

## 630X 开发指南

### 1. 工具安装

编辑器: codeblocks-16.01mingw-setup.exe

编译器: jl\_toolchain\_update\_2.4.2.exe(需要授权才能使用)

(安装文档: 杰理工具链注册码获取流程(2.2.6 版本以后).pdf)

包管理工具: 杰理包管理器-setup-1.0.13.exe

gatt 生成工具: gatt\_inc\_generator.exe

OTA 工具: otasdk\_20200420-release.apk

强制下载工具驱动: 3 分钟安装使用 USB Update V4.0

公版 SDK 和工具可在网盘内下载:

获取地址:

链接: [https://pan.baidu.com/s/1XbI8sD0\\_-7EXJZkn0b7Bmg](https://pan.baidu.com/s/1XbI8sD0_-7EXJZkn0b7Bmg)

提取码: hjfw

### 2. 各封装对应工程文件

632X 系列 -> bd19 -> sdk\apps\spp\_and\_le\board\bd19\AC632N\_spp\_and\_le.cbp

631X 系列 -> bd29 -> sdk\apps\spp\_and\_le\board\bd29\AC631X\_spp\_and\_le.cbp

635X 系列 -> br23 -> sdk\apps\spp\_and\_le\board\br23\AC635N\_spp\_and\_le.cbp

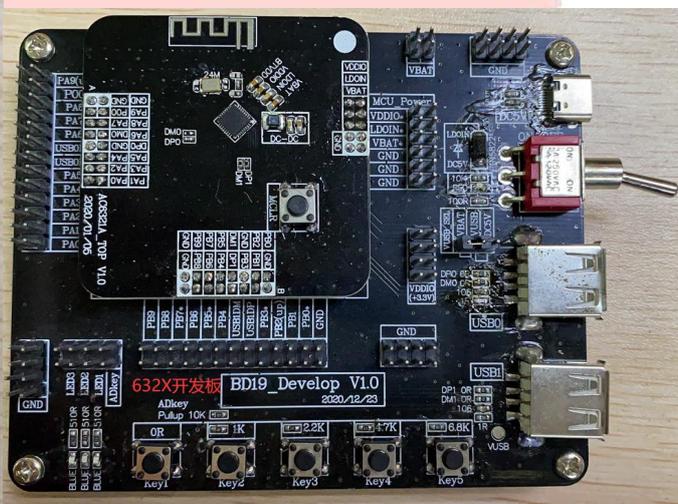
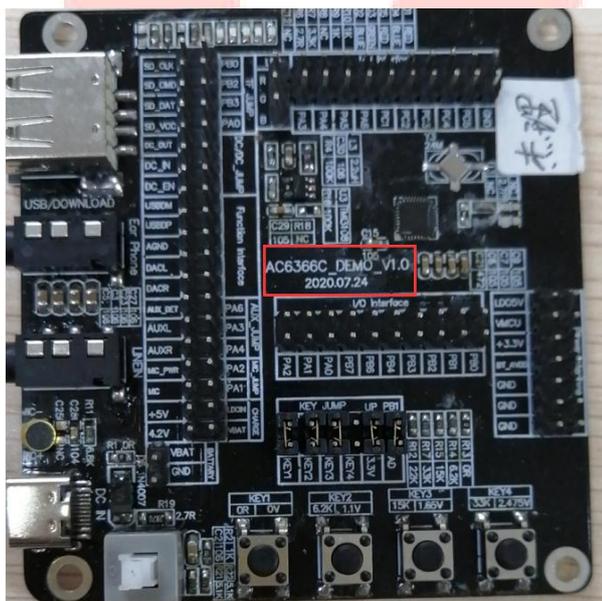
636X 系列 -> br25 -> sdk\apps\spp\_and\_le\board\br25\AC636N\_spp\_and\_le.cbp

