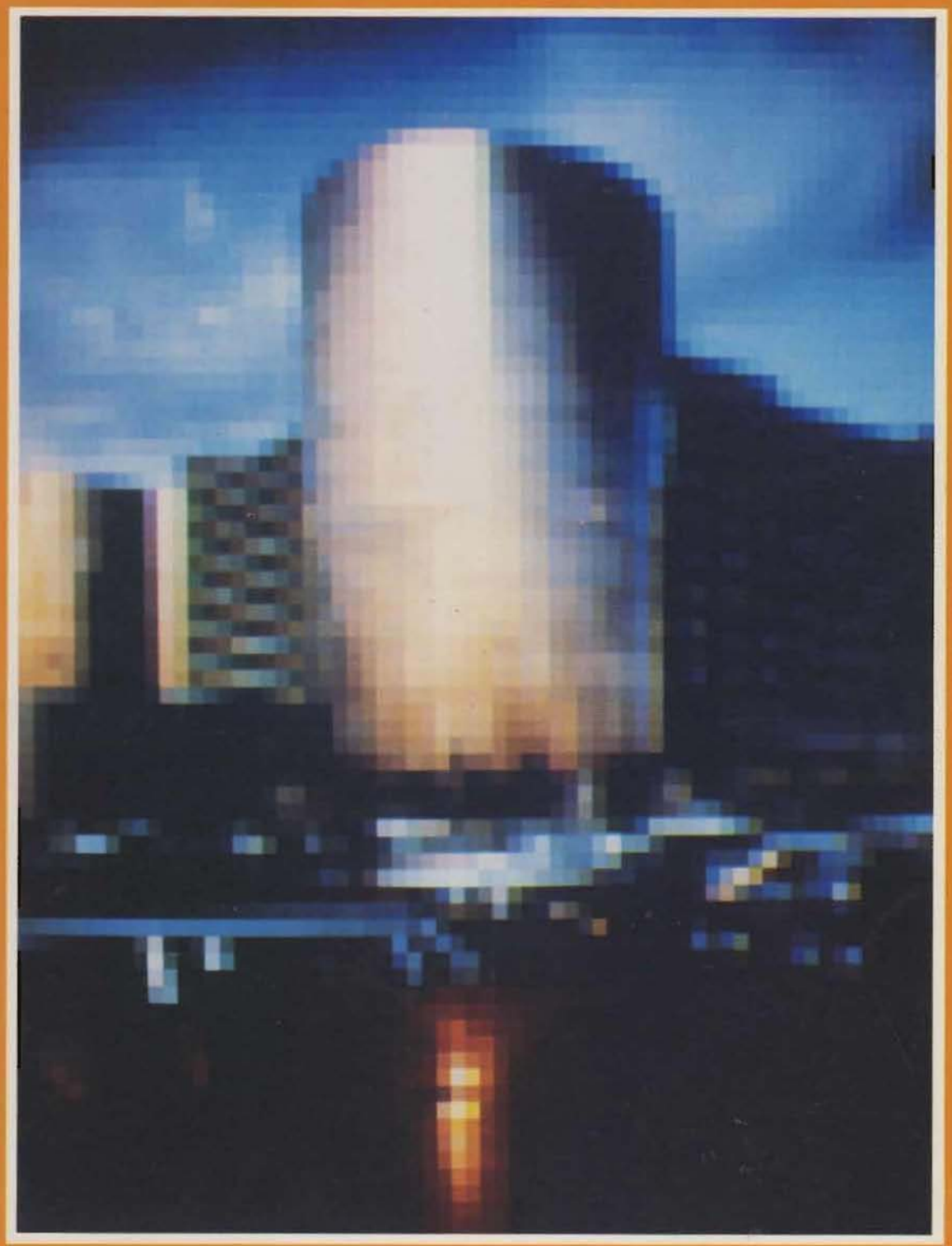
REMark®

Volume 6, Issue 7 • July 1985
P/N 885-2066 Issue 66

$2.50

Official magazine for users of HEATH ZENITH computer equipment.



Transfer "Foreign" Files in Seconds

NOW AVAILABLE FOR Z-100 AND Z-150

No modems and RS-232 links required!

ONLY $39.95!

READ, WRITE and FORMAT more than 70 different CP/M and MS-DOS disks on your computer with MEDIA MASTER™

Tired of hunting for software in your computer's disk format? Do you need to exchange your Lotus 1-2-3, dBase II, Multiplan, and Wordstar data files among different CP/M and MS-DOS computers?

Then MEDIA MASTER is for you!

MEDIA MASTER supports the following disk formats:

Actrix (SSDD)
Actrix (DSDD)
Avatar TC10 (DSDD)
Casio FP 1000 (DSDD)
Chameleon CP/M-80
Columbia MPC CP/M-80
Cromemco CDOS (SSDD)
Cromemco w/Int'l Term (DSDD)
Cromemco w/Int'l Term (SSDD)
DEC VT180 (SSDD)
Davidge (DSDD)
Digilog (DSDD)
Epson Multifont (DSDD)
Epson QX-10 (DSDD)
Fujitsu Micro 16s (DSDD)
Groupil III CP/M (DSDD)
H/Z Z-100 CP/M (DSDD) - later
H/Z Z-100 CP/M (SSDD)
H/Z Z-100 Z-DOS 1.xx (SSDD)
H/Z Z-100 Z-DOS 1.xx (DSDD)
H/Z Z-100 Z-DOS 2.xx (SSDD)
H/Z Z-100 Z-DOS 2.xx (DSDD)
H/Z Z-90 40 trk, 1 k blk (SSDD)
H/Z Z-90 40 trk, 2 k blk (SSDD)
Heath w/Magnolia CP/M (SSDD)
IBM PC CP/M-86 (DSDD)
IBM PC CP/M-86 (SSDD)
IBM PC-DOS 1.xx (DSDD)
IBM PC-DOS 1.xx (SSDD)
IBM PC-DOS 2.xx (DSDD)
IBM PC-DOS 2.xx (SSDD)
IDEA Bitelex (SSDD)
ISM CP/M (DSDD)
Insight Dev. IQ-120 (SSDD)
Kaypro II/2 (SSDD)
*Kaypro 4, 10 (DSDD)
LNW-80 (SSDD)
Lobo Max-80 (SSDD)
Lobo Max-80 512 (SSDD)
Micral 9050 CP/M-80 (DSDD)
Morrow MD 11 (DSDD)
Morrow MD 2 (SSDD)
Morrow MD 3 (DSDD)
NCR Decision Mate 5 (DSDD)
NEC PC-8001A (SSDD)
NEC PC-8001A (DSDD)
Olympia ETX II (SSDD)
Olympia EX100 (DSDD)
Osborne (SSDD)
Osborne 4 (DSDD)
Osborne Osmosis (SSDD)
Otrona (DSDD)
PMC Micromate (DSDD)
Reynolds & Reynolds (SSDD)
Sanyo (DSDD)
Superbrain (DSDD)
Superbrain Jr. (SSDD)
Systel II (SSDD)
Systel III (DSDD)
TI Professional CP/M-86 (DSDD)
TRS-80 III FEC CP/M (SSDD)
TRS-80 III FEC T805 (SSDD)
TRS-80 III Hurr. Labs (SSDD)
TRS-80 III Mem. Merch (SSDD)
TRS-80 IV CP/M+ (SSDD)
TRS-80 IV Mont. Micro (SSDD)
Teletek 40 trk (SSDD)
Toshiba T100 (DSDD)
TurboDos (DSDD)
Wang Maws CP/M (DSDD)
Xerox 820 II (SSDD)
*Zorba 40 trk (DSDD)

**Requires access to "foreign" computer*

***MEDIA MASTER** is also available for DEC Rainbow, Osborne, and Kaypro computers.*

NEW! — FILE MATE™

A Dozen Utilities in One! **ONLY $29.95**

Finally — a powerful, easy to use file utility that performs a host of functions. **FILE MATE's** numerous text sifting and paragraph reformatting modes help you exchange "foreign" text files. Other file inspection and repair tools include two binary dumps, an ASCII/HEX file compare, and CRC "fingerprint." FILE MATE is available for popular CP/M and MS-DOS computers.

INTERSECTING CONCEPTS
(formerly MDG & Associates)

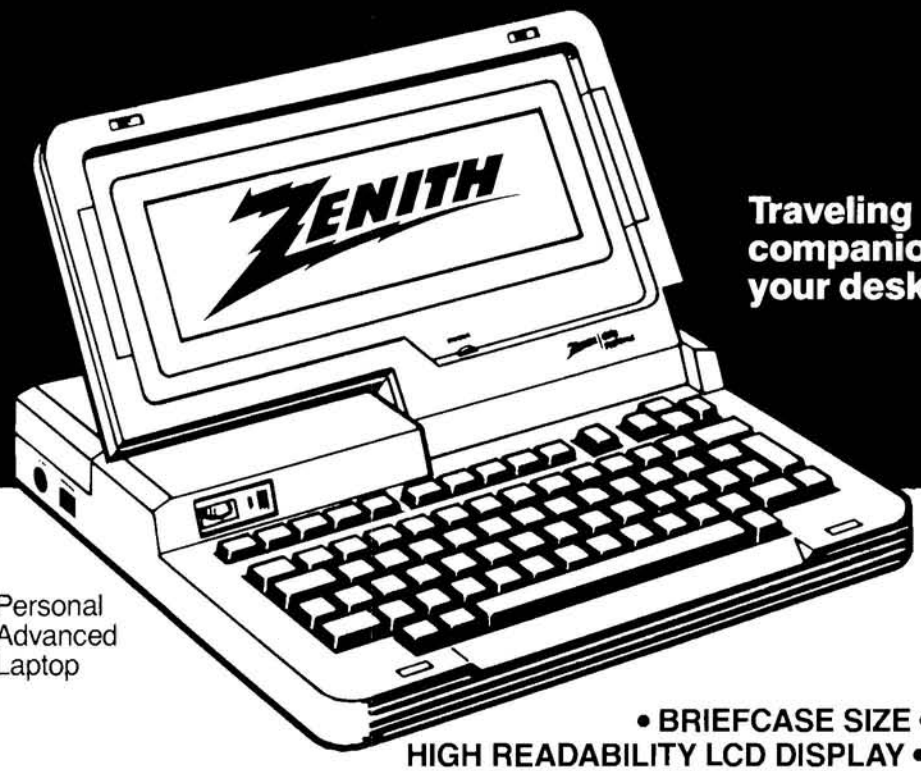
4573 Heatherglen Ct., Moorpark, CA 93021

ORDERING INFORMATION:

Include $3 per order for postage/handling. Overseas airmail add $10. To place COD orders:

Call TOLL FREE 24 hours:
800-824-7888 (Ask for Operator 251)
Alaska, Hawaii: 800-824-7919 (Ask for Operator 251)
Technical questions: call 805-529-5073

New from Heath/Zenith PAL*
7½ lb. LAPTOP POWERHOUSE



*Personal Advanced Laptop

Traveling companion to your desktop PC

HUG MEMBER DISCOUNTS APPLY

• BRIEFCASE SIZE • NEW TILT SCREEN/ HIGH READABILITY LCD DISPLAY • 214K of Built-in ROM

CONTAINS: Microsoft WORKS • Microsoft WORD • Microsoft MULTI PLAN • File Manager • BASIC • Calendar with Alarm • Communications and more. **• AC/BATTERY OPERATION • PARALLEL PRINTER PORT • SERIAL PORT • BUILT-IN MODEM**

PRICE ~~$1995~~ *SPECIAL INTRODUCTORY PRICE NOW* **$1195**

DSDD NASHUA DISKETTES

Now **$14.95** per box of 10

For IBM, Apple, Zenith, Commodore, Atari and more. Lifetime replacement guarantee. No limit.

SUPER PRICE

FREE DISK *Bring in ad for free diskette!*

PROTECTED DISK HOLDER

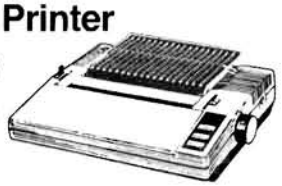
Lockable high-impact plastic holder keeps dust and fingers away from 70 disks.

Now **$12.88** Reg. $20

NEW Epson LX-80 Printer

$239.95

List Price $299.95



IBM-Compatible Computers

From **$1599**

Includes MS-DOS, MicroSoft WORD, MULTIPLAN and Diagnostics programs free.

PRICES GOOD THROUGH August 10, 1985.

Visit a store nearest you

ZENITH data systems Authorized Dealer

Your TOTAL SERVICE computer center

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

HZC-142

Heath ZENITH®

Units of Veritechnology Electronics Corporation

Computers & Electronics

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

REMark®

Official magazine for users of HEATH ZENITH computer equipment.

Volume 6, Issue 7 • July 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: *A computer's rendition of the Hyatt Regency O'Hare, Chicago, Illinois, site of the 4th Annual Heath/Zenith Users' Group International Conference.*

Photo Pixelization by Computer Creations, South Bend, Indiana.

VALUE BULLETIN

PRICE PERFORMANCE BREAKTHRUS

ZENITH data systems

FACTORY AUTHORIZED SERVICE

ALL SYSTEMS INCLUDE 320K RAM, COLOR GRAPHICS, DOS 2.1
Parallel & Serial Ports Microsoft Word & Multiplan.

COMPUTERS

Z150
Dual Floppy System ..... **$ CALL**

Z150
10 meg Winchester ..... **$ CALL**

Z160
Portable Dual Drive ..... **$ CALL**

Z160 Portable Winchester ..... **$ CALL**

MONITORS

ZVM122
Amber Monitor ..... **$ CALL**

ZVM123
Green Monitor ..... **$ CALL**

ZVM124
Hi Res Monchrome ..... **$ CALL**

ZVM133
Hi Res Color ..... **$ CALL**

OKI-DOKI

Factory Authorized Service

OKIMATE 20
Color Printer with Serial or Parallel Interface **$199**

OKI ML-182
120 CPS Dot Matrix IBM or Standard full graphics **$249**

OKI ML-192
160 CPS Dot Matrix IBM or Standard full graphics **$399**

FIRST QUALITY CABLES FOR ABOVE **$22.95**

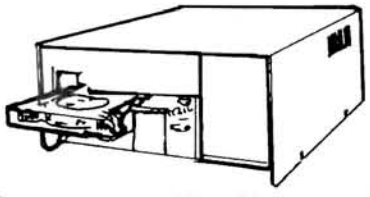
GUARDIAN DATA SYSTEMS "Protecting Your Information Investment"

25 MEGABYTE BACKUP Cipher 525 Cartridge tape drive uses shugart 824 floppy drive interface - G25X . . . . . **$ 495**

G25-Z100 Complete unit w/case power supply and software for Z100 . . . . . **$1095**

G25KT-1 Kit for Z100 includes drive software and interconnect cable . . . . **$ 695**

G25SFT Complete Software and Programmers Kit. Build your own with addition of drive . . . . . **$ 149**



G25PIK Programmers kit with source listing on disk - interface to non Z100 Systems . **$ 49**

DC600A Cartridge Tapes ea $29.00, 5 pack . . . . . **$ 139**

SYSTEMS INNOVATIONS INC.

HOURS: 9AM-6PM, MON.-FRI. • THURS. TIL 8PM • SAT. 10AM-3PM

44 Stedman Street,
Lowell, Massachusetts 01851
(617) 459-4449
"Your Complete Computer Resource Center"

VISA MasterCard AMERICAN EXPRESS

Controlled Data Recording Systems Inc.

Quality Products and Support for the Heath/Zenith Community

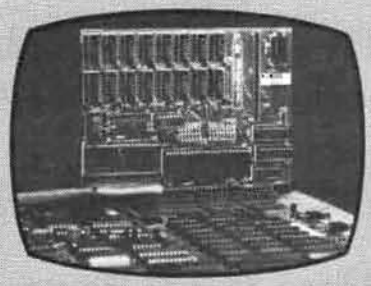
New for the H/Z89-90 Computer Users:

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.

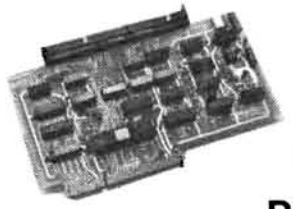
The Ram Drive Software (SRAM) allows one or two logical ram drives. The ram drive(s) can be located starting anywhere from logical A: to O: (standard drives get relocated). SRAM can be set to start at logical A: and warm boot with ram (no floppy disk accesses needed). Ram drive attatches to any of the 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

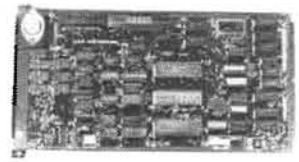
THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE:$395



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

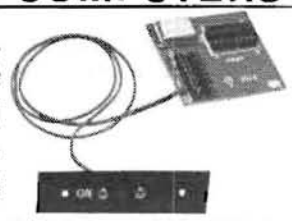
Allows owners of the FDC-880H or the FDC-H8 to read/write to the following disktypes:

Access S.S.D.D., Crommemco 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.

Heath/Zenith Users' Group

INTERNATIONAL
HEATH/ZENITH USERS' GROUP CONFERENCE

Official Conference Registration Form

O'Hare Hyatt Regency
Rosemont, Illinois
August 9, 10, 11

Name(s): _____

_____

_____

Company: _____

Address: _____

City: _____ **State:** _____ **Zip:** _____

Enclosed is $25.00 for each of the individuals listed above to attend the International HUG Conference being held the weekend of August 9, 10 and 11, 1985. Please send tickets along with information regarding hotel reservations and transportation.

Amt. Enclosed: _____ No. Attending: _____

For Our Information:

Which Heath/Zenith computer do you now operate? _____

Are you a Non-User-Attendee?	Yes	No
Are you a computer related manufacturer?	Yes	No
If yes, would you like exhibit information?	Yes	No
Are you, or anyone in your party, interested in activities in or around the Chicago area other than the Conference?	Yes	No

If yes, please indicate any suggestions you may have:

_____

Special Notice To Exhibitors:

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

For Your Information:

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

Please send your completed registration form or suitable copy to:

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

Registration(s) must be post marked no later than July 31, 1985. Cancellation will not be accepted after this date.

THE FOURTH INTERNATIONAL Heath/Zenith Users' Group CONFERENCE

Bob Ellerton
HUG Manager

It's time to pack your bags again! August 9–11, 1985 marks the dates for HUGCON IV. Since the introduction of the Heath/Zenith Users' Group Conference in 1982, users of Heath/Zenith computer products have had the opportunity to meet, share ideas, and generally have a good time with fellow HUG members from all over the globe. This year will be no exception to the traditions established at previous HUG Conferences. However, we have expanded a number of activities and the space required by these activities (make sure that you bring along your best walking shoes) so that you won't be rushed or crowded. The Fourth International Heath/Zenith Users' Group Conference will be one of the best ever. Let's take a look at the individual functions to give you a better idea of what to expect.

The Exhibit Area

Manufacturers from all over the country will be represented this year at the Heath/Zenith Users' Group Exhibit Area, located on the lower level of the O'Hare Hyatt Regency, Rosemont, Illinois. Unlike previous years, we have reserved the entire lower level for exhibits of interest to the Heath/Zenith users' community. A list of exhibitors appears elsewhere in this article and continues to grow each day as we approach the Conference. You will recognize some of the names both from ads that have appeared in REMark and from the list of exhibitors who attended last year's Conference. All Heath/Zenith computer products will be represented by a great variety of products scheduled to be shown at HUGCON IV. Does anybody know what the DEL board is? You'll get a chance to see it this year (according to reliable sources, someone will even get one)! To date, our generous exhibitors have donated over 50 software and hardware packages to be given away to lucky winners of the prize drawings within the Exhibit Area during the weekend.

Each of the prizes donated by the exhibitors will be selected from a drawing at the Heath/Zenith Users' Group Booth (#1). Prizes will be selected each hour with the winning numbers posted for two hours after the number is drawn. If a prize is not collected within the two hour limit, a new number will be drawn for that prize and posted along with other winning numbers. Be sure to check the number on your registration badge with the numbers appearing in the Heath/Zenith Users' Group Booth (#1) regularly so that you won't miss out on these exciting products.

The newly named Heath/Zenith Computer and Electronics Centers will be manned this year by selected store managers from all over the U.S. and Canada. These people make up the Retail Advisory Board for all the Centers and will be able to help you select a product right for you. Remember, "LET'S MAKE A DEAL!" are the key words for the temporary Center, again located on the lower level of the Hyatt. Those of you who have attended previous HUG Conferences will remember the great prices that these people offer. For the newcomer . . . Bring some bucks! You probably won't be able to find prices like these anywhere else.

The Exhibit Area will again have bulletin boards available so that you can post messages to fellow users. As in the past, the boards will have the following categories:

> **Wanted/Trade** – This bulletin board will be available for those of you wishing to purchase or trade items.
>
> **For Sale** – This bulletin board will be available for those of you wishing to sell computer related items.
>
> **Messages** – This bulletin board will be available for general messages of any nature.

These bulletin boards are made available for Heath Users' Group members. Special forms will be available for these boards. We ask that all ads placed on these boards be removed only by the individual that placed the information. Also, commercial ads are not to be posted on the bulletin boards. Space is provided for commercial advertising elsewhere in the Exhibit Area. If you have a commercial ad you would like to distribute, please contact Walt Gillespie, Exhibit Area Coordinator, Heath/Zenith Users' Group, Hilltop Road, St. Joseph, MI 49085, (616) 982-3789 for further information.

Introducing . . . The HUG Factory

Recently, the Heath Users' Group discovered "The HUG Factory" located at P.O. Box 4353, Louisville, Kentucky 40204. The HUG Factory sells a variety of HUG items, from T-shirts to coffee cups, and has agreed to be the guests of Heath Users' Group at the Fourth International HUG Conference. The HUG Factory will be a part of the Heath Users' Group Booth (#1). Judy and Barbara, from The HUG Factory, will be giving away FREE HUGS (guys, don't miss this one!) and showing off their unique products available at the Conference. If your club is interested in a special shirt or some special gift for your members, we suggest you stop by and chat with Judy and Barbara.

Friday Night At The Fourth International HUG Conference

The official opening of the Fourth International Heath Users' Group Conference is scheduled for Friday night, August 9, 1985,

in the Rosemont Ballroom within the O'Hare Hyatt Regency. The cash bar will open at 8:00 pm with actual activities slated for 9:00.

This year, we will be hearing from Chas Gilmore, Vice-president of Product Development from Heath Company. Chas was one of the original members of the team that brought the H-8 to life as a system back in 1977. He returned to Heath last fall with the tough assignment of revitalizing the kit products including Kit Computer Engineering. He will be discussing some of the results and expectations the last 10 months of effort have achieved and what the picture looks like to our user community. Chas will be open for any questions you may have after his talk.

Saturday At The Conference

Saturday morning highlights the beginning of the talks offered to those of you attending the Fourth International HUG Conference. This year, we are proud to have with us a whopping 23 speakers with expertise in just about all areas of computing. Probably your hardest choice will be the selection of the talks you would like to attend during the busy weekend. We have tried to select the widest possible range of speakers and topics that we could so that each of you attending this year's Conference will find a subject of specific interest to you. We have reserved most of the second level and part of the third level in the Hyatt for these discussion groups and seminars. Most of the talks are repeated so that you will have the opportunity to get to the sessions you really want to hear most. As you can see, a lot more space has been devoted to the Conference.

What's For Dinner?

The traditional HUG Dinner Party is planned for Saturday evening with a good menu and great company. Dinner warm-up and cash bar is set for 7:00 pm with dinner at 8:00 pm in the Rosemont Ballroom. Special awards will be presented at the party by Bill Johnson, President of Heath Company.

A lot of excitement is generated each year with the Grand Prize Drawing conducted as a portion of the dinner activities. What's up for this year? Let's take a look at some of the clues that were sent to HUG.

Carl Michelotti, President of Zenith Data Systems:

"You should be sure to be **AT** dinner!"

Bill Johnson, President of Heath Company:

"You could get **FULLY LOADED** at **8MHZ** with this one!"

Joe Schulte, President of Veritechnology:

"Three times **THE WORKS** is better than one!"

Well, there they are! It probably won't take a lot of research to determine what these prizes are, but we think they are all considered GRAND!

We have looked hard at the methods we have used to choose prizes in the past and have selected new rules for this Conference. These rules will be explained at the dinner. Remember, you must be present to win!

The dinner party will wind up the planned activities for Saturday, but we suggest you search out the various lounges and seating areas at the Hyatt for stray HUGgers at the end of the day. Even more fun than the formal activities, is the meeting between two or more HUG members who simply enjoy discussing computing (or whatever).

Sunday Brings More Talks

The final day of the Conference will allow you the opportunity to catch some of the last minute bargains and talks before we all head for home. In the past, the Heath/Zenith people have run some blue-light-specials. This tradition has carried over to many of the exhibitors that don't want to carry home all that extra baggage. Some advice though . . . DON'T WAIT! What you saw yesterday, may not be there today. Quantities available at HUGCON are very limited!

Review Of The Fourth International HUG Conference

As you can imagine, the weekend for August 9, 1985, is going to be a busy one. The increased seminar schedule and the amount of simple floor space should give you a good idea that this year's Conference is going to be the biggest yet. We have already heard that VEC (the Heath/Zenith stores) is planning a bunch of specials for your visit. The new exhibits and the number of prizes we already have is going to make for a fun weekend.

HUG wishes to extend thanks to the Exhibitors for making the Fourth International HUG Conference possible, and also for donating prizes to the members of the Heath/Zenith Users' Group who will be attending this year's event.

NOTE: The schedules, exhibitors, and activities of the Fourth International Heath/Zenith Users' Group Conference are subject to change. A final program for the events of the Conference will be available at the HUG Registration Booth on the days of the Conference. Changes that are made at the Conference will be announced within the Exhibit Area during normal hours. Tickets for the Conference are now being mailed every two weeks. If you have not received your tickets, please contact Margaret Bacon, HUG Secretary at (616) 982-3463. Those of you planning to attend for one day only may purchase tickets at the HUG Registration Booth. One-day tickets may be used for seminars and the Exhibit Area, however, one-day ticket holders will not be eligible for prizes selected either in the Exhibit Area or at the HUG Dinner Party. The $25.00 regular ticket entitles the holder to attend all events of the Conference. We cannot guarantee seating space to all seminars for all attendees. Seating will be on a first-come-first-served basis.

SPECIAL NOTICE: Dates furnished on the Hyatt Regency Reservation Cards have the Conference listed from August 8-12, 1985. Make sure you fill in the dates that you will be staying at the Hyatt, so that there is no mistake about your reservations and room requirements. Remember, actual Conference dates are from August 9-11,1985.

Fourth International Heath Users' Group Conference

Schedule of Events

Friday, August 9, 1985

2:00 pm ..... Registration Booth Opens
3:00 pm ..... Exhibit Area Opens
7:00 pm ..... Exhibit Area Closed
8:00 pm ..... Registration Booth Closed
8:00 pm ..... Grand Opening Warm-up (Rosemont)
9:00 pm ..... Grand Opening and Introductions (Rosemont)

Saturday, August 10, 1985

7:30 am ..... Registration Booth Opens
9:00 am ..... Exhibit Area Opens

Morning Discussion Groups Start Times

9:00 am 10:30 am

Afternoon Discussion Groups Start Times

12:00 pm 1:30 pm
3:00 pm 4:30 pm

5:00 pm ..... Exhibit Area Closed
5:30 pm ..... Registration Booth Closed
7:00 pm ..... Dinner Warm-up Open (Rosemont)
8:00 pm ..... Dinner, Keynote Address, Awards and Prizes (Rosemont)

Sunday, August 11, 1985

7:30 am ..... Registration Booth Opens
9:00 am ..... Exhibit Area Opens

Discussion Groups Start Times

9:00 am 10:30 am

12:00 pm ..... Registration Booth Closed
2:00 pm ..... Close of Fourth International HUG Conference

1985 International HUG Conference

Discussion Group Schedule

Saturday, August 10, 1985

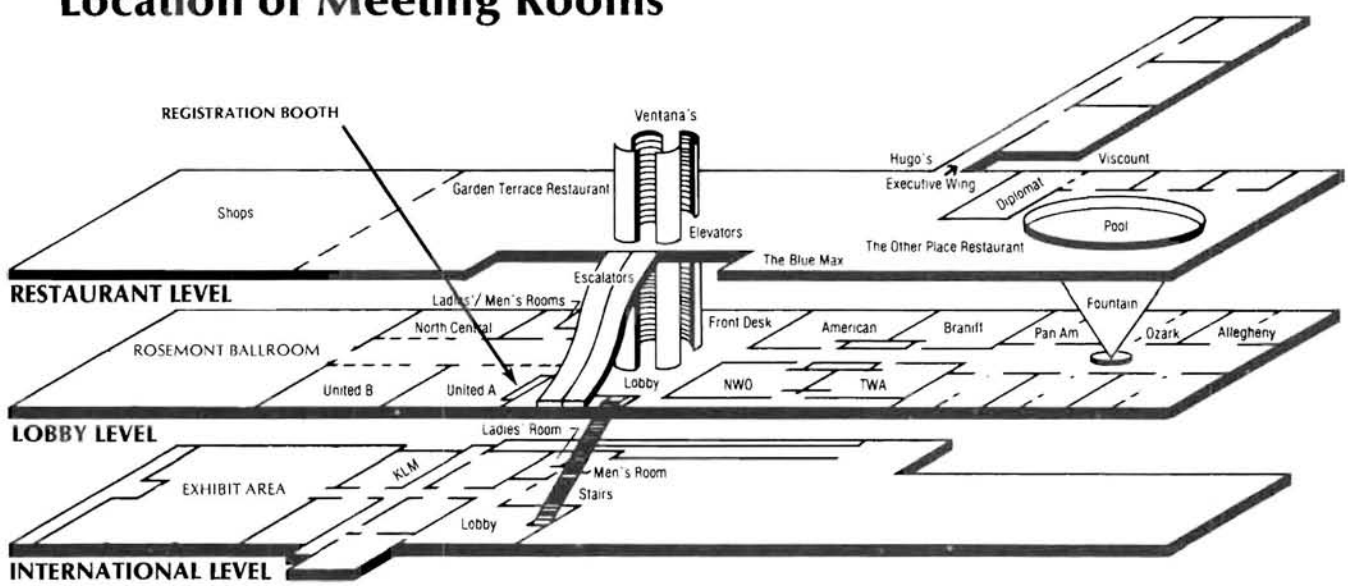
Time	Room	Subjects	Speaker
9:00 am	Ozark	Business Tools (A Comparison Of Some Popular Products)	Terry Jensen
	Pan Am	Introduction To 'C'	Walt Bilofsky
	American	MS-DOS Device Drivers	Mike Cogswell
	TWA/NWO	Hardware Workshop For Old And New Computer Users (Q&A Session)	Bruce Denton
	Diplomat	Making The Transition; Translating 8-Bit Software To 16-Bits	Pat Swayne
	Blue Max	Packet Radio For The Non-Ham	Wayne Wilson
10:30 am	Ozark	Data Communications With The Z-100 PCs	Bob Metz
	Pan Am	New Software Product(s)	Bob Meinschein
	American	MS-DOS Directory Trees, I/O Redirection, & Command Pathing	Bill Adney
	TWA/NWO	Software Workshop (Q&A Session)	Bill Parrott & Richard Mueller
	Diplomat	Heath's Role In Education	Ron Johnson & Ed Crowley
	Blue Max	New Systems And Peripherals	Jim Bungard
	Allegheny	Using The H/Z-100 For Business Graphics	Janet Hirsch
12:00 pm	Ozark	Intro To Computers For The Completely Intimidated (CP/M Oriented)	Susan Hayes
	Pan Am	A Look Back At Our First Eight Years	Charles Floto
	American	Graphics On the H/Z-100 Using 'C'	Dave Haskell
	TWA/NWO	New Software Product(s)	Bill Zurney & Dave Sandage
	Diplomat	New Hardware Product(s)	Mark Foster
	Blue Max	Public Domain Software	Bob Todd
1:30 pm	Ozark	Business Tools (A Comparison Of Some Popular Products)	Terry Jensen
	Pan Am	Publishing Software For 'Huggies'; Past And Present	Walt Bilofsky
	American	MS-DOS Device Drivers	Mike Cogswell
	TWA/NWO	Hardware Workshop For Old And New Computer Users (Q&A Session)	Bruce Denton
	Diplomat	Making The Transition; Translating 8-Bit Software To 16-Bits	Pat Swayne
	Blue Max	Introduction To The ZP-150	Jack Frank
3:00 pm	Ozark	Data Communications With The Z-100 PCs	Bob Metz
	Pan Am	New Software Product(s)	Bob Meinschein
	American	MS-DOS Directory Trees, I/O Redirection, & Command Pathing	Bill Adney
	TWA/NWO	Software Workshop (Q&A Session)	Bill Parrott & Richard Mueller
	Diplomat	Heath's Role In Education	Ron Johnson & Ed Crowley
	Allegheny	Using The H/Z-100 For Business Graphics	Janet Hirsch
4:30 pm	Ozark	Intro To Computers For The Completely Intimidated (MS-DOS Oriented)	Susan Hayes
	Pan Am	A Look Back At Our First Eight Years	Charles Floto
	American	Graphics On The H/Z-100 Using 'C'	Dave Haskell
	TWA/NWO	New Software Product(s)	Bill Zurney & Dave Sandage
	Diplomat	New Hardware Product(s)	Mark Foster

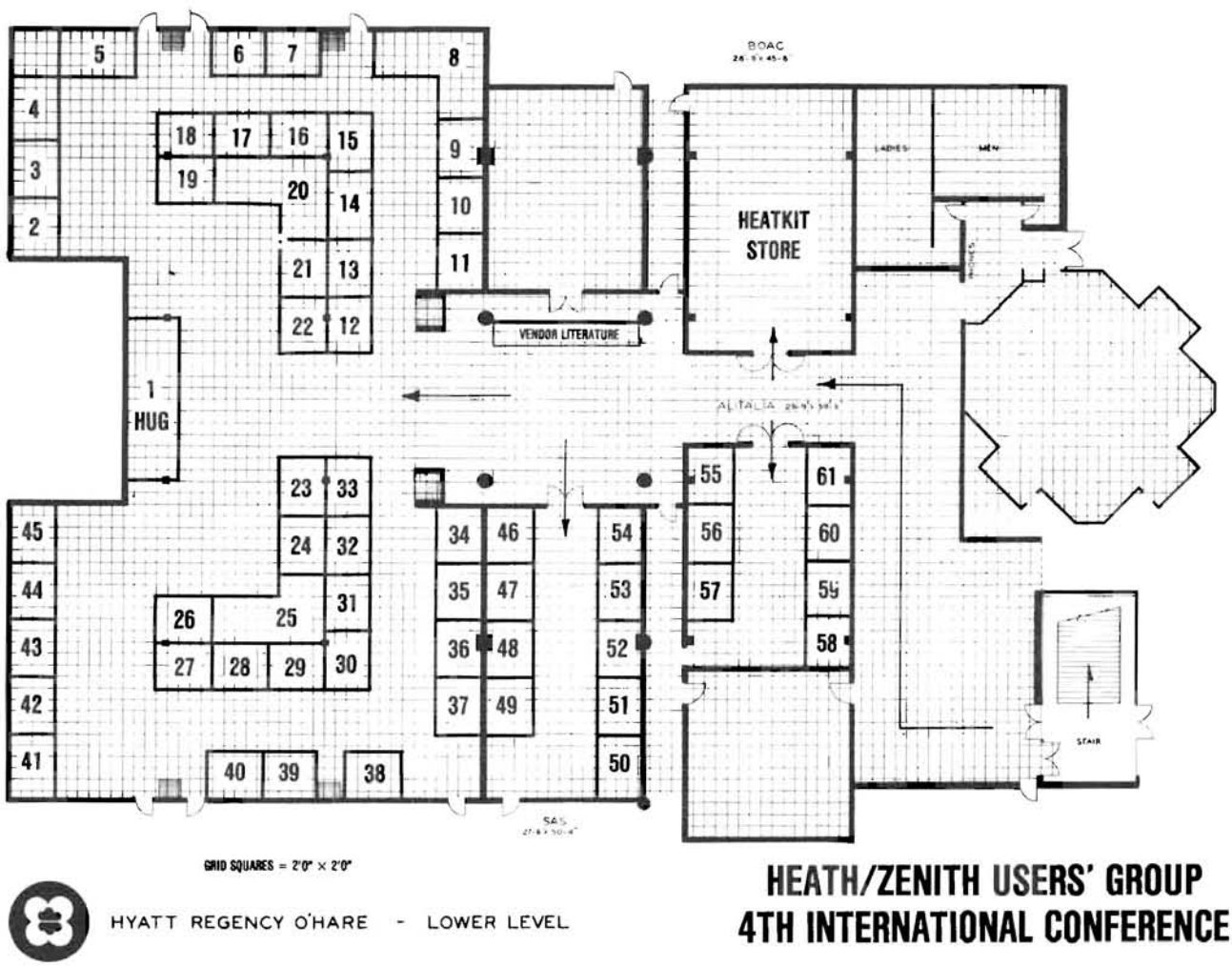
Discussion Group Schedule (Con't)

Sunday, August 11, 1985

9:00 am	Ozark	Business Tools (A Comparison Of Some Popular Products)	Terry Jensen
	Pan Am	Graphics On The H/Z-100 Using 'C'	Dave Haskell
	American	MS-DOS Device Drivers	Mike Cogswell
	TWA/NWO	Hardware Workshop For Old And New Computer Users (Q&A Session)	Bruce Denton
	Diplomat	Making The Transition; Translating 8-Bit Software To 16-Bits	Pat Swayne
	Allegheny	New Software Product(s)	Bill Zurney & Dave Sandage
10:30 am	Ozark	Intro To Computers For The Completely Intimidated (MS-DOS Oriented)	Susan Hayes
	Pan Am	MS-DOS Directory Trees, I/O Redirection, & Command Pathing	Bill Adney
	American	New Hardware Product(s)	Mark Foster
	TWA/NWO	Software Workshop (Q&A Session)	Bill Parrott & Richard Mueller
	Diplomat	Heath's Role In Education	Ron Johnson & Ed Crowley
	Allegheny	New Software Product(s)	Bob Meinschein

HYATT REGENCY O'HARE
Location of Meeting Rooms





1985 International HUG Conference Exhibitors

Advanced Software Technologies
BC Computers
CDR Systems, Inc.
Chuck Atkinson Products
Computer Consultants To Business
D-G Electronics
Del Professional Systems
Diverse Systems
Domino Computer Services
Dysan Corporation
ETTS Inc.
EWDP Software
Floppy Disk Services
Generic Computer Products
Graphic Design Systems, Inc.
Graphnet Systems, Inc.
Heath/Zenith Users' Group
Hilgraeve Inc.
Kres Engineering
Maximum Performance
New Orleans General Data
NewLine Software
Piiceon, Inc.
QuikData
Red-E Products
Redwood Development
S&K Technology, Inc.
Sextant Publishing Company
SigmaSoft & Systems, Inc.
Software Toolworks
Software Wizardry
Sunflower Software
Veritechnology Electronics Corporation
Barry Watzman
Weitzman Associates
Young Investment Research
Zeducomp
Zenith Data Systems
ZPay Payroll Systems

BUGGIN' HUG

Corrections To CONVERT! Data Lines

Dear Walt,

Referring to my article "CONVERT!" in Vol. 6, Issue 5, May 1985, I am receiving numerous complaints that the program, as published, will not convert between all parameters listed in the accompanying text.

To facilitate printing in your standard format (2 columns per page), you cut the lines of the DATA.CNV data file from the listing as I sent it to you, making almost all lines shorter. Herein lies the problem.

CONVERT! looks at each data line as a separate entity unless it is preceded with an asterisk (*), in which case it appends that line onto the previous one. By shortening the lines to fit your page format, those lines are erroneously assumed, by the program, to be separate data lines.

What to do?

If the reader will append each data line in the DATA.CNV listing, that is shown with an indented left margin, onto the line immediately above it, all will work as stated, OR if each indented line will be rewritten to include an asterisk followed by a single space in place of the three space indention, all will still work. Either solution will suffice. Dealer's choice! An example follows:

Current listing–

```
MIL 1000 INCH 12 FOOT 3 YARD 5.5 ROD 1 POLE 1 PERCH
   40 FURLONG 8 MILE 3 LEAGUE
```

Change to–

```
MIL  1000 INCH 12 FOOT 3 YARD 5.5 ROD 1 POLE 1 PERCH 40 FURLONG 8 MILE 3 LEAGUE
```

Or change to–

```
MIL 1000 INCH 12 FOOT 3 YARD 5.5 ROD 1 POLE 1 PERCH
* 40 FURLONG 8 MILE 3 LEAGUE
```

Hope this resolves everyone's problems with CONVERT!

Raymond Dotson
214 South Berkeley Boulevard
Goldsboro, NC 27530

Praises For The PT5000

Dear HUG,

When I traded my H-89 in for the H-120 in early 1984, I took advantage of the special on the PeachText 5000 package to replace and upgrade my '89 type word processor and data base managers. Since then, I have read a number of praises and criticisms of PT5000 in REMark and Sextant.

Let me add my experiences to the "Praises" side of the ledger. Although probably not the easiest program to grasp, the versatility of PT5000 for word processing (creating, editing, and formatting documents) and printing, plus the power of the List Manager module for data base creation and management, and the ability to integrate the modules have really streamlined my business applications.

Also, I have no complaint with the support from PeachTree. All of my inquiries have been answered in a timely and professional way. My only complaint would be the quality of the documentation. There are numerous errors, and sometimes insufficient explanations.

There have been repeated questions in this column regarding using PT5000 with "intelligent" printers, such as the MPI-99, Epson series, etc. I hope the following comments will lay this one to bed:

First, configure PT for a "draft" printer. The commands required by the various printers to change type style, page length, form feeds, etc. are usually ESC sequences, or simply a series of ASCII characters. These command codes are easily embedded in a document created with PeachText with the OUT command. Using the MPI-99 as an example, to start underlining command is ASCII[SO], or it's decimal equivalent (14). So to start underling in the text, you simply embed the Command×OUT14× immediately preceding the word or series to underline. ×OUT15× stops underlining. If the code is an ESC sequence, the command would look like ×OUT27,nl,n2,...× since the decimal value for ESC is "27". For another example, the start underline command for the Epson printer is ESC"-",1. Since the decimal value of "-" is 45, the embedded command would be: ×OUT27,45,1×. Very simple once you get the hang of it. (×OUT27,45,0× stops underlining). So, all ye out there with smart printers . . . do not despair. You can very easily communicate with your versatile printer using PeachText.

Here's a neat example of the Search and Replace function of PeachText. I just replaced my MPI-99 with and Epson LQ-1500, since I needed letter quality plus draft. (Incidentally, the LQ-1500 is quite a machine!) Well, naturally all of the printer commands I had embedded in my documents for the MPI-99 are different than the Epson commands. What to do? (Here I'll demonstrate changing the MPI underline command, OUT14, to the new Epson command, OUT27,45,1). Enter the EDIT mode in PeachText for the document to change. Invoke the Search and Replace feature with the F6 key and enter the following on the command line:

OUT14 (String to search for)
Hit F6 again and enter OUT27,34,1 (Replacement string)
Hit F6 again
Hit RETURN . . . and all occurrences of OUT14 are replaced with OUT27,45,1

Same can be done with any other commands. Sure, there are numerous other programs out there, some more powerful like DBASEII, Symphony, Framework, etc. Great, if you need the sophistication and can afford their prices and the additional RAM required. PT5000 can probably do about 80-90% of what these programs will do, BUT at a much more comfortable price!

Sincerely,

Bob Rothman
2119 Los Patos Drive
Palm Springs, CA 92264

P.S. Any HUGgies out here in the Palm Springs-Palm Desert area? If so, please contact me.

Similar Problems With The Print Utility

Dear HUG,

In regards to Richard Mueller's letter in REMark Dec. '84 issue concerning the MS-DOS 2's PRINT utility, I have had some similar problems. The cure that I have is to use the batch (.BAT) file feature of MS-DOS 2. Here you can always have the right set of PRINT parameters without continually specifying them and only specifying the file to print. I use LIST.BAT as my batch file to print listings. It contains, among other specifications, the following print statement:

```
PRINT %1 /P60/F/L5/S
```

```
where %1 is the name of file to be printed
      /P60  specifies 60 lines per page
      /F    form feed at end of file
      /L5   indent 5 characters (for hole punch)
      /S    spool print on (incase if switched off)
```

To use the batch file enter:

```
LIST <filename>
```

Regards,

David Podejke
27 MacIntosh Road
Norwalk, CT 06851

Strobe Pulse Won't Work

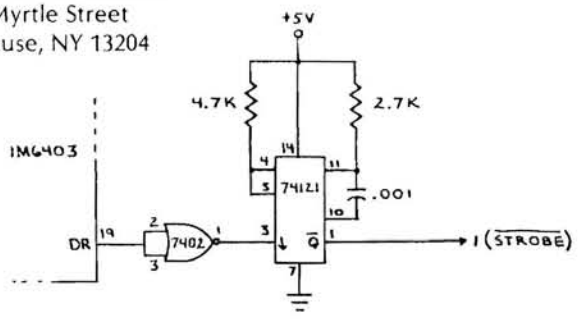
Dear HUG,

This letter is in reference to the April '84 article by J.D. Ross, "A Simple Serial To Centronics Parallel Converter." I built the circuit and experienced a random spacing problem between words. The problem was traced down to the strobe pulse. Mr. Ross used an RC network for this which will work, but required some toying with to obtain the desired results.

After conferring with other members of the SYRHUG, the following circuit was derived for the STROBE pulse which eliminates the guess work. The circuit generates a clean, 1.5uSec negative pulse and my converter now performs flawlessly. "Hats off," to J.D. Ross for his otherwise perfect article.

Sincerely,

Kurt G. Schotthoefer
903 Myrtle Street
Syracuse, NY 13204



GC1000 Most Accurate Clock

Dear HUG:

The program to use the serial output from the GC1000 Most Accurate Clock was a welcome addition to the H-100 utilities. I made the modifications, included here, to use it with serial port A, and to read the time when the tenths (seconds) digit is dim (which occurs frequently in my location). The program would give a "set error" with a dim tenths digit, because the clock transmits a question mark (?) in place of the tenths digit when same is dim. This would cause the DOS function to choke. The modification tests for this question mark and sets it to zero, if present. The question mark is output to the screen by the program after the time is set.

The modification for using power A allows a regular RS-232 cable to be used to connect the clock to the computer.

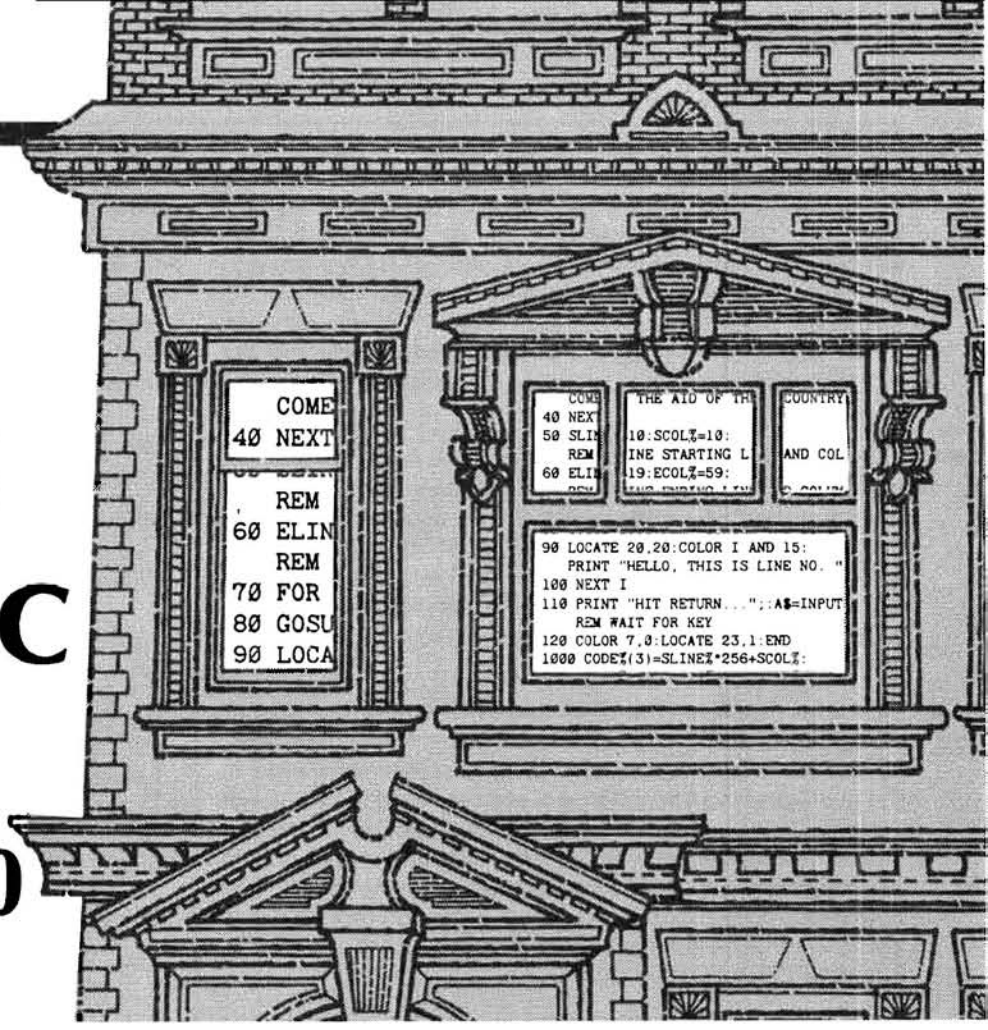
I have included only the new and modified program lines, and sufficient original lines for orientation to save space. All modifications occur in the first half of the program.

