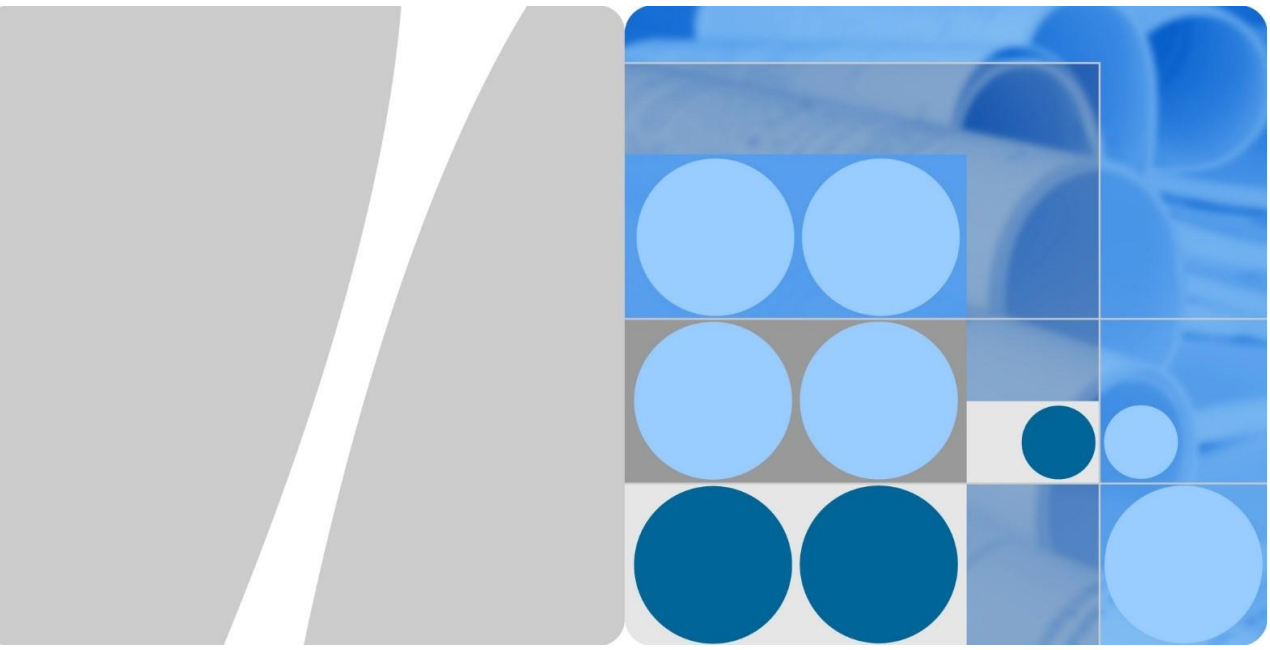


智能家居 HormonyOS Connect HiLink SDK

二次开发调测指导

文档版本 1.0

发布日期 2021-06-03

华为技术有限公司

HUAWEI

商标声明

 和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼

邮编： 518129

网址： http://www.huawei.com

客户服务邮箱： support@huawei.com

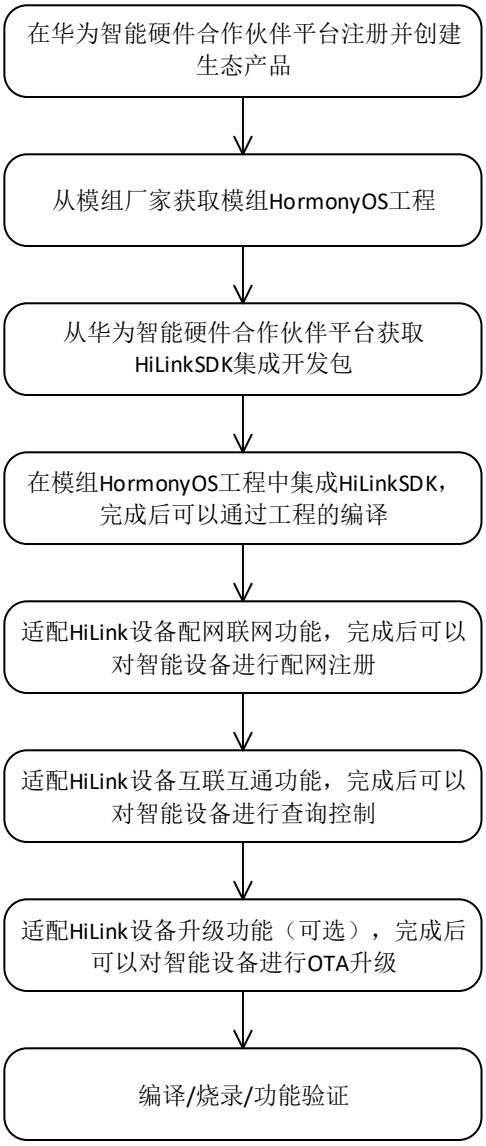
目录

概述

HiLink SDK 是运行在 HarmonyOS Connect 智能设备 WiFi 模组上的程序模块，用于实现设备的联网，以及设备与智能家居云和智慧生活 App 的互联互通。HiLink SDK 提供的功能包括设备联网、互联互通、设备升级、本地控制、错误诊断等。集成 HiLink SDK 可以帮助传统设备快速实现智能化。

