REMarks

On Heathkit Computing

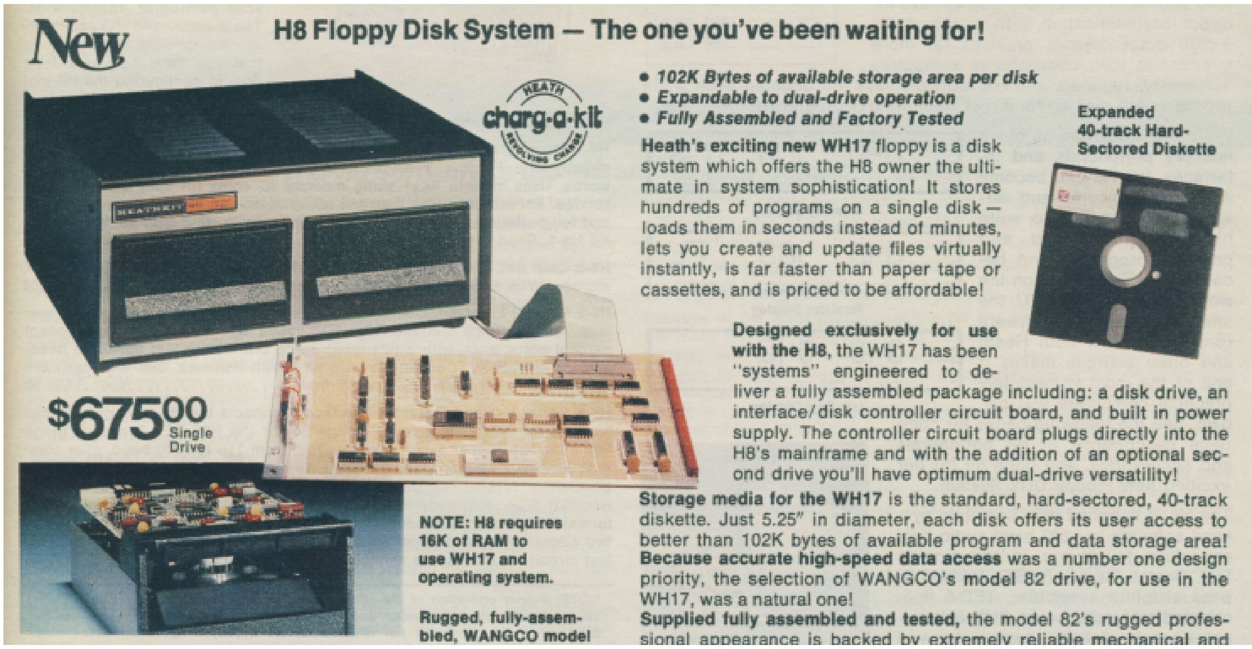
Issue 3 ● 1 January 2021

This is an experimental 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. - Glenn Roberts, Editor.

Starting From Scratch (No Disks!)

In REMarks Issue 1 we talked about the first basic steps in bringing back an old Heathkit computer to operating condition. In this issue, we'll talk about how to get your H8/H17 or H89 up and running the HDOS or CP/M disk operating system, even if you do not have access to any of the original disks. We'll also discuss how to re-create disk copies of archived software packages.

In the summer of 1978 Heathkit introduced their H17 disk drive accessory for the H8:



New

H8 Floppy Disk System — The one you've been waiting for!

- *102K Bytes of available storage area per disk*
- *Expandable to dual-drive operation*
- *Fully Assembled and Factory Tested*

Heath's exciting new WH17 floppy is a disk system which offers the H8 owner the ultimate in system sophistication! It stores hundreds of programs on a single disk — loads them in seconds instead of minutes, lets you create and update files virtually instantly, is far faster than paper tape or cassettes, and is priced to be affordable!

Designed exclusively for use with the H8, the WH17 has been "systems" engineered to deliver a fully assembled package including: a disk drive, an interface/disk controller circuit board, and built in power supply. The controller circuit board plugs directly into the H8's mainframe and with the addition of an optional second drive you'll have optimum dual-drive versatility!

Storage media for the WH17 is the standard, hard-sectored, 40-track diskette. Just 5.25" in diameter, each disk offers its user access to better than 102K bytes of available program and data storage area! **Because accurate high-speed data access** was a number one design priority, the selection of WANGCO's model 82 drive, for use in the WH17, was a natural one! **Supplied fully assembled and tested,** the model 82's rugged professional appearance is backed by extremely reliable mechanical and

Expanded 40-track Hard-Sectored Diskette

$675.00 Single Drive

NOTE: H8 requires 16K of RAM to use WH17 and operating system.

Rugged, fully-assembled, WANGCO model

https://heathkit.garlanger.com/catalogs/1978/Heathkit_Catalog_840.pdf

Prior to that, the only “mass storage” option was cassette tape, which was slow, tedious and somewhat unreliable. The H17 and the Heath Disk Operating System (HDOS) opened up a brave new world and enabled a much more sophisticated set of applications for the H8. A year later Heathkit introduced the new all-in-one H89 computer, which also used the H17 disk drive standard.

NEW **Introducing the Heathkit All-In-One Computer**

New personal computer brings everything together in one compact unit

- Floppy disk storage
- Smart video terminal
- Two Z80 microprocessors
- 16K RAM expandable to 48K
- Professional keyboard
- Numeric keypad

https://heathkit.garlanger.com/catalogs/1979/Heathkit_Catalog_845.pdf

The H17 controller used “hard-sectored” floppy disks. All floppy disks have index holes that can be detected by a photodiode in the disk drive. These index holes synchronize the read and write flow of the sector data with the rotation of the disk. In early disk designs, such as the H17, the index holes were used to delineate the individual sector boundaries. In the case of the H17 format [1] there are 10 holes defining the individual sectors plus an 11th hole marking the beginning of each rotation. As the technology evolved in later designs it was possible to have just a single index hole, with the sectors defined by intelligent disk controller chips. These were called “soft-sectored” disks.

