



密级：公开资料

TTC 2541 SDK 使用说明

文档版本：V1.2

深圳市昇润科技有限公司

2017年09月20日

版权所有

版本	修订日期	修订人	审稿人	修订内容
1.0	2017-05-31	郭高亮		初版发布
1.1	2017-08-31	陈炽华	张眼	1. 增加大容量 OAD 操作说明。 2. 增加快速传输操作说明。 3. 增加微信功能说明。
1.2	2017-09-20	陈炽华	张眼	1. 增加 SBL 操作说明

公开资料

目 录

1. 概述.....	3
1.1. SDK 简介.....	3
1.2. 公共特征.....	3
1.3. TTC SDK 优势.....	3
1.4. SDK 结构图.....	4
1.5. 程序空间.....	7
1.5.1. 从机示例空间.....	7
1.5.3. 主机示例空间.....	7
1.6. SDK 功耗.....	7
2. 工具使用说明.....	8
2.1. 模组连接.....	8
2.2. 手机 APP (TTC-BLE) 使用.....	10
2.3. PC 端软件 (TTC Demo) 使用.....	10
2.4. IAR 常见操作.....	11
2.4.1. IAR 版本.....	11
2.4.2. 工程文件路径.....	11
2.4.3. 宏定义.....	11
2.4.4. 工程配置.....	12
3. TTC SDK API 说明.....	13
3.1. TTC SDK 头文件说明.....	13
3.2. 蓝牙初始化 API (TTCBlePeripheralProcess.h).....	13
3.3. 蓝牙参数及相关蓝牙操作 API (TTCBlePeripheral.h).....	14
3.4. 蓝牙数据发送相关 API (TTCBleProfile.h).....	15
3.5. SDK 线程消息说明.....	16
3.5.1. TTCSDK_MSG_GET_BLE_STATE_EVENT 蓝牙状态处理.....	16
3.5.2. TTCSDK_MSG_GET_BLE_DATA_EVENT 蓝牙接收数据处理.....	16
3.5.3. TTCSDK_MSG_REFRESH_RSSI_EVENT 连接状态下 RSSI 值读取..	17
3.5.4. TTCSDK_MSG_GET_BLE_PARAM_EVENT 协商后的蓝牙参数.....	17
3.6. 设备信息服务参数设置 (TTCBleDevInfoService.h).....	17
3.7. 应用线程公共回调函数说明.....	18
4. 蓝牙角色 Demo 说明.....	20
4.1. 从机示例.....	20
4.1.1 示例功能说明.....	20
4.1.2. 程序编译下载.....	20
4.1.3. 相关代码实现.....	20
4.1.4 IO 功能演示.....	21
4.1.5 快速传输功能演示.....	22

4.1.6 微信功能演示.....	23
4.2. 主机示例.....	28
4.2.1. 示例功能说明.....	28
4.2.2. 程序编译下载.....	28
4.2.3. 相关代码实现.....	28
4.2.4. 功能演示.....	30
5. 空中升级说明 (OAD)	32
5.1. 小容量 OAD	32
5.1.1. 首先下载 BIM 工程.....	32
5.1.2. image A 生成.....	32
5.1.3. image B 生成.....	32
5.1.4. 手机 APP (TTC-BLE) 操作说明	33
5.2. 大容量 OAD	35
5.2.1. 首先下载 LARGE BIM 工程.....	35
5.2.2. 编译并下载 Large image A.....	35
5.2.3. Large image B 生成.....	35
5.2.4. 手机 APP (TTC-BLE) 操作说明	35
6. 串口升级 (SBL)	37
6.1. 首先下载 SBL 工程.....	37
6.2. BIN 文件生成.....	37
6.3. 进行串口升级.....	38
7. 联系我们.....	39
附录 A. 手机 APP 下载.....	40

