飞瑞教家居设备通信协议

前言

本协议统一数值格式低位在前，高位在后
时间格式统一为 0x 20 15 12 03，bcd 码。
设备名称，utf-8 编码
序列号长度 6 位，bcd 码，按顺序填写。

1 登陆、修改密码

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0Xfffffffffff	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xAF,0xAE.0xBF	控制类型,AF 手机端/客户端登录,AE 修改密码 BF 模拟主机登陆（通过外网连接时使用）
param_len	1		参数长度,取值 1-255
name_len	1		用户名长度
name	变长		用户名，最长 256，转成 ASCⅡ码
pass_len	1		密码长度
pass	变长		密码，md5 加密，32 位小写，后转成 ASCⅡ码，最长 256
}		// end _description	

如我输入 32 00 F1 80 11 4F 08 87 （SN 号）fe af 27 05 61 64 6d 69 6e（用户名） 20 32 31 32 33 32 66 32 39 37 61 35 37 61 35 61 37 34 33 38 39 34 61 30 65 34 61 38 30 31 66 63 33(密码)

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x40	登陆应答类型
length	1		后续包长度
result	1		登陆结果： 1.0x00:登陆成功 2.0x01:请连接应用服务器 3.0x02:用户名或者密码错误 4.0x03:重新登录 5.0x04:拒绝服务 6.0x05:协调器正在升级,请稍后(新

			增) 7.0x06:序列号错误,登陆失败 2、4、5、6 是服务器才会出现的
IP_len	1		应用服务器 IP 长度
IP			应用服务器 IP 地址
Port_len	1		应用服务器端口长度
port			应用服务器端口
}		// end _description	

数据返回：登录成功 40 01 00

2、设备控制

2.1 获取当前连接的 zigbee 设备

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0Xfffffffffff	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x81	控制类型，获取当前连接的所有设备
}		// end _description	

如我输入：0a 00 F1 80 11 4F 08 87 fe 81

请求后不仅回设备列表，还回设备状态，详细看设备状态应答

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x01	获取当前连接的所有设备应答类型
length	1		后续包长度
address	2		短地址(0x00 00 代表没有)
endpoint	1		端点地址, 取值 1-240（设备出场后就不会变）
profile	2	0x0401	固定值
device	2	低地址在前，高地址在后	设备类型: 1. 0x0001：可调光开关 2. 0x0002：电源开关 3. 0x0009：智能插座 4. 0x0101：可调亮度 LED 灯

			5. 0x0102：彩灯 LED 灯 6. 0x0103：随意贴开关（双控开关） 7. 0x0203：窗帘开关 8. 0x0107：人体（可以查状态，传感器的开启和关闭，报警有数据才会上传） 9. 0x0106：光照(目前没有这个设备) 10. 0x0302：温湿度 11. 0x0051：移动插座 12. 0x0006：红外转发器 13. 0x010a：燃气报警器 14. 0x0108：门磁感应器（可查询开关状态） 15. 0x000a：门锁 （可控可查） 16. 0x0109：漏水感应器 17. 0x0402：烟雾传感器 18 0x0054：场景控制器（暂时没用这个设备，是个面板） 19. 0x0403：无线声光报警器 操作和开光一样（可查可控） 20. 0x0204：门铃 21. 0x0404：无线紧急按键 上报信息和人体一样 22. 0x0055：背景音乐 13、16、17、21（只有报警时有数据，不能开关，只要加入一定是打开的，查询无数据）
areaId	1	0x00	区域 Id
name_len	1		设备名长度，取值 0-100（没有重名就为 00，name 长度为 0）
name	变长		设备名（重名名后的设备名，中文 UTF-8 编码后，再转 ASCⅡ码）
state	1		在线状态，取值 0-255；0 为不在线，其它值为在线 1：在线
Ieee	8		IEEE 地址（mac 地址不会变，将此与 endpoint 确定设备唯一）
sn_len	1		SN 号长度，取值 1-100
sn	变长		SN 号
}		// end _description	

数据返回

01 19 ee e9 08 04 01 02 01 设备类型 02 00 01 32 b7 97 0a 00 4b 12 00 （ieee）06 f1 80 11 4f 08 87 彩灯

01 19 a3 69 08 04 01 02 03 00 00 01 56 dd 19 01 00 4b 12 00 06 f1 80 11 4f 08 87 温湿度

01 19 a0 d9 08 04 01 08 01 00 00 01 91 8e 2e 09 00 4b 12 00 06 f1 80 11 4f 08 87 门磁

01 19 89 2a 08 04 01 **03 02** 00 00 01 bc 2d 5f 07 00 4b 12 00 06 f1 80 11 4f 08 87 窗帘

01 25 b1 9d 0a 04 01 02 00 00 0c e5 ae a2 e5 8e 85 e5 bc 80 e5 85 b3 01 61 a4 cc 01 00 4b 12 00 06 f1 80 11 4f 08 87 电源开关

01 25 b1 9d 08 04 01 02 00 00 0c e6 b5 b4 e5 ae a4 e5 bc 80 e5 85 b3 01 61 a4 cc 01 00 4b 12 00 06 f1 80 11 4f 08 87 电源开关

01 25 ab 16 08 04 01 02 00 00 0c e5 8d a7 e5 ae a4 e5 bc 80 e5 85 b3 01 c9 7c 2e 09 00 4b 12 00 06 f1 80 11 4f 08 87 电源开关

01 25 ab 16 0a 04 01 02 00 00 0c e5 8e a8 e6 88 bf e5 bc 80 e5 85 b3 01 c9 7c 2e 09 00 4b 12 00 06 f1 80 11 4f 08 87 电源开关

01 19 0b ff 08 04 01 06 00 00 00 01 89 c3 1a 01 00 4b 12 00 06 f1 80 11 4f 08 87 红外转发器

