REMark

Issue 22 • November 1981

Official magazine for users of Heath computer equipment.

on the cover . . . .

NOVEMBER MORNING — —

PHOTO BY: GERRY KABELMAN

on the stack

>CAT

"REMark" is a HUG membership magazine published ten times yearly. A subscription cannot be purchased separately without membership. the following rates apply.

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

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

Back issues are available at $2.50 plus 10% handling and shipping. Requests for magazines mailed to foreign countries should specify mailing method and add the appropriate cost.

Send payment to:

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

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

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

HUG Manager and Editor .... Bob Ellerton
Assistant Editor and
Software Developer .... Patrick Swayne
HUG Secretary .... Nancy Strunk
Software Developer .... Gerry Kabelman
HUG BB .... Terry Jensen

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

REMark

A Little Help From Our Friends

When I received the September Issue of H8SCOOP from Henry Fale I was truly impressed with a small article under his "PEEKING & POKING" section entitled THE BIG THREE. For those of you who are new to the fast moving world of personal computers, particularly Heath/Zenith computers, you will find a need for all the information that you can collect. Henry's article speaks for itself. So, I gave him a call to see if HUG could share his observations with readers of REMark as well. Here it is folks!

THE BIG THREE

The more I read other newsletters and talk to their editors, the more I am realizing we all have to work together, and yes, we all need each other. I'm mainly speaking of the big 3, on order of circulation REMark, BUSS and H8SCOOP. When I started H8SCOOP I did not start with the intention of running REMark or BUSS out of business, but to give you somthing different and extra to compliment the others. I figured the wise and serious computer individual will already be subscribing to REMark and BUSS.

Although some similar articles pop up sometimes in two or three of these newsletters, I try to discourage it knowing that most of you do subscribe to more than one. If you have been following H8SCOOP you know what my policy has been on that. The fact is, that sometimes it does happen and it is not avoidable since we (the editors) have no idea what has been submitted to the others and what they are going to print!

The format of H8SCOOP is totally different than that of BUSS, and is also totally different than that of REMark. They all have their places. While BUSS and H8SCOOP are more current than REMark, REMark has more long articles and general interest articles. This can be understood when you look at the average length of REMark being 32 pages and the others 8 pages. Because REMark has the glossier pages and slick format it takes longer to print. Does that mean nix on REMark? No way. It has its place and has been steadily improving. In fact they have all been steadily improving. That's a good sign because it means that they are all getting your support, and they are all providing support to make the Heathkit line of computers more useful and popular.

I feel it would be a major loss to lose any of THE BIG THREE at this point and I advocate your support of them by subscribing and sending articles, hints and information. I would not give up my subscription to REMark or BUSS for anything--they're too valuable to me. If you like them, then support them. Its always amusing that the ones who give least or no support are the ones who complain the most. This point was explained in a past H8SCOOP about my "friend" who points out and criticizes my spelling--He's not even a subscriber!

The more we stick together and help each other, the better we'll all be.

Henry Fale

As an additional comment, what Henry said about THE BIG THREE can be applied to your local users' group too. Also, the more we, as a unit, work with each other on an individual basis, the more we will see a growth in our chosen machines and "tools" that will go with them. Thanks to Henry of H8SCOOP, Charlie of BUSS, and all of the other individuals who continue to provide outstanding support for the Heath/Zenith computers.

For those of you who would like additional information on either H8SCOOP or BUSS here are a few more details:

H8SCOOP
Henry Fale (Editor)
2918 South 7th Street
Sheboygan, WI 53081

Rates: $15/YR. ($21/YR. overseas)

BUSS
Charlie Floto (Editor)
325 Pennsylvania Avenue, S.E.
Washington, DC 20003

Rates: $18/YR. ($25/YR. overseas)

EOF BE:

ATTENTION!

For those of you who are moving or about to move, be sure to forward to HUG, your new mailing address. Nancy tells us that failure to do so will result in lost issues of REMark since they ARE NOT forwarded!

Clock Watcher's Delight

When I first saw the CPS modem program in operation, I noticed right away the elapsed time clock that it maintains in the lower right corner of the screen. I thought that it would be nice to have a time of day clock similar to that on the screen all of the time. When Dale Lamm sent us his CKDVD.ASM over MicroNET, he provided me with the perfect start for a screen clock. When I had completed it, I liked it so much that I modified the Heath CP/M BIOS to make a clock for that operating system. In this article, I will present the modifications to CK.DVD and the CP/M BIOS that are necessary to make the screen clock.

My screen clock maintains a 24 hour time display in the upper right hand corner of the H19 or H89 screen. Each second, it saves the current cursor position, moves up to the upper right corner to write the time, and restores the cursor to where it was. To illustrate, if the current time is 8:45 am exactly, the screen shows 08:45:00 in the upper right hand corner. One of the first questions I am asked when I show people my clock is "Why didn't you put it on the 25th line?". The main reason is that I wanted to be able to have the clock going while I used other software that uses the 25th line, such as editors and word processors. The clock would interfere with such programs, and they could interfere with it. Because any user program can disable the 25th line, my clock would have to re-enable it every second. This would mean transmitting more ASCII characters each time, and by computer standards, it takes a lot of time to transmit just one character, even at 9600 baud. By using the upper right corner for the clock, I am able to run just about any program with it going except fast action graphic games, such as SEA BATTLE or INVADERS.

PART ONE -- HDOS

To implement the screen clock in HDOS, the first thing you need to do is type in the source for CK.DVD which starts on page 14 of REMark #20. Then add the following definitions to the EQU's at the beginning of the program.

```
SPORT    EQU      355Q             CONSOLE STATUS PORT
DPORT    EQU      350Q             CONSOLE DATA PORT
OUTBIT   EQU      40Q              OUTPUT READY BIT
S.SYSM   EQU      40320A           FWA RESIDENT SYSTEM
S.RFWA   EQU      40356A           FWA RESIDENT HDOS CODE
AIO.DTA  EQU      41053A           DEVICE TABLE ADDRESS
DEV.RES  EQU      2                OFFSET TO RESIDENCY FLAG
DR.PR    EQU      00000010B        FLAG BIT; PERMANENTLY RESIDENT
```

The port definitions are needed because the time data is written directly to the port, thus avoiding a system call within the driver. If you have an H8 with the H8-5 card, replace the first three equates with these:

```
SPORT    EQU      373Q             CONSOLE STATUS PORT
DPORT    EQU      372Q             CONSOLE DATA PORT
OUTBIT   EQU      1                OUTPUT READY BIT
```

To compensate for the time it takes to write the time on the screen each second (approximately 16 ms), I had to change the calibration factor CAL as follows.

```
CAL      EQU      -1-491           CLOCK CALIBRATION
```

This results in a clock that runs slightly fast, to allow for 5-inch disk accesses, which slow the clock some. If you use 492 instead of 491, you will have a nearly perfect clock. These values are for a 9600 baud terminal. If you use 19.2k baud, you will probably get good results with 495 or 496.

I modified the LOAD processor so that the driver is made permanently resident whether you load it with the HDOS LOAD command or from a program. The modified routine is listed below, with the added lines bracketed.

```
LOAD     EQU      *                COME HERE WHEN CK: IS LOADED
        ┌LHLD     S.SYSM           UPDATE SYSTEM FWA
        │SHLD     S.RFWA
        LHLD    AIO.DTA         GET DEVICE TABLE ADDRESS
        LXI     D,DEV.RES       OFFSET TO RESIDENCY FLAG
        DAD     D
        MOV     A,M
        ORI     DR.PR           FLAG CK: AS PERMANENT
        MOV     M,A
        LHLD    UIVEC+1         GET THE HDOS "TICCNT" VECTOR
        SHLD    CLKRET+1        INSTALL AT END OF CLOCK ROUTINE
        LXI     H,CLOCK         GET OUR CLOCK'S START ADDRESS
        SHLD    UIVEC+1         REPLACE THE HDOS "TICCNT" VECTOR
        XRA     A               CLEAR THE CARRY BIT
        RET                     RETURN, SHOWING NO ERROR
```

The next modification is to the clock processor itself. Starting at the line CLOCK EQU *, change every jump or conditional jump to CLKRET except the first one to a jump (or conditional jump) to PTIME. Then add the following lines before and after the line CLKRET EQU * as follows.

```
PTIME   LXI     H,SAVCUR
        CALL    PRINT           SAVE CURSOR AND MOVE IT
        LXI     H,TIMEBUF
        CALL    PRINT           PRINT TIME
        LXI     H,RESCUR
        CALL    PRINT           RESTORE CURSOR
        MVI     C,134           SET UP DELAY
DLY     DCR     C               WAIT 1 MS (FOR PORT TO CLEAR)
        JNZ     DLY
CLKRET  EQU     *               NOW CONTINUE WITH HDOS CLK INT
        JMP     0               NEW ADDRESS INSTALLED BY "LOAD"
PRINT   MOV     A,M             GET A CHARACTER
        CPI     NL              END OF STRING?
        RZ                      IF SO, RETURN
        CALL    OUTCH           PRINT CHARACTER
        INX     H               INCREMENT POINTER
        JMP     PRINT
OUTCH   MOV     C,A             SAVE CHARACTER
OUTCH1  IN      SPORT           CHECK CONSOLE STATUS
        ANI     OUTBIT          READY TO SEND?
        JZ      OUTCH1          LOOP UNTIL READY
        MOV     A,C             RESTORE CHARACTER
        OUT     DPORT           TRANSMIT IT
        RET
```

The final modification to CK.DVD is to add the following lines to the "STORAGE AREAS" part of the code.

```
SAVCUR  DB      27,'j',27,'Y h',NL
RESCUR  DB      27,'k',NL
```

Once CK.DVD is modified, assembled, and loaded, it can be read and set from the console as before. However, to make setting the clock easier, I wrote a program called SETTIME.ABS that prompts you to enter the time. Then you hit RETURN, and the time is set. SETTIME allows you to use any delimiter character between the numbers, and allows you to drop the leading zero on hours less than 10. For example, if you are setting the time to 7:30 am, you could enter 7-30-00, or 7/30/00, etc. You can also use SETTIME to reset the clock to 00:00:00 by hitting RETURN in response to the prompt. If you rename SETTIME.ABS to PROLOGUE.SYS, you can have HDOS prompt you for the time at boot-up. The source for SETTIME is listed below.

```
SETTIME -- TIME SETTING PROLOGUE FOR CK:                    HEATH ASM #104.06.00
BY P. SWAYNE  2-SEP-81                                      18-Sep-81  Page    1

                         00003  * THIS PROGRAM CAN BE USED TO SET THE TIME IN DALE LAMM'S
                         00004  * CK.DVD CLOCK DEVICE DRIVER (AS MODIFIED BY P. SWAYNE).  IF
                         00005  * THE .ABS FILE IS NAMED PROLOGUE.SYS, IT WILL ALLOW YOU TO
                         00006  * SET THE TIME AT BOOTUP.
                         00007
   000.000               00008  .EXIT   EQU     0
   000.001               00009  .SCIN   EQU     1
```

```
000.005                .WRITE  EQU    5
000.040                .LINK   EQU    40Q
000.043                .OPENW  EQU    43Q
000.046                .CLOSE  EQU    46Q
000.057                .ERROR  EQU    57Q
000.062                .LOADD  EQU    62Q
031.136                $TYPTX  EQU    195EH

042.200                        ORG    2280H

042.200  041 017 043   SETTIME LXI    H,CKNAME
042.203  377 062               SCALL  .LOADD           LOAD CK:
042.205  332 004 043           JC     ERR
042.210  315 136 031   GETIME  CALL   $TYPTX
042.213  012 124 151           DB     12Q,'Time (HH-MM-SS)?',240Q
042.235  041 053 043           LXI    H,TIMBUF         PUT TIME HERE
042.240  377 001       TILP    SCALL  .SCIN            GET A CHARACTER
042.242  332 240 042           JC     *-2              NOT READY YET
042.245  376 012               CPI    12Q              END OF LINE?
042.247  312 257 042           JZ     GOTIME
042.252  167                   MOV    M,A              STORE IT
042.253  043                   INX    H                INCREMENT POINTER
042.254  303 240 042           JMP    TILP             GET MORE
042.257  072 054 043   GOTIME  LDA    TIMBUF+1         GET SECOND CHARACTER
042.262  041 052 043           LXI    H,TIMBUF-1       ASSUME IT IS DELIMITER
042.265  376 060               CPI    '0'              LESS THAN ZERO?
042.267  332 302 042           JC     NLZ              IF SO, NO LEADING ZERO
042.272  376 072               CPI    '9'+1            GREATER THAN NINE?
042.274  322 302 042           JNC    NLZ
042.277  041 053 043           LXI    H,TIMBUF         POINT TO TIME ENTERED
042.302  345           NLZ     PUSH   H                SAVE ADDRESS
042.303  043                   INX    H
042.304  043                   INX    H                MOVE TO FIRST COLON
042.305  066 072               MVI    M,':'            INSERT COLON
042.307  043                   INX    H
042.310  043                   INX    H                MOVE TO SECOND COLON
042.311  043                   INX    H
042.312  066 072               MVI    M,':'            INSERT COLON
042.314  341                   POP    H                GET ORIGINAL ADDRESS
042.315  345                   PUSH   H                SAVE ADDRESS

                       * CHECK IF ENTRY IS ALL NUMBERS

042.316  016 010               MVI    C,8              CHECK 8 CHARACTERS
042.320  176           CHLP    MOV    A,M
042.321  376 060               CPI    '0'              LESS THAN ZERO?
042.323  332 013 043           JC     GETNEW           BAD ENTRY
042.326  376 073               CPI    ':'+1            MORE THAN ":"?
042.330  322 013 043           JNC    GETNEW           BAD ENTRY
042.333  043                   INX    H
042.334  015                   DCR    C                DONE?
042.335  302 320 042           JNZ    CHLP             IF NOT, CONTINUE

                       * SEND TIME TO CK: DEVICE

042.340  041 017 043           LXI    H,CKNAME
042.343  021 023 043           LXI    D,DEFALT
042.346  257                   XRA    A
042.347  377 043               SCALL  .OPENW           OPEN FOR WRITE
042.351  332 004 043           JC     ERR
042.354  321                   POP    D                DE = ADDRESS OF TIME
042.355  001 011 000           LXI    B,9              SEND 9 CHARACTERS
042.360  257                   XRA    A
042.361  377 005               SCALL  .WRITE           WRITE THE CHARACTERS
042.363  332 004 043           JC     ERR              NO GOOD
042.366  257                   XRA    A
042.367  377 046               SCALL  .CLOSE           CLOSE THE FILE
042.371  332 004 043           JC     ERR
042.374  041 031 043   EXIT    LXI    H,FNAME          FILE NAME TO LINK TO
042.377  377 040               SCALL  .LINK            TRY TO LINK TO IT
```

```
043.001  257          00080           XRA     A
043.002  377 000      00081           SCALL   .EXIT               ELSE EXIT
                      00082
                      00083  * PROCESS ERRORS
                      00084
043.004  046 007      00085  ERR      MVI     H,7
043.006  377 057      00086           SCALL   .ERROR              REPORT ERROR
043.010  303 374 042  00087           JMP     EXIT                AND LINK ANYHOW
043.013  341          00088  GETNEW   POP     H                   RESTORE STACK
043.014  303 210 042  00089           JMP     GETIME              GET NEW DATA
                      00090
                      00091  * STORAGE
                      00092
043.017  103 113 072  00093  CKNAME   DB      'CK:',0
043.023  123 131 060  00094  DEFALT   DB      'SY0',0,0,0
043.031  123 131 060  00095  FNAME    DB      'SY0:SYSCMD.SYS',0,0,0
043.052  060          00096           DB      '0'                 LEADING ZERO
043.053  060 060 072  00097  TIMBUF   DB      '00:00:00'
                      00098
043.063  000          00099           END     SETTIME
```

