



REMark

On Heathkit Computing

Issue 6 • 14 February 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.

Universal Serial Bus (USB) for the H8/H89

By Glenn Roberts

Modern personal computers include Universal Serial Bus (USB) capability [1] to allow the user to easily back up and transfer files using flash drives (also known as memory sticks) and other USB storage devices [2]. Hobbyists who collect, use, and restore legacy computers, such as the Heathkit H8 and H89, have historically not had such a capability since the USB protocols are complex and not easily implemented on smaller 8-bit platforms (and didn't even exist back in the H8/H89 days). The removable storage, file interchange, and backup solution originally developed for these systems was the floppy disk, but 40+ year-old floppy disk media and disk drives can be quite unreliable in the modern day.

Today USB is an industry standard that establishes specifications for interfacing computers to peripherals, including storage devices. First released in 1996 the USB interface has become omnipresent in the modern computing ecosphere and has evolved and matured through many evolutions of the standard. USB flash drives are ubiquitous and provide an inexpensive, universally accessible, portable tool for storing and interchanging files and data.

This issue of *REMark* documents the work performed by members of the Society of Eight-Bit Heathkit Computerists (SEBHC) [3] to interface the modern USB capability to the vintage Heathkit H8 and 89 computer platforms.

History

In early 2009 a discussion thread began on the SEBHC list [4] around ways to adapt vintage computers to store data using solid-state storage mechanisms such as Compact Flash and USB flash drives.

The need for this is real. Legacy machines like the Heathkit H8 and H89 were originally designed to store files on floppy disks, however the disk drives and floppy disk media from 40+ years ago can no longer be trusted for use in any serious work due to the unreliability of the hardware and media. The oxide coating on many of these old disks has become fragile, and hardware and media failures are a common problem. Although it may be possible to find “new old stock” (NOS) floppy disks, new media have not been manufactured for many years.

Modern memory sticks would seem to provide an ideal solution. They offer capacities tens of thousands of times bigger than the original H17 floppy disk and have proven to be rugged, inexpensive, reliable tools for storing and exchanging files. A single memory stick could store all the software ever developed for the H8/H89. The ability to store and exchange files using flash media has obvious value.

The challenge of enabling an H8 or H89 to access USB flash drives seemed like a daunting one. The USB Host interface is very sophisticated, as it manages all the details of the USB protocols. Implementing that kind of functionality in a microcomputer such as the H8 or H89 would be prohibitively complex. What was needed was a “smart” device.

The FTDI VDIP1

An article in the July 2007 issue of Circuit Cellar magazine [5] provided a possible solution. It described a device called the “VDIP1 Vinculum VNC1L Module” from Future Technology Devices International (FTDI). The article demonstrated how the VDIP1 could be used to add USB capability to legacy MS-DOS-based personal computers, and provided a reference software implementation written in Turbo C. The implication for microcomputer and embedded systems was clear: it was now possible, with relatively simple software, to access USB storage devices using the VDIP1.

Here is a brief, edited description of the device based on material taken from the FTDI datasheet [6]:

“The VDIP1 module is a microcontroller-to-embedded USB host controller development module. It is supplied on a printed circuit board designed to fit into a 24-pin dual in-line package (DIP) socket. The module is based on the Vinculum VNC1L integrated circuit (IC), which is the first of FTDI’s Vinculum family of embedded USB host controller ICs.”

“Not only is it able to handle the USB Host interface, and data transfer functions, but owing to the built-in microcontroller and embedded flash memory, Vinculum can encapsulate the USB device classes as well. When interfacing to mass storage devices such as USB flash drives, Vinculum also transparently handles the File Allocation Table (FAT) file structure, communicating via universal asynchronous receiver-transmitter (UART), Serial Peripheral Interface (SPI) or parallel first-in first-out (FIFO) interfaces via a simple command set. Vinculum provides a new cost-effective solution for providing USB Host capability into products that previously did not have the hardware resources available.”

This was a breakthrough capability, as the complexity of the USB protocol had previously precluded implementation on most microcomputer platforms.

