THE ADDRESS BUS

THE EIGHT-BIT R/W -- Letters

Favorite Routines. [From Bob Phillips, Charlotte, MI] "In reply to your experiment with self-adhesive labels on the last mailing. I have been doing mailings with self-adhesive labels for several years and have NOT had any labels come off in the mail. Some of the mailings are as small as 60-70 pieces and others as large as 1,000+. Please continue to use labels so you can spend your time putting **The Staunch 8/89'er** together, **not** hand-feeding envelopes.

"Hank Lotz's "This 'n' That" in issue #22/23 was very interesting. I always found the **REMark** column "My Favorite Subroutines" to be very interesting. Listed below is one of my favorite subroutines. [Bob's MBASIC listings have been edited to fit column width. -Ed.]

```
10 ' Date subroutine
20 ' Inputs date from HDOS (22-Jun-91) and outputs @
   DATE$ (June 22, 1991)
30 ' by Robert Phillips, Charlotte, Michigan.
40 ' For MBASIC 4.82 and HDOS 3.0x or HDOS 2.0
50 '
60 GOSUB 8000
70 '
80 ' Other parts of program
90 '
100 END
8000 FOR I=8383 TO 8391:DATE$=DATE$+CHR$(PEEK(I)): @
     NEXT
8010 D$=LEFT$(DATE$,2):Y$=RIGHT$(DATE$,2): @
     M$=MID$(DATE$,4,3)
8020 IF M$="Jan" THEN M$=M$+"uary"
8030 IF M$="Feb" THEN M$=M$+"ruary"
8040 IF M$="Mar" THEN M$=M$+"ch"
8050 IF M$="Apr" THEN M$=M$+"il"
8060 IF M$="Jun" THEN M$=M$+"e"
8070 IF M$="Jul" THEN M$=M$+"y"
8080 IF M$="Aug" THEN M$=M$+"ust"
8090 IF M$="Sep" THEN M$=M$+"tember"
8100 IF M$="Oct" THEN M$=M$+"ober"
8110 IF M$="Nov" THEN M$=M$+"ember"
8120 IF M$="Dec" THEN M$=M$+"ember"
8130 DATE$=M$+" "+D$+", 19"+Y$
8140 PRINT "The date is ";DATE$:PRINT
8150 RETURN
```

"Last fall I was doing some reports that I wanted printed on both [sides] of the paper, to read like a book. HUG has [a program] for MSDOS but not for HDOS. So I took the challenge of writing a program to print a file like a book. The result is a MBASIC program, listed below, called BOTHSIDE.BAS.

```
10 ' bothside.bas .... print an ASCII file on both
15 ' sides of paper.
20 ' by Robert Phillips .... Charlotte, Michigan
25 ' .... last update March 1991
30 ' for H/Z 89 .... HDOS 3.0x .... MBASIC 4.82
40 '
50 ' set variables
60 '
70 CLEAR 4000: DEFINT A-Z
80 ESC$ = CHR$(27)
90 CLS$ = ESC$+"E"
100 CURSROFF$ = ESC$+"x5"
110 POWERUPCNFIG$ = ESC$+"z"
120 BELL$ = CHR$(7)
130 FOMFEED$ = CHR$(12)
140 LP$="UT0:":' [printer device name - Ed.]
150 TAB = 0
160 FIRSTLNE = 1
170 LASTLNE = 60
180 DIM LI$(LASTLNE)
190 DFALTFILE$="ss0:conbylaw"
200 '
210 ' get name of file and open files
220 '
230 PRINT CLS$
240 PRINT
250 PRINT "BOTHSIDES -- will print ascii file on";:@
    PRINT " both sides of paper . . ."
260 PRINT
270 PRINT "Will be printing "LASTLNE;:@
    PRINT " lines per page."
280 PRINT
290 PRINT "Please enter . . ."
300 PRINT
310 PRINT "The name of drive and file for input
<";:@
    PRINT DFALTFILE$">....";: INPUT DEFALTFILE$
320 OPEN "I",1, DFALTFILE$
330 OPEN "O",#2, LP$
340 PRINT CURSROFF$
350 PRINT CLS$
360 '
370 ' read and print odd number pages
380 '
390 FOR ODDPAGE = FIRSTLNE TO LASTLNE
400 LINEINPUT #1,RL$:LI$(ODDPAGE)=RL$
410 IF EOF (1) THEN @
    FOR COMPLEPAGE = ODDPAGE + 1 TO LASTLNE:@
    LI$(COMPLEPAGE)="":NEXT:CLOSE #1:SWITCH = 1 :@
    GOTO 430
420 NEXT
430 FOR PRNTPAGE = FIRSTLNE TO LASTLNE
440 PRINT #2, TAB(TAB);LI$(PRNTPAGE)
450 NEXT
460 PRINT #2, FOMFEED$
470 '
480 ' read and throw away even number pages
490 '
500 IF SWITCH = 1 THEN GOTO 590
```

```
510 FOR TRASHPAGE = FIRSTLNE TO LASTLNE
520 IF EOF (1) THEN SWITCH = 1 :GOTO 550
530 LINEINPUT #1,TRASH$
540 NEXT
550 IF SWITCH = 1 THEN GOTO 590  ELSE GOTO 390
560 '
570 ' push all data to printer
580 '
590 CLOSE #2
600 OPEN "O",#2, LP$
610 '
620 ' tell operator to turn paper in printer over
630 '
640 PRINT CLS$
650 PRINT BELL$
660 PRINT
670 PRINT"Please remove paper from printer, turn";:@
    PRINT " over and put back in"
680 PRINT "Press any key when ready ";:@
    ANYKEY$=INPUT$(1)
690 PRINT
700 PRINT CLS$
710 '
720 ' start over from beginning of file (rewind)
730 '
740 SWITCH = 0
750 CLOSE #1
760 OPEN "I",1, DFALTFILE$
770 '
780 ' get and throw away odd number pages
790 '
800 FOR TRASHPAGE = FIRSTLNE TO LASTLNE
810 LINEINPUT #1,TRASH$
820 IF EOF (1) THEN GOTO 960
830 NEXT
840 '
850 ' get and print even number pages
860 '
870 FOR EVENPAGE = FIRSTLNE TO LASTLNE
880 LINEINPUT #1,RL$:LI$(EVENPAGE)=RL$
890 IF EOF (1) THEN @
    FOR COMPLEPAGE = EVENPAGE + 1 TO LASTLNE:@
    LI$(COMPLEPAGE)="":NEXT:SWITCH = 1 :GOTO 910
900 NEXT
910 FOR PRNTPAGE = FIRSTLNE TO LASTLNE
920 PRINT #2, TAB(TAB) LI$(PRNTPAGE)
930 NEXT
940 PRINT #2, FOMFEED$
950 IF SWITCH = 1 THEN GOTO 960 ELSE GOTO 800
960 PRINT #2, FOMFEED$
970 CLOSE
980 '
990 ' ending
1000 '
1010 PRINT POWERUPCNFIG$
1020 PRINT
1030 PRINT
1040 PRINT
1050 PRINT "File printed on both sides of paper"
1060 PRINT "Job complete . . ."
1070 PRINT "Press any key to return to operating";:@
     PRINT " system ";:ANYKEY$=INPUT$(1)
1080 PRINT
1090 PRINT "Good by . . . Have a nice day . . ."
1100 SYSTEM
```

Notes: All printer control is by way of the device driver (print size, lines per inch, etc.). All error handling is by HDOS 3.0x (file not found, etc.). Do NOT use printer control characters in the ASCII file to be printed. Most other printer control characters **will** work properly.

"I hope you and readers of **The Staunch 8/89'er** will be able to use [this program].

"In closing I have one question: How can the HDOS 3.0x clock be used in a MBASIC program? I have looked at all memory locations and have not been able to find the time. Can you or any readers help?" [Thanks for the nifty routines, Bob. Indeed, I would be happy to begin a "favorite routines" column (for any and every language you care to program in) if you readers will contribute material to publish. Further, if you would like **any** of the source code (BASIC or otherwise) that **Staunch** publishes, let me know. I've decided to breakout program code for distribution for those of you who'd rather not key it in. To obtain this material, merely send me a formatted disk with **postage-prepaid** return mailer and I'll transfer it for you at no charge. Just be sure you tell me your disk format and the issue number for the material you want. This issue's source code includes Bob's material above plus Gerry Kabelman's NUMBERS program presented later. ~Ed.]

