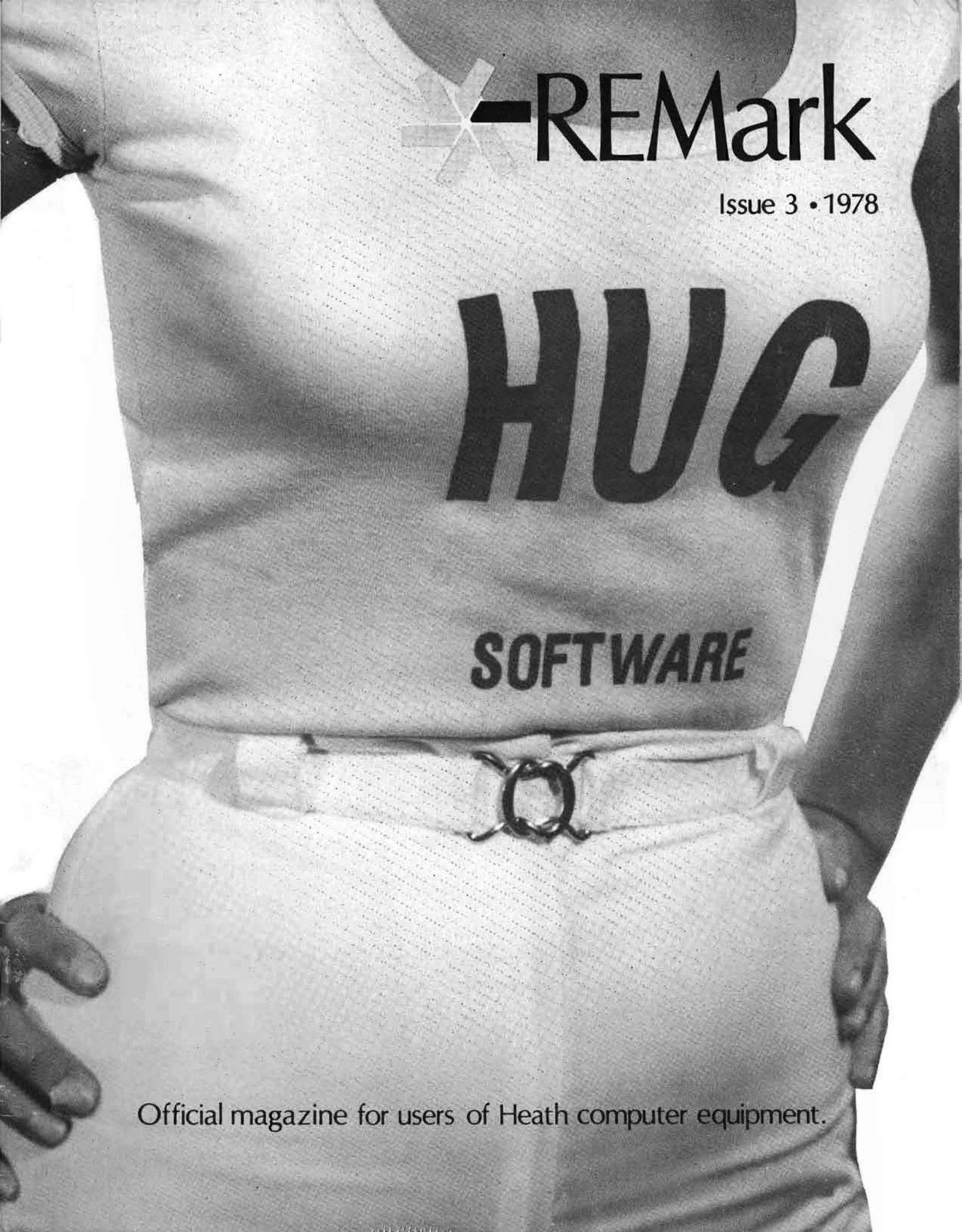
REMark

Issue 3 • 1978



Official magazine for users of Heath computer equipment.

An unidentified HUG member models a special tee shirt to announce the software library is "up and running." If you haven't received your catalog by now, please let us know. The library is growing every day . . . Do you have anything to contribute? Let's hear from you!

about the cover

on the stack

>CAT

"REMARK" is a HUG membership magazine published quarterly. The following membership rates apply.

	U.S. Domestic	Canada & Mexico	Internat'l*
Initial	$14	$16	$24
Renewal	$11	$13	$18

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

Send payment to: Heath User's Group, Hilltop Road, St. Joseph, MI 49085. back issues that are available cost $2.50 postpaid to U.S. destinations. Request for magazines mailed to foreign countries should specify mailing method and add the appropriate cost.

All prices shown in REMark are U.S. mail order prices. Products sold through Heathkit Electronic Centers are slightly higher. For international orders, contact your nearest Heath Co. distributor or the Heath International Division, PO Box 440, St. Joseph, MI 49085.

Articles submitted by users and published in REMark, which describe hardware modifications, are not supported by Heath Service Centers or technical consultants.

HUG Manager and Editor .....Jim Blake
Graphics .....Ron Hungerford

-REMark

IN THIS ISSUE

There seems to be a lot of interest among you in computer assisted instruction (CAI). Several programs of this nature have been submitted for inclusion in the software library. CAI is the subject of an excellent article and program written by Bill Nico of Dallas. (Bill is the author of Heath's EC-1100 BASIC programming course and the forthcoming 8080 programming course.)

Many of you have asked how to convert your H9 to lower case. Howard Nurse describes a pretty simple method in his article, "A New Case for the H9."

Bob Behar and Neal Rogers teamed up and present their version of how best to use GET, PUT, DUMP, LOAD and UNLOCK under program control.

Our congratulations to Michael Jones of Perrysburg, Ohio for winning the first software contest! His entry appeared in the last issue of REMark, before it was selected the winner. Mike's entry was a disassembler written in BASIC. Second place honors are held by Russel Wiedle, Topeka, Kansas for his Cash Flow/ Projection program. (Beautifully prepared!) And third place goes to Robert Bombach's Machine Code program which displays rotating messages on the H8 front panel while it plays music.

ET-3400 — Meet the Outside World.

Later this year, ET-3400 users will be able to communicate with the outside world with a new accessory. This new product offered both in kit and wired form, includes a cassette interface, RS-232 communications link for a terminal, such as the H9 and expanded memory to 4K and tiny basic in ROM. We understand it will sell for less than $200. Therefore, we should be able to expect the software library to include some interesting 6800 programs in the near future.

Variable Length Character String Cassette I/O In Extended Benton Harbor Basic #10.01.02

Gene R. Bellinger
USS Edward McDonnell (FF-1043)
FPO New York, NY 09501

Anyone who has been continuously frustrated by the lack of data I/O capabilities with their H8 system may find some temporary relief in the following routines.

These routines use the load and dump routines in the panel monitor to move specified locations of memory to and from tape. It is necessary to configure BASIC in such a manner that there is at least 256 bytes of memory above that which BASIC may use. The variable M is set equal to the address of the first available byte of user memory.

Dump Routine (1000 — 1120)

The length of the character string to be dumped (Z$) is determined and stored in location M, then the string itself is moved to locations M+1 to M+Z0. Next Z is set equal to the address of the last byte used (M+Z0). Now the address of the dump routine in PAM is stored in USRFNC. Line 1070 places the start dump address in locations 8192 and 8193 where PAM expects to find it. Line 1080 places the end dump address on the address buss. Next the USR function is invoked which causes a jump to the PAM address stored and the memory is dumped to cassette.

Load Routine (2000 — 2080)

The load routine address in PAM is stored in USRFNC then USR function is invoked which is actually a jump to the load routine in PAM. Since the dump routine stored the address in memory at which the dump started, the load routine puts the data in the same place it came from. The data is moved back into Z$ (2030 to 2050) and the routine returns with Z$ containing the string read from tape.

You may recall that I said "temporary relief" in the first paragraph. This relief leads to frustration due to the length of time wasted moving characters around in core, but some I/O is better than none, for the moment anyway.

At present I have found out how to locate the variable and its length in the symbol table created by BASIC. If anyone has found a method of determining where it is then stored I would appreciate knowing. This would allow the data to be dumped directly from resident memory with no conversions involved, thus resulting in considerable time savings. HELP!

Vectored to Page 11

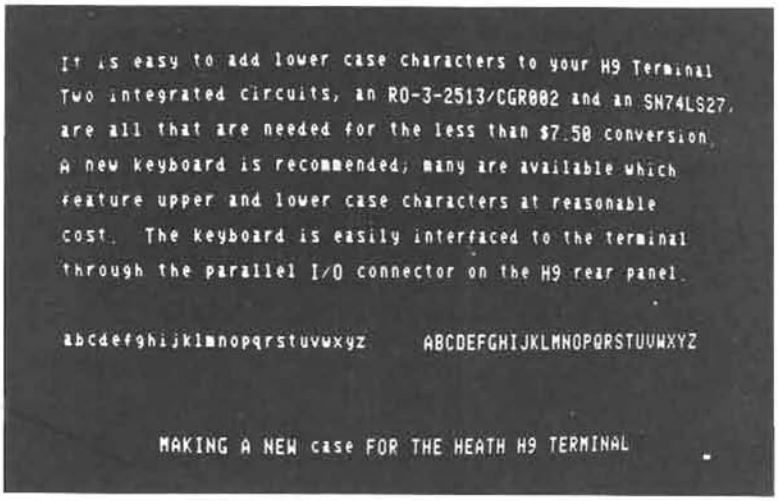


FIGURE 1

Have you wondered if it would be possible to display lower case alphabetic characters on your Heath H9 Terminal? If one reason you invested in a computer is to take advantage of word processing programs, a display restricted to upper case is as useful to you as a one octave piano is to a composer.

After studying the circuitry, I found it would be easy to modify the H9 to include the missing characters. The keybord on the H9 would be difficult to convert, but Heath has provided a convenient parallel I/O connector on the rear panel of the terminal, through which an external keyboard having upper and lower case can communicate.

This article describes how you can add the integrated circuits and keyboard to your H9 to make it a full upper/lower case terminal. A photograph showing the results of the modification is shown in figure 1.

Howard L. Nurse
665 Maybell Avenue
Palo Alto, California 94306

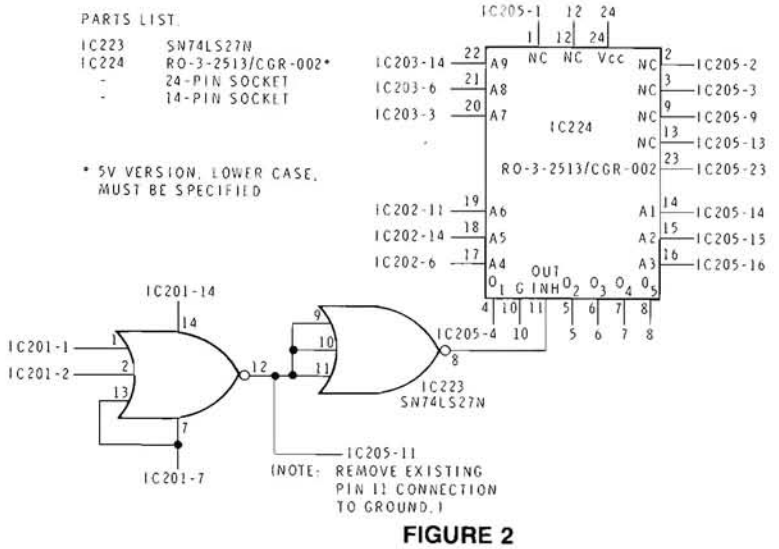


FIGURE 2

CIRCUIT MODIFICATIONS

A circuit diagram and parts list for the changes to the H9 Character Generator Circuit Board are shown in Figure 2. Two integrated circuits, an RO-3-2513/CGR-002 and an SN74LS27, must be added to generate the new characters.

The lower case memory, an RO-3-2513/CGR-002, has a surprise in store for the unwary. For some reason (probably perfectly logical) the addresses on the input of the CGR-002 chip are inverted with respect to its upper case twin. The H9 modification would have been simplified had the address formats of the two character generator chips corresponded. The General Instrument catalog lists a version of the 2513, the CGR-005, which has correct address levels, but this device is not readily available through mail order outlets. Fortunately, the required inverted addresses are available on the inverted Q outputs from IC202 and IC203. The outputs from the character generator chips are directly compatible.

The outputs from the character generator chips can be made active or high impedance, depending on the level of the enable inputs on pin 11 of each chip. When pin 11 is low the outputs of the corresponding chip are active.

IC223, and SN74LS27 Triple NOR Gate, has been added to the character generator board to select which chip will be active. When bits 6 and 7 of the incoming ASCII character are high, IC223 recognizes that a lower case character is present and enables the lower case character generator chip. At all other times the upper case chip is enabled by the same IC.

WIRING THE TWO INTEGRATED CIRCUITS

The two new Integrated Circuits are piggy-backed onto host IC's on the Character Generator board. A photograph of a modified board is shown in Figure 3. I found the easiest way to add the devices was to mount a socket on the original chip, and then to mount the new IC in the socket. A total of 7 external connections must be made to the character generator chip through pins on the socket which have been bent out at right angles from the socket.

Once the socket pins have been bent, place the socket over the existing chip (which you should remove from the board while performing the surgery) and solder the remaining pins to the corresponding pins on the IC. Note that the chip enable, pin 11, is also bent out from the original upper case chip. This is the only pin on an IC which must be bent as part of the modification.

IC223 is attached to IC201 in the same manner. For this IC, however; only pins 1, 2, 7, and 14 are connected to the host chip. All other pins on the socket must be bent so they do not touch the host chip below.

The wiring of the two integrated circuits is completed by connecting the six address lines to IC202 and IC203, and the two chip enable inputs to IC223 as shown in the schematic.

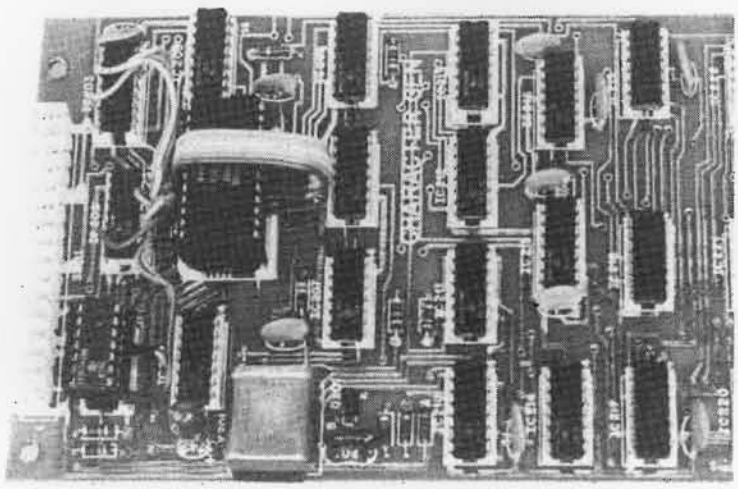


FIGURE 3

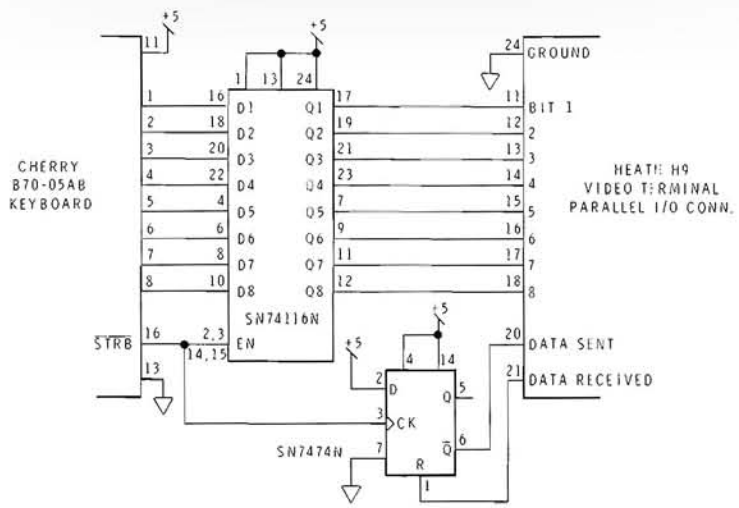


FIGURE 4

KEYBOARD INTERFACE

A variety of keyboards which provide lower case alphabetic characters are available at reasonable cost. Look for one which has a parallel output port and provisions for locking the alphabetic characters in upper case during the times when you don't want lower case. I chose the new Cherry "Pro" which offers much flexibility in addition to meeting the minimum requirements.

