DWIN UART LCM

COMMAND SET

Version 2.4

Index

1 Serial Port Specifications

1.1 Serial Port Work Mode

All UART LCM standard products of DWIN technology are asynchronous, full-duplex serial port (UART) mode. Serial port mode is 8n1; i.e. each data transmission has 10bits: one initial bit, eight data bits (Least Significant Bit, LSB) and one stop bit.

- If the terminal I/O 0 pin is high level or floating state when power is on, the baud rate of serial port is preset by users from 1200 to 115200 bps. The specific setting mode refers to the command 0xE0.
- If the terminal I/O 0 pin is low level when power is on, the baud rate of serial port is fixed at 921600 bps.

Note: There are two different kinds of setting modes about DWIN UART LCM I/O0 pin of HDL662:
Set jumper pins: If it is short circuit, I/O0 is low level; if it is open circuit, I/O is high level;
Directly control by the USB port: If DWIN UART LCM connects to the USB port, I/O0 is low level; if DWIN UART LCM does not connect to the USB port, I/O0 is high level.

1.2 Data Frame Structure

DWIN UART LCM serial port's data frame structure contains four data blocks, as described in the table below:

Data Block	1	2	3	4	5
Example	0xAA	0x70	0x01	Check_H:L	0xCC 0x33 0xC3 0x3C
Description	Header, fixed as 0xAA	Command	Data, maximum 248 bytes	2-byte checksum verification (Optional, H600 and K600+ only)	Frame end (Tail Frame)

When PARA2.1=0 (it can be set by the 0xE0/E3 command set or SD card), 2-byte checksum verification can be used (H600 and K600+ kernel only). The checksum is calculated from the command set to the end of the last data according the bytes accumulation. For example, the first picture is displayed using the K600+ kernel:

PARA2.1=1 AA 70 01 CC 33 C3 3C Don't start the frame checksum verification.
PARA2.1=1 AA 70 01 00 71 CC 33 C3 3C It start the frame checksum verification.

1.3 Communication Frame Buffer (FIFO)

DWIN UART LCM has a 24-frame communication buffer which is FIFO structure (First In First Out memory). As long as the communication buffer is not overflow, the user can continuously send data to the UART LCM.

DWIN UART LCM has a hardware pin ("BUSY" pin of the user's interface) whcih indicates the status of the FIFO buffer. Normally, the "BUSY" pin is high level (RS232 is the negative level). When the FIFO buffer only remains one frame buffer, the "BUSY" pin will immediately altered to low level (RS232 is the positive level).

The processing speed of DWIN UART LCM is very quick in general applications. So users do not need to judge the status of the BUSY signal (Such as DWIN picture software does not judge the status of the BUSY signal) .

Users need to send multiple data frames in some short time applications, such as refreshing hundreds of screen parameters. Recommend that customers use the BUSY signal to control the serial port to send data. When the Busy signal is at low level, it does not send data to the UART LCM.

The buffer may be overflow, if it appears "dropped frames" phenomenon (Some data is not shown) when the process of using UART LCM. Check the BUSY signal whether jump or not using an oscilloscope. If jump, users need to slow down the speed of sending or to increase the BUSY signal

hardware detection to judge the BUSY processing.

1.4 Byte Transmission Order

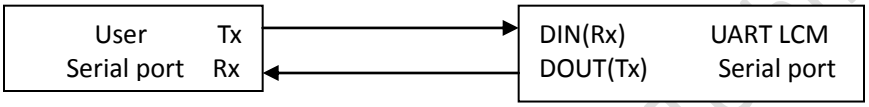
All commands or data of DWIN UART LCM are hexadecimal format (HEX). For word (2byte) data, high byte (Most Significant Bit, MSB) always first transmitted.

For example, X coordinate is 100 whose Hex mode format is 0x0064. The transmission order to the UART LCM is 0x00 0x64.

1.5 Data Transmission Direction

In DWIN UART LCM, the transmission direction is in accordance with the rules of the following definition:

- Downlink (TX): The user transmits data to UART LCM. Data is inputted from the "DIN" pin in UART LCM user interface;
- Uplink (RX): The user receives data from UART LCM. Data is outputted from the "DOUT" pin in UART LCM user interface.



2 Commands Table

CATEGORY	COMMAND	DESCRIPTION
Handshake	0x00	Check the configuration and version information
Display Parameter Configuration	0x40	Set palette
	0x41	Set characters spacing
	0x42	select color to the background color palette
	0x43	select color to the Foreground color palette
	0x44	Set cursor display mode
Text Display	0x53	8×8 dot matrix ASCII characters
	0x54	16×16 dot matrix GBK extended string to display
	0x55	32×32 dot matrix GB2312 internal code string display
	0x6E	12×12 dot matrix GBK extended code string display
	0x6F	24×24 dot matrix GB2312 internal code string display
	0x98	Any dot matrix, any code string display
	0x45	Open/Close the text frame limitation
Put Pixel	0x50	Put multiple pixels with background color (Delete pixels)
	0x51	Put multiple pixels with foreground color
	0x74	Dynamic curve fast puts pixels
	0x72	Direct memory operation
Line &Polygon	0x56	Connect specific pixels with the Foreground color line(Display polygon)
	0x5D	Connect specific pixels with the background color line (Delete polygon)
	0x75	Fast Draw continuous vertical line (Spectrum)
	0x76	Fast Draw line chart
	0x78	Connect offset
Arc &Circle Area	0x57	Reverse/Display multiple arc or circle area
Rectangle	0x59	Draw multiple rectangular with foreground color (Display rectangle)
	0x69	Draw multiple rectangular with background color (Delete rectangle)
Areas Operation	0x73	Fill special area with double color bitmap
	0x64	Fill special area
	0x52	Clear screen
	0x5A	Clear many special areas
	0x5B	Fill many specific areas
	0x5C	Reverse many specific areas color
	0x60	Many specific areas loop and shift left
	0x61	Many specific areas loop and shift right
	0x62	Many specific areas shift left
	0x63	Many specific areas shift right
Draw Picture/Icon	0x70	Draw a full screen picture
	0x7B	Draw a full screen picture and calculate CRC checksum
	0x71	Cut icon from the specific picture and paste to the current page
	0x9C	Cut icon from the specific picture to filter out background and

		paste to the current page. Automatically restore the original background before paste
	0x9D	Cut icon from the specific picture to filter out the background and paste to the current page.
	0x9E	Cut icon from the specific picture and paste to the current page after rotate specific angles. You can select to filter out background. 0x9E, H600 only
	0xE2	Save the current picture to the terminal
	0x97	Draw ICON (The maximum size is 255*255). K600+ only
	0x99	Users define icon display
	0xD0	Refresh a full screen compulsively once. K600, K600+ and H600 only
Animation Support	0x9A	Open/Close automatic execution command set that users presets
Temporary buffer operation (1280*800 and 1365*768 don't supported by DWIN)	0xC0	Write data to temporary buffer
	0xC101	Display data points of temporary buffer
	0xC102	Display data lines of temporary buffer
	0xC103	Display line chart with date point in temporary buffer (Zoom dynamic curve)
	0xC104	High-speed no flashing with date point in temporary buffer (Oscilloscopes)
	0xC105	Zoom and display line chart with data in temporary buffer
	0xC106	Zoom and display window to limit the two-way line chart with data in temporary buffer
	0xC107	Empty the pixel buffer
	0xC107	Put pixel in the pixel buffer
	0xC107	Restore the pixel buffer to the current page
	0xC108	Display parameters with temporary buffer
	0xC110	Simultaneous display by the command of temporary buffer
	0xC2	Read data back from temporary buffer
Database Operation	0xF2	Download font (Download ICON picture library must use SD card)
	0x90	Write date to users database
	0x91	Read date from users database
Keyboard Operation	0x71	Upload key code
	0xE5	Configure the key code interface
Touch Screen Operation	0x72	After releasing touch screen, upload data the last time (Set to turn off with the 0xE0 command)
	0x73	After press touch screen, upload data (Set to upload only once with the 0xE0 command)
	0xE4	Touch screen calibration
	0x78	In the touching function interface auto-switch mode, when touch screen is released, the default key code is automatically uploaded
	0x79	In the touch interface auto-switch mode, when touch screen is pressed, the default key code is automatically uploaded
	0x7C01	Touch screen input ASCII strings. K600+, H600 only
	0x7C02	Touch screen input mixed string in both Chinese and English. K600+ and H600 only
Buzzer Control	0x79	Buzzer screams one time

Video Switching	0x7A	Switch video and UART LCM
Backlight Operation	0x5E	Close backlighting or set the touching function(Key control)backlight model
	0x5F	Open backlighting or PWM model to adjust the backlight brightness
Clock Operation	0x9B	Open/Close clock automatically overlay and display; read the current clock
	0xE7	Set clock
Parameter Configuration	0xE0	Configure the speed of user serial port, touch screen data upload format, backlight control mode and display mode;H600/K600+ kernel does not save when power-down
	0xE3	Similarly 0xE0 command functions, but the power to save; H600 and K600+ only
Practical Algorithm	0xB001	Pinyin input method based in the first level of font
	0xB002	Calculate(A×B+C)/D,E is 4 bytes quotient，F is two bytes remainder
	0xB003	Unsigned integer(2 bytes) array sort
	0xB004	GBK font-based Pinyin input query
Sound Operation	0x30	Play music that specific storage location
	0x32	Real-time volume adjustment
	0x33	Immediately stop playing
	0x3F	Sound operation response
Software Upgrade	DWIN_M600_BOOT!	Kernels software are upgraded online by serial port

The sequence of kernel development keeps downward compatible with command set. The development order is : M100、M600、K600、H600、K600+ kernel.

- M100 kernel and M600 kernel command set are completely same;
- Kernels above K600 have different drivers. So we increase the 0xD0 command on the basis of M100、M600； K600, M100 and M600 kernels command set are completely same;
- Under the premise of H600 compatible with K600, we add the 0x7C、0x9E and 0xE3 commands. The part of serial port screens support SD card to download;
- Under the premise of K600+ compatible with H600, we add the 0x97 command. Download is completely supported with SD card in corresponding serial port screens;

3 Command Set Description

3.1 Handshaking (0x00)

Tx: AA 00 CC 33 C3 3C

Rx: AA 00 'OK_V*.*' *P1 P2 P3* Pic_ID CC 33 C3 3C

Or

Tx: AA 00 00 CC 33 C3 3C

Rx: AA 00 'OK_V*.*' *P1 P2 P3 P 4 Pic_ID* CC 33 C3 3C

- OK_V*.*, *.* is the current software version of UART LCM;
- P1 is the screen configuration mode of the current UART LCM (Please refer to the 0xE0 command about specific parameters);
- P2 is the serial port baud rate of the current setting;
- P3 is the mode 1 of the touch screen, buzzer and display configuration;
- P4 is the mode 2 of the display configuration;
- Pic_ID is the ID of the currently display picture;

The initialization of DWIN UART LCM power-on needs about 0.5-2 seconds (Depending on the power capacity of users and the speed of power-on). It does not respond to user commands before the initialization of power-on is completed. Users can send the handshake command to confirm whether the UART LCM has been completed the initialization of power-on or not.

3.2 Set the Current Palette (0x40)

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

Rx: None

- <FC>: The foreground color palette, 2 bytes (16 bit, 65k color). The default value is 0xFFFF (White) when reset.
- <BC>: The background color palette, 2 bytes (16 bit, 65k color). The default value is 0x001F (Blue) when reset.
- 16bit palette is defined 5R6G5B mode, the following table as described:

Bit definition of 16bit palette																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Define	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
	red 0xF800					green 0x07E0						blue 0x001F				

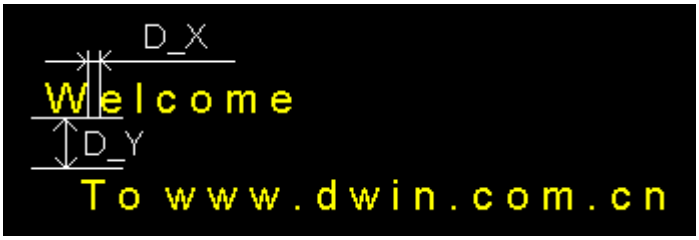
All values will be maintained unless users reset values. The default values will be restored if the user resets UART LCM hardware.

3.3 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) from 0x00 to 0x7F. Default reset value is 0x00.
- <D_Y>: Character spacing in Y direction (line spacing) from 0x00 to 0x7F. Default reset value is 0x00.



All values will be maintained unless users reset values. The default values will be restored if the user resets UART LCM hardware.

3.4 Select the Color of Specific Position (0x42, 0x43)

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

Rx: None

- <CMD>: Select the background color palette with the color of specific position using the 0x42 command. Select the foreground color palette with the specific position using the 0x43 command.
- <X> <Y>: The specific coordinates. (The coordinates of DWIN UART LCM are 2 bytes.)

Example 1:

AA 42 00 10 01 00 CC 33 C3 3C

Select color on x=16 (0x0010) y=256 (0x0100) to the background color palette.

Example 2:

AA 43 00 80 00 60 CC 33 C3 3C

Select color on x=128 (0x0080) y=96 (0x0060) to the foreground color palette.

3.5 Set Cursor Mode (0x44)

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

Rx: None

- <Cursor_EN>

  0x01: Set cursor OFF. The cursor will be displayed on (X, Y);

  0x00: Set cursor ON;
- <X>, <Y> are the left-up coordinates of the cursor;
- <Cursor_Width>: Cursor Wide (0x01-0x1F);
- <Cursor_Height>: Cursor High (0x01-0x1F);

If (Cursor_EN=0x00), other parameters do not make sense in this command.

Example:

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

Set cursor on (128, 96), the Cursor Width=16, the Cursor High=3

3.6 Text Display Command (0x53, 0x54, 0x55, 0x6E, 0x6F, 0x98, 0x45)

3.6.1 Display standard font (0x53, 0x54, 0x55, 0x6E, 0x6F)

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

Rx: None

- <CMD>

  0x53: Display 8*8 dot matrix ASCII string;

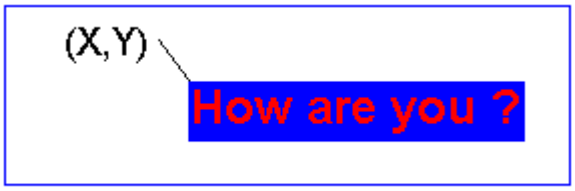
  0x54: Display 16*16 dot matrix extended Chinese Character string (ASCII characters are displayed with DBC case 8*16 dot matrix);

  0x55: Display 32*32 dot matrix internal Chinese Character string (ASCII characters are displayed with DBC case 16*32 dot matrix);

0x6E: Display 12*12 dot matrix extended Chinese Character string (ASCII characters are displayed with DBC case 6*12 dot matrix);

0x6F: Display 24*24 dot matrix internal Chinese Character string (ASCII characters are displayed with DBC case 12*24 dot matrix);

- <X> <Y> : The starting coordinates of string (The left-up coordinates of the first character);
- <String>: The print string. Chinese Characters are encoded by GB2312 (0x55, 0x6F; internal code) or GBK (0x54, 0x6E, extended internal code). Set the display color using the 0x40 command. Set the display character spacing using the 0x41 command, and line wrap at the end. The 0x0D and 0x0A commands are used to indicate the "return" and "line wrap". (The 0x0D and 0x0A commands represent/mean “return” and ”line wrap”)



Example:

AA 55 00 80 00 30 48 6F 77 20 61 72 65 20 79 6F 75 20 3F CC 33 C3 3C

Print 32*32 string "How are you?" on (128, 48).

