TFT LCD Module with MPCT

(Metallised Projected-Capacitive Touch)

GT-C9xxP series

“General Function” Software Specification

Model: GT-C9xxPA series

Specification No.:DS-2007-0003-00

Date of Issue: June 27, 2017 (00)

Revision:

Published by
NORITAKE ITRON Corp. / Japan
http://www.noritake-itron.jp

This specification is subject to change without prior notice.

Contents

1 General Description

1.1 Scope

This specification covers the software aspects and supported functions of the Noritake Itron TFT-LCD display module with Metallised Projected-Capacitive Touch (MPCT) panel, GT-C9xxP series.

The TFT-LCD module supports text and graphic display using simple ASCII-based commands, compatible with the GU-3000 series VFD modules produced by Noritake Itron.
Extra commands, and additional parameters to existing commands, are implemented in order to take advantage of the full-color display capability.

Related specifications:

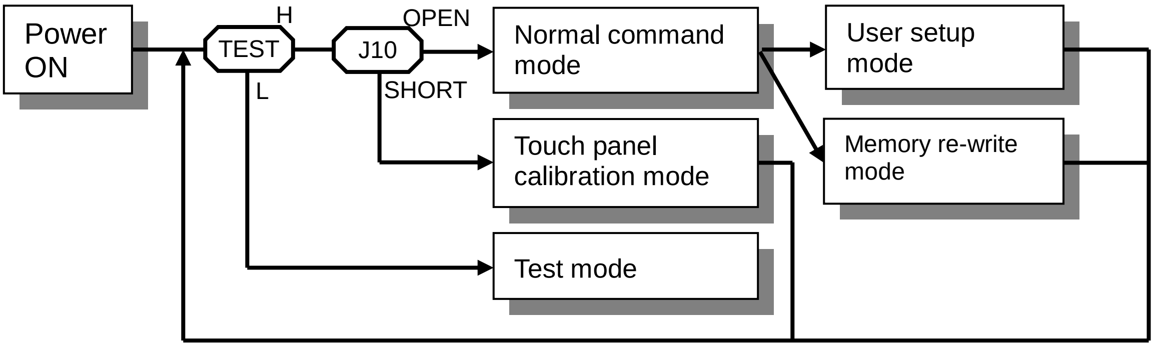
Hardware specification:	DS-2007-0001-xx
Program Macro specification:	DS-1940-0005-xx
	(Refer to 3.6.4.64 RAM Macro define / delete)
	(Refer to 3.6.4.65 FROM Macro define / delete)
Character font specifications:	(Refer to 3.1.2 Character display)

1.2 Functions

Character display, Graphic display, Control command, Display action command, Download (user-definable) font, User-definable font table function, Draw command, Window function, General-purpose I/O port control, Macro, Program Macro function, Bit Image download function, Memory SW, Data storage, Touch Panel control command.

2 Operating mode

The operating modes are as follows, selected by jumper settings or software command.



2.1 Normal command mode

Normal operation mode – the module can receive commands and data via the various interfaces.

2.2 User setup mode

This mode is used for saving Memory-SW and various data to FROM (FROM1).

2.3 Touch panel calibration mode

The display module includes a touch position correction function, for minimizing any inconsistency between actual touch position and the internally-calculated position (refer to Hardware Specification for details).

2.4 Memory re-write mode

Mode for re-writing firmware and built-in font data. Not for routine use.

2.5 Test mode

Test for display and internal operation. Used for factory test.

2.6 Power-on settings

At power-on, the various display settings are set to default value, or value stored in Memory SW (Refer to 4.2 Memory SW).
If “restore at power-on” is enabled, the applicable content in FROM is transferred to RAM before starting standard operation.
If “FROM Macro execution at power-on” is enabled, Macro or Program Macro is automatically executed.

2.7 Timing Unit

Timing unit length varies between different modules. The timing unit length for each module display dot size is shown below.

	GT800X480A-C903PA		
Timing unit (Typ.) ± 5% ***IntTime***	16.2ms		

Timing unit affects the timing of the following commands and operations:

3.6.4.11	US s n	(Horizontal scroll speed)
3.6.4.27	US (a 02h t	(Short Wait)
3.6.4.28	US (a 11h p t1 t2 c	(Blink)
3.6.4.29	US (a A0h sXL sXH sYL sYH cL cH s	(Scroll display action XY)
3.6.4.30	US (a A2h v s pR pG pB	(Curtain display action XY)
3.6.4.31	US (a A3h v s pXL pXH pYL pYH	(Spring display action XY)
3.6.4.32	US (a A4h v s pXL pXH pYL pYH	(Random display action XY)
3.6.4.33	US (a A5h s pXL pXH pYH pYL	(Fade In display action XY)
3.6.4.34	US (a A6h s	(Fade Out display action XY)
3.6.4.65	US (e 12h a pL pH t1 t2 [d(1) ... d(p)]	(FROM Macro define / delete)
3.6.4.66	US ^ a t1 t2	(Macro execution)

2.8 Display Memory configuration

Display Memory size and configuration varies between different modules. For each module, the following module-specific values are referred to throughout this specification:

Item	Description
DispXdots	The number of dots (pixels) in the X-direction (horizontal) on the display screen.
DispYdots	The number of dots (pixels) in the Y-direction (vertical) on the display screen.
Xdots	The number of dots (pixels) in the X-direction (horizontal) for the entire Display Memory.
Ydots	The number of dots (pixels) in the Y-direction (vertical) for the entire Display Memory.
Max_Xdot	Valid X-coordinate values range from 0 to ***Max_Xdot***. [***Xdots*** - 1]
Max_Ydot	Valid Y-coordinate values range from 0 to ***Max_Ydot***. [***Ydots*** - 1]
Max_Xdot_CurtWin	Maximum valid X-coordinate value for current window.
Max_Ydot_CurtWin	Maximum valid Y-coordinate value for current window.
DispMemSize	Size of Display Memory in bytes. (number of bytes)
Max_DispMemAddr	Valid Display Memory addresses range from 0 to Max_DispMemAddr. [***DispMemSize*** - 1]

・The configuration for each module display dot (pixel) size is shown below.

Item	Module Type	GT800X480A-C903PA		
Display area [***DispXdots*** × ***DispYdots***]		800×480		
Total area [***Xdots*** × ***Ydots***]		1600×960		
Display Memory	***Xdots***	1600 (0640h)		
	Ydots	960 (03C0h)		
	Max_Xdot	1599 (063Fh)		
	Max_Ydot	959 (03BFh)		
	Max_Xdot_CurtWin	0 to 1599 * (0000h to 063Fh)		
	Max_Ydot_CurtWin	0 to 959 * (0000h to 03BFh)		
	DispMemSize [byte]	3,072,000 (2EE000h)		
	Max_DispMemAddr	3,071,999 (2EDFFFh)		

* Depends on size of current window.

3 Normal command mode

3.1 Displayable image types

3.1.1 Graphic display

Number of pixels:	800×480 pixels, 65K colors (16-bit color (RGB 5-5-5 + 1bit Intensity)).

3.1.2 Character display

Character mode:
1-byte character: 6×8, 8×16, 12×24, 16×32 pixel character mode
2-byte character: 16×16, 32×32 pixel character mode
Outline (Scalable) font: Any displayable size

Built-in Character font type:
1-byte character: 6×8, 8×16, 12×24, 16×32 pixel
– ANK, International font (specification DS-1600-0004-XX)
2-byte character: 16×16 pixel character
– Japanese Kanji (specification DS-906-0002-XX)
– Korean (specification DS-954-0008-XX)
– Simplified Chinese (specification DS-954-0006-XX)
– Traditional Chinese (specification DS-954-0007-XX)
2-byte character: 32×32 pixel character
– Japanese Kanji (specification DS-906-0003-XX)
Outline font (Pre-loaded): Source Han Sans (version 1.000)
– SourceHanSansCN-Normal.otf
– SourceHanSansJP-Normal.otf
– SourceHanSansKR-Normal.otf
– SourceHanSansTWHK-Normal.otf
(Refer to https://github.com/adobe-fonts/source-han-sans/)

Standard fonts:

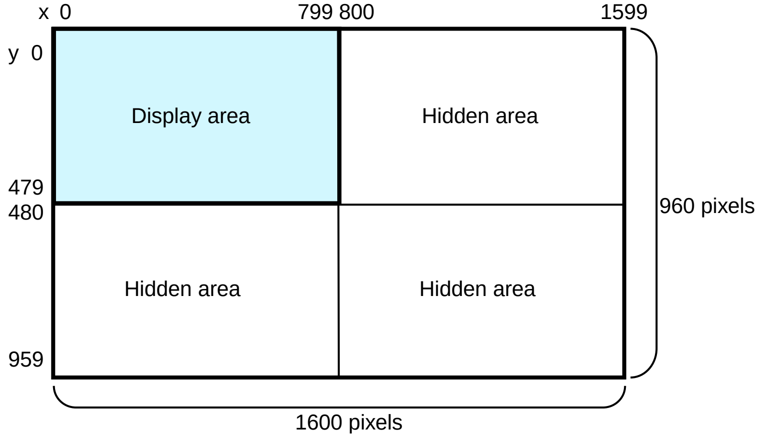
Font size	1-byte character	2-byte character				Outline Font
	International	Japanese	Korean	Simplified Chinese	Traditional Chinese	Source Han Sans
6×8	○	×	×	×	×	-
8×16	○	○ (16×16)	○ (16×16)	○ (16×16)	○ (16×16)	-
12×24	○	×	×	×	×	-
16×32	○	○ (32×32)	×	×	×	-
Any size	-	-	-	-	-	○

3.2 Memory

3.2.1 Display Memory

Display Memory is ***Xdots***×***Ydots*** – comprised of Display area (***DispXdots***×***DispYdots***) and Hidden areas.
By using “User Window” function, the memory area can be separated, and each separate window can be controlled independently (refer to 3.6.4.47 User Window define / cancel).
The “Display area” shown below, at position x=0, y=0, is the portion of the Display Memory that is visible on the display screen. Data in Hidden areas can be displayed using various display action commands.

[Example of GT800X480A-C903PA]



Memory address zero corresponds to the top left pixel of the display memory, and memory addresses increment (by 2 for each pixel) to the right, then continuing on the next lower row, as shown below.

Display memory address layout:

	x 0	1	-	799	800	801	-	1599
y 0	00.0000h	00.0002h	-	00.063Eh	00.0640h	00.0642h	-	00.0C7Fh
1	00.0C80h	00.0C82h	-	00.12BEh	00.12C0h	00.12C2h	-	00.18FEh
…	…	…	… Display Area	…	…	…	… Hidden Area	…
479	17.6380h	176382h	-	17.69BEh	17.69C0h	17.69C2h	-	17.6FFEh
480	17.7000h	17.7002h	-	17.763Eh	17.7640h	17.7642h	-	17.7C7Eh
481	17.7C80h	177C82h	-	17.82BEh	17.82C0h	17.82C2	-	17.88FEh
…	…	…	… Hidden Area	…	…	…	… Hidden Area	…
959	2E.D380h	2E.D382h	-	2E.D9BEh	2E.D9F0h	2E.D9F2h	-	2E.DFFEh

Each pixel is 16-bits (2 bytes), composed of 5-bits for each of Red, Green, and Blue, which are used as the upper 5 bits (b5…b1), and a single ‘I’ (Intensity) bit, which is used as a common least-significant bit (b0), as shown below. Note that memory addressing is little-endian.

Display memory pixel format (2 bytes per pixel):

MSB								LSB							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
I bit0	B bit5	B bit4	B bit3	B bit2	B bit1	G bit5	G bit4	G bit3	G bit2	G bit1	R bit5	R bit4	R bit3	R bit2	R bit1

I: Intensity bit. Used as common least-significant bit (bit 0) for each of R, G, B.

3.2.2 Bit image and font definition memory

Bit image definition

Arbitrary bit image data can be defined and saved using bit image definition commands.
RAM: 4,096 bytes
FROM1: 2,048K bytes
FROM2: 257,024K bytes (Note: Upper 27,648K used for standard built-in outline font)
Refer to 3.6.4.40 RAM bit image definition, 3.6.4.41 FROM bit image definition, and 3.6.4.70 FROM2 image store. Note: FROM2 can also be used as General-purpose memory.

User-defined fonts

Memory for arbitrary user-defined fonts is available as follows.

Download character
For each of the font sizes 6×8, 8×16, 12×24, and 16×32 pixel (1-byte character), and 16×16 and 32×32 pixel (2-byte character), a maximum of 16 characters can be defined to memory space in RAM.

FROM user font
For each of the font sizes 6×8, 8×16, 12×24, and 16×32 pixel (1-byte character), a maximum of 128 characters can be defined to memory space in FROM.

FROM extension font
A number of custom font tables, for 1-byte character codes, with font Y-size 8, 16, 24, or 32 pixels, can be defined to memory space in FROM.

Refer to 3.6.4.49 Download character definition, 3.6.4.51 16×16 Download character definition, 3.6.4.53 32×32 Download character definition, 3.6.4.57 FROM User Font definition, and 3.6.4.58 FROM extension font definition.

Outline (Scalable) font
User-supplied font files can be used (refer to 3.6.4.16).

User-defined fonts summary:

Font size	1-byte character			2-byte character
	Download character	FROM user font	FROM extension font	Download character
6×8	○	○	○	×
8×16	○	○	○	○ (16×16)
12×24	○	○	○	×
16×32	○	○	○	○ (32×32)

3.2.3 General-purpose memory

Arbitrary data can be stored to and retrieved from the memory.
General-purpose RAM: 1,024 bytes
General-purpose FROM: 4,096 bytes × 16 areas
General-purpose FROM2: 128K bytes × 2,008 areas

General-purpose RAM

000000h – 0003FFh

General-purpose FROM (1)

000000h – 000FFFh
001000h – 001FFFh
...
00F000h – 00FFFFh

General-purpose FROM2 (2)

00000000h – 0001FFFFh
00020000h – 0003FFFFh
...
0FAE0000h – 0FAFFFFFh

(1) FROM operation which would exceed a 4,096-byte memory area is not possible.
(2) FROM2 area from address 0E00.0000h is reserved for built-in outline fonts.

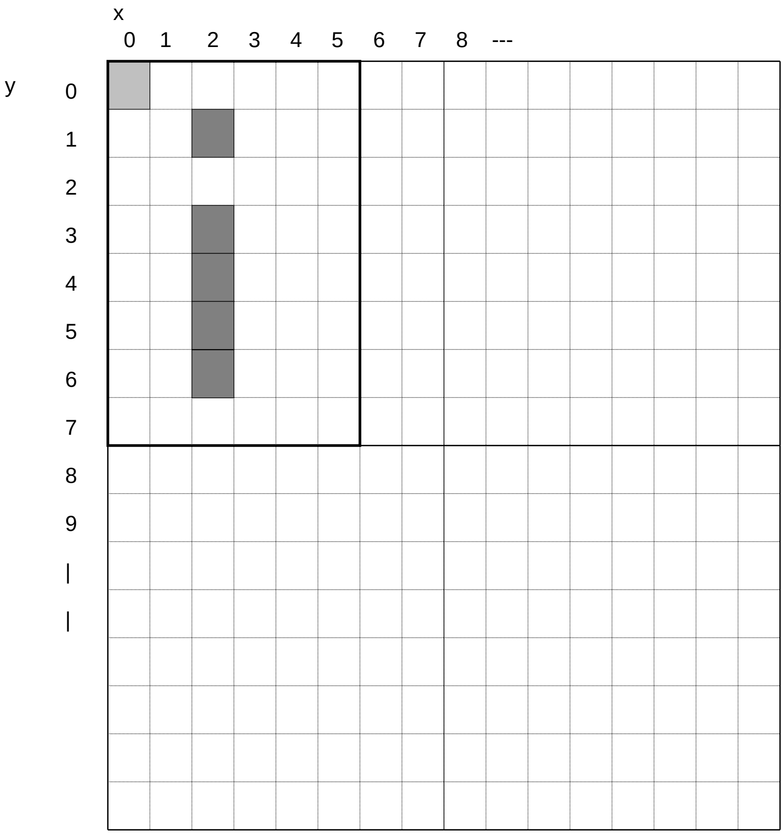
3.3 Cursor

Cursor indicates the write start position for displaying a character or image.

Characters and images are written to the right in the X direction and downwards in the Y direction from and including the Cursor position.

Cursor position can be moved by “Cursor set” command (refer to 3.6.4.3 Cursor set).

Cursor position relates to Display Memory as shown below.



Light grey:	Cursor
Dark grey:	Character
Thick line frame:	Space for one character (6×8 pixel)

3.4 Window

Window function enables the display screen to be divided into “windows” each of which can be controlled and displayed independently.

Display Memory is shared by all windows; individual windows do not have their own display memory.

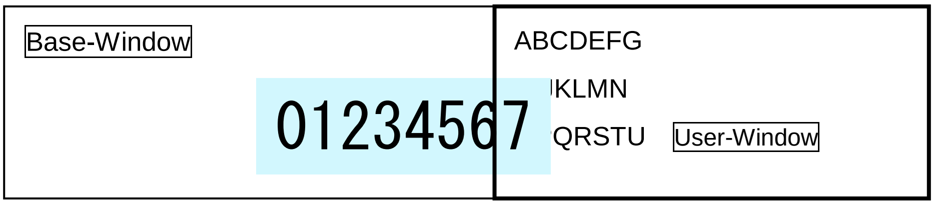
There are 2 types of “window”: Base-Window and User-Window.

Refer to 3.6.4.47 User Window define / cancel.

3.4.1 Base-Window

Base-Window covers the entire display screen. If no User-Windows are defined, all display operation is processed on this window. If one or more User-Windows are defined, display operation on any area not covered by a User-Window is done by selecting Base-Window. When Base-Window is selected, even if User-Window(s) are defined, all display operation is processed under Base-Window. Therefore the current display contents of User-Window(s) are overwritten.

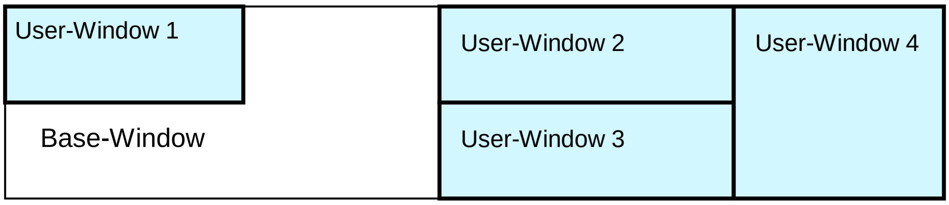
Operation on Base-Window depends on the setting of “Write screen mode” (refer to 3.5 Write screen mode).



3.4.2 User-Window

User-Window is defined by User-Window definition command. Display operation is processed on the window selected by Current Window select command.

A maximum of 4 User-Windows can be defined.



3.5 Write screen mode

This setting is only applicable for Base-Window.

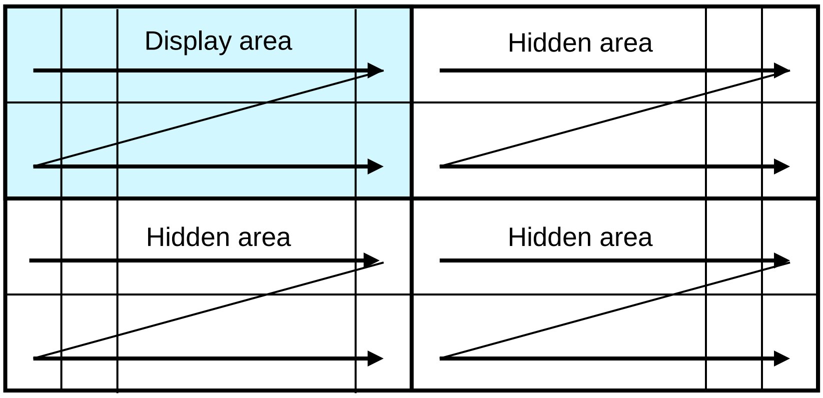
There are two Write screen modes, Display screen mode and All screen mode. The mode is set by command (refer to 3.6.4.4 Write screen mode select).

3.5.1 Display screen mode

When the cursor is located in the Display area, all operation will be done within Display area, and when cursor is located in a Hidden area, it will be done within the Hidden area.

Character write depends on the specified character display mode.

Bit images are written within the current area, and any data outside the area is ignored.

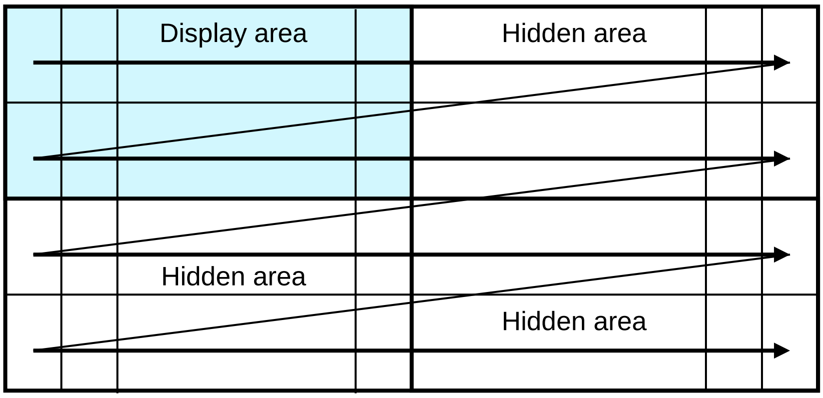


3.5.2 All screen mode

Regardless of the cursor position, operation will be done over the entire area.

Character write depends on the specified character display mode.

Bit images are written within the entire memory area, and any data outside the area is ignored.



3.6 Display control Commands

This section describes the operation of each command.
Within these explanations, Character (x-pixel) and Line (y-pixel) refer to the number of pixels determined by the "Font size select" and "Font magnification" settings, etc.
For commands that produce response data from the display, this data is placed in the send buffer, then transmitted. **When DSR=MARK (BUSY), data transmission is halted, and during any time when there is no space in the transmit buffer, command processing is halted. Caution is needed when using these commands via any unidirectional interface (transmit-data is queued in the transmit buffer, but is not sent from the module).**

3.6.1 Code set

3.6.1.1 Character code

Command Name	Hex Code	Operation	Page
Character display	20h – FFh or 2-byte character code (or 1-4 byte UTF-8 byte sequence)	Display character at the current cursor position.	p13

3.6.1.2 Control code

Command Name	Hex Code	Operation	Page
BS Back Space	08h	Cursor moves left by one character.	p17
HT Horizontal Tab	09h	Cursor moves right by one character.	p17
LF Line Feed	0Ah	Cursor moves down by one line.	p18
HOM Home Position	0Bh	Cursor moves to home position (top left).	p18
CR Carriage Return	0Dh	Cursor moves to left end of the current line.	p18
CLR Display Clear	0Ch	Display screen is cleared, cursor moves to home position.	p18
CAN Line Clear	18h	Current line is cleared and cursor moves to left end.	p18
RCLR Line end Clear	19h	Current line is cleared from cursor moves to right end.	p18

3.6.2 Detail of code set

3.6.2.1 Character display

Code: 20h – FFh or 2-byte character code (or 1-4 byte UTF-8 byte sequence)
Function: Display character at cursor position.

Font size can be selected, 6×8, 8×16, 12×24 or 16×32 (refer to 3.6.4.12 Font size select), or any displayable size for Outline fonts (refer to 3.6.4.15 Outline font size).
To display 2-bytes characters, the following settings are required:
Font size select = 8×16 pixel, 16×32 pixel or Outline font (m=02h, m=04h or 00h *)
2-byte character = ON (m=01h)
2-byte character type = Japanese, Korean, Simplified or Traditional Chinese
Refer to 3.6.4.13 2-byte character ON/OFF and 3.6.4.14 2-byte character type for details.
The 2-byte character code depends on the type of built-in character fonts. This module has the following built-in 2-byte character fonts:

Font type	**Code type**	**First byte**	**Second byte**
Japanese	JIS X0208 (Shift-JIS)	81h ≤ c1 ≤ 9Fh, E0h ≤ c1 ≤ Efh	40h ≤ c2 ≤ 7Eh, 80h ≤ c2 ≤ FCh
Korean	KSC5601-87	A1h ≤ c1 ≤ FEh	A1h ≤ c2 ≤ FEh
Simplified Chinese	GB2312-80	A1h ≤ c1 ≤ FEh	A1h ≤ c2 ≤ FEh
Traditional Chinese	Big-5	A1h ≤ c1 ≤ FEh	40h ≤ c2 ≤ 7Eh A1h ≤ c2 ≤ FEh

* Outline font type must be selected first (Refer to 3.6.4.17 Outline font type).

