GR551x ANCS Profile示例手册

版本： **1.6**

发布日期： **2020-06-30**

深圳市汇顶科技股份有限公司

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，不得暗中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：深圳市福田保税区腾飞工业大厦B座2层、13层

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

网址：www.goodix.com

前言

编写目的

本文档介绍如何使用和验证GR551x SDK中的ANCS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：

- GR551x用户
- GR551x开发人员
- GR551x测试人员
- iOS开发工程师
- 开发爱好者
- 文档工程师

版本说明

本文档为第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
1.3	2020-03-16	更新了文档页脚版本时间
1.5	2020-05-30	更新了“4.2交互流程及代码介绍”章节中的代码
1.6	2020-06-30	修改了示例中使用的配对码 更新了“4.2.4 控制命令”章节中的代码

版权所有 © 2020 深圳市汇顶科技股份有限公司

目录

前言 I

1 简介 1

2 Profile概述 2

3 初次运行 3

3.1 准备工作 3
3.2 硬件连接 3
3.3 下载固件 4
3.4 蓝牙连接 4
3.5 串口设置 5
3.6 测试验证 6

4 应用详解 8

4.1 工程目录 8
4.2 交互流程及代码介绍 8
4.2.1 连接、配对及绑定 10
4.2.2 发现服务 11
4.2.3 通知事件及解析 12
4.2.4 控制命令 15
4.2.4.1 获取通知属性 15
4.2.4.2 执行操作 16
4.2.4.3 交互 17

5 常见问题 19

5.1 串口终端没有输出信息 19
5.2 手机搜索不到蓝牙广播 19
5.3 连接成功后无法获取到通知 19

版权所有 © 2020 深圳市汇顶科技股份有限公司

1 简介

苹果通知中心服务（Apple Notification Center Service，ANCS）应用于BLE设备，如手环、手表等智能设备，可通过BLE获取iOS设备的各类通知；同时BLE设备也可以使用ANCS向iOS设备发送ANCS相关控制命令。

本文档主要介绍如何在GR551x平台实现ANCS Client。

在进行操作前，建议参考以下文档。

表 1-1 文档参考

名称	描述
ANCS Specification	ANCS协议：Apple Notification Center Service (ANCS) Specification
GR551x开发者指南	GR551x软硬件介绍、快速使用及资源总览
Bluetooth Core Spec v5.1	Bluetooth官方标准核心规范5.1： https://www.bluetooth.com/specifications/bluetooth-core-specification/
Bluetooth GATT Spec	Bluetooth Profile和Service的详细信息：https://www.bluetooth.com/specifications/gatt
J-Link用户指南	J-Link使用说明： https://www.segger.com/downloads/jlink/UM08001_JLink.pdf
Keil用户指南	Keil详细操作说明：htpps://www.keil.com/support/man/docs/uv4/

