Shenzhen Juntong Technology Co., LTD		Document Number	Version number	Classification level
		202511190919	V1.0	Internal Public
Document Name	Full-color Display Secondary Development Interaction Protocol (CoolLEDUX)		Date	2025-11-19

Full-color display secondary Development Interaction Protocol (CoolLEDUX)

Document Author:	**Yang Shun**	**Date:**	**2025-11-19**
Audit:		**Date:**	
Approval:		**Date:**	

Document revision records

Serial Number	Version Number	Change status	Change (+/-) instructions	Author	Date
1	V1.0	C	1. A secondary development protocol based on the current latest protocol.	Yang Shun	2025-11-19

* Change status: C - Create, A - Add, M - modify, D - delete

Catalogue

1、 Introduction

1.1 Document Purposes

1. Sort out the main functions of the display screen.
2. Provide display interaction protocols for developers to use

1.2 Scope of application (optional)

Developers

2、 Equipment Overview

2.1 Equipment Introduction

Full-color display screen, flexible screen, drip glue process, can be directly pasted and can be seamlessly attached without light leakage, lightweight, thin and easy to install. It supports mobile APP and remote control. The mobile APP can set common functions such as text, animation, doodle, music rhythm, etc. Supports multiple built-in contents (text, doodles, animations) and local microphone rhythms, combining information display and entertainment functions. The features may vary depending on the type of display.

2.2 Feature Overview

1. Support multiple built-in contents (text, doodles, animations are all fine)
2. Supports setting up text functionality
3. Support defining doodles and animations
4. Music and microphone rhythm, with multiple music rhythm effects, and support for local microphone functionality
5. Support multiple display modes and text color modes.

Note: Functionality may vary depending on the type of display.

3、 Power supply requirements

The display supply requires DC-5V. Do not use the voltage beyond the range, otherwise the device will not work properly or even be damaged.

Note: Different resolutions of displays may have different power requirements. Please refer to the display specification sheet.

4、 Data Format

Data should be sent in a fixed data format, otherwise the device cannot recognize the data.

1	2	3	4	5				n+3	n+4
0x01	n		data						0x03
Start byte	Data length		Data						End byte

- Start byte

  Fix the value to 0x01 and occupy one byte; In the interacting data, no other data can be this value except for the starting bit being 0x01.

- Data length

  It takes up two bytes and represents the length of the data segment being transmitted this time, excluding the start bit, end bit, and data length. One length represents one byte, with the high byte at the front. For example: If the data length is 0x0017, the second byte corresponds to: 0x00, and the third byte corresponds

to: 0x17.

- Data

  Data to be sent

- End byte

  The fixed value is 0x03 and it occupies one byte. In the interacting data, no other data can be this value except for the starting bit being 0x03.

- Data forced conversion

  Except for the start and end bits, all other data must undergo a forced conversion before being sent. The conversion requirements are as follows:
  If a byte in the data to be sent is less than 0x04 and greater than 0x00, then a byte with a value of 0x02 needs to be added before that byte and XOR that byte with 0x04 before sending. The added 0x02 is not included in the data length.
  Reference:

test.js

```
1  function getData(ch) {
2      var out = [];
3      if (ch > 0x00 && ch < 0x04) {
4          out.push("02");
5          ch ^= 0x04;
6      }
7
8      out.push(ch.toString(16));
9
10     return out;
11 }
```

Example:

For example, you need to send a piece of data as: 0x00 0x01 0x05

The data length is: 0x0003. Since the high byte 0x00 of the data length is not within the range of mandatory swapping, it remains unchanged, while 0x03 needs to be forcibly converted to: 0x02 0x07

The final data sent is: 0x01 0x00 0x02 0x07 0x00 0x02 0x05 0x05 0x03

Sample code for data formatting:


app_commpt.c


app_commpt.h

5、 Data subcontracting processing

Because the data is too long, and the device's RAM space is limited and cannot accept too much data at once,

and in order to improve the reliability of transmission, some of the sent data needs to be subpackaged.

package data format:

Package type	Total data length	Package id	The current package data length	Current Packet data	Verification
1byte	4byte	2byte	2byte	nbyte	1byte
Package type	It refers to the total length of the original data before subcontracting. (If compressed, that's the total length after compression; if not, that's the length of the original data)	Each time a new content is set, the packet id starts from 0, where 0 is the starting packet for that transfer, and then increments in sequence	The data length of the largest packet. The default is 1024.		0x00 and all the preceding data, XOR in sequence by byte, the final value obtained

Note:

1. Each packet data needs to be accompanied by a message type. The data sent is equivalent to: type+package.
2. The sent package data (type+package) all need to be converted through data format conversion for the device to recognize it normally.
3. Each time a package data (type+package) is sent, the device responds, and the APP needs to handle the reply message.
4. The app can only start the next action (transferring the next package, retransmitting data, showing transfer failure, etc.) after receiving the reply.

Response:

<type> Message type, one byte.

The packet id corresponding to <ack> transmits the result as follows:

Packet Type	Package id	error_code
1byte	2byte	1byte
Default to 0	0~0xFFFF	The values are as follows: 0x00 transfer successful Other transmission failures (retransmit all data)

When the APP is transmitting, it needs to consider the timeout. If no reply is received within 5 seconds after each packet is sent, it indicates that the transmission has timed out and needs to be retransmitted (restart the transmission). If the retransmission fails more than three times, it indicates that the setting doodle data failed this time.

Note: The reply data also meets the data format requirements, so reverse parsing of the reply data is necessary.

6、 Data compression

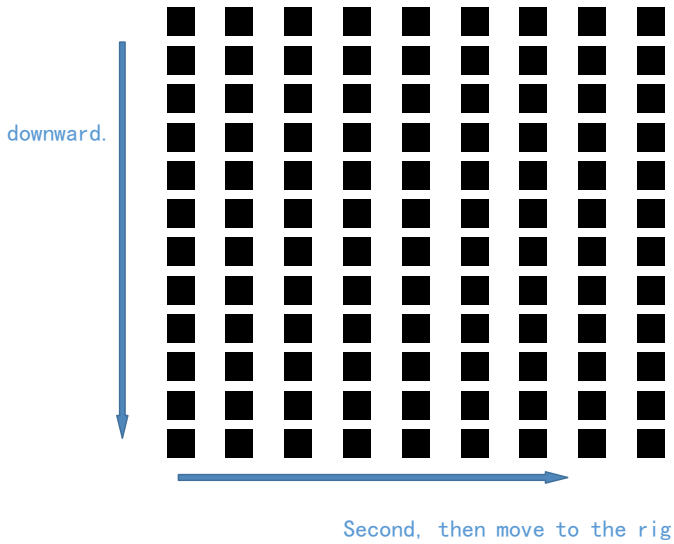
The data needs to be compressed once before the data packet is sent to reduce the amount of data sent, so that the data can be sent more quickly. Here, the lzss compression algorithm is used, which has a high compression ratio for data with high repeatability, and the data of doodles, animations, and text on the display screen have high repeatability, so it can generally achieve a compression ratio of 60% to 85%. The reference code is shown in the lzss.js file.