3.6.2 Display Custom Font Library (0x98)

Tx: AA 98 <X> <Y> <Lib_ID> <C_Mode> <C_dots> <Fcolor> <Bcolor> <String> CC 33 C3 3C

Rx: None

- <X> <Y>: The left-up coordinates of the first character.
- <Lib_ID>: The font library ID, 0x00-0x3B, corresponding to the font position with the command 0xF2.

The memory size of the UART LCM kernel is 32MB. The memory is arranged as 60 areas. The definition of Lib_ID is listed as follows:

Lib_ID	Capacity	Description	Factory default value
0x00-0x1F	128KB	There are 32 small fonts whose size is 128KB. Generally, the user can design different style for special icon or ASCII characters.	0x00=ASCII character library, please don’t modify it. 0x01=Pinyin input method thesaurus 0x02-0x19: Free 0x1A-0x1F: Touching function configuration files
0x20-0x3B	1MB	There are 28 large fonts whose size is 1 MB. ● Single font can contain the GBK extended font of maximum 16 dot matrix (12×12 or 16×16 dot matrix), or GB2312 secondary font of maximum 32 dot matrix (12×12, 16×16, 24×24,32×32); ● The font can be combined for supper large font library at most 28 MB. The 0x98 or 0xF2 command is the first font address when combination,; ***Example: The 32 dot matrix UNICODE font will take up the 8 MB font space. We can allocate the***	0x20=12×12 GBK Song 0x21=16×16 GBK Song 0x22=24×24 GBK Song 0x23=32×32 GBK Song 0x24—0x3B：Free

		Lib_ID=0x20-0x27 space to it. The next font will start from 0x28. Lib_ID=0x20 when using the 0x98 command.	

➢ <C_Mode>: Select the text display and encoding mode as follows:

Bit No	Bit7-Bit4	Bit3-Bit0
Definition	Display mode	Character encoding mode
Description	Bit7=1: Text Foreground Color Display Bit7=0: Text Foreground Color Non- Display Bit6=1: Text Background Color Display Bit6=0: Text Background Color Non-Display Bit5=1: Vertical Text Display Bit5=0: Horizontal Text Display Bit4=1: Automatic Adjustment Character's X Spacing Bit4=0: Dot Matrix Size Display According To Setting	0x00 8bit encoding (At most 256 characters in font) 0x01 GB2312 Chinese internal encoding 0x02 GBK Chinese extended code or Korean HANGUL encoding 0x03 BIG5 Traditional Chinese encoding 0x04 SJIS Japanese encoding 0x05 UNICODE UNICODE encoding (UTF16) 0x06-0x0F: Undefined

Description:

<C_Mode>: The high bits are bit7- bit 4 and the low bits are bit 3- bit 0.

When C_Mode.4=1 enabled, characters must be 8 bit encoding and download font that needs to rotate 90 ° (TS3, vertical mode 1)

Examples of display mode setting:

Foreground Color	ON	ON	OFF
Background Color	ON	OFF	ON
<C_mode> high Value 2bit	0xC*	0x8*	0x4*
Character Display Effect	B area	A area	C area

- <C_dots>: Characters size setting

C_Dots	Font type (C_Mode: Bit3-Bit0)		
	0x00 or 0x05	0x01-0x04	
		ASCII	Non-ASCII
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	—	40*80
0x09	12*12	—	48*96
0x0A	16*16	—	56*112
0x0B	24*24	—	64*128
0x0C	32*32	—	80*80
0x0D	40*40	—	96*96
0x0E	48*48	—	112*112
0x0F	56*56	—	128*128
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		
0x1E	64*48		

Example:

If C_Mode=0x*1 (Internal code encoding), then C_dots=0x07 will display 64*64 Chinese character and 32*64 ASCII.

- <Fcolor>: The foreground color of character display
- <Bcolor>: The background color of character display
- <String>: The string data. The character spacing is set with the 0x41 command and automatically wrap at the end of the line.

Example：

AA 98 00 80 00 30 23 C1 03 F8 00 00 1F 42 65 69 6A 69 6E 67 20 44 57 49 4E 20 54 65 63 68 6E 6F 6C 6F 67 79 CC 33 C3 3C

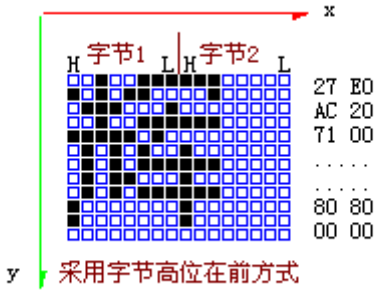
AA 98: Select font

00 80: The coordinate of x axis (Hex mode)
00 30: The coordinate of y axis (Hex mode)
23: <Lib_ID>, 32 dot matrix GB2312 Song font type
C1: <C_Mode>, the background color and foreground color are both displayed. Horizontal Text Display
03: <C_dots> selects 64*64.
F8 00: The foreground color is red.
00 1F: The background color is blue.
42 65 69 6A 69 6E 67 20 44 57 49 4E 20 54 65 63 68 6E 6F 6C 6F 67 79: The print string
Display a string "Beijing DWIN Technology" with the 0x98 command on (128,48), 32 dot matrix in GB2312 Song front type. The foreground color is red and the background color is blue. The result is shown as followings:

The user only can use custom font library to display ASCII (Or other fonts of 8 bit encoding with the 0x98 command. The user can't use 0x01-0x04 encoding mode (GB2312, GBK, BIG5, SJIS and GBK). In 0x01-0x40 encoding mode, UART LCM will automatically use the 0x00 font to display ASCII. The 0x00 font (DWINASC.HZK file) contents description is listed as below:

The description of 0x00 font (DWINASC.HZK) Content				
ASCII Dot matrix	Character Number	Bytes Number of per word	Occupied The Memory Size	Starting Storage Address
8*8	128	8	2KB	0x0:0000
8*12	128	12	3KB	0x0:0800
6*12	128	12	1.5KB	0x0:1400
8*16	128	16	2KB	0x0:1A00
12*24	128	48	6KB	0x0:2200
16*32	128	64	8KB	0x0:3A00
20*40	128	120	15KB	0x0:5A00
24*48	128	144	18KB	0x0:9600
28*56	128	224	28KB	0x1:0000
32*64	128	256	32KB	0x1:7000

Please consult DWIN technical supporters concerning the word 0x00 modify mode.
All files are encoded in X direction from left to right and y direction from top to bottom in font. The storage mode is high level first scan (Most Significant Bit, MSB). 1 byte=8 bit. Bits are written from high bit to low bit. The mode is shown as below:



Recommend that the user selects the dot matrix font extraction software TS3 to generate font.

3.6.3 Set /Cancel Textbox Limitation (0x45)

Set Textbox Limitation

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

Rx: None

- <Xs> <Ys> <Xe> <Ye>: The coordinates of textbox;

The text will be automatically wrapped within the textbox limitation area after textbox coordinates being set.

Cancel Textbox limitation

Tx: AA 45 00 CC 33 C3 3C

Rx: None

The text will be automatically wrapped within the full screen after canceling the textbox limitation.

3.7 Put Pixel Command (0x50, 0x51, 0x74, 0x72)

3.7.1 Put Pixel in specific coordinate (0x50, 0x51)

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

Rx: None

- <CMD>
  - 0x50: Put pixel with background color (Delete pixels)
  - 0x51: Put pixel with foreground color (Set pixels)
- < (x0, y0) (x1, y1) …… (xi, yi) >: A maximum number of 62 points could be displayed in one serial port data frame.

Example

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

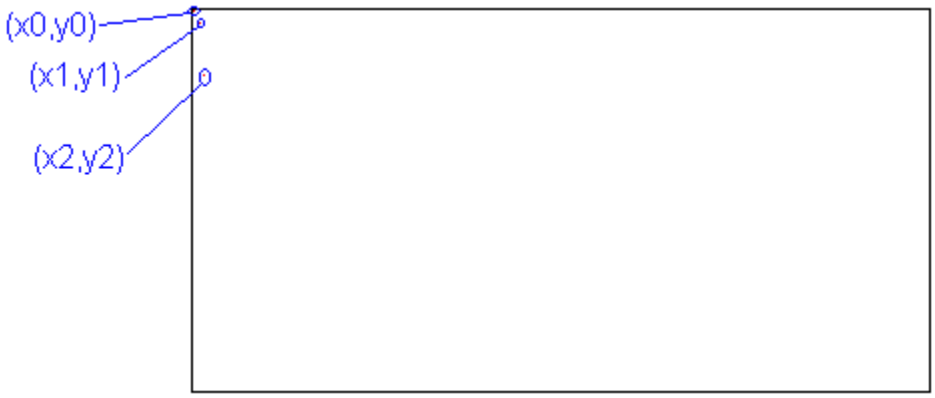
AA 51: Put pixel with the foreground color;

00 00 00 00: (0,0)

00 03 00 06: (3,6)

00 05 00 20: (5,32)

Put 3 pixels with the foreground color on (0,0) (3,6)(5,32). The result is shown as follows:



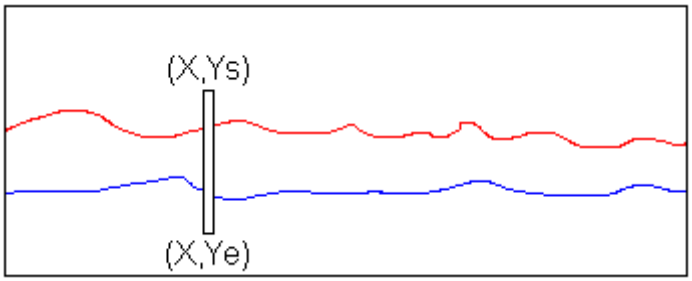
3.7.2 Dynamic Curve Display (0x74)

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

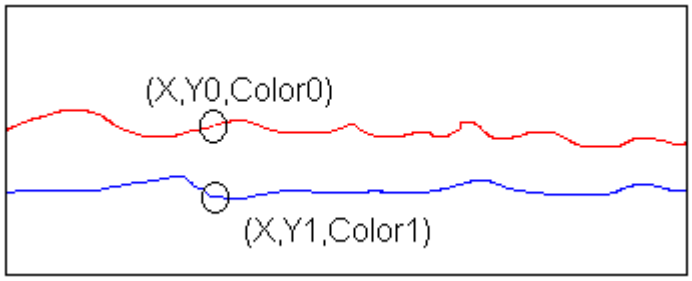
Rx: None

This command is mainly used to rapidly display multiple changing (Dynamic) curves in a window for the user. The detailed procedures are shown as follows:

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



Step 2： Put pixels on (X,Yi), using <Fcolori>.



The palette properties which are set by users will not be altered with this command.

3.7.3 Direct Memory Operation (0x72)

Tx: AA 72 <Address_H:M:L> <Point_data0……Point_datan> CC 33 C3 3C

Rx: None

This command is mainly used to download pictures to UART LCM. Generally, it's not necessary for the users.

3.8 Draw Line Command (0x56, 0x5D, 0x75, 0x76, 0x78)

3.8.1 Connect Specific Pixels (0x56, 0x5D)

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

Rx: None

- <CMD>

  0x56: Connect specific pixels to a line with the foreground color (Use the command 0x40 to set the foreground color).

  0x5D: Connect specific pixels to a line with the background color (Use the command 0x40 to set the background color).

- < (x0, y0) (x1, y1) …… (xi,yi)>: The coordinates of connected pixels.

Example：

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

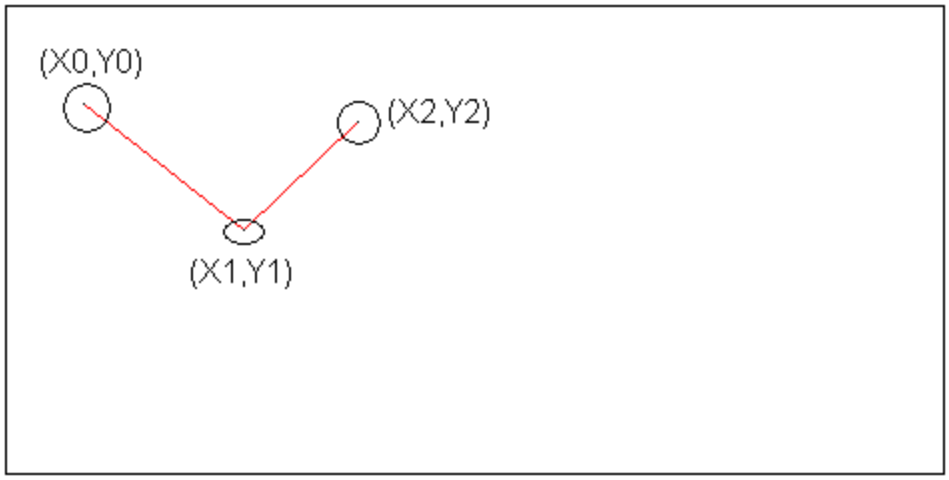
Connect three pixels to a line with the foreground color on (40, 50) (120,112) (177,58).

AA 56: Connect specific pixels to a line.

00 28 00 32: Hexadecimal value of (40,50);

00 78 00 70: Hexadecimal value of (120,112);

00 B1 00 3A: Hexadecimal value of (177, 58). The result is shown as follows:



3.8.2 Draw Spectrum Command (0x75)

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

Rx: None

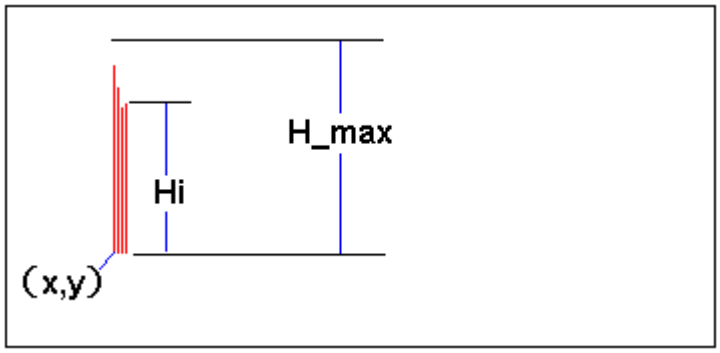
- < (x,y)>: x is the starting coordinates of spectrum. Set x=x+1 after displaying one spectrum line; Y is a level of benchmark coordinate of spectrum. Y is the starting coordinate of spectrum. (y-Hi) is the end coordinate of spectrum.
- <H_max>: The maximum height of spectrum line.

If H_max=0x01-0xFF, the height of spectrum line Hi is 1 byte variable.

If H_max=0x00, the following 2 bytes is Hmax. Hi is 2 bytes variable.

- <H0……Hi>: The height of one spectrum line, one byte or two bytes.

