



Android / IOS  

# SDK 开放接口

用  
户  
手  
册

[ 20140703 版本 ]

## 简介

BroadLink开放API可以让开发者直接将BroadLink的easyconfig技术，BroadLink现有产品的设备发现和控制集成到自己的应用中。

## SDK使用流程

### 1. 库初始化

在调用各个接口之前，首先使用network\_init命令对库进行初始化。

### 2. 配置设备上网

调用easyconfig命令，开始配置设备联网参数。

### 3. 查找设备

调用probe\_list命令后，可以搜索到所有局域网内的设备。

### 4. 对设备进行初始化

在probe\_list命令返回后，立刻调用device\_add对设备进行初始化，device\_add内的参数必须填写为probe\_list返回的内容。

### 5. 控制设备

调用不同的设备控制命令进行控制。

### 6. 设备信息保存

应用程序发现设备后，需要将设备信息（probe\_list返回的内容）保存到自己的数据库或者文件内。

APP每次重新启动调用network\_init命令后，必须重新从数据库或者文件内，将保存的设备信息通过device\_add命令传递给底层库。

## 接口说明

下表列出了所有请求，所有请求均为同步模式。

Request ID	Function	Comment	Command
1	网络线程的初始化	网络操作的前提，必须先执行该方法	network_init
2	获取SDK版本信息	获取当前SDK的版本信息与更新内容	SDK_version
11	查找新设备列表	通过该接口，可以查找到新设备列表，或者有内容更新的设备，用户需要自己判断该设备是否已经保存	probe_list
12	添加设备到网络线程	通过该接口，可以将设备加入网络线程中，后续对该设备进行的操作都以该步为前提	device_add
13	修改设备信息	通过该接口，可以更新设备的名称与锁定状态	device_update
14	将设备从网络线程中删除	通过该接口，可以将设备从网络线程中删除	device_delete
15	获取设备局域网IP地址	通过该接口，可以取得设备在本地网络中的IP地址	device_lan_ip
16	获取设备网络状态	通过该接口，可以取得设备的当前网络状态	device_state
41	SP1的登录	通过该接口，对SP1进行认证，只有认证过后的设备才能控制	sp1_auth
42	SP1的控制	通过该接口，可以对SP1进行开启/关闭操作	sp1_control
43	SP1的刷新	通过该接口，可以查看SP1当前的最新状态	sp1_refresh
44	SP1的任务	通过该接口，可以设置SP1的周期任务	sp1_task
71	SP2的刷新	通过该接口，可以直接获取SP2的当前状态，无需登录	sp2_refresh
72	SP2的控制	通过该接口，可以对SP2进行开启/关闭操作	sp2_control
73	SP2的任务	通过该接口，可以设置SP2的周期任务	sp2_task
74	SP2的实时功率	通过该接口，可以获取SP2的当前实时功率	sp2_current_power
75	SP2的最近24小时功率	通过该接口，可以获取SP2的最近24小时功率曲线	sp2_24_power
76	SP2的一周能耗统计	通过该接口，可以获取SP2的指定周的能耗统计	sp2_week_power
77	SP2的一月能耗统计	通过该接口，可以获取SP2的指定月份的能耗统计	sp2_month_power
78	SP2的一年能耗统计	通过该接口，可以获取SP2的指定年份的能耗统计	sp2_year_power
79	SP2获取待机功率	通过该接口，可以获取SP2的待机功率	sp2_get_standby_power
80	SP2设置待机功率	通过该接口，可以设置SP2的待机功率	sp2_set_standby_power
101	RM1的登录	通过该接口，对RM1进行认证，只有认证过后的设备才能控制	rm1_auth
102	RM1学习模式	通过该接口，让RM1进入学习模式	rm1_study
103	RM1查询code	通过该接口，获取RM1学习到的控制码	rm1_code
104	RM1发送code	通过该接口，让RM1发送指定的控制码	rm1_send
131	RM2的刷新	通过该接口，可以查看RM2所处环境的温度值	rm2_refresh
132	RM2学习模式	通过该接口，让RM2进入学习模式	rm2_study
133	RM2查询code	通过该接口，获取RM2学习到的控制码	rm2_code
134	RM2发送code	通过该接口，让RM2发送指定的控制码	rm2_send
161	A1的刷新	通过该接口，可以查看A1所处环境的温湿度、光照、空气质量、噪声信息	a1_refresh
162	A1的当前任务列表	通过该接口，可以获取到A1当前所有的任务列表	a1_task_list
163	A1添加任务	通过该接口，可以添加新的任务到A1中	a1_add_task
164	A1删除任务	通过该接口，可以从A1中删除指定的任务	a1_del_task
10000	easyconfig	通过该接口，可以将使用本公司模块的未联网设备配置到局域网中	easyconfig
10001	取消easyconfig	通过该接口，可以停止正在进行中的easyconfig	cancel_easyconfig

