SUPER-BIOS REFERENCE -- E1.5

Super-BIOS Interface Description

Setup: ax= Super-BIOS function number
es:bx= long pointer to parameter block

Invoked: Software interrupt 223

Return: As specified in function description.
Stack and Code environment is preserved.
Super-BIOS interface requires three words of caller's stack.
DS register is preserved.
AX,ES, and BX are reserved for return values.

Function Number

0 Super-BIOS Release Number
Returns a unique number for identifying the Super-BIOS release level.

inputs: none
outputs: ax= release number

1 Get Zone (primarily for diagnostics use)
Converts a track number to a zone number.

inputs: parm1= track number (word)
outputs: al= zone number

2 Get Sectors Per Track (primarily for diagnostics use)
Returns the number of sectors for a specified track.

inputs: parm1= track number (word)
outputs: al= number of sectors

3 Allocate Disk Job (primarily for diagnostics use)
Returns a pointer to disk job structure.

inputs: none
outputs: al= error code (0= successful)
es:bx= long pointer to job structure

4 Start Disk Job (primarily for diagnostics use)
Submit a job to the low-level disk driver.

inputs: parm1= long pointer to job structure
parm2= priority (word)
outputs: none

5 Get Drive Information (primarily for diagnostics use)
Return a pointer to a drive's information structure.

```
inputs:  parm1= drive number (word)
outputs: es:bx= long pointer to drive info structure
```

6 Read Sector
Read physical sector(s) into memory with full retries.

```
inputs:  parm1= drive number (word)
         parm2= track number (word)
         parm3= sector number (word)
         parm4= sector count (word)
         parm5= long pointer to memory buffer
outputs: al= floppy disk error code
```

7 Write Sector
Write physical sector(s) from memory with sector verify and full retries.

```
inputs:  parm1= drive number (word)
         parm2= track number (word)
         parm3= sector number (word)
         parm4= sector count (word)
         parm5= long pointer to memory buffer
outputs: al= floppy disk error code
```

8 Convert Logical Sector to Physical Disk Address
Convert an absolute sector number on disk to a track and sector number.

```
inputs:  parm1= absolute sector number (word)
         parm2= long pointer to return values
                value1= track number (word)
                value2= sector number (word)
outputs: return values set
```

9 Flush Buffers
Flush all updated buffers to disk.

```
inputs:  parm1= drive number (word)
outputs: al= floppy disk error code
```

10 Display Disk Error
Display BIOS error message for last disk operation.

```
inputs:  none
outputs: (message displayed on screen)
```

11 Get Dot Generator Address
Returns starting address and byte count of character set dot memory.

```
inputs:  none
outputs: es= starting paragraph of dot memory
         bx= byte count
```

12 Set Dot Generator Length

Sets byte count of character set dot memory.

inputs: parm1= new character set byte count (word)
outputs: al= error code (0= successful)

13 Get Screen Address

Return starting memory address of screen.

inputs: none
outputs: es= starting paragraph of screen memory

ERRATA SHEET

CHANGE IN SUPER-BIOS FUNCTION 8 SPECIFICATION

Func 8: Convert Logical Sector to Physical Disk Address

Convert an absolute sector number into its physical track and sector number.

inputs: parm1= drive number (word)
parm2= logical sector number (word)
parm3= interleave factor (word)

outputs: ah= track number; al= sector number
al= FF if error

SUPER-BIOS DISCLAIMER (A WORD OF CAUTION)

After using the Super-Bios for awhile, we feel that the Super-Bios interface could be made more unified. This would involve changing the mechanism for returning results and error codes from the Super-Bios functions in a way which would allow all of the functions to return identically. This makes the job of accessing the Super-Bios much easier for applications written in high level languages, such as Basic or PLM.

The functionallity of the Super-Bios will not change: only the way in which the functions are called and return will change. Thus the Super-Bios in this release of the operating system should only be used to test and evaluate the Super-Bios functions. Applications which incorporate any Super-Bios calls will need to be (slightly) modified when the interface changes.

Super-BIOS Interface Description

Setup: ax= Super-BIOS function number
es:bx= long pointer to parameter block

Invoked: Software interrupt 223

Return: As specified in function description.
Stack and Code environment is preserved.
Super-BIOS interface requires three words of caller's stack.
DS register is preserved.
AX,ES, and BX are reserved for return values.

Function Number

0 Super-BIOS Release Number
Returns a unique number for identifying the Super-BIOS release level.

inputs: none
outputs: ax= release number

1 Get Zone (primarily for diagnostics use)
Converts a track number to a zone number.

inputs: parm1= track number (word)
outputs: al= zone number

2 Get Sectors Per Track (primarily for diagnostics use)
Returns the number of sectors for a specified track.

inputs: parm1= track number (word)
outputs: al= number of sectors

3 Allocate Disk Job (primarily for diagnostics use)
Returns a pointer to disk job structure.

inputs: none
outputs: al= error code (0= successful)
es:bx= long pointer to job structure

4 Start Disk Job (primarily for diagnostics use)
Submit a job to the low-level disk driver.

inputs: parm1= long pointer to job structure
parm2= priority (word)
outputs: none

Return a pointer to a drive's information structure.

inputs: parm1= drive number (word)
outputs: es:bx= long pointer to drive info structure

6 Read Sector
Read physical sector(s) into memory with full retries.

inputs: parm1= drive number (word)
parm2= track number (word)
parm3= sector number (word)
parm4= sector count (word)
parm5= long pointer to memory buffer
outputs: al= floppy disk error code

7 Write Sector
Write physical sector(s) from memory with sector verify and full retries.

inputs: parm1= drive number (word)
parm2= track number (word)
parm3= sector number (word)
parm4= sector count (word)
parm5= long pointer to memory buffer
outputs: al= floppy disk error code

8 Convert Logical Sector to Physical Disk Address
Convert an absolute sector number on disk to a track and sector number.

inputs: parm1= absolute sector number (word)
parm2= long pointer to return values
value1= track number (word)
value2= sector number (word)
outputs: return values set

9 Flush Buffers
Flush all updated buffers to disk.

inputs: parm1= drive number (word)
outputs: al= floppy disk error code

10 Display Disk Error
Display BIOS error message for last disk operation.

inputs: none
outputs: (message displayed on screen)

11 Get Dot Generator Address
Returns starting address and byte count of character set dot memory.

inputs: none
outputs: es= starting paragraph of dot memory
bx= byte count

inputs: parm1= new character set byte count (word)
outputs: al= error code (0= successful)

13 Get Screen Address
Return starting memory address of screen.

inputs: none
outputs: es= starting paragraph of screen memory

Super-BIOS Interface Description

Calls to the Super-BIOS are made by loading the AX register with a function number and then doing a software interrupt (level 223). All parameters/results are passed to/from the Super-BIOS functions via a parameter block supplied by the caller. The parameter block is pointed to by a long pointer set by the caller in the ES:BX registers. The Super-BIOS function returns with an exception condition in the AX register: a zero value indicates that the function was succesfully completed.

