

Chain RGB UART Control Protocol

(Documentation V1.0)

1. Basic Information

Item	Description
Communication method	UART / serial port asynchronous communication
Device	Chain RGB
SKU	U218
Baud Rate	115200 bps
Data Bits	8 bit
Stop bit	1 bit
Parity	None
Flow Control	None
Document version	V1.0

2. Data Frame Format

All packets start with `0xAA 0x55` and end with `0x55 0xAA`. Multi-byte length fields are entered low byte first.

2.1 Request Frame Format

Field	length	Description
Header	2	Fixed to <code>0xAA 0x55</code>
Length	2	Number of bytes from <code>Index</code> to <code>CRC</code> , low byte first
Index	1	The index ID of the device in the Chain link
Cmd	1	request command code
Req Data	n	Request parameters; if there are no parameters, the length is <code>0</code>
CRC	1	See "3. Verification Instructions"
Tail	2	Fixed to <code>0x55 0xAA</code>

Request Frame:
AA 55 <Length_low> <Length_high> <Index> <Cmd> <Req Data...> <CRC> 55 AA

2.2 Response Frame Format

Field	length	Description
Header	2	Fixed to 0xAA 0x55
Length	2	Number of bytes from Index to CRC , low byte first
Index	1	The index ID of the device in the Chain link
Cmd	1	Consistent with request Cmd
Resp Data	n	Response data, the structure is defined by the corresponding command
CRC	1	See "3. Verification Instructions"
Tail	2	Fixed to 0x55 0xAA

Response Frame:
AA 55 <Length_low> <Length_high> <Index> <Cmd> <Resp Data...> <CRC> 55 AA

2.3 Response Data Convention

situation	response rules
Reply with status code	Operation_status = 0 means the operation failed, Operation_status = 1 means the operation was successful
Reply without status code	The judgment method of success or failure shall be based on the "Notes" of the corresponding instruction.

3. Checksum

Item	rules
Check field	CRC
Checksum range	Starting from Index , accumulate to the last byte of Data
Checksum algorithm	Excluding the packet header, packet trailer, Length and CRC, the remaining bytes are accumulated and the lower 8-bits are taken.
Empty Data	CRC = (Index + Cmd) & 0xFF
Checksum failure	discard the frame

CRC = (Index + Cmd + Data[0] + ... + Data[n]) & 0xFF

3.1 Checksum Example

```
#include <stdint.h>
#include <stdint.h>

uint8_t chain_calc_crc(uint8_t index, uint8_t cmd,
                      const uint8_t *data, size_t data_len)
{
    uint16_t sum = index + cmd;

    for (size_t i = 0; i < data_len; i++) {
        sum += data[i];
    }

    return (uint8_t)(sum & 0xFF);
}
```

4. Quick overview of commands

Cmd	Command name	Request Length	Reply Length	Request clarification	Response description	Details
0x10	Set display mode	0x0004	0x0004	Index_id (device index ID), Mode (display mode)	Operation_status	5.1
0x11	Get display mode	0x0003	0x0004	Index_id (device index ID)	Mode	5.2
0x30	Set pixels	0x0004+N um*3	0x0004	Index_id (device index ID), Num (number of pixels to be set), Coordinate_xy (pixel XY coordinates), Pixel_color (pixel color)	Operation_status	5.3
0x31	Full screen display cache refresh	0x0083	0x0004	Index_id (device index ID), Display_buffer (full screen display cache)	Operation_status	5.4
0x32	Get pixel value	0x0004+N um	0x0003+N um*2	Index_id (device index ID), Num (number of pixels to be obtained), Coordinate_xy (pixel coordinates)	Pixel_color (pixel color)	5.5
0x33	Get full screen display cache	0x0003	0x0083	Index_id (device index ID)	Display_buffer (full screen display buffer)	5.6
0x34	display characters	0x0007	0x0004	Index_id (device index ID), Char (ASCII code of the character to be displayed), Offset (displayed position offset), Color (character color)	Operation_status	5.7