01 19 5d 67 08 04 01 09 00 00 00 01 d1 8e 2e 09 00 4b 12 00 06 f1 80 11 4f 08 87 智能插座

01 19 fe 62 08 04 01 51 00 00 00 00 07 d0 19 01 00 4b 12 00 06 f1 80 11 4f 08 87 移动插座

所有设备状态（端点地址与短地址）

74 2f 00 00 08 01 04 10 48 53 4c 2d 30 33 32 32 37 31 2d 44 5a 44 4d 46 05 61 64 6d 69 6e 09 e9 a3 9e e7 91 9e e6 95 96 08 68 64 37 6f 73 6f 67 39 （摄像头）

2.2 设置指定设备的开关状态

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x82(关闭开关设备) 0xad(关闭传感器)	控制类型， 获取当前连接的所有设备
param_len	1		参数长度，取值 1-255 注意：
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
state	1		开/关状态： 0x00：关 0x01：开 窗帘： 0x00：关 0x01：开 0x02：停止 传感器：

			0x02 关闭 0x03 开启 背景音乐: 开/关状态： 0x00：关 0x01：开 0x02：暂停 0x04：播放 0x09：上一曲 0x0A:下一曲 0x0B:返回 0x0C: 打开收音机 0x0D: 打开音乐播放器 0x0e：打开蓝牙 0x0F: 关闭蓝牙 左 右 Ok
}		// end _description	

没有应答，需查询

2.3 设置指定设备的亮度

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x83	控制类型， 设置指定设备的亮度
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
level	1		亮度，取值 3d~f1 有效取值
time	2		切换时间，低位在前，高位在后
}		// end _description	

没有应答，需查询

2.4 设置指定设备的颜色

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x84	控制类型， 设置指定设备的颜色
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
hue	1		色调，取值 0-255
saturation	1		饱和度，取值 0-255
time	2		切换时间，低位在前，高位在后，单位是毫秒
}		// end _description	

没有应答，需查询该

2.5 获取指定设备的开关状态

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x85(开关设备) 0xb1（传感器）	控制类型， 获取指定设备的开关状态
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
}		// end _description	

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x07(开关设备) 0xb2(传感器)	获取指定设备的开关状态应答类型
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址，取值 1-240
state	1		开/关状态： 0x00：关 0x01：开
}		// end _description	

2.6 获取指定设备的亮度

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x86	控制类型，获取指定设备的亮度
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
}		// end _description	

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x08	获取指定设备的亮度应答类型
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址，取值 1-240
level	1		亮度，取值 0-255
}		// end _description	

2.7 获取指定设备的色调

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x87	控制类型，获取指定设备的色调
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
}		// end _description	

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x09	获取指定设备的色调应答类型
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址，取值 1-240
hue	1		色调，取值 0-255
}		// end _description	

2.8 获取指定设备的饱和度

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x88	控制类型，获取指定设备的饱和度
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留

}		// end _description	

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0A	获取指定设备的饱和度应答类型
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址，取值 1-240
hue	1		饱和度，取值 0-255
}		// end _description	

2.9 更改指定设备名

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x94	控制类型，更改指定设备名
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
endpoint	1		端点地址，取值 1-240
name_len	1		名称长度，取值 1-32
name	变长		
}		// end _description	

没有应答，需查询

2.10 删除指定设备

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x95	控制类型，删除指定设备
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式

address	2		短地址
Ieee	8		Ieee 地址
endpoint	1		端点地址, 取值 1-240
}		// end _description	

无应答数据，需查询所有设备信息

2.11 添加设备（指除摄像头外的 zigbee 设备扫完之后自动加入）

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x9f	控制类型， 删除指定设备
}		// end _description	

应答数据，与 3.2 获取所有连接设备信息协议一样：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x01	获取当前连接的所有设备应答类型
length	1		后续包长度
address	2		短地址
endpoint	1		端点地址, 取值 1-240
profile	2	0x0401	
device	2	低地址在前，高地址在后	设备类型: 0x0001：可调光开关 0x0002：电源开关 0x0009：智能插座 0x0101：可调亮度 LED 灯 0x0102：彩灯 LED 灯 0x0103：随意贴开关 0x0202：窗帘开关 0x0107：人体 0x0106：光照 0x0301：温湿度 0x0006：红外
keep	1	0x00	保留
name_len	1		设备名长度，取值 1-100
name	变长		设备名
state	1		在线状态，取值 0-255；0 为不在线，

			其它值为在线
ieee	8		IEEE 地址
sn_len	1		SN 号长度，取值 1-100
sn	变长		SN 号
}		// end _description	

2.12 修改开关默认状态

默认状态数据已保存在设备，不用再保存在程序里面，因此程序收到此指令时，只需要往串口发送一条数据。

软件 API 接口协议（socket 接口协议）请求数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xB0	控制类型， 修改当前设备状态
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
state	1		开/关状态： 0x00：关 0x01：开
}		// end _description	

返回数据协议：

无返回数据。

2.13 查询默认状态

软件 API 接口协议（socket 接口协议）请求数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA2	控制类型， 查询设备默认状态
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址

keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
}		// end _description	

我发送数据：

21 00

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xB3	获取指定设备默认状态应答类型
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址, 取值 1-240
hue	1		00 关，01 开
}		// end _description	

2.14 查询设备锁状态协议

新增应用软件与程序查询设备锁状态的协议，设备锁状态同样保存在设备里，不需保存在程序里面，当程序接收到该指令时，只需发相关指令向设备查询就行了。

请求数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xB1	控制类型， 获取设备锁状态
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
}		// end _description	

返回数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xB2	获取指定设备锁状态应答类型
length	1	0x04	后续包长度

address	2		短地址
endpoint	1		端点地址, 取值 1-240
hue	1		00 未上锁，01 已上锁
}		// end _description	

