REMarks
On Heathkit Computing

Issue 5 ● 24 January 2022

This is a community-supported publication to capture and share knowledge and experiences with Heathkit computers. Each issue will address one or more topics of interest to collectors, hobbyists and computer historians. If you have material or topic ideas to contribute, please contact me via the SEBHC Google Group at https://groups.google.com/g/sebhc

- Glenn Roberts, Editor.

Survival Guide for a Diskless System

By Glenn Roberts

If you have an H8 or H89 without a disk drive, you may feel like there's little you can do with your system other than watch the blinking lights. Many hobbyists today have expanded their systems with high-capacity floppy drives, GoTek [1] or SVD drives [2], and hard drive emulators such as the Z67-IDE+[3], but if you're stuck without a disk drive how can you create and run programs to show off your system?

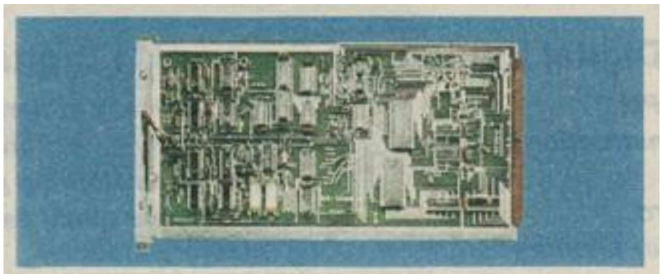
Remember that when the H8 first came out in the Fall of 1977 [4] the available "mass storage" options were cassette drive and paper tape. It wasn't until a year later that Heath introduced the H17 floppy disk system, and that was an expensive option, so many hobbyists continued to rely on cassette storage for some time (while they saved up their pennies for an H17!).

Today cassette tapes and drives are very much bygone technology and relying on older tapes and drives may be asking for trouble. Even in their heyday cassette tapes could be an unreliable storage medium and users were encouraged to make multiple copies. Nevertheless, Heath developed a simple and elegant approach to file storage on tape, and with a few enhancements these tools can now be used to store and retrieve files digitally with high reliability, and without the need for a disk drive!

While my discussion will center on the H8, the H89 had a cassette interface capability and much of this material should be applicable there as well.

The H8-5 Serial/Cassette Interface

When the H8 was first introduced Heath included a dual-purpose I/O interface board called the H8-5:



H8-5 Serial I/O and Cassette Interface. Connects the H8 to serial devices such as the H9 video terminal (page 9) or the H36 DEC Writer II (page 10). Features jumper selectable data rate from 110 to 9600 baud, common input/output interfaces including 20 mA current loop and EIA RS-232C compatible levels. The cassette recorder interface permits use with the Heathkit ECP-3801 cassette recorder/player (page 10). Uses the popular Byte/Manchester of "Kansas City" standard recording format. Control lines for remote start and stop of two cassette units allow separate record and playback for easy program or file editing. Also has full interrupt capability.

Kit H8-5, Shpg. wt. 3 lbs. **110.00**

This board provided an RS-232C (or, optionally, 20mA current loop) interface to service a system console or teletype, but it also included an interface for cassette storage and retrieval. Both channels used Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) integrated circuits [5], with the cassette channel having the appropriate circuitry to convert the analog tape signal to/from digital form via the popular "Kansas City" recording standard [6]. The cassette channel stored and retrieved data at 1200 BAUD. The board contained reed relay switches to activate the drive motor on the cassette player and could accommodate two cassette drives – one for read and one for write operations (e.g., for use in file editing or assembling source code).

Heathkit also sold a paper tape/punch system, model H-10, which used the same software interface but was physically interfaced via the H8-2 parallel board.

To make loading and executing software as simple as possible the Heath engineers built a feature into the Panel Monitor (PAM) ROM software that allows the user to load software from tape (paper or cassette) with a single button push (the "LOAD" button, key '8' on the keypad). This one-button load function worked only via the H8-5 cassette interface, or the H8-2 parallel interface and H-10 paper tape/punch. Similarly, a “DUMP” option allowed for writing data to tape via key ‘9’ on the keypad.