7、 Introduction to Grayscale

Grayscale is the level of brightness that can be adjusted for the color components supported by the display screen. For example, full-color 16 grayscale indicates that the device is an RGB device with 16 levels of brightness adjustment for the RGB components. Then each color component has a range of 0 to 15. Depending on the permutation and combination, then a total of 16*16*16=4096 colors can be represented.

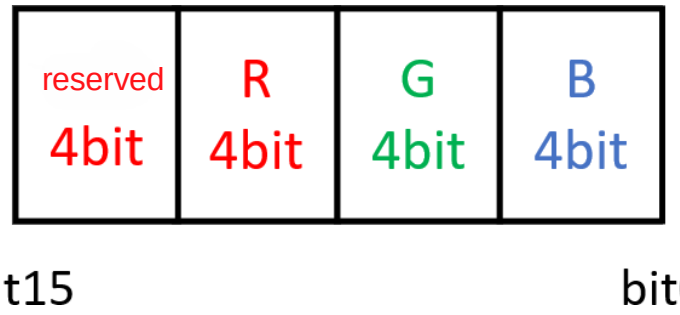
There are 256 levels of grayscale on the APP, and the data needs to be converted, with each component data value divided by 16.

8、 Data modulo method



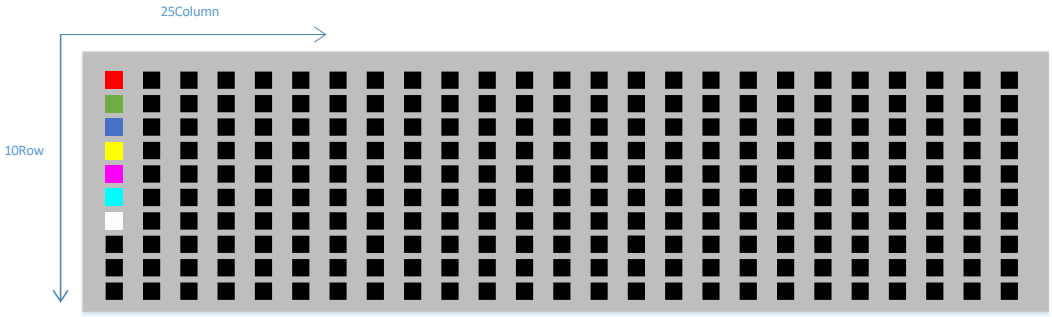
The modulus is taken as follows: top and bottom first, then left and right, with high bits in front.

9、 Data arrangement



All data is composed of three RGB components, with colors encoded according to RGB444, that is, the color of each point is represented by 2 bytes (a total of 16 bits), and each color component of each point occupies 4 bits. Bit0 to bit3 is the blue component, bit4 to bit7 is the green component, bit8 to bit11 is the red component, and bit12 to bit15 reserves all zeros. For example, for a 16x64 resolution full-color screen, then one frame of data =16x64x2=2048 bytes.

Example:



Take the figure above as an example. Each point is solid color, that is, the brightness is at maximum 0x0E. The representation of each point in each column:

First column:

Point 1, red: Red component 0x0E, green component 0x00, blue component 0x00, corresponding data 0x0E00

The second point, green: red component 0x00, green component 0x0E, blue component 0x00, corresponding data 0x00E0

Point 3, blue: Red component 0x00, green component 0x00, blue component 0x0E, corresponding data 0x000E

Point 4, yellow: Red component 0x0E, green component 0x0E, blue component 0x00, corresponding data 0x0EE0

Point 5, purple: Red component 0x0E, green component 0x00, blue component 0x0E, corresponding data 0x0E0E

Point 6, cyan: Red component 0x00, green component 0x0E, blue component 0x0E, corresponding data 0x00EE

Point 7, white: Red component 0x0E, green component 0x0E, blue component 0x0E, corresponding data 0x0EEE

Points 8 to 10, black: red component 0x00, green component 0x00, blue component 0x00, corresponding data 0x0000

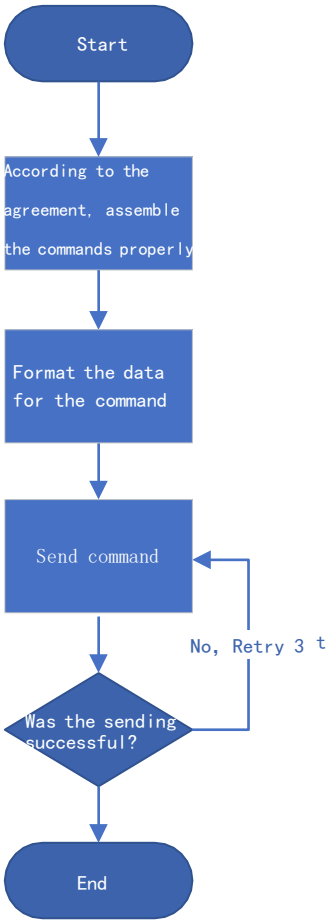
Data in the first column:

0E 00 00 E0 00 0E 0E E0 0E 0E 00 EE 0E EE 00 00 00 00 00 00

All data in other columns are 0

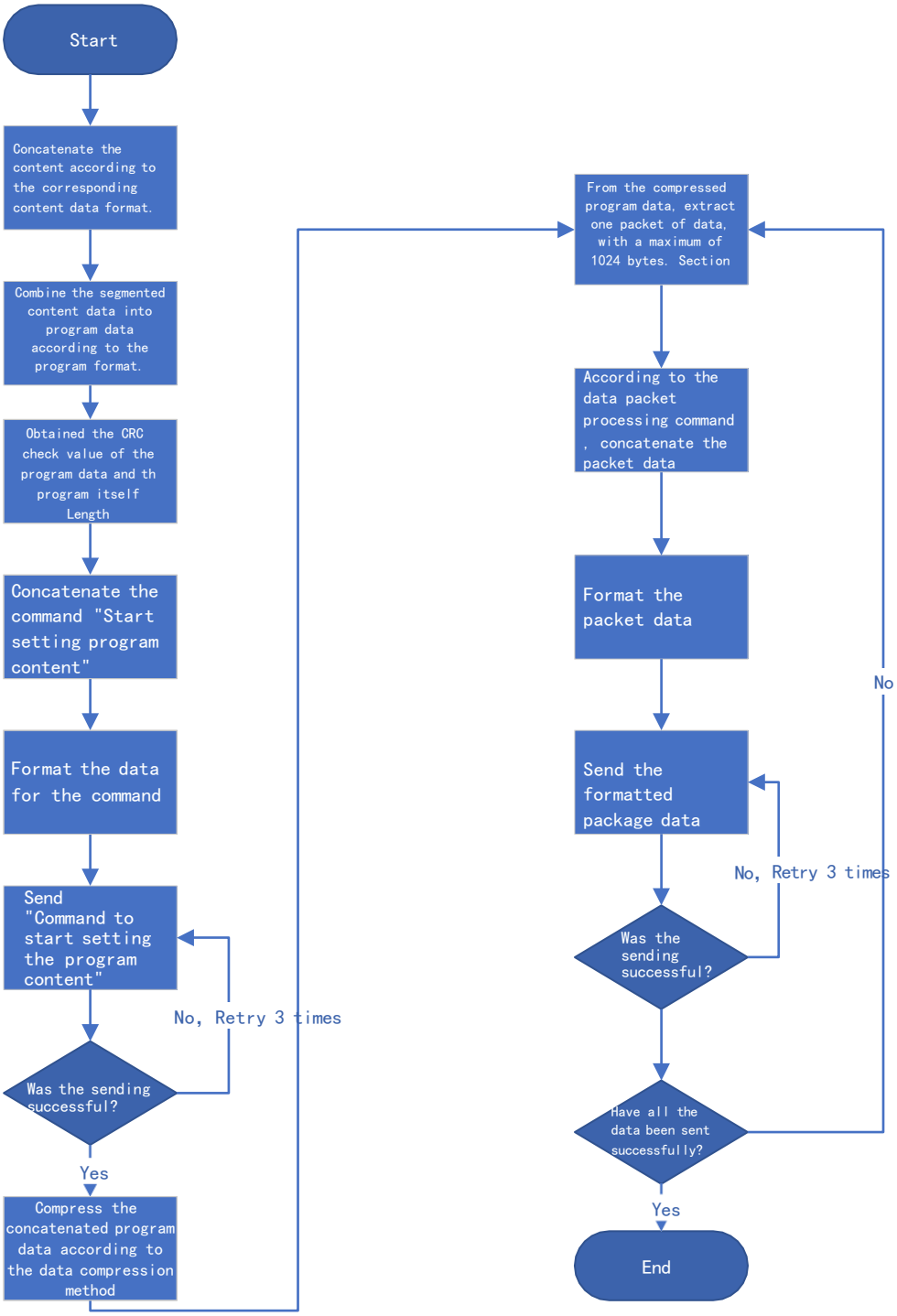
10、 Data sending process

10.1 General data sending process



10.2 Sending process with compression and subcontracting

Take the sending program process as an example, the OTA upgrade process is the same.



11、 Interaction Protocol

11.1 Mobile phone music rhythm mode (Microphone mode)

Command description: When the application is playing music or in the microphone state, extract the music

rhythm, convert it into bar data, and send it to the module.
Command format:

Direction	Type	Params	Response
APP→LIGHT	0x01	<type><param>	no

Data note:

<0x01> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment

<type> Music mode type, one byte, values refer to the Appendix - Music Mode section

<param> The parameter corresponding to the music mode, 8 bytes, < Column 1 height >< Column 2 height > .....< Column 8 height >

Note: Music rhythm data should not be sent out too quickly, with intervals of 50ms to 100ms.

11.2 Start setting up the program content

Command description: Through this command, start the transmission of program content. This command must be called before setting the program content.
Command Format:

Direction	Type	Params				Response
APP→LIGHT	0x02	Parameter Name	Length	Description	Notes	<0x02><Ack>
		<pro-data-crc>	4byte	The 32-bit CRC check of the current program data (the check before compression)	The value obtained by performing CRC32 checksums on the program data	
		<pro-data-len>	4byte	The length of the current program data (before compression)		
		<pro-index>	1byte	The sequential position of the current program in the list of programs to be sent this time	Value: 0 to n-1. For example, if two programs are to be sent, the first program to be sent will be numbered 0, and the second program to be sent will be numbered 1	
		<pro-num>	1byte	How many programs will be sent in total	Range: 1 to n The number of programs currently supported	

					needs to be obtained through the device.		
		<play-cnt>	1byte	Program plays.	Take values 1 to 255		

Data description:

<002> The message type occupies one byte and is always the first byte of a Bluetooth data segment.

Response:

<0x02> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< Ack> Response status, 0- success, no local copy, program content needs to be transmitted; 1- There is a local copy, no need to transfer the content of this program, directly prompt sending successful and start sending the next program; Others - Failed.

Notes:

1） Program data transmission can only begin when the response status is successful.

2） When the device successfully receives the command, it will first look locally for a copy of the program. If it does, there is no need to repeat the transmission.

For the CRC32 verification algorithm, see Appendix CRC32 Verification Algorithm.

11.3 Set up program content

Command Description: With this command, you can set the content of each program, and the content of each program can include multiple text, doodles, animations, etc.

Command Format:

Direction	Type	Params				Response
APP→LIGHT	0x03	Parameter Name	Length	Description	Notes	<0x03><ack>
		<reserved>	8byte	Reserved bytes	Default to 0	
		<pro-content-num>	1byte	How much content does the show contain	Range: 1 to n	
		<reserved>	1byte	Reserved bytes	(Originally for program plays)	
		<pro-cnt-data-1>	nbyte	Data corresponding to the content (the lower the rendering level Z where the content is placed ahead)	It can be text, doodles, animations. Refer to the data format corresponding to the content	
		……				
		<pro-cnt-data-n>	nbyte	Data corresponding to the content	It can be text, doodles, animations. Refer to the data format	

				(the lower the rendering level Z where the content is placed ahead)	corresponding to the content	

Data description:

<0x03> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment

< Params > This data needs to be compressed first and then subpackaged. See Data Compression, Data Subcontracting for details.

Response:

Reference subcontracting response data data subcontracting processing.

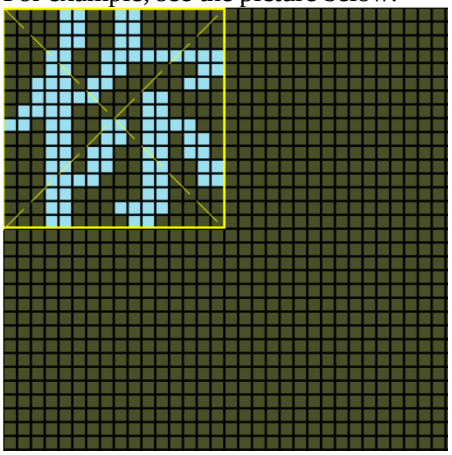
11.3.1 Text content data format

Note: The background color interval field is supported when the device version number is greater than 32 and less than 255.

