Technical Note

Wizee serial port screen instruction set

Items	Description
Keywords	Instruction set of configuration function serial port screen
Abstract	Wizee serial port screen instruction set includes configuration instruction set and basic instruction set

Customer Service

Elecrow

Email：info@elecrow.com tech@elecrow.com

Web：www.elecrow.com

Location：4F, Building A, Phase 2, Huafeng International Robot Industrial park,

Hangcheng Street , Bao'an Ave, Shenzhen, China

Directory

Customer Service 1
Elecrow 1
1. Instruction format and instruction set 6
1.1 Instruction format 6
1.1.1 Example of instruction operation 6
1.2 Color format 13
1.3 Difference between configuration instruction set and basic instruction set 13
1.4 How to quickly become familiar with the instruction set 13
1.5 Configure the instruction set table 14
1.6 Basic Directive Collection Table 38
2. Detailed description of the configuration instruction set 45
2.1 Handshake 45
2.2 Reset report 45
2.3 Reset the device 45
2.4 Get the device version 45
2.5 Backlight adjustment 45
2.6 Automatic screensaver mode 45
2.7 Buzzer control 46
2.8 Configure the touch screen 46
2.9 Touch screen calibration 47
2.10 Touchscreen experience 48
2.11 Set the baud rate 49
2.12 Switch screens 49
2.13 Use animation effects to switch screens 49
2.14 Reading the screen 50
2.15 Button control ID value uploaded 50
2.16 Set the button pop-up or pressed state 53
2.17 Internal instructions implement button mutex and status display 54
2.18 External Instruction Output 56
2.19 Read button control state 56
2.20 Update text control values 56
2.21 Clear text control content 59
2.22 Read the text control values 59
2.23 Set the cursor focus 59
2.24 Set text control flickering 59
2.25 Set the text control scroll speed 59
2.26 Set the background color of the document control 59
2.27 Set text background transparency 60
2.28 Set the text control foreground 60
2.29 Formatted text display 60
2.30 Text Control Numeric Increment Adjustment Instructions 61
2.31 Update the progress bar control values 61
2.32 Read the progress bar control value 62
2.33 Slider control upload format 62

2.34 Set the background color of the feed bar 63
2.35 Set the foreground color of the progress bar 63
2.36 Update slider control values 63
2.37 Read the slider control value 63
2.38 RTC clock settings 63
2.39 Read the RTC clock 64
2.40 Update gauge control values 64
2.41 Read the gauge control values 65
2.42 Animated control display 65
2.43 Icon control display 66
2.44 Read icon control values 67
2.45 Icon control value upload 68
2.46 Set the icon position 68
2.47 Bulk update control values 68
2.48 Curve control display 69
2.49 Set the drop-down menu to write a text control 71
2.50 Drop-down menu control value upload 73
2.51 Set the swipe selection control value 74
2.52 Read the sliding selection control value 74
2.53 Swipe to select control value upload 74
2.54 Update QR code controls 76
2.55 Start the timer 77
2.56 Set the timer 77
2.57 Start the timer 77
2.58 Pause the timer 77
2.59 Stop the timer 78
2.60 Read Timer 78
2.61 Manually disable/enable screen updates 78
2.62 Show/Hide Controls 78
2.63 RS485 screen address settings 78
2.64 RS485 screen address canceled 78
2.65 Lock the system configuration 79
2.66 Unlocking the System Configuration 79
2.67 Alarm event triggered 79
2.68 Alert event lifted 79
2.69 Data Record Control - Add General Records 80
2.70 Data Logging Control - Clear Record Data 80
2.71 Data Record Control - Sets the record display offset 80
2.72 Data Record Control - Gets the current number of records 80
2.73 Data Record Control - Reads a row of records 81
2.74 Data Record Control - Modify General Records 81
2.75 Data Record Control - Deletes a row of records 81
2.76 Data Record Control - Insert General Record 81
2.77 Data Record Control - Select a row of data records 81
2.78 Data Record Control - Add multiple records at once 82
2.79 Data Logging Control - SD Card Export Instructions (CSV format). 82

2.80 History curve setting sample values 82
2.81 History curve disable/enable sampling 82
2.82 History curve hides/shows channels 82
2.83 The historical curve sets the duration (i.e. the number of sample points). 83
2.84 The history curve zooms to full screen 83
2.85 Historical curve setting scale system 83
2.86 History curve setting value display range 83
2.87 Historical curve data is cleared 83
2.88 Rotate control sets the rotation angle 84
2.89 Multilingual switching instructions 84
2.90 Get the system language index 84
2.91 Video-related directives (applicable only to IoT type and M type). 84
2.91.1 Add Video Path 84
2.91.2 Play the video 85
2.91.3 Video Pause/Resume 85
2.91.4 Video stops 85
2.92 Audio playback related instructions (applicable only to IoT type, F series, M type).85
2.92.1 Audio playback 85
2.92.2 Stop audio playback 85
2.92.3 Audio playback instructions 85
2.92.4 Pause audio playback command 85
2.92.5 Resume audio playback command 85
2.92.6 Stop audio playback command 86
2.92.8 Volume adjustment 86
2.93 Matrix keyboard control 86
2.94 Write data to FLASH 87
2.95 Read data saved in FLASH 87
2.96 Save the current picture to the TF card 87
2.97 Save the wireless network settings 87
2.98 Get wireless network settings 87
2.99 Get Network Status 88
2.100 Search wireless networks 88
2.101 Save the network settings 88
2.102 Get the network settings 88
2.103 Save network services 88
2.104 Get network services 89
2.105 Send network data 89
2.106 Send network data (hexadecimal). 89
2.107 Save the network settings 89
2.108 Local/external AV input video 89
2.109 Displays the system keyboard 89
2.110 Hide System Keyboard 90
2.111 touch screen sensitivity adjustment command (only for capacitive screen). 90
2.112 enters standby 90
2.113 Enter the active state 90
2.114 goes into hibernation 90

2.115 Exit hibernation 90
2.116 Switch AV input signals 90
2.117 Switch sprite 91
2.118 The sprite window switches to the previous screen 91
2.119 The sprite window switches to the next screen 91
2.120 Enable Canvas 91
2.121 Basic drawing instructions 91
3. Appendix A details the basic instruction set 93
3.1 Set the front/back view 93
3.2 Automatically clear layers when switching screens 94
3.3 Set the spacing between text lines and columns 94
3.4 Set the text box 95
3.5 Set the image filter color 95
3.6 Text display 95
Figure 3-3 Text shows the rendering 96
3.7Cursor display 96
3.8 Full screen image display 96
3.9 Area picture display 97
3.10 Image Cut 97
3.11 Animated display 98
3.12 Foreground Drawing Points 99
3.13 Background color to draw dots (delete points). 99
3.14 Draw the line 100
3.15 Connect the equidistant X coordinates with the foreground color 100
3.16 Wire with foreground color according to coordinate offsets 100
3.17 Connect the specified coordinate points with a foreground color 101
3.18 Connect the specified coordinate points with a background color 101
3.19 Wire with background color according to coordinate offset 101
3.20 Draw hollow circles 102
3.21 Draw a solid circle 102
3.22 Draw a circular arc 102
3.23 Draw a hollow rectangle 103
3.24 Draw a solid rectangle/partial clearance 103
3.25 Draw a hollow ellipse 103
3.26 Draw a solid ellyt 104
3.27 Clear layers 104
3.28 Capture the current screen and save it in FLASH 104
3.29 Displays screenshots saved in FLASH 104
3.30 RTC mode settings 104
BIT0: Enable signal 104
0 : RTC off 1: RTC on 104

1. Instruction format and instruction set

1.1 Instruction format

A complete serial port instruction frame format is shown in Table 1.1. If the directive argument is greater than 1 byte , the high bytes are in the front, low bytes are in the back. The maximum length of the instruction is 1024 bytes, including the frame head and end of the frame, and the values are hexadecimal. Serial port format: 8-bit data bit, 1-bit stop bit, no CRC check bit.

Table 1.1 No CRC check instruction frame format

instruction	EE	XX	XX XX... XXX	FF FC FF FF
illustrate	frame header	instruction	instruction parameters	frame end

If you need CRC check format supported instructions, the instruction frame grid is shown in Table 1.2. The algorithmic program of the CRC can be downloaded in the website data download column. The CRC check does not include a frame head and frame end.

Table 1.2 Instruction frame formats with CRC checksum

instruction	EE	XX	XX XX... XXX	CRC16	FF FC FF FF
illustrate	frame header	instruction	instruction parameters	check bit	frame end

1.1.1 Example of instruction operation

1. Toggle the screen.

If you to display the screen shown in Figure 1-1 on the screen, the serial port instructions sent by the user host are as follows:

Figure 1-1 Toggle screen display

Microcontroller sending commands	EE [B1 00 00 01]FF FC FF FF
Command parsing	EE : the frame header
	B1 00 : toggle screen command
	00 01: the ID of the target screen to be displayed, 2 bytes
	FF FC FF FF : the end of the frame
prompt	The ID number of each screen is generated after being compiled by the host computer

2. Upload the button control ID.

If you tap the Start Run button in Figure 1-2, assume that the current screen ID is 2, the button ID is 1, the screen upload serial port instructions are as follows:

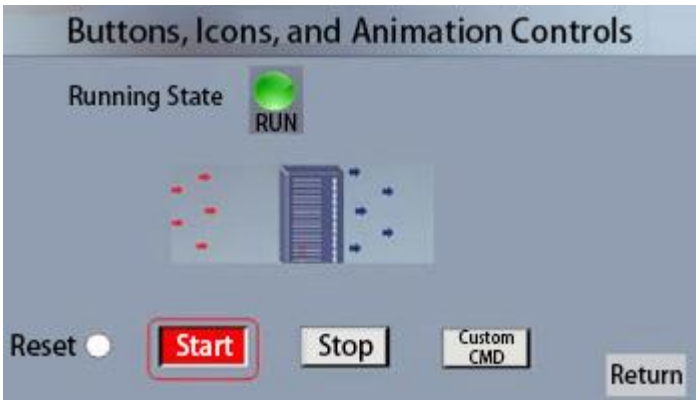

Figure 1-2 Upload the button control ID

screen upload instruction	EE【B1 11 00 02 00 01 10 01 01】FF FC FF FF
Command parsing	EE : the frame header
	B1 11 : the uploaded configuration control instructions;
	00 02 00 01 : the current screen ID is 2 and the control ID is 1
	10 : the control is a button control
	01 : the button control property is a switch type
	01 :t the button state changes from bounce up to pressed
	FF FC FF FF : the end of the frame
prompt	The ID numbers of all button controls are generated by the host computer configuration and can be modified yourself
illustrate	The user's single-chip microcomputer serial port can parse out which one is current, when it receives the above instruction data

3.Text display.

If you to display the current voltage number 220 in Figure 1-3, assume that the screen ID is 1 and the text control ID is 7 . Then the user host sends the following instructions:

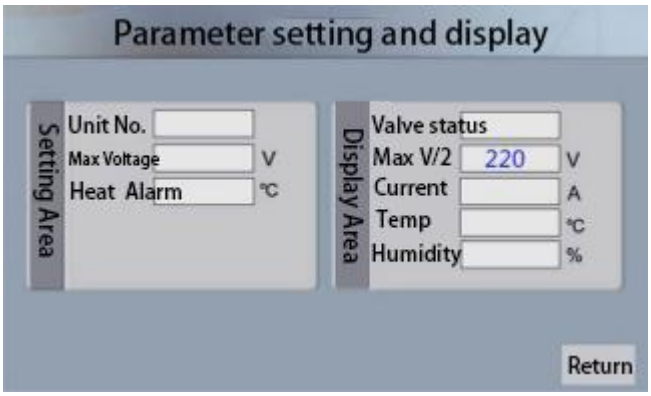

Figure 1-3 Text display

Microcontroller sending instructions	EE【B1 10 00 01 00 07 32 32 30】FF FC FF FF
Command parsing	EE : the frame header
	B1 10 : the configuration control instruction sent;
	00 01 00 07 : the screen ID is 1 and the control ID is 7
	32 32 30 : the ASCII code of the number 220;
	FF FC FF FF :the end of the frame
illustrate	Because the color, size, and position of the fonts are pre-configured on the PC, the user microcontroller only needs to send data for the ID number

4.System keyboard input.

If you need to display "1001" on the unit number in Figure 1-4, assume that the screen ID is 1 and the text control is 1, then just click on the text box, and then enter the input "1001 " in the pop-up system keyboard (need to switch between Chinese and English), and finally click OK, so that the entered text will be automatically embedded in the text box display, while uploading The instructions are as follows:

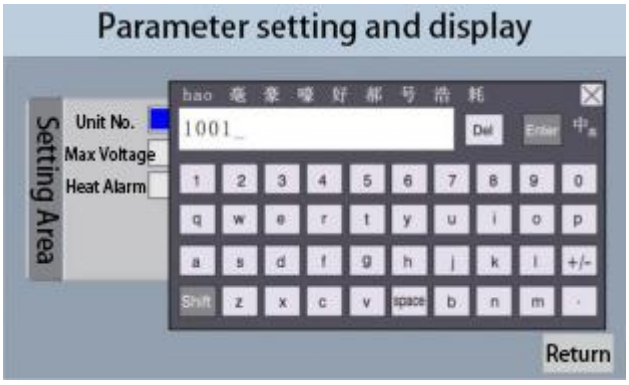

Figure 1-4 Keypad input parameter display

screen upload instruction	EE【B1 11 00 01 00 01 11 31 30 30 31 BA C5 00】FF FC FF FF
Command parsing	EE： the frame header
	B1 11： the received configuration control instructions;
	00 01 00 01： the screen ID is 1 and the control ID is 1
	11 ： that the control is a text control
	31 30 30 31 ： the ASCII code of the number 1001
	BA C5： the internal code of the Chinese character "号"
	00 ： the end of the character
	FF FC FF FF ： the end of the frame
illustrate	The user microcontroller receives the ASCII code uploaded on the screen to obtain the input information

5.Customize the instruction output.

Users can set the device to upload its own defined data strings after pressing a button. As shown in 1-5, the "Customize" button pops up, and the screen sends the command: 02; Send when pressed: 01.

Note: Custom data cannot contain "FF FC FF FF" group characters, otherwise it will conflict with the end of the frame, resulting in an abnormal execution. If multiple instructions need to be sent out at the same time, each instruction need to be separated between the orders.

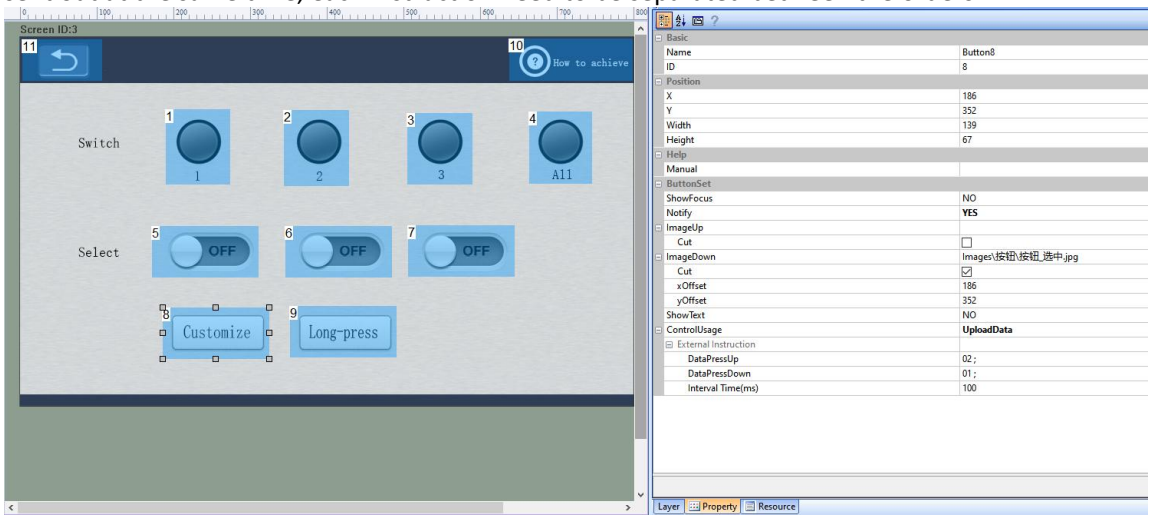

Figure 1-5 Customize the instruction output

6. Display the icon.

If the user wants to implement a status icon change display, for example, when running, display "run icon"; when stop, display "stop icon"; when reset, the icon disappears,; when abnormal the two icons flashing, then you can use the icon control to solve the problems.

First, all the status icons are generated into an ICON file (the ICON file contains all the state frame graphs) with the built-in icon generator in the software, and then the microcontroller sends instructions to control the frame playback of the ICON. For example, as shown in Figure 1-6, if the user wants to replace the "stop icon" with the "start icon", the sending command is as follows:

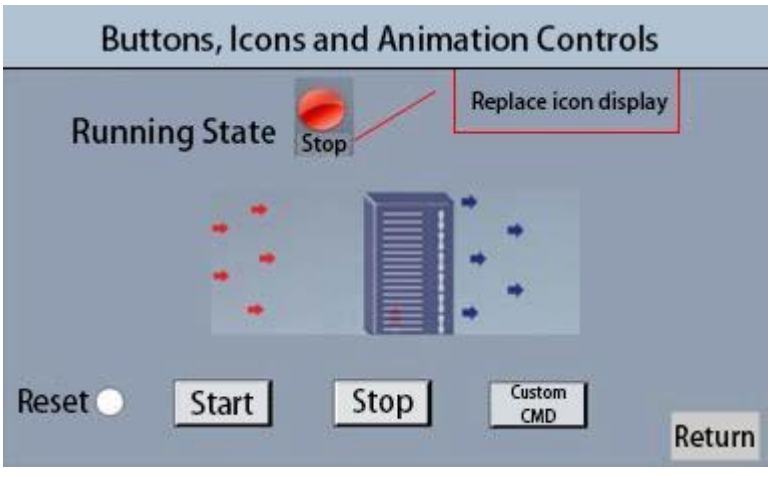

Figure 1-6 Display the icon

Microcontroller sending instructions	EE【 B1 23 00 03 00 01 01】FF FC FF FF
Command parsing	EE : the frame header
	B1 23 : an icon control instruction;
	00 03 00 01 : the picture ID is 3 and the control ID is 1
	01 : the frame ID of the "Run Icon"
	FF FC FF FF : the end of the frame
illustrate	ICON file has 2 picture frames, stop running is frame ID 0, start running is frame ID 1, so playing frame 1 shows the run icon and playing frame 0 shows the stop icon.

7.Gauge display.

If the user needs to turn the pointer of dial 1 in Figure 1-7 to degree 5, assuming that the screen ID is 4 and the instrument control ID is 1, the user's SCM sends the following instructions:

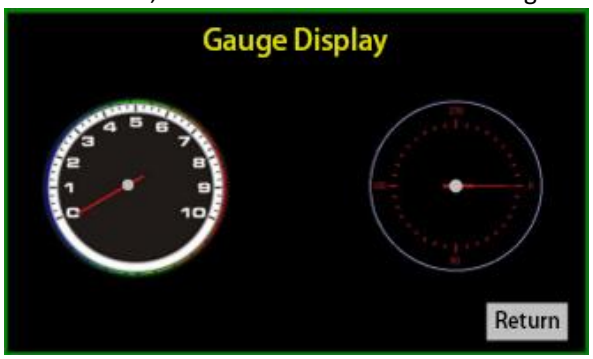

Figure 1-7 Gauge display

Device upload instructions	EE【B1 10 00 04 00 01 00 00 00 32】FF FC FF FF
Command parsing	EE： the frame header
	B1 10 ： the configuration control instruction sent
	00 04 00 01： the screen ID is 4 and the control ID is 1
	00 00 00 32： the numeric value 50
	FF FC FF FF ： the end of the frame
illustrate	Since the PC has preset the starting value of 0, the ending value of 100, and the sending value of 50, the pointer points exactly in the direction of scale 5

8.Curve display.

If the user needs to implement the curve display in Figure 1-8, the user host only needs to send the AD sampling sequence value, and the device will automatically zoom and pan to advance the display.

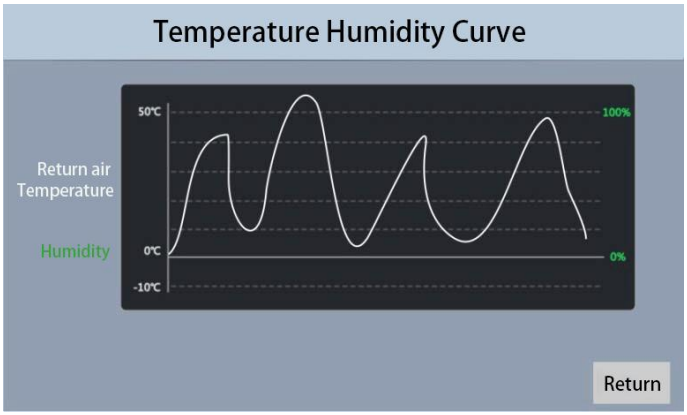

Figure 1-8 Curve display

9. Display progress bar and slider.

To achieve a rolling display of the progress bar shown in Figure 1-9, the user host only needs to send the displayed progress value. Suppose the progress bar configured on the PC starts at 0, ends at 100, and has a screen ID of 3 , the progress bar control ID is 1, then the user needs to display the progress bar in the center, send the following instruction:

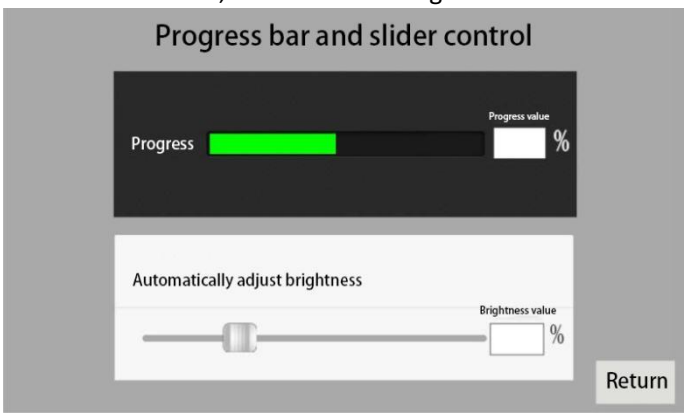

Figure 1-9 Progress and slider display

Device upload instructions	EE【B1 10 00 03 00 01 00 00 00 32】FF FC FF FF
Command parsing	EE represents the frame header
	B1 10 represents the configuration control instruction sent
	00 03 00 01 means that the picture ID is 3 and the control ID is 1
	00 00 00 32 represents the numeric value 50
	FF FC FF FF represents the end of the frame
illustrate	Since the PC presets the start value of the progress bar 0, the end value of 100 is sent Send a value of 50, and the progress bar is displayed exactly in the central position

10.The drop-down menu display.

Click the drop-down button, automatically pop up the drop-down menu, the drop-down menu option value is automatically loaded into the text box display, while the screen to pass the drop-down menu ID and option column values to the user microcontroller. The user can resolve which drop-down menu is currently in place and which option is selected. As shown in Figure 1-10, after the user clicks the drop-down button, the drop-down menu pops up, the user clicks 57.7V, and the data is automatically loaded into the text box.

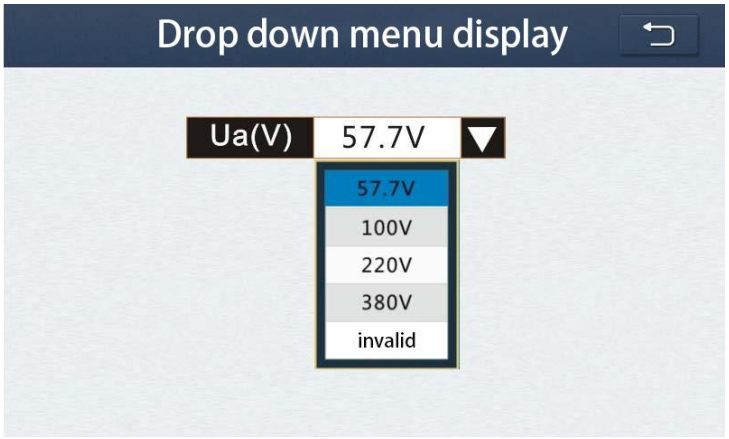

Figure 1-10 The drop-down menu display

11. Select Control Display.

The user pre-fills all the data of the selection item on the host computer, sets the number of displays, the magnification of the two ends, and then slides the parameter column, and the data is displayed in a rolling cycle. After scrolling stops, the screen uploads the ID of the preceding control and the selected item numeric value to the user microcontroller. The user can resolve which selection menu and which option is currently selected, and the slide selection control reference diagram is shown in Figure 1-11.

Figure 1-11 Select Control Display

12.QR code display.

The user microcomputer only needs to send the scan content character information, the screen automatically generates a two-dimensional code image of these contents, and the user's mobile phone scan to identify. The magnification of the two-dimensional code, the coding mode, and the display color can be set on the host computer. For example, if the user sends the character "www.elecrow.com" on the serial port, the screen automatically produces the corresponding two-dimensional code, as shown in Figure 1-12.

Figure 1-12 QR code display

1.2 Color format

The device supports a total of 2^16=65536 colors (65K colors), RGB is 565 format. The allocation of high and low bytes is shown in Table 1.3.

Table 1.3 RGB Color Assignment Format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Color assignment	R					G						B				

Example: Pure red F800H, pure blue : 00

Note: The user can debug and obtain the desired color and RGB values through the visualTFT host computer software .

1.3 Difference between configuration instruction set and basic instruction set

The instruction set is divided into two parts: the configuration instruction set and the basic instruction set. The main difference between the two is that the basic instruction set can be understood as the lowest instruction set, and most operations must contain coordinates, color, font and other parameter information; The configuration instruction set is directly an object-oriented ID operation, and the relevant parameters of these objects are all pre-configured in the host computer software and downloaded to the memory of the screen together with the picture.

The configuration instruction set meets 99% of user needs for simple, true "zero" code programming. In addition, the user can also combine the basic instructions and configuration instructions to complete the desired display function.

1.4 How to quickly become familiar with the instruction set

There are many serial port screen instruction sets, and users can use the matching serial port debugging board to debug the screen and the host computer online. Through the built-in command assistant of the host computer to issue serial port commands, simulating the user's single-chip microcomputer host, you can quickly familiarize yourself with the command function. The screen and the host computer VisualTFT online and command assistant are shown in Fig. 1-13 and Figure 1-14, respectively .

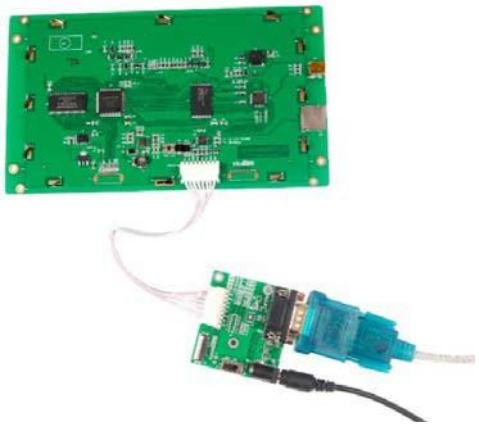

Figure 1-13 Screen and host computer online wiring diagram

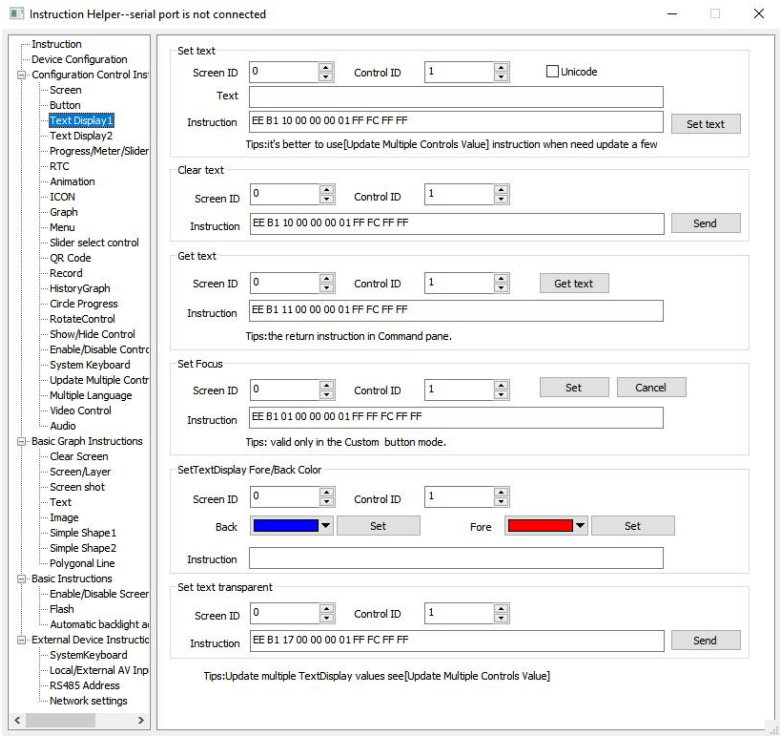

Figure 1-14 Instruction Helper

1.5 Configure the instruction set table

Table 1.4 Configuring the instruction set

category	instructions	instruction parameters	illustrate
Handshake commands	0x04	not	The handshake command is mainly used to determine whether the device is powered on and initialized, whether the communication is normal and whether it is online. After sending the command, the device returns 55 to indicate that the handshake is successful .

