网关功能接口协议

# 协议范围

智能家居系统中网关功能划分为两个部分，设备服务以及业务逻辑。两部分为独立的应用，两者间通过socket or mqtt通信，网关功能接口协议（gateway function interface protocol，简称GFIP），为设备服务层与业务逻辑层之间的通信协议，协议基于json格式进行定义。

协议功能总览

·协调器信息查询（信道，网络ID，）

·设备基本信息查询（节点类型，供电类型，设备类型，厂商ID，mac地址，支持属性命令集等）

·设备属性状态查询（网络地址，开关，cie颜色值，亮度，锁状态等等）

·设备入网上报（包含全部或部分基本信息）

·状态变化上报

·设备命令接口（控制，配置绑定，zigbee分组等）

# 约定

**大小写**

处理json消息时忽略大小写，为了阅读方便，应用名词、服务名词、属性名词、命令名词的所有单词首字母均大写。

**名称格式**

应用名词、服务名词、服务属性名词和命令名称分别基于zigbee规范中的设备名称、cluster名称、cluster属性名称和命令名词，去空白符后获得，同时将单词首字符大写。

# 协议格式

通信基本单位为消息，消息含多种类型及对应响应消息，有读属性、写属性、控制及上报四种消息及对应的响应消息类型。

## 属性表示方法

网关提供了设备属性的访问接口，属性ID使用字符串表示，每个ID字符串会有多个部分组成，每部分使用中划线连接。属性分为节点属性（硬件设备相关）、应用属性（应用端点相关）及服务属性（cluster相关）。

### 节点属性

节点属性指的是节点相关的属性，每个硬件设备有一组相关的属性，如厂商ID、zigbee网络地址、在线状态和应用（端点）列表等，所有可用的节点属性见TBD..。节点属性的字符串由两个部分组成，每个部分使用中划线连接，格式如下。

**<设备ID>-<属性名称>**

设备ID表示目标设备的ID，可能是个mac地址或其它唯一标识。属性名词为节点的属性名称，如在线状态表示如下：

*"device001-OnlineState"*

### 应用属性

应用属性是应用相关，也就是端点相关的属性，如服务（cluster）列表、应用profile（HA、SE、HC等），所有可用的应用属性见TBD..。应用属性的表示字符串由三个部分组成，格式如下

**<设备ID>-<应用名称>-<属性名称>**

设备ID表示目标设备的ID，可能是个mac地址或其它唯一标识。应用名称（对应endpoint端点）表示需要读的目标应用端点，如两路灯的应用名称就不一样。如访问服务列表可使用如下表示。

*"device001-On/OffLight0-ServiceList"*

### 服务属性

服务属性的字符串由四个部分组成，每个部分使用中划线连接，格式如下。

**<设备ID>-<应用名称>-<服务名称>-<属性名称>**

设备ID表示目标设备的ID，可能是个mac地址或其它唯一标识。应用名称（对应endpoint端点）表示需要读的目标应用端点，如两路灯的应用名称就不一样。服务名称（对应cluster名称）表示需要操作的属性或命令所属集合。如锁状态及调光灯亮度属性的例子：

*"device001-DoorLock0-Doorlock-LockState**"*

*"device002-DimmableLight0- LevelControl-CurrentLevel "*

## 值表示方法

协议范围使用的值分三种类型，bool、枚举和数值，bool类型用字符串true或false表示，枚举使用枚举名词字符串表示，数值类型使用数字，包含浮点数。适用于读写属性或命令参数。如：

bool：*"true"、"false"*

枚举*："Battery"、"DCsource"*

数值*：80、0.33.*

## 命令表示方法

### 系统命令

格式

**system-<命令名称>**

可用命令列表见

### 应用端点命令

### 服务命令

## 消息格式

协议基于json格式，基本字段（即json格式中的键）为type、msgid、msgctl，其他字段根据type不同而不同，debugmsg为可选字段。字段内容指键对应的值。

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| type | 字符串 | 表示消息类型。 |
| msgid | 字符串 | 消息ID。 |
| msgctl | 字符串 | 消息控制。 |
| debugmsg | 字符串 | 调试用途 |
| ~~devs~~ | ~~json对象数组~~ | ~~操作的设备及内容。~~ |
| readattrs | 字符数组 | 读取的目标属性列表 |
| raedattrsrsp | json对象数组 | 返回读取的目标属性值列表 |
| writeattrs | json对象数组 | 写目标属性列表 |
| writeattrsrsp | json对象数组 | 目标属性列表写结果 |
| control | json对象数组 | 控制目标列表 |
| controlrsp | json对象数组 | 目标控制结果列表 |

### type字段

表示消息类型，有效值如下表。

|  |  |
| --- | --- |
| 字段内容 | 含义 |
| read | 读属性消息 |
| write | 写属性消息 |
| control | 控制消息 |
| report | 上报消息，新增设备上报，属性上报 |
| readrsp | 读属性响应消息 |
| writersp | 写属性响应消息 |
| controlrsp | 控制响应消息 |