Parameter name	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Include 4 bytes of <content-size> itself
<content-type>	1byte	Indicates the type of the content	The value is fixed to 0x01
<reserved>	7byte	Reserved bytes	Default to 0
<blend-type>	1byte	The way this content is displayed, mixed with other levels of content	Value: 0- Mixing (the highest level does the mixing operation based on transparency and the lowest level) 1- Overlay (the highest level will directly overwrite the lowest level)
<show-start-column>	2byte	This content shows the starting column, equivalent to the x-coordinate	Value range: 1 to the number of lines on the display minus the number of lines of one text (at least one text should be displayed)
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	Range: 1 to the number of display columns minus the number of columns for one text (at least one text should be displayed)
<show-width>	2byte	This content shows the width	Range of values: The width of a text to the width of the display screen
<show-height>	2byte	This content shows height	Value range: Height of a text to display height
<show-mode>	1byte	Display mode	Reference for values: "Appendix - Display Mode" section
<show-speed>	1byte	Display speed (corresponding speed for display mode)	Value: 1 to 255 (the higher the value, the faster the speed)
<stay-time>	1byte	Dwell time (the dwell time after one-screen display is completed)	Value: 0 to 255 seconds (this parameter is invalid in continuous movement mode)
<movespace>	2byte	Movement interval	This parameter takes effect only when the

			content moves continuously to the left or right. Otherwise, the default is the current content width. Move interval values range: 0 to display content width (default to display content width)
<text-num>	2byte	Number of characters	Up to 150 characters (Note: one emoji takes up two characters)
<text-all-wide>	4byte	The sum of the widths occupied by all the text	
The following is an array of data for each text, arranged in order			
<text-wide>	1byte	The width occupied by the first text	
<text-type>	1byte	The type identifier of the first text	0- represents monochrome text, 1- represents polychrome text (expression)
<text-data>	nbyte	The display data of the first text	If it's monochrome text, just display the data If it's multicolor text (emoji), represent the text data as a doodle.
…..			
<text-wide>	1byte	Width occupied by the NTH character	
<text-type>	1byte	The type identifier of the NTH character	0- represents monochrome text, 1- represents polychrome text (emoji)
<text- data>	nbyte	Display data for the NTH text	If it's monochrome text, only display the data If it's multicolor text (emoji), represent the text data as a doodle.

Note: Since the text can be selected in different sizes, the position and size of the content displayed here should be adjusted according to the size of the text.
For example, see the picture below:



In a 32x32 display area, if 16x16 text size is selected, the text content cannot fill the entire area, so the display height should not be 32 at this time, otherwise it will cause the display to be messy. Instead, 16 should be used. If the user also selects alignment, then the position where the text content is displayed needs to be adjusted accordingly.

Text notes:

1. Get the way the text displays data

   There are two ways to get text display data:

   1） One is to directly provide a dot matrix font library, but in this way the font library size is fixed and cannot be scaled arbitrarily, but it can be modified and optimized for a specific text.

   2） The second way is to scale the text to a specified size based on the vector font library, and then perform dot matrix processing to obtain the corresponding display data. In this way, the size of the text can be scaled arbitrarily.

2. Example:

   1) Font library method

   The font library is uniformly Unicode (UCS-2) font library, supporting common simplified Chinese, traditional Chinese, Korean, Japanese, Latin, etc. It is a character encoding scheme that accommodates almost all characters and symbols in the world, so it is relatively reasonable to use it as a font library. Unicode (UCS-2) encoding uses two bytes to represent a character, so it can accommodate more than 60,000 characters in total. The font size varies, and so does the number of bytes a character occupies for display.

   The number of bytes occupied = ((fontlibSize +7)/ 8)* fontlibSize (fontlibSize is the size of the dot matrix font, and the "/" represents divisibility, taking only the integer part)

   Then read the bitmatrix data from the font file through the Unicode encoding of the characters.

   Location: offset = Unicode encoding of the character * number of bytes occupied

   Then read the number of bytes that occupy the length of bytes. There is no need to distinguish languages; just uniformly convert to Unicode encoding.

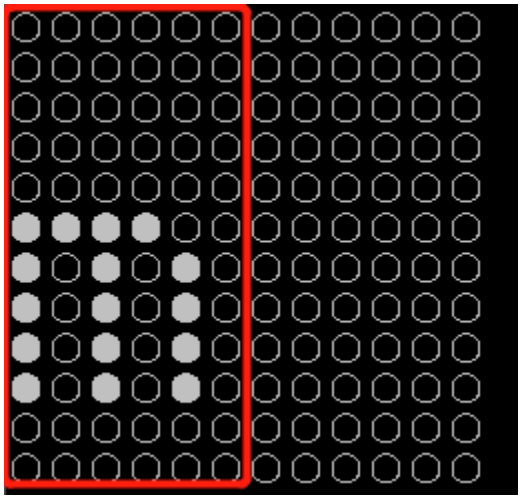
   For example, in the Unicode12 dot matrix font library, the font size of the dot matrix font library is 12, each character occupies 24 bytes, and the positioning formula is: the Unicode encoding of the character * 24, the number of bytes read each time is 24.

   The font file is a binary file that arranges the display data of the characters in the order of Unicode encoding, so you only need to read the font file in the form of a byte stream, note that the interface for operating the file varies from platform to platform.

3. Character display whitespace handling

   Since some characters themselves occupy only a few columns of dot matrix data, and the rest of the dot matrix data is all blank, which is equivalent to too much spacing, the extra blank dot matrix data needs to be removed, leaving only one column blank as the spacing between characters.

   For example, the dot matrix data of m is as follows:



(And the case where the text is on the right)

There are 12 columns of data, and only 5 are valid data, plus a blank column as an interval, so there are only 6 valid data. The remaining blank columns need to be deleted.

The display data corresponding to m is as follows:

unsigned char data[] = {

X04 x07 0, 0 xc0, 0, 0 x00, 0 x07, 0 xc0, 0 x04, 0 x00, 0 x03, 0 xc0, 0 x00, 0 x00,

0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00, 0 x00,

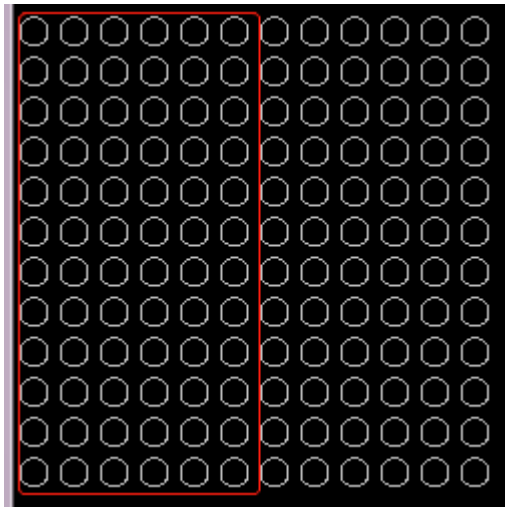
};

It is equivalent to traversing forward from the end of the data array, with every two bytes as a group of data, until one of the two data is not zero (equivalent to the column not being an empty column, that is, traversing to m's 4 columns (starting from column 0)), then copying the data from columns 0 to 4, and padding an empty column at the end of the array as an interval. The final result of m data is:

X04 xc0 x07 {0, 0, 0, 0 x00, 0 x07, 0 xc0, 0 x04, 0 x00, 0 x03, 0 xc0, 0 x00, 0 x00}

4. There are cases where all the data read from the font library is handled

   For example: Spaces. In such cases, you need to remove half of the data and leave only half.