The color of spectrum line is set with the 0x40 command. Spectrum lines (The height of Hi) are displayed with the foreground color. Other spectrum lines (The height of H_max-Hi) is displayed (Erased) with the background color when drawing spectrum line.



3.8.3 Draw Line Chart (0x76)

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

Rx: None

- <x>: The starting coordinates of line chart. If connect one pixel, x=x+x_dis;
- <x_dis>: The increments of x coordinate;
- <Y0……Yi>: The vertex coordinates of line chart. Draw connection line with the foreground color.

This function of command is similar with the 0x56 command except that coordinate X here is auto-calculated by HMI for fast connecting.

3.8.4 Connect Line by Offset (0x78)

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

Rx: None

- <x, y>: The starting coordinates of a connection line.
- <dxn, dyn>: 1 byte offset of x and y. The highest bit (.7) is sign bit. “1” indicates negative.

3.9 Draw Arc Command (0x57)

3.9.1 Draw Arc or Circle Area (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>: Arc or curve display mode

  0x00: Reverse the color of the arc or curve

  0x01: Display the foreground color of the arc or curve (Use the command 0x40 to set the foreground color)

  0x02: Reverse the color of the specific circle area

  0x03: Display the foreground color to the specific circle area (Use the command 0x40 to set the foreground color)
- <X_i> <Y_i>: The center coordinates of arc or curve
- <R_i>: The radius of arc or curve (0x01-0xFF)

Example：

AA 57 01 00 60 00 60 40 CC 33 C3 3C

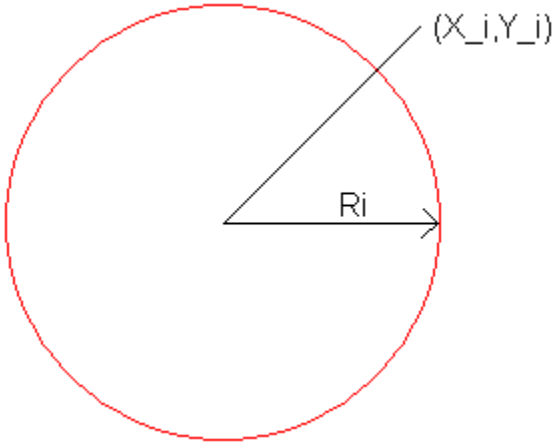
Draw an arc with the foreground color. The circle center is on (96, 96) and the radius is 64.

AA 57: Display an arc curve;

01: <Type_0>: Display the specific arc with the foreground color;

00 60 00 60: The coordinates on (96,96);

40: The circular radius is 64. The result is shown as follows:



3.9.2 Draw Arc Segment (0x5704)

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

Rx: None

- <X> <Y>: The center coordinates of arc
- <R>: The radius of arc (0x0001-0x03FF)
- <A_S>: The starting angle of arc, 0x00-0x02D0 (0-720). The unit is 0.5°.
- <A_E>: The ending angle of arc, 0x00-0x02D0 (0-720). The unit is 0.5°.

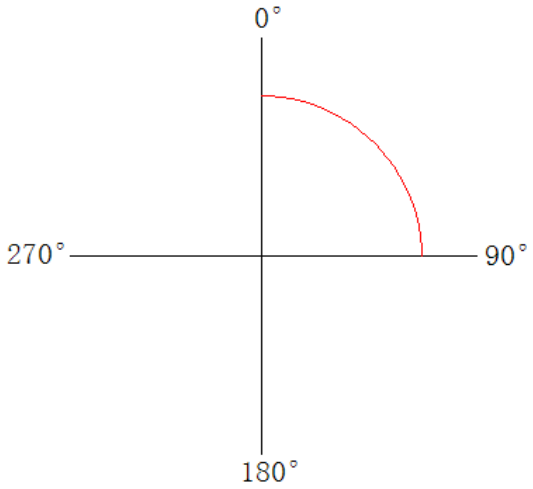
Display color is the foreground color which is set with the 0x40 command.

Example：

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

Display an arc with the foreground color. The center coordinates are (96, 96) and radius is 64. The

start angle is 0° and the end angle is 90°. The result is shown as follows:



3.10 Draw Area Command

3.10.1 Draw the Rectangle or Rectangle Area (0x59, 0x69, 0x5A, 0x5B, 0x5C)

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 with the foreground color (Use the command 0x40 to set the foreground color). The width of line is one dot matrix;

0x69: Display rectangle with the background color (Use the command 0x40 to set the background color). The width of line is one dot matrix;

0x5A: The area being filled with the background color (Use the command 0x40 to set the background color);

0x5B: The area being filled with the foreground color (Use the command 0x40 to set the foreground color);

0x5C: Reverse the color of the specific rectangle area (XOR 0xFF operation). The color will be restored to what it was.

- <Xs_i> <Ys_i> <Xe_i> <Ye_i>: (Xs_i,Ys_i) are the left-up coordinates of rectangle or rectangular area; (Xe_i,Ye_i) are the right-down coordinates of the rectangle or the rectangular area.

Example：

AA 5C 00 40 00 40 00 80 00 80 CC 33 C3 3C

Reverse the color of the rectangular area. The left-up coordinates are on (64, 64) and the right-down coordinates are on (128,128) as follows:

3.10.2 Fill Area (0x64)

Tx: AA 64 < (x, y) > < Color> CC 33 C3 3C

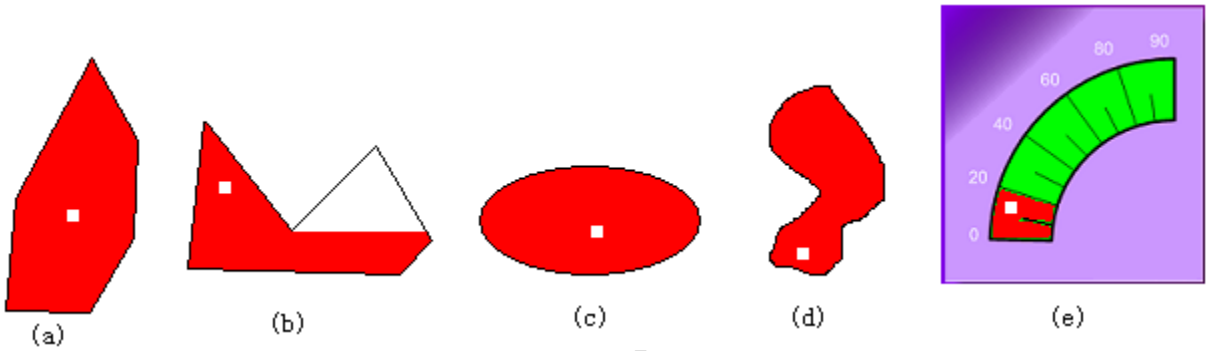
Rx: None

- < (x,y)> : The seed point coordinates of the area being filled;
- Color: Full color;

Note:

a. The initial color of the area being filled is similar with the color of seed point position. Otherwise full area of seed points in the same color only (other color area as the boundary treatment);

b. This command is only applied to "convex polygons". Some area cannot be filled (as shown in the following figure b) to "sunken polygon area" by setting different seed point positions to achieve "sunken polygon area" completely filling.



c. It will not change the properties of palette.

Example:

AA 64 00 64 00 64 F8 00 CC 33 C3 3C

White point is the position of seed point and been filled actually.

3.10.3 Fill Double Color Bitmap (0x73)

Tx: AA 73 Color0 Color1 (Xs, Ys) (Xe, Ye) (X, Y) Data CC 33 C3 3C

Rx: None

- Color0: 0 bit represents the color being filled, 16 bit;
- Color1: 1 bit represents the color being filled, 16 bit;
- (Xs, Ys): The left-up coordinates of bitmap display area boundary.
- (Xe, Ye): The right-down coordinates of the bitmap display area boundary.
- (X, Y): The starting coordinates of the bitmap being filled.
- Data is the double color bitmap information.

3.11 Clear Screen Command (0x52)

Tx: AA 52 CC 33 C3 3C

Rx: None

Clear the full screen with the background color (Use the 0x40 command to set) (Clear screen)

3.12 Move Specific Area Command (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: The specific area to loop and shift right from right to left. Move the far left area to the far right area.

0x61: The specific area to loop and shift left from left to right. Move the far right area to the far left area.

0x62: The specific area to shift right: from right to left. Lost the far left area and fill the far right area with the background color filling (Use the command 0x40).

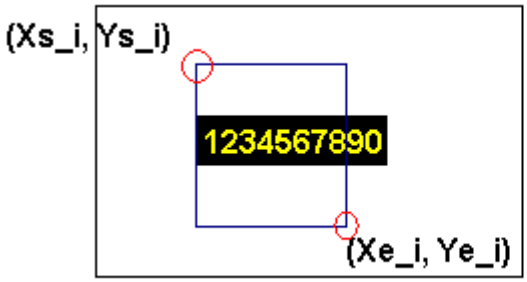
0x63: The specific area to shift left: from left to right. Lost the far right area and fill the far left area with the background color filling (Use the command 0x40).

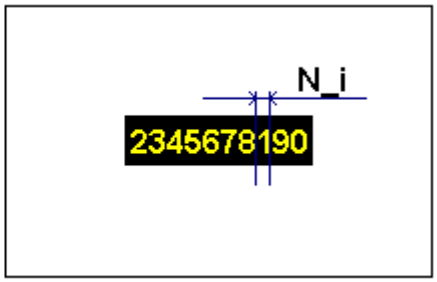
- <Xs_i> <Ys_i> <Xe_i><Ye_i>: (Xs_i, Ys_i) are coordinates of specific area at the left-up corner and right-down corner.
- <N_i>: The number of pixels in moving area, from 0x01 to 0x0F.

Example：

AA 60 00 40 00 40 00 80 00 80 08 CC 33 C3 3C

Move 8 pixels from left-up to right-down. The left-up coordinate is on (64,64) and the right-down coordinates is on(128,128). The result is shown as follows:





This command is not supported in the reflection 90° display version.

3.13 Display Picture or Icon Command (0x70, 0x71, 0x99, 0xE2, 0x7B, 0x9E,0x97)

3.13.1 Display Picture (0x70)

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

Rx: None

<Pic_ID>: The index ID (Corresponding to the command 0xE2) of pictures saved in the UART LCM Flash Memory

Example

AA 70 00 CC 33 C3 3C : The 0 picture is displayed and saved in UART LCM.

AA 70 01 02 CC 33 C3 3C : The 258th picture is displayed and saved in UART LCM.

If the number of saved pictures exceeds 256, the pictures' ID will be overtook 255 and be used with 2 bytes directly.

3.13.2 Display Picture & Calculate the CRC Checksum (0x7B)

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

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

- <Pic_ID>: The index ID (Use the command 0xE2) of the saved pictures in UART LCM Flash Memory
- <CheckSum_H:L>: The CRC-16 checksum of content of the current picture.

This command is used to check picture which has been downloaded in Flash in order that the download is correct.

3.13.3 Display Cutting 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>: The ID (Corresponding to the command 0xE2) of the saved pictures in UART LCM Flash

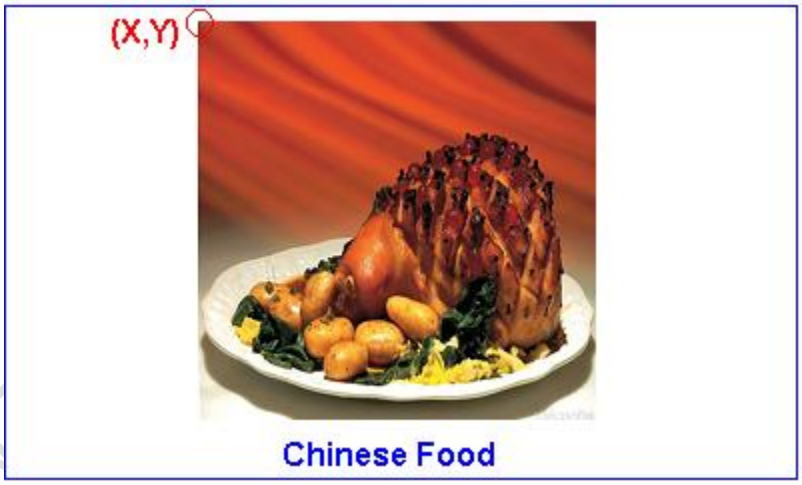
Memory

- <Xs> <Ys> <Xe> <Ye>: (Xs, Ys) is the left-up coordinates of the original picture of area that users will cut; (Xe,Ye) are the right-down coordinates.
- <X> <Y>: The left-up coordinates of the original picture of area that users have already cut.

Example1：

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

Cut the eighth picture's area on (400,0) (799,400) and display on (200,200) in current screen. The result is shown as follows:

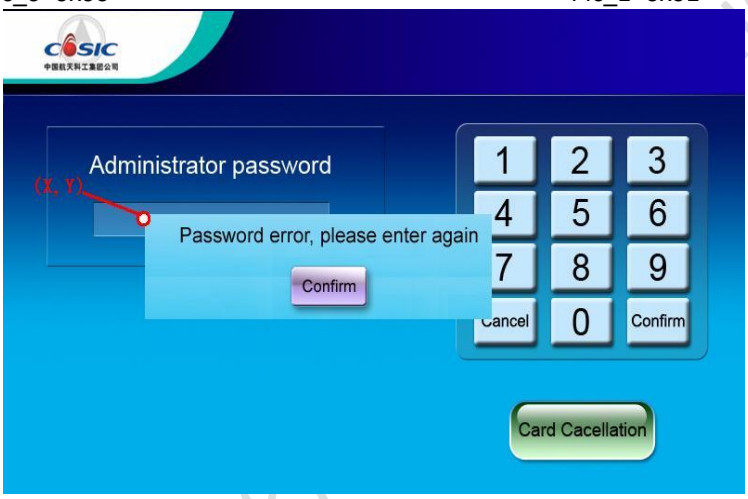


Example2：

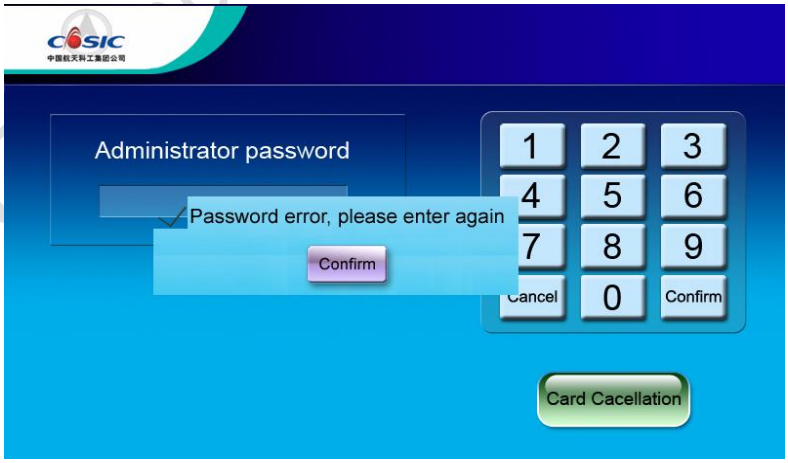
AA 9C 01 00 D0 00 D0 02 0A 01 3E 00 BA 00 AE CC 33 C3 3C

Cut the first picture's area on (208,208) (522,318) and display on (186,174) in current screen. The result is shown as follows:

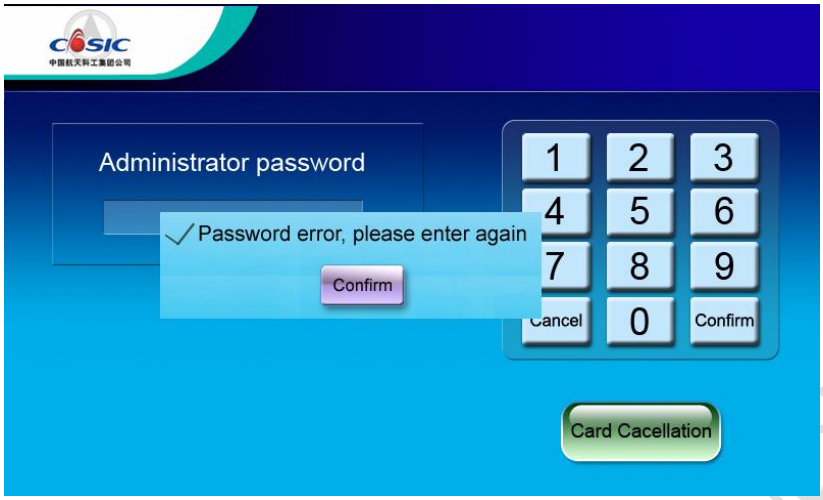
Pic_0=0x00 Pic_1=0x01



And then cut the first picture's area on (255,412) (308,442) and display on (186,174) in current screen using the 0x9C command. The result is shown as follows:



But use the 0x9D command, the result is shown as



The background color of cutting picture cannot be displayed to achieve "transparent" effect with the 0x9C and 0x9D commands. Note: The background must be solid color when using the 0x9C and 0x9D command. The background color of cutting picture can be automatically restored using the current background before executing the 0x9C command. But the background cannot be automatically restored before executing the 0x9D command. Pictures can be overlaid with the 0x9D command.