### msgid字段

消息的ID，范围1~65535。发送者每发送一个新的消息将ID增加1。响应方中响应时该字段内容使用发送方的值。

### msgctl字段

消息的配置控制，是否需要响应、rsp=false。

### ~~devs字段(弃用)~~

devs消息表示需要操作的设备列表，支持多个设备、多个应用（对应zigbee端点）、多个服务（对应zigbee cluster）。类型为json对象数组，每个元素dev（对应一个物理设备，注意dev这个关键字在不会出现在json消息中）表示需要读的目标设备。

dev包含两个字段，devid与apps。devid表示目标设备的ID，可能是个mac地址或其它唯一标识，设备ID会在设备入网后由网关提供给上层。apps字段类型为json对象数组，每个元素app（对应zigbee应用端点，注意这个app在不会出现在json消息中）表示需要读的目标应用端点。

app包含两个字段，appid与services。appid表示目标端点的ID，如“OnOff Light0”、“OnOff Light1”。services字段类型为json对象数组，每个元素service（对应zigbee cluster，注意这个service在不会出现在json消息中）表示需要读的目标cluster（簇）。

service包含两个字段serviceid和“其他”。serviceid表示目标cluster的ID，这里使用cluster 的名字，如“DoorLock”。“其他”详见具体的消息类型。

"devs": [

{

"devid": "device001",

"apps": [

{

"appid": "DoorLock0",

"services": [

{

"serviceid": "Doorlock",

"其他字段":"具体看对应类型消息"

}

]

}

]

}

### readattrs字段

readattrs（read attributes）为读取的目标属性列表，出现在read消息中，类型为字符串数组，元素为属性，属性表示方法见[属性表示方法](#_属性表示方法_1)。如读取门锁device001的门状态及剩余电量使用以下内容：

"readattrs": [

"device001-DoorLock0-Doorlock-LockState",

"device001-DoorLock0-Doorlock-DoorState",

"device001-DoorLock0-PowerConfiguration-BatteryPercentageRemaining"

]

读取两路灯device001的开关属性及调光灯device002的亮度属性：

"readattrs": [

"device001-OnOffLight0-On/Off-OnOff",

"device001-OnOffLight1-On/Off-OnOff ",

"device002-DimmableLight0- LevelControl-CurrentLevel "

]

### readattrsrsp 字段

readattrsrsp（read attributes response）表示读取的目标属性响应列表，出现在readrsp消息中，类型为json对象数组，每个元素含attr及value两个字段（键），attr表示属性，表示方法见[属性表示方法](#_属性表示方法_1)的描述，value表示值。值表示方法见[值表示方法](#_值表示方法_1)。如读取门锁锁状态、电源类型和调光灯亮度的返回：

"readatttsrsp": [

{

"attr": "device001-DoorLock0-Doorlock-LockState",

"value": "ture"

},

{

"attr": "device001-DoorLock0-Basic-PowerSource",

"value": " Battery"

},

{

"attr": "device002-DimmableLight0- LevelControl-CurrentLevel ",

"value": 120

}

]

### writeattrs字段

writeattrs（write attributes）表示写属性列表，出现在write消息中，类型为json对象数组，每个元素含attr及value两个字段（键），attr表示属性，表示方法见[属性表示方法](#_属性表示方法_1)的描述，value表示值。值见[值表示方法](#_值表示方法_1)。如写门锁开锁事件数量属性为0：

"writeattrs": [

{

"attr": "device001-DoorLock0-Doorlock-DoorOpenEvents",

"value": 0

}

]

### writeattrsrsp字段

writeattrsrsp（write attributes response）表示写属性响应列表，出现在write消息中，类型为json对象数组，每个元素含attr、value及status三个字段（键），attr表示属性，表示方法见[属性表示方法](#_属性表示方法_1)的描述，value表示值，见[值表示方法](#_值表示方法_1)。status表示写操作执行状态，类型为字符串，详见[执行返回码](#_执行返回码)。

"writeattrsrsp": [

{

"attr": "device001-DoorLock0-Doorlock-DoorOpenEvents",

"value": 0,

"status": "success or failed"

}

]

### control 字段

control 为控制命令的内容，出现在control消息中，类型为json对象数组。每个元素含command

"control": [

{

"command": "device001-DoorLock0-Doorlock-UnlockWithTimeout",

"params": "Timeout=5,PIN/RFID=123456"

},

{

"command": "system-PermitJoin",

"注释":"系统命令，允许入网",

"params": "Timeout=180"

}

]

### controlrsp字段

"controlrsp": [

{

"command": "device001-DoorLock0-Doorlock-UnlockWithTimeout",

"params": "Timeout=5,PIN/RFID=123456",

"status": "success or failed"

}

]