Modifications to GC1000 program (REMark Vol. 5, Issue 1) to also use port A, and to allow setting time when the tenths digit is dim.

```
; GC-1000       Serial port B     Serial port A
; pin 2         pin 3             pin 2
; pin 5         pin 20            pin 5
; pin 7         pin 7             pin 7

true      equ  0ffffh
false     equ  0
;    set the desired port to true, the other to false
aport     equ  true
bport     equ  false

          if aport
; Serial port A addresses:
epcib_dr  equ     0E8h      ;data register
epcib_sr  equ     0E9h      ;status register
epcib_mr  equ     0EAh      ;mode register
epcib_cr  equ     0EBh      ;command register
          endif

          if bport
; Serial port B addresses:
epcib_dr  equ     0ECh      ;data register
epcib_sr  equ     0EDh      ;status register
epcib_mr  equ     0EEh      ;mode register
epcib_cr  equ     0EFh      ;command register
          endif

; Since we have taken control of the port,
; we must disable the interrupt.

     in   al, 0F3h          ;read IMR

     if bport
     or   al, 00100000b     ;port B
     endif

     if aport
     or   al, 00010000b     ;port A
     endif

     out  0F3h, al          ;disable port interrupt

; In the event of a read error, we will loop back to here.

repeat:
     . . .
wait:
     . . .
     mov  al, 00100110b     ;enable Rx,  dtr starts GC1000
     out  epcib_cr, al
     . . .
```

Vectored to Page 96

Window Scrolling In BASIC On The H/Z-150



Pat Swayne
HUG Software Engineer

This article is the result of a letter from a user who asked if there is a way to put headings at the top or down the side of the screen on an H/Z-150, and protect the headings while text on the rest of the screen scrolled by. The ROM on the '150 has routines that allow you to scroll text on any portion of the screen while text on the rest of the screen remains stationary. This capability has always existed on PC-type machines including the original IBM PC (although its screen flickers badly when you scroll this way), but the writers of the various BASIC interpreters for these machines have never seen fit to include the capability in BASIC. They have, however, provided an easy way to call machine language routines from BASIC.

The program listed below illustrates a method that can be used to scroll text in a user defined window on the screen while text on the rest of the screen remains stationary.

```
10 SCREEN 0:WIDTH 80:CLS:FOR I%=0 TO 8:
   READ CODE%(I%):NEXT I%
20 FOR I=1 TO 23
30 PRINT "NOW IS THE TIME FOR ALL GOOD MEN TO
   COME TO THE AID OF THEIR COUNTRY."
40 NEXT I
50 SLINE%=10:SCOL%=10:
   REM DEFINE STARTING LINE AND COLUMN OF WINDOW
60 ELINE%=19:ECOL%=59:
   REM DEFINE ENDING LINE AND COLUMN OF WINDOW
70 FOR I=1 TO 30
80 GOSUB 1000:REM  SCROLL ONE LINE IN THE WINDOW
90 LOCATE 20,20:COLOR I AND 15:
   PRINT "HELLO, THIS IS LINE NO. ";I;
100 NEXT I
110 PRINT "HIT RETURN...";:A$=INPUT$(1):
    REM WAIT FOR KEY
120 COLOR 7,0:LOCATE 23,1:END
1000 CODE%(3)=SLINE%*256+SCOL%:
     CODE%(5)=ELINE%*256+ECOL%
1010 DEF USR=VARPTR(CODE%(0))
1020 CODE%(3)=USR(0):RETURN
2000 DATA &H6B4,&H1B0,&HB990,0,&HBA90,0,
     &H7B7,H10CD,&HCB
```

This program fills the screen with 23 lines of a message, and then opens a window in the lines and scrolls new text through the window. (Some of the lines of this program are broken due to a limit of 60 characters per line in listings placed in text columns. When you enter this program, type each numbered line as a single line.) The subroutine in line 1000 performs the scrolling within the window. It calls a short machine language routine that has been placed in the cells of the array CODE%(). Line 2000 contains the actual machine code in a DATA statement. This code is read into CODE%() at the beginning of the program. For those of you who work with assembly language, the code at line 2000 looks like this:

```
MOV     AH,6    ;GET SCROLL FUNCTION
MOV     AL,1    ;SCROLL ONE LINE
NOP
MOV     CX,0    ;DEFINE UPPER RIGHT CORNER
NOP
MOV     DX,0    ;DEFINE LOWER LEFT CORNER
MOV     BH,7    ;ATTRIBUTE FOR BLANK LINE
INT     10H     ;CALL ROM BIOS
RETF
```

The NOPs are for alignment, so that the values for the upper right and lower left corners for the window will be in their own cells of the array CODE%(). The subroutine fills in these values before it calls the machine code. At line 50 of the sample program, the

starting line and column are placed in the variables SLINE% and SCOL%. At line 60, the endling line and column are placed in ELINE% and ECOL%. The subroutine uses these values to fill in the cells of CODE%() so that the upper right corner and lower left corner of the window are properly defined. It is important to note that the subroutine numbers the lines and columns on the screen starting at zero, not one as the LOCATE command does. In line 60, the ending line of the window is defined as line 19, but in line 90, when we want to put text at the bottom of the window, we LOCATE to line 20.

The subroutine scrolls the text in the window upward one line each time it is called, and blanks the bottom line. To give the appearance of text scrolling upward in the window, as is done in the sample program, you can alternately call the subroutine and write text to the bottom line of the window. If you wanted to blank the window entirely before putting any text in it, you could just call the subroutine N times, where N is the number of lines in the window.

As written, the subroutine always blanks the bottom line to black after scrolling the lines up. If you would like it to blank to another color, locate the number &H7B7 in the data statement, and add the number of the new color between the H and the first 7. For example, if you wanted red, you would change the number to &H47B7. Change the number to that value, and run the program again. Notice that the background around the text in the window is now red, but behind the text it is still black. To make the entire background in the window red, change the COLOR statement in line 90 to COLOR 1 AND 15,4.

There are several variations that you can do to the subroutine to alter the way it works. For example, if you want text to scroll downward in the window instead of upward, change the number &H6B4 to &H7B4. If you want to scroll two lines each time the subroutine is called instead of one line, change &H1B0 to &H2B0. You can also blank the entire window, regardless of its size, by changing &H1B0 to &H0B0. For your application, you may want to use a separate subroutine for each variation, if you want to have several different window scrolling effects in the program. Or you could use just one subroutine, but several DATA statements containing the variations.

*

Graphics Characters And Machine Types

Mark J. Foster
Senior Systems Engineer
Systems Software Engineering
Zenith Data Systems Corporation

Hello once again! As I promised last time, this month's column will provide some details on how to define and display your own custom graphics character set. In addition, I will finally pick up on my February promise to provide you with some 'hooks' which will let your program determine what type of machine it is running on.

Defining Graphics Characters

While in text mode, the Z-100 PCs have a 256 character ROM-based 'font' — that is, there are 256 different characters which can be displayed. This font is contained not in the system ROM, but rather, in a special font ROM which is located on the video board. In graphics modes, however, only 128 characters are defined in the base machine. These 128 characters are located within the MFM-150 firmware. The 'other' 128 characters are therefore left available for you to define as you please. To allow you to do this, the firmware uses a pointer to a 128-character font buffer area, which is initialized to 0000:0000H. This pointer is stored in Interrupt vector 1FH (absolute location 0000:007CH). By pointing this interrupt vector to a buffer containing the new font, you can create your own custom font.

The ROM uses this new font when characters from 80H through 0FFH are displayed. The layout of the font buffer is eight bytes/character, where each byte is a single scan line of the character to be displayed. The total size of the buffer, therefore, is 1K byte (128 characters times eight bytes per character). The first eight bytes of the buffer are associated with character 80H, the next 81H, etc. The most significant bit of each of the bytes is displayed to the left, and the least significant bit to the right-most position in the character cell. The following example shows how a user-defined character might look:

```
 80H   40H  20H  10H  08H  04H  02H  01H
---------------------------------------
|XXXX|    |    |    |XXXX|    |    |    | = 88H
---------------------------------------
|    |XXXX|    |XXXX|    |    |    |    | = 50H
---------------------------------------
|    |    |XXXX|    |    |    |    |    | = 20H
---------------------------------------
|    |XXXX|    |XXXX|    |    |    |XXXX| = 51H
---------------------------------------
|XXXX|    |    |    |XXXX|    |XXXX|    | = 8AH
---------------------------------------
|    |    |    |    |    |XXXX|    |    | = 04H
---------------------------------------
|    |    |    |    |XXXX|    |XXXX|    | = 0AH
---------------------------------------
|    |    |    |XXXX|    |    |    |XXXX| = 11H
---------------------------------------
```

Therefore, the font table entry for this character would be: 88H, 50H, 20H, 51H, 8AH, 04H, 0AH and 11H. For this example, assume that the character above is the first character in the font table, so it will become character 80H. The following program fragment shows how the character could be displayed:

```
MOV   AL,80H      ;Get the character to be displayed
MOV   AH,0AH      ;Get the display-char function number
MOV   CX,1        ;Display character only once
INT   10H         ;Display the character!
```

Note that this code uses the write-character function within the ROM. Either this function (Function Code 10), or the write-character-with-attribute (Function Code 9) calls should be used. Use of the Write-Terminal (Function Code 14) with user-defined characters is not recommended, since early versions of the Z-100 PC firmware would not display user-defined characters using this function. Note, too, that the discussion last time concerning multiple-pages is not applicable here, since the graphics modes use all of the available video memory (in other words, there is only one video page in graphics modes).

A Cheap Trick

To create your own font it is first necessary to decide what you want the characters to look like. Many commercial packages are available to help you perform this function, but there is another technique which is very simple, and free! Assuming that you have the Microsoft Macro Assembler, you can define the characters in Binary. Just place a '1' where you want a pixel to be on, and a '0' where you want pixels to be off. This way, you can see what a character will look like as you define it! To define the character shown above, you would enter:

```
FONT    DB    10001000B      ;First Scan Line
        DB    01010000B      ;Second Scan Line
        DB    00100000B      ;Third Scan Line
        DB    01010001B      ;Fourth Scan Line
        DB    10001010B      ;Fifth Scan Line
        DB    00000100B      ;Sixth Scan Line
        DB    00001010B      ;Seventh Scan Line
        DB    00010001B      ;Eighth Scan Line
```

This example shows just a single character; to define more characters, just tack them on to the end of your font table. Once you've got all of the characters defined, you'll then need to set the font-buffer pointer. The following code segment shows how

you can do this (assuming that your font is called 'FONT', and that it is contained in your program's data segment):

```
INTR_SEGMENT SEGMENT AT (0000H)
       ORG  1FH * 4          ;Point to interrupt vector 1FH
FONT_OFS DW  ?               ;Define the font pointer offset
FONT_SEG DW  ?               ;Define the font pointer segment
INTR_SEGMENT ENDS

SETFNT:MOV  AX,INTR_SEGMENT ;Point to the interrupt segment
       MOV  ES,AX            ;Use ES to point to intr seg
       MOV  ES:FONT_OFS,OFFSET FONT ;Set up font offset
       MOV  ES:FONT_SEG,DS   ;Set up the font segment
```

Machine Identification

Lately, I've received several letters from software developers who need a method of determining whether or not their programs are running on a Z-100 PC, or on another machine. To do this, just have your program check bytes F000:800C through 800E. Contained here are the letters 'Z', 'D', 'S' (as in Zenith Data Systems). This information is contained in this location in every member of the Z-100 PC family. This can be useful for several reasons: first, you may want to ensure that your program will only run on a Z-100 PC (if you are developing custom software for a specific customer), and second, you may want to find out whether or not you can take advantage of a feature which is only offered in the Zenith machines.

The following code fragment will show you how to check the machine ID:

```
ROM_SEGMENT SEGMENT AT (0F000H)
       ORG  800CH            ;Point to start of ZDS ID
ZDS_ID DB   ?                ;Define variable to access ID
ROM_SEGMENT ENDS

CHK_ID: MOV  AX,ROM_SEGMENT   ;Point AX to the ROM segment
        MOV  ES,AX            ;Use ES to access ROM info
        CMP  ES:ZDS_ID[0],'Z' ;Is this a Zenith?
        JNE  NOT_ZENITH       ;No, must be something else
        CMP  ES:ZDS_ID[1],'D' ;Have we found ZD?
        JNE  NOT_ZENITH       ;No, bought the wrong machine!
        CMP  ES:ZDS_ID[2],'S' ;Last time - is it ZDS?
        JE   ZENITH           ;Hurrah! This is a Z-100 PC!
NOT_ZENITH:
        ... Complain about it here! ...
        JMP  DONE
ZENITH:
        ... Congratulate the user here! ...
DONE:
```

Another piece of information you may want to access is the ROM's version number. This is encoded as two ASCII digits at locations F000:8014 and F000:8015 (M.S. byte first). For example, ROM version 2.0 is listed as '2', '0'.

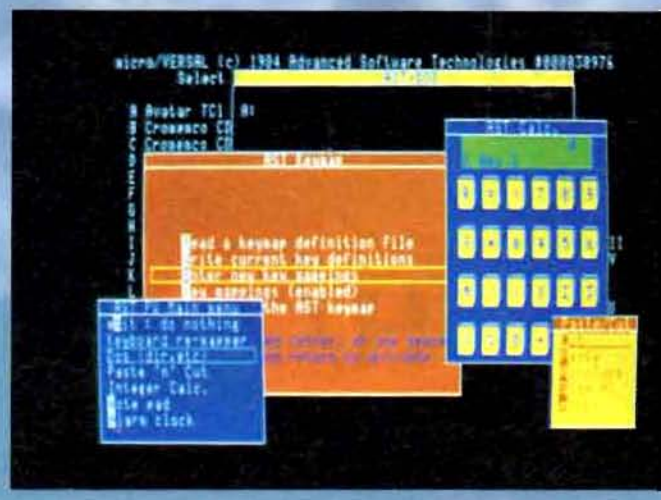
Next Time

In the next Z-100 PC Technical Column, I'll give you a quick introduction to some of the newer members in the Z-100 PC family, including the Z-138, Z-140, Z-158, and the Z-200! Until then, enjoy!

*

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.

Genie	Spotlight	Sidekick	
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

Installing The HA-108 Upgrade Kit On Older H/Z-100 CPU Boards

Pat Swayne
HUG Software Engineer

Hi, I just finished installing the 256k–8mhz modification in my computer, and I simply would like to say that this modification REALLY WORKS! It was my job to proofbuild this mod and these instructions on the H-110 in my office. My unit was one of the very first off the production line and had no updates to it at all. It is now running at 8 megahertz and has 768k of system RAM on the motherboard. The following hints will hopefully make your job a little easier and possibly eliminate some problems. First of all, this mod is not for the faint of heart or unsteady of hand! You will be cutting foils on your one thousand dollar motherboard! The only wire you should use is the kind used for wire wrapping (#30–#32 gauge solid conductor). Your iron should be of pencil variety and be no greater than 25 watts. Only the highest quality rosin core solder should be used. Obtain the IC sockets called for in the instructions from the Heath Parts Department; these sockets work properly. Remember, make one mistake, and the computer will no longer work! I made the same mistake twice during construction, and that was to place a jumper wire on the wrong IC pin number. The following hint should help you eliminate this problem. Before you begin, make some small labels out of masking tape or blank address label stock. These labels should fit between the two rows of pins on each IC that will have jumpers going to them. Remember, when you're looking at an IC from the bottom side, the pin number rows are reversed. The following is what one label may look like:

14		1
13	U	2
12	2	3
11	3	4
10	3	5
9		6
8		7

Paste these labels on the foil side of the motherboard. Believe me, these will help, and speed up construction because you won't have to be searching for the IC you're looking for while trying to concentrate on which pin the next jumper should go to. Another problem that can occur is during the insertion of IC's. It is very easy to have a pin on an IC bend underneath itself, or slip outside of the socket, or just fold like an accordion when being inserted. To help alleviate this problem, bend each row of IC pins against a flat surface so that they're at a 90 degree angle to the IC body. Do both rows of pins on each IC. Once the IC is inserted, examine it very carefully; look at each individual pin!

The following hint works quite well for me, and will assist you in making the jumper connections on your board. Each jumper connection you make should be done with wire wrap type wire as mentioned earlier. Strip back the insulation 1/32". Yes, that's one thirty second of an inch! Add only a very small amount of solder to the IC connection to which the jumper will be added. Now take your jumper wire, and re-heat the connection again until the bare wire slides into the solder. Remove your iron from the connection, and hold the wire steady until the connection cools. You DON'T have to pre-tin the wire with solder.

NOTE: The Heath HA-108 modification kit is required to perform this modification. Heath/Zenith can guarantee proper system operation only if these parts or additional memory I.C.'s are purchased from them. This warning is made due to a specification problem some I.C. manufacturers have had.

Perform each step of the instructions carefully. Under no circumstances should you omit the power up tests. These power up tests are meant to help catch problems during each phase of modification. Finally, don't, don't hurry. This modification DOES, and CAN work for you!!

Jim Buszkiewicz
HUG Software Developer

The HA-108 Upgrade kit allows you to alter newer H/Z-100 main circuit boards (boards with part no. 85-2806-1) to run at 8 MHz instead of 5 MHz and to use 256k RAM IC's instead of 64k IC's. This article contains instructions for modifying older boards (with part no. 85-2653-1) so that the kit can be installed. If your H-100 kit came with full height drive(s), you most likely have the old board. The part number on the main circuit board is printed near Y101 under the video board.

WARNING! WARNING! WARNING!

INSTALLATION OF THESE MODIFICATIONS REQUIRES THAT YOU CUT A NUMBER OF TRACES ON THE CPU BOARD AND CONNECT SEVERAL JUMPER WIRES. THIS WORK SHOULD ONLY BE DONE BY PERSONS EXPERIENCED IN MAKING SUCH CHANGES TO PRINTED CIRCUIT BOARDS. THE POSSIBILITY EXISTS OF PERMANENTLY DAMAGING YOUR CPU BOARD. NEITHER THE HEATH/ZENITH TECHNICAL CONSULTANTS NOR THE HEATH USERS' GROUP STAFF MAY BE ABLE TO HELP YOU IF YOU ATTEMPT THE MODIFICATION AND YOUR COMPUTER DOES NOT WORK.

These instructions are divided into three sections: some modifications that are common to both the memory and speed changes, the memory modifications, and the speed modifications. You may omit the modifications dealing specifically with either the speed or 256k memory changes if you wish. However, it is recommended that if you perform the speed modification that you also perform the memory modification, since your original memory IC's are probably too slow for 8 MHz, and will have to be replaced anyway.

Preparation

To perform these modifications, you will need the parts contained in the HA-108 kit plus 10 feet of 30 to 24 gauge solid wire (30 gauge "wire wrap" wire is best), two 18-pin IC sockets (Heath part no. 434-310), and a 14-pin socket (part no. 434-298).

Disassembly

Perform the steps in the HA-108 Installation Manual that apply to your computer up to but not including the page titled "Main Circuit Board Upgrade". Then perform the following steps:

( ) Remove any cables that are connected to the serial, parallel, monitor, or light pen connectors on the back of the computer.

( ) Remove all cards from the S-100 card cage. If you have a Winchester controller, you will have to remove the power connector from it first.

( ) Remove the two sheet metal screws that connect the card cage to the back of the computer. Remove the 4 8-40 X 3/8" screws that connect the card cage to the main circuit board. Set the screws aside temporarily.

( ) Remove the S-100 card cage and set it aside temporarily.

( ) Remove the 9 8-40 X 3/8" screws and the three hex standoffs that hold the main circuit board to the chassis and set them aside temporarily.

( ) Lift the main circuit board slightly and slide it forward until the connectors at the rear are clear of the computer back plate. Then lift the right side of the board carefully until it is above the side of the computer chassis. Unplug the power connectors at P101 and P102, and remove the main circuit board from the chassis. Set the chassis aside temporarily.

Main Circuit Board Modifications

Common Modifications

The following modifications should be performed regardless of whether you are going to make the memory modifications, the speed modifications, or both. Some of the changes have little to do with the speed or memory changes, but are design improvements.

Memory Modification (common)

The changes in this section affect memory timing and addressing.

( ) Prepare one of the 18-pin IC sockets by bending pins 9 and 10 in toward the center of the socket, and pins 15 and 16 outward to 90 degrees from the other pins.

( ) Connect one end of a 2" (5 cm) length of jumper wire to pins 9 and 10 of the socket (so that pins 9 and 10 are connected together). This wire should branch out at the pin 9 side of the socket. Connect one end of a 6" (15 cm) length of wire to pin 15, and another 6" length to pin 16. Set the prepared socket aside temporarily.

( ) Position the CPU board so that the component side is up, with the S-100 connectors away from you.

( ) Remove the IC at U111 and set it aside temporarily. Install the prepared 18-pin socket at U111 so that pin 1 of the socket is inserted at pin 1 of U111. Pins 9 and 10 of the 18-pin socket will be outside of U111. Make sure that pins 15 and 16 of the new socket do not contact the pins of the original socket that are directly below them.

( ) Connect the wire coming from pins 9 and 10 of the 18-pin socket to the leg of C112 that is nearest to pin 1 of U111. Before making the connection, route the wire close to the board and cut off any excess length.

( ) Connect the wire coming from pin 15 of the 18-pin socket to the feed-through hole that is nearest to pin 4 of U173. Route the wire close to the board and cut off any excess length before making the connection. (The feed-through hole is connected by a trace on the back of the board to U173-2.)

Note: To connect a wire to a feed-through hole, turn the board over to the solder side and make sure that there is no masking on the pad at the feed-through hole. If there is, gently scrape it off with a small knife. Then pass the end of the wire through the hole (from the component side) and solder it as you would solder the wire from a resistor or capacitor. Cut off any excess wire.

( ) Connect the wire coming from pin 16 of the 18-pin socket to the feed-through hole between U156 pin 7 and C169. Route the wire close to the board and cut off any excess length before making the connection. (The feed-through hole is connected to U173-1.)

( ) Reinstall the IC that you removed from U111, making sure

that pin 1 of the IC is in pin 1 of the socket. Pins 9 and 10 of the new socket will be empty.

( ) Cut the traces coming from pins 7 and 8 of U173 on the component side of the board. The trace from pin 8 passes to the right of C170. The trace from pin 7 ends at a feed-through hole 5/16" (.7 cm) below the lower leg of C170. See Figure 1.

( ) Cut the trace coming from pin 9 of U173 on the component side of the board. See Figure 1.

( ) Position the board so that the foil side is up, and the S-100 connectors are away from you.

Note: Pin numbers of U111 in the following steps refer to the original 16-pin socket numbering.

Note: Make each trace cut on the bottom side of the board on the specified trace before it reaches any other pad, feed-through hole, or another trace.

( ) Cut the trace coming from U166 pin 2. See Figure 2 for this and the next 4 steps.

( ) Cut the trace coming from U151 pin 10.

( ) Cut the trace coming from U173 pin 18.

( ) Cut the trace coming from U110 pin 4.

( ) Cut the trace coming from U111 pin 9. Note the position of the first feed-through hole that this trace comes to.

( ) Jumper U173 pin 7 to U111 pin 3.

( ) Jumper U173 pin 8 to U111 pin 2.

( ) Jumper U173 pin 9 to U149 pin 4.

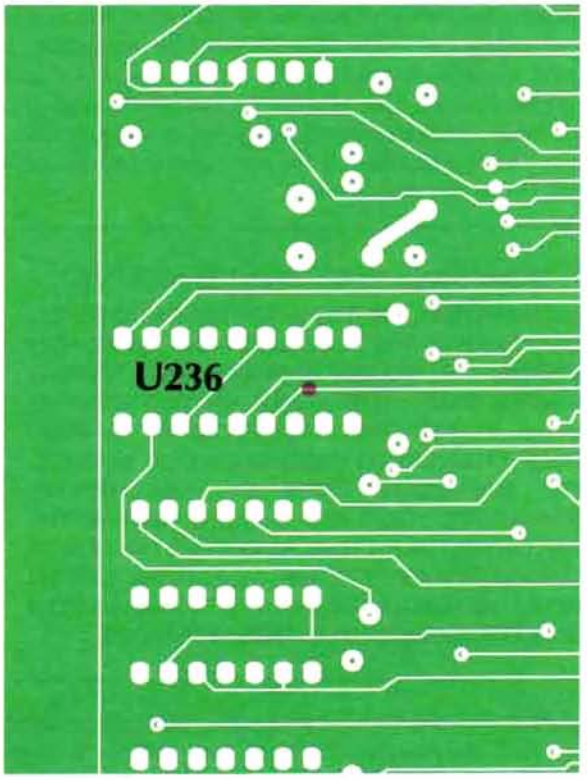


Figure 3. Trace cut at U236 on the foil side of the board.

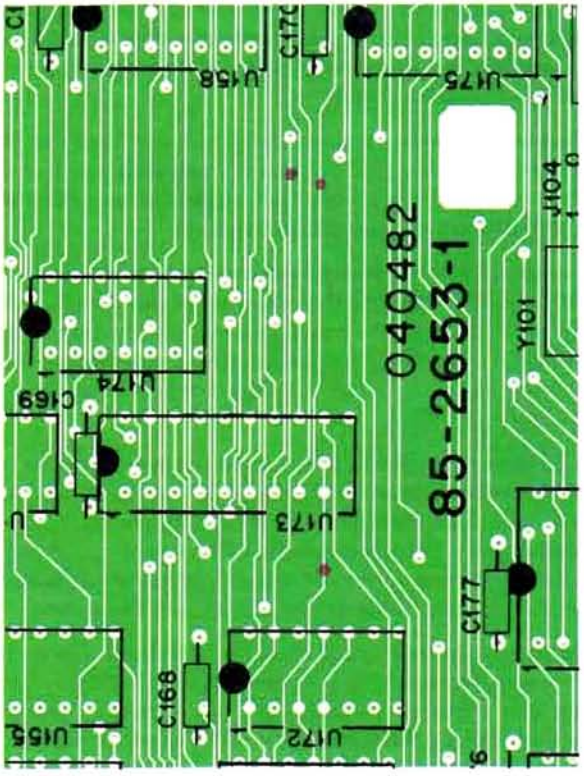


Figure 1. Component side trace cuts around U173.

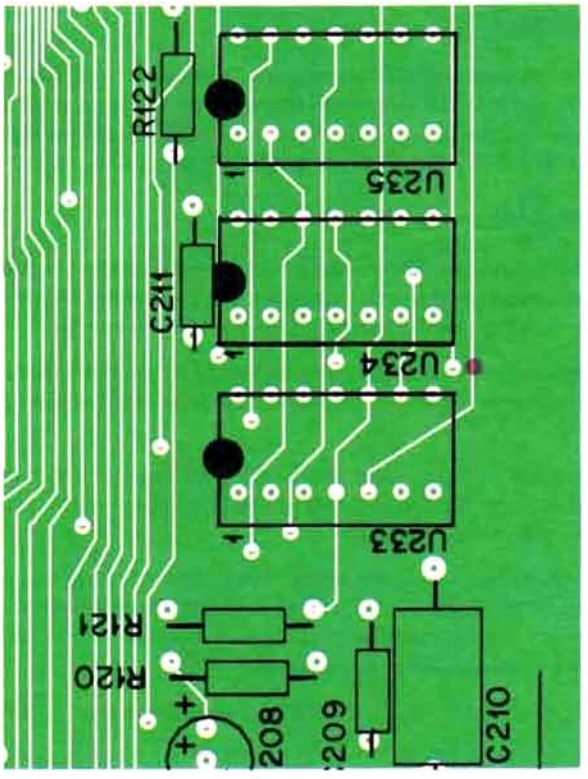


Figure 4. Component side trace cut near U233.

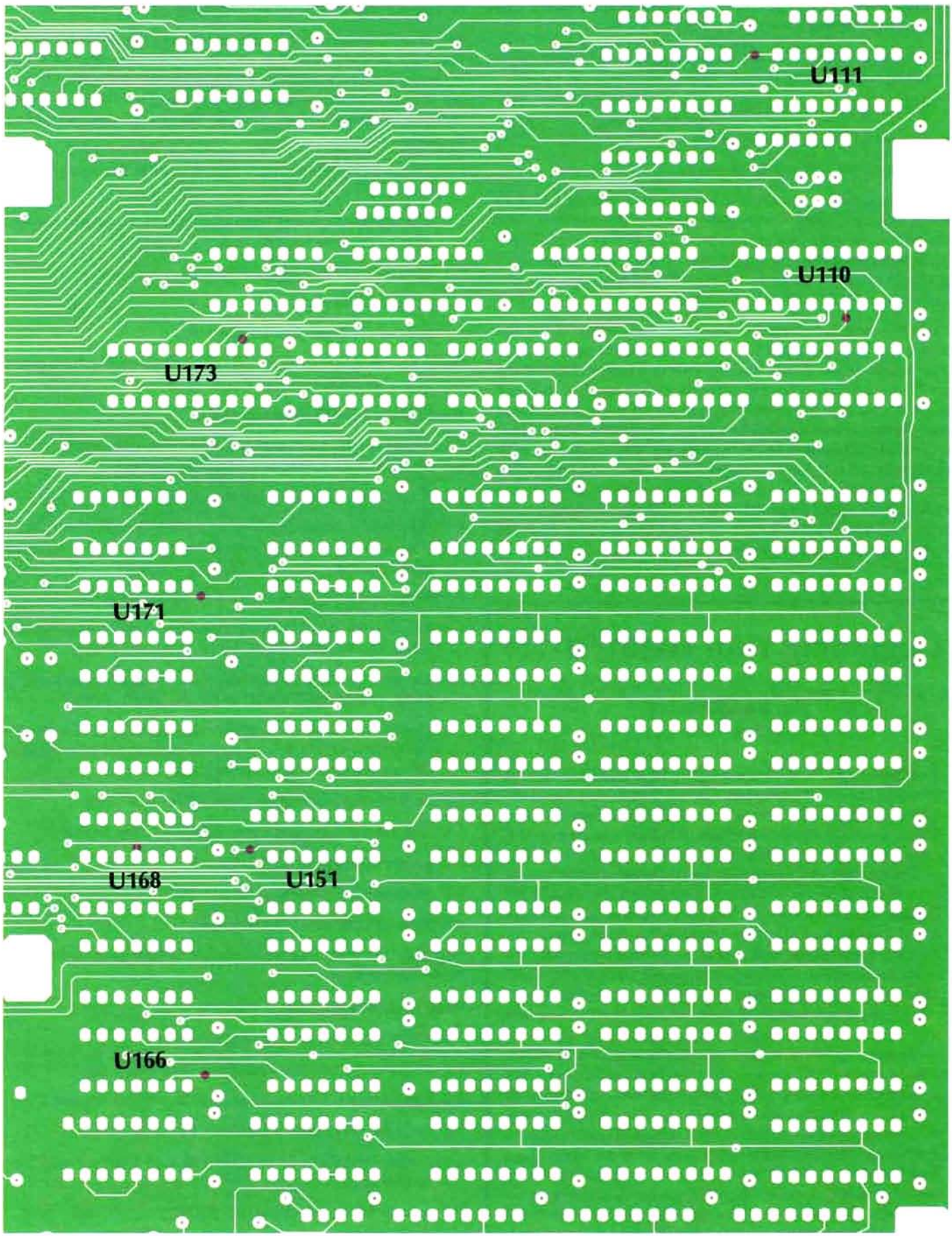


Figure 2. Trace cuts on the foil side of the board.

( ) Jumper U173 pin 9 to U110 pin 5. Note that there are two jumpers coming from U173 pin 9.

( ) Jumper U173 pin 17 to the feed-through hole that connects to the trace coming from U111 pin 9 that was cut earlier.

( ) Jumper U173 pin 18 to U149 pin 10.

( ) Jumper U110 pin 12 to U146 pin 1.

( ) Jumper U110 pin 4 to U149 pin 12.

Note: At this point, you should test the main circuit board to ensure that everything is working correctly. The test described here will work if you have version 2.5 or higher of MTR-100 installed in your board. If you do not, or you do not know what version you have, remove the IC at U190 and replace it with the IC marked 444-276 from the HA-108 kit.

To test the board, reconnect the power connectors (from the chassis) that were removed from P101 and P102. Do not install the video board or any S-100 boards. You can either reinstall the board in the chassis, or hold it so that no metal chassis parts touch the foil traces (hold the board at one of the screw pads), and plug in the computer and turn the power on. You should hear a short beep when the power is first turned on, and a longer beep about two seconds later. If you do not get these indications, your board is malfunctioning, and you should not continue with further modifications until you get it working again. Remove the power connectors from P101 and P102, and remove the board from the chassis when the tests are done.

Upgrading Your H-101/H-121

Jim Buszkiewicz
HUG Software Developer

For those of you fortunate enough to have purchased either the H-101 or H-121 computer systems, Heath now has available a kit which will increase your clock speed from 5mhz to 8mhz, as well as provide for 768k of system memory on the main motherboard (256k is provided with the kit). The model number of this upgrade kit is HA-108. This kit comes with about 38 new IC's (including memory), and a new CPU crystal. If you're careful, the installation of this mod will take about 2 hours. No foil cuts or jumper wires are required, and the only solder joints made are those to the new crystal. The main motherboard isn't even removed from the chassis! Other than a few problems with the assembly manual, my modification installed quite easily, and worked the very first time! Increasing the memory capacity of your system is easily accomplished by adding additional banks of 41256, 150nS memory IC's. Each bank requires an additional 9 IC's and two more banks can be added for whopping total of 768k! Even though these memory I.C.'s are available from outside vendors for about $6 each, Heath/Zenith can only guarantee proper system operation with memory chips purchased from them. It has been discovered that some manufacturers' memory chips are not being manufactured to proper specifications. If you thought your machine was fast running at five megahertz, wait till you see it at eight!

Refresh Clock Modification (common)

This modification reverses the polarity of the refresh clock. It may have already been done to your computer, since it is necessary for proper operation of the Winchester controller. If there are jumper wires connected to pins 8, 9, and 10 of U130, this modification has already been done, and the next 5 steps should not be performed.

( ) Position the main circuit board so that the solder side is up.

( ) Cut the trace coming from U168 pin 11. Make the cut right next to the pin. See Figure 2.

( ) Jumper U130 pin 5 to U130 pin 10.

( ) Jumper U130 pin 8 to U152 pin 1.

( ) Jumper U130 pin 9 to U225 pin 3.

Note: You should test the board as described earlier, before proceeding with other modifications.

Ready Logic Modification (common)

This modification is a design improvement, not directly related to the use of 256k RAMs or 8 MHz operation.

( ) Position the board so that the component side is up.

( ) Prepare the remaining 18-pin socke: by bending out pin 3 90 degrees from the other pins. Connect a 4" (10 cm) length of wire to pin 3.

( ) Prepare the 14-pin socket by bending out pin 6.

( ) Remove the IC at U220, insert the prepared 14-pin socket in U220, and reinstall the IC in the prepared socket.

( ) Remove the IC at U236, insert the prepared 18-pin socket in U236, and reinstall the IC in the prepared socket.

( ) Connect the wire coming from the socket in U236 to pin 6 of the socket in U220. Route the wire close to the board and cut off any excess before making the connection.

( ) Position the board so that the solder side is up.

( ) Cut the trace coming from U236 pin 4. See Figure 3.

( ) Jumper U236 pin 7 to U236 pin 9.

( ) Jumper U236 pin 4 to U206 pin 13.

( ) Jumper U206 pin 1 to U220 pin 5.

Test the board as described earlier before proceeding with other modifications.

Swap Logic Modification (common)

This modification is a design improvement.

( ) Position the board so that the solder side is up.

( ) Cut the trace coming from U171 pin 10. See Figure 2.

( ) Jumper U155 pin 8 to U156 pin 13.

( ) Jumper U156 pin 12 to U171 pin 1 or 2 (pins 1 and 2 are connected).

( ) Jumper U156 pin 11 to U171 pin 10.

Test the board as described earlier before proceeding with other modifications. If you do not intend to make the 256k RAM mod-

ification, skip the next section and proceed to "8 MHz Modifications".

256k RAM Modification

Perform this modification if you want to use 256k RAM IC's in place of the 64k IC's currently installed on your board. If you perform this modification, you will no longer be able to use a Z205 memory card within the normal 768k address space of the H/Z-100. The memory modification presented previously must be installed.

( ) Position the main circuit board so that the component side is up and the S-100 connectors are away from you.

( ) Remove the 9 IC's located at U101 through U109. Insert them into the conductive foam supplied with the HA-108 kit.

( ) Remove the IC's located at U117 through U125 and save them.

( ) If there are any IC's at U137 through 145, remove them and save them in conductive foam.

( ) Position the board so that the solder side is up.

( ) Jumper all pin 1's of U101 through U109 together.

( ) Jumper all pin 1's of U117 through U125 together.

( ) Jumper all pin 1's of U137 through U145 together.

( ) Jumper U109 pin 1 to U125 pin 1 to U145 pin 1.

( ) Jumper U125 pin 1 to RP101 pin 8. There will be 4 jumper wires at U125 pin 1 after this jumper is installed. Because of this, you may want to jumper U109 pin 1 instead of U125 pin 1 to RP101 pin 8.

( ) Jumper RP101 pin 9 to U155 pin 6.

( ) Jumper U155 pin 5 to U155 pin 11.

( ) Jumper U155 pin 3 to U155 pin 4.

( ) Jumper U155 pin 2 to U166 pin 8.

( ) Jumper U155 pin 13 to U166 pin 9.

( ) Jumper U155 pin 1 to U111 pin 9.

( ) Jumper U155 pin 12 to U111 pin 10.

( ) Jumper U155 pin 13 to U146 pin 1. There are now two jumpers at U146 pin 1, and two at U155 pin 13.

( ) Position the board so that the component side is up.

( ) Install 9 256k dynamic RAM IC's (#443-1268) at locations U101 through U109

( ) If you have 9 more 256k RAM's, install them at U117 through U125.

( ) If you have 9 more 256k RAM's, install them at U137 through U145.

( ) Remove the IC at U166 (74LS14, #443-872) and replace it with a 74ALS1004 (#443-1253).

( ) Remove the IC at U155 (74LS00, #443-728) and replace it with a 74AS00 (#443-1243).

( ) Remove the IC at U173 (#444-130) and replace it with a #444-368.

( ) Remove the IC at U110 (#444-126) and replace it with a #444-367.

( ) Remove the IC at U111 (#444-104) and replace it with a #444-366.

Test the board as described earlier before proceeding with other modifications. Proceed to "Reassembly" if you are not going to perform the 8 MHz modifications.

8 MHz Modifications

The 8 MHz modifications are given in three sections. You can test the board after each section is completed.

Wait State Modification

This modification adds one extra wait state to input/output operations.

( ) Position the board with the component side up and the S-100 connectors away from you.

( ) Cut the trace coming from U233 pin 5. See Figure 4.

( ) Position the board with the solder side up.

( ) Jumper U233 pin 1 to U234 pin 1.

( ) Jumper U233 pin 3 to U234 pin 3.

( ) Jumper U233 pin 4 to U234 pin 4.

( ) Jumper U233 pin 5 to U234 pin 2.

( ) Jumper U234 pin 5 to U226 pin 14. This will be a long jumper, so route it carefully.

Test the board as described earlier before proceeding with other modifications.

IC Replacement

Remove and replace IC's at the following locations. The locations marked with an asterisk (*) will have been replaced already if you did the 256k modification.

	IC number	Old IC	New IC
( )	U149	#41-10	#41-19

Note: Pin 1 of #41-19 is marked with the word "IN".

( )	U166*	74LS14, #443-872	74LS1004, #443-1253
( )	U200	74LS368, #443-1024	8T98, #443-1184
( )	U153	74LS280, #443-1001	74S280, #443-1186
( )	U221	74LS32, #443-875	74ALS1032, #443-1185
( )	U234	74ALS74, #443-1051	74S74, #443-900
( )	U155*	74LS00, #443-728	74AS00, #443-1243
( )	U146	74LS257, #443-1037	74F257, #443-1120
( )	U173*	#444-130	#444-368
( )	U190	#444-87-5 or other #	#444-276

Note: U190 may have already been replaced, to test the board.

( )	U126	74LS244, #443-791	74F244, #443-1065
( )	U128	74LS257, #443-1037	74F257, #443-1120
( )	U211	8088, #443-1009	8088-2, #443-1187

( )	U213	74LS373, #443-837	74ALS373, #443-1182

Note: Do not confuse the 74ALS373 with the 74AS373.

( )	U241	74LS244, #443-791	74ALS244, #443-1096
( )	U133	74LS373, #443-837	74AS373, #443-1266
( )	U195	74LS240, #443-754	74ALS240, #443-1181
( )	U214	74LS244, #443-791	74ALS244, #443-1096
( )	U162	74LS244, #443-791	74ALS244, #443-1096
( )	U180	74LS367, #443-857	74F367, #443-1267
( )	U196	74LS373, #443-837	74ALS373, #443-1182
( )	U163	74LS244, #443-791	74ALS244, #443-1096
( )	U181	74LS244, #443-791	74ALS244, #443-1096
( )	U197	74LS373, #443-837	74ALS373, #443-1182
( )	U198	74LS373, #443-837	74ALS373, #443-1182
( )	U217	74LS244, #443-791	74ALS244, #443-1096
( )	U227	74LS373, #443-837	74ALS373, #443-1182

Test the board as described earlier before proceeding with other modifications.

Crystal Replacement

Replace the crystal at Y103 with a 24 MHz crystal (#404-681) by following the first seven steps on the page titled "Main Circuit Board Upgrade" in the HA-108 Installation Manual. Test the main circuit board as described earlier.

Reassembly

Replace the main circuit board in the chassis by following these steps:

( ) Plug the power connectors at the bottom of the chassis to P101 and P102. Then carefully position the board to the chassis bottom and slide it backwards until the rear connectors are through the holes in the backplate.

( ) Secure the board with the 9 8-40 X 3/8" screws and the three hex stand-offs that you removed earlier.

( ) Replace the S-100 card cage and secure it to the main circuit board with 4 8-40 X 3/8" screws, and to the backplate with 2 #4 sheet metal screws.

( ) Replace all S-100 cards that you removed earlier, except for any Z205 cards if you made the 256k memory modification.

( ) Replace any cables that were unplugged from the back of the computer.

Complete the reassembly of your computer by proceeding to the section in the HA-108-1 Installation Manual entitled "Video Logic Circuit Board Upgrade". Perform all appropriate steps in the rest of the manual.

Heath/Zenith Users' Group

International HUG Conference

*** IV ***

August 9, 10 and 11, 1985
O'Hare Hyatt Regency
Chicago, Illinois

It's time to plan your visit with other members of the Heath/Zenith Users' Group. This year, we are returning to the newly remodeled O'Hare Hyatt Regency, Chicago for HUGCON IV.

More exhibits, more speakers and ***more room*** *highlight the Conference activities for 1985. The Official Conference Registration form will appear in next month's issue of* ***REMark****, the official magazine for users of Heath/Zenith computer products.*

Plan to attend now! We sure hope to see you there!

Sequential Files — Part 5

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

This is the last article in our five-part series on sequential files. When added to the series on random files, which was presented from February through November 1984, you should have sufficient information to handle any data file you want. This series is being written during my Christmas to New Year's vacation, and should be submitted to HUG headquarters in that same time frame. The next series will include some neat programming details, and should be more fun as we'll cover new areas each month. We'll provide more stand-alone articles for you, too. For now, let's wrap up sequential files.

As you saw last month, if you want to do anything other than write a file, read every entry of a file in order, or add data to an existing sequential file, you must read the entire contents of your file into the RAM available in your computer. If you've got one of the 16-bit monsters with 500+K of RAM, this may not bother you, even with the huge size of their operating systems, compilers, and interpreters. But with an 8-bit system and 64K or less of RAM, a 6K operating system, and a 26K interpreter, we could run into trouble. Even though 32K sounds like a very large file, our examples have been overly simplistic. If you decide to do inventory control, or manage a mailing list with several fields to indicate membership status, date renewal is due, etc., records of 128 to 256 bytes get to be commonplace. That's four to eight records per K of memory. Suddenly, we're limited to somewhere between 125 and 250 records. Many clubs, churches, bowling leagues, etc. would run out of space before they're anywhere near complete.

What can we do? In the past it was a matter of breaking our large file into smaller files, performing the necessary operations on each of those files, and finally merging them back into one monster file. That file could then be printed or used to produce mailing labels. We could do this and solve our large file problem, but it is a cumbersome method, prone to errors, and has too many opportunities for mistakes and lost data. What can we do to handle large files? If you followed the series on random files, you already know the answer. First you convert the file to a RANDOM FILE. Then you sort it, developing a key file, and use a binary search routine. Any file which will fit on your disk can be handled this way. Changing the file is handled as with all random files, and methods for updating it were covered in the series which ran from February through November of 1984.

So why not always use random files, and never touch a sequential file again? If you were printing the data in a file, starting at the first record and progressing to the last, the sequential file could be read and printed more quickly than the random file. Not only that, but most existing files are in the sequential format. You've got to understand them if you hope to convert them to random files.

Let's look at the steps necessary to convert a sequential file to a random file, or to convert a random file to a sequential file. Then we'll write programs to do each operation. First, sequential to random.

1. Open the sequential file for reading
2. Open the random file
3. Read a record from the sequential file
4. Separate the record into fields
5. Prepare the fields for the random file
6. Write a record to the random file
7. Do steps 3 thru 6 until all data is transferred
8. Close the sequential file
9. Close the random file

To convert a random file to a sequential file, perform the following steps.

1. Open the random file
2. Open the sequential file for writing
3. Read a record from the random file
4. Separate the record into fields
5. Strip unneeded spaces from the fields
6. Prepare a record for the sequential file
7. Write a record to the sequential file
8. Do steps 3 thru 7 until all data is transferred
9. Close the random file
10. Close the sequential file

As we develop the programs to make these file conversions, I

won't go into detail on the random file operations, as they were covered in a 10-part series last year. If you don't have them, you should get the REMark back issues for February through November 1984. Actually, a set of all back issues would prove a worthwhile investment. HUG offers reprints in book form at really great prices. They may not have made the model of computer you have when REMark was first published, but programming was the same then, and those early issues were chocked full of good information.

Our conversion program for sequential to random files will read our phone number file SEQFILE.DAT and create a random file called RANFILE.DAT.

```
2 ' **** SEQRAN.BAS
4 ' **** DAVID E. WARNICK
6 ' **** COPYRIGHT 1984
1000 OPEN "I",#1,"SEQFILE.DAT"     'OPEN SEQUENTIAL FILE
1010 OPEN "R",#2,"RANFILE.DAT",43 'OPEN RANDOM FILE
1020 FIELD #2,15 AS L$,12 AS F$,1 AS M$,15 AS P$
     'DESCRIBE RECORD
1030 FOR X=1 TO 1000'SET UP A LOOP
1040 IF EOF(1) THEN X=1000:GOTO 1190 'ESCAPE ON LAST RECORD
1050 INPUT #1, R$'READ A RECORD
1060 A=INSTR(1,R$,"\")'FIND THE FIRST \
1070 B=INSTR(A+1,R$,"\")'FIND THE SECOND \
1080 C=INSTR(B+1,R$,"\")'FIND THE THIRD \
1090 D=LEN(R$)'LENGTH OF RECORD
1100 LN$=LEFT$(R$,A-1)      'SEPARATE LAST NAME
1110 NM$=MID$(R$,A+1,B-A-1)  'SEPARATE FIRST NAME
1120 MI$=MID$(R$,B+1,C-B-1)  'SEPARATE MIDDLE INITIAL
1130 PN$=RIGHT$(R$,D-C)      'SEPARATE PHONE NUMBER
1140 LSET L$=LN$'READY FOR RANDOM FILE
1150 LSET F$=NM$'READY FOR RANDOM FILE
1160 LSET M$=MI$'READY FOR RANDOM FILE
1170 LSET P$=PN$'READY FOR RANDOM FILE
1180 PUT #2'PUT RECORD INTO RANDOM FILE
1190 NEXT X'CONTINUE THROUGH THE FILE
1200 CLOSE #1'CLOSE THE SEQUENTIAL FILE
1210 CLOSE #2'CLOSE THE RANDOM FILE
1220 END'ALL DONE
```

Enter and run the program SEQRAN.BAS. You can then exit MBASIC to your operating system and see that the file RANFILE.DAT does exist on the disk. Look at the contents of the file by using the command:

```
LIST RANFILE.DAT
```

The program above will read a sequential file and write the data to a random file. To use it with sequential files other than SEQFILE.DAT, lines 1060 through 1130 would have to be changed to produce the individual fields from the complete records of that file. There may be different field delimiters other than the "/" used, and there may be some number other than four fields for each record. You'll also have to change lines 1110 and 1120 to set up the random file records for the fields they must contain. Add or delete lines in the series 1140 through 1170 for the appropriate number of records. You can now convert any existing sequential file to a random file.

With your data stored in the random file format, you can sort, change, delete, or add information just as we did in our last series. This will also give you the ability to use a binary search routine to find any record quickly, even though the file is too large to fit in the available RAM in your machine. All these functions were shown in the series on random files.

Another advantage is the ability to sort the file in several different ways (i.e. last name, zip code, etc.) and keep a small key file for that sort. If you run a mailing list for a club or a church, you'll soon find that you need a field to indicate membership status. Then you can generate lists for mailing to officers only, life members only, etc.