1. 概述

1.1. SDK 简介

TTC SDK 是由我司提供的 CC2541 快速开发库。旨在让开发人员不再需要将大量精力放在蓝牙调试方面，只需将精力放在对 CC2541 功能上的开发。TTC SDK 提供了蓝牙参数设置、蓝牙数据收发、蓝牙状态处理等 API，同时也提供了测试程序，开发人员无需再设计测试程序。

使用 TTC SDK 能适配我司提供的 TTC-BLE 软件，方便调试数据收发，并且支持数据加密解密功能。能极大的缩短 CC2541 的开发周期。

1.2. 公共特征

- 蓝牙服务 UUID: 1000

蓝牙通道	UUID	通道特性	功能概述
UUID1	1001	Write_NoRsp/Read/Notify	蓝牙数据接收
UUID2	1002	Read/Notify	蓝牙数据发送
UUID3	1003	Write_NoRsp	寄存器写数据
UUID4	1004	Read	寄存器读数据
UUID5	1005	Write_NoRsp/Read	选定寄存器

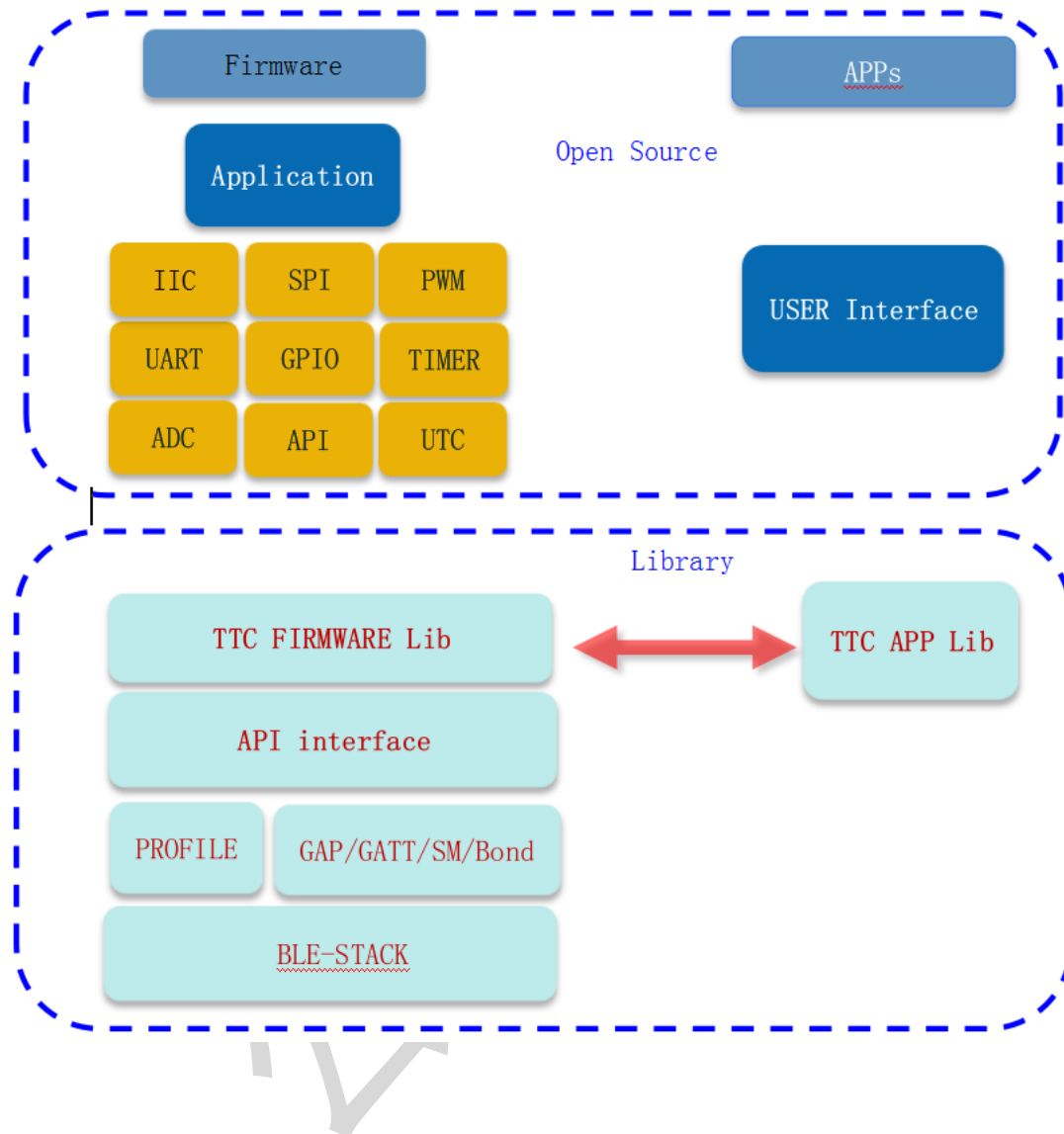
- 128 bit UUID 描述如下
 - Service: 00001000-0000-1000-8000-00805f9b34fb
 - UUID1 : 00001001-0000-1000-8000-00805f9b34fb
 - UUID2 : 00001002-0000-1000-8000-00805f9b34fb
 - UUID3 : 00001003-0000-1000-8000-00805f9b34fb
 - UUID4 : 00001004-0000-1000-8000-00805f9b34fb
 - UUID5 : 00001005-0000-1000-8000-00805f9b34fb

