



GR5xx HRS RSCS Relay示例手册

版本： 3.1

发布日期： 2023-11-06

版权所有 © 2023 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12-13层

电话：+86-755-33338828 邮编：518000

网址：www.goodix.com

前言

编写目的

本文档介绍如何使用和验证GR5xx SDK中的HRS RSCS Relay示例，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：

- 芯片用户
- 开发人员
- 测试人员
- 开发爱好者
- 文档工程师

版本说明

本文档为第3次发布，对应的产品为低功耗蓝牙GR5xx系列。

修订记录

版本	日期	修订内容
1.0	2023-01-10	首次发布
3.0	2023-03-30	新增支持多款芯片的相关描述
3.1	2023-11-06	更新GProgrammer、GRToolbox获取方式。

目录

前言..... I

1 简介..... 1

2 Profile概述..... 2

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..... 5

4 应用详解..... 9

 4.1 运行流程..... 9

 4.2 关键代码..... 9

 4.2.1 接收来自GRToolbox的指令..... 9

 4.2.2 连接HRS传感器指令..... 10

 4.2.3 开启HRS通知指令..... 11

 4.2.4 获取HRS传感器设备位置的指令..... 11

1 简介

HRS RSCS Relay（Heart Rate Sensor & Running Speed and Cadence Sensor Relay）示例演示了如何将GR5xx芯片应用于多角色（Peripheral和Central）、多连接的场景，实现了心率传感器和跑速与步频传感器中继设备功能。HRS RSCS Relay设备可同时作为采集器和传感器。

- 采集器

作为GATT客户端，接收来自心率传感器、跑速与步频传感器的测量数据。

- 传感器

作为GATT服务端，将接收测量数据发送至其他采集器设备（例如手机APP“GRToolbox”）。

本文将介绍如何使用和验证GR5xx SDK中的HRS RSCS Relay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
GR5xx应用及自定义Sample Service	介绍实现自定义Service的相关知识
对应芯片开发者指南	介绍GR5xx SDK以及基于SDK的应用开发和调试
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
J-Link用户指南	J-Link使用说明： www.segger.com/downloads/jlink/UM08001_JLink.pdf
Keil用户指南	Keil详细操作说明： www.keil.com/support/man/docs/uv4/

2 Profile概述

HRS RSCS Relay示例实现的Profile如下：

- 标准Profile：Heart Rate Profile、Running Speed and Cadence Profile，由蓝牙技术联盟（Bluetooth SIG）定义。
- 自定义Profile：Goodix HRS RSCS Relay Control Point Profile，由Goodix自定义。

以GRToolbox作为HRS RSCS Relay的采集器为例，应用场景如图 2-1所示。

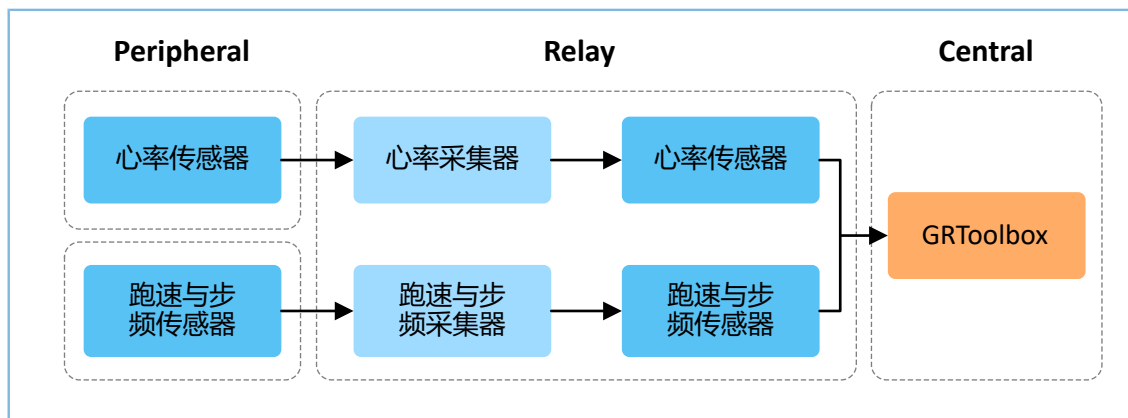


图 2-1 应用场景图

作为采集器，HRS RSCS Relay注册以下Profile：

- Heart Rate Client Profile：接收心率传感器的测量数据。
- Running Speed and Cadence Client Profile：接收跑速与步频传感器的测量数据。

作为传感器，HRS RSCS Relay注册以下Profile：

- Heart Rate Server Profile：将接收心率传感器的测量数据中继至GRToolbox。
- Running Speed and Cadence Server Profile：将接收跑速与步频传感器的测量数据中继至GRToolbox。
- Goodix HRS RSCS Relay Control Point Profile：接收来自GRToolbox的控制指令，并返回执行结果。

