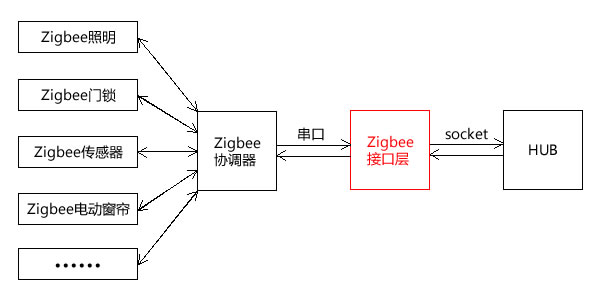
Zigbee接口层设计需求说明书 V1.0

# 概述

在智能网关中，hub是核心程序，负责信息的中转传输、设备和任务管理等。Zigbee接口层负责管理与控制zigbee设备，通过hub提供的标准接口来管理和控制底层的智能设备。与hub通信采用基于[HIPC](http://wiki.whatsmart.org/hub:hipc)协议的jsonrpc，与zigbee协调器通过串口进行通信。



Hub所提供的设备管理控制等接口对于zigbee接口层、Wifi接口层和蓝牙接口层是统一的，而这些接口层底层的设备的接口是不同的，接口层的作用就是实现统一的上层接口来对底层设备进行管理和控制。对hub和zigbee协调器的接口，他们的数据传输协议是不同的，但是其数据都为jsonrpc格式，所调用的方法是一样的，方法的参数和返回值也是相同的。这些方法都是标准的。

# 消息流

在智能网关设备中，hub是核心程序，其他所有的组件、服务等均通过unix socket连接到hub上，进程间通信地址为“/tmp/hub\_sock”。Hub程序采用python的异步io框架[asyncio](https://docs.python.org/3.4/library/asyncio.html)进行开发。Hub视所有连接到自身的程序为组件，在hub维护这些组件。当一个程序通过上述地址连接到hub时，hub会自动新建一个组件，并将该组件添加到列表中。当程序成功连接到hub后，应立刻发送自己的描述信息到hub（参考[component](http://wiki.whatsmart.org/hub:interface:component)），hub收到回更新该组件的名称和类型信息。

由于上述的系统结构，消息发出到真正的处理可能需要经过多个接口。例如，手机app发出关灯消息，消息首先会通过http发送到web接口程序中，但web接口不会处理该消息，web接口需要调用hub提供的统一的接口，因此web接口要封装该消息并发送给hub。Hub收到消息后，解析得知要进行关灯操作，通过设备id查找到该设备有zigbee接口层进行管理，因此它会将该消息发送给zigbee接口程序，zigbee接口程序执行具体的操作后，将操作结果返回给hub，hub收到结果后，再将其发送给web接口，web接口在返回给用户。

在消息的传输过程中，若消息类型为请求，则hub会根据消息所请求的资源能够正确的将消息发送到应该送达的地方。但若消息类型为响应，例如，灯光控制器操作成功，将结果返回到hub时，hub并不知道将该结果发送给谁，因为连接到hub的不仅仅有web接口，原始的请求关灯消息不一定有web接口发出，为了能够正确的将响应返回给请求者，需要使用HIPC协议消息头中的origin字段，该字段代表请求者是谁。

假设有两个用户A和B，他们在web接口中的id分别为1和2。当用户A发送关灯消息给web接口，web接口将消息发送给hub时，可设置origin字段值为“user/1”。对于hub来讲，web接口是一个组件，在其内部记录着该组件的id，当hub接收到web接口发来的消息时，解析得到为关灯，自己并不能处理，所将该消息发送给zigbee接口层，在发送之前会修改origin的值，在当前的值得前面加上组件的id信息，修改之后的origin值为“component/1/user/1”。当zigbee接口层返回结果时，其响应消息的消息头中的origin应该和收到的请求的消息头中的origin值一致，也应该为“component/1/user/1”，当hub收到该消息后，根据component id可得知应该将该响应发送给web接口，在发送给web接口之前，会重新修改origin为“user/1”，在web接口收到消息后能够根据user id正确的将结果发送用户A，而不是用户B。

# Hub接口

在HIPC的请求头中的请求行格式是“HIPC/1.0 request resource”，resource为资源地址。在zigbee接口层中，程序需根据resource来决定处理或转发消息。例如当resource为“device”时，说明是设备管理接口，应该有程序自身执行请求中的方法。当resource为“control/id”时，说明要对设备进行控制，接口程序应该根据id值来查找到该设备的网络地址，然后构建新的消息发送到该设备。

## 组件管理接口

组件是指通过HIPC连接到hub的所有程序。组件管理接口包含的方法有：register\_component。

### register\_component方法

请求方向：zigbee接口层 --> hub

该方法为注册组件信息到hub。资源为“component”。参数为name和type。Name是该组件的名称，type是该组件的类型。请求格式为：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "register\_component"  "params"**:** **{**  "name"**:** "zigbee\_link\_adapter"**,**  "type"**:** "link\_adapter:zigbee\_link\_adapter"**,**  **}**  "id"**:** 1  **}** |

