CHAPTER 15 VIRTUAL SCREEN

15.1 General

Note: This chapter contains descriptions related to the display controller for external display. This hardware is not available in every country. In countries where the display controller is not available, this chapter should be ignoring all references to the display controller and the external display.

The virtual screen is intended to allow the HX-20 to use a larger screen than the physical screen size of its LCD (20 columns by 4 lines). This function is good for both the LCD and the display controller (for external display).
The virtual screen has a maximum size of 255 columns by 255 lines. The display area where characters actually appear is called a "window". (The size of this window become 32 columns by 16 lines with the display controller.) It functions as a viewing window through which any part of the large internal screen can be seen. The virtual screen on the LCD is controlled by the master MCU, whereas that on the external display is controlled by the display controller via a high-speed serial communication interface.

15.2 Names and Technical Terms

(1) Virtual screen and physical screen
Only character (or text) information is handled by the virtual screen. Its maximum size is 255 columns by 255 lines. For the LCD, a screen image is produced on the MCU memory. As opposed to the virtual screen, the screen used for actual display is called a "physical screen".
The size of the physical screen is 20 columns by 4 lines for the LCD display and 32 columns by 16 lines for the display controller. Graphic display (straight line, etc.) is applicable to the physical screen only.

(2) Window
The window is a portion of the virtual screen that is actually displayed for viewing. The contents of the window are the same as those of the physical screen.

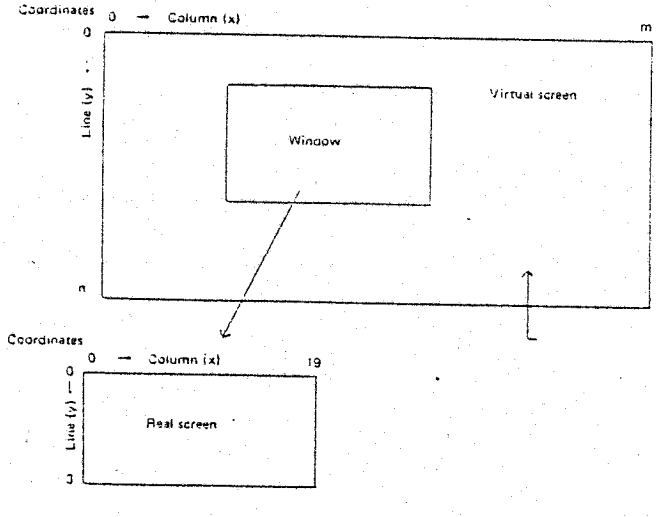


Fig. 15-1 Virtual Screen, Physical Screen and Window

(3) Coordinates on the screen
The screen in a horizontal direction is called "columns", while that in a vertical direction is called "lines". Coordinates are represented by x and y, which correspond to columns and lines, respectively. Column 0 indicates the left end of the screen, while line 0 indicates the top end of the screen. When the screen size is (m,n), the upper left end of the screen is identified by coordinates (0,0) and the lower right end by coordinates (m-1,n-1).

(4) Scroll
The scroll refers to the movement of the contents of the window up by one line. (Namely, the contents of the 4th line appear in the 3rd line, the contents of the 3rd line in the 2nd line, and the contents of the 2nd line in the 1st line. New data appears in the 4th line. In the HX-20, this function is also applicable to the movement of the screen in the upward, left and right directions.

(5) Scroll step
A character code to specify the number of scroll steps. When this code is accepted, the screen scrolls by the number of columns or lines specified by this code.

(6) Scroll of virtual screen
The scroll of virtual screen refers to the movement of the contents of the virtual screen up or down by one line.

(7) Line status
In some cases, two lines of data to be displayed are desired to be handled as a single line. To support this, a flag is provided to indicate a continuation line for each line. This flag is called a "line status flag" (see Fig. 9-2). The line status has a value "FF" if the line is a continuation of the preceding line and a value "00" if the line is a new line.

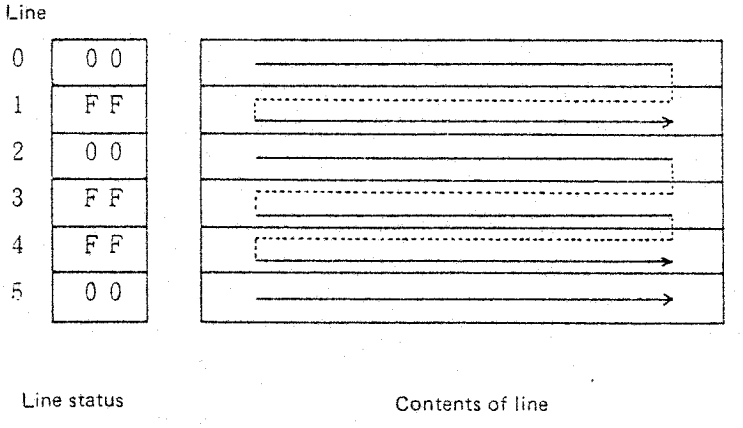


Fig. 15-2 Line Status

In Fig. 9-2, 0th and 1st lines, 2nd through 4th lines, and 5th line are logical single lines, respectively. The conditions for composing a logical single line are detailed in Section 9.4, Virtual Screen Functions.

(8) Cursor and cursor margin

The cursor indicates the position of a character to be displayed. At the same time, it also serves as a reference point for screen control.

The cursor is designed to always stay within the window. If the cursor moves out of the window, the window also moves with the cursor. When the cursor is at either end of the window, the next character cannot be identified. Therefore, a certain width from either end of the window must be predetermined so that the window moves when the cursor reaches this position. This width is called a "cursor margin".

In the following example, the screen size of 40 columns by 8 lines has been defined for LCD display. Assume that the cursor margin is set to a value of 3 and the position of the right margin is "RM", while that of the left margin is "LM". When the cursor is between the positions "LM" and "RM" (i.e., the shaded section in Fig. 9-3), window movement will not take place. When the cursor moves and reaches position "RM" (3rd column from the right), the cursor will not advance; instead the window will move to the right even if an attempt is made to move the cursor. This movement of the window stops when it reaches the right end of the virtual screen. From this point, the cursor moves again.