版权所有 © 2020 深圳市汇顶科技股份有限公司

2 Profile概述

ANCS Profile定义了两种设备角色：

1. Server端：iOS设备为中央设备（Central），提供Service以及数据源。
2. Client端：BLE设备为外围设备（Peripheral），连接iOS设备后发现其服务以及读写数据。

Server端与Client端的交互流程如图 2-1所示：

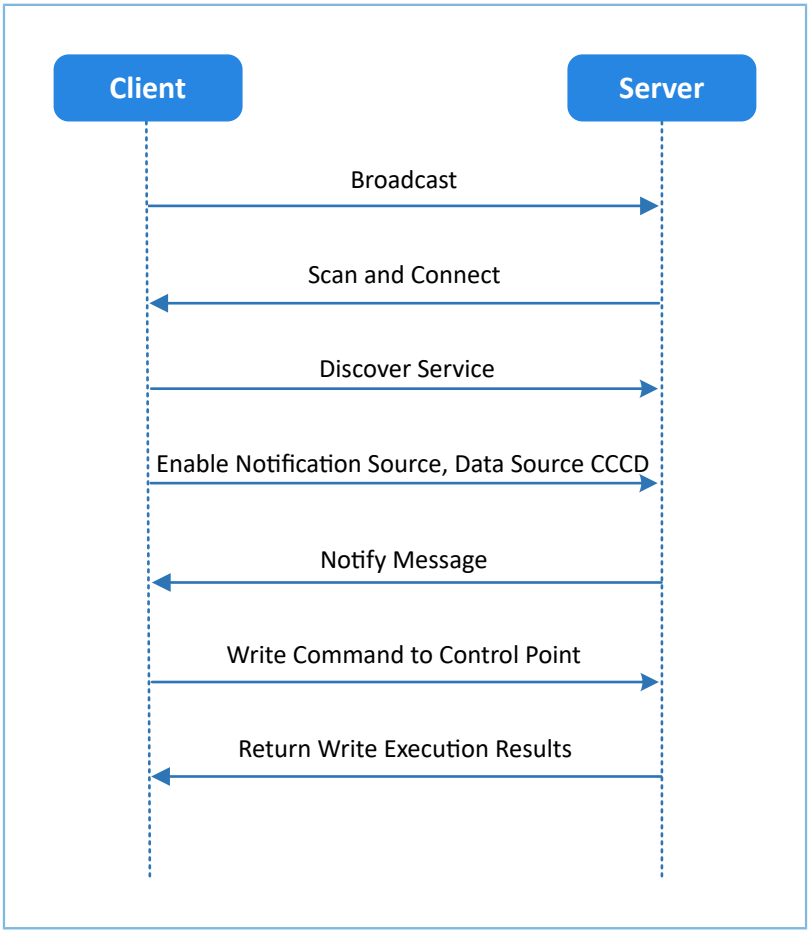

图 2-1 Client & Server交互流程图

ANCS主要包括以下3种特征值（Characteristic）。

表 2-1 ANCS Characteristic

Characteristic	UUID	Type	Support	Security	Properties
Notification Source	9FBF120D-6301-42D9-8C58-25E699A21DBD	128 bits	Mandatory	None	Notify
Control Point	69D1D8F3-45E1-49A8-9821-9BBDFDAAD9D9	128 bits	Mandatory	None	Write
Data Source	22EAC6E9-24D6-4BB5-BE44-B36ACE7C7BFB	128 bits	Mandatory	None	Notify

版权所有 © 2020 深圳市汇顶科技股份有限公司

3 初次运行

本章介绍如何快速验证GR551x SDK中的ANCS Client示例。

说明:

SDK_Folder为GR551x SDK的根目录。

3.1 准备工作

运行ANCS Client示例之前，需要完成以下准备工作。

- 硬件准备

表 3-1 硬件准备

名称	描述
J-Link工具	SEGGER公司推出的JTAG仿真器，如需更多了解，请访问 https://www.segger.com/products/debug-probes/j-link/
开发板	GR5515 Starter Kit开发板
数据线	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：https://www.segger.com/downloads/jlink/
Keil MDK5	IDE工具，下载网址：https://www.keil.com/download/product/
iOS设备	支持BLE4.0及以上的iOS设备，如iPhone 4s/iPad 3及其以上版本
GProgrammer（Windows）	GR551x Programming工具，位于`SDK_Folder\tools\GProgrammer`
GRUart（Windows）	GR551x串口调试工具，位于`SDK_Folder\tools\GRUart`

3.2 硬件连接

使用Micro USB 2.0数据线连接GR5515 Starter Kit开发板与计算机。

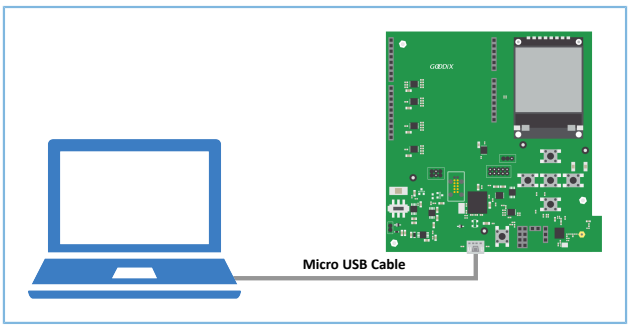

图 3-1 硬件连接示意图

版权所有 © 2020 深圳市汇顶科技股份有限公司

3.3 下载固件

下载ANCS示例的*ble_app_ancs_c_fw.bin*固件至开发板。具体操作方法，请参考《GProgrammer用户手册》。

说明:

*ble_app_ancs_c_fw.bin*位于`SDK_Folder\projects\ble\ble_peripheral\ble_app_ancs_c\build\`，其中SDK_Folder为GR551x SDK的根目录。

3.4 蓝牙连接

开发板上电后，打开iOS设备蓝牙功能，搜索周边蓝牙设备，发现广播设备名为Goodix_ANCS_C的蓝牙设备，如图 3-2所示。

说明:

本文档测试用机为iPhone 7，系统版本为11.03。若页面有差异，请以实际界面情况为准。

图 3-2 发现广播名Goodix_ANCS_C

点击“Goodix_ANCS_C”，连接开发板，弹出下图所示的配对请求信息，输入代码“123456”（参考4.2.1 连接、配对及绑定中app_sec_rcv_enc_req_cb()函数说明）后点击“配对”。

