

condition

halo协议新增condition命令介绍

Author 朱彤

E-mail nuaanjiziy@163.com

目录

- [condition](#)
- [协议格式](#)
- [注册时机](#)
- [内容介绍](#)

condition

`condition`用于halo客户端向服务器注册智能家居自定义协议. 目前只支持自定义RS485种类协议. 注册过程中要求客户端发送触发对应协议的目标, 条件和协议内容.

协议格式

```
{
  "cmd": "getconditions",
  "params": [
    {
      "target": {
        "id": "1(目标id)",
        "class": "DEVICE(目标类型)"
      },
      "condition": {
        "action": "ACTION_OPEN",
        "attribute": "",
        "attributeValue": ""
      },
      "payload": {
        "type": "RS485",
        "body": "aabbccdd"
      }
    },
    {
      "target": {
        "id": "20",
        "name": "照明1",
        "class": "DEVICE"
      },
```

```
        "condition": {
            "action": "ACTION_CLOSE",
            "attribute": "",
            "attributeValue": ""
        },
        "payload": {
            "type": "RS485",
            "body": "aabbccdd"
        }
    },
    {
        "target": {
            "id": "3",
            "class": "SCENE"
        },
        "condition": {
            "action": "ACTION_OPEN",
            "attribute": "",
            "attributeValue": ""
        },
        "payload": {
            "type": "RS485",
            "body": "aabbccdd"
        }
    }
]
}
```

注册时机

当客户端与服务器连接成功后, 客户端在发送完毕所有智能家居信息命令后主动发送`getcondition`. 如果成功, 服务端会回复`getcondition_cb`成功.

内容介绍

`condition`主要分为三个部分:`target`, `condition`和`payload`.

- `target` 表示受控目标, 与halo原始的设备列表和情景列表相似. 可以借用设备列表或者情景列表中已有的对象, 也可以在此新建新的对象, 新建的对象会被写入在服务端的智能家居列表中. 其中`id`字段表示目标id, 和设备列表或场景列表的id意义相同; `class`表示目标类型, 分别支持`DEVICE`和`SCENE`两种, 表示目标id对应的是设备类型还是场景类型:

引用已有的目标

```
"target": {
    "id": "3",
    "class": "SCENE"
}
"target": {
```

```

    "id": "1",
    "class": "DEVICE"
  },

```

新建目标

新建目标支持原halo协议中设备列表和情景列表的全部字段, 增加了class用于区分 设备/情景 .

```

"target": {
  "id": "20",
  "name": "照明1",
  "class": "DEVICE",
  "state": "设备状态(可选)",
  "type": "设备类型(可选)",
  "roomId": "房间id(可选)",
  "floorId": "楼层id(可选)"
},

```

- **condition** 表示触发自定义协议的条件, 整体可选. 条件格式参考halo协议中智能家居相关附录. 场景对象只支持**action**条件. 当没有**condition**或者**condition**内容非法(例如场景的条件中没有**action**而是**attribute**等)时, 默认使用**action: ACTION_OPEN** 作为条件:

```

"condition": {
  "action": "ACTION_OPEN(打开动作)",
},
"condition": {
  "action": "ACTION_TO(设置属性值)",
  "attribute": "ATTRIBUTE_TEMPERATURE(温度)",
  "attributeValue": "23"
},

```

- **payload** 字段为自定义协议的协议内容, 其中**type**表示协议类型, 目前只支持**RS485**; **body**表示协议字段内容, RS485默认使用**hex**转义后的字符串格式, 比如: 如果要发送一段16进制的hex指令 **0xaa 0x02 0x02 0x03 0xb0**, 则需要在**body**字段中填入**aa020203b0**. 长度必须为偶数:

```

"payload": {
  "type": "RS485",
  "body": "aa020203b0"
}

```