本文档用于指导开发者在 HarmonyOS Connect 智能设备中集成和调测 HiLink SDK，实现和验证设备的远程控制、设备状态上报和 OTA 升级等功能。

HiLink SDK 的开发集成整体流程如下：



说明：开发者下载 SDK 开发包时，平台会生成一个 SHA-256 校验值，开发者可以使用这个值校验 SDK 开发包的完整性。

开发包结构介绍

华为智能硬件合作伙伴平台根据开发者提供的设备模型定义和模组型号生成设备的开发包，其中包括 HiLink SDK 和 KitFramework 两部分，HiLink SDK 相关内容结构及文件说明如下表：

目录	文件名	说明
hilinksdk |-include	hilink.h	HiLink SDK 静态库主头文件，包含 HiLinkSDK 入口函数和时间获取函数
	hilink_device.h	产品功能适配头文件，包含： • 设备信息、设备模型定义 • 待实现设备服务控制、查询响应函数声明 • 供调用设备服务状态上报函数声明
	hilink_ota.h	OTA 功能定义头文件
	hilink_osadapter.h	HiLink SDK 操作系统适配头文件，包含系统时间、Flash 读写、字符串操作、内存操作等系统函数适配接口声明
	hilink_log.h	HiLink SDK 系统提供的日志打印接口函数声明头文件
	hilink_typedef.h	系统类型定义头文件
	hilink_netconfig_mode_mgt.h	HiLink SDK 配网模式管理头文件
hilinksdk	hilink_device.c	产品功能适配源文件，包含设备模型相关待实现函数
	hilink_ota.c	HiLink 设备 OTA 升级功能实现函数
	hilink_device_sdk.c	根据设备模型生成的内部处理源代码，请勿修改
	Makefile	Makefile 样例文件，供开发时参考
hilinksdk |-lib	libhilinkdevicesdk.a	HiLink SDK 静态库文件 Debug 版本用于设备调测时集成 Release 版本用于商用发布时集成
	libhilinkota.a	HOTA 升级特性静态库文件
	libhilinkbtsdk.a	HiLink SDK BLE 静态库文件

说明：开发包中的代码文件(.c 和.h)中，中文注释采用 GB2313 编码，开发者查看代码时请将对应 IDE 开发环境编码格式调整为 GBK 格式。

集成准备

模组工程集成

解压 HiLink SDK 开发包到智能设备模组 HormonyOS 工程源代码目录。通过修改 gn 文件，将 HiLink SDK 开发包中的源代码文件（*.c）、头文件（*.h）和静态库文件（*.a）添加到智能设备程序编译中，如果 HormonyOS 工程已经集成 HiLink SDK 则只需替换相应的头文件（*.h）和静态库文件（*.a）以更新 HiLink SDK 版本。

开发者完成模组工程集成后，先编译固件版本，待编译可以通过后再继续完整后续集成和业务功能开发。

联网功能集成

联网功能是指智慧生活 App 扫描添加设备、HiLinkSVC 靠近发现设备、绑定设备到用户账号的功能。设备可以通过集成 HiLink SDK 的联网功能，实现连接路由网络、注册及登录智能家居云、添加到智慧生活 App。

设置 HiLink SDK 属性（可选）

参考 include\hilink.h 中结构体 HILINK_SdkAttr 的描述，使用 HILINK_SetSdkAttrs 接口设置 HiLink SDK 的属性；使用 HILINK_GetSdkAttr 接口查询当前 HiLink SDK 的属性。各个属性值的说明如下。

1) monitorTaskStackSize：HiLink SDK 监控任务栈大小，开发者根据具体情况调整，默认为 1024 字节，在异常监控任务中会调用重启前处理接口 hilink_process_before_restart、软重启接口 rebootSoftware 和硬重启接口 rebootHardware，若在以上接口的实现中使用了较大的栈空间，调整此接口；

2) deviceMainTaskStackSize：普通设备形态，HiLink SDK 运行主任务栈大小，开发者根据具体情况调整，如果当前设备形态是普通设备，可以使用此接口调整栈大小；

3) bridgeMainTaskStackSize：网桥设备形态，HiLink SDK 运行主任务栈大小，开发者根据具体情况调整，如果当前设备形态是网桥设备，可以使用此接口调整栈大小；

