		M5Stack	Chain-l	Jart Prot	tocol			All packets	start with 0xAA 0	x55 and end with	0x55 0xAA	V1 (Version) 2024/10/30		
指令集	Byte	0	1	2	3	4	5	6	7	8	9	10		
担女米	指令格式	Length_low	Length_high	Index	Cmd	Data1	Data2	Data3	Data4	Data5	Data6	Data7		
	指令包格式	0x04	0x00	Index_id	0x10	I2c_speed	CRC							
	应答包格式	0x04	0x00	Index_id	0x10	Operation_status	CRC							
12C初始化	指令详解	(1) 功能说明: I2C初始化。 (2) 输入参数: Index_id(设备下标ID)、I2C_speed(I2C速度) (3) 返回参数: Operation status (4) 指令代码: 0x10 注:: Operation status 操作状态												
	指令包格式	0x05	0x00	Index_id	0x11	I2c_addr	length	CRC						
	应答包格式	(0x04+length)/(0x04)	0x00	Index_id	0x11	Operation_status	Data1	Data2	Data3		CRC			
12C读	指令详解	(1) 功能说明: I2C读。 (2) 输入参数: Index_id(设备下标D)、I2c_addr(I2C地址)、length(要读取的长度) (3) 返回参数: Operation_status(操作状态),Data(读取的数据) (4) 指令代码: Ox11 注:Operation_status 操作状态 ① 操作失数 1: 操作失数 1: 操作成功 2: 模式不配配(需要失配置I2C) 注2: 操作成功才会返回要读取的数据												
	指令包格式	0x08+length	0x00	Index_id	0x12	I2c_addr	length	Data1	Data2	Data3		CRC		
	应答包格式	0x04	0x00	Index_id	0x12	Operation_status	CRC							
12C写	指令详解	(1) 功能说明: I2C写。 (2) 输入参数: Index_id(设备下标D)、I2c_addr(I2C地址),length(要写入的数据长度),Data要写入的数据 (3) 返回参数: Operation_status (操作状态) (4) 指令代码: Ox12 注: Operation_status 提作状态 0: 操作失败 1: 操作成功 2: 模式不配配(需要先配置I2C)												
	指令包格式	0x08	0x00	Index_id	0x13	I2c_addr	Reg_addr_length	Reg_addr_low	Reg_addr_high	length	CRC			
	应答包格式	0x04+length	0x00	Index_id	0x13	Operation_status	Data1	Data2	Data3		CRC			
12C指定地址读	指令详解	(3) 返回参数: Oper (4) 指令代码: Ox13 注1: Operation_stati 0: 操作失败 1: 操作成功 2: 模式不匹配 注2: Reg_addr_lengt 1: 8位地址 2: 16位地址	:_id (设备下标ID ation_status (操 us 操作状态 (需要先配置I2C) th要控制的传感器	作状态),Data(加速 设备地址大小	读取的数据)	r_length(I2C设备寄存 中国的, CC设备寄存器地址是8位。		(寄存器地址),leng	th(要读取的长度)					

	指令包格式	0x08+length	0x00	Index_id	0x14	I2c_addr	Reg_addr_length	Reg_addr_low	Reg_addr_high	length	Data	CRC		
12C指定地址写	应答包格式	0x04	0x00	Index_id	0x14	Operation_status	CRC							
	指令详解	(1) 功能说明: I2C指定地址写。 (2) 输入参数: Index_id(设备下标D)、I2c_addr(I2C地址)、Reg_addr_length(I2C设备寄存器长度)、Reg_addr(寄存器地址),length(要写入数据长度)、Data(要写入的数据) (3) 返回参数: Operation status (操作状态) (4) 指令代用: Ox14 注1: Operation status 操作状态 0: 操作失数 1: 操作成功 2: 模式不匹配(需要先配置2C) 注2: Reg_addr_length要控制的传感器设备地址大小 1: 8位地址 2: 16位地址 2: 16位地址 2: 16位地址 2: 16位地址 2: 16位地址 2: 18式不见品(需要,Addr_low(如果i2c设备寄存器地址是8位,让高八位款认0即可)												
	指令包格式	0x03	0x00	Index_id	0x15	CRC								
	应答包格式	0x05+I2c_addr_num	0x00	Index_id	0x15	Operation_status	I2c_addr_num	Addr1	Addr2	Addr3		CRC		
获取连接设备的I2C地址	指令详解	(1) 功能说明: 获取连接设备的I2C地址。 (2) 输入参数: Index_id (设备下标D) (3) 返回参数: I2c_addr_num (I2c地址个数) 、Addr(I2c地址) (4) 指令代码: 0x15 注1: Operation status 操作状态 0: 操作失数 1: 操作成功 2: 模式不匹配(需要先配置I2C)												