When you encounter empty data, just leave half of the empty data.

Reference java code for optimizing intervals:

```
/**
 * Function: Optimize the display spacing between characters and remove the empty columns on the right side of each character.
 * Input: The display data corresponding to each character.
 * Output: Return the optimized display data.
 */
private static byte[] checkFontData(byte[] data){
    byte[] result = null;
    int copyLen = -1;

    // Traverse from the end of the array backward until the column is no longer empty.
    for (int i=data.length-1; i>0; i-=2) {
        if (data[i] !=0 || data[i-1] != 0) {
            copyLen = i+1;
            break;
        }
    }

    // All the data are not all zeros.
    if (copyLen > 0) {
        result = new byte[copyLen+2]; // A blank column needs to be added.
        //
        for (int i=0; i<copyLen; i++) {
            result[i] = data[i];
        }
        result[copyLen] = 0;
        result[copyLen+1] = 0;

        return result;
    }
    // All the data are all zeros. They might be spaces or other empty symbols. In that case, we only need to remove half of the data.
    if (copyLen < 0) {
        result = new byte[data.length/2];
        for (int i=0; i<result.length; i++) {
            result[i] = 0;
        }

        return result;
    }

    return data;
}
```

11.3.2 Doodle content data format

Parameter Name	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Include 4 bytes of <content-size> itself
<content-type>	1byte	Indicates the type of the content	The value is fixed to 0x02
<reserved>	7byte	Reserved bytes	Default to 0

<blend-type>	1byte	The way this content is displayed, mixed with other levels of content	Value: 0- Mixing (the highest level does the mixing operation based on transparency and the lowest level) 1- Overlay (the highest level will directly overwrite the lowest level)
<show-start-column>	2byte	This content displays the starting column, which is equivalent to the x-coordinate	Range: 0 to display width -1 (beyond will not be rendered)
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	Range: 0 to display height -1 (beyond will not be rendered)
<show-width>	2byte	This content shows width	Value range: 1 to display width (the starting column + width cannot exceed the display width)
<show-height>	2byte	This content shows height	Value range: 1 to display height (starting row + height cannot exceed display height), height will affect the number of bytes per column of data
<show-mode>	1byte	Display mode	Reference for values: "Appendix - Display Mode" section
<show-speed>	1byte	Show speed	Value: 1 to 255
<stay-time>	1byte	Dwell time (the time spent in one screen after completion)	Value: 0 to 255 seconds (this parameter is invalid in continuous movement mode)
<data-size>	4byte	Doodle the total length of the dot matrix data	
<data>	nbyte	The corresponding doodle dot matrix displays the data	

The data format refers to the data arrangement method.

11.3.3 Animation content data format

Animation data consists of doodle data frame by frame, with fixed time intervals between each frame, just like GIFs.

Parameter Names	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Include 4 bytes of <content-size> itself
<content-type>	1byte	Indicates the type of the content	The value is fixed to 0x03
<protocol-ver>	1byte	Protocol Version Number	Fixed to 0x01
<reserved>	6byte	Reserved bytes	Default to 0

<blend-type>	1byte	The way this content is displayed, mixed with other levels of content	Value: 0- Mixing (the highest level does the mixing operation based on transparency and the lowest level) 1- Overlay (the highest level will directly overwrite the lowest level)
<show-start-column>	2byte	This content shows the starting column, equivalent to the x-coordinate	Value range: 0 to display width -1 (any value exceeding this range will not be rendered)
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	Range: 0 to display height -1 (beyond will not be rendered)
<show-width>	2byte	This content shows width	Range of values: The width of a text to the width of the display screen
<show-height>	2byte	This content shows height	Value range: Height of a text to display height
<rev>	1byte	Reserved bytes	The default is 0.
<frame-num>	2byte	Frames	Up to 40 frames
<frameEveInterval>	nbyte	A collection of individual display times per frame	The independent display time per frame, occupying 2 bytes. Arrange them in sequence.
<frameData>	nbyte	A collection of frame data (corresponding to frameNum frames)	The size of each frame data must be consistent with the size of the display area, and frameData must be an integer multiple of the frame. The data composition varies depending on the device type.

11.3.4 Border content data format

Parameter Name	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Include 4 bytes of <content-size> itself
<content-type>	**1byte**	Indicates the type of the content	The value is fixed to 0x04
<reserved>	**7byte**	Reserved bytes	Default to 0
<blend-type>	1byte	The way this content is displayed, mixed with other levels of content	Value: 0- Mixing (the highest level does the mixing operation based on transparency and the lowest level) 1- Overlay (the highest level will directly overwrite the lowest level)
<show-start-column>	2byte	This content shows the starting column,	Range: 0 to display width -1 (beyond will not be rendered)

		equivalent to the x-coordinate	
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	Range: 0 to display height -1 (beyond will not be rendered)
<show-width>	2byte	This content shows width	Value range: 1 to display screen width (starting column + width exceeding the display screen width will be cropped)
<show-height>	2byte	This content shows height	Value range: 1 to display height (starting row + height beyond display height will be cropped), height will affect the number of bytes per column of data
<border-type>	1byte	Border display effect	0- Reserved 1- Rotate clockwise 2- Rotate counterclockwise 3- Flashing 4- Still
<border-speed>	1byte	Border change speed	Value: 1 to 255 (The larger the value, the faster the speed)
<border-height>	1byte	Border content height	Value range: 1 to half of the display height. For displays that are generally 16 in height, the maximum border height is 1; otherwise, too much of the display will be blocked.
<data-size>	2byte	The total length of the border data	
<data>	nbyte	Borders show data	Data composition is based on data modulo and data arrangement.

Border data only requires one segment of data, and the device will automatically repeat this segment of display content.

For the built-in border effect, please refer to the following file. The speed is at the user's discretion.

Full-color display with built-in border effect (CoolLEDU)

Example 1:

If you want to achieve a border with a height of 1 row arranged as "red and black" (the subsequent dots are repeated and no data is given) as shown above, follow the data and color arrangement rules.

Red: 0x0F,0x00

Black: 0x00,0x00

So

<border-height> is: 1

<data-size> is: 4

<data> is: 0x0F,0x00, 0x00,0x00

Example 2:

If you want to implement a border with a height of 2 rows as shown above.