PART TWO -- CP/M

In CP/M, communication to the outside world is done through the BIOS instead of through separate device drivers, so I added a screen clock by modifying the bios. You could also make a clock by writing a separate program that locates itself in high memory like DDT, but the BIOS modification was easier and did not require intercepting the interrupt vector.

All modifications were made to the BIOS's TICCNT processor. This code was cleverly written to use only one level on the stack, so I made sure that my changes do not use the stack. My modification requires the DE register pair, which the BIOS does not use, so it must be saved. Modify the code at the label "clock" as follows.

```
clock   shld    hsav            ;save hl
        pop     h               ;get return address
        shld    retsav          ;save it
        push    psw             ;save a,f
        XCHG                    ;MOVE DE TO HL
        SHLD    DSAVE           ;SAVE DE
```

Note that since the BIOS is written in lower case letters I can make my modifications stand out by putting them in upper case letters. This is the procedure I followed in the rest of the modification, which is listed below.

```
clkret: lda     ticcnt
        rar                     ;is it even, making 4mS big ticks
        jc      clkr2
        lxi     h,dlyw          ;check wait timer
        mov     a,m             ;  and decrement it if it is not
        ora     a               ;  already zero
        jz      clkr2
        dcr     m
clkr2:  LDA     ENABL           ;GET ENABLE BYTE
        ORA     A               ;TEST IT
        JZ      RCKRET          ;CLOCK DISABLED
        LHLD    TICKS           ;GET OUR TIC COUNTER
        INX     H               ;INCREMENT IT
        SHLD    TICKS           ;UPDATE IT
        XCHG                    ;SAVE IN DE
        LHLD    CALFAC          ;GET CALIBRATION FACTOR
        XCHG                    ;HL = TICKS, DE = CAL FACTOR
        DAD     D               ;HAVE WE PASSED IT?
        JNC     RCKRET          ;NOT TIME FOR A NEW SECOND
        SHLD    TICKS           ;CLEAR TIC COUNTER
        MVI     D,'0'           ;GET AN ASCII ZERO
INRS    LXI     H,TIMEBUF+7     ;POINT AT UNITS SECONDS
        INR     M               ;INCREMENT IT
        MOV     A,M
        CPI     '9'+1           ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
        MOV     M,D                 ;RESET UNITS SECONDS
INRTS   DCX     H                   ;POINT AT TENS SECONDS
        INR     M                   ;INCREMENT IT
        MOV     A,M
        CPI     '6'                 ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
        MOV     M,D                 ;RESET TENS SECONDS
INRM    DCX     H
        DCX     H                   ;POINT AT UNITS MINUTES
        INR     M                   ;INCREMENT IT
        MOV     A,M
        CPI     '9'+1               ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
        MOV     M,D                 ;RESET UNIT MINUTES
INRTM   DCX     H                   ;POINT AT TENS MINUTES
        INR     M                   ;INCREMENT IT
        MOV     A,M
        CPI     '6'                 ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
        MOV     M,D                 ;RESET TENS MINUTES
INRH    DCX     H
        DCX     H                   ;POINT AT UNITS HOURS
        INR     M                   ;INCREMENT IT
        MOV     A,M
        CPI     '4'                 ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
INRH1   DCX     H                   ;POINT AT TENS HOURS
        MOV     A,M
        CPI     '2'                 ;PASSED MIDNIGHT?
        JM      INRH2               ;IF NOT, CHECK MORE
        MOV     M,D                 ;RESET TENS HOURS
        INX     H                   ;POINT AT UNITS HOURS
        MOV     M,D                 ;RESET UNITS HOURS
        JMP     PTIME               ;PRINT TIME (MIDNIGHT)
INRH2   INX     H                   ;POINT AT UNITS HOURS
        MOV     A,M
        CPI     '9'+1               ;OVERFLOW?
        JM      PTIME               ;IF NOT, PRINT TIME
        MOV     M,D                 ;RESET UNITS HOURS
INRTH   DCX     H                   ;POINT AT TENS HOURS
        INR     M                   ;INCREMENT IT
;^L
; PRINT TIME IN UPPER RIGHT CORNER OF CONSOLE

PTIME   LXI     D,SAVCUR
        LXI     H,PTIME1
        JMP     PRINT               ;SAVE CURSOR AND MOVE IT
PTIME1  LXI     D,TIMEBUF
        LXI     H,PTIME2
        JMP     PRINT               ;PRINT TIME
PTIME2  LXI     D,RESCUR
        LXI     H,PTIME3
        JMP     PRINT               ;RESTORE CURSOR
PTIME3  MVI     D,134               ;SET UP DELAY
TIMDLY  DCR     D                   ;WAIT ABOUT 1 MS (FOR PORT TO CLEAR)
        JNZ     TIMDLY
RCKRET  pop     psw                 ;restore the machine state
        lhld    retsav
        push    h
        LHLD    DSAVE
        XCHG
        lhld    hsav
        ei
        ret
PRINT   LDAX    D                   ;GET A CHARACTER
        ORA     A                   ;END OF STRING
        STA     SAVCHR              ;SAVE CHARACTER
        JNZ     PRINT1              ;IF NOT, MOVE ON
        PCHL                        ;RETURN
```

```
PRINT1  LDA     mode
        ANI     1               ;H8-5 CARD?
        JNZ     H5PRT
OUTCH   IN      355Q            ;CHECK CONSOLE STATUS
        ANI     40Q             ;READY TO SEND?
        JZ      OUTCH           ;LOOP UNTIL READY
        LDA     SAVCHR          ;GET CHARACTER
        OUT     350Q            ;SEND IT
        INX     D               ;INCREMENT POINTER
        JMP     PRINT
H5PRT   IN      373Q
        ANI     1
        JZ      H5PRT
        LDA     SAVCHR
        OUT     372Q
        INX     D
        JMP     PRINT
TICKS   DW      0               ;OUR CLOCK'S TIC COUNTER
CALFAC  DW      (-1)-491        ;CALIBRATION FACTOR
ENABL   DB      0FFH            ;CLOCK ENABLE BYTE
TIMEBUF DB      '00:00:00'      ;TIME OF DAY MAINTIANED HERE
        DB      0
SAVCUR  DB      27,'j',27,'Y h',0
RESCUR  DB      27,'k',0
DSAVE   DW      0               ;SAVE DE HERE
SAVCHR  DB      0               ;SAVE CHARACTER HERE
;^L
;
; xok - exit from disk operation without error
; xit - exit flagging error in carry
;

xok:    xra     a               ;clear carry
xit:    push    psw
```

In HDOS, if you don't want the screen clock, you can simply choose not to load the device driver, but in CP/M, with the clock in the BIOS, we are stuck with it, so I wrote a special SETTIME program. It allows you to disable the clock or to change the calibration factor by 2 ms steps. To enable the clock (if it is disabled) and set the time, enter

```
SETTIME(cr)
```

where (cr) is the RETURN key. You will be prompted to enter the time, which is done in the same way as with the HDOS SETTIME program. To disable the clock, enter

```
SETTIME D(cr)
```