3.13.4 Display Self-defined Icon (0x99)

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

Rx: None

- <Xn,Yn>: The coordinates of the users' icon
- <Icon_IDn>: The index ID of the users' icon, 2 bytes

The icon of user needs pre-define in 0x1D configuration file. The details refer to 3.26.4 section

3.13.5 Save Current Screen Picture to UART LCM (0xE2)

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

Rx: None

- <Pic_ID>: The picture index ID to be saved

3.13.6 Save Current Screen Picture Area to Temporary Buffer (0xE9)

Tx: AA E9 Xs Ys Xe Ye CC 33 C3 3C

Rx: None

- Xs Ys Xe Ye: The left-up and right-down coordinates.

0xE9, K600, H600 and K600+_XGA only

3.13.7 Restore Picture Area Saved in Temporary Buffer (0x7F)

Tx: AA 7F Xs Ys Xe Ye CC 33 C3 3C

Rx: None

- Xs Ys Xe Ye: The left-up and right-down coordinates of the restored area.

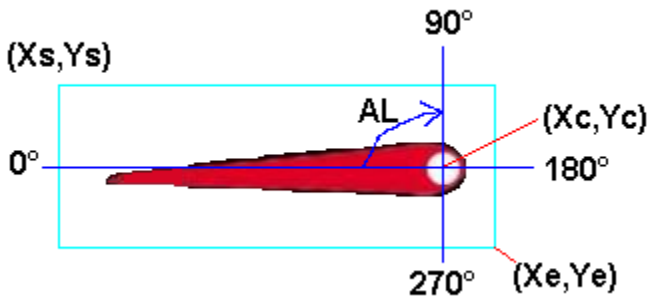
0x7F, K600_XGA only

3.13.8 Display Cutting Icon after Rotation (0x9E, *H600 and K600+ only*)

- Tx: AA 9E <Mode> <Pic_ID> <Xs> <Ys> <Xe> <Ye> <Xc> <Yc> <AL> <Xc1> <Yc1> CC 33 C3 3C
- Rx: None
- <Mode> 0x00=Transparently cut, (Xs, Ys) location is the background;
- 0x01=Opaque cut;
- <Pic_ID>: the index ID of saved picture in the UART LCM flash memory (Use the command 0xE2 to set). Two bytes (0x0000-0xFFFF);
- <Xs> <Ys> < Xe> <Ye> (Xs, Ys) are the left-up coordinates of the cutting area on the primary

picture. (Xe, Ye)are the right-down coordinates;

- <Xc> <Yc> are the coordinates of the icon rotation center in the original pictures;
- <AL>: The rotate angle of icons. 2 bytes ranges from (0-719) and the unit is 0.5°;
- <Xc1> <Yc1>: The rotation center coordinates of the cutting down icon in the current screen location



Tip: Put the original pointer icon at the 0 calibration. The dashboard graduation will be uniformly increased by clockwise. The angle of rotation and the actual parameters are linear corresponded.

3.13.9 Display ICON (0x97, *K600+ only*)

Tx: AA 97 <X> <Y> <Lib_ID> <MODE> <ICON_ID0>……<ICON_IDn> CC 33 C3 3C

Rx: None

- <X> <Y>: The first coordinates of icon. The subsequent icon will be automatically calculated coordinates. The Dis_X is set with 0x41 command decides the icon spacing. On the other word, the second icon position= X+ width of icon + Dis_X
- <Lib_ID>: The position of saved *.ICO file in (0x00-0x3B)
- <MODE>: The options of background. 0x00=filter out the background; others= display background
- <ICON_ID0>……<ICON_IDn>: The ICON index ID of the icon in *.ICO files. Each ID has two bytes from 0x0000-0xFFFF

Note:
The icon file (*.ICO file) is generated only with the DWIN icon generation tool software and ***the specific font position can be only used SD card interface to download in DWIN screen***.
Other picture files such as ICON, JPG, BMP can be generated with the DWIN icon generation tool software according to the users specific ID in order to be called easily. But the dot matrix of these files cannot be exceeded 255 X 255. You can download the DWIN icon generation tool software from DWIN website.

The 0x97 ICON display command is mainly used to solve the problem which is 0x71 command cutting icon on the picture. And the use is need to alignment.
The special effects are displayed perfectly with the 0x97 command, such as dashboard rotation, WordArt and so on.

3.14 Backlight Brightness Control Command (0x5E, 0x5F)

3.14.1 Turn off Backlight (0x5E)

Tx: AA 5E CC 33 C3 3C
Rx: None

3.14.2 Set Touching Function (Keyboard control) Backlight Mode (0x5E)

Tx: AA 5E 55 AA 5A A5 <V_ON> <V_OFF> <ON_TIME> CC 33 C3 3C
Rx: None

- <V_ON>: The backlight is automatically turned on brightness after clicking on touch screen (or keyboard).
- <V_OFF>: The backlight is automatically turned off brightness when touch screen (or keyboard) does not be clicked of a period of time.
- <ON_TIME>: The time of backlight is on. The unit is 0.5 second.

The backlight mode will be preserved. Backlight will be lit at first time and it will not be processed when the backlight is turned off.
The Backlight brightness of touching function (Keyboard control) is enabled with the 0xE0 command.

3.14.3 Open the Backlight to Max Brightness (0x5F)

Tx: AA 5F CC 33 C3 3C
Rx: None

3.14.4 Adjust Backlight Brightness (0x5F)

Tx: AA 5F <PWM_T> CC 33 C3 3C
Rx: None

- <PWM_T>: The backlight brightness values are controlled with PWM (0x00-0x3F); 0x00 is the closed backlight and 0x3F is maximum brightness of backlight.

The backlight brightness will not be saved; the brightness will be set to maximum when turn on the screen.
The Backlight brightness cannot be adjusted with PMN for backlight mode CCFL in UART LCM. But the backlight brightness is only controlled "on or off".

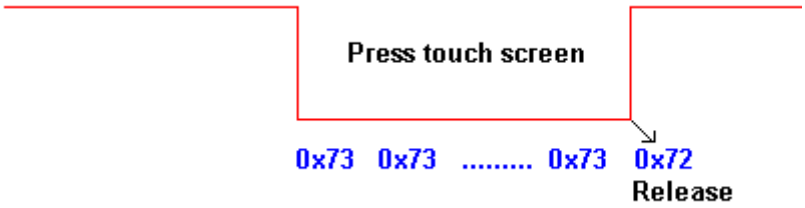
3.15 Touch screen Operation Command (0x72, 0x73, 0x78, 0x79, 0xE4)

3.15.1 Automatically Upload the Touch Position (0x72, 0x73)

The coordinates of touch positions are automatically uploaded with UART LCM as following format when pressing the touch screen.
Tx: None
Rx: AA 73 <X> <Y> CC 33 C3 3C when press the touch screen, it will uploaded one or more times (Use

command 0xE0 to set)
AA 72 <X> <Y> CC 33 C3 3C when release the touch screen, it will uploaded only once (Use command 0xE0 to set)

- <X> <Y>: The coordinates of touch position, corresponding to screen resolution



If users enable the touch interface processing function (Use command 0xE0 to set) and click on the invalid area which is not uploaded the location of coordinates, it will not be uploaded with the 0x72 and 0x73 commands when clicking on the touch screen.

3.15.2 Automatically Upload the Key Code of Touching Function (0x78, 0x79)

If users enable the interface of the touching control or keyboard function (Use the command 0xE0) and the key code of the touching control or keyboard function to transmit back, which is 2 bytes will be automatically uploaded in UART LCM (Defined with 0x1E and 0x1B configuration files) when click the valid area or keys. The touching code or key code is preset by the user.
Tx: None
Rx: AA 78 <Touch_Code> CC 33 C3 3C
Note: 0x79 is corresponding to 0x73 (Pressed the touch screen); 0x78 is corresponding to 0x72 (Released the touch screen)

3.15.3 Touch Screen Calibration Mode (0xE4)

Tx: AA E4 55 AA 5A A5 CC 33 C3 3C
According to the screen prompt operation click screen ” left-up ”, “right-up” and ” right-down” on the white cross of the suggestion touching position after sending commands, the UART LCM will upload the following commands when calibration is completed:
Rx: AA E4 4F 4B CC 33 C3 3C

DWIN's unique drift compensation technique can guarantee that products only need to calibrate once after equipped in product life circle unless users re-equipped the touch screen;

3.16 Work Mode Configuration Command (0xE0)

Tx: AA E0 55 AA 5A A5 <TFT_ID> <Bode_Set> <Para1> CC 33 C3 3C
Or: AA E0 55 AA 5A A5 <TFT_ID> <Bode_Set> <Para1> <Para2> CC 33 C3 3C
Rx: AA E0 <TFT_ID> <Bode_Set> <Para1>CC 33 C3 3C
Or: AA E0 <TFT_ID> <Bode_Set> <Para1> <Para2> CC 33 C3 3C (0xE0, H600 and K600+ only)
The 0xE3 command is only valid for V6.0 or above version of the internal software; It is almost the same with 0xE0 command, the only difference is , when the internal software is V6.0 or above, the parameters being modified by 0xE0 will not be saved after powering off, in other words, when the UART LCM is being switched on again, the parameter will be back to what it was. However, the modified parameters will be saved if using 0xE3.

- <TFT_ID>: To set TFT panel configurations ***(V5.3 or later versions are not open to users; please write to 0x00):***
- <Bode_Set>: <Bode_Set>: Set the baud rate of serial communication as following (The default value is 0x07 from DWIN factory):

Bode_set	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Baud rate	1200	2400	4800	9600	19200	38400	57600	115200
The following baud rate is not supported by the PC port, please set carefully								
Bode_set	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
Baud rate	28800	76800	62500	125000	250000	230400	345600	691200
The following baud rate is only supported by K600+ kernel serial port setting								
Bode_set	0x10	0x11 (Only set by SD card)						
Baud rate	921600	Users define the baud rate. Baud rate=6250000/R5:R9, R5 and R9 are set by CONFIG.TXT via SD card.						

- <Para1>: Configure the touch screen and keyboard work mode as follows:

Para1	Bit Description
.7	0= After clicking on the touch screen, the 0x73 command is automatically uploaded when releasing the touch screen. *It must be set to 0 in the touch mode.* 1= After clicking on the touch screen, it does not upload 0x72 command when leaving the touch screen.
.6	In the coordinates returning mode: 0=After clicking on the touch screen, the 0x73 command is automatically uploaded with the interval of time of 100 ms until releasing the touch screen. 1=After clicking on the touch screen, the 0x73 command is automatically uploaded only once. On the touch mode, Para1.0=1: 0= After clicking on the touch screen, the 0x79 command is automatically uploaded with the interval time of 100 ms until releasing the touch screen. 1= After clicking on the touch screen, the 0x79 command is automatically uploaded only once.
.5	0= After clicking on the touch screen, the UART LCM touch interface will not be switched automatically. 1= After clicking on the touch screen, the UART LCM touch interface will be switched automatically according to 0x1E configuration files.
.4	0= Backlight will not be controlled by touch screen or keyboard. 1= Backlight will be controlled by touch screen or keyboard. At the same time, the user can also through the 0x5E/0x5F command forced switch to control backlight.
.3	0= Buzzer switched on for touch screen or keyboard; 1= Buzzer switched off for touch screen or keyboard
.2	0= 0°display; 1= 90°display.
.1	0= In the touching function mode, buzzer is always switched on; 1= in the touching function mode, buzzer is only screamed once when clicking on the effective position; if ***Para1.1=1, set Para1.3＝1 at the same time.***
.0	In the touching function mode: 0= Not-Upload 0x79 command; 1= Upload 0x79 command.

- <Para2>: Configuration mode, set up as follows: ***（This function does not support the previous internal software V6.0. It is unnecessary to write this byte. H600 and K600+ only）***

Para2	Bit Description
.7	0=The previous V5.0 version refresh way (Automatically high-speed update, together with 0x71+0x98 will cause jittering);

	1=automatically refresh to display with 200ms. The command is executed automatically after the extension of the refresh time to ensure that specification can be simultaneously displayed.
.6	Return back the touch screen coordinates mode: 0= The touch screen coordinates don't changed the following Para1.2 when setting them back. 1=The touch screen coordinates are changed following Para1.2 when setting them back (Deflection 90°).
.5	0=Display deflection 180° (Reverse angle display); 1=Normal angle display.
.4	0=Text display (0x53、0x54、0x55、0x6E、0x6F、0x98), the picture background does not restored automatically and the text background color is ignored; 1=When text is displayed, the picture background does not automatically restored.
.3	Reservation and set to 1.
.2	0=Stop automatically refresh. Display must be refreshed with the 0xDo command; 1=Enable automatically refresh. The refresh interval is decided by Para2.7
.1	Reservation and set to 1.
.0	Reservation and set to 1.

Software above version V6.0, configuration parameters don't be saved with the 0xE0 command when power is off.

If you need to save it, please use 0xE3 command、DWIN configuration tool software or SD card configuration file to configure. If re-power on, parameter values will be restored to the user preset by the configuration tool (or configuration file).