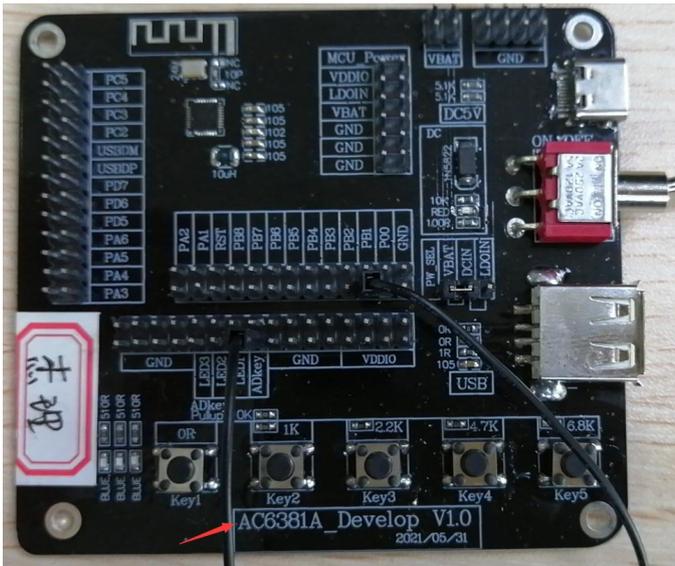
637X 系列 -> br30 -> sdk\apps\spp\_and\_le\board\br30\AC637N\_spp\_and\_le.cbp

638X 系列 -> br34 -> sdk\apps\spp\_and\_le\board\br34\AC638N\_spp\_and\_le.cbp

### 3. 硬件环境

开发板有 3 种, 632X 开发板(BD19), 6366C 开发板, 6381A 开发板。购买板子备注(632X, 或 6366, 或 6381)。635X 和 637X 可以先用耳机的开发板开发, 或者直接做样机板子调试

