

深圳聚点互动科技有限公司

智能家居对接协议

4.2

目录

1 对接基本要求.....	3
2 连接方式.....	3
3 协议格式.....	4
4 指令分配规则.....	5
5 各指令发送的数据.....	6
5.1 心跳请求.....	6
5.2 心跳响应.....	6
5.3 请求设备列表.....	6
5.4 返回设备列表，设备列表有更新.....	7
5.5 请求场景列表.....	9
5.6 返回场景列表.....	9
5.7 请求房间列表（可选）.....	10
5.8 返回房间列表（可选）.....	10
5.9 请求楼层列表（可选）.....	11
5.10 返回楼层列表（可选）.....	11
5.11 发送控制指令.....	12
5.12 响应控制指令，设备状态有更新.....	13
5.13 让背景音主机，主动说话（可选）.....	14
6 返回状态码.....	14
7 常见设备类型.....	15
8 控制指令.....	15
9 属性模式说明.....	157

1 对接基本要求

- 1、能够提供所有设备列表数据，设备信息中要包含设备 ID、设备名、设备类型、设备状态（可选）、房间 ID（可选）、楼层 ID（可选）。
- 2、能够提供所有场景列表数据，场景信息中要包含场景 ID、场景名。
- 3、能够提供所有房间列表数据，房间信息中要包含房间 ID、房间名。（可选）
- 4、能够提供所有楼层列表数据，楼层信息中要包含楼层 ID、楼层名。（可选）

2 连接方式

- 1、背景音主机发送 UDP 广播，端口（暂定）6666，发送数据如下：

```
{
  "type": "REQUEST_TCP",
  //背景音主机的sn码
  "sn": "xxxxx"
}
```

- 2、服务器收到背景音主机发送的广播后，发送 UDP 广播，端口（暂定）7777，发送数据如下：

```
{
  "type": "RESPONSE_TCP",
  "data": {
    "ip": "xxx.xxx.xxx.xxx",
    "port": 8888,
    "company": "xxxx"    //company字段代表对接合作方
    "protocolVersion": "v1"  //protocolVersion 字段表示使用的版本(可选 v1 和 v2。协议格式对应有所改变，v1 协议头是 0xAA,长度 2 字节。V2 协议头为 0xAB,长度 3 字节。v2 版本能接收更多的设备数据)
  }
}
```

- 3、背景音主机获取到服务器的 IP 后，通过 TCP 连接服务器，端口（暂定）8888.
- 4、然后通过下面的协议格式获取数据。

3 协议格式

protocolVersion=v1/v2

协议头 (0xAA)	长度 (0XXXXX)	指令编号 (0XXXXX)	数据	校验 (0XXXXX)
一个字节，v1 为固定的 0xAA, v2 为 0xAB	v1 为两个字节，v2 为三个字节，为协议中“数据”部分的长度，不包含指令编号和校验	两个字节，参见指令分配规则	指令内容，内容为 UTF-8 编码的字符串	一个字节，把 0 与“长度、指令、编号、数据”中的每个字节异或

协议长度：（v1 协议总长度=6+数据长度）（v2 协议总长度=7+数据长度）

核验算法：（把 0 与“长度、指令编号、数据”中的每个字节异或再相乘）

```
private boolean check(byte fcs, byte[] data) {
    int xorResult = 0;
    for (byte b : data) {
        xorResult ^= b;
    }
    return fcs == xorResult;
}
```

4 指令分配规则

指令编号	发送方	接收方	指令类型
0x0001	客户端	网关	心跳请求
0x0002	网关	客户端	心跳响应
0x0003	客户端	网关	发送控制指令
0x0004	网关	客户端	响应控制指令，设备状态有更新
0x0005	客户端	网关	请求设备列表
0x0006	网关	客户端	返回设备列表，设备列表有更新
0x0007	客户端	网关	请求场景列表
0x0008	网关	客户端	返回场景列表
0x0009	客户端	网关	请求房间列表（可选）
0x000A	网关	客户端	返回房间列表（可选）
0x000B	客户端	网关	请求楼层列表（可选）
0x000C	网关	客户端	返回楼层列表（可选）
0x000D	客户端	网关	（预留）
0x000E	网关	客户端	让背景音主机主动说话（可选）
0x000F	网关	客户端	通知屏端重新获取设备场景数据