The Cherry "Pro" has five user defined keys which you can wire to provide any output character you desire. I am using four of them to provide Control-I (TAB), Control-S, (Stop), Control-Q (Go); and Control-C. By adding an SN74123N dual monostable multivibrator IC, the keyboard has a "Typamatic" mode which repeats all keys if they are held down for more than a half second.

A schematic for the wiring between the keyboard and the terminal is shown in Figure 4. The data transfer handshaking requires that the parallel data from the keyboard be present from the keyboard strobe until the terminal has sent back a signal saying that the data has been accepted. This handshaking takes two additional IC's on the output of the keyboard to latch the data.

SOFTWARE

Heath software will support lower case characters. The Text Editor (Ted-8) and Extended BASIC are the two programs which can make the most use of the new capabilities of your terminal. Ted-8 is easily adapted to accept both upper and lower case characters by typing "L" while in the command mode, and then answering "Y" to the computer's question "LOWER CASE CHARACTERS (Y/N)"

Extended BASIC must be configured to accept lower case characters. Start with the distribution tape and add any required patches. Then follow the instructions in Appendix A of the BASIC section of your software manual. Type the prompt L, to which the computer will respond "LOWER CASE (Y/N)?" A "Y" response will configure the software to accept both upper and lower case characters as String Data.

EOF

(An Experiment in Computer Assisted Instruction)

MY OLD SCHOOL MARM,

Miss HEATH

By: Willard I. Nico
DELTA t
11211 Katy, Suite 470
Houston, Texas 77079

While working on a consulting assignment involving programmed instruction techniques, my research led to some fascinating information about what is being done to help people learn.

Educators are always seeking ways to improve the learning process. One of the most promising to appear in recent years puts the computer to work presenting facts to learn, asking questions and evaluating answers.

Computer-assisted instruction, or CAI as it's commonly called, has grown to include special computers for the purpose (the Digital Equipment Corporation CLASSIC model) and even a special programming language (PILOT).

Although visions of CAI have sent school administrators pouring over bank balances with their tongues hanging out, the idea is, surprisingly, not new. The basic technique was first described as a way to organize textbooks so that students could learn faster and easier.

"Programmed Instruction" is generally credited as having been "Invented" by B. F. Skinner in 1954. Applying ideas gained from his laboratory experiments in psychology to the learning process, Skinner suggested a method of teaching that has evolved into the CAI techniques used today.

Skinner once wrote, "One of the great sources of inefficiency in modern education is due to our effort to teach a group of students at the same rate." Contrast this with the psychology of a student learning from the computer: He proceeds at his own pace, is actively involved in the learning process and has no fear that his mistakes will be ridiculed by his fellow students. The goal is achievement rather than just trying to keep up with the rest of the class.

Acknowledging the basic principles of programmed instruction, psychologists sought ways to improve the technology of instructional design. For example, where Skinner proposed a fixed sequence of steps to be taken by every student who goes through the program, it was Norman A. Crowder who launched the "branching" style that uncorks the capacity of the computer to help us learn.

In an article titled "Intrinsic Programming," Crowder described the technique:

In each program step, the student is given a "unit" of material to read, usually a paragraph of thirty to seventy words. This material is followed by a multiple-choice question. The student's answer choice determines directly and automatically what material he will see next. If he chooses the right answer to the question, he is automatically presented with the next paragraph of material and the next question. If he chooses an incorrect answer, he is automatically presented with material written specifically to correct the particular error he has just made.

Later writings by Richard Atkinson, James Evans, Lassar Gotkin, Susan Markle and B.F. Skinner refined the use of "Teaching Machines."

After delving deep in the available literature, I started thinking of ways that CAI could be applied to the personal computer. How could I arrange to have my Heath H8 computer teach me something? The answer came at lunch one day.

While waiting for our food, my friend John Busch was doing one of his favorite tricks: showing off his encyclopedic memory. The subject for the day was the capital cities of the United States. As usual, John knew every one of them while the rest of us could only name ten or so. I was particularly good at California (where I was born) and Texas (where I now live), but they never suggested those states!

"North Dakota," someone asked. Before I could think of a single city in the whole state of North Dakota, John came back with "Bismark." And so it went.

Editor's Note:

Mr. Nico is the author of the EC-1100 BASIC programming course — and is currently preparing the 8080 programming course which will be available late this year. :JB:

Later that afternoon, I realized that the ideal subject for my experiment in CAI had presented itself at lunch. The data base was small enough to reside in the computer's memory and thus the widely varying types of external storage would not be an obstacle to others being able to use the program. While it's debatable whether the facts to be learned will be of any real value to everyone, at least it's interesting to be able to name Montpelier as the capital of Vermont in case the question comes up in casual conversation.

After writing the "Name the State Capitals" program and RUNning it a number of times, I can state that CAI is certainly a painless way to learn. Even had I taken the trouble to try, I would not have been able to learn the capitals of all 50 states by simply attempting to memorize them from a printed list. CAI grabbed my interest, forced my participation and engraved the answers in my mind.

While the program doesn't follow Crowder's idea of branching to special text based on the answer to each question, it does a good job of teaching by repetition and reinforcement.

Now that I can name all the capitals without fail, it's a pity that John has moved to Iowa (capital: Des Moines) and I can't show off!

THE PROGRAM

The "Name the State Capitals" program is a modest example of CAI, to be sure, but some of the important techniques are included.

First, there is the interactive dialog. The person taking the quiz (may I use the term quizee?), is asked for his name at the beginning of the program RUN. After that, many of the computer's responses include his name to increase the personal involvement "feel."

Second, there are several responses in each category of query, accolade and admonishment. The program selects messages from each group on a rotating basis to avoid a mechanical dialog and make the interaction more interesting.

Third, the states are quizzed in a random order so that the correct capital is associated with the proper state and not its position in the order of the quiz.

Fourth, the quizee gets more than a "yes" answer. If, for example, he guesses that Trenton is the capital of New Hampshire (it isn't; Concord is!), he is told that "Trenton is the capital of one of the states, but not New Hampshire". This reinforces and builds confidence; he has learned that Trenton is a capital, although not the capital he is looking for.

Fifth, the number of correct answers is tallied so that the person can measure his progress.

If you would like to be able to name all 50 state capitals from memory, just type in the program and take the quiz a few times. While the program was written in Benton Harbor BASIC, there are no gimmicks and it should RUN fine on just about anybody's version.

By the way, all REMark line numbers end in 5 so that you can use an automatic line numbering feature, such as BUILD 10,10 in Benton Harbor BASIC, when typing the program.

After you have learned the capitals of all the states, you'll be able to impress friends with your new-found knowledge — it's bound to be an asset at parties.

And, by the way, keep a look-out for John Busch. Ask him the capital of South Dakota (Pierre); he gets that one wrong sometimes.

HOW IT WORKS

Line 120 of the program clears the CRT screen of pre-RUN leftovers so that everything begins with a clean slate. If your CRT will accept some particular code as an "erase" command, use that instead. As an example:

```
120 PRINT  CHR$(26);
```

erases my screen.

After PRINTing the heading, line 210 READS the states and their capitals into the S$ array of 50 rows and 2 columns from the DATA list in lines 760 through 980.

Line 220 asks for the quiz-taker's name and saves it in variable N$ for use in some of the messages.

Line 260 sets the program variables; 50 states to quiz (variable S), no right or wrong answers (variables R and W), and the first cyclic message in each group: query (variable Q), response to correct answer (variable Y) and response to incorrect answer (variable N).

In line 270, variable A$, which receives the quizee's answers, is cleared of previous data. The query message pointer (variable Q) is set to the next number in the cycle, and the ON . . . . GOTO Statement causes the program to jump to the line containing that message.

At line 330, the Random Number Generator is called upon to pick from among the states still unquizzed. It works this way: Variable S monitors the number of states left to quiz. Remember that whatever number is

delivered by the RND function, it's bound to be less than one. Suppose RND delivers 0.329745 and there are 50 states left to quiz. The product of S times RND gives us 16.48725. Then, we add 1 to it so that we are always sure of getting a number between 1 and S. In this case we get 17.48725. The INT function discards the decimal portion of the number, leaving us with 17 which we assign to variable X and use to point to one of the rows in our S$ array to give us our randomly selected state.

In line 340, the state quizzed is PRINTed by selecting column 1 of row X; S$(X,1). The quizee types his answer into variable A$. If nothing is typed, a special message is PRINTed for variety.

At line 360, the answer in A$ is checked against the correct answer in column 2 of row X in our data array; S$(X,2). If the answer is correct, we GOTO the right part of the program. If it's wrong, an admonishment message is selected by line 410 and we give the customer another chance.

Line 510 gets the second try answer when appropriate and line 520 checks it.

If the answer is still wrong, line 530 tells the correct capital and adds one to the W variable which is keeping track of wrong answers.

Line 540 counts right answers in variable R and selects the next congratulatory response in the sequence.

Line 600 checks the content of variable S to see if the quiz is over.

Lines 620 through 640 shuffle the data around in our array so that no state is quizzed more than once. Here's how it works: The selected state and its capital are picked up in temporary variables T1$ and T2$. Then, all states and capitals from that point to the end of the array are moved down one place. Finally, the selected state and its capital are popped back into the array in the last position. Since variable S was counted down one in line 330, the next selection can't be the one we just used.

Line 650 checks to see if all the answers were right. If so, a special congratulations on a perfect score is PRINTed.

If the score wasn't perfect, lines 690 and 700 tell how many rights and wrongs were achieved.

Finally, lines 710 and 720 give an opportunity to go through the quiz again in an attempt to better the score.

If the quizee calls it quits, lines 730 and 740 bid him bye-bye and thank him for taking the quiz on state capitals.

I hope you enjoy learning the state capitals as much as I did, and that you get some satisfaction from having your computer help you learn. Of course, when everyone learns all the capitals, I won't have anyone to show off to — but I guess that's the Maine (Augusta) idea.

