$2.50

REMark®

Issue 47 • December 1983

```
Command: _
```

Official magazine for users of HEATH ZENITH computer equipment.

PERSONAL ROBOTS

Peripherals and Software for Personal Robots

NEW

VOICE COMMAND SYSTEM FOR HERO

MICROMATION proudly presents a new peripheral for HEATHKIT'S® HERO-I robot which elevates the robot to a new level of sophistication. We call this peripheral a Voice Command System (or VCS) because it not only consists of a voice recognizer, but also an advanced level machine language program for the robot which actually allows you to program robot movements by voice. We call the voice recognizer VOREC and the voice driven program VOCOL (VOice COmmand Language). Highlights of these two important parts of the VCS are described below.

VOREC

VOREC is a powerful, microprocessor controlled, speech recognition board which mounts next to, and interfaces with, our HERO-I MEMCOM BOARD. The recognizer has the following principal features and specifications:

- Speaker-dependent recognizer with nearly instantaneous word recognition rates.
- Recognition accuracy about 98%.
- Vocabulary of up to 256 words (stored as 16 word groups with 16 words in each group for greater recognition accuracy).
- 16K of onboard static RAM of which 14K is battery backed to retain recognized word parameters during power down.
- RS232 port for receiving commands from, and reporting status and words recognized to, the host (HERO).
- Highly sensitive audio input circuitry requires only an external speaker for audio input rather than a microphone. (This allows robot to receive commands from up to 15 feet away.)
- Utilizes state-of-the-art high speed (HC) CMOS chips and the new CMOS 65CO2 microprocessor for ultra low power consumption. Complete board consumes an incredibly low 45 ma while active and 1 ma when inactive.
- Speech recognition is accomplished by a software algorithm contained in a 2K EPROM. (Future product updates will require only replacement of this EPROM.)

VOCOL

This software is even more amazing than the voice recognition hardware. VOCOL is like a high level language for the robot (such as BASIC) which supports both deferred and immediate execution modes. The only difference is in BASIC you "write it," and in VOCOL you "speak it." The software is provided on an EPROM which plugs into a memory socket on our HERO-I MEMCOM BOARD. VOCOL has the following principal features:

- When first run, the robot talks to you through a voice training session in which you are asked to repeat words in his command vocabulary three (3) times.
- Following this training session, you can literally talk in a program of movements for later execution, or command immediate movement by voice.
- The robot prompts you for a command and when received, repeats it back to you for verification. If verified and if in immediate execution mode, the robot will execute the movement. If in deferred execution mode, the robot proceeds to write a machine language program in his memory for later execution. When your program of movements is complete, you signify this with a "STOP" command. A "GO" command will then cause the robot to execute the program it wrote in memory. After execution, the robot returns to the command mode.
- Complete instructions and installation manual.

The Voice Command System manual contains a complete description of how to use the VOREC board under program control from HERO. The 6808 Source Code for VOCOL is available on an APPLE® DOS 3.3 disk at additional cost. This source code is compatible with the SC-6800 CROSS ASSEMBLER.

VOCOL Source Code **$55.00** (not sold separately)

TOTAL SYSTEM PRICE $595.00

NEW

POET

This is an Artificial Intelligence program similar in concept to STORYTELLER, but more advanced. The program uses an advanced self-programming technique which allows the robot to speak self-generated, random three line Haiku poems on an endless list of subjects. After HERO speaks a poem and likes it enough, he will make a comment about it or do some meaningful body movement.

PRICE: TAPE (machine code) **$20.00**

PRICE: DISK (source code) **$30.00**

HERO MEMCOM BOARD

This product provides a means to develop programs for the robot using a personal computer, and expands the robot's memory with an additional 30K of RAM. This product includes:

- Two 8-bit bi-directional parallel ports with handshaking lines for superfast data transfers between the robot and a computer (connects directly to our APPLE-HERO COMMUNICATOR board), plus two 16-bit timers.
- An RS232 serial port for two-way communications between the robot and any computer having an RS232 serial port.
- Serial communications software in an onboard EPROM which allows uploading/downloading of programs via the serial port.
- Complete instruction manual and schematics.

PRICE $295.00

APPLE-HERO COMMUNICATOR

This product provides the hardware and software necessary to implement two-way high speed parallel communication between an APPLE® computer and a HERO-I robot equipped with our HERO MEMCOM BOARD. It includes:

- A peripheral card for an APPLE that contains two 8-bit parallel ports with handshaking lines, and two 16-bit timers.
- Data transfer software for the APPLE board and for the HERO MEMCOM BOARD burned into two 2716 EPROMS. These programs provide ultra fast two-way communications.
- A disk containing heavily commented 6808 and 6502 source codes for the communications software. These source codes are compatible with the S-C MACRO ASSEMBLER and the S-C 6800 CROSS ASSEMBLER available for the APPLE from the S-C SOFTWARE CORPORATION.

PRICE $159.00

Send check or money order to:

MICROMATION INC.

9104 Red Branch Rd.
Columbia, MD 21045

Add $3.00 for shipping.

For information call:

(301) 730-1237

9 am-5pm Monday through Friday
MasterCard & Visa welcome

CIRCLE 28

Heath/Zenith
Users'
Group

HUG Manager .... Bob Ellerton
Software Engineer .... Pat Swayne
HUG Bulletin Board
and Software Developer .... Terry Jensen
Software Coordinator .... Nancy Strunk
HUG Secretary .... Margaret Bacon

REMark Editor .... Walt Gillespie
Assistant Editor .... Donna Melland

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

REMark is a HUG membership magazine published 12 times yearly. A subscription cannot be purchased sparately without membership. The following rates apply.

	U.S. Domestic	Canada & Mexico	International
Initial	$18	$20*	$28*
Renewal	$15	$17*	$22*

*U.S. Funds.

Membership in England, France, Germany, Belgium, Holland, Sweden and Switzerland is acquired through the local distributor at the prevailing rate.

Limited back issues are availble at $2.50 plus 10% handling and shipping. 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 Users' Group
Hilltop Road
St. Joseph, MI 49085

616-982-3463

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

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

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

REMark®

Issue 47 • December 1983

on the stack

ON THE COVER: Christmas Wonderland developed using AutoCAD. See page 16 for an in-depth review.

Bob Ellerton
HUG Manager

HUG ID Card — Gold At The End Of The Rainbow

Have you ever wondered what to do with that ID card issued with your HUG membership? If you're like me, the ID serves to clutter a space in an already overcrowded wallet. Well folks, it's time that all of us pay a little more attention to the card that is now worth between $500 to $1000 dollars per year.

Beginning December 1, 1983, Heath Company and Veritechnology Electronics Corporation (Heathkit Electronics Centers) will be extending to members of the Heath/Zenith Users' Group **a 10% discount on all computer related kit items and a 20% discount on all computer related wired products**. That's right! Your HUG ID entitles you to discounts on a variety of Heath/Zenith computer products purchased either through your local Heathkit Electronics Center or through the Heathkit Mail Order Catalog. The following paragraphs will outline this new benefit offered exclusively to members of the Heath/Zenith Users' Group.

Limitations

The following limitations apply to the discount program:

1. You must be a member of the Heath/Zenith Users' Group for a period of 90 days before you can use your HUG ID to participate in the discount program. If you renew your membership in the Heath/Zenith Users' Group, you may use your ID card immediately to participate.

2. Your ID card can only be used at your local Heathkit Electronics Center or through mail order purchases placed with Heath Company. Telephone orders **will not** be accepted.

3. Your discount applies to the first $5,000 of products purchased per year. The $5,000 ceiling on purchases is based on the current list price of the products you purchase at the date shown on the invoice.

4. You may only purchase one of any given product per year. (e.g. one "Z" or "H" 100 series computer per year).

5. Your HUG ID card is not transferable. You will be required to show both your HUG ID and another suitable piece of identification to receive the benefits of the discount program.

6. You cannot make a purchase at the discounted rates without your HUG ID card.

7. Your HUG ID cannot be used to purchase individual parts or HUG software products under the discount program from the Centers or from Heath Company. HUG products are considered parts.

Discount Rules and Regulations

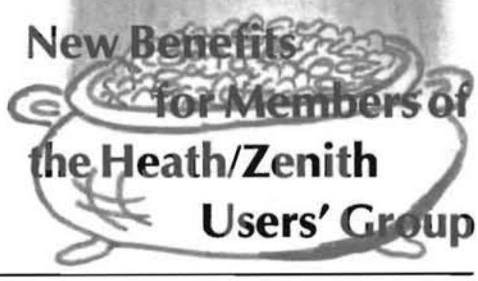
1. The ID card can be used for a 10% discount on computer related kit items only.

2. The ID card can be used for a 20% discount on assembled computer related items only.

3. Any item purchased which installs as a modification to the internal portions of the computer (i.e. boards, drives, ROMs, etc.) is considered a kit item.

4. Any item, unless purchased as a kit, which is an external addition to the computer as a system (e.g. complete wired drive enclosures, printers, modems, etc.) is considered a wired item.

5. Software (with the exception of HUG software) is considered a finished or assembled item and therefore, can be purchased at the 20% discount.



6. Unless stated specifically in "sales specials" that other discounts do not apply, the HUG ID can be used in addition to discounts offered through local Heathkit Electronic Centers or through the Heathkit Mail Order Catalog. However, the HUG ID may not be used with any promotional certificates offered by Heath Company or the Heathkit Electronic Centers.

7. Heathkit Mail Order Catalog purchases must be accompanied by your HUG ID card and self-addressed stamped envelope for return of your ID once the order has been processed.

8. If you lose your ID card, you can only receive a replacement by purchasing a new membership to HUG at the current renewal rate. You may however, use this ID immediately if you have been a HUG member for more than 90 days.

9. To receive a replacement ID, send your request along with your **HUG ID number (found on your REMark mailing label) and renewal fee to:**

Attention: HUG Secretary / New ID Card
Heath Users' Group
Hilltop Road
St. Joseph, MI 49085

New ID cards will be issued on renewal with a completely new look. However, your current ID will be acceptable as we take advantage of this new program. Our new ID card will be plastic and have a little more endurance than our current cards. HUG should have the new card available around January 1, 1984. Remember, you will receive the new ID card at your normal renewal date which can be determined by examining the last six digits of your current ID or on the mailing label found on your copy of REMark. If you have been a standing member of the Heath/Zenith Users' Group, your last six digits might look like this:

780112

This number breaks down as follows:

78 — became a member in 1978
01 — joined in January
12 — joined on the 12th

Therefore, if you have retained your membership over the years, you would renew your membership on or before January 12, 1984. You would then receive the new ID card shortly after renewing your membership for 1984.

HUG announced the existence of the discount program October 22, 1983, while attending the very successful Capitol Heath Users' Group Conference. Some of the questions asked at the Conference will be addressed in the following paragraphs.

Q. If I have renewed my membership for more than one year, how will I receive the new ID card?

A. We are now taking steps to provide a sort routine for our membership computer that will pick out those individuals that have memberships exceeding one year. After we have determined which members will require more than one ID, new cards will be provided for each of the years that the member has signed for. It will be the member's responsibility to keep track of and use the proper ID. You will receive a new card per month for the number of years you have chosen to renew (e.g. five years — you receive a new card each month for five months).

Q. Can I use my current ID card until I receive my new ID card?

A. Yes, provisions are being made at this time to honor your existing ID card through both the Centers and through Mail Order.

Q. Is software considered a kit or wired product?

A. Software is considered wired. Therefore you are eligible for the 20% discount on software products.

Q. Is the Winchester modification for the Z-100 computers considered a kit or wired product?

A. According to the rules, any product which is installed inside the computer is considered a modification or kit item. Therefore, you would receive a 10% discount on that item.

Q. I'm a life member of HUG (Larry Sites), how you gonna handle me?

A. We're working on that one, Larry! (Larry was awarded a life membership to HUG for outstanding support of both HUG and the Heathkit name. He built the first electronic kit offered by Heath in 1949.)

Q. How many of each item can I purchase using the HUG Discount Program?

A. You may only purchase one item from each model series available through the Heathkit Mail Order Catalog or at the Heathkit Electronics Centers. In other words, you may only purchase one computer, one printer, one modem, etc. Remember, HUG Software and replacement parts are not included in this program. If we abuse the program we may lose it!

Q. What is the limit on the program in dollars?

A. You may purchase up to $5,000 worth of equipment at the list price using the HUG Discount Program within a period covered by your one year membership to HUG.

Q. Can I purchase a computer or related item for someone that is not a member of the Heath/Zenith Users' Group?

A. No, the discount program is designed for HUG members only. The rights and privileges provided to you as a member of HUG cannot be transferred to another individual. Also, if you make a purchase for another person, the warranty will be in your name only. You will be required to show your ID card as well as another piece of identification to receive the benefits of the discount program.

I'm sure that additional questions and stipulations may occur over the next couple of weeks. If you have additional questions please feel free to contact the Heath/Zenith Users' Group. We will cover this information again in the January Issue of REMark as an update to HUG Benefits and Policies.

THE H-100 SERIES: THE WORLD'S FIRST 16-BIT COMPUTER KITS.

with authorized sales and service *only* through the Heath Company and Heathkit® Electronic Centers.*

The world-famous H-100 Series Heath/Zenith computers are the first to give you advanced 16-bit computing at a kit price. Many who have never considered kitbuilding are now interested because most circuit boards are prewired, making the H-100 Series our simplest computer kits. Dual microprocessors deliver 16-bit speed and 8-bit compatibility. The industry standard S-100 card slots allow a host of peripherals and memory expansion to 768K RAM. And our stores and catalogs have the software to help you take full advantage of the faster 16-bit operating speed.

When you're ready to move up to H-100 Series, remember: these computers are available *only* through the companies that give you solid service and a trustworthy warranty...Heath and Heathkit® Electronic Centers.* The companies that stand by their promise of support to kitbuilders. The companies with the pledge: *"We won't let you fail."*

Many companies would like to sell our product but no other dealer or vendor can resell the H-100 Series without voiding the original factory warranty. When buying the world's first 16-bit computer kit, buy only from the world leader in electronic kits.

Authorized sales and service available only through the Heath Company and your...

Heathkit®
ELECTRONIC CENTER*

*Heathkit Electronic Centers are units of Veritechnology Electronics Corporation.

VEC-770

BUGGIN' HUG

Corrections To Article In Issue #43, "Processing H/Z-19 Special Functions Using Heath ASseMbly Language"

Dear HUG,

I've received a couple of inquiries about my article in REMark, Issue 43, on use of special function keys of the H19. These users reported problems in assembling the DEMO.ASM program.

In going back to the printed source program, I note that lines 5 through 8 were omitted in REMark. Those lines contained EQUates for HDOS system calls which permitted the ASseMbly of the program without use of XTEXT's and '.ACM' files. I must confess, after adding comments to source program for publication, I didn't bother to ASseMble and test it again. I have discovered, as I suspect you must have too, that the resulting source generated errors when being assembled. Here are the corrections, with my sincere apologies:

The user can either add:

```
XTEXT   HOSDEF
```

.....to the published source program (and have HOSDEF.ACM on disk when assembling).....or fill in the corrected, missing lines 5 through 8:

```
.CONSL    EQU  00006A   5    H D O S
.CTLC     EQU  00041A   6  S Y S T E M
.EXIT     EQU  00000A   7    C A L L S
.SCIN     EQU  00001A   8 (S C A L L S)
```

Very sorry about any confusion caused!

William R. Rousseau, M.D.
1388 Highland Ave.
Salem, OH 44460

New HUG Clubs Forming

I am looking for other Heath/Zenith users in the Shreveport and Bossier City, Louisiana or the ARK-LA-TEX area. Anyone interested in starting a Users' club, contact me at 456-2473, 742-4098, or write to the address below.

Dr. Mick Hedrick
100 Barksdale Blvd. W.
Barksdale AFB, LA 71110

Attention Springfield, Illinois and surrounding area Heath/Zenith computer users. A local HUG group is forming and all interested parties are invited to contact: Bobby Wright (753- 5795 w) or Bill Bennum (544-4108 w) for information regarding the group.

John F. Sams
2313 Grandview
Springfield, IL 62702
(217) 544-3976

We in the Savannah area are going to form a local HUG. Contact people for the group will be:

William Gartelmann (912) 925-3765
John Gasper (912) 354-8504

William Gartelmann, Jr.
832 Dancy Avenue
Savannah, GA 31406

A local retailer has offered his store as a meeting place if there is interest in forming a Las Vegas Heath Users' Group. Interested people may contact me at the following address.

Anthony Marshall
4500 Draga Place
Las Vegas, NV 89115
(702) 644-5388

I am looking to form a Heath/Zenith Users' Group in New Zealand. Anyone interested, contact me at the following address.

R. Siebers
94 Dowse Dr Maungark
Lower Hutt
New Zealand

I am looking for interested "Flint HUGgers" to contact me in hopes of starting a new HUG club. My phone number is (313) 235-4016.

Ray Poli
2322 Calumet
Flint, MI 48503

I have an H89 with 48K, HDOS, and an H25, and am interested in having users' in my area contact me who have similar equipment to discuss problems, programs, etc. My phone number is (219) 747- 0108, please call after 8:00 p.m. Or write to the address below.

Ken Fuelling
7506 Ideal Avenue
Ft. Wayne, IN 46809

I am interested in having some contact with other HUG members in the Syracuse area. My phone number is (315) 437-6821 and would enjoy talking with other H89 users'.

Richard E. Williams
222 East Milford Drive
Syracuse, NY 13206

We are looking for other Heath/Zenith users' in the Lehigh Valley, PA area who would like to start a Heath Users' Group. Anyone interested can call:
Carol Block 215-770-4640 (w)
Jim Batug 215-770-5993 (w)

Or write to:
Jim Batug
1425 N. Broad St.
Allentown, PA 18104

Jim Batug
Carol Bloch

I am interested in finding other Heath/Zenith users in my area, which is any counties surrounding and including Marshall County, who might be interested in forming a users' group. My phone number is (304) 845-4544. Or write to me at the address below.

Chuck Horvath
RD 2, box 266
Moundsville, WV 26041

HELP!!

Dear HUG,

I purchased the PERCOM ZFD Double density controller board for my H89 in early 1982. Since that time, they have stopped all support of the product. Having recently purchased CP/M, I called them about obtaining their ZBIOS. They weren't even kind enough to tell me if they had issued the BIOS, as was their intent, judging by their earlier advertisements.

Would someone know if they issued the BIOS? If so, do you suppose they could object if I were to ask that someone supply it to me at a reasonable price?

Donald E. Risher
908 Charlotte Place
Charleston, WV 25314

Corrections To PACMAN In August Issue

Dear HUG,

The PACMAN for the H/Z-89 listing on page 8 of the August 1983 Issue of REMark had three errors. Line numbers 230, 270, and 510 should read as follows:

```
00230 FOR I=1 TO 31:  GOSUB 310:  C=C-1:  NEXT I

00270 FOR I=1 TO 31:  GOSUB 310:  C=C-1:  NEXT I

00510 D$=Y$+CHR$(R+3)+CHR$(C)+Q$+"       "
```

With these corrections, the program works just fine.

Joe P. Douglas
5014 Romford Dr.
Corpus Christi, TX 78413

☞

Correction to Hero Article In Issue 45

Dear HUG,

I was pleased to see my Hero article in the October (#45) issue of REMark. Unfortunately, lines 4150-4580 of the program did not make the published listing. I am sorry for any inconvenience this may have caused anyone.

Mike Curiale
23 Sullivan St.
Charlestown, MA 02129

```
                1010 *
                1020 *        CP3
                1030 *
                1040 *   by Mike Curiale - 1983
                1050 *
                1060 *------------------------------
                1070 *

                4150 later
02B0- 0C 06 09
02B3- 29 0E 3C
02B6- 29 03      4160        .hs    0c0609290e3c2903
02B8- 18 06 29
02BB- 2A 3A 3F 4170          .hs    1806292a3a3f
02BE- FF         4180        .hs    ff
                4190 *
                4200 *------------------------------
                4210 *
                4220 *  something in the way
                4230 *
                4240 comment2
02BF- 7D 0E FC 4250          tst    random
02C2- 2B 01      4260        bmi    .1
02C4- 39         4270        rts
02C5-            4280 .1     >spw   in.the.way
02C5- 72 02 C9 0000>          .DA    #$72,IN.THE.WAY
02C8- 39         4290        rts
                4300 in.the.way
02C9- 38 00 05
02CC- 00 2B 03 4310          .hs    380005002b03
02CF- 1F 3C 29
02D2- 0C 12 03 4320          .hs    1f3c290c1203
02D5- 2A 37 37
02D8- 03         4330        .hs    2a373703
02D9- 0E 3C 29
02DC- 03         4340        .hs    0e3c2903
02DD- 1F 32 23
02E0- 0C 38 09
02E3- 09 14 03 4350          .hs    1f32230c3809091403
02E6- 0B 09 0D
02E9- 03         4360        .hs    0b090d03
02EA- 0C 15 00
02ED- 3C 03      4370        .hs    0c15003c03
02EF- 2D 06 06
02F2- 09 25 25
02F5- 3F         4380        .hs    2d06060925253f
02F6- FF         4390        .hs    ff
                4400 *
                4410 *------------------------------
                4420 *
                4430 *  clear the way
                4440 *
                4450 comment3
02F7- 7D 0E FC 4460          tst    random
02FA- 2B 01      4470        bmi    .1
02FC- 39         4480        rts
02FD-            4490 .1     >spw   clear.the.way
02FD- 72 03 01 0000>          .DA    #$72,CLEAR.THE.WAY
0300- 39         4500        rts
                4510 *
                4520 clear.the.way
0301- 19 18 21
0304- 09 2B 03 4530          .hs    191821092b03
0307- 38 32 23
030A- 03         4540        .hs    38322303
030B- 2D 06 09
030E- 29 3F      4550        .hs    2d0609293f
0310- FF         4560        .hs    ff
                4570 *
                4580 *------------------------------

0000 ERRORS IN ASSEMBLY
```

Response to Comments About Maple

Dear HUG,

I wish to respond to Mr. Gilbertson's comments about MAPLE in a letter to HUG which appeared in the October, 1983 Issue of REMark (#45).

I am happy he likes the program. However, he listed two disappointments. The first concerned the terminal setting on exit, which Mr. Gilbertson would like to be 'normal' mode. Normal here is ambiguous, since many users set their terminal to their own liking (either off-line or by programs) after the system has been booted. Forcing a boot-up terminal setting on exit is just as objectionable to some as not doing so is to others. Unfortunately, the current setting of the terminal on entry to a program is not stored in a CP/M BIOS address, so that the program has no way of finding the user's favorite or current setting for the terminal. In any case, the documentation to MAPLE which accompanies the program gives the patch address for setting the terminal to any valid condition the user wishes on exit, including the boot-up default setting (i.e. ESC z code sent to terminal).

The second wish expressed by Mr. Gilbertson is to be able to send files stripped of linefeed/carriage returns. Here the options offered by MAPLE cover every conceivable possibility.

1. Line Mode: Transmission line by line, with indefinite wait state between lines until the remote system prompts for the next line, and with user setting of prompt character.

2. Block Modes: Transfer of file with CR/LF converted to CR. Transfer of file with CR/LF converted to LF. Transfer of file with CR/LF unconverted.

3. Absolute Mode: Transfer of file with all eight bits of each byte sent to the remote system (WordStar uses the eighth bit to control virtual CR/LF's).

I hope these clarifications will be of use to Mr Gilbertson and to your readers.

Dr. William C. Parke
(Author of MAPLE)
1820 S. St. NW
Washington, D.C. 20009

Response To 'Info Needed On UCSD Pascal'

Dear HUG,

This letter is in response to the 'Info Needed On UCSD Pascal' in the August (#43) Issue of REMark.

There is an easier way to send output to the PRINTER: (unit #6) rather than using UNITWRITE. I don't believe it is exactly what Bjorn Petersen is looking for.

First, define a variable of type TEXT:

```
VAR   TOPRINTER : TEXT;     (* set up output device *)
```

Next, using the REWRITE procedure, open up the PRINTER: (unit

#6) for write as file TOPRINTER:

```
REWRITE(TOPRINTER,'PRINTER:');  (* open output file *)
```

Then using the WRITE and WRITEIN procedures, send output to the output file as follows:

```
WRITE (STRING);             (* default; sends output string to CRT *)
WRITE (TOPRINTER,STRING);   (* send output string to printer *)
```

Several files can be opened at the same time by using the REWRITE procedure. The original file can be closed and then reopened to another output device, either a logical device or file, by the following commands:

```
CLOSE(TOPRINTER);
REWRITE(TOPRINTER,'CONSOLE:');
```

The above closes the output to the PRINTER: and then reopens the file for output to the CONSOLE:. So, both of the following lines will output STRING to the CONSOLE:.

```
WRITE (STRING);
WRITE (TOPRINTER,STRING);
```

There is no easy way to toggle between the CONSOLE: and the PRINTER:, but both files can be open at the same time for writing. The CONSOLE: is already open by default, but any other files must be opened either using the REWRITE (logical devices or new files) or RESET (existing files). The above examples are the preferred way to write to files as this is a standard way to open files in Pascal. Refer to any Pascal reference manual on the variations of the WRITE and WRITELN procedures in connection with files.

The following is a short program to show how to open the PRINTER: for write as a file in UCSD Pascal.

```
PROGRAM OPEN_FILE;

VAR   TO_PRINTER : TEXT;       (* set up identifier for output file *)

BEGIN
  REWRITE (TO_PRINTER,'PRINTER:');    (* open printer for output *)
  WRITELN ('This line of text is being printed on the CONSOLE:');
  WRITELN (TO_PRINTER,'This line of text is being printed
                                                    on the PRINTER:');
  CLOSE (TO_PRINTER);  (* close output file *)
END.
```

Christopher S. Simmons
2802 N. Greenwood St.
La Grande, OR 97850

A Bug In RFCAD Pkg.

Dear HUG,

File: BFILTER.BAS in RFCAD package (885-8020)

The line with the bug is as follows:

```
8090 IF W > F0 * .4 THEN GOTO 8070
```

I think that the line should be:

```
8090 IF (W * 1E+06) > F0 * .4 THEN GOTO 8070
```

The purpose of this line is to insure that the bandwidth for the filter (variable W) is less than 40% of the center frequency (variable F0). F0 is entered in MHz and converted to Hz after it is entered (W is also entered in MHz). Other calculations involving W perform the conversion (W to Hz) in the equations.

D.W. Gohr
Heath Co.

A Master Cataloging System

Dear HUG,

As many CP/M buffs know, CAT.COM and it's associated files FMAP.COM, UCAT.COM, and MAST.CAT (plus CAT2.COM) form a master cataloging system that allows the user to catalog all the files on all of his disks into a single index, and you know how good that is when you are looking for a particular file.

CAT.COM does a great job of an alphabetized index complete with spaces to allow someone using a continuous roll of paper in his printer to cut each page to proper length. The only drawback to this fine program is the fact that only one disk is listed for each file, which results in the same file being listed many times in the case of utility files, etc. On the other hand, CAT2.COM eliminates this problem by printing each filename only once with the disk names strung out after them. However, CAT2.COM has no provision for paging and the index comes out in one continuous sheet. If you have had this problem and are tired of slicing lines right in the middle, read on.

You can obtain a copy of the CAT2.ASM source program by using two utility programs to be found on the HUG disk #885-1212. They are UNLOAD.COM and DISASM.COM. You use the UNLOAD.COM program to obtain CAT2.HEX and from this you disassemble the program with DISASM.COM, which produces CAT2.ASM. This is the program referred to below and to which you will modify to obtain a program in which you will be able to vary the number of typed lines per page, and vary the number of blank lines at the bottom of the page for a perfect page each time.

When you disassemble a program as described above, the labels are shown as Lnnnn: where nnnn represents the line number.

Only the lines to be changed will be listed below plus any new lines to be added to the source listing.

In the routine at label L0116:, modify the user's stack from:

```
LXI     SP,003B2H
```

to:

```
LXI     SP,043B2H
```

In the routine at L0135 change from:

```
L0135:      (was blank)

            LXI     H,L032F
```

to:

```
L0135:      CALL    L03BC

L0136:      LXI     H,L032F
```

In the routine at L027E: (between STA L034D and RET):

```
INSERT---------->          CALL     L03BC
```

Since there are two different lengths of lines, we have to obtain counter information from two places and sum them in the counter subroutine.