The H17 hard-sectored disk format was the standard for Heathkit for a number of years and many of the Heath User Group (HUG) disks were distributed in this format during that time.

Even later, as users added soft-sectored controllers (H37) and hard drives, they kept their H17 drives in order to work with all of their legacy diskettes.

Heathkit originally supplied the H17-1 disk drive unit, which was a Wangco model 82 (later Siemens FDD-100-5.) These were single-side single-density 40-track devices capable of storing 100K bytes of data. As more advanced disk drive technology became available, it was possible to retain your H17 controller but support higher disk capacities. The Tandon TM-100-2 drive has two heads enabling the use of double-sided media and effectively doubling the disk capacity to 200K. The TM-100-4 supported 80 tracks/side (96 Tracks Per Inch - TPI) which could double capacity again (400K/disk). You still had to use hard-sectored media however as long as you were using the H17 controller. To support these higher capacities, HUG released an improved device driver for HDOS (885-1095) while CP/M users could use Ray Livingston's BIOS-80 BIOS.

Today it is fairly common for hobbyists to come into possession of a system with 24K or more of RAM and one or two working H17 drives but no working diskettes. Disks can easily get lost or damaged over the years and many are unreliable or unreadable due to deterioration of the oxide layers.

The good news is many of the original Heathkit images, HUG disks, third party software disks and user programs have been preserved [2] in H8D format - a linear stream of 102,400 bytes representing the 10 sectors x 40 tracks of 256-byte sectors on a disk [3]. The H8D format also supports the higher density 200K and 400K images. Additionally, Mark Garlanger has developed a more detailed capture format called H17Disk [4] which captures header information in addition to the raw user data. The procedures we will discuss here are focused on the H8D format.

Creating Bootable Disks or Images

There are multiple ways to get your H8/H17 or H89 to bootable condition, some of which involve replacing the floppy drive with something more modern, however to start the discussion let's assume you want to use the original H17 controller and disk drive unit(s) and floppy disk media.

The approach we will take is to use a PC running Microsoft Windows plus some special software (on both the PC and the Heathkit end) that transmits disk images over a serial cable. Here's what you will need:

- A Windows-based PC (I've even run this under Windows XP, but certainly more modern versions are recommended.) The PC should be equipped with a serial port (or for newer computers that lack a serial port you can use a USB-to-serial converter such as the Sabrent CB-9PCF or similar. You may have to load the appropriate COM port driver - see instructions that come with the converter.)
- For the H8 you'll need an H-8-4 or H-8-5 serial interface, H17 disk controller, at least one 5-¼" disk drive, a console (such as the H19 or H9) and sufficient memory to run your operating system (typically at least 32K, though 48K is recommended.)

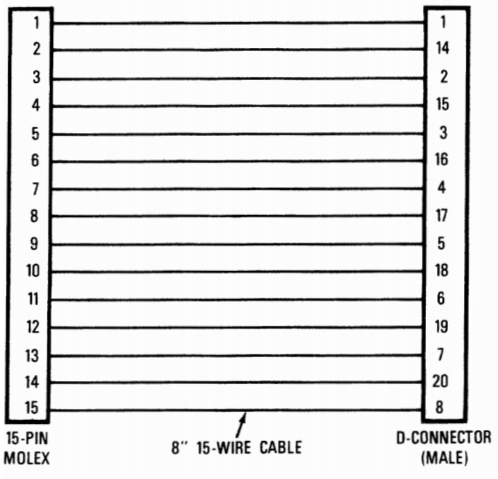
- For the H89 you'll need the H17 interface (H88-1) and drive, 48K RAM and an installed serial card (this is a pretty "standard" configuration.)
- You'll need a serial cable to connect the two machines. I use a DE-9 (female) to DB-25 (male) "no-twist" (i.e. not a "null modem") cable. These were common years ago for connecting peripherals to PCs. If you're using a USB-to-serial converter some of those are built into the cable itself, in which case you shouldn't require a separate cable.
- Most importantly, you will need 5-¼" hard-sectored floppy disks that use "10 Hard Sectors." These are no longer manufactured and even New Old Stock (NOS) ones are hard to find. One option is to make your own using a special hole punch device that adds the appropriate 10 holes to an existing soft-sectored disks (these are the "360K" DS/DD disks that were quite prevalent in the 1980s.) The SEBHC Google group (sebhc@googlegroups.com) has developed at least two versions over the years - an email to the group might provide more information and possibly a volunteer who will punch disks for you, or sell/give you some disks or a punch. (If you can't find hard-sectored disks there are some other hardware options that we'll discuss later.)

A floppy disk "punch" to convert soft-sector disks to hard.

Here are the steps to creating your disks:

Step 1 - Connect the Machines:

- On an H8 with H-8-4 card: Configure one of the H-8-4 ports for port number 340Q. Do not install the interrupt jumper for that port. Connect a Heath WH8-41 [5] cable (same cable as used by your console) to the DCE (upper) connector for the desired port (note: Pin 1 is to the right). Below is the wiring for the WH8-41 cable. You can also make one with a 15-pin Molex connector and a ribbon cable with DB-25 connector such as Jameco part number 115984.