Heath’s Tape “Operating System”

Concurrent with the introduction of the H8, Heathkit produced a suite of software to support tape-based operations. The software included a text editor (TED-8), assembler (HASL-8), debugger (BUG-8), and Heath’s implementation of the BASIC language marketed as “Benton Harbor” BASIC (Heathkit headquarters was in Benton Harbor Michigan). The software could be run on a system with as little as 8K of RAM (at the time 4K RAM boards were $140 and could be upgraded to 8K for an additional $95). For an additional $10 you could get Extended BH BASIC, but the real cost came in hardware: you needed to expand to 12K or, preferably, 16K. Extended BH BASIC added support for strings and advanced functions.

This tape-based operating environment was not given a marketing name or even sold separately – in the beginning it was simply included with each H8 computer (on cassette media). It is documented in “Software Reference Manual” [7], which is archived on the SEBHC site:

https://sebhc.github.io/sebhc/documentation.html

The software itself has also been preserved on the SEBHC site:

https://sebhc.github.io/sebhc/software.html#Cassette_Tape_Images

in the form of “H8T” files. These files contain the exact digital contents of the various utilities just as they would have been read off the cassette- or paper-tape interface. The tape file format is documented starting on page 0-12 of the Software Reference Manual.

Loading and Running Software

If you are fortunate enough to have the original tape media and a working cassette player (or paper tape and an operable H-10) you could load and run the original software for the H8 and create your own BASIC and assembly language programs. More likely, you have an operable H8 system with an H8-5 board, but no media.

If the audio from the cassette tape has been digitized, for example in .WAV format, it is possible to play it back with a laptop or tablet, feeding the headphone jack output directly into the H8-5 input jack, however I am not aware of an archive of the tape operating environment in .WAV format [8].

So how then do you load the .H8T digital files into the computer? I will discuss two approaches.

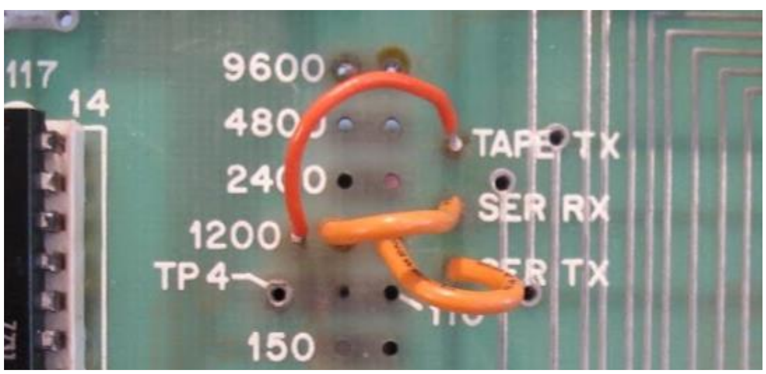
The Port Interchange Approach

The first approach makes use of the "port interchange" switch on the H8-5 board [9]. Heathkit provided this switch primarily to support teletype devices with embedded paper tape punch/readers [10]. The 8251 USART that services the console is normally addressed at port 372/373 (octal) and the cassette port at 370/371. The port interchange switch simply interchanges the two addresses. In the "NORM" position the USARTs are addressed at their correct port assignments, but in the "PORT INTCHG" position the LOAD and DUMP functions will operate over what's normally the console channel.

Using a teletype setup, one would select the "PORT INTCHG" mode, hit the "LOAD" key on the H8 and then activate the paper tape reader to download the software. Once the software was loaded the user could go back to "NORM" mode and use the teletype as a console device, however since this necessitated removal of the H8 cover to access the switch, Heath provided patches to all their programs to switch console I/O to the 370/371 port, allowing the user to leave the system permanently in "PORT INTCHG" mode.