The contents of the CX,DX,SI,DI and BP and flag registers return undefined. The Super-BIOS interface requires 3 words of the caller's stack.

Setup:	ax=	function number
		es:bx= pointer to the parameter block
Call:	INT	223
Return:	ax=	exception code
		return values set in parameter block

Super BIOS Functions

Function 0: Version Number

Returns a unique number used for identifing the release of the operating system.

parm1:	version number	(word/output)

Function 1: Get Zone (for diagnostic use)

Converts a track number into its zone on the disk.

parm1:	track number	(word/input)
parm2:	zone	(word/output)

Function 2: Get Sectors On Track (for diagnostic use)

Returns the number of sectors on the specified track.

parm1:	track number	(word/input)
parm2:	number of sectors	(word/output)

Function 3: Allocate Disk Job (for diagnostic use)

Returns a pointer to a disk job structure.

parm1:	job structure ptr	(long pointer/output)

Function 4: Start Disk Job (for diagnostic use)

Submit a job to the low-level floppy disk driver.

parm1:	job structure ptr	(long pointer/input)
parm2:	priority	(word/input)

Function 5: Get Drive Information (for diagnostic use)

Return a pointer to the specified drive's information structure.

parm1:	drive number	(word/input)
parm2:	DI structure ptr	(long pointer/output)

Function 6: Read Sector

Read physical sector(s) into memory buffer. Full retries are performed.

parm1:	drive number	(word/input)
parm2:	track number	(word/input)
parm3:	sector number	(word/input)
parm4:	sector count	(word/input)
parm5:	buffer address	(long pointer/input)

Function 7: Write Sector

Write physical sector(s) from memory buffer. Full retries are performed.

parm1:	drive number	(word/input)
parm2:	track number	(word/input)
parm3:	sector number	(word/input)
parm4:	sector count	(word/input)
parm5:	buffer address	(long pointer/input)

Function 8: Logical to Physical Disk Address

Convert an logical sector number into a physical track and sector number.

parm1:	drive number	(word/input)
parm2:	logical sector	(word/input)
parm3:	interleave factor	(word/input)
parm4:	track number	(word/output)
parm5:	sector number	(word/output)

Function 9: Flush Disk Buffer

Flush any updated ("dirty") disk buffers.

parm1:	drive number	(word/input)

Function 10: Display Disk Error Message

Display the disk error message for the last disk operation.

Function 11: Get Character Generator

Returns the length and the starting address of the character generator dots in memory.

parm1:	byte count	(word/output)
parm2:	base addr of dots	(word/output)

Function 12: Set Character Generator Size

Set the size of the character generator.

parm1:	byte count	(word/input)

Function 13: Get Screen Address

Return the location of screen memory.

parm1:	base of screen	(word/output)

Function 14: Get Device Vector

Get the current vector for the specified device driver.

parm1:	device number	(word/input)
parm2:	device vector	(long pointer/output)

Function 15: Set Device Vector

Set the vector for the specified device driver.

parm1:	device number	(word/input)
parm2:	device vector	(long pointer/input)

(*) not implemented in current version of MSDOS.

Display Driver Specification

A2.1 Introduction

The software in the Sirius developed BIOS display interface receives ASCII characters and either displays them or uses them for display control. Characters in the ASCII control set (hex 00 through hex 1F and hex 7F) are used as control characters and are not displayed (see "ESC 8" exception below.) However, they may affect the display. Most of the control characters act by themselves. However, for some action more than one character is required to specify the action. This is accomplished by using the ASCII escape code (ESC--hex 1B) followed by one or more characters. The control characters and the escape sequences are described below.

A2.2 Control Characters

Bell (ctrl G, hex 07)	This is not really a display control character. It sends to the CODEC a series of signals to make the sound of a bell.
Backspace (ctrl H, hex 08)	Positions the cursor back one column. If at column 1, then position cursor at column 80 of previous row; unless at columnm 1, row 1 in which case position to column 80, row 1.
Horizontal Tab (ctrl I, hex 09)	Positions the cursor on the next tab stop. Tab stops are fixed and are at columns 9, 17, 25, 33, 41, 49, 57, 65, and 72 through 80. If the cursor is at column 80, it remains there.
Line Feed (ctrl J, hex 0A)	Positions the cursor down one row. If at row 24, then scroll display up 1 row. (May also be treated as a carriage return -- see ESC x9.)
Carriage Return (ctrl M, hex 0D)	Positions the cursor at column 1 of the current row. (May also be treated as a line feed -- see ESC x8.)
Shift In (ctrl N, hex 0E)	Shift to display character set 1 (G1).
Shift Out(ctrl O,hex 0F)	Shift to display character set 0 (G0).

A3.2 Escape Sequences

Escape Sequence/Function	ASCII Code Generated (Hexadecimal)	_____Sequence_Definition_____
CURSOR FUNCTIONS		
Esc A	1B, 41	Moves the cursor up one line
Esc B	1B, 42	Moves the cursor down one line without changing columns.
Esc C	1B, 43	Moves the cursor forward one character position.
Esc D	1B, 44	Moves the cursor backward one character position.
Esc H	1B, 48	Sets the cursor at home position.
Esc I	1B, 49	Moves the cursor to the same horizontal position on the preceding line.
Esc n	1B, 6E	Reports the cursor position.
Esc j	1B, 6A	The display driver saves the cursor position.
Esc k	1B, 6B	Returns the cursor to the previously saved cursor position.
Esc Y[l][c]	1B, 59	Moves the cursor via direct cursor addressing, where 'l' represents the hexadecimal line number and 'c' represents the hexadecimal column number. The first line and the left column are both 20 (hex) (the smallest value of the printing characters) and increase from there. Since the lines are numbered from 1 to 19 (hex) (from top to bottom) and the columns from 1 to 50 (hex) (from left to right), you must add the proper line and column numbers to 1F. No movement is done, if l and/or c are invalid.

Escape Sequence/Function	ASCII Code Generated (Hexadecimal)	_____Sequence_Definition_____
EDITING FUNCTIONS		
Esc E	1B, 45	Erases the entire screen.
Esc b	1B, 62	Erases from the start of the screen up to and including the cursor position.
Esc J	1B, 4A	Erases from the cursor position to the end of the page.
Esc l	1B, 6C	Erases entire line.
Esc o	1B, 6F	Erases the beginning of the line up to and including the cursor position.
Esc K	1B, 4B	Erases from the cursor position to the end of the line.
Esc L	1B, 4C	Inserts a blank line; the current line and all following lines are moved down one line. The cursor is moved to the beginning of the blank line.
Esc M	1B, 4D	Deletes the current line, placing the cursor at the start of the line, and moves all following lines up one line. A blank line is inserted at line 24.
Esc N	1B, 4E	Deletes cursor-position character and shifts the rest of the line one character position to the left.
Esc @	1B, 40	Enters the insert character mode, allowing insert into text on the screen. As each new character is inserted, the character at the end of the line is lost.
Esc O	1B, 4F	Exits from the insert character mode.