版权所有 © 2020 深圳市汇顶科技股份有限公司

图 3-3 输入配对信息

配对成功后，提示：Goodix_ANCS_C已连接。

图 3-4 设备配对成功

3.5 串口设置

启动GRUart，按照表 3-3 中的参数配置串口。

表 3-3 GRUart串口配置参数

PortName	BaudRate	DataBits	Parity	StopBits	Flow Control
需根据实际选择	115200	8	None	1	不勾选

版权所有 © 2020 深圳市汇顶科技股份有限公司

配置完成后，点击“Open Port”打开，如图 3-5所示。

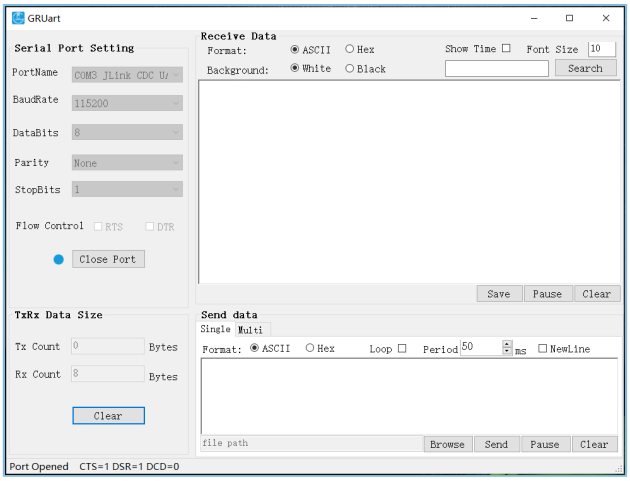

图 3-5 串口设置

此时还未有ANCS服务通知产生，因此界面上无任何显示。

3.6 测试验证

用户可根据串口终端的打印信息，验证ANCS服务是否运行正常。

下图以一条通知信息进行举例说明（如需了解详情，请参考Apple Notification Center Service (ANCS) Specification）。

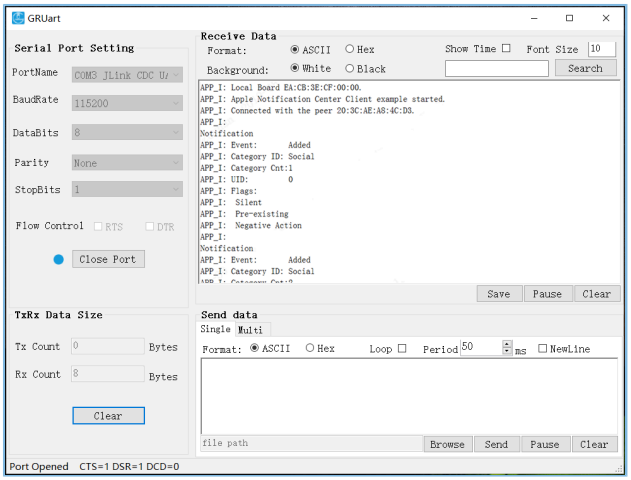

图 3-6 串口打印信息

串口输出信息解释如下：

表 3-4 通知信息说明

名称	描述
Notification	表示这是一条通知消息
Event: Added	事件类型为：添加
Category ID: Social	信息类型为：社交
Category Cnt: 1	类型数量为：1

版权所有 © 2020 深圳市汇顶科技股份有限公司

名称	描述
UID: 0	唯一标识符（UID）为0
Flags: Silent	信息类型：Silent
Pre-existing	存在于缓存区
Negative Action	表示可执行删除操作

以用户接打电话为例。使用其他手机给本iOS测试机拨打电话，拨通后本测试机将立刻推送通知到GR5515开发板，开发板将处理接收到的通知消息，并将信息打印到GRUart。

此时用户如需知道具体拨入的电话号码，可按下开发板的“OK”按键，GRUart将显示以下信息（包括当前拨入的电话号码）。

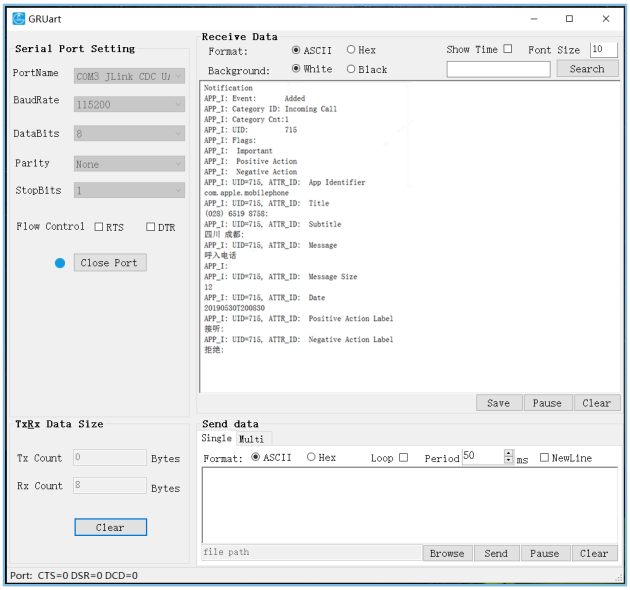

图 3-7 打印通知信息

如用户选择接听，可以按下开发板“RIGHT”键，查看到电话已接通；如用户拒绝接听，可以按下“LEFT”键，查看到电话已挂断。

说明:

GR5515 Starter Kit开发板按键的详细信息，请参考《GR5515 Starter Kit用户指南》“7 按键和指示灯”。

版权所有 © 2020 深圳市汇顶科技股份有限公司

4 应用详解

本章主要将介绍ANCS Client示例的交互流程及相关代码。

4.1 工程目录

ANCS Client示例的源代码和工程文件位于`SDK_Folder\projects\ble\ble_peripheral\ble_app_ancs_c`，其工程文件在Keil_5文件夹下。

双击打开工程文件*ble_app_ancs_c.uvprojx*，在Keil中查看ANCS Client示例ble_app_ancs_c工程目录结构，相关文件介绍如表 4-1 所示。

表 4-1 ble_app_ancs_c文件说明

Group	文件	描述
gr_profiles	ancs_c.c	ANCS中相关GATT操作
	ancs_protocol.c	ANCS中相关协议内容
user_callback	user_gap_callback.c	GAP Callback实现，如连接、断连、GAP参数更新等
	user_sm_callback.c	SM Callback实现，如配对绑定
user_platform	user_periph_setup.c	App log、设备地址和电源管理模式的配置
user_app	main.c	main()入口函数
	user_app.c	实现ANCS广播参数设置
	user_gui.c	按键处理及GUI显示

4.2 交互流程及代码介绍

本例程以ANCS Client实际交互为例，分步讲解了ANCS Client配对与绑定、发现服务、CCCD（Client Characteristic Configuration Descriptor）使能、通知消息处理和命令交互方式，便于用户深入了解各模块的主要代码。

