

Content

1 Serial Port Description.....	3
1.1 Working mode.....	3
1.2 Data frame structure.....	3
1.3 Communication frame buffer(FIFO).....	3
1.4 Byte transmission sequence.....	3
1.5 Transmission direction.....	3
2. Command Description.....	4
2.1 Handshake(0x00).....	4
2.2 Set the current color palette (0x40).....	4
2.3 Set character spacing(0x41).....	5
2.4 Access specified position's color(0x42,0x43).....	5
2.5 Cursor display (0x44).....	5
2.6 Text display(0x53,0x98,0x45).....	5
2.6.1 ASCII of 8*8 text display(0x53).....	6
2.6.2 Self-made font text display(0x98).....	6
2.6.3 Set and cancel textbox (0x45).....	6
2.7 Point display(0x50,0x51,0x74,0x72).....	8
2.7.1 Set point (0x50, 0x51).....	8
2.7.2 Dynamic curve display (0x74).....	9
2.7.3 Direct video memory operation (0x72).....	9
2.8 Line Display(0x56, 0x5D, 0x75, 0x76,0x78).....	10
2.8.1 Connect specified points (0x56,0x5D).....	10
2.8.2 Spectrum display (0x75).....	11
2.8.3 Broken line display (0x76).....	11
2.8.4 Line display as per offset (0x78).....	11
2.9 Arc curve display(0x57).....	11
2.9.1 Arc and circular domain display (0x57).....	11
2.9.2 Arc segment display (0x5704).....	12
2.10 Area display(0x59,0x69,0x5A,0x5B,0x5C,0x64).....	13
2.10.1 Rectangle box or rectangle domain display(0x59,0x69,0x5A,0x5B,0x5C).....	13
2.10.2 Area fill(0x64).....	13
2.11 Clear screen (0x52).....	14
2.12 Movement of the area selected (0x60,0x61,0x62,0x63).....	14
2.13 Image or icon display(0x70,0x71,0x7B,0x9C,0x9D,0x99,0xE2).....	15
2.13.1 Image display (0x70).....	15
2.13.2 Display one image and figure up (0x7B).....	15
2.13.3 Cut icon(0x71,0x9C,0x9D).....	15
2.13.4 User-defined icon display (0x99).....	16
2.13.5 Save the image on the current screen into the flash(0xE2).....	16
2.14 Backlight&brightness control(0x5E, 0x5F).....	16
2.14.1 Turn off backlight (0x5E).....	16
2.14.2 Set touchpad(keyed) backlight mode (0x5E).....	17
2.14.3 Set the backlight to maximum brightness (0x5F).....	17
2.14.4 Adjust backlight brightness (0x5F).....	17
2.15 Touch panel operation (0x72,0x73,0x78, 0x79,0xE4).....	17
2.15.1 Upload touch coordinate automatically (0x72,0x73).....	17
2.15.2 Upload touch code automatically (0x78, 0x79).....	18

Content

2.15.3 Touch panel calibration (0xE4)	18
2.16 Working mode configuration(0xE0)	18
2.17 Execute commands periodically and circularly(0x9A)	19
2.17.1 Turn on the function of executing commands periodically and circularly	19
2.17.2 Turn off the function of executing commands periodically and circularly	20
2.18 Random access memory operation (0xC0,0xC1,0xC2)	20
2.18.1 Write in random access memory (0xC0)	20
2.18.2 Read from random access memory(0xC2)	20
2.18.3 Set point using data in random access memory (0xC101)	20
2.18.4 Connect lines using data in random access memory (0xC102)	20
2.18.5 Display line chart using data in random access memory (0xC103)	21
2.18.6 Display line chart at high speed using data in random access memory (0xC104)	21
2.18.7 Scale broken line graph using data in random access memory (0xC105)	22
2.18.8 Scale window limit two-way broken line graph using data in random access memory (0xC106)	22
2.18.9 Take random access memory as buffer area for setting point (0xC107)	23
2.18.10 Display multi-parameters in random access memory area (0xC108)	24
2.18.11 Buffer command to realize displaying synchronously using random access memory (0xC110)	25
2.19 Keyboard operation(0x71,0xE5)	25
2.19.1 Upload key code (0x71)	25
2.19.2 Set key code (0xE5)	25
2.20 Read and write in user's memory (0x90,0x91)	25
2.20.1 Write in random data memory (0x90 64KB)	25
2.20.2 Write in sequence data memory (0x90 30MB)	26
2.20.3 Read data memory (0x91)	26
2.21 Download font file and configuration files (0xF2)	26
2.22 Simple algorithm support (0xB0)	26
2.22.1 MAC calculate (0xB002)	26
2.22.2 Array sorting (0xB003)	26
2.23 Buzzer control (0x79)	27
2.24 Date and time display (0x9B,0xE7)	27
2.24.1 Turn off date and time display(0x9B00)	27
2.24.2 Turn on date and time display(0x9BFF)	27
2.24.3 Adjust date and time(0xE7)	27
2.24.4 Read current date and time (the Gregorian calendar) (0x9B5A)	27
2.25 Configuration files	28
2.25.1 Touchpad interface transfer (Configuration file 0x1E)	28
2.25.2 Keyed interface transfer(Configuration file 0x1B)	31
2.25.3 Automatically and circularly execute command groups (Configuration file 0x1C)	32
2.25.4 Icon display(Configuration file 0x1D)	33
2.26 Upgrade full-screen display once(0x0D)	33

1 Serial Port Description

1.1 Working mode of serial port

Asynchronous, UART serial port is applied in our intelligent UART LCD Module with serial port mode of 8n1, i.e. 10 bits for every data transmission: one start bit, 8 data bits (LSB first) and one stop bit.

When powered on, if the terminal I/O 0 pin is high-level or floating, serial port baud rate (from 1200 to 115200 bps) is set by users in advance. (Please refer to Command 0xE0 for more details about the setting.)

When powered on, if the terminal I/O 0 pin is low-level, serial port baud rate is fixed at 921600 bps.

Note: I/O pin is:

set by jumper pad, in case of display modules of T series;

set by dip switch or jumper or the same to K series, in case of display modules of S series;

set by the status of USB port: when USB port is connected (or disconnected), I/O pin is low (or high), in case of display modules of K series.

1.2 Data frame structure

The standard data frame consists of 4 data blocks, as shown in the table below:

Data Block	1	2	3	4
Examples	0xAA	0x70	0x01	0xCC 0x33 0xC3 0x3C
Description	Frame header	command	Data, maximum 249 bytes	frame end

1.3 Communication frame buffer (FIFO)

There is a 24 Rx frame communication frame buffer with FIFO structure. Therefore, users can continuously send data to display module as long as the Rx frame doesn't overflow.

The display module has a hardware pin (namely /Busy in user's interface) to indicate the FIFO status. Normally, /Busy pin keeps high (RS232 is negative); when there is only one frame buffer area left, /Busy pin will go to low level. (RS232 is positive)

For general applications, as the processing speed is very fast, it is not necessary for the users to judge the status of /Busy pin.

However, for applications of sending multiple data frame within a short time, such as to rapidly refresh hundreds of monitor parameters at the same time, users are advised to use /Busy signal for controlling serial port; and when /Busy signal is low, stop sending data to HMI.

Note: If "drop-frame" phenomenon (some data are missing) occurs in the process of using display module, overflowing of buffers may be the reason. Oscilloscope can help to inspect whether Busy signal alters. If so, slow down the transmitting speed or add hardware inspection to deal with Busy signal.

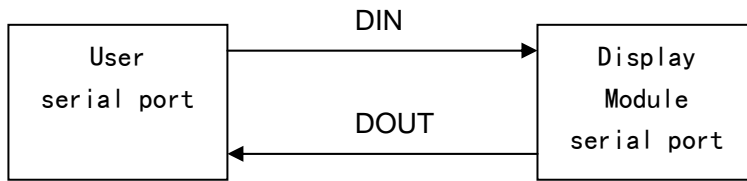
1.4 Byte transmission sequence

The format of all commands is Hex. For typestyle data (2 bytes), it is transmitted as per MSB. For example, X coordinate is 100, whose Hex format is 0x0064, then when X is transmitted, the proper sequence shall be 0x00 0x64.

1.5 Transmission direction

Transmission direction is defined as follows:

- Downstream (Tx) when user sends data to display module, data is input via DIN pin;
- Upstream (Rx) when display module sends data to user, data is output from DOUT pin.



2. Command Description

2.1 Handshake (0x00)

Tx: AA 00 CC 33 C3 3C

Rx: AA 00 'OK V*. *' P1 P2 P3 <Pic ID> CC 33 C3 3C

- OK_V*. *, *. * is the current version of display module
- P1 is the TFT ID of the current display module (Please refer to 0xE0 command for specific parameters)
- P2 is the serial baud rate set by current users
- P3 is setting mode of touch panel, buzzer
- Pic _ID: ID of the picture on the current screen.