Reset the report	0x07	not	Once the device is powered on, unexpectedly rebooted, or the monitoring chip is reset, the relevant data will be uploaded immediately to inform the user that the device has been reset. After the user's host detects an unexpected reset of the device, the control program is re initialized and executed from the beginning.
Reset the device	0x07+0x35 +0x5A+0x53+ 0xA5	not	The host resets the device through the serial port command during operation. It is recommended to add this command when the host initializes the device, so that the device will reset after the host accidentally resets Return instruction format: EE 07 FF FC FF FF FF

Gets the device version	0xFE+0x01	not	This instruction is mainly used to read the firmware version of the screen, taking 2.22.1025.260 as an example Return instruction format: EE FE 02 16 00 02 04 01 01 04 FF FC FF FF ; Parameter description: 02 refers to the number 2 in decimal; 22 refers to the number 16 in decimal; 00 02 refers to the type of screen, refers to the commercial type; 04 01 refers to the number 1025 in decimal; 01 04 refers to the number 260 in decimal;
Backlight adjustment	0x60	Light_level	Sets the backlight brightness value 0x00: Backlight brightest 0xFF: Backlight off
Automatic screensaver mode	0X77	Enable+BL_ON+BL_OFF +BL_ON_Time	Sets the backlight brightness value and time value for the screensaver mode. After a period of no touch action, the screen automatically lowers its brightness and enters power-saving mode until it wakes up when the touch is pressed. Enable: Enable signal 0x00: Turn off power-saving mode 0x01: Turn on power-saving mode BL_ON (1 byte): The brightness value of the backlight after touch activation BL_OFF (1 byte): The brightness value of the backlight after entering power-saving mode BL_ON_Time (2 bytes): How long to enter the province in no touch action Electrical mode, unit of 1s Note: Only suitable for models with touch, without TP products require a program-controlled backlight
Buzzer control	0x61	Time	Time(1 byte): Response time, in 10ms

Configure the touchscreen	0x70	Cmd	Cmd (1 byte): Configuration parameter BIT0: 1 means touch screen is on, 0 means touch screen is off; BIT1: 1 means that the buzzer automatically rings when touched, and 0 means that the buzzer does not ring; BIT4~BIT2 : Touch upload method 000: Indicates that coordinates are uploaded only once when the touch screen is pressed 001: Indicates that the touch screen is pressed until the coordinates are uploaded once after release 010: When the touch screen is pressed all the time, upload coordinates every 100ms, and upload coordinates once when released 011: Indicates that the coordinates are uploaded once when the touch screen is pressed and released BIT5: 1 means that in 4 seconds to continuously click on a certain area 20 times, the screen enters touch alignment mode, 0 means to disable this function; BIT7-BIT6: Reserved Touch Upload format: When pressed, upload format: EE 01 X Y FF FC FF FF When released, upload format: EE 03 X Y FF FC FF FF X coordinates , Y coordinates are both 2 bytes, high bytes first
Touch screen calibration	0x72	not	After calibration, the device returns to EE 04 FF FC FF FF or a quick 20 tap in 4 seconds at a certain point in the non-touch area. Automatically enters touch calibration mode and automatically returns to the current display after calibration is complete
Touchscreen experience	0x73	Enable	Enable: Enable signal 0x00: Turn off the experience 0x01: Experience enable After the user presses the touch, a solid red circle is displayed at the corresponding coordinates of the screen. Used to test touch screen accuracy.

Set the baud rate	0xA0	Baudset	Baudset (in bps, 1 byte), baud rate ordering: 0x00: 1200 0x01: 2400 0x02: 4800 0x03: 9600 0x04: 19200 0x05: 38400 0x06: 57600 0x07: 115200 0x08:1M 0x09:2M 0x0A : 218750 0x0B : 437500 0x0C : 875000 0x0D : 921800
Switch screens	0xB1+0x00	Screen_id	Switch from the current screen to the target screen display Screen_id(2bytes): The target screen ID This command mainly implements the switching screen display
Use animation effects to swap the frame	0xB1+0x05	SCREEN_ID(2Byte) EFFECT(1Byte) AREA_EN(1Byte) AREA_LEFT(2Byte) AREA_RIGHT(2Byte) AREA_TOP(2Byte) AREA_BOTTOM(2Byte)	SCREEN_ID (2Byte): The target screen ID EFFECT(1Byte): Toggle effect, 0 disable effect, 1 left to right 2 right to left 3 from top to bottom 4 from bottom to top AREA_EN (1 Byte): 0 full screen effect, 1 area effect AREA_LEFT (2Byte) area left distance, Pixel AREA_RIGHT(2Byte) area right distance AREA_TOP(2Byte) area upper distance AREA_BOTTOM(2Byte) area lower distance This instruction is mainly implemented Use animation effects to switch frames
Read the screen	0xB1+0x01	not	Reads the ID value of the current screen Instruction return format:EE B1 01 Screen_id FF FC FF FF Screen_id (2 bytes): The number of the current screen This command is mainly used to obtain which screen the current screen is in
Upload the button control ID value	0xB1+0x11	Screen_id+Control_id +Control_type+Subtype +Status	When a button is pressed, the device actively uploads the button ID information Upload format：EE【B1 11 Screen_id Control_id Control_type Subtype Status】FF FC FF FF Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Control_type (1 byte): A fixed value of 0x10, expressed as the button control type Subtype (1 byte). : A subtype of the button control Status (1 byte): button state 0x00: Screen switching, indicating that the screen switch button is currently pressed 0x01: The switch type, which indicates that the switch button is currently pressed 0x02: Custom key-value that represents the custom key-value button currently pressed 0x03: A custom instruction that represents the custom instruction button that is currently pressed 0x04: pop-up menu, which indicates the pop-up menu button 0x00: button changes from pressed to bounced 0x01: The button changes from bouncing to pressing This command mainly tells the user which screen the first few buttons have been pressed

Custom instruction output	not	not	The instructions issued after pressing a button control can be customized by the host computer, and multiple instructions can be output at the same time, and multiple instructions are separated by semicolons
Set the button bounce up and press state	0xB1+0x10	Screen_id+Control_id+ Status	Set a button to pop up or press the display state Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Status (1 byte): The button state 0x00: The button changes from pressed to bounce 0x01: The button changes from bounce up to press This command is mainly used to change the display state of the button

Read the button control state	0xB1+0x11	Screen_id+Control_id	Query whether a button is currently pressed or popped up Screen_id (2 bytes): screen number Control_id (2 bytes). : Control number Return instruction format: EE B1 11 Screen_id Control_id Control_type Subtype Status FF FC FF FF Control_type (1 byte): Fixed value 0x10, expressed as the button control type Subtype: A subtype of the button control 0x00: Screen switching, indicating that the screen switch button is currently pressed 0x01: The switch type, which indicates that the switch button is currently pressed 0x02: Custom key-value that represents the custom key-value button currently pressed 0x03: A custom instruction that represents the custom instruction button that is currently pressed 0x04: A pop-up menu that represents a pop-up menu button Status: 0x00 pop-up status 0x01 pressed state Note: The definition of the return parameter is consistent with the button control value upload instruction
Updates the Number of Text Controls value	0xB1+0x10	Screen_id+Control_id+ Strings	Writes text data to the specified text control Screen_id (2 bytes): screen number Control_id (2 bytes). : Control number Strings: The value of the text written by the user; This instruction is primarily used to implement text data display
Clears the text control content	0xB1+0x10	Screen_id+Control_id	Clears the text control contents Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number This directive is primarily used to clear the text control content

Read the Number of Text Controls value	0xB1+0x11	Screen_id+Control_id	Read the currently displayed numeric value of a text control Screen_id (2 bytes): screen number Control_id (2 bytes). : Control number Return instruction format: EE B1 11 Screen_id Control_id Control_type Strings FF FC FF FF FF Return parameter: Control_type (1 byte): Fixed value 0x11, represented as a text control type Strings: The currently displayed text value with 1 appended to the text 0x00 as the finisher
Sets the text control flash period	0xB1+0x15	Screen_id+Control_id+ Cycle	Set the text control blink speed Screen_id (2 bytes): Screen number Control_id (2 bytes): The control number Cycle (2 bytes): Blink period (10 millisecond units), 0 means no flickering
Sets the scroll speed of the text control	0xB1+0x16	Screen_id+Control_id+ Speed	Set the text control scroll speed (direction to the left) Screen_id (2 bytes): Screen number Control_id (2 bytes).): The control number Speed (2 bytes): Text scrolling speed (pixels per second), 0 means no scroll

Sets the text control back to the landscape	0xB1+0x18	Screen_id+Control_id+ BK_Color	Sets the text control background color Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number BK_Color (2 bytes): The background color RGB color value
Sets the text control foreground color	0xB1+0x19	Screen_id+Control_id+ FORE_ Color	Sets the foreground color of the text control Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number FORE_ Color (2 bytes): The foreground RGB color value
Sets the text backgrou nd clarity	0xB1+0x17	Screen_id+Control_id	Turns text with background into background transparent display Screen_id (2 bytes): screen number Control_id (2 bytes). : Control number

Format text display	0xB1+0x07	Screen_id+Control_id+Sign+Fill_zero+Value	Enter text to display according to corresponding conditions **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Sign** (1 byte): data type, respectively: Unsigned int: unsigned integer, 0x00;Int: signed integer, 0x01; Float: single precision floating point number, 0x02; Double: double precision floating point number, 0x03; **Fill_ Zero** (1 byte): the number of decimal places. If the number is less than 0, add 0x80; **Value** (4 bytes): added data, expressed in hexadecimal number;
External update picture display (IoT type not supported , other models supported).	0x39	X+Y+Wight+Height	Specify the coordinates and width of the picture display, and then send RGB data **X** (2 bytes): X axis coordinate value in point **Y** (2 bytes): Y-axis coordinate value in point **Width** (2 bytes): set the width of the text range **Height** (2 bytes): set the height of the text range **Note**: First send the position and width of the picture, then send the BMP data of the picture, with the RGB high byte of the picture first.
Updates the progress bar control numeric value	0xB1+0x10	Screen_id+Control_id+Progressvalue	Writes the displayed data to the specified progress bar control **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Progressvalue** (4 bytes): the new progress bar value written by the user. This instruction is mainly used to increase or decrease the progress bar
Read the progress bar control numeric value	0xB1+0x11	Screen_id+Control_id	Read the current value of the specified progress bar control **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Return instruction format:** EE B1 11 Screen_ id Control_ id Control_ type Progressvalue FF FC FF FF **Return parameter:** Control_ Type (1 byte): fixed value 0x12, representing the type of progress bar control **Progressvalue** (4 bytes): current progress bar value

The slider control upload format	not	not	When dragging the slider, the device continuously uploads the cursor value as follows: **Upload format:** EE [B1 11 Screen_id Control_id Control_type Slidervalue] FF FC FF FF **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Control_ Type** (1 byte): fixed value 0x13, represented as slider **control Slidervalue** (4 bytes): represents the current cursor value
Updates the slider control value	0xB1+0x10	Screen_id+Control_id+ Slidervalue	Write the displayed data to the specified slider control **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Slidervalue** (4 bytes): the new slider value written by the user This instruction is mainly used to control the position of the slider cursor
Read the slider control numeric value	0xB1+0x11	Screen_id+Control_id	Read the current cursor value of the specified slider control **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Return instruction format**: EE B1 11 Screen_ id Control_ id Control_ type Slidervalue FF FC FF FF **Return parameter:** Control_ Type (1 byte): fixed value 0x13, representing the sliding bar control type **Slidervalue** (4 bytes): the currently displayed slider value
Update the Number of Gauge Controls value	0xB1+0x10	Screen_id+Control_id+ Metervalue	Write the displayed data to the specified instrument control **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): control number **Metervalue** (4 bytes): the new instrument value written by the user This command mainly realizes the rotation of the instrument pointer
Read the Number of Gauge Controls value	0xB1+0x11	Screen_id+Control_id	Read the current value of the specified instrument control Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Return instruction format: EE B1 11 Screen_ id Control_ id Control_ type Metervalue FF FC FF FF Return parameter: Control_ Type (1 byte): fixed value 0x14, indicating the type of instrument control Metervalue (4 bytes): currently displayed instrument value Note: The definition of the return parameter is consistent with the value of the updated instrument control
Set the background color of the progress bar	0xB1+0x18	Screen_id+Control_id+ Bk_ Color	Set the background color of the progress bar Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Bk_ Color (2 bytes): background color RGB color value

Set the foreground color of the progress bar	0xB1+0x19	Screen_id+Control_id+ Fore_ Color	Sets the foreground color of the progress bar Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Fore_Color (2 bytes): The foreground RGB color value
Bulk update the Number of controls value	0xB1+0x12	Screen_id+Control_id0+ Len0+ Strings0+ ... Control_idn+ Lenn + Stringsn	Batch update text, progress bar, gauge, button and icon control values. The advantage of batch update is that one command can update all the contents of the current screen to improve the refresh speed and efficiency. Screen_ Id (2 bytes): picture number Control_ Id0 (2 bytes): the number of the first control Len0 (2 bytes): the byte length of the first control Strings0 (variable length): the value of the first control, the number of bytes Len0 length And determined Control_ Idn (2 bytes): the nth control number Lenn (2 bytes): the byte length of the nth control Stringsn (variable length): the numerical value of the nth control. The number of bytes depends on Lenn length

Sets the cursor focus	0xB1+0x02	SCREEN_ID + CONTROL_ID + ENABLE	Display cursor on the specified text control (only applicable to the mode of customized keyboard) Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Enable (1 byte): enable the cursor display. 0x00: Close the display; 0x01: Enable display
Manually disable/enable screen updates	0xB3	ENABLE	Manual Disable/Enable Screen Update Enable (1 byte): update enable 0x00: Update prohibited; 0x01: Enable Update This command mainly solves the problem that the number of real-time and dynamic updated controls in a certain screen is too large, resulting in slow screen update speed. How to use it: The user first sends the command to disable screen updating, then sends the content to be updated in the whole screen, and finally enables screen updating.
Disable/enable controls	0xB1+0x04	SCREEN_ID + CONTROL_ID + ENABLE	Disable/enable controls Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Enable (1 byte): control enable 0x00: Disable control; 0x01: Enable Control This command is mainly used to disable or enable certain controls in a certain screen. You can disable or enable certain controls according to user requirements.

Mask/hide controls	0xB1+0x03	SCREEN_ID + CONTROL_ID + ENABLE	Hide or mask a control. Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Enable (1 byte): mask or hide enable 0x00: Shielding or hiding controls; 0x01: Shielding/hiding removal This command is often used to disable the function of a specified button control at a certain time, or to hide a control from display
Animated controls display	0xB1+0x20	SCREEN_ID + CONTROL_ID	Start animation playback Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number After starting playback, the animation starts playing from frame header 0 each time, and the format only supports GIF
	0xB1+0x21	SCREEN_ID + CONTROL_ID	Stop animation playback Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): After the control number execution stops playing, the next time will start playing from frame header 0
	0xB1+0x22	SCREEN_ID + CONTROL_ID	Pause animation playback Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): After the control number is paused, it will continue playing from the pause frame next time
	0xB1+0x23	SCREEN_ID + CONTROL_ID + FlashImgae_ID	Specify frame playback Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number FlashImgae_ ID (1 byte): ID of an animation frame Specifies that playback starts at a frame
	0xB1+0x24	SCREEN_ID + CONTROL_ID	Play the previous frame Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number
	0xB1+0x25	SCREEN_ID + CONTROL_ID	Play the next frame Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number
Animation control value upload	0xB1+0x26	SCREEN_ID + CONTROL_ID+ Status + FlashImgae_ID	Relevant information uploaded by the device when pressing or sliding an animation control: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Status (1 byte): 0x00 means touch and press, 0x01 means pop; FlashImgae_ ID (1 byte): indicates the animation frame displayed when the screen is pressed; Note: Users can disable/enable animation control value upload in PC configuration.

The icon control display	0xB1+0x23	SCREEN_ID + CONTROL_ID +IconImgae_ID	The user first uses the icon generator provided by the host computer to synthesize all pictures in different states into an ICON file, and then the host computer specifies an icon frame to display. Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number IconImgae_ ID (1 byte): ID of an icon frame This command is often used to display several pictures in different states at the same position on the screen. Note: If the user needs the icon to disappear, he can make a transparent PNG image to achieve the effect of disappearance.
The icon control value uploads	0xB1+0x26	SCREEN_ID + CONTROL_ID+ Status + IconImgae_ID	When pressing or sliding an icon control, the device uploads relevant information: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Status (1 byte): 0x00 means touch and press, 0x01 means pop; IconImgae_ ID (1 byte): indicates the icon frame displayed when the screen is pressed; Note: Users can disable/enable icon control value upload in PC configuration
Sets the icon position	0xB1+0x28	Screen_id + Control_id + X + Y	Set icon position Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number X (2 bytes): X coordinate position displayed Y (2 bytes): displayed Y coordinate position This command is mainly used to dynamically adjust the position of icon control display.
Curve controls	0xB1+0x30	SCREEN_ID + CONTROL_ID +CHANNEL+COLOR	Adds the specified data channel Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7). COLOR (2 bytes): The color of the data channel
	0xB1+0x31	SCREEN_ID + CONTROL_ID+CHANNEL	Deletes the specified data channel Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7).
	0xB1+0x32	SCREEN_ID + CONTROL_ID+ CHANNEL+DATA_LEN+DATA	Add new data at the end of the specified channel Add new data at the end of the specified data channel. When the data length exceeds the buffer length, the old data is moved to the left. **Screen_ Id** (2 bytes): picture number; **Control_ Id** (2 bytes): control number **CHANNEL** (1 byte): 8 data channels in total, number range (0~7) **DATA_ LEN** (2 bytes): data length **DATA:** variable length data, the length is determined by DATA_ LEN specified format: the data represents the value in the Y direction, and the value in the X direction will automatically increase according to the horizontal scaling factor. For example, when the horizontal scaling factor is 1, X axis will automatically add 1 for each point inserted, and when the horizontal scaling factor is 5, X axis will automatically add 5 for each point inserted

	0xB1+0x33	SCREEN_ID + CONTROL_ID +CHANNEL	Clears the data for the specified channel Screen_ id (2 bytes): screen number; Control_id (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7).
	0xB1+0x34	SCREEN_ID + CONTROL_ID+ XOFFSET + XMUL + YOFFSET+YMUL	Specify vertical/horizontal zoom/pan Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number XOFFSET (2 bytes): horizontally offset data points, with positive left shift and negative right shift XMUL (2 bytes): horizontal scaling factor, unit: 0.01 YOFFSET (2 bytes): vertical offset value, with downward shift being positive and upward shift being negative YMUL (2 bytes): vertical scaling factor, unit: 0.01 Calculation formula of sampling point and coordinate point: the value of the Nth sampling point is V X coordinate=(N-XOFFSET) * XML * 0.01 Y coordinate=(V-YOFFSET) * YMUL * 0.01
	0xB1+0x35	SCREEN_ID + CONTROL_ID+ CHANNEL+DATA_LEN+DAT A	Add new data at the front end of the specified channel and insert new data at the front end of the specified data channel. When the data length exceeds the buffer length, the old data moves to the right Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number CHANNEL (1 byte): There are 8 data channels in total, and the number range is (0~7) DATA_ LEN (2 bytes): data length DATA: variable length data, the length is determined by DATA_ LEN designation
Set the drop-down menu control	0xB1+0x13	SCREEN_ID + CONTROL_ID+ Enable+Textctrl_ID	Set the text box to which menu option data is automatically written to when the drop-down menu pops up Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Enable (1 byte): menu enable bit, 00 is hidden, 01 is displayed, after clicking the menu, it is automatically hidden Textctrl_ ID (2 bytes): Text control
The drop-down menu control value Uploads	0xB1+0x14	SCREEN_ID + CONTROL_ID+0x1A+ Meundata_ID+ Status	Data frame format uploaded by menu control. After the drop-down menu data item is selected, the system automatically uploads the control ID value and the selected item value. Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Meundata_ ID (1 byte): Which option data Status (1 byte): touch status, 0 means pop up; 1 means press The corresponding command data will be uploaded when the pull-down menu control value upload command pops up or is pressed.

Sets the slide selection control current option	0xB1+0x10	SCREEN_ID + CONTROL_ID+ Select_data_ID	Sets the data item currently displayed by the slide selection control Screen_id (2 bytes): screen number Control_id (2 bytes). : Control number Select_data_ID (1 byte): The first option data
Slide the selection control value uploaded	0xB1+0x11	SCREEN_ID + CONTROL_ID+0x1B+ Select_data_ID	The scroll control mainly implements sliding up and down to select the required data, and the system automatically uploads the control ID value and the currently selected item after it is selected. Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Select_data_ID (1 byte): The first option data
Update the QR code control	0xB1+0x10	Screen_id+Control_id+ Strings	Generate 2D code map of specified display content Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): control number Strings (variable length): QR code character content; This instruction is mainly used to display the two-dimensional code map. The instruction format is the same as the text control
Set the timer	0xB1+0x40	Screen_id+Control_id+ timedata	Set the RTC timer timer Screen_id (2 bytes): Screen number Control_id (2 bytes): The RTC control number timedata (4 bytes): The value of the timer This instruction is only for product devices whose hardware supports RTC clocks, and the clock control properties are pre-set on the host computer
Start the timer	0xB1+0x41	Screen_id+Control_id	Start the timer Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The RTC control number The timer time ends, the screen uploads the event notification, the upload format: EE B1 43 Screen_idControl_id 17 FF FC FF FF FF
Pause the timer	0xB1+0x44	Screen_id+Control_id	Pause the timer Screen_id (2 bytes): Screen number Control_id (2 bytes): RTC control number After the pause timer, the send start timer continues to count
Stop the timer	0xB1+0x42	Screen_id+Control_id	Stop the timer Screen_ id (2 bytes): Screen number Control_ id (2 bytes): RTC control number When the timer is stopped, the timing value reverts to the original set value

Read the timer	0xB1+0x45	Screen_id+Control_id	Read the timer Screen_ id (2 bytes): Screen number Control_ id (2 bytes): RTC control number After reading the timer, the screen returns the corresponding timing value Return format: EE B1 45 Screen_idControl_id17 Strings FF FC FF Ff Return parameter: Screen_id (2 bytes): screen number Control_id (2 bytes): RTC control number 17 (1 byte): 0x17, Fixed code Strings (4 bytes): The timing value of the read

Set the 485 screen address (Only screens with 485 interfaces are supported).	0xA5+0x01	Addr	Set the 485 screen address Addr: 2 bytes, representing the screen address, the address starts at 1, 0 is the broadcast address After the address is set successfully, all instructions will add address information, such as: EE Addr1 Addr0 Screen_idScreen_id… FF FC FF FF
Cancel the 485 screen address (Only screens with 485 interfaces are supported).	0xA5+0x00	not	Cancel the 485 address information
Write data to FLASH	0x87	Addr + Data0… +Datan	Saves the data in the specified FLASH address and uses it as an EEPROM Addr (4 bytes): The starting address of the data write Datan (1 byte): The data written The storage space is 128K bytes and the address range is 0 to 0x1FFFF Returns after successful write: EE 0C FF FC FF FF Note: Because Nandflash has an erasing life (about 100,000 times), important parameters and core data that need to be rewritten repeatedly are not recommended
Read data saved in FLASH	0x88	Addr + Length	The data written to random or sequential memory is read out Addr (4 bytes): The data read start address Length: The length of the read data in bytes The format of the data returned is: EE 0B Data0 … Datan FF FC FF FF
Take a screenshot of the current screen and save it in FLASH	0x46	Image_ID	Saves the current screen display to the device FLASH Image_ID (1 byte): User-defined screen number saved in memory

Displays screenshots saved in FLASH (basic, commercial, economic, IoT Not supported).	0x47	Image_ID	Displays screenshots saved in the device FLASH Image_ID (1 byte): User-defined screen number saved in memory
Lock down the system configuration	0x08	0xA5+0x5A+0x5F+0xF5	Prevents the receipt of incorrect instruction frames during system operation from causing accidental modification of the system configuration. Once the configuration is locked, the device will not be able to receive external serial commands to modify it until the system is unblocked. Configuration parameters include: baud rate, touch and matrix keyboard operating modes, auto backlight adjustment parameters. Return format: EE 17 FF FC FF FF
Locking system configuration	0x09	0xDE+0xED+0x13+0x31	Once the system configuration lock is lifted, the device can re-receive external serial port commands to modify configuration parameters. Returns the format EE 18 FF FC FF FF

An alarm event is triggered	0xB1+0x50	Screen_id+Control_id+Value+（Sec+Min+Hour+Day+Week+Mon+Year）	Alarm event triggering Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): data record control number Value (2 bytes): alarm event ID number If the customer selects the command time, the command will add the following parameters: Sec (1 byte): seconds, represented by BCD code Min (1 byte): minute, represented by BCD code Hour (1 byte): when, BCD code represents Day (1 byte): day, represented by BCD code Week (1 byte): week, represented by BCD code Mon (1 byte): month, represented by BCD code Year (1 byte): year, represented by BCD code This command transfers the corresponding event ID to the screen to convert the corresponding event ID number into the alarm event output set by the user

The alarm event is lifted	0xB1+0x51	Screen_id+Control_id+Value+（Sec+Min+Hour+Day+Week+Mon+Year）	Alarm event cancellation Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): data record control number Value (2 bytes): alarm event ID number If the customer selects the command time, the command will add the following parameters: Sec (1 byte): seconds, represented by BCD code Min (1 byte): minute, represented by BCD code Hour (1 byte): when, BCD code represents Day (1 byte): day, represented by BCD code Week (1 byte): week, represented by BCD code Mon (1 byte): month, represented by BCD code Year (1 byte): year, represented by BCD code This command is used to transfer the corresponding event ID to the screen, so that the corresponding event ID number can be converted into the alarm release event output set by the user
The data record control adds a regular record	0xB1+0x52	Screen_id+Control_id+Strings	Add a regular record Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The data record control number Strings: The contents of a string added by the user
Data Record Control Cleans the record data	0xB1+0x53	Screen_id+Control_id	Clear the record data Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The data record control number
The data record control sets the record display offset (i.e. scroll bar position).	0xB1+0x54	Screen_id+Control_id+Offset	Set the record offset (i.e. scroll bar position) Screen_id (2 bytes): Screen number Control_id (2 bytes): The data record control number Offset (2 bytes): The line number (that is, the position of the scroll bar).
The data record control gets the current number of records	0xB1+0x55	Screen_id+Control_id	Gets the current number of data records Screen_id (2 bytes): Screen number Control_ id (2 bytes): The data record control number return format：EE B1 55 Screen_idControl_id 1D Count FF FC FF FF

The data record control reads a row of records	0xB1+0x56	Screen_id+Control_id+ Position	Read a row of records Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The data record control number Position: The line number of a row return format:EE B1 56 Screen_idControl_id 1D Strings FF FC FF FF
The data record control modifies regular records	0xB1+0x57	Screen_id+Control_id+ Position+Strings	Modify a regular record Screen_ id (2 bytes): Screen number Control_ id (2 bytes): Data record control number Position: The line number of a row Strings: The contents of a string added by the user
The data record control deletes the regular record	0xB1+0x58	Screen_id+Control_id+ Position	Delete regular records Screen_id (2 bytes): Screen number Control_ id (2 bytes): The data record control number Position: The line number of a row
The data record control is plugged into the regular record	0xB1+0x59	Screen_id+Control_id+ Position+Strings	Insert a regular record Screen_ id (2 bytes): Screen number Control_ id (2 bytes): Data record control number Position: The line number of a row Strings: The contents of a string added by the user
The data record control selects a record	0xB1+0x5A	Screen_id+Control_id+ Offset	Select a record Screen_id (2 bytes): Screen number Control_ id (2 bytes): The data record control number Offset (2 bytes): The line number
The data record control adds multiple records at a time	0xB1+0x5B	Screen_id+Control_id+ Count+Record0_size+ Record0+Record1_size+ Record1+…	Add multiple records at once Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Count: Number of records added Recordx_size (2 bytes): The size of each record Recordx (not definite length): The data corresponding to each record
Export Data with an SD card Records the data of the control (CSV format).	0xB1+0x5C	Screen_id+Control_id	The SD card exports the data for the data logging control (CSV format). Screen_ id (2 bytes): Screen number Control_id (2 bytes): The data record control number Return format: EE B1 5C Screen_idControl_id00 FF FC FF FF

History curve	0xB1+0x60	Screen_id+Control_id+ Value	Historical curve setting sample value Parameter description: **Screen_ Id** (2 bytes): picture number **Control_ Id** (2 bytes): data record control number **Value** (variable length): The data types are UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT. The data of each channel corresponds to the set data type. If there are two channels, the data type is UINT8, and two data types of UINT8 are added accordingly This instruction is used to add the sampling value data of the historical curve, and the corresponding data is added to the instruction. The instruction is sent to the screen, and the screen draws the curve according to the corresponding data value.
	0xB1+0x61	Screen_id+Control_id+ Enable	History curve disables/enables sampling Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Text control number Channel (1 byte): Channel ID Show (1 byte): Disable/enable bits are displayed 0x00: hidden; 0x01: Display; This directive is mainly used to disable/enable sampling of historical curves. If the user needs to not enter a certain piece of data, it can be implemented using the disable sampling instruction; If the user needs to re-assign the sampling function, it needs to be implemented with the enable sampling command.
	0xB1+0x62	Screen_id+Control_id+ Show	History Curve Hide/Show Channel Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number Channel (1 byte): channel ID Show (1 byte): display disable/enable bit 0x00: hidden; 0x01: display; This command is mainly used to display/hide some channels of the history curve. There are multiple channel curves in the user interface. If you need to view and analyze a channel in detail, you can use the command to hide the curve of the corresponding channel, or you can use the command to display the channel curve.
	0xB1+0x63	Screen_id+Control_id+00 +Sample_Count	Setting time length of historical curve (i.e. number of sampling points) Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number 00 (1 byte): control type ID number, 0x00: set time width Sample_ Count (2 bytes): number of sampling points displayed on the screen This instruction is mainly used to set the time length of the historical curve, that is, to set the sampling points of the historical curve.

