendors, they do not always take into consideration all of my suggestions. Some of my suggestions take up memory when operating or on disks and they do not like this. For over six months, I have been translating IBM/Z-150 programs to work in the Apple Macintosh Microsoft BASIC for a vendor of Nutritionist and Market Analyzing programs.

Now to the business at hand. This version of the Word-Search Puzzle Generator is for the Heath/Zenith 100 series computer using Z-DOS or MS-DOS and ZBASIC with the Gemini 10x printer. The line numbers have been changed from the original version so you may use the AUTO feature of ZBASIC to enter the lines more easily. Other changes from the original include putting the title at the beginning of the printed puzzle, allowance for use of single sheet paper with the answers on one sheet and the puzzle and word list on another (or the back of the first sheet).

For the experienced programmer, the rest of the article might seem to be a little too fundamental, but please bear in mind that this might be the first program that someone will have ever typed into a computer. For the beginner, please keep in mind that most of the spaces used in the program lines are essential for proper syntax and operation. Some of the spaces are so that you may more easily read and edit the program.

Line 10 is used to remind me for which computer this program is intended and what was the last revision date. Lines 20 and 30 show me where I can find the articles that have been published in reference to this program. When this article is published, I will use line 40 to reference this publication date. After the apostrophe, which is used in place of the REM statement, I press the tab key to get the spacing. The tab key is used to get the spacing

after my initials. The tab spacing is not exactly the same in ZBASIC and GWBASIC (BASICA).

Line 50 sets up the string variable "E$", so that each time this is used the computer will think the ESCape code has been used and the following coding will be executed. The string variable "CL$" will create 79 spaces in a row, and will be used in the program to clear a line starting at the first column.

Lines 80 through 120 are specific printer codes for the Star Gemini-10x printer, and I believe they will work on all Gemini printers, but please check your manual to be sure. Line 80 will make the printer print each letter twice with the second time offset a half dot horizontally. Line 90 will make the printer print each letter two additional times, offset a half dot vertically and again a half dot vertically and horizontally. The codes in line 100 will allow the printer to print the normal sized letters, used in printing the puzzle and the answer, and line 110 will allow for a smaller letter printing that will be used for the Word List below the puzzle. Line 120 string will be used to reset the printer configuration to its original start-up settings.

By using the emphasized and double strike printing modes of the Gemini printer, the individual dots of the matrix printer will not show up in the newspaper, and the quality of letters looks more like it might have been done on a typewriter. Not all printers may have both of these modes, but they could use something similar. Experimentation could be in order for uses with other printers.

Line 140 was put in to make sure the program is in the normal start-up configuration for text and color.

Line 150 is used to give proper credits to those that have done the work and to show that this is a copyrighted program. The spacing in the listing is accomplished in the following manner. After the colon of each fiscal line, the line feed key was pressed and then the tab key was pressed. On the PC, series you will press the Ctrl-J keys (at the same time) to make the line feed. This spacing is not needed and only would be for appearance for checking to see if you had typed in everything correctly. The LOCATE will position the lines according to lines and columns on the screen.

Line 160 is just a delay loop to keep the title page on the screen for a few seconds and then lines 170 through 190 will clear the screen and put on the simple instructions and the menu lines. In line 200, we have a loop that will accept only a 1 or a 2 for an answer. If you input something else, you will hear a beep and then line 200 starts again.

In line 210, we will define that all numerical variables from A through Z will be integer numbers, that is whole numbers, and that W$, the string variable we will use to store the words we will be using, will be able to hold up to 200 different words. Usually, I use between 40 and 60 words only, because I use words from 8 to 12 characters long. You could change this figure to a smaller number, but don't feel that that is any advantage. The A array is dimensioned to hold a 24 wide by 20 down character array on the screen (this is a 20 wide by 24 down on the printer) and this could be changed to make the puzzle a different shape. The B array is set up only to hold the 8 possible directions that a word could possibly start out.

Line 220 is to set up the random generator with a different seed each time you use the puzzle. Line 230 clears the screen, line 240 starts the word number counter, and line 250 first locates line 22 and the first column and erases the line (79 spaces), then locates line 22 and the first column again to be ready for our input routine which is in lines 260 through 280. This type of routine eliminates any chance for repetition of identical puzzles.

If you eliminate 260 to 280 and put in a new 260 LINE INPUT A$ and you have two machines, say a Z-150 and a Z-110 with the same date and time and start running this program at exactly the same time then input exactly the same words in the same sequence, both puzzles will look identical to each other. Apparently, Microsoft used the exact same program in both ZBASIC and GWBASIC for random numbers, because when you use the exact same seed numbers and the same inputs they always come out the same. I feel the random generator that Microsoft used in the original Radio Shack TRS-80 Model I was much better, and I could not see any definite patterns.

Line 290 checks to see if an input of something larger than a capital letter Z has been entered, if so, then we shall assume that it is a lower case letter and that the CAPS LOCK key has not been pressed. This is not the best routine, but should work in most cases. If the CAPS LOCK key had not been pressed, then the 22nd line on the monitor will print the reminder to do so for a few seconds and then return to line 250 for a new input.

Line 300 will store our word in the W$ array. Line 310 checks to see if the word STOP has been input as this is the keyword to stop the input of words. If STOP has been entered, then the program will proceed to line 640.

In line 320, we will pick out two random numbers for positioning the word that has been entered, the L is for the line or row and the U is for the column.

The routines from lines 330 through 450 are nested loops to check to see if the selected position to start the puzzle is a blank or if a letter is present at this location. Line 380 will make sure the word will stay within the boundaries of the puzzle. Line 400 will check to see if the letter of our word and the intersecting letter of any previous word do not agree. If any test fails, the counter B is set to zero and line 450 sends back to line 320 to start over again with a new location.

Line 460 through 550 will select the direction in which to write the word and to see if it will intersect with another word. If all criteria is met, then the routine from lines 560 through 620 will print the word on our screen. If you selected the blank screen mode, then line 600 will be skipped, as this is the line that prints each letter on the screen.

What you see is not what you will get on the paper from the printer. The row of letters across the top of the screen will be the left hand column of the printout and left column of the screen will be the top row of the puzzle. The bottom row of the screen will be the right hand column of the printout and the right hand column of the screen will be the bottom row of the printout. In other words, on the screen the width of the puzzle is 24 letters, but the printout width will be 20 and the height of the puzzle will be 24 letters.

Line 630 sends us back to line 240 to get the next word. When we enter STOP as our word, the routine will come to line 640. The routine from 640 through 670 will check the array of our matrix and if no letter has been put in that location, then a dash will be placed in that spot.

Line 680 will prompt us to make sure that the printer is ready and that the paper is in the correct position. When everything is ready you only have to press RETURN.

Starting with line 690, the printer is set to emphasized and double strike printing. Line 700 prints the heading for the answer to the puzzle. Then line 710 goes to the subroutine at line 820 through 860 to print the puzzle answer on paper.

Line 720 again clears the 22nd line and gives you a chance to reposition the paper, or to allow you to remove and insert single sheet paper. The way I do it is to have the answer sheet on one side and the puzzle and word list on the other side. The puzzle is checked for the hyphens in line 740 through line 770, and all of them are changed to a randomly selected letter A through Z with line 760. Lines 780 and 790 print the heading for the finished puzzle.

In line 800, we go to the routine that prints the puzzle in lines 820 to 850 and returned by line 860. Line 810 sends us past the print routine to the routine for printing the word listing in lines 870 through 940. This is set up so a maximum column width of 18 characters is produced (for maximum of 16 character words) using 17 character-per-inch lettering produced by the Gemini 10x printer. This produces a four column list of the words used. So, if your forty-fifth word was entered as STOP, then you would have four columns with eleven words in each column.

The printer is restored to its start-up status in line 950. Line 960 allows you to make another copy of the printed puzzle and word list. If the answer is no, the program is ended, otherwise, we are prompted in line 970 to make sure the printer is ready to go again and then return to line 780.

For those with the PC computers, the following changes in the program are suggested. In lines 180, 680, 730 and 970 change <RETURN> to <Return>. In line 290, change <CAPS LOCK> to <Caps Lock>. Change line 140 to the following:

```
140 KEY OFF: WIDTH 80: SCREEN 0,0,0: COLOR 7,0,0
```

so if you were using the computer for something else before running this program, we will be in the start-up condition and the KEY ON condition is canceled. You might try COLOR 0,7,7 and see how you like this condition. If you do this, then you might want to add at the beginning of line 960, COLOR 7,0,0: before the END.

Happy computing.

Listing

```
10 ' Version  6/15/1985 by LJO  for Z-100 and Gemini-10X
20 ' Softside        June 1981       page 92
30 ' REMark          September 1983  page 41
40 '
50 E$=CHR$(27):CL$=STRING$(79," ")
60 '
70 '            Star Gemini-10x printer codes
80 EMPH$=E$+CHR$(69)          ' emphasized printing
90 DBLS$=E$+CHR$(71)          ' double strike printing
100 NORM$=E$+CHR$(66)+CHR$(1)
    ' 10 character per inch printing
110 SMAL$=E$+CHR$(66)+CHR$(3)
    ' 17 character per inch printing
120 CLRP$=E$+CHR$(64)
    ' clear printer to start up values
130 '
140 KEY OFF:SCREEN 0,0:COLOR 7,0
150 CLS:LOCATE 3,29,0:PRINT "WORD SEARCH PUZZLE":
      LOCATE 5,29:PRINT "by David W  Durkee"
      LOCATE 7,31:PRINT "translated by".
      LOCATE 9,29:PRINT "Leonard J  Oswald":
      LOCATE 11,29:PRINT "Copyright (C) 1981"
160 FOR I=1 TO 2500  NEXT I
170 CLS:LOCATE 3,29:PRINT "WORD SEARCH PUZZLE"
180 LOCATE 6,7:PRINT
    "To create a puzzle, simply type in a word after
     the '?' prompt,":PRINT TAB(7);
    "and press [RETURN]. When you are finished,
     enter 'STOP' as the":PRINT TAB(7);
    "last word, and the computer will do the rest."
190 LOCATE 11,7:PRINT "Please choose:": LOCATE 13,14
    PRINT "1 - for normal display during entry":
    PRINT TAB(14);"2 - for blank display
     (so that you can't see the puzzle)"
200 K$=INPUT$(1): K=VAL(K$): IF K<1 OR K>2 THEN BEEP:
    GOTO 200
210 DEFINT A-Z: DIM W$(200),A(24,20),B(3,3)
220 RANDOMIZE VAL(TIME$)+1/VAL(DATE$)+1
230 CLS
240 Z=Z+1
250 LOCATE 22,1,1: PRINT CL$:LOCATE 22,1:PRINT"WORD #";Z;
260 RANDOMIZE RND(9999):A$=INKEY$:IF A$="" THEN 260
270 PRINT A$;:LINE INPUT;B$
280 A$=A$+B$:IF LEN(A$)=1 THEN 250
290 IF A$=>CHR$(91) THEN LOCATE 22,1: PRINT CL$:
    LOCATE 22,1:PRINT"Press [CAPS LOCK] key down":
    FOR I=1 TO 2500:NEXT I:GOTO 250
300 W$(Z)=A$
310 IF A$="STOP" THEN 640
320 L=INT(RND*24)+1: U=INT(RND*20)+1
330 FOR X=-1 TO 1: FOR Y=-1 TO 1
340 IF X=0 AND Y=0 THEN 440
350 X1=L: Y1=U
360 FOR C=1 TO LEN(A$)
370 X1=X1+X: Y1=Y1+Y
380 IF X1>24 OR X1<1 OR Y1>20 OR Y1<1 THEN B(X+2,Y+2)=0
    GOTO 440
390 IF A(X1,Y1)=0 THEN 420
400 IF A(X1,Y1)<>ASC(MID$(A$,C,1)) THEN B(X+2,Y+2)=0
    GOTO 440
410 B(X+2,Y+2)=B(X+2,Y+2)+1
420 NEXT C
430 B(X+2,Y+2)=B(X+2,Y+2)+1: B=B+1
440 NEXT Y,X
450 IF B=0 THEN 320
460 R=2  D=2
470 FOR X=1 TO 3: FOR Y=1 TO 3
480 IF B(X,Y)>B(R,D) THEN R=X: D=Y
490 NEXT Y,X
500 X=R-2: Y=D-2
510 IF X=-1 AND Y=-1 AND B(1,1)=1 THEN 530
520 GOTO 550
530 X=INT(RND*3)-1: Y=INT(RND*3)-1
540 IF (X=0 AND Y=0) OR B(X+2,Y+2)=0 THEN 530
550 X1=L: Y1=U
560 FOR C=1 TO LEN(A$)
570 X1=X1+X: Y1=Y1+Y
580 A(X1,Y1)=ASC(MID$(A$,C,1))
590 IF K=2 THEN 610
600 LOCATE Y1,X1*2:PRINT CHR$(A(X1,Y1));
610 NEXT C
620 B=0: FOR X=1 TO 3: FOR Y=1 TO 3: B(X,Y)=0: NEXT Y,X
630 LOCATE 22,1:PRINT CL$;: GOTO 240
640 FOR X=1 TO 24: FOR Y=1 TO 20
650 IF A(X,Y)<>0 THEN 670
660 A(X,Y)=45: LOCATE Y,X*2:PRINT "-";
670 NEXT Y,X
680 LOCATE 22,1:PRINT CL$;:LOCATE 22,1:LINE INPUT
    "Position paper and press [RETURN] ",K$
690 LPRINT EMPH$+DBLS$
700 LPRINT" ": LPRINT TAB(6),
    "WORD-SEARCH PUZZLE ANSWER KEY"
710 GOSUB 820
720 LOCATE 22,1:PRINT CL$;:LOCATE 22,1:PRINT
    "Please wait a few moments for me to create
     the puzzle .      ";
730 LOCATE 22,1:PRINT CL$;:LOCATE 22,1:INPUT
    "Position paper and press [RETURN]",K$
740 FOR X=1 TO 24: FOR Y=1 TO 20
750 IF A(X,Y)<>45 THEN 770
760 B=INT(RND*26)+65: A(X,Y)=B
```

```
770 NEXT Y,X
780 LPRINT " ". LPRINT TAB(11);"COMPUTER-GENERATED"
    LPRINT TAB(11);"WORD-SEARCH PUZZLE"
790 LPRINT TAB(11);"by Leonard J  Oswald"
800 GOSUB 820
810 GOTO 870
820 LPRINT " "
830 FOR X=1 TO 24: FOR Y=1 TO 20
840 LPRINT CHR$(A(X,Y));" ";
850 NEXT Y  LPRINT " "  NEXT X
860 RETURN
870 LPRINT " ": LPRINT " WORD LIST:"
880 LPRINT SMAL$
890 CT=0
900 FOR I=1 TO Z-1
910 LPRINT TAB(CT);W$(I);
920 CT=CT+18
930 IF CT=72 THEN LPRINT " ":CT=0
940 NEXT I
950 LPRINT NORM$
960 LOCATE 22,1:PRINT CL$;:LOCATE 22,1:INPUT
    "WOULD YOU LIKE ANOTHER COPY",K$:
    IF LEFT$(K$,1)="N" THEN LPRINT CLRP$:END
970 LOCATE 22,1.PRINT CL$;:LOCATE 22,1.INPUT
    "ADVANCE PAPER AND PRESS [RETURN]",K$: GOTO 780
980 END
```

On The Leading Edge

MS-DOS Command Piping And I/O Redirection Part 1

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

This last month has proved to be very interesting in a number of ways. It has been busy, enlightening, and expensive. The busy part has simply been a matter of wrapping up some of the book projects that I have been working on. That process of proofreading and making corrections always is enlightening, as well as time consuming. And it was expensive because the power supply on my trusty '100 decided to play the "Rice Krispies game" — snap, crackle, and POP! More on that later.

The subject for this month in the general information series is a discussion on the use and capabilities of I/O (Input/Output) Redirection. While this may be a rather new feature of MS-DOS — originally available in Version 2 and later — it has been available in the Unix operating system for years. In addition, the availability of I/O Redirection in the MS-DOS (and PC-DOS) environment is perhaps an indication of the increasing popularity of the C programming language. C was used to develop Unix, and it is impossible to separate one from the other. Indeed, the C language supports the use of I/O redirection, and all good C compilers for MS-DOS support this feature.

The only requirement for being able to use I/O redirection is that you must have MS-DOS (or PC-DOS) Version 2 or later. For the Heath/Zenith computers, this applies to MS-DOS for both the '100 and the '150 PC series, as well as the new '241. Although these features also apply to IBM PC-DOS Version 2 and later, there are some well documented and published bugs in IBM PC-DOS Version 3.0 which will not allow the proper use of some of the I/O redirection features. The end result is that, if you are using a Version 3.0 of PC-DOS, some of the features may not work as described in this article. That is a problem with the DOS which I understand has been corrected in PC-DOS Version 3.1. And that is also one of the reasons that I highly recommend the use of the Heath/Zenith MS-DOS. Heath and Zenith have an extensive software testing program that make it highly unlikely (though not impossible) that a standard DOS feature will not work in a production release of operating system software.

In a recent issue of PC_Week, the editors criticized IBM for an incredibly bad software update policy. According to the editorial, DOS corrections are handled by dealers. That means that you must take your distribution disk(s) to a dealer in order to obtain the corrected DOS, such as in the Version 3.0 case. Even worse, there is no general information about the correction that notifies registered owners that there is a problem in the first place. That means that you have to KNOW that there is a problem, locate your distribution disks, and trudge to the dealer to get the correction. It is easy to criticize Zenith for their software update policy, but the fact remains that they do a better job than IBM.

Although the Zenith updates may be slow at times, it is clear that a concerted effort is made to provide update notification (and occasionally free updates) to registered users of Heath/Zenith software. And although Zenith does make occasional mistakes in software registration, they do have a good process. IBM makes no such attempt to inform owners of DOS corrections, let alone the idea of providing ANY free updates. Since I have owned 3 different versions of PC-DOS (1.10, 2.10, and 3.0), I have not received any notifications of corrections from IBM. In fact, there is not even a "mail-in" registration card with the DOS for registration purposes. I have to conclude that the dealer is the only one who knows my name and address for software registration purposes. But, on to I/O redirection...

I/O Redirection — Introduction To Device Drivers

In order to understand something about I/O Redirection, we must as usual, look at some basics. In MS-DOS Version 2 and later, information is transferred between devices through the use of device drivers. The DEVICE DRIVER is simply a software interface between the operating system software and various kinds of computer hardware. There are two kinds of device drivers: block and character.

The Block Device Driver transfers 512 bytes of data at a time and is usually identified by a single-letter name. The obvious use of

block device drivers is the interface for disk drives identified by A, B, C, and so on. It is no accident that a block device driver uses a block size of 512 bytes which is also the standard sector size for disks in MS-DOS systems. Zenith provides a special block device driver with their operating systems which can be used to define a so-called RAM (Random Access Memory) disk drive. Depending on whether you have a '100 or a '150 PC series computer, it is usually identified as something like VDISK.SYS or MDISK.DVD and is implemented by adding the appropriate file name to the DEVICE= parameter in the CONFIG.SYS file.

Character Device Drivers transfer data on a character-by-character basis. They are usually identified by device names consisting of from one to eight characters and conform to the standard MS-DOS file naming conventions. Virtually all MS-DOS implementations contain standard device drivers such as AUX, CON, COM1, LPT1, and so on. The actual name of the device driver will vary depending on the specific hardware, and there are some differences in device names between the '100 and the '150 PC series. You should refer to your operating system documentation or my Zenith/Heath MS-DOS (Z-DOS) FlipFast book (page 18) for a description of the device names for your system. We will return to these device names a little later in this article since they are important to an understanding of I/O Redirection.

Some Definitions And Terminology

There are really three definitions that we must understand in order to use I/O redirection, although MS-DOS only effectively provides for the use of two of these features from a command perspective. Standard device drivers, available in Unix and most C language implementations, incorporate a standard input (called STDIN), a standard output (called STDOUT), and standard error output (called STDERR). MS-DOS uses the same approach in the definition of device drivers. More importantly, we can dynamically redefine the STDIN and STDOUT using appropriate symbols and command syntax in a standard DOS command.

The default Standard Input (STDIN) device is defined as the standard source of input characters. The default is the keyboard or terminal. Note that a keyboard is one part of a terminal such as the Zenith Z-49 terminal. Unless otherwise specified, MS-DOS will always expect input from the keyboard. That is the normal case when you enter a DOS command.

The default Standard Output (STDOUT) device is defined as the standard destination for output characters. The default is the CRT or terminal. Note that a CRT is the other main part of a terminal. The only reason for mentioning the parts of a terminal is that most references only mention a terminal. I have found it useful to be more specific in terms of the keyboard or the CRT. Unless otherwise specified in one of several ways, MS-DOS will always send all output to the CRT.

The default Standard Error Output (STDERR) is defined as the standard destination for error messages. Like the STDOUT, the default is the CRT or terminal. Unless otherwise specified, MS-DOS will always send all error messages to the CRT.

One of the advanced features of MS-DOS Version 2 and later is that you can dynamically redefine STDIN and STDOUT. Dynamic redefinition means that you can change the STDIN (from the default keyboard) and/or the STDOUT (from the default CRT) by using the appropriate command syntax and several special symbols.

There are two distinct ways to redefine STDIN and STDOUT: Command Piping and I/O Redirection.

In Command Piping, the output of one command (or program) is most often redirected or "Piped" to the input of another command (or program) although it can be used with devices too. When piping is used with devices, however, you must understand the characteristics of that device. Strange things can happen if you try to pipe the STDOUT of a printer (e.g PRN) to another output device. More on that later when we look at some examples. Piping allows the STDOUT of one program or device to be used as STDIN for another. The primary characteristic of the piping is that you can have more than one pipe symbol (vertical bar) in a command line. Piping is primarily used with some special programs, usually called filters, which we will look at in a minute.

In I/O Redirection, the STDIN can be redefined so that the system obtains input from a specified device other than the default keyboard. In addition, STDOUT can be redefined so that the output is sent to a device other than the default CRT. The primary characteristic of Redirection is that you can normally redirect STDIN and STDOUT only once in a command line.

Although piping is really another form of I/O Redirection that is part of the standard definition, I always like to think of them separately. The primary reason for the two definitions is that a command line may contain more than one pipe; however, redirection typically allows the STDIN and STDOUT to be redefined only once in a command line. Another reason for separating the two is that command piping is indicated by the vertical bar (|) and I/O Redirection is indicated by the appropriate use of the greater than (>) and less than (<) symbols. Technically speaking however, the normal definition of I/O Redirection includes Piping and I/O Redirection for both programs and devices, as well as all of the special symbols.

MS-DOS Piping

A pipe, indicated by the vertical bar (|), routes or "pipes" the STDOUT of one command to the STDIN of another. That is, the STDOUT from the command to the left of the pipe is used as input to the right of the pipe. The command line, including the pipe symbols (|) and spaces, may contain any number of commands up to the maximum command buffer limit of 127 characters including spaces. The general form of a pipe command line is:

```
command1 | command2 | command3.
```

Piping allows you to chain a group of commands with a specified redirection of STDIN and STDOUT. I have often thought it odd that DOS did not provide a directory listing sorted in alphabetical order by file name. The use of some standard DOS commands illustrates how you can still obtain a sorted directory listing, however, by using the following command line:

```
DIR | SORT
```

The DIR command normally displays a disk directory on the CRT (STDOUT). In this case, we are taking that STDOUT and piping it (through the use of the vertical bar) to the STDIN of the SORT program. The SORT program then performs its function and sends the output to STDOUT. Since there are no other specified pipes, we will see the sorted directory listing on the CRT.

We can extend this example by piping the STDOUT of the sort program to a standard system device such as a printer by using the following command line:

```
DIR ¦ SORT ¦ PRN
```

This command line performs exactly like the previous example except that the STDOUT of the SORT program is piped to the STDIN of the PRN device driver for the printer. It is important to understand that a printer can receive data (i.e. STDIN), but it cannot output (transmit) data to STDOUT. A printer's output is obviously paper which cannot be redirected or piped.

In summary, pipes may be used with any command that normally takes input from the keyboard (STDIN) and sends output to the CRT (STDOUT).

Piping And Filters

Standard implementations of MS-DOS Version 2 and later, including all versions of Heath/Zenith MS-DOS, include some special programs called filters. A Filter is generally defined as a program that reads data from the STDIN (default keyboard), changes it in some specific way, and then writes the modified data to STDOUT (default CRT). Filters are normally used with other commands and can be linked together to provide unusual flexibility in performing various data-related operations. Standard MS-DOS filters include:

- FIND: Searches for ASCII characters in STDIN.
- MORE: Displays STDIN one screenful at a time.
- SORT: Sorts STDIN and sends output to STDOUT.

Filters are usually combined with pipes to obtain results that are not available with standard commands. We have already seen how multiple pipes may be used to perform some interesting operations in the "DIR ¦ SORT ¦ PRN" example. Similarly, we can also combine the filters to perform useful operations such as the following:

```
DIR ¦ SORT ¦ MORE
```

We already know that the "DIR ¦ SORT" portion provides a sorted directory listing to the standard output (STDOUT) CRT. The additional pipe takes that sorted STDOUT (from SORT) and pipes it to the MORE filter which stops the CRT display every 24 lines until we press a key to continue.

Some Additional Information On Filters

If you know anything about the C programming language, it is almost a trivial exercise to write your own filters. You can easily write filters to perform needed functions such as stripping the high order bits from WordStar document files, as well as many other useful functions. That is one specific reason that I prefer C over any other programming language.

My favorite C compiler is the Computer Innovations C86 Compiler (CI-5063-1) that is available from Heath. Although the Heath catalog listing is somewhat misleading if you don't read it carefully, this compiler works on both the H/Z-100, as well as the PC series versions of MS-DOS. Jim Buszkiewicz, one of HUG's intrepid software developers, also tells me that he has used the C86 Compiler on his '100 with no problems.

One of the reasons that this is my favorite compiler is that the manual, which has been rewritten for Heath, contains some very humorous comments. Most manuals, particularly those dealing with very dry subjects such as assemblers and compilers, tend to be packed with facts, but they are very boring reading. This manual is different. In fact, my favorite sentence in the entire manual occurs on page 1-1: "The compiler can be run on a single drive Z-100 PC, but this may be hazardous to your sanity". In my personal experience, technical writers usually receive more criticism than kudos no matter how carefully they research and document a subject. The technical writer(s) of this manual should be applauded for their ability to inject humor into a very important point.

If you are not interested in programming your own filters, the Programmer's Utility Pack (PUP) provides an easy way to get some very useful utilities and filters. The PUP contains, among other things, 22 utilities and filters, source for the BIOS, the BSE editor, and an incredible amount of technical information on MS-DOS. The PUP for Version 2 is a single 3" three ring binder. The PUP for Version 3 has grown to two binders: a 3" binder and a 1-1/2" binder. While the PUP is not as inexpensive as the IBM Technical Reference, the Version 3 PUP contains at least 4 times the information plus all of the utilities, BIOS source, and other goodies. Even if you have an IBM PC, you will find that the PUP is much more useful than its IBM counterpart. If you have any interest in the technical features of the Heath/Zenith MS-DOS (or the utilities) at all, the PUP is an outstanding reference that is highly recommended.

MS-DOS I/O Redirection — To Be Continued . . .

As I was finishing up this article with the original intent of including a discussion of both Command Piping and I/O Redirection, it became apparent that the entire article was approaching book length. Because of the length, it was obvious that it was appropriate to divide the article into sections related to Piping with a follow-on article on I/O Redirection. The remainder of the discussion on MS-DOS I/O Redirection will appear in a future issue of REMark.

96 TPI Drive Support On The Z-100

One of the more interesting questions that has recently arisen concerns 5.25" 96 tpi (Tracks per Inch) drive support on the Z-100. There is some confusion as to whether or not the Z-100 really supports 5.25" 96 tpi drives. Yes, the Z-100 can support 96 tpi drives, as well as 8" drives. For those of you interested in the technical explanation, there are several requirements for drive support in any computer system.

There are three fundamental requirements for the support of ANY disk drives in ANY system. The first is software support which is usually part of the Basic Input/Output System (BIOS — IO.SYS, for MS-DOS) in any operating system. The second requirement is that the disk controller hardware must be capable of controlling the physical disk drive hardware. And of course, the last requirement is that you must have the appropriate disk drive — a 48 tpi drive is not hardware capable of dealing with a 96 tpi disk; you must have a 96 tpi drive.

If you have the MS-DOS Version 2 Programmer's Utility Pack (PUP), you can easily verify that BIOS support is available. Take a look at BBLKDEV.ASM (DSK_BPB1: label) on the Z-100 BIOS disk. This is the source for the block device driver for all of the possible Z-100 disk drives.

The Z-207 disk controller in the '100 contains a DIP switch, DS1. Section 0 of that switch determines whether 48 or 96 tpi drives are supported. If that section is set to zero, that's the normal 48 tpi drive. If set to one, it's 96 tpi. This is unique to the Z-100. It is unfortunate that the Z-100 will not support both 48 AND 96 tpi drives in the active system. You must choose one or the other and set the DIP switch accordingly.

I have received more than one letter from Heath/Zenith users about why their system will not "support" 96 tpi drives. In some

cases, I found that these users simply did not realize that there is more to it than just flipping the switch on the Z-207 controller — they were trying to use 48 tpi drives when the disk controller was configured to 96 tpi drives. In other cases, users were trying to "mix" 48 and 96 tpi drives in the active system — that does not work because of the DIP switch on the controller. In all cases, I was getting reports of some strange things that happened to people when they were using the system. That was usually in the form of "my 48 tpi (or 96 tpi drive depending on the Z-207 switch) drive will not work".

Those are the facts about 96 tpi drive support on the Z-100. But it is interesting to speculate on WHAT started the rumor about the lack of 96 tpi support.

Zenith has taken the public position, since the introduction of the H/Z-100, that 96 tpi drives are not officially supported. Note the word "officially". My guess is that this statement was probably interpreted by a number of people to mean: "96 tpi drives CANNOT be used with the H/Z-100". We can easily verify that the Z-100 supports 96 tpi drives, so we know that interpretation is false. I believe that the real interpretation of that statement should be something like: "You can use 96 tpi drives on the Z-100 system if you wish, but Zenith will not guarantee future software support of 96 tpi systems."

Perhaps a more interesting speculation is WHY Zenith chose to adopt that official position. Consider some history. The IBM PC and the H/Z-100 were both announced around 1982. There were all kinds of rumors floating around about the IBM PC, although it was a good bet that IBM would use "off the shelf" hardware for the PC. It was a fact that IBM had used eight inch drives on their dedicated word processing systems for years. Both 48 and 96 tpi floppy drives were readily available, not to mention hard disks.

Support for all of these drives was wisely designed into the Z-100. Perhaps we can thank Barry Watzman for that. As everyone now knows, IBM elected to use 48 tpi drives in the PC which became the de facto standard. The Z-100 could support anything, and by the quick flip of a switch, 48 tpi drives also became the Z-100 "standard". By the way, some industry pundits claim that IBM used the 48 tpi drives because 96 tpi drives were "not reliable" which I think is absurd. I used the Z-37 drives on my H-89 system with absolutely no reliability problems.

Well OK, 96 tpi drives are not "officially" supported by Zenith in policy, but they are currently supported by existing Z-100 hardware and software. The point is, if you have a problem with a 96 tpi system, you are on your own in terms of getting it solved. But there is at least one other interesting issue related to Z-100 96 tpi support and PC emulation on the Z-100.

The Gemini And The Easy PC

I have seen a number of complaints lately that the Gemini and the Easy PC do not support the 5.25" 96 tpi drives or 8" drives. As that statement is written, it is not strictly true. In my January article, I noted specifically (in the first paragraph on page 94) that "Other limitations on the Gemini are basically those imposed by the IBM PC. The most significant is perhaps the lack of support for 8 inch drives." Based on recent developments, it appears that my description was not specific enough.

First, and foremost, it is a fact that the IBM PC (and the H/Z-150 PC series) does not support 96 tpi drives period. It never has. The key point is that 96 tpi drives are NOT supported by the IBM PC (or Z-150 PC) BIOS (among other reasons — see above). If you have a Z-100 system, and intend to add a Gemini or Easy PC emulator, you MUST CHANGE THE 96 TPI DRIVES TO 48 TPI DRIVES. This is not a problem with the Z-100, Zenith's MS-DOS, PC-DOS or either of the emulator boards. This, I suspect, is one of the reasons that 96 tpi drives are not officially supported. The fact that the Z-100 hardware supports the 96 tpi drives is just not enough.

It is also true that IBM PC-DOS and Zenith's Z-150 PC MS-DOS do not support more than two floppy disk drives. Again, that is a specific limitation on the IBM PC which has been faithfully emulated by the Z-150 PCs, the Gemini, and the Easy PC. Although this is primarily due to a software restriction in the BIOS, the fact remains that no more than 2 floppy disks are supported by the IBM PC, the Z-150 PCs, the Gemini, and the Easy PC in the emulation mode.

And the final fact, as mentioned in my January article, is that neither the Gemini nor the Easy PC support 8" drives in the emulation mode — just like the IBM PC and the Z-150 PCs.

In summary, the IBM PC does not support 96 tpi drives, more than two 5.25" floppy disk drives or 8" drives. Neither do the Gemini nor the Easy PC in the PC EMULATION mode; however, you can have more than two floppy drives (including 8" drives) connected to the Z-100 system when you are in the PC emulation mode. You simply will not be able to use the "excess" drives in the PC emulation mode, but you WILL be able to use them in the standard Z-100 mode. The real problem is that you cannot use 96 tpi drives at all in the PC emulation mode — you must use 48 tpi drives.