So, how does all this apply to sequential files? We've already pointed out that for printing purposes, the sequential file, when read from beginning to end, is more efficient than the random file. If you're only going to be doing one printing, there's not much to be gained by converting a rondom file to a sequential file. The head movement over the disk to retrieve the records in order will be made in either case. However, if you're going to read the file several times to print labels, enter data into boilerplate letters, make lists, or whatever, you'll want the file you're using to be in sequential format. Therefore, you do need the ability to convert a random file to a sequential file.

For the purpose of our demonstration program, we'll convert the random file RANFILE.DAT back to a sequential file. We'll give it a new name, NEWSEQ.DAT, so we can list it on our printer and compare it to our original file, SEQFILE.DAT. If all goes well in the conversion to a random file, and back to a sequential file format, both files should be the same. To do the final conversion, we'll follow the ten steps we outlined above.

```
2 ' **** RANSEQ.BAS
4 ' **** DAVID E. WARNICK
6 ' **** COPYRIGHT 1984
1000 OPEN "R",#1,"RANFILE.DAT",43'OPEN RANDOM FILE
1010 FIELD #1, 15 AS L$,12 AS F$, 1 AS M$,15 AS P$
     'DESCRIBE RECORD
1020 OPEN "O",#2,"NEWSEQ.DAT" 'OPEN SEQUENTIAL FILE TO WRITE
1030 'THE FOLLOWING LINES COUNT THE NUMBER OF
1040 'RECORDS IN THE RANDOM FILE AND SETS THE
1050 'COUNTER, CT, TO THAT NUMBER
1060 A%=1:C%=32767'SET MIN AND MAX RECORDS
1070 B%=INT((A%+C%)/2)'FIND MIDDLE OF RANGE
1080 GET #1,B%'GET RECORD FROM MIDDLE OF RANGE
1090 IF EOF(1) THEN C%=B%-1 ELSE A%=B%+1
     'CHECK EOF AND ADJUST RANGE
1100 IF A% < C% GOTO 1070'LAST RECORD NOT FOUND
1110 IF C%=0 THEN PRINT "FILE IS EMPTY":GOTO 5000
1120 GET #1,C%'GET THE RECORD
1130 IF ASC(L$)=0 THEN C%=C%-1:GOTO 1110
     'EMPTY RECORD, TRY AGAIN
1140 CT=C%'SET COUNTER TO LAST RECORD
1150 FOR X=1 TO CT'SET UP A LOOP
1160 GET #1,X'GET A RECORD
1170 S$=L$:GOSUB 7000:L$=S$ 'STRIP SPACES FROM LAST NAME
1180 S$=F$:GOSUB 7000:F$=S$ 'STRIP SPACES FROM FIRST NAME
1190 S$=P$:GOSUB 7000:P$=S$
     'STRIP SPACES FROM PHONE NUMBER
1200 R$=L$+"\"+F$+"\"+M$+"\"+P$
     'PREPARE SEQUENTIAL RECORD
1210 PRINT #2,R$          'WRITE RECORD TO SEQUENTIAL FILE
1220 NEXT X               'CONTINUE TO END OF FILE
5000 CLOSE #1             'CLOSE THE RANDOM FILE
5010 CLOSE #2             'CLOSE THE SEQUENTIAL FILE
5020 END
```

That's everything we'll need to convert a random file to a sequential file except a routine to strip spaces from the right end of the variables as they are taken from the random file. By stripping those spaces one character at a time from the right, any embedded spaces within the data contained in a variable are preserved. The following routine checks the right most character of the string. If it is a space, the string is reduced in size and the process is repeated. It looks like this:

```
7000 LS=LEN(S$)'SET LENGTH OF STRING
7010 IF RIGHT$(S$,1)=" " THEN LS=LS-1:S$=LEFT$(S$,LS):
     GOTO 7010 'TEST FOR SPC
7020 RETURN'OTHER THAN A SPACE WAS FOUND
```

Type and run the program RANSEQ.BAS from line 2 through line

7020. When it is finished running, return to the operating system with the SYSTEM command. Then use the LIST command to print a hard copy of the original file SEQFILE.DAT, and another hard copy of the final file, NEWFILE.DAT. Notice that they're identical. You've successfully converted a sequential file to random format, and a random file to sequential format.

If you've followed this column since February 1984, you now have all the information necessary to form your own data base. You can maintain the large master file in a random format, and smaller printable lists as sequential files. We've covered a lot of information in the past five months, but we've also unlocked the secrets of file handling.

Computer power is used more for these types of functions than for anything else. My work with commercial machines deals with files and file handling more than ten times as much as it does with number crunching.

A few final notes are in order. The first regards the selection of the reverse slash "\" as a field separator for sequential files. There are 128 different characters available in the ASCII character set. These include all the letters of the alphabet (upper and lower case), numbers, punctuation, etc. What's left for the etc. part? The first 32 characters in the set are special characters. They are called 'non-printable' characters. Each has a special meaning and they include such things as carriage return, line feed, the DC1 and DC3 characters which control your printers data flow, BEL which makes the terminal beep, and many more. Four of these are of special interest in file handling. They are:

Character	Mnemonic	Meaning
28	FS	File Separator
29	GS	Group Separator
30	RS	Record Separator
31	US	Unit Separator

These special characters may be used in your data files. To insert the file separator between fields, rather than the backslash we used, we could write

```
R$=R$+CHR$(28)
```

To test for the same character, we could write

```
A=INSTR(1,R$,CHR$(28))
```

These characters won't print. That's why I used the backslash. You'll learn more if you can see what's going on.

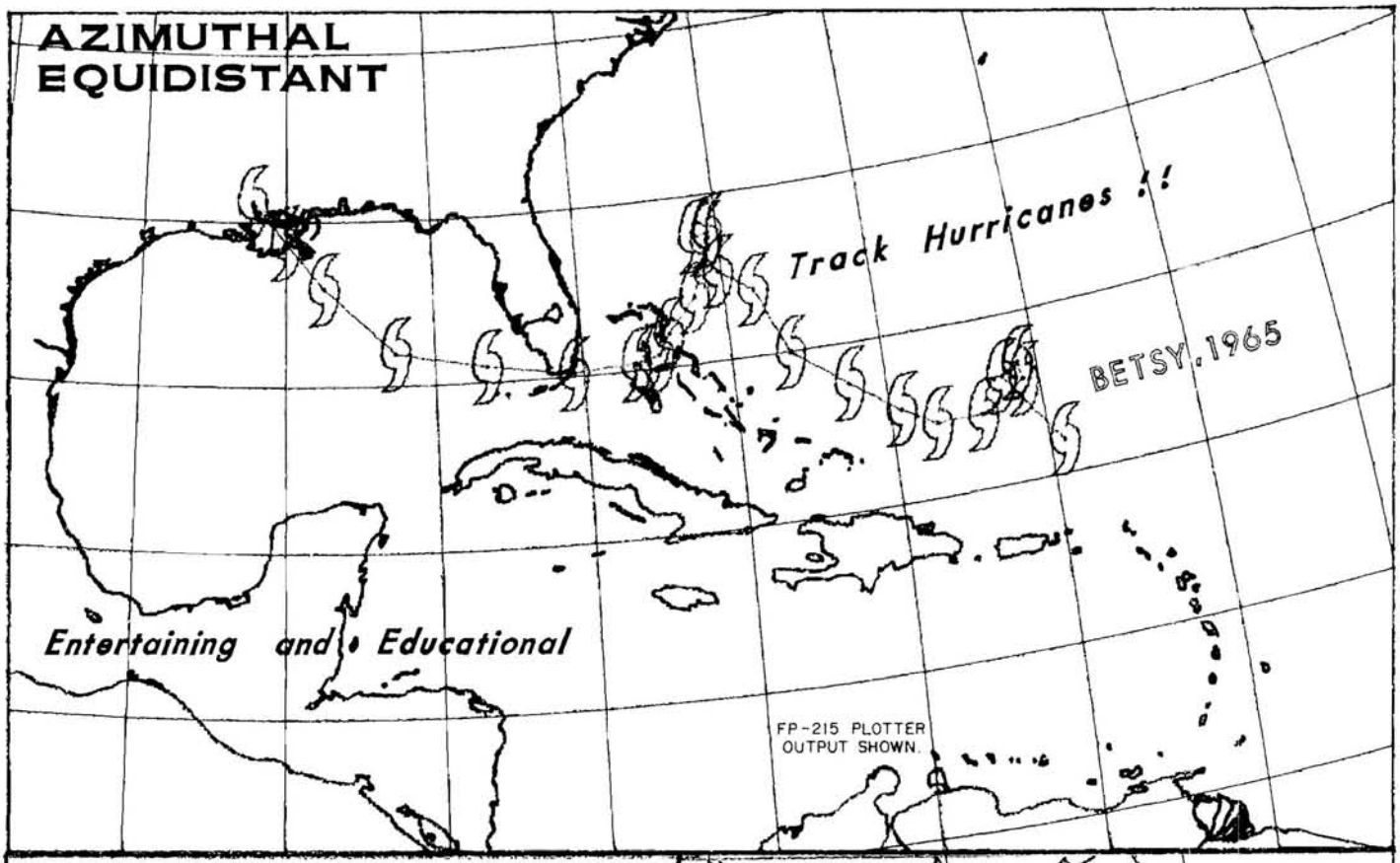
The second item deals with backup files. When we finished with a file and wanted to replace it with a temporary file, we killed it. That is, we removed it from the disk. In your applications you should consider the value of keeping a copy of the old version. Rather than KILL the old file, we could rename it as a backup with the program line:

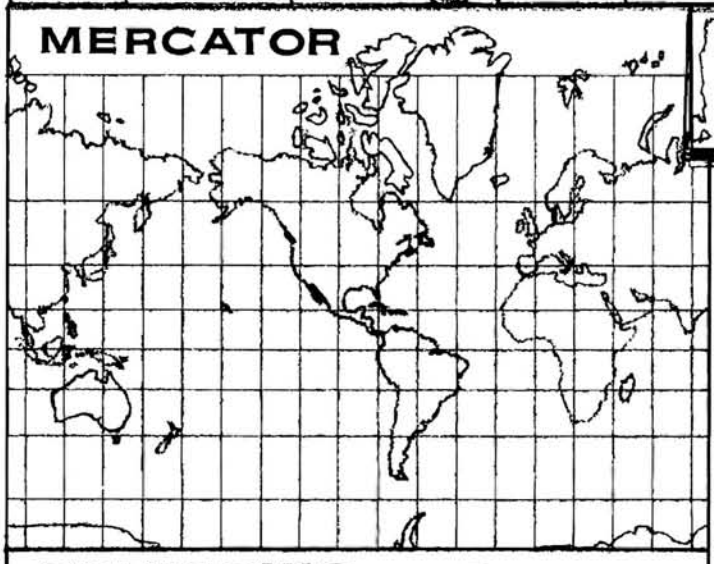
```
NAME "SEQFILE.DAT" AS "SEQFILE.BAK"
```

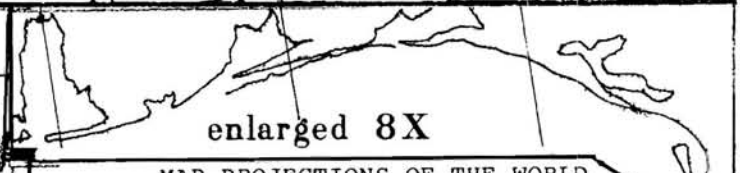
If you have a valuable data file, you should also make a duplicate of the file on a separate disk. And, DON'T STORE YOUR BACK-UP DISKS WITH YOUR WORKING DISKS. Keep them in separate rooms. It never ceases to amaze me that so many people can plunk down more than a thousand dollars on a computer system, but must save less than five bucks on a backup disk.

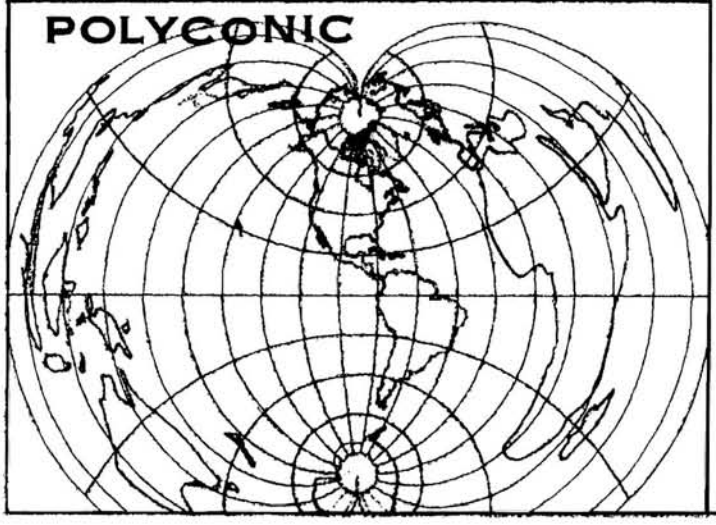
Well, that's it. In the past year and a half, we've covered files and file handling. There were menu-driven programs, frozen screens, and formatted outputs. Now it's up to you to write your own applications. Good luck with them.

See you next month. *









MAP PROJECTIONS OF THE WORLD

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

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

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

Watching Watchword Work Words And Other Goodies

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

Although I will generally resist the temptation to review editors, spreadsheets, and word processors (because of space) in this column, occasionally something comes along that does so many things well that it's not possible to ignore it. WatchWord is one of the few application software programs that is an exception my general rule. Why? Well, its price versus performance make it the best value for an all around editor/word processor that I've seen yet.

A Look At WatchWord

You'll note that I've called WatchWord an editor/word processor. It's really both, and you'll see why later...it's part of the discussion on performance. And for $100, I haven't seen anything else that provides so many features at such a reasonable price. It's relatively easy to use, and if you don't like the set up (e.g. the function keys), you can change that too. And, if it doesn't have a feature that you need, you can actually build it using the WatchWord macro facility. However, you can buy a separate Macro Utilities Package for an additional $20 which is well worth the money.

Since I suspect that most people are more interested in its word processing capabilities, we'll take a look at that first. If you're at all interested in technical things, like programming and hex files, you'll be interested to learn that WatchWord can edit hex files directly (like ATLCHAR.SYS).

WatchWord essentially provides a "what you see is what you get" type of display, which is like the ever popular WordStar. In addition, WatchWord has singularly unique features, like on-screen bold face display and underlining for the H-100. Bet you thought that the H-100 wasn't capable of that! From a strict technical perspective, the H-100 really can't do it, but of course those minor details don't really stop a clever programmer like Steve Robbins. . . the impossible just takes a little longer to program.

Word Processing Features

As I mentioned above, I consider a "what you see is what you get" type of displayed essential to word processing applications. WatchWord provides just about everything in that regard including on-screen boldface and underlining as well as strikeout, superscripts and subscripts. It sure makes it a lot easier to proofread files on the CRT, and it avoids a lot of control character displays, like WordStar, which can be confusing to a new user. And of course you can do all the standard kinds of word processing tasks such as centering, and so on. If you need to edit a spreadsheet file that's been written to disk, you can also edit lines longer than 80 characters, which is made easier with automatic horizontal scrolling based on cursor movement.

One feature that is particularly useful is the split screen mode. In addition to looking at different parts of the same file, you can also look at a different file. VERY useful for both word processing as well as programming.

Using WatchWord

I've used WatchWord for a few months now, and I continue to be impressed by its features and flexibility. I have to tell you that I have used WordStar (under CP/M-85) exclusively for writing the FlipFast books as well as this column. The only reason I mention that is because I haven't found much of anything that WatchWord can't do better (and usually easier!) than WordStar. In fact, WatchWord works so well that I'm seriously considering converting everything to MS-DOS so that I can use WatchWord. That's not an easy job, so I've been putting it off in order to finish "just one more book" before I can take the time for the conversion.

And now for a quick look at how WatchWord works words. The basic edit screen consists of twenty lines at the top with two lines for the commands and error messages underneath. The edit screen and the command/error message area are separated by a

zig-zag line which is quite effective visually. Line 24 is a solid separator line, and the 25th line is used as a status line. The status line contains useful information such as the location of the cursor (line and column), the current file name, the amount of free memory space available, a time-of-day clock, and the mode status (e.g. Insert, etc.). The keypad ENTER key is used as a toggle to change from the screen editing to the command mode.

WatchWord has all of the find and replace type features that you'll need. And if you get too carried away with deleting lines or characters, each delete function key can be "undeleted" by pressing ESC followed by the appropriate delete key. You can recover up to 100 characters or 20 lines using this technique.

Printing a file is performed with the PRINT command. And special print features, such as pause between pages, characters per inch, and lines per inch can be controlled through the use of easy to remember dot commands.

Virtually all of the H/Z-100 function keys and keypad have been used to provide one keystoke execution of commands. And if you don't like the way WatchWord is configured, you can change virtually everything from the key assignments to the screen display, and of course the printer defaults. That's done with the WWCONFIG.EXE program which includes appropriate documentation (WWCONFIG.DOC) on disk. Lots of luck in finding something more flexible than that! WatchWord even has the equivalent of an AUTOEXEC batch file (done with macros) which allows you to "initialize" WatchWord to your specific needs.

For those with a color monitor, WatchWord is configurable for any color scheme that you can imagine. Since I use a monochrome monitor (ZVM-122A), I didn't test the color configuration facility which was included as a macro.

What You Get For Your Money

Everything you need for WatchWord comes on a single disk which contains 38 files occupying about 304K. The basic WatchWord program, WW.EXE, takes 89,227 bytes according to the MS-DOS DIR command. The disk also contains 21 on-screen help files which require an average of about 1K each. Four character (CHR) files support the enhanced screen displays (superscripts, subscripts, bold, and Greek). The configuration program, and its associated documentation, plus 7 other documentation files are also on disk. Steve has also included 3 macro files for Fonts, Color, and Erase functions.

The supplied documentation consists of a 23 section manual (including an index) containing about 84 pages. Additional documentation is on the WatchWord disk which I mentioned above. As you can tell, the disk is just chock full of all sorts of useful things. As if that weren't enough, Steve has also thoughtfully included a paper strip which can be placed above the function keys on the H/Z-100 so that you don't have to remember which function key does what.

The Macro Facility

One feature that is generally not available in most word processors is the capability for the user to define a macro to perform certain repetitive tasks. For example, I enter a lot of typesetting commands in the manuscripts for the FlipFast books. One of the frequent commands is //LT, which means "left", and is somewhat like a carriage return. Since that kind of data (//LT, etc.) is very distracting while I'm writing, I use a grave accent (`) to represent that in the original files. Just before I send the disk to my publisher, I run a CP/M submit file which replaces the grave accent with the //LT command. That same kind of processing, and more, can be done with the WatchWord macro facility. Although the macro command language isn't difficult, there's an easier way.

The Macro Utilities Package

When you order the WatchWord package, you'll also get an advertisement for the WatchWord Macro Utilities Package. My advice is: Get it! It's a great buy for $20. Even if you just want to use it for learning about the power of WatchWord macros, it's worth the money. Although WatchWord doesn't have a defined mailmerge program, this package has WWMERGE.MAC which performs the same function. The package also includes macros to create a disk file catalog, and of course macros to convert WatchWord files to WordStar and vice versa. A total of 14 very useful macros are also on the disk including menu-driven printer setups for the H/Z-25, Gemini-10X, MPI, Transtar, and Prism. The disk includes a total of 35 files including both macro and various help and documentation files. This is one of those good deals that you shouldn't pass up when you order WatchWord. It's not only highly recommended, it's mandatory as far as I'm concerned.

WatchWord For Programmers

At the beginning of the article, I mentioned that I considered this as a word processor/editor. I've done some programming in my time, and I find that CP/M's DDT and MS-DOS's DEBUG are sometimes a real pain to use when I want to look at a file in hex. WatchWord allows you to directly edit hex files, in addition to doing all of the normal types of programming tasks. I've already mentioned the split-screen capability, and we just talked about the macro processing. If you're looking for a word processor and a programming editor, there's no question in my mind that WatchWord is the way to go. That's particularly true if you're budget minded (aren't we all!) and can't afford one program for word processing and another for programming.

A Final (Watch) Word

WatchWord is one of those rare programs that's easy enough for beginners to use and yet sophisticated enough for just about any advanced application. WatchWord is so flexible that it's difficult to imagine any type of word processing or editing that you can't do with it. Offhand, I can't think of anything that I've done or want to do that WatchWord can't handle. That includes everything from writing letters, books, and magazine articles to programming in various languages like assembler and C. During my testing process, I didn't find any "undocumented features" (better known as bugs), and WatchWord's error trapping techniques appear to be more than adequate, even for beginners. The only "bad" thing about WatchWord is that, as far as I know, it's only available for the H/Z-100's, not the H/Z-150's or the IBM PC, but I wouldn't be surprised to see a version which runs on the IBM and compatibles. Be sure to let Steve know if you're interested in an H/Z-150/IBM PC version. WatchWord, and the companion Macro Utilities Package, provide an extraordinary amount of capability and flexibility for a small price. Both are highly recommended.

On The Other Side

One of the reasons for the intermittent columns has been extensive traveling to New York which has been required as a part of my full time job as a computer consultant. One of the primary parts of this particular assignment was to develop a significant

amount of documentation related to computer security, one of my specialties. In the course of that assignment, I used a fully equipped IBM PC with the MultiMate word processor (version 3.22) to write several hundred pages of documentation. Although I know that MultiMate is available in the latest Heath catalog, the following comments are NOT an evaluation of the Heath version.

Even though MultiMate is relatively easy to use, I consider any software that has, what I call a catastrophic bug, to be unacceptable for my personal use. In this respect, I define a catastrophic bug to be anything which causes a loss of data on the system. One of the features that I use consistently on any word processor is the "word forward" and "word backward" cursor movement commands. In WordStar, the commands are CTRL F and CTRL A respectively. MultiMate uses CTRL right-arrow and CTRL left-arrow. Not a bad choice of commands. However, under some circumstances, use of those commands in MultiMate would result in a complete system freeze to the point that no commands would be accepted. And the only option was to reboot the system with the resultant loss of data. I never could duplicate the exact circumstances since it always happened when I was concentrating on writing instead of trying to find bugs.

I also found MultiMate to be very slow. It's a page oriented word processor which doesn't have anything like the speed I'm used to on my H-100 with any word processor. I couldn't tell whether that was due to the IBM PC and the floppy disk drives or a result of the software itself. Although it's fast while you're working within a single page, a page forward or backward command seems to take forever. The spell checker is the slowest I've ever seen — about a minute for a full page of text —, and I found it awkward to use.

One of the reasons that I didn't buy WordStar until a couple of years ago was the same kind of thing. The catastrophic and infamous "Disk Full" error was finally fixed. I'm currently using WordStar 3.30 on CP/M-85 and the H/Z version 3.3 on MS-DOS. After two years of hard use, I've never lost any data due to a program bug. In fact, the only time that I've lost data was due to a complete power loss in the house, and I can't blame the software or hardware for that. I've since cured that problem with an uninterruptible power supply (UPS) which I'll talk about in a future column.

And Still More On The Hard Disk...

My thanks to all of you who have written letters about the implementation of a hard disk on the H/Z-100 which appeared in the March issue of REMark. In fact, I received a rather REMarkable number of letters about that particular column. I've even been accused of providing incorrect information about the use of FORMAT and CHKDSK to determine the number of bad sectors on a hard disk. As it turns out, though, I don't agree that the information was incorrect — however, I will agree that it may have been misleading simply because I didn't go into any details. It was interesting to note that the one letter that accused me of incorrect information also admitted that he didn't know how the bad sector mapping was actually done, and it appeared he that also misinterpreted the information on pages 3.3-3.4 of the MS-DOS Version 2 Programmer's Utility Pack (PUP).

A Look At File Allocation

Before we go into the details of the hard disk, it's important to understand a few details of file allocation in CP/M and MS-DOS. Most operating systems, including mainframes, allocate disk space in terms of blocks or clusters — generally because it's more efficient. Mainframe operating systems, like MVS, also can allocate disk space in terms of tracks or cylinders, however, block allocation is generally recommended for efficiency reasons. Before we get too far, I'll take a moment to define some terms. All disks are "divided" into concentric circles called tracks. A single concentric circle is called a track. The 5.25" floppy disks, used in most micro systems, typically contain 40 tracks regardless of whether they're single- or double-sided.

Each track is further subdivided into a unit called a sector. The standard 5.25" floppy disk typically contains 8 or 9 sectors per track which depends on the operating system. For purposes of this discussion, we'll assume that each sector contains 512 bytes since that seems to be the standard for most micro systems. We can then compute the storage capacity of our disks as follows:

(bytes/sector) × (sectors/track) × (tracks/side) × (number of sides)

For a standard double—sided, 8 sector disk, the values are:

(512) × (8) × (40) × (2) = 327,680 bytes

For a standard double-sided, 9 sector disk, the values are:

(512) × (9) × (40) × (2) = 368,640 bytes

But if those numbers are correct, then why do we always talk about floppy disk capacities of 320K or 360K? The answer is because the STANDARD definition of a kilobyte (for disk capacity and memory) is 1024 bytes which is 2 to the tenth power. If you divide the above numbers by 1024, you will find that you'll have exactly 320K and 360K respectively. Similarly, a system will actually have 262,144 bytes of additional memory if you add a 256K (256 × 1024) memory card. You can prove these floppy disk numbers to yourself by using the MS-DOS FORMAT command without the /S (system) option. Defaults for the H/Z-100 and the H/Z-150 (and IBM PC) are different since they format to 8 (320K) and 9 (360K) sectors respectively.

Now that we've looked at some common definitions for floppy disks, you should know one more which is used primarily with hard disks for both micros and mainframes. The term cylinder is used to describe a VERTICAL series of tracks. If you imagine a double-sided floppy disk with Track 0 as the first track, the surface described by the top and bottom track is a cylinder. That floppy disk contains 40 cylinders or 80 tracks. An understanding of this concept is important since it explains why PC-DOS version 1 could not support a hard disk. Version 1 of PC-DOS used track mapping, instead of cylinder mapping when the disk was formatted. Track mapping simply means that the top side of the disk was formatted, tracks 0 to 39, and then the bottom side of the disk was formatted with tracks 40 to 79. All versions of H/Z MS-DOS and Z-DOS have consistently used cylinder mapping which is why hard disk support has always been available. This is just one of those obscure technical details that shows how Zenith develops consistently good operating system software.

Blocks And Clusters

Most microcomputer operating systems allocate file space using block type methods. CP/M allocates file space in units called "blocks" and MS-DOS (including PC-DOS) allocates file space in units called "clusters". A block or cluster contains a set number of sectors which is directly related to the disk type, number of sides, density, and various other factors. It just turns out, not by accident, that a sector usually contains 512 bytes for 5.25" soft-sectored, double-density disks and hard disks regardless of oper-

Table 1

CP/M-85 5.25" Disk Formats

Sectors/ Track	Tracks	Sides	Sectors/ Block	Minimum File Allocation Size
8	40	1	2	1024 (1K)
8	40	2	4	2048 (2K)

ating system (either CP/M or MS-DOS/PC-DOS).

For this discussion, I'll use a block to indicate a cluster as it applies to MS-DOS, as well as PC-DOS. CP/M-85 and MS-DOS allocate file space in terms of blocks. Table 1 shows the number of sectors per block for CP/M-85. Table 2 Shows the number of sectors per block (cluster) for floppy disks in MS-DOS. Table 3 shows the number of sectors per block for a hard disk which is dependent on the PARTITION size, not the overall size of the hard disk. The information contained in Tables 2 and 3 was extracted from

Table 2

MS-DOS/PC-DOS 5.25" Disk Formats

Sectors/ Track	Tracks	Sides	Sectors/ Cluster	Minimum File Allocation Size
8	40	1	1	512
8	40	2	2	1024
9	40	1	1	512
9	40	2	2	1024

Tables 3.1 and 3.3 in the Programmer's Utility Pack. I developed similar information for Table 1 using the STAT command.

By now you're probably wondering what blocks and clusters have to do with a hard disk. I mentioned that the bad sector table was created with the absolute disk address of each bad sector. Unfortunately, both CP/M-85 and MS-DOS cannot simply "lock out" a bad sector. Both must do the bad sector lock out based on their minimum file allocation unit. Tables 1 through 3 will show

Table 3

MS-DOS Hard Disk (Winchester) Partition Format

Partition Size (MB)	Sectors/ Track	Sectors/ Cluster	Minimum File Allocation Size
0-8	18	4	2048 (2K)
8-16	18	8	4096 (4k)
16-32	18	16	8192 (8K)
32-64	18	32	16384 (16K)

you the minimum disk space for a file on various disk types. In particular, Table 1 shows why small files that contain less than 1 kilobyte will take up less actual space on a single-sided disk than a double-sided disk. It's only because the minimum block size is smaller on a single-sided disk.

I've heard a number of people perpetuate the myth that MS-DOS is incredibly more efficient with disk space because they can see that a file only "takes" 33 bytes (or whatever) of disk space as shown by the DIR command. That's a myth since MS-DOS allocates a cluster to a file, and the minimum cluster size for that (see Table 2) is 512 bytes. It just happens that the MS-DOS directory contains the true file size as displayed by DIR, however, the actual disk space used by the file is still one cluster. If you look at Table 3 for a 32-64 megabyte hard disk partition, you'll see that this same 33 byte file would still take 1 cluster, but that amounts to 16K of actual disk space used for that small file. You should also note that, if you have a lot of small files on a hard disk, you should probably keep that partition size under 8 megabytes since each file takes the minimum cluster size available. That will help to maximize the efficient use of disk space. Now that we've taken a look at file allocation, let's get back to the details of setting up a hard disk.

It Begins With PREP

As I mentioned in a previous column, all hard disks are set up with the PREP command. PREP has three basic functions as follows:

- Format the entire hard disk
- Check out the media with read/write passes
- Set up the Software Boot Code (SBC) entries
- Set up the Superblocks defining the partitions
- Set up the bad sector tables

The first step, formatting the hard disk, performs essentially the same kind of function as the normal MS-DOS FORMAT program which we'll discuss later on in this article. Then PREP performs a series of exhaustive media tests (the most comprehensive in the industry) which basically involve a read/write operation to every sector on the disk. The read/write testing is performed in multiple passes on the hard disk so that every sector is checked out numerous times. During this testing process, PREP builds the bad sector table in memory as the testing is performed. In addition, PREP also keeps the hard disk parameters in memory based on your responses to the initial prompts such as the number of cylinders, number of heads, etc. Hard disk parameters and the beginning sector addresses for the Superblocks and Bad Sector Tables are written by PREP in a reserved area called the Software Boot Code (SBC).

During the last stage of PREP execution, the program writes out the Software Boot Code (SBC), the 2 Superblocks containing the partition information, and the 2 copies of the bad sector table. The reason that 2 copies are written is simply to provide a backup copy in case one of the sectors in the Superblock goes bad. By the way, the Superblocks are referred to as Superblock A and B. You should know that PREP reserves 36 sectors on the hard disk for this information for the SBC, the Superblocks, and the Bad Sector Tables.

The PART Command

After the media checkout and setup have been completed with PREP, the next step is to define the partitions on the hard disk. When you first execute the PART command, you will find that PREP has already defined two partitions for you...one for CP/M and one for MS-DOS. PREP is set up so that each partition uses 50% of the available disk space. PART allows you to change the size and number of those partitions. PART has two primary functions:

- Defining the name and size of the partitions
- Defining the default boot partition

When you have completed the definition of the name and size of each partition, and defined the default boot partition, the PART program writes, among other things, the name and starting sector address for each partition in each Superblock. The default partition boot string is written to the first part of the 5 sector SBC.

When you change the size of a partition with the PART command, you will note that you should have previously backed up all partitions on your hard disk. That's because PART changes the beginning address when you change the partition size, and that new address may not include some of the existing files. When in doubt, I follow instructions which means that I adhere to the Zenith recommendation that the hard disk be backed up before using the PART command.

The ASSIGN Program

Both CP/M and MS-DOS have ASSIGN programs which are basically used to define which partitions are to be used by the current operating system. Although their command syntax is different, they perform the same function.

The easiest way to understand the ASSIGN command is simply to think of the hard disk as multiple floppy disks. For example, CP/M-85 allows you to assign a maximum of 2 active partitions at any one time. MS-DOS and Concurrent DOS allow you to assign up to four partitions at any one time. It's interesting to note that, in CP/M-85, the boot drive is always drive A. That applies no matter whether the boot drive is a floppy or a hard disk.

Under MS-DOS and Concurrent DOS however, partition assignments are always made to drive E, F, G or H. That's kind of helpful since you don't have to remember which drive you booted from.

Regardless of which operating system you use to set up your hard disk, you must initially boot from a floppy disk and ASSIGN the partitions to be formatted.

For various reasons, the ASSIGN program notes the drive identifier and the beginning sector of each valid partition in memory. As a side note, the MAP command simply changes the drive identifier in memory so that the beginning sector of a partition is associated with a new drive name. MAP does not, of course, actually change any of the actual drive designators on the hardware which is the reason that it won't work with any program that accesses the disk drive directly. I won't go into a lot of detail on that since the restrictions on the MAP command are listed in my MS-DOS (Z-DOS) FlipFast book.

The MS-DOS FORMAT Program

Now you're ready to format the hard disk. I mentioned in the article on the hard disk implementation about using the MS-DOS FORMAT program to determine the actual number of bad sectors on the hard disk. Now I'll describe how that works.

FORMAT does a few different things when it formats a hard disk. Since it "knows" that it's formatting a hard disk (i.e. E, F, G or H drives), the program will prompt showing that it will "Format the partition assigned to drive d". Before FORMAT actually does the formatting, the program obtains the beginning sector address before it begins execution.

The FORMAT program also checks the Bad Sector Table for all of the bad sectors contained in that partition. When the program has completed, the display showing the number of bytes in bad sectors IS THE ACTUAL NUMBER OF BYTES CONTAINED IN THE BAD SECTORS FOR THAT PARTITION. As I previously mentioned, you can divide that number by 512 to determine the number of bad sectors. The FORMAT program report is based on the number of bad sector table entries for that partition. Now that you know the exact number of bad sectors, I suggest that you run the CHKDSK program to determine the exact amount of usable space in that partition. When you look at the CHKDSK report on the number of bytes in bad sectors, you will usually find that it's a different number than reported by FORMAT. Why?

The CHKDSK Mystery Solved!

The CHKDSK program also displays the number of "bytes in bad sectors" report, however, it does not get the information from the Bad Sector Table like FORMAT does. CHKDSK reads the File Allocation Table (FAT) and reports on the number of bytes contained in the CLUSTERS which contain the bad sectors. It's obvious that the CHKDSK report, in order to be technically correct, should really read "bytes in bad clusters". Refer again to Table 3, and let's assume the partition size is less than 8 megabytes. Let's also assume that FORMAT reported 1024 bytes in bad sectors. Now we know that we have 2 bad sectors in the partition. It's possible, though unlikely, that both sectors will be in the same cluster, so we'll assume that they're in different clusters. We know that a cluster contains 4 sectors (2048 bytes) from Table 3. Since we know that CHKDSK reads the FAT and finds an FF7 hex which indicates a bad CLUSTER, we should expect that CHKDSK will report 4096 bytes in bad sectors since we have 2 bad clusters (2048 bytes each) containing one bad sector each. Remember that the smallest allocation unit is a cluster. If the partition size is greater than 32 megabytes, Table 3 tells us that a cluster is 16,384 bytes. And then CHKDSK would accurately report that we've got 32,768 bytes in bad sectors because of the 2 bad sectors.

In summary, you can calculate the number of bad sectors on a disk using the information provided by either FORMAT or CHKDSK. Since FORMAT gives the exact number of bytes in bad sectors, just divide by 512. CHKDSK, on the other hand, actually reports the number of bytes in bad CLUSTERS, so you have to divide that number by the number of bytes per cluster. As long as the bad sectors are in different clusters, the calculated results from either FORMAT or CHKDSK should be the same.

The CP/M FORMAT Programs

It's a fact of life that CP/M doesn't have an inherent way to lock out bad sectors, so the clever people at ZDS were forced to come up with another way to identify the bad sectors on a partition. The CP/M FORMAT program also reads the bad sector table, but it creates a hidden, system file under a user number higher than 15 (I found it in user number 31) which identifies the blocks containing the bad sectors. In fact, the name of the file is BADBLOCK.SYS. Since that block has been assigned a valid file name, you don't have to worry about that bad sector.

Where To Find Information

By now you're probably wondering where I got all of the information about how all of these programs work. Most of it, and more, is contained in the documentation that comes with each H/Z operating system including CP/M-85, Z-DOS, MS-DOS, CP/M-86, and so on. Look under the explanations of the PREP and PART commands to find additional details. The rest of the information, like the difference between the FORMAT and CHKDSK "bytes in bad sectors" reports, was due to extensive testing of these commands for the FlipFast books.

Support For Your System

Occasionally I see or hear of people who haven't received what they thought was adequate support from Heath. It's unfortunate that we only seem to hear of complaints rather than some of the excellent support that Heath provides. Most of you who read this

column are aware that I had a problem with a 26 megabyte hard disk drive that failed. Although I bought the drive at the HUG Convention Heathkit store last year, arrangements were made to have the problem corrected at the local Fort Worth Heathkit store. The drive was replaced in a timely manner, and I particularly appreciate the excellent support that I've received there.

Even though you should look to your Heathkit store for support, they are only as good as their suppliers. I've mentioned that I have a "non-standard" 26 megabyte (23 MB formatted) in my system. Part of the excellent support that I received was due to the supplier of that disk drive — Anthem Systems in Scotts Valley, California. Since I hear that some store managers read this column, I would recommend that you consider Anthem Systems if you need a source for high capacity hard disks. By the way, Anthem Systems is a distributor and does not sell to end users, however, I thought I would report my personal experience with their good support to my local Heathkit store.

Software Registration

One of the more interesting discussions at last year's HUG Convention concerned a couple of complaints that some users were not receiving updates (or notification) to software that was purchased from Heath and Zenith. As a result of my move to Texas, I found that I received updates to MS-DOS and Z-DOS by way of my old California address. I've always been meticulous about sending in the license agreements, and I've never had a problem receiving updates from Zenith. So how come some people are having problems?

The most common reason that you may not receive an update is that you simply didn't send in the license agreement. If you haven't sent in the completed license form, don't expect to receive any updates. I was also told that some forms are received without the name and address information. And for some of the forms that are filled in, the name and address is not legible, so it's impossible to determine the owner of the software.

Your Part In The Software Registration Process

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

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

Since my writing tends to be illegible, I always PRINT my name and address on both license forms along with the other required information. Regardless of who you are — individual, business or military —, be SURE that you complete the name and address information legibly. After I've completed both forms, I take one of the labels that I previously removed from the box and tape it to my copy (the yellow one) of the license agreement. That's essentially an exact copy of the one on the Zenith copy (white) of the agreement which contains the name, model and serial numbers of the software. I've found that it's faster, and more accurate, to use the extra labels than to try to copy the information by hand.

The white copy (and only the white one) gets mailed to Zenith in the supplied envelope, and the yellow copy goes in my "License Agreements" file for future reference. But what can you do if you have a change of address like I did?

Software Registration — Change Of Address

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

There will inevitably be some people who think that this is too much work. Their argument will go along the ines that they should only have to send in a change of address card, and Zenith should do the rest of the work. Although there is some merit to that argument, you should already have a complete inventory of your hardware and software, complete with vendor, model and serial numbers. If you haven't got a COMPLETE hardware and software inventory, lots of luck if you ever have to file an insurance claim...let alone the idea of getting an insurance policy to cover your computing assets in the first place. How much insurance should you get? As a quick test, write down everything you can think of that either is part of or supports your computer system. Did you remember to include all of those connecting cables for printers? What about those additional memory boards in your system?

It occurs to me that I should share some details about how I developed my software and hardware "data base" in a future column. Although I set it up with WordStar, you can effectively do the same thing with just about any word processor or spreadsheet. I've also found that my data base is helpful at tax time since my computer system is used for business purposes. And of course, I found it very easy to provide Zenith Software Registration, as well as other vendors, with the information required for my change of address.

An S-100 Expansion Unit?

I thought it would never happen, but I now have all of the S-100 slots filled in my H-100. Does anyone know of an S-100 expansion unit for the H-100? Z-205 memory expansion boards, floppy and hard disk controllers, and a Z-204 Multiport board have all conspired to fill up all of the card slots.

For Hardware And Software Vendors

If you have something of interest to the HUG community, let me know about it. I will acknowledge all software and hardware as it's received, and let you know about the possibilities of reviewing it in this column.

For example, I recently received a review copy of WordKey which essentially replaces all of WordStar's control key commands in MS-DOS. Based on the manual, it looks like a nice piece of programming although I haven't had much of a chance to work with it yet.

The 1985 HUG Convention

Next month is the HUG Convention, and I'll be speaking there again this year. Based on a lot of interest in the MS-DOS

(Z-DOS) FlipFast book, I'll be giving several sessions on "How to use MS-DOS Directory Trees, Command Pathing, and I/O Redirection". Although the final name of the presentation may change, that will be the subject of the discussion. We'll talk about the subdirectories as they're set up on the distribution disks as well as ideas on how to set up your own subdirectories. Effective use of command pathing plus some interesting ways to use I/O redirection will also discussed. For those of you who aren't able to attend, I'll review the highlights of my presentation in a future column. Hope to see you there!

The FlipFast Series (To Be Continued)

Due to the large volume of requests for the Volume 2 FlipFast Guide which contains MS-DOS programming information (like interrupts, system calls, FCB formats, etc.), I've taken a leave of absence from my full time job in order to complete that book. In addition to that information, Volume 2 will also contain the FlipFast Programming Command Reference which includes DEBUG, MASM, LINK, LIB, CREF, EXE2BIN, and so on. Another FlipFast book, currently in progress, is for IBM PC-DOS which includes version 3.0. I'll make no comment on when these will be available since I learned my lesson the last time. All I'll say for now is that they're "in progress", but they should be completed Real Soon Now!

Next Month

I still have a lot of work to do in the review of software. Since most, if not all, of it will be available at the HUG Convention, it seems appropriate to review some of it next month. Although special attention will be focused on programs available from Software Toolworks, we'll also look at products from HUG, Newline Software, Studio Computers (Pro-Driver), Mycroft Labs (Mite and Compat), and Hilgraeve's Access communications program. We'll look at these programs with the idea that you can actually see most of them demonstrated at the HUG Convention.

Products Reviewed

FlipFast Command Guides

CP/M-80/85	$ 12.95
MS-DOS(Z-DOS)	$ 19.95
MS-DOS (Volume 2) Not Available Yet	
IBM PC-DOS Not Available Yet	

Heathkit Stores
S-A Design Books
515 W. Lambert, Bldg. E
Brea, CA 92621-3991
(714) 529-7999

WatchWord	$100.00

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

WatchWord Macro Utilities	$ 20.00

EASYware
Small Computer Systems Software
7202 Faro's Court
San Antonio, TX 78233
(512) 657-7109

WordKey	Unknown

DelSoft
564 Calle Anzuelo
Santa Barbara, CA 93111
(805) 967-9566

MS-DOS Version 2 Programmers Utility Pack (CB-5063-16)	$150.00
Z-204 Multiport Board (H/Z-100 only)	$395.00
Z-205 256K RAM Board (H/Z-100 only)	$799.00
HVM-122A amber monitor (kit)	$ 89.95
ZVM-122A amber monitor (assembled)	$139.95

Heathkit Stores
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(616) 982-3571

Heath/Zenith Data Systems Corporation
Software Registration
P.O. Box 1000
St. Joseph, MI 49085

Anthem Systems
c/o Marshall Post
6001 Butler Lane
Scotts Valley, CA 95066

*

A Super Intelligent, But Inexpensive, Peripheral For Heath Computers

Timothy D. Stanley
12801 Roma Ave NE
Albuquerque, NM 87123

One feature missing from some of the older but still otherwise very useful computers like the H-8 and H-89 is high resolution graphics. Several graphics boards have been made available, but they are quite expensive and have very limited software support. I have implemented what I have found to be an exceptionally cost effective solution to this problem that I believe other Heath Users would also find very acceptable. This solution is the interface of an inexpensive Atari computer with the Heath computer via the serial buss at 19.2 K baud. With this arrangement, the Atari computer can be used for graphics and games (which is what it does best) while the Heath computer is available for word processing, telecommunications, and science (which is what it does best). Another advantage is that they can be done simultaneously, ie. a program can be loaded into the Atari and then used on the Atari while the Heath computer is running another program. With this arrangement, only the computer for the Atari system is required; the disk drives, printer, and modem for the Heath system can be used by the Atari.

Why an Atari anyway? Let's take a look at what you get with an Atari computer. Last summer Atari 400 computers were available for $39.99. That $40 buys a computer with 320 by 192 in color graphics, 4 channel sound, 8 analog inputs, 16 parallel I/O lines (used primarily for Joy Sticks), a light pen interface, RF modulation, and hundreds of available software packages. The only problem with making up a complete Atari system is the expense of the peripherals like $400 for an Atari Disk drive (10 times the price of the computer). The solution to that problem for me was to interface the Atari computer to the Heath computer so that the Atari could use the Heath disk drives and printer.

The only remaining problem was to come up with the software to enable the Heath computer to emulate Atari drives and printer. The solution here was to purchase a software package called the "Atari Connection" from a company called USS Enterprises in San Jose, Ca. This software package allows a large (90k) CPM file to be formated as an ATARI disk with the ATARI directory and disk operating system imbedded within the CPM file. This software package also allows the Atari to use the Heath system as a huge printer buffer and also allows the Heath keyboard to be used rather than the Atari computer keyboard if desired.

The physical connection between the two computers consists of connecting the serial data out on one of the Heath computer RS232 ports to the serial data in on the Atari computer and connecting serial data in on the Heath computer to serial data out on the Atari. Naturally the signal ground must also be connected. Another slight interfacing problem also arises since the Atari uses 0 to 5 volt logic levels on it's serial buss rather than RS232 levels. The solution to this problem for me was to just remove the two RS232 driver circuits from one port of my serial interface board and replace them with jumpers on DIP headers.

Since completing this interface I have copied forty disks of public domain software from our local Atari users group using a borrowed Atari drive onto Heath hard sector format disks. I would be willing to make copies of these disks for a nominal charge to cover the cost of disks and postage for HUG members who are interested.

The bottom line is that for a price considerably below the price of a new system or even a graphics board you can have outstanding graphics and sound capabilities with your H-8 or H-89 computer.

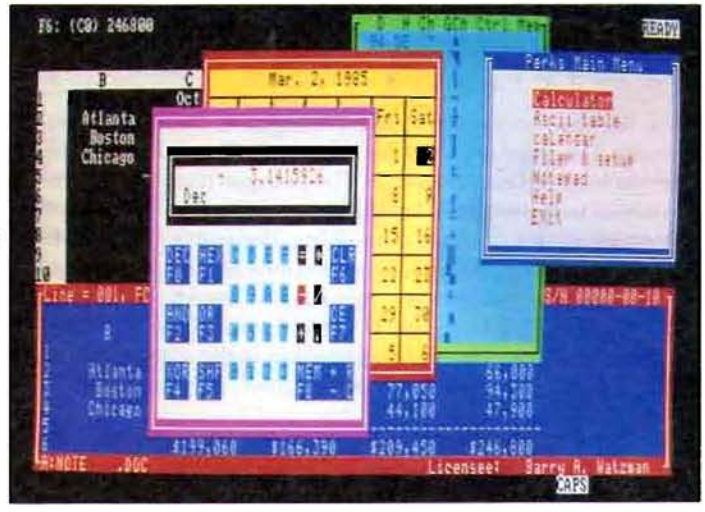
*

About The Author

Timothy D. Stanley *is an Air Force Major currently running the radiation effects on the electronics section of the Air Force Weapons Laboratory at Kirtland Air Force Base in Albuquerque, New Mexico. He has a BS degree in Physics from BYU, an MS in Economics from SDSU, an MS in Nuclear Engineering from AFIT, and has nearly completed a PhD in Electrical Engineering at UNM. He expects to have this degree completed by May 1985. His hobbies include computers, electronics, and music, and he is also active in Boy Scouting and church.*

Perk Up Your Z-100

A Review Of Barry Watzman's Perks (tm) Desktop Utility



Pat Swayne
HUG Software Engineer

If you are an H/Z-100 user who reads computer magazines aimed at IBM PC users, you are probably aware of a program called Sidekick (tm) and other programs like it that run on the PC and the "clone" machines, including the Z-150 series. A similar program, the Perks Desktop Utility, is now available for H/Z-100 series computers. In this review of Perks, I will make some comparisons to the Sidekick program, which is considered the standard by which to judge such programs.

Perks Features

Perks is a collection of 5 utility programs, or functions, integrated into one background program. It is called a background program because it loads itself into memory and remains there while you run other programs. At any time, while you are running a program or at the system prompt, you can call up Perks by typing a special key combination (Shift-Break), and use one or more of its functions. You can then return to your program or the system. The "windows" used by each function can be positioned anywhere you want them on the screen, and the positioning saved, so that they will be in your positions the next time you use your computer. You can turn each window on individually, or you can have any or all of them on the screen at once, even though only one can be active at a time.

The functions provided in Perks are a calculator, a calendar with appointment calendar, a note pad editor, a file manager, and an ASCII table. They can be selected from a main menu by pressing the first letter (or capitalized letter) in the function name, or by moving an indicator with the arrow keys, and then pressing RETURN or ENTER. You can also jump directly from one function to another, bypassing the menu.

The Perks Functions

The calculator is a four-function type that works in decimal or hexadecimal. In the decimal mode, it is a floating point calculator with 9 digit accuracy. By comparison, the Sidekick calculator uses a fixed decimal point, with four places after it. The Perks calculator is therefore more accurate when the results of calculations are small fractions.