EOF

```
100 REM  ** THE "NAME THE STATE CAPITALS" PROGRAM **
110 REM
115 REM  SIXTEEN LINE FEEDS TO CLEAR SCREEN
120 FOR I=1 TO 16:PRINT :NEXT I
125 REM  PRINT PROGRAM HEADING
130 PRINT TAB(10);"****************************************"
140 PRINT TAB(10);"* THE 'NAME THE STATE CAPITALS' PROGRAM *"
150 PRINT TAB(10);"*";TAB(50);"*"
160 PRINT TAB(10);"*      BY....  WILLARD I. NICO          *"
170 PRINT TAB(10);"****************************************"
180 PRINT:PRINT TAB(16);"HERE IS A QUIZ TO SEE HOW MANY"
190 PRINT TAB(17);"STATE CAPITALS YOU CAN NAME."
195 REM  DEFINE ARRAY - 50 ROWS AND 2 COLUMNS - FOR STATES
200 PRINT:PRINT:DIM S$(50,2)
205 REM  READ THE STATES AND THEIR CAPITALS INTO THE ARRAY
210 FOR A=1 TO 50:FOR B=1 TO 2:READ S$(A,B):NEXT B:NEXT A
215 REM  GET QUIZ-TAKER'S NAME FOR USE IN MESSAGES
220 LINE INPUT "FIRST, PLEASE TELL ME YOUR NAME....  ";N$
230 PRINT:PRINT "THANKS, ";N$;"!  HERE WE GO....."
240 PRINT "I'LL NAME A STATE AND YOU TELL ME ITS CAPITAL"
250 PRINT:PRINT "HERE'S THE FIRST ONE....."
255 REM  NO. STATES TO QUIZ = 50, ANSWER AND MESSAGE COUNTS = 0
260 S=50:R=0:W=0:Q=0:Y=0:N=0
```

```
265 REM  ZAP LAST ANSWER - BUMP MESSAGE COUNTER AND PICK NEXT
270 PRINT:PRINT:A$="":Q=Q+1:ON Q GOTO 280,290,300,310,320
275 REM  FIVE QUERY MESSAGES PICKED ON ROTATING BASIS
280 PRINT "WHAT CITY IS THE CAPITAL OF ";:GOTO 330
290 PRINT "HERE'S THE NEXT ONE..... ";:GOTO 330
300 PRINT "TELL ME THE CAPITAL OF ";:GOTO 330
310 PRINT "OK, ";N$;"!  HOW ABOUT THE CAPITAL OF ";:GOTO 330
320 PRINT "HERE'S A TOUGH ONE!  NAME THE CAPITAL OF ";:Q=0
325 REM  PICK ONE OF REMAINING STATES AT RANDOM - COUNT IT
330 X=INT(S*RND(1))+1:S=S-1
335 REM  PRINT NAME OF STATE AND GET GUESS OF ITS CAPITAL
340 PRINT S$(X,1);:LINE INPUT "?  ";A$:IF LEN(A$)>0 GOTO 360
345 REM  SPECIAL MESSAGE IF NO ENTRY ON KEYBOARD
350 PRINT "DON'T KNOW THAT ONE, ";N$;"?":GOTO 510
355 REM  CHECK IF IT'S THE RIGHT ANSWER
360 IF A$=S$(X,2) GOTO 540
365 REM  CHECK IF ANSWER IS THE CAPITAL OF SOME OTHER STATE
370 FOR I=1 TO 50:IF A$=S$(I,2) GOTO 390
380 NEXT I:GOTO 410
385 REM  SPECIAL MESSAGE IF IT'S A CAPITAL BUT NOT THE RIGHT ONE
390 PRINT A$;" IS THE CAPITAL OF ONE OF THE STATES,"
400 PRINT "     BUT IT'S NOT ";S$(X,1):GOTO 510
405 REM  BUMP MESSAGE COUNTER AND PICK NEXT
410 N=N+1:ON N GOTO 420,430,450,470,490
415 REM  FIVE WRONG ANSWER MESSAGES PICKED ON ROTATING BASIS
420 PRINT "SORRY, ";N$;"!  ";A$;" IS NOT CORRECT!":GOTO 510
430 PRINT "PEOPLE WHO LIVE IN ";S$(X,1);" KNOW THAT"
440 PRINT "     ";A$;" ISN'T THE CAPITAL!":GOTO 510
450 PRINT "MANY PEOPLE GUESS ";A$;", ";N$;","
460 PRINT "     BUT IT'S NOT THE CAPITAL OF ";S$(X,1);"!":GOTO 510
470 PRINT "THAT'S NOT RIGHT, ";N$;"!"
480 PRINT "     BUT I'LL GIVE YOU ANOTHER CHANCE.....":GOTO 510
490 PRINT "OOPS!  ";A$;" MAY BE A FAMILIAR CITY"
500 PRINT "     IN ";S$(X,1);", BUT IT'S NOT THE CAPITAL!":N=0
505 REM  GET SECOND CHANCE ANSWER
510 PRINT "TRY AGAIN! - ";S$(X,1);:LINE INPUT "?  ";A$
515 REM  CHECK IF IT'S RIGHT
520 IF A$=S$(X,2) GOTO 540
525 REM  REVEAL CORRECT ANSWER - COUNT ONE WRONG
530 PRINT "THE CORRECT ANSWER IS ";S$(X,2):W=W+1:GOTO 600
535 REM  BUMP MESSAGE COUNTER AND PICK NEXT
540 R=R+1:Y=Y+1:ON Y GOTO 550,560,570,580,590
545 REM  FIVE RIGHT ANSWER MESSAGES PICKED ON ROTATING BASIS
550 PRINT "GOOD GOING, ";N$;"!  THAT'S RIGHT!":GOTO 600
560 PRINT "IT SURE IS ";A$;", ";N$;"!  KEEP UP THE GOOD WORK!":GOTO 600
570 PRINT "ANOTHER RIGHT ANSWER, ";N$;"!  YOU'RE DOING FINE!":GOTO 600
580 PRINT "PEOPLE IN ";S$(X,1);" ARE PROUD OF YOU, ";N$;"!":GOTO 600
590 PRINT "THAT'S RIGHT, ";N$;"!  YOU'VE GOT";R;"RIGHT SO FAR!":Y=0
595 REM  CHECK IF ALL STATES QUIZZED
600 IF S=0 GOTO 650
605 REM  IF VERY FIRST STATE QUIZZED IS LAST IN LIST, SKIP SHUFFLE
610 IF X=50 GOTO 270
615 REM  SHUFFLE STATE AND CAPITAL TO END OF LIST, MOVE OTHERS DOWN
620 T1$=S$(X,1):T2$=S$(X,2)
630 FOR I=X TO 49:FOR B=1 TO 2:S$(I,B)=S$(I+1,B):NEXT B:NEXT I
640 S$(50,1)=T1$:S$(50,2)=T2$:GOTO 270
645 REM  QUIZ IS OVER - CHECK FOR PERFECT SCORE
650 PRINT:PRINT:PRINT:IF R<>50 GOTO 690
655 REM  PRINT PERFECT SCORE MESSAGE IF APPROPRIATE
660 PRINT CHR$(7);CHR$(7);"!!! A PERFECT SCORE, ";N$;" !!!"
```

```
670 PRINT "!!! C O N G R A T U L A T I O N S !!!";CHR$(7);CHR$(7)
680 GOTO 710
685 REM  SHOW TOTALS OF RIGHT AND WRONG ANSWERS
690 PRINT "HERE'S THE FINAL SCORE, ";N$;"!  OUT OF 50 STATES...."
700 PRINT "YOU GOT";R;"RIGHT AND";W;"WRONG."
705 REM  CHECK IF WANT TO TAKE THE QUIZ AGAIN
710 PRINT:PRINT:LINE INPUT "WANT TO TRY AGAIN?  ";A$
720 IF LEFT$(A$,1)="Y" GOTO 260
725 REM  NOPE - WANTS TO QUIT.  THANK HIM AND BYE-BYE!
730 PRINT:PRINT:PRINT "OK, ";N$;"!"
740 PRINT "THANKS FOR TAKING THE QUIZ ON STATE CAPITALS."
750 PRINT:PRINT:END
755 REM  HERE'S THE DATA BASE - THE STATES AND THEIR CAPITALS
760 DATA "ALABAMA","MONTGOMERY","ALASKA","JUNEAU","ARIZONA"
770 DATA "PHOENIX","ARKANSAS","LITTLE ROCK","CALIFORNIA"
780 DATA "SACRAMENTO","COLORADO","DENVER","CONNECTICUT"
790 DATA "HARTFORD","DELAWARE","DOVER","FLORIDA","TALLAHASSEE"
800 DATA "GEORGIA","ATLANTA","HAWAII","HONOLULU","IDAHO","BOISE"
810 DATA "ILLINOIS","SPRINGFIELD","INDIANA","INDIANAPOLIS"
820 DATA "IOWA","DES MOINES","KANSAS","TOPEKA","KENTUCKY"
830 DATA "FRANKFORT","LOUISIANA","BATON ROUGE","MAINE"
840 DATA "AUGUSTA","MARYLAND","ANNAPOLIS","MASSACHUSETTS"
850 DATA "BOSTON","MICHIGAN","LANSING","MINNESOTA","ST. PAUL"
860 DATA "MISSISSIPPI","JACKSON","MISSOURI","JEFFERSON CITY"
870 DATA "MONTANA","HELENA","NEBRASKA","LINCOLN","NEVADA"
880 DATA "CARSON CITY","NEW HAMPSHIRE","CONCORD","NEW JERSEY"
890 DATA "TRENTON","NEW MEXICO","SANTA FE","NEW YORK","ALBANY"
900 DATA "NORTH CAROLINA","RALEIGH","NORTH DAKOTA","BISMARCK"
910 DATA "OHIO","COLUMBUS","OKLAHOMA","OKLAHOMA CITY","OREGON"
920 DATA "SALEM","PENNSYLVANIA","HARRISBURG","RHODE ISLAND"
930 DATA "PROVIDENCE","SOUTH CAROLINA","COLUMBIA","SOUTH DAKOTA"
940 DATA "PIERRE","TENNESSEE","NASHVILLE","TEXAS","AUSTIN"
950 DATA "UTAH","SALT LAKE CITY","VERMONT","MONTPELIER"
960 DATA "VIRGINIA","RICHMOND","WASHINGTON","OLYMPIA"
970 DATA "WEST VIRGINIA","CHARLESTON","WISCONSIN","MADISON"
980 DATA "WYOMING","CHEYENNE"
```

```
          ****************************************
          * THE 'NAME THE STATE CAPITALS' PROGRAM *
          *                                      *
          *       BY....  WILLARD I. NICO        *
          ****************************************

               HERE IS A QUIZ TO SEE HOW MANY
                STATE CAPITALS YOU CAN NAME.


FIRST, PLEASE TELL ME YOUR NAME....  WILLARD

THANKS, WILLARD!  HERE WE GO.....
I'LL NAME A STATE AND YOU TELL ME ITS CAPITAL

HERE'S THE FIRST ONE.....


WHAT CITY IS THE CAPITAL OF ILLINOIS?  SPRINGFIELD
GOOD GOING, WILLARD!  THAT'S RIGHT!


HERE'S THE NEXT ONE..... KANSAS?  KANSAS CITY
SORRY, WILLARD!  KANSAS CITY IS NOT CORRECT!
TRY AGAIN! - KANSAS?  TOPEKA
IT SURE IS TOPEKA, WILLARD!  KEEP UP THE GOOD WORK!


TELL ME THE CAPITAL OF KENTUCKY?  MONTGOMERY
MONTGOMERY IS THE CAPITAL OF ONE OF THE STATES,
     BUT IT'S NOT KENTUCKY
TRY AGAIN! - KENTUCKY?  FRANKFORT
ANOTHER RIGHT ANSWER, WILLARD!  YOU'RE DOING FINE!
```

EOF

REFLEX TEST

Here is a program that not only will test your reflexes, but, we offer it as a good example of using the real time clock. :JB:

```
1 REM H8 REFLEX TEST
2 REM DENNIS L. SMITH
3 REM 91 WESTPARK DR., OTTAWA, ONTARIO, CANADA K1B-3G4
4 REM WRITTEN FOR 8 H EXTENDED BASIC 10.01.01
5 REM CAN BE RUN WITH 8 H BASIC 10.01.02.
6 REM     REQUIRES 12 K MEM WITH EITHER BASIC
7 REM     DELETING REMARKS PERMITS RUNNING WITH 8 K MEM AND 8 H BASIC
8 REM                        ***************
9 REM CLEAR H9 SCREEN TO PLAY
10 GOSUB 10000 : REM PRINT SOME BLANK LINES TO CLEAR SCREEN
15 GOSUB 10000 : REM THE GAME REPEATS STARTING HERE
20 PRINT "THIS IS A TEST OF YOUR REFLEXES.":PRINT
30 PRINT "I WILL WAIT A WHILE, THEN I WILL PRINT A '*'."
40 PRINT "AS SOON AS YOU SEE THE '*'...PUSH THE SPACE BAR.!"
45 PRINT "(PUSH THE SPACE BAR, WHEN YOU ARE READY TO PLAY)"
48 PRINT "(TO END GAME, TYPE CTRL C, THEN PUSH SPACE.)"
50 PAUSE : GOSUB 10000 : REM WAIT TO START GAME, THEN OPEN SPACE ON SCREEN
60 GOSUB 11000 : REM MOVE CURSOR FOR A RANDOM TIME
65 REM CLEAR CLOCK THEN SET WITH ALLOWANCE FOR PROGRAM OVERHEAD
70 POKE 8219,0:POKE 8220,255
80 PRINT "*";:POKE 8219,252
85 REM WAIT FOR INPUT AND GET TIME FROM CLOCK
90 PAUSE : T=PEEK(8219) : T1=PEEK(8220)
100 PRINT CHR$(13);TAB(40);"TIME : ";INT((256*T1+T)/5)/100;"SEC"
110 GOTO 15 : REM REPEAT THE GAME
9995 REM ********** PRINT 4 BLANK LINES **********
10000 FOR I=1 TO 4 : PRINT : NEXT I : RETURN
10100 PAUSE 200+5000*RND(1)
10110 IF PEEK(8301)=0THEN RETURN
10120 PRINT "DON'T JUMP THE GUN!!!":PRINT TAB(40);:POKE 8301,0:GOTO 10100
10995 REM ********** MOVE CURSER RANDOM DISTANCE AND WATCH FOR INPUT ********
11000 IF RND(1)>.5 GOTO 11060
11005 REM MOVE CURSER ONE FULL LINE
11010 PRINT TAB(78);CHR$(13);
11020 GOSUB 12000 : REM CHECK FOR EARLY INPUT
11050 GOTO 11000
11055 REM MOVE CURSER A PARTIAL LINE
11060 PRINT TAB(78*RND(1));
11070 GOSUB 12000 : REM CHECK FOR EARLY INPUT
11080 RETURN
11995 REM ********** CHECK FOR EARLY INPUT **********
12000 IF PEEK(8301)=0 THEN RETURN : REM CHECK BASIC INPUT BUFFER COUNT
12005 REM A CHARACTER IS IN THE INPUT BUFFER
12010 PRINT "DON'T JUMP THE GUN !!!!"
12015 REM CLEAR THE INPUT BUFFER
12020 POKE 8301,0
12030 RETURN
```

EOF

Vectored from Page 3

```
10 M=24ØØ1

20 DEF FN L(X)=X-256*INT(X/256)

30 DEF FN U(X)=INT(X/256)

1000 ZØ=LEN(Z$)

1010 POKE M,ZØ

1020 FOR Z=1 TO ZØ

1030 POKE M+Z,ASC(MID$(Z$,Z,1))

1040 NEXT Z

1050 Z=M+ZØ

1060 POKE 17267,252: POKE 17268,1

1070 POKE 8192,FN L(M): POKE 8193,FN U(M)

1080 POKE 8212,FN L(Z): POKE 8213,FN U(Z)

1090 Z=USR(Ø)

1100 Z1=Z1+1

1110 PRINT "RECORD #";Z1;"WRITTEN"

1120 RETURN


2000 POKE 17267,177: POKE 17268,1

2010 Z=USR(Ø)

2020 Z$=""

2030 FOR Z M+1 TO M+PEEK(M)

2040 Z$=Z$+CHR$(PEEK(Z))

2050 NEXT Z

2060 Z2=Z2+1

2070 PRINT "RECORD #";Z2;"READ"

2080 RETURN
```

EOF

AN ASCII/SELECTRIC DRIVER for THE HEATH H8 SYSTEM

Howard L. Nurse

IBM Selectric terminals offer the highest print quality for the lowest price in today's personal computer printer market place. The Pacific Office Systems 5-41, one of the most versatile Selectric terminals available, features a standard IBM mechanism, electronic circuitry to provide the control housekeeping required by all Selectrics, a heavy duty power supply, and an RS-232 interface. The Selectric, with its 6-bit code, can be easily interfaced to your computer using software code conversion routines.

This article describes how to interface the Pacific Office Systems 5-41 terminal to the Heath 8080-based H8 computer. A complete assembly listing is provided to allow the terminal to be used as a listing device — it is assumed that your computer system includes a video terminal, such as the H9, to input data. Using the 5-41 as an I/O terminal with the H8 will be the subject of a future article.

The Selectric code used by the 5-41 (and most other "Correspondence" Selectric's) differ quite a bit from the widely used ASCII 7-bit code. Similar to the 5-level Baudot code in several respects, the Selectric code contains 6 bits and requires the use of special upper and lower case characters. While the ASCII code has provisions for a total of 128 characters using 7 bits, the Selectric provides only 64 single-case characters. The use of an upper and lower case increases the number of possible characters to 126.

The look-up table included in this driver assumes a Courier 72 type ball is used. One of the advantages of the Selectric is the ease with which you can change type balls. If you do change type elements, the table must be modified to reflect any differences between the Courier 72 and the type ball you have chosen.

Hardware

The ASCII/Selectric driver was written for the Heath H8 computer system. This program requires that the interface to the terminal be made through an H8-2 Parallel I/O port. Since Heath software makes no distinction between serial and parallel ports, an H8-5 serial port could have been used. However, it is normal to have a CRT console already connected to a serial port, with the second port on the board automatically committed to a cassette tape I/O. If a second serial I/O board were used, the remaining tape I/O port would be useless, unless the board were heavily modified.

A Universal Asynchronous Receiver Transmitter (UART), a hex inverter integrated circuit, a 2152 Hz clock, and an RS-232 driver are required to complete the interface. A schematic and parts list for this outboard circuitry are given in Figure 1 and Table 1, respectively.

The Baud rate potentiometer, R2, should be adjusted to a frequency which is 16 times the terminal Baud rate. Since the Selectric terminal operates at 134.5 Hz, the NE-555 clock rate should be 2152 Hz.

