
IT'S HERE!!
HRUN
SEE PAGE 15
$2.50
REMark
Issue 37 • February 1983
XYZZY
Y2
FEE FIE FOE FOO
CONGRATULATIONS ON BRINGING LIGHT INTO THE DARK ROOM!
CAVE UNDER CONSTRUCTION
PLOVER
STOP PAY TROLL
CAUTION BEAR IN ROOM
TREASURE VAULT
DO NOT DISTURB THE DWARVES
Official magazine for users of
HEATH
computer equipment

BE A HAPPY HUGGER.

Get to know Floppy Disk Services.

We are contracted dealers for Siemens, Shugart and Tandon disk drives. You may do a double take when you see our prices, but there's no catch—we simply buy in large quantities and sell at reasonable prices.

We've sold thousands of drives to our Heath/Zenith friends. And with our good service and fair guarantee, we keep our friends friendly.

Check our prices on these Heath/Zenith compatible drives.

FDD-100-5b 'flippy' (Just like your Heath, but 'flippy') $215.00
FDD-221-5 DS/DD 96tpi (80 track) same as TM-100-4 ..... 330.00
FDD-200-5 same as 100-5 but double sided (2 heads) 250.00
FDD-111-5 5 milisecond SS/DD 48tpi same as TM-100-1 ..... 210.00
FDD-211-5 5 milisecond DS/DD same as TM-100-2 ..... 265.00
FDD-100-8d5 SS/DD 8 inch 3 milisecond step! 275.00
FDD-200-8p5 DS/DD 8 inch 3 milisecond step! 395.00
Look for our Half Height Drive Specials Coming Soon.

PAYMENT POLICY We accept Mastercard, Visa, personal checks & M.O. We reserve the right to wait 10 working days for personal checks to clear your bank. All shipping standard UPS rates unless otherwise requested. New Jersey residents *must* add 5% sales tax.

Due to production deadlines, prices in this ad could be as old as 2 months. If in doubt, call!

Prices and specs subject to change without notice.

Need an enclosure for 5¼ drives?

Single vertical case w/ps A&T styled to match Heath color ..... $ 50.00
Dual vertical case with power supply A&T 75.00
Dual horizontal case with dual floppy supply A&T 125.00

. . . or an enclosure for 8 inch drives?

Single 8 inch horizontal case w/ps A&T for 1 standard or 2 half height ..... Call
Dual horizontal case with commercial grade supply A&T $275.00
Dual horizontal case metal only, w/power switch, and hardware 115.00

Magnolia 8" Controller

Magnolia Microsystems controller for the Heath H88, 89, 90 enables you to use any combination of 8 or 5¼ single or double sided, single or double density—up to 4 of each size drive! You even get CP/M in 8 and 5¼ format serialized for you. This controller is available at a special price of $525.00.

Our pledge to you . . .

All Floppy Disk products are brand new and 100% warranted. If you are unhappy with our systems or components for any reason, we will refund your money promptly. There are two requirements: 1. the equipment must be in as good shape as you received it; and 2. you must call or write for a return authorization. This offer is good for 30 days, beginning with your invoice date. Feel free to call us with any questions.

Repairs, too.

We repair drives of all types. Call for details.

We make Huggers happy. Call today! (609) 799-4440

Dual Heath Add-on FDD-100-5

Dual Half Height or Single Full Size 8"

FLOPPY DISK SERVICES INC.

741 Alexander Rd.
Princeton, NJ 08540
(609) 799-4440

CP/M and H88-89-90 are registered trademarks of Digital Research and Heath Co., respectively.

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

REMark Editor . . . . . Walt Gillespie
Assistant Editor . . . . . Nancy Strunk

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

REMark

Issue 37 • February 1983

on the stack

ON THE COVER: Reprinted by permission of Edmund J. Florimont of Greyhound Electronics (NJ) is a painting by famous American artist Susan Heinz of the wonderful world of Adventure. A limited quantity of 30"x40" full color poster reproductions (numbered and signed by the artist) are available from the Heathkit Electronics Center, 1013 State Hwy. 35, Ocean, NJ 07712, (201) 775-1231. Price $39.95 plus shipping.

The BACKPLATE

a neater solution from KRES

Are you tired of having wires and cables stuffed out the holes of your computer when they should be mounted? The KRES Backplate is an upgrade replacement for the existing rear panel on your H/Z89-90. With this KRES solution, you should be able to run every cable you will ever need out the back of your computer cleanly and neatly.

$35.00

OTHER KRES PRODUCTS

Seven Slot Expansion $395.00

Programmable 2/4 MHz $79.95

Color Graphics

Music Board $125.00

Prototyping Cards $39.95

Joysticks $69.95

Supplemental Fan and Bracket $49.95

Power Supply - 5 volts at 10 amps

Software - wide selection

KRES ENGINEERING
P.O. Box 17328, Irvine, CA 92713
(714) 559-1047
or (213) 957-6322
Bulletin Board: (714) 559-8579

All prices FOB Irvine, CA
California Residents add 6%

Bob Ellerton
HUG Manager

Gearing Up For The Second National HUG Conference ....

Back in August of last year the Heath Users' Group was fortunate enough to host the First National HUG Conference. The Conference or central gathering of HUGgies from all over the country was an idea formed in the heads of several users with the major responsibility being placed on the shoulders of one Bill Parrott. Bill contacted several individuals within the Heath/Zenith organization suggesting the idea of a National HUG gathering. From there, the Heath Users' Group became the central point for organizing the event that took place at the O'Hare Hyatt Regency in Chicago, Illinois. The success of the event was measured by the fact that most of the 1000 attendees were ready to do it again this year.

As the dust settles from the events of 1982, HUG is beginning to lay the ground work for the Second National HUG Conference. We have received excellent feedback from those who attended the First Conference. From those comments we will be working toward creating an even better schedule (if possible) for 1983. If you attended, or if you think you might like to attend this year, please feel free to send your suggestions about the type of material you would like to see discussed to the Heath Users' Group. For the Local HUG Clubs .... how about a meeting aimed at the "direction" of the Second National HUG Conference? Any help in the selection of the topics for discussion would be appreciated.

Within the next month, REMark will begin carrying information which will include tentative schedules, speakers, registration information and exhibits as final date approaches. We have again arranged to meet at the O'Hare Hyatt Regency in Chicago, Illinois. The date of the Second National HUG Conference is now set for the 19th, 20th and 21st of August, 1983. So far, we can tell you that we have reserved much more space for the "second attempt" with several smaller meeting rooms available for those of you with special interests or for those of you who want to relax with the "computer widows" at the NUA (Non-User-Attendee) Lounge. Also, the size of the Vendor Exhibit Area has been nearly tripled so that each attendee will have the best opportunity to see that special product for your computer system.

As we proceed toward the Second National HUG Conference, the staff at HUG would sincerely welcome any comments that you might have that would help us in making this event more enjoyable for you or members of your family who may be attending. If you should care to volunteer your services while at the Conference, please feel free to contact us so that we can get in touch with you should the need arise.

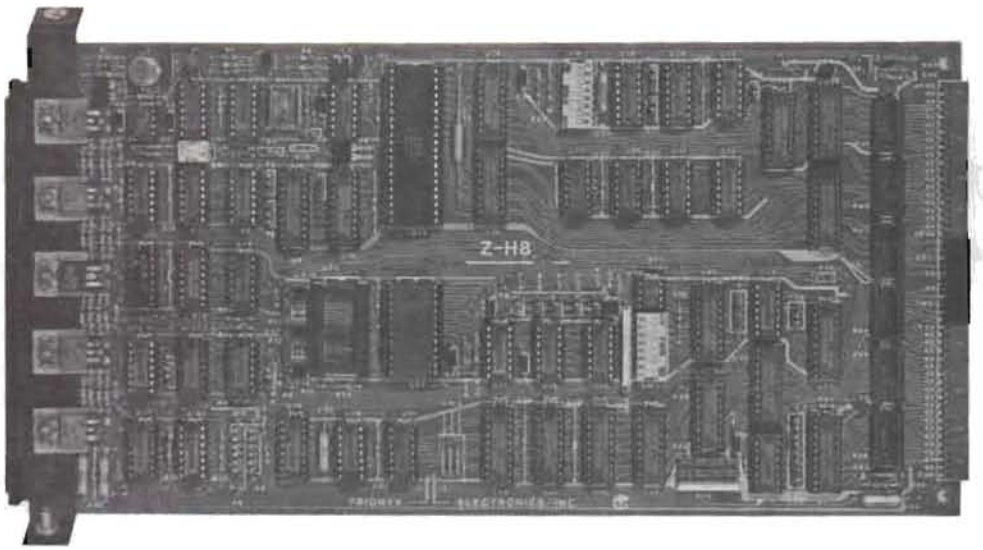
We would like to take this opportunity to thank each of you who attended the First National HUG Conference for your support and confidence in the Heath Users' Group. For those of you who missed the First Conference, we welcome you to participate in what is hoped to be an even bigger and better chance to get to know fellow HUGgies from all over the United States and the World.

Watch for the Official Registration Form in coming Issues of REMark along with additional details as they become available.

Bob Ellerton, Manager
Heath Users' Group

Z80 CPU BOARD for the H8*

* H8 is a Registered Trademark of the Heath Company



Model Z-H8

$ 300.00 Assembled $ 250.00 Kit

Programmable Clock Rate — 2 MHz or 4 MHz Operation

Clock Rate under Software Control — May Be Changed at Any Time

Selection of Wait States (None, 1/2, 1 or 2) at 4 MHz

Buss Termination Network • 51 Integrated Circuits • Transistor Power Supply

Includes Heath Extended Configuration (ORG ZERO) Circuits

Exclusive Z80 Front Panel Monitor ROM — Based Upon Heath XCON-8 ROM

Exclusive Software Speed Utilities (For Both HDOS* and CP/M**) on Diskette

---

Fully Compatable With All Heath H8* Hardware and Software

* H8 and HDOS are Registered Trademarks of the Heath Company.
** CP/M is a Registered Trademark of Digital Research.

Check • Money Order • VISA • MASTERCARD • C.O.D.

Phone Orders Welcome (714) 830-2092 – Send For Free Brochure

TRIONYX ELECTRONICS, INC.

P.O. BOX 5131, SANTA ANA, CA 92704

BUGGIN' HUG

Single Page Writer

ED:) The following letter was addressed to Karl Romer concerning his article "Single Page Writer for the MX80" which apeared in the December issue of REMark, a carbon copy was forwarded to HUG.

Dear Karl:

The article by you in the Dec 82 issue of RE-Mark was great. The program (SWP) is the most practical and useful information that I have ever gotten from the magazine. Thanks for sharing it with me and others. No doubt you have similar work that should be published and I anxiously look forward to it.

Do you have a program similar to SPW that will Print 132 characters on a line in compressed mode on the MX80 FT?

Sincerely,
F.O. Cornay
13203 NE 54th Place
Bellevue, WA 98005

Dear HUG:

I'd like to add a few comments to Karl Romer's article "Single Page Writer for the MX80" in REMark #35.

First, if you don't like paper shims or fish line, the "paper out" switch on the MX80-FT can be disabled under software control by sending ESC 8 to the printer. For example:
LPRINT CHR$(27)+"8" [MBASIC], or
OUT27,56 [Magic Wand].
(In the second case, note that 8 = ASCII 56 - it took me a while to figure out why OUT27,8 was not doing the job). ESC 9 will restore operation of the switch, as will turning off power to the printer. The comment on not overrunning the end of the page is a good one - I usually print a draft copy on fanfold paper to check the length.

Second, regarding the paper hitting the bail, I have been quite successful by leaving the bail and the tractor covers open, and letting the paper come straight up. I'll have to try Karl's paper-curling method.

I hope that this is of some help to MX80-FT users.

Very truly yours,
Robert B. Currie
57 Aronimink Place
Macungie, PA 18062

Vectored to 51 ☞

QUESTIONS & ANSWERS

(EDITOR'S NOTE: If you need answers to specific questions on software or hardware problems please drop us a note, Questions & Answers, Heath Users' Group, Hilltop Road, St. Joseph, MI 49085. Please keep your questions brief and to the point. We will do our best to answer you here in this column in future issues.)

Q. Can the H89 be used to do high resolution graphics?

A. Yes, but it requires a special circuit board which is available through the independent vendor, Cleveland Codonics, Inc., P.O. Box 45259 Cleveland, OH 44145. The board will access high resolution 504H by 247V intelligent graphics. Without this board, the H89 (H19) is limited to the defined graphics of the graphic mode.

Q. I have CP/M and when I looked at the DIRectory on my new MICROSOFT BASIC Interpreter, I noticed the file OBASIC.COM. What is OBASIC?

A. OBASIC is version 4.83 of Microsoft BASIC. MBASIC is version 5.21. The OBASIC version has been included because some programs written in the older version may not function correctly under MBASIC 5.21.

Q. I have the kit version H-89A computer. Why does the message, "Type SPACES to determine baud rate" appear when I try to boot CP/M version 2.2.03? [Other symptoms of this problem are: 1) CONFIGUR fails unpredictably to properly recognize hardware peripherals, and 2) at bootup time the message, "PUT DISK P IN DRIVE P" appears.]

A. The MTR-88 monitor ROM of the H-89A, part #444-40, located at U518 on the CPU board; and I/O mapping ROM, part #444,43, located at U550 also on the CPU board, should be replaced by part #'s 444-62 and 444-61, respectively. If you have this problem, the parts above plus the installation guide, part #595-2547, can be obtained free of charge by calling the Heath Parts Replacement Department at 616/982-3571. THIS WILL NOT BE NECESSARY IF YOU PLAN TO UPDATE YOUR UNIT TO RUN WITH MTR-90. (MTR-90 is supplied with the Z-89/Z-37 and the Z-89/Z-67.)

Q. When I tried to save my CP/M MBASIC program, I received the error "BDOS error on A: R/O" and lost my program. The disk was not write protected. How do I use more than one disk when I have only one drive?

A. If a disk is removed and replaced with another disk, the change is flagged by CP/M as R/O until CP/M has been properly informed of the change. Even with one physical drive you can still use all three logical drives: "A:", "B:", and "C:". CP/M keeps track of up to three disks for you. To inform CP/M of a change in physical disks, type CTRL C. When under MBASIC, use the RESET command.

Q. I have the Okidata Microline 82A. My question is, can I use the MX-80 device driver instead of the H-14 device driver I am now using, to use the u-82A graphics?

A. We don't really know. It is impossible for us to know the configuration for all printers of which most we have very little, if any, information to verify such possibilities. We can offer a hint or two; 1) consult the manufacture or check in the manual and find if it specifies a particular device driver use with Heath computers, 2) compare the type of hardware or software handshaking it has to existing device drivers, and finally, 3) the SOFTSHOP, 35 Shadow Oak Dr., Subbury, MA 01776, has a Universal Device Driver (UD.DVD) available which works with most any printer device. Contact Jim Teixeira for details.

CHEAPCALC

```
10  '  ************************
20  '  *                      *
30  '  * ELECTRONIC WORKSHEET  *
40  '  *  WRITTEN FOR THE     *
50  '  *   APPLE II BY        *
60  '  *  WILLIAM V R SMITH   *
70  '  *                      *
80  '  * MODIFIED FOR MBASIC   *
90  '  *   BOB MCFARLAND      *
100 '  *    11/6/82           *
110 '  *                      *
120 '  ************************
```

In the July, 1982 issue of the Softalk magazine, a BASIC spread sheet program written by W.V.R. Smith called BASICALC was described for the APPLE computer. This program has been modified and enlarged to run under MBASIC on the H-89 or H-8 with a H-19 terminal. It will not run properly on an H-8 which does not have a screen addressable terminal that follows the H-19 conventions.

Many people have expressed an interest in a spread sheet program for the H-89 but have not been ready to purchase either SUPERCALC or ZENCALC. This program can familiarize the user with a spread sheet program so that a rational decision can be made about purchasing the comercial program, and should be adequate for the needs of most users.

A MBASIC spread sheet program has to be relatively small compared to the commercial programs (15 columns by 40 rows) but is still large enough for the user to become familiar with the input requirements and capabilities for this type of program. The MBASIC program presented here is limited to simple mathematical operations as described below but is also easily modifiable to add special purpose routines which could be cumbersome to accomplish with either SUPERCALC or ZENCALC.

Spread sheet programs all use a common way to define program variables, which are limited to those definable as a 2 dimensional array. The indices of the variables correspond to the row and column of the spread

```
130  CLEAR 10000: MC%=15:MR%=40:DIM A$(MR%,MC%),B$(MR%,MC%),CW(MC%),YX%(MC%)
140 SY%= 1:XM% = 1:SX%= 1:WIDTH 255
150 BP$=CHR$(7)
160 E$=CHR$(27):EH$=E$+"H":'home cursor
170 CS$=E$+"E":' CLEAR SCREEN
180 CL$=E$+"K":' CLEAR TO EOL
190 IV$=E$+"p":' inverse video
200 NV$=E$+"q":' normal video
210 EN$=E$+"x1":' enable 25th line
220 DEF FN PC$(R%,C%)=E$+"Y"+CHR$(R%+31)+CHR$(C%+31):'direct screen addressing
230 S$="                              "
240 DS$=E$+"y1":'disable 25th line
250 PRINT CS$;EN$;FN PC$(25,1);CL$
260 PRINT  EN$;FN PC$(25,25);"Mbasic Basicalc Program Version 1.0";EH$
270  FOR X%= 1 TO MC%:YX%(X%)=1:CW(X%) = 9: NEXT:OF%=1
280 S$ = "                              "
290 T$ = "ABCDEFGHIJKLMNOP"
300 T1$ = "****************"
310  GOTO 2050
320  '
330  '  * VARABLE PARCER  *
340  '
350 L%=  LEN (A$(Y%,X%)):F = 2:A1 = 0:A2 = 0:P%= 1:H$ = ""
360  IF L%= 0 THEN 780
370  IF P%> L% THEN 780
380  GOSUB 1030
390 IF C%> 64 THEN  GOSUB 860: IF P%> L% THEN  RETURN:'character input
400  IF C%= 46 THEN 470:'decimal point
410  IF C%> 41 AND C%< 48 THEN GOSUB 680:F = C%- 41: GOTO 370:'operands
420 IF C%=94 THEN GOSUB 680:F=7:GOTO 370
430  IF C%= 38 THEN 1290:'& operator
440  IF C%> 47 AND C%< 58 THEN 470:'numbers
450  IF C%= 58 THEN 1470:': sets output format
460  GOTO 780
```

MBCALC
A Spread Sheet Program in Mbasic

B.L.McFarland
17175 Gunther
Granada Hills, CA 91344

```
470 H$ = H$ +  CHR$ (C%): IF P%> L% THEN  GOSUB 680: GOTO 1080
480 GOSUB 1030:GOTO 390
490  '
500  ' * INPUT STATEMENT **
510  '
520 IV=1:I$=A$:PRINT A$;:PRINT FN PC$(2,1);
530  A$=INPUT$(1):A=ASC(A$):IF A=27 THEN 2160
540 IF A<>8 THEN 580:'routine to handle backspace
550 L%=  LEN (I$):I$ = MID$ (" " + I$,2,L% - 1):PRINT FN PC$(2,1);I$;
560 IF LEN(I$)=0 THEN A$="":RETURN
570  GOTO 530
580  IF A  = 21 THEN 630:'ctrl-u
590  IF A  = 13  THEN 640:'ctrl-M(CR)
600  IF  A< 31 THEN 530:'any other ctrl character is ignored
610  IF A  = 34  THEN GOTO 530:'" mark for string input
620 I$ = I$ + A$:PRINT FN PC$(2,1);I$;:IV=LEN(I$):GOTO 530
630 A$ =  MID$ (A$(Y%,X%),IV+1,1): GOTO 620
640 A$=I$:RETURN
650  '
660  ' * PERFORM MATH FUNCTION
670  '
680 A2 =  VAL (H$):H$ = "":'math function subroutine
690 F1 = F:F = 2
700  ON F1 GOSUB 720,730,710,740,710,760,750
710  RETURN
720 A1 = A1 * A2: RETURN
730 A1 = A1 + A2: RETURN
740 A1 = A1 - A2: RETURN
750 A1 = A1 ^ A2: RETURN
760  IF A2 <  > 0 THEN A1 = A1 / A2
770  RETURN
780 H$ =  MID$ (A$(Y%,X%),1,L%) + S$
790 H$ =  LEFT$ (H$,CW(X%))
800 B$(Y%,X%) = H$
810  GOSUB 1230: '   * XM% AND YX% TEST
820  RETURN
830  '
840  ' * FIND MATH VALUE OF  SCREEN
850  '
860 X3%=C%-64:IF C%=94 THEN RETURN
870  IF X3% >MC% THEN  GOSUB 780: RETURN
880 H$ = "": IF L%= 1 THEN 780
890  GOSUB 1030: IF C%< 48 OR C%> 57 THEN  GOTO 780
900  GOTO 920
910  GOSUB 1030
920  IF C%< 48 OR C%> 57 THEN 960
```

Vectored ☞

sheet displayed cell. A letter index is commonly used to define the column location of the variable and a numerical index defines the row location in the spread sheet. Separate pages are used for the input and output definitions and in some cases an additional page is used for individual definitions of the cells. The MBCALC program uses the letters A-P for column defintions and the numbers from 1-40 for row definitions. The input array contains all user instructions for the problem. These instructions are executed to produce an output array which is displayed on the screen and can be printed on a hard copy device. Only the output array is limited in column width to that specifed in the problem. Limitations in the length of the input array were avoided in MBCALC by displaying the current cell contents at the top of the screen in line 2 of the CRT. Of course the output array limitations restrict the length of usable text labels in the input array to the output array column widths since this is all that will be displayed in the cell location.

The MBCALC program listed in this article runs under HDOS, but is easily converted to CP/M operation by deleting lines 3630,3660 & 3690 plus deleting the CLEAR statement on line 100, and the REM tokens on lines 3670, and 3700. The MBCALC program can be compiled with the Microsoft BASIC compiler to obtain faster operation in CP/M, which is needed since the operation of the special keys seems to be slower under CP/M than HDOS. As with most BASIC programs, the comment lines should be removed with a text editor to speed operation of the code if it is not compiled.

PROGRAM FEATURES

Several novel programming ideas are used in MBCALC that are worth highlighting. In particular MBCALC uses the INPUT$ function for single character input which enables separation of the user definable keys from the normal input keys in a straight forward manner. The decoding of text strings to produce usable BASIC commands is not used by many programs but is a very useful trick. The input decoding (lines 520-640 and 2150-2280) and the replication routine (lines

3780-3880) both use BASIC string manipulation to define the program operations. Only the arrow keys are used as two character input (lines 2150-2200) leaving the remaining special purpose keys available for user definition by a simple test after line 2330.