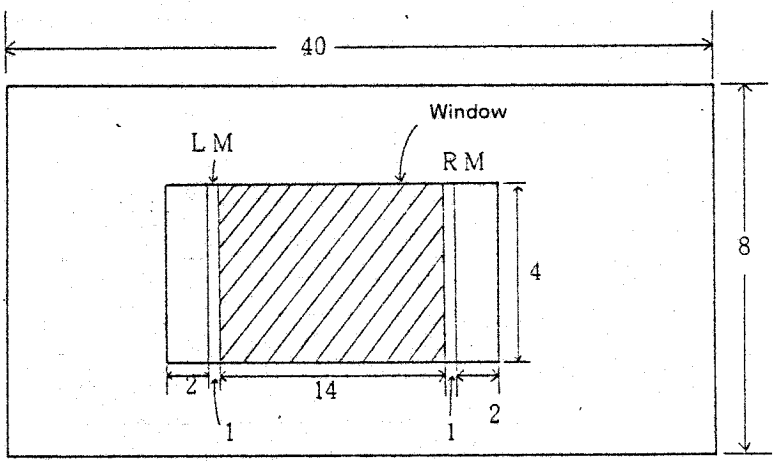


Fig. 15-3 Cursor Margin and Window Movement

The cursor margin may be specified by a value in the range of 1 to 10. If the value is 1, it indicates that no cursor margin is specified.

(9) List flag

If the window moves so that it contains the cursor, the displayed data is difficult to read. In some cases, the window may be desired to be fixed at the left end of the virtual screen (e.g., LIST command in BASIC.) The list flag controls the movement of the window. When the list flag is set, the window moves along the left end of the virtual screen (see the shaded section in Fig. 9-4).

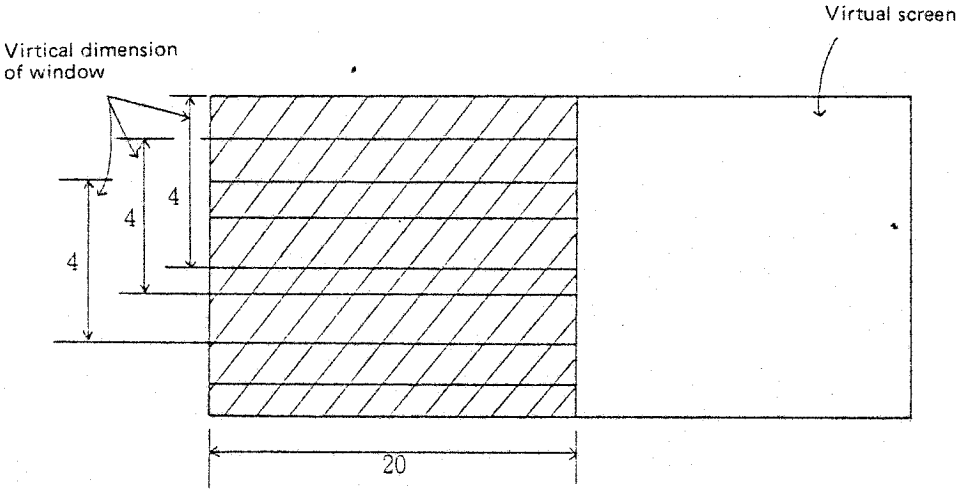


Fig. 15-4 Moving Range of Window when List Flag is ON.

When the list flag is ON, the window cannot move horizontally. However, its vertical movement is not restricted.

(10) Access pointer

When a character is to be input or output to or from the display controller, the location (i.e., coordinates on the virtual screen) where the character may be accessed for input/output must be specified. The access pointer functions to indicate this location.

15.3 Graphic Display

Only character codes may be handled by the virtual screen. It cannot handle graphic data. Graphic data processing is supported by both the LCD and display controller but in a manner different supported by both the LCD and display controller but in a manner different from each other.

(1) LCD

Graphic data is processed only on the physical screen. Display functions such as dot ON/OFF, straight line drawing, etc., are controlled directly against the controller. Therefore, the contents of the virtual screen will not be lost even if the graphic display is activated.

(2) Display controller

On the display controller, both text and graphic data cannot be displayed concurrently. Therefore, either the mode to effect the text display or that to effect the graphic display must be

selected by changing the display mode. Moreover, because of the limited memory size of the display controller, the contents of the virtual screen will be lost when the graphic display is activated.
The display controller is capable of color selection, which is different depending on the display mode. In Text display mode, the background colors are green or orange with the color of all characters fixed. In Graphic display mode, there are two color sets 0 and 1. All the colors in the same color set can be used as background colors. Other colors are available for dots.

Color set 0	Color set 1
Green	White
Yellow	Cyan
Blue	Magenta
Red	Orange

15.4 Virtual Screen Control

The movement of the cursor, deletion of one character, and other controls related to the display contents on the virtual screen are performed by using character codes. Special controls such as screen size specification, list control, etc., are provided as the functions of the virtual screen.
The character codes used are 00 through FF. Codes 20 through FF are those to be displayed on the screen as graphic characters. Code 00 through 1F are non-graphic characters which are not displayed on the screen. They are used as control characters for cursor movement, etc. The description of each character code follows.

(1) Graphic characters

(a) When not at the right-hand end on the bottom line
The next line is assumed to be a continuation line. (Line status is FF.) (See Fig. 9-6.)

(b) When at the right-hand end on the bottom line
The display contents are scrolled up by one line. The bottom line becomes a continuation line filled with blank codes (20).

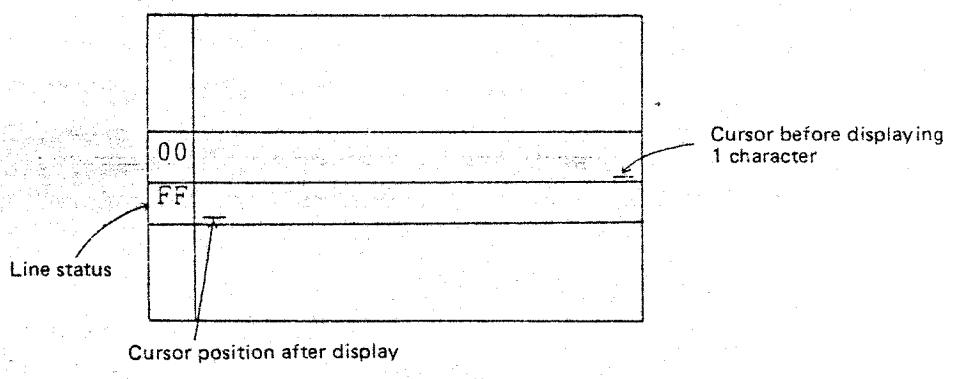


Fig. 15-5 Continuation Line

(2) Control codes

19 character codes can be used as control codes. The functions of the respective control codes are as follows:

(a) 01 (Window Left)

Positions the window to the left end of the virtual screen. The cursor moves to the 10th column of the window.