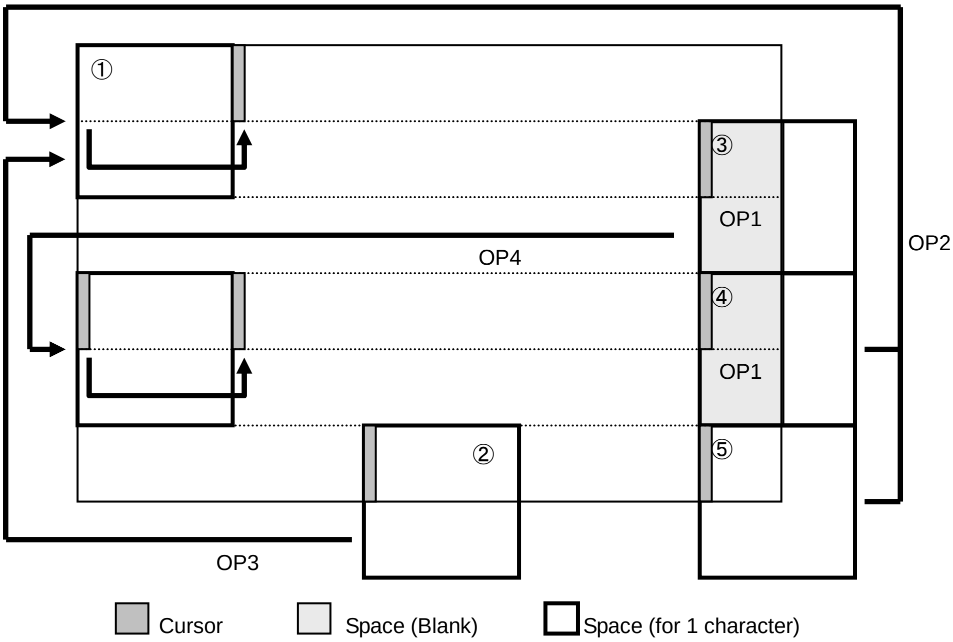
Alternatively, character codes can be entered in Unicode (UTF-8) format by setting Character Table type to FEh.

This command operates on the currently-selected window (refer to 3.6.4.46 Current window select). Regardless of the cursor position, if the character size (x and/or y) exceeds the window size, the command is ignored. Details of operation are as follows:

MD1 (Over-write mode)

Cursor position		Figure Number	Display Operation
X direction	Y direction		
Space for character on right side.	Space for character at current cursor position.	①	Display character on cursor. Horizontal Tab (HT).
	No space for character at current cursor position.	②	Cursor moves to the left end of top line (OP3). Display character at cursor. Horizontal Tab (HT).
No space for character on right side.	Space for character in next lower line.	③	Display space at cursor (OP1). Cursor moves to left end of next lower line (OP4). Display character at cursor. Horizontal Tab (HT).
	No space for character in next lower line.	④	Display space at cursor (OP1). Cursor moves to left end of top line (OP2). Display character at cursor. Horizontal Tab (HT).
	No space for character at current cursor position.	⑤	Cursor moves to left end of top line (OP2). Display character at cursor. Horizontal Tab (HT).

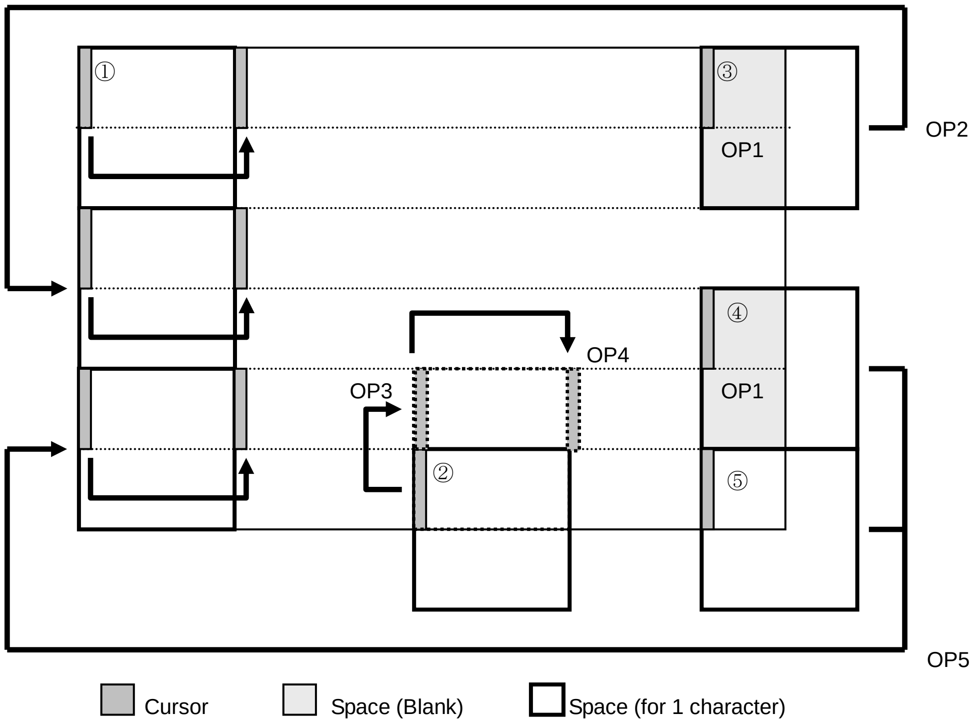
Note: HT operation depends on cursor position (refer to 3.6.2.3 Horizontal Tab).



MD2 (Vertical scroll mode)

Cursor position		Figure Number	Display Operation
X direction	Y direction		
Space for character on right side.	Space for character at current cursor position.	①	Display character at cursor. Horizontal Tab (HT) (OP4).
	No space for character at current cursor position.	②	Display contents are scrolled up the required number of pixels, and the bottom line is cleared. Cursor moves to the displayable upper position (OP3). Display character at cursor. Horizontal Tab (HT).
No space for character on right side.	Space for character in next lower line.	③	Display space at cursor (OP1). Cursor moves to left end of next lower line (OP2). Display character at cursor. Horizontal Tab (HT).
	No space for character in next lower line.	④	Display space at cursor (OP1). Display contents are scrolled up the required number of pixels, and the bottom line is cleared. Cursor moves to left end of bottom line (OP5). Display character at cursor. Horizontal Tab (HT).
	No space for character at current cursor position.	⑤	Display contents are scrolled up the required number of pixels, and the bottom line is cleared. Cursor moves to left end of bottom line (OP5). Display character at cursor. Horizontal Tab (HT).

Note: HT operation depends on cursor position (refer to 3.6.2.3 Horizontal Tab).



MD3 (Horizontal scroll mode)

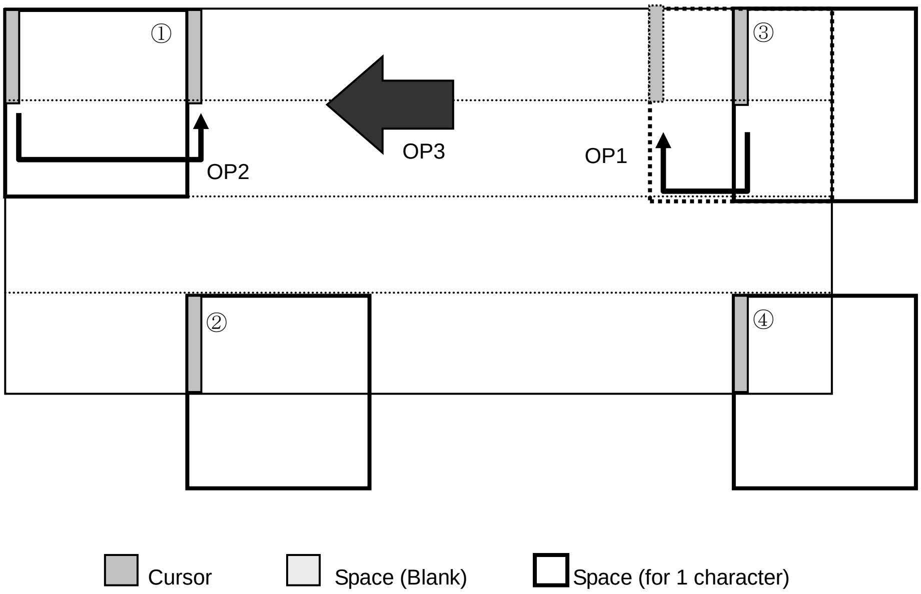
Cursor position			Figure Number	Display Operation
X direction		Y direction		
Space for character on right side.	Not right end.	-	①	Display character at cursor. Horizontal Tab (HT) (OP2).
	Right end (Refer Figure 2).		-	Display character at cursor. Shift to Scroll ON*.
	-	No space for character at current cursor position.	②	No action. Cursor does not move.
No space for character on right side.	-	-	③	Contents of current line scroll to left until sufficient space for character is available at the right end (OP3). Cursor moves to left edge of newly-created space (OP1). Display character at cursor. Shift to Scroll ON*.
	-	No space for character at current cursor position.	④	No action. Cursor does not move.

Note: HT operation depends on cursor position (refer to 3.6.2.3 Horizontal Tab).

* Note: Operation during “Scroll ON”:

Contents of current line scroll left until sufficient space for character is available at the right end, then character is displayed at cursor.

“Scroll ON” condition is cancelled by any command that moves the cursor except Character Display or Horizontal Tab.



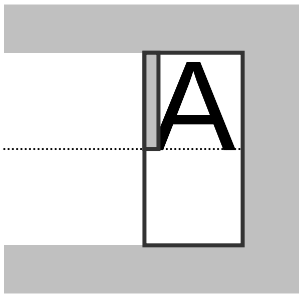


Figure 2

3.6.2.2 BS (Back Space)

Code: 08h

Function: Cursor moves to the left by one character.
This command has effect for the currently-selected window.

MD1 (Over-write mode) and MD2 (Vertical scroll mode)

Cursor position		**Display Operation**
X direction	**Y direction**	
Space for character on left side.	-	Cursor moves to left by one character.
No space for character on left end.	Space for one line above.	Cursor moves to right end of next upper line.
	No space for one line above.	Cursor does not move.

MD3 (Horizontal scroll mode)

Cursor position		**Display Operation**
X direction	**Y direction**	
Space for character on left side.	-	Cursor moves to left by one character.
No space for character on left end.	-	Cursor does not move.

3.6.2.3 HT (Horizontal Tab)

Code: 09h

Function: Cursor moves to the right by one character.
This command has effect for the currently-selected window.

MD1 (Over-write mode)

Cursor position		**Display Operation**
X direction	**Y direction**	
Space for character on right side.	-	Cursor moves to right by one character.
No space for character on right side.	Space for character in next lower line.	Cursor moves to left end of next lower line.
	No space for character in next lower line.	Cursor moves to left end of top line.

MD2 (Vertical scroll mode)

Cursor position		**Display Operation**
X direction	**Y direction**	
Space for character on right side.	-	Cursor moves to right by one character.
No space for character on right side.	Space for character in next lower line.	Cursor moves to left end of next lower line.
	No space for character in next lower line.	Display contents are scrolled up the required number of pixels, and the bottom line is cleared. Cursor moves to left end of bottom line.

MD3 (Horizontal scroll mode)

Cursor position			**Display Operation**
X direction		**Y direction**	
Space for character on right side.	Not right end	-	Cursor moves right by one character.
	Right end (refer to Figure 2, page 13).		Shift to Scroll ON*
No space for character on right side.	-	-	Contents on current line scroll left until sufficient space for character is available at the right end. Cursor moves to the left edge of newly-created space. Shift to Scroll ON*.

* Note: Operation during “Scroll ON”:
Contents of current line scroll left until sufficient space for character is available at the right end (cursor does not move).
“Scroll ON” condition is cancelled by any command that moves the cursor except Character Display or Horizontal Tab.

3.6.2.4 LF (Line Feed)

Code: 0Ah

Function: Cursor moves to next lower line.
This command has effect for the currently-selected window.

MD1 (Over-write mode)

Cursor position		Display Operation
X direction	Y direction	
-	Space for character in next lower line.	Cursor moves to the same position on next lower line.
	No space for character in next lower line.	Cursor moves to the same position on top line.

MD2 (Vertical scroll mode)

Cursor position		Display Operation
X direction	Y direction	
-	Space for character in next lower line.	Cursor moves to the same position on next lower line.
	No space for character in next lower line.	Display contents are scrolled up to the required number of pixels, and the bottom line is cleared. Cursor does not move.

MD3, MD5 (Horizontal scroll mode)

Cursor position		Display Operation
X direction	Y direction	
-	-	Cursor does not move.

3.6.2.5 HOM (Home Position)

Code: 0Bh

Function: Cursor moves to home position (top left).
This command has effect for the currently-selected window.

3.6.2.6 CR (Carriage Return)

Code: 0Dh

Function: Cursor moves to left end of current line.
This command has effect for the currently-selected window.

3.6.2.7 CLR (Display Clear)

Code: 0Ch

Function: Display screen is cleared and cursor moves to home position.
This command has effect for the currently-selected window.

3.6.2.8 CAN (Line Clear)

Code: 18h

Function: Current line is cleared and cursor moves to left end of current line.
This command has effect for the currently-selected window.

3.6.2.9 RCLR (Line end Clear)

Code: 19h

Function: Current line is cleared from cursor position to end of line (right end).
Cursor does not move.
This command has effect for the currently-selected window.

3.6.3 Command Set

3.6.3.1 General setting commands

Command Name	Hex Code	Operation	Page
Brightness level setting (for Backlight)	1Fh,58h,n **Default: n=FFh** or Memory SW setting.	Set backlight brightness level for entire display screen. Brightness level = (n / 255) * 100 [%]	p27
Initialize Display	1Bh,40h	Clear entire display screen and initialize all settings.	p27
Cursor set	1Fh,24h, xL,xH,yL,yH	Cursor moves to specified x,y position on Display Memory. xL: Cursor position x, lower byte. xH: Cursor position x, upper byte. yL: Cursor position y, lower byte. yH: Cursor position y, upper byte.	p27

Write screen mode	1Fh,28h,77h,10h,a
	Default a=00h or Memory SW setting.
		Sets the write screen mode for base window.
		a=00h: Display screen mode
		a=01h: All screen mode
			p27

3.6.3.2 Character display setting commands

Command Name	Hex Code	Operation	Page

International font set	1Bh,52h,n
	Default n=00h or Memory SW setting.
		Some character codes within the range 20h – 7Fh are selected from the types listed below.
		n=00h: America n=01h: France
		n=02h: Germany n=03h: England
		n=04h: Denmark 1 n=05h: Sweden
		n=06h: Italy n=07h: Spain1
		n=08h: Japan n=09h: Norway
		n=0Ah: Denmark2 n=0Bh: Spain2
		n=0Ch: Latin America n=0Dh: Korean
			p28

Character Table type	1Bh,74h,n
	Default n=00h or Memory SW setting.
		Character codes in the range 80h – FFh are selected from the types listed below.
		n=00h: PC437 (USA:Standard Europe)
		n=01h: Katakana,
		n=02h: PC850 (Multilingual)
		n=03h: PC860 (Portuguese)
		n=04h: PC863 (Canadian-French)
		n=05h: PC865 (Nordic),
		n=10h: WPC1252
		n=11h: PC866 (Cyrillic #2)
		n=12h: PC852 (Latin 2),
		n=13h: PC858
		n=FEh: UTF-8 input
		n=FFh: User table
			p28

Over-write mode	1Fh,01h	Set Over-write mode.	p29
Vertical scroll mode	1Fh,02h	Set Vertical scroll mode.	p29
Horizontal scroll mode	1Fh,03h	Set Horizontal scroll mode.	p29
Horizontal scroll mode, Scroll ON	1Fh,05h	Set Horizontal scroll mode, Scroll ON.	p29
Horizontal scroll speed	1Fh,73h,n **Default n=00h** or Memory SW setting.	Set Horizontal scroll speed.	p29

Font size select	1Fh,28h,67h,01h,m
	Default m=01h or Memory SW setting.
		Select font size of a character.
		m=00h: Outline font
		m=01h: 6×8 font
		m=02h: 8×16 font
		m=03h: 12×24 font
		m=04h: 16×32 font
			p29

2-byte character	1Fh,28h,67h,02h,m
	Default m=00h or Memory SW setting.
		Sets 2-byte character ON/OFF.
		m=01h: 2-byte character mode ON
		m=00h: 2-byte character mode OFF
			p29

2-byte character type	1Fh,28h,67h,03h,m
	Default: m=00h or Memory SW setting.
		Sets 2-byte character type.
		m=00h: Japanese
		m=01h: Korean
		m=02h: Simplified Chinese
		m=03h: Traditional Chinese
			p30

Command Name	Hex Code	Operation	Page

Font width	1Fh,28h,67h,04h,m
	Default: m=00h
		Character width select
		m=00h: Fixed width
		m=02h: Proportional 1
		m=03h: Proportional 2
		m=04h: Proportional 3
			p33

FROM extended font	1Fh,28h,67h,05h,n
	Default: n=00h
		FROM extended font select.
		n=00h: Normal font
		n=01h – FFh: FROM extended font
			p34

Font magnification	1Fh,28h,67h,40h, x,y **Default: x=01h, y=01h** or Memory SW setting.	Magnify the character by x times on the right, y times downward. x: X magnification factor y: Y magnification factor	p34

Character style	1Fh,28h,67h,41h,b
	Default: b=00h or Memory SW setting.
		Character style setting.
		b=00h: Normal
		b=01h: Bold
		b=02h: Shadow
		b=03h: Bordering
			p34

Character Color (Fore Color)	1Fh,28h,67h,50h,pR, pG,pB
	Default: pR=FFh, pG=FFh, pB=FFh
		Specify character color.
		pR = 00h–FFh: Red color level.
		pG = 00h–FFh: Green color level.
		pB = 00h–FFh: Blue color level.
			p34

Background Color	1Fh,28h,67h,51h,pR, pG,pB
	Default: pR=00h, pG=00h, pB=00h
		Specify character background color.
		pR = 00h–FFh: Red color level.
		pG = 00h–FFh: Green color level.
		pB = 00h–FFh: Blue color level.
			p34

Shadow and Bordering Color	1Fh,28h,67h,52h,pR, pG,pB
	Default: pR=00h, pG=00h, pB=00h
		Specify character shadow and bordering color.
		pR = 00h–FFh: Red color level.
		pG = 00h–FFh: Green color level.
		pB = 00h–FFh: Blue color level.
			p35

Background Color enable / disable	1Fh,28h,67h,58h,m
	Default m=00h
		Background Color enable / disable.
		m=01h: Enabled
		m=00h: Disabled
			p35

3.6.3.3 Display action setting commands

Command Name	Hex Code	Operation	Page
Wait	1Fh,28h,61h,01h,t	Processing is stopped for the specified time. t: Wait time (x approximately 0.47s)	p35
Short Wait	1Fh,28h,61h,02h,t	Processing is stopped for the specified time. t: Wait time (x ***IntTime***)	p35
Blink	1Fh,28h,61h,11h, p,t1,t2,c	Blink display action on display screen. p: Blink pattern t1: Normal display time t2: Blank display time c: Number of cycles	p36
Scroll display action XY	1Fh,28h,61h,A0h, sXL,sXH,sYL, sYH,cL,cH,s	Shifts the display screen, enabling horizontal and/or vertical display screen scroll action. sXL: Number of Display screen X shift (×1h) sXH: Number of Display screen X shift (×100h) sYL: Number of Display screen Y shift (×1h) sYH: Number of Display screen Y shift (×100h) cL: Number of cycles, lower byte cH: Number of cycles, upper byte s: Scroll speed	p36
Curtain display action	1Fh,28h,61h,A2h, v,s,pR,pG,pB	Curtain display action on display screen. v: Direction of Curtain action s: Curtain action speed pR: Red color level. pG: Green color level. pB: Blue color level.	p37
Spring display action XY	1Fh,28h,61h,A3h, v,s, pXL,pXH,pYL,pYH	Spring display action on display screen. v: Direction of spring action s: Spring action speed	p38

		pXL: Display memory X position (×1h) pXH: Display memory X position (×100h) pYL: Display memory Y position (×1h) pYH: Display memory Y position (×100h)	
Random display action XY	1Fh,28h,61h,A4h, v,s, pXL,pXH,pYL,pYH	Random display action on display screen. v: Random display action type s: Random display action speed pXL: Display memory X position (×1h) pXH: Display memory X position (×100h) pYL: Display memory Y position (×1h) pYH: Display memory Y position (×100h)	p39
Fade In display action XY	1Fh,28h,61h,A5h, s, pXL,pXH,pYL,pYH	Fade in display action on display screen. s: Fade in display action speed pXL: Display memory X position (×1h) pXH: Display memory X position (×100h) pYL: Display memory Y position (×1h) pYH: Display memory Y position (×100h)	p40
Fade Out display action XY	1Fh,28h,61h,A6h,s	Fade out display action on display screen. s: Fade out display action speed	p40

Display power ON/OFF	1Fh,28h,61h,40h,p
	Default: p=01h
		Controls display power ON/OFF
		p=01h: Display power ON
		p=00h: Display power OFF
			p40

3.6.3.4 Bit image display setting commands

Command Name	Hex Code	Operation	Page
Pixel drawing	1Fh,28h,64h,10h, pen, xL,xH,yL,yH	Display the pixel pattern on a drawing position or delete the pixel pattern already displayed. pen: Pixel display ON or OFF xL: Pixel pattern drawing position x, lower byte xH: Pixel pattern drawing position x, upper byte yL: Pixel pattern drawing position y, lower byte yH: Pixel pattern drawing position y, upper byte	p40
Line/Box pattern drawing	1Fh,28h,64h,11h, mode,pen, x1L,x1H,y1L,y1H, x2L,x2H,y2L,y2H	Display the Line, Box, Box FILL on the drawing area specified by x1,y1, x2,y2 or delete the pixel pattern already displayed. mode: Drawing mode select pen: Pixel ON or OFF x1L: Line/Box pattern drawing start position x1, lower byte x1H: Line/Box pattern drawing start position x1, upper byte y1L: Line/Box pattern drawing start position y1, lower byte y1H: Line/Box pattern drawing start position y1, upper byte x2L: Line/Box pattern drawing end position x2, lower byte x2H: Line/Box pattern drawing end position x2, upper byte y2L: Line/Box pattern drawing end position y2, lower byte y2H: Line/Box pattern drawing end position y2, upper byte	p41
Real-time bit image display	1Fh,28h,66h,11h, xL,xH,yL,yH,fmt, d(1)...d(n)	Display the supplied bit image data on the cursor position in real-time. xL: Bit image X size, lower byte (by 1 pixel) xH: Bit image X size, upper byte (by 1 pixel) yL: Bit image Y size, lower byte (by 1 pixel) yH: Bit image Y size, upper byte (by 1 pixel) fmt: Bit image format fmt=81h: Monochrome (1-bit) format fmt=86h: Color 6-bit format fmt=8Ch: Color 12-bit format fmt=90h: Color 16-bit format fmt=98h: Color 24-bit format fmt=F0h: BMP file format d(1)...d(n): Image data	p42
Packaged real-time bit image display	1Fh,28h,66h,21h, xL,xH,yL,yH,fmt, d(1)...d(n)	Display the supplied packaged bit image data on the cursor position in real-time. xL: Bit image X size, lower byte (by 1 pixel) xH: Bit image X size, upper byte (by 1 pixel) yL: Bit image Y size, lower byte (by 1 pixel) yH: Bit image Y size, upper byte (by 1 pixel) fmt: Bit image format fmt=86h: Color 6-bit format (packaged) fmt: 8Ch: Color 12-bit format (packaged) fmt: 98h: Color 24-bit format (packaged) d(1)...d(n): Packaged image data	p43

RAM bit image definition	1Fh,28h,66h,01h, aL,aH,aE,sL,sH,sE, d(1)...d(s)	Define user bit image to RAM. aL: Bit image data definition address, lower byte aH: Bit image data definition address, upper byte aE: Bit image data definition address, extension byte sL: Bit image data length, lower byte sH: Bit image data length, upper byte sE: Bit image data length, extension byte d(1)...d(s):Image data	p44
FROM bit image definition (Only valid in User setup mode)	1Fh,28h,65h,10h, aL,aH,aE,sL,sH,sE, d(1)...d(s)	Define user bit image to FROM1. aL: Bit image data definition address, lower byte aH: Bit image data definition address, upper byte aE: Bit image data definition address, extension byte sL: Bit image data length, lower byte sH: Bit image data length, upper byte sE: Bit image data length, extension byte d(1)...d(s):Image data	p45
Downloaded bit image display	1Fh,28h,66h,10h,m, aL,aH,aE,xSL,xSH, xL,xH,yL,yH, fmt	Display the RAM, FROM1, FROM2, or Display memory bit image defined on cursor position. m: Select bit image data memory aL: Bit image data definition address, lower byte aH: Bit image data definition address, upper byte aE: Bit image data definition address, extension byte xSL: Bit image defined X size, lower byte (by 1 pixel) xSH: Bit image defined X size, upper byte (by 1 pixel) xL: Bit image display X size, lower byte (by 1 pixel) xH: Bit image display X size, upper byte (by 1 pixel) yL: Bit image display Y size, lower byte (by 1 pixel) yH: Bit image display Y size, upper byte (by 1 pixel) fmt: Bit image format fmt=81h: Monochrome (1-bit) format fmt=86h: Color 6-bit format fmt=8Ch: Color 12-bit format fmt=90h: Color 16-bit format fmt=91h: Color 16-bit high-speed format fmt=98h: Color 24-bit format fmt=F0h: BMP file format	p46
Packaged Downloaded bit image display	1Fh,28h,66h,20h,m,aL,aH,aE,pL,pH, fmt	Display the packaged RAM, FROM1, or FROM2 bit image defined on cursor position. m: Select bit image data memory aL: Bit image data library address, lower byte aH: Bit image data library address, upper byte aE: Bit image data library address, extension byte pL: Bit image display number, lower byte pH: Bit image display number, upper byte fmt: Bit image format fmt=00h = Refer to Bit Image Index memory fmt=81h = Monochrome (1-bit) format fmt=86h = Color 6-bit format fmt=8Ch = Color 12-bit format fmt=98h = Color 24-bit format fmt=F0h = BMP file format	p48