PROGRAM OPERATIONS

MBCALC variables are limited to a single capital letter (A-P) followed by a numeric index that can vary from 1 to 40 (e.g. A1,B33,C39 are allowed while AA1 and C41 are illegal). Numerical constants can also be used in the mathematical operations described below.

Movement through the cells displayed on the screen is accomplished with the arrow keys (shift-keypad) or by the command mode discussed below which moves the entire screen display area to another part of the spread sheet.

The displayed cells can contain constants, operations, or text as selected by the user.

MBCALC operations can consist of combinations of the following mathematical operations:

+,-,*, & / are all usable operations with the usual definitions, but parentheses can not be included [12+A1/15-B4 is allowed but 3*(A1+B2) cannot be used]. The mathematical operation does not have to fit in the cell width since only the result of the operation is displayed in the output array.

Editing of the input cells is accomplished on the 2nd line of the CRT where the input cell parameters are displayed. The user must enter the first character of the input cell, but then can copy the characters from the screen by using CTRL-U to move the cursor over the screen characters. The backspace & delete keys both delete characters.

Certain characters are reserved for commands to MBCALC. @,:,!,/ and & are used in the first character of an input cell as commands.

@ is used to clear the input cell

: is used to define the format of the numerical results in all of the cells of the problem.

:I displays the operation result as an integer

:$ is the default format and displays dollars and cents

:F displays the operation results in floating notation

:* displays the operation results as ******

! causes recalculation of all operations

/ gives the user the choice of 7 commands

1. Set column width of output array at current column
2. Save data file on disk
3. Load data file from disk
4. Clear all data in input array to get ready for new problem input.
5. GOTO input array location which will then be the upper-left hand cell on the screen
6. Printout output array to hard copy device
7. Display this file on screen

& Defines range of summation operation by asking for the first cell location and the last location.

CONTROL-R is used to replicate cell parameters by first positioning the cursor cell at the cell to be replicated. Enter CONTROL-R and move the cursor cell to the desired location where CONTROL-R is again entered to replicate the parameters with incremented indices.

Vectored

```
930 H$ = H$ +  CHR$ (C%)
940  IF P%> L% THEN 960
950  GOTO 910
960 Y3% =  VAL (H$)
970  IF Y3% > MR% OR X3% > MC% THEN H$ = "ERROR":P% = L%+ 1: GOTO 1100
980 H$ = B$(Y3%,X3%)
990  GOSUB 680
1000  IF P%> L% THEN  GOSUB 1080: RETURN
1010  RETURN
1020  ' *** PARCE LINE FOR CHAR
1030 C%=  ASC ( MID$ (A$(Y%,X%),P%,1)):P%= P%+ 1
1040  RETURN
1050  '
1060  ' * ASSIGN ANSWER
1070  '
1080  IF A$(Y%,X%) = "" THEN  RETURN
1090  IF  LEN ( STR$ ( INT (A1))) > CW(X%) THEN H$ = "ERROR"
1100  ON OF% GOSUB 1140,1170,1180,1190
1110  IF OF% = 4 OR OF% = 1 THEN 1230
1120  B$(Y%,X%)=RIGHT$(S$ +  STR$ (A1),CW(X%))
1130  GOTO 1230
1140 A3 =  INT (A1):A1 = (A1 - A3) + 1.001
1150 H$ =  STR$ (A3) + "."+ MID$ ( STR$ (A1),4,2)
1160 B$(Y%,X%) =  RIGHT$ (S$ + H$,CW(X%)): RETURN
1170 A1 =  INT (A1): RETURN
1180  RETURN
1190 A1 =  INT (A1): IF A1 > 20 THEN A1 = 20
1200  IF A1 < 1 THEN A1 = 1
```

```
1650  NEXT
1660 X%=X2%:Y%=Y2%
1670  GOSUB 1820
1680  PRINT FN PC$(1,25);CL$;
1690  GOTO 2090
1700  '
1710  ' * SCREEN VALUE PRINTER
1720  '
1730 CW(0) = 0
1740  IF CW(X%) <  >  LEN (B$(Y%,X%)) THEN  GOSUB 350
1750  IF X%= SX% THEN CO% = 5: GOTO 1770
1760 CO% = 0: FOR X2% = SX% TO X%- 1:CO% = CO% + CW(X2%): NEXT X2%:CO% = CO% + 5
1770  PRINT FN PC$( Y%+ 5 - SY%,CO%);B$(Y%,X%);NV$;
1780  RETURN
```

```
1210 B$(Y%,X%)=LEFT$(LEFT$(T1$,A1)+S$,CW(X%))
1220  RETURN
1230  IF X%=> XM% THEN XM% = X%
1240  IF Y%> YX%(X%) THEN YX%(X%)=Y%
1250  RETURN
1260  '
1270  ' **** SUM(FUNCTION)
1280  '
1290 P%= P%+ 4: GOSUB 1030
1300  GOSUB 860:Y4% = Y3%:X4% = X3%
1310  GOSUB 1030: GOSUB 860
1320 A1 = 0:A2 = 0:X5% = X3%:Y5% = Y3%
1330  IF Y4% = Y5% THEN 1390
1340 X3% = X4%: FOR Y3% = Y4% TO Y5%
1350 P%= 1
1360  GOSUB 980
1370  NEXT
1380  GOSUB 1030: RETURN
1390 Y3% = Y4%: FOR X3% = X4% TO X5%
1400 P%= 1
1410  GOSUB 980
1420  NEXT
1430  GOSUB 1080: RETURN
1440  '
1450  ' * OUTPUT FORMAT *
1460  '
1470  GOSUB 680: GOSUB 1030
1480  IF C%= 36 THEN OF% = 1:' $
1490  IF C%= 73 THEN OF% = 2:' I
1500  IF C%= 70 THEN OF% = 3:' F
1510  IF C%= 42 THEN OF% = 4:' *
1520  GOTO 1080
1530  '
1540  ' * VIDEO SCREEN LAYOUT
1550  '
1560 X2%=X%:Y2%=Y%
1570  FOR X1% = 1 TO XM%
1580  PRINT FN PC$(1,25);: PRINT IV$;: PRINT "WORKING";NV$;
1590  FOR Y1% = 1 TO YX%(X1%)
1600  IF A$(Y1%,X1%) = "" THEN  1640
1610  PRINT FN PC$(1, 34);: PRINT  MID$ (T$,X1%,1);Y1%;
1620 X%= X1%:Y%= Y1%
1630  GOSUB 350
1640  NEXT
1790  '
1800  ' * SCREEN PRINT
1810  '
1820  PRINT CS$;FN PC$( 4,1);:PRINT IV$;LEFT$(S$,5);:MT%=MC%
1830 PP% = 0: FOR FX%= SX% TO MC%:PP% = PP% + CW(FX%)
1840 IF PP% > 77 THEN MT% = FX% - 1:FX%= MC%+1
1850 NEXT
1860  FOR FX%= SX% TO MT%
1870 H = CW(FX%) / 2:H1 =  INT (H):H2 =  INT (H - .2)
1880 H$ =  LEFT$ (S$,H1) +  MID$ (T$,FX%,1) +  LEFT$ (S$,H2)
1890  PRINT H$;
1900  NEXT:PRINT CL$;FN PC$(5,1);
1910  FOR FX%= SY% TO 18+ SY%: PRINT FX%;: IF FX% < 10 THEN  PRINT " ";
1920  PRINT : NEXT
1930  PRINT FN PC$( 5,5);NV$;CL$
1940 A = SX%:T = 5
1950  PRINT FN PC$( 5,1);
1960  FOR Y1% = 0 TO YX%(A)
1970  PRINT FN PC$(Y1%+5,T);B$(SY%+Y1%,A);
1980 IF LEN(B$(SY%+Y1%,A))<1 THEN PRINT FN PC$(Y1%+5,T);S$
1990  NEXT
2000 T = T + CW(A):A = A + 1: IF A =  < XM% THEN 1950
2010  RETURN
2020  '
2030  ' * PROMPT OF INPUT
2040  '
2050 DF = 1
2060  PRINT CS$
2070  GOSUB 1820
2080 X%= 1:Y%= 1
2090  PRINT FN PC$(1,1);MID$ (T$,X%,1);Y%;LEFT$(S$,CW(1)-4);"!<-";BP$
2100  PRINT FN PC$( 2,1);CL$;A$(Y%,X%);
2110 PRINT IV$;:GOSUB 1730
2120  '
2130  ' * INPUT AND PERFORM
2140  '
2150 A$=INPUT$(1):A=ASC(A$):IF A<>27 THEN 2200:' Test for ESC
2160 A$=INPUT$(1):IF A$="C" THEN DF=-1: GOTO 2290:'Move back
2170 IF A$="D" THEN DF=-1:GOTO 2420:'Move forward
2180 IF A$="B" THEN DF=1:GOTO 2290:MOVE UP
2190 IF A$="A" THEN DF=1:GOTO  2420:' Move down
2200  IF A = 47 THEN 2760:'/= GOTO command menu
2210  IF A=18 THEN 3780:' Replicate function
2220 IF A=4 THEN 3740:'CTRL D =quit program
```

```
2230  IF A = 64 THEN 2580:'ampersand to clear location
2240 IF A=33 THEN GOTO 1560:'execute functions
2250  IF A > 43 THEN 2570:'operands
2260  IF A = 38 THEN 2640:'& command
2270  IF A = 34 THEN  A$=INPUT$(1): GOTO 2570:'"=text input to cell follows
2280  A$=INPUT$(1): GOTO 2150
2290  GOSUB 1730
2300  ON DF + 2 GOTO 2310,2090,2340
2310 X%= X%+ 1: IF X%> MC% THEN X%= MC%: GOTO 2330
2320  IF X%> MT% THEN SX%= SX% + 1: GOSUB 1820: GOTO 2320
2330  PRINT IV$;: GOSUB 1730:GOTO 2090
2340 Y%= Y%+ 1: IF Y%> MR% THEN Y%= MR%: GOTO 2380
2350 IF Y>18+SY% THEN X3%= -1:SY%=SY%+10:Y%=SY%+10
2360 IF Y>MR% THEN Y=MR%:SY%= MR% - 18
2370  IF X3% =  - 1 THEN  GOSUB 1820:X3% = 0
2380  GOTO 2330
2390  '
2400 ' * MOVE CURSOR
2410 '
2420  ON DF + 2 GOTO 2430,2560,2480
2430  GOSUB 1730
2440 X%= X%- 1: IF X%> = SX% THEN  GOTO 2330
2450 SX%= SX% - 1: IF X%= 0 THEN X%= 1:SX%= 1: GOTO 2470
2460  GOSUB 1820
2470  GOTO 2330
2480  GOSUB 1730
2490 Y%= Y%- 1: IF Y%= > SY% THEN GOTO 2330
2500 SY%= SY% - 10:Y%= SY%: IF Y%<  = 0 THEN Y%= 1:SY%= 1
2510  GOSUB 1820
2520  GOTO 2330
2530  '
2540  ' ** INPUT STRING FOR PAGE
2550  '
2560  IF A$ = "&" THEN 2640:'command for summations
2570  PRINT FN PC$( 2,1);A$(Y%,X%): PRINT FN PC$( 2,1);: GOSUB 520
2580 IF A$ = CHR$(64) THEN A$(Y%,X%) =  LEFT$ (S$,CW(X%)): GOTO 2610
2590  IF A$ = "" THEN 2620
2600 A$(Y%,X%) = A$
2610  GOSUB 350
2620  GOTO 2090
2630  '
2640  ' ** SUM STATEMENT **
2650  '
2660  PRINT FN PC$( 1,1);CL$;
3110  PRINT#1, "<>"
3120 CLOSE#1
3130 Y%= 1:X%= 1
3140  GOTO 3720
3150  '
3160  ' * FILE IN *
3170  '
3180  PRINT FN PC$( 1,1);
3190  PRINT "READ FILE FROM DISK  FILENAME = ";:INPUT A$
3200  IF A$ = "" THEN  GOSUB 3340: GOTO 2090
3210  OPEN "I",1,A$
3220  INPUT#1, XM%
3230  FOR X%= 1 TO XM%
3240  INPUT#1, CW(X%),YX%(X%)
3250  FOR Y%= 1 TO YX%(X%)
3260  INPUT#1, A$(Y%,X%),B$(Y%,X%)
3270  NEXT
3280  INPUT#1, B$: ' ERROR IF NOT <>
3290  NEXT
3300  INPUT#1, B$: ' ERROR IF NOT <>
3310 CLOSE#1
3320  GOSUB 3340
3330 GOTO 3720
3340  PRINT FN PC$( 1,1);CL$;
3350  PRINT FN PC$( 2,1);CL$;
3360  PRINT FN PC$( 3,1);CL$;
3370  RETURN
3380  '
3390  ' ** GOTO LOCATION
3400  '
3410  GOSUB 3420: GOTO 2090
3420  GOSUB 3340
3430  PRINT FN PC$( 2,1);: INPUT "GO TO PAGE LOCATION :";A$
3440  GOSUB 3490
3450  IF X1% + Y1% = 0 THEN  RETURN
3460 X%= X1%:SX%= X1%:Y%= Y1%:SY%= Y1%:PRINT CS$
3470  GOSUB 1820
3480 GOTO 1690
3490 L%=  LEN (A$): IF L%< 2 THEN X1% = 0:Y1% = 0: RETURN
3500 X1% =  ASC ( LEFT$ (A$,1)) - 64
3510  IF X1% < 1 OR X1% > MC% THEN X1% = 0: RETURN
3520 Y1% =  VAL ( RIGHT$ (A$,L%- 1))
3530  IF Y1% < 1 OR Y1% > MR%-20  THEN X1% = 0:Y1% = 0
3540  RETURN
```

```
2670  PRINT FN PC$( 2,1);:INPUT "SUM(START = ";A$
2680  PRINT FN PC$( 2,1);CL$;"SUM(";A$;" THRU ";: INPUT "";B$
2690  PRINT FN PC$( 2,1);"SUM(";A$;" THRU "B$;")"
2700  IF A$ = "" OR B$ = "" THEN 2090
2710  A$(Y%,X%) = "&SUM(" + A$ + "-" + B$ + ")"
2720  GOSUB 350: GOTO 2090
2730  '
2740  ' * HANDLE GLOBAL COMMAND
2750  '
2760 CLOSE#1: PRINT FN PC$( 2,1);CL$;
2770  INPUT"1-WIDTH 2-SAVE 3-LOAD 4-CLEAR 5-GOTO LOCATION 6-PRINT 7-HELP";A$
2780  ON  VAL (A$) GOTO 2850,2960,3180,250,3410,3580,2790
2790 ON ERROR GOTO 2840:OPEN "I",1,"MBCALC.DOC"
2800 FOR I=1 TO 15:LINE INPUT#1, A$:PRINT FN PC$(I+5,8);CL$;A$:NEXT
2810 PRINT FN PC$(23,8);"HIT ANY KEY TO CONTINUE/ESC TO STOP":A$=INPUT$(1)
2820 IF A$<>CHR$(27) THEN 2800
2830 CLOSE#1:ON ERROR GOTO 0:GOTO 2050
2840 A$=INPUT$(1):RESUME 2830
2850  PRINT FN PC$( 2,1);CL$;:INPUT "WIDTH = ";A$:A=VAL(A$):IF A > 30 THEN 2850
2860  CW(X%) = A
2870  YH = Y%:XH=X%
2880  FOR Y%= 1 TO YX%(X%): GOSUB 780: NEXT
2890 Y2%=YH:X2%=YH
2900 GOSUB 1570
2910  GOSUB 3340: GOTO 2090
2920  '
2930  ' * DISK I/O
2940  ' * FILE OUT *
2950  '
2960  GOSUB 3340
2970  PRINT FN PC$( 1,1);CL$;
2980  PRINT "SAVE FILE TO DISK FILENAME = ";:INPUT A$
2990  IF A$ = "" THEN  GOSUB 3340: GOTO 2090
3000  PRINT FN PC$( 1,1);CL$
3010 OPEN "O",1,A$
3020  PRINT#1, XM%
3030  FOR X%= 1 TO XM%
3040  PRINT#1, CW(X%),YX%(X%)
3050  FOR Y%= 1 TO YX%(X%)
3060  PRINT#1, CHR$ (34);A$(Y%,X%); CHR$ (34)
3070  PRINT#1, CHR$ (34);B$(Y%,X%); CHR$ (34)
3080  NEXT
3090  PRINT#1, "<>"
3100  NEXT
3550  '
3560  ' *** PRINT OUT
3570  '
3580  GOSUB 3340
3590  PRINT FN PC$( 2,1);: INPUT "UPPER/LEFT CORNER:";A$: GOSUB 3490
3600 X3% = X1%:Y3% = Y1%
3610  PRINT FN PC$( 2,1);: INPUT "LOWER/RIGHT CORNER :";A$: GOSUB 3490
3620 X4% = X1%:Y4% = Y1%
3630  OPEN "O",1,"LP:"
3640  FOR Y1% = Y3% TO Y4%
3650  FOR X1% = X3% TO X4%
3660  PRINT#1, LEFT$ (B$(Y1%,X1%) + S$,CW(X1%));
3670 ' LPRINT   LEFT$ (B$(Y1%,X1%) + S$,CW(X1%));
3680  NEXT
3690  PRINT#1,
3700 ' LPRINT
3710  NEXT
3720  CLOSE#1
3730 X1% = 1:Y1% = 1:GOTO 3460
3740 PRINT DS$;CS$:X=FRE(0):PRINT X:END
3750  '
3760  ' REPLICATE FUNCTION
3770  '
3780 PRINT BP$:IF CR<1 THEN XH=X%:YH=Y%:CR=CR+1:GOTO 2130
3790 DX%=X%-XH:DY%=Y%-YH:LA=LEN(A$(YH,XH)):IF LA<1 THEN 2130
3800 A$(Y%,X%)=A$(YH,XH)
3810 FOR I=1 TO LA:A$=MID$(A$(YH,XH),I,1):A=ASC(A$)
3820 IF A<65 OR A>80 THEN 3880
3830 IF I=1 THEN A$(Y%,X%)=CHR$(A+DX%):GOTO 3850
3840 A$(Y%,X%)=MID$(A$(Y%,X%),1,I-1)+CHR$(A+DX%)
3850 N%=VAL(MID$(A$(YH,XH),I+1)):IF N%<1 THEN 3850
3860 N$=STR$(N%):LN=LEN(N$):N%=N%+DY%:N$=STR$(N%)
3870 A$(Y%,X%)=A$(Y%,X%)+MID$(N$,2)+MID$(A$(YH,XH),I+LN)
3880 NEXT:CR=0:GOTO 2090
```

Heath/Zenith Users—
Get the Information You Need:

An Independent Magazine. . .

SEXTANT The Independent Magazine for the Entire Zenith Computer Community

Brings you over 80 pages of Heath/Zenith-specific information, innovative ideas, and supplier advertisements exclusively for your system.

Every issue of *Sextant* is filled with:

- "how-to" articles
- short programs
- over 50 compatible product ads
- coverage of community affairs which highlights specific events, companies, and individuals
- hardware and software reviews for Zenith *and* the independent suppliers

Sextant is a quarterly publication, carefully edited to provide a wide coverage of the Heath/Zenith community for the benefit of the Heath/Zenith user.

An Independent Newsletter. . .

BUSS The Independent Newsletter of Heath Co. Computers

Brings you a direct link to the activities of the Heath/Zenith community, with news arriving every two to three weeks.

Join a community of over 5,000 Heath/Zenith users sharing timely ideas and information about your system.

Buss highlights:

News . . . technical tips . . . special feature reports . . . 200 independent suppliers . . . product announcements and updates . . . user contributions . . .

You'll receive *Buss* by first class mail, within a week of the editorial deadline. *Buss* will be published 20 times in 1983.

A Whole World of Information About Your System

Available to you now

Call Toll Free: DATATEL™ **800/341-1522 And Order Today!**
(M-F 6 a.m.-9 p.m. EST. For orders only.)

Charge your order to Visa or MasterCard, or we'll bill you later.

To order by mail, write to:
Buss/Sextant Fulfillment Dept., 716 E Street S.E., Washington DC 20003
phone: **202/544-0900**

Sextant ordering information:
8 issues for $19.94 ($23.00 to Canada)
4 issues for $9.97 ($11.50 to Canada;
$14.00 overseas via surface mail,
$35.00 overseas via air mail)

Buss ordering information:
36 issues for $44.00 ($65.00 overseas)
24 issues for $32.00 ($45.00 overseas)
12 issues for $20.00 ($25.00 overseas)

Allow 4-6 weeks delivery. (Checks must be in U.S. dollars payable on a U.S. bank.)

HDOS For Everyone
(Including H/Z100 Users)

Pat Swayne
Software Engineer

This article describes a new HUG software product, HRUN, which allows HDOS programs to be run under CP/M. It is presented here as well as in the New Hug Software column because of potential interest to HUG members.

As anyone knows who went to the HUG Convention or read about it in REMark, HDOS is very popular among HUGgies, and there is concern about its future as new computers come along. HUGgies are also an impatient lot (after all, isn't that what computers are all about?), and many wanted HDOS for the new H/Z100 about as soon as the first one rolled off the assembly line.

I had for some time been thinking of the possibility of running HDOS programs under CP/M using an interpreter, or "shell" program to intercept the HDOS system calls and translate them to appropriate CP/M calls. For many programs, that is not too difficult, as Doug Alsip demonstrated when he converted HUG's "Adventure" to run under CP/M. When I first conceived the idea of HRUN, it was going to be about the same thing, except that it would be a separate program. The user would place it and the HDOS program to be run on a disk, and enter HRUN FILENAME to run his program. HRUN would translate console, disk, and printer operations to their CP/M equivalents to run the program, and when it finished, control would return to CP/M.

Well, HRUN has grown quite a bit beyond that concept. It now provides almost a complete simulated HDOS environment. Virtually any non-hardware dependent HDOS program can be run under it, including sophisticated debugging programs such as HUG's ALDT and RDT. Not only does HRUN support nearly every HDOS system call and H17 ROM calls, but it maintains many of the memory tables that "real" HDOS uses, such as the Channel table, the Device table, the Active I/O table, and the System Date. And when a program finishes, it returns to HRUN, which displays it's own HDOS-like prompt on your console.

