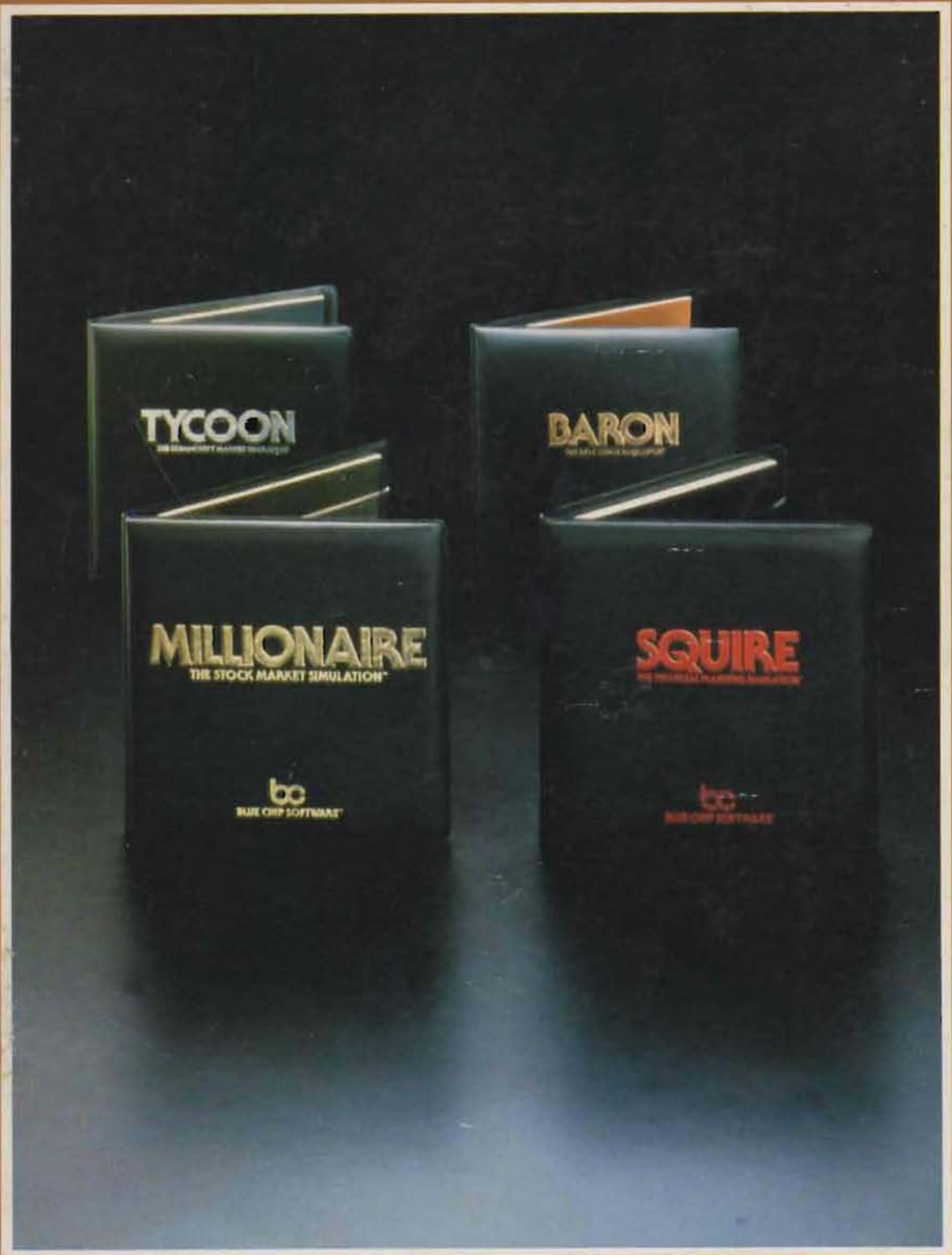
REMark®

Volume 6, Issue 8 • August 1985

P/N 885-2067 Issue 67

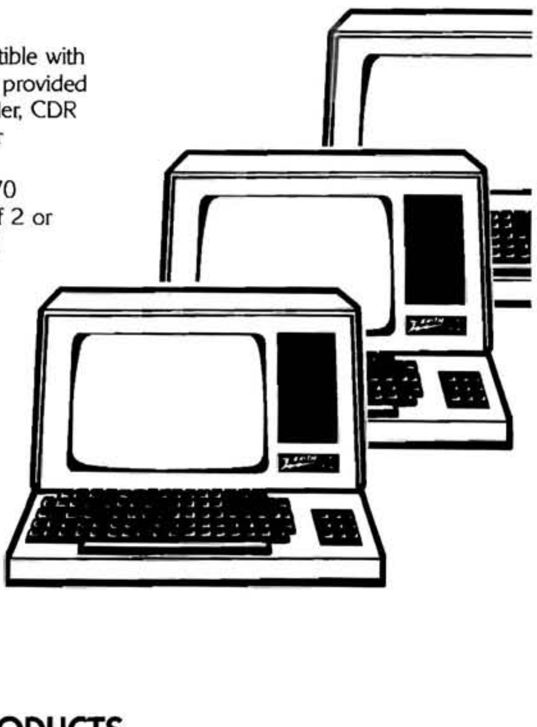
Official magazine for users of HEATH ZENITH computer equipment.

$2.50



Support and More (for Less)

Energize your tired old computer with...



SUPER 89 BIOS

The Super 89 BIOS is a greatly expanded operating system interface fully compatible with that provided with H/Z CP/M for use with the Super 89 and Heartbeat. Support is provided for all H/Z disk drive systems as well as Magnolia Microsystems 8"/5" disk controller, CDR systems FDC-880H, and hard disk systems using the popular XEBEC Winchester controller.

Expanded facilities of the BIOS include a built-in RAM Disk with capacity up to 170 Kbytes; a 48 Kbyte RAM print spooler; transparent operation at CPU clock rates of 2 or 4 MHz; increased TPA (Transient Program Area) for larger user programs; detailed error messages and user selection of recovery options; real-time clock support; simplified, self-prompting, configuration utility that is truly user-friendly; also, included are several useful utilities for expanded system operation. The Super 89 BIOS is available separately, $60.

THE SUPER 89

The DG SUPER 89 is a replacement central processor board for the Heath/Zenith 88-89 series of computers. The DG SUPER 89 offers advanced features not available on the standard Heath/Zenith 88-89 such as 4 MHz operation, real-time clock, optional AM9511A arithmetic processor, up to 256K of bank selectable RAM with parity check, and HDOS, CP/M and MP/M compatibility. By incorporating the state-of-the-art technology of the Z80, the DG SUPER 89 offers the user increased speed and system reliability for years to come. Super 89 BIOS included. Full compatibility with all Health/Zenith software and hardware products is designed into the DG SUPER 89. Price: $699.00 (128K) – $739.00 (192K) – $779.00 (256K).

HEARTBEAT

The DG Heartbeat is a compact computer system designed to be hardware and software compatible with the popular Heath/Zenith Z89/90 computer product line. The Heartbeat offers advanced features not found on the standard Heath/Zenith computer such as 4 MHz operation, real-time clock/calendar, two RS-232 serial ports, five peripheral expansion slots, 128 Kbytes (expandable to 256 Kbytes) parity checked RAM and provisions for an optional AM9511 Arithmetic Processor. Compatible with HDOS, CP/M and MP/M II (Multi-user) operating systems. Super 89 BIOS included. The Heartbeat may be used with most popular video terminals on the market although the Heath/Zenith H/Z19, H/Z29 and ZT-10/11 video terminals are recommended for full Heath/Zenith software compatibility. The Heartbeat cabinet design provides for inclusion of hard and/or floppy disk drives as well as other desired peripheral interfaces and is color-coordinated for use with the Zenith Z29 and ZT-10/11 video terminals. Price: $995.00 (128K) – $1075 (256K).

H8 PRODUCTS

The Most Extensive Line of Hardware Support for the H8®

- **DG-80/FP8**
  Z80® based CPU including the powerful FP8 monitor — both only $199.00. Revitalize your H8 with the speed and power of a Z80 CPU.
- **DG-64D/64K RAM Board**
  Reliable, Low Power, High Capacity Bank-selectable RAM Priced from $233.00 (0K) to $299.00 (64K)
- **DG Static 64**
  Fully Static, High Capacity, Bank-selectable RAM. Also can be used as EPROM/PROM board (2716 type EPROMS). Priced from $199.00 (0K) to $399.00 (64K).
- **DG-32D/32K RAM Board**
  Low cost, Dependable RAM for the H8 32K Version Only $179.00.
- **DG-ADP4**
  H17-4 MHz disk adaptor — $19.95

CP/M®, MP/M and MP/M II® are registered trademarks of Digital Research of Pacific Grove, California.
H88/89®, Z89/90®, H17®, H77®, H/Z 47®, Z67® and H-88-1® are registered trademarks of the Heath Company and Zenith Data Systems.
Z80® and Z80A® are the registered trademarks of Zilog Corporation.

D·G ELECTRONIC DEVELOPMENTS CO.

Ordering Information: Products listed available from DG Electronic Developments Co. 700 South Armstrong, Denison, Tx 75020. Check, Money Order, VISA or MasterCard accepted. Phone orders call (214) 465-7805. Freight prepaid. Allow 3 weeks for personal checks to clear. Texas residents add 5%. Foreign orders add 30%. Prices subject to change without notice.

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

REMark

Editor .... Walt Gillespie
(616) 982-3789

Editorial Assistant .... John Guldan

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

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

* U.S. Funds

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

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

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

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

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

REMark®

Official magazine for users of HEATH ZENITH computer equipment.

Volume 6, Issue 8 • August 1985

Index of Advertisers

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

On The Cover: *Pictured are four new products from Blue Chip Software, 6744 Eton Avenue, Canoga Park, CA 91303. These financial simulations are reviewed in this month's REMark beginning on page 12.*

Photo courtesy of Blue Chip Software.

H/Z STORAGE FROM QUIKDATA

Stick with Quikdata, Inc., the oldest independent Heath/Zenith vendor for all your computer storage needs. We have a complete line of memory chips, disk drives and winchester systems for all Heath/Zenith computers; H8, H/Z89/90, H/Z100 and H/Z150 series.

BARE FLOPPY DRIVES

Tandon TM100-1 SS ST (H17 type replacement)	$139
Tandon TM100-2 DS ST (48TPI H37, Z100 type)	$159
Tandon TM100-4 DS DT (96TPI Quad density)	$149
Tandon TM101-4 DS DT (96TPI Quad, newer type)	$249
Mitsubishi 4851 DS ST thin (like 100-2)	$139
Mitsubishi 4853 DS DT thin (like 100-4)	$149
Tandon 848-2E 8" DS thinline	$379

DRIVE SUBSYSTEMS

Dual 8" system includes cabinet with fan and power supply, two 8" Tandon 848-2E thinline drives and cable. Ready to plug into Z100 or H89 with Magnolia 77316 controller for 1.2 megs storage per drive.

Dual 8" subsystem	$969
Same, but single system	$595

H89 internal thinline system includes two Mitsubishi thinline drives, controller cable set and extra 5V power connector. Requires drilling 4 holes in mu-metal shield and mounting drives.

Dual internal 48TPI setup	$308
Dual internal 96TPI setup	$328

CABINETS

5" single horiz full size w/o fan or cables	$59
5" dual horiz full size with cables and fan	$175
5" dual horiz thinline w/o fan or cable	$95
5" winnie cabinet with fan and switching PS	$229

WINCHESTER DRIVES & SYSTEMS

We handle a few different winchester units which we use for all our systems. Note the megabyte size; 19/15, for instance, means 19 megs unformatted and 15 megs formatted. When only one size is given, it's the **formatted useable size,** not the unuseable size which many other vendors give you to trick you into thinking you're getting more than you are! Beware of these tricks when comparing sizes and prices! We have no intention of snowing or ripping off our customers.

BARE WINCHESTER DRIVES

Tandon TM503 19/15meg	$595
Rodime R203E 40/33meg	$1195

Z100 WINCHESTER SYSTEMS

These internal Z100 winchester systems assume your Z100 is winchester ready, i.e., has the two extra cables coming out of power supply for winchester cards. Setup includes new front bezel set, new 2.5 ROM, S100 card and winchester card, software and documentation.

Z217 controller set only	$995
Z217P front panel, ROM, hardware	$100
ZS217-10 10meg complete system w/drive	$1495
ZS217-15 15meg complete system w/drive	$1595
ZS217-32 32meg complete system w/drive	$2395

MEMORY CHIPS

Note: 4116 chips are used in some H8's and all H89's. 4164 chips are used in Z100 and Z150 systems. 41256 chips are used in modified older Z100 and Z150 systems and in new machines unmodified. Z100 & Z150 require 9 chips per memory bank because of parity. Memory chip prices are falling - call!

4116 150ns 16K memory chips	$1.95
4164 150ns 64K memory chips	$1.95
41256 150ns 256K memory chips	$4.49 ~~$5.49~~

H8/H89 QUIKSTOR WINCHESTER SYSTEMS

These **external subsystems** will work with the H8 via WH8-37 controller, and with the H89 via Z67 or Magnolia 77320 SASI host adapter card (which also includes three serial ports same as on the Heath card). Software features: 2/4MHz software, up to 15 user defined bootable partitions, HDOS 2.0 and CP/M 2.2.03 and 2.2.04 supported, up to 4 active CP/M partitions at one time and 8 HDOS partitions active at one time, multiple winchester support, supports any size winchester (we supply the 15 and 32 meg units), CP/M BIOS-80 support for H17/winchester systems, BIOS ZCPR option, easy installation, more!

Subsystem includes cabinet with fan and switching power supply, Xebec 1410 controller, drive and cables. **Slave** unit is for multiple winchester systems and does not include Xebec controller.

15 meg formatted subsystem ready to go	$995
33 meg formatted subsystem ready to go	$1695
15 meg slave unit ready to go	$795
33 meg slave unit ready to go	$1495
77320 H89 host adapter card	$225
Used WH8-37 for H8 (as available)	$249
QUIKSTOR software package with system purchase	$149

Note: If purchasing **only** the QUIKSTOR software to make up a system yourself support will be minimal since you assume the role of system integrator. Price for software/documentation **without subsystem purchase is $195** and requires license agreement to be filled out prior to shipment.

Z150 WINCHESTERS

These are our internal Z150 systems which include Xebec controller, winchester full sized drive (mounts under top floppy), cables, enhanced ROM permitting booting from any of up to 4 partitions and faster boot, enhanced PREP for a 300% winchester speed increase, and documentation.

16 meg system	$895
33 meg system	$1495

Z150 TAPE BACKUP SYSTEM

10 meg tape backup system for Z150 winchesters. Allows fast backup of winchesters by image, directory or file. Typical time is 8 minutes for 10 meg backup, with 16 minutes for backup and verify. Complete with cables, software, doc and tape. Tapes can be 'split' for large files or winchesters.

TAPESYSTEM	$695
Spare certified and tested tapes	$24

MEMORY BOARDS AND PRODUCTS

H89 16K memory expansion board for 48K systems $69

Z150MP is a memory mapping pal replacement for Z150 memory boards to allow the 256K chips to be used on a single board to give up to 640K memory when mixed with the 64K chips.

Z150MP	$19
Z100MP similar but for newer Z100's	$76

Z100 MEMORY BOARDS by UCI

These boards are capable of being populated with either the 64K or 256K chips for storage from 256K to 2 Megs! Bare board has all parts except memory chips. All boards above 256K will have 256K chips populated as noted below.

UCIB Bare board	$395 ~~$419~~
UCI2 256K board	$429
UCI5 512K board	$519
UCI1M 1 meg board	$619
UCI2M 2 meg board	$829

ZENITH data systems

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

VISA MasterCard
VISA and MASTERCARD

. . . AND THAT'S NOT ALL!

Of course we have more than storage devices! Quikdata is your full source supplier for **all** your H/Z computer needs. We carry a full line of proven hardware and software, and sell at the lowest possible prices. **CHECK US OUT:** *Huge inventory of products means 24 hour shipping on all orders whenever possible. *We **stand behind** everything we sell. ***Full service and support** mail order company. ***Tech line** for software and hardware support and consultation. ***In house repair** facilities; we repair what we sell. *Check out our **prices**! **The lowest available.** *Support only the Heath/Zenith line of computers. ***WE CARE** and we mean business!

Here are just a few examples. Call or write for a free catalog to obtain our current product line listing, to order, or receive additional information on any of our products.

SOFTWARE RELATED

SPELLBINDER word processor w/dictionary	$295
PIE text editor	$27
THE WORD PLUS Spell checker	$125
SPELL Spell checker	$47
dBASE II	$389
dBASE III	$495
FRAMEWORK	$495
RAMDRIVE-100 Z100 700K software	$47
RAMDRIVE-150 Z150 RAM drive software	$47
C/80 Compiler	$47
COMPUTER CHEF	$27
DKH17 HDOS 400K hard sector driver	$40
BIOS-80 CP/M 400K hard sector driver	$40
Z-GRAPH-100 Z100 Graphics support	$79
SIDEKICK super organizer for Z150	$45
NORTON UTILITIES for Z150 v 3	$79 ~~$69~~
FLIGHT SIMULATOR for Z150	$43
MIND PROBER for Z150	$44
NIGHT MISSION PINBALL for Z150	$35
LOTS OF GAMES AND EDUCATIONAL SOFTWARE	

PRINTERS & RELATED

Note: The dot matrix printers are Epson code compatible, while the 3151 is Diablo 630 compatible!

DIABLO 630 serial/parallel 40cps daisy	$1695
PANASONIC 3151 22cps daisy parallel	$595
Serial interface for 3151	$89
PANASONIC 1090 80cps 80 col dot mtrx	$295
PANASONIC 1091 120cps 80 col dot mtrx	$339
PANASONIC 1092 180cps 80 col dot mtrx	$479
PANASONIC 1093 160cps 132 col matrix	$659
ANGEL 64K serial/parallel buffer	$269
RS232 TRANSFER SWITCH	$69

MODEMS & RELATED

HAYES SMARTMODEM 300	$239
HAYES SMARTMODEM 1200	$479
HAYES 1200B Z150 INTERNAL w/software	$495
ACCESS Modem software - prices vary for version	

MISCELLANEOUS

ISO-BAR SPIKE PROTECTOR	$79
TRANSTECTOR SPIKE PROTECTOR	$99
425W STANDBY BACKUP POWER SUPPLY	$429
1200W LINE CONDITIONER	$169
1800W LINE CONDITIONER	$229
ZVM135 color monitor	$495
ZVM 123A anti-glare green monitor	$113
ROTRON Sprite Fan for H89	$24
CABLES: RS-232, Parallel, Disk drive, Winchester	
DYSAN Diskettes: All in plastic library cases	
104/2D 5" DS DD ST Soft	$31
107/2D 5" ST DS DD Hard	$31
204/2D 5" DS DD DT Soft	$43
207/2D 5" DS DD DT Hard	$43
3740/1 8" SS SD Soft	$31
3740/2D 8" DS DD Soft	$36
DYSAN analog and digital alignment disks	
Magnolia H89 products	
Kres 2/4 MHz H89 mod	
MICROCHART INSTANT DATA CARDS (5 types)	$5.25
HALON 14oz Micro fire extinguisher	$34
USED H89 AND Z100 BOARDS AND PARTS	

We also have a full line of computers, monitors, cables, diskettes, books, etc. Since Zenith is changing its entire line of computers as of this ad printing, we will not include the different computer systems in this ad. Please call to get the lowest prices on the new models!

DON'T FORGET H-SCOOP!

Want to get the most from your computer investment? If you are not already a subscriber to **H-SCOOP**, you're missing the boat! **H-SCOOP** is the most complete independent newsletter supporting the H8, H89/90, H/Z100 and H/Z150 series of computers. Monthly information includes: *What's going on in the H/Z related world! *Where can I go to get all the hardware and software I need and at the lowest prices! *What hardware and software do I need! *Hardware and software reports. *New vendor information - but we don't allow that to be more than 20% of our monthly newsletter. *Classifieds. *Who's doing what with their machines. *Need help? Try **REQUEST** column. *H-SCOOP disk library for low cost software. *What are **you** doing - it goes in **WHO'S WHO** column. *And that's just the start!

H-SCOOP RATES: $20 for a 12 issue year **USA First Class Mail** (includes CANADA), $27 foreign **AIR MAIL. SPECIAL DEAL!** Start with the current year as of January '85 (that means your subscription expires with the December '85 issue!) and get all available back issues for an additional $20 for a $40 total ($54 total foreign)!

Feel free to request a catalog or any additional information on any product we carry. Personal letters and phone calls welcomed! Note: H-SCOOP is a subsidiary of QUIKDATA, INC. Prices listed in this ad are subject to change without notice (prices usually go down on memory devices and disk drives, including winchester drives). Call to be sure. Any overpayement will be refunded, any underpayment will be invoiced. These listed prices are **cash prices** only. PO's from universities or established companies listed in D&B add 5% NET 30 payment charges and 3% S&H charges. COD's add $5 COD charge and 3% S&H (payment by Cashier or Certified check). VISA and MC accepted, add 3% S&H ($2 minimum).

BUGGIN' HUG

CP/M MAPLE Patch

Dear HUG:

I have received several letters concerning the modem program, MAPLE Versions 2.0.6 and 2.1.0 for CP/M.

There is a three byte patch required to run the 2.1.0 version under CP/M 2.2.04. This can be done as follows using the program DDT supplied with your operating system. Where you see a double quote, you will be required to type the characters enclosed by the quote, but not the quote itself. A single quote will inclose the screen response. A carriage return is signified by <cr>.

1. Put a copy of MAPLE.COM and DDT.COM on one disk.
2. Type "DDT MAPLE.COM<cr>" after the operating system prompt A>
3. After DDT introduces itself, you will see a '–' prompt. Type "S2E8F<cr>" after this prompt. This will put you in the Substitute byte mode at the hex address given.
4. DDT will repeat the address and show you the current byte at that address: '2E8F DA' and wait for your substitute byte. Type "00<cr>".
5. DDT will give you the next address and current byte: '2E90 94'. Type "00<cr>".
6. Again, after the next address and byte '2E91 2E', type "00<cr>".
7. Now exit the substitute byte mode by typing another "<cr>", and then exit DDT by typing control-C.
8. You must now save the memory image of MAPLE.COM. After the system prompt 'A>', type "SAVE 52 MAPLE.COM<cr>".

Sincerely,

William C. Parke
Author, MAPLE

Working Fine At Single-Density, Wants Double-Density

Dear Pat/HUG:

Keep up the good work Pat. It's appreciated.

Can you/anyone provide the patch locations to the MS-DOS 2.11 device tables and to the FORMAT program to allow my H-120 to use my pair of Shugart 801 8" floppy drives at double-density. They are working fine at single-density (/M), but 250 KB is most frustrating when I get 500 KB under good old CP/M 85!? 801's are single-sided.

Larry Snyder
S78 W17675 Canfield Drive
Muskego, WI 53150

A Patch For PUTSASI.COM

Dear HUG:

This is for any H-88/89/90 computer owner who may have recently purchased a 10 meg hard disk drive from Floppy Disk Services and wish/need to use the PUTSASI.COM from Magnolia Microsystems. My computer has only one H-17 disk drive (hard sector) which requires me to use this utility. After receiving and installing my drive, I wanted to change the partitions. This required me to format the drive using my 100k drive to run the software. I built the new MOVCPM.COM file, ran MOVCPM, and then tried to use PUTSASI.COM to place the new system on the drive. To my surprise it came back with 'DISK WRITE ERROR'. After using ZSID to find the problem, I came up with the following patch.

```
A0>DDT PUTSASI.COM
DDT VERS 2.2
NEXT  PC
0900 0100
-A06E9
06E9 MVI B,20                replaces MVI B,10
06EB .
-S02
0295 32 33                   changes vers from 2.242 to 2.243
0296 20 .
-G0

A0>SAVE 8 PUTSASI.COM        save the file.
```

PUTSASI.COM will now work every time. This patch simply increases the counter used in a loop that checks for drive "READY."

Sincerely,

Bruce C. Jones
9985 B Newell Avenue
Ellsworth AFB, SD 57706

Looking For Bowling And Real Estate Programs

Dear HUG:

I have been enjoying your magazine for 3 years now, and during that time I have been looking for a program to run on my H89 — CPM — 64K SSSD drives that would record and calculate bowling league averages, also any real estate appraising programs that are available.

I would appreciate any information you or your readers might have about such a program.

Sincerely,

Warren T. Bambury
29 Teele Avenue
Somerville, MA 02144

Foolproof "Bulletproof Garbage Filters"

Dear HUG:

Kurt Shultz's article, "Bulletproof Garbage Filters" (REMark, March 1985), was interesting to me since I have written a few programs for our company and have been concerned with inexperienced users inadvertently pushing the wrong keys. I have found the INPUT$ function an effective and fast way to filter responses. For example, when used to filter a simple YES/NO question, the following routine is just about foolproof:

```
100 PRINT "Do you want instructions? ";: Q$=INPUT$(1)
110 IF Q$="Y" OR Q$="y" THEN 1000 ELSE IF Q$="N" OR Q$="n"
     THEN 2000
120 PRINT:PRINT "Please answer with a 'Y' or 'N'":GOTO 100
130 '
1000 GOSUB 10000
```

The INPUT$ function will accept the first (and only one in this case) character typed by the user. Since it is not necessary to follow with a carriage return, less key strokes and faster execution is achieved. Line 110 will test the response for both upper and lower case entries and branch the program to the appropriate module. If other than the proper response is entered, line 120 will display the error message and return the user to the initial prompt.

Of course, there are many ways to enhance the routine by inserting a CHR$(7) signal and erasing part of the display after receiving an unacceptable user response. An audible signal from the computer is a must, since it immediately alerts the user that something is wrong.

Incidentally, the routine printed at the bottom of page 58 in Kurt's column doesn't quite filter out all of the garbage. If in response to the question a simple carriage return is typed (a common error with inexperienced users), the INSTR function will cause BRANCH% to be assigned a value of 1 and the program will go to the Instructions.

The reason for this is that the BRANCH%=INSTR (YNKEY$,ANSWER$) statement will return a value of 1 if ANSWER$ is a null, which is what it will be if only a carriage return is typed.

Sincerely,

Joseph Costanza, Jr.
1261 Charleston Road
Cherry Hill, NJ 08034

March '85 Letter By Rodger A. Gray

Dear HUG:

A year or so ago, I had a similar problem with driving a modem. In my case, the problem was with the original serial board which came with the Z-89 system and a null modem connecting the Z-89 to a Radio Shack Model II. The software I used was the MAPLE program (an excellent program!) sold by HUG. The problem showed up when I tried to send characters from the Model II to the Z-89. The Model II acted as though everything was working, sending characters happy as a clam. The Z-89 ignored them all when using the MAPLE program, but picked up all characters when using the PIP utility program. Also all characters could be sent to the Model II, with no trouble, using either program.

It turned out to be neither software nor operator error, but rather the serial board had not been modified to use interrupts with a modem. The solution was to move a simple jumper wire on the serial board to enable interrupts. No more problems!

Perhaps Mr. Gray's problem is different, since the interface boards are not the same. Still, I must sympathize with him, since the people at the Heathkit store, where we had purchased all our hardware and software, were stumped when we told them of our problem. It seems the simplest solution is the hardest to find. We finally found it ourselves. Good luck to Mr. Gray.

Sincerely,

K. Quebbeman
4311 Frederick Place
West Valley City, UT 84119

P.S. The MAPLE (version 2.1.0) program now works perfectly with CP/M 2.2.03, but throws up when using CP/M 2.2.04. The problem occurs when trying to log-on disk B; it cannot find the list of possible drive letters. Has this been reported by others or am I the only one with the problem?

ZPLOTTER

Dear HUG,

In studying mechanical engineering, I frequently am faced with an answer to a problem in the form of a mathematical function. Quite often the function represents an approximation to a real world occurrence and is rather long and difficult to visualize. But what if I want to see the function to get a better understanding of the problem? One answer, here at Lehigh, is to run down to the computing center and write a FORTRAN program which calls our handy plotter. This approach, while still a good one, can be somewhat tedious after just a couple of times. And besides, sometimes I don't need a neat, ultra accurate plot, I just want a quick picture of it so that I can better undertand the answer which I just calculated and, of course, to see whether it is feasible or not.

When I bought my H100, I planned to write a short, simple program which would be the answer to this problem. My old H89 certainly wasn't particularly good at plotting a function on the screen, even though it is an excellent computer in its own right. So, to make a long story short, here is my plotter program written in ZBASIC. I could've opted for FORTRAN, which would be defeating the purpose of this exercise. I wanted to be able to quickly run a plotter and get a pretty good rough idea of what the function actually looks like and I wanted to be able to possibly change the function easily. Well, the ZBASIC full screen editor is perfect for this. While it may not be the fastest language, it suited my needs perfectly for this program. I didn't even compile the program because that, too, would make changing the function quickly rather tedious.

The program itself is rather short. It does what it was meant to do (and quite well), but nothing else. It will even plot a rather difficult function with surprising accuracy. It will plot up to about 15000 data points under the interpreter with 192k RAM, which is more than enough to get a rather good picture of the function.

The program is in two parts, a routine to load an array full of values along the function, and the plotter itself. Try running it as is and then try putting different functions in. The function is found on line number 30. Lines 80 through 140 load the array full of values. Line 150 calls the plotter. The first few lines of the plotter draw and label the axes and finally, lines 52500 through 53200 draw the function on the screen. The program then waits for any input from the keyboard upon which it clears the screen and lists the function part of the program, thus making it even easier to change the function.

I've found the program to be perfect for my needs. It automatically sets the upper and lower extremes of the function being plotted. If these extremes exceed the PRINT USING format size, an overflow will show, but the function will still be plotted. I think this program could also be used as an educational tool for students learning about functions because of its ease of operation.

If anybody has any questions or suggestions about the program, please let me know.

One final note to all of those who work at HUG or contribute to REMark. Keep up the good work! I've been a member of HUG for a couple of years now and I always get worthwhile information out of the magazine. My thanks to everyone who makes it worthwhile!

Kenneth R. Van Wyk
68 New Garden Avenue
Lancaster, PA 17603

```
1Ø CLS
2Ø N=4ØØ:DIM Y(N)          'N is the number of points.
3Ø DEF FNY(X)=SIN(X)       'The function to be plotted.
4Ø YMIN=Ø:YMAX=Ø           'Initialize all minimum and maximum
5Ø XMIN=-4*3.14159         'limits. All of which can be
6Ø XMAX=4*3.14159          'manually altered.
7Ø J=Ø:K=Ø                 'Initialize counters.
8Ø FOR Z=XMIN TO XMAX STEP (XMAX-XMIN)/N  'Loop to fill
                                           array with
9Ø     J=J+1:K=K+1                        'values to be
                                           plotted.
1ØØ    IF J>N THEN 14Ø
11Ø    Y(J)=FNY(Z)
12Ø    IF Y(J)>YMAX THEN YMAX=Y(J)
13Ø    IF Y(J)<YMIN THEN YMIN=Y(J)
135    IF K=1Ø THEN LOCATE 13,36:PRINT J;"/";N;:K=Ø
14Ø NEXT Z
15Ø BEEP:GOSUB 5ØØØØ        'Array full, now call plotter.
16Ø END
5ØØØØ 'Subroutine to plot an arbitrary function
5Ø1ØØ '    The following values must be set as follows:
5Ø2ØØ '    XMAX, YMAX, XMIN, YMIN  max and min values for
           each axis.
5Ø3ØØ '    Array to be plotted Y with N values along X axis.
5Ø4ØØ '
5Ø5ØØ '    Written by Kenneth R. Van Wyk
4Ø6ØØ '
5Ø7ØØ '
5Ø8ØØ IF YMAX=YMIN THEN YMAX=.5:YMIN=-.5   'Just in case
                                            array is
5Ø9ØØ CLS:LINE(4Ø,5)-(4Ø,215),5            'empty...
51ØØØ LINE(4Ø,215)-(63Ø,215),5
511ØØ COLOR 6
512ØØ LOCATE 13,1:PRINT "Y";               'Draw axes and
                                            max/min
513ØØ LOCATE 25,42:PRINT "X";              'values.
514ØØ LOCATE 1,1:PRINT USING "###.#";YMAX;
515ØØ LOCATE 24,1:PRINT USING "###.#";YMIN;
516ØØ LOCATE 25,4:PRINT USING "####.#";XMIN;
517ØØ LOCATE 25,75:PRINT USING "####.#";XMAX;
518ØØ LINE(36,47)-(39,47):LINE(36,89)-(39,89):
        LINE(36,131)-(39,131)
519ØØ LINE(36,173)-(39,173):LINE(158,216)-(158,218):
        LINE(276,216)-(276,218)
52ØØØ LINE(394,216)-(394,218):LINE(512,216)-(512,218)
521ØØ COLOR 7
522ØØ YDELT=YMAX-YMIN           'Initialize some values used by
523ØØ XSTEP=58Ø/N               'plotter and a counter.
524ØØ ICOUNT=1
525ØØ Y1=INT(215-(((Y(ICOUNT)-YMIN)/YDELT(*21Ø))
        'X-Y point to screen
526ØØ FOR I=4Ø TO 62Ø STEP XSTEP         'position algorythm.
527ØØ  ICOUNT=ICOUNT+1:IF ICOUNT>N THEN 532ØØ
528ØØ  Y2=INT(215-(((Y(ICOUNT)-YMIN)/YDELT)*21Ø))
529ØØ  IF INT(I)=INT(I+XSTEP) AND Y1=Y2 THEN 532ØØ
53ØØØ  LINE(INT(I),Y1)-(INT(I+XSTEP),Y2)
531ØØ  Y1=Y2
532ØØ NEXT I
533ØØ A$=INPUT$(1):CLS:LIST -49999
534ØØ RETURN
```

Looking For Help

Dear HUG:

I'm looking for some help with my H89A.

1. The two (2) beeps I get on power-up are very weak, as is the keyclick. On rare occasions they are "normal", but only for a few minutes.
2. On reverse video I'm plagued with "flickering." In correspondence with Heath Company they called it "Hum Bars displayed on Screen or Random Oscillation." They sent me twelve 330 pf capacitors to install on my 85-2550-1 serial board. The results were negative.

Sure would appreciate hearing of solutions to the above two problems.

Robert A. Speidel
Box 95E, Route 1
Emmaus, PA 18049

OKIDUMP.BAS Helpful

Dear HUG:

I have been very busy lately and unable to voice my opinion on some of the helpful articles and programs that have been listed in REMark. Finally though, I would like to commend Timothy Ross for his article entitled OKIDUMP.BAS (June 1984). I am one of the few people that owns a Zenith and has hooked it up to an Okidata 92 printer. Until Timothy's article I was completely perplexed on how to do graphics on my '92. I must admit that nine minutes is a little much to wait for one picture, but I am sure that the time factor can be improved.

On a different note I would like to commend Zenith Data Systems for their support after buying the Z-100 computer. Just today I received my ZDOS update (Version 1.25) and I was pleasantly surprised to find that Zenith is still sending updated software to its customers instead of leaving them to fend for themselves. I am very happy with REMark and hope that there will be more articles such as OKIDUMP in the future. (I hope to find out more about the coming Gemini board for the 100 soon.)

Sincerely,

Chris Carson
Vice President of Development
Peppermint Programs
RD#1 Box 543
Coopersburg, PA 18036

Modem Connecting

Dear HUG:

In the March issue of REMark, Rodger Gray asked for help connecting a modem to the serial port on his Z-90. As anyone who has ever tried to interface pieces of serial equipment can tell you, there are lots of tricky hardware and software problems that can

Vectored to Page 81 ☛

Heath/Zenith Computers & Electronics welcomes you to the 4th International

HEATH USERS GROUP CONFERENCE

HYATT REGENCY O'HARE, CHICAGO, AUGUST 9-10-11

We're celebrating with SPECIAL SAVINGS like these at the Conference and in all stores August 9, 10 & 11.

ZP-150 PAL 7½ lb.
LAPTOP COMPUTER
Only **$999**
3 DAYS ONLY
AUGUST 9, 10, 11
Reg. $1995

HSA-1128-41
ALL-IN-ONE 8 MHz
COMPUTER SYSTEM
SUPER SURPRISE PRICE!

HS-1108
LOW-PROFILE 8 MHz
COMPUTER SYSTEM
Only **$1399**

HSA-1161-21 PORTABLE
IBM-PC COMPATIBLE
128K WITH WORD
AND MULTIPLAN
INCLUDED
Only **$1165**

ZF-111-32
LOW-PROFILE 192K
DUAL DISK COLOR
COMPUTER SYSTEM
256K COMPATIBLE
Only **$1599**

HUG DISCOUNTS ARE INCLUDED IN THESE PRICES

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

HEATH/ZENITH COMPUTERS... "STEAL" 'EM IN THE CARTON
DISCOUNTS TO 80%
ALL SALES FINAL!
JOE'S "BLUE LITE" SPECIALS EVERY HOUR
RAY'S "RED LITE" SPECIALS FREQUENTLY
FULL WARRANTY
BOB'S MYSTERY BOXES OF GREAT STUFF CHEAP!
"MEMORY CHIPS, DISKS, MONITORS, PRINTERS & MORE" GET 'EM WHILE THEY'RE HOT AT CURTS DISCOUNT KORNER!

Heath ZENITH®
Computers & Electronics

Units of Veritechnology Electronics Corporation

New Products From Blue Chip Software

Following are reviews of four new Blue Chip Software, Inc. products: Tycoon, Baron, Squire, and Millionaire. The reviews were done on the MS-DOS versions of these products which will operate on the H/Z-150 PC family of computers. A review of the CP/M version of Millionaire was previously published in the July 1983 issue of REMark; its operation is almost identical with that of the MS-DOS version.

Millionaire

Review By Tom Huber
Lead Computer Publications Writer
Heath Company

Essentially billed as a "Stock Market Simulation," Millionaire simulates the rise and fall of the stock market (and perhaps your savings as well). Five major industries (Computer, Oil & Gas, Retail, Auto, and Heavy) are represented by three stocks from each group. Companies typically include Control Data, IBM, and NCR (Computers); Conoco, Exxon, and Mobil (Oil & Gas); K-Mart, Sears, and Tandy-Radio Shack (Retail); GM, American Motors, and Bendix (Auto); and United States Steel, Dow Chemical, and Caterpillar Tractor (Heavy).

As the game progresses through each week, it reflects the somewhat typical behavior of the market with its overall trends, peaks and valleys, and maverick stocks (those that seemingly defy the rest of the market — for good or bad). The activity for each week is illustrated with a graph showing previous weeks' performances as well as a selected graph of one of the five industries (based on the performance of the three stocks of that group). The other industry's graphs may be viewed as well as those for individual stocks.

Weekly news bulletins, which affect stock performance, are randomly displayed. By carefully monitoring these bulletins, one can sometimes judge performance of an individual stock. Occasionally, no news is displayed. When that happens, the market tends to rise a bit, fulfilling the old adage that "no news is good news". However, I've had a long losing trend continue right through two or three weeks worth of "no news". So much for the old adage.

Capsule summaries of stock activity, showing the year's high and low, week's close, and change for each stock is also produced. New highs and lows, the number of stocks that were losers and winners, the gain or loss of the market in points, and the performance of an average share is shown.

Play of the game is straight-forward; you buy and sell stocks. Of course, knowing when to buy or sell can be somewhat tricky. The simulation supports corporate histories, loans, margin accounts, and call and put options. The basic functions are explained in the documentation, but a trip to the local library may be in order to understand what to do or how to handle transactions and terms which you may not be familiar with.

The documentation and packaging are well done, at a professional level. On the other hand, the lack of some information may seem to be a shortcoming until you realize that this is not a game; it is a simulation. Simulations are supposed to be learning experiences and by leaving out details concerning specific types of transactions (how to handle them), one is forced into doing some research. The results are a greater understanding of the stock market, and that, to me, ranks this simulation higher than those that supply too much information.

Even though this is a simulation, there is a goal which is clearly established: to become a millionaire by manipulating stocks. Like the real world of stocks, it is not easy to reach this goal. The play is divided into 91-week sessions, in which you advance from novice to investor, speculator, professional, broker, and finally to millionaire. Each session takes about an hour to complete, and I have experienced everything from a gain of almost 180 percent to a loss of 20 percent. My son did much worse and quit early on several occasions. But after watching me play through a game or two, he tried again with much better results.

The play is excellent, with plenty of random happenings to keep things interesting. However, we did note several irregularities with real world stocks, mostly in the form of the news bulletins. For instance, there was a news bulletin reporting dividend earnings for a stock that is not a dividend stock, but a growth stock. In several instances, "wildcat" strikes were reported for companies that have never experienced such labor problems. Items of this nature have no bearing on how the game is played;

they simply do not reflect the real world. Because of this and the limitations to the realism achievable, the program (or any other simulation, regardless of what it is or where it comes from) should not be construed as being exactly identical to the real world. This simulation does come the closest of any simulator to real life stock market investing and activity as any I have seen.

In summary, I give this product high marks all the way around, especially as an educational tool when combined with additional assignments. As a simulation, it is top-notch. As a game, it is a bit slow and repetitive, albeit interesting because of what can happen over the period of several 91-week sessions.

Baron

Review By Pat Westfield
Junior Copywriter
Heath Company Advertising

The folks from Blue Chip Software call Baron a real estate "simulation", but don't let that word fool you. Baron can be as exciting and stimulating as your favorite arcade game. Only with Baron the agony of defeat is more painful because it's not just your battered joystick that's failed you – your precious mind will have failed to keep you from seeing that you shouldn't have bought that house near the fault line, or that Kansas land is a loser in the market.

Not only is Baron exciting, but it's guilt-free. You don't have to feel bad about spending four hours playing instead of mowing the lawn or doing the laundry. Because this simulation lets you exercise your mind. And, you might even learn a little bit about real estate in the process of having all that fun.

In Baron, you start out as a novice real estate investor with $35,000 cash. As a novice, you may buy and sell property, take out 80% loans, purchase options on property, invest in second mortgages and make speculative real estate investments. As your net worth increases, so does your title. At $100,000 you are an investor able to get 90% loans, at $250,000 you are a broker eligible for low closing costs of 4%, and at the $1 million mark you are a real estate baron. You can buy this game without knowing what an option is and still become a millionaire, because everything you need to know is carefully explained in the manual. It reviews terminology, investments and their pros and cons, and has a glossary of terms in the back. It also takes you through a sample game, which is stored on the disk, so that you can get a feel for how the real estate market and the simulation work before you are on your own.

The simulation itself is menu driven and provides you with data on the residential, business, and land markets in five states. It also displays newspaper headlines which can foretell the future of various markets. You can also reference a general description of a state or real estate term. Don't be intimidated by all the figures, line graphs and charts. They are actually quite simple, and you will quickly learn how to interpret them when you become involved in the simulation. Either that – or you will quickly go bankrupt a couple of times! Each simulation lasts 60 months. Every time you advance a month, the market data for the properties you own or options you hold is displayed. If you own Florida land, a graph will automatically be displayed which shows how Florida land prices have been doing. You can select a number of graphs and charts to explore the market from many different angles. After assessing all this information, you may purchase property or six month options, invest in the second mortgage market, or in speculative investments, including limited real estate partnerships, real estate trust funds and Uncle Herman's Project. When considering a transaction, you may view fairly detailed information. If considering residential property, you view a real estate listing showing items such as footage, taxes, and net cash flow. You also get all the usual ambiguous real estate jargon such as: "great fixer-upper" and "cozy".

Like the "real" real estate business, the future may hold a surprise or two. When you advance a month, you may find that the "great fixer-upper" needs fixing up or that your gardener has rototilled your underground sprinklers and it's going to cost you $895. And just because you want to sell that rocky farmland doesn't mean you will find a buyer when you opt to sell it. Each new simulation is composed by the program before you begin, so each is different. You can also continue to play after you reach the baron status and go for your second million.

Yes, I did think of a couple of negative things to say about Baron. These are mainly in regard to the graphs. The graphs could be much more helpful if they went back about six more months so that it would be easier to see trends. The graphs can also be misleading because the scale changes from month to month. A graph may be based on increments of 1,000 in July and increments of 3,000 in August. You have to watch the scale or you may think a change or trend is greater or less than it actually is. You may also be confused by changing scales since the shape of the graph may change somewhat when the scale changes. As would be expected, the simulation has some limitations. It does not account for some important investment factors such as depreciation of business and residential property. The manual, however, explains these limitations and what their effect on investments might be in a real situation.

These complaints are trivial when compared to the enjoyment and educational value of Baron. All in all, this is one good program which will keep you poised nervously over the keyboard trying to answer nail-biting questions like: "Should I sell that Texas land now for a $10,000 profit, or should I take the risk that the market will rise again next month and double my profit? After all, it has been rising for a long time and could break now...but just think, a $20,000 profit!" You get the idea.

Baron is detailed enough to provide challenge at all levels of play yet is simple to play even if you barely know what real estate is. Of course you can't expect to be able to go out and become a millionaire in real estate investment even if you've become a Baron 42 times! But you can expect to gain a better understanding of real estate investment, plus many hours of fun, and sometimes nerve-wracking entertainment from Baron.

Tycoon

Review By Susan Walker
9138 Mill Creek Drive, #3
Kalamazoo, MI 49009

Tycoon is a complex and challenging commodities simulation which is thoughtfully presented – so that even the most inexperienced player can come to understand and enjoy the world of commodities trading.