Cmd	Command name	Request Length	Reply Length	Request clarification	Response description	Details
0x40	Show scrolling string	0x0009+String_length	0x0004	Index_id (device index ID), Scroll_mode (scrolling mode), Scroll_interval (scrolling interval), Color (character color), String_length (string length), char (character)	Operation_status	5.8
0x41	Get scrolling string	0x0003	0x0009+String_length	Index_id (device index ID)	Scroll_mode (scrolling mode), Scroll_interval (scrolling interval), Color (character color), String_length (string length), char (character)	5.9
0x42	Set scroll string status	0x0004	0x0004	Index_id (device index ID), Scroll_state (scroll state)	Operation_status	5.10
0x43	Set scroll string status	0x0003	0x0004	Index_id (device index ID)	Scroll_state (scroll state)	5.11
0xE0	Set screen rotation angle	0x0005	0x0004	Index_id (device index ID), Screen_Rotation (screen rotation angle), Save_to_flash (whether to save to internal Flash)	Operation_status	5.12
0xE1	Get screen rotation angle	0x0003	0x0004	Index_id (device index ID)	Screen_Rotation (screen rotation angle)	5.13
0xE2	Set screen brightness	0x0005	0x0004	Index_id (device index ID), Brightness (screen brightness), Save_to_flash (whether to save to internal Flash)	Operation_status	5.14
0xE3	Get screen brightness	0x0003	0x0004	Index_id (device index ID)	Brightness(screen brightness)	5.15
0xE4	clear screen	0x0003	0x0004	Index_id (device index ID)	Operation_status	5.16
0xF8	Query device unique UID	0x0004	0x0004+4/12	Index_id (device index ID), UID_Type	Operation_status、UID	5.17
0xF9	Query the upgrade program version number	0x0003	0x0004	Index_id (device index ID)	Bootloader_version	5.18
0xFA	Query the device software version number	0x0003	0x0004	Index_id (device index ID)	Firmware_version	5.19
0xFB	Query device type	0x0003	0x0005	Index_id (device index ID)	Device_type	5.20

Cmd	Command name	Request Length	Reply Length	Request clarification	Response description	Details
0xFC	Enumeration request, sent by the chain link change end device, and when the device is powered on, notifying the host to update the link device status	Device-initiated report	Device-initiated report	none	none	5.21
0xFD	Heartbeat packets, regular communication between chain devices, can self-discover whether they are end devices, and the host can also use heartbeat packets to determine whether there is a chain device connected.	0x0003	0x0003	none	none	5.22
0xFE	Enumerate to obtain the number of cascaded devices	0x0004	0x0004	Send_num (default 0, used to record the number of devices)	Receive_num (the value represents the number of devices)	5.23

5. Command Details

5.1 Set display mode - Cmd 0x10

Request data

Input parameters: Index_id (device index ID), Mode (display mode)

Request Frame:
AA 55 | 0x04 | 0x00 | Index_id | 0x10 | Mode | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:
AA 55 | 0x04 | 0x00 | Index_id | 0x10 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
1. 0: Operation failed

2. 1: Operation successful
2. Mode display mode (default 0)
1. 0: Pixel control mode (directly set pixel display)

2. 1: Scroll character mode (scrolling display string)

5.2 Get display mode - Cmd 0x11

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0x11 | CRC | 55 AA

Response Data

Return parameters: Mode (display mode)

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x11 | Mode | CRC | 55 AA

Things to note

1. Mode display mode (default 0)
 1. 0: Pixel control mode (directly set pixel display)
 2. 1: Scroll character mode (scrolling display string)

5.3 Set pixels - Cmd 0x30

Request data

Input parameters: Index_id (device index ID), Num (number of pixels to be set), Coordinate_xy (pixel XY coordinates), Pixel_color (pixel color)

Request Frame:

AA 55 | 0x04 + Num*3 | 0x00 | Index_id | 0x30 | Num | Coordinate_xy0 | Pixel_color_low0 | Pixel_color_high

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x30 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0: Operation failed
 2. 1: Operation successful
 3. 2: Mode mismatch (needs to be configured as pixel control mode)
2. Num range: 1 ~ 64
3. Coordinate_xy bit[7:6]: reserved bit[5:3]: X coordinate (0-7) bit[2:0]: Y coordinate (0-7)
4. Pixel_color uses RGB565 format 16-bit color: R(5-bit) + G(6-bit) + B(5-bit) bit[15:11] = Red bit[10:5] = Green bit[4:0] = Blue Pixel_color = (uint16_t)((Pixel_color_high << 8) | Pixel_color_low) Range 0~65535

5.4 Full-screen display cache refresh - Cmd 0x31

Request data

Input parameters: Index_id (device index ID), Display_buffer (full-screen display cache)

Request Frame:

AA 55 | 0x83 | 0x00 | Index_id | 0x31 | Display_buffer_low0 | Display_buffer_high0 | Display_buffer_low1 |