The Gemini Review — A Correction

In the January article, Listing 4 — List of Z-150/IBM PC Tested Software For This Review — correctly reports that I tested the COPYIIPC software on the Gemini. Unfortunately, I omitted a double asterisk (**) note on the listing referring to the text and also omitted the corresponding text in the article. This left the erroneous impression that COPYIIPC runs on the Gemini which is not true. COPYIIPC does NOT run on the Gemini. The following is the text that I failed to include in the Gemini Review. It should have been included as the last paragraph under the "Running PC Software" heading.

Since most of you know that I am particularly unhappy with copy protected software, I purchased the COPYIIPC program which can override many forms of copy protection. I tested COPYIIPC with several copy protected programs, and found no problem in copying protected software on a Z-150. When I tried to run COPYIIPC on the Gemini, however, strange things happened — usually an error message indicating that the software could not be copied. From what I was able to tell, COPYIIPC goes directly to the disk controller when it is overriding the copy protection. Unfortunately, the disk controller on the Z-100 is significantly different from the IBM PC, and because of the hardware differences, COPYIIPC failed.

I don't consider that a particularly significant problem only because I do not recommend or buy any copy protected software. Many vendors are finally beginning to respond to market pressure against copy protection. You can easily prove that to yourself by looking at any number of software ads that boldly proclaim "NOT COPY PROTECTED" as part of the ad.

As far as the Gemini is concerned, however, I was able to run all of the software that was copy protected as long as I did not use

the COPYIIPC program. Programs like dBase III, Framework, and Microsoft Chart seemed to run fine with no problems.

On The Home Front

Things on the home front have been rather chaotic of late. The biggest news, reported earlier, is that the power supply on my '100 decided to play the "Rice Krispies game". The power supply "popped" when I simply turned on the power on a Saturday afternoon. After some difficulty in locating a replacement heart (power supply), I was finally able to pick up the "patient" at the computer hospital (the Dallas Heathkit store) on Wednesday afternoon after a few days in the Intensive Care Unit.

The good news is that the heart transplant was successful, and to date, the patient has exhibited no signs of rejection. I have to admit that I was more than a little concerned that the sudden heart stoppage might also have caused a frontal lobotomy (or worse). I was not looking forward to performing a partial or complete brain transplant on the patient who was recovering from the heart problem. The bad news is that the "operation" was considerably more expensive that I had imagined.

Special thanks to Don Murray and his staff at the Dallas store for all of their help. Their excellent support is a definite asset to the Veritechnology Electronics Corporation. And from my own personal experience, many other Heathkit stores also provide excellent support which is the primary reason that I like Heath/Zenith computer equipment. That is also the reason that I recommend Heath and Zenith hardware and software to my friends.

Next Month

As I was finishing this article, I received the UCI Easy PC emulator, so we will be taking a look at that next month. That promises to be an interesting article. The following article will discuss the Gemini, the Easy PC, and HUG's ZPC Emulator with a specific comparison. Following that, we will return to the remaining discussion on I/O Redirection if things go as currently planned. Since things do change, there may be some surprises in store for all of us.

If you have any questions about the information in this article, please let me know. As usual, you should enclose a stamped, self-addressed envelope if you would like to have a personal reply.

Products Discussed

MS-DOS Version 2 Programmers Utility Pack (CB-5063-16)	$150.00	
MS-DOS Version 3 Programmers Utility Pack (CB-5063-30)	$150.00	
C86 C Compiler (CI-5063-1)	$395.00	
Easy PC Emulator (PC-250)	$699.00	
Z150 MS-DOS (OS-63-31)	$ 90.00	(when purchased with PC-250 only)
Gemini Emul. Brd (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	

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)

*

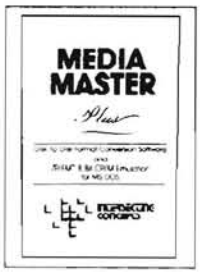
"MAKE YOUR Z-150 CP/M COMPATIBLE"

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!*

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*

Mouse Pack: A Review

Jerry Furst
17049 Vista Bluff
San Antonio, TX 78247

I recently had an opportunity to review a software product that I have been looking forward to, a mouse driver for the Z100. The product is called MOUSE PACK by Paul F. Herman (the Author of DOODLER), and was included in a combination pack that included a Logitech Logimouse R-7, version 3.05 of DOODLER for the Z100, and the MOUSE PACK. DOODLER has been previously covered in a recent issue of REMark, and I won't repeat what was said about it, other than to say that it is a fine graphics package. It works quite well with the mouse.

The documentation that came with the combo pack included three separate documents. The instructions for DOODLER, for the Logimouse, and for the mouse driver. The documentation for the R-7 and the accompanying software are, unfortunately, for the PC. This software includes a mouse driver for the Logimouse, a set-up routine that allows you to change the Logimouse protocols to act like any one of five different mice (including MicroSoft mouse), and a text and graphics 'mouse' program. This, of course, has been the frustration of Z100 users, the serial UART's used by a PC and a Z100 are different and a mouse driver written for the PC will not work on a Z100. Worse still, the Text and Graphics program (called TAG) written by Logitech are written for their mouse driver only. Basically, we loose out on all counts for the Logimouse software, but don't be discouraged, there is one that works. Another disappointment came after about 1/2 hour of operation, the mouse died! Don't take this to be an example of the Logimouse's quality though, I typically have the kind of luck where if the failure rate is 1 in a million, I buy the millionth one.

The Logimouse R-7 is also compatible with the R-5, and I had one of those sitting in my desk (gathering dust), so I decided to try it out. The Logimouse (both R-5 and R-7) has 3 buttons instead of 1 or 2. The mouse contains optomechanical movements that were designed in Switzerland and has the highest resolution available in a mouse (200 increments per inch). The mouse contains a microprocessor to handle the serial communication between mouse and computer. It will track accurately, even on a glass surface. Resolution and mouse formats are user selectable (through serial communications). The big difference between the R-5 and the R-7 is that the baud rate is software selectable on the R-7, while the R-5 must be set with jumpers inside the mouse.

The MOUSE PACK comes with 2 versions of the mouse driver. One is called PCMOUSE and the other is called MSMOUSE. The PCMOUSE driver supports the Logitech mouse and the MSMOUSE is for a MicroSoft Serial mouse. Both of these programs are mouse drivers for the Z100. The program was evaluated on MS-DOS 2.19 using a Z100 with 768K operating at 8MHz. The MP (MOUSE PACK) also contains the source code for the mouse driver, and that isn't something that you typically find in software anymore! You have the ability to reconfigure the software to anything you want. This would be especially handy if you had more than one computer and wanted mouse drivers for both computers, that is a PC and a Z100, for instance.

The mouse driver is installed by typing the program name (either PCMOUSE or MSMOUSE), followed by optional parameters. These parameters allow you to reconfigure the mouse to meet your particular needs. As an example, when using WatchWord with the mouse (WOW!), I program the left button to emulate the I CHR key, the middle button for the HELP key, and the right button for the ENTER key, but in MultiPlan, I would want a different set of functions available, like a space bar for instance. Each mouse key can be redefined to any Z100 key (shifted or unshifted) including control characters. A complete list of these are available in the back of the accompanying documentation. The port can be configured to either J1 or J2, depending on your RS-232 configuration for the mouse.

The sensitivity of the mouse is also programmable. While in a graphics package, such as DOODLER, we may want our sensitivity to be extremely high, but in a word processor, we want our movements to be more liberal. With the R-5 (or R-7) a

resolution of 200 increments per inch can work against you, especially if you are trying to draw a straight line, in a word processor. Any slight up and down movements would cause the mouse to jump up and down lines of text. There is also a variable damping factor that can be set with the MP. This damping factor can be set to assist with keeping your mouse "straight". Let's look at some examples.

```
PCMOUSE XS5 YS2 XD100 YD100 LB151 MD152 RB153 SP1
```

This would set the mouse for a step sensitivity of 5:2, with every 5 (X) increments moving the cursor 1 position left or right, and every 2 (Y) moving the cursor 1 position up or down. The damping factors of 100 will track precisely the relative X-Y movements of the mouse. The left button is programmed for f1 function key, middle button is f2, and right button is f3. The port the mouse is connected to is J1. This is a VERY sensitive mouse configuration, and would most likely be used while using a graphics package. The current mouse configuration can be interrogated at any time by typing the program name [space] ?? e.g. PCMOUSE ??. The default configuration is XS2 YS2 XD10 YD10 LB151(f1) MB152(f2) RB153(f3) and port J2. The mouse parameters can be given in any order and can be changed at any time (batch files to set up different mouse configurations will be a snap).

After the ill-fated demise of the Logimouse R-7, I grabbed the R-5 and began to try it out. Initially I was unsuccessful because the baud rate of the R-5 is set by internal jumpers to 9600 baud. The digitizer option with auto-cad operates the Logimouse at 9600 baud, so evidently Logitech assumed that this mouse would be used by auto-cad'ers. My first attempt was to reconfigure the J2 port to 9600 baud and try it, and again it didn't work. I haven't been able to figure out why except that the mouse driver may be unsuccessful at intercepting the increment reports at that high rate of speed.

The only other option left open to me was to open the Logimouse and readjust the baud rate. This is a relatively simple matter. There are four jumpers under the cover of the Logimouse in the lower corner of the board. Two of these jumpers are cut at the factory for 9600 baud. All four of the jumpers connected is 1200 baud. Adding these jumpers again yielded success with the Logimouse and I was on my way. I don't like pulling the cover off of anything to reset baud rates, and jumpers have a nasty habit of falling out, especially if the unit is stored in a desk, so I added a switchable baud rate to the R-5. This is a simple matter of adding a small Double Pole Single Throw (DPST) switch to the logimouse. There is sufficient clearance inside the mouse for this additional switch. The switch should be wired so that with the switch in one position, pins 3 and 4 (see schematic below) are connected to the common bus, and with the switch in the opposite position, pins 3 and 4 are opened from the common bus.

If you elect to incorporate this modification into your Logimouse R-5, then there is one additional comment I must make. When the Logimouse is plugged into it's power supply, the configuration as determined by these jumpers is set. Changing the configuration with power on the mouse will NOT change the baud rate. First, remove power from the Logimouse, then reposition the switch to change the baud rate. Then plug the Logimouse power back in. I had no loss of performance from the mouse at either 1200 or 9600 baud using this mod. Remember also that this modification is not required for the R-7, as the baud rate is software selected (including an auto-baud rate selection).

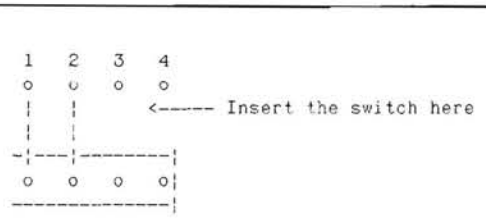


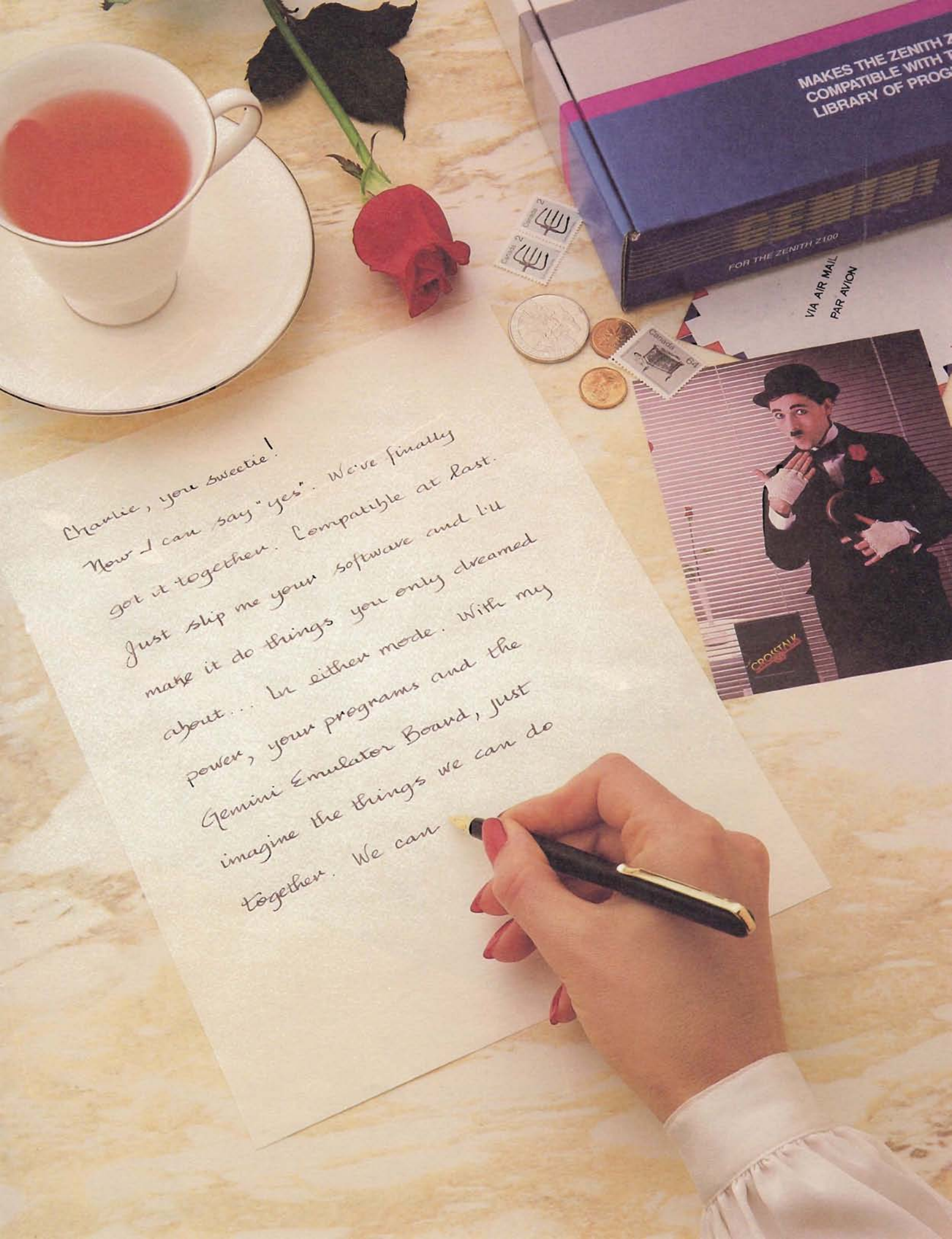
Figure 1

The documentation for the MOUSE PACK is well written. All of the topics discussed in this article are covered in depth in the documentation. There are examples of configurations, as well as the key-code charts for the Z100 and the Z150 as previously mentioned. The combination pack, including the MOUSE PACK, DOODLER, and the Logimouse R-7 are currently selling for $195.00. The MOUSE PACK software is, I understand, available separately for $29.95.

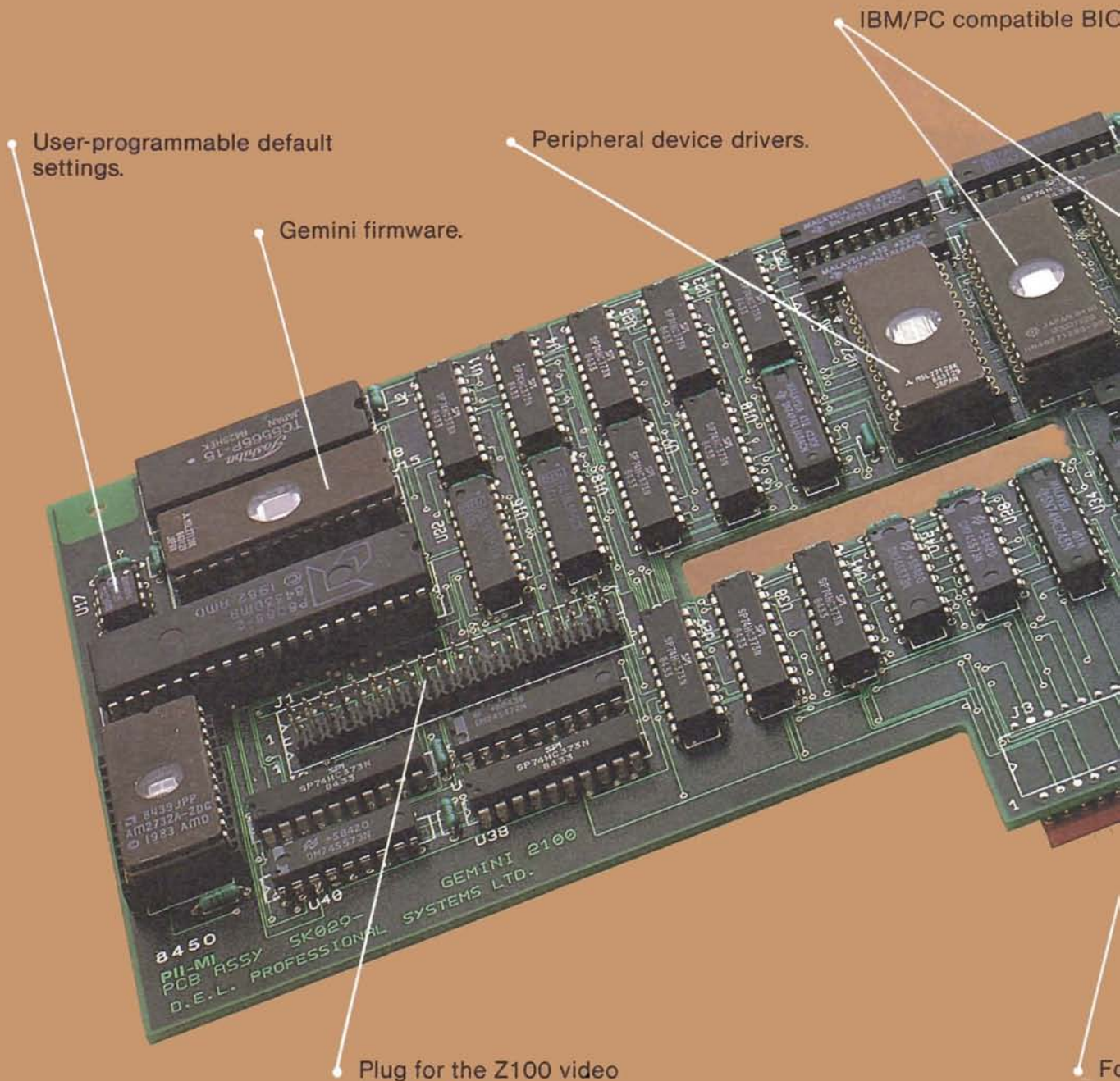
Conclusion: This is some very fine software! The mouse operates superbly and is a pleasure to use, as well. My only question is why it wasn't done sooner?!

MOUSE PACK and DOODLER
Copyright by
Paul F. Herman
Software Graphics Tools
BP.O. Box 206
Safety Harbor, FL. 33572

Logimouse is the Trademark of LOGITECH, Inc. *


Charlie, you sweetie!
Now I can say "yes". We've finally got it together. Compatible at last. Just slip me your software and I'll make it do things you only dreamed about... In either mode. With my power, your programs and the Gemini Emulator Board, just imagine the things we can do together. We can
MAKES THE ZENITH Z
COMPATIBLE WITH T
LIBRARY OF PROG
FOR THE ZENITH Z100
VIA AIR MAIL
PAR AVION
Canada 2
Canada 64
CROSSTALK

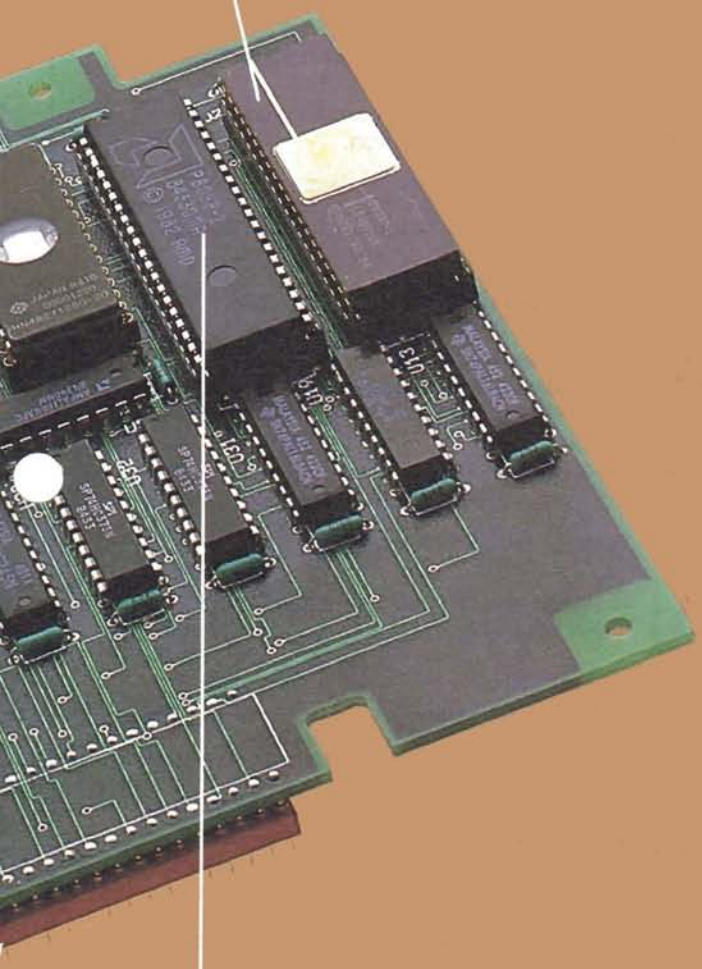
THE GEMINI EMULATOR BOARD



COMPATIBLE AT LAST!

The Gemini Emulator Board Converts Zenith Z100 Computers To True IBM/PC Compatibility

Optional 8087 mathematical co-processor, functional in both the Z100 and IBM/PC modes.



8088 processor, relocated from the Z100 mother board.

ty pin plug to Z100
h board.

OPERATIONAL REQUIREMENTS AND FEATURES

- Z100 video board must have 64K video RAM chips installed.
- IBM compatible operating system must be used in IBM/PC mode. The operating system is not supplied with the Gemini Emulator Board.
- Up to 8 MHz operation is possible on upgraded computers with 8 MHz processors installed. However, due to possible timing problems, operation at this speed is guaranteed only on factory-built 8 MHz computers.
- Gemini Firmware Upgrades: all regular firmware upgrades will be made available, free of charge, through a public computer information network, or from the manufacturer at a nominal charge.

WARRANTY

- During warranty period, dealer or factory will exchange any faulty Gemini Emulator Board free of charge.
- Warranty does not cover installation charges.
- Service: a two-year post-warranty exchange/ repair service is available. Exchange or repairs will be made at a nominal charge.

A Dozen Important Questions Regarding the Gemini Emulator Board.

Q: Does a Gemini enhanced Z100 operate with IBM software only?

A: No. Gemini provides two operating modes: Z100 or IBM.

Q: How do I select the Z100 or IBM operating mode?

A: The desired mode of operation is selected by a one key response to a screen prompt at the time of start-up or re-set.

Q: Can I use the software I've already purchased for my Z100?

A: Of course. The Gemini will not impede the original operations of your computer.

Q: Is there any additional software to be installed by the user?

A: The Gemini requires no additional software. However, an IBM compatible operating system will be required to operate in the IBM mode.

Q: Does the Gemini use any of the S100 slots?

A: No. Gemini was designed with a view to other important uses for the S100 slots.

Q: Where is the Gemini connected?

A: The Gemini Emulator Board simply plugs into the 8088 CPU socket of your Z100.

Q: Is the 8087 Numeric Data Processor supported?

A: Yes. The 8087 is supported on board (socket provided) and operates in both the Z100 and IBM modes.

Q: Are IBM graphics supported?

A: Yes. Both high resolution and medium resolution graphics are supported.

Q: Does Gemini run the Microsoft Flight Simulator?

A: Yes. However, the Z100 hardware does not support the sound portion.

Q: Can I install the Gemini Board myself?

A: Yes. It comes with a comprehensive installation guide for both All-In-One and Low Profile models of the Z100 computer.

Q: When? Where? How much?

A: Available now through local retail outlets. About $600 U.S.

Q: How would you evaluate Gemini's compatibility with IBM?

A: A Gemini enhanced Z100 is extremely compatible with IBM, within the limitation of the Z100 hardware. However, your evaluation is what counts. See your local dealer for a "hands-on" test. Write Gemini Technology Inc. for names and addresses of dealers nearest you.

Gemini Emulator Boards are backed by a 90-day warranty from the date of purchase. They are supported throughout North America by

11 - 11151 Horseshoe Way,
Richmond, B.C., Canada V7A 4S5

For general information or technical support, call (604) 274-1283.

LITHO'D IN CANADA

Recovering Lost Files

Fred W. Kent
1057 Lake Road
Conneaut, OH 44030

As a former LSI-11 owner I have been keenly interested in the ability to recover files or programs that have been accidentally deleted. The DEC LSI-11 system provided a relatively simple method for doing this as their disk system utilized contiguous file systems. It was a simple matter to locate the exact physical address of any deleted file, as the space was never reclaimed unless one did a special operation called squeeze which physically relocated disk files on consecutive sectors. Once, the apparently empty blocks were located one simply renamed the space with the designated number of blocks which PIP displayed, and presto, you had your file back and perhaps several others, as well. If there was more than one file present, you just deleted the newly created file and renamed again with a fewer number of blocks until you had the boundaries of your desired file.

In upgrading to a Z-100 some three years ago, I had to learn to live with the apparent fact that there simply was no way to recover anything that had been deleted unless I was willing to spend much time and effort to fathom the MS-DOS filing system and then the effort seemed dubious. It appeared to require that I write a machine code program to do absolute disk reads and then try to fathom the FAT or Allocation Table. In addition to other difficulties, I had to switch from octal to hexadecimal, something I can only do with a code book in one hand. The last straw occurred one day when in cleaning unused hard disk space, I forgot which subdirectory I was in and wiped out all the files on the local yacht club for which I was treasurer.

Most of my efforts with machine code and MASM assemblers have met with extremely loud crashes, although I finally did succeed in writing code to change color which mercifully worked. Later, a series of utility programs have been made available from HUG which are helpful in some of these areas as they provide insights into the inner workings of MS-DOS machine coding. Still nothing much came by to help with deleted file recovery. A call to the fine Cleveland Store #13 asking for a program to recover such files was fruitless. They muttered something about other companies offering such a program and hung up hastily.

Two events have lately occurred which revealed most of the key thinking I have gone through to solve this problem, the first being a series of fine articles by William Adney who gave much insight into the workings of MS-DOS. This gentleman obviously knows his way around the disk and I listen gratefully when he writes for HUG. I quote from his most recent article in the March issue the key phrase "Last Erased, First Used" (LEFU). It caught my immediate attention when he deported that somehow MS-DOS "remembers" what disk clusters have been erased and further that they are never physically altered. The first byte of the directory entry is set to E5 and some alterations are done to the FAT entry for the erased files to signal to DOS that the space is now free to be overwritten with a new file or an updated or appended file. Otherwise, the deleted file remains undisturbed on whatever tracks, clusters, blocks, etc. It was put on during creation. The question, how to locate these non-contiguous sectors, etc. now began to yield answers of a very simple nature, made simpler by an apparent bug in a local IBM-PC on which I was consulted.

It seemed that the IBM computer operator was prone to having sudden garbage appear in her random files without warning while using various programs that I wrote for her on my Z-100 and then adapted to run on her PC. Upon examination of this bug, I soon noticed that the expected filesize had greatly enlarged. Several times I had to create a new file for her and copy from the old file until the extraneous information was encountered and then delete the old file and rename the new. After three such episodes, her husband, the company owner made the observation that it always seemed to occur during updates of existing records and never during new record entry.

With this information we deduced that she was apparently trying to update a non-existent record. For example, she would try to update record 2034 and accidentally type 2134. This in itself would not alter a file. However, if one then "puts" a random record into any record location beyond the present filend, the FAT table is immediately upgraded so that the file pointer can reach such a record. In so doing, since the intervening records are not consecutively entered, the system does nothing to change the original contents of those records and the filesize jumps, in the above case, from 2034 to 2134. Now records from 2035 to 2133 contain apparent garbage. However, if the space so enlarged happens to overlay an ASCII file, one can immediately discern that these records contain fragments of previously erased files which have not really been erased at all.

We solved this problem rather neatly by writing an error trap into her update routine which would not allow her to update any record higher than filend and quit having to go down there nights to rewrite her damaged files.

Now we have all the clues necessary to recover an erased file at hand and it is just a matter of putting them together which can be done rather simply with a short BASIC program. Briefly, one has only to create a new random file with a record length of 1 and immediately "PUT" a record at any location known to be a higher total byte count than the erased file he is seeking.

For example, suppose we deleted a file named "ELMER.BAS" and then realized we still needed it. If memory is good enough that the operator can recall the approximate file length which we shall say here was approximately 18000 bytes, we then open a new file for temporary use as follows:

```
200 OPEN "R",1,"FILE.TMP",1:'       OPEN TEMPORARY FILE
210 FIELD 1,1 AS A$:'               FIELD RECORD LENGTH OF ONE
220 LSET A$=CHR$(0):PUT 1,20000:'   PUT A RECORD WELL BEYOND
                                    PROBABLE FILELENGTH
230 FOR X%=1 TO 20000:'             SET INTEGER LOOP
240 GET 1,X:'                       GET A BYTE
250  IF ASC(A$)=26 THEN STOP:'      CHR$(26) OR 1AH IS END OF FILE
                                    MARKER FOR BASIC FILES AND ALL
                                    SEQUENTIAL FILES
260 PRINT A$;:'                     DISPLAY CHR ON SCREEN
270 NEXT:'                          LOOP TILL LINE 250 TRUE
280 STOP:'                          EOF LOCATED
```

The above routine is the basic strategy. It is a simple matter to open a sequential file of the original erased file's name and write the incoming recovered file to disk. Upon completion of this task you have only to kill the temporary file created in line 200.

I have tried this program on every conceivable type of file that I have on my disk. The only drawbacks I have found are first, if the disk has been written to by intervening programs before file recovery is attempted, in all probability some or all of the deleted file has been overwritten and has thus destroyed the chance for recovery and second, execute files and random files have no standard end of file marker as do sequential files and BASIC files which are really sequential in nature.

There is one hope here for such files which I have not had time to try, that being that since the minimum space that any file will occupy on disk is a cluster regardless of file size. If one reads in such a file in increments of cluster size, he can usually find the file end by experimentation. Here we must bear in mind that cluster size is 2048 bytes on a hard disk of less than 8 megabytes and 1024 bytes on a DSDD 5 inch disk.

It is unwise to print "COM" and "EXE" files to screen except in ASC form. Some of the signals contained in these machine programs may crash the computer and force you to begin again with a reboot, however, I have recovered "COM" files already by just writing the incoming ASC forms to a sequential file with a "COM" suffix.

On long files, ZBASIC is rather sedate, so I have found it convenient to compile this BASIC program which I have aptly named "RECLAIM.BAS" with BASCOM to produce "RECLAIM.EXE".

Since the concept here is relatively simple and the need for some recovery process is acute, (don't we all make mistakes), I am surprised that no one has ever thought of this technique. I believe it would make a nice addition to the family of utility programs that Heath and IBM compatibles thoughtfully provide with each machine they market. I offer this for public domain and would be pleased to hear that others may find it useful.