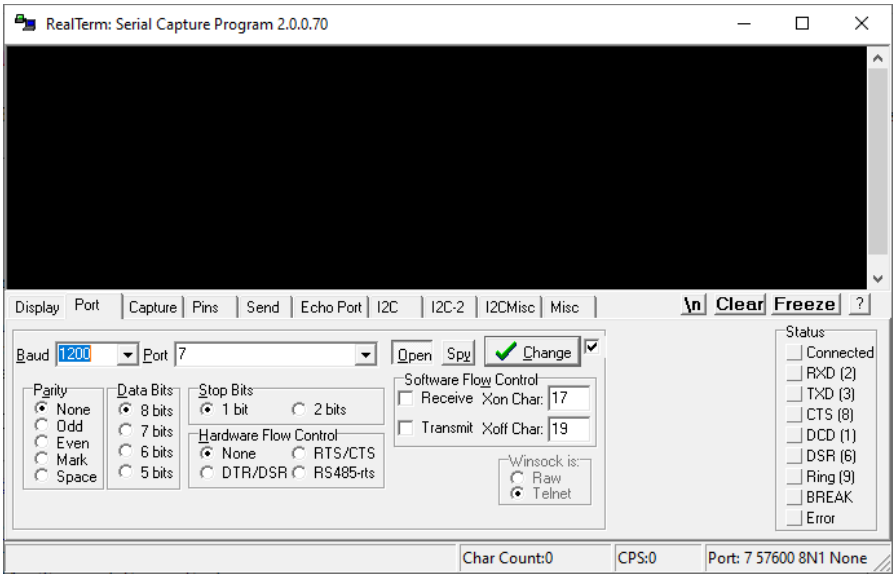
To load the H8T files we will simulate this "teletype" mode of operation using a serial communication package on a standard personal computer or laptop.

Step 1. Since we will be switching between the console and cassette ports, they must both be set at the same BAUD rate, namely 1200 BAUD (required by the cassette interface). Set the appropriate H8-5 jumpers as shown:



Step 2. If you don't already have the RealTerm software [11] on your PC or laptop, download and install it from here: https://realterm.sourceforge.io/

Step 3. Run RealTerm, select your serial port and configure it for 1200 BAUD, 8 Bits, 1 Stop Bit, No Parity (be sure and select the green check box marked "Change" to place your changes into effect). If your PC has no serial port, RealTerm will work with USB-to-serial converters such as the Sabrent CB-DB9P or equivalent.

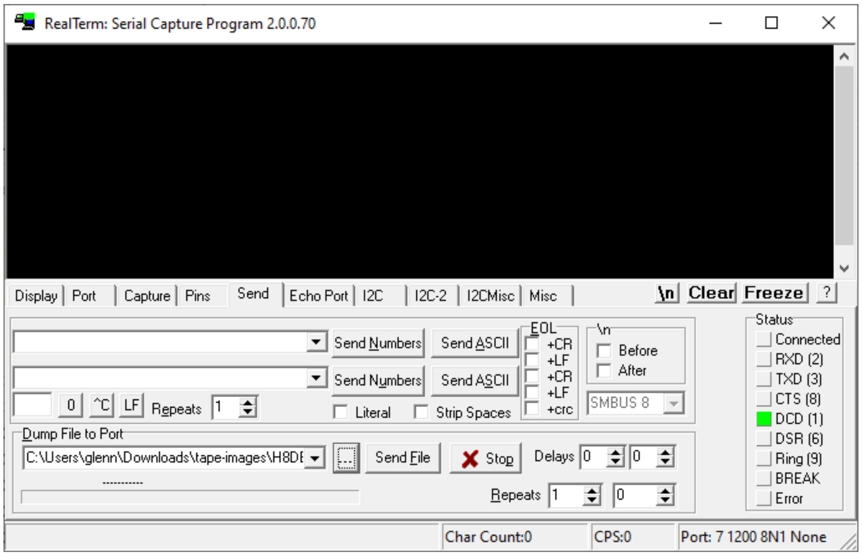


Step 4. Using a WH 8-51 adapter cable [12] or equivalent connect the PC serial port to the serial (console) port on the H8-5. You may have to fashion the appropriate cable and connectors using the Heath documentation [9, 12]. With the port interchange switch set to "NORM" (to the right) you can test that your connection is working using the following steps (from the H8-5 manual) [9]:

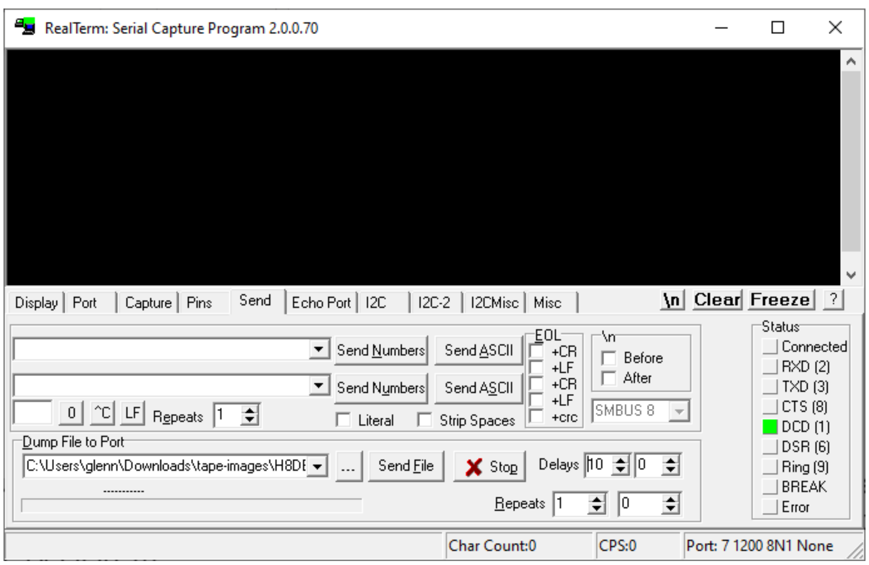
- Simultaneously press the 0 and RST/0 computer keys to reset the H8.
- Press the MEM key.
- Enter 316 373.
- Press the OUT key.
- Press the MEM key.
- Enter 005 373.
- Press the OUT key. (Your USART is now configured.)
- Press the MEM key.
- Enter 101 372.
- Press the OUT key. (The letter "A" should appear in the RealTerm console. You can continue to press OUT and each time an "A" should be transmitted.)

Step 5. Download and save the tape images from the SEBHC site:
https://sebhc.github.io/sebhc/software/tapes/tape-images.zip

Step 6. For starters we'll load the classic H8 demo (from the original H8 operations manual). The file is called H8DEMO.H8T. In RealTerm select the "Send" tab and in the "Dump File to Port" frame select the small square button with the ellipsis ("…"), then navigate to the folder with the H8T files and select H8DEMO.H8T:



Step 7. Because there is no handshaking on the port it's best to insert a small delay after each byte to avoid any data loss. I find 10 ms. to be acceptable. The leftmost of the boxes marked "Delays" specifies the per-byte delay. Set it to 10:



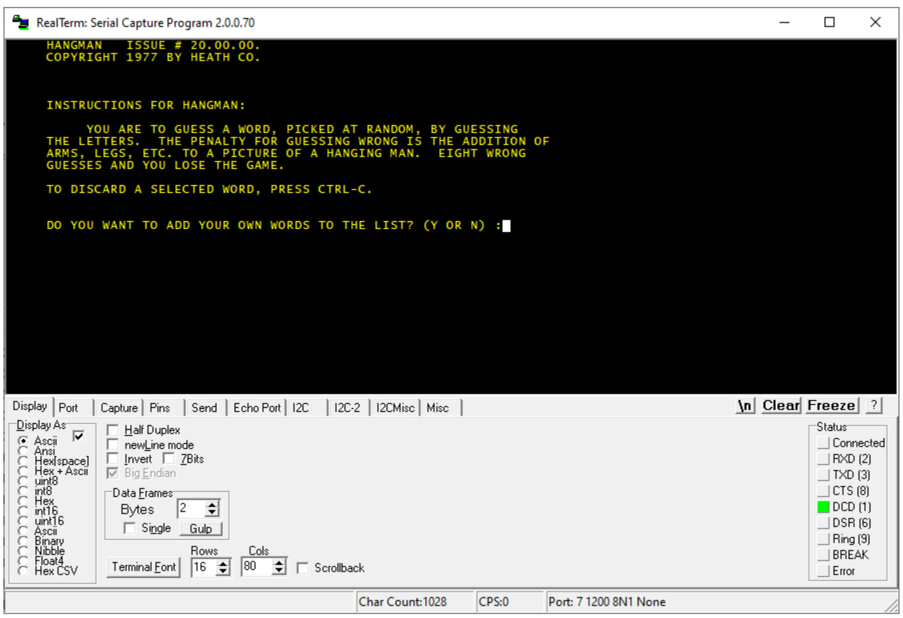
Step 8. Set the port interchange switch on the H8-5 board to "PORT INTCHG" (toward the left). Now data sent over the RS232 console wire will be routed to the cassette interface USART. If your H8 is not already powered on turn it on now.