To make it run faster by 2 ms per second, enter

```
SETTIME F(cr)
```

and to make it run slower, enter

```
SETTIME S(cr)
```

Repeated uses of F or S are cumulative. The source for SETTIME is listed below. This program writes changes directly into the BIOS, so you must be sure that the offset to the ASCII time buffer (TIMADR) is correct if you make further modifications to the BIOS. This address is labeled TIMEBUF in the BIOS modification. You can locate it by assembling the BIOS at ORG 0 and making a .PRN file. Find the address of TIMEBUF in the .PRN file and subtract 3 from it to get the value for TIMADR (because we are measuring from the warm boot entry point). You can also DDT BIOS.SYS, search for the string "00:00:00" with the D command, and subtract 203H from its address to get TIMADR.

```
; SETTIME
;
; THIS PROGRAM IS USED TO SET THE BIOS REAL TIME CLOCK
; AS IMPLEMENTED BY P. SWAYNE
```

```
                   ; BY P. SWAYNE  03-SEP-81

                   ; THE FOLLOWING LABEL IS THE OFFSET INTO THE BIOS TO
                   ; LOCATE THE ASCII TIME STRING USED BY THE REAL TIME
                   ; CLOCK.  THIS ADDRESS MAY CHANGE IF THE BIOS IS MODIFIED,
                   ; SO BE SURE TO CHANGE IT ACCORDINGLY.

074D =             TIMADR  EQU     74DH

074C =             ENABLE  EQU     TIMADR-1        ;CLOCK ENABLE BYTE
074A =             CALFAC  EQU     TIMADR-3                ;CALIBRATION FACTOR

0100                       ORG     100H

0100 210000        SETTIME LXI     H,0
0103 39                    DAD     SP              ;LOCATE CURRENT STACK
0104 22D101                SHLD    OLDSP           ;SAVE IT
0107 31DD02                LXI     SP,STACK        ;SET A NEW ONE
010A 3A5D00                LDA     5DH             ;GET FIRST FCB LETTER
010D 2A0100                LHLD    1               ;GET BIOS ADDRESS
0110 114C07                LXI     D,ENABLE        ;GET CLOCK ENABLE OFFSET
0113 19                    DAD     D               ;FIND CLOCK ENABLE ADDRESS
0114 36FF                  MVI     M,0FFH          ;ENABLE JUST IN CASE
0116 FE44                  CPI     'D'             ;DISABLE?
0118 C22001                JNZ     CHKF            ;IF NOT, CHECK FOR "F"
011B 3600                  MVI     M,0             ;DISABLE CLOCK
011D C3A601                JMP     EXIT            ;AND LEAVE
0120 FE46          CHKF    CPI     'F'             ;SET CLOCK FASTER?
0122 C23601                JNZ     CHKS            ;IF NOT, CHECK FOR "S"
0125 2A0100                LHLD    1               ;GET BIOS ADDRESS
0128 114A07                LXI     D,CALFAC
012B 19                    DAD     D               ;LOCATE CALIBRATION FACTOR
012C 5E                    MOV     E,M             ;GET LOW BYTE
012D 23                    INX     H
012E 56                    MOV     D,M             ;AND HIGH BYTE
012F 13                    INX     D               ;INCREMENT IT
0130 72                    MOV     M,D             ;AND REPLACE IT
0131 2B                    DCX     H
0132 73                    MOV     M,E
0133 C3A601                JMP     EXIT            ;LEAVE
0136 FE53          CHKS    CPI     'S'             ;SET CLOCK SLOWER?
0138 C24C01                JNZ     GETIME          ;IF NOT, GET TIME
013B 2A0100                LHLD    1               ;GET BIOS ADDRESS
013E 114A07                LXI     D,CALFAC
0141 19                    DAD     D               ;LOCATE CALIBRATION FACTOR
0142 5E                    MOV     E,M             ;GET LOW BYTE
0143 23                    INX     H
0144 56                    MOV     D,M             ;AND HIGH BYTE
0145 1B                    DCX     D               ;DECREMENT IT
0146 72                    MOV     M,D             ;AND REPLACE IT
0147 2B                    DCX     H
0148 73                    MOV     M,E
0149 C3A601                JMP     EXIT            ;EXIT TO CP/M
014C 11AF01        GETIME  LXI     D,ENTER         ;GET MESSAGE
014F 0E09                  MVI     C,9             ;PRINT FUNCTION
0151 CD0500                CALL    5               ;PRINT MESSAGE
0154 11D301                LXI     D,BUFFER        ;PUT ENTRY HERE
0157 0E0A                  MVI     C,10            ;READ CONSOLE FUNCTION
0159 CD0500                CALL    5               ;GET ENTRY
015C 3E30                  MVI     A,'0'           ;GET ASCII ZERO
015E 32D401                STA     BUFFER+1        ;OVERLAY CHAR COUNT
0161 3AD601                LDA     BUFFER+3        ;GET SECOND CHARACTER
0164 21D401                LXI     H,BUFFER+1      ;ASSUME IT IS DELIMITER
0167 FE30                  CPI     '0'             ;LESS THAN ZERO?
0169 DA7401                JC      NLZ             ;IF SO, NO LEADING ZERO TYPED
016C FE3A                  CPI     '9'+1           ;GREATER THAN 9?
016E D27401                JNC     NLZ
0171 21D501                LXI     H,BUFFER+2      ;POINT TO TIME ENTERED
0174 E5            NLZ     PUSH    H               ;SAVE IT
0175 23                    INX     H
0176 23                    INX     H               ;MOVE TO FIRST DELIMITER
```

```
0177 363A                MVI     M,':'           ;REPLACE WITH COLON
0179 23                  INX     H
017A 23                  INX     H
017B 23                  INX     H               ;MOVE TO SECOND DELIMITER
017C 363A                MVI     M,':'           ;REPLACE WITH COLON
017E E1                  POP     H               ;RESTORE TIME ADDRESS
017F E5                  PUSH    H               ;SAVE AGAIN

               ; CHECK IF ENTRY IS GOOD

0180 0E08                MVI     C,8             ;CHECK 8 CHARACTERS
0182 7E        CHLP      MOV     A,M
0183 FE30                CPI     '0'             ;LESS THAN ZERO?
0185 DAAB01              JC      GETNEW          ;BAD ENTRY
0188 FE3B                CPI     ':'+1           ;MORE THAN ":"?
018A D2AB01              JNC     GETNEW          ;BAD ENTRY
018D 23                  INX     H
018E 0D                  DCR     C               ;DONE?
018F C28201              JNZ     CHLP            ;IF NOT, CONTINUE

               ; SET TIME IN BIOS.SYS

0192 2A0100              LHLD    1               ;GET BIOS ADDRESS
0195 114D07              LXI     D,TIMADR        ;AND TIME ADDRESS
0198 19                  DAD     D               ;FIND TIME IN MEMORY
0199 D1                  POP     D               ;GET TIME ENTERED ADDRESS
019A 0E08                MVI     C,8             ;MOVE 8 CHARACTERS
019C F3                  DI                      ;DISABLE INTERRUPTS FOR MOVE
019D 1A        MOVLP     LDAX    D               ;GET A CHARACTER
019E 77                  MOV     M,A             ;PUT IT IN MEMORY
019F 13                  INX     D
01A0 23                  INX     H               ;INCREMENT POINTERS
01A1 0D                  DCR     C               ;DONE?
01A2 C29D01              JNZ     MOVLP           ;IF NOT, CONTINUE
01A5 FB                  EI                      ;RESTORE INTERRUPTS
01A6 2AD101    EXIT      LHLD    OLDSP           ;GET OLD STACK POINTER
01A9 F9                  SPHL                    ;SET IT
01AA C9                  RET                     ;RETURN TO CP/M
01AB E1        GETNEW    POP     H               ;FIX STACK
01AC C34C01              JMP     GETIME

01AF 0D0A456E74ENTER     DB      0DH,0AH,'Enter current time (HH-MM-SS): ','$'
01D1           OLDSP     DS      2
01D3 0A0030303ABUFFER    DB      10,0,'00:00:00'
02DD =         STACK     EQU     $+100H

01DD                     END
                                                                PS:
```

Continuing in CP/M

Last issue of REMark, we began the first in a series of articles on how to use the CP/M operating system. We briefly covered the history and basics of an operating system. It was pointed out that both of the Heath 8-bit computers, the H8 and H89, need the Extended Configuration Option in order to run CP/M. We advised purchasing additional material for learning CP/M because most beginners have found that the documentation supplied with CP/M is difficult to understand. That primarily brings us to where we can begin.

<u>"Getting Started in CP/M".</u>

The first step in starting to use CP/M will be for us to introduce the basics of the CP/M operating system. This must be done before we even turn on the computer. If you understand the basics, then when you sit down at the terminal, you will find your mind clear of uncertainties about the operating system and the functions available to you.

What can CP/M do? As explained in the last issue, CP/M is able to load and allow execution of user programs. It is able to handle all input and output (I/O) to any peripheral devices. Most important to a programmer, is the file management handling of CP/M. That is all CP/M can do, but we will see in a short while, how powerful these three functions actually are.

In order for CP/M to accomplish these three functions, it has been divided into four subsections. The segments consist of the CCP, the BIOS, the BDOS, and the TPA.

Console Command Processor

The Console Command Processor (CCP) is the portion of CP/M which is directly used by the operator of CP/M. This is what you will see on the terminal as you use CP/M. The CCP controls the interface between the operator and the CPU through the "command" mode of operation.

The commands of the CCP are of two levels: the built-in commands and the transient commands. Both levels are used to do the "housekeeping" functions of CP/M, which perform the creating, handling, listing, deleting, loading and running of any programs or files. The built-in or resident commands are permanently part of the CCP and may be accessed at any time from the command mode. The transient commands are those commands that are stored on disk and must be loaded from the disk and executed. The transient commands will be discussed at a later time.

The five built-in or resident commands are:

DIR	List the file names in a directory
TYPE	Type the contents of a file
ERA	Erase or delete a specified file
SAVE	Save what is in memory to a file
REN	Rename a specified file.

These resident commands are interpreted and executed immediately by the Console Command Processor when entered on the terminal.

Also available through the CCP, are the line editing functions it allows as input, while typing command lines. The editing options are executed by certain control-key sequences. These line editing functions are defined in the CP/M manual and will not be listed here.

Basic Input/Output System

The Basic Input/Output System (BIOS) handles all the input/output of the peripheral devices. Thus the BIOS is the only subsection of CP/M that is machine-dependent.

Accessing of the disk drives and other standard peripherals are the operations that the BIOS provides. The BIOS is the portion of CP/M that can be patched for any particular hardware environment. This is the advantage that was mentioned last issue; the ability for CP/M to be adapted to most any micro-system.

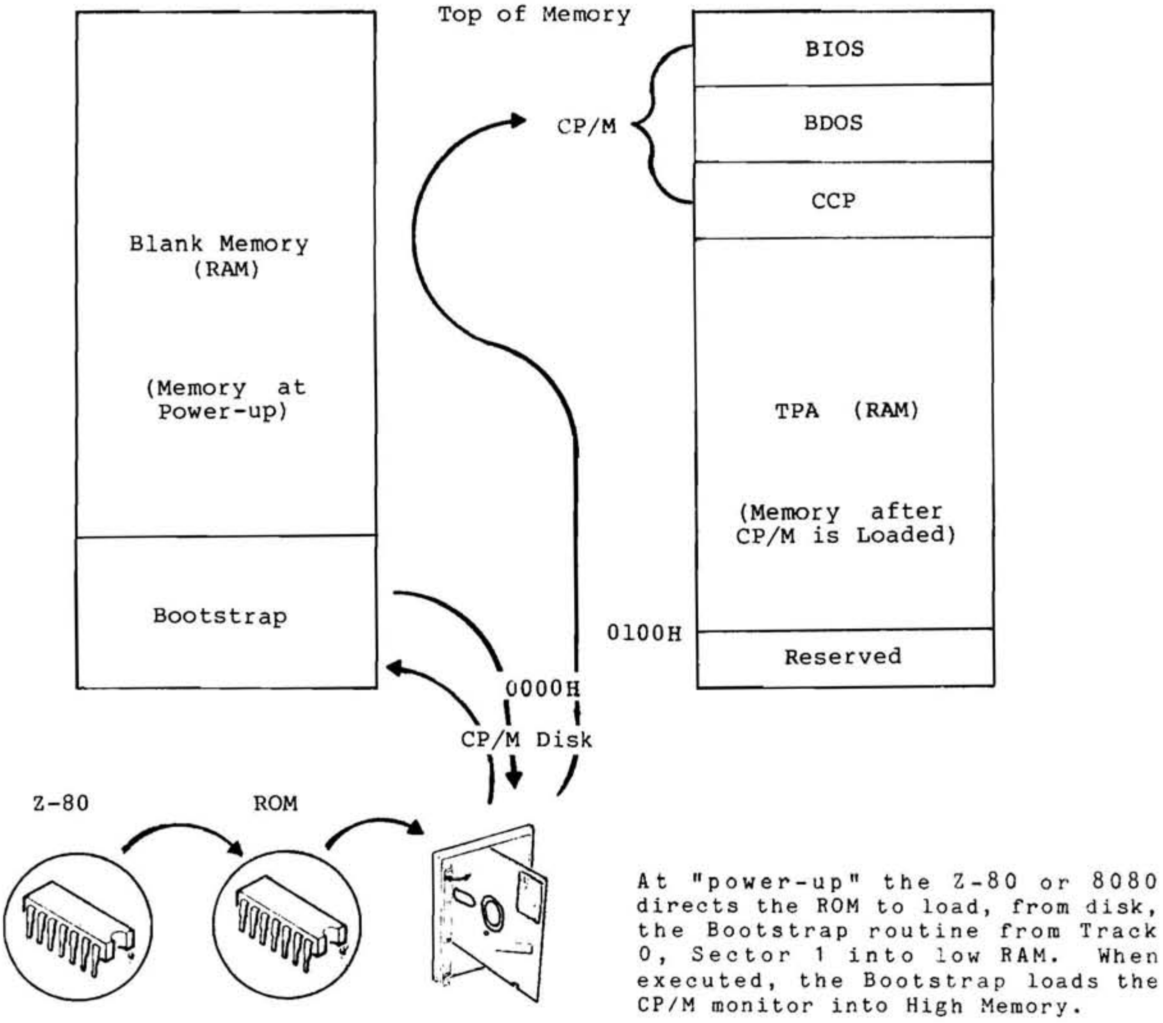
When you purchase the CP/M software package from Heath/Zenith, you will find that the BIOS has already been modified to operate on your Heath computer. No change is necessary. However, should you need to make any patches on your own, the CP/M package provides the standard BIOS and gives steps to modifying the BIOS.

In a future issue of REMark, maybe we will be able to examine the BIOS more closely and make a few modifications. As for now, it is good to know that all necessary changes have been made to the BIOS to operate on your Heath system.

Basic Disk Operating System

The Basic Disk Operating System (BDOS), sometimes referred to as the heart of CP/M, is the file management controller. The BDOS operations are completely independent and unseen by the user and/or programmer. Thus the programmer is free to deal with the pertinent matters of his duties.

The BDOS handles the basic disk file operations, such as the reading and writing of a record to/from a disk. It controls the "dynamic" allocation of the disk file construction, i.e. the BDOS will store a file in any available location(s) on the disk and remember where the entire file has been placed, for retrieval upon request.



At "power-up" the Z-80 or 8080 directs the ROM to load, from disk, the Bootstrap routine from Track 0, Sector 1 into low RAM. When executed, the Bootstrap loads the CP/M monitor into High Memory.

Diagram I

Transient Program Area

The Transient Program Area (TPA) is the segment that is not directly part of CP/M. The TPA is a term used to identify the area in memory, where user programs are stored and executed under CP/M. Any program loaded by CP/M is located at a starting address of 0100H and is stored upward in memory, filling bytes until the program is loaded or the system is out of memory.

With CP/M, when a program is loaded into the TPA, the CCP can be overwritten by the TPA, if the program is of sufficient size. A program will actually write right over the top of the CCP. It is even possible to overwrite the BDOS and BIOS, which means, a program can use virtually all memory for execution (with the exception of the addresses below 0100H). At the conclusion of the program, when control is to be returned to the CCP, it is necessary to reload the CCP by a "warm boot", if just the CCP has been overlapped. A "cold boot" must be done if the BIOS has been overwritten.

A general review, at this point, will show us that the CP/M monitor consists of the CCP, the BIOS, and the BDOS, which are located in high memory at bootup. The TPA is the memory area that user programs are loaded into and executed from. Please note the memory map layout for CP/M, see Diagram I.

Now it is time to move on with other basics of the CP/M operating system.

Bootup Procedure of CP/M

Most of you by now are familiar with the term "Boot". Just as HDOS has a "cold" and "warm" boot, so does CP/M. The bootup procedure is the same as for HDOS. The Monitor or ROM of the 8080 or Z80 (PAM-8 for the H8 and MTR-88 for the H89, respectively), directs the loading of the "bootstrap" program from off the disk into RAM. Then the "bootstrap" loader directs the loading, from disk, of the CP/M monitor into high memory, as indicated in Diagram I.

