REMarks
On Heathkit Computing

Issue 7 ● 9 April 2026

A community-supported publication to capture and share knowledge and experiences with Heathkit computers.

One of my 2026 New Year's resolutions was to get back to publishing these (after a long hiatus!), and so it pleases me to thank Jeff Tranter for this issue's content. Jeff has been an avid contributor to the Heathkit computing community, particularly through his YouTube videos[1] and his book on Heathkit computing[2]

If you have material or topic ideas to contribute, please contact me via the SEBHC Google Group (https://groups.google.com/g/sebhc) or on the SEBHC Discord channel.

- Glenn Roberts, Editor.

HDOS Support in Z88DK

By Jeff Tranter

There are several *native* C compilers available that run on Heath/Zenith 8-bit computers running HDOS, most notably the C/80 compiler originally developed by *The Software Toolworks*[3]. These compilers can be fun and nostalgic to use as this is the way most software was originally developed.

By contrast, a *cross compiler* is a software development tool that runs on one computer architecture (the host) but generates executable machine code for a different architecture or operating system (the target). The main advantage of a cross compiler is that it can run on a more powerful modern computer, making compilation faster and allowing the use of tools like source code editors and build tools (e.g. `make`) that run on the host computer.

Z88dk[4] is a comprehensive ANSI-compatible development kit designed for Z80 family (8080, 8085, Z80, Z180, etc.) processors. It includes an ANSI-compatible C compiler, assembler, run-time libraries and linker. *Z88dk* allows developers to create software for over one hundred different target platforms using C and/or assembly language. Native compilers like C/80 primarily support the original Kernighan and Ritchie ("K&R") coding style[5], however z88dk gives you the full power of ANSI C.

For some time *z88dk* has supported Heathkit 8-bit computers running the CP/M operating system, but until recently there was no support for Heathkit's original HDOS platform. As part of the initiative to support FujiNet[6] on the Heathkit H89, support for HDOS has recently been added to *z88dk*. Most of this work is being done by Stefano Bodrato.

Installing z88dk

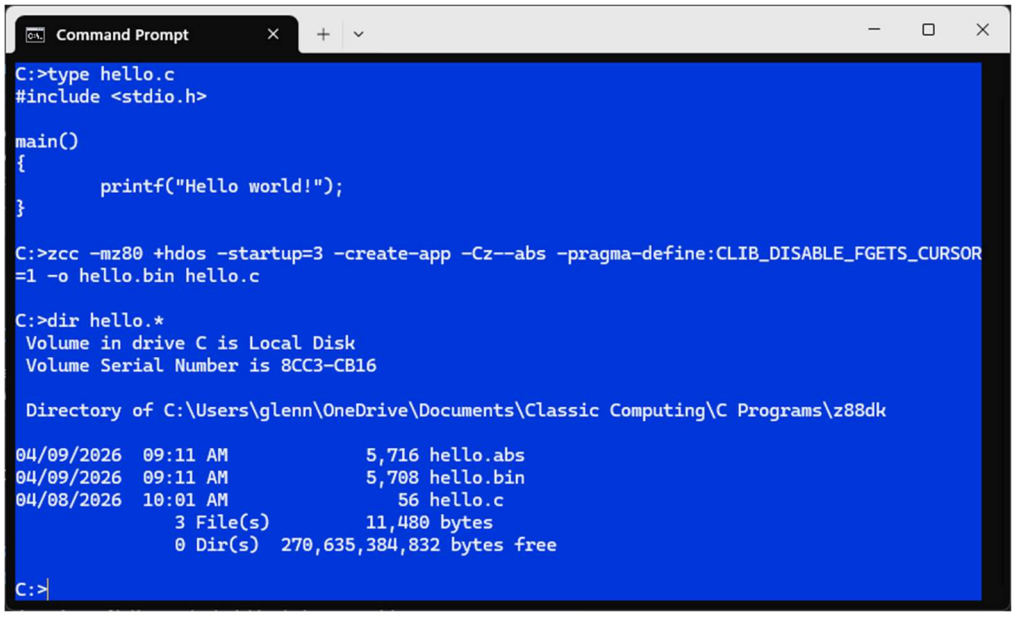
The *z88dk* tools run on Windows, macOS, and Linux platforms. The best place to start your installation is the *z88dk* download site[7] where you'll see links for the current release as well as the "nightly" automatic build. At the time of writing, HDOS support is not yet part of the official z88dk release (2.4) so you should follow the instructions for downloading the nightly build. HDOS support should be included in the next full release.

