

# **CHAPTER 8**

## **Functional Specifications of the PC-8401A Screen Driver**

NEC Corporation  
Copyright © 1984 by NEC Corporation  
All Rights Reserved

## SECTION 1

### GENERAL INFORMATION

#### *CALLING CONVENTION*

The Extended BIOS functions are called through use of the RST 2 instruction. The C register contains the function code at entry.

For value passing, the extended BIOS functions use a similar convention to BDOS functions. The E register is used to pass a byte value to the extended BIOS, and the DE register pair is used to pass a word value. Some functions use other registers as well to pass various additional values.

The return value is usually passed in A if it's a byte, or in HL if it's a word.

#### *REGISTER PRESERVATION.*

All the CPU registers, except the AF and AF' registers, are preserved unless they are used to return a value. The AF and AF' registers are always altered even when they are not used to return values.

#### *CURSOR*

The cursor is automatically erased during the execution of each screen function and restored when the cursor display is reenabled. The application program does not need to erase the cursor when calling screen driver functions.

#### *LINE AND COLUMN NUMBERING*

The screen driver uses two different numbering bases for line number and column number within function calls and escape sequences.

All the function calls (except Chput) which are used to pass an escape sequence consider the line number and column number as being numbered from 0.

All the VT-100 compatible escape sequences interpret the line number as being numbered from 1.

## SECTION 2

### TERMINAL EMULATION MODE

The PC-8401A screen driver emulates two popular terminals, the Digital VT-100 and the Soroc IQ-120. The SetMode function is used to select between the two emulations.

The functions in the driver are independent as to the terminal type selected, except for interpretation of control codes and escape sequences particular to either of the terminal emulations.

## SECTION 3

### CHARACTER DISPLAY FUNCTIONS

#### *CHPUT - DISPLAY A CHARACTER TO CONSOLE*

Name	Chput
Entry	C = 000H E = Character code
Exit	None

This function is used to display a character at the current cursor position. The character's ASCII code is passed through the E register.

The cursor is advanced to the next position. When it reaches the right margin of the current line, and if Auto Wraparound is enabled, the cursor moves to the left margin (column 0) of the next line. If Auto Wraparound is disabled and this occurs, the cursor remains at the right margin of the current line. If the Auto Wraparound is performed when the cursor is at the bottom margin, a Scroll Up is performed unless Screen Lock is in effect.

The control code processing depends on the terminal type selected. In the VT-100 mode, the control characters function as summarized below. Control characters other than those listed on the next page are simply ignored:

BEL (07H)	Generates an audible tone.
BS (08H)	Moves the cursor back one column unless the cursor is at the left margin in which case the Back Space code is ignored.
HT (09H)	Moves the cursor to next Horizontal Tab stop, or to the right margin if there are no more Tab stops.
LF (0AH)	When the Line Feed/New Line mode is reset, the cursor moves to next line at the same column. When that mode is set, the cursor moves to next line at the left margin. In both cases, when the cursor reaches the bottom margin, a Scroll Up is performed unless Screen Lock is selected.
VT (0BH)	Interpreted as a Line Feed.
FF (0CH)	When the Form Feed mode is set, the entire screen is cleared. This is independent of the currently set scrolling region. When the Form Feed mode is reset, a Form Feed is interpreted as a Line Feed.
CR (0DH)	Moves the cursor to the left margin of the current line.
CAN (018H)	Cancels a currently operating escape sequence if CAN is pressed during it's operation. Displayed as substitution character (crosshatch symbol.)
SUB (01AH)	Interpreted as a CAN.
ESC (01BH)	Interpreted as a VT-100 compatible escape sequence flag.

All control characters except CAN, SUB and ESC can interrupt an escape sequence. The control character is processed normally. Then, the escape sequence is resumed as if no control character had interrupted its operation.

The reception of an ESC character, during an escape sequence, cancels the current sequence and initializes a new escape sequence. The reception of a SUB or a CAN will cancel an operating escape sequence.

In the Soroc mode, a control character is interpreted as summarized on the next page:

BEL (07H)	Generates an audible tone.
BS (08H)	Moves the cursor back one column. If the cursor is at the left margin, the cursor moves to the far right margin of the preceding line. Unless the cursor is at the home position, in which case the character is ignored.
HT (09H)	Moves the cursor to the next Horizontal Tab stop, or to the right margin if there are no more Tab stops.
LF (0AH)	Moves the cursor to the left margin of the next line.
VT (0BH)	Moves the cursor up one line in the same column. If the cursor is at the top line of the screen, the character is ignored.
FF (0CH)	Moves the cursor right one column. If the cursor is at the right margin, the cursor moves to the left margin of the next line. If the cursor is at the right margin of the bottom of the screen, a Scroll Up is performed.
CR (0DH)	Moves the cursor to the left margin of the current line.
SYN (016H)	Moves the cursor down a line. If the cursor is at the bottom of the screen, the character is ignored.
SUB (01AH)	Erases the entire screen and moves the cursor to the home position of the screen.
RS (01EH)	Moves the cursor to the home position of the screen.
US (01FH)	Moves the cursor down one line. If the cursor is at the bottom margin, a Scroll Up is performed unless Screen Lock is in effect in which case the character is ignored.

The screen driver uses 8 bit character codes. The character codes are divided into 3 sets. Characters whose codes are between 000H and 01FH are members of the "C" (control character) set, codes between 020H and 07FH are in the "G0" (printable ASCII) set, and codes between 080H and 0FFH are in the "G1" (graphics) set. The actual character set from which a printable character's font is displayed is designated by the Select Character Set sequence or the Designate Character Set function.

#### *LITOUT - DISPLAY A CHARACTER LITERALLY*

Name	Litout
Entry	C = 001H E = Character code
Exit	None

This function is similar to the Chput function except that when a control character is passed to this routine, the associated function of the control character is not performed and the character is displayed literally with an associated special symbol.