Step 9. Select the "LOAD" command (key '8') from the PAM-8 keypad on the H8. There will be no change in the H8 LED display. The H8 is now waiting for data on port 370.

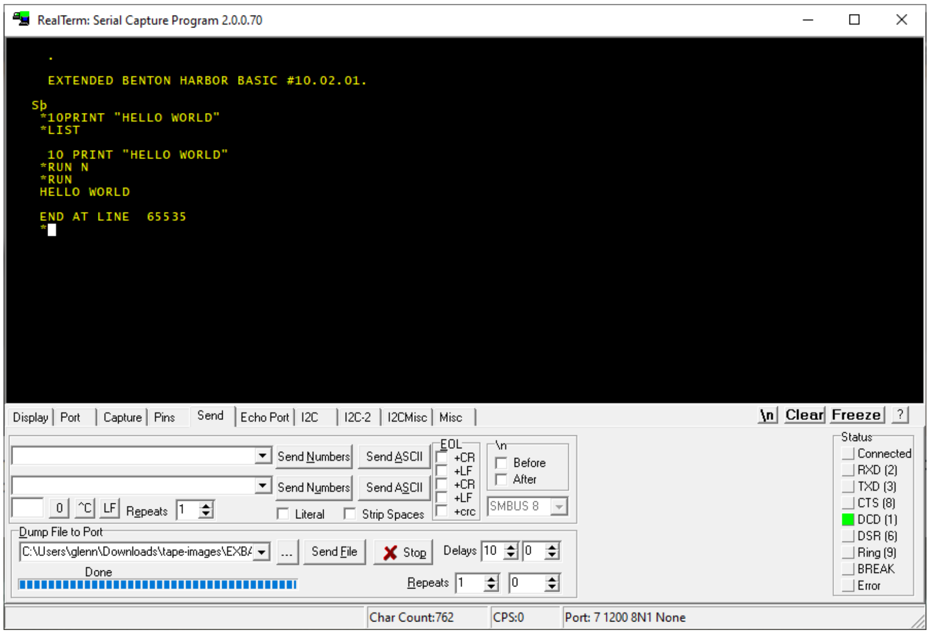
Step 10. In RealTerm click on the "Send File" button. If all is working you will see activity on the H8 LEDs followed by a single beep, indicating successful download. The program is short so this will not take long. The program is now in memory and the PAM-8 software has pre-loaded the Program Counter (PC) register with the start address.

Step 11. If the downloaded program were to make use of console I/O you would switch the port interchange switch back to “NORM” at this point. Since the H8 demo program uses only the LEDs that’s not necessary this time. Hit the “GO” key (‘4’ on the keypad) to start the program. You should see the familiar “your H8 is up and running H H H” on the LEDs, followed by a double beep from the speaker.

You may want to repeat the process with another program for practice. Try loading the Hangman game (H8HANGM.H8T). This time after the download you’ll have to change the port interchange switch back to the “NORM” position (Step 11) since the Hangman program performs console I/O.



You should now be able to download and run any of the Heath software tools, for example try loading and running EXBASIC2.H8T. This download will take a while. The front panel LEDs display the address of RAM being populated at any moment. It will count up from 040.100 to 112.175. RealTerm also shows a progress bar to track the download status.



Beware on these early versions of BH BASIC there was an "autocomplete" function that would automatically type the rest of a BASIC keyword once it has enough letters to recognize it. I'm sure Heath had good intentions, but this "feature" is quite annoying. You must constantly watch the screen and essentially learn an abbreviated BASIC syntax. You can see above that I typed "RUN" but as soon as I had typed "RU" the BASIC interpreter supplied the "N" so my extra "N" confused things. Somewhere (REMark? Hscoop?) I recall seeing a patch that disables this but I have been unable so far to find it.

If all you want to do is occasionally download and run an old program this method may suffice. It is annoying, however, to have to change the port interchange switch. I also have not found a satisfactory way to save programs using this approach. You must type the command to save your data (e.g., in BASIC you'd use "DUMP") and then immediately switch the port interchange switch and set the RealTerm program to capture mode. You are also restricted to running the console port at 1200 BAUD. Fortunately, there's a hardware approach that adds a "daughterboard" to the H8-5 and provides a more elegant solution.