The *z88dk* github repository[8] has the most recent complete releases, including full source code (for Linux, Win32 and Mac OSX) as well as installation instructions. If you prefer, you can build it yourself from the source code using the instructions on the site. The installation instructions[9] contain important and useful information. It is worth pointing out that the authors discovered that on the Windows platform you must not to have any spaces in the directory names for the installation path (some of the scripts will fail), so installation in `C:\z88dk` is recommended.

Compile Options

The main z88dk tool used is the C compiler, invoked as `zcc`. To specify the HDOS platform, use the option `+hdos`.

In practice, you typically want to specify additional options. Here is an example of compiling a source file named `hello.c`:



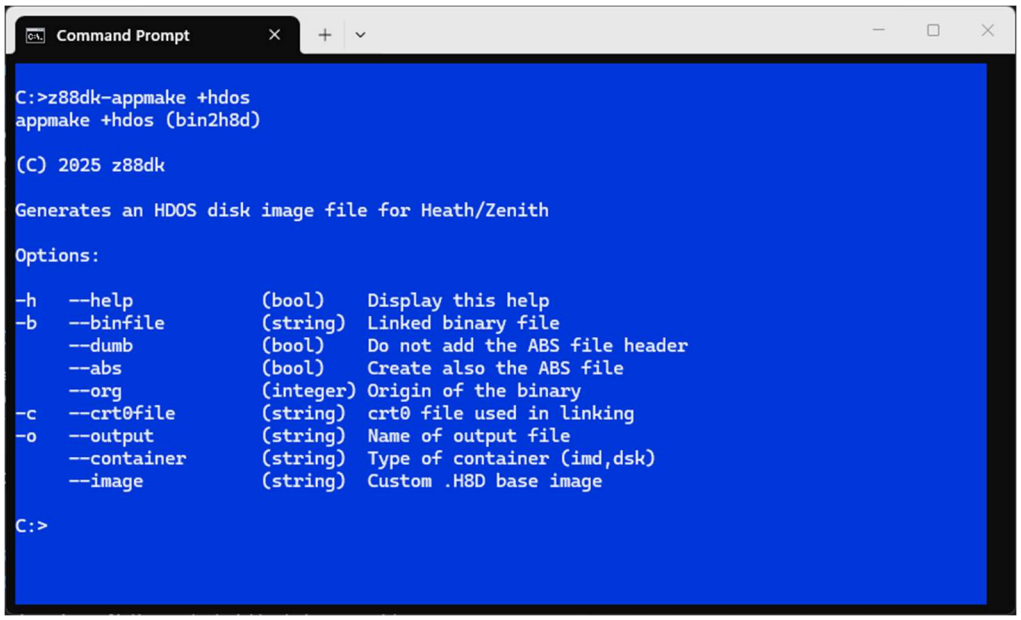
In the above command line:

`-mz80` specifies generation of code using the Z80 instruction set. This is not strictly necessary as it is the default.

`+hdos` specifies the HDOS platform.

`-startup=3` is a target-specific option, in this case for command line option parsing. The value 3 specifies to parse command line options and pass them to the `main()` function as standard `argc` and `argv` parameters. Without this option, the default is to prompt the user for command line options. The command line parsing support is still somewhat experimental but may become the default in the future.

`-Cz` passes options to the `z88dk-appmake` program, a utility which can produce files which are suitable for use in emulators or on real hardware. Typing that command on its own with the `+hdos` option displays the available functions:



Passing the `--abs` option specifies generation of a file in the HDOS “ABS” format, which can be directly executed, rather than the raw binary file normally generated by the compiler.

