



GR551x鼠标示例手册

版本： 1.8

发布日期： 2021-08-09

版权所有 © 2021 深圳市汇顶科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得对本手册内的任何部分擅自摘抄、复制、修改、翻译、传播，或将其全部或部分用于商业用途。

商标声明

GOODIX 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人持有。

免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

深圳市汇顶科技股份有限公司（以下简称“GOODIX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODIX对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经GOODIX书面批准，不得将GOODIX的产品用作生命维持系统中的关键组件。在GOODIX知识产权保护下，不得暗或以其他方式转让任何许可证。

深圳市汇顶科技股份有限公司

总部地址：深圳市福田区保税區腾飞工业大厦B座2层、13层

电话：+86-755-33338828 传真：+86-755-33338099

网址：www.goodix.com

前言

编写目的

本文档主要介绍Human Input Device（HID）Service、GR551x鼠标示例的初次运行和应用详解，旨在帮助用户快速进行二次开发。

读者对象

本文适用于以下读者：

- GR551x用户
- GR551x开发人员
- GR551x测试人员
- GR551x技术支持工程师
- 文档工程师

版本说明

本手册为第6次发布，对应的产品系列为GR551x。

修订记录

版本	日期	修订内容
1.0	2019-12-08	首次发布
1.3	2020-03-16	更新文档中页脚的时间
1.5	2020-05-30	更新“启动过程”章节中的流程图
1.6	2020-06-30	基于SDK刷新版本
1.7	2021-04-20	优化“初次运行”和“应用详解”章节
1.8	2021-08-09	更新“准备工作”章节

目录

前言.....	1
1 简介.....	1
2 HID over GATT Profile.....	2
2.1 设备角色.....	2
2.2 HID Service.....	2
2.3 安全要求.....	3
3 初次运行.....	4
3.1 准备工作.....	4
3.2 固件烧录.....	4
3.3 测试验证.....	4
4 应用详解.....	6
4.1 运行流程.....	6
4.2 关键代码.....	6
4.2.1 初始化HID Service.....	6
4.2.2 安全参数配置.....	7
4.2.3 发送按键请求.....	8
4.2.4 连接断开.....	9

1 简介

GR551x鼠标示例实现了人体学输入设备（Human Input Device，HID）固件示例。

在进行操作前，可参考以下文档。

表 1-1 文档参考

名称	描述
GR551x开发者指南	GR551x软硬件介绍、快速使用及资源总览
J-Link用户指南	J-Link的使用说明： www.segger.com/downloads/jlink/UM08001_JLink.pdf
Keil用户指南	Keil的详细操作： www.keil.com/support/man/docs/uv4/
Bluetooth Core Spec	Bluetooth官方标准核心规范
Bluetooth GATT Spec	Bluetooth Profile和Service的详细信息查看地址： www.bluetooth.com/specifications/gatt
GProgrammer用户手册	GProgrammer软件的操作使用说明，包括GR551x的固件下载、固件加密等。

2 HID over GATT Profile

本章主要介绍HID over GATT Profile（HOGP）定义的设备角色、HID Service以及安全要求。

2.1 设备角色

HOGP定义的设备角色包括HID Device和HID Host。

- HID Device

人体学输入设备，承担GAP Peripheral角色，作为GATT服务端。常见的HID Device有鼠标、键盘等。

HID Device必须包含至少一个HID Service实例、一个电池服务（Battery Service，BAS）实例，一个设备信息服务（Device Information Service，DIS）实例，以及一个可选的Scan Parameters Service实例。HID Device也可以包含一个或多个其他类型GATT Service的实例，但是这些GATT Service的实例并不作为HOGP的一部分。

GR551x SDK中实现HID Device的ble_app_hids_mouse示例包含一个HID Service实例、一个BAS实例和一个DIS实例。

- HID Host

解析HID Device发送的输入数据，承担GAP Central角色，作为GATT客户端。例如Android手机等。HID Host负责扫描、连接与配置HID Device；在建立连接后，可以对HID Device进行数据接收、读取和写入操作。