In the way the data is modulated, first vertically and then horizontally, then the modulated arrangement:

Red, purple, black, black (no need to give duplicate data later)

According to the data arrangement rules, color representation:

Red: 0x0F,0x00

Purple: 0x0F,0x0F

Black: 0x00,0x00

So:

<border-height> is: 2

<data-size> is: 8

<data> is: 0x0F,0x00, 0x0F,0x0F, 0x00,0x00, 0x00,0x00

According to the above rules, users can actually customize various styles of border effects. Here, for the convenience of users, choose the built-in border effect. If needed later, add the border effect or add the function of customizing the border effect.

11.3.5 Text color data format

This feature works only for text, and this segment of data must be placed before the text content data. In fact, text color and text content data are bound together, otherwise it won't work. The text effect can only be either the Color text effect or the custom text effect, not both.

11.3.5.1 Color Text effects

Parameter Names	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Four bytes including <content-size> itself
<content-type>	1byte	Indicates the type of the content	The value is fixed to 0x05
<reserved>	7byte	Reserved bytes	Default to 0
<show-start-column>	2byte	This content shows the starting column, equivalent to the x-coordinate	Take the same value as the starting row in the text content data
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	The value is the same as the starting column in the text content data
<show-width>	2byte	This content shows width	The values are the same as the display width in the text content data
<show-height>	2byte	This content shows height	The values are the same as the displayed height in the text content data

<textcolor-type>	1byte	Text color effects	0- Reserved 1- Scroll horizontally (left, right) 2- Horizontal static (none, direction defaults to 0) 3- Vertical scrolling (up, down) 4- Vertical relative rolling (towards the middle, towards both sides 5- Jump (None, direction defaults to 0) 6- Horizontal overlay (left, right) 7- Horizontal diagonal scrolling (left, right) 8- Windmill spin (left, right)
< textcolor -speed>	1byte	Text color effect change speed	Value: 1 to 255 (the larger the value, the faster the speed)
< textcolor-dir>	1byte	Text color effects show direction	0- Left 1- To the right 2- Up 3- Down 4- Towards the middle 5- To both sides Different effects support different directions
< reserved >	1byte	Reserved	Default to 0
<data-size>	2byte	Color data length	
<data>	nbyte	Color data	The color arrangement is formed according to the data arrangement

For a built-in text color effect, please refer to the following documentation. The speed can be adjusted by the user.

Full-color display with built -in text color mode effect (CoolLEDU)

Example:

The text color needs to be set to a seven-color horizontal static display effect. The seven colors are: red, yellow, green, cyan, blue and purple

According to the color arrangement rules:

Red: 0x0F, 0x00

Yellow: 0x0F, 0xF0

Green: 0x00, 0xF0

Cyan: 0x00, 0xFF

Blue: 0x00, 0x0F

Purple: 0x0F, 0x0F

So

<data-size> is: 0x0C (data length 12)

The < data > is: 0 x0f, 0 x00, 0 x0f, 0 xf0, 0 x00, 0 xf0, 0 x00, 0 XFF, 0 x00, 0 x0f, 0 x0f, 0 x0f

<textcolor-type> is: 2

< textcolor-dir> is: 0

According to the above rules, users can actually customize the text color effect. Here, for the convenience of users, choose the built-in effect. If needed later, add the effect or add the function of customizing the effect.

11.3.5.2 Customize the text effect

Custom text effects, that is, users can customize the color of each text.

Parameter name	Length	Description	Notes
<content-size>	4byte	The total length of all the data in that segment	Include 4 bytes of <content-size> itself
<content-type>	1byte	Indicates the type of the content	The value is fixed to 0x06
<reserved>	5byte	Reserved bytes	Default to 0
<movespace>	2byte	Movement interval	The values are the same as the movement intervals in the text content data
<show-start-column>	2byte	This content shows the starting column, equivalent to the x-coordinate	Take the same value as the starting row in the text content data
<show-start-row>	2byte	This content shows the starting row, equivalent to the y-coordinate	Take the same value as the starting column in the text content data
<show-width>	2byte	This content shows width	The values are the same as the display width in the text content data
<show-height>	2byte	This content shows height	The values are the same as the displayed height in the text content data
<show-mode>	1byte	Display mode	The values are the same as the display mode in the text content data
<show-speed>	1byte	Display speed (corresponding speed for display mode)	The values are the same as the display speed in the text content data
<stay-time>	1byte	Dwell time (the time spent in one screen after completion)	Take the same value as the dwell time in the text content data
< reserved >	1byte	Reserved	Default to 0
<text-num>	2byte	Number of	Up to 150 characters are supported

		characters	
<text-all-wide>	2byte	The sum of the widths occupied by all the text	
<text-eve-wide>	nbyte	A collection of widths occupied by each word	The width of each text is expressed in one byte, and the length of this data =<text-num>
<text-eve-color>	nbyte	A collection of colors for each text	Text colors can be arranged in the order of data arrangement. [If the text is multicolor (emoji), just use the default red to represent it]

Example:

For example, this setting has a total of 7 text, the text content is "Juntong Technology Flexible Screen", each text size is 32, and the text colors are "red green Blue Yellow cyan Purple White".

According to the color arrangement rule:

Red: 0x0F, 0x00

Green: 0x00, 0xF0

Blue: 0x00, 0x0F

Yellow: 0x0F, 0xF0

Cyan: 0x00, 0xFF

Purple: 0x0F, 0x0F

White: 0x0F,0xFF

Purple: 0x0F, 0x0F

So:

<text-num> : 00 07

< Text-all-wide > : 00E0 (the sum of the widths of each text)

<text-eve-wide> : 20 20 20 20 20 20 20 20

<text-eve-color> : 0x0F, 0x00, 0x00, 0xF0, 0x00, 0x0F, 0x0F, 0xF0, 0x00, 0xFF, 0x0F, 0x0F, 0x0F

Note: The amount of data would be 17 times larger if all colors were represented by doodles. And in actual use, text is dominant and emojis are secondary. So here the color representation of text and emojis is distinguished.

11.4 Set the display brightness

Command Description: The brightness of the display can be set by this command.

Command Format:

Direction	Type	Params	Response
APP→LIGHT	0x04	<bn>	<0x04><bn>

Data note:

<0x04> Message type, occupying one byte, and it must be the first byte of a Bluetooth data segment.

