

Hexer: Helios Floppy Drive Exerciser
Version 1.04
User's Manual

Martin Eberhard June 5, 2015

Revision History

Program Revision	Date	Author	Comment
1.00	2 JAN 2015	M. Eberhard	Created
1.01	15 MAY 2015	M. Eberhard	Detect Solos vs. Cuter during initialization, and set CPU clock value accordingly. Account for 1 wait-state during I/O cycles, imposed by the Sol-20. Interleave block numbers when formatting the PTDOS directory track. Also tidy up this manual and add discussion of Helios format.
1.02	21 May 2015	M. Eberhard	Add BO (boot) command. Improve timing when reading block 0. (Prior versions would sometimes miss.)
1.03	22 May 2015	M. Eberhard	Add DF (disk format), ID (Intel hex dump), and IL (Intel hex load) commands.
1.04	5 June 2015	M. Eberhard	Add BC command. Fix bug when selecting units above 1. Improve formatting. Default to decimal mode.

1.0 Description

Hexer is a Helios floppy drive exerciser program that runs under Processor Technology's Solos or Cuter monitor. Hexer is designed to exercise a Persci 270/277 disk drive, via the Helios disk controller board set.

Hexer does not require PTDOS, so that you can use it to align a Helios disk drive even when the Helios disk drive will not boot PTDOS. However, Hexer can be loaded and run from cassette tape or from PTDOS disk. It can also be loaded via the serial port, using a serial port loader program, such as HLOAD.

1.1 System Requirements

An 8080-based system running at one of the following frequencies:

- o 2.00000000 MHz, with 0 I/O waitstates (e.g. an Altair)
- o 2.04542857 MHz, with 1 I/O waitstate (e.g. a standard Sol-20)

Processor Technology-compatible I/O devices for the console

A Sol-20/CUTS cassette tape system, if your Helios drive won't boot Solos or Cuter system monitor program, located at the standard address, C000h

The above requirements imply either a Sol-20 or an 8080 system with Processor Technology's Subsystem B boards.

At least 13K of 0 wait-state RAM starting at address 0000

A Helios floppy disk controller board set at the standard Helios port addresses, connected to the Helios drive

1.2 Key Features

Position control of the read/write head (seeks)

Sequential seeks between multiple tracks, useful for alignment operations

Fast (full-speed) and slow (track-at-a-time) seeks

Whole-track pattern-write command specifically for tuning the read channel and data separator

Block header read and write commands, using a separate header buffer

Block header editing command, with decoding of header components

Block read and write commands, using a separate block data buffer

Data Buffer editing, so you can create or edit any sector's data

Head load/unload control

Track 0 sensor monitoring

Seek time measurement

Spindle rotational time measurement

Automatic detection of a 2.054 MHz Sol-20 (with Solos) versus a 2.000 MHz S-100 computer with Processor Technology's Subsystem B (with Cuter)

2.0 Hexer Operation

To initiate Hexer from Solos/Cuter, insert the cassette tape in your tape drive, and type XEQ HEXER at the Solos/Cuter prompt. To initiate Hexer from PTDOS, simply type HEXER at the PTDOS prompt.

Once Hexer loads, it will print its sign-on banner, and then the Hexer prompt, '%'. You can type '?' at the prompt to see its list of help screens. You can also type 'SE' to see Hexer's current settings, and the status of the selected unit, if any is selected.

2.1 Buffers

Header Buffer. Hexer contains a 13-byte buffer for manipulating block header data. For details about the organization of block headers, see *PTDOS 1.5 User's Manual*, page 5-4.

Data Buffer. Hexer contains a 4095-byte buffer for manipulating block data. All block read and write operations start with buffer address 0000, using as much buffer data as necessary for the current block size. You can display and edit the entire 4095-byte Data Buffer, regardless of the current block size.

Note that the WT command will overwrite the Data Buffer.

2.2 General Commands

% ? *Display the General help screen*

% BO *Boot from disk 0*

% QU *Quit to Solos/Cuter*

2.3 Hexer State Commands

% ?S *Display the Hexer State Command help screen*

% SE *Display Hexer settings and state. This will display all of the user settings (see below), as well as the drive's state, including:*

- 8080 CPU speed setting
- Verbose mode on/off
- Decimal or hex input for track and sector numbers
- Selected unit
- Current track
- Drive ready/not ready
- Spindle rotation time

% SE C={0/1} *Set CPU Frequency for timing loops. The choices are:*

<u>Setting</u>	<u>Frequency</u>	<u>Waitstates</u>	<u>Computer</u>
0	2.00000000 MHz	0	e.g. an Altair
1	2.04542857 MHz	1	Standard Sol-20

Note: The frequency is initialized to 2.04542857 MHz if Solos is detected. Otherwise, it is set to 2.000000 MHz.

