



## GR551x鼠标示例手册

版本： 1.6

发布日期： 2020-06-30

版权所有 © 2020 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](http://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4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，对应GR551x_SDK_1.0.0
1.3	2020-03-16	更新了文档中页脚的时间
1.5	2020-05-30	更新了“4.2.1 启动过程”章节中的流程图
1.6	2020-06-30	基于SDK刷新版本

# 目录

前言.....	I
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 硬件连接.....	4
3.3 下载固件.....	5
3.4 测试验证.....	5
4 应用详解.....	6
4.1 工程目录.....	6
4.2 运行流程.....	6
4.2.1 启动过程.....	7
4.2.2 已与HID Host连接.....	7
4.2.3 已与HID Host断开.....	7
4.3 关键代码.....	8
4.3.1 电池模拟模块的初始化.....	8
4.3.2 HID Service的初始化.....	9
4.3.3 安全参数配置.....	9

## 1 简介

GR551x鼠标示例基于GR551x SDK和GR5515 Starter Kit开发板实现人体学输入设备（Human Input Device，HID）固件示例。

修改和使用鼠标示例前，建议参考表 1-1 所示文档。

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： <a href="http://www.segger.com/downloads/jlink/UM08001_JLink.pdf">www.segger.com/downloads/jlink/UM08001_JLink.pdf</a>
Keil用户指南	Keil的详细操作： <a href="http://www.keil.com/support/man/docs/uv4/">www.keil.com/support/man/docs/uv4/</a>
Bluetooth Core Spec v5.1	Bluetooth官方标准核心规范5.1： <a href="https://www.bluetooth.com/specifications/bluetooth-core-specification/">https://www.bluetooth.com/specifications/bluetooth-core-specification/</a>
Bluetooth GATT Spec	Bluetooth Profile和服务的详细信息： <a href="http://www.bluetooth.com/specifications/gatt">www.bluetooth.com/specifications/gatt</a>
GProgrammer用户手册	GProgrammer软件的操作使用说明

## 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		Requirement	Description
Protocol Mode		Mandatory for Boot Protocol Mode support	Expose the current protocol mode, or set the desired protocol mode.
Report	Input Report Type	Mandatory to support at least one Report Type if the Report characteristic is supported	Exchange data between HID Device and HID Host.
	Output Report Type		
	Feature Report Type		
Report Map		Mandatory	Define formatting information for the data transferred between HID Device and HID Host.
Boot Keyboard Input Report		Mandatory for keyboards	Transfer fixed format and length Input Report data in Boot Protocol Mode
Boot Keyboard Output Report		Mandatory for keyboards	Transfer fixed format and length Output Report data in Boot Protocol Mode

Characteristic	Requirement	Description
Boot Mouse Input Report	Mandatory for mice	Transfer fixed format and length Input Report data in Boot Protocol Mode
HID Information	Mandatory	Hold a set of values known as the HID Device's HID Attributes
HID Control Point	Mandatory	A control-point for switching Suspend status

## 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[4.3.3 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
安卓手机	Android 4.4（KitKat）及以上版本且支持HOGP的手机
连接线	Micro USB 2.0串口连接线

- 软件准备

表 3-2 软件准备

名称	描述
Windows	Windows 7及以上操作系统
J-Link Driver	J-Link驱动程序，下载网址： <a href="http://www.segger.com/downloads/jlink/">www.segger.com/downloads/jlink/</a>
Keil MDK5	IDE工具，下载网址： <a href="http://www.keil.com/download/product/">www.keil.com/download/product/</a>
GProgrammer（Windows）	GR551x Programming工具，位于SDK_Folder\tools\GProgrammer

#### 3.2 硬件连接

使用Micro USB 2.0数据线连接开发板和电脑，连接示意图如图 3-1：

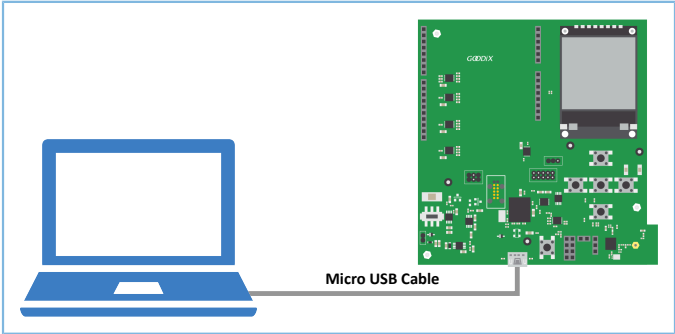


图 3-1 硬件连接示意图



### 3.3 下载固件

使用GProgrammer将`ble_app_hids_mouse_fw.bin`下载至开发板。GProgrammer的使用方法，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\`。