Goodix HRS RSCS Relay Control Point Profile 包括HRRCPs（HRS RSCS Relay Control Point Service），其专用128位UUID为A6ED0601-D344-460A-8075-B9E8EC90D71B。

HRRCPs包含两个特征：

- HRR Control Point Characteristic：接收Relay采集器的控制指令。
- HRR Control Point Response Characteristic：回应指令执行结果至Relay采集器。

Characteristic的具体描述如下表所示：

表 2-1 HRRCPs Characteristic

Characteristic	UUID	Type	Support	Security	Properties
HRR Control Point	A6ED0602-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Write

Characteristic	UUID	Type	Support	Security	Properties
HRR Control Point Response	A6ED0603-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Indicate

3 初次运行

本章介绍如何快速验证GR5xx SDK中的HRS RSCS Relay示例。

说明:

SDK_Folder为对应芯片SDK的根目录。

3.1 准备工作

运行HRS_RSCS_Relay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	对应芯片Starter Kit开发板（以下简称“开发板”）3块
连接线	USB Type-C（GR551x系列使用Micro USB 2.0连接线）

- 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/
GRToolbox（Android）	Bluetooth LE调试工具，下载网址： www.goodix.com/zh/software_tool/grtoolbox
GProgrammer（Windows）	Programming工具，下载网址： www.goodix.com/zh/software_tool/gprogrammer_ble

3.2 固件烧录

HRS RSCS Relay示例工程的源码位于SDK_Folder\projects\ble\ble_multi_role\ble_app_hrs_rscs_relay。

用户可通过GProgrammer依次将`ble_app_hrs_rscs_relay.bin`、`ble_app_hrs.bin`和`ble_app_rscs.bin`烧录至三块开发板，这三块开发板分别作为Relay设备、HRS传感器设备和RSCS传感器设备。GProgrammer烧录固件的具体操作方法，请参考《GProgrammer用户手册》。

说明:

- `ble_app_hrs_rscs_relay.bin`位于SDK_Folder\projects\ble\ble_multi_role\ble_app_hrs_rscs_relay\build。
- `ble_app_hrs.bin`位于SDK_Folder\projects\ble\ble_peripheral\ble_app_hrs\build。
- `ble_app_rscs.bin`位于SDK_Folder\projects\ble\ble_peripheral\ble_app_rscs\build。

3.3 测试验证

HRS RSCS Relay设备、HRS传感器设备、RSCS传感器设备和GRToolbox准备完毕，即可进行HRS RSCS Relay示例的测试验证，具体步骤如下：

1. 扫描HRS RSCS Relay设备

打开GRToolbox，选择“应用 > RELAY”。

手机扫描发现广播名为“Goodix_HRS_RSCS_RELAY”的设备（广播名可在`user_app.c`文件中修改），如图 3-1所示。

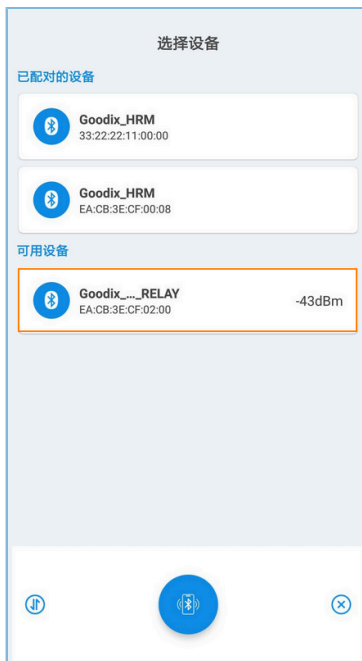


图 3-1 在手机端发现Goodix_HRS_RSCS_RELAY

说明:

本文中GRToolbox的截图仅供用户了解操作步骤，实际界面请参考最新版本GRToolbox。

设备全名若超出14个字符，名称中间将以省略号显示。

2. 连接HRS RSCS Relay设备

选中并连接“Goodix_HRS_RSCS_RELAY”，进入HRS RSCS RELAY界面。



图 3-2 HRS RSCS RELAY界面

3. 连接传感器设备



点击连接  按钮，控制HRS RSCS Relay设备扫描、连接HRS和RSC传感器设备。Relay设备连接成功后如下图所示。



图 3-3 连接传感器设备

4. 开启传感器通知

使用通知  按钮，控制HRS RSCS Relay设备通知HRS和RSC传感器上报采集的数据。

完成后即可接收HRS RSCS Relay设备中继出的心率、跑速与步频相关信息。



图 3-4 开启HRS传感器通知

5. 读取HRS传感器位置

点击“获取”按钮，控制HRS RSCS Relay设备读取HRS传感器位置信息。



图 3-5 读取HRS传感器位置

若实际情况符合上述说明，则HRS RSCS Relay应用示例运行成功。

4 应用详解

本章将介绍HRS RSCS Relay示例的运行流程和关键代码。

4.1 运行流程

HRS RSCS Relay示例开始运行后，将依次执行初始化外设和Bluetooth LE协议栈、添加Profiles以及开启广播等操作。

当GRToolbox扫描到该示例的广播并建立连接后，运行流程如图 4-1所示：

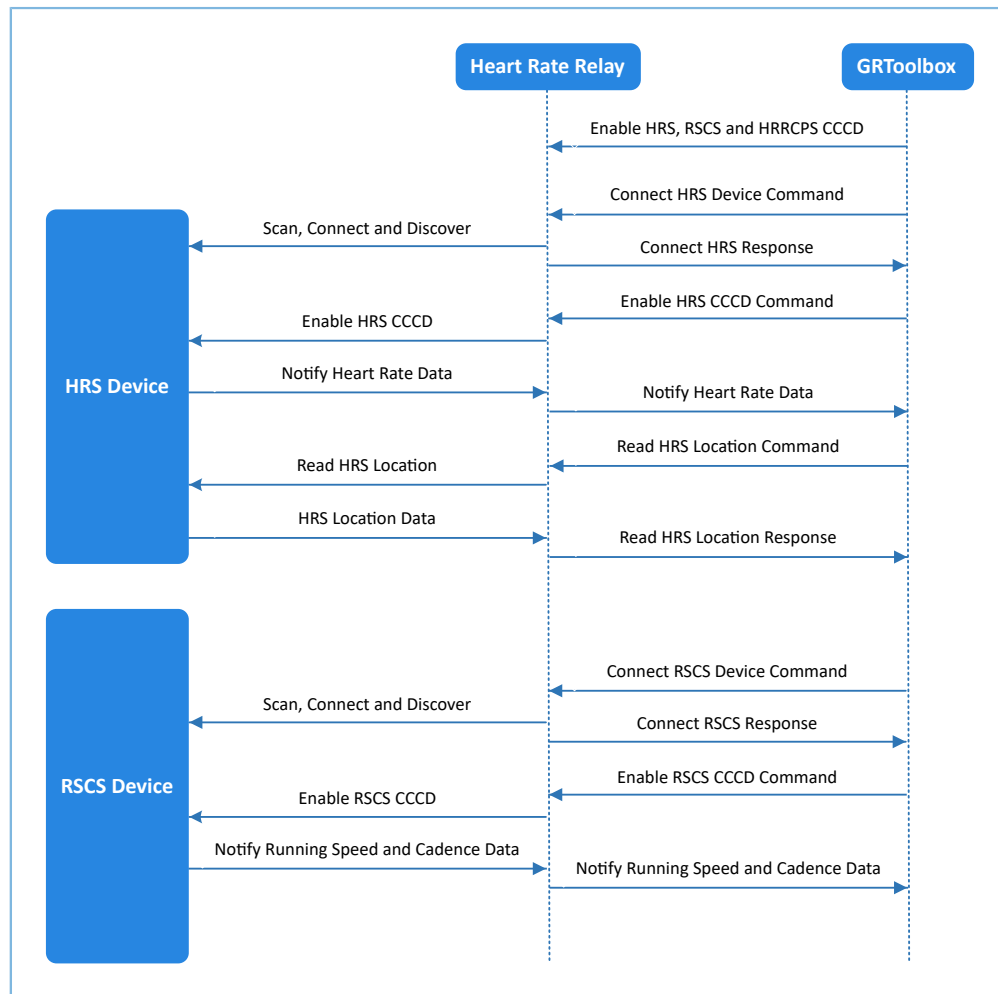


图 4-1 运行流程图

4.2 关键代码

下文以HRS传感器设备为例，详细介绍GRToolbox、Relay设备以及HRS传感器设备交互过程中的关键代码。

4.2.1 接收来自GRToolbox的指令

当HRR Control Point Characteristic Value接收到GRToolbox的控制指令数据时，会解析出相应的事件上报至应用层，并执行相应的指令。

路径：工程目录下user_app\user_app.c

名称: hrrcps_evt_process()

```
static void hrrcps_evt_process(hrrcps_evt_t *p_evt)
{
    ...
    switch (p_evt->evt_type)
    {
        case HRRCPES_EVT_CTRL_PT_IND_ENABLE:
            APP_LOG_DEBUG("HRR Control Point Indication is enabled.");
            break;

        case HRRCPES_EVT_CTRL_PT_IND_DISABLE:
            APP_LOG_DEBUG("HRR Control Point Indication is disabled.");
            break;

        .....
        default:
            break;
    }
    ...
}
```