A complete suggested program appears below. I have a hard disk drive and have, therefore, gone to extra trouble to be certain that my temporary file can never be too short. Only 2^15 can be addressed by BASIC or 32567 bytes. Therefore, I open the temporary file with a record size of ten and place the dummy "PUT" which now will enclose 325,670 bytes. Then, I close it having established a great deal of disk space into the dummy file and reopen the same file with a record size of one. (On a very large file you may have to define a larger record size and Field & Read each individual byte in order). This is fine if you have a Winchester drive. If you have just a disk drive, one trick is to read the directory at the bottom of which will appear the total available disk space. Use that number as a guide as to the total number of bytes you may wish to include. You may, otherwise, be unable to open the file and get an out of disk space error message.

```
10 '                  --Filename  RECLAIM.BAS--
20 '                        Written by Fred W Kent
30 ' DATE: 03-11-86                    1057 Lake Rd
40 '                                   Conneaut Oh 44030
50 '               Purpose: Regain deleted files
60 '
70 '          -Initialize and display startup message-
80 '
90 CLS:COLOR 2:DEFINT M-Z
100 PRINT TAB(20);"---Recover deleted files---
110 PRINT:PRINT"If you have written anything to disk
    since deleting your file"
120 PRINT
    "this program will probably not function correctly."
130 LINE INPUT "Shall we continue   Y/N ";Y$:
    IF Y$="Y" THEN 140 ELSE CLS:END
140 PRINT "Enter drive name ";:DR$=INPUT$(1)+":":PRINT
150 BUF$=DR$+"BUFFER.TMP"
160 PRINT:INPUT
    "Enter byte count well above lost file length";BC
170 FILEND=(BC/128)+1:
    '                Number of records to include in file
180 '
190 '                 -Main program section-
200 '
210 OPEN "R",1,BUF$,128:'           Open buffer file
220 FIELD 1,128 AS A$:'             Set field length
230 LSET A$=CHR$(0):PUT 1,FILEND:'  Sets filesize to
                                      BC+1 times 128
240 CLOSE 1:'                       Temporary buffer
                                      established
250 '
260 '              -Filetype selection menu-
270 '
280 DATA Basic,Sequential,Random,Com or Exe
290 PRINT:FOR X=1 TO 4:READ ITEM$
300 LOCATE,20
310 PRINT USING "# . . \        \";X;ITEM$
320 NEXT
330 PRINT:PRINT TAB(20);"Enter No. of filetype ";
    :Q$=INPUT$(1):G=VAL(Q$)
340 ON G GOTO 360,360,550,720
350 '
360 '      --Procedure for Basic or Sequential files--
370 '
380 LOCATE ,20:LINE INPUT
    "Enter full filename & type of lost file ¦ ";FILE$
390 FILE$=DR$+FILE$
400 OPEN "O",2,FILE$:'    Open recapture file for output
410 X=0:'                 Initialize byte counter
420 OPEN "I",1,BUF$:'     Open Buffer file for input
430 WHILE NOT EOF(1):'    Set conditional loop
440 X=X+1:'               Increment byte counter
450 A$=INPUT$(1,1):'      Get one byte
460 PRINT #2,A$;:'        Save to file
470 PRINT A$;:'           Display byte on screen
480 WEND:'                End loop
490 CLOSE:'               Make file permanent on disk
500 PRINT:PRINT"File ";FILE$;
    " has now been recovered. Filesize is";X;"bytes"
510 PRINT:LINE INPUT
```

```
    "Delete temporary file  |BUFFER.TMP|   Y/N  ";Y$
520 IF Y$="Y" OR Y$="y" THEN KILL "BUFFER.TMP"
530 END
540 '
550 '          --Random file instructions--
560 '
570 CLS
580 PRINT"Random files have no end of file marker
    They exist in Clusters"
590 PRINT "of 1028 bytes each  Therefore, recovery must
    be by means of reading"
600 PRINT "records starting at byte one until garbage
    is encountered.":PRINT
610 PRINT "File BUFFER.TMP now contains your lost random
    records":PRINT
620 PRINT "It is now necessary to open file BUFFER.TMP
    as a random file"
630 PRINT "with the proper record length and read the
    records one at a "
640 PRINT "until garbage appears  When this is determined,
    open another"
650 PRINT
    "random file of desired filename in order to copy"
660 PRINT "records from BUFFER.TMP to the new file
    Number of records"
670 PRINT "copied must be one less than the first record
    containing garbage "
680 PRINT:PRINT "When you are satisfied with your random
    file recovery, delete"
690 PRINT "BUFFER.TMP to free up the disk space "
700 END
710 '
720 '          --Procedure for COM & Exe files--
730 '
740 CLS:PRINT,"--Instructions for COM & EXE files--"
    PRINT
750 PRINT"These files have no end of file marker
    They exist in Clusters"
760 PRINT "of 1028 bytes each  Therefore, recovery must
    be by means of isolating"
770 PRINT "as many clusters as the original lost file
    required.":PRINT
780 PRINT:PRINT" Try opening BUFFER.COM as a random file
    with record length of"
790 PRINT "128 bytes and a COM or EXE file the same
    You must guess the probable"
800 PRINT "number of records to read into your new file
    (i.e  8 records will"
810 PRINT "contain one Cluster of 1024 bytes)  If program
    does not work, you"
820 PRINT"you must delete your COM or EXE file and restart
    recovery by copying"
830 PRINT"more groups of records. That is if 8 records is
    not enough you must try"
840 PRINT"transferring 16 records or 24 or 32 or whatever
    it requires to isolate"
850 PRINT"your lost file."
860 PRINT:PRINT"NOTE: If you rename BUFFER.TMP to
    BUFFER.COM or BUFFER.EXE  the program"
870 PRINT"will run  You probably will have a filesize
    that is much too large
880 PRINT"The only real drawback seems to be in
    recreating a machine program that
890 PRINT"is much too short and then deleting BUFFER.TMP
    As long as BUFFER.TMP
900 PRINT"exists, you can keep trying until you have
    success."
```

```
8000 '      -Subroutine to underline strings-
8010 '  Characters are 8 pixels horizontally by 9 pixels
      vertically
8020 '                                  by Fred W Kent
8030 '                                  1057 Lake Rd
8040 '                                  Conneaut,Oh 44030
8050 '
8060 '          -Initialize sample demo-
8070 '
8080 L=8:C=1
8090 CLS
8100 A$="This will demonstrate use of underline feature
    of this subroutine"
8110 LOCATE L,C:PRINT A$
8120 GOSUB 8170:'                  Call underline routine
8130 END
8140 '
8150 '        -Subroutine---Define underline
8160 '
8170 DEF FNA=(C-1)*8:'Calculate starting horizontal pixel
8180 DEF FNB=(L*9)-1:'Calculate vertical offset
8190 DEF FNC=(LEN(A$)*8)-FNA:'Calculate length of underline
8200 LINE(FNA,FNB)-(FNC,FNB) 'Draw line under desired string
8210 RETURN
```

*

WILDFIRE

INCREDIBLE PERFORMANCE IMPROVEMENT

from
Software Wizardry

Speed Finally, the ultimate speed-up solution for the Z-151 and Z-161.* With WILDFIRE your Z-151/161 will go faster than the 8 MHz Z-158.

This is not just a simple NEC V-20 addition — not just a simple crystal change — WILDFIRE provides much more speed improvement than a simple processor swap and with greater compatibility.

Compatible WILDFIRE incorporates the NEC V-20 chip and other enhancements to give an unbeatable combination of performance improvement and greater reliability. WILDFIRE surpasses the 80286 processor chip additions for compatibility with the IBM PC AND costs up to $600 less.

Contact your local dealer, or order direct from Software Wizardry by calling (314) 724-1738.

Dealer inquiries invited.

Featuring

- A daughter board which mounts on your processor board, fitting comfortably between it and the video board
- Several additional higher speed chips, to replace socketed chips on your processor board
- A high-speed/low-speed switch with an LED indicator to mount on the computer's front panel, and a hardware reset switch

Price Software Wizardry brings you WILDFIRE, with attractive front panel, complete installation instructions, and reliable daughter-board design for only $249.

No one else can give you so much performance for such a small price.

WILDFIRE

Software Wizardry

Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, MO 63301
(314) 724-1738.

*WILDFIRE only works on Zenith Z-151 and Z-161.

SPREADSHEET Corner

Part 14

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

This "SPREADSHEET Corner" article is a follow-on of the "SPREADSHEET Corner — Part 13" article! DO NOT TRY to do this project without having finished and understanding the previous article! You will not have the proper worksheet or the background for this one!! I hope that you readers are doing the assignments at the computer and not just reading about them and saying that when I have the time I will try this project. You will not learn the many useful things that can be done with the modern day spreadsheet programs. I will be doing this article with the H/Z-150 PC. It can be done on the H/Z-100, as well with the Lotus 1-2-3 software. You will need a printer that can do graphics, also.

I would like to review the various printing functions in this article:

1. Screen Print.
2. Spreadsheet Print.
3. Graph Print.

The Screen Print is very easy. With the Printer turned ON and the paper aligned, hold down the Shift key and press the PRTSC key on the H/Z-150 PC. Whatever is displayed on the screen will be printed, assuming that you have the computer and the printer configured correctly. This series is not a tutorial on these setups so I will not go into it any further. However, you must setup the print font desired, lines per inch, boldface, etc. Lookup all these items in your printer manual. I am going to describe the EPSON in this article because my questionnaire you readers answered for me showed that more of you are using this printer than any other.

The Print Spreadsheet as displayed requires a little more thought! Again with the printer turned ON and the paper aligned, type /PPR (Print, Printer, Range).

```
Enter the Print Range
```

Press HOME, type a period to anchor the Top Left-hand Corner of the Spreadsheet, press END, press HOME, and press Return. (Check the painted area to see if the area you wish to print is painted.) An alternate method would be to type in the upper left-hand Cell, a period, and the lower right-hand Cell. I will be using the painted area most of the time in this article because it is easier to see any errors that might be made. Now, type O (not zero). This selects the Options menu which allows the page formatting. This time we will want a Header on our printout. Type H (Selects HEADER which allows the entry of a page heading). Type the following header line:

```
@|YOUR NAME--SPREADSHEET Corner BUDGET|PAGE #
```

press Return. The @ symbol will print today's date if you entered a date when you booted up or if you have a clock board in your setup. The two | symbols cause everything to be centered and the # symbol will cause the page number to be printed. Next, type Q (Selects Quit to return to the printer menu), and type an A (Selects Align to reset the line-counter to 1). Now, type G (Selects GO to print the specified Range) and type a Q (Selects Quit to return you to the Ready Mode). Note! If you wish to Abort the printing at any time, press the CRTL key and the SCROLL/BREAK key! Use the File/Retrieve commands to bring the last article's file into memory. Try the above printing sequence for practice. Did it work out? If not, review what you did with the step-by-step procedure I describe above and check the computer/printer configuration with the printer manual. File/Save calling this file Figure_3.

In the last "SPREADSHEET Corner" article we did a Line Graph and a BAR Graph with Legends and Titles. We SAVED the Graph file on our Data Disk as a .PIC file. We also added a MACRO to SAVE the current worksheet and called it ALT-S. Do you find the two .PIC files on your Data Disk? For a Review of this work, we will do a PIE and a STACKED-BAR Graph.

Retrieve the FIGURE_3 spreadsheet file from the Data Disk using /FR. Press G (selects Graph), select Type, select PIE, and select View (only the A Data-Range will show). Press the Spacebar to return to the Graph Menu. Select Name, select Create, type PIE, and press Return. Select Save, and type PIEBUDG, and press Return. Select Quit. To SAVE this work, use the ALT-S MACRO! Hold down the ALT key and press the S key. Did it save the file? If you did not do the last assignment with this MACRO, do not bother to use the rest of this article without going back and DOING the assignment!!!

Now, we will review the View Various Graphs. First, view the current graph by pressing the F10 function key. What did you find? A PIE Graph? After viewing the Graph, press the spacebar to return to the Graph Menu. If you have been doing YOUR HOMEWORK, we will soon have a Parade of Graphs (sometimes called a "Slide Show". So, stay with me, it will be worth while!

Type /G (select the Graph Menu.), type N (select Name), type U (select Use which allows display of all NAMED Graphs.), and type LINE, press Return. Press the spacebar to return to the Graph Menu. (REMEMBER this has to be done at the Computer, DO NOT just read about it!!) I made it very simple so that it would not require a lot of keyboard entries. Many readers answered my Questionnaire telling me that they did not like LONG assigments like the INCOME TAX project. So, I am NOT going to do a new INCOME TAX Project this year. The readers that are interested will be able to review last year's project and know how to revise the program. I had planned on doing it this year with MACROS and MENUS with Protected Cells so that entries could only be made in the required spaces. I guess I am off the track, but it was a good time to tell you readers why I am not going ahead with that project. Many readers wanted to buy the program from me on a disk, but I am not doing this series for me to do your work. You will only learn the many things that can be done with spreadsheets by doing the work yourself.

Again, type N, type BAR, and press Return. Press the spacebar to return to the Graph Menu. Once again, type N, type U, type PIE, and press Return. Press the spacebar to return to the Graph Menu.

Before I go on, we will create a Stacked-Bar Graph. By now some of the readers should know the procedure. Do you? I will briefly do it again. Select Type, select Stacked-Bar, and select View (press the spacebar when you want to go back to the Graph Menu). Select Name, select Create, type STACK, and press Return. Select Save, type STABUDG, and press Return. Select Quit, hold down the ALT key and press the S key to SAVE using our MACRO!

Now, we will Format the Graph's Scale. Type /G, type N, select Use, and select BAR (press the spacebar to return to the Graph Menu). Select Options, select Scale, select Y Scale, select Format, select Currency, type 0 (zero decimal places), and press Return. Select Q twice and select View. The Y-axis will be displayed with $ signs. Do you find them? Press the spacebar to return to the Graph Menu. To save our Graph with these changes, select Name, select Create, select BARBUDG, and press Return, select Replace. Select Quit to exit the Graph Menu. Hold down the ALT key and press the S key to Save with our Save MACRO!

In the last article, I gave the instructions for creating a MACRO that will display the four Graphs. How many readers did this macro? If not, I am going to give the instructions again:

1. Goto Cell A16, type '\G and press the Right Arrow key
2. Type '/GNULINE~Q in Cell B16 and press the Down Arrow key
3. Type '/GNUBAR~Q in Cell B17 and press the Down Arrow key
4. Type '/GNUPIE~Q in Cell B18 and press the Down Arrow key
5. Type '/GNUSTACK~Q in Cell B19 and press Return
6. Goto Cell D16, type — DISPLAY LINE GRAPH—press Down Arrow key
7. Type — DISPLAY BAR GRAPH — in Cell D17 and press the Down Arrow
8. Type — DISPLAY PIE GRAPH — IN CELL D18 and press the Down Arrow
9. Type — DISPLAY STACKED GRAPH in Cell D19 and press Return

SPREADSHEET Corner COMPANY BUDGET IN THOUSANDS 15-Oct-85

YEAR	ORDERS	SALES	BACKLOG
1980	$2,683	$1,846	$4,529
1981	$3,270	$2,210	$5,480
1982	$4,230	$2,610	$6,840
1983	$5,093	$3,305	$8,398
1984	$7,873	$4,386	$12,259

\S	/FS~R	SAVES WORKSHEET
\G	/GNULINE~Q	DISPLAYS LINE GRAPH
	/GNUBAR~Q	DISPLAYS BAR GRAPH
	/GNUPIE~Q	DISPLAYS PIE GRAPH
	/GNUSTACK~Q	DISPLYS STACKED GRAPH

Figure 3

Next, we must ALWAYS name the macro! Goto Cell A16 and type /RNLR (Range Name Labels Right) and press Return. Call it \G and execute the macro? Hold down the ALT key and press the G key. Press the Spacebar after Viewing each Graph. Did you find four Graphs? If not, follow the keystrokes for each Graph and you should find the error. Another way to test a macro is to hold down the ALT key and press F1 to go into what is called a "step" mode. This will allow you to proceed through the macro one step at a time by pressing the spacebar to get each step. This is an interesting exercise to try even if your macro did work so that you can really see what it is doing. Macros are just a series of keystrokes! Remember, you can still press F10 to display the last Graph created and press the spacebar to return to the worksheet.

Again, I think that we should print our worksheet with the macro lines. The only change in the printer setup I would suggest would change the Range and the Header Line as follows:

```
|YOUR NAME--SPREADSHEET Corner BUDGET|BUDGETG
```

Also, be sure to SAVE the worksheet by holding down the ALT key and pressing the S key. How many times have we used this macro? Has it saved a lot of tedious typing? I think so because I do not like typing that much. Compare your printout with my Figure 3.

How about the last printing review — Print Graphs. You will need a printer that has the capability to print graphs! Goto the LOTUS ACCESS SYSTEM MENU and select PrintGraph. This portion of 1-2-3 permits you to control the way the graphs will be printed. First, we will print one graph, full size on one sheet of paper. With SELECT highlighted, press Return. With "Select Graphs for OUT-

PUT" displayed, select BARBUDG (it will be highlighted), and press the spacebar to MARK this as the selected file. Note the # symbol will appear to the left of the filename. If you make a mistake or change your mind, press the spacebar again, the # will disappear indicating the filename has been unmarked. When you have the correct selected file marked, press Return. Select Options, select Font, select 1 (Font 1 will be used for the first line of the Graph Title.), select BLOCK1 (Style selections are made the same way as graph selections from the displayed list in the status area of the Menu Screen.), select Size, select Full, select Quit, and select Quit again. Make sure that the printer is ON and the paper aligned. Select Align and select GO. (The Mode Indicator will flash "WAIT" and the printing of the graph starts.) Printing graphs takes time, so be patient; have a cup of coffee, walk the dog, etc. When the printing is complete, you will find a graph printout worth waiting for! Did you?

How about printing two graphs on the same page, each half size? Again with the SELECT highlighted, press Return. With "Select Graphs for Output" displayed, select PIEBUDG and STABUDG. Use the Down Arrow key to reach these files. When you reach one of the two, press the spacebar to MARK this file as one of the selected files for printing. A # symbol will appear to the left of the filename. Down arrow again to get to the second file to be printed and press the spacebar to MARK this file as the second of the selected files for printing. Again, the spacebar will be used to put the # mark to the left of the filename. When you have the two files that were chosen marked with the # symbol, press Return. Select Options, select Font, select 1, select BLOCK2 (These selections will be used for the first line of the graph), select Font, select 2, and select ROMAN1. (These selections will be used for the rest of the Graph.) Select Size, select Half, select Quit, and select Page. (The paper will be advanced so that the printer paper will be aligned for the next two graphs printing.) Select GO and again be prepared for a long wait, but it is worth it! Remember, a picture is worth a thousand or is it ten thousand words!

Use another select Quit and select Yes and the LOTUS ACCESS SYSTEM MENU will be displayed. When the reader thinks about the many options that were possible for graph Type, Fonts, etc., I will expect that the interested reader will want to try all the combinations! You will soon find the Options that you like the best and the ones that work with your printer the best! SO, BE SURE TO TRY THEM!!!

I will let the reader decide how to print the LINEBUDG Graph. Are you going to use full size? What Fonts will you choose? Whatever you decide, be sure to print the LINE Graph!

Next, with the printer ON and the paper aligned; type /PPR (same as used above) and you will find that the Range that you previously entered will appear painted this time. Press Return, type O (selects Options) and type H (selects HEADER) and Edit the last Header Line that you used. The cursor will be located at the end of the line, press the Left Arrow key 6 times, and type BUDGCELL. Press Return, press the spacebar, and press Return. You should now have the following Heading Line:

```
@¦YOUR NAME--SPREADSHEET Corner BUDGET¦BUDGCELL PAGE #
```

Have you followed this procedure so far? Now, type S. (Selects Set-Up which allows entry of printer codes.) Sometimes printer codes are called Escape Codes. Both mean the same thing here.

This time we are going to print the Cell-Formulas for our BUDGET Spreadsheet. The printout will be fairly long even for our simple example; so, we will change the print line spacing to 8 lines per inch. If you have not reviewed your printer manual lately, NOW is the time to look up how to change line-spacing with your printer using Escape Codes. To Clear any previous printer codes that have been used, it is always a good idea to use Clear! Enter \027\064, the Clear code; enter \027\077, selects the ELITE Type; and enter \027\048 to select 8 lines per inch line spacing. Thus, your command line should look like the following:

```
\027\064\027\077\027\048
```

and press Return. Type P (Selects Page-Length), the default is 66 lines for the normal 6 lines per inch, but we want to change it to 88 lines per page:

```
Enter Lines per Page(20. 100):66
```

Type 88 and press Return. Next, type M (Selects Margins) and type R (Selects Right Margin):

```
Enter Right Margin(0..240)·76 (default value)
```

Type 96 and press Return. Type O (Selects Others) and you find a choice of AS-DISPLAYED (Default we used in the first example), CELL-FORMULAS, FORMATTED, UNFORMATTED. Press the Right Arrow once, CELL-FORMULA will be highlighted, and press Return.

Again, type Q (Selects Quit) and you will return to the Printer menu. Type A (Selects Align) to reset the line-counter to 1. Type G (Selects GO) to print the specified Range that was painted. Your printout should show what data is in each Cell line-by-line with the HEADER at the top of the page. This printout is valuable for debugging a spreadsheet. Did you get this? Type Q (Selects Quit) to return to the Ready Mode. SAVE the spreadsheet by typing /FS and in response to "Enter save filename:" type BUDGCELL and press Return. It was not necessary to change the print size to ELITE, but I wanted you to try it for experience! Check your printout with my copy.

```
15-Oct-85        YOUR NAME--SPREADSHEET Corner COMPANY

SPREADSHEET Corner COMPANY  BUDGET IN THOUSANDS 15-Oct-85
=============================================================
YEAR       ¦¦     ORDERS          SALES          BACKLOG
-------------------------------------------------------------
1980       ¦¦          2683           1846 @SUM(C5,D5)
1981       ¦¦          3270           2210 @SUM(C6,D6)
1982       ¦¦          4230           2610 @SUM(C7,D7)
1983       ¦¦          5093           3305 @SUM(C8,D8)
1984       ¦¦          7873           4386 @SUM(C9,D9)
-------------------------------------------------------------

\S             /FS~R              SAVES WORKSHEET

\G             /GNULINE~Q         DISPLAYS LINE GRAPH
               /GNUBAR~Q          DISPLAYS BAR GRAPH
               /GNUPIE~Q          DISPLAYS PIE GRAPH
               /GNUSTACK~Q        DISPLYS STACKED GRAPH
```

Figure 4

Let's print a FORMAT TEXT next. Retrieve your original file because our BUDGCELL spreadsheet file has different settings saved with it. Type /FR, enter the Budget filename, and press Return. For this printout we must FORMAT the Range. Press HOME, type /RFT (Range, Format, Text):

```
Enter range to format. A1 .A1. anchor (type period),
press END key. press HOME. and press Return
```

Check your spreadsheet! It should Display in TEXT format the Formulas that are in the cells. Note! That in some cases you could

find that the column-width could be too narrow, resulting in the cell formula being truncated. In this case, you would go to one of the columns that need to be widened and type /WCS (Worksheet Column-Width Set):

```
Enter column-width (1..72).9
```

Press the Right Arrow key until all of the formula shows in the cell or type in the correct column-width if you know what it should be. Go to any other column that may require its column-width to be changed. Did you find any that needed changes? Be sure to check your worksheet for this type of problem, see if all formulas can be read!

Now, we will print this spreadsheet. Make sure that the printer is turned ON and the paper is aligned. Type /PPR and because the Range is the same as we have been using, even though we may have changed some column-width cells in the selected range, it will be the correct range as was painted previously. Do you find this? Adjust the range if it is not right for some reason. Type O (Options), type H (HEADER), and Edit the Header Line. The cursor will be located at the end of the header line. Press Left Arrow 6 times, type BUDGTEXT, press the DEL key 6 times, and press Return. The Header Line should look like the following:

```
@¦YOUR NAME--SPREADSHEET Corner BUDGET¦BUDGTEXT
```

I would like to have you print this worksheet with a different print size called CONDENSED type size by most printer manuals. To do this, type S (Selects setup), Clear old setups, and enter the following new printer codes:

```
\027\064\027\015
```

press Return. Type M (Selects Margins), type R (Selects Right Margin):

```
Enter Right Margin (0 .270)  76
```

Type 132 and press Return. Type Q (Selects Quit) to return to the Printer Menu, type A (Selects Align), type G (Selects GO) to start the printing, and when the printing is complete type Q (Selects Quit) to change to Ready Mode. Do you prefer the Cell-Formulas or the TEXT printout as a troubleshooting tool? Save the worksheet by typing \FS, name it BUDGTEXT and press Return. Before we start our next worksheet, always clear the worksheet by typing /WEY (Worksheet, Erase, Yes). It was not necessary to use this small type in this case, but again, I wanted the readers to practice changing print type. Check your printout with my Figure 4.

I hope that the really interested readers learned how to use their spreadsheet software and their printer to do nearly any type of printing and how to present Graphs! In future projects we will be using these procedures with complex projects where it would be harder to learn the fundamentals. I hope that you feel that we have accomplished this. As always if you have any problems, ideas, etc. I would like to hear from you. I will answer all questions if the reader's letter includes a stamped, self-addressed envelope (business size). I have fallen behind because my wife and I have had an accident breaking my foot and the wife's hip. We are on the road of recovery and I will catch up on your Letters.

In the meantime, Happy Spreadsheeting! I will most likely do another Database project using the spreadsheet and macros next. You might want to review the Database commands so you are ready to go!

```
A1: (T) ^SPREADSHEET Corner COMPANY   BUDGET IN THOUSANDS
E1: (D1) @TODAY
A2: (T) \=
B2: (T) \=
C2: (T) \=
D2: (T) \=
E2: (T) \=
A3: (T) ^YEAR
B3: (T) ^¦¦
C3: (T) ^ORDERS
D3: (T) ^SALES
E3: (T) ^BACKLOG
A4: (T) \-
B4: (T) \-
C4: (T) \-
D4: (T) \-
E4: (T) \-
A5: (T) ^1980
B5: (T) ^¦¦
C5: (C0) 2683
D5: (C0) 1846
E5: (C0) @SUM(C5,D5)
A6: (T) ^1981
B6: (T) ^¦¦
C6: (C0) 3270
D6: (C0) 2210
E6: (C0) @SUM(C6,D6)
A7: (T) ^1982
B7: (T) ^¦¦
C7: (C0) 4230
D7: (C0) 2610
E7: (C0) @SUM(C7,D7)
A8: (T) ^1983
B8: (T) ^¦¦
C8: (C0) 5093
D8: (C0) 3305
E8: (C0) @SUM(C8,D8)
A9: (T) ^1984
B9: (T) ^¦¦
C9: (C0) 7873
D9: (C0) 4386
E9: (C0) @SUM(C9,D9)
A10: (T) \-
B10: (T) \-
C10: (T) \-
D10: (T) \-
E10: (G) \-
A12: (T) '\S
B12: (T) '/FS~R
D12: (T) ^SAVES WORKSHEET
A14: (T) '\G
B14: (T) '/GNULINE~Q
D14: (T) ^DISPLAYS LINE GRAPH
B15: (T) '/GNUBAR~Q
D15: (T) ^DISPLAYS BAR GRAPH
B16: (T) '/GNUPIE~Q
D16: (T) ^DISPLAYS PIE GRAPH
B17: (T) '/GNUSTACK~Q
D17: (T) ^DISPLYS STACKED GRAPH
C19: "FIGURE 5
```

Figure 5

A Winchester For The '89

Part Four

Peter Ruber
P.O. Box 502
Oakdale, NY 11769

Hide A Winchester In Your '89

As a departure from the Winchester operating systems we've discussed in the last two installments of this series, I'm going to take you on a guided tour this month on how to stuff a hard disk in your '89. While it's also a splendid way of disguising a new computer acquisition without prompting a host of nagging questions from your wife, I have some reservations about the practicality of this concept over periods of extended use.

As Winchester drives diminished in size and made use of the latest chip designs, their reduced demand on power supplies has made it practical for system integrators to dispense with the costly case and power supply and offer an affordable package to the hobbyist. However, even most IBM-PCs and a host of the compatibles on the market are unable to accept an internally mounted hard disk drive without a new 135-watt power supply to replace the standard 65-watt unit. And herein lies the problem with the '89.

Mounting a hard disk drive in the '89 was the brainchild of Floppy Disk Services, Inc. FDS has made a specialty of selling disk drives and subsystems and was one of the earliest supporters of the '89. In recent years, when some firms have phased out their support of the '89 in favor of other computer systems (need we name names?), FDS has continued to adapt new hardware for the '89 and are obviously successful in doing so.

Late in the summer of 1984, when Microscience International launched the first of several half-height Winchester drives — the 10-megabyte HH-612 — FDS obtained production samples to test with the second generation SASI controller card from Xebec called S1410a. The S1410a had all the features of the S1410 but only half the chip count, improved diagnostic firmware, which increased the reliability while lowering the power requirements.

Full-height drives and S1410 SASI card combinations needed a switching power supply capable of delivering +5 volts at 5 amps and +12 volts at 4 amps. This is rather hefty when you consider that the '89s +5 volt rating is 5 amps for the entire computer. This made it impractical to mount a Winchester in the '89. The +12 volts in a hard disk drive is required by the motor to spin several rigid metal platters at 3600 rpm, and the biggest demand occurs when the drive is first turned on. After 10-20 seconds the drive attains its rotational velocity and the +12 volt requirement usually drops in half.

Being a die-hard H-89 enthusiast, I had to get my hands on one as soon as I could. My FDS system contained the Shugart SA712 equivalent of the Microscience HH-612. The Xebec S1410a drive controller was mounted with spacers to a metal shield in order to deflect RFI emissions from inside the computer. All cables were supplied, and the package was bundled with the Magnolia Microsystems 77320 SASI host adaptor, the MMS ROM set and CP/M manual. The cost? A little under a thousand dollars. The drive has a full-height bezel with a drive indicator light on the front. The mounting hardware is quite simply the same mounting bracket used by Heath/Zenith for their H-17 drive, except that a few extra holes have been added to accommodate the Xebec card and the position of the Winchester drive's mounting screws.

The entire installation process takes less than one hour. You remove your existing internal drive and all mounting hardware. Lift the CPU board out of the mounting frame after removing all plug-in cards and cables, replace the Monitor and Secondary Address Decoder ROMs with the Magnolia equivalents supplied with the kit. Set S501 switches 1, 4 and 5 to the left; all others to the right. If your system doesn't already have the MTR90 4k ROM, install the small jumper cable from top jumper block near U518 to pin 14 on any of the three lower left-hand expansion sockets (P507, P508 or P509). Set the jumper on the MMS 77320 to 78H if you plan to retain your hard-sector controller, or 7CH if you plan to use only the soft-sector controller. There is also a DIP Switch on the 77320 host adaptor card. Make sure all switches are turned to ON.

Replace the CPU board and plug in your boards and cables. Take the drive mounting bracket and secure it to front panel. Mount

the Xebec controller, slide in the Winchester, connect your cables and power up.

This is a simplified overview of the installation, but it does illustrate the ease with which the entire unit is assembled in your '89. Most half-height Winchester drives are about a half-inch longer than their full-height floppy counterparts, so when you slide the drive through the opening, you are likely to encounter blockage from the heatsink fins on the power supply distribution board. Simply bend them close to the metal shield they're mounted on (it won't hurt) and push the drive home.

Floppy Disk Services has prepared an excellent assembly and software installation manual that holds your hand through every step. Nothing has been left to chance. FDS claims to have written the manual step-by-step during an actual hardware and software installation procedure, and I don't doubt it. The manual is profusely illustrated and check-off boxes are supplied for each step so that you don't lose track of your installation steps in the event you're interrupted. Those of you who are accustomed to climbing in and out of your '89 every so often, will probably dispense with the manual after a quick reading.

Is There Enough Power?

My big concern prior to installing the FDS internal Winchester was the '89s power supply. My unit has had periodic power supply problems that I touched on briefly in an article in the April 1985 REMark. When I investigated the power supply problems further with the Heath Consultation Group and various H/Z support vendors around the country, I proceeded to compile a list of available plug-in boards and hardware enhancements for the '89 and the +5 volt power consumption of each board (including Heath's). I had planned an article around this information, but felt this guide would be more useful if included in an evaluation of an internal hard disk subsystem. It is appended to the end of this article.

If your '89 has the upgraded +5 and +12-volt regulators installed (which are supplied with the Z-37 controller kit), your system may be able to handle this internal Winchester installation. If yours is an older model, you should upgrade your +5 volt regulator to a UF78HO5SC or MC78TO5 (Heath part number 442-651) and install it at U101 with a multi-finned heat sink. Heatsinks should also be installed at U102 and U103. Since the Xebec SASI card and the hard disk will generate a significant amount of heat, I would also recommend that you reverse your fan to blow cold air into the computer, and leave the cover unlatched to increase ventilation.

A further modification recommended to me by Leon Cray, a Senior Technical Consultant at Heath, was to remove the yellow wires from P101 of the power distribution board and cut away the orange wires that go from P103 to the bridge rectifier. Then install the yellow wires on the bridge rectifier in place of the orange wires. "Any system configuration," Mr. Cray wrote, "that draws more than 4 Amps should have this modification made on that unit . . . the effect of eliminating these four connections within the plug and sockets is to decrease the overall circuit resistance on the primary side of the bridge rectifier. By reducing the amount of resistance within the primary circuit, we are also eliminating potential sources of heat generation . . . This wiring change was incorporated into all units produced or serviced since the latter half of 1982."

The Xebec S1410a hard disk controller will consume about 1.0 amp of +5 voltage and about 3 to 6 milliamps of +12 voltage, which is quite insignificant. The Microscience HH-612 will draw .9 amps at +5 volts and a +12 volt drain of .9 amps after a brief surge of 2.0 amps as the drive motor starts up. This occurs when you first power up your computer. By the time your system prompt appears on screen the entire power supply will have stabilized.

If your system has all three right-hand slots occupied, plus the standard 16k RAM expansion board on the left, as well as a parallel board, your unit (together with the Winchester drive and controller) will probably be drawing more than the 5 amps the +5 volt regulator can supply. This could lead to system shutdown. Since the Magnolia 77320 card also provides the standard serial ports, all other cards should be removed. The system shut down is usually related to the over-heating of some wires on the power distribution board, as these pins are only rated for 5 Amps.

My first try at this installation resulted in my blowing both the +5 and +12 volt regulators. I hadn't bothered to check into the power specifications of the Shugart SA712 drive prior to installation. After they were replaced, I removed all of my boards except the Z-37 floppy controller, and the system did run as advertised. However, there was a suspect build-up of heat within the computer that I felt might be detrimental to the long term reliability of the computer. ICs have a tendency to deteriorate faster in a hot environment than in a cool one.

If you subscribe to COMPUTER SHOPPER, you will be aware of the many supply houses that sell new, but surplus, computer power supplies at bargain prices, and you have a choice of dozens of units in the $25-50 range that can provide enough amperage for a Winchester drive and controller card. Stuff one of these into a cheap case and run the power cables directly to the "Y" connector on the drive and controller. The effect of this combination will allow the '89's power supply to run cooler and you will be able to fill all your expansion slots with your favorite I/O cards.

Another precautionary measure would be to mount a second fan on the top left of the cover to blow cool air to the CPU, TLB and plug-in boards. This can be wired to the AC lines leading to the internal fan, or to an external line cord to be plugged into an outlet. It's not elegant, but it works.

If you are inclined to leave your '89 "as-is" for extended periods of time, then your internal Winchester will be housed in a stable environment. But if you tinker around, I would recommend securing the FDS hard disk unit in an external cabinet. Although most of the current crop of Winchesters are relatively immune to casual handling, it isn't wise to risk damaging an expensive investment.