(b) 04 (Scroll Right)

Moves the window to the right by the number of columns specified by the horizontal scroll steps. However, the window will not move beyond the right end of the virtual screen.

(c) 05 (Clear to End of Line)

Changes all the characters from the cursor position to the end of the logical single line to blank codes (20).

(d) 06 (Window Right)

Positions the window to the right end of the virtual screen. The cursor moves to the 10th column of the window.

(e) 08 (Delete one Character)

Moves the cursor position back by one character and deletes the character at the cursor position. All the data following the deleted character on the line are shifted and a blank code (20) is entered at the end of the line. When the cursor is at the beginning of the line and therefore cannot be moved back, the character at the current cursor position is deleted.

(f) 09 (TAB)

Moves the cursor to the next Tab position. Tab positions are set at every 8 columns such as 0, 8, 16 .....

(g) 0A (Line Feed)

Moves the cursor down by one line. When the cursor is at the bottom line of the virtual screen, the virtual screen scrolls one line and the bottom line will be filled with blank codes.

(h) 0B (Home)

Positions the cursor to the upper left corner of the virtual screen. The window moves along with the cursor. (This position is referred to as "Home position").

(i) 0C (Clear)

Changes all the contents of the virtual screen to blank codes (20). The logical single line is set to the virtual screen width and the cursor returns to the home position.

(j) 0D (Carriage Return)

Terminates the logical single line. (The line status of the next line becomes 00.) The cursor moves to the left end of the line.

(k) 10 (Scroll Up)

Moves the window up by the number of lines specified by the vertical scroll steps. The window will not move beyond the top end of the virtual screen. The cursor moves to the 10th column of the virtual screen.

(l) 11 (Scroll Down)

Moves the window down by the number of lines specified by the vertical scroll steps. The window will not move below the bottom end of the virtual screen. The cursor moves to the 10th column of the virtual screen.

(m) 12 (Insert)
Inserts a blank code (20) into the cursor position. All the characters following the cursor position are shifted to the right by 1 column. If the last character in the logical single line is a blank code, that character is deleted. If the last character is not a blank code, another line filled with blank codes will be inserted (i.e., scrolling takes place above the cursor position) and the last character is positioned at the beginning of the inserted line.

(n) 13 (Scroll Left)
Moves the window to the left by the number of columns specified by the horizontal scroll steps. However, the window will not move beyond the left end of the virtual screen.

(q) 1A (Clear to End of Screen)
Changes the contents of the virtual screen from the current cursor position to the end of the virtual screen to blank codes. The logical single line is set to the virtual screen width. (Line status is changed to "00".)

(p) 1C (Cursor Right)
Moves the cursor to the right by one column. The cursor at the right end of a line will move to the beginning of the next line. If the cursor is on the bottom line, it will move to the left end of the same line.

(q) 1D (Cursor Left)
Moves the cursor to the left by one column. The cursor at the left end of a line will move to the right end of one immediately above the line. If the cursor is in the top line, it will move to the right end of the same line.

(r) 1E (Cursor Up)
Moves the cursor up by one line. The cursor will not move if it is in the top line.

(s) 1F (Cursor Down)
Moves the cursor down by one line. The cursor will not move if it is in the bottom line.

(3) Subroutine call for virtual screen
The virtual screen is supported by subroutine "SCRFNC". Parameters for this subroutine are given by the parameter packet used on the memory. The packet begins with a 1-byte function code which is followed by a series of several data. The return information is also included in the packet.

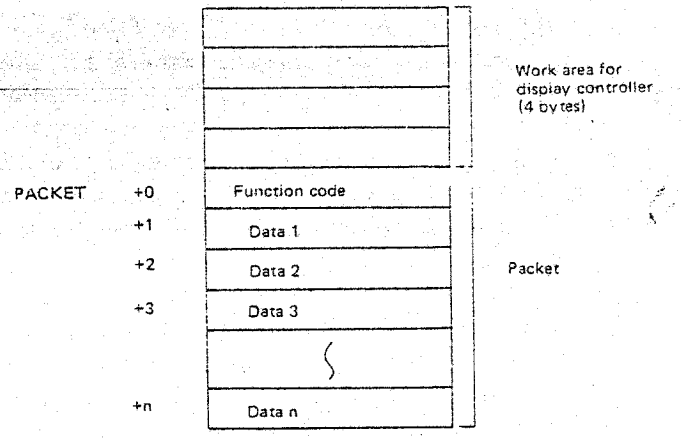


Fig. 15-6 Parameter Packet

A 4-byte work area is required before the packet for the display controller functions. (See Fig. 9-6.)
To call subroutine "SCRFNC" (Entry point FF5E), the top address of the packet must be given to the index register.
Example: To call the function to set the virtual screen. In this example, the screen size of 40 x 8 is defined for the LCD and the buffer address is specified at 5000.

```
SCRFNC    EQU   $FF5E
          LDX   #PACKET
          JSR   SCRFNC
          :
PACKET    FCB   $87        *Function code(define screen size)
          FCB   39         *Screen width
          FCB   7          *Screen depth
          FDB   $5000      *Buffer address
          :
          ORG   $5000
          RMB   40*9+1     *Buffer size
```

° Functions for Initialization of Virtual Screen

The following functions must be executed to initialize the virtual screen.

1. Function 84 (Screen device select)
2. Function 87 (Specification of screen size and buffer address)
3. Function C3 (Specification of scroll margin)
4. Function C4 (Specification of scroll steps)
5. Function CB (Specification of scrolling speed)

(Refer to Section 9.5 for detailed description of each function code.)

15.5 Virtual Screen Function Table

Packets for the virtual screen are as listed below. The virtual screen functions are divided into those shared by both the LCD and the display controller and those peculiar to either one. The device to which the particular function is applicable is shown in the "Function (Application)" column as (LCD) for the display and as (Disp) for the display controller. Data in each packet are numbered as 0, 1, 2, ... for each byte and their descriptions are given in order of the data number. These packets are used at the time of both the entry and return of each subroutine. In the following table, "XX" indicates that arbitrary 2-digit values may be used, and unless otherwise specified, all numeric values are hexadecimal.

