概述

分布式联动体系中，设备类型分为：条件设备、动作设备、条件+动作设备。
其中条件设备负责发送一些事件信息（如温度、湿度、亮度、开关等）；动作设备中集成了逻辑处理引擎，会订阅相关事件信息，然后计算配置的联动逻辑是否满足，并执行相关动作。

本文介绍基于 MESH 通用对接 2.6 版本及以上的固件，来开发具备分布式联动能力的设备。

一、条件设备开发

参考《蓝牙 Mesh 通用串口协议》，实现基本的功能（从开发者平台上下载的 MCU_SDK 正常跑起来，可以配网即为基本功能）。

对于分布式联动的条件设备开发，还需要实施以下步骤：
1）设备处于未配网状态时，通过 [Mesh 互传使能（CMD-0xB1）] 命令；
2）设备在配网成功后，会通过 [获取 8 个 PUB_ADDRESS（CMD-0xB3）]命令，通知 MCU 8 个事件发布地址。
- MCU 可以在 BT 主动通知 MCU 8 个 PUB_ADDRESS 时，将其存储在自己的 flash 中，用的时候从 flash 中取出
- MCU 也可以，在 BT 每次报告自己的工作状态 [发送模组工作状态（CMD-0x03）]，且状态为已绑定时，主动发送 [获取 8 个 PUB_ADDRESS（CMD-0xB3）]命令，将获取的值存储在全局变量中（这样 MCU 可以不用编写 flash 相关代码）
3）在原有代码基础上，每次调用 [状态上报（CMD-0x07）] 命令时，根据产品定义，如果该 dp 需要作为条件进行发布，需要用下面命令将其发布：（并不是所有需要上报的 dp 都需要作为条件事件发布！！）

以一个 bool 类型、dpid = 1 的门开事件为例子：

ref_code1.txt:

```
/*
 * check_sum 用于从帧头开始按字节求和得出的结果对 256 求余
 * create_data_frame 用于MESH 通用对接 MCU to BT 组包
 */
u8 check_sum(u8 *data,u8 len){
    u8 i,sum = 0;
    for(i=0;i<len;i++){
        sum+=data[i];
    }
    return sum;
}

u8 create_data_frame(char cmd, u8 *value, u8 data_len, u8 *frame_buf, u8
frame_buf_len){
    frame_buf[0] = 0x55;
    frame_buf[1] = 0xaa;
    frame_buf[2] = 0x00;
    frame_buf[3] = cmd;
    frame_buf[4] = (data_len >> 8) & 0xff;
    frame_buf[5] = data_len & 0xff;

    memcpy(&frame_buf[6],value,data_len);
    frame_buf[data_len+6]=check_sum(frame_buf,data_len+6);

    return (data_len+7);
}

/*
 * 高级封装 MESH 通用对接 MCU to BT 组包，并发送
 */
void auc_mcu_to_mesh(u8 cmd,u8 *value,u8 len){
    u8 frame_buf[256];
    u8 frame_len;

    //将 data 放入 frame,frame 是格式化的通信数据
    frame_len = create_data_frame(cmd,value,len,frame_buf,F_MAX_LEN);

    if(frame_len == 0)return;
    hal_uart_send(frame_buf,frame_len);//<--- 这个是底层串口发送函数，入参是一个数组和数组
长度
}

/*
 * 高级封装 本地联动配置透传（增删改查）
 */
void auc_mcu_to_mesh_local_auto_send(u16 dst_addr, u8 cmd, u8 *datas, u8 datas_len)
{
    u8 param[256];
    param[0] = (dst_addr >> 8) & 0xFF;//dst_addr
    param[1] = dst_addr & 0xFF;
    param[2] = 0;//no-ack
    param[3] = datas_len+2;//len

    param[4] = cmd;
```

```
    param[5] = datas_len;//len
    memcpy(&param[6],datas,datas_len);
    auc_mcu_to_mesh(0xc4,param,datas_len+6);
}

static void __door_report_false(char input){
      /*
       * 普通 dp 上报的 len 字段是 2 字节；事件消息发布 len 字段是 1 字节；其他无区别
       */
    u8 msg1[5] = {0x01,0x01,0x00,0x01,0x00};//dpid=1;bool;len(2B)=1;false
    u8 msg2[4] = {0x01,0x01,     0x01,0x00};//dpid=1;bool;len(1B)=1;false
    auc_mcu_to_mesh(0x07,msg1,5);//普通 DP 上报（到网关或手机）
    auc_mcu_to_mesh_local_auto_send(pub_address,0xA5,msg2,sizeof(msg2));//联动上报
}
```