WH8-41 Cable Wiring

In RS-232 terminology your H8 or H89 will serve as the Data Communications Equipment (DCE) and the PC you connect it to will be the Data Terminal Equipment (DTE).

- On an H8 with H-8-5 serial card: you will use the console connector so disconnect your console (the console is not needed during the transfer process). You'll need the cable from the H-8-5 to have the appropriate DB-25 connector (e.g. using Heath cable WH8-51 [5]).

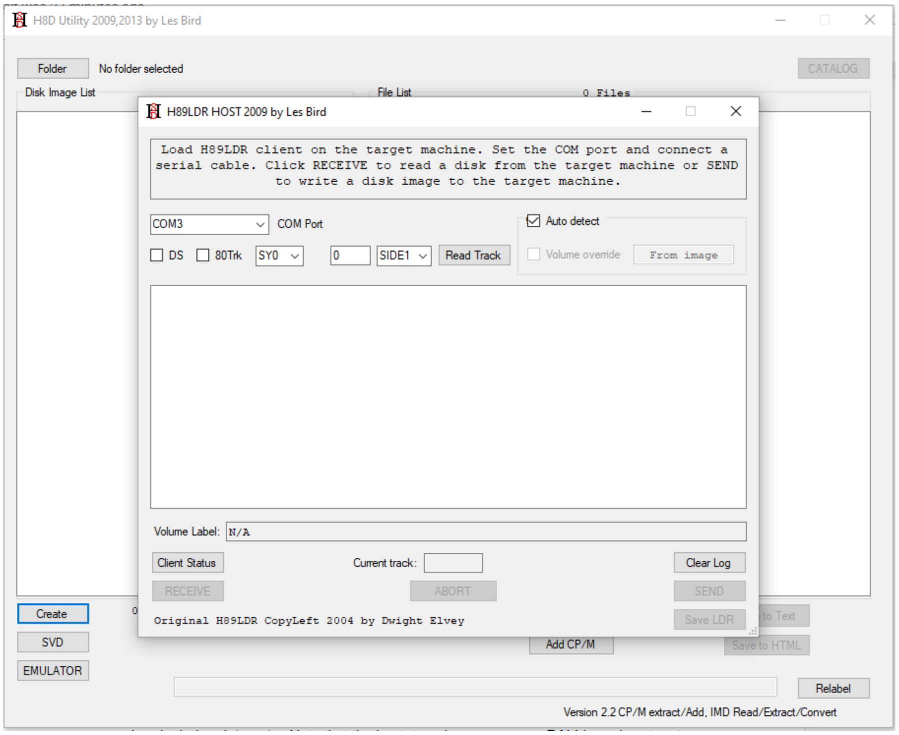
- On an H89: You will be using the port marked 340-347 DCE (left most DB-25 connector, female).

- Connect the H8/H89 port to your serial port on the PC using the serial cable. As this is connecting DCE (H8/89) to DTE (PC) there is no need for a null modem. As DCE ports normally have a female connector and DTE have male you generally should not need a gender changer, however these conventions were not always followed so your situation may vary.

Step 2 - Install the PC Client Software:

The technique you'll be using was originally developed by Dwight Elvey in 2004 [6] and uses a client/server architecture. A rudimentary server will be installed on the H8/89 and a client program on the Windows PC will provide a user interface and issue commands to the server. Though Dwight originally developed his client in the FORTH language for use under MS-DOS, a more modern version has been developed by Les Bird and included as one of the functions in his Windows-based "H8D Utility". Download and install the H8D Utility software on your Windows machine following the instructions in Les' documentation [7]. If you have an older version of the H8D Utility remove it first - the latest version (as of this writing) is 2.2. *You should* *read through all of the documentation on the H8D Utility as only a cursory overview is provided here.*

Run the H8D Utility program and from the main screen click the “Create” button in the lower left. This will bring up a screen similar to the following, where you should make sure the correct serial port is selected. Click on the “Client Status” button and after a short while an error message will come up (because the software isn’t running on the Heathkit side) but also there will be a new button labeled “Save” in the window - you’ll use that in a little bit.


H8D Utility 2009,2013 by Les Bird

Folder: No folder selected

H89LDR HOST 2009 by Les Bird

```
Load H89LDR client on the target machine. Set the COM port and connect a
serial cable. Click RECEIVE to read a disk from the target machine or SEND
to write a disk image to the target machine.
```

COM Port: COM3

☑ Auto detect

□ DS □ 80Trk SY0 0 SIDE1

□ Volume override

Volume Label: N/A

Original H89LDR CopyLeft 2004 by Dwight Elvey

Version 2.2 CP/M extract/Add, IMD Read/Extract/Convert

The H8D Utility screen after selecting the “Create” button.

Step 3 - Bootstrap the Server:

Installing the server on the Heathkit side would seem to be a dilemma - without a functioning operating system how do we load the program? Dwight’s solution was to develop a very simple bootstrapping program consisting of fewer than 50 bytes of machine code - small enough to be entered and executed using the computer’s ROM-based monitor. The bootstrapping program talks to the client to download and execute the full-function server, at which point everything can be driven from the Windows client.