Note: Display module needs 0.5~2 seconds for initialization and will not respond to any command until the initialization is finished. So users can confirm whether the initialization process is finished by sending handshake command.

2.2 Set the current color palette (0x40)

TX: AA 40 <FC> <BC> CC 33 C3 3C

Rx: None

- <FC> Foreground color palette, 2 bytes(16 bit, 65k color), default reset value is 0xFFFF(white)
- <BC> Background color palette, 2 bytes(16 bit, 65k color), default reset value is 0x001F(blue)
- color palette of 16 bit is defined to 5R6G5B mode as per the following table:

16 bit definitions of Color Palette																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Defin e	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
	Red 0xF800					Green 0x07E0						Blue 0x001F				



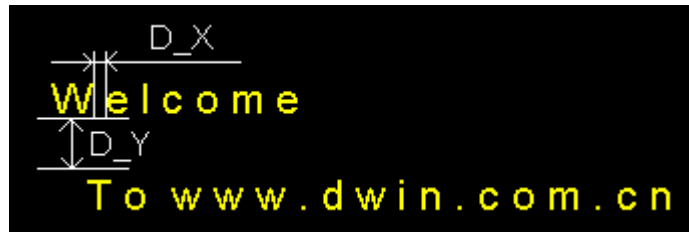
Note: Once set, the palette will remain that way unless it is reset; it will restore the default values once display module hardware resets.

2.3 Set character spacing (0x41)

Tx: AA 41 <D_X> <D_Y> CC 33 C3 3C

Rx: none

- <D_X> Character spacing in X direction (column spacing), ranging from 0x00 to 0x7F. Default reset value is 0x00.
- <D_Y> Character spacing in Y direction (line spacing), ranging from 0x00 to 0x7F. Default reset value is 0x00.



Note: Once set, the spacing will remain that way unless it is reset; it will restore the default values once HMI hardware resets.

2.4 Access specified point's color(0x42, 0x43)

Tx: AA <CMD> <X> <Y> CC 33 C3 3C

Rx: None

- <CMD>
 - 0x42 Access specified point's color as background color
 - 0x43 Access specified point's color as foreground color
- <X> <Y> Specified point's coordinate (2 bytes).

e.g. AA 42 00 10 01 00 CC 33 C3 3C

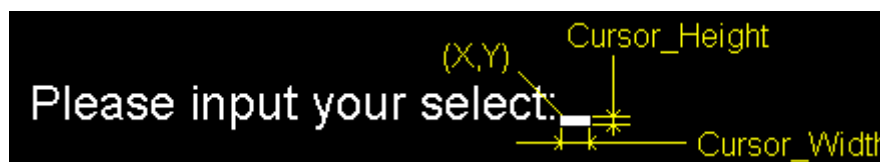
Access color of position x=16 (0x0010) y=256 (0x0100) as background color.

2.5 Cursor display (0x44)

Tx: AA 44 <Cursor_EN> <X> <Y> <Cursor_Width> <Cursor_Height> CC 33 C3 3C

Rx: None

- <Cursor_EN>
 - 0x01 Turn on cursor display on and the cursor will display at (x, y)
 - 0x00 Turn off cursor display
- <X>, <Y> Coordinate of the left top corner of the cursor
- <Cursor_Width> Width of cursor, ranging from 0x01 to 0x1F
- <Cursor_Height> Height of cursor, ranging from 0x01 to 0x1F



Note: If cursor display is turned off, other parameters of this command are of no use.

e.g. AA 44 01 00 80 00 60 10 03 CC 33 C3 3C

Display a cursor of 16*3 dot matrix at (128,96)

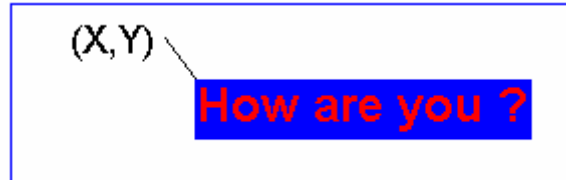
2.6 Text display(0x53, 0x98)

2.6.1 ASCII of 8*8 text display (0x53)

Tx: 53 <X> <Y> <String> CC 33 C3 3C

Rx: None

- 0x53 Display ASCII character of 8*8 dot matrix
- <X> <Y> Coordinate of the left top corner of the first character
- <String> String to be displayed. Color of the string is set by command 0x40 and characters spacing is set by command 0x41; auto-wrap to next line at the end of line. 0x0D and 0x0A indicate "back to line start" and "wrap to next line"



e.g. AA 53 00 80 00 30 48 6F 77 20 61 72 65 20 79 6F 75 20 3F CC 33 C3 3C
 Display string "How are you?" of 8*8 dot matrix at (128,48)

2.6.2 Self-made font text display (0x98)

Tx: AA 98 <X> <Y> <Lib_ID> <C_mode> <C_dots> <F_color> <B_color> <String> CC 33 C3 3C

Rx: None

- 0x98 Text display as per user's specific requirements.
- <X> <Y> Coordinate of the left top corner of the first character
- <Lib_ID> Character Library ID, ranging from 0x00 to 0x3B, which is set by the user through Command 0xF2 when saving the newly-made font files into the display module.

Note: 32 MB in the flash is available for 60 character libraries of different sizes. The definition of the character library is as follows:

Lib-ID	Capacity	Description	Default Lib_ID
0x00-0x1F	128KB	32 small character library files of maximum 128KB.	0x00=ASCII character library. (Please do not change the value unless necessary. 0x02-0x1F: None
0x20-0x3B	1MB	28 character library files of maximum 1MB.	0x20,0x21, 0x22 and 0x23 are for Chinese language. 0x24-0x3B: None

- <C_Mode> to set Text Display Mode and Encoding Mode as follows:

Bit	.7-.4	.3-.0
Definition	Display Mode	Character Encoding Method
Description	.7=1 display foreground color =0 not display foreground color	0x00 8bit Font file contains no more than 256 bytes
	.6=1 display background color =0 not display background color	0x01 GB2312 Chinese internal code 0x02 GBK Chinese extended internal code or Korea HANGUL code
	.5=1 display text 0° =0 display text 90°	0x03 BIG5 Traditional Chinese Code 0x04 SJIS Japanese Code
	.4=1 adjust character spacing automatically	0x05 UNICODE UNICODE Code (UTF16)
	.4=0 display spacing as per setting	0x06-0x0F: Undefined

Note: If you don't need the background color, the high bit (7-4) shall be 1000 which will be 8 after being converted into Hex. For the low bit (3-0), please always choose "0x00 8 bit". After 00 is converted into Hex, the figure shall be 0. So, you should input "80" as text display mode and encoding code.



Samples of Display Mode Setting:

Foreground color	ON	ON	OFF
Background color	ON	OFF	ON
<C _mode>high value 2bit	1100-0xC*	1000-0x8*	0100-0x4*
Character display effect	B area	A area	C area

➤ <C _dots> character size setting

C _Dots	Character Library Type (C_Mode set the lower 4bit)		
	When Character Encoding Method = 0x00 or 0x05	When Character Encoding Method = 0x01-0x04	
		ASCII Character	Non-ASCII Character
0x00	8*8	6*12	12*12
0x01	6*12	8*16	16*16
0x02	8*16	12*24	24*24
0x03	12*24	16*32	32*32
0x04	16*32	20*40	40*40
0x05	20*40	24*48	48*48
0x06	24*48	28*56	56*56
0x07	28*58	32*64	64*64
0x08	32*64	—	—
0x09	12*12	—	—
0x0A	16*16	—	—
0x0B	24*24	—	—
0x0C	32*32	—	—
0x0D	40*40	—	—
0x0E	48*48	—	—
0x0F	56*56	—	—
0x10	64*64	—	—

0x11	40*80	—	—
0x12	48*96	—	—
0x13	56*112	—	—
0x14	64*128	—	—
0x15	80*80	—	—
0x16	96*96	—	—
0x17	112*112	—	—
0x18	128*128	—	—
0x19	6*8	—	—
0x1A	8*10	-	-
0x1B	8*12	-	-
0x1C	100*200	-	-
0x1D	200*200	-	-

- <Fcolor> Foreground color of character
- <Bcolor> Background color of character
- <String> String data, display spacing is set by command 0x41 and automatically wrap at the end of a line.

Note: If users need to display ASCII character with their libraries (other libraries coded with 8 bit) by command 0x98, the coding method 0x01-0x04 (GB2312, GBK, BIG5, SJIS) in the table can not be used because ASCII character will be shown with font library 0x00 automatically in case of the four coding methods.

e.g.: AA 98 00 80 00 30 00 C0 04 F8 00 00 1F 48 6F 77 20 61 72 65 20 79 6F 75 20 3F CC 33 C3 3C

Display "How are you?" of 8*8 dot matrix at coordinate (128,128). The string is from font library 0x00 for ASCII. The foreground color is set as red and background color is set as white while there is no display of background color.

