# 背景

无

# 硬件端

1. 获取温湿度、烟雾浓度、火焰状态
2. 通过OLED显示的各个传感器设备的数据
3. 当烟雾浓度，温湿度超过设定阈值，LED报警灯亮和蜂鸣器响，实现声光报警
4. 通过ESP8266自动进行组网功能
5. 设备断网后自动重连机制

# APP

1. 可以查看各个传感器状态
2. 可以修改设备的各个数据阈值
3. 使用数据库进行数据储存与查询（按时间段）；
4. 可以远程修改蜂鸣器的状态

## 设备信息：

1. APP设备信息

#define PROID " 625921" // 产品ID

#define DEVID " 1181073141" // 设备ID

#define AUTH\_INFO "1" // 鉴权信息

1. 底层设备信息

#define PROID " 625921" // 产品ID

#define DEVID " 1181073142" // 设备ID

#define AUTH\_INFO "2" // 鉴权信息

1. MQTT主题

#define S\_TOPIC\_NAME "somg\_topic" // 硬件订阅的主题

#define P\_TOPIC\_NAME "somgapp\_topic" // 硬件发布的主题

# 协议对接：

## 设备->APP

{“temp”,”float” , “temp\_v”,”int” , “humi”, ”float”, “humi\_v”, ”int”, “somg”,”int”, “somg\_v”,”int”, “flage”,”int”,“fire”,”int” , “beep”,”int” }

解释：

temp：环境温度，取值范围0-100 ；

humi：环境湿度，取值范围0-100 ；

somg：烟雾浓度，取值范围0-400，越大，烟雾越高 ；

temp\_v：环境温度阈值，取值范围0-100 ；

humi\_v：环境湿度阈值，取值范围0-100 ；

somg\_v：烟雾浓度阈值，取值范围0-400，越大，烟雾越高 ；

flage：模式的选择，0为自动模式，1为手动模式；

fire：火焰状态，0为未检测到，1为检测到；

beep：蜂鸣器状态，0为关闭，1为开启；

## APP->设备

1. 0x01

{

"cmd":0x01,

"data": {

"flage": x

}

} (该数据表示设备的模式选择)，设备回{ “flage”,X}

2. 0x02

{

"cmd":0x02,

"data": {

" temp\_v ": x

" humi\_v ": x

" somg\_v ": x

}

} 同上

2. 0x03

{

"cmd":0x03,

"data": {

"beep": “1”

}

} beep：蜂鸣器状态，0为关闭，1为开启