#### *BITOUT - DISPLAY A CHARACTER WITH GIVEN BIT IMAGE*

Name	Bitout
Entry	C = 002H DE = Pointer to character font to be displayed.
Exit	None

The Bitout function displays a character from the font supplied by an application program. The character is displayed at the current cursor position, and the cursor position is advanced in the same manner as in the Chput function.

The font consists of 8 bytes for each character to be displayed. The 1st byte represents the bit image of the 1st row of the character, the 2nd byte for the 2nd row, and so on. Only the upper 6 bits of each byte are valid, the low 2 bits must be 0. The MSB, Bit 7 of each byte corresponds to the leftmost pixel of each row.

See section 14.1 for further details regarding bit images.

## SECTION 4

### CONSOLE SELECTION FUNCTIONS

#### *SELCON - SELECT CONSOLE*

Name	Selcon
Entry	C = 003H E = 0 selects the LCD as the current console. E = 1 selects the CRT as the current console.
Exit	Cy = 0 when successful. Cy = 1 when the CRT is requested but not available.

The PC-8401A supports two display devices, the LCD and the CRT. Only one display device can be assigned as console at any given time. All screen driver functions are directed to the display device which is selected as console device.

This function automatically invokes the Rescon function to reset the display status's initial values. See the description of the Rescon function for the initial settings.

E must contain either 0 or 1. The result when E is not 0 or 1 is undefined.

#### *ASKCON - RETURN CURRENT CONSOLE DEVICE*

Name	Askcon
Entry	C = 004H
Exit	A = 0 if the current console device is LCD. A = 1 if the current console device is CRT.  L = Physical height of the current console in lines.

The value returned is the physical height of the currently selected console, which includes the line used for softkey (function key) display. To get the number of lines an application program can use, subtract 1 if softkeys are to be used.

## SECTION 5

### ATTRIBUTES CONTROL FUNCTIONS

#### SETATR - SET CHARACTER ATTRIBUTES

Name      Setatr

Entry      C    = 005H  
             E    = Character attributes as encoded below:  
                  b0 = 0      turns off reversed video attributes.  
                  b0 = 1      turns on reversed video attributes.  
                  b1 == 0      reserved. Must be zero.  
                  b2 == 0      reserved. Must be zero.  
                  b3 == 0      reserved. Must be zero.  
                  b6 through b4 color code as defined below. Valid for CRT only.  
                  b7 == 0      reserved. Must be zero.

Note: "==" 0" denotes "Must be zero"

<i>Color Code</i>			<i>Color</i>
<i>b6</i>	<i>b5</i>	<i>b4</i>	
0	0	0	White
0	0	1	Cyan
0	1	0	Magenta
0	1	1	Blue
1	0	0	Yellow
1	0	1	Green
1	1	0	Red
1	1	1	Black

Exit      None

Characters passed to the Chput and Litout functions are displayed in the currently set attributes. The LCD supports reversed video attributes only. The CRT supports color attributes in addition to the reversed video attributes.

Note that erasing is always done using normal video attributes. Also, Scrolling Up or Down always clears the bottom or the top line in normal video attributes.



#### *ASKATR - RETURN CURRENT CHARACTER ATTRIBUTES*

Name      Askatr

Entry      C = 006H

Exit      A = Current character attributes as encoded below.

b0 = 0	reverse video attributes are reset.
b0 = 1	reverse video attributes are set.
b1	undefined.
b2	undefined.
b3	undefined.
b6 through b4	color code as defined in Setatr.
b7	undefined.

#### *REVCHR - REVERSE CHARACTERS ON SCREEN*

Name      Revchr

Entry      C = 007H  
            E = Number of characters to be reversed

Exit      None

This function is used to reverse (invert the color of) the specified number of characters that follow the cursor position including the cursor position.

When the given number is larger than the number of characters to the right of the cursor position, including the cursor position in the current line, all characters in the line from the cursor position on are reversed.

The current cursor position is preserved.

## SECTION 6

### SCREEN MODE CONTROL FUNCTIONS

#### SETMODE - SET SCREEN MODES

Name	Setmode	
Entry	C = 008H	
Exit	E = New screen mode as encoded below.	
	b0 = 1	enables auto wraparound.
	b0 = 0	disables auto wraparound.
	b1 = 1	sets screen lock.
	b1 = 0	resets screen lock.
	b2 = 1	sets form feed mode.
	b2 = 0	resets form feed mode.
	b3 = 1	enables softkey display.
	b3 = 0	disables softkey display.
	b4 = 1	sets line feed/new line mode.
	b4 = 0	resets line feed/new line mode.
	b5 == 0	Reserved for future use. Must be zero.
	b6 == 0	Reserved for future use. Must be zero.
	b7 = 1	selects Soroc IQ-120 emulation mode.
	b7 = 0	selects Digital VT-100 emulation mode.
Exit	None	

The wraparound mode determines how the cursor is advanced by the Chput and Litout functions when the cursor is at the right margin of line. When set, the cursor is moved to the next line at the left margin. When reset, the cursor remains at the right margin of the current line.

The screen lock mode controls the scrolling operation. When set, scrolling is disabled including implicit scrolling up by a Line Feed character when at the bottom line.

The form feed mode specifies how the Form Feed character is interpreted. When set, the Form Feed character causes the screen to be cleared. when reset, the Form Feed character is interpreted as a Line Feed.

The softkey display mode is used to display or inhibit the display of softkeys (function keys) on the bottom line of the screen. If the cursor is on the bottom line when the softkey display is turned on, a Scroll Up is performed. Otherwise, the contents of the bottom line are lost, and the softkeys are displayed over the bottom line. When the softkey display is turned off, the bottom line is cleared.

The line feed/new line mode specifies how the Line Feed character is interpreted. When set, the Line Feed character causes the cursor to move to the next line at the left margin. When reset, the Line Feed character causes the cursor to move the next line at same column position.

As mentioned before, the PC-8401A screen driver emulates two terminals; the Digital VT-100 and the Soroc IQ-120. The MSB of the mode byte passed to the Setmode function selects the terminal emulation.

Note that the character set in use is not automatically changed by the terminal selection. The character set selection is independent of the terminal selection, and designated by the Selset function.