At the end of the source program, add the following:

```
L03AE:
           NOP

L03BC:     LDA    L03AE           ;Typed line counter subroutine
           INR    A               ;begins here.
           STA    L03AE
           CPI    03AH            ;Change number of typed lines/page
           JZ     FFEED           ;Jump if zero
           RET
```

Vectored to 46 ☞

FREE CP/M-Plus™ from MAGNOLIA MICROSYSTEMS

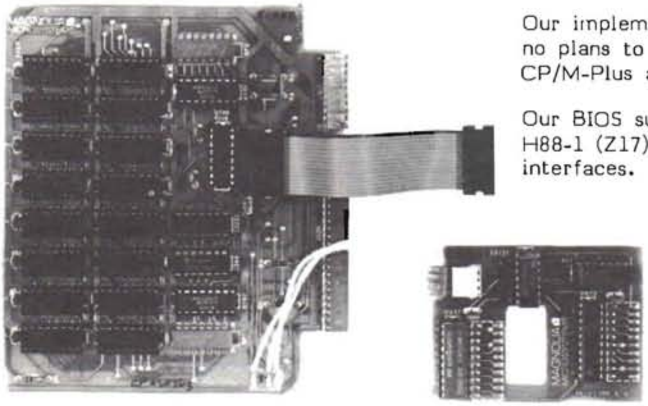
With purchase of 128K RAM Board at $595

Digital Research's list price for CP/M-Plus is $350, but it will be included AT NO EXTRA CHARGE with purchase of our 77318 128K RAM board at its regular price of $595 by mentioning this ad when ordering.

CP/M-Plus^{tm} support is available for Z89 and Z90 computers with our 128K RAM board and either the Z89-37 (Z90) or our own 77316 Double Density disk controller. Over 700K of files are included, so it will not be distributed on Hard Sectored media.

Using banked RAM, disk performance is enhanced through Hashed Directory tables, Directory Buffers, and LRU Data Buffers.

New utilities and features include a 'Help' command; optional Password protection and Time and Date Stamping of files; Console Redirection to or from disk files; and many others.



Our implementation of CP/M-Plus REQUIRES the use of our 128K RAM board. We have no plans to implement an un-banked memory version, most of the advantages of CP/M-Plus are neither available (nor practical) in an un-banked version.

Our BIOS supports both our 77316 and Zenith's Z89-37 Double Density controllers, Heath's H88-1 (Z17) Single Density controller, and our 77314 CORVUS and 77320 SASI-bus interfaces.

We are including the SOURCE code for our BIOS, together with Digital Research's MAC, RMAC, LINK, and SID software development tools, you may customize it to your specific application.

Orders must specify the Double Density controller and boot drive: the Z89-37 (as used in the Z90) or our 77316 (5-inch or 8-inch).

Earlier purchasers of the 128K RAM board, who are registered owners of Magnolia Microsystems CP/M 2 (S/N 2-175-xxxx) can update to CP/M-Plus for $100 (plus shipping and handling).

DISK INPUT/OUTPUT (I/O) BOARDS

These boards are made available apart from our subsystems for use by 'Systems Integrators'. End-users should carefully consider their skills and desired level of involvement before undertaking this responsibility; purchasing a complete subsystem from a qualified integrator may be a wise choice.

Although similar in name and apparent function, these boards fall into two distinct classes:

- INTERFACE boards connect a disk subsystem (containing a controller) to the computer. These boards include the Z89-47 and Z89-67 and our 77314, 77317, and 77320 interfaces.
- CONTROLLER boards actually control a plain disk drive. These boards include the H88-1 (Z17) and Z89-37, and our 77316 Double Density Controller.

All Magnolia I/O boards use proprietary techniques to expand the 89's I/O addresses, allowing use of more than Zenith's 5 (3 serial ports, 2 disk controllers). Since our INTERFACE boards also contain 3 RS232 ports (replacing the machine's H88-3) allowing expansion of the I/O capability without using an additional card slot.

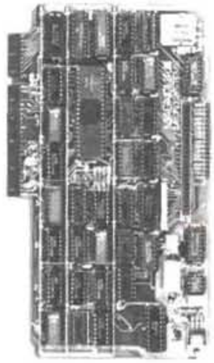
Each Magnolia I/O board package includes:

- The disk I/O board itself
- A new I/O decoder PROM
- A new Monitor EPROM
- A copy of MMS CP/M 2.2

Magnolia's CP/M software supports over 20 soft-sectored media formats, including all existing Zenith formats.

MAGNOLIA MICROSYSTEMS, INC. • 2264 - 15th Ave West
(206) 285-7266 (800) 426-2841 • Seattle, Washington 98119

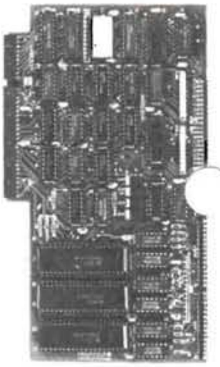
Double Density Disk Controller reduced to **$495**
order number 77316



This board adds complete hardware and software support for four 8" single- or double-sided AND four 5" SS or DS, 48- or 96-track-per-inch (40- or 80- track) drives, at both single and double recording densities.

This controller is compatible with 8" drives by Shugart, Qume, and Tandon; 5" drives by Siemens, Tandon, and MPI; and most other 'industry standard' drives (possibly depending on the skill level of the experienced 'Systems Integrator').

SASI-bus Winchester Interface **$350**
order number 77320



This board (functionally replacing the Z89-67 interface) contains 3 RS232 ports and can support up to 8 SASI-bus disk controllers, with up to 4 drives on each!

It is compatible with Winchester controlers by Xebec, DTC, Shugart, and others, as well as Zenith's Z67. CP/M software support for popular controller/drive combinations is included.

CP/M is a registered trademark, and CP/M-Plus, MAC, RMAC, and SID are trademarks of Digital Research

The CHUG Conference — Well Done!!

Terry L. Jensen
Software Developer

It was a pleasure to attend on October 22nd, the Second Capitol Heath Users' Group (CHUG) Conference held at the Tyson's Westpart Hotel near Washington D.C. What a relaxing and enjoyable time it was!! The CHUG members did an outstanding job of preparing and putting on an excellent conference. (Believe me, I know how much work goes into a conference.)

Due to the fact that I was not involved directly with the conference (other than attending as a guest speaker), it is impossible for me to give a complete picture of all the preparation that went into the conference, as well as to give complete recognition to all the workers that gave their time and talents to contribute to such an enjoyable and successful conference. I hope you enjoy the sample of photos that I took while at CHUGCON.

The 1983 CHUG Steering Committee was headed by Mike Supley, the new CHUG President. Mike, as I found, used his talent as an administrater to organize and delegate the many areas of the Conference to the appropriate individuals. The help offered by the CHUG members contributed to a well executed conference.

Two CHUG members, Gary Jewell and Mike Bailiff, were placed in charge of the CHUG Booth. This was the central location for the entire conference. Everyone that helped out at the CHUG booth deserves a big round of applause for their smooth handling of the questions and information relating to the conference. (For your information, Gary Jewell was given, as a keepsake, the award presented to CHUG from the National Heath Users' Group for being the first local HUG group to hold a regional Conference. Gary headed up last year's CHUG Conference.)

The luncheon and dinner meals, by reservation only, were prepared at the Westpark Hotel for conference attendees. Lee Schumacher did a tremendous job of coordinating the meals as well as selecting an excellent menu.

The CHUG Conference publicity was handled by Bill Cross and Susan Quarles. It was their job to announce the Conference and its events. (Let it be known that HUG will provide free space in REMark for advertising such an event.)

Mike Frieders was in charge of setting up and coordinating the lecture sessions. The agenda and speaker list were very well selected and organized as will be explained shortly. (After all these years of "talking" with Mike on the HUG Bulletin Board, I still did not get a chance to meet him face to face.)

The luncheon and dinner meetings were enjoyable for all.

Bob Ellerton and Mike Cogswell, the former CHUG President, prepare to award Gary Jewell for his efforts in leading the First CHUG Conference.

John Stetson gives his lecture on Z-100 software.

The CHUG "Swap Meet" was handled this year by Dr. Bill Parke. The "Swap Meet" was where any used computer equipment could be bought or sold. (It was great to get to meet Dr. Parke, the author of the MAPLE modem package.)

Bill Mulroney was responsible for contacting guest speakers for this year's CHUG Conference. It was indeed a pleasure to share the podium with such distinguished guests as Dr. Ruth Davis and Mr. Bob Ellerton. Dr. Davis is President and Founder of The Pymatuning Group, Inc. which specializes in science and technology management and development strategies. Bob Ellerton is the Manager of the Heath/Zenith Users' Group (HUG). (Bill Mulroney should be recognized as a distinguished individual in his own right. I am sure I can say this for Dr. Davis and Mr. Ellerton; Bill made us feel very comfortable, an excellent host!)

The Luncheon and Dinner meetings were exciting for me because I was invited to the Conference as a speaker. It was an honor to meet and listen to Dr. Davis at the luncheon. She provided insight into microcomputer trends, while mixing in a touch of humor. At dinner, I charged each person (including myself) and the CHUG Group as a whole to create "lines" of communication between computer users. Bob Ellerton's announcement at dinner concerning the discount on computer products for HUG members was, as expected, very well accepted.

Sam Yates had a busy day as he was in charge of the vendors that attended the CHUG Conference. Sam rapped up the luncheon and dinner sessions by announcing the winners of the many prizes which were offered by the vendors. (It was neat that after all the work by Sam, he won the top prize of the day.)

The lectures for the day were coordinated by Mike Frieders, as mentioned above. The CHUG organization has many knowledgeable members from which to select speakers. Look at this list of speakers, most of whom are directly associated with CHUG, and their topics:

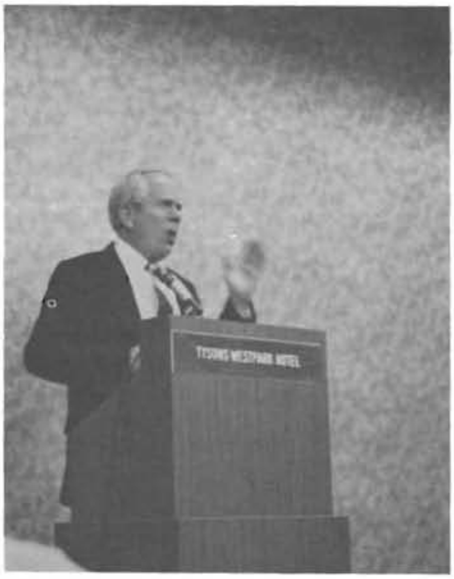
Bob Shon	Computer Video Generation
Carl Fosler	Low Cost DBMS Software
Ray & Janet Cherry	Introduction to "C"
Don Lewis	Public Domain Software
Arthur Goldsmith	dBASE II Beginner
Robert Todd	Public Domain Software for Experienced users
Bill Johnson	Genealogy and Computers
John Stetson	Z-100 Software
Paul Beck	dBASE II Advanced
Mike Cogswell	Z-100 Hardware
Leon Wittwer	Improving the H89

In addition to the fine lectures, thirty vendors were promoting and selling their wares. The vendor area was booming with spectators and customers. There were many conference sale specials within the vendor area throughout the day. I could tell by the smiles on the faces of the vendors that the conference was enjoyable for them also.

This short article does not do justice to such a full day at the CHUG conference. But then how do you explain something that is informative and fun at the same time ... well, I guess, like the National HUG Conference, you have to be there.

My congratulations to the entire Capital Heath Users' Group Organization, including the many spouses and family members that contributed to producing a spectacular show.

Mike Supley, CHUG President, makes the formal introductions.



Bill Mulroney, the CHUGCON Host, introduces the Guest Speakers.

Dr. Ruth Davis speaks at the luncheon.

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Disk Drives

Starting at **$169.00**

Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor

40 or 80 Tracks — Single or Dual Head — New 3½″ Drivette™
Our Disk Drives Are Capable Of Single **And** Dual Density Operation

The Newest Technology Major Brands Capable Of Operating *On:*
RADIO SHACK[1] — HEATH/ZENITH[2] — APPLE[3]/FRANKLIN[4]
IBM/PC[5]-TEXAS INSTRUMENTS[6] & MOST OTHER COMPUTERS

Drive a Hard Bargain!!™ — 5 M.B.-12 M.B. Hard Drive, Everything You Need
Complete Systems . . . . . . from $999.95

Diskette Breakthrough — 10 Pack in Library Case — $~~18.95~~

NEW LOW PRICE

Since We Are Always Finding Ways To Save You Money,
Please Call For Our Most Current Pricing.

TOLL FREE ORDERING
1-800-343-8841

GENERAL AND TECHNICAL
1-617-872-9090

Model I/III/IV Drives (0 1 2 3)	starting at **$169.00**
Color Computer Drive (0 1 2 3)	**$ Call Toll Free**
Apple/Franklin Compatible Add-On Drives with Case & Cables	$219.95
Apple/Franklin Compatible Drives Complete with Controller	**$259.95**
Model I/III/IV Memory Upgrade	**$ Call Toll Free**
Printers — Daisywheel/Dot Matrix	**$ Call Toll Free**
Diskettes in Library Cases	$~~18.95~~ **New Low Price**
Cases and Power Supplies — (Single-Dual-1/2 Height)	starting at $144.95
Printer **Buffers** 8K to 512K	starting at $143.95
Percom Double Density Controller (Model I)	**$ Call Toll Free**
Holmes Model I/III Speed-up Mod-VID/80	starting at $90.00
Gold Fingered **Edge Card Extenders**	starting at $13.00
Cables — Printer/Disk Drive	starting at $23.00
DOSPLUS	**$ Special Prices**
Repair Services Now Offered — Fast Turn-a-Round	**$ Call Toll Free**

Now 40 Track

Warranty on Disk Drives — 6 Months — Extended Warranty $ Call Toll Free

SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090
Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.

Canada
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular **Tel. (514) 845-1534**
Canadian Toll Free 800-361-5155

Dealer inquiries invited.

[1] TM TANDY CORPORATION
[2] TM ZENITH DATA SYSTEMS
[3] TM APPLE COMPUTER CORP.
[4] TM FRANKLIN COMPUTER, INC.
[5] TM IBM CORPORATION
[6] TM TEXAS INSTRUMENTS
© Copyright 1983

William M. Adney
4821 Sunnybrook
Buena Park, CA 90621

Trade Your H/Z-89 For An H/Z-100?

If you currently own an H/Z-89, you probably have been wondering about trading for an H/Z-100 or upgrading your current system with some of the hardware and software available from D-G Electronic Developments Co., Magnolia Microsystems, and Technical Micro Systems Inc. (TMSI). What does it cost for the upgrade as opposed to trading for an H/Z-100? What do you gain (or lose) by upgrading the H/Z-89? What is the best value for the money?

Heath Trade-In Policy

One of the first things that I learned was that the Heath policy allows up to 50% of the list price in the current catalog for a trade-in. The key words are "up to". What that really means is that the maximum trade-in is 50%, but the actual trade-in could be less. Since the policy gets a little confusing, I'll illustrate with an example. Let's say that 50% of the list price on your current equipment is $1000. You want to buy some equipment that is listed at $2500. Since your trade-in equipment is in excellent shape, you get the full 50% for it, and your net cost is $1500.

On the other hand, if you want to buy some equipment that is listed at $1500, the maximum trade-in that you can possibly get is $750 or 50% of the list price of the new equipment. The bottom line is that you will always pay at least 50% of the list price for any new equipment on a trade-in. I guess that the best word that describes the policy is "incentive" since it clearly is to your benefit to get the best trade-in possible. Note that the equipment must also be fully operational with no physical defects, like scratches, to get the full trade-in. As you might expect, the trade-in policy only applies to Heath retail stores, and it is not available through mail order.

Our Current System

Let's assume that our current system consists of an H-89 with green screen ($1429), the additional 16K memory ($115), the soft sectored disk controller ($299), and 2 soft-sector disk drives ($1295). Since the equipment is in good shape, we'll assume that we can get the maximum trade-in which is 50% of $3138.00 or $1569.00.

Basis For Comparison

Since the H/Z-100 has both 8-bit and 16-bit processors and now comes with 192K of memory, we would like to find an upgrade to the H/Z-89 that has those capabilities. Since the some of the upgrades will also require new operating system software, that cost will have to be considered too.

The Super 89 (D-G Electronics)

The D-G Super 89 has been advertised for quite a while and offers a lot of enhancement to the current H/Z-89 capabilities. In addition to running at 4MHz (it's also jumper selectable to 2MHz), the board can be upgraded to a total of 256K. D-G has a lot of nice add-ons, like the Electronic Disk, but the upgrade board provides only a Z-80A processor which still limits you to 8-bit software. Based on the April 15, 1983, price list; the Super 89 with 192K of memory is $909.00. D-G Electronics also offers a 256K version for $989.00.

Magnolia Microsystems

Magnolia Microsystems is also a well known name in the Heath/Zenith world. Their answer to the "4MHz 89 Question" is the MMS-Net Network Controller Board which is offered in a 64K version ($695) and a 256K version ($1295). Assuming that we want to try to come as close as possible to the H/Z-100 baseline, we'll choose the 256K version for comparison. According to the MMS representative at CP/M-83, we'll also need the MMS version of CP/M Plus ($200) to get the full bank select capability of the 256K board. The total cost is $1495.00 which is shown in their price list as of January 20, 1983.

The H-1000 (TMSI)

Technical Micro Systems Inc. has introduced an upgrade which provides a number of interesting capabilities. It features both 8-bit (Z-80) and 16-bit (8086) processors. The Z-80 is software selectable for 2 or 4MHz, and 128K of memory is standard. The 8086 runs at 8MHz. TMSI also includes your choice of either MSDOS or CP/M-86 in the purchase price of $1495.00. It is advertised to be fully compatible with existing Heath/Zenith software, so you shouldn't have to purchase a second operating system for the Z-80. According to the technical information, the board contains sockets for 256K (available for $250 additional) and a memory expansion bus for up to 1MB of RAM. The above prices were effective on February 8, 1983, and multiple unit discounts are available.

Now For The H-100

Since we want to get the most for our trade-in, let's look at the cost of the H-100 with some goodies that will help us achieve that goal. Note that some of the prices will include system discounts, and all prices were taken from the current Heathkit Retail Computer Catalog (801R).

The H-100 (low profile) is listed at $2199 and a second disk drive is $300 when purchased with the kit. The monochrome (green) monitor is $139.95. Our software choices include the HSS-100-1 Software Starter Package with both operating systems ($300), and the PeachText 5000 Software Starter Package at $275 (because we're now H-100 owners). You're probably wondering why I added all of that software...but wait, the total list price of all of that hardware and software comes to $3233.95. Since we assumed that our H-89 trade-in was worth $1569.00, the net cost is $1664.95 excluding sales tax and shipping if required.

The Bottom Line

Trying to compare the cost of these upgrades versus the cost of a new H-100 system has been somewhat like trying to compare apples and oranges, but here is a summary of what we have seen:

D-G Electronics Upgrade	$909.00
Magnolia Microsystems Upgrade	$1495.00
TMSI H-1000 Upgrade	$1495.00
New H-100 System	$1664.95

Since the cost of the H-100 system was nearly the same as the most similar upgrade (TMSI with the dual processors), I concluded that it was more cost effective to buy the H-100 rather than try to upgrade my H-89. It even makes more sense when you consider that you also get all of the software goodies along with the H-100 for essentially the same price, but that's comparing apples and oranges again.

There is one other consideration that I haven't mentioned yet. It's important to me that I can get local assistance in case I have any problems with my system. Although I've heard a lot of good things about these vendors, I always came back to the idea that it was better to have local support available, just in case. So far, I have been very satisfied with that decision.

The Z-89 Versus The ZF-100-21

I had some difficulty getting the prices for the assembled versions which seemed to be related to installation charges, but the basic Z-89 ($1999), additional 16K memory ($115), soft sectored disk controller ($299), and the Z-37 drives comes to $4112.00. The equipment is worth $2056.00 on the trade-in assuming the maximum 50% value.

Trying to get the Z-100 configured in the same way as the kit version presented even more problems since there was no assembled version listed in the catalog which was close to the H-100. We'll assume that the costs will be close to those listed in the catalog which includes the ZF-100-21 ($2899), additional drive ($395), memory expansion to 192K ($165), the monochrome (green) monitor ($139.95), the HSS-100-1 Software Starter Package ($300), and PeachText 5000 ($395). The total cost of that system is $4293.95.

Based on the above assumptions, the net cost of the new Z-100 system is $2237.95 which does not include any sales tax, shipping or installation charges. Upgrading the Z-89 would also result in some additional installation and configuration charges, and although I was not able to get a good estimate, I would guess it would be in the $100-$200 range. A comparison of the hardware/software costs shows a net difference of $742.95.

Since the cost differential between the Z-89 upgrade and the Z-100 system is still relatively modest, I would still conclude that it would be cost effective to buy the Z-100 system with all of the goodies and additional capabilities.

Software Compatibility

There may be a hidden cost in the conversion to the H/Z-100 related to software compatibility. Some software that will run on the H/Z-89's Z80 processor will not run on the H/Z-100's 8085 CPU. In general, the key is to determine whether your software was written using Z80 mnemonics or the standard Intel 8080 code. Some software may also be sensitive to the processing speed of the CPU, so the standard 2MHz speed of the H/Z-89 versus the 5MHz speed of the H/Z-100 could make a difference there also. That's not all bad either because the increase in processing speed will save a lot of your time at the CRT. For example, SuperCalc calculations are noticeably faster on the H-100.

Appendix H of the Z-100 User's Manual lists H/Z-89 software that is compatible with the H/Z-100 as follows:

SuperCalc
Microstat
WordStar
Magic Wand
DataStar
Despool
MACRO-80
CBASIC
Small Business Inventory
General Ledger II
Mail Merge
SuperSort
Word Master
SID/ZSID
Macro Assembler
Pearl III

The software in the current Heathkit catalogs is keyed with a color dot to identify compatibility with the H/Z-100, but I particularly noticed that the following H/Z programs will not work on the new system:

Some business software (e.g. General Ledger, etc.)
Condor DBMS
Magic Spell Proofreader
Microsoft BASIC Interpreter and Compiler
COBOL Compiler
FORTRAN Compiler

Be sure to check the current Heathkit catalog (or ask your local dealer) for the compatibility of each of your programs since I have not listed all of them.

And Finally...

Although you may think that I'm trying to sell the new H/Z-100's, I really believe that it is one of the finest micros on the market. I admit that I'm definitely biased, but I believe that there are a number of good reasons to migrate to the new system. The cost versus the capability plus the hardware and software support are certainly primary considerations. And the Heath/Zenith family of products seems to always provide an excellent value for the money.

Next Month

One of the nicest capabilities of the H/Z equipment is the software control of the hardware features. A very simple program can be written to clear the screen, use the 25th line, or control the H/Z-25 printer. Since the capability to use assembly language programs is provided with all H/Z operating systems, I'll show a sample format that can be used with all the hardware plus some useful programs for clearing the screen and implementing the 25th line. The sample program can also be used if the printer (e.g. H/Z-25) can be software controlled, and we'll even show Jerry Pournelle how to get rid of his "keyclick" problem that he mentioned in the June, 1983 issue of Byte.

CAD/CAM On A Z-100

Sarah P. Spencer
Future Technology, Inc.
13 New Salem St.
Wakefield, MA 01880

Evaluation of the AutoCAD drafting and design system from Autodesk, Inc., reveals a high performance, professional quality product that is convenient and practical for many engineering and drafting applications, and is within a price range that most small offices and independent consultants can afford. One is tempted to compare it to much larger industrial systems.

For our evaluation of AutoCAD, we used a Z-110-22 low-profile computer with a color monitor and Hitachi "Tiger Tablet" digitizer. The computer was upgraded to 192K memory needed to run the software. Additional memory would hold more of a drawing at once, thus reducing disk use and increasing speed. With a very complex drawing, the extra memory could be necessary for certain operations such as rotation, COPY or MOVE, which require recalculation of positions.

When creating or editing drawings, the basic procedure is to give a command, and then to follow the prompts as to what position or angle data is required for that command. This data may be entered through the keyboard, either with the arrow keys or numerically, or through a pointing device, such as a digitizer (or mouse). The MENU selections may also be made using the same method. First, the MENU cursor is keyed in, then the arrow keys move to the desired command. When position data is called for, a cross-hair appears on the display screen to allow the user to see the movement of the cursor from the first point to the second and aid in the position selection. When using the keyboard for position selection, the fast repeat key may be used to speed the movement of the cursor. For both MENU selection and position data, we have found the digitizer to be far superior to the keyboard entry. It is quicker, easier, and "intuitively obvious" to use. Once one has operated AutoCAD with the digitizer, it is next to impossible to go back to using the keyboard only. It feels awkward and slow in comparison.

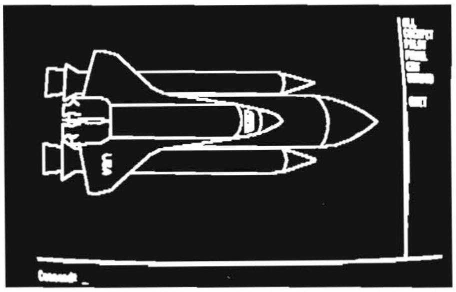
For those of you who are unfamiliar with a digitizer, this is how it works. There is a grid of wires inside the plastic surface which detects an electrical pulse from the stylus which establishes the position within .001". The digitizer therefore allows one to enter position data extremely accurately and conveniently, by touching the stylus to the desired position. In use, a cross-hair appears on the display and moves about until the stylus is touched to the tablet, then a + is placed on the display at the point selected. This method is commonly used with the large industrial CAD/CAM systems. The initial point is connnected to a "rubber band" cursor which gives the user a visual clue to the position of the stylus with respect to the display.

Configuration of the system is very simple with the Zenith, since color graphics and serial ports are built in. Graphics and text are both displayed on the same monitor. The optional digitizer (or mouse) and plotter may be attached either through one serial port with a switch, or alternatively, each may be attached to an individual port. Future Technology Inc. has also developed a power supply card for our digitizer which can be placed in the S-100 cage, so that no external power supply is required.

AutoCAD is supplied on two disks, the first consists of the copyrighted configuration and program files, and the second contains sample drawings for user familiarization and practice. The most popular with our staff is SHUTTLE, an outline drawing of the NASA Space Shuttle. It is pretty impressive to ZOOM in on the porthole, and expand the scale so that one may read the fine print on the control panel. A tremendous amount of detailed data is stored for each drawing to make this kind of thing possible. Each drawing is stored in a file listing the point coordinates of each entity, and this information may be accessed by an external program, making it possible to perform additional calculations on the data, for example, to produce a cost or price summary report. With a fixed Winchester disk, both program files and data files may reside on the hard disk, giving additional speed to the program.

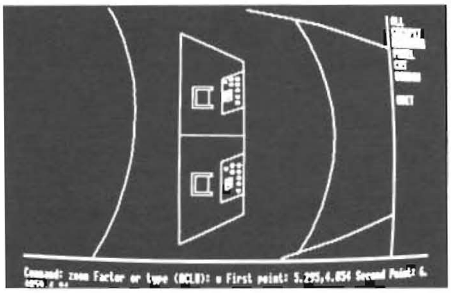
Since the AutoCAD program is menu driven and prompts for the command or data needed, it is very easy to learn, even for someone unfamiliar with computers. However, some knowledge of CAD terminology may be needed, so these basic terms are defined in the Users Manual.

In a short article such as this, it would be impractical to try to describe all the features



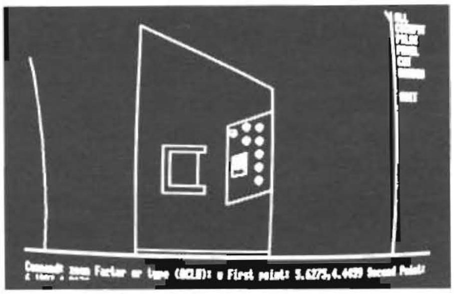
```
Command: _
```

Full drawing of the SHUTTLE.



```
Command: zoom Factor or type (ACLW): w First point: 5.295,4.054 Second Point: 6.
```

ZOOM on the window of the SHUTTLE.



```
Command: zoom Factor or type (ACLW): w First point: 5.6275,4.4439 Second Point:
```

the cockpit.