注：客户端是指背景音主机

5 各指令发送的数据

5.1 心跳请求

```
{  
  "type":"REQUEST_HEART_BEAT",  
  "sn":"xxxxx"  
}
```

5.2 心跳响应

```
{  
  "type":"RESPONSE_HEART_BEAT"  
}
```

5.3 请求设备列表

```
{  
  "type":"REQUEST_DEVICE ",  
  "sn":"xxxxx"  
}
```

5.4 返回设备列表，设备列表有更新

```
{
  "type": "RESPONSE_DEVICE"
  "deviceData": [
    //具体返回的数据
    {
      "deviceId": "设备ID",
      "deviceName": "设备名",
      "deviceType": "设备类型",
      "roomId": "房间ID（可选）",
      "floorId": "楼层ID（可选）",
      "state": {
        "order": "设备命令 开，关，暂停（可选）",
        "mode": "设备模式（可选）",
        "temperature": "温度（可选）",
        "fanSpeed": "风速（可选）",
        "fanDirection": "风向（可选）",
        "brightness": "亮度 0-100（可选）",
        "position": "进度 0-100（可选）",
        "angle": "角度 0-180",
        "sensorStatus": "1:有人,触发报警 0:无人，不触发",
        "battery": "电量",
        "humidity": "湿度",
        "universalSensorData": "万能传感器数据"
      }
    },
    {
      "deviceId": "设备ID",
      "deviceName": "设备名",
      "deviceType": "设备类型",
      "roomId": "房间ID（可选）",
      "floorId": "楼层ID（可选）",
      "state": {
        "order": "设备命令 开，关，暂停（可选）",
        "mode": "设备模式（可选）",
        "temperature": "温度（可选）",
        "fanSpeed": "风速（可选）",
        "fanDirection": "风向（可选）",

```

```

        "brightness": "亮度 0-100（可选）",
        "colorTemp": "色温值 0-100（可选）",
        "position": "进度 0-100（可选）",
        "angle": "角度 0-180",
        "sensorStatus": "1:有人,触发报警 0:无人,不触发",
        "battery": "电量",
        "humidity": "湿度",
        "universalSensorData": "万能传感器数据(jsonArray.toJSONString())"
    }
}
]

```

universalSensorData 说明

万能传感器 type=MULTIFUNCTIONAL_SENSOR

universalSensorData=jsonArray.toJSONString(); 具体传感器数据，该值是 jsonArray 形式，可以按以下方式传

```

JSONArray jsonArray = new JSONArray();
jsonArray.add(JSONObject.parse("{\"key\":\"key1xxx\",\"value\":\"value1xxx\"}"));
jsonArray.add(JSONObject.parse("{\"key\":\"key2xxx\",\"value\":\"value2xxx\"}"));
jsonArray.add(JSONObject.parse("{\"key\":\"key3xxx\",\"value\":\"value3xxx\"}"));
jsonArray.add(JSONObject.parse("{\"key\":\"key4xxx\",\"value\":\"value4xxx\"}"));

```

可以自行设置想要显示的，key 表示名称，比如

```

jsonArray.add(JSONObject.parse("{\"key\":\"照度值\",\"value\":\"50Lux\"}"));

```

5.5 请求场景列表

```
{
  "type":"REQUEST_SCENE",
  "sn":"xxxxx"
}
```

5.6 返回场景列表

```
{
  "type":"RESPONSE_SCENE",
  "data":[
    // 具体返回的数据
    {
      "id":"场景ID",
      "name":"场景名"
    },
    {
      "id":"场景ID",
      "name":"场景名"
    }
  ]
}
```