针对所有命令的返回数据中的code字段，0表示成功，非0表示失败。所有数据都以json格式传输

## 1.0 network\_init 请求参数说明

Arguments:

```
{
  "api_id"      : 1,
  "command"     : "network_init",
  "license"     : "000000000000000000000000"
}
```

api\_id: 命令ID。

command: 命令。(可选)

license: 用户申请api时获得的license。

Retval:

```
{
  "code":0,
  "msg":"network init success"
}
```

## 1.1 SDK\_version 请求参数说明

<pre>Arguments:

```
{
  "api_id" : 2,
  "command" : "SDK_version",
}
```

api\_id: 命令ID。

command: 命令。(可选)

</pre>

<pre>Retval:

```
{
  "code":0,
  "version":"V1.1.0-201406091100",
  "update":"V1.1.0-201406091100 update contents: ....."
}
```

&lt;/pre>

code:返回码

version:当前SDK的版本

update:当前SDK的更新内容

</pre>

## 1.2 probe\_list请求参数说明

```
<pre>Arguments:
{
  "api_id" : 11,
  "command" : "probe_list"
}
```

code: 返回码  
msg : 返回消息

```
</pre>
```

```
<pre>Retval:
```

```
{
  "code":0,
  "msg":"Execute success!",
  "list":[
    {
      "mac":"00:11:22:33:44:55",
      "type":"SP2",
      "name":"智能插座",
      "lock":0,
      "password":987961777,
      "id":0,
      "subdevice":0,
      "key":"097628343fe99e23765c1513accf8b02"
    },
    {
      "mac":"00:11:22:33:44:55",
      "type":"SP2",
      "name":"Example",
      "lock":0,
      "password":912023642,
      "id":0,
      "subdevice":0,
      "key":"097628343fe99e23765c1513accf8b02"
    }
  ]
}
```

code: 返回码

msg : 返回消息

list: 查询到的设备属性信息数组

mac: 设备的mac地址

type: 设备类型, 目前支持的设备类型, SP1/SP2/RM1/RM2 (注:必须大写) .

name: 设备名称, UTF8编码。

lock: 设备当前锁定状态

password: SP1/RM1认证所需密码, 该值由设备生成, 不能修改。

id: SP2/RM2通信所需, 该值由设备生成, 不能修改。

subdevice: 保留字段。置0

key: SP2/RM2通信密钥, 由设备产生, 不能修改。

```
</pre>
```

### 1.3 device\_add请求参数说明

```
<pre>Arguments:
{
  "api_id":12,
  "command":"device_add",
  "mac":"00:11:22:33:44:55",
  "type":"RM2",
  "name":"智能插座",
  "lock":0,
  "password":1028000492,
  "id":0,
  "subdevice":0,
  "key":"00000000000000000000000000000000"
}
```

api\_id: 命令ID。

command: 命令（可选）。

以下字段从probe\_list中获取到，不能修改。

mac: 设备的mac地址。

type: 设备类型，目前支持的设备类型，SP1/SP2/RM1/RM2。（注:必须大写）

name: 设备名称，UTF8编码。

lock: 设备当前锁定状态

password: SP1/RM1认证所需密码，该值由设备生成，不能修改。

id: SP2/RM2通信所需，该值由设备生成，不能修改。

subdevice: 保留字段。置0

key: SP2/RM2通信密钥，由设备产生，不能修改。

```
</pre>
```

```
<pre>Retval:
```

```
{
  "code":0,
  "msg":"Add device success"
}
```

code: 返回码

msg : 返回消息

```
</pre>
```

## 1.4 device\_update请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":13,  
  "command":"device_update",  
  "mac":"00:11:22:33:44:55",  
  "name":"智能插座",  
  "lock":0,  
}
```

api\_id: 命令ID。

command: 命令。 (可选)

mac: 要修改的设备的mac地址。 (必填)

name: 要修改的名称, UTF8编码。 (可选)

lock: 是否锁定。 (可选)

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "msg":"Update device info success"  
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.5 device\_delete请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":14,  
  "command":"device_delete",  
  "mac":"00:11:22:33:44:55"  
}
```

api\_id: 命令ID。

command: 命令。 (可选)

mac: 要删除的设备的MAC地址。 (必填)