### report字段

## read消息

## write消息

## 执行返回码

|  |  |
| --- | --- |
| 值 | 含义 |
| success | 成功 |
| failed | 失败 |
| attr- inexistence | 属性不存在 |
| non-writable | 不可写 |
| params-error | 参数错误 |
|  |  |
|  |  |

# 属性集

## 节点属性

节点属性都为只读。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 数据类型 | 访问方式 | 功能描述 |
| *OnlineState* | bool | Read | 在线状态 |
| manufactureID | int | Read | 厂商id |
| appsList | 字符数组 | Read | 设备的应用端点名称列表，返回结果如[“On/OffSwitch0”, “On/OffSwitch1”, …] |
| nwkAddr | 网络地址 | Read | 设备网络地址（调试引用） |

## 应用属性

应用属性都为只读。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 数据类型 | 访问方式 | 功能描述 |
| AppProfile | String | Read | 应用规范，如HA、SE、HC等，目前仅支持HA |
| ServiceList | 字符串数组 | Read | 设备的服务名称列表，返回结果如[“Basic”, ”On/Off”, “DoorLock”, …] |

## 服务属性

服务属性有下表的公共属性，针对每个服务支持的属性及命令见《zigbee产品设计参考手册》“簇规范”一章。通信协议中不分服务（cluster）的server side和client side，server side和client side支持接收的命令和实现的属性统一收集到命令和属性列表中。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 数据类型 | 访问方式 | 功能描述 |
| AttrsList | 字符串数组 | Read | 表示当前服务支持的属性列表，返回结果如[“LookState”, “DoorState”, …] |
| CommandList | 字符串数组 | Read | 表示当前服务支持接收的命令列表，返回结果如[“Unlock”, “UnlockWithTimeout”, …] |

# 命令集

## 系统命令

格式：

**system-<命令名称>**

|  |  |  |
| --- | --- | --- |
| 命令名称 | 参数 | 功能描述 |
| permitJoin | Timeout：数字：0~600 | 允许设备入网，0关闭，非0表示允许入网时间，单位秒 |
| DeleteDevice | DevName -字符串数组 - 元素为设备ID | 删除指定设备 |

# 新的方式

{

"type": "write or read or control or rsp",

"msgid": 1,

"readattrs": [

"device001-DoorLock0-Doorlock-LockState",

"device001-DoorLock0-Doorlock-DoorState",

"device001-DoorLock0-PowerConfiguration-BatteryPercentageRemaining",

*"device001-On/OffLight0-ServiceList"*

],

"readatttsrsp": [

{

"attr": "device001-DoorLock0-Doorlock-LockState",

"value": "ture"

},

{

"attr": "device001-DoorLock0-Doorlock-DoorState",

"value": "ture"

}

],

"writeattrs": [

{

"attr": "device001-DoorLock0-Doorlock-DoorOpenEvents",

"value": 0

}

],

"writeattrsrsp": [

{

"attr": "device001-DoorLock0-Doorlock-DoorOpenEvents",

"value": 0,

"status": "success or failed"

}

],

"control": [

{

"command": "device001-DoorLock0-Doorlock-UnlockWithTimeout",

"params": "Timeout=5,PIN/RFID=123456"

}

],

"controlrsp": [

{

"command": "device001-DoorLock0-Doorlock-UnlockWithTimeout",

"params": "Timeout=5,PIN/RFID=123456",

"status": "success or failed"

}

]

}

# 多级数组方式（弃用）

read消息用于读取设备的属性，一个消息支持读取多个设备、多个应用（对应zigbee端点）、多个服务（对应zigbee cluster）、多个属性的值.

下面为读取属性的示例，目标设备有两个device001和device002。device001中读取了门锁应用中的门状态、锁状态、剩余电量三个信息和联防应用中的联防状态信息。device002中读取了彩色灯应用的开关状态、亮度、亮度调节剩余时间等信息。可将下面示例复制粘贴到json在线编辑工具（如http://www.bejson.com/jsoneditoronline/）中，解析json便于阅读。

{

"type": "read",

"msgid": 1,

"devs": [

{

"devid": "device001",

"apps": [

{

"appid": "DoorLock0",

"services": [

{

"serviceid": "Doorlock",

"attrs": [

"LockState",

"DoorState"

]

},

{

"serviceid": "PowerConfiguration",

"attrs": [

"BatteryPercentageRemaining"

]

}

]

},

{

"appid": "IASZone1",

"services": [

{

"serviceid": "IASZone",

"attrs": [

"ZoneState",

"ZoneStatus"

]

}

]

}

]

},

{

"devid": "device002",

"apps": [

{

"appid": "ColorLight0",

"services": [

{

"serviceid": "LevelControl",

"attrs": [

"CurrentLevel",

"RemainingTime"

]

},

{

"serviceid": "OnOff",

"attrs": [

"OnOff"

]

}

]

}

]

}

]

}