A Review of AutoCAD Computer-Aided-Design Package

available with AutoCAD, so we will stress the ones we feel are most broadly applicable to both engineering and architectural design work. Our staff is most familiar with printed circuit design, an ideal application for AutoCAD, which is basically a two-dimensional system. Floor plans and landscaping are two other obvious uses.

The following is a brief description of AutoCAD terms and commands, condensed from the User's Manual.

ENTITIES

All drawings are made up of entities, a term referring to simple geometric figures and combinations thereof: line, circle, point, arc, text, and trace (a solid line), other solids, and (created) shapes. Each entity is defined on an X,Y coordinate system, either as absolute or relative position. The display of the drawing (or portion thereof) may be ZOOMed in or out. However, the coordinates always refer to fixed positions within the drawing, only the relative size on the screen is changed.

Each entity is entered with a command and associated prompts for necessary coordinate information. This may be entered from the keyboard, or by means of a digitizer tablet. Angles are entered numerically.

SHAPES

Shapes are an efficient way to store certain subdrawings or simple figures (composed of lines only, not circles or arcs) which will be repeated often. Various scales and positions for the SHAPE can then be chosen. Two sample .SHP files are included within the AutoCAD program: for printed circuit design and for electronic circuit schematics. Each .SHP file contains several defined shapes indicated by a unique shape number. Text fonts are stored in the same way (where the shape number is the ASCII number of that character). When a SHAPE is used, it is called for by its shape name. This is most easily done by setting up a MENU and pointing to the desired SHAPE with the digitizer. Size, position, and angle must also be specified. Before use, the .SHP file must be loaded.

DRAWING LIMITS, SNAP, GRID, ORTHO

The coordinate system which locates the entities within a drawing has definite borders called drawing limits, which may be increased or decreased at any time during the editing process.

The SNAP command rounds off position coordinates to automatically align points entered to an invisible grid as selected by the user. Various SNAP resolutions may be used within the same drawing. A change in the SNAP resolution will not affect points entered before the change was made.

An optional GRID may be displayed on the screen. The resolution may be adjusted automatically to fit the size of the drawing. This grid only appears on the screen as a visual aid, and is not really part of the drawing.

The ORTHO command will force all lines to be orthogonal to the coordinates (horizontal or vertical, whichever is closer). For free hand drawing, the SNAP and ORTHO commands should be turned off.

LAYERS

AutoCAD will support up to 127 different layers on each drawing. Each layer may be made visible separately or in combination with any other layer. Individual colors are assigned to each layer (up to 127 colors possible). The layers may be used to group certain types of entities or to isolate certain parts of the drawing for clarity. The user may think of them as several separate transparent sheets layered on top of one another. In order to prepare a multicolor drawing, several layers (at least one of each color) must be used.

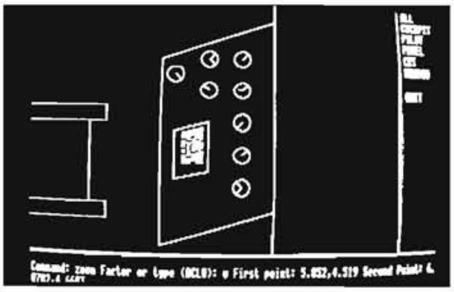
Entities may be CHANGEd from one layer to another at any time. This feature is most often used to alter the color of the entity.

WINDOW

Windows are created by marking the lower left and upper right corner. This is most easily done with a digitizer. The cross- hair stretches out into a rectangular window, surrounding the objects or area indicated.

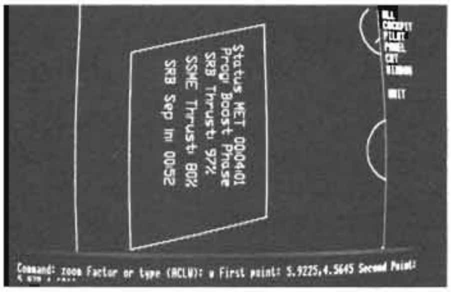
ZOOM and PAN

These commands allow various parts of the drawing to be displayed more conveniently. The ZOOM changes the size (either larger or smaller) either by a multiplication factor to affect the whole drawing or by setting a "window". PAN allows a set "window" to be shifted to another part of the



```
Command: zoom Factor or type (ACLW): w First point: 5.032,4.319 Second Point: 6.
[illegible]
```

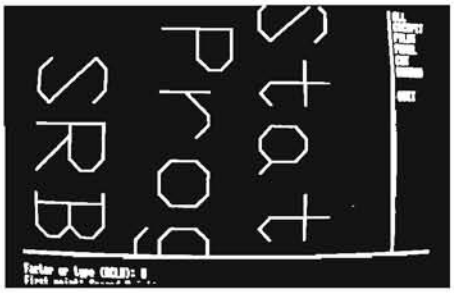
ZOOM on the instrument panel.



```
Status MET 000401
Prog Boost Phase
SRB Thrust 97%
SSME Thrust 80%
SRB Sep in 0052
```

```
Command: zoom Factor or type (ACLW): w First point: 5.9225,4.5645 Second Point:
[illegible]
```

the CRT on the instrument panel.



```
Factor or type (ACLW): W
First point: [illegible]
```

text on the CRT on the instrument panel.

drawing. Too enthusiastic use of these commands in conjunction with each other caused our SHUTTLE to be "Lost in Space". Once the original drawing limits and associated grid have gone off the screen, the command ZOOM ALL may be used to show the entire drawing, no matter how large it has become. Then the limits may be redefined if desired.

MOVE, COPY, ERASE, OOPS, CHANGE

An entity or group of entities may be edited with these commands. The MOVE, COPY, and ERASE commands seem self-explanatory. These commands are implemented by selecting an entity and indicating the displacement (for MOVE and COPY). This may be done by touching a point on the entity with the digitizer or entering coordinates from the keyboard. The OOPS command allows the user to cancel the most recent ERASE command and restores the entities which were accidently erased.

CHANGE may be used on certain entities (line, circle, text, BLOCKs) in limited ways to alter either their size or position. As mentioned above, any entity may CHANGE to another LAYER.

INSERT, BLOCK, SHAPE

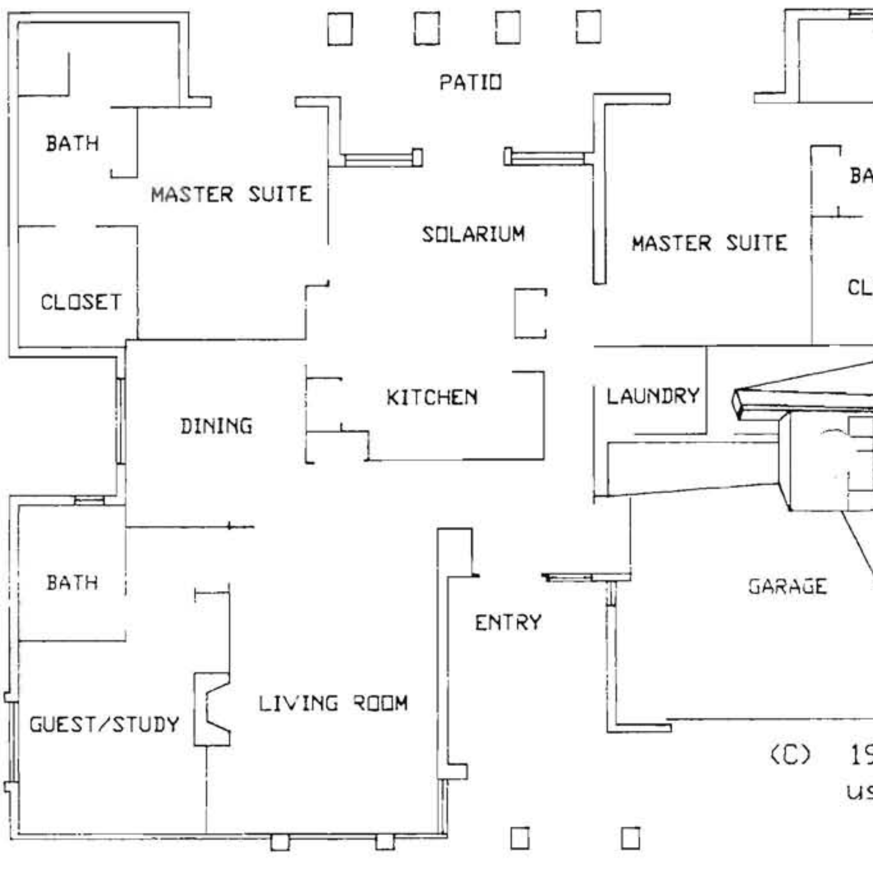
One of the most powerful features of AutoCAD is its ability to store drawings and/or parts of drawings as files, and then INSERT these into other drawings, one or more times. This is particularly necessary for any drawing with repetitious elements. The part (SHAPE or BLOCK) will be INSERTed at a user-defined scale and angle at a specified insertion point. Use of a negative scale factor will produce a mirror-image of the part. After a BLOCK or SHAPE is copied from the file to the drawing, on subsequent uses, it is copied from the first copy (with appropriate scale, rotation, and position) without having to return to the file (saving disk access time) and is included in the new drawing file.

A BLOCK is either selected from a file of previously created drawings and inserted in the new drawing, or may be created from the new drawing and repeated. After a BLOCK is assigned a name, it will be treated just like an entity. AutoCAD will support nested BLOCKs, but will not allow a repetition of BLOCK name.

MENUS

A general MENU file is supplied with the AutoCAD software, which includes the most common commands. Each drawing in the training module has a specialized .MNU file associated with it. This file will be loaded automatically with the drawing. Additional menus may be loaded at any time by selecting the MENU command.

MENUs may be created for each drawing or drawing type. Commands are displayed along the right edge of the screen and may be selected with the arrow keys or the digitizer. The screen menu is accessed along the extreme right side of the tablet (or with the arrow keys and MENU cursor). Submenus are located on the top area of the tablet. Up to forty menu choices are available for selection by touching the desired command with the stylus. Most of these menu commands go to various subcommands to complete the action chosen. These are referred to as modifiers, and usually consist of requests for data (points, angles, colors, etc.) or other additional information the computer needs to perform the task. Depending upon what type of data is needed, it may be entered either from the keyboard or with the digitizer.



The ability to create a custom MENU for each type of drawing is one of the best features of AutoCAD. Various SHAPES or BLOCKS may be entered in the MENU and then INSERTed into the drawing merely by touching the name with the stylus. The position on the drawing is also indicated with the stylus. If the size of the entity is known in advance, this may be made part of the MENU command as well.

Sequences of commands and positioning data may be set up ahead of time to perform routine operations. The creation of a custom MENU file is described in the Appendix of the User's Manual. The file is created outside of the AutoCAD program with any text editor or word processing program.

TABLET Mode

This feature makes it possible to copy drawings that are on paper into the computer. The digitizer tablet is configured to map the drawing directly into the computer's coordinate system. Large drawings may be entered in sections, which are aligned properly by recalibrating the digitizer for each partial drawing entered. The digitized drawing may then be treated just as if it had been created on the computer.

AUTOMATIC DIMENSIONING (Optional Feature)

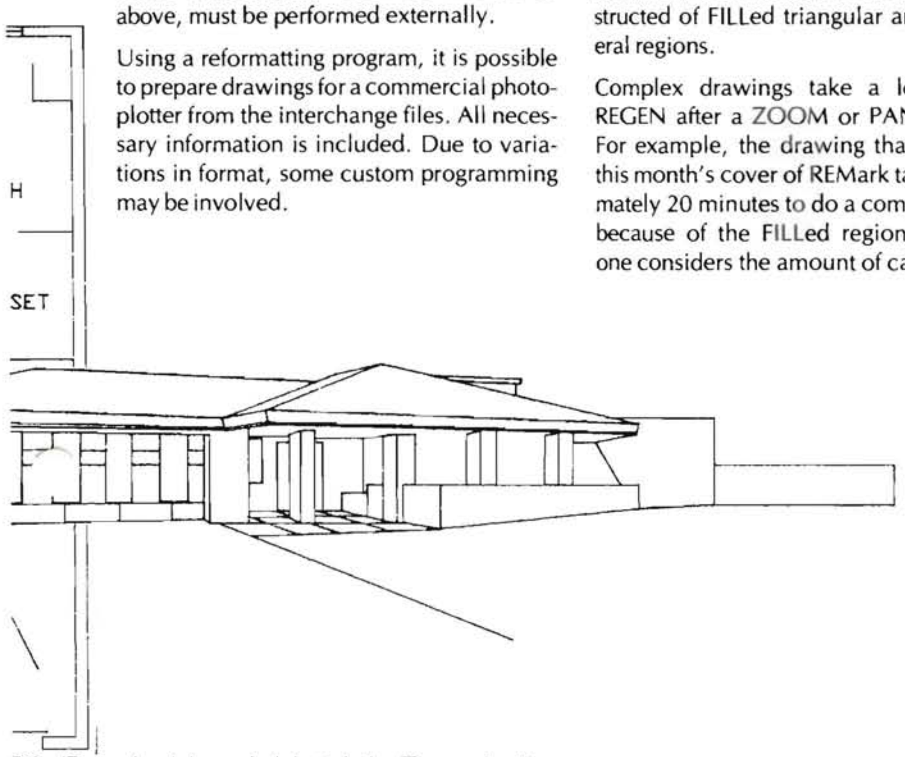
This option calculates and inserts horizontal and vertical dimensions where indicated on the drawing with associated text, witness lines, and arrows, drawn to ANSI standards: either English or metric dimensioning may be selected.

DRAWING INTERCHANGE FILES

Drawing interchange files make it possible to transfer data in AutoCAD files to an external program, in order to perform calculations or other data processing, and then to return the results for inclusion on the drawing. This feature may be used for preparing price and material lists, calculation of energy use, stress calculations, etc.

Distances between points and areas of bounded polygons are calculated and displayed by the AutoCAD program itself. Other calculations, such as mentioned above, must be performed externally.

Using a reformatting program, it is possible to prepare drawings for a commercial photoplotter from the interchange files. All necessary information is included. Due to variations in format, some custom programming may be involved.



⁷8 Frank Lloyd Wright Foundation
ed by permission

It is also possible to use the AutoCAD running on a micro- computer, such as the Zenith, as an auxiliary work station to a large CAD/CAM system, transferring the information back and forth (with some type of custom program to match formats). This would free the large system for the heavy work of three dimensional rotation, auto-placement routines, etc. which are beyond the scope of AutoCAD. This seems to be an area with great possibilities and is certainly worth further investigation.

After using AutoCAD for about a month in our office, and from observation of staff members in action, I would have to give this program an excellent rating. Since the staff members I am referring to are a professional PC design engineer and an ex-Computervision engineer/programmer, I believe they were in a position to put the program through its paces, and discover what its capabilities really are.

These particular staff members were prone to expect too much of the program and were often pleasantly surprised to see what it could do. The only real contributor to our "frustration factor" is that some of the operations are very slow. For example, SOLID and FILL seem unusually slow. FILL may be turned OFF, so that only outlines will be drawn during editing, to speed things up. There is no command equivalent to PAINT; with AutoCAD all solids are constructed of FILLed triangular and quadrilateral regions.

Complex drawings take a long time to REGEN after a ZOOM or PAN command. For example, the drawing that appears on this month's cover of REMark takes approximately 20 minutes to do a complete REGEN because of the FILLed regions. But when one considers the amount of calculation the program must do for each of the commands, the delay is not really surprising. This may be where additional memory capacity would be of value. Besides, it is unfair to compare AutoCAD to the large industrial systems and moan about the lack of tabulated cylinders and B-Splines and three dimensional rotation (which also can strain some of the big machines).

In Printed Circuit Design, quite often components or connections are duplicated or set up in arrays. If the REPEAT command is used, for example, to draw five more ICs in a row, all five will be considered as one entity by the program, and no part of any of those five may be erased without erasing the whole set of five. Rather than using the REPEAT command, we prefer to set up a custom MENU for each drawing (or set of similar drawings with the same types of components), and INSERT each component individually from the MENU. Then if a CHANGE, MOVE, or ERASE is required on one, it can be made without affecting the others.

Presently there is not a provision for hidden lines (or for dashed lines, which could indicate them in a drawing). Autodesk, Inc. will release some form of hidden line application soon.

Sometimes it is hard to remember that we are supposed to be evaluating a professional engineering tool and not a computer game, since AutoCAD is so much fun to work with. We demonstrated it a local computer show, to a very enthusiastic audience (many of whom probably did think it was a game). With the MENU placed on the digitizer tablet, a few minutes of instruction is enough for a volunteer from the audience to get started creating a drawing or modifying an existing one. Of course, to achieve professional results would require more knowledge of drafting techniques.

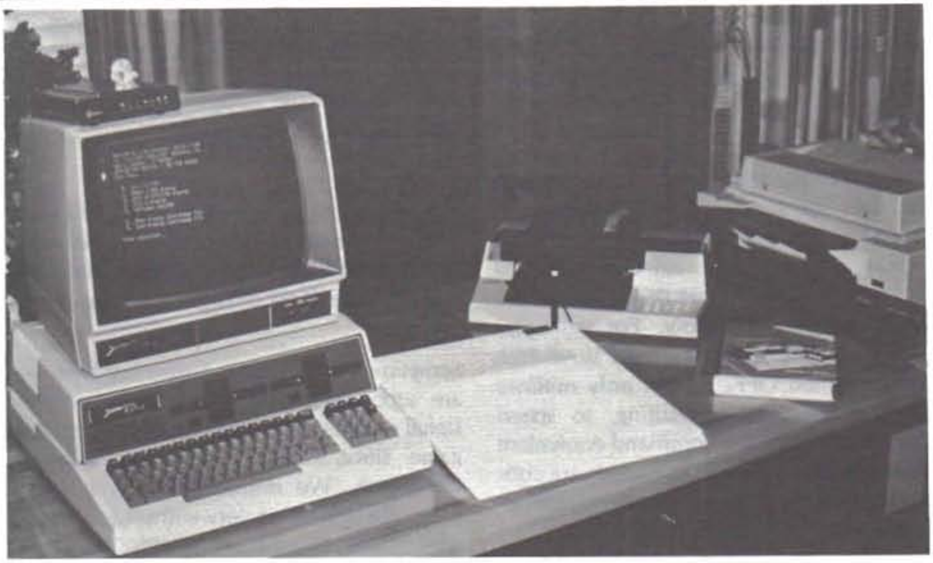
As an aid for the professional engineer, architect, builder, or anyone who must produce accurate drawings in two dimensions, the AutoCAD package is well worth consideration. We find the optional DIMENSIONing to be very useful in preparing architectural and construction drawings, since all calculations and resulting dimension lines, and numbers are inserted automatically. In an office this would result in a tremendous savings of time and effort for the draftsman.

Many X,Y plotters are compatible with the AutoCAD program. The connection is made through the serial port, as described in the Installation section at the beginning of the User's Manual. An explanation of how to hook up several types of digitizers is also included. This part of the Manual does assume some technical knowledge of the equipment involved.

Size of the drawing, colors (indicated by pen numbers, if available), scale and speed may be selected, depending upon the plotter used. Up to E size drawings may be produced with a full-sized plotter. Most plotters will be limited to smaller sizes.

The User's Manual is generally well written and easily understood. Terms and concepts are explained as they are encountered without overwhelming the reader with too much detail. More advanced technical aspects of the program are covered in the Appendices.

The HELP command brings up information on each command and its expected parameters and options. If an invalid command



is entered, a message is displayed. The user may add to the original help file any pertinent messages he wishes, such as the names of SHAPEs, BLOCKs, to be used with that command.

Unfortunately, we really had no other similar CAD program for micro-computers to compare AutoCAD to. It is way beyond the simple drawing programs we have seen previously, in the number of practical features for drafting. Other programs are just beginning to appear on the market, and we predict a lot of competition in this field. Now that micro-computers have enough power to handle the complex calculations required in a reasonable amount of time, the desktop CAD/CAM market should increase rapidly.

One of the things we liked the most about AutoCAD is its versatility and the wide range of applications that it may be suitable for. We are certainly going to continue our commitment to learn more about this valuable program and would very much like to hear from other users or soon-to-be users or anyone who shares our interest in CAD/CAM for Zenith computers.

Depending upon the response, we may offer an award for the most unusual application of AutoCAD. No hints will be given and our staff must be convinced that we never would have thought of it. Let's hear from you.

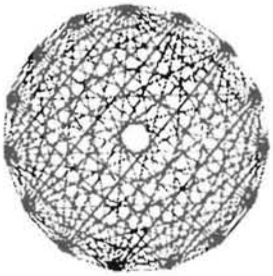
Send your cards and letters to:

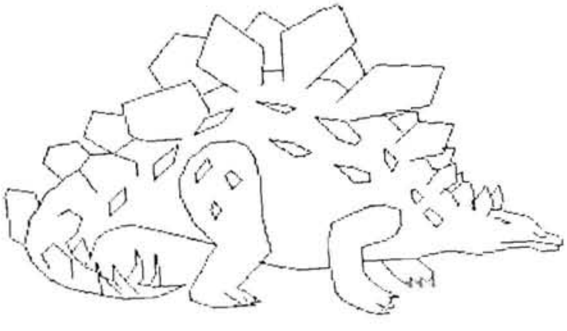
Future Technology,
Inc. 13 New Salem Street
Wakefield, MA 01880

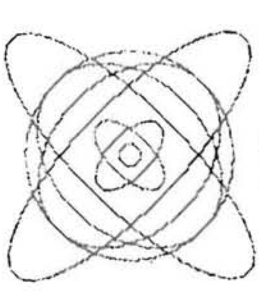
Vendor: Autodesk, Inc.
150 Shoreline Highway
Building B, Suite 20
Mill Valley, CA 94941
(415)331-0356

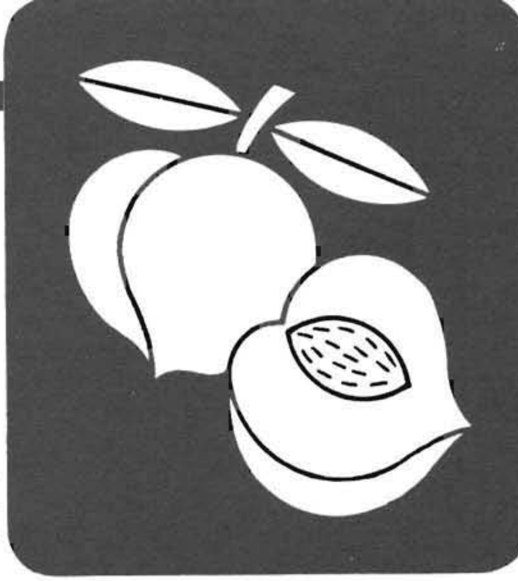
Price: $1000 AutoCAD software license for single computer
$ 500 Optional Auto Dimensioning

Machines: Zenith (Z-DOS)
IBM PC
Victor 9000
CP/M 80 computers









Part II

PeachText 5000

Your Complete Personal Productivity System

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

Introduction

The PeachText 5000 package is an interlocking group of six (6) micro-computer software programs designed to give the user a degree of control and flexibility previously available only with expensive dedicated systems. The packages are:

1. PeachText Word Processor.
2. Random House Electronic Thesaurus.
3. Spelling Proofreader.
4. List Manager.
5. PeachCalc Electronic Spreadsheet.
6. Access Pak data exchange programs.

Each of these packages are presented in a menu-driven format that allows the inexperienced user to quickly understand the functions and capabilities of the programs. Built-in "HELP" screens provide additional instructions to simplify the system use.

List Manager

I would like to discuss List Manager in detail. As I stated in Part I, I believe that List Manager is worth the price of PeachText 5000 alone for users who have a need for a data base manager!

List Manager works as a data based manager for as many as 32,000 separate records. The program's potential is limited only by disk storage space. Here is a list of some of the features:

1. User-definable format. The user decides what information he wants to include in each of his records. Up to fourteen (14) data items can be defined, any three (3) of which can be selected as index "keys" for quick access to specific records. Each index "key" can have up to twenty-three (23) characters and consist of numbers only, letters only, or any combination of numbers and letters. Each record can contain up to 509 characters.

2. Easy-to-maintain files. The file manager portion of the program gives immediate access to any individual record and allows the user to "browse" forward and backward through the file to find a record or to study the records. New records can be inserted automatically into the file without sorting. "Assumed Values" for new records speed up first-time entry of a data base. For example, if the user's records are going to have the same city, state and zip code, the "assumed values" function can be used to preset these fields to the appropriate data so that they do not have to be keyed in for each record. As the record is created, these fields will be automatically inserted.

3. Make file format changes easily. Since a separate definition is maintained for each file, the user may define as many files as required. The definitions can be altered at any time. If the user sets up a new definition for the file, List Manager will automatically convert all the records in that file to the new definition. For example, assume that a data base of one hundred (100) names and addresses has been established, but the phone number field has been left out. With a new definition, the phone numbers can be entered into each record without recreating the file.

4. Mix and match. One file can easily be combined with another. Mailing lists may be exchanged with another user, even if the files have different definitions. The user can give List Manager instructions about those items in the first file that correspond to the items in the second file and then combine them.

5. Create mailings with a personal touch. By using PeachText and List Manager together, the user can generate individualized mass mailings! The user can choose any of the fields set up in the file in combination with PeachText's variable and conditional commands. Thus, the user can send different letters to persons on the same mailing list depending on information in individual records.

6. Design personalized reports. By using Peachtree's "Visionary" screen manager, List Manager shows a screen representation of the report. The user can position the desired data items that will be printed. A "test" print out can be run before running the entire report. The reports can be printed with up to five (5) records across the page and the user can decide when to include page feeds. List Manager can store up to fourteen (14) different user designed print formats for later use.

7. Easy sorting. The file records can be sorted by any field or any combinations of up to three (3) fields, either in ascending or descending order.

8. Record control. By matching selected characteristics of individual records, List Manager will include or exclude records from any report. The total number of records selected will be displayed on the screen at the end of the report. If the user is unsure of the output of the report, it can be displayed on the screen before printing.

9. Duplicate labels (for mailing applications). The user can choose how many labels will be printed for EACH record so the user will not have to type (or rerun) the extra labels at the end of the label main run.

10. Multiple applications. List Manager can be used for nearly any type of file including membership lists, in-house telephone lists, glossaries, etc.

It would require many more pages to outline all the special features that I have found. The program's real worth lies in its many uses. I am sure that you HUG members can find hundreds of practical applications for this program! I would like to "highlight" a few ideas:

1. A user needed to identify his customers in a CERTAIN city who had purchased CERTAIN products at a CERTAIN price in the last year. This information will be useful to determine if he should advertise in this city, what are the best sellers in this city, and at what price the volume of sales changed considerably. The user could also use this information to conduct a direct mail campaign to these customers to increase follow-up sales!

2. A user has a file of nearly 1,000 customers. When the phone company changed the Area Code for part of his territory, his records were incorrect! The user simply redefined the Area Code field and assigned equivalents to which CERTAIN old Area Codes should be changed to. The file was then automatically updated by List Manager.

3. A user purchased another business that he wished to merge with his present business. The new business had a mailing list, although it contained some different items and definitions. If the user tells List Manager which information in the old list corresponds to information in the new one, the program will incorporate the two files into a single source and create fields for the new categories with no equivalents.

4. A customer phoned the user demanding an added discount for being a preferred, high volume customer. He claimed that he had sold over $220,000.00 of widgets in six (6) months and therefore qualified for the added discount. The user wrote the customer's name down incorrectly—Johnston, Howard. If the user enters the first four (4) letters, "JOHN", that he was sure he had correct; List Manager can then locate the customer's names meeting those letters. From these names the user finds the name—Johnson, Howard. This correct name and the associated records were located in less than one minute. The records showed that this customer, Mr. Johnson, had a volume of $120,000.00 in the last eight (8) months and did not qualify for the added discount.