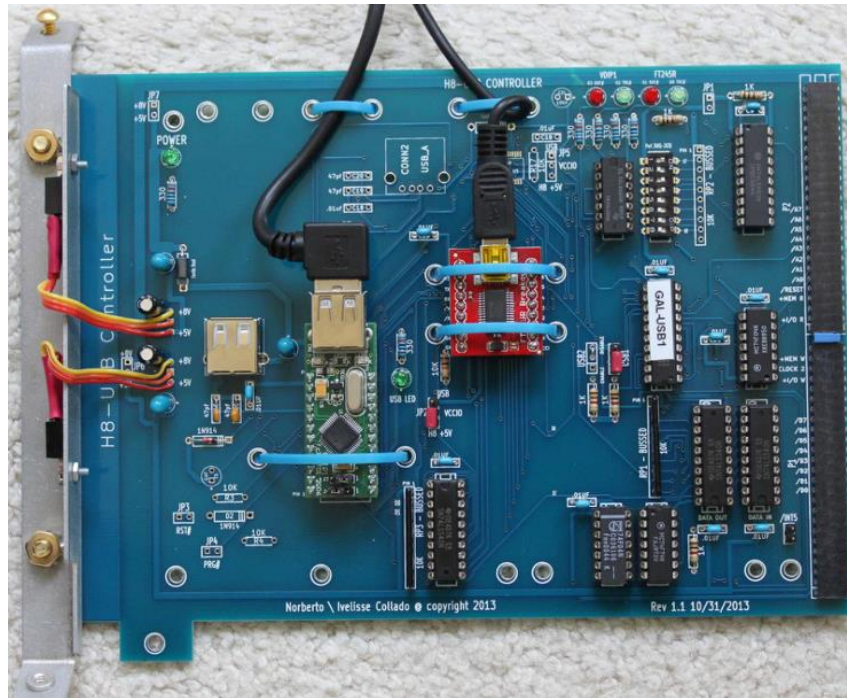


The FTDI VDIP1 module.

Around 2013 Norberto Collado developed the first USB interface board for the H8 [7]. It incorporated two USB interfaces: the VDIP1 and the FTDI FT245R [8]. The latter is also a USB to parallel FIFO device, but with different capabilities, as described below.

For the VDIP1 device Norberto had to choose among three different interfacing options: UART, SPI, or parallel FIFO. The Circuit Cellar application (for the legacy MS-DOS PC) used a UART interface via a standard serial port. The SPI approach is attractive for modern microcontrollers, which inherently support SPI protocols. For the H8/H89 application a parallel FIFO implementation offered speed and simplicity. The interface board provides address decoding and a bus interface, allowing the user to access the device via IN and OUT machine instructions. Both the VDIP1 and FT245R devices were interfaced in this fashion.

Two USB interfacing options (VDIP1 and FT245R) were included because both offer value, in different ways. The FT245 does not implement the USB Host protocol, so the H8/H89 appears as a Peripheral device to any attached Host system such as a personal computer.



The first H8-USB interface board.

The Software

To be useful, this new board required support software for the two predominant operating systems on the H8/H89: HDOS and CP/M. Two sets of software have been developed to support this initial board: one for the FT245R interface and one for the VDIP1 interface.

FT245R Software

At the time Norberto produced the H8 USB interface board Dan Emrick and Chris Elmquist had already been developing a client-server file sharing solution using the FT245. Their original work was done on an H89 using a proto-board and a SparkFun breakout board [9]. Because the FT245 presents the H89 as a peripheral, host software had to be developed on the PC end. Dan used the Qt development system and Slackware Linux to create his program. The program, which he eventually called Heathlink [10], let the PC host up to four disk images, in H8D format. An HDOS device driver on the H89 let the user mount and dismount images as needed. All disk I/O operations were handled by the device driver, translating them into high-speed data transfers between the host PC and the FT245 over the USB connection. In addition to remote disk mounting, Heathlink incorporated elements of Dwight Elvey's H89LDR2 [11] utility to allow for individual file transfers over the USB link.

VDIP1 Software

Since I had encouraged Norberto to include the VDIP1 in his USB board design, I took on the task of developing software to support it. I had the benefit of a starting point, in the form of the reference software implementation published in the Circuit Cellar [5] article.

As mentioned in the FTDI description of the VDIP1 (above), the device communicates via a “command set” – a simple ASCII-based interfacing protocol to the file system, documented in the Vinculum Firmware User Manual [12]. The commands are like MS-DOS commands and allow you to read and write files, perform directory operations, create and navigate subdirectories, and other useful functions on the USB flash drive. The simplicity of the ASCII-based command interface makes it feasible to develop software for embedded and microcomputer systems, like the H8 and H89.

I considered writing a device driver for the HDOS implementation, however the HDOS directory structure is completely different from the FAT-based structure on USB devices. This meant that I would have to emulate HDOS disk images using a single USB-based file, and (due to Vinculum firmware restrictions) only one such device could be open at a time. This would mean that a PC-based program would still be needed to insert files into the HDOS disk image files. This largely defeats the purpose, which was to freely exchange files with other platforms using a flash device. Therefore, I chose to implement a set of utility programs to perform the various functions. I chose to name these all starting with the letter “V,” reflecting FTDI’s “Vinculum” architecture branding.

My first application was a simple program: VTALK which connects the H8/H89 console directly to the VDIP1. This allowed me to explore and understand the command set. Subsequently I developed two programs: VPUT and VGET. These let one send and retrieve files (respectively) to/from the USB flash device. I eventually developed a full-blown Vinculum Peripheral Interchange Program (VPIP). VPIP largely includes the functionality of the other programs and may be the only program most users need. I have documented these programs in a small user manual entitled “The Vinculum Utilities for the H8 and H89 Computer” [13] and the software is available for download via the SEBHC GitHub site [14].

As a side product of this work, I developed a set of general-purpose routines for interfacing to the Vinculum firmware.