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x31 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0: Operation failed
 2. 1: Operation successful
 3. 2: Mode mismatch (needs to be configured as pixel control mode)
2. Display_buffer is in RGB565 format Data arrangement: The screen pixel color is from left to right and from top to bottom (row-first storage). The 8 pixels of row 0 are stored first, then the row 1, and so on. Display_buffer0 = (uint16_t)((Display_buffer_high0 << 8) | Display_buffer_low0) Range 0~65535

5.5 Get pixel value - Cmd 0x32

Request data

Input parameters: Index_id (device index ID), Num (number of pixels to be obtained), Coordinate_xy (pixel coordinates)

Request Frame:

AA 55 | 0x04 + Num | 0x00 | Index_id | 0x32 | Num | Coordinate_xy0 | Coordinate_xy1 | Coordinate_xy2 | Co

Response Data

Return parameters: Pixel_color (pixel color)

Response Frame:

AA 55 | 0x03 + Num*2 | 0x00 | Index_id | 0x32 | Pixel_color_low0 | Pixel_color_high0 | Pixel_color_low1 |

Things to note

1. Coordinate_xy bit[7:6]: reserved bit[5:3]: X coordinate (0-7) bit[2:0]: Y coordinate (0-7)
2. Pixel_color uses RGB565 format 16-bit color: R(5-bit) + G(6-bit) + B(5-bit) bit[15:11] = Red, bit[10:5] = Green, bit[4:0] = Blue Pixel_color = (uint16_t)((Pixel_color_high << 8) | Pixel_color_low) Range 0~65535

5.6 Get full-screen display cache - Cmd 0x33

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0x33 | CRC | 55 AA

Response Data

Return parameters: Display_buffer (full screen display cache)

Response Frame:

AA 55 | 0x83 | 0x00 | Index_id | 0x33 | Display_buffer_low0 | Display_buffer_high0 | Display_buffer_low1 |

Things to note

1. Display_buffer is in RGB565 format Data arrangement: The screen pixel color is from left to right and from top to bottom (row-first storage). The 8 pixels of row 0 are stored first, then the row 1, and so on. Display_buffer0 = (uint16_t)((Display_buffer_high0 << 8) | Display_buffer_low0) Range 0~65535

5.7 Display characters - Cmd 0x34

Request data

Input parameters: Index_id (device index ID), Char (ASCII code of the character to be displayed), Offset (displayed position offset), Color (character color)

Request Frame:

AA 55 | 0x07 | 0x00 | Index_id | 0x34 | Char | Offset | Color_low | Color_high | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x34 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0: Operation failed
 2. 1: Operation successful
 3. 2: Mode mismatch (needs to be configured as pixel control mode)
2. Char To display the ASCII code of characters, English letters, numbers and symbols with ASCII codes between 32~127 are supported, and the font size is 5x7
3. Offset [7:4]: x-axis offset, 0 ~ 7 [3:0]: y-axis offset, 0 ~ 7
4. Color uses RGB565 format 16-bit color: R(5-bit) + G(6-bit) + B(5-bit) bit[15:11] = Red, bit[10:5] = Green, bit[4:0] = Blue Color = (uint16_t)((Color_high << 8) | Color_low) Range 0~65535

5.8 Display scrolling string - Cmd 0x40

Request data

Input parameters: Index_id (device index ID), Scroll_mode (scrolling mode), Scroll_interval (scrolling interval), Color (character color), String_length (string length), char (character)

Request Frame:

AA 55 | 0x09+String_length | 0x00 | Index_id | 0x40 | Scroll_mode | Scroll_interval_low | Scroll_interval_

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x40 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status

1. **0**: Operation failed
 2. **1**: Operation successful
 3. **2**: Pattern mismatch (rolling character mode needs to be configured)
2. Scroll_mode [7:4]: Scroll direction 0: Scroll left, 1: Scroll right, 2: Scroll up, 3: Scroll down [3:0]: Scroll mode 0: Play once and stop, 1: Scroll in a loop, 2: Bounce back and forth
 3. Scroll_interval Scroll_interval movement speed unit is ms/pixel, each Scroll_interval time moves one pixel Scroll_interval =(uint16_t)((Scroll_interval_high << 8) | Scroll_interval_low) Range 0~65535, unit ms/pixel
 4. Color uses RGB565 format 16-bit color: R(5-bit) + G(6-bit) + B(5-bit) bit[15:11] = Red, bit[10:5] = Green, bit[4:0] = Blue Color = (uint16_t)((Color_high << 8) | Color_low) Range 0~65535 When Color is set to 0, the font color is the gradient color
 5. String_length Length of rolling string
 6. char The displayed characters support English letters, numbers and symbols with ASCII codes between 32~127, and the font size is 5x7