4.2.2 连接HRS传感器指令

HRRCPES解析该指令并以“HRRCPES_EVT_SCAN_HRS”事件上报至应用层，然后开启扫描、过滤HRS传感器设备。当Relay设备扫描到广播后，通过app_adv_report_handler()判断发出该广播的设备是否为目标设备，判断方法为检查广播数据中是否含有HRS UUID。当确定其为目标设备后停止扫描，通过app_scan_stop_handler()与其建立连接，再通过app_connected_handler()发现Heart Rate Service（上述函数均位于user_app.c中）。

路径: 工程目录下user_app\user_app.c

名称: hrrcps_evt_process()

```
static void hrrcps_evt_process(hrrcps_evt_t *p_evt)
{
    .....
    case HRRCPES_EVT_SCAN_HRS:
        .....
        error_code = ble_gap_scan_start();
        if (error_code != SDK_SUCCESS)
        {
            hrrcps_op_error_handler(HRRCPES_CTRL_PT_SCAN_HRS);
        }

        g_hrs_active_state = SCAN_DEV_STATE;
        break;
    .....
}
```

4.2.3 开启HRS通知指令

HRRCPs解析该指令并以“HRRCPs_EVT_ENABLE_HRS_NTF”事件上报至应用层，开启HRS传感器通知，然后将接收到心率数据中继至GRTtoolbox。

路径：工程目录下user_app\user_app.c

名称：hrrcps_evt_process()

```
static void hrrcps_evt_process(hrrcps_evt_t *p_evt)
{
    ...
    case HRRCPs_EVT_ENABLE_HRS_NTF:
        error_code = hrs_c_heart_rate_meas_notify_set(s_conn_idx_hrs_c, true);
        if (error_code != SDK_SUCCESS)
        {
            hrrcps_op_error_handler(HRRCPs_CTRL_PT_HRS_NTF_ENABLE);
        }
        s_user_write_id = USER_WR_HRS_NTF_EN;
        APP_LOG_DEBUG("Enable HRS notification.");
        break;
    ...
}
```

路径：工程目录下user_app\user_app.c

名称：hrs_c_evt_process()

```
static void hrs_c_evt_process(hrs_c_evt_t *p_evt)
{
    .....
    case HRS_C_EVT_HR_MEAS_VAL_RECEIVE:
        for (rr_intervals_idx = 0; rr_intervals_idx < p_evt->value.hr_meas_buff.rr_intervals_num; rr_intervals_idx++)
        {
            hrs_rr_interval_add(p_evt->value.hr_meas_buff.rr_intervals[rr_intervals_idx]);
        }
        hrs_sensor_contact_detected_update(p_evt->value.hr_meas_buff.is_sensor_contact_detected);

        hrs_heart_rate_measurement_send(s_conn_idx_collector, p_evt->value.hr_meas_buff.hr_value, p_evt->value.hr_meas_buff.energy_expended);

        break;
    .....
}
```

4.2.4 获取HRS传感器设备位置的指令

HRRCPs解析该指令并以“HRRCPs_EVT_HRS_SENSOR_LOC_READ”事件上报至应用层，然后读取HRS传感器位置，并将读取数据中继至GRTtoolbox。

路径：工程目录下user_app\user_app.c

名称：hrrcps_evt_process()

```
static void hrrcps_evt_process(hrrcps_evt_t *p_evt)
{
    .....
    case HRRCPES_EVT_HRS_SENSOR_LOC_READ:
        error_code = hrs_c_sensor_loc_read(s_conn_idx_hrs_c);
        if (error_code != SDK_SUCCESS)
        {
            hrrcps_op_error_handler(HRRCPES_CTRL_PT_HRS_SEN_LOC_READ);
        }
        APP_LOG_DEBUG("Read HRS sensor location.");
        break;
    .....
}
```

路径：工程目录下user_app\user_app.c

名称：hrs_c_evt_process()

```
static void hrs_c_evt_process(hrs_c_evt_t *p_evt)
{
    ...
    case HRS_C_EVT_SENSOR_LOC_READ_RSP:
        APP_LOG_DEBUG("HRS sensor location is got.");
        hrs_sensor_location_set((hrs_sensor_loc_t)p_evt->value.sensor_loc);
        rsp_val.cmd_id      = HRRCPES_CTRL_PT_HRS_SEN_LOC_READ;
        rsp_val.rsp_id      = HRRCPES_RSP_ID_OK;
        rsp_val.is_inc_prama = true;
        rsp_val.rsp_param    = p_evt->value.sensor_loc;
        error_code = hrrcps_ctrl_pt_rsp_send(s_conn_idx_collector, &rsp_val);
        APP_ERROR_CHECK(error_code);
        break;
    ...
}
```

说明:

GRTtoolbox控制HRS RSCS Relay设备与RSCS传感器设备交互流程，与上述流程类似，此处不再赘述。