



GR5xx ANCS Profile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：深圳市福田区保税區腾飞工业大厦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：

- 芯片用户
- 开发人员
- 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、GRUart获取方式。

目录

前言.....	1
1 简介.....	1
2 Profile概述.....	2
3 初次运行.....	3
3.1 准备工作.....	3
3.2 固件烧录.....	3
3.3 测试验证.....	4
4 应用详解.....	9
4.1 运行流程.....	9
4.2 关键代码.....	10
4.2.1 获取通知属性.....	10
4.2.2 执行操作.....	11
4.2.3 交互.....	11
5 常见问题.....	13
5.1 串口终端没有输出信息.....	13
5.2 手机搜索不到蓝牙广播.....	13
5.3 连接成功后无法获取到通知.....	13

1 简介

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

本文档主要介绍如何在GR5xx平台实现ANCS Client。

在进行操作前，可参考以下文档。

表 1-1 文档参考

名称	描述
ANCS Specification	ANCS协议: Apple Notification Center Service (ANCS) Specification
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: 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: https://www.keil.com/support/man/docs/uv4/

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：低功耗蓝牙设备为外围设备（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：

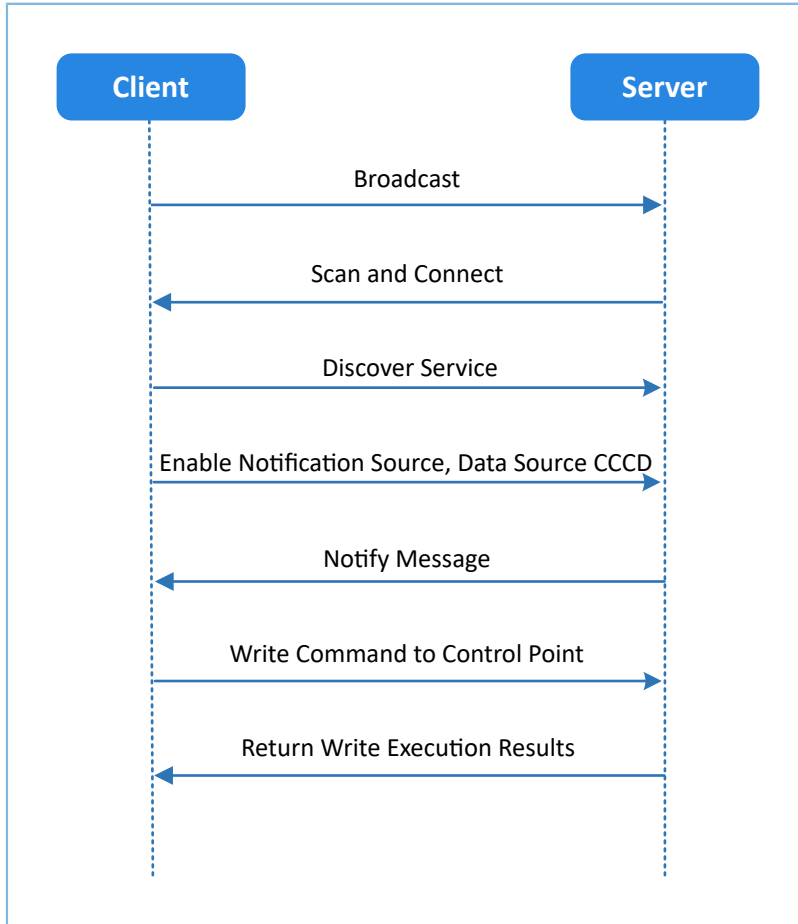


图 2-1 Client & Server交互流程图

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

3 初次运行

本章介绍如何快速验证GR5xx SDK中的ANCS Client示例。

说明:

SDK_Folder为对应芯片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
开发板	对应芯片Starter Kit开发板（以下简称“开发板”）
iOS设备	支持Bluetooth LE4.0及以上的iOS设备，如iPhone 4s及其以上版本、iPad 3及其以上版本
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/
GProgrammer（Windows）	Programming工具，下载网址： www.goodix.com/zh/software_tool/gprogrammer_ble
GRUart（Windows）	串口调试工具，下载网址： www.goodix.com/zh/download?objectId=64&objectType=software

3.2 固件烧录

ANCS示例工程的源码位于SDK_Folder\projects\ble\ble_peripheral\ble_app_ancs_c。

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

说明:

ble_app_ancs_c.bin位于SDK_Folder\projects\ble\ble_peripheral\ble_app_ancs_c\build\。

3.3 测试验证

1. 建立连接:

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 3-1所示。

说明:

本文档使用iOS设备进行测试。若页面有差异，请以实际界面情况为准。

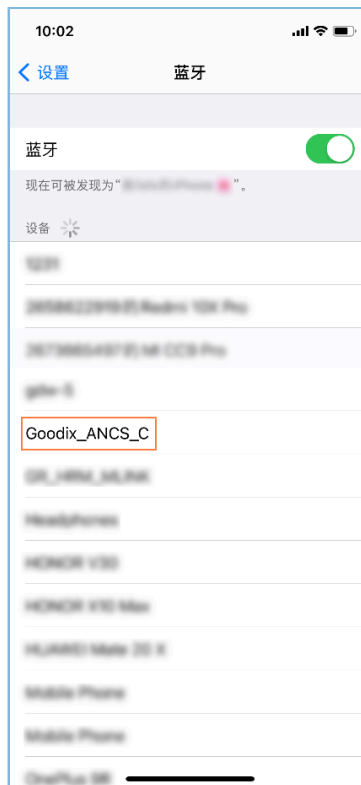


图 3-1 发现广播名Goodix_ANCS_C

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”后点击“配对”。



图 3-2 输入配对信息

配对成功后，提示：Goodix_ANCS_C已连接。

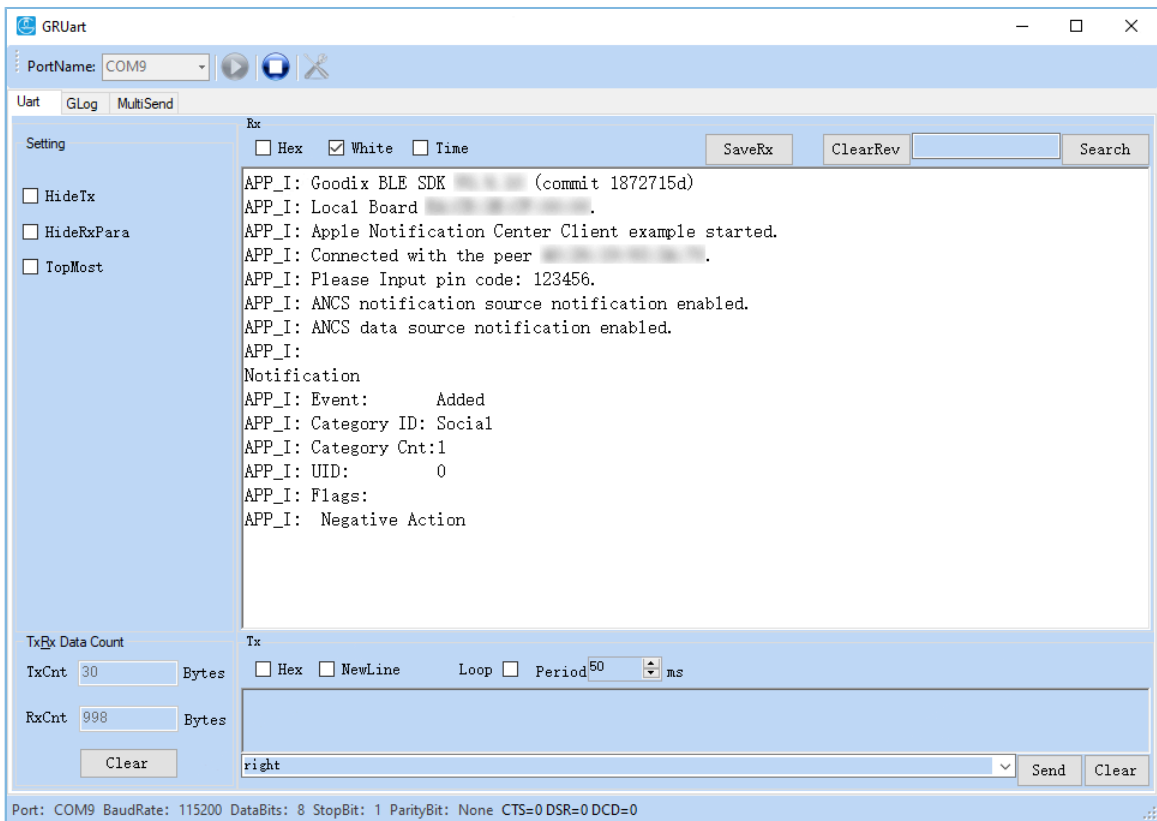


图 3-4 串口打印信息

串口输出信息解释如下：

表 3-3 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
UID: 0	唯一标识符（UID）：0
Flags	信息类型
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，通过串口工具输入“ok”并发送，GRUart将显示以下信息（包括当前拨入的电话号码），此时GRUart中“NewLine”不勾选，如图 3-5所示。