Ribbons 'N' Paint. [From Hank Lotz, 2024 Sampson St., Pittsburgh, PA 15221] "After reading Bill Lindley's hint on the use of WD-40 to revive ribbons, I'm sorry, but I must remind everyone of the dire warning **against** that practice; see **Staunch** #3, p. 8, 3rd Q/A pair down the column! And that caveat in #3 didn't mention daisywheels, but I don't **know** what WD-40 might do to 'space-age' plastic wheels, either. (I **do** know I'm sure not going to test it out on **my** daisywheels!)...

"Lee [Hart] sent me a card (Jan. 19 postmark) which you should print, as it replies to p. 10 of #20/21 [about the loss of data when saving complicated terminal graphics lines to disk with PAINT]:

> 'Dear Hank, ...On Steve Vagts' PAINT program, I think the problem was that lines with many ESC sequences exceeded 256 bytes per line, so his program didn't store them properly. A worst-case screen dump from the H-19 can exceed 10k, by the way. [**That** would've been hard to guess! -- HL]
>
> 'I wrote an EDIT program for my Write-Hand-Man. It's a 1.5k CP/M program that can accept information from a disk file or **the screen of any program** [his "italics" throughout], can edit this information **including graphics and reverse video**, and can save the edited data on disk, **print it**, or return it to the interrupted program. If you have Write-Hand-Man take a look at EDIT for a generalized screen dump program.'"

[For those of you who don't have easy access to issue #3 that Hank referred to above, a user described how he **accelerated** wear of his printer's printhead (a $100 item) by using WD-40 on his ribbons. Nylon printer ribbons dry out mainly because of the evaporation of a light oil that serves as ink carrier **and** printhead pin lubricant. Is there a reasonable substitute (other than WD-40) that anyone knows of? I suspect a lot of you would like to know! ~Ed.]

Comments on the Last Issue, Etc. [From Lee Hart, 323 W 19th St., Holland, MI 49423, 616/396-5085] "Great to finally get your latest issue. The self-stick label stayed stuck just fine. But June for the Jan-Apr issue? Must have taken a long time to grow the soybeans for the new ink!...

"Page 5: I have the SD Sales 64K RAM S-100 boards in case anyone wants one for his H-8. It actually accepts up to 256K, and has bank switching logic. Brand new, with RAM chips, in original box for $20 with 64K, or $30 with 256K of RAM installed.

"Page 8, Charles Horn: Yes, I'd like to order a 'master' HDOS-CP/M dual-format disk. There have been a couple of times when this would have been useful. If I don't get the disk now, I'll never have one when I need it. Dummy files in the HDOS and CP/M directories to lock out the opposite operating system's space is a good idea. This should make it possible to read or write to the disk with either operating system.

"Page 12: You inadvertently merged two magazine names; **Pieces of Eight** and **Eight Bits and Change.**

"Hey, I have the honor of the first color/graphics article in **Staunch**! Looks correct; I hope people can read it. If they complain, tell them to view it under tungsten light, which accents the red.

"Page 14, Mark Hunt: An 80-track drive can't write double-width tracks (for a 40-track drive) by recording in two passes. First, because a blank 'guard band' is recorded between tracks during writes (it helps erase the old data in case the head isn't precisely centered in the track). Second, disks don't rotate at precisely the same speed; thus a second track won't line up properly with an identical first track. Third, the 80-track drive's two passes should be half a track on each side of center, but normal floppy disk drives can't half-step their heads.

"But there are ways to help 80-track drives write 40-track disks. Be sure your 80-track drive is properly aligned. Start with a bulk-erased disk. Wipe it with a magnet if you're not sure. Copy the desired data onto it with a DUP-type program that formats the disk and writes the data in **one pass.** The purpose is to minimize the background noise left over from previous writes, because the wider 40-track head will pick it all up. I don't know of a Heath CP/M program that does this, but HDOS DUP21 and SDUP37 work.

"Note that Apple and Commodore computers directly control the head stepping motor windings, so they **can** half-step their heads. This is frequently done to produce bizarre unreadable disks as a form of copy protection. (I for one am thankful Heath saved us from this kind of nonsense.)

"That's not to say there aren't things you can do to get more out of your disk drives with hardware/software changes. Most disk drives have extra undocumented tracks at the center. If FORMAT was modified to use them, you'd get an extra track or two of data.

"The H-17 controller will work just fine with soft-sector (1-hole) disks. It would have to be single density, because that's all the hardware supports. By definition, such a disk would have just one huge sector per track (3K bytes long). Software would have to be written to use this capability. Same is true for a 3-1/2" drive; they are electrically interchangeable with 5-1/4" drives.

"Clay Montgomery expressed an interesting idea in the last **SEBHC Journal** [Vol. 5, #10 (May, '91), p. 1]. He proposed building a 'portable' H-89 with modern parts. I think this is a great idea and would be willing to help develop one.

"What I envision is a portable CP/M computer; a smaller, simpler, cheaper, lightweight alternative to the usual DOS machines. Something like the Radio Shack TRS-80 Model 100, but with a 25x80 screen and running CP/M.

"I'd like to know what your readers think. What features are important to you? Small and simple or big and feature laden? El cheapo or cost-is-no-objective? Real disk drives or RAM disks? Let's hear it, guys!" [Thanks for your input, Lee. My reason for dating this year's first issue as I did was to imply to the readers that they will be getting a **full** set of issues this year, even though I'm running behind. As for the two-color graphics for your auto-repeat articles, I was disappointed with the way it turned out. When I do that again (highly likely with the hardware troubleshooting series in the works), I plan to try another combination! -Ed.]

=====

NUMBERS

By Gerry Kabelman
(Reprinted from **REMark** #20 [Sept. '81], pp. 3-6)

"But Uncle Gerry, I don't know how to do that!!", exclaimed Melissa when the division problem appeared on the screen ...

Melissa's complaint is about the program called NUMBERS.BAS and follows on the next couple of pages.

NUMBERS.BAS is an example of a COMPUTER AIDED INSTRUCTION (CAI) program. Many educators, parents and students have requested CAI programs for use with the Heath/Zenith computers. HUG is offering the NUMBERS.BAS program as an example of what a CAI program can do. HUG also has available a complete disk of CAI programs running under HDOS and MBASIC. This disk, HUG part 885-1097 ($20.00), is described in detail in issue #18 of **REMark.**

NUMBERS.BAS is a very good example of a CAI program for addition, substraction, multiplication and division using one inch numbers for the early elementary student.

The program is broken down into four sections and may best be explained by reviewing the listing and trying it.

The sections are:

Description	Line Numbers
1. Startup	10 to 200
2. Letter & Number Setup	1000 to 1550
3. Main Program	2000 to 2560
4. Subroutines	10000 to 16040

The letter and number setup is the most important part of this program. Lines 1000 to 1040 create the parts of the characters, while lines 1100 to 1550 actually create the characters from the parts. Also note the subroutines for inputting and printing the large characters.

This program may not make you an Einstein, but

it will help you feel like you could be one.

[Although written for HDOS, this program by Gerry Kabelman can be modified for CP/M. The changes necessary for that include the following:

1. Delete the CLEAR statement in line 100.
2. Replace the @ (at-sign) **and** trailing carriage return at the end of each physical in the listing with CP/M MBASIC's LINEFEED character as you key or edit. If you request the code on disk from me--see further below--you might find it easier to edit the program source with a text editor to remove the at-signs and RETURNs **before** you load it into MBASIC for final editing and testing. Otherwise, MBASIC will gag when you try to load the program. However, line length will **exceed** 80 characters when you do this, so you'll need Magic Wand, WordStar, TXTPRO, or a similar editor that can handle super-long lines. You **can't** edit with Software Toolworks' PIE because this simple editor doesn't permit the merging of lines!
3. The location being PEEK'd (8219decimal = 201Bhex = 040.033A) in lines 2150, 2200, and 2230 is in the 8K monitor at the bottom of memory under HDOS 2.0 or in the system area under 3.0x. This is a neat "randomizer" for HDOS MBASIC's RND() function under **both** 2.0 and 3.0x; the address PEEK'd in these lines returns a number from 0 to 255 from HDOS's "tic counter." But this PEEK is **invalid** under CP/M. Change lines 2150, 2200, and 2230 to the general format,

```
RANDOMIZE RND : i = INT(RND * j) + 1
```

   where "i" is the variable W, X, or Y as needed and "j" is "10" in line 2150 and "100" in lines 2200 and 2230 in the program listing below. Actually, the last could be at your discretion (within limits) since "j" here defines the **highest** number used in the computation.
4. Change line 2220 to

```
RANDOMIZE RND : FOR I=1 TO W*RND: NEXT I
```

5. Change line 12010 to

```
RANDOMIZE RND : ON INT (RND * 5) +1
GOTO 12020,12030,12040,12050,12060
```

You might also want to play with the time delay in line 11000 if you have a CPU speedup installed.