	指令包格式	0x06	0x00	Index_id	0x20	R	G	В	CRC					
	应答包格式	0x04	0x00	Index_id	0x20	Operation_status	CRC							
设置RGB值	指令详解	(1) 功能说明: 设置RGB值. (2) 输入参数: Index, id (设备下标D)、RGB值 (3) 返回参数: Operation_status (4) 指令代码: 0x20 注2: Operation status 操作状态 0: 操作失败 1: 操作成功												
	指令包格式	0x03	0x00	Index_id	0x21	CRC								
获取RGB值	应答包格式	0x06	0x00	Index_id	0x21	R	G	В	CRC					
	指令详解	(1) 功能说明: 获取RG (2) 输入参数: Index_ (3) 返回参数: RGB值 (4) 指令代码: 0x21	id (设备下标ID)											
	指令包格式	0x04	0x00	Index_id	0x22	Light	CRC							
	应答包格式	0x04	0x00	Index_id	0x22	Operation_status	CRC							
设置RGB灯亮度	指令详解	(1) 功能说明: 设置R(2) 输入参数: Index (3) 返回参数: Opera(4) 指令代码: 0x22 注1: Operation_statt 0: 操作失政 1: 操作成功 注2: 设置成功即刻生3	_id(设备下标ID) ition_status is 操作状态		度)									
	指令包格式	0x03	0x00	Index_id	0x23	CRC								
获取RGB灯亮度	应答包格式	0x04	0x00	Index_id	0x23	Light	CRC							
	指令详解	(1) 功能说明: 获取RC (2) 输入参数: Index (3) 返回参数: Light (4) 指令代码: 0x23	id (设备下标ID))										

	指令包格式	0x08	0x00	Index_id	0x30	GPIO	GPIO_level	GPIO_mode	GPIO_up/down	GPIO_speed	CRC				
	应答包格式	0x04	0x00	Index_id	0x30	Operation_status	CRC								
GPIO輸出模式配置	指令详解	(2) 輸入参数: Index (3) 返回参数: Open (4) 指令代码: Ox30 注1: Operation stat O: 操作失政 1: 操作成功 注2: GPIO 1: 表示GPIO1 2: 表示GPIO2 注3: GPIO_level O: 表示而电平 注4: GPIO_mode O: 表示推挽 1: 表示开调 注5: GPIO_up/down O: 表示下拉	注1: Operation status 操作状态 0: 操件失数 1: 操作成功 1: 操作成功 1: 表示GPIO1 2: 表示GPIO2 13: GPIO tevel 0: 表示低电平 1: 表示高电 14: GPIO mode 0: 表示低电 1: 表示开趣 1: 表示开趣 1: 表示开趣 1: 表示开趣 1: 表示开趣 1: 表示下地 2: 表示不上拉下下拉 1: 表示下地 2: 表示不上拉下下拉 1: 表示中速 2: 表示不上拉下下拉 1: 表示中速 2: 表示代速 1: 表示中速 2: 表示代速												
	指令包格式	0x05	0x00	Index_id	0x40	GPIO	GPIO_up/down	CRC							
	应答包格式	0x04	0x00	Index_id	0x40	Operation_status	CRC								
GPIO輸入模式配置	指令详解	(1) 功能说明:输入模式配置。 (2) 输入参数: Index id (设备下际ID)、GPIO(表示引脚)、GPIO_up/down(表示是否需要上/下拉电阻) (3) 返回参数: Operation_status (4) 指令代理。0x40 注:Operation_status 提作状态 0: 操作失政 1: 操作成功 注:GPIO 1: 表示GPIO1 2: 表示GPIO2 注:示GPIO_up/down 0: 表示上拉 1: 表示下拉 2: 表示不上拉不下拉													
	指令包格式	0x04	0x00	Index_id	0x41	GPIO	CRC								
	应答包格式	0x04	0x00	Index_id	0x41	Operation_status	GPIO_level_status	CRC							
GPIO读取输入引脚电平	指令详解	(1) 功能説明: 读取输入引脚电平。 (2) 輸入参数: Index_id(设备下际D)、GPIO(表示引脚) (3) 返回参数: GPIO[evel_status (4) 指令代码: 0:x41 注:GPIO [evel_status 操作状态 ①: 低电平 1: 高配平 注2: GPIO 1: 表示GPIO1 2: 表示GPIO2 注3: Operation_status 操作状态 ①: 操作失数 1: 操作成功 3: 模式不匹配(需要先配置输入模式)													
	指令包格式	0x06	0x00	Index_id	0x50	GPIO	GPIO_up/down	Trigger_mode	CRC						
	应答包格式	0x04	0x00	Index_id	0x50	Operation_status	CRC								
GPIO外部中断模式配置	指令详解	(1) 功能说明: 外部中 (2) 输入参数: Index. (3) 返回参数: Opera (4) 指令代码: Ox50 注: Operation stat 0: 操作庆政 1: 操作成功 1注2: GPIQ_up/down 0: 表示上拉 1: 表示下拉 2: 表示不上拉不 注3: Trigger, mode 0: 上升沿 1: 下降沿 2: 上升/下降沿	id (设备下标ID) tion_status us 操作状态	、GPIO(表示引謝	l)) 、Trigger_moc	te. (触发方式)									