Escape Sequence/Function	ASCII Code Generated (Hexadecimal)	Sequence Definition
CONFIGURATION FUNCTIONS		
Esc x[Ps]	1B, 78	Sets mode(s) as follows: Ps — Mode 1 — Enable 25th line 3 — Hold Screen mode on 4 — Block cursor 5 — Cursor off 8 — Auto line feed on receipt of a carriage return 9 — Auto carriage return or receipt of a line feed A — Increase audio volume B — Increase CRT brightness C — Increase CRT contrast
Esc y[Ps]	1B, 79	Resets mode(s) as follows: Ps — Mode 1 — Disable 25th line 3 — Hold screen mode off 4 — Underscore cursor 5 — Cursor on 8 — No auto line feed 9 — No auto carriage return A — Decrease audio volume B — Decrease CRT brightness C — Decrease CRT contrast
Esc ^	1B, 5E	Toggle hold mode
Esc [	1B, 5B	Set hold mode
Esc \	1B, 5C	Clear hold mode
Esc |	1B, 7C	Activate User defined console (T.B.A.)
OPERATION MODE FUNCTIONS		
Esc p	1B, 70	Enters the reverse video mode.
Esc q	1B, 71	Exits the reverse video mode.

Escape Sequence/Function	ASCII Code Generated (Hexadecimal)	_____Sequence_Definition_____
SPECIAL FUNCTIONS		
Esc }	1B, 7D	Disables the keyboard.
Esc {	1B, 7B	Enables the keyboard.
Esc v	1B, 76	Enables wrap around at the end of the line.
Esc w	1B, 77	Disables wrap around at the end of the line.
Esc z	1B, 7A	Resets terminal to power-on configuration.
Esc $	1B, 24	Transmits the character at the cursor.
Esc]	1B, 5D	Transmits the 25th line.
Esc #	1B, 23	Transmits the page.
Esc (	1B, 28	Sets high intensity.
Esc)	1B, 29	Sets low intensity.
Esc +	1B, 2B	Clears the foreground. (High intensity displayed characters)
Esc Z	1B, 5A	Identifies display as emulating VT52 (the terminal responds with an ESC \ K).
Esc 0	1B, 30	Sets the underline mode.
Esc 1	1B, 31	Resets the underline mode.
Esc 2	1B, 32	Enables cursor blink.
Esc 3	1B, 33	Disables cursor blink.
Esc 8	1B, 38	Sets the test (literally) mode for the next single character.

Escape Sequence/Function	ASCII Code Generated (Hexadecimal)	_____Sequence_Definition_____
Esc i[n]	1B, 69	Displays the system diskette sign-on banner, as follows: (n represents the ASCII numeric character)

n	Display
0	The entire banner.
1	The company logo only.
2	The product name only.
3	Configuration information only.

A4.2 Comparison Summary

Cursor Functions	Sirius/ Victor	DEC VT52	Heathkit H19
ESC A Cusor up	x	x	x
ESC B Cusor down	x	x	x
ESC C Cusor forward	x	x	x
ESC D (BS 08H) Cusor back	x	x	x
ESC H Home	x	x	x
TAB (09H) Tab	x	x	x
LF (0AH) Line Feed	x	x	x
CR (0DH) Carriage Return	x	x	x
ESC I RVS Line Feed	x	x	x
ESC Y Address Cursor	x	x	x
ESC n Report Cursor	x		x
ESC k Restre Cursor	x		x
ESC j Save Cursor	x		x
Editing Functions			
ESC J Erase EOS	x	x	x
ESC K Erase EOL	x	x	x
ESC b Erase SOS	x		x
ESC l Erase Line	x		x
ESC o Erase SOL	x		x
ESC E Erase Screen	x		x
ESC L Insert Line	x		x
ESC M Delete Line	x		x
ESC @ Insert Mode	x		x
ESC O Exit Insert Mode	x		x
ESC N Delete Char	x		x
Special Functions			
ESC F Graphics Mode On	x	x	x
ESC G Graphics Mode Off	x	x	x
ESC = ALT. Keypad on		x	x
ESC > ALT. Keypad off		x	x
ESC $ Hold mode on	x	x	x
ESC / Hold mode off	x	x	x
ESC [Transmit 25th line	x		x
ESC # Transmit Page	x		x
Bell (07H)	x	x	x
(13H) (11H) Xon Xoff protocol		x	x
(00H) (7FH) NUL DEL Ignored	x	x	x
ESC { Enable Keyboard	x		x
ESC } Disable	x		x
ESC + Clear Foreground	x		
ESC 8 Test Mode	x		
ESC v Enable Wrap	x		x
ESC w Disable Wrap	x		x
ESC p Reverse ON	x		x

		Sirius/ Victor	DEC VT52	Heathkit H19
ESC q	Reverse OFF	x		x
ESC (	High Intensity ON	x		
ESC)	High Intensity OFF	x		
ESC 0	Underline ON	x		
ESC 1	Underline OFF	x		
ESC 2	Blink Cursor ON	x		
ESC 3	Blink Cursor OFF	x		
ESC 4	Set Key Function	x		
ESC i	Display system information	x		
ESC z	Reset	x		x
ESC Z	Identify as VT52	x	x	x
ESC	Enter ASCII mode			x
ESC t	Enter keypad shift mode			x
ESC u	Exit keypad shift mode			x

Set Reset Modes

			Sirius/ Victor	DEC VT52	Heathkit H19
ESC x/y	1	25th line	x		x
ESC x/y	2	Key Click			x
ESC x/y	3	Hold Screen	x		x
ESC x/y	4	Block Cursor	x		x
ESC x/y	5	Cursor Display	x		x
ESC x/y	6	Kpd Shifted			x
ESC x/y	7	Alt Keypad			x
ESC x/y	8	Auto LF	x		x
ESC x/y	9	Auto CR	x		x
ESC x/y	A	Volume Increase/Decrease	x		
ESC x/y	B	Brightness " "	x		
ESC x/y	C	Contrast " "	x		

Design notes on Vedio display
and keyboard testing

Cursor - The VT52 uses a blinking underscore. The Z19 uses a blinking box. The Sirius uses a nonblinking box. The Sirius and Z19 can change the cursor type to several forms.

Repeat key - The VT52 describes the rate of repetition may attain 30 cps but no futher characteristics are known. The Z19 describes the rate of repetition at 15 cps but no futher characters are are described. The Sirius repeat key will not start repetition for 1/2 second then reptition occures at 20 cps.

Cursor movement - The VT52 cursor moves to the right one position after each displayable character is typed until the 80th column is reached. The cursor will not move past the 80th column and each succeeding character over writes the preceeding one with the cursor over the character. The Z19 uses wrap around on start-up but can be changed to emulate the VT52. The Sirius emulates the Z19.

Return key - The VT52, Z19 and Sirius perform this action identically. The cursor is move the 1st column without changing lines. However the Z19 and Sirius have addational Escape sequences to alter this action. The control code M (^M, 0DH) causes the identical actior.