The process of loading the operating system into memory from powerup is referred to as, doing a "cold boot". When a program has finished and the command is to be transfered back to the CCP, a "warm boot" may be necessary if the program has over-written the CCP. (A "warm boot" may be executed at anytime by entering the control sequence, CTRL-C.)

As stated in our last issue, Heath/Zenith supports either the 5 1/4" or 8" disk versions of CP/M. Drives of either system are identified by the letters of the alphabet. For the CP/M system, the drives are identified by A:, B:, C:, D: and E:, depending on which drive, the 5" or 8", is the bootup drive. In order to bootup on your computer, you will need at least one disk drive. No matter which disk version of CP/M you have, the bootup drive will always be drive A:.

CP/M File Handling

When dealing with the CP/M file handling abilities of the CCP, we must become familiar with some of the definitions. Most of you are already familiar with the file handling of HDOS, CP/M is virtually identical. However, some of us have never been confronted by the terms before.

Filenames of CP/M are referenced by an eight character "filename" and its three character extension or "file type". The filename is generally a brief description of the file it is naming, making it easily recogizable. The file extension in CP/M is used to aid the user and/or programmer in avoiding confusion when identifying the file type. Some of the common CP/M file extensions are:

.ASM	Assembly source code	.BAS	BASIC source code
.COB	COBOL source code	.DAT	ASCII Data file
.FOR	FORTRAN source code	.SUB	SUBMIT command file
.$$$	ED or PIP temporary file		

These file types or extensions are suggested as a standard guide for naming all files. By maintaining the "filename" as a general description and by staying with the standard "file type" format, distinguishing files will be an easy task for yourself and other users.

Unambiguous filenames: Filenames that reference one and only one CP/M file.

Examples: STAT.COM, TEST.ASM, PIP.ASM, ANYTHING.BAS

Wild cards: The characters "*" and "?", which are used to match any character in a particular location of a filename.

Ambiguous filenames: Filenames that reference, by use of wild cards, more than one CP/M file.

Examples: T??T.ASM, ANY?????.BAS, *.BAS, TEST.*

The use of ambiguous files are for directory search and pattern matching, which allows you to reference similiar type files or programs in one command.

Rapping this Up

In the last issue of REMark, I stated, that in this issue, we would explain the set up procedure for CONFIGURing CP/M to your computer. In my continued study of CP/M, I felt that the basic facts and functions shown in this issue needed to be dealt with first. It appears from my vantage point, that we should be able to begin using CP/M in the next issue.

Vectored to Page 29

Pascal Corner — Part 1

by Henry E. Fale
2918 S. 7th. St.
Sheboygan, WI 53081
(414) 452-4172

Beginning with this month, there will be a monthly feature in REMark dealing with PASCAL. Why Pascal? Because Pascal is the up and coming language of the future, and it has lot of advantages over other languages like BASIC or assembly. Pascal is becoming the standard high level language of the computer industry. It is one of the least ambiguous programming languages you can find. It contains the best of BASIC, FORTRAN, and COBOL, but most important, it is easy to follow and is self-documenting.

Pascal unlike BASIC, is a structured language, requiring the programmer to define the nature, type and range of items in an exacting and precise way. This means you must think the program through before sitting down to write it, and define the variables, constants, procedures, and program segments.

Pascal is easier to learn as your first language, than if you are accustomed to BASIC, since in the later case, you must do alot of unlearning--but the rewards will pay for the difficulties. Pascal forces the programmer to develop good programming habits. The greatest benefit of Pascal is the ease in programming and debugging, since the program is broken down into many small modules which can easily be programmed, modified, and understood by yourself later, or by others. With a BASIC program, as time goes by, if not heavily documented by REMARK statements, you're likely to forget why you did something, and others trying to follow or modify your program are more likely to have problems.

There are more obvious advantages to Pascal. Since it is a compiled program, it executes faster and in less memory space. In contrast a BASIC interpreter must always be resident in memory and every instruction line is again and again decoded every time it is encountered by the interpreter. This means slower execution time, and more memory is consumed--something we can do without. Pascal is easier to program than assembly language, and is only a little slower than the assembled machine code, thus it has the advantages of BASIC (ease of programming) and assembly (fast execution with minimum memory overhead).

Pascal was developed to be fairly transferable from machine to machine. This holds true to some extent. There have been so many Pascal's developed, this is slowly disappearing. Although the most common Pascal in use is the UCSD version (the one Heath sells) I will be using the Lucidata Pascal for my examples here. I feel it is a better buy for many reasons. First, it runs under HDOS, which means all HDOS handling applies, it can be generated using a standard text editor, and since it supports HDOS, the 400K drives can be used for a super system. My configuration for this Pascal is an H89 with a Tandon 400K drive and a standard drive. This is a very flexible and powerful system. If you want more information on this, write me. Anyway, this Pascal is the ideal Pascal, because it is available for $95 in single quantities, or $65 in lots of 5 or more, ideal for HUG or group purchases. At this price, it is very powerful, and supports features not found in other Pascals such as HDOS and random file I/O.

If anyone wants to follow along and does not have Pascal, this is the one I recommend. It is available from Polybytes, 325 19th. ST. S.E., Cedar Rapids, IA 52403. If you already have the UCSD or other Pascal, you should still be able to follow along with the examples and text without much problem.

Before I start getting into Pascal, I want to tell you a little about myself. I am by no means a Pascal expert, in fact, I'm just beginning like alot of you. So why am I writing this column? Because someone has to start the ball rolling! I've wanted to see this happen for a long time, and it has not, so I figured I'd do it myself, and research and learn as I write. If anyone wants to take my place and has more experience, fine, write me and I'll step down. I know many of you could probably do a better job than me, but thus far no one has volunteered. Until someone wants the job, I'll be writing it.

Now about the workings of Pascal. It is a P-code system. The P-8080 compiler is a pascal program which is executed by the run-time system. To create a program, you generate the program text using a text editor (I'm using PIE), with a .PAS extension. When this is all done, you compile it and TYPE error checking is done. This is compiled in a .BIN extension, and is then run with the P-code compiler. This means that unlike an assembly language program which generates a runable machine code, this generates a code that must be run with a P-code interpreter. This takes the instruction

Vectored to Page 29

New HUG Software

885-1213 HUG CP/M DISK UTILITIES $20.00

This disk contains utilities to help you catalogue, test, identify, and patch your CP/M disks. It has the following programs.

SDUMP -- This is an expanded version of the DUMP program that was released on HUG disk 885-1201. It has been modified to run under Heath/Zenith CP/M, and to work on any size and density of CP/M floppy disk. It allows you to dump (in hex and ASCII) any file on the disk, and any sector, track, or group on the disk. You can also patch any sector. It can display the directory or the group allocation map on a disk, and can perform a non-destructive media check to locate bad sectors. A HELP command displays the available options while you are running SDUMP. Note: This program was written with Digital Research's MAC macro assembler (available from Heath), which you will need if you wish to re-assemble it.

CAT -- This is an alphabetizing disk directory for CP/M. It can display up to 66 files on the terminal without scrolling. Files are displayed in one to three columns and are alphabetized by column instead of by row for easier reading. CAT displays file attributes, user numbers, and the size in K for each file. The number of files and free space is shown, and if an ID number and label exist (see DISKID, below), they are displayed.

DISKID -- This program creates a disk ID number and label and writes them to the specified disk in a file called IDENT.SYS with the system attribute set.

LAB -- With this program you can display the disk ID number and label as created by DISKID. If you CONFIGUR CP/M to run LAB on cold boot, it will print the label information when you boot.

These programs require standard CP/M version 2.0 or higher and 32k of RAM. SDUMP and CAT must be re-assembled to run on any version of CP/M other than Heath/Zenith 2.2.02. Instructions are included in the source.

885-1090 MISC. HDOS UTILITIES $20.00

This disk is a collection of utilities for HDOS. It contains the following:

CCAT -- This is the HDOS version of the CAT program on 885-1213. It is an alphabetizing disk directory that can display up to 66 files on your terminal without scrolling. Files are alphabetized by column for easier reading. The size in sectors and flags of each file are shown. The program also displays the disk volume number, label, the number of files, and free sectors. Three switches are provided to control the display. They are /S to show system files, /A to show allocated sectors, and /P, which causes the directory to go to LP:.

HPLINK -- This is an HDOS version of the MPLINK program on HUG disk 885-1212. It is a modem communication program that provides for file transmit and receive, automatic log-on, and optional XON recognition. It uses the function keys and 25th line, and maintains a buffer free space indicator.

MBSORT -- This is a Schell-Metzner sort program designed to be called as a USR subroutine from MBASIC for fast sorting. Two versions are provided for general or keyed sorting.

RELOC -- This program can be appended to other programs to create a program that can be loaded anywhere in memory. It is used with MBSORT to put it in high memory, above MBASIC and your programs.

ENABLE -- If you have an H8 with the Extended Configuration Option, you can use this program to enable the low RAM in your system and use it to store USR programs for MBASIC, etc. Not for use with H89's.

AH -- With this program you can transfer programs from HDOS to Autoscribe and from Autoscribe to HDOS. When you go from HDOS to Autoscribe, you can instruct AH to force an End of Line (<) for each HDOS New-line character, or you can tell it to leave them off so Autoscribe can justify the file. In this mode, AH will insert End of Line characters when there are skipped lines in your text, to preserve paragraph separation. AH creates a new entry in the Autoscribe directory for files transferred from HDOS, counts the characters in the file, and places the name, count, and current HDOS date in the directory.

885-1106 MORSE 89 (H8/H19 or H89) $ 20.00

Send and receive Morse code on your Heath H89 computer. Morse 89 is intended to facilitate communication by Morse code over a wide range of code speeds, dot/dash ratios, signal strengths and noise conditions. The program is a modification to P/N 885-1052 adapted for use on the H89 (or H8/H19) including these added features: all operating control functions from special control keys; split screen operation with R/T data displayed on upper 2/3 of screen, two pretype buffers, and stored message buffer displayed on lower 1/3 of screen. Terminal runs at full 9600 baud allowing pretype or message buffer load to be accomplished and displayed on a character-by-character basis during and without missing receiver data; call from and exit to HDOS without any other operations; selection of manual, semi or fully automatic receive/transmit mode switch functions; continuous status display on Line 25 of: receive and transmit mode switch modes, receive mode (lock, track, hold) received speed WPM, transmit speed WPM, number of characters in pretype/type-a-head and message buffers, message buffer status display. The source (.ASM) is included.

Requires station interface hardware. (See "Adapting Morse 8 to H89", page 18 of this issue of REMark for design data and construction suggestions.)

HUG Product List

Part Number	Description	Selling Price
CASSETTE SOFTWARE (H8 and H88)		
885-1008	Volume I Documentation and Program Listings (some for H11)	$ 9.00
885-1009	Tape I Cassette	$ 7.00
885-1012	Tape II BASIC Cassette	$ 9.00
885-1013	Volume II Documentation and Program Listings	$ 12.00
885-1014	Tape II ASM Cassette H8 Only	$ 9.00
885-1015	Volume III Documentation and Program Listings	$ 12.00
885-1026	Tape III Cassette	$ 9.00
885-1036	Tape IV Cassette	$ 9.00
885-1037	Volume IV Documentation and Program Listings	$ 12.00
885-1039	WISE on Cassette H8 Only	$ 9.00
885-1057	Tape V Cassette	$ 9.00
885-1058	Volume V Documentation and Program Listings	$ 12.00
HDOS SOFTWARE (H8/H17 or H89 -- 5-inch only)		
MISCELLANEOUS COLLECTIONS		
885-1024	Disk I H8/H89	$ 18.00
885-1032	Disk V H8/H89	$ 18.00
885-1044	Disk VI H8/H89	$ 18.00
885-1064	Disk IX H8/H89	$ 18.00
885-1066	Disk X H8/H89	$ 18.00
885-1069	Disk XIII Misc H8/H89	$ 18.00
GAMES		
885-1010	Adventure Disk H8/H89	$ 10.00
885-1029	Disk II Games 1 H8/H89	$ 18.00
885-1030	Disk III Games 2 H8/H89	$ 18.00
885-1031	Disk IV Music H8 Only	$ 23.00
885-1067	Disk XI Graphic Games .ABS and B H BASIC (H19/H89)	$ 18.00
885-1068	Graphic Games (H19/H89)	* $ 18.00
885-1088	Graphic Games (H19/H89)	* $ 20.00
885-1093	Dungeons and Dragons Game Requires H89 or H8/H19	* $ 20.00
885-1096	Action Games (H19/H89)	* $ 20.00
885-1103	Sea Battle Game (H19/H89)	$ 20.00
UTILITIES		
885-1019	Device Drivers (HDOS 1.6)	$ 10.00
885-1022	HUG Editor (ED) Disk H8/H89	$ 15.00
885-1025	Runoff Disk H8/H89	$ 35.00
885-1043	MODEM Heath to Heath H8/H89	$ 21.00
885-1050	M.C.S. Modem for H8/H89	$ 18.00
885-1060	Disk VII H8/H89 SUBMIT, CLIST, FDUMP, ABSDUMP, etc.	$ 18.00
885-1061	TMI Cassette to Disk H8 only	$ 18.00
885-1062	Disk VIII H8/H89 (2 disks) MEMTEST, DUP, DUMP, DSM	$ 25.00
885-1063	Floating Point Disk H8/H89	$ 18.00
885-1065	Fixed Point Package H8/H89	$ 18.00
885-1075	HDOS Support Package H8/H89	$ 60.00
885-1077	TXTCON/BASCON H8/H89	$ 18.00
885-1079	HDOS Page Editor	$ 25.00
885-1080	EDITX H8/H19/H89	$ 20.00
885-1082	Programs for Printers H8/H89	$ 20.00
885-1083	Disk XVI RECOVER, etc.	$ 20.00
885-1089	MACRO, CTOH, and misc Utilities	$ 20.00
885-1090	Misc. HDOS Utilities CCAT, HPLINK, AH, MBSORT, etc.	$ 20.00
885-1092	RDT Debugging Tool H8/H89	$ 30.00
885-1095	HUG SY: Device Driver HDOS 2.0	$ 30.00
885-1098	H8/HA-8-3 Color .ABS/.ASM	$ 20.00
885-1099	H8/HA-8-3 Color in Tiny Pascal	$ 20.00
PROGRAMMING LANGUAGES		
885-1038	WISE on Disk H8/H89	$ 18.00
885-1042	PILOT H8/H89	$ 19.00
885-1059	FOCAL-8 H8/H89	$ 25.00
885-1078	HDOS Z80 Assembler	$ 25.00
885-1085	PILOT Documentation	$ 9.00
885-1086	Tiny Pascal H8/H89	$ 20.00
885-1094	HUG Fig-Forth H8/H89 2 Disks	$ 40.00
BUSINESS, FINANCE AND EDUCATION		
885-1047	Stocks H8/H89	$ 18.00
885-1048	Personal Account H8/H89	$ 18.00
885-1049	Income Tax Records H8/H89	$ 18.00
885-1051	Payroll H8/H89	$ 50.00
885-1055	Inventory H8/H89	* $ 30.00
885-1056	Mail List H8/H89	* $ 30.00
885-1070	Disk XIV Home Finance H8/H89	$ 18.00
885-1071	SmBusPkg III 3 Disks H8/H19 or H89	* $ 75.00
885-1091	Grade and Score Keeping	* $ 30.00
885-1097	Educational Quiz Disk H89 or H8/H19	* $ 20.00
AMATEUR RADIO		
885-1023	RTTY Disk H8 Only	$ 22.00
883-1106	Morse-89 H8/H19 or H89	$ 20.00