Below are the instructions for bootstrapping - *you may want to print these so you have hard copy handy during data entry*. Note that the bootstrapping server uses RAM-based vectors to access disk drive functions. These RAM locations are only loaded with the appropriate jump vectors when you attempt to actually boot the machine via the H17. For that reason it is important to first attempt to boot from disk (even though you don't yet have a bootable disk) as described in these steps:

For the H8:

- Power up the H8 and H17 with no disks in the drives.
- Perform a "fake" boot from the H17 to load the H17 jump tables):
    - REG
    - PC
    - ALTER
    - 0 3 0 0 0 0
    - ALTER
    - GO (drive 0 should go active…)
    - RST/0 (/ and 0 keys at the same time to reset)
- Hand enter the machine code found in the appropriate listing below (based on which serial interface board you have):
    - MEM
    - 0 4 3 0 0 0
    - ALTER (begins data entry)
    - Enter the values from the table, one byte at a time…
    - ALTER (turns off data entry)
- Execute the program:
    - REG
    - PC
    - ALTER
    - 0 4 3 0 0 0
    - ALTER
    - GO

  (nothing will appear to happen yet - go to Step 4 below to continue on the client side).

For the H89:

- Power up the H89.
- With no disk in the drive, perform a "fake" boot by entering 'B' (this loads the H17 jump tables in RAM).
- Press Shift-Reset to restore the H: prompt.
- Enter S43000<return> to begin code entry
- Hand enter the machine code found in the appropriate listing below. After entering each byte press the spacebar to sequence to the next address. When data entry is complete hit <return> to get back to the H: prompt
- From the H: prompt enter G43000<return> to execute the program.
  (nothing will appear to happen yet - go to Step 4 below to continue on the client side).

Bootstrap Server Code:

Use the following machine code *only for the **H89** or the **H8 with H-8-4** serial card*. The first number on each line is the address (in “split octal” as shown on the H8 LEDs) and the second number is the byte value to be stored there:

```
043000  257
043001  323
043002  343
043003  323
043004  341
043005  323
043006  344
043007  075
043010  323
043011  343
043012  076
043013  014
043014  323
043015  340
043016  257
043017  323
043020  341
043021  076
043022  007
043023  323
043024  343
043025  333
043026  345
043027  333
043030  340
043031  041
043032  037
043033  043
043034  021
043035  132
043036  046
043037  333
043040  345
043041  037
043042  322
043043  037
043044  043
043045  333
043046  340
043047  022
```

```
043050  033
043051  351
```

If you have an ***H8 with H-8-5*** *card* you should enter this version. The first number on each line is the address (in “split octal” as shown on the H8 LEDs) and the second number is the byte value to be stored there:

```
043000  076
043001  252
043002  323
043003  373
043004  076
043005  100
043006  323
043007  373
043010  076
043011  316
043012  323
043013  373
043014  076
043015  025
043016  323
043017  373
043020  333
043021  373
043022  333
043023  372
043024  041
043025  036
043026  043
043027  021
043030  132
043031  046
043032  000
043033  000
043034  000
043035  000
043036  333
043037  373
043040  037
043041  037
043042  322
043043  036
043044  043
```

```
043045  333
043046  372
043047  022
043050  033
043051  351
```

Step 4: Download the Server Code and Save

In the H8D client on your Windows PC select the “Save” button - this will cause the full server code to be downloaded and executed on the Heathkit side. You should see several messages:

```
Sending H89LDR3.BIN to target machine…
H89LDR3.BIN sent successfully
Client is not ready
Client is ready
```

At this point the two programs are successfully communicating! A pop-up window will ask whether you want to save to target. This is a feature to write the code to the boot track of a hard-sectored disk (to save you the trouble of going through the bootstrap entry process in the future). It is a good idea to do this, so insert a floppy disk in drive 0 (doesn’t need to be formatted) and click “Yes.”

At this point both client and server are configured and operational. Now whenever you want to use this procedure you can simply install your cable, boot your H8/89 directly from the floppy disk with the saved bootstrap and fire up the H8D Utility program on your PC.

Next we’ll cover how to create disk copies from the archival H8D images.

Step 5: Creating a Disk

There is a large and growing archive of useful disk images in H8D format maintained by Les Bird on his SEBHC site https://sebhc.github.io/sebhc/, operating system master disks [8], the HUG library [9], and a large collection of archived disks from users [10]. He also has images containing software tools, games and other applications of interest.

To create a disk start by locating and downloading the appropriate H8D images to your PC. With your Heathkit system connected to the PC and both systems up and running the client software, verify that you’re ready to transfer by clicking the “Client Status” button. You should see “Client is ready,” indicating that the two programs are successfully linked.

Insert a blank hard-sectored diskette (need not be formatted) in H17 drive 0 and click on the “Send” button. This will bring up a file browser window where you should select the image you want from the H8D files you previously downloaded. You will then be prompted to begin the transfer. Click “Yes” and you should soon hear disk activity. The program transfers one track at a time so for a standard 100K disk there will be 40 disk writes. Transfer is done at 9600 BAUD and it will take several minutes to complete a disk. Upon successful completion of transfer you'll get a "Done" message followed by "Client is ready." At this point your disk is ready to use. You can either proceed with creating more disks or reset your computer and boot up to use the new disk!

If you Can't Find Hard Sectored Disks Use an Emulator

Hard-sectored 5-¼" diskettes are hard to find but soft-sectored ones are still relatively plentiful. One solution to this problem is to use a device that simulates the missing index hole events. In 2009 Chris Elmquist developed the Hard Sector Floppy Emulator (HSFE) for this purpose [11]. The HSFE can only be reliably used with 3-½" floppy drives. Do-it-yourselfers can create an HSFE board using Chris' documentation or construct one of the HSFE-capable boards developed by Norberto Collado (koyado.com) - H89-Z17-HSFE or H8-H17-H36-H67-USB.