4) otaCheckTaskStackSize：HiLink OTA 检查升级版本任务栈大小，开发者根据具体情况调整，如果使用 HiLink 提供的 OTA 升级，可以使用此接口调整栈大小；

5) otaUpdateTaskStackSize：HiLink OTA 升级任务栈大小，开发者根据具体情况调整，如果使用 HiLink 提供的 OTA 升级，可以使用此接口调整栈大小；

6) rebootSoftware：HiLink SDK 异常、OTA 升级、设备删除等场景软重启接口，不影响设备硬件状态，如果用户注册此接口，首先使用；

7) rebootHardware：HiLink SDK 异常、OTA 升级、设备删除等场景硬重启接口，影响硬件状态，如果开发者未注册软重启接口，使用此接口重启。

示例：如果要调整 HiLink SDK 监控任务栈大小，由默认 1024 字节改为 2048 字节，该如何处理？

(1) 在调整该栈大小前先调用 HILINK_GetSdkAttr 函数接口获取当前各任务栈的大小；

(2) 调用 HILINK_SetSdkAttr 函数接口来设置要修改的任务栈的大小。

注：调用(1) (2)中两个函数都必须在 hilink_main 函数之前，否则不会生效。

```
void main(void)
{
    HILINK_SdkAttr *sdkAttr = NULL;
    /* 获取 HiLink SDK 当前属性值 */
    sdkAttr = HILINK_GetSdkAttr();
    if (sdkAttr == NULL) {
```

```
        printf("sdk attr is null\r\n");
        return;
    }

    /* 设置新的属性值 */
    sdkAttr->monitorTaskStackSize = 2048;
    HILINK_SetSdkAttr(*sdkAttr);

    /* 启动 HiLink 任务 */
    hilink_main();

    return;
}
```

设置配网方式（可选）

开发者可以设置设备要使用的配网方式(4G 或网线连网、WiFi 配网或其他配网)。开发者可以参考 include\hilink_netconfig_mode_mgt.h 中 HILINK_NetConfigMode 的描述，使用 HILINK_SetNetConfigMode 接口通知 HiLink SDK 设备使用的配网方式。若不设置，HiLink SDK 将默认使用 WiFi 配网方式。当前 HormonyOS 可用的配网模式说明如下。

1) HILINK_NETCONFIG_WIFI：WiFi 配网方式，选择该配网方式将会在在待配网状态时打开 SoftAp，并通过智慧生活 APP 扫描发现设备进行配网注册；

2) HILINK_NETCONFIG_OTHER：BLE 辅助配网方式，选择该配网方式将会在配网状态下打开蓝牙广播，通过 HiLinkSVC 弹窗或者智慧生活 APP 扫描发现设备进行配网注册，需要模组支持 BLE 并参考 4.2.2 蓝牙辅助配网进行适配；

3) HILINK_NETCONFIG_NAN_SOFTAP：WiFi 感知超短距配网和 SoftAp 组合，选择该配网方式将会使用 NAN 超短距协议进行配网，通过 HiLinkSVC 弹窗或者智慧生活 APP 扫描发现设备进行配网注册，并且支持在配网过程中进行秒控，目前仅 Hi3861 模组支持；

注：调用 HILINK_SetNetConfigMode 接口必须在 hilink_main 函数之前，否则不会生效。

wifi 靠近发现适配（可选）

在通过 WiFi 配网时，开发者可以适配 include\hilink.h 中获取设备表面最强发射功率接口 HILINK_GetDevSurfacePower 使用 wifi 靠近发现功能，最强点位置的确定以及功率测试方法，参照 hilink 认证 wifi 靠近发现功率设置及测试方法指导文档，如果厂商不想使用 wifi 靠近发现功能，接口直接返-1，如果需要使用 wifi 靠近发现，则接口返回 0，power 返回对应的功率值，单位为 dbm，目前仅 Hi3861 模组支持。

```
int HILINK_GetDevSurfacePower(char *power);
```

参数说明：

char *power：返回设备的表面最强点发射功率，有效值<=20;

返回值说明：-1 表示设备不支持 wifi 靠近发现，0 表示设备支持 wifi 靠近发现，power

返回对应的功率值

蓝牙辅助配网适配（可选）

在通过 BLE 辅助配网时，开发者需要适配 include\hilink_bt_api.h 中获取设备表面的最强点信号发射功率强度接口 HILINK_BT_GetDevSurfacePower，最强点位置的确定以及功率测试方法，参照 hilink 认证蓝牙靠近发现功率设置及测试方法指导文档，在使用 BLE 辅助配网时该接口必须适配。

```
int HILINK_BT_GetDevSurfacePower (char *power);
```

参数说明：

char *power：返回设备的表面蓝牙最强点发射功率;