[If you would like this program, just send me a formatted disk, plus a **postage-prepaid** return mailer, and I'll transfer the ASCII file for you at no charge. The code will be the HDOS version; I'll leave adapting it to you. (Bob Phillips's routines from "The Eight-Bit R/W" will be included with it.) If you run into trouble adapting it, I'll be happy to assist. Improvements to the program are certainly possible and I'd like to hear of any you make. -Ed.]

Listing

[This has been edited to fit column width. -Ed.]

```
10 '       NUMBERS.BAS     Version 07.01.81      :GK:
100  CLEAR 9000: WIDTH 255: DIM L$(130)
110  E$=CHR$(27): EP$=E$+"E": Y$=E$+"Y": F$=E$+"F":@
     G$=E$+"G": P$=E$+"p": Q$=E$+"q"
120  Y5$=E$+"y5"+Q$: X5$=E$+"x5": Q1$=CHR$(34):@
     A1$=CHR$(64): J$=E$+"J"
200  PRINT EP$X5$: LS$="    "
1000 B$=E$+"B": D$=E$+"D": D1$=D$+D$:@
     D$=D1$+D1$+D1$+B$: D1$=D1$+D$:@
   N1$="iiiiii"+D$:N2$="    ii"+D$:N3$="ii    "+D$:@
   N4$="  ii  "+D$:N5$="ii  ii"+D$:N6$="iiii  "+D$:@
   N7$="iii ii"+D$:N8$="ii ii "+D$
1010 M1$="ii iii"+D$: M2$=" iiii "+D$:@
   M3$="   ii "+D$:M4$=" ii   "+D$:M5$="  iiii"+D$:@
   M6$="iiiii "+D$:M7$="  iii "+D$:M8$=" ii ii"+D$:@
   M9$="      "+D$
1020 O1$="ii ii ii"+D1$: O2$="iii  iii"+D1$:@
     O3$="iiiiiiii"+D1$
1030 C$=E$+"C": C$=C$+C$+C$+C$:C$=C$+C$: A$=E$+"A":@
     R$=A$+A$+A$+A$+A$+C$: UP$=A$
1040 B1$=D1$+UP$: B2$="        "+D1$
1100 '                          Setup of Characters
1110 L$(7)="":'                                    Bell
1120 L$(8)=B1$+B2$+B2$+B2$+B2$+B2$+R$:'    Backspace
1130 L$(32)="      ":'                        ( ) SPACE
1140 L$(33)=N1$+N1$+N1$+N1$+M9$+N1$+R$+UP$:'       !
1150 L$(37)=N5$+M3$+N4$+M4$+N5$+R$:'               %
1160 L$(43)=M9$+N4$+N1$+N4$+M9$+R$:'               +
1170 L$(45)=M9$+M9$+N1$+M9$+M9$+R$:'               -
1180 L$(48)=N1$+N5$+N5$+N5$+N1$+R$:'               0
1190 L$(49)=" iii  "+D$+N4$+N4$+N4$+N1$+R$:'       1
1200 L$(50)=N1$+N2$+N1$+N3$+N1$+R$:'               2
1210 L$(51)=N1$+N2$+M5$+N2$+N1$+R$:'               3
1220 L$(52)=N5$+N5$+N1$+N2$+N2$+R$:'               4
1230 L$(53)=N1$+N3$+N1$+N2$+N1$+R$:'               5
1240 L$(54)=N1$+N3$+N1$+N5$+N1$+R$:'               6
1250 L$(55)=N1$+N2$+N2$+N2$+N2$+R$:'               7
1260 L$(56)=N1$+N5$+N1$+N5$+N1$+R$:'               8
1270 L$(57)=N1$+N5$+N1$+N2$+N1$+R$:'               9
1280 L$(63)=N1$+N5$+N2$+M5$+N4$+B$+N4$+R$+UP$+UP$:'?
1290 L$(65)=M2$+N5$+N1$+N5$+N5$+R$:'               A
1300 L$(66)=N1$+N5$+M6$+N5$+N1$+R$:'               B
1310 L$(67)=N1$+N3$+N3$+N3$+N1$+R$:'               C
1320 L$(68)=N1$+M8$+M8$+M8$+N1$+R$:'               D
1330 L$(69)=N1$+N3$+N6$+N3$+N1$+R$:'               E
1340 L$(70)=N1$+N3$+N6$+N3$+N3$+R$:'               F
1350 L$(71)=N1$+N3$+M1$+N5$+N1$+R$:'               G
1360 L$(72)=N5$+N5$+N1$+N5$+N5$+R$:'               H
1370 L$(73)=M2$+N4$+N4$+N4$+M2$+R$:'               I
1380 L$(74)=N2$+N2$+N2$+N5$+N1$+R$:'               J
1390 L$(75)=N5$+N8$+N6$+N8$+N5$+R$:'               K
1400 L$(76)=N3$+N3$+N3$+N3$+N1$+R$:'               L
1410 L$(77)=O2$+O3$+O1$+O1$+O1$+"  "+R$:'          M
1420 L$(78)=N5$+N7$+N1$+M1$+N5$+R$:'               N
1430 L$(79)=L$(48):'                               O
1440 L$(80)=N1$+N5$+N1$+N3$+N3$+R$:'               P
1450 L$(81)=N1$+N5$+N5$+M1$+N7$+R$:'               Q
1460 L$(82)=N1$+N5$+N1$+N8$+N5$+R$:'               R
1470 L$(83)=L$(53):'                               S
1480 L$(84)=N1$+N4$+N4$+N4$+N4$+R$:'               T
1490 L$(85)=N5$+N5$+N5$+N5$+N1$+R$:'               U
1500 L$(86)=N5$+N5$+N5$+M2$+N4$+R$:'               V
1510 L$(87)=O1$+O1$+O1$+O3$+O2$+"  "+R$:'          W
1520 L$(88)=N5$+M2$+N4$+M2$+N5$+R$:'               X
1530 L$(89)=N5$+N5$+M2$+N4$+N4$+R$:'               Y
1540 L$(90)=N1$+N2$+N4$+N3$+N1$+R$:'               Z
1550 L$(127)=L$(8):'                            Delete
2000 '                  Print "HELLO MY NAME IS Z89"
2010 A1=1: PRINT Y$" O";: S$="HELLO": GOSUB 16030:@
     PRINT Y$"& ";: S$="MY NAME IS": GOSUB 16030:@
     PRINT Y$",6";:S$="Z-89":GOSUB 16030:GOSUB 11000
2020 '                  Print "WHAT IS YOURS?"
```

```
2030 PRINT EP$;: S$="WHAT IS": GOSUB 16030:@
     PRINT Y$"& ";: S$="YOURS": GOSUB 16030:@
     PRINT Y5$" "Y$". ";: S$="?": GOSUB 16030
2040 GOSUB 15010
2050 '                    Print "HOW MANY QUESTIONS?"
2060 PRINT EP$;: S$="HOW MANY": GOSUB 16030
2070 A=0: B=0: C=0
2080 PRINT Y$"& ";: S$="QUESTIONS": GOSUB 16030
2090 PRINT Y$"00";: S$="? ": GOSUB 16030: N=1:@
     GOSUB 15010
2100 A=VAL(A1$)
2110 IF A<1 THEN 2060 ELSE 2120
2120 A=INT(A):'           Allow only integers
2130 PRINT EP$:
2140 '           Start selecting numbers to be used.
2150 W=INT(RND(1)*PEEK(8219)):'    Random Function @
                                   1=Addition      @
                                   2=Subtraction   @
                                   3=Multiplication@
                                   4=Division
2160 IF W>4 THEN 2150
2170 J=10: J1=10
2180 IF W=2 THEN J=20: GOTO 2200
2190 IF W=4 THEN J=100: GOTO 2200
2200 X=INT(RND(1)*PEEK(8219))+1:'  1st Random Number
2210 IF X>J THEN 2190
2220 FOR I=1 TO W*(RND(1)): NEXT I
2230 Y=INT(RND(1)*PEEK(8219))+1:'  2nd Random Number
2240 IF Y>J1 THEN 2200
2250 ON W GOTO 2260,2270,2280,2290
2260 Z=X+Y: GOTO 2320:'            Find Sum
2270 IF X>Y THEN Z=X-Y: GOTO 2320: ELSE GOTO 2150:@
     '                             Find subtrahend
2280 Z=X*Y: GOTO 2320:'            Find Multiplicand
2290 IF Y=0 THEN 2150
2300 IF INT(X/Y)=X/Y THEN Z=X/Y ELSE GOTO 2150:@
     '                             Find dividend
2310 '                             Print Numbers
2320 GOSUB 13010
2330 PRINT P$Y$"0?";: N=1: GOSUB 15010: Z1=VAL(A1$)
2340 PRINT Y$"0;"E$"J";
2350 IF LEN(A1$)=3 THEN 2380
2360 Z2=Z1: GOSUB 14010:@
     '         Preprint Answer if only one character
2370 '         If answer is CORRECT
2380 IF Z=Z1 THEN PRINT EP$;: S$="   RIGHT":@
     GOSUB 16030: PRINT Y$"& ";: GOSUB 12010:@
     GOSUB 11000: B=B+1: C=C+1: IF A=B THEN 2420@
     ELSE 2130
2390 '         If answer is INCORRECT
2400 Z2=Z: PRINT Y$"0 "J$;: S$="  WRONG!":@
     GOSUB 16030: GOSUB 11000: PRINT Y$"0 "J$;:@
     GOSUB 14010: B=B+1: IF A=B THEN 2420 ELSE 2130
2410 '                          Print up tally display
2420 PRINT EP$Y$"  ";: S$="END  SCORE": GOSUB 16030
2430 '                C=Score & A=Number of Problems
2440 PRINT Y$"& ";: S$=STR$(C): GOSUB 16000
2450 S$=" RIGHT": GOSUB 16030
2460 PRINT Y$", ";: S$="OUT OF ": GOSUB 16030
2470 S$=STR$(A): GOSUB 16030
2480 '       Figure Percentage Correct * Display it
2490 IF C/A=1 THEN PRINT F$Y$"20";: S$="100 %":@
     GOSUB 16030: GOTO 2510
2500 C1=INT(C/A*100): S$=STR$(C1)+" %":@
     PRINT Y$"2"CHR$(64);: GOSUB 16030
2510 GOSUB 11000
2520 '                          Delay Visual "BYE"
2530 GOSUB 11000
2540 S$="AGAIN? ": PRINT EP$;: GOSUB 16030:@
     PRINT Y5$;: A1$=INPUT$(1): PRINT X5$;
2550 IF A1$="Y" OR A1$="y" THEN 2060
2560 PRINT EP$;: S$="  BYE": GOSUB 16030:@
     PRINT Y$"& "Y5$: END
10000 '                          *** SUBROUTINES ***
11000 FOR S=1 TO 100: NEXT S:'Time Delay
11010 RETURN
12000 '              Select & Print Correct Message
12010 ON INT(RND(1)*5)+1 @
      GOTO 12020,12030,12040,12050,12060
12020 S$="   BRAIN": GOTO 16030
12030 S$="    WOW": GOTO 16030
12040 S$="HAVE MERCY": GOTO 16030
12050 S$="FANTASTIC": GOTO 16030
12060 S$=" EINSTEIN": GOTO 16030
13000 '                Print Numbers For Problem
13010 PRINT EP$Y$" ?";: S$=STR$(X): GOSUB 16000:@
      PRINT Y$"'?";: S$=STR$(Y): GOSUB 16000
13020 ON W GOTO 13030,13040,13050,13060
13030 PRINT F$Y$"(5"L$(43);: GOTO 13070
13040 PRINT F$Y$"(5"L$(45);: GOTO 13070
13050 PRINT F$Y$"(5"L$(88);: GOTO 13070
13060 PRINT F$Y$"(5"N4$M9$N1$M9$N4$;
13070 PRINT Y$".0"STRING$(46,"i")G$: RETURN
14000 '              Print Answer In Proper Position
14010 S$=STR$(Z2): PRINT Y$"0?";: GOSUB 16000:@
      GOSUB 11000: GOSUB 11000: RETURN
15000 '
15010 A1$="":'       Input Characters
15020 PRINT Y5$;: A$=INPUT$(1): PRINT X5$:@
      IF A$=CHR$(13) THEN 15090
15030 IF A$=" " THEN PRINT "        ";:@
      A1$=A1$+" ": GOTO 15020
15040 IF (A$=CHR$(127) OR A$=CHR$(8)) AND @
      LEN(A1$)>0 THEN 15100
15050 IF N=1 AND (A$>"/" AND A$<":") THEN @
      PRINT F$L$(ASC(A$))G$;: A1$=A1$+A$: GOTO 15020
15060 IF A$>"`" AND A$<"{" THEN A$=CHR$(ASC(A$)-32)
15070 IF N=1 OR (A$<"A" OR A$>"{") THEN PRINT "";:@
      GOTO 15020
15080 PRINT F$L$(ASC(A$))G$;: A1$=A1$+A$:@
      IF LEN(A1$)=9 THEN 15090 ELSE 15020
15090 N=0: GOTO 16040
15100 IF RIGHT$(A1$,1)="W" OR RIGHT$(A1$,1)="M" @
      THEN PRINT E$"D"E$"D";
15110 PRINT L$(8);: A1$=LEFT$(A1$,LEN(A1$)-1):@
      GOTO 15020
16000 S2$=S$:'           Print LARGE Characters
16010 IF LEN(S$)>2 THEN 16030
16020 IF LEN(S2$)<3 THEN PRINT L$(32)"  ";:@
      S2$=S2$+" ": GOTO 16020
16030 FOR I=1 TO LEN(S$):@
      PRINT F$L$(ASC(MID$(S$,I,1)))G$;:NEXT I
16040 RETURN
```