Considering all the high-storage Winchester options coming available on the market, perhaps it's time for some Heath/Zenith support vendor to design a replacement power supply board and heatsink assembly that is capable of providing enough reserve power for an internal hard disk drive. The practicality of having a Winchester in your '89 can help to minimize the clutter peripheral equipment creates on your desk.

Though it isn't widely advertised, Technical Micro Systems, Inc. (PO Box 7227, Ann Arbor, MI 48107) has a replacement heat-sink assembly available for about $30 that mounts the voltage regulators away from the CPU/TLB support bracket and allows the fan to be mounted on it. TMSI also has several kits available at reasonable prices that replace a number of heat-generating TTL ICs

with low-power CMOS chips. These modifications should be considered if mounting a Winchester in your '89 is in your future plans.

The Quikstor CP/M and HDOS system evaluated in the previous installment of this series will work quite nicely with this system, too, if you want the option of using your favorite HDOS programs.

Floppy Disk Services has created an interesting piece of system integration by mating the latest high storage technology and proven software with a well-documented installation manual. Unfortunately, it lacks related information on power supply limitations that I feel prospective customers should be made aware of so they can compensate by removing unnecessary boards from their system.

For more information, write to:

FLOPPY DISK SERVICES, INC.
39 Everett Drive, Bldg. D
Lawrenceville, NJ 08648
(609) 799-4440

[The 5th article in this series was originally scheduled to be a guide on how to build your own Winchester subsystem by taking advantage of some of the close-out bargains currently available from a variety of dealers. As I am still collecting and correlating a lot of research, this article will be pushed back one installment, in order to present a new product from C.D.R. Systems, Inc. called SUPER RAM 89. This ingenious enhancement sports a Direct Memory Access controller, 1-Megabyte of RAM, a non-volatile Clock and an SCSI hard disk interface. SCSI stands for Small Computer Systems Interface, and is an evolved upgrade of the SASI standard that most Winchester and Controller manufacturers are adopting in an industry-wide effort to create a greater interchange of hardware.

In addition, other high-storage options are coming available for the '89 which we will include in a future article in this series, as well as some interesting "quick and dirty" hard disk integrations suggested by readers of this series.]

H-89/90 +5-Volt Power Consumption Chart

CPU And Replacement Boards		
Heath Co.	CPU Logic Board w/48k	.94 A
Heath Co.	Terminal Logic Board	1.17 A
Technical Micro Systems	H-1000 Replacement CPU Board (Z80 & 8086 CPU's) w/1024k	1.30 A
D-G Development	Super 89 CPU Replacement Board	
	w/128k	1.10 A
	w/256k	1.15 A
	w/256k & AM9511 Math IC	1.19 A
Floppy Disk Controller Boards		
Heath Co.	H-88-1 5.25" Hard Sector Disk Controller Board	.20 A
Heath Co.	Z-89-37 5.25" Soft Sector Disk Controller Board	.50 A
Heath Co.	Z-89-47 8" Disk Controller Board	.50 A
Bit Zero	Livingston Logic Labs FDC89 8" Disk Controller Board	.21 A
C.D.R.	FDC-880H 5.25"/8" Disk Controller Board	.35 A
Magnolia Microsystems	XM-316 5.25"/8" Dble Density Disk Controller Board	.57 A
Hard Disk Controller Boards		
Heath Co.	Z-89-67 SASI Hard Disk Interface	.50 A
Magnolia Microsystems	MX-320 SASI Winchester Bus Interface	.64 A
Magnolia Microsystems	XM-314 Corvus Winchester Interface	.62 A
C.D.R. Systems	DMA SASI Hard Disk Interface (add-on for SuperRAM 89)	.35 A
Serial And Parallel I/O Boards		
Heath Co.	HA-88-3 3-Port Serial Card	.15 A
Heath Co.	Z-89-11 Multi I/O Board	.35 A
Technical Advisors	Parallel Multi-Port I/O Board (based on number of I/O lines)	.40-.60 A
SigmaSoft & Systems	Universal Parallel Card	.15 A
FBE Research Co.	H89PIP - 2-Port Parallel Card	.05 A
FBE Research Co.	H89UTI Utility Card w/Clock	
	w/No Serial ICs	.24 A
	w/2-Serial ICs	.37 A
	w/AM9511 Math IC	+.05A
FBE Research Co.	H89CTI-a/b/c Parallel Printer Interfaces	.10 A
Kres Engineering	PPT330/3-Port Parallel I/O Board	.18 A
Graphic Boards		
Cleveland Codonics	Imaginator Graphics Board w/Tektronix 4000/4014 Emulation	.50 A
Northwest Digital	Graphics Plus	.90 A
SigmaSoft & Systems	Interractive Graphics Controller (has separate power supply)	- 0 -
New Orleans General Data Systems	HA-89-3 Color Board	1.0 A
RAM Expansion Boards		
Heath Co.	WH-88-16/16k Ram Expansion Board	.05 A
Micro Research	16k Ram Expansion Board	.05 A
FBE Research Co.	16k Ram Expansion Board	- 0 -
FBE Research Co.	Spooldisk 89 / 128k Ram Expansion & Printer Spooler	.60 A
Magnolia Microsystems	XM-311 / 16k Ram Expansion Board	.01 A
Magnolia Microsystems	XM-318/128k Ram Expansion Board	.15 A
Magnolia Microsystems	XM-422-64 / 64K Magnet Network Controller Board	1.1 A
Magnolia Microsystems	XM-422-256 / 256k Magnet Network Controller Board (Consult the Notes section below)	
C.D.R. Systems	SuperRAM 89 Board w/256k	.50 A
	w/512k	.51 A
	w/768k	.53 A
	w/1024k	.54 A
Clock Board		
Bit Zero	Real Time Clock	.06 A
Analytical Products	TIM2 Real Time Clock	.02 A

C.D.R. Systems	Real Time Clock w/Battery B/U (add-on for SuperRAM 89)	.10 A
Miscellaneous Boards		
Analytical Products	REP2 Auto Repeat Key Board	10 uA
Mako Data Products	PSG X 2 Sound Board	.06 A
Magnolia Microsystems	Composite Video Generator	- 0 -
Kres Engineering	DSM240 Dual Speed Module	.13 A
Slot Expanders		
FBE Research Co.	SLOT 4	none
Mako Data Products	MH89+3 Expansion Board	.04 A
Microflash Co.	M-89 Interface Card for M-89 Expansion Box	.01 A
Kres Engineering	7-Slot Expansion Board	.20 A

Notes:

1. You must also allow for the +5-volt power consumption of an internally mounted drive. The standard Heath/Zenith single-sided 48-tpi Siemens drive is rated as:

   .40 A when idle
   .55 A when active

   Double-sided 48-tpi and 96-tpi half-height drives are generally rated as low power, but this statement isn't always true. The QumeTrack 142 is rated at .60 Amps. The Teac FD55B and the Shugart SA455, which is also known as the Panasonic or National JA-551-2, are rated at .38 Amps. While the Teac FD55F (96-tpi) drive is rated at .70 Amps. This disparity should be watched for when you install new drives inside the '89.
2. As of 9/85, the production rights of the Z-37 soft-sector controller were acquired by Technical Micro Systems, Inc., manufacturers of the H-1000.
3. If the Magnolia XM-422-64 Magnet Network Controller board is installed in the '89, the internal drive must be removed or disconnected because the board draws its power from the floppy drive's +5/+12 connector.
4. The Magnolia MX-422-256 Magnet Network Controller has such high power requirements that it must be installed in an external case with a separate power supply.

*

HUG Price List

The following HUG Price List contains a list of all products in the HUG Software Catalog. For a detailed abstract of these products, refer to the issue of REMark specified.

Part Number	Description of Product	Selling Price	Vol. Issue

HDOS HARDCOPY SOFTWARE

885-1008	Volume I Documentation	9.00	
885-1013	Volume II Documentation	12.00	
885-1015	Volume III Documentation	9.00	
885-1037	Volume IV Documentation	12.00	8
885-1058	Volume V Documentation	12.00	

MISCELLANEOUS HDOS COLLECTIONS

885-1032	Disk V H8/H9	18.00	8
885-1044-[37]	Disk VI H8/H89	18.00	
885-1064-[37]	Disk IX H8/H89 Disk	18.00	
885-1066-[37]	Disk X H8/H89	18.00	10
885-1069	Disk XIII Misc H8/H89	18.00	

GAMES

HDOS

885-1010	Adventure Disk H8/H89	10.00	4
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8
885-1031	Disk IV MUSIC H8 Only	20.00	25
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12
885-1068	Disk XII MBASIC Graphic Games	18.00	10
885-1088-[37]	Disk XVII MBASIC Graph Games	20.00	14
885-1093-[37]	D&D H8/H89 Disk	20.00	16
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23
885-1113-[37]	HDOS Action Games H8/H89	20.00	23
885-1114	H8 Color Raiders & Goop	20.00	23
885-1124	HUGMAN & Movie Animation Pkg	20.00	41
885-1125	MAZEMADNESS	20.00	41
885-1130	Star Battle	20.00	45
885-1133-[37]	HDOS Games Collection I	20.00	59
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32
885-8022	HDOS SHAPES	16.00	45
885-8026	HDOS Space Drop	16.00	49
885-8032-[37]	HDOS Castle	20.00	59

CP/M

885-1206-[37]	CP/M Games Disk	20.00	11
885-1209-[37]	CP/M MBASIC D&D	20.00	19
885-1211-[37]	CP/M Sea Battle	20.00	20
885-1220-[37]	CP/M Action Games	20.00	32
885-1222-[37]	CP/M Adventure	10.00	35
885-1227-[37]	CP/M Casino Games	20.00	38
885-1228-[37]	CP/M Fast Action Games	20.00	39
885-1236-[37]	CP/M Fun Disk I	20.00	55
885-1248-[37]	CP/M Fun Disk II	35.00	69

ZDOS

885-3004-37	ZDOS ZBASIC Graphic Games	20.00	37
885-3009-37	ZDOS ZBASIC D&D	20.00	50
885-3011-37	ZDOS ZBASIC Games Disk	20.00	52
885-3017-37	ZDOS Contest Games Disk	25.00	58

UTILITIES

HDOS

885-1022-[37]	HUG Editor (ED) Disk H8/H89	20.00	20
885-1025	Runoff Disk H8/H89	35.00	
885-1060-[37]	Disk VII H8/H89	18.00	
885-1061	TMI Load H8 ONLY Disk	18.00	
885-1062-[37]	Disk VIII H8/H89 (2 Disks)	25.00	
885-1063	Floating Point Disk H8/H89	18.00	
885-1065	Fix Point Package H8/H89 Disk	18.00	10
885-1075	HDOS Support Package H8/H89	60.00	
885-1077	TXTCON/BASCON H8/H89	18.00	
885-1079-[37]	HDOS Page Editor	25.00	15
885-1080	EDITX H8/H19/H89 Disk	20.00	
885-1082	Programs for Printers H8/H89	20.00	
885-1083-[37]	Disk XVI Misc H8/H89	20.00	11
885-1089-[37]	Disk XVIII Misc H8/H89	20.00	20
885-1090-[37]	Disk XIX Utilities H8/H89	20.00	22
885-1092-[37]	Relocating Debug Tool H8/H89	30.00	14
885-1098	H8 Color Graphics ASM	20.00	19
885-1099	H8 Color Graphics Tiny PASCAL	20.00	19
885-1105	HDOS Device Drivers H8/H89	20.00	24
885-1116	HDOS Z80 Debugging Tool	20.00	27
885-1119-[37]	BHBASIC Support	20.00	29
885-1120-[37]	HDOS 'WHEW' Utilities	20.00	33
885-1121	HDOS Hard Sec Sup Pkg 2 Disks	30.00	37
885-1123	XMET Robot Cross Assembler	20.00	40
885-1126	HDOS Utilities by PS:	20.00	42
885-1127-[37]	HDOS Soft Sector Support Pkg	30.00	45
885-1128-[37]	HDOS DISKVIEW	16.00	46
885-1129-[37]	HDOS CVT Color Video Terminal	20.00	46
885-8001	SE (Screen Editor)	25.00	28
885-8003	BHTOMB	25.00	28
885-8004	UDUMP	35.00	28
885-8006	HDOS SUBMIT	20.00	31
885-8007	EZITRANS.	30.00	30
885-8015	HDOS TEXTSET Formatter	30.00	42
885-8017	HDOS Programmers Helper	16.00	42
885-8024	HDOS BHBASIC Utilities Disk	16.00	46

CP/M

885-1210-[37]	CP/M ED (same as 885-1022)	20.00	20
885-1212-[37]	CP/M Utilities H8/H89	20.00	21
885-1213-[37]	CP/M Disk Utilities H8/H89	20.00	22
885-1217-[37]	HUG Disk Duplication Utilities	20.00	26
885-1223-[37]	HRUN HDOS Emulator 3 Disks	40.00	37
885-1225-[37]	CP/M Disk Dump & Edit Utility	30.00	40
885-1226-[37]	CP/M Utilities by PS.	20.00	40
885-1229-[37]	XMET Robot Cross Assembler	20.00	40
885-1230-[37]	CP/M Function Key Mapper	20.00	42
885-1231-[37]	Cross Ref Utilities for MBASIC	20.00	43
885-1232-[37]	CP/M Color Video Terminal	20.00	46
885-1235-37	CP/M COPYDOS	20.00	54
885-1237-[37]	CP/M Utilities	20.00	55
885-1245-37	CP/M-85 KEYMAP	20.00	63
885-1246[-37]	CP/M HUG File Manager & Utilities	20.00	64
885-1247-37	CP/M-85 HUG Bkgrd Print Spooler	20.00	67
885-5001-37	CP/M-86 KEYMAP	20.00	51
885-5002-37	CP/M-86 HUG Editor	20.00	52
885-5003-37	CP/M-86 Utilities by PS.	20.00	54
885-5008-37	CP/M 8080 To 8088 Trans. & HFM	20.00	64
885-5009-37	CP/M-86 HUG Bkgrd Print Spool	20.00	66
885-8018-[37]	CP/M Fast Eddy & Big Eddy	20.00	43
885-8019-[37]	DOCUMAT and DOCULIST	20.00	43
885-8025-37	CP/M-85/86 Fast Eddy	20.00	49

ZDOS/MSDOS

885-3005-37	ZDOS Etchdump	20.00	39
885-3007-37	ZDOS CP/EMulator	20.00	47
885-3008-37	ZDOS Utilities	20.00	47
885-3010-37	ZDOS Keymap	20.00	51
885-3022-37	ZDOS/MSDOS Useful Programs I	30.00	63
885-3023-37	ZDOS/MSDOS EZPLOT	20.00	63
885-3026-37	MSDOS SMALL C Compiler	25.00	65
885-3030-37	ZDOS/MSDOS Z-100 PC Emulator	40.00	68
885-3031-37	ZDOS/MSDOS Graphics	20.00	69
885-3034-37	ZDOS/MSDOS ZPC Support Pkg	10.00	72
885-8029-37	ZDOS Fast Eddy	20.00	53
885-8035-37	MSDOS DOCUMAT and DOCULIST	20.00	70
885-8041-37	ZDOS/MSDOS Orbits	25.00	75

H/Z100 ZDOS/MSDOS - H/Z150 PC MSDOS

885-3012-37§§	ZDOS HUG Editor	20.00	52
885-3014-37§§	ZDOS/MSDOS Utilities II	20.00	54
885-3016-37§	ZDOS/MSDOS Adventure	10.00	57
885-3020-37§	MSDOS HUG Menu System	20.00	62
885-3021-37§§	ZDOS/MSDOS Cardcat	20.00	63
885-3024-37§	ZDOS/MSDOS 8080 To 8088 Trans	20.00	64
885-3025-37§§	ZDOS/MSDOS Misc. Utilities	20.00	64
885-3029-37§§	ZDOS/MSDOS HUG Bg. Print Spool	20.00	66
885-3032-37§	MSDOS Halley's Comet Locater	20.00	70
885-3035-37§§	MSDOS SPELL5 & SPELL5F	20.00	72
885-8039-37§§	MSDOS DPATH	20.00	74
885-8040-37§§	MSDOS HELP Programs	20.00	74

§ All program files run on both

§§ Program files run partially on both

PC/IBM COMPATIBLE

885-6001-37	MSDOS Keymapper	20.00	59
885-6002-37	CP/EMulator II & ZEMulator	20.00	59
885-6003-37	MSDOS EZPLOT	20.00	65
885-6004-37	MSDOS CheapCalc	20.00	67
885-6005-37	MSDOS Skyviews	20.00	67
885-6006-37	MSDOS Cardcat	20.00	69
885-6007-37	MSDOS DND (Dung. & Dragons)	20.00	70
885-8033-37	MSDOS Fast Edit	20.00	62
885-8037-37	MSDOS Grade	20.00	70

PROGRAMMING LANGUAGES

HDOS

885-1038-[37]	Wise on Disk H8/H89	18.00	
885-1042-[37]	PILOT on Disk H8/H89	19.00	
885-1059	FOCAL-8 H8/H89 Disk	25.00	13
885-1078-[37]	HDOS Z80 Assembler	25.00	21
885-1085	PILOT Documentation	9.00	
885-1086-[37]	Tiny HDOS PASCAL H8/H89	20.00	13
885-1094	HDOS Fig-Forth H8/H89	40.00	18
885-1132-[37]	HDOS Tiny BASIC Compiler	25.00	59
885-1134	HDOS SMALL-C Compiler	30.00	63

CP/M

885-1208-[37]	CP/M Fig-Forth H8/H89 2 Disks	40.00	18
885-1215-[37]	CP/M BASIC-E	20.00	26

BUSINESS, FINANCE AND EDUCATION

HDOS

885-1047	Stocks H8/H89 Disk	18.00	
885-1048	Personal Account H8/H89 Disk	18.00	
885-1049	Income Tax Records H8/H89 Disk	18.00	
885-1055-[37]	MBASIC Inventory Disk H8/H89	30.00	
885-1056	MBASIC Mail List	30.00	
885-1070	Disk XIV Home Fin H8/H89	18.00	
885-1071-[37]	MBASIC SmBusPk H8/H19/H89	75.00	17
885-1091-[37]	Grade/Score Keeping H8/H89	30.00	14
885-1097-[37]	MBASIC Quiz Disk H8/H89	20.00	18
885-1118-[37]	MBASIC Payroll	60.00	30
885-1131-[37]	HDOS CheapCalc	20.00	47
885-8010	HDOS Checkoff	25.00	32
885-8021	HDOS Student's Statistics Pkg	20.00	44
885-8027	HDOS SciCalc	20.00	50

CP/M

885-1218-[37]	CP/M MBASIC Payroll	60.00	31
885-1233-[37]	CP/M CheapCalc	20.00	47
885-1239-[37]	Spread Sht. Contest Disk I	20.00	
885-1240-[37]	Spread Sht. Contest Disk II	20.00	
885-1241-[37]	Spread Sht. Contest Disk III	20.00	
885-1242-[37]	Spread Sht. Contest Disk IV	20.00	
885-1243-[37]	Spread Sht. Contest Disk V	20.00	
885-1244-[37]	Spread Sht. Contest Disk VI	20.00	
885-8011-[37]	CP/M Checkoff	25.00	32
885-8036-[37]	CP/M Grade	20.00	70

ZDOS

885-3006-37	ZDOS CheapCalc	20.00	47
885-3013-37	ZDOS Checkbook Manager	20.00	54
885-3018-37	ZDOS Contest Spreadsheet Disk	25.00	58
885-8028-37	ZDOS SciCalc	20.00	50
885-8030-37	ZDOS Mathflash	20.00	55

Continued on Page 83

Controlled Data Recording Systems Inc.

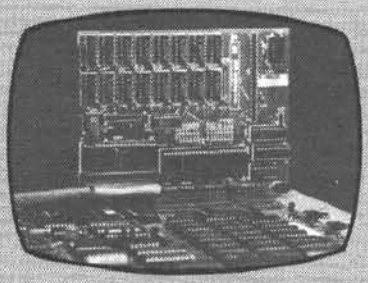
Quality Products and Support for the Heath/Zenith Community

New for the H/Z89-90 Computer Users:

The Super RAM 89 Package

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

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



Now get RAM 89

HDOS DRIVER Software Only **$35**

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

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

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

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

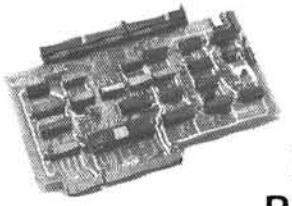
The Ram Drive Software (SRAM) allows one or two logical ram drives. The ram drive(s) can be located starting anywhere from logical A: to O: (standard drives get relocated). SRAM can be set to start at logical A: and warm boot with ram (no floppy disk accesses needed). Ram drive attatches to any of the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

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

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

FOR THE H/Z89-90 COMPUTERS

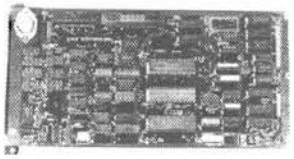
THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE:$395



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

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

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



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

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

MODIFY89

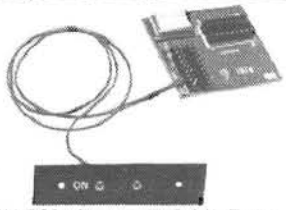
Allows owners of the FDC-880H or the FDC-H8 to read/write to the following disktypes:

Access S.S.D.D., Cromemco S.S.D.D., DEC VT180 S.S.D.D., IBM PC/Zenith 100, (CP/M) D.S.D.D.*, Kaypro II S.S.D.D., Morrow Micro Decisions S.S.D.D., NEC PC-8001 A S.S.D.D., Osborne S.S.D.D., Osborne S.S.D.D., Otrona D.S.D.D.*, Zorba D.S.D.D., TI Professional S.S.D.D., TRS-80 Model I (Omnikron CP/M), TRS-80 Model III (MM CP/M), Xerox 820 S.S.D.D., Zerox 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 II

Richard L. Mueller, Ph.D.
11890-65th Avenue North
Maple Grove, MN 55369

Overview

In my last article on the NEC V20 microprocessor chip, I covered a comparison of some operations using the 8088 chip and V20 chip. In this article, I will cover the enhanced instructions, the additional instructions, and the 8080 emulation mode of the V20 and V30 chips.

Some time ago, NEC came out with a new family of high-performance, low power CMOS, microprocessor chips, the "V" series. The ones that are interesting today are the V20 and V30 chips which are the replacements for the 8088 and 8086 chips, respectively. The instruction sets for the V20 and V30 are supersets of the 8088 and 8086, and are compatible with those earlier micros in terms of pins, functions, and object code. I'm not a "hardware" person, so I won't go into details of the chip hardware itself. I will discuss the software aspect of the chips; that is, the instructions and modes.

Modes

First, let me discuss the various modes of the V20/V30 chips. There are three modes: Native, 8080 Emulation, and Standby modes. All three modes are mutually exclusive; that is, the microprocessor can be in only one of three states at a time. In the Native mode (8088/8086), all the instructions of the 8088/8086, as well as the enhanced and additional instructions of the V20/V30, can be executed. In 8080 Emulation mode, the microprocessor executes 8080 code (programs). What this means is that CP/M-80 programs based on the 8080 instruction set can be executed. More on this later.

Standby Mode

The third mode (state) of the V20/V30 is the Standby mode. This places the V20/V30 in an idle state where the microprocessor consumes only 10% of its normal operating current while retaining all data necessary to keep the microprocessor operative. The only way to enter Standby mode is to execute an HLT instruction in either the Native mode or 8080 Emulation mode. The processor exits Standby mode in response to a RESET, NMI (nonmaskable interrupt), or an external INT (interrupt).

8080 Emulation Mode

Before discussing the Native mode new instructions, let me spend some time discussing the 8080 emulation mode. The V20/V30 chips have a special Mode Flag (MD) to select between the two operating modes. When initialized, the V20/V30 microprocessors are in Native mode and the Mode Flag is set to 1. In 8080 Emulation mode, the Mode Flag is set to 0. The Mode Flag is set and reset, directly and indirectly, by executing the mode manipulation instructions, three of which are new instructions.

Two of the new instructions are provided to allow one to switch the operating state from Native mode to 8080 Emulation mode and back again. The instructions are BRKEM (Break for Emulation) and RETEM (Return from Emulation). BRKEM is the basic instruction used to start 8080 Emulation mode. It operates basically the same as an INT instruction, except that the BRKEM sets the Mode Flag to 0. The Flags, Code Segment, and Instruction Pointer are all saved on the stack just as for an INT. The Interrupt Vector specified by the operand of the BRKEM instruction is loaded into the Code Segment and Instruction Pointer. The code pointed to by the Interrupt Vector is 8080 code and the CPU starts executing this code.

In 8080 Emulation mode, the registers and flags used are as follows:

	8080	8088/8086
Registers:	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	IP	IP
Flags:	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

In the Native mode, SP is used for the stack pointer, while the BP register is used for this function in 8080 Emulation mode. The SP, SI, DI, and AH registers along with the segment registers CS, SS, and ES are not affected by 8080 emulation mode operations. The Data Segment (DS) register is used in the Emulation mode for data and must be set to the Code Segment value by the user before entry is made to the 8080 Emulation mode.

When finished executing 8080 code, the execution of the RETEM instruction returns the operating mode to Native mode and sets the Mode Flag to 1. The microprocessor can now continue with the execution of 8088/8086 instructions.

NEC literature states that the V20/V30 microprocessors will run CP/M-80 (the version written in 8080 code and not in Z-80 code) operating system and programs. This is true, but it doesn't happen by magic. Since you are in the Native mode when the system is initialized and you are running MS-DOS (PC-DOS), you need a way to load CP/M-80 and 8080 programs into memory before you can execute the code. A small load program may be all that is necessary. However, there still is the question of disk files.

I haven't really looked into this very closely yet, but it may be necessary to convert CP/M-80 files to MS-DOS formats and utilize the second part of the 8080 Emulation mode which I will talk about shortly. Two companies have looked into this very closely and have products available for sale that will allow you to execute your CP/M-80 programs. Both of these companies have ADs in the February 1986 issue of REMark.

What I like about this 8080 Emulation mode is that while you are executing 8080 code, you can call an 8088/8086 routine by switching back to the native mode temporarily, execute the code there, and then return to 8080 Emulation mode. This is accomplished with a new instruction which is similar to the BRKEM and an existing instruction. The new instruction is the CALLN instruction which makes it possible to call Native mode subroutines. When the CALLN is executed, the Mode Flag is set to 1 and the Interrupt Vector pointed to by the operand of this instruction is loaded into the Code segment and Instruction Pointer registers. The microprocessor can now execute 8088/8086 code. Return back to 8080 Emulation mode by executing an existing instruction, the IRET (Interrupt Return) instruction.

To get a feeling how this switching between Native and 8080 Emulation modes works, I wrote a small test program in 8086/8088 assembly langauge that switched to 8080 Emulation to start processing a request, the 8080 code called a Native subroutine to complete the request, return to 8080 Emulation mode, and finally back to my main program in Native mode. The object of the program was to convert a series of binary numbers to ASCII decimal and display the numbers on the CRT.

The program sequence went like this:

- Set up Interrupt Vectors at 200 and at 201 to point to the start of 8080 Emulation code and Native mode subroutines, respectively.
- Place an 8-bit binary number in the AL register.
- BRKEM 200 — This causes the switch to 8080 Emulation code.
- The 8080 code starts the conversion process by determining what the hundreds digit is, converts that to ASCII, and saves it in the C register (CL). The remaining portion of the original number is left in the A register (AL).
- CALLN 201 — This causes a switch to Native mode to take place and the conversion processes continue. The tens digit is converted to ASCII and stored in the AL register (A) and the units digit converted to ASCII is stored in the CH register (B).
- IRET is executed by my Native mode subroutine to return to 8080 Emulation mode with the original number now fully converted to ASCII decimals stored in the registers indicated above.
- RETEM is executed by the 8080 Emulation mode to switch the state back to Native mode.
- The result is now displayed on the console CRT.
- The above sequence is repeated a number of times, each time using a different 8-bit integer to convert to ASCII decimals.

Now you are going to ask how I entered 8080 code into my program. Well, the way I did it, because the number of 8080 instructions were but a few, was by using 'DB' statements containing the HEX code of the 8080 instructions. I first wrote a small subroutine in 8080 code, used the 8080 assembler on CP/M-85 to assemble the code, got the HEX code from the listing, and put that into my 8086/8088 assembly program using the 'DB' statements. It's a way, but not necessarily the best way. However, it works. Another way would be to write a small disk read subroutine into your Native program that would read a file containing 8080 code and store it into your code segment where you want it. I'm sure there are other ways, as well. This little test program was tried on both of my Z-Machines and worked fine in both cases.

Enhanced Instructions

In addition to the "standard" set of 8086/8088 instructions, the V20/V30 have the following "enhanced" instructions:

- PUSH im — Pushes immediate data onto the stack.
- PUSH R — Pushes the contents of the four 16-bit general registers onto the stack (AX, BX, CX, DX).
- POP R — Pops the four 16-bit general purpose registers from the stack.
- MUL reg16,im16 — Multiplies the contents of a 16-bit register by 16-bit immediate data.
- MUL mem16,im16 — Multiplies the 16-bit contents of a memory location by 16-bit immediate data.
- SHL reg,im — Shifts specified register left by immediate value.
- SHR reg,im — Shifts specified register right by immediate value.
- SAR reg,im — Shifts specified register 'arithmetic right' by immediate value.

- ROL reg,im — Rotate specified register left by immediate value.
- ROR reg,im — Rotate specified register right by immediate value.
- RCL reg,iin — Rotate specified register left through carry by immediate value.
- RCR reg,im — Rotate specified register right through carry by immediate value.
- CHKIND reg16,mem32 — Checks array index against designated boundaries. This is added for support for high-level languages. The index value is in reg16 and the lower limit is in location mem32 and the upper limit is in mem32+2.
- INM — Used to input a string into memory when preceded by a repeat prefix (REP). The address for storing the string is contained in the DI register and the I/O port is specified in DX.
- OUTM — Used to output a string from memory when preceded by a repeat prefix (REP). The address of the string is contained in the SI register and the I/O port is specified in DX.
- PREPARE im16,im8 — Generate stack frames required by high-level languages such as PASCAL and ADA that use block structures. This instruction provides the facilities of creating and linking the stack frames. The stack frame is pointed to by the BP register. The stack framer is composed of a local variable area (im8) and a copy area for frame pointers (im16).
- DISPOSE — This instruction releases the last stack frame generated by the PREPARE instruction. It returns the stack and base pointers to the values they had before the PREPARE instruction was used to call a procedure.

Additional Instructions (Unique)

In addition to all the "standard" 8086/8088 instructions and the enhanced instructions above, there are a number of unique instructions in the V20/V30 microprocessor chips. A number of these unique instructions have been added to support advanced business applications (packed BCD string operations); and to support engineering work stations, computer graphics, higher-language computing environments, and record-type data structures used in high-level languages (bit field manipulations).

- INS reg8,reg8 — This instruction transfers low bits from the AX register (the number of bits specified by the second operand) to the memory location specified by the ES Segment Register, plus the byte offset specified by the DI register with the bit offset specified by the low 4-bits of the first operand.
- INS reg8,imm4 — Same as the preceding instruction.
- EXT reg8,reg8 — This instruction loads to the AX register the bit field data whose bit length is specified by the second operand from the memory location specified by the Data Segment Register DS with the byte offset specified by the SI register and the bit offset by the lower 4-bits of the first operand.
- EXT reg8,im4 — Same as the preceding instruction.
- ADD4S — This instruction adds the packed BCD string addressed by the SI register to the packed BCD string addressed by the DI register, and stores the result in the string addressed by the DI Register. The length of the string, number of BCD digits) is specified by the CI register.
- SUB4S — This instruction subtracts the packed BCD string addressed by the SI register from the packed BCD string addressed by the DI register, and stores the result in the string addresses by the DI register. The length of the string in BCD digits is specified by the CL register.
- CMP4S — This instruction performs the same operation as SUB4S, except that the result is not stored and only the Overflow Flag, Carry Flag, and Zero Flag are affected.
- ROL4 — This instruction treats the byte data of a register or memory location specified by the instruction byte as BCD data and uses the lower 4-bits of the AL register to rotate that data one BCD digit to the left.
- ROR4 — this is the same as the preceding instruction except that it rotates the data one BCD digit to the right.
- TEST1 — Tests a specified bit in a register or memory location and resets the Zero Flag to 0 if the bit is a 1, and sets the Zero Flag to a 1 if the bit is a 0.
- NOT1 — This instruction inverts a specified bit in a register or memory location.
- CLR1 — Clears a specified bit in a register or memory location.
- SET1 — Sets a specified bit in a register or memory location.
- REPC — Repeats the next instruction until the Carry Flag becomes cleared or the CX register becomes zero.
- REPNC — Repeats the next instruction until the Carry Flag is set.
- FPO2 — Currently performs the same function as the ESC instruction used in conjunction with other processors, such as the 8087 coprocessor.

Other V-Series Chips

The only thing left to cover in this article is to say a few words about some of the other chips in the "V" series that are either available or will be in the future. At the time of writing this article, I did not have the current status of the other chips. However, I will talk about them briefly based on the literature that I received from NEC.

The V25 is a single-chip microcomputer aimed at portable microcomputer applications. The processing unit is the V20 chip with on-chip ROM and RAM for the application programs. It also has on-chip timers and DMA and Serial Interface Channels.

The V40 and V50 microprocessors integrate four independent DMA channels, three programmable timers, programmable interrupt controller, and on-chip clock generator along with the V20 and V30, respectively.

