SOFTWARE WIZARDRY, INC.
122 YANKEE DRIVE ST. CHARLES, MO. 63301
(314) 946-1968

The ULTRA ROM

Version 2.4

Table of Contents

Introduction

The ULTRA ROM set consists of two (2) erasable programmable read-only memory (EPROM) devices. One is a 2532 (4k) EPROM and the other is a 2716 (2k) EPROM. The 2532 device contains the actual modified firmware and the 2716 contains keyboard encoding information used by the keyboard encoder IC. If the user desires to utilize the second page of screen RAM, he will need to obtain one additional IC. This is an HM6116 2k by 8 static RAM (see page 3).

Parts of this firmware originated from the Heath Users' Group terminal firmware package, written by Barry Watzman of Zenith Data Systems. These parts are being distributed with the permission of the Heath Users' Group. I have optimized the code for the H19 terminal logic boards Z80 processor and more than doubled the number of features found in the HUG firmware. In addition, I corrected flaws in that code as they appeared. The most notable difference between this and the HUG firmware is the lack of ANSI compatible escape sequences in this ROM. The reasoning involved in the decision to delete the ANSI code is discussed in further detail in this manual. Suffice it to say that I think this is a First Class upgrade to the H/Z19 terminal, and unequaled by any similar products.

Credits, Etc.

CP/M is a registered trademark of Digital Research, Inc.
HDOS is a trademark of Heath Company.

Thanks to the following people who have contributed time and ideas to this project and helped debug the code, proofread the documentation, etc: Charles Sanborn, Dale Lamm, and Dave Kobets.

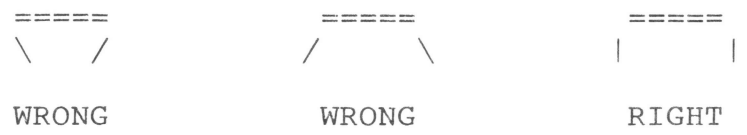
Richard Musgrave is responsible for the original draft of the appendix to this document and has put in considerable time and effort in helping to both debug and produce this firmware. Special thanks to him.

Installation

****
IMPORTANT -- READ THIS!

In this installation section, ICs and jumper strips on the terminal logic board will be referred to by their location as marked on the board itself. Locations given are for later (H19A-type) terminals. The old part designation will also be given in [brackets].

I probably don't have to tell you this, but before you install any IC into a socket, be sure that the pins have been bent perpendicular to the body of the chip (see below).



WRONG WRONG RIGHT

If, after installation, when you apply power to your terminal, strange noises are heard, or if the terminal does not function off line, IMMEDIATELY turn the terminal off and re-check all of the steps outlined under installation.
****

Installation of the new firmware requires that the H19 terminal logic board be removed. If you have a Heath/Zenith All-In-One (H88/H89/Z90) computer, you will need to gain access to this board done by removing the CPU board from the computer.

The ICs to be replaced are 24-pin packages located at U437 [U422] and U445 [U430]. U437 [U422] should be immediately below an empty socket U436 [U423]. In older (not -A) H19 terminals, U423 may contain an IC. If you have one of these early terminals with ICs at both U422 and U423, you should remove the device located at U423 and place it in a piece of conductive foam for safe keeping. In addition, the following jumpers on your terminal logic board must be changed to supply the proper address and power connections to the new EPROM. These jumper changes apply ONLY to those older (not -A) H19 terminals.

Jumper Strip	Remove Jumpers	Install Jumpers
JP20	1 - 2	2 - 3
JP21	3 - 4	1 - 2 4 - 5
JP22	2 - 3 5 - 6 7 - 8	1 - 2 4 - 5 6 - 7

Remove the device located at U437 [U422] and place it in a piece of conductive foam for safe keeping. Install the 2532 (firmware) EPROM at that location in its place. Be careful that none of the ICs pins are bent as it is inserted into the socket. Note that the orientation of the pin-1 end of the IC is toward the right side of the board.

Remove the device located at U445 [U430] and place it in a piece of conductive foam for safe keeping. Install the 2716 (keyboard encoder) EPROM at that location in its place. As above, be careful that none of the ICs pins are bent as it is inserted into the socket. The orientation of the pin-1 end of this IC should also be towards the right side of the board.

Second Page Display

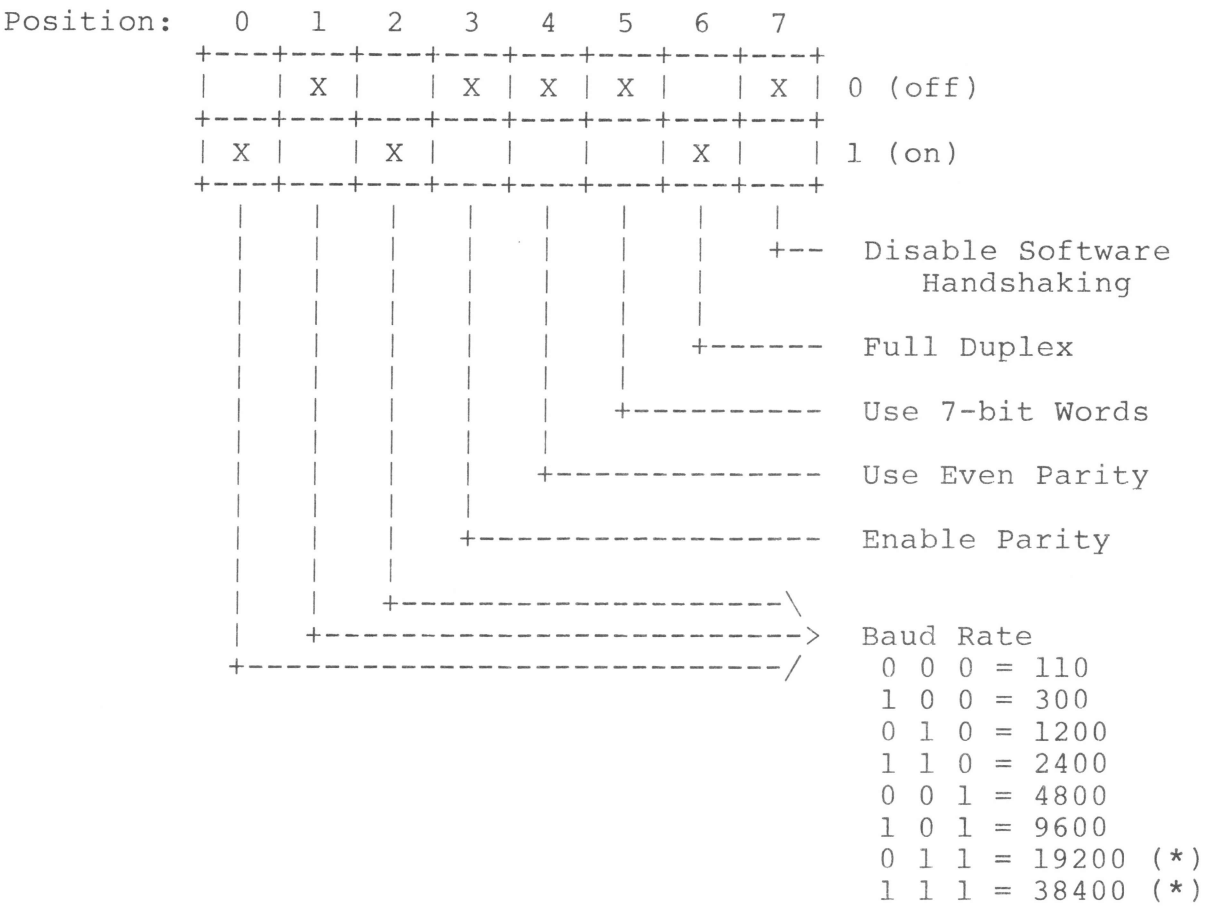
If you wish to implement the second page feature of this firmware, you will need an HM6116 2k x 8 static RAM (available separately for $14.95). This IC is a 24-pin package and is installed at U436 [U423]. Any jumpers installed at JP2 [JP23] should be removed. A jumper should then be installed between 6 and 7 on that jumper strip. In addition, the following wire jumpers must be installed on the terminal logic board. Before you proceed, carefully remove the Z80 CPU from U430 [U421] and place it in a piece of conductive foam so that it is not damaged. The following jumpers are to be installed on the BACK (foil side) of the terminal logic board. *CAREFULLY* solder a wire jumper from pin 21 of IC U430 [U421] to pin 20 of IC U436 [U423]. *CAREFULLY* solder a wire jumper from pin 22 of IC U430 [U421] to pin 21 of IC U436 [U423]. Check your work and be absolutely sure that there are no solder bridges. (Note: Wire-wrap wire works very well for the jumpers). When you are sure that the jumpers are in place properly, reinstall the Z80 CPU at U430 [U421]. Install the HM6116 static RAM at location U436 [U423]. Note the orientation of the pin-1 end of the device is to the right of the board as viewed from the front.