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
84		Screen device select.	
(Disp) (LCD)	00	84 (Function code) Device code 30: Disp 22: LCD	Return code 00: Normal FE: Device is not connected. FF: Device specification is invalid.
85		Initialization of the Display controller. The Display controller is initialized at the default value.	
(Disp)	00 01	85 (Function code) XX	Return code 00: Normal FF: I/O error
87		Specification of the virtual screen. By this function, the screen size and the top address of the buffer are specified. When the screen size is m columns by n lines, the size of the buffer must be m x n + 1 bytes.	
(Disp) (LCD)	00 01	87 (Function code) Screen width (Specify m - 1 for m columns.) Screen length (Specify n - 1 for n lines.) High-order byte of buffer's top address Low-order byte of buffer's top address NOTE: Buffer addressing is not required if the display controller is specified.	Return code 00: Normal FF: Screen oversize

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
88		Input of the virtual screen size. By this function, the currently defined size of the virtual screen is obtained.	
(Disp) (LCD)	00 01 02	88 (Function code) XX XX	 Screen width (m - 1 for m columns) Screen length (n - 1 for n lines)
89		Input of the window size.	
(Disp) (LCD)	00 01 02	89 (Function code) XX XX	 Width: 19 (D) for LCD 31 (D) for Display controller Length: 3 (D) for LCD 15 (D) for Display controller
8A		Input of the window position. By this function, the coordinate values at the upper left corner of the window on the virtual screen are given.	
(Disp) (LCD)	00 01 02	8A (Function code) XX XX	 Coordinate x Coordinate y
8C		Input of the cursor position. By this function, the position of the cursor on the virtual screen is obtained.	
(Disp) (LCD)	00 01 02	8C (Function code) XX XX	 Coordinate x Coordinate y
8D		Input of the cursor margin value.	
(Disp) (LCD)	00 01	8D (Function code) XX	 Margin value
8E		Input of the scroll steps.	
(Disp) (LCD)	00 01 02	8E (Function code)	 Number of horizontal scroll steps Number of vertical scroll steps

Function (Application)	Packet data number	Description (At Entry)	(At Return)
8F		By this function, the dot status at the specified position on the physical screen is obtained.	
(Disp)	00	8F (Function code)	XX
(LCD)	01	High-order byte of coordinate x	(1) LCD FF: ON 00: OFF (2) Display controller Color code
	02	Low-order byte of coordinate x	XX
	03	High-order byte of coordinate y	XX
	04	Low-order byte of coordinate y	XX
91		Input of the range of the logical single line. By this function, the range of the logical single line containing the cursor on the virtual screen is specified.	
(Disp)	00	91 (Function code)	XX
(LCD)	01	XX	First columm in the logical single line (Coordinate x with a value 0)
	02	XX	First line in the logical single line (Coordinate y)
	03	XX	Physical screen width (LCD: 19 (D), Disp: 32 (D))
	04	XX	Last line in the logical single line (Coordinate y)
92		Display of one character on the virtual screen.	
(Disp)	00	Character code	Coordinate x of the new cursor position
(LCD)	01	XX	Coordinate y of the new cursor position
93		Specification of a display mode for the display controller.	
(Disp)	00	93 (Character code)	XX
	01	Text mode 00: Graphic mode 01: Text mode	Return code 00: Normal FF: An error has occurred.
	02	Graphic mode 00: Text mode 01: Color graphic mode 02: Monochromatic graphic (high-resolution) mode Note: Text mode and graphic mode must be specified exclusively. In other words, either data 00 or 01 must be 00.	XX

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
		Graphic mode is supported on the physical screen. The resolution of the display is 128 x 64 dots in color graphic mode and 128 x 96 dots in monochromatic graphic mode (i.e., high-resolution mode).	
	03	Background color 00: Green 04: White 01: Yellow 05: Cyan 02: Blue 06: Magenta 03: Red 07: Orange Note: Background color selection is effective in Graphic mode only. A color set is defined by the COLOR command in Text mode.	XX
95		Input of one character on the display. By this command, the character at the coordinates specified by the access pointer is input.	
(Disp)	00	95 (Function code)	XX
	01	XX	Character code
	02	XX	Color code (Background color code)
97		Consecutive input of characters from the virtual screen. By this function, characters are input in the number specified from the coordinate positions where data read starts.	
(Disp) (LCD)	00	97 (Function code)	XX
	01	Coordinate x at the read start point	Input character 1
	02	Coordinate y at the read start point	Input character 2
	03	Number of read characters	Input characters 3
	04	XX	
98		Display of one character on the virtual screen. (Note that the packet generation in this case is different from that of function 92.)	
	00	98 (Function code)	XX
(Disp) (LCD)	01	XX	Coordinate x of the new cursor position
	02	XX	Coordinate y of the new cursor position
	03	XX	First line number in the logical single line containing the new cursor (Coordinate y)
	04	XX	Last line number in the logical single line containing the new cursor (Coordinate y)

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
C0		Setting of the window position. By this function, the upper left edge of the window is positioned at the specified address on the virtual screen.	
(Disp) (LCD)	00 01 02	C0 (Function code) Coordinate x on the virtual screen Coordinate y on the virtual screen Note: If the window position is outside the bounds of the virtual screen, the maximum values are set for both coordinates x and y.	XX XX XX
C2 (Disp) (LCD)		Specification of the cursor position. By this function, the cursor is placed at the specified position on the virtual screen, resulting in the movement of the window.	
	00 01 02	C2 (Function code) Coordinate x of the cursor position Coordinate y of the cursor position Note: The window movement is controlled as follows: (1) The window does not move when the specified cursor position is within the window area. (2) When the specified cursor position is not within the window area, the window moves so that the new cursor is located at the home position of the window. The cursor position cannot be located at the home position of the window, if the bottom edge of the window is in alignment with the bottom edge of the virtual screen. In such a case, the cursor position is set within the window area according to the same rule as that of function code C0.	XX XX XX
C3		Setting of the value of the cursor margin.	
(Disp) (LCD)	00 01	C3 (Function code) Cursor margin value (This value must be in the range from 1 to half the window value.)	XX XX
C4		Setting of the number of scroll steps.	
(Disp) (LCD)	00 01	C4 (Function code) Number of horizontal scroll steps (0 to 255 (D)) Number of vertical scroll steps (0 to 255 (D))	XX XX XX