The only other chips in this series, that I am aware of, are the V60 and V70. The information that I have at this time is a little sketchy since I was not able to get much information on these from NEC. However, I do have some general information on them. These are 32-bit general-purpose microprocessors that realize mainframe functions on a single chip. The performance estimation at this time is from 1 million to 3 million instructions per second. It will support up to 4 gigabytes of virtual memory space with fast translation from virtual addresses to physical (or real) addresses. More information, I'm sure will be available later this year.

Conclusion

Hopefully, this article and the previous one will give you some idea of what the NEC V20 (and other "V" series chips) microprocessor chip is, how it performs with relation to the 8088 chip, what are some of the new instructions or enhanced instructions, and how the 8080 Emulation mode works.

*

Performance.

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

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

Quality.

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

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

Availability.

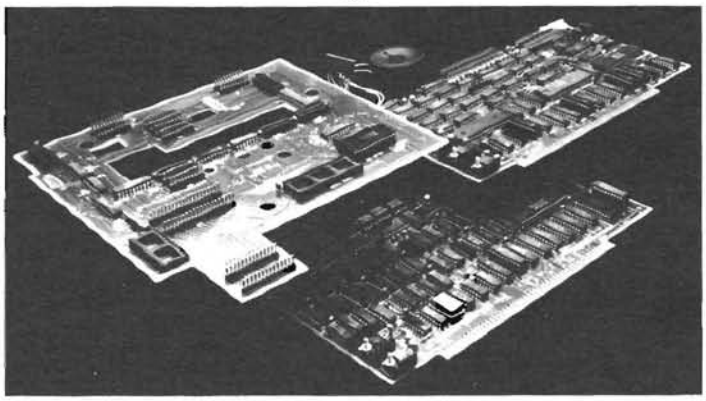
EasyPC is ready when you are. If you've been waiting too long for PC compatibility, your wait is over. EasyPC is here now. Complete documentation is included for quick and easy installation, so you'll be up and running fast. Your only problem will be which PC programs to get first.

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

Price.

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

EasyPC™



Full IBM PC emulation for your Z-100.

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

8" Capacity In A 5-1/4" Space

A Hardware Review Of The CDR 5-1/4" Superdrives

Jim Buszkiewicz
HUG Software Engineer

I've always felt that the 8" single-sided, single-density standard format that IBM established some years ago was a good idea, and it really was while it lasted. Fortunately, the large bulky 8" disks gave way to the more compact 5-1/4" variety. These newer, smaller drives had a total capacity of around 90 to 120 kilobytes. Shortly thereafter, came double-density drives, doubling the capacity, and finally double-sided, increasing the capacity to around 320 to 360 kilobytes per diskette. As technology marched on, 96 tpi (tracks per inch) drives emerged, increasing disk capacity even further.

Today, a new type of 'high-tech' drive has emerged. These drives have the capacity of the older style double-sided, double-density 8" drives! In fact, to an 8" controller board, they appear exactly like 8" drives! These new super drives are capable of 1.2Mb (1.2 million bytes) of storage and will connect directly to the H/Z-100 floppy controller card. It should also be mentioned, that if you're running the CDR FDC-880H controller card in your H/Z-89 system, these drives will also function in a similar manner. No hardware or software modifications are even necessary. Think of it, you can now store almost 4 times the amount of data on a single 5-1/4" floppy than you can with the standard floppies which originally comes with the computer.

If you're still running two floppies on your H/Z-100, and can't quite justify the expense of a hard disk, consider CDR's new super drives as an alternative. The increased capacity of these drives will now allow you to put all your 'C' compiler files on a single drive or even your 'Professional WordStar', and text files!!

Because of the extended density, high density diskettes must be used. If you decide to use STANDARD double-sided, double-density diskettes, a certain amount of BAD sectors will show up during the formatting procedure. These bad sectors will be marked and NOT used during the useful life of that diskette. Although 'high-tech' diskettes are more expensive than the ones you're used to buying, the increased price factor is not the same as the increased storage factor. As of the time of this writing, Heath Company is not carrying 'high-tech' diskettes in their product line, although many Heath/Zenith Computer and Electronic Centers are.

What I liked most about these drives was the fact that in order to install them, all I had to do was plug the interconnecting cable into my H-110, and turn on the power. They immediately worked with my existing MS-DOS and CP/M-86! CDR has just lowered the price on this super drive subsystem to $550. This price includes TWO superdrives, cabinet, power supply, and interconnecting cables! For more information, call (619) 560-1272, or write:

CONTROLLED DATA RECORDING SYSTEMS INC.
7210 Clairemont Mesa Boulevard
San Diego, CA 92111

CDR also has an information bulletin board system you can call at: (619) 560-8929. *

The H89 *SPEED* Center

4MHz mod $45

An easy to install plug-in module. No trace cutting or soldering. Speed may be toggled with software. Includes a replacement Z80A (4MHz). Includes CP/M software support for Heath, CDR Systems and Magnolia. Call or write for info on HDOS support. Specify disk format.

6MHz mod $59

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

CDR Super RAM 89

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

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

CDR Controllers

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

For the H89 FDC-88OH $349

REP3 - Automatic Key Repeat

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

Kit $35
Assembled $45

TIM2 - Real Time Clock

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

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

DATESTAMPER

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

CP/M - specify disk format $45

WSPATCH $20

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

NEWWORD - the better word processor from NewStar Software

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

CP/M - specify format $115
MSDOS - specify computer $195

EMULATE

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

Actrix	Eagle II	Morrow MD	Superbrain Jr
AMPRO	Epson QX-10	NCR DecMate 5	Televideo
Beehive Tpr	Fujitsu CP/M86	NEC PC-8001A	TRS80-1 CP/M
CDR	IBM CP/M86	Osborne 1	TRS80-3 CP/M
Cromemco	IMS 5000	Otrona	TRS80-4 CP/M
DEC VT180	Kaypro II	PMC MicroMate	Visual 1050
DEC Rainbow	Magnolia	Royal/Adler	Xerox 820
		Sanyo 1100	Zorba

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

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

DISKETTES

High quality private label diskettes at an economy price.

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

ANALYTICAL PRODUCTS 209/564-3687

20663 Ave. 352 Woodlake, CA 93286

Specify Disk Format on Software

CALL OR WRITE FOR CATALOG PRICES SUBJECT TO CHANGE

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

ZPC Version 2 Is Here!

Pat Swayne
HUG Software Engineer

By now, many of you are familiar with ZPC, the program I wrote that allows you to run IBM PC programs on an H/Z-100 dual processor computer. Some of you have paid me some nice complements about the program and its capabilities, but, as they say, you ain't seen nothin' yet! ZPC Version 2 is now ready, and it has so many improvements over the first version that I am not sure that I can remember all of them for this article. But that is what this article is about, so I will try.

Here Are The Improvements In ZPC Version 2

- Emulation of the monochrome adapter card. In addition to emulating the color/graphics card, as the old ZPC does, ZPC Version 2 emulates the monochrome text adapter card, also. Many PC programs that require patches before they will run in the color/graphics mode will run unpatched in the monochrome mode. Nearly complete emulation is provided, including normal and intense video, and underlining.

- Improved emulation of normal and intense video. If you use the new monochrome mode, or one of the black and white modes of the color/graphics emulation, and you have color memory installed, ZPC will use green or yellow (your choice, configurable) to represent normal video, and white to represent intense video. The result is very pleasing, whether you have a monochrome or color monitor.

- Improved SETZPC utility. The SETZPC utility, used to configure how ZPC emulates certain colors, etc., has been improved greatly. You can configure many more parameters, and you can make your changes either temporary or permanent.

- User selectable fonts. You can configure the character font used by ZPC while it is in the PC emulation mode. You can use either an IBM-style font, the standard Z-100 font, or any custom font you may have designed using the FONT editor that comes with Z-100 MS-DOS. The font you choose affects only the alphanumeric characters and standard symbols. The graphic symbols and special characters remain in the IBM style, for maximum compatibility.

- Support of video pages. ZPC Version 2 supports 4 video pages in the 80 column text modes, and 8 video pages in the 40 column modes, the same as a real PC. It does not require 64k color memory chips, as some "brand x" PC emulators do, but works fine with 32k chips.

- On screen Num Lock indication. A small white "blip" in the upper left corner of the screen indicates when Num Lock is on. The indication is true whether you toggle Num Lock yourself, or a program changes the state of it in memory. The indication is, therefore, even more accurate than the led on the Num Lock key of a Z-100 PC series computer.

- Faster scrolling. When it is scrolling the full screen, ZPC Version 2 is much faster than the old version. Some other screen changes are also faster.

- Disk interrupt emulation. ZPC Version 2 emulates the BIOS disk interrupt of an IBM PC. This allows it to read some (but not all) copy protected disks. If the copy protection method uses altered sector sizes and/or sector numbering, ZPC can read it. ZPC Version 2 can also run unprotected programs that access the disk via the BIOS interrupt, which could not run under the old ZPC.

- Command line editing. ZPC Version 2 supports command line (template) editing, which was not available while the old version was in the PC mode.
- Menu driven patcher. Some programs require patches before they will run under ZPC. With the old ZPC, patching was done using DEBUG and batch files. A menu driven patcher is provided with ZPC Version 2, which patches programs quickly and easily. The patcher uses a data file that is in ordinary text format. When patches for new programs appear in REMark, you can easily add them to the file using a text editor or word processor.
- Full support of the ZHS hardware. ZPC Version 2 fully supports the ZPC Hardware Support circuits described in last month's REMark. With ZHS, you can run many programs right out of the box.
- Compiled BASIC programs with sound run ok. Under the old ZPC, compiled BASIC programs that generated sounds had to be patched before they would run. Under ZPC Version 2, they run without patching.
- Previous bugs fixed. Bugs and oversights discovered in the original ZPC have been fixed in Version 2. None of the fixes presented in previous ZPC Update articles are required to make ZPC Version 2 run any supported program.

As you can see, ZPC Version 2 provides some features not supported by any other emulation method, and even some features not supported by a real PC. See the New HUG Products pages of this issue to find out how you can order ZPC Version 2, or upgrade from version 1. Included in the product description is a list of programs that run under the new ZPC as of 3-13-86. *

The Joys And Tribulations Of Perfect Writer

Richard Kelly
136 Sunrise Drive
Knoxville, TN 37919

I was first introduced to Perfect Writer in 1982. It came as part of the bundled software with my Kaypro II. Given the limited memory of my computer (64k), this word processor was amazingly powerful, and due to an innovative idea called "virtual memory", it enabled me to write and edit very long documents. As a writer who frequently wrote chapters ranging from thirty to fifty pages in length, this feature alone endeared the program to me. I continued to use Perfect Writer for a year and then decided to buy a Zenith Z-150 and sell my Kaypro. At that time, Perfect Software was making products for CP/M and MS-DOS machines, and I reluctantly learned WordStar 3.3. I have since come to understand and enjoy WordStar better than all of my other word processors, and if it could do four or five operations better than it does now, it would be more nearly perfect for me than Perfect Writer.

In any event, after working with WordStar for a year, I discovered that a British firm named Thorn EMI had purchased Perfect Software and that they were now offering Perfect Writer for MSDOS machines, and furthermore, this new version contained many features not found on the old one. I bought a copy and proceeded to experiment with it for several weeks, and before long I became an open advocate. I persuaded my department at the university where I teach to make this its official software. (Today some of the secretaries thank me, some curse me). Having now spent nearly a year working with the program on and off, I have gotten to know its foibles and imperfections, and can testify now that this is a fine word processor, but that it is certainly not perfect.

Whereas the old Perfect Writer relied upon elaborate commands (there being no function keys on the old Kaypros), the new version takes advantage of the function keys for certain commands, such as "Backword Word," "Cut Previous Word," "Fill Paragraph," and "Save to Disk." Most commands, however, are given through a series of pop-up menus. Pressing the Escape Key activates a main menu, and through this menu one can access a total of twenty-eight submenus. If, for example, you wanted to insert page breaks in your manuscript, you would press Escape to call up the main menu, then press "Appearance" for a submenu, then press "Document" for another submenu, and finally, press "Insert page break" in the last submenu in that branch. After you have threaded your way through this labryinth several times, the menu popping becomes quite easy, although sometimes unhappily time-consuming.

The main advantage of the pop-up menus is the ease with which this program can be mastered. At that point, however, they begin to get in the way and slow you down. The appearance of the menus can be delayed by up to nine seconds, allowing you to execute memorized commands more speedily. For example, if you wanted to fill a paragraph without moving through a series of menus, you would simply press the Escape key, then "A" (the first letter of the word "Appearance," which would have appeared in the submenu), then "P" (for "Paragraph," which would have appeared in the second submenu), and finally "F" (for "Fill," which would have appeared in the last submenu).

The virtual memory architecture of Perfect Writer allows you to edit extremely long documents, a hundred pages or more. You would be much wiser, of course, to work within the confines of several shorter documents — for the sake of speed of operation — and then merge them together either before or during printing. Nevertheless, this feature is useful if your computer has a small Random Access Memory. Perfect Writer uses a "swap file" to enable you to edit a document larger than your RAM. A "swap file" is a designated area on your disk which Perfect Writer uses to store portions of your document that will not fit into your RAM. Perfect Writer comes with a 40K swap file, which is suited only to short documents of about six to eight pages. The swap file can easily be expanded in increments of 8K up to 248K, but such a large file requires that you remove the speller, the printing facility, and the main selection menu from your edit disk. I have chosen to make an 80K swap file which, with 320K RAM on the Zenith, allows me to work easily with documents of thirty to forty pages. Any increase in the size of the swap file over the initial 40K requires that something be deleted from the edit disk (which has 0 bytes remaining). The obvious choice for removal is the spelling checker, which can happily reside on a separate disk.

Perfect Writer allows you to edit multiple documents through a two-window display. After selecting "2 Windows" from the submenu, the screen is divided into two areas, one on top of the other, and a different document can be displayed in each window. You can then easily copy or move text between the two documents displayed.

A Thesaurus diskette, supplied along with the Edit, Lessons and Installation diskettes, enables you to call upon a 50,000 word thesaurus for synonyms. Since this is an on-line thesaurus, you simply remove your document diskette from drive B, insert the Thesaurus diskette in its place, position the cursor on the word in your document (which is now held in RAM), and the Thesaurus will search for alternative words for the one you have selected. If you find a suitable synonym, you press "Replace" and the new word will be inserted into your text. As with any thesaurus, this feature is useful only in so far as you have a firm sense of the connotations of the various synonyms. I believe I could use a book thesaurus more quickly than I could exchange diskettes in drive B, call up the submenu, position the cursor, and select a replacement word. Nevertheless, some people may find this a useful feature. I don't.

The spelling checker which comes on the Edit diskette of Perfect Writer also contains about 50,000 words. Spelling checkers, in general, are extremely useful devices for catching both misspellings and typos. Perfect Speller, however, is not a satisfactory program, and I do not recommend you use it except for the most casual correspondence. The following misspellings/typos were not found by Perfect Speller: "ordinery," "occurence," "coperation," "corperation," "feable," "sufficent," "occurrance," and "ocurrence." I discovered these significant misses in a half hour and decided not to go on. With such misses, a spelling checker loses its credibility and is useless for anyone doing serious writing. I then checked these misspellings with another spelling checker, Word Plus, and all of them were caught. Word Plus, incidentally, contains only 45,000 words in its dictionary, but it has proven to be a much more reliable spelling checker. (The spelling checker which comes with WordPerfect, by the way, contains 100,000 words, is amazingly swift and accurate, and it even catches double words — a common typing error, such as "the the"). My recommendation would be that if you decide to buy Perfect Writer and want a spelling checker that actually works, that you also buy Word Plus, which is fairly inexpensive and very accurate and flexible.

Perfect Writer's document design commands are one of its truly outstanding features. By inserting commands prefixed by the "@" character, you can shape your document in practically any imaginable form. "@I(The Andy Griffith Show)," for example, will lead that title to be printed in italics. Typeface, margins, line spacing, footnotes, new page, batch files, page heading and page footing, indexing, superscripting/subscripting, and centering text are among the formats controlled by the "@" commands. Although boldface and underlining can be seen on the screen, when you use the "@" commands, you do not see what you get until your document is formatted before printing. You can then preview the document on screen, if you wish, before printing it out. The manual, which is written in remarkably clear prose, gives excellent examples of the effect each "@" command produces. Once these commands are mastered, they save a great deal of time, especially for someone working in an office where there are several standard formats in use. Once you have found a number of "@" commands which design your letter, memo, or manuscript format to your specifications, it is a good idea to use the "Print Screen" to record the series of prefixed commands to use later.

Although the Perfect Writer manual is one of the best I have ever used, it fails to inform you of one important fact. If you are using footnotes, you must use a justified right margin. A typical footnote is entered as follows in your text: "@Footnote (Richard Kelly, Graham Greene, Ungar, 1984, p. 23)." Unless you select Justified Right Margin before printing, a gap will be left in your text the length of the parentheses. I write all of my books with a ragged left right in order to avoid the gaps between words that a justified right margin inevitably causes. Typesetters have enough problems without my adding to them. And so, I do not use the footnote commands provided by Perfect Writer; rather, I simply type my footnotes out on the computer at the end of my manuscript. The inability to use a ragged right margin is a failing in the program. Furthermore, the number of words that Perfect Writer allows in each footnote is too limited for serious scholarship, which usually needs extended footnotes.

The self-teaching software that comes with Perfect Writer is very good indeed. It takes you through the program in logical fashion and allows you to learn the basics in a couple of hours. This diskette, combined with the manual, is a model of clear instruction. Despite the powerful features of Perfect Writer, it is a program designed to be learned easily. In order to use its more advanced features, however, you will need to spend a great deal of time in experimentation. In the event you encounter a seemingly unsolvable problem, Thorn EMI provides a free telephone customer service for ninety days after you purchase its program. I have used this service on two occasions and found the technicians to be prompt and knowledgeable in solving my problem.

The system requirement for Perfect Writer is 128K RAM, and DOS 1.1 or later, and two disk drives. A larger RAM, of course, would enable you to work with a smaller swap file. Although this new version of Perfect Writer is copy protected for CP/M, it is not copy protected for DOS. You receive three diskettes: an Edit diskette, containing both Perfect Writer and Perfect Speller; a Thesaurus diskette; and a Lessons and Installation diskette. After you make copies of these three diskettes, it is a good idea to remove the speller from the edit diskette and to increase the swap file of the Edit diskette. This is very easily accomplished by entering the following information at the A>: pw-swap 80 <CR>. The "80," of course, could be replaced by any figure, in increments of 8, that would be appropriate to your needs.

Configuring your printer is the next step. Perfect Writer comes with 39 popular printers already defined. All you have to do is to insert your Edit diskette in drive A and your Installation diskette into drive B and type "b:myprint." The list of predefined printers will appear and if you are lucky enough to find your printer there, you simply select it, and its configuration is automatically placed on your Edit diskette. If your printer is not there, more than likely it corresponds to one that is — Silver Reed, for example, emulates the Diablo 630. Of course, you always have the option of defining your own printer or editing the codes for any of the predefined printers.

Perfect Writer gives you two options for changing the default settings for such items as margins, space between paragraphs, type of footnotes (bottom or endnotes), line spacing, etc. Each of these settings can be changed by using "@" commands prefixed to the document you are writing. If you want the default settings changed permanently, then you simply call up the default style parameters through your Installation diskette. The eighteen settings will be listed on-screen and you can choose to change any or all of them. These changes will then be permanently recorded on your Edit diskette.

As was noted earlier, Perfect Writer is a powerful word processor and can do practically anything you want it to. Here are some of its major capabilities not already discussed: edits seven multiple documents, renames files, copies files, copies and moves text between files, saves text without exiting program, files are readable by DOS, can swap disks during editing, can chain documents for printing, prints part of file by page, automatic page numbers, prints multiple copies, prints without saving, prints sub/superscripts, prints italics, supports proportional spacing, reforms paragraphs (but there is no global command for this important function), moves cursor by word, sentence, paragraph, and screen, moves cursor from end/beginning of line and end/beginning of document, returns to cursor position after a save, deletes by word, sentence, paragraph, page (but not to the end of the document), searches and replaces both forward and backward.

Most of Perfect Writer's operations are accomplished with great speed. Reformatting text, however, is a laborious process since the Fill command works only paragraph by paragraph. (WordPerfect, for example, has a Rewrite command that reformats your entire document in a flash). Other desirable features absent in Perfect Writer are automatic page breaks, notations on screen as to the column, line, and page you are working on. Although you can call up a ruler line and insert page breaks through the menus, this is an awkward procedure and one which still does not indicate the page or line you are on. A serious problem with the Search and Search/Replace function is that it fails to pick up any words separated by a line break. For example, Search quickly located "Merry Christmas" when the two words were on one line, but when "Merry" happened to be the last word in a line and "Christmas" the first word in the next line, Perfect Writer failed to locate the phrase.

One of the features of Perfect Writer that I take special delight in is its ability to scroll quickly through a long document. Unlike WordStar, which has wait periods every three pages or so, Perfect Writer continues to scroll through page after page with rapidity. Furthermore, it moves from one end of a document to the other with lightning speed. WordStar, which continuously reads from disk instead of RAM, seems to take forever to make such giant leaps.

I have discovered one very annoying bug in Perfect Writer. If you accidentally hit the Alt-key instead of the Shift-key when making a capital letter — something easy to do since the two keys sit one right next to the other — the entire keyboard locks up. After an initial panic, I learned that pressing the Insert-key frees the keyboard up again. This problem needs to be corrected or at least Perfect Writer customers need to be notified as to how to get out of this problem, since it could lead to the loss of an unsaved file.

I have also noticed some peculiarities about Perfect Writer's predefined printers. I own both a Gemini 10X and a Star SG 10 (an updated Gemini). Even though the Gemini 10X is one of Perfect's predefined printers, it does not have the proper code for continuous underlining — each letter of a word is underlined but not the spaces between the words. When I use the Star SG 10 with the Gemini 10X definition, however, Perfect does produce a continuous underline.

A recent review of Perfect Writer in PC gives fair praise to the program but recommends it for personal rather than office use. My conclusions are just the opposite. The enormous range of document controls, once noted down, make this an ideal word processor for offices, especially those that use a variety of formats for their documents. For personal use, I would recommend WordPerfect or WordStar 3.3, despite some of the drawbacks of the latter. These two programs, unlike Perfect Writer, come much closer to allowing you to see on screen what your final document will look like. Furthermore, they both show you where you are in your document at all times. I like to see double spacing, page breaks, paragraphs set off from the text, a title actually centered on the page, etc. Perfect Writer does not show these things even though it accomplishes the same effects. On the other hand, I find myself cursing WordStar for its snail-like scrolling while I am editing my text on screen. I am emotionally attached to WordStar, but intellectually I am beginning to suspect that Perfect Writer may be the most nearly perfect word processor currently available. Perhaps a year from now I shall be able to inform you that I have, at last, found the perfect writer.

*

HUG NEW PRODUCTS

HUG P/N 885-1135-[37]
HDOS Variety Package $20.00

Introduction: This group of three disks contains various programs for the HDOS operating system. Rather than hold these programs for an 'all the same type of program' disk, the decision was made to release them in a 'variety' type package. These programs include, utilities, games, and two device drivers.

Requirements: These programs require the HDOS operating system version 2.0 or greater. An H-8/H-19 or H-89/90 computer with at least one disk drive is also required. In the case of the device drivers, the H/Z-25 or Diablo 630 printer will be needed. MBASIC Version 4.82 is required for SMART CRAPS.

The following files are included on the HUG P/N 885-1135-[37], HDOS Variety Package disk set:

Disk A

FREDIT	.ABS	WM25DVD	.ACM
FREDIT	.DOC	WM25DVD	.DVD
WM25DVD1	.ABS	WM25DVD	.DOC
WM25DVD1	.ASM	DEMO	.TXT
WM25DVD2	.ABS	README	.DOC
WM25DVD2	.ASM		

Disk B

CALENDAR	.BAS	HBR	.DOC
CALENDAR	.BAS	PD	.ASM
SNOOPY	.ART	PD	.DVD
HBR8H	.ABS	PD	.DOC
HBRZH	.ABS		

Disk C

CRAPINS1	.BAS	CRPSTA1	.DAT
CRAPINS2	.BAS	MPCRAPS	.BAS
CRAPS	.BAS	CRAPS	.DOC
CRAPSTAT	.BAS	SPCRAPS	.BAS
CRPSCR1	.DAT		

Program Authors:

FREDIT — Maynard Mansfield
WM25DVD — William W. Moss MD.
CALENDAR — Brent Malcolm
HBRIDGE — Robert Hassard
PD — Walt Bilofsky
SMART CRAPS — C.F. Mowery Jr.

Program Content:

FREDIT — Fredit makes available to the small system running under HDOS, the speed and simplicity of full-screen text editing. The editor occupies less than 5700 byte in RAM, so serious text editing may be undertaken on a 32k machine. Fredit makes use of all the editing functions provided by the H-19 terminal, and most of the H-19's special function keys are employed as

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.

well. The 25th line of the terminal screen provides a continuous display of the status of the editing process.

WM25DVD — This H-25 device driver is considerably more sophisticated than the one Heath distributes. It permits individual settings on each of eight logical units. Thus, the user can individualize the formats desired for various types of documents without having to change the device driver settings. Also, when a new setting is made, it goes into effect immediately even if the driver has already been loaded into memory; no need to reboot to make it work. There are 18 set options allowed by this driver.

CALENDAR — This program computes a calendar for any year and prints it in various forms. It prints an annual calendar with, if desired, an attached graphics of Snoopy on his doghouse. It also prints a monthly appointment calendar (one month per page) which has the usual format for all days of the week (Sunday through Saturday) or a second type with just Monday through Friday. The latter format allows larger spaces for your entries.

HBRIDGE — HBRidge is a program for playing Bridge. It is suitable for the novice bridge player. The computer plays three hands (or two if north or south are the declarer) and the human player plays one (or two). HBRidge bids using fundamental conventions, plays following the simplest of rules, and then scores each hand when finished. The score is recorded on a disk file bearing the human player's name. After playing a hand, a display of all four hands may be obtained, or the north-south hand may be switched with east-west and replayed, or a new deal may be called for. With this program, the computer plays a fair quality of bridge. It makes mistakes just as humans do. In this respect, it is human-like. For a person who also makes mistakes, this can be an enjoyable program.

PD — This is a device driver for the WH-44 style printers (Diablo 630) to allow the use of proportional spacing type print wheels even when NOT equipped with the Diablo word processing hardware option. It is a modified version of the LPH44 driver originally supplied by Heath with HDOS version 2.0.

SMART CRAPS — This game is designed not only to provide realistic and entertaining one-player and multi-player versions of the dice game, Craps, as it is played in most casinos, but also to assist both novice and experienced players alike in becoming 'smart' craps players, via the display of a wide variety of statistical information during play of the games and a separate statistical program. The data displayed is designed to help smart players develop betting systems that increase their winning potential by decreasing the house advantage.

TABLE C Rating: (9)

HUG P/N 885-3037-37 MS-DOS
ZPC Version 2 ..... $60.00

Introduction: ZPC Version 2 is a program that emulates an IBM PC or compatible computer on an H/Z-100 series (dual processor) computer. It allows you to run many IBM PC programs on your H/Z-100 without having to add an expensive hardware modification. It supports all video modes, including text and graphics, of an IBM color/graphics card. Version 2 of ZPC is a significant enhancement of the original ZPC (HUG p/n 885-3030-37), and with it you can run much of the important PC business software. A list of programs that will run under ZPC Version 2 as of 3-13-86 is supplied later in this description.

Note: This version of ZPC supersedes both the original ZPC and the ZPC Support Disk (885-3034-37). If you have ZPC version 1, you can upgrade to version two by sending your original distribution disk and $20.00 to Heath Users' Group, Attn: Nancy Strunk, Hilltop Road, St. Joseph, MI 49085. If you have both ZPC version 1 and the ZPC Support Disk, you can upgrade by sending both disks and $15.00 to HUG. Make checks payable to: Heath Users' Group.

For a description of the improvements in ZPC Version 2 compared to version 1, see the article "ZPC Version 2 is Here" in this issue.

Requirements: ZPC Version 2 requires an H/Z-100 or ET/ETA-100 series computer with MS-DOS version 2 or 3 and exactly 768k of memory. A small memory version of ZPC is supplied that will run in less than 768k of memory, but that version will only run a few PC programs. For best results, your computer should be equipped with color memory (either 32k or 64k chips).

The ZPC Version 2 disk contains these files:

README	.DOC	DISK	.ACM
ZPC	.COM	DOS	.ACM
ZPCSM	.COM	KEY	.ACM
PC	.COM	PIXEL	.ACM
Z100	.COM	PRINT	.ACM
SETZPC	.COM	PUTCHR	.ACM
ANSISYS	.COM	SCROLL	.ACM
SETANSI	.COM	PC	.ASM
PATCHER	.COM	Z100	.ASM
DEMO	.COM	SETZPC	.ASM
PATCHER	.DAT	ANSISYS	.ASM
FIXCB	.COM	SETANSI	.ASM
FIXQB	.COM	PATCHER	.ASM
FIXPSC	.COM	FIXCB	.ASM
FIXFWII	.COM	FIXQB	.ASM
ZPC	.ASM	FIXPSC	.ASM
COND	.ACM	FIXFWII	.ASM
DATA	.ACM	DEMO	.ASM

Author: Patrick Swayne, HUG Software Engineer.

ZPC.COM, ZPCSM.COM — These are the two versions of ZPC. ZPCSM.COM is for small memory systems (less than 768k), and ZPC.COM is for 768k systems. ZPC emulates the color/graphics adapter, the monochrome text adapter, the keyboard, printer I/O and disk I/O of an IBM PC or compatible computer.

ZPC is a background program that remains resident in memory after you run it. Once it has been loaded, you can turn on PC emulation to run IBM programs, and turn it off to run Z-100 programs. You do not need a second version of MS-DOS to run IBM programs under ZPC, and all your drives and partitions are supported.

Some programs require patching before they will run under ZPC. A patching utility is included, along with patches for several programs. ZPC Version 2 fully supports the ZPC Hardware Support (ZHS) circuitry that was described in the April 1986 issue of REMark. With ZHS installed, most supported programs do not have to be patched.

ZPC Version 2 can read some, but not all, copy protected programs. In particular, it cannot read a disk protected by SoftGuard. Fortunately, there are commercial programs available that let you use SoftGuard protected programs.

Continued on Page 81 ☛

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	April Issue Price	HUG Member Price	Qty Left
H-19-3	H-19 to H-88 Mod Kit	220.00		199.95	16
H-88-1	Floppy Disk Accessory	150.00		99.95	50
H-88-5	Cassette I/O Card	25.00	24.95	14.95	75
H-88-9	Drive Mounting & Inst	25.00	24.95	14.95	27
H-89A	W/O CRT, Ser I/O, Disk Cont.	----		399.95	10
HCA-9	Print Stand For H-125	99.00	79.95	69.95	19
PA-100-1	USR Telpak (PC&ZDOS)	99.00		79.95	10
PC-121-50	TEC Captain 320K (PC)	399.00		349.95	2
PC-131	USR PCM-5 Modem (PC)	349.95	299.95	269.95	47
PC-140	DG Magic RAM Expn. 0K	129.00	49.95	19.95	8
PC-200	USR Modem Card (S100)	399.00	329.95	299.00	56
PC-310	DG Super "89 CPU	699.00	399.95	349.95	29
PCA-120-1	6-PAK Plus Game Acc'y.	39.95	29.95	24.95	18
PP-110	Trippe Emer. Power Sys.	349.00	249.95	199.95	1
PP-210	DP P132 Color Printer	1795.00	999.00	799.00	1
PP-215	EPSON JX-80	699.95	499.95	499.95	38
PP-260	DTC 380Z Daisy Printer	1295.00	699.95	599.95	14
PP-400	Koalapad Touch Tablet	99.95	79.95	69.95	154
PP-401	WICO 50-0110 Joystick	39.95	29.95	24.95	12
PPA-210-1	Black Ribbon	12.95	6.99	5.49	23
PPA-210-2	Primary Color Ribbon	24.95	9.99	5.49	91
PPA-210-3	Process Color Ribbon	24.95	9.99	8.99	61
PPA-215-1	Ribbon: Epson JX-80	15.95		12.95	26
PPA-260-2	Cut-Sheet Feeder/380Z	574.95	349.95	299.95	3
PPA-260-3	Ribbons (3Pk) /380Z	9.95	4.99	3.99	237
PPA-260-4	Cable - RS-232/380Z	49.95	19.99	9.99	27
PPA-260-5	Cable - Parallel/380Z	49.95	19.99	9.99	20
PPA-260-6	Wheel - Brougham 10	26.95	9.99	4.99	46
PPA-260-8	Wheel - Anelia Prop.	26.95	9.99	4.99	29
PPA-260-9	Wheel - Script 10/12	26.95	9.99	4.99	41
PPA-260-20	Wheel - Quadro 15	26.95	9.99	4.99	68
PPA-270-1	Trac Feed - Stylewriter	149.95	89.95	69.95	2
PPA-270-2	Sheet Feed - Stylewriter	249.95	129.95	99.95	42
PPA-270-4	Black Ribbons (3Pk)	16.95	4.99	3.99	256
PPA-270-5	Red Ribbons (2PK)	6.95	3.99	2.99	38
PPA-400-1	KOA Speedkey Software	59.95	49.95	29.95	166
ZA-100-4	S-100 Extender Board	59.00	29.95	19.95	298
ZA-219	Amber CRT /H-121	75.00	17.95	14.95	575
ZG-219	Green CRT /H-121	75.00	17.95	14.95	136
ZW-219	White CRT /H-121	75.00	14.95	9.95	152

Heathkit®
Heath Company

A subsidiary of Zenith Electronics Corporation.
Some items may not be available at all store locations.
Prices, product availability and specifications subject to change without notice.

