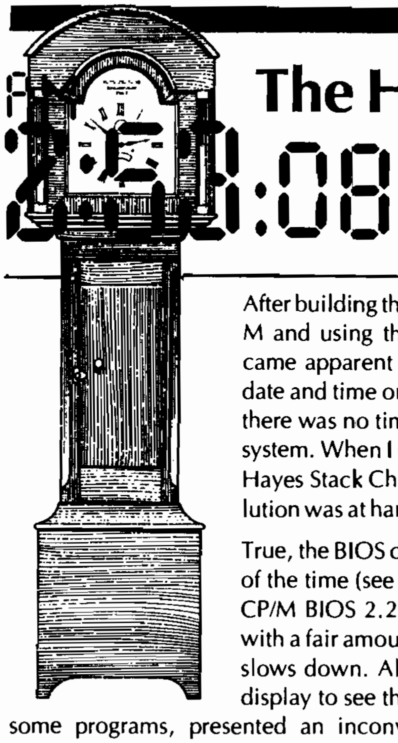

The Hayes Chronograph, an H89 and CP/M

Glen E. Hassebrock, Jr.
c/o Nims Associates, Inc.
363 South Main Street, Suite 420
Decatur, Illinois 62523

After building the Heathkit H89, booting CP/M and using the system for awhile, it became apparent that I would like to put the date and time on displays and printouts. But there was no time or date maintained in the system. When I saw an advertisement for the Hayes Stack Chronograph, I preceived a solution was at hand.

True, the BIOS could be altered to keep track of the time (see REMark November 1981 or CP/M BIOS 2.2.03 from Heath). However, with a fair amount of disk access this "clock" slows down. Also, the need to turn on the display to see the time, and turn it off to use some programs, presented an inconvenience.

The Chronograph costs about the same as a respectable modem, depending on where the purchase is made, from about $189 to $249. If you don't need the time that much, then this might be a bit steep for a "clock". But the Chronograph isn't just a clock, it also keeps track of the date and day of the week.

Physically, the Chronograph is an aluminum box with plastic caps at the front and rear ends. The day of week and time display is a green fluorescent display, similar to the Heathkit Digital Alarm Clock kit GC-1107. On the back is a DCE connector, a write protect switch, a hole for the power pack connector, and calibration access holes. Inside, there is a single circuit board with a place for 3 AA batteries in a holder (all supplied).

Yes that's right, a DCE connector. That means ordinarily connecting it to a DTE connector, of which the H89 with serial interface board has only one, probably already occupied by a modem. Well at first I thought this would present a problem, then I read the owner's manual that comes with the Chronograph.

The owner's manual is complete and has few typographical errors in it. It is complete and fully descriptive of everything a person may wish to know about using the Chronograph, from unpacking to test programs for some popular microcomputers (but not Heath/Zenith). There is also a copy of the two year limited warranty at the back.

Communication to and from the Chronograph is by means of ASCII strings (the characters normally used by Heath/Zenith computers in the USA). There are several commands for setting day, date, time, and dividers for time and date, and for reading the day, date and time. The Chronograph also has an alarm function which can be used to notify a program of a preselected time of day. All responses from the Chronograph end with a carriage return and optionally a line feed.

The first thing I wanted to know was if it survived the delivery service (I ordered it from a mail order company before Heath offered it in their catalog). Following the instructions in the manual, I took the circuit board out and installed the batteries. The display was still blank, as it should be, since only the time, day, and date are retained while the power pack is not supplying power. Plugging in the power pack and connecting it to the Chronograph brought the display to light, and now I pondered how to set it.

The easiest way at hand was to connect the Chronograph to the DTE connector on the H89, just as I would a modem, and use a modem control program to "communicate" with the Chronograph. Since the Chronograph determines baud rate (300 or 1200), parity, etc., from the command sent to it, there was no special modification needed. Following the manual, I set the time, date, and day with no trouble at all. The Chronograph responded with a zero every time, meaning everything was correct. The display had changed to the time and day of week, just as I had set it.