### 3.4 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 3-2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。

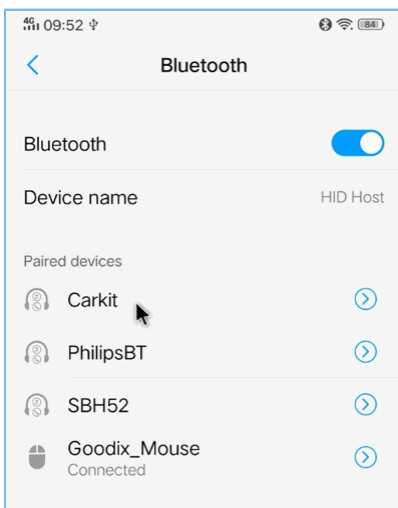


图 3-2 手机屏幕鼠标箭头示例

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

本章主要介绍鼠标示例的工程目录、运行流程以及关键代码。

### 4.1 工程目录

鼠标示例的源代码和工程文件位于SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_hids\_mouse，其中工程文件位于Keil\_5文件夹。

双击工程文件`ble_app_hids_mouse.uvprojx`，在Keil中查看鼠标示例的ble\_app\_hids\_mouse工程目录结构，相关文件说明如表 4-1 所示。

表 4-1 ble\_app\_hids\_mouse文件说明

Group	文件	描述
gr_profiles	hids.c	Human Input Device 服务实现
	bas.c	电量服务实现
	dis.c	设备信息服务实现
user_callback	user_xxx_callback.c	BLE SDK回调函数实现
user_platform	user_periph_setup.c	用户外设初始化入口和按键处理函数实现
user_app	main.c	main()入口函数
	user_app.c	用户应用层蓝牙广播、连接等控制逻辑实现
	user_mouse.c	初始化HOGP规定的GATT Service和Mouse数据发送

