

M5Stack Chain ToF Protocol								All packets start with 0xAA 0x55 and end with 0x55 0xAA				V1 (Version)
2025/10/22												
指令集	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, TOF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, TOF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ToF测量距离	指令包格式	0x03	0x00	Index_id	0x50	CRC						
	应答包格式	0x04	0x00	Index_id	0x50	Measure_distance_low	Measure_distance_high	CRC				
	指令详解	(1) 功能说明: 获取ToF测量距离。 (2) 输入参数: Index_id (设备下标ID) (3) 返回参数: Measure_distance (4) 指令代码: 0x50 注1: Measure_distance = (uint16_t)(Measure_distance_high << 8 Measure_distance_low (单位: mm)										
设置测量时间	指令包格式	0x04	0x00	Index_id	0x51	Time	CRC					
	应答包格式	0x04	0x00	Index_id	0x51	Operation_status	CRC					
	指令详解	(1) 功能说明: 设置测量时间。 (2) 输入参数: Index_id (设备下标ID)、Time (3) 返回参数: Operation_status (4) 指令代码: 0x51 注1: Operation_status 操作状态 0: 操作失败 1: 操作成功 注2: Time 测量时间 (测量时间越长, 精度越高) Value: 20 ~ 200ms Default: 33ms 注3: 该指令需要更改VLS3LOX的配置, 比较耗时。										
获取测量时间	指令包格式	0x03	0x00	Index_id	0x52	CRC						
	应答包格式	0x04	0x00	Index_id	0x52	Time	CRC					
	指令详解	(1) 功能说明: 获取测量模式。 (2) 输入参数: Index_id (设备下标ID) (3) 返回参数: Mode (4) 指令代码: 0x52 注1: Time 测量时间 (测量时间越长, 精度越高) Value: 20 ~ 200ms Default: 33ms										
设置测量模式	指令包格式	0x04	0x00	Index_id	0x53	Mode	CRC					
	应答包格式	0x04	0x00	Index_id	0x53	Operation_status	CRC					
	指令详解	(1) 功能说明: 设置测量模式。 (2) 输入参数: Index_id (设备下标ID)、Mode (3) 返回参数: Operation_status (4) 指令代码: 0x53 注1: Operation_status 操作状态 0: 操作失败 1: 操作成功 注2: Mode 测量模式 0: 停止模式 (停止测量, 功耗最低) 1: 单次模式 (默认不主动采样, 需要手动触发, 功耗与触发频率成正比) 2: 连续模式 (默认主动采样, 自动循环采样, 功耗最大) Default: 2										

获取测量模式	指令包格式	0x03	0x00	Index_id	0x54	CRC													
	应答包格式	0x04	0x00	Index_id	0x54	Mode	CRC												
	指令详解	<p>(1) 功能说明: 获取测量模式。 (2) 输入参数: Index_id (设备下标ID) (3) 返回参数: Mode (4) 指令代码: 0x54 注1: Mode 测量模式 0: 停止模式 (停止测量, 功耗最低) 1: 单次模式 (默认不主动采样, 需要手动触发, 功耗与触发频率成正比) 2: 连续模式 (默认主动采样, 自动循环采样, 功耗最大) Default: 2</p>																	
设置测量状态	指令包格式	0x04	0x00	Index_id	0x55	Status	CRC												
	应答包格式	0x04	0x00	Index_id	0x55	Operation_status	CRC												
	指令详解	<p>(1) 功能说明: 设置测量状态。 (2) 输入参数: Index_id (设备下标ID) 、 Status (测量状态) (3) 返回参数: Operation_status (4) 指令代码: 0x55 注1: Operation_status 操作状态 0: 操作失败 1: 操作成功 注2: Status 测量状态 0: 不测量 (停止模式恒0) 1: 测量 (连续模式恒1) 注3: 停止模式改状态是不允许为1, 连续模式改状态是不允许为0, 非法设置会返回失败 注4: 单次模式下测量结束, 其他模式转换为单次模式, 标志位会主动置0 注5: 其他模式转为停止模式, 标志位会主动置0 注6: 其他模式转为连续模式, 标志位会主动置1</p>																	
获取测量状态	指令包格式	0x03	0x00	Index_id	0x56	CRC													
	应答包格式	0x04	0x00	Index_id	0x56	Status	CRC												
	指令详解	<p>(1) 功能说明: 获取测量状态。 (2) 输入参数: Index_id (设备下标ID) (3) 返回参数: Status (测量状态) (4) 指令代码: 0x56 注1: Status 测量模式 0: 不测量 (停止模式恒0) 1: 测量 (连续模式恒1)</p>																	
获取测量完成标志	指令包格式	0x03	0x00	Index_id	0x57	CRC													
	应答包格式	0x04	0x00	Index_id	0x57	Flag	CRC												
	指令详解	<p>(1) 功能说明: 获取测量完成标志。 (2) 输入参数: Index_id (设备下标ID) (3) 返回参数: Flag (完成标志) (4) 指令代码: 0x57 注1: Flag 测量标志 0: 测量未完成 1: 测量完成 注2: 读取测量结果 (指令代码: 0x50) , 该标志位会被置0, 采集完成一次该标志位置1。</p>																	
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	<p>(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</p>																	
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	<p>(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</p>																	
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	<p>(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</p>																	
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	<p>(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: ToF的设备类型码是0x0005</p>																	
枚举请求	指令包格式	None																	
	应答包格式	0x03	0x00	0xFF	0xFC	CRC													
	指令详解	<p>(1) 功能说明: 枚举请求, chain链路变更未读设备发送, 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</p>																	

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: 枚举获取下级设备的个数。 (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										
<p>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</p> <pre> 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; } </pre>												