The goal in Tycoon is to make a million dollars by speculating in the exciting commodity futures market. One game session lasts 52 weeks. Each week you review newspaper headlines and price changes, research commodities and buy or sell commodities. You start out as a Novice with $10,000 cash. As a Novice you can take long positions and trade four types of commodities which may be purchased with a 10% margin. When your net worth rises, your status and investment alternatives are increased. At $15,000 you become an Investor – allowed to trade four more types of commodities with a decreased margin of 8%. At $30,000 you become a Speculator – able to take short positions. At $90,000 you are a Professional – now trading in a total of twelve commodities and with only a 6% margin. At $250,000 you are a Broker with 15 commodities to play with at a 4% margin. At $1 million you become a Tycoon. When you reach Tycoon status you are given $50,000 to start investing for your next million. The object this time is to see how quickly you can make that second million.

The best way to approach Tycoon is through the disk first and through the short manual second. To start, read only the first few pages of the manual to briefly review terms and situations. The manual will then wisely tell you that the only way to learn Tycoon is to sit down and play it. Get right into the program, which includes a demo that clearly explains the Point & Figure, Scatter Plot, Historical Bar Chart, Moving Average, Oscillator and Commodity Index graphs. It also explains long and short positions, terms and your options. The disk also contains a "saved" partially played game in which the player has $250,000 net worth. You can play with this game, which is outlined in the manual, to get a feel for how it all works before you start your own session with Tycoon.

A chronic "First Read The Directions" person, I made the mistake of reading the whole manual before I looked at the disk. This left me rather confused and feeling like I would never even understand the difference between a long and short position. After viewing the demo and the saved game, and playing one session on my own, I went back and read the manual again -more carefully. At this point I had enough experience with Tycoon to comprehend and benefit from the information presented in the manual.

In fact, my only complaint with Tycoon is that the manual left me confused. The manual is thorough and very helpful after you have worked with the program a little – but the first time user might be a bit apprehensive about playing Tycoon after reading the manual. Of course, you can solve this problem by not taking the manual too seriously until you have worked with Tycoon a while.

Tycoon is not complicated to learn if you aren't initially scared off by the jargon and technical information. When I first looked at Tycoon, I was somewhat overwhelmed by the range of graphs and information available on the 15 commodities in which the player speculates. Then I realized that as a novice I had only four commodities in which to speculate and could take only long positions. This simplified the game enough so that I could successfully play it while learning how to analyze the information and use it to make commodities price change predictions. About the same time I mastered these skills, I naturally progressed to more intricate, demanding levels.

It is exactly the complexity of the simulation which makes it extremely educational and enjoyable. You can devote all your brain cells to this game. You can also get so hooked that you end up devoting more of your time to it than you had planned. Oh well, that means it's a good game – right?! Each session is composed before you begin, so each one is different. You can play Tycoon forever. At the end of the 52 week session your holdings are converted to cash and your net worth and status automatically saved so that you can start a new game with the same status and cash. Another nice feature of Tycoon is that two or three people can pool their gray matter and play it together. This can lead to some interesting arguments and interpretations of the news reports and graphs. As a game, Tycoon is stimulating, involving, and fun. As an educational tool, Tycoon does a good job of introducing the commodities market and how to speculate in it. You can hardly go wrong with Tycoon if you are looking for an adult game which will teach you more than how to shoot down the martians.

Squire

Review By Miles K. Hoffman
Market Research Analyst
Heath Company

In "Millionaire" you play the stock market, in "Baron" you wheel and deal in real estate, and in "Tycoon" you stand in the pit of the commodities market. SQUIRE puts all these together in a game of life – at least the part of life that revolves around investments, financial planning, and personal goal setting.

SQUIRE is an easy game to get into; the "Financial Planning Primer" (Chapter One of the clearly written 76 page manual) gives a good overview of investment tools, practices, and concepts. Following the loading instructions copies COMMAND.COM from your DOS disk to the play disk. Type "START" and you are ready to begin.

The object of SQUIRE is to retire with enough money in the bank to live in comfort. There are two ways to play: The standard or default game and the "reality" mode. The first is not a simplified demonstration. It is a sophisticated financial simulation. You start with a pool of $70,500 cash and a long range goal of $1 million net worth by retirement time. The investment choices of this menu driven simulation include a money market account and an IRA. But these familiar choices are not likely by themselves to make you rich. So we must strike out and bravely invest in corporate bonds, rare coins, stamps, oriental rugs, porcelain, soybeans, gold, California and New York residential and commercial real estate, an oil and gas limited partnership, and even a herd of cattle to fatten for slaughter. The manual discusses each option in detail and exploratory screens guide you as you buy and sell.

There are also stocks, both the growth companies DEC, Federal Express, and the "Blue Chips", Commonwealth Edison and a Big Blue computer company that manufactures Heath/Zenith compatibles. I bought many shares of the computer company and watched the price go down as the months of the simulation went by.

Monthly graphs depicting the nation's economic performance and the performance of your investments, news in the financial journal, and price changes of all the possible investments provide the raw material for creative financial manipulation. But the news is not always a sure guide: The announcement that "Experts see higher gold prices" led me to by gold at $519 an ounce, right before it began its fall (it is $472.80 at this writing and still going down). If the downward spiral in several investments accelerates, you may fall into a negative income situation and be forced to liquidate some investments. And you can go bankrupt!

Taxes are also part of the play, and a wise investor is led by the simulation to consider the tax implications of his or her actions. Every "year" of the game you must pay Uncle Sam. As you advance to the next month, you pass the equivalent of "Go" and add $500 to your disposable income.

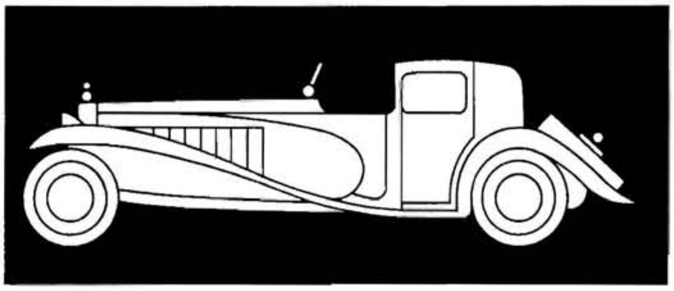
Unexpected events may befall you, of course, as in real life: Your son, the football star, suffers cracked ribs and a separated shoulder, and you must pay the insurance deductible. Your credit cards are stolen and you are liable for $50 on each card. Not so bad, except you had 57! Never a dull moment!

SQUIRE allows you to save your game and return to it at a later time. If you do begin again, SQUIRE assures that it will not be the same game again. It creates a new "economic environment" for your investment action.

When you end a session and fail, as I did, to accumulate a substantial net worth, SQUIRE informs you of your dismal performance and warns that the miniscule retirement income secured will ensure that "your ending days will be spent leaching off relatives".

Much good fun, a real challenge, and a good way to learn something about investments.

But you say retirement planning isn't the way you want to spend a long evening by the fire? You say you prefer to plan investment strategies with thoughts of exotic places, exciting races and foxy women? Well, SQUIRE'S "reality mode" is for you! In this game you enter your age and real assets, you define your retirement financial goals plus a set of interim goals. Then it's off to buy and sell, and your progress depends on your investing skill and luck.



So there you have it! Four ways to make your fortune, well almost. Please understand that the opinions expressed by each reviewer are their own, and none as yet have obtained the personal financial status of Millionaire let alone Squire . . .

Two new products are under development by Blue Chip. They are "A Teacher's Guide to our Economic Simulations" and "American Dream". We received an advanced copy of the Teacher's Guide here at the REMark office. The purpose of the Guide is to bring all the educational elements of the above economic simulations together in a teaching environment. Chapters progressively cover the students passage through the various levels of the simulations. A review test is provided at the end of each chapter plus a certificate suitable for framing is enclosed that must be completed by the teacher. The Teacher's Guide is available now, but no price was quoted. American Dream is scheduled for release in September, and again no price was quoted.

Vendor:	Blue Chip Software, Inc. 6744 Eton Avenue Canoga Park, CA 91303 (818) 346-0730
Price:	All Simulations; MS-DOS Versions $49.95
Media:	5" Disk, Padded Vinyl Manual
Machines:	H/Z 150 PC family with MS-DOS operating system.
Available:	Heath/Zenith Electronic Centers Blue Chip Software, Inc. Zenith Data Systems Dealers

*

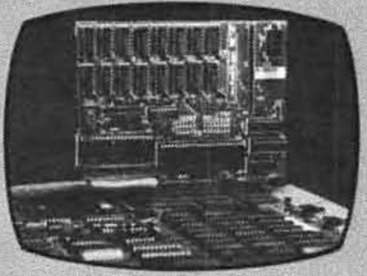
Controlled Data Recording Systems Inc.

Quality Products and Support for the Heath/Zenith Community

New for the H/Z89-90 Computer Users:

SUPER RAM 89

Heath/Zenith 89 and 90 computer users can now get the speed and power of a high capacity Ram Drive System at a reasonable price.



Using the Super 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.

Super 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.

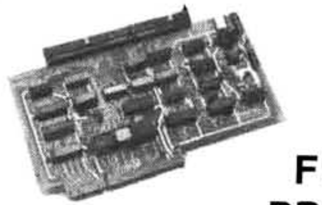
The Ram Drive Software (SRAM) allows one or two logical ram drives. The ram drive(s) can be located starting anywhere from logical A: to O: (standard drives get relocated). SRAM can be set to start at logical A: and warm boot with ram (no floppy disk accesses needed). Ram drive attatches to any of the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

Super Ram 89 Pricing:

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

FOR THE H/Z89-90 COMPUTERS

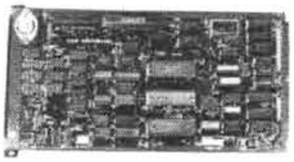
THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE:$395



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

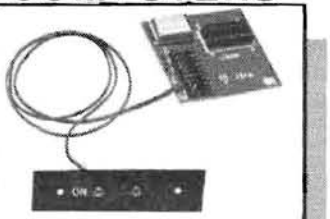
Allows owners of the FDC-880H or the FDC-H8 to read/write to the following disktypes:

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.

New: a powerful floppy drive package for any computer that can use 8" floppy drives.

Two extended technology 5.25" drives in a package with interface cables case and power supply, with the capability of looking exactly like two standard 8" drives to the computer.

Now have **2.4 megabytes** of **5.25" floppy file storage capacity** in one package. A switch enables the drives to look like 8" drives or 5.25" 96 TPI drives.

ASK FOR PRICE.

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.

Graphics Tricks For The H/Z-150

Pat Swayne
HUG Software Engineer

In attempting to convert some H/Z-100 programs to run on the H/Z-150 (and other PC-type computers), I have been playing around with the graphics modes of the '150, and have discovered a few tricks that can be used to extend its graphics capability. Since most of you probably program in BASIC, I have adapted my tricks to GW-BASIC, and have presented example programs or routines in this article.

Caution: Be sure to enter the example programs from this article exactly as printed. Be especially careful with the DATA statements.

Four Color Palettes

When you use the medium resolution graphics mode in GW-BASIC, you are allowed to have 4 colors on the screen at one time. These colors consist of a background color and a "palette" of three other colors. In GW-BASIC, there are only two palettes to choose from. However, there is a program called PC PALETTE available from IBM that allows you to choose from 4 palettes when you operate it in the medium resolution mode. I found that you can use the same 4 palettes in GW-BASIC if you write directly to a port called the "color select register" instead of using the COLOR command. The color select register is at port address 3D9 (hex), and its bits are "officially" defined as follows:

Bit	Definition
7	Unused
6	Unused
5	Palette select
4	Unused *
3	Intensity bit
2	Red
1	Green
0	Blue

Bits 0 through 3 are used to select the background color. For example, for a light magenta background, you would select bits 0, 2, and 3. The value of the binary number formed by these bits is 13 in decimal, which is the number that selects light magenta in the COLOR command. The palette select bit selects the green, red, and yellow palette if it is zero, and the cyan, magenta, and white palette if it is one. The bit marked with an asterisk is actually not unused, but it is another intensity bit, which selects the intensity of the palette colors. If you consider bits 4 and 5 as a separate 2-digit binary number, you can define the available palettes as shown below:

Palette no.	Colors
0	Green, red, brown (dark yellow)
1	Light green, light red, yellow
2	Cyan, magenta, white
3	Light cyan, light magenta, intense white

The GW-BASIC COLOR command uses 1 and 3 from this list as its two palettes (contrary to what the manual shows). If you take the number of the palette you desire from the list and multiply it by 16, and then add in the number of the background color you want, you can OUTput the result to the color select register in GW-BASIC to select the colors. The following example illustrates this.

```
10 SCREEN 1:'  SELECT MEDIUM RESOLUTION MODE
20 PALETTE%=2:BACKGROUND%=4
30 GOSUB 1000:'  SELECT COLORS
1000 OUT &H3D9,PALETTE%*16+BACKGROUND%
```

If you plan to change colors more than once in a program, you may want to use a subroutine as shown above. If you are only going to set the colors once, you could use:

```
10 SCREEN 1:OUT &H3D9,2*16+4
```

Of course, if the palette you want is palette 0, you only need to write the background color to the color select register to select it.

High Resolution Color

The color select register is also active when you use the high resolution monochrome graphics mode. In this mode, the lower 4 bits select the color of the pixels on the screen. The background color remains black. The default color in the high resolution mode is intense white, which means that if you have your monitor screen adjusted for the normal white which is the default in the text mode, programs that use the high resolution mode may be a bit rough on the eyes, forcing you to re-adjust your monitor. You can avoid this re-adjustment by selecting normal white via the color select register immediately after setting the high resolution mode, as follows:

```
1Ø SCREEN 2:OUT &H3D9,7:' SELECT NORMAL WHITE
```

You can also select any single color, by outputting its number to the color select port. The high resolution mode is still monochrome (one color), but at least you can use another color besides white.

Colorful Words

The trick of changing the color in the high resolution mode can be used in other programs besides BASIC. For example, Microsoft Word (tm) uses the high resolution mode, and you will experience the same intensity problem with it as with BASIC. Using the DEBUG program supplied with MS-DOS, you can patch Word to use normal white, or a color other than white. Here are the commands you would enter to DEBUG, assuming that DEBUG.COM is on drive A:, and WORD.COM is on drive B:.

```
A>DEBUG B:WORD.COM
-U21EC
xxxx:21EC AØ4BØE        MOV     AL,[ØE4B]
xxxx:21EF 32E4          XOR     AH,AH
xxxx:21F1 CD1Ø          INT     1Ø
xxxx:21F3 8Ø3E4BØEØ6    CMP     BYTE PTR [ØE4B],Ø6
xxxx:21F8 7418          JZ      2212
```

Use the U command, as above, to make sure that your version of Word is the same as mine (1.10). There will be more lines listed than what we have shown, but the first 5 lines should look like this. Instead of xxxx, there will be numbers in your listing, which will be different in different situations. If your code looks like what is shown here, use the A command to enter the patch:

```
-A21F3
xxxx:21F3  MOV     DX,3D9
xxxx:21F6  MOV     AL,7
xxxx:21F8  OUT     DX,AL
xxxx:21F9  JMP     2212
-NB:WORD1.COM
-W
```

The N command will rename the patched program to WORD1.COM, and the W command writes the patched program to the disk. You can leave off the N command if you want your patched version to be called WORD.COM, but make sure you have another copy of the original WORD.COM on another disk. When patched as in this example, Word will use normal white instead of intense white. If you would like to use a color, use the number of the color (in hex) in the MOV AL, instruction. For example, if you wanted dark yellow (to simulate an amber monitor on a color screen), you would use MOV AL,6.

BASIC Text Colors

When you operate GW-BASIC in the medium resolution mode, the color used for text (as in PRINT statements) is always color number 3 from the selected palette. This color can be changed by poking a new value to an address that Microsoft was thoughtful enough to place at a fixed location in the various versions of GW-BASIC and PC BASIC. The address is 4E (hex), and you can just POKE the number of the color you want for text to it, as in

```
1Ø POKE &H4E,1:' SET TEXT TO COLOR 1
```

The color takes effect on any text written after the POKE, but does not change text written before. You can therefore have up to three different colors of text on the screen at once. You should not use the background color (palette color 0) for text, because it will only produce invisible text. In fact, if you happen to set the text to color 0 while BASIC is in the command mode (not running a program), it will no longer accept any commands, and you will have to reset your computer to get out. You might think that you could set the text to color 0 and write over an area painted with another color and see, for example, black text on a white background (if color 0 is black and the painted color is white). However, it does not work that way, because the 8-by-8 pixel cell in which the text character is placed is always blanked to the background color, with the pixels of the text character set to the selected text color. If both the background and text color are 0, each character will just produce an 8-by-8 pixel block in the background color.

Reverse Video Text

Since setting the text color to 0 cannot produce "reverse video" text, you may be wondering if there is a way to do it. There is, but it is a little complicated. When you operate your computer in the text mode, the characters are formed from information in a ROM that is on the video card. When you use one of the graphics modes, the characters are formed from a table that is in the MFM-150 ROM on the CPU card. Only the first 128 (of 256) characters are defined in the table, and a means is provided for you to define the other 128 characters. Since the MFM-150 ROM is accessible from programs, you can copy the character table, and alter it to change the characters. If you make your own table consisting of the complement of each byte in the original table, it will produce reverse video text. The following program illustrates how this can be done in GW-BASIC.

```
1Ø ' PROGRAM TO DEMONSTRATE REVERSE VIDEO
2Ø ' IN THE GRAPHICS MODE
3Ø '
4Ø CLEAR ,65535-1Ø24:DIM CODE%(2Ø)
5Ø FOR I%=Ø TO 2Ø:READ CODE%(I%):NEXT I%
6Ø DEF USR=VARPTR(CODE%(Ø))
7Ø CODE%(Ø)=USR(Ø)
8Ø REVMSG$="REVERSE VIDEO"
9Ø SCREEN 1:OUT &H3D9,Ø
1ØØ CLS:LINE (Ø,1ØØ)-(319,191),3,BF
11Ø LOCATE 15,3
12Ø PRINT " THIS IS A ";
13Ø FOR I=1 TO LEN(REVMSG$)
14Ø A$=MID$(REVMSG$,I,1):A=ASC(A$):PRINT CHR$(A+128);
15Ø NEXT I
16Ø PRINT " MESSAGE.":LOCATE 11,3
17Ø PRINT " THIS IS A ";
18Ø FOR I=1 TO LEN(REVMSG$)
19Ø A$=MID$(REVMSG$,I,1):A=ASC(A$):PRINT CHR$(A+128);
2ØØ NEXT I
21Ø PRINT " MESSAGE.":LOCATE 1,1
22Ø A$=INPUT$(1):SCREEN Ø:WIDTH 8Ø
23Ø DATA &H61E,&H71E,&HB8,&H8EFØ
24Ø DATA &HBED8,&HFA6E,&HBF,&HB9FC
25Ø DATA &H4ØØ,&HACFC,&HDØF6,&HE2AA
26Ø DATA &H31FA,&H8ECØ,&HBED8,&H7C
27Ø DATA &H4C7,&HFCØØ,&H448C,&H7Ø2
28Ø DATA &HCB1F
```

When you run this program, it writes two lines on the screen consisting of partly normal and reverse characters. One line is written in a background colored area, and the other is in a painted area, so you can see the effects of both text types in each area. The program pauses after it runs, until you type any key. In line 40 of the program, space is cleared for the table of text characters (the table is 1024 bytes long), and then a machine language routine is called to copy the original table to the new space, with each byte complemented. The actual machine code is contained in the DATA statements at the end of the program, and in assembly language, it looks like this:

```
        PUSH    DS
        PUSH    ES              ;SAVE SEGMENT REGISTERS
        PUSH    DS
        POP     ES              ;ES = DS
        MOV     AX,ØFØØØH
        MOV     DS,AX           ;POINT TO ROM SEGMENT
        MOV     SI,ØFA6EH       ;POINT TO CHAR TABLE
        MOV     DI,ØFCØØH       ;PUT OUR TABLE HERE
        MOV     CX,Ø4ØØH        ;SIZE OF TABLE
        CLD
MVLP:   LODSB                   ;GET A CHARACTER
        NOT     AL              ;COMPLEMENT IT
        STOSB                   ;STORE IN NEW TABLE
        LOOP    MVLP
        XOR     AX,AX
        MOV     DS,AX           ;POINT TO INT. SEGMENT
        MOV     SI,Ø7CH         ;POINT TO TABLE VECTOR
        MOV     WORD PTR [SI],ØFCØØH ;INSERT ADDRESS
        MOV     [SI+Ø2],ES           ; OF OUR NEW TABLE
        POP     ES
        POP     DS              ;RESTORE REGISTERS
        RETF                    ;RETURN TO BASIC
```

To print a character from the new table, take the ASCII value of the character desired and add 128 to it. In the sample program, the text to be printed in reverse video is placed in a string variable in line 80. In lines 140 and 190, the characters from the text are printed by using the MID$ function to extract each character. The ASC function gets the ASCII value of each character, and the CHR$ function is used to print that value plus 128.

Other Special Text Effects

The technique used to create reverse video characters can be used to create other effects. For example, if you copy the original text table from the ROM, but replace the last byte for each character (each character is 8 bytes) with 0FF (hex), it will produced underlined text. The following program illustrates this:

```
1Ø ' PROGRAM TO DEMONSTRATE UNDERLINED TEXT
2Ø ' IN THE GRAPHICS MODE
3Ø '
4Ø CLEAR ,65535-1Ø24:DIM CODE%(23)
5Ø FOR I%=Ø TO 23:READ CODE%(I%):NEXT I%
6Ø DEF USR=VARPTR(CODE%(Ø))
7Ø CODE%(Ø)=USR(Ø)
8Ø UNLMSG$="UNDERLINED TEXT"
9Ø SCREEN 2,,,,1:OUT &H3D9,6:' YELLOW TEXT
1ØØ PRINT:PRINT "THIS IS AN ";
11Ø FOR I=1 TO LEN(UNLMSG$)
12Ø A$=MID$(UNLMSG$,I,1):A=ASC(A$):PRINT CHR$(A+128);
13Ø NEXT I
14Ø PRINT " MESSAGE."
15Ø DATA &H61E,&H71E,&HB8,&H8EFØ
16Ø DATA &HBED8,&HFA6E,&H1BF,&HB9FC
17Ø DATA &H3FF,&HACFC,&HC7F7,&H7
18Ø DATA &H275,&HFFBØ,&HE2AA,&H31F4
19Ø DATA &H8ECØ,&HBED8,&H7C,&H4C7
2ØØ DATA &HFCØ1,&H448C,&H7Ø2,&HCB1F
```

This program produces a new text table and prints using the same methods as the reverse video program. However, it uses the high resolution mode, and sets the color to yellow (line 90). The assembly form of the machine code in the DATA statements looks like this:

```
        PUSH    DS
        PUSH    ES              ;SAVE SEGMENT REGISTERS
        PUSH    DS
        POP     ES              ;ES = DS
        MOV     AX,ØFØØØH
        MOV     DS,AX           ;POINT TO ROM SEGMENT
        MOV     SI,ØFA6EH       ;POINT TO CHAR TABLE
        MOV     DI,ØFCØ1H       ;PUT OUR TABLE HERE
        MOV     CX,Ø3FFH        ;SIZE OF TABLE
        CLD
MVLP:   LODSB                   ;GET A CHARACTER
        TEST    DI,ØØØ7         ;TEST FOR 8TH BYTE
        JNZ     NOT8            ;NOT 8TH
        MOV     AL,ØFFH         ;ELSE, MAKE IT FF
NOT8:   STOSB                   ;STORE IN NEW TABLE
        LOOP    MVLP
        XOR     AX,AX
        MOV     DS,AX           ;POINT TO INT. SEGMENT
        MOV     SI,Ø7CH         ;POINT TO TABLE VECTOR
        MOV     WORD PTR [SI],ØFCØ1H ;INSERT ADDRESS
        MOV     [SI+Ø2],ES           ; OF OUR NEW TABLE
        POP     ES
        POP     DS              ;RESTORE REGISTERS
        RETF                    ;RETURN TO BASIC
```

You may notice that the table created by this program starts one byte higher in memory than the reverse video table. That was done to make it easier to test for the 8th byte of each character. I reduced the size of the table by one byte to compensate, which means that the last character is not complete. The last character is the Delete code, which is unprintable anyway, so no harm is done.

Machine Language Techniques

In developing machine language routines for use within BASIC programs, I use techniques that are not like those illustrated in the GW-BASIC manual. I place the code within an integer array instead of in a string variable as recommended by Microsoft. That makes it easier to obtain the starting address of the routine. With a string variable, the VARPTR function points to the address of the string, not the actual string, but with an integer array, it points to the actual data.

To call the routine, I use the USR function instead of the CALL statement. If you use USR, the program will be compilable without changes, but CALL is interpreted differently by the compiler than by the interpreter.

To enter short routines such as the two above, I use the A command in DEBUG to enter assembly code directly into memory. Start the program at 100 (hex), and when you are finished, use the R command to make the CX register equal to the size of the routine. Then you can give it a name with the N command, and write it to disk with the W command.

You can use the D command in DEBUG to obtain the hex values of the bytes in the routine, and code the DATA statements by hand. If you do, be sure to use two bytes from the routine for each number in the DATA statement, and reverse the order of each pair of bytes. For example, if two bytes appear as AA BB in the D command listing, use &HBBAA in the DATA statement. I used code DATA statements by hand, but since I found myself doing more and more little machine language routines, I developed the following program, which converts a binary file directly to DATA statements.

```
1Ø ' PROGRAM TO MAKE DATA STATEMENTS FROM BINARY FILE
2Ø DEFINT A-Z:DIM B$(1)
3Ø ON ERROR GOTO 27Ø
4Ø PRINT:PRINT "BINARY TO DATA STATEMENT CONVERSION"
5Ø PRINT:LINE INPUT "ENTER THE BINARY FILE NAME: ";FA$
6Ø OPEN "I",1,FA$:CLOSE
7Ø PRINT:LINE INPUT "ENTER THE OUTPUT FILE NAME: ";FB$
8Ø PRINT:LINE INPUT "ENTER THE STARTING LINE NUMBER: ";LN$
```

```
90 LN!=VAL(LN$):IF LN!=0 THEN 80
100 PRINT:LINE INPUT "BYTES PER NUMBER (1 OR 2): ";BPN$
110 BPN=VAL(BPN$):IF BPN<>1 AND BPN<>2 THEN 100
120 OPEN "R",1,FA$,2
130 OPEN "O",2,FB$
140 FOR I=0 TO 1:FIELD 1,I AS D$,1 AS B$(I):NEXT I
150 L=1:CNT=1
160 ' CONVERSION LOOP STARTS HERE
170 GET 1,L
180 IF EOF(1) THEN 250
190 IF CNT=1 THEN LN$=STR$(LN!):LLN=LEN(LN$):
    LN$=RIGHT$(LN$,LLN-1):PRINT #2,LN$;" DATA &H";:
    ELSE PRINT #2,",&H";
200 A!=ASC(B$(0)):B!=ASC(B$(1))
210 IF BPN=1 THEN PRINT #2,HEX$(A!);",&H";HEX$(B!);:
    ELSE A!=B!*256+A!:PRINT #2,HEX$(A!);
220 CNT=CNT+1:IF CNT=5 THEN PRINT #2,:CNT=1:LN!=LN!+10
230 L=L+1
240 GOTO 170
250 IF CNT<>1 THEN PRINT #2,
260 CLOSE:END
270 IF ERR=53 AND ERL=60 THEN PRINT CHR$(7)
280 RESUME 50
```

This program will convert either a file saved by DEBUG or a .COM file to DATA statements. You have the choice of either 1-byte or 2-byte numbers in the DATA statements. Use 2-byte numbers for routines that will be placed into integer arrays and called from BASIC, and use 1-byte number for routines that will be written by BASIC as .COM files (as in the article "Recovering from the Protected Mode in ZBASIC or GW-BASIC"). This program should be run using GW-BASIC version 2 if possible, because it may create extra lines of data statements if run under version 1 (or ZBASIC). Version 1 reads files in multiples of 128 bytes, and will not indicate the end of a file except on a 128-byte boundary. If you must use version 1, remove the extra DATA statements (the numbers in them will all be &H0). If you use GW-BASIC version 2, you should make sure that your routine is an even number of bytes long (add a dummy byte if not) because the conversion program reads your binary file 2 bytes at a time.

The output of the conversion program can be MERGEd directly into a BASIC program you are developing. It will eliminate the possibility of making a typing error, which could occur when you code DATA statements by hand.

*

Get *more news* about your Heath or Zenith computer system *more often*—AND get a *FREE* directory of over 400 sources of support for your computer.

Subscribe to **Buss: The Independent Newsletter of Heath Co. Computers**

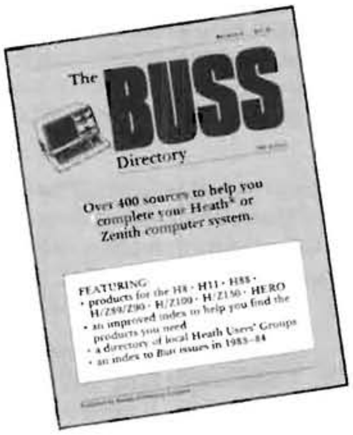
and also receive the directory *at no additional cost*

You can't get more frequent news about your system than with a subscription to *Buss*. Twenty times a year, in each and every eight-page issue, you'll find out what other users and independent suppliers have to say about:

- □ H8
- □ H/Z89
- □ H/Z100
- □ H/Z150
- □ compatible hardware
- □ printers
- □ terminals
- □ disks and drives
- □ compatible software
- □ business applications
- □ home/hobby applications
- □ games
- □ graphics
- □ utilities
- □ Z-DOS
- □ MS-DOS
- □ CP/M

Twenty times a year, you'll get a complete report on the latest news about your Heath/Zenith system—delivered directly to you by first class mail. *Buss* helps you:

- Learn what new products are being developed, and where you can get them.
- Benefit from the advice of users when you write in to *Buss*'s "Requests for Assistance" column.
- Save money when you add to your system! *Buss* offers a free "For Sale" listing for Heath/Zenith users where you can pick up a bargain.
- Avoid bugs in your system as you read how other users have solved similar problems with their computers.
- Discover news and events in the Heath/Zenith community that affect you.
- Meet other users in your area through the "Local Club Notes" section of *Buss*.
- Get unbiased coverage of your system from the oldest independent newsletter for Heath/Zenith users. Since 1977, *Buss* has been committed to supplying information exclusively for Heath/Zenith users.


The BUSS Directory
Over 400 sources to help you complete your Heath® or Zenith computer system.
FEATURING:
• products for the H8 · H11 · H88 · H/Z89/Z90 · H/Z100 · H/Z150 · HERO
• an improved index to help you find the products you need
• a directory of local Heath Users' Groups
• an index to Buss issues in 1983–84

And, as a free bonus with your *Buss* subscription, you'll receive your FREE copy of the 1985 edition of *The Buss Directory*—a 92-page reference tool which provides you with information on over 400 sources of independent support for your Heath/Zenith system.

The Buss Directory is a $12.50 value—and it's yours FREE with your subscription to *Buss!* We'll send it to you by first class mail as soon as we receive your subscription order.

Order your subscription today, and start getting MORE NEWS, MORE OFTEN about your Heath/Zenith system!

Call Toll Free: 800/341-1522

(DATATEL 800™, for orders only, Monday-Friday 8 a.m.–9 p.m., and Saturday 9 a.m.–5 p.m., Eastern Time)

Or use this coupon, and mail your order to:
Buss, Dept. R85
716 E Street, S.E.
Washington, D.C. 20003
202/544-0900

Your satisfaction with *Buss* is completely guaranteed. If at any time you are not satisfied, your money will be refunded—in full.

Yes! Sign me up for *Buss*, and send me my free copy of *The Buss Directory*!

I want:
□ 30 issues for $38 ($55 overseas)
□ 20 issues for $28 ($40 overseas)
□ 10 issues for $18 ($24 overseas)

Name____________________
Address ____________________
____
____

□ Payment enclosed (checks must be in U.S. dollars payable on a U.S. bank).
□ Bill me.
□ Charge my: □ Visa
□ MasterCard
Card # ________________
Expiration date________
(MC code #________)

Buss, Dept.R85, 716 E Street S.E., Washington DC 20003

Serial Interfacing On The Z-100 PC

Rick Housh
5811 West 85th Terrace
Overland Park, KS 66207

Some of us, perhaps regretfully, have found it necessary or desirable to switch from our reliable and beloved H/Z-89's or H-8's to an IBM compatible format for one reason or another. In our case, my wife is a writer, and our computer is used extensively as a word processor.

The availability of more powerful word processing programs in the MS-DOS format made it difficult to resist, especially when Heath began bundling Microsoft Word with the H/Z-100 PC series computers. With the HUG discount, the system is almost irresistible, and the H/Z-150's and 160's, of course, are the best of the IBM compatible machines.

There are problems, though, in converting from one of the eight bit machines to the new H/Z-100 PC series. Many new owners simply buy the new machine, a new parallel printer, connect them with the appropriate cable, and use them. If you are a former '89 or H8 owner, however, accustomed only to serial ports (and plenty of them), you may have a problem.

MS-DOS, as implemented on the H/Z-100 PC series of computers, is capable of supporting two serial ports of the RS-232C variety, wired as data terminal equipment (DTE). The RS-232 is simply a convention used to assure (allegedly) that the proper signals appear on the proper pins of each port, regardless of the brand of machine to which they are attached. Thus a peripheral device with a standard RS-232 port should operate correctly when connected to another device with such a port with no concern for compatibility. This is certainly true to a degree, but interfacing through these ports requires some thought and under standing. This article will attempt to explain some of the problems which may arise and, through examples, explain how to deal with them.

Enabling The Second Port

As currently supplied, although the sockets and most of the components for two serial RS-232 ports are present on the floppy disk controller board on H/Z-100 PC machines, the board only has one port enabled, which is called COM1 in MS-DOS. Earlier versions of these computers came with both ports enabled (the second is recognized by the operating system as COM2). Apparently this caused a problem when users added accessory boards, most of which contain a second port, as suddenly there were three ports, and the software was only capable of supporting two. The solution was for Heath to disable COM2.

If you need the second port and do not plan to add an accessory board it is fairly simple to restore the use of COM2 (the lower outlet on the back of the floppy disk controller board) with some small circuit changes and a little work. In fact, Heath will soon have available a kit to accomplish this change, but the parts are available now from Heath (or other sources), and the procedure is simple.

First, remove the floppy disk controller board from the computer. Handle it carefully. Examine the component side of the board near U532 (the component designations and pin numbers are clearly printed on the board). You will probably see a small gouge or cut in one of the foil traces leading away from U532. This cut is how the second port was disconnected during assembly, effectively severing the connection between pin 17 of U532 and pin 19 of U534. If you now look on the reverse (solder) side of the board you will see a jumper wire soldered between pin 14 of U532 and pin 19 of U534. The combination of these factory modifications was to change the original connection between pin 19 of U534 and pin 17 of U532 to one between pin 19 of U534 and pin 14 of U532.

It is only necessary to restore the original connection and supply any missing parts to use COM2. The original connection is restored by unsoldering one end of the jumper wire from pin 14 of U532 and resoldering that same end to pin 17 of U532. This restores the cut trace.

Next, the missing parts must be replaced. Generally only the integrated circuits U506, U535, and U536, and possibly their sockets, are missing. Check the sockets. If any are missing, solder new ones in place. U506 uses a standard 40 pin IC socket. U535 and U536 use 24 pin sockets. Buy low profile sockets.

Purchase and install the necessary ICs at U506, U535, and U536. U506 is a standard 8250 ACE (Heath PN 443-952), U535 a standard 75189 or 1489 EIA receiver (Heath PN 443-795), and U536 a standard 75188 or 1488 EIA driver (Heath PN 443-794). These are the same parts used on the H/89 serial ports. Make sure the IC's are positioned correctly in their sockets.

Locate J502 on the component side of the board. Install a jumper between IR3 (the connecting point for interrupt request 3) on J502B and the opposite hole on J502A. The factory board uses a pin strip and removable jumper at this position, but as you will probably never change it, there should be no problem with soldering in a short wire jumper.

The installation of COM2 is now complete. There are no switches to reset, and the operating system software should now see a COM2. Reinstall the board and run CONFIGUR to confirm this, and while at it configure the port however you wish.

Making A DCE From A DTE

RS-232C serial ports come in two different varieties, DTE's and DCE's. These abbreviations stand for "data terminal equipment" and "data communication equipment," respectively. Both ports on the H/Z-100 PC series are configured as DTE's. On the other hand, the '89 had one DTE port and one or two DCE ports.

A DTE port is designed to communicate with a DCE and vice versa. In simple terms the input and output pins are reversed between them. Except in special cases, of the 25 pins on the standard RS-232 port, pins 1 through 8 and pin 20, and sometimes 22, are the only ones used. They are called the same on each, as follows.

1. Protective Ground
2. Sout or TX data (Outpuւ on DTE, input on DCE)
3. Sin or RX data (Input on DTE, output on DCE)
4. RTS–Ready to send (Output on DTE, input on DCE)
5. CTS–Clear to send (Input on DTE, output on DCE)
6. DSR–Data set ready (Input on DTE, output on DCE)
7. Signal ground
8. RLSD or CD–Received line signal or carrier detect (Input on DTE, output on DCE)
20. DTR–Data Transmitter Ready (Output on DTE, input on DCE)
22. Ring detector (DTE only, input)

It is the calling of what is an input on one connector by the same name as the output on another which causes much of the confusion that exists with respect to RS-232 ports. On the other hand if you are connecting a DTE, such as a computer used as a terminal, to a DCE such as a modem, this convention allows you to connect the devices with a straight cable with no complications. There is a pin-for-pin functional correspondence. The problem comes when we need to connect one DTE to another DTE (such as a computer to a printer).

Even so, the problem should be easily solved, simply by reversing the corresponding pin connections on one end of the cable. Yes, but what are the corresponding pin connections to allow one DTE to talk to another? Well, first eliminate the ones which carry no signal, and are common to each port. These are the ground connections, pins 1 and 7. They go directly to the other end of the cable, to the same pins. This leaves pins 2, 3, 4, 5, 6, 8, and 20. I have omitted 22 as it is used only for special purposes beyond the scope of this article.

The remaining pins fall into two categories; 1) those carrying data signals, pins 2 and 3; and 2) those carrying control signals, pins 4, 5, 6, 8, and 20.

At this point it is necessary to discuss "handshaking." Handshaking is the means by which one of the connected devices, the sender, tells the other it wants to send data and the receiver tells the sender whether it is prepared to receive it. If you are sending data only one way and you know the receiver is always prepared to receive (i.e. will never be busy) when the sender wants to send, you can usually omit handshaking. Here you may use a simple two-wire cable. With two DTE's connect pin 7 at both ends of the cable, pin 2 at the sender to pin 3 at the receiver, and let it go at that. Otherwise you need handshaking, and must provide for it.

There are two forms of handshaking, 1) "software" and 2) "hardware". The names are slight misnomers as both use hardware and software to accomplish their purpose. The difference is that software handshaking uses special codes on the data lines mixed in with the data signals to ask and tell of busy or ready states while hardware handshaking uses one of the control lines to send a separate signal to achieve the same purpose.

Software handshaking is commonly one of two forms; 1) DC1/DC3 (x/on-x/off) or 2) ETX/ACK. These are simply the standard names of the ASCII characters that are sent over the data signal lines to signal busy and ready. DC1/DC3 are the Control-S and Control-Q characters so familiar to CP/M users and ETX/ACK are Control-C, Control-F. If you use one of these forms of handshaking you can get by with a three wire cable even for two-way communication. Connect pin 7 of one plug to pin 7 of the other, pin 3 of the first to pin 2 of the second, and pin 2 of the first to pin 3 of the second. You are using only the data lines for signals. This type of handshaking will work in almost all cases. However, at very high baud rates, and depending upon the processing speed of the hardware-software combination, the software form of handshaking may not be fast enough, and some data or characters may be lost in transmission. Examine printouts carefully for this problem, if what you are interfacing is a printer, for example. What look like misspelled words on paper may in fact be data loss in transmission.

What if you need, or want, hardware handshaking? Well, you already know to connect the grounds together and cross pins 2 and 3, the data lines. Then, to implement hardware handshaking (assuming your equipment is programmed for it) you need only know which of the remaining control lines (input and output) correspond and therefore, how they must be connected.

There are 5 lines unaccounted for, as yet. On a DTE they are RTS (pin 4), CTS (pin 5), DTR (pin 20), DSR (pin 6), and RLSD (pin 8). The answer to the question of which goes to which is: RTS to CTS (4-5), CTS to RTS (5-4), DTR to DSR (20-6) and DSR to DTR (6-20). What of pin 8? Good question. On most DCE's (such as the '89's, for example) pin 8 is an output. There is no corresponding pin on a DTE, and therefore it is not possible to make a fully operational DCE from a DTE. In fact Heath sells two "null modem" cables, as those are called which make all the above cross-connections for you, the HCA-52, six feet long, and the HCA-100-PC, twelve inches, which are pre-wired and reverse all the necessary lines, but the HCA-100-PC does not connect pin 8 at all, and the HCA-52 carries pin 8 on one DTE directly to pin 8 on the other. These are both inputs on DTE's and of course do nothing when wired together. This is no problem in most cases, as very few devices use the signal on pin 8 for any purpose, but some do. It took me a very long time to discover that our Brother HR-15 printer required a high signal to its own pin 8 to accept data. Of course with either of the above null modem cables it was getting no signal, and thus would not operate, although it had worked flawlessly on the '89's DCE, which provides a high signal to pin 8 on the printer when the computer wants to send.

There is of course, a way around this lack of a signal to pin 8. If you are using DTR-DSR handshaking use a jumper between pins 4 and 5 on the first end of the cable and connect them to 8 on the second end. Then connect 4 and 5 on the second end to 8 on the first end of the cable. This will disable RTS while providing the necessary signal to RLSD on pin 8, and still allow DTR to handshake. If you are using RTS–CTS handshaking, place the jumper between pins 6 and 20 on one end of the cable and connect that line to pin 8 on the other end. Connect pin 8 on the second to pins 6 and 20 on the first. This will disable DTR and allow RTS handshaking.

The above suggestions may need to be modified in certain cases. Printers, especially, do not necessarily have pin connections which correspond exactly with the standard. In many cases the pins may have the proper names assigned in the documentation, while serving a slightly different function than intended by the standard. As an example, many printers have a signal on pin 11 called SCA, which is in reality a reverse RTS signal; that is, when the printer needs to send a busy signal to the computer the normally high signal to the computer on printer pin 11 goes low. Although this can certainly be connected to pin 5 on the computer and used to configure your communication protocol to "low RTS", it should be thought of not as RTS, but simply as a signal which goes low when the printer is busy. Thus, if this signal is available, you may connect pin 11 of the printer to pin 6 of the computer and configure your system to "low DTR." By the same token, you could connect the DTR (pin 20) at your printer to pin 5 at the computer and configure for "high RTS." The point is that the computer will not know what the signal is called, and the only thing that matters is that the signal that it sees is the right polarity at the right time.

So, if you have made the switch from the H/Z89 or the H8 to the new H/Z 100 PC, don't despair. Your serial peripherals will work, and you can have almost as many ports. With a pinout diagram of your printer or other peripheral and a good chart of the functions of the signals of those pins at their busy and ready states you can make almost anything talk to anything. The trick is to know how to make or modify a cable to put the right signal in the right place at the right time. ✱

Break The RAM Barrier & Turn Your H-89 Into A 448K Monster!

Part Two

Interactive Graphics Controller From SigmaSoft And Systems

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

In Part I of this two part new products evaluation, I discussed the FBE Research Co.'s SPOOLDISK 89, a versatile 128k RAM printer buffer that plugs into your H-89. The reason they share billing is because SPOOLDISK 89 and the Interactive Graphics Controller are able to coexist in the H-89 without creating I/O conflicts. That is, if you've loaded SPOOLDISK 89 with a series of documents for printing, you simply RESET the computer and BOOT up a system disk containing the Interactive Graphics software without interrupting the data flow to the printer.

By the same token, if you've created some graphics designs and want to put them on hold, you can (under HDOS) store them temporarily in the PSEUDO DISK portion of the IGC by resetting the computer and booting up with yet another system disk — all this while SPOOLDISK 89 feeds your printer. This creates some interesting applications I'll discuss later in this article.

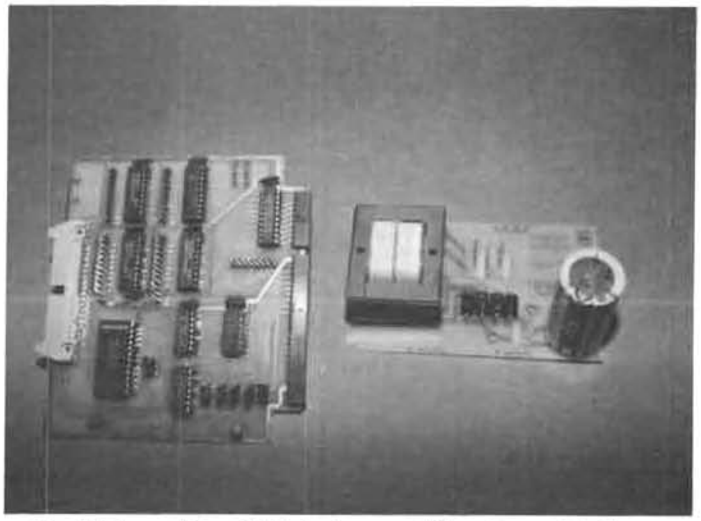
Graphics enhancements for the H-89 have been around for several years in both Color and Monochrome versions. In the former category, the H-89-3 from New Orleans General Data Systems was the forerunner — an improvement over the original color card they had produced for the old Heath H-8 computer, by adding speech synthesis through a Votrax SC-O1A.