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
C5 (Disp) (LCD)	00	Turning the list flag ON.	
		C5 (Function code)	XX
C6 (Disp)	00	Resetting of the list flag.	
		C6 (Function code)	XX
C7 (Disp) (LCD)		Setting of a dot at the specified position. This function is effective in Graphic mode.	
	00	C7 (Function code)	XX
	01	High-order byte of coordinate x	XX
	02	Low-order byte of coordinate x	XX
	03	High-order byte of coordinate y	XX
	04	Low-order byte of coordinate y	XX
	05	Color code With LCD, 00: OFF, FF: ON With Display controller, if color set 0 is specified 00: Green 01: Yellow 02: Blue 03: Red if color set 1 is specified 00: White 01: Cyan 02: Magenta 03: Orange	
C8 (Disp) (LCD)		Drawing a straight line between any two points on the graphic screen.	
	00	C8 (Function code)	XX
	01	High-order byte of coordinate x at the start point	XX
	02	Low-order byte of coordinate x at the start point	XX
	03	High-order byte of coordinate y at the start point	XX
	04	Low-order byte of coordinate y at the start point	XX
	05	High-order byte of coordinate x at the end point	XX
	06	Low-order byte of coordinate x at the end point	XX
	07	High-order byte of coordinate y at the end point	XX
	08	Low-order byte of coordinate y at the end point	XX
	09	Color code. Same as function code 07	XX

Function (Application)	Packet data number	Description (At Entry)	Description (At Return)
C9		Termination of the logical single line. By this function, the line status of the specified line is reset to 00.	
(Disp) (LCD)	00 01	C9 (Function code) Line number (coordinate y)	XX XX
CA		Clearing of the screen in Graphic mode. This function is effective for the graphic screen when the Display controller is used, and for the physical screen when the LCD display is used.	
(Disp)	00 01	CA (Function code) Background color (Effective only with Display controller)	XX XX
CB		Setting of the scrolling speed. This function specifies the scrolling speed of the physical screen.	
(LCD)	00 01	CB (Function code) Speed A value in the range of 00 to 09 is used to specify the scrolling speed. 9 is the highest scrolling speed.	XX
CD		Output of one character to the position specified by the access pointer.	
(Disp)	00 01	CD (Function code) Character code	XX XX
CE (Disp)		Specification of the access pointer. By this function, the character position that can read/write on the virtual screen when the Display controller is used is specified.	
	00 01 02	CE (Function code) Coordinate x of the access pointer Coordinate y of the access pointer	XX XX XX
CF		Specification of a color set. Two color sets each consisting of 4 different colors are selectable when the Display controller is used.	
(Disp)	00 01	CF (Function code) Color set 00: Color set 0 01: Color set 1 If color set 0 is specified, green, yellow, blue and red can be used. If color set 1 is specified, white, cyan, magenta and orange can be used.	XX XX

15.6 EPSP Message Format Table for Screen

In the following table, SS and MM refer to the slave and master device numbers, respectively. Numeric values are all hexadecimal.
"XX" indicates that arbitrary 2-digit values may be used.

(1) Function Codes for Display Controller

Function code	FMT	DID	SID	FNC	SIZ	Text data number	Description of function and text
84	00	SS	MM	84	00	00	Screen device select Device number (30)
	01	MM	SS	84	00	00	Return code 00: Normal FE: Device is not ready. FF: Device number is invalid.
85	00	SS	MM	85	00	00	Initialization of screen XX
	01	MM	SS	85	00	00	Return code 00: Normal FF: An error has occurred.
87	00	SS	MM	87	03	00 01 02 03	Specification of the screen size Virtual screen width (maximum value of coordinate x) Virtual screen length (maximum value of coordinate y) XX XX
	01	MM	SS	87	00	00	Return code 00: Normal FF: Size specification is invalid.
88	00	SS	MM	88	00	00	Input of the virtual screen size. XX
	01	MM	SS	88	01	00 01	Virtual screen width (maximum value of coordinate x) Virtual screen length (maximum value of coordinate y)
89	00	SS	MM	89	00	00	Input of the physical screen
	01	MM	SS	89	01	00 01	Screen width (maximum value of coordinate x) Screen length (maximum value of coordinate y)

Function code	FMT	DID	SID	FNC	SIZ	Text data number	Description of function and text
C0							Positioning of the physical screen on the virtual screen. Position values are given with respect to the position (0,0) of the physical screen.
	00	SS	MM	C0	01	00	Coordinate x of the specified position.
						01	Coordinate y of the specified position.
	01	MM	SS	C0	00	00	xx
8A							Input of the physical screen position on the virtual screen. Position values are given with respect to the position (0,0) of the physical screen.
	00	SS	MM	8A	00	00	XX
	01	MM	SS	8A	01	00	Coordinate x of the specified position
						01	Coordinate y of the specified position
C2							Specification of the cursor position on the virtual screen.
	00	SS	MM	C2	01	00	Coordinate x of the specified position
						01	Coordinate y of the specified position
	01	MM	SS	C2	00	00	XX
8C							Input of the cursor position on the virtual screen
	00	SS	MM	8C	00	00	XX
	01	MM	SS	8C	01	00	Coordinate x of the specified position
						01	Coordinate y of the specified position
C3							Setting of the margin value of the cursor
	00	SS	MM	C3	00	00	Margin value
	01	MM	SS	C3	00	00	XX
8D							Input of the cursor margin value
	00	SS	MM	8D	00	00	XX
	01	MM	SS	8D	00	00	Margin value
C4							Setting of the number of scroll steps
	00	SS	MM	C4	01	00	Number of horizontal scroll steps
						01	Number of vertical scroll steps
	01	MM	SS	C4	00	00	XX

Function code	FMT	DID	SID	FNC	SIZ	Text data number	Description of function and text
8E							Input of scroll steps
	00	SS	MM	8E	00	00	XX
	01	MM	SS	8E	01	00	Number of horizontal scroll steps
						01	Number of vertical scroll steps
C5							Setting of the list flag
	00	SS	MM	C5	00	00	XX
	01	MM	SS	C5	00	00	XX
C6							Resetting of the list flag
	00	SS	MM	C6	00	00	XX
	01	MM	SS	C6	00	00	XX
C7							Setting of a dot at the specified position
	00	SS	MM	C7	04	00	High-order byte of coordinate x
						01	Low-order byte of coordinate x
						02	High-order byte of coordinate y
						03	Low-order byte of coordinate y
						04	Color code
	01	MM	SS	C7	00	00	XX
8F							Input of the dot status at the specified position
	00	SS	MM	8F	03	00	High-order byte of coordinate x
						01	Low-order byte of coordinate x
						02	High-order byte of coordinate y
						03	Low-order byte of coordinate y
	01	MM	SS	8F	00	00	Color code
C8							Drawing of a straight line
	00	SS	MM	C8	08	00	High-order byte of coordinate x at the start point
						01	Low-order byte of coordinate x at the start point
						02	High-order byte of coordinate y at the start point
						03	Low-order byte of coordinate y at the start point
						04	High-order byte of coordinate x at the end point
						05	Low-order byte of coordinate x at the end point
						06	High-order byte of coordinate y at the end point
						07	Low-order byte of coordinate y at the end point
						08	Color code
	00	MM	SS	C8	00	00	XX