CP-236

Intelligent Navigation By HERO

S. K. Tanejo
Auburn University
Auburn, AL 36849

A mobile Robot, like HERO, should be able to navigate around the barriers and reach a target in an unexplored environment. The program included with this article gives you an opportunity to judge HERO's capability yourself.

Initially, HERO requests permission to start his navigation. Any loud sound is a signal for him to start. He rotates around and stops wherever maximum light intensity is detected. He straightens the wheel and starts moving in that direction. He takes out his "Laser gun" if any barrier is detected and gives warning. If barrier is not removed, HERO tries other ways of reaching the target by looking around. Several subroutines added for making HERO talkative make his navigation fun to watch.

```
0401    3F          Robot Language
0402    72 04 70    JSR 1 'Request for Permission'
0405    BD 04 83    JSR 2 'Check for Permission'
0408    72 FB 6B    speak,wait "I heard that"
040B    72 FD 46    speak,wait "Your wish is my command"
040E    BD 04 9A    JSR 3 'Check for direction of
                    maximum light intensity'
0411    72 FB 37    speak,wait "I see light"
0414    72 FA F1    speak,wait " and I can move about."
0417    BD 04 C4    JSR 4 'wheel straightening and
                    moving forward.'
041A    BD 04 D0    JSR 5 'check for range < 40H'
041D    02          Abort drive
041E    71 04 F2    JSR 6 'speak,continue "stand back
                    or I will blast you with my laser
                    gun."'
0421    C3 4C 35    Raise manipulator arm.
0424    C3 3C 30    Extend the arm up.
0427    C3 7C 30    Rotate the wrist
042A    C3 BC 20    Open the gripper
042D    C3 B8 00    Close the gripper to the original
                    position
0430    C3 78 00    Rotate back the wrist to the
                    original position
0433    C3 38 00    Retract the arm
0436    8F 00 08    Pause for 3/4 second
0439    DC 08 FF    Move forward
043C    BD 04 D0    JSR 4 'check for range < 40H'
043F    02          Abort drive
0440    72 FB 7B    Speak,wait "There is something in
                    my way "
0443    C3 F4 02    Turn wheel left 90 degrees abs
0446    C3 08 05    Rotate anticlockwise
0449    7E 05 59    JMP
044C    C3 0C 05    Rotate clockwise
044F    7E 05 59    JMP
0452    01 01       NO-OPS
0454    BD 04 9A    JSR 2 'Check for direction of
                    maximum light intensity '
0457    BD 04 C4    JSR 3 'wheel straightening and
                    moving forward '
045A    BD 04 D0    JSR 4 'Check for range value < 40H '
045D    02          Abort drive
045E    72 FB 37    Speak,wait "I see light "
0461    72 FC 9D    Speak,wait "Gosh! I think I am just
                    about perfect."
0464    72 05 40    JSR 7 'Speak,wait "Do I start over?"'
0467    BD 04 83    JSR 1 'Request for permission '
046A    7E 04 0B    JMP 'Restart'
046D    01 01 01    NO-OPS
```

Subroutine 1 — Request For Permission To Start

```
0470    1E 36 36 37       Do
0474    3F
0475    15 00 09 29       I
0479    3F
047A    1F 2A 15 2B 2A    Start?
047F    3F FF
0481    01 01             NO-OPS
```

Subroutine 2 — Check For Voice Command

```
0483    42          Enable sound detector
0484    8F 00 10    Pause for 1 second
0487    B6 C2 40    LDA
048A    BD F6 4E    JSR REDIS
048D    BD F7 AD    JSR OUTBYT
0490    0C          CLC
0491    81 D0       CMPA with D0
0493    25 F2       BCS
0495    52          Disable sound detector
0496    39          RTS
0497    01 01 01    NO-OPS
```

Subroutine 3 — Checking Direction Of Maximum Light Intensity

```
049A    C3 F4 00    Turn wheels 90 degrees left, wait,
                    abs
```

```
049D    41          Enable light sensor
049E    DC 08 55    Rotate body, 360 degrees, cont
                    relative
04A1    B6 C2 40    LDAA with value in C240
04A4    1C 02       Branch if busy
04A6    20 07       BRA
04A8    81 C2 40    CMPA with sense port
04AB    2B F4       BMI
04AD    20 F5       BRA
04AF    DC 08 C0    Continue to spin
04B2    41          Enable light sensor
04B3    4A          DECA
04B4    B1 C2 40    CMPA with sense port
04B7    2A FB       BPL
04B9    02          Abort drive
04BA    8F 00 10    Pause for 1 second
04BD    51          Disable light detector
04BE    D3 10 0D    Turn body forward to align with dir-
                    ection of maximum light intensity
04C1    39          RTS
04C2    01 01       NO-OPS
```

Subroutine 4 — Straightening Wheel And Moving Forward

```
04C4    C3 F0 47    Straighten wheels.
04C7    8F 00 10    Pause for 1 second
04CA    DC 08 FF    Move forward
04CD    39          RTS
04CE    01 01       NO-OPS
```

Routine 5 — Checking For Range

```
04D0    45          Enable ultrasonic ranging
04D1    8F 00 02    Pause for 1/8 second
04D4    96 11       LDAA
04D6    BD F6 4E    JSR REDIS
04D9    BD F7 AD    JSR OUTBYT
04DC    81 40       CMPA 40H 'Check for < 40H'
04DE    25 0F       BCS
04E0    8F 00 02    Pause for 1/4 second
04E3    D6 11       LDAB
04E5    BD F6 4E    JSR REDIS
04E8    BD F7 AD    JSR OUTBYT
04EB    C1 40       CMPB 40H 'Check for < 40H'
04ED    24 E5       BCC
04EF    5B          Disable ultrasonic ranging
04F0    39          RTS
04F1    01          NO-OPS
```

Subroutine 6 — Laser Warning

```
04F2    5F 6A 6F 40       "Stand
04F6    4D 5E
04F8    CE EF C0 D9       back
04FC    3E 5E
04FE    34 34 2B          or
0501    15 00 09 29       I'll
0505    18
0506    4E 58 6F 40       blast
050A    5F 6A
050C    22 36 37 37       you
0510    2D 0B 09 39       with
0514    0C 15 00 09 29    my
0519    98 86 A1 A9       laser
051D    92 42 7A
0520    5C 73 4D          gun"
0523    3E
0524    1E 1A 1E 1A
0528    1E 1A 1E 1A       Laser gun sound
052C    1E 1A 1E 1A
0530    1E 1A 1E 1A
0534    1E 1A 1E 1A
0538    1E 1A
053A    3E 3F FF
053D    01 01 01          NO-OPS
```

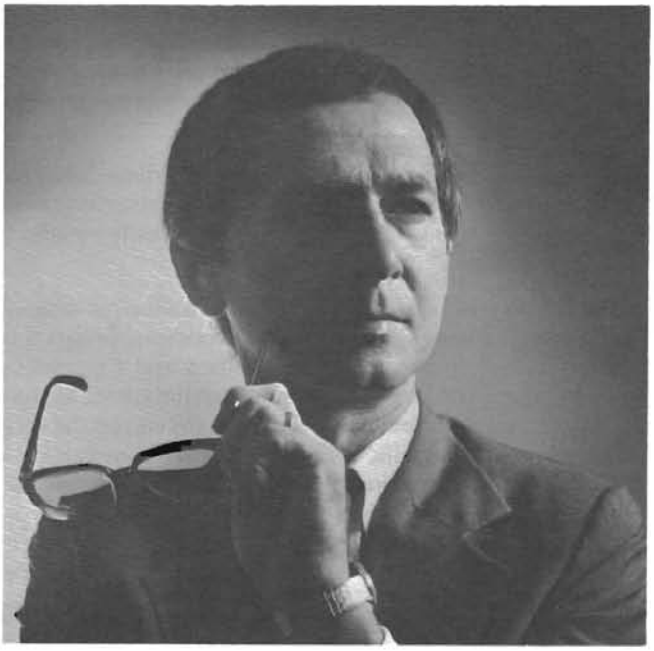
Subroutine 7 — Request For Permission

```
0540    1E 36 37 37       "Do
0544    3F
0545    15 00 09 29       I
0549    3F
054A    1F 2A 15 2B 2A    Start
054F    3F
0550    35 34 0F 3A       over?"
0554    3F FF
0556    01 01 01          NO-OPS
```

Subroutine 8 — Look Around And Check For Obstacles

```
0559    45          Enable ultrasonic ranging
055A    8F 00 02    Pause for 1/8 second
055D    96 11       LDAA
055F    BD F6 4E    JSR REDIS
0562    BD F7 AD    JSR OUTBYT
0565    8F 00 02    Pause for 1/8 second
0568    D6 11       LDAB
056A    BD F6 4E    JSR REDIS
056D    BD F7 AD    JSR OUTBYT
0570    81 60       CMPA 60H 'Check for < 60H'
0572    24 06       BCC
0574    C1 60       CMPB 60H 'Check for < 60H'
0576    24 02       BCC
0578    20 0A       BRA
057A    BD 04 C4    JSR 4 'Straightening wheels and
                    moving forward.'
057D    BD 04 D0    JSR 5 'Check for range < 40H'
0580    02          Abort drive
0581    7E 04 54    JMP
0584    7E 04 4C    JMP
```

*



Mainstream Computing

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

Although this column is being published under the auspices of the Heath/Zenith User's Group, it is not really "about" Heath/Zenith computers. It's about using mainstream software and hardware with mainstream microcomputers. I'll be writing about the kinds of computers IBM calls the "PC", "XT", and the "AT." Heath and Zenith call those same kinds of computers the "148," "150," "158," and "241." More are sure to come, and when they do I'll cover them.

What you can expect here is explorations of mainstream software, such as Microsoft's Windows and Borland's Turbo Lightning, and mainstream hardware, such as Iomega's Bernoulli Box and Intel's Above Board. Software, hardware, books, accessories, programming and other tools for mainstream computers — I'm interested here in exploring their uses and benefits. What are they? What do they do? Are they worth the money?

If you think that this column is the kind you would expect to find in an IBM-specific magazine like PC Magazine, PC World, or PC Tech Journal, you understand me very well.

But my perspective is different. Those magazines assume — as do most software and hardware vendors — the perspective of the IBM world. What you'll find here is the perspective of the Heath/Zenith world. Those two perspectives are close, but not the same.

There are significant differences between Heath/Zenith microcomputers and the other compatibles, whether produced by IBM or any other company. Those differences do not necessarily affect the functions of either manufacturer's machines, but ignoring the differences can be costly. As a Heath/Zenith user, you need special attention to those differences. You simply will not get that attention, however, from a magazine or organization that doesn't have the proper perspective and appropriate sources of information. They can't give it to you even if they want to.

Here's an example of what I mean. PC Magazine's "PC Advisor" is kind of a "Dear Abby" for owners of IBM machines. It's a good column, truly good. So to one owner of a Heath/Zenith compatible it seemed the right place for advice on adding memory to his computer:

I own a Zenith Z-150 PC. I read in another publication that you can make a memory upgrade directly to the Z-150 motherboard using 256K chips and so save buying an additional board and/or using 64K chips. If so, please advise me of a publication or supplier I could consult. — PC Magazine (11 February 1986, p. 23)

The PC Advisor's advice was to buy an auxiliary memory board because "the motherboard, which comes with 128K bytes of RAM, can recognize only a total of 640K bytes (any additional memory up to a total of 640K bytes must be on an expansion board) . . . ."

Valid advice about an IBM machine, it is costly advice about a Heath/Zenith: First Capitol Computer sells RAMPAL, two replacement PALs ("Programmable Array Logic" chips) and instructions for plugging them into the Z-150 CPU board so it can take all 640KB of RAM. The catalog number is SWI-RAMPAL, the price $39.95 — substantially less than the cost of a memory board and, as a bonus, one does not waste a slot on the backplane. (The H/Z-158, of course, already has the PALs for putting 640KB on the CPU board. Get your 256KB chips and plug them in.) You surely do need a Heath/Zenith perspective to get the most from your computer system.

Take the reset switch as another example. Neither IBM nor Heath/Zenith provides one on its PC/XT/150/158 machines. You can add one to any of these machines (although it's easier on the Heaths and Zeniths), but you can't do it the same way for both. Try it and you will fry the computer. After a reset switch has been properly installed on either brand of machine, however, you use it the same way with the same results. I'm interested in results. You probably are too.

My real point, though, is that while similar kinds of machines — all of them mainstream computers — will function similarly, they often do their functioning in different ways. The concern in this

column is with both the functions and the functioning — simultaneously. You can't get that kind of concern in any of the magazines with a PC-specific focus.

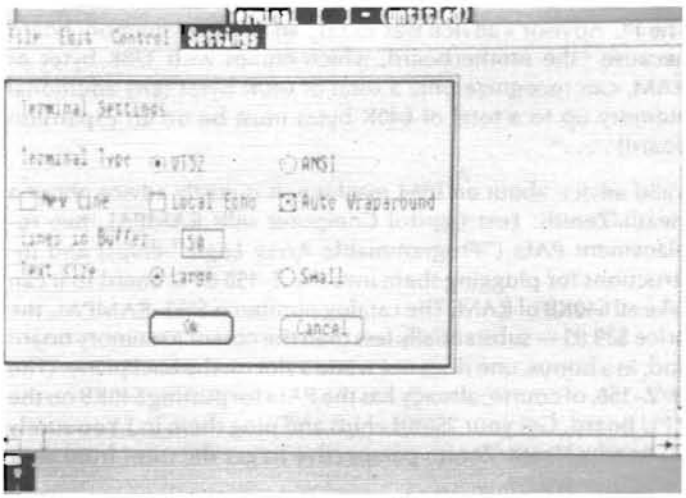
If you think I've scuttled away from my opening statement, the one that probably shocked you, you're wrong. I meant just what I said.

My focus here is on doing things with standard microcomputers made by Heath/Zenith. I'm not interested, for example, in playing the game called "Will-it-run?" That game was appropriate for a while after Heath/Zenith started making the H/Z-150, in part because the rules governing compatibility were still being established. Every user of every machine imitating an IBM PC or PC-XT played the game, and it was great fun. But now it's an outdated game. I expect standard software to run in my standard machines -- my mainstream Heath/Zenith computers.

Unfortunately, Will-it-run? also is a terribly destructive game. It reinforces the conviction held by many vendors that the Heath/Zenith market lies somewhere on the fringes of the mainstream. That's why most of them don't tell you about new products: it's bad marketing to aim at a small target instead of a big one. You're part of the big one; but, since few vendors know or believe that, you've been relegated to the fringes.

Of course, you can't afford to take a chance on buying a product that won't run on your machine, and no number of generalities will — or ought to — convince you to take that gamble. I'll do the gambling instead. You'll see me in the process of running the stuff for you. If I find any incompatibilities, I'll say so. Expect that in most cases I won't find any. Recognize, though, that in either case I'm not playing Will-it-run? I'm using the product.

Let's get on with it.



Setting Microsoft Windows Terminal.

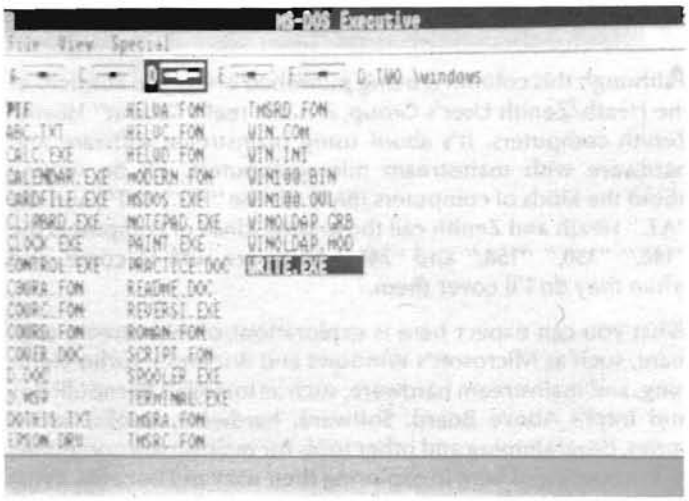
Microsoft Windows Is A Big Pane

You may remember talk a few years ago about how Microsoft's forthcoming new product would solve the compatibility problems among 16-bit machines that didn't follow the IBM PC "standard." Those days we put quotation marks around "standard" because doubters and damned fools really didn't think that the PC would be. I was somewhere in the pack of damned fools: I had no doubts at all; I was sure no one sensible would accept the PC as a "standard." I knew about the H/Z-100. Chalk one up for me.

At any rate, Microsoft announced Windows on 10 November 1983 and for almost the next two years gave demonstrations at trade shows of what it could do when eventually they released it. My best look at one of those demonstrations was at COMDEX in the fall of 1984, and boy-oh-boy, did the forthcoming new product look good.

One idea was for Microsoft Windows to mediate between the operating system and applications programs, to serve as an interface, so differences among 16-bit machines really wouldn't isolate the mavericks.

Another idea was for Microsoft Windows to allow, or to simulate, multitasking. You could have a word processing program open in one window, a spreadsheet in another, and a graphics program in a third. Hot dog! Now, for the first time, it would be easy to enliven a report about why your company was on the verge of bankruptcy, with specific numbers to prove your calculations, graphs to make them comprehensible, and cute pictures of just where in Argentina you had stashed the assets.



Microsoft Windows MS-DOS Executive: Microsoft Windows Write up for selection.

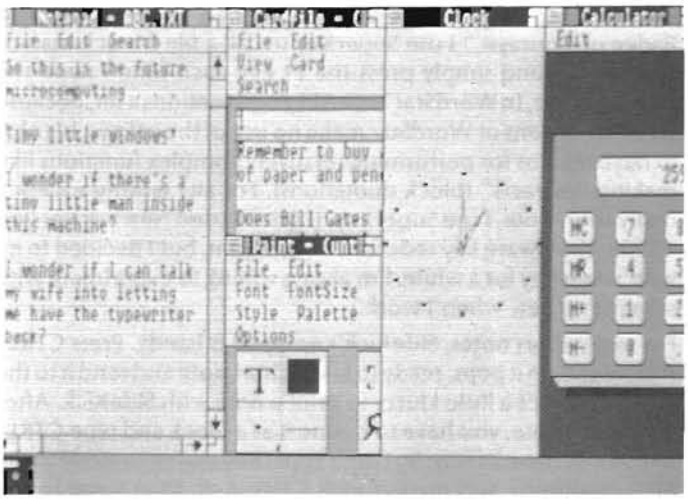
Joking aside, I fell in love. I wanted to possess Microsoft Windows. Failing that, I wanted to own a copy. Since that was not yet possible, because the product was still in development, I at least wanted to try it out then and there. When the demonstrator reacted as if I had just asked for the loan of his toothbrush, I was crushed.

So when I finally got Windows, I dropped everything and ripped open the package: four paperback manuals of varying thickness, a Windows Update booklet, just about the biggest "Quick Reference Card" I'd ever seen, and five diskettes.

Windows is HUGE. If you have only floppy disk drives, forget about Windows: there is no way on earth you are going to be able to use it. Someone at Microsoft has an odd sense of humor that led to instructions in the User's Guide on how "To Set Up Windows on a Two-Drive System." Forget it unless you enjoy disk swapping. If you have a hard disk, be prepared to devote a

big subdirectory to Windows: the installation program put 103 files in my H-158's.

Remember that Windows is only an operating system environment, so you need room for DOS itself. I assume, too, that you might like to do something useful with your computer after you have DOS and Windows going. If so, you need room for your applications programs also. The humorist who conceived that joke about being able to use Windows on a two-floppy-drive system must be slapping his thighs at the thought of you using, say, dBASE III as your database, Microsoft Word as your word-processing program, and Lotus 1-2-3 as your spreadsheet! Big programs all, and copy-protected too. By the time you get these things ready for you to do anything, you'll wonder what it was you wanted to do. It's a thigh-slapper, all right.



Microsoft Windows: Write, Cardfile, Clock, Calc, and Paint (five windows — the utilimatee pane).

With a nice, big, hard drive you'll certainly be able to do things even with that combination of software — if you have enough RAM. Windows GULPS memory. You need RAM to do anything real with Windows: the 640KB in my machines is about enough to play with Windows Paint and Windows Write — two applications programs Microsoft supplies with Windows in "Special Offer!" packages — and the Notepad, Cardfile, Terminal, Calendar, Clock, and Reversi, most of which seem a response to folks who have come to depend on programs like SideKick.

The minimum requirement just to open your Windows with MS-DOS 2.0 is 256KB. You need more with DOS 3.0 or for running multiple applications. In any case, you need 320KB of RAM for Windows Write.

Even so, everything slows down to a crawl periodically. You probably need an H/Z-241 with an Intel Above Board — lots of extended memory — for Windows to be more than a novelty. I'll try it sometime.

First, though, I'll want a mouse. Sure, you can do things from the keyboard alone, but that is nuts-making. There are pull-down menus for everything. With just the keyboard available, some choices are accepted by pressing the RETURN key and some with the spacebar. I can't keep in my mind just which is which, perhaps because I am overly preoccupied with mundane tasks such as writing — the kind of thing for which I bought a computer in the first place.

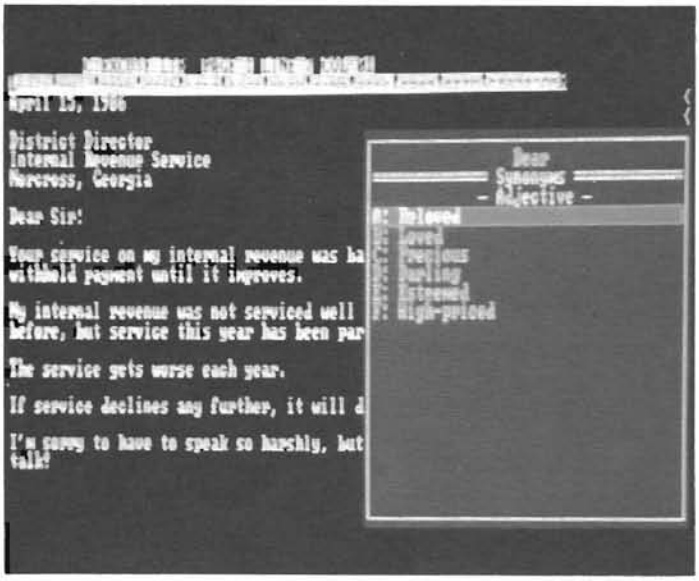
Another reason why I'll need a mouse before trying Microsoft Windows again is that my manual dexterity is even weaker than my memory. Janet is a Graphics Designer and the Media Consultant for the University's College of Business Administration. She also is a closet computer freak, so badly hooked that she shows me ads in computer magazines and explains in tedious detail why I need a particular product, most of which somehow turn out to have something to do with graphics or design or typesetting or media. When she surprised me at the end of a long afternoon in which I was working on an elaborate drawing with Windows Paint, my first thought was that I'd better save the thing to disk and let her take over the controls. But she looked at my drawing on the screen, laughed, and shook her head "No." She still has chuckling fits. I guess I'll have to start looking seriously at mice.

Microsoft Windows Paint — Katz's "Great Art Talent Test"; or, "Look Ma, No Mouse!"

Oh yes, about the "compatibility" that Windows would bring. Not yet. It's available only for machines that follow the Windows standard — that, in other words, already are completely compatible. Those are computers like the H/Z-150, H/Z-158, and H/Z-241, computers like the COMPAQ 286, and of course computers like the IBM PC XT and IBM PC AT. (There are problems, however, with the COMPAQ 286 and MS-DOS 3.00, so upgrade to 3.1 if you have that machine. "Compatibility" is a perception, not an absolute.)

As far as program compatibility goes, I've had no trouble running any of the software I've tried so far. But there's a catch: a BIG catch. For a program to run so you can use more than one window in Windows, it must be "well behaved." It must not do things like write directly to the screen. To put that requirement a different way, programs must be compatible with Microsoft Windows! My first thought was that a programming editor like Edix ought to be a fair test of software not on Microsoft's bill-of-fare that I could get Microsoft Windows to accept, but no soap. The only way I could get Edix going was to have Windows let it take over the entire screen — with no windows. That sort of misses the point.

Write off my tone as that of the lover falling to earth. As I see Microsoft Windows right now, it offers a vista of one possible future for microcomputing, but not a vision of what is practical today on an XT compatible. Expect the $99 price for Windows to be the first, and smallest, outlay for making the product really useful. But I'm not ready to call it curtains on Windows yet. After all, I'm just a prisoner of love.



Borland's Turbo Lightening checking a WordStar file.

Borland's SuperKey, Turbo Lightning, And SideKick

The order is important. I find the combination of those three Borland International RAM-resident products extremely useful when I sit down for a serious writing session nowadays, but one must be careful setting up to use software like this — the kind that sits in memory waiting to be called into action. These products, and others like them, monitor various interrupts and there's the potential for a slugfest when two or more of these things have to fight it out.

So, if you decide to run SuperKey, Lightning, and SideKick all at once, make sure you load them in just that order. I keep a subdirectory on my hard disks reserved for RAM resident software. They go there only after really thorough testing for safety.

First, I put the new stuff on a system floppy, and boot from it. That way I don't risk the hard disk if something goes crazy. After a while I add to that floppy any of the tried-and-true RAM residents I plan on using with the new stuff, and give those combinations a workout. Only after I feel confident that nothing dangerous will leap out to bite me in the back of the neck do I put the new software in a temporary subdirectory on the hard disk.

Oh yes, I also make sure there's no full moon out and that I do nothing at the stroke of midnight during any of this preliminary testing. Sometimes I forget to wear the bag of camphor, but I'm not sure it really does any good. Superstitious? Who, me? You just don't recognize CAUTION when you see it in action. As things are, I blow hard disks more often than onion slicers blow noses.

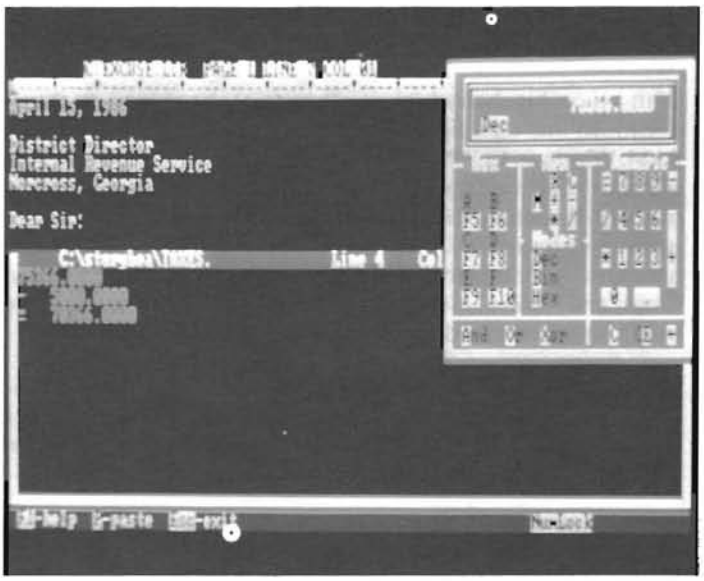
Well, the combination of Borland stuff passed all my tests some time ago and was installed in my permanent RESIDENT subdirectory. When I want to use all three programs at once, I run a batch file called "BORLAND.BAT." Here it is:

```
ECHO OFF
echo Loading SuperKey
key >nul
echo Loading Lightning ..
light >nul
echo Loading SideKick
sk >nul
```

That way I don't have to remember the right order in which to load the darned things when I want them all at once. Now, here's why I load them all at once.

SuperKey (I have version 1.11A) does all sorts of snappy things, but what I depend on it for is the ability to bind frequently used words and phrases to a set of keys. Then I don't have to retype the same characters over and over and over. I think that Stephen Crane's, "The Red Badge of Courage" is a marvelous novel in addition to being a classic, but I have typed the phrase "Stephen Crane's, The Red Badge of Courage" enough to last me several lifetimes. So when I write about "Stephen Crane's, The Red Badge of Courage," I use SuperKey to call a file of such macros (RBC.MAC) and simply press the F1 key each time I need the whole phrase. In WordStar I use ALT combinations a lot, because current versions of WordStar make no use of those keys. It's nice to have macros for performing relatively complex functions like making "extracts" (block quotations). For any writing except a really short note, I use SuperKey. (I briefly tried Newkey Version 1.1, the shareware key redefinition program, but I decided to go with SuperKey for a while. I'm always willing to experiment with something new when I work.)

For really short notes, SideKick's notepad is handy. Press CTRL-ALT-N and up it pops, ready to take a short note and send it to the printer. But it's a little klutzy to print a note with SideKick. After typing the note, you have to define it as a block and type CTRL-K-P to send it on its way. So I use a SuperKey macro to move to the file's beginning and mark it with CTRL-K-B, then jump to the file's end with CTRL-K-K.



SideKick's Calculator using a SuperKey "calculator tape" macro with the SideKick Notepad in a WordStar file.

(If those look like WordStar commands to you, they ought to: SideKick's notepad uses many WordStar-like commands. In fact they're really more like the commands in Borland's Turbo Pascal editor, which in turn are adaptations of WordStar commands. As Li'l Abner used to say, "Amoozing but confoozing.")

I use SideKick's notepad a great deal in WordStar, especially since version 1.05 added the capability for exporting text. Earlier versions could only import text. Press F4 to import text from a screen display, define the desired part as a block, then type CTRL-K-C and it pops into the notepad — stripped of screen attributes like reverse video, but with the graphics characters preserved. (You can see them in the notepad if you type CTRL-Q-G, which also allows typing graphics characters in the notepad. It's a quick way to build pretty screens for a program.)

To export text from the notepad to the screen for capture by another program, you sort of reverse the process. Define a block in the notepad, type CTRL-K-E, and then define a key to trigger the function that emits the marked block. You can have it zap out the entire block without a pause, or dole it out a line at a time. I use both ways, depending on what I'm doing. The block mode is good for exporting text into WordStar (WordStar protests the speed by sending out a confused howl of exclamation points, but settles down when the export is complete.) The line mode is useful in creating kind of a super "batch" file to feed MS-DOS command lines into a program that doesn't understand pipes or redirection.

Mostly, though, I use the import/export features of SideKick's notepad to make it function something like the Macintosh or Microsoft Windows clipboard. That capability is extraordinarily useful for pulling text from one file to another, or from a spreadsheet to a document, or from messages in the HUG SIG on Compuserve into a file for later reference.

The SideKick/SuperKey combination is heightened, incidentally, by a sample macro supplied with SuperKey to emulate a "calculator tape." When both programs are loaded and you execute the macro, anytime you do a calculation in the SideKick calculator its steps are recorded on the "tape" in the notepad. It's mighty handy to be able to zap that tape to the printer as a permanent record of how you arrived at a result.

I find it especially useful in programming, when I need to calculate memory allocations. It's great to have printouts of just how I arrived at each miscalculation that made my computer run amok. Some day, when son Matthew is ready for his first C compiler, I'll be able to hand him my folders of calculations and say, "My boy, these are the mistakes I made in life. Learn from them." Every young man deserves a legacy.

Lightning is the latest Borland product. You've seen the blaring ads for it everywhere. Lightning is a combination spelling-checker and thesaurus. Go ahead and laugh at the notion of an English Professor needing either. Sure I need them. I am a clumsy typist, which is the reason for the spelling checker: it catches typos for me. As for the thesaurus, like most writers I tend to fall into patterns during a prolonged writing session. The same words get repeated, where synonyms would relieve your boredom and mine. I use the thesaurus. If you haven't used one, buy it in book form. The best time to use an online computer thesaurus is when your vocabulary is good enough so you don't really need one.

That's no joke. It's professional advice. What you should know, though, is that none of the thesaurus programs currently available on microcomputers are a substitute for a good, old-fashioned thesaurus in book form, and none of the respectable thesaurus programs really pretend to be. What the good ones do, at this stage in the development of this genre, is suggest a list of words that fall somewhere in the same denotative realm as the word you're checking. If your vocabulary is good, the list serves as an excellent way to jog your memory about alternatives — perhaps those with connotations better suited to the situation.

But if your vocabulary is bad — if, for example, you're not sensitive to connotations — use of a computerized thesaurus will make you appear semiliterate. Should anyone want to know why, I'll be happy to explain at some other time. Ask me nicely and I might be persuaded to explain why computer spelling checkers aren't really spelling checkers.

For someone with a good vocabulary, though, the Lightning thesaurus is nice. I don't think it is as nice yet as Writing Consultants' Word Finder (formerly Synonym Finder), but that's a specialized tool with only one function: finding synonyms. Sometimes it offers more synonyms than Lightning, sometimes fewer. My impression is that it offers synonyms for more words. If you don't need Lightning's RAM resident spelling checker, and you already have a good program like V-Spell or CorrectStar, and you want a good RAM-resident thesaurus, you ought to consider Word Finder. Unless I'm having an extraordinarily tough go and my brain has congealed to the consistency of leftover grits, Lightning is adequate for me.

Moreover, the spelling checker in Lightning is really good. I haven't detected any misspellings yet, and the algorithm for popping up "close" spellings is really good. Because it is good, Lightning likely will have the right spelling for the word you meant if the way you wrote it is anywhere close to the way other people spell it.

Remember I said that Lightning is RAM resident. It has been programmed so it can monitor each of your keystrokes and beep if it doesn't recognize the set of characters just completed when you hit the spacebar or other indicator of a word ending — punctuation marks such as a period, question mark, and exclamation point, for example. The beeps are irritating for a while each session, then I don't notice them at all. Hmm . . . if I don't notice them, what's their value?

You can install Lightning for beepless action, and you can toggle "auto-proof" on and off from a menu or by pressing ALT-F9. There are all sorts of installation options, so Lightning can be set to work in various "environments": the DOS command line, all popular word processing programs, Lotus 1-2-3, and terminal programs, for example.