	指令包格式						None							
	应答包格式	0x05	0x00	Index_id	0x51	GPIO	Trigger_mode	CRC						
GPIO外部中断返回	指令详解	(1) 功能说明:中断模式配置返回主动。 (2) 输入参数: Index_id(设备下标D)、GPIO(表示引脚))、Trigger_mode: (触发方式) (3) 返回参数: GPIO、Trigger_mode (4) 指令代码: 0.551 注: GFIO :表示GPIO1 :表示GPIO1 2: 表示GPIO2 注2: Trigger_mode 0: 上升语 1: 下降沿 2: 上升/下降沿												
	指令包格式	0x05	0x00	Index_id	0x60	ADC_channel1	ADC_channel2	CRC						
	应答包格式	0x04	0x00	Index_id	0x60	Operation_status	CRC							
ADC模式配置	指令详解	(2) 输入参数: Index_ (3) 返回参数: Opera (4) 指令代码: 0x60	注1: Operation_status 操作状态 0: 操作失攻 1: 操作成功 注2: ADC_channel 0: 失能適道											
	指令包格式	0x03	0x00	Index_id	0x61	CRC								
	应答包格式	0x04+Channel*3	0x00	Index_id	0x61	Operation_status	Channel_nums	ADC_value1_high	ADC_value1_low	ADC_value1_high	ADC_value1_low	CRC		
ADC读取	指令详解	(1) 功能说明: 读取采集数据。 (2) 输入参数: Index (d (设备下标D) (3) 返回参数: Channel_nums (通道数量) 、ADC_value (ADC采集值) (4) 指令代码: 0x61 注1: Operation status 操作状态 0: 操作失数 1: 操作成功 3: 模式不匹配: (未配置ADC模式, 请先配置) 注2: ADC_value = (uint16_t)((ADC_value_ligh << 8) ADC_value_low)												
	指令包格式	0x03	0x00	Index_id	0x70	CRC								
	应答包格式	0x05	0x00	Index_id	0x70	GPIO1_work_status	GPIO2_work_status	CRC						
工作钦态查询	指令详解	(1) 功能说明: 工作状态直询。 (2) 输入参数: Index_id (设备下标D) (3) 返回参数: GPIO_work_status (引脚工作状态) (4) 指令代码: 0x70 注注: GPIO] work_status 0: 表示未配置工作状态 1: 输出工作状态 2: 输入工作状态 3: 表示NVIC工作状态 4: ADC工作状态 5: I2C工作状态 5: I2C工作状态												
	指令包格式	0x03	0x00	Index_id	0xF9	CRC								
zwilano dec + c	应答包格式	0x04	0x00	Index_id	0xF9	Bootloader_version	CRC							
查询升级程序版本号	指令详解	(1) 功能说明: 查询升 (2) 输入参数: Index (3) 返回参数: Bootle (4) 指令代码: 0xF9	id (设备下标ID)											

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D) (3) 返回参数: Firmware_version (4) 指令代码: OxFA												
	指令包格式	0x03	0x00	Index_id	0xFB	CRC								
查询设备类型	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) 输入多数: Index_id (设备下标D) (3) 返回参数: Device_type (4) 指令代码: 0xF8 注1: Device_type = (uint16_t)((Device_type_high << 8) Device_type_low)												
	指令包格式	None												
枚举请求	应答包格式	0x03	0x00	0xFF	0xFC	CRC								
	指令详解	(1) 功能说明: 校举请求,chain链路变更未端设备发送、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												
	指令包格式	0x03	0x00	0xFF	0xFD	CRC								
心跳包	应答包格式	0x03	0x00	0xFF	0xFD	CRC								
	指令详解	(1) 功能说明: 心跳包 (2) 输入参数: none (3) 返回参数: none (4) 指令代码: 0xFD	, chain设备之间	定时通信,可以自	发现自己是不是末鲜	端设备,主机也可以通过	心跳包来判断是否有ch	ain设备连接。						
	指令包格式	0x04	0x00	0xFF	0xFE	Send_num	CRC							
枚举	应答包格式	0x04	0x00	0xFF	0xFE	Receive_num	CRC							
	指令详解	(1) 功能说明: 枚举获 (2) 输入参数: Send_ (3) 返回参数: Receiv (4) 指令代码: 0xFE	num (默认0,用	于记录设备个数)										