* Means MBASIC is required

Vectored to Page 31

Adapting the MORSE-8 CW Program for Use with the H89

By: Robert R. Anderson, K2BJG
69 Page Drive
Oakland, NJ 07436

The HUG MORSE 8 program 885-1052 for the H-8 can be adapted for use with the H-89 with only minor software changes. Hardware interface is approximately the same differing only in the computer connection points, and a reduction of some circuitry. All operating features are retained except the transmit and receive speed readout,and other data displayed on the H-8 front panel. This information and other data can be displayed on the 25th line on the H-89 or H-8 using the H-19 terminal with further software changes.

This article describes the minimum required,and some recommended software changes,and details of hardware interface to use the H-89 as a terminal to send and receive morse code. Except for the actual point of connection to the computer the interface data presented is equally applicable to the H-8 as well as the H-89, and is much more comprehensive than the interface data supplied with the HUG 885-1052 program.

With the program running during CW transmit mode the computer produces the CW signal at D7 of a parallel port as an off/on dc keying signal. In the H-8 this signal is the signal that turns audio to the front panel speaker (Horn) on and off allowing the operator to audibly monitor the CW transmit output. The station transmitter keying signal is derived from this CW audio signal via an external optical isolator and rectifier/filter circuit to restore the dc off/on CW keying signal. The H-89 does not have this internal audio keying circuitry or speaker. The dc keying signal may be used directly without the need for external dc restoration circuitry. Audible monitoring can be accomplished by an external keyed audio oscillator if desired for off the air code practice or transmit monitoring. It should be noted that this function is provided for transmit monitoring in most of commercial amateur transmitter equipment in use today.

The dc CW keying output signal from the H-89 is TTL compatible, low-off (space), high-on (dot or dash) and is available at the general purpose port 360Q, D7.

With the program running during CW receive mode, the program requires the computer to receive a level 5 interrupt signal on the leading edge, and a level 6 interrupt signal on the trailing edge of each CW dot or dash. This is accomplished by an external interrupt generator consisting of one 74LS123 dual one-shot multivibrator. The "A" section generates the "ON" interrupt pulse signal being connected to trigger on the leading edge of the dc CW keying input signal. Similarly the "B" section generates the "OFF" interrupt pulse signal being connected to trigger on the trailing edge of the input keying signal. The not Q outputs are connected to the computer interrupt decoder via diodes. This allows other computer interrupt signals to operate on levels 5 and 6 terminals if required. The timing resistor and capacitor for both multivibrators are set to provide pulses about 3ms wide. Limits are not less than 2ms or more than 4ms.

Figure 1 shows all circuitry required to interface the computer to transmit and receive CW keying lines at TTL compatible levels. Input and output CW keying line polarity is HI for off (key up) and LO for on (key down). If interface to station equipment requires the opposite keying polarity for receive, reverse interrupt connections to the computer and replace R1 with a 330 ohm resistor connecting it to ground instead of to +5 Vdc. For transmit connect directly deleting R2 and Q1.

COMPUTER CONNECTIONS:

SIGNAL	H-89
ON INTERRUPT	U-557 PIN 2
OFF INTERRUPT	U-557 PIN 3
OUTPUT	U-552 PIN 19
SIGNAL GROUND	U-557 PIN 8

**USE SEPARATE GROUND CONNECTION TO THE REAR PANEL FOR PROTECTIVE GROUND.

Physical wiring within the H-89 to make connections can be accomplished in several ways. The choice depends on the owner's attitude concerning soldering directly to the CPU board, and the other accessory boards installed. As a means of mounting a suitable rear panel connector without adding any holes, the author used a 6 pin male MOLEX type 1261 (Radio Shack 274-226) mounted in the opening marked 488.

For those who do not wish to solder directly to the CPU board, or to provide a convenient method of disconnection when removing the CPU board, use part or all of the following connection techniques.

For interrupt connections to U-557 solder a 16 pin DIP socket (Radio Shack 276-1988) directly on the top of a 16 pin DIP header (Radio Shack 276-1980), make connections at the joints of pins 2, 3, and 8. Remove U-557 from the CPU board, inserting it in the new socket mounted on the header. Insert the assembly into the U-557 socket on the CPU board.

The same technique can be used for the output connection to U-552 however this is a 20 pin DIP therefore the DIP header may not be readily available. An alternate does exist if the 3 port serial or cassette cards are not installed. This is a small female connector (available from Heath p/n 432-120) pushed on to P-510 or P-511 pin 17. This is port 362, D7, and is indicated on the schematic as I/O 1 H. If necessary a 20 pin DIP header can be made from a 16 pin and an 8 pin cut from a 16 pin. Solder to a 20 pin DIP socket and then use plastic cement. Once the IC is inserted no problem will exist.

The transmit and receive keying signals must be interfaced to whatever keying sources and destinations desired. For transmit, the CW keying signal from Q1 may be connected directly to key jacks of most modern solid state amateur transmitter equipment such as the Heath SB-104 that require pulling a positive low dc voltage to ground. Transmitter equipment of the tube type employing grid block keying such as the Heath HW-100 series or the older SB series that require pulling a negative high (up to approximately 100 volts) to ground will require interfacing with another transistor. This must be of the high voltage type. An optical isolator may be used. Several circuits of this type are shown in amateur publications such as the ARRL Radio Amateurs Handbook.

Stations equipped with electronic keyers such as the Heath HD-1410 may connect directly to the auxillary (hand key) jack of this unit. Transmitter interface for both positive and negative keying is provided by the Heath keyer. As mentioned previously, audible monitoring of the transmit CW signal is usually a function of transmitters or electronic keyers. If an external keyed audio oscillator is required, the circuit using the 555 IC shown on the schematic that comes with the HUG 885-1052 or any of this type shown in amateur publications is suitable. Note that this circuit requires a HI for on. Connect to a appropriate point.

For CW receive using the computer, a hand or electronic keyer may be connected directly to the interrupt generator for code practice. To receive CW signals off the air, hardware circuitry is required to convert the CW audio beatnote from the station receiver to a dc CW keying signal for input to the interrupt generators. Several designs of this type have been published in amateur publications. Ref 1, 2, and 3. The circuit that comes with the HUG 885-1052 is useable but does not include a noise burst eliminator which is a worthwhile feature.

One consideration is necessary in the design and construction of whatever circuit you select. All circuits of this type include an active bandpass filter followed by a PLL tone decoder. The operating frequency of these stages must match the receiver's audio beat note frequency when the receiver is tuned to properly center the RF signal in the receiver IF filter passband. For most Heath amateur receivers this is approximately 1000 Hz.

Figure 2 shows the receive audio to dc keying circuit used at the author's station. This is based on the circuit from ref 1 modified to delete audio output not required in this application. The active bandpass filter and PLL decoder with the component values indicated, is tuned to 1 khz. to match a Heath SB-303 receiver. The 10k feedback resistor shown produces a bandwidth of approximately 100 Hz. Use a 25k potentiometer or other fixed resistor to obtain the bandwidth desired.

The noise filter is very effective in eliminating false triggering by man made and atmospheric noise. Eight cycles of a valid signal are required to initiate the start of a dot or dash. This delay being very short as compared to a CW dot or dash, is not noticable.

Figure 3 shows a suggested wiring for interconnecting system components to route transmit and receive signals without switching. By the use of simple diode "or" circuits, signals are routed to their proper destination without interfering with other destinations. When using an electronic keyer such as the Heath 1410 or equivlent the blocks labeled xmt keyer and audio oscillator are part of that unit. It is only necessary to break the internal connection on the keyers PC board to obtain access to these circuits separately for connection of the "or" circuits. In the author's station, the keyer is an "accu-memory" electronic keyer. The interrupt generators, receiver audio to DC converter, and diode "or" circuits were built into the same box.

SOFTWARE CHANGES H-89

The following are minimum required source code changes required to make the HUG 885-1052 MORSE 8 program operate on the H-89. Enter "CODE.ASM" via EDIT, make these and other recommended changes described below and reassemble.

MINIMUM REQUIRED:

```
1. Before      ON         LXI    D,120320A     ON=120Q OFF=320Q
   After       ON         LXI    D,202002A     ON=202Q OFF=002Q

2. Before      OFF        MVI    D,320Q        320Q=OFF
   After       OFF        MVI    D,002Q        002Q=OFF

3. Before p/o off         MVI    E,320Q
   After  p/o off         MVI    E,002Q

4. Before p/o snd 1       OUT    360Q          TURN SPEAKER OFF
   After  p/o snd 1       OUT    362Q          TURN OUTPUT OFF

5. Before p/o snd 1       OUT    360Q          SPKR OFF
   After  p/o snd 1       OUT    362Q          OUTPUT OFF

6. Before p/o spcntl      CPI    356Q          CTL J
   After  p/o spcntl      CPI    356Q          CTL L

7. Before p/o spcntl      CPI    357Q          CTL I (TAB)
   After  p/o spcntl      CPI    357Q          CTL K

8. Before p/o spcntl      CPI    361Q          CTL/SHIFT P
   After  p/o spcntl      CPI    361Q          CTL @

9.(p/o brynt):

   Before line 4          DB     357Q,356Q,000Q
   After  line 4          DB     000Q,000Q,357Q
   Before line 5          DB     0,0,0
   After  line 5          DB     356Q,000Q,000Q
```

Steps 3 and 4 change the CW output port from the front panel port 360Q used in the H-8 to the general purpose port 362Q used in the H-89.

Steps 1 and 2 change the off/on control bit patterns from that required for port 360Q H-8 to that required for port 362Q H-89. Bit 7 is the CW on/off control bit in both cases. Since, control of the internal 2ms clock and single step interrupt is via these ports bits 6 and 4 used for these functions on the H-8 port 360Q must be changed to bits 1 and 0 port 362Q on the H-89. CAUTION -- DO NOT attempt to reverse polarity of the output signal from D7 by changing bit 7 in software. Doing so would cause the 2ms interrupt signal to be outputted between dot or dash "on" signals. Use external hardware to invert polarity if required.

Steps 6 thru 9 change control keys as follows:

```
        CTL J  to   CTL L
        CTL I  to   CTL K
        CTL P  to   CTL @
```

The requirement for these changes is the fact that CTRL J, I and P are processed by the "smart" terminal used in the H-89; therefore cannot be used for control functions. These changes are also required when using the H-19 terminal with the H-8. This problem was previously pointed out in REMark Issue 13 page 29.

RECOMMENDED SOFTWARE CHANGES:

```
1. Add after title  - FOR THE H-89

2. Before p/o spcntl  JE    QRL
   After  p/o spcntl  JE    QSL

3. Before      QRL
   After       QSL

4. Before p/o qrl     MVI   M,012Q    R
   After  p/o qsl     MVI   M,010Q    S

5. Before      CLL    MVI   016Q      W   (AND SUBSEQUENT AS REQUIRED)
   After       CLL    MVI   *         *   (AND SUBSEQUENT AS REQUIRED)
```

* Use table 1 to obtain proper brynt code for your call letters.