## ASKMODE - RETURN CURRENT SCREEN MODE

Name        Askmode

Entry       C    = 009H

Exit        A    = Current screen mode as encoded below.

b0 = 1        when auto wraparound is enabled.

b0 = 0        when auto wraparound is disabled.

b1 = 1        when screen lock is set.

b1 = 0        when screen lock is reset.

b2 = 1        when form feed mode is set.

b2 = 0        when form feed mode is reset.

b3 = 1        when softkeys are displayed.

b3 = 0        when softkey are not displayed.

b4 = 1        when line feed/new line mode is set.

b4 = 0        when Line feed/new line mode is reset.

b5 =           always 0.

b6 =           always 0.

b7 =           1 when Soroc IQ-120 emulation is selected.

b7 =           0 when Digital VT-100 emulation is selected.

## SECTION 7

### CURSOR CONTROL FUNCTIONS

#### *CSRCTL - TURN CURSOR ON OR OFF.*

Name	Csrctl
Entry	C = 00AH E = 0 turns the cursor display off. E = 1 turns on the block cursor. E = 2 turns on the underline cursor.
Exit	None

#### *ASKCSR - RETURN CURRENT CURSOR DISPLAY STATUS*

Name	Askcsr
Entry	C = 00BH
Exit	A = 0 when the cursor is not displayed. A = 1 when the block cursor is displayed. A = 2 when the underline cursor is displayed.

## SECTION 8

### CURSOR POSITIONING FUNCTIONS

#### *LOCATE - LOCATE CURSOR*

Name	Locate
Entry	C = 00CH D = Column number E = Line number
Exit	None

The locate function moves the cursor to the specified position. Column and line numbers are relative to 0.

The line number value is interpreted depending on the current origin mode setting. When the origin mode is set, line number is relative to the top margin. Line 0 is the top margin. When the origin mode is reset, line number is relative to the top of screen, independent to the scrolling region. Line 0 is the top line of the screen.

When the origin mode is set, the cursor can never be located outside of the scrolling region. When the given line number is below the scrolling region's bottom margin, the line at the bottom margin is assumed.

When origin mode is reset, the cursor can be located outside the scrolling region. When the given line number is below the bottom of the screen, the screen's bottom line is assumed.

Specification of a column number larger than the right margin (=79) is interpreted as specifying the far right margin.

#### *GOUP - MOVE CURSOR UP*

Name	Goup
Entry	C = 00DH E = Number of lines to move the cursor up.
Exit	None

The Goup function moves the cursor up by the specified number of lines. The cursor stops at the top margin.

#### *GODOWN - MOVE CURSOR DOWN*

Name      Godown

Entry      C   = 00EH  
            E   = Number of lines to move the cursor down.

Exit        None

The cursor is moved the specified number of lines downward. The cursor stops at the bottom margin.

#### *GORIGHT - MOVE CURSOR RIGHT*

Name      Goright

Entry      C   = 00FH  
            E   = Number of columns to move the cursor to the right.

Exit        None

The cursor is moved the specified number of columns to the right. The cursor stops at the right margin.

#### *GOLEFT - MOVE CURSOR LEFT*

Name      Goleft

Entry      C   = 010H  
            E   = Number of columns to move the cursor to the left.

Exit        None

Cursor is moved the specified number of columns to the left. The cursor stops at the left margin.

#### *INDEX - INDEX*

Name      Index

Entry      C   = 011H

Exit        None

The index moves the cursor down one line at the same column. When the cursor is at the bottom margin, a Scroll Up is performed unless Screen Lock is set, in which case the cursor remains at the bottom margin. This function is equivalent to sending a Line Feed control character, except Index is independent of the currently set line feed/new line mode.

#### *RINDEX - REVERSE INDEX*

Name	Rindex
Entry	C = 012H
Exit	None

Rindex function moves the cursor one line up at the same column. When the cursor is at the top margin, a Scroll Down is performed unless Screen Lock is set, in which case the cursor remains at the top margin.

#### *ASKPOS - RETURN CURSOR POSITION*

Name	Askpos
Entry	C = 013H
Exit	H = Column number of the cursor position. L = Line number of the cursor position

The column number and line number are relative to 0.

The line number depends on the current origin mode setting. When the origin mode is reset, the line number returned is relative to the top margin. When reset, the line number is relative to the top of the screen.



## SECTION 9

### SCROLLING REGION CONTROL FUNCTIONS

#### *SETSCR - SET SCROLLING REGION*

Name        Setscr

Entry        C = 014H  
              B = 0 resets origin mode.  
              B = 1 sets origin mode.  
              D = line number of the top margin of scrolling region  
              E = line number of the bottom margin of scrolling region.

Exit         None

Setscr is used to specify the scrolling region and origin mode. The line numbers given in this function are always absolute and begin from 0; i.e., line 0 is always the top line of the screen regardless of the current origin mode.

If the top margin is below the bottom margin, the request is ignored, and the current scrolling region remains in effect.

If one or both of the given line numbers is below the screen's bottom, the bottom of the screen is assumed.

The origin mode determines how line numbers passed to the Locate function or returned by the Askpos function are interpreted. When set, the line number is relative to the top margin of the scrolling region. Line 0 is interpreted as the line at the top margin. When reset, absolute numbering is in effect. Line 0 is always the top line of the screen, independent of the scrolling region setting.

In addition to line numbering, the origin mode affects the ability to locate the cursor outside of the scrolling region. When set, the Locate function will not move the cursor outside the scrolling region. When reset, the Locate function can move the cursor anywhere on the screen, independent of the scrolling region.

When Setscr is called, if the origin mode is set, the cursor is moved to the origin of the new scrolling region or, if the origin mode is reset, the cursor is moved to the home position of the screen.

## ASKSCR - RETURN CURRENT SCROLLING REGION SETTING

Name      Askscr

Entry      C = 015H

Exit      A = Current origin mode as encoded below.  
            A = 0 when origin mode is reset.  
            A = 1 when origin mode is set.  
            H = Line number of the top margin  
            L = Line number of the bottom margin

The line numbers returned are independent of the origin mode, and the values are always absolute, with the lines numbered from 0.