Don't confuse "environments" with "supplementary dictionaries": an environment is the surroundings — generally speaking, the program — in which you are using Lightning. It comes with a set of predefined environments, but you can define your own. Of course, you also can build supplementary dictionaries with specialized lists of words.

The first thing I do with any spelling checker, by the way, is run it over a file I keep on a diskette for breaking in the new ones. That file contains words I use that are present in no spelling checker: "Joseph," "Janet," "Matthew," and "Katz" are some of those words. The addition of my personal word list reduces the number of "false alarms" and lets me get on with things. I probably should say, too, that I wrote a program in C called Words that goes through a document and extracts all its unique words so I can build specialized dictionaries in any field.

My only complaint about Lightning so far is the omission of any way to do a spelling check of an entire document at one time. Its only options are to check individual words, or screens of text:

what you see on the screen while you type. Chris Dunford has a test version of a program called "LSPELL" circulating, and it might take up the slack — but so far it can handle only straight ASCII files. It can't handle codes a program like WordStar introduces into the file, so at present it's of limited use.

Oh yes, the version I have of Lightning is 1.00A, which I suppose is the first release. I've encountered no bugs yet.

Reset Switches For The H/Z-150 And H/Z-158

I can't imagine why Heath/Zenith didn't put one on its compatibles. The parts for a reset switch — the switch itself, something on which to mount it, and a few inches of wire — cost well under a dollar and not much labor is involved Maybe it's because IBM didn't put one on its machines. Maybe compatibility can be carried too far?

At any rate, when your computer is locked up so that pressing CTRL-ALT-DEL doesn't do a warm boot, you need a reset switch. Heed the voice of experience: a microcomputer without a reset switch is sure to cause days without sunshine, especially if you program.

It's easy to add a reset switch on either the 150 or the 158. Unless you have experience tinkering with the innards of your computer, though, have a technician do the work.

You want a small, single-pole, momentary-contact, pushbutton switch, like the kind Radio Shack sells in packs of five for $2.49 (catalog number 275-1547). Mount one in a spare blank on the back of the computer.

The switch should be connected so it can short out a capacitor on the motherboard — but not just any capacitor, only one of the following capacitors on the CPU board. On the H/Z-150, it's C207; on the H/Z-158 it's C210. Ben Klausner's technique is to put a 1K resistor between one terminal of the switch and a ground; the other switch terminal is wired to the positive side of the capacitor. I've omitted the resistor and wired one switch terminal to the capacitor's positive, the other terminal to its negative, and there have been no problems on either our Z-150 or our H-158. Ben's way is probably better. In either case, when the computer locks up (and notice I've said "when," not "if"), I press the switch and the computer resets.

If you'd prefer swapping money for the time you'd spend shopping the few parts, First Capitol Computer sells Reset-Z, a kit with the parts you need and instructions on putting them in the H/Z-150, for under $30. Substitute C210 for C207 in those instructions and the kit can be used on the 158.

Don't credit me with the work on these contributions to better living. Ben Klausner came up with the modification to the 150, Jerome Jankura did it for the 158. Each shared his knowledge with the rest of us on Compuserve's HUG SIG.

Any HUG member who subscribes to Compuserve can join the HUG SIG: the command is "GO PCS 48." Request membership and you're in.

One more kind of thing needs saying, by way of orientation and ground rules. I am a user, neither a guru nor a wizard. I make mistakes. But my heart is pure: I try to do right as I see right, showing neither fear nor favor. You ought to have figured out by this point that I speak for no one but myself. Certainly I do not speak for Heath/Zenith nor for any vendor.

Then, why are you bound to read a great deal about some vendors and little or nothing about others? The reason is simple: I write about what I use. I use a great many products, and I constantly try to peer over the horizon, but you really can't expect me to buy every product on the market to winnow out the best things for you and warn you away from the worst. I simply don't have that kind of money. So I must depend on vendors keeping in touch with me and taking the initiative.

Nor am I prejudiced in favor of big business. In fact, if a little firm has something really good, I tend to lean its way. I want that kind of company around a long time so it can continue to contribute good products. I'm a user and I'm not dependent on the microcomputer industry for my living — or for much else except good products I can use. If you're a vendor and want your products considered here, take the initiative and send them directly to me. But please don't ask me to send you a prepublication copy of my comments or to clear with you what I am going to say. I don't do advertising or house reviews here.

That kind of thing probably ought to be said once, for the record, but shouldn't have to be said twice. 'Nuff said.

Sources

RAMPAL $39.95
Reset-Z $27.95
First Capitol Computer
1106 First Capitol Drive
St. Charles, MO 63301
(800) 862-8948 for orders

SideKick
copy-protected $54.95
unprotected $84.95
SuperKey $69.95
Turbo Lightning $99.00
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066

Edix $199.00
Emerging Technology Consultants
4760 Walnut
Boulder, CO 80301
(303) 447-9495

Microsoft Windows $99.00
Microsoft Corporation
10700 Northup Way
Box 92700
Bellevue, WA 98009
(800) 426-9400

Newkey shareware
Frank A. Bell
20950 Smallwood
Birmingham, MI 48010

Word Finder $79.95
Writing Consultants
300 Main Street
East Rochester, NY 14445
(716) 377-0130

Keyboard Template For ZPC Users

Benjamin H. Weiner
3012 Athens Circle
Bowie, MD 20716

If you are doing a lot of Pat Swayne's ZPC, maybe you could use the template that I made using PeachText 5000 spreadsheet and word processor. Figure 1 shows the right side of the template summarizing the changes in the Z-100 keyboard layout as shown on page 6 of the ZPC documentation. Figure 2 shows the left side of the template summarizing the "Special Key Combinations" referred to on page 8 of the ZPC documentation.

```
                  B                                   C
----------------------------------- /////////// -------------------------------------
                                    ///////////     LEAVE THIS BORDER IN PLACE
                                    ///////////
                                    ///////////
                                     Use This   -------------------------------------
                                     Center
                                     Strip to
                                    Adjust the
      ------CUT OUT THIS SIDE------- Template     ----CUT OUT THIS BOX TO FIT OVER----
                                     left-right          THE NUMERIC KEYPAD
                                     When You
                                     Cut Out
                                     the Side
                                     Pieces
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
                                    ///////////
---------------------------------------------------------------------------------------
  ---IBM PC KEYBOARD---------  ---Z100 KEY(S)----------  ---IBM PC KEYBOARD---------  ---Z100 KEY(S)----------
  Alt.........................HELP                      PG DN.......................I Chr or Keypad #3
  Num Lock....................F11                       Ins.........................Ins Line or Keypad #0
  Scroll Lock.................F12                       Del.........................Del Line(Shift-Ins Line)
  Home........................Home or Keypad #7                                     or Keypad "period" (.)
  End.........................Shift-Home or Keypad #1   Arrow Keys..................Arrows or Keypad 2,4,6,8
  Pg Up.......................D Chr(Shift-I Chr) or    Keypad Plus (+).............Shift-Keypad Minus (-)
                              Keypad #9
---------------------------------------------------------------------------------------
```

Figure 1
Right Side Of ZPC Template

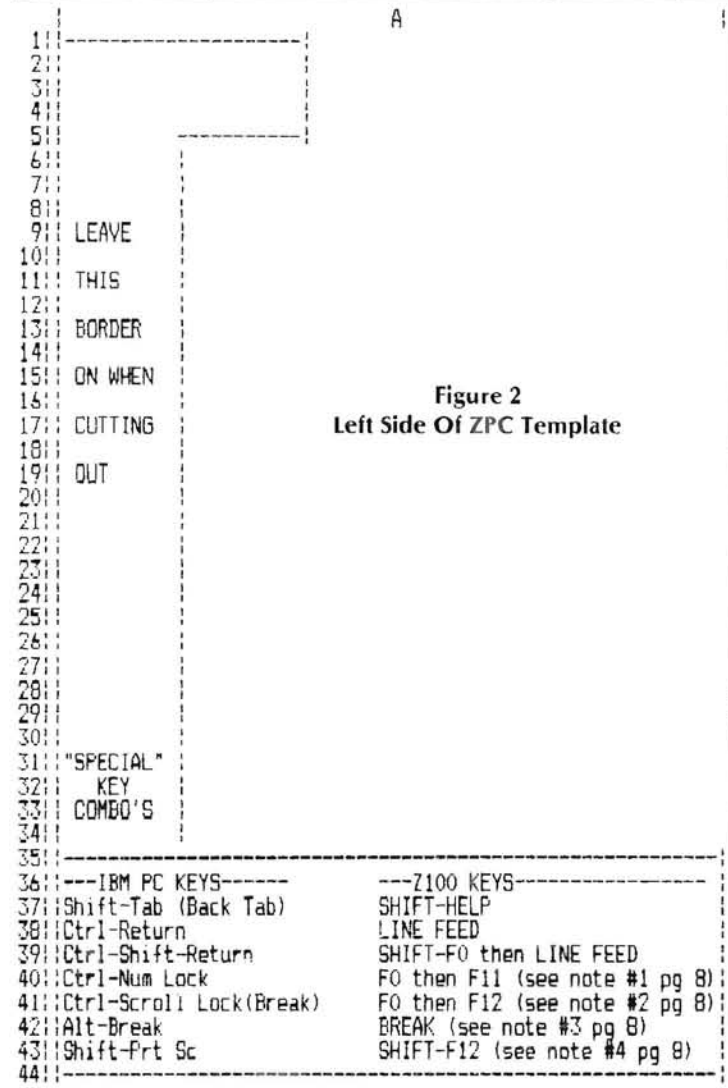


Figure 2
Left Side Of ZPC Template

---IBM PC KEYS------	---Z100 KEYS----------
Shift-Tab (Back Tab)	SHIFT-HELP
Ctrl-Return	LINE FEED
Ctrl-Shift-Return	SHIFT-F0 then LINE FEED
Ctrl-Num Lock	F0 then F11 (see note #1 pg 8)
Ctrl-Scroll Lock(Break)	F0 then F12 (see note #2 pg 8)
Alt-Break	BREAK (see note #3 pg 8)
Shift-Prt Sc	SHIFT-F12 (see note #4 pg 8)

To get full use from the template I mounted it on a "legal size" manilla folder and then cut out the template as shown on the enclosed Figures. I mounted each side of the template on either side of the folder, so I could use the folder's natural "fold" to close up and store the template when not in use. Before mounting the templates I opened the folder, held it against the keyboard and marked the total width I would need by using the left and right side of the "depressed border" that runs around the Z100 keyboard. After the templates are positioned correctly on the folder you have to cut out the center of the folder to fit around the "space bar" and the remaining keys.

Both Figures 1 and 2 can be Xeroxed and used as is to make your template. All you have to supply is the manilla folder. If you want to improve the model, you can use the spreadsheet border information as a guide for setting up your own version of the template or for making your own templates for other software you are using.

If you decide to improve or make your own templates, here are a few things you need to know. Both sides of the template were printed in compressed mode (17 cpi) and at 8 lines to an inch. The right side was also printed with a right margin of 132. Because the right side portion is 127 characters long and I only have an 80 character wide printer, I had to print this sheet to disk (a .PRN file) and then edit it with Peachtext to install the necessary "OUT" commands for compressed mode and 8 lines to an inch.

Printer Ready? A Solution To A Z-100 Hang-up

N. B. Day
UNO P.O. Box 1312
New Orleans, LA 70148

The other day my friend was feeling angry and frustrated. He's been working for six months on the definitive ZBASIC mailing list program, but with the pressures of work and other interests he can only sit down for a long computer session about once a week or so. That morning he complained:

"Do you know, I spent over an hour last night trying to fix a bug that wasn't there!"

"Those sure are the toughest kind. What wasn't it?"

"I'd been playing around with label formats and instead of printing everything on my parallel printer while I was testing, I changed all the LPRINT statements to PRINT (or thought I had). Last night I knew I didn't need the printer for testing, so I never turned it on. Well, one of those darned LPRINTs got by me and the program hung up waiting for the printer. I thought I'd made some change that caused a fatal error and wasted all kinds of time going over the code I'd just written."

"Life sure is unfair."

Fiendishly. "There ought to be a way for a program to check that the printer is there and ready to print without actually trying to print something."

That got me thinking ....

The Z-100 parallel printer port conforms to the so-called "Centronics standard" for parallel printers. Any communication between a computer and a peripheral device (which may be made by different manufacturers) must conform to an agreed-upon standard, so that data which leaves the computer is intelligible to the peripheral. One facet of the intelligibility problem is speed. The computer and peripheral must agree on the speed at which data is to be sent. This problem is particularly acute between computers which are very speedy devices indeed and printers which (being mechanical) are rather slow.

The Centronics standard decrees that the computer will transfer characters eight bits at a time and bring a signal line called NOT STROBE low when the data is ready for transfer. When the printer has received and understood the data, it is to change the state of a line called ACKNLG, indicating that it is ready for the next character. The computer is not to send any data if the line called BUSY is high. Since the printer takes almost forever (in computer time) to print a line of data, the busy signal is often high. There is also a line called NOT ERROR, which is used to signal things like "out of paper" and thus halt the computer. A line called NOT INIT or NOT INPUT PRIME is used by the computer to initialize some printers. All of this signaling back and forth is referred to as handshaking.

It was obvious that the solution to the hangup problem lay in the handshaking between printer and computer. Figure 1 is a partial schematic diagram of the Z-100's parallel printer port. At the extreme right is the actual DB25 female connector on the back where the cable from the printer plugs in. To reduce noise interference to a minimum, each signal line is provided with its own ground return. The signal line and its ground travel through the printer cable as a "twisted pair."

To the left of the connector are shown gates from three different integrated circuits which act as buffers between the outside world and the internals of the computer. We hope that these gates will take care of any noise or zap-signals. Finally, on the left, is the parallel interface chip itself, a 68B21 peripheral interface adapter (PIA). This very complex chip serves as a two-port interface between the computer's data lines and the outside and also as a kind of clearing house for interrupt signals. The two eight-bit parallel ports are referred to as the "A side" and the "B side"; each has lines numbered from 0 to 7. CA1, CA2, CB1, and CB2 can act either as output lines or inputs for interrupt signals. The 8088 microprocessor (and therefore, any program) communicates with the PIA at hexidecimal port addresses E0 through E3.

Looking at the diagram we can see that the various printer lines are all connected to the PIA. NOT STROBE originates at PA2, while the eight data bits are spread around between the A side

and B side. BUSY and NOT ERROR are read by PB0 and PB1, respectively. ACKNLG is connected to CB1 and there is also a connection between BUSY and CB2.

One way to test for the presence of a printer at the printer port would be to send it some data and wait around for an acknowledgment. This is what Z-DOS does. ACKNLG, being connected to CB1, is obviously meant to generate an interrupt which allows processing to continue. The problem arises when the ACKNLG never arrives — now we're hung up. We could send some null data over the data lines, bring NOT STROBE low and then (with some complex manipulation of the PIA's internal registers) look for an interrupt flag for a while before deciding that there really is no printer out there. Fortunately, there is a much easier way.

Note that BUSY is connected from the DB25 connector to the input of a buffer gate. Notice also that the BUSY signal is "pulled high" (as are all of the other handshaking lines) by a 4700 Ohm resistor connected to +5 volts. The output of the buffer gate is connected to the PIA at PB0. When the printer is not busy, that is, when it is ready to accept character data, BUSY will be low, and hence PB0 will also be low. When the printer is busy the reverse will be the case. But what if the printer is off or not there at all? In this case, the 4700 Ohm pull-up resistor will make it appear to the buffer gate and PB0 that the BUSY signal is true (high).

It seems, then, if we look at PB0 and find it to be high, we can be pretty sure that the printer is not there; that it is either disconnected or turned off. There is some possible ambiguity. Perhaps we have just sent the printer a line of data and it is busy printing it. But we are interested in reminding a user of our program to turn on the printer before we send anything to it. If the printer cannot be made ready in a certain length of time, we then want to abort the program to avoid the lock-up that infuriated my friend.

Listing 1 is an example of how this might be done. It is a subroutine to be called prior to the first use of the printer by a program. If all is well with the printer, the user will be unaware that a printer check has been made. If the printer is not available, then the flashing and beeping start and continue until the printer is made ready or time-out occurs.

This little bit of structured BASIC has several interesting features. Line 6020 is a direct read of a hardware port: in this case, the B side of the parallel printer PIA which is at port address E2 hexidecimal. Line 6030 uses the AND logic operator to isolate bit 0, the BUSY line. Line 6040 then determines the truth or falsity of the proposition NOPRN based on bit 0 and sets all of the bits (-1) in NOPRN% if it is true. The WHILE/WEND loop which presents the warning that the printer is not ready is performed as long as there are any set bits in the counter variable TRYCNT% and NOPRN is true, testing for this condition by ANDing them. Line 6070 uses the XOR operator to switch back and forth between inverse and normal video by toggling the value of the variable VIDEO% between 0 and 1. Line 6130 makes the procedure recursive; that is, it calls itself repeatedly unless the external conditions change or time-out occurs. Because BASIC has a "user stack", it does not matter if the routine runs until time-out and ends with calls nested 15 deep. Everything is reset with the next "run" command.

Advocates of "strong typing" will want to initialize all of the variables before the routine is called, but in ZBASIC this is not strictly required. Because recursion and WHILE/WEND are both profligate with memory, those with smaller systems or who want a longer warning period may want to try the procedure in Listing 2, which is not recursive.

Listing 1

```
10 '*************************************************
20 '* ROUTINE TO BLOCK THE 'PRINTER NOT READY HANG-UP' *
30 '* IN ZBASIC. Easily adaptable to other high-level  *
40 '* ulanguages or to assembly language  Demonstrates *
50 '* direct read of hardware port, use of logical ops.*
60 '* and recursion  Checks 'BUSY' line from Centronics*
70 '* standard parallel printer, which is low if print-*
Department of Physics, University of New Orleans, New  *
80 '* er is there and ready, and high if not--N.B. Day.*
90 '* Dept  of Physics, Univ  of New Orleans, New Or-  *
100 '* leans, LA 70148, 18 February 1985                *
110 '*************************************************
6000 IF TRYCNT%<=0 THEN LET TRYCNT%=15
       'Initialize counter ~ 15 sec
6010 LET TRYCNT%=TRYCNT%-1    'Down count
6020 LET NOPRN%=INP(&HE2)     'Read the printer PIA B side
6030 LET NOPRN%=NOPRN% AND 1
       'All we care about is bit 0 (busy)
6040 IF NOPRN% THEN LET NOPRN%=-1
       'If busy high - 'no printer' true
6050 WHILE NOPRN% AND TRYCNT%
       'Do this loop if any set bits in
6060   SCREEN ,VIDEO%         'trycnt% and noprn% is true
6070   LET VIDEO%=VIDEO% XOR 1
        'Toggle inverse video on warning.
6080   LOCATE 1,35            'Seldom-used place on screen
6090   PRINT "Can't continue until printer is ready  "
6100   BEEP                'Delete if this drives you crazy
6110   TIM!=TIME              'Wait until next second
6120   TIM1!=TIME: IF TIM1!=TIM! THEN 6120
6130   GOSUB 6000       'Recursive call to this procedure
6140 WEND
6150 SCREEN ,0          'Make sure to leave in normal video
6160 IF TRYCNT% THEN RETURN 'Escape route if printer ready
6170 CLS                'Here to abort calling program if
6180 LOCATE 10,1             'LPTR never ready.
6190 PRINT "Printer never became ready
       Program halted to prevent lock-up."
6200 BEEP
6210 END
```

Listing 2

```
10 '*************************************************
20 '* ROUTINE TO BLOCK THE 'PRINTER NOT READY HANG-UP' *
30 '* IN ZBASIC. Non-recursive version for systems with*
40 '* limited memory or for those who want a longer    *
50 '* warning period                                   *
90 '*************************************************
100 '
8000 IF TRYCNT%<=0 THEN LET TRYCNT%=20
8010 GOSUB 8190
8020 WHILE NOPRN% AND TRYCNT%
8030   SCREEN ,VIDEO%
8040   LET VIDEO%=VIDEO% XOR 1
8050   LOCATE 1,35
8060   PRINT "Can't continue until printer is ready  "
8070   BEEP
8080   TIM!=TIME
8090   TIM1!=TIME: IF TIM1!=TIM! THEN 8090
8100   GOSUB 8190
8110 WEND
8120 SCREEN ,0
8130 IF TRYCNT% THEN RETURN
8140 CLS
8150 LOCATE 10,1
8160 PRINT "Printer never became ready
       Program halted to prevent lock-up."
8170 BEEP
8180 END
8190 REM
8200 LET TRYCNT%=TRYCNT%-1
8210 LET NOPRN%=INP(&HE2)
8220 LET NOPRN%=NOPRN% AND 1
8230 IF NOPRN% THEN LET NOPRN%=-1
8240 RETURN
```

Use The Cassette Interface Of Your Z-89

Ronald La Claustra
882 57th Street
Brooklyn, NY 11220

As a storage medium for home computers, the audio cassette is viewed either as a relic of the past or the desperate choice of "toy computer" users. For speed, of course, disk systems have it over tape, but a properly designed and used tape system can play an important part in microcomputer storage. I'm thinking specifically of long term storage; an archival system.

Disks, especially high density types, are not as permanent as you might think. Aside from physical delicacy they suffer from environmental instability, expanding or contracting with changes in temperature and humidity. The recorded signals move with the disk eventually becoming eccentric to the drive heads and thus unreadable.

The magnetic signal stored on the disk is also fugitive, weakening below retrievability with time.

These factors tell us that any long term (years) storage on disk will call for special vigilance. You could re-initialize and recopy your back-up disks every year which would keep the data fresh on them, or you could transfer them to another media. This is where tape comes in.

Cassettes are more rugged than disks. They also are much less affected by environmental changes which are further nullified by the recording system. The magnetic information is stored at a higher level and over a wider surface area meaning little fadeout over the years.

In addition to dumping your old files on tape, you could use it as a data recording medium in somewhat hostile environments or for scientific data recording on a multi-track recorder.

You may ask "If it's so great, how come nobody uses it?" Ah! there's the rub! Due to chance and unfortunate decisions, the Z-89 cassette recording capability has been undermined, but with a little work we can revive it. It's my purpose here to tell you just how.

As usual, we have hardware and software problems to overcome. I'll begin with the hardware which is straightforward, then to the software which has several possibilities. The task requires some electronic construction skill, but isn't very complex.

The Z-89 Interface Method

To understand the hardware we need to examine the Z-89's expansion connectors. It's a bit technical, but when we get to doing things, it's not that bad.

Inside the Z-89, standing vertically and facing front, is the CPU panel the "brain" of the machine. On either side are rows of three pin connector pairs. The group on the right are the interface expansion set. Here are connected the various disk interfaces, serial interfaces, and other purpose devices which are viewed as "Port" devices by the Z-80 microprocessor IC (more on "Port" in a moment).

The group on the left are for "Memory" devices. Originally envisioned for carrying ROM (Read Only Memory) type ICs containing a built-in BASIC, Zenith has only used these connectors for its 64K memory upgrade.

The smaller 10 pin connector at the top of all pairs carries the 8 Data signals. The lower 25 pin connectors carry different signals on each side, hence the lack of interchangeability. On the left "memory" side the lower connectors all carry the Z-80 processor's addressing bits from A0 to A12, bank select signals and various power lines. The right side has address bits A0 to A2, Read/Write select signals, Port select signals, system clock, Reset, various Interrupt and power lines.

I'll talk now of the difference between a "Port" and "Memory" location without making this a course in computer architecture. The microprocessor has to have a method of finding and storing the information it is to operate on. The Z-80 IC, as all microprocessor (MP) ICs, has a "memory" select method of Read or Write signals with a combination of the 16 addressing signals. The memory devices are arranged in a manner which allows the MP to access a particular cell out of the possible 65,000 odd combinations of those 16 address lines.

Port addressing is a capability of 8080 and Z-80 MPs which allow the separate access to other DEVICES rather than memory cells. A serial interface, say, can be wired into the Port address side rather than clutter up the main memory wasting space.

The Port addressing method is similar to the Memory except that a special notifying signal (IORQ) and only the lower 8 address signals (A0-A7) are used. These 8 will allow a total of 256 input and/or output "channels" or "ports" to be accessed.

Any circuitry to be accessed through one of these I/O Ports must be able to discern when it is being called upon to receive or return data. The 8 address lines must be monitored for the unique combination (address number) which, in conjunction with an I/O Port Request signal (IORQ), will allow the circuit to act.

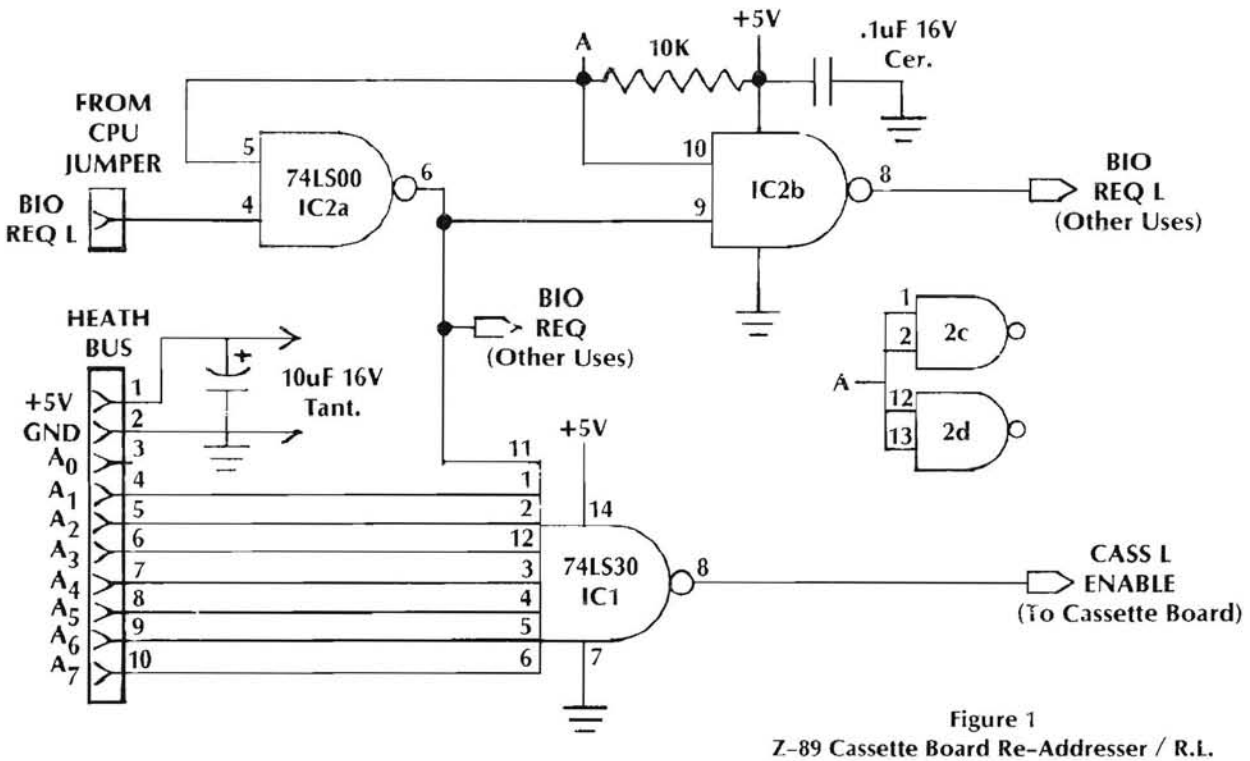


Figure 1
Z-89 Cassette Board Re-Addresser / R.L.

The Zenith implementation of this scheme is to split the address detection between the CPU panel and the interface circuitry connected to the expansion pins. The top 5 of the 8 (A7-A3) are detected by ROMs on the CPU and are reduced to a single activating signal. This signal is now called by the name of the device it is to activate and is brought out to a specific pin on the lower expansion connector (right side). This means that those ROMs determine what particular address will control which connector pin and which device will be activated.

Here we get to the point!, those ROMs have been changed since the original H-88! Faced with an increased demand for disk drive capacity and a lack of space in the computer, Zenith dumped the cassette from the picture and gave its activating pin (12th from the top on the 25 pin) to the alternate disk interface. The computer will no longer work with the cassette interface plugged in and so it was relegated to the status of high tech paperweight. (For completeness: The right-most connector was set specifically for the hard-sectored disk controller and its pins 11 and 12 differ from the other two rows. Also, the cassette had the top 7 (A7-A2) address bits combined on the CPU panel.)

Hardware

This is why we need to make alterations to allow the cassette board back into the *Z*-89. We must readdress the cassette board to an unoccupied address of the Port space. This requires two steps, the second will vary depending on your accessories. The first part of the hardware change involves the cassette board.

With the parts side up, count down the lower large connector to the 12th pin. Make two cuts with a razor knife about 1/4th inch apart into the trace of foil leading away from this pin. With a hot soldering iron remove the detached segment (Photo 1).

Follow the cut segment away from the connector and after a downward 90 degree turn, it ends in a feedthrough pad. This goes to the back of the board and is a convenient place to put our new connector. Clean out the hole of solder and insert (parts side) a male .1 inch straight pin connector. They're sold in rows of 36, so just cut one off and solder in place.

Now add a 10K ohm resistor from this point on the solder side to the +5 Volt supply at pins 13 and 14 of the nearby IC U706 (Photo 2). This will keep the board deactivated without any connection to the new pin, otherwise the computer won't boot up with the cassette board in place. That's all for the cassette board.

In the second part of the hardware change, we need to develop the enabling signal for the cassette board. This signal is the one derived from the combination of the address bits and the I/O request bit. The interface wants two adjacent ports one at the even address, the other at the odd as in 62, 63.

The simplest way out is to make use of an unused serial port enable line. This assumes you have the three port serial board and that you have only two of the ports filled with ICs. Even if one IC set is missing, you still must be certain that no other accessories use the vacant addresses.

If all is clear, then all you need to do is attach a jumper from the unused port's addressing pin and have it attach to the newly installed connector pin on the cassette board.

The pin to pick from will be 11, 9, or 10 of the 25 pin group. This order corresponds, top to bottom, with the position of the ICs on the board. Usually, the center group is vacant and so the jumper should pick the signal from pin 9.

Just tack solder a wire to the connector pad on the serial board and at the other end of the wire attach a female .1 inch connector (see below). Plug this connector over the new pin on the cassette board.

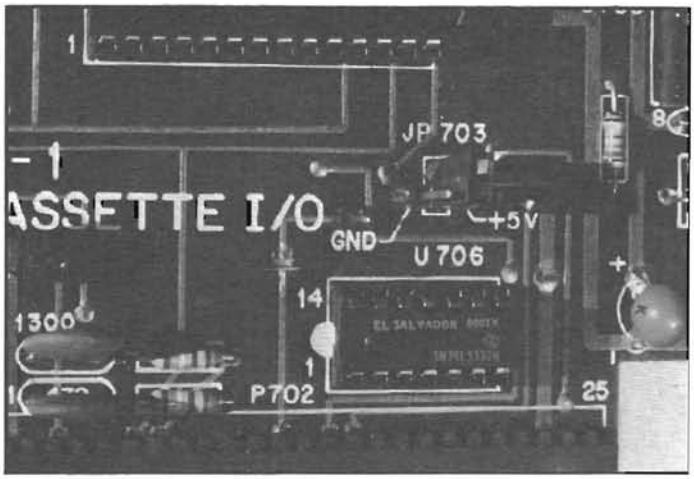


Photo 1
Showing location of the cut trace and the installed .1 inch male connector just over the printed "GND".

Doing the addressing this way puts the cassette board in the block of addresses from 208 to 215 decimal. When you program for this connection use the addresses of 208 and 209.

Some non-Heath accessory boards may develop similar "extra" port activation signals which you may be able to use if your serial ports are all filled. It needs to be true on low, check the accessories' manual.

If this method is impossible for you, there is another albeit more involved way out. You'll have to add a jumper to the CPU panel and build a small circuit to isolate other port addresses to control the tape board.

The jumper to the CPU should be done with care. It will have no consequence in the usual operation of the computer and so can be done without reservation. The object is to bring the I/O Request signal (BIO REQ L on the schematic) up to where we can use it. Unfortunately, Zenith didn't put it out on one of the interface pins so we've got to pick it off. It's a useful signal and can be used in other expansion projects.

Remove the CPU panel carefully noting what wire goes where. If you built the kit this should be old stuff. If not, WRITE NOTES ON PAPER. It'd be real dumb to put on a nice jumper then forget how to put the computer back together!

Prepare a 9 inch length of thin, flexible shielded cable. Connect the inner lead to pin 9 of IC U509 and the shield to pin 10. See your manuals for the circuit board maps. U509 is down in the lower right corner (solder side) near the big Z-80. On the other end of the cable cut back the outer insulation and shield leaving out about 1 inch of the insulated center conductor. Strip about 1/2 inch from the end and wrap and solder it around the end of a female .1 inch connector. These connectors are of the Printed Circuit type with rather short tails, but they will work. Again, they come in 36 pin strips so cut one off. Tape up the connector end or use heat shrink tubing. Reassemble the computer bringing the new cable to the front by running it under the CPU panel. If you get the two beeps at turn-on, all is well.

You'll now need to construct the circuit diagrammed in Figure 1. It uses an 8 input NAND gate to look for the coincident occurrence of the address bits A1-A7 and the I/O request signal. The BIO REQ L line (from the CPU jumper) must first be inverted via a NAND gate to true on high. The resulting signal from this combination will enable the cassette board at addresses 254 and 255. Although this is in the Zenith reserved port address area, it's unlikely that they will ever use these locations. That's fortunate because any other address would require more complicated circuitry.