返回值说明：-1 表示不支持 BLE 辅助配网，0 表示支持 BLE 辅助配网

在通过 BLE 辅助配网时，发者需要调用 include\ble_cfg_net_api.h 中的 BLE_CfgNetInit 接口对 BLE 辅助配网进行初始化，示例如下：

```
include "hilink_bt_api.h"
...
int hilink_ble_init(void){
        BLE_ConfPara isBlePair;
        isBlePair.isBlePair = 0;
        BLE_InitPara initPara;
        BLE_AdvInfo advInfo;
        initPara.confPara = &isBlePair;
        initPara.gattList = NULL;
        initPara.advInfo = NULL;
        BLE_CfgNetCb bleCfgNetCb = {NULL, NULL. NULL, NULL, NULL};
        int ret = BLE_CfgNetInit(&initPara, & bleCfgNetCb);
        if (ret != 0) {
                printf("ble cfg net init fail");
                return ret;
        }
 }
```

注：调用 BLE_CfgNetInit 接口必须在 hilink_main 函数之前，否则不会生效。

启动 HiLink SDK 主程序

在设备程序中直接调用 hilink_main 函数，启动 HiLink SDK 主流程，示例如下：

```
include "hilink.h"
void main(void)
{
        hilink_main();
}
```

注意：

(1) 由于 hilink_main()接口内部会创建线程，所以直接调用 hilink_main 接口即可，不要添加到循环或其他线程中调用。

(2) 开发者完成联网功能集成后，先不要继续下面的互联互通和升级功能集成，也不要添加设备业务功能的实现。开发者先编译固件版本，烧录验证设备联网功能（App 添加设备）是否正常。待验证联网功能正常后，再继续完整后续集成和业务功能开发。

互联互通功能集成

互联互通功能是 HiLink SDK 提供的设备与智慧生活 App 之间的命令交互、状态同步的功能。功能包括：智能设备响应华为智慧生活 App 的服务状态控制命令；智能设备上报服务状态给华为智慧生活 App；智能设备处理华为智慧生活 App 的服务状态查询命令。

集成互联互通功能，请参考如下步骤：

设备信息修改

本小节介绍开发者需要根据具体产品完成的相关信息的修改和配置。

版本信息

在 hilinksdk/include/hilink_device.h 文件中，默认版本号均设置为“1.0.0”，开发者可根据实际情况修改设备的固件版本、软件版本、硬件版本。例如：

```
#define FIRMWARE_VER "1.1.1"
#define SOFTWARE_VER "2.2.2"
#define HARDWARE_VER "3.3.3"
```

厂商及设备名称修改

hilinksdk/include/hilink_device.h 文件中有厂商和设备的名称定义的宏，请根据华为智能硬件合作伙伴平台上定义的设备信息进行修改，例如：

```
#define DEVICE_TYPE_NAME "Switch" // 设备类型名称
#define MANUAFACTURER_NAME "Huawei" // 设备厂商名称
```

HiLink SDK 会使用到厂商名称和设备类型名称来组装 SoftAP 的 SSID，开发者需要确保这两个字符串的长度之和不能超过 17 字符，否则会进行截断，导致显示不全。建议名称字符串使用简短单词或单词缩写来表示。

设备 SN 录入(可选)

设备默认使用 MAC 地址作为 SN 号，如果开发者需要根据实际情况录入 SN，可以实现 hilinksdk/hilink_device.c 中的 HilinkGetDeviceSn 接口，将 SN 数据传递给 HiLink SDK。示例如下

```
void HilinkGetDeviceSn(unsigned int len, char *sn)
```

参数说明：

(1) unsigned int len: SN 缓冲区长度，限定 SN 最大长度为 39 字节；

(2) char *sn: 录入的 SN 信息。

设备 PIN 适配

HiLink SDK 提供了 PIN 码配网功能，开发者可以实现 hilinksdk/hilink_device.c 中的 HiLinkGetPinCode 接口，将 PIN 码传递给 HiLink SDK。对于自动 PIN 码的设备返回-1 即可，对于手动 PIN 码的设备则需返回 8 位数字；

示例如下：

```
int HiLinkGetPinCode (void)
{
    return 12345678; // (调试用)
}
```

设备子型号适配(可选)

如果设备定义有子型号，则 HiLink SDK 在组装 SoftAp 的 SSID 阶段、向云端注册阶段、登录录后设备信息同步阶段，三个阶段均会使用设备子型号，开发者需要适配 hilink_device.c 中的如下接口，将设备的子型号 ID 传递给 HiLink SDK。

```
int HILINK_GetSubProdId(char *subProdId, int len);
```

参数说明:

(1) char *subProdId: 保存子型号的缓冲区;

