海曼MQTT设备端SDK开发文档

(V1.1)

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 胡伟涵 | 日期 | 2018年01月08日 |
| 审核 |  | 日期 |  |
| 批准 |  | 日期 |  |

**文档修改记录**

| 修改日期 | 版本/状态 | 作者 | 审批 | 修改描述 |
| --- | --- | --- | --- | --- |
| 2018-01-08 | V1.0 | 胡伟涵 |  | 初稿编写 |
|  |  |  |  |  |

**目录**

[海曼MQTT设备端SDK开发文档 1](file:///D:\MQTT\MQTT设备端SDK开发文档.docx#_Toc504401124)

[1. 硬件SDK介绍 1](#_Toc504401125)

[1.1. 主要特点 1](#_Toc504401126)

[1.2. 术语 1](#_Toc504401127)

[1.3. 资源消耗 1](#_Toc504401128)

[2. 使用方法 2](#_Toc504401129)

[2.1. 使用步骤 2](#_Toc504401130)

[2.2. 使用注意 2](#_Toc504401131)

[3. 接口说明 3](#_Toc504401132)

[3.1. 初始化SDK 3](#_Toc504401133)

[3.2. SDK MQTT初始化连接服务端 3](#_Toc504401134)

[3.3. SDK MQTT 主题订阅初始化 4](#_Toc504401135)

[3.4. SDK MQTT 循环处理PUBLISH Data 函数 4](#_Toc504401136)

1. 硬件SDK介绍

海曼物联硬件SDK主要为传统厂商的产品接入互联网，并能通过APP对产品进行监测、控制和统计分析等。

* 1. 主要特点
* 本地设备自动探测和扫描联网；
* 对应相应的协议文档开发，可接入其他MQTT服务器；
* 提供云端固件升级和远程配置、诊断和管理服务；
* 支持设备安全认证和数据加密机制；
* 纯C代码；
* 使用操作简单快捷。
  1. 术语
* 设备：指使用SDK的硬件；
* 服务器：指云平台，设备所要连接的服务器；
* 外网：指设备通过互联网连接到服务器的通道；

1. 使用方法
   1. 使用步骤

* 首先需要给产品分配个4 byte的企业ID和5 byte的产品ID；
* 在MQTT平台上接入设备的唯一是通过4 byte企业ID+5 byte产品ID+12 byte设备的MAC区分；
* 然后转到接入设备开发，先选择对应SDK，定义修改数据类型；
* 初始化TCP并连接上TCP服务器，将函数xlink\_send\_cb()里面的数据通过外网发送出去，接收到外网数据后调用xlink\_receive\_data()往SDK传入数据；
* 初始化UDP，将函数xlink\_send\_cb()里面的数据通过内网发送出去，接收到内网数据后调用xlink\_receive\_data()往SDK传入数据；
* 调用“network\_sdk\_init()”初始化（当连接上外网TCP时调用network\_sdk\_connect\_cloud()启动外网通讯，如果断开即调用network\_sdk\_disconnect\_cloud()停止）；
* 循环执行“network\_sdk\_process()”；
* 如果使用外网，需等待“EVENT\_TYPE\_STATUS”事件触发且status = 1，才能向服务器发送数据；
* 设备连接平台成功后，便可收发数据，如果需要使用到数据端点，请参考“数据端点文档”；
* 需要停止时，调用“xlink\_sdk\_uninit()”停止SDK；
* 如需要重置SDK时，调用network\_sdk\_reset();
* 如需要上报数据端点时，调用network\_update\_datapoint();
* 如需要请求数据时，调用network\_request\_event()。
  1. 使用注意
* sdk对不同类型产品，有不同的公钥，目前Publickey为bd17df6d548211e7。
* 请严格按照海曼与MQTT服务器通讯协议文档和WIFI网关应用层透传协议协议文档，进行相应的开发。

1. 接口说明
   1. 初始化SDK

|  |
| --- |
| int network\_sdk\_init(struct xlink\_sdk\_instance\_t\*\* sdk\_instance); |

功能：

初始化SDK函数。

参数：

| **参数** | **说明** |
| --- | --- |
| \*\*sdk\_instance | sdk 实例指针的指针 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 成功 |
| 1 | 失败 |

* 1. SDK MQTT初始化连接服务端

int mqtt\_initconnect\_func(void);

功能：

SDK MQTT初始化连接服务端。

参数：

| **参数** | **说明** |
| --- | --- |
| NULL | 无 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK MQTT 主题订阅初始化

int mqtt\_topicsubscibe\_func(void);