=====

Square One for Computerphiles
Part 3 -- H-89 Terminal Configuration

by Hank Lotz / 2024 Sampson St. / Pgh, PA 15221

First a Chat. This is the third in a series of attempts to give the abject novice a handle on computers and computer work. Each of these **Square One**s is independent of the others. For example, you don't have to read Part 2 before Part 3. Each installment covers a different area, so you can start anywhere.

I'm kind of surprised to be doing a Part 3, though, because during Part 1 (in #14) I wasn't so sure there'd even be a Part 2. Although I haven't gotten a lot of mail, I'm still convinced there's a need. Many years back, when all I had were questions, I was often hesitant to ask (and display my utter ignorance). But I followed the example of braver specimens who probably also felt trepidation but asked anyway! Another thing I learned is you always have more questions! Personally, I still have tons. But about that mail, please let me know what you're thinking. If you have questions, I won't **promise** I'll have your particular answer, but I'll reply to each letter individually and personally.

Down to Business. Some people seem reluctant to take a few minutes to look at the ways the H-89 (H-19 terminal) can be configured, **and to decide** just what choices are best for them. Today I want to attack that reluctance and encourage you to set up your terminal the way **you** want it.

I'm talking about the choices afforded by the Terminal Logic Board "DIP" switch S402. This switch sets your cursor to be either block or underscore, turns key click on or off, turns wraparound on or off, shifts or unshifts the keypad, and does a few other things. In order to control these multiple functions, switch S402 has eight "sections" (numbered "0" through "7"), each "section" being no more than just a tiny switch. The manual shows which switch section controls which feature, so I'm taking only a couple as examples in this article.

The book says to set all the little subswitches to the "0" position (up), but implies you can reconfigure them to suit your own tastes. However you configure them, the DIP sections are read (and their settings thus enforced) **only at power-up, or at a hard reset.** (For a restatement of that, see the last paragraph under **Hardware Configuration,** below.)