若请求执行成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 00000001**,**  "message"**:** "register\_component error"  **}**  "id"**:** 1  **}** |

## 设备管理接口

接口资源地址为“device”。

### discover\_device方法

请求方向：hub --> zigbee接口层

zigbee接口层中，discover\_device方法为打开zigbee协调器的可见性，该方法是用户添加新设备时执行，用来开启可见性，以便于设备能够搜索到这些网络。当用户点击app上的添加设备图标时，执行此方法。请求格式为：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "discover\_devices"**,**  "id"**:** 1  **}** |

若成功，说明网关打开了zigbee协调器的可见性当成功时，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"  "id"**:** 23 |

若失败，则返回为错误：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000003,  "message"**:** "discover\_device error"  **}**  "id"**:** 1  **}** |

### add\_device方法

请求方向：zigbee接口层 → hub

向hub添加新设备，资源为“device”。参数为该设备。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "add\_device"**,**  "params"**:** **{**  "vender"**:** "Obama"**,**  "hwversion"**:** "2.3"**,**  "swversion"**:** "1.6"**,**  "type"**:** "lighting"**,**  "operations"**:** **[**"power\_on"**,** "power\_off"**,** "get\_brightness"**,** "set\_brightness"**],**  "status"**:** **{**  "power"**:** "on"**,**  "color"**:** 0x345423  "brightness"**:** 60  **}**  **}**  "id"**:** 1  **}** |

若成功，返回设备的id：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** 231**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000005**,**  "message"**:** "add\_device error"  **}**  "id"**:** 1  **}** |

### remove\_device方法

请求方向：zigbee接口层 → hub

从hub中删除一个设备。资源为“device/id”，id为添加设备成功时返回的id。参数为空。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "remove\_device"**,**  "id"**:** 1  **}** |

若成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000006**,**  "message"**:** "set\_position error"  **}**  "id"**:** 1  **}** |

## 设备控制接口

设备控制接口的资源为“control/id”，id为设备的id。

### 智能照明设备控制接口

下述方法为智能网关定义的标准方法，实际设备可能不支持部分方法，在使用过程中需要根据设备的operations来判断。

power\_on方法

请求方向：hub → zigbee接口层

该方法为打开设备电源，参数为空。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "power\_on"**,**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

power\_off方法

请求方向：hub → zigbee接口层

该方法为关闭设备电源，参数为空。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "power\_off"**,**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

get\_power\_state方法

请求方向：hub → zigbee接口层

该方法为获取设备电源打开状态，参数为空。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "get\_power\_state"**,**  "id"**:** 23  **}** |

若请求成功，返回照明设备的开关状态on或off：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "on|off"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

set\_brightness方法

请求方向：hub → zigbee接口层

该方法设置灯光亮度，参数0-100的整数值。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "set\_brightness"**,**  "params"**:** **{**  "brightness"**: 60**  **}**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

get\_brightness方法

请求方向：hub → zigbee接口层

该方法获取灯光亮度值，参数为空，返回为0-100的整数值。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "get\_brightness"**,**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

set\_color方法

请求方向：hub → zigbee接口层

该方法设置灯光颜色，参数整数代表的颜色值，如0xff0000为红色。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "set\_color"**,**  "params"**:** **{**  "color"**: 0xff0000**  **}**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

get\_color方法

请求方向：hub → zigbee接口层

该方法获取灯光颜色值，参数为空，返回为整数代表的颜色值。

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "get\_color"**,**  "id"**:** 23  **}** |

若请求成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** 0xff0000**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 20010000**,**  "message"**:** "control error"  **}**  "id"**:** 23  **}** |

## 事件接口

事件接口用来事件系统内事件的广播。例如hub需要重启，则向所有组件发送重启事件，如设备接口层发现新设备，则向hub报告新设备发现事件。Hub将新设备事件发送到所有其他组件，若某个组件关心该事件，它应该事先该类型事件的处理。此接口中包括的方法有：send\_event。

### send\_event方法

请求方向：hub ←→ zigbee接口层

该方法用来在hub和组件之间传递事件。其请求格式为：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "send\_event"**,**  "params"**:** **{**  "name"**:** "brightness\_changed"**,**  "data"**:** **{**  "id"**:** 123**,**  "brightness"**:** 50  **}**  **}**  "id"**:** 23  **}** |

name为事件名称，data是与事件相关的数据。在灯光亮度发生变化时，data中id为该设备的id，brightness为新的亮度。若成功，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若失败，则返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 40000001**,**  "message"**:** "event error"  **}**  "id"**:** 23  **}** |