3.6.3.5 General display setting commands

Command Name	Hex Code	Operation	Page

Reverse display	1Fh,72h,n
	
	Default: n=00h or Memory SW setting.
		Reverse display setting ON/OFF.
		n=00h: Reverse display OFF
		n=01h: Reverse display ON
			p50

Write mixture display mode	1Fh,77h,n **Default: n=10h** or Memory SW setting.	Sets the write mixture mode. New character or graphic image data is mixed with the current display image when written to the Display Memory. n=10h: Normal display write (Background pixels are written to display memory) n=11h: Thru write (Background pixels are skipped when writing to display memory)	p50

3.6.3.6 Window display setting commands

Command Name	Hex Code	Operation	Page

Current Window select	1Fh,28h,77h,01h,a	Selects current window
		a=00h: Base-Window
		a=01h: User-Window 1
		a=02h: User-Window 2
		a=03h: User-Window 3
		a=04h: User-Window 4
			p50

User Window define / cancel	1Fh,28h,77h,02h,a,b, xPL,xPH,yPL,yPH, xSL,xSH,ySL,ySH	Define or Cancel User-Window a: Definable window No. b:Define or Cancel xPL: Left position of window, lower byte (by 1 pixel) xPH: Left position of window, upper byte (by 1 pixel) yPL: Top position of window, lower byte (by 1 pixel) yPH: Top position of window, upper byte (by 1 pixel) xSL: X size of window, lower byte (by 1 pixel) xSH: X size of window, upper byte (by 1 pixel) ySL: Y size of window, lower byte (by 1 pixel) ySH: Y size of window, upper byte (by 1 pixel)	p51

3.6.3.7 Download character setting commands

Command Name	Hex Code	Operation	Page

Download character ON/OFF	1Bh,25h,n
	Default: n=00h
		Specify enable or disable for download character
		n=01h: Enable
		n=00h: Disable
			p52

Download character definition	1Bh,26h,a,c1,c2, x1,d1...dx1, ... , xk,d1...dxk	Define 6×8, 8×16, 12×24 or 16×32 pixel download characters into RAM. a: Select character type c1: Start character code c2: End character code x: Number of pixels for X-direction d1...dxk: Defined data	p52
Downloaded character delete	1Bh,3Fh,a,c	Delete defined 6×8, 8×16, 12×24 or 16×32 pixel download character. a: Select character type c: Delete Character code	p52
16×16 Download character definition	1Fh,28h,67h,10h, c1,c2,d1...d32	Defines the 16×16 download character in specified code. c1: Character code upper byte c2: Character code lower byte d: Definition data	p53
16×16 Downloaded character delete	1Fh,28h,67h,11h, c1,c2	Delete the 16×16 download character defined in the specified code. c1: Character code, upper byte c2: Character code, lower byte	p53
32×32 Download character Definition	1Fh,28h,67h,14h, c1,c2,d1...d128	Define 32×32 pixel download characters into RAM. c1: Character code, upper byte c2: Character code, lower byte d: Definition data	p53
32×32 Downloaded character Delete	1Fh,28h,67h,15h, c1,c2	Delete defined 32×32 pixel download character. c1: Delete character code, upper byte c2: Delete character code, lower byte	p54

Download character save
(Only valid in User setup mode)
	1Fh,28h,65h,11h,a	Save download characters defined on RAM to FROM.
		a: Font size
		a=01h: 6×8 pixel
		a=02h: 8×16 pixel
		a=03h: 16×16 pixel
		a=04h: 16×32 pixel
		a=05h: 32×32 pixel
		a=06h: 12×24 pixel
			p54

Download character restore	1Fh,28h,65h,21h,a	Transfer the download characters saved in FROM to RAM.
		a: Font size
		a=01h: 6×8 pixel
		a=02h: 8×16 pixel
		a=03h: 16×16 pixel
		a=04h: 16×32 pixel
		a=05h: 32×32 pixel
		a=06h: 12×24 pixel
			p54

FROM user font definition
(Only valid in User setup mode)
	1Fh,28h,65h,13h,m, P(80h-1),P(80h-2), ... ,P(FFh-n)	Define the user font for each size of 1-byte code to the user table.
		m: User table
		m=01h: 6×8 pixel
		m=02h: 8×16 pixel
		m=03h: 12×24 pixel
		m=04h: 16×32 pixel
		p: Definition data
			p54

FROM extension font definition	1Fh,28h,65h,15h,a,b, p(1), ... p(65536)	Define or delete FROM extension font a: Bank b: Define/Delete p: Definition data (if Define)	p55

3.6.3.8 User setup mode setting commands

Command Name	Hex Code	Operation	Page
User setup mode start	1Fh,28h,65h,01h, 49h,4Eh	User setup mode start.	p55
User setup mode end (Only valid in User setup mode)	1Fh,28h,65h,02h, 4Fh,55h,54h	User setup mode end.	p55

3.6.3.9 General-purpose I/O Port control commands

Command Name	Hex Code	Operation	Page

I/O Port Input / Output setting	1Fh,28h,70h,01h,n,a	Set input or output for general-purpose I/O ports.
		n: I/O port number
		n=00h: Port 0
		n=01h: Port 1
		a: Set Input / Output (bit-wise)
		bit = 0: Input, bit = 1: Output
			p56

I/O Port Output	1Fh,28h,70h,10h,n,a	Output data to general-purpose I/O port. n: I/O port number n=00h: Port 0 n=01h: Port 1 a: Output data value	p56

I/O Port Input	1Fh,28h,70h,20h,n	The state of a general-purpose I/O port is transmitted.
		n: I/O port number
		n=00h: Port 0
		n=01h: Port 1
			p56

3.6.3.10 Macro setting commands

Command Name	Hex Code	Operation	Page
RAM Macro define / delete	1Fh,3Ah,pL,pH, d1...dk	Define or delete RAM Macro or RAM Program Macro. pL: RAM Macro data length, lower byte pH: RAM Macro data length, upper byte d1...dk: RAM Macro data	p57
FROM Macro define / delete (Only valid in User setup mode)	1Fh,28h,65h,12h, a,pL,pH,t1,t2, d(1)...d(p)	Define or delete FROM Macro or FROM Program Macro. a: FROM Macro definition number pL: FROM Macro data length, lower byte pH: FROM Macro data length, upper byte t1: Display time interval t2: Idle time of macro repetition d(1)...d(p): FROM Macro data	p57

Macro execution	1Fh,5Eh,a,t1,t2	Continuously execute Macro.
		a: Macro processing definition number
		a=00h: RAM Macro
		a=01h–04h: FROM Macro 1–4
		a=80h: RAM Program Macro
		a=81h–84h: FROM Program Macro 1–4
		t1: Display time interval
		t2: Idle time for Macro repetition
			p58

Macro end condition	1Fh,28h,69h,20h, a,b,c **Default: a=00h, b=00h, c=00h** or Memory SW setting.	Macro end condition set. (Not applicable for Program Macro) a: Macro end code Enable/Disable b: Macro end code c: Macro end Clear Screen setting	p59

3.6.3.11 Other setting commands

Command Name	Hex Code	Operation	Page
Memory SW setting (Only valid in User setup mode)	1Fh,28h,65h,03h,a,b	Set Memory SW. a: Memory SW number b: Setting data	p59
Memory SW data send	1Fh,28h,65h,04h,a	Send the contents of Memory SW data. a: Memory SW number	p59
General-purpose memory store / FROM2 image store	1Fh,28h,65h,18h, sL,sH,sE,m1, aL,aH,aE, d[1]...d[s]	Store the supplied data into general-purpose memory. sL: Data size, lower byte sH: Data size, upper byte sE: Data size, extension byte m1: Memory select aL: Memory address, lower byte aH: Memory address, upper byte aE: Memory address, extension byte d: Data to store	p60
General-purpose memory transfer	1Fh,28h,65h,19h, sL,sH,sE, m1,a1L,a1H,a1E, m2,a2L,a2H,a2E	Transfer data between general-purpose memory areas. sL: Transfer size, lower byte sH: Transfer size, upper byte sE: Transfer size, extension byte m1: Destination memory select a1L: Destination address, lower byte a1H: Destination address, upper byte a1E: Destination address, extension byte m2: Source memory select a2L: Source address, lower byte a2H: Source address, upper byte a2E: Source address, extension byte	p61
General-purpose memory send	1Fh,28h,65h,28h, sL,sH,sE,m1, aL,aH,aE	Send data stored in general-purpose memory. sL: Data size, lower byte sH: Data size, upper byte sE: Data size, extension byte m1: Memory select aL: Memory address, lower byte aH: Memory address, upper byte aE: Memory address, extension byte	p61
Display status send	1Fh,28h,65h,40h, a,b,c	Send display status information a: Informarion name a=01h: Boot version information a=02h: Firmware version information a=10h: Kanji font information a=20h: Memory check sum information a=30h: Product information a=40h: Display x pixel information a=41h: Display y pixel information b: Start address c: Data length	p63
Memory re-write mode shift	1Ch,7Ch,4Dh, D0h,4Dh,4Fh,44h, 45h,49h,4Eh	Shift to “Memory re-write mode” from “Normal mode”.	p63

3.6.3.12 Touch Panel control commands

Command Name	Hex Code	Operation	Page
Touch Mode Selection: Single-Touch Mode/ Multi-Touch Mode	1Fh,50h,01h,n	Set the specified channel to Coordinates mode. n: Select Touch Mode and maximum simultaneous touch detection (for Multi-Touch Mode)	p72
Touch Panel Control Mode Coordinates mode	1Fh,50h,10h,ch,00h	Set the specified channel to Coordinates mode. ch: Channel	p72
Touch Panel Control Mode Switch Matrix mode	1Fh,50h,10h,ch,01h, nx,cx,ny,cy	Sets the specified channel to Switch Matrix mode. ch: Channel nx: Number of switch cx: Clearance ny: Number of switch cy: Clearance	p73
Touch Panel Control Mode Custom Switch mode	1Fh,50h,10h,ch,02h, sn, px1L, px1H, py1L, py1H, sx1L, sx1H, sy1L, sy1H, [...px(sn)L,px(sn)H, py(sn)L,py(sn)H, sx(sn)L,sx(sn)H , sy(sn)L,sy(sn)H]	Sets the specified channel to Custom Switch mode. ch: Channel md: Mode sn: Number of Switch buttons pxL: Switch button left horizontal position, lower byte pxH: Switch button left horizontal position, upper byte pyL: Switch button top vertical position, lower byte pyH: Switch button top vertical position, lower byte sxL: Switch button x-size, lower byte sxH: Switch button x-size, upper byte syL: Switch button y-size, lower byte syH: Switch button y-size, upper byte	p75
Touch Panel Data Transmit ON/OFF	1Fh,50h,20h,m	Sets whether or not touch operation data is transmitted. m: Transmit ON/OFF	p76
Touch Panel Channel Select	1Fh,50h,21h,ch	Selects the currently-active touch panel control channel. ch: Channel	p76

3.6.4 Command Set Details

3.6.4.1 US X n (Brightness level setting (for Backlight))

Code: 1Fh 58h n

n: Brightness level setting

Definable area: 00h ≤ n ≤ FFh

Default: n = FFh or Memory SW setting.

Function: Set display brightness level.

Brightness level = (n / 255) × 100 [%]

3.6.4.2 ESC @ (Initialize Display)

Code: 1Bh 40h

Function: Clear entire display screen and initialize all settings.

Settings return to default values.
Jumper settings are not reloaded.
Contents of receive buffer remain in memory.
General-purpose I/O port settings are unchanged.

3.6.4.3 US $ x y (Cursor set)

Code: 1Fh 24h xL xH yL yH

xL:	Cursor position x, lower byte (1 pixel / unit)
xH:	Cursor position x, upper byte (1 pixel / unit)
yL:	Cursor position y, lower byte (1 pixel / unit)
yH:	Cursor position y, upper byte (1 pixel / unit)

Definable area: 0000h ≤ (xL + xH×100h) ≤ ***Max_Xdot***
0000h ≤ (yL + yH×100h) ≤ ***Max_Ydot***

Function: Cursor moves to the specified (X, Y) position on Display Memory.

If the specified X, Y position (X and/or Y) is outside the definable area, the command is ignored and the cursor remains in the same position.
This command has effect for the currently-selected window.

3.6.4.4 US (w 10h a (Write screen mode select)

Code: 1Fh 28h 77h 10h a

a: Write screen mode

Definable area: 00h ≤ a ≤ 01h

a = 00h:	Display screen mode
a = 01h:	All screen mode

Default: a= 00h or Memory SW setting.

Function: Select the write screen mode.
This setting is only applicable for Base-Window.

Display screen mode: Display action is valid within area of either Display Area or Hidden Area, depending on cursor position.
All screen mode: Display action is valid over the entire display memory.

3.6.4.5 ESC R n (International font set)

Code: 1Bh 52h n
Definable area: 00h ≤ n ≤ 0Dh
Default: n = 00h or Memory SW setting.
Function: Select international font set.

Characters already displayed are not affected.

n	Font set
00h	America
01h	France
02h	Germany
03h	England
04h	Denmark 1
05h	Sweden
06h	Italy
07h	Spain1
08h	Japan
09h	Norway
0Ah	Denmark2
0Bh	Spain2
0Ch	Latin America
0Dh	Korea

Note: This setting is not used if UTF-8 input is selected (refer to 3.6.4.6 Character table type).

3.6.4.6 ESC t n (Character table type)

Code: 1Bh 74h n
Definable area: n = 00h, 01h, 02h, 03h, 04h, 05h, 10h, 11h, 12h, 13h, FEh,FFh
Default: n = 00h or Memory SW setting.
Function: Select Character table type.

Characters already displayed are not affected.
FFh (User table): User-defined font table (refer to 3.6.4.57 FROM User font definition command).

n	Font code type
00h	PC437(USA – Euro std)
01h	Katakana – Japanese
02h	PC850 (Multilingual)
03h	PC860 (Portuguese)
04h	PC863 (Canadian-French)
05h	PC865 (Nordic)
10h	WPC1252
11h	PC866 (Cyrillic #2)
12h	PC852 (Latin 2)
13h	PC858
FEh	UTF-8 input
FFh	User table

3.6.4.7 US MD1 (Over-write mode)

Code: 1Fh 01h
Function: Display mode set to Over-write mode.
This command has effect for the currently-selected window.

3.6.4.8 US MD2 (Vertical scroll mode)

Code: 1Fh 02h
Function: Display mode set to Vertical scroll mode.
This command has effect for the currently-selected window.

3.6.4.9 US MD3 (Horizontal scroll mode)

Code: 1Fh 03h
Function: Display mode set to Horizontal scroll mode.
This command has effect for the currently-selected window.

3.6.4.10 US MD5 (Horizontal scroll mode, Scroll ON)

Code: 1Fh 05h
Function: Display mode set to Horizontal scroll mode, scroll ON state.
After this command, operation is same as MD3 mode.
This command has effect for the currently-selected window.

3.6.4.11 US s n (Horizontal scroll speed)

Code: 1Fh 73h n
Definable area: 00h ≤ n ≤ 1Fh
Default: n = 00h or Memory SW setting.
Function: Set speed for Horizontal scroll mode.

Scroll speed is set by ‘n’.
Subsequent commands are not processed until scroll is completed.
Scroll base time period ‘T’ is ***IntTime***, but may be longer due to screen mode or character size, etc.

n	Speed
00h	Instantaneous
01h	T ms / 2 pixels
02h – 1Fh	(n-1) × T ms / pixel

Note: Scroll speed is approximate.
Depending on the scrolling area, scroll may reduce in speed or flicker.

3.6.4.12 US (g 01h m (Font size select)

Code: 1Fh 28h 67h 01h m
Definable area: m = 00h,01h, 02h, 03h, 04h
Default: m = 01h or Memory SW setting.
Function: Sets the font size for 1-byte characters.

m	Function
00h	Outline font*
01h	6×8 pixel character
02h	8×16 pixel character
03h	12×24 pixel character
04h	16×32 pixel character

*Outline font type must be selected first, or else setting is ignored (refer to 3.6.4.17).

3.6.4.13 US (g 02h m (2-byte character)

Code: 1Fh 28h 67h 02h m
Definable area: m = 00h, 01h
Default: m = 00h or Memory SW setting.

Function: Sets 2-byte character ON/OFF.

m	Function
00h	2-byte character mode OFF
01h	2-byte character mode ON

Note: This setting is not used if UTF-8 input is selected (refer to 3.6.4.6 Character table type).

3.6.4.14 US (g 03h m (2-byte character type)

Code: 1Fh 28h 67h 03h m
or 1Fh 28h 67h 0Fh m
Definable area: m = 00h, 01h, 02h, 03h
Default: m = 00h or Memory SW setting.
Function: Sets 2-byte character type.

m	Function	Code Type
00h	Japanese	JIS X0208 (Shift-JIS)
01h	Korean	KSC5601-87
02h	Simplified Chinese	GB2312-80
03h	Traditional Chinese	Big-5

To display a 16×16 pixel, 2-byte character:

Font size select:	Code:	1Fh 28h 67h 01h **02h**
2-byte character ON:	Code:	1Fh 28h 67h 02h **01h**
2-byte character type:	Code:	1Fh 28h 67h 03h 00h Japanese
		1Fh 28h 67h 03h 01h Korean
		1Fh 28h 67h 03h 02h Simplified Chinese
		1Fh 28h 67h 03h 03h Traditional Chinese
2-byte character code input:	Code:	88h 9Fh (“亜” Example Japanese character)

Note: This setting is not used if UTF-8 input is selected (refer to 3.6.4.6 Character table type).

3.6.4.15 US (g 06 ylsL ylsH ysL ysH xsL xsH boL boH (Outline font size)

Code: 1Fh 28h 67h 06h ylsL ylsH ysL ysH xsL xsH boL boH

ylsL:	Y-size (height) of character line (line spacing), in pixels (x1h)
ylsH:	Y-size (height) of character line (line spacing), in pixels (x100h)
ysL:	Nominal character Y-size (height), in pixels (x1h)
ysH:	Nominal character Y-size (height), in pixels (x100h)
xsL:	Nominal character X-size (width), in pixels (x1h)
xsH:	Nominal character X-size (width), in pixels (x100h)
boL:	Baseline Y-offset from cursor position, in pixels (x1h)
boH:	Baseline Y-offset from cursor position, in pixels (x100h)

Definable area: 0001h ≤ (ylsL + ylsH×100h) ≤ ***Ydots*** (other values: Default or exisiting setting)
0001h ≤ (ysL + ysH×100h) ≤ 03FFh (if higher: capped, if zero: use xs or auto)
0001h ≤ (xsL + xsH×100h) ≤ 03FFh (if higher: capped, if zero: use ys or auto)
FC00h ≤ (boL + boH×100h) ≤ 03FFh, 8000h (16-bit signed integer)
(if higher or lower: capped, if zero: auto, if 8000h: zero offset)

Default: yls = 64, ys, xs, bo = 0 (auto-calculated)

Function: Set character size and line spacing and alignment for outline (scalable) font display.

For displaying outline (scalable) font characters, the size and alignment is specified using four 16-bit parameters, yls, ys, xs, and bo.

‘yls’ is the most basic parameter, specifying the number of pixels in the y-direction for the region within which the characters will be written (and background cleared, if applicable). This is used as the y-line size (character size), when determining character display operation, and cursor behavior, etc, as described in section 3.6.2.1. When multiple lines of text are displayed, this value determines the (vertical) line pitch. This parameter value is capped (if necessary) at ***Ydots***. If set to zero, the previous setting (or default setting, 64) is used.

‘ys’ and ‘xs’ set the nominal character size in pixels for the y- and x-directions respectively. This is normally less than ‘yls’. If either parameter is zero, the value is set to the same as the other parameter. If both are zero, both are set to values automatically calculated with reference to ‘yls’. The actual size of displayed characters depends on the font design, so ‘ys’ and ‘xs’ are relative values.

‘bo’ sets the offset in pixels, from the cursor position (top, left-hand corner) to the baseline on which the outline font text is placed. This is a 16-bit signed value. Typically, positive values are used, which places the baseline below the cursor (negative values place the baseline above the cursor). If set to zero, the value is automatically calculated with reference to the other parameters to attempt to place the text vertically centered within the line. To explicitly set an offset value of zero, set ‘bo’ to 8000h.

3.6.4.16 US (g 07 ad0 ad1 ad2 ad3 sz0 sz1 sz2 sz3 (User-supplied font file address and size)

Code: 1Fh 28h 67h 07h ad0 ad1 ad2 ad3 sz0 sz1 sz2 sz3

ad3:ad2:ad1:ad0: Address of start of font file (binary mage) in General-purpose FROM2.
sz3:sz2:sz1:sz0: Size in bytes of font file (binary image) in General-purpose FROM2.

Definable area: 0000h ≤ (ad3:ad2:ad1:ad0) ≤ 0FAFFFFFh
0001h ≤ (sz3:sz2:sz1:sz0) ≤ 0FB00000h – (ad3:ad2:ad1:ad0)

Default: ad, sz = FFFFFFFFh (no font file specified)

Function: Set address and size in General-purpose FROM2 for the user-supplied font file (binary image).

Loading of the specified font file is attempted when FFh is selected as the outline font type (refer to 3.6.4.17).

3.6.4.17 US (g 08 n (Outline font type select)

Code: 1Fh 28h 67h 08h n

n: Outline font type

Definable area: n = 00h: Japanese
n = 01h: Korean
n = 02h: Simplified Chinese
n = 03h: Traditional Chinese
n = 80h: None (no outline font selected)
n = FFh: User-supplied font file (refer to 3.6.4.16).

Default: n = 80h

Function: Sets the outline font type.

If font loading fails (eg, due to incompatible or corrupted font file), outline font is set to “none”.

After selecting an outline font type, the Font size select command (refer to 3.6.4.12, m = 00h) can be used to enable display of outline characters.

Note: For n = 01h (Korean), a small number of symbol characters, available in the built-in pixel font, are not available in the built-in outline font.

3.6.4.18 US (g 04h m (Font Width)

Code: 1Fh 28h 67h 04h m

m: Font width setting

Definable area: m = 00h, 02h, 03h, 04h

Default: m = 00h or Memory SW setting.

Function: Sets the character width.

For fixed-width font, all characters are displayed as font size.
For proportional font, characters are displayed as:
Display width = right blank + character width + left blank
Note: This command is applicable for pixel character font only, not outline (scalable) font.

6×8 pixel character

m	Character width	Left blank pixels	Right blank pixels	Space (20h) character width
00h	Fixed-width	0	0	6
02h	Proportional 1	0	1	2
03h	Proportional 2	1	1	2
04h	Proportional 3	2	2	2

8×16 pixel character

m	Character width	Left blank pixels	Right blank pixels	Space (20h) character width
00h	Fixed-width	0	0	8
02h	Proportional 1	0	1	4
03h	Proportional 2	1	1	4
04h	Proportional 3	2	2	4

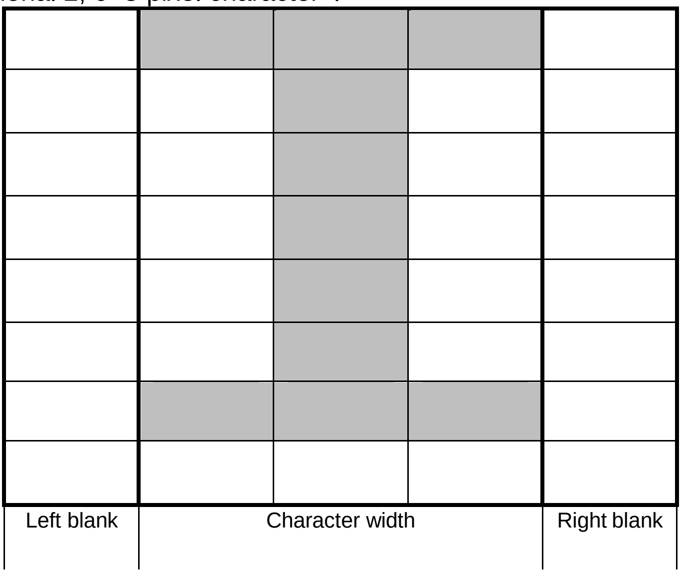
12×24 pixel character

m	Character width	Left blank pixels	Right blank pixels	Space (20h) character width
00h	Fixed-width	0	0	12
02h	Proportional 1	0	2	6
03h	Proportional 2	2	2	6
04h	Proportional 3	4	4	6

16×32 pixel character

m	Character width	Left blank pixels	Right blank pixels	Space (20h) character width
00h	Fixed-width	0	0	16
02h	Proportional 1	0	2	8
03h	Proportional 2	2	2	8
04h	Proportional 3	4	4	8

Example: Proportional 2, 6×8 pixel character ‘I’



3.6.4.19 US (g 05h n (FROM Extended Font)

Code: 1Fh 28h 67h 05h n

n: FROM Extended Font select

Definable area: 00h ≤ n ≤ FFh
n = 00h: Normal font.
n = 01h–FFh: FROM Extended Font (if FROM Extended Font is defined).