Then I sent the commands to read the date, time and day and got back the date, time and 1 digit for the day. The day is represented by the digit zero meaning Monday, 1 for Tuesday, and so on, to 6 for Sunday. The time can be set in 24 hour mode or 12 hour mode, with A or P for AM and PM indication. Any ASCII character, other than null or carriage return, can be used to separate the date and time digits to provide easier reading. However, these are lost when the Chronograph is running on battery power.

Since my H14 printer is attached at port 340Q and a modem is often attached to port 330Q, this leaves port 320Q available for the Chronograph. This is a DCE connector which would be incompatible with the DCE connector on the Chronograph. But the owner's manual states how to connect the Chronograph to this type of port and referring to a back issue of BYTE, I constructed a null modem cable with male connectors at both ends. Changing the modem control program to communicate with the 320Q port and attempting to get the time showed that this is perfectly acceptable, and now I have the Chronograph attached all the time.

Next, to be useful, there had to be a way to get the Chronograph information without using the modem control program. Attempts to get an MBASIC program to get the time showed that the interpreter was just too slow, if the baud rate for the port is 1200 (300 works fine). Besides I wanted the time and date in other than BASIC programs. My solution: modify the CP/M BIOS so that the Chronograph access code is always available.

Now this really isn't all that difficult. I didn't need to change existing code, I just had to add code. The code needed was (1) set the time and date dividers upon cold boot (2) access the Chronograph for the time, date, and day of week. The difficult part was the need to edit the BIOS with CP/M's ED editor. If you Have the old hard sectored disks, you will need to know how to edit a program on one disk and save it onto another (e.g. A>ED B:BIOS.ASMC:).

Since the new 2.2.03 BIOS can be setup for various disk options, most of the code can be in different locations, depending on the chosen option. Well as any CP/M programmer should know, there is always one thing you can find in the BIOS. That is the jump table, the address of which is at memory locations 1-2. Preceding the warm boot jump (the one actually pointed to by locations 1-2) is the cold boot jump. This jump is not needed after the cold boot has started, so why not use it as a jump pointer to the Chronograph access code!

The first change to the BIOS was to put a TRUE/FALSE indicator in so that the BIOS could be assembled without the Chronograph code, if need be. This goes near the other such indicators, at the beginning of the BIOS, and should look similar to…

```
        IF      (2-(H17T+H37T+H47T+H67T)) SHR 15
%:      TOO MANY DISK DRIVE TYPES SPECIFIED
        ENDIF
CHRONO  EQU     TRUE   ; <=== ADD THIS LINE FOR CHRONOGRAPH
PARTITN EQU     TRUE AND H67T
```

Next comes the actual access code. A safe and easy place to put this is after the disk parameter tables and just before the warm boot code. The processes involved are

(1) send "read the time command"
(2) get the time
(3) send the "read the date" command
(4) get the date
(5) send the "read the day of week" command
(6) get the day of week code
(7) translate the code to a day literal