How HDOS Works (for CP/M Users)

Before I get in too deep, perhaps I should briefly describe how HDOS works, for those of you who might be using CP/M only at this time. HDOS is a file oriented operating system system. That means it is broken up into several little programs, each of which does a particular job. (CP/M in its original form, by contrast, used a reserved area of the system disk for the entire system, which was invisible to the user.) The central part of HDOS is a program called HDOS.SYS, which manages the use of the other programs. It loads them into memory from the disk as required, and frees up memory space when segments are not required. The programs that "talk" to the outside world (that is, to disks, printers, etc.) are called "device drivers". The device drivers and other temporary parts of HDOS (called Overlays) can be permanently loaded into memory if you wish for faster operation at the expense of memory space. Tables are maintained in memory to indicate to HDOS (and to user programs that wish to use them) which device drivers are in use at the time, their status in memory (temporary or permanent), and how they are being used (read, write, etc.), along with other information.

The Device Table is the table that indicates which device drivers are available, where they are in memory (if they are in memory), and their capabilities, among other things. The Channel Table is used to show when a device is actually in use, and shows how it is being used. For disk file operations, the channel table will contain a copy of the file's directory entry. (Note: You do not need to know all of this to use HDOS. It is presented to help you understand how it works.)

Probably the most important thing to remember about HDOS is that it treats all communications with the outside world in basically the same way. Here are two HDOS Microsoft BASIC programs to illustrate this.

```
10 OPEN "O",1,"SY1:DATA.DAT"
20 PRINT #1,"THIS IS A TEST"
30 CLOSE #1
```

```
10 OPEN "O",1,"LP:"
20 PRINT #1,"THIS IS A TEST"
30 CLOSE #1
```

The first of these programs writes a line of text to a disk file, and the second one writes it to a printer. The only difference is in line 10, in the argument to the OPEN command. We can analyze the "SY1:DATA.DAT" and the "LP:" as follows: The "SY" and the "LP" are the names of the device drivers used in each case. Every HDOS device driver has a two letter name. The number 1 following the "SY" is the unit number within that device. Since "SY" is a disk driver (or, more properly, a "directory device"), the unit number corresponds to the particular drive in use. Unit 1 corresponds to CP/M drive B:. If a device driver only has one unit (or if the unit wanted is zero), the unit number may be left off, as in the case with "LP". Following the device driver name and unit number (if any) is a colon, which completes the device driver designation. Following the colon in the first program is the file name, which is required if the device is a directory device. HDOS is very protective, and will spit error messages at you if any of this stuff is not "just so".

How HRUN Works

The HRUN program emulates the parts of HDOS discussed above. It combines the functions of HDOS.SYS, the HDOS overlays, and 5 device drivers into one program. I decided to emulate device drivers in HRUN rather than trying to make it use "real" ones for the following reasons. First, none of the existing device drivers will work on H/Z100 computers, which, for example, use different UART chips to communicate with printers, etc. Second, it would have used more memory to have real device drivers, and I wanted to allow the user as much memory as possible. As it turned out, HRUN gives the user more memory in some applications than real HDOS. Finally, it would have taken longer to implement the use of real device drivers, and I wanted to make HRUN available as soon as possible.

The following chart lists the 5 pseudo device drivers in HRUN.

DEVICE	USE
SY:	The directory (disk) device. Unit numbers are mapped to corresponding CP/M drive letters (0 = A, 1 = B, etc.)
TT:	The console device. Output is the CRT screen, and input is the keyboard.
LP:	The printer device. It uses the CP/M LST: device. Provision is made to expand tabs to spaces (within HRUN) for printers that cannot handle tabs.
AT:	The alternate terminal device. It uses the CP/M TTY: device. AT: can be used to drive a second printer in some applications.
ND:	The null device. This device can be used to test the read-ability of a disk file by copying it to ND:, or to create an empty directory entry by copying ND: to a disk.

HDOS users will notice that there is only one disk device driver. Real HDOS uses a separate device driver for each controller in the computer. For example, if you have an H89 with a hard sector controller and a soft sector controller, you would have two disk device drivers. Since HRUN works through CP/M, only one disk device driver is needed, which will "talk" to any drive that CP/M can access. That means that the drive you access as DK0: under real HDOS might be SY3: under HRUN.

Using HRUN

HRUN was designed to emulate HDOS in the command mode, in addition to running programs. That means that if you copy HRUN.COM to your system disk and enter HRUN, it will sign on normally, and prompt for your entries with the standard HDOS prompt (the character ">"). You can also go directly into an HDOS program from CP/M by entering HRUN FILENAME, and you can even enter more complex commands, such as

A>HRUN MBASIC SY1:MENU

This command would start HRUN, which would load and execute HDOS Microsoft BASIC, which would then load and execute the program MENU.BAS from disk SY1: (CP/M disk B:). Real HDOS has the ability to execute a program at boot-up, simply by renaming the program in question to PROLOGUE.SYS. HRUN can simulate that ability if you configure your CP/M system disk to execute an HRUN command line at cold boot.

Since HRUN runs under CP/M, any program to be run under it must be copied to a CP/M disk from its HDOS disk. To accomplish this, a new version of HUG's HTOC program is included with HRUN. It has been expanded to support more disk formats, and now allows "wild cards" when you specify the files to copy. For example, if you want to copy all of the BASIC programs from an HDOS disk, you could enter *.BAS as the file to copy.

HRUN comes with several other support programs. There are so many, in fact, that the complete package takes up 3 5.25 inch single density hard sector disks. We provide documentation for HRUN and its support programs in printed form.

Among the support programs are several that are supplied with real HDOS, as listed in the chart below. (Our thanks to Barry Watzman for letting us include these programs.)

PROGRAM	USE
PIP.ABS	This HDOS's Peripheral Interchange Program, used to copy between devices.
FLAGS.ABS	This program is used to set and reset file flags, the HDOS equivalent of CP/M file attributes. HDOS flags are mapped to CP/M attributes and vice versa by HRUN.
SET.ABS	This program is used to change device driver characteristics. In HRUN, it only works on the TT: device, and can be used to set line length, case mapping (lower to upper on input and/or output), and other parameters.
BASIC.ABS	This is Heath's Benton Harbor BASIC, normally supplied with HDOS.
EDIT.ABS	This is the standard HDOS text editor. It is not considered to be the best editor around, but will do if you have no other HDOS editor.
ASM.ABS	The HDOS 8080 assembler.
XREF.ABS	A cross reference utility for the HDOS assembler.
PATCH.ABS	A utility for patching (altering) machine language files.

Several other utility programs are included to make HRUN more useful.

MAKEDIR.ABS
: This program constructs an HDOS disk directory (a file called DIRECT.SYS) from the CP/M disk directory. DIRECT.SYS is required if you want to view the directory via PIP (the CAT command), or to use wild cards in PIP operations, or the FILES command in HDOS Microsoft BASIC.

DIR.ABS
: With this program, you can view the directory on a disk that does not have a DIRECT.SYS file. HRUN is set up so that the command DIR uses DIR.ABS to show the directory, and CAT uses PIP.ABS to show the files in DIRECT.SYS, in standard HDOS format.

SUBMIT.ABS
: This program is a command file processor for HRUN. Its operation is nearly identical to the CP/M SUBMIT program except that it allows commands to be passed on to any HDOS program. This program is provided because neither of the HUG SUBMIT programs for HDOS will work with HRUN.

XFORM.ABS
: This program converts text files from CP/M format (a CRLF after each line) to HDOS format (LF only after each line) and vice versa. It allows you to use CP/M editors and word processors to create text files for HRUN.

HELP.ABS

This program prints a list of built-in HRUN commands on your console when you type HELP.

MAKESYS.SUB

This is a SUBMIT file that makes it easy for you to generate new HRUN system disks by copying the files used in normal operation to a new disk. It also runs FLAGS to write protect the files.

Advantages and Disadvantages of HRUN

HRUN is intended to allow you to run HDOS programs on systems for which HDOS is not available, not to replace HDOS. However, HRUN has some advantages that make it better than real HDOS for some applications. Among the advantages are:

1. HRUN will run on virtually any CP/M (2.0 or greater) compatible computer. Imagine running the HUG Small Business Package on a Fruit Tree computer (with Z80 card, of course)!

2. HRUN will access any disk format that CP/M can access. HDOS in its present form cannot handle a disk sector size greater than 256 bytes, which rules out extended density on the H/Z47 and H/Z37. HRUN lets you use those densities, which not only means more disk space, but faster disk access speed (disk access speed is generally faster with larger sector sizes). HRUN also lets you use hard disks, etc., that you may not have device drivers for. (If you don't know about extended density on the H/Z37, run FORMAT and type E when it asks you "which density".)

3. HRUN lets you keep your favorite programs on one disk, whether they are HDOS or CP/M programs.

4. You have more memory space (exception noted below) with HRUN than you have HDOS with both overlays and an LP: loaded. That means more room for MBASIC programs.

5. Assembly programmers can make "hybrid" programs with system calls to both HDOS and CP/M in their programs, using whatever best suits the need.

There are also some disadvantages to HRUN, compared to real HDOS. Among them are:

1. You cannot use real device drivers with HRUN. That rules out all of those fancy printer drivers that are out there. However, most programs that use printers handle the fancy stuff within themselves.

2. The HDOS TIC counter (the HDOS equivalent of the CP/M TIC counter at 0BH) will not work on H/Z100 computers or others that do not have the appropriate clock interrupt. An effort was made to make the HDOS TIC Counter work on H/Z100's by updating it during system calls (as the CP/M TIC Counter changes), but delay loops that do not include system calls will still not work. Programs can use the CP/M TIC Counter on H/Z100's, but will have to adjust for the fact that it is incremented at 10 ms intervals on those machines, instead of at 2 ms intervals as on H8's and H89's.

3. Programs that insert commands into the HDOS type-ahead buffer or test it will not work. However, HRUN maintains a pseudo type-ahead buffer that can be used for those purposes. Instructions on using it are included with HRUN. It is actually easier to use than the real HDOS type-ahead buffer.

4. Debugger programs (such as RDT) will have to be modified to run on computers (such as the H/Z100) that do not allow the use of level 2 interrupts. Provision is made in HRUN to use interrupt level 6 on H/Z100's.

5. If you have an H89 with less than 64k of memory, your memory space under HRUN will be 8k less than with real HDOS. This is because under HDOS, your RAM memory is mapped to start at 2000H instead of 0 when you are running HDOS.

6. Disk access speed on hard sector disks will be slower, because it is slower under CP/M. However, disk access speed on soft sector 5.25 inch disks will be faster.

I think you will agree that the advantages of HRUN far outweigh its disadvantages, especially if HDOS is not available for your system. HRUN goes a long way toward providing complete HDOS support for the H/Z100 and other CP/M compatible computers. See the New HUG Software column in this issue for more information.

TERMBUG

HUG Part No. 885-3003

A small bug exists in all versions of TERM that run in the TPA. It will only occur when filling memory (using the store mode) and the file size reaches your memory limit. The problem is in a routine called LDMEM. The following is a corrected version of that routine. Change yours to look like that, reassemble, load, and you're ready to go.

```
LDMEM   LDA     LDFLG           ;CHECK THE LOAD FLAG
        CPI     0
        RZ                      ;QUIT IF NOT SET
        MOV     M,C
        INX     H
        INX     D
        PUSH    H               ;SAVE THE BUFFER POINTER
        PUSH    D               ;SAVE THE COUNT
        XCHG                    ;PUT POINTER IN DE
        LHLD    6               ;GET THE TOP OF MEMORY
        DCX     H
        MOV     A,H             ;LET'S SEE IF WE'RE EQUAL YET
        CMP     D
        JZ      LDMEM1
        POP     D               ;NOT ZERO YET THEN...
        POP     H
        RET                     ;QUIT
LDMEM1  MOV     A,L             ;CHECK LOW BYTES
        CMP     E               ;SAME ?
        JNZ     LDMEM2          ;NO THEN KEEP GOING
        XRA     A
        STA     LDFLG           ;CLEAR THE LOAD MEMORY FLAG
        DCR     A
        STA     FULFLG
        LXI     D,MSG8          ;WARNING MESSAGE
        CALL    PMSG            ;PRINT IT
LDMEM2  POP     D
        POP     H
        RET
```

Weather Station-
Computer Interface

Edward T. Wright Jr.
Rt. 1 Box 19B
Samuels, Idaho 83862

When I received the ID-4001 Heath Weather Station, I was intrigued with the Applications Section of the Operations Manual, and in particular, the "Computer Interface" paragraphs which listed the output signals on the Weather Station for interconnection to an external computer.

This was all well and good, but, for a novice programmer such as I, the statement that "your own imagination and programming expertise are the only limitations—for computer assisted tabulation ", left me in the well-known position of most people for whom the thought of assembly programming leads to a mind- boggling experience, to say the least.

Fortunately, for people like me, there are some fine programmers around who seem to "eat up" assembly language programming. One such person, John Randolph, E. Liverpool, OH., placed an excellent device driver in the Micronet system, which provided the answer to some of my problems of interfacing. The device driver developed by John was for the interface by use of the Heath H8-2 parallel port board and the H8 computer. My system consists of a H89 with an Environmental Control Systems PC89, 9 port parallel board installed in the P501 & P507 slots on the left side of the H89 PCU board. Due to the excellent design and simplicity of initialization of the PC89 board, the problems of connection of the ID-4001 Weather Station through the PC89 parallel port board to the H89 Computer, and the modification of the device driver to operate with this parallel port board proved to be within my level of expertise and the results are provided for use by others who may have the H89/PC89/ ID-4001 system.

The modification of the device driver consists of changing the parallel board initialization, port address, and scan sub- routines. The revised device driver program for use with the PC89 board is shown by listing 1.

The pin assignments and interconnections for the ID-4001 Weather Station and the PC89 parallel board are shown by listing 2. It will be noticed that pins 8 & 9 of P202 on the ID-4001 printed circuit board are not used with this device driver since the decoding subroutine does not require these signals to properly translate the values indicated by the LEDs on the weather station panel. Further, pin 20, the weather station strobe signal is not connected since the device driver uses the HDOS clock interrupt processing and the strobe signal is not required.

A simple MBASIC program for printout of the ID-4001 weather information is shown by listing 3, and a sample printout is shown by listing 4. It should be mentioned at this point that the real- time clock on the PC89 board is used to provide time records on the weather record printout. The real-time clock device driver, provided with the PC89 board is loaded along with the printer and weather station device drivers preliminary to loading MBASIC and the operating program.

The weather station device driver is capable of sensing the position of the "measuring units" switches located on the rear of the weather station and will automatically cause printout in the correct units, i.e. Degrees (F or C), Wind Velocity (mph or km/h), Barometric pressure (inches or millibars).

After the weather station driver is assembled, using the ASM.ABS program, operation of the system consists of loading the device drivers for the weather station, real-time clock and printer, then, MBASIC and the operating program for readouts:

```
LOAD LP: <CR>
LOAD RT: <CR>
LOAD WX: <CR>
LOAD MBASIC <CR>
LOAD "WXSTA.MBA" <CR>
RUN <CR>
```

In the MBASIC program, lines 40-90 provide readouts every ten minutes. This can be changed by modifying these instructions as desired. The term "MID$(A$,4,2)" provides access to the "minutes" portion of the real-time clock in the RT: device driver. The printout will start whenever the time is equal to those listed. Lines 320-340 provide a delay so only one printout each ten minutes will be made. If more than one printout is desired, the value of "15000" can be reduced. It will be noticed that the information will not change until the weather station has completed the next scan cycle. This is not a problem for slowly varying parameters such as temperature or pressure, but will provide accurate readings for wind direction and velocity only at the time of scan, at which time the parameters are up- dated.

A further refinement in the program could provide printout at a longer interval such as every half hour and comparison of the temperature with a pre-determined level with action to turn on- off heating equipment. The possibilities are unlimited or, as Heath stated, "imagination and programming expertise are the only limitations——". I hope that this meager effort on my part will provide others with the courage to try the interface so that they may realize the same pleasure and sense of accomplishment that I did. GOOD LUCK!!!

```
SAMPLE PRINTOUT FROM ID-4001 WEATHER STATION


17:40:00  Thursday  November 4, 1982

Temperature Degs F  Outdoor= +43   Indoor= + 70
Pressure 29.75 Inches and Steady
Wind out of NNW at 10 Mi/Hr
```

```
    TITLE   'HEATH ID-4001 Weather Station Driver'
    STL     'Assembly Options'
    EJECT
*           This driver, placed on Micronet by:
*
*                John Randolph
*                P.O. Box 280
*                E. Liverpool, OH 43920
*                MNet 70325,205
*
*           was modified for use with the ECS PC89
*           9 port parallel board by:
*
*                Edward T. Wright Jr.
*                Rt.1 Box 19B
*                Samuels, ID 83862
*                MNet 70355,130
*
*           The following options are available:
*
*   TEST    If .TRUE will allow testing driver on
*           a system without the weather computer
*           interface. Must be .FALSE to operate
*           properly with the ID-4001 connected.
*
*   FORMAT  If .TRUE will provide the weather data
*           in narrative form. If .FALSE will only
*           provide data separated by commas.
*
*   Pn.PA   This allows setting the base port for the
*           8255A PPI, as an example:
*
*                PPI #1 Base Port (P1.PA)=000
*                PPI #2 Base Port (P2.PA)=004
*                PPI #3 Base Port (P3.PA)=014Q
*
*
.TRUE   EQU 0
.FALSE  EQU 1
*
TEST    EQU .FALSE
FORMAT  EQU .TRUE
*
P2.PA   EQU 004Q        PORT A BASE ADDRESS FOR PPI 2
P2.PB   EQU P2.PA+1     PORT B OF PPI 2
P2.PC   EQU P2.PA+2     PORT C OF PPI 2
P2.CWR  EQU P2.PA+3     CONTROL WORD REG. FOR PPI 2
*
        STL             'DEFINITIONS'
        EJECT
        XTEXT   PICDEF
        XTEXT   DVDDEF
USERFWA EQU     042200A
        XTEXT   SETCAL
$MOVE   EQU     030252A
$DADA.  EQU     030101A
$TYPTX  EQU     031136A
$TBRA   EQU     031076A
EC.DNS  EQU     005Q
EC.EOF  EQU     001Q
EC.ILR  EQU     012Q
EC.UUN  EQU     033Q
EC.ILO  EQU     041Q
EC.DDA  EQU     027Q
EC.UNA  EQU     036Q
EC.ILV  EQU     040Q
TICCNT  EQU     040033A
UIVEC   EQU     040037Q
RCVRR   EQU     2
DP.BIT  EQU     40Q
FS.BIT  EQU     100Q
RS.BIT  EQU     200Q
F.BIT   EQU     10Q
*
*       Default Device Options
*
*       Code header
*
        CODE    PIC
```

Vectored ☞

```
        DB      DVDFLV          THIS IS A DVD
        DB      002Q            READ ONLY
        DB      00000001B       ONE UNIT
        DB      1               MAXIMUM # UNITS
        DB      002Q
        DS      7
        DB      DVDFLV          SETable
        CODE    -REL
        SET     *
        CODE    +REL
        DS      DVD.STE-.       RESERVED
        STL     'SET CODE'
        EJECT
*
*       SET CODE
*
SETNTY  EQU     *
        ERRNZ   *-DVD.STE
        ANA     A
        JNZ     SET1
        MOV     B,D
        MOV     C,E             (BC)=PARAMETER LIST ADDR
        LXI     D,PRCTAB
        LXI     H,OPTTAB
        CALL    $SOP            PROCESS OPTION
        RC
        CALL    $SNA            SKIP TO NEXT ARG
        RZ
        MVI     A,EC.ILO
        STC
        RET

SET1    MVI     A,EC.UUN
        STC
        RET
*       PRINT HELP INFO
HELP    CALL    $TYPTX
        DB      12Q,'WS: Options',12Q
        DB      12Q,'   SCAN n     Scan interval in 1/2 secs'
        DB      12Q,12Q,'       HELP       Type this list'
        DB      212Q
        XRA     A
        RET
*       Option Table
OPTTAB  DW      OPTTABE
        DB      6
```

```
        STC
        RET
*
*       IGNORE REQUEST
*
WSNOP   EQU     *
        XRA     A
        RET                     REPORT NO ERROR
*
*       LOAD WEATHER STATION DRIVER
*
WSLOAD  EQU     *
        CALL    INIT            INITIALIZE WS PORTS
        LHLD    UIVEC+1         GET HDOS CLOCK VECTOR
        SHLD    CLKRET+1        PUT AT END OF OUT ROUTINE
        LXI     H,CLOCK         GET OUR CLOCK START ADDR
        SHLD    UIVEC+1         REPLACE CLOCK VECTOR
        XRA     A
        RET

WSREAD  EQU     *
        MOV     A,B
        CPI     1               MUST BE >=1
        MVI     A,EC.ILR        PREPARE FOR ERROR
        RC                      TOO FEW BYTES REQUESTED
        DCR     B               ACCOUNT FOR THOSE SENT
        PUSH    B               SAVE FOR LATER
        MVI     B,0             256 BYTES
        LXI     H,CDATA-1
        DI                      PREVENT UPDATE DURING
WSRD1   INX     H               POINT TO NEXT
        MOV     A,M             GET IT
        ANI     177Q            TRIM PARITY
        STAX    D               HAND IT OVER
        INX     D
        DCR     B               KEEP TRACK OF # SENT
        CMP     M               END OF DATA?
        JZ      WSRD1           IF NOT
        EI                      CRITICAL PART DONE
WSRD2   XRA     A               FINISH SENDING ZEROS
        STAX    D
        INX     D
        DCR     B               B=0 IF 256 SENT
        JNZ     WSRD2           IF NOT DONE
        POP     B               BC=REMAINING COUNT
        MOV     A,B
```

```
        DB      'SCA','N'+200Q,VALI,10,1,120
        DW      SCFLFV
        DB      'HEL','P'+200Q,HELPI
        DB      0,0,0,0,0
OPTTABE DB      0
*       PRCTAB - Processor Table
PRCTAB  DS      0
FLAGI   EQU     *-PRCTAB/2
        DW      $PBF
VALI    EQU     *-PRCTAB/2
        DW      $PBV
HELPI   EQU     *-PRCTAB/2
        DW      HELP
        CODE    -REL
        SET     *
        CODE    +REL
        DS      DVD.ENT-.
        STL     'Driver Code'
        EJECT
        ERRNZ   *-2000A
ENTRY   EQU     *
        CPI     11              ONLY 11 POSSIBLE
        JNC     WSILLGL         FUNCTION CALLS
        CALL    $TBRA
        DB      WSREAD-*        READ
        DB      WSILLGL-*       WRITE (ILLEGAL)
        DB      WSILLGL-*       READR (ILLEGAL)
        DB      WSNOP-*         OPENR
        DB      WSILLGL-*       OPENW (ILLEGAL)
        DB      WSILLGL-*       OPENU (ILLEGAL)
        DB      WSNOP-*         CLOSE
        DB      WSABT-*         ABORT (ABORT)
        DB      WSNOP-*         MOUNT (IGNORE)
        DB      WSLOAD-*        LOAD
        DB      WSNOP-*         READY (IGNORE)
*
*       ILLEGAL OPERATION
*
WSILLGL MVI     A,EC.DNS
        STC
        RET
*
*       ABORT OPERATION
*
WSABT   EQU     *
        MVI     A,EC.DDA        DD ABORT
        ORA     C
        RZ                      COMES OUT EVEN 256
        MVI     A,EC.EOF        EOF IF MORE REQUESTED
        STC                     FLAG ERROR
        RET
CLOCK   EQU     *
        LDA     TICCNT          GET CLOCK COUNTER
        ANA     A               SEE IF 1/2 SEC IS UP
        JNZ     CLKRET          RET IF NOT
        LXI     H,SCFLG         SCAN FLAG VALUE
        DCR     M               SHOW ANOTHER 1/2 SEC
        JNZ     CLKRET          IF NOT TIME FOR SCAN
        MVI     M,5             RESET COUNTER
SCFLGV  EQU     *-1
        IF      TEST
        JMP     ERR
        ENDIF
        CALL    SCAN            SCAN WEATHER STATION
        JZ      ERR             IF SCAN ERROR
        CALL    VALID           SEE IF DATA OK
        CNC     UPDATE          UPDATE CURRENT DATA
CLKRET  JMP     0               CLK VECTOR GOES HERE
ERR     EQU     *
        CALL    INIT            OPT TO RE-INIT ON SCAN ERR
        LXI     H,ERCNT+1       COUNT SCAN ERROR
        INR     M               ADD ONE MORE
        MVI     A,'9'
        CMP     M               OVERFLO?
        JNC     CLKRET
        MVI     M,'0'
        DCX     H
        INR     M
        CMP     M
        JNC     CLKRET
        MVI     M,'0'
        JMP     CLKRET

INIT    MVI     A,233Q          CONFIGURE PPI FOR ALL LINES IN
        OUT     P2.CWR          TO CWR PORT
        RET                     DONE!!

UPDATE  EQU     *
.TEMP   LDA     RDATA+8         TEMP 1'S
        PUSH    PSW             SAVE
        ANI     RS.BIT          DEG. C?
        MVI     A,'C'           ASSUME IT
```