功能：

SDK MQTT主题订阅。

参数：

| **参数** | **说明** |
| --- | --- |
| NULL | 无 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK MQTT 循环处理PUBLISH Data 函数

int mqtt\_loophandle\_func(void);

功能：

SDK MQTT循环处理 PUBLISH Data函数。

参数：

| **参数** | **说明** |
| --- | --- |
| NULL | 无 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 初始化设置系统时间

|  |
| --- |
| int hm\_getServeTimeAndSetSysTime(int \*time\_p); |

功能：

此函数初始化服务器返回的系统时间。

参数：

| **参数** | **说明** |
| --- | --- |
| \*time\_p | 返回服务器获取的时间戳值 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 校正系统时间

|  |
| --- |
| int hm\_adjust\_times(void)； |

功能：

此函数为校准SDK的系统时间。现设置时间间隔请求为1 min。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK获取当前的系统时间

|  |
| --- |
| uint32\_t get\_user\_times(void)； |

功能：

此函数为获取SDK的当前系统时间。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| Uint32\_t | 返回当前系统时间戳 |

* 1. SDK PASS数据处理

|  |
| --- |
| void pass\_data\_process(int cmddata,char\* data,int data\_len)； |

功能：

此函数为处理PASS数据，这接口数据为APP通过MQTT发送的数据。

参数：

| **参数** | **说明** |
| --- | --- |
| cmddata | app发送的cmd指令类型 |
| data | app发送的数据内容 |
| data\_len | app发送的数据长度 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设备上报在线

|  |
| --- |
| int HM\_gw\_report\_online(void); |

功能：

此函数为设备上报在线。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 设备发送获取分配的DeviceId指令

|  |
| --- |
| int HM\_GW\_get\_ServerDeviceId(void); |

功能：

此函数为获取服务器分配的DeviceId。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

int hm\_get\_deviceid(int server\_deviceid)

* 1. SDK 获得服务器分配的DeviceId

|  |
| --- |
| int hm\_get\_deviceid(int server\_deviceid); |

功能：

此函数为获得服务器分配的DeviceId。

参数：

| **参数** | **说明** |
| --- | --- |
| server\_deviceid | 服务器分配下去唯一的DeviceId |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 通过pass主题发送设备信息

|  |
| --- |
| int hm\_sent\_pass\_mesgdata(void); |

功能：

此函数为设备通过pass主题发送设备信息。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 通过point主题发送设备信息

|  |
| --- |
| int hm\_sent\_point\_mesgdata(void); |

功能：

此函数为设备通过point主题发送设备信息。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 通过pass主题发送网关参数

|  |
| --- |
| int hm\_sent\_pass\_gatewayparm(void); |

功能：

此函数为设备通过pass主题发送网关参数。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 通过pass主题发送私钥

|  |
| --- |
| int HM\_sent\_pass\_askeymsg(int cmddata); |

功能：

此函数为设备通过pass主题发送设备私钥。

参数：

| **参数** | **说明** |
| --- | --- |
| cmddata | app发送的cmd指令类型 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 通过point主题发送私钥

|  |
| --- |
| int HM\_sent\_point\_aeskeymsg(void); |

功能：

此函数为设备通过point主题发送设备私钥。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 发送获取服务器时间指令

|  |
| --- |
| int HM\_get\_systemtime(void); |

功能：

此函数为设备发送获取服务器时间指令。

参数：

| **参数** | **说明** |
| --- | --- |
| --- | --- |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 清除网关子列表信息

|  |
| --- |
| int HM\_sent\_zigbeelist\_clearmsg(unsigned char type\_mode,unsigned int  server\_deviceid); |

功能：

此函数为设备发送清除网关子列表信息。

参数：

| **参数** | **说明** |
| --- | --- |
| type\_mode | 删除列表类型 |
| server\_deviceid | 服务器分配的deviceid |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 发送确认PASS数据应答

|  |
| --- |
| int HM\_pass\_ack\_data(int cmddata,char\* data,int data\_len); |

功能：

此函数为设备发送确认PASS数据应答。

参数：

| **参数** | **说明** |
| --- | --- |
| cmd | app发送的cmd指令类型 |
| data | app发送pass的数据内容 |
| data\_len | app发送pass的数据长度 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

int HM\_Sent\_point\_data(char\* data, int length)