2.2 HID Service

HID Service将HID Device上的数据和与之关联的格式（由[USB HID Specification](#)定义）呈现给HID Host。

HID Service的Characteristics提供了对HID数据的访问，对Characteristic的详细描述参考表 2-1。

表 2-1 HID Service的Characteristics

Characteristic	UUID	Type	Support	Security	Properties
Protocol Mode	2A4E	16 bits	Mandatory for Boot Protocol Mode support	None	Read, Write
Report	2A4D	16 bits	Mandatory to support at least one Report Type if the Report characteristic is supported	None	Read, Notify, Write
					Read, Write, Write Without Response
					Read, Write
Report Map	2A4B	16 bits	Mandatory	None	Read
Boot Keyboard Input Report	2A22	16 bits	Mandatory for keyboards	None	Read, Notify, Write

Characteristic	UUID	Type	Support	Security	Properties
Boot Keyboard Output Report	2A32	16 bits	Mandatory for keyboards	None	Read, Write, Write Without Response
Boot Mouse Input Report	2A33	16 bits	Mandatory for mice	None	Read, Notify, Write
HID Information	2A4A	16 bits	Mandatory	None	Read
HID Control Point	2A4C	16 bits	Mandatory	None	Write Without Response

- Protocol Mode特性：用户可以通过该特性设置或读取HID Service当前的协议模式；
- Report特性：用于在HID Device和HID Host间交换数据；
- Report Map特性：用于定义HID Device和HID Host间传输的Input Report、Output Report和Feature Report数据的格式信息；
- Boot Keyboard Input Report/Boot Keyboard Output Report特性：当HID Host运行于Boot协议模式下时，可以通过Boot Keyboard Input Report特性或Boot Keyboard Output Report特性和一个与Boot Keyboard相对应的HID Device传输固定格式和长度的Input Report或Output Report数据；
- Boot Mouse Input Report特性：当HID Host运行于Boot协议模式下时，可以通过Boot Mouse Input Report特性和一个与Boot Mouse相对应的HID Device传输固定格式和长度的Input Report数据。
- HID Information特性：用于保存一组称为HID Device的HID属性的值；
- HID Control Point特性：是一个控制点属性，用户可通过它定义暂停或退出暂停的HID Command。

2.3 安全要求

LE Security Mode 1包含Security Level 2和3。

- Security Level 2: Encrypted Link required; MITM protection not necessary.
- Security Level 3: MITM-protected encrypted link required.

根据[HOGP Specification](#)的规定，HID Device必须支持以上任意一种安全等级（Security level）：

- HID Service包含的characteristics的Security Property必须被设置为LE Security Mode 1中上述两种Security Level之一。
- Device Information Service， Scan Parameters Service和Battery Service包含的characteristics也建议被设置为同样的Security Mode和Security Level。

GR551x鼠标示例按照上述要求在gap_params_init()中设置Security parameters，请参考[4.2.2 安全参数配置](#)。

3 初次运行

本章主要介绍如何运行和验证GR551x鼠标示例。

说明:

SDK_Folder为GR551x SDK的根目录。

3.1 准备工作

运行鼠标示例之前，请完成以下准备工作。

- 硬件准备

表 3-1 硬件准备

名称	描述
开发板	GR5515 Starter Kit开发板（以下简称开发板）
安卓手机	Android 5.0（KitKat）及以上版本
连接线	Micro USB 2.0数据线

- 软件准备

表 3-2 软件准备

名称	描述
Windows	Windows 7/Windows 10操作系统
J-Link Driver	J-Link驱动程序，下载网址： www.segger.com/downloads/jlink/
Keil MDK5	IDE工具，支持MDK-ARM 5.20 及以上版本，下载网址： www.keil.com/download/product/
GProgrammer（Windows）	Programming工具，位于SDK_Folder\tools\GProgrammer

3.2 固件烧录

鼠标示例工程的源码位于SDK_Folder\projects\ble\ble_peripheral\ble_app_hids_mouse。

用户可使用GProgrammer将ble_app_hids_mouse_fw.bin烧录至开发板。GProgrammer烧录固件的具体操作方法，请参考《GProgrammer用户手册》。

说明:

- ble_app_hids_mouse_fw.bin位于SDK_Folder\projects\ble\ble_peripheral\ble_app_hids_mouse\build\。
- GProgrammer位于SDK_Folder\tools\GProgrammer。