Default: n = 00h

Function: FROM extended font select.

3.6.4.20 US (g 40h x y (Font magnification)

Code: 1Fh 28h 67h 40h x y

x: X magnification factor
y: Y magnification factor

Definable area: 01h ≤ x ≤ 04h
01h ≤ y ≤ 04h

Default: x = 01h or Memory SW setting.
y = 01h or Memory SW setting.

Function: Sets character magnification ‘x’ times to the right and ‘y’ times downward.

Note: This command is applicable for pixel character font only, not outline (scalable) font.

3.6.4.21 US (g 41h b (Character style)

Code: 1Fh 28h 67h 41h b

b: Style

Definable area: 00h ≤ b ≤ 03h
b = 00h: Normal
b = 01h: Bold
b = 02h: Shadow
b = 03h: Bordering

Default: b = 00h or Memory SW setting.

Function: Character style setting *(Settings may reduce legibility)*.

Note: This command is applicable for pixel character font only, not outline (scalable) font.

3.6.4.22 US (g 50h pR pG pB (Character Color)

Code: 1Fh 28h 67h 50h pR pG pB

pB: Blue Brightness
pG: Green Brightness
pR: Red Brightness

Definable area: 00h ≤ pR, pG, pB ≤ FFh
00h: OFF
:
7Fh: 50%
:
FFh: Maximum

Default: pR = FFh, pG = FFh, pB = FFh

Function: Specifies the character color.

3.6.4.23 US (g 51h pR pG pB (Background Color)

Code: 1Fh 28h 67h 51h pR pG pB

pB: Blue Brightness
pG: Green Brightness
pR: Red Brightness

Definable area: 00h ≤ pR, pG, pB ≤ FFh
00h: OFF
:
7Fh: 50%
:
FFh: Maximum

Default: pR = 00h, pG = 00h, pB = 00h

Function: Specifies the character background color.

This setting is valid when Background Color is enabled (Refer to 3.6.4.25 Background Color enable/disable).

3.6.4.24 US (g 52h pR pG pB (Shadow and Bordering Color)

Code: 1Fh 28h 67h 52h pR pG pB

pB: Blue Brightness
pG: Green Brightness
pR: Red Brightness

Definable area: 00h ≤ pR, pG, pB ≤ FFh
00h: OFF
:
7Fh: 50%
:
FFh: Maximum

Default: bB = 00h, pG = 00h, pR = 00h

Function: Specifies the character shadow and bordering color.

This setting is valid when Shadow or Bordering is selected (Refer to 3.6.4.21 Character style).
Note: This command is applicable for pixel character font only, not outline (scalable) font.

3.6.4.25 US (g 58h b (Background Color enable / disable)

Code: 1Fh 28h 67h 58h b

b: Background enable/disable

Definable area: b = 00h, 01h
b = 00h: Background disabled
b = 01h: Background enabled

Default: b = 00h

Function: Background color for character display enabled or disabled.

Background Color disabled is functionally equivalent to Background Color enabled with Background Color = black (pR = 00h, pG = 00h, pB = 00h).

Background setting is not used if Write mixture display mode is set to “Thru write” mode (background is transparent in this case). For details, refer to 3.8 Color combination format.

3.6.4.26 US (a 01h t (Wait)

Code: 1Fh 28h 61h 01h t

t: Wait time

Definable area: 00h ≤ t ≤ FFh

Function: Waits for the specified time (command and data processing is stopped).

Wait time = t × approximately 0.47s
Command / data processing does not resume until wait time is completed.
It is possible to interrupt this command if the command is defined and run in a Macro.

3.6.4.27 US (a 02h t (Short Wait)

Code: 1Fh 28h 61h 02h t

t: Wait time

Definable area: 00h ≤ t ≤ FFh

Function: Waits for the specified time (command and data processing is stopped).

Wait time = t ×***IntTime*** (refer to 2.7 Timing Unit)
Command / data processing does not resume until wait time is completed.
It is possible to interrupt this command if the command is defined and run in a Macro.

3.6.4.28 US (a 11h p t1 t2 c (Blink)

Code: 1Fh 28h 61h 11h p t1 t2 c

p: Blink pattern
t1: Normal display time
t2: Blank display time
c: Number of cycles

Definable area: 00h ≤ p ≤ 02h
p = 00h: Normal display.
p = 01h: Blink display (alternately Normal and Blank display).
p = 02h: Reserved.
01h ≤ t1 ≤ FFh
01h ≤ t2 ≤ FFh
00h ≤ c ≤ FFh

Function: Blink display action Blink pattern specified by ‘p’.

Time specified by ‘t1’, ‘t2’.
A: t1 ×***IntTime*** Normal display
B: t2 ×***IntTime*** Blank display
Repeated ‘c’ times.

This command does not affect Display Memory.
c = 00h: Blink continues during subsequent command and data processing, until c=01h–FFh is set, or Initialize command.
c = 01h–FFh: Blink display is repeated 1–255 times while command and data processing is stopped. After display blinking is completed, Normal display returns and command and data processing resumes.
Command / data processing does not resume until operation is completed.
This command cannot be interrupted when running in a Macro.

3.6.4.29 US (a A0h sXL sXH sYL sYH cL cH s (Scroll display action XY)

Code: 1Fh 28h 61h A0h sXL sXH sYL sYH cL cH s

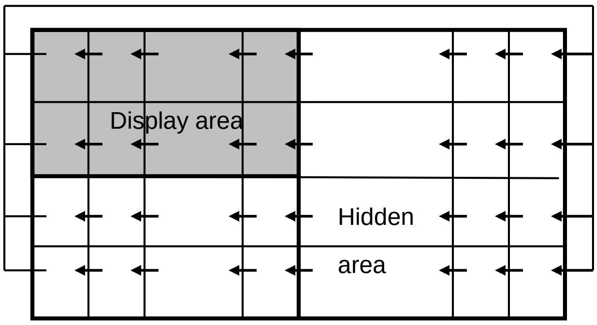
sXL: Number of Display screen X shift (x1h)
sXH: Number of Display screen X shift (x100h)
sYL: Number of Display screen Y shift (x1h)
sYH: Number of Display screen Y shift (x100h)
cL: Number of repetition lower byte
cH: Number of repetition upper byte
s: Scroll speed

Definable area: 0000h ≤ (sXL + sXH×100h) ≤ FFFFh
0000h ≤ (sYL + sYH×100h) ≤ FFFFh
0001h ≤ (cL + cH×100h) ≤ FFFFh
00h ≤ s ≤ FFh

Function: Shift the display screen.

Scroll speed is specified by ‘s’.
Scroll speed = s ×***IntTime*** / shift
Command / data processing does not resume until scroll is completed.
It is possible to interrupt this command if the command is defined and run in a Macro.

For example: 1-pixel scroll to the left: sXL=01h, sXH=00h, sYL=00h, sYH=00h



3.6.4.30 US (a A2h v s pR pG pB (Curtain display action XY)

Code: 1Fh 28h 61h A2h v s pR pG pB

v:	Direction of Curtain action
s:	Curtain action speed
pR:	Red Brightness
pG:	Green Brightness
pB:	Blue Brightness

Definable area: 00h ≤ v ≤ 03h

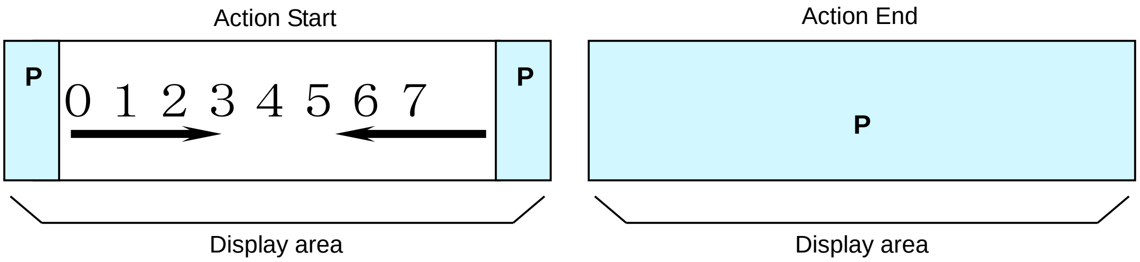
v = 00h: To the Right from the Left edge
v = 01h: To the Left from the Right edge
v = 02h: To the Left and Right separately from the Center
v = 03h: To the Center from Left edge and Right edge

00h ≤ s ≤ FFh
00h ≤ pR ≤ FFh
00h ≤ pG ≤ FFh
00h ≤ pB ≤ FFh

Function: Curtain display action on display screen.

Curtain action is displayed from the direction specified by ‘v’.
Curtain action speed = 60 × s ×***IntTime***
This command only affects the display area.
The non-display area memory is not affected.
Command / data processing does not resume until operation is completed.
If it is necessary to be able to cancel the display action during processing, this is possible if the command is defined and run in a Macro.

For examples: v = 03h



3.6.4.31 US (a A3h v s pXL pXH pYL pYH (Spring display action XY)

Code: 1Fh 28h 61h A3h v s pXL pXH pYL pYH

v:	Direction of spring action
s:	Spring action speed
pXL:	Display Memory X position (x1h)
pXH:	Display Memory X position (x100h)
pYL:	Display Memory Y position (x1h)
pYH:	Display Memory Y position (x100h)

Definable area: 00h ≤ v ≤ 03h

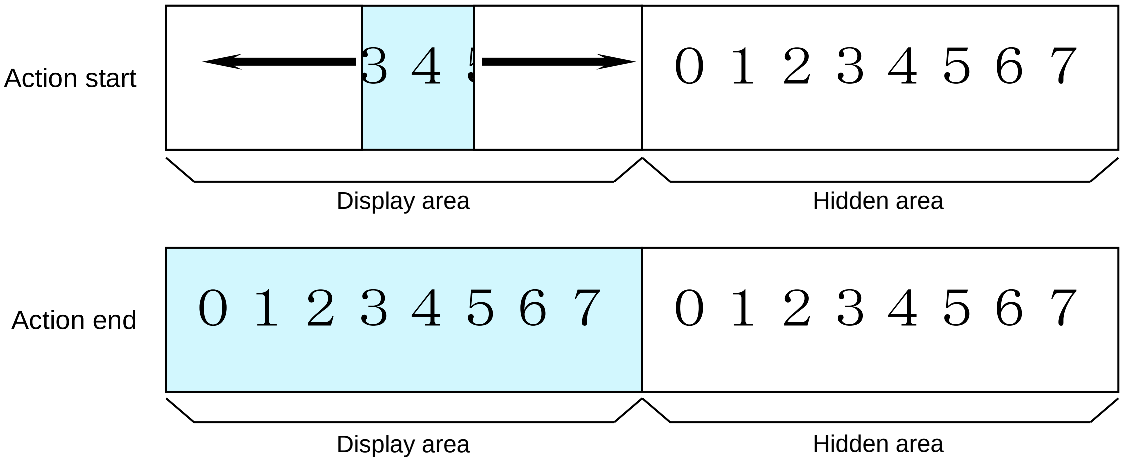
v=00h: To the Right from the Left edge.
v=01h: To the Left from the Right edge.
v=02h: To the Left and Right separately from the Center.
v=03h: To the Center from Left edge and Right edge.

00h ≤ s ≤ FFh
0000h ≤ (pXL + pXH×100h) ≤ ***Max_Xdot***
0000h ≤ (pYL + pYH×100h) ≤ ***Max_Ydot***

Function: Spring display action on display screen.

Pattern ‘p’ specified by Display Memory pattern address is displayed from the direction specified by ‘v’.
Spring action speed = 60 × s ×***IntTime***
This command affects only the display area, not hidden area.
Command / data processing does not resume until operation is completed.
If it is necessary to be able to cancel the display action during processing, this is possible if the command is defined and run in a Macro.

Example is shown below.



3.6.4.32 US (a A4h v s pXL pXH pYL pYH (Random display action XY)

Code: 1Fh 28h 61h A4h v s pXL pXH pYL pYH

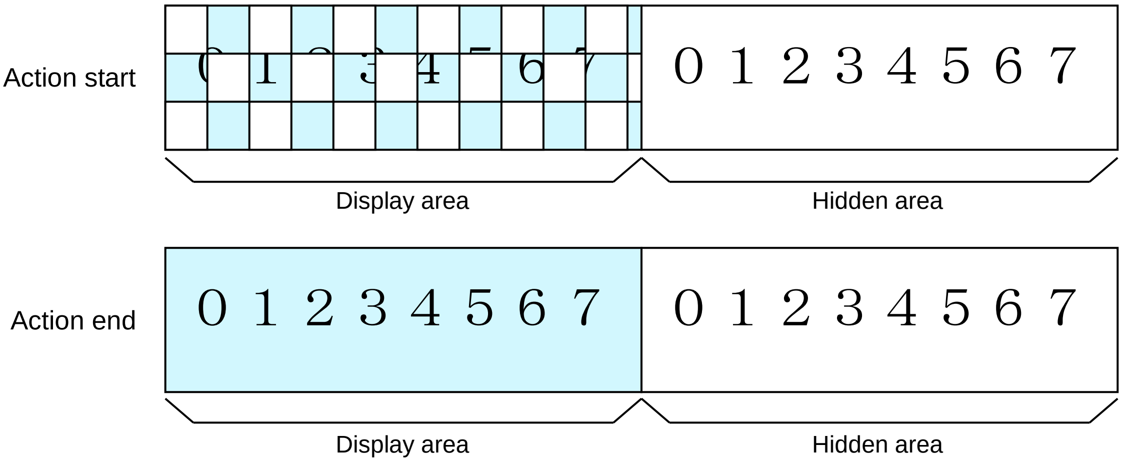
v: Random display action type
s: Random display action speed
pXL: Display Memory X position (×1h)
pXH: Display Memory X position (×100h)
pYL: Display Memory Y position (×1h)
pYH: Display Memory Y position (×100h)

Definable area: 00h ≤ v ≤ 02h
00h ≤ s ≤ FFh
0000h ≤ (pXL + pXH×100h) ≤ ***Max_Xdot***
0000h ≤ (pYL + pYH×100h) ≤ ***Max_Ydot***

Function: Random display action on display screen.

Pattern ‘p’ specified by Display Memory pattern address is displayed randomly, using one of three different transition patterns selected by ‘v’.
Random display action is completed in 16 steps, at approximately (s × 4 ×***IntTime***) / step.
This command affects only the display area, not hidden area.
Command / data processing does not resume until operation is completed.
If it is necessary to be able to cancel the display action during processing, this is possible if the command is defined and run in a Macro.

Example is shown below.



Transition patterns:

15	5	8	13
1	12	4	6
9	7	16	2
14	3	11	10

v=00h

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

v=01h

3	5	7	9
5	7	9	11
7	9	11	13
9	11	13	15

v=02h

Within each 16×16-pixel area on the display screen, sixteen 4×4-pixel cells are arranged in a grid. These transition to the new display pattern on the timing step (1 – 16) indicated by the number in that cell.

3.6.4.33 US (a A5h s pXL pXH pYH pYL (Fade In display action XY)

Code: 1Fh 28h 61h A5h s pXL pXH pYL pYH

s: Fade in display action speed
pXL: Display Memory X position (×1h)
pXH: Display Memory X position (×100h)
pYL: Display Memory Y position (×1h)
pYH: Display Memory Y position (×100h)

Definable area: 00h ≤ s ≤ FFh
0000h ≤ (pXL + pXH×100h) ≤ ***Max_Xdot***
0000h ≤ (pYL + pYH×100h) ≤ ***Max_Ydot***

Function: Fade in display action on display screen.

Pattern ‘p’ specified by Display Memory pattern address is displayed by fade in.
Fade In time = s × approximately 0.5s
This command affects only the display area, not hidden area.
Command / data processing does not resume until operation is completed.
If it is necessary to be able to cancel the display action during processing, this is possible if the command is defined and run in a Macro.

3.6.4.34 US (a A6h s (Fade Out display action XY)

Code: 1Fh 28h 61h A6h s

s: Fade out display action speed

Definable area: 00h ≤ s ≤ FFh

Function: Fade out display action on display screen.

Display area is cleared by fade out.
Fade Out time = s × approximately 0.5s
This command affects only the display area, not hidden area.
Command / data processing does not resume until operation is completed.
If it is necessary to be able to cancel the display action during processing, this is possible if the command is defined and run in a Macro.

3.6.4.35 US (a 40h p (Display power ON/OFF)

Code: 1Fh 28h 61h 40h p

p: Set display power ON/OFF

Definable area: 00h ≤ p ≤ 01h
p = 00h: Power OFF (Display OFF, Power save mode)
p = 01h: Power ON (Display ON)

Default: p = 01h

Function: Control Display power ON / OFF.

Display power ON/OFF setting applies until the next Display power or Initialize command, or power-off.

3.6.4.36 US (d 10h pen xL xH yL yH (Pixel drawing)

Code: 1Fh 28h 64h 10h pen xL xH yL yH

pen: Pixel display ON or OFF
xL: Pixel position x, lower byte
xH: Pixel position x, upper byte
yL: Pixel position y, lower byte
yH: Pixel position y, upper byte

Definable area: 00h ≤ pen ≤ 01h
pen = 00h: Pixel Display OFF
* Pen color is set to the background color (refer to 3.6.4.23).
pen = 01h: Pixel Display ON
* Pen color is set to the character color (refer to 3.6.4.22).
0000h ≤ (xL + xH×100h) ≤ ***Max_Xdot***
0000h ≤ (yL + yH×100h) ≤ ***Max_Ydot***

Function: Display the pixel at the position specified, or set pixel to background color.

This command has effect for the currently-selected window.
If Pixel display ON/OFF, or Pixel position is outside the definable area, the command is cancelled at that point and the following data is treated as standard data.

3.6.4.37 US (d 11h mode pen x1L x1H y1L y1H x2L x2H y2L y2H (Line/Box pattern drawing))

Code: 1Fh 28h 64h 11h mode pen x1L x1H y1L y1H x2L x2H y2L y2H

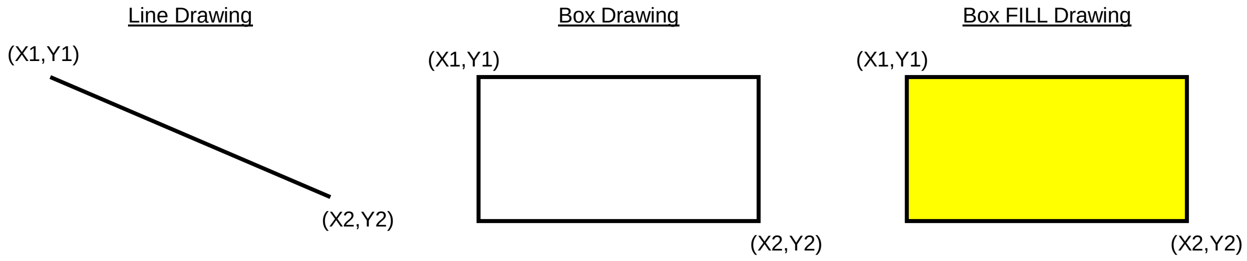
mode:	Line / Box / Box FILL select (Refer to illustration)
pen:	Pixel display ON or OFF
x1L:	Line/Box pattern drawing start position x, lower byte
x1H:	Line/Box pattern drawing start position x, upper byte
y1L:	Line/Box pattern drawing start position y, lower byte
y1H:	Line/Box pattern drawing start position y, upper byte
x2L:	Line/Box pattern drawing end position x, lower byte
x2H:	Line/Box pattern drawing end position x, upper byte
y2L:	Line/Box pattern drawing end position y, lower byte
y2H:	Line/Box pattern drawing end position y, upper byte

Definable area: 00h ≤ mode ≤ 02h
mode = 00h: Line
mode = 01h: Box
mode = 02h: Box FILL
00h ≤ pen ≤ 01h
pen = 00h: Line/Box Display OFF
* Pen color is set to the background color (refer to 3.6.4.23).
pen = 01h: Line/Box Display ON
* Pen color is set to the character color (refer to 3.6.4.22).
0000h ≤ (x1L + x1H×100h) ≤ ***Max_Xdot_CurtWin***
0000h ≤ (y1L + y1H×100h) ≤ ***Max_Ydot_CurtWin***
0000h ≤ (x2L + x2H×100h) ≤ ***Max_Xdot_CurtWin***
0000h ≤ (y2L + y2H×100h) ≤ ***Max_Ydot_CurtWin***

Function: Display a Line, Box, or Box FILL on the drawing area specified by (x1,y1)–(x2,y2) or set the pixels to the background color.

This command has effect for the currently-selected window.
If Display ON/OFF or Pixel pattern drawing position is outside the definable area, the command is cancelled at that point and the following data is treated as standard data.
If a diagonal line is specified, parts of the line may be 2 or more pixels in width.

Drawing mode



3.6.4.38 US (f 11h xL xH yL yH fmt d(1)...d(n) (Real-time bit image display)

Code: 1Fh 28h 66h 11h xL xH yL yH fmt d(1)...d(n)

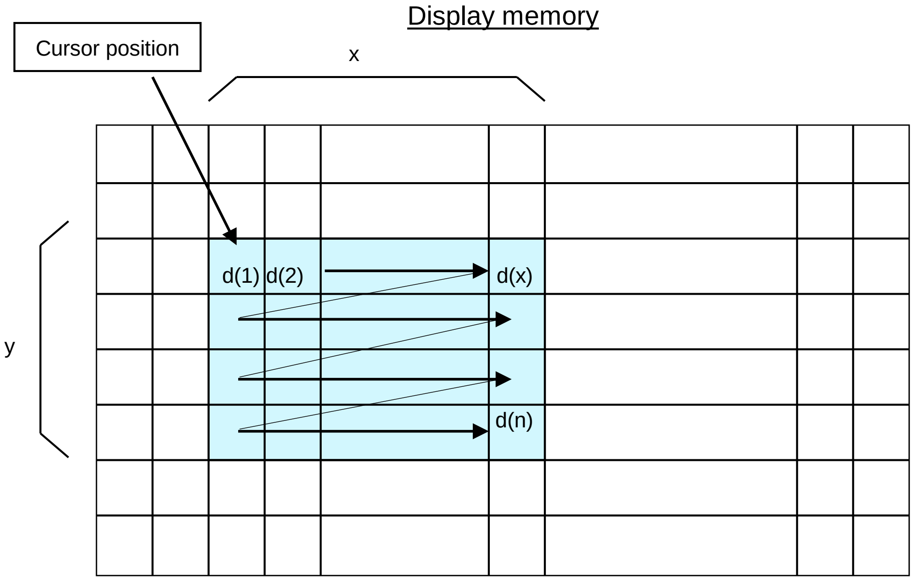
xL: Bit image X size, lower byte (by 1 pixel)
xH: Bit image X size, upper byte (by 1 pixel)
yL: Bit image Y size, lower byte (by 1 pixel)
yH: Bit image Y size, upper byte (by 1 pixel)
fmt: Image format
d(1) – d(n): Image data (see below)

Definable area: 0001h ≤ (xL + xH×100h) ≤ ***Xdots***
0001h ≤ (yL + yH×100h) ≤ ***Ydots***
fmt = 81h, 86h, 8Ch, 90h, 98h, F0h
fmt = 81h: Monochrome (1-bit) format*
*: 1 byte per 8 pixels or part thereof for each horizontal line.
(Refer to “3.7.1 Normal Bit image data format”)
fmt = 86h: Color 6-bit format
fmt = 8Ch: Color 12-bit format
fmt = 90h: Color 16-bit format
fmt = 98h: Color 24-bit format
fmt = F0h: BMP file format
00h ≤ d ≤ FFh (Refer to 3.7.1 Normal Bit image data format)

Function: Display the bit image data at the cursor position in real-time.
Cursor position does not change.

If bit image exceeds the bounds of the current window, only the portion within the currently-selected window is displayed.
If Display position or display size etc, are outside the definable area, the command is cancelled at the point where the error is detected, and the remaining data is treated as standard data.
Display the bit image data at the cursor position in real-time.
Note: For BMP file format, the image size specified in the file header is used. 1, 4, 8, 24, and 32 bit BMP files are supported. 16 bit BMP files are not supported.