2.15 修改设备锁状态

修改设备锁状态的协议已存在，可参考：
0xAD 启用与禁用设备

参数长度	地址模式	短地址		保留						Endpoint	保留		开/关
1-255	0x02	0-7	8-15							1-240			03/02

插座儿童锁：03 上锁，02 是解锁。
若是传感器，01 启用，00 关闭。

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xad(关闭设备锁)	控制类型，关闭设备锁，与关闭传感器指令一样
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
state	1		传感器： 0x00 关闭 0x01 开启 插座： 0x03 上锁 0x02 解锁
}		// end _description	

2.16 插座自动断电功率设置

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA3	控制类型，设置插座自动断电功率
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
Value	1		值转为 10 进制再除 10，区间为 0.1~25.5W,如为 0，则不会断电
}		// end _description	

返回数据：

暂无返回数据。

2.17 查询插座自动断电功率

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA4	控制类型，查询插座自动断电功率
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xB4	获取指定设备的电量统计应答类型
length	1	0x0E	后续包长度
address	2		短地址

endpoint	1		端点地址, 取值 1-240
state	1		结果保留 1 位小数点
}		// end _description	

2.18 设置窗帘位置或音乐音量

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xD1	控制类型， 控制窗帘位置
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
level	1		位置 0-100 (0x00 ~ 0x64)（窗帘位置） 对于背景音乐: 代表音量 0x00:静音 0xf0:音量加 0x0f:音量减
}		// end _description	

返回数据格式:同温湿度主动上传

2.19 查询窗帘位置或音乐音量

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xD2	控制类型， 获取窗帘位置
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址

keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
}		// end _description	

音量的返回数据见 6.1

2.20 设置设备数据上传间隔

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x9E	控制类型， 获取窗帘位置
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
clusterId	2		0x0402 温度 0x0405 湿度 0x0102 窗帘位置 0x0b04 是能插座功率上报 0x0406 人体/门磁单独报警的（有或者没有）
attributeId	2		温度数据或者光照、烟雾、人体，低位在前，高位在后:0x0000 窗帘:0x0008 插座:0x050b
dataType	1		数据类型: 有三种类型： 0x29 有两个字节，且有两位小数点 0x21 有两个字节，没有小数点 0x20 有一个字节，没有小数点
data	变长		根据 dataType 确定
}		// end _description	

3、分组（备用协议）

3.1 添加组

组实为区域，以名字为主键，比如开关属于那个房间。组不需单独添加，在修改开关时可以修改所属组的名字，该属性可以为空，如果属性名字不存在，主机会自动创建该组，如果存在，则添加到该组里，如果组里不存在任何设备，则自动删除。

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8f	控制类型， 添加组
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
Group Name Len	1		组名字长，长度为 0-64
Group Name			Ascii 编码
}		// end _description	

返回数据

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0B	添加指定指定设备到组应答类型
length	1		后续包长度
Group ID	2		组 ID
Group Name Len	1		组名字长，长度为 0-64
Group Name			Ascii 编码
}		// end _description	

3.2 获取组

仅获取组属性等信息，不包含成员。

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			

length	2		数据长度,低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8e	控制类型， 获取组信息
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0C	获取组信息应答类型
length	1		后续包长度
Group ID	2		组 ID
Group Name Len	1		组名字长，长度为 0-64
Group Name			Ascii 编码
}		// end _description	

3.3 获取组成员

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x98	控制类型， 获取组成员
length	1		参数长度
Group ID			指定组 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x10	获取组成员应答类型
length	1		后续包长度
Group ID	2		组 ID
Group memberacount	1		成员个数，取值范围 0-30
address	2		短地址
endpoint	1		端点地址,取值 1-240
address	2		短地址
endpoint	1		端点地址,取值 1-240
……			一直重复至开关列尽
}		// end _description	

3.4 删除组成员

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x97	控制类型，删除组成员
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
Group Name Len	1		组名字长，长度为 0-64
Group Name			Ascii 编码
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0F	删除组成员应答类型
length	1		后续包长度
Group ID	2		组 ID
Group Name Len	1		组名字长，长度为 0-64
Group Name			Ascii 编码
}		// end _description	

4、情景模式

4.1 添加情景模式

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xd0	控制类型，添加情景模式

parameLen	1		参数长度
Sence Name Len	1		情景名字长，长度为 0-64
Sence Name			情景名称，Ascii 编码
Sence logo	1		原来是场景总数 sence sum，现修改为：当前场景图片编号，0~255
status	1		1 成功、0 失败，失败原因多数应为场景数量已达上限
}		// end _description	

返回数据

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0e	获取情景模式应答类型
length	1		后续包长度
Sence ID	2		情景 ID
Sence Name Len	1		情景名字长，长度为 0-64
Sence Name			情景名称，Ascii 编码
Sence logo	1		原来是场景总数 sence sum，现修改为：当前场景图片编号，0~255
status	1		1 成功、0 失败，失败原因多数应为场景数量已达上限
}		// end _description	

4.2 获取情景模式

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x90	控制类型， 获取场景
}		// end _description	

返回数据

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0e	获取情景模式应答类型
length	1		后续包长度
Sence ID	2		情景 ID
Sence Name Len	1		情景名字长，长度为 0-64
Sence Name			情景名称，Ascii 编码
Sence logo	1		原来是场景总数 sence sum，现修改为：当前场景图片编号，0~255

status	1		状态，0，当前失效；1，当前生效；3、生效，但状态异常，关联的联动被禁用；4、锁定。场景状态为 1 的只能等于或者小于一个，处于锁定状态的场景为生效并且不能修改。
}		// end _description	