Steps 2 thru 4 change CW output Q code signal from QRL? to QSL?. QRL is not often used in amateur radio practice whereas QSL is used extensively by amateur CW contest operators. Step 5, and table 1 provide instructions to program your own amateur call letters for transmission by the stroke of only one key.

TABLE 1

Use to obtain brynt code for change of call letters.

0	077Q	A	006Q	N	005Q
1	076Q	B	021Q	O	017Q
2	074Q	C	025Q	P	026Q
3	070Q	D	011Q	Q	033Q
4	060Q	E	002Q	R	012Q
5	040Q	F	024Q	S	010Q
6	041Q	G	013Q	T	003Q
7	043Q	H	020Q	U	014Q
8	047Q	I	004Q	V	030Q
9	057Q	J	036Q	W	016Q
		K	015Q	X	031Q
		L	022Q	Y	035Q
		M	007Q	Z	023Q

CONCLUSION:

The hardware interface information presented is applicable to the HUG 885-1052 program for both the H-8 and H-89 computers. Specific connection information is provided for the H-89. The software changes presented are required to operate this program using the H-89. Some are required for the H-8 when using the H-19 terminal. The author is in the process of developing further changes to this program for the H-89. These include rx and tx speed readout and other status indications displayed on the 25th line as well as other operatingimprovements. This software project is on its way with the 25th line display operating. A new disk incorporating theseimprovements is planned to be submitted to HUG in the near future.

REFERENCES:

Ref 1	THE CW FILTER-LIMITER	QST	1/79 page 42
Ref 2	CW REGENERATOR	HAM RADIO	4/74 page 54
Ref 3	CW SIGNAL PROCESSOR	HAM RADIO	10/78 page 34

Editors Note: Bob Anderson notified us that there is a General Purpose CW Computer Interface, model MFJ-1200, available from MFJ Enterprizes Inc., Box 494, Mississippi St., MS 39762, for $69. He said that the only addition to the package is the interrupt generator that is shown in Figure 1.

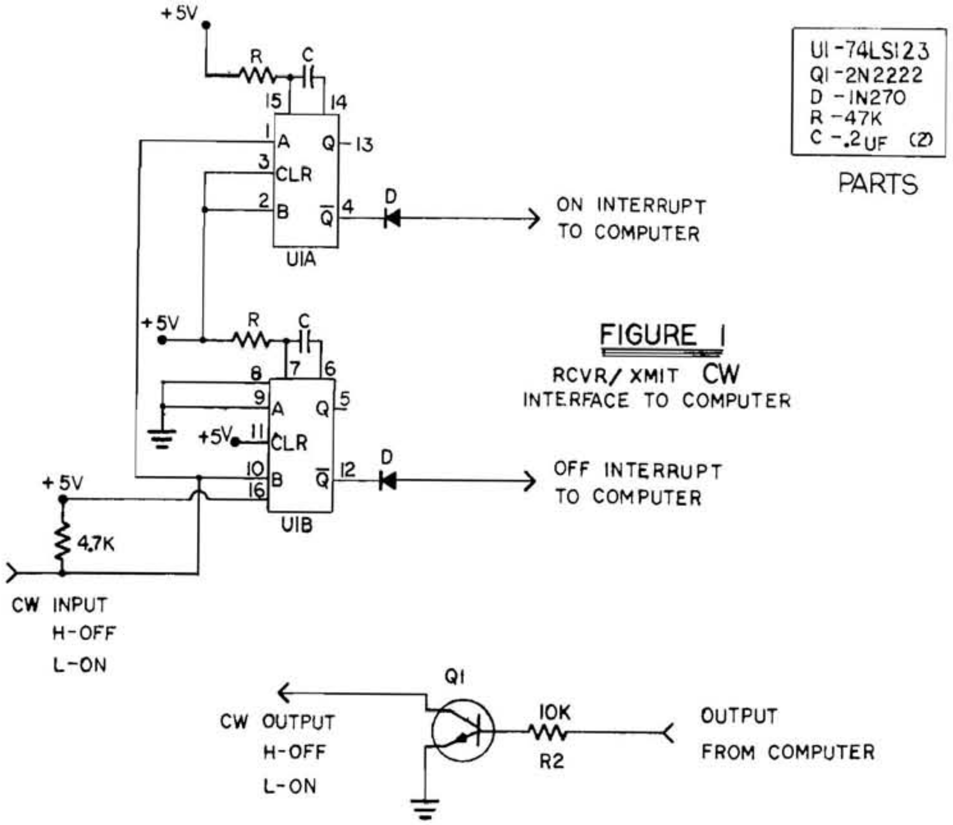


FIGURE 1

RCVR/XMIT CW INTERFACE TO COMPUTER

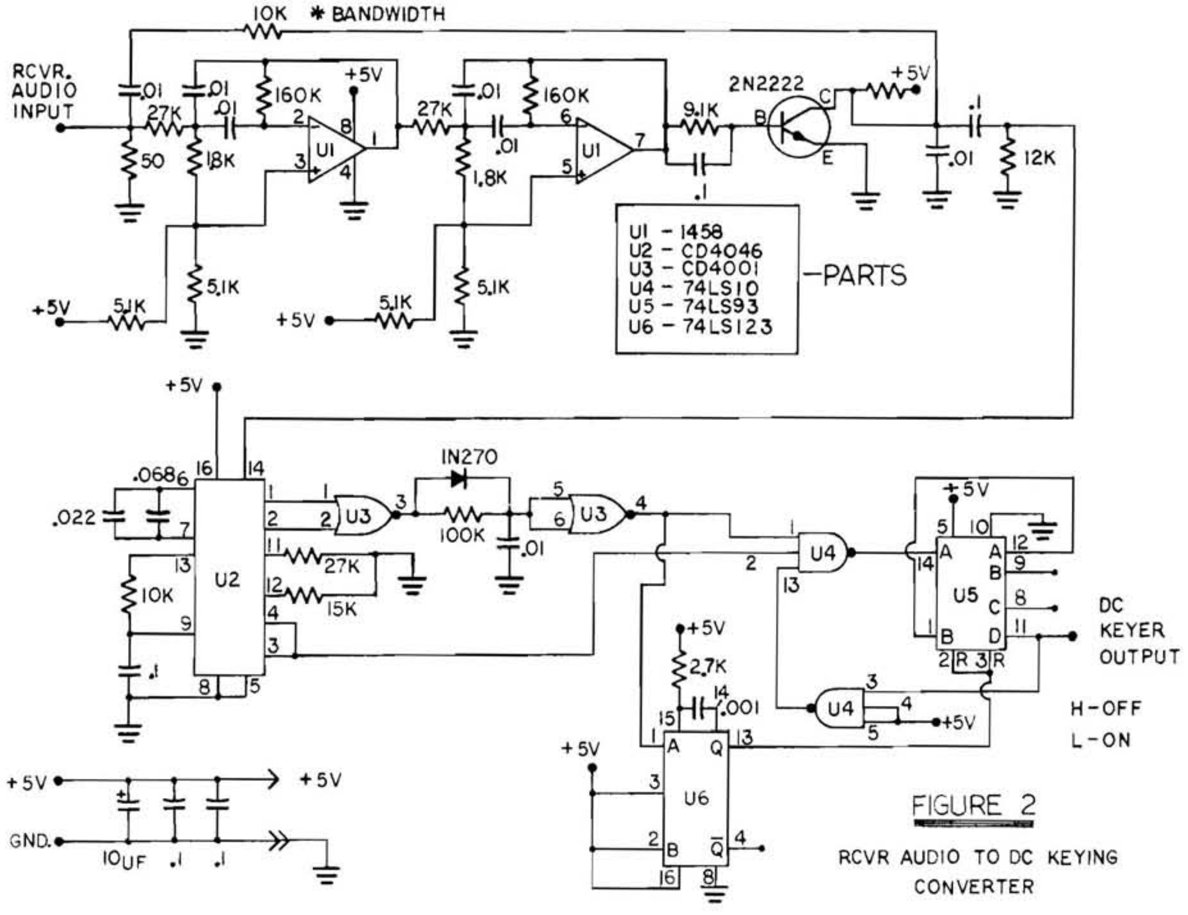


FIGURE 2

RCVR AUDIO TO DC KEYING CONVERTER

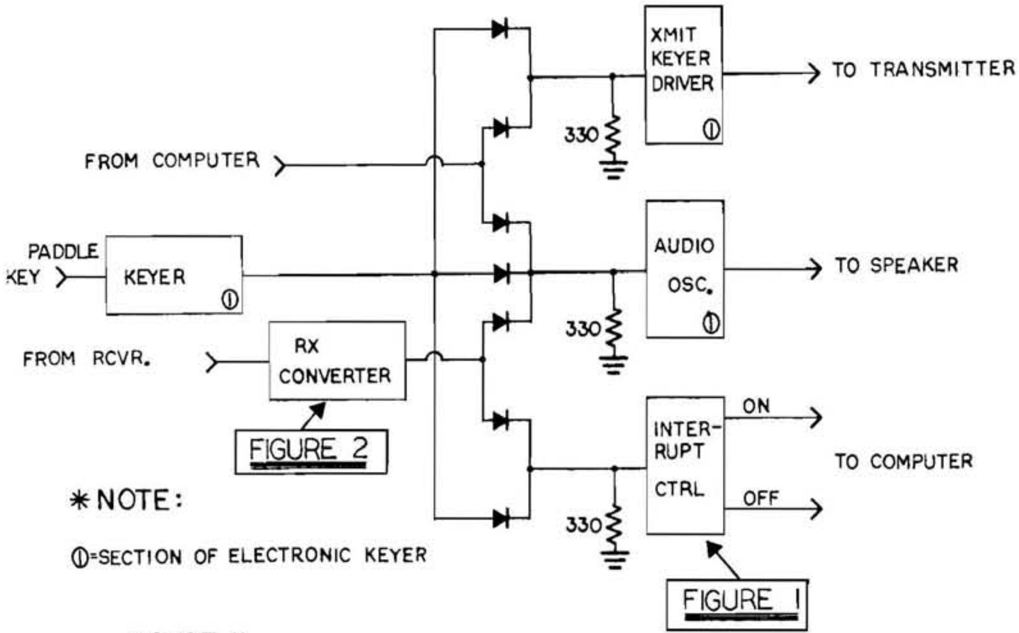


FIGURE 3

Corrections to 885-1091 Grading and Scorekeeping

The following changes should be made to both the EDIT10.BAS and EDIT4.BAS programs.

```
  20 DIM S$(40,1),S(40,15),W(40),V(15),NA(40),WT(15),T(15),TT(15)

 810 N1=N1+1:IF N1>15 THEN PRINT E1$Y$"'0Maximum of 15 scores!":@
           N1=15:GOSUB 2120:GOTO 100

 815 J=N1:WT(J)=1:V(J)=1:FOR I=1 TO N
```

Corrections or changes are underlined. Version numbers of the programs being changed should be changed to 09.29.81. The version number is located in line #10 of all programs.

SBPIII on 8-inch Drives

The Small Business Package III (885-1071) may be used on the H47, Z47 or Z67 eight inch drives by copying ALL files onto ONE eight inch disk and making three line changes.

The changes in the lines have been underlined and are all in the STARTUP.BAS program.

```
 410 DK$="SY0:":GOSUB 1000:GOSUB 940:GOSUB 720:GOTO 600

 710 OPEN"I",3,"COMPANY.LST":INPUT #3,YC$,YJ$,YA$,YS$:GOTO 790

1180 IF ERL=710 THEN DK$="SY0:":GOSUB 720:PRINT #3,NA:PRINT #3,SC:CLOSE:RESUME 710
```

The operation of the system will be the same except you must specify one drive on bootup.

Give a Voice to your ET-3400

A review of Heath's NEW
Voice Synthesis Course
By: John Hubbard

You've undoubtedly read a lot about speech synthesis. Perhaps you've even heard the VOTRAX TYPE-'N-TALK* or the Texas Instruments SPEAK and SPELL**. Now you can make your ET-3400 talk too! The new Heathkit/Zenith Educational Systems Voice Synthesis course, EE-3403, not only teaches you how voice synthesis circuits work, but also provides you with the chips you need to wire the two most popular types of speech synthesizers on your Microprocessor Trainer.

The purpose of this course is to teach you how modern electronics is used to simulate the functions of the human vocal tract. The first unit contains a brief but pertinent anatomy lesson and other background material. In the second unit, you learn the basics of speech synthesis and digital storage of analog signals. With the fundamental and background material out of the way, the third unit gets you into the practical, hands-on information. Using the Texas Instrument TMS5100 as an example, you learn the hardware and software requirements for a linear predictive coding (LPC) system. After you learn the theory, you are guided through four experiments with the LPC circuit. The next two units teach you about phoneme speech synthesis (PSS) systems. You will complete six experiments with the VOTRAX SC-01 chip that give you experience with the circuitry and programming for phoneme synthesizers.

When you finish the course, you will have wired and programmed both types of synthesizers. The LPC functions much like the one used in the SPEAK and SPELL**. It includes a voice processor and a vocabulary ROM. This "canned vocabulary" approach provides excellent quality speech with a minimum programming effort. The PSS uses the same processor chip as the TYPE-'N-TALK*. Using fractional parts of syllables called phonemes, you can create practically any word you want. This gives the phoneme speech synthesizer an "unlimited vocabulary". The speech synthesizer chips are actually special purpose microprocessors. In the first experiment you simply give commands to the voice processor chips a key stoke at a time. By the end of the course, you are storing phrases through the keyboard and playing them back with a single stroke for an entire phrase.