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "msg":"Delete device success"  
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.6 device\_lan\_ip请求参数说明

<pre>Arguments:

```
{
  "api_id":15,
  "command":"device_lan_ip",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要删除的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "lan_ip":"192.168.1.113",
  "msg":"mac:00:11:22:33:44:55 get lan ip success"
}
```

code: 返回码

lan\_ip: 设备当前的局域网IP地址.

msg : 返回消息

</pre>

## 1.7 device\_state请求参数说明

<pre>Arguments:

```
{
  "api_id":16,
  "command":"device_state",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要删除的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "status":"LOCAL",
  "msg":"mac:00:11:22:33:44:55 get state success"
}
```

code: 返回码

status: 设备当前的网络状态. (NOT\_INIT:未初始化 LOCAL:局域网 REMOTE:远程 OFFLINE:离线)

msg : 返回消息

</pre>



## 1.8 sp1\_auth请求参数说明

<pre>Arguments:

```
{
  "api_id":41,
  "command":"sp1_auth",
  "mac":"00:11:22:33:44:55",
  "password":1028000492
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要登录的设备的MAC地址。（必填）

password: 设备的登录密码。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"sp1_auth success",
  "status":1,
  "name":"Smart Plug 1",
  "lock":0,
  "periodic_task":[
    {
      "enable":1,
      "on_time":"08:00",
      "off_time":"15:00",
      "repeat":0
    },
    {
      "enable":1,
      "on_time":"--:--",
      "off_time":"00:00",
      "repeat":127
    }
  ]
}
```

code: 返回码

msg : 返回消息

status: 插座当前状态（1: 开 0: 关）

name: 插座的名称.

lock: 插座是否上锁.

periodic\_task: 插座的周期任务.

enable: 周期任务使能.

on\_time: 周期任务开启时间, 若没有设置, 则为"--:--".

off\_time: 周期任务关闭时间, 若没有设置, 则为"--:--".

repeat: 周期任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。

</pre>

## 1.9 sp1\_control请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":42,  
  "command":"sp1_control",  
  "mac":"00:11:22:33:44:55",  
  "status": 0  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

status: 要设置的状态(0:关闭 1: 开启)。（必填）

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "msg":"Set sp1 status success"  
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.10 spl\_refresh请求参数说明

<pre>Arguments:

```
{
  "api_id":43,
  "command":"spl_refresh",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"Refresh spl status success"
  "status": 0
  "name":"Smart Plug 1",
  "lock":0,
  "periodic_task":[
    {
      "enable":1,
      "on_time":"08:00",
      "off_time":"15:00",
      "repeat":0
    },
    {
      "enable":1,
      "on_time":"--:--",
      "off_time":"00:00",
      "repeat":127
    }
  ]
}
```

code: 返回码

msg : 返回消息

status: 当前开关状态（0：关闭 1：开启）

name: 插座的名称.

lock: 插座是否上锁.

periodic\_task: 插座的周期任务.

enable: 周期任务使能.

on\_time: 周期任务开启时间, 若没有设置, 则为"--:--".

off\_time: 周期任务关闭时间, 若没有设置, 则为"--:--".

repeat: 周期任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。

</pre>

## sp1\_task请求参数说明

<pre>Arguments:

```
{
  "api_id":44,
  "command":"sp1_task",
  "mac":"00:11:22:33:44:55",
  "name":"Smart Plug 1",
  "lock":0,
  "periodic_task":[
    {
      "enable":1,
      "on_time":"08:00",
      "off_time":"15:00",
      "repeat":0
    },
    {
      "enable":1,
      "on_time":"--:--",
      "off_time":"00:00",
      "repeat":127
    }
  ]
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

name: 插座的名称。

lock: 插座是否上锁。

periodic\_task: 插座的周期任务。

enable: 周期任务使能。

on\_time: 周期任务开启时间, 若没有设置, 则为"--:--".

off\_time: 周期任务关闭时间, 若没有设置, 则为"--:--".

repeat: 周期任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。

注意: 设置任务需要把原来的任务也同时设置, 添加与删除为同一个接口。同时还需要设置设备名称和锁定状态

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"Set sp1 task success"
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.11 sp2\_refresh请求参数说明