### 4.2 运行流程

GR551x鼠标示例的运行流程可以分为三个阶段：启动、已与HID Host连接、已与HID Host断开。

### 4.2.1 启动过程

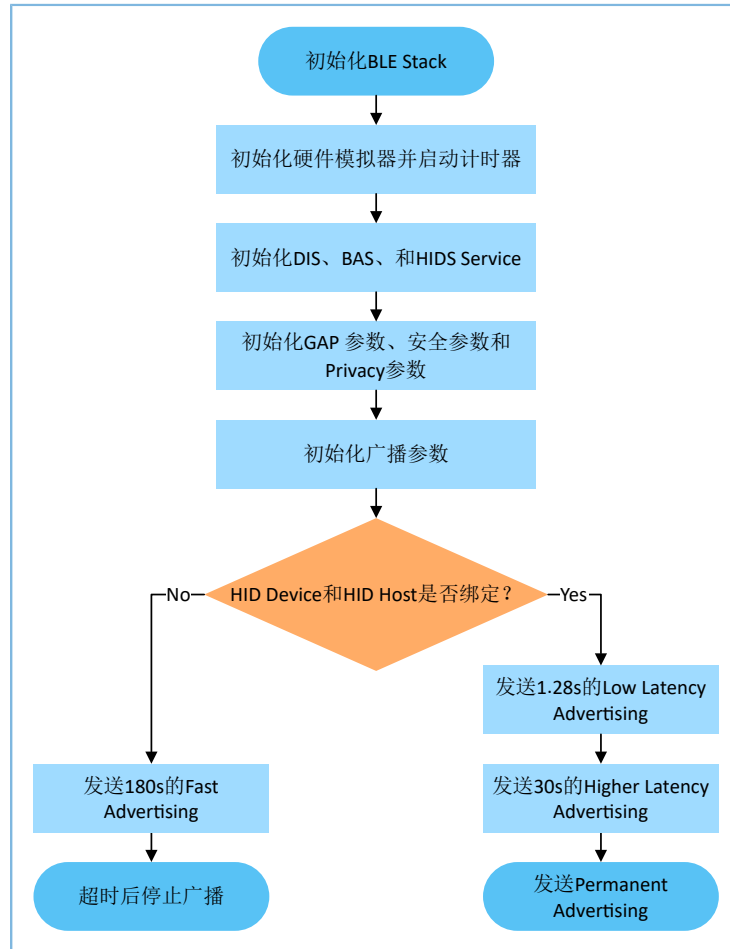


图 4-1 GR551x 鼠标示例启动流程

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_params_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。

### 4.2.2 已与HID Host连接

HID Host与HID Device完成连接、配对、绑定后，会使能HID Device的HID Report Notification。

鼠标示例的`key_continue_press_task()`会收到来自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。

### 4.2.3 已与HID Host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。`app_disconnected_handler()`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。

## 4.3 关键代码

鼠标示例的业务逻辑代码位于Keil工程目录下的以下文件中：

- user\_app\user\_app.c
- user\_app\user\_mouse.c
- user\_callback\user\_gap\_callback.c
- user\_callback\user\_sm\_callback.c

### 4.3.1 电池模拟模块的初始化

user\_app.c中的hw\_simulator\_init()初始化Battery Simulator。其状态由Timer的Timeout Handler函数hw\_simulator\_timer\_handler()更新。

```
void ble_init_cmpl_callback(void)
{
    ...
    hw_simulator_init();

    error_code = app_timer_create(&s_hw_simulator_timer_id, ATIMER_REPEAT,
                                  hw_simulator_timer_handler);
    APP_ERROR_CHECK(error_code);
    error_code = app_timer_start(s_hw_simulator_timer_id,
                                  HW_SIM_UPDATE_INTERVAL, NULL);
    APP_ERROR_CHECK(error_code);
    ...
}
```

待鼠标示例完成初始化之后，Battery Simulator将开始模拟电量的变化。

```
static void hw_simulator_timer_handler(void *p_arg)
{
    uint8_t battery_level = (uint8_t)sensorsim_measure(&s_battery_sim_state,
                                                         &s_battery_sim_cfg);
    bas_batt_lvl_update(0, 0, battery_level);
}
```

#### 说明:

鼠标示例使用Sensor Simulator库所提供的API来模拟电池电量的变化。Sensor Simulator库的源代码位于SDK\_Folder\components\libraries\sensorsim\。

### 4.3.2 HID Service的初始化

`user_mouse.c`中的`hids_init()`函数负责完成HID Service的初始化参数配置。

该文件中定义了`static const uint8_t rep_map_data[]`。按照[USB HID Specification](#)规定的格式，这个数组定义了包含鼠标Report和多媒体按键Report的Report Map。

#### 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`。

```
/**Limitation of length, as per Section 2.6.1 in HIDS Spec, V 1.0 */
#define REPORT_MAP_MAX_SIZE 512
```

如果`hids_init_t::hid_info::flags`的`NormallyConnectable` bit被置位，则“`s_normally_connectable`”需要被设置为“`true`”。

```
hid_info_flags = HIDS_INFO_FLAG_REMOTE_WAKE_MSK |
                 HIDS_INFO_FLAG_NORMALLY_CONNECTABLE_MSK;
s_normally_connectable = true;
```

### 4.3.3 安全参数配置

`user_app.c`中的`gap_params_init()`函数设置了如下安全参数，以满足[HOGP Specification](#)所规定的相关2.3 安全要求。

```
//set the default security parameters.
sec_param_t sec_param =
{
    .level = SEC_MODEL_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,
};
```

关于`sec_param_t`的设置，请参考[《GR551x BLE Stack用户指南》](#)“4 安全管理（SM）”。

`gap_params_init()`函数使用以下函数开启了隐私模式。

```
ble_gap_privacy_params_set(900, true);
```

在此模式下，鼠标示例会每隔900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。

`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3.4 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。

```
// user need to input the password
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;
```

关于处理配对加密请求，请参考[《GR551x BLE Stack用户指南》](#)“4.2.1 开启绑定功能”。