The calculator will sometimes display a result using 10 digits, if the internal register has not overflowed. In the hex mode, it works with integers only, in the range 0 to FFFFFFFF. In this mode, the functions AND, OR, XOR are available, along with the capability to shift a number left or right by a specified amount. Numbers above 9 in the hex mode are entered using the letters A-F on the main keyboard. On the Sidekick calculator, you must enter A-F using the function keys F5 through F10, because the letters A, B, etc., are used to select functions and modes. The Perks calculator uses function keys for most functions and modes. The Sidekick calculator has an additional number mode, binary, and in the hex mode it can handle numbers up to FFFFFFFFFFFF, but it lacks the shift functions.

The calculator has the ability to export the result of a calculation into your foreground program. To use it, you press the P key, followed by any key, which becomes the Program key. Then you exit Perks and return to your program. When you press your selected Program key again, the result is entered into your application at the current cursor location as if you had typed it in yourself.

When you select the Perks calendar, a calendar display for the current month appears, with the current day highlighted and the current year at the top of the display. The day, month, and year displayed represent the system date in effect at the time Perks was loaded. By using the arrow keypad keys or other keys, you can easily change the day, month, or year on the display. If you change the system date after Perks is loaded, the Perks date does not change. With Sidekick, the date does change.

While you are in the calendar, you can press the L key, and Perks will switch to the appointment calendar, for the date on the monthly calendar. The appointment calendar is like a little notepad with the time listed in half hour increments down the left

side, from 7:00 am to 7:30 pm in two pages. The Sidekick appointment calendar is similar, but lists the time from 8:00 am to 8:30 pm. I guess Barry Watzman gets up earlier than the folks at Borland International (the producers of Sidekick).

You can enter short items beside any of the times on the appointment calendar, and save information on a disk. By changing the date on the monthly calendar, you can record appointments for future dates, and then all you have to do is look at the appointment calendar each day when you turn on your computer, to see if you have an appointment for that day.

The notepad editor works like WordStar in the non-document mode. The window used by the notepad can be adjusted in height from 10 lines to 16 lines. The notepad has a fixed 4k buffer in which to store text. In Sidekick, the buffer size is adjustable by using a separate configuration utility. The editor uses WordStar-style control keys for cursor and text movement and other functions, and in addition, the arrow keys will move the cursor, and the INS LINE, DEL LINE, and D CHAR keys will perform their described function. There is no provision in the current release to redefine the function of the editing keys, as in Sidekick.

The notepad can import text left by the foreground program from the screen into its buffer. In the import mode, the notepad window and any other Perks windows are removed from the screen. You can then mark the portion of the screen to imported, hit the ESC key, and the marked portion will be copied to the buffer at the current cursor position in the buffer, and the notepad window (but not any other [inactive] windows) will be restored. The text to be imported is marked by corners, so that text in any quadrant of the screen can be marked for importation, excluding all other text on the screen. You can, for example, import the lines from a BASIC program without the line numbers. The importation feature works with just about every program with one notable exception. It will not work with the Heath/Zenith version of WordStar 3.3, if the text and background highlighting colors are both white. However, if you change either color (with INSTALL), text can then be imported. (You may also notice that WordStar scrolls a bit slower if you change colors.) If you have patched your WordStar as described in recent REMark articles, the import feature will work regardless of color. Imported text can be saved in a file, as can text entered in any other way, and you can also read files directly from a disk.

The file manager contains the setup parameters for Perks, similar to the Setup section of Sidekick, but it also contains features not found in Sidekick. These features are the abilities to change the default drive and directory, list the directory, erase files, to set up a screen saver, and to set some timers. The abilities to change the default drive and directory are powerful and potentially dangerous features that can be used to extend the usefulness of pre-MS-DOS version 2 programs. For example, if you have a program that cannot directly access other directories, you can start the program, enter Perks, change the directory, and then continue with the program.

The 8 timers in the file manager can each be set to a future time and date, and when that time and date are reached, the computer will beep once a second for one minute or until you press a key. The timers will work whether you are in Perks or a foreground program. In order for them to work properly, the date and time must have been set before Perks was loaded.

The screen saver, when enabled, will blank the screen after a user selected period of screen and/or keyboard inactivity. It allows you to leave your computer on all day without fear of "burning" the CRT screen.

The ASCII table function presents a list of all ASCII codes, 0 through 127. The table displays 16 lines of codes at a time, and you can scroll through the table using the arrow keys or the WordStar-type text movement keys (Control-C, Control-R, etc.). The decimal and hex value of each code is shown along with the character representing the code. For control codes, the mnemonic of the code is shown (ETX for Control-C, etc.), and for the codes that show up as graphic characters when the computer is in the H19 graphic mode, the graphic characters are shown.

Perks does not contain a telephone dialer function, as Sidekick does. It was decided to leave off that function because of the difficulty in supporting different brands of modems. The Sidekick dialer will only work with a genuine Hayes modem, or a 100% compatible. The word is that users with modems that are supposed to be Hayes compatible have experienced difficulties using the Sidekick dialer.

Documentation And Help

The printed documentation provided with Perks is a bound booklet of 30 pages that is punched for a PC-sized 3-ring binder. It has no index, but the table of contents is complete enough for you to find just about anything you might be looking for. Perks also provides on-screen help. If you press the HELP key while a function is active, information appropriate to that function will be displayed. The help feature will only work if the help file provided with Perks is on a user-selected drive and directory. In Sidekick, the user cannot select the drive and directory for the help file.

Perks does not use the 25th line for prompts, as Sidekick does, nor does it allow any part of the windows to occupy the 25th line.

General Observations

Perks functions well, and is well documented. One thing that I would like to see added to it is the ability for it to recognize changes to the system time after it is loaded. Perhaps a new item in the file manager menu could be added to take care of this. Perks worked well with just about any program that I ran "over" it, with the exception of Heath/Zenith WordStar 3.3. In addition to the import problem noted above, Perks does not restore the cursor to the place where it was before you entered Perks. However, directing the cursor in any fashion, or typing any text will restore the cursor. Although Barry Watzman warned in the documentation that the calculator export feature will not work with some Heath/Zenith custom installations of WordStar, it worked fine in the normal customer version that I tried.

One program that Perks will not work with at all is the Professional Text Processor (PTP) by Newline Software. No harm is done by having Perks loaded when you run PTP, but you cannot invoke any Perks function while you are running PTP.

If you use a lot of other additional background utilities in addition to Perks, care should be taken as to the order in which they are loaded. Perks has the ability to unload itself completely from memory, but it can only do so if it is the last background utility loaded. All HUG background utilities, such as FASTIO, SCRNCLK, and KEYMAP will work with Perks, but if KEYMAP is loaded before perks, and you are running WordStar, the Shift-Break to invoke Perks will not be recognized until you type

another key. You can use an arrow key, so that no harm is done to the WordStar document. You can also load KEYMAP after Perks to eliminate the problem, but you cannot unload Perks later if you do. The keys are not mapped by KEYMAP while Perks is active, so you cannot use KEYMAP to improve the Perks notepad editor.

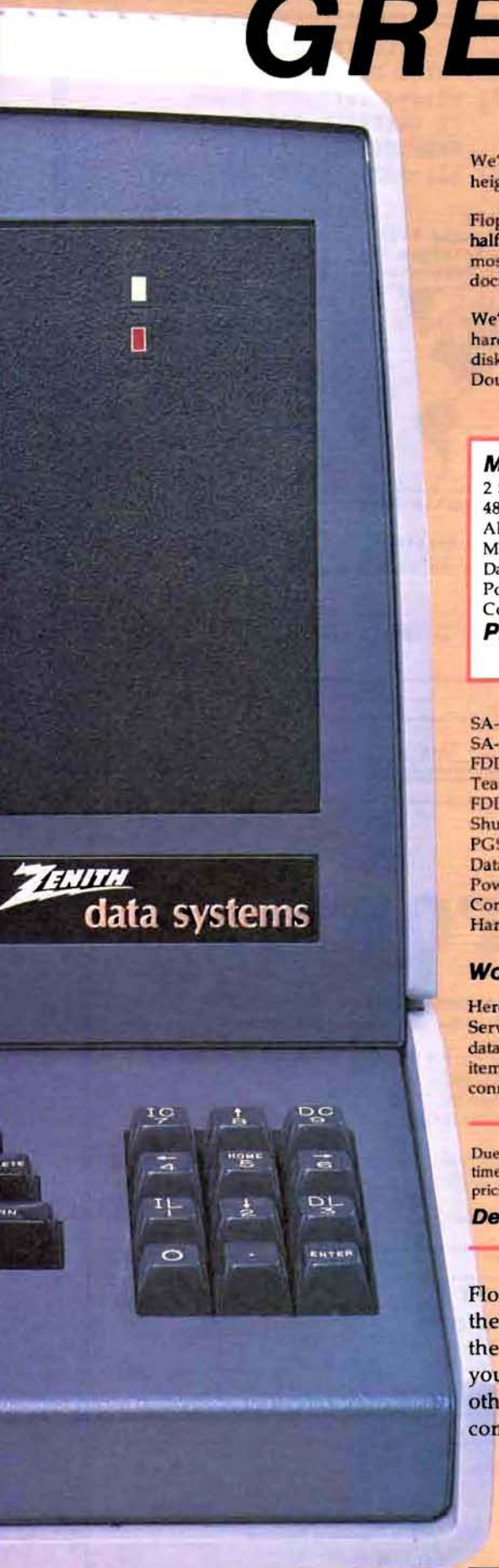
The screen clock produced by the SCRNCLK utility cannot be turned off once Perks is loaded. A new version of SCRNCLK to correct this is planned as of this writing, and patches to the old version will be published in REMark. If SCRNCLK is loaded after Perks, the clock can be turned off and on, but, as with KEYMAP, you cannot unload Perks.

Perks works equally well where you have text or graphics on the screen when you invoke it, and completely restores the screen when you exit from it. If you enter Sidekick while you are in a graphics mode, it must force the screen into the text mode while it works, and the rest of the screen outside the Sidekick windows looks like a real mess. Sometimes it does not faithfully restore the screen to what it was when it exits.

Perks is not copy protected, but employs a method of protection in which 80 percent of the code is encrypted with the user's name and serial number in the encrypted portion. The program will not run if any alterations are made to the name or serial number.

If you are a Z-100 user who has been envying Z-150 and other PC users for their ability to use Sidekick, now you can get Perks and let them envy you for a while. Perks is available from Barry Watzman, 560 Sunset Rd., Benton Harbor, MI 49022, phone (616) 925-3136. Perks sells for $99.97. A signed license agreement is required before the program will be shipped.

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.

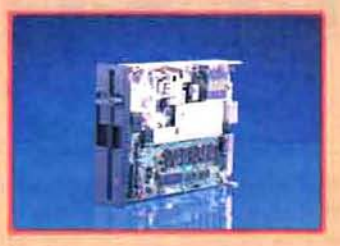
Floppy Disk Services inc. has been supplying custom enclosures and disk drives to the hobby and professional market for 6 years now! We have specifically supported the Heath/Zenith community with a high degree of success. We care about you and your system integrity. That's why we have been a success 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.

INTELLIBURNER

EPROM–EEPROM–MICROCOMPUTER PROGRAMMER

UNIVERSAL PROGRAMMING CAPABILITIES AT AN AFFORDABLE PRICE

- **Ultra Fast Programming – 2716's in 16 Seconds**
- **Programs & Verifies 8K thru 256K Single Voltage EPROMs**
- **Erases, Programs & Verifies 2815 & 2816 EEPROMs**
- **Programs & Verifies 8748 and 8751 Series MICROCOMPUTERS***
- **Programming Chracteristics Selected by Convenient Personality Jumper Plug (DIP Header)**
- **Program, Verify, Status, & Diagnostic Display with Tricolor LED**
- **Serial Interface — 3, 4, or 5 wire — 1200 to 19200 Baud**
- **Supports XON/XOFF and READY/BUSY Protocols**

NO SPECIAL SOFTWARE REQUIRED. Transfer disk files (Intel Hex Format) to EPROM with your system's line printer or modem software. Transfer EPROM contents to disk file in Intel Hex Format with your system's modem software. Or use the supplied software ** to transfer any binary or ASCII file to/from EPROM.

PROGRAMS :

2758 2716 2712B 27C16 8741* 8748*
2516 2732 2712BA2815 8742* 8749*
2532 2732A 27256 2816 8751*
2564 2764 68764 2817 X2212* 8755*
*Requires low cost personality adapter

SOFTWARE AVAILABLE FOR : **

CP/M systems on 8" SSSD - many 5¼ formats
TRS-80 Model I & III TRSDOS
Heath H8/H89 HDOS & CP/M
ZENITH Z90 & Z100 CP/M - Z100 ZDOS
KAYPRO II/IV
IBM PC-DOS - many MS-DOS systems

IntelliBurner Programmer with Software $ 299.00
RS-232 Interconnect Cable 12.00
Intelliburner PC Boards, EPROM, Plans & Software 99.00

Low Cost "DumBurner" serial programmers harness the power of your personal computer with the supplied software for full programming capabilities :

DumBurner II Programmer for 28 Pin and 24 Pin EPROMs and EEPROMs with software $ 199
DumBurner II Bare PC Board, Plans & Software 59
16K/32K DumBurner for 24 Pin EPROMs with software 149
16K/32K DumBurner PC Board, Plans & Software 59

ROSS CUSTOM ELECTRONICS
1307 Darlene Way - Suite A12
Boulder City, Nevada 89005
PHONE (702) 293-7426

!! NOW SHIPPING with ALL 'BURNERS !!
TOOLKIT Software for Editing EPROM Information, Plus HEX/MOTOROLA S/Binary File Conversion Utilities

Ultra Violet Products EPROM Erasers
Model QE-4 - Holds 8 EPROMS - Special Prices

Add $3 Shipping & handling. $2 Bare Boards. C.O.D. accepted. Foreign Orders add required postage. Specify Environment and Media Requirements. H8, H89, Z100 are TM Heath/Zenith. CP/M is TM Digital Research. TRS-80 Model III is TM Tandy Corp. IBM PC and PC-DOS are TM IBM. MS-DOS is TM Microsoft.

Welcome to DOODLER

It's Great!

Graphics Package

. . . for the Zenith Z-100 and Z-150/160

- Full feature graphics design package
- Save designs on disk for later use
- Playback mode for error correction
- Extended text capability includes. . .

User designed character fonts
Italic or backslant styles
All text may be scaled

See DOODLER at your local Heathkit Electronic Center

. . .or Send $ 79.95 directly to. . .

PAUL F. HERMAN
DATA SYSTEMS CONSULTANT
P.O. Box 206
Safety Harbor, FL 33572

Specification Sheet available on Request

HEATH/ZENITH 88, 89, 90 PERIPHERALS

16K RAM EXPANSION CARD

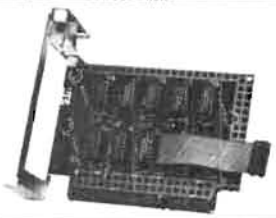
Expand your H/Z 88, 89 RAM Memory to a **FULL 64K** and begin using larger and more powerful programs with our **16K RAM** card

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

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

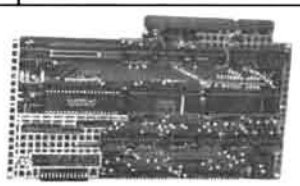
Field reliability record exceeding 3 years

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:

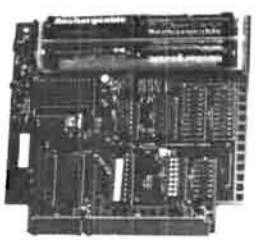
• 2 Serial Ports Supporting Ring Input, and External Clock • 3 Parallel Ports, 24 Total Digital Input/Output Lines • Fully Compatible with All Models of H/Z 88, 89, 90 using Heath/Zenith CP/M or HDOS • Now Supporting CP/M Version 2.2.04 • Choice of Centronics Line Printer Support Software for the CP/M 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 Price List

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

Part Number	Description of Product	Selling Price	Vol. Issue
HDOS HARDCOPY SOFTWARE			
885-1008	Volume I Documentation	9.00	
885-1013	Volume II Documentation	12.00	
885-1015	Volume III Documentation	9.00	
885-1037	Volume IV Documentation	12.00	8
885-1058	Volume V Documentation	12.00	
MISCELLANEOUS HDOS COLLECTIONS			
885-1032	Disk V H8/H9	18.00	8
885-1044-[37]	Disk VI H8/H89	18.00	
885-1064-[37]	Disk IX H8/H89 Disk	18.00	
885-1066-[37]	Disk X H8/H89	18.00	10
885-1069	Disk XIII Misc H8/H89	18.00	
GAMES			
HDOS			
885-1010	Adventure Disk H8/H89	10.00	4
885-1029-[37]	Disk II Games 1 H8/H89	18.00	8
885-1030-[37]	Disk III Games 2 H8/H89	18.00	8
885-1031	Disk IV MUSIC H8 Only	20.00	25
885-1067-[37]	Disk XI H8/H19/H89 Games	18.00	12
885-1068	Disk XII MBASIC Graphic Games	18.00	10
885-1088-[37]	Disk XVII MBASIC Graph. Games	20.00	14
885-1093-[37]	D&D H8/H89 Disk	20.00	16
885-1096-[37]	MBASIC Action Games H8/H89	20.00	18
885-1103	Sea Battle HDOS H19/H8/H89	20.00	20
885-1111-[37]	HDOS MBASIC Games H8/H89	20.00	23
885-1112-[37]	HDOS Graphic Games H8/H89	20.00	23
885-1113-[37]	HDOS Action Games H8/H89	20.00	23
885-1114	H8 Color Raiders & Goop	20.00	23
885-1124	HUGMAN & Movie Animation Pkg	20.00	41
885-1125	MAZEMADNESS	20.00	41
885-1130	Star Battle	20.00	47
885-1133-[37]	HDOS Games Collection I	20.00	59
885-8009-[37]	HDOS & CP/M Galactic Warrior	20.00	32
885-8022	HDOS SHAPES	16.00	45
885-8026	HDOS Space Drop	16.00	49
885-8032-[37]	HDOS Castle	20.00	59
CP/M			
885-1206-[37]	CP/M Games Disk	20.00	11
885-1209-[37]	CP/M MBASIC D&D	20.00	19
885-1211-[37]	CP/M Sea Battle	20.00	20
885-1220-[37]	CP/M Action Games	20.00	32
885-1222-[37]	CP/M Adventure	10.00	35
885-1227-[37]	CP/M Casino Games	20.00	38
885-1228-[37]	CP/M Fast Action Games	20.00	39
885-1236-[37]	CP/M Fun Disk I	20.00	55
ZDOS			
885-3004-37	ZDOS ZBASIC Graphic Games	20.00	37
885-3009-37	ZDOS ZBASIC D&D	20.00	50
885-3011-37	ZDOS ZBASIC Games Disk	20.00	52
885-3017-37	ZDOS Contest Games Disk	25.00	58
UTILITIES			
HDOS			
885-1022-[37]	HUG Editor (ED) Disk H8/H89	20.00	20
885-1025	Runoff Disk H8/H89	35.00	
885-1060-[37]	Disk VII H8/H89	18.00	
885-1061	TMI Load H8 ONLY Disk	18.00	
885-1062-[37]	Disk VIII H8/H89 (2 Disks)	25.00	
885-1063	Floating Point Disk H8/H89	18.00	
885-1065	Fix Point Package H8/H89 Disk	18.00	10
885-1075	HDOS Support Package H8/H89	60.00	
885-1077	TXTCON/BASCON H8/H89	18.00	
885-1079-[37]	HDOS Page Editor	25.00	15
885-1080	EDITX H8/H19/H89 Disk	20.00	
885-1082	Programs for Printers H8/H89	20.00	
885-1083-[37]	Disk XVI Misc H8/H89	20.00	11
885-1089-[37]	Disk XVIII Misc H8/H89	20.00	20
885-1090-[37]	Disk XIX Utilities H8/H89	20.00	22
885-1092-[37]	Relocating Debug Tool H8/H89	30.00	14
885-1098	H8 Color Graphics ASM	20.00	19
885-1099	H8 Color Graphics Tiny PASCAL	20.00	19
885-1105	HDOS Device Drivers H8/H89	20.00	24
885-1116	HDOS Z80 Debugging Tool	20.00	27
885-1119-[37]	BHBASIC Support	20.00	29
885-1120-[37]	HDOS 'WHEW' Utilities	20.00	33
885-1121	HDOS Hard Sec Sup Pkg 2 Disks	30.00	37
885-1123	XMET Robot Cross Assembler	20.00	40
885-1126	HDOS Utilities by PS:	20.00	42
885-1127-[37]	HDOS Soft Sector Support Pkg	30.00	45
885-1128-[37]	HDOS DISKVIEW	16.00	46
885-1129-[37]	HDOS CVT Color Video Terminal	20.00	46
885-8001	SE (Screen Editor)	25.00	28
885-8003	BHTOMB	25.00	28
885-8004	UDUMP	35.00	28
885-8006	HDOS SUBMIT	20.00	31
885-8007	EZITRANS	30.00	30
885-8015	HDOS TEXTSET Formatter	30.00	42
885-8017	HDOS Programmers Helper	16.00	42
885-8024	HDOS BHBASIC Utilities Disk	16.00	46
CP/M			
885-1210-[37]	CP/M ED (same as 885-1022)	20.00	20
885-1212-[37]	CP/M Utilities H8/H89	20.00	21
885-1213-[37]	CP/M Disk Utilities H8/H89	20.00	22
885-1217-[37]	HUG Disk Duplication Utilities	20.00	26
885-1223-[37]	HRUN HDOS Emulator 3 Disks	40.00	37
885-1225-[37]	CP/M Disk Dump & Edit Utility	30.00	40
885-1226-[37]	CP/M Utilities by PS:	20.00	40
885-1229-[37]	XMET Robot Cross Assembler	20.00	40
885-1230-[37]	CP/M Function Key Mapper	20.00	42
885-1231-[37]	Cross Ref Utilities for MBASIC	20.00	43
885-1232-[37]	CP/M Color Video Terminal	20.00	46
885-1235-37	CP/M COPYDOS	20.00	54
885-1237-[37]	CP/M Utilities	20.00	55
885-1245-37	CP/M 85 KEYMAP	20.00	63
885-1246[-37]	CP/M HUG File Manager & Utilities	20.00	64
885-5001-37	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
885-8018-[37]	CP/M FAST EDDY & BIG EDDY	20.00	43
885-8019-[37]	DOCUMAT and DOCULIST	20.00	43
885-8025-37	CP/M 85/86 FAST EDDY	20.00	49
ZDOS			
885-3005-37	ZDOS ETCHDUMP	20.00	39
885-3007-37	ZDOS CP/EMulator	20.00	47
885-3008-37	ZDOS Utilities	20.00	47
885-3010-37	ZDOS KEYMAP	20.00	51
885-3022-37	ZDOS/MSDOS Useful Programs I	30.00	63
885-3023-37	ZDOS/MSDOS EZPLOT	20.00	63
885-3026-37	MSDOS SMALL C Compiler	25.00	65
885-8029-37	ZDOS FAST EDDY	20.00	53
H/Z100 ZDOS - H/Z150 MSDOS			
885-3012-37§§	ZDOS HUG Editor	20.00	52
885-3014-37§§	ZDOS/MSDOS Utilities II	20.00	54
885-3016-37§	ZDOS/MSDOS Adventure	10.00	57
885-3020-37§	MSDOS HUG Menu System	20.00	62
885-3021-37§§	ZDOS/MSDOS Cardcat	20.00	63
885-3024-37§	ZDOS/MSDOS 8080 To 8088 Trans	20.00	64
885-3025-37§§	ZDOS/MSDOS Misc. Utilities	20.00	64
PC/IBM COMPATIBLE			
885-6001-37	MSDOS Keymapper	20.00	59
885-6002-37	CP/EMulator II & ZEMulator	20.00	59
885-6003-37	MSDOS EZPLOT	20.00	65
885-8033-37	MSDOS Fast Edit	20.00	62
PROGRAMMING LANGUAGES			
HDOS			
885-1038-[37]	Wise on Disk H8/H89	18.00	
885-1042-[37]	PILOT on Disk H8/H89	19.00	
885-1059	FOCAL-8 H8/H89 Disk	25.00	13
885-1078-[37]	HDOS Z80 Assembler	25.00	21
885-1085	PILOT Documentation	9.00	
885-1086-[37]	Tiny HDOS PASCAL H8/H89	20.00	13
885-1094	HDOS Fig-Forth H8/H89	40.00	18
885-1132-[37]	HDOS Tiny BASIC Compiler	20.00	59
885-1134	HDOS SMALL-C Compiler	30.00	63
CP/M			
885-1208-[37]	CP/M Fig-Forth H8/H89 2 Disks	40.00	18
885-1215-[37]	CP/M BASIC-E	20.00	26
BUSINESS, FINANCE AND EDUCATION			
HDOS			
885-1047	Stocks H8/H89 Disk	18.00	
885-1048	Personal Account H8/H89 Disk	18.00	
885-1049	Income Tax Records H8/H89 Disk	18.00	
885-1055-[37]	MBASIC Inventory Disk H8/H89	30.00	
885-1056	MBASIC Mail List	30.00	
885-1070	Disk XIV Home Fin H8/H89	18.00	
885-1071-[37]	MBASIC SmBusPk H8/H19/H89	75.00	17
885-1091-[37]	Grade/Score Keeping H8/H89	30.00	14
885-1097-[37]	MBASIC Quiz Disk H8/H89	20.00	18
885-1118-[37]	MBASIC Payroll	60.00	30
885-1131-[37]	HDOS CHEAPCALC	20.00	47
885-8010	HDOS CHECKOFF	25.00	32
885-8021	HDOS Student's Statistics Pkg	20.00	44
885-8027	HDOS SCICALC	20.00	50
CP/M			
885-1218-[37]	CP/M MBASIC Payroll	60.00	31
885-1233-[37]	CP/M CHEAPCALC	20.00	47
885-1239-[37]	Spread Sht. Contest Disk I	20.00	
885-1240-[37]	Spread Sht. Contest Disk II	20.00	
885-1241-[37]	Spread Sht. Contest Disk III	20.00	
885-1242-[37]	Spread Sht. Contest Disk IV	20.00	
885-1243-[37]	Spread Sht. Contest Disk V	20.00	
885-1244-[37]	Spread Sht. Contest Disk VI	20.00	
885-8011-[37]	CP/M CHECKOFF	25.00	32
ZDOS			
885-3006-37	ZDOS CHEAPCALC	20.00	47
885-3013-37	ZDOS Checkbook Manager	20.00	54
885-3018-37	ZDOS Contest Spreadsheet Disk	25.00	58
885-8028-37	ZDOS SCICALC	20.00	50
885-8030-37	ZDOS MATHFLASH	20.00	55
DATA BASE MANAGEMENT SYSTEMS			
HDOS			
885-1107-[37]	HDOS Data Base System H8/H89	30.00	23
885-1108-[37]	HDOS MBASIC Data Base Sys	30.00	23
885-1109-[37]	HDOS Retriever ASM (3 Disks)	40.00	23
885-1110	HDOS Autofile (2 Disks)	30.00	23
885-1115-[37]	HDOS Navigational Program	20.00	25
885-8008	Farm Accounting System	45.00	30

§ All program files run on both

§§ Program files run partially on both

Vectored to Page 99

HUG NEW PRODUCTS

HUG P/N 885-3027-37 MS-DOS
HUGPBBS $40.00

Introduction: HUGPBBS or HUG Personal Bulletin Board System, is a program that turns your computer system into an electronic bulletin board service. This bulletin board can be closed or opened to the general public. If this board is closed, each caller will be required to enter his own individual pre-assigned password as well as his name before he can leave or kill messages, and upload or download files. In addition to this capability, it also has a program database from which a user can upload and download files. Another feature is its ability to allow a caller to communicate, by way of his keyboard and crt, with you, the Sysop, should he desire. Although somewhat limited, another feature it has is its ability to allow you to access your computer, at system level, from a remote location. This ability is protected by three levels of security and can be totally disabled at configuration time.

Requirements: HUGPBBS requires the Heath/Zenith MS-DOS 2.0 (or higher) operating system on any H/Z-100 or H/Z-100 PC computer system. A minimum of two disk drives, any size are needed. One 5" drive is needed for the message base, and one 5" drive for the program data base. If a larger drive is available for the program data base, a greater amount of files can be stored. The program database feature CANNOT be disabled to eliminate the need for two drives. Finally, an auto-answer type modem is required. This software was written around the Hayes Smartmodem although it uses no Smartmodem English commands. All modem protocol is handled by way of the RS-232 signals. Aside from the standard signal and ground pins, pins 8 and 20 are used. Pin 8 is DCD, or data carrier detect. This line goes from -12v to +12v when a carrier is detected and vice versa when the carrier is lost. Pin 20 is DTR or data terminal ready. This line should be capable of causing the modem to dis-connect when it goes to -12v. It should also prevent the modem from answering the phone when at this level. It is the responsibility of the user to make his modem work properly with this software if he is not using a Hayes Smartmodem, Smartmodem 1200, or U.S. Robotics Courier 2400. Three baud rates are supported by the software (300, 1200, and 2400). It is not necessary, however, for the modem to be capable of all three. At the time of this writing, the Hayes 2400 baud modem will NOT work with this software. Although not necessary, a printer would allow the Sysop to see the name of each caller while he was away from the console as well as get a hard copy of any message on the bulletin board. It would also enable any user to leave a short three line message to the Sysop upon exit from the board.

On the H/Z-100 PC, either COM1 or COM2 can be used for the modem.

The following files are included on the HUG P/N 885-3027-37 HUGPBBS disk:

HUGPBBS	.EXE	SBULETIN	.HPB
BULLETIN	.HPB	NEWHELP	.HPB
USERLOG	.HPB	MERGE	.BAT
EDIT	.COM	EDIT	.DOC
README	.DOC	AUTOEXEC	.BAT

An extensive users' manual is also included with this software.

Program Author: Jim Buszkiewicz (Heath Users' Group)

Program Content: The bulletin board portion of this software has the following commands available to the user when he calls in:

B - Bulletins (file created by Sysop)
C - Catalog Of Program Data Base
D - Download A File
E - Enter A Message
G - Goodbye (dis-connect)
H - Help
I -Information (file created by Sysop)
K - Kill (a message)
N - Name
RI - Retrieve Individual Messages
RC - Retrieve Continuous Messages
SM - Scan And Match
SQ - Scan Quick
SR - Scan And Retrieve
SS - Scan Subject Headers
T - Talk (with Sysop)
U - Upload A file

When the board is inactive, the following commands are available to the Sysop at the main console:

Control-C - Warm Boot (available anytime and dis-connects)
Control-E - Enter (message)
Control-H - Help
Control-K - Kill (ANY message)
Control-Q - Quick Scan
Control-R - Retrieve (message)
Control-S - Scan
Control-X - Configure System

Operating Notes: Uploading and downloading files to the program database can be achieved using either Xmodem protocol (with checksum error testing), or with a simple capture buffer (ascii files only). A standard 192k H/Z-100 system will have a

65520 byte capture buffer available for file uploading.

The short bulletin, regular bulletin and information files can be created using any text editor or word processor. The standard MS-DOS HUG Editor was included with this software for this purpose. This is the editor found on the HUG P/N 885-3012-37 disk. The short bulletin and regular bulletin files automatically appear when a user first signs on. The regular bulletin can be recalled with the 'B' command. The difference between the two bulletin files is that the short bulletin file is forced upon the user and must be read when he logs in. This file is usually kept quite short. It can also contain your own personalized header to identify the bulletin board system the caller has logged onto. The regular bulletin file is also presented to the user, but can be aborted at any time. The Information file will be displayed for the user only when the 'I' command is issued.

The bulletin board system can contain a total of 128 messages. These messages can contain a maximum of 16 lines with 77 characters in each line. A simple line replacement editor is available to the user to edit a message he may be working on.

Messages can be protected from being killed by other users with a password of up to 5 characters. Both upper and lower case is recognized in this password (i.e. the password 'FIDO' is not the same as 'fido').

The 'SM' (Scan and Match) command allows a user to search one of three fields (To, From, and Subject) to match a string which he enters. With this command, a user could possibly search for any messages to him, from his girl friend, or any subject matter. Character case is ignored in this command.

For a closed bulletin board system, user passwords must be issued and changed by you, the Sysop. A user cannot enter, or change his own password.

Simplified software configuration was a key factor in the writing of this system and is accomplished by answering questions when the software is first executed. This configuration is then permanently recorded. Re-configuring the system is just as easy and can be done at any time.

Some of the options you are allowed to change during configuration time are:

1. Drive designation where each system file is to be placed.
2. Open or closed bulletin board selection.
3. Dead time period (no keyboard activity from caller) to dis-connect.
4. Enable system level access.

and more.

Files that are uploaded to the database are not immediately made available to the general public until the Sysop has had a chance to view the contents and validity of those files. Once these files are 'merged' into the database by the Sysop, only then will the general public see them in the database catalog.

While a caller is using the bulletin board system, the Sysop is allowed to observe all activity on his CRT. During this period of time, the Sysop is not allowed to interfere with the activities of the caller other than to totally abort the program which immediately dis-connects the phone connection.

A note of warning: Although system level access is attainable from a remote location, MOST, and I repeat MOST MS-DOS SOFTWARE WILL NOT EXECUTE PROPERLY! The only software that will, is software which is totally hardware independent. The HUG editor supplied with this program is just such a piece of software and will work properly from a remote terminal. Also, due to BIOS inadequacies, Control-S pause and Control-C break will not work on the H/Z-100 PC from a remote terminal. They will work, however, on an H/Z-100.

Comments: Knowing how to use some sort of text editor or word processor is imperative to the successful operation of this software.

TABLE C Rating: (0), (1), (2), (5), (10)

HUG P/N 885-3028-37 MS-DOS
HUGPBBS SOURCE LISTING $60.00

Description: This disk contains the source listing for the program 'HUGPBBS', HUG p/n 885-3027-37. This listing contains all com-

TABLE C
Product Rating

10 - Very Good
9 - Good
8 - Average

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

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

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

ORDERING INFORMATION

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

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

NOTE

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

ments and can be assembled using Microsofts' assembler, MASM, version 1.27 or greater. This source listing is being made available on an 'as-is' basis and modifications made to it are done so at the sole responsibility of the user. No other files or documents are included with this product.

Requirements: An H/Z-100 or H/Z-100 PC system running MS-DOS version 2.0 or greater is required to assemble this program. This source can be assembled on two 5" drives if a listing file is not sent to either drive. If a listing file is required, an 8" or Winchester drive will be required. The source file is approximately 114k and the listing file generated will be greater than 300k.

The following files are included on the HUG P/N 885-3028-37 HUGPBBS SOURCE LISTING disk:

HUGPBBS	.A86	DEFMS	.ASM

Program Author: Jim Buszkiewicz (Heath Users' Group)

Comments: None

TABLE C Rating: n/a

HUG P/N 885-3029-37 MS-DOS
HUG Background Print Spooler $20.00

Introduction: The HUG Background Print Spooler (HBPS) is a program that allows you to assign some of your computer's memory (any amount from 4k to 512k) to be used as a print buffer. With HBPS loaded, when a program sends text to the printer (via normal DOS functions), it does not go directly to the printer, but into the HBPS buffer. HBPS then takes the characters from the buffer and sends them to your printer. Since HBPS can accept characters into its buffer more rapidly than the printer can accept characters, your computer can complete a print operation faster than without HBPS, and go onto 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, or H/Z-100 PC series (H/Z-150, etc.) computer, at least 192k of system RAM, and the MS-DOS or Z-DOS operating system. If Z-DOS is used, it must be the latest release. The maximum amount of memory assignable to HBPS with MS-DOS version 1 or Z-DOS is 64k (512k with MS-DOS version 2).

This disk contains the following files:

README	.DOC	HBPS	.Z1F
HBPS	.Z2F	HBPS	.Z1S
HBPS	.Z2S	HBPS	.PC1
HBPS	.PC2	HBPS	.DOC
SETSP	.COM	PR	.COM
BASPATCH	.COM	WSPATCH	.COM
WRDPATCH	.COM	SCRNCLKZ	.WHT
SCRNCLKZ	.RED	SCRNCLKZ	.GRN
CLKZ	.COM	SCRNCLK	.WHT
SCRNCLK	.RED	SCRNCLK	.GRN
CLK	.COM	HBPS	.ASM
SETSP	.ASM	PR	.ASM
BASPATCH	.ASM	WSPATCH	.ASM
WRDPATCH	.ASM	SCRNCLKZ	.ASM
CLKZ	.ASM	SCRNCLK	.ASM
CLK	.ASM		

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

HBPS.Z1F, HBPS.Z2F, HBPS.Z1S, HBPS.Z2S, HBPS.PC1, HBPS.PC2 — These are versions of HBPS for use on different equipment, and under different situations. Instructions for selecting a version, and for using HBPS are in the file HBPS.DOC.

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

HBPS.DOC — Instructions for using HBPS.

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

PR.COM — This is a program that allows you to rapidly copy disk files to the HBPS print buffer. Because the system command processor does not output to the printer via the normal DOS "channel", using the COPY command, as in

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

to copy a file to the printer will not take advantage of HBPS. Instead, you can copy PR.COM to your system disk, and enter

```
A>PR FILENAME.EXT
```

where FILENAME.EXT is the name of the file to print. If PR is used under MS-DOS version 2, a full path name can be used. Very large files can be "printed" in seconds using HBPS and PR.

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

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

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

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

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

SCRNCLK.WHT, SCRNCLK.RED, SCRNCLK.GRN — This is a

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

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

HBPS.ASM, SETSP.ASM, PR.ASM, BASPATCH.ASM, WSPATCH.ASM, WRDPATCH.ASM, SCRNCLKZ.ASM, CLKZ.ASM, SCRNCLK.ASM, CLK.ASM —These are the assembly language source files for the above programs.

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

HUG P/N 885-5009-37 CP/M-86
HUG Background Print Spooler $20.00

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

Requirements: HBPS requires an H/Z-100 series computer, at least 192k of system RAM, and the CP/M-86 operating system (either the Heath/Zenith or the Watzman implementation).

This disk contains the following files:

README	.DOC	HBPS	.CMD
HBPS	.DOC	SIZESP	.CMD
SETSP	.CMD	FIXSUB	.SUB
FIXBAS	.SUB	SCRNCLK	.WHT
SCRNCLK	.RED	SCRNCLK	.GRN
CLOCK	.CMD	HBPS	.A86
SIZESP	.A86	SETSP	.A86
SCRNCLK	.A86	CLOCK	.A86

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

HBPS.CMD — This is the HBPS spooler program, ready to be copied to your system disk and loaded into memory.

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 a file to LST: to print it.

HBPS.DOC — Instructions for using HBPS.

SIZESP.CMD — This program allows you to adjust the size of the buffer used by HBPS. It can be set to any value from 4 to 512k.

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

FIXSUB.SUB — The HXSUB utility on HUG disk 885-5003-37 will not work with HBPS installed. This file is a SUBMIT file that can be used to fix HXSUB, so it will work with HBPS installed.

FIXBAS.SUB — This is a SUBMIT file for patching the Heath/Zenith release of MBASIC version 5.22 so that it outputs to the printer via DOS function 5. Without this patch, MBASIC bypasses HBPS when you use LLIST or LPRINT. Instructions for installing the patch are in HBPS.DOC

SCRNCLK.WHT, SCRNCLK.RED, SCRNCLK.GRN — This is a modi-fied version of the SCRNCLK program for the Z-100 originally released on HUG disk 885-5003-37. The original version is slightly incompatible with HBPS in that the display cannot be turned off if it is loaded before HBPS. This version can be loaded before or after HBPS. It also works with the Watzman implementation of CP/M-86, as well as the Heath/Zenith implementation. When loaded, this program causes a digital clock display to appear in the upper right corner of your screen. The file name extension determines the color of the clock display (white, red, or green).

CLOCK.CMD — This program is used to control the SCRNCLK program. It allows you to turn the clock display off or on.

HBPS.A86, SIZESP.A86, SETSP.A86, SCRNCLK.A86, CLOCK.A86 — These are the assembly language source files for the above programs.

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

*

WATCHWORD

VERSION 2

The word processor and full screen editor for the Z100 and ZDOS.

NEW FEATURES:

- *Files of arbitrary length*
- *Extensive configuration facility*
- *Multiple fonts loaded from ALTCHR files*
- *Support for ZDOS Version 2 subdirectories*

Plus all of the features of Version 1 including: superscripts, subscripts, underlining, boldface, centering, formatting, microjustification, arbitrary line length with horizontal scrolling, word wrap, auto save, split screen, macros, user settable defaults and definable keys, global search and replace with settable zone.

For additional details, send a SASE.
Demo Disk: $3.00
WatchWord with Manual: $100.00

S & K Technology, Inc.
4610 Spotted Oak Woods
San Antonio, TX 78249
512-492-3384
ATTN: Steve Robbins

$AVE TIME AND MONEY

Spend your precious time executing your applications instead of processing MS-DOS commands with the Genesis *MENU SHELL*. This integrated, interactive menu maintenance utility allows you to create your own menus with a powerful batch job generator. All user applications can be executed by entering a number or acronym name. *ORDER YOURS TODAY!!!* **Only 34.95!**

The Genesis Check disbursement System permits you to enter information from your checkbook and produce a detailed report by expense area or check number sequence. Then you can produce a spreadsheet which summarizes each expense area by month! **Only 34.95!**

OTHER SPECIALS INCLUDE

DS/DD Diskettes 10 per box 5 year guarantee	**$ 24.95**
Z-150 PC IBM compatible	**CALL**
Lotus 1-2-3 for your Z-100	**250.00**
MICROSOFT WORD	**99.00**
MULTIPLAN	**195.00**
8 Bit and 16 bit software	**CALL**
Z-150 44MB internal hard disk kit includes 640K	**2995.00**

PLACE YOUR ORDER TODAY!!
CALL DOMINO COMPUTER SERVICES, INC.
(312) 870-8707
Doug Balut

IBM-PC is a trademark of International Business Machines.
MS-DOS is a trademark of Microsoft Corporation.

DISK CONTROLLER CARDS

DTC-5150BX Controller Card. $ 295.00
For internal Z-150/160/IBM Winchester

DTC-10-1 S-100 Host Adaptor $ 225.00
For Z-100 external Winchester drives with SASI/SCSI interface

DTC-510B SASI Controller $ 295.00
Supports one or two Winchester drives

Diskettes

Double sided, double density with hub rings and labels. Hard or soft sector format (specify). $ ~~16~~ 15.00 per ten.

Cases

Translucent smoke-color plastic case holds ten diskettes. $3.00 ea. with your purchase of $16.00 or more.

Winchester

For Z-100 series computers

External mounting.
Includes host adapter, controller card, power supply, drive unit, and software for MS-DOS 2.
5 Megabyte system $ 1099.00
Add $200 for 10Mb drive,
$600 for 26Mb drive.

For Z-150 series computers/PC's

Internal half-height mounting.
Includes drive adapter and drive unit.
10 Megabyte system $ 999.00
Add $400 for 26Mb drive.

LARGE CAPACITY DRIVES AVAILABLE

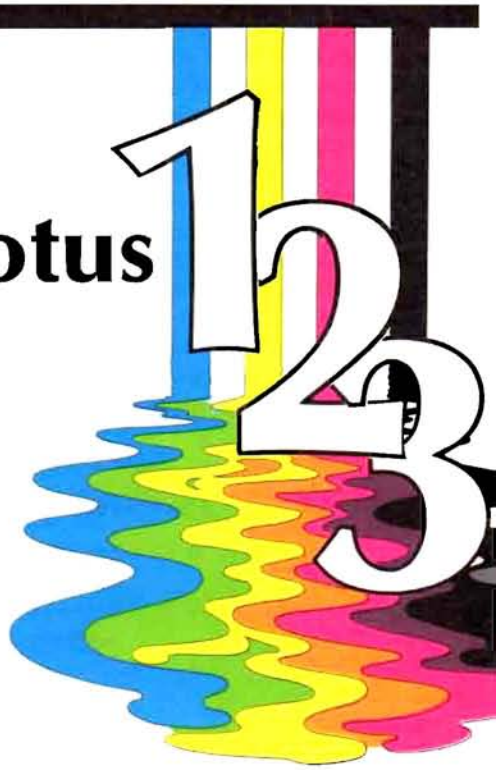
Call for latest pricing and availability of 64Mb to 380Mb Winchester external drives in the 5.25" size.

Payment by Personal or Company check, money order, or COD.
Certify checks over $100. U.S. shipment included in prices.

LINDLEY SYSTEMS 21 Hancock Street, Bedford, MA 01730 (617) 275-6821 6-10 PM and all day Sat.

Colour Your Lotus World

Putting Colors Into Lotus



Dean J. Rath
667 Sixth Street
Trafford, PA 15085

This article addresses modifications to the supplied Lotus 1-2-3 text display driver for an H/Z-100 with color RAM or an H/Z-150 with color/graphics and a color monitor. This allows for better utilization of the color capabilities than the original driver, and two implementation alternatives for an H/Z-100 are discussed, as well as a similar modification for an H/Z-150. A color patch for WordStar is also included.

I am a dedicated user of Lotus 1-2-3, having graduated from the use of a previous manifestation of a spreadsheet program, which shall remain nameless, to the thrill and power of driving such a deluxe software product. Most of my experience had been gleaned from using 1-2-3 on an IBM Personal Computer at work (I know, I know — but sometimes paychecks come before pride), so when it was decided to purchase a computer for home use, the fact that a copy of 1-2-3 was included with the purchase of an H/Z-100 system as part of a Zenith Data Systems promotion, that weighed heavily in favor of Heath/Zenith. The system that was finally procured contained the color random access memory (RAM) for all three color planes, and a Zenith ZVM-135 RGB color monitor.

Disappointment

After faithfully following the Lotus instructions for the installation of ZDOS and the driver installation onto each of the Lotus disks, it was disappointing to say the least, when presented with a spreadsheet display in black and white on the nice new color monitor. Reading the Lotus literature provided no insight whatsoever on how to change the colors for the display. The only color at all on the 1-2-3 program was that the WAIT indicator in the upper right corner of the screen was in cyan, and it only appeared for short intervals while the program was busy doing file or recalculation operations. Some sleuthing revealed that there is quite a bit more color capability available in the driver than Lotus, for some strange reason, decided to implement or explain. The information that follows is the result of experimentation with the Lotus program files, and represents the deductions of information obtained empirically.

Drivers

The Lotus 1-2-3 program uses four overlay drivers to perform special functions that are system dependent. When you perform the color installation procedure of the Lotus literature, you are actually copying four specially created driver files (they all have 2-letter names and a .DRV extension) to each of the Lotus disks. The one that we are interested in for this article is the TD.DRV, or text display driver file. This file contains the information on how to set up the text and message displays, establish the border area for the spreadsheet, clear the screen, and set the colors. Just so your enlightenment is complete, the other three driver files control the graphics displays, the keyboard, and the printer interface. The following information directly applies to ZDOS version 1.1 and version 1A of the Lotus 1-2-3 program, but should be applicable to ZDOS(MSDOS) version 2.0 or higher.

Remember, when using DEBUG to patch a program or data files on disk, always make a backup copy of the file or disk that you intend to modify.

The first two modifications are for implementation on an H/Z-100, while the third is for use on an H/Z-150. For all modification methods described below, it will be assumed that the disk that will be used for the patching contains both the TD.DRV file from the Lotus disk and the DEBUG.COM file from the ZDOS disk. Those familiar with the use of DEBUG may wish to deviate from the example.

H/Z-100 Modification 1

This modification is for those people who like having a black background for their spreadsheet, but still desire more color from the program. There are four bytes that control aspects of the display characteristics that will be discussed. The first byte is at address xxxx:19d. This byte determines the display color of the control panel information and the work area information of the 1-2-3 display. The second byte is at address xxxx:19e and it controls the display of the border area of the worksheet. The third byte is at xxxx:19f and the display characteristics that it controls do not appear in the 1-2-3 program, but it determines the color of the headings shown in the PrintGraph portion of the Lotus Access System. The fourth byte at address xxxx:1a6 affects the color of the WAIT indicator.

Before presenting an example of a modification that can be

made to the program is presented, I would like to describe my findings about the color selections available for this modification. Remembering that the H/Z-100 supports eight colors for display purposes, an explanation of how the bytes are used should be helpful. If you understand that the DEBUG program, when run, displays all byte information as two characters in the base 16 (or hexadecimal) number system, then the following explanation and information shown in Table 1 will be straightforward. The first hexadecimal character of the byte represents the color that is desired, according to the list in Table 1. The second hexadecimal character of the byte represents a display mode for presenting the color indicated by the first hexadecimal character. The results that were obtained by trying combinations of these parameters are summarized in the table.

Table 1

# (in hex)	Color	# (in hex)	Characteristic
0	White	0	Chosen color is foreground on Black background
1	Cyan	1	Chosen color is background with black letters
2	Magenta	2	Chosen color is foreground on Black background with an underscore
3	Blue	3	Chosen color is background with black letters with an underscore
4	Yellow		
5	Green		
6	Red		

Numbers higher than 6 for colors are error conditions.

Any characteristic numbers higher than 3 are reduced modulo 4.

For example, let's suppose that it is desired to have the menus and cell entries be green on a black background, and the border to be cyan with black letters. We also wish the PrintGraph headings to be red letters on a black background, and the WAIT indicator to be black letters on a red background. The modifications to the TD.DRV would be as follows:

```
>debug td.drv       (this invokes debug at the DOS level)
-e19d
xxxx:019d 00.50  01.11  20.60
-ela6
xxxx:01a6 11:61
-w
writing 0EE4 bytes
-q
>                   (back at DOS level)
```

The newly entered byte at xxxx:019d indicates that menus and cell information are to be in green (color 5 from Table 1) and the color is the foreground color with a black background (the second hexadecimal character being 0). The same logic holds true for each of the other bytes in the example.