< bn > shows brightness, occupies one byte, value range: 0 to 255

Response:

<0x04> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< bn > shows brightness, occupies one byte, value range: 0 to 255

11.5 Set the switch status

Command Description: This command is used to set the on-off status of the device
Command Format:

Direction	Type	Params	Response
APP→LIGHT	0x05	<status>	<0x05><status>

Data description

<0x05> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< status > Switch status, one byte, values: 0- off, 1- on

Response:

<0x04> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

<status> The current on-off status, occupies one byte, values: 0- off, 1- on

11.6 Play the content of the specified program list

Command description: With this command, you can play a particular item from the specified program list. If there is no corresponding content for that index, play the specified content according to the Settings.
Command format:

Direction	Type	Params	Response
APP→LIGHT	0x07	<index>	<0x07><index>

Data note:

<0x07> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< index > The number of the program, occupying one byte, values 0 to 8, 0xFF indicates automatic loop playback of all programs

Response:

<0x07> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< index > The list of programs currently being played, occupying one byte, with values ranging from 0 to 8,0 xFF indicates automatic looping of all programs

11.7 Delete the content of the specified program list

Command description: With this command, you can delete something from the specified program list. If the deleted program is the one currently playing: 1) Play the next program if autoplay is effective. 2) If auto-play is ineffective, play the default content. 3) Play the default program if all played programs are deleted.
Command format:

Direction	Type	Params	Response
APP→LIGHT	0x08	<index>	<0x08><index>

Data note:

<0x08> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< index > The number of that program, occupying one byte, values 0 to 8, 0xFF indicates deletion of all programs

Response:

<0x08> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< index > The number of the program, occupying one byte, values 0 to 8,0 xFF indicates deleting all programs

11.8 Set the flip status

Command Description: This command can be used to set the flip status of the displayed content on the device.
Command Format:

Direction	Type	Params	Response
APP→LIGHT	0x0C	<status>	<0x0C ><status >

Data note:

<0x0C> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< status > Flip status, one byte, values: 0-normal, 1-XY flip, 2-X flip, 3-Y flip

Response:

<0x0C > Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

< status > Flip status, one byte, values: 0-normal, 1-XY flip, 2-X flip, 3-Y flip

11.9 Get device information

Command description: Various status information of the device can be obtained through this command.
Command Format:

Direction	Type	Params	Response
APP→LIGHT	0x1F	no	<0x1F><resp>

Data Note:

<0x1F> Message type, occupies one byte, and must be the first byte of a Bluetooth data segment.

Response:

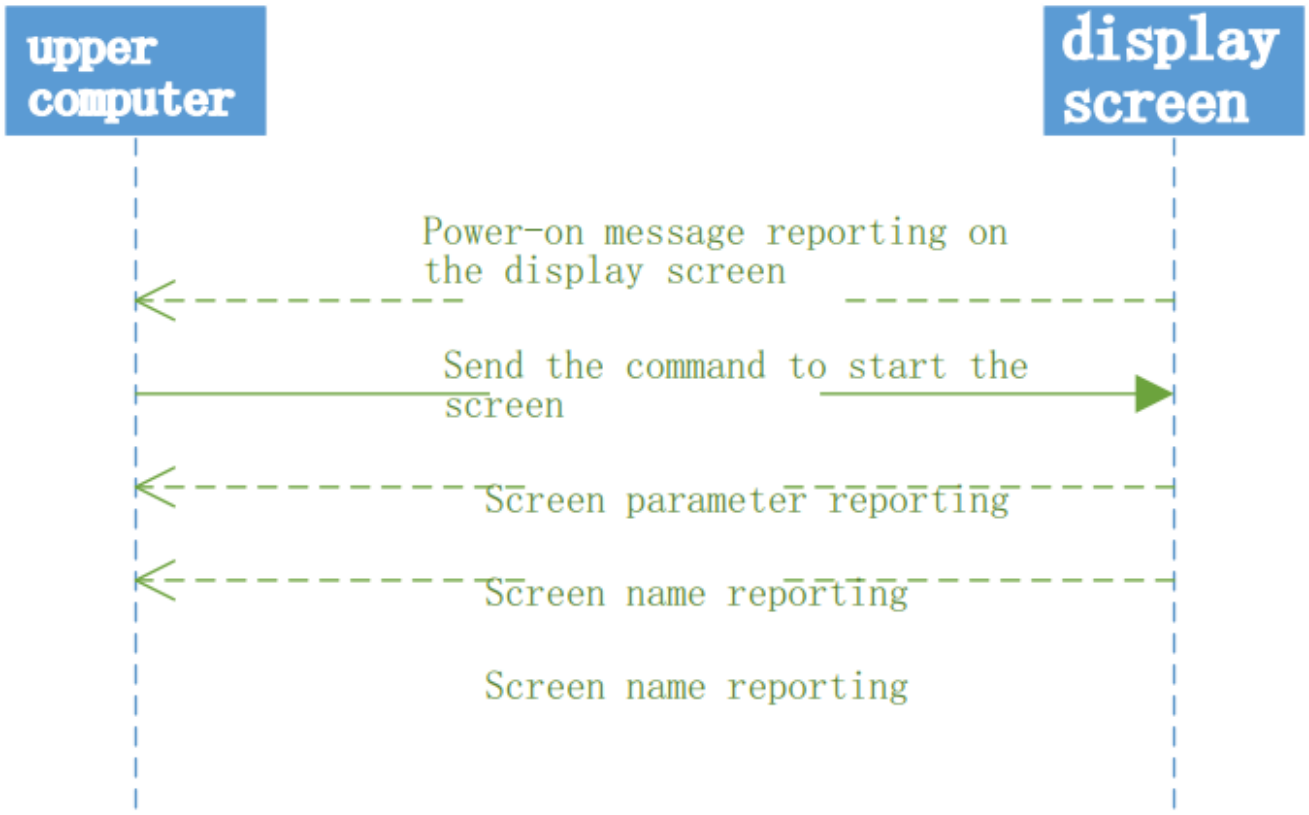
Parameter Name	Length	Description
<0x1F>	1byte	Message Type
< switch-s >	1byte	Device switch status, values: 0- off, 1- on
< bn >	1byte	Display brightness, range: 0 to 255
< flip >	1byte	Flip status, values: 0-normal, 1-XY flip, 2-X flip, 3-Y flip

Note: Ignore other fields returned.

12、 Startup Phase Protocol (serial port)

The commands for this phase are interactive commands for the screen startup phase. Serial port development must handle the interactive protocol for this part and send the commands in the specified order for the screen to start properly.

If it's a BLE secondary development, ignore the protocol content of this part.



12.1 Power-on report on the screen

Command description: This command is proactively reported to the host computer when the screen is powered on or restarted.
Command format:

Direction	Type	Params	Response
LIGHT→APP	0x32	<0x04>	<0x32><0x04><ack>

Data description:

<0x32> Message type, occupies one byte, and must be the first byte of a piece of data.
< 0x04> parameter, with the value: 0x04

Response:

<0x32> Message type, occupies one byte, and must be the first byte of a piece of data.
<0x04> parameter, with the value: 0x04
<ack> response status, 0- parsing successful, other - parsing failed

Note: When the host computer receives this command, if it needs to start the screen, it needs to send the "Start screen" command.