5.9 Get scrolling string - **Cmd 0x41**

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0x41 | CRC | 55 AA

Response Data

Return parameters: Scroll_mode (scrolling mode), Scroll_interval (scrolling interval), Color (character color), String_length (string length), char (character)

Response Frame:

AA 55 | 0x09+String_length | 0x00 | Index_id | 0x41 | Scroll_mode | Scroll_interval_low | Scroll_interval_

Things to note

1. Scroll_mode [7:4]: Scroll direction 0: Scroll left, 1: Scroll right, 2: Scroll up, 3: Scroll down [3:0]: Scroll mode 0: Play once and stop, 1: Scroll in a loop, 2: Bounce back and forth
2. Scroll_interval Scroll_interval movement speed unit is ms/pixel, each Scroll_interval time moves one pixel Scroll_interval =(uint16_t)((Scroll_interval_high << 8) | Scroll_interval_low) Range 0~65535, unit ms/pixel
3. Color uses RGB565 format 16-bit color: R(5-bit) + G(6-bit) + B(5-bit) bit[15:11] = Red, bit[10:5] = Green, bit[4:0] = Blue Color = (uint16_t)((Color_high << 8) | Color_low) Range 0~65535 When Color is set to 0, the font color is the gradient color
4. String_length Length of rolling string
5. Char The displayed characters support English letters, numbers and symbols with ASCII codes between 32~127, and the font size is 5x7

5.10 Set scrolling string status - **Cmd 0x42**

Request data

Input parameters: Index_id (device index ID), Scroll_state (scrolling state)

Request Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x42 | Scroll_state | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x42 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0: Operation failed
 2. 1: Operation successful
 3. 2: Pattern mismatch (rolling character mode needs to be configured)
2. Scroll_state
 1. 0: Start/continue scrolling
 2. 1: Pause scrolling (the screen keeps the current character displayed statically)
 3. 2: Stop and clear (clear the displayed characters and start from the beginning next time the setting is started)

5.11 Set scrolling string status - Cmd 0x43

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0x43 | CRC | 55 AA

Response Data

Return parameters: Scroll_state (scrolling state)

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0x43 | Scroll_state | CRC | 55 AA

Things to note

1. Scroll_state
 1. 0: scrolling
 2. 1: suspended state
 3. 2: idle/stop state

5.12 Set screen rotation angle - Cmd 0xE0

Request data

Input parameters: Index_id (device index ID), Screen_Rotation (screen rotation angle), Save_to_flash (whether to save to internal Flash)

Request Frame:

AA 55 | 0x05 | 0x00 | Index_id | 0xE0 | Screen_Rotation | Save_to_flash | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xE0 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. **0** : Operation failed
 2. **1** : Operation successful
2. Screen_Rotation
 1. **0** : Default display angle
 2. **1** : Rotate 90° clockwise
 3. **2** : Rotate 180° clockwise
 4. **3** : Rotate 270° clockwise (90° counterclockwise)
3. Save_to_flash Whether to save internal Flash
 1. **0** : Do not save
 2. **1** : Save
4. Successful setting takes effect immediately
5. Saving to the internal Flash requires erasing the page, which is a time-consuming process (about 20ms). During this period, the serial port interrupt will be turned off, and frequent operations will affect the life of the device.

5.13 Get the screen rotation angle - **Cmd 0xE1**

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xE1 | CRC | 55 AA

Response Data

Return parameters: Screen_Rotation (screen rotation angle)

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xE1 | Screen_Rotation | CRC | 55 AA

Things to note

1. Screen_Rotation
 1. **0** : Default display angle
 2. **1** : Rotate 90° clockwise
 3. **2** : Rotate 180° clockwise
 4. **3** : Rotate 270° clockwise (90° counterclockwise)

5.14 Set screen brightness - **Cmd 0xE2**

Request data

Input parameters: Index_id (device index ID), Brightness (screen brightness), Save_to_flash (whether to save to internal Flash)

Request Frame:

AA 55 | 0x05 | 0x00 | Index_id | 0xE2 | Brightness | Save_to_flash | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xE2 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0: Operation failed
 2. 1: Operation successful
2. Brightness default 50% Range 0 ~ 100% Driving at high brightness for a long time may cause damage. It is recommended to use a brightness lower than 60% to reduce heat generation and power consumption.
3. Save_to_flash Whether to save internal Flash
 1. 0: Do not save
 2. 1: Save
4. Successful setting takes effect immediately
5. Saving to the internal Flash requires erasing the page, which is a time-consuming process (about 20ms). During this period, the serial port interrupt will be turned off, and frequent operations will affect the life of the device.

