

VPIP (Vinculum Peripheral Interchange Program)

Version 3.2 for CP/M 3

March 2020

Glenn Roberts

glenn.f.roberts@gmail.com

With the availability now of a Universal Serial Bus (USB) interface card for the Heathkit H8 and H89 computers (<http://www.koyado.com/>) these classic machines now have the ability to access files on USB media such as memory sticks. This is a significant step forward for these computer systems, which previously have relied on floppy-disks for removable storage and backup. The heart of this USB capability is the VDIP-1 interface module from Future Technology Devices International (<http://www.ftdichip.com/>) running the Vinculum VNC1L Firmware. This module and firmware implement all the complex details of the USB interface protocol. The purpose of the VPIP program is to provide a convenient interface to the USB device from the CP/M command prompt. This version, written expressly for CP/M 3, takes advantage of features of the OS such as file time/date stamping.

VPIP is patterned after the PIP utility provided with CP/M, however it only implements disk-to-disk copy and file listing (directory) commands. You'll still need PIP to delete, rename or copy files to the screen or printer. Similarly to manipulate (copy, delete, rename, etc.) files on the flash drive you'll need to use a personal computer or other device. VPIP currently can only be used to access *storage* devices via the USB port, not printers, keyboards or other accessories.

You can run VPIP two ways: with a single command provided on the command line, or interactively. To run VPIP interactively type VPIP at the command prompt:

```
>VPIP ↵  
:V:
```

The `:V:` prompt will be displayed at the left margin of the system console whenever the VPIP program is awaiting input. To exit VPIP, type CTRL-C, CTRL-Z or simply enter a blank line.

VPIP refers to the USB device via a "pseudo device" designated USB:. In VPIP commands this looks and acts like a CP/M device would.

Copying Files

The general form of the command for copying files specifies a "destination" followed by an "=" and then one or more "source" specifications:

```
:V:x:DESTINAT.EXT=USB:SOURCE.EXT ↵
```

or

```
:V:USB:DESTINAT.EXT=x:SOURCE.EXT ↵
```

Where 'x' is a one-letter CP/M drive designator. VPIP can only be used to copy files from a CP/M storage device to the USB device *or* from the USB device to a CP/M storage device. As an example:

```
:V:USB:MYPROG.BAK=A:MYPROG.FOR ↵
```

1 Files Copied

In this case, the destination is a file named `MYPROG.BAK` on the USB device and the source file is a file called `MYPROG.FOR`, located on the `A:` drive.

You can omit storage device specifications and VPIP will attempt to do the right thing. For example:

```
:V:A:*.*=MYPROG.C ↵
```

Will cause VPIP to assume that the USB device is the source device (since the destination is an HDOS disk), and will look on the USB device for the program `MYPROG.C` and copy it to `A:`. If you specify only the USB device and not the system device, VPIP will assume `SY0:`, for example:

```
:V:*.*=USB:MYPROG.C ↵
```

Will look on the USB device for a file `MYPROG.C` and copy it to the current default drive.

If you omit both source and destination devices VPIP assumes the source device is the USB drive and the destination device is the current default drive. For example:

```
:V:=*.c ↵
```

Would copy all files on the USB device matching the file specification `"*.c"` to the current drive. *It is important to note that currently VPIP currently does not check whether a file already exists,* so the above command would (without any warning) *overwrite* (replace) any existing CP/M files on the default drive with files of the same name on the USB device.

The following are some examples of *illegal* VPIP commands:

Command	Reason for being illegal
<code>USB:*.*=USB:TEST.*</code>	USB to USB transfer not supported
<code>A:TEST.DAT=B:MYTEST.DAT</code>	Either source or destination needs to be USB:
<code>TT:=USB:MYPROG.C</code>	VPIP can only copy to/from storage class devices

Wildcards and Multiple File Designation

Wildcards

The “*.*” wildcard is another way of accessing multiple files. A “*” can be substituted for the file name or extension portion of a file specification, for example:

B:*.EXT

or

USB:FNAME.*

or

A:*.*

are all valid uses of the “*” wild card. You can also use “*” to complete a field. For example

USB:V*.*

Will match any file on the USB flash drive that starts with the letter “V”. If the “*” character starts a name or extension field *the characters beyond it in that field will be ignored*. So

A:*V.*

Is the same as

A:*.*

The “?” wild card can be used to match single letters in a portion of a file name. For example

CHAPTER?.DOC

Will match CHAPTER1.DOC, CHAPTER2.DOC, etc...

If you use “?” in a portion of a file designation you must use at least as many “?”s as there are characters in the name of the file you want to match. Thus

????.*

will match all files whose name contains *four or fewer* characters in the name portion of the filename. The file specification “?????????.???” is identical to “*.?”.

As a convenience, when copying files from one device to another you may omit the “*.?” altogether for the destination device – it will be implied. For example

:V:USB:=*.C

Will copy all “C” files on the current drive to the USB drive.

Listing Files (Directory)

VPIP also provides a way to simply list files on either the CP/M drive or the USB drive. This feature only works from the command line (not the interactive prompt). To specify a listing request include the switch “-l” (separated by at least one space. For example:

```
VPIP USB:*. * -l
```

or

```
VPIP B:*. * -l
```

Specifying an Alternate Port

VPIP assumes the USB card is set to use ports 331/332 (octal). If your card is set to a different port number you can specify that on the command line using the “-p” switch and specifying the port number of the *first* of the paired I/O ports *in octal* with no space between the switch and the port number, for example:

```
VPIP USB:=B:*.C -p261
```

If you specify the incorrect port number the utility will time out after several attempts and print the message:

```
Error initializing VDIP-1 device!
```

Notes on Version 3.2

1. Files copied to the USB device are time stamped with the current time and date returned by the CP/M 3 BDOS.
2. If a CP/M file system becomes full during a copy operation, the last file written will be closed even though it is only partially copied.
3. There is currently no special handling of Control-C. If you Control-C during a file transfer you may leave the USB interface in an unknown state that requires a reboot.

Future Plans

1. Support for file concatenation
2. Support for text devices (e.g. TT:, LP:, etc.)
3. Support for subdirectories on the USB device.
4. Warning before overwriting files.
5. Better error handling (including capturing Control-C interrupts).
6. Ability to use CP/M file time/date stamp (either “access” or “update”) when transferring to the USB device (useful for tracking backups).