M5Stack Chain RGB Protocol		All packets start with 0xAA 0x55 and end with 0x55 0xAA												V1.0 (Version) 2026/6/1
指令集	Byte / 指令格式	0	1	2	3	4	5	6	7	8	9	10	11	12
		Length_low	Length_high	Index	Cmd	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9
设置显示模式	指令包格式	0x04	0x00	Index_id	0x10	Mode	CRC							
	应答包格式	0x04	0x00	Index_id	0x10	Operation_status	CRC							
	指令详解	（1）功能说明：设置显示模式 （2）输入参数：Index_id（设备下标ID）、Mode（显示模式） （3）返回参数：Operation_status （4）指令代码：0x10 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Mode 显示模式（默认0） 0：像素点控制模式（直接设置像素显示） 1：滚动字符模式 （滚动显示字符串）												
获取显示模式	指令包格式	0x03	0x00	Index_id	0x11	CRC								
	应答包格式	0x04	0x00	Index_id	0x11	Mode	CRC							
	指令详解	（1）功能说明：获取显示模式 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Mode（显示模式） （4）指令代码：0x11 注1：Mode 显示模式（默认0） 0：像素点控制模式（直接设置像素显示） 1：滚动字符模式 （滚动显示字符串）												
设置像素	指令包格式	0x04 + Num*3	0x00	Index_id	0x30	Num	Coordinate_xy0	Pixel_color_low0	Pixel_color_high0	Coordinate_xy1	Pixel_color_low1	Pixel_color_low1	...	CRC
	应答包格式	0x04	0x00	Index_id	0x30	Operation_status	CRC							
	指令详解	（1）功能说明：设置像素。 （2）输入参数：Index_id（设备下标ID）、Num（要设置的像素数量）、Coordinate_xy（像素XY坐标）、Pixel_color（像素颜色） （3）返回参数：Operation_status （4）指令代码：0x30 注1：Operation_status 操作状态 0：操作失败 1：操作成功 2：模式不匹配（需要配置为像素点控制模式） 注2：Num范围：1～64 注3：Coordinate_xy bit[7:6]：保留 bit[5:3]：X坐标 (0-7) bit[2:0]：Y坐标 (0-7) 注4：Pixel_color 采用 RGB565 格式 16位色彩：R(5位) + G(6位) + B(5位) bit[15:11] = Red bit[10:5] = Green bit[4:0] = Blue Pixel_color = (uint16_t)((Pixel_color_high << 8) | Pixel_color_low) 范围 0～65535												
全屏显示缓存刷新	指令包格式	0x83	0x00	Index_id	0x31	Display_buffer_low0	Display_buffer_high0	Display_buffer_low1	Display_buffer_high1	Display_buffer_low2	Display_buffer_high2	...	Display_buffer_high63	CRC
	应答包格式	0x04	0x00	Index_id	0x31	Operation_status	CRC							
	指令详解	（1）功能说明：全屏显示缓存刷新 （2）输入参数：Index_id（设备下标ID）、Display_buffer(全屏显示缓存) （3）返回参数：Operation_status （4）指令代码：0x31 注1：Operation_status 操作状态 0：操作失败 1：操作成功 2：模式不匹配（需要配置为像素点控制模式） 注2：Display_buffer 采用 RGB565 格式 数据排列：屏幕像素颜色从左往右，从上往下（行优先存储），先存储第0行的8个像素，再存储第1行，依此类推 Display_buffer0 = （uint16_t)((Display_buffer_high0 << 8) | Display_buffer_low0 ）范围 0～65535												
获取像素值	指令包格式	0x04 + Num	0x00	Index_id	0x32	Num	Coordinate_xy0	Coordinate_xy1	Coordinate_xy2	Coordinate_xy3	Coordinate_xy4	Coordinate_xy5	...	CRC
	应答包格式	0x03 + Num*2	0x00	Index_id	0x32	Pixel_color_low0	Pixel_color_high0	Pixel_color_low1	Pixel_color_high1	Pixel_color_low2	Pixel_color_high2	Pixel_color_low3	...	CRC
	指令详解	（1）功能说明：获取像素值。 （2）输入参数：Index_id（设备下标ID）、Num（要获取的像素数量）、Coordinate_xy（像素坐标） （3）返回参数：Pixel_color（像素点颜色） （4）指令代码：0x32 注1：Coordinate_xy bit[7:6]：保留 bit[5:3]：X坐标 (0-7) bit[2:0]：Y坐标 (0-7) 注2：Pixel_color 采用 RGB565 格式 16位色彩：R(5位) + G(6位) + B(5位) bit[15:11] = Red， bit[10:5] = Green， bit[4:0] = Blue Pixel_color = (uint16_t)((Pixel_color_high << 8) | Pixel_color_low ）范围 0～65535												
获取全屏显示缓存	指令包格式	0x03	0x00	Index_id	0x33	CRC								
	应答包格式	0x83	0x00	Index_id	0x33	Display_buffer_low0	Display_buffer_high0	Display_buffer_low1	Display_buffer_high1	Display_buffer_low2	Display_buffer_high2	...	Display_buffer_high63	CRC
	指令详解	（1）获取整个屏幕刷新状态：获取全屏显示缓存 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Display_buffer(全屏显示缓存) （4）指令代码：0x33 注1：Display_buffer 采用 RGB565 格式 数据排列：屏幕像素颜色从左往右，从上往下（行优先存储），先存储第0行的8个像素，再存储第1行，依此类推 Display_buffer0 = （uint16_t)((Display_buffer_high0 << 8) | Display_buffer_low0 ）范围 0～65535												
显示字符	指令包格式	0x07	0x00	Index_id	0x34	Char	Offset	Color_low	Color_high	CRC				
	应答包格式	0x04	0x00	Index_id	0x34	Operation_status	CRC							
	指令详解	（1）功能说明：显示字符 （2）输入参数：Index_id（设备下标ID）、Char（要显示字符的ASCII码）、Offset（显示的位置偏移）、Color（字符颜色） （3）返回参数：Operation_status （4）指令代码：0x34 注1：Operation_status 操作状态 0：操作失败 1：操作成功 2：模式不匹配（需要配置为像素点控制模式） 注2：Char 要显示字符的ASCII码，支持ASCIIC在32~127之间的英文字母，数字和符号，字体尺寸为5x7 注3：Offset [7:4]：x轴偏移，0~7 [3:0]：y轴偏移，0~7 注4：Color 采用 RGB565 格式 16位色彩：R(5位) + G(6位) + B(5位) bit[15:11] = Red， bit[10:5] = Green， bit[4:0] = Blue Color = (uint16_t)((Color_high << 8) | Color_low ）范围 0～65535												