12.2 Screen parameter reporting

Command description: After receiving the "Start Screen" command from the host computer, the screen end will report the screen parameters within a short period of time (approximately 50ms).
Command format:

Direction	Type	Params	Response
LIGHT→APP	0x30	<ID-H><ID-L><RES-ROW><RES-C-H><RES-C-L><DEV-TYPE><DEV-VERSION>	<0x30><ack>

Data description:

<0x30> Message type, occupies one byte, and must be the first byte of a piece of data.
<ID-H> the high byte of the device ID, occupying one byte
< ID-l > the low byte of the device ID, occupying one byte
<RES-ROW> Screen resolution row count, one byte
<RES-C-H> The higher byte of the screen resolution column count, occupying one byte
<RES-C-L> The lower byte of the screen resolution column count, occupying one byte
<DEV-TYPE> device type, one byte, with a fixed value of 0x02
<DEV-VERSION> device version number, one byte, ranging from 0 to 255

Response:

<0x30> Message type, occupies one byte, and must be the first byte of a piece of data.
<ack> Response status, 0- parsing successful, other - parsing failed

Note: This parameter is not required for the upper computer and can be ignored.

12.3 Screen Name reporting

Command Description: The screen ends will report the screen name within a short time (approximately 50ms) after receiving the "Start Screen" command from the host computer. The host computer can distinguish different products by that name. The full-color display is called CoolLEDU

Command format:

Direction	Type	Params	Response
LIGHT→APP	0x31	<name-arr>	<0x31><ack>

Data description:

<0x31> Message type, occupies one byte, and must be the first byte of a piece of data.

< Name-arr > Screen name, up to 20 bytes.

Response:

<0x31> Message type, occupies one byte, and must be the first byte of a piece of data.

<ack> Response status, 0- parsing successful, other - parsing failed

Note: The upper computer does not require a name and can be ignored.

12.4 Startup screen

Command Description: You can start the screen with this command. The screen will start up normally only after the host computer sends the command.
Command format:

Direction	Type	Params	Response
APP→LIGHT	0x23	<0x01><0x00>	<0x23><FLAG>

Data description:

<0x23> Message type, occupies one byte, and must be the first byte of a piece of data.
< 0x01 > parameter, value: 0x01
<0x00> parameter, fixed value 0x00

Response:

<0x23> Message type, occupies one byte, and must be the first byte of a piece of data.

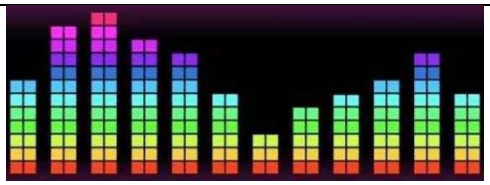
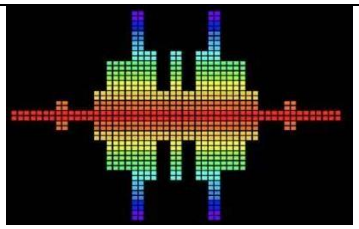
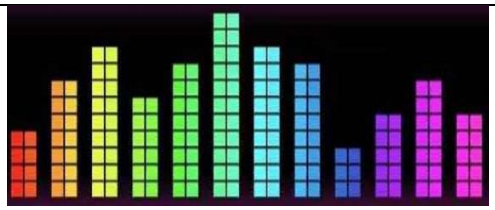
The <FLAG> flag occupies one byte, and the host computer does not need to care about the specific value of this variable.

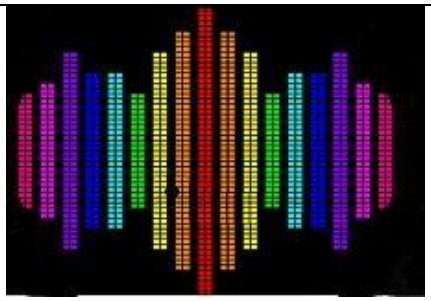
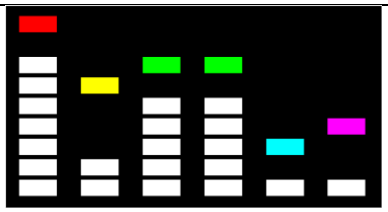
Note: The device will start setting screen-related parameters in a very short time after startup.

13、 Appendix

13.1 Music patterns

Different music rhythm effects, values and corresponding effects are as follows:

type	describe	View References
0x01	Seven colors light up vertically to one side, with each column of colors gradually changing from bottom to top in the order of red, yellow, green, cyan, blue, purple and red, and each row of colors is the same	
0x02	The seven colors light up vertically to the sides, with each column of colors gradually changing from the middle to the sides in the order of red, yellow, green, cyan, blue, purple, red, and each row of colors is the same	
0x03	Seven colors light up vertically to one side, with each column of colors gradually changing from left to right in the order of red, yellow, green, cyan, blue, purple, red, and each column of colors is the same	

0x04	Light up horizontally from the middle to the sides, with colors in each column gradually changing from the middle to the left and right in the pattern of red, yellow, green, cyan, blue, purple, red, symmetrical left and right, with the same color in each column	
0x05	White with landing points moving to one side pattern, the color of the landing points changing randomly	

13.2 Display mode

Different display modes for text or graffiti:

Take values	Pattern Name	Remarks
1	Static	Content static is displayed in a loop frame by frame
2	Continuous left shift	Content moves to the left in a loop display
3	Continuous rightward shift	Content moves to the right in a loop display
4	Up	Content moves up loop display
5	Down	Content moves down loop display
6	Piling up	The content is piled down line by line
7	Scroll	The content unfolds in a loop from the middle to the sides like a scroll
8	Flickering	Each frame of content flashes once in a loop for display
9	Pan left	The content moves to the left in a loop display, pausing for a specified time for each screen shown
10	Pan right	The content moves to the left in a loop display, pausing for a specified time for each screen

		shown
11	Left overwrite	
12	Cover to the right	
13	Horizontal interweaving	

13.3 CRC32 check algorithm

To fully leverage the CRC32 verification capabilities of the device hardware, the host computer needs to use the same verification algorithm as the device. The CRC32 verification algorithm is as follows:

crc.js

14、 Example

14.1 Set the device switch status to on

Now you want to set the display to be on

1. According to the protocol, the original data is:
   05 01
2. The data needs to be formatted, and after formatting (the final data sent) :
   01 00 02 06 05 02 05 03
   Field description:
   01 // Start byte
   00 02 06 // Data length, forced conversion since the second byte of the data length is 02
   05 02 05 // data, mandatory conversion is required according to the rule
   03 // End byte
3. Device response
   01 00 02 06 05 02 05 03
   The response data needs to be parsed according to the protocol of data formatting.
   The parsed data is: 05 01