Function code	FMT	DID	SID	FNC	SIZ	Text data number	Description of function and text
91							Input of the range of the logical single line containing the cursor
	00	SS	MM	91	00	00	XX
	01	MM	SS	91	03	00	00
						01	Coordinate y of the first line in the logical single line
						02	Column size of the physical screen
						03	Coordinate y of the last line in the logical single line
C9							Resetting of the line status of the specified line (i.e., partitioning of the logical single line)
	00	SS	MM	C9	00	00	Line number
	01	MM	SS	C9	00	00	XX
92							Display of one character on the virtual screen
	00	SS	MM	92	00	00	Character code
	01	MM	SS	92	01	00	Coordinate x of the cursor
							Coordinate y of the cursor
CA							Specification of the background color in Graphic mode
	00	SS	MM	CA	00	00	Color code
	01	MM	SS	CA	00	00	XX
CB	00	SS	MM	CB	00	00	Setting of the scrolling speed A value in the range of 0 to 9 is used to specify the scroll.
	01	MM	SS	CB	00	00	XX
93	00	SS	MM	93	02	00	Specification of display mode Text mode 00: Mode other than Text mode 01: Text mode
						01	Graphic mode 00: Mode other than Graphic mode 01: Graphic mode 1 02: Graphic mode 2
							Note: Either data 00 or data 01 must be "00". Both data must not be "00".
						02	Background color 00: Green 01: Yellow 02: Blue 03: Red 04: White 05: Cyan 06: Magenta 07: Orange
	01	MM	SS	93	00	00	Return code 00: Normal FF: An error has occurred.

Function code	FMT	DID	SID	FNC	SIZ	Text data number	Description of function and text
CD							Writing of one character into the access pointer
	00	SS	MM	CD	01	00	Character code
						01	Color code
	01	MM	SS	CD	00	00	XX
CE							Specification of the access pointer against the virtual screen
	00	SS	MM	CE	01	00	Coordinate x
						01	Coordinate y
	01	MM	SS	CE	00	00	XX
95							Input of one character from the access pointer
	00	SS	MM	95	00	00	XX
95	01	MM	SS	95	01	00	Character code
						01	Color code
CF							Selection of a color set
	00	SS	MM	CF	00	00	00: Color set 0
							01: Color set 1
	01	MM	SS	CF	00	00	XX
97							Input of characters at the positions specified consecutively on the virtual screen
	00	SS	MM	97	03	00	Coordinate x of the start point
						01	Coordinate y of the start point
						02	High-order byte of the number of input characters
						03	Low-order byte of the number of input characters
	01	MM	SS	97	mm	00	00~mm denote the character codes of the input
						mm	characters.
98							Display of one character on the virtual screen followed by the input of the first and last line numbers of the logical single line including the newly set cursor position.
	00	SS	MM	98	00	00	Character code.
	01	MM	SS	98	03	00	Coordinate x of the new cursor position
						01	Coordinate y of the new cursor position
						02	First line number in the logical single line
						03	Last line number in the logical single line.

CHAPTER 16 MENU

16.1 General

The title or entry point of a program can be registered or displayed by the MENU function of the HX-20. This chapter first describes the ID structure of the application programs stored in the ROMs of the HX-20, then explains how the ID information is displayed by the MENU with examples.

16.2 ID Structure

The ID (identifying information also called a "header") for both the ROM and user (RAM) application programs is structured as described below.
When the user writes an application program into a ROM or RAM and wishes to display the program on the MENU, he must write the header information to identify the program. Particularly, for an application program stored in a ROM, this header information must be at the top address (low-order address) of the ROM.

16.2.1 Header of ROM/RAM application program

(1) ID 1 (1 byte)
Bit 0 - bit 6 ':' (Code 3A)
Bit 7
0 = The header contains a link address to the next program on the ROM or RAM. The linked program is not displayed on the MENU. This bit can be used when the user writes programs using an EPROM. In other words, if the user wishes to erase a program on the EPROM, bit 7 should be changed from logic "1" to "0" using an EPROM writer. (Bit 7 is "1" with the EPROM in the initialized state.)
1 = Header contains a link address with the next program on the ROM or RAM, the starting address (entry point) of a program and its program name.

(2) ID 2 (1 byte)
Bit 0 - bit 6 = The header information contains one of the following codes:
"A" = Application program (application for general use)
"B" = BASIC interpreter
"E" = End of link (No application program follows this header information.)
(RAM application for general use)
Bit 7
0 = Indicates that the linkage with the next program is an absolute value (i.e., absolute address).
1 = Indicates that the linkage with the next program is a relative value (i.e., offset value from the header).
If the ROMs are made available for use on any sockets, programs are relocatable and thus bit 7 must be set to logic "1".

(3) Pointer to next header (2 bytes)
This header information is also called a "link address".
This two-byte data is used as a pointer to the location of the next header. If no next header exists within the same ROM, the value of this data is "FFFF".
If the MENU finds value "FFFF" on a ROM, it scans the next ROM for header information.

(4) Starting address of program (entry point) (2 bytes)
This header information indicates the starting address of a program. The starting address is an absolute value if the bit 7 of ID2 is logic "1" and an offset value from the beginning of this header information if bit 7 is "0".
(5) Filename (program name) (17 bytes max.)
A filename is entered in a maximum of 16 bytes in ASCII code. The last byte of this header information is always "00".