2.6.3 Set and cancel textbox (0x45)

Set textbox

Tx: AA 45 <Xs> <Ys> <Xe> <Ye> CC 33 C3 3C

Rx: None

- <Xs> <Ys> <Xe> <Ye> Left top and right bottom coordinates of the textbox

Note: After the textbox is set, line wrap can be realized when displaying text by command 0x98 as long as the the first character is within the textbox.

Cancel textbox

Tx: AA 45 00 CC 33 C3 3C

Rx: None

2.7 Point display(0x50, 0x51, 0x74, 0x72)

2.7.1 Set point (0x50, 0x51)

Tx: AA <CMD> <(x0, y0) (x1,y1)(xi, yi)> CC 33 C3 3C

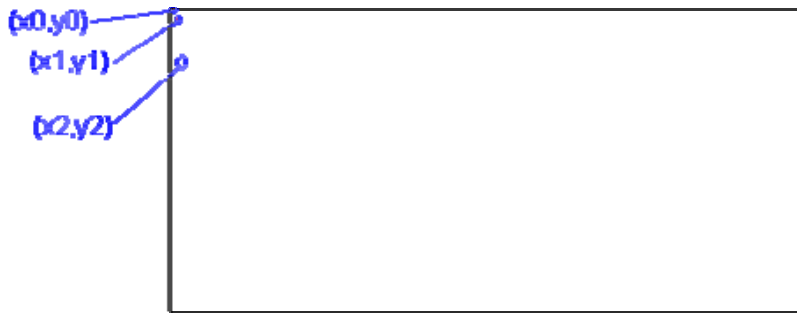
Rx: None

- <CMD>
 - 0x50: Display point using background color (delete point)
 - 0x51: Display point using foreground color (set point)
- <(x0,y0) (x1,y1)(xi,yi)> Coordinates of the points to be set. The maximum point

number of one frame is 62.

e.g. AA 51 00 00 00 00 03 00 06 00 05 00 20 CC 33 C3 3C

Display 3 points using foreground color at (0, 0), (3, 6), (5, 32) as follows:



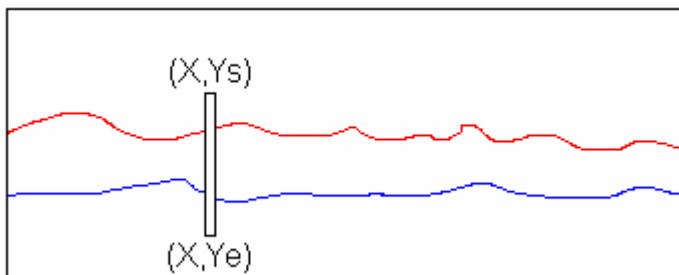
2.7.2 Dynamic curve display (0x74)

Tx: AA 74 <X> <Ys> <Ye> <Bcolor> <(Y0,Fcolor0), (Y1,Fcolor1).....(Yi, Fcolor i)> CC 33 C3 3C

Rx: none

This command is mainly used for displaying multi-dynamic curves. The following steps shall be followed when the display module receives the command 0x74:

Step 1: Erase the vertical line from (X,Ys) to (X ,Ye), using specified color<Bcolor>. Empty the original content.



Step 2: Set point at (X,Yi) using specified color<Fcolori>.



Note: This command will not change Palette's default setting.

2.7.3 Direct video memory operation (0x72)

Tx: AA 72 <Address_H:M:L> <Pixel_data0.....Pixel_data n> CC 33 C3 3C

Rx: None

- 0x 72 Command for sending data into video memory
- Address_H:M:L Starting address of video memory

Calculation of video memory address

If horizontal pixel (X direction) is H and vertical pixel(Y direction) is V, the video memory

address of the dot (x, y) is calculated like this: $Adr=(y*H+x)$.

For example, the video memory address of (150,100) on the screen of display module whose resolution is 640×480 is $100*640+150=64150(0XFA96)$.

➤ Pixel_data0.....Pixel_data n Data to be sent into the video memory

Our display module is of 65K color and every pixel dot consists of data of two bytes (16 bits). The display data is in conformity with color palette 5R6G5B and it is sent as MSB (Most Significant Bit).

For example, if you need to display a red dot at (150,100), please send the command as follows: AA 72 00 FA 96 F8 00 CC 33 C3 3C.

Note:

★Line width and line wrap

Command 0x72 can send maximum 248 bytes (124 continuous dots). Therefore, when the actual pixel dots are more than 124, the data shall be sent by a couple of frames separately. After line wrap, as the video memory address is not continuous, the coordinate of next video memory address shall be re-calculated before new data is sent.

★Speed and time for download

The speed of serial port data communication is set by the user. The maximum value is 921600bps. For a $H \times V$ picture, the approximate time is

$T=(H \times V \times 2) / ((B \times 0.1) \times 0.9)$ second.

★This command is mainly used to realize real-time upgrade of pictures. Used together with Command 0XE2, pictures can be downloaded into the display module through serial port.

2.8 Line display (0x56, 0x5D, 0x75, 0x76, 0x78)

2.8.1 Connect specified points (0x56, 0x5D)

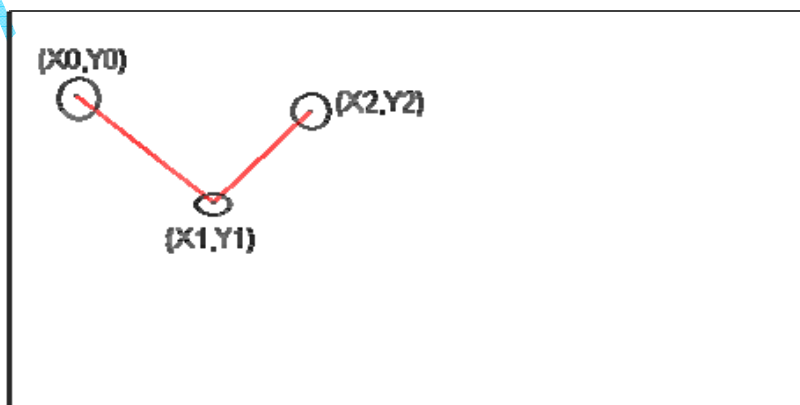
Tx: AA <CMD> <(x0,y0) (x1,y1)(xi, yi)> CC 33 C3 3C

Rx: None

- <CMD>
 - 0x56 Connect specified points by lines, using foreground color (set by command 0x40).
 - 0x5D Connect specified points by lines, using background color (set by command 0x40).
- <(x0,y0) (x1,y1)(xi,yi)> Coordinates of specified points.

e.g. AA 56 00 28 00 32 00 78 00 70 00 B1 00 3A CC 33 C3 3C

Connect 3 points, (40,50),(120,112),(177,58) to lines, using foreground color, as follows:



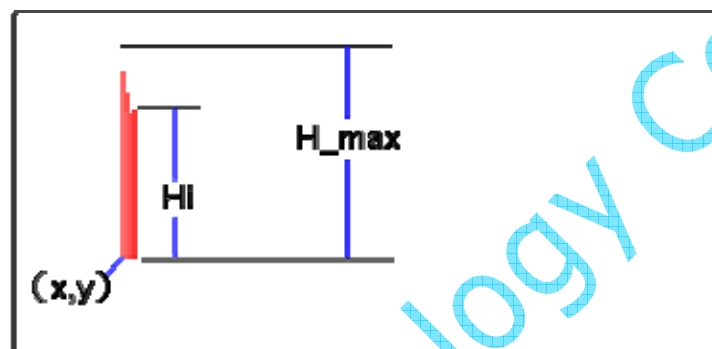
2.8.2 Spectrum display (0x75)

Tx: AA 75 <(x, y)>, <H_max>, <H0.....Hi> CC 33 C3 3C

Rx: None

- <(x, y)> X is starting coordinate on spectrum's X-axis. $x=x+1$ after displaying another spectrum line; Y is horizontal position on spectrum. Starting and ending coordinates of every spectrum line's are y and (y-Hi) respectively.
- <H_max> Maximum height of a spectrum line.
If $H_max=0x01-0xFF$, spectrum line's height H_i is a variable of 1 byte.
If $H_max=0x00$, the following 2 bytes represent Hmax. H_i is a variable of 2 bytes.
- <H0.....Hi> Height of one spectrum line, one or two bytes.

Note: Spectrum line's color is set by Command 0x40 .Spectrum line (height H_i) will be displayed with foreground color, other spectrum lines (H_max-H_i) will be displayed (be erased) by background color.