4.3 添加情景模式设备

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x91	控制类型，添加情景模式设备
param_len	1		参数长度, 取值 1-255
Sence ID	2		情景 ID
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
deviceId	2		设备类型
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x0D	获取情景模式应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		设备短地址
endpoint	1		端点地址
status	1		1 成功、0 失败，失败原因多为该情景已被删除
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）

4.4 调用情景模式

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x92	控制类型， 调用场景
paramlens	1		参数长度
Sence ID	2		情景 ID
}		// end _description	

返回数据 同 1.17 获取情景模式

4.5 获取情景模式详细信息

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8A	控制类型， 获取情景模式详细信息
paramLen	1		参数长度
Sence ID	2		情景 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x20	获取情景模式详细信息应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		短地址
endpoint	1		端点地址，取值 1-240
Device ID	2		设备类型
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
oid	1		指令序列号
End oid	1		结束序列号
}		// end _description	

4.6 删除情景设备

若短地址和endpoint 分别为0XFFFF,0XFF，主机则删除该场景

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8B	控制类型，删除情景模式指定设备
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留

endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
TaskType	1		任务类型（1.开关任务； 2.彩灯任务; 3.遥控器任务）
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
Sence ID	1		情景 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x21	获取情景模式应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		设备短地址
endpoint	1		端点地址
success	1		成功/失败
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务; 3.遥控器任务）
}		// end _description	

4.7 修改情景模式

仅限于修改图片和名称

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前， 高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xCF	控制类型， 修改情景模式
param len	1		参数长度
senceid	2		情景模式 ID
Sence Name Len	1		情景名字长，长度为 0-64
Sence Name			情景名称，Ascii 编码
Sence logo	1		原来是场景总数 sence sum，现修改为：当前场景图片编号，0~255

}		// end _description	

返回数据：同 1.18 添加情景模式

4.8 情景面板设置绑定

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xDD	控制类型，删除情景模式指定设备
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
Sence ID	2		情景 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x7a	获取情景模式绑定应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		设备短地址
endpoint	1		端点地址
success	1		0 成功/1 失败
}		// end _description	

4.9 情景面板查询绑定

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xDE	控制类型，删除情景模式指定设备
param_len	1		参数长度，取值 1-255

mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
Sence ID	2		情景 ID，目前没有，配置为 00 00
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x7b	获取情景模式绑定应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		设备短地址
endpoint	1		端点地址
}		// end _description	

4.10 情景面板解除绑定

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xDF	控制类型，删除情景模式指定设备
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址，取值 1-240
keep	2	0x0000	保留
Sence ID	2		情景 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x7c	获取情景模式绑定应答类型
length	1		后续包长度
Sence ID	2		情景 ID
address	2		设备短地址
endpoint	1		端点地址

success	1		0 成功/1 失败
}		// end _description	

4.10 情景面板按下返回数据

无请求,返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x7d	获取情景模式绑定应答类型
length	1		后续包长度
address	2		设备短地址
endpoint	1		端点地址
data	1		数据,0x01
}		// end _description	

5、定时任务

5.1 添加定时开关

每个定时开关都只能添加一个设备，默认以每天循环模式开启。
请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x9A	控制类型， 添加定时任务
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务； 4.场景调用任务）
SceneId	2		场景 id
Week type	1		周工作模式： 0x01 周一有效

			0x02 周二有效 0x04 周三有效 0x08 周四有效 0x10 周五有效 0x20 周六有效 0x40 周日有效 0x7F 每天有效
hour	1		时
minute	1		分
second	1		秒
status	1		定时状态：0，失效；1，生效；
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x12	添加定时任务应答类型
length	1		后续包长度
Task ID	1		定时任务 ID，1-255。如果 TaskID 为 0，添加失败。
}		// end _description	

5.2 获取定时开关数据

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号

flag	1	0xFE	控制标志
type	1	0x99	控制类型， 获取定时任务
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x11	获取定时任务应答类型
length	1		后续包长度
Task ID	1		定时任务 ID，0-64。
TaskType	1		任务类型（1.开关任务； 2.彩灯任务； 3.遥控器任务； 4.场景调用任务）
SceneId	2		场景 id
address	2		短地址
endpoint	1		端点地址, 取值 1-240
Week type	1		周工作模式： 0x01 周一有效 0x02 周二有效 0x04 周三有效 0x08 周四有效 0x10 周五有效 0x20 周六有效 0x40 周日有效 0x7F 每天有效
hour	1		时
minute	1		分
second	1		秒
status	1		定时状态：0，失效；1，生效；
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
}		// end _description	

5.3 删除定时任务

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x9B	控制类型， 删除定时任务
param_len	1		参数长度,取值 1-255
Task ID	1		定时任务 ID，0-64
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x13	删除定时任务应答类型
length	1		后续包长度
result	1		为 0，更新失败,否则成功
taskId	1		任务 Id
}		// end _description	

5.4 更新定时任务

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x9C	控制类型， 删除定时任务
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
Task ID	1		定时任务 ID：1-64
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务； 4.场景调用任务）
SceneId	2		场景 id
Week type	1		周工作模式：

			0x01 周一有效 0x02 周二有效 0x04 周三有效 0x08 周四有效 0x10 周五有效 0x20 周六有效 0x40 周日有效 0x7F 每天有效
hour	1		时
minute	1		分
second	1		秒
status	1		定时状态：0，失效；1，生效；
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x14	更新定时任务应答类型
length	1		后续包长度
result	1		为 0，更新失败,否则成功
taskId	1		任务 Id
}		// end _description	

5.5 修改定时状态

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号

flag	1	0xFE	控制标志
type	1	0xb5	控制类型， 修改定时任务状态
param_len	1		参数长度,取值 1-255
taskId	1		定时 ID
status	1		定时状态：0，失效；1，生效；
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x15	修改定时任务返回
length	1		后续包长度
taskId	1		定时 id
result	1		定时状态修改结果；为 0，修改失败，否则成功
taskStatus	1		定时任务的当前状态:0:失效 1:生效
}		// end _description	