交互流程如图 4-1所示。

版权所有 © 2020 深圳市汇顶科技股份有限公司

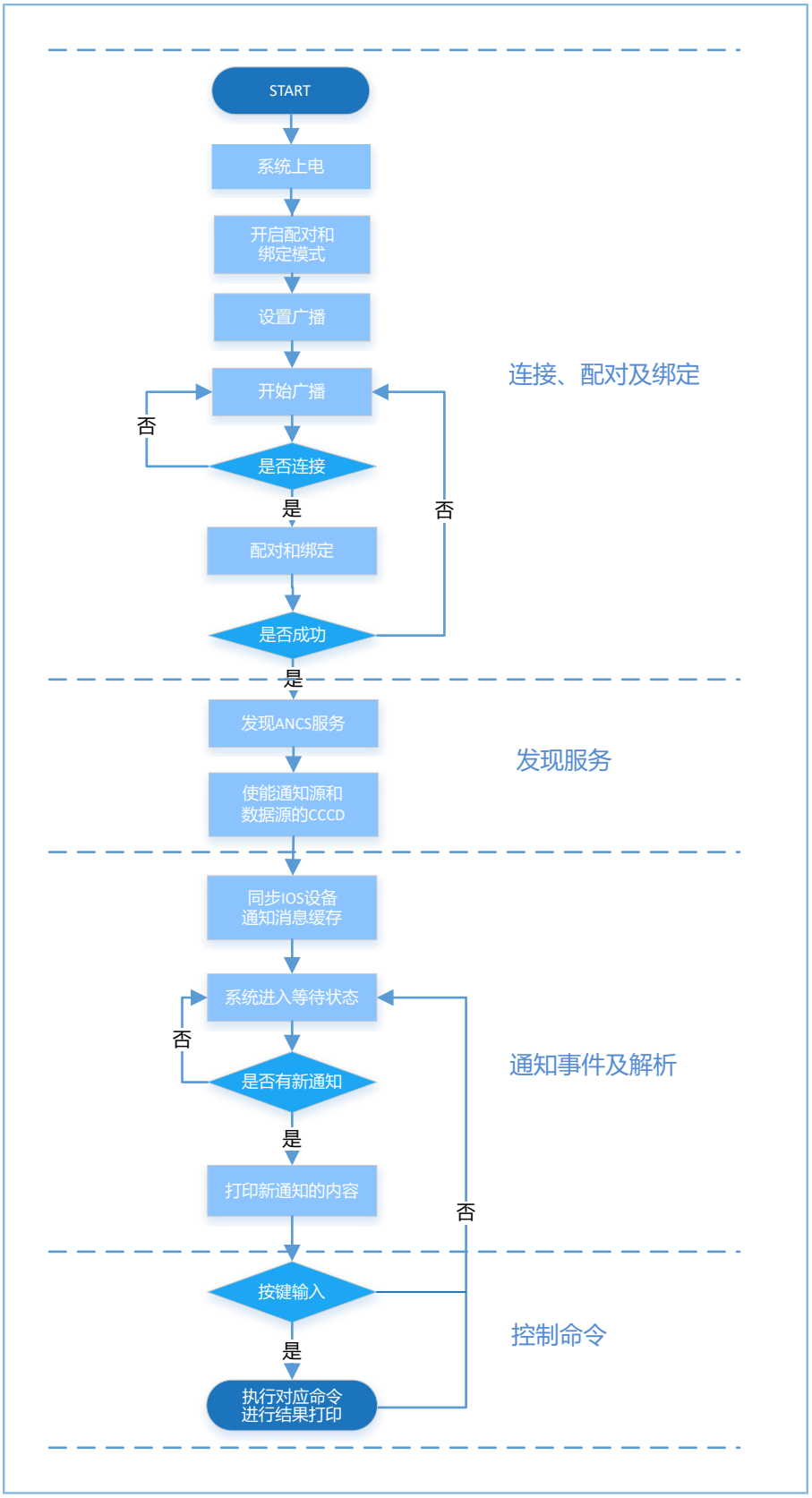

图 4-1 ANCS交互流程图

版权所有 © 2020 深圳市汇顶科技股份有限公司

4.2.1 连接、配对及绑定

GR5515开发板上电后，作为Peripheral角色进行广播；iOS设备作为Central角色，可搜索周边蓝牙设备。

iOS设备打开蓝牙功能后，扫描到GR5515开发板并发起连接，然后进行配对绑定。主要代码介绍如下。

路径：工程目录下的`user_app/user_app.c`

名称：gap_params_init();

使能配对和绑定模式、设置当前安全级别（默认为Security Mode Level 3，用户可自定义）和广播参数，下面列出关键代码。

说明:

关于LE Security Modes Level设置的详细信息，请参考Bluetooth Core Spec v5.1中的“10.2 LE security modes（Vol 3，Part C）”。

```
static void gap_params_init(void)
{
    sdk_err_t error_code;
    ble_gap_pair_enable(true);
    error_code = ble_gap_privacy_params_set(150, true);
    APP_ERROR_CHECK(error_code);

    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_ENCKEY | KDIST_IDKEY,
        .rkey_dist = KDIST_ENCKEY | KDIST_IDKEY,
    };
    ...
}
```

路径：工程目录下的`user_callback/user_sm_callback.c`

名称：app_sec_rcv_enc_req_cb();

设置自定义配对密码“123456”并赋值到cfm_enc.data.tk.key[4]数组，代码如下示。

```
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)
    {
        ...
        // user need to input the password
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
        case TK_REQ:
            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;

        default:
            break;
    }

    ble_sec_enc_cfm(conn_idx, &cfm_enc);
}
```

路径：工程目录下的`user_callback/user_sm_callback.c`

名称：app_sec_rcv_enc_ind_cb();

GR551x SDK通知ANCS Client示例配对完毕。如果配对成功，则ANCS Client示例将调用timeout为1秒的计时器函数，延时1秒之后在超时响应函数中调用ancs_c_discovery_service()发起服务发现。

说明:

当iOS设备重新启动时，可能会在ANCS服务启动之前，自动连接已绑定的GR5515开发板，这将导致GR5515开发板发现ANCS服务失败。配对后延时1秒才发起服务发现就是为了避免这种情况。

```
static void app_sec_rcv_enc_ind_cb(uint8_t conn_idx, sec_enc_ind_t enc_ind, uint8_t auth)
{
    if (ENC_SUCCESS == enc_ind)
    {
        ancs_c_disc_delay_timer_start(conn_idx);
    }
}
```

4.2.2 发现服务

ANCS Client示例在建立连接并配对成功后，发起搜索iOS设备ANCS服务请求，然后根据Service Browse回调函数的结果，打开Notification Source以及Data Source的Notify。

路径：工程目录下的`gr_profiles/ancs_c.c`

名称：ancs_c_discovery_service();

通过指定的UUID查找/发现目标设备上对应的Service。查询结果将通过回调的方式返回（回调函数*user_gattc_callback.c*中的ancs_c_on_browse_svc_evt()）。

```
sdk_err_t ancs_c_discovery_service(uint8_t conn_idx)
{
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
    ble_uuid_t ble_ancs_uuid =
    {
        .uuid_len = BLE_ATT_UUID_128_LEN,
        .uuid = (uint8_t*)ancs_service_uuid,
    };
    return ble_gattc_prf_services_browse(s_ancs_c_env.prf_id, conn_idx, &ble_ancs_uuid);
}
```