5.15 Get screen brightness - Cmd 0xE3

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xE3 | CRC | 55 AA

Response Data

Return parameters: Brightness (screen brightness)

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xE3 | Brightness | CRC | 55 AA

Things to note

1. Brightness Range 0 ~ 100%

5.16 Clear screen - Cmd 0xE4

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xE4 | CRC | 55 AA

Response Data

Return parameters: Operation_status

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xE4 | Operation_status | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0 : Operation failed
 2. 1 : Operation successful

5.17 Query the unique UID of the device - Cmd 0xF8

Request data

Input parameters: Index_id (device index ID), UID_Type

Request Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xF8 | UID_Type | CRC | 55 AA

Response Data

Return parameters: Operation_status, UID

Response Frame:

AA 55 | 0x04+4/12 | 0x00 | Index_id | 0xF8 | Operation_status | UID (multi-byte) | CRC | 55 AA

Things to note

1. Operation_status operation status
 1. 0 : Operation failed
 2. 1 : Operation successful
2. UID_Type UID type
 1. 0 : 4byte UID
 2. 1 : 12byte UID

5.18 Query the upgrade program version number - Cmd 0xF9

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xF9 | CRC | 55 AA

Response Data

Return parameters: Bootloader_version

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xF9 | Bootloader_version | CRC | 55 AA

Things to note

None.

5.19 Query the device software version number - Cmd 0xFA

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xFA | CRC | 55 AA

Response Data

Return parameters: Firmware_version

Response Frame:

AA 55 | 0x04 | 0x00 | Index_id | 0xFA | Firmware_version | CRC | 55 AA

Things to note

None.

5.20 Query device type - Cmd 0xFB

Request data

Input parameter: Index_id (device index ID)

Request Frame:

AA 55 | 0x03 | 0x00 | Index_id | 0xFB | CRC | 55 AA

Response Data

Return parameters: Device_type

Response Frame:

AA 55 | 0x05 | 0x00 | Index_id | 0xFB | Device_type_low | Device_type_high | CRC | 55 AA

Things to note

1. Device_type = (uint16_t)((Device_type_high << 8) | Device_type_low)
2. The device type code for RGB is 0x000E

5.21 Enumeration request, sent by the end device of chain link change, and sent by the device when it is powered on, notifying the host to update the link device status - Cmd 0xFC

Request data

Input parameters: none

Request Frame:

Device-initiated report, No host request frame

Response Data

Return parameters: none

Response Frame:

Device-initiated report, No response frame

Things to note

None.

5.22 Heartbeat packet, regular communication between chain devices, you can self-discover whether it is an end device, and the host can also use the heartbeat packet to determine whether there is a chain device connection - `Cmd 0xFD`

Request data

Input parameters: none

Request Frame:

AA 55 | 0x03 | 0x00 | 0xFF | 0xFD | CRC | 55 AA

Response Data

Return parameters: none

Response Frame:

AA 55 | 0x03 | 0x00 | 0xFF | 0xFD | CRC | 55 AA

Things to note

None.

5.23 Enumerate to obtain the number of cascaded devices - `Cmd 0xFE`

Request data

Input parameter: Send_num (default 0, used to record the number of devices)

Request Frame:

AA 55 | 0x04 | 0x00 | 0xFF | 0xFE | Send_num | CRC | 55 AA

Response Data

Return parameter: Receive_num (the value represents the number of devices)

Response Frame:

AA 55 | 0x04 | 0x00 | 0xFF | 0xFE | Receive_num | CRC | 55 AA

Things to note

None.

6. Status code/enumeration value

6.1 Operating status

value	Name	Description
0	Failure	Operation failed
1	Success	Operation successful

Other modes, ranges, and event enumeration values are defined in the "Notes" of the corresponding instructions.

7. Version records

Document version	Date	Description
V1.0	2026-6-1	Organized according to the official Chain RGB protocol