5.6 查询网关时间

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC9	控制类型， 查询时间
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x18	查询时间返回
length	1		后续包长度
min	1		分
hour	1		时
day	1		日
month	1		月
year	2		年(低位在前)
}		// end _description	

5.7 同步网关时间

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xCA	控制类型， 查询联动
param_len	1		参数长度,取值 1-255
min	1		分
hour	1		时
day	1		日
month	1		月
year	2		年(低位在前)
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x19	删除联动返回
length	1		后续包长度
result	1		成功 1/失败 0
}		// end _description	

6、主动上传数据

6.1 温湿度传感器数据+设备状态主动上传

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x70	主动上传数据应答类型
length	1		后续包长度
address	2		短地址
endpoint	1		端点地址,取值 1-240
ClusterId	2		0x0402 温度 0x0405 湿度 0x0102 窗帘位置/音乐音量 0x0b04 是能插座功率上报 0x0406 人体/门磁单独报警的（有或者没有）
Acount	1		上传数据报告的数量： 人体为一个：0x01,有或者没有

			光照为一个：0x01,光照数值 温度为一个：0x01,温度数值 湿度为一个：0x01,湿度数值
attrID	2	0x00 00	温度数据或者光照、烟雾、人体，低位在前，高位在后:0x0000 窗帘:0x0008 插座:0x050b
dataType	1	0x29	有三种类型： 0x29 有两个字节，且有两位小数点 0x21 有两个字节，没有小数点 0x20 有一个字节，没有小数点 0x23 有三个字节，分别代表背景音量三个状态 状态 1 为 02，对应 2 进制 0000 0010 状态 2 为 13，对应 2 进制 0001 0011 状态 3 为 82，对应 2 进制 1000 0010
value	2		温度数据，低位在前，高位在后。整合后再除 100.
attrID	2	0x04 00	湿度数据
Unit8	1	0x20	代表一个字节长度，没有小数点
Value	2		湿度，转 10 进制
}		// end _description	

例子：温湿度传感器上报数据

70 10 85 06 08 04 01 02 00 00 29 88 0c 04 00 29 14 1a

开关按物理按键时，也会主动上传数据，如果是开关数据，如下：

70 0A 27 A0 09 04 01 01 00 00 20 00

ClusterId :0x0B04 attrId:0x050B 代表智能插座功率上报

例子:背景音乐:

70 0d 2e dc 08 02 01 01 08 00 23 82 10 82 00

状态 2 为 10，对应 2 进制 0001 0000

第一状态位

开关状态	状态值	介质（无、U 盘、SD）	播放状态
关机	<0x80		
开机	0x80	无	停止
	0x81	无	暂停
	0x82	无	播放
	0x90	U 盘	停止
	0x91	U 盘	暂停
	0x92	U 盘	播放

	0xa0	SD	停止
	0xa1	SD	暂停
	0xa2	SD	播放

第二状态位, 播放状态和主界面状态

播放状态	状态值	主界面	
不播放	0x00	主界面	
	0x01	音乐	
	0x02	图片	
	0x03	收音	
	0x04	aux	
	0x05	日历	
	0x06	设置	
	0x07	蓝牙	
	0x08	文件浏览器	
USB 播放	0x10	主界面	
	0x11	音乐	
	0x12	图片	
	0x13	收音	
	0x14	aux	
	0x15	日历	
	0x16	设置	
	0x17	蓝牙	
	0x18	文件浏览器	
SD 播放	0x20	主界面	
	0x21	音乐	
	0x22	图片	
	0x23	收音	
	0x24	aux	
	0x25	日历	
	0x26	设置	
	0x27	蓝牙	
	0x28	文件浏览器	

第三位状态，音量

音量	值	音量等级	
静音	>= 0x80	0	
有声	0x00	0	
	0x01	1	
	0x02	2	
	0x03	3	
	0x04	4	

	0x05	5	
	0x06	6	
	0x07	7	
	0x08	8	
	0x09	9	
	0x0a	10	
	0x0b	11	
	0x0c	12	
	0x0d	13	
	0x0e	14	
	0x0f	15	
	0x10	16	
	0x11	17	
	0x12	18	
	0x13	19	
	0x14	20	
	0x15	21	
	0x16	22	
	0x17	23	
	0x18	24	
	0x19	25	
	0x1a	26	
	0x1b	27	
	0x1c	28	
	0x1d	29	
	0x1e	30	

7、电量查询

7.1 区间电量查询

本程序先保存设备上传的电量数据，然后设计应用软件与软件通信接口。
程序保存数据量 365 * 2（2 年）条，如大于 730 条数据，则删除最旧一条数据。设备每小时上传一次数据，程序接收到数据后按当前日期进行累加。
请求数据协议：

结构	长度（byte）	标识符	备注
_ description (){			

length	2		数据长度,低字节在前，高字节在后
sn	4	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA5	控制类型， 获取指定设备的电量
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
Start time	4	0x20 0x14 0x10 0x12	开始时间
End time	4	0x20 14 10 12	结束时间
}		// end _description	

返回数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xB5	获取指定设备的电量统计应答类型
length	1	0x0b	后续包长度
address	2		短地址
endpoint	1		端点地址,取值 1-240
Start time	4	0x20141012	统计开始时间
state	4		结果保留两位小数点
}		// end _description	

7.2 电量清零

只清除程序数据，其它不用管。

请求数据协议：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA6	控制类型， 数据清零
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留

Start time	4	0x20141012	开始时间
End time	4	0x20141012	结束时间
}		// end _description	

返回数据：
没有返回数据。

7.3 区间电量返回协议

主机主动上传区间电量数据，无请求数据。插座打开和关闭期间产生的用电电量。
返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xB6	获取指定设备的电量统计应答类型
length	1	0x0E	后续包长度
address	2		短地址
endpoint	1		端点地址, 取值 1-240
state	4		结果保留两位小数点
}		// end _description	