Another option is the Virtual Sector Generator (VSG) available from Mike Douglas [12]. Mike has kits or assembled units available for purchase from his web site. Mike's VSG units are advertised to work with 5-¼" drives.

The Semi-Virtual Diskette

Around 2007, Eric Rothfus [13] developed something he called the Semi-Virtual Diskette (SVD), which is a microcontroller-based emulator of 5-¼" disk drives. Through an RS-232 or USB link to your Windows- or Linux-based PC you can download disk images and mount them directly on the H8/89 side. The device plugs into the 34-pin ribbon connection coming from the H17 controller board.

The SVD is no longer in production and the original software is rather old as no development work has been done in some time (I have to run it on my old Windows XP machine as it does not seem to run in Windows 10.) In 2017 Dave Runkle added some life to the SVD community by creating an updated version [14], however that seems to have been a one-time effort and I believe he no longer is engaged in that.

The SVD is a good idea as it eliminates reliance on old physical drives and media, however it is suffering a bit from a lack of support. Les' current H8D Utility can load SVD images and I have tested that successfully with several OS images.

GoTek

Another "virtual" drive option is to use a GoTek floppy disk emulator[15]. These are marketed as drop-in replacements for 3-½" floppy drives used in old computers, musical instruments and industrial machinery. They can, however, be made to emulate a 5-¼" drive using special firmware developed by Jean-François Del Nero (a.k.a. Jeff) [16]. Currently this firmware supports read-only functionality on the H17 format, however this does let you take legacy disk images and read them to transfer the contents to more modern storage formats.

In 2018 Darrell Pelan took some time to document his recommendations and extensive insight into how to configure the GoTek device. I have included Darrell’s writeup as an appendix to this issue - a “must read” for anyone who wants to delve into this device.

If you are interested in experimenting with the SVD or GoTek you could reach out to the SEBHC list (sebhc@googlegroups.com) to see if others can help you get started.

So that’s it for this issue. Hopefully these instructions will get you to the point of being able to boot and operate your computer via your H17 drive!

- Glenn (glenn.f.roberts@gmail.com)

References

[1] Jorgensen, E. Tom, 1981, “Dissecting the HDOS Diskette,” *Microcomputing*, July, 1981, https://sebhc.github.io/sebhc/documentation/supplemental/HDOS_Disk.pdf

[2] “Society of Eight-Bit Heathkit Computerists (SEBHC) - Software Home”, https://sebhc.github.io/sebhc/software.htm.

[3] “H8D Format,” https://sebhc.github.io/sebhc/h8d.html

[4] Garlanger, M., “H17Disk - New H17 Hard-sectored Disk Image,” https://heathkit.garlanger.com/diskformats/H17Disk-draft.pdf

[5] Heathkit, “Instructions for Adapter Cables,” https://sebhc.github.io/sebhc/documentation/software/docs/AdapterCables_597-1836-01.pdf

[6] Elvey, D., 2004, “Disk Image Transfer Program Vers. 8”, https://sebhc.github.io/sebhc/software/Utilities/H8DUtility/README.TXT

[7] Bird, L., “H8D Utility,” documentation:
https://sebhc.github.io/sebhc/software/Utilities/H8DUtility.html
Software download:
https://sebhc.github.io/sebhc/software/Utilities/H8DUtility.zip

[8] Bird, L., “Boot Images,” (various bootable images in H8D format), https://sebhc.github.io/sebhc/software/Utilities/H8DBootables.zip

[9] Bird, L., “HUG Library,” (archive of 107 HUG disk images), https://sebhc.github.io/sebhc/software/HDOS/HUGLibrary.zip

[10] Bird, L., “Application Archive,” (archive of H8D images), https://sebhc.github.io/sebhc/software.html#Application_Archive

[11] Elmquist, C., 2009, “Hard Sector Floppy Emulator,” https://sebhc.github.io/sebhc/storage/hsfe.pdf

[12] Douglas, M., “Virtual Sector Generator,” https://deramp.com/vsg.html

[13] Rothfus, E., “Semi-Virtual Diskette,” http://www.thesvd.com/

[14] Runkle, D., “Semi-Virtual Disk Project,” http://www.astrorat.com/semivirtualdiskproject.html

[15] GoTek, “GoTek Floppy Emulator,” http://www.gotekemulator.com/

[16] Del Nero, Jean-François, “HxC Floppy Drive Emulator,” https://torlus.com/floppy/forum/

Appendix

Heathkit Gotek Floppy Emulator Notes

Darrell Pelan

2 January 2021

Table of Contents

Acknowledgements

Jeff at HxC for developing the firmware and software to modify the Gotek floppy emulator for Heathkit computer systems

Steven Hirsh, Mark Garlanger, Norberto Collado, Les Bird, and many others for providing the detailed technical knowledge of the Heathkit disk systems.