Line feed - The cursor is moved down one line (row) without changing columns. upward scrolling occures if the cursor is at line 24. Scrolling causes line 1 to be replaced by line 2 and line 2 to be replaced by line 3 ect. ect. until line 24 is replaced with 80 spaces and the cursor placed in the column determined by the action that cuased the scrolling. In this case the cursor does not change columns. The control code J (^J, 0AH) causes the identical action.

Tab key - The cursor moves to the right at least one column, and continues moving right until it reaches a horizontal TAB stop. It stays on the same line. The TAB stops are fixed in columns 9, 17, 33, 41, 49, 57, 65, and 73. If the cursor was at TAB stop to begin with, it moves rightward to the next TAB stop. If the cursor was in columns 73 - 79 it simply moves rightward one column until it reaches column 80, it will not move past the 80th column. The control code I (^I, 09H causes the identical action.

Back space key - The cursor is moved on coluumn to the left. (If the cursor was at the start of a line, it will not go past the first column.) No characters are replaced. The control code H (^H, 08H) causes the identical action.

Shift lock key - The ASCII characters in the code range 61H to 7AH (a-z) are converted the upper case only when this key is pressed. Pressing this key again reverses the shifting action so that these characters can be type in both lower and upper case depending on the pressing of the shift keys. No code is generated.

Bell - The control code G (^G ,07H) causes a tone sound. There are no futher specifications described for the VT52. The Z19 describes the frequency of the bell at 1000 Hz for 200 milliseconds.

- The VT52 and Z19 both use the audio to output a 'click' as each key is pressed. The Z19 describes the 'click' as the same 1000 Hz tone with a duration of six milliseconds. There is no description for the

cancelled.

25th line - The 25th line is considered a seperate 1 line terminal displa seperate form the first 24 lines on the screen. The 25th line can be enabled or disabled allowing or disallowing operations to take place there. All the operations available for the fisrt 24 line also operate on the 25th line. However some operations will display no action due to the screen being limited to 1 line.

---

* Note: VT52 compatible escape sequences

Note: VT52 escape sequence actions;

When a VT50 series terminal receives ESC, it will interpret rather than display the next displayable character. This displayable character should directly follow ESC. It is known as the 'final character' of the Escape Sequence. If a control code is sent to the terminal between the ESC and the final character, the function specified by the control code is performed when the control code is received, and the function specified by the Escape Sequence is performed when the final character is received.

note: The above is quoted from the DECscope User's Manual. pg. 1: The Z19 does not describe Escape sequence action. My experience shows that Zenith follows the DECscope description.

---------- ESCAPE SEQUENCES ----------

Sequence	Hex code	Sequence Defination and test notes
*Esc H	1BH,48H	The cursor is moved to the HOME position - the character position at the upper left corner of the screen. (row 1, column 1). Esc H works from the 25th line and is not affected by the status of the 25th line. Test: the screen should be cleared then the cursor should be place in some position on the screen and then "HOME"ed then a character printed in the HOME position and the cursor "HOME"ed again the procedure is repeated from the positioning of the cursor until all positions on the screen are tested. A successful test will have one character in the HOME position.
*Esc C	1BH,43H	The cursor is moved one column to the right. The cursor will not move past the 80th column. This escape sequence is not affected the wrap around flag. This has the identical action on the 25th line if the 25th line is enabled. Test: The screen should be cleared and the cursor "HOME"ed. Each row should be labeled 1-9,and A-I. The cursor should be place on a row then moved right by the sequence move than 80 times and the process repeated until all 24 lines are tested. A sucessful test will not affect the row numbering or cause scrolling.
*Esc D	1BH,44H	The cursor is moved one column to the left. The

the line is enabled.

Test: Same as Esc C test, except the cursor is placed at the 80th column.

Esc B 1BH,42H

The cursor is moved down one row without changing columns. The sequence will not cause the cursor to move past the 24th row. Another words no scrolling will occure. This has no action on the 25th line.

Test: The screen should be cleared. A character is printed on row 1 of some column and the cursor moved over this character. The escape sequence is repeated until all columns on row 1 are filled with the character and then repeated several more times. A sucessful test will leave the top row intact not causing scrolling.

*Esc A 1BH,41H

The cursor is moved up one row without changing columns. The sequence will not cause the cursor to move past the 1st row. That is, no reverse scrolling should occure. This has no action on the 25th line.

Test: Same as above but the characters are placed in the 24th row. A sucessful test will leave the bottom row intact not causing scrolling.

*Esc I 1BH,49H

Causes the same action as Esc A except causing reverse scrolling to occure. If the cursor is in the 25th line the line should be cleared. However in the Z19 no action odccures.

Test: Five rows of characters should be placed in rows 19-24, the cursor moved to some column on the 24th row the escape sequence is repeated 28 times. This proceedure is repeated until all 80 columns have been tested. A sucessful test will leave only one line of characters on the 24th row.

*Esc Y[l][c]
1BH,59H,lH,cH

The cursor is directly moved to the row [l] and the column [c]. The [l] and [c] codes are formed as Hexidecimal numbers. To avoid conflict with the first 32 ASCII control characters, an offset of 1FH must be added to the desired row and column numbers. Example: to position cursor at row 10 and colum 20, 10D (decimal) = 0AH and 20D = 14H. The final code is [l] <= 20H = 0AH + 1FH and [c] <= 33H = 14H + 1FH.

Test: The screen should be cleared. Several sentences could be printed by alternating direct cursor positioning of each character with a random positioning of the cursor. A sucessful test would print the sentences correctly without causing another video event from occuring.

* Special note: the Z19 executes the [l] and [c] as soon as they are received. If the line is out of range the cursor will not change lines while the cursor will move to the proper column. If the column is out of range, the cursor will move to the proper line and goto column 80. If both line and colum are

the end of the page are changed to spaces. If the cursor is on the 25th line, the Esc J is the same as Esc K and has the identical action as on any line.

Test: The screen should be filled randomly with characters, the cursor should be place randomly on the screen and a sentence printed pointing out that all characters following it should become spaces. This proceedure should be repeated say 20 times with the cursor being placed at least at the home position and the 24th row and 80th column. A sucessful test will leave the screen with no characters following the cursor position.

*Esc K 1BH,4BH

All characters for the current cursor position to the end of the current line are replaced with spaces. This has identical effect if the cursor is on the 25th line.

Test: The screen is filled with some character. The cursor is "HOME"ed. The escape sequence is executed for each line while the cursor is in column 1, the line is refilled with the characters and the next line is tested with the escape sequence until all 24 lines are cleared and replaced. A sucessful test will cause a blank line to move down the screen leaving the screen filled with characters.

*Esc Z 1BH,5AH

This sequence causes the display to respond with an escape /A(VT50) or /H(VT50H) or /K(VT52) or /C(VT55) or /J(VT50H with copier) or /L(VT52 with copier).

Test: The escape sequence will cause the Sirius system to print on the screen ^[K.