8、遥控器

8.1 红外控制

请求数据结构

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8C	控制类型
param_len	1		参数长度,取值 1-255 （这指什么？）
mode	1	0x02	地址模式
address	2		短地址(红外转发器的短地址)
keep	6	0x000000000000	保留

endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
Data_len	1		指令长度
Data			指令（红外指令固定开头为 30 00，空调为 30 01）
}		// end _description	

例：我发送 风扇开的命令

22 00 f1 80 11 4f 08 87 fe 8c 17 02 cc 31 00 00 00 00 00 00 08 00 00 0a 30 00（红外指令固定开头） 63 20 00 03 03 00 00 b9（校验位，红外指令部分各位和的最后一字节）

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x71	红外接收是否成功
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址, 取值 1-240
status	1		0 失败，1 成功
}		// end _description	

8.2 红外学码

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x8d	控制类型，学码
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
remotetype	1		遥控器类型
cols	1		低位在前
rows	1		
Key_sn	2		键值序列号
}		// end _description	

应答数据结构

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x72	红外学码
length	1	0x04	后续包长度
address	2		短地址
endpoint	1		端点地址, 取值 1-240
Order_len			对应指令长度
remotetype	2		遥控器类型
cols	2		低位在前
rows	1		
Key_sn	2		键值序列号
Key_order	1		对应指令码
}		// end _description	

8.2 红外加入情景

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA7	控制类型，学码
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
key_status	1		开关状态
Key_class	1		类型
Key_Brand	1		品牌
Key_type	1		型号
Key_lengh	1		指令长度
Key_order			指令码
}		// end _description	

8.4 保存遥控器

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xA8/0xAB	控制类型，A8 保存遥控器 AB 修改遥控器
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x000000000000	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
device_type	2		设备类型（固定的红外）0006
remotetype	2		遥控器类型（空调 49152、DVD24576、机顶盒 16384、风扇 32768、电视 8192、投影仪 40960、自定义）
cols	2		低位在前 （品牌，如创维、格力）
rows	1		（品牌型号）？需要继续确定
name Len	1		0-32 名字长度
name			变长

返回数据

同步遥控器

安卓调 c 指令（.so 文件）

重要数据（device_type、remotetype、cols、rows 决定了使用哪类遥控器）

8.5 获取遥控器

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xa9	控制类型， 获取遥控器
}		// end _description	

例如：0a 00 f1 80 11 53 a5 52 FE a9

73 17 25 d3 08 06 00（红外转发器） 00 80(类型为风扇) 00 00（第 0 组） 02（第三行） 0c e6 a0 bc e5 8a 9b 28 47 72 65 65 29

返回数据

结构	长度（byte）	标识符	备注

_ description (){			
tag	1	0x73	获取遥控器应答类型
length	1		后续包长度
address	2		短地址
endpoint	1		端点地址,取值 1-240
device_type	2		设备类型
remotetype	2		遥控器类型
cols	2		低位在前（选定指令组）
rows	1		（选定品牌）
name Len	1		0-32 名字长度
name			变长
}		// end _description	

73 17 25 d3 08 06 00 00 80 00 00 02 0c e6 a0 bc e5 8a 9b 28 47 72 65 65 29

ff 01 12

73 17 25 d3 08 06 00 00 80 00 00 02 0c e6 a0 bc e5 8a 9b 28 47 72 65 65 29

8.6 删除遥控器

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xAA	控制类型， 删除遥控器
param_len	1		参数长度,取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
remoteType	2		遥控器类型
cols	1		
rows	2		低位在前
}		// end _description	

返回数据：

同步遥控器

9、摄像头

9.1 添加摄像头

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号（主机的）
flag	1	0xFE	控制标志
type	1	0xC0	控制类型， 添加摄像头
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
sinlen	1		摄像头设备 ID 长度
sin			设备 ID 变长（摄像头的 sn 号，二维码可扫）
AccountLen	1		账号长度
Account	变长		账号
namelen	1		名字长度
name			变长
pwdlen	1		密码长度
pwd			密码 变长
}		// end _description	

返回数据：

【先调用摄像头修改密码的 SDK 的方法（带密码参数，但是 app 中是自动生成）
待返回后，发此添加指令给主机（主机只是记录）】

9.2 获取摄像头

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后

sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC1	控制类型，获取摄像头
}		// end _description	

例如我发送： 0a 00 f1 80 11 53 a5 52 FE C1

返回 ff 01 06 代表没有摄像头

当摄像头加入后返回

74 28 00 00 08 01 04 10 48 53 4c 2d 32 33 31 35 30 34 2d 47 58 4a 55 56 05 61 64 6d 69 6e 02 63 6d 08

74 2d 00 00 08 01 04 10 48 53 4c 2d 33 30 36 34 38 39 2d 48 50 50 57 58 05 61 64 6d 69 6e 07 57 49 46 49 43 41 4d（用户名，对照 asc Ⅱ码） 08 66 62 77 7a 69 37 65 65（密码对照 asc Ⅱ码即可获得）

返回数据

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x74	获取遥控器应答类型
length	1		后续包长度
address	2		短地址（摄像头默认是 00 00）
endpoint	1		端点地址,取值 0-240（摄像头默认是 08）
type	2	0x0401	摄像头类型 (低位在前，高位在后)
sinlen	1		摄像头 ID 长度 （一般是 0x10）
sin			摄像头 ID （摄像头的二维码扫出的 sin 号）（16 位）
AccountLen	1		账号长度
Account	变长		账号
namelen	1		摄像头名称长度
Name			摄像头名称
pwdlen	1		密码长度
pwd			密码
}		// end _description	