2.8.3 Broken line display (0x76)

Tx: AA 76 <x>, <x_dis>, <Y0.....Yi> CC 33 C3 3C

Rx: None

- <x> Starting coordinate on broken line's X-axis. $x=x+x_dis$ after one more point is connected.
- <x_dis> Increment of coordinate on X-axis.
- <Y0.....Yi> Apex coordinate of the broken line. The points are connected with foreground color.(Use Command 0x40,0x42,0x43 to change the foreground color when necessary.)

e.g. AA 76 00 32 10 00 20 00 10 00 20 00 30 00 40 00 50 00 60 00 70 00 80 CC 33 C3 3C

Note: This command is similar to command 0x56 except that coordinate X here is calculated automatically, which improves connecting speed.

2.8.4 Line display as per offset (0x78)

Tx: AA 78 <x,y>, <dx0,dy0>, <dx1,dy1>,.....<dxn,dyn> CC 33 C3 3C

Rx: None

- <x,y> Starting coordinate.
- <dxn,dyn> 1-byte offset of x,y. The highest bit (.7) is sign bit, "1" means "negative"

2.9 Arc curve display (0x57)

2.9.1 Arc and circular domain display (0x57)

Tx: AA 57 (<Type 0> <X 0> <Y 0> <R 0>)(<Type i> <X i> <Y i> <R i>) CC 33 C3

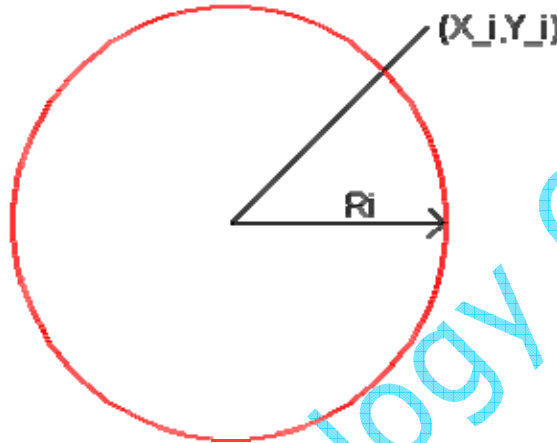
3C

Rx: None

- <Type_i>
 - 0x00 Display specified arc, using invert color
 - 0x01 Display specified arc, using foreground color (set by command 0x40)
 - 0x02 Display specified circular domain, using invert color
 - 0x03 Fill specified circular domain, using foreground color (set by command 0x40)
- <X_i> <Y_i> Coordinates of the center of arc or circular domain
- <R_i> Radius of the arc or circular domain, ranging from 0x01 to 0xFF

e.g. AA 57 01 00 60 00 60 40 CC 33 C3 3C

Display an arc, using foreground color, with the center (96,96) and radius 64 as follows:



2.9.2 Arc segment display (0x5704)

Tx: AA 57 04 <X><Y><R><A_S><A_E>CC 33 C3 3C

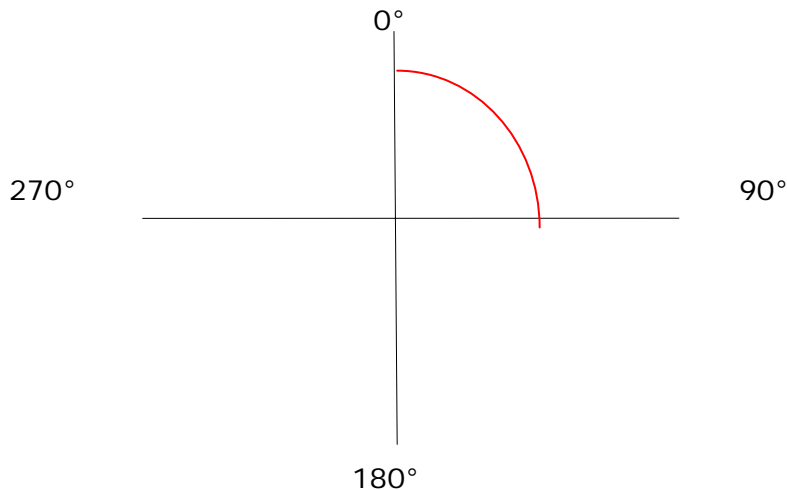
Rx: None

- <X><Y> Coordinates of the center of arc
- <R> Radius of the arc, ranging 0x0001-0x03FF
- <A_S> starting angle of arc, ranging 0x00-0x02D0(0-720), 0.5° as the unit
- <A_E> Ending angle of arc, ranging 0x00-0x02D0(0-720), 0.5° as the unit

Note: Display color is the foreground set by Command 0x40.

e.g. AA 57 04 00 60 00 60 00 40 00 00 00 B4 CC 33 C3 3C

Display an arc, using foreground color, with the center (96,96) and radius 64 ,starting angle 0°, ending angle 90° as follows:



2.10 Area display

2.10.1 Rectangle box or rectangle domain display (0x59, 0x69, 0x5A, 0x5B, 0x5C, 0x6C)

Tx: AA <CMD> (<Xs 0> <Ys 0> <Xe 0><Ye 0>)(<Xs i> <Ys i> <Xe i><Ye i>)

CC 33 C3 3C

Rx: None

- <CMD >
 - 0x59 Display rectangle box, using foreground color (set by command 0x40); line width is 1 dot matrix.
 - 0x69 Display rectangle box, using background color (set by command 0x40); line width is 1 dot matrix.
 - 0x5A Fill rectangle domain, using background color (set by command 0x40)
 - 0x5B Fill rectangle domain, using foreground color (set by command 0x40)
 - 0x5C Display specified rectangle domain, using invert color (XOR 0xFF operation).
- <Xs_i> <Ys_i> Left top coordinate of rectangle box or domain
- <Xe_i> <Ye_i> Right bottom coordinate of rectangle box or domain.

e.g. AA 5C 00 40 00 40 00 80 00 80 CC 33 C3 3C

Invert color of rectangle domain specified by top left coordinates (64,64) and bottom right coordinate (128,128) as follows:



2.10.2 Area fill (0x64)

Tx: AA 64 <x> <y> <color> CC 33 C3 3C

Rx: None

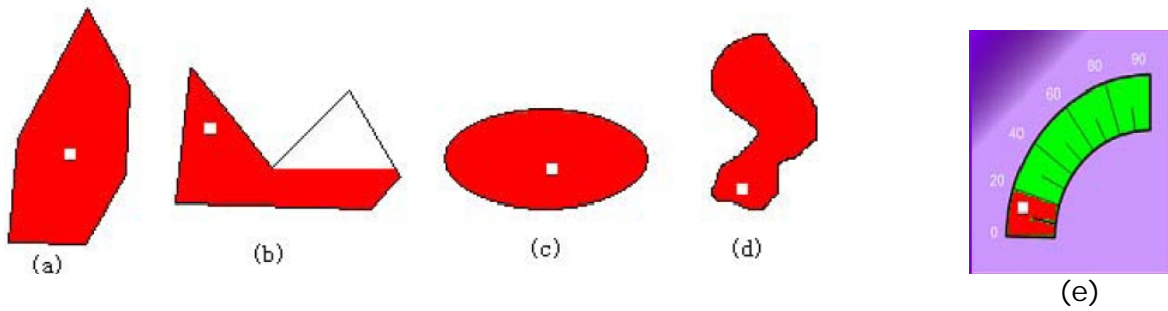
- <x> <y> Coordinate of the seed point
- <Color>: Color for area filling

Note:

★The initial color of the filling area shall be the same with that of seed point, otherwise, only the area having the same color with the seed point will be filled. (Area with the other color is handled as the border)

★It fits for filling the convex polygon area. For the concave polygon, some area can not be filled (as shown in figure b). However, you could set more seed points of different positions to realize full filling.

★This command will not change Palette's default setting.



The effect of area filling
(The white point is the seed point and it is filled actually)

e.g. AA 64 00 64 00 64 F8 00 CC 33 C3 3C

2.11 Clear Screen (0x52)

Tx: AA 52 CC 33 C3 3C

Rx: None

Fill full screen with background color (set by command 0x40)

2.12 Movement of the area selected (0x60, 0x61, 0x62, 0x63)

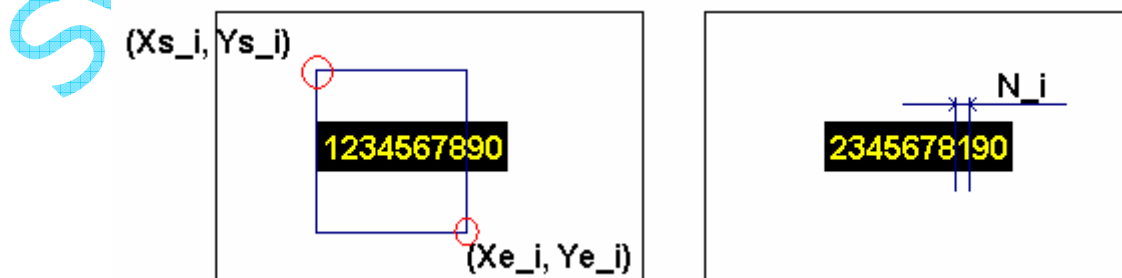
Tx: AA <CMD> (<Xs 0> <Ys 0> <Xe 0><Ye 0> <N 0>)(<Xs i> <Ys i> <Xe i><Ye i> <N i>)
CC 33 C3 3C

Rx: None

- <CMD >
 - 0x60 Right-circular movement of the area selected, i.e. to move the area from the far-left to far-right.
 - 0x61 Left-circular movement of the area selected, i.e. to move the area from the far-right area to far-left.
 - 0x62 Right movement of the area selected, i.e. to move from right to left, far-left area missing, using background color to fill far-right area (set by command 0x40)
 - 0x63 Left movement of the area selected, i.e. to move from left to right, far-right area missing, using background color to fill far-left area (set by command 0x40)
- <Xs_i> <Ys_i> <Xe_i><Ye_i>: Left top and bottom right coordinates of the area selected.
- <N_i> Dot matrix of area movement, ranging from 0x01 to 0x0F.

e.g. AA 60 00 40 00 40 00 80 00 80 08 CC 33 C3 3C

Move the selected area at left top coordinate(64,64) and bottom right coordinate(128,128) for 8 dot matrix from right to left as follows:



Note: this command can not be used in case of display with 90 degree rotation.