* 1. SDK 发送point主题数据

|  |
| --- |
| int HM\_Sent\_point\_data(char\* data, int length); |

功能：

此函数为设备发送point主题数据。

参数：

| **参数** | **说明** |
| --- | --- |
| data | 发送point的数据内容 |
| length | 发送point的数据长度 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 发送其他主题数据

|  |
| --- |
| int HM\_Sent\_mfunc\_data(char\* data, int length); |

功能：

此函数为设备发送其他主题数据。

参数：

| **参数** | **说明** |
| --- | --- |
| data | 发送mfunc的数据内容 |
| length | 发送mfunc的数据长度 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 发送pass主题数据

|  |
| --- |
| int HM\_Sent\_pass\_data(char\* data, int length); |

功能：

此函数为设备发送其他主题数据。

参数：

| **参数** | **说明** |
| --- | --- |
| data | 发送pass的数据内容 |
| length | 发送pass的数据长度 |

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |

* 1. SDK 获取子设备门磁的状态SS

|  |
| --- |
| void hm\_get\_doorsensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_DOOR\_PARAM zb\_type); |

功能：

此函数为获取子设备门磁的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 门磁的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备门磁的SE参数

|  |
| --- |
| int hm\_set\_doorsensor\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_DOOR\_PARAM zb\_type); |

功能：

此函数为设置子设备门磁的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 门磁的SE参数 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备门磁的SE参数

|  |
| --- |
| int hm\_get\_doorsensor\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_DOOR\_PARAM zb\_type); |

功能：

此函数为设置子设备门磁的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 门磁的SE参数 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备水浸的状态SS

|  |
| --- |
| void hm\_get\_watersensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_WATER\_PARAM zb\_type); |

功能：

此函数为获取子设备水浸的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 水浸的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备红外的状态SS

|  |
| --- |
| void hm\_get\_pirsensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_PIR\_PARAM zb\_type); |

功能：

此函数为获取子设备红外的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 红外的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK设置子设备红外的SE参数

|  |
| --- |
| int hm\_set\_pirsensor\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_PIR\_PARAM zb\_type); |

功能：

此函数为设置子设备红外的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 红外的SE参数 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK获取子设备红外的SE参数

|  |
| --- |
| int hm\_get\_pirsensor\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_PIR\_PARAM zb\_type); |

功能：

此函数为获取子设备红外的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 红外的SE参数 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备烟感的状态SS

|  |
| --- |
| void hm\_get\_smokesensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_SMOKE\_PARAM zb\_type); |

功能：

此函数为获取子设备烟感的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 烟感的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备气感的状态SS

|  |
| --- |
| void hm\_get\_gassensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_GAS\_PARAM zb\_type); |

功能：

此函数为获取子设备气感的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 气感的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备CO的状态SS

|  |
| --- |
| void hm\_get\_cosensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_CO\_PARAM zb\_type)； |

功能：

此函数为获取子设备气感的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | CO的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备紧急按钮的状态SS

|  |
| --- |
| void hm\_get\_sossensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_SOS\_PARAM zb\_type)； |

功能：

此函数为获取子设备紧急按钮的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | CO的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备遥控器的状态SS

|  |
| --- |
| void hm\_get\_remotecontrol\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_REMOTE\_PARAM zb\_type)； |

功能：

此函数为获取子设备紧遥控器的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 遥控器的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备温湿度的状态SS

|  |
| --- |
| void hm\_get\_humituresensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_HUMITURE\_PARAM zb\_type)； |

功能：

此函数为获取子设备温湿度的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 温湿度的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备温湿度的SE参数

|  |
| --- |
| void hm\_set\_humituresensor\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_HUMITURE\_PARAM zb\_type)； |

功能：

此函数为设置子设备温湿度的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 温湿度的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备温湿度的SE参数

|  |
| --- |
| void hm\_get\_humituresensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_HUMITURE\_PARAM zb\_type)； |

功能：

此函数为获取子设备温湿度的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 温湿度的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备RGB灯的状态SS

|  |
| --- |
| void hm\_get\_rgblight\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_RGB\_PARAM zb\_type)； |

功能：

此函数为获取子设备RGB灯的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | RGB灯的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备RGB灯的SE参数

|  |
| --- |
| void hm\_set\_rgblight\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_RGB\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备RGB灯的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | RGB灯的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备RGB灯的SE参数

|  |
| --- |
| void hm\_get\_rgblight\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_RGB\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备RGB灯的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | RGB灯的状态参数SS |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备USB插座的状态SS

|  |
| --- |
| void hm\_get\_usbplug\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_UPLUG\_PARAM zb\_type)； |

功能：

此函数为获取子设备USB插座的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | USB插座的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备USB插座的SE参数

|  |
| --- |
| void hm\_set\_rgblight\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_RGB\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备USB插座的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | USB插座的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备USB插座的SE参数

|  |
| --- |
| void hm\_get\_rgblight\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_RGB\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备USB插座的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | USB插座的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备计量插座的状态SS

|  |
| --- |
| void hm\_get\_measureplug\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_SPLUG\_PARAM zb\_type)； |