## SECTION 10

### ERASING FUNCTIONS

#### *ERADSP - ERASE DISPLAY FUNCTION*

Name Eradsp

Entry C = 016H  
E = Function code as encoded below.

E = 0 Erase from the cursor position to end of screen.  
E = 1 Erase from the top of screen to the cursor position.  
E = 2 Erase entire screen.

Exit None

In all cases, the character at the cursor position is erased. The cursor position however, is preserved.

The erasing is done in normal video attributes, regardless of the current attribute selection.

Erasing is independent of the scrolling region. Erasing always involves the entire screen.

#### *ERALIN - ERASE LINE FUNCTION*

Name Eralin

Entry C = 017H  
E = Function code as encoded below.  
E = 0 Erase from the cursor to the end of the line.  
E = 1 Erase from the top of the current line to the cursor position.  
E = 2 Erase the entire current line.

Exit None

In all cases, the character at the cursor position is erased. The cursor position is preserved.

Erasing is made in normal video attributes, regardless of the current attributes selected.

## SECTION 11

### CHARACTER SET CONTROL FUNCTIONS

#### *SELSET0 - SELECT G0 CHARACTER SET*

Name Selset0

Entry C = 05CH  
E = G0 character set as encoded below.

E = 0 US ASCII set  
E = 1 United Kingdom set  
E = 2 German set  
E = 3 PC-8800A alternate character set  
E = 4 VT-100 line drawing character set

Exit None

The SelSet0 function designates one of 5 character sets as the G0 set. The G1 set selection remains unchanged.

#### *SELSET1 - SELECT G1 CHARACTER SET*

Name Selset1

Entry C = 01CH  
E = G1 character set as encoded below.

E = 0 US ASCII set  
E = 1 United Kingdom set  
E = 2 German set  
E = 3 PC-8800A alternate character set  
E = 4 VT-100 line drawing character set

Exit None

The SetSet1 function designates one of 5 character sets as the G1 set. The G0 set selection remains unchanged.

## *ASKSET - RETURN CURRENT CHARACTER SET*

Name      Askset

Entry      C = 01DH

Exit      H = Current G0 character set as encoded below.

H = 0 US ASCII set  
H = 1 United Kingdom set  
H = 2 German set  
H = 3 PC-8800A alternate character set  
H = 4 VT-100 line drawing character set

L = Current G1 character set as encoded below.

L = 0 US ASCII set  
L = 1 United Kingdom set  
L = 2 German set  
L = 3 PC-8800A alternate character set  
L = 4 VT-100 line drawing character set

The AskSet function returns the character set encodes currently assigned to the G0 and G1 sets.

## SECTION 12

### TABULATION CONTROL FUNCTIONS

#### *TABSET - SET TAB*

Name	Tabset
Entry	C = 01EH
Exit	None

The Tabset function is equivalent to the Horizontal Tabulation Set escape sequence and sets a tab stop at the cursor position.

#### *RESTAB - RESET TAB*

Name	Restab
Entry	C = 01FH E = 0 Clears a tab stop at the cursor position. E = 1 Clears all the tab stops.
Exit	None

## SECTION 13

### RESET FUNCTION

#### *RESSCN - RESET SCREEN*

Name	Resscn
Entry	C = 020H
Exit	None

The Resscn function resets screen driver status to initial state as below.

1. Complete screen becomes the scrolling region.
2. Origin mode is reset.
3. Cursor position moves to the home position of the screen.
4. Auto wraparound mode is set.
5. Form Feed mode is reset.
6. Screen lock is reset.
7. The softkeys are erased.
8. New line mode is reset.
9. Normal video attributes are selected.
10. The block cursor is displayed.
11. The U.S. ASCII character set (U.S. Version), United Kingdom set (U.K. version) or German character set (German version) is designated as the G0 set. The G1 set becomes the VT-100 compatible line drawing set if terminal emulation mode is VT-100, or The PC-8801A compatible alternate character set if terminal emulation mode is Soroc IQ-120.
12. Tab stops are set at every 8th column.
13. Complete screen is cleared.

Note that terminal emulation mode is not affected by this function.

## SECTION 14

### LOW LEVEL VRAM ACCESS FUNCTIONS

#### READPIX - READ PIXELS OF A LINE

Name        Readpix

Entry        C = 021H  
               DE = Address in memory where pixels are passed.

Exit         None

The Readpix function reads the pixels in VRAM (Video/LCD RAM) at the current line into a user supplied buffer. Each line consists of 8 rasters, and each raster holds 80 bytes. Therefore the size of the buffer must be 640 bytes.

Only the high 6 bits in each byte passed into buffer are valid. The other two data bits are not necessarily zeroed and may contain random data. The high 6 bits of the 1st byte represent pixels at X positions 000H through 005H. The high 6 bits of the 2nd byte represent pixels at X position 006H through 00BH, and so on.

For example, a Readpix dump to memory location 200H of a screen containing an asterisk at the home position and the number one immediately to it's right on an otherwise empty screen, will contain:

```
Dot line 1  0200:  20  12  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              0210: zeros (no other characters on screen), up to:
Dot line 2  0250:  AA  31  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              0260: ...zeros, etc., up to:
Dot line 3  02A0:  70  70  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              02B0: ...zeros, etc., up to:
Dot line 4  02F0:  70  32  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              0300: ...zeros, etc., up to:
Dot line 5  0340:  73  33  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              0350: ...zeros, etc., up to:
Dot line 6  0390:  AA  30  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              03A0: ...zeros, etc., up to:
Dot line 7  03E0:  21  79  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              03F0: ...zeros, etc., up to:
Dot line 8  0430:  00  03  00  00  00  00  00  00  00  00  00  00  00  00  00  00
              0440: ...zeros, etc., up to 47FH, which is the last line of this 640 byte buffer.
```



After performing the Boolean operation AND 0FCH to get the six significant bits of the characters represented on each dot line we get:  
(For the \* and 1 characters described:)

<i>Y Position</i>		<i>X Position</i>												
Dot line 1	20 10 >	expresses the pattern	>	.	.	*	.	.	.	.	.	.	*	.
Dot line 2	A2 30 >		>	*	.	*	.	*	.	.	.	*	*	.
Dot line 3	70 70 >		>	.	*	*	*	.	.	.	*	*	*	.
Dot line 4	70 30 >		>	.	*	*	*	.	.	.	*	*	*	.
Dot line 5	70 30 >		>	.	*	*	*	.	.	.	*	*	*	.
Dot line 6	A2 30 >		>	*	.	*	.	*	.	.	*	*	*	.
Dot line 7	20 78 >		>	.	.	*	.	.	.	.	*	*	*	*
Dot line 8	00 00 >		>	.	.	.	.	.	.	.	.	.	.	.