The access code is…

```
        DB    1
        DB    8
        DB    0,0,0,0
        ENDIF
        IF    CHRONO            ; <== START ADDING THIS CODE
GETTIM  DB    'ATRT',CR         ; READ TIME COMMAND
GETDAT  DB    'ATRD',CR         ; READ DATE COMMAND
GETDAY  DB    'ATRW',CR         ; READ WEEKDAY COMMAND
        DB    16                ; LOCAL STACK SPACE
DAYS    DB    'MONDAY',0,0,0    ; WEEKDAY LITERALS
        DB    'TUESDAY',0,0     ; AND STACK LABEL
        DB    'WEDNESDAY'       ; EACH DAY MUST BE 9
        DB    'THURSDAY',0      ; CHARACTERS LONG THEREFORE
        DB    'FRIDAY',0,0,0    ; PAD SHORT ONES WITH NULL
        DB    'SATURDAY',0      ; CHARACTER
        DB    'SUNDAY',0,0,0
SAVSP   DS    2              ; SAVE CALLERS STACK POINTER
SDAY    DS    9              ; PLACE TO PUT CURRENT DAY LITERAL
SDATE   DS    9              ; PLACE TO PUT THE DATE
STIME   DS    9              ; PLACE TO PUT THE TIME
TIMER0  LXI  H,0             ; GET THE CALLERS STACK POINTER
        DAD     SP
        SHLD    SAVSP              ; SAVE IT
        LXI     SP,DAYS            ; POINT TO LOCAL STACK
        MVI     B,5                ; COMMAND LENGTH
        LXI     H,GETTIM           ; POINT TO COMMAND
        CALL    CLCKOT             ; SEND IT TO CHRONO.
        MVI     B,9                ; EXPECTED RESPONSE LENGTH
        LXI     H,STIME            ; WHERE TO PUT RESPONSE
        CALL    CLCKIN             ; GET IT
        MVI     B,5                ; REPEAT THE ABOVE FOR DATE
        LXI     H,GETDAT
        CALL    CLCKOT
        LXI     H,SDATE
        MVI     B,9
        CALL    CLCKIN
        MVI     B,5                ; GO GET THE DAY CODE
        LXI     H,GETDAY
        CALL    CLCKOT
        CALL    TTYIN   ; GET THE CODE
        PUSH    PSW     ; SAVE CODE
        CALL    TTYIN   ; GET THE ENDING CR
        POP     PSW     ; RESTORE CODE
        LXI     B,9     ; LENGTH OF DAY LITERALS
        LXI     H,DAYS  ; POINT TO LITERALS
        SUI     30H     ; CONVERT CODE FROM ASCII TO BINARY
        JZ      MOVDAY  ; IF MONDAY, THEN WE'RE SET TO MOVE
TIMER1  DAD     B       ; IT ELSE INDEX TO PROPER DAY
        DCR     A       ; LITERAL
        JNZ     TIMER1
MOVDAY  LXI     D,SDAY  ; POINT WHERE TO STORE CURRENT DAY
TIMER2  MOV     A,M     ; MOVE LITERAL THERE
        STAX    D
        INX     H
        INX     D
        DCR     C
        JNZ     TIMER2
        LHLD    SAVSP   ; RESTORE CALLER'S STACK POINTER
        SPHL
        RET             ; AND RETURN TO HIM
CLCKIN  PUSH    H       ; ROUTINE TO GET CHRONO. RESPONSES
        PUSH    B
        CALL    TTYIN   ; INPUT FROM PORT 320Q
        POP     B
        POP     H
        MOV     M,A
        INX     H
        DCR     B
        JNZ     CLCKIN
        RET
CLCKOT  MOV     C,M     ; ROUTINE TO SEND CHRONO. COMMANDS
        PUSH    H
        PUSH    B
        CALL    TTYOUT  ; OUTPUT TO PORT 320Q
        POP     B
        POP     H
        INX     H
        DCR     B
        JNZ     CLCKOT
        RET
        ENDIF                     ; <== LAST LINE TO ADD
WBOOT:  LXI     SP,STACK
        EI
```

I must mention that this routine works if one sets the Chronograph to the 24 hour clock, or military time. In this mode, the time is always the same length; with AM/PM time, the response can be 9 or 10 characters long and one needs to test for the CR instead of a constant length.