16.2.2 Header of BASIC application program
The header of a BASIC application program (i.e., an application program written in BASIC by the user) is different from that of a ROM/RAM application program.
BASIC application programs have no linkage with ROM/RAM application programs. However ROM/RAM application programs are displayed automatically by the MENU function.
(1) Link offset (2 bytes)
This is a pointer to indicate the starting address of the header of the next BASIC program. For example, program 1 points at program 2, while program 2 points at program 3. When the link offset value is FFFF, it indicates that no next header exists.
(2) Filename (program name) (8 bytes)
The filename of a program is specified by the TITLE command of BASIC. If the program has no filename, blanks must be entered as the filename in the header.

16.2.3 Bit map (2 bytes) and link tables (4 bytes, 013C to 013F)
After the input of "CTRL/@"; the MENU generates a bit map which indicates the presence of the header of a ROM application program, and a link table for linkage with an RAM application program. Bit map addresses are 013A and 013B.

013A	Bit 7	*	ROM at addresses E000 to FFFF of bank 0
	Bit 6	*	ROM at addresses C000 to DFFF of bank 0
	Bit 5	*	ROM at addresses A000 to BFFF of bank 0
	Bit 4	*	ROM at addresses 8000 to 9FFF of bank 0
	Bit 3	*	ROM at addresses 6000 to 7FFF of bank 0
	Bit 2	*	ROM at addresses 4000 to 5FFF of bank 0
	Bit 1	*	ROM at addresses 2000 to 3FFF of bank 0
	Bit 0	*	ROM at addresses 0000 to 1FFF of bank 0
013B	Bit 7	*	ROM at addresses E000 to FFFF of bank 1
	Bit 6	*	ROM at addresses C000 to DFFF of bank 1
	Bit 5	*	ROM at addresses A000 to BFFF of bank 1
	Bit 4	*	ROM at addresses 8000 to 9FFF of bank 1
	Bit 3	*	ROM at addresses 6000 to 7FFF of bank 1
	Bit 2	*	ROM at addresses 4000 to 5FFF of bank 1
	Bit 1	*	ROM at addresses 2000 to 3FFF of bank 1
	Bit 0	*	ROM at addresses 0000 to 1FFF of bank 1

```
* = 0 :  No header exists in the specified ROM socket.
* = 1 :  Header exists in the specified ROM socket.
Bank 0 : Main memory of HX-20
Bank 1 : Memory in the expansion unit for HX-20
```

The link table after the input of "CTRL/@" contains 4-byte data "1:/'E'/FF/FF/". If the user wishes to display any program on the RAM in the MENU, he just needs to link this 4-byte data in the link table to his object program. For example, if the user writes an application program from address 1000, the header of the RAM application program and its link table should be written as follows.

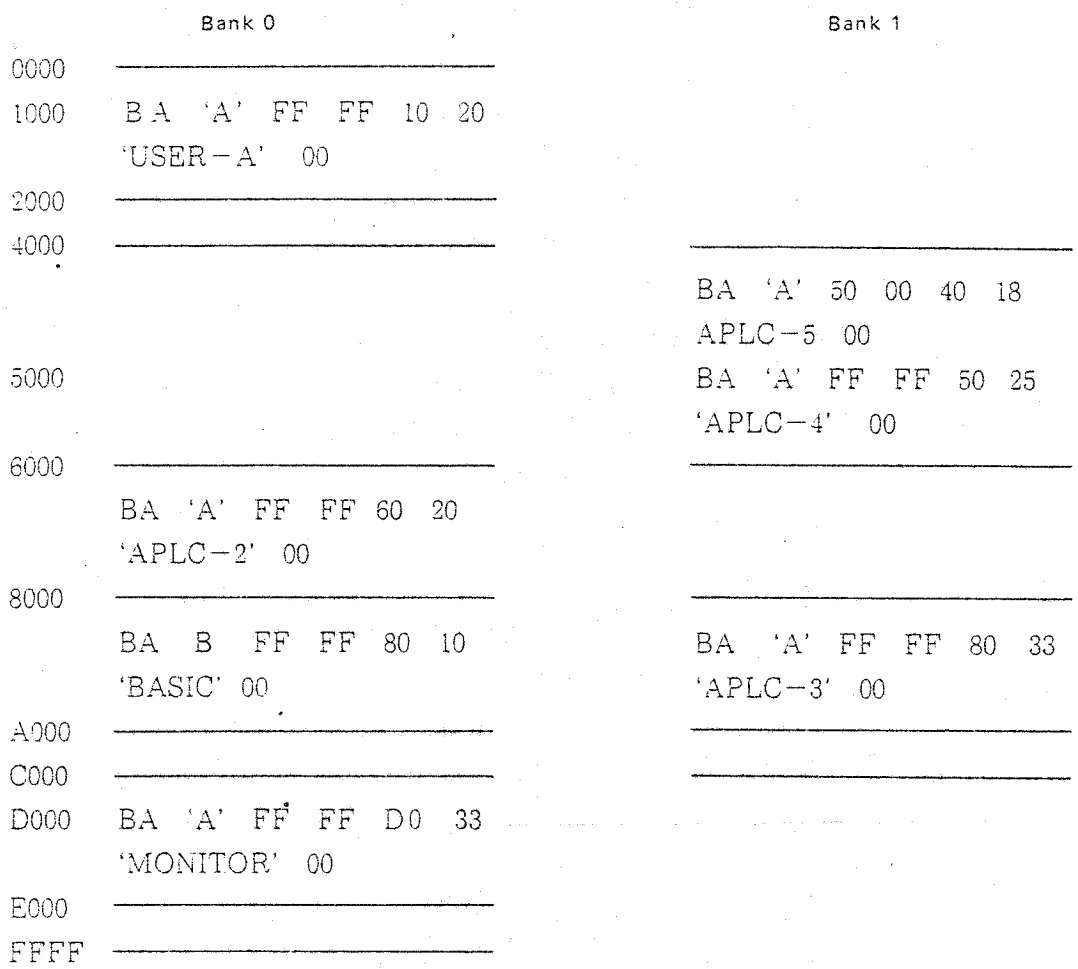
```
1000    /: (bit7=1)/'A'/FF/FF/10/20/U/S/E/R/00/
13C     /: (bit7=0)/'A'/10/00/
```

16.2.4 How bit map and link table are generated

Neither a bit map nor a link table exists before the HX-20 system is initialized (by pressing CTRL and @ keys) (see Section 1.3). Before the system is cold started by "CTRL/@", "CTRL/@ Initialize", "1 MONITOR" and dummy names (19 max.) will appear in the MENU on the LCD. After pressing CTRL and @ keys, the MENU generates a bit map and a link table. When generating a bit map by the MENU, program linking starts from address D000 (MONITOR). Next, scanning of addresses starts from A000 (bank 1 also if an expansion unit is connected) and progresses to addresses 8000, 6000 and 4000 in the order named. The MENU sets the bit map depending on whether or not the header of an application program exists, and writes "/:/'E'/FF/FF/" into the link table.