Example:



3.6.4.39 US (f 21h xL xH yL yH fmt d(1)...d(n) (Packaged Real-time bit image display)

Code: **1Fh 28h 66h 21h xL xH yL yH fmt d(1)...d(n)**

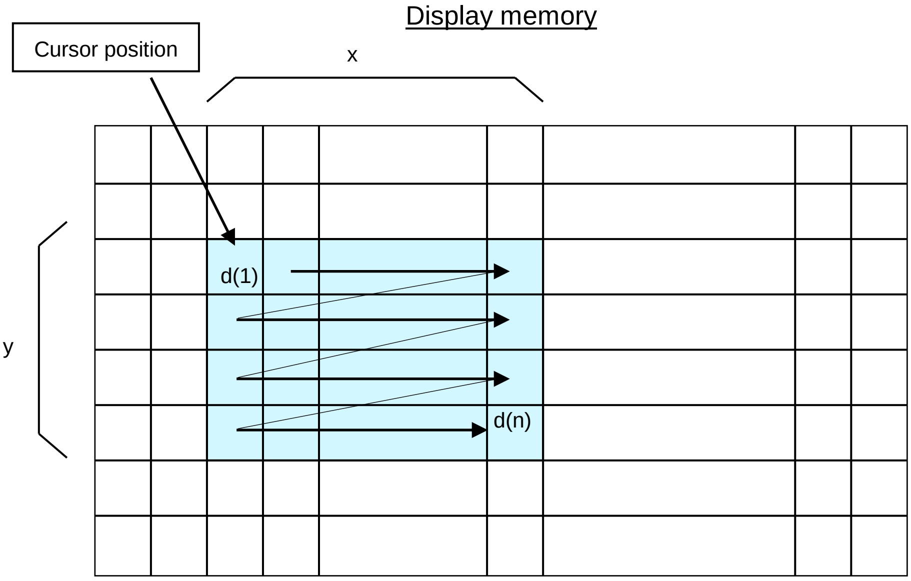
xL: Bit image X size, lower byte (by 1 pixel)
xH: Bit image X size, upper byte (by 1 pixel)
yL: Bit image Y size, lower byte (by 1 pixel)
yH: Bit image Y size, upper byte (by 1 pixel)
fmt: Image format
d(1) – d(n): Image data (Refer to figure as follows)

Definable area: 0001h ≤ (xL + xH×100h) ≤ ***Xdots***
0001h ≤ (yL + yH×100h) ≤ ***Ydots***
fmt = 86h, 8Ch, 98h
fmt = 86h: Color 6-bit format
fmt = 8Ch: Color 12-bit format
fmt = 98h: Color 24-bit format
00h ≤ d ≤ FFh (Refer to 3.7.2 Packaged Bit image data format)

Function: Display the bit image data at the cursor position in real-time.
Cursor position does not change.

If bit image exceeds the bounds of the current window, only the portion within the currently-selected window is displayed.
If Display position or display size etc, are outside the definable area, the command is cancelled at the point where the error is detected, and the remaining data is treated as standard data.

Example:



3.6.4.40 US (f 01h aL aH aE sL sH sE d(1)…d(s) (RAM bit image definition)

Code: 1Fh 28h 66h 01h aL aH aE sL sH sE d(1)...d(s)

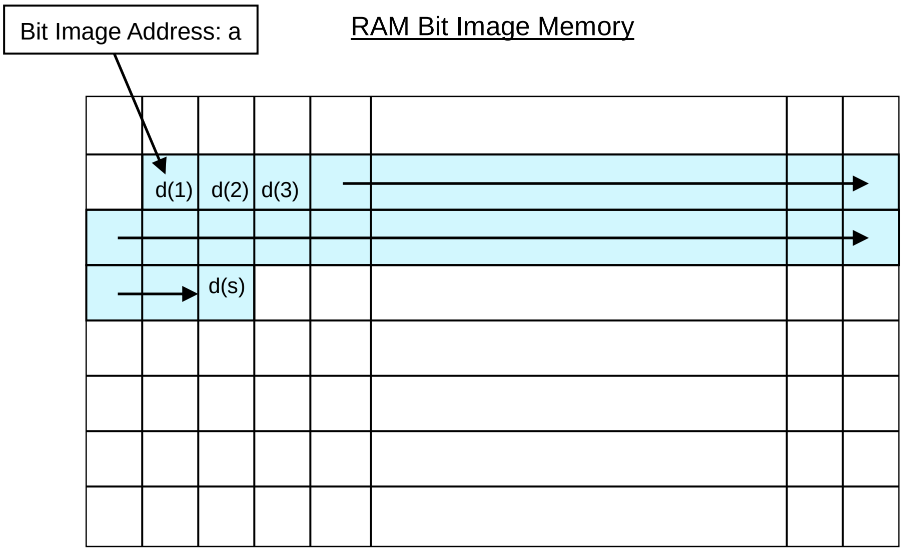
aL: Bit image data definition address, lower byte
aH: Bit image data definition address, upper byte
aE: Bit image data definition address, extension byte
sL: Bit image data length, lower byte
sH: Bit image data length, upper byte
sE: Bit image data length, extension byte
d(1) – d(s): Image data (see below)

Definable area: 000000h ≤ (aL + aH×100h + aE×10000h) ≤ 000FFFh
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 001000h
00h ≤ d ≤ FFh

Function: Define user bit image to the RAM.

RAM bit image capacity is 4,096 bytes.
Bit image data at arbitrary addresses can be defined or changed by appropriately setting Bit image data definition address and Bit image data length.
Bit images defined in RAM can be displayed using 3.6.4.42 Downloaded bit image display command.
If Bit image data definition address or Bit image data length is outside the definable area, the command is cancelled at that point, and the remaining data is treated as standard data.

Example:



3.6.4.41 US (e 10h aL aH aE sL sH sE d(1)...d(s) (FROM bit image definition)

Code: 1Fh 28h 65h 10h aL aH aE sL sH sE d(1)...d(s)

n:	Classifies the command
aL:	Bit image data definition address, lower byte (bit 0 ignored)
aH:	Bit image data definition address, upper byte
aE:	Bit image data definition address, extension byte
sL:	Bit image data length, lower byte (bit 0 ignored)
sH:	Bit image data length, upper byte
sE:	Bit image data length, extension byte
d(1) – d(s):	Image data (see below)

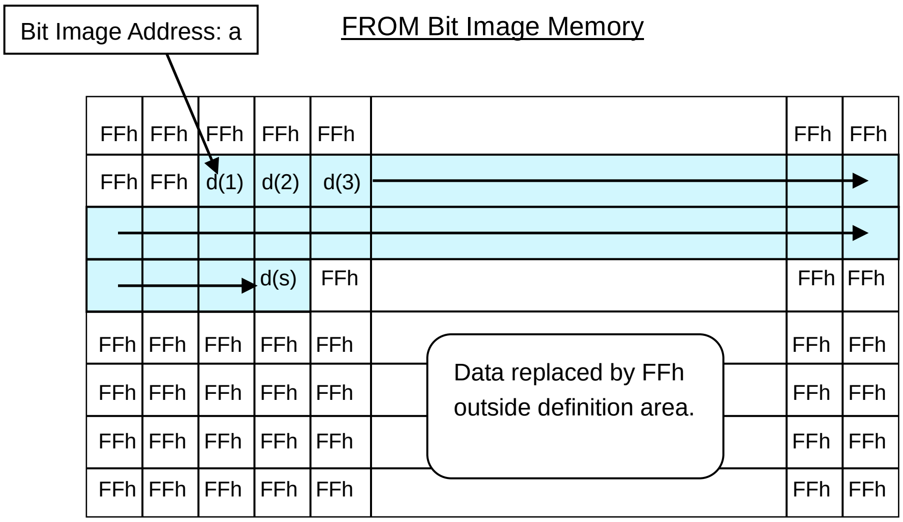
Definable area: aE = 00h–1Fh
000000h ≤ ((aL & FEh) + aH×100h + aE×10000h) ≤ 1FFFFEh
000002h ≤ ((sL & FEh) + sH×100h + sE×10000h) ≤ 200000h
00h ≤ d ≤ FFh

Function: Define user bit image to the FROM1.

FROM1 bit image capacity is 2048Kbytes.
Data is defined or changed from the specified Bit image data definition address for the number of bytes specified by Bit image data length.
Bit images defined in FROM1 can be displayed using 3.6.4.42 Downloaded bit image display command.
The least significant bit for both Bit image data definition address and Bit image data length is ignored – these are processed as even values.
If Bit image data definition address or Bit image data length is outside the definable area, the command is cancelled at that point, and the remaining data is treated as standard data.
This command is only valid in User setup mode.

Total definable area is 000000h to 1FFFFFh (2,097,152 bytes). Bit image definition is performed in units of 131,072 bytes (128KB). For example, if 10KB of bit image data is defined, the remaining 118KB data is set to FFh.
Bit image data definition address cannot be specified in such a way that the definition data would overflow the FROM1 area. The command is cancelled if this situation is detected, and the remaining data is treated as standard data.
Defined contents are not guaranteed if an error occurs.
To store image data in FROM2, refer to 3.6.4.70 General-purpose memory store / FROM2 image store.

Note that the first 2MB of FROM2 overlaps the FROM1 Bit Image memory; changing the content of either memory will affect the other.



3.6.4.42 US (f 10h m aL aH aE xSL xSH xL xH yL yH fmt (Downloaded bit image display)

Code: 1Fh 28h 66h 10h m aL aH aE xSL xSH xL xH yL yH fmt

m:	Select bit image data memory
aL:	Bit image data definition address, lower byte
aH:	Bit image data definition address, upper byte
aE:	Bit image data definition address, extension byte
xSL:	Bit image defined X size, lower byte (by 1 pixel)
xSH:	Bit image defined X size, upper byte (by 1 pixel)
xL:	Bit image display X size, lower byte (by 1 pixel)
xH:	Bit image display X size, upper byte (by 1 pixel)
yL:	Bit image display Y size, lower byte (by 1 pixel)
yH:	Bit image display Y size, upper byte (by 1 pixel)
fmt:	Image format

Definable area: 00h ≤ m ≤ 02h, 10h ≤ m ≤ 1Fh

m = 00h:	RAM bit image
m = 01h:	FROM1 bit image
m = 02h:	Display memory
m = 10h:	FROM2 base address 0000.0000h
m = 11h:	FROM2 base address 0100.0000h
m = 12h:	FROM2 base address 0200.0000h
m = 13h:	FROM2 base address 0300.0000h
m = 14h:	FROM2 base address 0400.0000h
m = 15h:	FROM2 base address 0500.0000h
m = 16h:	FROM2 base address 0600.0000h
m = 17h:	FROM2 base address 0700.0000h
m = 18h:	FROM2 base address 0800.0000h
m = 19h:	FROM2 base address 0900.0000h
m = 1Ah:	FROM2 base address 0A00.0000h
m = 1Bh:	FROM2 base address 0B00.0000h
m = 1Ch:	FROM2 base address 0C00.0000h
m = 1Dh:	FROM2 base address 0D00.0000h
m = 1Eh:	FROM2 base address 0E00.0000h
m = 1Fh:	FROM2 base address 0F00.0000h

When RAM bit image is selected:

000000h ≤ (aL + aH×100h + aE×10000h) ≤ 000FFFh

When FROM1 bit image is selected:

000000h ≤ (aL + aH×100h + aE×10000h) ≤ 1FFFFFh

When Display memory is selected:

000000h ≤ (aL + aH×100h + aE×10000h) ≤ 176FFFh

When FROM2 is selected:

000000h ≤ (aL + aH×100h + aE×10000h) ≤ FFFFFFh (m=10h–1Eh)
000000h ≤ (aL + aH×100h + aE×10000h) ≤ AFFFFFh (m=1Fh)

0000h ≤ (xSL + xSH×100h) ≤ FFFFh
0001h ≤ (xL + xH×100h) ≤ ***Xdots***
0001h ≤ (yL + yH×100h) ≤ ***Ydots***
fmt = 81h, 86h, 8Ch, 90h, 91h, 98h, F0h

fmt = 81h: Monochrome (1-bit) format
fmt = 86h: Color 6-bit format
fmt = 8Ch: Color 12-bit format
fmt = 90h: Color 16-bit format
fmt = 91h: Color 16-bit high-speed format
fmt = 98h: Color 24-bit format
fmt = F0h: BMP file format

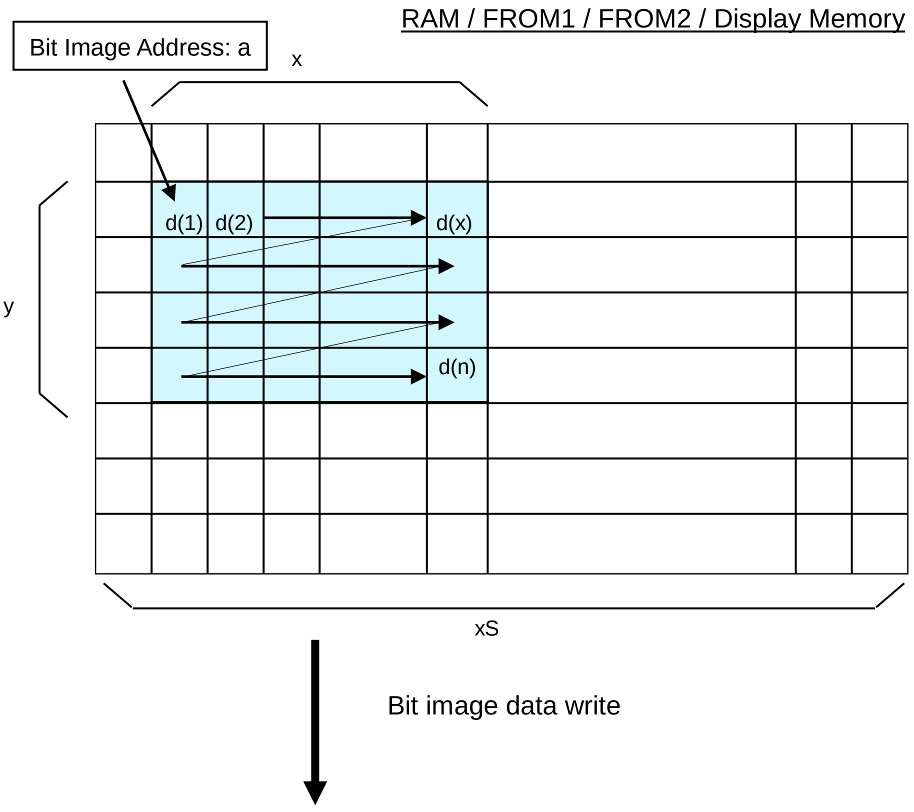
Function: Display, at the cursor position, the bit image defined in RAM, FROM1, FROM2, or in Display Memory.
Cursor position does not change.

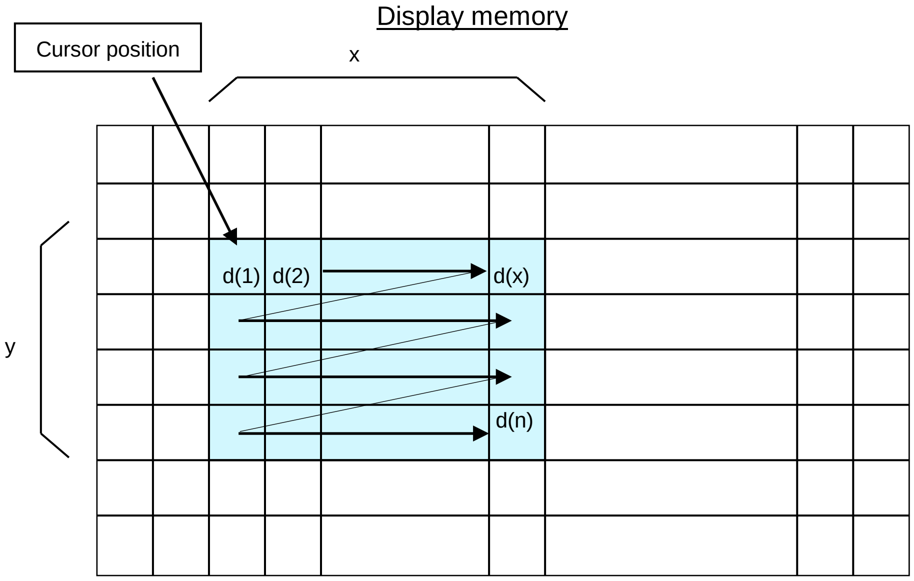
Select RAM, FROM1, FROM2 or Display Memory bit image by Select Bit image data memory ‘m’.
Set Bit image defined X size to the same X size of the bit image defined in memory.
A portion of the Defined bit image can be displayed by setting Bit image display X size less than Defined bit image X size, or by changing Bit image display Y size and/or Bit image data definition address.
If the bit image extends beyond the currently-selected window, only the portion within the current window is displayed.
When the image is being written to the Display Memory, if the image memory area is exceeded,

undefined data is displayed.

Note: To use m=02h (read from Display memory) to copy images within the Display Memory, use fmt=98h, Bit image defined X size = ***Xdots***, and data definition address = X + (Y × ***Xdots***), where (X,Y) is the start (top, left) point of the image.
Note: For BMP file format, the image size specified in the file header is used. 1, 4, 8, 24, and 32 bit BMP files are supported. 16 bit BMP files are not supported.
Note for FROM1 memory: Bit image data can be read from 000000h to 1FFFFFh continuously (bit image display can cross a 128KB block boundary).

For fmt=91h (Color 16-bit high-speed format), data is transferred directly to display memory, so Reverse and Write mixture settings are not used.





3.6.4.43 US (f 20h m aL aH aE pL pH fmt (Packaged Downloaded bit image display)

Code: **1Fh 28h 66h 20h m aL aH aE pL pH fmt**

m:	Select bit image data display memory
aL:	Bit image index start address, lower byte
aH:	Bit image index start address, upper byte
aE:	Bit image index start address, extension byte
pL:	Bit image number, lower byte
pH:	Bit image number, upper byte
fmt:	Image format

Definable area: 00h ≤ m ≤ 01h, 10h ≤ m ≤ 1Fh

m = 00h:	RAM bit image
m = 01h:	FROM1 bit image
m = 10h:	FROM2 base address 0000.0000h
m = 11h:	FROM2 base address 0100.0000h
m = 12h:	FROM2 base address 0200.0000h
m = 13h:	FROM2 base address 0300.0000h
m = 14h:	FROM2 base address 0400.0000h
m = 15h:	FROM2 base address 0500.0000h
m = 16h:	FROM2 base address 0600.0000h
m = 17h:	FROM2 base address 0700.0000h
m = 18h:	FROM2 base address 0800.0000h
m = 19h:	FROM2 base address 0900.0000h
m = 1Ah:	FROM2 base address 0A00.0000h
m = 1Bh:	FROM2 base address 0B00.0000h
m = 1Ch:	FROM2 base address 0C00.0000h
m = 1Dh:	FROM2 base address 0D00.0000h
m = 1Eh:	FROM2 base address 0E00.0000h
m = 1Fh:	FROM2 base address 0F00.0000h

When RAM bit image is selected;

000000h ≤ (aL + aH×100h + aE×10000h) ≤ 000FFFh

When FROM1 bit image is selected;

000000h ≤ (aL + aH×100h + aE×10000h) ≤ 1FFFFFh

When FROM2 is selected;

000000h ≤ (aL + aH×100h + aE×10000h) ≤ FFFFFFh (m=10h–1Eh)
000000h ≤ (aL + aH×100h + aE×10000h) ≤ AFFFFFh (m=1Fh)

0000h ≤ (pL + pH×100h) ≤ FFFFh
fmt = 00h, 81h, 86h, 8Ch, 98h, F0h

fmt = 00h: Refer to Bit Image Index memory
fmt = 81h: Monochrome (1-bit) format
fmt = 86h: Color 6-bit format
fmt = 8Ch: Color 12-bit format
fmt = 98h: Color 24-bit format
fmt = F0h: BMP file format

Function: Display, at the cursor position, the packaged bit image defined in RAM, FROM1, or FROM2.
Cursor position does not change.

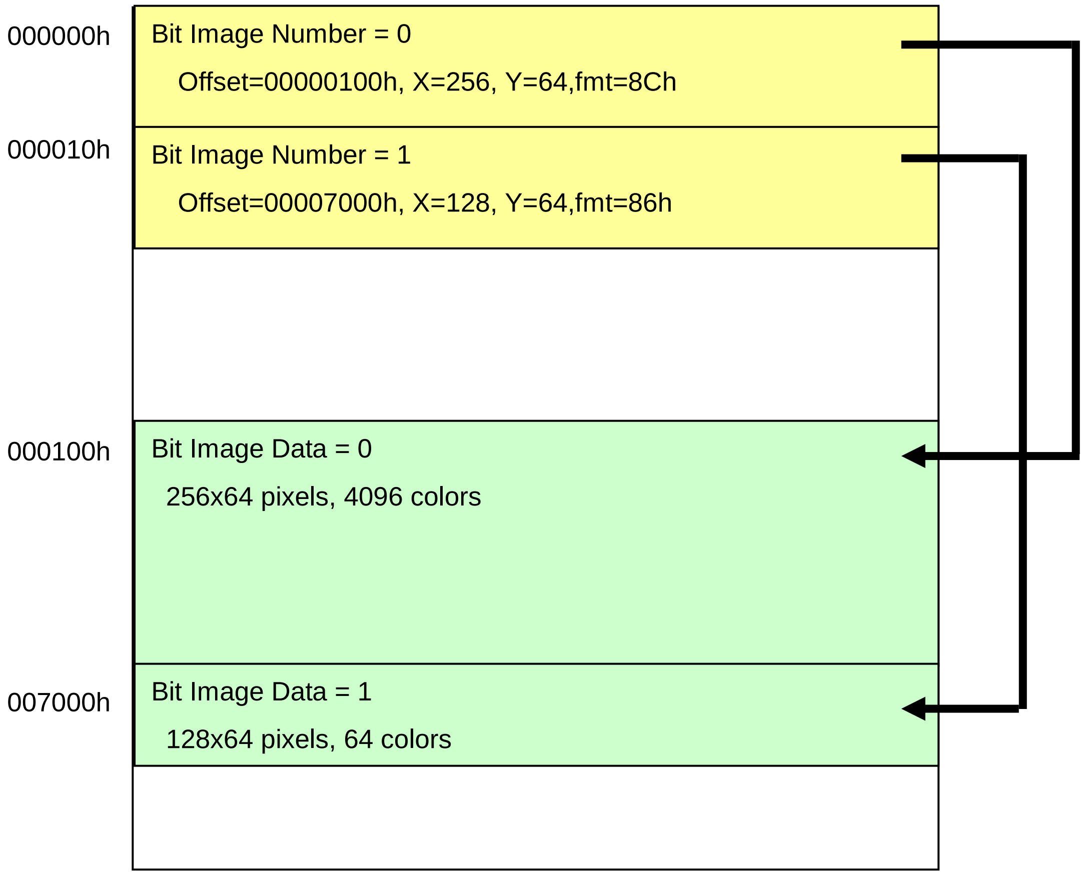
Select RAM, FROM1 or FROM2 bit image by Select Bit image data display memory ‘m’.
Displayed Bit image is specified by Bit Image Index address and Bit Image Number.
If the bit image extends beyond the currently-selected window, only the portion within the current window is displayed.
When an image in FROM is being written to the Display Memory, if the read memory area is exceeded, undefined data is displayed.
Note: For BMP file format, the image size specified in the file header is used. 1, 4, 8, 24, and 32 bit BMP files are supported. 16 bit BMP files are not supported.

Index Format / 1 Image Data

Index count	Contents	
xxxxx0h	Bit Image Data Offset (Offset)	x 1000000h
xxxxx1h		x 10000h
xxxxx2h		x 100h
xxxxx3h		x 1h
xxxxx4h	Reserved	-
xxxxx5h	Reserved	-
xxxxx6h	Pixel size X (X)	x 100h
xxxxx7h		x 1h
xxxxx8h	Pixel size Y (Y)	x 100h
xxxxx9h		x 1h
xxxxxAh	Format (fmt)	81h, 86h, 8Ch, 98h, F0h
xxxxxBh	Reserved	-
xxxxxCh	Reserved	-
xxxxxDh	Reserved	-
xxxxxEh	Reserved	-
xxxxxFh	Reserved	-

Example

RAM / FROM1 / FROM2 Bit Image Memory



000000h Bit Image Number = 0

Offset=00000100h, X=256, Y=64,fmt=8Ch

000010h Bit Image Number = 1

Offset=00007000h, X=128, Y=64,fmt=86h

000100h Bit Image Data = 0

256x64 pixels, 4096 colors

007000h Bit Image Data = 1

128x64 pixels, 64 colors

3.6.4.44 US r n (Reverse display)

Code: 1Fh 72h n

n: Reverse display ON/OFF

Definable area: 00h ≤ n ≤ 01h
n = 00h: Reverse OFF
n = 01h: Reverse ON

Default: n = 00h or Memory SW setting.

Function: Reverse display ON/OFF for character and image display.

Changing this setting only affects subsequent data.
Content already displayed is not affected.
For text and monochrome bit images, background and foreground are interchanged. For color images (including 2-color monochrome BMP file format), the displayed color is inverted. Refer to 3.8 Color combination format for details.