To set the time and date dividers and add the jump to the access code, one must next modify the cold boot code. While in ED, enter the command NCB12: ↑ Z and this will take you to the appropriate place. Then add the following code to accomplish the last of the BIOS modification.

```
CB12:
        XRA     A
        STA     LOGDSK
        LXI     H,BIOS
        SHLD    BBIOS
        IF      CHRONO          ; <== START ADDING THIS CODE
TIMER3  MVI     B,6
        LXI     H,SETTDV
        CALL    CLCKOT          ; SEND SET TIME DIVIDER COMMAND
        CALL    TTYIN           ; GET '0' RESPONSE
        CALL    TTYIN
        MVI     B,6
        LXI     H,SETDDV
        CALL    CLCKOT          ; SEND GET DATE DIVIDER COMMAND
        CALL    TTYIN           ; GET '0' RESPONSE
        CALL    TTYIN
        LXI     H,TIMER0        ; GET ACCESS CODE  ADDRESS
        SHLD    WBOOTE-2        ; WRITE OVER COLD BOOT JUMP
        ENDIF                   ; <== LAST LINE TO ADD
        MVI     A,BT$CD
        JMP     GOW
        IF      CHRONO          ; <== START ADDING AGAIN
SETTDV  DB      'ATVT:',CR      ;SET TIME DIVIDER
SETDDV  DB      'ATDT/',CR      ;SET DATE DIVIDER
        ENDIF                   ; <== THE LAST LINE TO ADD
SMSG0:  DB      CR,LF,LF,0
SMSG1:  .....
```

This modifies the BIOS, but "how do you get the date and time from this?" I hear being asked. Well as I said before I wanted to put the time and date on printouts, so I wrote a program to call the access code and print it at the end of the listings on my printer. I gave it the original name... TIME.ASM. Here's its commented listing.

```
; THIS PROGRAM CALLS THE MODIFIED BIOS CHRONOGRAPH ACCESS
; ROUTINE AND THEN PRINTS THE DAY, DATE AND TIME ON THE CRT.
;
; AUTHOR: GLEN E. HASSEBROCK, JR.
;
; COMMAND FOR CRT DISPLAY ONLY:      A>TIME
;
; COMMAND FOR CRT AND LST DISPLAY:   A>TIME P
;
; THIS IS THE 8080 VERSION
;
BASE    EQU     0       ; CP/M EQUATES
TPA     EQU     BASE+100H
;
BDOS    EQU     5
LSTCHR  EQU     5
PSTRNG  EQU     9
CR      EQU     13
LF      EQU     10
DAYOFF  EQU     27      ; OFFSET OF "DAY" FROM EXECUTABLE CODE
DMA     EQU     128
;
        ORG     TPA
TIME:
        LXI     H,0     ; GET CALLER'S STACK AND SAVE IT
        DAD     SP
        SHLD    SAVESP
        LXI     SP,STACK
        ORA     A               ; CLEAR CARRY FLAG
        LHLD    BASE+1          ; GET ADDRESS OF WARM BOOT JMP
        MOV     A,L
        SBI     2               ; POINT BACK TO ADDRESS IN
        MOV     L,A             ; COLD BOOT        \OVERWRITTEN
        MOV     E,M             ; GET ADDRESS OF CHRONO. ACCESS
        INX     H                                    \ROUTINE
        MOV     D,M
        XCHG
        PUSH    H               ; SAVE IT
        LXI     D,TIME1         ; GET RETURN ADDRESS
        PUSH    D               ; PUT ON STACK
        PCHL                    ; CALL CHRONO. ACCESS ROUTINE
TIME1:
        POP     H               ; GET BACK ADDRESS OF CHRONO.
        ORA     A               ; CLEAR CARRY        \ROUTINE
        MOV     A,L             ; SUBTRACT BACK TO DAY ADDRESS
        SBI     DAYOFF          ; SINCE IT LIES BEFORE START OF
        MOV     L,A             ; ROUTINE EXECUTABLE CODE
        JNC     TIME2
        DCR     H
TIME2:
        LXI     D,MDAY          ; POINT TO DAY IN MSG
        MVI     B,9
TIME3:                          ; MOVE DAY FROM ROUTINE TO MSG
        MOV     A,M
        STAX    D
        INX     H
        INX     D
        DCR     B
        JNZ     TIME3
;
        LXI     D,MDATE         ; POINT TO DATE IN MSG
        MVI     B,8
TIME4:                          ; MOVE DATE FROM ROUTINE TO MSG
        MOV     A,M
        STAX    D
        INX     H
        INX     D
        DCR     B
        JNZ     TIME4
;
        INX     H               ; SKIP OVER CARRIAGE RETURN
;
        LXI     D,MTIME         ; POINT TO TIME IN MSG
        MVI     B,8
TIME5:                          ; MOVE IT TO MSG
        MOV     A,M
        STAX    D
        INX     H
        INX     D
        DCR     B
        JNZ     TIME5
;
        LXI     D,MSG           ; PRINT THE MSG ON CON:
        MVI     C,PSTRNG
        CALL    BDOS
;
        LDA     DMA     ; CHECK FOR CHARACTER ENTERED WITH
        CPI     2       ; 2 = SPACE + 'P'        \COMMAND
        JNZ     ENDIT   ; IF NOT, WE'RE FINISHED
;
        INX     H       ; POINT TO CHARACTER ENTERED
        INX     H
        MOV     A,M     ; GET IT AND SEE IF IT WAS A 'P'
        CPI     'P'
        JNZ     ENDIT   ; IF NOT, WE'RE FINISHED
;
        LXI     H,MSG   ; ELSE PRINT ON LST: DEVICE
TIME6:
        MVI     C,LSTCHR
        MOV     A,M
        CPI     '$'     ; UNTIL '$' = END OF MSG
        JZ      ENDIT
        MOV     E,A
        INX     H
        PUSH    H
        CALL    BDOS
        POP     H
        JMP     TIME6
ENDIT:
        LHLD    SAVESP  ; RESTORE CALLER'S STACK
        SPHL
        RET             ; RETURN TO CALLER
MSG:
        DB      CR,LF,'TODAY IS '
MDAY:
        DS      9
        DB      ', '
MDATE:
        DS      8
        DB      '.  THE TIME IS '
MTIME:
        DS      8
        DB      '.',CR,LF,'$'
SAVESP:
        DS      2       ; PLACE TO SAVE CALLER'S STACK
        DS      16      ; LOCAL STACK
STACK:
        END
```