That "0" position gives you, among other things, **key click ON, keypad UNshifted,** and **wraparound OFF** (when wraparound is off, that's called "discard" and we'll talk more about it in a moment).

You can also make the various choices by off-line keyboard commands, without ever touching the switches, or by on-line software (which still doesn't touch the switches), and I'll give an example of that kind of software, for CP/M.

The Shifted Keypad Function. I keep my keypad **shifted** (section 6 down). Why? Because the number keys (0 thru 9) are also on the main keyboard, not just on the keypad, and I don't need **two** sets. The keypad is more useful to me for its screen-editing mode, so why should I have to hold down the shift key to access that mode?

You **may** not know that if you press and hold the CTRL key while hitting one of the **upshifted** keypad keys, the function labeled on the key is performed **without** being transmitted to the operating system. In other words, you can change your screen around without HDOS's or CP/M's noticing it, even if you **don't** press "OFF LINE." (One of my uses of this is to delay things from scrolling off the screen. When your cursor is at the bottom and you type a new command, the screen scrolls up 1 or more lines, and you lose the top lines. If you want to preserve for the moment what's at the top, do this first: Move the cursor up several lines with CTRL-uparrow. Then hit CTRL-ERASE to clean off everything below the cursor, and you're OK again for a while, with fresh workspace on the lower screen.)

The Wraparound Function. Let me dwell on the wraparound feature at greater length and use it as a model here, because I think it's widely misunderstood. But after we've explored it fully, you'll also be able to apply some of that knowledge to turn any of the other features on or off with either hardware **or** software.

I've run across at least one person who did not know, **at all,** what "wraparound" is, so first let's explain it. When you type (or the computer types) an entire line across your screen, one of **two things** will happen if you try to keep typing when you reach column 80: Either the cursor will stay in col 80 and any further characters you type will "pile up," displaying only your most-recently-typed character, **or** the cursor will automatically return to column 1 of the **next line,** and you can continue typing legibly. In the latter case the line is said to "wrap around," and that only happens when **wraparound** is enabled; and there we have defined the term. On the other hand, **discard** occurs in the former case, where the characters pile up (and get "discarded").

I keep wraparound enabled; I think "discard" is an absolute nuisance and almost never do I see any need for it. Yet, I strongly suspect **most** users just **put up** with discard mode on their terminals. Wraparound is one change you really might want to seriously consider making!

Another reader wrote me asking for a way to turn on wraparound at bootup. The following, in essence, is what I told him. (Again, the following enable/-disable techniques for the wraparound mode can **also** be applied to the other terminal features: **key click, shifted keypad, underscore cursor,** etc.)

I have a way to enable wraparound at bootup, with software, in CP/M. It's a little program you can run via the auto-command line in CONFIGUR. (Or, you may be able to add my program to a SUBMIT list instead, and call up the SUBMIT program in the CONFIGUR line.)

Hardware Configuration. But before getting to my little program, I must tell you, the **best** way to default your terminal to wraparound (or whatever) is with the **hardware**! After all, if wraparound is what you **want,** the hardware is **designed** to give it to you. Set the Terminal Logic Board DIP Switch S402 for wraparound! You either have an H-89 or an H-89A (mine is the "A"). If it is the older H-89, see your Operation Manual. But for mine, the switch S402 is on the TLB (the rear board), 3" from your right as you face the screen with the lid open, and I keep **down** the third little white toggle from the **left.** (Believe it or not, that is "section 2" of S402.) Be sure not to get the wrong DIP switch -- there is an S401 farther down on the board and farther to the right; you don't want that one.

I must emphasize what I touched on earlier: throwing these toggles does **not** in itself turn a feature ON or OFF, it merely **sets up** the computer so that when you **turn on the power,** or **hit**

SHIFT-RESET, the switch settings will become effective. Only then are the DIP switch settings looked at by circuitry, and not before.

Software Configuration. We come now to the "software" method, a little assembly program and step-by-step instructions how to install it:

```
        ORG     0100H
BDOS    EQU     005
        PUSH    PSW
        PUSH    B
        PUSH    D
        PUSH    H
        LXI     D,STRING
        MVI     C,9  ; 9=BDOS "Print string" functn
        CALL    BDOS
        POP     H
        POP     D
        POP     B
        POP     PSW
        JMP     0       ;Does warm boot.
STRING  DB      27,118  ;Decimal for ("ESC v"),
                        ;the code for wraparound ON
        DB      '$' ;Need at end of data w/functn 9
        END
```

Just type the above source with your editor, calling your file WRAP.ASM. Be sure to put WRAP.ASM on the same default drive with ASM.COM and LOAD.COM. Then type ASM WRAP next to the CP/M prompt for that drive. When the CP/M prompt reappears, type LOAD WRAP -- you'll then get the file WRAP.COM on your disk. It's an executable file that will send the 2 wraparound bytes to the terminal. Go into CONFIGUR and answer N when asked if you want a "Standard System." Then select menu option D. This brings up another menu. Here you must toggle item A until it is set to TRUE, then select item C and type WRAP as the command line. Exit CONFIGUR saving the changes (select item Y, and again Y). One more thing -- you must put a copy of **WRAP.COM** on the disk in drive A -- the disk you will boot from. Now do a "hard reset" and boot up cold. Program WRAP should run automatically and your terminal should now wrap around if you type a line longer than 80 characters.

Another method is to enter the bytes 27 and 118 (their HEX values = 1B,76) into a "scrap" disk file sector using a disk dump program (like Super Zap). Be sure to terminate those 2 bytes with 1A hex (that is, use 1B,76,1A). (One quick way to create the "scrap" file in the first place is with SAVE 1 FILENAME. If you terminate with the 1A as I state, then it doesn't matter what other garbage is in the SAVE file, because 1A hex is CTRL-Z, which is an End Of File byte.) Finally, use TYPE FILENAME as your CONFIGUR command line instead of running the assembly program WRAP.

An even simpler way is to run DDT and change **whatever** you see in memory at 0100H (etc.) to 1B,76,1A. Exit DDT with ^C and do a SAVE 1 FILENAME. Use TYPE FILENAME in CONFIGUR as described in the above paragraph. Of course, in any case, the commands "WRAP" or "TYPE FILENAME" will also work if entered as a user command line -- you don't put them into CONFIGUR unless you want them to auto-run at bootup.

By the way, notice in the ASM listing above, you can send ANY info to the screen this way, because whatever you put after the DB will be sent, until BDOS sees the $ sign. That makes it stop.

Finally, my correspondent also inquired about using BASIC instructions (PRINT CHR$(27) + CHR$(118)) to do the job at bootup. That **will** send the "ESCape v" code to the terminal, but I would definitely not use BASIC because: 1) You'd always need to have the BASIC interpreter on your bootup disk, 2) it takes **forever** for the interpreter to load and run just to send a couple little bytes to the terminal, 3) even if you **compile** the BASIC program, the COM file would be **way** too large, and 4) there **are** faster and more compact ways to do the job -- we've discussed some in this article, right folks? Write, folks!

=====

Troubleshooting the '89 - Part 1: Introduction

By Kirk L. Thompson with Daniel N. Jerome

INTRODUCTION. With this issue, **The Staunch 8/89'er** begins a sequence of articles on hardware troubleshooting. As mentioned in issue #22/23, Heath/Zenith has now discontinued servicing of the computer. Several long-time supporters of our systems continue to provide spare parts and/or service. These include:

Henry Fale
Quikdata Inc.
2618 Penn Circle
Sheboygan, WI 53081-4250
414/452-6854

Clay Montgomery
SigmaSoft and Systems
2413 Winterstone Dr.
Plano, TX 75023
214/596-0116

Lee Hart
TMSI
323 West 19th St.
Holland, MI 49423
616/396-5085

You should mark these names and addresses down in your "little black book" or file away hardcopy from a database in case your computer crashes and you need instant help!

You should also seriously consider maintaining an inventory of spare boards or even have a spare computer available. This equipment will be useful as a source of parts so that you can swap with those in your working computer as you narrow your problem to a specific board or component. (Caution: Do not attempt to swap boards or other parts at random, as this could cause more problems than it solves.) Spare boards will also be useful as temporary replacements when you send the bad one off for repair. Your inventory should include at least one each of the following:

Terminal Logic Board (TLB)
Central Processor (CPU) Board
Serial or Parallel Interface
16K RAM Expansion Board
Floppy Controller(s)
Keyboard

Other parts are less likely to be troublesome. For example, the CRT will usually last almost

indefinitely, although one of us (Kirk) has already seen one (a third-party amber tube) fail. Other things, such as replacement bridge rectifiers and voltage regulators for the power supply can be easily found at your local electronics parts supplier, for example, Radio Shack.