3.6.4.45 US w n (Write mixture display mode)

Code: 1Fh 77h n

n: Display write mode

Definable area: n = 10h, 11h
n = 10h: Normal display write
(Background pixels are written to display memory)
n = 11h: Thru write
(Background pixels are not written to display memory)

Default: n = 10h or Memory SW setting.

Function: Specifies write mixture mode.

For newly-written characters and images, background pixels can either overwrite the existing display memory content (Normal display write), or not overwrite (Thru write), which leaves the currently-displayed pixel unchanged. The effect of “Thru write” is to make the background “transparent”. For color images, black pixels (if any) are treated as background. Refer to 3.8 Color combination format for details.

3.6.4.46 US (w 01h a (Current Window select)

Code: 1Fh 28h 77h 01h a

a: Window number

Definable area: 00h ≤ a ≤ 04h
a = 00h: Base-Window
a = 01h – 04h: User-Window

Function: Select current window

Command is ignored if Window number is for a User-Window that is not defined.

3.6.4.47 US (w 02h a b[xPL xPH yPL yPH xSL xSH ySL ySH] (User Window define / cancel)

Code: 1Fh 28h 77h 02h a b [xPL xPH yPL yPH xSL xSH ySL ySH]

a:	Definable window number (No.1 – 4)
b:	Define or Cancel
xPL:	Left position of window x, lower byte (by 1 pixel)
xPH:	Left position of window x, upper byte (by 1 pixel)
yPL:	Top position of window y, lower byte (by 1 pixel)
yPH:	Top position of window y, upper byte (by 1 pixel)
xSL:	X size of window, lower byte (by 1 pixel)
xSH:	X size of window, upper byte (by 1 pixel)
ySL:	Y size of window, lower byte (by 1 pixel)
ySH:	Y size of window, upper byte (by 1 pixel)

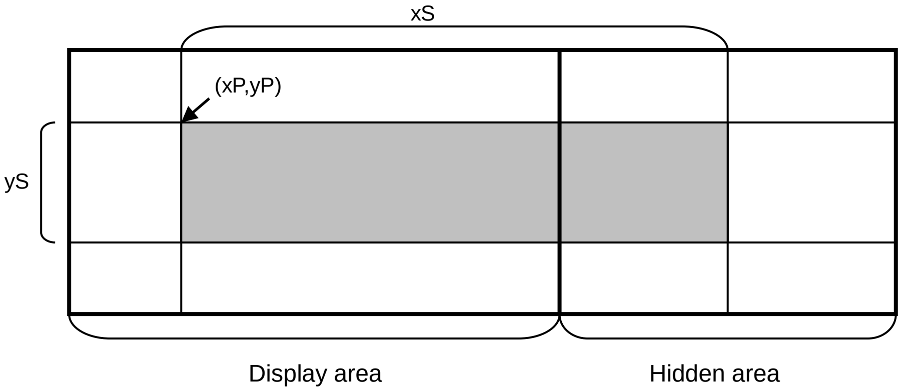
Definable area: 01h ≤ a ≤ 04h
00h ≤ b ≤ 01h
b = 00h: Cancel
b = 01h: Define
0000h ≤ (xPL + xPH×100h) ≤ ***Max_Xdot***
0000h ≤ (yPL + yPH×100h) ≤ ***Max_Ydot***
0001h ≤ (xSL + xSH×100h) ≤ (***Xdots*** – (xPL + xPH×100h))
0001h ≤ (ySL + ySH×100h) ≤ (***Ydots*** – (yPL + yPH×100h))

Function: Define or cancel User-Window

Display contents are not changed by this command.

User-Window define (b=01h)

Specify User-Window number, window position, and window size.
Window position and Window size are specified in units of 1 pixel.



Up to 4 User-Windows can be defined.
The cursor position for the window is initialized to top left (X=0, Y=0).

User-Window cancel (b=00h)

For User-Window cancel, window range parameters [xPL – ySH] are not used.
If the currently-selected window is cancelled, the Base-Window becomes the currently-selected window.
If any of ‘a’, ‘b’, ‘xP’, ‘yP’, ‘xS’, or ‘yS’ are outside the definable area, the command is cancelled at that point and the following data is treated as standard data.

3.6.4.48 ESC % n (Download character ON/OFF)

Code: 1Bh 25h n

Definable area: n = 00h, 01h
n = 01h: Enable
(If download character is not defined, built-in character is displayed)
n = 00h: Disable

Function: Enable or disable display of download characters
(6×8, 8×16, 12×24, and 16×32 pixel).

Characters already displayed are not affected.

3.6.4.49 ESC & a c1 c2 [x1 d1...d(y×x1)] ... [xk d1...d(y×xk) (Download character definition)

Code: 1Bh 26h a c1 c2 [x1 d1...d(y×x1)] ... [xk d1...d(y×xk)]

a:	Select character type
c1:	Start character code
c2:	End character code
x:	Number of pixels for X-direction
d:	Defined data (Refer to 3.9 Download character format)

Definable area: 01h ≤ a ≤ 04h
a = 01h: Select 6×8 pixel character
a = 02h: Select 8×16 pixel character
a = 03h: Select 12×24 pixel character
a = 04h: Select 16×32 pixel character

When 6×8 pixel character is selected;
00h ≤ x ≤ 06h

When 8×16 pixel character is selected;
00h ≤ x ≤ 08h

When 12×24 pixel character is selected;
00h ≤ x ≤ 0Ch

When 16×32 pixel character is selected;
00h ≤ x ≤ 10h
20h ≤ c1 ≤ c2 ≤ FFh
00h ≤ d ≤ FFh
k = c2 – c1 + 1

Function: Define download characters (1-byte characters) into RAM.
For each font size, a maximum of 16 download characters can be defined.
After the maximum number of download characters are defined, in order to define other character codes, space must first be obtained using the Download character delete command.
Downloaded characters are valid until redefined, an initialize (ESC@) sequence is executed, or the power is turned off.
To display download characters the commands Download character definition and Download character ON/OFF (set to ON) are required.
If ‘x’ is smaller than the character width, the remaining space on the right is filled with blank (non-display) pixels.
If a currently-displayed download character is re-defined, there is no affect on the currently-displayed character. It is effective only for newly input characters.
Download characters can be saved into FROM using the Save download character command.

3.6.4.50 ESC ? a c (Downloaded character delete)

Code: 1Bh 3Fh a c

a:	Select character type
c:	Delete Character code

Definable area: 01h ≤ a ≤ 04h
a = 01h: Select 6×8 pixel character
a = 02h: Select 8×16 pixel character
a = 03h: Select 12×24 pixel character
a = 04h: Select 16×32 pixel character
20h ≤ c ≤ FFh

Function: Delete defined download character (1-byte character).
Built-in character is displayed after download character is deleted.
Characters already displayed are not affected.
Command is ignored if download character is not defined for the given character code.

3.6.4.51 US (g 10h c1 c2 d1...dk (16×16 Download character definition)

Code: 1Fh 28h 67h 10h c1 c2 d1...d32

c1: Character code, upper byte
c2: Character code, lower byte
d: Definition data (refer to 3.9 Download character format)

Definable area: c1, c2: Depends on currently-selected language:

Language	Encoding	c1	c2
Japanese	JIS X0208 (SHIFT-JIS)	c1 = ECh	40h ≤ c2 ≤ 4Fh
Korean	KSC5601-87	c1 = FEh	A1h ≤ c2 ≤ B0h
Simplified Chinese	GB2312-80	c1 = FEh	A1h ≤ c2 ≤ B0h
Traditional Chinese	Big-5	c1 = FEh	A1h ≤ c2 ≤ B0h

00h ≤ d ≤ FFh

Function: Defines a 16×16 pixel downloaded character (2-byte character) in character code specified by c1 and c2.

A maximum 16 download characters can be defined.
Definition data ‘d’ is processed as character pattern data in column format, and is stored sequentially from the left.
Download character is temporary stored in RAM, but can be stored in FROM using Download character save command.

3.6.4.52 US (g 11h c1 c2 (16×16 Downloaded character delete)

Code: 1Fh 28h 67h 11h c1 c2

c1: Character code, upper byte
c2: Character code, lower byte

Definable area: c1, c2: Depends on currently-selected language:

Language	Encoding	c1	c2
Japanese	JIS X0208 (SHIFT-JIS)	c1 = ECh	40h ≤ c2 ≤ 4Fh
Korean	KSC5601-87	c1 = FEh	A1h ≤ c2 ≤ B0h
Simplified Chinese	GB2312-80	c1 = FEh	A1h ≤ c2 ≤ B0h
Traditional Chinese	Big-5	c1 = FEh	A1h ≤ c2 ≤ B0h

Function: Delete defined 16×16 download character in code specified by c1 and c2.

3.6.4.53 US (g 14h c1 c2 d1...d128 (32×32 Download character definition)

Code: 1Fh 28h 67h 14h c1 c2 d1...d128

c1: Character code, upper byte
c2: Character code, lower byte
d: Definition data (refer to 3.9 Download character format)

Definable area: c1, c2: Depends on currently-selected language:

Language	Encoding	c1	c2
Japanese	JIS X0208 (SHIFT-JIS)	c1 = ECh	40h ≤ c2 ≤ 4Fh

00h ≤ d ≤ FFh

Function: Defines a 32×32 pixel downloaded character (2-byte character) in character code specified by c1 and c2.

A maximum 16 download characters can be defined.
Definition data ‘d’ is processed as character pattern data in column format, and is stored sequentially from the left.
Download character is temporary stored in RAM, but can be stored in FROM using Download character save command.
This command is invalid if language selection is not set to Japanese.

3.6.4.54 US (g 15h c1 c2 (32×32 Downloaded character delete)

Code: 1Fh 28h 67h 15h c1 c2

c1: Character code, upper byte
c2: Character code, lower byte

Definable area: c1, c2: Depends on currently-selected language:

Language	Encoding	c1	c2
Japanese	JIS X0208 (SHIFT-JIS)	c1 = ECh	40h ≤ c2 ≤ 4Fh

Function: Delete defined 32×32 download character in code specified by c1 and c2.
This command is invalid if language selection is not set to Japanese.

3.6.4.55 US (e 11h a (Download character save)

Code: 1Fh 28h 65h 11h a

a: Font size

Definable area: 01h ≤ a ≤ 06h
a = 01h: 6×8 pixel
a = 02h: 8×16 pixel
a = 03h: 16×16 pixel
a = 04h: 16×32 pixel
a = 05h: 32×32 pixel
a = 06h: 12×24 pixel

Function: Save the download characters defined on RAM to FROM (RAM→FROM).

The saved content is re-enabled using the Download character restore command.
This command is only valid in User setup mode.
Data is not committed to FROM until User setup mode end.

3.6.4.56 US (e 21h a (Download character restore)

Code: 1Fh 28h 65h 21h a

a: Font size

Definable area: 01h ≤ a ≤ 06h
a = 01h: 6×8 pixel
a = 02h: 8×16 pixel
a = 03h: 16×16 pixel
a = 04h: 16×32 pixel
a = 05h: 32×32 pixel
a = 06h: 12×24 pixel

Function: Transfer the download characters saved in FROM to RAM.

Command is ignored if specified font size download characters are not registered in FROM.
Command is valid in both User setup mode and Normal mode.

3.6.4.57 US (e 13h m P(80h-1) P(80h-2) ... P(FFh-n) (FROM User font definition)

Code: 1Fh 28h 65h 13h m P(80h-1) P(80h-2) ... P(FFh-n)

m: User table
p: Definition data (refer to 3.8 Download character format)

Definable area: 01h ≤ m ≤ 04h
m = 01h: 6×8 pixel user table
m = 02h: 8×16 pixel user table
m = 03h: 12×24 pixel user table
m = 04h: 16×32 pixel user table

m=01h: P(80h-1) ... P(80h-6) ... P(FFh-6)	6 Bytes / font × 128 characters (768 bytes)
m=02h: P(80h-1) ... P(80h-16) ... P(FFh-16)	16 Bytes / font × 128 characters (2,048 bytes)
m=03h: P(80h-1) ... P(80h-36) ... P(FFh-36)	36 Bytes / font × 128 characters (4,608 bytes)
m=04h: P(80h-1) ... P(80h-64) ... P(FFh-64)	64 Bytes / font × 128 characters (8,192 bytes)

Function: Define the user font for each size of 1-byte code to the user table.

This command defines all 128 characters at once; it is not possible to only define a part of the character code space.
User font tables for each font size are set to blank (00h) when shipped.
This command is only valid in User setup mode.
Data is not committed to FROM until User setup mode end.
FROM User table display is selected using "Character table type" command (refer to 3.6.4.6).

3.6.4.58 US (e 15h a b p(1) ... p(65536) (FROM extension font definition)

Code: 1Fh 28h 65h 15h a b p(1) ... p(65536)

a: Bank
b: Define / Delete
p: Definition data (if Define)

Definable area: a = 01h
b = 00h, 01h
b = 00h: FROM extension font is deleted.
Definition data parameter is not used.
b = 01h: FROM extension font is defined.
Consult manufacturer for definition data format.
FROM extension font is in deleted state when shipped.
00h ≤ p ≤ FFh

Function: Define or delete FROM extension font.

FROM extension font is in deleted state when shipped.
This command is only valid in User setup mode.
Data is not committed to FROM until User setup mode end.

3.6.4.59 US (e 01h d1 d2 (User setup mode start)

Code: 1Fh 28h 65h 01h 49h 4Eh

Definable area: d1 = 49h (Character ‘I’)
d2 = 4Eh (Character ‘N’)

Function: Start User setup mode.

The following data is transmitted from the currently-active interface:

Transmitted data	Hex	Data length
(1) Header	28h	1 byte
(2) Identifier 1	65h	1 byte
(3) Identifier 2	01h	1 byte
(4) NUL	00h	1 byte

This command is only valid in Normal mode.
Display screen is blanked.

3.6.4.60 US (e 02h d1 d2 d3 (User setup mode end)

Code: 1Fh 28h 65h 02h 4Fh 55h 54h

Definable area: d1 = 4Fh (Character ‘O’)
d2 = 55h (Character ‘U’)
d3 = 54h (Character ‘T’)

Function: End User setup mode, and software reset of display as follows:
(1) Wait for any in-progress operations (memory control, information transmission, etc) to complete.
(2) Output display BUSY signal.
(3) Software reset.

This command is only valid in User setup mode.
This command clears the receive buffer, and all settings (Download character, Macro settings, GPIO settings, etc) are reset to power-on state.

3.6.4.61 US (p 01h n a (I/O Port Input / Output setting)

Code: 1Fh 28h 70h 01h n a

n: I/O port number

a: Set Input / Output (bit-wise)

Definable area: 00h ≤ n ≤ 03h

n = 00h: Port 0 (GPIO 0 – 7)
n = 01h: Port 1 (GPIO 8 – 15)
n = 02h: Port 2 (GPIO 16 – 23)
n = 03h: Port 3 (GPIO 24, 25)

00h ≤ a ≤ FFh

Bit value = 0: Input
Bit value = 1: Output

Default: a = 00h (All ports set to input)

Function: Set input or output for general-purpose I/O ports.

Port input / output is set by value of ‘a’.

Bit assignment is as follows:

I/O port number	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
n = 00h	GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
n = 01h	GPIO 15	GPIO 14	GPIO 13	GPIO 12	GPIO 11	GPIO 10	GPIO 9	GPIO 8
n = 02h	GPIO 23	GPIO 22	GPIO 21	GPIO 20	GPIO 19	GPIO 18	GPIO 17	GPIO 16
n = 03h	-	-	-	-	-	-	GPIO 25	GPIO 24

Caution: I/O port is intended for simple peripheral switches and for controlling lights, etc, and should not be used for applications where high reliability is required.

3.6.4.62 US (p 10h n a (I/O Port Output)

Code: 1Fh 28h 70h 10h n a

n: I/O port number

a: Output data value

Definable area: 00h ≤ n ≤ 03h

n = 00h: Port 0 (GPIO 0 – 7)
n = 01h: Port 1 (GPIO 8 – 15)
n = 02h: Port 2 (GPIO 16 – 23)
n = 03h: Port 3 (GPIO 24, 25)

00h ≤ a ≤ FFh

Function: Output data to general-purpose I/O port.

Output data is set by value of ‘a’.

Bit assignment is as follows:

I/O port number	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
n = 00h	GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
n = 01h	GPIO 15	GPIO 14	GPIO 13	GPIO 12	GPIO 11	GPIO 10	GPIO 9	GPIO 8
n = 02h	GPIO 23	GPIO 22	GPIO 21	GPIO 20	GPIO 19	GPIO 18	GPIO 17	GPIO 16
n = 03h	-	-	-	-	-	-	GPIO 25	GPIO 24

3.6.4.63 US (p 20h n (I/O Port Input)

Code: 1Fh 28h 70h 20h n

n: I/O port number

Definable area: 00h ≤ n ≤ 01h

n = 00h: Port 0 (GPIO 0 – 7)
n = 01h: Port 1 (GPIO 8 – 15)
n = 02h: Port 2 (GPIO 16 – 23)
n = 03h: Port 3 (GPIO 24 – 25)

Function: The state of a general-purpose I/O port at the time this command is processed is transmitted.

The following data is transmitted from the currently-active interface:

Transmitted data	Hex	Data length
(1) Header	28h	1 byte
(2) Identifier (1)	70h	1 byte
(3) Identifier (2)	20h	1 byte
(4) Data	00h–FFh	1 byte

Response time varies depending on the state of the receive buffer.

3.6.4.64 US : pL pH [d1 ... dk] (RAM Macro define / delete)

Code: 1Fh 3Ah pL pH [d1 ... dk]

pL: RAM Macro data length, lower byte
pH: RAM Macro data length, upper byte
d: RAM Macro data

Definable area: 0000h ≤ (pL + pH×100h) ≤ 0400h
(pL + pH×100h) > 0000h: Supplied data ‘d’ is stored as Macro.
(pL + pH×100h) = 0000h: Macro is deleted.

Function: Define or delete RAM Macro or RAM Program Macro.

If Macro data length is outside the definable area, the command is cancelled, and the following data is treated as standard data.
Do not define any of the following commands in a Macro:
Initialize, Macro execution, RAM Macro processing definition, User setup mode start, [US (e] group commands (FROM bit image definition, Download character save, etc), Macro execution settings Memory re-write mode, General-purpose memory commands.
Program Macro details: Refer to specification DS-1940-0005-xx Program Macro.

3.6.4.65 US (e 12h a pL pH t1 t2 [d(1) ... d(p)] (FROM Macro define / delete)

Code: 1Fh 28h 65h 12h a pL pH t1 t2 [d1 ... d(p)]

a: FROM Macro registration number
pL: FROM Macro data length, lower byte
pH: FROM Macro data length, upper byte
t1: Display time interval (t1 ×***IntTime***)
t2: Idle time for Macro repetition (t2 × ***IntTime***)
d: FROM Macro data

Definable area: 01h ≤ a ≤ 04h: FROM Macro number 1 – 4
0000h ≤ (pL + pH×100h) ≤ 2000h (if using 4 Macros), 8000h (if using 1 Macro)
(pL + pH×100h) > 0: Supplied data ‘d’ is stored as Macro.
(pL + pH×100h) = 0: Macro is deleted.
00h ≤ t1 ≤ FFh
00h ≤ t2 ≤ FFh
00h ≤ d ≤ FFh

Function: Define or delete FROM Macro or FROM Program Macro.

FROM Macro storage capacity is a total of 32KB, 8KB / Macro when using 4 Macros.
For Macros exceeding 8KB, multiple Macro definition areas are used, which may result in some Macro number areas being undefined.
If Macro data length is outside the definable area, the command is cancelled, and the following data is treated as standard data.
Display time interval (t1) and Idle time (t2) settings are used when FROM Macro execution at power on is used.
Display time interval refers to the interval time between displaying characters, and does not affect the processing speed of command code.
Idle time refers to the time period from processing the last Macro data until the Macro is re-executed.
This command is only valid in User setup mode.
Do not define any of the following commands in a Macro:
Initialize, Macro execution, RAM Macro define / delete, User setup mode start, [US (e] group commands (FROM bit image definition, Download character save, etc), Macro execution settings, Memory re-write mode, General-purpose memory commands.
Data is not committed to FROM until User setup mode end.
Program Macro details: Refer to specification DS-1940-0005-xx Program Macro.

Example:

FROM Macro 1 area	Undefined
FROM Macro 2 area	Undefined
FROM Macro 3 area	Undefined
FROM Macro 4 area	Undefined

↓Define 8KB Macros in Macro 1 – 4

FROM Macro 1 area	FROM Macro 1: 8KB
FROM Macro 2 area	FROM Macro 2: 8KB
FROM Macro 3 area	FROM Macro 3: 8KB
FROM Macro 4 area	FROM Macro 4: 8KB

↓Define 16KB Macro in Macro 2

FROM Macro 1 area	FROM Macro 1: 8KB
FROM Macro 2 area	FROM Macro 2: 16KB
FROM Macro 3 area	
FROM Macro 4 area	FROM Macro 4: 8KB

↓Define 16KB Macro in Macro 3

FROM Macro 1 area	FROM Macro 1: 8KB
FROM Macro 2 area	Undefined
FROM Macro 3 area	FROM Macro 3: 16KB
FROM Macro 4 area	

↓Define 24KB Macro in Macro 1

FROM Macro 1 area	FROM Macro 1: 24KB
FROM Macro 2 area	
FROM Macro 3 area	
FROM Macro 4 area	Undefined

3.6.4.66 US ^ a t1 t2 (Macro execution)

Code: 1Fh 5Eh a t1 t2

a: Macro processing definition number
t1: Display time interval (t1 ×***IntTime***)
t2: Idle time for Macro repetition (t2 ×***IntTime***)

Definable area: 00h ≤ a ≤ 04h, 80h ≤ a ≤ 84h
a = 00h: RAM Macro 0
01h ≤ a ≤ 04h: FROM Macro 1 – 4
a = 80h: RAM Program Macro 0
81h ≤ a ≤ 84h: FROM Program Macro 1 – 4
00h ≤ t1 ≤ FFh
00h ≤ t2 ≤ FFh

Function: Continuously execute contents of defined Macro ‘a’.

Display time interval refers to the interval time between displaying characters, and does not affect the processing speed of command code.
Idle time refers to the time period from processing the last Macro data until the Macro is re-executed.
If Macro 'a' is not defined, or is outside the definable area, the entire command (up to t2) is ignored.
Macro execution is stopped when a command is input.
The current window (Write screen mode area if Base-Window) is cleared and cursor moves to home position.
Display settings remain in the current state when the Macro ended.

3.6.4.67 US (i 20h a b c (Macro end condition)

Code: 1Fh 28h 69h 20h a b c

a: Macro end code Enable / Disable
b: Macro end code
c: Macro end Clear Screen setting

Definable area: a = 00h, 01h
a = 00h: Macro end code Disabled
a = 01h: Macro end code Enabled
00h ≤ b ≤ FFh
c = 00h, 01h
c = 00h: Clear Screen at Macro end
c = 01h: Do not clear screen at Macro end

Default: a = 00h or Memory SW setting.
b = 00h or Memory SW setting.
c = 00h or Memory SW setting.

Function: Macro end condition set.

a = 00h: Macro will unconditionally end if data is received.
a = 01h: Macro will end if data byte 'b' is received. All other values are ignored.
c = 00h: Clear screen on Macro end.
c = 01h: Do not clear screen on Macro end.
The received byte code that ends the Macro is processed as the first byte of the next command.
This setting is not applicable for Program Macro.

3.6.4.68 US (e 03h a b (Memory SW setting)

Code: 1Fh 28h 65h 03h a b
or 1Fh 28h 65h 03h a b c[1] d[1] [... c[b] d[b]]

a: Memory SW Number
b: Setting data

Definable area:

Single Memory SW setting:
00h ≤ a ≤ 3Fh
00h ≤ b ≤ FFh

Multiple Memory SW setting:
a = FFh
01h ≤ b ≤ FFh
00h ≤ c ≤ 3Fh
00h ≤ d ≤ FFh

Function: Set Memory SW.

A single Memory switch can be set (a=00h–3Fh) or multiple Memory switches can be set (a=FFh).
Single setting (a=00h–3Fh): a = Memory SW number, b = Setting value.
Multiple setting (a=FFh): b = Number of settings, c = Memory SW number, d = Setting value.
This command is only valid in User setup mode.
Data is not committed to FROM until User setup mode end.
Memory SW details: Refer to section 4.2 Memory SW.

3.6.4.69 US (e 04h a (Memory SW data send)

Code: 1Fh 28h 65h 04h a
or 1Fh 28h 65h 04h a b c[1] [... c[b]]

a: Memory SW Number

Definable area:

Single Memory SW read:
00h ≤ a ≤ 3Fh

Multiple Memory SW read:
a = FFh
01h ≤ b ≤ FFh
00h ≤ c ≤ 3Fh

Function: Send the the contents of Memory SW data.