备注：主机只能连接带有服务 UUID 0x1000 的从机设备（如 SDK 从机）。

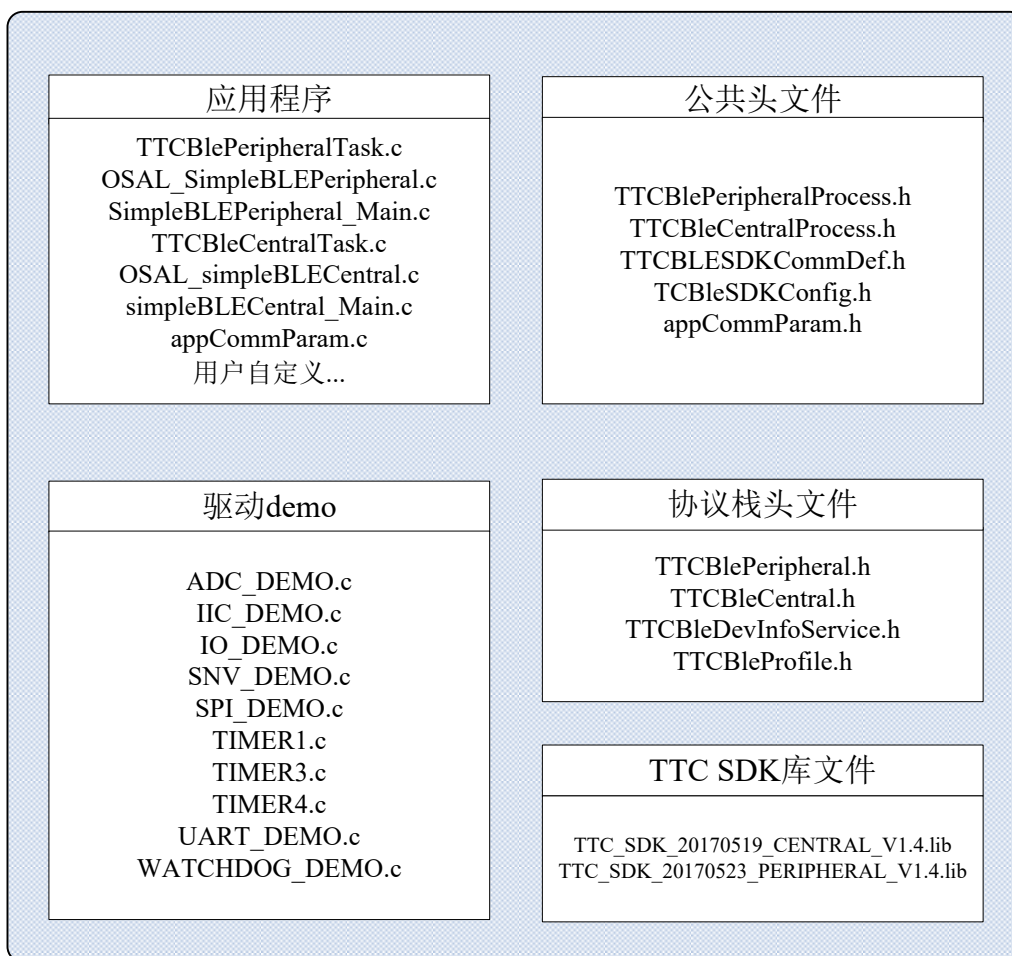
1.3. TTC SDK 优势

- 完整的蓝牙解决方案(IC+固件+ APP+ 云端)
- 简单的蓝牙设置以及轻松更新固件
- 类似串口数据收发的蓝牙交互模式
- 完整的 SDK 以及工具提供（SDK 就绪）
- 快速启动时间（RTOS < 500ms，OSAL < 500ms）
- 超低功耗特性，CC2541 低至 0.3uA，可用电池供电
- 数据支持 AES 加密解密
- 完整的蓝牙参数验证，适应安卓、IOS 两大平台，用户无需烦恼蓝牙参数适配问题
- 完整的测试方案提供，SDK 已包含测试程序，用户无需设计蓝牙测试程序
- 配套的专业测试架、SDK 包、开发调试工具、DEMO 板
- 减少由于蓝牙导致设备工作异常的情况出现

1.4. SDK 结构图



在 SDK 中，各文件之间的关系如下图：



TTC SDK 路径下各个文件介绍：

1. 应用程序

- (1) SimpleBLEPeripheral_Main.c、simpleBLECentral_Main.c
程序入口，主函数中包含硬件、OSAL 初始化，最后启动 OASL 系统。
- (2) OSAL_SimpleBLEPeripheral.c、OSAL_simpleBLECentral.c
OSAL 系统，任务初始化以及任务事件处理函数列表。
- (3) TTCBlePeripheralTask.c、TTCBleCentralTask.c
用户应用程序
- (4) appCommParam.c
公共变量，以及主机 AT 指令的实现及 UART 信息打印。