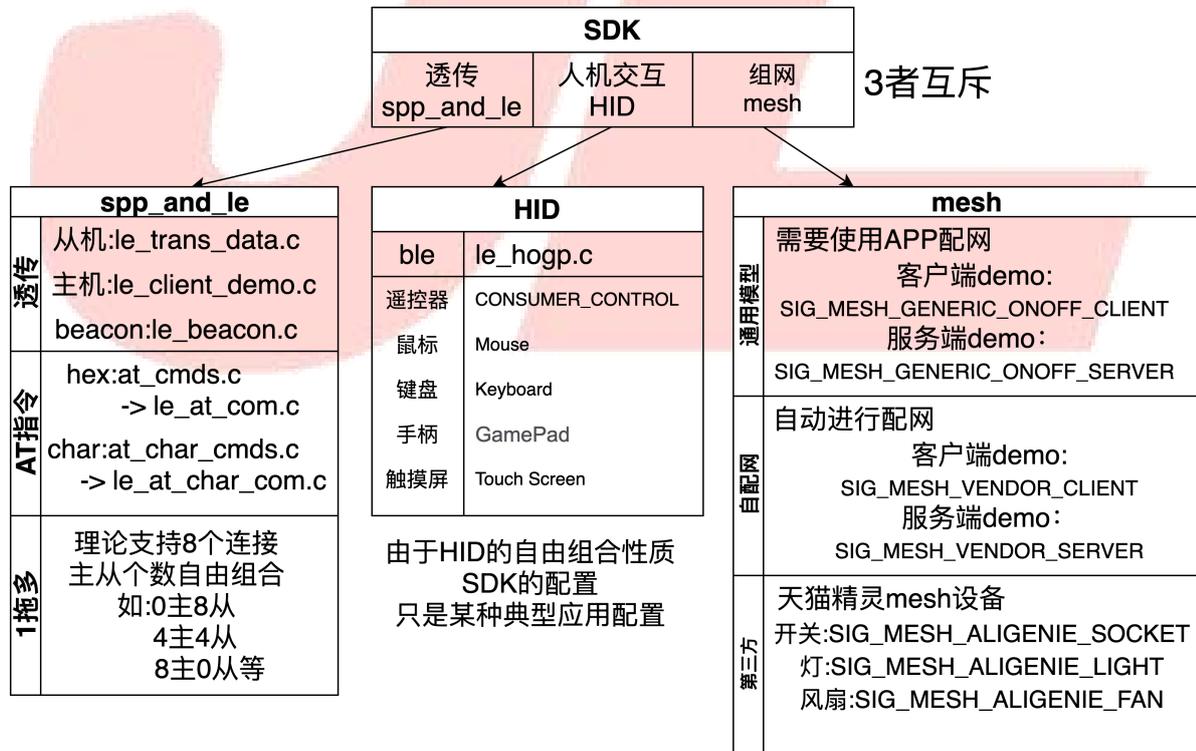




强制下载工具



#### 4. SDK 工程应用架构



## 5. 蓝牙版本及工程配置

- **AC632N\_hid.cbp** 是用于 HID 产品，支持 EDR 或 BLE 模式，

➢ EDR 模式只有 HID 协议，对应的宏配置是：

```
#define TCFG_USER_BLE_ENABLE 0 //BLE 功能使能,  
#define TCFG_USER_EDR_ENABLE 1 //EDR 功能使能  
#define USER_SUPPORT_PROFILE_HID 1
```

➢ BLE 模式，对应的宏配置

```
#define TCFG_USER_BLE_ENABLE 1 //BLE 功能使能,  
#define TCFG_USER_EDR_ENABLE 0 //EDR 功能使能  
#define USER_SUPPORT_PROFILE_HID 0
```

- **AC632N\_spp\_and\_le.cbp** 是用于数传功能，搭配客户 APP 开发，支持 EDR 或 BLE 模式

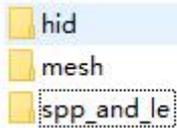
➢ EDR 模式只有 SPP 协议，对应的宏配置是：

```
#define TCFG_USER_BLE_ENABLE 0 //BLE 功能使能,  
#define TCFG_USER_EDR_ENABLE 1 //EDR 功能使能  
#define USER_SUPPORT_PROFILE_SPP 1
```

➢ BLE 模式，对应的宏配置

```
#define TCFG_USER_BLE_ENABLE 1 //BLE 功能使能,  
#define TCFG_USER_EDR_ENABLE 0 //EDR 功能使能  
#define USER_SUPPORT_PROFILE_SPP 0
```

## 6. BLE 设备类型选择



有 3 种工程，3 者选 1。

spp\_and\_le 透传工程有 3 种应用

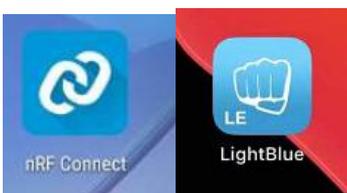
```
#define TRANS_DATA_EN 1 //蓝牙双模透传  
#define TRANS_CLIENT_EN 0 //蓝牙(ble主机)透传  
#define BEACON_MODE_EN 0 //蓝牙BLE ibeacon
```

3 者择 1。注意：以上 3 个类型，**互斥使用**，只能打开一个

## 7. 普通透传测试---做数传搭配 APP 通信

如果没使用过，不知固件，怎么跟手机通信，请先以公版 SDK 测试，和网上的标准测试 BLE 的 APP，进行通信测试，再进行下一步的案子开发。

- 手机端，安卓安装 NRF connect APP。苹果安装 lightblue。



- 代码配置成数传 demo

AC632N\_spp\_and\_le.cbp

打开透传的工程

app\_config.h 配置文件

```
#define CONFIG_DEBUG_LIB (x & LIB_DEBUG)
#define CONFIG_DEBUG_ENABLE
```

使能打印

```
//app case 选择,只能选1个,要配置对应的board_config.h
#define CONFIG_APP_SPP_LE 1 //SPP + LE or LE's client
#define CONFIG_APP_AT_COM 0 //AT com HEX格式命令
#define CONFIG_APP_AT_CHAR_COM 0 //AT com 字符串格式命令
#define CONFIG_APP_DONGLE 0 //board_dongle ,TCFG_PC_ENABLE
#define CONFIG_APP_MULTI 0 //蓝牙LE多连

//配置对应的APP的蓝牙功能
#if CONFIG_APP_SPP_LE
#define TRANS_DATA_EN 1 //蓝牙双模透传
#define TRANS_CLIENT_EN 0 //蓝牙(ble主机)透传
#define BEACON_MODE_EN 0 //蓝牙BLE ibeacon
#define XM_MMA_EN 0
```