3.3 测试验证

使用Android手机即可测试验证鼠标示例。

1. 按下开发板上的“RESET”键，开发板进入广播状态。
2. 进入手机系统设置的蓝牙界面，打开蓝牙开关。等待手机搜索到名为“Goodix_Mouse”的蓝牙设备。
3. 在手机蓝牙界面点击“Goodix_Mouse”连接该设备。
4. 在手机弹出的配对对话框中输入pin code “123456”。

在配对成功后，名为“Goodix_Mouse”的设备会出现在手机蓝牙界面的“Paired devices”（已配对设备列表）中，并显示状态为“Connected”（已连接）。如图 3-1所示，用户长按开发板上的按键“UP”、“DOWN”、“LEFT”、“RIGHT”，可以移动鼠标箭头。

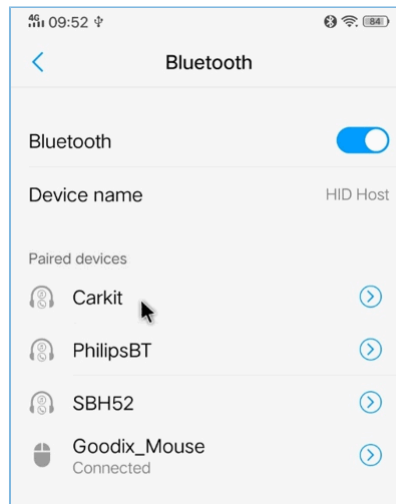


图 3-1 手机屏幕鼠标箭头示例

该示例还支持媒体播放控制键。用户双击“UP”键，增加音量；双击“DOWN”键，减小音量；双击“OK”键，暂停/继续播放；双击“RIGHT”键，切换下一首音乐；双击“LEFT”键，切换上一首音乐。需要注意，部分Android手机由于操作系统的原因，不能完全支持媒体播放控制键。