```
<pre>Arguments:
{
  "api_id":71,
  "command":"sp2_refresh",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。  
command: 命令。(可选)  
mac: 要刷新的设备的MAC地址。(必填)

```
</pre>
<pre>Retval:
{
  "code":0,
  "msg":"Refresh sp2 status success",
  "status":0,
  "name":"Smart Plug 2",
  "lock":0,
  "periodic_task":[
    {
      "enable":1,
      "on_time":"08:00",
      "off_time":"15:00",
      "repeat":0
    },
    {
      "enable":1,
      "on_time":"--:--",
      "off_time":"00:00",
      "repeat":127
    }
  ],
  "timer_task":[
    {
      "on_enable":1,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":0,
      "off_time":"2014-07-02 15:58:03"
    },
    {
      "on_enable":0,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":1,
      "off_time":"2014-07-03 16:58:03"
    },
    {
      "on_enable":1,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":1,
      "off_time":"2014-07-05 18:58:03"
    }
  ]
}
```

code: 返回码

msg : 返回消息

status: 当前设备的开关状态 (0: 关闭 1: 开启)

name: 插座的名称.

lock: 插座是否上锁.

periodic\_task: 插座的周期任务.

enable: 周期任务使能.

on\_time: 周期任务开启时间, 若没有设置, 则为"--:--".

off\_time: 周期任务关闭时间, 若没有设置, 则为"--:--".

repeat: 周期任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。

timer\_task: 插座的定时任务.

on\_enable: 定时任务的开启使能

on\_time: 定时任务的开启时间点

off\_enable: 定时任务的关闭使能

off\_time: 定时任务的关闭时间点

```
</pre>
```

## 1.12 sp2\_control请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":72,  
  "command":"sp2_control",  
  "mac":"00:11:22:33:44:55",  
  "status":1  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

status: 要设置的开关状态（0:关闭 1: 开启）。（必填）

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "msg":"Set sp2 status success"  
}
```

code: 返回码

msg : 返回消息

</pre>

## sp2\_task请求参数

```
<pre>Arguments:
{
  "api_id":73,
  "command":"sp2_task",
  "mac":"00:11:22:33:44:55",
  "name":"Smart Plug 2",
  "lock":0,
  "periodic_task":[
    {
      "enable":1,
      "on_time":"08:00",
      "off_time":"15:00",
      "repeat":0
    },
    {
      "enable":1,
      "on_time":"--:--",
      "off_time":"00:00",
      "repeat":127
    }
  ],
  "timer_task":[
    {
      "on_enable":1,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":0,
      "off_time":"2014-07-02 15:58:03"
    },
    {
      "on_enable":0,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":1,
      "off_time":"2014-07-03 16:58:03"
    },
    {
      "on_enable":1,
      "on_time":"2014-07-02 14:58:03",
      "off_enable":1,
      "off_time":"2014-07-05 18:58:03"
    }
  ]
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

name: 插座的名称。

lock: 插座是否上锁。

periodic\_task: 插座的周期任务。

enable: 周期任务使能。

on\_time: 周期任务开启时间, 若没有设置, 则为"--:--"。

off\_time: 周期任务关闭时间, 若没有设置, 则为"--:--"。

repeat: 周期任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6:周六)。

timer\_task: 插座的定时任务。

on\_enable: 定时任务的开启使能

on\_time: 定时任务的开启时间点

off\_enable: 定时任务的关闭使能

off\_time: 定时任务的关闭时间点

注意: 设置任务需要把原来的任务也同时设置, 添加与删除为同一个接口。同时还需要设置设备名称和锁定状态

```
</pre>
```

```
<pre>Retval:
```

```
{
  "code":0,
  "msg":"Set sp2 task success"
}
```

code: 返回码

msg : 返回消息

```
</pre>
```

## sp2\_current\_power请求参数说明

<pre>Arguments:

```
{
  "api_id":74,
  "command":"sp2_current_power",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "current_power":121.000000,
  "msg":"Get sp2 current power success"
}
```

code: 返回码

current\_power: 设备当前的实时功率。

msg : 返回消息

</pre>

## sp2\_24\_power请求参数说明

<pre>Arguments:

```
{
  "api_id":75,
  "command":"sp2_24_power",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "current_power":121.000000,
  "24hour_power":[0.000000, 60.000000, 80.000000, ...],
  "msg":"Get sp2 last 24hour power success"
}
```

code: 返回码

current\_power: 设备当前的实时功率。

24hour\_power: 设备最近24小时的功率，每5分钟一个数据点，总共288个。

msg : 返回消息

</pre>



## sp2\_week\_power请求参数说明

<pre>Arguments:

```
{
  "api_id":76,
  "command":"sp2_week_power",
  "mac":"00:11:22:33:44:55",
  "week_index":2
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

week\_index: 当前年的第几周（从1开始）

</pre>

<pre>Retval:

```
{
  "code":0,
  "week_energy":[
    {
      "peak_energy":0.250000,
      "low_energy":0.750000,
      "on_time":50,
      "standby_energy":0.180000
    },
    {
      "peak_energy":0.450000,
      "low_energy":1.250000,
      "on_time":300,
      "standby_energy":0.580000
    }
  ],
  "msg":"Get sp2 week's energy info success."
}
```

code: 返回码

week\_energy: 设备指定周的能耗统计，每天一个数据点，最多7个点。

peak\_energy: 峰电能耗统计

low\_energy: 谷电能耗统计

on\_time: 运行时间

standby\_energy: 待机能耗统计

msg : 返回消息

</pre>

## sp2\_month\_power请求参数说明

<pre>Arguments:

```
{
  "api_id":77,
  "command":"sp2_month_power",
  "mac":"00:11:22:33:44:55",
  "year":2014,
  "month_index":3
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

year: 指定年份

month\_index: 指定月份

</pre>

<pre>Retval:

```
{
  "code":0,
  "month_energy":[
    {
      "peak_energy":0.250000,
      "low_energy":0.750000,
      "on_time":50,
      "standby_energy":0.180000
    },
    {
      "peak_energy":0.450000,
      "low_energy":1.250000,
      "on_time":300,
      "standby_energy":0.580000
    }
  ],
  "msg":"Get sp2 month's energy info success."
}
```

code: 返回码

month\_energy: 设备指定年月的能耗统计，每天一个数据点，最多31个点。

peak\_energy: 峰电能耗统计

low\_energy: 谷电能耗统计

on\_time: 运行时间

standby\_energy: 待机能耗统计

msg : 返回消息

</pre>

## sp2\_year\_power请求参数说明

<pre>Arguments:

```
{
  "api_id":78,
  "command":"sp2_year_power",
  "mac":"00:11:22:33:44:55",
  "year":2014
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

year: 指定年份）

</pre>

<pre>Retval:

```
{
  "code":0,
  "year_energy":[
    {
      "peak_energy":100.250000,
      "low_energy":150.750000,
      "on_time":1550,
      "standby_energy":30.180000
    },
    {
      "peak_energy":120.450000,
      "low_energy":170.250000,
      "on_time":3000,
      "standby_energy":80.580000
    }
  ],
  "msg":"Get sp2 year's energy info success."
}
```

code: 返回码

year\_energy: 设备指定年份的能耗统计，每月一个数据点，最多12个点。

peak\_energy: 峰电能耗统计

low\_energy: 谷电能耗统计

on\_time: 运行时间

standby\_energy: 待机能耗统计

msg : 返回消息

</pre>

## sp2\_get\_standby\_power请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":79,  
  "command":"sp2_get_standby_power",  
  "mac":"00:11:22:33:44:55",  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "enable":1,  
  "standby_power":10  
  "msg":"Get sp2 standby power success."  
}
```

code: 返回码

enable: 待机功率是否生效

standby\_power: 待机功率值

msg : 返回消息

</pre>

## sp2\_set\_standby\_power请求参数说明

---

<pre>Arguments:

```
{  
  "api_id":80,  
  "command":"sp2_set_standby_power",  
  "mac":"00:11:22:33:44:55",  
  "enable":1,  
  "standby_power":5  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

enable: 设置待机功率是否生效

standby\_power: 设置待机功率

</pre>

<pre>Retval:

```
{  
  "code":0,  
  "msg":"Set sp2 standby power success."  
}
```

code: 返回码

msg : 返回消息

</pre>

### 1.13 rm1\_auth请求参数说明