DIP Switches

The settings of the DIP switches S401 and S402 on the terminal logic board must be changed at this time. The following pages contain reference sheets which will guide you in setting these switches.

S401 - Communications

X = Recommended 'standard' settings

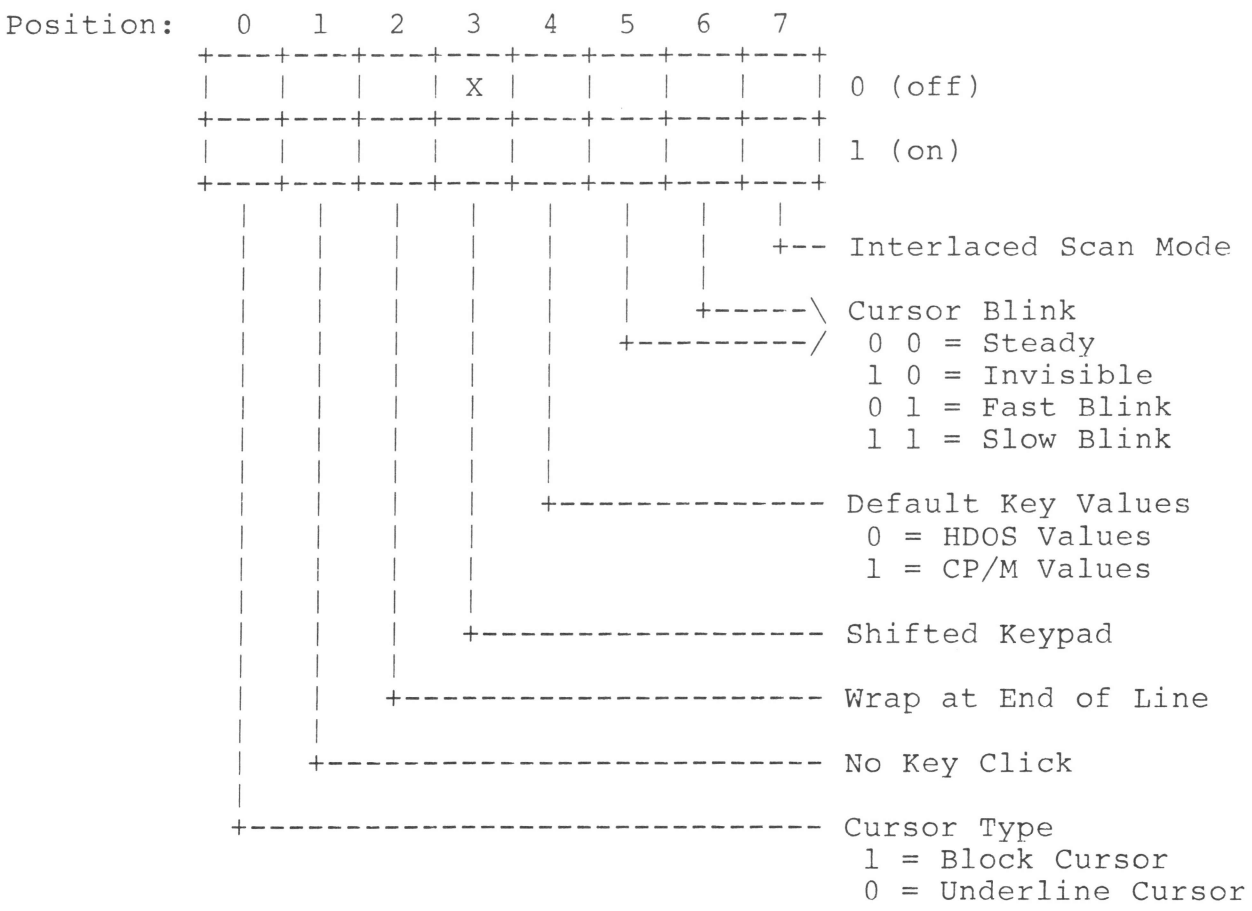


(*) Note:

For reliable operation at these high rates of speed, it is necessary to modify your software to use hardware handshaking. The necessary modifications for Heath/Zenith CP/M 2.2.03 and for HDOS 2.0 may be found in the back of this document. At any rate, there is little gain unless the computer's processor is operating at 4 MHz or faster. It is also necessary to modify your terminal clock rate to 3 or 4 MHz to utilize the higher transfer rates. This modification is not a simple change in all terminals and has therfore been omitted from this document. Users interested in modifying their terminal logic boards for higher clock rates should contact Software Wizardry for the necessary information.

S402 - Terminal Options

X = Recommended 'standard' settings



Operation

It is not the purpose of this document to describe each intricate detail of every ESCAPE sequence recognized by this new firmware. Since all of the original escape sequences of the H19 terminal have been retained (with the exception of the ANSI mode and it's associated sequences), a complete description of each would be redundant. The user is referred to the H19 operation manual for a more complete description. All of the *NEW* ESCAPE sequences will be described here.

Differences and Deletions

The most significant difference between this firmware and the original H19 firmware is the lack of the ANSI compatible ESCAPE sequences. I made the decision to omit them based on a number of factors. The most compelling reason is that nearly all software written for the Heath computers uses the standard VT-52 (Heath mode) sequences. (Actually, I am not aware of any software for the Heath computers which uses the ANSI mode, although I'm sure some must exist somewhere). I felt that the limited program area with which I had to work, and the value (to me) of the features which I wished to add, outweighed the importance of retaining the ANSI codes. The most significant drawback to this decision is that the H25 printer responds to the ANSI codes in a manner identical to the H19. I know of no one, however, who has capitalized on this fact, sooo ... no more ANSI.