4 应用详解

本章将介绍鼠标示例的运行流程和关键代码。

4.1 运行流程

鼠标示例的运行流程可以分为两个阶段：启动流程和交互处理流程。具体运行流程如下图所示。

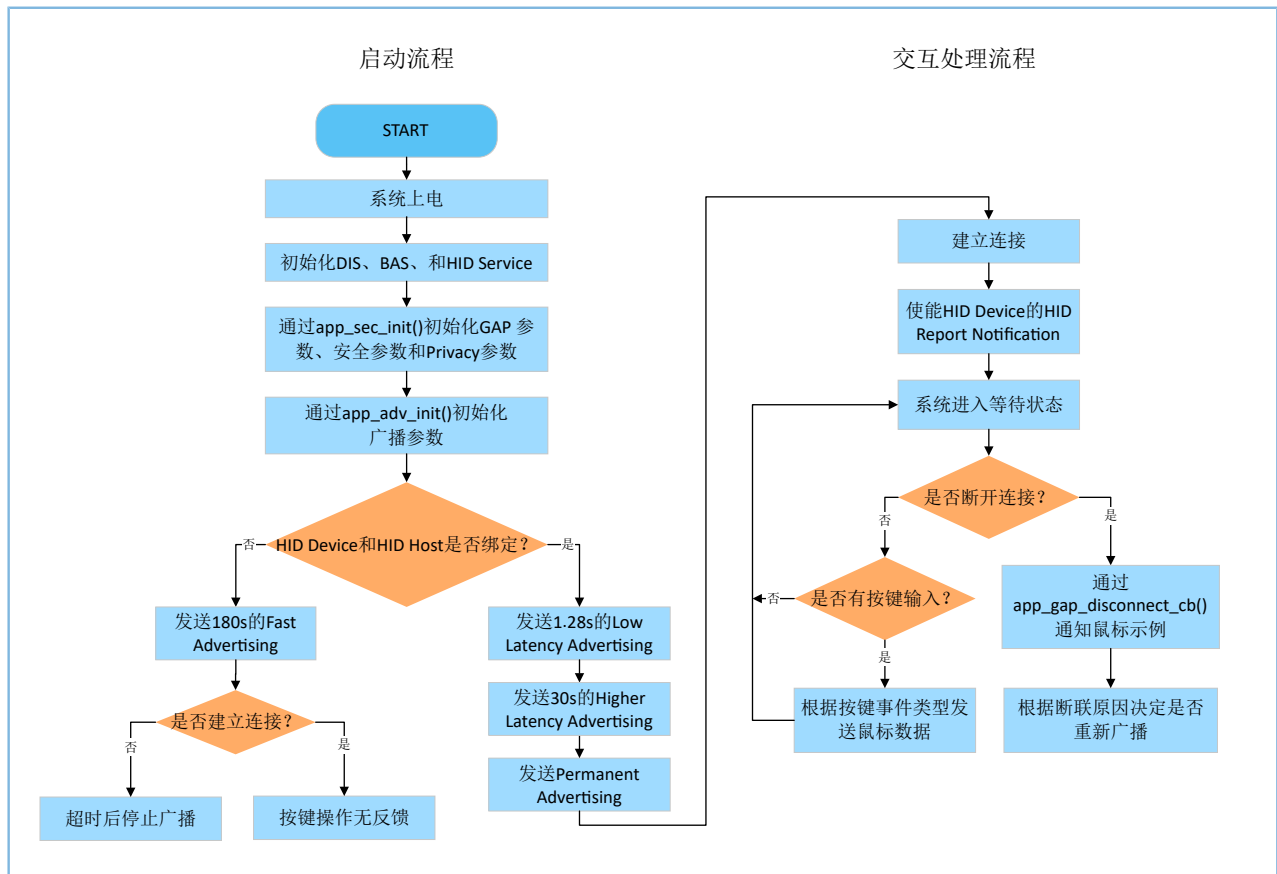


图 4-1 鼠标示例运行流程

启动流程中需要注意，HID Device与HID Host之间的绑定关系将影响Advertising的参数。详见[HOGP Specification](#)中“Section 5 Connection Establishment”。

说明:

`void adv_sec_init(bool erase_bond)`的入参如果为“true”，将导致每次重启设备后，绑定信息被清除。

4.2 关键代码

鼠标示例的业务逻辑代码位于Keil工程目录下的如下文件中。

4.2.1 初始化HID Service

`hids_init()`函数负责对HIDS Service的初始化参数进行配置，初始化参数中包含了`rep_map_data[]`，该数组按照[USB HID Specification](#)规定的格式，定义了包含鼠标Report和多媒体按键Report的Report Map。

路径：工程目录下的user_app\user_mouse.c

名称：hids_init();

```
static void hids_init(void)
{
    hids_init_t hids_init;

    hids_init.report_map.p_map = (uint8_t*)&rep_map_data;
    hids_init.report_map.len   = sizeof(rep_map_data);

    .....

    hids_service_init(&hids_init);
}
```

📖 说明:

static uint8_t rep_map_data[]数组的长度不能超过hids.h中定义的REPORT_MAP_MAX_SIZE。

4.2.2 安全参数配置

为了满足HOGP Specification所规定的相关安全要求，app_sec_init()函数设置了如下安全参数。关于sec_param_t的设置，请参考《GR551x BLE Stack用户指南》“安全管理（SM）”。

app_sec_init()函数使用ble_gap_privacy_params_set()开启了隐私模式，在此模式下，鼠标示例会每隔PRIVACY_RENEW_DURATION秒重新生成设备地址，且不响应已绑定的HID Host使用Public Address发起的Connect请求。其中PRIVACY_RENEW_DURATION也在user_app.c中被定义。

路径：工程目录下的user_app\user_app.c

名称：app_sec_init();

```
static void app_sec_init(bool erase_bond)
{
    .....

    error_code = ble_gap_privacy_params_set(PRIVACY_RENEW_DURATION, true);
    APP_ERROR_CHECK(error_code);

    //set the default security parameters.
    sec_param_t sec_param =
    {
        .level   = SEC_MODE1_LEVEL3,
        .io_cap  = IO_DISPLAY_ONLY,
        .oob     = false,
        .auth    = AUTH_BOND | AUTH_MITM | AUTH_SEC_CON,
        .key_size = 16,
        .ikey_dist = KDIST_ALL,
        .rkey_dist = KDIST_ALL,
    };
    error_code = ble_sec_params_set(&sec_param);
}
```

```
APP_ERROR_CHECK(error_code);
}
```

`user_sm_callback.c`中的`app_sec_rcv_enc_req_cb()`函数用于响应HID Host发起的配对加密请求。在[3.3 测试验证](#)中，手机上输入的Pin Code即为该函数中设置的`tk`变量。