- % **SE D={0/1}** *Set decimal input mode.* DE=0 selects hex; (Default is decimal.) DE=1 selects decimal input for track and block numbers. (All other numbers are always input in hex.)
- % **SE U={0-7}** *Set the disk unit.* Note that each Persci 277 drive contains 2 units. The even units are on the left.
- % **SE V={0/1}** *Set verbose mode off/on.* (Default is on.) When on, the track number and seek time are displayed after every seek operation, including when in automatic mode. The time between seeks is much less when verbose mode is off.

2.4 Actuator and Sensor Commands

- % **?A** *Display the Actuator and Sensor Command help screen*
- % **HL** *Head-load test mode.* Press Space to toggle the head in the selected unit. To exit head-load test mode, type 'Q' or Mode/Select. The head will automatically unload after exiting head-load mode.
- % **MS** *Spindle motor measurement.* Hexer will continuously measure and report the spindle's rotational time. To exit spindle motor test mode, type 'Q' or Mode/Select.

2.5 Head Positioner Commands

For these commands, track numbers <Tn> are either hex or decimal, depending on the mode set by the 'SE D=' command.

- % **?P** *Display the Positioner Command help screen*
- % **RE** *Restore the selected unit to track 0.*
- % **SK** *Seek previous track(s).* This will seek the same track or list of tracks from the previous SK or ST command. If there is no previous track, then it will seek track 0.
- % **SK <T>** *Seek track <T>.* This will issue a fast (voicecoil) seek. If Verbose Mode is enabled, then the seek time and ending track number will be displayed.
- % **SK <T1> <T2> ...<Tn>** *Sequentially seek tracks.* If Verbose Mode is enabled, then the seek time and ending track number will be displayed after each seek. If Automatic mode is enabled, then this command will seek rapidly through the list of tracks until told to stop. If Manual mode is selected, then seeking will pause at each listed track until Space is pressed, to proceed to the next track. Type 'Q' or Mode/Select to quit. Type 'A' to switch to automatic mode, and 'M' to switch to manual mode.
- % **ST** *Step to previous track(s).* This will step (one track at a time) to the same track or list of tracks from the previous SK or ST command. If there is no previous track, then it will step to track 0.
- % **ST <T>** *Step to track <T>.* Step (one track at a time) to the specified track. If Verbose Mode is enabled, then the total step time and ending track number will be displayed.

- % **ST** **<T1> <T2> ...<Tn>** *Sequentially step to the listed tracks.* In this mode, Hexer will wait until the head has arrived at every track on its way to a destination track, before moving on. If Verbose Mode is enabled, then the total step time and ending track number will be displayed after each seek. If Automatic mode is enabled, then this command will step continuously through the list of tracks until told to stop. If Manual mode is selected, then stepping will pause at each listed track until Space is pressed, to proceed to the next track. Type 'Q' or Mode/Select to quit. Type 'A' to switch to automatic mode, and 'M' to switch to manual mode.
- % **ST I** *Step in one track (toward higher track numbers).* Stepping will change direction at track 76. 'A' will cause automatic stepping; 'M' cancels automatic stepping.
- % **ST O** *Step out one track (toward lower track numbers).* Stepping will change directions at track 0. 'A' will cause automatic stepping; 'M' cancels automatic stepping.

2.6 Read/Write Commands

PTDOS accesses disks using a technique called "firm sectoring," described in section 5.3 of the *PTDOS 1.5 User's Manual*. The disk itself has 32 hard sector holes, of which only the 16 even holes are used to mark the beginning of PTDOS sectors. PTDOS partitions data into "blocks," which start on a sector boundary, but may span more than one sector. However, sector 0 must be the beginning of a block.

Each block has identifying information written into the sector header at the beginning of the block. However, sector headers in the middle of blocks (including their sync marks) are overwritten, and cannot be found by the controller hardware. (See *Helios II Disk Memory System Manual*, Section 7.4, starting on page 7-6.)

For these commands, block numbers are either hex or decimal, depending on the mode set by the 'SE D=' command.

% **?R** *Display the Read/Write Command help screen*

% **BC** ** [<C>]** *Compare block on the current track to the data in the Data Buffer starting at buffer address 0000. Hexer will read the header of the specified block to determine how many sectors it spans and how many bytes the block contains. Optional parameter <C> specifies the maximum number of bytes to compare, and the compare will be limited to this number of bytes if it is less than the number of bytes specified in the header. Hexer will report the number of bytes read from the block, as well as the number of bytes that did not match.*

Each mismatching data byte will be reported, showing the address, the expected data, and the data found in the block.