Then came "The Entertainer", from Sandia Computer Products. It sported the same color, sound and sprite manipulation as the H-89-3, but it did not appear to have a wide distribution, even though I've been told that it is easier to use because the documentation contained programming examples to get the user started. If I'm ever able to scare up an evaluation unit of "The Entertainer", I'll resurrect my H-89-3 and do an earnest comparison of the two.

In the Monochrome department, Cleveland Codonics and Northwest Digital have excellent high-resolution graphics boards that boast pixel resolution of 504 x 247 and 512 x 250, respectively. The Cleveland Codonics unit also has a Tektronix emulator available for even higher resolution. I almost purchased "The Imaginator" board from Cleveland Codonics about a year ago, but I hesitated for several reasons. I couldn't locate any ready-to-run software and because the maximum memory available for screen display was only 16k. These limitations hold true for the other three boards as well.

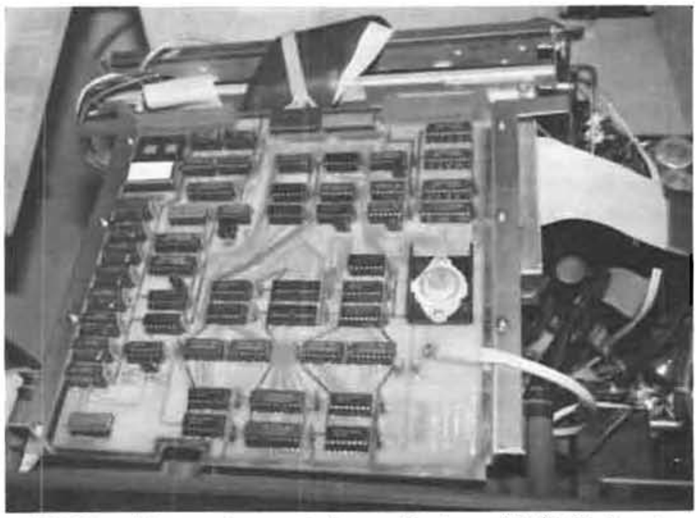
When SigmaSoft and Systems recently announced their new Interactive Graphics Controller, I began to have second thoughts about adding yet another graphics card to my system. I took the cautious route and ordered their User's Manual.

The primary features were rather intriguing. The user could install anywhere from 64k-256k RAM for graphics programming or as a "pseudo disk." Normal high-resolution graphics were 640 x 250 pixels, or 640 x 500 through the Interlaced Mode. Three pages of graphics were available in either Bank or Interlaced Mode. In addition, through the Universal Parallel Interface that comes with the IGC, the user could add Atari/Wico compatible Joysticks or Trackball Controllers, a Light Pen, and up to six separate printers (by installing 3 Universal Parallel Interface cards).



The Universal Parallel Interface and Supplemental Power Supply.

Assorted Hardware & Software required for installation.



IGC Graphics card mounted over the top of H/Z-89. Front rests on CRT and swings up and out of the way for easy access to computer.

But the feature that impressed me the most was the simplicity of Graphics Operating System, because it would allow me to create complex designs with a minimum of fuss. The IGC Command Set is quite specific: Dot, Line, Box, Circle, Arc, Paint, Style, Draw, and about 15 other commands permit the utmost freedom to design charts, graphs, scientific and engineering drawings, animated graphics and artistic designs, from almost any high-level programming language — including a simple Text Editor like PIE — by defining the Cartesian coordinates with "x" and "y" placement. Any solid border could be painted to add depth and definition by the PAINT command, followed by a number from a built-in set of 50 different patterns and designs. There is also a PRINT command that will allow you to create a permanent hard copy of your creation on an Epson or IDS printer.

Before we get to the all-important installation and graphics usage, it appears that SigmaSoft and Systems is leaving no stone unturned. There are now modified versions of the IGC available for the H/Z-19 and H-8, as well as for H-89 systems containing the DG-89 or TMSI H-1000 16-bit replacement CPU boards. Since the IGC contains its own power supply, you can move a wire jumper for 240-Volt usage and a jumper pin on the IGC to adjust the screen refresh rate from 60 Hz to 50 Hz. This is ideal for those who are accustomed to stuffing their H-89's into pet carriers for quick trips abroad.

Installing The IGC

At first glance, you'll wonder where all the stuff goes. The IGC is a complex board measuring approximately 9" x 9.25", flanked by two sturdy mounting brackets. There are 40 IC's, not counting RAM chips. Resistor networks are used to minimize the discrete component count. I had ordered the 128k version which had 64k chips piggy-backed on top of each other. Not long after, I found a source that could sell me 256K/150ns chips for less than $16.00, so I replaced the value of one Silver Mica capacitor, moved one jumper and had instant "over-kill".

You also get (as mentioned earlier) SigmaSoft's Universal Parallel Interface board which plugs into the left-side of the CPU board, a separate 1.9 Amp power supply that wedges in between the H-89's Power Distribution board and the Power Transformer, plus three plastic bags containing assorted cables, wire harnesses, mounting hardware and related items, four software disks, a 96-page user's manual, and a large pile of styrofoam chips to clean up.

Depending on your proficiency, installing the IGC hardware can take from 3-5 hours. Those who purchased an assembled H-89 should order an Assembly Manual from Heath, or at least read the IGC Manual several times to become familiar with the installation steps and take notes when they begin to strip down the computer. Fortunately, the only tools required are a screwdriver and an IC puller, though a small non-magnetic screwdriver will serve the purpose.

I thought I'd spend a few pages detailing the installation because there are a few tricky parts that might save you from interrupting your work to write or call SigmaSoft for help. Owners of H/Z-19 Terminals who wish to use the IGC with an H-8 Computer can safely skip over all parts that deal with the Supplemental Power Supply and the desecration of the Terminal Logic board.

Unplug the AC power cord. Open the top shell, disconnect the power plug to the fan and lay the shell aside. Carefully remove all cables attached to your peripheral boards, then unplug the cable from P513 on the lower right side of the CPU board. Remove the Molex sockets from P514 and P515 on the upper left corner. Undo the two screws that secure the CPU board to vertical mounting brackets and slide the CPU board out of the cabinet.

Next, remove any internally mounted disk drives, as well as the mounting shield, power plug and edge card connector located on the back of the drive. If you have the new Mu-Metal Shield containing one or more high-density drives, I would suggest using a socket wrench to remove the hexhead screws underneath the shield so that they don't fall and roll under the Video board.

The most delicate portion of the assembly is mounting the Supplemental Power Supply. First, remove the sockets from P102 and P104 on the Power Distribution board. Do it carefully as you have almost no slack. Then, remove the two screws that secure the large Capacitor that's mounted on posts to the base of the cabinet between the Power Distribution board and the Power Transformer. It has a blue plastic cap and a circular clamp holding a wire harness. DO NOT REMOVE the clamp. Now, looking down on the top of the transformer, remove the hex-nut from the upper left leg and take off the two green ground wires and the braided RFI grounding strap.

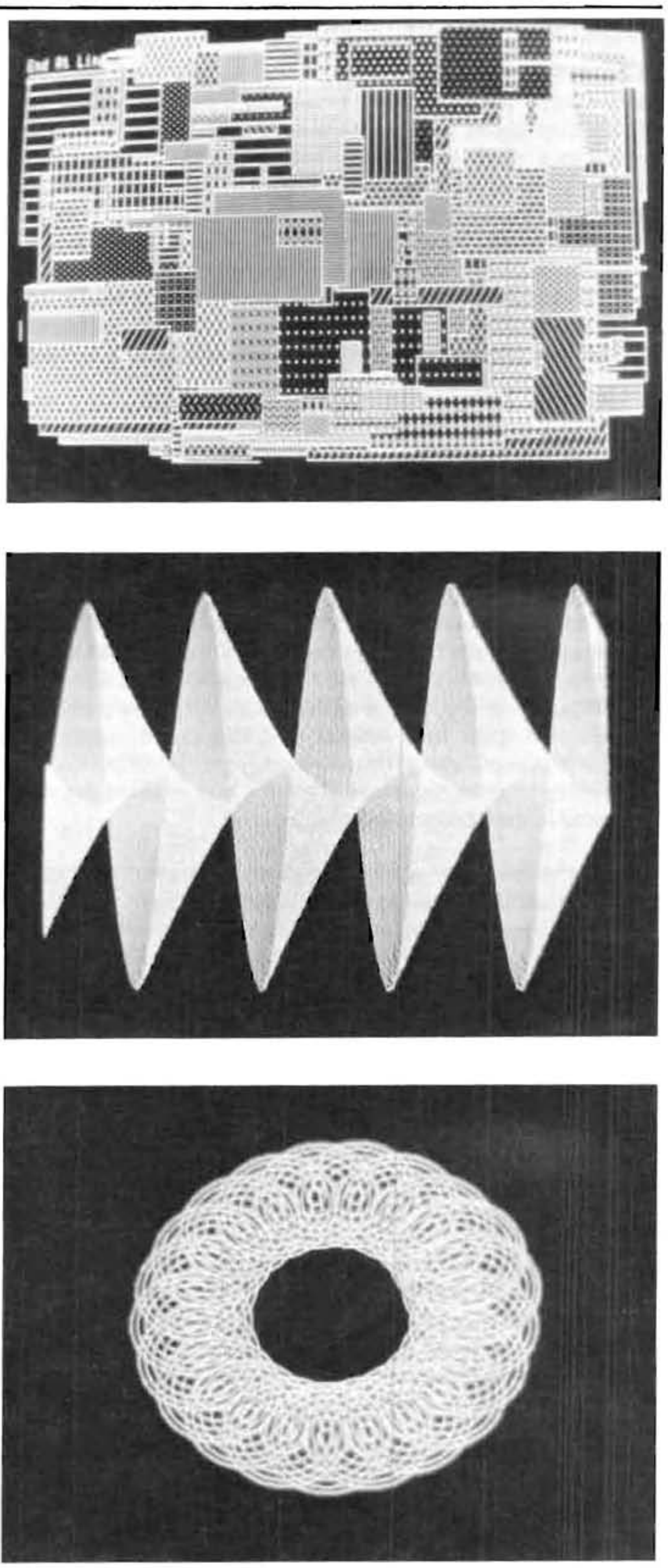
Trace the several wire harnesses in and around the area and judiciously cut through any plastic cable ties and throw them away. This is necessary to provide slack to lift up the capacitor. Run your finger along the braided strap toward the back of the case and remove the screw that secures it near the base of the vertical mounting brackets. Lift both the braided strap and the capacitor with your left hand and, with the right hand holding the Supplemental Power Supply in a vertical position with the transformer pointing down, work it carefully under the wiring harnesses, the capacitor and the braided strap. Line up the two holes on the power board over the mounting posts, position the legs of the capacitor clamp over them and screw both units to the mounting posts.

Part of the hardware you'll receive with the IGC package contains two self-adhesive insulating strips. Take one and cut off a small piece and attach the back end of the braided strap to the side of the box-like structure in the right rear corner that houses the AC power switch. This will position the braided strap away from any exposed metal traces or circuit board components.

Reconnect the green grounding wires and braided strap to the top left leg of the transformer. To do so, you'll have to lift up the front of the computer and slide your right hand under the base of the cabinet to hold the screw in place while you tighten the nut. While the manual advises you to make certain the braided strap is not near any boards or components, it is a good idea to take some insulated electrical tape and wrap it around that section which overlaps the supplemental power supply you've just installed. Most hobbyists usually prowl around inside their H-89 periodically, and it is possible that pressure on the strap will cause it to ultimately make contact with areas it shouldn't.

Replace sockets P102 and P104 and bundle up the wire harnesses with the extra cable ties that SigmaSoft has provided for this purpose. Breathe a sigh of relief.

Take the other self-adhesive insulation tape and place it over the metal band that you see on the top front section of the CRT. Use the remaining portion of the first piece if you don't have enough to completely cover it. The front of the IGC Graphics board will rest over the top of the CRT, and the ground plane on the board could come in contact with the CRT. Plastic feet have been mounted underneath the IGC board as an added precautionary measure, indicative of the thoughtful design that has gone into



Sample Graphic Screen Displays

this package.

AC current to the Supplemental Power Supply is accessed in a unique way. The Molex socket emanating from the H-89's Power Distribution board that connects to the fan motor is now attached to a short, white cable and is plugged into the Supplemental Power Supply board at the point indicated. A second, longer cord is plugged into an adjoining socket and now becomes the power source for the fan. The longest white cable plugs into a solitary socket on the Supplemental Power Supply and will supply DC current for the IGC Graphics board. The sockets on these cables have locking ramps. This means they can only be inserted one way and cannot be pulled loose accidentally.

Before proceeding any further, plug in the computer's power cord, and take a DC voltage reading between the two pins of the plug that will go to the IGC board. You should obtain a reading of 9-Volts, plus or minus 2-Volts. Mine had a reading of 12.5-Volts, which I suspected was probably due to different types of transformers used by SigmaSoft. Later correspondence with Clay Montgomery of SigmaSoft bore out this contention. The IGC board has an onboard MLM 309K Transistor that will further reduce and filter the DC current to a tolerance between 4.5 and 5.5-Volts. SigmaSoft advised me that the IGC draws about 800 milliamps depending on the options installed. "Our early tests showed that most fully loaded H/Z-89's could supply this amount of power through the existing supply. But Heath Co. and some other sources disagreed. We felt it best to avoid the issue and include the power supply."

Unplug the computer again and pick up the wide ribbon cable that has four DIP plugs sprouting from the lower end like a deformed squid. Follow the clear pictorial and the text, and remove four (4) ICs from the Terminal Logic board. The DIP plugs will go into the sockets as illustrated. Owing to variations in the Terminal Logic board, the older model and the newer designated "A" unit, the chips to be removed will have different ID numbers. All ICs removed will no longer be used, except for the Character Generator ROM at location U473 or U420. This will be placed into the IGC Graphics board. The functionally equivalent 74LS series ICs that you removed from the Terminal Logic board are on the IGC, so place them in protective foam or aluminum foil and put them away. You will have to exercise care when you insert the DIP plugs because they have notoriously weak pins that easily break and bend. The cable I received did not have the clusters of wires separated enough for me to reach some of the sockets, so you may have to widen the tear a bit. Ribbon cables are delicate, so make certain that none of the surrounding insulation tears and exposes the wire strands. If this does happen, use electrical tape to cover them.

In order to use the Interlaced Mode of graphics operation on the IGC for 640 x 500 pixel resolution, the matrix of the H/Z-89 or H/Z-19 character set must be increased from 8 x 10 to 8 x 20, and this can only be accomplished with ROM's designed for this purpose. Neither Heath's standard ROM on the Terminal Logic board nor the enhanced Watzman ROM (from the Heath/Zenith Users' Group) support Interlaced resolution. So before having proceeded with any part of the installation, it would be advisable to install either the ULTRA ROM from Software Wizardry (now also called First Capital Computers, 1106 First Capital Drive, St. Charles, MO 63301, and sells for $49.95) or the SUPER-19 ROM from Accusonics, which is available from ATG Systems, Inc., 11 Intervale Road, Wellesley Hills, MA 02181, also for $49.95. If you opt to use either of these ROM's, you should NOT install the U473/U420 ROM on the IGC.

Now, turn your attention to the CPU board and the Universal Parallel Interface that came with the IGC package. The UPI plugs into any of the 3 left-side expansion slots on the CPU Logic board. If you have the Heath (or equivalent) 16k RAM expansion board in your H-89 (at P503), and the H/Z-37 Soft-Sectored Disk Controller, which requires a jumper cable be plugged into pin 14 on P502, I would suggest moving this jumper to P501 and install the UPI at P502. If you use P501, the UPI will be wedged very tightly against socket P515 when you reinsert the CPU Logic board, and it is possible that component leads from the UPI might touch one of the metal shoes inside the P515 socket.

Locate U553 on the CPU Logic board. This is the first IC to the left of DIP Switch S501. Remove it and put it away. Take the CPU Control Cable, which has a single 8-pin connector at one end, and plug it "slot" side up to the 8-pin horizontal mounting connector on the UPI. The free end of this cable has a 14-pin DIP plug which goes to the U553 socket. The fit is relatively tight and I have suggested to SigmaSoft that they allow an extra inch or so of slack to make it easier to remove the UPI board if it has to be taken out. Next, install the Parallel Interface cable. This has a DB-25 connector at one end and two single row connectors at the other end — one a 10-pin, the other a 12-pin. Plug these single pin connectors to their respective vertical header posts on the UPI. There are two such units for Parallel output to printers or plotters.

Now replace the CPU board by sliding it down the tracks in the mounting brackets. Plug in the socket to P515, followed by P514 and P513.

Two small spacers are included in the IGC hardware package. Each has a smaller end which you must twist into the mounting hole on the inside left and right mounting brackets on the IGC. At this point, position the Parallel Interface cable from the UPI over the top of the CPU and Terminal Logic boards and out of the back of the case. Place the IGC board over the CRT, and then push the long screws provided by SigmaSoft through the spacers and the mounting brackets and screw them into the retaining lugs on the sides of the CPU board.

In the final steps, take the free end of the ribbon cable you had attached to the Terminal Logic board and push the 34-pin socket into the IGC header on the left marked "Terminal". Now take the 12-inch ribbon cable with the 34-pin sockets at each end and push one end into the UPI header socket. Trace the top edge of the ribbon from the UPI socket to the other end, making certain that it is on the left, work it through the space between the IGC and the CPU boards and plug into the IGC header marked "CPU".

When I did my installation of this cable, I had the plug reversed because the text was not very explicit. In consequence, when I tried to run the graphics test utilities, the computer locked up with a horizontal display of lines. If you encounter the same difficulty, just flip the plug at either end of the cable and the unit will perform as expected. These units are hand assembled, fully tested and burned in prior to shipment, so any malfunction can generally be traced to a minor installation error. An orientation stripe at one edge of the ribbon cable would be helpful.

If you ordered the Atari/Wico Joystick controller cable, connect it to either the left or right joystick ports on the back of the IGC. Install the disk drive and shield and plug in the cables to all of your accessory cards. If you find that the additional cables sloping over the top of your CPU and Terminal Logic boards interferes with the closing of the cabinet shell, you can remove the two hinges and install spacers. Or you can order a new backplane that provides you with an additional DB-25 cutout, as well as DB-9 cutouts, from either Magnolia Microsystems or Kres Engineering for approximately $35.00. This will give your installation a finished look.

Plug in the computer. Turn it on.

If it beeps twice you didn't screw things up. Boot-up a system disk and run the IGCTEST utility which checks out all the installed boards, cables and memory. No error messages? OK —let's play IGC graphics.

Turning On With The IGC

The IGC Software comes on two HDOS and two CP/M disks. One disk of each set contains the IGC and PS (Pseudo Disk) device drivers, a test utility, sample BASIC programs, source files for graphic display character sets and a demo program. The sample program files are more extensive under HDOS because the CP/M disk requires an additional file called LOADD.COM.

This is a very interesting and unique feature of the IGC, as it allows for the painless installation of the CP/M version onto your system volume. We'll get to this in just a moment.

The second disk for each version contains the Universal Parallel Device Drive. If you use both the HDOS and CP/M Operating Systems, create a System Volume for each version, and include the FORMAT.COM, PIP.COM, STAT.COM amd SYSGEN.COM on the latter. Don't bother with MOVCPM.17 and/or MOVCPM37. You won't need them.

Create a back-up copy of each of the four disks and put the originals in a safe place. PIP the appropriate IGC.DVD, UP.DVD and PS.DVD drivers from the HDOS distribution disk to your new System Volume. It's advisable to use your full System Volume version which includes INIT, SET and SYSGEN because they will be required to make additional working disks and enable you to create a Bootable device out of your Pseudo Disk. If you are using high-density disks, I would suggest including all the sample programs and source files because it makes it easier to work with. You should also add either MBASIC.ABS or BENTON HARBOR BASIC and a Text Editor, like PIE or EDIT.

Your CP/M System Volume will require the IGC.COM, UPC.COM, PDC.COM and LOADD.COM files. Bear in mind that both versions of the IGC Graphics Device Driver contain Serial and Parallel versions for Epson and IDS printers, and you must choose the appropriate driver. After the DVD and COM files have been transferred to your System Volumes, you must rename them to remove the numerical designation that identified the various drivers on the original distribution disks. Then you must reBOOT the computer so that HDOS will recognize the existence of the newly installed drivers. Using your SET.ABS and SET.COM utilities, you will then have to set the various parameters of each device driver: the I/O ports, terminal characteristics, printer initialization sequences, etc. Rather a boring procedure, at best. Fortunately, the User Manual has laid out all the appropriate procedures with reasonable clarity. Again, after SETting all your parameters, reBOOT so that the System Volume will recognize the options.

The CP/M procedure is virtually identical to HDOS, and that's because SigmaSoft and Systems created CP/M Device Drivers patterned after HDOS conventions. With HDOS, you access a device or device driver with the MOUNT command. CP/M, on the other hand, requires that I/O drivers and other continually resident programs be written and assembled into the system BIOS. SigmaSoft's LOADD.COM program does it for you. Once you have your CP/M System Volume installed with the device drivers, you access each of the Modules (Graphics, Pseudo Disk and Printers) with LOADD.IGC or LOADD.PDC. This enables the user to mount and dismount the devices as the IGC and PS functions cannot be resident at the same time. That is, you cannot run a graphics program from IGC while trying to use the RAM space for editing or assembling. Your files will overlap and be destroyed.

It also doesn't matter which implementation of CP/M you may have, whether it's any of the Heath/Zenith versions, Magnolia Microsystems, CDR, Livingston BIOS-80, etc. The SigmaSoft LOADD.COM file will access each one with the same ease without altering them. You do have to LOADD each module every time you BOOT your system, but that's a small price for such versatility. And it certainly speeds software development by not having to allow for the peculiarities of every CP/M BIOS available. I actually got to like CP/M after playing with the IGC for a few weeks.

The User Manual provides you with 7 full BASIC graphic programs for a variety of circles, line, boxes and graphic displays. The programs are written in Benton Harbor BASIC and are easily converted to MBASIC by changing the channel access command at the beginning of the program. Hard copy prints demonstrating the results of these programs are also included. One is struck by the sparseness of the programming required to produce rather spectacular results.

Once you've written a program, you can use the PIP command to put it on your disk, to run it on your console terminal or to temporarily dump it to the Pseudo Disk. The Pseudo Disk, under HDOS becomes device PS:. Under CP/M it's M:. To execute a program under HDOS, you PIP IG:; and with CP/M you PIP PUN:. I wonder if one was intended. There are numerous ways in which to communicate with each of the HDOS and CP/M device drivers, and that will send you scurrying to your Operating Manuals.

There are 24 Command Statements to contend with. Fortunately, most of them are self-explanatory. The DISPLAY command, for example, allows any of three video pages to be selected for display on the screen. The DRAW command allows any of three video pages to be selected for drawing. The creation of a graphic display is accessed either through a Text Editor or a high-level programming language. The former requires much less typing, the latter more sophistication with complex displays that may require Machine Language subroutines. In each case, you can specify under the Cartesian coordinates whether your 0,0 origin (for X and Y) originates in the Center, the Bottom or Top of your working screen. In addition, you can define absolute and relative coordinates.

As an example, let's DRAW three triangles and SCALE them to different sizes. Comments in parentheses are mine.

```
Editor:

      Origin Bottom  (This establishes your coordinate)
      Mode -Erase    (This erases page to black)
      Style Ø        (Sets solid line)
      Go Ø,Ø         (Place relative coordinates in center)
      Scale 1,3      (1st triangle)
      Line = #Ø,5Ø = #5Ø,-5Ø = #-5Ø,Ø
      Go 1ØØ,1ØØ     (Places relative coordinates 1ØØ pixels
                          up and to the right)
       Scale 2,2         (2nd triangle)
       Line = #Ø,5Ø = #5Ø,-5Ø = #-5Ø,Ø
       Go 2ØØ,2ØØ        (Places relative coordinates 2ØØ pixels
                          up and to the right of center)
       Scale 3,1          (3rd triangle)
       Line = #Ø,5Ø = #5Ø,-5Ø = #-5Ø,Ø
```

Under Benton Harbor BASIC, the same program would have to be written as follows:

```
ØØØ1Ø REM Extended Benton Harbor BASIC
ØØØ2Ø REM
ØØØ3Ø OPEN "IG:" FOR WRITE AS FILE #1
ØØØ4Ø X=1:Y=3
ØØØ5Ø PRINT CHR$(1);"Origin Bottom"
ØØØ6Ø PRINT CHR$(1);"Mode -Erase"
ØØØ7Ø PRINT CHR$(1);"Style Ø"
ØØØ8Ø PRINT CHR$(1);"Go Ø,Ø"
ØØØ9Ø FOR A=1 TO 3
ØØ1ØØ PRINT CHR$(1);"Scale",X,Y
ØØ11Ø PRINT CHR$(1);"Line = #Ø,5Ø = #5Ø,-5Ø";
ØØ12Ø PRINT CHR$(1);"= #-5Ø,Ø"
ØØ13Ø PRINT CHR$(1);"Go #1ØØ,1ØØ"
ØØ14Ø X=X+1:Y=Y+1
ØØ15Ø NEXT A
ØØ16Ø CLOSE #1
ØØ17Ø END
```

The SCALE command also enlarges and flips images drawn with LINE commands in relative coordinates. You must specify a scaling factor (multiplier) for both X and Y coordinates. It's obvious to see the difference and the relative ease by which a graphic representation can be created and executed on the screen through a Text Editor, though BASIC or Assembly Language will certainly give you greater flexibility. However, a Text Editor will allow you to create Graphic Files for special character sets, animated figures, electrical symbols for circuit design, graphs, etc. for quick access and integration into a larger program.

Whether you are working with Normal pixel resolution or Interlaced pixel modes, your constants are always 640 x 250. Interlaced mode, which has 640 x 500 pixels, automatically scales that count to a constant of 500 for compatibility between non-interlaced and interlace software.

Frank T. Clark, a Zenith Data Systems Software Consultant whom I've pestered occasionally in the last two years with boring questions on CP/M, had an interesting article in the December 1984 REMark, entitled "Interlaced Anyone?" in which he deals with some of the problems associated with creating high-resolution Interlaced graphics for the Z-100 in terms of man hours and the ultimate consumer cost for such a product as it relates to the small demand he envisions for such software. It would be interesting to see how he would react to the IGC's capabilities, if he could be persuaded to dig out a dusty H-89 from his St. Joseph, MI office and install an IGC from SigmaSoft.

SigmaSoft's manual is very readable. Certainly one of the better efforts from a Heath/Zenith peripheral manufacturer, and the material is presented in a logical progression. Each of the Command Statements is defined with examples for Editor and BASIC usage. Examples are also given for using the Joystick ports and a Lightpen. There is a section that is geared for experienced Assembly Language programmers. Included is a troubleshooting section that will allow you to isolate most installation problems quickly. Any later problems can be attributed either to a loose connection or a rare chip failure. At any rate, SigmaSoft has proven itself to be a concerned supplier that will support its product to the fullest extent.

The final pages provide pinout information on all the many connectors on the IGC and UPI boards, along with a complete list of all parts, chips, components by name, serial or ID number and the current SigmaSoft replacement cost. Except for bare boards, proprietary ROM's and the special cables, alternate sources for parts (with addresses) are given on each item from suppliers such as Jameco Electronics, Hamilton Avnet Electronics, Heath Co., and God forbid, even Radio Shack.

Missing are page numbers in the manual (?) and a working schematic. Although the manual comes in an attractive Lexhide binder with a clip on the spine, all the abuse I've given the manual hasn't resulted in any pages coming loose. I will, however, number the pages, punch some holes and stick the manual in a ring binder for more permanence. SigmaSoft informs me that a schematic was not complete at the time the IGC was released in October 1984, but that one would be sent to all registered owners in early Spring 1985.

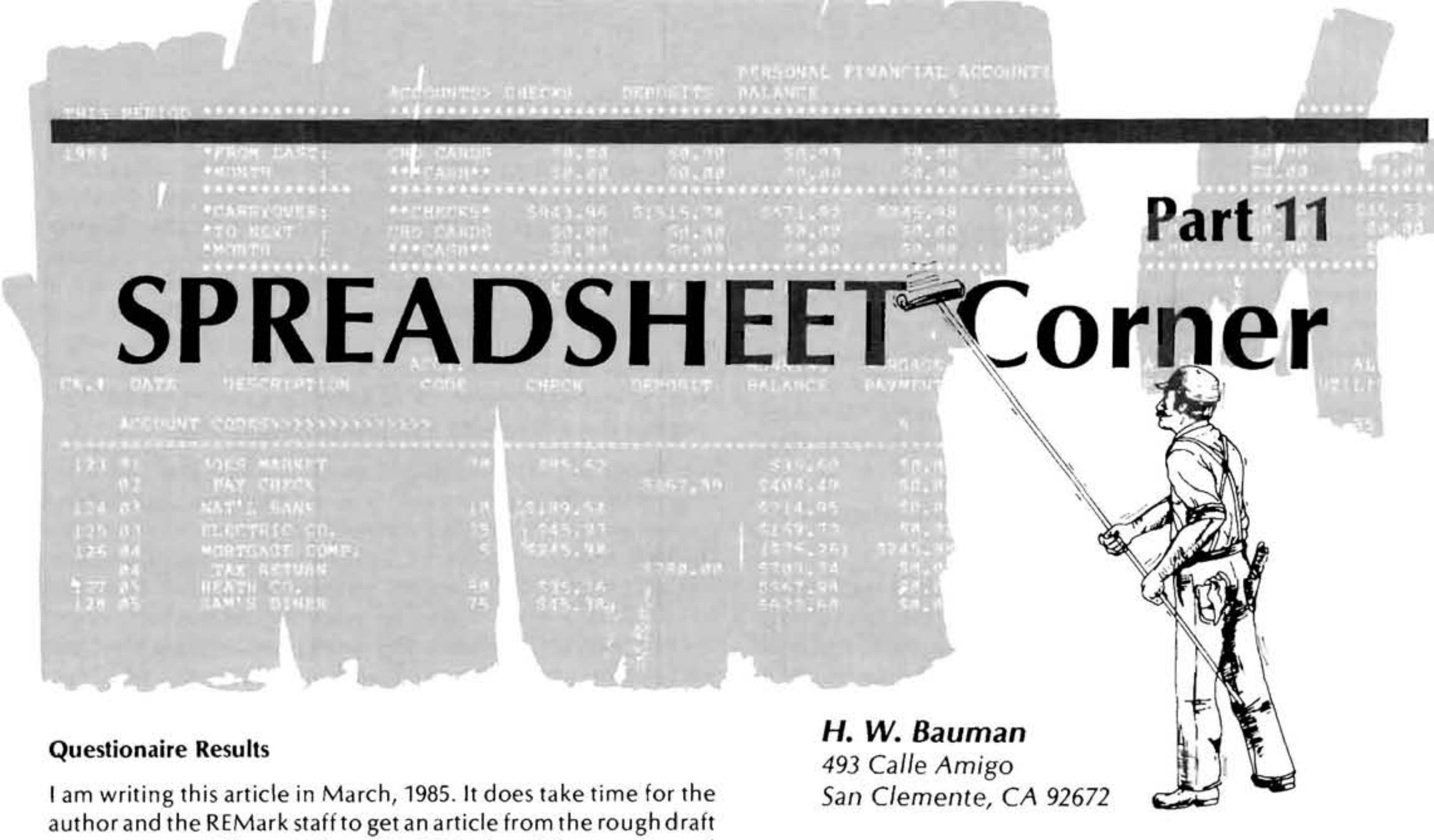
The product workmanship is up to Sigmasoft's usual high standards, and it will be interesting to see what kind of software development some of the Heath/Zenith software houses will come up with in terms of support. The large 256k RAM memory should certainly be an incentive for a mini CAD graphic development system. In conjunction with SPOOLDISK 89 from FBE Research, the H-89 user can create with software that he already owns, a complete integrated working environment for most business applications. The Pseudo Disk on the IGC (as with the SS: device under SPOOLDISK 89) can also be used not only to assemble and compile source codes and programs, but sort database files and mailing lists in a fraction of the time required to perform similar functions with disk drives.

Except for loading and saving files, my drives have been curiously silent, and the ability to RESET each of these two systems independently and store in memory any work in progress, while using the computer for yet a third function, gives the old H-89 capabilities and sophistication that rivals the new, sleek beauties you see in the expensive color ads.

SigmaSoft is selling the Source Codes for both the IGC Graphics and Pseudo Disk device drivers at a very reasonable $95.00 each. I can think of a few enterprising outfits that could have a field day if they turned their programmers loose.

The price of the basic IGC package with the Universal Parallel Interface, all hardware, software and 64k RAM is $495. This is quite a bargain when you consider its potentials. For a complete and current price list, write to:

SigmaSoft and Systems
4488 Spring Valley #107
Dallas, Texas 75234
(214) 392-1025



Part 11 SPREADSHEET Corner

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

Questionaire Results

I am writing this article in March, 1985. It does take time for the author and the REMark staff to get an article from the rough draft stage to your mail box. Walt and his REMark staff do a really good job of getting the articles into print. I have received a large number of letters (now over 170) concerning "SPREADSHEET Corner"! I appreciate these letters and I WILL answer the ones with a SASE (business-size) enclosed when appropriate. I was surprised how many upset readers wrote to me asking why there wasn't any "SPREADSHEET Corner" articles in the December, 1984 and January, 1985 issues. Scheduling explains December, and the January issue is sort of a catalog issue without the regular features. With this kind of reader response, I will keep "SPREADSHEET Corner" coming to you on a regular basis. Did you notice that the February, 1985 issue had two articles to make up for December?

"SPREADSHEET Corner Part 7" had a list of questions requesting information from the readers about their desires, ideas, and comments. I am still receiving letters nearly everyday, but I think that I have enough responses to report on the trends:

1. Which type of article do you like best?

   First, articles for novices. They want short, easy projects that will do something useful. They prefer articles that are not step-by-step, but the article should include a lot of detailed explanations. Most readers told me that I explained items better than what they found in the manuals and purchased books. My examples rated very high. One pat-on-the-back for the author!

   Second, the advanced readers said that they did not like the step-by-step approach. They want the commands, formulas, and functions explained where the documentation is missing or poorly done in the manual. They rated my articles high for my explanations. This group consists of mainly business users.

2. Do you like short software reviews in articles like the one on MYCALC in this article?

   Nearly all readers said that they liked SHORT reviews, but they did not want the articles to become Software Reviews because they find too many of these in other magazines already. They want to read about the new or revised software; just its main features, which H/Z computers the software will work with, what makes it special to H/Z.

3. I have worked with LOTUS 1-2-3. Would you rather have articles prepared with a different program? If so, which?

   Again, nearly all readers suggested that I stay with 1-2-3. They do not want me to change with each article or repeat the article with different software. Some readers asked me to point out where there are major differences to look for with other computers and software and explain these differences.

4. Tell me what computer set-up you are using. Do you have an H8, H/Z-89/90, H/Z-100, H/Z-150/160, or what? How much working memory (RAM) do you have? What Printer will you use? Are you planning any changes soon?

   First, most of the readers are using the H/Z-100 with two DSDD drives and 192K RAM. Second, the H/Z-150/160 (along with something called IBM) with two DSDD drives and 128 to 320K RAM is being used. To my surprise, the H8 and H/Z-89/90 computers were way down on the list and they had two drives and 64K RAM. The types of printers really varied. Epson and MPI were at the top by a small margin with H/Z-25 and Okidata following. Most readers indicated that a printer is the item that they are thinking of changing. They asked what I advise them to buy for spreadsheet graphics. Other changes planned are added RAM and color monitors. Again, they would like to see these written as short reviews in "SPREADSHEET Corner".

5. What spreadsheet software are you using? Are you planning any changes soon?

   LOTUS 1-2-3 is the big leader by the readers that responded. This was followed by Multiplan and PeachCalc/SuperCalc. Many readers are planning to add LOTUS 1-2-3 unless I can advise them about something better. What are the pros and cons!

6. Do you have a database management program? If not, do you plan on adding one? Which one? Would you like me to include short software reviews on some? Which?

   Most of the readers do not have a database management program. Most of them would like to have one, but complain that they are too expensive and that they do not know how or when to use one. Nearly all readers asked to have more articles about database programs. dBASEII was the one readers asked about by name the most often. The readers asked which database program I used and why? They want more articles with short, easy programs showing how they can use these programs.

7. What project would you like to see covered in future articles?

   This is where I received a BIG response. Many readers did not answer the other questions, but did refer to this one. Novice, advanced, business–user, investors, and teachers all asked for a large variety of subjects to be in future articles. I have enough ideas to write "SPREADSHEET Corner" articles for years! I will not attempt to list them all here, as the list would not fit into this article. Three subjects appeared the most times — Stock Portfolios, programs related to taxes, and Business–Analysis Programs. Keep the ideas coming! The ones that I receive the greatest requests for will become the "squeaky wheel" and will be done first! I will do a Stock Portfolio in this article (only one of many that could be done on the subject).

8. Would you like to submit a project for "SPREADSHEET Corner" to publish in a future article? When would you have it?

   This was disappointing to me. I wanted to get reader participation so that the readers would see other people's ways of using spreadsheets. I received letters saying "I am not qualified", "I do not know how to write up a project", "why should I write one when you are doing a good job", "I would not know how or where to start", etc. I will work with anyone who would like to try!

9. Do you have any comments, suggestions, corrections, etc. about "SPREADSHEET Corner"?

   Here again, I was disappointed! I was hoping for the readers to start participating, NOT "keep up the good work", "please continue the articles", etc. A few readers gave me a few ideas. First, tell the readers at the beginning of the article what computer, software, etc. will be needed to do the project so that they do not have to read the whole article to find out. Second, repeat the usage of difficult commands, formulas, and functions so that the reader does not have to refer back to previous articles and manuals as much. Tell us how to use graphics and print graphics with spreadsheets.

I want to thank the readers that DID participate! If you did not, I would still like to receive your answers. With Walt's and REMark's concurrence, "SPREADSHEET Corner" will continue to be a regular REMark feature! But, the readers must support it. Letters to REMark and/or author are the way we know you want them and what you want in them.

Stock Portfolio Project

If you own a portfolio of common stocks, analysis of your holdings should be made every business day to determine if you are obtaining the maximum return from the invested funds. With calculator, pencil & paper portfolio tracking can be tedious and time consuming if you hold more than a couple of different stocks. With your computer and a spreadsheet program like the one we will prepare in this article, this task can be transformed into a routine that can be done in a matter of minutes. In fact, it will take you longer to find and read the data on the financial pages of the morning paper than the recalculation, when needed for the changes, which will take but a few minutes at most. As we progress with "SPREADSHEET Corner", we will do a program that will include a database and graphing.

The owner of the stocks should watch the morning's financial pages or the TV financial channel, if you have one, and find the current prices of the stocks held as well as how much their prices changed compared to the market as a whole. This information will tell you how much money you've made or lost since you bought the shares, AND it will tell you how your stocks are doing as compared to the market. This newspaper or TV data will not tell you what your selling cost would be if you were to sell today. The paper's data will not advise you of what your annualized percentage gain or loss is and it will not tell you how long you have had the stock. The spreadsheet program does answer these important facts for you easily and everyday! You will enter the data and then examine the output figures on the screen (they could be printed also) to check their values, to learn their percentage change on an annualized basis, how long you have held the stock for long term capital gain purposes, and know how many dollars you would receive if you were to sell today after selling commissions. I want to keep this project rather simple, easy to use and have it flexible enough so that it can be used with nearly everyone's computer and software. We could do other projects that would do our calculations and reports in a better format, but would they meet the above specifications?

I will create this project using three modules. Remember, if we want to keep our projects simple, break them down into modules that are easy to follow and understand. It really does not take extra time or effort to do them in modules if they are planned for at the start! Each module should be and will be independent of the other as much as possible. Therefore, if the user wants to avoid "forward reference" and "circular reference" problems, they MUST be arranged in a special order on this worksheet. This is a project that can get into these problems very easily. If you do not know about these problems, I have discussed them in past articles that you can refer to or refer to your manual. I am sorry that I do not have the space in this article for a review of these forward and circular references again.

Module #3 will be the Stock Tracking Worksheet. Module #2 will contain the formulas to calculate the stock buying and selling commissions. Module #1 will have the Lookup Tables to determine Selling/Buying Commissions and the holding period in days. I used a lookup table for the holding period because most of the early spreadsheet packages did not provide means for determining days between dates like the newer generation spreadsheets, like 1–2–3, etc. If the reader knows how to use this capability, the module can be simplified. If you do not, this method will work with any version. In a future article, I will discuss how to use the DATE functions for some of the newer generation spreadsheets. I picked and put in fictitious commissions that may or may not be like those that you would have to pay. So, be sure to enter YOUR rates into the tables. I am sure that you can see how to do it from my example. I will discuss this part of the table again later in the article. I have also used fictitious stocks and stock prices. BEWARE! Neither the author nor REMark are stock experts, so the reader is advised to discuss their investments with trained personnel!

By creating the modules separately, it is easy to vary any one module without affecting the other modules. I do not think that you will have to worry about saving worksheet space or RAM, because this project does not require much of either unless the user has an extraordinarily large portfolio! After you consider the forward and circular reference problem, decide how you want to arrange your modules on the worksheet. The modules could be arranged vertically, horizontally, or step-like on the worksheet. I am choosing the step-like arrangement. This allows rows to be added to the modules without worry about running into another module that could be below the one being enlarged so that nearly any number of stock holdings could be added.

Even though I am going to design Module #3 first, I am going to use it in the lower, right portion of my worksheet. See if you can decide why I would do that? I think that the learning process works better when the user has to give a little thought to the projects. I hope you agree.

I will be using my 1-2-3 package with the H-100, but the readers should be able to adapt the project to their spreadsheet program if they have been following "SPREADSHEET Corner" and doing the various projects. Refer back to these articles and your manual if need be. Also, do not forget the HELP screens that most spreadsheet packages provide!

Figure 3 will show this module. From this figure I expect the reader to prepare a "SPREADSHEET Preparation Form" as I have discussed many times. I will continue to stress the need for this planning tool. This must be your first step in creating a spreadsheet prog-am!

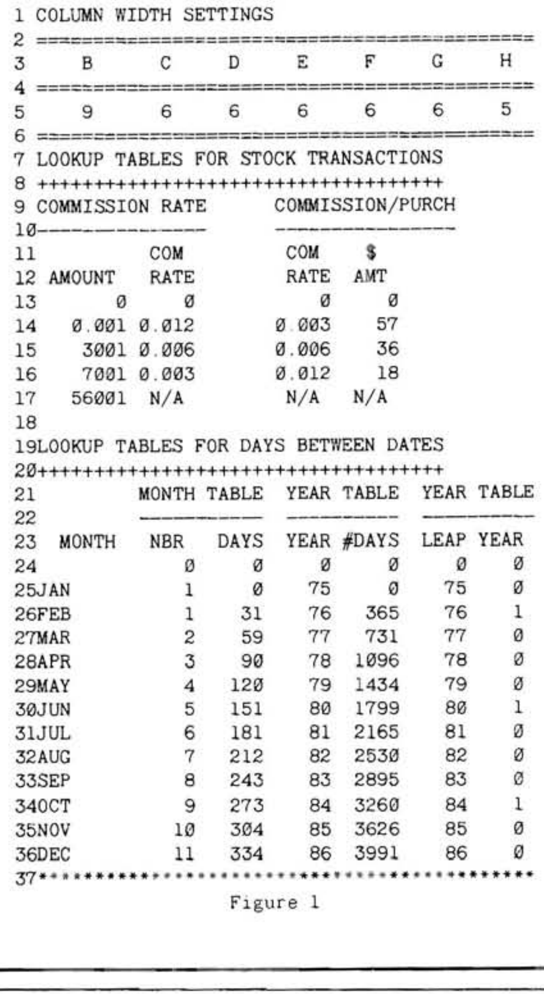
The module makes use of several special features that nearly all spreadsheet packages provide. We will make use of the ability to COPY values from one part of a program to another without copying the underlying formulas. We will use the FORMAT option to vary column widths. It is not necessary to use the same column widths that I used. Remember, that most printers can be configured to vary the type size to increase the number of columns across a single page. I have used FORMAT to set fixed, 2 decimal in some columns while using integers in others. Again, except for the earliest spreadsheet packages, it is possible to mix these formats on one worksheet. I will be using LOOKUP, SUM, and IF functions often. I do not have the room in this article for a review, so if you do not remember these functions, please refer to past articles, the manual, and HELP screens for a review. I will take the space for a Review in a coming article for new readers that have just joined HUG, REMark, and "SPREADSHEET Corner".

Figure 3 shows what stocks you own, how much you invested in each of them, what they are worth today, how well they are doing as compared to the market as a whole, and so forth. This is the module where you will do the daily analysis.

From your "SPREADSHEET Corner Preparation Form" (You did one, RIGHT?) for this module (Figure 3), position the cursor at the upper-right corner of the module's location and set the column widths that you have chosen. Next, enter the titles and column headings you have decided on. Oh Yes! This is good project to try the use of the Global Format to set the worksheet for the fixed, two decimal point format as I have shown, then set the individual columns/ranges where integers are needed — dates, number of shares, days, etc. — using the integer format. I also set the columns/ranges containing percentages to the percent format with two decimal places.