Here the asterisk and the one can be clearly seen as reconstructed from the bit image of the 1st eight lines.

#### WRTPIX - WRITE PIXELS OF A LINE

Name      Wrtpix

Entry      C = 022H  
DE = Address in memory where pixels to be written are stored.

Exit        None

The Wrtpix function writes the pixels prepared in a user supplied buffer into VRAM on the current line. The format of the buffer is the same as described for Readpix.

## SECTION 15

### GRAPHICS FUNCTIONS

#### *MOVE - MOVE GRAPHICS CURSOR POSITION*

Name        Move

Entry        C = 025H  
              DE = Y position of the new graphics point  
              HL = X position of the new graphics point

Exit         None

Locates the current graphic point to the (x,y) coordinate specified.

#### *PSET - PLOT A PIXEL*

Name        Pset

Entry        C = 026H

Exit         None

Sets a graphic dot at the current graphic point in the currently set graphics color (see Setcol.) This will turn the dot on if color is 1 or turn it off if the color is 0. This point should be set prior to issuing this call by the Move routine.

#### *LINE - DRAW A LINE*

Name        Line

Entry        C = 027H  
              DE = Y position of the ending point  
              HL = X position of the ending point

Exit         None

Draws a line from the current graphic point to the (x,y) coordinate specified by this call in the currently set graphics color (see Setcol.) This will turn the dots in the line on if color is 1 or turn them off if the color is 0. The current graphic point is updated to the (x,y) coordinate specified by this call upon exit.

### *BOX - DRAW A RECTANGLE*

Name      Box

Entry      C    = 028H  
            DE = Y position of the diagonal point  
            HL = X position of the diagonal point

Exit        None

Draws a rectangle from the current graphic point to the (x,y) coordinate specified by this call in the currently set graphics color (see Setcol.) This will turn the dots in the rectangle's lines on if color is 1 or turn them off if the color is 0. The current graphic point is updated to the (x,y) coordinate specified by this call upon exit.

### *BOXFILL - DRAW A RECTANGLE AND FILL INSIDE*

Name      Boxfill

Entry      C    = 029H  
            B    = Fill mode as encoded below  
                  B = 0 Draw in the current graphics color.  
                  B = 1 Draw XORing (reversing) the screen contents.  
            DE = Y position of the diagonal point  
            HL = X position of the diagonal point

Exit        None

Draws a filled rectangle from the current graphic point to the (x,y) coordinate specified by this call. If XORing is selected, all screen dots intercepted by the rectangle will be inverted. If drawing in the current graphics color is selected, this will turn the dots in the rectangle on if color is 1 or turn them off if the color is 0. The current graphic point is updated to the (x,y) coordinate specified by this call upon exit.

### *SETCOL - SET GRAPHICS COLOR*

Name        Setcol

Entry        C = 02AH  
              E = 0 sets graphics color to 0.  
              E = 1 sets graphics color to 1.

Exit         None

### *ASKGCP - RETURN GRAPHICS CURRENT POINT*

Name        Askgcp

Entry        C = 02BH

Exit         HL = X position of the current graphics point  
              DE = Y position of the current graphics point

### *ASKCOL - RETURN CURRENT GRAPHICS COLOR*

Name        Askcol

Entry        C = 02CH

Exit         A = Current graphics color, 0 or 1.

### *POINT - RETURN PIXEL VALUE*

Name        Point

Entry        C = 02DH

Exit         A = Set/Reset value of the pixel (0 or 1) at the current graphics point.

## SECTION 16

### VT-100 COMPATIBLE ESCAPE SEQUENCES

All sequences begin with ESC, and end with an ASCII character. ASCII characters are used for the arguments, which may occur between the ESC character and the ending ASCII character. For example to move the cursor up 10 lines, use the CUU sequence:

ESC [ 10 A

In hexadecimal: 01BH 05BH 031H 030H 041H

In BASIC: PRINT CHR\$(27) "[10A";

#### *CURSOR POSITIONING SEQUENCES*

##### CUU - Cursor Up Sequence

Name        CUU

Sequence   ESC [ Pn A

Default     Pn = 1

Moves the cursor up Pn lines at same column. The cursor stops at the top margin.

##### CUD - Cursor Down Sequence

Name        CUD

Sequence   ESC [ Pn B

Default     Pn = 1

Moves the cursor down Pn lines at same column. The cursor stops at the bottom margin.

##### CUF - Cursor Forward Sequence

Name        CUF

Sequence   ESC [ Pn C

Default     Pn = 1

Moves the cursor right Pn columns in the current line. The cursor stops at the right margin.

#### CUB - Cursor Backward Sequence

Name CUB

Sequence ESC [ Pn D

Default Pn = 1

Moves the cursor left Pn columns in the current line. The cursor stops at the left margin.

#### CUP - Cursor Position

Name CUP

Sequence ESC [ P1 ; Pc H

Default P1 = 1  
Pc = 1

Moves the cursor to line P1, column Pc. If either P1 or Pc are not specified or are specified as 0, the cursor moves to the first line or the first column.

The numbering of lines and the ability to move the cursor beyond the current scrolling margins depends on the origin mode selection.

#### HVP - Horizontal And Vertical Position

Name HVP

Sequence ESC [ P1 ; Pc f

Default P1 = 1  
Pc = 1

This sequence operates the same as the cursor position (CUP).

