

设置蜂鸣器模式	指令包格式	0x04	0x00	Index_id	0x30	Mode	CRC						
	应答包格式	0x04	0x00	Index_id	0x30	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器模式 (2) 输入参数：Index_id (设备下标ID)、Mode (蜂鸣器模式) (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 注2：Mode 蜂鸣器的模式 (默认0) 0：自动播放模式 1：手动播放模式 2：音符播放模式											
获取蜂鸣器模式	指令包格式	0x03	0x00	Index_id	0x31	CRC							
	应答包格式	0x04	0x00	Index_id	0x31	Mode	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器模式 (2) 输入参数：Index_id (设备下标ID) (3) 返回参数：Mode (蜂鸣器模式) (4) 指令代码：0x31 注1：Mode 蜂鸣器的模式 (默认0) 0：自动播放模式 1：手动播放模式 2：音符播放模式											
设置蜂鸣器自动播放	指令包格式	0x08	0x00	Index_id	0x32	Freq_Low	Freq_High	Duty	Time_low	Time_High	CRC		
	应答包格式	0x04	0x00	Index_id	0x32	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器自动播放 (2) 输入参数：Index_id (设备下标ID)、Freq (播放频率)、Duty (播放占空比)、Time (播放时间) (3) 返回参数：Operation_status (4) 指令代码：0x32 注1：Operation_status 操作状态 0：操作失败 1：操作成功 2：模式不匹配 (需要配置自动播放模式) 注2：使用该指令需要先设置为自动播放模式 注3：Freq = (uint16_t)((Freq_High << 8) Freq_Low) 范围 100 ~ 10000 单位HZ 注4：Duty 范围0 ~ 100 注5：Time = (uint16_t)((Time_High << 8) Time_Low) 范围 0 ~ 65535, 单位毫秒											
设置蜂鸣器的频率	指令包格式	0x05	0x00	Index_id	0x33	Freq_Low	Freq_High	CRC					
	应答包格式	0x04	0x00	Index_id	0x33	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器的频率 (2) 输入参数：Index_id (设备下标ID)、Freq (播放频率) (3) 返回参数：Operation_status (4) 指令代码：0x33 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Freq = (uint16_t)((Freq_High << 8) Freq_Low) 范围 100 ~ 10000 单位HZ (默认2700)											
获取蜂鸣器的频率	指令包格式	0x03	0x00	Index_id	0x34	CRC							
	应答包格式	0x05	0x00	Index_id	0x34	Freq_Low	Freq_High	CRC					
	指令详解	(1) 功能说明：设置蜂鸣器的频率 (2) 输入参数：Index_id (设备下标ID) (3) 返回参数：Freq (播放频率) (4) 指令代码：0x34 注1：Freq = (uint16_t)((Freq_High << 8) Freq_Low) 范围 100 ~ 10000 单位HZ (默认2700)											

设置蜂鸣器的占空比	指令包格式	0x04	0x00	Index_id	0x35	Duty	CRC						
	应答包格式	0x04	0x00	Index_id	0x35	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器的占空比 (2) 输入参数：Index_id (设备下标ID)、Duty (播放占空比) (3) 返回参数：Operation_status (4) 指令代码：0x35 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Duty 范围 0 ~ 100 (默认50)											
获取蜂鸣器的占空比	指令包格式	0x03	0x00	Index_id	0x36	CRC							
	应答包格式	0x04	0x00	Index_id	0x36	Duty	CRC						
	指令详解	(1) 功能说明：获取蜂鸣器的占空比 (2) 输入参数：Index_id (设备下标ID) (3) 返回参数：Duty (播放占空比) (4) 指令代码：0x36 注1：Duty 范围 0 ~ 100											
设置蜂鸣器的状态	指令包格式	0x04	0x00	Index_id	0x37	Status	CRC						
	应答包格式	0x04	0x00	Index_id	0x37	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器的状态 (2) 输入参数：Index_id (设备下标ID)、Status (蜂鸣器的状态) (3) 返回参数：Operation_status (4) 指令代码：0x37 注1：Operation_status 操作状态 0：操作失败 1：操作成功 注2：Status 蜂鸣器的状态 (上电默认0) 0：关闭 1：开启 注3：蜂鸣器的播放频率与占空比，可以通过0x33、0x35设置。 注4：需要切换蜂鸣器为手动模式，才会生效。											
获取蜂鸣器的状态	指令包格式	0x03	0x00	Index_id	0x38	CRC							
	应答包格式	0x04	0x00	Index_id	0x38	Status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器的状态 (2) 输入参数：Index_id (设备下标ID) (3) 返回参数：Status (蜂鸣器的状态) (4) 指令代码：0x38 注1：Status 蜂鸣器的状态 (上电默认0) 0：关闭 1：开启											
设置蜂鸣器音符播放	指令包格式	0x06	0x00	Index_id	0x39	Note_index	Time_low	Time_High	CRC				
	应答包格式	0x04	0x00	Index_id	0x39	Operation_status	CRC						
	指令详解	(1) 功能说明：设置蜂鸣器音符播放 (2) 输入参数：Index_id (设备下标ID)、Note_index (音符索引)、Time (播放时间) (3) 返回参数：Operation_status (4) 指令代码：0x39 注1：Operation_status 操作状态 0：操作失败 1：操作成功 2：模式不匹配 (需要配置音符播放模式) 注2：使用该指令需要先设置为音符播放模式 注3：Note_index 音符索引 0 : REST (0 Hz) 1-12 : Octave 3 (C3, C#3, D3, D#3, E3, F3, F#3, G3, G#3, A3, A#3, B3) 13-24 : Octave 4 (C4, C#4, D4, D#4, E4, F4, F#4, G4, G#4, A4, A#4, B4) 25-36 : Octave 5 (C5, C#5, D5, D#5, E5, F5, F#5, G5, G#5, A5, A#5, B5) 37-48 : Octave 6 (C6, C#6, D6, D#6, E6, F6, F#6, G6, G#6, A6, A#6, B6) 49-60 : Octave 7 (C7, C#7, D7, D#7, E7, F7, F#7, G7, G#7, A7, A#7, B7) 61 : Octave 8 (C8 only) 注4：Time = (uint16_t)((Time_High << 8) Time_Low) 范围 0 ~ 65535, 单位毫秒											

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;
}

```