Vectored ▱

```
        JZ      .TPI
        MVI     A,'F'           OTHERWISE F
.TPI    STA     TMPU
        LDA     RDATA+7         SIGN & OVFLO
        PUSH    PSW
        ANI     DP.BIT          SEE IF INDOOR
        LXI     D,TMPO          ASSUME NOT
        JZ      .TP2            IF OUTDOOR
        LXI     D,TMPI          IF IS INDOOR
.TP2    POP     PSW             SIGN & OVFLO
        PUSH    PSW
        ANI     4               TRIM TO SIGN
        MVI     A,'-'           ASSUME MINUS
        JZ      .TP3            IF SO
        MVI     A,'+'           OTHERWISE PLUS
.TP3    STAX    D
        INX     D
        POP     PSW             SIGN & OVFLO
        ANI     2               TRIM TO DIGIT
        CALL    DECODE
        INX     D
        LDA     RDATA           TEMP 10'S
        CALL    DECODE
        INX     D
        POP     PSW             TEMP 1'S
        CALL    DECODE
.PRESS  LXI     H,PRU           DET OF PR. UNITS
        LDA     RDATA+7         [7] SET IF INCHES
        LXI     D,PRUM          MILLIBARS SOURCE
        ANI     RS.BIT          MBRS SWITCH IN?
        JZ      .PRI            IF YES
        LXI     D,PRUI          INCHES SOURCE
.PRI    LXI     B,10            10 BYTES TO MOVE
        CALL    $MOVE
        LXI     D,PR            PRESSURE VALUE
        LDA     RDATA+1         PRESS 10'S
        PUSH    PSW
        ANI     27Q             TRIM OUT f
        CALL    DECODE
        INX     D
        LDA     RDATA+9         PRESS 1'S
        CALL    DECODE
        INX     D
        INX     D
        LDA     RDATA+2         PRESS 1/10
        CALL    DECODE
```

```
        CALL    $MOVE
        ANA     A
        RET                     ALL DONE!

SCAN    EQU     *
        MVI     A,233Q          CONFIGURE FOR DATA FROM PPI
        OUT     P2.CWR          TO CONTROL WORD REG PPI2
        IN      P2.PA           CLEAN GARBAGE
        IN      P2.PB           CLEAN GARBAGE
        LXI     H,RDATA         RAW DATA PNTR
        LXI     D,44*256+1      D=TRYS E=DISP TO GET
SC1     CALL    GETPAIR
        DCR     D
        JZ      SCFAIL
        CMP     E               RIGHT ONE?
        JNZ     SC1             IF NO TRY AGAIN
SC2     MOV     M,C             SAVE DATA
        INX     H               POINT TO NEXT
        CPI     17Q             WAS IT LAST ONE?
        JZ      SCOK            DONE
        INR     E               E=NEXT EXPECTED
        MOV     D,A             D=LAST GOTTEN
SC3     CALL    GETPAIR
        JZ      SC3             DSP=0 IS A GLITCH
        CMP     D               SAME AS LAST?
        JZ      SC3             TRY AGAIN
        CMP     E               RIGHT ONE?
        JZ      SC2             SAVE AND REPEAT
SCFAIL  STC
SCOK    MOV     M,B             SAVE WIND DIR
        RET

GETPAIR EQU     *
        IN      P2.PA           YES, GET IT
        MOV     B,A             B=DISPLY #
GP1     IN      P2.PB
        MOV     C,A             C=DATA
        MOV     A,B
        ANI     17Q             TRIM OFF WIND
        RET

VALID   EQU     *
        LXI     H,RDATA+5
        MVI     B,8
        MOV     A,M
.VA1    ANA     M
```

```
        INX     D
        LDA     RDATA+10        PRESS 1/100
        CALL    DECODE
        POP     PSW
        LXI     H,PRC           POINT TO CHG
        LXI     D,PRCS          ASSUME STEADY
        LXI     B,7             7 BYTES TO MOVE
        ANI     DP.BIT+F.BIT    RIS & FAL FLAGS
        JZ      .PR2            IF STEADY
        LXI     D,PRCR          ASSUME RISING
        ANI     DP.BIT          FALLING FLAG
        JZ      .PR2            IF RISING
        LXI     D,PRCF
.PR2    CALL    $MOVE
.WIND   LXI     H,WU            DET OF WIND UNITS
        LXI     B,5             5 BYTES
        LXI     D,WUK           ASSUME KM/HR
        LDA     RDATA+6         KM/HR SWITCH
        ANI     RS.BIT          SET FLAGS
        JZ      .WI             IF KM/HR
        LXI     D,WUN           ASSUME KNOTS
        LDA     RDATA+5         KNOTS SWITCH
        ANI     RS.BIT          SET FLAGS
        JZ      .WI             IF KNOTS
        LXI     D,WUM           MI/HR
.WI     CALL    $MOVE
        LXI     D,WSPD          WIND SPEED
        LDA     RDATA+3         WIND 10'S
        CALL    DECODE
        INX     D
        LDA     RDATA+11        WIND 1'S
        CALL    DECODE
        LDA     RDATA+15        WIND DIR
        RLC
        RLC
        RLC
        RLC
        ANI     17Q
        MOV     B,A
        ADD     A
        ADD     B               A=A*3
        LXI     H,WDIRDAT
        CALL    $DADA.
        XCHG                    D=PNTR WIND DIR
        LXI     H,WDIR          WIND DIR DATA
        LXI     B,3             MOVE 3 BYTES
        INX     H
        DCR     B
        JNZ     .VA1
        ANI     FS.BIT
        RNZ                     IF ALL SWITCHES OUT
        STC
        RET                     SWITCH IN, DATA BAD

DECODE  EQU     *
        ANI     37Q             TRIM
        LXI     H,DIGIT-1
        MVI     B,13Q           11 CHARS MAX
DCD1    INX     H               POINT TO PATTERN
        CMP     M               MATCH?
        JZ      DCD2            IF YES
        INX     H               POINT TO DIGIT
        DCR     B
        JNZ     DCD1            LOOK AT NEXT
DCD2    INX     H               POINT TO DIGIT
        MOV     A,M             GET IT
        STAX    D               STORE IT
        RET

DIGIT   DB      0,' '
        DB      17Q,'0'
        DB      2,'1'
        DB      27Q,'2'
        DB      23Q,'3'
        DB      32Q,'4'
        DB      31Q,'5'
        DB      35Q,'6'
        DB      3,'7'
        DB      37Q,'8'
        DB      33Q,'9'
        DB      '0'

RDATA   DS      16                      15 BYTES +1
CDATA   EQU     *
        IF      FORMAT
DATE    DB      'Temperature Degs '
TMPU    DB      'F Outdoor= '
TMPO    DB      '+000  Indoor= '
TMPI    DB      '+000',12Q
        DB      'Pressure '
PR      DB      '00.00 '
PRU     EQU     *-4
```

```
        DB      'Inches and '
PRC     DB      'Falling',12Q
        DB      'Wind out of '
WDIR    DB      'NNE at '
WSPD    DB      '00 '
WU      DB      'Mi/Hr',212Q
ERCNT   DB      '00',212Q
        ELSE
TMPU    DB      'F,'
TMPO    DB      '+000,'
TMPI    DB      '+000,'
PR      DB      '00.00,'
PRU     EQU     *-4
        DB      'Inches,'
PRC     DB      'Falling,'
WDIR    DB      'NNE,'
WSPD    DB      '00,'
WU      DB      "Mi/Hr,',212Q
ERCNT   DB      '00'212Q
        ENDIF

PRUI    DB      '.   Inches'
PRUM    DB      0,'   Milbrs'
PRCR    DB      'Rising '
PRCS    DB      'Steady '
PRCF    DB      'Falling'
WDIRDAT DB      ' N NNWNNNW SSWSW  W WSW'
        DB      'NNENE  E ENE S SSEESESE '
WUM     DB      'Mi/Hr'
WUN     DB      'Knots'
WUK     DB      'Km/Hr'

SCFLG   DB      5
        END
```

MBASIC PROGRAM FOR H89/WEATHER STATION OPERATION

```
10 CLEAR 1000
20 OPEN "I",#1,"RT3:"           REM REAL-TIME CLOCK DRIVER
30 LINE INPUT #1,A$             REM GET TIME/DAY/DATE
40 IF MID$(A$,4,2)="10" THEN PRINT A$:GOTO 110  REM RECORD TIMES
50 IF MID$(A$,4,2)="20" THEN PRINT A$:GOTO 110
60 IF MID$(A$,4,2)="30" THEN PRINT A$:GOTO 110
70 IF MID$(A$,4,2)="40" THEN PRINT A$:GOTO 110
80 IF MID$(A$,4,2)="50" THEN PRINT A$:GOTO 110
90 IF MID$(A$,4,2)="00" THEN PRINT A$:GOTO 110
100 CLOSE #1:GOTO 20
110 OPEN "O",#2,"LP:"
120 PRINT #2,CHR$(10)
130 PRINT #2,A$                 REM PRINT TIME/DAY/DATE
140 PRINT #2,CHR$(10)
150 CLOSE #2
160 CLOSE #1
170 A=4:        REM BASE PORT ADDRESS FOR WEATHER STATION
180 B=A+1:C=A+2:CWR=A+3
190 OUT CWR,155     REM SET ALL PORTS IN
200 PRINT
210 P1=A:P2=B:P3=C
220 D1=INP(P1):D2=INP(P2)
230 OPEN "I",#1,"WX:"           REM WEATHER STATION DRIVER
240 LINE INPUT #1,A$            REM TEMPERATURE DATA
250 LINE INPUT #1,B$            REM BAROMETRIC DATA
260 LINE INPUT #1,C$            REM WIND DATA
270 PRINT:PRINT A$:PRINT B$:PRINT C$:PRINT:PRINT
280 CLOSE #1
290 OPEN "O",#2,"LP:"
300 PRINT #2,A$:PRINT #2,B$:PRINT #2,C$    REM PRINT WEATHER DATA
310 CLOSE #2
320 FOR N=1 TO 15000
330 GOTO 340
340 NEXT N
350 GOTO 20
360 END
```

PC89/PC12A

8255A PPI Line Assignment Worksheet

PLUG PA:

PPI2
MODE 0

Wire	Color	Pin	Line	Connected To: HEATH ID-4001 P202--Pin	Function
1	BROWN	14	A7	16	WIND DIRECTION SEL CODE D
2	RED	1	A6	15	WIND DIRECTION SEL CODE C
3	ORANGE	13	A5	14	WIND DIRECTION SEL CODE B
4	YELLOW	2	A4	13	WIND DIRECTION SEL CODE A
5	GREEN	12	A3	4	DIGIT SEL CODE D
6	BLUE	3	A2	3	DIGIT SEL CODE C
7	VIOLET	11	A1	2	DIGIT SEL CODE B
8	GREY	4	A0	1	DIGIT SEL CODE A
9	WHITE	10	C7		
10	BLACK	5	C6		
11	BROWN	9	C5		
12	RED	6	C4		
13	ORANGE	8	GND	25	GND
14	YELLOW	7	+5V		

PLUG PB:

Wire	Color	Pin	Line		
1	BROWN	14	C0		
2	RED	1	C1		
3	ORANGE	13	C2		
4	YELLOW	2	C3		
5	GREEN	12	B0	6	DIGIT SEGMENT (a)
6	BLUE	3	B1	7	DIGIT SEGMENT (b)
7	VIOLET	11	B2	10	DIGIT SEGMENT (e)
8	GREY	4	B3	11	DIGIT SEGMENT (f)
9	WHITE	10	B4	12	DIGIT SEGMENT (g)
10	BLACK	5	B5	5	DIGIT SEGMENT (dp)
11	BROWN	9	B6	18	FRONT PANEL SW.BUSS
12	RED	6	B7	19	REAR PANEL SW.BUS
13	ORANGE	8	GND	25	GROUND
14	YELLOW	7	+5V		

Address PORT A .BASE 04Q
PORT B .+1 05Q
PORT C .+2 06Q
C W R .+3 07Q

Direction GROUP A IN
GROUP C7-C4 IN
GROUP C0-C3 IN
GROUP B IN

Configuration 233 OCTAL
Command 155 DECIMAL

By: E.T.Wright Jr. Date: 11-12-82

NEW HUG PRODUCTS

P/N 885-1121 HDOS
Hard Sector Support Package $30.00

Introduction: The HUG Hard Sector Support Package is a collection of software designed to help users get the best use from a hard sector 5.25 inch disk system. It features the HSY.DVD device driver, an enhanced version of the original HUG SY: device driver written by Dean Gibson of UltiMeth Corporation. Several support programs are included to help the user test, duplicate, and modify disks.

Requirements: This two disk set requires the HDOS operating system, version 2.0, on an H8/H17 or H89 with 32K of memory. Only one disk drive is required, however two are recommended. A line printer is not required but is recommended due to the large amount of documentation files, which could be printed.

NOTE: *The H8 requires the extended configuration option or the Heath Z80 board to used double sided drives. The H19 terminal is not required.*

This package is a two disk set. Disk B contains the assembly source code for the files on Disk A.

Disk A

README	.DOC
HSY	.DOC
HSY	.DVD
HSY3	.DVD
HSY4	.DVD
SETDSK	.ABS
INITAUTO	.ABS
PRINIT	.ABS
DUMP	.ABS
DUMP	.DOC
SDUP	.ABS
SDUP	.DOC
TEST40	.ABS
TEST80	.ABS
TINIT	.ABS
COMBINE	.ABS

Disk B

README	.DOC
HSYDVD	.ASM
HSYINIT	.ASM
MFREADY	.ACM
MFDVD	.ACM
MFINIT	.ACM
SETDSK	.ASM
DUMP	.ASM
SDUP	.ASM
TINIT	.ASM
COMBINE	.ASM
ROMSUBS	.ACM

Authors:

HSY.DVD — Dean K. Gibson, modified by Patrick Swayne
SETDSK, SDUP, TINIT, COMBINE — Patrick Swayne
DUMP — HUG/Heath Staff, modifications by Patrick Swayne
INITAUTO, PRINIT, TEST40, TEST80 — Modifications of Heath/Zenith software. INITAUTO modifications by Dean Gibson.

HSY — The HSY.DVD device driver is a replacement for the standard 5.25 inch hard sector device driver provided by Heath, and offers many additional features. The features are supported under HDOS 2.0 without any changes except replacement of the existing hard sector device driver, and the disk drives, if higher capacity drives are to be used.

Some of the features are as follows:

A 35 percent reduction in time to load large single block files, such as MBASIC.ABS, and when copying such files with PIP.

The ability to SET the step rate, motor on time, and head delay time for each unit. A long head unload delay time allows the head to remain loaded during rapid multiple disk accesses, reducing head and media wear.

The step time is recorded in the boot track, resulting in faster booting for fast drives.

A media check can be performed during INIT, eliminating the need to run TEST17 just for that purpose.

Support for double sided and/or double track density (80 track, or 96 tracks per inch) drives, such as the H17-4. Single sided disks can be read and written in double sided drives, and 40 track disks can be read but not written in 80 track drives.

NOTE: *The H-8 requires the extended configuration option or the Heath Z80 board to use double sided drives).*

Improved error recovery, which temporarily increases the seek step time during error retries.

This new version is fully compatible with the Heath and other soft sector 5.25 inch disk device drivers.

HSY3 and HSY — These are 3 and 4 MHz versions of the device driver, for the programmable 3 and 4 MHz modifications presented in REMark. They will also work with conventional 3 and 4 MHz modifications, providing that the H17 ROM is also modified.

SETDSK — This program allows the user to change the boot configuration on a disk, so that he can make changes in the hardware without having to re-initialize the disks.

The current configuration of the drives is written into the boot track of any disk that is initialized with HSY.DVD. For example, if SY0: and SY1: are each 40 track drives, and SY2: is an 80 track drive, that information is written into the boot tracks of the disks. If the drives are changed, either by changing the programming jumpers or by replacing a drive with another type, the information in the boot track will have to be modified to the new system. The program SETDSK is provided for this purpose.

INITAUTO — This program is a modification of the standard INIT program. A disk initialized by it, when SYSGENed, will boot without prompting with the ACTION <BOOT> message.

PRININ — This is another modification of INIT that allows the user to make his own boot code patches, and then transfer them to another disk when INITializing. PRINIT will work even if the target disk is of another type.

DUMP — This program allows the user to patch any file or area on a disk by track and sector. It is a modification of the DUMP program which was released on the HUG P/N 885-1062. This version recognizes 80 track and/or double sided disks.

SDUP — This a disk duplication utility. It can duplicate any size disk supported by HSY.DVD, and can duplicate in one drive, if necessary.

The source disk can be placed in any drive capable of reading the disk (for example, a 40 track single sided disk in an 80 track double sided drive). The destination must be placed in a write compatible drive.

SDUP uses the device driver for all disk reading and writing, and has its own initialization routine for formatting the destination disk. Rather than copying a disk one track at a time as some DUP programs do, it reads as much as will fit in memory, and then writes it to the destination disk. To verify disks, it calculates a checksum of all data read on each pass, then reads the destination disk and calculates a checksum on its data and compares the two.

TEST40 and TEST80 — These are modified version of TEST17 that allow the user to test any kind of disk supported by HSY.DVD. TEST40 is for 40 track disks, and TEST80 is for 80 track disks. The improvements are outlined in the documentation.

TINIT — This is a special disk initialization program for preparing disks for TEST40 and TEST80. It initializes double sided disks with the same track numbers on each side so that each side can be tested independently. TINIT must be used if the user wants to test double sided disks/drives.

COMBINE — This program is required when the user re-assembles the device driver. Instructions are included in the re-assembling documentation.

Comments: This package is a must for any HDOS user who has high capacity disk drives. Some of the features will be of practical use to H17 type drives also.

P/N 885-1122 HDOS and P/N 885-1224 CP/M MicroNET Connection $16.00

Introduction: These packages provide the user with a User IDentification number to access the CompuServe timeshare system. The packages include a modem utility package, a CompuServe User ID and secret password to get on the system, plus some limited documentation to help the user get started using the system.

P/N 885-1122 HDOS Requirements: This package contains a single sided hard-sectored diskette that requires the HDOS operating system version 2.0 on an H19/H8/H17 with a four port serial card or H89 with 32k of memory.

P/N 885-1224 CP/M Requirements: P/N 885-1224 contains a single sided hard-sectored diskette which requires CP/M version 2.2 or later on an H19/H8/H17 or H89 with 32k of memory. Only one drive is required.

The P/N 885-1224-37 contains a single sided, 48 tpi soft- sectored diskette which requires CP/M version 2.2.03 on an H19/H8 or H89 with the soft-sectored controller card.

The soft-sectored P/N 885-1224-37 will execute on the CP/M-85 operating system of the H/Z-100 series computer.

General Requirements: Any of the above computer systems will need a modem capable of originate mode with FULL duplex at 300 baud operation. (This is a standard feature on most modems.) There are accoustic and direct modems. The user is responsible for choosing a modem which fits his particular system.

The following files are released on the HUG P/N 885-1122 HDOS hard-sectored disk:

README	.DOC
HTERM	.ABS
HTERM	.DOC
FIRSTIME	.NET

Refer to P/N 885-1089 in the new HUG Software Catalog for a description of the modem program HTERM.

The following files are released on the HUG P/N 885-1224 CP/M hard and soft-sectored disk:

README	.DOC
ZTERM8	.COM
ZTERM89	.COM
ZTERM100	.COM
ZTERM	.DOC
FIRSTIME	.NET

Refer to P/N 885-3003 in the January 1983 issue of REMark for a description of ZTERM.

Package Description: This package has been put together to allow any member of HUG to get on the CompuServe timeshare system as soon as he/she receives the package, provided he/she has a modem as explained above.

The CompuServe User ID and secret password are sealed in an enclosed envelope in the package. The user is responsible for filling out and mailing the "Service Continuation/Request and Agreement" form to CompuServe.