I could expand this list of applications to any person or business that manages information! I am sure that the reader's imagination can do even better. But, only added profits will really tell the value of the package.

Access Pak

Next, I would like to review Access Pak for you. It is a Data Exchange package that consists of the following four (4) programs:

1. Information Access (see the next part of this article).
2. Wordstar-to-PeachText translator.
3. Easywriter I-to-PeachText translator.
4. VisiCalc-to-PeachCalc translator.

The file conversion programs translate ASCII files produced by the other software into proper format for PeachText and PeachCalc. They also convert any embedded commands into Peachtree software equivalents. A few commands are not resolved and these differences are displayed on the screen and/or printed out.

The reader can see that these programs eliminate any obsolescent of the user's files from previously used software and enables the user to fully use PeachText 5000 Integrated Modules!

Information Access

In recent years, the micro marketplace has been flooded with individual software applications that can balance your checkbook, compose a well-written letter, maintain a mailing list for the user, help your doctor diagnose illnesses, etc. Yet the collective power and potential of this collection of software has been squandered because of an inadequacy in the programs: A LACK OF INTERACTIVITY!

The mark of quality software producers WILL NOT be how many programs that they can write, but how their programs can work TOGETHER to SOLVE concrete problems from start-to-finish! Peachtree has met this challenge with PeachText 5000 and "Information Access"!

Information Access serves as a "bridge" between Peachtree's Accounting packages and the PeachText 5000 packages. It allows the user to create procedures that will convert accounting data files to formats that are readable by PeachText, PeachCalc, and/or List Manager. For example, the user could extract data from his Accounts Receivable customer files to construct a PeachText/List Manager personalized form letter to be sent to all customers with accounts past due! WHERE can you find "Interactive" programs like this? How can anyone who has this or similar problems be without this software package and an H/Z-100? The concept of TOTAL INTEGRATION is needed in the computer user's "arsenal" now!

To accomplish this TASK, Information Access uses a comprehensive "data dictionary" to describe data in the accounting system files and any other file created by Information Access. The user selects the data he needs and puts it into fields he defines with maximum and minimum ranges (if desired). The user can also introduce additional information, such as text, numbers, and dates. The user can also do arithmetic computations. Information Access then "reaches" into these data files and builds a new file that can be used with the PeachText 5000 packages. The user controls all information and how he wants it to appear.

Some of the main features of Information Access follow:

1. Machine-readable descriptions of the nine most significant accounting system data files are included.

2. Ability to convert the accounting user files to PeachText, PeachCalc, and List Manager formats.

3. Ability to define output formats. Information Access does not produce reports but provides the data required for PeachText or PeachCalc to produce the reports.

4. Ability to add user-defined data to the output file.

5. Ability to perform additional calculations during the information access.

6. Ability to select output records based on high and/or low limits of selected fields.

7. Adaptable, with a little experience, to files other than those included as part of Information Access. For example, Transaction Files.

The best way to complete the description of the value and versatility of Information Access is to "highlight" some possible applications for the readers:

1. Use PeachText/List Manager to write a personalized form letter to all customers who have paid promptly (say within 30 days) for the past year, congratulating them for their "good" payment record. This promotes "good will"!

2. Use PeachCalc to "kick around" an expense budget for next year using the required balances of all expense accounts from the General Ledger. The user could also compare the actual expenses

to budget amounts.

3. Print mailing labels using List Manager for the employees listed in the Payroll file who have been with the company over five (5) years and use this data with PeachText/List Manager to prepare an individualized letter advising the employees about their 5, 10, 15, etc. year Service Pins.

4. Generate a Sales Report for a sales meeting from the Accounts Receivable transactions using PeachCalc to format and total the information.

These are but a few practical applications. The potential of Information Access is limited only by the user's imagination. The linking of Peachtree's Accounting packages with the PeachText 5000 package represents an evolutionary innovation in software technology that will significantly improve any serious computer user's productivity!

Closing

I hope that in these few pages I used in Parts I and II, I have been able to explain this innovative concept of "TOTAL INTEGRATION"! If I did, I hope that the reader's imagination will make this system and his H/Z-100 worth the investment. If my enthusiasm for these products has been "carried away", I apologize. I have really enjoyed telling you about something that I believe is really NEW!

I would like to leave you with one thought. The concept of "total integration" would NOT BE POSSIBLE if there were no 16-bit H/Z-100 type computers with their added RAM capacity. This is true NOW for this software package and will be true for many other NEW software packages that will be coming. So, HUG members, join the H/Z-100 owners!

Part III

An Introduction To 'C'

Brian Polk
86-02 Little Neck Parkway
Floral Park, NY 11001

This is the third of a series of articles designed to introduce the 'C' programming language.

At the top, several comments must be made about the program in the prior article. That program printed out the special 'C' character constants in decimal, octal, and hex. First, there was one mistake in the listing which would have caused an error when the program was compiled, but the correction should have been obvious if you understood what the program was doing. The error was in the last 'format' statement, which printed out 'bit pattern'. It was missing the closing double quotes ("). That line should have read:

```
format("bit pattern: dec=%d oct=%o hex=%x\n");
```

The next problem with the program involved the release of the C/80 compiler being used. I had been using release 1.5 of C/80 and was surprised to hear that there was a 2.0. What was even more surprising was that the program would not run under release 2.0. Subsequently I found out that version 3.0 was just released along with 'Mathpack' to allow 'long' and 'float' variable types. (More on 'Mathpack' in future articles.) The problem with the new release was the 'format' and 'printf' statements. Their usage changed in order to correspond to standard 'C' syntax. This now makes using the C/80 compiler easier because the 'printf' statement has the same format used in any 'C' programming language book. This fact also makes our program significantly easier to write in terms of size and readability. The program for releases 2.0 and later looks like this:

```
#include "printf.c"
main()
/* This is a sample 'C' program which prints out
   the 'C' special character constants in decimal, octal and hex.
   Modified to be used with C/80 release 2.0 and later. */
     {
     printf("newline: dec=%d oct=%o hex=%x\n",'\n','\n','\n');
     printf("tab: dec=%d oct=%o hex=%x\n",'\t','\t','\t');
     printf("backspace: dec=%d oct=%o hex=%x\n",'\b','\b','\b');
     printf("carriage return: dec=%d oct=%o hex=%x\n",'\r','\r','\r');
     printf("form feed: dec=%d oct=%o hex=%x\n",'\f','\f','\f');
     printf("backslash: dec=%d oct=%o hex=%x\n",'\\','\\','\\');
     printf("apostrophe: dec=%d oct=%o hex=%x\n",'\'','\'','\'');
     printf("bit pattern: dec=%d oct=%o hex=%x\n",'\123','\123','\123');
     }
```

As you can see, this greatly simplifies the program. The 'format' statement no longer exists. It would probably have been better if release 2.0 rejected the format statement and generated a compile error. The contents of the 'format' statement have been combined with the 'printf' statement resulting in one statement controlling both the print formats and the variables to be printed. Otherwise the concept is the same; each succeeding variable in the list refers back to the next indicated formatting character. From this point on I will be using release 3.0 of the C/80 compiler. I wish to thank Harry Albaum of the Jericho HUG (JERIHUG) for pointing out and assisting in the debugging of this problem.

On to new things.

The program we are going to write today is called 'more' and is a facsimile of the UNIX command of the same name. This command lists an ASCII file at the terminal one screen at a time. In writing this program, we will have to introduce several new concepts and commands.

The first is the 'if' statement. This is equivalent to an 'if' statement in any other language: if a specified condition is true, then the following statement is executed; otherwise, the following statement is skipped. In 'C', one statement can be replaced by many statements simply by surrounding the statements with 'curly brackets' ('{' and '}'), just as we surround our program with brackets. Therefore, the one statement to be executed can actually be several.

The next statement is the 'for' statement. This is similar to a 'DO' statement in FORTRAN or a 'FOR' statement in BASIC. The 'C' 'for' statement is composed of three parts. The first part is executed only once at the beginning of the 'for' loop. This is convenient for initializing variables. The second part is the condition test. As long as this condition is true, the 'for' loop continues. The third part is executed for each execution of the 'for' loop. This section is often used for incrementing a variable each time through the loop. Besides these three parts, one statement (or several via the curly bracket method) becomes the 'body' of the 'for' loop. Here's an example:

```
BASIC:  X=0                          C:  for (x=0, i=1; i <= 10; i=i+1)
        FOR I=1,10                             x=x+i*i;
        X=X+I*I
        NEXT I
```

The example is a loop which sums the squares of the first ten numbers. I will presume that you understand the BASIC example. Let's break down the 'C' example into the three parts mentioned above.

Part 1: x=0, i=1;
We are initializing two variables. We can put as many statements as needed in this section by separating them with commas. 'C' also lets us do multiple assignments of variables to the same constant, as in 'x=i=0'.

Part 2: i <= 10;
This is the conditional test. As long as this condition is true, the 'for' loop will continue. So as long as 'i' is less than or equal to '10', the 'for' loop continues.

Part 3: i=i+1
This statement will be executed for each iteration of the 'for' loop. This part can also be multiple statements, the same as in Part 1. **Note:** this section is executed AFTER the 'body' of the 'for' statement.

Body: x=x+i*i;

The body here only contains one statement, so we don't need to enclose it in brackets. This statement is a simple assignment statement with the same operator precedence as in other languages. Note the semicolon terminating this statement and the lack of a semicolon after the 'for'. If all the processing can be accomplished within the scope of the three parts of the 'for' statement, the semicolon would follow the 'for' statement, indicating there was no 'body'.

Another new statement is the 'while' statement. This indicates a condition which, while true, causes the following statement to be executed. Obviously, the following statement must do something to at some point alter the condition, or else an infinite loop would exist.

Next, some new I/O statements.

1) getchar — gets one character from the standard input device (normally the terminal).

2) putchar — opposite of 'getchar'.

3) getc — gets one character from the specified channel number, with '0' always being the terminal.

4) putc — opposite of 'getc'.

One more thing before we can write our 'more' program. 'C' has several other operators besides addition, subtraction, division, and multiplication. One we will be using is '++'. When used immediately before or after a variable, it increments the variable by 1. This is immensely useful since very often (as in our example above) we have to increment a variable by one. The difference between putting the operator before or after a variable will be covered in a following article. For now, suffice it to say that it makes no difference. For example:

i=i+1   now becomes   i++ or ++i

Now, finally, we are ready for our program. I hope this ties together several of the concepts I have covered and gives you some insight into further uses of the commands.

Here's the program:

```
#include "printf.c"
#define EOF -1
main()
{
int c, line_number;

for (line_number=0; (c=getchar()) != EOF; putchar(c))
   {
   if (c == '\n')
      line_number++;
   if (line_number == 23)
      {
      putchar(27);
      putchar(106);  /* save the cursor position */
      putchar(27);
      putchar(120);
      putchar(49);   /* turn on the 25th line */
      putchar(27);
      putchar(89);
      putchar(56);
      putchar(32);   /* position cursor on the 25th line */
      putchar(27);
      putchar(112);  /* enter reverse video */
      printf("--More--");
      putchar(27);
      putchar(113);  /* exit reverse video */
      while (getc(0) != '\n');  /* wait for a carriage return */
      putchar(27);
      putchar(107);  /* return cursor to saved position */
      putchar(27);
      putchar(121);
      putchar(49);   /* turn off 25th line */
      line_number = 0;
      }
   }
}
```

Some things to think about for next time:

1) We haven't explicitly indicated the input file anywhere. If we execute this program, it will wait for input from the terminal — not very useful! Here's a clue: read the section on I/O redirection in the C/80 guide.

2) Why did we use 'getc(0)' versus 'getchar()' while waiting for a carriage return. Clue: it has to do with the answer to number 1.

3) This program shows how to use the H-19 special configurations from a 'C' program. You might want to experiment trying others (e.g. try turning off the cursor while waiting for the carriage return).

We'll spend more time looking at this program next time. If you can put the pieces of the puzzle together by yourself — congratulations. Otherwise....

'C' you later!

HUG NEW PRODUCTS

885-1131[-37] HDOS
885-1233[-37] CP/M
885-3006-37 Z-DOS
CHEAPCALC $20.00

Introduction: CheapCalc is a minimal but useful "spread sheet" program that can introduce you to spread sheet computing at little cost. It was developed from the program that originally appeared in the February 1983 issue of REMark (#38), and has been improved considerably and compiled. With CheapCalc, your computer screen becomes the window to a large worksheet onto which you can write mathematical problems and have them solved almost immediately.

Requirements: The HDOS version requires 56K of RAM in an H/Z-89 or 90 computer or an H8 with a Heath/Zenith terminal (H/Z-19, H/Z-29, etc.). The CP/M version requires 64K of RAM for the above systems, and will also work on an H/Z-100 with CP/M-85. The Z-DOS version requires 128K of RAM.

The following files are included on the CheapCalc disks:

885-1131[-37]		885-1233[-37]		885-3006-37	
README	.DOC	README	.DOC	README	.DOC
CC	.ABS	CC	.COM	CC	.EXE
CC	.HLP	CC	.HLP	CC	.HLP
SAMPLE	.CAL	SAMPLE	.CAL	SAMPLE	.CAL
CC	.BAS	CC	.BAS	CC	.BAS

Program Authors:
Original program by William V. R. Smith
First Heath version by Bob McFarland
Compilable version by P. Swayne, HUG

CC.ABS, CC.COM, CC.EXE — This is the compiled, ready-to-use CheapCalc program. It runs by itself without BASIC or any other language required. It supports a worksheet containing 40 rows and 15 columns. Operations supported are addition (+), subtraction (-), multiplication (*), division (/), exponential (↑), and SUM. (The SUM operator lets you add up to an entire row or column of numbers.)

All calculations are done in double precision with 16 significant digits. Numbers can be displayed on the screen in 4 formats: Floating, Dollars-and-Cents, Integer, and Graphic (a line of asterisks represents each number). In the dollars-and-cents mode, rounding off to the nearest cent is done to correct possible binary floating point errors. (To illustrate this kind of error, run MBASIC and enter PRINT 100-99.99).

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

CC.HLP — This is a file containing instructions for CheapCalc. When you use the Help command while running CheapCalc, this file is loaded and displayed on your worksheet (without destroying any data you have entered), so you can get help at any time while running the program.

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

CC.BAS — This is the source code for CheapCalc. It can be run using MBASIC (HDOS or CP/M) or ZBASIC (Z-DOS), but with performance considerably degraded compared to the compiled version.

Comments: CHEAPCALC will not only introduce the user to the workings of a spreadsheet, but will provide many useful applications of a spreadsheet.

885-3007-37 Z-DOS
CP/EMulator $20.00

Introduction: CP/EMulator is a program that lets you run standard 8-bit CP/M programs under Z-DOS. Nearly every program that runs under CP/M-85 will work under CP/EMulator including MBASIC, word processors, assemblers, debuggers, etc. It offers the convenience of having your most used programs, whether CP/M or Z-DOS, on the same disk, and lets you do such things as editing Z-DOS files with CP/M editors, etc.

Requirements: CP/EMulator requires Z-DOS on an H/Z-100 series computer and 128K of RAM. (CP/EMulator will not work on the ET-100.)

The following files are included on the HUG CP/EMulator disk:

README .DOC
CPM .COM
CPM .DOC
CPM .ASM
CMD .SYS
CMD .ASM
HD5COPY .EXE
HD5COPY .ASM
EDIT .COM
EDIT .ASM
EDIT .DOC

Program Authors:
CPM - Robert A. Metz, Modified by P. Swayne, HUG
CMD - Developed from ZCPR by P. Swayne, HUG

HD5COPY - Carl H. Eaton
EDIT - HUG Staff

Program Content:

CPM — This is the CP/EMulator program. It supports nearly all BDOS calls of real CP/M and the I/O-related BIOS calls. It also supports a CP/M-85-style Tic counter, which allows you to run HUG action games designed for CP/M-85. It can be used in two modes: the direct mode and the command mode. In the direct mode, it runs a single CP/M program and returns to Z-DOS when the program is finished. In the command mode, it loads in the file CMD.SYS, which provides a CP/M-like environment with the regular CP/M commands such as DIR, SAVE, etc. available, and you can run any number of CP/M programs. The command BYE returns you to Z-DOS when you are through running CP/M programs.

CP/EMulator has certain advantages and disadvantages over real CP/M. Some of the advantages are:

1. You can put your favorite programs, whether Z-DOS or CP/M, on the same disks.

2. CP/EMulator provides a very large 63K TPA. This means that programs have more memory space than under CP/M-85.

3. CP/EMulator allows you to use CP/M editors and word processors on Z-DOS files.

Some disadvantages of CP/EMulator are:

1. Extension programs such as DESPOOL or HUG's KEYMAP may only work while you are running programs that use BIOS calls for I/O, such as MBASIC. Use of such programs with CP/EMulator is, therefore, very limited.

2. Programs that use non-I/O BIOS calls (including disk calls) will not work at all. This includes many utility programs and some directory programs. However, HUG's DIR19 program works because CP/EMulator constructs simulated disk parameter and allocation tables. Other programs requiring these tables may or may not work.

CMD — This program is a system command processor for CP/EMulator that provides the usual CP/M command such as DIR, ERA, SAVE, etc. It was developed from the public domain program ZCPR, and has many of its advanced features. Because it uses no Z-80 code, it could possibly be used as a replacement for the CCP of CP/M-85, but no development has been done in that direction.

HD5COPY — This program is similar to RDCPM except that it copies files from HDOS disks instead of CP/M disks. It is included on this disk because HUG's HRUN HDOS emulator will run under CP/EMulator, and it provides a way to get HDOS programs to your Z-DOS disks. However, it is somewhat limited in that it can only read 5.25-inch soft sector double density, single or double sided HDOS disks.

EDIT.COM — The standard CP/M editor ED will not work properly under CP/EMulator, and since we have not been able to determine why, we have included HUG's EDIT program on this disk.

Comments: CP/EMulator is a program to add greater flexibility to the serious ZDOS and CP/M-85 user.

885-3008-37 Z-DOS

Z-DOS UTILITIES $20.00

Introduction: This disk contains several utilities to aid the Z-DOS user, including two Modem communication programs, a Menu executive, printer control programs, a directory program, and a screen dump program for IDS Prism color printers.

Requirements: These programs will run on any H/Z-100 computer with Z-DOS and standard equipment. The printer programs require the specific printer they were intended for use with.

The following files are included on the Z-DOS Utilities Disk:

README	.DOC	X	.BAT	S	.BAS
TERMZ100	.COM	XA	.BAT	T	.BAS
TERMZ100	.ASM	XM	.BAT	PRISM	.COM
KTBL	.ASM	AUTOEXEC	.BAT	PRISM	.ASM
TERMZ100	.DOC	BUILD	.BAS	PRINTEK	.COM
DTERM	.COM	SDUMP	.COM	PRINTEK	.ASM
DTERM	.ASM	SDUMP	.ASM	PSET25	.COM
MENU	.COM	MENU	.BAS	PSET25	.ASM
MENU	.ASM	B	.BAS	PSETMX	.COM
LODR	.ASM	D	.BAS	PSETMX	.ASM
MENU	.DOC	F	.BAS	DIR100	.COM
MENU	.DAT	L	.BAS	DIR100	.ASM
MENU	.TBL	P	.BAS	COLOR	.COM
				COLOR	.ASM

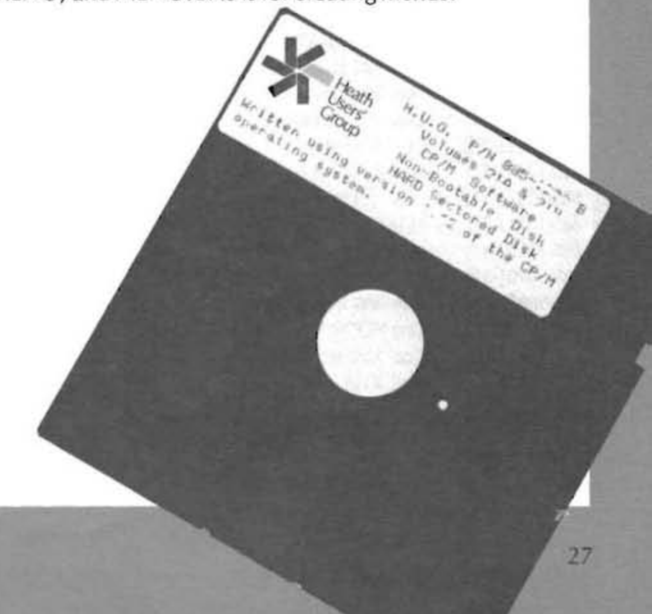
Program Authors:
TERMZ100, DTERM, MENU, SDUMP,
PRISM, PRINTEK, COLOR — Robert A. Metz
PSET25, PSETMX, DIR100 — P. Swayne, HUG

Content of Programs:

TERMZ100 — This is a modem control program patterned after the Heath/Zenith CPS program. Features include function key control of operating commands, emulation of different terminals via a user alterable table, user selectable baud rate, ASCII file upload and download, and binary file upload and download with other computers using TERMZ100.

DTERM — This is a "dumb terminal" modem program, provided for those who want a modem program that does not intercept control characters, escape sequences, etc. This program intercepts only Control-E, which is used to return control to Z-DOS.

MENU — This program provides the ability to run other programs from a screen menu. Once installed, you only need to move an indicator to the name of the function you want to perform with the arrow keys, and press the RETURN or ENTER key. This program could be of help to your friends who are new users and are unfamiliar with computers. However, a knowledgeable person will be required to set the program up for the user. The .BAT files on this disk are for use with MENU, and MENU.BAS is for creating menus.



SDUMP — This is a screen dump utility for IDS Prism color printers. Included with the program are modified versions of the Zenith demonstration programs that will dump their displays to the printer when the F10 key is pressed.

PRISM — This is a program that allows you to set up several parameters on an IDS Prism printer before using it. Some of the selectable parameters are ribbon select, ribbon advance, normal or enhanced characters, fixed or proportional spacing, draft or quality mode, justify mode, character size, and lines per inch.

PRINTEK — This is a program that allows you to set up parameters on a PRINTEK printer. Selectable parameters include normal or enhanced characters, draft or quality mode, character size, fixed or proportional spacing, and lines per inch.

PSET25 — This is a Z-DOS translation of the PSET25 CP/M program previously released by HUG. It allows you to set up character size and lines per inch on an H/Z-25 or H/Z-125 printer.

PSETMX — This is a Z-DOS translation of HUG's PSETMX CP/M program. With it, you can set compressed, emphasized, double strike, or normal characters, and lines per inch on an MX-80 printer.

DIR100 — This is a Z-DOS translation of the DIR19 disk directory program from HUG. It can display up to 80 files on the screen, and allows you to "page" through screens of files for disks with more than 80 files. It can alphabetize files, or present them in their "natural" order, and can show hidden files. File size information is shown for each file, and for the whole disk. Graphic lines are used to make the display more readable.

Comments: This ZDOS utilities disk brings many useful utilities to the ZDOS user.

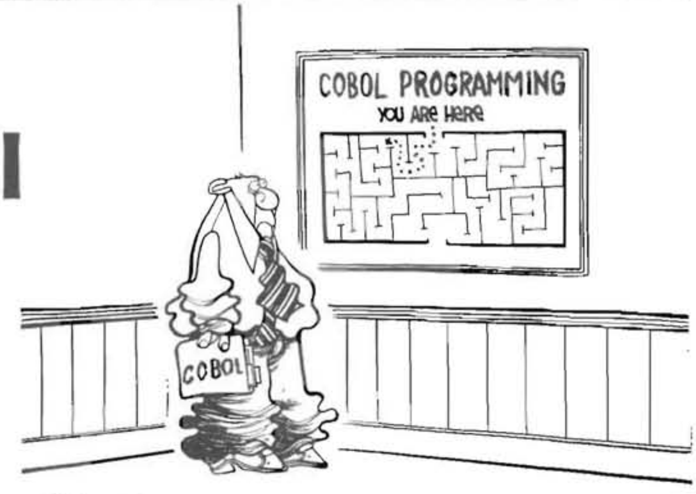
HUG Price List

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	REMark Issue
HDOS			
885-1022[-37]	HUG Editor (ED) Disk H8/H89	$ 20.00	44
885-1029[-37]	Disk II Games 1 H8/H89	$ 18.00	40
885-1030[-37]	Disk III Games 2 H8/H89	$ 18.00	45
885-1038[-37]	Wise on Disk H8/H89	$ 18.00	42
885-1042[-37]	PILOT on Disk H8/H89	$ 19.00	42
885-1044[-37]	Utilities Disk VI	$ 18.00	43
885-1055[-37]	MBASIC Inventory Disk H8/H89	$ 30.00	44
885-1060[-37]	Disk VII H8/H89	$ 18.00	40
885-1062[-37]	Disk VIII H8/H89 (2 Disks)	$ 25.00	40
885-1064[-37]	Disk IX H8/H89	$ 18.00	42
885-1066[-37]	Disk X H8/H89	$ 18.00	45
885-1067[-37]	Disk XI H8/H89 Games	$ 18.00	40
885-1071[-37]	MBASIC SmBusPk H8/H19/H89	$ 75.00	41
885-1078[-37]	HDOS Z80 Assembler	$ 25.00	42
885-1079[-37]	Page Editor PAGED	$ 25.00	43
885-1083[-37]	Disk XVI Misc. Utilities	$ 20.00	43
885-1086[-37]	Tiny HDOS Pascal H8/H89	$ 20.00	40
885-1088[-37]	Disk XVII MBASIC Graphic Games	$ 20.00	45
885-1089[-37]	Disk XVIII Misc H8/H89	$ 20.00	41
885-1090[-37]	Disk XIX Utilities H8/H89	$ 20.00	41
885-1092[-37[	Relocating Debug Tool H8/H89	$ 30.00	44
885-1093[-37]	Dungeons and Dragons H8/H89	$ 20.00	43
885-1096[-37]	MBASIC Action Games H8/H89	$ 20.00	45
885-1097[-37]	MBASIC Quiz Disk H8/H89	$ 20.00	41
885-1107[-37]	HDOS Data Base System H8/H89	$ 30.00	42
885-1108[-37]	HDOS MBASIC Data Base System	$ 30.00	41
885-1111[-37]	HDOS MBASIC Games H8/H89	$ 20.00	44
885-1112[-37]	Graphics Games Disk	$ 20.00	43
885-1113[-37]	HDOS Fast Action Games H8/H89	$ 20.00	44
885-1115[-37]	Navigational Program	$ 20.00	46
885-1118[-37]	MBASIC Payroll	$ 60.00	46
885-1119[-37]	BH BASIC Support	$ 20.00	46
885-1120[-37]	HDOS "WHEW" Utilities	$ 20.00	46
885-1121	Hard Sectored Support Package	$ 30.00	37
885-1122[-37]	MicroNET Connection	$ 16.00	37
885-1123	XMET Robot Cross Assembler	$ 20.00	40
885-1124	HUGMAN & Movie Animation Pkg	$ 20.00	41
885-1125	MAZEMADNESS	$ 20.00	41
885-1126	HDOS UTILITIES by PS:	$ 20.00	42
885-1127	HDOS Soft Sector Support Pkg	$ 30.00	45
885-1127[-37]	HDOS Soft Sector Support Pkg	$ 30.00	45
885-1128[-37]	DISKVIEW	$ 16.00	46
885-1129[-37]	CVT Color Video Terminal	$ 20.00	46
885-1130	HDOS Star Battle	$ 20.00	45
885-8010	CHECKOFF	$ 25.00	46
885-8015	TEXTSET Formatter	$ 30.00	42
885-8016	Morse Code Transceiver Ver 2.0	$ 20.00	41
885-8017	HDOS Programmers Helper	$ 16.00	42
885-8021	HDOS Student's Statistics Pkg	$ 20.00	44
885-8022	HDOS SHAPES	$ 16.00	45
885-8024	BH BASIC Utilities Disk	$ 16.00	46
CP/M			
885-1211 [-37]	Sea Battle	$ 20.00	36
885-1222 [-37]	Adventure	$ 10.00	36
885-1223 [-37]	HRUN HDOS Emulator	$ 40.00	37
885-1224 [-37]	MicroNET Connection	$ 16.00	37
885-1225 [-37]	Disk Dump and Edit Utility (DDEU)	$ 30.00	38
885-1226 [-37]	CP/M Utilities by PS:	$ 20.00	38
885-1227 [-37]	CP/M Cassino Graphic Games	$ 20.00	38
885-1228 [-37]	CP/M Fast Action Games	$ 20.00	39
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 Reference Utilities for MBASIC	$ 20.00	43
885-1232 [-37]	CVT Color Video Terminal	$ 20.00	46
885-3003 [-37]	ZTERM Modem Package	$ 20.00	36
885-8011 [-37]	CHECKOFF	$ 25.00	46
885-8012 [-37]	Modem Appl. Effector (MAPLE)	$ 35.00	36
885-8018 [-37]	FAST EDDY Text Editor and BIG EDDY	$ 20.00	46
885-8019 [-37]	DOCUMAT Formatter and DOCULIST	$ 20.00	43
885-8020 [-37]	CP/M RF Computer Aided Design	$ 30.00	44
885-8023 [-37]	CP/M 85 MAPLE	$ 35.00	45
ZDOS			
885-3004-37	ZBASIC Graphic Games Disk	$ 20.00	37
885-3005-37	ZDOS ETCHDUMP	$ 20.00	39
MISCELLANEOUS			
885-0004	HUG 3-Ring Binder	$ 5.75	
885-4001	REMark VOLUME 1, Issues 1-13	$ 20.00	
885-4002	REMark VOLUME 2, Issues 14-23	$ 20.00	
885-4003	REMark VOLUME 3, Issues 24-35	$ 20.00	
885-4600	Watzman/HUG ROM	$ 45.00	41

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

COBOL Corner III



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

Introduction

Welcome back to "COBOL Corner". We are going to assume that you have your COBOL system up and running AND know how to compile and execute a COBOL program. If not, please review and read and work with our previous articles and your COBOL-80 Manual! If I have not made these procedures clear or if you are having problems, now is the time to work them out. Your success with this and following "COBOL Corner" articles will require your understanding and familiarity with this information. If you feel that you cannot work out your problems, send me a SASE, business size, with a complete description of your problem. If it is a problem with HDOS or some other COBOL compiler, I may not be able to be of help. If it seems to be a general problem, I will bring it up in "COBOL Corner" and find out if one of our readers can be of help. Please, no phone calls. I might state again that this is your "COBOL Corner"; so, if you have any constructive criticism, suggestions or other questions to raise, send them to me!