If you are using an emulator or other tool which can accept a floppy disk image in H8D format, you can use the `--image` option to specify to generate a disk image rather than an ABS file.

The option `-pragma-define:CLIB_DISABLE_FGETS_CURSOR=1` tells the C run-time library to suppress displaying an input cursor character during keyboard input from the `fgets()`

function. This is not unique to the HDOS version and is usually desired on the H89, as the hardware generates the cursor.

Finally, the `-o` option specifies the name of the binary output file. In this case it is subsequently converted to ABS format by the `z88dk-appmake` program.

The last argument is the C source file, `hello.c`.

There are many other options to `zcc` which you can look up in the documentation.

Running the Code

Once you have generated an ABS file you need to transfer it to your H8/H89 system. With a real H89 system you can do this over a serial port using a communications program like MAPLE on the H89 and a terminal emulator or file transfer program on the host computer that supports xmodem protocol. A more efficient method is to use an H89 VDIP1 board which adds support for reading and writing files on a USB mass storage device[10].

Current Features

Basic console input and output is working. File I/O is mostly implemented, although not heavily tested. Executables should run on all versions of HDOS (1.x, 2.0 and 3.x).

You can adjust how many files can be open simultaneously using a compile option like:

```
-pragma-define:CLIB_OPEN_MAX=5
```

In this case specifying 5, which is the maximum allowed by HDOS. Smaller numbers will make more memory available at run-time. The default is currently 3.

Command line redirection is supported, allowing you to read standard input or direct standard from/to a file on the HDOS command line, e.g.

```
> foo >output
> foo <input
```

You can turn this off, if desired, in source code with `#pragma output noredir` or with the equivalent compiler option.

Date and time support is implemented via the functions `time()` and `localtime()`. Some HDOS-specific functions are defined in the header file `<arch/hdos.h>`. There is not yet a lot of documentation of these so examine the include file itself to learn more.

The library assumes a clock speed of 2 MHz. If you are running your CPU at a different speed you can define the symbol `__CPU_CLOCK` to the clock speed in Hertz. Some library functions like `sleep()` use this value to calculate timing delays.

By default, z88dk uses CP/M style line termination (CR/LF) for reading and writing text files, whereas HDOS text files use a single newline (LF) character. If you want the standard HDOS line terminator, open the files in binary mode when calling `fopen()`.

While z88dk does support generating code for the Intel 8080 processor, many library functions use Z80 code, so support for an 8080-based machine, like a stock Heathkit H8, would require additional work. An H8 with a Z80 board will work fine.

Future Work

As FujiNet support is implemented for the H89, additional features, bug fixes, and performance improvements to z88k for HDOS are expected to be made.

Summary

I have found the z88dk HDOS port useful for porting even relatively large C programs to HDOS, typically with no changes to the source code. As well as enabling a port of FujiNet to the H89 running HDOS, it provides a path to port other CP/M and Unix/Linux C programs to HDOS and for developing new code in ANSI C on a fast development machine.

References

[1] Jeff Tranter's YouTube channel: https://www.youtube.com/@jefftranter

[2] "*Classic Heathkit Computers, Calculators, and Robots,"* Jeff Tranter, Feb. 2024, ISBN 978-0-9921382-2-6, Amazon.com link: https://a.co/d/0i1KP0Rn

[3] C/80 for HDOS can be downloaded from the SEBHC Github Wiki: https://github.com/sebhc/sebhc/wiki#c80-working-disks

[4] Z88dk main site: https://z88dk.org/site/

[5] "*The C Programming Language,*" Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall Inc., 1978, ISBN 0-13-110163-3.

[6] https://fujinet.online/

[7] Z88dk download site: https://z88dk.org/site/download

[8] Github repository for the z88dk development kit, https://github.com/z88dk/z88dk

[9] Z88dk installation notes: https://github.com/z88dk/z88dk/wiki/installation

[10] "*Universal Serial Bus (USB) for the H8/H89*," Glenn Roberts, *REMarks*, Issue #6, https://sebhc.github.io/sebhc/REMarks/REMarks%20Issue%206%20%20-%2014%20February%202022.pdf