路径：工程目录下的gr_profiles/ancs_c

名称：ancs_c_srvc_browse_cb ();

获取手机端ANCS的服务，通过UUID枚举每一个服务对应的Handle。

```
void ancs_c_srvc_browse_cb (uint8_t conn_idx, const ble_gattc_browse_srvc_t *p_browse_srvc)
{
    ...
    for(i=0;i<(p_browse_srvc->end_hdl- p_browse_srvc->start_hdl); i++)
    {
        if(p_browse_srvc->info[i].attr_type == BLE_GATTC_BROWSE_ATTR_VAL)
        {
        ...
        //find ancs notification source characteristic
        //find ancs control point characteristic
        //find ancs data source characteristic
        ...
        }
        else if((p_browse_srvc->info[i].attr_type ==
                 BLE_GATTC_BROWSE_ATTR_DESC) &&   //find cccd
                 ((*(uint16_t *)(p_browse_srvc->info[i].attr.uuid)) ==
                  BLE_ATT_DESC_CLIENT_CHAR_CFG))
        {
        ...
        // find notification source cccd
        // find data source cccd
        ...
        }
    }
    ...
}
```

4.2.3 通知事件及解析

当iOS设备产生通知时，iOS设备会发送信息给ANCS Client示例，并绑定相关应用信息。此时ANCS Client示例处于接收最新通知就绪状态。主要代码介绍如下。

路径：工程目录下的gr_profiles/ancs_c.c

名称：ancs_c_ntf_source_notify_set ();和ancs_c_data_source_notify_set ();

当应用状态为ANCS_CCCD_ENABLE时，通过发现服务中所获取到的Handle使能Notification source和Data source通知属性。

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
sdk_err_t ancs_c_ntf_source_notify_set(uint8_t conn_idx, bool is_enable)
{
    gattc_write_attr_value_t write_attr_value;
    uint16_t ntf_value = is_enable ? PRF_CLI_START_NTF : PRF_CLI_STOP_NTFIND;

    if (BLE_ATT_INVALID_HDL == s_ancs_c_env.handles.ancs_ntf_source_cccd_handle)
    {
        return BLE_ATT_ERR_INVALID_HANDLE;
    }
    write_attr_value.handle = s_ancs_c_env.handles.ancs_ntf_source_cccd_handle;
    write_attr_value.offset  = 0;
    write_attr_value.length  = 2;
    write_attr_value.p_value = (uint8_t *)&ntf_value;

    return ble_gattc_prf_write(s_ancs_c_env.prf_id, conn_idx, &write_attr_value);
}
sdk_err_t ancs_c_data_source_notify_set(uint8_t conn_idx, bool is_enable)
{
    gattc_write_attr_value_t write_attr_value;
    uint16_t ntf_value = is_enable ? PRF_CLI_START_NTF : PRF_CLI_STOP_NTFIND;

    if (BLE_ATT_INVALID_HDL == s_ancs_c_env.handles.ancs_data_source_cccd_handle)
    {
        return BLE_ATT_ERR_INVALID_HANDLE;
    }
    write_attr_value.handle  = s_ancs_c_env.handles.ancs_data_source_cccd_handle;
    write_attr_value.offset  = 0;
    write_attr_value.length  = 2;
    write_attr_value.p_value = (uint8_t *)&ntf_value;

    return ble_gattc_prf_write(s_ancs_c_env.prf_id, conn_idx, &write_attr_value);
}
```

路径：工程目录下的gr_profiles/ancs_c.c

名称：ancs_c_att_ntf_ind_cb();

派发Notification source和Data source两类信息，同时将对应信息传给对应的解析函数。

```
static void ancs_c_att_ntf_ind_cb(uint8_t conn_idx, const ble_gattc_ntf_ind_t *p_ntf_ind)
{
    ancs_c_evt_t ancs_c_evt;

    ancs_c_evt.conn_idx = conn_idx;
    ancs_c_evt.evt_type = BLE_ANCS_C_EVT_INVALID;

    if (p_ntf_ind->handle == s_ancs_c_env.handles.ancs_ntf_source_handle)
    {
        ancs_c_evt.evt_type = BLE_ANCS_C_EVT_NTF_SOURCE_RECEIVE;
        ancs_decode_notification_source(p_ntf_ind->p_value, p_ntf_ind->length);
    }
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
    else if (p_ntf_ind->handle == s_ancs_c_env.handles.ancs_data_source_handle)
    {
        ancs_c_evt.evt_type = BLE_ANCS_C_EVT_DATA_SOURCE_RECEIVE;
        ancs_decode_data_source(p_ntf_ind->p_value, p_ntf_ind->length);
    }

    ancs_c_evt_handler_excute(&ancs_c_evt);
}
```

路径：工程目录下的gr_profiles/ancs_protocol.c

名称：ancs_decode_notification_source();

Notification source数据解析函数。在本测试例程中，系统只是根据ANCS协议要求打印通知信息，具体协议细节请参考Apple Notification Center Service (ANCS) Specification。