9.3 修改摄像头数据

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志

type	1	0xC2	控制类型， 修改摄像头名称
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
sinlen	1		摄像头 ID 号长度
sin			摄像头 ID
AccountLen	1		账号长度
Account	变长		账号
namelen	1		名称长度
name			名称
pwdlen	1		密码长度
pwd			密码
}		// end _description	

返回数据：
同步摄像头
页面也是 SDK 的方法

9.4 删除摄像头

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前， 高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC3	控制类型， 删除摄像头
param_len	1		参数长度, 取值 1-255
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址, 取值 1-240
keep	2	0x0000	保留
sinlen	1		摄像头 ID 号长度
sin			摄像头 ID
}		// end _description	

返回数据：
同步摄像头

12、报警日志 （需连外网，日志服务器在阿里云）

本协议仅针对连接外网的设备有效，内网暂时无法读取报警日志。

请求数据协议：23 00 f1 80 11 56 fe 62 fe be 11 02 12 c4 08 00 00 df 07 0b 01 12 3f 3e df 07 0b 02 12 3f 3e 01 00 0a 00

结构	长度（byte）	标识符	备注
_description (){			
length	2		数据长度，低字节在前，高字节在后
sn	5	0xFFFFFFFF	被控端 SN 号
flag	1	0xFF	控制标志
type	1	0xbe	控制类型，获取指定设备的电量
param_len	1		参数长度，取值 1-255
mode	1	0x02	地址模式
address	2		短地址
endpoint	1		端点地址，取值 1-240
keep	5	0x0000000000	保留
Start time	7	0x20 0x14 0x10 0x12 hhmmss	开始时间如果是查询当天的报警日志则发送当前系统时间，如果是查询其他某天的报警日志，则发送时间为某天的 23:59:59 （只按天数查询）
End time	7	0x20 0x14 0x10 0x12 hhmmss	结束时间 如果是查询当天的报警日志则发送当前系统时间，如果是查询其他某天的报警日志，则发送时间为某天的 23:59:59 （只按天数查询）
Current point	2		本次开始查询点，低位在前，高位在后
lentth	2		本次查询长度，低位在前，高位在后
}		//end _description	

返回数据协议：

结构	长度（byte）	标识符	备注
_description (){			
tag	1	0xBd	获取指定设备的电量统计应答类型
length	1		后续包长度
address	2		短地址
endpoint	1		端点地址，取值 1-240
Datatime	7	0x20141012 0xhhmmss	报警日期

Sum	2	0x0000	返回报警设备报警信息总条数 低位在前
status	1		状态：1 报警，2 电量少于 20%，3 电量少于 5% 0 无报警信息返回 4 返回异常
}		// end _description	

13、随意贴 （暂不管，双控开关）

13.1 随意贴绑定设备

请求数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x89	控制类型， 随意贴绑定
param_len	1		参数长度, 取值 1-255
address	2		随意贴短地址
endpoint	1		随意贴端点 EP, 取值 1-240
ieee	4	0x00 00 00 00	随意贴 ieee 地址
to_address	2		目标短地址
To_endpoint	1		目标开关 EP
To_ieee	4		目标开关 ieee 地址
ClusterID	2	0x06 0x00	Clusterid，固定值
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x29	绑定结果上传信息
length	1	0x03	后续包长度
address	2		短地址
result	1		00 成功，否则失败
}		// end _description	

13.2 随意贴取消绑定

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0x96	控制类型， 获取指定设备的电量
param_len	1		参数长度，取值 1-255
address	2		随意贴短地址
endpoint	1		随意贴端点地址，取值 1-240
ieee	4	0x00 00 00 00	随意贴 ieee 地址
to_address	2		目标短地址
To_endpoint	1		目标开关短地址
To_ieee	4		目标开关 ieee 地址
ClusterID	2	0x06 0x00	Clusterid，固定值
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x28	解除绑定上传信息
length	1	0x03	后续包长度
address	2		短地址
result	1		00 成功，否则失败
}		// end _description	

13.3 随意贴绑定设备查询

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xBD	控制类型， 获取指定设备的电量
param_len	1		参数长度，取值 1-255
address	2		随意贴短地址
endpoint	1		随意贴端点地址，取值 1-240

}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x27	解除绑定上传信息
length	1	0x06	后续包长度
srcAddr	2		随意贴短地址
srcEp	1		随意贴端点
dstAddr	2		绑定端短地址
dstEp	1		绑定端端点
}		// end _description	

13.4 随意贴保存绑定

请求数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xCB	控制类型， 随意贴绑定
param_len	1		参数长度, 取值 1-255
address	2		随意贴短地址
endpoint	1		随意贴端点 EP, 取值 1-240
ieee	4	0x00 00 00 00	随意贴 ieee 地址
to_address	2		目标短地址
To_endpoint	1		目标开关 EP
To_ieee	4		目标开关 ieee 地址
ClusterID	2	0x06 0x00	Clusterid，固定值
}		// end _description	

返回数据无

13.5 随意贴保存解绑

请求数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度, 低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志

type	1	0xCC	控制类型， 随意贴绑定
param_len	1		参数长度,取值 1-255
address	2		随意贴短地址
endpoint	1		随意贴端点 EP,取值 1-240
ieee	4	0x00 00 00 00	随意贴 ieee 地址
to_address	2		目标短地址
To_endpoint	1		目标开关 EP
To_ieee	4		目标开关 ieee 地址
ClusterID	2	0x06 0x00	Clusterid，固定值
}		// end _description	