2.13 Image or icon display (0x70, 0x71, 0x7B, 0x9C, 0x9D, 0x99, 0xE2)

2.13.1 Image display (0x70)

Tx: AA 70 <Pic_ID> CC 33 C3 3C

Rx: None

- <Pic_ID> ID (set by command 0xE2) of the image saved in flash

Note: If the number of saved images exceeds 256, picture no shall be represented by 2 bytes.

2.13.2 Display one image and figure up (0x7B)

Tx: AA 7B <Pic_ID> CC 33 C3 3C

Rx: AA 7B <Checksum_H:L> CC 33 C3 3C

- <Pic_ID> ID (set by command 0xE2) of the image saved in the flash
- <Checksum_H:L> 2-byte sum of the content of the current image on the screen

Note: This command is used to verify the image downloaded to the flash to guarantee correct downloading.

2.13.3 Cut icon (0x71, 0x9C, 0x9D)

Tx: AA 71 <Pic_ID> <Xs> <Ys> <Xe> <Ye> <X> <Y> CC 33 C3 3C

Or AA 9C <Pic_ID> <Xs> <Ys> <Xe> <Ye> <X> <Y> CC 33 C3 3C

Or AA 9D <Pic_ID> <Xs> <Ys> <Xe> <Ye> <X> <Y> CC 33 C3 3C

Rx: None

- <Pic_ID> ID (set by command 0xE2) of icon saved in the flash
- <Xs> <Ys> Left top coordinate of cutting area of target image
- <Xe> <Ye> Right bottom coordinate of cutting area
- <X> <Y> Left top coordinate of cut icon on the current screen

Note: The difference between command 0x9C&0x9D and 0x71 is that, the background color of the image is not displayed when it is cut for command 0x9C&0x9D. Thus the effect of "transparent" can be realized.

Besides, the difference between command 0x9C and 0x9D is that, background color will be displayed before carrying out command 0x9C, but background color will not be displayed before carrying out command 0x9D.

e.g. AA 71 08 01 90 00 00 03 1F 01 90 00 C8 00 14 CC 33 C3 3C

Cut 8th image's area (400,0), (799,400) and display at (200,200) on the current screen as follows:



2.13.4 User-defined icon display (0x99)

Tx: AA 99 <X0,Y0,Icon_ID0>.....<Xn,Yn,Icon_IDn> CC 33 C3 3C

Rx: None

- <Xn,Yn> the display position of user icon
- <Icon_IDn> Index ID of user icon, 2 bytes

Note: User icon shall be pre-defined in 0x1D configuration file. Please refer to more details from 2.25.4.

2.13.5 Save the image on the current screen into the flash (0xE2)

Tx: AA E2 <Pic_ID> CC 33 C3 3C

Rx: AA E2 4F 4B CC 33 C3 3C

- <Pic_ID> ID of the image to be saved

2.14 Backlight Brightness Control (0x5E, 0x5F)

2.14.1 Turn off backlight (0x5E)

Tx: AA 5E CC 33 C3 3C

Rx: None

2.14.2 Set touchpad(keyed) backlight mode (0x5E)

Tx: AA 5E 55 AA 5A A5 <V_ON> <V_OFF> <ON_TIME> CC 33 C3 3C

Rx: None

- <V_ON> Brightness of the backlight when touch screen or keyboard is clicked, ranging 0x00-0x3F.
- <V_OFF> Brightness of the backlight when there is no clicking for some time, ranging 0x00-0x3F.
- <ON_TIME> Duration for backlight. The time ranges 0x00-0xFF and the unit is 0.5 second (127.5 seconds as maximum value).

Note: The first clicking after backlight is off will receive no response from the module. This command shall be validated by command 0XE0.

2.14.3 Set backlight to the maximum brightness (0x5F)

Tx: AA 5F CC 33 C3 3C

Rx: None

2.14.4 Adjust backlight brightness (0x5F)

Tx: AA 5F <PWM_T> CC 33 C3 3C

Rx: None

- <PWM_T> Backlight brightness value controlled by PWM, ranging from 0x00 to 0x3F; 0x00 means no backlight while 0x3F represents the maxim brightness.

Note: In case of display module of CCFL as backlight mode, backlight brightness cannot be adjusted by PWN. The backlight mode is either "on" or "off".

2.15 Touch panel operation (0x72, 0x73, 0x78, 0x79,0xE4)

2.15.1 Upload touch coordinate automatically(0x72, 0x73)

When touch panel is clicked, coordinate of touch position will be automatically uploaded as per the following format

Tx: None

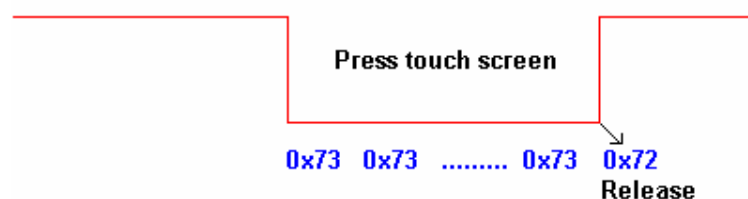
Rx: AA 73 <X> <Y> CC 33 C3 3C

Upload coordinate of touch position, one or more times (set by command 0xE0)

AA 72 <X> <Y> CC 33 C3 3C

Upload coordinate touch position when the user release the touch panel, only once (set by command 0xE0)

- <X> <Y> Coordinate of touch area, corresponding to screen resolution



Note: coordinates of touch position will not be uploaded as per Command 0x72 and 0x73 in case invalid touch position is clicked by the user. The invalid touch position is set by Command 0XE0.

2.15.2 Automatically upload touch code (0x78, 0x79)

For easier programming, users can set a touch code for some valid touch area (display with touch panel) or some key (display with no touch panel) and call it in the program. This touch code (hex) can be set by the user as per Configuration file 0x1E or using our software "SysDefEn". Then when the valid touch area is clicked or the key is pressed, the touch code, instead of coordinate of touch position, will be uploaded automatically. Under this circumstance, coordinate of the invalid touch position will not be uploaded.

Tx: None

Rx: AA 78 <Touch Code> CC 33 C3 3C

Touch code will be uploaded when the touch panel is clicked or the keyboard is pressed.

Tx: None

Rx: AA 79 <Touch Code> CC 33 C3 3C

Touch code will be uploaded when the user releases the touch panel.

2.15.3 Touch panel calibration (0xE4)

Tx: AA E4 55 AA 5A A5 CC 33 C3 3C

After sending this command, click three white spots displayed at different corners of the screen in turn

When the calibration is finished, the following command will be uploaded.

Rx: AA E4 4F 4B CC 33 C3 3C

2.16 Working mode configuration (0xE0) (Important)

Tx: AA E0 55 AA 5A A5 <TFT_ID> <Bode_Set> <Para1> CC 33 C3 3C

Rx: AA E0 4F 4B <TFT_ID> <Bode_Set> <Para1> CC 33 C3 3C

- <TFT_ID> 0x00, set before delivery.
- <Bode_Set> Default value is 0x07 (115200bps). In case of other values, please refer to the following table:

Bode_set	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Baud rate	1200	2400	4800	9600	19200	38400	57600	115200

- <Para1> Default value is 0x00 (corresponding 00000000 as per the following table)
To set touch panel and keyboard work mode as per the specific requirement, please refer to more details from the following table.