说明：上面代码段实现了 0x07 命令 dp 数据上报和 0xC4 命令条件信息发布。**MCU 开发者原来代码可能已经实现了**
auc_mcu_to_mesh 功能，那么该函数及其上面的两个函数可以去掉，换为自己的就行。

第 64 行的 pub_address 就是第 2 步中获取的 8 个中的第一个。

二、动作设备开发

参考《蓝牙 Mesh 通用串口协议》，实现基本的功能（从开发者平台上下载的 MCU_SDK 正常跑起来，可以配网即为基本功能）。

动作设备开发相对简单，在原有开发好的程序基础上，加上下面新增命令处理：

动作通知（分布式联动动作通知） 0xC3

当通用对接使用分布式联动功能时（作为被控设备），当配置的联动导致 DP 相关的动作时，会通过 0xC3 指令通知 MCU，做相应处理。

模组发送

序号	长度（字节）	字段	说明
0 1	2	帧头	0x55 0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0xC3
4 5	2	数据长度 Len	Len 高 8 位 Len 低 8 位
6~6+Len-1	Len	datas	见下表
6+Len	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

datas：

序号	长度（字节）	字段	说明
0~1	2	autoid	自动化 ID
2	1	dpid	DP ID
3	1	dpkind	DP 类型（参考 [DP 格式说明]）
4	1	op 运算符	参考下图：图 1. 动作描述
5	1	datas_len	参数长度
6~(5+datas_len)	datas_len	datas	参数（长度会随着 dp 类型和运算符的不一样而不一样）

<table>
<tr><th>动作类型</th><th colspan="4">动作描述</th></tr>
<tr><td>1 byte</td><td>1 byte</td><td>4 bits</td><td colspan="2">4 bits</td><td>0\1\2\N</td></tr>
<tr><td rowspan="8">0 : dp</td><td rowspan="8">dpid</td><td>dp类型</td><td colspan="2">运算符</td><td>参数</td></tr>
<tr><td rowspan="4">Value = 0</td><td>:= a 赋值</td><td>1</td><td rowspan="3">a (2bytes)</td></tr>
<tr><td>+= a 累加</td><td>2</td></tr>
<tr><td>-= a 累减</td><td>3</td></tr>
<tr><td>...</td><td></td><td></td></tr>
<tr><td>Bool = 1</td><td>:=! 取反</td><td>0</td><td>无参数</td></tr>
<tr><td>Enum = 2</td><td>:= a 赋值</td><td rowspan="2">1</td><td>a (1bytes)</td></tr>
<tr><td>Raw = 3
String = 4</td><td>:= a 赋值</td><td>a [长度N (1byte) + N 字节的数据]</td></tr>
<tr><td>1 : 延时</td><td colspan="4">延时时间 (2byte)

- 单位 S
- 最大不超过 5 小时</td></tr>
<tr><td>2 : 自动化</td><td colspan="3">自动化 ID (2bytes)</td><td>禁用 = 0 / 启用 = 1/ 执行 = 2</td></tr>
<tr><td>...</td><td colspan="4"></td></tr>
</table>

图 1. 动作描述

参考代码：（当收到 0xC3 串口命令后，MCU 按照上表提取对应数据，送入下面函数）

ref_code2.txt:

```
static void ty_mesh_local_auto_action_cb(u16 autoid, u8 dpid, u8 dpkind, u8 op ,u8
*datas, u8 datas_len){
    switch(op){
    case 0:break;//取反
    case 1:{
           switch(dpid){
                case 1:
                    //app_light_set_onoff(datas[0]);
                    break;
                case 2:
                    //app_light_set_mode(datas[0]);
                    break;
                case 3:{
                    //u8 lum = ((datas[0]<<8) | datas[1])/10;
                    //app_light_set_lum(lum);
```

```
                    }break;
                case 4:{
                    //u8 wc = ((datas[0]<<8) | datas[1])/10;
                    //app_light_set_wc(wc);
                    }break;
                case 5:{
                    //PR_DEBUG("SET COLOR, datas=[");
                    //PR_DEBUG_HEX_ARRAY(datas,datas_len);
                    //PR_DEBUG_RAW("]\n");
                    }break;
                case 6:{
                    //u8 scene_id = (datas[0]-'0')<<8 | (datas[1]-'0');
                    //app_light_set_scene(scene_id);
                              }break;
                default:break;
             }
            }
            break;//赋值
    case 2:break;//累加
    case 3:break;//累减
    default:break;
    }
}
```

三、条件+动作设备开发

只需要将上述两个修改要求同时实现即可。