3.17 Timing and Cycle Execution Command (0x9A)

This command is mainly used for users to save regularly specification (Example use 0x71 to display animation) in UART LCM, without the user intervention in order to reduce codes.

3.17.1 Open timing and Cycle Function

Tx: AA 9A Pack_ID CC 33 C3 3C

Rx: None

- <Pack_ID> is automatic cycle command groups ID, 0x00-0x0F. Each group has 8KB and contains maximum number of 64 UART LCM commands, each UART LCM command ranges from 128KB.

UART LCM can only operate 1 group specification, which is defined in 0x1C configuration file. UART LCM contains up to 16 command groups.

3.17.2 Close Timing and Cycle Function

Tx: AA 9A FF CC 33 C3 3C

Rx: None

3.18 Temporary Buffer Operation Command (0xC0, 0xC1, 0xC2)

The content of temporary buffer can be changed with 0x52 (Clear screen).

3.18.1 Write Temporary Buffer (0xC0)

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

Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Word0……Wordn> is data to be written.

3.18.2 Read Contend of Temporary Buffer (0xC2)

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

Rx: AA C2 <Data_Pack> CC 33 C3 3C

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Data_Length>: The length of the data read (Word) is 2 bytes.
- <Data_Pack>: The data read.

3.18.3 Point Setting by Temporary Buffer (0xC101)

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

Rx: None

- <Address>: The starting address (word address) of temporary buffer (RAM) is totally 40k word from 0x0000 to 0x9FFF.
- <Point_Number>: The number of pixels, at most 13653. Each point is consisted of three words. Point data format is defined in temporary buffer: (X, Y, Color).

3.18.4 Connect Lines with Data in Temporary Buffer (0xC102)

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

Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Line_Number>: The maximum number of lines is 8191. Every line takes up 5 word unit.

line data format defined in temporary buffer: (Xs, Ys, Xe, Ye, Color).

3.18.5 Display Line Chart with Data in Temporary Buffer (0xC103)

Tx: AA C1 03 <Address> <x> <y> <Line_Number> <D_x> <Dis_x> <K_y> <Color> CC 33 C3 3C

Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <x> is the starting coordinate.
- <y>: The zero coordinate (The lowest point); the actual coordinate of connection point equals to y-Ly.
- <Line_Number>: The maximum number of lines is 40960 from 0x000 to 0x9FFF. Every line takes up 1 word unit.
- <D_X>: The points' distance in read buffer from 0x01 to 0xFF, i.e. Address=Address+D_x.
- <Dis_X>: The increment of X-axis from 0x01 to 0x0F, i.e. x=x+Dis_x.
- <K_y>: The amplification factor of Y-axis from 0x00 to 0xFF; 1/16 is the unit, i.e. K_y=32 means amplify Y-axis twice.
- <Color>is the color of lines without changing any variation in system palette.

The line data format in temporary buffer is defined as Ly (2 bytes), which is the height of point.

3.18.6 High-speed Display Line Chart with Data in Temporary Buffer (0xC104)

Tx: AA C1 04 <Adr1> <x> <y> <Line_Number> <D_x> <Dis_x> <Color1> <Addr0> <Color0> CC 33 C3 3C

Rx: None

- <x> : The starting coordinate .
- <y>: The zero coordinate (the lowest point); the actual coordinate of connection point equals to y-Ly.
- <Line_Number>: The maximum number of lines is 40960. Every line takes up 1 word unit.
- <D_x> Always sets to 0x01.
- <Dis_X>: The increment of X-axis, from 0x01 to 0x0F; i.e. x=x+Dis_x after connecting one line.
- <Addr0> <Addr1>: The starting address (word address) of temporary buffer (RAM), totally 40k word from 0x0000 to 0x9FFF.

  <Addr0>: The first address of the history curve of erasing in current window.

<Addr1>: The first address of the history curve of displaying in current window.
The absolute value is 100 from Addr1 to Addr0, if the window connects 100 data points and the temporary buffer is only one curve,.

- <Color0> <Color1> is the color of lines without any variation in system palette.
  <Color0> should be set into the background color of the window.
  <Color1> should be set the color of curve that is to be displayed.

The line data format in temporary buffer is defined as Ly (2 bytes), which is the height of point.

This command is generally similar with the 0xC103 command except that:
a. The points' distance in read buffer is fixed to 1.
b. Avoided flashing during refresh. The previous point 0 (Defined by Addr0 and Color0) will be erased, before connecting next point (Defined by Addr1 and Color1) in order to display ***without flashing***.
The speed of display line chart is the maximum 5500 points per second in this command when the baud rate is 115200bps.

3.18.7 Display and Zoom Line Chart with Data in Temporary Buffer (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>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <x>: The starting coordinates of X-axis.
- <y>: The zero coordinate (The lowest point); the actual coordinate of connection point equals to y-Ly.
- <Line_Number>: The maximum number of lines is 40960 from 0x0000 to 0x9FFF. Every line takes up 1 word unit.
- <D_x> : The points' distance in read buffer from 0x01 to 0xFF; i.e. Address=Address+D_x after connecting one line.
- <Dis_x>: The increment of X-axis from 0x01 to 0x0F; i.e. x=x+Dis_x after connecting one line.
- <M_y> <D_y>: Amplification factor of Y-axis from 0x00 to 0xFF; and height equals to Y×M_y/D_y. For example, M_y=4, D_y=2 means amplify Y-axis twice.
- <Color>: color of lines without changing any variation in system palette.

The line data format in temporary buffer is defined as Ly (2 bytes), which is the height of point.

3.18.8 Display and Zoom Window Limited Two-way Line Chart with Data in Temporary Buffer (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>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <x>: The start of coordinate.
- <y>: The zero coordinate (The lowest point); the actual position of connection point equals to y-Ly or y+Ly.
- <Line_Number>: The maximum number of lines is 40960 from 0x0000 to 0x9FFF. Every line takes up 1 word unit.
- <D_x>: The points' distance in read buffer from 0x01 to 0xFF; For example, Address=Address+D_x after connecting one line.
- <Dis_x> : The increment of X-axis from 0x01 to 0x0F; i.e. x=x+Dis_x after connecting one line.
- <M_y> <D_y>: Amplification factor of Y-axis from 0x00 to 0xFF; and height equals to Y×M_y/D_y. For example, M_y=4, D_y=2 means amplify Y-axis twice.
- <Color>: Color of lines without changing any variation in system palette.

- <Ymin>: The coordinate of the lower limit on Y-axis (The lower limit of window);
- <Ymax>: The coordinate of the upper limit on Y-axis (The upper limit of window);

The line data format in temporary buffer is defined as Ly (2 bytes), as following direction:
Ly.15 is the connecting direction, 0=positive direction, 1=negative direction. Ly=0x8010 means negative direction whose height is 16.
Ly.14- Ly.0 is connecting height.

3.18.9 Use Temporary Buffer as Pixel Buffer (0xC107)

Clear vacant pixel buffer:

Tx: AA C1 07 00 <Address> <X_Len> <Y_len> CC 33 C3 3C
Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <X_Len>: The X-direction width of pixel buffer screen window (Pixel number), 0x0000-0x0FFFF.
- <Y_len>: The Y-direction width of pixel buffer screen window (Pixel number), 0x0000-0x0FFF.

Note: The size of the RAM address is occupied by the pixel buffer= (X*Y)/16, Example build a 64*64 pixel buffer from temporary buffer address 0x0000. Then occupy the temporary buffer address space is 0x0000-0x00FF.

Put pixel in pixel buffer:

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>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <X_Len>: The X-direction width of pixel buffer screen window (dots format), 0x0000-0x0FFFF.
- <Y_len>: The Y-direction width of pixel buffer screen window (dots format), 0x0000-0x0FFF.
- <Xs,Ys>: The coordinates of pixel buffer corresponding to the screen window.
- <Color> is the color of pixel /delete point, not affect 0x40 command of the color palette.
- < (X0,Y0)… (Xi,Yi)> are the coordinates of pixels. If cross border, then don't put pixels. Display actual coordinates in current page X=Xs+Xi Y=Ys+Yi
- <Mode>: Mode of drawing pixel.
  0x00=delete point in pixel buffer, 0x10=delete point in pixel buffer and current page simultaneously.
  0x01=put pixel in pixel buffer, 0x11=put pixel in pixel buffer and current page simultaneously.

Recover pixel buffer to current page:

Tx: AA C1 07 02 <Address> <X_Len> <Y_len> <Xs,Ys> <Color> CC 33 C3 3C
Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <X_Len>: X-direction width of pixel buffer screen window (Pixel number), 0x0000-0x0FFFF.
- <Y_len>: Y-direction width of pixel buffer screen window (Pixel number), 0x0000-0x0FFF.
- <Xs,Ys>: The coordinates of pixel buffer corresponding to the screen window.
- <Color>: Color of pixel, the same as current page displayed color.

Only put pixel in pixel buffer (01 mode). Users must clear pixel buffer before recover to current page.

This command is mainly provided "layer" for users in order to achieve of the operation of some pure color icon flexibly and expediently, such as the movement of cross cursor.

3.18.10 Display Multi-parameters with Temporary Buffer (0xC108)

Tx: AA C1 08 00 <Address> <Parameter_N> CC 33 C3 3C
Rx: None

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Parameter_N>: The number of multi-parameters, 0x01-0xFF

Note: Display as right-aligned.

The parameters data of temporary buffer are defined as follows:

Temporary Buffer Relative Address	Definition	Description
0x00	Mode	Mode.15:.12 The maximum integer digits number could be displayed; Mode.11:.8 The maximum decimal digits number could be displayed. 0 means no decimal digital; Mode.7 0=invalid integer bit zero is not displayed 1=invalid integer bit zero is displayed Such as if the mode.15:.12=3, Mode.7=1, 0.3 will be displayed as 000.3. Mode.6 0=text defines the background color with 0xC108 command 1=text automatically take the background color and display. The coordinates are (X-1, Y) and the background is pure color. Mode.5-Mode.4: Undefined and set to 0. Mode.3-Mode.0: Display the size of font (Based on Lib_ID=0, default ASCII font） 0=8*8 1=8*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 undefined
0x01	X	The starting coordinate of parameter display on X-axis
0x02	Y	The starting coordinate of parameter display on Y-axis
0x03	F_Color	The foreground color of parameter display, not-change palette setting
0x04	B_Color	The background color of parameter display, not-change palette setting
0x05	Parameter	Parameter data; 4 bytes, singed integer, Parameter.31 is the sign bit

The example of parameter setting

Display data	Parameter	Mode （display font 16*32 dot matrix 6.3 mode, an invalid integer bit 0 is not displayed）
12345	0x00 00 30 39	0x6005
123.450	0x00 01 E2 3A	0x6305
-123.450	0xFF FE 1D C6	0x6305

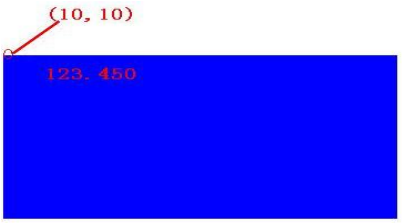
Example1:

AA C0 00 00 63 05 00 0A 00 0A F8 00 00 1F 00 01 E2 3A CC 33 C3 3C AA C1 08 00 00 01 CC 33 C3 3C

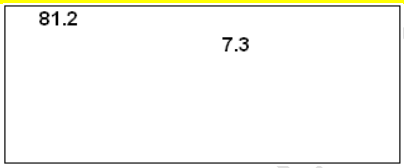
Display 123.450 at the starting coordinates (10, 10) as following:

AA C0 00 00 63 05 00 0A 00 0A F8 00 00 1F 00 01 E2 3A CC 33 C3 3C—Write temporary buffer

00 00: <Address>
63 05: From the above table, Mode: 0x6305
00 0A: X coordinate; 00 0A: Y coordinate (10,10)
F8 00: F_Color : The foreground color of parameter
00 1F: B_Color: The background color of parameter
00 01 E2 3A: From the above table, Parameter: 0x00 01 E2 3A
AA C1 08 00 00 01 CC 33 C3 3C—Display Multi-parameters with Temporary Buffer
00 00 01：<Parameter_N> : The number of multi-parameters



Example1: Two data is displayed on the one screen. The first data is 81.2; the second data is 7.3.
AA C0 00 00 61 44 00 0A 00 0A 00 00 FF FF 00 00 03 2C 61 44 01 AF 00 40 00 00 FF FF 00 00 00 49 CC 33 C3 3C AA C1 08 00 00 02 CC 33 C3 3C AA



3.18.11 Use Temporary Buffer to Buffer Commands to Achieve Simultaneously Display (0xC110)

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

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Frame_Number>: The number of specification frames to continuously display, 0x01-0xFF

Command frames of temporary buffer frame are defined as follows:

The starting address	Definition	Frame structure
<Address>	The first specification frame	Frame length + specification + data Example: Display the number zero picture, defined as 03 70 00 00 ***Note: Because the temporary buffer data is stored in the form of words, the number of bytes written must be an even number.***
<Address>+0x80	The second specification frame	
……		
<Address>+0x80×(K-1)	The Kth specification frame	

Example:
Write commands in the buffer:
AA C0 00 00 09 55 00 00 00 00 30 31 32 33 CC 33 C3 3C
AA C0 00 80 09 53 00 64 00 64 34 35 36 37 CC 33 C3 3C
Synchronous display:
AA C1 10 00 00 02 CC 33 C3 3C
Description:
Since temporary buffer store data as word, a command frame address is 0x80. But the actual storage space length is 256 bytes.

This command is mainly used to solve the problem that many parameters are needed to refresh and displayed simultaneously in order to avoid this case which. Parameters are not asynchronous in the process of refreshing by the communication delay (particularly low baud rate).

3.19 Keyboard Operation Command (0x71, 0xE5)

3.19.1 Key Code Upload (0x71)

Tx: None

Rx: AA 71 <K_code> CC 33 C3 3C

- <K_code>: Users preset key codes. Key code will be automatically uploaded with 5 keys per second when the keyboard of UART LCM interface scans the pressing action.

3.19.2 Key Code Setting (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.

3.20 Read and Write in User's Memory (0x90, 0x91)

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

3.20.1 Write in Random Data Memory (0x90 64KB)

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

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

- <Address>: The starting address (Word address) of temporary buffer (RAM); totally 40k word, from 0x0000 to 0x9FFF.
- <Data0……Datai>: Data series to be written in form of bytes.

3.20.2 Write in Sequential Data Memory (0x90 30MB)

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

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

- <Address>: The starting address (word address) of temporary buffer (RAM) is totally 40k word, from 0x0000 to 0x9FFF.
- <Data0……Datai>: Data series to be in form of bytes.

Different from random data memory, sequential data memory can only write sequentially. It cannot write randomly.

The whole sequential memory is divided into 239 of 128KB data page. When encounter to the head of page (address=* *******02 00 00), it will automatically erase the current page with no data backup before erasing and will not affect other pages. It is recommended to do some continuous data of large amount storage such as paperless recording and audio recording.

3.20.3 Read Data Memory (0x91)

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

Rx: AA 91 <Address> <Length> <Data0……Datai> CC 33 C3 3C

- <Address>: The starting address to write in data memory has 4 bytes from 0x00000000 to 0x01DDFFFF.