普通透传应用

board\_config.h 板级配置文件

```
/*
 * 板级配置选择
 */

#define CONFIG_BOARD_AC632N_DEMO
// #define CONFIG_BOARD_AC6321A_DEMO
// #define CONFIG_BOARD_AC6323A_DEMO
// #define CONFIG_BOARD_AC6328A_DEMO
// #define CONFIG_BOARD_AC6328B_DONGLE
// #define CONFIG_BOARD_AC6329B_DEMO
// #define CONFIG_BOARD_AC6329C_DEMO
// #define CONFIG_BOARD_AC6329E_DEMO
// #define CONFIG_BOARD_AC6329F_DEMO

#include "board_ac632n_demo_cfg.h"
#include "board_ac6321a_demo_cfg.h"
#include "board_ac6323a_demo_cfg.h"
#include "board_ac6328a_demo_cfg.h"
#include "board_ac6328b_dongle_cfg.h"
#include "board_ac6329b_demo_cfg.h"
#include "board_ac6329c_demo_cfg.h"
#include "board_ac6329e_demo_cfg.h"
#include "board_ac6329f_demo_cfg.h"
```

根据芯片型号,选择其他一个宏,对应一个头文件

```
*****//
//          UART配置          //
*****//
#define TCFG_UART0_ENABLE          ENABLE_THIS_MOUDLE
#define TCFG_UART0_RX_PORT        NO_CONFIG_PORT
#define TCFG_UART0_TX_PORT        IO_PORT_DP//IO_PORTA_05
#define TCFG_UART0_BAUDRATE       1000000
```

建议修改成 USB DP 打印

```
#define TCFG_USER_TWS_ENABLE      0    //tws功能使能
#define TCFG_USER_BLE_ENABLE     1    //BLE功能使能,--
#define TCFG_USER_EDR_ENABLE     0    //EDR功能使能  建议关闭 EDR
```