Hardware

- Gotek Floppy emulator ($27.99 from Amazon)
- USB to TTL Serial Cable ($6.99 from Amazon)
- USB flash drive (32GB works, probably could use smaller)
- Floppy disk cable with twist

Software

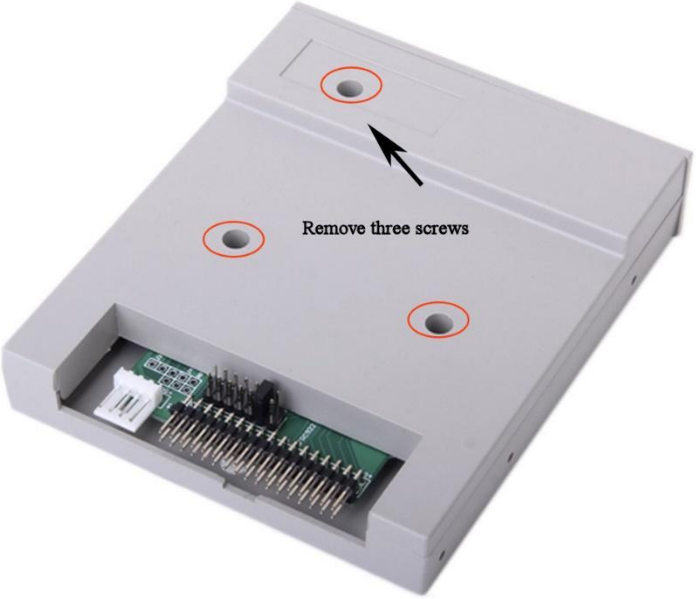
Download the software from the following webpage

- USB STM32 HxC Floppy Emulator firmware for Gotek update: HXCFEUSB_HFE_beta_firmware.zip
- Last firmware release notes : USB_HFE_hxc_floppy_emulator_firmware_release_notes.txt
- Online Bootloader programmer: STM32_SerialBridge.zip
- You can purchase the firmware on this page : http://hxc2001.com/store/

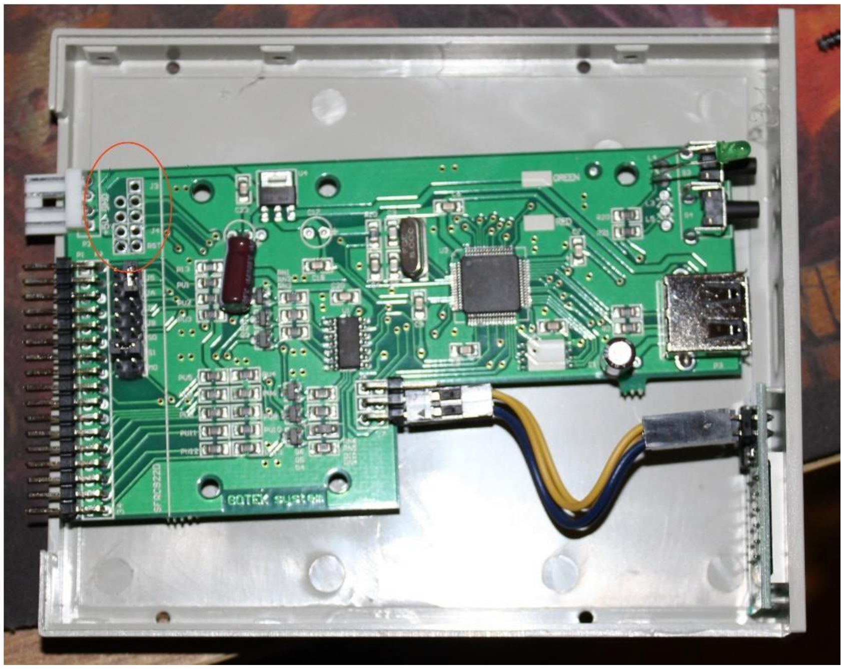
Gotek

Hardware modification

Your drive may need to be modified in order to program it. It needs a set of jumper pins added to connect the serial cable.



Solder a set of dual jumper pins in the area indicated.