显示滚动字符串

指令包格式	0x09+String_length	0x00	Index_id	0x40	Scroll_mode	Scroll_interval_low	Scroll_interval_high	Color_low	Color_high	String_length	char1	...	CRC
应答包格式	0x04	0x00	Index_id	0x40	Operation_status	CRC							

指令详解

（1）功能说明：显示滚动字符串
（2）输入参数：Index_id（设备下标ID）、Scroll_mode（滚动模式）、Scroll_interval（滚动间隔）、Color（字符颜色）、String_length(字符串长度)、char(字符)
（3）返回参数：Operation_status
（4）指令代码：0x40
注1：Operation_status 操作状态
0：操作失败
1：操作成功
2：模式不匹配（需要配置滚动字符模式）
注2：Scroll_mode
[7:4]：滚动方向 0：往左滚动，1：往右滚动，2：往上滚动，3：往下滚动
[3:0]：滚动模式 0：播放一次停住，1：循环滚动，2： 来回反弹
注3：Scroll_interval
Scroll_interval移动速度单位为ms/pixel，每个Scroll_interval时间移动一个像素
Scroll_interval =(uint16_t)((Scroll_interval_high < < 8) | Scroll_interval_low) 范围 0～65535，单位ms/pixel
注4：Color 采用 RGB565 格式
16位色彩：R(5位) + G(6位) + B(5位)
bit[15:11] = Red， bit[10:5] = Green， bit[4:0] = Blue
Color = (uint16_t)((Color_high << 8) | Color_low) 范围 0～65535
当Color设置为0时，字体的颜色为渐变的彩色
注5：String_length
滚动字符串的长度
注6：char
显示的字符，支持ASCIIC在32~127之间的英文字母，数字和符号，字体尺寸为5x7

获取滚动字符串

指令包格式	0x03	0x00	Index_id	0x41	CRC								
应答包格式	0x09+String_length	0x00	Index_id	0x41	Scroll_mode	Scroll_interval_low	Scroll_interval_high	Color_low	Color_high	String_length	Char1	...	CRC