The memory address of random data ranges from 0x01 DE 00 00 to 0x01 DE FF FF

- <Length>: The length of data memory to read, 2 bytes. Most read 64KB once.
- <Data0……Datai>: The read data strings in the form of bytes.

3.21 Download Font and Configuration File Command (0xF2)

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

Rx: Please view Tx Text_Lib!

Then users can download corresponding word font.

After finishing saving font, UART LCM will response: ******One Text_Lib Saved OK! ******

- <Lib_ID>: The storage position of font. A total of 60 fonts (0x00-0x3B), which 0x00-0x1F are 32 small fonts (configured files) in 128KB and 0x20-0x3B are 28 large fonts in 1MB.

Unless users need to design their own Chinese font, please do not modify font that located at Lib_ID=0x00, 0x20, 0x21, 0x22 and0x23; otherwise, 0x53, 0x54, 0x55, 0x6E and 0x6F command will wrongly display.

3.22 Simple Algorithm Command (0xB0)

3.22.1 Pinyin Input Method (0xB001、0xB004)

Tx: AA B0 01 <PY_Code> CC 33 C3 3C or AA B0 04 <PY_Code> CC 33 C3 3C

Rx: AA B0 01 <HZ_Num> <HZ_String> CC 33 C3 3C or AA B0 04 <HZ_Num> <HZ_String> CC 33 C3 3C

- <PY_Code>: Chinese pinyin, capital display with maximum 6 bytes
- <HZ_Num>: The number of Chinese character of pinyin. 0 represents the Pinyin is error. B001 specification is 1 byte, B004 specification is 2 bytes.
- <HZ_String>: All Chinese characters of pinyin using internal code

0xB001 is for the first class of GB2312-80 first font. 0xB004 is for the GBK expansion font.

3.22.2 MAC Calculation (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>: to calculate (A×B＋C)/D while A,B,C and D are all unsigned integral of 2 bytes
- <E,F> is the result of calculation. E is 4 bytes quotient and F is 2 bytes remainder

3.22.3 Array Sort (0xB003)

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

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

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

3.23 Buzzer Control Command (0x79)

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

Rx: None

- <On _Time>: The time when turn on the buzzer , 0x01-0xEF. The unit is 10ms.

The specific time when turn on the buzzer.

3.24 Display and Read Clock Command (RTC) (0x9B, 0xE7)

3.24.1 Close clock display

Tx: AA 9B 00 CC 33 C3 3C

Rx: None

3.24.2 Open clock display

Tx: AA 9B FF <RTC_Mode> <Text_Mode> <Color> <X> <Y> CC 33 C3 3C

Rx: None

- <RTC_Mode>: The clock mode
  - 0x00：HH:MM:SS
  - 0x01: 20YY-MM-DD HH:MM:SS
- <Text_Mode>: The foreground type of clock
  - 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>: The characters' color
- <X> <Y>: The coordinates of clock

3.24.3 Clock Setting

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>: Time format to be set as year:month:day:hour:minute:second, by BCD code.

Example:
AA E7 55 AA 5A A5 08 11 28 12 57 00 CC 33 C3 3C
Set current time to 2008:11:28:12:57:00.

3.24.4 Read Current Time (The Gregorian calendar)

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>: The current clock data

Example:
08 12 25 04 09 58 00, represents 2008:12:25:Thursday:09:58:00.
If clock function is not supported by UART LCM, the unknown results will be returned.

3.24.5 Read Current Time (The Luna calendar)

Tx: AA 9B 5B CC 33 C3 3C
Rx: AA 9B 5B <YY:MM:DD:Chinese Zodiac:Heavenly stem: earthly branches> CC 33 C3 3C

- <YY:MM:DD:Chinese Zodiac:Heavenly stem: earthly branches> is current Luna calendar, including Chinese Zodiac, Heavenly stem, earthly branches, represented by internal code.

Example:
09 02 03 "niu yi chou", represents current lunar calendar Feb. 3rd, 2009, in which 2009 is "niu" (Cattle) year and chronogram is "yichou"
If clock function is not supported by UART LCM, it will be returned unknown results.

3.25 Play Music Command (The hardware suppose is required in 0x30, 0x32, 0x33 commands)

These commands need corresponding hardware (DWIN DMA5601 stereo sound recording and playing module) support.

3.25.1 Play specific position music (0x30)

Tx: AA 30 <Start_SEG> <SEG_Number> <Play_Time> CC 33 C3 3C
Rx: AA 3F 4F 4B start to play
AA 3F 4F 4B end playing

- <Start_SEG>: The starting address of music segments to be played from 0x00 to 0xFF；
- <SEG_Number> : The number of music segments to be played from 0x00 to 0xFF；
- <Play_Time>: Replaying times from 0x00 to 0xFF.

3.25.2 Volume adjustment (0x32)

Tx: AA 32 <Volume_L > <Volume_R> 00 CC 33 C3 3C
Rx: AA 3F 4F 4B

- <Volume_L>: The Volume of the left channel from 0x00 to 0x3F.
- <Volume_R>: The Volume of the right channel from 0x00 to 0x3F.

3.25.3 Stop playing (0x33)

Tx: AA 33 55 AA 5A CC 33 C3 3C
Rx: AA 3F 4F 4B

3.26 The Use of Configuration File Command (Touch interface, Keyboard interface, Animation and Icon library)

DWIN UART LCM achieves a simple OS function through the following configuration file in order to greatly reduce the workload of codes.

3.26.1 Touching Function Interface Switch Automatically (0x1E and 0x1A configuration files)

In order to reduce the amount of users' codes, DWIN UART LCM with touch screen can pre-download the configuration file to UART LCM and set touch interface to automatically switch mode to achieve the "non-intervention" of touch interface.

Developing process:

Step 1: Design UART LCM physical resolution of the same user interface, and download to UART LCM.

Example:

DMT64480S057_11WT means that interface resolution is designed to 640 x 480 point matrix.

Step 2: Generate configuration file

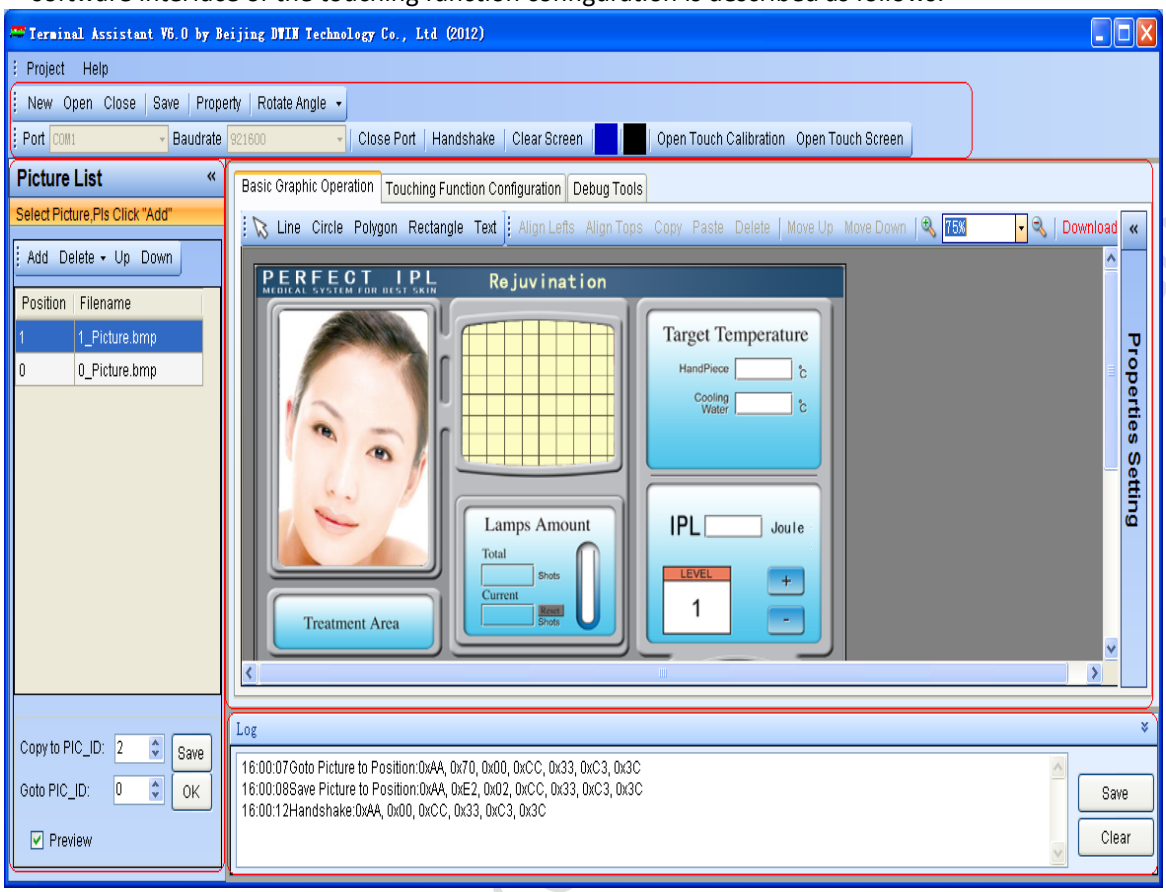
Configuration file is a binary file with maximum 8192 touch commands. Each command takes up 16 bytes, defined as table 3-26-1:

Starting address	Length of data （Byte）	Definition	Description
0x00	2	Pic_Now	the picture number of currently displayed screen; If Pic_Now of the high byte is 0xFF, touch command will end.
0x02	4	x_s,y_s	the left-up coordinates of effectively touching function area
0x06	4	x_e,y_e	the right-down coordinates of effectively touch area
0x0A	2	Pic_Next	the picture number of the next screen after clicking the effective touch area; If the high byte of Pic_Next is 0xFF, interface will not be switched.
0x0C	2	Pic_Cut	the number of touching function animated picture; If the high byte of Pic_Cut is 0xFF, there is not the touching function animated pictures.
0x0E	2	Touch_Code	After clicking the effective touching function area, touching function code will be uploaded(as a trigger message of the user software); If high byte of Touch_Code is 0xFF, touch key will not uploaded. If high byte of Touch_Code is 00xF0-0xF3, upload the data string index to the 0x1A configuration file. At this moment, (Touch_Code -0xF000) is the index ID. In 0x1A configuration file, the length of each index is fixed at 128 bytes and the first byte is the effective length of the index article.

Table 3-26-1 the definitions of touching function interface configuration file

Actually users arrange the process of the interface switching. PC tools software (integrated in DWIN screen debugging assistant position in debugging assistant, peripherals, touch screen and touching function interface configuration) can be easily implemented during the process of designing the interface during the process of generating touching function configuration file. The

software interface of the touching function configuration is described as follows:



The DWIN touch interface configuration software interface specification

Step 3: Download the configuration file to UART LCM terminal.

Use 0xF2 font to download command, and download the generated binary configuration file to 0x1E font in UART LCM terminal.

Step 4: Set UART LCM to touching interface auto-switch mode

Use the command 0xE0 to set the fifth bit (Para1.5, 0x20) to 1. When users press the screen, UART LCM terminal will automatically switch touching function interface and upload pre-defined key code instead of uploading coordinates.

The definition of 0x1A configuration file Configuration file 0x1A consists of maximum 1024 commands with a maximum of 127 bytes per command. It occupies 128 bytes in memory space. A single command is defined as follows.			
Starting Address	Length of Data（Byte）	Definition	Description
0x00	1	Length_Command	the length of command If Length_Command=0x00, the command is the end of a combination command.
0x01	indefinite	Command	If Length_Command is not 0x00, 0x01 is the starting command to send.
		Command pointer （Cmd_EN,Cmd_ID,Tx_Delay）	If Length_Command=0x00, other commands are the pointer of combination. Each pointer takes up 4 bytes, at most the 31 command pointers (Combination command). The pointer definition is listed as follows:

			Cmd_EN: 0x00 represents the end of combination command sending; and others represent the combination command sending. Cmd_ID: The ID of commands to be sent from 0x000 to 0x3FF; corresponds to the non-combination command numbers in 0x1A; It does not support nested of combination command. Tx_Delay: The sending interval of each commands; The unit is 0.1 second. ***When sending combination commands, UART LCM does not responded by users' commands and peripherals operation.***

Not only greatly reduce the amount of secondary development codes, but also reduce the difficulty development by the configuration file to design the touching function interface. More importantly, we hope to be able to change the design and marketing ideas about our products.

- **Completely separate products' algorithm and design interface**. Algorithm is the core competitiveness of an enterprise. So the precious research and development resources should not waste on a large amount of copious interface code designing.
- **Products research can process in parallel**. Not only the interface designing and algorithm can process in parallel, several graphic designers can design interface at the same time. The key code of touching function has a function of "trigger message". So engineers who are responsible for the different part of the algorithm can design and debug algorithm in parallel.
- **Improve the reliability of products**. In principle, all users' programmers lie in the same parallel level. So the function modules are independent and easy for testing.
- **Easily upgrade**. The upgrade of products are basically the upgrade of "interface" while "algorithm" needs little after products are stable.(Our products are very stable. So the upgrades of products are basically the upgrade of "interface" while "algorithm" needs little.)
- It is easy to achieve the interface of **customization or style** (Many skins), using the configuration file method, because users only need to insert different interfaces in configuration file. But the key code uploaded is the same.
- **Greatly shorten the developing time of new products which improve market competitiveness**. Develops a product for customers as follows:

a. Spend 1 to 3 working days to confirm basic demands with customers.

b. Spend 1 to 2 working days, our graphic designers can provide the product's interface based on the standard UART LCM. Meanwhile our structure engineers are ready to provide product's three-dimensional pictures for customers to confirm.

c. Our hardware engineers start to design the PCB after confirmation. The structure engineers submit the drawings to factories in Shenzhen for fast molding. The graphic designers begin to modify the interfaces and submit the process documents to software engineers.

d. One week later, we can submit a full-fledged prototype for inspection to customers.

3.26.2 Keyboard Control Interface Auto-switch (0x1B configuration file)

The keyboard control interface is also triggered through buttons, similarly with the touching function interface. The configuration files are saved in the 0x1B configuration file. The 0x1B configured file is consisted of maximum 5957 keyboard control commands. Each keyboard control command contains 22 bytes. The definition is listed in table 3-26-2:

Starting of Address	Length of Data（Byte）	Definition	Description
0x00	2	Pic_Now	The ID of picture display on the current screen. If the high byte of Pic_Now is 0xFF, it means the end of keyboard command.
0x02	2	0x00:Key_Code	Key code
0x04	2	Pic_Next	Switch the picture ID to the next interface; If the high byte of Pic_Next is 0xFE, it does not switch interfaces. If the high of byte Pic_Next is 0xFE, it switches the picture area.
0x06	14	Pic_Cut,Xs,Ys,Xe,Ye,X,Y	The high byte of Pic_next＝0xFE: Definitions of picture area are cutting area; The high byte of Pic_next=other is undefined
0x14	2	Touch_Code	The upload of fter pressing (As the message to trigger user software); if high byte of Touch_Code is 0xFF, it means no code is uploaded. If high byte of Touch_Code is 00xF0-0xF3, the data string of upload has indexed to the configuration file 0x1A (Touch_Code — 0xF000) is the index ID. In 0x1A configuration file, each length of index is fixed to 128 bytes and the first byte is the effective length of this command index.