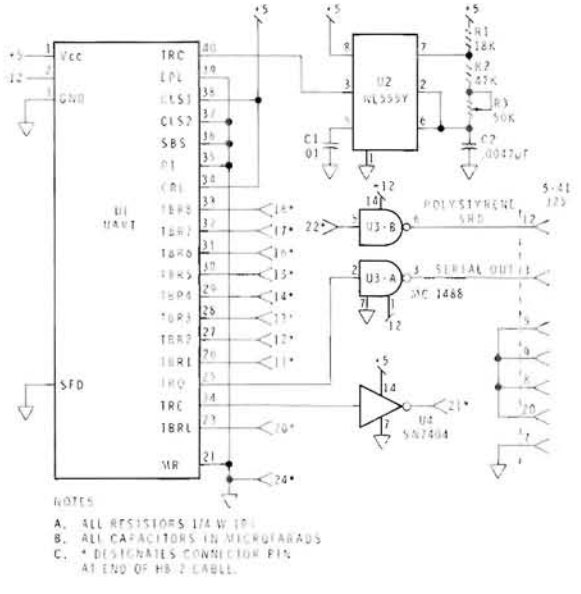


FIGURE 1

INTEGRATED CIRCUITS

U1	AY-5-1013A	UART
U2	NE555V	Timer
U3	MC 1488	RS232 Driver
U4	SN7404N	Hex Inverter

RESISTORS

R1	18K, 1/4 W
R2	47K, 1/4 W
R3	50K, 10 Turn, Potentiometer

CAPACITORS

C1	.01 μFd Disk Ceramic
C2	.0047 μFd Polystyrene (High Stability)

TABLE 1 — INTERFACE PARTS LIST

Tabel 2 shows the connections required between the interface circuitry and the H8-2 Parallel I/O cable. The connector on the end of the cable has been wired as recommended for the H10 Tape Reader/Punch in the Parallel I/O manual. The jumpers on the Parallel I/O channel you use should be configured in the saine way as for an H10 (also listed in the manual). Finally, the software uses I/O port 070Q, which should be established by the program jumpers provided on the board for that purpose.

H8-2 INTERFACE			UART INTERFACE		
FUNCTION	PIN	WIRE COLOR	SIGNAL FLOW	PIN	FUNCTION
Data Ø	1	White-Black	→	11	TBR1
Data 1	2	White-Brown	→	12	TBR2
Data 2	3	White-Red	→	13	TBR3
Data 3	4	White-Orange	→	14	TBR4
Data 4	5	White-Yellow	→	15	TBR5
Data 5	6	White-Green	→	16	TBR6
Data 6	7	White-Blue	→	17	TBR7
Data 7	8	White-Violet	→	18	TBR8
Ground	9	White-Grey		24	GND
Data Taken	10	Red-Black	←	21	TRE
Take Data	11	Red-Yellow	→	20	TBRL
Device Cntrl	12	Red-Green			
Device Rdy	13	White-Black-Red			
Device Cntrl	14	Pink			
Send Data	15	Tan			
Data Sent	16	White			
Ground	17	Grey			
Data 7	18	Violet			
Data 6	19	Blue			
Data 5	20	Green			
Data 4	21	Yellow			
Data 3	22	Orange			
Data 2	23	Red			
Data 1	24	Brown			
Data Ø	25	Black			

TABLE 2 — H8-2 TO UART INTERFACE

Software

The ASCII/Selectric Driver uses a look-up table to convert the ASCII character being sent to your CRT terminal to a Selectric character compatible with the 5-41 terminal. The routine is placed at the upper end of user RAM, just below the stack, where provisions have been included in Heath software to allow user memory space.

Heath programs must be reconfigured to allow the ASCII/Selectric Driver to be used. The changes include:

1. Redefining the high memory limit as detailed in Table 3.

2. Changing 5 memory locations in each Heath program, as shown in Table 4.

1. Load Distribution Tape.
2. Change High Memory using "Reconfigured High Memory" below.
3. Change Pad from 4 to 1 (optional).
4. Record Heath program using "SAVE".
5. Proceed to instructions in Table 4.

TOTAL USER RAM	ACTUAL DECIMAL LIMIT	ACTUAL OCTAL LIMIT	BEGIN EQU	RECONFIGURED HIGH MEMORY
4K	12287	Ø57377A	Ø56ØØØA	11775
8K	16383	Ø77377A	Ø76ØØØA	15871
12K	2Ø479	117377A	116ØØØA	19967
16K	24575	137377A	136ØØØA	24Ø63
2ØK	28671	157377A	156ØØØA	28159
24K	32767	177377A	176ØØØA	32255
28K	36863	217377A	216ØØØA	36351
32K	4Ø959	237377A	236ØØØA	4Ø447

TABLE 3 — RECONFIGURING HEATH PROGRAMS

1. Load tape prepared using instructions from Table 3.
2. Note Stop Address.
3. Change contents of five memory locations using Table below.
4. Re-enter Stop Address noted above.
5. Put new tape in record cassette machine.
6. Push "DUMP".
7. You now have a tape containing a program, the Editor for example, which has a new High Memory point (to protect the ASCII/Baudot Driver), a new Pad, and five changes in the I/O Driver portion to provide jumps into the ASCII/Baudot Driver Routines. Note that this has taken two load and dump procedures.

MEMORY ADDRESS	WAS	CHANGE TO	COMMENTS
Ø4Ø365A	111O	Ø74O	ENTRY LO
Ø4Ø366A	Ø4ØO	"BEGIN" HI*	ENTRY HI
Ø41Ø4ØA	3Ø3O	315O	CALL
Ø41Ø41A	117O	ØØØO	INIT LO
Ø41Ø42A	Ø4ØO	"BEGIN" HI*	INIT HI

*Use high byte of BEGIN EQU from Table 3. For example, if your H8 has 16K of RAM, you would use 1360 at memory locations 040366A and 041042A.

TABLE 4 — MODIFYING HEATH I/O DRIVER

Once your Heath software has been reconfigured, the ASCII/Selectric routine should be loaded with the Heath program. The ASCII/Selectric routine need only be loaded once when going from one Heath reconfigured program to another. For example, the following steps describe a typical use of the driver with the H8 Editor and Assembler.

1. Load ASCII/Selectric Driver from cassette tape.
2. Load reconfigured H8 Editor.
3. Push GO. The Selectric will now print each character sent to the CRT, with substitutions where required (see look-up table).
4. Write and/or edit source code.
5. Save source code on cassette tape.
6. Load reconfigured H8 Assembler.
7. Push GO. The assembly will proceed normally, with all information printed on the Selectric terminal as above.

The program requires approximately 275 RAM locations.

The Driver can be easily modified to be compatible with the amount of RAM in your H8 system. The program as shown in Figure 3 will run with a system having 24K of RAM. If the amount of memory you have differs from this, the BEGIN EQU statement must be modified to reflect your high memory limit in offset octal. For example, if your system has a total of 8K of RAM, the BEGIN EQU statement would be:

```
BEGIN EQU 76000A
```

If you wish to use an output port other than 70Q, the TTY EQU statement must be changed to the new port number.

Once you have defined the changes to be made to the ASCII/Selectric Driver, the source code listing can be changed and the program reassembled.

```
SELECTRIC TYPEWRITER DRIVER FOR H8
INITIALIZATION


                        * SYMBOLIC CONSTANTS

176.000                 BASE    EQU     7E00H   START ADDRESS OF DRIVER
177.000                 TBLBS   EQU     7F00H   START ADDRESS OF TRANS TABLE
040.111                 $CDOUT  EQU     40111A  H8 CONSOLE DATA OUT
040.117                 $CISO   EQU     40117A  H8 CONSOLE INPUT STATUS OUT

                        * CONSTANTS FOR OUTPUT PORT

000.071                 SRD     EQU     39H     SRD LINE OUTPUT PORT
000.045                 SRDRB   EQU     25H     RESET SRD CODE
000.005                 SRDSB   EQU     5H      SET SRD CODE

                        * WRITE MODE INITIATOR

176.000                         ORG     BASE
176.000  365           TTYWR   PUSH    PSW     SAVE REGISTERS
176.001  305                   PUSH    B
176.002  345                   PUSH    H
176.003  325                   PUSH    D
176.004  315 117 040           CALL    $CISO
176.007  323 071               OUT     SRD
176.011  076 100               MVI     A,100O
176.013  323 071               OUT     SRD
176.015  076 116               MVI     A,116O
176.017  323 071               OUT     SRD
176.021  076 045               MVI     A,SRDRB
176.023  323 071               OUT     SRD
176.025  006 074               MVI     B,3CH   SEND EOT
176.027  315 207 176           CALL    CO       IN CASE IN WRITE NOW
176.032  026 040               MVI     D,20H   TIME FOR SRD PULSE
176.034  315 172 176           CALL    TMR
176.037  076 005               MVI     A,SRDSB SET SRD HI
176.041  323 071               OUT     SRD
176.043  026 100               MVI     D,40H   WAIT TTY MODE SWITCH
176.045  315 172 176           CALL    TMR
176.050  006 064               MVI     B,34H   SEND EOA
176.052  315 207 176            CALL    CO
176.055  006 037               MVI     B,1FH   TTY TO LOWER CASE
176.057  315 207 176           CALL    CO
```

```
176.062  076 001            MVI     A,1
176.064  062 223 176        STA     STATUS  SET STATUS TO WRT & LC
176.067  321                POP     D
176.070  341                POP     H
176.071  301                POP     B
176.072  361                POP     PSW
176.073  311                RET
```

SELECTRIC TYPEWRITER DRIVER FOR H8
MAIN WRITE ROUTINE

```
                        * REGISTER A SHOULD CONTAIN THE CHARACTER TO BE OUTPUT

176.074  365          OUTCHR PUSH    PSW
176.075  305                 PUSH    B
176.076  325                 PUSH    D
176.077  345                 PUSH    H
176.100  041 000 177         LXI     H,TBLBS
176.103  006 000             MVI     B,0     CLEAR B FOR DOUBLE ADD
176.105  117                 MOV     C,A
176.106  011                 DAD     B       GET ADDRESS FOR SEL CODE
176.107  106                 MOV     B,M     GET SELECTRIC CODE
176.110  072 223 176         LDA     STATUS  CHECK STATUS
176.113  250                 XRA     B       CHECK IF TTY IN RIGHT CASE
176.114  362 143 176         JP      MA2     YES, JUST SEND CHARACTER
176.117  120                 MOV     D,B     NO, NEED TO SHIFT TTY
176.120  072 223 176         LDA     STATUS
176.123  306 200             ADI     80H     ADJUST STATUS FLAG
176.125  062 223 176         STA     STATUS
176.130  006 037             MVI     B,1FH   SHIFT LOWER CASE CODE
176.132  332 137 176         JC      MA1
176.135  006 034             MVI     B,1CH   NO, SHIFT UPPER CASE
176.137  315 207 176  MA1    CALL    CO      SHIFT TTY
176.142  102                 MOV     B,D     RESTORE CHARACTER
176.143  315 207 176  MA2    CALL    CO      SEND CHARACTER
176.146  026 377             MVI     D,0FFH  SET TIMER FOR RETURN & TAB
176.150  170                 MOV     A,B
176.151  376 055             CPI     2DH     IF RETURN?
176.153  314 172 176         CZ      TMR     YES, WAIT
176.156  376 057             CPI     2FH     IF TAB?
176.160  314 172 176         CZ      TMR     YES, WAIT
176.163  341                 POP     H
176.164  321                 POP     D
176.165  301                 POP     B
176.166  361                 POP     PSW
176.167  303 111 040         JMP     $CDOUT
```

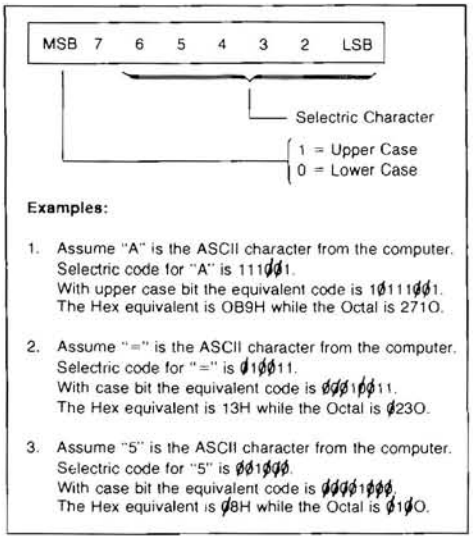
SELECTRIC TYPEWRITER DRIVER FOR H8
TIMER AND CHARACTER OUT

```
                        * REGISTER D CONTAINS THE TIME CONSTANT

176.172  305           TMR     PUSH    B
176.173  006 000               MVI     B,0
176.175  005           LOOP1   DCR     B
176.176  302 175 176           JNZ     LOOP1
176.201  025                   DCR     D
176.202  302 175 176           JNZ     LOOP1
176.205  301                   POP     B
176.206  311                   RET

                        * REGISTER B CONTAINS THE CHARACTER

176.207  365           CO      PUSH    PSW
176.210  333 071       STAT    IN      SRD
176.212  017                   RRC
176.213  322 210 176           JNC     STAT
176.216  170                   MOV     A,B
176.217  323 070               OUT     SRD-1
176.221  361                   POP     PSW
176.222  311                   RET

                        * STATUS BIT
                        *   BIT 0 - SET = TTY IN WRITE MODE
                        *           RESET = TTY IN READ MODE
                        *   BIT 1 - SET = TTY IN UPPER CASE
                        *           RESET = TTY IS IN LOWER CASE

176.223                 STATUS DS       1
```

```
SELECTRIC TYPEWRITER DRIVER FOR H8
ASCII TO SELECTRIC CODE TABLE


                        * 0 - 5 BIT = SELECTRIC CODE
                        * 7     BIT = 1, UPPER CASE

177.000                         ORG     TBLBS
177.000  075                   DB      3DH
177.001  075                   DB      3DH
177.002  075                   DB      3DH
177.003  075                   DB      3DH
177.004  075                   DB      3DH
177.005  075                   DB      3DH
177.006  075                   DB      3DH
177.007  075                   DB      3DH
177.010  075                   DB      3DH
177.011  057                   DB      2FH     TAB
177.012  055                   DB      2DH     LF
177.013  075                   DB      3DH
177.014  075                   DB      3DH
177.015  075                   DB      3DH     CR
177.016  075                   DB      3DH
177.017  075                   DB      3DH
177.020  075                   DB      3DH
177.021  075                   DB      3DH
177.022  075                   DB      3DH
177.023  075                   DB      3DH
177.024  075                   DB      3DH
177.025  075                   DB      3DH
177.026  075                   DB      3DH
177.027  075                   DB      3DH
177.030  075                   DB      3DH
177.031  075                   DB      3DH
177.032  075                   DB      3DH
177.033  075                   DB      3DH
177.034  075                   DB      3DH
177.035  075                   DB      3DH
177.036  075                   DB      3DH
177.037  075                   DB      3DH
177.040  000                   DB      00H     SPACE
177.041  240                   DB      0A0H    !
177.042  211                   DB      89H     "
177.043  260                   DB      0B0H    #
177.044  204                   DB      84H     $
177.045  210                   DB      88H     %
177.046  250                   DB      0A8H    &
177.047  011                   DB      09H     '
177.050  264                   DB      0B4H    (
177.051  244                   DB      0A4H    )
177.052  270                   DB      0B8H    *
177.053  223                   DB      93H     +
177.054  073                   DB      3BH     ,
177.055  067                   DB      37H     -
177.056  021                   DB      11H     .
177.057  007                   DB      07H     /


SELECTRIC TYPEWRITER DRIVER FOR H8
TABLE, CONTINUED


177.060  044                   DB      24H     0
177.061  040                   DB      20H     1
177.062  020                   DB      10H     2
177.063  060                   DB      30H     3
177.064  004                   DB      04H     4
177.065  010                   DB      08H     5
177.066  030                   DB      18H     6
177.067  050                   DB      28H     7
177.070  070                   DB      38H     8
177.071  064                   DB      34H     9
177.072  253                   DB      0ABH    :
177.073  053                   DB      2BH     ;
177.074  264                   DB      0B4H    (
177.075  023                   DB      13H     =
177.076  244                   DB      0A4H    )
177.077  207                   DB      87H     ?
177.100  220                   DB      90H     @
177.101  271                   DB      0B9H    A
177.102  266                   DB      0B6H    B
177.103  272                   DB      0BAH    C
177.104  252                   DB      0AAH    D
177.105  212                   DB      08AH    E
177.106  263                   DB      0B3H    F
177.107  243                   DB      0A3H    G
177.110  246                   DB      0A6H    H
177.111  231                   DB      99H     I
177.112  203                   DB      83H     J
177.113  232                   DB      9AH     K
177.114  206                   DB      86H     L
177.115  241                   DB      0A1H    M
177.116  222                   DB      92H     N
177.117  205                   DB      85H     O
177.120  213                   DB      8BH     P
177.121  233                   DB      9BH     Q
177.122  251                   DB      0A9H    R
177.123  245                   DB      0A5H    S
177.124  202                   DB      82H     T
177.125  262                   DB      0B2H    U
177.126  261                   DB      0B1H    V
177.127  265                   DB      0B5H    W
177.130  242                   DB      0A2H    X
177.131  247                   DB      0A7H    Y
177.132  224                   DB      94H     Z
177.133  264                   DB      0B4H    (
177.134  007                   DB      07H     /
177.135  244                   DB      0A4H    )
177.136  075                   DB      3DH
177.137  267                   DB      0B7H
177.140  075                   DB      3DH     _


SELECTRIC TYPEWRITER DRIVER FOR H8
TABLE, LOWER CASE


177.141  071                   DB      39H     A
177.142  066                   DB      36H     B
177.143  072                   DB      3AH     C
177.144  052                   DB      2AH     D
177.145  012                   DB      0AH     E
177.146  063                   DB      33H     F
177.147  043                   DB      23H     G
177.150  046                   DB      26H     H
177.151  031                   DB      19H     I
177.152  003                   DB      03H     J
177.153  032                   DB      1AH     K
177.154  006                   DB      06H     L
177.155  041                   DB      21H     M
177.156  022                   DB      12H     N
177.157  005                   DB      05H     O
177.160  013                   DB      0BH     P
177.161  033                   DB      1BH     Q
177.162  051                   DB      29H     R
177.163  045                   DB      25H     S
177.164  002                   DB      02H     T
177.165  062                   DB      32H     U
177.166  061                   DB      31H     V
177.167  065                   DB      35H     W
177.170  042                   DB      22H     X
177.171  047                   DB      27H     Y
177.172  024                   DB      14H     Z
177.173  264                   DB      0B4H    (
177.174  007                   DB      07H     /
177.175  244                   DB      0A4H    )
177.176  075                   DB      3DH
177.177  075                   DB      3DH
177.200                         END     TTYWR


STATEMENTS = 00262
FREE BYTES = 17887
NO ERRORS DETECTED.
```