Hierarchy of a COBOL Program

1—DIVISIONS.

Every COBOL program MUST have four (4) specific divisions. Each division MUST appear in the proper sequence.

2—SECTIONS.

ENVIRONMENT and DATA DIVISIONS contain sections of specific names. IDENTIFICATION DIVISION never has sections. The PROCEDURE DIVISION may have sections and they will be named by the programmer.

3—PARAGRAPHS.

DATA DIVISION does not have paragraphs. IDENTIFICATION and ENVIRONMENT DIVISIONS have paragraphs with specific names. PROCEDURE DIVISION has paragraphs with names that are assigned by the programmer.

4—STATEMENTS.

Appear in the PROCEDURE DIVISION. A Statement is a COBOL syntactically valid combination of words and symbols beginning with a COBOL Verb and ending with a Period.

COBOL Coding Line

Each line of COBOL coding (writing the program on Coding Forms) has 80 columns divided into five (5) areas:

1—Columns 1-6.

Contain line sequence numbers (usage is optional with some COBOL compilers but WE will use them for "Good" programming). They must be in ascending order.

2—Column 7.

A "-" is used for a line continuation character (we will seldom use this). The "*" is used to tell the compiler that this is a comment line (like REM in BASIC). A "/" is used for page eject (like a Form Feed for Printer).

3—Columns 8-11.

Called AREA A. Reserved for: division headers, section names, paragraph names, level indicators, or certain level numbers.

4—Columns 12-72.

Called AREA B. It is used for all COBOL code that cannot be in AREA A.

5—Columns 73-80.

Used for program identification. We will not use these columns with COBOL-80.

COBOL Structure

(**NOTE:** The line of numbers shown below are furnished to show the Columns the various items fall into. They are not part of the COBOL program.)

```
         0         1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890123456789012
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.     YOUR PROGRAM NAME.
000030 AUTHOR.         YOUR NAME.
000040 INSTALLATION.   YOUR COMPANY NAME.
000050 DATE-WRITTEN.   TODAYS DATE.
000060 DATE-COMPILE.   WHEN YOU DO IT.
000070 SECURITY.       NONE.
000080*
(Some space for a documentation paragraph(s))
000100 ENVIRONMENT DIVISION.
(Space to show Hardware used with program)
000150 CONFIGURATION SECTION.
(Specify computer used with program)
000200 INPUT-OUTPUT SECTION.
(Describe Data Files.  Each file is identified by a User-Defined name
 and ASSIGNED to a I/O device.)
000300 DATA DIVISION.
 (The Data to be processed)
000350 FILE SECTION.
(File description for each file used in the program)
       LEVEL NUMBERS:
            01     Record description entry
            02-49  Data items (fields)
       RECORD is a collection of fields.  Sub-fields further define
       fields.  Level numbers of sub-fields must have a higher number
       than the level number of the field it is describing.
000400 WORKING-STORAGE SECTION.
(Additional data fields you need to use in your program, other than the
ones appearing in the above I/O Records, are defined in this section.)
000500 PROCEDURE DIVISION.
(Paragraph names in this division are User-Defined.  Within each
paragraph a group of COBOL Statements that will tell the compiler how to
do the program.)
```

The above Structure is a brief framework of a COBOL program. It shows the order that DIVISIONS and SECTIONS must be in and the use of the column AREAS. Note that we will start each coding line beginning with a six digit number (Columns 1-6), followed by a space in Column 7 and when the line contains a DIVISION or SECTION, it starts in AREA A (Columns 8-11); otherwise, words start in AREA B (Columns 12-72). You MUST know this order and the use of AREAS!

If you have done your "HOMEWORK" and have read your COBOL-80 Manual, you will know that some of the points covered above are optional. However, if we are going to write "Good" programs that are portable to other COBOL systems, we will make the extra effort to follow the way we are presenting in "COBOL Corner". Think of yourself as an employee of "COBOL Corner" and you have been told what Programming Standard Practice will be used at your place of employment! This will become more important as we increase the complexity of the sample COBOL Programs.

Keying (Writing Code At Your Terminal) The Source Program

Using Disk A in Drive A that has your Editor, key-in your first COBOL program exactly like it is shown in the LISTING.

```
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.    COBOL1.
000030 AUTHOR.        YOUR NAME.
000040 INSTALLATION.  COBOL CORNER.
000050 DATE-WRITTEN.  WINTER, 1983.
000060 DATE-COMPILED. SPRING, 1983.
000070 SECURITY.      NONE.
000080*
000090******************************************************************
000100*                                                                *
000110*  THIS PROGRAM WILL PRINT YOUR NAME ON THE PRINTER AS DEFINED   *
000120*  IN THE WORKING-STORAGE SECTION.                               *
000130*                                                                *
000140******************************************************************
000150*
000160 ENVIRONMENT DIVISION.
000170*
000180 SOURCE-COMPUTER.  HEATH H-8.
000190 OBJECT-COMPUTER.  HEATH H-8.
000200*
000210 INPUT-OUTPUT SECTION.
000220*
000230 FILE-CONTROL.
000240     SELECT PRINT-FILE
000250         ASSIGN TO PRINTER.
000260*
000270 DATA DIVISION.
000280*
000290 FILE SECTION.
000300*
000310 FD  PRINT-FILE
000320     RECORD CONTAINS 132 CHARACTERS
000330     LABEL RECORDS ARE OMITTED.
000340*
000350 01  PRINT-RECORD                        PIC X(132).
000360*
000370 WORKING-STORAGE SECTION.
000380*
000390 01  PRINT-LINE.
000400     05  FILLER                          PIC X(05).
000410     05  FILLER                          PIC X(20)
000420                                     VALUE "YOUR NAME
000430     05  FILLER                          PIC X(107).
000440*
000450 PROCEDURE DIVISION.
000460*
000470 MAIN-PROCEDURE.
000480*
000490     OPEN OUTPUT PRINT-FILE.
000500     WRITE PRINT-RECORD FROM PRINT-LINE.
000510     CLOSE PRINT-FILE.
000520     STOP RUN.
```

Remember to use the correct column AREAS, watch for periods & hyphens in the correct places and be sure to spell your reserved words correctly. In future articles we will explain what this CODE is doing; but for now, take it at face value. Please, NEVER forget that "STOP RUN" must be in every COBOL program. If you forget that, you can get some strange results that can be very difficult to debug! Also, watch for "O" (the letter) vs. "0" (the number).

This simple program provides Terminal Practice (keying), Editor Practice, and COBOL Structure Practice. Note the line-up of the code in columns! The Program-ID, Author, etc. should have the first letter start in Column 8 and the descriptions should start directly below the "D" in DIVISION. The word SELECT should start in Column 12 (AREA B) and the word ASSIGN should be indented four (4) spaces to the right of the "S" in SELECT under the "C" for readability. We will always use a separate code line for each COBOL Verb in our Structured COBOL Programs, again for readability.

FD is a Reserved Word in COBOL meaning file description. It always starts in Column 8 (AREA A). Note the period does not come until the end of the clause (Sentence). Also, note that in the WORKING-STORAGE SECTION, the second FILLER clause has the Reserved Word VALUE after the PIC X(20) (we will go into detail about the PIC clause in later articles). We had to go to another line for the VALUE item as we did not have enough AREA B (Columns 12-72) space for the literal "YOUR NAME (fill with spaces until your name plus the spaces total 20 characters as specified) " on the coding line. We started the next coding line with a line number and placed the continuation VALUE just to the left of PIC and, of course, no period until we finished the FILLER code line after the literal. We ASSIGNED the output to Printer providing for a 132 column print-out. Be sure to have your printer set up for this when you Execute this program!

Save Source File

Using whatever SAVE command that your Editor uses, save your Source Code on Disk A with the User-Defined Name A:COBOL1.COB.

Compile and Execute Your Source Code

Put Disk B in Drive B and Type B: and Return. Now, Type your command line:

```
B> COBOL A:COBOL1,TTY:=A:COBOL1 and Return
```

If you have not made any ERRORS, you should get the following message at the end of your Listing on the CRT:

```
"NO ERRORS OR WARNINGS"
```

If you get any error messages other than the one above, Type the following command line:

```
B> COBOL A:COBOL1,LST:=A:COBOL1 and Return
```

This will provide you with a print-out LISTING of your Source Code with the ERRORS detailed by compiler provided line numbers. Referring to these messages and the corresponding line numbers should enable you to find your errors. Correct your errors on the listing in red pencil.

Now, Type A: and Return. Using your Editor, make your corrections to the A:COBOL1.COB file and re-save using the same file name. Now, you are ready to compile again as described before. Do this until you get a "Clean Run" (No Errors)!

When you have no compiler errors and you have the file A:COBOL1.REL on Disk A, put Disk C in Drive B. Type B: and Return. Next, Type the following command line:

```
B> L80 A:COBOL1/G/E
```

Your printer should print out YOUR NAME indented five (5) spaces!

If it does not, go back and review the Coding for errors by comparing with the article's program listing. You should be able to find the cause of any errors.

Closing

In the next "COBOL Corner" we are going to look at Data Concepts, COBOL User-Defined Words & Names, and start our Sample Program #1 which will demonstrate "COBOL READ and PRINT". To work along with future "COBOL Corner" Sample Programs, you will need the HUG COBOL Corner Disk-I. Please obtain it before the next article. For your "HOMEWORK", please study the READ Statement and the WRITE Statement in your COBOL-80 Reference Manual.

To obtain

HUG COBOL Corner Disk I

Send a blank disk and $5.00 to:

Heath Users' Group
Hilltop Road
St. Joseph, MI 49085
Attn: Nancy Strunk

Cassette Modem Software

Pat Swayne
Software Engineer

Occasionally we get calls here at HUG from members who are still using cassette tape systems, and want a modem program for their system. Since many of these users are not familiar with assembly language, I attempted to write a modem program in the Cassette version of Extended Benton Harbor BASIC. Listing 1 is the result of my efforts. As it turned out, Benton Harbor BASIC is not fast enough to do the job, so the actual modem software in my program is a machine language program that is POKEd into memory and then called with the USR function. The DATA statements make up the machine language portion of the program, and are produced by the assembly language program in Listing 2. You do not need to enter the program in Listing 2 to use the modem program.

This program is a very basic one that does nothing more than allow you to communicate over a modem. It does not have provision for saving data, sending saved data, or sending output to a printer. As listed, it will work on an H/Z-89 or an H8 with both the console and modem on an H8-4 card. The modem must be connected to port 330Q (216 decimal), to the DTE connector. Modifications are included in REM statements and listings of alternate DATA statements for other configurations. Be sure you use the correct USER address in line 80. Addresses for various versions of Benton Harbot BASIC are in line 90. You need at least 16K of RAM to use this program, and your BASIC should not be configured with the high address lowered.

Listing 1

```
10 REM CASSETTE BASIC MODEM PROGRAM
20 REM BY P. SWAYNE, HUG  21-SEP-83
30 REM
40 PRINT "ENTERING COMMUNICATION LOOP"
50 PRINT "TYPE CONTROL-E TO RETURN TO BASIC"
60 REM
70 REM FIX USR ADDRESS
80 X=19022:REM USER FUNCTION ADDRESS
90 REM BASIC 10.02=17990, 10.05=18353, 10.06=19022
100 POKE X, (22528 AND 255):POKE X+1,INT(22528/256)
110 REM
120 REM INSERT MODEM PROGRAM INTO MEMORY
130 REM
140 FOR I=0 TO 44
150 READ A:POKE 22528+I,A:NEXT I
160 REM
170 REM PROGRAM THE PORTS
180 REM
190 PORT OUT 216,300:REM MODEM PORT (330Q, 300 BAUD)
195 REM   IF YOUR MODEM IS ON AN H8-5 CARD AT PORT 374,
196 REM   REPLACE LINE 190 WITH OUT 252,300
200 OUT 217,0:OUT 233,0:REM KILL INTERRUPTS
205 REM   USE OUT 251,21 FOR THE ABOVE LINE IF YOUR
206 REM   CONSOLE IS ON AN H8-5 CARD
210 REM
220 A=USR(0):REM EXECUTE THE MODEM PROGRAM
230 REM
240 REM DATA STATMENTS CONTAINING THE MODEM PROGRAM
245 REM   USE OTHER DATA STATEMENTS FOR THE H8-5 CARD!
250 REM
260 DATA 219,221,230,1,194,31,88,219,237,230
270 DATA 1,202,0,88,219,221,230,32,202,14
280 DATA 88,219,232,254,5,200,211,216,195,0
290 DATA 88,219,237,230,32,202,31,88,219,216
300 DATA 211,232,195,0,88
310 REM
320 REM RESTORE CONSOLE PORT
330 REM
340 OUT 233,1
345 REM   USE OUT 251,23 FOR THE ABOVE LINE IF YOUR
346 REM   CONSOLE IS ON AN H8-5 CARD
```

The data statements below are for an H8 with the console on an H8-5 card, and the modem on an H8-4 card at port 330Q.

```
260 DATA 219,221,230,1,194,31,88,219,251,230
270 DATA 2,202,0,88,219,221,230,32,202,14
280 DATA 88,219,250,254,5,200,211,216,195,0
290 DATA 88,219,251,230,1,202,31,88,219,216
300 DATA 211,250,195,0,88,0
```

The data statements below are for an H8 with the console on an H8-5 card, and the modem on another H8-5 card at port 374Q.

```
260 DATA 219,253,230,2,194,31,88,219,251,230
270 DATA 2,202,0,88,219,253,230,1,202,14
280 DATA 88,219,250,254,5,200,211,252,195,0
290 DATA 88,219,251,230,1,202,31,88,219,252
300 DATA 211,250,195,0,88,0
```

Listing 2

```
*         MINI MODEM PROGRAM
*         FOR USE AS A USR PROGRAM IN
*         CASSETTE BH BASIC.

CSPORT   EQU     355Q                 CONSOLE STATUS PORT
CDPORT   EQU     350Q                 CONSOLE DATA PORT
CINBIT   EQU     1                    CONSOLE INPUT STATUS BIT
COUTBIT  EQU     40Q                  CONSOLE OUTPUT STATUS BIT
MSPORT   EQU     335Q                 MODEM STATUS PORT
MDPORT   EQU     330Q                 MODEM DATA PORT
MINBIT   EQU     1                    MODEM INPUT STATUS BIT
MOUTBIT  EQU     40Q                  MODEM OUTPUT STATUS BIT

         ORG     130000A
LOOP     IN      MSPORT               TEST FOR MODEM INPUT
         ANI     MINBIT
         JNZ     LOOP1                GOT MODEM INPUT
         IN      CSPORT               TEST FOR KEYBOARD INPUT
         ANI     CINBIT
         JZ      LOOP                 NO KEYBOARD INPUT
LOOP0    IN      MSPORT               TEST FOR MODEM XMIT READY
         ANI     MOUTBIT
         JZ      LOOP0                LOOP UNTIL READY
         IN      CDPORT               GET KEYBOARD INPUT
         CPI     5                    CONTROL-E?
         RZ                           IF SO, RETURN TO BASIC
         OUT     MDPORT               SEND IT TO MODEM
         JMP     LOOP                 LOOP FOREVER
LOOP1    IN      CSPORT               TEST FOR SCREEN XMIT READY
         ANI     COUTBIT
         JZ      LOOP1                LOOP UNTIL READY
         IN      MDPORT               GET MODEM INPUT
         OUT     CDPORT               PUT IT ON THE SCREEN
         JMP     LOOP
         END     LOOP
```

Z MAG

"The Software Library" from **Husker Systems of Nebraska**

Z-MAG — A WAY TO BUILD YOUR PROGRAM LIBRARY WITH AMAZING EASE...

If you're like most personal computing system owners, you get bored quickly with your software purchases....Three, four, maybe half a dozen uses and the diskette is filed in your bin, seldom to be used again. This is especially true of games. You probably run across these products now and then and think *"Why did I spend $50 (or $100 or $200) on that? I never use it."*

Husker Systems of Nebraska, Inc. now offers an answer to your frustration. It's **Z-MAG**, a new subscription service which provides you with a variety of new software 8 times, over each annual subscription period.

VARIETY...

Each **Z-MAG** issue contains up to 15 programs that cover the wide range of your application interests: from games to self-tutoring education to programming and systems tools to personal and home help programs.

ECONOMY...

The entire year's subscription costs no more than many of your single program purchases. Since there are 8 'issues' per subscription year and up to 15 programs per issue, you stand to gain as many as 120 programs. At the full retail price, that's only $2.00 per program...and it's less than $1.25 per program at our Special Introductory Price!!!

Why worry about an investment in programs you only occasionally use? With **Z-MAG**, your per program cost is so low it removes the worry.

EASE OF USE...

Z-MAG routines are useful because they cover a wide variety of interests. Also, many of them are offered in MBASIC (or ZBASIC) so you can modify them or integrate them with other systems you have.

...AND A PLEASANT SURPRISE

All of us want to get something of value in the mail rather than just bills and trash mail. Wouldn't you enjoy finding a diskette with valuable software on it in your mail box every 6 weeks...without really anticipating it...and knowing that no future payments are required?

HERE'S THE DEAL...

YOU GET...

8 issues, each mailed to you in approximately six-week intervals.

10 to 15 programs per issue.

A user manual with each issue which not only explains each program but also offers helpful hints.

Up to 8 calls to us every subscription year for software or hardware support (not limited to Z-MAG products).

Although the regular retail price per subscription is $240 ($30 per issue), you get our Special Introductory Price of $147 (valid through December 31, 1983).

If at any time during your subscription year, you wish to cancel your subscription, full refund of all remaining issues is granted without question.

Start enlarging your program library **TODAY!** Clip and send in the handy order blank provided below, or better yet, call Husker Systems of Nebraska, Inc. (402) 558-5702, 4517 North 61st St., Omaha, NE 68104. **DON'T WAIT!!!**

Fine Zenith/Heath Software
VISA and MasterCard Accepted

HUSKER SYSTEMS OF NEBRASKA INC.

YES, Please start my subscription to the **Z-MAG** Software Library for the Special Introductory Price of $147 for 8 issues which is a big savings of $93.00 over the individual cost.

Check the Z-MAG Service you wish: ____**H/Z89 (CP/M)** ____**H/Z100 (CP/M)** ____**H/Z100 (ZDOS)**

Check One:______Payment Enclosed for $147.00

________I would like the Easy Payment Option — Enclosed is $75.00. Please bill me in 60 days for the Balance of $72.00.

________Please Charge My Credit Card ____VISA ____MasterCard

Credit Card No.____Exp. Date____________

Name:____

Address:____

City: ______________________________ State:______________ Zip:______________

□ **Check Here for Free Catalog.** **Mail Coupon Today To: Husker Systems, 4517 No. 61st St., Omaha, NE 68104**

Automatic Title Program For BASIC Programs

Ricky S. LeBlanc
Rte. 3, Box 498
Breaux Bridge, LA 70517

Recently I undertook the task of sorting out that large stack of fanfold paper that has been accumulating for the last year. As I went through the stack, I thought to myself "I am sure glad that I took the time to put a title on each one of these listings". It sure was handy just to quickly glance at the top of each listing to see just what program the listing was and the date I had written it.

I can remember when I was first struggling through learning BASIC programming that putting a title on a listing was a neat idea. I soon found myself spending sometimes 30 minutes centering the title and getting everything just right. Then it occurred to me, why take all this valuable programming time to develop a title for my program. I should let "Isidore" do it. (Isidore is short for Isidore Computeaux, my H-89 Computer. Being from south Louisiana and Cajun Country, my computer had to have a Cajun name.) So, Isidore and I set out to write an automatic Title program, which I have included here.

I will give a line by line explanation of what I attempted to accomplish for the benefit of the novice computer owner just getting started with BASIC. I should mention at this time that it is important to number this program in increments of one (1) due to the fact that the program self erases itself from your main program. The program lines 10 through 110 are a sample of the generated title and was made by the program itself. Start entering the program with line 111.

```
10 '         *********************************************************
20 '         *********************************************************
30 '         **********  AUTOMATIC TITLE PROGRAM  ver 1.04  **********
40 '         **********                                     **********
50 '         **********        Ricky Stephen LeBlanc        **********
60 '         **********            Rte 3 Box 498            **********
70 '         **********      Breaux Bridge, LA   70517      **********
80 '         **********                                     **********
90 '         **********           11 AUGUST 1983            **********
100 '        *********************************************************
110 '        *********************************************************
111 OPTION BASE 1:DIM T$(11),T(11):DEFINT C,I,K,X
112 PRINT CHR$(27);"E";"PLEASE ENTER THE FOLLOWING INFORMATION :":PRINT
113 LINE INPUT"THE NAME OF PROGRAM........> ";T$(3)
114 LINE INPUT"THE CREATION DATE..........> ";T$(9)
115 LINE INPUT"THE VERSION NUMBER.........> ";T1$
116 READ T$(5),T$(6),T$(7)
117 S=LEN(T$(3)):IF INT(S/2)=S/2 THEN S=1 ELSE S=2
118 T$(3)=T$(3)+SPACE$(S)+"ver "+T1$
119 FOR I=3 TO 9
120 T(I)=LEN(T$(I)):IF T(I)>X THEN X=T(I)
121 NEXT I
122 T$(1)=STRING$(X+24,"*"):T$(2)=T$(1):T$(10)=T$(1):T$(11)=T$(1)
123 A$=STRING$(10,"*"):T$(4)=SPACE$(X):T$(8)=T$(4):T(4)=X:T(8)=X:C=(50-X)/2
124 FOR I=3 TO 9
125 IF X=T(I) THEN T$(I)=A$+"  "+T$(I)+"  "+A$:GOTO 127
126 S=INT(((X-T(I))/2))+2:T$(I)=A$+SPACE$(S)+T$(I)+SPACE$(X-T(I)-S+4)+A$
127 NEXT I
128 FOR I=1 TO 9:T$(I)=STR$(I*10)+" '"+STRING$(C+1," ")+T$(I):NEXT I
129 FOR I=10 TO 11:T$(I)=STR$(I*10)+" '"+STRING$(C," ")+T$(I):NEXT I
130 OPEN "O",#1,"TMPTITLE.BAS"
131 FOR I=1 TO 11:PRINT T$(I):PRINT #1,T$(I):NEXT I
132 FOR K=111 TO 133:PRINT #1,K:NEXT K:CLOSE:MERGE"TMPTITLE.BAS":END
133 DATA "Ricky Stephen LeBlanc","Rte 3 Box 498","Breaux Bridge, LA   70517"
```

LINE 111. Dimensions the arrays T$() and T() with option base I which provides the eleven lines that form the title and T() as the length of each line. C, I, K, and X are defined as integer numbers to speed up things.

LINE 112. Clears the screen and prints a prompt to enter information.

LINE 113. Prompts the user to enter the name of the title which can be any ASCII string up to 30 characters. The title is assigned to T$(3).

LINE 114. Prompts the user to enter the creation date of the program or the date that the title is added. The creation date is assigned to T$(9).

LINE 115. Prompts the user to enter the version number of the program. I use the x.xx format as the version number, but any kind of format can be used as long as the number of characters are taken into account when centering the title. The version number is assigned to variable T$ to be combined later with T$(3) to complete the title.

LINE 116. Reads in the DATA statement from line 133 which contains the authors name and address, assigning them to T$(5), T$(6), and T$(7).

LINE 117. Checks to see if the title in T$(3) has an even or odd number of characters and sets S to 1 if it is even and S=2 if it is odd. The reason for this is there are 21 characters in my full name (including spaces), thus S adds the required spaces to the title to center it over the name. If your name is even, then change line 117 to read: 117 S=LEN(T$(3)):IF INT(S/2)=S/2 THEN S=2 ELSE S=1 : I know this is getting picky, but I like things to look neat and even. On the same token, when typing in line 133, add spaces in your address to make it come out even or odd to match your name.

LINE 118. Concatenates T$(3) to equal T$(3) along with the spaces determined in line 117 with "ver" and the version number T$. This is set up for the x.xx version format, other formats will need to be adjusted by line 117 and the variable S.

LINE 119-121. Sets up a FOR/NEXT loop to check for length of the longest string and set that value to X.

LINE 122. Sets T$(1) to 24+X asterisks (*) to form the border of the title, then also sets T$(2), T$(10), and T$(11) to the same value.

LINE 123. Gives A$ the value of 10 asterisks, then sets T$(4) and T$(8) to a value of spaces equal to the longest string. C is set to the number of spaces needed to center the title block allowing for the 20 asterisks, 4 spaces and 4 characters for the line number and remark (') notation. Finally T(4) and T(8) are set to the same value as X.

LINES 124-127. Set up a FOR/NEXT loop. Line 126 adds two spaces before and after the longest string and puts 10 asterisks before and after the spaces. Four spaces are added to T$(4) and T$(8) as well as the 10 asterisks to even it out to the longest string.

LINE 127. Adjusts the number of spaces before and after the shortest strings by first finding half the difference of X and T(I) and adding the two spaces to that value. T$(I) is then made up of A$, the correct spaces, T$(I), then the remainder of spaces to center it, then A$ again aligning it with the other strings.

LINE 128. Prints the numbers 10 to 90, the remark symbol, and the necessary spaces to center the title adding 1 to C to compensate for the three digit numbers 100 and 110.

LINE 129. Does the same for the line numbers 100 and 110.

LINE 130. Opens the file "TMPTILE.BAS" as a sequential file.

LINE 131. Prints T$(1) to T$(11) to the screen and to the disk. It is important to use the PRINT#1 statement here because WRITE#1 inserts quotation marks as delimiters on the disk file causing a DIRECT STATEMENT IN FILE error when trying to run the program.

LINE 132. Prints out the line numbers 111-133 to the file "TMPTITLE.BAS". By doing this, the program self-erases itself from the main program due to the fact that entering a BASIC line number and no statement erases any statement already there. The file is closed, then merged back to the main program, thus erasing the title program leaving only the title block.