The Daughterboard Approach

Back in 2016 Dave Runkle posted a description on the SEBHC Google Group of a "daughterboard" design he did for the H8-5
https://groups.google.com/g/sebhc/c/WXGm3Qmr5SM/m/QZ88sGRoCAAJ

He has documented this on his site
http://www.astrorat.com/heathkit/heathkith8computer.html

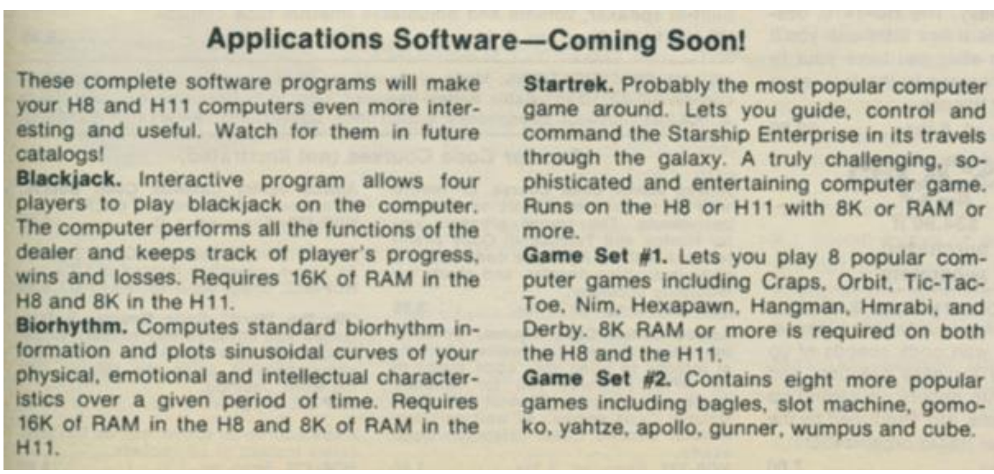
You can read the details online but in summary Dave's solution provides an RS-232 interface to the cassette port USART that allows you to use your PC or laptop to simulate the tape function (as we've done above) but without the need to switch back and forth between console and cassette. Your console (e.g. H9 or H19 terminal) can be left alone and can be run at any supported BAUD rate. Dave even provides for a switch to allow you to revert to cassette operation, should the need arise.

Dave's techniques and instructions are fully documented on his site. Though he is not in the business of supplying these boards there is a possibility of fabricating new ones (watch for a discussion on this topic on the SEBHC Google Group).

I have successfully used Dave's daughterboard via the RealTerm approach he documents; however, it would also be an interesting project to replace the PC with a stand-alone box (e.g., based on an Arduino, Raspberry Pi, or similar small form factor computer) that replicates the tape player functionality. You could even house such a device in an old, gutted cassette player housing!

The Heath Users Group (HUG)

In the Fall 1977 catalog [4] where Heath introduced the H8 and H11 computers there's an interesting teaser about coming entertainment software:



Applications Software—Coming Soon!

These complete software programs will make your H8 and H11 computers even more interesting and useful. Watch for them in future catalogs!

Blackjack. Interactive program allows four players to play blackjack on the computer. The computer performs all the functions of the dealer and keeps track of player's progress, wins and losses. Requires 16K of RAM in the H8 and 8K in the H11.

Biorhythm. Computes standard biorhythm information and plots sinusoidal curves of your physical, emotional and intellectual characteristics over a given period of time. Requires 16K of RAM in the H8 and 8K of RAM in the H11.

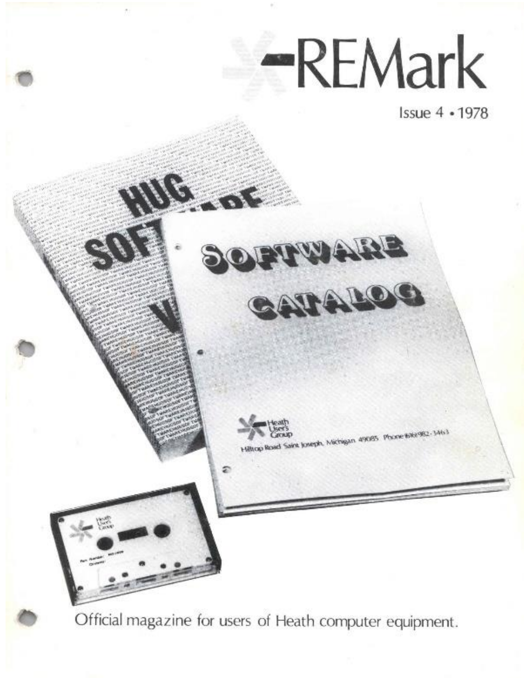
Startrek. Probably the most popular computer game around. Lets you guide, control and command the Starship Enterprise in its travels through the galaxy. A truly challenging, sophisticated and entertaining computer game. Runs on the H8 or H11 with 8K or RAM or more.