	0xB1+0x63	Screen_id+Control_id+01	Zoom history curve to full screen Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number 01 (1 byte): control type ID number, 0x01: zoom to full screen This command is mainly used to set the scaling of historical curves, that is, you can set the scaling ratio of historical curves to full screen.
	0xB1+0x63	Screen_id+Control_id+02 +Zoom+Max_Zoom+ Min_Zoom	Historical curve setting scaling factor Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number 02 (1 byte): control type ID number, 0x02: scaling factor Zoom (2 bytes): zoom percentage Max_ Zoom (2 bytes): zoom limit, maximum number of sampling points can be displayed on one screen Min_ Zoom (2 bytes): zoom limit, minimum number of sampling points displayed on one screen This command is mainly used to set the scaling system of the historical curve according to the user settings The number is displayed so that the drawn curve can conform to the angle viewed by the user.
	0xB1+0x64	Screen_id+Control_id+ Max+Min	Historical curves set the range of numeric values Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Text control number Max(4 bytes): Maximum Min (4 bytes): Min This instruction is mainly used to set the maximum and minimum values of the historical curve, i.e. set Sets the display range of the history curve.

Sets the rotation angle of the rotation control	0xB1+0x10	Value	Rotation control Set rotation angle Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number Value (2 bytes): rotation angle, 0-360 Degree; At the same time, the rotation control supports small angle rotation with granularity of 0.1 degree, Value=actual angle * 10+0x8000, for example, 5.3 degree setting value=5.3 * 10+0x8000 This command is mainly used to set the rotation angle of the rotation control. The user can use this command to rotate the pointer according to the angle set by the rotation center
Multi-language switching instructions	C1	Ui_lang + check	Multi-language switching instructions Parameter description: Ui_lang (1 byte): BIT3~BIT0 : Identity language index, range: 0- 9 BIT6~BIT4 : Reserved, default 0 BIT7: Identify system keyboard language, 0:Chinese, 1: English check (1 byte): For verification, check = 0xC1 + Ui_lang, take the lower bytes . The instruction is generally used for multi-language switching instructions, the user can use the instructions to set the keyboard output language is Chinese or English.
Obtain the system language reference	C2	not	Command format: EE [C2] FF FC FF FF Parameter description: None This instruction is used to obtain the language index currently used by the serial port screen; Return instruction: EE [C2 Lang] FF FC FF FF Return parameter: Lang (1 byte): BIT3~BIT0: identification language index, range: 0-9 BIT6~BIT4: Reserved, 0 by default BIT7: Identification system keyboard language, 0: Chinese, 1: English

Text control numeric variable adjustment	0xB1+0x1A	Screen_id+Control_id+ Option+Delta+Min_Limit+ Max_Limit	Digital variable adjustment Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): text control number Option (1 byte): bit0-0 decreases and 1 increases; Bit1-0 no cycle regulation, 1 cycle regulation Delta (2 bytes): increment Min_ Limit (2 bytes): minimum value Max_ Limit (2 bytes): maximum value
RTC time settings (Hardware support required).	0x81	Sec+Min+Hour+Day +Week+Mon+Year	Time parameter setting Sec: seconds setting Min: sub setting; Hour: hour setting Day: day setting Week: week setting Mon: month setting Year: year setting It is recommended that users directly use the clock control to complete RTC time display and calibration. Note: 1 byte each, represented by BCD code, is set as 0x00 on Sunday
Read the RTC clock (Hardware support required).	0x82	not	Data return format: EE+0xF7+Year+Mon+Week+Day+Hour+Min+Sec+FF FC FF FF Note: 1 byte each, represented by BCD code It is recommended that the user directly use the clock control to complete the RTC display and time calibration
Store the current picture to TF card (basic, commercial and economical support, but not IoT)	0x34	00	The corresponding instruction format for storing the current picture in the TF card is EE 34 00 FF FC FF FF After sending the corresponding command, the corresponding command will be automatically generated in the TF card Bmp file. If the storage is successful, the return format is EE 10 FF FC FF FF If the storage fails, the return format is EE 11 FF FC FF FF
Get the wireless network settings placed (IoT support only).	0xD0+0xA1	not	Gets the wireless network settings The corresponding instruction format is: EE D0 A1 FF FC FF FF FF Return format: EE D0 A1 WIFI_GET_CFG FF FC FF FF
Gets the network status (IoT support only).	0xD0+0xA2	not	Gets the network status The corresponding instruction format is: EE D0 A2 FF FC FF FF FF Return format: EE D0 A2 WIFI_GET_STATE FF FC FF FF

Gets the network settings (IoT support only).	0xD0+0xA5	not	Gets the network settings The corresponding instruction format is: EE D0 A5 FF FC FF FF Return format: EE D0 A5 NETWORK_GET_CFG FF FC FF FF
Get network services (IoT support only).	0xD0+0xA7	not	Gets the network settings The corresponding instruction format is: EE D0 A7 FF FC FF FF Return format: EE D0 A7 NETWORK_SERVICE_GET_CFG FF FC FF FF
Search for wireless networks (IoT support only).	0xD0+0xA3	not	Search for wireless networks The corresponding instruction format is: EE D0 A3 FF FC FF FF Return format: EE D0 A3 WIFI_SCAN _APLIST FF FC FF FF
Save the network settings (IoT support only).	0xD0+0xA4	NETWORK_SET_CFG	Save the network settings The corresponding instruction formats are: EE D0 A4 NETWORK_SET_CFG FF FC FF FF
Save the network service (IoT support only).	0xD0+0xA6	NETWORK_SERVICE_S AND	Save the network service The corresponding instruction formats are: EE D0 A6 NETWORK_SERVICE_SET FF FC FF FF FF
Save wireless network settings (IoT support only).	0xD0+0xA0	WIFI_SET_CFG	Save the wireless network settings The corresponding instruction formats are: EE D0 A0 WIFI_SET_CFG FF FC FF FF
Send network data (IoT support only).	0xD0+0xAC	Count+Strings	Send network data The corresponding instruction format is: EE D0 AC Count Strings FF FC FF FF Parameter format: Count: The number of bytes sent Strings: The data that the user chooses to send is written in a software-defined format, for example: 12, written as 31 32

Send network data (hexadecimal, only IoT supported).	0xD0+0xAC	Count+Strings	Send network data The corresponding instruction format is: EE D0 AC Count Strings (hexadecimal) FF FC FF FF FF Parameter format: Count: The number of bytes sent Strings: The data that the user selects to send is written in the appropriate format, for example: 12 13, written as 12 13
Save network settings (only IoT support)	0xD0+0xAA	not	Save the network settings The corresponding instruction format is: EE D0 AA FF FC FF FF FF
Playback of audio files	0x94	The audio file path	Audio file playback Parameter description: audio file path (not definite length): the audio file path set by the user; This command is mainly used to play the audio file set by the user, the screen according to the corresponding text The piece path plays the audio file.
The audio file stops	0x95	not	The audio file stops This command is mainly used to make an audio file change from playback state to stop
Volume adjustment	0x93	Value	Volume adjustment Parameter description: Value: Volume value in the range 0-100 This command is mainly used to adjust the volume of the speaker output.
Local/external AV input video	0x4B	Enable	Local/external AV input video Parameter description: Enable(1 byte): Local/external screen switching flag bit 0x00: Switch to local image display 0x01: Switch to AV Video Input Display This command is mainly used for switching between the local picture display and the AV video input display .
Show/hide the system keys	0x86	Show+x+y+type+Option+max_len	Show/hide the system keyboard Parameter description: Show(1 byte): 0 hidden, 1 show; x (2 bytes): the x-coordinate of the keyboard pop-up ; y (2 bytes): the y-coordinate of the keyboard pop-up ; type(1 byte): 0 keypad, 1 full keyboard; Option(1 byte): 0 normal characters, 1 password, 2 time settings; max_len (1 byte): keyboard input character length limit; This command is mainly used to show/hide the system keyboard.
Touch screen sensitivity adjustment instructions (for the capacitive screen only).	8A 5A A5	Xx	Touch screen sensitivity adjustment command (for capacitive screens only) Parameter description: XX (1 byte): XX value range is 0-7, the higher the higher the bit the more sensitive, can support a thicker cover This command is used to adjust the sensitivity of the touch screen, suitable for capacitive screens.

Play the local video	Set the local video path	0xB1+0x74	Screen_id+ Control_id+ Videos_Path	Command to set local video path: Command format: EE [B1 74 Screen_id Control_id Videos_Path] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Videos_ Path: Local video path (can be transferred in the command assistant Replace)

	Play The video	0xB1+0x70	Screen_id+ Control_id+ Number	Play command: Command format: EE [B1 70 Screen_id Control_id Number] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Number (2 bytes): the number of times the video is played circularly, 0 is the number of times the video is played circularly
Play the video on the USB flash drive	Set the USB flash drive video path	0xB1+0x74	Screen_id+ Control_id+ Videos_Path	Command to set U disk video path: Command format: EE [B1 74 Screen_id Control_id Videos_Path] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Videos_ Path: the video path in the USB flash disk (can be converted in the command assistant)
	Play The video	0xB1+0x70	Screen_id+ Control_id+ Number	Play command: Command format: EE [B1 70 Screen_id Control_id Number] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Number (2 bytes): the number of times the video is played circularly, 0 is the number of times the video is played circularly
Play the video from the SD card	Set the SD card video path	0xB1+0x74	Screen_id+ Control_id+ Videos_Path	Set SD card video command: Command format: EE [B1 74 Screen_id Control_id Videos_Path] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Videos_ Path: video path in SD card (can be converted in the command assistant)
	Play The video	0xB1+0x70	Screen_id+ Control_id+ Number	Play command: Command format: EE [B1 70 Screen_id Control_id Number] FF FC FF FF Parameter description: Screen_ Id (2 bytes): picture number Control_ Id (2 bytes): video playback control number Number (2 bytes): the number of times the video is played circularly, 0 is the number of times the video is played circularly

Pause/resume video playback	0xB1+0x72	Screen_id+ Control_id	Pause/resume playback command: Instruction format: EE [B1 72 screen_id control_id] FF FC FF FF Parameter description: screen_ Id (2 bytes): picture number control_ Id (2 bytes): video playback control number
Stop playing the video	0xB1+0x71	Screen_id+ Control_id	Stop command: Instruction format: EE [B1 71 screen_id control_id] FF FC FF FF Parameter description: screen_ Id (2 bytes): picture number control_ Id (2 bytes): video playback control number
Audio ID playback	0x90+0x01	Audio ID + number of plays	Audio playback command: Instruction format: EE [90 01 sounds_id time] FF FC FF FF Parameter description: sounds_ Id (2 bytes): audio ID Time (1 byte): Playback times (0 represents looping)
Audio pauses	0x90+0x02		Audio pause command: Instruction format: EE [90 02] FF FC FF FF
Audio resumes playback	0x90+0x03		Audio Recovery Playback Instructions: Instruction format: EE [90 03] FF FC FF FF
Audio stops playing	0x90+0x00		Audio stop playback command: Instruction format: EE [90 00] FF FC FF FF
Audio sequence playback	0x90+0x04+ 0x00+0x01	Audio ID + number of plays	Audio sequence playback command: Instruction format: EE [90 04 00 01 sounds_id time] FF FC FF FF Parameter description: sounds_ ID (2 bytes): audio sequence ID (use; separate sequence ID) Time (1 byte): Playback times (0 represents looping)
Lock down the system configuration	0x08+0xA5 +0x5A+0x5 F+0xF5	not	Lock System Configuration: This directive locks the system configuration (i.e. the settings saved by the power loss), and if successfully provisioned , the device returns EE 17 FF FC FF FF. Return instruction format: EE 17 FF FC FF FF FF
Unlock the system configura tion	0x09+0xD E+0xED+0 x13+0x31	not	Unlock the System Configuration: This directive is used to unlock the function of the system configuration lock (i.e. the settings saved by the power loss), and if the command is successfully configured, the device returns to EE 18 FF FC FF FF. Return instruction format: EE 18 FF FC FF FF FF

1.6 Basic Directive Collection Table

In the event that the configuration instruction set does not meet the display, the user can implement it through the basic instruction set, which is described in Appendix A. Users basically don't care about the basic instruction set, which can meet 99% of the user needs. The basic set of instructions is shown in Table 1.5 below.

Table 1.5 Basic Directive Collection

category	directives	Directive parameters	illustrate
Set the foreground color	0x41	Fcolor	The foreground color is used to specify the color of points, lines, circles, graphics, and text Fcolor (2 bytes): RGB color value
Sets the background color	0x42	Bcolor	The background color is used to specify colors such as screen clearance, text background color, and curved background Bcolor (2 bytes): RGB color value
Clear screen	0x01	not	Clear the screen according to the specified color Note: The screen clearance color depends on the background color setting and defaults to blue
Sets the text line and column spacing	0x43	Y_W+ X_W	Y_W (1 byte) of line spacing in points, take values 00 to 3F X_W (1 byte) column spacing in points, and values 00 to 3F
Set the text box	0x45	Enable+Width+Hight	Limit the text display area so that word wrap displays Enable (1 byte). 0x01: Turn on text box restriction enable, 0x00: Turn off text box restriction enable Width (2 bytes): The width of the text display box Hight (2 bytes): The height of the text display box
Sets the image filtering color	0x44	FilterColor	Colors in the picture are not displayed when they have the same filter color value FillColor (2 bytes): Filter color RGB value
Text display	0x20	X+Y+Back+Font+String	Displays text content of the specified size at any coordinate X (2 bytes): X-axis coordinate value in points Y (2 bytes): The Y-axis coordinate value in points Back (background color, 1 byte). 0x01: Turn on background color display 0x00: Turn off background color display Font (font library encoding, 1 byte). 0x00: 8x12 dot matrix (ASCII). 0x01: 8x16 dot matrix (ASCII). 0x02: 12x24 dot matrix (ASCII). 0x03: 16x32 dot matrix (ASCII). 0x04 12 x 12 dot matrix (GBK). 0x05: 16 x 16 dot matrix (GBK). 0x06: 24 x 24 dot matrix (GBK). 0x07: 32 x 32 dot matrix (GB2312). 0x08: 32 x 64 dot matrix (ASCII). 0x09: 64 x 64 dot matrix (GB2312). Strings: A string written by the user (high byte first) Note: The text font color is the same as the foreground color, and the background color is the background color

The cursor is displayed	0x21	Enable+X+Y+ Width+Hight	A cursor of the specified size is displayed at any coordinate Enable(1 byte): The cursor enables the signal 0x00: Close 0x01: On X (2 bytes): X-axis coordinate value in points Y (2 bytes): Y-axis coordinate value in points Width (1 bytes): Cursor width Hight (1 byte): Cursor height Note: The color of the cursor is opposite to the color from the beginning of the current cursor area, and the blink time is silently 1 second
Full-screen picture display	0x31	Image_ID+MaskEn	A picture is displayed on full screen with fixed (0,0) coordinates at the starting position Image _ID (2 bytes): Image number MaskEn (1 byte): Filtering enabled 0x00: The color is not filtered ; 0x01 Perform color filtering Note: The filtered color depends on the setting of the picture filter color, and the resolution of the downloaded picture cannot exceed the resolution of the current screen, otherwise it cannot be displayed.
Area picture display	0x32	X+Y+ Image_ID+MaskEn	Displays a picture at any coordinate X (2 bytes): X axis coordinate value in points Y (2 bytes): The Y axis coordinate value in points Image _ID (2). bytes): image number MaskEn (1 byte): filtering enabled 0x00: color is not filtered; 0x01 Perform color filtering Note: The filtered color depends on the settings of the filtered color.
Picture clipping display	0x33	X+Y+Image_ID+Image_X+Image_Y+Image_W+ Image_H+MaskEn	Pictures cut from a picture are displayed at any coordinates X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point Image_ ID (2 bytes): picture number to be cut Image_ X (2 bytes): X coordinate of the starting point of the cut picture Image_ Y (2 bytes): Y coordinate of the starting point of the cut picture Image_ W (2 bytes): Cut width Image_ H (2 bytes): cutting height MaskEn (1 byte): filter enable 0x00: color does not filter; 0x01 Perform color filtering Note: The filtered color depends on the filter color setting

Animation display	0x80	X+Y+FlashImgae_ID + Enable+Playnum	Show GIF animation at any coordinates X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point FlashImage_ ID (2 bytes): animation number Enable (1 byte): play enable 0x00: Close animation playback; 0x01: Enable animation playback Playnum (1 byte): playback times 0x00: Repeat playback; 0x01~0xFF: Stop after playing the specified number of times Serial screen upload EE 02 FF FC FF FF indicates the end of animation playback Note: Animation only supports * gif format, and only one animation can be played for one screen. If you want to play multiple gif animations at the same time, as well as animation pause, stop, play up/down frames and other functions, please use the configuration animation control, as shown in the following configuration command table.
Foreground scenery painting spots	0x50	X+Y	A dot is displayed on the screen, and the color of the dot depends on the foreground color setting X (2 bytes): The X-axis coordinate value in points Y (2 bytes): The Y-axis coordinate value in points
Background color point of view (delete point)	0x58	X+Y	A point is displayed on the screen, and the color of the point depends on the background color setting X (2 bytes): The X-axis coordinate value in points Y (2 bytes): The Y-axis coordinate value in points Note: It is mainly used with foreground painting points and can be used to clear the points of foreground painting
Draw the line	0x51	X0 +Y0+X1+Y1	Connects the specified two coordinate points X_0 (2 bytes): The coordinate value of the start point of the X-axis of the line, in points Y_0 (2 bytes): The coordinate value of the start point of the Y axis of the straight line, in points X_1 (2 bytes): The coordinate value of the end point of the line X-axis in points Y_1 (2 bytes): The coordinate value of the end point of the line Y axis in points Note: The color value of the line depends on the foreground color setting
Connect with foreground color according to evenly spaced X coordinate	0x59	X_0 +Xspace+Y_0 +...+Y n	Quickly connects multiple equally spaced X-axis coordinates that you specify X (2 bytes): The X-axis coordinate value in points Xspace (2 bytes): The X-axis interval value in points, with the fixed spacing of adjacent front and back points being Xspace Y (2 bytes): The Y-axis coordinate value in points Note: Since the command does not need to send the X coordinate value, the command sending time is saved by half and the drawing speed is doubled. The color value of the line depends on the foreground color setting

Offset by coordinate by using foreground color to wire	0x75	(X,Y) $_0$+(X1o, Y1o) +.. +(Xno,There)	Quickly connect multiple specified offset coordinate points with foreground color X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point X1o (1 byte): X-axis offset in point Y1o (1 byte): Y-axis offset in point Note: (X, Y) is the absolute coordinate of the first point, and each subsequent point is determined by the previous one The absolute coordinates of the point plus the current offset. The highest bit of the offset is a sign bit, 0 represents a positive offset, 1 represents a negative offset, and the maximum offset value is ±127 points.
Offset by coordinate Wire with background color	0x76	(X,Y) $_0$+(X1o, Y1o) +.. +(Xno,There)	Quickly connect the specified multiple offset coordinate points with the background color X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point X1o (1 byte): X-axis offset in point Y1o (1 byte): Y-axis offset in point Note: (X, Y) The first point is the absolute coordinate, and each subsequent point is determined by the previous one The absolute coordinates of the point plus the current offset. The highest bit of the offset is a sign bit, 0 represents a positive offset, 1 represents a negative offset, and the maximum offset value is ±127 points.
Lines the specified coordinate point with foreground color	0x68	(X,Y) $_0$+(X,Y)$_1$ ...+(X,Y)$_n$	Connect multiple specified coordinate points with foreground color X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point Note: The color value of the line depends on the foreground color setting
Lines the specified coordinate points with a background color	0x69	(X,Y) $_0$+(X,Y)$_1$ ...+(X,Y)$_n$	Connect multiple specified coordinate points with the background color X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point Note: The color value of the line depends on the background color setting
Draw hollow circles	0x52	X_0 +Y_0 +R	Draw a hollow circle with radius R at any coordinate X0 (2 bytes): X coordinate value of the center in point Y0 (2 bytes): the Y coordinate value of the center of the circle in points R (2 bytes): radius of the hollow circle Note: The color value depends on the foreground color setting
Draw a solid circle	0x53	X_0 +Y_0 +R	Draw a solid circle with radius R at any coordinate X0 (2 bytes): X coordinate value of the center in point Y0 (2 bytes): the Y coordinate value of the center of the circle in points R (2 bytes): radius of the hollow circle Note: The color value depends on the foreground color setting

Draw a circular arc	0x67	X_0 +Y_0 +R +EA+SA	Draw an arc at any coordinate X0 (2 bytes): X coordinate value of the center in point Y0 (2 bytes): the Y coordinate value of the center of the circle in points R (2 bytes): radius of circle EA (2 bytes): end angle SA (2 bytes): starting angle Note: The clock is 0 degrees at 3 o'clock, calculated counterclockwise; The color value depends on the foreground tone palette setting

Draw a hollow rectangle	0x54	X_0 +Y_0 +X_1 +Y_1	Drawing a hollow rectangle at any position can also be used for local screen clearing X_0 (2 bytes): The X-coordinate value of the upper-left corner of a hollow rectangle in points Y_0 (2 bytes): The Y-coordinate value of the upper-left corner of the hollow rectangle in points X_1 (2 bytes): The X-coordinate value of the lower-right corner of a hollow rectangle in points Y_1 (2 bytes): The Y-coordinate value of the lower-right corner of a hollow rectangle in points Note: The color value depends on the foreground color setting
Draw a solid rectangle/partial screen	0x55	X_0 +Y_0 +X_1 +Y_1	Draw a solid rectangle anywhere X_0 (2 bytes): The X coordinate value of the upper-left corner of a solid rectangle in points Y_0 (2 bytes): The Y-coordinate value of the upper-left corner of a solid rectangle in points X_1 (2 bytes): The X coordinate value of the lower-right corner of a solid rectangle in points Y_1 (2 bytes): The Y-coordinate value of the lower-right corner of a solid rectangle in points Note: The color value depends on the foreground color setting
Draw a hollow ellipse	0x56	X_0 +Y_0 +X_1 +Y_1	Draw a hollow ellipse at any position X0 (2 bytes): the leftmost X coordinate value of the hollow ellipse in points Y0 (2 bytes): the top Y coordinate value of the hollow ellipse in points X1 (2 bytes): X coordinate value at the rightmost end of the hollow ellipse in points Y1 (2 bytes): the lowest Y coordinate value of the hollow ellipse in points Note: The color value depends on the foreground color settings
Draw a solid ellipse	0x57	X_0 +Y_0 +X_1 +Y_1	Draw a solid ellipse at any position X0 (2 bytes): the leftmost X coordinate value of a solid ellipse in points Y0 (2 bytes): the top Y coordinate value of a solid ellipse in points X1 (2 bytes): X coordinate value of the rightmost end of a solid ellipse in points Y1 (2 bytes): the lowest Y coordinate value of a solid ellipse in points Note: The color value depends on the foreground color settings

Clear the layer	0x05	Layer	Clears the contents of a layer Layer (1 byte): The layer written (value range 0 to 1).
Automatically clears the current layer when switching	0x06	Enable	Set whether to automatically clear the current user layer when switching the screen Enable (1 byte): enable signal 0x01: Clear layer automatically 0x00: Disable layer cleanup By default, all contents in the user's two layers are cleared when the screen is switched.
RTC display settings (Hardware support required).	0x85	Cmd+DisMode+Font +Color +X+Y	RTC display settings Cmd (1 byte): parameter configuration BIT0: enable signal 0: RTC off 1: RTC on BIT7-BIT1: Reserved DisMode (1 byte): display mode 0x00: Format HH: MM: SS 0x01: Format 20XX-MM-DD HH: MM: SS Font (1 byte): font selection 0x00: 8x12 lattice (ASCII) 0x01: 8x16 lattice (ASCII) 0x02: 12x24 lattice (ASCII) 0x03: 16x32 lattice (ASCII) 0x04 12 x 12 dot matrix (GBK) 0x05: 16 x 16 dot matrix (GBK) 0x06: 24 x 24 dot matrix (GBK) 0x07: 32 x 32 dot matrix (GB2312) 0x08: 32 x 64 dot matrix (ASCII) 0x09: 64 x 64 dot matrix (GB2312) Color (2 bytes): display color X (2 bytes): X axis coordinate value in point Y (2 bytes): Y-axis coordinate value in point It is recommended that users directly use the clock control to complete RTC time display and calibration.

2. Detailed description of the configuration instruction set

The following sections describe the configuration instruction set functions and usage, and the driver libraries involved can be viewed in the reference program examples. Sample programs can be downloaded from the website.

2.1 Handshake

Instruction format: EE [04] FF FC FF FF

Device returns: EE 55 FF FC FF FF

The handshake command is mainly used to determine whether the device is powered on and initialized, whether the communication is normal and whether it is online. After sending the command, the device returns 55 to indicate a successful handshake.

2.2 Reset report

Instruction format: None

Device return: EE 07 FF FC FF FF

Once the device is powered on, unexpectedly rebooted, or the monitoring chip is reset, the relevant data will be uploaded immediately to inform the user that the device has been reset. After the user host detects an unexpected reset of the device, it needs to control the program to re-initialize the execution from scratch.

2.3 Reset the device

Instruction format: EE [07 35 5A 53 A5] FF FC FF FF

Device return: EE 07 FF FC FF FF

The host resets the device through serial port instructions during operation. It is recommended that you increase this command when the host initializes the device so that the device resets after the host is unexpectedly reset.

2.4 Get the device version

Instruction format: EE [FE 01] FF FC FF FF

Device Return: EE FE 02 16 00 02 04 01 01 04 FF FC FF FF

Parameter description: The device returns the firmware version number of the screen 2.22.1025.260, where the indication of the instruction is hexadecimal, and the corresponding conversion is as follows: 02 conversion is 2, 16 conversion is 22,00 02 Refers to the type number of the screen, refers to the commercial type, 04 01 conversion is 1025, 01 04 conversion is 260 Other types of screens also get the version number in this way when reading the corresponding firmware version.

2.5 Backlight adjustment

Instruction format: EE [60 Light_ level] FF FC FF FF

Parameter description: Light_level (1 byte) : backlight brightness value This instruction is mainly used for adjusting the brightness of the liquid crystal backlight, and the value range is 00H~FFH. 00H indicates the brightest, FFH

Indicates the darkest backlight with a total of 255 levels of brightness adjustment. If there is no operation action on the screen for a certain period of time, it is recommended that the user reduce the backlight brightness to about 30% to improve the backlight life.

2.6 Automatic screensaver mode

Command format: EE [77 Enable BL_ON BL_OFF BL_ON_Time] FF FC FF FF

Parameter description: Enable(1 byte): Enable signal

0x00: Turn off power-saving mode 0x01: Turn on power-saving mode

0x03: Turn on battery saver mode and turn on backlight notifications

BL_ON (1 byte): The brightness value of the backlight when activated by touch

BL_OFF (1 byte): The brightness value of the backlight after entering screensaver mode

BL_ ON_ Time (2 bytes): the time to enter screensaver mode when there is no touch action (unit: 1 second)

This command is mainly used to set the backlight brightness value when the screensaver mode is activated and the screensaver mode is entered. The power-saving mode can not only prolong the backlight life of the LCD screen, but also reduce the external heat emitted by the LCD.