数传使用的是 le\_trans\_data.c 文件

```
static void ae02_send(void *priv)
{
    static u8 data[20];
    static u8 cnt = 0;
    if(++cnt > 20) cnt = 0;
    memset(data, 0x00, sizeof(data));
    memset(data, cnt, cnt);
    app_send_user_data(ATT_CHARACTERISTIC_ae02_01_VALUE_HANDLE, data, cnt, ATT_OP_AUTO_READ_CCC);
}

static int att_write_callback(hci_con_handle_t connection_handle, uint16_t att_handle, uint16_t transaction_mode)
{
    int result = 0;
    u16 tmp16;

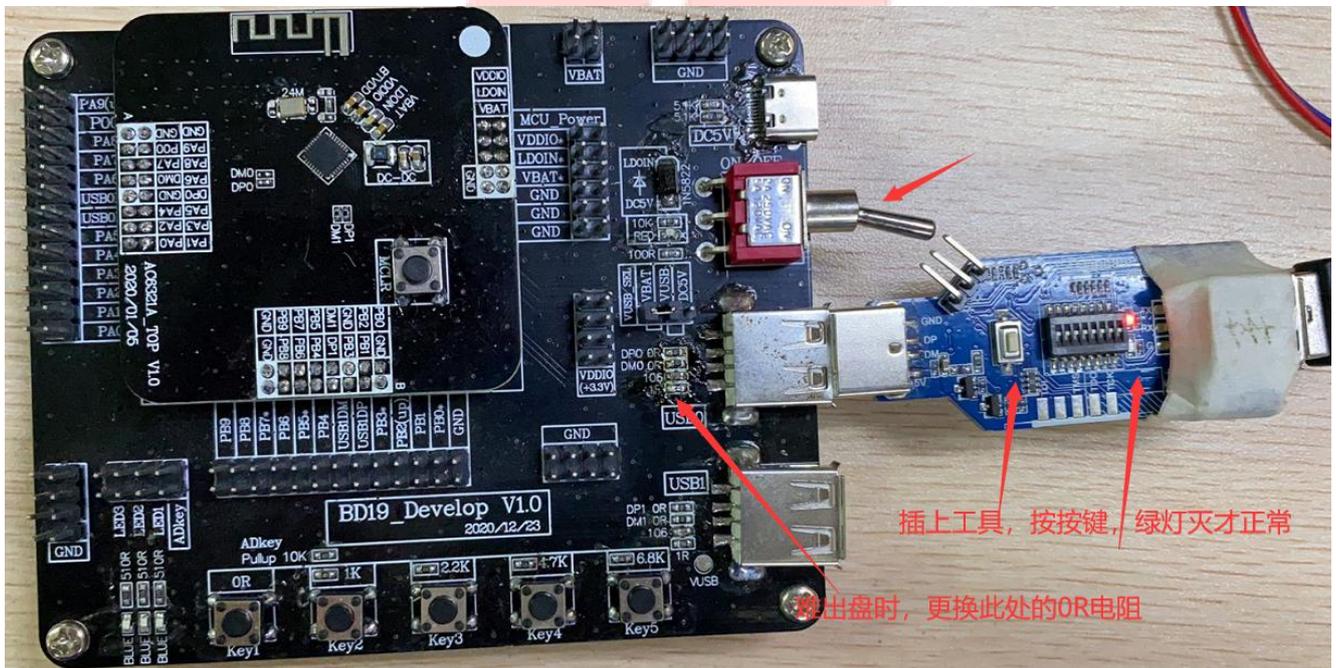
    u16 handle = att_handle;

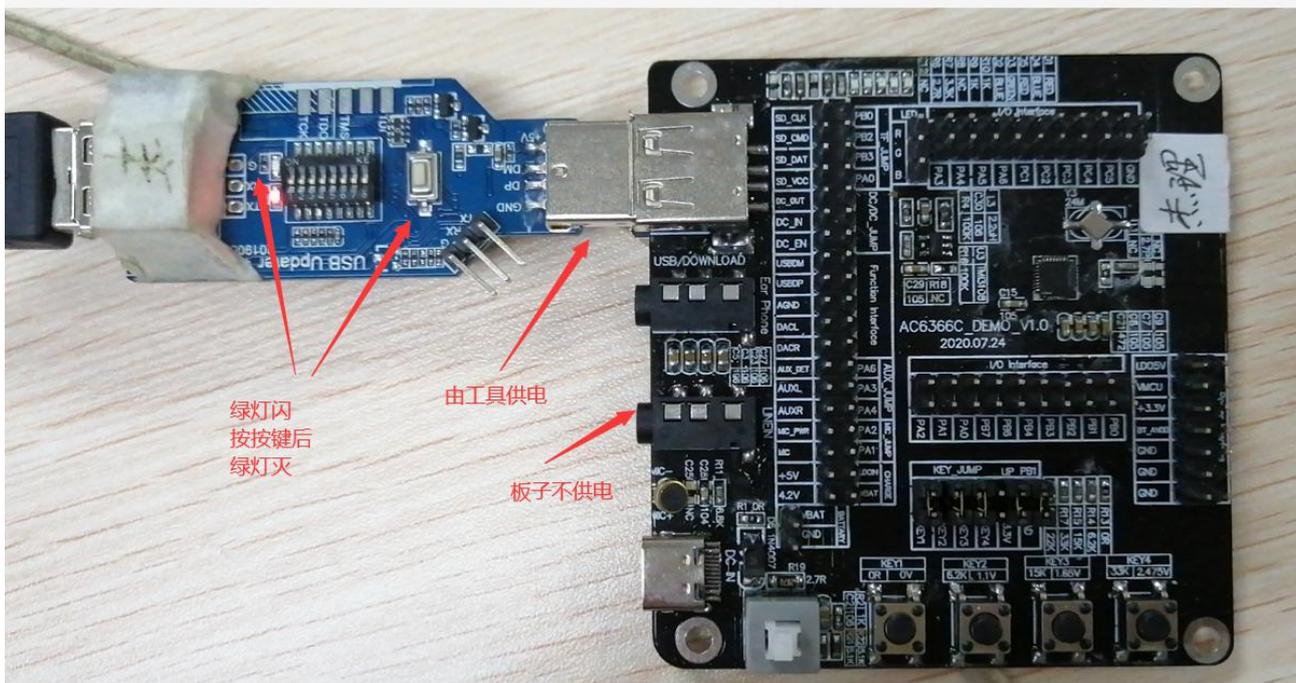
    r_printf("write_callback, handle= 0x%04x,size = %d\n", handle, buffer_size);

    switch (handle) {
        case ATT_CHARACTERISTIC_2a00_01_VALUE_HANDLE:
            break;

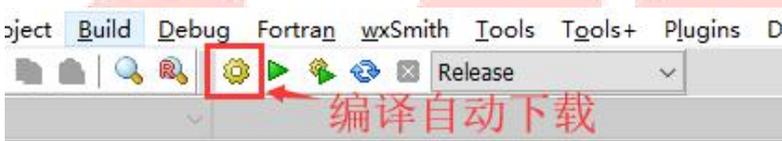
        case ATT_CHARACTERISTIC_ae02_01_CLIENT_CONFIGURATION_HANDLE:
            printf("open_ae02_notify");
            sys_timer_add(NULL, ae02_send, 2000);
    }
}
```