A single Memory switch can be read (a=00h–3Fh) or multiple Memory switches can be read (a=FFh).
Single read (a=00h–3Fh): a = Memory SW number.
Multiple read (a=FFh): b = Number of reads, c = Memory SW number.

The following data is transmitted from the currently-active interface:

Transmitted data	Hex	Data length
(1) Header	28h	1 byte
(2) Identifier 1	65h	1 byte
(3) Identifier 2	04h	1 byte
(4) Data	00h–FFh	1 byte / b byte(s)

This command is valid in both User setup mode and Normal mode.
Memory SW details: Refer to section 4.2 Memory SW.

3.6.4.70 US (e 18h sL sH sE m1 a1L a1H a1E d[1] ... d[s] (General-purpose memory store / FROM2 image store)

Code: 1Fh 28h 65h 18h sL sH sE m1 a1L a1H a1E d[1] ... d[s]

sL: Data size, lower byte
sH: Data size, upper byte
sE: Data size, extension byte
m1: Memory select
a1L: Memory address, lower byte
a1H: Memory address, upper byte
a1E: Memory address, extension byte
d: Data to store

Definable area: 10h ≤ m ≤ 17h, m = 30h, 31h
m1 = 10h: FROM2 base address 0000.0000h
m1 = 11h: FROM2 base address 0100.0000h
m1 = 12h: FROM2 base address 0200.0000h
m1 = 13h: FROM2 base address 0300.0000h
m1 = 14h: FROM2 base address 0400.0000h
m1 = 15h: FROM2 base address 0500.0000h
m1 = 16h: FROM2 base address 0600.0000h
m1 = 17h: FROM2 base address 0700.0000h
m1 = 18h: FROM2 base address 0800.0000h
m1 = 19h: FROM2 base address 0900.0000h
m1 = 1Ah: FROM2 base address 0A00.0000h
m1 = 1Bh: FROM2 base address 0B00.0000h
m1 = 1Ch: FROM2 base address 0C00.0000h
m1 = 1Dh: FROM2 base address 0D00.0000h
m1 = 1Eh: FROM2 base address 0E00.0000h
m1 = 1Fh: FROM2 base address 0F00.0000h
000001h ≤ (sL + sH×100h + sE×10000h) ≤ FFFFFFh
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ FFFFFFh
m1 = 30h (General-purpose RAM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 000400h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 0003FFh
m1 = 31h (General-purpose FROM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 001000h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 00FFFFh
00h ≤ d ≤ FFh

Function: Store the supplied data into general-purpose memory / FROM2.
Note: **FROM2 valid area is 0000.0000h – 0FAF.FFFFh (257,024KB).**
General-purpose FROM is in units of 001000h, with a total of 16 areas.

Stored data can be read using General-purpose memory send command or by Program Macro.
Images stored in FROM2 can be displayed using the commands “Downloaded bit image display” and “Packaged Downloaded bit image display”.
Data is stored into the specified address range, and all other memory locations in the same 128KB FROM2 block area are set to FFh.

For General-purpose RAM, data is stored only into the specified address range.
For General-purpose FROM, data is stored into the specified address range, and all other memory locations in the same FROM area are set to FFh. Further, it is not possible to specify data storage that would exceed a general-purpose FROM area.

Storage that would exceed the address range cannot be set.
FROM2 area from 0E00.0000h is reserved for built-in outline font.
This command is valid in both User setup mode and Normal mode.

3.6.4.71 US (e 19h sL sH sE m1 a1L a1H a1E m2 a2L a2H a2E (General-purpose memory transfer)

Code: 1Fh 28h 65h 19h sL sH sE m1 a1L a1H a1E m2 a2L a2H a2E

sL:	Transfer data size, lower byte
sH:	Transfer data size, upper byte
sE:	Transfer data size, extension byte
m1:	Destination memory select
a1L:	Destination address, lower byte
a1H:	Destination address, upper byte
a1E:	Destination address, extension byte
m2:	Source Memory select
a2L:	Source address, lower byte
a2H:	Source address, upper byte
a2E:	Source address, extension byte

Definable area: 10h ≤ m1, m2 ≤ 17h, m1, m2 = 30h, 31h

m1, m2 = 10h: FROM2 base address 0000.0000h
m1, m2 = 11h: FROM2 base address 0100.0000h
m1, m2 = 12h: FROM2 base address 0200.0000h
m1, m2 = 13h: FROM2 base address 0300.0000h
m1, m2 = 14h: FROM2 base address 0400.0000h
m1, m2 = 15h: FROM2 base address 0500.0000h
m1, m2 = 16h: FROM2 base address 0600.0000h
m1, m2 = 17h: FROM2 base address 0700.0000h
m1, m2 = 18h: FROM2 base address 0800.0000h
m1, m2 = 19h: FROM2 base address 0900.0000h
m1, m2 = 1Ah: FROM2 base address 0A00.0000h
m1, m2 = 1Bh: FROM2 base address 0B00.0000h
m1, m2 = 1Ch: FROM2 base address 0C00.0000h
m1, m2 = 1Dh: FROM2 base address 0D00.0000h
m1, m2 = 1Eh: FROM2 base address 0E00.0000h
m1, m2 = 1Fh: FROM2 base address 0F00.0000h
000000h ≤ (sL + sH×100h + sE×10000h) ≤ FFFFFFh
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ FFFFFFh
000000h ≤ (a2L + a2H×100h + a2E×10000h) ≤ FFFFFFh
m1, m2 = 30h (General-purpose RAM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 000400h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 0003FFh
000000h ≤ (a2L + a2H×100h + a2E×10000h) ≤ 0003FFh
m1, m2 = 31h (General-purpose FROM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 001000h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 00FFFFh
000000h ≤ (a2L + a2H×100h + a2E×10000h) ≤ 00FFFFh
00h ≤ d ≤ FFh

Function: Transfer data between general-purpose memory / FROM2 areas.

Note: **FROM2 valid area is 0000.0000h – 0FAF.FFFFh (257,024KB).**
General-purpose FROM is in units of 001000h, with a total of 16 areas.

For General-purpose RAM, data is transferred only into the specified address range.
For General-purpose FROM, data is transferred into the specified address range, and all other memory locations in the same FROM area are set to FFh. Further, it is not possible to specify a data transfer that would exceed a general-purpose FROM area.
For FROM2, data is transferred into the specified address range, and all other memory locations in the same 128KB block area are set to FFh.

Storage that would exceed the address range cannot be set.

FROM2 area from 0E00.0000h is reserved for built-in outline font.
This command is valid in both User setup mode and Normal mode.
If size is 000000h, no data is transferred.

3.6.4.72 US (e 28h sL sH sE m1 a1L a1H a1E (General-purpose memory send)

Code: 1Fh 28h 65h 28h sL sH sE m1 a1L a1H a1E

n: Classify command
sL: Data size, lower byte
sH: Data size, upper byte
sE: Data size, extension byte
m1: Memory select
a1L: Memory address, lower byte
a1H: Memory address, upper byte
a1E: Memory address, extension byte

Definable area: 10h ≤ m1 ≤ 17h, m1 = 30h, 31h

m1 = 10h: FROM2 base address 0000.0000h
m1 = 11h: FROM2 base address 0100.0000h
m1 = 12h: FROM2 base address 0200.0000h
m1 = 13h: FROM2 base address 0300.0000h
m1 = 14h: FROM2 base address 0400.0000h
m1 = 15h: FROM2 base address 0500.0000h
m1 = 16h: FROM2 base address 0600.0000h
m1 = 17h: FROM2 base address 0700.0000h
m1 = 18h: FROM2 base address 0800.0000h
m1 = 19h: FROM2 base address 0900.0000h
m1 = 1Ah: FROM2 base address 0A00.0000h
m1 = 1Bh: FROM2 base address 0B00.0000h
m1 = 1Ch: FROM2 base address 0C00.0000h
m1 = 1Dh: FROM2 base address 0D00.0000h
m1 = 1Eh: FROM2 base address 0E00.0000h
m1 = 1Fh: FROM2 base address 0F00.0000h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ FFFFFFh
000000h ≤ (sL + sH×100h + sE×10000h) ≤ FFFFFFh
m1 = 30h (General-purpose RAM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 000400h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 0003FFh
m1 = 31h (General-purpose FROM):
000001h ≤ (sL + sH×100h + sE×10000h) ≤ 001000h
000000h ≤ (a1L + a1H×100h + a1E×10000h) ≤ 00FFFFh

Function: Send data stored in general-purpose memory.

Note: **FROM2 valid area is 0000.0000h – 0FAF.FFFFh (257,024KB).**
General-purpose FROM is in units of 001000h, with a total of 16 areas.

Data read that would exceed the address range cannot be set.
This command is valid in both User setup mode and Normal mode.

The following data is transmitted from the currently-active interface:

Transmitted data	Hex	Data length
(1) Header	28h	1 byte
(2) Identifier 1	65h	1 byte
(3) Identifier 2	28h	1 byte
(4) Data	00h–FFh	s byte(s)

3.6.4.73 US (e 40h a [b c] (Display status send)

Code: 1Fh 28h 65h 40h a [b c]

Definable area: a = 01h, 02h, 10h, 11h, 20h, 30h, 40h, 41h
a = 01h: Boot version information (b, c not used)
a = 02h: Firmware version information (b, c not used)
a = 10h: 2-byte character code information (b, c not used)
a = 11h: Language type information (b, c not used)
a = 20h: Memory checksum information
00h ≤ b ≤ FFh: Start address (Effective address = b×10000h)
01h ≤ c ≤ FFh: Data length (Effective data length = c×10000h)
a = 30h: Product type information (b, c not used)
a = 40h: Display x pixel information (b, c not used)
a = 41h: Display y pixel information (b, c not used)

Function: Send display status information.

The following data is transmitted from the currently-active interface:

Transmitted data	Hex	Data length
(1) Header	28h	1 byte
(2) Identifier 1	65h	1 byte
(3) Identifier 2	40h	1 byte

(4) Data	00h–FFh	a = 01h: 4 bytes
		a = 02h: 4 bytes
		a = 10h: 15 bytes
		a = 11h: 15 bytes
		a = 20h: 4 bytes
		a = 30h: 15 bytes
		a = 40h: 3 bytes
		a = 41h: 3 bytes

This command is valid in both User setup mode and Normal mode.

3.6.4.74 FS | M m d1 ... d6 (Memory re-write mode start)

Code: 1Ch 7Ch 4Dh m d1 ... d6

Definable area: m = D0h
d1 ... d6 = “MODEIN”

Function: Shift to “Memory re-write mode” from “Normal mode”.

Memory re-write mode is used for changing the firmware and fonts, etc in FROM that cannot be changed in User setup mode.
Changing this FROM requires special commands and tools.
Do not use this command.

3.7 Bit image data format

3.7.1 Normal Bit image data format

Monochrome (1-bit) format (1 byte per 8 pixels or part thereof for each horizontal line):

D7	D6	D5	D4	D3	D2	D1	D0
F.C.	F.C.	F.C.	F.C.	F.C.	F.C.	F.C.	F.C.

F.C.: Fore Color (Color of Character)
F.C. = 1: Fore Color (Color of Character)
F.C. = 0: Background Color

*Note: D7 represents leftmost pixel.
If horizontal width is not a multiple of 8, the remaining lower-order bits are ignored.

Color 6-bit format (1 byte per pixel):

D7	D6	D5	D4	D3	D2	D1	D0
X	X	B bit1	B bit0	G bit1	G bit0	R bit1	R bit0

Color 12-bit format (2 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	B bit3	B bit2	B bit1	B bit0	G bit3	G bit2	G bit1	G bit0	R bit3	R bit2	R bit1	R bit0

Color 16-bit format (2 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
I bit0	B bit5	B bit4	B bit3	B bit2	B bit1	G bit5	G bit4	G bit3	G bit2	G bit1	R bit5	R bit4	R bit3	R bit2	R bit1

I: Intensity bit. Used as common least-significant bit for each of R, G, B.

Color 16-bit high-speed format (2 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
G bit3	G bit2	G bit1	R bit5	R bit4	R bit3	R bit2	R bit1	I bit0	B bit5	B bit4	B bit3	B bit2	B bit1	G bit5	G bit4

I: Intensity bit. Used as common least-significant bit for each of R, G, B.
Note: 16-bit high-speed format is only supported for Downloaded bit image display command.

Color 24-bit format (3 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
B bit7	B bit6	B bit5	B bit4	B bit3	B bit2	B bit1	B bit0	G bit7	G bit6	G bit5	G bit4	G bit3	G bit2	G bit1	G bit0

3rd byte							
D7	D6	D5	D4	D3	D2	D1	D0
R bit7	R bit6	R bit5	R bit4	R bit3	R bit2	R bit1	R bit0

3.7.2 Packaged Bit image data format

Color 6-bit format (1 byte per pixel):

D7	D6	D5	D4	D3	D2	D1	D0
LEN bit1	LEN bit0	B bit1	B bit0	G bit1	G bit0	R bit1	R bit0

LEN: Continued pixel length
LEN = 0: until right end
LEN = 1: 1 pixel
LEN = 2: 2 pixels
LEN = 3: 3 pixels

Color 12-bit format (2 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
LEN Bit3	LEN bit2	LEN bit1	LEN bit0	B bit3	B bit2	B bit1	B bit0	G bit3	G bit2	G bit1	G bit0	R bit3	R bit2	R bit1	R bit0

LEN: Continued pixel length
LEN = 0: until right end
LEN = 1: 1 pixel
LEN = 2: 2 pixels
LEN = 3: 3 pixels
:
LEN = 15: 15 pixels

Color 24-bit format (4 bytes per pixel):

1st byte								2nd byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
LEN bit7	LEN bit6	LEN bit5	LEN bit4	LEN bit3	LEN bit2	LEN bit1	LEN bit0	B bit7	B bit6	B bit5	B bit4	B bit3	B bit2	B bit1	B bit0

3rd byte								4th byte							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
G bit7	G bit6	G bit5	G bit4	G bit3	G bit2	G bit1	G bit0	R bit7	R bit6	R bit5	R bit4	R bit3	R bit2	R bit1	R bit0

LEN: Continued pixel length
LEN = 0: until right end
LEN = 1: 1 pixel
LEN = 2: 2 pixels
LEN = 3: 3 pixels
:
LEN = 255: 255 pixels

Example

8×8 pixels Bit Image, Color 12-bit format

0000h	0FFFh	0FFFh	0FFFh	0FFFh	0FFFh	0000h	0000h
0000h	0FFFh	0555h	0555h	0555h	0555h	0555h	0000h
0000h	0FFFh	0555h	0000h	0000h	0000h	0000h	0000h
0000h	0FFFh	0FFFh	0FFFh	0FFFh	0FFFh	0000h	0000h
0000h	0000h	0555h	0555h	0555h	0FFFh	0555h	0000h
0000h	0000h	0000h	0000h	0000h	0FFFh	0555h	0000h
0000h	0FFFh	0FFFh	0FFFh	0FFFh	0FFFh	0555h	0000h
0000h	0000h	0555h	0555h	0555h	0555h	0555h	0000h

d(1, 2)=1000h d(3, 4)=5FFFh d(5, 6)=0000h
d(7, 8)=1000h d(9,10)=1FFFh d(11,12)=5555h d(13,14)=0000h
d(15,16)=1000h d(17,18)=1FFFh d(19,20)=1555h d(21,22)=0000h
d(23,24)=1000h d(25,26)=5FFFh d(27,28)=0000h
d(29,30)=2000h d(31,32)=3555h d(33,34)=1FFFh d(35,36)=1555h d(37,38)=0000h
d(39,40)=5000h d(41,42)=1FFFh d(43,44)=1555h d(45,46)=0000h
d(47,48)=1000h d(49,50)=5FFFh d(51,52)=1555h d(53,54)=0000h
d(55,56)=2000h d(57,58)=5555h d(59,60)=0000h

Data length comparison

Normal Bit image data format:	2×8×8 = 128 bytes
Packaged Bit image data format:	60 bytes

3.8 Color combination format

This section explains how the combination of Reverse display and Write mixture display mode affect the display of text and images.

3.8.1 Character display color combination format

Reverse display setting interchanges the Character color (fore color) and background, and Write mixture display mode determines whether the background is transparent or opaque, as summarized in the following table. Pixels for Shadow or Bordering Character Style (if applicable) are not affected. Pixels for Bold Character Style (if applicable) are treated the same as for the character.

C 1: Character Color, C 2: Background Color, C 3: Shadow and Bordering Color

Write mixture display mode	Reverse display	Character Style	Character	Background	Character Style
Normal display write	OFF	OFF	C 1	C 2	
		Bold	C 1	C 2	C 1
		Shadow	C 1	C 2	C 3
		Bordering	C 1	C 2	C 3
	ON	OFF	C 2	C 1	
		Bold	C 2	C 1	C 2
		Shadow	C 2	C 1	C 3
		Bordering	C 2	C 1	C 3
Thru write	OFF	OFF	C 1	Transparent	
		Bold	C 1	Transparent	C 1
		Shadow	C 1	Transparent	C 3
		Bordering	C 1	Transparent	C 3
	ON	OFF	Transparent	C 1	
		Bold	Transparent	C 1	Transparent
		Shadow	Transparent	C 1	C 3
		Bordering	Transparent	C 1	C 3

3.8.2 Monochrome bit image color combination format

Monochrome bit images are handled similar to character display. Reverse display setting interchanges the Character color (fore color) and background, and Write mixture display mode determines whether the background is transparent or opaque, as summarized in the following table:

C 1: Character Color, C 2: Background Color

Write mixture display mode	Reverse display	Bit = 1	Bit = 0
Normal display write	OFF	C 1	C 2
	ON	C 2	C 1
Thru write	OFF	C 1	Transparent
	ON	Transparent	C 1

3.8.3 Color image / BMP file color combination format

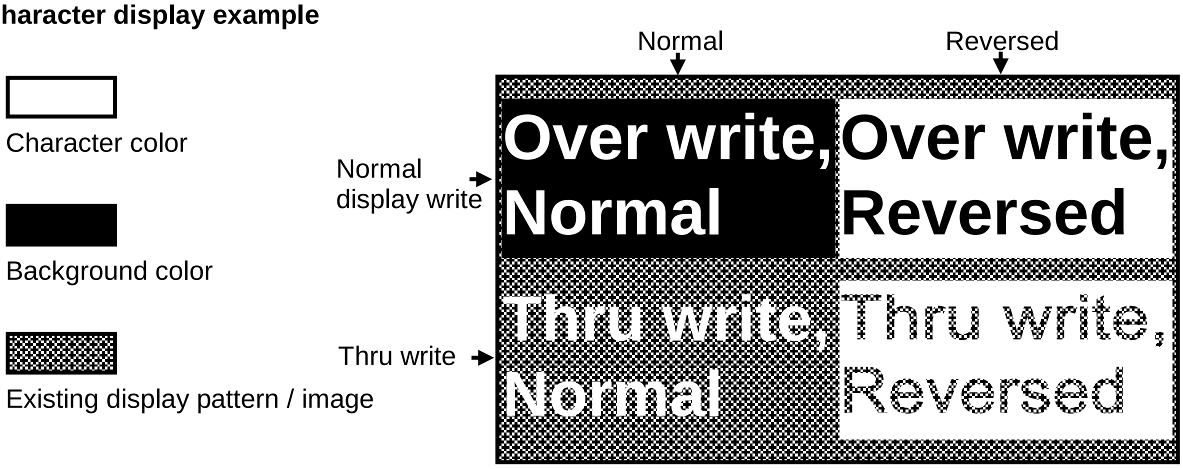
For color images (including 2-color monochrome BMP files), the Reverse display setting inverts the color data for each pixel that is displayed, and Write mixture display mode determines whether the black pixels in the image (if any) are transparent or opaque, as summarized in the following table:

C 1: Pixel color

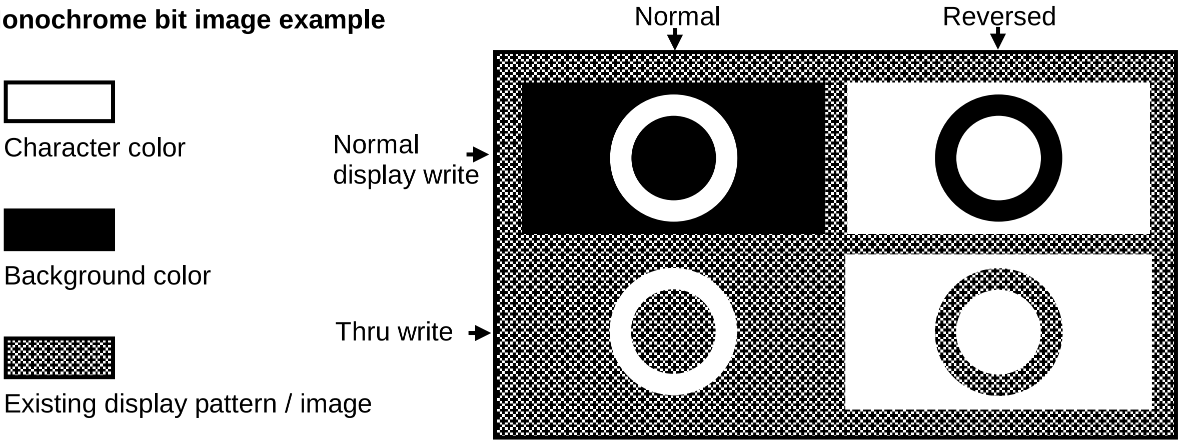
Write mixture display mode	Reverse display	Pixel Color	
		C 1 = Black	C 1 = Other
Normla display write	OFF	C 1	C 1
	ON	~C 1	~C 1
Thru write	OFF	Transparent	C 1
	ON	Transparent	~C 1

3.8.4 Color combination examples

Character display example

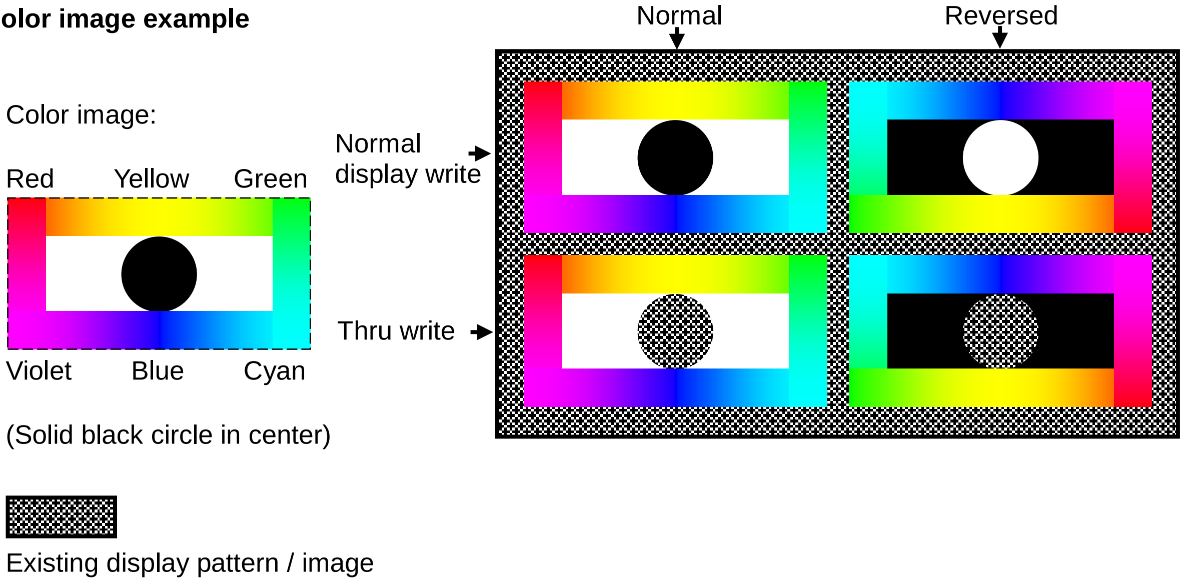


Monochrome bit image example



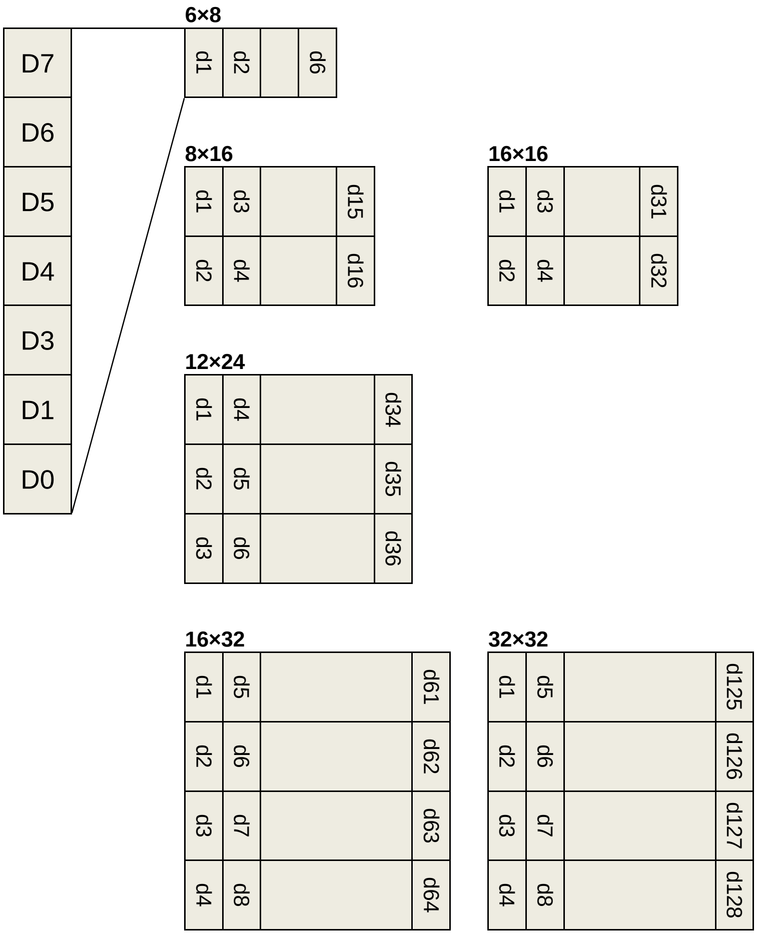
(Character color = Fore color)

Color image example



3.9 Download character format

Download character format is shown below.