2.7 Buzzer control

Instruction format: EE [61 Time] FF FC FF FF

Parameter description: Time (1 byte): The time of the buzzer signal, in 10ms

This command is used for the control of the buzzer, and different frequencies are heard by setting the Time parameter. Generally touch time is set to 100ms.

2.8 Configure the touch screen

Instruction format: EE [70 Cmd] FF FC FF FF

Parameter description: Cmd (1 byte): Configuration parameter

BIT0: 1 means that the touch screen is on, and 0 means that the touch screen is off; BIT1: 1 means that the buzzer automatically rings when touched, and 0 means that it does not ring; BIT4~BIT2 : Touch coordinate value upload method

000: Indicates that coordinates are uploaded only once when the touch screen is pressed

001: Indicates that the touch screen is pressed until the coordinates are uploaded once after release

010: When the touch is pressed all the time, the coordinates are uploaded once every 100ms, and the coordinates are also uploaded once when released

011: Indicates that the coordinates are uploaded once when the touch screen is pressed and released

100: Turns off touch coordinate upload

BIT5: 0 means that within 4 seconds of continuously clicking on a non-touch area 20, the screen enters touch calibration mode, 1 means that this function is disabled;

BIT7-BIT6: Reserved

Touch coordinate value upload format when pressed, upload format: EE 01 X Y FF FC FF FF

Uploading format when releasing: EE 03 X Y FF FC FF FF,

X and Y are both 2 bytes, and the high byte command includes touch enable, open/close buzzer and coordinate value upload mode.

As shown in Figure 2-1 below, if the touch upload format is configured as "000", after the user presses the position of the screen (50100), the device uploads data: EE 01 [00 32 00 64] FF FC FF FF.

The user host can determine whether the current touch is valid by judging whether the received coordinates (X, Y) are within the valid touch area. The touch pressure value is sampled and calculated many times by the device itself, so the user does not need to perform secondary calculation.

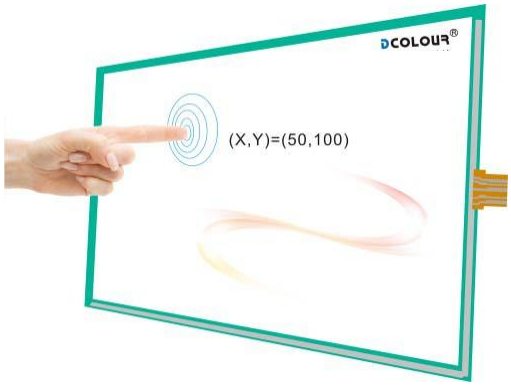

Figure 2-1 Touch screen work description

2.9 Touch screen calibration

Instruction format: EE [72] FF FC FF FF

Parameter description: None This command is used for the calibration of the touch screen. The equipment is calibrated at the factory and does not need to be calibrated again. Send the calibration life

After the order, tap the corresponding cursor as prompted by the screen, as shown in Figure 2-2 . After clicking on it, the device will indicate whether the calibration was successful, otherwise it will need to be recalibrated. The user can also send instructions for calibration via the host computer software.

In addition, the user in the non-touch area of a point within 4 seconds of a quick click 20, the system will automatically enter the touch mode, after the calibration is automatically returned to the current display. This feature is more suitable for touch calibration in the field.

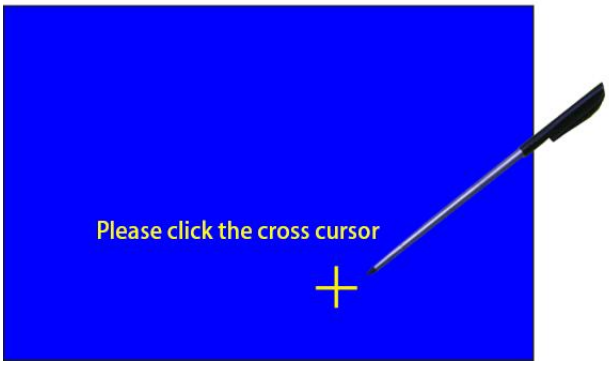

Figure 2-2 Touch screen calibration schematic

2.10 Touchscreen experience

Instruction format: EE [73] FF FC FF FF

Parameter description: None

This directive is a test command. As shown in Fig. 2-3, after the user presses the touch, a red solid circle will be displayed at the corresponding coordinates, which is convenient for the user to intuitively test the quality of the touch screen and experience the accuracy of the touch value. After the device is successfully connected to the PC, the user can tap "Experience Touch" in the VisualTFT software toolbar to experience the sensitivity and accuracy of the touch.

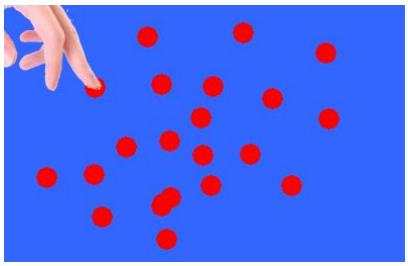

Figure 2-3 Touch Experience Renderings

2.11 Set the baud rate

Instruction format: EE [A0 Baudset] FF FC FF FF

Parameter description: Baudset (1 byte): baud rate sequencing, in bps

0x00: 1200 0x01: 2400 0x02: 4800

0x03: 9600 0x04: 19200 0x05: 38400

0x06: 57600 0x07: 115200 0x08:1M

0x09:2M 0x0A :2187500x0B : 437500

0x0C :875000 0x0D : 921800

This instruction is primarily used in the configuration of baud rates in the range of 1200-2Mbps. The new baud rate value is saved by power failure. The user can configure the new baud rate directly through the VisualTFT's Debug Assistant, as shown in Figure 2-4.

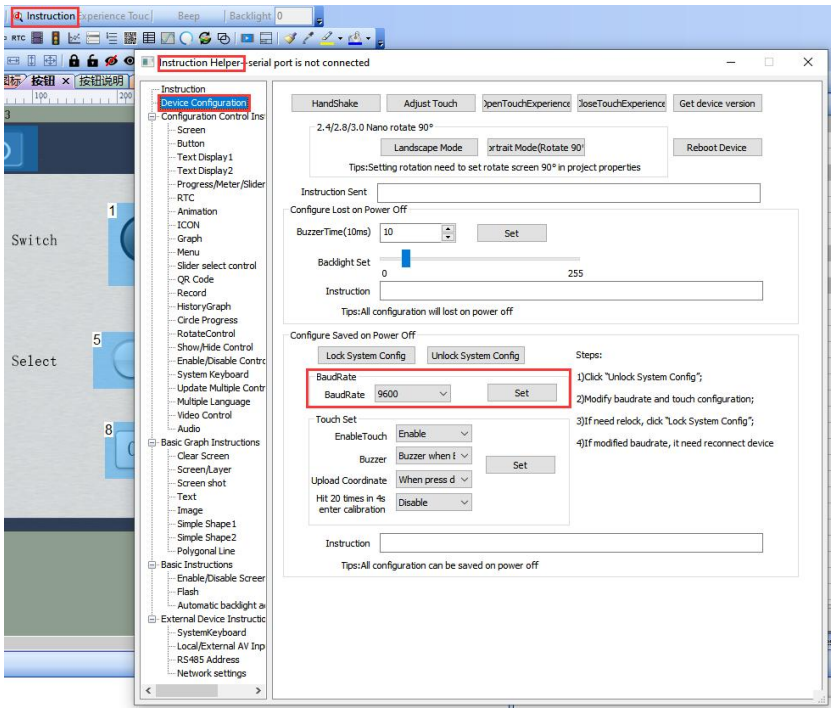

Figure 2-4 Baud rate settings

2.12 Switch screens

Instruction format: EE [B1 00 Screen_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Although the button control can set which screen to automatically switch to after pressing a button, there are some occasions that also require a host

After making a logical judgment, control the target screen display. Program reference code:

```
{SetScreen(2);  // Switch to Screen_id =2 of the picture
}
```

2.13 Use animation effects to switch screens

Instruction format: EE [B1 05 SCREEN_ID (2Byte) EFFECT (1Byte) AREA_EN (1Byte) AREA_LEFT (2Byte) AREA_RIGHT (2Byte) AREA_TOP (2Byte) AREA_BOTTOM (2Byte)] FF FC FF FF

Parameter description: SCREEN_ID (2Byte): The ID of the
target screen
EFFECT(1Byte): Toggle effect, 0 disable effect, 1 left to right 2 from right to left 3 from top to bottom 4 from bottom to above
AREA_EN (1 Byte): 0 full-screen effect, 1 area effect AREA_LEFT (2Byte) area left distance, pixel
AREA_RIGHT (2Byte) area right distance
AREA_TOP (2Byte) area upper distance
AREA_BOTTOM (2Byte) area lower distance

2.14 Reading the screen

Instruction format: EE [B1 01] FF FC FF FF
Parameter description: None
This directive is mainly used to get the ID value of the current screen . In reliability applications, the host sends this command to determine whether the screen is successfully switched.
Instruction return format: EE B1 01 Screen_id FF FC FF FF
Screen_id (2 bytes): the number of the current screen

2.15 Button control ID value uploaded

Button controls serve five purposes: toggle screens, switch descriptions, custom keys, custom commands, and pop-up menus.
(1) Switch screens. Switching the screen means that the screen automatically switches to another screen display after pressing a button. For example, as shown in Figure 2-5, the user wants to switch to the screen after clicking the "Dialog Box" icon to switch to the screen Screen1 display
First set the entire button to the touch area, then select the Touch Purpose → Switch Screen in the Properties window; The target image → Screen1, and finally run the "Virtual Serial Interface" for effect verification. For detailed operation of the host computer, please refer to the "Dacai Configuration Serial Port Screen Quick Start Manual".

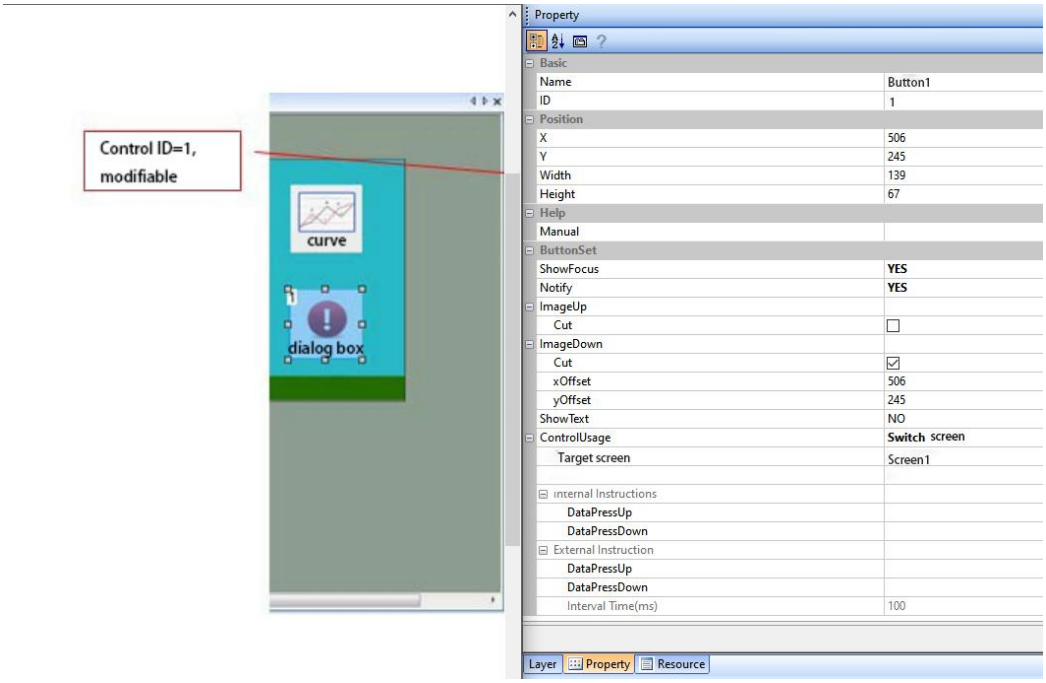

Figure 2-5 Button Control-Toggle Screen Configuration Diagram

Screen toggle button upload instruction format:

EE [B1 11 Screen_id Control_id Control_type Subtype Status]FF FC FF FF

Parameter description: Screen_id (2 bytes): Screen ID

Control_id (2 bytes). : Control ID number

Control_type (1 byte): A fixed value of 0x10, expressed as a button

control subtype (1 byte): A fixed value of 0x00 indicates that the

current button function is Switch Screen Status (1 byte): reserved

The switch description. Switch description refers to the button as a button function as a press or pop-up switch, divided into 4 styles, the content is as follows:

Transients. After pressing, the switch automatically bounces up, similar to the function of a tactile switch; Switch. After pressing, the switch changes from bounce up to press or from press to bounce up, similar to the function of a lock switch; Assertion. The switch can only be changed from bounce up to press;

Reposition. The switch can only be changed from pressing to bounce;

For example, as shown in (1) (2) in Figure 2-6, the user needs to use the "Stop Running" button as a switch function, and first set the entire button to the touch area , then select Touch Usage → Switch Description in the Properties window; Operation style → switch; Press the Picture → button to select the UI, and finally run the "Virtual Serial Screen" to verify the effect.

After running the virtual serial port screen, you can see the effect of pressing the "Stop Running" button, as shown.

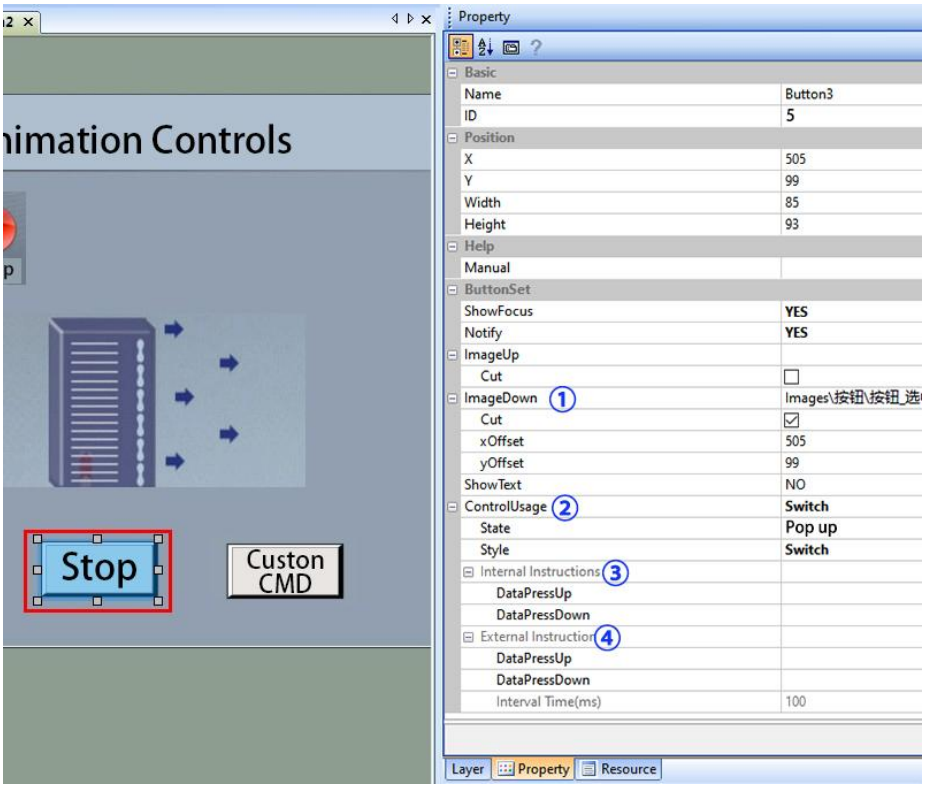

Figure 2-6 Button Control-Switch Type Configuration Diagram

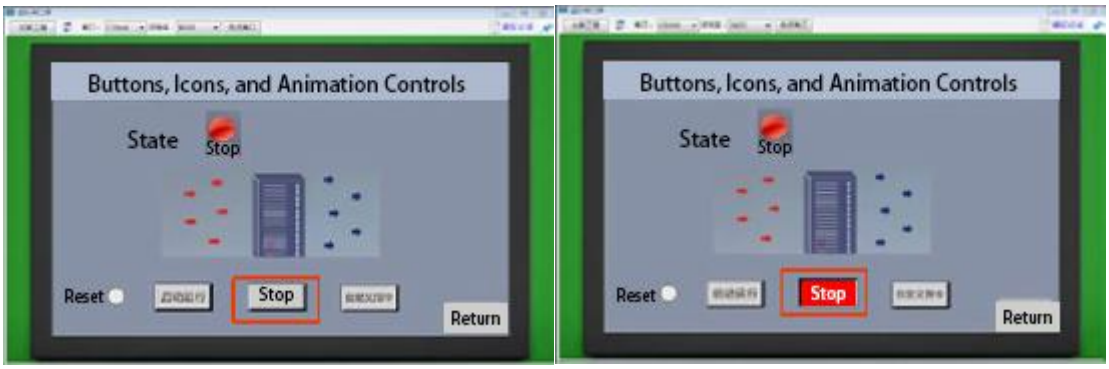

Figure 2-7 Run Virtual Serial Port Screen View button pressed

Switch type button control upload format:

EE [B1 11 Screen_id Control_id Control_type Subtype Status]FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The button control ID number

Control_type (1 byte): A fixed value of 0x10, expressed as a button

control subtype (1 byte): A fixed value of 0x01 indicates that the current button function is a switch description Status (1 byte): button state

0x00: The button changes from pressed to bounced

0x01: The button changes from bouncing to pressing

The user microcontroller can determine which button of the current screen is pressed or bounced up by parsing the instructions. In the figure (3) (4) is the extension instruction, indicating that while the button is pressed, additional instructions can also be output internally/externally, in detail

The introduction can be described in this section 4.5 and 4.6. Custom keys. Custom keys are designed by the user on the background image in advance for the desired keyboard, and then each one at a time

Key values are set to match, such as some keys are numbers, some are delete functions, some are Enter functions, and so on. After the setting is completed, the keyboard can be bound to the text control of the current screen, that is, the value entered by the user's keyboard can be automatically displayed in the text box.

For example, as shown in Figure 2-8, if the user needs to use the "Number 9" button as a key value of 9, the entire button is first set to the touch area. Then select Custom key values in the Properties window: Touch Usage →; Type→ characters; Character → is written to 9, and finally the "virtual serial port screen" is run for simulation test. In addition, the key-value type can select "Enter", "Clear", "In addition to characters" Backspace", "Esc", or "Shift" special function keys.

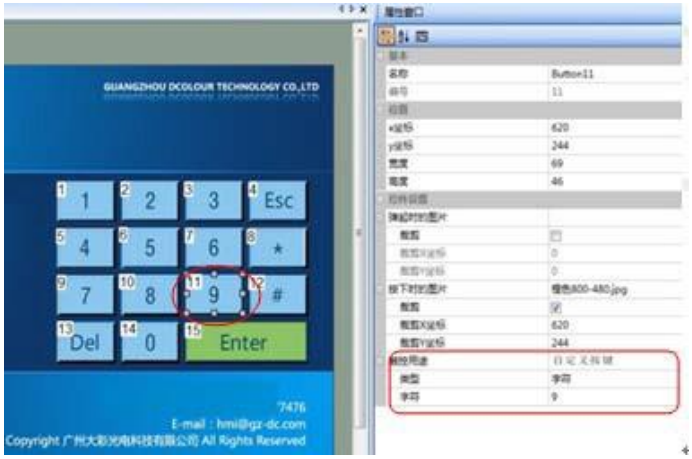

Figure 2-8 Button Control - Custom Key Configuration Diagram

Custom keystroke upload format:

EE【B1 11 Screen_id Control_id Control_type Subtype Key_value Status】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type: A fixed value of 0x10, expressed as a button control

Subtype: A fixed value of 0x02 indicating that the current button function is a custom key. Key_value: A user-defined key value expressed in ASCII codes

Status (1 byte): Reserved

Custom instructions. Users can set the device to upload its own defined data strings after pressing a button.

As shown in 2-9, the "Customize" button pops up, and the screen sends the command: 02;

Send when pressed: 01.

Tip: Custom instructions cannot contain FF FC FF FF combination characters, otherwise they will conflict with regular instructions and cause errors.

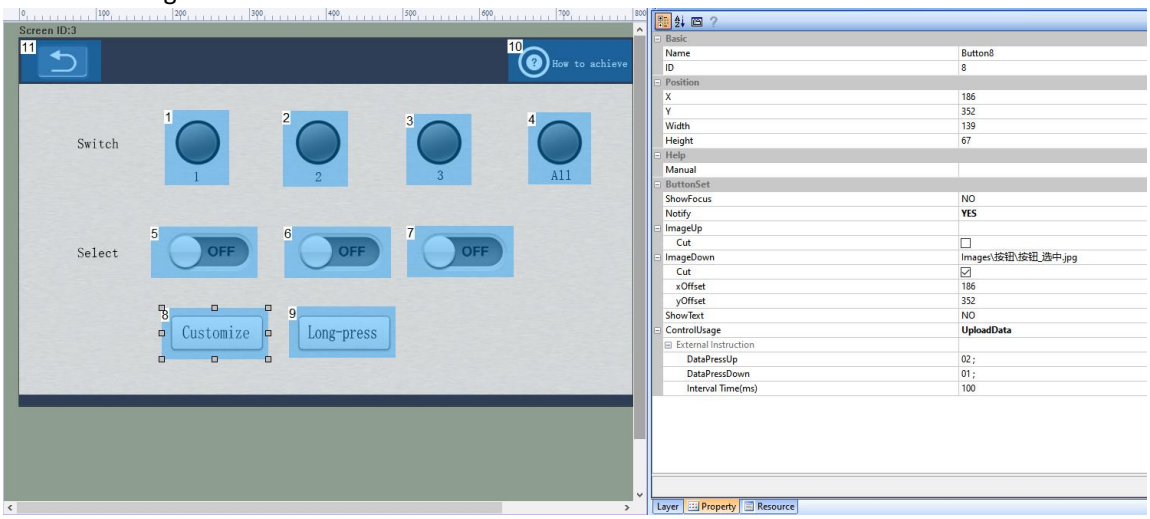

Figure 2-9 Custom Directive Data

Pop-up menu. This instruction is mainly used with the drop-down menu, and the detailed use can refer to the settings drop-down menu content of this section .

2.16 Set the button pop-up or pressed state

The command mainly implements a button in the screen to forcibly press down or bounce up, that is to say, in addition to clicking the touch setting button press or bounce up, the microcontroller can also send commands to set the button state.

Instruction format： EE [B1 10 Screen_id Control_id Status]FF FC FF FF

Parameter description: Screen_id (2 bytes): Screen

number Control_id (2 bytes): Button control

number Status (1 byte).): Button status

0x00: The setting button changes from pressed to bounced

0x01: The setup button changes from pop-up to pressed

This command is mainly used to turn the "press" button into "bounce" or the "bounce" button into "press." " status. In some cases, there is a mutual exclusion between the buttons, when a button is

pressed, another button must bounce up, you can use the command to achieve.

For example, as shown in Figure 2-10, after the user presses the "Start to Run" button, he needs to put the "Stop Running" button (screen ID 2, control ID 4) from "press" to "bounce up." "status, then the microcontroller sends the instruction: EE B1 10 00 02 00 04 00 FF FC FF FF。

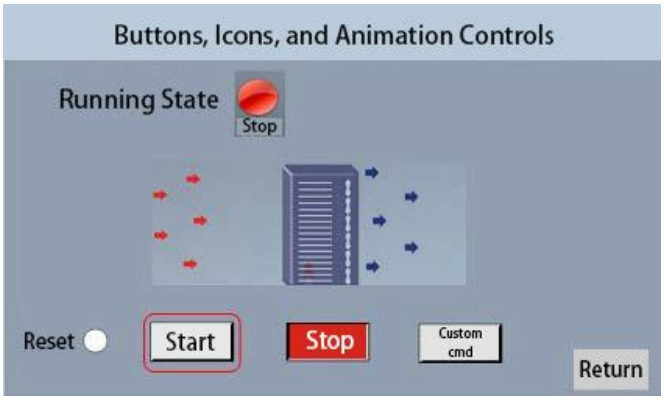

Figure 2-10 Setting the button press and pop-up states

Program reference code:

```
{
  . . ..                          Detect that the start run
  SetBuTtonValue(2,4, 0)；  // Set the Stop Running button on screen 2 and control ID
 }
```

2.17 Internal instructions implement button mutex and status display

The so-called internal instructions are that the user can set that after pressing a key, some command strings are also executed inside the screen to achieve some status and logic display, without the participation of external microcontrollers, saving program code.

As shown in Figure 2-11, if the user presses the Start Run button, the Stop Run button (screen ID is 4, the control ID is 2) to bounce up, and it can be configured directly in the PC. Properties window selection: Inward Directive - When pressed→EE B1 10 00 04 00 02 00 FF FC FF FF, so that after pressing the "Start Running" button, the screen will send a pop-up "Stop Running" command to the inside to enable the button to automatically pop up, without the participation of the external microcontroller described in Section 4.4.

Of course, in addition to configuring pressing "Start Run" and popping up "Stop Running", you can also enter a number of other instructions in the internal command box, such as starting gif animation, changing the run state flag, etc. Multiple instructions are separated by semicolons.

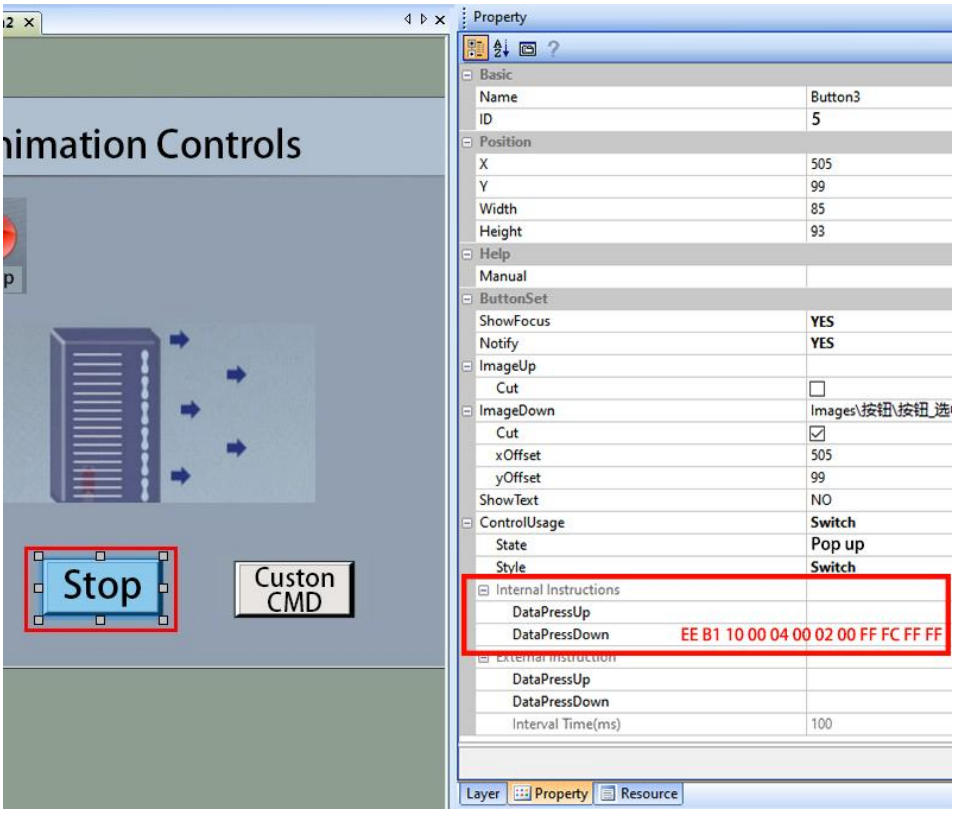

Figure 2-11 Button mutex configuration

2.18 External Instruction Output

The so-called external instructions are that the user can set the user-defined instruction string to be executed externally on the screen after pressing a key, similar to the custom instructions in the 4.3 chapter button control.

2.19 Read button control state

Instruction format: EE [B1 11 Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): Control number This directive is primarily used to query whether a button is currently "pressed" or "popped up" State.

Return instruction format: EE B1 11 Screen_id Control_id Control_type Subtype Status FF FC FF FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type (1 byte): A fixed value of 0x10, expressed as a button

control subtype (1 byte): A fixed value 0x01, indicating that the button describes the type of switch

Status (1 byte): The button state

0x00: The button is popped up

0x01: The button is pressed

2.20 Update text control values

There are three ways to update text: user host input, pop-up system keyboard input, and custom keyboard input. 1. User host input. The data displayed on the screen comes from the user's microcontroller input.