LINE 133. Is a DATA statement containing your name and address.

To use "TITLE.BAS", first type in lines 111-133, putting your own name and address. Next, save the BASIC file in ASCII form by typing the following:

```
SAVE "TITLE.BAS",A
```

Then "TITLE.BAS" is ready to use. Load the program you wish to make a title for, then renumber the program starting at 200 (I use 200 because it is a round figure, you could use any number larger than 133). Type the command: RENUM,200. Then type: MERGE"TITLE.BAS". When the BASIC prompt is displayed again type: RUN Enter the information as the program calls for it. After the version number is entered, the title will appear before your eyes on the screen as well as being written to the disk and into the program. NOTE: Don't forget to resave your program with the new title!!!

I would like to interject some pointers here if I may. Whenever I develop a new program, I call it ver. 1.01, and each day I work on it I increment the decimal portion by .01. That way I know that ver 1.15 is the 15th day I have worked on that particular program. This makes it easy to sort through that tall stack of program listings in that old Heathkit box stashed away in the corner "because I may need that someday". Thumbing through those listing is so much easier with a title standing out to greet you. Another thing I do when developing a program is to save each days work on disk with a title that reflects the version number. By doing this I always have the last 3 or 4 versions on disk in the event that I don't like the way things are developing and want to go back to a previous version to take another approach.

I hope that I might have helped someone to understand this strange new world of computers a bit better; after all, we are all here to help each other. I wish you much enjoyment out of this program and as Isidore and I would say down here in South Louisiana "Lessier, Les Bon Temps Roulie" (Let the good times roll).

About the Author:

Ricky Stephen LeBlanc is a 5th generation native-born Louisianian and a 15th generation Acadian descendant. He resides in Breaux Bridge, La. Ricky is employed as a microwave telecommunications technician on the Western Gulf Microwave System (WGMS). He helps maintain this system which brings telephone, data, and supervisory control to some 60-70 offshore production platforms in the Gulf of Mexico. He's been in electronics since 1976 when he went to the U.S. Army Signal school in Multichannel Communications. Ricky has a BS degree in Applied Science from the University of Southwestern Louisiana, with major areas of concentration in Architecture and Industrial Technology. He built his H-88 as part of the Master Course in Microcomputers offered by National Technical Schools and later upgraded it to an H-89. His hobbies include genealogy, photography, and woodworking to include presently building his own house (which he someday hopes to computerize). He programs mostly in MBASIC/CP/M, and is currently taking the plunge into C language programming.

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

Those Special Function Keys

I've received a lot of questions on the special function keys. My previous article in the May, 1982 Issue 28 of REMark addressed their use in Benton Harbor BASIC. However, it appears that it's time to look at them again. This time we'll work in MBASIC. We'll look at some more sophisticated means of decoding them, and then we'll add twelve more keys by entering the alternate keypad mode.

Many times in our programming, it would be nice to touch a single key and perform a special function. Games, for instance, can shoot at aliens, put up defense shields, etc. Programs which accept inputs from all the normal keys, such as Morse code senders, need a unique input to perform special functions. Our word game used the red key to bypass the guessing part of the game.

The first thing we must find out is what makes those eight keys labeled F1 through F5, and RED, BLUE, and WHITE look any different to the computer than the other keys. To do this, we have to understand how the keyboard works. When any key is struck, it closes a set of contacts. Circuitry in the Terminal Logic Board senses this closure and encodes it as an ASCII character. From the Terminal Logic Board it is sent to the computer at a speed of transmission called 9600 baud.

A look at Section 12 of our Operating Manual shows us what all the ASCII codes are. There are 128 different codes with decimal values of 0 to 127. A little math will show us that's 2 to the 7th power or 7 bits of our 8-bit binary word. The eighth bit is called a parity bit. We'll discuss that in a later article. The only disturbing fact is that all 128 codes have a definition, and none of them is one of the function keys. We'll have to look farther in Section 12. You'll find a table of the Function Keys and Heath Escape Sequences. So the secret is unlocked. When one of the function keys is pressed, the Terminal Logic Board sends the computer an ESCape (ASCII character number 27) followed by a capital letter, P through W, depending on the key pressed as follows:

KEY	CHARACTERS SENT
Blue	ESC P
Red	ESC Q
White	ESC R
F1	ESC S
F2	ESC T
F3	ESC U
F4	ESC V
F5	ESC W

To determine which key has been pressed, all we have to do is get the characters, one at a time, and test them for the characters those keys send.

There are two functions in MBASIC which will take an input from the terminal one character at a time. They are INKEY$ and INPUT$. For now we'll use INPUT$. With this function, you must specify how many characters you want at a time. When that number is reached, the input is completed and your program continues execution. We'll use "1" so we can test each character as we get it. When we find an ESC we'll test for the second character to see which key was struck. From the Operating Manual we find that ESC is ASCII character 27. That's what we'll test for. It looks like this.

```
10 PRINT "PRESS ONE OF THE FUNCTION KEYS"    'MESSAGE TO SCREEN
20 A$=INPUT$(1)             'GET ONE CHARACTER
30 IF A$=CHR$(27) GOTO 60     'TEST FOR ESC
40  PRINT  "YOU  DIDN'T PRESS A FUNCTION KEY-TRY  AGAIN"   'ERROR
MESSAGE
50 GOTO 10     'GO BACK TO TRY AGAIN
60 REM ESC WAS FOUND SO WE'LL INPUT THE SECOND CHARACTER AND TEST
IT
70 B$=INPUT$(1)     'GET THE CHARACTER
80 IF B$="P" THEN PRINT "BLUE KEY WAS STRUCK":GOTO 160
90 IF B$="Q" THEN PRINT "RED KEY WAS STRUCK":GOTO 160
100 IF B$="R" THEN PRINT "WHITE KEY WAS STRUCK":GOTO 160
110 IF B$="S" THEN PRINT "F1 KEY WAS STRUCK":GOTO 160
120 IF B$="T" THEN PRINT "F2 KEY WAS STRUCK":GOTO 160
130 IF B$="U" THEN PRINT "F3 KEY WAS STRUCK":GOTO 160
140 IF B$="V" THEN PRINT "F4 KEY WAS STRUCK":GOTO 160
150 IF B$="W" THEN PRINT "F5 KEY WAS STRUCK":GOTO 160
160 END
```

When you run this program, striking a function key will generate a message. Rather than a print statement, we could have used a GOTO, GOSUB, or any function or subroutine we wanted. The important thing is that you understand the method of decoding these keys.

It would be nice if we could find a shorter method of decoding. A separate test for each key requires a lot of typing and is kludgy. If the F5 key is struck, eight tests have to be made to find it. Fortunately, the designers at Heath used consecutive letters in the assigning of the function keys. This means they also have consecutive ASCII values and we can use the MBASIC ASC function to find those values. This function returns the value of each character. For the "P" it will return the number 80 and each character will increase by 1 until we get to "W" at 87. Thus, if we use the program line

B=ASC(B$)

we should get a value for B in the range of 80 to 87. In the MBASIC Manual we find the command ON.....GOTO. It works on the result of a calculation. Based on that number, it can send program execution to any of several program lines. Now, it prefers to work with a series of numbers beginning at 1 so we'll have to subtract 79 from the ASCII values we just looked at. Now replace lines 80 to 160 in the program above with the following.

```
80 B=ASC(B$)-79          'GET VALUE OF SECOND CHARACTER
90 IF B<1 OR B>8 THEN PRINT "YOU CAN'T FOOL ME.   THAT WASN'T ONE
OF  THE  FUNCTION  KEYS":GOTO 10      'SECOND  INPUT  NOT  WITHIN
LIMITS
100 ON B GOTO 110,120,130,140,150,160,170,180
110 PRINT "BLUE KEY WAS STRUCK":GOTO 10
120 PRINT "RED KEY WAS STRUCK":GOTO 10
130 PRINT "WHITE KEY WAS STRUCK":GOTO 10
140 PRINT "F1 KEY WAS STRUCK":GOTO 10
150 PRINT "F2 KEY WAS STRUCK":GOTO 10
160 PRINT "F3 KEY WAS STRUCK":GOTO 10
170 PRINT "F4 KEY WAS STRUCK":GOTO 10
180 PRINT "THAT'S ALL FOLKS---BYE":END
```

Now run this program. You'll have to type the F5 key to get it to end. Try pressing the ESC key and a letter P through W. Now press ESC and any other letter. That's what line 90 is looking for. If you anticipate most possible input errors, you can make your programs pretty smart.

So much for the eight function keys. Did you know there are 12 more on your terminal? These Heath machines are packed with more features than there are ants at a picnic. Let's go back to Section 12 of the Operating Manual again. Look for a table showing the Alternate Keypad Mode. That's one of the instructions we included in CONTROL2.BAS last month, and we'll use it now.

While many of the keys on the keyboard are capable of sending two different characters by striking them with or without the shift key being held down (Capital vs. Lower Case letters is one example), the keys of the keypad can send three different characters each. To get the third set, which are undefined functions (that means we must put their meaning in the program just as we do for the function keys), we must enter the alternate keypad mode. Just like the eight keys we used at the beginning of the article, we'll run tests to separate them. First, we must enter the alternate keypad mode. We do this by sending the Terminal "ESC =". Using our module CONTROL2.BAS, we can PRINT KA$.

Continuing with our manual, we find the keys send the following characters when in the Alternate Mode.

KEY	CHARACTERS SENT
. (PERIOD)	ESC ? n
0	ESC ? p
1	ESC ? q
2	ESC ? r
3	ESC ? s
4	ESC ? t
5	ESC ? u
6	ESC ? v
7	ESC ? w
8	ESC ? x
9	ESC ? y
ENTER	ESC ? M

Having completed the first exercise, it's easy to see that we just:

1) Test for ESC
2) Test for ?
3) Decode the last character.

Right? Weelll......, yes and no. This is one of the reasons I twitch occasionally. I even called Heath in Benton Harbor to verify my findings. In the third and fourth paragraph of this article, I pointed out that characters generated by the Terminal Logic Board are dumped to the computer at 9600 baud. There's no buffer here, so if the computer isn't ready and a string of characters is sent, some may be lost. At normal typing speeds this doesn't present a problem as a machine running at 2MHz does several hundred thousand things a second and humans are verrrry slow. The Terminal Logic Board, however, is just as fast as the computer. So the "?" may be lost. Those of you with 3MHz and 4MHz modifications may not have a problem. How can you tell? The following program will set the keypad to its Alternate Mode, take inputs, and print all characters except ESC on our CRT. Then we'll know what was received by the computer.

```
10 E$=CHR$(27)       'ESCAPE
20 KA$=E$+"="        'KEYPAD ALTERNATE MODE
30 KX$=E$+">"        'KEYPAD EXIT ALTERNATE MODE
40 PRINT KA$         'SET KEYPAD TO ALTERNATE MODE
50 PRINT "INPUT A KEYPAD CHARACTER"     'MESSAGE ON SCREEN
60 A$=INPUT$(1)      'GET 1 CHARACTER
70 IF A$=CHR$(27) GOTO 90               'ESCAPE RECEIVED
80 PRINT "ESC NOR RECEIVED":GOTO 50     'WRONG KEY STRUCK
90 B$=INPUT$(1)      'GET SECOND CHARACTER
100 PRINT "B$ WAS RECEIVED AS ";B$      'TELL WHAT WAS RECEIVED
110 IF B$<>"?" GOTO 140  'MUST HAVE MISSED "?".  GOTO END
120 C$=INPUT$(1)     'WE GOT THE "?" SO GET THIRD CHARACTER
130 PRINT "C$ WAS RECEIVED AS A ";C$    'TELL WHAT WAS RECEIVED
140 PRINT KX$        'SET KEYPAD BACK TO NORMAL
150 END
```

Having run that little 15 line program, you now know whether your computer receives the question mark. If it does not, there's nothing wrong. It's just a matter of timing and you will have to delete the input and test for the question mark on lines 2020-2080 of the following program.

The program we'll use to demonstrate decoding of the keypad in its Alternate Mode will use just the keys 0 through 9. Each key 1 through 9 will print a 6x6 block of graphic X's on the screen in the same relative position as the key in the keypad. The 0 key erases the display. You must press the RED key to end program execution. That's decoded on line 2010.

While writing this program, I followed certain basic steps each of us must use. They are:

1) Make a flow chart of what the program is to do.
2) Do the work in modules with standards per last month's column.
3) Test program to see that it works.
4) Write additional lines to allow for every possible incorrect input.

Some of the special features you'll find are:

Lines 450,460	User Functions to save programming
Lines 540,550	25th Line used
Lines 4000-4030	Loop used to print 6 identical lines
Line 460	STRING$ used to repeat the same character 6 times

Notice the end routine resets each special feature on the terminal to normal before ending program execution. Some of the features were discussed last month and all are explained in Sections 6 & 12 of your Operating Manual or in your MBASIC Manual. Type in the following program and MERGE CONTROL2.BAS from last month's article. Then save it as any name you like and run it.

```
450   DEF  FNCA$(Y,X)=CA$+CHR$(31+Y)+CHR$(31+X)  'DEFINE   CURSOR
ADDRESS FUNCTION
460 DEF FNGL$(X,Y,Z)=CA$+CHR$(31+Y+Z)+CHR$(31+X)+STRING$(6,119)
     'DEFINE GRAPHIC LINE FUNCTION
500  PRINT  CH$  'FORCE CURSOR HOME BEFORE ENTERING  HOLD  SCREEN
MODE
510 PRINT ED$  'ERASE THE DISPLAY
520 PRINT SH$  'ENTER HOLD SCREEN MODE
```

```
530 PRINT CX$  'TURN CURSOR OFF
540 PRINT GM$  'ENTER GRAPHICS MODE
550 PRINT LX$  'ENABLE 25TH LINE
560 PRINT KA$  'ENTER ALTERNATE KEYPAD MODE
570  PRINT  FNCA$(25,1)"PRESS RED KEY TO QUIT";CH$   'MESSAGE  ON
25TH LINE
1000  PRINT FNCA$(1,10)"PRESS A KEY 0-9 ON THE KEYPAD"   'MESSAGE
ON LINE 1
1010 A$=INPUT$(1)   'GET THE FIRST CHARACTER
1020 IF A$=CHR$(27) GOTO 2000       'GO TO DECODE ROUTINE IF ESC
1030  PRINT  FNCA$(23,10)"YOU  DIDN'T  PRESS A  KEY  0-9  ON  THE
KEYPAD";CH$    'ERROR MESSAGE ON LINE 23
1040 FOR X=1 TO 500:NEXT X    'DELAY SO MESSAGE CAN BE READ
1050 PRINT FNCA$(1,1);EL$     'ERASE LINE 1
1060 PRINT FNCA$(23,1);EL$;CH$     'ERASE LINE 23
1070 GOTO 1000
2000 A$=INPUT$(1)   'GET SECOND CHARACTER
2010 IF A$="Q" GOTO 5000       'END ROUTINE (RED KEY)
2020 IF A$="?" GOTO 2080 'TEST FOR "?"
2030 PRINT FNCA$(23,10)"YOU DIDN'T PRESS A KEY 0-9 ON THE KEYPAD"
     'ERROR MESSAGE ON LINE 23
2040 FOR X=1 TO 500:NEXT X    'DELAY SO MESSAGE CAN BE READ
2050 PRINT FNCA$(1,1);EL$     'ERASE LINE 1
2060 PRINT FNCA$(23,1);EL$    'ERASE LINE 23
2070 GOTO 1000       'BACK FOR NEXT INPUT
2080 A$=INPUT$(1)   'GET THIRD CHARACTER
2090 A=ASC(A$)-111        'CALCULATE VALUE FOR ON...GOTO
2100 IF A>0 AND A<11 GOTO 3000      'CHECK FOR VALID INPUT
2110 PRINT FNCA$(23,10)"YOU DIDN'T PRESS A KEY 0-9 ON THE KEYPAD"
     ERROR MESSAGE ON LINE 23
2120 FOR X=1 TO 500:NEXT X    'DELAY SO MESSAGE CAN BE READ
2130 PRINT FNCA$(1,1);EL$     'ERASE LINE 1
2140 PRINT FNCA$(23,1);EL$    'ERASE LINE 23
2150 GOTO 1000       'BACK FOR NEXT INPUT
3000 ON A GOTO 3100,3010,3020,3030,3040,3050,3060,3070,3080,3090
     'GO TO CORRECT SUB-ROUTINE FOR EACH KEY
3010 X=21:Y=17:GOTO 4000 'COLUMN 21, LINE 17, PRINT BOX
3020 X=28:Y=17:GOTO 4000 'COLUMN 28, LINE 17, PRINT BOX
3030 X=35:Y=17:GOTO 4000 'COLUMN 35, LINE 17, PRINT BOX
3040 X=21:Y=10:GOTO 4000 'COLUMN 21, LINE 10, PRINT BOX
3050 X=28:Y=10:GOTO 4000 'COLUMN 28, LINE 10, PRINT BOX
3060 X=35:Y=10:GOTO 4000 'COLUMN 35, LINE 10, PRINT BOX
3070 X=21:Y=3:GOTO 4000  'COLUMN 21, LINE 3, PRINT BOX
3080 X=28:Y=3:GOTO 4000  'COLUMN 28, LINE 3, PRINT BOX
3090 X=35:Y=3:GOTO 4000  'COLUMN 35, LINE 3, PRINT BOX
3100 PRINT ED$:GOTO 1000 'ERASE DISPLAY AND START OVER
4000 FOR Z=1 TO 6        'FOR EACH LINE OF THE BOX
4010 PRINT FNGL$(X,Y,Z)  'PRINT GRAPHIC LINE
4020 NEXT Z              'CONTINUE FOR EACH LINE OF BOX
4030 GOTO 1000           'BACK FOR NEXT INPUT
5000 PRINT ED$ 'ERASE DISPLAY
5010 PRINT SS$ 'SCREEN SCROLL (EXIT HOLD MODE)
5020 PRINT CO$ 'CURSOR ON
5030 PRINT GO$ 'GRAPHICS OFF
5040 PRINT LO$ 'LINE 25 OFF
5050 PRINT KX$ 'EXIT ALTERNATE KEYPAD MODE
5060 END
```

Now you can apply these features to your own programs to save operating time and effort.

A word about compiling MBASIC. If you use a compiler, be sure to include the test for "?" when using the Alternate Keypad Mode. MBASIC programs run much faster when compiled and the question mark will be received by the computer even though it wasn't when the same program was run under the interpreter. That fact contributed greatly to your author's impending premature baldness.

This article and all programming is copyrighted by the author. If you have any questions or wish to express yourself about this column, please write me at the address given at the beginning of the article. If you require an answer, please include a self-addressed stamped envelope.

See you next month.

Something for Everyone

H19/H89

THE ULTRA ROM

ULTRA is an enhanced ROM for your H19 terminal display which retains compatibility with present software (except ANSI sequences), yet adds new capabilities that you probably thought were impossible: self-test, second page display (with option), operation to 38.4K baud, user function keys, and much, much more. Comes with a pair of ROM's, binder, and complete documentation.

ULTRA	$49.95
Page Two Option	$14.95

H89/H100

ZLYNK/II

ZLYNK/II is a CP/M or CP/M-85 modem system like nothing else in the entire industry!ZLYNK/II actually contains its own text editor and language (ZLINGO). It can be used directly by the operator, or else programmed to operate completely independently. Eight protocols are supplied with ZLYNK/II, but more can be added via overlays at any time.

ZLYNK/II (hard sector)	$59.95
ZLYNK/II (soft sector)	$59.95

H100

PALETTE

An interactive graphics system for artists and home users alike. With PALETTE you can draw superb color or monochrome graphics displays and even dump to compatible printers. PALETTE currently supports both a lightpen and interlace mode, and it comes with a complete tutorial! NOT to be confused with others that SEEM to be similar. PALETTE is truly one-of-a-kind!

PALETTE (ZDOS)	$69.95
Z100 Lightpen	$150.00

H8/H89

THE HDOS TOOLKIT

The HDOS Toolkit is a group of utilities for both the comfort of the average user, and for the education of developing assembly-language programmers. We supply full source code with the TOOLKIT with comments to unlock truly 'Wizard-Class' tricks. Any single program in the package is worth the price of the entire group. If you like HDOS, you'll LOVE this!

Toolkit (hard sector)	$29.95
Toolkit (soft sector)	$29.95

All of the above items (and a LOT more!) are available direct from Software Wizardry, Inc. as well as from most Heathkit Electronic Centers and many Zenith Data Systems dealers and distributors as well. Please add $2 minimum (or 2%, whichever is greater) shipping/handling plus sales tax if shipped to a Missouri address. Software Wizardry, Inc. also has over 450 other items of interest to Heath/Zenith users as well. Call or write for our FREE price list. Dealer inquiries and HUG group purchases welcome!

VISA MasterCard

ZENITH data systems
AUTHORIZED SALES & SERVICE

Software Wizardry
THE MAGIC TOUCH

122 Yankee Drive
St. Charles, MO 63301
(314) 946-1968

Getting Started With Assembly Language

(or, It's Your Turn To Help the Beginner)

Pat Swayne
Software Engineer

In this month's "Getting Started", the program that I have provided for your instruction is one that your friends who are new to computers may find helpful. But before we get started (pun intended), let me say (as usual) that if you are new to HUG and to assembly language, this article is a continuation of a series that started with the April (#39) issue of REMark, and you might want to get some back issues and get caught up. Since all of this year's issues will be available in a bound volume soon, you may want to wait until it is available.

Part VIII — Disk Files in HDOS, Part 2

In the last two installments, we introduced reading disk files in HDOS and CP/M. I pointed out that a file can be read into memory from the disk in two ways: by reading in individual records or sectors until the whole file is read, or by reading in the whole file (or as much as will fit) in one operation. The sample program I presented used the first method, so I will use the second method this time, and will cover writing to the disk as well as reading from it. The sample program for this month is a file copy utility called XFER (which stands for "transfer") that takes its argument from the command line using the syntax

```
>XFER (source drive):FILENAME.EXT TO (destination drive):
```

If the source drive is SY0:, its designation may be left off. For example, if you wanted to copy ASM.ABS from SY0: to SY1:, you could enter:

```
>XFER ASM.ABS TO SY1:
```

Since many new computer users are confused by the "destination = source" syntax of PIP, they may find this program helpful. So, after you have read this article and entered and studied the program, feel free to make it available to any new users you may know. This program is limited in that it does not allow "wild cards", or allow you to change the file name while copying, but it will copy files of any size, and is easy to use.

The XFER program starts out by looking for a command line argument using the procedure presented previously, and if it does not detect an argument, an error message is printed. If an argument is detected, the program first sets things up before examining the argument. Since this program reads large amounts of data at a time from the disk, it must reserve some memory space. In HDOS, any time you want to use a lot of memory space outside of your program, you should reserve the space, using the .SETTP (sometimes written as .SETTOP) system call. The .SETTP system call sets the user memory limit to the value in the HL register pair. If the top requested is too high, HDOS returns the actual maximum available in HL, so you can specify a value that is too big on purpose to find out how much memory space you have. In XFER, I put -1 (which is really 0FFFFH or 377377 split octal) in HL for the first call to .SETTP.

When you need to reserve a lot of memory, one thing that you should consider is whether the program you are writing will use either of the two HDOS overlays. The program will run faster if you load the overlay or overlays required before reserving memory. However, there is no need to load both overlays as shown in the HDOS System Programmer's Guide if you do not need both. The first overlay is required if your program opens or closes files, so I decided to load it in XFER. The second overlay is used in mounting and dismounting disks, which XFER does not do, so it is not loaded. The .LOADO system call is used to load overlays, with the number of the overlay (0 or 1) in the A register. Loading overlays should always be done before using .SETTP.

After the overlay is loaded and the memory limit is found, XFER determines how much buffer space it has by subtracting the buffer start from the memory limit. This result is converted to a multiple of 256 (sometimes called a "page multiple") by putting 0 in the L register, and then saved. Then .SETTP is used again to set the upper limit.

With memory reservation taken care of, the program next examines the command line argument. It first skips over any spaces that may be at the beginning of the argument. Since the program has to skip over spaces more than once, I coded the part of the program that skips spaces as a subroutine. It is a good idea to make any section of a program that will be used more than once a subroutine, so that you only have to write it once. When you are using the space character as a delimiter separating parts of a command line, you should allow for the possibility of the operator entering more than one space.

When the first non-space character is found, its address is saved on the stack and then XFER goes on to see if the next part of the command line is the word "TO" followed by at least one space. If it is not, a message is printed, and the program exits. Otherwise, the program goes to the next word, which should be the destination drive designation, and moves it to the output default block. The code to move characters is a subroutine, since it will be used again later. Notice that the routine only moves three characters. This means that the colon after the destination drive designation is ignored, and may be left off when you run the program. So, to transfer ASM.ABS as in the example above, you could enter:

```
>XFER ASM.ABS TO SY1
```

Notice also that the move subroutine quits early if one of the characters being moved is a zero. This was done so that when you are moving files from another drive to SY0:, you can leave off the 0, as in

```
>XFER SY1:BASIC.ABS TO SY
```

In this case, you MUST leave off the colon, or an illegal drive designation will be copied to the default block. The last character in a command line is zero, so the move subroutine stops after moving SY. This syntax will work for unit zero of any driver, such as DK0:, and you can also copy files to a printer, with a command like

```
>XFER GAME.BAS TO LP
```

as long as you leave off the colon, and the printer driver is already loaded (LOAD LP:).

After the output default block is set up, the program restores the address of the first command line character, and uses the .DECODE system call to break up the information into its component parts: the drive designation, the file name, and the name extension. This allows us to use the name and extension for the output file with a new drive designation. The drive designation is copied to the input default block, and the extension is copied to both blocks. Then a zero is put into the first byte after the name area of the .DECODE table so that the name will be properly terminated. The .DECODE system call puts zeros after a name that is less than 8 characters, but one that is 8 characters long would run into the next part of the .DECODE table if we did not put in the zero. One peculiarity of the .DECODE system call is that it decodes the drive unit number into a binary number instead of an ASCII (text) number, so the program uses an OR statement (ORI '0') to fix that.

After the input information is all decoded and set up, XFER opens the input and output files, and enters a copy loop. The size of the buffer calculated earlier is loaded into the BC registers so that HDOS will read from the input file until the buffer is full, or the file end is found. If the end is found, a flag is set and the amount of data read is calculated. HDOS returns two useful items of information whenever a program reads to the end of a file. In the BC register pair, it places the number of bytes remaining in the buffer, and in the DE registers, it places the next unused address. However, what we need to know is how much data was read, which is not returned directly. The HDOS System Programmer's Guide shows how to get this from the information in the BC registers, but it is easier to just subtract the buffer start from the next unused address in DE, so that is what is done in XFER. The amount of data read is placed in location MSIZE, which previously held the buffer size.

After the read part of the loop, the data in the buffer is written to the output file. The number in MSIZE (either the buffer size or the amount read) is used for the amount to write. When the data is written, the flag indicating the end of the file is checked, and if it is not set, the program loops back to the read portion of the loop, and continues until the entire file is copied. You can see, then, that a file of any size can be copied with this program. We could have copied the file one sector at a time, and saved the trouble of setting up the memory buffer, etc., but it would take MUCH longer to copy a file that way.