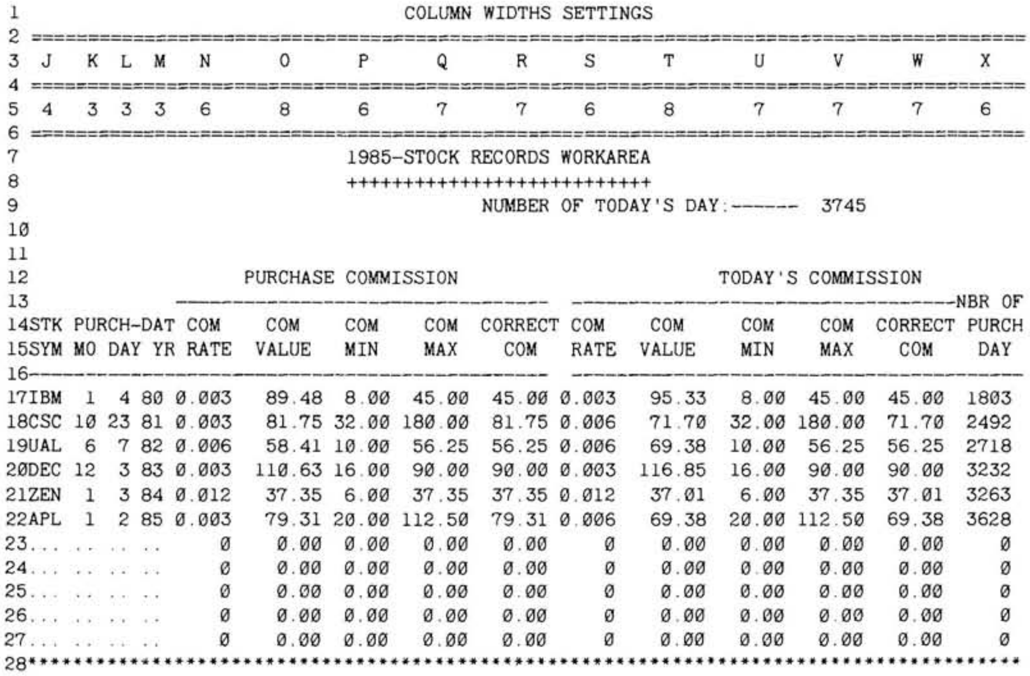
Construction of the worksheet will start with the formulas entered into their cells. (I will use the cell "names" from my Figure 3.)

1. Column AC — multiply the number of shares times the purchase price plus the buying commission (Module 2). Cell AC19 will have — +AA19*AB19+R17 and this formula will be copied down the column for the number of rows needed based on the number of stock holdings.
2. Column AD — use an IF function to check if a stock is held for that row and if the value is zero, the result is zero, else calculate the number of days that the stock has been held. Module 2 has the "number" of Today's Day in cell V9. Column X in Module 2 has the "number" of the purchase day. (We will find out about these when the Module 2 is discussed shortly.) — Cell AD19 will contain this formula — @IF(AA19=0,0,V9-X17). (Do you remember why we use V9? Do you remember the discussion about "relative" vs "absolute" references from a previous article? Check back if need be. I have used the $ sign to tell the program that the cell V9 is an "absolute" reference — that it does not change as we COPY the formula down the number of rows that are being used for the stocks held.)



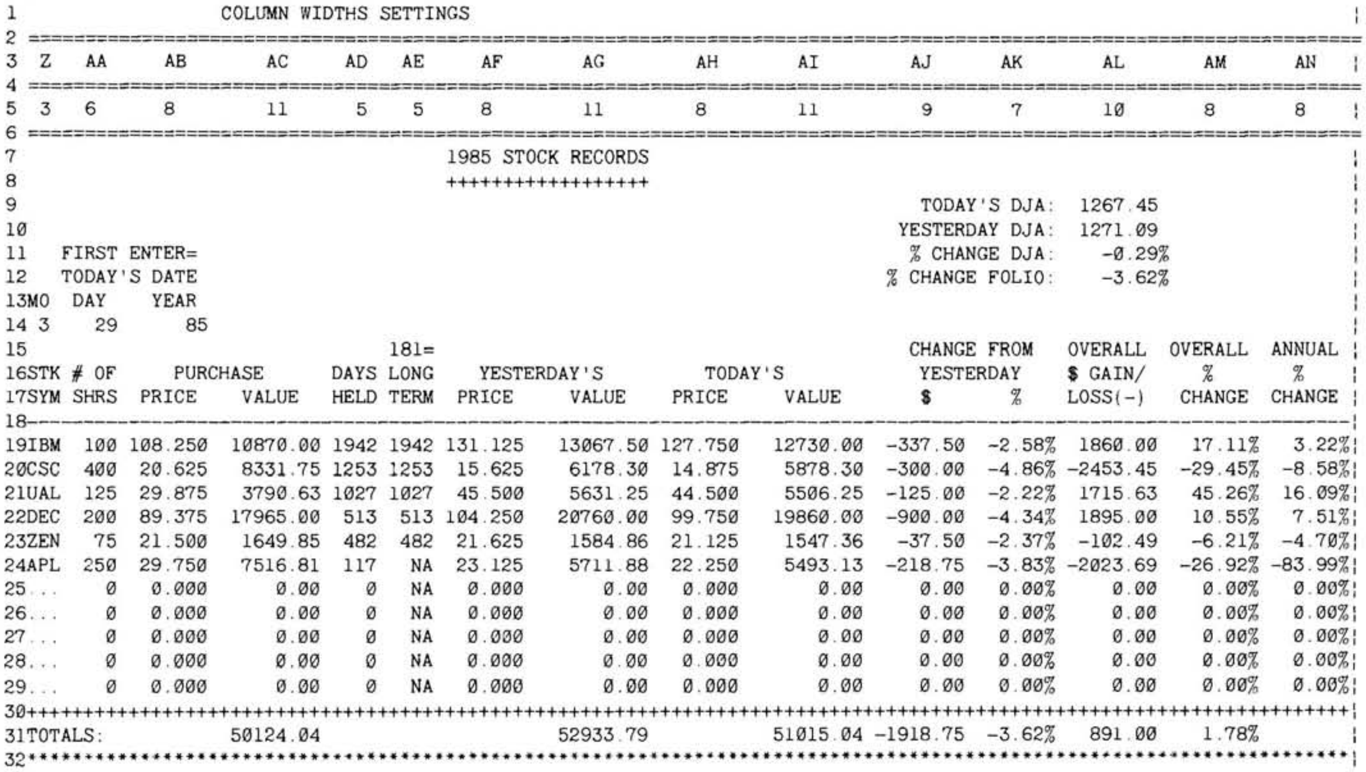
```
1 COLUMN WIDTH SETTINGS
2 ===========================================
3     B      C     D      E     F     G     H
4 ===========================================
5     9      6     6      6     6     6     5
6 ===========================================
7 LOOKUP TABLES FOR STOCK TRANSACTIONS
8 ++++++++++++++++++++++++++++++++++++
9 COMMISSION RATE       COMMISSION/PURCH
10---------------       ----------------
11          COM          COM    $
12 AMOUNT   RATE         RATE  AMT
13        Ø     Ø           Ø     Ø
14    Ø.ØØ1 Ø.Ø12       Ø.ØØ3    57
15     3ØØ1 Ø.ØØ6       Ø.ØØ6    36
16     7ØØ1 Ø.ØØ3       Ø.Ø12    18
17    56ØØ1 N/A          N/A   N/A
18
19LOOKUP TABLES FOR DAYS BETWEEN DATES
20++++++++++++++++++++++++++++++++++++
21         MONTH TABLE  YEAR TABLE  YEAR TABLE
22         -----------  ----------  ----------
23  MONTH   NBR   DAYS  YEAR #DAYS  LEAP YEAR
24            Ø      Ø     Ø     Ø     Ø    Ø
25JAN         1      Ø    75     Ø    75    Ø
26FEB         1     31    76   365    76    1
27MAR         2     59    77   731    77    Ø
28APR         3     9Ø    78  1Ø96    78    Ø
29MAY         4    12Ø    79  1434    79    Ø
3ØJUN         5    151    8Ø  1799    8Ø    1
31JUL         6    181    81  2165    81    Ø
32AUG         7    212    82  253Ø    82    Ø
33SEP         8    243    83  2895    83    Ø
34OCT         9    273    84  326Ø    84    1
35NOV        1Ø    3Ø4    85  3626    85    Ø
36DEC        11    334    86  3991    86    Ø
37*******************************************
```

Figure 1



```
1                                        COLUMN WIDTHS SETTINGS
2 ================================================================================
3  J   K  L  M   N      O      P      Q      R     S      T      U      V      W     X
4 ================================================================================
5  4   3  3  3   6      8      6      7      7     6      8      7      7      7     6
6 ================================================================================
7                                     1985-STOCK RECORDS WORKAREA
8                                     ++++++++++++++++++++++++++
9                                               NUMBER OF TODAY'S DAY:------  3745
1Ø
11
12                       PURCHASE COMMISSION                        TODAY'S COMMISSION
13                 -------------------------------  --------------------------------NBR OF
14STK PURCH-DAT COM     COM    COM     COM  CORRECT COM     COM     COM     COM CORRECT PURCH
15SYM MO DAY YR RATE  VALUE    MIN     MAX     COM RATE  VALUE     MIN     MAX     COM   DAY
16-----------------------------------------------  --------------------------------------
17IBM  1  4 8Ø Ø.ØØ3   89.48  8.ØØ   45.ØØ   45.ØØ Ø.ØØ3   95.33   8.ØØ  45.ØØ   45.ØØ  18Ø3
18CSC 1Ø 23 81 Ø.ØØ3   81.75 32.ØØ  18Ø.ØØ   81.75 Ø.ØØ6   71.7Ø  32.ØØ 18Ø.ØØ   71.7Ø  2492
19UAL  6  7 82 Ø.ØØ6   58.41 1Ø.ØØ   56.25   56.25 Ø.ØØ6   69.38  1Ø.ØØ  56.25   56.25  2718
2ØDEC 12  3 83 Ø.ØØ3  11Ø.63 16.ØØ   9Ø.ØØ   9Ø.ØØ Ø.ØØ3  116.85  16.ØØ  9Ø.ØØ   9Ø.ØØ  3232
21ZEN  1  3 84 Ø.Ø12   37.35  6.ØØ   37.35   37.35 Ø.Ø12   37.Ø1   6.ØØ  37.35   37.Ø1  3263
22APL  1  2 85 Ø.ØØ3   79.31 2Ø.ØØ  112.5Ø   79.31 Ø.ØØ6   69.38  2Ø.ØØ 112.5Ø   69.38  3628
23... .. .. ..      Ø   Ø.ØØ  Ø.ØØ    Ø.ØØ    Ø.ØØ     Ø    Ø.ØØ   Ø.ØØ   Ø.ØØ    Ø.ØØ     Ø
24... .. .. ..      Ø   Ø.ØØ  Ø.ØØ    Ø.ØØ    Ø.ØØ     Ø    Ø.ØØ   Ø.ØØ   Ø.ØØ    Ø.ØØ     Ø
25... .. .. ..      Ø   Ø.ØØ  Ø.ØØ    Ø.ØØ    Ø.ØØ     Ø    Ø.ØØ   Ø.ØØ   Ø.ØØ    Ø.ØØ     Ø
26... .. .. ..      Ø   Ø.ØØ  Ø.ØØ    Ø.ØØ    Ø.ØØ     Ø    Ø.ØØ   Ø.ØØ   Ø.ØØ    Ø.ØØ     Ø
27... .. .. ..      Ø   Ø.ØØ  Ø.ØØ    Ø.ØØ    Ø.ØØ     Ø    Ø.ØØ   Ø.ØØ   Ø.ØØ    Ø.ØØ     Ø
28*************************************************************************************
```

Figure 2



```
1                 COLUMN WIDTHS SETTINGS                                                                         ¦
2 ============================================================================================================
3  Z   AA     AB         AC     AD   AE      AF         AG         AH         AI         AJ      AK       AL       AM      AN  ¦
4 ============================================================================================================
5  3   6      8          11     5    5       8          11         8          11         9       7        1Ø       8       8   ¦
6 ============================================================================================================
7                                            1985 STOCK RECORDS                                                  ¦
8                                            +++++++++++++++++                                                   ¦
9                                                                                   TODAY'S DJA:  1267.45       ¦
1Ø                                                                              YESTERDAY DJA:  1271.Ø9       ¦
11   FIRST ENTER=                                                                 % CHANGE DJA:    -Ø.29%      ¦
12   TODAY'S DATE                                                               % CHANGE FOLIO:    -3.62%      ¦
13MO  DAY    YEAR                                                                                              ¦
14 3    29      85                                                                                             ¦
15                                181=                                          CHANGE FROM   OVERALL  OVERALL  ANNUAL ¦
16STK # OF    PURCHASE      DAYS LONG    YESTERDAY'S          TODAY'S            YESTERDAY   $ GAIN/     %        %   ¦
17SYM SHRS  PRICE     VALUE  HELD TERM  PRICE       VALUE   PRICE       VALUE        $       %   LOSS(-)   CHANGE  CHANGE ¦
18---------------------------------------------------------------------------------------------------------------¦
19IBM  1ØØ 1Ø8.25Ø  1Ø87Ø.ØØ 1942 1942 131.125   13Ø67.5Ø 127.75Ø   1273Ø.ØØ  -337.5Ø  -2.58%   186Ø.ØØ   17.11%   3.22%¦
2ØCSC  4ØØ  2Ø.625   8331.75 1253 1253  15.625    6178.3Ø  14.875    5878.3Ø  -3ØØ.ØØ  -4.86% -2453.45  -29.45%  -8.58%¦
21UAL  125  29.875   379Ø.63 1Ø27 1Ø27  45.5ØØ    5631.25  44.5ØØ    55Ø6.25  -125.ØØ  -2.22%  1715.63   45.26%  16.Ø9%¦
22DEC  2ØØ  89.375  17965.ØØ  513  513 1Ø4.25Ø   2Ø76Ø.ØØ  99.75Ø   1986Ø.ØØ  -9ØØ.ØØ  -4.34%  1895.ØØ   1Ø.55%   7.51%¦
23ZEN   75  21.5ØØ   1649.85  482  482  21.625    1584.86  21.125    1547.36   -37.5Ø  -2.37%  -1Ø2.49   -6.21%  -4.7Ø%¦
24APL  25Ø  29.75Ø   7516.81  117   NA  23.125    5711.88  22.25Ø    5493.13  -218.75  -3.83% -2Ø23.69  -26.92% -83.99%¦
25...    Ø   Ø.ØØØ      Ø.ØØ    Ø   NA   Ø.ØØØ       Ø.ØØ   Ø.ØØØ       Ø.ØØ     Ø.ØØ   Ø.ØØ%     Ø.ØØ    Ø.ØØ%   Ø.ØØ%¦
26...    Ø   Ø.ØØØ      Ø.ØØ    Ø   NA   Ø.ØØØ       Ø.ØØ   Ø.ØØØ       Ø.ØØ     Ø.ØØ   Ø.ØØ%     Ø.ØØ    Ø.ØØ%   Ø.ØØ%¦
27...    Ø   Ø.ØØØ      Ø.ØØ    Ø   NA   Ø.ØØØ       Ø.ØØ   Ø.ØØØ       Ø.ØØ     Ø.ØØ   Ø.ØØ%     Ø.ØØ    Ø.ØØ%   Ø.ØØ%¦
28...    Ø   Ø.ØØØ      Ø.ØØ    Ø   NA   Ø.ØØØ       Ø.ØØ   Ø.ØØØ       Ø.ØØ     Ø.ØØ   Ø.ØØ%     Ø.ØØ    Ø.ØØ%   Ø.ØØ%¦
29...    Ø   Ø.ØØØ      Ø.ØØ    Ø   NA   Ø.ØØØ       Ø.ØØ   Ø.ØØØ       Ø.ØØ     Ø.ØØ   Ø.ØØ%     Ø.ØØ    Ø.ØØ%   Ø.ØØ%¦
3Ø+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++¦
31TOTALS:          5Ø124.Ø4                        52933.79             51Ø15.Ø4 -1918.75  -3.62%   891.ØØ    1.78%        ¦
32***************************************************************************************************************¦
```

Figure 3

3. Column AE — use an IF function for decisions. We want to know if a stock has been held for enough days for it to qualify as a "Long Term Capital Gain" or not. If the holding period is not long enough, we will put "NA" in the column, else we will put in the same number of days as in column AD. Cell AE19 will have — @IF(AD19<181,@NA,AD19). Do you see how it works? This formula will be copied down the rest of the column. You can put anything you like for NA.
4. Column AG — this one is simple, we multiply the number of shares times Yesterday's Price and subtract the selling commission from Module 2. Cell AG19 would look like this — +AF19*AA19-W17. This formula would be copied down the column as before.
5. Column AI — again, multiply Today's Price times the number of shares held and subtract the selling commission from Module 2. Cell AI19 would be — +AH19*AA19-W17. This formula is then copied down the rest of the column.
6. Column AJ — this is a simple subtraction — Today's Value minus Yesterday's Value. So, Cell AJ19 would be — +AI19-AG19. Copy this formula down the rest of the column.
7. Column AK — an IF function is needed here to prevent a divide by zero error for the case where AG19=0, else subtract Today's Value from Yesterday's Value and divide by Yesterday's Value. Cell AK19 would be — @IF(AG19=0,0,(AI19-AG19)/AG19). Again, copy the formula down the rest of the column.
8. Column AL — this is another simple subtraction of Today's Value minus Purchase Value. So, Cell AL19 is — +AI19-AC19. This formula would be copied down the rest of the column.
9. Column AM — use an IF function to avoid a divide by zero error if AL19=0, else divide Overall Gain by Purchase Value. Cell AM19 would be — @IF(AL19=0,0,AL19/AC19). Now copy this formula down the rest of the column.
10. Column AN — again start with an IF function to prevent a divide by zero error if AD19=0, else divide Overall Change by Day's Held and multiply by 365. Cell AN19 looks like this — @IF(AD19=0,0,(AM19/AD19)*365). Then copy the formula down the column.
11. Cell AL11 requires another IF function in case AL10 would be zero, else subtract Yesterday's DJA from Today's DJA and divide by Yesterday's DJA. Cell AL11 looks like @IF(AL10=0,0,(AL9-AL10)/AL10).
12. Cell AL12 should have +AK31.
13. Cell AC31 will total column AC with a SUM function looking like this — @SUM(AC19..AC30). (Remember if you increase or decrease the module table size these SUM functions would have to be changed.)
14. Cell AG31 totals column AG with this function — @SUM(AG19..AG30).
15. Cell AI31 sums column AI with this function — @SUM(AI19..AI30).
16. Cell AJ31 also uses the SUM function — @SUM (AJ19..AJ30).
17. Cell AK31 — uses an IF function to check for AG31=0 to prevent a divide by zero error, else divide Change from Yesterday's Total by Yesterday's Value Total. Cell AK31 will be —@IF(AG31=0,0,AJ31/AG31).
18. Cell AL31 requires the SUM function — @SUM (AL19..AL30).
19. Cell AM31 uses another IF function to avoid a divide by zero error, else divide Overall Total by Purchase Value Total and subtract 1 from it. Cell AM31 will be — @IF(AC31=0,0,(AI31/AC31)-1).

This completes the construction of this Module.

I will discuss Figure 1 Tables next. First, the data for commission rates is fictitious. Be sure to contact the broker that the user is working with and find out their rates. USE THOSE RATES. Stock commissions are usually made up of a combination of fees. There is a flat fee, a percentage fee based on the value of the share of stock and/or transaction, and each transaction is subject to a minimum or maximum amount based on the number of shares purchased. I have picked some typical values that could be found from some broker.

The easiest way to determine a commission rate is to use the LOOKUP function with the tables shown in Figure 1 (of course you would have your rates in place of mine). Every LOOKUP function will consist of a search argument and a range of cells (plus a column offset for a program like LOTUS 1-2-3) that provides the location of the lookup table on the worksheet. The LOOKUP function for Cell N17, as an example, will lookup the commission rate based on the value of the purchase transaction. The search argument would be AA19*AB19 in this example. The location of the lookup table would be B13..C17 in this example. (The column offset for LOTUS 1-2-3 would be 1 in this case.) The function would return 0.003 in this example because the transaction exceeds the maximum amount of 56001. This, of course, means that 0.003 would be the commission rate that would be used in this example.

We calculate the commission value with another LOOKUP function for Cell Q17 to determine the flat dollar amount — @VLOOKUP(N17,E13..F17,1) — which provides 57 in our example. We will add the commission by multiplying the number of shares times the share purchase price times the 0.003 commission rate. Also, we must calculate the minimum and maximum commission determined by the number of shares purchased for the transaction. Cell P17 is where the minimum commission is calculated using an IF function — @IF(AA19<600,AA19*0.08,((AA10-600)*0.04)+48). I have assumed the minimum commission rate would be 8 cents per share on transactions up to 600 and 4 cents per share for the number of shares purchased beyond the 600 shares. So, the IF function tests if AA19 is less than 600 shares, and if less than 600 shares, the first value is used — calculating the rate of commission at 8 cents per share — and if greater than the 600 shares, the second value would be used — calculate the number of shares over 600 and multiply those shares by the 4 cents per share and add the 48 (600*0.08). In the example, with 100 shares times 8 cents per share the result would be 8.00. An IF function is used to check if there are 100 or more shares, and if so, multiply the number of shares times 45 cents per share, else use Cell O17. In this example, 100 shares times 45 cents per share results in 45.00.

Now we have three possible commissions for the transaction: the commission rate based on the value of the transaction, the minimum commission, and the maximum commission. Another IF function will determine the correct commission in Cell R17 —

@IF(P17>O17,P17,@IF(Q17<O17,Q17,O17)). Again with this formula, Cell O17 is the commission based on the value of the transaction, Cell P17 is the Minimum commission, and Cell P17 is the Maximum commission. The IF function make the decision: Is the Minimum commission greater than the value commission — the answer is NO in this example, so the second IF function asks: Is the Maximum commission less than the value commission — the answer is YES in this example, so the Correct commission Cell Q17 (Maximum commission) is the answer for the example.

The bottom portion of this Module has the Tables for determining the number of days since the purchase transaction. For TAX purposes, a stock held more than 6 months plus one day (current rules) is a long term capital asset and gains on long term assets are taxed at substantially lower rates than ordinary income. Therefore, knowing the number of days that the stock has been held is a FACT that should always be known to the investor. The complete set of tables can be eliminated if the reader is using one of the spreadsheet packages like LOTUS 1-2-3 that has DATE functions. I did not use these functions for this project, so that all readers could do this project with their software. I do not have the space to discuss DATE functions and their use, but I will use them in a future article and explain them at that time.

I will briefly cover the main item about these Tables. The Month Table shows the numbers assigned to each month. Next to the month numbers are listed the number of days that have elapsed since the first of the year up to the first day of each month. The Years Table gives the number of days that have elapsed each year. I have started with zero for January 1, 1975. (The reader can adjust the table if they have any stock holdings before that date.) Finally, leap year has to be included as shown. I will let the reader examine this numbering system to obtain an understanding of it or just use it!

Next, we will discuss Figure 2, the Stock Records Workarea which I call Module 2 (Did you make your "SPREADSHEET Preparation Form" by referring to my Figure 2?). Set the column widths that you selected OR the ones I have shown and enter the Titles and Headers used for the Module. Now enter the formulas into the cells. (I will use the cell "names" from Figure 2.)

1. Cell V9 has a formula that really looks complicated until you break it down into parts and examine it. (Remember that 1–2–3 has @LOOKUP function with a "column offset". Other spreadsheets do not use '@V'LOOKUP or the 'column offset', but their LOOKUP function will work with this project. Leave off the '@V' and the ',column offset'.) My Cell V9 — +AA14+@VLOOKUP(Z14,C24..D36,1) +@VLOOKUP (AB14,E24..F36,1) +@IF(Z14>2,@VLOOKUP(AB14, G24..H36,0). If the reader is using 1-2-3 with the DATE functions, this could be changed! The above formula uses a series of three LOOKUP functions. To assign a number to Today's Day it adds the number of the day of the month, the number of days between the first day of the present year and the first day of the month we are using, and the number of days that have elapsed since the beginning of the numbering system table, plus an adjustment for a leap year when necessary. The reader will have to study this OR take my word that the Tables will do the work.

2. Column N — will use a LOOKUP function and "absolute" cell addressing for finding the commission data. Cell N17 will have this formula — @VLOOKUP(AA19*AB19, B13..C17,1). Copy this formula down the rest of the column.

3. Column O — use another LOOKUP function and an addition calculation, Cell O17 will be — @VLOOKUP(N17, E13..F17,1)+ AA19*AB19*N17. Again, copy this formula down the rest of the column.

4. Column P — use an IF function for Cell P17 — @IF (AA19<600,AA19*0.08, ((AA19-600)*0.04)+48). This formula will be copied down the column.

5. Column Q — another IF function for Cell Q17 — @IF (AA19>=100,AA19*0.45,O17). Copy this formula down the balance of the column.

6. Column R — nested IF functions are used for Cell R17 — @IF(P17>O17,P17,@IF(Q17<O17,Q17,O17)). Also, copy this formula down the column.

The next five columns are very similar to those above, so I will just provide the formula to start the user out.

7. Column S — Cell S17 — @VLOOKUP(AA19*AH19, B13..C17,1).

8. Column T — Cell T17 — @VLOOKUP(S17,E13..F17,1) +AA19*AH19*S17.

9. Column U — Cell U17 — @IF(AA19<600,AA19*0.08, ((AA19-600)*0.04)+48).

10. Column V — Cell V17 — @IF(AA19>=100,AA19*0.45, O17).

11. Column W — Cell W17 — @IF(U17>T17,U17,@IF (V17<T17,V17,T17).

12. Column X — this takes a complicated looking formula to calculate the Number of the Purchase Day, but break it down into steps to study and it works out easily. Cell X17 — +L17+@VLOOKUP(K17,C24..D36,1)+@VLOOKUP (M17,E24..F36,1)+@I F(K17>2,@VLOOKUP(M17, G24..H36,1),0) and copy the formula down the rest of the column.

This completes this Module.

Constructing the three Modules for this worksheet is an exacting task, but otherwise it is rather simple. The results should be worthwhile. Next, enter the initial data into the Modules as follows:

1. Column Z–enter a three letter Stock Symbol for each stock held, one stock purchase per row.
2. Column AA–enter the number of shares of each stock held, one line for each stock purchase.
3. Column AB–enter the Purchase Price per share that was paid for each stock transaction, one transaction per row.
4. Column J–either enter or copy the Stock Symbol (same as column Z).
5. Column K–enter 1 or 2 digit month number for the purchase date for each stock transaction, one transaction per row.
6. Column L–do the same for the day number as the month number.
7. Column M–repeat for the year number.

Now, the worksheet is ready for the daily updates. Here is a list of the daily entries:

1. Cell Z14–enter the 1 or 2 digit today's month number.
2. Cell AA14–enter the 1 or 2 digit today's day number.
3. Cell AB14–enter the 2 digit today's year number.
4. Column AF–MOVE the values in the cells in column AH to

the cells in column AF. The COPY function could be used.

5. Column AH–enter Today's Share Price for each stock held in their cells in column AH.
6. Cell AL10–MOVE (could use COPY) Cell AL9 value to Cell AL10.
7. Cell AL9–enter Today's DJA value into Cell AL9.

The worksheet will recalculate all values.

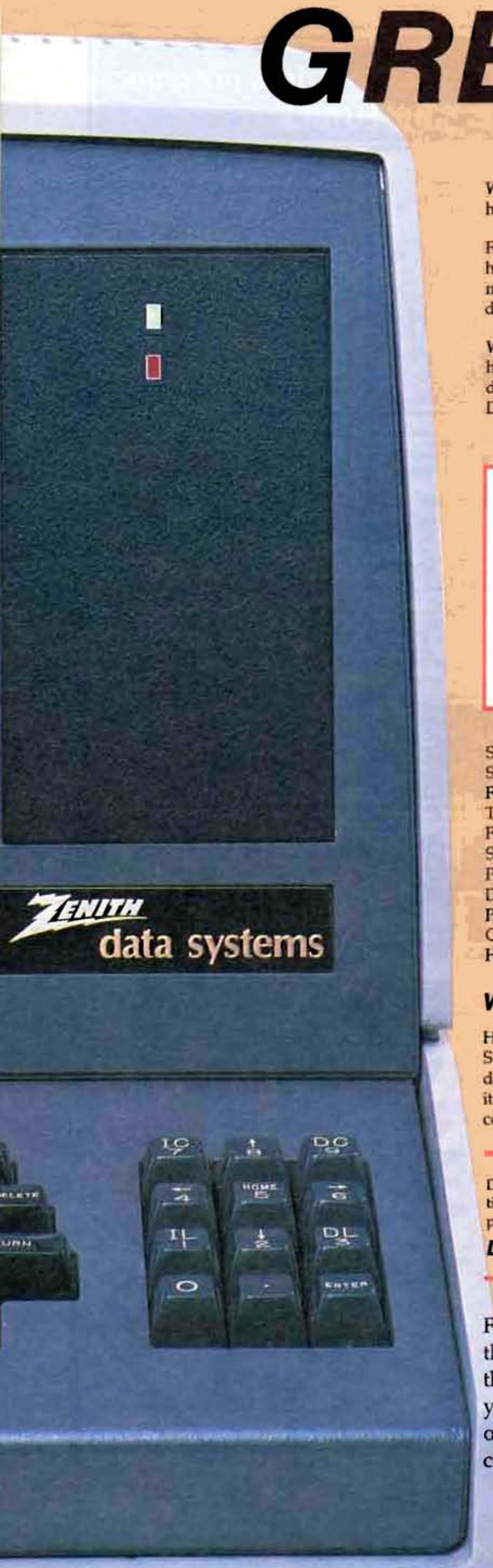
If the user has added some new stock purchase or has sold a stock holding, the appropriate cells in columns Z, AA, AB, J, K, L, AF, and AH should be changed. I am sure that the user will see why.

Thus, with a minimum of effort the investor can know just how well each stock holding is doing compared to the other stocks held and compare them to the performance of the overall market as given by the DJA! The user can use this worksheet as a starting point for modifications. One advantage of this worksheet is that it can be easily enlarged to meet the investor's needs. I would suggest that the user "protect" all cells with the global protection command if it is available, and then "unprotect" each cell or range of cells that are updated for daily data or for purchases and sales. If the user wants to get fancy, a page of instructions of how to use this project could be added. A Menu would be one way.

I hope that you find this project interesting and useful. I will be doing another "stock worksheet" with a database in a future article that could graph the "trend" of the user's stock holdings. Happy "SPREADSHEETING"!

*

GREAT DRIVES!



The H89 TWOET systems Just Got Hard to beat!

We've got a great idea for your H-88, 89 or 90. It's a dual internal half height drive system. Two of our half height 5¼" drives can replace your built-in disk drive, doubling your information storage capacity.

Floppy Disk Services provides you with everything you need. That's two double-sided, double density, half height drives in either 48 or 96 tpi format, all hardware, cables and power connector adaptors. And most important, you get easy, step-by-step instructions, in the Heath/Zenith tradition of good, clear documentation.

We've thoroughly tested the TWOET/Heath set-up. And now by popular demand, the new 10 megabyte hard disk system! We are now able to offer you the storage you want and need for a fair price! The hard disk system comes with all software needed to run with the 17 or 37 controller and/or the Magnolia Double Density controller.

NEW!

Model TWOET 455

2 Shugart SA-455 half height
48 tpi double sided
All hardware
Metal, shielded mounting plates
Data cable with chassis connector
Power "Y" connector
Complete documentation
Price....... $355.00

Model TWOET 55F

Two Teac 55F half height
96 tpi double sided
All hardware
Metal, shielded mounting plates
Data cable with chassis connector
Power"Y" connector
Complete documentation
Price....... $385.00

***HARD DISK
for
H-89
$995.00
Internal***

SA-860 DS/DD half hgt 8"	505.00 2@ 495.00 ea.
SA-455 DS 48tpi ½ hgt 5.25"	140.00
FDD-100-8 SS/DD 8"	125.00
Teac 55F 96tpi DS/DD half height	155.00
FDD-200-8 DS/DD 8"	***Sale 185.00***
Shugart 10 MB half height hard disk	725.00
PGS, HX-12 RGB color monitor	525.00
Data connectors of all types	***CALL***
Power connectors for all drives	***CALL***
Controllers	***CALL***
Hard disk Z150/160	Internal 750.00

Wondering what to do with your internal drive if you buy this system?

Here's the solution. If you purchase a dual half height system for your Heath computer from Floppy Disk Services, just include an extra $60.00 plus shipping and receive a single 5¼ case with power supply and data cable ready to receive your SIEMENS internal drive! the case with data cable is normally a $80.00 item. And the cable that comes with your TWOET system includes the external chassis disk I/O connector.

Due to production deadlines, this ad is 2 months old at time of printing. We encourage you to call for our latest pricing and catalog requests.

Dealer inquiries invited.

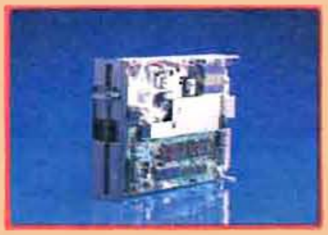
We accept cash, MasterCharge, Visa, personal checks (allow 10 days to clear the bank), wire transfers, money orders, purchase orders from government agencies, and approved businesses. Call for info.

Floppy Disk Services inc. has been supplying custom enclosures and disk drives to the hobby and professional market for 6 years now! We have specifically supported the Heath/Zenith community with a high degree of success. We care about you and your system integrity. That's why we have been a success in a time when many others are going out of business. We are proud to be a part of the Heath/Zenith community and will continue our support with new products and ideas.

H-88/89/90 and Z-150/160 are registered trademarks of Heath Corp.

GREAT PRICES!

SA-455



Shugart SA-455 half height 5¼ is the standard for excellence in the disk drive community. Backed by a one year warranty and manufactured by the leader of drive technology, Shugart Corp. Ready to ship at $140.00 ea. Compatible with ALL of the Heath/Zenith computers.

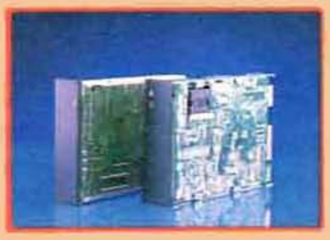
Z-150/160 Upgrade

We can upgrade your Z-150/160 with half height floppies, hard disks, 8 inch drive systems, clock modules, memory and much more! Glance over our ad and call toll free outside NJ for our latest catalog. Or call and ask our sales/technical experts for their upgrade suggestions...

Maynard and DTC

We can supply PC compatible upgrade boards for your Z-150, Z160 system. Hard disk controllers, memory upgrade, and space saving Maynard modular boards. As an example, you can install a Maynard Modular 5 & 8 controller that will run 5¼, 8 inch and optional boards such as a clock/calendar module, game port, serial port or parallel port. Call for details on the slick way to upgrade and save a slot at the same time...

SA-712 & MicroScience



Half height 10 megabyte (20mb available soon) hard disk drives. The MicroScience has a plated media and is available with a full size bezel option. The Shugart has an on board low current option and is backed by a one year warranty. It also has a full size bezel option. both drives are available for the H/Z-89, H/Z-150, H/Z-160. $550.00 each.

Irwin-110

Data loss! Its a terrible thought. Minimize your chance of lost programs and data files with the new 110 tape drive from Floppy Disk Services inc. It works off the standard 5¼ controller which means you save money over standard tape backups that you must buy the controller for. The data cartridge holds 10mb of data and can retrieve in streaming or individual file form. Software included. Available in many configurations. For the IBM-PC, Z-150 or Z-160 only. (CP/M software is being worked on). Tape unit $395 ea., tape cartridge $29.95 each...

Connectors

Floppy Disk Services inc. can supply custom data and power cables for your needs. We use only the best! UL rated or listed components, and all power connectors are crimped by our AMP electric certified crimper. No hand crimping to vary the degree of accuracy. Only the best! Floppy Disk Services inc. can make 1 or 1000 connectors to your specs, fast! Call and ask for details...

Special Sale!

Because of our enormous buying power in the industry we have made and excellent purchase of disk drives for your Heath/Zenith computer. Two Teac 55F 80 track double sided drives, in case with power supply and shielded data cable, your price $395.00. Single drive system, $245.00.

Enclosures!

We manufacture and stock enclosures for every disk drive on this page and for almost any other drive you are likely to find. Available assembled and tested with state-of-the-art power supplies, these enclosures house a wide combination of half- and full-height drives to complement most available computers.

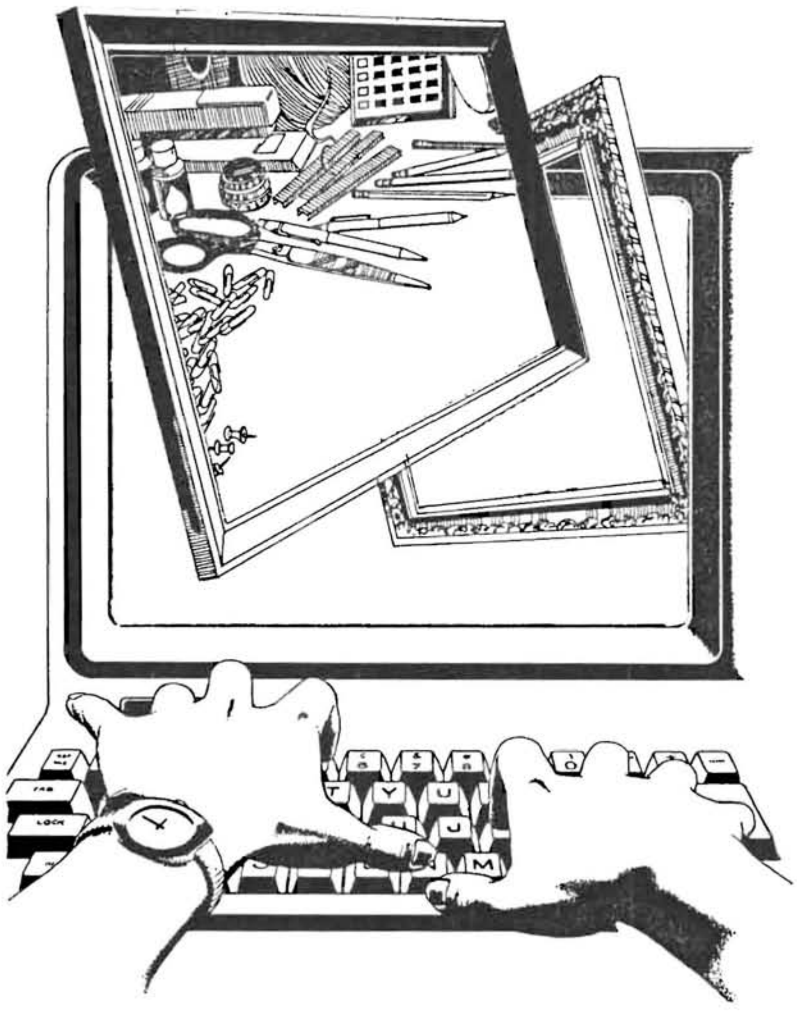
Toll Free Order Line:
(800) 223-0306

Tech Help or Info:
(609) 799-4440

39 Everett Drive Bldg. D, Lawrenceville, New Jersey 08648

FLOPPY
DISK
SERVICES™
INC.

49^{95}—



WHIZ

Memory Resident Utility For The H/Z-100

WHIZ is a multi-purpose utility for the H/Z-100. It remains resident while other programs are loaded and run, and is available at the touch of a key, even while your application programs are running. WHIZ functions are accessed through pop-up windows, which can be moved around on the screen by the user. The initial release of WHIZ incorporates the functions described below. Continuing development will provide future updates which add additional functions to WHIZ.

Context sensitive HELP - provides you with the information you need when you need it.

ALARM CLOCK - beeps once per second for 10 seconds at the desired time of day, no matter what else is going on at the time.

ASCII Table - displays decimal and hex equivalents of ASCII characters, including control characters, and the current graphics set. A boon for programmers.

NOTEPAD - 80 character by 40 line notepad lets you make notes to yourself which are always available. It also serves as a text editor, letting you read, write and edit small text files and view files of any length.

APPOINTMENT CALENDAR - provides a 10 line by 40 character record for each date until well into the 22nd CENTURY! Only active records are stored to minimize disk space usage. You can "step through" the date file by day, week, or month.

SCIENTIFIC CALCULATOR - Built-in, full floating point algebraic expression evaluator. Rather than a cute but inefficient image of a handheld calculator, WHIZ allows direct entry of complex algebraic/trigonometric functions for immediate evaluation. Available under both the Notepad and Appointment Calendar functions, the calculator supports square root, sine, cosine, tangent,arc sine, arc cosine, arc tangent, hyperbolic sine, hyperbolic cosine, hyperbolic tangent, base 10 log,natural log, and natural anti-log.

Requires H/Z-100 at least 192K RAM; MS-DOS 2.13 or higher; does not run under ZDOS 1.xx.

Memory Resident - WHIZ requires 70K to 110K (user specified) of available system memory. All functions work, even on a minimum memory allocation, just not all at the same time.

Software Wizardry
THE MAGIC TOUCH

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

HUG Price List

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

HDOS HARDCOPY SOFTWARE

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
885-1010	Adventure Disk H8/H89	10.00	4
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8
885-1031	Disk IV MUSIC H8 Only	20.00	25
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12
885-1068	Disk XII MBASIC Graphic Games	18.00	10
885-1088-[37]	Disk XVII MBASIC Graph. Games	20.00	14
885-1093-[37]	D&D H8/H89 Disk	20.00	16
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23
885-1113-[37]	HDOS Action Games H8/H89	20.00	23
885-1114	H8 Color Raiders & Goop	20.00	23
885-1124	HUGMAN & Movie Animation Pkg	20.00	41
885-1125	MAZEMADNESS	20.00	41
885-1130	Star Battle	20.00	47
885-1133-[37]	HDOS Games Collection I	20.00	59
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32
885-8022	HDOS SHAPES	16.00	45
885-8026	HDOS Space Drop	16.00	49
885-8032-[37]	HDOS Castle	20.00	59

CP/M

Part Number	Description of Product	Selling Price	Vol. Issue
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

ZDOS

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
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
885-5001-37	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 Bkgrnd 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

Part Number	Description of Product	Selling Price	Vol. Issue
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
885-8029-37	ZDOS FAST EDDY	20.00	53

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

Part Number	Description of Product	Selling Price	Vol. Issue
885-3012-37§§	ZDOS HUG Editor	20.00	52
885-3014-37§§	ZDOS/MSDOS Utilities II	20.00	54
885-3016-37§	ZDOS/MSDOS Adventure	10.00	57
885-3020-37§	MSDOS HUG Menu System	20.00	62
885-3021-37§§	ZDOS/MSDOS Cardcat	20.00	63
885-3024-37§	ZDOS/MSDOS 8080 To 8088 Trans	20.00	64
885-3025-37§§	ZDOS/MSDOS Misc. Utilities	20.00	64
885-3029-37§§	ZDOS/MSDOS HUG Bg. Print Spool	20.00	66

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

PC/IBM COMPATIBLE

Part Number	Description of Product	Selling Price	Vol. Issue
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
885-8033-37	MSDOS Fast Edit	20.00	62

PROGRAMMING LANGUAGES

HDOS

Part Number	Description of Product	Selling Price	Vol. Issue
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	20.00	59
885-1134	HDOS SMALL-C Compiler	30.00	63

CP/M

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
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

Part Number	Description of Product	Selling Price	Vol. Issue
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

ZDOS

Part Number	Description of Product	Selling Price	Vol. Issue
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

Vectored to Page 80 ☛

HUG NEW PRODUCTS

HUG P/N 885-1247-37 CP/M-85
HUG Background Print Spooler $20.00

Introduction: The HUG Background Print Spooler (HBPS) is a program that allows you to use 64k of the memory above CP/M-85 as a print buffer. With HBPS loaded, when a program sends text to the printer, it does not go directly to the printer, but into the HBPS buffer. HBPS then takes the characters from the buffer and sends them to your printer. Since HBPS can accept characters into its buffer more rapidly than the printer can accept characters, your computer can complete a print operation faster than without HBPS, and go on to other tasks while HBPS is printing characters from its buffer. HBPS gives you the advantages of a hardware print buffer without the hardware.

Requirements: HBPS requires an H/Z-100 series computer, at least 192k of RAM, and the CP/M-85 operating system, version 2.2.103. Note: HBPS will not work with earlier versions of CP/M-85.

This disk contains the following files:

README	.DOC	HBPS	.COM
HBPS	.DOC	SETSP	.COM
SCR	.COM	UNSCR	.COM
HBPS	.ASM	SETSP	.ASM
SCR	.ASM	UNSCR	.ASM

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

HBPS.COM — This is the HBPS spooler program, ready to be copied to your system disk and loaded into memory. HBPS normally uses the first 64k segment of memory above CP/M-85 for its buffer, but you can make it use a higher segment, in case you have another utility that requires memory directly above CP/M-85.

Note: Some programs are slow in sending characters to a printer. If your printer is faster than a particular program, using HBPS or any other print buffering method will not result in a time savings. The slower your printer is, the more time will be saved by using HBPS. The most time will be saved when you PIP large files to LST: to print them.

HBPS.DOC — Instructions for using HBPS.

SETSP.COM — This is a program for controlling HBPS once it is loaded. It allows you to empty the HBPS buffer to halt printing, to disable HBPS so that printer output of programs goes directly to the printer, to re-enable HBPS, or to unload HBPS from memory.

SCR.COM — This is a modified version of the screen dump program originally released on disk 885-1237[-37]. This version can reside in memory along with other memory resident utilities, such as KEYMAP (885-1245-37). It is possible to have KEYMAP, SCR, and HBPS all resident at the same time.