#### IND - Index

Name IND

Sequence ESC D

Moves the cursor down one line in the same column. If the cursor is at the bottom margin, a Scroll Up is performed unless screen lock mode is set. In that case the sequence is ignored.

#### RI - Reverse Index

Name RI

Sequence ESC M

Moves the cursor up one line in the same column. If the cursor is at the top margin, a Scroll Down is performed unless Screen Lock is set in which case the sequence is ignored.

#### NEL - Next Line

Name NEL

Sequence ESC E

Moves the cursor to the first column on the next line. If the cursor is at the bottom margin, a Scroll Up is performed unless Screen Lock is set in which case the cursor is moved to first column on the bottom line. A Scroll Up is not performed.

#### SC - Save Cursor

Name SC

Sequence ESC 7

Save the cursor position, the graphics rendition (attributes) and the character set selection. The Last values saved are lost.

#### RC - Restore Cursor

Name RC

Sequence ESC 8

Restores the previously saved cursor position, graphics rendition and character set selections. If none were saved previously, the cursor simply moves to the screen home position.

## ERASING SEQUENCES

### EL - Erase In Line

Name        EL  
Sequence    ESC [ Ps K  
Default     Ps = 0

When Ps = 0 the characters are erased from the cursor to the end of the line. When Ps = 1 characters from the beginning of the current line to the cursor position are erased. When Ps = 2 the entire line is erased.

In all cases, the character at the cursor position is erased, but the cursor position is preserved.

Like the Eralin BIOS function, erasing is made in the normal video attributes, regardless of the current attributes selection.

### ED - Erase In Display

Name        ED  
Sequence    ESC [ Ps J  
Default     Ps = 0

When Ps = 0 the characters are erased from the cursor position to the end of the screen. When Ps = 1 characters from the beginning of the screen to the cursor position are erased. When Ps = 2 the entire display is erased.

In all cases, the character at the cursor position is erased, but the cursor position is preserved.

Like the Eradsp BIOS function, erasing is made in the normal video attributes, regardless of the current attributes, or scrolling region selections.



## *TABULATION CONTROL SEQUENCES*

### Horizontal Tabulation Set

Name        HTS

Sequence    ESC H

Sets a horizontal tab stop at the current cursor position.

### Tabulation Clear

Name        TBC

Sequence    ESC [ Ps g

Default     Ps = 0

When Ps = 0 a horizontal tab stop at the current cursor position is cleared. When Ps = 3 all horizontal tab stops are cleared.

## CHARACTER SET SELECTION SEQUENCES

### SCS - Select Character Set Sequence

The G0 and G1 character sets are designated from one of the five possible character sets. The G0 and G1 sets are invoked automatically by the character codes passed to the screen driver. The characters whose codes range from 020H to 07FH invoke the G0 set, while characters whose codes range from 080H to 0FFH invoke the G1 set.

<i>G0 Set Sequence</i>	<i>G1 Set Sequence</i>	<i>Designated Set</i>
ESC ( A	ESC ) A	United Kingdom set
ESC ( B	ESC ) B	US ASCII set
ESC ( 0	ESC ) 0	VT-100 graphics
ESC ( 1	ESC ) 1	Germany set
ESC ( 2	ESC ) 2	8800 graphics

## GRAPHICS RENDITION CONTROL SEQUENCES

### SGR - Select Graphics Rendition

Name        SGR

Sequence    ESC [ Ps ; ... ; Ps m

Default     Ps = 0

<i>Ps</i>	<i>Function</i>
0	Turns off all graphics rendition (character attributes.)
4	Selects reversed video attributes.
7	Selects reversed video attributes.
30	Set color to black.
31	Set color to red.
32	Set color to green.
33	Set color to yellow.
34	Set color to blue.
35	Set color to magenta.
36	Set color to cyan.
37	Set color to white.

Color selection (Ps = 30 through Ps = 37) is valid only when the CRT is selected as the current console.

## SM AND RM - SET MODE AND RESET MODE SEQUENCES

### Line Feed/New Line

This mode selects how the Carriage Return and Line Feed are interpreted.

Set        ESC [ 2 0 h

Causes the screen driver to move the cursor to first column of the next line when a Line Feed is received, and causes the keyboard driver to generate a Carriage Return Line Feed sequence when the RETURN key is pressed.

Reset     ESC [ 2 0 l

Causes the screen driver to move the cursor to the next line while remaining in the same column position when a line feed is received, and causes the keyboard driver to only generate a Carriage Return when the RETURN key is pressed.

### Cursor Key Character Selection

This sequence selects the set of characters generated by the keyboard driver when the cursor keys (arrow keys) are pressed.

Set        ESC [ ? 1 h

Selects the cursor keys to generate application control functions.

Reset     ESC [ ? 1 l

Selects the cursor keys to generate ANSI cursor control sequences.

Cursor keycodes in each mode are shown below:

<i>Cursor</i>	<i>Reset</i> <i>(Cursor control)</i>	<i>Set</i> <i>(Application)</i>
up	ESC [ A	ESC O A
down	ESC [ B	ESC O B
right	ESC [ C	ESC O C
left	ESC [ D	ESC O D

## Terminal Emulation Mode

Reset      ESC [ ? 2 1

Exits from the VT-100 mode and enters the Soroc IQ-120 emulation mode.

A VT-100 system would use this sequence to select the VT-52 emulation mode. Since the PC-8401A does not support VT-52 emulation, the Soroc IQ-120 is the alternate emulation mode selected by issuing this sequence.

## Origin

Set        ESC [ ? 6 h

Sets the origin to line 1, column 1 of the currently set scrolling region. Line numbers are relative to the start of the scrolling region. The cursor moves to the origin.

Reset      ESC [ ? 6 l

Sets the origin to be the upper-left corner of the screen (home.) Line numbers are independent of the scrolling region (absolute.) The cursor moves to the home position.

## Auto Wraparound

Set        ESC [ ? 7 h

Enables auto wraparound. When a character is displayed at the right margin, the cursor is moved to the first column in the next line.