指令详解

（1）功能说明：获取滚动字符串。
（2）输入参数：Index_id（设备下标ID）
（3）返回参数：Scroll_mode（滚动模式）、Scroll_interval（滚动间隔）、Color（字符颜色）、String_length(字符串长度)、char(字符)
（4）指令代码：0x41
注1：Scroll_mode
[7:4]：滚动方向 0：往左滚动，1：往右滚动，2：往上滚动，3：往下滚动
[3:0]：滚动模式 0：播放一次停住，1：循环滚动，2： 来回反弹
注2：Scroll_interval
Scroll_interval移动速度单位为ms/pixel，每个Scroll_interval时间移动一个像素
Scroll_interval =(uint16_t)((Scroll_interval_high < < 8) | Scroll_interval_low) 范围 0～65535，单位ms/pixel
注3：Color 采用 RGB565 格式
16位色彩：R(5位) + G(6位) + B(5位)
bit[15:11] = Red， bit[10:5] = Green， bit[4:0] = Blue
Color = (uint16_t)((Color_high << 8) | Color_low) 范围 0～65535
当Color设置为0时，字体的颜色为渐变的彩色
注4：String_length
滚动字符串的长度
注5：Char
显示的字符，支持ASCIIC在32~127之间的英文字母，数字和符号，字体尺寸为5x7

设置滚动字符串状态

指令包格式	0x04	0x00	Index_id	0x42	Scroll_state	CRC							
应答包格式	0x04	0x00	Index_id	0x42	Operation_status	CRC							

指令详解

（1）功能说明：设置滚动字符串状态。
（2）输入参数：Index_id（设备下标ID），Scroll_state（滚动状态）
（3）返回参数：Operation_status
（4）指令代码：0x42
注1：Operation_status 操作状态
0：操作失败
1：操作成功
2：模式不匹配（需要配置滚动字符模式）
注2：Scroll_state
0：开始/继续滚动
1：暂停滚动（画面保持当前字符静止显示）
2：停止并清除（清除显示的字符，下次设置开始将从头开始）

获取滚动字符串状态

指令包格式	0x03	0x00	Index_id	0x43	CRC								
应答包格式	0x04	0x00	Index_id	0x43	Scroll_state	CRC							

指令详解

（1）功能说明：设置滚动字符串状态。
（2）输入参数：Index_id（设备下标ID）
（3）返回参数：Scroll_state（滚动状态）
（4）指令代码：0x43
注1：Scroll_state
0：正在滚动
1：暂停状态
2：空闲/停止状态

设置屏幕旋转角度

指令包格式	0x05	0x00	Index_id	0xE0	Screen_Rotation	Save_to_flash	CRC						
应答包格式	0x04	0x00	Index_id	0xE0	Operation_status	CRC							

指令详解

（1）功能说明：设置屏幕旋转角度。
（2）输入参数：Index_id（设备下标ID）、Screen_Rotation(屏幕旋转角度)、Save_to_flash（是否保存到内部Flash）
（3）返回参数：Operation_status
（4）指令代码：0xE0
注1：Operation_status 操作状态
0：操作失败
1：操作成功
注2：Screen_Rotation
0：默认显示角度
1：顺时针旋转90°
2：顺时针旋转180°
3：顺时针旋转270°（逆时针90°）
注3：Save_to_flash 是否保存内部Flash
0：不保存
1：保存
注4：设置成功即刻生效
注5：保存至内部 Flash 需要擦除页面，过程较为耗时（约20ms），期间会关闭串口中断，且频繁操作会影响设备寿命。

获取屏幕旋转角度

指令包格式	0x03	0x00	Index_id	0xE1	CRC								
应答包格式	0x04	0x00	Index_id	0xE1	Screen_Rotation	CRC							

指令详解

（1）功能说明：获取屏幕旋转角度。
（2）输入参数：Index_id（设备下标ID）
（3）返回参数：Screen_Rotation(屏幕旋转角度)
（4）指令代码：0xE1
注1：Screen_Rotation
0：默认显示角度
1：顺时针旋转90°
2：顺时针旋转180°
3：顺时针旋转270°（逆时针90°）