UNSCR.COM — This program is used to unload SCR from memory.

HBPS.ASM, SETSP.ASM, SCR.ASM, UNSCR.ASM — The assembly source code for the above programs.

Table C Rating: (2),(4),(10)

Note: The following other versions of HBPS are available:

885-3029-37 — HBPS for Z-DOS or MS-DOS, for H/Z-100 and H/Z-150 and similar computers. You can assign any amount of your memory from 4k to 512k to HBPS for use as a print buffer.

885-5009-37 — HBPS for CP/M-86, for H/Z-100 computers. Works with either Heath/Zenith or Watzman CP/M-86, and lets you assign 4k to 512k of memory to the print buffer.

HUG P/N 885-6004-37 MS-DOS
CheapCalc $20.00

Introduction: CheapCalc is a minimal but useful "spread sheet" program that can introduce you to spread sheet computing at little cost. With CheapCalc, your computer screen becomes the window to a large worksheet onto which you can write mathematical problems and have them solved almost immediately.

Requirements: An H/Z-150 or other PC-compatible computer with MS-DOS and at least 128k of system RAM.

The following files are included on this disk:

README	.DOC	CC	.DOC
CC	.EXE	CC	.HLP
SAMPLE	.CAL	CC	.BAS

Program Authors:
Original program by William V. R. Smith
First Heath version by Bob McFarland
H/Z-150 version by P. Swayne, HUG

CC.DOC — This file contains instructions for using CheapCalc.

CC.EXE — This is the compiled, ready-to-use CheapCalc program, which runs by itself without BASIC or any other language. It supports a worksheet containing 40 rows and 15 columns. Operations supported are addition, subtraction, multiplication, division, powers, and SUM. Fractional powers can be used for roots (25^.5 for the square root of 25). The SUM function allows you to add rows or columns of numbers.

All calculations are done in double precision arithmetic, with precision equivalent to that supported by GW-BASIC. Numbers can be displayed in 4 formats: floating, dollars-and-cents, integer, and graphic (a line of asterisks represents each number). In the dollars-and-cents format, rounding off to the nearest cent is done to correct possible binary floating point errors.

CheapCalc lets you enter comments as well as formulas and numbers into the worksheet, and the width of a column can be adjusted to any width from 4 to 30 characters to accomodate your entries. The contents of a "cell" on the worksheet can be copied or replicated to any other cell. You can save your worksheet to a file, load previously saved worksheets, and print all or part of a worksheet on your printer.

CC.HLP — This file contains instructions for CheapCalc which are displayed on the screen when you use the Help command while running CheapCalc. You can get help at any time without destroying your worksheet data.

SAMPLE.CAL — This is a sample worksheet that will help you to understand the workings of CheapCalc.

CC.BAS — This is the source code for CheapCalc. It can be run using GW-BASIC, but the performance is considerably degraded compared to the compiled version, and the arrow keys do not work when you use the interpreter to run it.

Table C Rating: (1),(3),(10)

Note: These other versions of CheapCalc are available:

885-1131[-37] — HDOS CheapCalc, for H8, H/Z-89,90.

885-1233[-37] — CP/M CheapCalc, for H8, H/Z-89,90, and CP/M-85, CP/M-86 (on Z-100).

885-3006-37 — Z-DOS CheapCalc, for H/Z-100 and Z-DOS or MS-DOS.

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.

HUG P/N 885-6005-37 MS-DOS
Skyviews $20.00

Introduction: Skyviews is a program that plots the positions of major stars (4.0 magnitude or brighter) and the sun, moon, and planets. In addition to providing a graphic view of the celestial objects, Skyviews provides right ascention-declination, azimuth-elevation, and other information about the objects that is useful to astronomers. The program will mark major constellations on the screen to aid in locating them.

Requirements: A H/Z-150 or similar computer, MS-DOS, and at least 192k of system memory. The high resolution monochrome graphic mode is used.

Program Authors: Eugene L. Davis. Z-150 version by P. Swayne, HUG.

This disk contains the following files:

README	.DOC	KYVIEWS	.DOC
SKYVIEWS	.EXE	PLANET	.DAT
CSTNM	.DAT	BRIGHTEST	.DAT
OBSLOC	.DAT	SKYVIEWS	.BAS

SKYVIEWS.DOC — Instructions for using SKYVIEWS.

SKYVIEWS.EXE — This is the compiled ready-to-use Skyviews program. When you run it, it will prompt you for the date, local standard time, time zone, longitude, and latitude of the observer. You have the choice of calculating the positions of the sun, moon, and planets; or the sun, moon, planets and major stars.

PLANET.DAT, CSTNM.DAT, BRIGHTEST.DAT, OBSLOC.DAT — These are data files used by SKYVIEWS when it runs.

SKYVIEWS.BAS — This is the BASIC source for SKYVIEWS. It can be run using GW-BASIC, but at a reduced performance level when compared to the compiled version.

Table C Rating: (10)

Note: Skyviews is also available for H/Z-100 computers, as part no. 885-3015-37. This version provides even more information, by using color to indicate star magnitude.

Watch next month's REMark for a program to help you locate Halley's Comet!

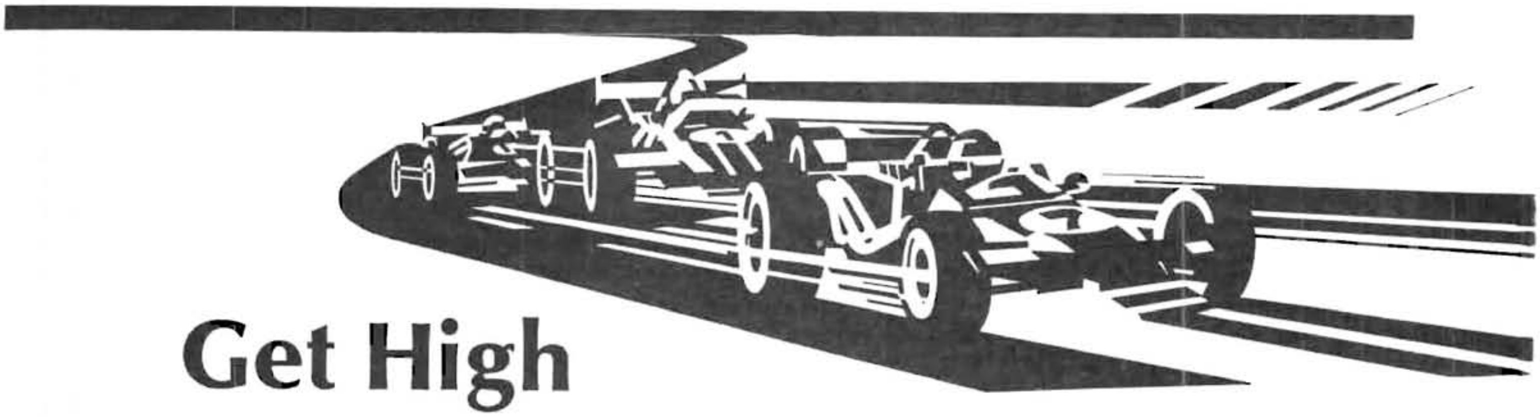
ORDERING INFORMATION

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.



Get High On High Speed!

A Review Of The US Robotics Courier 2400 High Speed Modem

Jim Buszkiewicz
HUG Software Developer

Communicating via Ma-Bell at 2400 baud is like trying to walk through a cow pasture on a misty morning without slipping! However, U.S. Robotics has done it, without slipping, getting wet, and smelling like a rose!! Their recent release of the Courier 2400 high speed modem proves it.

The manual received with this modem is most impressive. It comes as part of an attractive hard covered binder which is the same height as all of the newer Zenith software manuals and can easily be used as a quick reference manual. Because of its metal spiral binding, the manual stays open to the page that was referenced. This manual appears to thoroughly cover both the 'hows' and 'whys' of making this modem work with your system. It also goes beyond the actual product itself and covers the concepts of modem communications. All software commands and modem switch settings are explained in detail. Included with the manual in one of the inside binder pockets is the warranty card, product registration card, and a quick reference card containing all the commands the unit is capable of.

Anxious to see this baby fly, I left the switch settings as recommended by the factory, and connected the modem to the same cable that was previously hooked to a Hayes Smartmodem 1200. Since the USR uses the standard 'AT' command set, I did not notice any difference in operation in either 'originate', OR 'answer' mode. What a let down. I guess I expected color pictures on my monochrome monitor! Ok, so it works like my Hayes at 300 and 1200 baud, let's take her out to the open road, and floor it! Changing from 1200 to 2400 baud is just as easy as changing from 300 to 1200: set the baud rate in you terminal (or terminal program), and type 'AT'. The Courier switches automatically and responds with 'OK' (hmm, just like the old Hayes). Let's see... who could I call that's running at 2400 baud? I could only think of one bulletin board system: CBBS in Chicago. Quick, type in "ATDT 13125458086". Rats! Busy! Try again, and again, and finally. The modem responded "CONNECT 2400"!! Wait! I can't read that fast!! The speed difference was almost as impressive as going from 300 to 1200.

Since that first call, I've been able to communicate with many different systems using various manufacturers' modems. The Courier performed flawlessly with each. I was also able to communicate with the local Tymnet at 2400 baud although I don't know whose modem they were using. I have also had another Courier 2400 user call my system at 2400 baud, and download some files using XMODEM protocol. He reported "..had no problems, and only took a couple of minutes for a 20k file..". I have also discovered that U.S. Robotics maintains its own bulletin board system called FIDO ("..sit UBU sit"). Its purpose is to provide a way of exchanging ideas and solving problems as well as a place to test your new 2400 baud modem. FIDO can be reached at: (312) 960-5928. The MS-DOS version of HUGPBBS is also up and running at 2400 baud here in my office. You can call it at (616) 982-3956.

The best feature, I've found, which this modem has, is its ability to automatically switch baud rates without user intervention. In the local, or 'originate' mode, the modem 'adjusts' itself to the host terminal (or computer) baud rate when the 'AT' (attention) command is sent. In the answer mode, the modem sets itself to the baud rate of the caller. This automatic switching is for all three baud rates, 300, 1200, and 2400 baud. NOTE: It is my understanding that some 2400 baud modems (such as the Hayes) require some sort of operator intervention or programming to switch over to 2400 baud.

Some other nice features I've found on this unit is its ability to respond to both upper and lower case commands. Also, if a manual is not readily handy (or you lose it), typing 'AT$' will bring up a detailed 'help' screen showing what commands are available. Typing 'ATS$' will bring up another 'help' screen showing how each of the user settable 'S' registers are presently configured. A third 'help' screen, brought up by typing 'ATD$', shows all of the different dialing commands available. If you forget what command to issue for the 'help' screen, a list of commands is also available on the bottom of the cabinet! Boy, USR thought of everything!

Two other convenient features are the 'slide' type speaker volume control on the side of the unit, and the user settable 'DIP' switches can be easily accessed from the bottom of the cabinet. Besides the 10, user settable, DIP switches on the bottom of the unit, there is an additional group of 4. This quad, when switched, reverses the functions of RS-232 pins 2 and 3 (send and receive data lines). This allows the modem to be connected to either a DCE or DTE connector without having to resort to a "null-modem" cable.

The modem's backplate contains the power switch, RS-232 connector, and two modular telephone jacks: one for the telephone unit, the other for the phone line. This feature, of course, saves you the cost of a "Y" plug for the modular outlet. The front of the modem contains 9 leds, the first eight of which indicate the same functions as the Hayes Smartmodem 1200: High Speed, Auto Answer, Carrier Detect, Off Hook, Receive Data, Send Data, Terminal Ready, and Modem Ready/Power On. The ninth led indicates Analog Loopback.

At A Glance

Product Name:	Courier 2400
Company:	U.S. Robotics Inc.
Address:	1123 West Washington Blvd. Chicago, Il. 60607
Telephone:	(312) 733-04997 or (800) 342-5877
Price:	$699

It used to be that if you wanted a pseudo industry standard in modems, you bought one with a 'wrap-around' aluminum case. Not any more! U.S. Robotics has taken the 'bull by the horns' (from the aforementioned cow pasture) so to speak, and is setting new industry standards with their Courier 2400.

*

Heath/Zenith Users' Group

2400 Baud Patch

Jim Buszkiewicz
HUG Software Developer

The CP/M-86 version of HUGPBBS is capable of supporting a 2400 baud modem such as the U.S. Robotics Courier 2400. This can be accomplished with a single byte change in the program. The trade off is that you lose the 110 baud capability, but that modem doesn't support that baud rate anyway. Make a copy of the program HUGPBBS.CMD from the original distribution disk. On this copy place DDT86.CMD. Boot that disk and do the following:

```
A>DDT86                        <—execute DDT86
-RHUGPBBS.CMD                  <—read in the old file
  START      END
 xxxx:0000 xxxx:E37F           <—ignore all these x's
-S1223                         <—substitute memory at 1223
 xxxx:1223 73 7B               <—change this byte from 73 to 7B hex
 xxxx:1224 E8                  <—type a period to exit this mode
-WHUGPBBS.CMD                  <—write the new modified file to disk
-.C                            <—and exit by typing control-C
A>
```

This new version of HUGPBBS.CMD will now work at 2400 baud.

If you have the source code, it can also be changed by changing the constant EPB110 in the beginning of the program from 3 to 11.

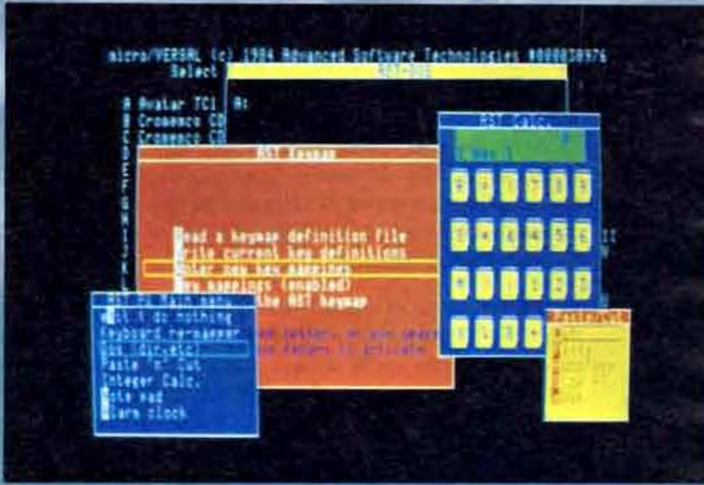
```
EPB110    equ  3    <—change this 3 to 11 in the source code
```

Now reassemble and gencmd the source. Your new HUGPBBS.CMD file will now support 2400 baud.

*

Genie™

Software Magic from Advanced Software Technologies



Shown are Genie windows as they appear on the Z-100. From the left: The Genie main menu, the Genie keyboard macro generator the Genie DOS, Calculator, and Cut and Paste menu. In the back is our micro/VERSAL disk to disk file transfer program, running as a regular application.

	Sidekick	Spotlight	Genie
Key macros	69.95	No	**Yes**
DOS services	No	Yes	**Yes**
Cut and Paste	No	No	**Yes**
Notepad	Yes	Yes	**Yes**
Ascii table	Yes	No	**Yes**
Calculator	One	One	**Two**
Calendar	Yes	Yes	**Yes**
Rolodex	Yes	Yes	**Yes**
Alarm	No	No	**Yes**
Load separately	No	No	**Yes**
	$120	$150	**$49.95**

AST's Genie is always there with you. With a simple keystroke it appears, and is at your command to help you, and make your work easier. Genie can move data between your applications, take notes, schedule appointments, and much more. Genie helps with everything from BASIC to Lotus 1-2-3.

Genie is available for the Zenith Z-100 and IBM-PC compatibles. Genie requires MS-DOS version 2 or above.

Special Introductory Offer:

$49.95

plus $4.00 shipping and handling
New York residents add 8.25% sales tax

To order send MC/VISA check or M.O. to:

ADVANCED SOFTWARE TECHNOLOGIES

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

Sidekick, Spotlight, MS-DOS, Lotus 1-2-3, IBM-PC, and Z-100 are trademarks of Borland International, Software Arts, Microsoft, Lotus Development, IBM, and Zenith.

H/Z-100
Remote Keyboard Adapter

Part Two

Tim Ross
1716 S. Solano, Apt. 30
Las Cruces, NM 88001

In Part 1 of my two-part series I explained the operation of the Remote Keyboard Adapter. In this article I will explain how to construct the adapter, supply a parts list, suggest possible sources for parts, and I will describe how to send me your 2716 EPROM which I will program and return to you with a program listing.

Construction

The Encoder

Both the keyboard and encoder sit inside the ET-100's keyboard enclosure. The keyboard is a perfect fit. Installing the encoder requires a few tricks. I built the encoder on a Vector 8001 circboard. The integrated circuits are mounted towards the rear of the board to allow maximum clearance between the integrated circuits and the underside of the keyboard. The keyboard is inclined inside the enclosure so the front of the keyboard is lower than the rear.

In Part 1 of my series I recommended that you have a high degree of skill in wire wrapping and soldering. Now you'll see why.

The encoder uses wire wrap sockets. The pins of these sockets are about an inch long. I cut them to 1/4 inch, again for clearance. Things start getting sticky. When wire wrapping, it is customary to strip one inch of insulation from the wire and then wrap an IC pin or square post. For this project I had to limit the height of a completed wrap. Most of the IC pins require two levels of wrap. When the pins protrude 1/4 inch below the circuit board, wrap height is restricted. I would strip about 1/2 inch of insulation from a wire, then wrap each connection using a manual tool. Once the project was completed and working properly, I soldered each and every connection. Planning is essential in laying out your wires. Try not to place three wraps on a single post. The IC pins are too short for three levels of wrap and there is not enough room inside the enclosure to allow longer wire wrap pins. Be tidy in the placement of your wires, not allowing a wire to touch any posts as it traverses the circuit board. Those square posts will cut through the insulation. When a post is heated up to solder a connection, the insulation of a neighboring wire which happens to touch the post will melt in an instant, resulting in an unexpected and undesirable extra connection. Short circuits are a no-no!

The encoder circuit board does not use the ET-100 enclosure's existing PC board mountings. I removed all of these stand-offs to prevent snagging wires and then mounted the circuit board with .3" plastic spacers, screws, washers, etc. The board is mounted towards the rear of the enclosure and is centered left-to-right. If you used the original PC board mountings there is a good chance that excessive strain would be put on the keyboard's ribbon cables, causing erroneous key entries. I discovered this the hard way.

Right angle male connectors are mounted on the encoder which mate with the keyboard's ribbon cable connectors. Molex calls these .100" Center Wafers, 4094 Series, Right Angle. The encoder requires one 20 circuit, one 10 circuit, and one 6 circuit .100" Center Wafer (right angle). The parts list will give specific Molex part numbers. Mount the 10 circuit Wafer (J107) closest to the rear of the circuit board, pin 10 is the rear-most pin. Leave a gap of .4" and then mount the 20 circuit Wafer (J105). Pin 1 is at the front of the circuit board. See the encoder photograph. These right-angle Wafers are not suitable for wire wrapping. I could not locate wire-wrap right-angle .100" center connectors. I soldered these Wafers to the circuit board, placed a wire-wrap pin (Vector T44) in the next circuit board hole, and then soldered a 30 gauge jumper from the Wafer pin to the wire-wrap pin. I was then able to use my short-wrap and solder technique to connect these Wafers to the encoder circuitry.

The six-pin Wafer is J1 and connects power and signal lines between the encoder and the decoder. This Wafer requires a mating connector (Molex offers a connector housing and individual terminals) and six 12 inch 22 gauge wires which are soldered to an AMP modular jack which mates with the coiled cord. Take care when soldering the six wires to the modular connector. It is fragile and you should be certain that when the enclosure is buttoned up these wires do not wiggle at the modular connector. I drilled two holes in the back of the enclosure top and fed a small cable wrap through the holes and around the six wires, providing a strain relief. The wires, from the strain relief to the connector, must be immobile. This strain relief is not visible in the encoder photo, it is obscured by the ten circuit ribbon cable. Look closely at the encoder photo and you will see the AMP modular connector glued to the side of the keyboard enclosure. I used auto body putty for this task. When the body putty cured it shrank a bit and detached itself from the enclosure so I used a drop of super glue to stick it in place.

Resistors and capacitors are mounted using Vector T44 and T42-1 terminals. To conserve space, the resistors connected to J107 pins 3 through 10 are soldered directly to the T44 terminals which are adjacent to the Molex Wafer. The Molex Wafer is connected to this post with 30 gauge wire. See above. The Z80 interrupt jumpers use Vector T46 pins. The tops of these pins are even with the top surface of the Z80.

Another approach to the encoder might be to use a different circuit board layout which would not require the cable from the AMP modular connector to the six-pin Wafer. The six pin Wafer could then be deleted. A right angle AMP modular connector, AMP part number 520250-3 (similar to the light pen connector) would mount directly to the circuit board and still protrude through the enclosure's connector hole. Of course, the circuit board would have to be mounted at the correct height to allow the AMP connector to mate with the rectangular hole in the enclosure. The Molex Wafers and the IC's might have to be relocated. A different style circuit board may be required. I will leave this alternate encoder layout to you guys. My prototype is an exercise in compromise. Mine is not necessarily the best way, but it definitely works.

The Decoder

I wanted the adapter to fit any HZ-100 with any hardware configuration. My HZ-100 is a bare-bones model with only one S-100 slot occupied. Some of you undoubtedly have every S-100 slot filled with memory boards, a hard disk controller, a floppy controller, and a modem. The prototype does not occupy an S-100 slot.

If you are dissatisfied with my configuration you may want to build the decoder on an S-100 board, but be aware of three things. First, the decoder will draw DC power from the S-100 bus, no S-100 signal lines are used. Take a look at your floppy drive controller to see how +5 volts is obtained. Second, you will need ribbon cables long enough to reach from the card cage to the mother board's keyboard connectors. I have never seen the inside of an H/Z-100 with several S-100 slots filled so I cannot presume to advise you on ribbon cable length, routing, or connector configuration. If you do go this route you may want your ribbon cables to have connectors on each end. These connectors may have to face opposite directions when the cable is laid flat on a table. Third, you will have to mount your modular connector somewhere. Be creative.

On to the decoder prototype. I built the decoder on a Vector 8003 circboard. The placement of LED's and the AMP modular connector seemed to rule out a bus-type circuit board. Radio Shack sells a board for 1/4 the price of the Vector board which should work, but be sure the size of the board's drilled holes will match the terminals you use for component mounting. Component layout on the decoder is a bit more relaxed when compared to the encoder. Use the whole board, spread your chips out, kick back, and breath easy. But not too easy. With my prototype, the circuit board is centered in the alphabetical / function key cutout using the space bar's opening as a guide. This places the rear third of the board above the H/Z-100's video board. Adequate clearance must be maintained between the video board and the decoder IC pins. I used my short-wrap and solder technique on the decoder also. With the decoder attached to a piece of plexiglas with 1/2" spacers I can slip my index finger between the video board and the decoder. It follows that the IC socket pins can be longer than 1/4". I leave it to you to determine an acceptable pin length, but remember, standard length IC socket pins are too long. I soldered the AMP modular connector directly to the decoder board. This requires drilling two 1/8" holes and bending the connector's pins just a bit to fit the .100" center holes, but it will fit.

I soldered one foot long Molex ribbon cables with Molex connectors to Vector T42-1 terminals: a row of twenty terminals, a gap of .4", then a row of ten terminals. I then inserted Vector T44 terminals in the holes adjacent to these T42-1's, jumpered the terminals together as I did with the encoder, and was able to wire wrap and mount resistors as needed. The Molex ribbon cable is more rigid than the ribbon cable attached to the keyboard. This allows the cable to be bent away from the sharp protruding pins present on the video board. The cable is self-supporting. Pierced insulation is a no-no. The ribbon cables are positioned as follows. The twenty circuit cable (P105) mounts towards the rear, pin 1 being rearmost. The ten circuit (P107) cable comes next, pin 10 towards the front. When the ribbon cables are stretched out, the connectors point down. See the decoder photo.

When you purchase your ribbon cables and connectors, see if you can have the connectors installed by your dealer. Doing it yourself is tedious. A special machine can do it in five seconds. The dealer may have another tool to strip insulation from the end of the cables. If he has it, use it!

The test circuit requires further explanation. At the lower right of the decoder schematic I have drawn a 74LS04 with dotted lines showing signal flow. These dotted lines are actually inverters which drive the signal line's associated LED. Take special note of the buses feeding these 74LS04's because I have provided you with a means of monitoring the data present in the 74LS164, the 74LS273, and the 74LS174. Arrange your LED's in rows: a row of eight, a row of seven, and a row of six. Place these LED arrays in such a way that you can readily distinguish between the different words or buses. Now we come to a case of "Don't do what I do, do what I say." I mounted the LED's bass-ackwards so that the MSB is on the right and the LSB is on the left. The first few times I tried to monitor data I had to stand on my head. Take a tip from me and place the MSB on the left so when you view your LED's you can readily see what is happening. You may want to use bar graph LED arrays or alpha-numeric types with their associated drivers. I leave the choices and the design considerations up to you. I bought discrete LED's from Radio Shack. Twenty LED's cost $1.98.

When I completed the decoder I needed a means to mount the board in the HZ-100 and also cover the large hole left by the removed keyboard. I took the HZ-100's plastic cabinet (the manual refers to it as the main base) to a window repair shop and had a 3/32" thick piece of clear plexiglas cut which fits the 17-5/8" by 5-1/4" depression on the top of the cabinet. They even rounded the corners for me. Total cost for this service was $2.25. I placed the plexiglas on the cabinet and drilled 1/8" holes in the plexiglas slightly inside the corners of the keyboard cutout (near the

RESET, FAST REPEAT, UP ARROW, and ENTER keys). I then super-glued spacers to the corners of the decoder board, drilled holes in the board using the spacers as a guide, and made a plexiglas / cabinet / decoder sandwich. I centered the decoder in the keyboard cutout and scribed the outline of the AMP modular connector on the surface of the plexiglas, then drilled and filed this small area until I had a nice fit between the connector and the plexiglas. With the decoder held in place by the connector, I then drilled four holes in the plexiglas, using the decoder's spacers as guides. I used 1/2" spacers with a small washer stuck to the top of each. This provides a height equal to the height of a ridge molded into the AMP connector, so the decoder board can be screwed down nice and tight without flexing the board. A gap is left between the IC's and the plexiglas to allow for air flow. To attach the plexiglas to the cabinet I drilled a 1/8" hole in four small squares of circuit board material. I placed these on the underside of the cabinet at the corners of the cutouts and used screws and nuts to cinch the plexiglas in place. No holes are drilled in the cabinet itself.

Decoder Power

I attached 18 gauge wires to the decoder's power planes, red for +5 volts and black for ground. I crimped insulated lug connectors to the ends of these wires. The +5 volt line has a female lug and the ground line has a male lug. These lines are therefore polarized and an improper connection is not possible.

I draw power from the DC power supply by using Tap-In Squeeze Connectors, the type U-Haul uses when they hook a trailer's lights to a car's electrical system. These connectors or splices are attached to the +5 volt and ground wires near P101. Use 18 to 14 gauge wire or the splices won't give a good connection. Crimp insulated lug connectors to these spliced wires, again polarized. Make sure that your power leads are long enough to enable you to easily install the decoder.

Perhaps you are a bit squeamish about splicing into the power supply's wires. There is an alternate method. I installed a 7805 regulator and heat sink to a Vector S-100 board and drew DC power for the adapter from the 7805. Both of these methods work. If your S-100 bus is not filled, I highly recommend the latter method. Look at your floppy drive controller for more insight.

The Coiled Cord

I have not assigned pin numbers to the AMP modular connectors. Make your connections between the encoder and decoder as follows. Attach the modular connectors to the enclosure and decoder. Attach the coiled cord, then run the cable from the encoder's modular connector to the encoder circuit board and use an ohm meter to establish pin numbers. You will find that the coiled cord's connectors are reversed, that is, pin 1 at one end will be pin 6 at the other end. Alternate your signal lines through the coiled cord as follows: +5 VOLTS, DATA, GROUND, DATA CLOCK, GROUND, and SHIFT CLOCK. Alternating between signal lines and power lines will aid in shielding your coiled cord.

Bypass Capacitors

I have not drawn bypass capacitors on the schematics to minimize clutter. You should attach a .1 uF cap between +5 Volts and ground for every integrated circuit. I located mine on the underside of the boards. If you use tantalum capacitors, install them correctly, tantalum capacitors should have the '+' lead connected to +5 volts.

Testing And Problems

Check all connections, looking for shorts, missing components, and missing wires. Connect your adapter up to a 5 volt DC power supply and run the diagnostic program as follows. Place the interrupt jumper in the 'B' position, connect the power jumper to the LED driver IC's, connect your two modules together with the coiled cord, turn the power supply on, and watch the LED's. The sequence of events is as follows. Upon applying power, all of the LED's will be off. The diagnostic will write a zero to RAM address 1000H, read it, serially transmit at a rate of one bit every 1/10 second until 8 bits have been sent, then send a SHIFT CLOCK to send the code to the appropriate buffer. This code will remain for about a second and then 01H will be written to RAM address 1001H, read, transmitted, etc. The codes sent to the decoder will range from 00H to FFH, while RAM addresses will range from 1000H to 107FH. The diagnostic takes quite a while to step through the entire sequence. You should see each and every bit serially shifted into the decoder and then see the code shifted into the character buffer if the MSB=0 or the control key buffer if the MSB=1.

If you don't see data being serially shifted into the input register, then a problem could exist in the encoder's wiring. Check the Z80, the 2716, the 6810, the 74LS139, the 74LS74 and the 7406. Check the coiled cord connections. Check the decoder's input circuitry, the 74LS14 and the 74LS164. Check the LED's, resistors, and 74LS04's. Check both power-on clear circuits. Check the +5 volt and ground wiring.

If data is shifted in but not sent to the character key buffer, check the encoder's 74LS139 and the 7406. Then check the coiled cord connections. Next look at the decoder's 74LS14, the 74LS08, the 74LS273, and the associated LED's and drivers. Check the power-on clear circuit.

If the input section and the character key section operate correctly but nothing reaches the control key buffer, insure the input register's MSB LED is lit. Then check the encoder's 74LS139 and 7406. Check the coiled cord connections. Check the decoder's 74LS14, 74LS08, 74LS174, the power-on clear circuit, and the control key LED's, drivers, and resistors.

Did you forget to place the resistors in the decoder's input circuit? Are any of the encoder's modular connector pins shorted?

Once your adapter passes the diagnostic and you are certain that there are no wiring problems, turn off your power supply, place the Z80's interrupt jumper in the 'A' position to disable interrupts, connect the keyboard, and start typing. Each and every key should have its own code. Watch the LED's. The LED's should show the last code sent to the decoder. If no keys are pressed, all LED's should be off (the MSB of the input register may be high if a control key buffer clear was the last transmitted code). If CAPS LOCK is depressed, the control key LED's should show this. If a character key is depressed and held, the input LED's and character key LED's should be lit. If you release a key, the LED's should turn off. Refer to your hardware manual's keyboard matrix figure and my first article to determine the key codes. Test every key. Only one character key code will be sent to the decoder at a time. Any combination of control keys can be sent to the decoder. If you experience problems, check the keyboard scan circuitry: 74159, 74LS139, 7406, both 74LS240's, ribbon cables, etc.

Once you are satisfied that your adapter is working properly, forget that it is working and check every connection again. If you

Parts List

description	type/part no.	manufacturer	quantity
ribbon cable 24 ga. 20 ckt	8997-20	Molex	one foot
ribbon cable 24 ga. 20 ckt	8997-10	Molex	one foot
ribbon cable connector	22-26-7202	Molex	1
ribbon cable connector	22-26-7102	Molex	1
Wafer, 20 ckt	22-05-2201	Molex	1
Wafer, 10 ckt	22-05-2101	Molex	1
Wafer, 6 ckt	22-05-2061	Molex	1
crimp terminal housing,6 ckt	22-01-2061	Molex	1
2759 series crimp terminals	08-50-0114	Molex	6
Modular connector	520258-3	AMP	2
ET-100 keybd enclosure	92-769 92-777	Heath Parts	one top one bottom
Rubber feet	64-2342	Radio Shack	one package
H-29 keyboard coiled cable	134-1209	Heath Parts	1
circuit board	8001	Vector	1
circuit board	8003	Vector	1
CPU	Z80		1
RAM	6810	Motorola	1
ROM	2716		1
1 MHz osc.	OSC 1.000	Jameco	1
TTL IC's	74159		1
	74LS139		1
	74LS74		1
	74LS240		2
	7406		1
	74LS14		1
	74LS08		1
	74LS164		1
	74LS04		5
	74LS273		1
	74LS174		1
	74150		1
	74LS138		1
	74LS54		1
Wire wrap IC sockets, tin	40 pin		1
	24 pin		4
	20 pin		3
	16 pin		3
	14 pin		11
LED's	276-1622	Radio Shack	21
Resistors 1/4 watt	1 k ohm		21
	2.7 k ohm		3
	10 k ohm		34
	24 k ohm		1
Capacitors	.1 uF bypass ceramic / tant		24
	10 uF tant.		2
Diodes	1N3600		2
Shielded Lug Connectors	64-3049	Radio Shack	one package
Tap-In Squeeze Connectors	64-3052	Radio Shack	one package

Miscellaneous

several colors of 30 gauge wire wrap wire
red and black 18 ga. stranded wire - 2 feet of each color
stranded wire - 22 ga.
self tapping screws to fit the enclosure
auto body putty ???
chewing gum
super glue ???
PC board spacers
small diameter screws with nuts, washers, lock washers
plexiglas to cover the hole left by the keyboard
Vector T-44 (tin) and T-42-1 terminals - lots of them
Vector T-46 (tin) terminals - quantity 5

are 100% satisfied that there are no wiring problems, take a break, have a cup of coffee, run around the block, check every connection one last time and then connect the adapter to the H/Z-100. Plug it in and start typing. If you experience problems, (shame on you) check the decoder output circuits: 74150, 74LS138, 74LS04, 74LS54, ribbon cables, etc.

Parts

Most of the parts can be purchased through the parts houses advertised in magazines such as Byte and Popular Electronics. I purchased the 1 MHz oscillator from Jameco. You will have to do a bit of hunting to locate the Molex ribbon cables and connectors and the AMP modular connectors. You can order the ET-100 keyboard enclosure and the H-29 keyboard cord from Heath Parts.

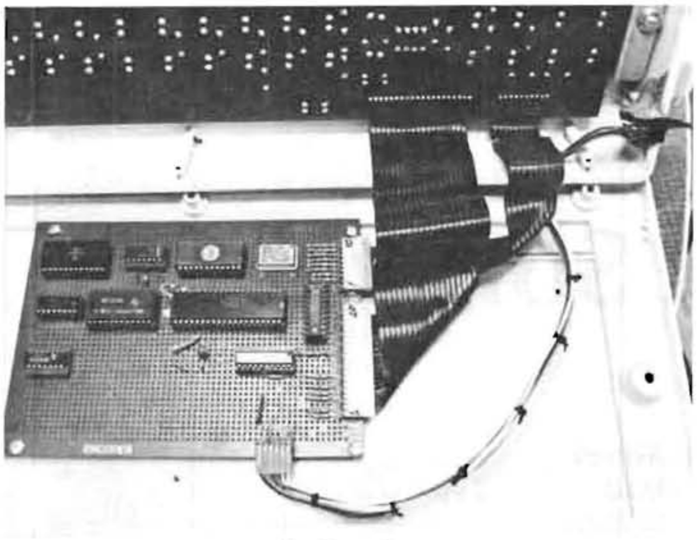
Do not substitute for any the integrated circuits. Different families of TTL devices may be wired differently internally. The 74LS54 is not connected the same way as a 7454. Different families of TTL IC's may require more current and the coiled cord cannot handle substantial increases in current.

Molex's corporate headquarters is:

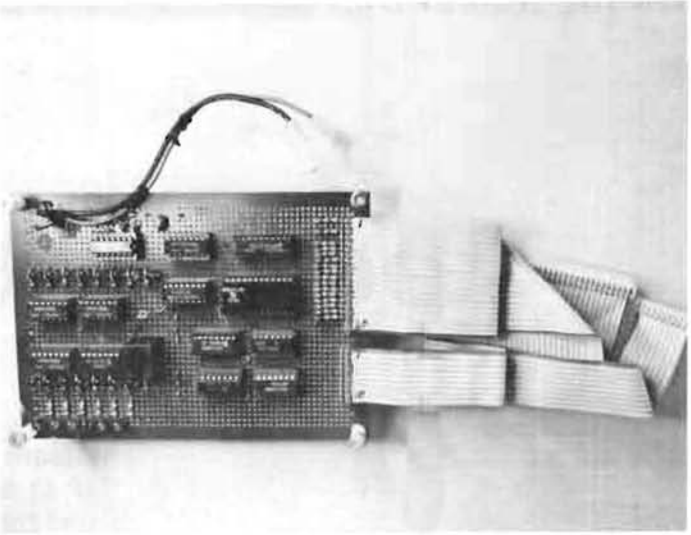
Molex Incorporated
2222 Wellington Court
Lisle, Illinois 60532 Phone 312-969-4550

AMP's address is:

AMP Incorporated
Harrisburg, PA 17105 Phone 717-564-0100



The Encoder



The Decoder

Jameco is one source I used for many of the IC's, circuit boards, etc.

Jameco Electronics
1355 Shoreway Road
Belmont, CA 94002
Phone 415–592–8097

There are Radio Shack stores all over the country.

Phone AMP and Molex requesting the name of the modular connector or ribbon cable dealer nearest you. If you call Jameco they will be glad to send you a parts catalog.

Sending Me Your 2716

When you receive your 2716 ROM from your supplier, send it to me with a check or money order for $25.00 and I will then program it and return it to you with a program listing. When you send the 2716 to me, try to ship it via UPS and insure that it is in a package which will survive a round trip. I have nightmares of a letter carrier trying to stuff 20 boxes through my mail slot. UPS will place your package right in my hands.

My address is:

Tim Ross
1716 S. Solano, APT. 30
Las Cruces, NM 88001

I ask that you do not send me your ROM during the last two weeks of April and during the months of May through September. I will be cramming for finals and, when I graduate, I will be traveling to a job somewhere in the continental US. I will send a letter to REMark informing you of my new location. REMark has a lead time between receipt of a letter and its publication. If you wish to send me your 2716 after April 14, wait a few months because my address will not be valid.

Conclusion

I spent hundreds of hours designing and constructing this adapter. I devoted a good part of this time to the program. I think you will enjoy building and using the adapter. The prototype works like a dream.

Approach this project with caution. Double check every connection before installing this adapter in your HZ–100. I built and rebuilt the prototype several times. If I was not satisfied with my soldering or wire wrapping, I removed the offending wires and started over. Take your time. I will make every effort to answer any questions you may have about the adapter. Be careful and you'll soon be sitting in your swivel chair with your feet on the desk, typing away.

Looking At Software

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

You may think that when you've spent a couple of thousand dollars for your computer, monitor, and printer, that's it. As it usually turns out, most people spend at least that same amount for software since that's the heart of the capabilities that computers give us. That's one of the reasons that I try to look at software in each column. This entire column is devoted to software, with one exception, and that exception happens to be software related since it's a speed-up module for the Z-100.

Software Vendors

All of the vendors that I discuss in this column have shown me a considerable amount of reliability with respect to their products. Although nobody's perfect, I won't report on any software that I have reservations about as a result of the testing that I do. That doesn't mean that all software that I receive, but don't review, has some problem. It may just be a matter of space and my judgment as to what your interests are. In some cases, my best guess may be wrong, but that's the way things are. When I do find problems with software, I always notify the vendor about the problem. That seems to be fair since I usually check out software using a technique called "stress testing". That includes such things as purposely making mistakes that are contrary to the published instructions in order to check out the "error trapping" capability of the program. I do that primarily because I've found that very few people read documentation beyond the installation instructions. It also allows me to check out the functionality of the program since "error trapping" should be at least good enough to avoid catastrophic loss of data and/or a system crash.

Although I've tried to present a number of the Heath/Zenith software vendors in this article, there are a number of other good vendors that also provide reliable software. If you're specifically interested in a particular vendor, you should subscribe to H-SCOOP as mentioned last month. Henry Fale does an excellent job of reporting on vendor reliability to the extent that he specifically names the "good guys" and the "bad guys" for both software and hardware. That includes both quality as well as being reliable for delivering the advertised software or hardware. I have had a painful experience with one of his poorly rated disk drive vendors, but have had good experience with his recommended suppliers. I will not knowingly recommend a vendor with a poor record, but it's happened once, and a promised follow-up article never appeared for that reason. But enough of that...

The intent of this column is to provide you with a look at a variety of software for a number of different purposes. Most, if not all of these vendors, will be available at the HUG Convention so that you will be able to see the programs in action. Since it's always difficult to arrange this kind of article, I've decided to talk about the programs in alphabetical order by company name.

The Z100 Speed Module (CDR Systems)

Although this column is primarily devoted to software, I think it's appropriate to discuss CDR's Z100 Speed Module. With all of the interest in speed improvement, particularly with the availability of the RAM/clock speed upgrade kit that I saw in the latest Heath mailing, it seems worthwhile to discuss it.

If you're interested in improving the clock speed of your Z-100, Z-110 or Z-120 series computer, CDR has the right package for you. For about $50, you can do it, quickly and easily. The best part is that you can select clock speeds with a switch on the back of your Z-100 and there is no soldering involved. The Z100 Speed Module is a small, plug-in board that will not void your warranty.

First of all, why would you want to improve the clock speed? The best answer is that it makes your computer work faster which is quite visible when you are looking at spreadsheets. The Z100 Speed Module provides an approximate 50% improvement in the speed of the 8088 clock...4.77 megahertz to about 7.5 megahertz. Based on some tests that I've done with some of my spreadsheets, it looks like you can expect at least a 50% improvement in recalculation time - in some of my spreadsheets, I observed that recalculations were done nearly twice as fast. In short, almost everything from directory displays to spreadsheets is noticeably faster.

I expect that this module will work with most Z-100's without any problem, although the manual notes that some of the older Z-100 may have some slower chips. I'm particularly impressed by the fact that the CDR manual states that if the Z100 Speed Module does not work with your computer because of the slow chips, you can return it within 30 days for a refund. There is also the normal warranty against defects, but that's a good deal. You can also upgrade your system with faster chips, but of course that's up to you.

Installation is easy and requires disassembly of your computer in order to get to the mother board. The Speed Module itself is a small printed circuit board, approximately 2 inches square, that plugs into U236. The existing chip at U236 is plugged into the Speed Module. Connected to the speed module is a small switch that is mounted in one of the spare DB25 holes on the back of the computer. Reassemble the computer and you're ready to go.

One of the questions that I received in a recent letter asked about software running at higher clock speeds. During the time that I've had the Z100 Speed Module in my H-100, I've had no problem with any of the software. One thing you should know is that a lot of games have "timing loops" in them. These timing loops are programmed to give a certain amount of delay (e.g. one or two seconds) in responses or displays. These games, like spreadsheets will also be speeded up, which can provide an increased challenge. Of course, you can easily select the standard clock speed by simply changing the switch provided on the Speed Module. Although that requires that you power off the computer, it's an incredible amount of flexibility. I should also specifically note that the Speed Module increases the speed of the 16 bit 8088 CPU only. It doesn't do a thing for the 8 bit 8085 chip (which runs the CP/M-85), so you won't really see any improvement on the 8 bit side even though processor swapping does occur in CP/M-85. Some of you may be aware that the 8088 performs memory and display management functions under CP/M-85 which is the reason for the BIOS88.SYS file. Even so, the speed increase is not noticeable on the 8 bit side like it is for the 16 bit processing.

I have found that the Z100 Speed Module does not work with my Z-205 memory board. That's apparently due to the slower chips on the board. The Piiceon 256K memory board that I bought at my local Heathkit store seems to work perfectly at the higher speed however.

Based on the performance, ease of installation, and cost of the Z100 Speed Module, it's a highly useful (time saving) and recommended addition to your hardware.

WordKey For WordStar (DelSoft)

As a word processor, many people consider WordStar to be the standard to which all other similar software should be compared. Although I like WordStar and use it almost exclusively for most of my writing, I have to admit that some of the commands are difficult to remember, and it would be much easier to have them "connected" to a specified function or keypad key. WordKey does that for you and overcomes a lot of difficulty in remembering the commands. WordKey works with the Z-100 MS-DOS version of WordStar (version 3.21 or 3.30) and defines over 90 key combinations in a thoughtful and useful arrangement. If you have WordStar 3.30, which provides the capability to dynamically redefine function keys, you can still use that feature without destroying any of WordKey's definitions.

WordKey is very fast and does not interfere with the standard WordStar commands. It also has a help feature which displays all of the commands for the function keys and the keypad. The function keys (F0-F12) have three sets of associated commands...the unshifted set, the shifted set, and an alternate set which is preceded by pressing the HOME key. The keypad has two sets of commands...the regular set and an alternate set used with the HOME key.

Installation of WordKey is simple and straightforward. All you have to do is copy three files to your WordStar disk. That's it! One of the three files is WK.BAT, and the only change that you have to make to your word processing procedure is to enter the WK command instead of WS for WordStar.

I won't describe which key does what since that would take a while for all of the 90+ keys. Suffice it to say that I found the key arrangement to be easy to learn, and there is an extra page provided with the manual which has all of the key labels. The well documented manual even includes a lot of useful information about WordStar in case you are a new user. If you use the Z-100 version of WordStar under MS-DOS, this is a highly recommended addition to your software library.