I have reduced the size of the baud rate table to 8 entries. This was done to gain access to an additional DIP switch for setting up the terminal handshaking mode. The most commonly used baud rates have been retained. The baud rates found in the original H19 ROM are (with deleted entries in [brackets]): 110, [150], 300, [600], 1200, [1800], [2000], 2400, [3600], 4800, [7200], 9600, 19200. An additional entry for 38400 baud has been added.

As you may have noticed in the Installation section (or will notice when you get around to the installation section) I have changed the assignments of several of the terminal DIP switches. This was done in an effort to provide more control over those terminal options which I felt would vary the most from system to system. These include interlaced scan, hardware handshaking, and cursor display characteristics. Sacrificed were ANSI (discussed above), 50/60 hertz, Auto CR on LF, and Auto LF on CR. (Does anyone ever really use the last two?) The other changes to the DIP switches merely involve changes in the locations of certain switches. This was done for my benefit in working on the firmware and should be of no consequence.

Enhancements

I have added a number of functions to the firmware. Among these are: Self-test mode; which verifies the terminals RAM and ROM, Selection of cursor type and blink rates, Hardware handshaking for high speed operation, Interlaced scan mode, Single key entry to and exit from SCROLL mode, Back tab, Read terminal configuration, User programmable function keys (you determine what the key SENDS), An improved transparent mode, Alternate 'soft' terminal reset from the keyboard, Keyboard escape from transparent mode and keyboard disable, Line editing with the ability to send the edited line, Transmit current line, the ability to generate a 'click' under program control, Terminal status report, Reverse screen, Reverse field, Expansion horizontally and vertically (may be used for drawing 'lines'), Fill screen, and several predefined 'user' function key values.

The most significant addition is the support by this firmware of a second screen page of memory. This is accomplished by taking advantage of the facts that the H19 terminal processor board has an empty 24-pin ROM socket and that there exists a 2k x 8 static RAM device which is nearly pin for pin compatible with an EPROM of the type typically used in the H19. I have also re-written portions of the existing code in an effort to streamline the operation. Each of these features and their use will be described in detail.

Summary of Escape Sequences

This is a summary of all escape codes and sequences processed by the new firmware. Those marked with '*' are new to this firmware, and will be discussed later. Others are standard H19.

Cursor Functions

Escape Sequence	Definition
ESC H	Cursor Home
ESC C	Cursor Forward
ESC D	Cursor Backward
ESC B	Cursor Down
ESC A	Cursor Up
ESC I	Reverse Index (reverse line feed)
ESC n	Cursor Position Report
ESC j	Save Cursor Position
ESC k	Restore Cursor Position
ESC Y <ll> <cc>	Direct Cursor Addressing

Erasing and Editing

ESC E	Erase Display
ESC b	Erase Beginning of Display
ESC J	Erase to End of Page
ESC l	Erase Entire Line
ESC o	Erase Beginning of Line
ESC K	Erase to End of Line
ESC L	Insert Line
ESC M	Delete Line
ESC N	Delete Character
ESC @	Enter Insert Character Mode
ESC O	Exit Insert Character Mode

Configuration

ESC z — Reset to Power-Up Configuration

ESC r
 * Modify Baud Rate (A=110, B=300, C=1200, D=2400, E=4800, F=9600, G=19200, H=38400)

ESC x <mode> — Set Mode
- 1 = Enable 25th Line
- 2 = Disable Key Click
- 3 = Hold Screen Mode
- 4 = Block Cursor
- 5 = Cursor Off
- 6 = Keypad Shifted
- 7 = Alternate Keypad Mode
- 8 = Auto Line Feed on Carriage Return
- 9 = Auto Carriage Return on Line Feed

Configuration (continued)

Escape Sequence		Definition
ESC y <mode>		Reset Mode 1 = Disable 25th Line 2 = Enable Key Click 3 = Exit Hold Screen Mode 4 = Underscore Cursor 5 = Cursor On 6 = Keypad Unshifted 7 = Exit Alternate Keypad Mode 8 = No Auto Line Feed 9 = No Auto Carriage Return
ESC h <mode>	*	Set Mode 2 1 = Enable Software Handshaking 2 = Disable Software Handshaking 3 = Set Normal Scan Mode 4 = Set Interlaced Scan Mode 5 = Enable User Function Keys 6 = Disable User Function Keys 7 = Select CP/M Key Values 8 = Select HDOS Key Values 9 = Select MBasic Key Values
ESC S <cursor>	*	Set Cursor Type 1 = Underscore, steady 2 = Underscore, not displayed 3 = Underscore, fast blink 4 = Underscore, slow blink 5 = Block, steady 6 = Block, not displayed 7 = Block, fast blink 8 = Block, slow blink
ESC Z		Identify as VT-52
ESC ?	*	Send Configuration Report
ESC [		Enter Hold Screen Mode
ESC \		Exit Hold Screen Mode
ESC p		Enter Reverse Video Mode
ESC q		Exit Reverse Video Mode
ESC F		Enter Graphics Mode
ESC G		Exit Graphics Mode
ESC t		Enter Keypad Shifted Mode
ESC u		Exit Keypad Shifted Mode
ESC =		Enter Alternate Keypad Mode
ESC >		Exit Alternate Keypad Mode

Modes of Operation (continued)

Escape Sequence		Definition
ESC v		Wrap at End of Line
ESC w		Discard at End of Line
ESC P	*	Enter Native Keyboard Mode
ESC Q	*	Exit Native Keyboard Mode
ESC U	*	Set Half Duplex
ESC V	*	Set Full Duplex
ESC c	*	Enable Clock Display
ESC d	*	Disable Clock Display

Additional Functions

ESC }		Disable Keyboard Input
ESC {		Enable Keyboard Input
ESC T	*	Enter Transparent Mode
ESC |	*	Execute Self Test
ESC X	*	Set the Clock
ESC e	*	Send the Time to the Host
ESC ^	*	Reset the Clock to 00:00:00
DC2 (ctrl/R)	*	Emit Click
DC4 (ctrl/T)	*	Back Tab
ESC a	*	Load User Function Keys
ESC m	*	Restore User Function Key Defaults
ESC s	*	Swap Display Memory to Page 2
ESC R <page>	*	Copy To/From Screen Display and Page 2
ESC f <no> <byte>	*	Expand Vertically into <no> <bytes>
ESC g <no> <byte>	*	Expand Horizontally into <no> <bytes>
ESC i <byte>	*	Fill the Display With All <byte>
ESC _ <no>	*	Reverse <no> bytes
ESC ~	*	Reverse the Entire Display
ESC #		Transmit Page
ESC :	*	Transmit Current Line
ESC ;	*	Transmit Current Line Edited
ESC]		Transmit 25th Line
ESC W	*	Transmit Character at Cursor

Summary of Keyboard Operation

This section contains a summary of all special keys recognized by this firmware. Those which are new are marked with '*'.