Look-Up Table

The look-up table contains one data byte per ASCII character. Each byte contains the 6-bit Selectric character to be printed plus one control bit. As shown in Figure 4, the most significant bit in the byte is 1 if the character to be printed is upper case.



Examples:

1. Assume "A" is the ASCII character from the computer.
   Selectric code for "A" is 111ØØ1.
   With upper case bit the equivalent code is 1Ø111ØØ1.
   The Hex equivalent is OB9H while the Octal is 271O.

2. Assume "=" is the ASCII character from the computer.
   Selectric code for "=" is Ø1ØØ11.
   With case bit the equivalent code is ØØØ1ØØ11.
   The Hex equivalent is 13H while the Octal is Ø23O.

3. Assume "5" is the ASCII character from the computer.
   Selectric code for "5" is ØØ1ØØØ.
   With case bit the equivalent code is ØØØ1ØØØ.
   The Hex equivalent is Ø8H while the Octal is Ø1ØO.

Acknowledgments

The ASCII/Selectric software presented in this article is based on a similar program written by Kachun Lee for Pacific Office Systems.

EOF

```
Example:
 *  Load  "Sample Program"
       is the equivalent of -
 *  Z$ = "Sample Program"
 *  Load Z$

Sample routines are as follows:

   65,000  REM Tape 'Get' routine
   65,010  POKE 8302,85: POKE 8303,78: POKE 8304,13
   65,020  POKE 8305,71: POKE 8306,69: POKE 8307,90: POKE 8308,36: POKE 8309,13
   65,030  POKE 8310,89
   65,040  POKE 8311,67: POKE 8312,79: POKE 8313,78: POKE 8314,13
   65,050  POKE 8301,13
   65,060  RETURN

   65,100  REM Tape 'Put' routine
   65,110  POKE 8302,80: POKE 8303,85: POKE 8304,90: POKE 8305,36: POKE 8306,13
   65,120  POKE 8307,67: POKE 8308,79: POKE 8309,78: POKE 8310,13
   65,130  POKE 8301,9
   65,140  RETURN

   65,200  REM Tape 'Load' routine
   65,210  POKE 8302,85: POKE 8303,78: POKE 8304,13
   65,220  POKE 8305,76: POKE 8306,79: POKE 8307,65: POKE 8308,90: POKE 8309,36:
           POKE 8310,13
   65,230  POKE 8311,89
   65,240  POKE 8312,67: POKE 8313,79: POKE 8314,78: POKE 8315,13
   65,250  POKE 8301,14
   65,260  RETURN

   65,300  REM Tape 'Dump' routine
   65,310  POKE 8302,68: POKE 8303,85: POKE 8304,90: POKE 8305,36: POKE 8306,13
   65,320  POKE 8307,67: POKE 8308,79: POKE 8309,78: POKE 8310,13
   65,330  POKE 8301,12
   65,340  RETURN
```

PUT

BY ROBERT BEHAR
AND NEAL ROGERS

In order to use the full power of the computer, you need to be able to GET', 'PUT', 'LOAD', 'DUMP', and 'UNLOCK' under program control. In the current version of Heath Extended BASIC (Issue 10.02.01) these are command level statements only. There is a simple way to get around this limitation however.

When the computer returns to the command mode, it does not go to the keyboard of the designated I/O device for its next command, it instead looks at the input buffer.

The input buffer is an area in memory labeled $INBUF by the assembler. Location 8301 decimal holds the count of items in the buffer and the next 30 addresses buffer the keyboard.

If you poke the decimal representation of the correct commands into the buffer, setting the counter to the correct value, the computer will accept the commands as though they had been typed in from the keyboard. (It does not know the difference.)

The correct POKE'S can be put into subroutine and then called as necessary by the program.

NOTE — A string variable can be used instead of a literal to name a file to simplify these routines and keep them flexible.

I suggest that you put all three of these programs on a single tape file. Then load it before you start to write a new program that you will need them in. You can then use them by a simple GOSUB. All four subroutines occupy a total of only 585 bytes.

The attached sample programs and sample runs illustrate the use of these subroutines in admittedly frivolous routines.

GET

EOF

Our thanks to Bob and Neal for this excellent piece on the New BASIC — HC8-13.:JB:

HDOS PRIMER

By Jim Blake

One of the most important things you can do after making the preliminary checks on your H17 is to turn it off . . . walk out of the room and sit down and read the software reference manual from front to back about three times! It is written in a tutorialized fashion, leading you step-by-step on getting your system up and running with the least amount of frustration. If you are unfamiliar with operating systems and you jump right in unprepared, the results will be disappointing, frustrating, a crashed system diskette, and many other things too numerous to predict. I don't mean for this to seem complicated or discouraging, but it is important that you get off to the right start and there is no room in a reference manual for editorializing.

After you have convinced yourself that you understand most everything you've read, you are ready to sit down in front of this marvelous machine and experience a new world of computing. The H8 with cassette based software is fun and educational . . . even useful. The H8 with a disk operating system is a powerful tool.

Now you are ready to follow the step-by-step procedure in the manual. Although, it isn't obvious at the time, one of the first things you are instructed to do is make a back up copy of the distribution disk. This is very important. Because, if you make a mistake and crash the system diskette, your H17 is a useless piece of hardware. Make two copies and file away the original and a back up; then you can afford to make a mistake.

After that somewhat pessimistic opening dissertation, let's talk about some of the exciting experiences in store for you.

First, let's look at the over all picture. The controller electronics for the system drive units resides in the H8 like any of the other boards. Bootstrap ROM resides in a 2 K slot beginning at 030.000 and a 1 K write-protected RAM buffer resides at 024.000. At least 12 K of memory in the H8 is necessary to support the system. You will learn that 24 K is desirable. This piece, like most of the magazine was prepared in the EDITOR. But now, we will escape that and go to the HDOS command mode and you can watch over my shoulder as we put this guy through some of his paces.

```
>CAT

NAME     .EXT     SIZE       DATE            FLAGS    21-MAY-78

SYSGEN   .ABS     14       09-MAY-78         LW
TEST17   .ABS     18       09-MAY-78         LW
INIT17   .ABS     19       09-MAY-78         LW
TXTCON   .ABS     9        01-APR-78         LW
BASCON   .ABS     12       01-APR-78         LW
PATCH    .ABS     11       09-MAY-78         LW
DBUG     .ABS     14       01-APR-78         LW
EDIT     .ABS     16       01-APR-78         LW
ASM      .ABS     27       09-MAY-78         LW
BASIC    .ABS     41       09-MAY-78         LW
DEMO     .ASM     4        01-APR-78         LW
DEMO2    .ASM     6        01-APR-78         LW
DEMO3    .ASM     6        01-APR-78         LW
HDOS     .ACM     2        01-APR-78         LW
DEMO     .BAS     3        01-APR-78         LW

   18 FILES, USING 210 SECTORS (10 FREE)
>CAT SY1:

NAME     .EXT     SIZE       DATE            FLAGS    21-MAY-78

PRUSING  .BAS     4        16-MAY-78
CLASFIED .ED      2        15-MAY-78
PLOT     .ED      1        15-MAY-78
PAGE2A   .ED      5        15-MAY-78
BITS     .ED      16       15-MAY-78
CALLART  .ED      3        15-MAY-78
BIT1     .ED      3        16-MAY-78
DISASEM  .BAS     14       16-MAY-78
CLUBNEWS.ED       1        17-MAY-78
BUGHUG1  .ED      2        17-MAY-78
LETTER   .BAS     5        18-MAY-78
MICRO    .ED      2        18-MAY-78
CONTEST  .ED      4        18-MAY-78
MAG      .ED      1        18-MAY-78
WILEY    .ED      5        20-MAY-78
FIDLEMN  .BAS     34       19-MAY-78
LABEL    .BAS     13       20-MAY-78
CAT      .ED      3        21-MAY-78
FUTURE   .ED      2        20-MAY-78
PAYROL   .BAS     33       20-MAY-78

   20 FILES, USING 155 SECTORS (202 FREE)
```

Here you can see that I asked for a catalog of all files that reside on SY0 and SY1. Unless specified, SY0 is the default device. You notice that on the SY1 catalog, there are no system files at all. This disk was only initialized, given a volume number and label. Therefore, I have almost the entire diskette available for my use. You may want to do the same thing. Let's say that you intend to do all of your work in BASIC. You could create a diskette with the necessary system files to boot up the system and run BASIC only, thus allowing plenty of space on the diskette for your individual programs. Let's do that.

```
>INIT17

DISMOUNTING ALL DISKS:

VOLUME 002  DISMOUNTED FROM SY1
 LABEL: REMARK #3 MANUSCRIPT

VOLUME 126  DISMOUNTED FROM SY0
 LABEL: HDOS 1.0 ISSUE #50.00.00 (COPYRIGHT (C) HEATH CO 1978) 890-1

REMOVE THE DISK(S). HIT RETURN WHEN READY:

INIT17  ISSUE  #50.00.00

        THIS ROUTINE IS USED TO INITIALIZE FLOPPY DISKS.
IT IS A STAND-ALONE UTILITY, AND WILL DESTROY ANY FILES ON
THE DISKS IT INITIALIZES. DO NOT ATTEMPT TO USE THIS PROGRAM UNTIL
YOU HAVE STUDIED THE APPROPRIATE MANUAL.

PROCEED (YES/NO) <NO> ?YES

 INSERT THE VOLUME YOU WISH TO INITIALIZE INTO SY0:;
 REMEMBER, ANY DATA ON THIS VOLUME WILL BE DESTROYED.

HIT RETURN WHEN READY.
READY?

THE VOLUME NOW IN THE DRIVE ...
IS VOLUME # 111
 LABEL = "SYSTEM VOLUME BACKUP
TYPE NO TO CANCEL, TYPE YES TO ERASE AND INITIALIZE THE DISK. (YES/NO) ?YES

ENTER A UNIQUE VOLUME SERIAL NUMBER FROM 1 TO 255: 111
ENTER A VOLUME LABEL OF 60 CHARACTERS OR LESS
USER DISKETTE...W/BASIC AND EDIT

ENTER THE NUMBERS OF THE BAD SECTORS ONE AT A TIME. HIT RETURN
AFTER EACH ENTRY, AND WHEN FINISHED.
SECTOR?
```

REM: We don't know how many bad sectors there are yet . . . we will cover that later. Hit RETURN.

```
 DISK INITIALIZATION COMPLETE.

DO YOU HAVE ANY MORE DISKS TO INITIALIZE (YES/NO) <NO> ?

INSTALL A BOOTABLE DISK IN SY0:. HIT RETURN TO REBOOT:

ACTION? <BOOT> BOOT

SYSTEM HAS 32K OF RAM

VOLUME 126, MOUNTED ON SY0:
LABEL: HDOS 1.0 ISSUE #50.00.00 (COPYRIGHT (C) HEATH CO 1978) 890-1


HDOS VERSION 1.0
 ISSUE # 50.00.00
DATE (30-MAY-78)?
>SYSGEN

SYSGEN ISSUE # 50.00.00

DISMOUNTING ALL DISKS:

VOLUME 126  DISMOUNTED FROM SY0
 LABEL: HDOS 1.0 ISSUE #50.00.00 (COPYRIGHT (C) HEATH CO 1978) 890-1

REMOVE THE DISK(S). HIT RETURN WHEN READY:

INSERT THE SYSTEM DISTRIBUTION SOURCE DISK. HIT RETURN WHEN READY:

INSERT DESTINATION:
```