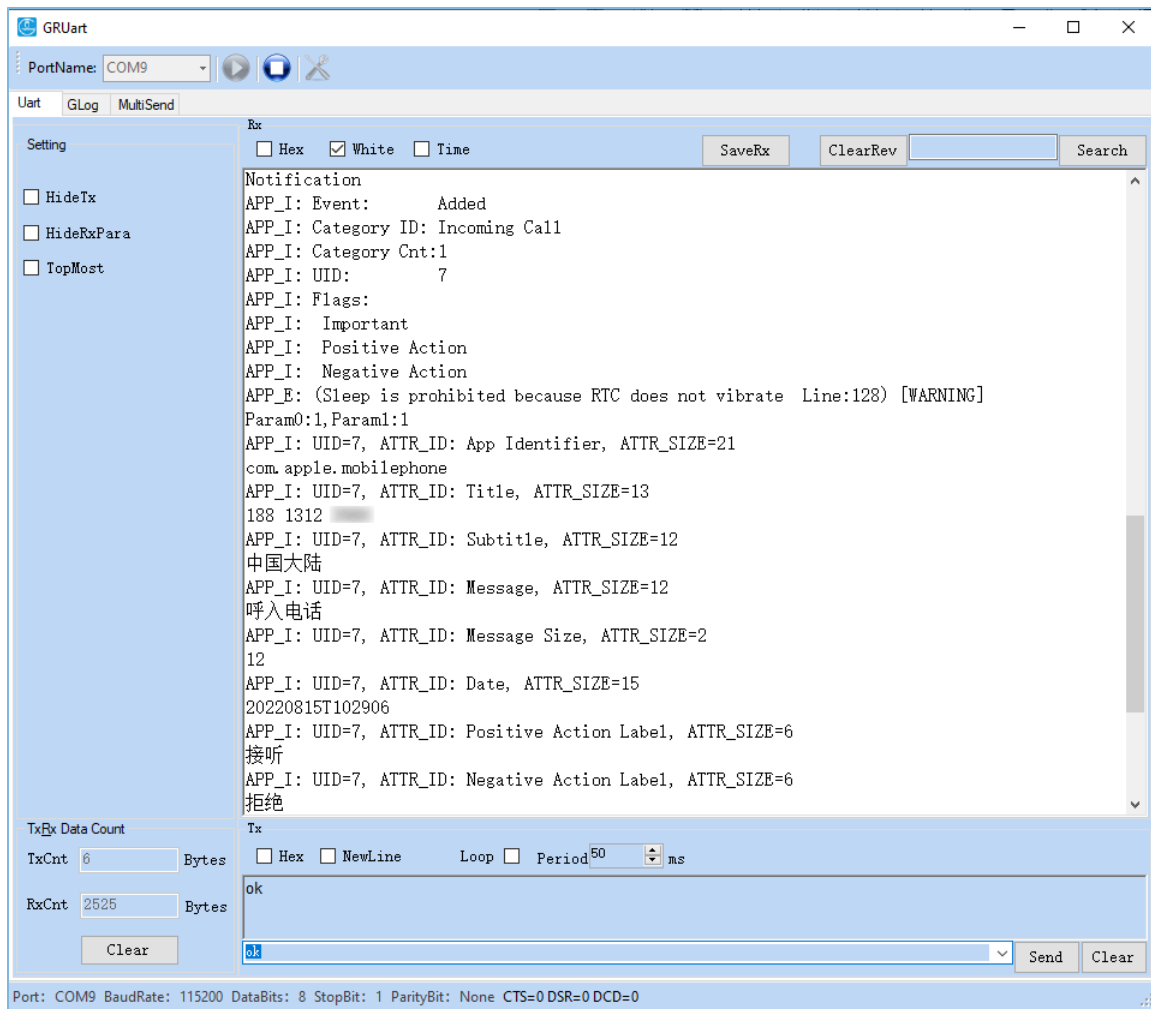


图 3-5 打印通知信息

如用户选择接听，可以通过串口输入“right”并发送，此时可查看到电话已接通；如用户拒绝接听，可以通过串口输入“left”并发送，此时可查看到电话已挂断。

4 应用详解

本章将介绍ANCS Client示例的运行流程和关键代码。

4.1 运行流程

ANCS Client示例开始运行后，将依次执行广播、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等操作。本节以ANCS Client为例，列举出的交互流程如下图所示。