Reassemble the unit

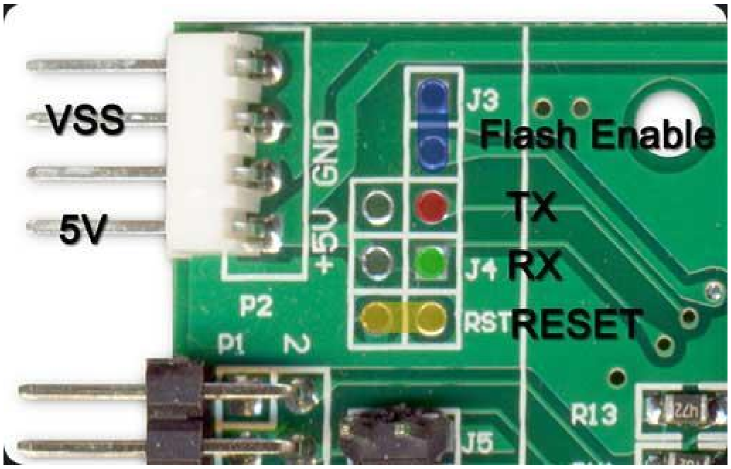
Update the HXC

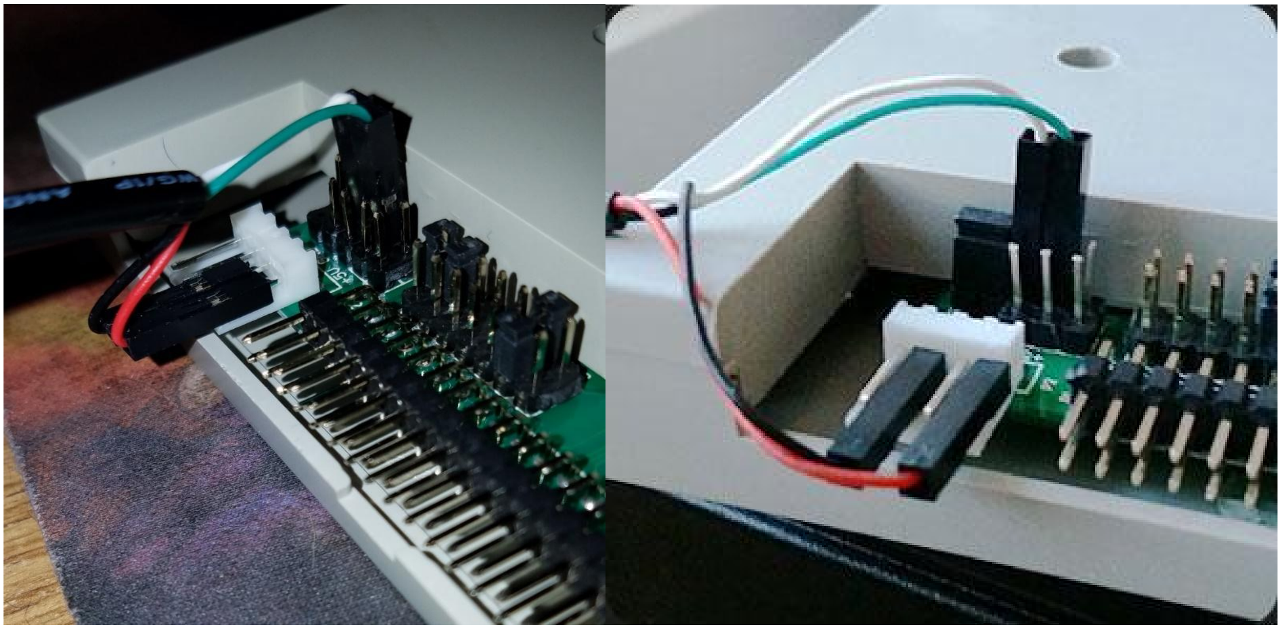
You should have purchased the software and downloaded the serial loader software, SecSerialBridge. The purchase email listed the server IP address and login information.

Connect the serial cable as indicated in the **First time bootloader Flashing/Programming process** at the following link. Note the jumper at J3. You may need to use a wire to briefly ground the Reset pin if the software doesn't like up initially.

Serial cable pinout

Red cable: +5V
Black cable: GND
Green cable: TXD
White cable: RXD





Flash the Device

Follow the directions to flash the unit with the new bootloader

- Start the SecSerialBridge software and select the COM port of your serial adapter. Click on the button "Open"
- Enter the server and your contacts data. Click on the button "Connect".
- If the connection to the server succeeds you will get the "Link is UP !" status.
- Some seconds later you should get the "Device detected." status.
- Note : If you don't get it, try to reset the STM32 by making a short connection of the reset signal (RST) to GND.
- Once the device is detected, you can press the "Program" button.
- The download and programming process finish with a "Done !" Status. The whole process takes some minutes.
- Once done, remove all the connections previously done on the programming pins.

- The device should now display "Ldr" at power up and wait for a USB flash drive with a UPD file to use.

Update the Device

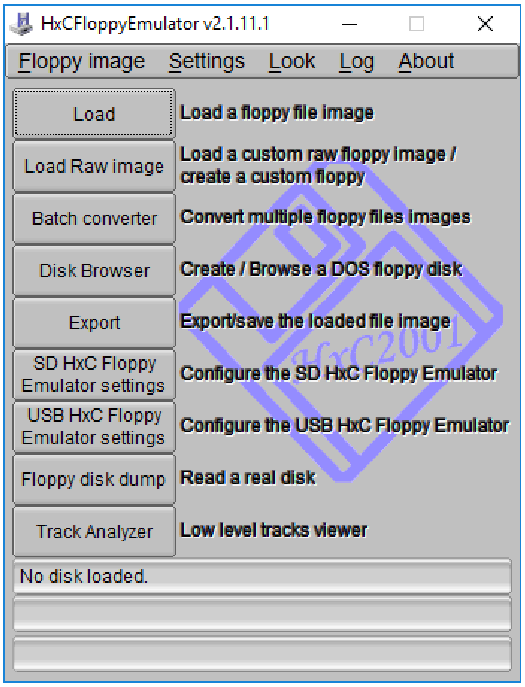
- Copy the new firmware file (*.upd) to a FAT32/FAT16/FAT12 USB flash device.
- Insert the USB Stick into the emulator and press both buttons.
- Power up the device and keep both buttons pressed during at least 2 seconds.
- Wait some seconds... its done !

Set the Device Config

- Download file HXCFEUSB_HFE_beta_firmware from web page http://torlus.com/floppy/forum/viewtopic.php?f=33&t=1683. Direct link: http://hxc2001.com/download/floppy_drive_emulator/HXCFEUSB_HFE_beta_firmware.zip
- Select the config file for index mode

Create Disk Image

Open the HXC Floppy Emulator software


HxCFloppyEmulator v2.1.11.1
Floppy image
Settings
Look
Log
About
Load
Load a floppy file image
Load Raw image
Load a custom raw floppy image / create a custom floppy
Batch converter
Convert multiple floppy files images
Disk Browser
Create / Browse a DOS floppy disk
Export
Export/save the loaded file image
SD HxC Floppy Emulator settings
Configure the SD HxC Floppy Emulator
USB HxC Floppy Emulator settings
Configure the USB HxC Floppy Emulator
Floppy disk dump
Read a real disk
Track Analyzer
Low level tracks viewer
HxC2001

No disk loaded.

Soft Sector

Click **Load** and navigate to the directory with .IMD file you wish to load.

Click **Export** and select "HFE" as the export file type for soft sector controllers

Copy the exported file to the USB flash drive. You will also need the latest Beta firmware from the HXC site the first time you load the Gotek.