Instruction format: EE [B1 10 Screen_id Control_id Strings] FF FC FF FF

parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Strings: A string written by the user This directive is primarily used for text display.

Since the system has allocated memory addresses to all text variables, when the screen returns to the text interface again from other interfaces, the text data is still saved, and there is no need to refresh the data again. With the text control, the user first configures the relevant parameters on the upper computer, such as font size, foreground color, background color, and text input method, as shown in Figure 2-12. Then the host only needs to directly write the changed data to the corresponding ID.

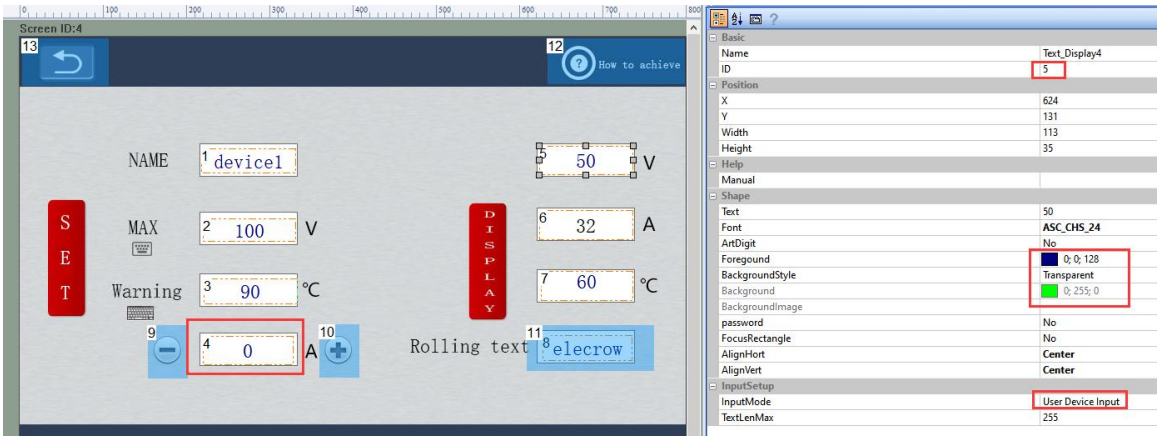

Figure 2-12 Text Control Parameter Configuration

Pop up system keyboard input.

When the text box is clicked, the built-in keyboard automatically pops up, and after the user has entered the required characters, click OK, and the device will display the entered characters in the text box and upload the ASCII code of the characters to the host.

For example, the user presets the text control and selects "pop-up keyboard input" as entered, as shown in Figure 2-13. Click on the text box, the screen will automatically pop up the system keypad, enter "123" and click Enter, the entered number will be automatically displayed in the text box, while uploading the input ASCII code, upload in the same format as the "Read Text Control Values" return format described below. The user microcontroller parses the uploaded instructions to know the entered data.

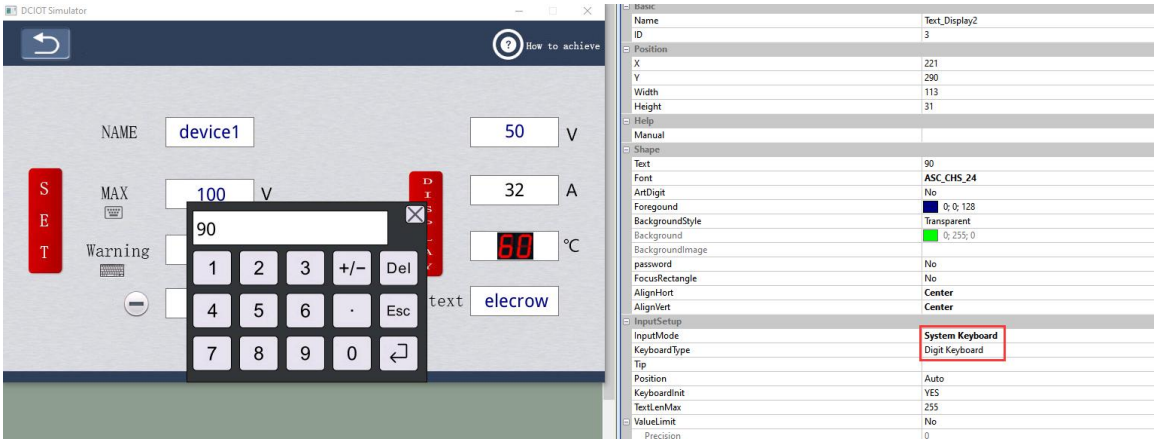

Figure 2-13 Eject the system keypad

If the user needs to enter Chinese characters, simply select the keyboard type as full keyboard, and then click the Chinese and English switch keys on the keyboard, as shown in Figure 2-14.

Note: Due to the display area, full keyboard and input method are only supported for sizes above 3.5 inches (excluding 3.5 inches).

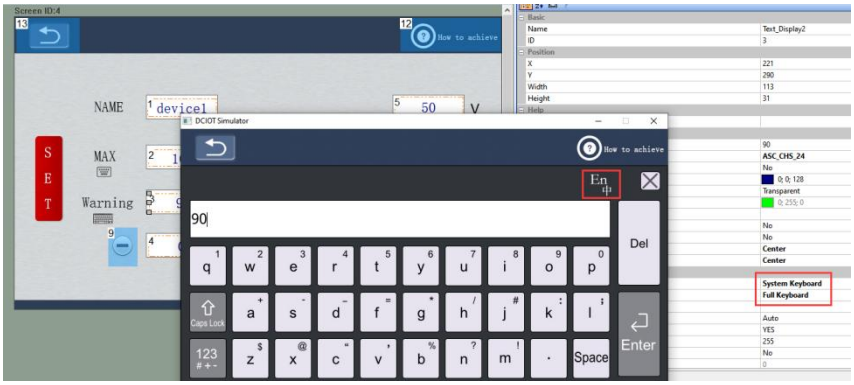

Figure 2-14 Eject system full keyboard and Chinese and English input

Customize key input. The data in the text box comes from the keyboard input in the same screen. As shown in Figure 2-15, if the user needs to input the password, first use the user-defined key value function of the button control to define each button as the corresponding ASCII character (see Figure 2-8), then place a text control in the password box, and set the attribute input mode to "user-defined key input". In this way, after clicking the text box, the cursor will automatically flash on the interface, and then click the right keyboard to enter, the corresponding number will be automatically displayed in the text box, and the input character ASCII code will also be uploaded to the host computer.

Prompt: After configuration, it is recommended to run the "Virtual Serial Port Screen" to view the information uploaded on the screen after pressing each button.

Figure 2-15 Custom key input

2.21 Clear text control content

Command format: EE [B1 10 Screen_id Control_id] FF FC FF FF

Parameter description:

Screen_ Id (2 bytes): picture number

Control_ Id (2 bytes): control number

This command is mainly used to clear the value of the current text control.

2.22 Read the text control values

Instruction format: EE [B1 11 Screen_id Control_id] FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): Control Number

This directive is primarily used to get the numeric value of the current text control.

For some important parameters, the user can use the command to obtain the value of the text control to re verify.

Return instruction format: EE B1 11 Screen_id Control_id Control_type Strings FF FC FF FF FF

Return parameter: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type(1 byte): A fixed value of 0x11, expressed as a text control Strings (not necessarily long): The currently displayed text value with 1 0x00 appended to the text

2.23 Set the cursor focus

Instruction format： EE [B1 02 Screen_id Control_id Enable]FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Enable (1 byte): 0x00 means off, 0x01 means turn on the display; This directive mainly ports the cursor symbol to the specified text control display, which more intuitively reminds the user to enter the text.

Tip: Valid only in custom keyboards.

2.24 Set text control flickering

Instruction format： EE [B1 15 Screen_id Control_id Cycle]FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Cycle (2 bytes): Blink period (10 millisecond units), 0 means no flicker This instruction is mainly used to implement text control alternate blinking, Cycle Stop flashing when the value is 0.

2.25 Set the text control scroll speed

Instruction format： EE [B1 16 Screen_id Control_id Speed]FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Speed (2 bytes): Text scrolling speed (pixels moving per second), 0 means no scrolling This directive is mainly used to implement text scrolling display, scrolling from right to left by default A Speed value of 0 stops scrolling.

2.26 Set the background color of the document control

Instruction format: EE [B1 18 Screen_id Control_id BK_Color] FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

BK_Color (2 bytes): Background color RGB color value This directive is primarily used to add text controls to the background display during runtime to achieve the effect that the text is selected or highlighted. How do I get background/foreground RGB color values? The user can arbitrarily create a new screen, draw a rectangle, and place a rectangle

The color is set to its desired color, and then the mouse is placed on the rectangle, and the lower left corner of the software displays the RGB565 color value of the current screen mouse point , as shown in Figure 2-16.

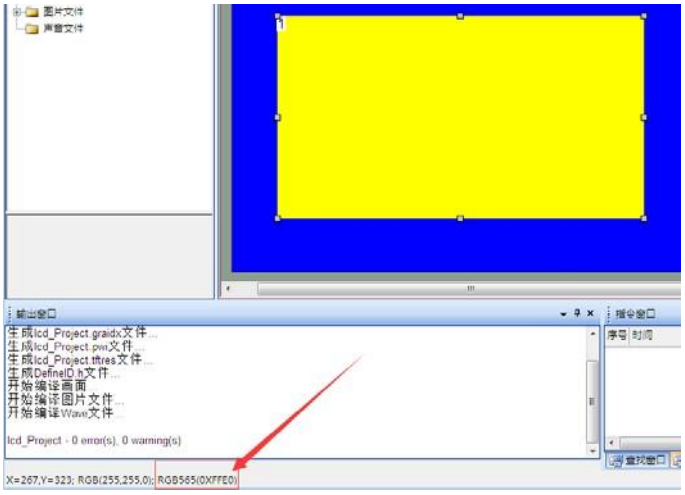

Figure 2-16 Foreground/background color extraction

2.27 Set text background transparency

This directive is primarily used to suppress the display of a text control with a background

Instruction format: EE [B1 17 Screen_id Control_id] FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

2.28 Set the text control foreground

This command is mainly used to change the color display of text controls during operation, to achieve some alarms, highlighting, etc. Instruction format: EE [B1 19 Screen_id Control_id FORE_Color] FF FC FF FF parameter description:

Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

FORE_Color (2 bytes): The foreground RGB color value

2.29 Formatted text display

Instruction format：EE 【B1 07 Screen_id Control_idSign Fill_zero Value】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Sign (1 byte): The data type, which is:

Unsigned int: unsigned integer, 0x00;

Int: signed integer, 0x01;

Float: a single-precision floating-point number, 0x02;

Double: double-precision floating-point number, 0x03;

Fill_zero (1 byte): the number of decimal places, if you select less than 0, you need to add 0x80;
Value (4 bytes): The added data, expressed as a hexadecimal number;

This instruction is mainly used to input text according to corresponding conditions and display text data according to corresponding restrictions.

2.30 Text Control Numeric Increment Adjustment Instructions

Instruction format：EE【B1 1A Screen_idControl_idOptionDeltaMin_LimitMax_Limit】FF FC FF Ff

Parameter description: Screen_id (2 bytes): screen number
Control_ id (2 bytes): The text control number
Option: bit0-0 decreases, 1 increases; bit1-0 does not cycle, 1 cycles. Add the Do Not Loop instruction to 0x01; Increase the loop instruction to 0x03; Reduce the no-loop instruction to 0x00; Reduce the loop instruction to 0x02;
Delta(2 bytes): increment, i.e. the amplitude of each adjustment;
Min_Limit (2 bytes): Minimum;
Max_Limit (2 bytes): maximum;

This directive is generally used to increment or decrease the display value of the text control when the button control is clicked, and is generally used as a button
The control's inline directive. If the action style of the button control is set to long and time, the text control can be continuously incremented or decremented.

2.31 Update the progress bar control values

Instruction format: EE [B1 10 Screen_id Control_id Progressvalue] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Control Number Progressvalue (4 bytes): The new progress bar value

The directive mainly achieves incremental or decreasing progress. When using the progress bar control, the user first configures the control-related parameters on the host computer, such as the foreground map, the background map, and the map
The maximum and minimum values of the numeric values, as shown in Figure 2-17, then the host only needs to write the progress bar value to the corresponding ID to achieve the progress bar scrolling.

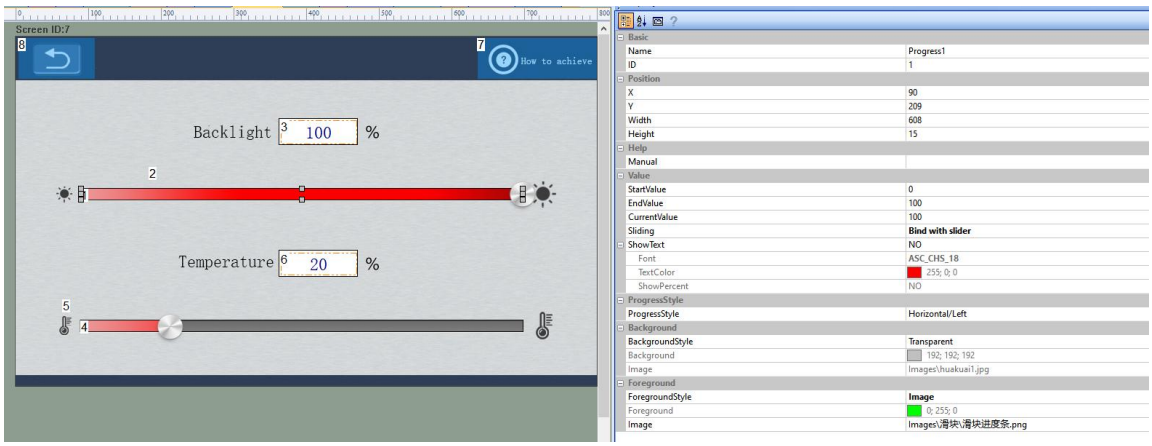

Figure 2-17 Progress bar control configuration information

2.32 Read the progress bar control value

Instruction format：EE 【B1 11 Screen_id Control_id】FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): Control Number

This directive is primarily used to get the numeric value of the current progress bar.

Return instruction format : EE B1 11 [Screen_id Control_id Control_type Progressvalue] FF FC FF FF

Return parameter: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type(1 byte): A fixed value of 0x12, represented as the progress bar control Progressvalue(4 bytes).): The current progress bar value

2.33 Slider control upload format

When using the slider control, the user first configures the control-related parameters on the host computer, such as ruler length, direction, cursor picture, background image and value, as shown in Figure 2-18, when the user drags the cursor, the device will continue to upload the current cursor value to the host, The current slider position can be determined by judging the cursor value.

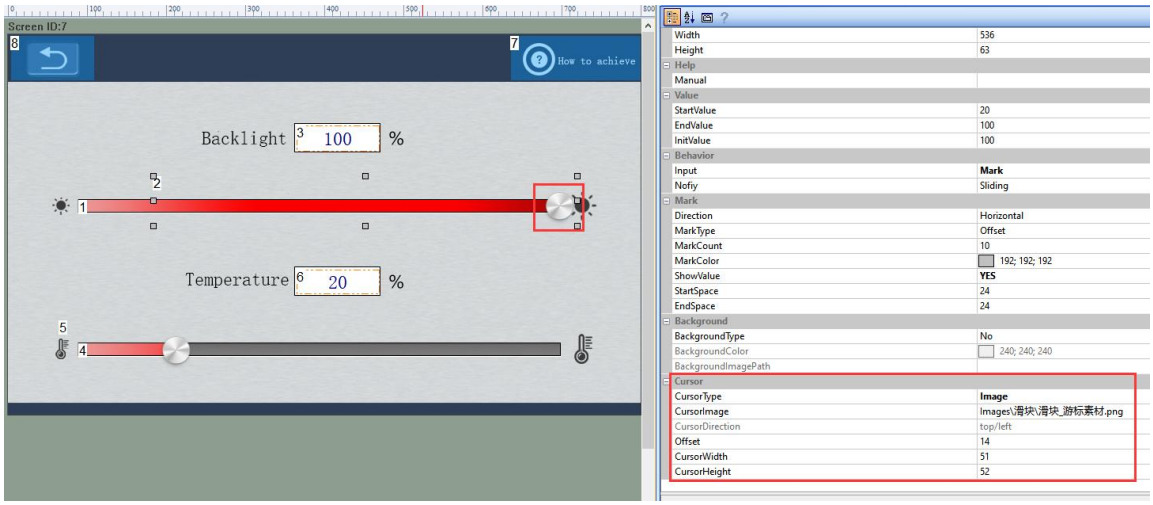

Figure 2-18 Slider control configuration diagram

Slider control upload format:

EE [B1 11 Screen_id Control_id Control_type Slidervalue]FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type(1 byte): The fixed value 0x13, expressed as the slider control Slidervalue (4 bytes).): Represents the current cursor value

2.34 Set the background color of the feed bar

Instruction format: EE [B1 18 Screen_id Control_id Bk_Color] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): control number

Bk_Color (2 bytes): Background color RGB color value This directive is mainly used to set the background color of the progress bar, and the user can modify the background color of the progress bar through the instruction.

2.35 Set the foreground color of the progress bar

Instruction format：EE [B1 19 Screen_id Control_id Bk_Color] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): The control number

Fore_Color (2 bytes): foreground color RGB color value This directive is mainly used to set the foreground color of the progress bar, and the user can modify the foreground color of the progress bar through the instruction.

2.36 Update slider control values

Instruction format：EE [B1 10 Screen_id Control_id Slidervalue] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Control Number Slidervalue (4 bytes): The new cursor value

This directive is mainly used to control the position of the slider cursor display. The user host can send the appropriate instructions to control the cursor forcing display in a certain location.

2.37 Read the slider control value

Instruction format: EE [B1 11 Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): Control Number This directive is primarily used to get the numeric value of the current cursor.

Return instruction format: EE B1 11 [Screen_id Control_id Control_type Slidervalue] FF FC FF FF

Return parameter: Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type(1 byte): The fixed value 0x13, expressed as the slider control Slidervalue(4 bytes): Represents the current cursor value

2.38 RTC clock settings

Instruction format：EE[81 Sec Min Hour Day Week Mon Year]FF FC FF FF

Parameter description: Sec: seconds set Min:

minutes set Hour: hours set Day: day set Week: day set Mon: day of the week set Mon : Month setting

Year: Year setting

All of the above parameters are 1 byte, expressed in BCD codes, and set to 0x00 on Sunday

This command is mainly used for the setting of the current time parameters, and it is recommended that the user set it directly through the host computer software. It is recommended that users use the clock control to display the RTC, and can directly tap the touch eject keyboard to calibrate the current time.

2.39 Read the RTC clock

Instruction format: EE [82] FF FC FF FF

Parameter description: None

This instruction is mainly used to get the current time value. Data upload format: EE F7 Year Mon Week Day Hour Min Sec FF FC FF FF FF 。 Each of the above parameters is 1 byte and is represented by a BCD code

2.40 Update gauge control values

Instruction format: EE [B1 10 Screen_id Control_id Metervalue] FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes):

Control Number Metervalue (4 bytes): The new gauge value

When using the instrument control, the user first configures the control-related parameters on the host computer, such as the dial, scale, pointer and value, as shown in Figure 2-19, and then the host only needs to send the corresponding value to realize the rotation of the instrument pointer.

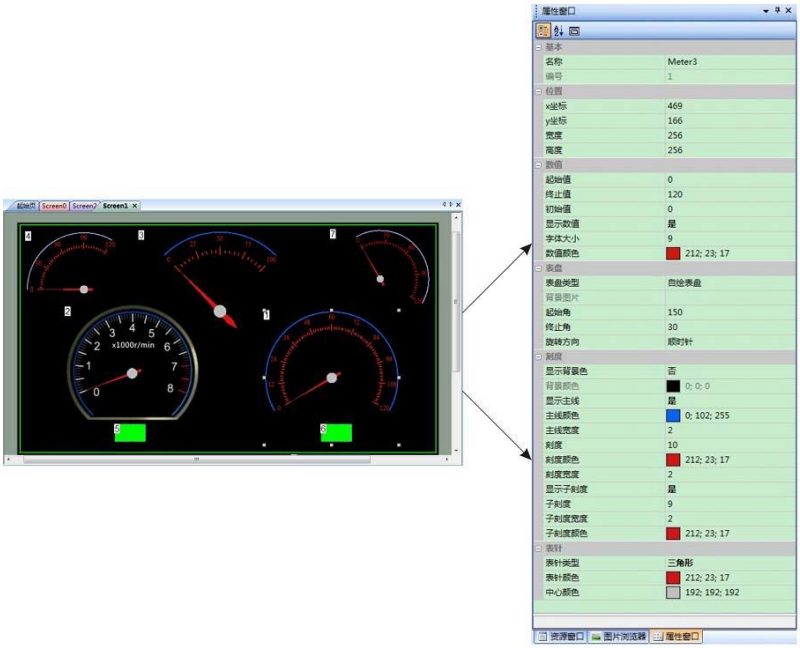

Figure 2-19 Gauge control configuration information

For example, the user needs to write the value 100 to the meter control with the control ID of 4 on the 0th screen, and the program code is as follows. Program reference code:

```
{
  . . ....
  SetMeterValue (0,4 ,1 0 0) // Gauge control with control ID 4 for screen 0 piece write
  numeric value 1 0 0,
```

2.41 Read the gauge control values

Instruction format: EE [B1 11 Screen_id Control_id] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): Control Number This command is primarily used to get the numeric value displayed by the current gauge.
Return instruction format: EE B1 11 Screen_id Control_id Control_type Metervalue FF FC FF FF FF
Return parameter: Screen_id (2 bytes): screen number
Control_ id (2 bytes): The control number
Control_type(1 byte): The fixed value 0x14, expressed as the meter control type Metervalue(4 bytes): The currently displayed gauge value

2.42 Animated control display

Through the animation control to call the gif animation display, you can support multiple gif display in the same screen, the user can send the corresponding command can control the animation playback, stop, pause and up/down frame and other functions, the instructions are introduced as Table 2.1 shown.

Table 2.1 Animation Control Instruction List

function	The directive format
Starts animation playback	EE 【B1 20Screen_id Control_id】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number After you start playback, the animation starts playing from frame head 0 each time
Stops the animation from playing	EE 【B1 21Screen_id Control_id】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number When playback stops, playback starts at frame head 0 next time
Pauses animation playback	EE 【B1 22Screen_id Control_id】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number After pausing, playback resumes next time from the pause frame
Specifies the frame playback	EE 【B1 23Screen_id Control_idFlashImgae_ID】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number FlashImgae_ID (1 byte): An animation frame ID specifies that playback starts at a certain frame

Play the previous frame	EE 【B1 24Screen_id Control_id】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number Displays the contents of the previous frame of the current frame
Play the next frame	EE 【B1 25Screen_id Control_id】FF FC FF FF
	Screen_ id (2 bytes): Screen number Control_id (2 bytes): Control number Displays the next frame of the current frame
Animation control values are uploaded	EE 【B1 26Screen_id Control_idStatusFlashImgae_ID】FF FC FF FF
	When you press an animation control, the device uploads the following information: Screen_ id (2 bytes): Screen number Control_ id (2 bytes): The control number Status (1 byte): 0x00 for touch press, 0x01 for bounce FlashImgae_ ID (1 byte): indicates the frame ID displayed when the screen is pressed

When used, the user first calls the gif animation through the animation control for display, and then sets whether touch notifications are required (also is click on the animation whether you need to upload the animation control ID value) and autoplay, as shown in Figure 2-20. The interval between animation playback times is automatically extracted from the original gif file.

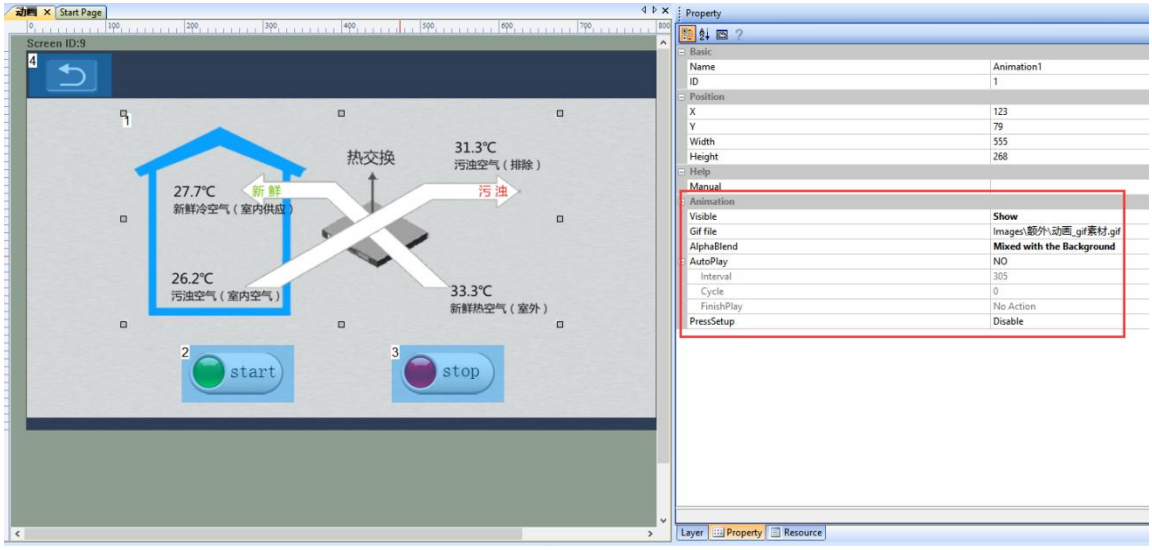

Figure 2-20 Animated Control Configuration Diagram

2.43 Icon control display

Instruction format：EE [B1 23 Screen_id Control_id IconImgae_ID]FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

IconImgae_ID (1 byte): The frame ID of an icon in the ICON file

This command mainly solves the toggle or disappear display of different status maps in the same position in the screen.

How do I create an icon? First clicks on the **Tools** menu of the VisualTFT software, selects **Icon Creator**, pops up the screen shown in Figure 2-21, and then click frame to add 3 pre prepared 62x82 pixel pictures (stop, run and transparent PNG). According to the order of arrangement, the first frame is the stop frame, the second frame is the run frame, the third frame is the pause frame,and the forth frame is the transparent picture. Finally, click the generation icon. In this way, a new ICON file is generated, which contains 4 pictures

Figure 2-21 Icon Creator Tool

When used, the user first invokes the newly generated ICON through the icon control for display, and then sets whether a touch notification is required (that is, whether the click icon control requires an upload ID) and autoplay, as shown in Figure 2-22. The screen displays the first frame of the ICON by default, and if you need to switch the status graph display, the host program only needs to send the ICON number and the first few frames to complete the display.

The transparent frame is actually an empty PNG image, and displaying the transparent frame can achieve the effect that the icon disappears or hides.

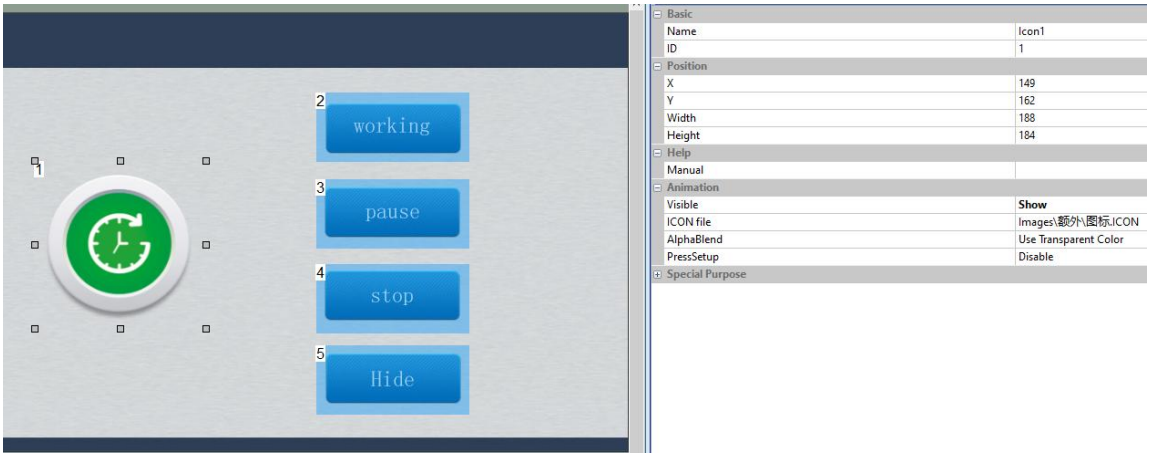

Figure 2-22 Icon control configuration diagram

2.44 Read icon control values

Instruction format:EE [B1 11 Screen_id Control_id]FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): Control Number

This directive is primarily used to get the frame ID displayed by the current icon.