A free gate can invert BIO REQ again back to true on low for other circuitry if you wish.

I built mine on perf board, cementing on the female .1 inch connectors to one edge with silicone rubber adhesive. The Zenith boards use right angle female connectors, but I've found no source for them other than Zenith, so just use the straight PC type and use wire jumper leads to connect it to the circuitry. The circuit being so simple, I soldered directly to the IC pins. The male .1" connector to receive the CPU jumper can be cemented near the lower edge of the board for access. The new output signal is a 14" wire, shielded if you like, ending again in a female .1" connector. This will plug over the pin we added to the cassette board.

Figure 2 is the artwork for making an etched circuit board for the project. The parts and connections are as described in the schematic. The artwork is a 1:1 positive representation of the copper as viewed from the copper side..The row of pads for the bus connections should be on the RIGHT.

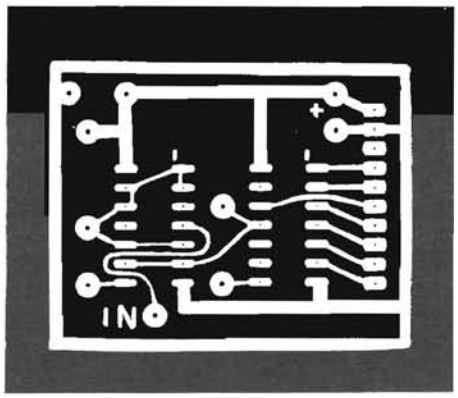


Figure 2
Reverse to positive and print 1:1 as shown.

Install the new board on the top 10 pins of any of the left hand 25 pin "Memory" connectors. Connect the I/O Request jumper. Install the cassette board on either of the two left connectors on the right side of the CPU panel and attach the new enabling signal cable. Be sure to have the jumpers of the cassette interface set correctly as directed in the manual. The recorder cables from the board connect as usual.
If you have a jumper in the way on the connector pins, you can solder the wire to the new circuit board's pads.

When you apply the power and hear the two beeps everything is fine and you can go on. If not, recheck your wiring. See if you get the beeps without the cassette board installed. If not, recheck the new interface circuit.

The Software

Now that the cassette board is back in place, we have to get some software to run it and that's where it gets a bit sticky.

Zenith has no software that will run with the reinstalled interface. The original tape programming was contained in the old non-CP/M H-88 ROMs which are in few Heaths and no Zeniths. What's worse is that most third party disk-tape software relies on these routines.

What is needed is ROM-free coding and the ability to set the port address for the tape interface board. The latter ideally would be done without having to mess with assembly code. You'll have to change whatever address the software was written with to the one you've reset your tape board to.

CP/M fans seem to have an alternative in a package from Kandueasy Computer Software Services, 6218 Blossom Lane, Alexandria, VA 22310. TAPEIO seems to fill the bill, but I have no experience with it. If interested, be certain to check about the Port address setting point. Since it's for CP/M it can't be ROM dependent.

You can also write your own software The 8251 serial IC used on the tape board is not hard to program. The IC contains two registers which are accessed at the two port addresses we've gone to such lengths to set aside. Sending various numbers out to these registers controls the IC.

First is the Control register. This register is at the odd address and is written to set the chip up for read or write, turn on/off the tape, and reset the chip.

"Hidden" behind this register is the Mode Control register. It is involved with setting the type of serial output wanted from the chip. This register can only be written to either after a hardware reset or a chip reset (64) being sent to the usual Control register (CR). Other circuitry on the tape board forces certain choices here. A 77 sent to this register will set the IC for the usual (Heath) tape parameters.

A 52 sent to the CR will start the player, a 17 will start the recorder (if you have the "one machine" jumper set on the tape board, either will start the deck). A 16 will stop the tape.

Reading the CR will return a "status" byte which can be checked for various errors and ready bits

The even address accesses the data register through which you get or send the data to the deck. In both cases, you must first check the CR to see if the IC is ready before moving the data.

More detailed information can be gotten from the IC data sheet which may be available from the chip's manufacturer(s). Unfortunately, Zenith did not include a copy in the computer manuals.

The difficulty, for an archiving use, is speed. In order to store the maximum amount of data per length of tape, the controlling program has to be fast.

Listing 1 is a Heath Users' Group, Tiny Pascal (P/N 885-1086-[37]) program to dump to and load a file from tape. It is a SAMPLE program; it works but not prefectly or fully. It does show the major points in writing tape software. The program might make occasional reading errors (try it again) and one would like to add error checking and file names.

The program follows Heath tape standards as I know them (incompletely). It will read (not write) a tape written to Heath standards if it's a text file. It won't convert an H-88 tape program to RUN on a disk system. This is also a warning about other software packages, they may not convert tape programs either, but few people will need this ability.

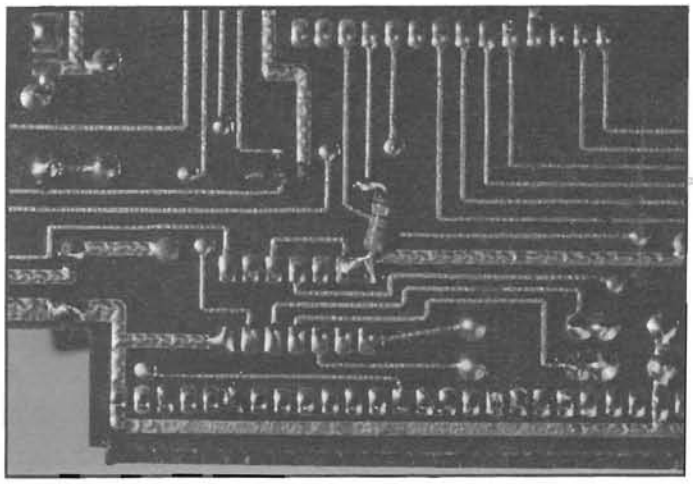


Photo 2
Showing placement of the 10k pullup resistor from the new connector's bottom to the +5 volt supply.

Being in Pascal, the program's structure should be discernible. It reads the file from disk or tape into a buffer in 256 byte blocks. This is done so the tape machine can be turned off when not in use, so tape isn't wasted and no data is missed.

When writing to tape, the buffer is filled with the recorder off, so disk accessing doesn't cause gaps on the tape. The tape is written with 32 syncronizing bytes, a "start of data" flag byte, 4 bytes of semi-implemented header items, then the data is dumped from the buffer. A non-Heath touch is added with a closing 4 bytes of "end of data" bytes. Before each byte is written, the 8251 control register is checked to see if its ready. A delay is included in the OUTOK routine which slows the writing to make reading easier for Tiny.

When the tape is to be read, the 8251 control register is checked for a new byte and if so found, the byte is checked for that "start of data" byte. If it is this byte, then the next 260 bytes will be stored in the buffer. When full, the tape is stopped and the data part of the buffer is dumped into the appointed disk file. You can see that if the start byte is missed, then you'll loose 256 bytes of data, and get no warning.

Also note most of the header is ignored. Only the byte after "start" (TYPEFILE) is checked. It was set at 32, except during the last block to be written when 128 was added to it. This is the Heath way of denoting "end of the tape file", so when it's found to be greater than 127, it's time to close up.

The tape is recorded at a rate of 1200 bits per second which pushes Tiny to its limits. Machine language (via Assembler) would be best. BASIC can be used with a large delay when writing and the penalty of wasted tape. Disregard of format can help too; its really only needed when storing files.

The Proper Tape Method

Getting a good recording is the name of the game for tape storage. The recorder should be in good shape: heads in align-

ment and tape speed accurate and constant. It needn't be anything grand. If music sounds clear, quiet and steady, it will probably work OK. The recorder should have a high level "line" or "aux" input, as well as a jack for remote pause control.

The heads should be cleaned and demagnitized. What's good procedure for music recording is mandatory for data recording. The Zenith circuitry is well designed and will get your data off the tape IF it's there to be gotten!

You can use the recorder from your Hi-Fi system if you don't have a decent portable recorder. Just leave the noise reduction (Dolby, etc.) OFF and set the level to just under 0dB. Use a Y patch cord to record on both channels of a stereo machine, but take the playback from only the best (strongest) one.

You can use any good quality audio tape. Freedom from dropouts is more important than frequency response in this application, so you don't need premium, but avoid the "three-for-a-dollar" variety. Cassettes should not be longer than C-90s with C-60 as the limit for really important files, although that length won't hold a full 100K disk.

I've said cassette a lot in this article, but you can use an open reel machine with better results both in quality and storage amount, although having the computer control the pause of such a deck may be a problem.

If the deck has no provision for remote pause and uses an electric switch to activate its pause function, ("logic controlled" or "soft pushbutton") you may be able to add a jack by soldering one, via wires, across the switch's terminals.

The faster the tape speed, the more secure the data. It's also probably best to save your programs as ASCII text files rather than compressed or binary, just so you can see any errors that creep in.

I hope you haven't been daunted by all my directions, it's really not that tough to use the cassette board. Even if you have to buy a new one from Zenith, it may prove one of your longer lasting investments with the dividends of data in a secure bank!

Figure 1

```
{PROGRAM CASSETTE 1/29/83  1/29/85  2/28/85}
CONST
 DR=254, CR=255, {MY EXTERNAL PORTS-- DR=62, CR=63;}
                 {ORIG HEATH-- DR=248, CR=249;}
VAR
 A,B,BUF,SYNC,TYPEFILE:INTEGER,
 FILEBYTE:ARRAY[260] OF INTEGER,

PROCEDURE INOK;                    {DATA OF TO READ?}
BEGIN
REPEAT
UNTIL (PORT[CR] AND 2)=2,
END, {INOK}

PROCEDURE OUTOK;                   {OK TO WRITE?}
BEGIN
REPEAT
FOR B:=0 TO 10 DO;                  {DELAY}
UNTIL (PORT[CR] AND 1)=1;
END; {OUTOK}

PROCEDURE CREAD,  {READ TAPE, STORE IN DISC FILE}
BEGIN
WRITE('ENTER FILE NAME TO WRITE TO ');
REWRITE('?'),
REPEAT                             {TILL EOF}
 PORT[CR]:=52,                     {START TAPE PLAYER}
 REPEAT                            {WAIT FOR START BYTE}
  INOK;
 UNTIL PORT[DR]=2,
 FOR BUF:=0 TO 259 DO              {READ DATA BLOCK, STUFF INTO ARRAY,
  BEGIN INOK; FILEBYTE[BUF]:=PORT[DR]; END;
 PORT[CR] :=16;                    {STOP TAPE}
 FOR BUF:=4 TO 259 DO
  BEGIN PUT(FILEBYTE[BUF]); WRITE(FILEBYTE[BUF]); END,
                  {TRANSFER DATA TO FILE IGNORING HEADER  WRITE ALSO TO CRT}
 UNTIL FILEBYTE[0]>127;
 REWRITE('*'),
 END, {CREAD}

PROCEDURE CWRITE,                  {WRITE TO TAPE FROM DISC FILE}
BEGIN
TYPEFILE:=32;                      {NON-HEATH TYPE NUMBER}
WRITE('ENTER DISC FILE NAME TO READ FROM '),
RESET('?')
IF IORESULT<>0 THEN BEGIN WRITE(10,'FILE NOT FOUND',10); EXIT; END;
REPEAT
 FOR BUF:=0 TO 255 DO              {256 BYTE BUFFER}
  BEGIN
   GET(FILEBYTE[BUF]);
   IF EOF THEN FILEBYTE[BUF]:=0;   {IF EOF FILL BUFFER WITH NULLS}
   WRITE(FILEBYTE[BUF]),
  END,
 IF EOF THEN TYPEFILE:=TYPEFILE+128,     {IF DISC EOF THEN SET TAPE EOF}
 PORT[CR]:=17;                           {START TAPE RECORDER}
 FOR SYNC:=0 TO 32 DO
  BEGIN OUTOK; PORT[DR]:=22, END;        {WRITE SYNC BYTES}
  OUTOK; PORT[DR]:=2;            {STX}   {WRITE HEATH STYLE HEADER}
  OUTOK, PORT[DR]:=TYPEFILE;     {TYPE}
  OUTOK; PORT[DR]:=0;            {SEQ}
  OUTOK, PORT[DR]:=0,            {HI CNT}
  OUTOK; PORT[DR]:=0;            {LO CNT}
  FOR BUF:=0 TO 255 DO
   BEGIN OUTOK, PORT[DR]:=FILEBYTE[BUF], END,    {WRITE BUFFER TO TAPE}
  FOR SYNC:=0 TO 4 DO
   BEGIN OUTOK, PORT[DR]:=3; END,        {ETX}   {SEND OUT 'END' BYTE  NON-HEATH}
  PORT[CR]:=16;                          {STOP TAPE}
 UNTIL EOF;
 RESET('*'),
 END; {CWRITE}

BEGIN {CASSETTE}                   {FIND OK COMMAND}
REPEAT
 A:=0,
 PORT[CR]:=64; PORT[CR]:=77;       {RESET IC; INITIALIZE}
 WRITE(10,'DO YOU WANT TO  1)READ  2)WRITE  3)END ?'),
 READ(A#);
 IF A=1 THEN CREAD,
 IF A=2 THEN CWRITE;
UNTIL A=3,
END  {CASSETTE}
```

*

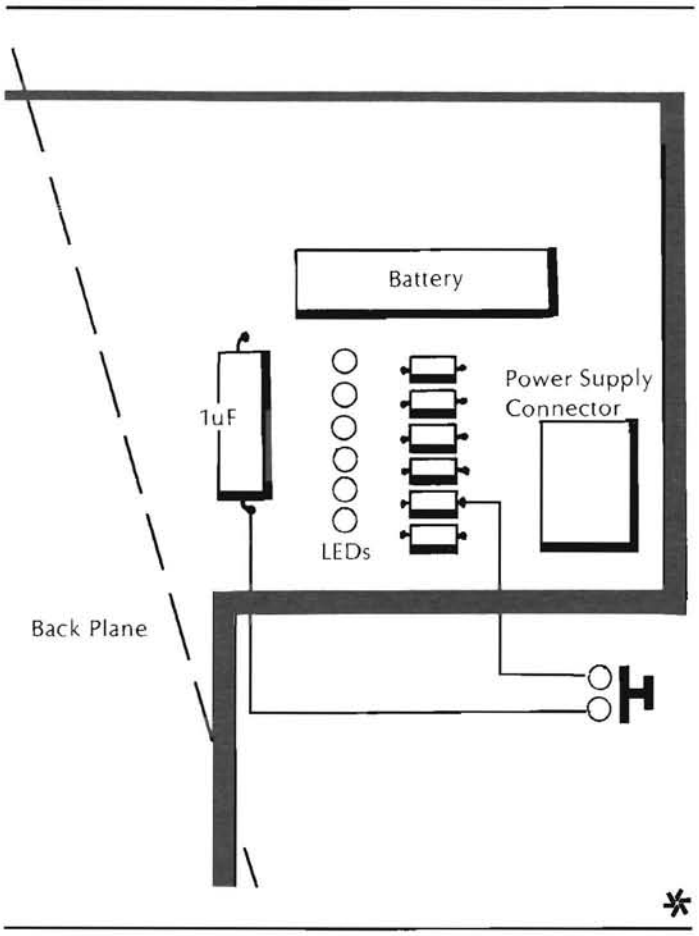
Don't Take No For An Answer!
Or
How To Install A Reset Switch On Your H/Z-241

Jim Buszkiewicz
HUG Software Developer

Have you ever written an assembly language program that plays with interrupts only to go off into the weeds during the development stages? How about that super graphics game that uses ALL of your system memory and can't exit back to DOS because the author of the program didn't know how? Whatever your reason may be, ALL, and I mean ALL, computers need a real hardware reset switch. The H/Z-241 is no exception, and I'll never understand why its designers never put one in.

Well, never fear, installing one is easy. The first thing you need to do is decide where you want to install the switch. I chose to install it on the back panel of the Z-329 card. Since this card is installed in the left most slot on my backplane, reaching the switch is convenient. I provided a means of quickly disconnecting the switch, should the Z-329 card need removal. The easiest place to mount the switch would probably be either on the chassis back panel or on a blank circuit card panel. This is probably the toughest decision you'll have to make all year!

The switch itself should be a momentary contact normally open variety. Heath part number 64-685 is the proper type, and can be ordered from the Heath Parts Department at (616) 982-3571. This switch is small and mounts in a 1/4" hole. The switch contacts should be wired between the DC-OK line (buss connector Y19) and ground (buss connector B1). These two points can be found directly on the backplane near the power supply connector. Next to the six green LEDs is a large 1uF capacitor. The end of this capacitor, furthest away from the battery, is ground. One of the switch leads can be tack soldered there. The other switch lead can be tack soldered to the lead on the right side of R106 (this is the side furthest away from the LEDs). R106 is the fifth resistor away from the battery, and it is connected to LED D107. This LED is the DC-OK indicator.



Modify HDOS LPH24.DVD

Scott Swain
P.O. Box 2282
Goleta, CA 93118

This article describes how to modify the standard HDOS printer device driver, LPH24.DVD (or similar), to insert a left margin on printouts. This is convenient when you want to punch holes in the printout for inclusion in a loose-leaf binder. Normally, printer output is printed flush left on the page. Parts of the printout are inevitably lost when holes are punched in the paper. Here's a way to eliminate this problem.

These modifications cause the device driver to issue one (or more) tab(s) at the start of each line, shifting the text to the right. You also have the option of inserting one or more blank lines at the top of each new page.

A potential problem with the insertion of a left margin is that if the length of the original line, plus the inserted tab(s) exceeds the printer line width, part of the line will be split into a new line (also with a margin). This isn't a problem with program listings (e.g., a listing in BASIC), where line lengths can exceed 80 characters. However, it is distracting (and wasteful of paper) when printing program documentation (such as a HUG README.DOC file). Every other line will contain but a few characters. The solution is to use the condensed printing mode (132 chars/line) of your printer, if available. When in the condensed mode, an 80 character line, plus the left margin will fit comfortably on one line.

The following modifications apply to the HDOS 2.0 device driver assembly language file LPH24.DVD. If you are using another device driver, these modifications may or may not work, depending on the similarity of the drivers, and the availability of the assembly language listing for your driver. I'm assuming you know how to assemble the modified code using the HDOS assembler, ASM. A brief rundown of the process appears in the article "Doctor Your Daisywheel Device Driver," by Charles E. Cohn on pages 37–40 of the May–June issue of Sextant. You can find a detailed explanation of device drivers in the article "The HDOS Device Driver Programmer's Guide" by Al Dallas, Dale Lamm, and Tom Jorgenson in REMark #20 (September 1981), starting on page 7.

Using a text editor, make the following modifications to a copy of LPH24.DVD. First, locate the following code and add lines 5a and 5b after line 5 (line numbers on the left are for ease of reference only, and do not appear in the actual assembly code listing):

```
        LPOPENW EQU   *


(1)     *       INITIALIZE LP:
(2)
(3)             CALL  INITLP
(4)             MVI   A,CR
(5)             CALL  LPOUTCH
(5a)            MVI   A,TAB     OUTPUT A TAB FOR MARGIN
(5b)            CALL  LPOUTCH
(6)             RET
```

If you desire more than one tab (each tab adds an 8 space left margin), repeat lines 5a and 5b as many times as desired, (i.e., 5a,5b,5a,5b, etc.).

If you want one or more blank lines at the top of each page to form an upper margin, in addition to the left margin, change the "CR" in line 4 to an "NL". For several blank lines at the top, repeat lines 4 and 5 as many times as desired.

Now, locate the following code, and insert lines 16a and 16b:

```
(8)     LPOT6   CPI   NL
(9)             JNZ   LPOT7
(10)            MVI   A,CR
(11)            CALL  LPOUTCH
(12)            MVI   A,LF+200Q   AVOID THE INFINITE
                                  RECURSE
(13)            CALL  LPOUTCH
(14)            LDA   TLP.LX
(15)            INR   A           UPDATE LINE INDEX
(16)            STA   TLP.LX
(16a)           MVI   A,TAB       1 TAB LEFT MARGIN
(16b)           CALL  LPOUTCH
(17)            JMP   LPOT9
```

This will add one tab (8 spaces) to each line. If you want to add more tabs for a wider left margin, duplicate lines 16a and 16b as many times as lines 5a and 5b were previously duplicated.

Use ASM or another assembler to produce the machine language version of the device driver. Unless you want ALL of your printouts to have a left margin, it's a good idea to keep your original LP device driver, and name this one with another two-letter file name. I use PL.DVD, using the mnemonic "Program Lister" to distinguish this version from the original LP.DVD.

INTERNATIONAL HUGCON H86

Heath/Zenith Users' Group • Hilltop Road • St. Joseph, Michigan 49085 • (616) 982-3463

INTERNATIONAL HEATH/ZENITH USERS' GROUP CONFERENCE

Official Conference Registration Form

O'Hare Hyatt Regency
Rosemont, Illinois
August 15, 16, 17, 1986

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

Registration(s) must be post marked no later than July 31, 1986. Cancellation will not be accepted after this date.
Sorry, We cannot accept purchase orders

Continued from Page 57

The following is a list of programs that can have been tested under ZPC Version 2 (with 768k of memory), as of 3-18-86.

Program:	See Notes:
BENCHMARK Word Processor vers. 4.4	1
CORNERSTONE database	1
Compiled PC GW-BASIC Programs	1
Compiled QUICKBASIC Programs	5
DAC EASY ACCOUNTING	1
DBASE III version 1.1	2,3,4
DBASE III + version 1.0	2,3,4
EDIX version 2.05	2,3
EINSTEIN WRITER version 7.2	2,3
ENABLE version 1.1	2,3
FRAMEWORK version 1.1	2,3,4
FRAMEWORK II version 1.0	2,3,4,5
GW-BASIC (Zenith PC versions)	2,3
LOTUS 1-2-3 release 1A	2,3
LOTUS SYMPHONY	1,4
MICROSOFT WORD vers. 1.1 (Zenith PC)	2,3
MICROSOFT WORD version 2.0	1
MULTIMATE version 3.3	2,3
MULTIPLAN version 1.2 (Zenith PC)	2,3
NORTON UTILITIES	1
PC FILE	1
PC PALETTE version 1.0	3
PC WRITE version 2.4 or 2.55	3
PRINT MASTER	1
RUN/C	1
SIDEWAYS version 2.02	1
SUPERCALC3 version 2.0, 2.1	3
TURBO PASCAL	1
VOLKSWRITER DELUXE version 2.0	3
WORD FINDER	1
WORD PERFECT version 4.1	2,3

Notes:

1. Runs without any patches or hardware support.
2. Runs in the monochrome mode without any patches or hardware support.
3. Runs without patches if the ZPC Hardware Support circuitry is installed. Otherwise, you must use the patches supplied with ZPC.
4. Copy protection must be removed before you can run this program.
5. Requires a special patcher, supplied with ZPC.

PC.COM — This program is used to turn on the IBM emulation mode after ZPC is loaded into memory. With PC.COM on your system disk, you just enter

```
A>PC
```

to turn IBM emulation on. PC can also be used to set a specific video mode, much like the MODE program used on IBM PCs.

Z100.COM — This program is used to turn off the IBM emulation mode. With Z100.COM on your system disk, you enter

```
A>Z100
```

to enter normal Z-100 operation. You can then run your Z-100 programs as usual.

SETZPC.COM — This program is used to set several parameters, such as the way ZPC emulates bright colors, etc, and the default video mode (when you run PC). You can also specify the character font to be used when ZPC is in the IBM mode. You can choose from an IBM-style font, the default Z-100 font, or a user supplied font (from a custom ALTCHAR.SYS). All changes made by SETZPC can be either temporary or permanent.

ANSISYS.COM — This program emulates the ANSI.SYS device driver that is used on PC-type computers. It will allow you to run programs that use ANSI codes.

SETANSI.COM — This program allows you to turn ANSI emulation on or off after you have loaded ANSISYS.

PATCHER.COM — This program is used to apply patches to programs that need them to run with ZPC. It is menu driven and much easier to use than DEBUG. The patch information is stored in a data file, PATCHER.DAT, which is an ordinary text file that can be modified or added to by using an editor or word processor. As patches for more programs are developed, they will be printed in REMark in PATCHER data file format.

PATCHER.DAT — The patcher data file. It contains patches for all programs in the list above that have a 3 in the note column.

FIXCB.COM — This program fixes compiled PC GW-BASIC programs so that they will run with the small memory version of ZPC. Patching is not needed if you have 768k of memory.

FIXQB.COM — This program fixes compiled QuickBASIC programs so that they will run with ZPC. It is for stand-alone programs, compiled using the /O option.

FIXPSC.COM — This program fixes Heath/Zenith and Clarkston screen printing utilities (Z-100 versions) so they can be used to dump graphic displays to a printer while ZPC is in the IBM mode.

FIXFWII.COM — This program applies a special patch to Framework II, that cannot be done with PATCHER, to make it work with ZPC.

DEMO.COM — A program that demonstrates ZPC.

***.ASM, *.ACM** — These are the source code files for ZPC and its support programs.

TABLE C Rating: (2), (7), (10)

HUG P/N 885-6009-37 MS-DOS
Screen Saver Plus $20.00

Introduction: This disk contains four MS-DOS utilities for the H/Z-100 PC computer systems: ScreenSaver, DualScreen, ChangeSpeed, and a Print SCreen utility for Text Modes.

Requirements: All four programs require MS-DOS version 2.0 or greater, and a minimum amount of memory. The ChangeSpeed utility is specifically for the H/Z-200 computer system, while the other three, will run on any H/Z-150/160/200 or IBM compatible. CS, requires that both the Z-309/409 color graphics card and the Z-329 high resolution monochrome card be installed in the computer system, as well as a monitor be connected to both cards.

The following files are included on the HUG P/N 885-6009-37 Screen Saver Plus disk:

SS	.ASM	SS	.COM
DS	.ASM	DS	.COM
CS	.ASM	CS	.COM
PSCTM	.ASM	PSCTM	.COM
README	.DOC		

Authors: SS, DS, and CS, by Jim Buszkiewicz, PSCTM by Pat Swayne.

SS (ScreenSaver) is a program specifically for the H/Z-100 PC series of computers (H/Z-150/160/200) or IBM compatible. SS will, after a predetermined length of time of screen and keyboard inactivity, blank the color graphics screen or high resolution monochrome graphics screen, eliminating the possibility of 'burned' phosphor on the CRT. Upon any key entry, or any screen activity, the original screen information will be restored and updated. ScreenSaver works in all video modes except 4 and 5 (medium resolution color graphics mode), which is normally used for game playing. ScreenSaver can be set for any length of time between 1 and 60 minutes of delay before blanking the screen because of inactivity.

DS (DualScreen) is a utility that allows any text that appears on the color graphics screen to appear on the Z-329 high resolution monochrome screen. This eliminates the need for changing video modes when doing word processing or editing. The 'action' of this program can be turned 'on' and 'off' at will, once it has been installed in memory. Take note that, due to the time needed to refresh the video screen, this utility will not work simultaneously with modem communication packages.

CS (ChangeSpeed) is a utility for the H/Z-241 series of computers. It is also the same program that appeared on page 50 of the March 1986 issue of REMark magazine. This utility takes advantage of an undocumented feature of the '241, and allows the user to change the speed of the computer to that of a standard 5 Mhz H/Z-150/160, and then back again to the computer's normal speed at will.

PSCTM (Print SCreen utility for Text Modes) is designed to be used in conjunction with a graphic PSC (Print Screen) utility, to allow it to work in the text modes. It allows you to print a duplicate of what is on the screen, including text mode graphic characters and special symbols. It works with any printer for which you have a graphic print screen utility.

TABLE C Rating: (10)

Attention HERO 2000 Owners!

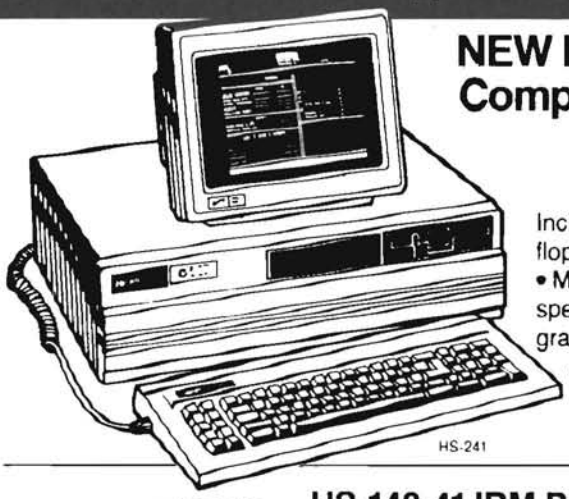
Looking for an inexpensive modem package that'll interface your H-100 or H-100 PC to your new super robot? Look no further! HUGMCP was designed with HERO 2000 in mind! HUGMCP responds properly to the ANSI 'disable keyboard' command as required by HERO 2000. It will also do software transfers using standard XMODEM protocol, or ASCII transfers using XON/XOFF protocol. HUGMCP is available for only $40 from HUG, and has a part number of 885-3033-37. Why pay 4 times more than you really need to? Order your copy today from the Heath Parts Department by calling area code (616) 982-3571. *

Heath/Zenith Related Products

Jim Buszkiewicz
HUG Software Developer

Software Toolworks and **Intersecting Concepts,** co-authors, announce the availability of a new hardware/software package called ACCELERATE 8/16. This new product allows MS-DOS users to read CP/M disks and run CP/M software on their MS-DOS machines. The hardware component of ACCELERATE 8/16 is the NEC V20 processor, which simply replaces the 8088 processor in most MS-DOS computers. A bonus feature of the V20 chip is that MS-DOS programs will run as much as 40% faster. What makes this product so valuable is that all the software and data CP/M users spent years collecting can be move to any MS-DOS or PC-DOS computer in minutes, with no loss of usefulness. The software component of this product consists of two programs; Media Master, a disk-to-disk format conversion program, from Intersecting Concepts, and a new CP/M emulation program called ACCEL from Software Toolworks. ACCELERATE 8/16 without the NEC V20 is $89.95 and with the V20, $99.95. ACCELERATE 8/16 is not copy protected. For more information, contact the Software Toolworks, 14478 Glorietta Drive, Sherman Oaks, CA 91423, or call (818) 986-4885. *

IBM AT Compatible Personal Computer only $2899


HS-241

NEW IBM AT Compatible Advanced Personal Computer Kit

Priced Hundreds of Dollars Less Than the IBM AT!

Only **$289.00** Less HUG Discount

Includes: • 512K RAM expandable to over 15MB • single 1.2MB 5.25" floppy disk drive • controller card for two floppy and three hard disk drives • MS-DOS operating system • 8 expansion slots • 6 MHz clock for high speed processing • compatibility with almost any PC video card for color, graphics, and high resolution video options. **Monitors from $79.95**

See the catalog for SuperSaver discounts on accessories.

Take your HUG Discount on Heathkit and Zenith Data System Products

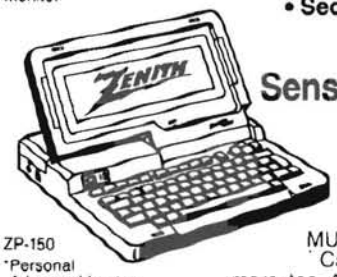

HS-148-41 IBM-PC Compatible Computer Kit with HVM-122A Monitor

HS-148-41 IBM PC Compatible Computer with Disk Drive and 256K Memory

Kit Only **$899.00**
With HUG Discount

• MS-DOS operating system • accesses IBM software • optional 10 Meg, 20 Meg and 30 Meg Winchester hard disk drive available • 360K disk storage • 256K memory expandable to 640K • start-up diagnostic • famous Heath/Zenith quality

• Expand to 640K Memory for Just...$110.00
• Second Disk Drive Just $125.00

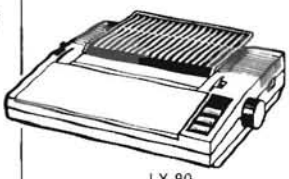


ZP-150 *Personal Advanced Laptop

Sensational price on this new PAL* LAPTOP COMPUTER

7½ lb. briefcase size unit has Microsoft WORKS, WORD, MULTIPLAN, and BASIC...Modem.. Calendar...Alarm...all built in. Much more, too. Act now for this unbelievable price.

Apply your HUG Discount **$899.00**


LX-80

Epson LX-80 Printer

Just **$239.00**

• Correspondence quality FRICTION FEED printer prints directly on your letterhead • Fast! 100 characters per second • Select typestyles at a touch. Over 150 combinations • Mix text and graphics on same line • One-Year warranty.

SYQUEST 10 MEG REMOVABLE WINCHESTER CARTRIDGE DRIVES
Includes drive, one 10 meg cartridge, controller board **$1195.00**

ZOOM/MODEM PC 1200 INTERNAL 1200/300 BPS Modem
For PC, XT, and AT computers. Hayes-compatible. Includes touch tone decoder, onboard RAM, battery backup **$299.00**

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
While quantities last **$449.00**

Smart Outlet Box with Increased Protection
Provides single switch control of up to seven electrical devices. UL approved with new, superior surge protection and noise filtering . . **Special $49.95**

NASHUA DSDD Diskettes . . . . Per Box **$12.95**

ZENITH data systems
AUTHORIZED DEALER

LEASING AVAILABLE FOR YOUR BUSINESS NEEDS

Fine Electronics Since 1962

HZC-209

Prices good through June 15, 1986.
Call 616-982-3614 for store locations or consult your Heathkit Catalog.

EXPERTS: CAD NETWORKING COMMUNICATIONS

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

Heath® ZENITH®
Computers & Electronics

Units of Veritechnology Electronics Corporation

Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

BULK RATE
U.S. Postage
PAID
Heath Users' Group

POSTMASTER: If undeliverable, please do not return.

P/N 885-2076