H/Z-100 Modification 2

This modification is for those people who would like to have more color from the program than just a plain black background. Besides the four bytes that control the aspects of the display discussed under the first modification , I was able to locate another byte that allows the background color to be changed from black to one of the other colors. There are several side effects to changing the background colors that manifest themselves in the manner of choosing appropriate bytes for the same four display bytes mentioned in the first modification. This additional byte is at address xxxx:B30, and it controls the amount of background color to appear on the screen. By testing different values, it was determined that the bit pattern of the byte at this location determines the appearance of the background with 00 being none (black background), hexadecimal FF being completely solid, and the other values in between form vertical bar patterns of various widths and spacings. Another major side effect is that the color table is altered. The new color information for this case is presented in Table 2.

Table 2

# (in hex)	Color	# (in hex)	Characteristic
0	Black	0	Chosen color is background with white letters
1	Red	1	Chosen color is foreground on white background
2	Green	2	Chosen color is background with white letters with an underscore
3	Yellow	3	Chosen color is foreground on white background with an underscore
4	Blue		
5	Magenta		
6	Cyan		

Numbers higher than 6 for colors are error conditions.

Any characteristic numbers higher than 3 are reduced modulo 4.

To use another example, suppose that it is desired that the menus, cell entries, and border be on a blue background, and the border is to be white with red letters. We also wish the PrintGraph headings to be white letters on a red background, and the WAIT indicator to also be white letters on a red background. The modifications to the TD.DRV would be as follows:

```
>debug td.drv       (Starts debug from the DOS level)
-e19d
xxxx:019d 00.40  01.11  20.10
-ela6
xxxx:01a6 11.10
-eb30
xxxx:0b30 00.ff
-w
writing 0EE4 bytes
-q
>                   (return to DOS level)
```

The logic for understanding each of the bytes entered is identical to that presented in the first modification section. The key to this modification is that the first color byte at address xxxx:019d represents the background color for the whole screen. In this case the text will be white letters on a blue background (a 4 for blue and a 0 for white letters on blue background) for everything on the screen except the border.

H/Z-150 Or 160 Modification

For those who own or use an H/Z-150,160 or an IBM-PC, there are similar bytes to the ones discussed for the H/Z-100 that control the foreground and background colors in these machines, provided that color/graphics capability is available. These are easier to explain than the H/Z-100 setup due to the different 6845 video chip utilization in the H/Z-150. The color selections for text, border, PrintGraph headings, and the WAIT indicator follow the color pallette described for the text color capabilities, with the background colors from 0 to 7 and foreground colors from 0 to 15. The pallette is presented in Table 3.

Table 3

# (in Hex)	Color	# (in Hex)	Color
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	A	Light Green
3	Cyan	B	Light Cyan
4	Red	C	Light Red
5	Magenta	D	Light Magenta
6	Brown	E	Yellow
7	White	F	High-Intensity White

Colors 8–F may be used only for foreground colors — attempts to use colors 8–F for background colors causes the selected color combination to blink.

The following is an example equivalent to Modification 2 for the H/Z-100, and will provide white text on a blue background, red letters on a white border, and high-intensity white letters on a red background for PrintGraph headings. The WAIT indicator will also be high-intensity white letters on a red background.

```
>debug td.drv
-e17d
xxxx:017d  07.17  30.74  0A.4F
-e186
xxxx:0186  C0.4F
```

In the H/Z-150, the first hexadecimal character of the byte represents the background color, while the second hexadecimal character represents the foreground color to be utilized. Thus, the byte at xxxx:17d represents a background color of blue (1) with a foreground color of white (7).

Summary

Once you have made the changes to the TD.DRV file in the manner desired, the next obvious thing is to test the changes. The goal of this whole article is to provide you with the means to generate a set of colors that you enjoy, and the combinations presented are for example only, although they are not unpleasant on my system. In order to test the changes, the modified TD.DRV file must be copied to the Lotus 1-2-3 system disk. If you find that the color combination is not appealing, then use DEBUG again to try a different combination. Different people perceive colors differently, and even different monitors of the same model and manufacturer do not present colors identically. Also, do not forget that screen intensity affects color perception. Once you have found a combination of colors that are pleasing to you, then copy the modified TD.DRV file to all of your Lotus disks. You will be amazed at what a difference colors can make to the other parts of the Lotus system, particularly the tutorial.

H/Z-100 WordStar Patch For Color

I greatly enjoyed reading the patches for WordStar that Pat Swayne presented in his article in the February, 1985 issue of REMark. Since the whole gist of this article has been on the subject of color, I would like to pass along a patch that has been implemented on a version 3.2x of the WordStar program that enables me to have color presentation on an H/Z-100. By making appropriate modifications to the terminal strings :TRMINI, :IVON, and :IVOFF, the WordStar screen consists of green text on a black background, while the menus are made up of white letters on a blue background.

```
>debug ws.com
-e292
xxxx:0292  02.06  1B.1B  45.6D  00.34  00.30  00.1B  00.45
```

Note that the change above to :TRMINI assumes that the patch that Mr. Swayne presented in his article has been implemented. What has been done is to precede his recommended bytes by the command to be included for setting the colors. If your setup string varies for some reason, add the first four bytes to any setup string you may be using — up to the 8 byte limit — or use a jump to the user patch area if more space is needed.

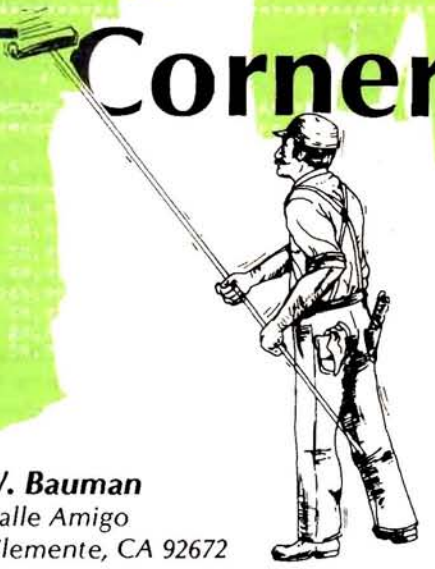
```
-e284
xxxx:0284  00.04  00.1B  00.6D  00.37  00.31
-e28b
xxxx:028b  00.04  00.1B  00.6D  00.34  00.30
```

These entries to the :IVON and :IVOFF strings change the highlighting (which is used for the menus) from ordinary inverse video into color changes. The third and fourth bytes of the :IVON string — in this example the hexadecimal 37 and 31 — set the foreground and background colors to white on blue. The colors for the third and fourth bytes of the :IVOFF string should match the colors used in the :TRMINI string, so that screen updates leave no residual color patches.

*

Part 10

SPREADSHEET Corner



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

Many computer owners believe that the dBASEII database management program is too complicated and difficult to use. I think that is because it is so "powerful" and has an extensive set of commands without any form of user friendly menus. It is really a dual software package! First, you can use the built-in features to easily create a database model and put it to valuable and useful purposes. Second, it is a programming language that you have to learn like BASIC, but I think it is easier. Learning to program is time consuming because to learn it you have to program over-and-over until it becomes automatic.

This article will demonstrate that you do not have to be a programmer to use dBASEII. I will show you how to create a database model in a very short time and you will only have to learn a few of the many commands and still produce a useful database program!

I would like to go over one very important thought with you readers before we go any further with dBASEII. To choose your computer system, the critical hardware item for a microcomputer database set-up is the disk drive capacity, NOT working memory as found with spreadsheet software. The disk capacity must be sufficient to store the entire set of files for the database on a single drive. Therefore, I would suggest 320K drive capacity will prove adequate for most applications. Under CP/M, dBASEII will need 64K of RAM and with MS-DOS 128K or 192K of RAM would be the minimum RAM. The reason for this turnabout from spreadsheets is that all of the spreadsheet program is always in RAM and it is manipulated in RAM, but with dBASEII the files are stored in disk files and worked with from disk files. Thus, if a "sort" is called for, a portion of the disk file will be read at a time into RAM to be worked on and then put into a second disk file and so on. This should help you define your computer needs.

I have discussed things like second-generation software in "Spreadsheet Corner" and I am sure that you have heard of third generation hardware. But, I wonder if you have heard about fourth generation computer users! I would like to explain:

First generation computer users were hobbyist. I know a lot of us can identify with this group. This group bought a computer with no particular application in mind. They just wanted a computer and wanted to know how to make it work.

Second generation users were also hobbyist but their interest was software more than hardware. Some of this group liked to program while others liked games. Many of this software group used their computers as a challenge for nonessential applications.

Third generation computer users bought their computers as productivity tools. They either wanted to make a career with it or use it with their career. It was spreadsheet and word processing software that got their attention. This third generation does not know as much about their computers as the above two. Their needs are fairly well defined so that they could make intelligent software selections and put these software tools to effective use. This then leads them to the type of computer that they require. I think that most of the readers of "Spreadsheet Corner" fit into this group!

Fourth generation computer users, like the third, are application oriented. But besides wanting to process words and/or numbers, they want to process data! Here, words and numbers are thrown together in a endless array of files and records that will require retrieving, manipulation, and printing in diverse and unanticipated ways.

This is where Database Programs, such as dBASEII and Condor to name a couple, come into the picture. Database Programs are expensive and complicated as compared to word processors and spreadsheets. Experts do not always agree on what constitutes a database management system. I do want you to see that spreadsheets and database programs go together like bread-and-butter. With this in mind, many software companies have produced integrated software packages — Lotus 1-2-3 and Symphony as examples.

Currently, the micro database managers fall into two groups:

1. File Managers (sometimes called flat files).
2. Data Managers.

As yet, none of the integrated packages that I have worked with offer database features found in stand-alone database programs. Furthermore, the database portions of the integrated packages usually are not relational programs that have become the indus-

try standard. They are simple file managers that provide an interactive environment for the application. I do not mean to say that they are not useful, but they are limited in use. Therefore, if your use NEEDS a full feature, sophisticated database program, the database residing in the integrated package is generally not your answer. Going a step further, larger databases will not fit into the working memory (RAM) oriented integrated packages.

Database Programs, in contrast, allow users large database files that are limited by disk storage capacity rather than RAM! They further allow the user to interact with at least two and often many more files concurrently. These programs are equipped with query languages and report generation features. They are the more VERSATILE business data processing tool!

Probably the most oversold and overpraised program ever produced for microcomputers is dBASEII database management system. It really is not all that good. It is slow, especially for sorting. After several revisions it still has bugs, although documented in the industry. The manual is horrendous. It is an 8-bit CP/M program that has been "ported" to 16-bit MS-DOS systems. However, it is the "de facto" standard relational database in the microcomputer marketplace! It is the granddaddy of databases. It is a tradition! I could have started with Condor (I prefer it) in place of dBASEII, but I am sure that more readers will already have dBASEII and really do not know how to use it.

I believe that you readers have seen the dBASEIII reviews telling us that it is light years ahead of dBASEII! This is an exaggeration, at the very least. Simply said, dBASEIII has somewhat more power and fewer limitations but requires more computer memory. It has an improved manual, but not a good one! It has a few additional features for the dBASE programmer type. I am not ready to pay out more money and relearn the dBASEIII. If I was going out to buy now, I would of course buy the dBASEIII.

Once you get beyond my quibbles, you will find that I believe that Database Programs are very useful to the business environment. Creating a Database Management System for a business requires hard work and takes time, but MOST computer processes that yield bottom-line results take hard work. In a database structure, actual data that make up the business must be understood and organized in the most natural fashion. I will attempt to get you thinking that way in this article.

I will be using dBASEII, Version 2.41 that has some real improvements over earlier versions with my H/Z-100. It is available from Heath/Zenith as catalog number AT-5065-1. This software can be used with either the H/Z-100 or the H/Z-150 type computers using the MS-DOS operating system. It is an Ashton-Tate product. The package comes preinstalled for the H/Z-150 computers. An INSTALL program is included to adapt the package to the H/Z-100 types. A SET COLOR option is available for the H/Z-150. The Screen Editor is furnished in two versions. SED100.MSG for the H/Z-100 and SED150.MSG for the H/Z-150. The function keys are programmed as follows:

dBASEII	Z-150	Z-100	Command
F1	F1	HELP	HELP SCREENS
F2	F2	F0	DISPLAY;
F3	F3	F1	LIST;
F4	F4	F2	LIST FILES;
F5	F5	F3	LIST STRU;
F6	F6	F4	LIST STAT;
F7	F7	F5	LIST MEMO;
F8	F8	F6	CREATE;
F9	F9	F7	APPEND;
F10	F10	F8	EDIT;

Beginning in 1985, the IRS will require the business user to keep adequate contemporaneous (How do you like that word?) records to substantiate the following business expenses:

1. Travel including meals and lodging — also local travel.
2. Entertainment and gift expenses.
3. Computers and accessory equipment NOT used EXCLUSIVELY at a regular business establishment or a QUALIFIED home office.

If the reader fails to keep such records, no deduction or credit will be allowed for the expense. The adequate contemporaneous record requirement can be met by keeping a diary, daily log, or similar record that contains a separate entry identifying each business expense and each business use of the property.

"Spreadsheet Corner" readers would most likely be interested in #3, but I will include the others for this project. To meet the requirements for an automobile or a computer, the entries MUST show the mileage an automobile was driven for business or nonbusiness purposes. For computers and accessories the entries must show the time used for business or nonbusiness. NOTE! I keep track of the nonbusiness use so that the percentage of business use can be calculated (should be 75% or higher).

Good records are needed on a day-by-day basis and in the case of computers on an hour-by-hour basis. To provide such records with a simple system, I am suggesting a data base! Guess what? I will use this project to start the readers using dBASEII (or some similar program the reader may choose). I am suggesting a simple, basic type of dBASE program. I am sure that many readers may be able to improve on it. How about sharing your ideas with the rest of the readers? Please send your programs to me for distribution.

Now, I would like to define some terms that apply to most data base programs, including dBASEII (For ease in typing the remainder of this article as well as reading it, I am going to use dBASE in place of dBASEII.):

1. DATABASE — organized collection of records. Your phone directory would be an example.
2. FILES — each database is made up of files.
3. RECORDS — a database file is made up of records and each record would be a line in the "file table". Back to the phone directory, the customer file has a record consisting of a name, address, and phone number on a line record.
4. FIELDS — each field is a column in the "file table" and has its own title (label). Thus, each name is a field. Each address a field and each phone number is a field. IMPORTANT! The field names exists only to help use the data in the "file table".
5. DATABASE STRUCTURE — is the directory to your data.

I did not use high-power, technical definitions for the above terms, but I think that you can visualize the terms in the database and that is what will count when it comes to understanding what data management is about! Did I hear you ask why do I need a database? This is a commonly asked question. The answer is that a database will allow you to ask questions from your file and quickly receive answers about the data. The more data that you have in the file, the more important the database model becomes!

The first step in creating a database model is to decide what infor-

mation you will need in the file. Entering information data into the model takes time and storing this information data uses disk storage capacity. Therefore, a database structure MUST be planned to provide room for all the data items that will be used without cluttering the disk storage with unnecessary information. This requires careful thought. A good practice is to minimize the number of fields in the database. Each field must be either filled in or left with blank spaces, both of which takes entry time and disk storage capacity. To save disk capacity and avoid complexity, I have kept the IRS 1040 FORM C EXPENSE RECORDS database structure simple, short and versatile, thus very adaptable to the readers needs.

When "creating" a dBASE program, we are actually writing a software program where dBASE will do most of the work. Therefore, this software program, like all programs, uses four design steps:

1. Design the program by determining what the program should do.
2. Write the program.
3. Run the program to test it.
4. Debug the program.

Let's take a look at each of these in detail:

1. Write down the general idea of the program on paper. Some ideas like the following (not in any special order):

   A. Miles.
   B. Time.
   C. Type of use.
   D. Purpose of use.
   E. Date of use.

2a. Design a data base structure for the program on paper. What is the name of each field? The name assigned can be up to ten characters long without spaces or punctuation except for a colon. What type of data will be stored in the field? dBASE allows three types:

   A. (C) character, for non-numeric data like names and addresses.
   B. (N) numeric, for numbers that will be used for math like miles and time.
   C. (L) logical, where the field is either true or false (not used in this program).

   Next, dBASE must know the width of each field. If it is a numeric, the number of decimal places must be specified.

   Here is an example of a dBASE Structure:

Field	Fieldname	Type	Width	Decimal Places
001	Lastname	C	20	0
002	Firstname	C	15	0
003	Date	C	8	0
004	Miles	N	4	0
005	Time	N	5	2
006	Expenses	N	10	2

2b. It is easier to write a program, if a reasonable facsimile of it is prepared in plain English (called pseudocode). This should specify the logic and series of events that will occur in the program. It does not, and should not, be in dBASE language. When it has been defined, translate the pseudocode to dBASE language and the program will be ready to enter into the computer.

2c. To create a data base program, load dBASE and type the word CREATE (or use the correct function key) next to the dBASE 'dot' prompt and press the return key.dBASE will ask—ENTER FILENAME:. Every program must have a unique filename up to eight characters long without spaces or punctuation. dBASE will add the three letter file extension. Now, type in — TEST and press return.

Next, dBASE starts the structure of the data base with a screen message as follows:

```
ENTER RECORD STRUCTURE AS FOLLOWS:
FIELD   NAME,TYPE,WIDTH,DECIMAL PLACES
001
```

Now, as an example type in the following:

```
001  LASTNAME,C,20,0                PRESS RETURN
002  FIRSTNAME,C,15,0               PRESS RETURN
003  DATE,C,8,0                     PRESS RETURN
004  MILES,N,4,0                    PRESS RETURN
005  TIME,N,5,2                     PRESS RETURN
006  EXPENSES,N,10,2                PRESS RETURN
007  PRESS RETURN
```

NOTE! For field #5, I choose 5 digits with 2 decimal places (the decimal point counts for one digit). The five characters allows for time up to 99.99 hours, for example. Rather than typing in field #7, just press Return to end the entries. dBASE will now ask if the user wants to — INPUT DATA NOW? Type N for "no" and the dot prompt will appear. The data base called TEST.DBF has been created on the default drive disk.

Check the structure for errors by typing:

```
LIST STRUCTURE (or use function key) PRESS RETURN
```

If it checks out OK, it is ready for data entry.

2d. At the dBASE dot prompt, type the following:

```
USE TEST                              PRESS RETURN
```

The USE command tells dBASE which data base to work with. The command to add new data to a data base is APPEND. So now, type the following:

```
APPEND                                PRESS RETURN.
```

This displays this form on the screen to be filled in:

```
RECORD NUMBER: 00001
LASTNAME:                    :
FIRSTNAME:                   :
DATE:          :
MILES:      :
TIME:       :
EXPENSES:              :
```

dBASE is now waiting for the user to fill in the spaces between the colons. Please type in the following data (Note — the cursor is waiting at the first space in the LASTNAME field.):

```
JONES                                 PRESS RETURN
THOMAS                                PRESS RETURN
12-15-84                              PRESS RETURN
127                                   PRESS RETURN
6.25                                  PRESS RETURN
1234.56                               PRESS RETURN
```

Did dBASE surprise you when you put in the date field? If the data fills in the field it sounds the bell and does the Return for you. dBASE is now waiting for the second record data to be added. If there are any errors to correct, they are easiest to change before leaving the APPEND mode! Check your en-

tries as you proceed. Next, add the following three records the same way as above:

DOE	JAMES	ANDREW
JOHN	BETTY	JACK
12-17-84	12-25-84	12-30-84
76	0	53
0	8.1	0
0	0	76.27

After the fourth record has been entered, dBASE will ask for the fifth record. To end the entries, press Return in place of adding data.

3a. There are two basic commands for viewing the contents of the data base:

A. DISPLAY — displays the data of the last record. DISPLAY will have better uses in other work we will be doing, so I will not make use of it now.

B. LIST — displays all the records in the data base that dBASE has been told to USE.

Type USE TEST and then type DISPLAY (follow commands by pressing Return key). The screen now shows the first record in the data base. Do you know why? We just called the file so the first record is on top!

3b. Type LIST and Return. The screen now shows the four records in the data base called TEST.

3c. The LIST command can be used many ways. For example, try the following commands and watch the screen display:

```
LIST LASTNAME,FIRSTNAME              PRESS RETURN
LIST FIRSTNAME,LASTNAME              PRESS RETURN
LIST LASTNAME,DATE,TIME              PRESS RETURN
LIST LASTNAME,MILES,EXPENSE          PRESS RETURN
```

Now, please try some commands that you can think of using LIST. Be creative and don't worry, you cannot do any harm. When you finish this session with dBASE, type the following:

```
QUIT                          PRESS RETURN
```

ALWAYS use the command QUIT prior to removing any disks from drives!

4. I will not discuss DEBUG portion of programming at this time. It is best explained by usage.

Now, I will start the project for this article!

First, I will determine what I want the program to do! I want daily records so I need a DATE field. I want to keep track of computer usage, car usage, travel, entertainment, business associated expenses and a short explanation of each record. So, next I will need a TYPE field to separate the various items I have just listed. I will want to record whether the items are business or nonbusiness. I will define a list of abbreviations that I propose to use in the TYPE field:

1. NCPTR — nonbusiness computer usage.
2. BCPTR — business computer usage.
3. NAUTO — nonbusiness car usage.
4. BAUTO — business car usage.
5. BTRVL — business travel other than by car.
6. $EXPS — dollar expenses that may require some adjustment by percent of business vs nonbusiness use.
7. BENTR — business entertainment and/or gifts.

I want to keep track of computer usage by TIME in hours or decimal parts of an hour in its field. For the car I will use a field called MILES. To keep a record of dollars spent, I will use a field called EXPENSES. Note, some adjustments may be required with some of these fields. For example, say the car works out to be 75% business usage. The car insurance expense business portion would be 75% of the total premium. I cannot do the adjustments until the end of the year because I will need the business usage vs nonbusiness usage to calculate the percentage. This will be easy to do with the records that I have kept in the data base! I will also have a field that I will call MEMO where I will enter a short explanation of the entry and a WHOM field description.

Second, I will design the data base structure. Each field in dBASE requires a record structure — the fields must be named, assigned a type (C or N), width defined, and in the case of numeric field the number of decimal places needed.

Now, load dBASE, if it has not been loaded, and at the dot prompt type this command:

```
CREATE                        PRESS RETURN
```

Next, when dBASE asks — ENTER FILENAME: — type in IRSBEXP5 and Return. Remember that the filename can have up to eight characters without spaces or punctuation and dBASE will assign the extension. My choice of a filename tells me that the program is for the IRS about Business EXPenses for 1985. Do you agree with this name? Use your own if you like.

dBASE will display the following:

```
ENTER RECORD STRUCTURE AS FOLLOWS:
FIELD   NAME,TYPE,WIDTH,DECIMAL PLACES
001
```

Type in the following entries:

```
001   DATE,C,2,0                 PRESS RETURN
002   TYPE,C,5,0                 PRESS RETURN
003   TIME,N,5,2                 PRESS RETURN
004   MILES,N,5,0                PRESS RETURN
005   EXPENSES,N,10,2            PRESS RETURN
006   WHOM,C,12,0                PRESS RETURN
007   MEMO,C,20,0                PRESS RETURN
008   PRESS RETURN
```

dBASE will ask — INPUT DATA NOW? Type N for "no" and the dot prompt will be displayed. dBASE has put a data base on the default disk called IRSBEXP5.DBF. Unless the user is a perfect typist and thus does not make errors, changes could be required. Find out by typing the following commands:

```
USE IRSBEXP5                  PRESS RETURN
LIST STRU                     PRESS RETURN
```

If changes are required, use the following command:

```
MODIFY STRU                   PRESS RETURN
```

Now using the cursor direction keys and return key, make the necessary changes. When the corrections have been completed, type CTRL-W to save the revised structure and get back to the dot prompt. To recheck, type — LIST STRU and Return. NOTE! dBASE will allow the user to use 4 letter commands like STRU for STRUCTURE. Use this to save typing and avoid changes for typing errors.

The Record Structure is waiting for data to be entered into the data base. With the dot prompt showing, let dBASE know what data base we want to add data to. The command for this — USE IRSBEXP5 and Return. Do not forget that the USE command tells dBASE which database we want. There could be many databases on the disk. The command to add new data to the database follows:

```
APPEND                                  PRESS RETURN
```

This brings the form that we will fill out to the screen as shown below:

```
RECORD NUMBER: 00001
DATE:   :
TYPE:       :
TIME:       :
MILES:        :
EXPENSES:               :
WHOM:                   :
MEMO:                           :
```

dBASE is waiting for the data entries in the blanks between the colons. The data cannot extend pass the colon but does not need to fill the blank space or have an entry (just press Return). When ready for the next field press Return. If the data should fill the blank space, dBASE will sound the "BELL" and automatically go to the next blank field! (This may surprise you the first few times.) I will list the sample data that I would like to have you enter into this database below:

```
02 BCPTR  3.3     0          0   SPDSHT COR ENTER TAX RCDS FOR 1984
02 BAUTO  2.6    13          0   BANK/S&L   BANKING-SAN JUAN CAP
02 BAUTO  1.3     8      29.83   SPDSHT COR SUPPLIES-SAN CLEMENTE
03 BCPTR  8.1     0          0   CoCal Corp YR-END-CLOSING
04 BCPTR  7.8     0          0   BODEGA Inc YR-END-CLOSING
05 NAUTO    0    21          0   SELF       SHOPPING
07 BAUTO 10.1   121          0   CoCal Corp CONFERENCE-AGOURA
07 BENTR  1.1     6      11.97   CoCal Corp LUNCH MTG.
08 BCPTR  6.2     0          0   DR. FORD   YR-END-CLOSING
08 BAUTO   .5    20          0   DR. FORD   CONFERENCE-MISSION VIEGO
08 BENTR  1.0     0      10.43   DR. FORD   LUNCH MTG.
09 BAUTO  6.3   162          0   JAMES INV  NEG.CONTRACT-WESTWOOD
09 BENTR  1.5     6      11.91   MR. JAMES  LUNCH MTG.
10 BCPTR  7.7     0          0   JAMES INV. CREATE DBASE PROGRAM
11 BAUTO  6.6   167          0   JAMES INV. CONFERENCE-WESTWOOD
11 NENTR  1.5     6      11.21   MR. JAMES  LUNCH MTG.
12 BCPTR  3.3     0          0   SPDSHT COR POST EXPENSES
12 NAUTO    0    12          0   SELF       SHOPPING
14 BCPTR  4.4     0          0   SPDSHT COR BUS. CORRESPONDENCE
14 BAUTO  2.8    21      18.87   SPDSHT COR P.O. & SUPPLIES
15 BAUTO  6.5   137          0   DR. STEIN  REV. ACCTG SYSTEM-S.D.
15 BENTR  1.1     3       9.71   DR. STEIN  LUNCH MTG
16 BCPTR  6.7     0          0   DR. STEIN  REVISE ACCTG SYSTEM
16 BAUTO  1.5    17       1.70   SPDSHT COR POST OFFICE-S.C.
17 BTRVL  9.3    68     227 60   JAMES INV. AIR TICKET-HOUSTON
18 BTRVL 10.2    69     271.81   JAMES INV. REV INV EXPS.CAR,MEALS,MOTEL
21 BCPTR  7.7     0          0   JAMES INV. POST HOUSTON INV EXPENSES
22 BAUTO  6.6   166          0   JAMES INV. REVIEW HOUSTON INV-WESTWOOD
22 BENTR  1.3     4      12.77   MR. JAMES  LUNCH MTG
23 NAUTO    0    22          0   SELF       DENTIST-D.P.
23 NCPTR  4.3     0          0   SELF       CORRESPONDENCE
24 BAUTO  4.4    38          0   L.A. CPA'S INSTALL ACCTG SYSTEM-EL TORO
24 BCPTR  4.1     0          0   L.A. CPA'S MODEM TESTING
25 BCPTR  5.6     0          0   SPDSHT COR PREPARE INVOICES
25 BAUTO  2.1    11      18.80   SPDSHT COR P.O. & CAR GAS
28 BAUTO  4.7    31          0   MED-VAC    SET UP NEW ACCT
28 BCPTR  4.0     0          0   MED-VAC    SET UP MODEM
29 BCPTR  4.2     0          0   MED-VAC    DEBUG NEW ACCOUNT
29 BCPTR  4.0     0          0   SPDSHT COR REPAIR & CLEAN PRINTER
30 NAUTO    0    23          0   SELF       DENTIST-D.P.
30 $EXPS  3.8    11      62.83   SPDSHT COR BUY RIBBONS & FORMS
30 BCPTR  2.1     0          0   SPDSHT COR INSTALL RIBBON & TEST PTR.
31 BCPTR  7.7     0          0   SPDSHT COR POST & CLOSE MONTH
```

After the last data item has been added, type Return rather than entering a new DATE. dBASE will leave the APPEND mode and return the dot prompt. Now, type the following two commands:

```
USE IRSBEXP5                    PRESS RETURN
LIST                            PRESS RETURN
```

All the data that was entered above will be displayed. Please try some of the LIST and DISPLAY commands as discussed above. Remember, when you want to finish the session, type — QUIT and press Return. ALWAYS do this prior to removing any disks from their drive!

Next, I will show some "search" the database commands. If dBASE is not loaded, do this now. Then, type — USE IRSBEXP5 and Return. I will begin with the LIST FOR command. Let's suppose that I would like to know what I did on the 8th of the month. I would type the following command:

```
LIST FOR DATE = '08'            PRESS RETURN
```

Note that the "=" means equal to. What display did you find? If I want to display only certain fields for this DATE, I can specify those fields in the command as follows:

```
LIST FOR DATE = '08' DATE,WHOM,MEMO     PRESS RETURN
```

The display will contain the requested fields. Let's say I want to know how many days this month I spent over $25.00 for expenses. That is easy with this command:

```
LIST FOR EXPENSES > 25.00       PRESS RETURN
```

This command tells dBASE to search and display the entries where expenses are "greater than" $25.00. What was displayed?

NOTE! Always use the same order even though the commands are different. Use LIST FOR fieldname,operator,'condition',plus fields, followed by Return. That is the proper SYNTAX for the LIST FOR command and any deviation may cause an error.

I would like to change the looks of the display, so I will use the TRIM function. Here is an example:

```
LIST DATE,TYPE,TIME,MILES,EXPENSES,TRIM(WHOM),TRIM(MEMO)
     PRESS RETURN
```

Did you find the display changed? dBASE pads the contents of every field with trailing blank spaces so that the information in each field forms vertical columns. The TRIM function removes these trailing blank spaces. Its syntax always requires the parentheses — TRIM(fieldname). This command really works well with people's names. The database in this article does not have a good example for this function but I wanted the readers to know about it.

Here is a search problem to look out for. Jones and JONES are not the same from dBASE point of view. dBASE has a uppercase function (!) that will display all lowercase letters in a character field in uppercase. An example would be as follows:

```
LIST !(lastname)                PRESS RETURN
```

Jones and JONES will be displayed with this command. dBASE can also be asked to display everyone whose lastname is not Jones by using the "not equal" dBASE sign (#) as follows:

```
LIST FOR !(LASTNAME)#'Jones'LASTNAME,FIRSTNAME   PRESS RETURN
```

dBASE has a LOCATE command for searching for a position of a record based upon a desired characteristic. Since LOCATE will not display it but only find it, use DISPLAY command with LOCATE command to find what dBASE has found. Note, the FOR statement is required with the LOCATE command. An example:

```
LOCATE FOR WHOM = 'DR. STEIN'       PRESS RETURN
```

dBASE will only display the RECORD NUMBER. This is not useful, but if we type the following:

```
DISPLAY             PRESS RETURN
```

dBASE will now display what has been LOCATED. What did you see? Next, type the following commands:

```
CONTINUE            PRESS RETURN
DISPLAY             PRESS RETURN
```

What is displayed? Try the CONTINUE and DISPLAY several times and watch the screen. Did the "END OF FILE ENCOUN-

TERED" message show up? How about the reader trying some of their variations? Also, try the following commands (follow each command with Return):

```
LIST FOR WHOM = 'DR. STEIN' .OR. WHOM = 'MR. JAMES'
LIST FOR WHOM = 'DR. STEIN' .AND. WHOM = 'MR. JAMES'
```

How did the two displays differ? Why? I used two "operators" — .OR. and .AND.! NOTE: The periods before and after the operator must be there. The .OR. operator broadens the search, since only one of the conditions must be met for dBASE to display data on the screen. Otherwise, the .AND. operator narrows the search, since both conditions must be met to display data. An example of this follows:

```
LIST FOR TYPE = 'BENTR' .AND. WHOM = 'DR. STEIN'      PRESS RETURN
```

And last but not least, there is a search where we need to know if a field roughly matches what we know about the field. For example, we want to search for JAMES, no matter if it is DR. JAMES, MS. JAMES, MR. JAMES, etc. If I type LIST or LOCATE FOR WHOM = 'JAMES', no match would be found because all the JAMES are embedded in the WHOM fields. The command must say — Display all records that have the name JAMES embedded in the WHOM field. That sounds complicated, but not to dBASE because it has a ($) function that will find embedded words. An example would be:

```
LOCATE FOR 'JAMES'$WHOM          PRESS RETURN
DISPLAY                          PRESS RETURN
```

That worked pretty well. It displayed a record with JAMES in it. The CONTINUE and DISPLAY commands can carry the example further. Keep in mind that our search commands work with both LOCATE and LIST commands, so try this:

```
LIST FOR 'CPTR'$TYPE .AND. 'JAMES'$WHOM      PRESS RETURN
```

Note, the DISPLAY command is not required with the LIST FOR command. There are some other variations. The .OR. and .AND. search operators can be combined with our examples.

```
LIST FOR 'CPTR'$TYPE .AND. ('STEIN'$WHOM .OR. 'JAMES'$WHOM)
```

What did dBASE display? Why? dBASE has a built-in exact parameter which, unless specifically called for, is in the OFF mode. It is turned ON with the following command:

```
SET EXACT ON              PRESS RETURN
```

This would change the results of the search in the cases where the match is not exact. This is useful if you want the database entries to be all uppercase for example. I am sure that you can see how this would work.

I could spend the rest of this article on search possibilities, but the reader will learn better by experimenting. Experience is the best teacher. Don't be intimidated by ERRORS, they teach you how to avoid them in the future! Have you found dBASE to be difficult so far in this article? It is not! It requires usage. I will not get into the more difficult programming in this article, but I do want to discuss some other dBASE commands.

In using our IRSBEXP5 database or any other database, it is often necessary to add new data to the database as the data becomes available. At some point, it will be handy to arrange the records into some other meaningful order. dBASE has two commands that will sort the database — SORT and INDEX!

When the SORT command is used, dBASE requires that we create a new database file to store the sorted records. When the SORT is completed, COPY can be used to put the contents of the sorted database back into the original database file. There could be occasions when the new sorted file would be kept for flexibility. Here is an example. Load dBASE, if it is not loaded, and type — USE IRSBEXP5 and Return. Now, type the following:

```
SORT ON TYPE TO EXPTYPE          PRESS RETURN
```

Wait until dBASE shows — SORT COMPLETE. Now type LIST and Return. Notice the changes in the database! What happened? The records do not look sorted by TYPE. Remember, we sorted to a new file called — EXPTYPE. So, type the following commands:

```
USE EXPTYPE                      PRESS RETURN
LIST                             PRESS RETURN
```

Does the display show the records sorted by the TYPE field? dBASE required that the data be sorted to the new file EXPTYPE. dBASE must be told to look for the new file by using — USE EXPTYPE. The default disk will now have two files for this database. If the original file is no longer needed and disk space was important, type the following:

```
COPY TO IRSBEXP5                 PRESS RETURN
```

The EXPTYPE file could be deleted. DO NOT DO THIS. We will use the two files. Sorting in this fashion is useful, but there are disadvantages. First, the two sorted files waste disk storage space. Second, every record number changes in the new sorted database, so the user can't be sure of an individual record number in applications where this might be desirable. Try some other SORTs for practice!

The INDEX command provides a much quicker and more efficient method of reordering the records. The two commands look similar, but the approach is different. An example would be helpful. Again, the INDEX command will be used to reorder the TYPE field and I will use a file called TYPEEXP. Type the following commands:

```
USE IRSBEXP5                     PRESS RETURN
INDEX ON TYPE TO TYPEEXP         PRESS RETURN
LIST                             PRESS RETURN
```

The display shows the records have been arranged in proper order, but the record numbers have remained the same. Also, it is not necessary to use the COPY command because dBASE was asked to INDEX to TYPEEXP which is a data file (Not a database file.). It is a special file, an INDEX FILE named TYPEEXP.NDX. Its contents look like a book index with a list of keywords in alphabetical order and record numbers. To check this out, type — QUIT and Return. At the DOS prompt, type DIR and compare the size of the different files. Also, type — TYPE TYPEEXP.NDX and Return. What did the file look like? Notice the size of the TYPEEXP.NDX file. It is much smaller and uses less disk space!

Reload dBASE and try another example:

```
USE IRSBEXP5                     PRESS RETURN
INDEX ON TIME TO TIMEEXP         PRESS RETURN
LIST                             PRESS RETURN
```

After the indexing, when listing, dBASE automatically uses the information from the last INDEX file to determine the proper order to display the records of IRSBEXP5 database. The records in IRSBEXP5 database are still in the same original order; the index file, however, tells dBASE what order to display the records. We now have the following files on the default disk:

1. IRSBEXP5.DBF
2. TYPEEXP.NDX
3. TIMEEXP.NDX

Now, here is the real advantage of indexing. If I want to display

the database sorted by TYPE, I do not have to INDEX or SORT again. Just type the following:

```
USE IRSBEXP5 INDEX TYPEEXP        PRESS RETURN
LIST                              PRESS RETURN
```

Or, if the user wants indexing on TIME field, type this:

```
USE IRSBEXP5 INDEX TIMEEXP        PRESS RETURN
LIST                              PRESS RETURN
```

Furthermore, the index (.NDX) files take up so much less room on the disk that it is not necessary to DELETE any of them until the user is sure that they will never be needed. The INDEX command will do nearly everything that the SORT command will do, faster and more efficiently. Please practice it! Did I hear you ask about SORTS within SORTS? Let's say we want to really shape up the IRSBEXP5 database by INDEXing by TYPE and then each TYPE by TIME. Did you say just use two SORTs? Sorry, I do not need to do that! I will do one INDEX on two separate fields. [Numeric fields, such as TIME, must be converted to character type. TIME is a numeric field with 5 positions and it has 2 decimal places. The STR function, which I will cover in later articles performs the conversion — STR(TIME,5,2).] The example follows:

```
USE IRSBEXP5                                    PRESS RETURN
INDEX ON TIME + STR(TIME,5,2) TO BOTHEXP        PRESS RETURN
LIST                                            PRESS RETURN
```

Check the listing. Did it sort TIME within TYPE? That was easy! Readers, please try some of your own ideas. Be sure to try one with three fields!

Through all of the above, I have stayed away from EDIT, which means to change existing data. This command is used when we need to change a record within the database because of some change that comes up, or when we find a mistake that happened when entering the original data. There could be many other reasons. Editing is a rather easy task with dBASE if we know the record number where the change is needed. If the database is large, this could make it a hard problem. Let's suppose that DR. STEIN turns his practice over to DR. SMITH. Would we know what records require changes? We have studied LIST FOR and know it would search for such records. Try it with the following:

```
USE IRSBEXP5                      PRESS RETURN
LIST FOR WHOM = 'DR. STEIN'       PRESS RETURN
```

This will display the records that include DR. STEIN. Write down the record numbers. Now, to change these records from DR. STEIN to DR. SMITH type the following:

```
EDIT record number                PRESS RETURN
```

The structure for that record number will be displayed on the screen. Using the cursor control keys, move the cursor under the first letter of STEIN and type SMITH (which removes STEIN). When the correction has been made, save the new data by typing CTRL-W. Repeat for each record number that requires changes.

Did you say dBASE ought to have a better way? It does! dBASE has a BROWSE command that allows the user to scroll through the database, both vertically and horizontally. dBASE will always display as much of the database as will fit on the screen, but this is one command that provides the horizontal scroll that can be really useful on some databases. Try the BROWSE mode by typing the following:

```
USE IRSBEXP5                      PRESS RETURN
BROWSE                            PRESS RETURN
```

The control key commands for use with BROWSE are:

CRTL-E	MOVES CURSOR LEFT 1 FIELD, OR UP 1 LINE IF THERE ARE NO MORE FIELDS TO THE LEFT
CRTL-X	MOVES CURSOR RIGHT 1 FIELD, OR DOWN 1 LINE IF THERE ARE NO MORE FIELDS TO THE RIGHT
CRTL-U	DELETES/UNDELETES A RECORD
CRTL-Q	QUITS BROWSE
CTRL-W	SAVES BROWSE CHANGES

What information do you find displayed on the screen? Move the cursor to whatever data should be altered. To change data of a field or record, position the cursor to where the change is needed and type the new data on top of the old data. (I should warn you to always keep a backup copy of the original database just in case of an error.) To exit after the changes have been made use CRTL-W, and if you want to quit without making permanent changes use CRTL-Q. If the user wants to limit the data displayed try this example:

```
BROWSE FIELDS DATE,WHOM           PRESS RETURN
```

Notice that the field names must be separated by commas and only the listed fields will be displayed and can be edited!

In cases where the user wishes to make the SAME change to numerous records in the same database, called global editing, dBASE provides a CHANGE and a REPLACE command. CHANGE can be best explained with an example. Let's change DR. STEIN to DR. SMITH in all the WHOM fields. I will ask dBASE to do this with the following command:

```
USE IRSBEXP5                                PRESS RETURN
CHANGE FIELD WHOM FOR WHOM = 'DR. STEIN'    PRESS RETURN
```

DBASE will display DR. STEIN for each record where it appears in the WHOM field one at a time. dBASE will ask CHANGE and TO. Do you find this? Make the change and dBASE will step through the database showing each DR. STEIN record. This is a slow but safe way to perform a global edit.

Next, we can do a faster global edit by using REPLACE command. It is similar to CHANGE in that they both change contents of a given field. REPLACE does not ask questions. It MAKES the changes without any entries. Let's change the DR. SMITH back to DR. STEIN by typing the following:

```
REPLACE ALL WHOM WITH 'DR. STEIN' FOR WHOM = 'DR. SMITH'
     PRESS RETURN
```

This command tells dBASE, "Anywhere there is a DR. SMITH in the WHOM field, replace it with DR. STEIN." Again, keep a back-up copy of the original!

What if we want to eliminate some record(s) in the database? The DELETE command is used to MARK records for deletion. It does NOT delete the record! The PACK command is use to PERMANENTLY delete the record. If a record is MARKED for deletion, but you wish to leave it in the database, there is a RECALL command used like this:

```
DELETE RECORD 5                   PRESS RETURN
LIST                              PRESS RETURN
```

Did you find record 5 MARKED? Now, type:

```
RECALL RECORD 5                   PRESS RETURN
LIST                              PRESS RETURN
```

What did you find out about record 5? Now, type:

```
DELETE RECORD 5                   PRESS RETURN
PACK                              PRESS RETURN
LIST                              PRESS RETURN
```

What happened to record 5? Next, type — DELETE FILE EXPTYPE.DBF and Return.

Next, I will show how to modify the database structure. The command for this is — MODIFY STRUCTURE. Be sure to always have a backup file, as this command will destroy ALL data in the database! I always feel a little queasy when using this command on a long and important database. Type the following:

```
USE IRSBEXP5                    PRESS RETURN
COPY TO TEMP                    PRESS RETURN
MODIFY STRU                     PRESS RETURN
```

dBASE will display the warning:

```
MODIFY ERASES ALL RECORDS IN A DATABASE.  PROCEED? (Y/N):
```

Type "Y" and the revised structure will be displayed. Move the cursor down to 008 and type in MTH for month and type CRTL-W to save. If you LIST, nothing will be displayed just as dBASE warned. Type the following:

```
APPEND FROM TEMP                PRESS RETURN
```

The message RECORDS ADDED and the dot prompt will show. Try LIST again. Did the records return? We now have MTH for month with a 2 character field.

Let's try a few interesting things. Would you like to know how much money we spent for expenses? Type — SUM EXPENSES and Return. How about how much time we used? Type — SUM TIME and Return. You try miles. How about this? Type — SUM EXPENSES FOR TYPE = 'BCPTR' and Return. You try some of these.

Finally, I have gotten to the point where dBASE can really shine without much effort on our part. Let's create a Report.

First, create a Report Form. The procedure is easy. Name the report with up to 8 characters without spaces or punctuation. dBASE will store this form on the default disk with our name and add the extension (.FRM). I named it FRMC0185. Type the following:

```
USE IRSBEXP5                    PRESS RETURN
REPORT FORM FRMC0185            PRESS RETURN
```

dBASE will start asking these questions:

```
ENTER OPTIONS,M=LEFT MARGIN,L=LINES/PAGE,W=PAGE WIDTH
     TYPE IN--M=1,L=55,W=70
PAGE HEADING (Y/N)?
     TYPE IN--Y
ENTER PAGE HEADING:
     TYPE IN--JANUARY, 1985 FORM C EXPENSES
DOUBLE SPACE REPORT? (Y/N)
     TYPE IN--N
ARE TOTALS REQUIRED? (Y/N)
     TYPE IN--Y
SUBTOTALS IN REPORT? (Y/N)
     TYPE IN--Y
ENTER SUBTOTAL FIELD:
     TYPE IN--TYPE
SUMMARY REPORT ONLY? (Y/N)
     TYPE IN--N
EJECT PAGE AFTER SUBTOTALS? (Y/N)
     TYPE IN--N
ENTER SUBTOTAL HEADING:
     TYPE IN--FUNCTION
```

Next, dBASE wants the column formats:

```
COL   WIDTH,CONTENTS
001   6,TYPE
ENTER HEADING:  TYPE
002   6,DATE
ENTER HEADING:  DATE
003   6,TIME
ARE TOTALS REQUIRED? (Y/N)   Y
ENTER HEADING:  TIME
004   7,MILES
ARE TOTALS REQUIRED? (Y/N)  Y
ENTER HEADING:  MILES
005   10,EXPENSES
ARE TOTALS REQUIRED? (Y/N)  Y
ENTER HEADING:  EXPENSES
006   12,WHOM
ENTER HEADING:  WHOM
007   20,MEMO
ENTER HEADING:  MEMO
```

By pressing return at 008, dBASE will know that the Report Form is complete and it will be displayed. One disadvantage of this Report is that it is not sorted. We can easily correct this, as before, by INDEXing the database. Type the following:

```
INDEX ON TYPE TO IRSBEXP5       PRESS RETURN
REPORT FORM FRMC0185            PRESS RETURN
```

How does that look? Many other Reports can be done with this database by varying the Report Form and the INDEXing. Please try some!

For Income Tax purposes, the user can either prepare one of these Reports for each month or keep this Report going for the twelve months. We added the MTH field for month data if you remember! I am sure the readers can see how to have dBASE do a Report for each month and how to get a yearly Report by using the MTH field and INDEXing on MTH. I discussed SUM above, so if we want to calculate the percentage of business use vs non-business use of the computer or car, the SUM command will provide what is needed for the calculation. I will let the reader do this.

dBASE has a COUNT command that I should discuss. It is used like this:

```
USE IRSBEXP5                    PRESS RETURN
COUNT FOR TYPE = 'BCPTR'        PRESS RETURN
```

What did you find? Type this to check — LIST FOR TYPE = 'BCPTR' and Return. Now, what did you find? Do they check out?

dBASE provides a number of ways to do customized reports. I will do some in later articles. With a little practice and this article, the reader should be able to create a Report that will impress the IRS! I would transfer the results to Form C and attach a copy of the dBASE Report to Form C.

The same principles that I have established in this article can be used for many applications, even though this example is simple — Income Records, Billing, Household Expenses, etc. The most important thing that I hope the reader discovered from this article is that dBASE is not as difficult to use as some would have us believe! Because dBASEII lacks menus, using it is not as intuitive, but a little practice pays off very quickly. If you write to the author with questions, ideas, etc., please enclose a stamped, self-addressed business size envelope if you expect a reply. I invite reactions.

Current Local

Clubs

AK, Eagle River
Alaska HUG
206 E Firweed Ln, Suite 208
Anchorage, AK 99503
907 276-5917 Group Size 15
Contact Person: Roger Pickels or Ben Sevier
RBBS 9pm-9am Pacific time 907 276-5917

AK, Ft. Greely
Club no longer in existance

AL, Birmingham
BEARHUG (Birmingham HUG)
C/O Jack Goertz Rte 19, Box 248
Birmingham, AL 35244
205 991-5519 Group Size 20
Contact Person: Jack Goertz
Meeting time and place varies

AL, Huntsville
Club no longer in existance

AL, Mobile
MOBHUG (Mobile HUG)
354 Teakwood Drive
Satsuma, AL 36572
205 675-9742 Group Size 22
Contact Person: Bob Small
Usually meet the last Friday of the month

AL, Montgomery
HUG of Montgomery
2948 Willow Lane Drive
Montgomery, AL 36109
205 272-6964 Group Size 50+
Contact Person: Ronald Travis or John Guenther
Meet 1st Tues 7:30 pm at AF Logs Mgt Ctr

AR, Fayetteville
TRY-STATE HUG
2617 Country Way
Fayetteville, AR 72701
501 521-4818 Group Size
Contact Person: Gil Hoellerich
Meet 3rd Sat 1pm at NW Voc-Tech School Springdale
Group just starting, but growing!