Para1	Bit description
.7	0= to upload command 0x72 when release touch panel 1= not to upload command 0x72 when release touch panel
.6	In case of touch position to be uploaded: 0= After the touch screen is clicked, command 0x73 will be uploaded every 100ms until the touch screen is released. 1= After the touch screen is clicked, command 0x73 will be uploaded once. In case key code to be uploaded and Para1.0=1. 0= After the touch screen is clicked, command 0x79 will be uploaded every 100ms until the touch screen is released. 1= After the touch screen is clicked, command 0x79 will be uploaded once.
.5	0= the display will not go to the next interface when touch screen is clicked or the keyboard is pressed. 1= the display will go to the next interface set as per configured file 0x1E when the touch screen is clicked or the keyboard is pressed.
.4	0= backlight will not be controlled by touch screen or keyboard. 1= backlight will be controlled by touch screen or keyboard; meanwhile, users can control it through command 0x5E/0x5F.
.3	0= buzzer rings when the touch screen is clicked or the keyboard is pressed. 1= buzzer does not ring when the touch screen is clicked or the keyboard is pressed.
.2	0=0°display(Image) 1= 90°display(Image)
.1	0= buzzer rings when any position of the touch panel is clicked, in case Para1.5=1. 1= buzzer only rings when valid position of the touch panel is clicked, in case Para1.5=1.
.0	0= command 0x79 will not uploaded, in case Para1.5=1. 1= command 0x79 will be uploaded, in case Para1.5=1.

Note: Configuration parameters are saved in the flash and those parameters will not be changed when the display module powers off. Therefore, parameters only need to be set once.

2.17 Execute commands periodically and circularly (0x9A)

The command is to save some commands (such as command 0x71) used periodically in the flash beforehand.

2.17.1 Turn on the function of executing command periodically and circularly

Tx: AA 9A Pack_ID CC 33 C3 3C

Rx: None

- < Pack_ID > ID of automatic and circular command group. It is defined by Configuration 0x1C. The configuration file consists of 16 command groups of 8KB. The length of each command group is always 8KB. At most 64 commands are included while each command occupies the space of 128KB.

Note: Please refer to 2.25.3 for more details about how to make configuration file 0x1C.

2.17.2 Turn off the function of executing command periodically and circularly.

Tx: AA 9A FF CC 33 C3 3C

Rx: None

2.18 Random access memory area operation (0xC0, 0xC1,0XC2)

Note: Drive module M100 doesn't support this command. Content in the random access memory will be changed by command 0x52 (clear screen).

2.18.1 Write in random access memory(0xC0)

Tx: AA C0 <Address> <Word0.....Wordn> CC 33 C3 3C

Rx: None

- <Address> Starting address (word address) in random access memory (RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <Word0.....Word n> Data to be written into RAM.

2.18.2 Read from random access memory (0xC2)

Tx: AA C2 <Address> <Data Length> CC 33 C3 3C

Rx: AA C2 <Data Pack> CC 33 C3 3C

- <Address> Starting address(word address) in random access memory(RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <Data _Length> Length of the data to be read out
- <Data _Pack> Data to be read out

2.18.3 Set point using data in random access memory (0xC101)

Tx: AA C1 01 <Address> <Pixel Number> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in random access memory(RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <Pixel_Number> Number of pixels. The maximum quantity is 13653 and every pixel takes up 3 words.

Note: Pixel data format is defined in random access memory as (X, Y, Color).

2.18.4 Connect lines using data in random access memory (0xC102)

Tx: AA C1 02 <Address> <Line Number> CC 33 C3 3C

Rx: None

- <Address> Starting address (word address) in random access memory; totally 40k word, ranging from 0x0000 to 0x9FFF.
- <Line_Number> Number of lines. The maximum quantity is 8191 and every line takes up 5 words.

Line data format is defined in random access memory as(Xs, Ys, Xe, Ye, Color).

2.18.5 Display broken line graph using data in random access memory (0xC103)

Tx: AA C1 03 <Address> <x> <y> <Line Number> <D_x> <Dis_x> <K_y> <Color> CC 33 C3 3C
Rx: None

- <Address> Starting address(word address) in random access memory (RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <x> X coordinate of the starting position
- <y> Zero position(the lowest point) in Y-axis; actual position equals to y-Ly
- <Line _ Number> Number of lines. The maximum quantity is 40960 and every line takes up 1 word.
- <D_X> Pixels' distance in random access memory, ranging from 0x01 to 0xFF, i.e. Address=Address+D_x
- <Dis_X> Increment in X-axis, ranging from 0x01 to 0x0F, i.e. x=x+Dis_x
- <K_y> Amplification factor of Y-axis, ranging from 0x00 to 0xFF; 1/16 is the unit, i.e. K_y=32 means amplify Y-axis twice.
- <Color> Color of lines. The property of palette will not be affected.

Note: The line data format in random access memory is defined as Ly (2 bytes), which is the height of pixel.

2.18.6 Display broken line graph at high speed using data in random access memory (0xC104)

Tx: AA C1 04 <Adr1> <x> <y> <Line Number> <D_x> <Dis_x> <Color1> <Addr0> <Color0> CC 33 C3 3C
Rx: None

- <x> Starting coordinate in X-axis
- <y> Zero position(the lowest point) in Y-axis; actual pixel position equals to y-Ly.
- <Line_Number> Number of lines. The maximum quantity is 40960 and every line takes up 1 word.
- <D_x> 0x01 as fixed value
- <Dis_X> Increment in X-axis, ranging from 0x01 to 0x0F; i.e. x=x+Dis_x.
- <Addr0> <Addr1> Starting address (word address) in random access memory (RAM), totally 40k word, ranging from 0x0000 to 0x9FFF.
 <Addr0> Starting address of the history curve to be erased
 <Addr1> Starting address of the curve to be displayed

Note: Supposing data points for connecting lines are 100 and there is only 1 curve, then absolute value difference between Addr1 and Addr0 is 100.

- <Color0> <Color1> Color of lines. The property of palette will not be affected.
 <Color0> Background color
 <Color1> Color of the curve to be displayed

Note: The line data format in random access memory is defined as Ly (2 bytes), which is the height of pixel.

This command is similar to command 0xC103 except:

A. the pixels' distance in read buffer is fixed as 1 for command 0XC104.

B. Flicker-free display is realized for command 0XC104 as the original pixel (set by Addr0 and Color0) is erased before connecting another pixel (set by Adr1 and Color1).

When baud rate 115200bps is applied, this command enable the maximum displaying speed

for broken line graph be 5500 pixel per second.

2.18.7 Scale broken line graph using data in random access memory (0xC105)

Tx: AA C1 05 <Address> <x> <y> <Line Number> <D_x> <Dis_x> <M_y> <D_y>
<Color> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in random access memory (RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <x> Starting coordinate in X-axis.
- <y> Zero position(the lowest point) in Y-axis; actual pixel position equals to $y - Ly$.
- <Line_Number> Number of lines. The maximum quantity is 40960 and every line takes up 1 word.
- <D_x> Pixels' distance in read buffer area, ranging from 0x01 to 0xFF; i.e. $Address = Address + D_x$ after connecting.
- <Dis_x> Increment in X-axis, ranging from 0x01 to 0x0F; i.e. $x = x + Dis_x$.
- <M_y> <D_y> Amplification factor of Y-axis, ranging from 0x00 to 0xFF; and the displaying height equals to $Y \times M_y / D_y$, e.g. $M_y = 4$, $D_y = 2$ means amplifying Y-axis twice.
- <Color> Color of lines. The property of palette will not be affected.

Note: The line data format in random access memory is defined as Ly , 2 bytes, which is the height of pixel.

2.18.8 Scale window limit two-way broken line graph using data in random access memory (0xC106)

Tx: AA C1 06 <Address> <x> <y> <Line Number> <D_x> <Dis_x>
<M_y> <D_y> <Color> <Ymin> <Ymax> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in random access memory (RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <x> X coordinates of the starting position
- <y> Zero position(the lowest point) in Y-axis; actual pixel position equals to $y - Ly$ or $y + Ly$.
- <Line_Number> Number of connection lines, ranging 0x0000-0x9FFF. Every line is one word and there are maximum 40960 lines.
- <D_x> Pixel distance in the read buffer area, ranging 0x01—0xFF, i.e. $Address = Address + D_x$;
- <Dis_x> Increment of X coordinate, ranging 0x01-0x0F, i.e. $x = x + Dis_x$;
- <M_y> <D_y> Y coordinate amplification factor, ranging 0x00-0xFF, the displaying height equals to $Y \times M_y / D_y$, e.g. $M_y = 4$, $D_y = 2$ means amplifying Y-axis twice.
- <Color> Color of lines. The property of palette will not be affected.
- <Ymin> Lower limit coordinate of Y-axis
- <Ymax> Upper limit coordinate of Y-axis

Note: The line data format in random access memory area is defined as Ly , 2 bytes, description as follow:

Ly.15 Connecting direction, 0=positive, 1=negative;

e.g. $Ly = 0x8010$ means negative height 16;

Ly.14-Ly.0 Connecting height.

2.18.9 Take random access memory as buffer area for setting point (0XC107)

★ Clear buffer area for setting point

Tx: AA C1 07 <Address><X_Len><Y_Len> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in setting point buffer area (RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <X_Len> Width (dot matrix) in X-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.
- <Y_Len> Width (dot matrix) in Y-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.