CompuServe is a large timeshare data base system that has many areas of service, information, interest and fun. The HUG Special Interest Group (SIG) or Bulletin Board, if you will, is a very small part of the entire system, however, the Bulletin Board (BB) is a large system in itself. The member can leave, retrieve, search, scan, and reply to messages on the BB. In addition, the HUGBB has one of the largest, if not the largest, data base on CompuServe for the member to download files from the host or upload files to the system for others to download.

To access CompuServe the user must have a telephone number in his/her area that links into CompuServe. There are direct numbers, TYMNET, and TELENET numbers that access CompuServe. TYMNET and TELENET have a surcharge per hour over the cost of CompuServe.

NOTE: *To find out if you have a telephone link in your area, call the CompuServe Customer Service Toll Free number 800-848-8980 or (614) 457-8650.*

The FIRSTIME.NET file on the disks is a file of what the user will see the first time he/she links into CompuServe (the host computer). This file should be studied before going on CompuServe to help in understanding what the host timeshare system is doing. The sample link will show how to get to the HUG BB as well as some other options, which are significant to HUG members.

Documentation about the system is available from CompuServe for an additional charge:

1) CIS (CompuServe) User Guide

2) Personal Computing Guide
3) Special Interest Group (SIG) Manual

Refer to FIRSTIME.NET for help in ordering documentation and prices while on the system or contact CompuServe directly.

Special NOTE: *CompuServe charges are $5.00 an hour for regular hours and open areas. Any member of the HUG Bulletin Board (or SIG) receives a 10% or $.50 an hour discount for the time spent while on the HUG Bulletin Board.*

The user may already have a modem package. HTERM and ZTERM are supplied in its respective package for the new users convenience. The source code is not included for these programs, but is available on the part number (mentioned above).

Comments: This package will introduce a user to the timeshare system of CompuServe and access to the features of the HUG Bulletin Board.

P/N 885-1223 CP/M
HRUN HDOS Emulator $40.00

Introduction: HRUN is a CP/M program that emulates the Heath Disk Operating System (HDOS). It allows virtually any non-hardware dependent HDOS program to be run on a CP/M compatable computer. For a more complete description of HRUN, see "HDOS For Everyone" in this issue.

Requirements: HRUN requires the CP/M operating system, version 2.0 or higher. It will run on H8, H89, Z89/Z90, and H/Z100 computers using Heath/Zenith, Magnolia, or other suitable CP/M. Since it uses no BIOS calls, it should also run on MP/M-II systems (HTOC is not MP/M compatable). At least 32k of memory is required. Two disk drives are recommended unless you have Heath/Zenith CP/M configured for one drive operation.

HRUN is distributed on three 5.25-inch disks containing the following files:

Disk A

README	.DOC
HRUN	.COM
HRUNT	.COM
HRUN100	.COM
HRUN100T	.COM
HTOC	.COM
PIP	.ABS
DIR	.ABS
MAKEDIR	.ABS
SUBMIT	.ABS
HELP	.ABS
SET	.ABS
FLAGS	.ABS
MAKESYS	.SUB
ERRORS	
BASIC	.ABS
EDIT	.ABS
ASM	.ABS
XREF	.ABS
PATCH	.ABS
XFORM	.ABS

Disk B

README	.DOC
HRUN	.ASM
H17ROM	.HEX

Disk C

README	.DOC
HTOC	.ASM
DIR	.ASM
MAKEDIR	.ASM
SUBMIT	.ASM
HELP	.ASM
XFORM	.ASM
HOSDEF	.ACM
HOSEQU	.ACM
ESVAL	.ACM

Authors: PIP, SET, FLAGS, BASIC, EDIT, ASM, XREF, and PATCH are Zenith Data Systems programs. HTOC is by Bob Mathias, William W. Moss, and P. Swayne. All others are by P. Swayne, HUG.

Program Contents: HRUN, HRUNT, HRUN100, and HRUN100T are pre-assembled versions of HRUN with various parameters set. The letter T in the name means that tabs are expanded to spaces by HRUN's printer driver. The number 100 in the name means that HRUN is assembled for use on H/Z100 computers, or other computers that do not allow level 2 interrupts (RST 2). (Details of HRUN are explained in the article "HDOS For Everyone".)

HTOC is a program that copies files from HDOS disks to CP/M disks. Programs that are to be run with HRUN must reside on a CP/M disk.

PIP, SET, FLAGS, BASIC, EDIT, EDIT, ASM, XREF, and PATCH are programs normally supplied with HDOS.

DIR, MAKEDIR, SUBMIT, HELP, MAKESYS, and XFORM are utility programs supplied with HRUN. They are explained in the article "HDOS For Everyone" in this issue.

ERRORS is a list of HDOS system error messages.

HOSDEF, HOSEQU, and ESVAL are files required to assemble some of the source files on disk C.

Comments: HRUN opens most of the HUG Software Library to all CP/M users, including the H/Z-100 series.

P/N 885-3004-37 ZDOS
ZBASIC Graphic Games Disk $20.00

Introduction: This ZBASIC graphic games disk has a number of games which have been modified to use the special commands of the powerful ZBASIC graphics.

Requirements: This disk requires the ZDOS operating system on a H/Z-100 series computer with a minimum of one disk drive. The programs require ZBASIC.

NOTE: *When ordering this disk, you must include the "-37" in the part number. All soft-sectored HUG disk products require a "-37" including ZDOS.*

The following programs are included on the HUG P/N 885-3004-37 ZBASIC Graphic Games disk:

README	.DOC
HANOI	.BAS
OTHELLO	.BAS
HANGMAN	.BAS
TICTAC	.BAS
SINK	.BAS
HEROI	.BAS

Program Authors and Descriptions:

For authors and abstracts to the above programs refer to P/N 885-1068 in the new HUG Software Catalog (page 22 & 23).

All modifications to ZBASIC have been done by Gerry Kableman.

(HEROI is the popular game NIM rewritten to display the HERO I robot sold by Heathkit.)

Comments: These programs show the features of ZBASIC and should spark interest in updating or writing of new software in ZBASIC.

Additional Products

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
885-0004	HUG Binder	$5.75

Introduction To Z-BASIC Part III

Gerry Kabelman, C.E.T.
Zenith Data Systems

This is the third article in a series of articles dealing with the new commands of the Z-100's Z-BASIC over BASIC-80. Previous articles dealt with the CLS, LINE, and LOCATE commands, plus using the 25th line. This month we will look at the PSET and DRAW commands using them to create objects on the screen.

The PSET command is a very useful command when addressing an individual dot on the screen. The syntax for using the PSET command is: PSET [STEP](x,x)[,attribute]

The PSET command is actually broken down into four parts; first the actual command PSET, the second being the STEP command, the third is the location (x,y) and the last is the attribute or color as used on the Z-100. The second (STEP) and the last parts are optional and may both be left out or used individually.

The PSET command will always use the first part of the command as it is the command starter. The second part (STEP) sets an invisible pointer, which I will call the Dot Pointer, to the last location pointed at by the PSET or any other command that uses individual dot addressing. If no command has used the Dot Pointer, the middle of the screen (319 dots from left edge & 112 dots from top) is the starting point.

The location coordinates (x,y) are the next part of the PSET command. The values for the X and Y coordinates may be any integar between -32768 and 32767. A number beyond this range will cause an overflow error. With numbers as large as 32000 the dot to be created will go off the edge of the screen, the normal range for the horizontal is 0 to 639 and the vertical is 0 to 224. When the larger numbers are used the dot simply is not displayed, however, when a move is made back into the normal screen display area the dot will appear. This gives the user the ability to make larger moves, for magnifying or other purposes, than would be possible on the screen. Large moves of this type are common in the DRAW and other commands which will be discussed in future articles.

```
EXAMPLES:    10 PSET STEP(0,0),7
             20 PSET (319,112),4
             30 PSET (26,3),6
```

The first two examples actually address the same location, because using the STEP command started at the center of the screen while line 20 goes directly to the location of the center of the screen and does not look at the Dot Pointer for it's location.

Line 30 is a location in the upper left corner of the screen twenty-six dots from the left and three dots down. This location will be where the DRAW command will start the drawing. Remember the Dot Pointer will change each time a command uses the individual dots, such as the PSET, LINE or DRAW commands.

The DRAW command is actually many commands used to draw lines, angles, or any other shape on the screen. The list below is a summary of the commands. The best way to learn how to use them is to USE THEM.

Un	Up	En	Up and right
Dn	Down	Fn	Down and right
Ln	Left	Gn	Down and left
Rn	Right	Hn	Up and left
Mn	Absolute or relative	Cn	Set color
Bn	Invisible movement	Sn	Set scale factor
Nn	Move but return to starting point	Z	Execute substring
An	Set angle		

The above list may look complex but let's start by just trying a couple of the commands and see how they work.

```
30 PSET(26,3),6:DRAW"F3R3G3D3H3G3U3H3R3E3"
```

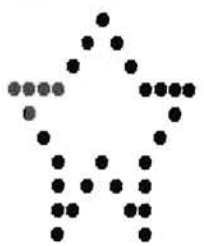
That look's too complex, but wait let's take a closer look. First we set the Dot Pointer to the horizontal dot twenty six right of the left side of screen, vertical dot three down from the top of the screen, and turn the dot to yellow (6). Then we use the DRAW to start our draw and what is inside the quotes is the series of drawing commands. Look at the chart above and follow through the string. The F moves down and to the right, in our case three dots, then R moves to the right three dots, then G moves down and left three dots, then D moves down three dots and so on.

Ok, that lines, looks a little confusing, let's break it up.

```
30 PSET (26,3),6:DRAW "F3 R3 G3 D3 H3 G3 U3 H3 R3 E3"
```

The added spaces have no effect on the results of the above line and are recommended for the first time user of the DRAW command.

The results of line 30 will be the star at the right. It is not a perfect star, however, it will do for the purposes of learning how to use the DRAW command.



The commands in the left column above are simple direction commands while those on the right have special uses as explained below.

Mn - Absolute or relative moves are made when using this command. If the X and Y coordinates are preceded by a + or - sign the value of X and Y are added to the current Dot Pointer, if not then the location is the absolute point on the screen.

```
EXAMPLE:        DRAW "M123,2"
```

Draws a line from the current Dot Pointer to the location one hundred twenty three dots from the left margin and two dots down from top of the screen.

Bn - Invisible move precedes a movement command and will allow the move without changing the dots that have been moved through.

```
EXAMPLE:        DRAW "BR10"
```

This command does NOT draw a line but rather moves the Dot Pointer to a new location.

Nn - Move but return to starting position does exactly that, in that it does the move requested and then returns the Dot Pointer to the location it was in before the move.

EXAMPLE: DRAW "NL30"

Draws a line from the Dot Pointer LEFT thirty dots and does NOT move the Dot Pointer.

An - Sets the angle at which the DRAW is to work from. The normal mode is to draw right when a R command is requested. If an angle of one ("A1") is done before the R is requested the right command will now draw up and the left draws down. If an angle of two ("A2") is done the right command draws left and the left draws right. The up and down commands follow along with right and left commands. The angle command must use an integer from 0 to 3.

EXAMPLE: DRAW "A1R40"

Draws a line from the Dot Pointer UP forty dots. ALL DRAW commands from this point on will be offset by 90 degrees counter clockwise, until another angle change is made.

Cn - Sets the color that is to be used by the DRAW command until it is changed again either by using the C again or another color changing command. The colors range is in integers from 0 to 7.

EXAMPLE: DRAW "C4"

Changes the color to RED.

Sn - Sets the factor for drawing on the screen. Four is the default for the factor setting. The range is from 1 to 255.

EXAMPLE: DRAW "S40R10S4"

Draws a line from the Dot Pointer to the right one hundred dots ((40/4)*10=100) and then returns the factor to the default (S4).

X - Executes a substring within a string command when using the DRAW command.

```
EXAMPLE:  D$="R10U30L40":DRAW "R30D10XD$;U10L20"
          DRAW "R30D10"+D$+"U10L20"
```

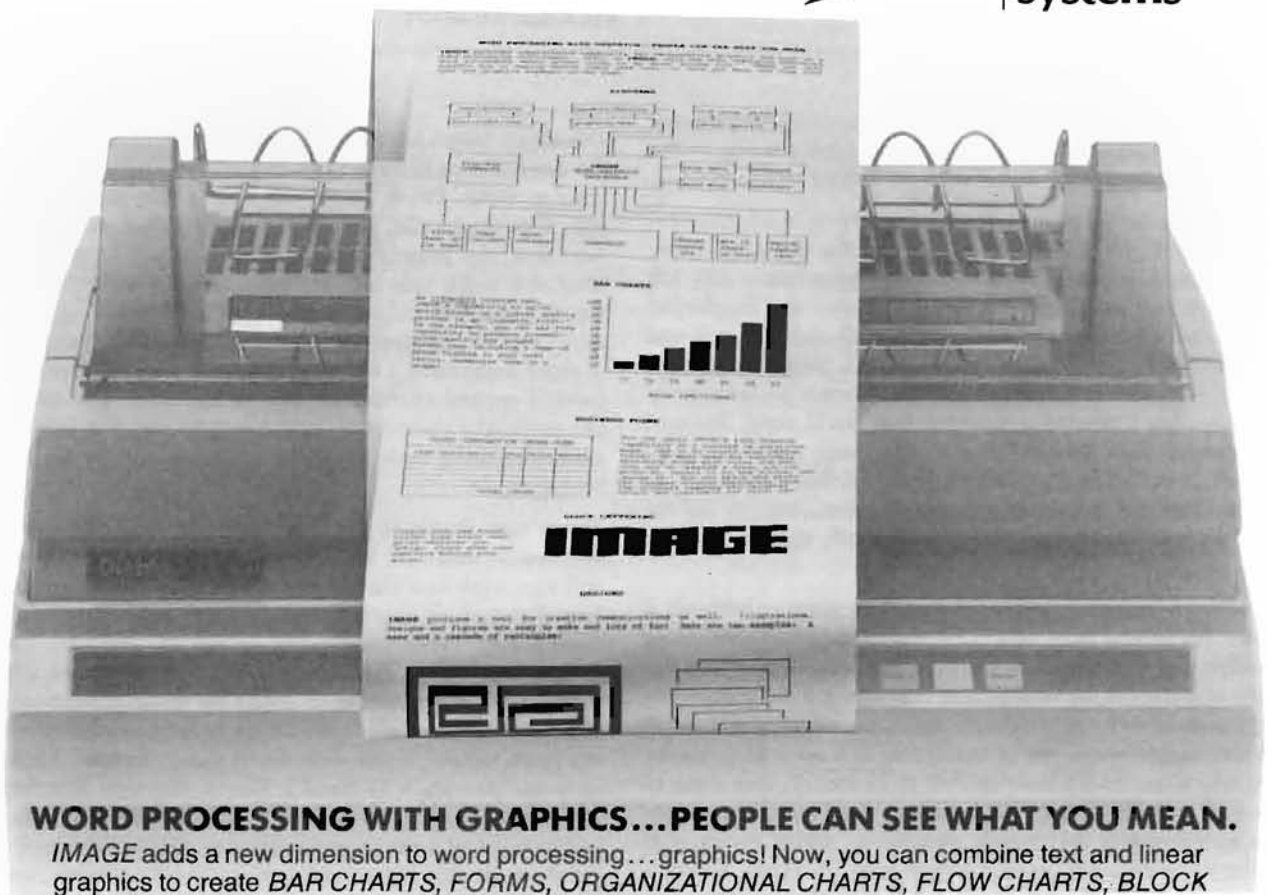
Draws a line to right 30 dots, down 10 dots, RIGHT 10 DOTS, UP 30 DOTS, LEFT 40 DOTS, up 10 dots and left 20 dots from the Dot Pointer. The second line in the above example does the same thing as the first line except it takes two additional characters.

Below is a program which uses all of the DRAW commands and the PSET command in the absolute mode. Try the program and take a very close look at line 50 as it is the heart of the program, also note line 60 which uses the INKEY$ and the LIST commands.

```
10 '      DRAWDEMO.BAS        12.28.82      GK:
20 CLS:S$="F3 R3 G3 D3 H3 G3 U3 H3 R3 E3"
30 LOCATE 25,10:PRINT"Press any key to stop!"
:
40 PSET(319,5),C:C=C+1:IF C=8 THEN C=1
50 DRAW"S30XS$;BM319,50D10S9XS$;C"
                              +STR$(C)+"M-0,-34A2S4XS$;A0"
:
60 A$=INKEY$:IF A$="" THEN 30 ELSE LOCATE 14,1:LIST
```

WORD PROCESSING IS NOW MORE THAN PROCESSING WORDS.

ANNOUNCING **IMAGE**™ FOR ZENITH data systems



WORD PROCESSING WITH GRAPHICS...PEOPLE CAN SEE WHAT YOU MEAN.

IMAGE adds a new dimension to word processing...graphics! Now, you can combine text and linear graphics to create *BAR CHARTS, FORMS, ORGANIZATIONAL CHARTS, FLOW CHARTS, BLOCK LETTERS,* and much more. The result? More powerful and more effective communication. People can *SEE* what you mean.

TOP MARKS FROM INFOWORLD

IMAGE won top marks from *INFOWORLD* for its innovation, quality, reliability, and ease of use:

"*IMAGE* certainly deserves accolades in the *performance* category."

"The *documentation* is simply superb. It is professionally done from cover to cover."

"Without a doubt, this program is *easy to use*."

"The program is *bombproofed* so well that I had trouble finding any errors."

IMAGE is a trademark of MicroArt Corporation.

InfoWorld
Software Report Card

Image

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	■
Documentation	☐	☐	☐	■
Ease of Use	☐	☐	☐	■
Error Handling	☐	☐	☐	■

All quotes are from InfoWorld's *IMAGE* software review, by Marty Petersen, June 14, 1982.

EXTRAORDINARY VALUE...

"The modest $295 Price Tag is a Bargain."

IMAGE runs on any Zenith Z/89 or Z/90 computer, on any Heath H/89 or H/90 computer, or on any Z-80-based CP/M system linked to a Z/19 or H/19 terminal.

TO ORDER CALL TOLL FREE:
1-800-MICROART (1-800-642-7627)
in Oregon, call: **1-692-3950**

OR WRITE:
MICRO ART
MicroArt Corporation
200 Market Bldg.
Suite 961
Portland, OR 97201

Mastercard, Visa and COD orders accepted.

Computer Romance
The Adventures of DaunWand!!

Laura Sparrow
(A.K.A. Jocelyn Griffin)
1421 West Huron
Ann Arbor, MI 48103

There it was— the letter from my publisher accepting the proposal for my third romance novel. A check would be coming in the mail, and it was clearly time for me to begin writing on a computer.

Not one of those facile writers who turn out a book in a fortnight— one draft, and it's off to the printers— I was the sort who wrote, and rewrote, and rewrote. The last book had been nearly 100,000 words, and it had gone through several revisions as I lengthened it by 50%, added characters and subplots, changed names and inserted dialogue... all in all, a process which took innumerable hours and destroyed one manual and two electric portable typewriters. Knowing I needed a machine that could stand fourteen hour marathons and wouldn't have to cope with eraser shavings in its works, I got an IBM Correcting Selectric, but even that was starting to show a rising number of peculiarities. And my bill for lift-off tapes was astonishing. Obviously, I still needed a better solution.

As a matter of fact, this wasn't the first time we'd considered a computer. For several years now my husband and I had been doing our reading and looking for a machine that would offer him the scientific applications he wanted and also give me the word- processing capabilities I was desperate for. I knew I wanted a green screen that would show me as much text at a time as possible, and 80 characters by 24 lines proved to be exactly one page of manuscript; enough memory for a program that could compensate for my after-thoughts and typing atrocities— and a system that would be so user-friendly even a person with absolutely no computer experience could learn to use it quickly. And so, one afternoon in June, 1981, we visited our local Heathkit store and came away with five cartons of parts and the stack of manuals for building our H-89.

In the next eight weeks, we built the H-89, added the two side floppies of an H-77, bought a Diablo 630 printer for the editors who won't read dot matrix manuscripts, and learned to run the whole system. By the end of summer, in fact, we were reasonably comfortable with both HDOS and CPM and in the market for a good word-processing program. Everywhere we turned, there was someone claiming his favorite program was superior to all the others, but we settled on MAGIC WAND with the Magic Menu to simplify learning it. The program also came with a printed strip to stick above the keyboard, assigning the various function keys to "forward page", "back line", etc. Attaching it, I started working my way through the excellent documentation, where I learned by editing a fractured version of the Gettysburg Address— as it might have been given by Bob Newhart— until it looked like Lincoln's text again. When I'd finished with that, I was comfortable with basics like moving the cursor from the keypad and had built up some confidence about using the program.

The Magic Menu added to my confidence, too, by giving me all the essential functions in plain English; I just had to type the first letter to E(dit), P(rint), C(opy) or K(ill) a file, B(ackup) an entire disk, R(ename) a file, get the S(tatus) of a disk, or list the F(iles). After that, the program asked another simple question or two— what disk a file was on, for example, or whether it was a new file. It also protected me from myself: before killing any files, it asked me if I was sure I really wanted to do that. More than once in those early days, I definitely was not, and I appreciated being given a second chance! In spite of the program's safeguards, I did eventually discover ways to lose text and that few things are more infuriating than losing the perfect phrase when it took an hour to create.

At any rate, thanks to the Menu I finally decided I was ready to start word-processing the new book. At first, I'd only enter Edit and then work as if the computer were a televised typewriter, but even that was an improvement over any kind of writing I'd ever experienced. Typos vanished in an instant, and I no longer had to install a new lift-off tape every four hours! Better still, fixing simple typos soon led me to trying the other corrections I could make. With the stroke of a key on the keypad, I could insert characters in an existing word; with the stroke of another, I could open an entire paragraph to insert a line of forgotten description and then hit that key again time to return the rest of the paragraph. Similarly, I had a third key for deleting single characters, a fourth (on the right side of the keyboard) for deleting words, and a fifth for deleting whole lines at once. —Better yet, that last one required two strokes before it would destroy a line, and with my bad aim I loved it.

In fact, I loved the entire program. Much to my surprise, I was finding that romance writers really didn't need to use quill pens, and on a rising tide of enthusiasm I started searching out more features of the MAGIC WAND that would help me cope with the problems I discovered when I'd finished the rough draft and began to go over my story. For example, I had gone way off the track with several long passages of description, and when I began rewriting I realized even a 100,000 word novel doesn't have room for everything. That led me to discovering Block Delete. Using the gray function key, I could mark the beginning and end of a passage to be scrapped, then go to the command screen and type BD; a moment later the program would tell me how many letters were included and ask for confirmation they were all to be deleted— again, that second chance! Block Insert, Block Move, and Block Print all followed quickly.