Table 3-26-2 Definitions in configuration file of key interface

3.26.3 Automatically Loop command group (0x1C configuration file)

Composed of 16 groups specification set, each specification group is 8KB, including up to 64 UART LCM specifications, each specification occupies 128 bytes storage space. Each UART LCM specification format is defined as follows:

Starting of Address	Definition	Description
0x00	Command_Delay	Delay time after executing commands, the unit is 8ms. 0x00 means it hasn't delay time
0x01	Command_Length	Length of current commands: 0x00: This command is invalid. Others: commands length (Calculate from 0x20)
0x02-0x7F	Command	Standard UART LCM without frame header (0xAA) and frame end (0xCC 33 C3 3C)

Example:

Three pictures are looped and switched. The configuration file is listed as follows:

ORG 0000H ; the first group, PADIO=00H

```
DB 125, 2,70H, 00H      ; display picture 0
ORG 0080H
DB 125, 2,70H, 01H      ; display picture 1
ORG 0100H
DB 125, 2,70H, 02H      ; display picture 2
ORG 0200H               ; the second group, PADIO=01H
DB 125, 2,70H, 03H      ; display picture 3
ORG 0280H
DB 125, 2,70H, 04H      ; display picture 4
```

Use the 0x9A command to call：

```
AA 9A 00 CC 33 C3 3C      ; picture 0, 1, 2 loop switching
AA 9A 01 CC 33 C3 3C      ; picture 3, 4 loop switching
```

Description:

1. Each command is fixed to account for 128-byte memory space. Therefore the first address of each command is 0000H、0080H、0100H…
2. Each command group is fixed to account for 8-byte memory space. Therefore the first address of each command is 0000H、0200H、0400H…

3.26.4 Icon Display (0x1D configuration file)

DWIN UART LCM has one picture cut command 0x71, which cut an area of picture saved in UART LCM and paste to the specific location in the current interface (Details can be found in section 3.13.2). However this is not convenient for customers in actual use, we add the command 0x99 and Icon definition library file 0x1D to call icon by text method for customers.

Format of command 0x99 is described as table 3-26-3:

Command	Data	Description
0x99	$(x,y,Icon\_ID)_0$+……+$(x,y,Icon\_ID)_n$	(x,y) is the left-up coordinates of target place to display icons; Icon_ID is the index ID in icon library.

Table 3-26-3 The 0x99 format

The icon library configuration file is up to 13107(corresponding Icon_ID=0x0000-0x3332) Icons in the definition of a binary file. The definition of each icon contains 10 bytes, as defined in Table3-26-4:

The Starting Address	Length of Data (Byte)	Definition	Description
0x00	2	Pic_ID	Picture ID of saving Icon
0x02	4	x_s,y_s	The left-up coordinates of the icon area
0x06	4	x_e,y_e	The right-down coordinates of the icon area

Table 3-26-4 The definition of icons

Example

```
ORG 0000H
DW 10,0,0,100,100 ；  Icon_ID is 0000H
DW 11,0,0,100,100;    Icon_ID is 0001H
DW 12,0,0,100,100;    Icon_ID is 0002H
```

Use the 99 command：

AA 99 00 00 00 00 00 00 00 64 00 64 00 01 00 00 00 64 00 02 CC 33 C3 3C

Display icon on (0, 0), (100, 100) and (0, 100)

When receive the command 0x99 to DWIN UART LCM, it will follow these steps:

- This command remove the Pic_ID, (Xs,Ys) and (Xe,Ye) from position Icon_ID×10 in 0x1D configured file according to Icon_ID.
- This command cuts a picture similarly as 0x71 command: 0x71 Pic_ID, (Xs,Ys), (Xe,Ye) (X,Y)
- It process the next icon.

Use the 0x99 command with the configuration file in order to easily address the following issues:

- To analog dial, we can make different scales into icons and **directly, expediently** call it according to the variables instead of viewing table again.
- To **special characters**, we can **call it in the way of icon**, which eliminate character library trouble even if the UNICODE encoding does not contained,.
- **It is convenient to use some interface "elements" of Windows, such as cursor, mouse pointer and so on**

.

Recommend to use the ICON command 0x97 for using the DWIN screen K600+ kernel customers.

3.27 Switching Between UART LCM and Video (0x74. The hardware support is required)

For analog video playing:

Tx: AA 7A <Work_Mode> <Video_Mode> <Video_CH> CC 33 C3 3C

Rx: None

- <Work_Mode>: 0x00=UART LCM 0x01=Video
- <Video_Mode>: 0x00=PAL 0x01=NTSC
- <Video_CH>: 0x00=CVBS input video signal, 0x01=S interface input video signal.

For digital video playing based on SD card:

Tx: AA 7A <Work_Mode> <KEY_VALUE> CC 33 C3 3C

Rx: None

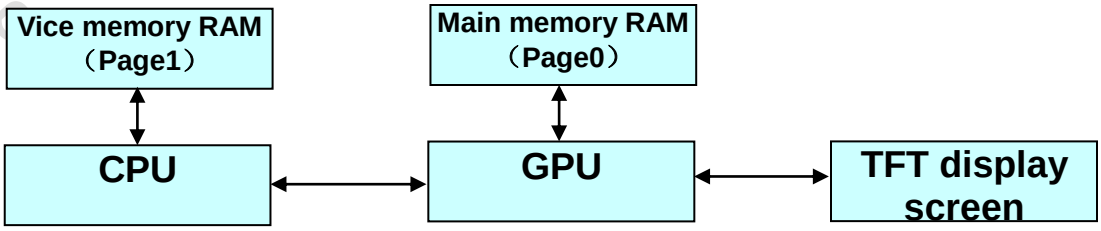
- <Work_Mode>: 0x00=UART LCM 0x01=Video
- <KEY_VALUE>: in video broadcast mode, the function key value is defined as follows:

KEY_VALUE	Function Key Value Description
0x00	lift
0x01	Play/pause press
0x02	Voice switch on
0x03	Voice switch off
0x04	Menu operation：Enter press
0x05	Menu operation：Backward press
0x06	Menu operation：Forward press
0x07	Menu operation：Esc press
0x08	Menu operation：volume up press
0x09	Menu operation：volume down press
0x0A	Menu operation：Chinese/English select press

3.28 Refresh A Full Screen Display Compulsively once (0XD0)

Tx: AA D0 CC 33 C3 3C

Rx: None



For the intrinsic software above version V5.5 of K600, H600, K600+ kernel the dual-memory structure has been applied in order to avoid the following problems:

- It will cause flickering or jittering when using the background restoration (Such as 0x71 + 0x98 commands to display text without background color).
- When the large numbers of parameters are displayed on a page, it will cause asynchrony display (The lower the baud rate is, the more obvious the problem will be).

When using the dual-memory structure, the interface that users can see is main memory (Page 0), while the interface of instruction operation is vice memory (Page 1). The CPU timing (100ms) can refresh the display content from vice memory to main memory in 100ms in order to synchronously display. But there are two exceptions:

a. CPU will automatically and immediately refresh the display content from vice memory to main memory after displaying pictures w the 0x70 command.

b. When CPU is processing the command, which relates to display, it will automatically delay the refresh cycle extended to 30ms even if it's time to display. If the display commands are not received any more after 30ms, it will be refreshed and displayed.

It will immediately refresh content from vice memory to main memory when the function of 0xD0 command is regardless of refresh on time,.

Recommend that new customers select "0xD0 command not auto-execution mode" and send the 0xD0 command to display once after completing the process. You can get very good synchronous display results and achieve the jitter-free display.

Refresh mode configuration table (PARA2 parameters specification please refer to 0xE0 command)		
mode	PARA2.2	PARA2.7
Automatically execute 0xD0 command to refresh in every 30ms	1	0
Automatically execute 0xD0 command to refresh in every 200ms	1	1
The 0xD0 command does not executed automatically except that 0x70 and touching function refresh. Only refresh when users are sending the 0xD0 command.	0	Any value

3.29 Input Parameters or Text with Touch Screen Command (0x7C, *H600 and K600+ only)*

Touch screen input parameters. The following conditions must be met:

1. Use the 0x1E touching function configuration file.
2. Low byte key code must be valid and in accordance with the following specifications:

Key code (Hex mode)	Definition	Description
0xF0	ESC	Exit
0xF1	Enter	Confirm
0xF2	BackSpace	Backspace
0xF3	Page_up	Pageup
0xF4	Page_down	Pagedown
0xF5	CapsLock	Caps lock, limited only A-Z (a-z), the button is highlighted if effective. If preset key code is A-Z: after clicking CapsLock will be a-z; If the preset key code a-z: after clicking CapsLock will be A-Z。 If use this button, you cannot design animation in configuration file.
0xFD	Return	return

3. The background of textbox which used to input display must be a solid color.
4. DWIN_PY_GBK_01.BIN ***must*** be placed in the 0x01 font the DWIN GBK input lexicon (DWIN_PY_01.BIN) when Chinese input.

3.29.1 Input Pure ASCII String (0x7C01)

Tx: AA 7C 01 <R_ID> <VP_ID> X Y <Str_Max_Num> < Str_Scale> < Str_Color > <[Init_V]> CC 33 C3 3C
Rx: AA 7C 01 <R_ID> <Str_Number> <String> CC 33 C3 3C

- <R_ID>: A two-byte returned default value (0x0000-0xFFFF), which is used by user software for returned parameter identification
- <VP_ID>: The virtual page number used for input method, VP_ID=0xFFFF represents by the current page.
- <X, Y>: The left-up positon of input text display area. Since the text display background color is read from address (X, Y), the actual display address is start at from (X+1, Y), by the right alignment displayed format.
- <Str_Max_Num>: The maximum length of input string, 0x01-0x40, up to 64 ASCII characters.
- < Str_Scale>: The display format of input string.

  <Str_Scale.7>: 0=Display characters normally; 1=Display text masked with "*", by display of input password and so on.

  <Str_Scale.6>: 0=input no initial value; 1=the input initial value.

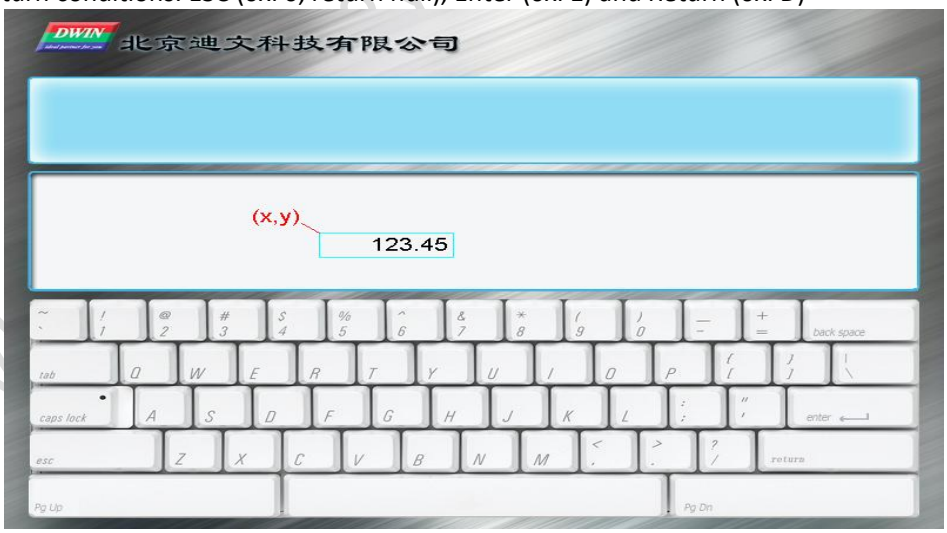
  <Str_Scale.5-.4>: Reservation and set to 0.

  <Str_Scale.3-.0>: Size of input text, 0x00-0x07, up to 8 fonts.

Str_Scale	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Font size	8*8	6*12	8*16	12*24	16*32	20*40	24*48	28*56

- < Str_Color >: The display color of input string by 65K color mode.
- <[Init_V]>: The Initial value in ASCII encoding. To set this value, <Str_Scale.6> have to be set to 1.
- <Str_Number>: The number of actually returned ASCII characters, 0x00 represents null value.
- <String>: The returned ASCII.

End of return conditions: ESC (0xF0, return null), Enter (0xF1) and Return (0xFD)



3.29.2 Input Mixed String both Chinese and English (0x7C02)

Tx: AA 7C 02 R_ID VP_ID X Y Str_Max_Num Str_Scale Str_Color T_Color Tx Ty [Init_V] CC 33 C3 3C
Rx: AA 7C 02 R_ID String_Number String CC 33 C3 3C

- <R_ID>: A two-byte returned default value is 0x0000-0xFFFF, which is used with user software to return the parameter identification
- <VP_ID>: The virtual touching function page ID which used for input method, VP_ID=0xFFFF

represents to use the current page.

- <X, Y>: The left-up address of input text display area. Since the text display background color reads from address (X, Y), the actual display address is start from (X+1, Y), by the right alignment displayed format.
- <Str_Max_Num>: The maximum length of input string is 0x01-0x40, at most 64 ASCII characters or 32 Chinese characters.
- <Str_Scale>: The format for input strings.
  - <Str_Scale.7>: Reserved, written 0.
  - <Str_Scale.6>: 0=input no initial value; 1=the input initial value.
  - <Str_Scale.5-.0>: Reservation and set to 0.

ASCII characters display in 8*16 dot matrix and Chinese display in 16*16 dot matrix in mixed input mode (Chinese & English).

- < Str_Color >: The display color of input string with 65K color mode.
- < T_Color >: The display color of Chinese list with 65K color mode.
- < Tx, Ty >: The left-up address of Chinese list display area. The text background color is read from address (Tx, Ty). So the actual display address is start from (Tx+1, Ty), Chinese list by fixed 16*16 dot matrix, 8 point for each column. In order to facilitate the touch selection, actually each character occupies 40*40 display space.
- <[Init_V]> : The initial value with ASCII encoding. <Str_Scale.6> have to be set to 1 when set ting the initial value.
- <Str_Number>: The number of actually returned bytes.
- <String> returns to ASCII characters.

In this input method, Enter (0xF1) will directly input ASCII character. The special characters "BD" will select the SBC case punctuation input.

The end of return conditions: ESC (0xF0, return null), Return (0xFD)

3.29.3 Quit Text Input Method Compulsively (0x7C 00)

TX: AA 7C 00 CC 33 C3 3C

RX: None

Attention:

- ***UART LCM will not feedback data to users until the input is complete during with the 0x7C command. But UART LCM can execute other commands which are not related to touch screen during the touch screen input.***

- ***If the virtual page is not current page, then the button animation effect cannot be set.***

4 Upgrade Method of UART LCM Software

Tools:

- A DC regulated power supply; The output voltage adjusts the suitable value;
- A serial port line;
- A computer has hardware serial port and serial debugging assistant SSCOM3.2 software.