*Note: The address of setting point buffer in RAM is $(X*Y)/16$. For example, if the user creates a 64*64 setting point buffer area at 0x0000 of the random access memory area, then the address of setting point buffer area in the random access memory area is 0x0000-0x00FF.*

★ Set point in buffer area for setting point

Tx: AA C1 07 01 <Address><X_Len><Y_Len><Xs, Ys><Color><Mode><(X0,Y0)...(Xi,Yi)>
CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in setting point buffer area; totally 40k word, ranging from 0x0000 to 0x9FFF.
- <X_Len> Width (dot matrix) in X-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.
- <Y_Len> Width (dot matrix) in Y-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.
- <Xs, Ys> Coordinate of corresponding screen window of setting point buffer area,
- <Color> Color of the point to be set or erased. The property of palette will not be affected.
- <(X0,Y0)...(Xi,Yi)> Coordinate of setting points. The actual displaying coordinate is $X=Xs+Xi$ $Y=Ys+Yi$;
- <Mode> mode of setting point
0x00 Delete points only in the setting point buffer area
0x10 Delete points both in the setting point buffer area and the current screen;
0x01 Set points only in the setting point buffer area
0x11 Set points both in the setting buffer area and the current screen.

★ Recover buffer area for setting point area to the current screen

Tx: AA C1 07 02 <Address><X_Len><Y_Len><Xs,Ys><Color> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in setting point buffer area; totally 40k word, ranging from 0x0000 to 0x9FFF.
- <X_Len> Width (dot matrix) in X-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.
- <Y_Len> Width (dot matrix) in Y-axis of corresponding screen window of setting point buffer area, ranging from 0x0000 to 0xFFFF.
- <Xs, Ys> Coordinate of corresponding screen window of setting point buffer area,
- <Color> Color of the point to be set or erased. The property of palette will not be affected.

Note: This command is for flexible and convenient operation of icons of pure color, such as movement of cross cursor.

2.18.10 Display multi-parameters in random access memory (0xC108)

Tx: AA C1 08 <Address><Parameter_N> CC 33 C3 3C

Rx: None

- <Address> Starting address(word address) in random access memory; totally 40k word, ranging from 0x0000 to 0x9FFF.
- < Parameter_N > Quantity of the parameters to be displayed, ranging from 0x01 to 0xFF.

Parameter data format in the random access memory is as follows:

Relative address in the random access memory	Definition	Description
0x00	Mode	Mode.15:.12 Maximum digit of integer to be displayed; Mode.11:.8 Digit of decimal, 0 means no decimal digit; Mode.7 0= Zero is not to be displayed in case of invalid integer digit. 1= Zero is to be displayed in case of invalid integer digit. For example Mode.15:.12=3, Mode.7=1, 0.3 will be displayed as 000.3. Mode.6-Mode.4 No definition, 0 as the default value. Mode.3-Mode.0 Size of the character to be displayed (Based on Lib_ID=0, default ASCII library) 0=8*8 1=6*12 2=6*12 3=8*16 4=12*24 5=16*32 6=20*40 7=24*48 8=28*56 9=32*64 0x0A-0x0F No definition
0x01	X	Starting address of X-axis of the parameter to be displayed
0x02	Y	Starting address of Y-axis of the parameter to be displayed
0x03	F_Color	Foreground color of the parameter to be displayed. Palette will not be affected.
0x04	B_Color	Background color of the parameter to be displayed. Palette will not be affected.
0x05	Parameter	Parameter data, 4-byte signed integer. Parameter.31 is the sign digit.

Parameter setting example

Data	Parameter	Mode
		The character is 16*32, Mode 6.3, Zero is not to be displayed in case of invalid integer digit.
12345	0x00 00 30 39	0x6005
123.450	0x00 01 E2 3A	0x6305
-123.450	0xFF F3 1D C6	0x6305

e.g. AA C0 00 00 63 05 00 0A 00 0A F8 00 00 1F 00 01 E3 3A CC 33 C3 3C AA C1 08 00 00

01 CC 33 C3 3C
 12.450 will be display at (10,10).

2.18.11 Buffer command to realize displaying synchronously using random access memory (0xC110)

Tx: AA C1 10 <Address> <Frame Number> CC 33 C3 3C
Rx: None

- <Address> Starting address(word address)in random access memory(RAM); totally 40k word, ranging from 0x0000 to 0x9FFF.
- <Frame _Number> Number of command frame to be displayed continuously, ranging 0x01-0xFF;

Definition of command frame in random access memory is as follow:

Starting Address	Definition	Frame Structure
<Address>	The first command frame	frame length +command +data e.g. display the image 0,definition is 02 70 00
<Address>+0x80	The second command frame	
.....		
<Address>+0x80×(K-1)	The Kth command frame	

Note: As data is saved by word in random access memory, the address in which one command frame occupies is 0x80. But and actual memory space is 256 bytes.

This command is mainly used to avoid the phenomenon that the updated parameters are not displayed synchronously due to delay in communication, especially in case of low baudrate.

2.19 Keyboard Operation(0x71, 0xE5)

2.19.1 Upload key code (0x71)

Tx: None
Rx: AA 71 <K_code> CC 33 C3 3C

- <K_code>: Key code pre-set by the user. Relevant key code will be uploaded automatically once relevant keys are pressed. The speed is 5 keys per second.

2.19.2 Set key code (0xE5)

Tx: AA E5 55 AA 5A A5 <K_Code0.....K_Code63> CC 33 C3 3C
Rx: None

- <K_Code0.....K_Code63> Key codes to be set. The number is fixed at 64. Only 16 key codes are valid for 4x4 keyboard interface.

2.20 Read and write in user's memory (0x90, 0x91)

Note: The physical media of database is NAND Flash with 100 thousand erasable times and 10 years life circle.

2.20.1 Write in random data memory (0x90 64KB)

Tx: AA 90 55 AA 5A A5 01 DE <Address> <Data0.....Data1> CC 33 C3 3C

Rx: AA 90 4F 4B CC 33 C3 3C

- <Address> Starting address of the data to be written into data memory, 2 bytes, ranging from 0x0000 to 0xFFFF.
- <Data0.....Data1> Data to be written into the data memory.

2.20.2 Write in sequence data memory (0x90 30MB)

Tx: AA 90 55 AA 5A A5 <Address> <Data0.....Data1> CC 33 C3 3C

Rx: AA 90 4F 4B CC 33 C3 3C

- <Address> Starting address of the data to be written into data memory, 4 bytes, ranging from 0x00000000 to 0x01DDFFFF.
- <Data0.....Data1> Data to be written into the data memory.

Note: Different from random data memory, sequence data memory can only write in sequence. Writing randomly is not possible.

*The whole sequence memory is divided into 239 128KB data pages. When it comes to page heading (address=*****0 00 00), the current page will be erased. There is no data back-up before erasing, which will not affect the data on the other pages. It fits for continuous and big data storage, such as non-paper record and audio recording.*

2.20.3 Read from data memory (0x91)

Tx: AA 91 <Address> <Length> CC 33 C3 3C

Rx: AA 91 <Address> <Length> <Data0.....Data1> CC 33 C3 3C

- <Address> Starting address of the data to be read out in data memory, 4 bytes, ranging from 0x00000000 to 0x01DDFFFF; RAM address ranges 0x01DE0000-0xDEFFFF
- <Length> Length of data to be read out, 2 bytes. Maximum 64KB can be read out once.
- <Data0.....Data 1> Data to be read out.

2.21 Download font file and configuration Files (0xF2)

Tx: AA F2 F2 F2 5A A5 <Lib ID> CC 33 C3 3C

After receiving "Please Tx Text Lib !", the users can download corresponding font library. When font library is saving successfully, the display module will send "*****One Text Lib Saved OK !*****"

2.22 Simple Algorithm Support (0xB0)

2.22.1 MAC calculate (0xB002)

Tx: AA B0 02 <A,B,C,D> CC 33 C3 3C

Rx: AA B0 02 <E,F> CC 33 C3 3C

- <A,B,C,D> Calculate $(A \times B + C) / D$ while A,B,C and D are all unsigned integer of 2 bytes
- <E,F> Result. while E is 4-byte quotient and F is 2-byte remainder

2.22.2 Array sorting (0xB003)

Tx: AA B0 03 <Pack0> CC 33 C3 3C

Rx: AA B0 03 <Pack1> CC 33 C3 3C

- <Pack0> Array to be sorted, 2 bytes
- <Pack1> Array after sorting, 2 bytes, ascending sort.

2.23 Buzzer control (0x79)

Tx: AA 79 <On Time>CC 33 C3 3C

Rx: None

- <On _Time> 0x01-0xFF, duration of the buzzer's ringing, 10ms as the unit.