Investment Master (Generic Computer Products)

Although I've looked at a number of programs in various columns, I've never spent much time with "financial" programs. In fact, I haven't had much opportunity to work with most of the financial information that I learned for my MBA. In this context, I'm using the word "financial" to mean the analysis of loan amortization, lump sum deposits and withdrawals, and annuities. Investment Master is one of those programs that can make life so much easier when you are struggling to work out the complexities of annuities and lump sum investments. Easy to say, but what does the program do for you?

The objective of the program is to calculate an unknown based on one of the following: current value of investment, interest rate, time, future value of investment, and the number of payments. While that sounds very easy, it's actually a little tricky in practice since it normally requires a number of tables to do the calculations. Investment Master can solve those problems for you quickly and easily. It can solve a number of those "what if" problems for you.

I think that one of the most popular examples is probably the Individual Retirement Account (IRA). Since my understanding is that an individual can invest $2,000 a year, it's interesting to play the what-if question based on 12% interest a year, compounded monthly, for 25 years. The program tells me that the amount is $327,000.

An annuity is essentially an amount paid at regular intervals, usually annually. It can be useful to you if you want to know how much money to save in order to provide a specified monthly income (say $500) during your 20 year retirement. An example in the manual shows that you would need a $51,812.31 investment to provide the specified income. It sure would have been nice to have this kind of program when I was working on my MBA instead of sweating over those very dry tables and a calculator for my homework. A useful program which is recommended.

Loan Master (Generic Computer Products)

Loan Master provides you with an easy way to calculate the payments required for a loan. Obvious uses for this program include calculating home mortgage and car payments which can be very useful. The program even handles "balloon" payments which are sometimes a part of second trust deeds or mortgages. Payment schedules are also provided so that you can see the value of the principle at any point.

Since most people have a considerable amount of experience with loans, I won't spend any time on examples of how the program is used. Suffice it to say that it will provide the information you'll need to analyze a loan or determine the payments for a car or house. I should note that it only applies to fixed rate loans which, of course, excludes the so-called adjustable rate mortgages (ARM's) for homes. That would be a real tough one to program since there are so many variations as I found out last year. Regardless, Loan Master provides some very useful information and is recommended.

Both of the GSPI programs include a terminal configuration program which allows you to configure to 12 different specified terminals, or perform a custom configuration for any other terminal – an extremely useful feature! All information from the program can be displayed on the terminal, written to a disk file, or printed on your system printer. The manual provides enough examples to introduce the programs and the kind of information that they provide.

ACCESS (Hilgraeve)

The ACCESS telecommunications software is one of the nicest programs of its kind that I've ever seen. Although I've had the opportunity to work with CrossTalk, ACCESS is much better in terms of price, performance, and ease of use. It has an incredible amount of flexibility which includes the capability to do some custom "programming" using the AUTOPILOT language. AUTOPILOT is easy to learn if you're so inclined, however, Hilgraeve has included everything you really need (such as an autodialer) so you don't even have to be concerned about the programming, and ACCESS is so powerful that it will probably be the only communications software you will ever need. Even if you already have CrossTalk, I recommend that you consider ACCESS. I think you'll find that it has more features which also makes it easier to use. A worthwhile investment if you do any kind of telecommunications. Highly recommended.

Professional Text Processor (Newline Software)

The Professional Text Processor (PTP) is Newline Software's entry into the high quality word processing software arena. It's available for the Z-100 (CP/M-86, Concurrent DOS, and MS-DOS versions) and the Z-150/IBM PC. Among other features, the optional spelling checker, Professional Spelling Checker, is also available at additional cost.

As I've said before, one of the mandatory features that all word processors worthy of the name should have is the "what you see is what you get" type of display. PTP has that to the extent that it includes on-screen bolding and underline which can even be used in combination. The keypad is used for commands which are arranged in a relatively easy to learn order. The SHIFT and CTRL keys are also used to provide an expanded or modified use of some of the keys. For example, RIGHT-ARROW moves the cursor one position, and SHIFT RIGHT-ARROW moves the cursor to the right side of the screen. The BREAK key is used as the "OOPS" key which can be used to undo the effect of the last command. Valuable when you delete too many lines! The HELP key also provides on-screen help so that, in many cases, you won't have to keep referring to the documentation.

The CTRL key provides two unique features. First, it's used with some of the keypad keys to provide modified commands. For example, the keypad 4 is the FIND command which is used when you don't care to specify upper or lower case. If you enter CTRL keypad 4, the command becomes case sensitive so that the FIND command will search for exactly the specified information. The other use of the CTRL key is the "expert" operations, which is primarily used for changing modes of the system such as setting margins, bolding, underlining, centering, etc. It's also important to note that word wrapping can be turned off so that PTP can also be used as an editor to write program code.

In my opinion, one of the critical features that any word processor should have is the automatic backup feature. That is, when you change a file called "filename", the word processor automatically creates a "filename.BAK" file during the file saving process. PTP has a nice feature which also allows you to turn that feature off. One argument against that kind of feature says that it's an unnecessary waste of disk space. I, for one, have found the automatic backup feature invaluable since I use WordStar. Version 3.22 of MultiMate, which I mentioned last month, does not even allow you the option of creating a backup file – another reason why I don't care for it much. I have heard that MultiMate version 3.30 does allow the automatic backup option, but that may be too late for a lot of users. Back to PTP.

PTP also has a rather complete array of configuration options, including color changes (if you have a color monitor), printer configurations, and display changes. A particularly unique feature of PTP is that it has a series of menus, like Lotus 1-2-3 and MultiPlan, which are activated by selecting the OPTIONS key (keypad 0). These menus allow you to set/reset margins, tab stops, cursor position, line functions (e.g. centering), paragraph functions (e.g. justification), file operations (e.g. read/write file), and change disks (reset system). I found that this menu system allows a new user to quickly learn how to perform these functions with very little reference to the manual.

User defined macros are also supported by PTP. A macro is simply a single computer instruction (or keystroke) that stands for a sequence of multiple computer operations. PTP allows you to define a series of operations and assign them to a function key. All function keys can be programmed with user macros if desired. Examples are included in the manual for double spacing a document and converting tab stops to spaces. In WordStar, a useful macro would be Ctrl-KS Ctrl-QP, which allows you to save a file (Ctrl-KS) and return to the current position in the file (Ctrl-QP). As you become more familiar with word processing, you will undoubtedly find particular macros that will be useful in your work.

A friend of mine has recently had a lot of trouble trying to get his printer to work properly with some of his software. Although most of the old CP/M software allowed you to do custom configurations for your printer (like implementing bold face by using a double strike mode), it's my observation that much of the MS-DOS/PC-DOS software does not allow that kind of versatility. As a consequence, if the software that you want does not directly support your printer, you are probably out of luck until the manufacturer decides to add a printer driver for that particular printer. The only reason that this is worth mentioning is that PTP very CLEVERLY allows you to input virtually all printer control sequences (except CTRL-J and CTRL-Z) into a source document. And that includes all Control (CTRL) and Escape (ESC) sequences except as noted above. That's just in case that PTP does not directly support the particular feature on your printer.

Although this particular version of PTP was reviewed using the CP/M-85 operating system on an H-100, the MS-DOS or PC-DOS version functions in a similar way. PTP is recommended and is a good choice for anyone who needs a full featured word processor at a reasonable price.

Z-TOUR 1000 Game (Newline Software)

Before I talk about this game, I have to tell you that the Mille Bornes card game by Parker Brothers is just about my favorite game. Z-TOUR 1000 is the computer game that is almost completely identical in rules, scoring, and display. As a result, I really like Z-TOUR 1000 since it remains faithful to the basic card game version. It's unfortunate, from my perspective, that I don't have a color monitor since that's one of the requirements of the game. Since I know the card game so well, I was able to successfully play Z-TOUR 1000 with my monochrome monitor (ZVM-123A), however, I don't recommend it unless you KNOW the game. The "ACCIDENT" card doesn't show up at all, and the pointers for the game extension and your card play don't either. After some experimentation, I was able to determine that the card pointer always appears on the middle card (out of seven in your hand) displayed at the bottom of the screen. Similarly, the default response to the request for the game extension is "No". After I figured that out, playing the game was simple and enjoyable.

Aside from the color monitor requirement, you must also have the ZBASIC interpreter for the Z-100. Unfortunately, Z-TOUR 1000 is not available for the Z-150/IBM PC. The good news is that the game is very simple to learn and can provide hours of entertainment. It's a recommended addition to your game library.

Boxes & Switches Game (Newline Software)

The Boxes & Switches game package is actually two games. Boxes is essentially a "connect the dot" type of game that sometimes appears in crossword puzzle books. The objective of the game is to create boxes, and the last one to complete the box gets the score. You can custom configure the game using a menu which allows you to set the size of the boxes, number of players, overall size of the grid, color, and so on. You can even customize help screens for this program. Very simple to learn and Newline says it's for players of all ages.

Switches is a puzzle composed of a series of rectangles. The objective of the game is to completely switch all of the puzzle parts over to a different color. While that sounds simple to describe, the catch is that when you switch one rectangle's color, the color on each of the adjacent rectangles also switches. In some ways, this game reminds me of a kind of computerized "Rubik's Cube", however, the Switches game is an original idea by the author of the program, Terry Wilk.

If you (or your children) enjoy these kinds of games, they're recommended. I found these games to be habit forming.

Software Toolworks

A review of software for the Heath/Zenith systems wouldn't be complete without including one of the original vendors, Software Toolworks. Walt Bilofsky is known for providing us with quality software at reasonable prices and was recognized for that at last year's HUG Convention. As I said at the beginning of the column, I wanted to provide a little variety in the software reviews, so we'll look at some of Walt's compilers for various programming languages.

C/80 Compiler And MATHPAK (Software Toolworks)

The C/80 Compiler is one of Walt's best known programming products. It has been widely recognized as an excellent programming tool which produces good code. For those of you interested in learning a programming language, I recommend the C language as a good place to start. It has the power of assembler plus I think it's easier to learn. That, plus the fact that most vendors are providing C compilers for nearly every make of microcomputer system available, makes it a good investment in both time and money. One of the biggest disadvantages of assembler is that you have to learn a new assembler for every type of microcomputer processor (i.e. CPU). The 8080 assembler (for 8-bit software) is different from the 8086 version (16-bit). If you decide to use a 6502 computer chip (like the Apple, heaven forbid), that's also a different assembler. They are different in the sense of how the instructions are programmed, although they essentially perform the same kind of functions.

The C/80 Compiler has some limitations, but for the price ($50), it's an excellent value if you're not sure which programming language that you want to use. Although there's no reason why the C/80 Compiler can't give you just about all the capability you need, you can also add the optional MATHPAK which provides the capability for long, float, and double keyword declarations as well as a transcendental function library to provide sine, cosine, tangent, exponential, and logarithm functions. C/80 and the MATHPAK are available for both CP/M and MS-DOS operating systems and are recommended.

LISP/80 (Software Toolworks)

LISP/80 is an interpreter for the LISP programming language used in experimenting with artificial intelligence applications. More than 75 built-in functions are included and this package provides the basic LISP data structures, list operations, recursion, string operations, file I/O, and so on. Also included in this package are a simple editor and file package, written in LISP. Two sample programs are included: a guessing game and a simple version of the ELIZA psychiatrist program.

This is the kind of software package that I could spend hours trying to describe. While it's not new, it seems that artificial intelligence is becoming another computer buzzword that everyone will hear, so it seems appropriate to discuss the subject here.

After spending some time with the language, it does appear easy to learn and lives up to its promise of letting you concentrate on the programming instead of the code. If you like to experiment with new kinds of languages, the reasonable price of the LISP/80 package provides that for you.

RATFOR (Software Toolworks)

RATFOR is an acronym for RATional FORtran. It's an extension of the FORTRAN programming language which allows for structured programming and other features in FORTRAN. Since we're talking about acronyms, you also should know that FORTRAN stands for FORmula TRANslation which is a programming language designed primarily for the processing of arithmetic formulas. For what it's worth, I originally learned FORTRAN when I was studying for my degree in Electrical Engineering at Purdue University.

In order to use RATFOR, you must also have the MicroSoft FORTRAN compiler. RATFOR itself is a preprocessor which is followed by the standard FORTRAN compile and link. In many ways, RATFOR reminds me of the C language. STDIN and STDOUT are defined in the same way that they are used in C. A library of routines is provided which is composed of recognizable C constructs such as getch, putch, fopen, ctoi, and so on. Anyone who works with both C and FORTRAN would probably find that RATFOR is a valuable addition to their programming library. It's recommended.

Perks (Barry Watzman)

Most of us with Z-100's feel neglected when it comes to a lot of the software that's available for the Z-150/IBM PC. Utilities like Sidekick simply have not been available to us since they won't run on the Z-100's. That's unfortunate, in my opinion, since the Z-100 system is probably still the best designed computer on the market. From a technical perspective, the IBM PC used two year old technology at the time it was introduced – the Z-100 was innovative and a credit to its designers. I only mention that because Barry Watzman was responsible for the architecture in the Z-100 as well as bringing it to us.

Barry has continued to support the Z-100 by providing operating systems – CP/M-86, CP/M Plus, and M/M-86 – for the Z-100. He has just finished the development of an extremely useful MS-DOS utility package, Perks, which is just as impressive as his other achievements.

Perks has all of the needed features in the "desktop" utility packages...a calculator, an appointment calendar, an ASCII table, a "notepad" with editing features, and various configuration options in the "filer". For those of you concerned about the problem of "burning" an image on your monitor when it's left on for long periods of time without use, Perks has a Screen Saver which turns off the video after a set "inactivity threshold" (which you can adjust). The screen image is restored when you press any key on the keyboard – a particularly useful feature for me.

Perks is loaded into memory one time, and can then be activated at any time (even during another program like WordStar, Lotus 1-2-3, etc.) with the Shift-Break keys. Windows are used for Perks, which is provided by Barry's own PANE-RELIEF window manager. That is, you can activate one or more of Perks' functions at any time, and the window(s) will be displayed on the screen. Perks is menu driven and provides appropriate help screens by simply pressing the Help key. I found that Perks provided me with the best laugh that I've had in weeks. After working with a lot of software and writing books, I was (I thought) ready for anything. The first time in ran Perks, I saw the sign-on message: "Perks Adjusting for Monochrome-Only Z-100 (Ugh!)." After a lot of work on the rather dry material for Volume 2 of the MS-DOS FlipFast book, that really hit me as funny. I guess you just had to be there. Obviously Perks can adjust for either color or monochrome, and although it's great in monochrome, my guess is that it would be spectacular in color.

The calculator provides the normal 4 functions with memory and provides the capability to do operations in both hex and decimal. Function keys and the keypad are used in the operation of the calculator.

The Notepad provides basic editing functions using WordStar-like commands. Cursor movement including scrolling, insert/delete functions, search/replace operations, and block/file functions are all provided. As if that weren't enough, the display size of the notepad is also adjustable, and you can also use the import module to copy any of the alphanumeric contents of the screen that existed prior to the activation of Perks. If you were working on a SuperCalc file prior to activating Perks, that means you can copy part of the spreadsheet into the notepad.

The ASCII Table shows all of the standard 128 characters plus the standard Z-100 graphics characters. Scrolling through the table is easy...just hit the up/down cursor keys. The Perpetual Calendar shows any month in the 20th century. The Appointment Calendar is activated by CTRL-L (among others) and allows you to create, change or print your appointments.

The Filer and Setup module provides a number of features. You can adjust the Screen Saver threshold, set up to 8 timers, list the directory, erase files, set the default drive and directory, and define the filespec for the Perks files.

Movement within Perks is accomplished by using the Break or CTRL keys in combination with the first letter of the function. As previously mentioned, Perks itself is activated by the Shift-Break keys, which results in a display of the menu. If you need the calculator, press "C". If you're not in the menu and want to bring up additional modules of Perks, you can use the Break or CTRL key in combination with the first letter of the function as usual. For example, if you're currently using the calculator and decide that you need the ASCII Table, Break-C or CTRL-C will display the table leaving the calculator displayed on the screen. All of the functions can be displayed in a similar manner. You can also press the Help key at any time to get information for Perks.

Although some of the other desktop utility packages also include an autodialer, I've heard that the autodialer function has created a number of headaches for the developers related to the so-called Hayes compatible modems. If the modem is not EXACTLY Hayes compatible and the configuration switches are not set properly, the autodialer will not work. As you may have guessed, the autodialer function requires a modem which is another expense. While a computer can perform a number of useful functions, I prefer a telephone with the autodialer feature. I can't imagine why anyone would want to power up a computer to make a simple telephone call. Of course if you have the computer and modem turned on anyway, I suppose that might be a useful feature, but I still prefer an external dialer such as the Heath HOTSHOT (GT-2218).

As many of you know by now, I am personally against any of the "copy protection" schemes since they are a pain in the neck for users, particularly when you have a hard disk. For those of you who don't know, I will not even review any of the copy protected software in this column as a matter of personal policy.

Perks is not copy protected, although Barry has cleverly developed a scheme of "revenue protection" which is intended to accomplish the same result without interfering with the operation of Perks. Aside from the software serial numbers, Barry requires that you must sign a license agreement before you receive Perks. Retail versions of Perks are limited in that some functions are not provided and other functions are not complete. If you buy the retail version, you must send in the license agreement in order to receive the entire package. In addition, Barry customizes each package which includes up to 20 characters, usually your name, so that it would be obvious to any software "thief" that the product was stolen each time it was used. That idea also makes it easier to prosecute any violations of the license agreement since the software is customized to that extent. I don't think that this idea imposes any hardship on users since you always should send in your license agreements anyway...if for no other reason than to be registered for update purposes. Don't be concerned about the fact that you won't have the complete version of Perks...I received mine in about 5 working days after my request to Barry, and that includes time in the mail.

As far as Perks is concerned, I found it to be a highly useful utility, and it's highly recommended. It's easy to use, well documented, and performs a number of functions that I've found to be very helpful.

In The Future

As I've said before, the intent of this column is to provide information that is most useful to HUG members. Based on a number of letters that I've received plus information provided by HUG, I am writing a series of columns which are specifically intended to introduce microcomputing to a lot of our new members. I have a friend, Richard Fairlamb by name, who has been asking me a lot of questions about microcomputing, probably because he bought an H-150 on my recommendation about 6 months ago. Although Richard has some experience in mainframes, micros are new to him, and he's asked a lot of questions about the workings of the computer and MS-DOS. Although I will specifically deal with a number of questions on the Heath/Zenith version of MS-DOS, most of the information presented in this series will be related to microcomputing in general. That is, it will apply to anything you do with a micro regardless of the brand. MS-DOS specifics will also apply to IBM PC-DOS with very few exceptions.

I haven't developed titles for all of these columns yet, but I'll give you some descriptive examples of the subjects to be covered in the following paragraphs.

Setting up your system – This includes some basic things that you should know in setting up any computer system. For example, what is the real difference between a system disk and a data disk? How should you set up a floppy disk library?

Files and FAT's – What is a file? What is a disk directory? How is it used? What are File Allocation Tables (FAT's)? Why are different file types (extensions) used like COM and EXE?

MS-DOS/PC-DOS commands you must know – What are the basic operating system commands that you must know in order to successfully operate your system? What are built-in and disk resident commands? How and when can they be used?

MS-DOS Directory Trees, I/O Redirection, and Command Pathing –How to effectively define and use MS-DOS directory tree structures in your system. What is I/O redirection, and how is it used? A discussion on the real uses of command pathing.

Batch file processing – How to develop batch files that perform useful functions in your system. Includes AUTOEXEC.BAT files and how batch files are processed.

Copy protection – What is it? What are the different forms of copy protection and why is it dangerous to the health of you and your data (the answer is Worms!)? What does copy protection mean to you? Why are manufacturers completely ignoring the problem and trying to solve it with copy protection?

Data Bases – What is a data base? Why is it useful? Includes a look at some popular data base software – dBase and Condor.

Integrated Software – Is it worth it? Are there better ways of doing your job? What are the advantages and disadvantages?

As you can probably tell, not all of these ideas are fully developed at this point, and it's likely that some of them will lead to more than one column because of length. The ideas are not arranged in any particular order, so there is no promise that they will appear in the order listed.

I'm always interested in hearing about your comments and questions. If you have a particular subject that you would like to see discussed in this column, be sure to let me know. I'll acknowledge all subject ideas by printing your name in that column. As always, if you want a personal reply to your question or comment, please enclose a stamped, self addressed envelope with your letter.

I'll also continue to comment on various software and hardware for your computer systems. HUG has a lot of excellent software that will be discussed as appropriate in the column. One example is an excellent program by Pat Swayne which lists the directory in alphabetical order.

As I'm finishing this in June, we are supposed to hear that MicroSoft has released Windows to manufacturers this month. We'll see if that happens. Regardless of when it's released, I plan to include a column on that when it's available for the Heath/Zenith computer systems.

In any event, I'm getting ready for the HUG Convention with my presentation on How to Use MS-DOS Directory Trees, I/O Redirection, and Command Pathing. Hope to see you there!

Products Reviewed

H-Scoop (newsletter) (12 issues) 2618 Penn Circle Sheboygan, WI 53081	$20.00 per year
Z100 Speed Module (Z-100 only) Heathkit Stores CDR Systems, Inc. 7210 Clairemont Mesa Blvd. San Diego, CA 92111 (619) 560-1272	$49.95
WordKey (MS-DOS, Z-100 only) Heathkit Stores DelSoft 564 Calle Anzuelo Santa Barbara, CA 93111 (805) 967-9566 (eves. & weekends)	$49.95
Investment Master or Loan Master	
Z-89 Version	
HDOS	$29.95
CP/M-80	$29.95
Z-100 Version	
CP/M-85	$29.95
MS-DOS/Z-DOS	$49.95
Z-150/IBM PC Version	
MS-DOS/PC-DOS	$49.95
Heathkit Stores Generic Computer Products, Inc. P.O. Box 790 Marquette, MI 49855 (906) 249-9801	
ACCESS	
H-8, H/Z-89 (CP/M-80)	$49.95
Z-100 (MS-DOS, CP/M-85, CP/M-86)	$59.95
Z-150/IBM PC (MS-DOS, CP/M-86)	$69.95
Heathkit Stores Hilgraeve, Inc. P.O. Box 941 Monroe, MI 48161 (313) 243-0576	

Professional Text Processor	
Z-100 Version	
MS-DOS/Z-DOS	$99.00
CP/M-86	$99.00
Z-150/IBM PC Version	
MS-DOS/PC-DOS	$149.00
CP/M-86	$149.00
Z-TOUR 1000 Game	
Z-100 MS-DOS only (color monitor)	$29.95
Boxes & Switches Game	
Z-100 MS-DOS only	$29.95

Heathkit Stores
Newline Software
P.O. Box 289
Tiverton, RI 02878 (401) 624-3322

C/80 Compiler (CP/M or MS-DOS)	$49.95
C/80 MATHPAK (CP/M or MS-DOS)	$29.95
LISP/80 (CP/M or MS-DOS)	$39.95
RATFOR (CP/M only)	$39.95

Heathkit Stores
Software Toolworks
15233 Ventura Blvd.
Sherman Oaks, CA 91403 (818) 986-4885

Perks	
MS-DOS (Z-100 only)	$99.97

Heathkit Stores
Barry Watzman
560 Sunset Road
Benton Harbor, MI 49022 (616) 925-3136

*

Double the Speed of your H/Z89-90
Cut computing time by as much as 50%

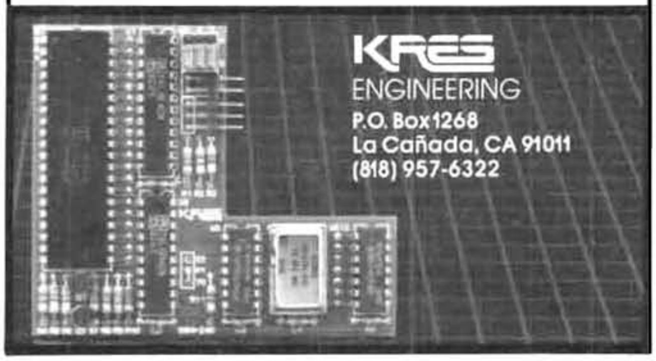
Fast Operation
With the Dual Speed Module, software that used to be slow and inefficient now zips right along. Run programs like SuperCalc (in "auto mode") and Spellstar without long waits.

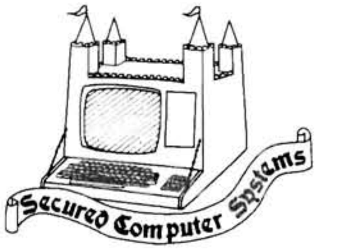
Easy
Simply replace your existing microprocessor IC with the Dual Speed Module and stop wasting time. Installation requires no trace cutting or soldering.

Fully Supported
Software support for CP/M, HDOS, AND CP/M + Works with Magnolia Microsystems*, CDR, and many other disk systems.

Dual Speed Module DSM-240 $99.97**
+ $3.50 shipping and handling [includes CP/M or HDOS; add $10.00 for both]

*Requires Software Patch PMM-100 $49.95 and Monitor ROM KMR-100 $59.95
**All prices subject to change without notice
CP/M is a registered trademark of Digital Research.


KRES
ENGINEERING
P.O. Box 1268
La Cañada, CA 91011
(818) 957-6322



HEATH/ZENITH 88, 89, 90 PERIPHERALS

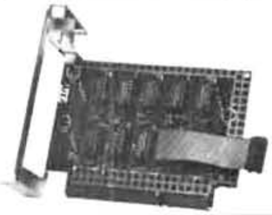
16K RAM EXPANSION CARD
Expand your H/Z 88, 89 RAM Memory to a **FULL 64K** and begin using larger and more powerful programs with our **16K RAM** card.

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

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

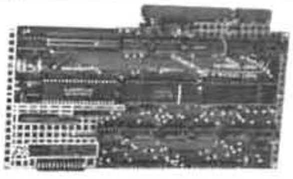
Field reliability record exceeding 3 years

Only $65.00 Shipping & Handling $5.00



2/3 PORT SERIAL — CARD — I/O PORT PARALLEL

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



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

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

Complete with Installation Instructions, Documentation, 90 Day Warranty, Two Serial Cables and a Parallel Cable Internal to the Computer

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

REAL TIME CLOCK

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

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

Price $105.00 with Batteries **$89.00 without Batteries**
$ 25.00 Software for Second Operating System
Shipping & Handling $5.00

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

ZENITH
data systems
SERVICE CENTER

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

HUG-Aids

Pat Swayne
HUG Software Engineer

Recovering From The Protected Mode In ZBASIC Or GW-BASIC

In ZBASIC or GW-BASIC (BASICA), you can save a program in a "protected" format by entering

```
SAVE "PROGRAM",P
```

when you save the program to disk. A program saved this way cannot be listed or edited when it has been re-loaded from the disk. This is fine if you want to keep others from looking at your code, but it can also give a programmer headaches. I have been called by users who say that they have either "crashed" the disk containing the unprotected copy of a program they have been working on, or they accidently saved a program in protected mode on to their development disk, overwriting the unprotected copy.

Various methods have been developed for removing the "protection" from protected BASIC programs. Probably the simplest method was developed by HUG member Bill Parrot, who wrote an assembly language program that could be loaded into memory and then activated via a function key to remove the protection from a loaded ZBASIC program. His method involved changing single a byte in ZBASIC's working memory space that apparantly is a flag to ZBASIC indicating a protected program. Bill Parrot's program was released into the public domain on the HUG SIG on Compuserve, but unfortunately, it only worked with a specific version of ZBASIC.

I have developed two universal protection removing programs based on Bill's original program. One is for the H/Z-100, and the other is for the H/Z-150 and other PC-type computers. These versions work with any version of ZBASIC or GW-BASIC. Since many BASIC users are uncomfortable with assembly language, I have provided the following two BASIC programs that can create the unprotect program:

```
1Ø REM THIS PROGRAM CREATES UNPRO.COM (Z-1ØØ)
2Ø OPEN "O",1,"UNPRO.COM"
3Ø FOR I=1 TO 54
4Ø READ BYTE:PRINT #1,CHR$(BYTE);:NEXT I
5Ø CLOSE #1:SYSTEM
6Ø DATA &HEB,&H1D,&H5Ø,&H51,&H56,&HBE,&HØØ,&HØ3
7Ø DATA &HB9,&HØØ,&HØ8,&HFC,&HAC,&H3C,&HFE,&H75
8Ø DATA &HØ8,&HC6,&H44,&HFF,&HØØ,&HBØ,&HØ7,&HE6
9Ø DATA &HF5,&HE2,&HF1,&H5E,&H59,&H58,&HCF,&H1E
1ØØ DATA &H33,&HCØ,&H8E,&HD8,&HBE,&H14,&HØØ,&HFA
11Ø DATA &HC7,&HØ4,&HØ2,&HØ1,&H8C,&H4C,&HØ2,&HFB
12Ø DATA &H1F,&HBA,&H1F,&HØ1,&HCD,&H27
```

```
1Ø REM THIS PROGRAM CREATES UNPRO.COM (Z-15Ø)
2Ø OPEN "O",1,"UNPRO.COM"
3Ø FOR I=1 TO 96
4Ø READ BYTE:PRINT #1,CHR$(BYTE);:NEXT I
5Ø CLOSE #1:SYSTEM
6Ø DATA &HEB,&H3D,&H8Ø,&HFC,&HØØ,&H75,&H2E,&H53
7Ø DATA &H9C,&HØE,&HBB,&H1Ø,&HØ1,&H53,&HEB,&H25
8Ø DATA &H3D,&HØ4,&H2Ø,&H75,&H1E,&H51,&H56,&HBE
9Ø DATA &HØØ,&HØ3,&HB9,&HØØ,&HØ8,&HBB,&HØØ,&HØ3
1ØØ DATA &HFC,&HAC,&H3C,&HFE,&H75,&HØ7,&HC6,&H44
11Ø DATA &HFF,&HØØ,&HBB,&HØ7,&H22,&HE2,&HF2,&H5E
12Ø DATA &H59,&H8B,&HC3,&H5B,&HCF,&H2E,&HFF,&H2E
13Ø DATA &H3A,&HØ1,&HØØ,&HØØ,&HØØ,&HØØ,&HØØ,&H1E
14Ø DATA &H33,&HCØ,&H8E,&HD8,&HBE,&H58,&HØØ,&HC4
15Ø DATA &H3C,&HFA,&HC7,&HØ4,&HØ2,&HØ1,&H8C,&H4C
16Ø DATA &HØ2,&H1F,&H89,&H3E,&H3A,&HØ1,&H8C,&HØ6
17Ø DATA &H3C,&HØ1,&HFB,&HBA,&H3F,&HØ1,&HCD,&H27
```

All you have to do is type in the appropriate program and RUN it using any version of ZBASIC or GW-BASIC. The program will create a file called UNPRO.COM on your disk, which is the actual protection removing program.

To use UNPRO.COM, make sure that it is on your system disk, and enter

```
A>UNPRO
```

The program will load itself into memory and return you to the system. Now, run ZBASIC or GW-BASIC and load the protected program that you wish to unprotect. Activate UNPRO by typing SHIFT-F12 on a Z-100, or Control-D on a Z-150. If you hear a beep, it means that UNPRO was probably successful in fixing your program. To make sure, try to LIST the program. If you can list it, IMMEDIATELY save the program to disk (without the ,P option!), and re-boot your system to remove UNPRO from memory. The reason for immediately removing UNPRO from memory after using it is because it may alter the operation of a program or the operating system if it is accidently activated when ZBASIC or GW-BASIC is not running.

If you would rather create UNPRO.COM using an assembler, or would just like to study the assembly source code, you can use one of the two listings following this article. Those of you who are assembly programmers will notice the "DUMMY" stack segment in each program. It is there to prevent LINK from issuing the "Warning – No Stack Segment" message that it normally displays when you link a program designed in the .COM file model. The dummy segment has no other effect.

The programs work by searching a designated area in the ZBASIC or GW-BASIC data segment for the number FE (hex). If this number is found, it is patched to 0. It is possible that another FE besides the one designating a protected program could exist in the searched area, and it would also be patched to zero. This is another reason why you are directed to save the BASIC program immediately after fixing it, in case the BASIC interpreter has been "messed up". However, in my tests, no ill effects were observed.

The Z-100 version of UNPRO uses the print screen interrupt (produced by SHIFT-F12) to start the search for the protection flag. If the search is successful, it beeps by writing the appropriate code to the keyboard/screen control port. The Z-150 version intercepts the keyboard interrupt, and starts the search when it detects the key code for Control-D. If the search is successful, it returns the key code for Control-G (2207) to GW-BASIC, which causes a beep. If the search is not successful, a null code (0300) is returned, which GW-BASIC does not act upon.

```
;        UNPRO - UNPROTECTOR FOR BASICA (Z-1ØØ) AND ZBASIC
;
;        THIS PROGRAM UNPROTECTS BASICA AND ZBASIC PROGRAMS
;        THAT WERE SAVED WITH THE ",P" SWITCH.
;
;        TO USE THIS PROGRAM, RUN UNPRO.  THEN RUN BASICA AND
;        LOAD THE PROTECTED PROGRAM.  TO UNPROTECT THE FILE,
;        TYPE SHIFT-F12.  IF THE UNPROTECTION IS SUCCESSFUL,
;        YOU WILL HEAR A BEEP, AND YOU WILL BE ABLE TO LIST
;        THE PROTECTED PROGRAM.
;
;        BY P. SWAYNE, HUG  Ø9-MAY-85
;        BASED ON BILL PARROTT'S Z-BASIC UNPROTECTION PROGRAM
;

DUMMY    SEGMENT STACK              ; AVOID LINK ERR MSG.
DUMMY    ENDS
CODE     SEGMENT
         ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
         ORG     1ØØH

START:   JMP     SHORT SETUP

;        PROCESS INT 5 HERE, WITH CODE TO UNPROTECT PROGRAMS

INT_5:   PUSH    AX                 ; SAVE REGISTERS
         PUSH    CX
         PUSH    SI

;        SEARCH Z-BASIC DATA AREA FOR THE PROTECT FLAG

         MOV     SI,Ø3ØØH           ; POINT TO SEARCH AREA
         MOV     CX,8ØØH            ; AMOUNT TO SEARCH
         CLD                        ; SEARCH FORWARD
FLOOP:   LODSB                      ; GET A BYTE
         CMP     AL,ØFEH            ; PROTECT FLAG?
         JNE     NOTPRO             ; NO
         MOV     BYTE PTR -1[SI],Ø ; ELSE, REMOVE IT
         MOV     AL,7
         OUT     ØF5H,AL            ; SIGNAL FLAG REMOVED
NOTPRO:  LOOP    FLOOP              ; KEEP LOOKING
         POP     SI                 ; RESTORE REGISTERS
         POP     CX
         POP     AX
         IRET                       ; RETURN FROM INTERRUPT

ENDRES   LABEL   NEAR

;        SET UP INT 5 VECTOR TO THIS PROGRAM

SETUP:   PUSH    DS
         XOR     AX,AX
         MOV     DS,AX              ; POINT TO INT SEGMENT
         MOV     SI,5*4             ; POINT TO INT 5 VECTOR
         CLI                        ; KILL INTERRUPTS
         MOV     WORD PTR [SI],OFFSET INT_5
         MOV     WORD PTR [SI+2],CS ; INSERT INT 5 VECTOR
         STI                        ; FIX INTERRUPTS
         POP     DS
         MOV     DX,OFFSET ENDRES ; POINT TO END OF RES. CODE
         INT     27H                ; EXIT, CODE RESIDENT

CODE     ENDS
         END     START
```

```
;        UNPRO - UNPROTECTOR FOR BASICA (Z-15Ø)
;
;        THIS PROGRAM UNPROTECTS BASICA PROGRAMS THAT WERE
;        SAVED WITH THE ",P" SWITCH.
;
;        TO USE THIS PROGRAM, RUN UNPRO.  THEN RUN BASICA AND
;        LOAD THE PROTECTED PROGRAM.  TO UNPROTECT THE FILE,
;        TYPE CONTROL-D.  IF THE UNPROTECTION IS SUCCESSFUL,
;        YOU WILL HEAR A BEEP, AND YOU WILL BE ABLE TO LIST
;        THE PROTECTED PROGRAM.
;
;        BY P. SWAYNE, HUG  Ø9-MAY-85
;        BASED ON BILL PARROTT'S Z-BASIC UNPROTECTION PROGRAM
;

DUMMY    SEGMENT STACK              ; AVOID LINK ERR MSG.
DUMMY    ENDS
CODE     SEGMENT
         ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
         ORG     1ØØH

START:   JMP     SHORT SETUP

;        INTERCEPT INT 16H AND LOOK FOR CONTROL-D

INT_16:  CMP     AH,Ø               ; GET KEY?
         JNZ     EXINT              ; IF NOT, GO TO BIOS
         PUSH    BX                 ; ELSE, SAVE BX
         PUSHF                      ; PUT FLAGS ON STACK
         PUSH    CS                 ; AND CS
         MOV     BX,OFFSET RETADR
         PUSH    BX                 ; AND RETURN ADDRESS
         JMP     SHORT EXINT        ; LET DOS GET KEY
RETADR:  CMP     AX,2ØØ4H           ; CONTROL-D?
         JNZ     NOTD               ; NOPE
        PUSH    CX
         PUSH    SI                 ; SAVE REGISTERS

;        SEARCH BASICA DATA AREA FOR THE PROTECT FLAG

         MOV     SI,Ø3ØØH           ; POINT TO SEARCH AREA
         MOV     CX,8ØØH            ; AMOUNT TO SEARCH
         MOV     BX,Ø3ØØH           ; ASSUME NO PROTECT FLAG
         CLD                        ; SEARCH FORWARD
FLOOP:   LODSB                      ; GET A BYTE
         CMP     AL,ØFEH            ; PROTECT FLAG?
         JNE     NOTPRO             ; NOT A PROTECT FLAG
         MOV     BYTE PTR -1[SI],Ø ; REMOVE FLAG
         MOV     BX,22Ø7H           ; SET SUCCESS SIGNAL
NOTPRO:  LOOP    FLOOP              ; KEEP LOOKING
         POP     SI                 ; RESTORE REGISTERS
         POP     CX
         MOV     AX,BX              ; RETURN RESULT SIGNAL
NOTD:    POP     BX
         IRET

EXINT:   JMP     DWORD PTR CS:EXADR ; EXIT TO BIOS CODE

EXADR    DW      Ø,Ø                ; PUT EXIT ADDRESS HERE
         DB      Ø
ENDRES   LABEL   NEAR

;        SET UP INT 16H VECTOR SO THAT WE CAN LOOK FOR
;        CONTROL-D

SETUP:   PUSH    DS
         XOR     AX,AX
         MOV     DS,AX              ; POINT TO INT. SEGMENT
         MOV     SI,16H*4           ; POINT TO INT 16 VECTOR
         LES     DI,DWORD PTR [SI] ; GET IT
        CLI                              ; KILL INTERRUPTS
        MOV     WORD PTR [SI],OFFSET INT_16
        MOV     WORD PTR [SI+2],CS ; INSERT NEW VECTOR
        POP     DS                 ; FIX DS
        MOV     EXADR,DI
        MOV     EXADR+2,ES         ; SET UP CHAIN TO OLD CODE
        STI                        ; FIX INTERRUPTS
        MOV     DX,OFFSET ENDRES ; END OF RESIDENT CODE
        INT     27H                ; EXIT, CODE RESIDENT

CODE    ENDS
        END     START
```

Fixing Back Space On Z-150 Wordstar

When you run WordStar on a Z-150 or other PC-type computer, the Back Space key does not perform its normal function of removing the last character typed, but instead it just backs up the cursor like the left arrow key. In order to delete the last character, you either have to use the Del key, which is out of reach from the normal hand position while typing, or you can use Control-Back Space, which requires two hands. When you type Control-Back Space, the computer produces a true Delete code, which is the code WordStar uses for deleting the last character typed.

The assembly listing that follows this program will produce, when assembled, a program that allows you to use your Back Space key to produce either the Back Space or Delete code when it is pressed by itself. The combination Control-Back Space is used to toggle the response of the Back Space key between the two codes.

If you cannot assemble the assembly listing, you can use the BASIC program that follows it, which produces the file FIXBS.COM when it is run (be sure that you type the DATA statements exactly as shown). After you make FIXBS.COM from either the assembly listing or the BASIC program, place it on your system disk, and enter

```
A>FIXBS
```

The program will be loaded into memory and remain there until you re-boot your computer. While it is loaded, you can run WordStar and type Control-Back Space once, and from then on you can use the Back Space key by itself to correct typing errors in WordStar. After you exit WordStar, you can type Control-Back Space again to restore the output of Back Space to its normal code. Many programs, and MS-DOS itself, allow the use of either the Back Space or Delete codes for deleting typed characters, so if you forget which state you have left the key in, it may not make any difference. However, BASICA requires the Back Space code, and will echo a graphic character if a Delete code is typed into it.

If you use KEYMAP (HUG no. 885-6001-37) for mapping keys in WordStar, you should load FIXBS before you load KEYMAP.

```
;       FIXBS - FIX BACK SPACE KEY
;
;       THIS PROGRAM ALLOWS THE BACKSPACE KEY ON A Z-15Ø
;       TO BE USED AS A BACKSPACE OR DELETE KEY.
;       CONTROL-BACK SPACE TOGGLES THE FUNCTION OF THE
;       KEY BETWEEN BACK SPACE OR DELETE.
;
;       BY P. SWAYNE, HUG  Ø3-JUN-85
;
BSCODE  EQU     ØEØ8H              ; BACKSPACE KEY CODE
DELCODE EQU     ØE7FH              ; DEL. KEY CODE

DUMMY   SEGMENT STACK              ; AVOID LINK ERR MSG.
DUMMY   ENDS

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     1ØØH

START:  JMP     SHORT SETUP

;       INTERCEPT INT 16H AND LOOK FOR DELETE, BS

INT_16: CMP     AH,Ø               ; GET KEY?
        JNZ     EXINT              ; IF NOT, GO TO BIOS
        PUSH    BX                 ; ELSE, SAVE BX
GETKEY: PUSHF                      ; PUT FLAGS ON STACK
        PUSH    CS                 ; AND CS
        MOV     BX,OFFSET RETADR
        PUSH    BX                 ; AND RETURN ADDRESS
        JMP     SHORT EXINT        ; LET DOS GET KEY
RETADR: CMP     AX,BSCODE          ; BACKSPACE
        JNZ     NOTBS              ; NOPE
        CMP     CS:BYTE PTR MAPFLG,Ø ; MAP KEY?
        JZ      INTRET             ; NO
        MOV     AX,DELCODE         ; ELSE, USE DEL CODE
        JMP     SHORT INTRET
NOTBS:  CMP     AX,DELCODE         ; DEL CODE?
        JNZ     INTRET             ; NO
        XOR     CS:BYTE PTR MAPFLG,1 ; ELSE, TOGGLE FLAG
        XOR     AX,AX
        JMP     GETKEY             ; GET ANOTHER KEY
INTRET: POP     BX
        IRET

EXINT:  JMP     CS:DWORD PTR EXADR ; EXIT TO BIOS CODE

EXADR   DW      Ø,Ø                ; PUT EXIT ADDRESS HERE
MAPFLG  DB      Ø                  ; KEY MAP FLAG
        DB      Ø
ENDRES  LABEL   NEAR

;       SET UP INT 16H VECTOR SO THAT WE CAN LOOK FOR
;       BS, DEL

SETUP:  PUSH    DS
        XOR     AX,AX
        MOV     DS,AX              ; POINT TO INT. SEGMENT
        MOV     SI,16H*4           ; POINT TO INT 16 VECTOR
        LES     DI,DWORD PTR [SI] ; GET IT
        CLI                        ; KILL INTERRUPTS
        MOV     WORD PTR [SI],OFFSET INT_16
        MOV     WORD PTR [SI+2],CS ; INSERT NEW VECTOR
        POP     DS                 ; FIX DS
        MOV     EXADR,DI
        MOV     EXADR+2,ES         ; SET UP CHAIN TO OLD CODE
        STI                        ; FIX INTERRUPTS
        MOV     DX,OFFSET ENDRES ; END OF RESIDENT CODE
        INT     27H                ; EXIT, CODE RESIDENT

CODE    ENDS
        END     START
```

```
1Ø REM THIS PROGRAM CREATES FIXBS.COM
2Ø OPEN "O",1,"FIXBS.COM"
3Ø FOR I=1 TO 95
4Ø READ BYTE:PRINT #1,CHR$(BYTE);:NEXT I
5Ø CLOSE #1:SYSTEM
6Ø DATA &HEB,&H3C,&H8Ø,&HFC,&HØØ,&H75,&H2C,&H53
7Ø DATA &H9C,&HØE,&HBB,&H1Ø,&HØ1,&H53,&HEB,&H23
8Ø DATA &H3D,&HØ8,&HØE,&H75,&HØD,&H2E,&H8Ø,&H3E
9Ø DATA &H3C,&HØ1,&HØØ,&H74,&H14,&HB8,&H7F,&HØE
1ØØ DATA &HEB,&HØF,&H3D,&H7F,&HØE,&H75,&HØA,&H2E
11Ø DATA &H8Ø,&H36,&H3C,&HØ1,&HØ1,&H33,&HCØ,&HEB
12Ø DATA &HD7,&H5B,&HCF,&H2E,&HFF,&H2E,&H38,&HØ1
13Ø DATA &HØØ,&HØØ,&HØØ,&HØØ,&HØØ,&HØØ,&H1E,&H33
14Ø DATA &HCØ,&H8E,&HD8,&HBE,&H58,&HØØ,&HC4,&H3C
15Ø DATA &HFA,&HC7,&HØ4,&HØ2,&HØ1,&H8C,&H4C,&HØ2
16Ø DATA &H1F,&H89,&H3E,&H38,&HØ1,&H8C,&HØ6,&H3A
17Ø DATA &HØ1,&HFB,&HBA,&H3E,&HØ1,&HCD,&H27
```

Patch Page

This article presents patches to the HFM2 program, from HUG disk 885-3025-37, and the SCRNCLK program from disk 885-3014-37. A correction to the last Patch Page (April '85) is also presented, along with the source code for the FIXPRT program, which was accidentally left off of disk 885-3025-37. Note: the comments on some of the following source code segments are "chopped off" to fit the column width of this magazine.