```
void ancs_decode_notification_source(uint8_t *p_data, uint16_t length)
{
    ntf_source_pdu_t *pdu = (ntf_source_pdu_t*)p_data;
    notification_content_print(pdu);

    ancs_set_uid((unsigned int) pdu->notification_uid);}
```

路径：工程目录下的gr_profiles/ancs_protocol.c

名称：ancs_decode_data_source ();

Data source数据解析函数。在本测试例程中，系统按照ANCS协议要求打印数据内容。具体协议细节请参考Apple Notification Center Service (ANCS) Specification。

```
void ancs_decode_data_source(uint8_t *p_data, uint16_t length)
{
    //It's the begginning of a attr info.
    if (0 == s_buf_index)
    {
        memcpy(&s_attr_size, &p_data[6], 2);
        memcpy(s_attr_buf, p_data, length);
        //It's the end of the attr info, a complete attr info is already
          stored in the buffer.
        if (s_attr_size == length - 8)
        {
            ancs_notify_attr_print();
            s_buf_index = 0;
        }
        //It's not the end of the attr info.
        else
        {
            s_buf_index = length;
        }
    }

    //It isn't the begginning of a attr info.
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
    else
    {
        memcpy(&s_attr_buf[s_buf_index], p_data, length);
        //It's the end of the attr info, print the buffer.
        if (s_attr_size == (s_buf_index - 8) + length)
        {
            ancs_notify_attr_print();
            s_buf_index = 0;
        }
        //It's not the end of the attr info.
        else
        {
            s_buf_index = s_buf_index + length;
        }
    }
}
```

4.2.4 控制命令

通过ANCS Client向ANCS Server的Control Point写入控制信息，在iOS设备返回的Data Source中可了解某条通知中的详细信息。以下章节将详细介绍相关的控制命令。

4.2.4.1 获取通知属性

路径：工程目录下的gr_profiles/ancs_protocol.h

名称：ancs_protocol.h

宏定义用于获取通知属性可选值：

```
typedef enum
{
    ANCS_NOTIF_ATTR_ID_APP_IDENTIFIER = 0,     /**< Identifies that the
    attribute data is of an "App Identifier" type. */
    ANCS_NOTIF_ATTR_ID_TITLE,                  /**< Identifies that the
    attribute data is a "Title". */
    ANCS_NOTIF_ATTR_ID_SUBTITLE,               /**< Identifies that the
    attribute data is a "Subtitle". */
    ANCS_NOTIF_ATTR_ID_MESSAGE,                /**< Identifies that the
    attribute data is a "Message". */
    ANCS_NOTIF_ATTR_ID_MESSAGE_SIZE,           /**< Identifies that the
    attribute data is a "Message Size". */
    ANCS_NOTIF_ATTR_ID_DATE,                   /**< Identifies that the
    attribute data is a "Date". */
    ANCS_NOTIF_ATTR_ID_POSITIVE_ACTION_LABEL,  /**< The notification has a
    "Positive action" that can be executed associated with it. */
    ANCS_NOTIF_ATTR_ID_NEGATIVE_ACTION_LABEL,  /**< The notification has a
    "Negative action" that can be executed associated with it. */
} ancs_notification_attr_t;
```

路径：工程目录下的gr_profiles/ancs_protocol.c

版权所有 © 2020 深圳市汇顶科技股份有限公司

名称：ancs_notify_attr_get();

获取对应UID通知的属性，例如：iOS设备发送E-Mail邮件时，系统可通过该函数去查询邮件的详细内容、接收时间、发送人等信息。

```
void ancs_notify_attr_get(int uid, char noti_attr)
{
    int len = 0;
    uint8_t buf[8];
    buf[0] = CTRL_POINT_GET_NTF_ATTRIBUTE;
    memcpy(&buf[1], &uid, 4);
    buf[5] = noti_attr;
    if (ANCS_NOTIF_ATTR_ID_TITLE == noti_attr || ANCS_NOTIF_ATTR_ID_SUBTITLE== noti_attr
                                              || ANCS_NOTIF_ATTR_ID_MESSAGE== noti_attr)
    {
        len = CFG_ANCS_ATTRIBUTE_MAXLEN;
        buf[6] = (len & 0xff);
        buf[7] = (len>>8) & 0xff;
        ancs_c_write_control_point(0, buf, 8);
    }
    else
    {
        ancs_c_write_control_point(0, buf, 6);
    }
}
```

4.2.4.2 执行操作

路径：工程目录下的gr_profiles/ancs_protocol.h

名称：ancs_protocol.h

针对每一条通知消息，用户可做两种选择：0表示同意，1表示拒绝。