---------- Z19 ESCAPE SEQUENCES ----------

Esc n 1BH,4EH

Reports the cursor position to the video display screen as characters formed from the Hex code position. This acts identical if the cursor is on the 25th line.

Test: Print a message describing where the cursor will be positioned in hex. Move the cursor there execute an Esc j then move back to the message then execute the Esc n sequence. A sucessful test will leave matching cursor code positions. What happens when the cursor is moved off screen then the Esc n executed?

Esc j 1BH,6AH

Save the cursor position in the BIOS display driver. This acts indentical if the cursor is on the 25th line.

Test: Move the cursor randomly to some position. Excute the sequence then print a pointer to the position then move the cursor to several other positions then ask the cursor to be moved back using then Esc k sequence. This proceedure would be repeated 20 times each time leaving a special character where the cursor was moved back to.

Test: Print several sentences, one character at a time. The character is printed, the cursor is saved by the Esc j sequence, the cursor is moved to some other position (sometimes off page), the cursor is restored via the Esc k sequence, and the proceedure repeats until all the characters of the sentences are printed. A sucessful test will print readable sentences.

Esc E 1BH,45H

Erases all the characters on the screen. The screen is filled with spaces (20H) and the cursor is placed in the home position. If the cursor is on the 25th line, the 25th line should be cleared and the cursor put in column 1. However the Z19 clears the screen but leaves the cursor position unchanged.

Test: The screen should be completely filled with displayable characters. A message should be output explaining the the screen will be cleared with enough delay to see the message then the Esc E executed. A sucessful test will leave the cursor in the home position and no displayable characters on the screen.

Esc b 1BH,62H

Erases the display from the start of the screen to and including the cursor position leaving the cursor position unchanged. This is not affected by wrap around mode. This acts identically if the cursor is on the 25th line.

Test: The screen is filled to some random position in the screen, a message is output describing the test, a delay is set so that the messages can be read, and the Esc b is executed. A sucessful test will clear the screen up to the cursor without affecting any other portion of the screen.

Esc l 1BH,6CH

Erases the entire line, including the cursor position the cursor position remains unchanged. This is not affected by wrap around mode. This acts identically if the cursor is on the 25th line.

Test: Three rows of characters are printed. An arrow is positioned to point to where the cursor is to be. The cursor is moved to the middle row and the Esc l executed. A sucessful test will clear the middle row only and leave the cursor where the arrow is pointing

Esc o 1BH,6FH

Erases from the beginning of the line to and including the cursor position. The cursor position remains unchanged. This is not affected by wrap around mode. This acts identically if the cursor is on the 25th line.

Test: Same test as above. A sucessful test will leave the cursor position unchanged with all characters to left of the cursor to the beginning of the line cleared.

Esc L 1BH,4CH

Inserts a new blank line by moving the line that the cursor is on, and all the following lines, down one

cursor in column 1. However the Z19 takes this action after it splits the line from the right of the cursor position 2 characters and places the left portion on row one right justified and it places the right portion on row 2 left justified.

Test: Three lines of characters are printed together, then two lines of text should be printed on rows 23 and 24. The 23d row line describes that it will remain while the 24th row will explain it will be scrolled off. A delay is set to allow the reading of of the last to sentences and the Esc L executed. A sucessful test will clear a line where the cusor was leaving the cursor in column one and only 23d line now on the 24th row.

Esc M 1BH,4DH

Deletes the contents of the line the cursor is currently on moving the cursor to column 1 and moving all following lines up one row adding a blank line on row 24. This is not affected by wrap around mode. If the cursor is on the 25th line the should effectively be cleared. However on the Z19 takes some portion of row 1 and the remaining columns are taken from row 2 the combine 80 characters are put on line 25.

Test: Three lines of text are printed on three consecutive rows. The remaining rows below are filled with characters. The cursor is move some place on the middle line. The Esc M is executed. A sucessful test will delete the line the cursor was on and the following lines moved up one row with row 24 filled with spaces.

Esc N 1BH,4EH

Deletes the character at the cursor position and shifts all the following characters on that line one column to the left while adding a space at column 80. Only the current line that the cursor is on is affected. This sequence is not affected by the wrap around mode. If the cursor is on the 25th line the action is identical.

Test: Three lines of text are printed on three consecutive rows. The cursor is move to some column on the middle line and the Esc N is repeated more than the number of columns on the screen. A sucessful will leave the cursor position unchanged and spaces from the cursor to the end of the line.

Esc @ 1BH,40H

Enter insert character mode. This mode allows the insertion of characters into a line the characters at the 80th column are lost. This mode is unaffected by the wrap around mode. If the cursor is on the 25th line the action is identical.

Test: Three lines of text are printed on consecutive rows. The cursor is moved to some column on the middle row. The Esc @ is executed and more than the number of columns on the screen are filled with a character. A sucessful test will fill the line with the character. If the wrap around mode is enabled, the test will cause the next line to be filled with

leading Escape sequence. If the cursor is on the 25th line the action is identical.

Test: the test is the same as above except that the Esc) and Esc 0 are executed alternately. A message will be output describing which mode is in affect and the results that should be observed.

Esc z 1BH,7AH

Nullifies all previously set escape modes and returns to the power-up configuration.

* special note: The Z19 defines the power-up configuration as the mode set in the configuration switches S401 and S402. S401 sets the buad rate, parity, and duplex. S402 set the following.

0	0=underscore cursor	1=block cursor
1	0=key click	1=no key click
2	0=discard past 80th column	1=wrap around
3	0=no auto LF on CR	1=auto LF on CR
4	0=no auto CR on LF	1=auto CR on LF
5	0=Heath mode	1=ANSI mode
6	0=unshifted keypad	1=shifted keypad
7	0=60 Hz refresh	1=50 Hz screen refresh

The Z19 clears the screen including to 25th line in this sequence.

* Sirius does not have a specification for the power-up condition. But, obviously the Heath escape mode is the default as is the 60 Hz screen refresh and most likely to unshifted keypad. Currently the cursor is block mode, no key click, wrap around, no auto LF on CR or CR on LF, the screen is cleared including the 25th line.

Test: no test can be preformed until a specification for the defaults is defined.

Esc x [n] 1BH,78H

Certain operation modes can be enabled or disabled with this sequence. The operation modes are as follows.

1 = Enable the 25th line. This allows the use of the 25th line. The cursor can be moved to the 25th line only with the Esc Y[l][c] and the Esc k sequences. The 25th line acts like a 1 line terminal. All operations that are used on the 24 line screen operate on the 25th line. However, some of these operations will show no effect.
3 = Hold Screen mode.
4 = Block cursor on.
5 = Display cursor off
8 = Auto LF on CR
9 = Auto CR on LF
* A = Volume increase step
* B = Brightness increase step
* C = Contrast increase step

* Note: supported by Sirius but not on Z19
mode 2,6,7 not implemented on Sirius

on the 25th line
3 = Exit Hold screen mode
4 = Underline cursor
5 = Display cursor on.
8 = No Auto LF on CR
9 = No Auto CR on LF
* A = Volume decrease step
* B = Brightness decrease step
* C = Contrast decrease step

Esc [1BH,5AH

Enter Hold Screen mode. The Hold screen mode allows the user to hold output to the screen. Keypress of the unshifted scroll will cause on line of text to be output to the screen. Keypress of the shifted scroll causes 24 lines to be output to the screen. Remember when the cursor is at the start of a line of text, the screen is probably waiting for a scroll command. If the cursor is on the 25th line unshifted scroll should act identically. The shifted scroll should cause the 25th line to be replaced with the 24th received line of text after the keypress.