Key(s)		Code Sent	Description
ESC		ESC	Escape
ctrl-shift/ESC	*	none	Unlock keyboard, clear Transparent Mode, and clear Native Keyboard Mode.
TAB		TAB	Tab to next 8th column
ctrl/TAB	*	DC4	Tab to previous 8th column
SCROLL		none	Scroll terminal one line
shift/SCROLL		none	Scroll terminal one page
shift/SCROLL	*	none	Enter Scroll Mode
ctrl/SCROLL	*	none	Exit Scroll Mode
f1		ESC S	Function Key 1
shift/f1	*	ESC s	
ctrl/f1	*	none	Enter Graphics Mode
ctrl-shift/f1	*	none	Exit Graphics Mode
f2		ESC T	Function Key 2
shift/f2	*	ESC t	
ctrl/f2	*	none	Enter Reverse Video Mode
ctrl-shift/f2	*	none	Exit Reverse Video Mode
f3		ESC U	Function Key 3
shift/f3	*	ESC u	
ctrl/f3	*	none	Set Wrap at End-of-Line
ctrl-shift/f3	*	none	Set Discard at End-of-Line
f4		ESC V	Function Key 4
shift/f4	*	ESC v	
ctrl/f4	*	none	Enter Keypad Shifted Mode
ctrl-shift/f4	*	none	Exit Keypad Shifted Mode
f5		ESC W	Function Key 5
shift/f5	*	ESC w	
ctrl/f5	*	none	Enter Native Keyboard Mode
ctrl-shift/f5	*	none	Enter Transpatent Mode
BLUE		ESC P	Blue Function Key
shift/BLUE	*	ESC p	
ctrl/BLUE	*	none	Enable User Function Keys
ctrl-shift/BLUE	*	none	Disable User Function Keys

RED		ESC Q	Red Function Key
shift/RED	*	ESC q	
ctrl/RED	*	none	Copy page 1 to page 2
ctrl-shift/RED	*	none	Copy page 2 to page 1
WHITE		ESC R	White Function Key
shift/WHITE	*	ESC r	
ctrl/WHITE	*	none	Swap page 1 with page 2
ctrl-shift/WHITE	*	none	Reset the Clock to 00:00:00
ERASE		ESC J	Erase to End of Display
shift/ERASE		ESC E	Erase Entire Display
ctrl/ERASE		none	Erase to End of Display
ctrl-shift/ERASE		none	Erase Entire Display
BACKSPACE		BS	Back Space
LINE FEED		LF	Line Feed
RETURN		CR	Carriage Return
ctrl/RETURN	*	none	Carriage Return
ctrl-shift/RETURN	*	variable	Transmit Current Line Edited
DELETE		DEL	Delete (Rubout)
ctrl-shift/DELETE	*	none	Reset the terminal

Key(s)	Code Sent	Description
Keypad Shifted		
ENTER	CR	Carriage Return
.	.	
0	0	
1	ESC L	Insert Line
2	ESC B	Cursor Down
3	ESC M	Delete Line
4	ESC D	Cursor Backward
5	ESC H	Cursor Home
6	ESC C	Cursor Forward
7	ESC @	Enter Insert Character Mode
	ESC O	Exit Insert Character Mode
8	ESC A	Cursor Up
9	ESC N	Delete Character
Alternate Keypad		
ENTER	ESC ? M	
.	ESC ? n	
0	ESC ? p	
1	ESC ? q	
2	ESC ? r	
3	ESC ? s	
4	ESC ? t	
5	ESC ? u	
6	ESC ? v	
7	ESC ? w	
8	ESC ? x	
9	ESC ? y	

Escape Sequence Descriptions

This section contains detailed descriptions of all of the new or changed escape sequences in the new firmware. Only those sequences marked with '*' above are discussed here. For information on all other sequences, see the H19 Operation Manual.

Set Baud Rate

ESC r <baudrate>

This escape sequence may be used to change the baud rate of the H19 terminal. The <baudrate> must be one of those in the table below.

Baud	Code
110	A
300	B
1200	C
2400	D
4800	E
9600	F
19200	G (see note)
38400	H (see note)

Note:

For reliable operation at these high rates of speed, it is necessary to modify your software to use hardware handshaking. The necessary modifications for Heath/Zenith CP/M 2.2.03 and for HDOS 2.0 may be found in the back of this document. At any rate, there is little gain unless the computer's processor is operating at 4 MHz or faster. It is also necessary to modify your terminal clock rate to 3 or 4 MHz to utilize the higher transfer rates. This modification is not a simple change in all terminals and has therfore been omitted from this document. Users interested in modifying their terminal logic boards for higher clock rates should contact Software Wizardry for the necessary information. Performing the modification WILL INVALIDATE ANY WARRANTY by Heath on the equipment.

Set Mode 2

ESC h <mode>

The purpose of this escape sequence is to permit the user to enable and disable certain operating modes, either from the keyboard or under program control. A brief discussion of each of these modes and their implications follows.

Mode: 1 = Enable Software Handshaking

The H19 terminal formerly was only capable of this type of handshaking, but with the advent of super high speed communications (at rates up to 38,400 Baud), it has become necessary to implement hardware handshaking. Hardware handshaking is always used. If your software does not support this type of handshaking, it is ignored. If your system does, however, support hardware handshaking, it may become desirable to disable software handshaking. This may be especially true when running under CP/M as the software codes for start and stop transmit may be misinterpreted by the program running. The problem is not so critical whenrunning under HDOS, though, because HDOS has built-in support for software handshaking. CP/M does not. If you do determine to disable software handshaking, you must apply the patches at the back of this document to support the hardware handshaking.

2 = Disable Software Handshaking

This escape sequence is used to disable software handshaking by the H19 terminal. In this mode, the H19 will perform ONLY hardware handshaking. See the discussion above for implications and remedies.

3 = Set Normal Scan Mode

This firmware has the ability to enable the interlaced scan mode of the CRT controller chip. This sequence resets the scan mode to the normal (non-interlaced) mode.

4 = Set Interlaced Scan Mode

Interlaced scan mode is provided by the CRT controller and is used to fill each character to enhance readability. With this mode enabled, the dots comprising each character will not be individually discernable.

5 = Enable User Function Keys

This code may be used to enable the user programmed function keys. (see 'ESC a' for information on programming these keys). When this mode is enabled, the functions keys f1 thru f5, BLUE, RED, and WHITE will transmit user specified data rather than the standard H19 escape codes. Note that when the user function keys are enabled, a dot 'indicator' will show on the 25th line next to the time of day if the clock display is enabled.

6 = Disable User Function Keys
This code is used to disable the user function keys and restore the H19 to its normal operating mode.

7 = Select CP/M Function Key Values
This code causes the CP/M function key values to be loaded. It does not automatically enable the programmed function keys. The CP/M values are:

```
f1    = dir␣
f2    = type␣
f3    = list␣
f4    = stat␣
f5    = pip␣
BLUE  = ren␣
RED   = era␣
WHITE = user␣
```

8 = Select HDOS Function Key Values
This code causes the HDOS function key values to be loaded. It does not automatically enable the programmed function keys. The HDOS values are:

```
f1    = mount␣
f2    = dismount
f3    = reset␣
f4    = copy␣
f5    = type␣
BLUE  = cat␣
RED   = sy1:
WHITE = sy2:
```

9 = Select MBasic Function Key Values
This code causes the MBasic function key values to be loaded. It does not automatically enable the programmed function keys. The MBasic values are:

```
f1    = goto␣
f2    = gosub␣
f3    = return
f4    = for␣
f5    = next␣
BLUE  = load "
RED   = save "
WHITE = list␣
```

Note that for the previous three sequences, the symbol ␣ denotes a space.