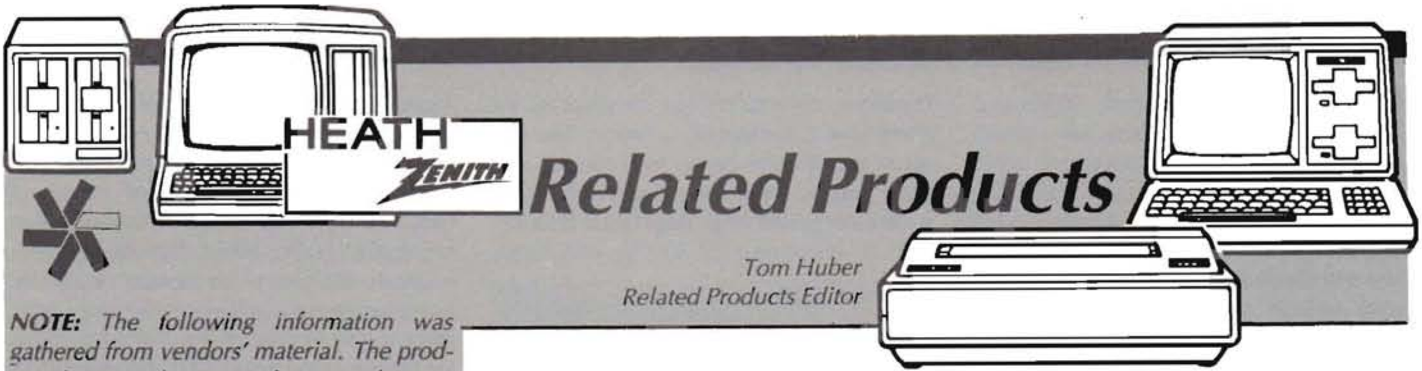
When the copy operation is completed, the program closes both files and exits. In the next installment, I will present a CP/M version of this program. I will also announce a little contest for those of you who have been following this series. I will not give out any specifics of the contest until then, to be fair to CP/M users, but it will involve writing an assembly language program to certain specifications. There will be two categories, HDOS and CP/M, with PRIZES for both winners! Don't miss it!

P.S. One reader who wrote to me asked why I didn't ORG the VIEW program in REMark Issue #45. It's probably because I forgot to, but the HDOS assembler will assume that you want a standard ORG address of 42200A (2280H) if you do not specify any. Notice that today's program does not have an ORG address either.

```
*        XFER - FILE TRANSFER PROGRAM
*
*        THIS PROGRAM DEMONSTRATES READING AND WRITING
*        DISK FILES UNDER HDOS.  FILES ARE READ AND
*        WRITTEN IN LARGE BLOCKS.
*
*        THIS PROGRAM COPIES FILES USING THE FOLLOWING
*        SYNTAX:
*
*        >XFER dev:FILENAME.EXT TO dev:
*
*        WHERE dev: IS A DRIVE DESIGNATION.

*        BY P. SWAYNE, HUG  26-OCT-83

*        HDOS DEFINITIONS

.EXIT    EQU     0
.READ    EQU     4
.WRITE   EQU     5
.LOADO   EQU     10Q
.OPENR   EQU     42Q
.OPENW   EQU     43Q
.CLOSE   EQU     46Q
.SETTP   EQU     52Q
.DECODE  EQU     53Q
.ERROR   EQU     57Q
$TYPTX   EQU     31136A
EC.EOF   EQU     1                    END OF FILE FLAG
EC.NEM   EQU     21Q                  NOT ENOUGH MEMORY

*        MAIN PROGRAM

START    LXI     H,0                  LOCATE STACK
         DAD     SP
         MOV     A,L
         CPI     200Q                 HAS IT MOVED?
         JNZ     GOTFILE              IF SO, CONTINUE
ERREX    CALL    $TYPTX
         DB      10,'The correct use of this program is',10,10
         DB      '>XFER dev:FILENAME.EXT TO dev:',10,10
         DB      'where dev: is a drive designation '
         DB      '(SY1:, DK0:, etc.),',10
         DB      'and FILENAME.EXT is file you want to XFER.'
         DB      10+80H
         XRA     A
         SCALL   .EXIT                EXIT TO HDOS
GOTFILE  PUSH    H                    SAVE ARGUMENT ADDRESS
         XRA     A
         SCALL   .LOADO               LOAD FIRST HDOS OVERLAY
         JC      ERROR
         LXI     H,-1
         SCALL   .SETTP               FIND MEMORY LIMIT
         PUSH    H                    SAVE IT
         LXI     D,-BUFFER
         DAD     D                    CALCULATE FREE SPACE
         MVI     L,0                  MAKE IT PAGE MULTIPLE
         MOV     A,H
         ORA     A                    ANY MEMORY THERE?
         MVI     A,EC.NEM             ASSUME NOT
         JZ      ERROR                NO MEMORY, EXIT
         SHLD    MSIZE                ELSE, SAVE MEMORY SIZE
         POP     H                    RESTORE MEMORY TOP
         SCALL   .SETTP               SET IT
         JC      ERROR
         POP     H                    GET ARGUMENT ADDRESS
         CALL    SOS                  SKIP OVER SPACES
         PUSH    H                    SAVE INPUT FILE NAME
FNS      MOV     A,M                  GET A CHARACTER
         CPI     ' '                  SPACE?
         JZ      FNDSP                FOUND SPACE
         INX     H                    ELSE, INCREMENT POINTER
         JMP     FNS                  AND FIND NEXT SPACE
FNDSP    CALL    SOS                  SKIP OVER IT
         MOV     A,M
         CPI     'T'                  LOOK FOR 'TO'
```

```
        JNZ     ERREX           EXIT IF NOT FOUND
        INX     H
        MOV     A,M
        CPI     'O'
        JNZ     ERREX
        INX     H
        MOV     A,M
        CPI     ' '
        JNZ     ERREX
        CALL    SOS             FOUND "TO", SKIP TO DEV.
        LXI     D,OUTDEF        POINT TO OUTPUT DEFAULTS
        CALL    MOVE3           MOVE DEVICE NAME THERE
        POP     H               GET FILE NAME ADDRESS
        LXI     B,DECTBL        POINT TO DECODE TABLE
        LXI     D,INDEF AND DEFAULT BLOCK
        SCALL   .DECODE         DECODE FILE NAME
        JC      ERROR           BAD NAME
        LDA     DEV+2           GET UNIT NO.
        ORI     '0'             MAKE IT ASCII
        STA     DEV+2
        LXI     H,DEV           POINT TO DECODED DEVICE
        LXI     D,INDEF PUT IT HERE
        CALL    MOVE3
        LXI     H,EXT           POINT TO DECODED EXTENSION
        PUSH    H               SAVE ADDR
        LXI     D,INDEF+3       PUT IT HERE
        CALL    MOVE3
        POP     H               GET DECODED EXTENSION
        LXI     D,OUTDEF+3      PUT IT HERE, TOO
        CALL    MOVE3
        XRA     A
        STA     EXT             ENSURE NAME TERMINATED
        XRA     A
        LXI     H,NAME
        LXI     D,INDEF
        SCALL   .OPENR          OPEN INPUT FILE
        JC      ERROR
        MVI     A,1
        LXI     H,NAME
        LXI     D,OUTDEF
        SCALL   .OPENW          OPEN OUTPUT FILE
        JC      ERROR

*       COPY LOOP

LOOP    LHLD    MSIZE           GET MEMORY SIZE
        MOV     B,H
        MOV     C,L             IN BC
        LXI     D,BUFFER        POINT TO BUFFER
        XRA     A               CHANNEL ZERO
        SCALL   .READ           READ INPUT FILE
        JNC     GOODRD          READ OK
        CPI     EC.EOF          END OF FILE?
        JNZ     ERROR           NO, ERROR
        STA     LASTFLG         ELSE, FLAG LAST SEGMENT
        XCHG                    HL = NEXT UNUSED ADDRESS
        LXI     D,-BUFFER
        DAD     D               CALCULATE AMOUNT READ
        SHLD    MSIZE           FIX MSIZE
GOODRD  LHLD    MSIZE           GET AMOUNT TO WRITE
        MOV     B,H
        MOV     C,L             IN BC
        LXI     D,BUFFER        POINT TO BUFFER
        MVI     A,1             CHANNEL 1
        SCALL   .WRITE          WRITE OUTPUT FILE
        JC      ERROR
        LDA     LASTFLG
        ORA     A               LAST SEGMENT?
        JZ      LOOP            IF NOT LOOP

*       COPY DONE, CLOSE FILES

        XRA     A
        SCALL   .CLOSE          CLOSE INPUT FILE
        MVI     A,1
        SCALL   .CLOSE          CLOSE OUTPUT FILE
```

```
        JC      ERROR
        CALL    $TYPTX
        DB      10,'DONE!',10+80H
        XRA     A
        SCALL   .EXIT

*       PROCESS ERRORS

ERROR   PUSH    PSW             SAVE ERROR CODE
        CALL    $TYPTX
        DB      10,'ERROR --',' '+80H
        POP     PSW
        MVI     H,7             LET .ERROR RING BELL
        SCALL   .ERROR
        XRA     A
        SCALL   .EXIT

*       SUBROUTINES

*       SKIP OVER SPACES

SOS     MOV     A,M             GET A CHARACTER
        CPI     ' '             SPACE?
        RNZ                     IF NOT, RETURN
        INX     H               ELSE, MOVE TO NEXT CHAR
        JMP     SOS             TRY AGAIN

*       MOVE THREE BYTES FROM (HL) TO (DE)
*       IF ZERO ENCOUNTERED, TERMINATE MOVE

MOVE3   MVI     B,3             SET A COUNTER
MOVE31  MOV     A,M             GET A BYTE
        ORA     A               ZERO?
        RZ                      IF SO, RETURN
        STAX    D               ELSE, STORE BYTE
        INX     H               INCREMENT POINTERS
        INX     D
        DCR     B
        JNZ     MOVE31          LOOP UNTIL DONE
        RET

*       DATA AREA

MSIZE   DW      0               BUFFER MEMORY SIZE
LASTFLG DB      0               LAST SEGMENT FLAG
INDEF   DB      'SY0',0,0,0     INPUT DEFAULT
OUTDEF  DB      'SY0',0,0,0     OUTPUT DEFAULT
DECTBL  DB      0               DECODE TABLE START
DEV     DS      3               DECODED DEVICE
NAME    DS      8               DECOCED NAME
EXT     DS      3               DECODED EXTENSION
        DS      4               RESERVED DECODE BYTES
BUFFER  EQU     *               BUFFER STARTS HERE

        END     START
```



Related Products

Tom Huber
Related Products Editor

NOTE: The following information was gathered from vendors' material. The products have not been tested nor are they endorsed by HUG. We are not responsible for errors in descriptions or prices.

H17 Disk Support System

The Soft Firm has announced a complete CP/M disk diagnostic and data recovery system for 8-bit Heath/Zenith Computers using the hard-sectored controller. The diagnostics include a soft error counter; error mapping of a complete disk by track, sector, and error type; measure disk speed; measure or change head seek time; verify operation of all disk control signals and status functions; and generate repetitive signals as a trouble shooting aid. The disk recovery functions include a disk directory of all files, both active and erased; a disk usage map; an unerase function of previously erased disk files; disk formatting and bad sector lockout; and a sector modification program. Requires CP/M 2.2.03 or higher, 48K, and a hard-sectored controller.

Vendor: The Soft Firm
P.O. Box 1125
Picayune, MS 39466
(601) 798-0740
Price: $65.00
(MS res. add 5% sales tax)

Multifunction Utility Board for H/Z-89/90

The H89UTI is a multifunction utility board that replaces that standard serial I/O board and includes HDOS and CP/M support software on disk. The board features a quartz crystal controlled real time clock, a bidirectional 8-bit parallel interface, and two standard RS-232 serial ports. Options include a hardware math package, battery backup, modem serial port, and several printer packages (including cables). Contact the vendor for pricing of options and information on other products.

Vendor: FBE Research Company, Inc.
P.O. Box 68234
Seattle, WA 98168
(206) 246-9815 (evenings only)
Price: $159.95 (no options,
WA res. add 6.6% sales tax)

New Products from Generic Software

Fund-Master is a program that acts as a stock market investment aid, useful for the management of stocks and mutual funds for casual investors who do not have the time to become familiar with all aspects of the stock market, but who need effective buy and sell advice. Fund-Master takes a conservative approach to stock investments, with the emphasis on obtaining consistent returns. Requires H-8/H/Z-89/90 and 48K for HDOS or 56K for CP/M and MBASIC. Specify hard/soft-sector disk.

Castle-Gor is the first program in a series of Medieval Adventure games. Journey to the Black Forest in search of Castle-Gor, where strange creatures and huge fortunes await! The object of the game is to collect as many treasures as possible, then escape the mysterious dungeons of the castle. Requires H-8/H/Z-89/90, 64K, CP/M, and MBASIC. Specify 5.25-inch disk type.

AUDIT (Automatic, User-Defined, Information handling Tools) is a full-featured data base management system that accommodates user- defined (multiple) input forms, editing, default record information, indexed access to data base, transaction log, sorting, report and label generation, computed values, and utility functions. Requires 64K and CP/M-80 or CP/M-85. Specify 5.25-inch disk type.

Micro-Cabinet provides a low-overhead data filing and retrieval system with keyed access, user-defined input forms, default record information, fixed or variable length fields, and user- defined output formats. Requires 48K and HDOS, CP/M-80, or CP/M-85.

Catalog-Master is a utility for creating and maintaining a master file directory for multiple disk volumes. It features a file description editor, file selection options, sorting, merging into a master catalog, and output to a terminal, printer, or disk file. Requires 48K and HDOS 2.0, CP/M-80, or CP/M-85.

Vendor: Generic Software
P.O. Box 790
Marquette, MI 49855
(906) 249-9801
Prices: Fund-Master $39.95 + $2 S&H
Castle-Gor $24.95 + $2 S&H
Audit $129.00 + $4 S&H
Micro-Cabinet $75.00 + $4 S&H
Catalog-Master $29.95 + $2 S&H
(MI res. add 4% sales tax)

Portable Printer from MPI

MPI has introduced the Sprinter, a portable printer that features 80 columns, 160 characters per second, parallel port, 5 character sets, user-programmable character design, a 4K buffer, full graphics, and friction or tractor feed. Options include buffers up to 68K, IEEE-488 interface, RS-232 interface, and MPI's SoftSwitch keypad.

Vendor: Micro Peripherals, Inc.
4426 South Century Drive
Salt Lake City, UT 84107
(800) 821-8848
Price: $795.00 (standard model)

Word-Magic Menu for CP/M and Z-DOS

Word-Magic Menu, available for H/Z-89/90/100 series computers, is a driver menu for existing software. It brings up and accesses word processors, electronic spreadsheets, communications software, spelling checkers, and other operating system utility programs. Single keystroke operation works with WordStar, Magic Wand, Peachtext 5000, SuperCalc, Multiplan, and others. The support software occupies 6K of memory and is available in 5.25- and 8-inch, CP/M or Z-DOS disks.

Vendor: Moxon Inc.
338 Indigo Street
Mystic, CT 06355
(203) 536-6953
Price: $49.95

Magnolia Announces a Local Area Network

MAGNet, a local area network, allows multiple users to perform various tasks simultaneously, using separate processors, while sharing common resources such as disk files and printers. MAGNet uses a physical bus architecture that allows up to 64 connections and a logical ring "token passing" technique (which stabilizes network performance as compared to CSMA/CD schemes). Software compatibility is provided through Digital Research's CP/M, CP/Net, and MP/M II. Hardware support via Magnolia supplied systems and Heath/Zenith terminals and H/Z-89/90 and H/Z-100 computers. Contact the vendor for full information and pricing.

Vendor: Magnolia Microsystems, Inc.
2264 - 15th Ave. West
Seattle, WA 98119
(800) 426-2841/(206) 285-7266

New Software Line

Budget Software is a new company offering a variety of software packages. **Capture** is an action game where the player is trapped in a warehouse full of boxes and hungry beasts. **Arena** is a two player war simulation game where you have been selected to represent your planet as commander of a robot army. **Letterit** uses a Diablo printer for special lettering and signs in old English or your own font. **Music89** supplies music entered directly from sheet music through the standard H-88-3 or H-8-4 interface cards. **Speller** is a CAI program for teaching spelling through drill and practice (requires a text to speech synthesizer such as the Votrax). Send a self-addressed, stamped envelope for a copy of the vendor's catalog, licensing agreement, and pricing. All software requires a Heath/Zenith computer running CP/M-80 or CP/M-85 (and additional peripherals as noted—the H/Z-100 series cannot run Music89).

Vendor: Budget Software
P.O. Box 221
Berrien Springs, MI 49013
Prices: $14.99 each program

CP/M Public Domain Software on 5.25-Inch Disks

Headware is offering copies of SIG/M volumes 1 through 114 and CP/MUG volumes 1 through 91 on Heath format CP/M disks for a nominal fee.

Vendor: Headware
2865 Akron St.
Atlanta, GA 30344
Price: $17.00 (3-disk Hard-sectored)
$12.00 (2-disk Single-sided, Soft-sectored)
$7.00 (1-disk Double-sided, Soft-sectored)

Kleen-Line Conditioner

ELectronic Specialists has announced the Kleen-Line Conditioner, a power line regulator that features heavy load startup, sine wave output, input spike suppression, and wide-bank prefiltering. Regulation is to 117 VAC ± 4% output for 90-140 VAC input.

Models are available in 250, 500, 1000, and 2000 watt ratings. For more information and full pricing, contact the vendor.

Vendor: Electronic Specialists, Inc.
171 South Main Street
P.O. Box 389
Natick, MA 01760
(800) 225-4876

High Speed Graphics Enhancement for H/Z-29

The Imaginator 2 provides comprehensive graphic capabilities for the H/Z-29 computer terminals. Developed by Cleveland Codonics, Inc., features 672 by 500 pixel display, 500,000 pixel/second drawing rate, and dual on-board processors. The megabit of memory may be arranged in horizontal, vertical, or rectangular formats and viewed by panning, scrolling, and zoom (up to 16x) features. Software support is through a superset of DEC ReGIS, which can receive instructions from any high level language (PL/1, FORTRAN, Pascal, BASIC, etc.). The enhancement can be field installed and is available with X-ray safe, green (P42 and P39) or amber phospher CRT's. Typical applications include graphic data formats for business, scientific, engineering, and personal use; CAD/CAM; and a variety of control activities. Options include Tektronix 4010/4014 emulation, composite video output and input, automatic synchronization with external video sources, and printer and parallel communication ports.

Vendor: Cleveland Codonics, Inc.
8001 Fnglewood
Cleveland, OH 44130
(216) 243-1198
Price: $895.00

More for HERO

A new book, "Heath's Robot HERO". 68 Experiments: Fundamentals and Applications, provides 68 experiments that cover the fundamentals behind the control of all of HERO's I/O systems along with applications for security, home, office, "fun and games", hospitals, etc. One of the manual's purposes is to educate the user to become innovative and capable of conceiving and implementing programs to control and utilize HERO's I/O capabilities. 175 pages, 8-1/2 x 11.

Vendor: Dr. Howard Boyet
Microprocessor Training, Inc.
(MTI Publications)
14 East 8th Street
New York, NY 10003
(212) 473-4947
Price: $21.95

Two for Z-DOS from Westcomp

Screen Potpourri is a program that accepts single-word commands on the Z-DOS command line to control the terminal functions. They can be chained to accomplish multiple functions. Five font files are also provided to redefine the characters on the display. Requires Z-DOS and 128K.

Piglat is a program that takes an ASCII text file and translates it to pig latin. Written in C, the source code is supplied free. Just what everyone needs for those special notes and letters to those special people on those special occasions. Requires Z-DOS and 128K.

Vendor: Westcomp
P.O. Box 494
Temple City, CA 91780
Prices: Screen Potpourri
$16.95 + $2.00 S&H
Piglat $29.95 + $2.00 S&H
(CA res. add 6.5% sales tax)

StoreMinder Software Series Available for Z-100

XtraSoft is offering individual modules in its StoreMinder software series as stand-alone applications or part of a fully- integrated system. The modules include SalesMinder, a point-of- sale package; StockMinder, an inventory management system; and MailMinder, a mailing list/customer profile system. StoreMinder can handle a bar code reader, preset multiple pricing, custom price labels, fractional pricing, assignment of commission rates, and multiple terminal operation. Contact the vendor for more information and pricing.

Vendor: XtraSoft, Inc.
553 Weddell Drive
Sunnyvale, CA 94089
(408) 747-1400
(800) 448-1400

HERO Gets a New Head and a Handle

House of York gave HERO I a new personality by attaching a unique head to the "shoulders". The head gives a focal point, eye to eye contact, and turns HERO into a "high-class gentleman" robot. It adds about a foot to his height. The head is eight inches in diameter, has a colorful hat, bow tie, and is securely attached to the shoulder.

House of York is also offering a secure and convenient handle for carrying HERO I without fear of dislodging any of the outside panels. The handle snaps on or off in about ten seconds and, when not in use, folds to 1-1/2 x 2 x 16-1/2 inches.

Vendor: House of York
1992 Borchers Drive
San Jose, CA 95124

Prices:	Head	$69.95
	Handle	$69.95

HDOS Z-80 Debug Package for H/Z-89/90 Computers

DB89 features access to all of the Z-80's registers, 4 different I/O number base support, byte and word operations, up to 2 breakpoints, 10 offset biases for debugging relocatable modules, disk mount/dismount, disk file loading/writing, and descriptive error messages. Available on hard-sector.

Vendor: JRA Systems, Inc.
Box 555
St. Clair Shores, MI 48080
(313) 773-7082
Price: $34.95

Running Log Tracks Jogger's Miles

For runners/joggers who need a better systems for tracking their statistics, Van Christopher is offering Running Log. The program will store and display dates, miles run, run time, terrain, location of run, and limited comments. Requires CP/M 2.2.03 and MBASIC. Specify disk format: 5.25-inch (hard sector) or 8- inch.

Vendor: Van Christopher
13751 32nd Avenue NE #113
Seattle, WA 98125
(206) 364-6166
Price: $20.00 + $2.00 S&H
(add $3.00 for 8-inch format)

Correction to New Products For Microservices

Microservices announces **ZANIMATE**, a new animation program for ZBASIC users'. Please note below a correction to their phone number and prices of the 2 new products published in the October (#45) Issue of REMark.

Vendor: Microservices
P.O. Box 7093
Menlo Park, CA 94025
(415) 851-3414

Prices:	ZPATTERN	$24.95
	ZPALETTE	$59.75
	ZANIMATE	$64.95

☜ Vectored from 9

```
FFEED:   MVI   A,0H
         STA   L03AE         ;Zero the counter

LOOP:    PUSH  B
         PUSH  D
         PUSH  H
         MVI   C,0002H       ;CONOUT
         MVI   E,0DH         ;CR
         CALL  0005H         ;CALL BDOS
         MVI   C,0002H       ;CONOUT
         MVI   E,0AH         ;LF
         CALL  0005H         ;CALL BDOS
         LDA   L03AE         ;Initialize blank line counter
         INR   A             ;Increment counter
         STA   L03AE         ;Store count
         CPI   0AH           ;Change number of blank lines
         JNZ   LOOP          ;Loop if not zero
         POP   H
         POP   D
         POP   B
         MVI   A,0H
         STA   L03AE         ;Zero counter
         JMP   L0136         ;Start new page

         END
```

Use ASM.COM to assemble the whole program after the above modification and then use LOAD.COM to obtain CAT2.COM.

My copy of the cataloging system files which include CAT2.COM credits Ward Christensen with writing these programs, while a disk which I purchased from QUICKDATA, Inc. (Henry Fale, H-SCOOP) which includes FMAP.COM, CAT.COM, UCAT.COM but not CAT2.COM, states that most of the programs on this disk were donated by Rick Martinek of Associates Business Group. Whether or not this applies to the master cataloging system, I have no way of knowing but it is mentioned here to acknowledge possible authorship or at least the modification of these programs to run on Heath equipment.

William G. Coe
1220-513 Vienna Dr.
Sunnyvale, CA 94086

An Assembly Language Program

Dear REMark,

"Transmit Page - ESC # - Transmits lines 1 through 24. (The computer requires a special routine to use this feature.)"

The above paragraph is one of only two things I don't like about the H-89 Operation Manual. The other is the paragraph that proceeds it.

"Transmit Line - ESC] - See 'Transmit Page below"

These are, in my opinion, the only two areas where the fine Heath Manual leaves you hanging with no leads as to how to proceed to use a function. The above referenced 'special routine' was therefore the first idea that came to mind when, in the process of learning Assembly Language through the Heath course and Pat Swayne's "Getting Started ..." articles, I found myself in need of an .ASM program project.

REMark readers may be interested in the results of that project provided in this listing of "XMITPAGE", a Transmit Page Function Demonstration.

My initial versions of this program were dismal failures in that many randomly distributed characters of the transmitted page were not recovered. Using the technique of disabling console interrupts during page transmission employed in Walter Rison's Hard Copy Device Driver (HC.DVD) corrected this problem. The DUMP subroutine is a simplification of code from that program. DUMP will probably not work for any screen (page) incorporating reverse video, graphics, or similar screen commands. I believe that accommodation of these features would require detection and storage of the Escape codes for these functions as well for the 80 displayed characters on each line. That's the next project I have in mind as soon as I figure out how to transmit Escape codes with the FILL subroutine.

The development of this program also revealed a very interesting characteristic of my machine, i.e., it would not print the NUM1 pattern in Column 80 of Line 1. After almost getting ulcers (PS: Why didn't you warn us of this danger?), I finally worked around the problem with the instructions in the second and third lines of the FILL subroutine. Can anyone tell me what I did wrong?