Now for those of you with a desire to stay away from assembly language programming, there are two methods for you to use. The first is to modify the BIOS, as above, and call it from a BASIC program, then peek into memory to get the day, date and time. The second is to write BASIC code to get the information from the Chronograph.

The former is a bit more complex than the latter and would have to be changed if the BASIC program is to be compiled versus interpreted. Therefore, I will present the second option.

In the back of the Chronograph's manual are listings of BASIC routines for systems other than Heath/Zenith, as I stated before. However, it can still be done on Heath/Zenith '89s and H8s (with the H-8-4). The following BASIC program is an adaptation of the manual's programs, and will work on MBASIC 5.21, even compiled if you so desire. To run on earlier versions of MBASIC, the names of all variables must be shortened to two significant characters. As I stated earlier the baud rate for the Chronograph port must be 300 for use by interpreted MBASIC.

```
10 DIM DAY$(6):REM ESTABLISH TABLE OF DAY LITERALS
20 DAY$(0)="MONDAY"
30 DAY$(1)="TUESDAY"
40 DAY$(2)="WEDNESDAY"
50 DAY$(3)="THURSDAY"
60 DAY$(4)="FRIDAY"
70 DAY$(5)="SATURDAY"
80 DAY$(6)="SUNDAY"
90 OST$="ATVT:":GOSUB 300:REM SET TIME DIVIDER
100 OST$="ATVD/":GOSUB 300:REM SET DATE DIVIDER
110 GOSUB 240:REM GET THE TIME
120 GOSUB 260:REM GET THE DATE
130 GOSUB 280:REM GET THE DAY
140 PRINT
150 PRINT "TODAY IS "DAY$", "DATE$".  THE TIME IS "TIME$
160 PRINT
170 PRINT "SHALL I COPY TO LST:? <Y OR N> ":A$=INPUT$(1)
180 IF A$<>"Y" THEN 390
190 LPRINT
200 LPRINT "TODAY IS "DAY$", "DATE$".  THE TIME IS "TIME$
210 LPRINT
220 GOTO 390
230 ' CHRONO TIME ROUTINE
240 OST$="ATRT":GOSUB 300:TIME$=IST$:RETURN
250 ' CHRONO DATE + TIME ROUTINE
260 OST$="ATRD":GOSUB 300:DATE$=IST$:RETURN
270 ' CHRONO DAY ROUTINE
280 OST$="ATRW":GOSUB 300:DAY$=DAY$(VAL(IST$)):RETURN
290 ' CLOCK I/O
300 IF OST$="" THEN 380
310 FOR I%=1 TO LEN(OST$):OUT &O320,ASC(MID$(OST$,I%,1))
320 IF (INP(&O325) AND 32)=0 THEN 320
330 NEXT I%:OST$=""
340 OUT &O320,13:IST$=""
350 IF (INP(&O325) AND 1)=0 THEN 350
360 C%=INP(&O320)
370 IF C%<>13 THEN IST$=IST$+CHR$(C%):GOTO 350
380 RETURN
390 END
```