Reset      ESC [ ? 7 l

Disables auto wraparound. When a character is displayed at the right margin, the cursor position remains.

#### Auto Key Repeat

Set        ESC [ ? 8 h

Enables key auto repeat. Key repeating is not performed for the ESC, and RETURN keys, nor with CTRL pressed with any key.

Reset     ESC [ ? 8 I

Disables key auto repeat.

#### Softkey Display

Set        ESC [ > 1 h

Displays the softkeys (function keys) on the bottom line of the screen.

Reset     ESC [ > 1 I

Softkey display is disabled.

The two sequences above are not found on a VT-100 terminal, and are private to the PC-8401A.

#### Cursor Display

Set        ESC [ > 5 h

Turn the cursor display on. The cursor style is determined by the last Cursor Style sequence or the equivalent BIOS function.

Reset     ESC [ > 5 I

Disable the cursor display.

The two sequences above are not found on a VT-100 terminal, and are unique to the PC-8401A.

## Cursor Style

Set           ESC [ > 6 h

Selects the underline cursor. The cursor is not automatically turned on by this sequence.

Reset        ESC [ > 6 l

Selects the block cursor. Again, the cursor is not automatically turned on by this sequence.

The two sequences above are not found on a VT-100 terminal, and are private to the PC-8401A.

## *SCROLLING REGION SEQUENCE*

Name        STBM

Sequence   ESC [ Pt ; Pb r

Default     Pt = 1  
            Pb = Screen bottom

The STBM sequence sets the scrolling region with the top margin at the line specified by Pt, and the bottom margin at the line specified by Pb.

When both parameters are omitted, the complete screen is set as the scrolling region. Pt and Pb are always absolute line numbers. (The top line of the screen is 1.)

The cursor moves to the origin of the screen, depending on the origin mode.

## DSR - Device Status Report

Name        DSR

Sequence    ESC [ 6 n

Return the current cursor position by giving a Cursor Position Report (CPR) sequence.

The format of the CPR is shown below:

ESC [ P1 ; Pc R

where P1 and Pc are the current line number and column number respectively.

When the origin mode is set, the line number is relative to the top margin of the scrolling region. The top margin is line 1.

When the origin mode is reset, the line number is absolute. The top of the screen is line 1.

*RIS - RESET TO INITIAL STATUS*

Name RIS

Sequence ESC c

RIS resets the cursor key mode and origin mode, and selects the entire screen as the scrolling region.



## *INCOMPATIBILITY WITH VT-100*

The following features found on a VT-100 are not supported by the PC-8401A:

1. Scrolling mode  
The PC-8401A does not support smooth scrolling.
2. Keypad character mode  
Since the PC-8401A does not have an auxiliary keypad, the keypad character mode is not supported.
3. ENQ control character
4. SI and SO control characters  
The screen driver of the PC-8401A uses 8 bit codes, while the VT-100 uses 7 bit codes. The PC-8401A selects display of a G0 or a G1 set character by the ASCII code's 8th bit status, so SI/SO for changing between character sets is not supported.
5. Screen Background mode (ESC [ ? 5 h and ESC [ ? 5 l).
6. VT-52 mode  
PC-8401A only supports the ANSI compatible sequences.
7. 132 column mode
8. Interlace mode.
9. Single Shift 2
10. Single Shift 3
11. Double height characters
12. Single height characters
13. Some DSR sequences  
PC-8401A only supports the DSR sequence where PN = 6.
14. DA (Device Attributes) sequences
15. Identify Terminal sequence
16. All the LED control sequences
17. All the test and adjustment sequences

## SECTION 17

### SOROC COMPATIBLE ESCAPE SEQUENCES

#### *KEYBOARD RELATED SEQUENCES*

Unlock Keyboard

Sequence ESC "

Unlock keyboard. Keyboard input is enabled. This is the default state of the keyboard.

Lock Keyboard

Sequence ESC #

Lock keyboard. Keyboard input is disabled until the Unlock Keyboard sequence is received.

#### *ERASING SEQUENCES*

Erase Entire Screen

Sequence ESC \*

Erases the entire screen in the normal video attributes. The cursor position remains unchanged.

Erase To End Of Screen

Sequence ESC Y

Erase to the end of the screen in the normal video attributes. The cursor position remains unchanged.

Erase To End Of Line

Sequence ESC T

Erase to the end of the line in normal video attributes. The cursor position remains unchanged.

## *ATTRIBUTE CONTROL SEQUENCES*

The two attribute control sequences below are not Soroc sequences. A Soroc terminal uses these sequences for field protection control. The PC-8401A does not support protected fields.

### Select Normal Video Attributes

Sequence ESC (

Select normal video attributes.

### Selects Reversed Video Attributes

Sequence ESC )

Selects reverse video attributes. Successive characters are displayed in reverse video until the Select Normal Video Attributes sequence is received.

## CURSOR POSITIONING SEQUENCES

Locate The cursor

Sequence ESC = Row Column

Locate the cursor at the specified position. The character encodes for the row and column positioning are given by specifying a single character for the row and one for the column. The chart below shows the position and the associated character:

<i>Position</i>	<i>Character</i>	<i>Position</i>	<i>Character</i>
1	Space	40	G
2	!	41	H
3	"	42	I
4	#	43	J
5	\$	44	K
6	%	45	L
7	&	46	M
8	'	47	N
9	(	48	O
10	)	49	P
11	*	50	Q
12	+	51	R
13	,	52	S
14	-	53	T
15	.	54	U
16	/	55	V
17	0	56	W
18	1	57	X
19	2	58	Y
20	3	59	Z
21	4	60	[
22	5	61	\
23	6	62	]
24	7	63	^
25	8	64	_
26	9	65	`
27	:	66	a
28	;	67	b
29	<	68	c
30	=	69	d

<i>Position</i>	<i>Character</i>	<i>Position</i>	<i>Character</i>
31	>	70	e
32	?	71	f
33	@	72	g
34	A	73	h
35	B	74	i
36	C	75	j
37	D	76	k
38	E	77	l
39	F	78	m
		79	n
		80	o

## Reverse Index

Sequence ESC j

Performs a reverse index. This is not a Soroc sequence, but is compatible with Televideo 950 terminal emulation.