返回数据无

14、联动

14.1 添加联动

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC4	控制类型， 添加联动
param_len	1		参数长度,取值 1-255
address	2		触发器短地址
endpoint	1		触发器端点地址,取值 1-240
linkId	2		联动 id
factor	1		条件(大于 1,等于 2,小于 3)
attrId	2		值的意义
value	2		触发器的有效值
sceneId	2		包含的场景
starttime	2	分 时	执行开始时间段，例 18：30
endtime	2	分 时	执行结束时间段，例 18：30
repeat	1	0 or 1	时间段内，1 每次触发都执行，0 每天只执行一次
status	1		联动状态：0，失效；1，生效；2，生效且锁定。每个状态 1 的联动只允许存在一个。
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x22	添加联动返回
length	1		后续包长度
srcAddr	2		触发器短地址
srcEp	1		触发器端点
linkID	2		联动 ID(00 为失败)
}		// end _description	

14.2 添加联动设备

请求数据

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC6	控制类型， 添加情景模式设备
param_len	1		参数长度,取值 1-255
link ID	2		联动 ID
mode	1	0x02	地址模式
address	2		短地址
keep	6	0x00 00 00 00 00 00	保留
endpoint	1		端点地址,取值 1-240
keep	2	0x0000	保留
deviceId	2		设备类型
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留
Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x2b	添加联动设备返回
length	1		后续包长度
linkID	2		联动 ID
address	2		设备短地址
endpoint	1		端点地址
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）
status	1		1 成功、0 失败，失败原因多为该情景已被删除
}		// end _description	

14.3 查询联动

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC5	控制类型， 查询联动
param_len	1		参数长度，取值 1-255
Sence ID	2		场景 ID
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x23	查询联动信息
length	1		后续包长度
linkID	2		联动 ID
address	2		触发器短地址
endpoint	1		触发器端点地址，取值 1-240
factor	1		条件（大于 1，等于 2，小于 3）
attrId	2		值的意义
value	2		触发器的有效值
sceneId	2		指定场景执行
starttime	2	时 分	执行开始时间段，例 18：30

endtime	2	时 分	执行结束时间段，例 18：30
repeat	1	0 or 1	时间段内，1 每次触发都执行，0 每天只执行一次
status	1		联动状态：0，失效；1，生效；2，生效且锁定。每个状态 1 的联动只允许存在一个。
}		// end _description	

14.4 查询联动设备信息

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度，低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xD3	控制类型， 获取联动设备信息
paramLen	1		参数长度
linkID	2		联动 ID
}		// end _description	

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x2c	获取情景模式详细信息应答类型
length	1		后续包长度
linkID	2		情景 ID
address	2		短地址
endpoint	1		端点地址，取值 1-240
Device ID	2		设备类型
remoteType	2		遥控器类型
cols	2		列--对应型号
rows	1		行--对应品牌
TaskType	1		任务类型（1.开关任务； 2.彩灯任务；3.遥控器任务）
Data1	1		开关/遥控器开关数据
Data2	1		亮度/遥控器温度值
Data3	1		色调/遥控器风量
Data4	1		饱和度/遥控器手动风向
Data5	1		遥控器自动风向
Data6	1		遥控器模式(抽湿...)
Data7	1		预留

Data8	1		预留
Data_len	1		遥控器指令值总长度
Data9	变长		遥控器指令
oid	1		指令序列号
End oid	1		结束序列号
}		// end _description	

14.5 修改联动状态

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xCE	控制类型， 修改联动状态
param_len	1		参数长度,取值 1-255
Linkid	2		联动 ID
status	1		联动状态：0，失效；1，生效； 2，上锁。3 解锁 每个状态 1 的联动只允许存在一个。
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x24	删除联动返回
length	1		后续包长度
linkid	2		联动 id
result	1		联动状态修改结果
linkStatus	1		联动当前状态:0:失效 1:生效 2:上锁且失效 3:上锁且生效
}		// end _description	

14.6 删除联动

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志

type	1	0xC7	控制类型， 查询联动
param_len	1		参数长度,取值 1-255
linkid	2		联动 id
}		// end _description	

返回数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x25	删除联动返回
length	1		后续包长度
linkid	2		联动 id
result	1		1 成功/0 失败
}		// end _description	

14.7 联动被调用(暂不使用)

无请求数据,主动返回：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x2a	修改联动返回
length	1		后续包长度
srcAddr	2		触发器短地址
srcEp	1		触发器端点
linkID	2		联动 ID
factor	1		条件(大于 1,等于 2,小于 3)
value	2		触发器的有效值
}		// end _description	

15 获取软件版本

请求数据格式：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前， 高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xCD	控制类型， 随意贴绑定
}		// end _description	

返回数据格式

结构	长度（byte）	标识符	备注

_ description (){			
tag	1	0x30	协调器版本号
length	1		后续包长度
SN	6		协调器 SN
corVersion	1		协调器版本号
sysVersion	2		ipk 软件版本号(低位在前)
}		// end _description	

16、数据同步

16.1 获取所有信息

请求数据：

结构	长度（byte）	标识符	备注
_ description (){			
length	2		数据长度,低字节在前，高字节在后
sn	6	0xFFFFFFFF	被控端 SN 号
flag	1	0xFE	控制标志
type	1	0xC8	控制类型， 查询联动
}		// end _description	

16.2 信息获取失败

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0xFF	获取信息失败或结果为空
length	1	0x01	后续包长度
tagtype	1		0x01 设备为空 0x02 对应双控开关未绑定 0x03 绑定_解绑被占用 0x04 场景里面的设备为空 0x0e 情景为空 0x11 定时开关为空 0x12 遥控器为空 0X13 获取分组为空 0x04 改分组内设备为空 0x14 获取联动信息为空 0x05 获取联动设备返回为空
}		// end _description	

17、网关出错崩溃

17.1 网关崩溃自动返回信息

返回数据：

结构	长度（byte）	标识符	备注
_ description (){			
tag	1	0x31	协调器版本号
length	1	0x01	后续包长度
Error	1	02/03/04...暂无定义	崩溃类型编号(测试用)
}		// end _description	