5.7 请求房间列表（可选）

```
{
  "type":"REQUEST_ROOM",
  "sn":"xxxxx"
}
```

5.8 返回房间列表（可选）

```
{
  "type":"RESPONSE_ROOM",
  "data":[
    //具体返回的数据
    {
      "id":"房间ID",
      "name":"房间名"
    },
    {
      "id":"房间ID",
      "name":"房间名"
    }
  ]
}
```

5.9 请求楼层列表（可选）

```
{
  "type":"REQUEST_FLOOR",
  "sn":"xxxxx"
}
```

5.10 返回楼层列表（可选）

```
{
  "type":"RESPONSE_FLOOR",
  "data":[
    //具体返回的数据
    {
      "id":"楼层ID",
      "name":"楼层名"
    },
    {
      "id":"楼层ID",
      "name":"楼层名"
    }
  ]
}
```

5.11 发送控制指令

```
{
  "type":"REQUEST_CONTROL",
  "sn":"xxxxx",
  "original":"原语",
  "data":[
    {
      //具体发送的控制指令
      "id":"设备ID",
      "name":"设备名",
      "scene":"场景",
      "state":"状态",
      "action":"动作",
      "attribute":"属性",
      "attributeValue":"属性值",
      "mode":"模式",
      "floor":"楼层",
      "room":"房间"
    },
    {
      //具体发送的控制指令
      "id":"设备ID",
      "name":"设备名",
      "scene":"场景",
      "state":"状态",
      "action":"动作",
      "attribute":"属性",
      "attributeValue":"属性值",
      "mode":"模式",
      "floor":"楼层",
      "room":"房间"
    }
  ]
}
```

5.12 响应控制指令，或设备状态有更新

```
{
  "type": "RESPONSE_CONTROL",
  //返回码
  "code":0,
  "msg":"返回信息",
  //为空则默认回答【好的】
  "msgVoice":"需要背景音主机说的话",
  "deviceId":"设备ID",
  "deviceName":"设备名称",
  "deviceType":"设备类型",
  "deviceState":{
    //具体返回设备状态数据
    "order":"设备命令 开，关，暂停",
    "mode":"设备模式",
    "temperature":"温度",
    "fanSpeed":"风速",
    "fanDirection":"风向",
    "brightness":"亮度 0-100",
    "position":"进度 0-100 ",
    "angle":"角度 0-180",
    "sensorStatus":"1:有人，触发 0:无人，不触发",
    "battery":"电量",
    "humidity":"湿度",
    "universalSensorData":"万能传感(jsonArray.toJSONString())"
  }
}
```

备注：5.4 获取设备列表和 5.12 状态刷新中的 order 字段为空时，会默认设为 STATE_OFF|关

5.13 让背景音主机，主动说话（可选）

```
{
  "type":"RESPONSE_VOICE",
  //需要背景音主机主动说的话
  "msgVoice":"需要背景音主机主动说的话"
}
```

6 返回状态码

状态码	表达的意思
0	控制成功
1	不存在此设备
2	设备不支持此操作
3	控制传入的参数有误
4	设备离线
5	设备控制超时
6	其他

7 常见设备类型

设备名	英文类型
灯	LIGHT
调光灯	DIMMING_LAMP
插座	SOCKET
窗帘	CURTAIN
可调进度窗帘	ADJUST_CURTAIN
风扇	ELECTRIC_FAN
空调	AIR_CONDITION
电视	TV
温控器	THERMOSTAT
投影仪	PROJECT
梦幻帘	CURTAIN_ANGLE
门窗磁	DOOR_MAGNETIC_SENSOR
水浸	DOOR_WATER_SENSOR
人体红外	INFRARED_SENSOR
烟雾	SMOKER_SENSOR
SOS	SOS_SENSOR
万能传感器	CURTAIN_ANGLE
温湿度显示	TEMP_HUMIDITY_SENSOR
新风	FRESH_AIR
地暖	FLOOR_HEATING
开关	SWITCH
色温灯	CW_LAMP