Return instruction format: EE B1 11 Screen_ id Control_ id Control_ type IconImgae_ ID FF FC FF FF

Return parameter:

Screen_id (2 bytes): screen number

Control_ id (2 bytes): The control number

Control_type(1 byte): The fixed value 0x16, expressed as the icon control type

IconImgae_ID(2 bytes): The frame ID displayed by the current icon

2.45 Icon control value upload

Instruction upload format: EE [B1 26 Screen_id Control_id Status IconImgae_ID] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Control_type (1 byte): Fixed value 0x16, expressed as icon control type

IconImgae_ID (1 byte): ICON The frame ID of a picture in the file

If the PC software selects the icon control touch notification, the device uploads the ID of the icon control after the user taps the icon and the ID of the current picture frame.

2.46 Set the icon position

Instruction upload format: EE [B1 28 Screen_id Control_id X Y] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

X (2 bytes): The X coordinate position displayed

Y (2 bytes): Displayed Y coordinate position

This directive is mainly used to dynamically adjust the position of the icon control display.

2.47 Bulk update control values

Instruction upload format: EE [B1 12 Screen_id+Control_id0+Len0+Strings0++..." Control_idn+ Lenn+ Stringsn】FF FC FF FF

Parameter description:

Screen_ Id (2 bytes): picture number

Control_ Id0 (2 bytes): the number of the first control

Len0 (2 bytes): the byte length of the first control

Strings0 (variable length): the value of the first control. The number of bytes depends on Len0 length

Control_ Idn (2 bytes): the nth control number

Lenn (2 bytes): the byte length of the nth control

Stringsn (variable length): the numerical value of the nth control. The number of bytes depends on Lenn length

The batch update control instruction solves the problem that there is too much update data in a picture and the refresh speed is low. Using batch update not only saves the time of instruction transmission, but also updates all data at the same time. For the test of batch update instructions, you can open the Instruction Assistant of VisualTFT, and then enter several text control values for verification test, as shown in Figure 2-23.

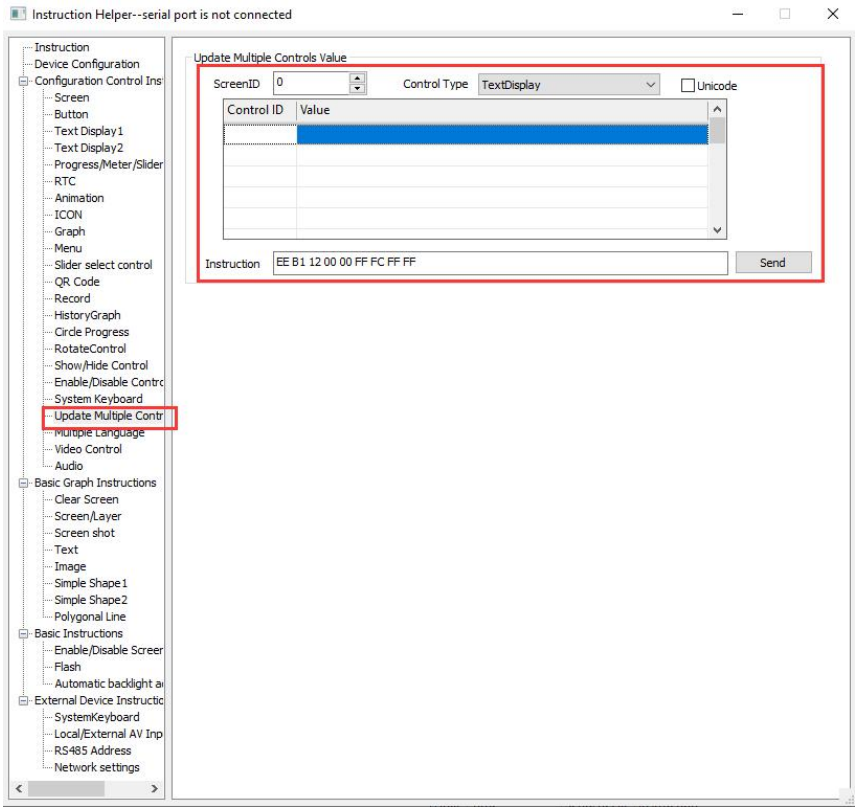

Figure 2-23 Bulk update text controls

2.48 Curve control display

If you need to realize dynamic chart data acquisition, automatic left and right panning, curve control will bring great convenience to users. The instructions are described in Table 2.2 ，and the host computer software configuration diagram is shown in Figure 2-24.

Recommendation: For a detailed tutorial on using the curve control, you can log in to the www.gz-dc.com data download bar to download it.

Table 2.2 Curve Control Instruction List

function	The directive format
Add a data channel	EE 【B1 30 SCREEN_ID CONTROL_ID CHANNEL COLOR】FF FC FF FF
	Specifies the data channel and curve colors SCREEN_ID (2 bytes): Screen number CONTROL_ID (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7). COLOR (2 bytes): The color of the data channel
	EE 【B1 31 SCREEN_IDCONTROL_ID CHANNEL】FF FC FF FF

	SCREEN_ID (2 bytes): Screen number CONTROL_ID (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7).
Specifies vertical/horizontal zoom/panning	EE 【B1 34 SCREEN_IDCONTROL_ID XOFFSET XMUL YOFFSET YMUL】FF FC FF FF
	Specify vertical/horizontal zoom/pan SCREEN_ ID (2 bytes): picture number CONTROL_ ID (2 bytes): control number XOFFSET (2 bytes): horizontally offset data points, with positive left shift and negative right shift XMUL (2 bytes): horizontal scaling factor, unit: 0.01 YOFFSET (2 bytes): vertical offset value, with downward shift being positive and upward shift being negative YMUL (2 bytes): vertical scaling factor, unit: 0.01 Calculation formula of sampling point and coordinate point: the value of the Nth sampling point is V X coordinate=(N-XOFFSET) * XMUL * 0.01 Y coordinate=(V-YOFFSET) * YMUL * 0.01
Specifies that new data is added to the channel frontend	EE 【B1 32 SCREEN_ID CONTROL_ID CHANNEL DATA_LEN DATA】FF FC FF FF
	Adds new data at the front end of the specified data channel, and when the data length exceeds the buffer length , the old data is shifted to the right SCREEN_ID (2 bytes): Screen number CONTROL_ID (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7). DATA_ LEN (2 bytes): Data length DATA : Indefinite length data, length specified by DATA_LEN: Data Representation is the value of the Y axis direction, and the X axis direction is automatically incremented according to the horizontal scale factor, such as when the horizontal scale factor is 1 The X axis is automatically added to each point inserted, and 5 is automatically added for each point X axis inserted when the horizontal scaling factor is 5
Specifies that new data is added at the end of the channel	EE【B1 35 SCREEN_ID CONTROL_ID CHANNEL DATA_LEN DATA】FF FC FF FF
	Inserts new data at the end of the specified data channel, and when the data length exceeds the buffer length , the old data is shifted to the left SCREEN_ID (2 bytes): Screen number CONTROL_ID (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7). DATA_ LEN (2 bytes): Data length DATA：Indefinite length data, length specified by DATA_LEN
Clears the data for the specified channel	EE 【B1 33 SCREEN_ID CONTROL_ID CHANNEL】FF FC FF FF

	Clears the data for the specified channel SCREEN_ID (2 bytes): Screen number CONTROL_ID (2 bytes): The control number CHANNEL (1 byte): A total of 8 data channels, numbered range (0~7).

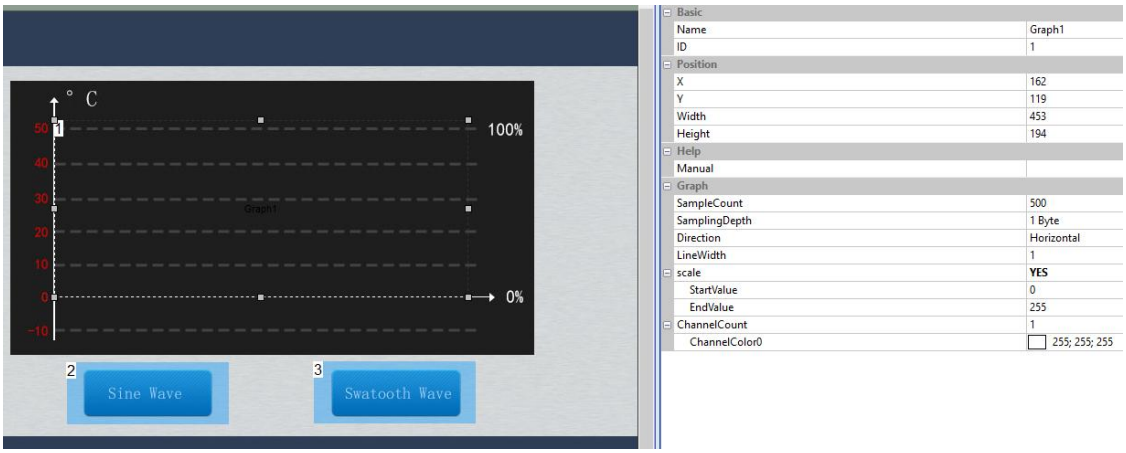

Figure 2-24 Curve Control Configuration Diagram

2.49 Set the drop-down menu to write a text control

Instruction format：EE 【B1 13Screen_id Control_id Enable Textctrl_ID】FF FC FF FF

Parameter description:

Screen_ Id (2 bytes): picture number

Control_ Id (2 bytes): control number

Enable (1 byte): menu Enable. 1 Display menu; 0 Do not show menu

Textctrl_ ID (2 bytes): number of text control to be written to menu data

This command mainly sets the text control to which the menu option data is written after the pop-up pull-down menu.

Suppose that the user needs to realize the following functions: click the pull-down button ① as shown in Figure 2-25, and a pull-down menu ② pops up. The user selects

Select 57.7V data item ③, and the data will be automatically written into the text control (text box). At the same time, the screen uploads the menu control ID and selection item number to the user's microcontroller, and the operation is as follows.

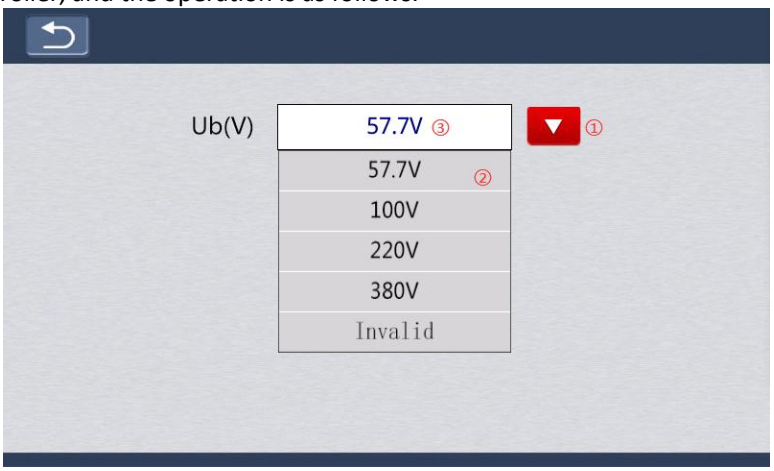

Figure 2-25 The drop-down menu shows the effect

Procedure:

1. Prepare two menu pictures, as shown in Figure 2-26, selected and unselected. If you do not need to press down the selection effect, you only need 1 picture that is not selected.

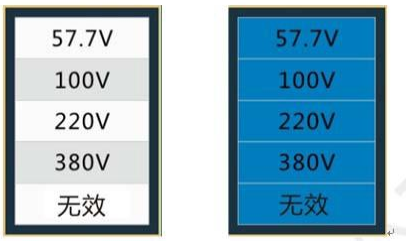

Figure 2-26 Drop-down menu picture with unchecked (left) and selected (right).

Set the drop-down menu properties. As shown in Figure 2-27, click the drop-down menu control at (1), drag and drop it at (2) position in the figure, and at the property window (3) on the right, set the menu appearance to: Customize Picture, and then (4) respectively , (5) set the menu picture to bounce up and press, and set the menu number (6) to 5, indicating that there are 5 drop-down items, and finally in (7). Fill in all drop-down data for . The data is separated from the data previously used separately and must be a semicolon in half an angle.

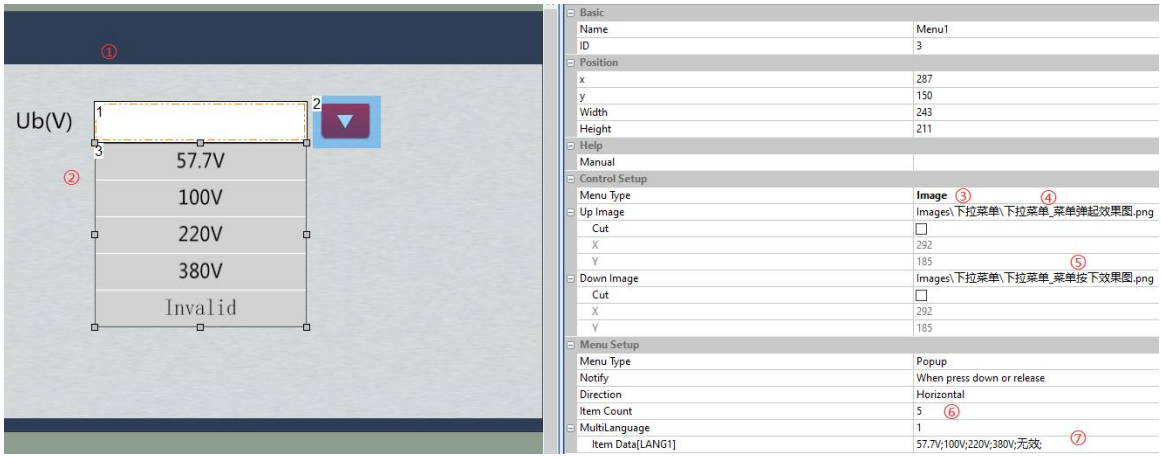

Figure 2-27 Setting the drop-down menu properties

Set the button of the pop-up menu and the text control to be written. As shown in Figure 2-27, place a text control at ① with the control ID of 2, then place a button control at ②, set the touch purpose as pop-up menu, as shown in ③, and finally fill in the menu control ID and text control ID values at ④ and ⑤ respectively.

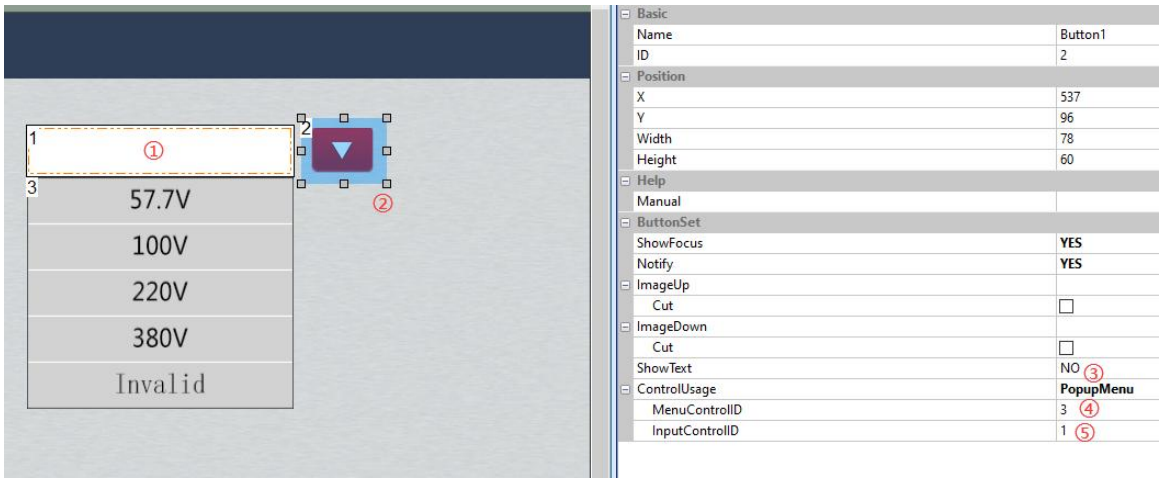

Figure 2-28 Setting the drop-down menu button and text control

The operation is complete. After compilation, run the virtual serial port screen, click the drop-down button, the menu will pop up, and then select the menu item, load the data into the text control and upload the ID information, as shown in Figure 2-29.

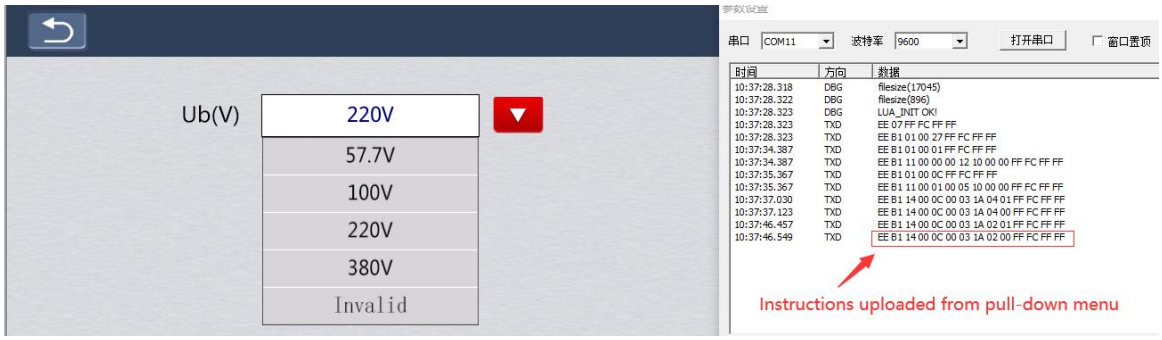

Figure 2-29 Drop-down menu upload instructions

2.50 Drop-down menu control value upload

Instruction format：EE 【B1 14 Screen_idControl_id1A Meundata_ID Status】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

0x1A (1 byte): A fixed number that represents a drop-down menu

Meundata_ID (1 byte): The data for the first few options in the drop-down menu

Status: Touch state, 0 indicates that the button pops up at this moment, 1 indicates that the button is pressed When the drop-down menu pops up and the user selects the required data, the screen will now change the current menu control ID and the number selected

The value item number is uploaded to the user's microcontroller. By parsing the serial port data, the user can understand the data currently pressed.

2.51 Set the swipe selection control value

Instruction format: EE [B1 10 Screen_idControl_idSelect_data_ID] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Select_data_ID (2 bytes): Slide the option ID in the selection control

This directive is mainly used to set the current option of the sliding selection control, that is, to index the option ID number of the sliding selection control, which can be used to swipe the selection control's "status display" function, that is, to send the corresponding index ID, sliding to select The Select control displays the current data item.

2.52 Read the sliding selection control value

Instruction format: EE [B1 11 Screen_idControl_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): Control Number This directive is primarily used to get the current option for a sliding selection control.

Return instruction format: EE B1 11 Screen_id Control_id 1Bvalue FF FC FF FF FF

Return parameter: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

0x1B (1 byte): A fixed numeric value that represents a swipe selection control value(1 byte): The index value of the currently displayed slide selection control selection

2.53 Swipe to select control value upload

Instruction format: EE [B111Screen_idControl_id1B Select_data_ID] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

0x1B (1 byte): A fixed numeric value that represents a swipe selection control

Select_data_ID (1 byte): Select the first few options in the menu This directive mainly implements sliding up and down to select the required data, and the system automatically uploads the control ID value and the currently selected value when selected. Suppose the user needs to implement the brightness setting effect shown in Figure 2-30 by swiping his finger up and down to select the desired value

After the screen uploads the selection control ID and selection item number to the user microcontroller, the operation is as follows.

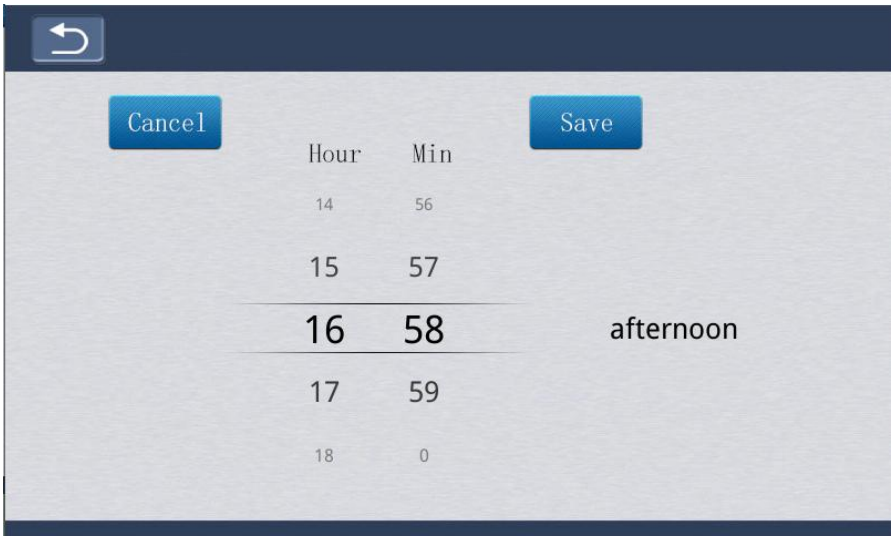

Figure 2-30 The sliding selection control shows the effect

Procedure:

1. As shown in Fig. 2-31, click the sliding selection control at ① of the software menu bar, and place a sliding selection control in Fig. ②.

2. Set the font size to ASCII16*32 at the Properties window (3), which is the font size displayed when selected.

3. Some parameters are defined as follows: Center Color: Indicates that the color is displayed after being selected; Color at both ends: indicates the top and bottom data display colors;

Scaled at both ends: Indicates the highest and lowest font sizes as percentage in the middle, 50% indicates that the fonts at both ends are half in the middle, and 0 means that all font sizes are the same, not scaled.

Number of items displayed: Fixed to 3 and 5 . Initial Options: Represents the data that is selected by default. Candidate Data: Represents all selected data.

4. Set the parameters at (4), (5), (6) and (7) in the Properties window.

5. Set candidate data at the Properties window (8): 0; 10; 20; 30; 40; 50; 60; 70; 80; 90。

6. The operation is complete. Run the virtual serial port screen, when you slide up and down, you can see the command data uploaded on the screen, such as Figure 2-32 shows.

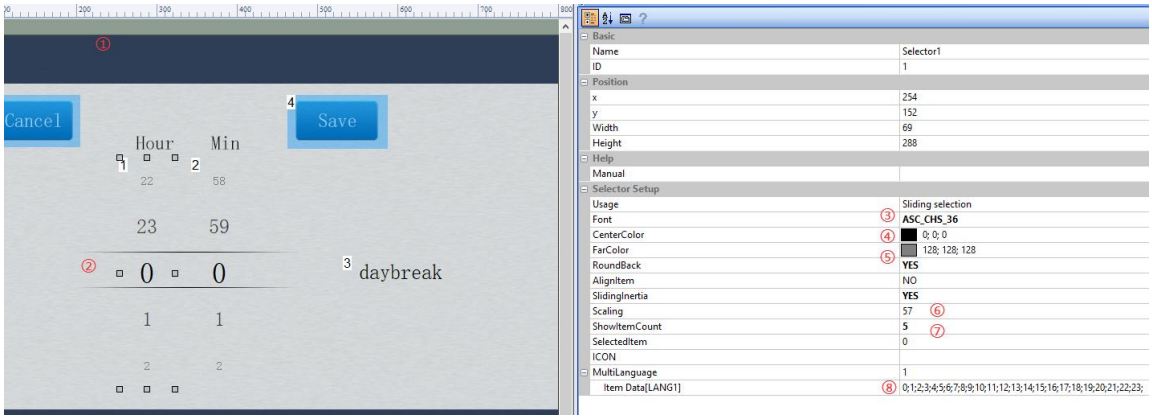

Figure 2-31 Slide Selection Control Property Settings

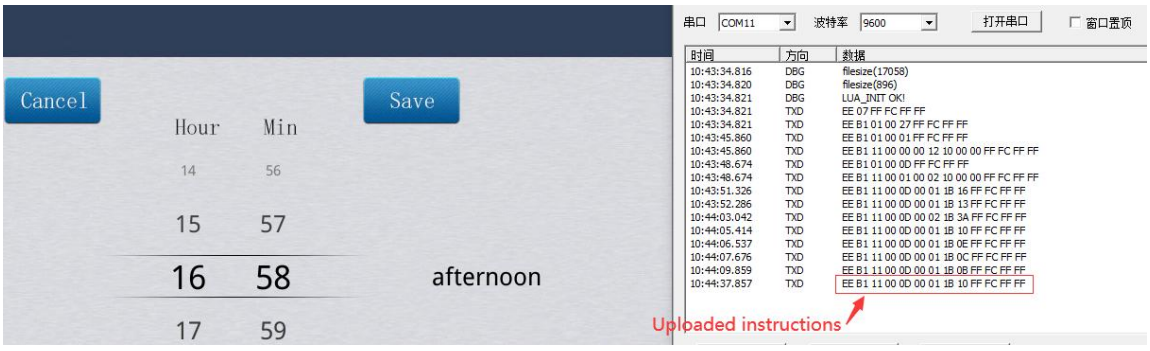

Figure 2-32 The swipe selection control uploads instruction data

2.54 Update QR code controls

Instruction format：EE【B1 10 Screen_id Control_id Strings 】FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Strings: QR code character content, encoding type UTF8

Note: If the sent QR code content has Chinese characters, it must be converted to UTF8 format, the user needs to be familiar with the UTF8 code format, and then convert and send.

This instruction is mainly used to implement the two-dimensional code map display, and the instruction format is the same as that of the text control. The user needs to display the QR code of what content, and directly send the character content.

Suppose the user needs to display the QR code shown in Figure 2-33, scan the content: https://www.elecrow.com/, then the operation is as follows:

Figure 2-33 The QR code control is displayed

Procedure:

1. As shown in Figure 2-34, click the QR code control icon at navigation (1) and place it in the area of (2) of the screen.
2. In the Properties window text (3), enter: https://www.elecrow.com/
3. The operation is complete. Run the virtual serial port screen to display the two-dimensional code image shown in Figure 2-33, which can be accurately identified by scanning with a mobile phone.
4. If the user needs to update the current QR code content of the microcomputer, send the corresponding command. Suppose the picture in Figure 2-34

With an ID of 5, a QR code control number of 1, and a display of 123, the command is sent as follows: EE B1 10 00 05 00 01 31 32 33 FF FC FF FF, where 31 32 33 represents the ASCII character of 123.

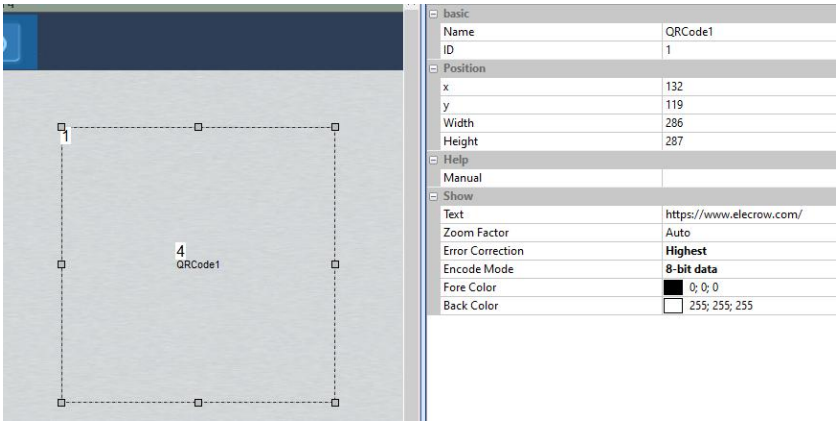

Figure 2-34 QR code display settings

2.55 Start the timer

Instruction format: EE [B1 41Screen_id Control_id] FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_id (2 bytes): The RTC control number

This command is only for screens where the hardware supports RTC. When using, first click the RTC control in the navigation bar, place it in the upper right corner of the picture, set the count format, timing length and display mode, as shown in Figure 2-35, and then the user microcontroller sends the start timer command, and the time begins to time in turn.

When the timer time arrives, the screen will upload the event notification in EE [B1 43 Timedata 17] FF FC FF FF. Upload parameter description: Timedata (4 bytes): The timer time value set by the user, with the high bit first.