Tom Colman
550 Temple St.
Satellite Beach, FL 32937

```
        TITLE   'XMITPAGE.ASM'

        STL     'by Thomas M Colman, based on techniques'
        STL     'used by Walter A Rison in HC.DVD'

***     PROGRAM EQUATES
*

.CONSL  EQU     6
.EXIT   EQU     0
.PRINT  EQU     3
.SCIN   EQU     1
.SCOUT  EQU     2
$TYPTX  EQU     031136A
USERFWA EQU     042200A

***     PROGRAM STRUCTURE
*

        ORG     USERFWA
START   CALL    DESCRIB           PRINT DESCRIPTION OF PROGRAM
        CALL    FILL              PRINT DISTINCTIVE PATTERN ON SCREEN
        CALL    DUMP              DUMP SCREEN TO MEMORY
        CALL    RECOVER           RECOVER SCREEN PATTERN FROM MEMORY
        CALL    EXIT              RETURN TO HDOS

***     PRINT DESCRIPTION OF PROGRAM
*

DESCRIB CALL    CLEAR
        CALL    $TYPTX            GET TYPE TEXT FUNCTION
        DB      012Q,012Q
        DB      011Q,'                        "TRANSMI'
        DB           'T PAGE"',012Q,012Q
        DB      011Q,'This program demonstrates the '
        DB           'Transmit Page (ESC #) Function  ',012Q
        DB      011Q,'of the H-89 and provides a routi'
        DB           'ne to store the characters of   ',012Q
        DB      011Q,'the transmitted screen in memory'
        DB           ' for future use.  The program   ',012Q
        DB      011Q,'will first fill the monitor scre'
        DB           'en with a distinctive pattern.  ',012Q
        DB      011Q,'The computer may then be command'
        DB           'ed to store the pattern in      ',012Q
        DB      011Q,'memory and then reprint the patt'
        DB           'ern on the screen again.        ',012Q,012Q
        DB      011Q,'                 Hit BLUE Key t'
        DB           'o Fill Screen                   ',012Q,012Q
        DB      011Q,'                 Hit ESC # to T'
        DB           'ransmit Screen                  ',012Q,012Q
        DB      011Q,'                 Hit RED Key to'
        DB           ' Recover Screen                 ',012Q,012Q
        DB      011Q,'                 Hit GREY Key t'
        DB           'o Return to HDOS                ',200Q

        CALL    NECHO             SET CHAR MODE WITHOUT ECHO
        MVI     E,120Q            SET CHECK FOR BLUE KEY
        CALL    CHECK             CHECK FOR FILL SCREEN CMND

        RET
```

```
***     PRINT DISTINCTIVE PATTERN ON SCREEN
*

FILL    CALL    CLEAR           CLEAR SCREEN

        LXI     H,NUMØ          THESE STEPS REQUIRED BECAUSE MY
        SCALL   .PRINT          MACHINE WON'T PRINT THE FOLLOWING
*                               MESSAGE ON COLUMN 8Ø OF LINE 1

        LXI     H,NUM1          SET HL TO FIRST LINE PATTERN
        MVI     B,25            B = 24 LINES + 1
PATTRN  MVI     C,8             C = 8 PATTERNS PER LINE
        DCR     B               NUMBER OF LINES REMAINING
        JZ      CMND            PATTERN COMPLETED

PRINT   SCALL   .PRINT          PRINT OUT 1Ø CHAR PATTERN
        DCR     C               NO. OF PATTERNS LEFT ON LINE
        JZ      PATTRN          DO THE NEXT LINE
        MVI     D,1Ø            D = 1Ø CHAR PER PATTERN
DCRH    DCX     H               MOVE TO NEXT CHAR IN PATTERN
        DCR     D               NO. OF CHAR LEFT IN PATTERN
        JNZ     DCRH            LOOP UNTIL D = Ø
        JMP     PRINT           PRINT OUT ANOTHER PATTERN

CMND    MVI     E,Ø43Q          SET CHECK FOR 'ESC #'
        CALL    CHECK           CHECK FOR XMIT SCREEN CMND

        RET

***     DUMP SCREEN TO MEMORY
*

DUMP    CALL    CW255           SET WIDTH = 255 (NO AUTO CR)

        LXI     B,1944          B = LENGTH OF BUFFER
        LXI     H,BUFFR         POINT TO START OF BUFFER

        IN      351Q            GET CONSOLE INTERRUPT STATUS
        STA     INTSTAT         SAVE STATUS
        XRA     A               A = Ø
        OUT     351Q            DISABLE CONSOLE INTERRUPTS

        STA     CCOUNT          CCOUNT = Ø

        CALL    $TYPTX          GET TYPE TEXT FUNCTION
        DB      Ø33Q,'#',2ØØQ   SEND TRANSMIT PAGE COMMAND

READY   CALL    READ            CALL FOR A CHARACTER
        MOV     M,A             PUT CHAR IN BUFFER
        INX     H               POINT TO NEXT BUFFER LOCATION
        DCX     B               LENGTH OF BUFFER REMAINING
        MOV     A,B             SEE IF B AND
        ORA     C               C ARE BOTH ZERO
        JNZ     READY           SCREEN DUMP NOT COMPLETE

        DCX     H               ERASE LAST NL CMND
        MVI     A,2ØØQ          SET END OF PAGE INDICATOR
        MOV     M,A             PUT IT IN BUFFER

        LDA     INTSTAT         RECALL INTERRUPT STATUS
        OUT     351Q            PUT IT INTO EFFECT

        CALL    CLEAR           CLEAR SCREEN

        MVI     E,121Q          SET CHECK FOR RED KEY
        CALL    CHECK           CHECK FOR RECOVER SCREEN CMND

        RET

READ    LDA     CCOUNT          A = NO. OF CHAR IN THE LINE
        INR     A               A = A + 1
        STA     CCOUNT          NO. OF CHAR IN THE LINE + 1
        CPI     81              IS CCOUNT = 81?
        JC      READ1           GET A CHAR IF NOT
        XRA     A               A = Ø
        STA     CCOUNT          CCOUNT = Ø
        MVI     A,Ø12Q          INSERT A NEW LINE CHAR
        RET

READ1   IN      355Q            IS A CHARACTER
        ANI     1               READY TO BE READ?
        JZ      READ1           NO - KEEP LOOKING
        IN      35ØQ            YES - READ IT
        RET                     AND PUT IN BUFFER

***     RECOVER SCREEN PATTERN FROM MEMORY
*

RECOVER LXI     H,BUFFR         POINT TO START OF BUFFER
```

```
WRITE   MOV     A,M             GET A CHAR FROM BUFFER
        INX     H               POINT TO NEXT BUFFER LOCATION
        SCALL   .SCOUT          SEND IT TO CONSOLE
        CPI     200Q            IS IT END OF PAGE INDICATOR?
        JNZ     WRITE           IF NOT, GET ANOTHER CHAR

        MVI     E,122Q          SET CHECK FOR GREY KEY
        CALL    CHECK           CHECK FOR RETURN TO HDOS CMND

        RET

***     RETURN TO HDOS
*

EXIT    CALL    CLEAR           CLEAR SCREEN
        CALL    CW80            SET CONSOLE WIDTH = 8Ø
        CALL    CHAR            SET CHAR MODE WITH ECHO
        XRA     A               A = Ø
        SCALL   .EXIT           EXIT TO HDOS

***     COMMON SUBROUTINES
*

CLEAR   CALL    $TYPTX          GET TYPE TEXT FUNCTION
        DB      Ø33Q,'E'+200Q   CLEAR SCREEN COMMAND
        RET

CHAR    XRA     A               A = Ø
        MVI     B,001Q          CODE FOR CHAR MODE WITH ECHO
        MVI     C,201Q          MASK FOR BIT CHANGES
        SCALL   .CONSL          INVOKE CONSOLE SET MODE
        RET

NECHO   XRA     A               A = Ø
        MVI     B,201Q          CODE FOR CHAR MODE W/O ECHO
        MVI     C,201Q          MASK FOR BIT CHANGES
        SCALL   .CONSL          INVOKE CONSOLE SET MODE
        RET

CHECK   SCALL   .SCIN           READ A CHARACTER
        JC      CHECK           CHAR NOT AVAILABLE
        CPI     Ø33Q            IS IT 'ESC'?
        JNZ     CHECK           NO - KEEP LOOKING
CHECK1  SCALL   .SCIN           READ ANOTHER CHARACTER
        JC      CHECK1          CHAR NOT AVAILABLE
        CMP     E               IS IT SPECIFIED CHAR?
        JNZ     CHECK           NO - KEEP LOOKING
        RET                     YES

CW255   MVI     A,3             A = 3, SET CONSOLE WIDTH
        MVI     B,255           TO 255 (NO AUTO CR/LF)
        MVI     C,377Q          ENABLE CHANGE OF FULL BYTE
        SCALL   .CONSL          INVOKE CONSOLE SET MODE
        RET

CW8Ø    MVI     A,3             A = 3, SET CONSOLE WIDTH
        MVI     B,8Ø            TO 8Ø (AUTO CR/LF)
        MVI     C,377Q          ENABLE CHANGE OF FULL BYTE
        SCALL   .CONSL          INVOKE CONSOLE SET MODE
        RET

***     PROGRAM DATA AND STORAGE
*

INTSTAT DB      Ø

CCOUNT  DB      Ø

NUMØ    DB      Ø12Q+2ØØQ
NUM1    DB      'A12345678','9'+2ØØQ
NUM2    DB      'B12345678','9'+2ØØQ
NUM3    DB      'C12345678','9'+2ØØQ
NUM4    DB      'D12345678','9'+2ØØQ
NUM5    DB      'E12345678','9'+2ØØQ
NUM6    DB      'F12345678','9'+2ØØQ
NUM7    DB      'G12345678','9'+2ØØQ
NUM8    DB      'H12345678','9'+2ØØQ
NUM9    DB      'I12345678','9'+2ØØQ
NUM1Ø   DB      'J12345678','9'+2ØØQ
NUM11   DB      'K12345678','9'+2ØØQ
NUM12   DB      'L12345678','9'+2ØØQ
NUM13   DB      'M12345678','9'+2ØØQ
NUM14   DB      'N12345678','9'+2ØØQ
NUM15   DB      'O12345678','9'+2ØØQ
NUM16   DB      'P12345678','9'+2ØØQ
NUM17   DB      'Q12345678','9'+2ØØQ
NUM18   DB      'R12345678','9'+2ØØQ
NUM19   DB      'S12345678','9'+2ØØQ
NUM2Ø   DB      'T12345678','9'+2ØØQ
NUM21   DB      'U12345678','9'+2ØØQ
```

```
NUM22   DB      'V12345678','9'+200Q
NUM23   DB      'W12345678','9'+200Q
NUM24   DB      'X12345678','9'+200Q

BUFFR   DS      1944

        END     START
```

Pat:) The reason why the program would not print the pattern in column 80 is because the program depends on HDOS, with the console width set to 80, to put a carriage-return line-feed after each line. When the program clears the screen to start by printing ESCape E, HDOS counts the E, and therefore ends the first line after 79 characters have been printed.

Actually, it is bad practice to rely on the width setting of HDOS to start new lines, and it is even worse to assume that the user has his width set to 80. (This program will not work properly unless the width is set to 80 upon entry.) And if you must mess with the width setting at all, you should save the user's current setting and restore it at the end of the program, instead of arbitrarily setting the width to your own preference, as this program does.

More On the C.ITOH Prowriter 8510

Dear HUG,

Because I need a printer capable of Greek letters, subscripts, enhanced printing, and underlining, I recently purchased a C.ITOH Prowriter 8510 which I have now interfaced to my H89. I hope that the solutions I have used will aid others who may be following a similar course.

I connected the printer to the H89 with a cable, available from Inmac, interconnecting pins 1, 2, 3, 7, 8, and 20. Pin 5 of the H89 must not be connected or data will not be transmitted.

I used the HDOS LPH24.DVD device driver modified as suggested by Ted Andrews (REMark #18, page 20) to accept XON/XOFF protocol. Additionally, the printer initialization sequence in INITLP was modified as follows to stop the printing of several stray unrecognized H24 escape characters.

Old Code

```
        LDA     TLP.LPI
        CPI     6
        MVI     A,'4'
        JZ      LPI0
        MVI     A,'5'
LPIO    STA     INIA+1
        LDA     TLP.LEN
        STA     INIB+2
        LXI     H,INIA
INI1    MOV     A,M
        CPI     377Q
        RZ
        CALL    OUTCHAR
        INX     H
        JMP     INI1
        INIA    DB      ESC,0
        INIB    DB      ESC,'2',0,CR
                DB      377Q
```

New Code

```
        LDA     TLP.LPI
        CPI     6
        MVI     A,'A'
        JZ      LPI0
        MVI     A,'B'
LPI0    STA     INIA+1


        LXI     H,INIA
INI1    MOV     A,M
        CPI     377Q
        RZ
        CALL    OUTCHAR
        INX     H
        JMP     INI1
INIA    DB      ESC,0
INIB    DB      CR
        DB      377Q
```

The Prowriter allows printing of the Greek, superscript, and graphics characters in an eight bit mode or in a seven bit mode by changing to these character sets using an escape sequence. In the seven bit mode, when either of my text formatters attempted to place Greek characters or the in-line superscripts in a line containing underlines, Greek a's were superimposed on the character. This seems to occur because the line is formed in two passes and the printer has the idiosyncrasy that is coded by 'space+200Q'. The solution was to set the Prowriter DIP switches for eight bit operation and to ignore parity, thus allowing the parity bit to carry the graphics information.

The DIP switch settings are as follows:

	POSITION							
	8	7	6	5	4	3	2	1
Switch 1	O	X	O	O	O	O	X	O
Switch 2	O	X	O	O	O	O	O	O
Switch 21	X	O	O	X	X	O	O	O
Switch 22					O	X	O	O
Switch 23			O	X	O	O	O	X
Switch 24	X	O	O	X	X	O	X	O

X=Closed
O=Open

Set up in this way, the printer fails to line feed using LPH24.DVD because the subroutine LPOUTCH outputs a CR followed by a 'LF+200Q' on receipt of the new line character to stop recursion in some cases. The latter is the code for a broad vertical bar which appears in the first over-printed column.

My solution was to truncate subroutine LPOUTCH to:

```
CALL    OUTCHAR
RET
```

With this solution, all characters are printed, but the printer driver does not handle line length or page length, a modest trade-off since the printer handles these itself.

G. Edwin Wilson, Jr.
41 Hitchner Ave.
Bridgeton, NJ 08302

A Typing Error In The Book "The C Primer"

Dear HUG,

Readers who are following the illustrative examples in the book "The C Primer" by Les Hancock and Morris Krieger should be aware of a typing error in the palindromia excercise on page 145. The statement

char *strptr2 = strptr;

should not include the asterisk. That is, the correct statement should read:

char strptr2 = strptr;

Vectored to 50 ☞

DUMP YOUR Z100!

PRINTMATE 150
Welcome to MPI's World of Graphics!
Logic Seeking 150 Character Per Second
PrintMate 150
mpi

Dump your Z100 screen image, that is, onto an MPI Printmate 150g printer! This American made high-quality printer can output any ZDOS graphics image displayed on your Z100 color or monochrome monitor, when used with the MPI Artisan software. Both high-resolution graphics and textual material can be printed out simultaneously. Different colors are represented by various shades of gray on your paper. And, you also may design custom type fonts to use during normal printouts. They can range from mathematical symbols to Hebrew, or script. Several pre-defined fonts are already included with the Artisan software.

For Z100, Z89 and Z90 CP/M users, it's simple to interface for standard text output or user-defined graphic routines. You have the ability to design and print with user-defined type fonts, however screen dumps are not currently supported under CP/M.

The 150g prints at a speed of 150 characters per second in a bidirectional logic-seeking mode using high-speed tabs for maximum throughput. Character densities, baud rates, form lengths and a variety of other programmable features may be selected from the optional keypad.

Standard print formats include an 11x9 correspondence quality density, as well as 10, 12, 15 and 17 cpi (231 columns wide). Double-wide characters, underlining, upper and lower case characters with true descenders, and dot-addressable graphics are all standard.

If a more compact printer is desirable, we also carry the Printmate 99g. It has many of the above features including Z100 screen dumps, and handles both tractor and friction feed with serial and parallel interfaces standard!

Before purchasing any printer, give us a call and request our fully illustrated catalog on Zenith products and peripherals, and a special PRINTMATE-PAC of color brochures with sample printouts. Printmates start from under $500.

Our prices and service are just plain tough to beat!

Studio Computers inc.

999 South Adams
Birmingham, MI 48011
Phone (313)-645-5365

Also, users of the MX-80 printer with CP/M who are cranking up their MBASIC to change the print mode using an LPRINT CHR$(mode) statement will benefit from the following brief assembly language routines.

To set or reset the compressed mode, the basic form is:

```
org     100h
mvi     e,mode   ; mode = 15 sets compressed, 18 resets.
mvi     c,5
call    5
```

I named the Set Compressed program SC, its reset RC. To enter those modes that employ escapes, such as the emphasized, which requires ESC E, insert the following lines immediately below the org declaration.

```
mvi     e,27
mvi     c,5
call    5
```

To set the emphasized mode, enter 'E' for mode; to return from the emphasized, enter 'F'. I named the Set Emphasized program SE; its return RE.

These routines are real time savers as well as reducing disk drive wear and tear (disk on time with these is ever so much shorter!). While the mode values given apply to the Epson printer, the routines should be applicable to any printer that responds to LPRINT instructions for mode changes.

Clem Pepper
3270-96 Caminito East Bluff
La Jolla, CA 92037

A Spiffy Little Program

Dear HUG,

```
1 'HERE IS A SPIFFY LITTLE PROGRAM THAT WILL TURN YOUR $2000 TOY INTO A
2 '$4.98 TOY.  IT WAS ORIGINALLY INTENDED TO BE USED TO BALANCE THE CHECK
3 'BOOK, AND IT DOES THAT NICELY.  THE RIBBONS AND BOWS WERE ADDED LATER.
4 'THIS COULD BE RENUMBERED AND USED AS A SUBROUTINE FOR ANOTHER PROGRAM
5 'WITH A RETURN ADDED, AND THE 25TH LINE DISABLED AFTER RETURN.  THE BLUE
6 'KEY WAS LEFT OPEN AS A LOCAL OPTION.
10 PRINT CHR$(27)+"x1";CHR$(27)+"Y8$";
20 PRINT "  ADD     SUB     MULT    DVD     EXP    CLEAR          1/Y     SWAP"
30 PRINT CHR$(27)+"H";CHR$(27)+"E":
40 INPUT "IS DOUBLE PRECISION REQUIRED (Y/N) ";O$
50 IF O$=CHR$(89) THEN 60 ELSE 70
60 DEFDBL X,Y
70 INFUT "ENTER FIRST NUMBER  ",X
80 PRINT
90 INPUT"NEXT NUMBER  ",Y
100 PRINT:PRINT"FUNCTION  ";
110 Z$=INPUT$(2)
120 IF Z$=CHR$(27)+"S" THEN X=X+Y:PRINT"X+Y":GOTO 200
130 IF Z$=CHR$(27)+"T" THEN X=X-Y:PRINT"X-Y":GOTO 200
140 IF Z$=CHR$(27)+"U" THEN X=X*Y:PRINT"X*Y":GOTO 200
150 IF Z$=CHR$(27)+"V" THEN X=X/Y:PRINT"X/Y":GOTO 200
160 IF Z$=CHR$(27)+"Q" THEN Y=1/Y:X=X^Y:PRINT"X^(1/Y)":GOTO 200
170 IF Z$=CHR$(27)+"W" THEN X=X^Y:PRINT"X^Y":GOTO 200
180 IF Z$=CHR$(27)+"R" THEN SWAP X,Y:PRINT"X=Y,Y=X":GOTO 100
190 IF Z$=CHR$(27)+"J" THEN X=0:PRINT"X=0":GOTO 70'CLEAR
200 PRINT:PRINT "ANSWER IS ";X
210 PRINT:GOTO 90
```

You have my permission to use the above material as you see fit.

R. C. Perkins
PO Box 114
De Bary, FL 32713

A Short Program In ZBASIC

Dear HUG,

I have just finished a short program in ZBASIC and would like to share it will all the other HUGgies.

```
10 'THIS PROGRAM PRINTS OUT WHATEVER IS ON THE SCREEN,
       JUST PRESS <F8>.
20 'THIS PROGRAM WAS WRITTEN ON THE H-100 FOR THE C.ITOH
       PROWRITER PRINTER.
30 'IT ONLY WORKS ON THE GREEN PLANE OF VIDEO RAM SO
       COLOR IS NOT SUPPORTED.
40 'IT SHOULD WORK ON MOST OTHER PRINTERS IF THEY USE
       BIT IMAGE GRAPHICS.
50 'YOU MIGHT HAVE TO CHANGE SOME OF THE PRINT COMMANDS
       TO USE OTHER PRINTERS.
60 'THE PROGRAM STARTS AT THE TOP RIGHT SIDE OF THE
       SCREEN AND WORKS DOWN.
70 'EACH PIT IS TURNED INTO A CHARACTER CODE AND JOINED WITH P$.
80 'WHEN ONE COLUMN IS FINISHED (255 BYTES) THEN IT IS PRINTED OUT
90 'AND THE NEXT COLUMN IS DONE.
100 DIM P(224) : R=0 : DEF SEG = &HE000 : KEY(8) ON:
       ON KEY(8) GOSUB 150
110 'CALULATE STARTING POINT FOR EACH LINE
120 FOR J=0 TO 392 STEP 16:FOR I =0 TO 8:P(R)=(I+J)*128:
       R=R+1:NEXT I,J
130 'PUT YOUR PROGRAM HERE AND PRESS <F8> WHEN READY TO PRINT
140 END
150 'SET TO ENLARGE PRINT AND VERTICAL LINE FEED TO 16/144 INCHES
160 LPRINT CHR$(14)+CHR$(27)+"T16";
170 'CREATE P$ EQUAL TO ONE LINE OF GRAPHICS
180 FOR J=79 TO 0 STEP -1:P$="":FOR I=0 TO 244:
       P$=P$+CHR$(PEEK(P(I)+J)):NEXT I
190 'SET TO GRAPHIC MODE AND PRINT 255 GRAPHIC CHARACTERS
       (LENGHT OF P$)
200 LPRINT CHR$(27)+"S0225"+P$:NEXT J:RETURN
```

Kenneth Terry
4572 Dolly Ridge Road
Birmingham, AL 35243

Another Parallel Board

Dear Hug,

I want to respond to a couple of the items in the October REMark but first I would like to thank the Heath Users' Group organization for putting on a fine conference in August. I attended both the first and second conferences and thoroughly enjoyed them both. I enjoyed all of Mr. Johnson's remarks, particularly the history of the Heath Company, which he gave in 1982. Mr. Johnson's review of Heath customer demographics was quite interesting and really hit home. Anyway, thanks to all the HUG staff for the hard work it took to put on a really good conference.

With regard to the evaluation of parallel interface boards discussed in the October issue, I wonder why Mr. Morin did not consider the FBE H89UTI board. I have recently purchased two of these boards and was impressed by their high quality. They provide a parallel interface, two serial interfaces, and a real time clock, as well as a socket for the high speed arithmetic chip. The FBE board seems to be competitively priced with the other interface boards offered at the HUGCON, and will instantly fall into a stock '89. Mr. Dave Brockman of FBE was very helpful in assisting with the minor difficulties in getting this board interfaced with my H-1000 CPU board.

With regard to the question on plotters asked by Mr. Robinson, I have recently integrated a Houston Instruments DMP-4 plotter with my H-89 and obtained low cost software capable of producing high quality transparencies, among other things. The Radio Shack multipen plotter is nothing more than the DMP-4 in Tandy Corporation colors. This plotter can be interfaced in a variety of ways, including RS232 serial interface, but the serial mode is best operated with the Xon/Xoff protocol. A parallel interface board is also available. Houston Instruments sponsors a Micrographic Users' Group (MUG) which offers free listings of plotter software. Some plotter programs are available on disk or tape for a nominal media charge. In particular, the Higraph program is a generic CP/M software package which will produce a variety of charts and transparencies using the DMP series plotters. There are other more expensive plotting software packages around, but those distributed by the Micrographic Users' Group should give Mr. Robinson a good start on making a few bucks.

Again, thanks for all the user support.

Charles Smith
8048 El Manor
Los Angeles, CA 90045

Dear HUG,

I enjoy reading REMark, which comes to me at half the price of BYTE, and also comes by air in an envelope (as opposed to BYTE's brown wrapper and sea mail), but of course REMark is a little smaller. I am writing in response to a problem I noted with Frank Clark's "Changing Your Tone" article in REMark Vol. 38. I finally got the program up and running (I had the same problems everyone else had reading it), and found it would not work on Port 320 but would on the other two. I had had a defective device connected to that port, and I was ready to pull the line driver chips when I noticed that the instructions said to connect to pins 1 and 3. Pin 1 is the protective ground and pin 7 is the signal ground. Wiring the output to pins 3 and 7 allows me to use port 320. I don't know if anyone else has had that problem, but apparently pins 1 and 7 are not always connected together.

Hope this clears things up for anyone else with this 'bug'.

Jonathan E. Yuen
P.O. Box 42, Shanhua, Tainan, 741,
Taiwan, Republic of China

---

2 New Programs From Computer Consultants To Business

Dear Walt,

I just received two pieces of software yesterday from Computer Consultants to Business (George Sellers) in Cumberland, MD. I bought their disk transfer programs (KZ90 and ZK90) that allow reading and writing of KAYPRO II disks. I am writing to let everyone in the Heath/Zenith Community know that these two programs ($20.00 plus $1.00 postage each) do indeed work and are well worth the money if one needs to transfer files between a H/Z-89 microcomputer and a KAYPRO portable. The programs are restricted for use with soft-sectored disk-controllers, and only for single-sided 48 TPI double-density disk drives.

Everyone should realize that these programs are very basic. In fact, each is only 2K long, so one should not expect too many features with them. All one can do with these programs is determine what is on the KAYPRO directory and transfer files between the H/Z-89 and the KAYPRO. In the directory listing, there is no indication whatsoever of sizes of the files there already or how much room remains on the KAYPRO disk for new files — the user takes his chances in trying to transfer files over to the KAYPRO disk which may not have enough room to accomodate the additional files. Also the programs do not allow wildcard transfer of files from one format to the other. This can be very inconvenient if one has many files to transfer, because each file must be separately transferred. The only other inconvenience I encountered in using these programs was confusing which one to use to transfer files. It would be much nicer if these two programs were combined into a single program which could either automatically determine the format of the disk in each drive, or prompt the user to input that information. From then on, the program would control the formats being read from or written to.

I hope my experiences with these two programs will help some people determine their usefulness in their applications. Please include this information in REMark if applicable.

Victor M. Felarca
610 Clemson Drive
Warner Robins, GA 31093

THIS IS YOUR LAST CHANCE

to vote for your
favorite article
published in 1983

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 ==

1983 ARTICLE OF THE YEAR
— BALLOT —

My Choice for "Article of the Year" is:

____

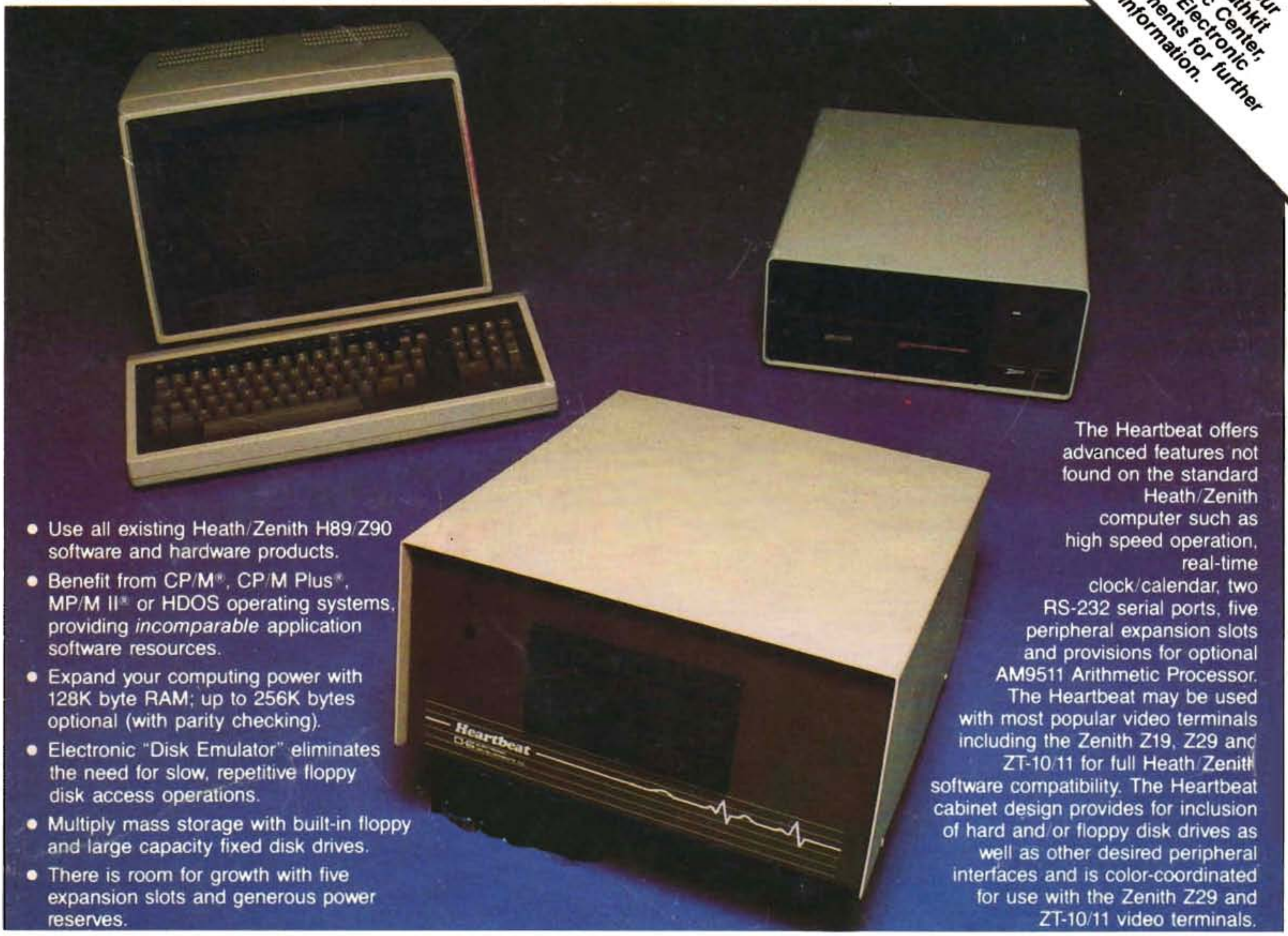
Author: ______________________ Issue: __________ Page: ________

The winner of the "1983 Article of the Year" Award
will be announced in the March, 1984 issue of REMark.

NOTE: Your ballot must be received at the HUG office before FEB. 1, 1984 to be counted in the judging.

Mail your ballot to:

Article of the Year
Heath/Zenith Users' Group
Hilltop Road
St. Joseph, Michigan 49085



Heath/Zenith Users' Group

Hilltop Road
Saint Joseph, Michigan 49085

Volume 4, Issue 12

POSTMASTER: If undeliverable, please do not return.

885-2047