(2) int len: 缓冲区的长度；

如果设备定义有子型号，则子型号长度固定为 2 字节，取值为十六进制的字符串表示，范围 00 ~FF，其中有字母则字母大写，并以'\0'结束，函数返回 0。

如果未定义有设备子型号，函数返回-1。

示例如下：

```
int HILINK_GetSubProdId(char *subProdId, int len)
{
    strcpy(subProdId, "0A");
    return 0;
}
```

设备控制业务开发

设备服务状态控制功能

开发者需要根据产品功能定义，在 hilink_device_sdk.c 中适配 hilink_put_char_state 函数。

设备服务状态查询功能

开发者需要根据产品功能定义，在 hilink_device_sdk.c 中适配 hilink_get_char_state 函数。

实现设备服务状态上报功能

开发者需要根据产品功能定义，在 hilink_device.c 中添加实现某些服务的状态上报接口。在设备状态主动发生变化，智慧生活 App 控制下发后设备状态没有立刻改变、过段时间完成更改后主动上报设备的状态。请开发者调用 include\hilink.h 中的 hilink_report_char_state 接口实现

设备重启前预处理功能

在 hilink_device.c 中定义了 hilink_process_before_restart 接口提供开发者实现。HiLink SDK 在异常、OTA 升级、设备删除等情况下，会请求重启 WiFi 模组，重启前会调用本接口。

开发者可以实现本接口完成模组重启前的一些必要操作，如数据备份等，以确保模组重启后的运行状态与重启前一致。其中返回值可以参考如下描述实现：

函数返回 0 表示处理成功，系统可以重启，使用硬重启接口重启;

函数返回 1 表示处理成功，系统可以重启，使用软重启（软重启接口需要开发者设置 HiLink SDK 属性提前适配并注册）;

函数返回负值表示处理失败，系统不能重启，HiLink SDK 将继续等待。

这里的参数 flag 表示了不同的重启原因：

当 flag 为 0 时，表示 HiLink SDK 线程看门狗触发导致即将重启，此时函数返回值由开发者根据具体业务状态决定;

当 flag 为 1 时，表示 App 删除设备即将重启，此时函数返回值必须为 0 或 1(允许重启)，否则将导致删除设备功能异常。

注意：请将该接口实现为非阻塞函数，建议该接口在 1 秒内返回，避免阻塞 SDK 主流程。

获取设备在线状态（可选）

开发者可以通过接口获取设备当前状态，当前支持状态列表如下：

```
/* 设备与云端连接断开(版本向前兼容) */
#define HILINK_M2M_CLOUD_OFFLINE 0
/* 设备连接云端成功，处于正常工作态(版本向前兼容) */
#define HILINK_M2M_CLOUD_ONLINE 1
/* 设备与云端连接长时间断开(版本向前兼容) */
#define HILINK_M2M_LONG_OFFLINE 2
/* 设备与云端连接长时间断开后进行重启(版本向前兼容) */
#define HILINK_M2M_LONG_OFFLINE_REBOOT 3
/* HiLink 线程未启动 */
```

```
#define HILINK_UNINITIALIZED                4
/* 设备处于配网模式 */
#define HILINK_LINK_UNDER_AUTO_CONFIG    5
/* 设备处于 10 分钟超时状态 */
#define HILINK_LINK_CONFIG_TIMEOUT        6
/* 设备正在连接路由器 */
#define HILINK_LINK_CONNECTTING_WIFI      7
/* 设备已经连上路由器 */
#define HILINK_LINK_CONNECTED_WIFI        8
/* 设备正在连接云端 */
#define HILINK_M2M_CONNECTTING_CLOUD      9
/* 设备与路由器的连接断开 */
#define HILINK_M2M_CLOUD_DISCONNECT       10
/* 设备被注册 */
#define HILINK_DEVICE_REGISTERED          11
/* 设备被解绑 */
#define HILINK_DEVICE_UNREGISTER          12
/* 设备复位标记置位 */
#define HILINK_REVOKE_FLAG_SET            13
/* 设备协商注册信息失败 */
#define HILINK_NEGO_REG_INFO_FAIL         14
```

（1）状态查询接口

hilinksdk/include/hilink.h 中声明了 int hilink_get_devstatus(void)函数，函数返回设备当前状态，值对应上面不同的状态。开发者可以调用本接口查询当前设备状态。

```
int devStatus = hilink_get_devstatus();
```

（2）状态变化回调函数

hilinksdk/hilink_device.c 中定义了虚函数 hilink_notify_devstatus(int status)。HiLink SDK 内部在设备状态变化时会调用本接口将状态通知给开发者。开发者可以实现这个接口，根据参数 status 的值，添加相应状态下的业务处理。

```
void hilink_notify_devstatus(int status)
{
    switch(status) {
        case HILINK_M2M_CLOUD_OFFLINE:
            // 设备与云端连接断开，请在此处添加实现
            break;
        case HILINK_M2M_CLOUD_ONLINE:
            // 设备连接云端成功，请在此处添加实现
            break;
        …
        default:
            break;
    }
}
```