```
<pre>Arguments:
```

```
{  
  "api_id":101,  
  "command":"rm1_auth",  
  "mac":"00:11:22:33:44:55",  
  "password":1028000492  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

```
</pre>
```

```
<pre>Retval:
```

```
{  
  "code":0,  
  "temperature":23.100000,  
  "msg":"rm1 auth success"  
}
```

code: 返回码

temperature: 设备当前所在环境的温度。

msg : 返回消息

```
</pre>
```

#### h3. 1.14 rm1\_study请求参数说明

```
<pre>Arguments:
```

```
{  
  "api_id":102,  
  "command":"rm1_study",  
  "mac":"00:11:22:33:44:55"  
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

```
</pre>
```

```
<pre>Retval:
```

```
{  
  "code":0,  
  "msg":"rm1 enter study mode success"  
}
```

code: 返回码

msg : 返回消息

```
</pre>
```

## 1.15 rm1\_code请求参数说明

<pre>Arguments:

```
{
  "api_id":103,
  "command":"rm1_code",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":-100,
  "msg":"rm1 check data failed"
}
```

Retval:

```
{
  "code":0,
  "data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e",
  "msg":"rm1 check data success"
}
```

code: 返回码，-100表示没查到学习到的数据，0表示查到学习的数据

data: 学习到的控制码。

msg : 返回消息

</pre>

## 1.16 rm1\_send请求参数说明

<pre>Arguments:

```
{
  "api_id":104,
  "command":"rm1_send",
  "mac":"00:11:22:33:44:55",
  "data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"rm1 send data success"
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.17 rm2\_refresh请求参数说明

---

<pre>Arguments:

```
{
  "api_id":131,
  "command":"rm2_refresh",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "temperature":23.100000,
  "msg":"Refresh rm2 temperature success"
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.18 rm2\_study请求参数说明

---

<pre>Arguments:

```
{
  "api_id":132,
  "command":"rm2_study",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"rm2 enter study mode success"
}
```

code: 返回码

msg : 返回消息

</pre>

## 1.19 rm2\_code请求参数说明

<pre>Arguments:

```
{
  "api_id":133,
  "command":"rm2_code",
  "mac":"00:11:22:33:44:55"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":-1,
  "msg":"rm2 check data failed"
}
```

Retval:

```
{
  "code":0,
  "data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e",
  "msg":"rm2 check data success"
}
```

code: 返回码，-1表示没查到学习到的数据，0表示查到

data: 学习到的控制码。

msg : 返回消息

</pre>

## 1.20 rm2\_send请求参数说明

<pre>Arguments:

```
{
  "api_id":134,
  "command":"rm2_send",
  "mac":"00:11:22:33:44:55",
  "data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e"
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "msg":"rm2 send data success"
}
```

code: 返回码

msg : 返回消息

</pre>



## 1.21 a1\_refresh请求参数说明

<pre>Arguments:

```
{
  "api_id":161,
  "command":"a1_refresh",
  "mac":"00:11:22:33:44:55",
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

</pre>

<pre>Retval:

```
{
  "code":0,
  "temperature":25.0,
  "humidity":56.3,
  "light":3,
  "air":2,
  "noisy":1,
  "name":"A1 Eair",
  "lock":0,
  "msg":"Refresh a1 sensor's value success"
}
```

code: 返回码

temperature: 温度值, 浮点型

humidity: 湿度值, 浮点型

light: 光照度, 整型 (0:暗 1:昏暗 2:正常 3:亮)

air: 空气质量, 整形 (0:优 1:良 2:正常 3:差)

noisy: 噪声, 整形 (0:寂静 1:正常 2:吵闹)

name: 设备名称, 字符串

lock: 设备锁定状态, 整型 (0:未锁 1:锁定)

msg : 返回消息

</pre>

## 1.22 a1\_task\_list请求参数说明

<pre>Arguments:

```
{
  "api_id":162,
  "command":"a1_task_list",
  "mac":"00:11:22:33:44:55",
}
```

api\_id: 命令ID。

command: 命令。 (可选)

mac: 要操作的设备的MAC地址。(必填)