You should also keep your own skills in mind when troubleshooting your computer system. While replacing boards in the machine is fairly simple, replacing a burned-out bridge rectifier, for example, requires some background in electronics and soldering. The reason for this caution is that, in some cases during this series, you will be guided through repair procedures that you can do yourself. If you've assembled an electronics kit, that's all the better! But if you don't have the electronics background, yet keep the above-suggested supply of spare boards, you can at least **narrow** your trouble down to a specific board. And you can significantly reduce your repair expense because shipping cost alone for a single defective board is considerably **less** than it is to ship the entire, fully-loaded computer.

And even if you don't have any electronics background and your problem is less easily replaceable, such as in the power supply, a few simple tests with inexpensive equipment can narrow the trouble to a general location. With that knowledge, you could take the machine to a friend, to a local service agency for the repair, or send the stripped-down machine off to a servicer.

For the remainder of this introductory installment, we'll discuss additional material you may find helpful. For example, if your computer is still functioning reasonably well (that is, it can boot and run at least **some** programs), the system monitor ROM (MTR-88, MTR-89, or MTR-90) includes a few basic test routines. And there's quite a bit of software available that can aid you in your troubleshooting. You'll find much of the latter in **Staunch**'s software library. But first there's printed matter to consider.

REFERENCES AND GUIDES. There's some printed material that you may find helpful during the troubleshooting process. Only **one** of these was prepared specifically for use with the '89.

Zenith Data Systems **Z-89/90 Service Manual** (Part #860-8): this is the service manual for our computer. It includes sections, schematics, and diagrams for the TLB, CPU, and serial interface boards; memory expansion; and H-17 and H-37 controllers; and descriptions of the Siemens/Wangco and Tandon floppy drives. Some of this duplicates information already available in the Operations Manual for the '89 and '89A, but the information on the floppy controllers appears, to our knowledge, nowhere else. However, as is typical of this kind of material, it presumes existing experience. We've tried to locate a source for this service manual by contacting both Heath Co. and Zenith Data Systems. Unfortunately, it appears that whatever stock there was has been trashed!

Harry F. Beechhold, **Plain English Repair and Maintenance Guide for Home Computers** (Simon & Schuster, 1984, softcover): this book is one of a number that appeared during the first half of the 1980's on computer maintenance. This particular one is something of a "general" guide to the subject, that is, it doesn't specialize in any particular make or model. But it includes good, introductory material on electronics, chips, and general maintenance. It's a little short on machine-specific troubleshooting, but includes a chapter on "home-brew" projects. The six appendices cover number systems and digital gate logic; a brief discussion of soldering, desoldering, tools, parts, and instruments; a short troubleshooting guide; resistor coding, Ohm's Law, and a synopsis of CPUs and computer buses; a list of suppliers (now somewhat out of date) for I/O interfaces, tools, parts, and kits; and a bibliography. The book concludes with a glossary.

Robert G. Middleton, **New Handbook of Troubleshooting Techniques for Microprocessors and Microcomputers** (Prentice-Hall, 1984, hardcover): although the word "troubleshooting" appears in the title, this particular volume is **not** what you might initially think it is. It's actually a discussion of how chips (latches, flip-flops, counters, shift registers, and so forth) work and of computer **architecture** using four- and eight-bit CPUs (including the Z80) as examples. This is on a level "grittier" than the one we'll be working at, but this book could provide you with a knowledge of **how** that system on your desk functions. Like the Beechhold book, this one concludes with a chapter on "home-brew" test devices and has a (briefer) glossary.

Martin D. Seyer, **RS-232 Made Easy: Connecting Computers, Printers, Terminals, and Modems** (Prentice-Hall, 1986, hardcover): an earlier, softcover edition of this book was mentioned on these pages back in issue #2 (p.4). This is a good reference on how the "standard" RS-232 serial interface works. Even more useful are the charts (occupying about half the book) of computer, printer, and terminal connector pin-outs and cable wiring diagrams. Included among these are the '89 computer and most popular older-make printers. Appendices include extracts of the RS-232 and RS-449 specifications, a figure summarizing the CCITT V.24 equivalents of these two standards (CCITT is the European serial I/O standard), and a description of various tools and instruments for testing or adapting RS-232 connections. We **recommend** the inclusion of this book in your library.

Robert J. Kalman, et.al., **Microcomputer Troubleshooting** (Heathkit/Zenith Educational Systems, 1987, softcover, text - part no. EB-150, workbook - part no. EB-150-40): this two-volume set is the most detailed published material we've seen about computer troubleshooting. Regrettably, the caveat is that this course is directed toward PC clones and presumes an XT is available. It also presumes that an extensive selection of test instruments (multimeter, oscilloscope, logic probe, and diagnostic disk minimum) is around. This book set contains those troubleshooting details not likely to be found anywhere other than in a formal technical school course on computer repair (but, alas, for a different machine).

SOFTWARE. Software to assist in the troubleshooting process is available in either of two forms. There are, of course, extensive collections of diagnostic

programs on disk that perform various specific functions and you can find a good selection of those in **Staunch**'s library. Some will only run under HDOS, so if you use CP/M exclusively, you should consider bringing up a **minimal** HDOS system to run those programs. Write or call **Staunch** at the address and phone number given on the last page of this issue if you need assistance doing this.

However, there are also some elementary diagnostic routines built into the monitor ROM of the '89. Accessing those depends on which version of the ROM (MTR-88, MTR-89, or MTR-90) is installed in your computer. These routines provide you with a RAM memory test and a rotational speed test for **hard-sector** drive 0 (only!). (This drive will usually be your internal one, if installed.) These may be accessed via the keyboard when you are at the monitor ("H:") prompt. If your prompt, after you do a shift-reset, is **other** than "H:", you have a third-party ROM; check the documentation that came with the ROM to determine what tests are available. For the standard (Heath) ROMs, these tests are called with the following commands:

Monitor	Memory Test	Rotational Test
MTR-88	G(o)7375<Ret>	G(o)7372<Ret>
MTR-89	G(o)7375<Ret>	G(o)7372<Ret>
MTR-90	T(est Memory)	G(o)7372<Ret>

If you're unsure which ROM you have, there's a simple test to determine this. Press "V" at the monitor prompt. If the terminal beeps, you have either MTR-88 or MTR-89. If the computer completes the word "View", you have MTR-90; press shift-reset to return to the monitor prompt.

We should note several things about these tests. First, when using the "Go" command, once you press "G", the computer **automatically** adds "o" and a trailing space, then waits for you to key in the memory address given in the table. A similar situation applies to the "T" command, except that a trailing carriage return is also added. Thus, the latter test begins automatically. Second, when performing the memory test, you should let the pass number reach 377 and start over from 000. This will take up to a quarter hour, depending on how much RAM is installed. Third, the rotational speed test **requires** a disk in the drive. This is a non-destructive test, so any hard-sector disk will do. Further, the "speed" displayed on the screen should be as close to 200 as possible. A significant departure from 200 (such as 30 below or 10 above) will cause increased read/write error rates. Another thing to be aware of is that the result of this test and the speed determined in Heath's TEST17 program don't quite agree! But this test will at least make you aware of a trouble spot if you are having floppy drive problems. Either the memory or rotational speed tests may be interrupted with a shift-reset.

On-disk software to assist in troubleshooting covers a wide range of material. We recommend you include the following in your "toolkit":

Floppy drive/controller diagnostic (TESTxx)
RAM and ROM memory test(s)
Terminal logic board escape test(s)
Third-party hardware diagnostics

By way of notes on these suggestions, first, the available floppy drive diagnostics (TESTxx) will ONLY run under HDOS or as a separate, independent package. These were either originally included with HDOS 2.0 or provided with the H/Z-37 (soft-sector) controller and are specific to the controller the drive(s) are connected to. That is, TEST17 is only for hard-sector, TEST37 only for Heath/Zenith's soft-, and TEST47 for eight-inch; if you don't have HDOS, these can be had from **Staunch** as separate packages and were originally listed (except for TEST47) in issue #11. If you have another manufacturer's controller, such as those from C.D.R. or Magnolia Microsystems, testing may be problematical since we don't believe Heath's software will run reliably with either of these products. The reverse side of this same coin is that if you buy a **new** soft-sector controller today, the ones offered by Quikdata and TMSI are clones of H/Z's board. TEST37 will function with these controllers.