Game Set #1. Lets you play 8 popular computer games including Craps, Orbit, Tic-Tac-Toe, Nim, Hexapawn, Hangman, Hmrabi, and Derby. 8K RAM or more is required on both the H8 and the H11.

Game Set #2. Contains eight more popular games including bagles, slot machine, gomoko, yahtze, apollo, gunner, wumpus and cube.

Heath knew people would want to play games and do interesting things with their computers, so they planned to develop a product line to meet this need. To my knowledge these software products never came to be, and the reason (I believe) is that this demand was met by the Heath Users' Group (HUG).

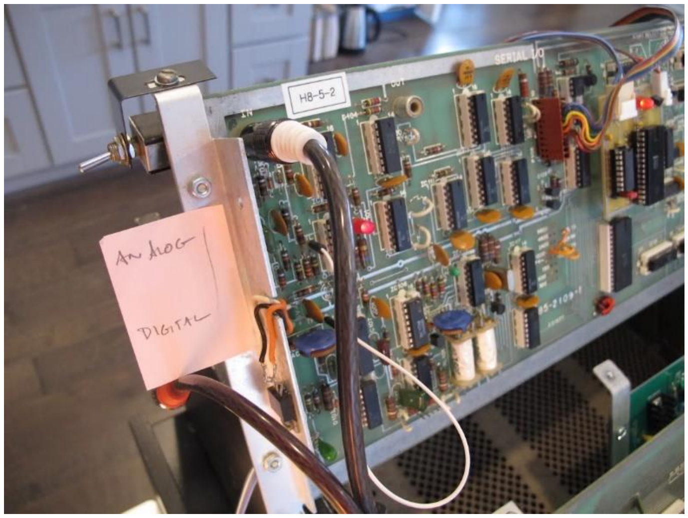
About a year after the release of the H8 the first HUG software release was announced on the cover of REMark magazine [14]:



This was the first entry into what would eventually become a large HUG software library. It was a compendium of mostly BASIC programs, plus a handful of machine language programs, contributed by the Heathkit user community. It was published two ways: "HUG Software Volume 1", a hard copy printout of all the programs (885-1008) and a cassette tape version (885-1009).

Mark Garlanger has preserved the audio tape contents of 885-1009 in the form of .WAV files: https://heathkit.garlanger.com/software/library/HUG/

I was unable to find a version in digital (H8T) form, however I have successfully converted the .WAV files to .H8T by using Dave Runkle's daughterboard adapter. I used an Apple iPad at about 90% volume and had great success (once I got the H8-5 pots properly calibrated!) Dave's daughterboard includes a capability to switch between analog and digital I/O, which was very handy. I would set it to analog, load the .WAV file, then switch to digital and save as .H8T.



The resulting .H8T files are now archived on the SEBHC site:
https://sebhc.github.io/sebhc/software.html#Cassette_Tape_Images

There are five categories of software in this volume: Amateur Radio, Computer-Assisted Instruction, Financial, Games, and Utilities. I would say the game software has best stood the test of time. The financial software looks pretty trivial in hindsight, but back then the idea of keeping your checkbook on a computer was quite exciting. It's interesting to study the software here and fun to play some of the games and reminisce. They mostly used Extended Benton Harbor Basic 10.01 or 10.02 (available in the afore-mentioned Heath tape archive at this same URL), although there are a handful of machine language programs.

Moving to the Disk World

When Heath came out with the H17 disk drive system and associated Heath Disk Operating System (HDOS) in late 1978, they realized users would want a way to convert their existing text and BASIC files to disk. This problem was solved with two programs: BASCON and TXTCON, which were included with the early releases of HDOS and made available (including source) on HUG disk 885-1077 [15]. These programs let you read your files via the H8-5 interface and save them as the appropriate disk files. There was also a HUG program called MEMCON, released as HUG disk 885-1061 [15], that let you save machine programs in Tape Memory Image (TMI) format. The disk included a program GET.ABS which would load and run the TMI program (and in the process wipe out HDOS since the machine images generally were designed to load in low RAM at 040.100).