路径：工程目录下的`user_callbacks\user_sm_callback.c`

名称：`app_sec_rcv_enc_req_cb()`;

```
static void app_sec_rcv_enc_req_cb(uint8_t conn_idx, sec_enc_req_t *p_enc_req)
{
    ...
    switch (p_enc_req->req_type)
    {
        .....
        case TK_REQ:
            APP_LOG_INFO("Please Input pin code: 123456");
            cfm_enc.req_type = TK_REQ;
            cfm_enc.accept = true;
            tk = 123456;
            memset(cfm_enc.data.tk.key, 0, 16);
            cfm_enc.data.tk.key[0] = (uint8_t)((tk & 0x000000FF) >> 0);
            cfm_enc.data.tk.key[1] = (uint8_t)((tk & 0x0000FF00) >> 8);
            cfm_enc.data.tk.key[2] = (uint8_t)((tk & 0x00FF0000) >> 16);
            cfm_enc.data.tk.key[3] = (uint8_t)((tk & 0xFF000000) >> 24);
            break;
    }
    ble_sec_enc_cfm(conn_idx, &cfm_enc);
}
```

关于处理配对加密请求，请参考《GR551x BLE Stack用户指南》“开启绑定功能”。

4.2.3 发送按键请求

当用户使用开发板进行按键类操作时，`app_key_evt_handler()`会接收到来自BSP层的按键事件，并调用HIDS模块的`hids_input_rep_send()`接口将鼠标数据发送给HID Host。鼠标数据分为两类：`mouse_data_t[]`和`media_data_t[]`。

路径：工程目录下的`gr_profiles\hids.c`

名称：`hids_input_rep_send()`;

```
sdk_err_t hids_input_rep_send(uint8_t conn_idx, uint8_t rep_idx,
                             uint8_t *p_data, uint16_t length)
{
    static const uint8_t char_idx[] = {HIDS_IDX_INPUT1_REPORT_VAL,
                                       HIDS_IDX_INPUT2_REPORT_VAL, HIDS_IDX_INPUT3_REPORT_VAL};
    sdk_err_t error_code = SDK_ERR_NTF_DISABLED;
    if(rep_idx >= IN_REPORT_MAX_COUNT || p_data == NULL || length == 0)
    {
        return SDK_ERR_INVALID_PARAM;
    }
}
```

```

}
length = ((length > HIDS_REPORT_MAX_SIZE) ? HIDS_REPORT_MAX_SIZE : length);
memcpy(&s_hids_env.input_report_val[rep_idx], p_data, length);
if(s_hids_env.input_cccd[rep_idx][conn_idx] == PRF_CLI_START_NTF)
{
    error_code = hids_in_rep_notify(conn_idx, char_idx[rep_idx], p_data, length);
}
return error_code;
}

```

路径：工程目录下的user_app\use_mouse.h

名称：mouse_data_t[]和media_data_t;

```

typedef struct
{
    bool left_button_press;
    bool middle_button_press;
    bool right_button_press;
    int8_t x_delta;
    int8_t y_delta;
    int8_t wheel_delta;
} mouse_data_t;

typedef struct
{
    uint8_t play_pause:1;
    uint8_t al_control:1;
    uint8_t next_track:1;
    uint8_t previous_track:1;
    uint8_t volume_down:1;
    uint8_t volume_up:1;
    uint8_t ac_foward:1;
    uint8_t ac_back:1;
} media_data_t;

```

4.2.4 连接断开

HID Host与HID Device之间的连接断开后，BLE协议栈会把该事件通过gap_cb_fun_t::app_gap_disconnect_cb()通知给鼠标示例。ble_adv_disconnected()会根据连接断开的原因，决定是否重新开始广播。

[HOGP Specification](#)规定，如果连接由于Link Loss而断开，HID Device需重新开始广播。但是为方便测试，对于Remote User Terminated Connection的原因，鼠标示例也会重新开始广播。如果已经绑定过，鼠标示例会依次开始发送Low Latency Advertising，Higher Latency Advertising和Permanent Advertising。

路径：工程目录下ble_module\ble_advertising.c

名称：ble_adv_disconnected();

```

static void ble_adv_disconnected(void)

```

```
{
    if (adv_env.adv_mode_cfg.adv_on_disconnect_enabled && !adv_env.adv_act_exist)
    {
        ble_advertising_start(BLE_ADV_MODE_DIRECTED_HIGH_DUTY);
    }
}
```