2.24 Date and time display (0x9B, 0xE7)

3.24.1 Turn off date and time display (0x9B00)

Tx: AA 9B 00 CC 33 C3 3C

Rx: None

3.24.2 Turn on date and time display (0x9BFF)

Tx: AA 9B FF <RTC Mode> <Text Mode> <Color> <X> <Y> CC 33 C3 3C

Rx: None

- <RTC_Mode>: Date and time display mode
 - 0x00 HH:MM:SS
 - 0x01 20YY-MM-DD HH:MM:SS
- <Text_Mode>: Date and time display character setting

0x00 8*8	0x01 6*12	0x02 8*16
0x03 12*24	0x04 16*32	0x05 20*40
0x06 24*48	0x07 28*56	
- <Color> Character's color
- <X> <Y> Coordinate of date and time to be displayed

2.24.3 Adjust date and time (0xE7)

Tx: AA E7 55 AA 5A A5 <YY:MM:DD:HH:MM:SS> CC 33 C3 3C

Rx: None

- <YY:MM:DD:HH:MM:SS> Date and time format to be set as year : month : day : hour : minute : second, as per BCD code.

➤

e.g. AA E7 55 AA 5A A5 08 11 28 12 57 00 CC 33 C3 3C

Set current time as 12:57:00, Nov. 28th, 2008.

2.24.4 Read current date and time (the Gregorian calendar) (0x9B5A)

Tx: AA 9B 5A CC 33 C3 3C

Rx: AA 9B 5A <YY: MM: DD: WW: HH: MM: SS> CC 33 C3 3C

- <YY:MM:DD:WW:HH:MM:SS> Current date and time data
08 12 25 04 09 58 00 means 09:58:00, Thursday, Dec. 25th, 2008.

Note: Not all models support Clock display.

2.25 Configuration file

Note: The display module can realize simplified Operating System function through following configuration files, which can reduce considerable codes.

2.25.1 Touchpad interface transfer (Configuration file 0x1E)

Development process:

Step One: Design a user interface with the same physical resolution with display module and download it to display module.

For example, set interface's resolution as 800x600 in case of DMT80600T080_02WT.

Step Two: Create configuration file

Configuration file 0x1E is a binary file with maximum 8192 touchpad commands. Every command takes up 16 bytes, defined as follows:

Starting address	Length (Byte)	Definition	Description
0x00	2	Pic_Now	ID of the picture displayed on the current screen if high byte of Pic_Now is 0xFF, it means touchpad command ends.
0x02	4	x_s, y_s	Left top coordinate of valid touch area
0x06	4	x_e, y_e	Right bottom coordinate of valid touch area
0x0A	2	Pic_Next	ID of the picture to be displayed next. if high byte of Pic_Next is 0xFF, it means no more interface transfer.
0x0C	2	Pic_Cut	ID of the picture with the animation effect when certain touch area is clicked. if high byte of Pic_Cut is 0xFF, it means no animation effect when certain touch area is clicked.
0x0E	2	Touch_Code	Touch code to be uploaded when valid touch area is clicked (act as a message to trigger users' software) if high byte of key code is 0xFF, it means no touch code will be uploaded. if high byte of key code is 0xF0-0xF3, it means the uploaded data will be indexed to 0x1A configuration file. Every index length is fixed to 256 bytes with the first one as the valid length.


Note: The process of creating configuration files is the process in which users arrange interface transfer, which can be finished even by a graphic designer. The configuration files can be compiled by "UltraEdit" or some kinds of compile system, such as C51 and ASM51.

The following example will show how to compile a configuration file 0x1E with the classic ASM51 compiler in DOS system, with which most engineers are familiar.





Sample module DMT80600T080_02WT

Requirement: 1. When the touch area  of 1st picture is clicked, the display module will go to the 2nd picture and upload touch code (0x0031, set as you wish)

2. When the touch area  of 1st picture is clicked, it will be



as per 3rd picture;

3. (407,497) and (510,574) is the valid touch area.

Example command of configuration file 0x1E

```

ORG 0000H
DW 01,407,497,510,574,02,03,1 // One touchpad command
DW OFFFHH //All commands definition finished.
END
    
```

The above example only compiles one touchpad key and one interface transfer. More interface transfers need users' more commands.

Compile finished configuration file *.ASM to *.HEX file using compiler ASM51.exe; then convert *.HEX file to *.BIN file which can be recognized by the display module.

Step 3: Download the configuration file into the display module.

Use command 0xF2 to download the finished binary configuration file save it at 0x1E of the display. Downloading configuration file can be also done through our software "Terminal Assistant".

Step 4: Set the working mode of display module as automatic interface transfer

Set the 5th bit of Para1(Para1.5,0x20) as 1, using command 0xE0. When users click the valid touch area, the display module will go to the next interface as per setting and upload user pre-defined touch code instead of uploading coordinates. This step can also be finished by our software "Terminal Assistant".

Step 5: Test whether interface transfer works well. If not, repeat step 2 and 3.

2.25.2 keyed interface transfer (configuration file 0x1B)

The configuration 0x1B consists of maximum 5957 keyed commands and every command is 22 byte.

Starting address	Length (Byte)	Definition	Description
0x00	2	Pic_Now	ID of picture that is displaying on the current screen. If high byte of Pic_Now is 0xFF, it means the end of keyed command.
0x02	2	0x00:Key_Code	Key code
0x04	2	Pic_Next	ID of picture that will be displayed next. If high byte of Pic_Next is 0xFF, it means no picture transfer. If high byte of Pic_Next is 0xFE, it means area picture transfer.
0x06	14	Pic_Cut,Xs,Ys,Xe,Ye,X,Y	In case high byte of Pic_Next is 0xFE, the relevant definition is as follows: Pic_Cut: ID of the picture that will be cut. Xs,Ys,: left top coordinate of the area picture that will be cut. Xe,Ye,: right bottom coordinate of the area picture that will be cut. No definition if high byte of Pic_next=others.
0x14	2	Touch_Code	Keycode uploaded after some button is pressed. If high byte of Touch_Code is 0xFF, it means that no code will be uploaded.

Example command of configuration file 0x1B

```
ORG 0000H
```

```
DW 0 // Current Picture No
```

```
DW 0 // Keycode
DW FF // No interface change
DW 0,0,0,0,0,0,0 // No definition
DW 3 // Return key value
END
```

2.25.3 Automatic and circular executing command group (configuration file 0x1C)

The configuration file consists of 16 command groups of 8KB. The length of each command group is always 8KB, at most 64 commands included while each command occupies the space of 128KB.

The definition of each command is as follow:

Starting Address	Definition	Description
0x00	Command_Delay	The time for command delay. The unit is 8ms. 0x00 means no delay
0x01	Command_Length	The length of this command 0x00: this command invalid Others: the length of this command (starting from 0x02)
0x02-0x7F	command	Standard command without frame header 0xAA and frame end 0xCC 33 C3 3C.

Example commands of configuration file 0x1C

```
ORG 0000H
DB 125,2,70H,00H // Every command occupies 128 byte.

ORG 0080H
DB 125,2,70H,01H //The first picute will be shown when the time delay is 1 second.

ORG 0100H
DB 125,2,70H,02H

ORG 0180H
DB 125,2,70H,03H

ORG 0200H
DB 125,2,70H,04H

END
```

Note: If the length of each command is less than 128 byte, 00 will be used to make the command complete.

125: time delay

*8ms*125=1s. It can be set by the user according to different requirements.*

02: the following command is 2 byte

70 01: the command AA 70 01 CC 33 C3 3C (The display will show PIC_ID=1)

*70 02 the command AA 70 02 CC 33 C3 3C (The display will show PIC_ID=2)
And so on.*

2.25.4 Icon display (configuration file 0x1D)

Configuration file of icon library is a binary file consisting of maximum 13,107 icon definition. ICON_ID ranges from 0000H to 3332H. It starts from 0. For example:

DW 01,00,00,100,100 ICON_ID address is 0000H.

DW 02,00,00,100,100 ICON_ID address is 0001H.

Icon definition is as follows:

Starting address	length (Byte)	Definition	Description
0x00	2	Pic_ID	Picture ID of the saved icon
0x02	4	x _s ,y _s	The left top coordinate of icon area
0x06	4	x _e ,y _e	The right bottom coordinate of icon area

Example command of configuration file 0x1D

```
ORG 0000H
DW PIC_NUMBER, XS, YS, XE, YE // PIC_NUMBER: Pic No in the flash.
// XS, YS, XE, YE: Coordinate of icon area.
DW OFFFH
END
```

*Note: When receiving command 0x99, the display module shall carry out it as follows,
A. Cut Pic_ID (Xs,Ys) (Xe,Ye) from position Icon_ID×10 in configured file 0x1D
B. Display the cut icon at (X,Y) on the current screen.
C. Display the next icon.*

2.26 Upgrade full-screen display once (0XD0)

Tx: AA D0 CC 33 C3 3C

Rx: None

Note: This command aims to clear data in RAM.