Send Configuration Report

ESC ?

This command has been implemented to provide information regarding the operating modes of the terminal to the host. When this sequence is received by the terminal, a 16 character string terminated by a carriage return is transmitted to the host. This string contains encoded information including all modes currently set or reset in the terminal, the status of DIP switches S401 and S402, and a version number indicating the version of the firmware in use. (This version is 2.4). The data is represented in hexadecimal format and is described below.

Mode Descriptions

Mode A

```
00000001 = Hold Screen Mode
00000010 = Graphics Mode
00000100 = Native Keyboard Mode
00001000 = Auto Line Feed on Carriage Return
00010000 = Auto Carriage Return on Line Feed
00100000 = <not available to host>
01000000 = Insert Character Mode
10000000 = Reverse Video Mode
```

Mode B

```
0000000x = Cursor Type
             0 = Underscore
             1 = Block
00000010 = No Key Click
00000100 = Wrap at End of Line
00001000 = Shifted Keypad Mode
000x0000 = User Key Defaults
             0 = HDOS Keys
             1 = CP/M Keys
0xx00000 = Cursor Display Characteristics
             00 = Not Blinking
             01 = Not Displayed
             10 = Blinking Fast
             11 = Blinking Slow
10000000 = Interlaced Scan Mode
```

Mode Descriptions (continued)

Mode I

```
00000001 = <not available to host>
00000010 = Transparent Mode
00000100 = Keyboard Disabled
00001000 = User Function Keys Enabled
00010000 = <not available to host>
00100000 = Page 2 Display is Available
01000000 = Clock Display is Disabled
10000000 = 25th Line is Enabled
```

Mode S

```
00000xxx = Baud Rate
             000 = 110 baud
             001 = 300 baud
             010 = 1200 baud
             011 = 2400 baud
             100 = 4800 baud
             101 = 9600 baud
             110 = 19200 baud
             111 = 38400 baud
00001000 = Parity Enabled
00010000 = Even Parity
00100000 = Word Length  (1=7 bits, 0=8 bits)
01000000 = Full Duplex Mode
10000000 = Software Handshaking is Disabled
```

Mode C

```
00000001 = Alternate Keypad Mode
00000010 = Unused
00000100 = Unused
00001000 = Unused
00010000 = Unused
00100000 = Unused
01000000 = Unused
10000000 = Unused
```

S401 - Bit assignments are the same as Mode S.

S402 - Bit assignments are the same as Mode B.

Version - Encoded as BCD.

A typical configuration string might be '00F9A0C700C7F124'. The breakdown is as follows: 00 = Mode A; F9 = Mode B; A0 = Mode I; C7 = Mode S; 00 = Mode C; C7 = S401; F1 = S402, 24 = Version. When this is transmitted to the host, it will be terminated by a carriage return.

Set Cursor Type

ESC S <cursortype>

This sequence may be used to set the cursor display characteristics. The sequence is ESC S <cursor>. The <cursor> must be one of those given in the table below.

Cursor	Characteristics
1	Underscore, not blinking
2	Underscore, not displayed
3	Underscore, blinking fast
4	Underscore, blinking slow
5	Block, not blinking
6	Block, not displayed
7	Block, blinking fast
8	Block, blinking slow

Native Keyboard Mode

ESC P - Enable Native Keyboard
ESC Q - Disable Native Keyboard

This enables the native keyboard mode of the H19 firmware. When running with this mode, each key produces a unique one-byte code, rather than escape sequences. All of the normal ASCII characters are returned by the alphabetic and numeric keys. The differences occur in those keys which transmit escape sequences to the host when depressed. The following table lists the keys and their native mode codes. Keys not listed are not affected by native mode. All values are given in hexadecimal.

Key	Unshift	Shift	Ctrl	Ctrl-Shift
f1	D3	F3	D3	F3
f2	D4	F4	D4	F4
f3	D5	F5	D5	F5
f4	D6	F6	D6	F6
f5	D7	F7	D7	F7
ERASE	CA	C5	CA	C5
BLUE	D0	F0	D0	F0
RED	D1	F1	D1	F1
WHITE	D2	F2	D2	F2
0 (keypad)	80	90	80	90
1 (keypad)	81	91	81	91
2 (keypad)	82	92	82	92
3 (keypad)	83	93	83	93
4 (keypad)	84	94	84	94
5 (keypad)	85	95	85	95
6 (keypad)	86	96	86	96
7 (keypad)	87	97	87	97
8 (keypad)	88	98	88	98
9 (keypad)	89	99	89	99
. (keypad)	8A	9A	8A	9A
ENTER	8B	9B	8B	9B

Note that the H19 must not be configured for 7 bit words if this mode is to be used. This is due to the fact that all of these character codes require 8 bits to transmit.

Set Half/Full Duplex

ESC U - Set Half Duplex
ESC V - Set Full Duplex

These sequences allow the user to set the H19 terminal into full or half duplex from, the keyboard, or under software. This would typically be desired when using the terminal in a direct connect mode with a modem. All Heath/Zenith software supports full duplex operation.

Enable/Disable Clock Display

ESC c - Enable Clock Display
ESC d - Disable Clock Display

When the H19 is reset, a clock display appears in the lower right-hand corner of the CRT on line 25. These sequences may be used to disable/enable the clock display. It is not necessary to disable the clock display when using the 25th line. When the 25th line is enabled by software or from the keyboard, the clock display will automatically be removed. When the 25th line is later disabled, the clock display will be restored to the 25th line if display of the clock has not been disabled.

Send Time to Host

ESC e

This code causes the H19 to transmit the current time to the host. The format of the transmitted string will be HHMMSS, and the string will be terminated by a carriage return. For example, the time 21:14:44 would be received by the computer as '211444'.

Set the Clock

ESC X HHMMSS <cr>

Using this escape sequence, the user may set the terminal clock to any time in 24-hour (military) format. The time may be set either from the keyboard or from the computer. Note the <cr> at the end of this sequence. This character determines when the clock will be started after it is set. The clock will not begin to run until the <cr> character has been received. <cr> may be any character, but I use carriage return (which is why I call it <cr>).

For example, to set the time to 01:26 P.M., you would enter (if you were doing it from the keyboard):

ESC X 1 3 2 6 0 0

followed by a <cr> when you were ready for the clock to begin running. Note that since the clock is generated internally by the terminal, it will probably not be your quality time piece. I find that my terminal loses about a minute per day. The time is not retained when the terminal is powered off.

Reset the Clock

ESC ^

This escape code may be used to reset the clock to 00:00:00. This may be desired if the terminal clock were being utilized as an elapsed timer, rather than a 'time-of-day' clock.

Enter Transparent Mode

ESC T

This mode disables ALL (without exception) escape code processing by the H19 terminal. All characters are displayed as they are received. All normally displayable ASCII characters are displayed in their normal fashion. Normally displayable ASCII characters which are received with the parity bit (bit 7) set high are displayed in inverse video. All non-displayable characters are displayed as two hexadecimal digits within <angle brackets>, and in reverse video. This mode may be exited only by resetting the terminal from the keyboard, or by depressing ctrl-shift/ESC.