Another discovery came when my editor announced that the word "mistress" was old-fashioned, and I should use "woman" instead;

that one sent me to the blue function key for Search/Replace. Using it, I needed only seconds to locate and replace every "mistress" in a chapter, shifting the lines after the shorter word. And at the end, I was even told how many times it had been done— another feature I learned to rely on for counting uses of my favorite words, so I could stop wearing them out.

The WAND didn't just count words on a Search/Replace, either. Every time I used ESCape to return to the command screen, it automatically told me how many words of text now existed in the file I was working on. At long last I could tell exactly how many words I'd written, without trying to estimate the number on an average page, allow for partial pages and dialogue, add in the extra paragraphs I'd inserted in pencil or any of the other maneuvers I'd gone through in the past. And I could tell my editor exactly how long a manuscript was, without leaving her to send it down to the basement department whose miserable job is trying to figure that out. Every day I found more features to like about word processing and the MAGIC WAND.

Inevitably, of course, it wasn't all smooth sailing; I had my share of disasters in the first few months. Like most people, I learned about making frequent back-ups after the power failed, and I lost a whole scene I'd been struggling with. And more than once I managed to make my own little crises by doing things like panicking because I'd edited a chapter from B: to C: when I forgot to put a disk in C:— or trying to store an editing job on a disk that was almost full, because I hadn't remembered to check its status before I began working— or doing a series of rewrites without using different file names, so I lost track of which was the latest one.

Then, if I couldn't bail myself out with the documentation, it was time for a frantic call to Ray Massa at Studio Computers. He had originally sold me the MAGIC WAND, and over those next months I got more and more sold on it when he provided endlessly patient support. Thanks to them both, I was making terrific progress, writing faster and more easily than I had ever imagined it could be done. Even my typing speed was improving!

I was also getting more and more at home with the entire program— enough so, in fact, that I stopped using the Magic Menu and began working directly through CPM. The Menu had taught me an enormous amount, but it also used up 9K of RAM, in addition to the l6K for the rest of Edit. Most of the time that was no problem, but with a few of my longest chapters I wanted all the RAM I could get. The WAND let me use a Read/Write option to load text in segments, but I preferred not to interrupt myself for that— it always seemed to catch me in the exact middle of an important dialogue or (worse!) a love scene, when I wanted to be able to look back and forth through the text to check for continuity.

The manuscript as a whole was beginning to have some real continuity, too; because the WAND made it so easy to revise and rewrite, I could keep going over things until they were genuinely right, instead of accepting almost right because I was too tired to try again. But before I printed up a clean copy to send my editor, I had fifteen long chapters to proof-read, and it was at that point I discovered the terrific SPELL program Walt Bilofsky sells.

A dictionary that can recognize 50,000 words, it was able to read an entire chapter of 6000-7000 words in only a couple of minutes, checking every word and quietly ticking off the number read so far and the number questioned. The questioned ones were put on an alphabetized list, and for each word at that point, I had a choice of ignoring it if it was correct (like a flowery romance writer's term!), adding it or its root to the dictionary if I expected to use it often, or marking it in the text if it needed correcting. Each option can be chosen by typing a single letter: I, A or R, or M.

That last option, of course, was the most important to me, and whenever I chose it, the program marked the word in the text by using a # symbol to replace its last letter.Then all I had to do was exit from the dictionary and load the chapter with MAGIC WAND again, using Search/Replace to find a # and correct the error; the red function key, Repeat Search, took me to the next one. Much to my delight, the whole process took 5-10 minutes per chapter, rather than the hour or so I usually needed for proof-reading, and SPELL proved to be much more accurate than I am. — Its error rate is 2.2 missed incorrect spellings in 1000 errors; mine is probably five times that.

Proof-reading done, I was finally ready to print my clean copy, and for that job too the MAGIC WAND was ideal. Entering "print B:chap1", for example, took me to a display screen that let me establish margins and spacing, specify single sheet or continuous feed, select regular or proportional spacing, and— for that last— even justify the right margin, either by character or by word.By embedding a single line in the text at the top of each chapter, I could also direct the printer to number pages automatically and abbreviate the title on each one. I could even embed commands inside a chapter and tell the printer to underline for emphasis or to center chapter titles and boldface them. And all of this would be printed out at 35 cps, bidirectionally! I spent hours just hanging over the printer, watching it go and calculating how many eons I would have needed to do manually what it could do in minutes automatically.

Absolutely the only snag I ran into at that stage was learning editors won't read long manuscripts that come in printed on tear- off paper; the slight fringe that's left behind makes pages catch on each other. For a while I solved the problem by setting aside my tractor drive and hand-feeding single sheets of heavy typing paper. That, however, quickly lost its charms; hovering around with the next sheet ready wasn't a lot more fun than manual typing. The solution, though, turned out to be a high quality Xerox place. Copies are on paper that feels like heavyweight typing bond and, because the photocopying process expands the image very slightly, tear-off fringes are off the edges of the copy and don't reproduce.

Even after that, there was one last crisis— and it was the one I'd always feared. Going back compulsively to have one last crack at clarifying the hero's motives, I opened up the manuscript, tinkered around here and there— and watched my screen begin to fade erratically, then go completely dark. With a deadline staring me dead in the eye and no confidence we could locate and fix the problem fast enough, I considered moving to Timbuktoo and leaving no forwarding address. I also considered hara-kiri. Then I went back to our Heathkit store, carrying my dead beast.

Memory told me there would be the usual crowd in the store, and there was; reason told me there would be other computers in for repair, and I might have to wait for days. But knowing my machine was my livelihood, the store technician had a look at it as soon as the crowd cleared enough so he could put it down on a workbench. The trouble turned out to be on the power supply board, and in minutes he'd located it and installed the updated

parts the factory recommended for the problem. An hour after I trudged in, I floated out; a day later, with those last few chapters xeroxed again, the manuscript left by courier. Then I used Software Toolworks' PACK program to give me archival storage for my text; that way, a compressed copy of the book took only four disks instead of the eight it had originally needed.

Publication is slated for next spring— but it would probably be a year away if I were still depending on an ordinary typewriter.

Deciding to switch to word-processing, getting my Heath system, and choosing excellent programs for the job have changed my writing forever, freeing me from the mechanics of my trade so that I can concentrate solely on the creative aspects of it. Looking back only eighteen months, I can't imagine now what I ever did before Heathcliffe —any more than I can imagine a romance novelist calling her H-89 by any other name.

Venting Hot Air

Or How To Cool Off Your '89

Bob Small
354 Teakwood Dr.
Satsuma, AL 36572

During the last several years there have been several articles in various publications on ways to increase the cooling efficiency of the H/Z-89. The usual answer was either to install a new fan or cut new air vents in the back of the cabinet.

The biggest problem that I can see with the current design is that there are two vent ribs positioned directly over the tips of the fan blades. The back portion of the cabinet also overlaps the fan blade tips. It is at these very points that there is supposed to be maximum air flow. The center of the fan blade, of course, has a near zero air velocity. You can literally feel the air flow problem. The air movement is a 45 degree angle which indicates an obstruction to the required perpendicular angle. The simplest solution would be to cut out all the"ribs" above the fan housing, but this would expose the fan to possible damage from pencils, paper clips, etc. and might expose"little" fingers to some pain.

I have recently made a modification to my H-89 that has increased the air flow over the power pack without any extensive mutilation of the cabinet.

After disconnecting the cabinet top, I used a hack saw blade (with fine teeth) and a coping saw blade to extend the current rib slots towards the back of the cabinet 1" (see diagram). Then I cut out the first and sixth ribs (the ones positioned over the fan blade tips). I saved one of these ribs for later use. Turn the cabinet over and with either a sharp knife or a Dremel hobby tool very carefully cut off excess plastic from the bottom edges of the ribs. The Dremel tool must be used very carefully because the friction will melt the plastic. This last modification will help channel the air flow better through the"ribs".

The piece of saved ribbing was then cut into two small pieces and glued cross ways in front of the previously cut first and sixth ribs. To make the job look a little neater, I used some flat black modeling paint to paint the insides of the ribs.

Removing these two ribs should increase the air flow about 30%. The air flow is noticeably faster and the temperature is several degrees lower plus the air flow is almost perpendicular. I hope this information can be useful to other HUGgies.

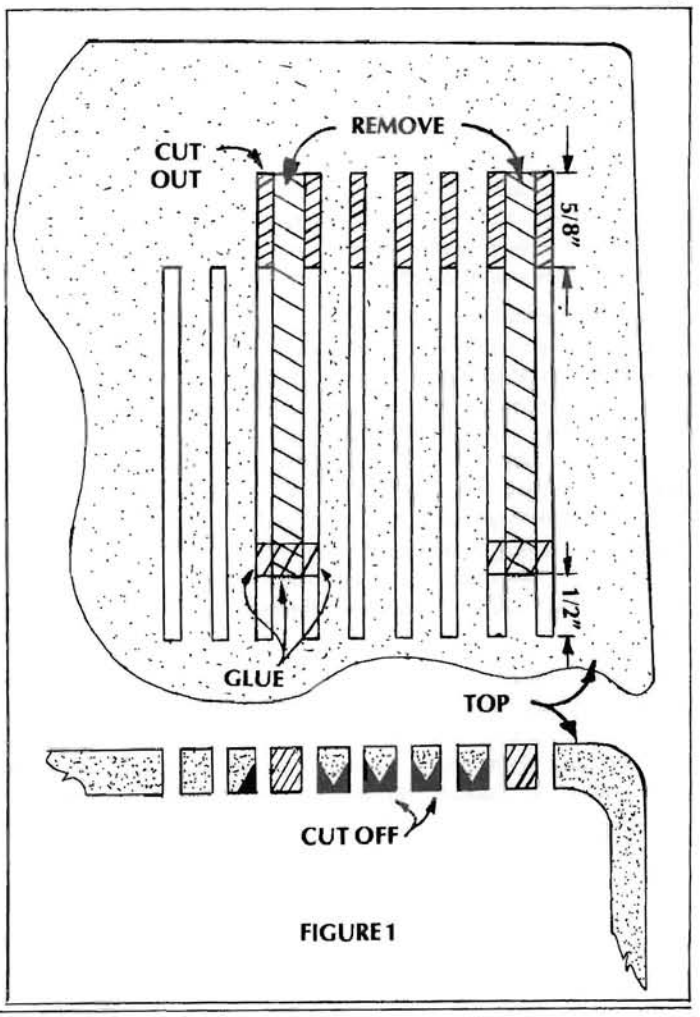
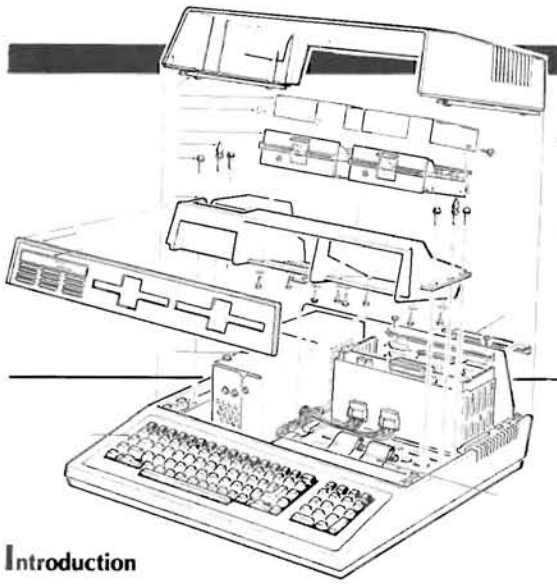


FIGURE 1



The H-100 Kit in Review

Terry Jensen
Software Developer

Introduction

The H-100 computer kit series is now available from Heath Company and has been announced in the Winter 1983 Heathkit catalog. These computer kits are the same units as the assembled Z-100 units. The H-100 is the low-profile version and the H-120 is the all-in-one kit version.

Both H-100 kits come standard with the 8088 and 8085 (16 and 8 bit) microprocessors with 128K of RAM and one 5 1/4 inch double sided soft-sectored 48 tpi disk drive. Both computers have the professional keyboard and they both support the 640 by 225 high-resolution pixel video screen. The H-207 disk controller card will support 8 inch drives as well as 5 1/4 inch drives. Each computer has two RS-232C serial ports, one parallel port, a light pen port, and the five slot S-100 expansion bus.

The H-100 series will run either CP/M or ZDOS, along with any other program that will run on the Z-100 series. The kit contains a ZDOS ZBASIC demonstration disk, which shows some of the features of the high-resolution graphic screen.

The H-100 & H-120 Kits

The H-100 and H-120 kits are almost identical with the primary difference being the installation of the high-resolution 12" diagonal monochrome video display in the all-in-one kit. The power supply units for the H-100 and H-120 kits are not the same size but the installation is similar.

The H-120 kit comprises two circuit board assemblies; the H-207 disk controller board assembly, and the video board assembly. The H-100 version has the H-207 disk controller board assembly only, because it does not have the built in CRT, which requires the video board. The mother board (the Main Circuit Board) and the Color Video Processor Circuit Board come assembled, tested, packed, and ready to install in both kit versions.

The H-207 disk controller board can simultaneously control as many as four 5-1/4 or 8 inch single or double sided disk drives using single or double density soft-sectored diskettes. The assembly and programming data of the H-207 is contained in a separate Assembly Manual. The appendix of the manual contains data information for the FD179X-02 Floppy Disk Formatter/Controller, the WD1691 Floppy Support Logic and the WD2143-03 Four Phase Clock Generator.

The Video Driver Circuit Board assembly is required for the monochrome CRT of the all-in-one version. The card will drive any of the three non-glare CRT screens for the H-120.

Neither the H-100 nor the H-120 come standard with the Z-219-1 Color Video RAM Chip Sets. Two Color Video RAM Chip sets are required to provide full color capability on a color monitor. The color chips are useful on monochrome video screens, by producing incremented levels of light intensity on the CRT display. (When the color chips are installed on the low-profile version, the H-100 is refered to as the H-110.)

An additional 64K of memory is available to upgrade the H-100 to 192K of RAM. Both the color chips and memory chips may be purchased and installed after the kit is assembled, however they are easier to install if purchased with the kit. This, of course, eliminates the disassembly of the computer to reach the mother board and video board.

A second 5 1/4 inch disk drive is available and may be mounted inside of the H/Z-100 computers to increase disk storage. This drive may be installed at any time.

The Assembly

The H-120 assembly consists of mounting the mother board and Color Video Processor board on the bottom of the chassis. Next the back panel, the S-100 card rack, the power supply, and the keyboard are mounted to the chassis. The disk drive subassembly, video driver board, and CRT screen are mounted to the front panel. Once the H-207 disk controller board is installed, the screen adjustments are made just prior to installing the front panel assembly to the chassis. The test and adjustment procedures are the last major sections of the kit. When the cover is in place the kit is complete.

The H-100 low-profile kit assembly is similar to the H-120 with the exception that the front panel assembly does not include the video driver board and the CRT subassembly.

Kit Statistics

The H-207 disk controller board assembly averages approximately 6.5 hours, with the majority of builders taking between 5 and 10 hours. The remainder of the H-100 low profile kit averages 4.3 hours to build the kit, with the majority finishing between 3.5 and 5.0 hours. The H-120 all-in-one kit took on the average 15.1 hours, with the majority of the kit builders finishing between 11 and 14 hours. (These statistics are provided by the Evalution Department of Heath Company.)

The H-100 low-profile weights just under 50 pounds.
The H-120 all-in-one weights under 65 pounds.

The H-100 low-profile retails for $2199.
The H-120 all-in-one retails for $2349.
The Z-205-1 memory upgrade retails for $180.
The Z-219-1 color video RAM retails for $160 per set.
The Z-207-3 5 1/4 inch (second) drive retails for $395.

(See the Winter 1983 Catalog or your local Heathkit Electronic Center for details and current prices.)

Personal Notes:

I built the H-120 kit with two 5 1/4 inch disk drives. I also had the 64K memory upgrade to 192K of RAM, plus the two color video RAM sets for full color capability.

The kit and instructions showed no major problems within the manual. There are, however a few observations that I would like to make.

The first comment I have is a caution note. The H-207 Disk Controller board is a high quality double sided plated-through hole, circuit card. On single sided boards, a good solder connection will flow up and around the solder lead on the foil side of the board. When soldering on a circuit board with plated-through holes, be aware that the solder flows through the hole and not up onto the lead. The tendency is to add more solder to get the high solder appearance on the foil side being soldered. This can cause major problems later if too much solder is applied. For example, if the component which is being soldered is an IC socket, the solder will follow the socket lead and fill the socket with solder. Then when it comes time to install the IC, the lead on the IC will bend due to the cooled solder in the socket. The solution is to use just enough solder to fill the hole with this type of board.

The second comment has to do with mounting some of the hex-head screws to the base. For some of the screws, I had to get a large ratchet to tighten the screws. This was frustration, in that I thought I might twist and break the mounting column, which fortunately did not happen.

Another observation; the manual calls for "cutting" the strips on the disk drive programming plugs. A better solution is simply to bend the leads so that they will not insert into the IC socket holder. In this way, the programming plug can easily be "adjusted" for another drive in the future by straightening and bending the proper leads.

One of the steps of the Video Adjustment section contains a simple eleven line ZBASIC program, which displays on the screen eight color bars. This program will be enjoyable to anyone who has the color chips.

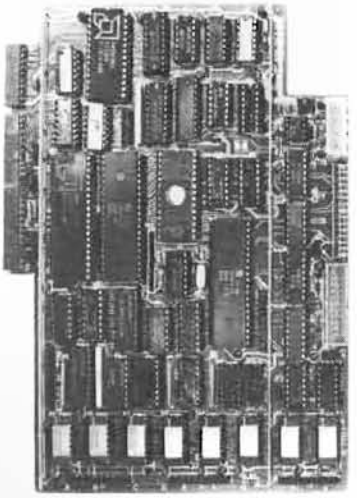
The slide rail for mounting the cover of the all-in-one, I found did not allow the top cover to install and remove quite as easily as I had expected.

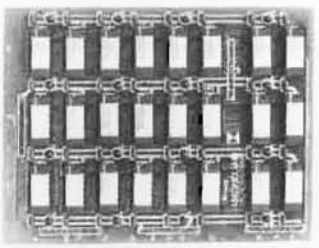
The construction of the H-120 consists of a steel chassis and back panel, with a cast fiberglass base, center base, front panel, and top cover. The machine weights far more than an H89, such that it takes care and effort to move the computer around.

Conclusion

The H-100 and H-120 kit manuals are written in the typical Heathkit easy-to-assemble style. As with any kit, there are the frustrating moments, but when the final assembly of either computer is complete, believe me, it is a pretty sight. Building the kits will be enjoyable for the beginner or advanced kit builder. Additional pleasure will come when the kit is complete and the builder can research the phenomenal potential of the H-100 or H-120 computer.

INTRODUCING -- MAGNOLIA's MOST POWERFUL BOARD YET!





The Z89/90 is a good solid computer. But it does have certain limits.

We've had many requests to increase its CPU speed to 4MHz, but nothing met our standards for both field installation and reliability. And who needs anything less.

We increased its RAM to 176K with our 128K board, allowing use of MP/M™. However, without DMA, multi-user operations under MP/M are slow.

Networking '89s together has been a long-term goal of ours. Through the Corvus Constellation multiplexer we provided the first stage -- allowing computers to share Winchester disk storage. But that technique leaves much to be desired -- since each computer independently manages its disk allocation there are significant operational limitations on sharing disk files.

We first announced our impending 'RS422 Network Interface' at the 1982 West Coast Computer Faire. Our design goals included:

- RS422 Network Communication at 800K baud
- On-board Network Control and Data Buffering

As it progressed, we realized that we had designed an 'I/O Board' which was FAR MORE POWERFUL THAN THE '89 ITSELF!

After agonizing delays, the board is now in production as the '77422 Network Controller'. Compare these specifications with the Z89 itself (or the new 'upgrade' CPU boards):

- 4MHz Z80™ CPU
- 64K RAM, 256K (bank selectable) optional
- 8 K EPROM
- 4 CHANNEL DMA controller
- 2 Serial Ports: RS422 at 500KBaud
  RS232 at up to 19.2 KBaud

We also changed its name from '77321 Network Interface' to '77422 Network Controller'. Our 773xx products must be used in a Z89/90 computer. Although this product CAN be an I/O card in an '89 or 90, it can also be used independently as a stand-alone station on the network:

- permitting a terminal (Z19 or other) as a network station (with local disk storage), or
- permitting a printer to spool data directly from the network.

Network applications software is currently being developed and tested.

Software for an exciting non-network application is included. The workload is split between the Z89/90 CPU and the CPU on the '422 board:

- The Z89's 2MHz CPU continues to run Magnolia CP/M™, handling all physical I/O (console, printer, floppy and Winchester disks)
- The '422 board's 4MHz CPU runs an interface program which 'looks like' CP/M to an application program -- but passes CP/M system calls to the Z89's CPU for execution, allowing a 63K TPA on the '422 board!

With this division, the following Z89 limitations are overcome:

- The application runs on a 4MHz CPU
- The application has 63K of RAM available, independent of BIOS size
- The DMA controller (making a slave I/O processor of the '89 board) frees the application CPU from detailed I/O functions.

This last advantage is especially significant when executing MP/M on the '422 board, using the optional bank-selectable memory.

Hardware-dependent programs (including utilities like 'format') which perform actual physical I/O, must run on the 'slow' Z89 CPU. A special utility is provided to 'mark' those application programs which can execute more efficiently on the '422 board CPU.

The board is presently available in 64K and 256K versions which mount in an I/O slot (P504/P510) on the '89 CPU board. To limit power dissipation within the '89, no disk drive may be used internally.