</pre>

<pre>Retval:

```
{
  "code":0,
  "list" :[
    {
      "mac": "00:11:22:33:44:55",
      "time_enable":0,
      "task_enable":1,
      "index":0,
      "start_time":"10:30",
      "end_time":"18:30",
      "repeat":7,
      "sensor_type":0,
      "sensor_trigger": 0,
      "sensor_value":25.0,
      "task_name":"打开电视机"
    },
    {
      "mac": "55:44:33:22:11:00",
      "time_enable":1,
      "task_enable":1,
      "index":2,
      "start_time":"10:30",
      "end_time":"17:30",
      "repeat":5,
      "sensor_type":2,
      "sensor_trigger": 1,
      "sensor_value":3.0,
      "task_name":"打开音响"
    }
  ],
  "msg":"Get a1 task list success"
}
```

code: 返回码

list: 任务列表

mac: 执行任务的设备mac地址。字符串

time\_enable: 时间使能。 整型

task\_enable: 任务使能。 整型

index: 任务编号, (删除任务时使用)。 整型

start\_time: 任务执行的开始时间。 字符串

end\_time: 任务执行的结束时间。 字符串

repeat: 任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 整型

sensor\_type: 任务执行条件的传感器类型, (0: 温度 1:湿度 2:光照 3:空气质量 4:噪声)。 整型

sensor\_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升 1:下降)。 整型

sensor\_value: 任务执行条件传感器的值。 浮点型

task\_name: 执行任务的名称。 字符串

msg : 返回消息

</pre>

### 1.23 a1\_add\_task请求参数说明

```
<pre>Arguments:
{
  "api_id":163,
  "command":"a1_add_task",
  "mac":"00:11:22:33:44:55",
  "task_name": "关闭空调",
  "time_enable":1,
  "task_enable": 1,
  "start_time": "2014-06-06 10:30:00",
  "end_time": "2014-06-06 18:30:00",
  "repeat": 7,
  "sensor_type": 0,
  "sensor_trigger": 1,
  "sensor_value" : 20.0,
  "device_mac": "aa:bb:cc:dd:ee:ff",
  "device_id": 25,
  "device_key": "097628343fe99e23765c1513accf8b02",
  "device_type": "RM2",
  "task_data":
  "2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e",
  "status": 0
}
```

api\_id: 命令ID。

command: 命令。（可选）

mac: 要操作的设备的MAC地址。（必填）

task\_name: 添加的任务的名称。

time\_enable: 任务时间使能。

task\_enable: 任务使能。

start\_time: 任务开始时间。必须按照该格式传入时间参数，其中年月日只在time\_enable为0时生效(年月日必须为当天的准确时间)。

end\_time: 任务结束时间。必须按照该格式传入时间参数，其中年月日只在time\_enable为0时生效(年月日必须为当天的准确时间)。

repeat: 任务重复周期。在time\_enable为1时有效。

sensor\_type: 任务执行条件的传感器类型。

sensor\_trigger: 任务执行条件的传感器值变化趋势。

sensor\_value: 任务执行条件的传感器值。

device\_mac: 执行任务的设备的mac地址。

device\_id: 执行任务的设备的ID。(该值从probe\_list中返回)

device\_key: 执行任务的设备的key。(该值从probe\_list中返回)

device\_type: 执行任务的设备的类型。(该值从probe\_list中返回)

task\_data: 执行任务的命令。(当执行任务设备为SP2时无效，若是RM2，则设置为rm2\_code中返回的data)

status: 执行任务的设备要设置的状态。(仅对执行任务设备为SP2时有效，非SP2设备可取消该字段)

</pre>

```
<pre>Retval:
```

```
{
  "code":0,
  "list":[
  {
    "mac": "00:11:22:33:44:55",
    "time_enable":0,
    "task_enable":1,
```

```
"index":0,
"start_time":"10:30",
"end_time":"18:30",
"repeat":7,
"sensor_type":0,
"sensor_trigger": 0,
"sensor_value":25.0,
"task_name":"打开电视机"
},
{
"mac": "55:44:33:22:11:00",
"time_enable":1,
"task_enable":1,
"index":2,
"start_time":"10:30",
"end_time":"17:30",
"repeat":5,
"sensor_type":2,
"sensor_trigger": 1,
"sensor_value":3.0,
"task_name":"打开音响"
}
],
"msg":"Get all task list success"
}
```

code: 返回码

list: 任务列表

mac: 执行任务的设备mac地址。字符串

time\_enable: 时间使能。 整型

task\_enable: 任务使能。 整形

index: 任务编号, (删除任务时使用)。 整形

start\_time: 任务执行的开始时间。 字符串

end\_time: 任务执行的结束时间。 字符串

repeat: 任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 整型

sensor\_type: 任务执行条件的传感器类型, (0: 温度 1:湿度 2:光照 3:空气质量 4:噪声)。 整型

sensor\_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升 1:下降)。 整型

sensor\_value: 任务执行条件传感器的值。 浮点型

task\_name: 执行任务的名称。 字符串

msg : 返回消息

</pre>

## 1.24 a1\_del\_task请求参数说明

<pre>Arguments:

```
{
  "api_id":164,
  "command":"a1_del_task",
  "mac":"00:11:22:33:44:55",
  "index":0
}
```

api\_id: 命令ID。

command: 命令。 (可选)

mac: 要操作的设备的MAC地址。(必填)

index: 要删除的任务的index。(该值从返回的列表中获取)