A disk error message will be printed in any of the following circumstances:

- The track number in the block header does not match Hexer's track counter
- An abort or CRC error occurs while reading the sector header or the sector data
- Any timeouts waiting for the controller

% **BR** **** [**<C>**] *Read block from the current track into the Data Buffer starting at buffer address 0000. Hexer will read the header of the specified block to determine how many sectors it spans and how many bytes the block contains. Optional parameter <C> specifies the maximum number of bytes to read, and the read will be limited to this number of bytes if it is less than the number of bytes specified in the header. Hexer will report the number of bytes read from the block.*

An error message will be printed in any of the following circumstances:

- The track number in the block header does not match Hexer's track counter
- An abort or CRC error occurs while reading the sector header or the sector data
- Any timeouts waiting for the controller

% **BW** **** [**<C>**] *Write block data on the current track, with data from the Data Buffer starting at buffer address 0000. Hexer will read the header of the specified block to determine how many bytes of data to write to the block. Optional parameter <C> specifies the maximum number of bytes to write, and the write will be limited to this number of bytes if it is less than the number of bytes specified in the header. Hexer will report the number of bytes written to the block.*

No write will occur and an error message will be printed in any of the following circumstances:

- The track number in the block header does not match Hexer's track counter
- A CRC error occurs while reading the sector header
- Any timeouts waiting for the controller

The maximum bytes in a block depends on how many sectors the block spans. (See *PTDOS 1.5 User's Manual* page 5-4):

Maximum Bytes			Maximum Bytes		
Sectors	(decimal)	(hex)	Sectors	(decimal)	(hex)
1	256	100	8	2496	9C0
2	576	240	9	2816	B00
3	896	380	10	3136	C40
4	1216	4C0	11	3456	D80
5	1536	600	12	3776	EC0
6	1856	740	13	4095	FFF
7	2176	880			

% **DF** **<H>** *Disk Format.* For each track on the entire disk, this performs an Erase Track function, and then writes the correct track and sector numbers into all headers, and 256 bytes of **<H>** to the block data fields. On PTDOS's directory track (track 25), the block numbers are written in an interleaved pattern: 0,8,1,9,2,10, 3,11,4,12,5,13,6,14,7,15. Otherwise, the block numbers are sequential. (This interleave is not documented in the *PTDOS 1.5 User's Manual*. However PTDOS's DISKCOPY command does this interleave for the directory track when formatting a disk.)

Note that this command creates an incomplete format because several necessary files are still missing on the disk: DIRECTRY,FSMAP,NEXTID, and SYSGLOBL. Without these files, PTDOS still cannot use the disk.

This command takes about 8 minutes.

% **HR** **** *Read Block Header * from current track. Hexer will read the specified block header into the Header Buffer, and display the result. (See *PTDOS 1.5 User's Manual*, section 5.3.) The Block Header is then displayed via the HE command.

% **HW** **** *Write Block Header * on current track. Hexer will write the Header Buffer into the specified block's header. This command will calculate how far from the last sector mark is the end of the data field. If it is less than 108 bytes from the sector mark, then the length (byte count) will be padded to 108 bytes long, to meet the Helios controller's spec.

% **TE** *Track Erase.* (See *Helios II Disk Memory System Manual* page 7-36.) All header bytes are written as FFh, and the data fields have one byte of FFh. This command will create 1-byte data fields, in violation of the Helios controller's requirement that a data field must end at least 108 bytes past the sector mark. Also, the track and block numbers are not correct (for PTDOS) in the headers.

% **TF** **<H>** *Track Format.* This performs an Erase Track function, and then writes the correct track and sector numbers into all headers, and 256 bytes of **<H>** to the block data fields. (Note that the ET command will create 1-byte data fields, in violation of the Helios controller's requirement that a data field must end at least 108 bytes past the sector mark.) If the current track is PTDOS's directory track (track 25), then the block numbers are written in an interleaved pattern: 0,8,1,9,2,10, 3,11,4,12,5,13,6,14,7,15. Otherwise, the block numbers are sequential. (This interleave is not documented in the *PTDOS 1.5 User's Manual*. However PTDOS's DISKCOPY command does this interleave for the directory track when formatting a disk.) This command takes about 8 seconds.

% **TW** <H1> <H2> ... <Hn> *Track Write with pattern.* This command creates a track of data that is useful for inspecting and tuning the read channel. The entire track is written with the specified sequence of hex values, with only a few bytes of other values.

This command also overwrites the Data Buffer with the specified hex value pattern.

2.7 Buffer Manipulation Commands

For these commands, all numbers are input in hex.

% **?B** *Display the Buffer Command help screen*

% **BD** <A> *Display the Block Data Buffer* from address <A> through address . <A> and are specified in hex. (The data buffer is used for BR and BW commands.) This displays the specified portion of the data buffer, 16 bytes per line. Pause the display with the Space bar. 'Q' or Mode/Select will end the display.