Now since some of you don't have modem control programs, I suspect that you would like some way to set the Chronograph. Think about it a bit, consider all that Dr. Campbell has taught you in the past issues of REMark, and see if you can write the program on your own. If you need some help, here is a little BASIC program to get you started.

```
5 DIM DAY$(6):REM ESTABLISH TABLE OF DAY LITERALS
10 DAY$(0)="MONDAY"
15 DAY$(1)="TUESDAY"
20 DAY$(2)="WEDNESDAY"
25 DAY$(3)="THURSDAY"
30 DAY$(4)="FRIDAY"
35 DAY$(5)="SATURDAY"
40 DAY$(6)="SUNDAY"
45 GOTO 105
```

```
50 ' CLOCK I/O
55 IF OST$="" THEN 95
60 FOR I%=1 TO LEN(OST$):OUT &O320,ASC(MID$(OST$,I%,1))
65 IF (INP(&O325) AND 32)=0 THEN 65
70 NEXT I%:OST$=""
75 OUT &O320,13:IST$=""
80 IF (INP(&O325) AND 1)=0 THEN 80
85 C%=INP(&O320)
90 IF C%<>13 THEN IST$=IST$+CHR$(C%):GOTO 80
95 RETURN
100 ' MENU SUBROUTINE
105 PRINT
110 PRINT"0.  All finished; end"
115 PRINT"1.  Set to display time"
120 PRINT"2.  Set to display date"
125 PRINT"3.  Set the time"
130 PRINT"4.  Read the time"
135 PRINT"5.  Set the date"
140 PRINT"6.  Read the date"
145 PRINT"7.  Set the day of week"
150 PRINT"8.  Read the day of week"
155 PRINT"9.  Set the date separator"
160 PRINT"10. Remove the date separator"
165 PRINT"11. Set the time separator"
170 PRINT"12. Remove the date separator"
175 PRINT"13. Set the alarm time"
180 PRINT"14. Remove the alarm time"
185 PRINT"15. Turn line feed option on"
190 PRINT"16. Turn line feed option off"
195 PRINT
200 PRINT"Select a function: ";:INPUT F%
205 IF F%<0 OR F%>16 THEN 555
210 IF F%=0 THEN END
215 ON F% GOTO 225,235,245,275,290,315,330,390,405,425,435,
                                          450,460,500,510,520
220 '   FUNCTION 1
225 OST$="ATDT":GOTO 530
230 '   FUNCTION 2
235 OST$="ATDD":GOTO 530
240 '   FUNCTION 3
245 PRINT"Input the current time in ONE of the forms:"
250 PRINT:PRINT"    HHMM00 or HHMM00A or HHMM00P ";
                                                   :INPUT T$
255 IF LEN(T$)<6 OR LEN(T$)>7 THEN 555
260 OST$="ATST"+T$
265 GOTO 530
270 '   FUNCTION 4
275 OST$="ATRT":GOSUB 55
280 PRINT"The time is set to "IST$".":GOTO 105
285 '   FUNCTION 5
290 PRINT"Input the current date in the form YYMMDD: ";
                                                   :INPUT T$
295 IF LEN(T$)<>6 THEN 555
300 OST$="ATSD"+T$
305 GOTO 530
310 '   FUNCTION 6
315 OST$="ATRD":GOSUB 55
320 PRINT"The date is set to "IST$".":GOTO 105
325 '   FUNCTION 7
330 PRINT"0 = MONDAY"
335 PRINT"1 = TUESDAY"
340 PRINT"2 = WEDNESDAY"
345 PRINT"3 = THURSDAY"
350 PRINT"4 = FRIDAY"
355 PRINT"5 = SATURDAY"
360 PRINT"6 = SUNDAY"
365 PRINT"Enter the number corresponding to today: ";
370 T$=INPUT$(1):PRINT T$
375 IF T$<"0" OR T$>"6" THEN 555
380 OST$="ATSW"+T$:GOTO 530
385 '   FUNCTION 8
390 OST$="ATRW":GOSUB 55
395 PRINT"The day is set to "DAY$(VAL(IST$))".":GOTO 105
400 '   FUNCTION 9
405 PRINT"Type the character to use as a separator
                                              in the date: ";
410 T$=INPUT$(1):PRINT T$
415 OST$="ATVD"+T$:GOTO 530
420 '   FUNCTION 10
425 OST$="ATVD":GOTO 530
430 '   FUNCTION 11
435 PRINT"Type the character to use as a separator
                                              in the time: ";
440 T$=INPUT$(1):PRINT T$
445 OST$="ATVT"+T$:GOTO 530
450 OST$="ATVT":GOTO 530
455 '   FUNCTION 13
460 PRINT"Input the alarm time in ONE of the forms:"
465 PRINT:PRINT"    HHMM or HHMMA or HHMMP "
470 PRINT"(note that this form must match the
                                          current time form)"
475 INPUT T$
480 IF LEN(T$)<4 OR LEN(T$)>5 THEN 555
485 OST$="ATAS"+T$
490 GOTO 530
495 '   FUNCTION 14
500 OST$="ATAC":GOTO 530
505 '   FUNCTION 15
510 OST$="ATLS":GOTO 530
515 '   FUNCTION 16
520 OST$="ATLC"
525 '    COMMON COMMAND CHRONO. CHECK RESPONSE
530 GOSUB 55
535 OK=VAL(IST$)
540 IF OK=0 THEN PRINT "DONE":GOTO 105
545 IF OK=8 THEN PRINT "program error for this function.":
    GOTO 105
550 PRINT "The write-protect switch is up."
555 PRINT CHR$(7)"This command was ignored.":GOTO 105
```