Here I must insert the disk that we just initialized.

```
INSERT SOURCE:
```

Put the distribution disk in SY0.

```
INSERT DESTINATION:
13 FILES COPIED
```

SEE?! We have a bootable disk.

```
ACTION? <BOOT> BOOT

SYSTEM HAS 32K OF RAM

VOLUME 111, MOUNTED ON SY0:
LABEL: USER DISKETTE...W/BASIC AND EDIT


HDOS VERSION 1.0
 ISSUE # 50.00.00
DATE (30-MAY-78)? 30-JUL-78
>CAT
NAME     .EXT    SIZE      DATE          FLAGS   30-JUL-78

   0 FILES, USING 0 SECTORS (228 FREE)
>CAT/S
NAME     .EXT    SIZE      DATE          FLAGS   30-JUL-78

SYSHELP  .DOC    3         30-MAY-78     SW
HELP     .       2         30-MAY-78     SW
ONECOPY  .ABS    19        30-MAY-78     SW
FLAGS    .ABS    5         30-MAY-78     SW
SET      .ABS    7         30-MAY-78     SW
AT       .DVD    5         30-MAY-78     SW
ND       .DVD    4         30-MAY-78     SW
AT2      .DVD    5         30-MAY-78     SW
PIP      .ABS    17        30-MAY-78     SLW
HDOS     .SYS    24        30-MAY-78     SLW
SYSCMD   .SYS    12        30-MAY-78     SLW
HDOSOVL  .SYS    22        30-MAY-78     SLW
ERRORMSG.SYS     8         30-MAY-78     SW
RGT      .SYS    1         30-MAY-78     SLW
GRT      .SYS    1         30-MAY-78     SLW
DIRECT   .SYS    18        30-MAY-78     SLW

   16 FILES, USING 153 SECTORS (228 FREE)
```

OK . . . but there is still some stuff on there that we don't need, so let's get rid of it and get BASIC on.

```
>DELETE SYSHELP.DOC

ERROR ON FILE SY0:SYSHELP.DOC - ?02 ATTEMPTED WRITE PROTECTION VIOLATION
>FLAGS
```

RIGHT!!! Can't do that, but watch.

```
FLAGS ISSUE #50.00.00.

INSTRUCTIONS (YES/NO) <NO>?

FILE NAME? SYSHELP.DOC
CURRENT FLAGS = SW
 NEW FLAGS:

FILE NAME? ONECOPY.ABS
CURRENT FLAGS = SW
 NEW FLAGS:

FILE NAME?
```

REM: A carriage return in each case removed the flags. Now control D will get us back to the command mode.

```
>CAT
NAME     .EXT    SIZE      DATE          FLAGS   30-JUL-78

SYSHELP  .DOC    3         30-MAY-78
ONECOPY  .ABS    19        30-MAY-78

   2 FILES, USING 22 SECTORS (228 FREE)
```

```
>DELETE SYSHELP.DOC
>CAT

NAME    .EXT    SIZE      DATE          FLAGS   21-MAY-78

HELP    .       2       21-MAY-78
ONECOPY .ABS    19      21-MAY-78

   2 FILES, USING 21 SECTORS (232 FREE)
>DELETE HELP.
>ONECOPY

ONECOPY ISSUE #  50.00.00

        ONECOPY IS USED TO COPY FILES FOR SYSTEMS WITH ONLY ONE
FLOPPY DRIVE. READ THE APPROPRIATE MANUAL BEFORE USING.

DISMOUNTING ALL DISKS:

VOLUME 200  DISMOUNTED FROM SY0
 LABEL: TEST DISKETTE...BOOTABLE ABD WITH BASIC

REMOVE THE DISK(S). HIT RETURN WHEN READY:

INSERT THE INITIAL SOURCE DISK. HIT RETURN WHEN READY:
:0C:BASIC.ABS

INSERT DESTINATION:

INSERT SOURCE:
1 FILES COPIED
:0C:

ACTION? <BOOT> BOOT

SYSTEM HAS 32K OF RAM

VOLUME 111, MOUNTED ON SY0:
LABEL: USER DISKETTE...W/BASIC AND EDIT


HDOS VERSION 1.0
 ISSUE # 50.00.00
DATE (30-JUL-78)?
>CAT

NAME    .EXT    SIZE      DATE          FLAGS   27-JUL-78

BASIC   .ABS    41      21-MAY-78
ONECOPY .ABS    19      21-MAY-78

   2 FILES, USING 60 SECTORS (192 FREE)
>
```

How about that? Now, we know BASIC is there, but if we are running at 300 baud, we don't want to see ONECOPY and BASIC every time.

So let's suppress the listing. Watch.

```
FLAGS

FLAGS ISSUE #50.00.00.

INSTRUCTIONS (YES/NO) <NO>? YES

FLAGS IS USED TO SET AND/OR CLEAR THE FILE FLAGS. WHEN
PROMPTED FOR THE NEW FLAGS, SPECIFY ALL THE FLAGS THAT ARE
TO BE SET. NOTE THAT IF YOU SET THE 'L' FLAG, YOU WILL
NOT BE ABLE TO CLEAR IT AGAIN. THE LEGAL FLAGS ARE:

W       WRITE PROTECT FILE. MAY NOT BE RENAMED, REPLACED, OR DELETED.
S       SUPPRESS NORMAL LISTING OR COPYING OF FILE.
L       LOCK THE FILE FROM FURTHER FLAG CHANGES.

FILE NAME? BASIC.ABS
CURRENT FLAGS =
 NEW FLAGS: SW

FILE NAME? ONECOPY.ABS
CURRENT FLAGS =
 NEW FLAGS: SW
```

```
FILE NAME?
>CAT

NAME     .EXT     SIZE       DATE           FLAGS    27-JUL-78

    0 FILES, USING 0 SECTORS (192 FREE)
```

VIOLA! BASIC and ONECOPY are still on the disk, but we don't have to watch the listing every time we do a CAT.

Let's do some other neat stuff. The same thing with the benifit of two drives.

```
>MOUNT SY1:
VOLUME 200, MOUNTED ON SY1:
LABEL: TEST DISKETTE...BOOTABLE ABD WITH BASIC
>PIP
:P:SY1:EDIT.AB?=SY0:EDIT.*

1 FILES COPIED
:P:
>
```

Here we copied the editor over to SY1: in one painless operation, in about 10 seconds. Notice that I used a 'Wild Card' the '?' and '*'.

We don't have to actually enter PIP . . . We can do it this way.

```
>PIP AT=SY1:BASIC.ABS
```

This caused garbage to be written on the AT (alternate terminal) which is located at port 374. It is very handy with editing material on the H9 and wanting to see a hard copy on the line printer.

Let's optimize our system before we go any further.

```
>SET HELP

SET OPTIONS:

TT: BKS          CRT TERMINAL ALLOWS BACKSPACE CHARACTERS
TT: MLI          MAP LOWER CASE INPUT TO UPPER CASE
TT: MLO          MAP LOWER CASE OUTPUT TO UPPER CASE
TT: BKM          TREAT 'BKSP' CODES (ON INPUT) AS "DELETE" ("RUBOUT")
TT: TAB          TERMINAL CAN PROCESS TAB CODES

        THE ABOVE OPTIONS CAN BE PRECEDED BY 'NO' TO NEGATE THEIR
        EFFECT. (I.E.  SET TT: NOTAB )

TT: 1SB          USE ONE STOP BIT FOR CONSOLE TERMINAL
TT: 2SB          USE TWO STOP BITS FOR CONSOLE TERMINAL
TT: WIDTH NN     SET CONSOLE WIDTH TO NN CHARACTERS
TT: FILL CC NN   PAD OCCURRENCES OF CHARACTER CC WITH NN NULL
                 CHARACTERS

SY: STEP NN      SET TRACK STEP TIME
```

Here we see several options to optimize the system. I have already set backspace and track seek time, but I'll do it again, so you can see (back to the command mode).

```
>SET SY: STEP 08
>SET TT: NOBKS
>DISMOUYT/TY/NT SY1:
VOLUME 200, DISMOUNTED FROM SY1:
LABEL: TEST DISKETTE...BOOTABLE ABD WITH BASIC
>SET TT: BKS
```

This is a very good option . . . you must, however, perform TEST 17 before setting the seek time. Seek time as shipped is about 30 MS. Mine performs with no errors at 8 MS . . . and this saves time.

```
LIST DEMO.BAS
00010 REM
00020 REM                BASIC DEMO PROGRAM
00030 REM
00040 PRINT "HI, I'M A BASIC DEMO PROGRAM!"
00050 PRINT "   I'M GOING TO OPEN THE FILE 'ERRORMSG.SYS' FOR READ"
00060 OPEN "SY0:ERRORMSG.SYS" FOR READ AS FILE #1
00070 REM
00080 REM                LIST THE FIRST 10 LINES TO THE CONSOLE
00090 REM
00100 FOR I=1 TO 10
00110 LINE INPUT #1,A$
00120 PRINT A$
00130 NEXT I
00140 REM
00150 REM        CLOSE THE FILE
00160 REM
00165 PRINT "   NOW I'M GOING TO CLOSE IT ..."
00170 CLOSE #1
00180 PRINT "DEMO ALL DONE..."
00190 STOP
00200 END
 BASIC

EXTENDED BENTON HARBOR BASIC #110.00.00
*CHAIN"DEMO.BAS"
HI, I'M A BASIC DEMO PROGRAM!
        I'M GOING TO OPEN THE FILE 'ERRORMSG.SYS' FOR READ
128CTL-C STRUCK
129CTL-D STRUCK
130DATA EXHAUSTED
131ATTEMPTED DIVIDE BY ZERO
132ILLEGAL NUMBER VALUE
133ILLEGAL USAGE
134DATA LOCK ENGAGED
135CAN'T FIND VARIABLE MENTIONED IN NEXT STATEMENT
136FLOATING POINT OVERFLOW (NUMBER TOO LARGE)
137NO CORRESPONDING GOSUB FOR THIS RETURN STATEMENT
        NOW I'M GOING TO CLOSE IT ...
DEMO ALL DONE...
STOP AT LINE  190
*BYE
SURE?Y
```

Notice some interesting things . . . the ability to open up to five files at a time for READ or WRITE and the CHAIN command which allows to load a program and run it in one command. We could chain several programs together also. For instance, if we had a very large program, and not enough memory to support it, we could break it down into modules, and chain from one to another as we need them, all under program control 'on the fly'. Also, notice the **LIST** under HDOS command — no need to load BASIC or the EDITOR to see what you have.

We were asked earlier, during INIT17, to list bad sectors, if any . . . here's a diagnostic routine that tests our system . . . Watch.

```
:TEST17

DISMOUNTING ALL DISKS:

VOLUME 103  DISMOUNTED FROM SY1
 LABEL: HUG BASIC/EDIT USER VOLUME

VOLUME 000  DISMOUNTED FROM SY0
 LABEL: HDOS 1.0 ISSUE #50.00.00 (COPYRIGHT (C) HEATH CO 1978) 890-1

REMOVE THE DISK(S), HIT RETURN WHEN READY:

TEST17 ISSUE #50.00.00

THIS PROGRAM TESTS YOUR DISK SYSTEM. IT DESTROYS THE
DATA ON THE VOLUME UNDER TEST. THIS VOLUME MUST HAVE BEEN
INITIALIZED AT LEAST ONCE, AND WILL HAVE TO BE REINITIALIZED
BEFORE BEING USED FOR ANYTHING ELSE.

TO PROCEED, TYPE YES; TO CANCEL TYPE CTL-C
PROCEED? YES

WHICH DRIVE (0/1) ? 0

INSERT THE DISKETTE YOU WISH TO USE FOR THIS TEST
INTO DRIVE SY0:, AND HIT RETURN.
 READY?
T - DISPLAY DRIVE ROTATIONAL SPEED
D - PERFORM GENERAL DRIVE CHECKOUT
M - PERFORM MEDIA CHECK (SECTOR VALIDITY)
S - PERFORM SEEK TIME CHECKOUT
U - SELECT ANOTHER DRIVE UNIT
E - EXIT TO BOOT PROGRAM
CTL-C CANCELS THE TEST IN PROGRESS.
 OPTION: S
SEEK TIMING TEST: SEE THE MANUAL BEFORE RUNNING THIS TEST.
PROCEED (YES/NO)? YES

TRYING 16 MILLISECONDS PER TRACK - OK!
TRYING 16 MILLISECONDS PER TRACK - OK!
TRYING 14 MILLISECONDS PER TRACK - OK!
TRYING 12 MILLISECONDS PER TRACK - OK!
TRYING 10 MILLISECONDS PER TRACK - OK!
TRYING 8 MILLISECONDS PER TRACK - OK!
DRIVE PERFORMS RELIABLY AT 8 MILLISECONDS PER TRACK.

FUNCTIONS AVAILABLE:

T - DISPLAY DRIVE ROTATIONAL SPEED
D - PERFORM GENERAL DRIVE CHECKOUT
M - PERFORM MEDIA CHECK (SECTOR VALIDITY)
S - PERFORM SEEK TIME CHECKOUT
U - SELECT ANOTHER DRIVE UNIT
E - EXIT TO BOOT PROGRAM

CTL-C CANCELS THE TEST IN PROGRESS.
 OPTION: M

000 BAD SECTORS LOCATED
```

I see I'm running out of space, but I hope this has been interesting and informative . . . have fun and until next issue . . . . . EOF

--EDIT

WANTED — TWO STOP BITS

When using a device that requires two stop bits at another PORT in addition to your terminal, make the following patch; in BASIC version 10.01.02, change 047.277 to 316. In BASIC version 10.02.01, change location 047.327 to 316. Of course, you will have to add option patch #1 also.

VARIOUS USES FOR PARALLEL BOARD IN H8

A. Interfacing printers to H8-2.

1. TI Silent 700.

2. Anderson Jacobson Selectric Terminal.

1. Reconfigure PORT addresses of board. Example . . . move MSB of address decoders to 200 instead of 300. The PORTS will then respond to 184, 188, 190. (Decimal) The test routines starting on Page 34 of parallel board manual will still function if addresses for status and data are changed accordingly. These changes are necessary to prevent the two I/O boards from conflicting with one another.

2. Cut A1 and B1 jumpers; leave E1 connected to E2. This allows handshaking and puts the data bus at positive true. Take data-pin 11 is normally high, it goes low to indicate good stuff. Data taken-pin 10 is normally high; also it must be pulled low by the printer to indicate that it is ready for the next character. In the case of the TI printer, it was necessary to invert the 'busy line', a user option on the printer. The other output of IC1078 can be used if an inverted take data line is needed. It would then be normally low.

The above steps are all that is necessary to operate a printer off of the H8-2. To operate a hard copy terminal (A&J selectric), it is necessary to add several jumpers to H8-2 and make two small changes to the terminal.

Remove the platen, turn the two hold down screws that hold top cover, fold up tab indicators, remove top cover.

1. On main processor board in printer, remove small .01 cap across pins 10-11 on 74121 located at top right hand side of board, replace with 1 mfd tantalum. This is necessary to increase time of data sent handshake from printer. From 1.4 μS to about 15 μS.

2. Below the same IC there are 6 pins and 2 jumpers, one strapped at 'p' leave this one alone.

   Remove the small jumpers at 'S' and install it at 'T'. This sets up proper handshaking.

3. The parallel board requires a jumper at (desired PORT) E1 and E2, jumper at H1 and H2, and a jumper from INIT3 to whichever PORT you have selected. All of the connections between the two can now be made directly.

With the above addition, you can transfer the console between the H9 and the selectric, PORT=188:LIST would list the program on the printer and return the H9, much the same with PORT=188 :RUN.