设置屏幕亮度	指令包格式	0x05	0x00	Index_id	0xE2	Brightness	Save_to_flash	CRC						
	应答包格式	0x04	0x00	Index_id	0xE2	Operation_status	CRC							
	指令详解	（1）功能说明：设置屏幕亮度。 （2）输入参数：Index_id（设备下标ID）、Brightness(屏幕亮度)、Save_to_flash（是否保存到内部Flash） （3）返回参数：Operation_status （4）指令代码：0xE2 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Brightness 默认50% 范围 0 ~ 100% 长时间高亮度驱动可能导致损坏，推荐使用的低于 60% 的亮度，减少发热和功耗。 注3：Save_to_flash 是否保存内部Flash 0：不保存 1：保存 注4：设置成功即刻生效 注5：保存至内部 Flash 需要擦除页面，过程较为耗时（约20ms），期间会关闭串口中断，且频繁操作会影响设备寿命。												
获取屏幕亮度	指令包格式	0x03	0x00	Index_id	0xE3	CRC								
	应答包格式	0x04	0x00	Index_id	0xE3	Brightness	CRC							
	指令详解	（1）功能说明：获取屏幕亮度。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Brightness(屏幕亮度) （4）指令代码：0xE3 注1：Brightness 范围 0 ~ 100%													
清屏	指令包格式	0x03	0x00	Index_id	0xE4	CRC								
	应答包格式	0x04	0x00	Index_id	0xE4	Operation_status	CRC							
	指令详解	（1）功能说明：清屏。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Operation_status （4）指令代码：0xE4 注1：Operation_status 操作状态 0：操作失败 1：操作成功													
查询设备唯一UID	指令包格式	0x04	0x00	Index_id	0xF8	UID_Type	CRC							
	应答包格式	0x04+4/12	0x00	Index_id	0xF8	Operation_status	UID（多字节）	CRC						
	指令详解	（1）功能说明：查询设备唯一UID。 （2）输入参数：Index_id（设备下标ID）、UID_Type （3）返回参数：Operation_status、UID （4）指令代码：0xF8 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：UID_Type UID的类型 0：4byte UID 1：12byte UID													
查询升级程序版本号	指令包格式	0x03	0x00	Index_id	0xF9	CRC								
	应答包格式	0x04	0x00	Index_id	0xF9	Bootloader_version	CRC							
	指令详解	（1）功能说明：查询升级程序版本号。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Bootloader_version （4）指令代码：0xF9													
查询设备软件版本号	指令包格式	0x03	0x00	Index_id	0xFA	CRC								
	应答包格式	0x04	0x00	Index_id	0xFA	Firmware_version	CRC							
	指令详解	（1）功能说明：查询设备软件版本号。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Firmware_version （4）指令代码：0xFA													
查询设备类型	指令包格式	0x03	0x00	Index_id	0xFB	CRC								
	应答包格式	0x05	0x00	Index_id	0xFB	Device_type_low	Device_type_high	CRC						
	指令详解	（1）功能说明：查询设备类型。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Device_type （4）指令代码：0xFB 注1：Device_type = (uint16_t)((Device_type_high << 8) | Device_type_low) 注2：RGB的设备类型码是0x000E													
枚举请求	指令包格式	None												
	应答包格式	0x03	0x00	0xFF	0xFC	CRC								
	指令详解	（1）功能说明：枚举请求，chain链路变更末端设备发送、以及设备上电发送，通知主机更新链路设备状态。 （2）输入参数：none （3）返回参数：none （4）指令代码：0xFC													
心跳包	指令包格式	0x03	0x00	0xFF	0xFD	CRC								
	应答包格式	0x03	0x00	0xFF	0xFD	CRC								
	指令详解	（1）功能说明：心跳包，chain设备之间定时通信，可以自发现自己是不是末端设备，主机也可以通过心跳包来判断是否有chain设备连接。 （2）输入参数：none （3）返回参数：none （4）指令代码：0xFD													
枚举	指令包格式	0x04	0x00	0xFF	0xFE	Send_num	CRC							
	应答包格式	0x04	0x00	0xFF	0xFE	Receive_num	CRC							
	指令详解	（1）功能说明：枚举获取联级设备的个数。 （2）输入参数：Send_num（默认0，用于记录设备个数） （3）返回参数：Receive_num（数值代表设备个数） （4）指令代码：0xFE													

注1：数据包最大长度是 256 byte，串口
注2：数据长度是Index_id到CRC，包括Index_id以及CRC不包括数据长度本身
注3：CRC 计算时，需要排除包头、包尾、长度和 CRC 字段本身，仅对其余数据求和
注4：串口通讯波特率115200，8位数据位，1停止位，无校验位

```
uint8_t calculateCRC(const uint8_t *buffer, uint16_t size)
{
    uint8_t crc8 = 0;
    for (uint8_t i = 4; i < (size - 3); i++) {
        crc8 += buffer[i];
    }
    return crc8;
}
```