AR, Little Rock
Little Rock H/ZUG
C/o David Schade 113 Dakota
Jacksonville, AR 72076
501 988-5273 Group Size 30
Contact Person: David Schade
Meet 2nd Sat 12:00 noon at AR College of Tech.
Phone for Fri meeting time & place

AZ, Phoenix
PHUG (Phoenix Heath Users' GP)
C/o Will Summers PO Box 37783
Phoenix, AZ 85069
Group Size 75
Contact Person: Will Summers, President
2nd Tues at 7:00 pm at Phoenix HEC
Membership $5 initiation $12/year

AZ, Sierra Vista
HuacHUGa
1964 Viola Dr c/o Gerald King
Sierra Vista, AZ 85635
602 459-2119 Group Size 30
Contact Person: Gerald King
Meet monthly at homes of members

AZ, Sun City
Sun Cities HUG
12739 Mesa Verde Drive
Sun City West, AZ 85375
602 584-9181 Group Size 12+
Contact Person: Andrew Weilkeiwicz
Meet at 1st Fed Savings and Loan Meeting Room
Meet 3rd Fri at 1:00pm La Ronde Ctr

AZ, Tucson
SUNHUG (Tucson HUG)
7109 E Broadway
Tucson, AZ 85710
602 325-0096 Group Size 40
Contact Person: Allan Anderson
Meet even months first Sunday 2:00 pm Tucson HEC
Meet odd months first Tues. 7:00

CA, Anaheim
ANAHUG (Anaheim HUG)
330 E. Ball Road
Anaheim, CA 92805
714 529-7535 Group Size 103
Contact Person: Al Solomon, President
3rd Thursday 7:30 PM at HEC
BB 714 774-7860

CA, Campbell
San Jose HUG
2350 S Bascom Avenue
Campbell, CA 95008
408 377-8472 Group Size 70
Contact Person: Gerlene York, Sec.
Meet 1st Wed from 7-8pm at the Campbell HEC

CA, El Cerrito
ECHUG (El Cerrito HUG)
6000 Potrero Avenue
El Cerrito, CA 94530
415-236-8870
Contact Person: Alan Biocca
Meet 4th Tues

CA, El Monte
ETUG (ET/ETA 3400 Users Group)
11231 Oak Street
El Monte, CA 91731
Group Size 100
Contact Person: Charles Van Dyke
Newsletter 4 times year

CA, Fresno
FresHUG (Fresno HUG)
4833 East Santa Ana
Fresno, CA 93726
209-291-6258 Group Size 12
Contact Person: Harlen Collins

CA, Glendora
Southern CA H11 Users Group
430 W. Highland Avenue
Redlands, CA 92373
714-886-4766 Group Size 40
Contact Person: Dr. M.J. Di Girolamo
Meets at 625 E. Palm, Glendora, CA

CA, Lancaster
AVZUG (Antelope Valley ZUG)
PO Box 4619
Lancaster, CA 93535
805 942-7576 Group Size 25
Contact Person: Jerry Jones
Meet 4th Sat 1:00 pm at United Methodist Church
Club newley formed 10/84

CA, Los Angeles
Los Angeles HUG
4469 E Olympic Blvd
Los Angeles, CA 90023
213 248-1580 Group Size 90
Contact Person: Doug Holser
1st Thursday 7:00 PM at HEC
BB 213 749-8442

CA, Los Angeles
LAETUG (Los Angeles ET3400 GP)
2309 S Flower
Los Angeles, CA 90007
213 749-0261 Group Size
Contact Person: Gilbert Murillo
Other contact Charlie at 213 443-2237
Contact for meeting time and place

CA, Monterey
Naval Pstgrd Sch Hobby Com Clb
C/o Dave Smith 376 Bergin Dr #F
Monterey, CA 93940
408 373-4202 Group Size 175
Contact Person: Dave Smith, President

CA, Pomona
Pomona HUG
1555 N Orange Grove Avenue
Pomona, CA 91767
714 985-5303 Group Size 102
Contact Person: Herb Friedman, President
Meet 4th Thursday each month at 7:30p.m. at HEC
BB 714 624-5191

CA, Redding
Redding Heath Users' Group
22526 Bridlewood Lane PO BOX 370
Palo Cedro, CA 96073
916 547-3461 Group Size 6
Contact Person: Dave Ballard
Meet monthly various locations

CA, Redwood City
BAHUG Bay Area HUG
2001 Middlefield Road
Redwood City, CA 94063
415 365-8155 Group Size 219+
Contact Person: Bob Bance, Sec.
Meet 4th Tues 7:00 pm at HEC

CA, Riverside
Tri-HUG
5705 Via Sotelo
Riverside, CA 92506
714-683-2929 Group Size 20
Contact Person: Kenny Adcock

CA, Sacramento
SHUG (Sacramento HUG)
7607 Maran Ave
Sacramento, CA 95820
916 739-8074 Group Size 55
Contact Person: Delven Hamric, President
Meet 2nd Wed 7:30pm at Sacramento HEC
Send mail ATTN: Delven Hamric

CA, San Diego
San Diego HUG
PO Box 33046
San Diego, CA 92103
619 484-5120 Group Size 170
Contact Person: Clem Pepper
Meet 1st Wed 7:00pm at Kearny-Mesa Rec. Ctr.
Monthly guest speakers, Newsletter

CA, San Diego
San Diego Comp Soc H/Z SIG
PO Box 81444
San Diego, CA 92138
619 698-2945 Group Size 127
Contact Person: Jon Melby, Chairperson
Meet 1st Sat 10:00 am at La Mesa HEC
24hr. RC System 619 461-5117

CA, Santa Maria
4168 Glenview Drive
Santa Maria, CA 93455
805 937-6938 Group Size 24
Contact Person: Raymond S. Isenson
Meet 1st Mon 7:00 pm at Vandenburg Air Force Base
Meet in Base Library

CA, Santee
WCHUG (West Coast HUG)
C/o Van Christopher
10784 Magnolia Ave 2H Santee,CA 92071
619 449-7298 Group Size
Contact Person: Van Christopher
Monthly Newsletter - 12 issues/$15.00
Software Lib ZDS Bulletin Bd Notebk

CA, Visalia
Visalia HUG
5230 San Joaquin Ave
Visalia, CA 93277
209 739-7228 Group Size 3
Contact Person: Peter Shkabara
Meeting time and place not on regular schedule

CA, Woodland Hills
LUVAHUG
22504 Ventura Blvd.
Woodland Hills, CA 91364
818 906-7425 Group Size 80
Contact Person: Rick Gaitley
Meet 2nd Thurs. 7:00 pm at HEC

CO, Colorado Springs
CSHUG (Colorado Springs HUG)
C/o Innovative Computer Service
125 E Arvada St Colorado Spg,CO 80906
303 632-3019
Contact Person: Richard Evers
Group is reforming - meeting times not set
24hr BB 303 598-4662

CO, Denver
DENHUG (Denver HUG)
PO Box 449 Contract Station 22
Denver, CO 80221
303 426-7404 Group Size 160
Contact Person: Bob Eson Sec/Tres
BB 303 423-3224 24hrs Support newsltr exchange
Meet 2nd Monday 7:00 pm at HEC

CO, Denver
ZUG (Zenith Users' Group)
AFAFC/MPA Bldg 444 Lowry AFB
Denver, CO 80279
303 370-7153 Group Size 75
Contact Person: S.L. Brantley, President
Meet 2nd Wed at Gilchrist Building

CO, Ft. Collins
FT.HUG (Fort Collins HUG)
3317 Buckskin Trail
Laporte, CO 80535
303 493-2987 Group Size 30
Contact Person: Charles McJilton
Meet 2nd Wed 7:00 pm at members homes
BB expected late 84. $10.00 dues, NL

CT, Avon
CONNHUG (Connecticut HUG)
C/o Bob Conlon 71 Pardee
Bristol, CT 06010
203 589-3824 Group Size 50+
Contact Person: BOB Conlon, President
Meet at Greater Hartford Chapter Amer. Red Cross
Meet 1st Wed at 7:00 pm

CT, Mystic
Mystic ZDS/HUG
PO Box 279
Mystic, CT 06355
203 536-6953 Group Size 45
Contact Person: Larry Moxon
Meet last Thurs 7:00pm at 11 Allen St. Mystic, CT

FL, Cocoa Beach
Brevard Heath Users' Group
680 Java Road
Cocoa Beach, FL 3293l
305 783-6352 Group Size 50
Contact Person: Gene E. Stillman
Meet last Sun of each mo at 6:00 pm
Meet at Patrick AFB, Comet Rec. Ctr.

FL, Fort Myers
SWFHUG (Southwest Florida HUG)
PO Box 05-0037
Tice, FL 33905
813 334-6190 Group Size 30
Contact Person: Robert Sloat
Meet 2nd Wed 8:00 pm
Meet at J. Hamilton Welch Academy

FL, Fort Walton
NWFHUG (NorthWest Florida HUG)
38 Berwick Circle
Shalimar, FL 32579
904 651-2970 Group Size 45
Contact Person: Allan White, President
Meetings 2nd Wed at DATATEC Inc. 7 PM

FL, Jacksonville
Jacksonville Users Group Inc.
8262 Arlington Expressway
Jacksonville, FL 32211
904 264-9763 Group Size 150
Contact Person: Harry Walker
Meet 1st Wed at 7:00 pm at Englewood HS Library
BB 904 725-4995 24 hrs

FL, Miami
Miami Amateur Computer Club
C/o Heathkit Electronics Center
4705 W 16th Ave Hialeah, FL 33012
305 823-2280 Group Size 85
Contact Person: Charlie Robertson, Secretary
Meet 2nd Thurs each month 7:00 pm at HEC
BB (305) 823-2281

FL, Orlando
HUG of Central FL Computer Sc.
121 Talmeda Trail
Maitland, FL 32751
305 644-6847 Group Size 20
Contact Person: Joseph Walker, President
Meet 4th Wed at Kane Furniture (address below)
Intersection of 436 and Red Bug Road

FL, Pensacola
PENSAHUG
221 E. Government
Pensacola, FL 32501
904 438-7805 Group Size 15
Contact Person: Gerry Harris
Meet 2nd Sun each month 2:00 pm at above address
Meet at Professional Business Sys.

FL, Plantation
PHUG Plantation HUG
7173 W Broward Blvd
Plantation, FL 33317
305 791-7300 Group Size 50
Contact Person: Paul Price
Meet 2nd Thurs 8-10 pm at HEC
BB 305 791-7302 24hrs on H/Z100

FL, Tallahassee
Tally HUG
C/o LPLC PO Box 4276
Tallahassee, FL 32315-4276
904 562-1412 Group Size 10
Contact Person: Bill Hill
Meet 3rd Tues at Leon Co. Resource Center
In the lower level Northwood Mall

FL, Tampa
Al Lynch HUG
PO Box 151762
Tampa, FL 33684-1762
813 885-1923 Group Size 80
Contact Person: Jim Cottingham, Secretary
Meet 1st and 3rd Wed 7:30 pm at Tampa HEC
Dues $10.00 year

GA, Atlanta
HUG-GA (Georgia HUG)
2775 NE Expressway Apt 0-3
Atlanta, GA 30345
404 449-3328 Group Size 30
Contact Person: Tom Campbell
Meet 2nd Mon at 7:00 pm at HEC
BB 404 252-4345 24 hrs.

GA, Warner Robins
MGHUG Middle Georgia H/ZUG
PO Box 1989
Warner Robins, GA 31099-1989
912 923-1353 Group Size 30
Contact Person: Gary H Arnett, President
Meet 3rd Tues at Nola Brantley Public Library

GA, Augusta
CSRA Computer Club
PO Box 284
Augusta, GA 30903
803 648-3603 Group Size 10
Contact Person: Dave Howard
Meet 4th Wed-location rotates
BB 803 279-5392

HI, Hilo
BIHUG (Big Island HUG)
P.O. Box 4271
Hilo, HI 96720
808 959-8985 Group Size 20
Contact Person: R.A. Curtis
Meet 1st Thurs 7:00 pm at HELCO Conference Room
BB 808 961-4818

HI, Honolulu
HUGH (HUG Hawaii)
1255 Nuuanu Avenue # 1405
Hononlulu, HI 96817
808-531-8843 Group Size 107
Contact Person: Jim Branchaud, President
Meet 3rd Wed 7:00 pm at Pearl City HEC
BB 808 487-8755

IA, Ames
Ames HUG
5006 Todd Dr c/o George Covert
Ames, IA 50010
515 292-1231 Group Size 40+
Contact Person: George F. Covert
Meet 1st Wed 7:00pm Will exchange Newsletters
Meet in room 204 Eng Anx Bldg at ISU

IA, Des Moines
DMA HUG (Des Moines Area HUG)
10275 NE 23rd Avenue
Mitchellville, IA 50169
515 967-6042 Group Size 21
Contact Person: Harold Dykens
Meet third Mon each month 7:00 pm

IL, Champaign
CCCC Champaign County Cmp Club
1004 Kinch
Urbana, IL 61801
217 344-2178 Group Size 150
Contact Person: Jim Mullen
Meet 1st Wed 7:30 pm at Urbana Civic Center
24hr Public Service BB 217 359-9090

IL, Chicago
CHI-HUG (Chicago HUG)
C/o Heathkit Electronics Center
3466 W Devon Chicago, IL 60645
312 674-2491 Group Size 20
Contact Person: Hank LaBarbara, President
Meet 2nd Thurs 7:30pm at Chicago HEC
BB 312 679-8980 after store hours

IL, Davis
NI-HUG Northern IL HUG
427 Lockwood Rt 1
Davis, IL 61019
815 248-2241 Group Size
Contact Person: Jim Isenhart
Just starting

IL, Downers Grove
I-HUG (Illinois HUG)
6116 Lane
Downers Grove, IL 60516
312-971-1660 Group Size 15
Contact Person: Len Bateman
3rd Wednesday at various locations

IL, Downers Grove
HUG Metro (Local Chicago)
124 Arlington Road
Barrington, IL 60010
312 358-6293 Group Size 40
Contact Person: Marvin Gino, Secretary
Meet 2nd Monday of each Month 7:30pm at HEC

IL, Peoria
CIHUG (Central Illinois HUG)
2422 Willow
Pekin, IL 61554
309 347-3366 Group Size 18
Contact Person: John Cole, Jr.
3rd Sunday at 3 PM (Jan, Mar, May, Jul, etc.)

IL, Springfield
Springfield HUG
2621 S. College
Springfield, IL 62704
217 525-1878 Group Size 12
Contact Person: Jim or Bobbie Suttie
Club just forming

IN, Indianapolis
Indiana HUG (IHUG)
11425 Lakeshore Dr West
Carmel, IN 46032
317 852-3530 Group Size 75+
Contact Person: Charles C. Hillman Jr.
Meet 2nd Wed 7:30pm at HEC

IN, South Bend
MIHUG (Michiana HUG)
51061 US 31 North
South Bend, IN 46637
219 255-3923 Group Size 10
Contact Person: Mark L. Meidel
Meet 3rd Monday 7:30 pm

KS, Mission
MUG (Mission Users' Group)
5960 Lamar Avenue
Mission, KS 66202
913 381-5470 Group Size 80+
Contact Person: Jerry Royse, Sec/Treas
Meet last Sun of the month 2:00pm at Mission HEC
BB 913 362-9583 and Newsletter

KS, Wichita
Wichita HUG
8010 E. Zimmerly
Wichita, KS 67207
316 794-2698 Group Size 18
Contact Person: Don Robinson
Meet 2nd Sun of each month at 2:00 pm
Phone for location of meetings

KY, Louisville
LHUG (Louisville HUG)
6802 Crossmoor Lane
Louisville, KY 40222
502 426-9433 Group Size
Contact Person: Ray Donner
Meet last Sun at 8:00pm at Louisville HEC

LA, Baton Rouge
LSU H/Z Users' Group
Dept of Chem Eng LA State Univ
Baton Rouge, LA 70803
504 388-1426 Group Size 40
Contact Person: Danny Reible, President
Meet 2nd Wed at 4:00pm Ctr for Eng & Bus Admin
$5.00 dues/yr

LA, Lafayette
ZUG (Zenith Users' Group)
318 W. St. Mary Blvd
Lafayette, LA 70506
318 948-7804 Group Size 40
Contact Person: Tommy Billiodeaux
Meet every other Tues 6:00 pm
Meet at Zenith Computer Depot

LA, New Orleans
NOHUG
1900 Veterans Blvd.
Kenner, LA 70062
504-467-6321 Group Size 80
Contact Person: John Ligda
Meet 3rd Wed at 7:30 pm at HEC

LA, Shreveport
SHRUG Srvpt Heath Regional UG
C/o Colvin L Sammons POB 752
Barksdale AFB, LA 71110
318 742-8552 Group Size 31
Contact Person: Colvin L. Sammons
Meet 3rd Wed 7:00 pm at Shreveport-Bossier Votech

MA, Northampton
Hampshire Computer Club
37 Drewson Drive
Northampton, MA 01060
617 584-6227 Group Size 80
Contact Person: George Scheurer
2nd Tuesday 7 PM at McConnel Hall Smith College
Beginners Group 1st Tuesday

MA, Peabody
HUG North Shore
12 Stanley Road
Lynnfield, MA 02940
617 334-5128 Group Size 60
Contact Person: Ernest Bay, President
Meet 2nd Wed at Peabody Heathkit Center
BB 617 531-9332 24hrs

MA, Pittsfield
BerCHUG (Berkshire County HUG)
Box 1985
Pittsfield, MA 01202
413 499-2903 Group Size 12
Contact Person: Paul E. Ouellette, President
Meeting place and time vary
Evening phone number 413 443-1862

MA, Wellesley
HUG'EM
165 Worcester Ave
Wellesley, MA 02181
617 237-1510 Group Size 200
Contact Person: Malcolm Partridge, Director
3rd Wed 7:00 p.m. at HEC
BB 617 237-1511 24hrs

MD, Baltimore
Baltimore HUG
C/o Heathkit Electronics Center
1713 E Joppa Rd Baltimore, MD 21234
Group Size 50
Contact Person: James Kratzer
2nd Mon 7:00 pm at Park School - Old Court Road
May also meet at the Heathkit Center

MD, Rockville
MD Z100 Special Interest Group
C/o HEC 5542 Nicholson Lane
Rockville, MD 20852
301 384-1040 Group Size
Contact Person: Jerry Horwitz
Meet 1st Mon at 8:00pm at Rockville HEC

MI, Ann Arbor
A-SQR-HUG
895 Starwick Drive
Ann Arbor, MI 48105
313 769-6052 Group Size 15+
Contact Person: Leonard E Geisler
Meet last Mon 7-9:30 pm Jun-Aug Ann Arbor Pub Lbr
Meet Sep-May at Northside School

MI, Detroit
Metro Detroit Area HUG
35681 Hees
Livonia, MI 48150
313 427-3905 Group Size 65+
Contact Person: Neil Coffin-Sec, Tres, Librn
Meet 2nd Sat of alternate months at the 2 HECs
Club meets in the evening

MI, Kalamazoo
SMHUG (Southwest Michigan HUG)
1054 Blanchard SW
Wyoming, MI 49509
616 532-3875 Group Size 20
Contact Person: Bob Hamel
Meet 2nd Sat 10:00am Kalamazoo Val Comm College
Main building Room 4010D

MI, Okemos
H/Z SIG a part of M3G
2283 Knob Hill Drive #12
Okemos, MI 48864
517 349-9657 Group Size 10+
Contact Person: Bill Goodwin
Meet 2nd Wed at 7:30pm
Meet at All Saints Episcopal Church

MI, Saint Joseph
BLHUG (Blossomland HUG)
P.O. Box 414
Saint Joseph, MI 49085
616 983-0161 Group Size 50
Contact Person: Vance A. Fisher, President
1st Tues 7:00 pm at St Joe High Sch Cmptr Classrm
$15.00 dues/yr Monthly Newsletter

MN, St. Paul-Minneapolis
SMUGH
15533 Sandhill Circle
Eden Prairie, MN 55344
612 934-0159 Group Size 150
Contact Person: Mike Warren, President
Meet last Sun 2:00 pm at Falcon Hgts Comm Ctr
BB 612 778-1213 24 hrs

MO, St. Louis
SLHUG (St. Louis HUG)
3794 McKelvey Road
Bridgeton, MO 63044
618 259-8113 Group Size 120
Contact Person: Brad Pulaski, Treasurer
Meet 2nd Wed 7:30 pm at HEC

NC, Charlotte
HUG Charlotte
4415 Emory Lane
Charlotte, NC 28211
704 364-9667 Group Size 25
Contact Person: Mike Lafleur
Meet 1st Tues at 7:30 pm

NC, Fayetteville
Club no longer in existance

NC, Glen Alpine
Western Piedmont HUG
Rt 2, Box 371
Morganton, NC 28655
704 584-3684 Group Size 10
Contact Person: Bill Poteat
Meeting time and place varies
Will have BB

NC, Greensboro
Carolina HUG
2711 Azalea Drive
Greensboro, NC 27407
919 855-5188 Group Size 40
Contact Person: Graham Andrews
Meet 3rd Thurs at 7:30 pm at Greensboro HEC
BB 919 292-5392 after hours

NC, Hillsborough
HUG-RTP
Rt 3, Box 39A
Hillsborough, NC 27278
Group Size
Contact Person: Joe Williams
Meeting place and time unknown

NE, Omaha
OMAHUG (Omaha HUG)
PO Box 777
Bellevue, NE 68005
402 291-4676 Group Size 100
Contact Person: Robert Wasilewski, President
3rd Sun odd mos. 6:30 Bellevue W HS or Offutt AFB
Meet even mos Amer Red Cross 6:30 pm

NH, Amherst
HUG of New Hampshire
61 Stearns Road
Amherst, NH 03031
603 673-6040 Group Size 23
Contact Person: Dean Hayden-Macy
Meet 2nd Mon 7:30 pm at Systematic Solutions Inc.
BB 603 673-7366 Newsletter on BB

NJ, Fairlawn
HUGNJ (HUG of New Jersey)
C/O AMBI-TECH 319 Knickerbocker
Hillsdale, NJ 07642
201 666-0504 Group Size 155
Contact Person: Matt Baum
BB 201 791-6936 evenings
3rd Monday 8:00pm at HEC

NJ, Ocean
SHUG (Shore HUG)
1013 State Hwy 35
Ocean, NJ 07712
201 775-1231 Group Size 71
Contact Person: James J Jones Jr. (Sec)
Meet 1st Wed 7:30 p.m. at Ocean HEC
BB 201 775-8705 24hrs.

NM, Albuquerque
Albuquerque HUG
7909 Hendrix NE
Albuquerque, NM 87110
505 294-1658 Group Size 25+
Contact Person: Ken Benson
Meet 3rd Sun at members homes

NY, APO New York
BWHUG (Bentwaters HUG)
PSC Box 3703 RAF Bentwaters
APO New York, NY 09755
Group Size 6
Contact Person: TSGT Rodney Jones
Phone Autouon 225-2161 or Saxmundham 3519

NY, APO New York
BAHUG (Bad Aibling HUG)
UNIT AA Box 561
APO New York, NY 09098
Group Size 10
Contact Person: Louis J. DeMichele
Phone 08061-4519/6340 West Germany

NY, Buffalo
BUG (Buffalo Users Group)
223 Clark Road
Kenmore, NY 14223
Group Size 75
Contact Person: Bob Allen
Meet 3rd Tues 7:00 pm at Amherst HEC
Club doesn't meet July, Aug or Dec

NY, Long Island
Jeri-HUG (Jericho HUG)
5 Helen Place
Glen Cove, NY 11542
516 676-5616 Group Size 75
Contact Person: Alan Scott Dodge, Sec./Tres.
Meet 2nd Thurs at 8:00 pm at the Jericho HEC
Newsletter, Software library, BB

NY, North White Plains
North White Plains HUG
Elliott Ser Co 720 White Plns Rd
Scarsdale, NY 10583
Group Size 50
Contact Person: Peter Abramson
Meet 2nd Tues. ea mo 7:30 pm at HEC

NY, Potsdam
CCT HUG (Clarkston College)
Woodstock Vlg Apt 3B24
Potsdam, NY 13676
315 268-6455 Group Size 60
Contact Person: Marc A. Rubin
Meet monthly-call for date, time and place
Club just getting started

NY, Rochester
RH/ZUG (Rochester H/ZUG)
937 Jefferson Road
Rochester, NY 14623
716 424-2560 Group Size 55
Contact Person: RHUG Editor, Blanche Nail
Meet last Tuesday each month 7:30 pm at HEC
BB 716 424-2576

NY, Schenectady
Schenectady HUG
C/o T. Budge 715 Sanders Ave.
Scotia, NY 12302
518 377-4273 Group Size 20
Contact Person: Walter Whipple
Meet 3rd Wed 7:30 pm at above address

NY, Syracuse
SYRHUG (Syracuse HUG)
C/o Garrett Voss Box 6
Oran, NY 13125
315 682-5113 Group Size 18
Contact Person: Garrett Voss
Meet 1st Wed at 7:00 pm
24 hr BB 315 682-6912

OH, Cincinnati
Cincinnati HUG
10133 Springfield Pike
Woodlawn, OH 45215
513-771-8850 Group Size 90
Contact Person: President
2nd Tuesday 7:00 pm at HEC, $10.00 dues/year
Newsletter, 24hr BB 513 772-6190

OH, Cleveland
NOHUG (Northeastern Ohio HUG)
7838 Valley Villas Dr.
Parma, OH 44130
216 845-6752 Group Size 54
Contact Person: Don Danko, Sec.
Meet 2nd & 4th Thurs 7pm at St. Gregorys Church

OH, Cleveland
Cleveland HUG
28100 Chagrin Blvd
Cleveland, OH 44122
216 291-0698 Group Size 30
Contact Person: Kent Currie
First Thurs 7:00 p.m. at HEC
BB 216 292-7554 Non-store hrs only

OH, Columbus
Columbus HUG
2095 Milden Road
Columbus, OH 43221
614 457-0419 Group Size 40
Contact Person: Jay Eikes
Meet 2nd Tues at HEC
BB 614 475-7201 after store hours

OH, Dayton
Dayton HUG
1670 N Laddie Ct
Beaver Creek, OH 45432
513 426-5014 Group Size 160
Contact Person: George Elwood
Meet 1st Thurs 4:00 pm for H8/89 users
Meet 3rd Thurs for H/Z100 users 4:15

OH, Toledo
THUG (Toledo HUG)
48 S. Byrne Road
Toledo, OH 43615
419 729-4621 Group Size 300
Contact Person: Ryck Zarich
Meet last Sunday of the month at 7:00 p.m at HEC
BB 419 537-1888 24hrs also 729-4221

OH, Youngstown
YOU-HUG
1502 S. Raccoon Road
Youngstown, OH 44515
216 799-5028 Group Size 12
Contact Person: Mario DeSantis
Club just getting started

OK, Oklahoma City
OKC TUGS
C/o Bill Cadwallader PO Box 1171
Lawton, OK 73502
405-848-7593 Group Size 40
Contact Person: Bill Cadwallader
Meet 3rd Thurs 7:30 pm at HEC
BBS 405-848-9329 24 hours

OK, Tulsa
Tulsa HUG
Rt1, Box 813
Sperry, OK 74073
Group Size 5
Contact Person: Christian Kessler
Meet 2nd Tues at members homes
Club just starting - No dues

OR, Salem
SHUG Salem Heath Users' Group
PO Box 13434
Salem, OR 97309
503 393-0786 Group Size 21
Contact Person: Ken Hiigel
Meet 2nd Tues each month
Contact Ken Hiigel for location

PA, Allentown
Lehigh Valley HUG
C/o Bob Kendi Cpt Ctr Lehigh Un
Mart Library 8-B Bethlehem, PA 18015
215 770-5993 Group Size 30
Contact Person: James Batug
2nd contact Bob Kendi 215 861-3992
Meet 2nd Sat at Lehigh University

PA, Frazer
FUG (Frazer Users Group)
1641 Princess Anne Drive
Lancaster, PA 17601
717-397-3146 Group Size 80
Contact Person: Dave Hendrie, President
1st Saturday 4:00 pm at Frazer HEC
BB 215 644-7661

PA, Harrisburg
CPaHUG (Cent Pennsylvania HUG)
7540 Mourningstar Dr %E. Asper
Harrisburg, PA 17112
717 545-2764 Group Size 7
Contact Person: Ernest E. Asper
Meeting time & place varies
Club just getting started

PA, Philadelphia
Philadelphia Heath Users'Group
6318 Roosevelt Blvd
Philadelphia, PA 19149
215 748-1439 Group Size 75
Contact Person: Colin C. McGowan
Meet 2nd Wed. each month 7:30 p.m. at HEC
RBBS (phone # to be announced)

PA, Pittsburgh
Pittsburgh HUG
3482 William Penn Highway
Pittsburgh, PA 15235
412 882-5932 Group Size 35
Contact Person: Bill Pridemore, President
Meet 3rd Tues at 7:00 pm at HEC
Newsletter

RI, Warwick
HUG-`RI' (HUG of Rhode Island)
558 Greenwich Avenue
Warwick, RI 02886
401 738-5150 Group Size 150
Contact Person: Dave Haskell or Bill Rothman
2nd Wed 8:00 pm at HEC

SC, Anderson
Anderson HUG
401 Tiffany Drive Regency Park
Anderson, SC 29621
803 225-0084 Group Size 2
Contact Person: John R. Miller
Meet monthly, place varies, club just starting

SD, Sioux Falls
Sioux Falls Area HUG

2001 S Spring Avenue
Sioux Falls, SD 57105-2820
605 336-8629 Group Size 20
Contact Person: Lorin Dobson
Meet once a month on Sat-time and place varies
BB 605 336-3935 24 hrs

TN, Knoxville
ETCHUG East Tenn Central HUG
7608 Luscombe Dr
Knoxville, TN 37919
615 690-3864 Group Size 20
Contact Person: Walter M. Scott III
Meet 3rd Thurs 7:30 pm
Meet at John XXIII Center

TN, Memphis
Memphis HUG
6874 Kirby Brooks Drive
Memphis, TN 38115
901-362-8860 Group Size 16
Contact Person: Morris Proctor
Meet 2nd Tues 7:00 pm at The Computer Center

TN, Nashville
Mi Te HUG (Middle Tenn HUG)
C/o Radio Ser Ctr 116 17th Ave S
Nashville, TN 37203
615 242-0556 Group Size
Contact Person: Charlie Wolf
Meet 2nd Monday 6:30 pm at Radio Service Center

TX, Austin
AHUG Austin Heath Users Group
4206 Tamarack Trail
Austin, TX 78727
512 255-0376 Group Size 60
Contact Person: George Koehler
Meet 1st Thurs 8:00pm Univ. of TX $10.00/yr dues
Meet at Robert Lee Moore Hall

TX, Dallas
DFW HUG (Dallas-Fort Worth)
2715 Ross Avenue
Dallas, TX 75201
214-826-4053 Group Size 70
Contact Person: Henry Gardiner, President
1st Thursday and 15 days later (Wed.) at 7:30 PM
At HEC BB 214-742-1380

TX, El Paso
MOBHUG (Mobile HUG)
354 Teakwood Drive
Satsuma, AL 36572
205 675-9742 Group Size 22
Contact Person: Bob Small
Usually meet the last Friday of the month

TX, Ft. Worth
FWHUG
6825A Greenoakes Road
Ft. Worth, TX 76116
817 737-8822 Group Size 60
Contact Person: Kent Young
Meet 4th Thurs 7:30 pm at HEC

TX, Houston
HUG-H
7798 Braniff
Houston, TX 77061
713-644-5689 Group Size 75
Contact Person: Tom McCormick, President

TX, Houston
NHHUG (North Houston HUG)
20207 Cotton Glade
Humble, TX 77338
Group Size 50+
Contact Person: Barbara Gabner, Secretary
Meet 3rd Tues 7:30 at HEC
24 hour BB 713 583-1287

TX, San Antonio
San Antonio (SAHUG)
7111 Blanco Road
San Antonio, TX 78216
512-341-8876 Group Size 65
Contact Person: Tom Schneider
First Tuesday at HEC, 7:30 PM

TX, Wichita Falls
NORTEX HUG (N. Texas S. Okla)
2413 Kemp Blvd in Office World
Wichita Falls, TX 76309
817 322-1007 Group Size 24
Contact Person: Alan D. Martin
Meet third Sat 9 am at above address

UT, Castle Dale
Castle Mesa Computer Group
670 N 90 E Box 587
Castle Dale, UT 84513
801 381-5173 Group Size 10
Contact Person: Doug Sorensen
Meet 2nd Mon 5:30 pm at above address

UT, Midvale
UHUG (Utah HUG)
58 E. 7200 South
Midvale, UT 84047
801 262-8810 Group Size 130
Contact Person: Wayne Newland
2nd Wednesday 7:00 PM at HEC
BB 801 566-4551

VA, Christiansburg
New River Valley HUG
C/o CCS Data Sta. 8 Roanoke St
Christiansburg, VA 24073
703 382-4234 Group Size 42
Contact Person: Ted Fleshman
Meet 1st Thurs 7:30 pm Christiansburg High School

VA, Fairfax
CHUG (Capital HUG)
P.O. Box 2653
Fairfax, VA 22031
703 360-3812 Group Size 1000
Contact Person: John Roach
Meet 3rd Mon 7:00 pm at Fairfax High School
Large Software Library (150+ disks)

VA, Richmond
RHUG (Richmond HUG)
4302 Smithdeal Avenue
Richmond, VA 23225
804 231-6759 Group Size 20+
Contact Person: Carlos Chafin
Meet 3rd Mon 7:30 pm
Meet at Alpha Audio 2049 W Broad

VA, Virginia Beach
THUG (Tidewater HUG)
1055 Independence Blvd.
Virginia Beach, VA 23455
804 464-4156 Group Size 90
Contact Person: Alison Phillips
1st & 3rd Tues 7:30 pm at HEC

WA, Bellevue
PNHUG (Pacific Northwest HUG)
14213 SE 52nd Place
Bellevue, WA 98006
206 643-6651 Group Size 250
Contact Person: Barry Dupler
Meet 2nd Thurs odd months Tukwila HEC (both 7:00)
Meet 2nd Mon even months Seattle HEC

WA, Spokane
SPOHUG (Spokane HUG)
S. 3810 Havana
Spokane, WA 99204
509 448-9727 Group Size 25
Contact Person: Charles Ballinger/Ron Hodges
Meet last Thurs 7-9pm at Acme Business Computers
Alternate phone # 509 448-5009

WA, Vancouver
Portland-Vancouver HUG
516 SE Chkalov Drive
Vancouver, WA 98663
206-254-4441 Group Size 30
Contact Person: Dan Heims
1st Thursday at 7:30 PM at HEC
Portland OR and Vancouver Area

WA, Walla Walla
HUG/ZUG of Walla Walla
112 N. Division
Walla Walla, WA 99362
509 525-8404 Group Size 8+
Contact Person: Pat Hanna
Meet 2nd & 4th Tues 8pm at 112 N Division
2nd cont Pete Parcells 509 527-5267

WI, Madison
Madison Area HUG
3519 Tally Ho Lane
Shorewood Hills, WI 53705
608 233-4588 Group Size 25
Contact Person: Thomas Gans
Meet 1st Wed 7:30pm at Wisconsin Union South

WI, Madison
UWHUG (Univ of Wisconsin HUG)
109 N Few
Madison, WI 53703
608 257-0373 Group Size 30
Contact Person: Walter Burt
Meet 1st Wed 7:30pm at Univ WI Union South
Club newly formed 12/83

WI, Milwaukee
MHUG Milwaukee Heath Users' Gp
9040 N. Lake Drive
Milwaukee, WI 53217
414 352-3346 Group Size 80
Contact Person: Marvin Olson, Tres.
Meet 3rd Sat 2:00pm at Milw Sch of Eng Rm L-100

WI, Mosinee
CWHUG-Central Wisconsin HUG
PO Box 66
Rothschild, WI 54474
715 359-4156 Group Size 10
Contact Person: Edward Ignace Porwit
Meet 2nd Tues

CANADA, Calgary, ALBERTA
Heath Users of Calgary
#101-5809 Macleod Trail South
Calgary, Alberta T2H 0J9 CANADA
403-252-2688 Group Size 20
Contact Person: Bill Jones
Meet 1st Tues at 7:00pm at Calgary HEC
Club reorganized March 1985

CANADA, Edmonton, ALBERTA
HUGOE (H/Z Users' of Edmonton
17314 106 Avenue
Edmonton, Alberta CANADA T5S 1H9
403 483-4656 Group Size 12
Contact Person: Edward Hrdlicka
Meet 1st Wed at 7:30pm Meeting place varies.
BB 403 454-6093

CANADA, Ottawa, ONTARIO
HUG `O' (HUG Ottawa)
866 Merivale Road
Ottawa, ONTARIO K1Z 5Z6 CANADA
613-728-3731 Group Size 30
Contact Person: Brian Fultz, President
2nd Wednesday 8:00 PM at HEC

CANADA, Toronto, ONTARIO
THUG (Toronto and area HUG)
54 Camrose Crescent
Scarborough, Ont. CANADA M1L 2B7
416-273-3797 Group Size 20
Contact Person: Stephen Dugas, President
Meet 2nd Thurs 7:30pm - meeting place varies
BB 416 755-8823

CANADA, Vancouver BC
VANHUG (Vancouver HUG)
3058 Kingsway Attn Robert Hudak
Vancouver BC, CANADA V5R 5I7
604 437-7626 Group Size 50+
Contact Person: Robert J Hudak
Meet last Tues 7:30 pm at HEC

CANADA, Vancouver, BC
Vancouver Island HUG
Suite 304 560 Johnson St
Victoria, BC CANADA V8W 3C6
604 384-3134 Group Size 9

Contact Person: Greg Greene, President
Meet 2nd Wed 7:30 pm at Dogwood Software

ENGLAND, Gloucester
United Kingdom Users' Group
ZDS Limited Bristol Road
Gloucester,GL2 6EE, ENGLAND
0452-29451 Group Size 80
Contact Person: Philip Meek
Group does not hold regular meetings

FRANCE, Paris
GUFIH (French HUG)
37 Boulevard Saint-Jacques
75014 Paris, FRANCE
1 565-10-11 Group Size 350
Contact Person: Dr. Bernard Pidoux
Meet every WED at 7:00 pm at club address
CBBS 1 565-10-09 Monthly newsletter

HOLLAND, Apeldorn
See NETHERLANDS club listing

HONG KONG
Compudragon
273 Prince Edward Road
11/C Kowloon, HONG KONG
3-711-8904 Group Size
Contact Person: K.T. Lee
Club just organizing

KOREA, Yongsan, Seoul
K-HUG
HHC 1st SIG BDE c/o LTC Dismukes
APO San Fran, CA 96301
293-4132 Group Size 6
Contact Person: LTC William Dismukes
Just getting started-monthly not established yet.

NETHERLANDS
Dutch Heath Users' Group
NIEUWE KERKHOF 16
9712 PV Groningen, NETHERLANDS
050-180203 Group Size 120
Contact Person: Evert Jan Stokking
Meet quarterly at Amersfoort

NEW ZEALAND
HUG New Zealand
94 Dowse Dr
Maungaraki, Lower Hutt, NEW ZEALAND
695-924 Group Size 1
Contact Person: Mr. R. Siebers
Would appreciate New Zlnd REMark readers contact
Eager to expand group

OKINAWA
OKIHUG (Okinawa Users Group)
Kenneth K Bailey,Jr Box38 USAFSO
APO San Francisco, CA 96331
Group Size 22
Contact Person: Kenneth K Bailey Jr, President
Meet 2nd Fri at 7:00 pm at American Video Office
Alternate phone # 631-4244

PANAMA CANAL
Club no longer in existance

PUERTO RICO, Rosario
PRHUG (Puerto Rico HUG)
Calle La Paz #706, Miramar
Santurce, PR 00907
809 722-1612 Group Size 30
Contact Person: Joseph Gonzalez
Meet 2nd Sunday of odd numbered months

SCOTLAND, EDZELL
EDZELL HUG
Box 517 NSGA EDZELL
FPO New York, NY 09518
EDZELL-7254 Group Size 3
Contact Person: Lt. Dave Smith
Club very interested in expanding!

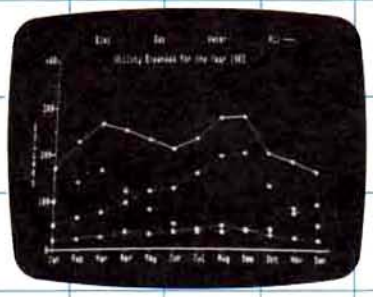
W. GERMANY, Frankfurt
Club no longer in existance

Heath/Zenith Users' Group

This is the One

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

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



Superior Graphics!

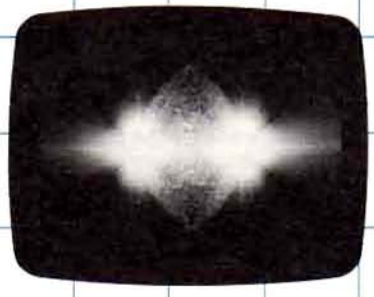
- Highest Pixel Resolution, 640x250 (or 640x500)
- Fastest Drawing Speeds Available
- 3 Graphics Pages for Animation
- Hardcopy Output to most Dot Matrix Printers
- Adds Graphics Commands to all of your Languages

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

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

New Print Spooler!

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



Many Other Features!

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

Conversational Command Set

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

An Example in BASIC:

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

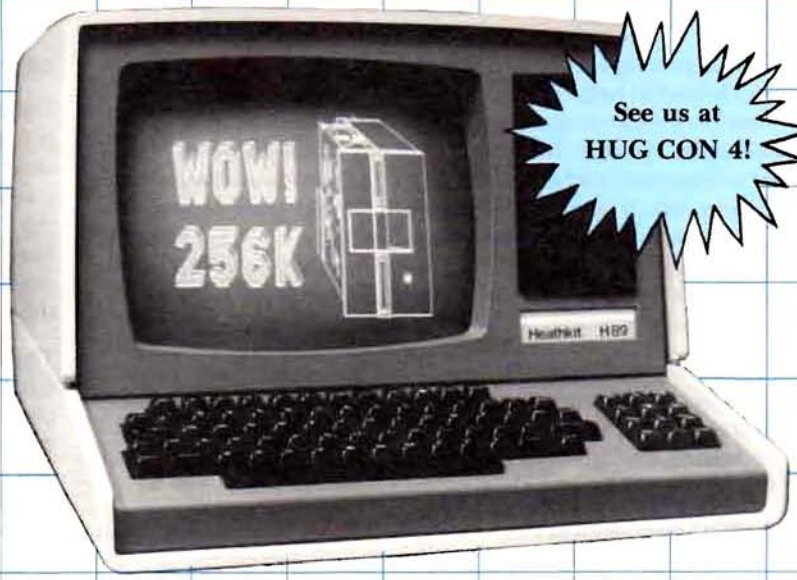
Our engineers would love to answer all of your questions about our products. Call and talk to one of them right now!

Now Shipping from Stock!

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

MasterCard VISA

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



SigmaSoft AND SYSTEMS

"Support is the most important feature."

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

Transport Your H-89 To The 16-Bit Worlds Of The 8086 & 68000 CPU's

Peter Ruber
P.O. Box 502
Oakdale, N. Y. 11769

Loran R. Porter Jr.
4545 So. Mission Road 190
Tucson, Arizona 85714

There is no escaping the Blue Plague from Boca Raton. Whatever lackluster features have been standardized during the last few years by IBM's entrance into the personal computer market place, we are stuck with them. Worse still, so are the vast majority of computer manufacturers who are caught up in a financially destructive whirlwind of catch-me-if-you-can situation. Having used a variety of large and small computer systems over the years, I find it amusing that one can get the same 320 × 200 multi-color resolution on a $150 Commodore-64 as with a $2500 IBM PC, including sprite graphics and a sophisticated sound/music chip.

But the real irony is that innovation has not only been stifled, but has gone bankrupt in the process. Somehow or other the mother of invention perseveres and we are occasionally presented with something unique.

Several years ago, Hallock Systems Co. was a consulting firm that specialized in helping businesses and institutions in the upstate New York area transport their files and records to the new 16-bit computer systems. This was a time consuming process, highly labor intensive, and ultimately costly for the contracting organizations. Yet, it was still cheaper than rewriting all the files and risking errors in the process. The biggest problem for HSC was that their clients used a variety of different systems. This meant writing a specific cross-assembler and special software for each client.

Then, early in 1983, one of their engineers began to look into the possibility of a hardware device that could handle the major task of transporting 8-bit software to the 16-bit format. There was a common bond among most of their clients' computers: the Z80 CPU and CP/M.

Within a year, their brainchild resulted in a desktop console that interfaced with nearly every Z80 computer system on the market. They called it the CO-16 Attached Resource Processor which, by itself, is really a stand-alone computer console without its own I/O control capabilities. These functions are performed by the host computer system's Z80 CPU through the CP/M operating system.

The CO-16 is mated to the computer via a small Interface Card. You simply remove the Z80 chip, mount it on the Interface Card, and then plug the board into the Z80 socket on the H/Z-89's CPU Logic Board. The Interface Card contains the necessary buffer logic and drivers to transport the required Z80 Data and Address lines to the CO-16 console via a 25-wire ribbon cable. This cable is 14" long and terminates in a female DB-25 plug that can be easily mounted on the H/Z-89's backplane. A male DB-25 plug at the end of a 24" cable connected to the CO-16 console can then be plugged into the connector on the backplane.

The Interface Card for the H/Z-89 is not ideal and reflects a compromise in order to accommodate the position of the Z80 CPU. Due to the fact that the mounting bolts for the CRT's flyback transformer interfere with the piggy-backing of any PC board in the Z80 socket, installing the Interface Card is a little tricky. You have no choice but to take a pair of wire cutters and trim off the extension of these bolts, otherwise you will not be able to slide the CPU Logic Board back down into the case.

The ideal method (after unplugging your computer) is to remove the 4-40 lockwasher nuts that hold the flyback transformer in place, trim the bolts closer to the heatsink panel through which they mount and then fasten them with standard 4-40 nuts. You will then have a comfortable margin of space. The cable connected to the Interface Card plugs onto a post connector, so you can easily disconnect it if you have occasion to remove the CPU

Logic Board.

Now let's examine the 8086 version of the CO-16.

MSDOS For The H/Z-89?

The heart of the CO-16 is a densely-packed Logic Board measuring 7.25" X 8.50". This board contains contains 256K Ram using 4 banks of 9/64K DRAMS each. The unit was originally designed to use Intel's advanced 80186 CPU, but due to the scarcity of this chip during the last year, HSC designed a small daughter board containing a 6 Mhz 8086 that plugs into the Logic Board. There is an expansion connector on the Logic Board that will allow the user to post-mount and plug in an expansion card containing an additional 512K DRAM, for a total on-board capacity of 768K.

At a later date, if the user is inclined to spend an additional $80-90, he can purchase the Intel 80186 chip and install it himself. Consultation with HSC, however, is advised because several support chips will have to be changed, including the PAL.

There is a provision to add the Intel 8087 Math Chip for complex floating point calculations. At the factory level, HSC can also install such enhancements as a Real Time Clock, an Interrupt Controller, DMA Logic, 2 Serial Ports, as well as some other features.

HSC currently supports version 2.11 of Microsoft's MSDOS, which has been modified by them to allow access of this operating system through CP/M. The onboard BIOS ROMs are capable of an 80% emulation of the IBM BIOS. Due to the limitations of CP/M-80, a greater degree of emulation is not possible at this time. That means you will be able to use all generic MSDOS programs on the market, but not programs that directly address the IBM BIOS, nor some of the heavily protected programs. A specific listing of such programs will be found at the end of this article.

The software package supplied by HSC is rather impressive. You get the CO-1686 User's Manual, the Microsoft MSDOS User's Manual, the MSDOS Programmer's Reference Manual, the Z80 Assembler User's Manual, and the 8086 Cross Assembler User's Manual. You also get 3 full disks that contain a total of 39 files and utilities, which is enough to keep you glued to the computer for weeks while you familiarize yourself with their use.

A capability I found most interesting is that you can use either 48-tpi DS/DD or 96-tpi DS/DD drives, providing your CP/M System Volume has been configured to recognize both types of drives.

Let's review the basic set-up procedures that are required to integrate the CO-1686 MSDOS conversion into the H/Z-89.

Take a System Volume containing your present BIOS, plus the FORMAT.COM, SYSGEN.COM, and PIP.COM utilities and BOOT from it in Drive A:. Take Disk #1 from the 3 you received and mount it in Drive B:. PIP the following files from B: to A:

MSDOS.SYS	(this is the basic Operating System)
EX16.COM	(this is the HSC BIOS that invokes the MSDOS Operating System after you boot up your system under CP/M-80)
RAMDSK.COM	(this allows you set aside a portion of available memory for use as a Ramdisk; the first 128k are reserved for the Operating System)
INSTALL.COM	(this is the configuration utility necessary to set up your system)

(An essential feature I should have mentioned earlier is that the Interface Card you install on the CPU Logic Board contains a 6-position Switch that enables you to assign the proper HEX format I/O Port assignment through which the Z80 and the 8086 CPU's will communicate. The CO-1686 User's Manual provides a chart designating 64 available I/O ports. Both Loran Porter and I have assigned Port 80 for our 8086 and 68000 units, which is the selection recommended by HSC for the H/Z-8 and H/Z-89 computers. The 1983 Buss Directory contains a list in the back of the book on all the I/O Port assignments used by various Heath/Zenith peripheral manufacturers. Consult this list to make certain that you DO NOT duplicate an I/O Port already installed in your system if you choose a port other than 80.)