The experiments in this course use the Microprocessor Trainer's LED display as an output port to allow you to get all the circuitry on the existing breadboard. This simplifies the wiring and allows you to use the course even if your trainer has been modified for use with the ETA-3400 accessory unit. Although there is no power amplifier, and the experiments must be done when there isn't much background noise, the sound quality is excellent. It isn't difficult to connect the voice output to an audio system for a really big voice. It's so much fun to have a voice for your computer that you won't want to disconnect the circuit, but this spring there's an interfacing course that you won't want to miss. Among other things, it teaches you how to have your ET-3400 "burn" EPROMs for you.

* TYPE-'N-TALK is a registered trademark of Votrax division of Federal Screw Works.

** SPEAK and SPELL is a registered trademark of Texas Intruments, Inc.

EOF

H-19 TOUCH 'DEM KEYS

Charlie and Karen Kotan from out California way stopped by HUG the other day to see what was going on at the "trailer". During the normal conversations Karen mentioned that she didn't care too much for the H-19 keyboard 'cause it didn't have "locators" for the "D" and "K" keys. I had to ask the obvious: "Why!?" Karen explained that she only types at about 120 words per minute (hmmmmm?) and that it would sure help to have some reference to start from without having to look at the keyboard.

Charlie explained that he had modified the H-19 by placing a drop of Superglue on the D and K. He said the drop turned white for a time, but soon disappeared with normal wear. Further, Karen can now "feel" the keys and is able to resume her normal lightening pace using the H-19.

I thought that some of the other speedy people out there might appreciate this suggestion. Me? I still peek and haven't exceeded several pokes a minute yet!

BE:

BUGGIN' HUG

Dear HUG,

I am writing this letter to pass on to other readers of REMark, my experience with the Epson MX80 printer. I have read several articles in Heath newsletter type publications, some with partially correct information. I have also received information and mis-information over the phone from Heathkit stores and from Benton Harbor. What follows is the interfacing and software that works for me.

My system consists of the H-8 computer, the H-19 terminal and the WH-8-4 serial board. I purchased an Epson MX80 printer with their 8141 serial interface. Epson also offers a 8150 buffered serial interface which is necessary for use with the bit map graphics.

For interfacing you need a male DB25 connector coming from the WH-8-4 board. It can be wired pin for pin with the Heath Configuration but you must move the connection from Heath pin #7 at DB25 pin #4 to DB25 pin #20. This change will provide the Data Termial Ready signal.

If you are making your own connector, you actually only need four of the connections. These four connections are: Heath pin #1 to DB25 pin #1, Heath pin #5 to DB25 pin #3, Heath pin #13 to DB25 pin #7, and Heath pin #7 to DB25 pin #20. This completes the hardware interface.

In the Epson printer there is a control circuit board with two dip switches. You do not have to change the settings of either from the factory settings, which are: SW1 -- 1-on, 2-on, 3-on, 4-off, 5-on, 6-on, 7-off and 8-on; SW2 -- 1-on, 2-on, 3-off and 4-off. These settings work fine on the Heath system.

The serial interface board has one dip switch which should be set as follows: 1-baud, 2-off, 3-baud, 4-baud, 5-on or off, 6-off, 7-baud and 8-n/a. The baud rate is selected from the baud rate selection chart. The last four baud rates are duplicates and should not be used. As an example, I am running at 4800 baud to match software written for the H-14 printer and have SW1-on, SW3-on, SW4-off and SW7-off.

The interface board also has a JNOR and a JREV wire jumpers. The printer comes from the factory in the JNOR position and needs to be changed to the JREV position. The jumper wire is soldered in postion and it is a simple job to cut or unsolder the JNOR. Solder a new jumper wire in the JREV position.

The device driver caused me some problems at first as I was using what was "obviously" the correct one ... ATH84 ... WRONG! Either the LPH14 or, better yet, the LPH24 from the "Software Tools" disk are the proper choices. The ATH84 driver does not send or accept signals for Data Terminal Ready and you end up with garbage on the printer.

After a lot of frustration and time, my Epson MX80 is up and running. I feel that I got a lot of printer for my money and recommend it to anyone looking for a printer. It has a lot of features not found on printers for twice the price.

Louis Berry
2259 N. Marter Ct.
Simi Valley, CA 93065

Dear Lou,

We have heard that Heath is working on a special .DVD just for the MX80 with documented interfacing information. This will truly help with the MX80 problems experienced in the past.

Uncle HUG

Dear HUG,

Thanks to Gene Sevin for his Menu-Driven Demo Program in Issue 19 of REMark. The following changes will allow it to be run in an H-88 using 10.06.00 Extended Benton Harbor BASIC.

```
290 REM -- NO NEED FOR FN P( )--
300 Z=8280


4000 REM
4010 REM *** FUNCTION KEY INTERPRETER
4020 REM
4030 POKE Z,0 :REM CLEAR INCORRECT KEY
4040 IF PEEK(Z)=0 THEN 4040
4050 IF PEEK(Z+1)[|27 THEN S=1:GOTO 4090
4060 IF PEEK(Z+2)=82 THEN S=2:GOTO 4090
4070 IF PEEK(Z+2)=83 THEN S=3:GOTO 4090
4080 GOTO 4030
4090 POKE Z,0:RETURN:REM CLEAR FOR 'Y'
```

This also takes care of the stray characters.

Allen Zimmer
Eagle, NE 68347

Dear HUG,

PIN can be used to eliminate the necessity of a CR (Carriage Return) during input functions when only a single character is needed (e.t. an "N" or a "Y" for Yes and No). However, since the receiver buffer of the 8250 USART only changes data when it is overwritten, subsequent PINs of the port receive the same data as the original PIN. In other words, if you answer your first question with a "YES", your second question answers "YES" automatically. This presents no problem if you are alternating between questions that require different data, but if not, the following routine will solve the situation. (This routine is designed for the H8-4 Multi-Port Serial Card with the console set to 350 octal.)

```
 10 REM *** INKEY ROUTINE
 20 REM *** DISABLE CONSOLE INTERRUPTS
 30 OUT 233,0
 40 REM *** CHECK RECEIVER STATUS
 50 IF PIN(237)[|97 THEN GOTO 50
 60 REM *** GET THE DATA
 70 D=PIN(232)
 80 REM *** ENABLE CONSOLE INTERRUPTS
 90 OUT 233,1
100 RETURN
```

To convert the data to an ASCII value, subtract 128. This routine ignores the MLI and MLO console sets. Therefore, if you want upper case ASCII values, subtract 160 or, if you want to convert to numeric values directly, subtract 176. I have run into one difficulty using a "LINE INPUT" or an "INPUT" statement after using the routine described. Both the LINE INPUT and INPUT assume that if the data at the port in not a CR, it is part of the input data. This situation can be overcome by using LINE INPUT then adjusting the string as follows:

```
D$=MID$(D$,2,LEN(D$)-1)
```

or

```
D$=RIGHT$(D$,LEN(D$)-1)
```

One of the biggest advantages I find to routine is that you can qualify the input before you accept it. If the input is supposed to be a number, then the addition of the following line will reject all inputs except numbers and will return with D= to the inputted value:

```
75 D=D-176:IF D[0 OR D|9 THEN GOTO 50
```

William Stackpole
Port Orchard, WA 98366

BB's Via MNET

For the past year we have been mentioning the HUG Bulletin Board system on MicroNET. We have, in bits and pieces, given advice, hints and directions of the what, how and where's of the HUGBB. We are going to give just a short review at the close of this article . . but first, I need to explain about the other BB's that pertain, and are available, to you on MNET.

1) H11 users have long been on their own when it comes to software, hardware and "anyware" with their systems. Well, there is available on the MicroNET system the MNET11 Special Interest Group (SIG), for any owners of the H11 or DEC PDP-11 computers.

The MNET11 SIG or Bulletin Board, is run by Chuck Sadoian and is free to all who have an interest in the H11 or DEC PDP-11 type computers. They have an on-line data base of 2000 blocks of software that run on HT-11 and/or RT-11 operating systems. They have established a software exchange via the US mail for larger programs, including many from DECUS.

Chuck has submitted a modem package, to HUG, for the H11. The program will be released in next month's issue of REMark. . . watch for it!! If you need it sooner, contact Chuck.

For any serious H11 owner, the MNET11 SIG is a must and the place to be!! For help, advice or software, the entire SIG is there to assist you in finding a solution to any problem or for simply bringing you up to date as to the availability of H11 software.

For detailed information on H11 interests, you may feel free to contact Chuck at

Chuck Sadoian
PO Box 397
Dinuba, CA 93618
(209) 591-2631 (work)
(209) 591-6849 (home)

2) Now that we have begun the CP/M series, it should be of interest to many of you that there is a CP/M SIG (CP-MIG) on MNET.

They have over two megabytes of on-line CP/M programs, be it public domain software or user submitted programs.

Many of the programs are oriented to Heath systems. (Also of interest, the CP-MIG now is distributing the MicroNET Executive.) There are no restrictions or fees for becoming a member of the CP-MIG. Many of the HUGBB members access the CP-MIG.

The CP-MIG has three SYSOPs: Charles Strom in NY, David Kozinn in CA, and Tom Jorgenson, our good friend and HUG supporter, in St. Louis. Anyone, familiar with Tom and his contributions to HDOS or CP/M, know what high caliber the CP-MIG must be. There is really nothing more that needs to be said about the CP-MIG.

Now for a short review of how to access the Special Interest Groups on MNET, we will make reference to each of the MNET11, CP-MIG and HUG SIGs. Each of the SIGs (or Bulletin Boards) are data management programs that were created by CompuServe Inc.

To access the SIGs, you need to get an account with CompuServe. At present you may go to a Radio Shack dealer to acquire a User ID. Yes, there have been many problems with the Radio Shack managers not willing to deal with Heath users. I was told by Tandy, that the managers MUST order P/N 26-2224 for you for $19.95!!!!! Not $29.95!! Also, one of the local HUG groups (SMUGH) has been sending in the CompuServe Application form, bypassing Tandy, and has been getting a quick response, receiving their User ID's within a week and a half. This may be an option you may try if you have access to the application forms.

The two terms, MicroNET and CompuServe, for all practical purposes, can be used synonymously. CompuServe is a time share computer system, that allows many users to "talk" to the "Host" computer system, at the same time. They have provided the SIGs with a means of using their system to "chat" about things that relate to the respective SIG.

To access a SIG from CompuServe, you enter "MIC" at the main menu. This will take you to the MNET prompt "OK". Now enter "R SIGS(group name)" and you will drop through to the SIG of your choice, e.g. "R SIGS(MNET11)" will take you to the MNET11 SIG, "R SIGS(CP-MIG)" to CP-MIG, and simply "R HUG" to the HUG SIG.

To be added to the MNET11 or CP-MIG, simply leave a message to the SYSOP, asking to be included. For the HUGBB, you must leave a message to SYSOP, that's me, telling me you are there. You must include your HUG membership number, because only HUG members are allowed to access the HUGBB, while anyone may access the MNET11 or CP-MIG. If there is no HUG membership number, I cannot add you.

Once the SYSOP has added you to the membership base, you will be added to the respective SIG. At that point, you are on your own to read the many instruction files that are available for you to help you become more familiar with the SIG. The SYSOPs are there to help you if you need it. The best advice I can give to a new member of any SIG, is to practice with the many options to acquaint yourself with the structure of the SIG. Each SIG has slightly different options available to the user. This is done because each SIG serves a different purpose and therefore the options need to be used in the most advantageous way.

The Special Interest Groups are there for your benefit. If you have the opportunity to set up your system with a modem, remember the SIGs are at your service.

*SYSOP <TLJ>

DMM's Are Coming

This issue of REMark was to have an indepth look and review of Data Base Management Systems. However, due to tight scheduling, the review is being heldover for the December issue.

The review will define Data Base Management Systems. An explanation will be made as to what to look for in a DBM. The issue will be used to announce the available DBM's that are to be released from HUG.

If you are not sure what DBM's might mean for you or for those of you who are in need of a data management system, watch for next months issue of REMark.

The All New H25/Z25 Line Printer

Heath/Zenith has finally come up with a high-quality, high-speed dot matrix printer at an affordable price of only $1095 in kit form (H25) and $1595 fully assembled (Z25).

The Heath Users' Group was loaned a new H25 printer in mid September to prepare some demo software. Here is a brief description of the events surrounding the setup and operation of the H25 line printer.

The first noteable item was that there were NO special hookups required. Simply plug the H25 cable into the H8-4 four port serial I/O or the HA-88-3 three port serial I/O. No interrupts or special jumpers were needed for the hookup.

To load the paper into the printer the paper is fed into the printer from one of three directions; back, bottom or front. A paper guide allows the paper to come up to the lower paper tractor.

Yes, there are two sets of tractors in the new H25 line printer. A lower tractor which is below the print head and an upper tractor which guides the paper out the back of the printer. There are two main reasons for the dual sets of tractors, the first allowing the paper to be held tight for reverse indexing of the paper and second to allow the removal of the last sheet of paper that was printed, without form feeding to the next sheet. Removal of the last sheet of paper requires that the upper tractor be released and the lower tractor remain closed. When the printer is started up again the upper tractor may be closed. Reverse indexing should NOT be done until the upper tractor is closed.