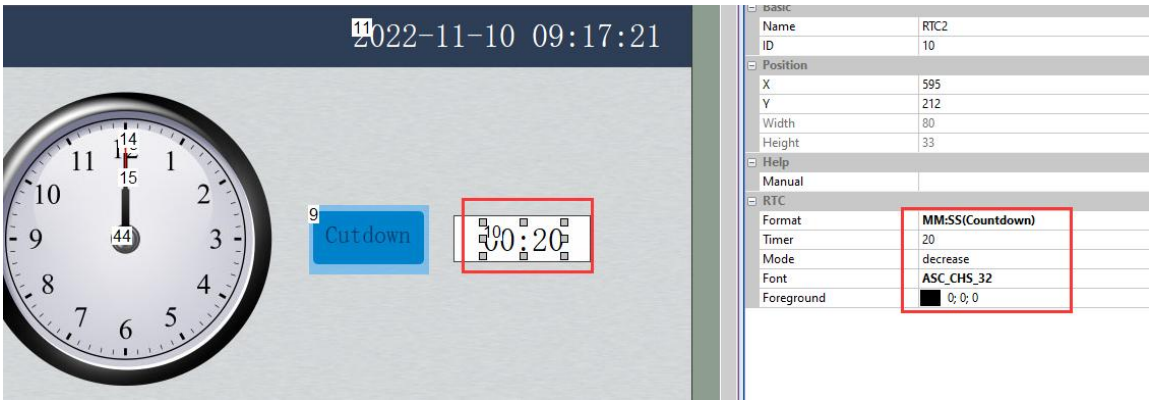

Figure 2-35 Timer Settings

The user can also use the Set Timer command to modify the timing parameters.

2.56 Set the timer

Instruction format: EE [B1 40Screen_id Control_id Timedata] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): RTC control number Timedata (4 bytes): The timer value

This directive is only for screens where hardware supports RTC and is used to modify timer time parameters.

2.57 Start the timer

Instruction format: EE [B1 41 Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): RTC control number This directive is for screens where the hardware supports RTC only and is used to start the timer.

2.58 Pause the timer

Instruction format: EE [B1 44Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): The RTC control number

This command is only for screens where hardware supports RTC, and is used to pause the timer, start the timer and then continue the timer.

2.59 Stop the timer

Instruction format: EE [B1 42Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The RTC control number

This directive is only for screens where the hardware supports RTC , and is used to stop the timer and the time value will return to the initial value.

2.60 Read Timer

Instruction format: EE [B1 45 Screen_id Control_id] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): RTC control number This instruction reads the timing value of the corresponding timer, and when the instruction is sent, the screen returns the corresponding timing value. Return format: EE B1 45 Screen_id Control_id 17 Strings FF FC FF FF Return parameter: Screen_ id (2 bytes): Screen number

Control_id (2 bytes): RTC control

number 17 (1 byte): 0x17, Fixed Code

Strings (4 bytes): The timing value of

the read

2.61 Manually disable/enable screen updates

Instruction format: EE [B3 Enable] FF FC FF FF

Parameter description: Enable (1 byte): 0x00 means to disable updates, 0x01 means to enable updates; This command mainly solves the problem that the number of controls updated dynamically in real time in a certain screen is too large, resulting in slow screen updates.

Because the system defaults to 100ms to automatically refresh the screen once, and in some occasions, the amount of user data is huge, and the system automatic update cannot meet the timeliness requirements, so the manual update screen command can be used.

How to use: The program first sends a command to disable screen updates, then sends the content that needs to be updated in the entire screen, and finally enables screen updates, so that the updated data is immediately displayed on the screen.

2.62 Show/Hide Controls

指令格式：EE 【B1 03 Screen_idControl_id Enable】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The control number

Enable (1 byte): 0x00 for hidden controls, 0x01 for display controls; This directive is commonly used at a certain point to disable the functionality of a specified button control until the mask is removed, and can also be used to move a certain button

The Paint/Icon control display disappears.

2.63 RS485 screen address settings

Instruction format: EE [A501 Addr] FF FC FF FF

Parameter description: Addr (2 bytes): Address information

This directive only supports screens for the 485 interface. This command is mainly used for screen address setting to facilitate system networking. After the address is set successfully, all instructions contain the address

Information, for example, the screen clearing instruction is: EE 01 FF FC FF FF, and the address is EE Addr1 Addr0 01 FF after the address is added FC FF FF。

Note: All addresses must be set from 1 onwards, 0 is the broadcast address, and all screens can receive broadcast information.

2.64 RS485 screen address canceled

Instruction format: EE [00 00 A500] FF FC FF FF

Parameter description: None

This directive only supports screens for the 485 interface. This command is mainly used for screen address cancellation, and the address in the instruction is directly the broadcast address.

2.65 Lock the system configuration

Instruction format： EE【08 A5 5A 5F F5】FF FC FF FF

Parameter description: None This directive prevents the system configuration from being accidentally modified due to receiving a host error instruction frame during system operation. Once configured

When locked, the device will not be able to receive external serial commands to modify until the lock is unlocked. Configuration parameters include: baud rate, touch and matrix keyboard working modes, automatic backlight adjustment parameters, which can be configured directly in the PC's command assistant.

2.66 Unlocking the System Configuration

Instruction format:EE【09 DE ED 13 31】FF FC FF FF

Parameter description: None Once the system configuration lock is released, the device can re-receive external serial port commands to modify configuration parameters. The user can be straight

Configure it in the PC's Instruction Assistant.

2.67 Alarm event triggered

Instruction format： EE【B1 50 Screen_idControl_idValue（SecMinHourDayWeekMonYear）】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The data record control number

Value(2 bytes): Alarm event ID number If the customer selects the instruction time, the instruction adds the following parameters:

Sec (1 byte): Seconds, indicated by BCD code
Min (1 byte): Minutes, BCD code represents Hour (1 byte): when, BCD code represents Day(1 byte): Day, BCD code represents Week (1 byte): Week, BCD code for Mon (1 byte): Month, BCD code for Year (1 byte) : Year, BCD code indicates

The instruction is to transmit the corresponding event ID to the screen, so that the corresponding event ID number is converted into an alarm event output set by the user.

2.68 Alert event lifted

Instruction format： EE【B151 Screen_id Control_idValue（Sec Min Hour Day Week Mon Year）】

FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The data record control number

Value(2 bytes): Alarm event ID number If the

customer selects the instruction time, the instruction adds the following parameters:

Sec (1 byte): Seconds, indicated by BCD code
Min (1 byte): Minutes, BCD code
represents Hour (1 byte): when, BCD code represents Day(1 byte): Day, BCD code represents Week (1 byte): Week, The BCD code indicates
Mon (1 byte): Month, BCD code represents Year : Year, BCD code means

The command is to transmit the corresponding event ID to the screen, so that the corresponding event ID number is converted into a user-set release alarm event output.

2.69 Data Record Control - Add General Records

Instruction format: EE【B1 52Screen_id Control_id String】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number String (Indefinite length): The contents of a string added by the user

The directive is used to add user data in the data record control, for displaying some important parameters and the corresponding data table, the user can use the instruction to add the corresponding user data to the data record control.

2.70 Data Logging Control - Clear Record Data

Instruction format: EE [B1 53Screen_idControl_id] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): Data Record Control Number This directive is used in the data logging control to clear all recorded data.

2.71 Data Record Control - Sets the record display offset

Instruction format: EE【B1 54 Screen_id Control_id Offset】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Offset(2 bytes): The line number

This directive is used to set the position of the scroll bar, which is the line number corresponding to the first line of the record control.

2.72 Data Record Control - Gets the current number of records

Instruction format: EE [B1 55 Screen_id Control_id] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_ id (2 bytes): data record control number This directive is mainly used by the data record control to obtain the number of records of the current data Return instruction: EE[B1]. 55 Screen_id Control_id 1D Count] FF FC FF FF returns the parameter: Screen_id (2 bytes): Screen number

Control_id (2 bytes): The data record control number
1D (1 byte): 0x1D, fixed value, representing the record control

Count(2 bytes): The current number of records

2.73 Data Record Control - Reads a row of records

Instruction format：EE【B1 56 Screen_id Control_id Position】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Position(2 bytes): The line number of a row

This directive is used by a data record control to read a row of records and returns the corresponding data instruction. Return instruction: EE [B1 56 Screen_idControl_id 1D String] FF FC FF FF Return parameter: Screen_id (2 bytes): screen number Control_id (2 bytes): Data record control number 1D (1 byte): 0x1D, fixed number String (Indefinite length): The contents of a string added by the user

2.74 Data Record Control - Modify General Records

Instruction format：EE【B1 57 Screen_id Control_id Position String】FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Data record control number Position: The line number of a line is String (not a certain length): The content of a string added by the user

This directive is used by the data record control to modify a regular record, which can be modified accordingly for a row.

2.75 Data Record Control - Deletes a row of records

Instruction format：EE【B1 58 Screen_id Control_id Position】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Position(2 bytes): The line number of a row

This directive is used in a data record control to delete a row of records, which can make it easier for users to delete a row of records.

2.76 Data Record Control - Insert General Record

Instruction format：EE【B1 59 Screen_id Control_id PositionString】FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Data record control number Position: The line number of a line is String (not a certain length): The content of a string added by the user

This directive is used in a data record control to insert the corresponding regular record at a row position, after which the record as a whole is moved back.

2.77 Data Record Control - Select a row of data records

Instruction format：EE【B1 5A Screen_id Control_id Offset】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Offset(2 bytes): The line number

This directive is used to select a row of data records in a data record control and display the corresponding selected color. Note: Users using this directive must select inside the row where

the data record control has a record and allow the Select option to be turned on.

2.78 Data Record Control - Add multiple records at once

Instruction format：EE【B1 5B Screen_id Control_id Count Record0_size Record0 Record1_size Record1… 】FF FC FF FF

Parameter description: Screen_id (2 bytes): Screen number Control_id (2 bytes): Data record control number Count(2 bytes): The number of records added

Recordx_size (2 bytes): The size of each record,"; "Also count as a byte

Recordx: Each record corresponds to the corresponding data, based on the length of Recordx_size This directive can mainly realize that multiple data records can be added at a time in the data record control.

2.79 Data Logging Control - SD Card Export Instructions (CSV format).

指令格式:EE【B1 5C Screen_id Control_id】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Data record control number This directive is primarily used to export data record controls to an SD card in CSV format. Return instruction: EE [B1 5C Screen_id Control_id Status] FF FC FF FF Return parameter: Screen_id (2 bytes): screen number Control_id (2 bytes): The data record control number Status (1 byte): 0x00 indicates success, others indicate failure

2.80 History curve setting sample values

Instruction format：EE【B1 60 Screen_id Control_id Value 】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): The text control number Value (Indefinite length): The data types are: UINT8, INT8, UINT16, INT16, UINT32 , INT32, FLOAT, the data of each channel corresponds to the set data type, if there are two channels, the data type is UINT8, and the corresponding two data types are added as UINT8 Data This instruction is used to add the sampled value data of the historical curve, and the corresponding data is added to the instruction, and the instruction is sent to the screen The screen draws the curve according to the corresponding data value.

2.81 History curve disable/enable sampling

Instruction format：EE【B1 61 Screen_id Control_id Enable】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Text control number Enable: Disable/enable bit

0x00: Prohibited;

0x01: Enable; This directive is mainly used to disable/enable sampling of historical curves. If the user needs to not enter a certain piece of data If you enter, you can use the disable sampling instruction to implement it; If the user needs to re-assign the sampling function, it needs to be implemented with the enable sampling instruction.

2.82 History curve hides/shows channels

Instruction format：EE【B1 62 Screen_id Control_idChannel Show】FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number Control_id (2 bytes): Text

control number Channel (1 byte):
Channel ID
Show (1 byte): Disable/enable bits are displayed
0x00: hidden;
0x01: Display; This command is mainly used to show/hide certain channels of the history curve. There are several kinds in the user interface Channel curve, if the user needs to view and analyze a channel in detail, you can use the command to hide the curve of the corresponding channel, and in the corresponding case, you can also use the instruction to display the channel curve.

2.83 The historical curve sets the duration (i.e. the number of sample points).

Instruction format: EE [B1 63 Screen_id Control_id00 Sample_Count] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): The text control number
00 (1 byte): Controls the type ID number, 0x00: sets the time width
Sample_Count (2 bytes): The number of sample points displayed on the screen This command is mainly used to set the duration of the historical curve, that is, the number of sampling points of the historical curve can be set.

2.84 The history curve zooms to full screen

Instruction format: EE [B1 63 Screen_id Control_id 01] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): The text control number
01 (1 byte): Control type ID number, 0x01: Zoom to full screen This command is mainly used to set the zoom of the historical curve, that is, the historical curve zoom can be set to reach full screen.

2.85 Historical curve setting scale system

Instruction format: EE [B1 63 Screen_id Control_id 02 ZoomMax_ZoomMin_Zoom] FF FC FF Ff
Parameter description: Screen_id (2 bytes): screen number
Control_ id (2 bytes): The text control number
02 (1 byte): Controls the type ID number, 0x02: Scaling factor
Zoom (2 bytes): Zoom percentage
Max_Zoom (2 bytes): Zooms to the maximum limit to display the maximum number of sample points on one screen
Min_Zoom (2 bytes): Minimum limit for scaling, minimum number of sample points displayed on one screen This instruction is mainly used to set the historical curve to be displayed according to the user-set scale factor so that the curve is drawn
Ability to match the angle of view of the user.

2.86 History curve setting value display range

Instruction format:EE【B1 64 Screen_id Control_id Max Min】FF FC FF FF
Parameter description: Screen_id (2 bytes): Screen
number Control_id (2 bytes): Text
control number Max(4 bytes): Maximum
Min (4 bytes): Min This instruction is mainly used to set the maximum and minimum values of the historical curve, that is, to set the display range of the historical curve.

2.87 Historical curve data is cleared

Instruction format: EE B1 65 [Screen_id Conrol_id] FF FC FF FF

Parameter description: Screen_id: Screen ID (two bytes).

Conrol_id: Control ID (two bytes).

This command is mainly used to empty the data of all channels, but the control cannot be emptied after power-down storage is turned on.

2.88 Rotate control sets the rotation angle

Instruction format：EE【B1 10 Screen_id Control_id Value 】FF FC FF FF

Parameter description:

Screen_id (2 bytes): screen number

Control_ id (2 bytes): The text control number

Value (2 bytes): rotation angle, rotation angle of 0-360 degrees; At the same time, the rotation control supports small angle rotation with a granularity of 0.1 degrees, and Value = Actual angle *10+0x8000, for example, 5.3 degrees set value = 5.3*10+0x8000

This instruction is mainly used to set the rotation angle of the rotation control, and the user can use this instruction to rotate the pointer according to the angle set by the rotation center rotation.

2.89 Multilingual switching instructions

Instruction format:EE【C1 Lang Check】FF FC FF FF

Parameter description: Lang (1 byte):

BIT3~BIT 0 : Identity Language Index, Range: 0-9

BIT6~BIT4 : Reserved, Default 0 BIT7: Identify System Keyboard Language, 0:Chinese, 1 : English

Check (1 byte): for verification, Check = 0xC1 + Lang, take low bytes This instruction is generally used for multi-language switching instructions, users can use the change instruction to set the output language of the keyboard

Whether it is Chinese or English.

2.90 Get the system language index

Instruction format: EE [C2] FF FC FF FF parameter description: None This instruction is used to obtain the language index currently used by the serial port screen; Return instruction: EE [C2 Lang] FF FC FF FF Return parameter: Lang (1 byte):

BIT3~BIT0 : Identity Language Index, Range: 0-9

BIT6~BIT4 : Reserved, Default 0 BIT7: Identify System Keyboard Language, 0:Chinese, 1: English

2.91 Video-related directives (applicable only to IoT type and M type).

2.91.1 Add Video Path

Instruction format: EE [B1 74 Screen_id Control_id Videos_Path] FF FC FF FF

Parameter description: Screen_id (2 bytes): screen number

Control_id (2 bytes): The number of the video playback control

Videos_Path: Video path (can be converted in the Instruction Assistant) This command is used to add the video path that needs to be played before playback. Note: Please refer to the relevant technical documentation for video playback on a USB flash drive or SD card.

2.91.2 Play the video

Instruction format：EE【B1 70 Screen_id Control_id Number】FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): The number of the video playback control
Number (2 bytes): the number of times the video loops play, 0 is the loop playback This command is mainly used to control the video playback control of the playback video, you can set the corresponding number of video playback loops, so that the video plays in accordance with the corresponding settings. (Note: The video playback controls must first set the video path).

2.91.3 Video Pause/Resume

Instruction format: EE [B1 72 Screen_id Control_id] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_ id (2 bytes): The number of the video playback control

2.91.4 Video stops

Instruction format: EE [B1 71 Screen_id Control_id] FF FC FF FF
Parameter description: Screen_id (2 bytes): screen number
Control_id (2 bytes): The number of the video playback control

2.92 Audio playback related instructions (applicable only to IoT type, F series, M type).

2.92.1 Audio playback

Instruction format: EE [94 Sounds_Path] FF FC FF FF
Parameter description: Sounds_Path: Audio file path (reference instruction assistant)
This command is used to specify the playback of an audio file under a path.
Note: Please refer to the relevant technical documentation for audio from a USB flash drive or SD card to play it.

2.92.2 Stop audio playback

Instruction format: EE [95] FF FC FF FF
This command is used to stop the audio from playing

2.92.3 Audio playback instructions

Instruction format：EE【90 01 sounds_id time】FF FC FF FF
Parameter description: sounds_id (2 bytes): Audio ID (only numeric IDs are supported) Time (1 byte).): The number of times to play (0 represents loop playback).
This directive is used to play audio files named after numbers.

2.92.4 Pause audio playback command

Instruction format: EE [90 02] FF FC FF FF
This command pauses the playback of the audio file.

2.92.5 Resume audio playback command

Instruction format: EE [90 03] FF FC FF FF

This command is used to resume audio file playback.

2.92.6 Stop audio playback command

Instruction format: EE [90 00] FF FC FF FF
This command pauses the playback of the audio file.

2.92.7 Audio sequence playback (for audio named after numbers).

Audio sequence playback command:
Instruction format：EE【90 04 00 01 sounds_id time】FF FC FF FF
Parameter description: sounds_ id (2 bytes): Audio ID (with ; Spaced)).
time (1 byte): The number of times to play (0 represents looping).

2.92.8 Volume adjustment

Instruction format: EE [93 Value] FF FC FF FF Parameter
Description: Value (1 byte): Volume value, ranging from 0-100
This command is mainly used to adjust the volume of the speaker output.

2.93 Matrix keyboard control

Instruction format: EE [79cmd] FF FC FF FF
Parameter description: Cmd (1 byte): Configuration parameter

BIT0: 1 indicates matrix keyboard enable, 0 indicates matrix keyboard off; BIT1: 1 means that the buzzer automatically rings when the key is pressed, and 0 means that it does not ring; BIT4~BIT2 : Matrix keyboard upload method

000: Indicates that coordinates are uploaded once when the keyboard is pressed

001: Indicates that the keyboard is pressed until the coordinates are uploaded once after release

010: When the keyboard is pressed for a long time, upload the coordinates once every 100ms, and upload the coordinates once when released

011: Indicates that the coordinates are uploaded once when the keyboard is pressed and released

BIT7-BIT5: Reserved

The device supports external 4 *4 matrix keyboard input. After pressing the keyboard, the key value is uploaded to the host through the serial port. Upload format for key-value encoding:
Upload format when pressed: EE 12 K FF FC FF FF F;
Upload format at release: EE 13 K FF FC FF FF F, K is 1 byte, representing the uploaded key value
The keyboard is a 4*4 matrix keyboard with a key-value encoding range of 0 to 15, the correspondence is shown in the following table 2.3 keycode query table.

Table 2.3 Key Code Lookup Table

Column / rows	K6	K7	K8	K9
K1	0	1	2	3
K2	4	5	6	7
K3	8	9	10	11
K4	12	13	14	15

2.94 Write data to FLASH

Instruction format：EE【87 Addr Data0 Data1 Data2… Datan】FF FC FF FF
Parameter description: Addr (4 bytes): The starting address at which data is written
Datan (1 byte): The data written has 128K bytes of storage space and an address range of 0 to 0x1FFFF. After a successful write, the device returns: EE 0C FF FC FF FF.
This directive mainly saves some data from the user part in the Flash of the device and uses it as an EEPROM.

2.95 Read data saved in FLASH

Instruction format：EE【88 Addr Length】FF FC FF FF
Parameter description: Addr (4 bytes): The data read start address
Length (2 bytes): The length of the read data in bytes The data returned is in the following format: EE 0B Data0 Data1 Data2 ... Datan FF FC FF FF
This instruction is primarily used to read out data written to memory.

2.96 Save the current picture to the TF card

Instruction format: EE [34 00] FF FC FF FF parameter description: None
The directive supports basic, commercial, and economical types, and the IoT type does not. After the directive is sent, the corresponding .bmp file is automatically generated in the TF card. If storage succeeds, the return format is: EE 10 FF FC FF FF F If storage fails, the return format is: EE 11 FF FC FF FF

2.97 Save the wireless network settings

Instruction format: EE [D0 A0] FF FC FF FF parameter description: None
Return instructions: EE D0 A0 WIFI_SET_CFG FF FC FF FF
Return parameter: WIFI_SET_CFG (not definite length): Save wireless network settings This command only supports IoT screens.
This command is mainly used to save the wireless network settings and save the user's wireless network settings. Wireless network settings include: disabled, NIC mode, hotspot mode.

2.98 Get wireless network settings

Instruction format: EE [D0 A1] FF FC FF FF Parameter description: None

Return instruction: EE D0 A1 WIFI_ GET_ CFG FF FC FF FF

Return parameter: WIFI_ GET_ CFG (variable length): Get the settings of wireless network
This command only supports the IoT screen. This command is mainly used to obtain the corresponding settings of the wireless network. After setting, the screen will return the corresponding parameter commands of the wireless network

2.99 Get Network Status

Instruction format: EE [D0 A2] FF FC FF FF parameter
description: None
Return instructions: EE D0 A2 WIFI_GET_STATE FF FC FF FF
Return parameter: WIFI_GET_STATE (not a certain length): Get
network status This directive only supports IoT screens.
The command is mainly used to obtain the network status, after setting, the network status is divided into wireless network connection, wired network connection, remote server connection, customer connection number. When the Get Network Status instruction is sent, the corresponding network status is selected.

2.100 Search wireless networks

Instruction format: EE [D0 A3] FF FC FF FF parameter
description: None
Return instructions: EE D0 A3 WIFI_SCAN_APLIST FF FC FF FF
Return parameter: WIFI_SCAN_APLIST (not a certain length): Search for wireless
networks This directive only supports IoT screens.
The instruction is mainly used to search for wireless networks, search wireless networks, so that users can search for the corresponding WIFI, wireless connection.

2.101 Save the network settings

Instruction format: EE [D0 A4] FF FC FF FF parameter
description: None
Return instructions: EE D0 A4 NETWORK_SET_CFG FF FC FF FF
Return parameter: NETWORK_SET_CFG (not definite length): Save network settings This directive only supports IoT screens. This command is mainly used to save network settings and save the user's network settings.

2.102 Get the network settings

Instruction format: EE [D0 A5] FF FC FF FF parameter
description: None
Return instructions: EE D0 A5 NETWORK_GET_CFG FF FC FF FF
Return parameter: NETWORK_GET_CFG (not definite length):
Get network settings This directive only supports IoT screens.
This directive is mainly used to obtain network settings, including: IP address, subnet mask, default gateway, DNS server.

2.103 Save network services

Instruction format: EE [D0 A6] FF FC FF FF parameter
description: None
Return instructions: EE D0 A6 NETWORK_SERVICE_SET FF FC FF FF
Return parameter: NETWORK_SERVICE_SET (not definite length): Save network service This directive only supports IoT screens. This command is mainly used to save network settings and save the user's network services. Network services include: disabled, guest
Client, server.

2.104 Get network services

Instruction format: EE [D0 A7] FF FC FF FF parameter
description: None
Return instructions: EE D0 A7 NETWORK_SERVICE_GET_CFG FF FC FF FF
Return parameter: NETWORK_SERVICE_GET_CFG (not a certain length): Get network service This directive only supports IoT screens. This directive is mainly used to obtain network services, which include: disabled, client, server.

2.105 Send network data

Instruction format: EE D0 AC Count Strings FF FC FF FF
Parameter format: Count (2 bytes): The number of bytes sent Strings (not definite length): The user selects the data to be sent and writes it in a software-defined format, for example: 12 , write as 31-32
This directive only supports IoT screens. This instruction is mainly used to
send network data, which is sent according to the corresponding ASCII code.

2.106 Send network data (hexadecimal).

Instruction format: EE D0 AC Count Strings FF FC FF FF
Parameter format:
Count: The number of bytes sent
Strings: The user selects to send the data and writes it in the appropriate format, for example: 12 13, write as 12 13
This directive only supports IoT screens. This instruction is mainly
used to send network data, sent in the appropriate format.

2.107 Save the network settings

Instruction format: EE [D0 AA] FF FC FF FF
parameter description: None This
instruction only supports IoT screens. This
directive is mainly used to save the
corresponding network settings.

2.108 Local/external AV input video

Instruction format: EE [4B Enable] FF FC FF FF parameter
description: Enable(1 byte): Local/external screen toggle flag bit
0x00: Switch to local image display
0x01: Switch to AV video input display This instruction only applies to basic, commercial, etc., not to the IoT type and F series, and the instruction is mainly used for local images
Switch between the display screen and the AV video
input display. Switch between the display
screen and the AV video input display.

2.109 Displays the system keyboard

Instruction format: EE [86 01 x y type Option max_len] FF FC FF FF parameter
description: 01 (1 bytes): fixed value, display keyboard;
x (2 bytes): the x-coordinate of the keyboard pop-up ;
y (2 bytes): the y-coordinate of the keyboard pop-up ;
type(1 byte): 0keypad, 1 full keyboard;
Option(1 byte): 0 normal characters, 1 password, 2 time settings;
max_len (1 byte): keyboard input character length limit;
This command is mainly used to show/hide the system keyboard; The value entered

by the keyboard is output through the serial port, and the instruction format EE[86 01 Strings] FF FC FF FF FF F is output.

Strings: The ascii code of the character entered by the user, such as keyboard input 123, in this case 31 32 33;

2.110 Hide System Keyboard

Instruction format: EE [86 00] FF FC FF FF

parameter description: None This instruction is mainly used to hide the system keyboard;

2.111 touch screen sensitivity adjustment command (only for capacitive screen).

Instruction format:EE【8A 5A A5 XX】FF FC FF FF

Parameter description: XX (1 byte): XX value range is 0-7, the higher the bit is more sensitive, the serial port screen factory default is

0. Adjust the touch sensitivity for supporting thicker covers. Do not adjust the sensitivity arbitrarily if it is not necessary.

This command is used to adjust the sensitivity of the touch screen and is suitable for capacitive screens.

2.112 enters standby

Instruction format: EE [60 00] FF FC FF FF

Parameter description: None This instruction is mainly used to make the screen enter the standby state, from the active state to the standby state;

2.113 Enter the active state

Instruction format: EE [60 01] FF FC FF FF parameter description: None

This command is mainly used to make the screen enter the active state, from standby to active state;

2.114 goes into hibernation

Instruction format： EE 【AA 02 AC】FF FC FF FF

Parameter description: None This instruction is mainly used to make the screen enter the sleep state, enter the sleep state from the active state, and reduce power consumption. Note: This directive only applies to IoT models, F-series, and M-series.

2.115 Exit hibernation

Instruction format:EE 【AA 00 00 00 00 FF FF FF FF 00 00 00 00】FF FC FF FF

Parameter description: None This directive is mainly used to make the screen enter the active state, from the sleep state to the active state. Note: This directive only applies to IoT models, F-series, and M-series.

2.116 Switch AV input signals

Instruction format:EE B1 77【Screen_id Control_id AV_ Serial】FF FC FF FF

Parameter description: Screen_id: Screen ID (two bytes).

Control_id: Control ID (two bytes).

AV_ Serial: Number of input channels (0/1 first input/second input).

This command is primarily used to switch between two AV video inputs.

Note: This directive only applies to type M.

2.117 Switch sprite

Instruction format: EE B1 0A 【Screen_id】FF FC FF FF parameter

description: Screen_id: Screen ID (two bytes) This command is mainly used to switch the sub-window screen, and it is required to switch to the PNG transparent picture.

Note: This directive only applies to IoT types.

2.118 The sprite window switches to the previous screen

Instruction format： EE B1 80 S【creen_id Conrol_id】FF FC FF FF

Parameter description: Screen_id: screen ID (two bytes).

Conrol_id: Control ID (two bytes).

This command is mainly used to empty the data of all channels, but it cannot be emptied after the storage is enabled.

2.119 The sprite window switches to the next screen

Instruction format: EE B1 81 [Screen_id Conrol_id] FF FC FF FF

Parameter description: Screen_id: screen ID (two bytes).

Conrol_id: Control ID (two bytes).

This command is mainly used to empty the data of all channels, but it cannot be emptied after the storage is enabled.