## Report The Cursor Position

Sequence ESC ?

This sequence generates a two character string which represents the current position, followed by a Carriage Return code (00DH.) The 1st character represents the row position, and the 2nd character represents the column position. Each character is encoded as defined in the Locate sequence.

This is not a Soroc sequence, but is compatible with Televideo 950 terminal emulation.

## *INSERTION AND DELETION SEQUENCES*

The insertion and deletion sequences below are not Soroc sequences, but are compatible with Televideo 950 terminal emulation.

### Insert Line

Sequence ESC E

Insert a blank line in normal video attributes immediately above the current line. The bottom line on the screen is lost. The cursor position remains unchanged.

### Delete Line

Sequences ESC R

Delete the current line. Lines below the current line are moved up a line. A blank line in normal video attributes is inserted at the bottom line. The cursor position remains unchanged.

## *PRINTER RELATED SEQUENCES*

### Turn Printer Echo On

Sequence ESC @

Turns the printer echo on. All successive characters are directed to printer as well as display. Note that escape sequences are also echoed to printer.

Printer echo is in effect only in the Soroc mode. It is automatically disabled when the terminal type is changed to VT-100.

### Turn Printer Echo Off

Sequence ESC A

Turns the printer echo off. The successive characters are directed only to the display.

## *SOFTKEY CONTROL SEQUENCES*

### Turn Off Softkey Display

Sequence ESC g

Turns the softkey (function key) display off, and clears the bottom line. This is not a Soroc sequence.

### Turn On Softkey Display

Sequence ESC h

Turns the softkey display at the bottom line of the screen on. If the cursor is on the bottom line of the screen, a Scroll Up is done and the cursor moves one line up at same column position. This is not a Soroc sequence.

## *RETURN TO VT-100 MODE*

Sequence ESC V

Exit from the Soroc mode and return to the VT-100 mode.

### *INCOMPATIBILITY WITH SOROC IQ-120*

The following features found in a Soroc IQ-120 terminal are not supported by the PC-8401A Soroc emulation.

1. Block mode
2. Protect mode
3. The Escape sequences listed below.
  - a. ESC &      Set Protect Mode
  - b. ESC '      Reset Protect Mode
  - c. ESC 4      Send Line Unprotected
  - d. ESC 5      Send Page Unprotected
  - e. ESC +      Clear Unprotected Display to Nulls

Soroc IQ-120 uses ESC ) and ESC ( for protected field control. These two sequences are used for character attribute control in the PC-8401A implementation.

### *NOTES ON TELEVIDEO 950*

The Televideo 950 terminal supports a superset of the Soroc IQ-120 sequences. Some of the Televideo 950 sequences are supported in the Soroc emulation mode of the PC-8401A screen driver. These are listed below:

1. ESC E      Insert line
2. ESC R      Delete line
3. ESC j      Reverse index
4. ESC ?      Report the cursor position



# **CHAPTER 9**

## **Initialization of Screen and Keyboard Options in the PC-8401A**

NEC Corporation  
Copyright © 1984 by NEC Corporation  
All Rights Reserved

The PC-8401A BIOS has many operating variables. Character set selection, softkey displayed on/off, terminal emulation (VT100 or SOROC), to list a few.

When an application program is invoked through the CCP, some modes are initialized to default values, and some are left as they were previously set.

Listed below are modes which are set to default values by the CCP:

<i>Mode</i>	<i>Value</i>
Auto wraparound	On
Screen lock	Off
Formfeed	Off
Linefeed/newline	Off
Scrolling region	Complete screen
Softkey expansion	On
Stopkey code	3

Other modes are handled in a different way. When control is returned to the CCP, it saves the current settings of the modes, and restores them when invoking an application program. These modes are listed below:

*Mode*

- Softkey display
- Terminal tape
- Character set
- Softkey contents
- Delete key code
- Cursor key mode
- Cursor key code

Each application program must re-initialize them when needed. Some modes can be intentionally left uninitialized. For instance, an application program which does not use softkeys can leave the softkey contents uninitialized. In this way, the user can assign frequently used strings to the softkeys and use them in the application program, instead of having to type the strings manually.

# CHAPTER 10

## **Specifications of POWER OFF and SLEEP**

NEC Corporation  
Copyright © 1984 by NEC Corporation  
All Rights Reserved

## SECTION 1

### POWER OFF AND SLEEP

#### *AUTO POWER OFF*

This function sets the timer and forces the system to enter the Auto Power Off Mode. Under this mode, the timer causes a count down to occur when there are no key depressions. Typing a key, while under this mode, resets the timer to the initial count value. If there are no keys depressed, the timer counts down to zero. When that point is reached, the current status of all registers and the program counter are saved and the PC-8401A turns itself off.

Name	Autpoff
Entry	C = 057H E = Number of minutes to be set into the counter 00H disables Auto Power Off mode
Exit	None

If this call is entered with a zero placed in register E, the Auto Power Off mode is cancelled. Also, a power off will not occur if the CRT is assigned as the console device.

#### *ENTER SLEEP MODE*

This function makes the PC-8401A enter the Sleep Mode.

Name	Slepmod
Entry	C = 058H
Exit	Cy = Set if the wake-up time is undefined.

If the wake-up time is not set, the PC-8401A does not enter the Sleep Mode.

This function sets the wake-up time.

Entry C = 059H  
DE = Wake-up time to be set:  
E = bit0-bit3 minute (lower digit) 0-9  
bit4-bit7 (higher digit) 0-5  
D = bit0-bit3 hour (lower digit) 0-9  
bit0-bit2 (higher digit) 0-2

If an invalid parameter is given, the wake-up time is left unchanged.

Name	Askwktim
------	----------

Exit      Cy = Set if the wake-up time has not been set.  
           Cy = Reset if the wake-up time has been read.

10-3