HFM2 Patch

The HFM2 program on disk 885-3025-37 has an error that will not allow it to copy a file from one branch directory to another branch directory unless the full file name is specified. The problem can be corrected using DEBUG, as in this example:

```
A>DEBUG HFM2.COM
-E316C
xxxx:316C  ØD:F7
-W
Writing 385C bytes
-Q
```

If you wish, you can patch the source code for HFM2 by locating the procedure FIXNAM and patching the first 8 lines of it to look like this:

```
FIXNAM  PROC    NEAR
        MOV     DI,OFFSET BA.LINEBUF+2   ; POINT TO DEST. FIL
        MOV     CL,-1[DI]                ; GET COUNT
        XOR     CH,CH
FIXN1:  REPNZ   SCASB                    ; LOOK FOR CHAR IN A
        JNZ     NOCOL                    ; NOT FOUND
        CMP     BYTE PTR [DI],Ø          ; FOLLOWED BY ZERO?
        JNZ     FIXN1                    ; NO
```

SCRNCLK Patch

In the review of the Perks(tm) program that was printed in the July issue, I stated that the display produced Z-100 SCRNCLK program could not be turned off if it was loaded before Perks. This problem will also occur if SCRNCLK is loaded before any program that intercepts the software clock interrupt. A new version of SCRNCLK was released on disk 885-3029-37, but if you would like to patch the old version, you can by altering the source code as follows. First, locate the label LEND:, and add this code before it:

```
;       LOCAL SYSTEM INTERRUPT PROCESSOR

MYSYS:  CMP     AH,ØF2H                  ;CLOCK PROCESS CODE?
        JZ      CLKSYS                   ;YES
        JMPF                             ;ELSE, EXIT
SYSADR  DW      Ø,Ø
CLKSYS: OR      AL,AL                    ;KILL CLOCK?
        JNZ     SETCLK                   ;NO, SET IT
       MOV     CS:CLKFLG,Ø              ;DISABLE CLOCK
        IRET
SETCLK: MOV     CS:CLKFLG,1              ;ENABLE CLOCK
        IRET
LEND:
```

Now, under the label SETUP:, locate the line containing MOV DX, OFFSET LEND, and add this code:

```
        PUSH    DS
        XOR     AX,AX
        MOV     DS,AX                    ;DS AT Ø
        MOV     SI,OFFSET SYSINT         ;POINT TO SYSTEM INT
        LES     DI,DWORD PTR [SI]        ;GET VECTOR
        MOV     WORD PTR [SI],OFFSET MYSYS     ;PUT IN MY V
        MOV     2[SI],CS
        POP     DS
        MOV     WORD PTR SYSADR,DI       ;PUT OLD VECTOR HERE
        MOV     WORD PTR SYSADR+2,ES
        MOV     DX,OFFSET LEND           ;POINT TO END OF RES
```

With these changes, the file SCRNCLK.ASM can now be re-assembled to make a new SCRNCLK.COM. The CLOCK program must also be modified. Edit the source code, and replace all of the lines from START to END with these lines.

```
START:  MOV     SI,OFFSET CMDBUF         ;POINT TO COMMAND LN
        LODSB                            ;GET COUNT BYTE
        OR      AL,AL                    ;ANY ARGUMENT?
        JZ      CLKON                    ;NO, TURN CLOCK ON
SOS:    LODSB                            ;GET NEXT BYTE
        CMP     AL,' '                   ;SPACE?
        JZ      SOS                      ;IF SO, SKIP IT
        AND     AL,5FH                   ;CAPITALIZE
        CMP     AL,'O'                   ;OFF?
        JNZ     CLKON                    ;NO, TURN CLOCK ON
        MOV     AX,ØF2ØØH                ;CLOCK FUNC., CLK OFF
        INT     21H                      ;KILL CLOCK
EXIT:   INT     2ØH                      ;AND EXIT
CLKON:  MOV     AX,ØF2Ø1H                ;CLOCK FUNC, ENABLE
        INT     21H                      ;SET CLOCK
        INT     2ØH                      ;AND EXIT
CLK     ENDS
        END     START
```

Re-assemble CLOCK.ASM to make a new CLOCK.COM.

April Patch Page Correction

The source code at the bottom of page 54 in the April REMark contains a couple of errors, and should be changed to look like this:

```
        ADD     DI,3                     ;SKIP "MOV SI" INST.
CKIN:   CMP     Byte Ptr [DI],ØEAH       ;TEST FOR FAR JUMP
        JZ      MAPPED                   ;KEYMAP IS IN
        CMP     Byte Ptr [DI],ØFAH       ;MS-DOS 2.18 OR >?
        JNZ     NOTMAP                   ;NOT MAPPED, EXIT
       ADD     DI,1FH                   ;CORRECT FOR NEW VER
        JMP     CKIN                     ;AND CHECK AGAIN
MAPPED: INC     DI                       ;POINT TO KEYMAP ADD
```

The Missing FIXPRT Program

The FIXPRT program was somehow left off of disk 885-3025-37. This program will not be needed by most users of the disk, since it is for early releases of MS-DOS version 2 only, and many users have recieved updates. For those who are still using versions 2.13 or 2.15 of MS-DOS for the Z-100, here is the source code of the FIXPRT program

```
        PAGE    ,132
;       THIS PROGRAM FIXES THE PRINT OUTPUT OF MS-DOS
;       VERSIONS 2.13 AND 2.15 FOR THE Z-1ØØ

ADDR213 EQU     1FEØH                    ;PATCH ADDR FOR 2.13
ADDR215 EQU     1FF2H                    ;PATCH ADDR FOR 2.15
BIOSSEG EQU     4ØH                      ;BIOS SEGMENT

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
        ORG     1ØØH

START:  MOV     AX,4ØH
        MOV     ES,AX                    ;PUT ES IN BIOS SEG
        MOV     DI,OFFSET ADDR213        ;CHECK FOR 2.13
        CMP     ES:WORD PTR [DI],8Ø2EH
        JNZ     TRY215                   ;NOT 2.13
        CMP     ES:WORD PTR 2[DI],3DH
        JNZ     TRY215
        JMP     SHORT PATCH              ;INSTALL THE PATCH
TRY215: MOV     DI,OFFSET ADDR215        ;CHECK FOR 2.15
        CMP     ES:WORD PTR [DI],8Ø2EH
        JNZ     NOPATCH                  ;NOT 2.15
```

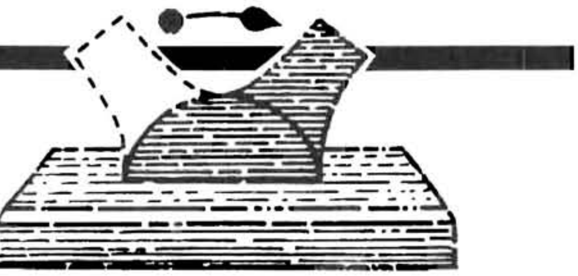
```
         CMP     ES:WORD PTR 2[DI],3DH
         JNZ     NOPATCH
PATCH:   MOV     SI,OFFSET NEWCODE          ;POINT TO NEW CODE
         MOV     CX,(OFFSET CODEND)-(OFFSET NEWCODE) ;SIZE
         CLD
         REP     MOVSB                      ;INSTALL THE PATCH
         MOV     DX,OFFSET GOOD
         MOV     AH,9
         INT     21H                        ;SAY PATCH INSTALLED
         INT     2ØH
NOPATCH:MOV      DX,OFFSET BAD
         MOV     AH,9
         INT     21H                        ;SAY PATCH NOT INST.
         INT     2ØH

;        THIS IS THE PATCH CODE

NEWCODE:CMP      AX,5
         NOP
         STC
         MOV     AX,5
         JZ      NEWC1
        NOP
         NOP
         NOP
         NOP
         NOP
         NOP
         NOP
         NOP
         CLC
NEWC1:   RET
CODEND:

GOOD     DB      'Patch installed.$'
BAD      DB      'Correct code not found.  BIOS'
         DB      ' not patched.$'
CODE     ENDS
         END     START
```

*



Switching Z-100 Hardware Printer Ports

William S. Hall
Mathematical Reviews
416 Fourth Street
Ann Arbor, MI 48104

Introduction

Although the Z-100 has two printer ports, one serial and one parallel, it is possible for the computer to talk to only one at a time via the MS-DOS PRN: device. Which of the two physical ports is connected to the logical PRN device is usually selected by running the CONFIGUR program. For someone needing both ports on a daily basis, this can be tiresome. At Mathematical Reviews we use our Z-100's in a Reviewer Assigning program, which requires having the AUX device connected to a network, a printer on the parallel port, and a barcode reader attached to the serial printer port. However, since the printer and barcode reader are not in use at the same time, we found we could drive both printer and reader by writing to and reading from PRN. All that was necessary was to find a way to switch PRN between the hardware ports without running CONFIGUR each time.

In this article we are going to share our experiences in writing such a routine in assembly language. We have also made it a bit more complicated than necessary. The reason is to help you set up some useful library functions for MS-DOS system calls and to help you with some of the difficulties of using segmentation on the 8088. In particular, we will show you a convenient way to write .EXE programs using the small memory model. If you know some 8080 or 8088 assembly language, this article will not be too difficult to follow. In addition, it will be enough to have MS-DOS version 1 with the MASM assembler to carry out the program we present.

Description Of The Program

SETPRN.EXE can do several useful things. Run by itself with the command 'SETPRN', it will tell you to which hardware port the PRN device is currently set. If the 'L' (load) switch and a file name are given on the command line, the program will look for that file and alter the current configuration of the PRN device according to the data in that file. Finally, the 'M' (make) switch and a file name will record the current configuration.

As an illustration of its use, suppose we run CONFIGUR and set the PRN device for a parallel printer, say an Okidata. Now give the command 'setprn/m okidata.dat' to record the data. Next, run CONFIGUR again, this time setting the port for a Diablo on the serial port. A suitable command is 'setprn/m diablo.dat'. Now execute 'setprn/l okidata.dat' to reload the parallel port configuration and 'setprn/l diablo.dat' to use the Diablo printer on the serial port. If you forget which port, serial or parallel, is in use, just say 'setprn' and the current status will be reported.

It is easy to write a batch file to include 'setprn' as part of a set of commands to list files to the desired printer. At Mathematical Reviews, we have a customized data file for the serial port to accommodate an Intermec bar code reader. This is used during the assignment phase when we are running the assigning program, and a batch file is used to set the port and to run dBASE II. Assignments are read directly into dBASE II files using the barcoder. (This itself is a neat trick.) Later when the assignments have been made, the results are printed to an Okidata 82 using a simple batch file which first connects PRN to the parallel port before executing the other supporting programs.

The Small Memory Model On The 8088

There are many similarities between the 8088 and its predecessors, the 8080 and 8085. Basically, the 8080 has an 8-bit wide data path and a 16-bit wide address space. This means that 2**16 or about 64K different addresses can be located by the 8080. In the 8088, both the data and basic address lines remain 8 and 16 bits wide, respectively. To address more memory, four additional lines have been added bringing the addressing capability up to 2**20, or about one million bytes. If you will imagine the memory area of a computer as a line of boxes, each able to hold a byte of data, with locations beginning from 0 to 2**20, it is easy to see how the 8088 locates one of these boxes. First, a base point is chosen along this line which is within 64K of the box to be located. Then an offset value is added to the base point. Hence an absolute location in memory can be described in terms of a base point and an offset value. The top four address lines tell you something about the base point, and the offset value is determined by the first sixteen address lines.

In the 8088, the base point is given by the value in a segment register; the offset address is then a 16 bit value just as in the 8080.

For example, suppose a segment register contains 56FF hex and the offset address is 3123 hex. The absolute address in the memory store is then found by shifting the segment register 4 bits left and adding this to 3123. A four-bit left shift is equivalent to placing a '0' on the right of the hex address. Thus the absolute location from 0 of 56ff:3123 is 56FF0 + 3123 = 5A113 hex = 368,915 decimal.

In the 8080, there is only one 64K segment which can be addressed. Hence a program for the 8080 consists of a linear store of bytes, some of which are executable codes and some of which are data. So the 8080 has one 64K segment, and code, data, and the stack lie there.

In the 8088, the situation is more complicated. At any time, the chip can have four active segments. These are called the code, data, stack, and extra segments. The segments can be disjoint, overlapping, or coincident. If the 8088 is fetching an instruction located at a certain offset value in its segment, it uses the value of the code segment register to determine its base point. If it is referencing certain types of data, it uses the data or extra segments registers to determine the base point, and references to the stack, such as 'push' and 'pop', start with the stack segment register. By now you probably have the idea that we can make the 8088 look just like the 8080 as far as memory is concerned by making all segment registers equal in value. On the other hand, we can use some of the power of the chip by making these segments different. In this way we can write larger, more versatile programs.

Now, the program we are going to present here would work fine in the 8080 model. But it is instructive to see how another memory model can be used. This is the 'small' memory model where up to four different, possibly exclusive segments are used, namely a code segment, a data segment, an extra segment, and a stack segment. It is easy to set up; in fact, the very first thing we are going to present are some macros to make it easier. Don't panic at the word 'macro' if you have never seen it before. Many assemblers and compilers are also text processors. You define a certain string and give it a name. When the assembler or compiler sees the name, it substitutes the string for that name. It's that simple, and it can make programming easier to perform and understand.

The Segmentation Macros

Every piece of a program written in assembly language for the 8088 has to be inside some kind of segment declaration. The declaration is

```
segname SEGMENT [align] [combine] ['class']
                          .
                          .
                          .
segname ENDS
```

'Segname' is any unique name assigned by the programmer. 'Align' describes the boundary on which the segment begins. A segment can begin on a byte, word, paragraph, or page boundary. 'Combine' tells the linker how to arrange the segments of a particular class name. A segment may be private, and hence cannot be combined with another segment, or it may be public and hence can be combined with other segments with the same name and class name. In the latter case, there is only one base address for all such segments, and the offset is measured from the beginning of the first segment through the last one loaded. A combine class may also be of type 'stack'. In this instance, it is treated as public, but the stack pointer is set to the first address of the first stack segment. Other possibilities are 'common', 'memory', and 'at', but these will not concern us here. Finally, class names may be any valid name. A class is a collection of segments, and the inclusion of segments in a class controls the order and relative placement of segments in memory.

Just before the SEGMENT declaration, it is useful to have a group statement,

```
name     GROUP    segname
```

A group is a collection of segments whose total length is less than 64K. The group directive is also used by the linker to know which segments to load together. (The order, however, is determined by the class statement discussed above as well as the order in which modules are supplied to the linker.)

Finally, the SEGMENT directive is followed by an ASSUME statement.

```
ASSUME  segreg:segname
```

ASSUME tells the assembler that the symbols in the segment or group can be accessed by using this segment register. For example, 'ASSUME cs:code' allows the assembler to assume that any references to the code segment cs refer to the segment named 'code'.

Unfortunately, all of this is a lot to remember when writing assembly code. It makes it easier to put all these declarations together into a macro. Here is an example.

```
pseg    macro   align
pgroup  GROUP   prog
prog    SEGMENT align public 'prog'
        ASSUME  cs:pgroup
        endm
```

Now inside your program it is enough to say 'pseg para' to get the three lines of declarations

```
pgroup  GROUP   prog
prog    SEGMENT para public 'prog'
        ASSUME  cs:pgroup
```

For the small memory model, such a declaration does everything we want in setting up the start of a segment where code goes. In each separate module we write, (and in our program, we shall have several), we can use a similar declaration. Of course, each module has its own code segment, but when the modules are finally put together by the linker, we will effectively have only one code segment which can be accessed via offsets measured from the value of the code segment register determined by the location of the first segment of the group.

We might mention that our macro has a parameter, namely the 'align' value. For the code segment in the main program, we will choose a paragraph alignment. This will be the beginning of the code segment in the final program. In the separate library modules, a byte alignment will be chosen. In this way, all segments can be placed adjacent to one another, which keeps code size smaller.

Now look at the listing for 'segmacro.mac' and you will see all the segment macros we shall use. Note that the extra and data segment macros are quite similar to the one just described. The stack macro has as its free parameter the size of the stack. It is unusual to have more than one stack segment, so the align parameter is not important. But the size is. MS-DOS wants at least 80h bytes of stack space for system calls. 100h bytes will suffice for our simple program. Finally, several short macros provide us with a simple way to end each of the segments.

If you have read this far, you have been through some of the most difficult part of the discussion. Take a short break and type in the segmacro file before continuing.

Some MS-DOS System Calls

A system service in MS-DOS, such as writing to the terminal or opening a disk file, can be made by placing the appropriate number into the AH register of the 8088 and executing the software interrupt number 21h. Of course, other registers have to be prepared; which ones depend on the call. Take a look at the listing for 'outstr'. This routine prints a string which is terminated by a '$' sign on the console. Let us read through the program.

A title directive, giving a descriptive string of the module is followed by the name directive. The name is very useful when building a library of such programs. Next are some comments explaining the entry and exits states of the CPU when the program is called. Note in particular that the DX register must be loaded with the address of the string to be written before calling the routine.

Next, two header files, one of which we have discussed, must be included during the assembly. The second file, 'dosifunc.h' contains a list of manifest constants for the system functions we shall need. It is short, and it might be wise to type in and check this file at this time.

Following the include directives we see the 'pseg' macro invoked with a 'byte' alignment parameter. Then, the name of the routine, 'outstr' is declared public. This makes 'outstr' available to other modules. In particular, the main program, which needs to call 'outstr', will be able to find it.

Outstr is a procedure, and the next line so declares it. It also tells the assembler that it is a 'near' procedure. Here comes more about segmentation. The only way the code segment can be changed is by a jump or call to another code segment or a return from one. The programmer cannot alter it directly. Such a jump, call, etc. is called a far jump, etc. But in our model, we will only have one final code segment. Hence, when the main program calls 'outstr', it does not have to make a far call but rather a near call. Now a far call not only places the offset address on the stack for the next instruction to be executed upon its return, it also puts the current code segment there. But a near call need only put the offset value on the stack. Hence the return has to pop the segment and offset on a far return but only the offset on a near return. By declaring the procedure to be near, the assembler will assemble the correct return code when it sees 'ret', in this case a near return.

The actual code is quite simple. Move the system service number into AH, do an interrupt 21H, and return (near). The remaining directives note the end of the procedure (endp), the end of the code segment (the macro 'endps'), and an 'end' directive to the assembler. Try typing the program into the file 'outstr.asm'. When done, run the assembler with the command MASM. Answer the questions as follows: The source file is 'outstr'; accept MASM's default for the .OBJ file; get listing and cross-reference files only if you wish. (For the latter you will then have to run CREF. Both can be very instructive to study.) Later we will put the object file generated into a library.

If you have errors, look over your files and compare them carefully with the text. Fix any problems and again assemble.

Look now at the remaining MS-DOS function files. They are very similar in design and structure to 'outstr'. Now would be a good time to type in 'opfile' (open a file), 'seqread' (sequential read), 'crfile' (create a file), 'seqwrite' (sequential write), and 'clfile' (close file). Assemble each one as just described.

Making A Library

You should now have six object files, one for each of the MS-DOS system calls just described. We are now ready to combine them into a library. Run LIB. Answer the name request with SYSCALLS and say yes to the 'create' question. The operations string is now

```
+outstr+seqread+seqwrite+clfile+crfile+opfile
```

which will add each of the object files we have created to SYSCALLS. Finally, answer SYSCALLS.LST to the listing file question. When LIB exits you will have SYSCALLS.LIB and SYSCALLS.LST in your directory. Later when we finally link the main program, the linker can search the library we just created to obtain the modules needed. It is also easy to add more functions to the library as well as to remove them. The documentation on LIB provides the necessary information.

Zenith Bios Calls For Character Devices

MS-DOS provides a limited number of system calls for manipulating the PRN, CON, and AUX logical devices and the associated hardware. Zenith has expanded on these by means of special subroutines which can be used by the programmer. We are interested in using the 'bios_prnfunc' call to read and modify a table containing the information for the printer ports.

In fact, these bios functions can do many things including reading from a port, writing to a port, checking port status, and changing the status. Not very clear explanations of their uses are given in the ZDOS 1 manual in an Appendix. We won't try to show how to use all of them, but rather just the two we need.

Look at the listing for the module 'cdgetprn.asm' (character device subroutine call, get printer data). Note the entry requires that the BX register point to a table in the extra segment of the program. When the routine exits, the table is filled in with the current data. For this we need two additional include files. The first, 'biosseg.h' defines a piece of the segment where the device function calls can be found. This is necessary, since instead of performing an interrupt, a far call will be made to an address in the bios segment. In addition, before making the call, the type of call and any subfunction must be set in the AH, AL pair. These manifests are contained in 'chrdev.h'. Finally, the segmentation macros are required as usual.

At this time, you should type in this and its companion 'cdsetprn.asm', along with the necessary include files and assemble. When all errors have been corrected, place the object files into a library called BIOSCALL by running LIB and performing the operations '+cdgetprn+cdsetprn'. Again, obtain a listing file BIOSCALL.LST. Check this file to see that now it contains the two functions just prepared.

A Remark On The Include Files

Except for the files 'segmacro.mac' and the soon-to-be discussed file 'devtbl.h', all the header files used so far have been extracted from various .ASM files supplied by Zenith in the ZDOS 1 release or the MS-DOS 2 Programmer's Utility Pack. We have given them explicitly since not everyone will have access to the original .ASM files. If you are serious about your assembly language programming, then use the corresponding Zenith files instead of the ones we have presented. Assembly will be slower since they are bigger and hence will require more symbol table space, but their use will reflect the current Zenith standard. Since the names of the Zenith files vary some between releases, you will have to search them to find out which ones you need. Their names should be suggestive, such as CHRDEV.ASM, BIOSSEG.ASM, etc.

The Main Program

Now we can start discussing the main program. There is still a lot to cover. It may require more than one reading, since we need to show you how to set up an exit, how to establish code, data, and extra segments, how to use a file control block to open a file, and how to use the disk transfer address to read and write a file. Along the way, we shall describe how MS-DOS loads and executes a program. It is a bit more complicated than CP/M, but since DOS inherited much from CP/M, many of you will have a basis for understanding what follows.

Look at the program 'setprn.asm'. For the moment skip over the constants defined at the beginning and let us discuss the files which are included in the assembly. As usual there is the segment macro file. The new one is 'devtbl.h'. Do you remember that a table is needed in order to read or change the port information? We have given this table in the form of a structure declaration. A structure is simply a way of keeping together under one name various items which are naturally associated. The analogy in Pascal is a record, in C it is also called a structure. A structure declaration does not in itself reserve space but simply creates a template. The fields in the device table are commented and are self-explanatory. All are byte-sized except the port address.

Reading further down the program we see a macro 'setseg' to be discussed later. Next we declare 100h bytes of a stack segment with 'sseg' (from 'segmacro.mac'). This is followed by the beginning of the extra segment, which we align on a paragraph. Following is the declaration of the table 'prntbl' inside the extra segment. Here we use the template 'devtbl' to create the table needed by the Zenith print function bios calls, which must be in the extra segment. The '<>' is essential to the assembler. It tells it to make the space with the default values given in the structure declaration. By appropriate substitutions, these values can be changed for some of the fields. See the MASM manual on structures.

The data segment then follows. Note that its contents consist solely of message strings, terminated by newlines and the '$' sign marking the end of the string as needed by 'outstr'.

So now we arrive at the main program. First a code segment declaration is made, then the main program is declared as a far procedure. Why do we do this? Keep in mind that it is MS-DOS which calls the program. To do so, it must load it into memory and then jump to its entry point. But, of course, MS-DOS is running in its own code segment. The only way to change code segments into the program is by a far jump, call, or return. Hence main is a far procedure.

Next, the library routines are declared as external and near. Hence the assembler will generate near calls for them. Later, the linker will give values to their addresses when it loads them.

Now comes more stuff connected with segmentation and the way MS-DOS enters a user program. The particular code segment is determined by the operating system. The starting offset is determined by the label following the end statement in the file, in this case 'start'. The data and extra segments are set to the beginning of the program segment prefix (PSP), a 256-byte piece of memory containing all sorts of useful information. The first two entries are the codes for an interrupt 20h. To return to MS-DOS, the program only has to perform this interrupt, but the code segment register MUST have the value of the data segment when the program begins. Since we can change the code segment only with a far call, return, etc., we set up the stack so that a return from main will accomplish this. We do this by pushing the value of DS (the data segment register) onto the stack followed by the offset location (zero) of the interrupt 20 in the PSP. Since main is a far procedure, any return from there will pop the first word on the stack (in this case zero) into the instruction pointer (IP) and the second word into CS. The CPU will then execute the instruction found there, namely interrupt 20h, and will hence return to DOS gracefully. If you would like to practice crashing the system, just perform an interrupt 20h without first setting the CS register as we have described.

The next two lines show how to change the extra entry to the one in our program. Note that we have not yet changed the data segment. At the moment it remains as handed to us by MS-DOS and points to the bottom of the PSP. There are two locations there of considerable interest. One is at offset 5Ch, and the other is at 80h. CP/M programmers will recognize these numbers.

When MS-DOS starts a program, the command line minus the program's name is placed in the PSP at offset 80h (DEFDTA) in the data segment. The first byte is the count of the number of elements on the command line. In our program, if we just type 'setprn', then the value at DEFDTA will be 0. Hence we jump to the part of the program which examines the tables and reports the current physical port which is connected to PRN. You can see this in the jump to the label 'show'.

Otherwise, the next byte (at offset DEFDTA+1) is examined for a valid switch character '/'. If found, the program continues. If not, DX is loaded with the address of a message to print and a jump is made to 'error'. After the message is displayed, the program then falls through to 'show' and exits.

Look down the program far enough to find 'show'. Note that the very first action is to execute the short macro 'setseg', which is found at the beginning of the program. Note that the parameters given to the macro are the name of the segment and the segment register. In this case we set DS to the value of OUR data segment. Up to now, we have used the value given to us by MS-DOS since all references have been to the PSP. Now, however, we are going to print some messages whose addresses are in our data segment. In order to do so, we must set the base of the segment first.

Returning to where we left off in processing the switch, we next get the switch character (at offset DEFDTA+2). It is converted to upper case with the routine MCU, which is given in the file 'mcu.asm'. If you are tired of reading for a while, type this one in and assemble it. You will see later how to link it with SETPRN.

The program continues with checks for either an 'L' or an 'M'. If not found, error messages are generated. Otherwise, it takes action according to the value of the switch by jumping to 'make' if 'M' is specified or to 'change' if 'L' is given.

Creating And Writing A Disk File

At this point, your attention should be at the label 'make'. Our goal is to create a file, fill the device table with the current values describing PRN, and write this file to disk.

To create a file in MS-DOS (using the old, CP/M-type system services), one must point the dx register at a so-called file control block (FCB) and call 'crfile', which you should have already placed in your library. FCB's require a lot of explanation, and you can look at your documentation to read more about them. Basically, however, a FCB is a 36-byte string describing the disk file under consideration. The 0th byte has a drive letter in it, coded as 0 for the default drive, or 1 for A:, 2 for B:, etc. The next 8 bytes hold the file name, blank padded on the right if necessary, the next 3 the extension, and so on. When MS-DOS starts a program, it looks on the command line for any string which might be a file name and makes a FCB from it beginning at offset 5C hex in the PSP. If a second file name is seen, it puts its FCB at 6C hex. These can be used by the programmer. These FCB's, by the way, are called unopened, since the system has not yet tried to use them to open or create a file.

In our case, we expect to see a single file name at 5C hex. Hence before calling 'crfile', we load DX with the offset value DEFFCB1 (= 5C). If the call is successful, the FCB is filled in by MS-DOS with various parameters and a 0 is returned in AL. A non-zero value indicates an error of some kind, and our action is simply to point to the appropriate message and jump to 'error'. Note that any unopened FCB at 6C hex will be partly overwritten if the call is successful. This does not concern us here since we are only interested in one file name. However, the programmer has to move the FCB at 6C before using the one at 5C if both are needed.

One of the parameters in the opened FCB is at offset 20 hex from the beginning of the FCB. This is the current record number. The record size defaults to 80h bytes. Since our data will have only 16 bytes in it, we will be writing to the first record of the file. Hence, one of our first actions is to set CURREC (= 20h) to zero. This is an essential step; the operating system allows you to determine to which record you want to write. Failing to set this parameter during a file opening or creation can result in mysterious errors in your programs. However, once the record number has been set, each sequential read or write of a record of the file will increment the record number by one.

After setting CURREC, we next set the file size. This is a two-byte parameter. All we have to do is to set RECSIZ1, the low-order byte to 16. RECSIZ1 and RECSIZ2 are located at offsets 0E and 0F hex from the beginning of the FCB, respectively.

Next, we point BX at 'prntbl' in the extra segment and call 'cd_getprn' to load the table with the current configuration. A carry is set in case of failure, and in that event, an error is noted. Otherwise, 16 bytes are moved from the now filled-in table to DEFDTA.

To write to a disk file, MS-DOS expects the record to be written to be located at the disk transfer address (DTA) and for DX to be pointed at the applicable, opened FCB. The DTA can be set by the programmer; MS-DOS enters a program with DTA set to offset 80h in the PSP. That's why we put the 16 bytes from the table at DEFDTA (= 80h).

Having set all parameters, we can now write the record using 'seqwrite'. We then close the file with 'clfile'. Any errors cause the usual printing of a message and an exit. Otherwise, we announce success and exit. Again notice (at label 'wrtok'), the shift to our data segment before attempting to write the string at 'wrtfil'.

Reading A Disk File

Let us look at the program beginning at the label 'change'. To

Program Listings

Segmacro.mac

```
;; Segment definition macros for the small memory model

; Data segment

dseg    macro   align
dgroup  group   data
data    segment align public 'data'
        assume  ds:dgroup
        endm

; End data segment

endds   macro
data    ends
        endm

; Extra segment

eseg    macro   align
egroup  group   extra
extra   segment align public 'extra'
        assume  es:egroup
        endm

; End extra segment

endes   macro
extra   ends
        endm

; Code segment

pseg    macro   align
pgroup  group   prog
prog    segment align public 'prog'
        assume  cs:pgroup
        endm

; End code segment
endps   macro
prog    ends
        endm

; Make a stack

sseg    macro   num
stack   segment para stack 'stack'
        db      num&h dup(?)
stack   ends
        endm
```

Dosifunc.h

```
;; msdos function call definitions

; Define the interrupts
```

open a file for reading we use the 'opfile' call. If the file cannot be found, a non-zero value is returned in AL and we exit with an error. Otherwise, as before, we set CURREC to zero before executing a read with 'seqread'. In this case we expect less than the default record size of 80h bytes to be read.

During a read, a record will be moved from the disk to the DTA at 80h. The 16 bytes read are then moved into 'prntbl' and then 'cd_setprn' is called to alter the PRN device. Error handling is very similar to the discussion above. Again we emphasize the importance of setting CURREC before trying to read the file.

Showing The Status Of PRN

At label 'show', we switch to our own data segment and examine the contents of 'prntbl' after first filling it by calling 'cd_getprn'. Look again at 'devtbl.h', and in particular at the field 'class'. If this value is 0, then PRN points to the console. If the number there is 1, then PRN is attached to a serial port. Finally the value 2 means the parallel port is being referenced by PRN. The program ends by printing the appropriate message and exiting via a return from main.

Notice how a field in a structure is referenced. The structure's name is followed by a period and the field name, just as in Pascal or C. Consider also how powerful the idea of a structure can be. We can define any number of data structures using a single template, then each field can be referenced as 'struct_name. field_name'. This can be very difficult to simulate in FORTRAN or COBOL or with more conventional assemblers.

Linking SETPRN

Now that you understand the main program, type it in and assemble it using MASM just as you have done for the other modules. To generate the final program run LINK. The answers to the questions are:

```
object files:  setprn+mcu
run file:  setprn
list file:  NUL (or setprn)
libraries: syscalls+bioscall
```

and now you are done. You might like to look at the listing file and try the various options given in the LINK manual. It will help you to learn more about segmentation and how a program is loaded in MS-DOS, about which this short article can be but a brief introduction.

• NEXT MONTH •

A Program To Help You Locate

HALLEY'S COMET

```
dosi_func       equ     21h             ; Perform a function

; Define the function numbers

;dosf_term      equ     0               ; program terminate
;dosf_conin     equ     1               ; console input
;dosf_conout    equ     2               ; console output
;dosf_auxin     equ     3               ; aux input
;dosf_auxout    equ     4               ; aux output
;dosf_printout  equ     5               ; printer output
;dosf_drcio     equ     6               ; direct console I/O
;dosf_drci      equ     7               ; direct condole input
;dosf_drcine    equ     8               ; console input (no echo)
dosf_outstr     equ     9               ; output string
;dosf_instr     equ     10              ; input string
;dosf_stcon     equ     11              ; status of console
;dosf_coninf    equ     12              ; flush keyboard buffer and input
;dosf_rsdisk    equ     13              ; disk system reset
;dosf_seldisk   equ     14              ; select default disk
dosf_opfile     equ     15              ; open file
dosf_clfile     equ     16              ; close file
;dosf_srhfi     equ     17              ; search for first
;dosf_srhnx     equ     18              ; search for next
;dosf_defile    equ     19              ; delete file
dosf_seqread    equ     20              ; sequential read
dosf_seqwrite   equ     21              ; sequential write
dosf_crfile     equ     22              ; create file
;dosf_refile    equ     23              ; rename file
;dosf_getdisk   equ     25              ; get default disk
;dosf_sdioa     equ     26              ; set disk i/o address (DTA)
```

Outstr.asm

```
        title   outstr - output string
        name    outstr

;       entry - (dx) = offset pointer to string
;       exit - none

        include dosifunc.h
        include segmacro.mac

        pseg    byte
        public  outstr
outstr  proc    near
        mov     ah,dosf_outstr
        int     dosi_func
        ret
outstr  endp
       endps
        end
```

Opfile.asm

```
        title   opfile - open a disk file
        name    opfile

;       entry - ds:dx points to unopened FCB.
;       exit - (al) = FF if failure, ØØ if success.

        include dosifunc.h
        include segmacro.mac

        pseg    byte
        public  opfile
opfile  proc    near
        mov     ah,dosf_opfile
        int     dosi_func
        ret
opfile  endp
        endps
        end
```

Seqread.asm

```
        title   seqread - read a block from an open file
        name    seqread

;       entry - ds:dx points to opened FCB
;       exit - (al) = ØØ if success
;                   = Ø1 EOF no data in record
;                   = Ø2 not enough room in transfer segment
;                   = Ø3 EOF partial record, Ø filled

        include dosifunc.h
        include segmacro.mac

        pseg    byte
        public  seqread
seqread proc    near
        mov     ah,dosf_seqread
        int     dosi_func
        ret
seqread endp
        endps
        end
```

Crfile.asm

```
        title   crfile - create a file
        name    crfile

;       entry - ds:dx points to unopened FCB.
;      exit - (al) = FF if no empty entries,
;                   = ØØ if success, open FCB created.

        include dosifunc.h
        include segmacro.mac

        pseg    byte
        public  crfile
crfile  proc    near
```

```
        include biosseg.h
        include chrdev.h
        include segmacro.mac
        .list

        pseg    byte
        public  cd_getprn
cd_getprn proc  near
        mov     ah, chr_status
        mov     al, chr_sfgc
        call    bios_prnfunc
        ret
cd_getprn endp
        endps
        end
```

Cdsetprn.asm

```
        title   cdsetprn - set parameters of prn device
        name    cdsetprn

;       entry - es:bx has offset address of prn device table
;               rest of table set as needed
;       exit - port table and device updated
;       The device table is defined in the include file 'devtbl.h'.


        .xlist
        include biosseg.h
        include chrdev.h
        include segmacro.mac
        .list

        pseg    byte
        public  cd_setprn
cd_setprn proc  near
        mov     ah, chr_control
        mov     al, chr_cfsu
        call    bios_prnfunc
       ret
cd_setprn endp
        endps
        end
```

Biosseg.h

```
; biosseg.h

bios_seg segment at 4Øh

        org     24*3
bios_dskfunc    label   far             ; disk function
        org     25*3
bios_prnfunc    label   far             ; printer function
        org     26*3
bios_auxfunc    label   far             ; aux function
        org     27*3
bios_confunc    label   far             ; console function

bios_seg ends
```

```
        mov     ah,dosf_crfile
        int     dosi_func
        ret
crfile  endp
        endps
        end
```

Seqwrite.asm

```
        title   seqwrite - write sequentially to a disk file
        name    seqwrite

;       entry - ds:dx points to opened FCB.
;       exit - (al) = ØØ - success,
;                     Ø1 - disk full,
;                     Ø2 - not enough room in DTA.

        include dosifunc.h
        include segmacro.mac

         pseg   byte
         public  seqwrite
seqwrite proc   near
         mov    ah,dosf_seqwrite
         int    dosi_func
         ret
seqwrite endp
         endps
         end
```

Clfile.asm

```
        title   clfile - close a file
        name    clfile

;       entry - ds:dx points to opened FCB.
;       exit - (al) = FF if failure, ØØ if success.

        include dosifunc.h
        include segmacro.mac

        pseg    byte
        public  clfile
clfile  proc    near
        mov     ah,dosf_clfile
        int     dosi_func
       ret
clfile  endp
        endps
        end
```

Cdgetprn.asm

```
        title   cdgetprn - get parameters of prn device
        name    cdgetprn

;       entry - es:bx has offset address of prn device table
;       exit - prn device table filled in.
;       Device table is defined in the include file 'devtbl.h'.

        .xlist
```

Chrdev.h

```
; devtbl.h

chr_write       equ     Ø               ; write function
chr_read        equ     chr_write+1     ; read function
chr_status      equ     chr_read+1      ; status function
chr_sfgs        equ     Ø               ; get status subfunction
chr_sfgc        equ     chr_sfgs+1      ; get configuration information
chr_control     equ     chr_status+1    ; control function
chr_cfsu        equ     Ø               ; set up new configuration
chr_cfci        equ     chr_cfsu+1      ; clear input subfunction
chr_cfco        equ     chr_cfci+1      ; clear output subfunction
chr_look        equ     chr_control+1   ; non-destructive read function
chr_fmax        equ     chr_look        ; maximum function number
```

Devtbl.h

```
; table for accessing character device port parameters

devtbl  struc
class   db      Ø               ; device class
attr    db      Ø               ; attributes (parity/case mapping)
port    db      Ø               ; port number
        db      Ø
baud    db      Ø               ; baud rate
hshk    db      Ø               ; handshaking protocol
bctl    db      Ø               ; stop bits/parity/char length
ecnt    db      Ø               ; ETdev/ACK information
ncnt    db      Ø               ; number of nulls to send.
nchr    db      Ø               ; char after which to send null
res     db      6 dup (Ø)       ; reserved
devtbl  ends
```

Mcu.asm

```
        title   mcu - map character to upper case
        name    mcu

;       entry - character in al
;       exit - 'a' to 'z' mapped to upper case in al

        include segmacro.mac

        pseg    byte
        public  mcu
mcu     proc    near
        cmp     al,'a'
        jc      mcul
        cmp     al,'z'+1
        jnc     mcul
        sub     al,' '
mcul:   ret
mcu     endp
        endps

        end
```

Setprn.asm

```
        title   setprn - configur PRN according to input file

DEFDTA  equ     Ø8Øh            ; default disk transfer address (DTA)
DEFFCB1 equ     Ø5ch            ; default file control block (FCB)
CURREC  equ     Ø2Øh            ; current record number in FCB
RECSIZ1 equ     14              ; LSB of record size in FCB
RECSIZ2 equ     15              ; MSB of record size in FCB
cc_cr   equ     13              ; carriage return
cc_lf   equ     1Ø              ; line feed

        include segmacro.mac            ; segment macro
        include devtbl.h                ; device table structure

setseg  macro   name, reg               ;; set a segment macro
        mov     ax,name
        mov     reg,ax
        endm

        sseg    1ØØ                     ; make a stack

        eseg    para                    ; create extra segment
        prntbl  devtbl  <>              ; make space for the table
        endes

        dseg    para                    ; create data segment
serprt  db      'PRN device attached to serial port', cc_cr, cc_lf,'$'
parprt  db      'PRN device attached to parallel port', cc_cr,cc_lf,'$'
conprt  db      'PRN device attached to console port', cc_cr,cc_lf,'$'
unknown db      'PRN device not attached to external port', cc_cr,cc_lf,'$'
notfnd  db      'Configuration file not found', cc_cr, cc_lf,'$'
rderr   db      'Error in reading configuration file', cc_cr, cc_lf, '$'
deverr  db      'Error in setting port', cc_cr, cc_lf, '$'
rddev   db      'Error in reading port data', cc_cr, cc_lf, '$'
wrtfil  db      'Writing new configuration file', cc_cr, cc_lf,'$'
crerr   db      'Unable to create new configuration file', cc_cr, cc_lf, '$'
wrterr  db      'Error in writing new configuration file', cc_cr, cc_lf, '$'
nochar  db      'Wrong or no character specified in switch', cc_cr, cc_lf, '$'
badswt  db      'Wrong or no switch given', cc_cr, cc_lf, '$'
        endds

        pseg    para                    ; create program segment
main    proc    far
        extrn   cd_getprn:near          ; get printer device config
        extrn   cd_setprn:near          ; change printer device config
        extrn   outstr:near             ; print string
        extrn   opfile:near             ; open file
        extrn   seqread:near            ; sequential read
        extrn   crfile:near             ; create a file
        extrn   seqwrite:near           ; sequential write
        extrn   clfile:near             ; close file
        extrn   mcu:near                ; map to upper case
start:
        push    ds                      ; set up for exit to system
        mov     ax,Ø
        push    ax
        mov     ax,extra                ; set up extra segment
        mov     es,ax
        mov     bx, offset DEFDTA       ; anything on command line ?
```

```
        cmp     al,Ø                    ; no errors
        jz      wrtok
        mov     dx, offset wrterr       ; error in closing file
        jmp     error
wrtok:
        setseg  data,ds                 ; now switch to our data segment
        mov     dx, offset wrtfil       ; write out the information
        call    outstr                  ; back to MS-DOS
        ret

;; Come here to change the printer data

change:
        mov     dx, offset DEFFCB1      ; yes, open file
        call    opfile
        cmp     al,Ø                    ; success, found file
        jz      read
        mov     dx, offset notfnd       ; not found
        jmp     error
read:
        mov     bx, offset DEFFCB1      ; point to FCB
       mov     al,Ø                    ; set current record to zero
        mov     [bx+CURREC],al
        call    seqread                 ; read 8Øh bytes
        cmp     al,Ø3                   ; should show partial read
        jz      readl
        mov     dx, offset rderr        ; something is wrong
        jmp     error
readl:
        mov     cx,16                   ; get 16 bytes
        mov     di,offset prntbl        ; point to print table
        mov     bx, offset DEFDTA
lread:
        mov     al,[bx]                 ; move bytes in sequentially
        mov     es:[di],al
        inc     bx
        inc     di
        loop    lread                   ; repeat until 1Ø are moved
        mov     bx, offset prntbl       ; now set the device
        call    cd_setprn
        jnc     show                    ; no carry, no errors
        mov     dx, offset deverr       ; unable to set
error:
        setseg  data, ds                ; set our data segment
        call    outstr
        jmp     showl

;; come here to show the results

show:
        setseg  data, ds
showl:
        mov     bx, offset prntbl       ; point to table
        call    cd_getprn               ; see what it says
        cmp     prntbl.class,Ø          ; Ø is the console
        jz      consol
        cmp     prntbl.class,1          ; 1 means a serial device
        jz      serial
        cmp     prntbl.class,2          ; 2 is the parallel device
```

```
next:   mov     al,[bx]
        cmp     al,Ø
        jnz     next                    ; yes, see what it is
        jmp     show                    ; no, just show the status

next:   mov     al,[bx+1]
        cmp     al,'/'                  ; switch?
        jz      nextl                   ; yes, so make a new data file
        mov     dx, offset badswt       ; wrong or no switch
        jmp     error

nextl:  mov     al,[bx+2]               ; get the switch character
        call    mcu                     ; map to upper case
        cmp     al, 'L'                 ; load a data file ?
        jnz     next2                   ; no
        jmp     change                  ; yes, change data

next2:  cmp     al, 'M'                 ; make a new data file ?
        jz      make                    ; yes
        mov     dx, offset nochar       ; bad or no switch character
        jmp     error

make:   mov     dx, offset DEFFCB1      ; create a file
        call    crfile
        cmp     al,Ø                    ; no errors if Ø
      jz        makel
        mov     dx, offset crerr        ; print error
        jmp     error

makel:  mov     al,Ø                    ; set record number to write to zero
        mov     bx, offset DEFFCB1
        mov     [bx+CURREC], al
        mov     al,16                   ; set size of file in FCB to 16
        mov     bx, offset DEFFCB1
        mov     [bx+RECSIZ1],al
        mov     bx, offset prntbl
        call    cd_getprn               ; get the data for the port
        jnc     make2
        mov     dx, offset rddev        ; error in reading the port
        jmp     error

make2:  mov     cx,16                   ; get 16 bytes
        mov     di,offset prntbl        ; point to print table
        mov     bx, offset DEFDTA       ; point to DTA

lstore: mov     al,es:[di]              ; move the data
        mov     [bx],al
        inc     bx
        inc     di
        loop    lstore                  ; repeat until 1Ø are moved
        mov     dx, offset DEFFCB1      ; write to the file
        call    seqwrite                ; write out the data
        cmp     al,Ø                    ; no errors
        jz      lstorel
        mov     dx, offset wrterr       ; write error
        jmp     error

lstorel: mov    dx, offset DEFFCB1      ; close the file
        call    clfile
```

```
        jz      paral
        mov     dx, offset unknown
        call    outstr
        ret
consol: mov     dx,offset conprt        ; device attached to console
        call    outstr
        ret
paral:  mov     dx,offset parprt        ; must be parallel
        call    outstr
        ret
serial: mov     dx,offset serprt
        call    outstr
        ret
main    endp
        endps

        end     start
```

*

Checkoff Search And Print (CSP)

Jim Meyers
13A Riggs Parkway
Las Vegas, Nevada 89115

CSP will search files created by HUG's CHECKOFF program, and the terminology used in CSP is consistent with CHECKOFF's. The first prompt affords you the opportunity to search the file(s) by number, comment, or amount, respectively. Enter a 4 to see a catalog of the working disk or a 5 to exit back to the operating system. To exit CSP and remain in MBASIC, enter Control-C at this prompt.