64K Network Controller Board order 77422-064-xx **$695**

256K Network Controller Set order 77422-256-xx **$1295**

CP/M Operating System by MMS if needed order CPM-xx add **$100**

MP/M II Operating System requires 77422 order MPM-xx **$595**

CP/NET™ Network Software requires 2 or more 77422 boards, one with MPM order CP/NET-xx **$450**

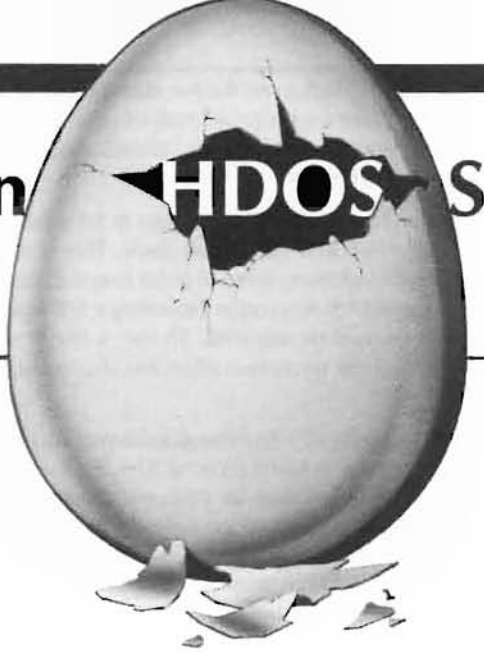
Use 'xx' to specify preferred media format

MAGNOLIA MICROSYSTEMS

2264 - 15th W • Seattle, WA 98119

(206) 285-7266 (800) 426-2841

An HDOS Shell for CP/M



Doug Alsip
8 Mimosa Drive
N. Cape May, NJ 08204

How would you like to be able to write a program that will work under both HDOS and CP/M? Or, would you rather just convert an existing HDOS program to run under CP/M? This project started one day when I was discussing personal computer matters with a friend (we both own H89's) when he said "You know, Doug, the only things we use HDOS for anymore are ADVENTURE and Diagnostics (no offense meant to HDOS fans). We ought to use the HDOS to CP/M file transfer utility to convert these things over." I scoffed at that, saying "It's much harder than a mere file format conversion; the HDOS and CP/M system calls differ like night and day. It would be a huge task to convert Adventure to run under CP/M."

However, as days went by, the idea just wouldn't go away. I realized that I'd never seen the original Adventure available for CP/M, and I thought it would be an interesting exercise to make it work under CP/M. Therefore, I undertook the conversion, which ended up being a one-week project. I am told that REMark readers would be interested in finding out how to do it, so here goes....

I'm going to make a long story short and simply explain the steps required to convert an HDOS program to CP/M. I treat the HDOS program like a "Black Box", and implement an HDOS shell under CP/M. You can use the information I present to write your own HDOS programs that will also run under CP/M with little or no modification! Those of you who are moving on to the Z-100 can use the same ideas to do the same thing for ZDOS (or MSDOS) and CP/M-86. For this operation to be feasible, you need the following tools:

1) HDOS2CPM, an HDOS to CP/M-80 file transfer utility (see below)

2) MACRO-80, by MicroSoft (or another suitable assembler, see below)

3) LINK-80, by MicroSoft (needed only because MACRO-80 only outputs .REL files)

4) DDT, by Digital Research (Comes with your CP/M package; or any suitable Debugger. SID/ZSID are fine; all that is required is to be able to read in a .COM file with an offset and make changes to it.)

5) An HDOS "shell" for CP/M 2.2, included as Listing 1. This is the heart of the conversion process; it provides a friendly environment under which the HDOS program (Adventure) can run under CP/M.

6) Anyone who tries to do this or any other assembly language programming task without the aid of a screen editor; you have my extreme sympathy. Get one before you lose your mind.

STEP 1: Do a file format conversion on the files ADVENT.ABS, NEWGAME.CAV, and ADVENTUR.DTB. This is by far the easiest part. You do this with the help of HDOS2CPM (HDOS to CP/M file transfer utility), a program written under CP/M (HUG P/N 885-1201A). **** NOTE: The original version of this program was designed for a dual drive H17-type system with the HDOS 1.6 Directory Structure. If you have another type of drive or you don't have access to HDOS 1.6, perhaps a later version of this program is available to you. (You can make the version I used work with other drives, but you must do some "doctoring" and you must be careful.) **** Follow the prompts provided by the program. If you are asked if you want ASCII file translation, say "No". Don't do anything more to the files NEWGAME.CAV and ADVENTUR.DTB. By the way, any other .CAV files that you might have saved up can be transferred over in the same manner.

STEP 2: I am going to treat the Adventure program as a "Black Box". That is, I am going to leave it as intact as possible, modifying only the default filenames and taking out one subroutine call. After first RENaming ADVENT.ABS to ADVENT.COM, use DDT to perform some minor operations on the file ADVENT.COM. Bring it in: "DDT ADVENT.COM<CR>". Now Move it down 8 bytes: "M108,23FF,100". This gets rid of the HDOS file header present on all .ABS files. (By the way, if you ever want to convert an HDOS file that doesn't execute at 2280H, you must properly account for this. That is, you must run it at the same address at which it used to run under HDOS. The alternative is to fully disassemble it; this can be a complex procedure and is beyond the scope of this article. I've never seen an HDOS .ABS file that doesn't execute at 2280H.) Now, using the Set command of DDT, alter the filenames handled by Adventure as per the following table:

ABS Addr	Before	After
055EH	'SY0:NEWGAME.CAV'	'A:NEWGAME.CAV',0,0
05A4H	'SY0:NEWGAME.CAV'	0,0,'A:NEWGAME.CAV'
0638H	'NEWGAME',0,'SY0CAV'	'A:NEWGAME.CAV',0
083DH	'SY0:ADVENTUR.DTB'	'ADVENTUR.DTB',0,0,0,0

Now, make two changes to the code of ADVENT using DDT's A or S command:

	Before		After	
ABS Addr	Hex	Mnemonics	Hex	Mnemonics
05F2H	CD	CALL $MOVE	00	NOP
	AA		00	NOP
	18		00	NOP
010CH	2A	LD HL,(201B)	2A	LD HL,(000B)
	1B		0B	
	20		00	

Finally, get out of DDT and "SAVE 35 ADVENT.COM".

STEP 3: I now describe the "meat" of the conversion process: Providing a friendly environment under which an HDOS program can run. This "shell" makes CP/M act like HDOS while the program is running. This basically involves "trapping", or intercepting all HDOS System Calls and performing the actions that the program expects.

HDOS programs interface with the operating system through the use of "software interrupts". These are just like hardware interrupts, but are generated by the software. Specifically, the execution of an RST 7 instruction provides the systems programmer with a means to "talk to" the operating system. The single byte following the RST 7 instruction tells HDOS which of its many functions you want to perform. You must previously set up some of the 8080's registers, which pass data into a System Call. This varies depending on the function and is adequately explained in the HDOS System Programmer's Guide. The RST 7 instruction causes a CALL to location 38H in memory. At cold boot, HDOS places a Jump instruction pointing to itself, so it can service the software interrupts that occur.

The shell I describe here is designed to tag along with an HDOS program, and in fact, a portion of it must be executed first when the program is loaded. The initialization code places a Jump instruction at 38H, sets up the stack pointer, and then jumps to 2280H. This jump instruction insures that all HDOS Syscalls are processed by the shell. The rest of the shell converts the HDOS Syscalls into appropriate CP/M Syscalls while reordering register information as necessary. The shell in Listing 1 doesn't implement all of the HDOS SCALLs, just those used by Adventure (note that this is a considerable subset of HDOS). Those readers who understand what's going on with the shell concept should have no trouble implementing any system calls they might need.

A secondary portion of the shell duplicates some of the H17 ROM code. As most of you Assembly Hackers must know, the H17 ROM contains several general purpose subroutines. The code for these is listed in your HDOS manual and isn't duplicated here. Since CP/M version 2.2 locks out the H17 ROM upon cold boot, my shell contains code that Adventure uses out of that ROM.

To use the shell, you must get it into your machine in source code form (type it in, or whatever). I apologize to any Intel Mnemonics fans out there. I never could "think" in Intel Mnemonics; Zilog's Mnemonics are more logical and my best Assembler, Debugger and Disassembler all use Zilog Mnemonics. However, I used the 8080 subset of Z80 code, so it will work on the 8080 or 8085. Also, it is a simple matter to translate into Intel Mnemonics, if so desired. Use an assembler/linker to convert it into a .COM file. The following commands will do the job with MACRO-80 and LINK-80:

```
M80=SHELL<CR>
L80 SHELL,SHELL/N/E<CR>
```

Now that you have the shell in a .COM file on disk, do the following:

```
DDT SHELL.COM<CR>
IADVENT.COM<CR>
R2180<CR>
G0<CR>
SAVE 69 ATEST.COM<CR>
```

The above commands bring SHELL into memory under the debugger, concatenate ADVENT to it, and save the aggregate as the file "ATEST.COM". If you've gotten this far and are hopelessly lost, I'm prepared to offer you a hint: The Shell I describe here has nothing to do with a certain "Enormous Clam Shell with its lid tightly closed". Why another file name? Well, now the fun starts: It's time to test it! Unless you're brave, place a write protect tab on your diskette. If you made any mistakes, you'll probably hear "chewing noises" in your disk drive as soon as you hit "go".

The astute reader will notice that there seems to be a lot of wasted space between the Shell and Adventure itself. This is due to the "Black Box" approach I chose to take. In order to reduce this wasted space, a full disassembly of Adventure including a full examination of its data structure would be required. To me, a few K savings in code does not warrant the enormous effort this disassembly would require.

In the remainder of this article, I go through each section of the shell, briefly explaining its more esoteric aspects. The Shell starts off with its initialization section. This must be executed prior to the HDOS program. It sets up the stack, sets up the software interrupt "vector", and then jumps to the HDOS program. The "SYS" section is executed upon any RST 7 software interrupt (ie HDOS SCALL). It first gets the SCALL parameter (one byte) and does some initial conversion to CP/M system call codes. This aids me in consolidating read and write routines into a single routine later on. At around SYS5, I index into a "lookup" table of SCALL handler addresses after saving the user's return address.

All SCALL handlers jump to SRET, which simply gets the user's return address back, and returns to it. STABL is the SCALL address lookup table. Following that is SERR, which prints an error message for undefined SCALLs. If you decide to implement a given SCALL, simply remove its name from the list above SERR and include your own handler named "scallname". (i.e. the same name you removed!) The various HDOS SCALL handlers follow. I explain only the unstraightforward ones here.

My SCIN implements only the "line mode with echo" method of console input. This is the one used by Adventure. The GLINE subroutine gets a line from the console whenever SCIN has used all characters in the previous line. It should be a simple matter to generalize these routines and add support for CONSL, which I currently ignore.

OPENR and OPENW are performed by a single routine. OPENW is just like OPENR except it DELETEs and CREATEs before it OPENs. CP/M does not discriminate between an open for read and an open for write. A word about the differences between HDOS and CP/M in the handling of filenames is in order. HDOS names its drives as SYn and DKn. CP/M uses A..P (not discriminating between types of drives; this is left as an exercise for the BIOS). The main part and the extension of the name are identical between the two operating systems. HDOS has a couple of features regarding file names that CP/M doesn't have: The .NAME SCALL, which gets the full filename from a channel including the drive it came from, and the use of default drive and extension. CP/M, in this regard, has the "Get Current Disk" System Call and the general concept of the "Current Disk". The latter is the one CP/M programmers usually utilize in providing a default drive on which to expect to find a given file. I avoided the issue in my shell and chose to change the default filenames in the converted Adventure program; this was a simple matter. Essentially, this means that the file ADVENTUR.DTB must be on the "Current Disk". However, it doesn't look too difficult for one to provide a .NAME SCALL, or a default block within the framework of my shell.

FFCB (Find File Control Block) translates HDOS channel numbers into CP/M File Control Block (FCB) addresses. FNFCB formats filenames into the form required by CP/M in its FCBs. RREAD and WRITE are also implemented in a single routine. They perform two CP/M random read/writes for each HDOS sector involved. POSIT

manipulates the CP/M random record pointer in the FCB associated with the given channel. ERROR prints a small subset of the HDOS error messages. This can be expanded at will, or taken to a disk file a la HDOS.

The last portion of the shell consists of duplicated H17 ROM code. Listing 1 doesn't show the code; you can find it in the HDOS Reference Manual. Note that these routines must assemble at the same addresses as they did under HDOS. This is again due to my "Black Box" approach. The system stack space is next, followed by the HDOS program.

Well, that's all there is to it! I hope this article stimulates some of you "bit bashers" out there.

About the author...

Doug is a Lieutenant in the US Coast Guard working at the USCG Electronics Engineering Center in Wildwood, NJ. He graduated from the USCG Academy in 1975 and received his Masters' at MIT in 1979. Doug's home computing interests center around his H89, H37 and H14; plus fooling around with all hardware and software aspects of his system.

Vectored ☞

```
                SUBTTL  CP/M Shell for Adventure
                .Z80                    ;(C) 1982 by Doug Alsip
                ASEG
                ORG     100H

                ;
                ;
SCALL           MACRO   ?CODE
                DB      0FFH
                DB      ?CODE
                ENDM
                ;
                ;
NL              EQU     0AH
BL              EQU     07H
LF              EQU     0AH
CR              EQU     0DH
BASE            EQU     0
BDOS            EQU     BASE+5
TICONT          EQU     BASE+0BH
SVEC            EQU     38H
                ;
                ;
START           EQU     $
                LD      SP,STACK        ;SET UP STACK POINTER
                LD      HL,SYS          ;POINT TO SHELL ISR
                LD      A,0C3H          ;8080 JUMP INSTR
                LD      (SVEC),A        ;RST 7 VECTOR
                LD      (SVEC+1),HL     ;PUT IN OUR ADDR
                JP      ADVST           ;GO RUN HDOS PROGRAM
                ;
                ;
SYS             EQU     $
                EX      (SP),HL         ;SAVE HL, POINT TO PARAM
                PUSH    AF
                LD      A,(HL)          ;GET SCALL PARAM
                CP      4               ;READ?
                JP      NZ,SYS2
                LD      A,33            ;USE CP/M RANDOM READ CODE
                LD      (RTYP),A
                JP      SYS5
SYS2:           CP      5               ;WRITE?
                JP      NZ,SYS3
                LD      A,34            ;USE CP/M RANDOM WRITE
                LD      (RTYP),A
```

```
        JP      SYS5
SYS3:   CP      22H             ;OPENR?
        JP      NZ,SYS4
        LD      A,15            ;CP/M OPEN FILE
        LD      (RTYP),A
        JP      SYS5
SYS4:   CP      23H             ;OPENW?
        JP      NZ,SYS5
        LD      A,19            ;CP/M FILE DELETE
        LD      (RTYP),A
SYS5:   LD      A,(HL)          ;GET SCALL PARAM AGAIN
        AND     A               ;CLEAR CY
        RLA                     ;A=A*2
        INC     HL              ;INCR TO USER'S RET ADDR
        LD      (RETADR),HL     ;SAVE IT
        LD      HL,STABL        ;POINT TO SCALL ADDR TABLE
        CALL    $DADA.          ;INDEX TO PROPER SCALL ADDR
        CALL    $HLIHL          ;GET HANDLER ADDRESS
        POP     AF
        JP      (HL)            ;GO DO IT
;
SRET    EQU     $               ;RETURN FROM SCALL HANDLERS
;       ASSUMES NOTHING IS ON THE STACK ON RETURN!
        PUSH    HL
        LD      HL,(RETADR)     ;GET USER'S RET ADDRESS
        EX      (SP),HL         ;PUT ON STACK & GET BACK HL
        RET                     ;GIT BACK THERE
;
;
;       TABLE OF SCALLS AND HANDLER ADDRESSES
;
STABL   EQU     $
        DW      0               ;.EXIT (JP(0) FOR THIS)
        DW      .SCIN
        DW      .SCOUT
        DW      .PRINT
        DW      .READ
        DW      .WRITE
        DW      .CONSL
        DW      .CLRCO
        DW      .SERR
        DW      .LOADO
;       ENSURE THAT .LINK SCALL LANDS AT #20H
        REPT    22
        DW      .SERR
        ENDM
```

```
        LD      (CLEFT),A       ;UPDATE #CHARS LEFT
        LD      A,(HL)          ;GET A CHAR
        POP     BC
        POP     DE
        POP     HL
        JP      SRET
;
;
GLINE   EQU     $               ;GET A LINE FM CONSOLE
;       ZERO CONSOLE BUFFER
        LD      HL,CLIN
        LD      B,CLEN
        XOR     A
ZR01:   LD      (HL),A          ;ZERO OUT LINE BUFFER
        INC     HL
        DEC     B
        JP      NZ,ZR01
        LD      DE,CBUF         ;POINT TO CONSOLE LINE BUFFER
        LD      C,10            ;READ BUFFERED LINE
        CALL    BDOS
        LD      E,CR            ;ECHO CR,LF TO USER
        LD      C,2
        CALL    BDOS
        LD      E,LF
        LD      C,2
        CALL    BDOS
;SEARCH FOR A CR
        LD      HL,CLIN
        LD      B,CLEN
        XOR     A
SCR1:   CP      (HL)
        JP      Z,SCR2
        INC     HL
        DEC     B
        JP      NZ,SCR1
        JP      SCR3
SCR2:   LD      (HL),NL         ;REPLACE W/ HDOS NL
SCR3:   LD      DE,CLIN
        EX      DE,HL
        CALL    $CHL
        ADD     HL,DE           ;(HL)=POS(NL)
        LD      A,L
        INC     A
        LD      (CLEFT),A       ;UPDATE LINE PARAMS
        LD      (POSNL),A
        RET
```

```
        DW      .LINK
        DW      .CTLC
        DW      .OPENR
        DW      .OPENW
        DW      .OPENU
        DW      .SERR
        DW      .CLOSE
        DW      .POSIT
        DW      .DELETE
        DW      .RENAME
        DW      .SETTOP
        DW      .DECODE
        DW      .NAME
        DW      .CLEAR
        DW      .SERR
        DW      .ERROR
        DW      .CHFLG
RETADR: DW      0
;
;
;       ALL THESE ARE CONSIDERED ERRORS:
;
        PUSH    DE              ;SAVE REGISTERS
        PUSH    BC
        LD      DE,EMSG
        LD      C,9             ;PRINT ERROR MSGS
        CALL    BDOS
        POP     BC              ;RESTORE REGISTERS
        POP     DE
        POP     HL
        JP      SRET
;
EMSG:   DB      CR,LF,'Unimplemented HDOS System Call',CR,LF,'$'
;
        PUSH    DE
        PUSH    BC
        LD      A,(CLEFT)
        AND     A               ;ANY LEFT?
        CALL    Z,GLINE         ;NOPE.
        LD      E,A
        LD      A,(POSNL)
        SUB     E               ;(A)=POSNL-CLEFT
        LD      HL,CLIN         ;POINT TO START OF LINE
        CALL    $DADA.          ;ADD CHARACTER OFFSET
        LD      A,(CLEFT)
        DEC     A
;
;
CBUF:   DB      CLEN            ;CONSOLE LINE BUFFER
CLEFT:  DB      0
CLIN:   DS      80
CLEN    EQU     $-CLIN
POSNL:  DS      1
;
        PUSH    DE
        PUSH    BC
        PUSH    AF
        CP      0AH             ;HDOS NEWLINE?
        JP      NZ,SC01
        LD      E,A             ;OUTPUT A LF
        LD      C,2
        CALL    BDOS
        LD      A,0DH           ;OUTPUT A CR, TOO
SC01:   LD      E,A
        LD      C,2
        CALL    BDOS            ;OUTPUT THE CHAR
        POP     AF
        POP     BC
        POP     DE
        POP     HL
        JP      SRET
;
;
        POP     HL
        CALL    $TPTX.
        JP      SRET
;
;
        POP     HL
        PUSH    HL
        CALL    FFCB            ;FIND FCB OF "CHANNEL" (A)
        PUSH    DE              ;SAVE FCB ADDR
        CALL    FNFCB           ;FORMAT NAME TO FCB
        POP     HL              ;GET FCB ADDR BACK
        PUSH    HL
        LD      A,20H
        CALL    $DADA.
        XOR     A
        LD      (HL),A          ;ZERO (NR)
        INC     HL
        LD      (HL),A          ;ZERO RANDOM RECORD #
        INC     HL
```

```
        LD      (HL),A
        INC     HL
        LD      (HL),A
        POP     HL
        PUSH    HL
        LD      A,0CH
        CALL    $DADA.
        LD      (HL),0
        LD      A,(RTYP)
        CP      19              ;OPEN FOR WRITE?
        JP      NZ,OPEN1
        POP     DE              ;GET FCB ADDR
        PUSH    DE
        LD      C,A
        CALL    BDOS            ;DELETE FILE
        LD      C,22            ;CREATE
        POP     DE
        PUSH    DE
        CALL    BDOS
        INC     A               ;ERROR?
        AND     A
        JP      NZ,OPEN1        ;NO, OK
        POP     DE
        SCF
        LD      A,15
        JP      OPEN2
OPEN1:  LD      C,15            ;OPEN FILE
        POP     DE
        CALL    BDOS
        INC     A               ;ERROR?
        AND     A
        JP      NZ,OPEN2
        SCF
        LD      A,12
OPEN2:  POP     HL
        JP      SRET
;
;
FFCB    EQU     $               ;FIND FCB OF "CHANNEL" (A)
        INC     A
        EX      DE,HL           ;SAVE HL
        LD      HL,FCBL         ;FCB LIST START
        AND     A
        RLA                     ;A=A*2
        CALL    $DADA.
        CALL    $HLIHL          ;GET FCB ADDRESS
```

```
        LD      (DE),A
        INC     DE
        DEC     B
        JP      NZ,FNDON
EXTNT0: LD      B,3             ;NOW MOVE THE EXTENSION
        LD      A,(HL)
        CALL    CAPIT
        CP      '.'
        JP      NZ,EXTDON
EXTENT: INC     HL
        LD      A,(HL)
        CALL    CAPIT
        CP      30H
        JP      C,EXTDON
        LD      (DE),A
        INC     DE
        DEC     B
        JP      NZ,EXTENT
        JP      ALDON
EXTDON: LD      A,' '
        LD      (DE),A
        INC     DE
        DEC     B
        JP      NZ,EXTDON
ALDON:  LD      B,4
ALDON1: XOR     A               ;NULL OUT NEXT 4 BYTES
        LD      (DE),A
        INC     DE
        DEC     B
        JP      NZ,ALDON1
        RET
;
;
CAPIT   EQU     $               ;ASCII CAPITALIZATION ROUTINE
        CP      'a'
        RET     C
        CP      'z'+1
        RET     NC
        SUB     20H             ;WE NEED TO CAP. SO DO IT
        RET
;
;
;       DO A DOUBLE CP/M READ OR WRITE FOR EACH
;       HDOS RECORD
        LD      (ASAV),A        ;SAVE CHANNEL #
        PUSH    BC
```

```
        EX      DE,HL           ;GET HL BACK, FCB ADDR TO DE
        RET
;
;
FCBL:   DW      FCB1            ;FCB LIST
        DW      FCB2
        DW      FCB3
        DW      FCB4
        DW      FCB5
        DW      FCB6
        DW      FCB7
        DW      FCB8
;
;
FNFCB   EQU     $               ;FORMAT NAME TO FCB
        INC     HL
        LD      A,(HL)          ;TEST FOR DISK DESIGNATOR
        CP      ':'
        DEC     HL
        LD      A,0             ;DON'T CHANGE THIS!
        JP      NZ,DSKDES
        LD      A,(HL)
        CALL    CAPIT           ;CAPITALIZE
        INC     HL              ;SKIP COLON
        INC     HL
        SUB     40H             ;CONV TO DRIVE NO.
DSKDES: LD      (DE),A
        INC     DE
        LD      B,8
        DEC     HL              ;BACK UP POINTER
FNLOOP: INC     HL              ;MOVE THE FILENAME
        LD      A,(HL)
        CALL    CAPIT
        AND     A
        JP      Z,FNDON
        CP      '.'
        JP      Z,FNDON
        CP      21H
        JP      C,FNDON
        LD      (DE),A
        INC     DE
        DEC     B
        JP      NZ,FNLOOP
        INC     HL
        JP      EXTNT0
FNDON:  LD      A,' '
```

```
RLOOP:  LD      A,B
        AND     A               ;SEE IF (B)=0
        JP      Z,RDONE
        PUSH    DE              ;SAVE DMA ADDR
        LD      C,26            ;SET DMA ADDR
        CALL    BDOS
        LD      A,(ASAV)        ;GET CHAN #
        CALL    FFCB            ;FIND FCB ADDR
        LD      A,(RTYP)        ;33=RAND READ,34=RAND WRITE
        LD      C,A             ;CPM READ/WRITE
        CALL    BDOS
        POP     DE              ;GET DMA ADDR BACK
        CALL    RERR
        JP      C,RDONE
        LD      HL,80H
        ADD     HL,DE
        EX      DE,HL           ;DE HAS NEW DMA ADDR
        PUSH    DE
        LD      C,26            ;SET DMA ADDR
        CALL    BDOS
        LD      A,(ASAV)
        CALL    FFCB
        PUSH    DE              ;SAVE FCB ADDR
        EX      DE,HL           ;HL NOW HAS FCB ADDR
        LD      A,21H
        CALL    $DADA.          ;POINT TO RAND REC #
        PUSH    HL
        CALL    $HLIHL          ;GET CONTENTS
        INC     HL              ;POINT TO NEXT REC
        POP     DE              ;GET RAND REC ADDR
        EX      DE,HL           ;HL HAS ADDR, DE HAS REC #
        LD      (HL),E          ;SAVE NEW RAND REC
        INC     HL
        LD      (HL),D
        INC     HL
        LD      (HL),0
READ01: LD      A,(RTYP)
        LD      C,A
        POP     DE              ;GET FCB ADDR BACK
        CALL    BDOS            ;DO 2ND READ/WRITE
        CALL    RERR
        JP      NC,READ02
        POP     DE
        JP      RDONE
READ02: LD      A,(ASAV)        ;SET UP FOR NEXT RREC
        CALL    FFCB
```

```
        EX      DE,HL
        LD      A,21H
        CALL    $DADA.
        PUSH    HL
        CALL    $HLIHL
        INC     HL
        POP     DE
        EX      DE,HL
        LD      (HL),E
        INC     HL
        LD      (HL),D
        INC     HL
        LD      (HL),0
READ03: POP     DE              ;GET DMA ADDR BACK
        LD      HL,80H
        ADD     HL,DE
        EX      DE,HL           ;DE HAS NEW DMA ADDR
        POP     BC
        DEC     B
        PUSH    BC
        JP      RLOOP
RDONE:  POP     BC
        POP     HL
        JP      SRET
;
RERR    EQU     $
        AND     A               ;ERROR?
        RET     Z               ;NO, OK
        LD      A,(RTYP)
        CP      34              ;WRITE?
        JP      NZ,RERR1
        LD      A,2
        JP      REDN
RERR1:  LD      A,1             ;IT'S A READ ERROR
REDN:   SCF
        RET
;
RTYP:   DB      0
ASAV:   DB      0
FCB1:   DS      36              ;A SET OF FCB'S
FCB2:   DS      36
FCB3:   DS      36
FCB4:   DS      36
FCB5:   DS      36
FCB6:   DS      36
FC7     DS      36
```

```
        JP      NZ,ERR2
        LD      DE,MSG2
        JP      ERREND
ERR2:   CP      12
        JP      NZ,ERR3
        LD      DE,MSG3
        JP      ERREND
ERR3:   CP      15
        JP      NZ,ERR4
        LD      DE,MSG4
        JP      ERREND
ERR4:   LD      DE,WMSG
ERREND: LD      C,9
        CALL    BDOS
        POP     HL
        AND     A
        JP      SRET
;
;
MSG1:   DB      'End of file',CR,LF,'$'
MSG2:   DB      'Out of disk space',CR,LF,'$'
MSG3:   DB      'File not found',CR,LF,'$'
MSG4:   DB      'No directory space',CR,LF,'$'
WMSG:   DB      'Gross error: Contact your local software guru',CR,LF,'$'
;
;       H17 ROM ROUTINES
;
        DS      183AH-$
$DADA   EQU     $
        RET
;
$DADA.  EQU     $
        RET
;
$DU66   EQU     $
        RET
;
$HLIHL  EQU     $
        RET
;
        DS      1894H-$
$CHL    EQU     $
        RET
;
        DS      18AAH-$
$MOVE   EQU     $
```

```
FCBs:   DS      36
;
;;
        CALL    FFCB            ;POINT TO PROPER FCB
        AND     A               ;CLEAR CY
        LD      A,C             ;GET LO BYTE REC NR
        RLA
        LD      L,A
        LD      A,B             ;HI BYTE
        RLA
        LD      H,A             ;(HL)=(BC)*2
        EX      DE,HL
        LD      A,21H
        CALL    $DADA.          ;POINT TO RAND REC NR
        LD      (HL),E
        INC     HL
        LD      (HL),D
        INC     HL
        LD      (HL),0
        AND     A               ;CLEAR CY
        POP     HL
        JP      SRET
;
;;
        CALL    FFCB
        LD      C,16
        CALL    BDOS
        INC     A               ;ERROR?
        AND     A
        JP      NZ,CLOS1        ;NO, OK
        SCF
        LD      A,12
CLOS1:  POP     HL
        JP      SRET
;;;
        AND     A               ;THESE ARE IGNORED
        POP     HL
        JP      SRET
;;;
        CP      1               ;EOF?
        JP      NZ,ERR1
        LD      DE,MSG1
        JP      ERREND
ERR1:   CP      2               ;OUT OF DISK SPACE?
        RET
;
        DS      18DFH-$
$MU66   EQU     $
        RET
;
        DS      1931H-$
$TJMP   EQU     $
        RET
;
$TBRA   EQU     $
        RET
;
        DS      195EH-$
$TPTX   EQU     $
        RET
;
        DS      2280H-$         ;MAKE SURE PROGRAM CODE STAYS
                                ;WHERE IT USED TO BE UNDER HDOS
STACK   EQU     $               ;STACK GROWS DOWNWARD FROM HERE
;
ADVST   EQU     $
        END     START
```