添加 notify 定时发送函数，方便测试。





- 电脑连接工具再接入开发板



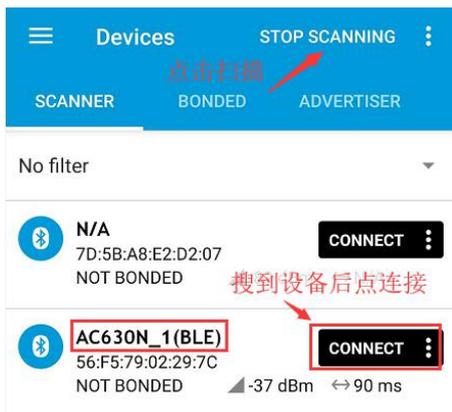
直接编译就能下载

```

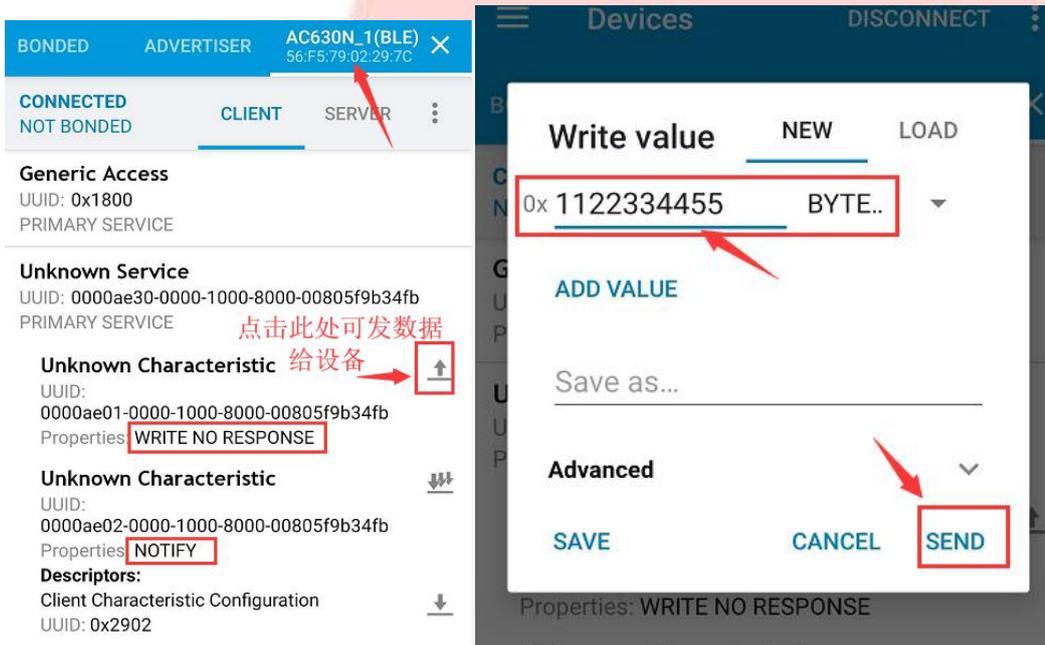
开始下载 .....
Write sector:41 ..
40 .
39 .
38 .
37 .
36 .
35 .
34 .
33 .
32 .
Write block:1 ..
0 .
下载完毕.
    
```