Getting back to the loading of the paper, before pushing the paper into the printer, the cover of the H25 should be opened and cover support bracket locked in place, allowing access to the upper tractor and to the carriage assembly.

What in the world is a carriage assembly? The H25 line printer has a very unique design for loading the paper and ribbons. The entire head, ribbon, and front switch assembly will pull forward by the release of two latches giving full access to the print head, the ribbon and the paper as it is fed into the printer. This carriage assembly is hinged at the bottom. Having the carriage assembly open allows the removal of the ribbon cartridge with only the release of a spring clip. Replacement of the ribbon is also a snap, as it only requires the hooking of one end of the cartridge onto a hook and the other end is held by the spring clip. Ribbon changes take only a few seconds and NO more inky messes. Cartridges sell two for $20.00.

With the cartridge assembly open the paper may be fed into the printer and the paper will rise in front of the lower tractor. Opening the tractors and adjusting them may be done as the paper is fed to the upper tractors. Once the paper is in position the carriage assembly may be lowered into position and the two latches closed.

The cover support bracket should be released and the cover lowered to the original position. The cover completely covers all mechanical parts to reduce the noise of the printing. No more of those buzz saw sounds with the H25 printer.

Before using the printer the RESET button on the front panel must be pressed to tell the printer that you are through loading the paper and that you are ready to start using the printer.

In attempting to drive the H25 the LPH24.DVD device driver was used, only to find no output. After a quick check of the dip switches on the back of the H25 it was determined that it was set to 9600 baud. (H25 is selectable from 110 to 9600 baud.) Using the set option the device driver was set to 9600 and printing began.

The dip switches on the back of the H25 may set by using a screw driver or pen. Several options may be selected with the dip switches including; baud, parity, horizontal pitch, vertical pitch, skip of the fold, busy polarity, page length, discard past end, and auto line feed.

Once the proper baud was set the printer performed without a hickup for the four days until shipment to the Mid-Atlantic Computer Show in Washington, D.C. It was first shown at the local Capital Heath Users' Group (CHUG) and then at the show for four days printing calendars for 1982. The calendars printed were a full 11 by 14-1/2 with over printing and with use of the 33 graphic characters.

Over five hundred calendars were printed and the H25 required only a minor adjustment of the pulley on the head drive motor. An allen wrench took care of that problem in a hurry. Even after all those calendars the ribbon still printed darker then most other ribbons fresh out of the box.

The overall reaction of the show visitors was, "ONLY $1095 in kit! When are they going to be available, I want one".

:GK:

The SOURCE — DIRECTADD and DISEARCH

For those of you who are using the SOURCE instead of MicroNET to communicate with fellow Huggies or Heath/Zenith computer owners, there may be a special feature of the system you have missed. The SOURCE contains two programs known as "DIRECTADD" and "DISEARCH". With these two programs, you can locate fellow HUG types, people who are interested in Heath equipment, or just fellow users that may have similar interests.

You will notice that two programs were listed above. DIRECTADD is the first

Vectored to Page 32

Vectored from 15

of the .BIN extension file and executes it into machine code. It must be present during run time, and has an overhead of 3 to 8 K bytes, depending on the length and complexity of the program. I do not want to go to deep into this and confuse you, I just wanted you to know basically how it worked.

Any program logic errors must then be corrected by going back to the text editor and again compiling the program before it can be run. Like any compiler, it's a bit more of a hassle than an interpreter such as BASIC, but once the debugged program is finished, the results are well worth it.

Any serious individual wanting to learn Pascal will not learn it by reading this column. I recommend several learning aids. I have the Heath Pascal programming course, which is a good start. Two books I highly recommend are PASCAL PRIMER by David Fox and Mitchell Waite, published by Howard Sams & Co. Inc., and PASCAL, by Paul M. Chirlian, published by Matrix publishers, Portland Oregon. I obtained both of these from my local Heath store. Keep in mind you cannot effectively learn any language by reading. You must set up your system and try the examples and read the books and try some more. Until you begin to write and debug, you will know very little. This column is just to introduce you to the language and give you a push.

Now to get into the meat of things. The steps to produce a good Pascal program are outlined below. Define exactly what you want your program to do. Get straight what will be input and output, and the format. Write out the program flow in English, by breaking your program into sections, each which performs an individual task, each of which can be individually designed and debugged. These will be your modules. Continue to use plain English and break your parts down into sub parts, until each section will be easy to write and debug on its own. This is the basis of structured programming. Now translate the small English modules into Pascal code, debugging each subsection before continuing on to the next. Go through each section mentally and ask yourself-- is this what I want it to do? Will this give me the desired result? If not, redo it, or you'll end up doing it over and having to re-compile it later.

Now begin to put all the small pieces together into a main program, calling each module (think of this as a subroutine in BASIC) as you need it. Remember, each variable, constant, or procedure must be defined BEFORE you use it. This keeps things straight and makes you think more about what you are attempting to do.

The general Pascal program has the following format:

```
Program title and declaration of files
used.
Declaration of constants.
Declaration of special variable types.
Declaration of the variables themselves.

Procedure heading I
  Declaration of any local constants or
  variables
  Procedure body
Procedure heading II
  .
  .
  .
Function Heading
  Declaration and function body similar
  to that of procedures

MAIN program body
End
```

Just like BASIC, there are reserved KEYWORDS that cannot be used for anything like variables or constants, but I'll cover these as we use them.

I'll end here for the month. That will give you a chance to get your reading materials, Pascal if you don't have it, and start studying. We'll meet next month and start with some simple programs. See you then.

EOF

Vectored from 14

We will show examples of some of the basic commands of the CCP to get you started in using CP/M for yourself. We will briefly study the system initialization programs, FORMAT, SYSGEN, and MOVCPM, which are part of the transient programs that come in the CP/M software package.

As for this issue, I feel that you should be beginning to feel more comfortable with CP/M, after studying this article. For those of you who are beginning in computers, this article should be of help in understanding the CP/M operating system. For any HDOS users, much of this is "old hat" and needs very little study. If anything, it should be a plain good review of how an operating system works, be it HDOS or CP/M.

<TLJ>

Special HUG/Heath Sale

It is with great pleasure that I announce the following discount program for HUG members only. The Heath Users' Group in cooperation the Heath Company has developed a limited purchase plan for various Heath products listed below. This is a first for HUG and we sincerely hope to continue this trend in the future.

PLEASE READ THIS INFORMATION CAREFULLY!

To place your order for any of the products listed you must complete the information on the form and RETURN THE ENTIRE PAGE or an EXACT COPY OF THE ENTIRE PAGE. Be sure that you include all of the information contained on the form to qualify for these exceptional values. Return the form or copy to the HEATH COMPANY ORDER ENTRY DEPARTMENT, not to HUG. This purchase plan is available through MAIL ORDER ONLY. Therefore, you must return your information to the Heath Company within the time limits specified to complete your purchase of the desired product or products. At this time, there are no limits on the number of each item you can obtain or the number of items appearing on the list that you can buy. You must include your HUG ID to receive the benefits of this puchase plan.

THREE NOTES:

1. Listings for HDOS will carry the same update policy as if purchased at the full price.

2. The H-19-2 is not designed for the newer RFI versions of the H-19. Be sure to check your H-19 if you intend to purchase this particular product.

3. For additional information, refer to your Heathkit catalog for a full description and further mailing instructions.

I am a member in good standing with the Heath Users' Group. My HUG ID is____________________. Please mail me the following items:

Quan.	Model	Description	Regular	Sale	Total
___	H-8-9	H8 PAM-GO ROM	$20.00	$10.00	$______
___	H-11-10	H-11 Wirewrap Card	$35.00	$20.00	$______
___	H-17-3	3 Drive Mod Kit	$85.00	$59.00	$______
___	H-19-2	H-19/H-88 Conversion Kit	$695.00	$495.00	$______
___	H-88-5	Cassette I/O Card	$100.00	$70.00	$______
___	HA-8-2	H8 Music Board	$159.00	$129.00	$______
___	HA-8-3	H8 Color Graphics	$495.00	$379.00	$______
___	HA-8-8	H8 Ext. Config. Opt.	$65.00	$49.00	$______
___	HOS-1-SL	HDOS 2.0 Source Listings	$195.00	$65.00	$______
___	WH-13	Cat Modem	$159.00	$139.00	$______

*Add $3.00 per item ordered for shipping and handling

Enclosed is my __check __ money order __ charge to my __Visa __MasterCard __Heath Revolving Charge

Account No.____________________ Signature____________________

Exp. Date____________________ Name____________________

Merchandise Total....__________ Address____________________

Michigan Residents
add 4% sales tax.....__________ City______________State________

Zip__________

Shipping/Handling....__________

Total Amount of Order__________

Send to: HEATH COMPANY
Benton Harbor, MI 49022

*This offer expires January 31, 1982. Quantities are limited.

Vectored from 17

H11 SOFTWARE

885-1008	Volume I Documentation and Program Listings (some for H11)	$ 9.00
885-1033	HT-11 Disk I	$ 19.00

CP/M SOFTWARE (5-inch only)

885-1201	CP/M (TM) Volumes H1 and H2	%	$ 21.00
885-1202	CP/M Volumes 4 and 21-C	%%	$ 21.00
885-1203	CP/M Volumes 21-A and B	%%	$ 21.00
885-1204	CP/M Volumes 26/27-A and B	%%	$ 21.00
885-1205	CP/M Volumes 26/27-C and D	%%	$ 21.00
885-1206	CP/M Games Disk	%%	$ 21.00

The above CP/M products are 2 disks each.

885-1207	TERM and H8COPY	$ 20.00
885-1208	HUG Fig-Forth H8/H89 2 Disks	$ 40.00
885-1209	Dungeons and Dragons Game MBASIC and H89 or H8/H19	$ 20.00
885-1210	HUG Editor	$ 20.00
885-1211	Sea Battle Game for CP/M	$ 20.00
885-1212	CP/M Utilities I	$ 20.00
885-1213	CP/M Disk Utilities	$ 20.00

% Means CP/M 1.43 only (ORG-4200)
%% Means CP/M 1.43 or 2.2 (Heath)
Other CP/M disks are for 2.2

MISCELLANEOUS

885-0017	H8 Poster	$ 2.95
885-0018	H89 Poster	$ 2.95
885-0019	Color Graphics Poster	$ 2.95
885-4	HUG Binder	$ 5.75

CP/M is a registered trademark of
Digital Research Corp.

Corrections to PrinText

My apologies to Steve Hagins! In REMark, Issue #20 we presented Steve's PASCAL program called PrinText with only two lines missing! To fix Steve's work, please perform the following additions to the listing shown on page 27:

Under the section "PROCEDURE INTRO;" add the following lines directly after the line which now reads "WRITE('FILENAME -->');" ....

```
READLN(FILENAME);
WRITE('ENTER THE NUMBER OF TO BE LEFT AT THE TOP OF PAGE 1: -->');
```

Steve also indicates that there is a small spelling error in a remaining line of the text under the "BEGIN" section. The word spelled "SE" should read "SEE" in the fifth line. Thanks Steve, for bringing these errors to my attention. It seems that my vision was slightly out of tune. BE:

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.

Vectored from 28

program we will discuss here. DIRECTADD or "Directory Add" allows the user to "sign-in" or become a member of the SOURCE Interest Log. At the command prompt (>) type: DIRECTADD. The remaining portions of the program are self-explanatory. By entering your name and various interests in this simple list you become a part of the "known" SOURCE community. Entering your name using DIRECTADD will enable other Heath users locate you provided you have placed in the "interest" catagory, keywords such as "HUG", "HEATH", or "ZENITH". Since HUG would like to know how many of you are using the SOURCE, please consider the DIRECTADD function.

DISEARCH, or "Directory Search", is available to users even if you do not wish to add yourself to the "Directory" using the program described above. DISEARCH is a simple program which is entered by typing DISEARCH at the command prompt (>). Once you have entered the program, you will receive the simple instructions that are necessary to locate other users by searching name, state, or interests. And, if you perform a DISEARCH, each individual that has interests similar to yours will be marked with an asterisk (*) if they are on the system with you. This little additional feature would allow you to directly communicate using the SOURCE "Chat" function.

I hope that all of you that are on the SOURCE would place yourself on the "Interest Directory" so that we at HUG will be able to locate you when we are "ONLINE". Further, adding your name to the Directory will tell us how strong the HUG population on the SOURCE is!

For those of you that are new to the SOURCE, you may wish to examine the HUG section of the POST catagory. To access the HUG general bulletin board, enter the following at the command prompt:

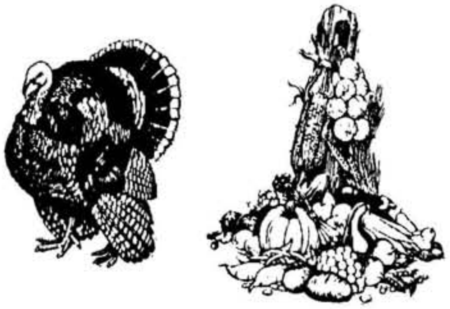
```
>POST READ HUG
```

If you choose to leave a message to other HUGs in the general section type:

```
>POST SEND HUG
```

Anyway, see the information that is contained on this general catagory as there have been several "local" bulletin boards with their phone numbers listed for your convenience.

BE:



Heath
Users'
Group
Hilltop Road
St. Joseph MI 49085