Subsequently, the MENU displays the filename of a ROM application filename according to the bit map. Next, if there is any linked RAM (user) application program, then the name of the RAM application program is displayed, followed by BASIC application programs.

16.3 Examples



Assume that there are 2 BASIC application programs (APLC-1 and APLC-2) in addition to the above ROM/RAM application program.
The bit map in this case will be as follows:

```
                MSB      LSB
    13A          01011000
    13B          00010100
```

and the link table will be as follows.

```
    13C          /:/'A'/10/00
```

The following information will appear in the MENU on the LCD display.

```
    CTRL/@       Initialize
    1 MONITOR
    2 BASIC
    3 APLC-3
    4 APLC-2
    5 APLC-5
    6 APLC-4
    7 USER-A
    8 APLC-1
    9 APLC-2
```

16.4 MENU Work Areas

Address (from)	(to)	Variable name	Bytes	Description
2D0	48A	SCNBUF	442	Buffer for MENU display
78	78	INTFLG	1	Initialize flag (0: Request; 1: Complete) Bit 0: MENU Bit 7: BASIC Condense (garbage collection) flag (1: Condense request) Bit 6: (BASIC, Application) Condense
7B	7B	RUNMOD	1	Run mode 01: MENU
7E	7E	SFTSWH	1	Software switch 1 Bit 4: Bank switch number currently selected (0: Bank 0; 1: Bank 1) Bit 5: Bank switch number selected before current number Bit 6: Bank number in which BASIC programs are stored Bits 5 and 6 are used to condense application.
80	81	TMPBF1	2	Temporary buffer
82	83	CNTMNU	2	Indicates the top address of ROM (C000, A000, 8000, ...).
84	84	CNTMNU	1	Number of items currently on the MENU display - 1.
85	85	MNUNUB	1	MENU number
86	86	BITMP	1	Bit map value of a bank (for temporary use)
87	87	BBTMP0	1	Buffer for BITMP0 (bit map of bank 0)
88	88	BBTMP1	1	Buffer for BITMP1 (bit map of bank 1)
89	89	STKLIN	1	Maximum number of lines on MENU display
8A	8A	MXMNUB	1	Maximum number of MENUs (ASCII code)
8B	8B	BSAPNB	1	BASIC application number
8C	8C	CNTFLG	1	Work area for temporary use
8D	8D	DISFLG	1	Work area for temporary use

Address (from)	(to)	Variable name	Bytes	Description
8E	92	PCKT	5	LCD buffer work area for virtual screen packet
13A	13A	BITMP0	1	Bit map for bank 0 Indicates whether the header of a ROM application program exists in one of the ROM chips in bank 0. (0: No header exists; 1: A header exists.) Bit 0: Address 0000 of bank 0 Bit 1: Address 2000 of bank 0 Bit 2: Address 4000 of bank 0 Bit 3: Address 6000 of bank 0 Bit 4: Address 8000 of bank 0 Bit 5: Address A000 of bank 0 Bit 6: Address C000 of bank 0 Bit 7: Address E000 of bank 0
13B	13B	BITMP1	1	Bit map for bank 1 Indicates whether the header of a ROM application program exists in one of the ROM chips in bank 1. (0: Header does not exist; 1: Header exists.) Bit 0: Address 0000 of bank 1 Bit 1: Address 2000 of bank 1 Bit 2: Address 4000 of bank 1 Bit 3: Address 6000 of bank 1 Bit 4: Address 8000 of bank 1 Bit 5: Address A000 of bank 1 Bit 6: Address C000 of bank 1 Bit 7: Address E000 of bank 1
13C	140	LNKTBL	4	Link table for RAM application programs (I) When RAM application program does not exist : E FF FF (II) When the header of a RAM application program exists : A Address of the RAM application program

CHAPTER 17 MONITOR

17.1 General

The Monitor is located in the ROM (ROM2) area from C000 to DFFF and has two entry points DFF7-DFF9 and DFFA-DFFC. The former is for entry from the menu display, etc., while the latter is for entry when a trap interrupt is generated. If one of the trap interrupt addresses (0106 through 0108) is specified, the default assumption is "JMP $DFFA". The display of data by the Monitor is always on the physical screen and the virtual screen is never used for the monitor display.
The HX-20 Monitor has 10 types of commands as listed below.

(1) S (Set) command : Displays and changes the contents of the memory.
(2) D (Dump) command : Displays the contents of the memory.
(3) G (Go) command : Executes a program.
(4) X (Examine) command : Displays and changes the contents of each register.
(5) R (Read) command : Loads a program or data into the memory from an external storage.
(6) W (Write) command : Saves the contents of the memory to an external storage.
(7) V (Verify) command: Verifies the data output to an external storage.
(8) K (Key) command : Specifies the data for automatic key input when the power switch is turned ON.
(9) A (Address) command : Specifies the range of the memory space when loading from an external storage or saving data to an external storage.
(10) B (Back) command : Returns control to the procedure by which the Monitor was called.

Refer to the HX-20 OPERATION MANUAL for detailed description of each monitor command.

17.2 About Trap

If an attempt to execute a command not defined for the MCU is made, a trap interrupt is generated. By utilizing this characteristic, a breakpoint is set by the G command. For example, write "00" (undefined code) in the address specified as a breakpoint. Then, try to execute the command at this address, and a trap interrupt will be generated, causing the HX-20 to return to the Monitor mode again.

Address (from)	(to)	Variable name	Bytes	Description
2A0	2A1	BP1	2	Stores the address specified as a breakpoint.
2A2	2A2	BPD1	1	Stores the contents of the breakpoint address.
2A3	2A3	LCDSTS	1	Stores the LCD status ('DISSTS': Address 0280) when the HX-20 enters Monitor mode.
2A4	2BE		27	Work area for packets of binary dump/load routine.
2BF	2C0	PC	2	Stores the program counter value.
2C1	2C2	RTNADD	2	Stores Return address on execution of B command.
2C3	2C4	LINLST	2	Stores the Buffer address corresponding to the end of the first line of the physical screen.
2C5	2C5	SRNMOD	1	Stores the R option of R command.
2C6	2CF		10	Unused.