PORT=−188 would transfer the console PORT. The only gotcha in this configuration is that the '*' prompt character is an upper case character, and the selectric defaults to lower case upon receiving a '*'; this requires pushing the printer reset button frequently; however, it is still quite usable.

T. Stowe
Store #49D, 61Q, 31H

INTERFACING PRINTER TO H8-2

(Example TI Silent 700)

1. Reconfigure PORT address, example . . . (move MSB of address decoder to 200 instead of 300). PORTS will respond to 184, 188, 190. (Decimal). Test routines in manual for PORT 1, 2 will still function if addresses for status and data are changed accordingly.

   PORT 0 would require a copy of the PORT 1 or PORT 2 routine; these changes are necessary to prevent the two I/O boards from conflicting with one another.

2. Cut A1 and B1 jumpers. Leave E1 connected to E2. This puts the data bus at positive true.

Take data-normally high goes low to indicated good stuff; data taken-normally high also must be pulled low by printer to indicate that it is ready for next character. In the case of the TI printer, it was necessary to invert the 'busy line', a user option on the printer. If it is necessary to have an inverted take data line (normally low), use the other output of IC107B.

This is a small test program used to output data to a specific PORT; a test to see if any lines are inverted on data bus to printer and to see what it prints anyway. See Pictorial 1-3 of H9 illustration manual for data on any specific control character.

T. Stowe
Store #49D, 61Q, 31H

```
10 OUT 189,64
20 OUT 189,78
30 OUT 189,5
40 FOR X=32 TO 127
50 OUT 188,X
60 PRINT CHR$(X)
70 IF X=127 THEN 10
80 NEXT X
```

INTERFACING A SELECTERM WITH A HEATH H8 COMPUTER

1. Check the serial number on the back of the electronics package. If the number is 1150 or higher, go directly to step 3. If the number is below 1150, start at step 2.

2. The ready line for most computers is normally positive going, and this is how our electronic package is wired when you receive it. The H8 computer; however, requires a negative ready which is accomplished by modifying the selecterm electronics as follows:

   A) Using the parts layout sheet supplied, locate integrated circuit A10 (next to the 1702 PROM) and also locate A7 at the other end of the same row. Working from the underside (solder side) of the board, connect a jumper from A10, pin 11 to A7, pin 1.

   B) In a like manner, connect a jumper from A7, pin 2 to the connector J1 (lower left), pin 4.

3. Make the following modification in the supplied cable that goes from the Selecterm electronics to the computer. The modification is made on the blue edge card connector. This will allow the Selecterm to be used with the negative strobe of the H8 computer.

   A) Open the connector and notice that there are two wires soldered to a single pin and that pin also has a jumper going to the adjacent pin. The pin with the two wires soldered to it is designated as pin 6. The bare wire jumper, then, goes from pin 6 to pin 5. The pin adjacent to pin 5 (pin 4) is not used and should have no wire soldered to it. The next pin adjacent to the unused pin is pin 3. Move that wire (from pin 3) to the unused pin (pin 4).

   B) Remove the jumper between pins 5 and 6.

   C) You should now have a wire on pin 4 (the one moved from pin 3) and pin 3 and 5 should be empty.

   D) Now move the wire from the pin directly across from pin 3 (designated pin C) to pin 5.

   E) You should now have two wires on pin 6, one wire each on all other pins except for pin C and pin 3 which should be unused. Check you work for shorts and solder bridges, and then reassemble the connector.

4. The Selecterm is supplied with a male DB25 connector to go to most computers. The H8, however, uses a female connector which goes directly to the parallel I/O board. You could make an adaptor, but this is discouraged as the two extra connectors might degrade the data ALINE and cause errors. The best method is to replace the supplied male DB25 connector with the correct Heath connector, according to the following wiring modification:

MORE INTERFACING IDEAS

On your H8-5 serial card.

Make a solder bridge between pins 12 and 13 of IC122 — improves noise immunity and stability.

Interfacing H8 (H8-5) to a SWTP CT-4.

Install a 4.7 K resistor from the anode of D108 to −18V.

BUGGIN' HUG

Dear Hug,

I have enclosed some pictures of a small audio cassette control box that I have built for use with the H8.

I'll hold them up so you can see them. :JB:

Capabilities include:

> Switching control to permit operation of the cassette player or recorder while the H8 interconnect plug remains in place.
>
> Metering during playback to permit setting and monitoring the playback level during DUMP.
>
> Simple two-jumper-wire interconnect to allow use of one or two audio cassette units without having to remove the Serial I/O board from the H8.

I would be willing to prepare a description, make copies and mail them to any interested individuals who write to me and provide an addressed envelope with a stamp.

This project started with the frustration of having to pull the control plug from the cassette unit everytime that I wanted to rewind. Naturally, in this process I always managed to bump the volume control. Having the Control Box has eliminated this problem and has added the advantage of, due to mounting, protecting the volume control.

Dale Grundon
11456 Links Dr.
Restun, VA 22090

Dear Hug

After just receiving my binder and handbook from you, I'm in agreement with you when you state "more programs will be accepted early in the development of the library than when it is built up." I believe this job may be of use to others and in any event, if rejected, it will give me a better idea of what you're looking for from the user.

I've enclosed a listing and two runs, the first in decimal and the second in octal. I've used the job for getting a complete listing of Extended Benton Harbor BASIC for the purpose of adding commands such as RENname to it. (or RENUmber for renumbering lines for dummies like me who run out of space when adding corrections or modifications). A word about the job. It could have solved the octal computations easier and faster (the IF—THEN statements slow it down) but I choose this format because it preserved the leading and trailing zeros, the simplest and most uniform. True, we could read a listing understanding where the zeros are, but I find it very comforting to see them so I sacrificed the time element. As a point of interest, if you're interested, I ran them off on my Dataspeed 40 terminal KDP at 1200 baud. I note this because I see Heath recommends a speed of 600 baud and I don't know why, I have been running my terminal at the full 9600 baud with absolutely no trouble. I am only forced to run at reduced speed because my printer lacks the required buffer for 9600 and hence gets sick at that speed. The screen loves it.

Thank Heath for me for a great product (they're all great, I own 64 kits myself) and thank you for what I'm sure will be a great User's group.

James P Amoroso D.E.

Jim's dissaembler is in the library — It does not, however, print out the nemonics.:JB:

Dear Hug,

A few weeks ago, just before the first REM was mailed, I wrote a harsh letter to you . . . . now it's time for encouragement.

Your first REM was a good step in the right direction. The spirit of using it to distribute patches for already very effective software (within its specifications) is great. It's a good precedent for the members to share software too. In that spirit, I have enclosed my humble entry in the first annual software contest . . . . this year it's a simple game . . . . imagine the software we'll be seeing in two years . . . maybe in 1980 an APL interpreter will be the winner! . . . . . My entry is fully self-documenting and may be removed from this letter; I regret that without a hard copy device, listings and sample runs aren't feasible.

The IF THEN statement does not relate to the multiple statement per line rule in the same fashion as most BASICS or as the EXT BH BASIC manual says on page 5-45. The rule is that if the IF test fails, the next statement, whether on that line or the next, is processed. In fact, if the test fails, EXT BH BASIC always jumps to the next line. Any statements on the same line as an IF and following it can be reached only if the test condition is true. A patch for this would be ideal,

but at least members should be warned. Otherwise, they will have a tendency to treat that bug as a feature, only to discover too late that their BASIC software is not only not compatible with other BASICS on that fundamental characteristic, but also that the later releases of EXT BH BASIC may not execute IF's in the manner they expect. The safe short term user fix is to not follow IF's with additional statements on the same line.

That's it for now. Thanks again for a good start. I'm looking forward to more and bigger REMs as we users begin to do our part. Thank you.

Dennis L Smith
Ottawa

Bob Morse (Cambridge, MA) writes "... I'm interested in remote temperature sensing for home energy management applications."

Anyone care to help Bob? :JB:

From James McKinley (Winnipeg, Manitoba Canada) . . . "I bought the Heath computer because of the excellent documentation and the convenience of local repairs. My knowledge of computers is limited to looking at the holes punched in the gas bill! Thanks to the local store for their help and patience.

Dear Heath User's Group:

I have some interesting little things for anyone who is interested.

1. To beep the horn in a program or game in BASIC, you can call the horn routine as a USR subroutine. The easiest way is to POKE the horn address into the USRFCN at 103163 by the following:

   ```
   POKE 17267, 94
   POKE 17268, 02
   ```

   A statement such as Z = USR(A) will beep the horn (Z and A = dummy arg).

2. Some games such as Lunar Lander are interesting in real time. By using the keyboard port #250 and a statement such as:

   ```
   A = PIN(250)
   PAUSE(X)
   ```

   A — Must be modified to the appropriate value for the program. In addition, a rubout key must be pressed to clear the keyboard buffer at the end of the run.

3. Using the Plot function of the terminal is also best handled by using the OUT 250, (variable — modified to be 0-128). This keeps the control M, control J and four pad characters from interfering with the plot: for example:

   ```
   00010 PRINT "HIT PLOT ERASE PAGE, AND ANY KEY"
   00015 PAUSE
   00020 R=57.296
   00040 FOR I=1 TO 80
   00050 N=I*10
   00060 X=INT(64*SIN(N/R)+64)
   00070 OUT 250,X
   00080 NEXT I
   00090 END
   ```

4. It is possible to use many USR subs in one BASIC program. By calling a BASIC GOSUB routine that uses the POKE 17267, X: POKE 17268, Y. Where X and Y are the low and high bytes of the machine language subroutine address, a variety of USR routines can be used.

   In conclusion, I would like to see some comments or articles on music synthesis, voice synthesis and recognition and video graphics that anyone may be doing with an H8 system.

   How about an article on control of the horn as in the H8 Demo program for possible music or sound effects using BASIC and possibly a machine language subroutine.

   John Richardson

Be sure to limit high memory in the opening dialogue! :JB:

Dear HUG

I encountered a problem with the serial I/O card for our H11 that caused intermittent transmission errors between the H11 and the H9 video terminal. Using the EIA connections, the negative going part of the pulse train was too small, caused by a loading down of the −8V source. After a second set of parts for this source did not correct the problem (in fact, it got worse), I traced the real problem to Q4. The transistor supplied here (Heath part #417-820) has only marginal gain. I replaced Q4 with a better transistor (MPSA20) and have not had any trouble since.

I also noted an apparent error on P27 of Issue 2 of REMark where a 470 Ω resistor is to be placed between pins 3 and 14 of IC9 of the H11-5 serial I/O card (not between pins 3 & 4). An error in the H11-5 schematic also exists in the wiring of IC24. Pin 14 goes to the +12V buss, while pins 9 and 10 go to +5V.

Wesley E. Swartz
Ithaca

A MAIL LABEL PROGRAM

BY ROBERT BEHAR, WA4HCI

This program was written in Heath Extended BASIC; Issue 10.02.01. The program was written to provide a simple mail label program for the use of the Columbus Amateur Radio Club and to illustrate the 'GET' and 'PUT' command statements of the new basic.

You are allowed to create new label files or maintain existing ones. Under maintenance, you are allowed to print, correct, add, delete, or exit from the program.

The program provides for a maximum of 100 mail labels per file — with an unlimited number of tape files. 'GET' the file you want to work on, then 'PUT' the corrected file on tape. The only major problem is the necessity to 'GET' and 'PUT' in the command mode (and I am working on that). See Bob's article *PUT and GET* on page 17.

The program has been written in a modular fashion so that it may be easily changed, new routines added (a sort will be added shortly) and program maintenance simplified. Since all routines are shared. Whenever possible, you may change major parts of the program by changing very little code. For example, all print routines can be changed by changing the 3000 subroutine only. Call sign could become company name just as easily.

Variables used by the program are:

Z$	=	Answer to interactive questions
I	=	Index to arrays
A$(I)	=	Last Name
B$(I)	=	First Name, Middle initial
C$(I)	=	Call Sign
D$(I)	=	Street name
E$(I)	=	City
F$(I)	=	State
G(I)	=	Zip Code
G$	=	Answer to interactive questions
Z	=	Answer to interactive questions

The 100 labels per file is arbitrary. My computer has 24K; a 16K system would require smaller files. A 32K would allow larger file sizes.

I hope you enjoy this program. Please send any comments to:

Robert Behar
PO Box 6342
Columbia, Georgia 31907

012
015

```
00010 REM LABEL PROGRAM
00020 PRINT "DO YOU WANT TO CREATE A NEW LABEL FILE OR MAINTAIN AN EXISTING FILE"
00030 LINE INPUT "       NEW --- MAINTAIN    ";Z$
00040 IF Z$="NEW" THEN 100
00050 IF Z$="MAINTAIN" THEN 200
00060 PRINT "INVALID OPTION -- PLEASE RE-ENTER":GOTO 30
00100 REM NEW LABEL FILE ENTRY ROUTINE
00110 CLEAR
00120 DIM A$(100),B$(100),C$(100),D$(100),E$(100),F$(100),G(100)
00130 I=1
00140 GOSUB 2000
00150 GOTO 30
00200 REM MAINTENANCE ROUTINES
00210 PRINT "'GET' LABEL LIST THAT REQUIRES MAINTENANCE"
00220 PRINT "       'CONTINUE' TO CONTINUE"
00225 STOP
00230 LINE INPUT "PRINT\CORRECT\ADD\DELETE\EXIT    ";Z$
00240 I=0
00250 IF Z$="PRINT" THEN 400
00260 IF Z$="CORRECT" THEN 500
00270 IF Z$="ADD" THEN 600
00280 IF Z$="DELETE" THEN 700
00290 IF Z$="EXIT" THEN 900
00300 PRINT "INVALID OPTION -- PLEASE RE-ENTER":GOTO 230
00400 I=I+1
00410 IF A$(I)="END" THEN 470
00420 IF I>100 THEN 470
00430 PRINT
00440 GOSUB 3000
00450 PRINT
00460 GOTO 400
00470 PRINT "END OF LABEL FILE. 'GET' NEXT FILE ,THEN 'CONTINUE' TO CONTINUE."
00480 STOP
00490 GOTO 230
```