Second, although the monitor ROM in the '89 includes a RAM testing routine, you should have others in your library. The test in the ROM is not particularly sophisticated and will only detect gross problems, such as bad chips. Other more sophisticated tests will detect other kinds of trouble (such as intermittents) that would otherwise be difficult to track down.

A third class of diagnostic is specifically for the TLB. These programs determine whether the escape codes built into the TLB's ROM function properly. Often, they will require your interaction with them as they execute. One limitation of most of these programs is that they presume the standard board as produced by Heath. So they usually won't test the enhancements provided by such upgrades as "Super19" or "SuperSet". However, the codes for Heath's standard board are usually a **subset** of those provided in these upgrades, so you can at least test the basics.

Fourth and finally, if you have any third-party accessory equipment, diagnostic software may be included with them. In these cases, you will only be able to test these "non-Heath" units with the vendor's software. So be sure you have properly archived or stored the original floppies that came with these products.

CONCLUSION. In this introductory installment, we've covered essential preliminary material. We've mentioned a basic inventory of spare boards and parts you should maintain. We've discussed a number of books or manuals that could serve as references as you proceed with the hardware troubleshooting process. We've considered, in a general way, four classes of software you should include in your library that will assist you as you hunt for the problem(s) from which your computer may be suffering. We've also taken a closer look at the preparatory testing possible using the standard monitor ROM in the machine.

In installments to come, we'll take a closer look at specific boards or areas of the computer and **detail** how and what to look for as you troubleshoot. To assist in that task, **Staunch** is commissioning well-known personalities in the community supporting our computers to write specific segments in this series. That way, you can take

direct advantage of their knowledge and skills so that you can continue getting the most out of that '89 of yours. And we should reemphasize that the thrust of this series will be on the "how-to".

Finally, we welcome your input. Let us know what books, manuals, and software you've found particularly helpful in your own hardware troubleshooting. Meanwhile, in the next installment we'll look at the tools you'll need for the troubleshooting process (screwdrivers, IC puller, multimeter, hammer--when you get frustrated!--etc.) and one computer component that has **already** taken quite a pounding from you. That's, the keyboard. See you then!

=====

A Patch for Wordstar 3.0 and the Heath 89 Numeric Keypad

By Joseph Mendez

The Problem: I wanted to implement the cursor keys on the Heath '89 numeric keypad within WordStar 3.0. The Heath Users Group (HUG) offers a solution in their Keymap CP/M-80 program (part # 885-1230 [-37], $20.00). I'm not a member of HUG and since their national publication **REMark** is almost exclusively MSDOS-oriented I wasn't motivated to take out membership just for access to their moribund CP/M library. I had heard of the patchible nature of WordStar and wanted to try my hand at "fixing" it. My solution allows the 2, 4, 6 and 8 keys on the numeric keypad to be used as cursor keys when pressed simultaneously with the SHIFT key. This is not the ideal, one-keypress solution but it does allow the left hand to remain in a relatively natural position on the keyboard pressing the SHIFT key while the right hand makes the switch between the cursor keys and the main keyboard. I find it an improvement over the contortions my left hand made trying to hit the correct key on the WordStar diamond while simultaneously pressing the CONTROL key. The following paragraph is a step-by-step description of what I did to install a set of cursor keys in my numeric keypad.

The Solution: I have a fairly standard Heath '89 64K system with 3 90K hard-sector drives. In Drive A:, I install my system disk, which includes CP/M, BIOS.SYS and a variety of utilities. The most important utility, for purposes of this exercise is DDTZ.COM, a Z80-specific enhancement of DDT.COM written by Charles Falconer. A formatted disk with the two WordStar overlay programs: WSMSGS.OVR and WSOVLY1.OVR, is inserted in drive B:. The disk in Drive C: has a copy of the original WordStar 3.0 program, WS.COM. After booting up, we use DDTZ to load WS.COM into memory by entering on the command line: "DDTZ C:WS.COM" and RETURN. This will result in the following display, including the DDT "dash" prompt.

```
NEXT     PC     SAVE
3F00   0100       62
-
```

At the dash prompt enter the following: "s0649" and RETURN. What follows is a line display with a four-digit hex number (the address in memory), space, a two-digit hex number (the value at that address) and room for you to enter a new two-digit hex number.

```
0649 00 __
```

Every time you enter a two-digit value and RETURN you will be faced with a new line displaying the next memory address and the value it contains. Enter the following two-digit numbers in the order presented.

****

Table 1

1B 41 3E 64	CURSOR UP
1B 42 24 64	CURSOR DOWN
1B 43 5B 63	CURSOR RIGHT
1B 44 65 63	CURSOR LEFT

****

Once all 16 two digit numbers are entered press the PERIOD key to return to the DDTZ dash prompt. You exit DDTZ with a Control-C. At the CP/M command line enter the following:

```
SAVE 62 B:WS.COM
```

Your newly modified WordStar program will be written to disk and you are ready to run WS and test your new cursor keys.

The Explanation: Before the above will make any sense you need to know three things: the emitted codes from the H-89 keypad, WordStar 3.0 patch locations and their meanings, the use of the DDT(Z) (S)et command and CP/M's built-in SAVE command.

HEATH '89 NUMERIC KEYPAD CODES. Page 11-8 of the 1981 edition of the H-89A Operations Manual gives the character values emitted by the numeric keypad in its shifted and unshifted states. In the case of the shifted up-arrow; it emits an ESC A or in ASCII Hex 1B 41. Notice that this hex value corresponds to the first two hex values entered on the CURSOR UP line shown in Table 1. If you attempt to use the shifted up-arrow within the original WordStar you will get a capitalized A at the current cursor position. WordStar will ignore the ESC portion of the two-byte sequence 1B 41 and accept the second byte as a straight ASCII character. Our goal is to make WordStar recognize the shifted up-arrow as a two-byte command sequence to perform the predefined CURSOR UP action.

WORDSTAR 3.0 PATCH LOCATIONS. WordStar 3.0 commands are grouped in function tables between addresses 0248H and 0838H in memory. Many CP/M BBS's will have text files with a name like "WSPATCH.LBR" which identify the function labels and their location in memory. The CHUG library has a file by this name dated 1984. This file gives a fairly complete description of each label, the operation the label performs, and its location in memory. It is less complete in describing default settings, their meaning, and how to enter alternative settings. Using WSPATCH, I was able to locate the CURSOR UP command at memory location 04ADH. You can

examine this memory location by using the DDT Display command at the dash prompt as follows:

```
-d04AD
```

This results in the following display:

```
04AD 05 00 3E 64 xx xx xx xx ....etc.
```

I was finally able to decipher the above display with the help of a 1986 text, **The WordStar Customizing Guide** by Stuart E. Bonney. 05 is the ASCII representation of Control-E, the WordStar default command for UP CURSOR. 00 is an unused byte. 3E 64 is the address of the program routine that actually performs the called function. More generally, the format of the command table is as follows:

```
Control Character
|    Second Command Character (Optional)
|    |    Internal address of the called function
|    |    |    |          routine
|    |    |    |
v    v    v    v
00   00   00   00     or in the specific case of our
                            example
05   00   3E   64
```

Thus, we see that a one- or two-byte command is used to call the program logic located elsewhere in WordStar at a separate two-byte address. All we have to do is get the two-byte shifted up-arrow key (1B 41) to point to the address of the program logic that performs the UP CURSOR function. We could overwrite the default two-byte command sequence (05 00) with our new sequence; but then we would lose the WordStar diamond completely. As it turns out, there is space at the end of each function table for a limited number of additional commands. In the present example we are working in the Edit Command Table which has room for eight additional commands, more than enough to implement the four cursor keys. According to Bonney, the Edit Command Table Expansion Space begins at memory location 064DH.