设备恢复出厂设置（可选）

用户手动操作设备(如按键、长按、组合操作等)进行恢复出厂设备，此时需要清除设备 Flash 存储的配网信息及华为智能家居云记录的设备绑定信息，使设备重新进入配网状

态。

开发者实现上述功能，可以在手动操作触发恢复出厂的接口中，调用如下接口实现。

```
hilink_restore_factory_settings();
```

本接口声明在 hilinksdk/include/hilink.h 文件中。

存取设备状态配置（可选）

HiLink SDK 提供了接口供开发者将一些简单状态信息保存到 Flash，Flash 空间大小 32 字节。对没有电控板 MCU 的设备(业务功能实现在模组上)，可调用下面的接口保存或获取 Flash 中的设备状态。

需要注意：

1)App 删除设备或设备手动恢复出厂后，保存的状态会被清除。

2)接口仅提供数据存储 Flash 的能力，不对数据进行加密。若用户保存重要信息请自行加密后再调用本接口保存。

存取接口均声明在 hilinksdk/include/hilink.h 文件中。

（1）保存设备状态

调用以下函数保存设备状态配置到模组 Flash，最长 32 字节。注意：每次保存都会覆盖之前旧的内容。

```
int HilinkSetUserConfig(const char* config, unsigned short len);
```

使用示例：

```
char inConfig[32] = { /* 用户配置或状态 */ };
int ret = HilinkSetUserConfig(inConfig, 32);
if (ret == 0) {
    /* 保存成功 */
} else {
    /* 保存失败 */
}
```

（2）获取用户设备状态

调用以下函数获取之前保存的设备状态配置，最长 32 字节。

```
int HilinkGetUserConfig(unsigned short len, char* config);
```

参数 config 是出参，调用者需要预先分配好内存，获取到的设备状态配置会被保存到 config 对应的内存中。使用示例：

```
char outConfig[32] = {0};
int ret = HilinkGetUserConfig(32, outConfig);
if (ret == 0) {
    /* 获取成功，信息被获取到 outConfig 中 */
} else {
    /* 获取失败 */
}
```

设备离线场景 App 删除设备 SDK 处理适配（可选）

设备离线时，如果在 App 上删除了该设备，设备下次上线后云端会给设备下发 Errcode=5(登录错误)或 Errcode=6(设备已被删除)错误码。此时设备接收到这两个错误码后，会根据接口 HILINK_EnableProcessDelErrCode (enable)的设置结果来判断是否需要清除注册信息进入配网。

enable 为 0 表示 SDK 不处理云端下发的 Errcode=5 或 Errcode=6 错误码，此时 SDK 不会清除设备端注册信息，需要用户手动硬件恢复出厂设置，设备才能重新进入配网状态。

enable 为非 0 表示 SDK 处理云端下发的 Errcode=5 或 Errcode=6 错误码，此时 SDK 会清除设备端注册信息，并重新进入配网状态。

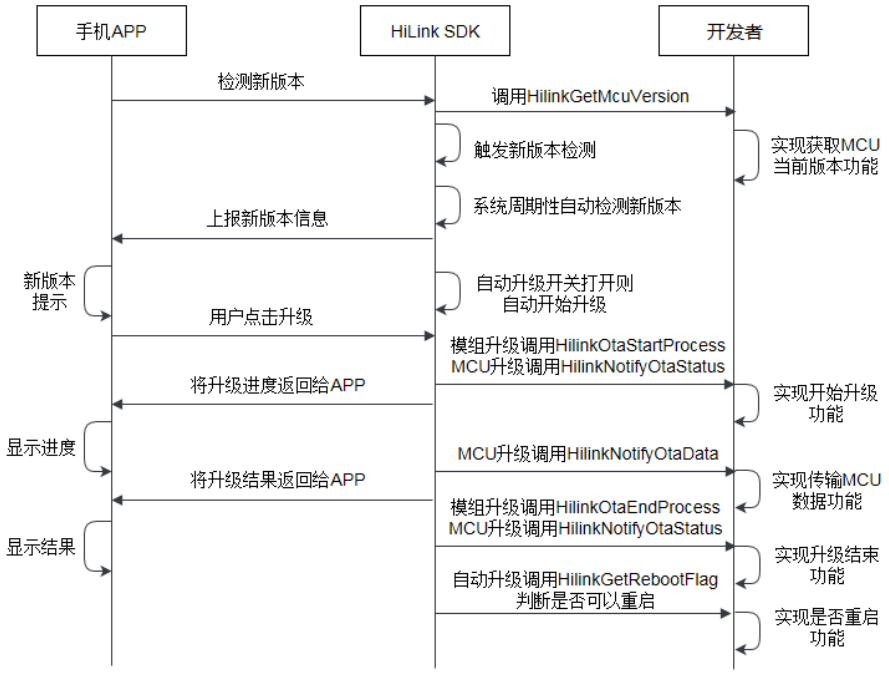
如果不设置，默认 enable 为 1。

如果设备希望设计为 APP 删除后不马上进行本地清除注册的场景，可以将本接口设置为 0。

至此，HiLink SDK 的基础功能集成就完成了，此时开发者可以编译固件，烧录版本，通过 App 添加设备、控制设备，查看日志打印，验证基本功能（配网、注册、控制、上报等）和状态流程是否正常。