Test: A message should be output explaining the effects of the test. Then text will be output to the screen. The user will press the scroll key and confirm the action. The user will press the shifted scroll and confirm the action.

Esc \ 1BH,5CH

Exits the Hold Screen Mode. This nullifies the above Esc [sequence.

Test: As above a message should be output. Then text should be output to the screen. The user will press the shifted and unshifted scroll key and confirm that they cause no action on the output.

Esc p 1BH,70H

Enter reverse video mode. All characters are printed in a reversed field. That is all characters are normally green on a black field, the reverse is black characters on a green field. This action is identical on the 25th line.

Test: Simply print part of a message in normal characters and part of the message in reverse video. A successful test will show reverse video and normal video where the text describes.

Esc q 1BH,71H

Exit reverse video mode. This nullifies the Esc p sequence. This action is identical on the 25th line.

Test: Same as above.

Esc } 1BH,7DH

Inhibits the output from the keyboard.

) { 1BH,7BH

A computer sent code to enable the keyboard after it has be disabled by the Esc } sequence.

Test: A message is output describing that the keyboard is disabled. The user is asked to try any keypresses. The keyboard is reenabled and the keyboard presses are echoed. A sucessful test will not echo the keypresses while the keyboard is disabled.

If the character was already on the last row when the attempt was made, the screen is scrolled and the character moved to column 1 of the last row on the screen. If the character to print past the end of the line is on the 25th line, the line should be cleared and the character place in column 1. However the Z19 does not clear the line but overwrites in column 1.

Test: Simply print a line that is longer than the line lenght. A second line should be printed from the last line of the screen to demonstrate the scrolling. A successful test will will show continuation on the following line.

Esc w 1BH,77H

Discard end of line. After the last column in a line the characters are overwritten in the last column of the line. If the cursor is on line 25, the action is identical.

Test: Write a message that has more characters than the line lenght. a successful test will print only one line while overwritting the last column of the line.

A.type videsign.doc

Design notes on Vedio display
and keyboard testing

Cursor - The VT52 uses a blinking underscore. The Z19 uses a blinking box. The Sirius uses a nonblinking box. The Sirius and Z19 can change the cursor type to several forms.

Repeat key - The VT52 describes the rate of repetition may attain 30 cps but no futher characteristics are known. The Z19 describes the rate of repetition at 15 cps but no futher characters are are described. The Sirius repeat key will not start repetition for 1/2 second then

A>

132 COLUMN SPECIFICATION

PURPOSE:

The 132 column module provides a simulated 132 column display in the 800 dot by 400 line HIRES display mode of the SIRIUS computer.

RESULT:

The normal VT52 print interface recognizes an escape sequence to enable the 132 column mode. The characters are displayed in a 5 by 7 dot matrix in a 6 by 10 cell to give the 132 column display. A standard display of 80 columns by 25 lines is also simulated with an 8 by 11 dot matrix in a 10 by 16 cell.

INSTALLATION:

The 132C program installs itself into the BIOS through a super BIOS call. The program requires approximately 50K of memory. The 800 by 400 HIRES screen requires 40K, the character sets and code are the rest. The 132c program copies itself into the lowest 64K block of memory and removes the 50K required bytes from the system permanently.

CONTROL CODES:

BELL - CTRL G (07H)
Generate a bell sound. Pass thru to VT52

BACKSPACE - CTRL H (08H)
Positions the cursor back one column. If wrap around mode is enabled and the cursor was at column 1, then position cursor at last column of previous row; unless at column 1, row 1 in which case position to last column in row 1. If in discard mode, then the cursor will not move from column 1.

HORIZONTAL TAB - CTRL I (09H)
Positions cursor the cursor forward to the next tab stop. Tab stops are fixed and are at columns 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, and 129. If the cursor is at the last column it remains there.

LINE FEED - CTRL J (0AH)
Moves cursor to next line same horizontal position, scrolls the screen up if a line feed occures on the bottom line. If the cursor is on bottom line + 1 then no action is taken.

RETURN - CTRL M (0DH)
Moves cursor to left most column of same line.

SHIFT OUT - CTRL O (0EH)
Switch the character cell size to 6 by 10 resulting in a display of 133 columns by 40 lines. The top of screen will be set to line 1 and the bottom of screen will be set to line 39. The bottom line + 1 will be line 40 similar to VT52's 25th line. The cursor will home.

SHIFT IN - CTRL N (0EH)
Switch the character cell size to 10 by 16 resulting in a display of 80 columns by 25 lines. The top of screen will be set to line 1 and the bottom of screen will be set to line 24. The cursor will home.

CANCEL _ CTRL X (18H)
Abort any Escape sequence in progress. Starts displaying characters as normal ASCII.

ESCAPE - CTRL [(1BH)
Start an escape sequence.

ESCAPE SEQUENCES:

TRANSMIT PAGE - ESC # (1BH,23H)
Will transmit only a RETURN (0DH) LINE FEED (0AH).

TRANSMIT CHARACTER AT CURSOR - ESC $ (1BH,24H)
Will transmit only a RETURN (0DH) LINE FEED (0AH).

SET HIGH INTENSITY - ESC ((1BH,28H)
Simulates high intensity mode by shadow printing.

SET LOW INTENSITY - ESC) (1BH,29H)
Prints normal characters.

ENTER UNDERLINE CHARACTER MODE - ESC 0 (1BH,30H)
Sets the underline mode.

EXIT UNDERLINE CHARACTER MODE - ESC 1 (1BH,31H)
Resets the underline mode.

SET KEY VALUE - ESC 4[n][lk][kc] (1BH,34H,XXH,XXH,XXH)
Set key value. Five characters are passed thru to VT52 to set new key values.

LITERAL CHARACTER - ESC 8 (1BH,38H)
Display next character literally.

ENTER INSERT CHARACTER MODE - ESC @ (1BH,40H)
Enters insert character mode, allowing insert into text on the screen. As you type in new characters, existing text to the right of the cursor shifts to the right. As each new character is inserted, the character at the end of the line is lost.

CURSOR UP - ESC A (1BH,41H)
Moves cursor up one line without changing columns. If the cursor reaches the top line, it remains there and no scrolling occurs. No action taken on bottom line + 1.

CURSOR DOWN - ESC B (1BH,42H)
Moves cursor down one line without changing columns. The cursor will not move past the bottom line and no scrolling will take place. No action taken on bottom line + 1.

CURSOR FORWARD - ESC C (1BH,43H)
Moves cursor one character position to the right. If the cursor is at the right end of the line, it will remain there.