Execute Self Test

ESC |

This mode performs several internal diagnostic tests on the H19 terminal. They are: Video RAM memory test; Page 2 RAM memory test (if page 2 is available); Scratchpad RAM memory test; and a Firmware ROM CRC verification. Upon completion of these tests, the terminal should display:

-> RAM Ok
-> PG2 Ok
-> MEM Ok
-> ROM Ok

-> Test End

If the page 2 RAM is not present, the message 'PG2 n/a' will appear in place of 'PG2 Ok'. Any failure will be indicated by the word 'Error'. After the self test has been run, the terminal must be reset using the Shift/RESET keys to resume operation. If any section fails, the following parts may be suspect. The [bracketed] IC numbers are for older H19s.

RAM Err	-	U408 U409 U410 U411 [U467 U468 U469 U470]
PG2 Err	-	U436 [U423]
MEM Err	-	U438 U439 [U424 U425]
ROM Err	-	U437 [U422]

Swap Screen Display with Page 2

ESC s

This escape sequence causes the H19 to swap the contents of the video RAM with the PAGE 2 memory if it is installed on the board. If no PAGE 2 RAM is present, this code is ignored. In addition, the current cursor location is saved with the display, so in effect you have seperate cursors for each page. Only the cursor's position is saved, not its display attributes. Neither are any of the terminal attributes such as inverse video or graphics modes saved. This command requires a relatively long time to execute due to the nature of the processing required to perform the swap. Whenever possible, the Copy Page (described next) command should be used. Approximate time required for the terminal to complete this command is 85 ms (56 ms at 3 MHz, 42 ms at 4 MHz).

Copy To/From Screen Display and Page 2

ESC R <page>

This sequence may be used to copy the contents of either video RAM or Page 2 to the other. The <page> indicates the target of the copy function. Values are '1' for copy Page 2 to VRAM, and '2' for copy VRAM to Page 2. This command executes much faster then the 'swap' described above and is preferred for loading the contents of one page from another page. Approximate time required for the terminal to complete this command is 21 ms (14 ms at 3 MHz, 11 ms at 4 MHz).

Load User Function Keys

ESC a <key> <d1><d2><d3>...<d8>

This escape code allows the user to load any desired data into the specified function key. <key> is a single digit identifying the key to be loaded. Values are '1' for f1, '2' for f2, '3' for f3, ..., and '8' for WHITE. The data may be from zero to eight characters in length. If less than eight characters are to be loaded, however, the load string must be terminated by a DELETE character. Once the keys are loaded, they may be activated via the 'ESC h 5' sequence (described above). If no data is loaded into the keys, they will be programmed by default to the following, depending upon the setting of S402.

Key	CP/M Data	HDOS Data
f1	dir␣	mount␣
f2	type␣	dismount
f3	list␣	reset␣
f4	stat␣	copy␣
f5	pip␣	type␣
BLUE	ren␣	cat␣
RED	era␣	sy1:
WHITE	user␣	sy2:

Use of the function keys in this mode will not affect operation of the keys in the shifted mode. The function keys will not however, transmit their normal unshifted escape sequences with this mode enabled. As above, the symbol ␣ denotes a space.

Restore User Function Keys

ESC m

Upon receipt of this escape sequence, the H19 will restore the programmed function key values to their defaults (listed in the table above). This sequence may be used to restore the keys upon exit from an application program.

Fill Screen with a Byte

ESC i <byte>

This sequence may be issued to rapidly fill the display with a single byte. Graphic and inverse video characters may also be used in this sequence. Graphics should be presented as control characters and inverse video with the high order bit (bit 7) set.

Expand Bytes Horizontally

ESC g <count> <byte>

The purpose of this sequence is to enable the transfer of strings of repeated bytes in a short sequence. This code is expanded into the number of <bytes> designated by <count>. The expansion takes place from the current location of the cursor and proceeds exactly as though the characters to be displayed had been received from the host. The value <count> should be presented as a binary number ranging from 0 to 255. <byte> is as above, where graphics are entered as control characters and inverse video has the high order bit set. For example, to fill a line with '-' characters: 'ESC g P -'. This is assuming that the cursor is already located in column one. This sequence would typically be used for drawing horizontal lines.

Expand Bytes Vertically

ESC f <count> <byte>

This sequence provides a mechanism similar to 'ESC g' above, but in this case, expansion is done vertically into the number of <bytes> indicated by the <count>. The expansion takes place from the current position of the cursor. Expansion will stop when line 24 is reached, regardless of the count. This sequence would typically be used for drawing vertical lines.

Reverse Field

ESC _ <count>

This sequence may be used to reverse <count> characters, beginning at the current location of the cursor. If the number of characters to be reversed is greater than the distance from the cursor to the end of the line, the operation will wrap, or stall in column 80 depending upon the mode currently set (wrap or discard at EOL).

Reverse Entire Display

ESC `

This sequence causes the contents of the entire display, from line 1 column 1 to line 24 column 80 to be reversed.

Transmit Character at Cursor

ESC W

This causes the H19 to transmit the character which is located at the current position of the cursor. Graphics characters are transmitted as control characters, and inverse video characters are sent with the parity bit (bit 7) set. Note that if the terminal is set for 7 bit words, it will not be possible to transmit bit 7, making it impossible to detect inverse video.

Transmit Current Line

ESC :

This causes the H19 to transmit the contents of the current line to the computer, including all necessary escape codes for entering and exiting graphics and reverse video modes. All trailing spaces in the line will also be sent. As with the transmit page function, the data received by the host will be terminated by a carriage return.

Transmit Current Line Edited

ESC ;

This code causes the H19 to transmit the contents of the current line exactly as it appears on the terminal. Trailing spaces are removed and the line is terminated with a carriage return. Graphic characters are transmitted as control characters and inverse video characters are transmitted with the parity bit (bit 7) set high. As noted above, if the terminal is set to communicate using 7 bit words, it will not be possible to transmit bit 7, making it impossible to detect inverse video.

Special Keyboard Sequences

This section will describe several key combinations which may be used to manipulate the H19 and the data displayed.

Ctrl-Shift / ESC

This key combination may be used to clear the transparent mode of the terminal if it is set. It will also unlock the keyboard if it is locked and clear native keyboard mode if it is set. No codes are transmitted to the computer.

Ctrl / TAB

This key combination causes the terminal to TAB backwards to the previous tab stop. If the cursor is already in column 1, it will not move. The code DC4 (hex 14) is transmitted to the computer when this key is depressed.

Shift / SCROLL

This key will cause the H19 to enter the scroll mode if it is not currently in it. If the terminal is already in the scroll mode, this key behaves as normal and causes the display to advance one page (24 lines). No codes are transmitted when this key is depressed.

Ctrl / SCROLL

This key causes the H19 to exit the scroll mode if it is set. If the terminal is not in scroll mode, the key is ignored. No codes are transmitted when this key is depressed.

Shift / <function key>

The keys f1 thru f5, BLUE, RED, and WHITE will produce unique escape sequences if depressed while shifted.