Doing More with USB

Following the release of the H8 USB board, Norberto developed a similar capability for the H89 [15]. More recently two new CPU boards have been developed for the H8 (Rev. 3.1 and 4.0) [16, 17] and these accommodate a serial I/O daughterboard that can contain the VDIP1. Support for the VDIP1 was also included in a recent disk interface board [18].

Although it has now been around for almost 15 years, the VDIP1 continues to be a good choice for this application and is still widely available from popular electronics distributors [19, 20]. FTDI did update their design to the VDIP2 [21] but the part is more expensive, uses a different form factor (40-pin DIP), and is more difficult to get than the older VDIP1. The VDIP2 does not appear to offer significant new functionality, therefore the SEBHC designs have stuck with the original VDIP1.

The latest Z80 board for the H8 (Rev. 4) can use either the legacy system ROM images from Heathkit and Magnolia Microsystems (MMS), or a newly developed ROM that enhances the legacy functionality. One of the features of this new ROM is the ability to “boot” the system from a file on the USB flash drive. While currently this does not support the booting of entire operating systems, it does make for a very handy way to run diagnostics and other programs that don’t require full operating system support. The program must be stored on the flash drive in “.SYS” format. I have written a small program to convert from CP/M hex format to .SYS format. This allows the standard CP/M assembler to be used to create .SYS programs. Programs that run this way must, of course, be stand alone (no operating system calls).

The VDIP1 includes an ability to talk to a second USB device. This is intended to allow support for accessories such as printers. To date we have not taken advantage of this capability, but this indicates that there are still interesting avenues to explore in the USB arena!

So that’s it for this issue. Hopefully this information, and the accompanying resources, will either stimulate your interest in adding USB to your Heathkit computer, or fill in some of the gaps in your understanding. It has been a fun and rewarding project to work on.

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

References

- [1] USB, *Wikipedia*, <https://en.wikipedia.org/wiki/USB>
- [2] USB Flash Drive, https://en.wikipedia.org/wiki/USB_flash_drive
- [3] Society for Eight-Bit Heathkit Computerists (SEBHC), <https://sebhc.github.io/sebhc/>
- [4] SEBHC Google Group <https://groups.google.com/g/sebhc>
- [5] Hyde, John, 2007, "Flash Drive Connection – Add a Flash Drive to Your Next Design," *Circuit Cellar*, Issue 204, July 2007, pp. 59-67.
- [6] Future Technology Devices International, "VDIP1 Vinculum VNC1L Module Datasheet," https://ftdichip.com/wp-content/uploads/2020/08/DS_VDIP1.pdf
- [7] Collado, Norberto, 2013, "H-8 USB Board", http://koyado.com/Heathkit/H-8_USB.html.
- [8] Future Technology Devices International, "FT245R USB FIFO IC Datasheet," http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245R.pdf
- [9] Sparkfun, "USB to FIFO Breakout - FT245RL," breakout board #07841 (now retired), <https://www.sparkfun.com/products/retired/7841>.
- [10] Emrick, Dan, "H89 Projects Downloads – USB Adapter," <http://danemricksite.net/downloads/usbproject.html>
- [11] Elvey, Dwight, "Disk Image Transfer Program," https://sebhc.github.io/sebhc/software/Utilities/H89LDR9_README.TXT
<https://sebhc.github.io/sebhc/software/Utilities/H89LDR9.zip>
- [12] Future Technology Devices International, "Vinculum Firmware User Manual," https://www.ftdichip.com/Firmware/Precompiled/UM_VinculumFirmware_V205.pdf
- [13] Roberts, Glenn, "The Vinculum Utilities for the H8 and H89 Computer," https://sebhc.github.io/sebhc/software/Utilities/Vinculum_USB/Vinculum%20Utilities.pdf
- [14] Roberts, Glenn, Vinculum Utilities Software V. 4.0, <https://sebhc.github.io/sebhc/software.html#Utilities>
- [15] Collado, Norberto, 2014, "H-89 Serial/USB - FT245R/VDIP1 Controller", http://koyado.com/Heathkit/H-89_USB_Serial.html.
- [16] Collado, Norberto, 2019, "Heathkit H8 Z80 CPU Board V3.1", <http://koyado.com/Heathkit/H8-Z80-64K-RTC-ORG0-V3.html>.

[17] Collado, Norberto, 2021, "Heathkit H8 Z80 CPU Board V4", <http://koyado.com/Heathkit/H8-Z80-64K-RTC-ORG0-V4.html>.

[18] Collado, Norberto, 2021, "H8 Storage Controller", <http://koyado.com/Heathkit/H8-H17-H37-H67-USB.html>.

[19] Digi-Key Electronics, "VDIP1", Part number 768-1001-ND, <https://www.digikey.com/en/products/detail/ftdi-future-technology-devices-international-ltd/VDIP1/1836379>.

[20] Mouser Electronics, "FTDI VDIP1", Part number 895-VDIP1.

[21] Future Technology Devices International, "Vinculum USB Host/Device Controller Development/Prototype Module, 2 USB Port", <https://ftdichip.com/products/vdip2/>.