</pre>

<pre>Retval:

```
{
  "code":0,
  "list":[
    {
      "mac": "00:11:22:33:44:55",
      "time_enable":0,
      "task_enable":1,
      "index":0,
      "start_time":"10:30",
      "end_time":"18:30",
      "repeat":7,
      "sensor_type":0,
      "sensor_trigger": 0,
      "sensor_value":25.0,
      "task_name":"打开电视机"
    },
    {
      "mac": "55:44:33:22:11:00",
      "time_enable":1,
      "task_enable":1,
      "index":2,
      "start_time":"10:30",
      "end_time":"17:30",
      "repeat":5,
      "sensor_type":2,
      "sensor_trigger": 1,
      "sensor_value":3.0,
      "task_name":"打开音响"
    }
  ],
  "msg":"Get a1 task list success"
}
```

code: 返回码

list: 任务列表

mac: 执行任务的设备mac地址。字符串

time\_enable: 时间使能。 整型

task\_enable: 任务使能。 整型

index: 任务编号, (删除任务时使用)。 整型

start\_time: 任务执行的开始时间。 字符串

end\_time: 任务执行的结束时间。 字符串

repeat: 任务的重复周期, 以周为一个周期, 0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6:周六)。 整型

sensor\_type: 任务执行条件的传感器类型, (0: 温度 1:湿度 2:光照 3:空气质量 4:噪声)。 整型

sensor\_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升 1:下降)。 整型

sensor\_value: 任务执行条件传感器的值。 浮点型

task\_name: 执行任务的名称。 字符串

msg : 返回消息

</pre>

## easyconfig请求参数说明

```
<pre>Arguments:
```

```
{  
  "api_id":10000,  
  "command":"easyconfig",  
  "ssid":"abcdefg",  
  "password":"11111111",  
  "broadlinkv2":1,  
  "dst":"192.168.1.1"  
}
```

api\_id: 命令ID。

command: 命令。（可选）

ssid: 设备将要连接的网络名称。（必填）

password: 设备将要连接的网络密码。（必填）

broadlinkv2: 将要配置的设备是否采用二代（含）以后的broadlink无线模块。（选项，默认是二代模块）

dst: 网关地址。（选填，填写网关地址可加快配置速度）

\*注意：该接口非线程安全\*

```
</pre>
```

```
<pre>Retval:
```

```
{  
  "code":0,  
  "msg":"easyconfig success"  
}
```

code: 返回码 0表示成功，1表示超时，-1表示用户取消或者其他错误

msg : 返回消息

```
</pre>
```

### h1. 取消easyconfig请求参数说明

```
<pre>Arguments:
```

```
{  
  "api_id":10001,  
  "command":"cancel_easyconfig"  
}
```

api\_id: 命令ID。

command: 命令。（可选）

\*注意：该接口非线程安全\*

```
</pre>
```

```
<pre>Retval:
```

```
{  
  "code":0,  
  "msg":"easyconfig success"  
}
```

code: 返回码 0表示成功，1表示超时，-1表示用户取消或者其他错误

msg : 返回消息

```
</pre>
```

注意：返回的code除了已经说明的，其余返回的code对应的错误信息如下：

0	成功
-1	设备所在网络已改变或网络密码已经修改。
-2	设备已在其他地方登录，如需继续控制，请重新登录(针对rm1/sp1)
-3	设备不在线
-4	不支持的操作
-5	空间满
-6	数据结构异常
-7	设备已经复位，需进入局域网重新认证。(针对sp1/rm1以外的设备)
-100	超时
-101	网络线程找不到该设备。
-102	内存不足
-103	设备未初始化
-104	网络线程已暂停
-105	返回消息类型错误
-106	操作过于频繁
-107	服务器已拒绝该license操作, 请联系客服处理
-108	设备不在局域网中
-10000	未知错误