2.120 Enable Canvas

Instruction format:EE 1E Enable X Y W H FF FC FF FF

Parameter description: Enable: 0/1 enabled or disabled

X: The X-axis starting coordinate (two bytes).

Y: The starting coordinate of the Y axis (two bytes).

W: Width (two bytes).

H: Height (two bytes).

The instruction is mainly used to set the scope size of the canvas, to avoid the basic instruction set and the configuration instruction set conflict, the scope of the canvas shows the instruction display range of the basic instruction set, after setting the canvas, and then send the basic instruction set of instructions, configuration controls and basic instruction set does not conflict

2.121 Basic drawing instructions

Instruction format： EE B1 10 【Screen_id Conrol_id Number Color X0 X1 Y0 Y1】 FF FC FF FF

Parameter description: Screen_id: screen ID (two bytes).

Conrol_id: Control ID (two bytes).

Number: Drawing function code (01 line, 02 multipoint wire, 03 hollow rectangle, 04 solid rectangle,

05 Hollow Ellipse (circle), 06 Solid Ellipse (circle), 07 Image crop).

X0 : The upper-left X-axis coordinate

X1 : Lower right X-axis coordinates

Y0 : The upper-left Y-axis coordinate

Y1 : Lower right Y-axis coordinates This directive takes effect only after using the canvas and is used to draw a base graphic.

3. Appendix A details the basic instruction set

Since the configuration instruction set can meet 95% of engineering applications, and it is simple to operate, most of the configuration is done in the PC, so we do not recommend the use of basic instruction sets. Relative to the configuration instruction set, the basic instruction set belongs to the lowest level of instructions, the flexibility is stronger, in the case of configuration instructions can not be completed, you can use the basic set instructions to achieve. Detailed instruction set descriptions are described in the following sections.

3.1 Set the front/back view

Instruction format: Set foreground color: EE [41 Fcolor] FF FC FF FF Set background color: EE [42 Bcolor FF FC FF FF Set foreground and background colors: EE [40 Fcolor]. Bcolor】FF FC FF FF

Parameter description: Fcolor, Bcolor (2 bytes) are the RGB values of foreground color and background color foreground color is mainly used to specify the color of text, points, lines and circles, etc., and the background color is mainly used for screen clearance and text background color . For example, the instructions for setting the front/background color can be used to display the contents shown in Figure 3-1. Reference program:

```
{
    SandtBcolor(31);  //Set the background color to blue to specify the screen color
    GUI_ClshenScrand n( );   //Clear screen
    SetFcolor(65516);       //Set the foreground color to yellow and the user specifies the text display color
    SETBcollor (63488);      // Set the background color to red, which specifies the text background color (base color)
    DisText (50, 50, 1, 6, "Hello, color! ");   // Write a string of characters at coordinates (5 0,50), with a background color
    DisText(50, 90, 0, 6, "Hello, serial screen! ");   // Write the string at coordinates (5 0,90), no background color
    SetFcolor(2016);                    //Resets the foreground color to green
    GUI_RectangleFill(256, 57, 370, 116);         //Draw a solid rectangle
}
```

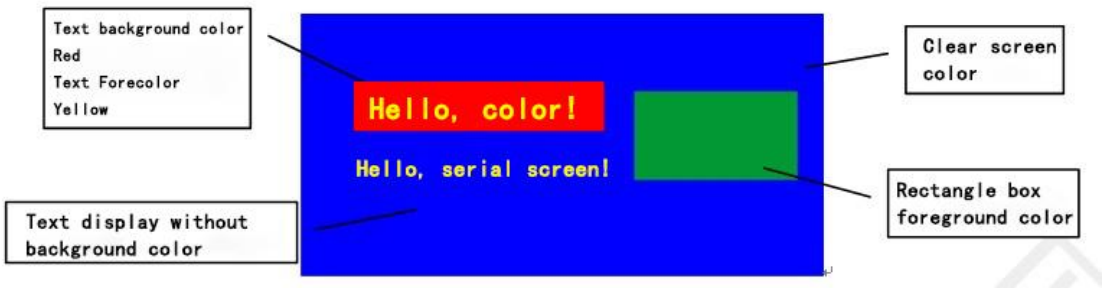

Figure 3-1 Background foreground color description

3.2 Automatically clear layers when switching screens

Instruction format: EE [06 Enable] FF FC FF FF

Parameter description: Enable (1 byte): Clears the flag bit

0x00: Do not clear

0x01: Auto clear This command is used to automatically clear the layer when switching the screen, that is, to automatically clear the layer when switching the screen

Non-control graphics (points, lines, rectangles, circles, ellipses, text, pictures).

3.3 Set the spacing between text lines and columns

Instruction format: EE [06 Enable] FF FC FF FF

Instruction format: EE [43 Y_W X_W] FF FC FF FF

Parameter description:

Y_ W (1 byte) is the line spacing in points, and the value range is 00~3F;

X_ W (1 byte) is the column spacing in points, and the value range is 00~3F.

This command is used to set the line spacing between text. If the text has only one line, the line spacing is 0. For example, two lines of 32 * 32 strings are displayed on the screen, and the line spacing is 24 and the column spacing is 16. The program is as follows.

Reference program:

```
{
    SetFcolor(65504);          //Set the foreground color to yellow and specify
    the text display color
     SetBcolor(63488);         //Set the background color to red, and specify
    the text background color
    SetTextSpace(24,16);       //Set text line spacing to 24 and column
    spacing to 16
    DisText(50, 51, 1, 7, "Elecrow HMI Display");    //Display Text String
}
```

3.4 Set the text box

Instruction format: EE [45 Enable Width Right] FF FC FF FF

Parameter Description: Enable (1 byte): turn on/off the text box limit enable Width (2 bytes): text box width;

Height (2 bytes): the height of the text box.

After setting the text box, the text will automatically wrap in the limit box.

3.5 Set the image filter color

Command format: EE [44 FillColor] FF FC FF FF

Parameter description: FillColor (2 bytes): RGB value of filter color; After setting the filter color, when a pixel value of the picture is exactly the same as the filter color value, the point will be shielded and cannot be displayed on the screen.

The contrast before and after setting the filter color is shown in Figure 3-2.

Reference code:

```
{
  DisArea_Image(0, 0, 0, 0);          //Grassland background map is displayed at (0,0)
  DisArea_Image(61,130, 1, 0);        //Unfiltered butterfly pattern is displayed at (61130)
  SetFilterColor(65535);              //Set white as filter color, RGB value is 65535
  DisArea_Image(258,68, 2, 1);        //The filtered butterfly pattern is displayed at (258,68)
}
```

Fig. 3-2 Comparison before and after setting filter color

3.6 Text display

Instruction format：EE【20 X Y Back Font Strings】FF FC FF FF

Parameter description: X (2 bytes): X-axis coordinate value in points;

Y (2 bytes): The Y-axis coordinate value in points;

Back (1 byte): Background color enabled

0x01: Background color display 0x00: The background color is not displayed

Font (font encoding, 1 byte).

0x00: 8x12 Dot Matrix (ASCII)0x01: 8x16 Dot Matrix (ASCII).

0x02: 12x24 dot matrix (ASCII)0x03: 16x32 dot matrix (ASCII).

0x04: 12 x 12 dot matrix (GBK)0x05: 16 x 16 dot matrix (GBK).

0x06: 24 x 24 dot matrix (GBK).

0x07: 32 x 32 dot matrix (GB2312).

0x08: 32 x 64 dot matrix (ASCII).

0x09: 64 x 64 dot matrix (GB2312).

Strings: A string written by the user, with the high byte first. This directive is used to display the specified text anywhere on the screen. GBK contains Chinese characters and Japanese and Korean font libraries; GB2312 contains only Chinese characters and characters; ASCII cannot display Chinese characters. In practice, after the user determines the foreground color, background color, and

word library encoding of the text, Chinese characters or strings can be written continuously, and the device will automatically wrap and match the display in Chinese and English. The text is displayed as shown in Figure 3-3.

Reference program:

```
{
 SetFcolor(65504);          //Set text foreground color, yellow
 DisText(46, 21, 0, 7, "工业串口屏 LCM 32*32");    //Text is displayed at the coordinates (46, 21), and the font library is 7 point font
```

Figure 3-3 Text shows the rendering

3.7Cursor display

Instruction format：EE【21 Enable X Y Width Hight】FF FC FF FF

Parameter description: Enable(1 byte): The cursor enables the signal

0x00: Cursor off, 0x01: Cursor on

X (2 bytes): The X-axis coordinate value in points Y (2 bytes): Y in points Axis coordinate value Width (1 byte): Cursor width

Hight (1 byte): Cursor height This instruction is mainly used to control the blinking and closing of the cursor. For example, the Chinese character appendix of the user in 24 *24 displays the width 16,

A cursor at height 8, as shown in Figure 3-4.

Program reference code:

```
{
  SetBcolor(31);                  // Set the blue background color
  GUI_ClshenScrand n( );          The background is clear blue
  DisCursor(1,359, 40,16,8);      The cursor blinks to enable the width at (359, 40). 16 Height 8 cursor
}
```

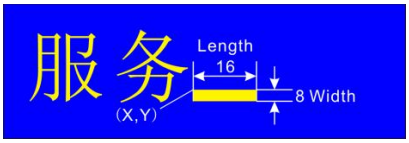

Figure 3-4 Cursor parameter description

3.8 Full screen image display

Instruction format: EE [31 Image_ID MaskEn] FF FC FF FF

Parameter description: Image_ID (2 bytes): Picture number

MaskEn (1 byte): Filter color enabled

0x00: color is not filtered; 0x01 Perform color filtering

This command mainly displays a full-screen picture with a fixed starting coordinate (0,0), and the filtered color depends on the filtered color setting. If the image size is less than the full screen size, only the actual size of the picture is displayed when the command is executed.

3.9 Area picture display

Instruction format:EE【32 X Y Image_ID MaskEn】FF FC FF FF

Parameter description: X (2 bytes): The X-axis coordinate value in dots Y (2 bytes): Y-axis coordinate value in points, Image_ID (2 bytes): Image number MaskEn (1 byte): Filter color enabled

0x00: color is not filtered; 0x01 Perform color filtering

This directive is used to implement the display of pictures at any location. Users need to pay attention to the starting coordinates and the width of the picture to prevent the display from exceeding the screen range. For example, if the user wants to display a picture with ID 2 at the screen (100,50), as shown in Figure 3-5, the reference program is shown below.

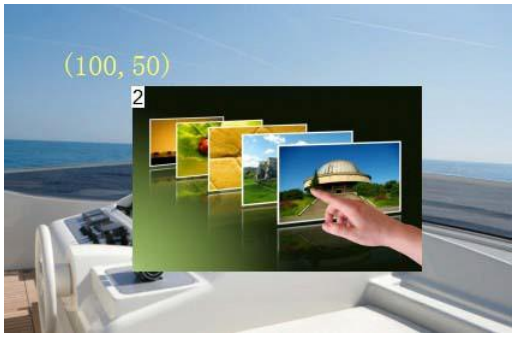

Figure 3-5 Area Picture Shows

Program reference code:

```
{
  DisFull_Image(0,0);                  // Full screen picture display
  DisArshe_Image(100,5 0, 2, 0);       The area picture is displayed, and ID=2 is displayed at coordinates (1 0 0, 50). The picture
 }
```

3.10 Image Cut

Instruction format:EE【33 X Y Image_ID Image_X Image_Y Image_W Image_H MaskEn】FF FC FF FF

Parameter description: X (2 bytes): The X-axis coordinate value in points
Y (2 bytes): The Y-axis coordinate value in points
Image_ID (2 bytes): Number of the picture to be cut Image_X (2 bytes): The starting point of the cut picture X coordinate Image_Y (2 bytes): The Y coordinate of the start point of the cut image Image_W (2 bytes): The width of the clipping
Image_H (2 bytes): The height of the clipping
MaskEn (1 byte) 0x00: colors are not filtered ; 0x01 Perform color filtering Note: The colors that are filtered depend on the settings of the filtered colors

This directive is used to display the clipped image at any coordinate of the screen. The user can locally crop any image stored in Flash as shown in Figure 3-6.

3.11 Animated display

Instruction format：EE【80 X Y FlashImage_ID Enable PlayNum 】FF FC FF FF

Parameter description: X (2 bytes): X axis coordinate value in points Y (2 bytes): Y-axis coordinate value in points FlashImage_ID (2 bytes): Animation number Enable: Enable signal

0x00: Turn off animation playback; 0x01: Turn on animation playback

PlayNum (1 byte).

0x00: repeat playback; 0x01 to 0xFF: Play a specified number of times

After playback stops, the device returns EE 02 FF FC FF FF to indicate that the animation playback ends. This directive is used to implement the display of gif animations at any position. Animations only support gif format, not two on the same screen

The above gif animation plays at the same time. If you expect multiple gif animations to play at the same time on one screen, as well as functions such as moviepause,stop, and play up/down frames, use the animation control instructions, as detailed in the Configuration Control Instructions Table. The image is animated as shown in Figure 3-6.

Program reference code:

```
{
DisFull_Image(0 , 0);                          // Full screen display of children's pictures, numbered 0
DisFlashImage(330,5, 1,1,0);                   // Insert the cow Flash animation at the coordinates
(330,5) and play it repeatedly
DisCut_Imag e(343,137,0,95,30,92,116,0);       // Cut size 92 x116 from picture 0 coordinate (95,30)
                                               //Picture of, displayed on the screen (343137)

}
```

Figure 3-6 Picture animated renderings

3.12 Foreground Drawing Points

Instruction format: EE [50 X Y] FF FC FF FF

Parameter description: X (2 bytes): The X-axis coordinate value in points

Y (2 bytes): The Y-axis coordinate value in points

This command mainly implements the drawing of points at any position on the screen, and the color value of the points depends on the setting of the foreground color.

3.13 Background color to draw dots (delete points).

Instruction format: EE [58 X Y] FF FC FF FF

Parameter description: X (2 bytes): The X-axis coordinate value in points

Y (2 bytes): Y-axis coordinate value in dots This directive mainly implements the use of drawing points anywhere on the screen, and the color value of the points depends on the setting of the background color. The background color is dotted

It is often used with foreground painting points, and you can delete the points painted by foreground color.

As shown in Figure 3-7, the user can display a yellow five-pointed star pattern on the black background map by drawing the foreground color point, and if you want to modify or delete this pattern, you can override the foreground written point at the same coordinate by calling the background color paint point command. Of course, if the user does not have too high requirements for data update speed, you can also clear the screen in the area first, and then redraw.

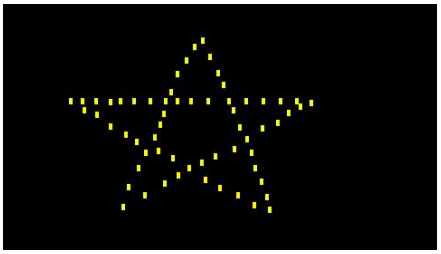

Figure 3-7 Foreground/background color dots

3.14 Draw the line

Instruction format: EE [51 X0 Y0 X1 Y1] FF FC FF FF

Parameter description: X0 (2 bytes): coordinate value of starting point of X axis of straight line in point

Y0 (2 bytes): coordinate value of starting point of Y axis of straight line in point

X1 (2 bytes): coordinate value of the end point of X axis of the straight line in point

Y1 (2 bytes): endpoint coordinate value of the Y axis of a straight line in point This command is mainly used to draw a line between any two points on the screen. The color value of the line depends on the setting of the foreground color.

As shown in Fig. 3-8. For example, a simple table is implemented by calling the line drawing assignment. The actual display effect is shown in Figure 3-8.

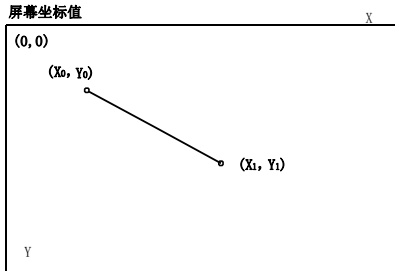

Fig. 3-8 Description of Line Drawing Parameters

3.15 Connect the equidistant X coordinates with the foreground color

Instruction format:EE【 59 X_0 Xspace Y_0 Y_1 Y_2。。。Y_n 】FF FC FF FF

Parameter description: X0 (2 bytes): The X-axis coordinate value in points

Xspace (2 bytes): A fixed interval value between two points adjacent to the X axis in points

Yn (2 bytes): Y-axis coordinate value in points This directive primarily implements fast drawing of polylines. Because the distance between the two points before and after the X axis is a fixed Xspace, so

None of the directive parameters from point 2 require an X-coordinate value. Draw polylines against foreground/background colors and double the speed.

3.16 Wire with foreground color according to coordinate offsets

Instruction format:EE【 75 X_0 Y_0 X_{1o} Y_{1o} X_{2o} Y_{2o} 。。。X_{no} Y_{no} 】FF FC FF FF

Parameter description: X0 (2 bytes): X-axis coordinate value in dots Y0 (2 bytes): Y in dots Axis coordinate value Xno (1 byte): The X-axis offset in points is Yno (1 byte). : The Y-axis offset in points

(X,Y) is the absolute coordinate of the first point, and each subsequent point consists of the absolute coordinate of the previous point plus the current offset. The highest bit of the offset is the sign bit, 0 represents the positive offset, 1 represents the negative offset, and the maximum offset value is plus or minus 127 points.

This command mainly uses the coordinate offset to turn the original 4 bytes of each coordinate into 2 bytes, reducing the command parameters by half, and doubling the drawing polyline speed. The color of the line is determined by the foreground color setting.

3.17 Connect the specified coordinate points with a foreground color

Instruction format:EE【68 X_0 Y_0 X_1 Y_1 ... X_n Y_n】FF FC FF FF

Parameter description: X n (2 bytes): Y_n (2 bytes) of the
coordinate value of the starting point of the X-axis
of the straight line in points : The coordinate value
of the start point of the Y axis of the line in points

This directive mainly implements the use of foreground color to connect multiple coordinate points specified. For example, to implement the polyline and hexagon shapes shown in Figure 3-9, the program is shown below.

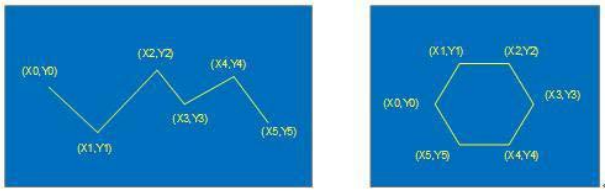

Figure 3-9 Specifying Coordinate Points Connected to the Renderings with a Foreground Color

Program reference code:

```
{
    SandtFcolor                                  //Sets the foreground color of the line to yellow
    GUI_FcolorConDots ( x 0,and 0,x 1,and 1,x 2,and 2,x 3,and 3,x 4,and 4,x 5,and 5); //Draw a polyline from (x0, y0) to (x5, y5)
    GUI_FcolorConDots (x 0,and 0,x 1,and 1,x 2,and 2,x 3,and 3,x 4,and 4,x 5,and 5,x 0,and 0); //Draw hexagon, end to end

 }
```

3.18 Connect the specified coordinate points with a background color

Instruction format:EE【69 X_0 Y_0 X_1 Y_1 ... X_n Y_n】FF FC FF FF

Parameter description: X n (2 bytes): Y_n (2 bytes) of the
coordinate value of the starting point of the X-axis
of the straight line in points : The coordinate value
of the start point of the Y axis of the line in points

This directive mainly implements the use of a background color to connect multiple coordinate points specified. The background color drawing polyline is usually used with the foreground color to draw polylines, and you can use the polyline drawn by the foreground color to be removed.

3.19 Wire with background color according to coordinate offset

Instruction format:EE[76 X_0 Y_0 X_{1o} Y_{1o} X_{2o} Y_{2o} ... X_{no} Y_{no}]FF FC FF FF

Parameter description:

X0 (2 bytes): X-axis coordinate value in dots
Y0 (2 bytes): Y in dots Axis coordinate value
Xno (1 byte): The X-axis offset in points is
Yno (1 byte). : The Y-axis offset in points

(X,Y) is the absolute coordinate of the first point, and each subsequent point consists of the absolute coordinate of the previous point plus the current offset. The highest bit of the offset is the sign bit, 0 represents the positive offset, 1 represents the negative offset, and the maximum offset value is plus or minus 127 points.

This command mainly uses the coordinate offset to turn the original 4 bytes of each coordinate into 2 bytes, reducing the command parameters by half, and doubling the drawing polyline speed. The color of the line is determined by the background color setting.

3.20 Draw hollow circles

Instruction format:EE【 52 X_0 Y_0 R 】FF FC FF FF

Parameter description: X_0 (2 bytes): The center X coordinate value in points

Y_0 (2 bytes): The center Y coordinate value in points

R (2 bytes): Radius of a hollow circle This directive is used to draw a radius R hollow circle at the specified coordinates , and the line color of the circle depends on the foreground color setting

Place. The parameter description is shown in Figure 3-10 .

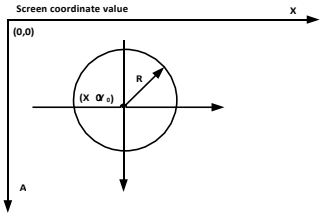

Figure 3-10 Drawing a Hollow Circle parameter description

3.21 Draw a solid circle

Instruction format:EE【 53 X_0 Y_0 R 】FF FC FF FF

Parameter description: X_0 (2 bytes): The center X coordinate value in points

Y_0 (2 bytes): The center Y coordinate value in points

R (2 bytes): The radius of a solid circle

This directive is used to draw a solid circle of radius R at the specified coordinates , and the color filling inside the circle depends on the setting of the foreground color. The parameter description is the same as in Figure 3-10.

3.22 Draw a circular arc

Instruction format:EE【 67 X_0 Y_0 R SA EA】FF FC FF FF

Parameter description: X_0 (2 bytes): The center X coordinate value in points

Y_0 (2 bytes): The center Y coordinate value in points

R (2 bytes): Radius of the circle SA (2 bytes): Start angle EA (2 bytes): End angle

This directive is used to draw an arc of radius R at the specified coordinates , and the arc color depends on the setting of the foreground color. The clock 3 point direction is 0 degrees from the starting angle, and the angle increases clockwise in turn, and the reference coordinates are shown in Figure 3-11.

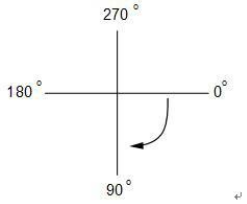

Figure 3-11 Reference Diagram of the Starting Angle of the Circular Arc

3.23 Draw a hollow rectangle

Instruction format:EE【 54 X_0 Y_0 X_1 Y_1】FF FC FF FF

Parameter description: X_0 (2 bytes): The X-coordinate value of the upper-left corner of the hollow rectangle in points

Y_0 (2 bytes): The Y-coordinate value of the upper-left corner of the hollow rectangle in points

X_1 (2 bytes): The X-coordinate value of the lower-right corner of a hollow rectangle in points

Y_1 (2 bytes): The Y coordinate value of the lower-right corner of the hollow rectangle in dots This directive is used to draw a hollow rectangle anywhere on the screen, and the color of the rectangular border depends on the setting of the foreground color. Ginseng

The number definition is shown in Figure 3-12.

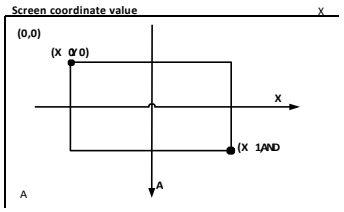

Figure 3-12 Drawing a hollow rectangle parameter description

3.24 Draw a solid rectangle/partial clearance

Instruction format:EE【 55 X_0 Y_0 X_1 Y_1】FF FC FF FF

Parameter description: X_0 (2 bytes): The X coordinate value of the upper-left corner of a solid rectangle in points

Y_0 (2 bytes): The Y coordinate value of the upper-left corner of a solid rectangle in points

X_1 (2 bytes): The X coordinate value of the lower-right corner of a solid rectangle in points

Y_1 (2 bytes): The Y coordinate value of the lower right corner of a solid rectangle in dots This directive is used to draw a solid rectangle anywhere on the screen, and the rectangle fill color depends on the setting of the foreground color. Parameters

The definition is the same as in Figure 3-12 . This function can also be used as a partial screen clearance.

3.25 Draw a hollow ellipse

Instruction format:EE【 56 X_0 Y_0 X_1 Y_1】FF FC FF FF

Parameter description: X_0 (2 bytes): The leftmost X-coordinate value of the hollow ellipse in points

Y_0 (2 bytes): The topmost Y coordinate value of a hollow ellipse in points

X_1 (2 bytes): The rightmost X-coordinate value of a hollow ellipse in points

Y_1 (2 bytes): The bottommost Y coordinate value of a hollow ellipse in dots This directive is used to draw a hollow ellipse anywhere on the screen, and the ellipse border color depends on the setting of the foreground color. Ginseng

The number definition description is shown in Figure 3-13.

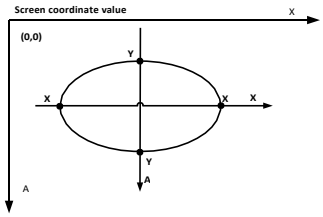

Figure 3-13 Drawing a hollow ellipse parameter description

3.26 Draw a solid ellyt

Instruction format:EE【 57 X_0 Y_0 X_1 Y_1】FF FC FF FF

Parameter description: X_0 (2 bytes): The leftmost X-coordinate value of the solid ellipse in points

Y_0 (2 bytes): The topmost Y coordinate value of a solid ellipse in points

X_1 (2 bytes): The rightmost X-coordinate value of a solid ellipse in points

Y_1 (2 bytes): The Y-coordinate value at the bottom of the solid ellipse in dots This directive is used to draw a solid ellipse anywhere on the screen, and the ellipse fill color depends on the setting of the foreground color.

3.27 Clear layers

Instruction format : EE [05 Layer] FF FC FF

FF parameter description: Layer (1 byte): Clear Layers This directive is primarily used to clear the specified layer.

3.28 Capture the current screen and save it in FLASH

Instruction format: EE [0x46 Image_ID] FF FC FF FF

Parameter description: Image_ID (1 byte): User-defined screen number saved in memory This instruction supports basic, commercial, economic, and IoT types do not support. This directive is primarily used to save the current screen display in Flash . The Image_ID value can be set arbitrarily

Does not conflict with the project picture ID value.

3.29 Displays screenshots saved in FLASH

Instruction format: EE [0x47 Image_ID] FF FC FF FF parameter description: Image_ID (1 byte The directive supports basic, commercial, economic, and IoT types do not. This command is primarily used to display screenshots saved in the device FLASH.

3.30 RTC mode settings

Instruction format：EE【85 Cmd DisMode TextMode Color Xpoint Ypoint 】FF FC FF FF

Parameter description: Cmd (1 byte):

Parameter configuration

BIT0: Enable signal

0 : RTC off 1: RTC on

BIT7-BIT1: Reserved

DisMode (1 byte): Display mode

0x00 : Format HH:MM:SS

0x01 : Forma 20XX-MM-DD HH:MM:SS

Font (1 byte): Font selection

0x00: 8x12 Dot Matrix (ASCII)0x01: 8x16 Dot Matrix (ASCII).

0x02: 12x24 dot matrix (ASCII)0x03: 16x32 dot matrix (ASCII).

0x04 12 x 12 dot matrix (GBK) 0x05: 16 x 16 dot matrix (GBK).

0x06: 24 x 24 dot matrix (GBK)0x07: 32 x 32 dot matrix (GB2312).

0x08: 32 x 64 dot matrix (ASCII)0x09: 64 x 64 dot matrix (GB2312).

Color (2 bytes): Displays the color

X (2 bytes): The X-axis coordinate value in points

Y (2 bytes): The Y-axis coordinate value in points

This command is mainly used for setting RTC display parameters. Display different clock formats, fonts and positions by setting corresponding parameters. The user is recommended to directly use the upper computer VisualTFT to set RTC related setting instructions. The setting reference interface is shown in Figure 3-14.

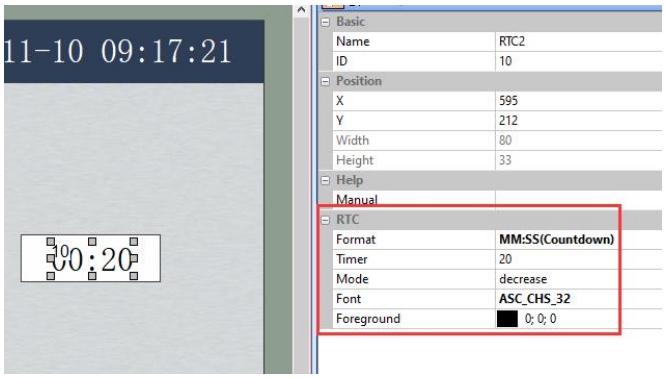

Figure 3-14RTC setup reference diagram