Whenever I'm ready to install a complex addition to my H/Z-89, I approach the software installation with a certain degree of trepidation. I was especially concerned about using the Heath/Zenith 2.2.04 version of CP/M and leaving my computer fully stocked with every board and peripheral I owned, but the installation went smoothly.

1. BOOT your CP/M disk containing the utilities mentioned above.
2. At the A: prompt, type INSTALL [RETURN].
3. The display will then prompt you with 3 choices:
   1/INSTALL
   2/DISPLAY PORTS
   3/TEST
4. Type #1.
5. The program will ask you to type in the I/O Port. Type 80. The program will execute several tests to determine if this is a valid, unused Port on your system. If you obtain an Error Message, you must check the Port assignments on your peripheral equipment. If you have no conflicts, the program will automatically install Port 80 on the disk.
6. Next, you will be asked what type of Terminal is connected to your Z80. Type in the number that corresponds to the H-19, and hit RETURN.
7. You are then asked if you wish to invoke the IBM PC function key emulator. Type N. (The user has the option to designate special key sequences to not only emulate the 42 unique IBM PC function keys, but the upper 128 IBM PC display characters as well. I have not yet purchased an IBM manual nor any of the many books on the PC that describe the purpose of the function keys and the display characters, so I have not fiddled with this option. To implement this feature, two additional HSC utilities are required on your BOOT disk: ROME.SYS and INST.DAT. If you chose to assign a Control Key designation to each of the above, the program will automatically record this data on your disk.)
8. If you wish to invoke the Ramdisk feature, type Y at the prompt and then designate the amount of Ram you wish to allocate for this feature. Remember, the first 128K Ram are reserved for the operating systems. The size of the Ramdisk can be any remaining portion of your available memory. If you have the full 768K system, 640K can be assigned as a Ramdisk by CP/M-80, which is means the entire contents of a 96-tpi drive can be dumped into Ram.

9. The next step in the Installation procedure will ask you to assign a letter value to your Ramdisk. Your choice is from A to P. Select a letter that does not currently conflict with any Disk Drive designations.

10. Now the program asks you to define the size of your CCP (Console Command Processor). Type 6 because CP/M uses 6K.

11. The final step involves assigning a letter to every Disk Drive recognized by your CP/M Bios. You have the option of designating all or part of your drive set-up as a Logical Disk Device for use by MSDOS. This allows you to have a mix of 48 and 96-tpi drives assigned for MSDOS use, as well as 8" drives, which gives you the utmost freedom for software readability.

The above installation procedures take about 10-15 minutes at most. Any of the assigned values can be altered at a future date. You can also create additional BOOT disks with different values written to the BIOS if you have a specific need.

If you are an experienced Assembly Language programmer, HSC provides you with the Source Code on Disk #3 to 6 of the utilities so that you can reconfigure the BIOS for your Ramdisk, the Installation sequence, and the IBM PC emulator functions. To a limited extent, you can also alter the 8086/Z80 BIOS. To those who are carefree and confident, FMTMS.COM utility allows you to FORMAT disks in 8, 9 or 10 sector formats so that you can read and write to a variety of MSDOS formats.

It is advisable to make back-up copies of the 3-disk package you receive from HSC. This is accomplished by using the PIP B:*.*=C:*.* command. I recommend that you make 2 back-up copies of each disk: one set for your daily work; the other set to create additional working disks. Put the originals away with those of your other important Operating Systems and Program Disks.

Those of you who have installed 4 Mhz options in your computers and have experienced more rapid I/O access, an 8086 running at 6 Mhz will even outshine a Z-110/120 containing an 8 Mhz modification because the 8088 is only a hybrid 8/16-bit CPU.

Using your new CO-16 is rather simple. You Boot under CP/M as always. Then, to enter the MSDOS Operating System, you type EX16 [RETURN]. If you wish to temporarily put aside any work or program under MSDOS, do a CTRL-C and the warm boot will return you to CP/M. Thus, you can easily switch in and out of each Operating System at will. While you are able to place both CP/M and MSDOS on a single disk, you must create a Working Disk for each format in order to gain the maximum amount of storage space. If you BOOT CP/M from A:, you can place your working MSDOS disk in B:. Transfer of MSDOS control of the computer is then assigned by typing B:EX16 [RETURN].

Doing so will load EX16 into the Z80 RAM space. After some brief disk activity which loads the appropriate MSDOS operating system modules into memory, the MSDOS sign-on message will appear. You are then prompted to record the date, a function similar to HDOS. You may choose to ignore the date entry feature by pressing the RETURN key. One final prompt allows you to also set aside RAM disk space under MSDOS.

The speed by which everything transpires during the BOOT process is a little disconcerting at first. There are prescribed steps that must be taken. A negative feature of HSC's 8086/Z80 Bios is that it is not as forgiving as HDOS. You are not always presented with an Error Message defining your mistake. During such instances the computer sometimes has a tendency to go into an infinite loop that is only recoverable through a hard reset.

If you are forced into SHIFT/RESET on the H/Z-89, you must remember to also press the RESET rocker switch on the front if the CO-16. Failure to so do will result in the inability of the BOOT process to load the respective operating systems. I found this an irritant at first, primarily because I have become so used to HDOS and CP/M that I don't always allow for the fact that unfamiliarity does result in system errors. And owing to the fact that HSC has tried a universal approach to mating the 8086 to a variety of Z80 systems, one cannot really expect total perfection.

I have thus far used PFS WRITE, MULTIPLAN and BASIC without any hitches. I have also borrowed from friends a variety of small programs and utilities that were typed in from program listings found in the assorted PC magazines. I have tried to stay away from programs that are IBM PC hardware specific, because one is unable to use this software with the CO-16. This includes graphics programs which are not supported. With the IBM PC ROM BIOS emulation under MSDOS, I/O access to the CRT, Keyboard, Printer, Disk Drives and RS-232 functions are translated to the corresponding Z80 CP/M system functions via the translate table created by the INSTALL PROGRAM.

Another type of program that cannot run on the CO-16 are copy protected programs such as LOTUS 1-2-3, Symphony, SuperCalc 3, R: BASE and DBASE III. In the case of the IBM PC, I suspect the copy protection schemes employed make calls to the IBM BIOS when the programs are mounted. HSC does, however, state that a number of users have written to them with information pertaining to both the elimination of the copy protection schemes and with details of some implemented logic that will permit the copy protection schemes of these programs to actually run on the CO-16. As both methods are not condoned by HSC, they maintain an official posture that these programs will not run on the CO-16. I, personally, don't think that is much of a loss. Most copy protected programs are so vastly over-priced to begin with, that alternative programs with relatively similar features are probably your best course.

What Does It Cost?

The basic CO-16 Logic Board package retails for $695. This includes 256K DRAM, an Interface Card of your choice (the "B" version is most suitable for the H-89), HSC's implementation of MSDOS 2.11, plus all the necessary cables, manuals described earlier, as well as a host of utility programs and assembly development tools. For an additional $125 you can purchase a handsome cabinet and fused linear power supply. This cabinet, by the way, has the most solid fit of any I have ever seen, and does credit to the highly professional execution of the Logic Board. The same case with a whisper fan will cost $175, and is required if you are planning to add the 8087 Math IC, or the eventual substitution of the 80186 CPU. They require faster cooling than the excellent convection cooling scheme employed by the case.

For an additional $190 you can purchase an unpopulated 512K Ram Expansion Board. All sockets and passive components are soldered in place; you can add your own 64K chips at bargain prices.

The obvious benefits are many: you can retain your present H-89 system and all its software and the costly peripherals you have acquired over the years, and also be able to use all of it with a true

MSDOS system for a relatively small cost. The CO-16 will also run CP/M-86, which should certainly please most Digital Research fanatics.

Experienced Assembly Language programmers will delight in the many development software tools HSC has supplied. It will allow them to customize the software for specific applications. It is quite obvious that HSC accomplished a good deal of software wizardry with these utilities.

As mentioned earlier, there has not been any conflict with any hardware I have installed in my H-89. This includes Heath/Zenith accessory board, as well as those from SigmaSoft & Systems, FBE Research Company, Microflash and New Orleans General Data Systems.

The only negative feature is the documentation. The manuals prepared by HSC (the CO-16 User's Manual and the Z80 and 8086 Cross Assembler Manuals) are offset from an italicized type face, that has also been reduced to almost half-size. This makes the reading of it somewhat difficult. The setting up and installation copy is sometimes sketchy and lacking in details, and I get the distinct feeling that the writer of the User's Manual geared the text for one of his colleagues rather than for the general user. The MSDOS User's Manual and the MSDOS Programmer's Reference Manual were direct offset copies of Microsoft's standard manuals, but HSC has somehow managed to cram (via photographic reduction) two pages on one by positioning the pages sideways. This manual, too, has been stapled, and the result is that it will not lie flat for reference. If the printer had positioned the pages a little better, there would have been enough room to three-hole punch the pages and place them in a binder. Curiosity seekers will be pleased, however, that HSC has included 10 pp. of schematics in the User's Manual illustrating all available interfaces and Logic Board options.

But, as I like to end on a positive note, HSC does provide decent technical support. My letters were answered promptly, and my telephone queries were courteous and detailed. For $100 per year, HSC will provide the purchaser with an Annual Service Package that includes: automatic notification of software changes/corrections/enhancements; software updates on an "as-requested" basis (diskette costs not included); unlimited phone consultation; Quarterly Newsletter, and a 12-month Extended Equipment Warranty.

Enter The 68000 CPU

If the H-89 owner decides to purchase the HSC CO-1668, containing a 6 Mhz Motorola 68000 CPU, he will find himself in possession of one of the most powerful micro-computers currently available.

The CO-1668 has been set up as a small Unix system. Unix was developed by AT&T as a multi-user system, and HSC's upgrade will support the interfacing of three H/Z-89's at the same time. This feature can be quite attractive to many small businesses who have several H/Z-89's and may not be inclined or able to invest heavily in a totally new multi-user system after only a few short years.

The CO-1668 has a basic package price of $995. (The case with power supply is extra.) This includes 256K Ram; an Interface Card of your choice; all necessary cables; a memory expansion bus; a Real Time Clock; an Interrupt Controller; a provision for 4 math co-processors; CP/M-68K (with "C" Compiler); a RAM disk, BIOS and EX16 Sources; assembly development tools, and a CO-1668 User's Manual. (For some odd reason, the manual for Digital Research's CP/M-68K is not included and will cost the user an additional $82.)

Although this cost may at first appear slightly out of proportion to the CO-1686 version, certain features that are available as an option on the CO-1686 are standard on the CO-1668. The computing power this unit is able to provide is realized when you add the Memory Expansion Card. You can populate it with 8 banks of 64K Ram for 512K additional memory; or with 4 banks of the new 256K Ram chips for an additional 1024K of workspace over and above the 256K on the 68000 CPU Logic Board.

Quite possibly the most expensive feature is the ability to install 4 National 16081 Math Co-Processors. My little Canon pocket calculator winces in shame. The National chips are currently selling for approximately $350 each. But what you now have is a small mainframe computer hooked to your H/Z-89. This is quite awesome in terms of raw computing power and speed at a fraction of what you would have to pay for other systems of this magnitude with multi-user capability.

Both the CO-1686 and the CO-1668 will support Winchester hard disk drives and tape back-up units as long as your version of CP/M-80 has the proper utilities to format and partition these drives. As such, once CP/M-68K is installed on the hard-disk, along with CP/M-80, you can boot the system directly from the Winchester drive.

A good part of the CO-1668's computing power is derived from the "C" language. "C" was also developed by AT&T, and is to 16-bit systems what COBOL is to mainframes. Anything written in "C" can very easily be made to run on any other computer, including a mainframe. By including a file called SCRCTL.H on my disk I tried several experiments in which I kept the keywords and altered only the coding. I was then able to run my sample programs on an H-100, an IBM PC, as well as the H/Z-89.

If you have an interest in programming and feel restricted by BASIC, "C" is, indeed, a unique language to experiment with. Knowing "C" is almost essential if you are interested in using the CO-1668. Two books that I consider required reading are: 68000: Principles And Programming by Leo J. Scanlon, and C Primer Plus by Mitchell Waite, Stephen Prata, and Donald Martin. Both are SAMS publication.

Using CP/M-68K with "C" is simplified by the utilities provided by HSC. CP/M-80 has a type ahead program called "SUBMIT". To call up a type ahead routine you write the routine and call it a "*.SUB" file — such as "BEGIN SUB". To run the type ahead routine you write "SUBMIT BEGIN". With CP/M-68K, the type ahead program is in the CCP. So, if you have such a routine written, you write "BEGIN" to run it. The command file extension of CP/M-68K is *.68K — not *.COM. Four SUB routines have been written by HSC to make it easy to invoke the "C" compiler: C.SUB, CLINK.SUB, CLINKE.SUB and CLINKF.SUB. A "C" program must have the *.C extention. To compile it you simply write C PROGRAM NAME. CLINK.SUB is used to link programs with integer math or arithmetic only. To link such programs you write CLINK PROGRAM NAME. CLINKF is for floating point programs, and CLINKE is for floating point programs with scientific notations.

Installing the CO-1668 software on the H/Z-89 is quite similar to the MSDOS version. The only real variation is that you are able to define certain CONTROL keys to act as toggle to move in an out of CP/M-80 and CP/M-68K. My system has been set up to recog-

nize CTRL R followed by CTRL C as the means to transfer control from one operating system to the other.

I have been using the CO-1668 for nearly a year without any mechanical difficulties. My list of peripherals is quite different from those Peter Ruber is using with his CO-1686 MSDOS unit. Mine contains the Magnolia 128K Ram board, a Cleveland Codonics Imaginator Graphics Board, a Heath Z-89-11 Multi I/O board, and a 20 Meg. Corvus Winchester hard disk drive. The only sacrifice I have had to make is the temporary removal of my Kres 2/4 Mhz modification in order to make room for the CO-1668 Interface Card. I am planning, however, to hard wire the Kres and HSC Interface Cards in series.

We cannot say with true conviction that either of the HSC units we have just reviewed will satisfy everyone's need. There are some limitations because even the most clever software cannot totally trick an 8-bit CPU into actually believing it is a 16-bit CPU. As such, you must sacrifice not being able to run some of the sophisticated graphics-intensive programs available for 16-bit machines.

This is not to detract from the usefulness of either the CO-1686 or the CO-1668. The engineering is first-rate and the software is extensive. Its cost-effectiveness in relation to buying a completely new 16-bit system is most favorable; and, as mentioned earlier, you can still use all the accessory cards and peripherals attached to your H/Z-89 system.

Another point worth mentioning, is that both the CO-1686 and CO-1668 are transportable. That means, if you have a Radio Shack, Epson QX-10, TI Professional, NRC Decision Mate, Kaypro II or 4/10, Cromenco — or any standard S-100 system — in addition to your H/Z-89, you can buy additional Interface Cards for about $70, and use these units on just about any Z80 on the market.

Both units have been an excellent learning experience.

For further information on all available options and current pricing, contact:

Hallock Systems Company, Inc.
P.O. Box 86
Herkimer, NY 13350
(315) 866-7125

Applications Software

The C0-1686 Attached Resource Processor will run all generic MSDOS 2.11 and most MSDOS 1.0 and IBM PC programs that are not hardware specific. In view of the ambiguous nature of this statement, HSC has a standing policy that they will pre-test any software forwarded to them by their prospective customers. The following is a list of PC software that is known to run on the CO-1686. It is by no means complete and is listed for reference purposes only.

Will Run

R:BASE 4000 — Microrim
Benchmark — Metasoft
Multiplan — Microsoft
T/MakerIII — TMaker
SuperCalc2 — Sorcim
dBaseII — Ashton-Tate
PerfectWriter — PerfectSoft
Wordstar Series — Micropro
PFS Write — Soft. Pub.
DosEase/DataEase — Soft. Sol.
KnowledgeMan — MDBS
Income A/P,A/R,G/L,OE,Purch,
Inv.Cont. — MC
TOTAL Series — TCS
Formula II — DMA
Condor 20-1, 20-2 — Condor
Infostar — Micropro
Electronic Circuit — Tatum Labs
The Final Word — Mark of the Unicorn
MicroGANTT — Earth Data
ProFin/PlanFin — Business Software
RealEstate Analysis — Real Data
RE Comm/Industrial — Real Data
StatPak — NWA
Check & Balances — CDE
MBasic — Microsoft
C Compiler MS — Microsoft
M2CBASIC — Buzzwords
Basic Compiler — Microsoft
Business Basic — Microsoft
COBOL MS — Microsoft
Fortran MS — Microsoft
Janus/ADA — RR Soft
Pascal MS — Microsoft
CB86 Compiler — DRI
PL/I-86 — DRI
Pascal/MT+86 — DRI
muLISP/muSTAR — Microsoft
Macro-MS — Microsoft
C86 — Comp. Innovations

Will Not Run

Lotus 1-2-3 — Lotus
Flight Simulator
Harvard Project Manager
Jack2
SuperCalc3 — Sorcim
dBaseIII — Ashton-Tate
DBMaster

Powerful High-Level Languages For The CO-1668

A number of very powerful high-level programming languages have been located for the CO1668 running under CP/M-68K. In addition to the inherent fourfold processing speed advantage of the 68000, these languages all bring miniframe level programming to the H/Z-89 system owner — using the entire 768K RAM for program/operating system space. Below is a list of these packages and the distributor of each:

Empirical Research Group (ERG)
PO Box 1176
Milton, WA 98354
— FORTH — ERG
— Business Basic Compiler (Alpha Micro compat.) — ERG
— Screen Editor (VEDIT) — ERG
— C Compiler — Whitesmith
— Fortran 77 — Silicon Valley
— Pascal — Silicon Valley
— Basic-Plus — Silicon Valley

The Computer Company (TCC)
PO Box 6987
Richmond, Virginia 23230
— APL.68000 — TCC

Unified Software
PO Box 2644
New Carrollton, MD 20784
— UNIFORTH — US

Volition Systems
1-619-481-2286
— Modula 2

Laboratory Microsystems
PO Box 10430
Marina del Ray, CA 90295
— FORTH

Westico
25 Van Zant St.
Norwalk, CT 06855
— CBASIC 68K — DRI
— Fortran 77 — DRI

*

Part One

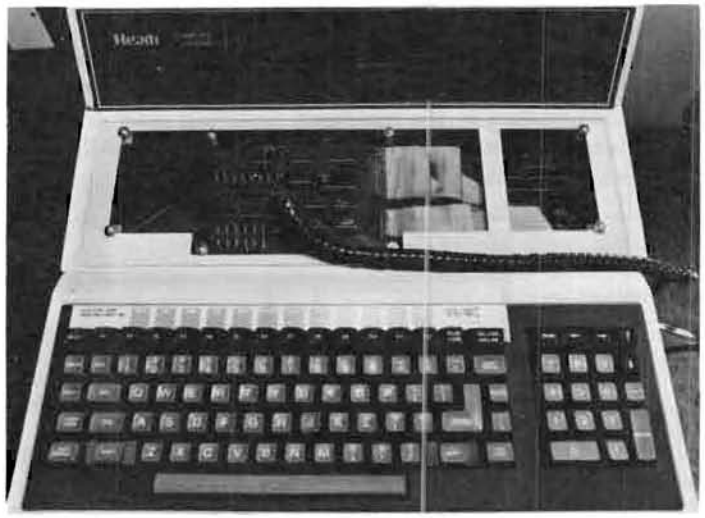
H/Z-100 Remote Keyboard Adapter

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

I've owned my H/Z-100 for two years now. It has two disk drives, 192k of RAM, and a monochrome monitor. I have used the H/Z-100 as a learning tool while attending New Mexico State University's College of Engineering. I must say it has served me well. I have become dangerously fluent in FORTRAN, BASIC, and Assembly Language. I am, as time permits, learning C and Pascal. I also use the computer to type term papers and magazine articles (this is my second) and to keep track of expenses (I like to know why I'm always broke).

I must say the H/Z-100 is a competent and flexible computer with a large variety of software. I can't wait to buy the speed and memory modification offered by Heath. But I've always had one gripe with the H/Z-100 - the fixed keyboard. I often spend hours at the computer typing term papers or programming and many times I have wished I could move the keyboard just a bit to make those marathon sessions a little easier.

Last year I began work on this project, devoting semester breaks to its development. The specifications for the remote keyboard adapter are simple. I wanted to be able to sit in a swivel chair with my feet on the desk (I have a terminal case of bad posture) and still be able to enter programs or type purple prose. Since the H/Z-100 has the best keyboard in creation I wanted to retain the original keyboard. I wanted the remote keyboard to operate perfectly with any operating system and any hardware configuration. I did not want to modify the mother board in any way. If, for any reason, the adapter should fail, the adapter could be removed and the keyboard simply restored to its original position inside the H/Z-100's cabinet. I did not want to have to write special configuration programs for every conceivable operating system in existence. I did not want to use an S-100 slot. Finally, I wanted the modification to be a simple plug-in installation. When I first started designing this adapter I wanted to use an infra-red link for the keyboard so it would operate in a manner similar to the IBM PC Jr's keyboard. I found that IBM used an infra-red transmitter and receiver chip set manufactured by Texas Instruments. A problem occurred. IBM discontinued the PC Jr. and Texas Instruments halted production and support of these infra-red devices. I tried to find another source for infra-red transmitter and receiver chips but found none. An infra-red link was no longer possible, so I settled on a wire link from the keyboard to the computer.



The Adapter — Installed

I was an electronics technician for twelve years prior to attending college, so I have a great deal of experience in soldering, wire wrapping, drilling holes off-center, and the like. I will assume anyone undertaking this project has constructed several electronic devices and has the technical skills necessary to insure successful completion.

You must be able to locate a few critical components which are not available at your local electronics shop or the parts houses in the electronics magazines. I spent a great deal of time on the phone locating parts and data books and have done most of the leg work but you are entitled to a little fun hunting up special parts.

There are many ways to design a device of this nature. I have made every effort to minimize the number of parts required. I welcome any improvements. Please share your comments and suggestions with all members of HUG by writing to REMark.

If you write to me I will make every effort to answer any questions you may have and will try to help with any problems encountered.

One comment about this project is in order. IT WORKS !!

In the following discussion I will assume you have access to the H/Z-100 technical manuals and miscellaneous integrated circuit data books.

General Theory

The H/Z-100's keyboard circuits are designed around Intel's 8041/8741 universal peripheral interface (UPI). I read Intel's data sheets and came to the conclusion that this chip reads the keyboard under control of a program in its internal ROM. The exact operation of the internal program is not important to the operation of the adapter. The keyboard is composed of rows and columns of momentary contact switches arranged in a matrix. If you don't have an H/Z-100 tech manual, look at a small section of window screen and imagine a switch at every intersection of the screen's wires. The UPI scans the keyboard one column at a time through two 74LS156 demultiplexers. The output of the 74LS156 is active low. If a key or several keys in the column currently being read are pressed, the low output of the 74LS156 will be routed to the current column, through the active key or keys, and back to the 8041 UPI where the resulting 8-bit code is interpreted and sent to the CPU. The character / function keys are on one large matrix. The modifier keys (SHIFT, CAPS LOCK, and FAST REPEAT) are on a separate matrix and are also scanned by the UPI. Two keys, CTRL and RESET, are not scanned by the UPI. They are tied to other circuitry so that their presence can be sensed immediately by the computer. From this point on I will lump the control and modifier keys into one group and will refer to these six keys as control keys.

The adapter consists of two units: the encoder and the decoder (I had to call them something). The encoder, under the control of a program I wrote, scans the keyboard in a manner similar to the 8041. It then encodes a word which is serially transmitted through a coiled cord and received by the decoder. The decoder interprets this code, and in essence, simulates a pressed key. Only one key can be simulated at a time with my design, but this is not a hindrance for any use I know of. The encoder is capable of finding roughly 50 keys per second which works out to 500 words per minute at 6 characters per word. The world's fastest typist approaches 200 words per minute. She would be satisfied with my design. You will find that this adapter is transparent to both the H/Z-100 and the user. In other words, once installed, you will retain the speed and functionality of the original keyboard configuration, but the keyboard can be located up to ten feet away from the computer.

Reading the Schematics

Take a look at Figure 1, the encoder schematic. You will find various integrated circuits, resistors, capacitors, a diode, connectors, and a system of buses. Let's concentrate on tracing out a signal line to determine where it originates and where it ends up. A signal line's label or name is located above the respective integrated circuit or connector pin: Z80, pin 33 is A3. Look closely at the schematics. You will notice wide black lines and narrow black lines. The wide black lines represent buses. The narrow lines represent individual signals. A signal line enters or leaves a bus at a 45 degree angle. A signal line which crosses a bus

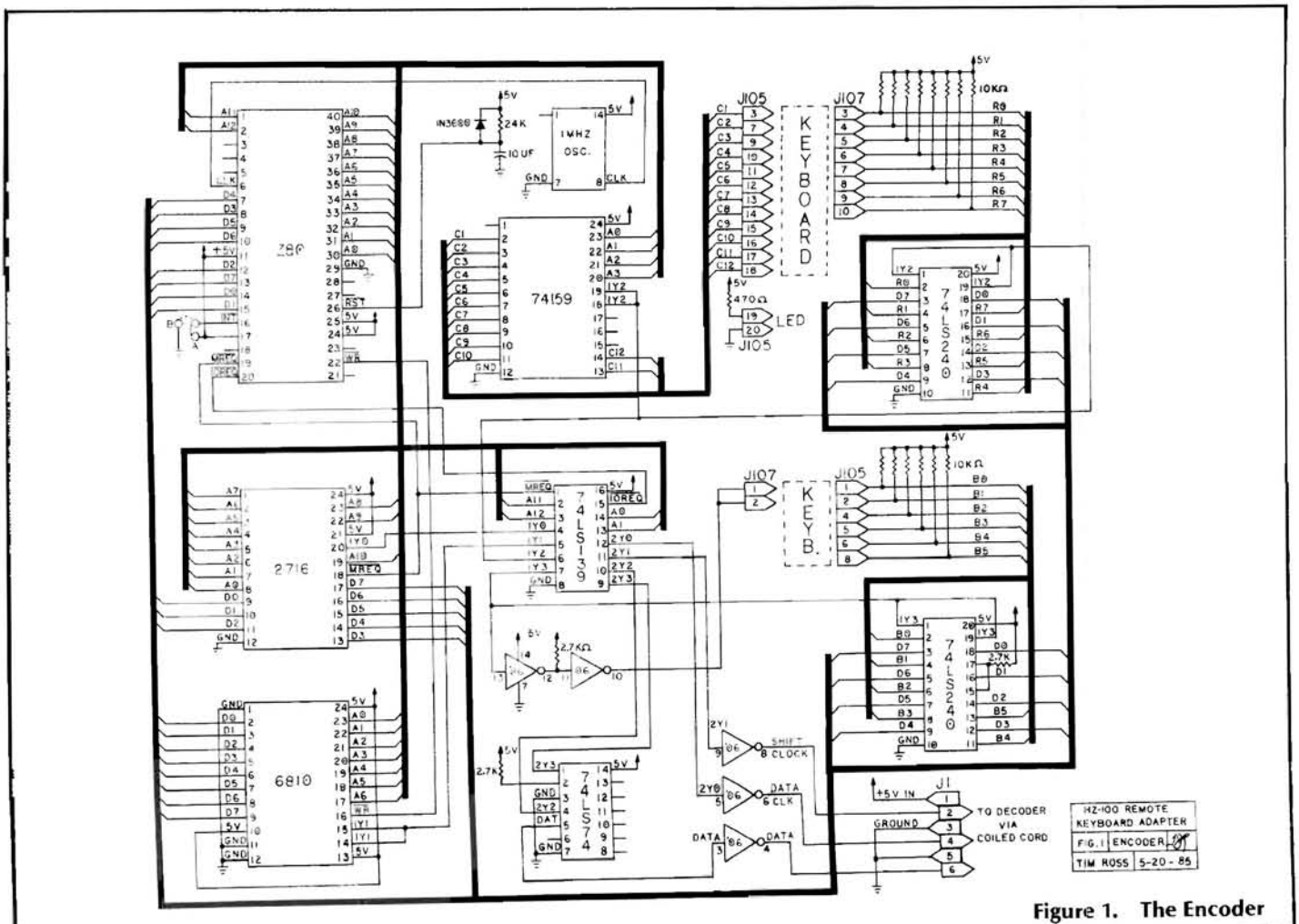


Figure 1. The Encoder

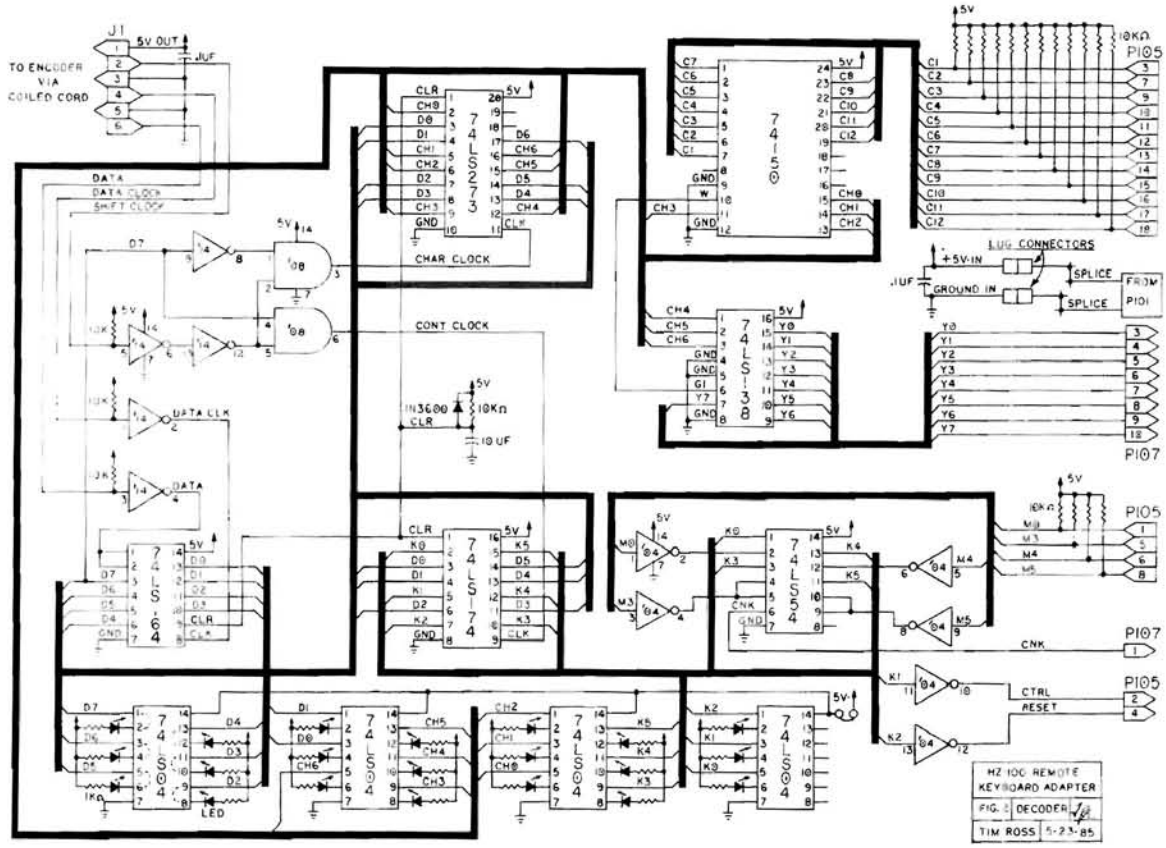


Figure 2. The Decoder

at a 90 degree angle is independent of that bus. If I tried to draw every wire connecting the various integrated circuits I would make a thousand errors drafting the schematics and you would have a hard time following the resulting prints. Drawing the schematics in this fashion, while perhaps initially confusing, is both easier to lay out and to follow. A sense of data flow is more readily discernible using this method.

Examples: Both a data bus and an address bus are tied to the Z80, the 2716, and the 6810. Look at pin 33 of the Z80. This is the A3 line or Address Bit 3. Trace this signal, starting at Z80 pin 33, to the address bus, then follow the bus, looking for the places where A3 exits the bus. You will find that it is applied to the following: 74159 pin 20, 2716 pin 5, and 6810 pin 20. A0 is connected to the following: Z80 pin 30, 74159 pin 23, 2716 pin 8, 6810 pin 23, and 74LS139 pin 14. The data bus is wired to many integrated circuits. D0 is connected to the following: Z80 pin 14, 2716 pin 9, 6810 pin 2, and both 74LS240's, pin 18.

There are many signals which are not data lines or address lines. Look at Figure 2, the decoder schematic. There is a signal called CLR. Its function is to provide a power-on clear to three IC's in the decoder. It is tied to 74LS273 pin 1, 74LS164 pin 9, 74LS174 pin 1, and the resistor / capacitor / diode network. Let's trace another signal on the decoder schematic. Look at P105 pin 1 at the lower right of the decoder's print. This signal comes from the H/Z-100 mother board, through the decoder's ribbon cable where I name it M0, travels through one of the buses, and is applied to pin 1 of a 74LS04. This signal is inverted and applied to 74LS54 pin 2, where it is processed.

Another possible source of confusion exists in my depiction of the keyboard connectors. I have drawn the schematics to give the reader a sense of the flow of signal lines. In Figure 1 I show a keyboard bus tied to the 74159. The individual lines are named C1 through C12. Following the arrows, you see a box representing the keyboard, then another series of signals named R0 through R7. These 'R lines' are the keyboard row lines and are tied to a 74LS240. Each line is pulled up with a 10k ohm resistor.

On the encoder schematic you will find several places where J105 and J107's pins are depicted. I have broken up J107 and J105, again to show signal flow. There are only two keyboard connectors on the encoder. These connectors are right-angle males which mate with the female ribbon cable connectors from the keyboard itself. J107 is a ten-pin connector and J105 is a twenty-pin connector.

A similar situation is found in the decoder schematic. There are only two decoder keyboard connectors, both females, both are connected to the decoder through 24 gauge .100" center ribbon cables. These connectors plug into the mother board in place of the original keyboard.

At the bottom of the decoder schematic I have drawn four 74LS04's which drive Light-Emitting Diodes. Only one resistor is labeled as a 1k ohm resistor. In fact, each of these resistors is a 1k ohm resistor. There simply was not enough space on the print to label each and every resistor and LED.

Tied to the keyboard connectors on both schematics you will see banks of resistors. These are all 10k ohm resistors and are used to pull up the open-collector outputs of the devices driving the keyboard.

Adapter Circuit Theory

The Encoder

The central processor is a Z80. It is capable of addressing 64k bytes of memory. The encoder has 2k of ROM, 128 bytes of RAM, and 14 keyboard addresses. I am not taxing the Z80. I chose the Z80 because it is inexpensive, it is easy to interface and program, and there is a Z80 assembler which operates with CP-EMULATE and MS-DOS. The assembler is called UVMAC and is sold by The Software Toolworks.

The keyboard is treated as fourteen memory addresses. The Z80 reads a keyboard address and an 8-bit word corresponding to the key pattern in the current column is saved. If a key is pressed, a one is generated for that key's data bit, if a key is not pressed, a zero is generated for that data bit. Tests are made on this 8-bit pattern to determine if any keys are pressed. If no keys are found in a column, the Z80 will read the next column address and so on until all twelve character columns have been scanned. The control keys are then scanned, then the character key scan routine restarts.

There are three valid character key conditions. First: no keys in the entire character matrix are active. Second: one key is active. Third: originally there was one character key pressed but now this original key and a second key are active. If more than two keys are pressed at the same time anywhere in the character key matrix, only two keys will be found (just like the 8041). The program will locate the first key pressed, debounce it (7 milliseconds), verify it, transmit it to the decoder, and will remember where this key is located in the matrix. The program will then scan all of the character keys, looking for another active key. Control keys are also read, once per scan. If a second character key is found, it too will be transmitted and its location in the character key matrix remembered. These two keys will be repeatedly scanned until one is released, then the whole routine starts over. When all keys have been released, the decoder character buffer is cleared.

The six control keys may be active in any combination. These control keys reside at memory address 1800H. The Z80 reads this address and transmits the control key pattern to the decoder.

The program resides in a 2716 2k EPROM. For RAM I use a Motorola 6810 static RAM. I chose the 6810 because it is inexpensive and no support circuitry is necessary. Both the 2716 and the 6810 are more than adequate for the needs of this project.

A 74LS139 is used to select the various devices accessed by the Z80. The truth table follows.

	$\overline{MREQ}$	A12	A11	BASE ADDRESS	DEVICE
MEMORY ACCESS	0	0	0	0000H	2716
	0	0	1	0800H	6810
	0	1	0	1000H	CHAR KEYS
	0	1	1	1800H	CONT KEYS

	$\overline{IOREQ}$	A1	A0	PORT	SIGNAL
I/O	0	0	0	00H	DATA CLOCK
	0	0	1	01H	SHIFT CLOCK
	0	1	0	02H	DATA = 1
	0	1	1	03H	DATA = 0

As can be seen, the 74LS139 is used to access both memory devices and I/O devices. More on I/O later.

A 74159 4-to-16 line demultiplexer is used to scan the character / function keys of the keyboard. Twelve columns are scanned, corresponding to keyboard connector J105 pins 3, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18. The base address of the keyboard is 1000H which corresponds to address bit A12 from the Z80. Z80 address bits A0-A3, with A12, will access the character keys. J105 pin 3 is at address 1001H, pin 18 is at 100CH. The four address bits, A0-A3 are tied to the 74159 and will enable me to read up to 16 keyboard columns, one at a time. Of course, there are only 12 character key columns. Column 0 is not used because a no-key-active code is required to clear the contents of the decoder character buffer.

The output of the 74159 is routed through the active keyboard column and fans out through any keys pressed. An eight-bit pattern, corresponding to the eight possible keys in a column, is present at the input of a 74LS240. The 74LS240 is a tri-state octal line driver. Its output is enabled at the same time the keyboard character keys are scanned. This output is sent to the Z80 on the 8-bit data bus.

The 1Y3 output of the 74LS139 is active when A11 and A12 are both active, corresponding to an address of 1800H. This 1Y3 signal is sent through two stages of the 7406 open collector inverter (the output of the first stage is pulled up with a 10K ohm resistor) and then to J107 pins 1 and 2 and the control keys. The output from the control keys is sent to another 74LS240. The control keys require a tri-state buffer separate from that of the character keys. If both the character keys and control keys were on the same matrix, that is, if they were tied to the input of the same buffer, it is possible for false characters to be generated. Example: '>' could be interpreted as '?' if both the CAPS LOCK and SHIFTR were held and then '>' pressed. I found this out the hard way.

Once either a character or a control key is found it is formatted into an 8-bit word for presentation to the decoder. This formatted word contains several pieces of information. For character keys the MSB will be a zero, the next three bits will tell where in the column the key is located, and the four least significant bits are determined by the active column. A control key word's MSB will be set, the next bit does not matter and will be clear, and the next six bits will tell the decoder which of the six control keys are active.

Examples: If 'q' is pressed, 05H or 00000101B will be transmitted. This corresponds to the zero'th (did I just invent a new word?) key of the 5th column. If 'F1' is pressed, 13H or 00010011B will be transmitted. If 'CAPS LOCK' and 'FAST REPEAT' are pressed, 91H or 10010001B will be sent to the decoder. If the previously active character key is released, 00H will be sent to the decoder to clear the character buffer, a 74LS273. If the previously active control key is released, 80H will be sent to clear the control key buffer, a 74LS174. Codes are transmitted only once per key closure / release. If the code was transmitted repeatedly, two things would happen. The UPI would "see" a bouncing key and the Z80 would be busy transmitting and would never find other active keys.

As stated above, the 74LS139 is used for I/O. The first few times I designed this contraption I used the D0 line as a data output line. The problem with this is that there is an awful lot of activity on the Z80's data lines and I wanted to minimize the problems in interfacing a ten-foot long coiled cord between the two circuit boards. I decided to use a D flip-flop to store the current data bit

being transmitted. I use the 2Y2 and 2Y3 outputs of the 74LS139 to preset or clear the D flip-flop (74LS74). The Q output of the 74LS74 is sent to the decoder through a section of the 7406. This output is rock-steady. Once a bit is stored in the 74LS74, I send a DATA CLOCK to the decoder to serially shift this bit into a serial-to-parallel shift register (a 74LS164). After eight bits have been transmitted, I generate a SHIFT CLOCK which will send the data stored in the decoder's 74LS164 to the appropriate buffer and the H/Z-100's UPI will sense the presence or absence of a key.

A 1 megahertz clock is provided to run the Z80. There is a power up reset circuit tied to pin 26: a resistor, a capacitor, and a diode. A similar circuit is in the decoder, though the resistor value is smaller to clear the decoder buffers before the Z80 comes on line.

The three encoder outputs, DATA, DATA CLOCK, and SHIFT CLOCK, are sent through open-collector inverters, through a short cable connecting the encoder PC board to the coiled cord connector, through the coiled cord, through another connector, and finally to the input section of the decoder.

The Decoder

The three signals from the encoder are sent through their respective sections of a 74LS14, a hex Schmitt Trigger which is used to clean up the signals and give the clocks nice leading and trailing edges.

DATA is present at pins 1 and 2 of the 74LS164, a serial-to-parallel shift register. DATA is clocked in, one bit at a time, until eight bits have been shifted in. Seven of these bits are tied to the data inputs of the character buffer, a 74LS273 octal D flip-flop. The six least significant bits from the 74LS164 are also sent to the control key buffer, a 74LS174 hex D flip-flop. After a word has been shifted into the 74LS164, a SHIFT CLOCK is generated by the encoder. This SHIFT CLOCK is AND'ed with the MSB from the word just received. If the MSB is high, a control key code is present and the control key buffer is toggled, storing the the new control key word. If the MSB is low, it is inverted and the result is again AND'ed with the SHIFT CLOCK. The character buffer is toggled and the word in the 74LS164 is then stored in the character buffer.

The output of the character buffer (74LS273) is sent to two places. Bits 0, 1, 2, and 3 contain column information and are sent to the data select inputs of the 74150, a one of 16 multiplexer. The data inputs of the 74150 are tied to the character / function key column pins of connector P105. When the H/Z-100's UPI scans the character keys, it looks at the columns in a sequence determined by its internal program. The 'C' inputs of the 74150 are pulsed according to this sequence. If a non-zero is present in the character buffer, the appropriate 'C' (column) input of the 74150 is selected, inverted, and sent to the W output. The W output of the 74150 is sent to the G1 input of the 74LS138. Bits 4, 5, and 6 from the character buffer contain the active key's position in the column and are present at the select inputs of the 74LS138, a 3-to-8 line demultiplexer. When the W output of the 74150 arrives at the G1 input of the 74LS138, the character key information encoded in the 3 select bits enables one of the data outputs. The UPI reads a keyboard column and finds one key active.

Example: If 'q' has been pressed, 05H (00000101B) is sent to the decoder and routed to the character key buffer. Bits 0 and 2 are high, allowing the input present on pin 3 of the 74150 (P105, pin 11) to pass to the 74LS138. Bits 4, 5, and 6 of the character word are low, corresponding to the zero'th key in this particular column being active, so the Y0 output of the 74LS138 is enabled. The UPI places a pulse on J105, pin 11 and sees a pulse on J107, pin3. This corresponds to a 'q'.

The output of the control key buffer is also sent to two places. Bits 0, 3, 4, and 5 are sent to the 74LS54, a 4-wide AND-OR invert gate. The H/Z-100's UPI looks for FAST REPEAT, SHIFTL, CAPS LOCK, and SHIFTR by pulsing J105, pins 1, 5, 6, and 8 individually. If one of these keys is pressed, this pulse will be routed from the above connector pins, through the pressed key, through J107 pin 1, and back to the UPI. With the adapter, coincidence between the UPI's control key scan pulses and bits 0, 3, 4, and 5 from the control key buffer will result in an output from the 74LS54, pin 6 which is sent to P107, pin 1.

Example: If 'CAPS LOCK' and 'SHIFTR' are pressed, the control key buffer will hold 30H, 110000B. Bit 4 is felt on pin 13 of the 74LS54, bit 5 on pin 11. When the UPI scan reaches J105, pin 6, a pulse is sent to pin 12 of the 74LS54. Pins 12 and 13 are AND'ed together, the output sent back to the UPI. When the UPI's scan reaches J105, pin 8, pins 9, 10 and 11 of the 74LS54 are AND'ed together, the output is again sent to the UPI. The end result is these two control keys are sensed by the UPI.

Bits 1 and 2 of the control key buffer are inverted and sent out on P105, pins 2 and 4. These bits correspond to the CTRL and RESET keys. You may recall that these two keys are not sensed by the UPI. In the original configuration, without the adapter, they have ground applied to them and when pressed they send a low to circuitry on the mother board. An inverted high accomplishes the same thing. When CTRL and RESET are pressed at the same time, the computer resets. IT WORKS !!

Test Circuitry

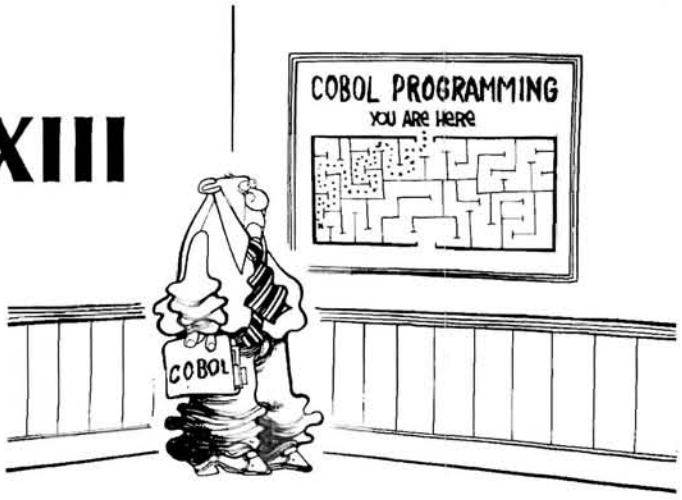
I have included test circuitry in this design. Pin 16 of the Z80 has a jumper drawn on the print, presently +5 volts is tied to pin 16. This is the Z80's interrupt line. When this line is tied high, interrupts are disabled. When this line is tied low, interrupts are enabled and the program I have written contains a diagnostic which will, on detection of an interrupt, jump into an endless loop which repeatedly writes to RAM, reads RAM, and then transmits the results to the decoder.

At the bottom of the decoder schematic you will see four 74LS04's (inverters) tied to the outputs of the 74LS164, 74LS273, and 74LS174. These 74LS04's drive light emitting diodes. Each LED has a 1k ohm resistor between it and +5 volts to limit current. By monitoring these LED's, you will be able to see DATA being shifted in, one bit at a time, and you will be able to see the contents of the character and control key buffers. If you experience any problems with your adapter you will find this circuit invaluable. Once your adapter is up and running, you can easily monitor its operation. Since LED's draw power and electric power costs money you may wish to defeat this circuit. In the prototype I installed a jumper which will connect +5V to the four 74LS04's. I simply remove the jumper to disable the test circuit, but the circuit is always available. Do not connect the fifth 74LS04 to this power jumper because it is used by the control key circuitry.

Conclusion

In my next article, I will describe in detail how to construct the H/Z-100 Remote Keyboard Adapter, supply the parts list, suggest possible sources for parts, and describe how to send me your 2716 EPROM, which I will program and return to you with a listing. I will charge $25.00 for this service, which will help to offset my development costs. *

COBOL Corner XIII



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

Introduction

I know that I gave you readers a lot of "homework" last month. I hope that you were successful in completing Program #5 — PRGM05. You MUST know Program #5 in order to complete this month's Program #6 — PRGM06. I am not going to supply you with as much help, because this program is very similar to Program #5. Of course, if you cannot understand or complete Program #5, BE SURE to write me about your problems! I will need detailed written descriptions including a print-out of your coding. Be sure to include a self-addressed, business size envelope with sufficient postage so that I can supply you with a written reply! NO PHONE CALLS, PLEASE! Also, did you readers that have COBOL-80 Version 4.6 get your system up and running? Again, if you had any problems or questions let me know as I have described above. Did you try Version 4.6 on all of the "COBOL Corner" programs to date? Did they work as before? Which version do you like best? You COBOL-86 readers do not have to make a choice. You will find COBOL-80 Version 4.6 very similar to COBOL-86. Of course, the readers using NEVADA COBOL will find that all of this did not apply to them.

Program #6 Information

Program #6 (1) READS Employee Payroll Records; (2) Computes Total-Hours, Shift-Differential, Regular-Earnings, Overtime-Earnings, and Total-Earnings for each employee; and (3) Prints an Earnings Report Line for each employee. The Earnings Report will have two (2) Program Header Lines and two (2) Column Header Lines on the top of each page. There will be no need for Page or Report Total Lines.

To do this program properly, you must prepare a Structure Chart, a Flow Chart or Pseudo code, an Output Print Chart, an Input Record Chart ,and of course, the CODING SHEETS! If you do not remember what these are, reread your previous "COBOL Corner" articles. You NEED to know these and how to prepare them like you know the "back-of-your-hand". Proper documentation is the "backbone" of GOOD COBOL programming!

As stated in previous "COBOL Corner" articles, I will supply the Program Specifications:

```
-------------------------------------------------------
                PROGRAM SPECIFICATIONS
-------------------------------------------------------
PROGRAM NAME:  EARNINGS REPORT          PROGRAM ID:  PRGM06
-------------------------------------------------------
```

Program Description:

This program prints an Earnings Report from an input file called Employee Payroll Records.

Input File:

FILEL4.

Output File:

Earning Report as described below.