The process of developing and testing assembly language programs on a tape-based system is quite tedious, but with a disk-based system and HDOS it gets much easier. To support stand-alone demonstrations (just the H8, no disk drives) I wanted to be able to develop .H8T files using my disk-based HDOS setup, however ASM, the HDOS assembler, produces files in .ABS format. To solve this problem, I developed a small program to do the format conversion. The program is written in the 'C' language. I have posted it in the UTILITIES section of the SEBHC wiki:

https://github.com/sebhc/sebhc/wiki

Writing machine programs for conversion to .H8T format requires some care as you will not have access to any disk operating system calls. For console and tape I/O there is a driver documented in Chapter 0, Appendix B of the Software Reference Manual [7]. You can also use any of the routines in the PAM-8 ROM monitor (Chapter 1 of the Software Reference Manual [7]). These provide access to the front panel keypad, seven-segment LEDs, speaker and 2ms interrupt handler.

You could also write code to access the H17 ROM routines [16], but that will only be functional on a disk-based system. In this way it should be possible, for example, to convert the HDOS-based TEST17 program [17] to a stand-alone utility that could run without first having to boot HDOS (a handy capability to have if you're encountering disk problems!)

So that's it for this issue. Hopefully these instructions will get you to the point of being able to load and run some of the very first software ever produced for the H8, and maybe produce some classics of your own!

- Glenn

References

[1] GoTek USB Floppy Emulator, http://www.gotekemulator.com/

[2] The Semi-Virtual Disk, http://thesvd.com/

[3] Z67 IDE+, http://koyado.com/Heathkit/Z67-IDE-plus.html

[4] Heathkit Catalog #817, Heath Company, Benton Harbor, MI,1977, https://heathkit.garlanger.com/catalogs/1977/Heathkit_Catalog_817.pdf

[5] Intel 8251, Wikipedia, https://en.wikipedia.org/wiki/Intel_8251

[6] Kansas City standard (KCS), https://en.wikipedia.org/wiki/Kansas_City_standard

[7] *Software Reference Manual*, part #595-2048, Heath Company, Benton Harbor, MI, 1977, https://sebhc.github.io/sebhc/documentation.html

[8] WAV, https://en.wikipedia.org/wiki/WAV

[9] Heathkit Manual for the Serial I/O and Cassette Interface Card, Model H8-5, part #595-2032, Heath Company, Benton Harbor, MI, 1977, https://sebhc.github.io/sebhc/documentation/hardware/H8/H8-5_AO_Ib_Sc.zip

[10] Teletype Model 33, https://en.wikipedia.org/wiki/Teletype_Model_33

[11] RealTerm: Serial/TCP Terminal, https://sourceforge.net/projects/realterm/

[12] Instructions for the Heathkit Adapter Cables WH9-1, WH8-51, WH11-51, WH8-41, part #597-1836, Heath Company, Benton Harbor, MI, https://sebhc.github.io/sebhc/documentation/software/docs/AdapterCables_597-1836-01.pdf

[13] *Adding a RS232 Interface to the H8-5 Cassette Interface*, http://www.astrorat.com/heathkit/heathkith8computer.html

[14] REMark, Issue 4, Heath Users' Group, St. Joseph, MI, 1978, http://pestingers.net/pdfs/remark-scans/1978/remark-issue4-1978.pdf

[15] Heath Users' Group application library, https://sebhc.github.io/sebhc/software.html#HUG_Application_Library

[16] Software Reference Manual, HDOS System, Chapter 2, General Operations, Appendix E – H17 ROM Code Listing, Heath Company, Benton Harbor, MI, https://sebhc.github.io/sebhc/documentation/software/hdos-2/HDOS_Reference_Chapter2.pdf

[17] Software Reference Manual, HDOS System, Chapter 1, System Configuration, Heath Company, Benton Harbor, MI, https://sebhc.github.io/sebhc/documentation/software/hdos-2/HDOS_Reference_Chapter1.pdf