功能：

此函数为获取子设备计量插座的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 计量插座的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备计量插座的SE参数

|  |
| --- |
| void hm\_set\_measureplug\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_SPLUG\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备计量插座的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 计量插座的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备计量插座的SE参数

|  |
| --- |
| void hm\_get\_measureplug\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_SPLUG\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备计量插座的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 计量插座的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备一路开关的状态SS

|  |
| --- |
| void hm\_get\_onewayswitch\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight1\_PARAM zb\_type)； |

功能：

此函数为获取子设备一路开关的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 一路开关的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备一路开关的SE参数

|  |
| --- |
| void hm\_set\_ai1sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight1\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备一路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 一路开关的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备一路开关的SE参数

|  |
| --- |
| void hm\_get\_ai1sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight1\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备一路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 一路开关的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备二路开关的状态SS

|  |
| --- |
| void hm\_get\_twowayswitch\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight2\_PARAM zb\_type)； |

功能：

此函数为获取子设备二路开关的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 二路开关的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备二路开关的SE参数

|  |
| --- |
| void hm\_set\_ai2sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight2\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备二路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 二路开关的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备二路开关的SE参数

|  |
| --- |
| void hm\_get\_ai2sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight2\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备二路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 二路开关的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备三路开关的状态SS

|  |
| --- |
| void hm\_get\_threewayswitch\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight3\_PARAM zb\_type)； |

功能：

此函数为获取子设备三路开关的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 三路开关的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备三路开关的SE参数

|  |
| --- |
| void hm\_set\_ai3sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight3\_PARAM zb\_type,  DEVINFOOPTION devOption); |

功能：

此函数为设置子设备三路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 三路开关的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备三路开关的SE参数

|  |
| --- |
| void hm\_get\_ai3sw\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_Slight3\_PARAM zb\_type,  DEVINFOOPTION devOption); |

功能：

此函数为获取子设备三路开关的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 三路开关的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备声光警号的状态SS

|  |
| --- |
| void hm\_get\_acoustoopticalarm\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_SAlARM\_PARAM zb\_type)； |

功能：

此函数为获取子设备声光警号的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 声光警号的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备声光警号的SE参数

|  |
| --- |
| void hm\_set\_sensoralarm\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_SAlARM\_PARAM zb\_type); |

功能：

此函数为设置子设备声光警号的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 声光警号的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备声光警号的SE参数

|  |
| --- |
| void hm\_get\_sensoralarm\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_SAlARM\_PARAM zb\_type); |

功能：

此函数为获取子设备声光警号的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 声光警号的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备四驱灯控的状态SS

|  |
| --- |
| void hm\_get\_fouredrivelamp\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_FOURLAMP\_PARAM zb\_type)； |

功能：

此函数为获取子设备四驱灯控的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 四驱灯控的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 设置子设备四驱灯控的SE参数

|  |
| --- |
| void hm\_set\_fouredrivelamp\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_FOURLAMP\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为设置子设备四驱灯控的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 四驱灯控的SE参数 |
| devOption | SE参数设置的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备四驱灯控的SE参数

|  |
| --- |
| void hm\_get\_fouredrivelamp\_separam(HM\_DEV\_RAM\_LIST \*pni,ZB\_FOURLAMP\_PARAM zb\_type,  DEVINFOOPTION devOption)； |

功能：

此函数为获取子设备四驱灯控的SE参数。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 四驱灯控的SE参数 |
| devOption | SE参数获取的标志位 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK 获取子设备振动探测器的状态SS

|  |
| --- |
| void hm\_get\_vibrationsensor\_state(HM\_DEV\_RAM\_LIST \*pni,ZB\_VIBRATION\_PARAM zb\_type)； |

功能：

此函数为获取子设备振动探测器的状态SS。

参数：

| **参数** | **说明** |
| --- | --- |
| pni | 查询到在网关链表的参数 |
| zb\_type | 振动探测器的状态参数SS |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

* 1. SDK wifi固件的OTA升级

|  |
| --- |
| void HMUpdataDeviceSoftware(LINK\_UPGRADE \*data)； |

功能：

此函数为wifi固件的OTA升级函数。

参数：

| **参数** | **说明** |
| --- | --- |
| LINK\_UPGRADE | Wifi OTA升级的参数 |

返回值：

| **值** | **说明** |
| --- | --- |
| --- | --- |

1. 返回值说明
   1. 返回值具体说明

返回值：

| **值** | **说明** |
| --- | --- |
| 0 | 失败 |
| 1 | 成功 |
| 2 | 连接服务器失败 |
| 3 | 参数错误 |