List Of Program Operations:

A. Read each input Employee Payroll Record.

B. For each record, the program should do the following processing:

   1. Compute the Total-Hours-Worked by adding the Regular-Hours and the Overtime-Hours.

   2. Compute the Shift-Differential as follows:

      a. Shift-Code = 1 : No Shift-Differential.
      b. Shift-Code = 2 : Shift-Differential = 10% of Hourly-Rate times Total-Hours-Worked.
      c. Shift-Code = 3 : Shift-Differential = 12.5% of Hourly-Rate times Total-Hours-Worked.

   3. Compute the Regular-Earnings by multiplying the Regular-Hours-Worked by the Hourly-Rate.

   4. Compute the Overtime-Earnings by multiplying the Hourly-Rate by 1.5 and then multiplying this product by the Overtime-Hours-Worked.

   5. Compute the Total-Earnings by adding the Regular-Earnings, the Overtime-Earnings, and the Shift-Differential.

   6. Print a detail line for each employee containing the following fields in accordance with the format shown on your Print Chart.

      a. Employee Social Security Number.
      b. Employee Name, last name first.
      c. Employee Shift-Code.

d. Employee Rate of Pay.
e. Total Hours Worked.
f. Regular Hours Worked.
g. Overtime Hours Worked.
h. Shift Differential.
i. Regular Earnings.
j. Overtime Earnings.
k. Total Earnings.

C. Each detail line should be single spaced from the previous detail line.

D. Heading Lines should be printed on the first page of the Report. After 50 lines have been used on a report page, the program should go to the next page and print the heading lines again.

1. The Run-Date should be obtained from the WORKING-STORAGE-SECTION of the program and printed on the first program heading line in accordance with your Print Chart.

2. The Page-Number should be incremented each time the heading lines are reprinted and it will be placed on the second program heading line in accordance with the format shown on your Print Chart.

3. The Column Header Lines shall be double spaced from the second program heading line.

4. The first detail line will be printed after the second column heading and double spaced from this second column heading line.

E. COBOL will be the programming language.

Output Report Line Format

PRINT POSITIONS	FIELD NAME	COMMENTS
	DETAIL LINE	
	**********	
1-2	FILLER	PROVIDE LEFT MARGIN.
3-5	SOC-SEC-NBR	FIRST THREE DIGITS.
6	FILLER	PROVIDE A HYPHEN (/).
7-8	SOC-SEC-NBR	NEXT TWO DIGITS.
9	FILLER	PROVIDE A HYPHEN (/).
10-13	SOC-SEC-NBR	LAST FOUR DIGITS.
14-16	FILLER	PROVIDE SPACING.
17-34	EMPLOYEE NAME	LAST NAME FIRST.
35-39	FILLER	PROVIDE SPACING.
40	SHIFT-CODE	ONE DIGIT.
41-45	FILLER	PROVIDE SPACING.
46-50	HOURLY-RATE	ZERO-SUPPRESS NON-SIGNIFICANT DOLLAR POSITION ZEROS INSERT DECIMAL POINT.
51-54	FILLER	PROVIDE SPACING.
55-60	TOTAL-HOURS	ZERO-SUPPRESS NON-SIGNIFICANT ZEROS, INSERT DECIMAL POINT PROVIDE FOR SIGN.
61-63	FILLER	PROVIDE SPACING.
64-70	REGULAR-HOURS	ZERO-SUPPRESS NON-SIGNIFICANT ZEROS, INSERT DECIMAL POINT PROVIDE FOR SIGN.
71-73	FILLER	PROVIDE SPACING.
74-80	OVERTIME-HOURS	ZERO-SUPPRESS NON-SIGNIFICANT ZEROS, INSERT DECIMAL POINT PROVIDE FOR SIGN.
81-82	FILLER	PROVIDE SPACING.
83-89	SHIFT-DIFF	ZERO-SUPPRESS NON-SIGNIFICANT DOLLAR POSITION ZEROS INSERT COMMA & DECIMAL PT.
90-92	FILLER	PROVIDE SPACING.
93-101	REGULAR-EARNINGS	ZERO-SUPPRESS NON-SIGNIFICANT DOLLAR POSITION ZEROS INSERT COMMA & DECIMAL PT.
102-104	FILLER	PROVIDE SPACING.
105-113	OVERTIME-EARNINGS	ZERO-SUPPRESS NON-SIGNIFICANT DOLLAR POSITION ZEROS INSERT COMMA & DECIMAL PT.
114-116	FILLER	PROVIDE SPACING.
117-125	TOTAL-EARNINGS	ZERO-SUPPRESS NON-SIGNIFICANT DOLLAR POSITION ZEROS INSERT COMMA & DECIMAL PT.
126-132	FILLER	PROVIDE SPACING.
	PROGRAM HEADER LINE #1	
	**********	
1-3	FILLER	PROVIDE LEFT MARGIN.
4-40	COMPANY NAME	
41-116	FILLER	
117-118	MONTH	ONE OR TWO DIGITS.
119	FILLER	PROVIDE SLASH (/).
120-121	DAY	ONE OR TWO DIGITS.
122	FILLER	PROVIDE SLASH (/).
123-124	YEAR	TWO DIGITS.
125-132	FILLER	PROVIDE RIGHT MARGIN.
	PROGRAM HEADER LINE #2	
	**********	
1-3	FILLER	PROVIDE LEFT MARGIN.
4-20	REPORT NAME	
21-116	FILLER	
117-121	LABEL	"PAGE ".
122-124	PAGE NBR	UP TO 3 DIGITS.
125-132	FILLER	PROVIDE RIGHT MARGIN.
	COLUMN HEADER LINE #1	
	**********	
1-20	FILLERS & LABELS	" SOC SEC ".
21-40	FILLERS & LABELS	" EMPLOYEE SHI".
41-60	FILLERS & LABELS	"FT HOURLY TOTAL".
61-80	FILLERS & LABELS	" REG O.T.".
81-100	FILLERS & LABELS	" SHIFT REGULA".
101-120	FILLERS & LABELS	"R OVERTIME TO".
121-132	FILLERS & LABELS	"TAL ".
	COLUMN HEADER LINE #2	
	**********	
1-20	FILLERS & LABELS	" NUMBER ".
21-40	FILLERS & LABELS	" NAME CO".
41-60	FILLERS & LABELS	"DE RATE HOURS".
61-80	FILLERS & LABELS	" HOURS HOURS".
81-100	FILLERS & LABELS	" DIFF EARNING".
101-120	FILLERS & LABELS	"S EARNINGS EARN".
121-132	FILLERS & LABELS	"INGS ".

Input Record Format

FIELD POSITIONS	FIELD NAME	DATA CLASS	COMMENTS
1-2	FILLER		SKIP CODE "L4"
3-11	SOC SEC NBR	NUMERIC	
12-29	EMPLOYEE NAME	ALPHANUMERIC	
30-34	REGULAR-HOURS	NUMERIC	
35-39	OVERTIME-HOURS	NUMERIC	
40-45	FILLER		DATA NOT USED
46-49	HOURLY-RATE	NUMERIC	
50-69	FILLER		DATA NOT USED
70	SHIFT-CODE	NUMERIC	
71-80	FILLER		DATA NOT USED

Definition Of Input Records In Working Storage

In Program #5, we defined the Output Record Field Description in WORKING-STORAGE-SECTION. Program #6 will provide the Input Record Field Descriptions in WORKING-STORAGE-SECTION as well. (Review "COBOL Corner VI" for more information.) The coding changes for this method follow:

```
FILE SECTION.

FD  FILEL4
    VALUE OF FILE-ID IS "A:FILEL4.DAT"
    RECORD CONTAINS 80 CHARACTERS
    LABEL RECORDS ARE STANDARD.

01  PAYROLL-RECORD.
    05  FILLER                          PIC X(80).
```

Observe that only one (1) 01-level record-description entry has been defined with no field descriptions. Only a FILLER specification for 80 characters has been specified. Actually, PAYROLL-RECORD in Program #6 will be used for every employee listed in the Input Record File and they will all use the same format. The format will be defined as follows in the WORKING-STORAGE-SECTION.

Working-Storage-Section (Showing The Input Record)

```
 WORKING-STORAGE-SECTION.

01  PR-PAYROLL-RECORD.
     05  FILLER                         PIC X(02).
     05  PR-SOC-SEC-NBR.
         10  PR-SOC-SEC-NBR-1           PIC 9(03).
         10  PR-SOC-SEC-NBR-2           PIC 9(02).
         10  PR-SOC-SEC-NBR-3           PIC 9(04).
     05  PR-EMPLOYEE-NAME               PIC X(18).
     05  PR-REGULAR-HOURS               PIC 9(03)V99.
     05  PR-OVERTIME-HOURS              PIC 9(03)V99.
     05  FILLER                         PIC X(06).
     05  PR-HOURLY-RATE                 PIC 9(02)V99.
     05  FILLER                         PIC X(20).
     05  PR-SHIFT-CODE.
         88  SHIFT-CODE-1                       VALUE 1.
         88  SHIFT-CODE-2                       VALUE 2.
         88  SHIFT-CODE-3                       VALUE 3.
     05  FILLER                         PIC X(10).
```

Note: I will discuss the 88-levels a little later in this article.

Here are a few other WORKING-STORAGE-SECTION changes I would like to suggest for Program #6 using 88-conditions:

```
01  WS-SWITCHES.
    05  WS-EOF-SWITCH           PIC X(03).
        88  END-OF-FILE                    VALUE "YES".

01  WS-REPORT-CONTROLS.
    05  WS-PAGE-COUNT           PIC S9(03)  VALUE +1 COMP-3.
        88  FIRST-PAGE                     VALUE +1.
    05  WS-LINES-USED           PIC S9(02)  VALUE ZERO
                                             COMP-SYNC.
        88  FULL-PAGE                      VALUE +50.
    05  WS-LINE-SPACING         PIC S9(02).
```

Note: The above changes are a couple of more applications for 88-level usage. I hope to clear these up with the discussion of LEVEL 88 Condition Names later in this article.

```
01  WS-WORK-AREAS.
    05  WS-SHIFT-DIFF               PIC S9(03)V99  COMP-3.
    05  WS-REGULAR-EARNINGS         PIC S9(04)V99  COMP-3.
    05  WS-OVERTIME-EARNINGS        PIC S9(04)V99  COMP-3.
    05  WS-TOTAL-HOURS              PIC S9(03)V99  COMP-3.
    05  WS-TOTAL-EARNINGS           PIC S9(04)V99  COMP-3.
```

A WORK-AREA is nearly always required in a COBOL program that performs computations. This provides a temporary storage area for the results of the computation until the programmer is ready to use the results again. Many programs will require an ACCUMULATOR AREA as well. Do not confuse one with the other. We have discussed the PIC clause in prior articles. If you do not remember, review the previous "COBOL Corner" articles. In fact, this is a good idea to do from time to time as a refresher!

The USAGE clause, which I have used in the common abbreviated form — COMP-3, defines a packed (internal decimal) field. Back a few years, memory was very expensive and thus at a premium. Today this clause is optional and many programmers do not use it, as most computers have sufficient memory. You may want to read about COMPUTATIONAL-3 USAGE clause in your Manual. I used it here so that you would know what it is if you were to find a program using it.

Level 88 Condition Names

Modern COBOL programmers use level 88 condition-name entries often. Therefore, we must learn to use them. It is a hard item to teach except by learning by using them. That is the way I am going to do it! We will learn by using the 88-level over and over anywhere we can find an application place. (Please read the Level 88 Condition-Names section in your Manual.) 88-levels are a special way of writing a relation condition in the DATA DIVISION or the WORKING-STORAGE-SECTION. For now, we are going to use them in WORKING-STORAGE. After the level number 88, the user-defined condition-name is specified, followed by a VALUE clause with a literal or literals that apply to the value of that condition. The level-88 item MUST immediately follow the definition of the field with which it is associated. The VALUE clause may contain multiple values or a range of values. A condition-name VALUE clause is required for each level-88 item description. The general format for the VALUE clause follows:

```
VALUE   literal-1[literal-2...].
VALUES  literal-1 THRU literal-2.
```

An example might clear up all of those words! Here is a look at a level-88 in Program #6:

```
01  WS-SWITCHES.
    05  WS-EOF-SWITCH               PIC X(03).
        88  END-OF-FILE                        VALUE "YES".
```

In the PROCEDURE DIVISION we will test for the 88-condition:

```
PERFORM 200-PROCESS-EARNINGS-REPORT
    UNTIL END-OF-FILE.
```

If END-OF-FILE is "YES", we have met the condition! If it is "NO", we repeat the process.

Look at the FIRST-PAGE and FULL-PAGE 88-conditions we specified a few paragraphs back. They work in a similar way, except we use an IF statement in the PROCEDURE DIVISION to make the test.

Looking further, a "nested IF statement" in the PROCEDURE DIVISION performs the test for the SHIFT-CODE 88-condition. Did you look back where we specified these 88-conditions?

To understand the 88-condition, try asking yourself "is the 88-item true or false". That is what the program is really testing.

Nested IF Statement

The IF statement is a powerful COBOL statement. We have already used the IF statement in a previous program to test for when to change to a new page. Do you remember? If not, look back at your previous articles. Now, we are going to use the reserved word IF more than once within an IF statement, which is referred to as a "nested" IF. IF statements can be classified as "linear" or "non-linear". More on that later. Also, we are going to use the IF statement with the reserved word ELSE. The ELSE statement forms the IF then ELSE structure. We have used these in our Program #6. Here is an example of the "nested IF then ELSE" from our program in the PROCEDURE DIVISION:

```
200-PROCESS-EARNINGS-REPORT.
    XXXXXXXX
    XXXXXXXX
    IF SHIFT-CODE-1
        MOVE ZEROS                          TO WS-SHIFT-DIFF
    ELSE
        IF SHIFT-CODE-2
            COMPUTE WS-SHIFT-DIFF ROUNDED = .1
                * WS-TOTAL-HOURS
                * PR-HOURLY-RATE
        ELSE
            COMPUTE WS-SHIFT-DIFF ROUNDED = .125
                * WS-TOTAL-HOURS
                * PR-HOURLY-RATE.
    XXXXXXXX
    XXXXXXXX
```

That "nested IF then ELSE" statement looks pretty simple, but it is very powerful!

1. I made it easy to see by proper indentation. Do you see this?
2. Did you find the statement period? All of the lines form ONE IF then ELSE statement.
3. This is a "linear" nested IF statement. (Definition will follow in the next paragraph.)
4. Did you notice that I did not use SHIFT-CODE-3? I assumed that if it was not SHIFT-CODE-1 or SHIFT-CODE-2, it had to be SHIFT-CODE-3.

The "linear" nested IF is relatively simple to write and understand, but its use is limited to situations where one condition (88-condition in our case) applies to each action statement group, such as when one field is being tested for various values. This form is called "linear" because each ELSE immediately follows the IF condition, one after the other. Comparisons (Tests) are made until a true condition is encountered (or in some cases but not this one, until the end-of-sentence is reached). When the true condition occurs, the specified action (MOVE or COMPUTE in this case) is executed. Indentation for the "linear" nested IF is handled as shown in the example above: The first IF and ELSE are aligned and each action is indented from its IF and ELSE. The next IF and ELSE are aligned and indented from the preceding ones. Thus, it is easy to see that the IF and ELSE are properly matched.

We will define the "non-linear" nested IF when we use it in a future program. It is somewhat more complicated, but with proper indentation, it can be followed.

The complexity of nested IF statements is reduced when:

1. The programmer thoroughly understands how ELSE statements are paired with IF conditions.
2. Proper indentation (this cannot be repeated too often) is used when the IF statement is coded.
3. The number of levels of nesting is limited to three or four.

READ INTO

We introduced the READ statement some articles back and we have been using a READ statement in every program. Now, I would like to show you an example of the READ statement with the INTO phrase. I would like to have you use this example in Program #6:

```
800-READ-PAYROLL-RECORD.
    READ FILEL4
        INTO PR-PAYROLL-RECORD
            AT END
                MOVE "YES"              TO WS-EOF-SWITCH.
```

The READ statement with the INTO phrase will read a record from the Input File and transfer the data read from the file to the location written after the reserved word INTO. This INTO phrase is used when fields have been defined in WORKING-STORAGE-SECTION as we have done in Program #6. If this method is not used, the READ statement followed by a MOVE statement would be required as follows (NEVADA COBOL readers note!):

```
800-READ-PAYROLL-RECORD.
    READ FILEL4
        AT END
            MOVE "YES"                  TO WS-EOF-SWITCH.
    MOVE PAYROLL-RECORD                 TO PR-PAYROLL-RECORD.
```

Procedure Division Program #6 Suggestions

I am going to leave most of Program #6 coding for the readers to do. Here are a few last suggestions:

```
000-PRINT-EARNINGS-REPORT.
    OPEN  INPUT FILEL4
          OUTPUT EARNINGS-REPORT.
    PERFORM 100-INITIALIZE-VARIABLE-FIELDS.
    PERFORM 800-READ-PAYROLL-RECORD.
    PERFORM 200-PROCESS-EARNINGS-REPORT
        UNTIL END-OF-FILE.
    CLOSE FILEL4
          EARNINGS-REPORT.
    STOP RUN.
```

This looks pretty normal. NO surprises! I have discussed the 200 and 800 above. I will show the 100 below:

```
100-INITIALIZE-VARIABLE-FIELDS.
    MOVE "NO "                          TO WS-EOF-SWITCH.
    MOVE WS-YEAR                        TO H1-YEAR.
    MOVE WS-MONTH                       TO H1-MONTH.
    MOVE WS-DAY                         TO H1-DAY.
    MOVE ZEROS                          TO WS-PAGE-COUNT.
```

As sort of a test of your understanding of 88-conditions, do you see anything about the above coding that you would change? It could be improved. That is all I will say! Remember there is always more than one way to code a program.

Closing

Review what steps you must do to complete Program #6:

1. Prepare the Structure Chart-if you will look at the Module-000 that I supplied above, you will have the information that is needed.
2. Prepare the Flowchart-the Program Specifications provide the necessary design data.
3. Prepare the Print Chart-look back at the Output Report Line Format.
4. Prepare the Record Chart-look back at the Input Record Format.
5. Code the program on CODING SHEETS-I have supplied a lot of hints and suggestions. Also, refer to Program #5.
6. KEY-IN the above code.
7. Use PIP to transfer FILEL4.DAT from HUG COBOL Corner Disk-I to your "A" disk.
8. COMPILE your PRGM06.COB source file. Did you get any ERROR messages? If you did, I bet you did not do a walk-through! I did not list this step above to see if you would remember the step. Did you catch it? If you found any ERRORS, go back and review you code and correct it using your Editor. Compile again. You should not have any

ERRORS this time if you did a careful job. If you still have ERRORS and YOU cannot find the ERRORS, prepare another "A" disk as we have discussed in previous articles many times. Use PIP and transfer PRGM06.COB and FILEL4.DAT from your HUG COBOL Corner Disk-I to this new disk. Compile this disk and print a hard-copy listing of the program code. You should not have any ERRORS. Now, compare this code with your program code, correct your code with the Editor and Compile again. If you have been careful, you should have a "clean" ERROR FREE compile.

9. LINK/EXECUTE your PRGM06.REL and print your Report. The Report should match your Print Chart and my Program Specifications! Does it? If not, recheck your documentation, your Print Chart, and then your Coding. Make the needed corrections and repeat the above steps. Do you now have a GOOD REPORT?

You should be sure to learn these program steps. I do not want to use the "COBOL Corner" space for them again. Our following programs are going to be harder and I will need all the space to help you with the new programming.

Next month, we will start one of COBOL'S real challenges in programming. I found it to be one of the hardest Report Formats to learn — CONTROL BREAKS. *

"My Favorite Subroutines"

Dear HUG,

Here is a little subroutine I use often. Maybe other members will also find it useful.

The program prints a design that can be used to layout screen displays. The top and left side have symbols used for direct cursor addressing, and the bottom and right side for user defined addressing using row and column numbers.

```
10 DEFSTR A,B:A=CHR$(27):BRE=A+"c":BGR=A+"[10m":
   BGX=A+"11m":B10=A+"[w"'. . . COMMAND SET UP
20 '
30 '.................L A Y O U T
40 '           printing screen layouts
50 '
60 '
70 ' OBJECT  To print hard copy of screen layout for
     correct addressing and designing graphic layouts.
80 ' REQ. Program is in MBASIC (BASIC-80 Rev. 5.21
     using CP/M "R" 2.2) driving a Zenith 25 printer.
90 '
100 LPRINT BRE B10" ";:FOR X=32 TO 111:LPRINT CHR$(X);:
    NEXT
110 LPRINT BGR:FOR X=32 TO 56 STEP 2
120 LPRINT CHR$(X);:FOR Y=1 TO 8:LPRINT"i i i i i!";:
    NEXT:LPRINT X-31
130 LPRINT" ";:FOR Y=1 TO 40:LPRINT"i ";:NEXT:LPRINT:
    IF X=56 GOTO 160
140 LPRINT CHR$(X+1);:FOR Y=1 TO 40:LPRINT" i";:NEXT:
    LPRINT X-30
150 LPRINT" ";:FOR Y=1 TO 40:LPRINT" i";:NEXT Y:LPRINT:
    NEXT X
160 LPRINT" 12345678911111111112222222223333333333
    444444444455555555556666666666777777777778"
170 LPRINT"          01234567890123456789012345678
    9012345678901234567890123456789012345678901234567890"
180 LPRINT BRE
```

Sidney Allen
2012 East 14th
The Dalles, OR 97058

Dear HUG,

I've been an enthusiastic reader of "My Favorite Subroutines" since its inception and find it interesting and informative. A welcome addition to a fine publication. Keep up the good work.

My first submission May '84, was a two line loop that would provide a one or more second delay in a program regardless of whether it was interpreted or compiled. In the July '84 issue, there appeared a more elegant version of this idea by George Holt using the WHILE/WEND command.

While both of these delay subroutines will function as given, their accuracy is dependent on when they start. That is, how much of the current second has elapsed when the delay routine is encountered in a program. The actual elapsed time produced by the first second of delay will vary between zero and one second.

After a little thought I was able to overcome this problem and provide an additional feature as well, time delays of less than one second. The listing below will provide accurate delays in both the interpreted and compiled versions with a minimum range of .1 second to 30 minutes.

```
N     A=TIME : WHILE A+1 > TIME : WEND
          '(thank you, George Holt)
N+1   FOR B=0 TO 1000 : IF A+2 > TIME THEN NEXT
N(P) FOR C=0 TO B*D   : IF A-2 < TIME THEN NEXT
```

Lines N and N+1 should be placed or called in the beginning of the program and need to be read only once. Line N waits until any fractional part of a second remaining, when the routine is first encountered, has elapsed. Line N+1 retains as a variable B, the number of times this FOR NEXT loop will iterate in the next full second under the conditions extant. Line N(P) may be placed in the program as often as, and any place that, a delay is needed. The product B*D establishes the resultant time delay in seconds. Note that the IF THEN statement in line N(P) does nothing and is there only to match the execution time of lines N+1 and N+(P) in all operating modes.

The value for the variable D can be specified in the initial part of the program and/or fixed values from .1 to 1800 seconds may be inserted as required. I have tested this routine on my H-100, (Z-BASIC), in which a CDR speed module is installed. The accuracy of the delay routine is maintained whether it is run at 5 or 7.5 megahertz, interpreted, or compiled.

I realize that all of this is hardly of earth shaking importance, but it was fun working it out and it is even more fun sharing.

Arthur Calhoun
16 Cedar Valley Lane
Huntington, NY 11743

Dear HUG,

This is a handy little subroutine I "developed" to determine the CPU type. It can determine whether an 8080, 8085, or Z80 CPU is in use. The outstanding features of this code are that absolutely all instructions used are legal on all three processors (e.g. I didn't try using Z80 opcodes which would translate to illegal 8080 or 8085 instructions) and its small size. It is reliable and accurate in its determination. This magic is possible because of very subtle, almost insignificant (ALMOST), differences in the way in which certain flags are affected.

Without getting bogged down in details, the routine first checks for a Z80 processor. This test uses the fact that the PARITY flag in the 8080 and the 8085 serves a dual purpose in the Z80 processor. This flag has the additional function of indicating OVERFLOW

for certain operations in the Z80. If the routine determines that the processor is NOT a Z80, it then checks whether an 8080 or 8085 is used. The 8080 and 8085 affect the Auxiliary Carry (AC) in different manners when and AND (either ANA or ANI) operation is performed. In the 8080, the AC flag is always cleared, while in the 8085 the AC flag is always set after either of these instructions is executed.

Now, without further adieu, here is the routine:

*** Perform check for Z80:**

```
MVI     A,127       ; 127 is highest 8-bit signed value
INR     A           ; bump one
JPE     ISZ80       ; overflow if Z80
```

*** Have an 808x. Decide which one:**

```
XRA     A           ; make 0
ANA     A           ; set/clear AC
DAA                 ; 0 is 8080, 6 if 8085
JNZ     IS8085
```

*** Have an 8080**

```
IS8080    < do 8080 routine >
```

*** Have an 8085**

```
IS8085    < do 8085 routine >
```

*** Have a Z80**

```
ISZ80     < do Z80 routine >
```

Bill Parrott
7010 Caenen
Shawnee, KS 66216

Dear HUG,

Here is a small program that demonstrates the "TILING" option of the PAINT command in BASIC version 2 (MS-5163-13). The program generates a color palette of 125 different colors.

```
10 ' Color Palette for the PAINT function of BASIC v2.
20 '
30 ' M. Brenner   1/85
40 '
50 ' This program demonstrates the use of tiling (fill
55 ' patterns) with the PAINT command. It generates a
60 ' color array of 125 different colors by varying the
70 ' number of pixels of each color which are inserted
80 ' in each box.
90 '
100 ' Requires Z-100 (or Z-100 PC with Z-319 board)
110 '
120 COLOR 7,0
130 KEY OFF
140 CLS                     'Clear screen to black
150 R=0:G=0:B=0             'Initialize pattern selectors
160 FG=1                    'Set the box outline color
170 '
180 ' Define arrays for 5 patterns and their complements
190 '
200 P$(0)=CHR$(&H0) : Q$(0)=CHR$(&H0)
    '                  0 of 8 (00000000 00000000)
210 P$(1)=CHR$(&H1) : Q$(1)=CHR$(&H10)
    '                  1 of 8 (00000001 00010000)
220 P$(2)=CHR$(&H11): Q$(2)=CHR$(&H44)
    '                  2 of 8 (00010001 01000100)
230 P$(3)=CHR$(&H55): Q$(3)=CHR$(&HAA)
    '                  4 of 8 (01010101 10101010)
240 P$(4)=CHR$(&HFF): Q$(4)=CHR$(&HFF)
    '                  8 of 8 (11111111 11111111)
250 '
260 ' Loop for all 125 boxes (25x5)
270 '
280 FOR I=0 TO 24
290 FOR J=0 TO 4
300 K=I*25
310 L=J*40
320 LINE(K,L)-(K+25,L+40),FG,B        ' Draw a box
330 '
340 '  ¦ 1st row of bits ¦  ¦ 2nd row of bits ¦
350 C$=P$(B)+P$(G)+P$(R)    +Q$(B)+Q$(G)+Q$(R)
    ' build the tiling string
360 '
370 IF R=0 AND G=0 AND B=0 THEN GOTO 400 'Skip 1st one
380 PAINT(K+1,L+1),C$,FG     'Paint box with pattern
390                          'Update pattern selectors
400 R=R+1:IF R>4 THEN R=0:G=G+1:IF G>4 THEN G+0:
    B=B+1:IF B>4 THEN B=0
410 NEXT J
420 NEXT I
430 LOCATE 23,1
```

Mike Brenner
Zenith Data Systems

Dear HUG,

The enclosed listing is for your "Favorite Subroutines" column. The subroutines, CINP and COUT, are for use with C/80 from Software Toolworks. They are emulations of the INP and OUT special functions provided with Microsoft BASIC.

The main program is a simple terminal emulator which serves to illustrate the use of the routines. The routines themselves are examples of the "inline assembly language" capability of C/80. Both of these routines use self-modifying code to load the port address into the IN or OUT instruction.

```
}
/*****************************************************
    CINP.C - This routine is an emulation of the INP(I)
    function in Microsoft BASIC. The function returns
    the value input from port i, and is called by the
    statement:  cinp(i);
*****************************************************/
cinp(port)
{
#asm
     POP     B      ;pop return address to BC
     POP     D      ;pop port to DE
     LXI     H,$+5  ;load HL with address of IN instr.
     MOV     M,E    ;move port address to IN instruction
     IN      0      ;input from port
     MOV     L,A    ;move input byte from A to L
     PUSH    D      ;pushes must equal pops
     PUSH    B      ;push return address back on stack
#endasm
}
#include "stdlib.c"

main() /* MODEM.C - TERMINAL EMULATOR PROGRAM FOR H-89 */
{
      static int port=216, stat=221;
      int byte;
      /* Set port up for 1200 baud 8 bit operation */
      cout(port+3,131);
      cout(port+4,3);
      cout(port,96);
      cout(port+1,0);
      cout(port+3,3);
      cinp(port);
      /* Main loop */
      while((byte=bdos(6,255)) !=4)
      {
            while(byte>0)
            {
                  if(cinp(stat) & 32)
                  {
                        cout(port,byte);
                        byte=0;
```

```
                }
            }
            if(cinp(stat) & 1) putchar(cinp(port));
        }
        exit();
}
/*****************************************************
     COUT.C - This routine is an emulation of the
     OUT I,J function in Microsoft BASIC. The purpose is
     to output the byte j to the port number i.
     The function is called by the statement:
     cout(i,j);
*****************************************************/
cout(port,byte)
{
#asm
     POP    B       ;store return address in BC
     POP    D       ;pop byte into DE
     MOV    A,E     ;move byte to A
     POP    D       ;pop port into DE
     LXI    H,$+5   ;load address of OUT instruction
     MOV    M,E     ;move port value into instruction
     OUT    Ø       ;output byte to port
     PUSH   D       ;the number of pushes must
     PUSH   D       ;equal the number of pops
     PUSH   B       ;put return address back on stack
#endasm
```

Allen Gilchrist, Jr.
Route 2 Box 827
Rosharon, TX 77583

Dear HUG,

I have another little tid bit which you might like to include in a future REMark. It started when I tried to get the CONDOR file management system to accept special commands for various printer options.

In normal operation, CONDOR file management system will not allow special characters to be sent to the printer. Because of this, it is difficult to switch back and forth between different print functions without exiting CONDOR and loading either another program or using the command sequence as described by Charles Layman (Feb. 1984 REMark, page 16). There is a solution even though it is somewhat lengthy. Here's how it's done.

Copy the PRINT utility distributed with ZDOS (version 1.0) over to your CONDOR A: disk. Rename it to PRINT1.DBM. It is important that the extension of this print utility be called .DBM. CONDOR now thinks PRINT1.DBM is another command file and will run it as though it was part of the DBMS system.

Now generate a file(s) as described by Charles Layman in the February issue of REMark, but do not use an extension. If your printer uses CTRL-(something) for its commands, the appendices (B.1) in the Z-100 User's Manual define which CTRL-(something) is generated by which Z-100 key. In my case (an Okidata 82A), I needed and ASCII GS (Decimal 29) to change to 16CPI. This is a CTRL] on the Z-100.

Using EDLIN, make a file called 16CPI. (remember, no extension) and in the first line put only a CTRL]. EDLIN will respond with

```
1*.]
2*.Z.C
```

Be sure to put a CTRL-Z and then a CRTL-C after the 2*. End (E) EDLIN to save the file called 16CPI. Make sure it is saved on the B: disk.

PRINT1 can be called directly with the file to run or it can be included in a batch file (CONDOR defines it as command file), for example:

```
PRINT1 16CPI
```

or in a command file:

```
PRINT1 16CPI
     ; causes printer to change to 16CPI.
PRINT DB BY DATE,TIME,DISTANCE, etc.
     ; causes normal CONDOR print command to run.
PRINT1 1Ø CPI
     ; causes the printer to change back to 1Ø CPI.
```

The only disadvantage to this technique is that it takes time for the two print routines to load, run, and then load and run again. If time is not a problem or you have a print buffer installed, it goes quickly.

Larry T. Wier
1068 149th Place S.E.
Bellevue, WA 98007

*

Vectored from Page 16

```
in_loop:
     . . .
in_loop_wait:
     . . .

; convert time to ZDOS format

     pop  si                ;reset pointer to in_buffer
     mov  al, out_buffer+9  ;get tenths seconds
     cmp  al, '?'           ;bad value?
     jne  tok               ;good value, go ahead
     mov  al,0
     mov  [si+9], al        ;put zero for tenths second
tok:
     call short convert     ;get hours
     . . .
```

Sincerely,

Charles Kingston
6 Surrey Close
White Plains, NY 10607

Has Anyone Installed A Micro-Science Hard Disk Drive?

Dear HUG:

I do not have a modem, so I can't check the bulletin board yet, but I do need some information on the HS-151. Has anyone out there installed a Micro-science 10 or 20Mb hard disk drive in an HS-151? This is the drive marketed by Qubie, QIC, Kamerman Labs, etc., and seems to be pretty popular. I am anticipating buying one and want to avoid the expensive pitfalls or traps that sometimes come with third-party hardware purchases.

Any information you can provide will be appreciated. The disk drive manufacturers always paint a rosy picture of compatibility and installation ease which often fades when the bag of parts arrives with no instructions. Some first hand experience from a Micro-science/HS-151 user would be reassuring.

Sincerely,

Robert E. Hawkins
Consulting Engineer
P.O. Box 4533
811 Highway #1 South
Greenville, MS 38704-4533

Graphics Layout For The Z-100

Dear HUG,

Recently, I wanted to work out a graphics display on my Z-100, and remembering prior struggles with graph paper and conversion factors, I dug through my back issues of REMark to find John Elam's layout sheet (August 1984, Pg. 64). After some playing with his idea, I decided to take a different approach and here is what I came up with.

In ZBASIC (Rev. 1.0), using ZDOS (Vers. 1.25) and a C. Itoh Model 8510A "Prowriter," the program prints a grid the long way on a sheet, permitting a complete representation of all 224 vertical and 639 horizontal pixels. By using the condensed type and incremental vertical spacing features of the "Prowriter," the resulting grid is very close to the 7 by 9 inch face of the ZVM-123 monitor. The grid depicts each even 10 pixels in both directions, except that the lowest vertical grid line depicts pixel row 224. The origin depicts 0 pixels in both directions: they cannot be used, of course. And the rightmost horizontal grid line depicts pixel column 640, which also cannot be used. Pixel row and column references appear on all four edges of the grid and an "X" appears at the center position.

Using friction feed and a sheet of translucent paper, a grid can be printed that can be used to trace a graphic from another source. If the source is not too large for the grid, the center "X" can be used to assist in centering. If the source is smaller than the grid, trace the graphic in the upper left corner leaving appropriate left and upper margins, construct the right and lower margins, and divide those margin values into 639 and 224, respectively. The SMALLER number is used as a multiplier for BOTH horizontal and vertical grid values of points in the traced graphic. If the source is too large for the grid, it is possible to print four grids, trim the edges and tape them together. It is possible to modify the program to print revised pixel row and column references in this case, but somebody else can do that.

Something should be said about accuracy, though it is hard to imagine that, given the inherent distortions of the monitor, anybody would care. The grid is approximately 6% undersize, that is the vertical representation is 6.58823529411 inches (.94117647058 of 7 inches) and the horizontal representation is 8.43124999971 inches (.9368055552 of 9 inches). This means that the grid represents the entire screen, but that the actual screen display will be 6% larger than the graphic on the grid.

Notice that the two dimensions differ slightly in the amount of error, but this should not be a problem in most situations. The display on the screen will be stretched horizontally in proportion to the vertical dimension by a factor of .4% (the factor is 1.00466576552). These calculations assume that the printer prints exactly 17 characters per inch across and 19/144 inches per line down, which isn't likely using friction feed.

Hope this makes graphics layout easier for a few HUGgies.

Sincerely,

David K. Wheeler
306 Winslow Street
Watertown, NY 13601

```
100 '  PROGRAM LAYOUT: to print a Z-100 graphics layout
110 '      sheet on a C. ITOH Model 8510A printer.
120 CLS:LOCATE 6,1,0
130 PRINT "This program prints a graphics layout sheet for
      the Z-100. The"
140 PRINT "  long dimension (down the page) of the layout
      sheet represents"
150 PRINT "  the horizontal dimension (across) of the
      monitor. The short"
160 PRINT "  dimension (across) of the layout sheet repre-
      sents the vertical"
170 PRINT "  dimension of the monitor. The layout sheet is
      slightly small-"
180 PRINT "  er (6%) than the monitor. "
190 PRINT:PRINT
      "Will you be using continuous feed paper? (Y/N)"
200 P$=INKEY$:IF P$="" GOTO 200
210 IF P$="Y" OR P$="y" THEN SHEET$="CONT":GOTO 230
220 SHEET$="SING"
230 CLS:LOCATE 10,10,0:PRINT
      "Make sure printer is ready. Press RETURN when ready."
      :BEEP
240 A$=INKEY$:IF A$="" GOTO 240
250 CLS
260 LOCATE 12,24,0:PRINT "Printing graphics layout sheet..."
270 E$=CHR$(27) 'e$=escape
280 LPRINT E$+"N" 'set pica pitch (10CPI)
290 LPRINT E$+"L005" 'set left margin to five
```

```
300 LPRINT "Z-100 GRAPHICS LAYOUT SHEET:
    Title"+STRING$(25,95)+"Date"+STRING$(10,95)
    'print title line
310 LPRINT E$+"Q" 'set condensed pitch (17CPI)
320 V=220 'set vert. pixel indicator counter
330 LPRINT "    ";V;:
    'print first vert. pixel indicator at top
340 FOR J=1 TO 22
    'print remaining vert. pixel indicators at top
350   V=V-10  'increment vert. pixel indicator counter
360   IF V<10 THEN LPRINT USING "###";V;:GOTO 380
370   LPRINT USING"####";V;:LPRINT " ";:
380   NEXT J
390 LPRINT E$+"T19" 'set 19/144 in/line (7.6 lines/inch)
400 H=0 'set horiz. pixel indicator counter
410 FOR I=1 TO 65 'print grid
420   LPRINT USING "###";H;:LPRINT E$+"#";
        :LPRINT CHR$(87)+CHR$(81)+CHR$(81)+CHR$(87);:
        'print left horiz. pixel ind. & first 4 char. of
         grid line
430   FOR J=1 TO 22 'print remainder of grid line
440     IF I=33 AND J=11
        THEN LPRINT CHR$(81)+CHR$(81)+CHR$(81)+CHR$(80)
        +CHR$(87);:GOTO 460 'print cross in center
450    LPRINT CHR$(81)+CHR$(81)+CHR$(81)+CHR$(81)+CHR$(87);:
460     NEXT J
470   LPRINT E$+"$";:LPRINT H:H=H+10
    'print right horiz. pixel indicator & increment horiz.
    pixel indicator counter
480   NEXT I
490 V=220 'set vert. pixel counter
500 LPRINT "    ";V;:
    'print first vert. pixel indicator at bottom
510 FOR J=1 TO 22
    'print remaining vert. pixel indicators at bottom
520   V=V-10 'increment vert. pixel indicator counter
530   IF V<10 THEN LPRINT USING "###";V;:GOTO 550
540   LPRINT USING"####";V;:LPRINT " ";:
550   NEXT J
560 LPRINT E$+"A"
570 IF SHEET$="CONT" THEN FOR K=1 TO 9:LPRINT:NEXT K
    'advance to top of page
580 CLS
590 LOCATE 6,10,0:PRINT "Print another sheet? (Y/N)":BEEP
600 A$=INKEY$:IF A$="" GOTO 600
610 IF A$="Y" OR A$="y" GOTO 630
620 LOCATE 23,1,1:LPRINT E$+"N";:LPRINT E$+"L001";:END
    'reset left margin to one and reset pica pitch (10CPI)
630 LOCATE 10,10,0:PRINT
    "Make sure printer is ready. Press RETURN when ready."
    :BEEP
640 A$=INKEY$:IF A$="" GOTO 640
650 IF ASC(A$)=13 GOTO 250
660 GOTO 630
```

Patch For MAPLE 2.1.0 To CP/M 2.204

Dear Bob:

Several purchasers of MAPLE 2.1.0 for CP/M-80 in recent days have written me concerning operating the program under CP/M 2.204. Apparently they were sent copies which did not incorporate the patch I sent to HUG about a year ago. Without the patch, MAPLE does not correctly calculate the number of drives present in the system, since the BIOS storage for the active number of drives was changed from 2.203 to 2.204.

The patch involves changes of three consecutive bytes to 0. It would certainly help us if your master disk was patched as described on the attached sheet. For packages already made up, you could possibly insert a copy of the enclosed instructions.

If you are using CP/M 2.204 rather than an earlier version number, MAPLE 2.1.0 requires a patch to properly access the disk drives.

You may make the patch yourself as follows. Be sure to have a backup copy of MAPLE in case patch is entered incorrectly. Load the CP/M utility DDT with the command:

```
DDT MAPLE.COM
```

Change the bytes at addresses 2E8F to 2E91 to 0 using the substitute command of DDT:

```
Screen:       You Type:
-               S2E8F
2E8F DA         00
2E90 94         00
2E91 2E         00
2E92 3E         -
-
```

Now exit DDT with a Control-C and type:

```
SAVE 52 MAPLE.COM
```

Hope this letter finds you and HUG healthy,

Sincerely,

Dr. William C. Parke
1820 S Street NW
Washington, D.C. 20009

MAPLE Patches

Dear Walt:

It may be helpful to MAPLE users to have the following patches published.

Apply patches to MAPLE only if purpose described is desired. Under HDOS, PATCH or UDUMP will work fine in applying the patches. Under CP/M, use DDT S or A command, then SAVE the file with the record number given. Byte patch changes are given in form: Address Old-byte New-byte.

MAPLE 2.0.8c For HDOS

To operate at 4MHz, change the time delay count at 070.015, from 166 to 377.

To use 25th line for Z29 terminal with H8, change the following bytes (addresses shown in split octal and bytes shown in octal). The HELP key becomes functional; the 'UNCLEAR' function is changed to ESC '^'; the twenty-fifth line shows function key definitions, and terminal hardware handshaking is added for the slower Z29 screen.

```
054.016  176  140      103.201  170  346
054.030  110  176      103.202  102  040
054.035  150  060      103.203  076  312
054.037  240  220      103.204  004  177
054.040  067  103      103.205  377  103
054.057  034  030      103.206  046  361
054.060  064  067      103.207  303  323
054.061  330  034      103.210  020  350
054.062  061  064      103.211  052  311
054.063  106  330      103.212  031  315
054.064  054  061      103.213  257  044
054.160  112  060      103.214  175  070
054.162  271  226      103.215  062  303
054.163  065  103      103.216  020  132
054.175  131  271      103.217  103  077
054.176  076  065      103.220  037  315
054.177  365  131      103.221  107  212
054.200  050  076      103.222  076  103
054.201  173  365      103.223  000  303
054.202  051  050      103.224  210  106
```

```
077.340  355  356      103.225  107  054
077.342  040  020      103.226  016  315
077.346  361  303      103.227  000  212
077.347  323  177      103.230  041  103
077.350  350  077      103.231  000  303
103.177  102  333      103.232  000  173
103.200  332  355      103.233  311  051
```

MAPLE 2.1.0 For CP/M On H89

To properly size drive number with CP/M 2.204, change the following three bytes (now in hex). The size of the file is 52 (decimal) records.

```
2E8F  DA  00      2E90  94  00      2E91  2E  00
```

To save SND setting on Exit, change:

```
2F84  21  3A      302C  32  3A      3030  EE  BA
```

MAPLE For CP/M-85 On Z100

To write settings to physical disk rather than virtual disk, add the code which follows with the A command in DDT. When finished, SAVE 42 records.

```
1820 CALL 26D0        26D0 PUSH H
                      26D1 CALL 2140
                      26D4 POP H
                      26D5 JMP 210D
```

To save SND settings on Exit, change the byte at 27D8 from 21 to 3A.

To save current default drive on Exit, change:

```
2857  32  3A      285B  33  FF      285C  09  08
```

These constitute all of the public patches to MAPLE.

Sincerely,

Dr. William C. Parke
1820 S Street NW
Washington, D.C. 20009

Peeved About Serious Omission

Dear Walt:

I just finished reading the article in REMark for March 1985 titled "The Ultimate in Dot Matrix?" by Jim Buszkiewicz. Although I enjoyed the article, I was somewhat peeved because of what I consider to be a serious omission, not only in this article, but on almost all magazine articles which describe and review printers.

No mention was made as to whether the Printek 930 possessed the capability of incremental spacing which allows it to take advantage of the microjustification feature of word processing programs, such as WordStar. To many users, this feature may seem unimportant, but many users do consider it important, otherwise, why would most of the better daisy wheel printers, and a few of the more expensive dot matrix printers incorporate it? I, for one, want my printers to incorporate it. Should not a review article on this type printer give me this information?

It seems to me that an article on microjustification would be in order. Such an article should include the following:

1. A set of precise definitions and explanations of terms such as margin justification, microjustification, and proportional spacing.
2. A description of how a word processing program/printer combination accomplishes microjustification.
3. A list of which of the popular word processing programs and printers, daisy wheel, dot matrix, and other types provide for microjustification. Also, which printers provide an internal program to microjustify proportionally spaced print.

I would like to see future articles on word processing programs and printers include specifications and comments on their microjustification capabilities.

Sincerely,

James G. Barr
6057 S. Lakeview Street
Littleton, CO 80120

Correction On "High Resolution Graphics With CP/M 85," February '85

Dear HUG:

I have a correction for my article in the February REMark issue. In my article, "High Resolution Graphics With CP/M 85," I misinformed the readers about getting version 2.2.103 of CP/M 85 as a free update to registered users. As it turns out, this is not true. Actually, if you are a registered user of CP/M 85, then the update is available for a small fee. I believe for about $24.00 or so. This is almost free though and I believe that the features of CP/M 85 2.2.103 are worth every dollar of the update. I'm sorry about any inconvenience this may have caused any fellow HUGgies.

Yours truly,

Robert B. Owens
1052 Felix Avenue
Windsor, Ontario
Canada N9C 3L5

Won't Work Under MS-DOS Version 2

Dear HUG:

In the January 1984 issue of REMark, Jim Schuster published a program that interfaces the GC1000 Most Accurate Clock with the Z-100. I have been using the program for quite some time under Z-DOS. However, I cannot get it to work under MS-DOS Version 2. I would appreciate any help I can get with modifications to the source code to make the program work under MS-DOS Version 2.

Thanks,

Steve Munger
1970 O'Shea Lane
Marietta, GA 30062

Heath/Zenith Users' Group

Vectored from Page 49

Part Number	Description of Product	Selling Price	Vol. Issue
CP/M			
885-1219-[37]	CP/M Navigational Program	20.00	31
AMATEUR RADIO			
HDOS			
885-8016	Morse Code Transceiver Ver 2.0	20.00	42
CP/M			
885-1214-[37]	CP/M MBASIC Log Book (64k)	30.00	23
885-1234-[37]	CP/M Ham Help	20.00	49
885-1238-[37]	CP/M ASCIRITY	20.00	57
885-8020-[37]	CP/M RF Comp. Aided Design	30.00	44
885-8031-[37]	CP/M Morse Code Transceiver	20.00	57
COMMUNICATION			
HDOS			
885-1122-[37]	HDOS MicroNET Connection	16.00	37
CP/M			
885-1207-[37]	CP/M TERM & HTOC	20.00	26
885-1224-[37]	CP/M MicroNET Connection	16.00	37
885-3003-[37]	CP/M ZTERM (Z100 Modem Pkg)	20.00	34
885-5004-37	CP/M 86 TERM86 and DSKED	20.00	56
885-5005-37	CP/M 86 16 Bit MicroNET Conn.	16.00	61
885-5006-37	CP/M 86 HUGPBBS	40.00	62
885-5007-37	CP/M 86 HUGPBBS Source List.	60.00	62
885-8005	MAPLE (Modem Appl. Effector)	35.00	29
885-8012-[37]	CP/M MAPLE (Modem Program)	35.00	34
885-8023-37	CP/M 85 MAPLE	35.00	45
ZDOS			
885-3019-37	ZDOS 16 Bit MicroNET Connect.	16.00	61
MISCELLANEOUS			
885-0004	HUG Binder	5.75	
885-1221-[37]	Watzman ROM Source Code/Doc	30.00	33
885-4001	REMark Vol. I Issues 1-13	20.00	
885-4002	REMark Vol. II Issues 14-23	20.00	
885-4003	REMark Vol. III Issues 24-35	20.00	
885-4004	REMark Vol. IV Issues 36-47	20.00	
885-4005	REMark Vol. V Issues 48-59	25.00	
885-4500	HUG Software Catalog	9.75	
885-4600	Watzman/HUG ROM	45.00	41
885-4700	HUG Bulletin Board Handbook	5.00	50
885-3015-37	ZDOS SKYVIEWS	20.00	55

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

Changing your address? Be sure and let us know since the software catalog and REMark are mailed bulk rate and it is not forwarded or returned.

CUT ALONG THIS LINE

HUG MEMBERSHIP RENEWAL FORM

HUG ID Number: _____

Check your ID card for your expiration date.

IS THE INFORMATION ON THE REVERSE SIDE CORRECT? IF NOT, FILL IN BELOW.

Name _____

Address _____

City-State _____

Zip _____

REMEMBER - ENCLOSE CHECK OR MONEY ORDER

CHECK THE APPROPRIATE BOX AND RETURN TO HUG

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

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

Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

BULK RATE
U.S. Postage
PAID
Heath Users' Group

POSTMASTER: If undeliverable, please do not return.

P/N 885-2066