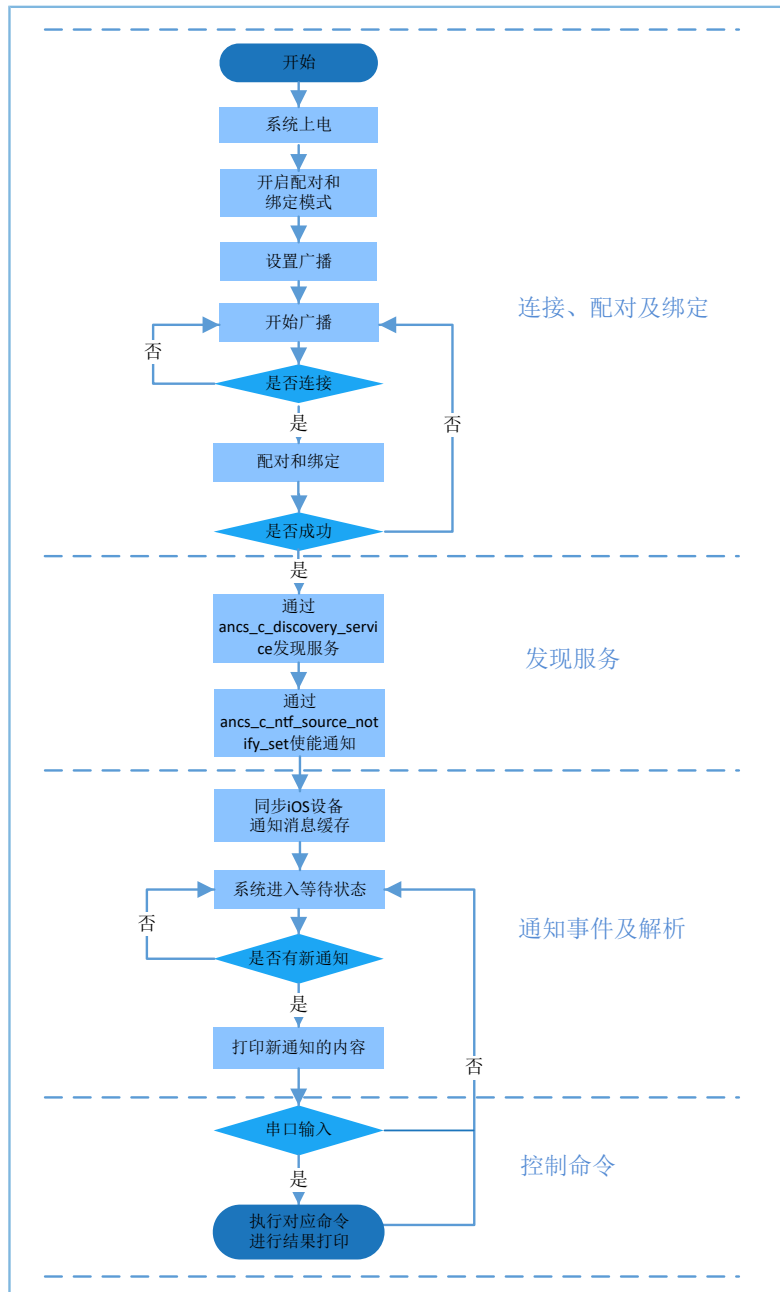


图 4-1 ANCS交互流程图

4.2 关键代码

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。下文将详细介绍相关的控制命令的关键代码。

4.2.1 获取通知属性

路径：SDK_Folder\components\profiles\ancs_c\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：SDK_Folder\components\profiles\ancs_c\ancs_protocol.c

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);
}
}

```

4.2.2 执行操作

路径: SDK_Folder\components\profiles\ancs_c\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: SDK_Folder\components\profiles\ancs_c\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;
    ancs_c_write_control_point(0, buf, 6);
}

```

4.2.3 交互

为方便用户进行ANCS Client中的交互测试, 本示例实现了基于串口的命令操作, 用户发送指定的串口命令可对Control Point进行操作。

路径: 工程目录下的Src\platform\user_periph_setup.c

名称: app_key_evt_handler();

当串口事件处理函数接收到用户输入的串口指令时会对其进行解析, 并将解析结果传给app_key_evt_handler(), 该函数提供了串口输入触发响应处理的流程。当用户通过串口输入特定指令, 系统会发生对应的交互命令。

具体串口命令对应的功能如下:

- **ok**: 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**: 表示Yes，即同意。例如来电时，Yes表示接听电话。
- **left**: 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 (VIR_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 (VIR_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);
            }
        }
        else if (VIR_KEY_RIGHT_ID == key_id)
        {
            APP_LOG_INFO("pressed key right");
            uid = ancs_get_uid();
            ancs_action_perform(uid, ACTION_ID_POSITIVE);
        }
    }
}
```


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GR5xx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

手机连接低功耗蓝牙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。