Key	Code Transmitted
f1	ESC s
f2	ESC t
f3	ESC u
f4	ESC v
f5	ESC w
BLUE	ESC p
RED	ESC q
WHITE	ESC r

Ctrl-Shift / DELETE

This sequence causes a 'soft' reset of the H19 terminal. It may be used at any time, and causes no codes to be transmitted to the host. The clock will not be reset as with a 'hard' reset, and the user function key data (if any loaded) will not be lost. This was implemented to provide a convenient (but not too easy to do accidentally) method of resetting the terminal without resetting the computer as happens with the H88/H89/Z90 systems when Shift/RESET is done. No audible key click will be heard.

Ctrl / RETURN

This key may be used to return the cursor to column one without actually transmitting a carriage return character to the host. See the next key for application.

Ctrl-Shift / RETURN

This key causes the H19 to transmit the contents of the current line exactly as it appears on the terminal. Trailing spaces are removed and the line is terminated with a carriage return. Graphic characters are transmitted as control characters and inverse video characters are transmitted with the parity bit (bit 7) set high. This may be used to edit a command line on the screen (using the arrow keys off-line to move around) and the resultant command may then be transmitted to the host. This can be especially useful if a long command line is entered and a mistake is noted at the beginning. Using this key, it would be a simple matter to correct the mistake and then send the corrected line to the computer just as if you had typed it.

Software Patches

This section contains patches designed to facilitate using the new firmware with your computer. These patches are intended for use only if you plan on using your terminal at a baud rate greater than 9600, or if you plan on using the H19 with software handshaking disabled (recommended for CP/M).

HDOS 2.0

The following patch will cause HDOS to use the hardware handshaking provided by the H19 in addition to the software handshaking already recognized. (This patch may not function properly with some versions of the Heath H/Z37 device driver due to a conflict in use of predefined patch areas within HDOS.SYS).

Program: HDOS.SYS

Patch Id: iegjih
Prerequisite Code: ifbeiadpgeffcf
Patch Check Code: clmnogoo

Address	Old Data	New Data
000.002	004	012
034.002	000	156
	000	071
	000	076
	000	100
	000	104
	000	100
	000	000
	000	000
013.201	361	303
	323	242
	350	064
013.242	014	333
	012	356
	101	346
	116	020
	117	312
	124	242
	110	064
	105	361
	122	323
	040	350
	106	303
	111	204
	116	064

Address	Old Data	New Data
020.155	361	303
	323	071
	350	100
027.071	275	333
	246	356
	337	346
	270	020
	276	312
	274	071
	337	100
	266	361
	261	323
	337	350
	255	303
	272	160
	262	071

If you plan on running your terminal at 38400 baud under HDOS, you must make the following patch to track 0, sector 0 so that HDOS will have the correct baud rate.

Byte	Value
5	003
6	000

The following patch will enable INIT.ABS to detect 38400 baud on a newly initialized disk by typing spaces. This patch should be applied if you plan on using your system at 38k baud. Once installed, all disks initialized with the patched INIT will be capable of doing auto baud rate selection by typing spaces.

Program: INIT.ABS

Patch Id: iegjih
Prerequisite Code: ifbeiadpgeffcf
Patch Check Code: mkidcbef

Address	Old Data	New Data
051.036	242	322
051.260	242	322
051.322	000	003
	006	000

Heath/Zenith CP/M 2.2.03

The necessary modifications to CP/M must be made in the BIOS source code. Insert the following lines in the file BIOS.ASM with an editor and then run the MAKEBIOS procedure to implement the changes in your system.

```
CRTOS1: CALL    UOS
        JZ      CRTOSB
        MVI     A,6         (*)
        CALL    PINX        (*)
        ANI     10H         (*)
        JZ      CRTOSB      (*)
```

The lines marked (*) are the ones which are to be added. The 1st two lines are shown here to provide a reference as to where the modification is to be made.

In addition, the size of the type-ahead buffer under CP/M should be expanded to 80 characters. Generally, the type-ahead buffer in the standard BIOS will fill up and overrun when receiving long lines of input data in the manner presented by 'transmit edited'. The following statement demonstrates the necessary modification:

```
CRTBUF: DS      80
```

Note that the previous value here was 40. This statement may be found near the end of the BIOS.ASM.

Alphabetical Escape Code List

Code		Description
#		Transmit Page
:	*	Transmit Current Line
;	*	Transmit Line Edited
<	*	unused [formerly ANSI mode]
=		Enter Alternate Keypad Mode
>		Exit Alternate Keypad Mode
?	*	Send Configuration Report
@		Set Insert Character Mode
A		Cursor Up
B		Cursor Down
C		Cursor Right
D		Cursor Left
E		Clear Screen
F		Enter Graphics Mode
G		Exit Graphics Mode
H		Cursor Home
I		Reverse Line Feed
J		Erase to End of Page
K		Erase to End of Line
L		Insert Line
M		Delete Line
N		Delete Character
O		Exit Insert Character Mode
P	*	Enter Native Keyboard Mode
Q	*	Exit Native Keyboard Mode
R	*	Copy Display to/from VRAM
S	*	Set Cursor Type
T	*	Enter Transparent Mode
U	*	Set Half Duplex
V	*	Set Full Duplex
W	*	Send Character at Cursor
X	*	Set Clock
Y		Direct Cursor Addressing
Z		Identify as VT-52
[		Enter Hold Screen Mode
\		Exit Hold Screen Mode
]		Transmit 25th Line
^	*	Reset Clock to 00:00:00
_	*	Reverse Field on Screen
`	*	Reverse Entire Screen
a	*	Load Programmed Function Keys
b		Erase Beginning of Display
c	*	Enable Clock Display on Line 25
d	*	Disable Clock Display on Line 25
e	*	Send Time to Host
f	*	Expand Bytes Vertically
g	*	Expand Bytes Horizontally
h	*	Set/Clear Modes 2
i	*	Fill Screen with Byte
j		Save Cursor Position
k		Unsave Cursor Position

Alphabetical Escape Code List (continued)

Code		Description
l		Erase Entire Line
m	*	Reset Programmed Function Keys
n		Cursor Position Report
o		Erase Beginning of Line
p		Enter Reverse Video Mode
q		Exit Reverse Video Mode
r	*	Set Baud Rate
s	*	Swap Display Memory with VRAM
t		Enter Keypad Shifted Mode
u		Exit Keypad Shifted Mode
v		Set Wrap Mode
w		Set Discard Mode
x		Set Heath Mode
y		Reset Heath Mode
z		Initialize to Power-up Configuration
{		Enable Keyboard Input
|	*	Terminal Self Test
}		Disable Keyboard Input

Heath Set Modes

Code	Description
1	Enable 25th Line
2	Disable Key Click
3	Enter Hold Screen Mode
4	Select Block Cursor
5	Cursor Off
6	Enter Keypad Shifted Mode
7	Enter Alternate Keypad Mode
8	Auto Line Feed on Carriage Return
9	Auto Carriage Return on Line Feed

Heath Reset Modes

Code	Description
1	Disable 25th Line
2	Enable Key Click
3	Exit Hold Screen Mode
4	Select Underscore Cursor
5	Cursor On
6	Exit Keypad Shifted Mode
7	Exit Alternate Keypad Mode
8	No Auto Line Feed on Carriage Return
9	No Auto Carriage Return on Line Feed

Set/Reset Modes 2

Code		Description
1	*	Enable Software Handshake
2	*	Disable Software Handshake
3	*	Set Normal Scan Mode
4	*	Set Interlaced Scan Mode
5	*	Enable Programmed Function Keys
6	*	Disable Programmed Function Keys
7	*	Select CP/M Key Values
8	*	Select HDOS Key Values
9	*	Select MBasic Key Values

Set Baud Rate

Code		Baud Rate
A		110
B		300
C		1200
D		2400
E		4800
F		9600
G		19200
H	*	38400

Set Cursor Attributes

Code		Attributes
1	*	Underscore, no blink
2	*	Underscore, non-display
3	*	Underscore, fast blink
4	*	Underscore, slow blink
5	*	Block, no blink
6	*	Block, non-display
7	*	Block, fast blink
8	*	Block, slow blink

'*' denotes new or changed escape sequences.