# Zigbee协调器接口

## 通信协议

串口通信由消息来完成，一条消息包含起始符，消息头，消息体，校验和和结束符。起始符合结束符恒为0x7e；消息头由address、size、type、serial构成，address是设备的网络地址，当消息是由接口层发送给协调器时，address代表目的设备，当消息是由协调器发送给接口层时，address代表数据来源地址，size是消息体的字节数，type是消息类型，type=0说明消息为请求，type=1，说明消息为应答，serial是消息流水号，当消息为请求时，serial代表该消息的id，当type为应答时，serial代表应答的消息id，请求消息的流水号依次在上一条请求的流水号上+1。body是消息体，消息体为[jsonrpc](http://www.jsonrpc.org/specification)的请求、响应、错误对象或为空。checksum是消息体的crc8位校验。每条请求消息必须有应答，表示消息的接收成功。在整个消息中，数据以大端格式对齐。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Start** | **Address** | **Size** | **Type** | **Serial** | **Body** | **Checksum** | **End** |
| 1byte | 2bytes | 2bytes | 1byte | 2bytes | n bytes | 1byte | 1byte |

起始符合结束符为0x7e，因此，除了起始符和结束符之外，消息中不可再出现0x7e。所以在消息发送之前，要对消息进行转义处理，0x7e变成0x7d0x01，0x7d变成0x7d0x02。在消息接收后，需要对消息进行反转义处理，将0x7d0x01变成0x7e，0x7d0x02变成0x7d。

Body的示例：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "power\_on"**,**  "id"**:** 23  **}** |

在协调器收到消息后，根据address判断该消息是否发给自身，若是自身，则有自身处理该条消息。若address是网络中的其他设备，应该将消息转发到目的设备。

### 消息的处理

在协调器或其他设备处理消息时，应先取得jsonrpc请求对象，然后得到需要执行的函数以及对应的参数。执行完成后将结果返回。

### 消息的应答

在消息处理完成后需要将结果返回。应答消息的消息体格式为jsonrpc中定义的响应对象。示例：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 23  **}** |

若请求对象为通知（notification），则应答消息的消息体为空。

## 设备管理

### 添加设备

方法名：add\_device

消息方向：zigbee协调器→ zigbee接口层

当一个新设备加入到zigbee网络之后，协调器应获得设备的厂商、软硬件版本和设备类型等信息，然后向接口层发送该消息。注意，zigbee协调器本身也是zigbee设备，在收到接口层准备好命令后，首先要添加协调器本身，其支持的操作为“set\_visibility”，类型为“zigbee\_coordinator”，status属性包含“visivility”。

消息体格式：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "add\_device"**,**  "params"**:** **{**  "vender"**:** "Obama"**,**  "hwversion"**:** "2.3"**,**  "swversion"**:** "1.6"**,**  "type"**:** "lighting"**,**  "operations"**:** **[**"power\_on"**,** "power\_off"**,** "get\_brightness"**,** "set\_brightness"**],**  "status"**:** **{**  "power"**:** "on"**,**  "color"**:** 0x345423  "brightness"**:** 60  **}**  **}**  "id"**:** 1  **}** |

若成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000001**,**  "message"**:** "add\_device error"  **}**  "id"**:** 1  **}** |

### 移除设备

方法名：remove\_device

消息方向：zigbee协调器→ zigbee接口层

当一个设备在zigbee网络中消失（如一定时间未收到心跳、断电），协调器应该向接口层发送该命令。

消息体格式：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "remove\_device"**,**  "params"**:** **{**  "address"**:** 232  **}**  "id"**:** 1  **}** |

若成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000002**,**  "message"**:** "remove\_device error"  **}**  "id"**:** 1  **}** |

### 获取设备状态

方法名：get\_status

消息方向：zigbee接口层 → zigbee协调器

该命令用来获取设备的状态。

消息体格式：

消息体格式：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "get\_status"**,**  "params"**:** **{**  "address"**:** 232  **}**  "id"**:** 1  **}** |

若成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** **{**  "power"**:** "on"**,**  "brightness"**:** 60  **}**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000003**,**  "message"**:** "get\_status error"  **}**  "id"**:** 1  **}** |

## 设备控制

设备控制类似设备管理，消息的格式是一样的。协调器根据地址转发消息。设备收到消息后，根据rpc方法执行对应的操作。

### 智能灯光控制

#### 打开电源

方法名：power\_on

消息方向：zigbee接口层 → zigbee协调器

消息体格式：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "power\_on"**,**  "id"**:** 1  **}** |

若成功，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "result"**:** "success"**,**  "id"**:** 1  **}** |

若失败，返回：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "error"**:** **{**  "code"**:** 10000003**,**  "message"**:** "get\_status error"  **}**  "id"**:** 1  **}** |

关闭电源、获取电源状态、获取设置亮度和颜色类似。

## 事件报告

方法名：send\_event

消息方向：zigbee接口层 ←→ zigbee协调器

事件报告消息用来在网关和设备之间传送事件。

消息体格式：

|  |
| --- |
| **{**  "jsonrpc"**:** "2.0"**,**  "method"**:** "send\_event"**,**  "params"**:** **{**  "name"**:** "sys\_reboot"**,**  "data"**:** **{**  "time"**:** 5  **}**  **}**  **}** |

事件报告为通知类型（notification），不关心结果，收到消息后返回一个包含空消息体的应答消息，表示接收到消息。