华为 HOTA 升级功能集成（可选）

本小节介绍使用华为 HOTA 升级功能的模组，适配升级功能的实现。目前支持 HiLinkSDK 的模组都具备支持华为 HOTA 升级的能力。升级分为模组升级和电控板 MCU 升级。HiLink 设备的升级通过华为智慧生活 App 触发或者用户打开自动升级功能自动触发。HiLink SDK 模组升级已由模组厂商适配，而 MCU 升级业务，开发者只需要实现几个接口，整体流程如下图所示：



实现升级接口函数

实现 hilinksdk/hilink_ota.c 中升级接口函数：

1. 获取 MCU 当前版本信息函数

实现获取 MCU 当前版本号 HilinkGetMcuVersion (char *version, unsigned int inLen, unsigned int *outLen)，开发者实现获取 MCU 的当前版本号。

如果获取不到 MCU 的版本，则不对 MCU 进行升级。建议开发者在 MCU 正常启动后，或升级启动后，就将 MCU 的版本号传递给模组，确保模组可以获取到 MCU 的版本。如果没有 MCU 的返回 RETURN_ERROR_NO_MCU，获取成功则返回 RETURN_OK，获取失败则返回 RETURN_ERROR。

2. 模组开始升级函数

实现模组开始升级函数 HilinkOtaStartProcess (int type)，开发者可在此函数中实现升级开始时需要添加的功能。

在手动升级场景下，HiLink SDK 在接收到用户发出的升级指令后，将直接调用此接口。

开机后 10 分钟到 1 小时内随机时间检测一次是否有新版本，之后以当前时间为起点，23 小时加 1 小时内随机值周期性检测新版本。如果用户打开了自动升级开关，检测到有新版本并且是可以重启的情况下，就进行新版本的下载，下载完成后自动重启。在自动升级场景下，当 HiLink SDK 在调用 HilinkGetRebootFlag 接口返回值是

MODULE_CAN_REBOOT 时，HiLink SDK 将调用此接口。厂商可在此接口中完成和升级流程相关的处理。如果处理正常就返回 RETURN_OK，处理异常请返回 RETURN_ERROR。

3. 模组结束升级函数

实现模组结束升级函数 HilinkOtaEndProcess (int status)，开发者可在此函数中实现在升级结束时需要添加的功能。

HiLink SDK 在将固件写入到 OTA 升级区后，且完整性校验通过后，将调用厂商适配的此接口；厂商可在此接口中完成和升级流程相关的处理。开机后 10 分钟到 1 小时内随机时间检测一次是否有新版本，之后以当前时间为起点，23 小时加 1 小时内随机值周期性检测新版本。如果用户打开了自动升级开关，检测到有新版本并且是可以重启的情况下，就进行新版本的下载，下载完成后自动重启；升级类型是否为自动升级可参考接口 HilinkOtaStartProcess 的参数 type 的描述。如果处理正常就返回 RETURN_OK，处理异常请返回 RETURN_ERROR。

4. 通知 MCU 升级状态函数

实现通知 MCU 升级状态函数 HilinkNotifyOtaStatus (int flag, unsigned int len, unsigned int type)，开发者可在此函数中实现 MCU 升级状态改变时需要添加的功能。没有 MCU 可不用实现此函数。

HiLink SDK 调用开发者适配的此接口通知 MCU 固件传输的状态。当 flag 是 STOP_SEND_DATA 时，此接口需返回 MCU 侧固件升级的结果；当 flag 是其它值时，需返回接口接收到此消息后的处理结果。开机后 10 分钟到 1 小时内随机时间检测一次是否有新版本，之后以当前时间为起点，23 小时加 1 小时内随机值周期性检测新版本。如果用户打开了自动升级开关，检测到有新版本并且是可以重启的情况下，就进行新版本的下载，下载完成后自动重启。

5. MCU 数据传输函数

实现 MCU 数据传输函数 HilinkNotifyOtaData(unsigned char *data, unsigned int len, unsigned int offset)，开发者可在此函数中实现传输 MCU 固件数据给 MCU 的功能。没有 MCU 可不用实现此函数。