Just before we went to press Bob submitted the program modification in the box to add a SORT routine — we're glad we could include it. :JB:

```
00500 GOSUB 5000
00510 GOTO 230
00600 I=0
00605 I=I+1
00610 IF A$(I)="END" THEN 640
00620 IF A$(I)="" THEN 640
00622 IF I=100 THEN PRINT "FILE FULL":GOTO 230
00630 GOTO 605
00640 GOSUB 2000
00650 GOTO 230
00700 LINE INPUT "CALL SIGN OF LABEL TO BE DELETED   ";Z$
00705 I=0
00710 I=I+1
00720 IF Z$=C$(I) THEN 760
00730 IF I< 100 THEN 710
00740 PRINT Z$;"  NOT FOUND"
00750 GOTO 230
00760 A$(I)=A$(I+1)
00770 B$(I)=B$(I+1)
00780 C$(I)=C$(I+1)
00790 D$(I)=D$(I+1)
00800 E$(I)=E$(I+1)
00810 F$(I)=F$(I+1)
00820 G(I)=G(I+1)
00830 I=I+1:IF I< 99 THEN 760
00840 PRINT Z$;"  DELETED":Z$=""
00850 GOTO 230
00900 END
02000 REM DATA ENTRY SUB-ROUTINE
02010 LINE INPUT "LAST NAME   ";A$(I)
02020 IF A$(1)="END" THEN 2210
02030 LINE INPUT "FIRST NAME, MIDDLE INITIAL   ";B$(I)
02040 LINE INPUT "CALL SIGN   ";C$(I)
02050 LINE INPUT "STREET NAME   ";D$(I)
02060 LINE INPUT "CITY   ";E$(I)
02070 LINE INPUT "STATE   ";F$(I)
02080 INPUT "ZIP CODE   ";G(I)
02090 PRINT :PRINT
02100 PRINT "IS ADDRESS CORRECT AS PRINTED BELOW?"
02110 GOSUB 3000
02120 PRINT
02130 LINE INPUT "YES ---- NO   ";G$
02140 IF LEFT$(G$,1)<> "Y" THEN GOSUB 5000
02150 I=I+1:IF I < 101 GOTO 2000
02160 PRINT "THIS LABEL LIST FILE IS FULL.   PUT LABEL ON TAPE."
02170 PRINT "ADDITIONAL LABELS MUST BE PUT IN ANOTHER FILE."
02180 PRINT "       'CONTINUE' TO CONTINUE"
02190 Z$="":I=0:STOP
02200 RETURN
02210 PRINT "LABEL ENTRY HAS BEEN ENDED. 'PUT' LABELS ON TAPE."
02220 PRINT "           'CONTINUE' TO CONTINUE"
02230 Z$="":I=0:STOP
02240 RETURN
03000 REM LABEL PRINT SUB-ROUTINE
03010 PRINT B$(I);" ";A$(I)
03020 PRINT C$(I)
03030 PRINT D$(I)
03040 PRINT E$(I);" ";F$(I);" ";G(I)
03050 RETURN
04000 REM FIELD CORRECTION SUB-ROUTINE
04010 PRINT "LAST NAME=1; FIRST NAME, MIDDLE INITIAL=2"
04020 PRINT "CALL SIGN=3; STREET=4"
04030 PRINT "CITY=5; STATE=6; ZIP CODE=7"
04040 INPUT "INPUT NUMBER OF FIELD TO BE CORRECTED   ";Z
04050 ON Z GOTO 4060,4070,4080,4090,4100,4110,4120
04060 LINE INPUT "LAST NAME   ";A$(I):GOTO 4130
04070 LINE INPUT "FIRST NAME, MIDDLE INITIAL   ";B$(I):GOTO 4130
04080 LINE INPUT "CALL SIGN   ";C$(I):GOTO 4130
04090 LINE INPUT "STREET NAME   ";D$(I):GOTO 4130
04100 LINE INPUT "CITY   ";E$(I):GOTO 4130
04110 LINE INPUT "STATE   ";F$(I):GOTO 4130
04120 INPUT "ZIP CODE   ";G(I):GOTO 4130
04130 PRINT
04140 GOSUB 3000
04150 PRINT
04160 LINE INPUT "DO YOU NEED TO CORRECT ANOTHER FIELD IN THIS RECORD?   ";Z$
04170 IF LEFT$(Z$,1) "Y" THEN 4000
04180 RETURN
05000 REM ERROR CORRECTION SUB-ROUTINE
05010 I=0
05020 LINE INPUT "CALL SIGN - LABEL TO BE CORRECTED   ";Z$
05030 I=I+1
05040 IF Z$=C$(I) THEN 5070
05050 IF I< 101 THEN 5030
05060 PRINT Z$;"  NOT FOUND":RETURN
05070 PRINT "LABEL IS AS FOLLOWS:"
05080 PRINT
05090 GOSUB 3000
05100 PRINT
05110 GOSUB 4000
05120 RETURN
```

IF I<100 THEN 5030

```
00230 LINE INPUT "PRINT\CORRECT\ADD\DELETE\SORT\EXIT   ";Z$
00295 IF Z$="SORT"THEN 6000
06000 REM SORT ROUTINE
06010 PRINT "LAST NAME=1; CALL SIGN=2; ZIP CODE=3"
06020 INPUT "INPUT NUMBER OF KEY FIELD ";Z
06021 IF Z<1THEN 6010
06022 IF Z>3THEN 6010
06025 CLEAR H$(:DIM H$(100)
06030 FOR I=1TO 100
06040 IF A$(I)=""THEN 6100
06041 IF A$(I)="END"THEN 6100
06050 IF Z=1THEN H$(I)=A$(I)
06060 IF Z=2THEN H$(I)=C$(I)
06070 IF Z=3THEN THEN H$(I)=STR$(G(I))
06090 NEXT I
06100 N=I-1:I=0
06110 M=N
06120 N=INT(M/2)
06130 IF M=0THEN 6310
06140 J=1:K=N-M
06150 I=J
06160 L=I+M
06170 IF H$(I)<H$(L)THEN 6280
06175 T$=H$(I):H$(I)=H$(L):H$(L)=T$
06180 T$=A$(I):A$(I)=A$(L):A$(L)=T$
06190 T$=B$(I):B$(I)=B$(L):B$(L)=T$
06200 T$=C$(I):C$(I)=C$(L):C$(L)=T$
06210 T$=D$(I):D$(I)=D$(L):D$(L)=T$
06220 T$=E$(I):E$(I)=E$(L):E$(L)=T$
06230 T$=F$(I):F$(I)=F$(L):F$(L)=T$
06240 T=G(I):G(I)=G(L):G(L)=T
06250 I=I-M
06260 IF I<1THEN 6280
06270 GOTO 6160
06280 J=J+1
06290 IF J>K THEN 6120
06300 GOTO 6150
06310 PRINT "SORT COMPLETE "
06320 GOTO 230
```

EOF

CLASSIFIED

FOR SALE: HEATH H8 COMPUTER SYSTEM

H8 mainframe, 16k memory, Serial I/O Cassette board, Tape recorder, SWTP CT-64 terminal w/rf modulator, all boards have w/w terminals, latest software, Heath BASIC programming course, ASSEMBLED AND RUNNING — ASKING $1200

Bob Bileski — 201-368-9300 — 8-4PM M-F

Ad placement is free to all members. Send your ads typewritten to use. Allow approximately 10 weeks for the ad to appear. We reserve the right to reject any ads that do not serve the best interest of Heath Co., HUG or its members.

MEETINGS and CLUB NOTICES

GREENSBORO, NC

I would like to meet other Heathkit H8 owners in my area. I will be glad to help anyone who is having trouble either in construction or programming the H8. I would like to exchange programs and ideas with anyone interested, and would be glad to give demonstrations to non-owners.

If interested, contact:

Hughes B Hoyle III
716 S Elam Ave.
Greensboro NC 27403
(919) - 378-1050

NORTHBROOK, IL

Contact Bob Kaplow, 1644 Fendale, Northbrook, IL 60062 . . . He is interested in H11 user's group.

SIMSBURY, CT

Our Computer Club is open to anyone interested in computers and our regular meetings are on the second Thursday of every month. Some of our meetings are held at various computer installations such as Connecticut General Life Insurance Company, June 8; Wethersfield Computer Center (Town Hall), July 13; Talcott Mountain Science Center, August 10; and other locations that have some impact on Electronic Data Processing.

For further information contact Harald Bender, The Computer Club, 6 Maureen Drive, Simsbury, CT 06070.

LOVES PARK, IL

". . . I have become President of the Rockford, Illinois "Blackhawk Bit Burners" Computer Club. We meet every second Wednesday night 7 PM at Barber Colman Company, 1354 Clifford Avenue, Loves Park, Illinois. We have over 40 members and offer our help and friendship to all who visit. There is also a Chicago and Madison, Wisconsin group — not related."

Frank D Dougherty

ROCKFORD, IL

Please print in your next REMark that we have an active 40 member computer club going in the Rockford, IL area. I have a Heath ET-3400 trainer and course and am President of the club. Persons interested can phone evenings or write, enclosing a SASE for details. I'm impressed with the wealth of helpful programs and do it yourself information in REMark. It's worth the cost of the membership!

In a month or two, I should have an ET-3400 program to submit. Meanwhile, REMark is great!

Frank Dougherty
Blackhawk BIT Burners Computer Club
325 Beacon Dr, Belvidere, IL 61008
1(815) 544-5206

AURORA, IL

Just talked with Bill Lord on the phone . . . He's anxious to get together with other H8 users in the Aurora, Ill. Area and swap ideas. Give Bill a call at 892-6573. Keep in touch, fellows.

MONTREAL

HUM . . . That's the name of a new user's group up North. President Bernard Tremblay writes . . . "With the help of George Girad and Micheal Caldarola from the Montreal Heath Electronic Center, H8 user's have recently formed a club.

. . . We are planning to hold our meetings on the second Wednesday of each month. At this stage we plan to divide the meetings into segments dealing with the minutes of club affairs -- interesting lectures -- and a period of questions and answers.

We hope to hear from other H8 clubs soon.

Contact HUM through Bernard Tramblay, President, 250 Oak St. Rosemer, J7A 3K6 Quebec. Good computing fellows! :JB:

SEATTLE

Martin Lindal wants to get a local user's group started in the Seattle area. Contact Martin at 10411 Empire Way, PO# 78006, Seattle, WA 98178

Popular method with complete documentation on how to swindle your local bank out of many K bucks using a homebrew computer and Model 77 Gotcha Modem. Send no money, just two dozen chocolate chip cookies to: Fred Biten, Federal Prison.

BITS AND NIBBLES

SO YOU WANT TO BE A WRITER?!

We have already received some very good articles that will appear in future issues of REMark. This is your club and your magazine and if you don't 'PUT' . . . you don't 'GET'. Well, it isn't quite that bad, but we would like to hear from you . . . What are you doing with your computer? And how? We will take anything from notes on a grocery bag to typewritten manuscripts. (Double space and leave 2" margins). Clearly mark all drawings and photos (only black and white).

HUG SOFTWARE LIBRARY

The Heath User's Group Software Library has been established and is growing every day! By now, you should have received the catalog which contains about 100 programs. It is a loose leaf catalog suitable for keeping in your leatherette binder which allows us to easily supplement it periodically.

As a greater benefit to the membership, we have selected a unique procedure for distributing the software. We have prepared a book which includes the complete listing and documentation of every program in the library. Volume I is available to all members for the reproduction and handling costs of $9.00. The P/N is 885-1008. A cassette of all the H8 programs is also available for $7.00. Its P/N is 885-1009. All orders for software must be accompanied by the proper amount, P/N and submitted on the green order form.

TAKING INVENTORY

As you read this, HUG is about 8 months old. And, as with any new organization, we experienced a few start-up problems, but we think we've worked out the 'bugs'.

The "Blizzard of '78" gobbled up hundreds of the first issue of REMark and initial handbooks so many of you were sitting out there waiting for something to happen. We apologize for that, but were unaware of it until we started hearing from you.

Since the first of May, I've enjoyed the position as your manager. Don't hesitate to call or write when you have a question or suggestion. You can expect a prompt response.

Just received an addendum to the 'Fox Hill Farms Guide to the H8 Computer System'. William 'Doc' Campbell has prepared an excellant guide to creating and maintaining a mailing list program for the H17 Floppy Disc. Doc's commentary is clear, concise and very helpful to the new computer operator. You may order it for the price of a stamp . . . just drop a note to Sue, here at the HUG office. Tell her you want 'Addendum Number Two' to the Fox Hill Farm Guide. Incidentally Doc operates a large "pick your own" vegetable farm down in Elam, Glen Mills, PA. Stop in and see him.

On page 18 of REMark #2 we published a 'Lower to Upper Case' mod for the H9. Here's an easier way to accomplish the same thing from George Frye, President of Frye Electronics in Tigard, Oregon.

On the character generator circuit board . . .

1 — Cut the foil trace between IC203-15 and IC206-1
2 — Solder a wire from IC203-10 to IC218-11
3 — Solder a wire from IC-219-10 to IC206-1
IC-219-11

IC219 is a hex inverter with one section not used. We found this to be a simple and useful modification . . . Thanks George. . :JB:

An article in a recent issue of Interface Age described a method of modifying the H9 to display 24 lines. This article has created considerable inquiries here at HUG. We modified an H9 here in the office . : It took most of the day between phone calls and didn't work! Our mistake. Three errors! My opinion? . . . 24 lines is nice. However, this limits interline spacing to only two scan lines and it is a bit hard to read.

No . . . Heath does not plan to offer a similiar conversion.
No . . . REMark will not elaborate on the modification. If you take your time, you will have no problem.

PC '78

Billed as "a major convention for personal and business computing" Personal Computing '78 will be held at the Philadelphia Civic Center August 24 through the 27th. Four big days! John Dilks, Show Director, told us that the opening session will feature a full-day industry trade show. Special meetings and seminars are planned for the first night. He expects as many as 300 exhibitors with other attractions such as an art show, music festival, computerized mouse maze and personal Computing College TM which includes over 80 hours of free in-depth seminars conducted by some of the country's leading experts in the business.

PC '78 will also mark the first nation-wide gathering of the Heath User's Group. Details of our meeting will be available by separate mailing and at the show.

I'll be looking forward to meeting you there.:JB:

THE

BACK PAGE —

NEW STORES

Just got the word of some new Heath Electronic Centers opening near you members in Salt Lake City . . . St. Paul, and Oklahoma City. Of course, they will carry the full line of Heath computer products, and books and have a staff to help solve any problems that may pop up. Managers and exact locations were not known at this writing, but stop in, say hello and browse around.

Opening date for St.Paul is late August — Salt Lake City and Oklahoma City, early September.

OMAHA

Meet your new Manager, Roger Svoboda. He is a graduate of Franklin University in Columbus, with an EE degree, and is a real computer buff.

HUG ME

HUG tee shirts, similar to that shown on the front cover, (except the Heath User's Group logo will appear in the upper left portion) are available through the parts department for $4.50 each . . . use your green order form to order and enclose check, cash or money order for the correct amount. They are available in blue in three sizes. Order by part number. Small—885-1100. Med—885-1101. Large—885-1102.

SOFTWARE CONTEST #3

Your experience is showing!

The third software contest is three contests in one! A separate contest for H11 paper punchers, H8 operators, and ET-3400 number crunchers; open to all HUG members. Programs may be submitted in any category.

Here again are the rules.

Entries may be submitted which run on any Heath computer product which does not require hardware modifications . . . submit a listing of the program, if possible and a tape, if applicable.

Entries will be judged on:

Usefulness and popularity
Program stability and compactness
Ease of use
Completeness of work

All winners will be determined by a committee at HUG. All decisions will be final. Schlumberger/Heath employees and relatives are not eligible for prizes. All entries will be considered for inclusion in the HUG library.

. . . and here are the prizes.

H11 . . . . . .GD-1186 Digital Scales
H8 . . . . . .H-17 Floppy Disc
ET-3400 . . . . . .GD-1186 Digital Scales

So get those one's and zero's organized and send them in before October 1, 1978. Winners will be announced by separate mailing. Please mark all materials submitted with "CONTEST #3".

and that is . . .the last word :JB:

Heath
User's
Group
Hilltop Road
St. Joseph MI 49085