*

Orion brings you a new choice in general ledger software . . .

The Orion general ledger system is designed to do general ledger accounting for the professional person or small business. It may also be used by an accounting firm for client accounting, with multiple sets of books on the same disks. The system has many outstanding features designed to make its use as easy as possible and to provide reports to assist in making wise management decisions. / A true double-entry accounting system is maintained. / Familiar debit-credit terminology is used for making journal entries. / Reports provide both current period and year-to-date information. / A true audit trail is maintained. Each journal entry is automatically assigned a sequence number; this sequence number is then printed in the ledger. Each transaction also has a reference of up to 10 characters and a free-form description of up to 26 characters. / Use of the system may be begun at any time. All that is required is a current trial balance. / The system is 100% menu driven. It is not necessary to remember complicated instructions for running the programs. / Amounts up to $999,999,999.99 are handled by the system. / The system lets you specify your own headings, subtotals, and totals for reports. You specify titles for the financial statements, the length of your reporting period, and other items unique to your company. / Account number ranges for assets, liabilities, capital, revenue and expenses are specified by you. Account numbers may be up to six digits plus two decimal places in length. The chart of accounts may contain up to 250 lines including headings, subtotals, and totals. / Each individual transaction is checked for validity as it is being entered: (1) uneven debits and credits are not allowed, (2) entries to nonexistent accounts are not allowed. New accounts may be opened during journalizing, if necessary. / Each journal entry may be edited or deleted prior to posting. Posting is done automatically and instantaneously at the end of each journal entry. / Account balance inquiries may be made on the terminal. Account balances are current as of the last journal entry./ Well-planned reports are automatically prepared: a. Chart of Accounts/ Trial Balance/Worksheet. b. Journal in chronological order. c. Ledger showing beginning balance, details of each transaction, and balance after each transaction. d. Income Statement (may be renamed by user) showing amounts for current period and for year-to-date. Each amount is also expressed as a percent of revenue. e. Balance Sheet (may be renamed by user) showing beginning of current period, change and end of period balances. Each balance is also expressed as a percent of assets. / Closing entries are made automatically. / Excellent documentation guides you through the entire accounting cycle. / Menus and conversational question and answer prompts mean minimal training for your accounting personnel.

With General Ledger's powerful combination of features, you get professional results with a minimum of effort.

Because of its built-in flexibility, General Ledger adapts to your accounting needs.

If you know basic accounting, you can use General Ledger.

With its low cost, General Ledger is good for your bottom line in more ways than one.

ORDER FROM: Clark Systems Corporation
P.O. Box 490156, Atlanta, GA 30349

Please send the following:

- ☐ Stock No. 145.1 **GENERAL LEDGER** $179.95
  Requires Heath/Zenith Z89/90 or Z100, CP/M, 64K
- ☐ Manual only for General Ledger (may be applied toward purchase of system). $ 15.00
- ☐ Trial Pack (Includes manual and program disk. Everything works, but the chart of accounts is limited to 20 lines and the transaction file size is very limited. May be applied toward purchase of system.) $ 25.00

Check disk type: ☐ Z100 ☐ Hard Z89/90 ☐ Soft Z89/90

Name _____
Street Address _____
City, State, Zip _____

Another Quality Product Developed by

ORION MANAGEMENT SYSTEMS

Orion Mgt. Systems products are distributed by Clark Systems Corporation.

Easy Livin'
(A Review of SUPER SYSMOD2)

Pat Swayne
Software Engineer

SUPER SYSMOD2 by SoftShop (Jim Teixeira) is a program for HDOS 2.0 that produces a modified system command processor (SYSCMD), and makes other system modifications. The system command processor is the part of HDOS that processes things you type while you are "in the system", that is, when HDOS prompts you with ">" for input. SUPER SYSMOD2 adds several new commands to the system. Some of these are abbreviated versions of existing commands. For example, you can enter C for CAT, and M1 for MOUNT SY1:. Other new commands are completely new, such as FREE to display the amount of free space on a disk. Among the new commands is a File Manager command (FMAN) with several subcommands. It allows you to selectively copy files, delete the source file after copying, reset disks, change file flags, and other functions.

SUPER SYSMOD2 also modifies the PIP program, and handles it differently. In standard HDOS, whenever you use a system command that involves files (such as CAT, COPY, or DELETE), the system command processor loads PIP into memory from the disk, and PIP performs the task requested. Then SYSCMD is loaded back into memory to prompt you for the next command. With SUPER SYSMOD2, SYSCMD is addressed so that it can reside in memory along with PIP, and both are loaded when you boot or warm boot (return to HDOS from a program). File commands are executed immediately, and when they are finished, the HDOS prompt returns immediately. The only exception is when you use PIP explicitly as a command. In this case, PIP is loaded from the disk as before, and since most of the patches to it are dynamic (made by SYSCMD to the memory image of PIP), it functions nearly normally. There is a permanent patch made to PIP that causes it to list files as they are copied or deleted (similar to the patch in REMark #27).

Installing SUPER SYSMOD2

SUPER SYSMOD2 is actually a file called SUPERSM2.ABS. When you copy it to a system disk and RUN it, it makes the changes to the system. When it starts, you have the option of specifying a custom prompt, to replace the normal ">". It then deposits onto the disk a new SYSCMD.SYS and a new SYSHELP.DOC file so that the HELP command will display the new features. It also patches PIP, and when it is finished with these tasks, it returns you to HDOS.

When it is finished, you will have a SYSCMD.SYS file that is 24 sectors big instead of 12 sectors, and a larger SYSHELP.DOC file, but PIP will actually be one sector smaller. That is because SUPER SYSMOD2 removes the Patch History Table from PIP (see "PATCH Mysteries Revealed" in REMark #28). Once a disk is modified, SYSGEN can be used to make more modified disks.

Using SUPER SYSMOD2

As I mentioned before, SUPER SYSMOD2 adds abbreviated versions of the system commands. Not only are the commands themselves abbreviated, but the drive designations are also abbreviated. For example, to see the files on SY1:, you can type the usual CAT SY1: (or DIR SY1:), or you can type C1. To access the alternate (DK:) drives, you just put a slash (/) before the command, so that to mount DK2: you could type /M2. A second alternate device (DY:) is also supported with the backslash (\). If a command access unit 0 of the device, the 0 may be specified or left off with the abbreviated command. Even the new commands have abbreviated versions. You can, for example, abbreviate FREE SY1: with FR1, and FMAN DK2: with /FM2. A few of the new commands are available in abbreviated form only, such as PC, which sends a disk directory to your LP: device. By the way, all disk directories produced by CAT, DIR, C, or PC are alphabetized.

Here is a partial list of SUPER SYSMOD commands.

Rn — Reset unit n. Stand-alone must be set to use this command.

TY FNAME — Type a file (replaces TYPE).

CO — Copy a file (replaces COPY). New ways of specifying arguments to this command have been added. For example, you can say CO SY1:TEST.ABS=SY2: instead of CO SY1:*.*=SY2:TEST.ABS. When you copy a file with CO, COPY, or any of the variations, the date on the new file is the same as the date of the old file, and not necessarily the current date. There is also a /D switch, which causes only files having the current date to be copied.

COR — Copy a file regardless of bad sectors. This command copies a file one sector at a time, and if any sector is bad, "garbage" is put in its place and the copying continues.

DOCOM FNAME=COMMAND1;COMMAND2;...COMMANDn. — This command creates a submit-type file where COMMAND1, etc., are normal HDOS commands that would normally be entered individually, such as CAT SY1: (or C1). The DOCOM command can also be used to examine an existing command file to see the commands it contains.

PRINT DVn:FNAME (or PRn FNAME) — This replaces COPY LP:=FNAME.

GO FNAME — Search all units of the specified device for FNAME and then run it. For example, /GO BASIC will search all of the DK: units for BASIC.ABS and run it if found.

LABn — Change or examine the label on a disk.

FMAN DVn: (or FMn) — Run the File Manager on the selected drive.

The files on the disk are listed one at a time and you can move forward or backward through the list and perform any of the following subcommands on the current file shown.

DEL — Delete the file.

REN NEWNAME — Rename the file to NEWNAME.

CO — Copy the file, with arguments determining the destination. For example, CO SY1: copies the file to SY1:.

CS — Copy the file to the same device specified in the previous copy command.

COD — Copy and delete the original if the copy is successful.

CSD — Copy and delete using the previously specified device.

Rn — Reset unit n.

F (NEWFLAGS) — Change the flags (including L) on the file.

V — View the first 22 lines of the file.

There are also commands to move forward and backward in the list of files, and to exit the File Manager.

Scroll Mode — If you type a period before any command, it puts SUPER SYSMOD2 in the scroll mode. This causes scrolling to stop when a screen full of information has been displayed. You can then move forward a line at a time by pressing the SCROLL key, or move forward a page at a time by pressing SHIFT and SCROLL. Scroll Mode stops when you return to the HDOS prompt.

Evaluation

SUPER SYSMOD2 comes with 8 pages (4 sheets printed both sides) of documentation, which is reasonably well written and covers all of the commands as well as installation. It assumes at least a basic knowledge of the use of HDOS.

Once installed, SUPER SYSMOD2 performs all functions well. File operations are faster than with standard HDOS due to PIP being resident in memory, but warm boots are a bit slower. This is because both the larger SYSCMD.SYS and PIP must be loaded into memory.

I found myself wishing that I could "turn off" two of SUPER SYSMOD2's features. Those are the alphabetized directory, and the preserving of a file's original date when it is copied. Sometimes I like to see the files on a disk in the order in which I placed them rather than in alphabetical order, and to have the current date on a copy rather than the source file's date. There is one way to "turn off" those features, and that is by running PIP as a command. For example, to see an unalphabetized directory on SY1:, you can type PIP SY1:/L, but it wipes out the speed advantages since PIP must be loaded from the disk, and the system warm boots when the operation is finished.

I was able to test the COR command in a real situation when a friend brought in an Autoscribe disk containing his only copies of important files that was messed up (when will people learn to back up their files?). I was able to mount the disk using FAKEMNT (from 885-8004), but I still could not copy some of the files using the normal COPY command. So I tried COR, and was able to get most of them.

In all, SUPER SYSMOD2 is a useful extension to HDOS that can make your work easier, with only a few minor disadvantages. Jim Teixeira also has a regular SYSMOD2 for HDOS 2.0 and a SYSMOD for HDOS 1.6. These also provide abbreviated versions of the commands and faster file operation with PIP resident in memory, but they do not have the more sophisticated features of SUPER SYSMOD, such as the File Manager.

SUPER SYSMOD2 is available from SoftShop, 35 Shadow Oak Dr., Sudbury, MA 01776, phone (617) 443-9693 (Jim Teixeira). It sells for $29.95, shipping included. Massachusetts residents add 5% sales tax.

Vectored from 7

Keyboard Polling Misprint

Dear Walt:

You may wish to point out two misprints in the article "Keyboard Pollinf from BASIC": On pages 30 and 35, lines 530 are A$= , but should be A$="" (a null)!

Roy Reichert
29 Blazier Rd.
Warren, NJ 07060

4MHz Errors...

Dear Pat,

Installed the 4MHz Mod for the H89 — Works Great!

There were two errors in the article that other HUGgies should know about.

1. The ground connections should be U1 pin 7 to U2 pin 8 and U3 pin 7 (not U1 pin 8).

2. The new MTR90 monitor (ROM) requires the 444-83 to be installed in U516 not U550.

An article on the MTR90 may also be informative.

Regards,

Jim Smail
2 Ginger Ln., RT 2
Johnson City, IN 37601

BUGGIN' HUG Error

There is an error in the Buggin' HUG column, December 82, Issue 35. In the letter from Alan Swayze on page 7 in the listing under DUPLEX1, the CALL .DLV should be CALL .DLY.

Carefull Poke

Dear Fellow HUGgers:

I am not on the Source or Compuserve, so I have to send you hard copy.

RE: REMark Issue 33, p. 7 — 2nd Item "How do I get my MX-80 printer to do graphics?"

Thanks for letting me know at what address "127" is with Ext. Benton Harbor BASIC! You saved me a lot of time looking for it. I figured it was in there and planned an attempted search for it. So, I "poked" 255 at 12121 and can report that the MX-80 graphics do respond.

But a word of warning!...It may be better to POKE 255 at 12121 only in a subroutine restricted to PRINT #1 (LP: FILE #1) program statements to print MX-80 graphics and to POKE 127 back in following the CLOSE #1 statements. I noticed that certain wierd things happen to BASIC when 255 is left at 12121. For instance—

Try this:

```
*POKE 12121,255

10 OPEN "LP:" FOR WRITE AS FILE #1
20 A$=CHR$(172)
30 B$=CHR$(190)
40 FOR X=1 TO 72
50 PRINT #1,A$;
60 NEXT X
70 FOR X=1 TO 72
80 PRINT #1,B$
90 NEXT X
100 CLOSE #1
110 END
```

This "bombs" my BASIC every time. When you *RUN, you get the "Illegal Character" error message at line 30. When you *LIST, two wierd line numbers appear at the top - "2560" and line 10 becomes "8860". Hit * RUN again and all heck breaks loose between the disc drive and the terminal! Since my system is H8+H19+H17, I have to "RST/0", reboot and reload. If I avoid using A$=CHR$() and include POKE 12121,255 as part of the program along with POKE 12121,127 after the CLOSE #1 statement, BASIC runs normally.

As the HDOS documentation states: "BE CAREFUL WHEN YOU GO POKIN' AROUND!"

Regards,

Paul B. Boivin, Jr.
242 Old River Road, RR4
Lincoln, RI 02865

Correction

A correction to the Local HUG Club listings in the January 1983 issue of REMark: CHUG (Capital Heath Users Group), President and contact person is Mike Cogswell. The address remains the same, but the phone number is (703) 620-9176. They also have a continuous BB that can be reached at (703) 360-3812. All other information contained in that listing remains the same.

Please note that the HUGNJ (HUG of New Jersey) phone numbers in the January 1983 REMark are incorrect. They should be: (201) 791-6935 for contact; (201) 791-6936 for the bulletin board.

Please keep those letters and cards coming in, your comments and suggestions are helpful and most welcome.

Changing your address? Be sure and let us know since the software catalog and REMark are mailed bulk rate and it is not forwarded or returned.

CUT ALONG THIS LINE

HUG MEMBERSHIP RENEWAL FORM

When was the last time you renewed?

Check your ID card for your expiration date.

IS THE INFORMATION ON THE REVERSE SIDE CORRECT? IF NOT FILL IN BELOW.

Name _____

Address _____

City-State _____

Zip _____

REMEMBER — ENCLOSE CHECK OR MONEY ORDER

CHECK THE APPROPRIATE BOX AND RETURN TO HUG

		NEW MEMBERSHIP FEE IS:
RENEWAL RATES		
US DOMESTIC	$15 ☐	$18 ☐
CANADA	$17 ☐ US FUNDS	$20 ☐
INTERNAT'L*	$22 ☐ US FUNDS	$28 ☐

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

Superior Support

BULK RATE
U.S. Postage
PAID
Heath Users' Group

Heath Users' Group
Hilltop Road
St. Joseph, MI 49085

POSTMASTER: If undeliverable, please do not return.

585-2037

Volume 4, Issue 2