The Steps of Upgrade:

a. The UART LCM is ***powered off*** and serial port is connected to the computer's COM1.

b. Open the SSCOM3.2 software and click "Open File" to ***select UART LCM's program***, such as M600_V40.BIN.

c. Write ***"DWIN_M600_BOOT!"*** on send column and set ***auto-send*** interval to 10

d. Hook on ***"send new line"*** and ***"auto-send"***, then UART LCM power on.

e. The serial port will receive "Erase .....". If the serial port does not receive "Erase.....", serial port is not connected or UART LCM is damaged. Please check.

f. It will receive "Please Tx File!" about one second, then hook off "auto-send" and click "send file".

g. Wait another 3 to 10 seconds. If the serial port receives "**********END**********", it means download is finished.

h. UART LCM is ***Power-off*** and the upgrade of software is successful.

The serial port interface before UART LCM power-on

5 Download Specification of SD Card Interface

Processing steps:

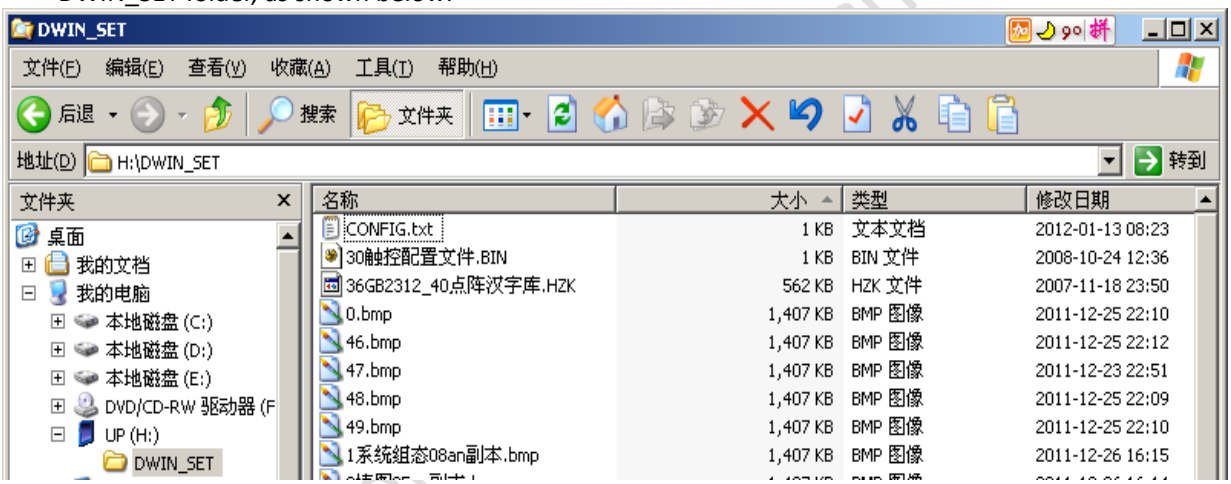
- Powering off the DWIN UART LCM and inserting the SD card(SD card hot-swappable is valid for K600+ kernel)；
- Powering on the DWIN UART LCM again, the DWIN UART LCM will automatically load the content of the DWIN_SET folder in SD card, and save it to the FLASH which is placed in screen.
- After updating the DWIN UART LCM, it will automatically reset and reboot to enter the normal work mode. At this moment, the user can remove the SD card.

Note:

During the process of download, the display may be abnormal. The backlight may be blinking or accompanied with the sound of "Zizi". But this is a normal phenomenon and does not impact the DWIN screen hardware and software performance.

File Format Specifications

- To establish ***DWIN_SET folder*** in the ***SD card root directory***；
- The pictures, fonts and configuration files which need to be downloaded are put in the DWIN_SET folder, as shown below:



SD card file format Description			
File Type	Naming Rules	For example	Description
Picture file	The location of picture storage+ (optional) file name.BMP	00 boot interface .BMP	Must be the same as DWIN screen resolution 24-bit color BMP files
Font file	The location of font storage+ (optional) file name.BIN/HZK/DZK/ICO	32_GBK12 Chinese font.DZK; 30 touch file.BIN; 0DWINASC.HZK; 038 icon library.ICO	
Configuration file	CONFIG.TXT		

The CONFIG.TXT configuration files specifications:

CONFIG.TXT file is used to configure the DWIN screen operation mode (corresponding to the 0XE0 command) and backlight mode (corresponding to the 0x5E command).

CONFIG.TXT file of similar scripting language used to describe the parameter registers. Each line describe has one parameter. The unused parameters cannot write, as follows:

Parameter register name	Range	Description
R0	Depends on DWIN screen	The DWIN screen driver mode selection, users don not configure it
R1	0x00-0x11	Set baud rate, 0x00-0x10 corresponds to 1200bps—921600bps
R2	0x00-0xFF	0xE0 command of PARA1
R3	0x00-0xFF	0xE0 command of PARA2
R5	Uncertain	K600+ kernel only. When R1=0x11, baud rate configures the high bit of a word. R5:R9=6250000/users can define the baud rate. e.g.: set 10000bps baud rate, R5:R9=6250000/10000=625=0x0271 R5=02 R9=71
R6	0x00-0x40	After starting the backlight with touch screen control, click on the touch screen, the backlight light brightness
R7	0x00-0x40	After starting the backlight with touch screen control, a period of time do not click on the touch screen, the backlight light brightness
R8	0x01-0xFF	After starting the backlight with touch screen control, the time of touch screen backlight lit, unit is 0.5 seconds
R9	Uncertain	K600+ kernel only. When R1=0x11, baud rate configures the low bit of a word.
The parameter of configuration file is one byte Hex mode, such as 0A represents decimal 10; the parameter of configuration file must be 2 bits, such as 00 cannot be written as 0.		

The typical configuration file

R1=07 ; baud rate, 07=115200bps

R2=30 ; PARA1, 30=enable touching function, the backlight brightness switch is controlled with clicking touch screen

R6=40 ; on the touch screen backlight automatically control mode, the brightness of backlight lit, 40=100% brightness (the brightest)

R7=10 ; on the touch screen backlight automatically control mode, the brightness of backlight turns off, 10=25% brightness

R8=14 ; The time of backlight, the unit is 0.5 seconds. 14=10 seconds

6 C51 Serial Communication Program Reference

```
#define uchar unsigned char
#define uint    unsigned int
bit RXAAOK;        // The 0xAA header is received by serial port
bit RXFRMOK;       // A valid data frame is received by serial port
bit TI0FLAG;       // Send a TI sign back to the main program’s serial port
uchar RXBUF[32];   // Buffer is received and saved by serial port. But the buffer must be remove
                   //0XAA frame head and the UART LCM response of un-fixed format has the
                   //length return. So it needs to record length
uchar Rx_P;    // Data position is received and saved by serial port

//The program of serial port interrupt
void UART0_ISR() interrupt 4
{ uchar i;
    if(RI)         //The Interrupt is received by serial port
{     i=SBUF;
      RI=0;
      if(RXFRMOK==0)    //if RXFRMOK=1 means that the current data is not be processed in
                        //foreground and remove
  {    if(RXAAOK)          //received 0xAA
  {    RXBUF[Rx_P]=i;
if((Rx_P>3)&&(RXBUF[Rx_P-3]==0xCC)&&(RXBUF[Rx_P-2]==0x33)&&(RXBUF[Rx_P-1]==0xC3)&&(RXB
UF[RX_P]==0x3C))
    {RXFRMOK=1;
      RXAAOK=0;}
      Rx_P++;}
      if(!RXAAOK&&(i==0xaa))
  {    RXAAOK=1;
      Rx_P=0;}}
}
   if(TI)         // The interrupt is sent by serial port
{   TI=0;
    TI0FLAG=1;}
   // The FSTA register should be cleared 0 to the CPU of taking FRAME_ERROR interrupt in DSP and
ARM

//Application example 1: The main program query the state of touch screen
      if(RXFRMOK)                  // Check it whether the UART LCM data is received in foreground
                                   //main program
{     if(RXBUF[0]==0x78)      //Touching function button
      {}
      if(RXBUF[0]==0x7C)      //Touch screen Pinyin input is end
      {}
      RXFRMOK=0;                  //Finish and clear messages
  }
//Application example    2：Power is on and query whether the serial port screen is ready
void acklcd()              // Send the 0x00 handshake command check whether LCD is ready, or waiting
                              //LCD ready when power on
{   RXFRMOK=0;
```

```
    while(RXFRMOK==0)
{   txword(0xaa00);
    txeof();
    delay10ms(10);}
    RXFRMOK=0;}

void txbyte(uchar i)        // A byte is sent to serial port by the query mode
 {  SBUF=i;
    whilie(TI0FLG==0);
    TI0FLG=0;}

void txword(uint i)        //Send a byte to serial port
{   txbyte(i/256);
    txbyte(i%256);}

void txeof()                //Send terminator frames to serial port
{   txword(0xcc33);
    txword(0xc33c);}
```

7 Revision History

Revise Date	Contents	Kernel Version	Document Version
2008.12.01	First Publish.	--	Ver1.0
2008.12.09	Revise specifications of configuration files and add specifications about keyboard interface.	--	Ver1.1
2008.12.11	Add several special fonts in 0x98 command	--	Ver1.1
2008.12.15	Add the 0xC104 command. Revise the 0xC103 command and add the function of amplifying Y-axis in scale.	--	Ver1.1
2008.12.25	Unify command set of T and S serials. Add the 0x9B command to read clock.	--	Ver1.2
2009.01.19	Add the command 0x5E to the backlight of touching function or keyboard control function. Add the command 0xE0 to the definition of PARA1.4 which controls to open backlight.	--	Ver1.3
2009.02.02	Add the 0x98 command on the lunar calendar support	--	Ver1.3
2009.02.12	Add 0x1A configuration file which supports command to combination commands in touchpad operation.	--	Ver1.3
2009.03.13	Add the 0x64 command (Area filling).	--	Ver1.4
2009.04.07	Add the 0x5704 command (Display arc).	--	Ver1.4
2009.06.29	Add the 0x9C command to achieve transparent icon overlay of a solid color background. Add the 0xC106 command to supplement the 0xC105 command: Positive and negative two-way curves; window's limitation.	--	Ver1.5
2009.07.16	Modify the 0x9A and 0x1C commands configuration file specification. The M600 products can be automatically looped the command set at fixed time in order to design animation easily	--	Ver1.5
2009.07.23	Revise the definitions of the configuration file of 0x1A and 0x1E. So the user commands extended to 1024.	--	Ver1.5
2009.08.07	Add the 0xC110 command, added the function of controlling the buzzer by touching or software.	--	Ver1.5
2009.08.08	Modify 0x79, 0xE0 command, added the buzzer control and buzzer software control.	--	Ver1.6
2009.09.03	Add the 0x7A to switch the UART LCM model and the video model.	--	Ver1.6
2009.09.17	Add the 0xC2 command back and forth to read data of temporary buffer in order to provide the RAM function of the use RMA.	--	Ver1.6
2009.12.27	Add the 0x7B command to verify the users' pictures to ensure a successful download.	--	Ver1.6
2010.01.04	**K600 kernel production;** because of kernel modification, modify 0xE0 command <TFT_ID> corresponds to the product list; modify 3 calibration point position of the touch screen calibration detective (0xE4).	Ver4.5	Ver1.7
2010.02.21	Add the 0x98 command to the definition of C_Mode.5 in order to support the text vertical wrap.	Ver4.5	Ver1.7
For 128MB version of the UART LCM kernel, upgrade Ver5.0 or later UART LCM software needs to download fonts and pictures again!			
2010.03.06	Add the upload ID of the currently displayed pictures in the	Ver5.0	Ver1.8

	response of the 0x00 command; Add the 0xE0 command to define Para 1.2. Para 1.2 command integrate the 0°and deflection 90°kernel; **The serial port is no longer supported frame time-out mode and only support frame terminators.**		
2010.06.08	Add the definition of C_Mode.4 to support ASCII spacing automatically adjustment in the 0x98 command	Ver5.1	Ver1.8
2010.07.01	Add the 0x9D command, which is the same as 0x9C without auto-recovery of current background. Add the 0xD0 command which aimed for forcing refreshing. Add the 0xB004 command which is expand to GBK font	Ver5.2	Ver1.9
2010.07.15	Add the touching uploading 0x79 command with increasing Para1.1 and Para1.0 definition for touching-control pattern and buzzer sound only when click the effective touch area.	Ver5.3	Ver1.9
2010.08.02	Cancel the 0xE0 command to modify the function of TFT_ID to prevent users from setting error in order to lead to the display irregularities; Added 0xC107 pixel buffer operating command for the layer operation.	Ver5.3	Ver1.9
2010.08.09	Add the 0xC108 command for the convenience of multi-parameter display.	Ver5.3	Ver1.9
2010.08.23	Add the 0x78 command for quick incremental connection and add 64X64 font of the 0x98 command.	Ver5.3	Ver1.9
2010.10.20	Add the extra textbox limitation and cancel the command 0x45 to display the text paragraph in specific areas; Modify the touch screen calibration is the cross mark; Add two fonts 24x48, 28x56 in the 0x9B clock command.	Ver5.3	Ver2.0
2011.01.27	**H600 kernel production;** add the 0x7A command to support that UART LCM is played by videos based on SD card.	Ver5.5	Ver2.0
2011.05.11	Add the 0x73 command to support double color bitmap filling. The 0x98 command added function of supporting large dot matrix characters.	Ver5.5	Ver2.0
2011.07.01	Modify 0xE0: TFT_ID can't be modified, parameters are not saved after power-off, added background automatically restored; Added the high-speed (6.225 or 12Mbps) download function for pictures and fonts.	Ver6.0	Ver2.1
2011.07.15	Add the 0x7C touch screen input method command and users can input English and Chinese text or parameters by touch screen.	Ver6.1	Ver2.1
2011.08.01	Add the 0x9E Command to cut and rotate picture and make it easier to Stimulate the dashboard with the transparent pointer.	Ver6.1	Ver2.2
2011.10.18	Update the 0x7C command. The <Str_Scale> parameter uses to define has initial value or not.	Ver6.2	Ver2.2
2012.01.12	Add the SD card interface in H600 kernel to support for offline, high-speed character, pictures download and parameters configurations. Add the definition of the parameter configuration word Para2.2 to support user automatically refresh.	Ver6.3	Ver2.3

2012.01.30	**K600+ kernel production;** add the 0x97 ICON command to display icon and the WordArt in K600+.	Ver6.5	Ver2.3
2012.03.31	K600+ kernel: add the function that users can define the baud rate by the CONFIG.TXT (configure file) via SD card	Ver6.6	Ver2.4
2012.05.22	Add the 2-byte checksum verification (Optional, H600 and K600+ only) in the data frame structure.	Ver6.6	Ver2.4

*If you have any questions about this document or DWIN UART LCM, please feel free to send mail to our UART LCM product support engineers with the email address: **dwinhmi@263.net**.*
Or you can call: (86)10-62555113 or (86)10-62628965.
Welcome to visit our web site which contains DWIN UART LCM's latest documents: www.dwin.com.cn
Thank you for your support to DWIN, which motives us to progress.
Thanks a lot!