Well my Chronograph is telling me it's time to let you go back to playing with your own systems. I hope this helps those of you who need, or prefer, accurate time, or maybe just the convenience of not having to set it every time the system is booted. My Chronograph has performed flawlessly since I got it, except it has been losing a few seconds per couple of weeks. Now if I could persuade my pocketbook to let me get a Heathkit frequency counter...

Some final notes...I set CP/M to run TIME.COM upon cold boot and have the date and time printed as a greeting by the time the CRT warms up. Don't forget, CP/M is a registered trademark of Digital Research, Hayes Stack Chronograph is a registered trademark of Hayes Microcomputer Products, and of course you know Heathkit, H89 and H8 are registered trademarks of Heath Company.

About the Author:

Glen is a data processing consultant for Nims Associates, Inc., located in Decatur, Illinois. His "computer" background includes COBOL, PL/I, FORTRAN, BASIC, TSO, TSO/SPF, IBM's IMS DB/DC & MFS, Z80 assembler, and 8080 assembler. Besides consulting work, he has taught at the Decatur area community college. He graduated from Illinois State University, in December 1977, with a B.S. degree, after majoring in Chemistry and Applied Computer Science. By the time you read this, he will have recently become married, not to his H89, but to his heartthrob of 11.2635 years. His major hobbies are building Heathkit kits (who doesn't?), reading (to keep up with the computers) and trying things out on his H89.