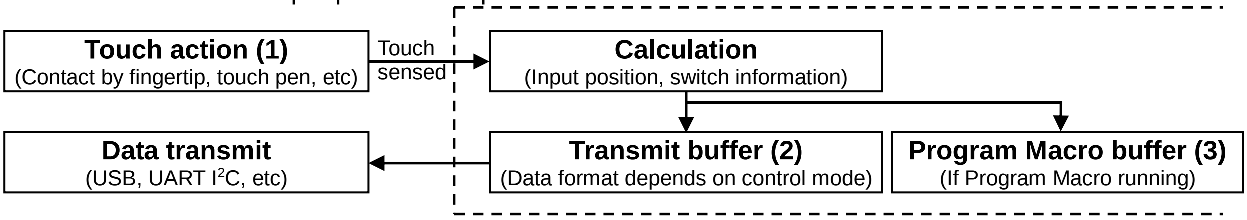
3.10 Touch Panel

3.10.1 Outline

Detection method: Projected-capacitive touch (multi-touch (multiple-point input) supported)

3.10.2 Basic Operation

The display module features a touch panel for handling input by fingertip or touch pen, etc. The touch panel function sends data for the input position or operation of user-defined switch buttons.


Touch action (1)
(Contact by fingertip, touch pen, etc)
Touch sensed
Calculation
(Input position, switch information)
Data transmit
(USB, UART I2C, etc)
Transmit buffer (2)
(Data format depends on control mode)
Program Macro buffer (3)
(If Program Macro running)

Notes:

(1) The number of simultaneous touches recognized depends on the Touch Mode.
(2) Touch information is queued when Touch Panel Data Transmit is ON and sufficient space is available in the transmit buffer (buffer capacity: 128 bytes). When there is insufficient space, touch actions are not queued, so the queued data should be periodically read.
(3) Touch information is queued when Program Macro is running and space is available in the buffer (buffer capacity: 1 touch). Refer to Program Macro specification for usage details.

3.10.3 Touch Modes

There are two Touch Modes. Single-Touch Mode (default) recognizes only one touch at a time, generating continuous touch reports while the touch continues, stopping the reports when touch is released. This mode is software-compatible with resistive touch-panel modules. Multi-Touch Mode recognizes a maximum of 1 to 10 (configurable) touches, generating touch reports only when changes (touch / release / touch position change) occur.

3.10.4 Control Modes

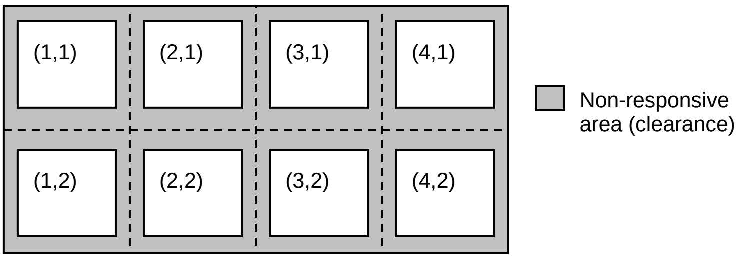
There are three control modes. The control mode can be set independently for up to four separate channels. By selecting a channel, the control mode can be easily changed during use. The control modes are explained below.

3.10.4.1 Coordinates Mode

In this mode, the touch panel is configured as a display area. The (x,y) coordinate values of the input position (in 1-pixel units) are reported.

3.10.4.2 Matrix Switch Mode

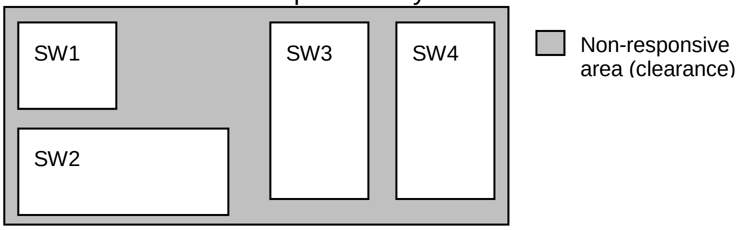
The touch panel is configured as a matrix of switch buttons, of x columns and y rows. The non-responsive (clearance) margin around each switch can also be specified.
Switch information is reported only when the area within a switch is touched.



Example: x: 4 switches, y: 2 switches

3.10.4.3 Custom Switch Mode

Touch panel is configured with switch button areas of user-specified position and size.
Switch information is reported only when the area within a switch is touched.



3.11 Touch Panel Commands

3.11.1 US P 01h n (Touch Mode Selection: Single-Touch Mode/ Multi-Touch Mode)

Code: 1Fh 50h 01h n

n: Select Touch Mode and maximum simultaneous touch detection (for Multi-Touch Mode)
00h: Single-Touch Mode
01h ≤ n ≤ 0Ah: Multi-Touch Mode (n = maximum simultaneous touches)
Default = 00h

Definable area: 00h ≤ n ≤ 0Ah

Function: Selection touch mode.

3.11.2 US P 10h ch md (Coordinates Mode)

Code: 1Fh 50h 10h ch 00h

h: Channel
md: Mode

Definable area: 00h ≤ ch ≤ 03h
md = 00h

Function: Set the specified channel to Coordinates Mode.

In Coordinates Mode, (x,y) coordinate values corresponding to the touch position are transmitted.

Transmitted data format (Single-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	00h	1 byte
(3) Data	00h–FFh	4 bytes tXL: x-coordinate, lower byte tXH: x-coordinate, upper byte tYL: y-coordinate, lower byte tYH: y-coordinate, upper byte

Touch data is transmitted when touch panel is touched.

Transmitted data format (Multi-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	10h, 11h	1 byte 10h: Released 11h: Touched
(3) Touch number	01h–0Ah	1 byte
(4) Data	00h–FFh	4 bytes tXL: x-coordinate, lower byte tXH: x-coordinate, upper byte tYL: y-coordinate, lower byte tYH: y-coordinate, upper byte

Touch data is transmitted, for each detected touch, when touch panel is touched or released, or a touch position changes.

3.11.3 US P 10h ch md nx cx ny cy (Switch Matrix Mode)

Code: 1Fh 50h 10h ch 01h nx cx ny cy

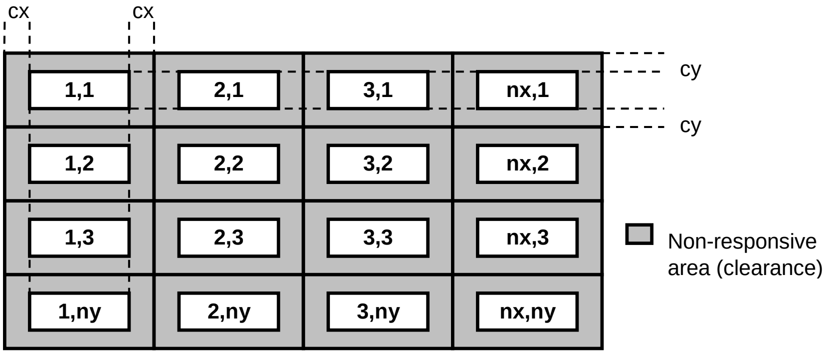
ch: Channel
md: Mode

Definable area: 00h ≤ ch ≤ 03h
md = 01h
01h ≤ nx ≤ 10h
01h ≤ cx ≤ 10h
01h ≤ ny ≤ 10h
01h ≤ cy ≤ 10h

Function: Sets the specified channel to Switch Matrix mode.

In Switch Matrix mode, the touch panel is configured as a matrix of switch buttons, specified by nx, ny. The size of the non-responsive (clearance) area in-between switch buttons is specified by parameters cx, cy.

Configuration details are shown below.



Transmitted data format (Single-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	01h	1 byte
(3) Data	01h–10h	2 bytes tX: x-direction switch number tY: y-direction switch number

Touch data is transmitted when touch panel is touched.

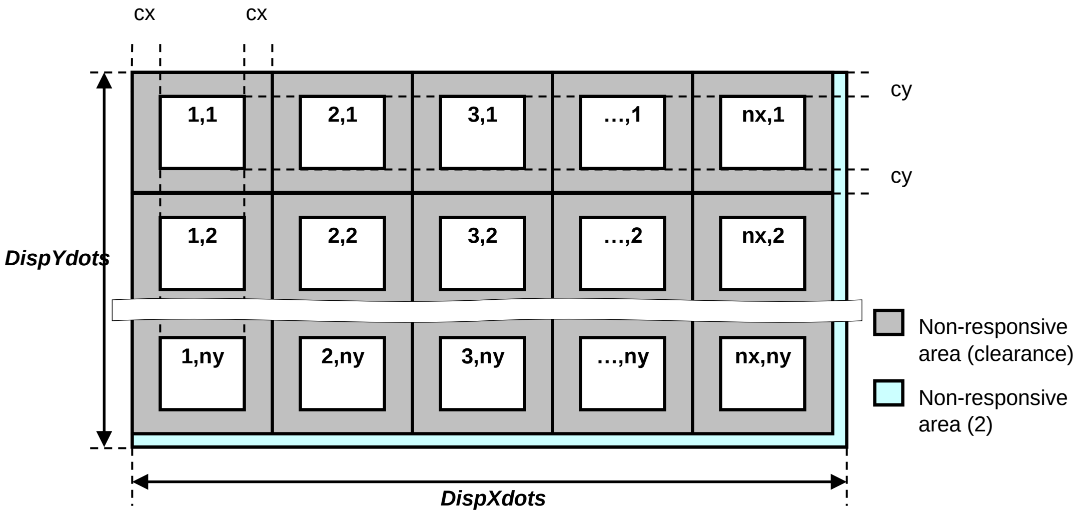
Transmitted data format (Multi-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	20h, 21h	1 byte 20h: Released 21h: Touched
(3) Data	01h–10h	2 bytes tX: x-direction switch number tY: y-direction switch number

Touch data is transmitted, for each Switch, when its touch status (touched / released) changes.

Switch button size is calculated from the supplied parameters.
Depending on the set values, the calculated size may not be an integer number of pixels.
In this case, the size of each switch button is rounded down to an integer number of pixels, and the remaining pixel area forms an extra non-responsive (clearance) area on the bottom and/or right edge (shown as (2), below).
Switch button pixel-size calculation (fractions discarded):
X (horizontal) size: (***DispXdots*** − (cx × 2 × nx)) / nx
Y (vertical) size: (***DispYdots*** − (cy × 2 × ny)) / ny

Example: ***DispXdots*** = 800, ***DispYdots*** = 480, cx = 02h, nx = 06h, cy = 04h, ny = 07h
Switch button X-size: (800 − (2 × 2 × 6)) / 6 = 776 / 6 = 129.3 → 129 pixels
Switch button Y-size: (480 − (4 × 2 × 7)) / 7 = 424 / 7 = 60.6 → 60 pixels


cx
cx
cy
cy
1,1
2,1
3,1
…,1
nx,1
1,2
2,2
3,2
…,2
nx,2
1,ny
2,ny
3,ny
…,ny
nx,ny
DispYdots
DispXdots
Non-responsive area (clearance)
Non-responsive area (2)

3.11.4 US P 10h ch md sn px1L px1H py1L py1H sx1L sx1H sy1L sy1H [... px(sn)L px(sn)H py(sn)L py(sn)H sx(sn)L sx(sn)H sy(sn)L sy(sn)H] (Custom Switch Mode)

Code: 1Fh 50h 10h ch 02h sn px1L px1H py1L py1H sx1L sx1H sy1L sy1H [... px(sn)L px(sn)H py(sn)L py(sn)H sx(sn)L sx(sn)H sy(sn)L sy(sn)H]

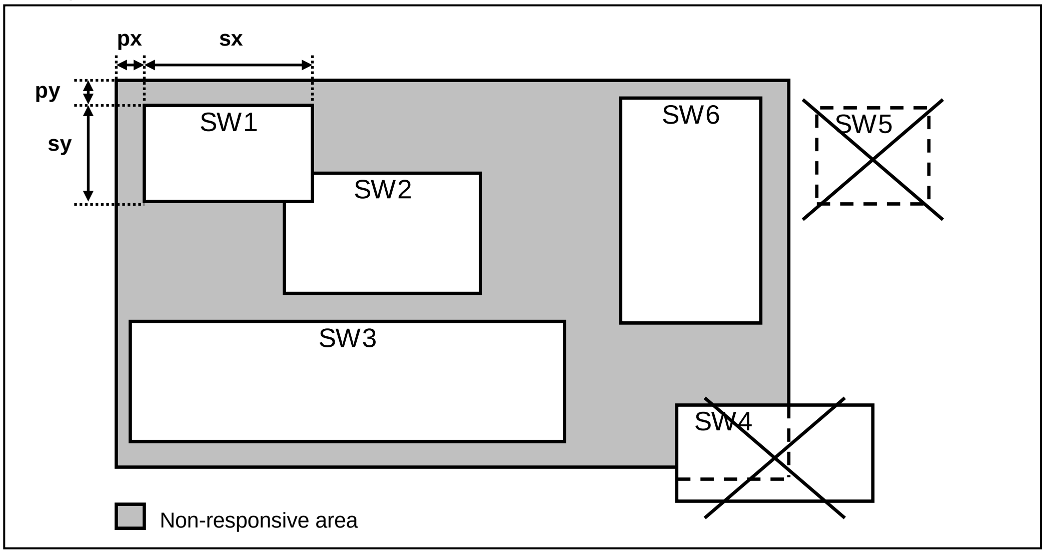
ch: Channel
md: Mode
sn: Number of Switch buttons
pxL: Switch button left horizontal position, lower byte (by 1 pixel)
pxH: Switch button left horizontal position, upper byte (by 1 pixel)
pyL: Switch button top vertical position, lower byte (by 1 pixel)
pyH: Switch button top vertical position, lower byte (by 1 pixel)
sxL: Switch button x-size, lower byte (by 1 pixel)
sxH: Switch button x-size, upper byte (by 1 pixel)
syL: Switch button y-size, lower byte (by 1 pixel)
syH: Switch button y-size, upper byte (by 1 pixel)

Definable area: 00h ≤ ch ≤ 03h
md = 02h
00h ≤ sn ≤ 20h
0000h ≤ (pxH × 100h + pxL) ≤ ***DispXdots − 1***
0000h ≤ (pyH × 100h + pyL) ≤ ***DispYdots − 1***
0001h ≤ (sxH × 100h + sxL) ≤ (***DispXdots*** − px)
0001h ≤ (syH × 100h + syL) ≤ (***DispYdots*** − py)

Function: Sets the specified channel to Custom Switch Mode.

In Custom Switch Mode, the touch panel is configured with a user-defined number of switch buttons, each of user-defined size and position.
Any remaining area is non-responsive.

Configuration details are shown below.



Switch button areas that overlap may be defined, however only the lowest switch button number is responsive in any overlapped area.
If the start position (px,py) and/or size (sx,sy) for a switch button is outside the defined area, that switch button number is invalid.

Transmitted data format (Single-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	02h	1 byte
(3) Data	01h–20h	1 byte tn: Switch button number

Touch data is transmitted when touch panel is touched.

Transmitted data format (Multi-Touch Mode):

Transmitted data	Hex	Data length
(1) Header	10h	1 byte
(2) Identifier	30h, 31h	1 byte 30h: Released 31h: Touched
(3) Data	01h–20h	1 bytes tn: Switch button number

Touch data is transmitted, for each Switch, when its touch status (touched / released) changes.

3.11.5 US P 20h m (Touch Panel Data Transmit ON/OFF)

Code: 1Fh 50h 20h m

m: Transmit ON/OFF

Definable area: m = 00h, 01h
m = 00h: Transmit OFF
m = 01h: Transmit ON

Default: m = 00h (Transmit OFF)

Function: Sets whether or not touch operation data is transmitted to the host.

When OFF, touch operation data is not placed in the transmit buffer.
Setting is not applicable for reading touch data within a Program Macro.

3.11.6 US P 21h ch (Touch Panel Channel Select)

Code: 1Fh 50h 21h ch

ch: Channel number

Definable area: 00h ≤ ch ≤ 03h

Default: ch = 00h

Function: Selects the currently-active touch panel control channel.

If channel is undefined, Coordinates mode is selected.

3.11.7 US K 70h （Touch Parameter Setting)

Code: 1Fh 4Bh 70h a [b [c]]

a : parameter selection/ operation designation
b, c : value

Definable area:
a = 00h : Threshold Setting (c is unnecessary)
00h ≤ b ≤ FFh : threshold value
a = 04h : Gain Setting (c is unnecessary)
00h ≤ b ≤ 0Fh : gain value
a = 06h : Touch Standard Reference Setting
00h ≤ b ≤ FFh : maximum noise value
(If something channel over this value, this command will be failed.)
00h ≤ c ≤ FFh : designation of measurement numbers
a = 07h : Touch Standard Reference Use (valid/ invalid) (c is unnecessary)
00h ≤ b ≤ 01h : 00h (invalid), 01h (valid)
a = 08h : Touch Standard Reference Status Read (b, c are unnecessary)

Function: Touch parameter setting.

Threshold and gain (a= 00h/ a = 04h)
These commands for adjusting touch sensitivities.
Once decreacing threshold value, the sensitivity will be dull. Once increasing threshold value, it will be sensitive.
Since the optimum gain value depends on the features of touch sencers, there is no need to change it from the default value.
Default value is set Memory SW Setting (threshold = MSW59, gain = MSW58)

Touch standard reference rerated command (a= 06h/ 07h/ 08h)

Touch standard reference function is an option prepared to improve the reliability of touch detection in the environment changing environment such as water. In order to improve the reliability of touch detection using this function, it is necessary to execute "touch standard reference setting" once (see below) for each module and confirm the result of "success". In order to set up an accurate reference, it is important to understand that "the environment being controlled" (that is, the state where the module is mounted on the actual used casing and cover glass, etc. without being hit by touch, moisture or foreign matter)) (For example, the final stage of the product assembly process).
If touch standard reference setting is done in advance as mentioned above, this function can be enabled with "touch standard reference reference" (see below).
If there is no standard reference setting, or even if there is "touch standard reference use" is not enabled, this function will be invalid (touch detection will be base level performance in this case).

a = 06h: Touch standard reference setting

In order to measure an accurate reference value, the reference setting is defined as the "controlled environment" (that is, the influence of touch / moisture and foreign object contact / noise on the actual use state (casing or cover glass) of the product. Please do it in an environment not received).
When the touch standard reference setting command is executed, the touch panel is measured twice and if the difference (noise level) between the first and second measurement points (channels) is less than b, the measurement is "successful". If exceeded, the measurement is "failed". The noise level of the channel with the most noise and the x, y sensor position of that channel are notified to the response data.
If the measurement is successful, touch standard reference use is set to "valid", if it fails, set to "invalid".
The reference value is saved in the touch controller, but the touch standard reference setting setting is not saved, so you need to use the touch standard reference use (valid) command to use it after reset or restart.

Response data (4 bytes)

00h NNh NXh NYh = Success (the amount of noise is within the limit)
01h NNh NXh NYh = Failure (amount of noise exceeds the limit)
02h XXh XXh XXh = Failure (other problems / defects)
NNh = noise amount of the channel with the largest amount of noise
NXh = X of the channel with the greatest amount of noise
NYh = Y of the channel with the largest amount of noise

Transmit data	Hex	Data length
Status	00h～02h	1 byte
NNh / XXh	00h～FFh	1 byte
NXh / XXh	00h～FFh	1 byte
NYh / XXh	00h～FFh	1 byte

a = 07h: Touch Standard Reference Use (valid / invalid) (c is unnecessary)

b = 00h: Invalid (Initial value)
b = 01h: valid

Note: Even if it is set to "Enabled", there is no effect if a valid reference value is not stored in the touch controller with the above "Touch Standard Reference Setting" command.

a = 08h: Touch Standard Reference Use Status Read (b, c are unnecessary)

Response data (1 byte): 00h = invalid 01h = valid

Note: The following data will be transmitted from the interface that is currently enabled.

Transmit data	Hex	Data length
Data	00h/01h	1 byte

4 Setup

4.1 Jumper

JP No.	Function	Default
J0	I2C Slave Address	OPEN
J1		OPEN
J2	Reserved	OPEN
J3	Baud Rate	OPEN
J4		OPEN
J5	Serial Interface mode	OPEN
J6		OPEN
J7	Program Macro Start	OPEN
J8	Reserved	OPEN
J9	Reserved	OPEN
J10	Touch panel calibration mode	OPEN

4.1.1 Display Address (I2C Interface)

J0	J1	Function
OPEN	OPEN	I2C Slave Adreess = 50h
SHORT	OPEN	I2C Slave Adreess = 51h
OPEN	SHORT	I2C Slave Adreess = 52h
SHORT	SHORT	I2C Slave Adreess = 53h

4.1.2 Baud Rate (UART Interface)

J3	J4	Function
OPEN	OPEN	38,400bps
SHORT	OPEN	19,200bps
OPEN	SHORT	Apply Memory SW Setting (Switch No. 48)
SHORT	SHORT	115,200bps

For Memory SW setting details, refer to 4.2 Memory SW.

4.1.1 Serial Interface type

J5	J6	Function
OPEN	OPEN	SPI Mode
OPEN	SHORT	Reserved
SHORT	OPEN	I2C Mode
SHORT	SHORT	UART Mode

4.1.2 Program Macro Start

J7	Function
OPEN	Apply Memory SW Setting (Switch No. 19)
SHORT	No start

4.1.3 Operating Mode

J10	TEST Pin	Function
-	L	Test Mode
OPEN	H (NC)	Normal Command mode
SHORT	H (NC)	Touch panel calibration mode

4.2 Memory SW

Switch No.	Function	Valid range	Default
0	International font set	00h–0Dh	00h
1	Character table type	00h–05h,10h–13h,FEh, FFh	00h
2	Horizontal scroll speed	00h–1Fh	00h
3	Reverse display	00h,01h	00h
4	Write mixture display mode	10h,11h	10h
5	Brightness level setting	00h–FFh	FFh
6	Reserved	-	-
7	Write screen mode	00h,01h	00h
8	Font size	01h,02h,03h,04h	01h
9	2-byte character	00h,01h	00h
10	Font magnification X	01h–04h	01h
11	Font magnification Y	01h–04h	01h
12	Character style	00h–03h	00h
13	2-byte character type	00h–03h	00h
14,15	Reserved	-	-
16	Download character restore at power-on (FROM→RAM) 6×8 pixel (00h = Don't restore)	00h,01h	00h
17	Download character restore at power-on (FROM→RAM) 8×16 pixel (00h = Don't restore) *1	00h,01h	00h
18	Download character restore at power-on (FROM→RAM) 16×16 pixel (00h = Don't restore) *2	00h,01h	00h
19	FROM Macro execution at power-on (00h = Not execute)	00h–04h,81h–84h	00h
20–47	Reserved	-	-
48	UART Baud rate setting 00h: 19200bps (default) 01h: 4800bps 02h: 9600bps 03h: 19200bps 04h: 38400bps 05h: 57600bps 06h: 115200bps	00h–06h	00h
49	UART Parity 00h: None 01h: Even 02h: Odd	00h–02h	00h
50,51	Reserved	-	-
52	Macro end code Enable/Disable	00h,01h	00h
53	Macro end code	00h–FFh	00h
54	Macro end Clear Screen setting	00h,01h	00h
55–57	Reserved	-	-
58	Touch sensitivity (signal gain) setting	00h–0Fh	06h
59	Touch sensitivity (threshold) setting	00h–FFh	50h
60–63	Reserved	-	-

Note: Module operates with default value if Memory SW value is outside the valid range.

*1: If setting is 01h, download characters for 12×24 pixel and 16×32 pixel are also restored.
*2: If setting is 01h, download characters for 32×32 pixel are also restored.

Revision Note

Specification No.	Date	Revision
DS-2007-0003-00	Jun. 27, 2017	Initial release