8 控制指令

```
{
  "id": "设备ID",
  "name": "设备名",
  "scene": "场景",
  "state": "状态",
  "action": "动作",
  "attribute": "属性",
  "attributeValue": "属性值",
  "mode": "模式",
  "floor": "楼层",
  "room": "房间"
}
```

梦幻帘（带角度调节）：

控制下发信息：state= STATE_ON（开启，进度 100），state=STATE_OFF（关闭，进度 0），state=STATE_STOP（暂停，进度 50）

进度：action=ACTION_TO，attribute=ATTRIBUTE_PROGRESS，attributeValue=具体进度值

角度：action=ACTION_TO，attribute=ATTRIBUTE_ANGLE，attributeValue=具体角度值

（说明：|号左边第一个英文或数字是 json 中的数据值，|号剩下的是英文或数字表示的意思）

- 状态
 - STATE_ON|开
 - STATE_OFF|关
 - STATE_STOP|停

● 动作

ACTION_TO|调到|调至|调为|调成|设为|设成|设至|变成|变为|改为|改成|切换|换到|设置|设置为

ACTION_ADD|调亮|调高|调暖|调热|调大|增大|加大|增加|亮点|亮一点|高点|高一点|下一|升高|大点|大一点|加|提高||拉上|拉开|拉起来|拉上去

ACTION_REDUCE|调暗|调低|调冷|调小|减小|减少|暗点|暗一点|低点|低一点|上一|降低|小点|小一点|减|拉下|拉下来

● 模式

MODE_COOL|制冷

MODE_HEAT|制热

MODE_FAN|送风

MODE_DRY|抽湿|干燥

MODE_AUTO|自动

MODE_WIND_HIGH|高风

MODE_WIND_MID|中风

MODE_WIND_LOW|低风

MODE_WIND_AUTO|自动风

MODE_WIND_MUTE|静音风

MODE_PLAY|播放

MODE_PAUSE|暂停

MODE_MUTE|静音

MODE_NEXT|下一首|空调模式切换

MODE_PREVIOUS|上一首

MODE_SWING_ANGLE|摇头|摆风

MODE_SWING_NO|停止摆风|停止摇头

MODE_SWING_LEFT_RIGHT|左右摆风

MODE_SWING_UP_DOWN|上下摆风

MODE_PURIFIER|空气净化

● 属性

ATTRIBUTE_MODE|模式

ATTRIBUTE_WIND_SPEED|风速

ATTRIBUTE_WIND_DIRECTION|风向
ATTRIBUTE_TEMPERATURE|温度
ATTRIBUTE_HUMIDITY|湿度
ATTRIBUTE_GEAR|档位
ATTRIBUTE_BRIGHTNESS|亮度
ATTRIBUTE_COLOR|颜色
ATTRIBUTE_COLOR_TEMP|色温
ATTRIBUTE_VOLUME|音量|声音
ATTRIBUTE_CHANNEL|频道|节目|台
ATTRIBUTE_HOUR|小时
ATTRIBUTE_MINUTE|分钟
ATTRIBUTE_SECOND|秒钟
ATTRIBUTE_PROGRESS|进度
ATTRIBUTE_ANGLE|角度

● 属性值

VALUE_COLOR_WHITE|白色
VALUE_COLOR_RED|红色
VALUE_COLOR_ORANGE|橙色
VALUE_COLOR_YELLOW|黄色
VALUE_COLOR_GREEN|绿色
VALUE_COLOR_CYAN|青色
VALUE_COLOR_BLUE|蓝色
VALUE_COLOR_PURPLE|紫色
VALUE_COLOR_PINK|粉红|粉色|粉红色
VALUE_MAX|最亮|最大|最高|最暖
VALUE_MIN|最暗|最小|最低|最冷
VALUE_LITTLE|一点|一点点
0.5|半|一半
1.5|一个半
2.5|两个半|二个半
3.5|三个半
4.5|四个半