有如下提示表示下载完成

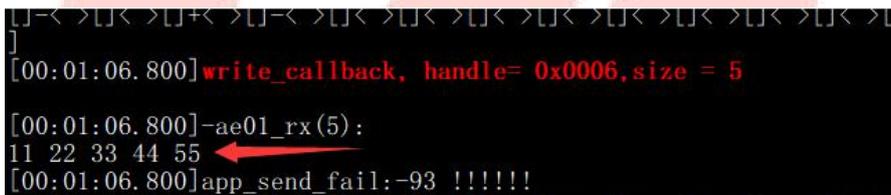
- 开发板上电开机, 就可以手机打开 NRF 的 APP, 搜索 BLE 设备了



不同系列，默认蓝牙名就是芯片系列名字



手机发送数据后，蓝牙设备会有如下打印：



蓝牙设备想要发送数据给手机，需要手机先打开 notify 通道





- gatt\_inc\_generator.exe
- gatt\_inc\_generator\_exe.h
- gatt\_profile.cfg
- gatt\_profile\_cfg.h
- make\_gatt\_services工具说明.pdf
- README.txt

请按照客户定义的需求，模仿下面 demo，自行修改

```
1 PRIMARY_SERVICE, 0x1800
2 CHARACTERISTIC, 2A00, READ | DYNAMIC,
3 CHARACTERISTIC, 2A01, READ | DYNAMIC,
4
5 PRIMARY_SERVICE, 0x1801
6 CHARACTERISTIC, 2A05, READ | DYNAMIC,
7
8 PRIMARY_SERVICE, 0000ffe0-0000-1000-8000-00805f9b34fb
9 CHARACTERISTIC, 0000ffe1-0000-1000-8000-00805f9b34fb, WRITE | READ | NOTIFY | WRITE_WITHOUT_RESPONSE | DYNAMIC,
10 CHARACTERISTIC, 0000ffe5-0000-1000-8000-00805f9b34fb, WRITE | NOTIFY | DYNAMIC,
11
12 PRIMARY_SERVICE, 0x180A
13 CHARACTERISTIC, 2A25, READ | DYNAMIC,
14 CHARACTERISTIC, 2A28, READ | DYNAMIC,
15 CHARACTERISTIC, 2A27, READ | DYNAMIC,
16 CHARACTERISTIC, 2A29, READ | DYNAMIC,
17 CHARACTERISTIC, 2A24, READ | DYNAMIC,
18
19 PRIMARY_SERVICE, 0000fee7-0000-1000-8000-00805f9b34fb
20 CHARACTERISTIC, 0000fec7-0000-1000-8000-00805f9b34fb, WRITE | DYNAMIC,
21 CHARACTERISTIC, 0000fec8-0000-1000-8000-00805f9b34fb, INDICATE | DYNAMIC,
22 CHARACTERISTIC, 0000fec9-0000-1000-8000-00805f9b34fb, READ | DYNAMIC,
23
```

此段为标准服务，必须保留

此段为标准服务，建议保留

16bit自定义复位  
参考设置  
可以任意多项

128bit自定义服务  
参考设置  
可以任意多项

所有特征属性都或上一个动态属性  
才会回调应用层

910 gatt\_profile.cfg Conf [Space]

128bits 的 UUID 写法，参照截图。

除标准服务，建议保留外，其他服务，根据产品需求任意裁剪，或添加。