If you input a 1, 2, or 3 you'll be asked to enter the number, comment, or amount you are seeking. When you satisfy that request, you'll be asked "Starting with file number?" Enter the number of the file where the search is to begin. Similarly, the prompt "Ending with file number?" requires you to enter the file where the search is to end. The search will begin and the screen will be updated to indicate which file CSP is currently in and the listing of the "hits" it has made. After the list is printed, you may request that the list be sent to your printer. If no hits are found, the option to print is skipped automatically.

Search by comment provides the benefit of showing all the transactions that contain your input, even if it is only a single character. If you mark all tax-deductible transactions with an "*", then simply searching for an "*" will display all tax-deductible entries and present their total from as many files as you request.

Be sure to only enter numbers for the starting and ending files. Don't enter the entire file name. To search from CHECKS2 to CHECKS14, enter only a 2 for the starting number and a 14 for the ending number, for example.

CSP Notes — CP/M

In line 130 change ONLY the B to the letter of the drive that contains your check files. Leave the line alone if your data disk will be on drive B, of course.

CSP Notes — HDOS

In line 130 change the SY0 to the drive that contains your data disk. If your data is on SY1, then change it to read SY1 and leave all remaining characters on the line alone. Also, on line 130 you may change the name of your printer device by changing the LP: to whatever designation you are using. Currently it is set to print to the LP device. Whatever device you select, it is IMPERATIVE that you load this driver before you load MBASIC or you will not be able to use your printer.

CP/M Version

```
10 '    CP/M CHECKOFF 3.Ø SEARCH AND PRINT
        by JIM MEYERS version 1.3
1ØØ PRINT CHR$(27)"E"
11Ø PRINT TAB(25) "CHECKOFF SEARCH AND PRINT" :
    PRINT : PRINT
12Ø DEFINT A-Z : DEFDBL T : ON ERROR GOTO 1ØØØ :
    AA$="$###,###.##"
13Ø F$ = "B:CHECKS"
14Ø LP = Ø : N$ = "3.141" : C$ = N$ : A$ = C$
2ØØ PRINT : PRINT
21Ø PRINT "Search transactions by" : PRINT
22Ø PRINT "1. number" : PRINT"2. comment" :
    PRINT"3. amount" : PRINT
23Ø PRINT "or you may"
24Ø PRINT "  4. print disk catalog"
25Ø PRINT "  5. exit" : PRINT
3ØØ NV = Ø : INPUT "Your selection? (1,2,3,4,5) ";NV
31Ø IF (NV < 1 OR NV > 5) THEN 3ØØ
32Ø ON NV GOSUB 33Ø, 38Ø, 39Ø, 4ØØ, 11ØØ
33Ø LINE INPUT "Transaction number? ";N$
34Ø IF LEFT$(N$,1) = "-" THEN N$ = RIGHT$(N$,LEN(N$) - 1)
35Ø IF INSTR(N$,"A") = 1 OR INSTR(N$,"a") = 1
    THEN N$ = MID$(N$,2) + "A"
36Ø IF RIGHT$(N$,1) = "a"
    THEN N$ = LEFT$(N$,LEN(N$) - 1) + "A"
37Ø GOTO 5ØØ
38Ø LINE INPUT "Comment? ";C$ : GOTO 5ØØ
39Ø LINE INPUT "Amount? ";A$ : GOTO 5ØØ
4ØØ WIDTH 6Ø : FILE$ = "FILES" : FILES F$ + "??" :
    WIDTH 255 : GOTO 14Ø
41Ø RETURN
5ØØ LINE INPUT "Starting with file number? ";J$
51Ø LINE INPUT "Ending with file number? ";K$
515 IF VAL(K$) - VAL(J$) < Ø THEN
    PRINT"Start > End; try again." : GOTO 5ØØ
52Ø FIN$ = J$
53Ø T = Ø : F = Ø
54Ø N = VAL(FIN$)
55Ø PRINT : PRINT "Searching..."
56Ø FILE$ = F$ + MID$(STR$(N),2)
57Ø OPEN "I", 1, FILE$
58Ø INPUT #1, DU, DU, X, DU, DA$
59Ø PRINT FILE$;"  ";DA$
6ØØ FOR I = 1 TO X
61Ø INPUT #1, NU$, CO$, AM$
62Ø IF INSTR(CO$,C$) <> Ø THEN GOSUB 9ØØ:
    T = ABS(VAL(AM$)) + T : IF LP = 1  THEN GOSUB 8ØØ
63Ø IF ABS(VAL(NU$)) = VAL(N$) THEN GOSUB 9ØØ:
    IF LP = 1 THEN GOSUB 8ØØ
64Ø IF ABS(VAL(AM$)) = VAL(A$) THEN GOSUB 9ØØ :
    IF LP = 1 THEN GOSUB 800
650 NEXT I
660 CLOSE
670 N = N + 1
680 IF N =< VAL(K$) THEN GOTO 560
690 PRINT "Search complete."
700 IF T <> Ø THEN
    PRINT "** Total of";F;"items is ";USING AA$;T:
    IF LP=1 THEN LP=Ø:
    LPRINT "TOTAL OF";F;"ITEMS IS ";USING AA$;T :
    GOTO 14Ø
71Ø Y$ = "":
    IF F > Ø AND LP = Ø THEN
    LINE INPUT "Want the list printed? (Y/N)? "; Y$
72Ø IF F = Ø THEN PRINT "Item not found."
73Ø IF LEFT$(Y$,1) = "y" OR LEFT$(Y$,1) = "Y"
    THEN Y$ = "Y"
74Ø IF Y$ = "Y" THEN LP = 1 : GOTO 52Ø
75Ø GOTO 14Ø
8ØØ LPRINT NU$; TAB(1Ø)CO$; TAB(5Ø)USING AA$; VAL(AM$);
81Ø LPRINT TAB(7Ø)DA$" #";N : RETURN
9ØØ F = F + 1 :
    PRINT NU$; TAB(1Ø)CO$; TAB(5Ø)USING AA$; VAL(AM$) :
    RETURN
1ØØØ IF ERR = 53 THEN PRINT FILE$;" MISSING." :
     WIDTH 255 : N = N + 1 : RESUME 68Ø
1Ø1Ø ON ERROR GOTO Ø
11ØØ SYSTEM
12ØØ END
```

HDOS Version

```
1Ø  '  HDOS CHECKOFF 3.Ø SEARCH AND PRINT
       by JIM MEYERS vers 1.3
1ØØ CLEAR 1ØØØ : PRINT CHR$(27)"E"
11Ø PRINT TAB(25) "CHECKOFF SEARCH AND PRINT" :
    PRINT : PRINT
12Ø DEFINT A-Z : DEFDBL T : ON ERROR GOTO 1ØØØ :
    AA$="$###,###.##"
13Ø F$ = "SYØ:CHECKS" : P$ = "LP:"
14Ø LP = Ø : CLOSE : N$ = "3.141" : C$ = N$ : A$ = C$
2ØØ PRINT : PRINT
21Ø PRINT "Search transactions by" : PRINT
22Ø PRINT "1. number" : PRINT "2. comment" :
    PRINT "3. amount" : PRINT
23Ø PRINT "or you may"
24Ø PRINT "  4. print disk catalog"
25Ø PRINT "  5. exit":PRINT
3ØØ NV = Ø : INPUT "Your selection? (1,2,3,4,5) ";NV
31Ø IF (NV < 1 OR NV > 5) THEN 3ØØ
32Ø ON NV GOSUB 33Ø, 38Ø, 39Ø, 4ØØ, 11ØØ
33Ø LINE INPUT "Transaction number? ";N$
34Ø IF LEFT$(N$,1) = "-" THEN N$ = RIGHT$(N$,LEN(N$) - 1)
35Ø IF INSTR(N$,"A") = 1 OR INSTR(N$,"a") = 1
    THEN N$ = MID$(N$,2) + "A"
36Ø IF RIGHT$(N$,1) = "a"
    THEN N$ = LEFT$(N$,LEN(N$) - 1) + "A"
37Ø GOTO 5ØØ
38Ø LINE INPUT "Comment? ";C$ : GOTO 5ØØ
39Ø LINE INPUT "Amount? ";A$ : GOTO 5ØØ
4ØØ FILES LEFT$(F$,4) : GOTO 14Ø
41Ø RETURN
5ØØ LINE INPUT "Starting with file number? ";J$
51Ø LINE INPUT "Ending with file number? ";K$
515 IF VAL(K$) - VAL(J$) < Ø THEN
    PRINT "Start > End; try again." : GOTO 5ØØ
52Ø FIN$ = J$
53Ø T = Ø : F = Ø
54Ø N = VAL(FIN$)
55Ø PRINT : PRINT "Searching..."
56Ø FILE$ = F$ + MID$(STR$(N),2)
57Ø OPEN "I", 1, FILE$
58Ø INPUT #1, DU, DU, X, DU, DA$
59Ø PRINT FILE$;"  ";DA$
6ØØ FOR I = 1 TO X
61Ø INPUT #1, NU$, CO$, AM$
62Ø IF INSTR(CO$,C$) <> Ø THEN GOSUB 9ØØ:@
    T = ABS(VAL(AM$)) + T : IF LP = 1 THEN GOSUB 8ØØ
63Ø IF ABS(VAL(NU$)) = VAL(N$) THEN GOSUB 9ØØ:@
    IF LP = 1 THEN GOSUB 8ØØ
64Ø IF ABS(VAL(AM$)) = VAL(A$) THEN GOSUB 9ØØ :
    IF LP = 1 THEN GOSUB 8ØØ
65Ø NEXT I
66Ø CLOSE #1
67Ø N = N + 1
68Ø IF N =< VAL(K$) THEN GOTO 56Ø
69Ø PRINT "Search complete."
7ØØ IF T <> Ø THEN
    PRINT "** Total of";F;"items is "USING AA$;T : @
    IF LP = 1 THEN LP = Ø : @
    PRINT #2, "TOTAL OF ";F;"ITEMS IS ";USING AA$;T :
    GOTO 14Ø
71Ø Y$ = "" : @
    IF F > Ø AND LP = Ø THEN
    LINE INPUT "Want the list printed? (Y/N)? "; Y$
72Ø IF F = Ø THEN PRINT "Item not found."
73Ø IF LEFT$(Y$,1) = "y" OR LEFT$(Y$,1) = "Y"
    THEN Y$ = "Y"
74Ø IF Y$ = "Y" THEN LP = 1 : OPEN "O", 2, P$ : GOTO 52Ø
75Ø GOTO 14Ø
8ØØ PRINT #2,
    NU$; TAB(1Ø)CO$; TAB(5Ø)USING AA$; VAL(AM$);
81Ø PRINT #2, TAB(7Ø)DA$" #";N : RETURN
9ØØ F = F + 1 :
    PRINT NU$; TAB(1Ø)CO$; TAB(5Ø)USING AA$; VAL(AM$) :
    RETURN
1ØØØ IF ERR = 53 THEN PRINT FILE$;" MISSING." : @
     N = N + 1 : RESUME 68Ø
1Ø1Ø ON ERROR GOTO Ø
11ØØ SYSTEM
12ØØ END
```

Professional Text Processor

Copyright 1984
Robert S. De Wolf
1055 Richman Avenue
Fullerton, CA 92635

The Professional Text Processor (also known as PTP-100) is a new editor/word processor from Newline Software that runs on both the Z-110/120 and Z-150/160 computers under MS-DOS. PTP is designed so that it can do most ordinary word processing.

Before I get into PTP, I think I should give you my experience as a reviewer. I have about 20 years experience in the computer field. I have used editors on many systems and early in my career I wrote a text editor myself. These days, I work as a system manager-programmer on VAX superminicomputer systems made by Digital Equipment Corp. Some of the editors I am familiar with are SOS, EDT, and TECO from DEC, INED (similar to EMACS) from Interactive Systems, and PeachText 5000 from MSA. I have used the NROFF text formatter from UNIX and Digital's RUNOFF text formatter.

The main reason I was looking for a word processor was that the PeachText 5000 system was just too slow and clumsy for a lot of the work that I was doing with it. One thing that particularly annoys me about PeachText is that when you use the up-arrow cursor control key at the top of the screen, PeachText does a reverse scroll by clearing the screen and refreshing it! A simple escape sequence to the Z-100 screen will tell it to shift all the text up one line to make room for a line at the bottom. This is the preferred technique, and is a lot easier on the eyes. Another annoying feature about PeachText is the slowness of it. There seems to be a lot of unnecessary disk activity.

The most striking feature of PTP is its quickness. Once you load the program, PTP does not need to use the disk except to read in a new file, write text to a file, or to read the HELP file. When you page forward, PTP does it smoothly and with real snap. It is not necessary to write a file to disk in order to print it. PTP prints directly from the buffer. Sometimes you want to write a short note to someone and have no need to save it on the disk. PTP does this as efficiently as you can do it with a typewriter. Of course, if you do need to save it in a file you can, it's just nice to be able to avoid it when you don't want it.

When you are using PeachText, if you hold down the up-arrow key until it begins to auto-repeat, the type-ahead buffer will fill with commands faster than PeachText can process them. The screen goes blank. You take your finger off the key and wait helplessly while PeachText works off the queue of up-arrow commands. Then PeachText stops at some point well beyond the point you were looking for! This never happens with PTP because PTP scrolls correctly and disables type-ahead so that the cursor displayed is where it will stop when you take your finger off the key.

PeachText has the audaciousness to reset the terminal as you enter the editor. Therefore, if you prefer a block cursor or some other terminal initialization, PeachText insists that you do things its way. PTP politely leaves things be. You can have a non-blinking block cursor if you wish. PTP does not work in inverted video, however.

PTP uses on-screen formatting. PTP does not have formatting commands that can be embedded in a document and interpreted by a formatter before printing. You can center lines and fill or justify paragraphs on the screen using a margin/tab ruler displayed at the top of the screen. This feature saves a lot of time, since you don't need to print or format the document in order to see what it will look like.

PTP does not support micro-justification or subscripting. The only specialized printer functions that are supported are bolding and underscore. PTP displays bolding and underscore on the screen as they would appear on paper. When you are installing PTP, you enter the escape sequences that control bolding and underscore for your printer.

Both insert and overstrike mode are available in PTP, depending on your taste. I use the insert mode, primarily because it is most like the editor that I use at work. It is confusing to switch back and forth.

One thing a text formatter does well that PTP does not, is to prepare a large document that needs to be maintained over a period of time. For example, software design documentation is often maintained and kept consistent with a program. A text formatter can make changes throughout a document by changing a few commands at the beginning. Some text formatters can automatically maintain the section numbers, table of contents, and the index. If you have a formatter that uses standard ASCII files for input, you could edit them with PTP, provided that you do not use bold, underscore or tab fill mode.

Editing Functions

All PTP editing functions are invoked by using the dedicated keys (DEL LINE, INS LINE, etc.) on the Z-100 keyboard plus the numeric keypad. The function keys are reserved for user-defined macros. Control characters are not used, but there is an 'expert mode' that uses both the control and shift key with a letter to gain quicker access to some of the PTP functions. PTP does not provide for re-definition of the keypad functions. It would not be desirable to do so, because Newline has done such a good job of laying out the keypad already. All functions that need an indication of direction use the cursor control keys for that indication.

For example:

Cursor up one line	Up-arrow
Scroll up one screen	Shift Up-arrow
Search up for string	Keypad-4 <string> Up-arrow

There are two ways to move up (or down) one screen in a file. One way to move up is by pressing Shift Up-arrow. This scrolls up one screen, adding a line at the top for each line that is deleted from the bottom. A second way to do this is with the Keypad-1 key. This key is referred to as prev screen. When you use prev screen, the screen is cleared and refreshed quickly with the screen above. I prefer to use prev screen rather than Shift Up-arrow.

PTP can find a string, find and replace a string, or find and delete a string. All of these commands are available either with a case-sensitive search, or a case-insensitive search. Case-insensitive search is the default. To get case-sensitivity, you just hold down the control key while entering the command. This is a logical design that makes it easier to remember how to use all the capabilities that are there.

I said earlier that no control codes are used for editing functions. If you enter one of the control codes in the ASCII character set, it will be entered into the file and displayed in reverse video. This makes it very convenient to put escape sequences for printer control directly into the document file. Only two characters cannot be represented this way, Control-Z and Control-J. This feature will enable you to use specialized printer control features such as subscripting, but is not as convenient as it would be if they were displayed on the screen.

The cut and paste functions are similar to the EDT editor on the VAX. You position cursor on one side of the text you wish to cut and press the select key (Keypad"."). Then you move the cursor to the other side of the text you wish to cut. As you move the cursor, the text between the first position and the cursor will be converted to inverted video. When you get as far as you want to go you press SHIFT DEL LINE to perform the cut. All of the text displayed in inverted video disappears. You can then paste it somewhere else by pressing select followed by DEL LINE. You can also cut without removing the text for the purpose of duplicating some text in a different location. This function can be done with EDT, but only with a keypad macro.

Files

The file produced by PTP is a standard ASCII text file, if you do not use bolding or underscore. Many other word processors use unusual file formatting or a peculiar data format. Non-standard formatting can give you problems, if you plan to transfer files to a mini or mainframe computer.

The files produced by PTP are compatible with PeachText, but not vice-versa. The PeachText spelling checker works just fine with files produced by PTP. The PeachText editor puts a carriage return character only at the end of the paragraph. The standard ASCII text files produced by PTP have carriage returns at the end of every record. Since the longest record PTP can handle is 255 characters, PTP may have difficulty with some files created by PeachText. A simple program could be written to convert PeachText files so that they could be read by PTP. Once you start using PTP, it will likely be difficult to go back to PeachText, however.

You can edit any size file with PTP. If the file is larger than memory, you must use the write command to write a portion of the file to disk, then read in more of the file using the read command. This is similar to PeachText. Although I have never needed this feature, I think it is clumsy. The editors on the full-sized computers I have used all handle this function automatically, hiding the problem from the user.

It is possible to read in a second file while you are using PTP. It is not possible to open a window into a second file, cut some text out of it and place it in the primary file. EDT allows this as does INED, WatchWord and many other editors. PeachText 5000 can do something almost as good, but not quite. Of course, you can get the job done without this feature, but it is just a little bit clumsier. I can't say how much of an inconvenience this will be in actual practice. I think it depends on the kind of editing you will be doing.

Advanced Features

PTP has some features that I wish were available on the VAX. It has a display tabs mode where the tab characters contained in a file are displayed on the screen. This is helpful if you will be processing the file with a program that is sensitive to files containing tabs. PTP has a mode which will tab-fill the file, putting in tab characters wherever possible to save space. If you are working in an environment that can tolerate this, it saves disk space by reducing the space occupied in a file by strings of blanks.

PTP has a center mode, which will keep the line you are working on in the center of the screen, displaying ten lines of text on either side of it. This is handy, because you can see the contents of what you are writing.

Macros are entered by holding down the shift key and pressing one of the function keys. Then you enter the sequence of keys that comprise the macro, followed by any function key. Every time you press the function key, all operations will be repeated. This is similar in operation to the EDT editor that I use at work. Macro operations are significantly slower than PTP built-in functions. Even so, they are not slow enough to be a problem.

The BREAK key is used as an OOPS key when editing. It reverses the effect of the previous operation if you make a mistake. This is a very useful feature. Most of the time it will help you recover from errors. Sometimes it does the unexpected, but that has never been disastrous. The BREAK key is also used to exit from any of the menus used by PTP.

The Z-100 HELP key is used to get help. It reads an ordinary text file and displays the help information on the screen. The user can edit the HELP file and put any information in it that is convenient. After you press the HELP key, the page of the HELP file is displayed. If you want to see additional pages from the HELP file, you press the RETURN key. Pressing the BREAK key will return to the file you are editing. EDT has a more sophisticated help file. When you press the HELP key in EDT, the keypad is displayed. If you press the SPACE bar, you will return to the file you were editing. If you enter an EDT command at the keypad, EDT will display a paragraph of specific help text for that command (key sequence). Thus, the keypad display serves as a kind of menu for lower level help screens. This is all very fine, but it prevents you from modifying the help file, and it is impossible to keep the help file correct when you use macros to redefine the keypad functions.

Summary

In summary, PTP is an excellent tool for memos, magazine articles, and the like. For specifications and large documents, I still think I would prefer the editor/text formatter approach. Text formatters are clumsy for the things PTP is good at though. Often, one wishes to write a memo without saving it on a disk. Why save a copy when you don't need it? You can do this with PTP because PTP can print directly from memory. For little things PTP is a lot faster.

Professional Text Processor is available from Newline Software, P.O. Box 289, Tiverton RI 02878, (401) 624-3322.

recover from a

FATAL ERROR

in just sixty seconds with Rescue & Repair.

At last, there is a fast, simple way to recover data lost because of "Read Only" errors and operator mistakes. In fact, Rescue & Repair will allow you to retrieve your "lost" data after *any* error (except power failure or major technical malfunction), even if the machine has been warm booted or *reset.* Reviewers have been unanimous in their praise. **Interface Age** wrote *"The name says it all. . . The 66-page manual is excellent."* **H-SCOOP** *calls Rescue & Repair "a powerful and very useful program. . . It is extremely easy to use."* **Sextant** calls it *"a real life saver,"* and states that *"Rescue & Repair will recover files lost due to practically anything."* Rescue & Repair is available for MS, PC and Z-DOS* machines for $75 and most CP/M** machines for $55.

Ask your Heath/Zenith dealer or contact:

21st CENTURY DATA, INC.

P.O. Box 1139
Solana Beach, CA
92075
(619) 755-6218
California residents add 6% sales tax.

*Z-DOS is a trademark of Zenith Data Systems
**CP/M is a trademark of Digital Research Corp

WORDKEY™ TAMES THE POWERFUL WORDSTAR

WordKey makes the popular WordStar^{TM} program the easy-to-use professional word processor it should be. More than 90 simple keypad and function-key commands help you use and remember ALL of WordStar's many capabilities, without EVER using WordStar's awkward 'CTRL' key. Your secretary will love it, and so will you and your family.

> Full onscreen help is always instantly available. And you'll like its many other features. A letter to H-Scoop from a satisfied user says, "I found [WordKey] made WordStar much easier to use and ...I learned some capabilities that I did not know WS had...I am extremely pleased with it."

Use WordKey with either WordStar version 3.21 or 3.30, Z-DOS 1.25 or MS-DOS 2.13, on your H/Z-100 computer. To order, just mail a check to DelSoft for $49.95. (Calif. residents please add 6% tax.)

PSC-PROWRITER GRAPHICS SCREEN DUMP UTILITY

PSC-PRO prints an exact copy of your H/Z-100 monochrome (B&W or green) screen on the C.Itoh Prowriter printer from within ANY program simply by pressing Shift-F12. $19.95 (Calif. residents please add 6% tax.)

DelSoft
Gary Deley, 564 Calle Anzuelo, Santa Barbara, CA 93111
(805) 967- 9566 eves and weekends

RUN CP/M* 80 PROGRAMS ON YOUR Z150, IBM* PC (/XT) OR PC CLONE AT MORE THAN TWICE THE SPEED OF A Z89 USING THE UN/EMULATOR

Using the UN/EMULATOR you can run CP/M 80 programs without modifications on your Z150, IBM PC(/XT) or any 8088/86 MS*-DOS computer. UN/EMULATOR runs CP/M 80 programs at more than twice the speed of a stock Z89. Even MS-DOS programs will run 10% faster. UN/EMULATOR is not an emulator! It allows your 8088/86 computer to run 8080 code as a native language within a CP/M 80 environment. It outperforms CP/M 80 emulators 10 to 1! The list price for UN/EMULATOR starts at $149.95. As a HUGCON special you can order UN/EMULATOR packages for as little as $99.95 each with coupon (NO COPIES PLEASE!). 8 MHz versions of UN/EMULATOR are available for an additional $20.00.

Dealer Inquiries Welcome / See Us At Booth 11

*CP/M, IBM and MS-DOS are registered trademarks of Digital Research Corp., International Business Machines and Microsoft respectively

	HUGCON SPECIAL	
UN/EMULATOR 8088 ~~$149.95~~ . . .	$ 99.95	____
8MHz UPGRADE (8088).	$ 20.00	____
UN/EMULATOR 8086 ~~$159.95~~ . . .	$109.95	____
8MHz UPGRADE (8086).	$ 20.00	____
Shipping & Handling - add	$ 5.00	
FL Residents Please add 5% Sales Tax		

COUPON GOOD THROUGH 9/30/85

Send Check or Money Order To:

WAWA
WEITZMAN AND WOOD ASSOCIATES
580 N.W. 99th Way, Pembroke Pines, FL 33024
Phone: (305) 432-8025

Name: ____________________

Address: ____________________

City/State/Zip: ____________________

Phone: (   ) ____________________

System: ______________ Qty: ________

Vectored from Page 41

DATA BASE MANAGEMENT SYSTEMS

HDOS

Part Number	Description of Product	Selling Price	Vol. Issue
885-1107-[37]	HDOS Data Base System H8/H89	30.00	23
885-1108-[37]	HDOS MBASIC Data Base Sys.	30.00	23
885-1109-[37]	HDOS Retriever ASM (3 Disks)	40.00	23
885-1110	HDOS Autofile (2 Disks)	30.00	23
885-1115-[37]	HDOS Navigational Program	20.00	25
885-8008	Farm Accounting System	45.00	30

CP/M

Part Number	Description of Product	Selling Price	Vol. Issue
885-1219-[37]	CP/M Navigational Program	20.00	31

AMATEUR RADIO

HDOS

Part Number	Description of Product	Selling Price	Vol. Issue
885-8016	Morse Code Transceiver Ver 2.0	20.00	42

CP/M

Part Number	Description of Product	Selling Price	Vol. Issue
885-1214-[37]	CP/M MBASIC Log Book (64k)	30.00	23
885-1234-[37]	CP/M Ham Help	20.00	49
885-1238-[37]	CP/M ASCIRITY	20.00	57
885-8020-[37]	CP/M RF Comp. Aided Design	30.00	44
885-8031-[37]	CP/M Morse Code Transceiver	20.00	57

COMMUNICATION

HDOS

Part Number	Description of Product	Selling Price	Vol. Issue
885-1122-[37]	HDOS MicroNET Connection	16.00	37

CP/M

Part Number	Description of Product	Selling Price	Vol. Issue
885-1207-[37]	CP/M TERM & HTOC	20.00	26
885-1224-[37]	CP/M MicroNET Connection	16.00	37
885-3003-[37]	CP/M ZTERM (Z100 Modem Pkg)	20.00	34
885-5004-37	CP/M 86 TERM86 and DSKED	20.00	56
885-5005-37	CP/M 86 16 Bit MicroNET Conn.	16.00	61
885-5006-37	CP/M 86 HUGPBBS	40.00	62
885-5007-37	CP/M 86 HUGPBBS Source List.	60.00	62
885-8005	MAPLE (Modem Appl. Effector)	35.00	29
885-8012-[37]	CP/M MAPLE (Modem Program)	35.00	34
885-8023-37	CP/M 85 MAPLE	35.00	45

MSDOS H/Z100 - H/Z150 PC

Part Number	Description of Product	Selling Price	Vol. Issue
885-3019-37	ZDOS 16 Bit MicroNET Connect.	16.00	61
885-3027-37	MSDOS HUG PBBS	40.00	66
885-3028-37	MSDOS HUG PBBS Source Listing	60.00	66

MISCELLANEOUS

Part Number	Description of Product	Selling Price	Vol. Issue
885-0004	HUG Binder	5.75	
885-1221-[37]	Watzman ROM Source Code/Doc	30.00	33
885-4001	REMark Vol. I Issues 1-13	20.00	
885-4002	REMark Vol. II Issues 14-23	20.00	
885-4003	REMark Vol. III Issues 24-35	20.00	
885-4004	REMark Vol. IV Issues 36-47	20.00	
885-4005	REMark Vol. V Issues 48-59	25.00	
885-4500	HUG Software Catalog	9.75	
885-4600	Watzman/HUG ROM	45.00	41
885-4700	HUG Bulletin Board Handbook	5.00	50
885-3015-37	ZDOS SKYVIEWS	20.00	55

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

Vectored from Page 10

trip you up, but the one I've found to be the most common is the failure to understand the difference between DCE and DTE connections. There's a good chance this is at least part of Mr. Gray's problem.

Although almost all RS-232C connections physically look alike (they use male or female DB-25 connectors), they actually come in two varieties: Data Communications Equipment (DCE) connections and Data Terminal Equipment (DTE) connections. The difference between the two is that they use exactly opposite conventions for transmitting and receiving data and/or line control information. This is necessary so that while one piece of equipment is "talking" on one wire the other is "listening" on that same wire. If there didn't exist two conventions, the two pieces of equipment would be trying to "talk" on the same wire. The important rule to remember is that two of the same type connections will never work together, that is, DTE equipment always expects to be connected to DCE equipment and vice versa.

Modems are almost always configured as DCE, and terminals (such as the H-19) are almost always configured as DTE, but there appears to be no consensus as to how computer serial ports should be configured. Which configuration supplied depends on whether the serial interface is intended to be connected to a modem (in which case it is probably DTE) or to a printer (in which case it is probably DCE). One nice feature of the old H-8-4 serial interface for the H-8 is there are two sets of connectors so you can decide for each connection whether to use DCE or DTE. On the Z-100 there is actually one of each (J1 is DCE, J2 is DTE).

One way you can sometimes tell if you're looking at a DTE or DCE connection is to see whether the connector is male or female. The convention (which is by no means universally followed) is that DCE uses a female DB-25 and DTE uses a male DB-25 connector.

The first thing to do when you're hooking up a modem to your computer is to determine if your serial interface is DTE or DCE. This may be indicated on the circuit board or connector or in the owner's manual. Some boards (like the H-8-4) can be configured either way via jumpers or switch settings. To connect to a modem the port should be set up as DTE. To connect to a printer the port should be set up as DCE.

If you have incompatible interfaces (two DCE's or two DTE's) you can also solve the problem using a "null modem". A null modem is nothing more than a cable that switches certain pairs of wires to make one piece of equipment look like the other. When hooked up to DCE the null modem makes the connection look like DTE. When hooked up to DTE the null modem makes the connection look like DCE.

You can buy a prefabricated null modem at a ridiculously high price (e.g. $30), or you can make one using two DB-25 connectors and an appropriate cable. Here's how my null modem cable is wired:

```
One Connector          The Other Connector
          1 <--------------------> 1
          2 <--------------------> 3
          3 <--------------------> 2
          7 <--------------------> 7
          8 <--------------------> 2Ø
         2Ø <--------------------> 8
```

In addition, pins 4 and 5 are connected together on each end as are pins 6 and 8.

Note that the wires hooked to pins 2 and 3 are swapped, as are the wires connected to pins 8 and 20. If you're connecting to a modem, you often only need to run wires 1, 2, and 3 since 8 and 20 are used for carrier detect information which many modem packages ignore. Some printers use lines 4 and 5 for "handshaking". In these cases wires for 4 and 5 must be run through and swapped (like 2 and 3 were).

Sincerely,

Glenn F. Roberts, Ph.D.
12048 Greywing Square, #C-3
Reston, VA 22091

Garbage Filter Corrections

Dear Walt:

Please publish the following corrections to my article "Bulletproof Garbage Filters" (REMark, Volume 6, Issue 3):

Page 55, right-hand column, first paragraph after the '******'. The third sentence should read: "The choices are: C, D, E, F, H, M AND T (for Cleric, Dwarf, Elf, Fighter, Halfling, Magic-user and Thief, respectively)."

Page 57, left-hand column, last two lines. The REM in line 780 of the code sample should state that the segment creates a 0-letter to 7-letter password. The last four characters of line 790 should be omitted.

Page 58, left-hand column. The first occurrence of line 50 should be identical to the second occurrence (i.e. the 'n' should have been 'N').

Also, please note that the second-to-the-last paragraph in the left-hand column of page 55 starts with the sentence: "This will, of course, cause the second test. . .". This was written with the point of view that the garbage filter mentioned in the previous paragraph was the new 'first' test.

It seems that Mr. Murphy, of 'Murphy's Law' fame, was standing nearby (and snickering) when my final draft was being reviewed, prior to submission, since not one of the three reviewers spotted these problems. After I had submitted the article, I saw them, but because of hardware failure I was unable to get the corrections to REMark before the issue had been sent to the printer. My apologies to everyone who got distracted and/or confused by these errors, and my deepest apologies to any Elves who might have felt slighted by my omission on Page 55.

I would also like, at this time, to publicly thank HUG/REMark for providing me with a vehicle which allowed me to repay a "bread upon the waters" type debt. When I first learned to program, there were numerous folks that patiently assisted me, and without such help, I no doubt would have given up in frustration long ago.

Sincerely yours,

Kurt A. Schultz
115-1 Roxanne Court
Walnut Creek, CA 94596

Printer To H/Z 89 Problem

Dear HUG:

A rather obscure problem appeared this past week which I would like to share with H/Z 89A users with a MX-80 Epson

Printer using the Buffered "Serial" Interface Board and with GRAFTRAX Plus. The problem did not show up until I tried to use a CALLIGRAPHY package. In addition, the problem also showed up when I tried to use a graphics program I had downloaded from a local BB. In both instances, my MX-80 would only print 7 dots at any one time. For normal printing this is okay. In fact in the MX-80 User's Manual, most of the guide lessons printed okay as they were geared to 7 dots. The manual did say that some computers would only send 7-bit words. Graphic and Calligraphy Programs generally require an 8-bit word (8 dots). After much frustration and close review of the manuals, I discovered the cause. The H/Z Set-up Instructions that came with the printer and serial board conditioned switch SW2 on the Serial Board in the printer for a 7-bit word. Where the confusion arises is that the User's Manual for the printer does not mention anything about the Serial Board. The printer main board also has switches labeled SW1 and SW2 (same as the Serial Board). When I checked the User's Manual there was no indication on how to set the word length. Luckily I had the small User's Manual for the Serial Board — RS-232C/Current Loop Interface Type 2. There it was SW2-1 (Serial Board) that needed to be in the OFF position for an 8-bit word. It would be interesting to know why the Set-up Instructions from H/Z used the 7-bit word!

Hopefully this bit of information may help someone who is experiencing similar problems or happens to purchase a second hand printer without the appropriate manuals.

Sincerely,

John P. Gallagher
625 Hooiki Street
Pearl City, HI 96782

A Devout BASIC User

Dear HUG:

After reading the March issue of REMark, I had to write. I am a devout BASIC user and read REMark each month for anything that will make my efforts easier. I hope that in the future I will have the time to submit some of my programs and thoughts on using BASIC, including programs to merge libraries and base programs, provide program listings in a readable form, and CP/M to MS-DOS upgrade programs.

What actually prompted me to write was the program 'DOSMENU.BAS' by Stephen Chiavetta, III. While I don't have a Winchester disk and can't vouche for the usefulness of the program, I can say that it can be cut down significantly and accomplish the same task. Below is my listing of the same program, 'DOSMENU.BAS'.

My only comment is that if I were to use the program, I would delete line 350 and replace line 80 with 'OPEN "O", #1, XTEMP:CLOSE:KILL XTEMP'. That way, when I compile the program, the /E and /X switches would not have to be used and the compiled program would be smaller.

Sincerely,

John H. Mosow
2238 Southwood Place
Lincoln, NE 68512

```
10 ' DOSMENU.BAS
20 '
30 DEFINT A-L:DEFSTR M-Z
40 CLS
50 'DELETES THE PREVIOUSLY CHOSEN DATA FILE THE LAST TIME
    THE PROGRAM WAS RUN
60 FOR I = 1 TO 12
61 '
62 ' LINE 70 IS USED TO SET UP THE FILE NAME TO BE REMOVED.
     THE STR$ FUNCTION PUTS A SPACE BEFORE THE NUMBER AND
     MUST BE STRIPPED OFF.
63 '
70 XTEMP = RIGHT$(STR$(I),LEN(STR$(I))-1) + ".DAT"
80 ON ERROR GOTO 350:KILL XTEMP
90 NEXT I
100 'PRINT THE DIRECTORY AND OPTION NUMBER
110 LOCATE 2,27:PRINT "MENU OF PROGRAMS"
120 LINE (0,0)-(639,224),1,B
130 COLOR 7,0
140 LOCATE 4,15:PRINT "DIRECTORY OF E:"
150 LOCATE 6,15:PRINT "1) MS-DOS 2.1 FILES"
160 LOCATE 7,15:PRINT "2) UTILITIES"
170 LOCATE 8,15:PRINT "3) LOTUS"
180 LOCATE 9,15:PRINT "4) WATCHWORD"
190 LOCATE 10,15:PRINT "5) WATCHWORD - STEVE'S DOCUMENTS"
200 LOCATE 11,15:PRINT "6) WATCHWORD - JANET'S DOCUMENTS"
210 LOCATE 12,15:PRINT "7) WATCHWORD - JOHN'S DOCUMENTS"
220 LOCATE 13,15:PRINT "8) WATCHWORD - STEPHEN'S DOCUMENTS"
230 LOCATE 15,15:PRINT "DIRECTORY OF F:"
240 LOCATE 17,15:PRINT "9) GAMES"
250 LOCATE 18,14:PRINT "10) BASIC"
260 LOCATE 19,14:PRINT "11) BASIC - STEVE"
270 LOCATE 20,14:PRINT "12) BASIC - JOHN"
280 COLOR 7,1:LOCATE 22,19:LINE INPUT
    "PLEASE ENTER THE NUMBER OF YOUR CHOICE - ";TANS
290 IF VAL(TANS) < 0 OR VAL(TANS) > 12 THEN BEEP:GOTO 280
300 COLOR 7,0
310 'CREATES A DATA FILE THAT CORRESPONDS TO THE SELECTED
     DIRECTORY
320 OPEN "O", #1, TANS + ".DAT"
330 WRITE#1,TANS
340 CLOSE:CLS:SYSTEM
350 RESUME NEXT
```

ZD Help File

Dear Walt:

Like other Heath/Zenith users, I enjoyed Jeff Kalis' contribution in the April '84 (page 14) REMark. The enhancement provided by Timothy Gonsalves in the March '85 (page 9) REMark is another useful feature. One feature which would be helpful is just that, a help file. I did not use ZD often enough to remember what the options were or how they were invoked. Aha!! Add the following and a HELP file will be printed on the CRT:

Step One:

Make the corrections and changes as shown in the March '85 issue (page 9).

Step Two:

After the comment — Based on etc. . . add

```
;    Error correction and enhancement from REMark 1985 page 9
;        by Timothy A. Gonsalves - Version 1.2
;
;        Help files add at OPSMSG and label CNKT0:
;        by Larry T. Wier - Version 1.3
;
```

Step Three:

At the label VERSN — Change to. . .

```
VERSN   DB      'ZD - Version 1.3  March 1985'
```

At the label YEAR — Change to. . .

```
YEAR    DB      '     <<<',13,10,0
```

After the label STKCNT — Add. . .

```
;
OPSMSG  DB    13,10,'              OPTIONS LIST:',13,10,10
        DB    '     C-Clear screen before printing directory.'
        DB    13,10,'   F - FF  Form Feed for printer.',13,10
        DB    '     H-Prints this list.',13,10
        DB    '     P-Sends directory to printer.',13,10
        DB    '     T-Allows user to enter title.',13,10
        DB    '     V-Displays version number.',13,10,10
        DB    'Example: ZD/C/V clr scrn and print version.'
        DB    13,10,'        ZD/T/P enter title then print.'
        DB    13,10,'$'
```

Step Four:
After the label CKNXT:

```
CKNXT:  INC     BX
        . . .
        . . .
        . . .
        . . .
        OR      AL,' '          ;Force Lower Case

        Add the following. . .
;
        CMP     AL,'h'          ;If chara is 'h' print HELP
        JNZ     CKNTØ           ;file and return to system
        MOV     DX,OFFSET OPSMSG
        MOV     AH,9            ;print it
        INT     21H             ;
        INT     2ØH             ;
;
CKNTØ:  CMP     AL,'c'          ;
```

Note: The CKNT0: label is added to the code where the CMP AL,'c' line already exists.

Operation:

Anytime ZD is invoked as ZD/H or ZD/(any character)/H, the HELP file will be printed on the CRT. If spaces or incorrect entry is used, a >>>NO FILES FOUND<<< message will be printed. In addition, the label change at YEAR was done to add a space between the version entry and the first line of the directory. It looks somewhat better when printed on the printer this way.

I hope this addition will help some users who may not remember ZD well enough to recall which option does what.

Larry T. Wier
1068 149 Place S.E.
Bellevue, WA 98007

Changing your address? Be sure and let us know since the software catalog and REMark are mailed bulk rate and it is not forwarded or returned.

CUT ALONG THIS LINE

HUG MEMBERSHIP RENEWAL FORM

HUG ID Number: ______________________

Check your ID card for your expiration date.

IS THE INFORMATION ON THE REVERSE SIDE CORRECT? IF NOT, FILL IN BELOW.

Name ______________________

Address ______________________

City-State ______________________

Zip ______________________

REMEMBER - ENCLOSE CHECK OR MONEY ORDER

CHECK THE APPROPRIATE BOX AND RETURN TO HUG

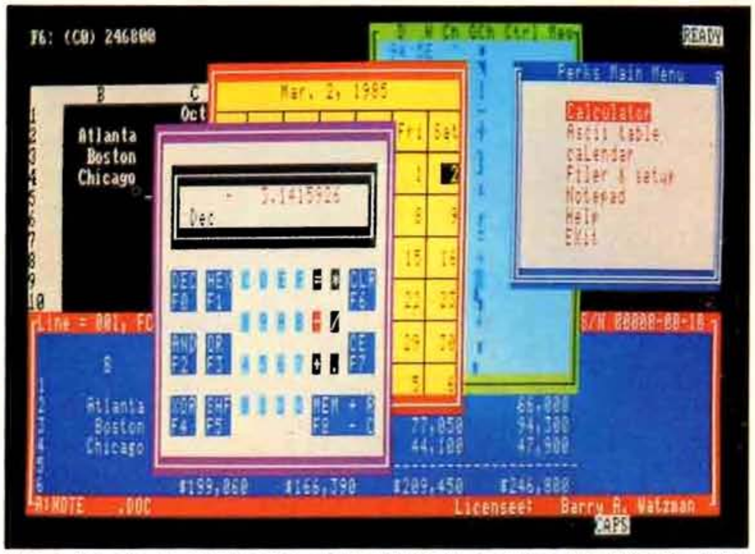
	NEW MEMBERSHIP RATES	**RENEWAL RATES**
U.S. DOMESTIC	$20 □	$17 □
FPO/APO & ALL OTHERS*	$35 □	$30 □ U.S. FUNDS

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

If you have a Z-100, treat yourself to some

PERKS™

Perks is the background desktop utility program for Z-100's featuring full-color windows, a calculator, a notepad, an ASCII table, a Filer, a screen saver, alarm timers, context sensitive help screens and a calendar with an appointment schedule. Perks works with all Z-100's having 192K or more of memory and using the Z-DOS/MS-DOS 1.X or 2.X operating systems.



Shown above is an actual screen photo of Perks in operation. The Notepad window contains data "imported" from the Lotus 1-2-3 worksheet being prepared when Perks was activated.

Here are some actual unsolicited comments from customers:

"My heartiest congratulations and thanks for a fine product. I expect that Perks will be the best software dollar I have yet spent, and I have spent lots. Anyone with a Z-100 should run to buy Perks"

"I know I said this over the phone, however, once again I would like to thank you for a GREAT product."

"You're the best friend a Z-100 ever had! Mine loves Perks, and I want to thank you for the prompt filling of my order. Perks is everything and more than I thought it was going to be. The Notepad function alone is worth the price. And to think, I'd thought about buying a Programmer's Calculator from Hewlett Packard! Yours does everything I'd have used HP's for. Your Program is simply beautiful and elegant."

Perks is available at Heath/Zenith Computers & Electronics Centers, many independent Zenith Data Systems dealers or directly from:

BARRY A. WATZMAN Microcomputer Systems & Consulting

560 Sunset Road • Benton Harbor, Michigan 49022-7142 • (616) 925-3136

Perks is a trademark of Barry Watzman.

Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

P/N 885-2067