```
typedef enum
{
    ACTION_ID_POSITIVE = 0,                  /**< Positive action. */
    ACTION_ID_NEGATIVE                       /**< Negative action. */
} ancs_c_action_id_t;
```

路径：工程目录下的gr_profiles/ancs_protocol.c

名称：ancs_action_perform();

通知消息处理函数。

```
void ancs_action_perform(int uid, int action)
{
    uint8_t buf[6];
    buf[0] = CTRL_POINT_PERFORM_NTF_ACTION;
    memcpy(&buf[1], &uid, 4);
    buf[5] = action;
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
    ancs_c_write_control_point(0, buf, 6);
}
```

4.2.4.3 交互

为方便用户进行ANCS Client中的交互测试，本示例实现了基于按键的命令操作，用户按下指定的按键可对Control Point进行操作。

路径：工程目录下的user_app/user_gui.c

名称：app_key_evt_handler();

下列函数提供了开发板中按键触发响应处理的流程。当用户按下指定按键，系统会发生对应的交互命令。

具体按键对应的功能如下：

- OK按键：获取各类属性值，并进行打印。比如获取短信的详细内容，查询邮件的详细内容，发送时间等。
- RIGHT按键：表示Yes，即同意。例如来电时，Yes表示接听电话。
- LEFT按键：表示No，即拒绝。例如来电时，No表示拒绝接听电话。

命令的具体测试方法，请参考Apple Notification Center Service (ANCS) Specification。

```
void app_key_evt_handler(uint8_t key_id, app_key_click_type_t key_click_type)
{
    uint16_t uid;
    if (key_click_type == APP_KEY_SINGLE_CLICK)
    {
        if (BSP_KEY_OK_ID == key_id)
        {
            pwr_mgmt_mode_set(PMR_MGMT_IDLE_MODE);
            uid = ancs_get_uid();

            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_APP_IDENTIFIER);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_TITLE);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_SUBTITLE);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_MESSAGE);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_MESSAGE_SIZE);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_DATE);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_POSITIVE_ACTION_LABEL);
            ancs_notify_attr_get(uid, ANCS_NOTIF_ATTR_ID_NEGATIVE_ACTION_LABEL);
        }
        else if (BSP_KEY_LEFT_ID == key_id)
        {
            APP_LOG_INFO("pressed key left");
            uid = ancs_get_uid();
            if (uid > 0)
            {
                ancs_action_perform(uid, ACTION_ID_NEGATIVE);
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

```
            }
        }
        else if (BSP_KEY_RIGHT_ID == key_id)
        {
            APP_LOG_INFO("pressed key right");
            uid = ancs_get_uid();

            ancs_action_perform(uid, ACTION_ID_POSITIVE);
        }
    }
}
```

版权所有 © 2020 深圳市汇顶科技股份有限公司

5 常见问题

本章描述了在验证ANCS Client示例时，可能出现的问题、原因及处理方法。

5.1 串口终端没有输出信息

- 问题描述

  终端没有打印信息，或者打印乱码。

- 问题分析

  程序没有正确烧录到板子运行，或者串口波特率不正确导致信息不能正常打印在终端。

- 处理方法

  请检查波特率是否为115200，数据位为8，停止位为1，无校验位，无流控，同时检查串口线接入是否正确。

  如果串口正常，请重新烧录固件，同时保证工程没有任何改动，使用GProgrammer直接下载固件。

5.2 手机搜索不到蓝牙广播

- 问题描述

  打开手机App后，无法搜索到以Goodix_ANCS_C为名称的广播。

- 问题分析

  蓝牙天线连接或固件异常。

- 处理方法

  1. 请确定手机蓝牙功能是否打开。如果打开，请检查GR551x平台天线是否连接成功。
  2. 如果均为正常，用户可通过下载默认出厂测试固件来检查硬件是否异常。

5.3 连接成功后无法获取到通知

- 问题描述

  手机连接BLE设备成功后无法获取到通知。

- 问题分析

  用户关闭蓝牙连接，设备可能曾经连接过，或手机通知功能未打开。

- 处理方法

  1. 在手机“设置”中查看是否曾经连接过本蓝牙设备。如果是，用户可以点击忽略此设备，重新扫描、配对和绑定。
  2. 确认已打开手机应用的通知功能。

版权所有 © 2020 深圳市汇顶科技股份有限公司