% **BE** <A> *Edit the Block Data Buffer* starting at address <A>. <A> is specified in hex. If no address is given, then editing begins at 0000. <A> must be no higher than 0FFFh. To leave a value unchanged, press Space. To change the value, enter a new value, and then press Space. Each time you press Space, the contents of the next buffer address are displayed for editing. To quit editing, type 'Q' or Mode/Select.

% **BF** <H> *Fill the Block Data Buffer* with hex value <H>.

% **ID** <C> *Intel Hex Dump Buffer.* Hexer will send the first <C> bytes from the Block Buffer to the serial port in Intel hex format. <C> is specified in hex. If <C> is not specified, then the value will be taken from the byte count in the Header Buffer. If <C> equals 0 or is greater than 0FFFh then the entire Block Buffer will be dumped.

% **IL** <O> *Intel Hex Load Buffer.* Hexer will load Intel hex records from the serial port into the Block Buffer until an end-of-file record is received. Address offset <O> is subtracted from every record's address to compute the buffer location for the record's data. If the resulting address is not within the buffer, then the data is simply ignored. Bad hex or a checksum error will abort the load.

% **HE** *Display the Header Buffer.* The first column is the Header Buffer item number. The second column is the item definition. The third is the value in hex, and the fourth is the same item's value in decimal.

```

0          Sector: hh    dd
1          Track:  hh    dd
2      FBCP Sector: hh    dd
3      FBCP Track:  hh    dd
4      BBCP Sector: hh    dd
5      BBCP Track:  hh    dd
6          File ID: hhhh dddd
7  Length (secs):  hh    dd
8  Length (bytes): hhhh dddd
9      Reserved:  hhhh dddd

```

% **HE** <i> <H> *Edit Header Buffer.* Writes value <H> to Header Buffer item <i>. <H> is specified in hex. See HD command for Header Buffer item numbers.

3.0 Helios "Firm Sectoring" Disk Format

The Helios controller ignores every other hard sector hole, dividing the disk into 15 "sectors," numbered 0 to 15. The sector that follows the index hole is sector 0.

Each track is divided into "blocks", where a block may be any number of bytes from 1 up to 4095, with certain restrictions (see below). As such, a block may span up to 13 sectors.

A block must begin at the beginning of a sector, and no block may span from sector 15 to sector 0. (In other words, sector 0 must be the beginning of a block.)

One poorly-documented "feature" of the Helios controller is that the last sector of a block cannot end fewer than 108 bytes after a sector mark. Special code in Hexer's BW command pads blocks to meet this requirement (similar to PTDOS). Note that the Helios controller's "Erase" command violates this rule, creating 1-byte block data fields. (Hexer's FT command fixes this by writing 256-byte data fields.)

Each block begins with a block header, which contains 13 bytes of data that may be used by the operating system. PTDOS uses these bytes to indicate the track and sector number, as well as for other PTDOS functions. The controller does not interpret any bytes within this header. Hexer's HE command displays the contents of the header, interpreting its bytes as PTDOS does.

The block header and block data fields are bracketed by preambles/postambles that include special sync marks, which are recognizable by the controller hardware. The headers and data fields are written by separate controller commands.

Track 25 is the PTDOS directory track. (See *PTDOS 1.5 User's Manual* page 5-5.) This track is the only track that has interleaved sector numbers written into its block headers. The interleave pattern is: 0,8,1,9,2,

10,3,11,4,12,5,13,6,14,7,15. This interleave is not mentioned in the *PTDOS 1.5 User's Manual*, but is implemented in the disk format portion of PTDOS's DISKCOPY program.

On a bootable disk, block 0 of track 0 contains the loader. This loader appears as an Image file called BOOTLOAD with a file ID of 0006 in the PTDOS directory. BOOTLOAD gets loaded into RAM at address 0 and executed there. Solos's boot code will always load 896 bytes (which works out to be exactly 3 sectors) when it boots. Hexer's BO command will look at sector 0, block 0's header to determine the boot loader's size, and will load the specified number of bytes.

BOOTLOAD is stored on disk as a PTDOS Image file, which has a 4-byte header at the beginning, and a CRC byte and execution address at the end. The boot code in Solos (invoked by the 'BO' command) loads BOOTLOAD, and starts execution at address 0 - executing the non-executable 4-byte header of the Image file. Fortunately, executing these particular 4 bytes has no ill effect.

On a bootable PTDOS 1.5 disk, block 1 of track 0 and block 0 of track 1 are each 4095 bytes (13 sectors) long. These two tracks contain the file RESIDENT, which is the PTDOS operating system's main program. The loader loads this program into memory starting at address A002h and executes it at address BCC0h.