Hard Sector

Click **Load** and navigate to the directory with .H8D file you wish to load.

Click **Export** and select "HFE version 3" as the export file type for hard sector controllers

Copy the exported file to the USB flash drive. You will also need the latest Beta firmware from the HXC site the first time you load the Gotek.

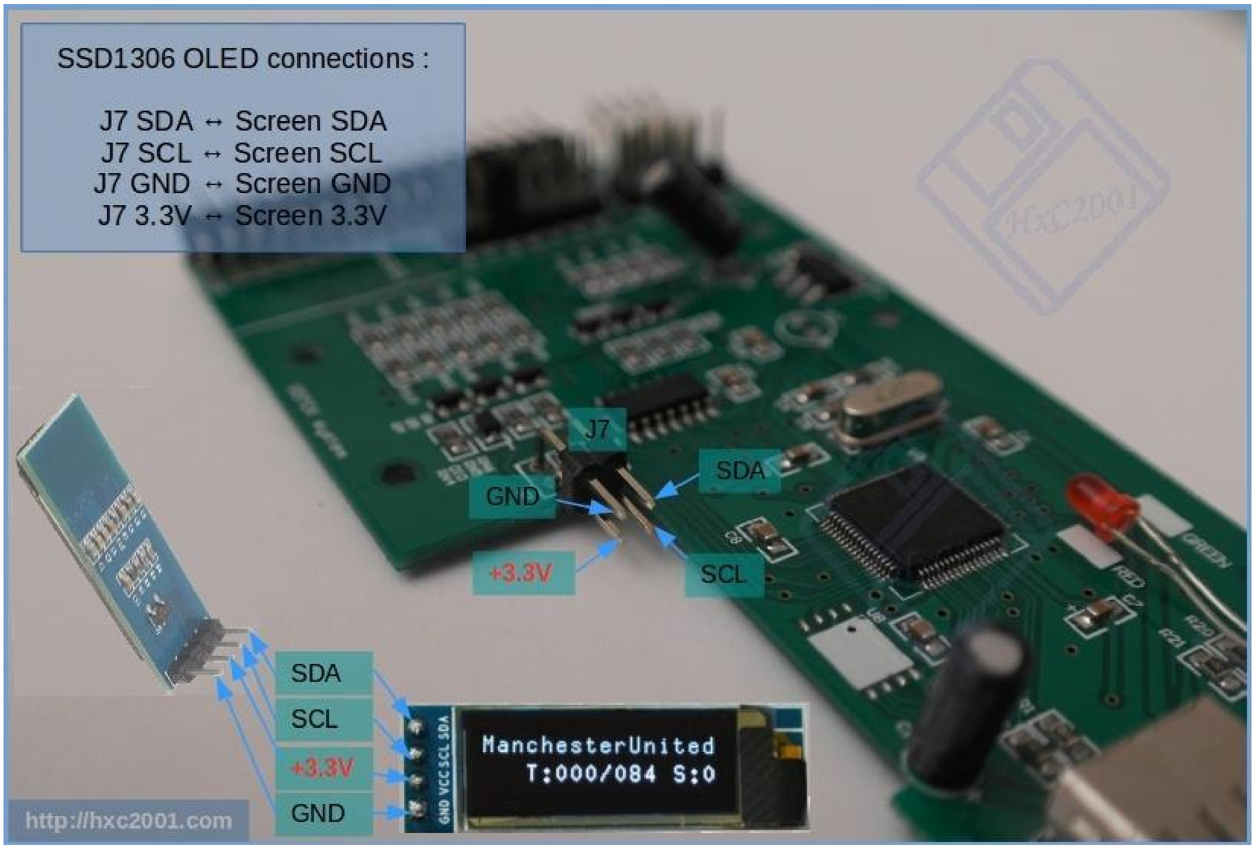
Connect to the H-89

- H-17 Hard Sector controller
    - The Gotek should be connected after the twist for drive 0
- H-37 soft sector controller

- The Gotek can be connected using a standard PC twisted cable using either connection

LCD Screen

- Purchase LCD screen https://www.ebay.com/itm/302110929185
- You can connect the screen directly to the J7 connector. (additional pull-up resistors NOT required.)
- Use Normal mode
- 3D print new case https://www.thingiverse.com/thing:2725715



Speaker

- Add a speaker to hear track changes

Usage Notes

General

Once you flash the Gotek with the HxC software, it's operation changes. I used Dave Dunfield's IMD software to read my floppy disks from the 80's on my Windows 95 system, hardware circa 1995, to read the disks into a IMD format. The HxC Floppy Emulator software converts an IMD image to HFE. The flashed Gotek uses the HFE format on the flash drive. Data can be read and written to this format for soft sector images. HFE is a bit map of the MFM data on the floppy disk.

The USB drive is formatted in DOS FAT32. Each floppy disk image is a separate file. Each file needs to be in the expected format for the target system. The LCD screen is nice because it shows the DOS file name so you know which “disk” you are using.

H8DUtility

You can use HxC Floppy Emulator to convert the HFE file back to an IMD image. The IMD file format uses a storage method where sectors with single byte value are stored with an identifier, byte count, and byte value. This method makes it hard for image reading software since the sectors are not the same size.

The H8DUtility can convert the IMD files to an H37 format. The H37 format is a byte for byte image of the disk.

The H8DUtility program can extract and add CP/M files, but not delete them. It can also create a blank image that you can add file to in order to transfer data. On a H-89, the first sector contains a byte that tells the operating system how big the disk is, including the number of tracks and sides. This allows you to have images of both 40 and 80 tracks.