M5Stack Chain Angle Protocol								All packets start with 0xAA 0x55 and end with 0x55 0xAA				V1 (Version) 2025/10/23
指令集	Byte 指令格式	0	1	2	3	4	5	6	7	8	9	10
		Length_low	Length_high	Index	Cmd	Data1	Data2	Data3	Data4	Data5	Data6	Data7
设置RGB值	指令包格式	0x05+Num*3	0x00	Index_id	0x20	Index	Num	R	G	B	...	CRC
	应答包格式	0x04	0x00	Index_id	0x20	Operation_status	CRC					
	指令详解	（1）功能说明：设置RGB值。 （2）输入参数：Index_id（设备下标ID）、Index（RGB灯起始坐标）、Num（要设置的RGB数量）、RGB值 （3）返回参数：Operation_status （4）指令代码：0x20 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Index是RGB灯起始坐标，从0开始，Angle设备只有一个RGB灯，所以Index = 0，Num = 1										
获取RGB值	指令包格式	0x05	0x00	Index_id	0x21	Index	Num	CRC				
	应答包格式	0x04+Num*3/0x04	0x00	Index_id	0x21	Operation_status	R	G	B	...	CRC	
	指令详解	（1）功能说明：获取RGB值。 （2）输入参数：Index_id（设备下标ID）、Index（RGB灯起始坐标）、Num（要读取的RGB灯数量） （3）返回参数：RGB值 （4）指令代码：0x21 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Index是RGB灯起始坐标，从0开始，Angle设备只有一个RGB灯，所以Index = 0，Num = 1 注3：操作失败返回数据包不包含RGB值										
设置RGB灯亮度	指令包格式	0x05	0x00	Index_id	0x22	Light	Save_to_flash	CRC				
	应答包格式	0x04	0x00	Index_id	0x22	Operation_status	CRC					
	指令详解	（1）功能说明：设置RGB灯亮度 （2）输入参数：Index_id（设备下标ID）、Light（RGB亮度，0~100，默认40）、Save_to_flash（是否保存到内部Flash） （3）返回参数：Operation_status （4）指令代码：0x22 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Save_to_flash 是否保存内部Flash 0：不保存 1：保存 注3：设置成功即刻生效 注4：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，如需要频繁调整亮度，请避免每次都保存至内部 Flash。										
获取RGB灯亮度	指令包格式	0x03	0x00	Index_id	0x23	CRC						
	应答包格式	0x04	0x00	Index_id	0x23	Light	CRC					
	指令详解	（1）功能说明：获取RGB灯亮度。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Light（RGB亮度） （4）指令代码：0x23										
获取12bitADC	指令包格式	0x03	0x00	Index_id	0x30	CRC						
	应答包格式	0x05	0x00	Index_id	0x30	ADC_low	ADC_high	CRC				
	指令详解	（1）功能说明：读取Angle的12bit的ADC。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：ADC值 （4）指令代码：0x30 注1：Angle_ADC = (uint16_t)((ADC_high << 8) | ADC_low) 注2：理论取值范围为 0 ~ 4095，但受器件差异影响，上下限各存在 0 ~ 10 的误差偏移，因此实际可用范围会相应缩减。										
获取8bitADC	指令包格式	0x03	0x00	Index_id	0x31	CRC						
	应答包格式	0x04	0x00	Index_id	0x31	ADC	CRC					
	指令详解	（1）功能说明：读取Angle的8bit的ADC。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：ADC值 （4）指令代码：0x31 注1：Angle_ADC = ADC 注2：理论取值范围为 0 ~ 255，但受器件差异影响，上下限各存在 0 ~ 3 的误差偏移，因此实际可用范围会相应缩减。										
设置旋转控制状态	指令包格式	0x05	0x00	Index_id	0x32	Clockwise_direct	Save_to_flash	CRC				
	应答包格式	0x04	0x00	Index_id	0x32	Operation_status	CRC					
	指令详解	（1）功能说明：设置旋转控制状态，决定顺时针旋转编码器值是增加还是减小。 （2）输入参数：Index_id（设备下标ID）、Clockwise_direct（顺时针状态方向）、Save_to_flash（是否保存配置到内部Flash） （3）返回参数：Operation_status （4）指令代码：0x32 注1：Clockwise_direct 编码器方向 0：顺时针减小 1：顺时针增加（默认状态） 注2：Operation_status 操作状态 0：操作失败 1：操作成功 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。										
获取旋转控制状态	指令包格式	0x03	0x00	Index_id	0x33	CRC						
	应答包格式	0x04	0x00	Index_id	0x33	Clockwise_direct	CRC					
	指令详解	（1）功能说明：获取旋转控制状态。 （2）输入参数：Index_id（设备下标ID） （3）返回参数：Clockwise_directr值 （4）指令代码：0x33 注1：Clockwise_direct 编码器方向 0：顺时针减小 1：顺时针增加（默认状态）										

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：Angle的设备类型码是0x0002										
枚举请求	指令包格式	None										
	应答包格式	0x03	0x00	Index_id	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：用户可以通过枚举获取链路设备个数。 （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;
}
```