ESCAPE SEQUENCES:

CURSOR BACKWARD - ESC D (1BH,44H)
Moves the cursor one character position to the left. If the cursor is at the start (left end) of a line, it will remain there.

CLEAR DISPLAY - ESC E (1BH,45H)
Erase the screen from the defined top line to the defined bottom line. If on bottom line + 1 the erases bottom line + 1 only. Places the cursor in the home position.

ENTER GRAPHICS MODE - ESC F (1BH,46H)
VT52 graphics characters appear in character numbers 94 to 127 of the ASCII character set.

EXIT GRAPHICS MODE - ESC G (1BH,47H)
Normal lower case characters appear in character numbers 94 to 127.

CURSOR HOME - ESC H (1BH,48H)
Moves the cursor to the first character position on the defined top line.

REVERSE LINE FEED - ESC I (1BH,49H)
Moves the cursor to the same horizontal position on the preceding line. If the cursor is on the defined top screen line, a scroll down is performed. No action taken if on bottom line + 1.

ERASE TO END OF PAGE - ESC J (1BH,4AH)
Erases all the information from the cursor (including the cursor position) to the end of the defined bottom line. If on bottom line + 1 then erase to end of line only.

ERASE TO END OF LINE - ESC K (1BH,4BH)
Erases from the cursor (including the cursor position) to the end of the line.

INSERT LINE - ESC L (1BH,4CH)
Inserts a new blank line by moving the line that the cursor is on, and all the following lines, down one line, to the defined bottom line. Then the cursor is moved to the beginning of of the new blank line. No action taken if on bottom line + 1.

DELETE LINE - ESC M (1BH,4DH)
Deletes the contents of the line that the cursor is on, places the cursor at the beginning of the line, moves all the following lines up one line, and adds a blank line at the defined bottom line. No action taken if on bottom line + 1.

ESCAPE SEQUENCES:

DELETE CHARACTER - ESC N (1BH,4EH)

Deletes the character at the cursor position and shifts any existing text that is to the right of the cursor one character position to the left.

EXIT INSERT CHARACTER MODE - ESC O (1BH,4FH)

Exits from insert character mode.

DIRECT CURSOR ADDRESSING - ESC Y[l#][c#] (1BH,59H,XXH,XXH)

Moves the cursor to a position on the screen by entering the escape code, the character which represents the line number, and the character which represents the column number. The 'l#' represents the hexadecimal line number and 'c#' represents the hexadecimal column number. The first line and the left column are both 20 (hex) (the smallest value of the printing characters) and increase from there. Since the lines are numbered from 1 up (from top to bottom) and the columns from 1 up (from left to right), You must add the proper line and column numbers to 1F (hex). If the line number entered is smaller than the defined top line, the cursor will be positioned to the top line. If the line number entered is greater than the defined bottom line + 1, the cursor will not more from it present line. If the column number is too high, the cursor will move to the end of the line.

IDENTIFY AS VT52 - ESC Z (1BH,5AH)

CONIN responds with "ESC/K" to indicate that it can perform as VT52.

TRANSMIT BOTTOM LINE - ESC] (1BH,5DH)

Will transmit only a RETURN (0DH) LINE FEED (0AH).

TOGGLE DEBUG MODE - ESC _ (1BH,5FH)

Toggles debug mode on or off. In debug mode the bottom line + 1 displays the hex codes for print stream.

ERASE BEGINNING OF DISPLAY - ESC b (1BH,62H)

Erases from the start of the screen to the cursor, and includes the cursor position.

REVERSE TAB - ESC h (1BH,68H)

Moves cursor left to next mod 8 position. Stops on left side of screen.

ESCAPE SEQUENCES:

SAVE CURSOR POSITION - ESC j (1BH,6AH)
The present cursor position is saved so the cursor can be returned here later when given the set cursor to saved position command.

SET CURSOR TO SAVED POSITION - ESC k (1BH,6BH)
Returns the cursor to the position where it was when it recieved the save cursor position command. If the line number restored is smaller than the defined top line, the cursor will be positioned to the top line. If the line number restored is greater than the defined bottom line + 1, the cursor will not more from it present line. If the column number is too high, the cursor will move to the end of the line.

ERASE ENTIRE LINE - ESC l (1BH,6CH)
Erases all the line including the cursor position.

SET SIZE - ESC m[c1][c2][c3] (1BH,6DH,XXH,XXH,XXH,)
If c1 is ASCII "1" then set cell width and height. The c2 character equals width+1FH and the c3 character equals height+1FH. Width and heigth can range from 1 to 16. The maximum number of of character columns and lines are determined by: max columns = 800/width, max lines = 400/height. After set cell size the top line is set to 1 and the bottom line is set to max lines - 1. The cursor will home after the cell size is set.
If c1 is ASCII "2" then set screen top and bottom. The c2 character equals top line+1FH and the c3 character equals bottom line+1FH. Top line and bottom line can range from 1 to max lines with bottom line always greater than or equal to top line. The cursor will home after the screen size is set.

CURSOR POSITION REPORT - ESC n (1BH,6EH,XXH,XXH)
CONIN reports the cursor position in the form of ESC Y line# column#.

ERASE BEGINNING OF LINE - ESC o (1BH,6FH)
Erases from the beginning of the line to the cursor, and includes the cursor position.

ENTER REVERSE VIDEO MODE - ESC p (1BH,70H)
Enters the reverse video mode so that characters are displayed as black characters on a white background.

EXIT REVERSE VIDEO MODE - ESC q (1BH,71H)
Exits the reverse video mode.

ESCAPE SEQUENCES:

WRAP AROUND AT END OF LINE - ESC v (1BH,76H)
A print to the last column of the line will position to the first column of the next line. The page scrolls up if necessary.

DISCARD AT END OF LINE - ESC w (1BH,77H)
A print to the last column of the line will not change the cursor position and overprinting occures. Therefore, only the last character received will be displayed in the last column position.

SET MODE(S) - ESC x [Ps] (1BH,78H,XXH)
Sets the following modes, where Ps equals:
1 = Enable bottom line
4 = Block cursor
5 = Cursor off
8 = Auto line feed on receipt of CR
9 = Auto CR on receipt of line feed
A = Send to VT52
B = Send to VT52
C = Send to VT52

RESET MODE(S) - ESC y [Ps] (1BH,79H,XXH)
Resets the following modes, where Ps equals:
1 = Disable bottom line
4 = Underscore cursor (ok if character height > 9)
5 = Cursor on
8 = No auto line feed
9 = No auto CR
A = Send to VT52
B = Send to VT52
C = Send to VT52

RESET TO POWER-UP CONFIGURATION - ESC z (1BH,7AH)
Reset back to 80 column mode.

KEYBOARD ENABLED - ESC { (1BH,7BH)
Enables the keyboard after it was inhibited by a keyboard disable command. Pass thru to VT52.

ENABLE 132 COLUMN DISPLAY - ESC | (1BH,7CH)
Enable 132 column mode.