HiLink SDK 调用开发者适配的此接口通知开发者发送 MCU 固件文件数据。HiLink SDK 通知发送 MCU 固件文件时，将 MCU 固件文件拆分成若干个数据包通知给模组。开发者适配的此接口需要返回 MCU 接收这部分数据的处理结果。当此接口返回 RETURN_OK 时，HiLink SDK 继续正常处理后续流程；当此接口返回 RETURN_ERROR 时，HiLink SDK 将终止此次的 MCU 固件升级。如果处理正常就返回 RETURN_OK，处理异常请返回 RETURN_ERROR。

6. 判断模组是否能重启函数

实现判断模组是否能重启函数 HilinkGetRebootFlag(void)，开发者可在此函数中实现重启前保存数据之类的功能。

在用户同意设备可以自动升级的情况下，HiLink SDK 调用此接口获取设备当前业务状态下，模组是否可以立即升级并重启的标志。只有当设备处于业务空闲状态时，接口才可以返回 MODULE_CAN_REBOOT。当设备处于业务非空闲状态时，接口返回 MODULE_CANNOT_REBOOT。

部署华为 HOTA 服务器的升级包格式

HOTA 服务器部署升级包需将编译后生成的 OTA 固件打包给华为相关支撑人员进行版本的 HOTA 服务器部署。

功能验证

概述

根据集成开发包中的认证测试用例《HiLink 智能家居解决方案 SDK 基本功能测试用例》验证智能设备基本功能。

根据产品定义的功能，验证业务功能。使用配套的华为智慧生活 App（实现该设备对应的添加、注册、控制功能）与设备共同验证。

特别注意，请关闭手机应用市场中华为智慧生活 App 的自动升级功能。该 App 为测试专用 App，请勿升级至其他版本，否则 App 无法进行正常调试工作。

App 调试环境设置

1. 运行华为智慧生活 App，可以直接选择商用环境、认证环境和开发测试环境。

1.1 可根据自己需求自定义环境类型，选择自定义，选择家居云服务器，选择“厂家认证云”。

1.2 选择音箱云服务器，任选一个即可，点击“确认”按钮。

2. 阅读相关用户协议和隐私声明，勾选加入用户体验计划（也可后面在 App 中打开），点击“同意”按钮。

3. 允许 App 获取的权限。

4. 登录华为账号（若之前手机已登录过，App 会自动登录），观看使用引导。

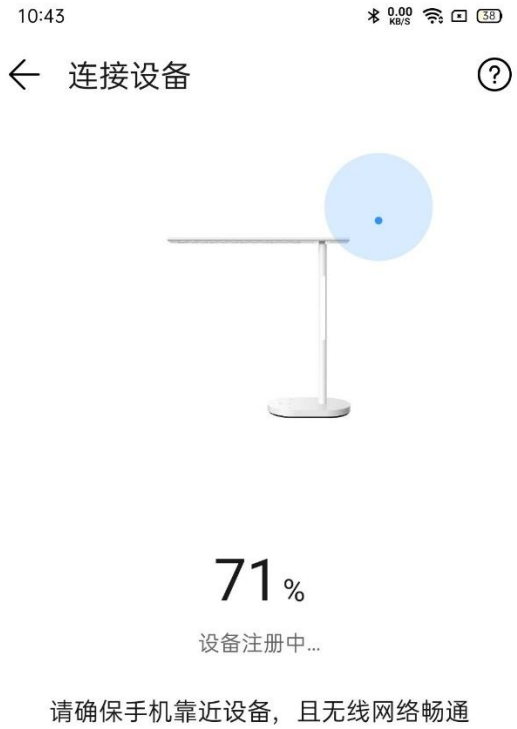
搜索添加待测设备

1. 在智慧生活 App“家居”页面，点击右上角的“+”号按钮，选择“添加设备”。

2. 华为智慧生活 App 开始自动扫描周围设备。

3. 选择扫描出的对应设备，点击“连接”按钮，进入配网流程。

4. 在“连接设备”页面输入手机当前连接的 WLAN 热点的密码，单击“下一步”，设备开始自动连网注册。



说明

1、如果设备实现了定制 PIN 码配网，还需要手动输入 PIN 码。

5. 在“设备设置”页面，设置设备名称及选择设备所属房间。如果设备需要使用自动升级功能，可以打开自动升级开关。点击“完成”按钮，设备即添加成功。

6. 设备添加完成后，可在华为智慧生活 App“家居”页面查看到已添加设备。点击设备图标进入设备页面，即可完成设备功能的控制等操作。

验证设备控制功能

1. 进入设备页面，操作相关控件，控制设备功能。
2. 查看设备状态是否按预期改变，以及 App 状态与设备侧状态是否一致。

具体页面和操作，与具体设备类型和功能相关，此处不再赘述。