Notes, Tips, Etc.

The purpose of this section of the manual is to cover all of the little details that don't fit in very well any place else. Primarily covered are details pertaining to certain functions which may be still unclear, examples of use of certain features, and idiosyncrasies. This section has no particular organization since I'm just writing these things down as I get to them on my list. Soooo ...

#########

The amount of time required to perform a terminal reset under software control varies depending of the presence or absence of the PAGE 2 RAM. Times at various clock rates are:

	Without Page 2	With Page 2
2 MHz	17.3 ms	38.3 ms
3 MHz	11.6 ms	25.6 ms
4 MHz	8.7 ms	19.2 ms

#########

It is possible to set the terminal clock to times from 24:00:00 up to 29:59:59. These times are not valid and the clock will not cycle properly if the clock is set in this manner. There was a trade-off made in the code which actually sets the clock and I felt that most folks would be intelligent enough to do it right.

#########

The sequences 'ESC g', 'ESC f', and 'ESC _' will do nothing if a count of zero is given. For example, the MBASIC statement:

```
10 PRINT CHR$(27);"g";CHR$(0);"Z"
```

will do absolutely nothing.

#########

Most operating systems strip the parity bit from data as it is read from the terminal, so applications which need to use features such as native keyboard mode, and transmit character at cursor will typically need to do their own I/O. The following program segment in MBASIC illustrates how this may be accomplished on the Heath/Zenith H8 and H89 computers:

```
10 A=INP(&O351):'              save interrupt enable register
20 OUT &O351,(A AND &O376):'   clear receiver interrupts
30 WAIT &O355,1:'              wait for a character to appear
40 C=INP(&O350):'              read the character from the port
50 OUT &O351,A:'               restore the interrupt register
```

This will work under HDOS, and under non-interrupt driven CP/M. It will not work under CP/M with console interrupts because the

interrupts are re-enabled by the BIOS before the program can do it's thing.

#########

In a similar manner to that described above, some operating systems strip parity from outbound data. The user will need to do the necessary I/O in order to send 8 data bits to the terminal. The following program segment illustrates how this may be done in MBASIC on the H8 and H89 computers:

```
100 PRINT CHR$(27);"g";:'      set up to draw horizontal
110 C=160:'                    2 lines of data
120 WAIT &O355,&O140:'         wait for transmitter ready
130 OUT &O350,C:'              send C to the terminal
140 PRINT "-";:'               draw 2 lines of "-"
```

#########

An application program may determine if this firmware is installed in the terminal by issuing the 'ESC ?' sequence and waiting an appropriate period of time for a response. If no data is received from the terminal in the allotted time, it may be assumed that this firmware is not present.

#########

Due to the manner in which the cursor attributes may be set and reset, it is not possible for the host to determine whether the firmware keeps track of the cursor's display attributes (blink, block, underscore) but not whether it is actually enabled. All other terminal status kept by the terminal may be read via the 'ESC ?' sequence. This is the only exception.

#########

The following program segment illustrates the use of the 'ESC g' and 'ESC f' sequences as they may be employed:

```
100 E$=CHR$(27)
110 C$=E$+"E":'            erase screen
120 Y$=E$+"Y":'            address cursor
130 H$=E$+"g":'            expand horizontal
140 V$=E$+"f":'            expand vertical
150 Z$="q"
160 DOT$=CHR$(127)
170 WIDTH 255
180 '
190 DEF FNX$(Y,X)=Y$+CHR$(31+X)+CHR$(31+Y)
200 DEF FNH$(C)=H$+CHR$(C)+DOT$
210 DEF FNV$(C)=V$+CHR$(C)+DOT$
220 '
230 PRINT C$
240 Y=1
250 PRINT E$;Z$;
```

```
260 FOR X=21 TO 1 STEP -4
270 PRINT FNX$(X,Y);FNH$(80-(X*2-1));
280 PRINT FNV$(24-(Y*2-1));
290 PRINT FNX$(X,Y);FNV$(24-(Y*2-1));
300 PRINT FNH$(80-(X*2-2));
310 Y=Y+2
320 NEXT
330 Z$=CHR$(ASC(Z$) XOR 1)
340 GOTO 240
```

#########

The following program segment demonstrates how the function keys may be loaded under program control.

```
100 FOR X=0 TO 7
110 READ K$
120 IF LEN(K$)>8 THEN K$=LEFT$(K$,8):PRINT "Truncated."
130 PRINT CHR$(27);"a";CHR$(ASC("1")+X);
140 FOR Y=1 TO LEN(K$)
150 PRINT MID$(K$,Y,1);
160 NEXT
170 IF LEN(K$)<8 THEN PRINT CHR$(127);
180 NEXT
190 STOP
200 DATA "MOUNT","DISMOUNT","LOAD","RESET","SAVE"
210 DATA "SY0:","SY1:","SY2:"
```

#########

When the user programmed function keys are being used, be sure to turn them back off before invoking an application program which expects them to function in their normal manner.

#########

The 'ESC x 4' and 'ESC y 4' escape sequences will have no effect upon the blink rate of the cursor. Previously, 'ESC x 4' set a slow blink block cursor, and 'ESC y 4' set a fast blink underscore cursor.

Disclaimer

This ROM and documentation set is sold without warranty or guarantee to the purchaser except as expressed below. Operation of this program by the licensee is strictly at the risk of the licensee, and neither the author nor Software Wizardry, Inc. accepts any responsibility for such use or any damages resulting from such use. The author and Software Wizardry, Inc. warrant only to provide a reasonable amount of assistance to the licensee in eliminating problems as they may occur.

Product Discrepancy Report

Software Wizardry, Inc. spends a great deal of time testing and improving its products - but from time to time (being realists) we know errors may accidentally be included in an otherwise exceptional product no matter how much care is used.

Should you identify a problem area, please let us know by sending in this sheet describing the problem in as clear and concise a manner as possible. If the problem is of such a nature as to make program operation awkward to you, please return your distribution disk along with this form and we will try to return a corrected version to you ASAP (certain program fixes and documentation changes may take some time to repair). Be sure to include your phone number in case additional information is required. THANK YOU!!

YOUR NAME: _____

ADDRESS: _____

_____

PHONE: _____

PRODUCT: _____

VERSION: _____

PROBLEM AREA (Attach printouts & additional sheets if necessary):

_____

_____

_____

_____

_____

_____

_____

_____

MAIL TO: Software Wizardry, Inc., 122 Yankee Drive,
St. Charles, Mo. 63301