KEYBOARD DISABLE - ESC } (1BH,7DH)
Inhibits the output of the keyboard. Pass thru to VT52.

PORT CONFIGURATION USER GUIDE

The Port Configuration utility provides for the modification of the serial and parallel ports' device assignment and/or attributes.

INTRODUCTION

The Port Configuration program provides the capability to change communications and printer device assignments (and/or associated attributes) after the Victor 9000 is booted up and set to its Sys Gen states. The Sys Gen parameters for the two serial and the one parallel port may be modified while the system is up and without resorting to a complete Sys Gen process. For example, this utility allows for easy toggling between a connected serial and parallel printer.

It is required that the system diskette be loaded in one of the drives. The Port Configuration utility may be invoked to modify or restore the port assignments and attributes.

Rebooting or off/on cycling of the computer restores the system to the original Sys Gen assignments.

INVOKING PORT CONFIGURATION (PORTCONF)

To invoke PORTCONF, enter the following input from the keyboard from the Operating System prompt.

PORTCONF .

Screen Response:

VICTOR 9000 COMMUNICATION CONFIGURATION PROGRAM

SELECT

1. Port A (RS-232)
2. Centronics/Parallel
3. Port B (RS-232)

To select a listing device enter a number from menu:

If 1 or 3 above is selected, the following menu occurs:

COMMONLY USED TRANSMISSION SPEEDS

BAUD RATES

A. 50
B. 75
C. 110
D. 134.5
E. 150
F. 300
G. 600
H. 1200
I. 1800
J. 2000
K. 2400
L. 3600
M. 4800
N. 9600

To set transmission speed enter a letter from menu:

Return to the Operating System is automatic.

If No. 2 is selected from the main menu (Centronics/Parallel), the system is set to drive a parallel printer. Return to the Operating System is automatic.

Character Set Editor - User Notes

Introduction

CEDIT is an interim program, between EDOT and the new character editor, which is used to edit or create character set tables.

Files

You must have the following files to run CEDIT:

ALLOC.CMD	Machine language interface.
PBASIC86.CMD	Microsoft run-time BASIC interpreter.
CEDIT.BAS	The edit program.
*.CHR	The character set files.
CHARGEN.SUB	Submit file, optional.

Invoking

To run CEDIT do either:

```
A>SUBMIT CHARGEN(cr)
```

or

```
A>ALLOC(cr)
A>PBASIC86 CEDIT(cr)
```

CEDIT must be run on a system configured with a logo character set.

It then prompts you for the source character set files to be collected and edited (appear on the right-half of the screen). Enter the file names (.CHR is default) followed by a (cr). After the last one, enter "END".

You are then prompted for one more character set. This is the one which is to be modified and then written out. It appears on the left-half of the screen.

After this character set is read, all the character sets are displayed and the program is in "COPY" mode.

CEDIT Modes

COPY	Copies characters from the right-half of the screen to the left-half. It replaces the character under the left cursor by the character under the right cursor.
DISPLAY	Displays the dot pattern of the character under the right cursor.
EDIT	Allows changes to the dot pattern of a character.

Using CEDIT

In all modes, the number 7, 8, 9, 4, 6, 1, 2, 3 keys are used to move the cursor. The direction of the movement is illustrated by this diagram, where the cursor is at the X and the movement is in the direction of the number pressed:

```
7  8  9
4  X  6
1  2  3
```

The number 5 key causes different actions, depending on the mode. In the COPY mode, it causes a character copy; in the DISPLAY mode, it enters the EDIT mode; and in the EDIT mode, it inverts the current dot.

COPY mode commands:

R	Activate right cursor only.
L	Activate left cursor only.
B	Activate both cursors.
E	Enter DISPLAY mode.
W	Write left character set.

DISPLAY mode commands:

Only the number keys are used in DISPLAY mode. However, any of the COPY mode commands, except E, return the program to the COPY mode and is performed.

EDIT mode commands:

These commands are the same as the old EDOT commands.

E or P	Position only mode.
T	Toggle mode.
S	Set mode.
R	Reset mode.
C	Clear character.
(cr)	Return to DISPLAY mode.

Saving the Character Set

In either COPY or DISPLAY mode type a "W". This command writes the left character set to the designated file.

You are prompted for a file name, if you respond with just a (cr) then the name of the input file is used. You are then prompted for any changes to the header record.

For now, 256 characters are always written, regardless of the set size.

USER GUIDE FOR CHARACTER GENERATION AND EDIT (CEDIT)

G. Gleason
March 23, 1982

CREATING AND EDITING CHARACTERS UNDER OPERATING SYSTEM 2.1

USER'S GUIDE

INTRODUCTION

CEDIT is used to create new character sets from existing ones. It also includes the ability to create new characters by editing the dot configuration of old characters. By clearing existing characters, new characters can be created from scratch.

FILES

You must have the following files to run CEDIT:

ALLOC.CMD	Machine language interface
BASIC86.CMD	Microsoft basic
CEDIT.BAS	The edit program
*.CHR	Character set files

CEDIT MODES

Mode	Description
Copy	Copies characters from right-half of screen to left-half - substitute the character under the right cursor to position under left cursor.
EDIT	Changes dot configuration of character.
DISPLAY	Displays the character that is under the cursor on the right-half. The display appears on the left-half in bit cell detail.

Source character sets appear on the right-half; new or edited set appears on the left-half.

INVOKING

Type the following:

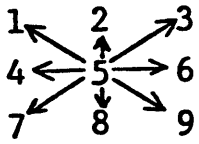
```
A>ALLOC(cr)
A>BASIC86 CEDIT(cr)
```

It will then prompt you for source character set names to be collected and edited (appears on the right-half of screen). Enter the names in caps without an extension (.CHR assumed) followed by a carriage return. After the last one, enter END. You will be prompted for one more character set.

This is the one you will modify and then write out (save). This appears on the left-half of screen. When this is done, the character sets will all be displayed and you will be in the "COPY" mode with no cursors displayed and no control functions enabled.

USING CEDIT

In all modes, the number keys move the cursor as indicated in this diagram:



The "5" key copies the character from under the right cursor to under the left one in COPY mode, enters EDIT mode when in DISPLAY mode, and inverts the current dot in EDIT mode.

Here are the commands in COPY mode:

R - use right cursor
L - use left cursor
B - use both cursors
E - enter DISPLAY mode
W - write character set

In DISPLAY mode, the cursor keys work on the right-hand cursor to select the displayed characters. The same commands are available only "E" has no effect and the others take you back to COPY mode.

EDIT mode is the same as the old EDOT with the following commands:

E or P	-	position only mode
T	-	toggle mode
S	-	set mode
R	-	reset mode
C	-	clear character to all off
(CR)	-	exit to display mode

When you are done and ready to save the character set, use the "W" command. You will, again, be prompted for a file name. Same rules apply as for input files. The character set will be written with a header record. This record will have a "C" in the first byte to identify it as a character set. The rest of the header will be blank. For now, 128 characters are written regardless of actual set size.