USING DDT(Z) AND SAVE TO MAKE CHANGES TO WORDSTAR 3.0. There are many good CP/M texts that explain the operation and use of DDT and SAVE. I will limit my comments on these commands to our specific case. We are going to use DDT(Z) to make changes to our WordStar program in memory and then we are going to use information DDT(Z) provides us when we first load WS.COM in memory to SAVE the modified WS to disk.

When you first load DDTZ and WS.COM in memory, take note of the "SAVE" value and write it on a piece of paper. My example shows a value of 62. You will need it when you are ready to SAVE your modified WS.COM. If you only have the original DDT.COM that came with CP/M, record the hex value under NEXT.

If you look at the original solution above, you will see that we use the DDT(Z) **s** (SET) command with the beginning address of the Expansion Space. We enter hex numbers representing the two-byte command sequence followed by the two byte address of the appropriate function logic. We continue until all four cursor keys are entered. When you have exited DDTZ, the modified WS.COM remains in memory. You have to use SAVE to write the modified program to disk. SAVE requires that you enter the number of "pages" of memory that is to be saved in decimal form. DDTZ gives you that information up front. In this case, we are SAVEing 62 pages. If you are using DDT, you must convert the NEXT value (3F) to decimal (63) and subtract one from it to get the correct number of pages.

If you have WSPATCH.LBR along with this article, you should be able to figure out how to make many other changes to your WordStar command set. If you want to delve into WordStar customization more deeply, I highly recommend Bonney's book. Look for it in used bookshops or the bargain bin of regular bookstores.

Reference: Bonney, Stuart E., **WordStar Customizing Guide, The** (1986); ISBN 0-915381-81-8; published by Wordware Publishing, Inc.; 1506 Capital Avenue, Plano, TX 75074; ph: 1-800/231-7467; 1-214/-423-0090.

[Hank Lotz has a concluding comment to Joe's article: [Joe] "...says in his first paragraph that his solution (to use the cursor keys) also requires pressing the SHIFT key. This is to shift the keypad. As I have stated thousands of times in articles and elsewhere, I keep my keypad shifted at all times because the numerals are available on the top row, and the keypad is most useful for its unique editing functions (arrows, etc.). Therefore, **if** WS does not force a downshift of the keypad, Joe or a user of his patch could just configure the H-89 to have its keypad shifted (ESC x 6) before entering WS. Then he'd **have** a 'one-keypress solution.'" See Hank's discussion earlier in this issue.

[**Staunch** has Peter Shkabara's WSPATCH.LBR in its software library. This item is **not** the same as the WSPATCH.LBR mentioned by Joe. Pete's product adds the shifted keypad **and function keys** to WS 3.0 and 3.3 using SUBMIT and DDT. See the General Software Catalog or issue #7 for further information. Other WS patches are in the works! -Ed.]

=====

Pete on CP/M

By Peter Shkabara

Here we go again. It is a good thing that Kirk had a backlog of material to publish since it took me about eight months to get around to writing some more. I will start off by responding to some comments by Mark Hunt in the Jan-Apr '91 issue of **Staunch**. Mark asked why we can't use soft-sector disks with the H-17 controller. The answer is that we can - and it had been done (at least in HDOS) some time ago. Unfortunately, I do not remember which article it was in, but it seems about 1984 or so in **Remark**. Of course, it would require special device drivers in HDOS and CP/M. The latter would need changes in the BIOS.ASM file to incorporate the device driver. Only single density can be supported. You could conceivably then read other CP/M single density formats such as Osborne and Xerox (remember those Z80 machines?).

Mark also brought up dual format CP/M:HDOS disks. This is possible to do for soft- or hard-

sector. The way it works is that CP/M keeps its directory in the first 2 or 3 tracks of the disk, while HDOS puts it in the center tracks (20 of a 40 track disk). By creating dummy files in HDOS and CP/M, the directory and file area of each system can be protected. The CP/M files would be in the first half of the disk (before the HDOS directory) and the HDOS could be the second half. As long as the directory structures themselves are protected, you could use as much or as little of the disk for each system as you wanted. For single density soft sector, CP/M does not have any special markings in track 0 and all should be well. However, for double and extended density, the disk parameters are coded in the first sector of track 0. That is why the H-89 with Heath BIOS and H37 controller can read user-created disks. I don't know what HDOS does for track 0 on double density, but it is possible that they are compatible with CP/M - I don't know for sure. The procedures for doing this have been covered in other articles and I only wanted to give a cursory explanation of how it can be done.

About a year ago, Hank Lotz sent me a letter with a number of questions which could be used as topics for **Staunch** articles. Well... here I finally get around to answering some of them. The questions have been paraphrased from what he actually wrote, but the sense should be the same.

Question 1: How to position the head of an H-17 drive to any track desired? Answer: Why would you want to do it? Assuming you have a good reason for doing it, the procedures can be found in the Heath CP/M BIOS source code. This is in 8080 assembly language so it could be used under HDOS as well as CP/M. The particular routine is in the H-17 device driver and is labeled SDT (Seek Desired Track). Track stepping is done by repeated OUT instructions to the appropriate I/O port on the H-17 controller. If you do not relate to assembly language well, the sequence could be written in some other language, including (ugh!) BASIC using INP and OUT instructions.

Question 2: Can the H-17 controller provide a head position report of what track or sector it is reading? Answer: NO. Now that we know it can't, let us look at the reason why. This is not only for the H-17 controller, but the H-37 and others as well. The disk drive does not have a head position sensing mechanism except for detecting when the head is at track zero. Controllers must first reset the head to track zero, and then step it in the desired number of tracks. The software or smart controller IC must then keep track of track positioning (pun intended). It may be possible to query the software or the H-37 controller IC to determine what track you are on, but the information may not be reliable. As to the sector, the question is moot since by the time you got an answer (if it were possible) the sector would already be gone. Remember that the disk is spinning!

Question 3: How can one write memory resident programs to intercept terminal I/O or disk I/O calls? Answer: I have done this, but it requires a very good understanding of CP/M (or HDOS) and assembly language. Actually, to have the first (understand CP/M or HDOS) would qualify you for the second (understand assembly language). In HDOS it would be a device driver, and in CP/M it could load into high TPA area and make patches in CP/M to protect itself. If there are budding assembly language programmers out there who want such an article, let Kirk know and I will do it. It is too involved to cover it here.

[Concluded next issue. -Ed]

=====

CONTACTS

(A Wanted/For Sale/Swap Column)

Bernard L. Waltuck (1411 No. Flagler Dr., #7700, West Palm Beach, FL 33408, 407/650-6300) "For sale: 32K STATIC RAM for H-8. 2 - 8K Static RAM boards, $10.00 each; 1 - 16K Static RAM board, $15.00. All three boards removed from a working H-8. Complete with 1 manual for each board. Price plus shipping; buy all three boards and I'll pay shipping within continental U.S."

George Istvan (3187 Reva Drive, Concord, CA 94519, 415/687-8913) "Thank you for your letter of June 4th about placing an ad in the 'CONTACTS' column. Here is a list of items I would like to sell all together at $500 or best offer:

Heath H-89A, 64K RAM.
Z89-37 soft-sector controller installed, hard-sector controller (H-88-1) in box, never used.
Three floppy drives:
2 Siemens, 160K, single-sided;
1 Tandon 320K, double-sided;
two of the drives mounted in a Zenith Z-87 unit.
Two printers:
Epson RX-80;
Brother HR-15 daisy wheel with detached keyboard (allows printer to be used as a typewriter).
CP/M (V2.2) and copious other software."

THE STAUNCH 8/89'er, created by Hank Lotz, is a bimonthly newsletter on 8-bit H/Z computers. The editor is Kirk L. Thompson; P.O. Box 548; West Branch, IA 52358; home: 319/643-7136. Subscriptions always start and end with the calendar year. Rate: $12.00/year. (Overseas, add $4.) Single copies: $2. Make checks payable to "Kirk L. Thompson". **Staunch** pays authors for their articles; write for an author's guide. It also accepts commercial ads for a modest fee; contact the editor. Neither this newsletter nor its editor is responsible for damages or losses resulting from use of any information presented herein. Info from **THE STAUNCH 8/89'er** may be reprinted only if this publication's name and address is included. Credit should also be given to authors and other sources of said material, if known. This publication is archived by the University of Iowa Libraries. CP/M is a registered trademark of Digital Research, Inc. **REMark** is a registered trademark of Zenith Users' Group. EOF