2. 驱动 demo

IO/ADC/IIC/SNV/SPI/TIMER/UART/WATCHDOG 示例程序。

3. 公共头文件

- (1) TTCBleCentralProcess.h
主机相关操作处理声明

(2) TTCBleDevInfoService.h

此头文件与蓝牙设备信息有关，可以设置读取软件版本、硬件版本、生产厂商等信息。

(3) TTCBlePeripheralProcess.h

从机相关操作处理声明

(4) TTCBleProfile.h

此头文件与蓝牙服务有关，有设置蓝牙通道数据的声明，可通过此函数发送数据至 APP。

4. 蓝牙功能头文件

(1) TTCBeacon.h

SDK 支持 Beacon 功能，此头文件包含 Beacon 相关参数初始值定义(宏定义)，参数设置、读取等相关函数的声明。

(2) TTCBleWechat.h

SDK 支持微信功能，此头文件包含微信相关参数初始值定义(宏定义)，参数设置、读取等相关函数的声明。

5. 协议栈头文件

(1) TTCBleCentral.h

主机角色线程声明

(2) TTCBlePeripheral.h

从机角色线程声明

6. TTC SDK 库文件

(1) TTC_SDK_20170824_V1.6_PERIPHERAL.lib

从机角色对应的库文件；

(2) TTC_SDK_20170824_V1.6_CENTRAL.lib

主机角色对应的库文件；

1.5. 程序空间

SDK 中含有 2 种蓝牙角色，部分角色根据不同的功能有不同的配置，配置区别以及可用空间计算如下表。

说明：

(1) 以下程序空间计算时，主机未使能驱动 Demo 相关功能，即在默认程序的基础上再屏蔽宏定义 TTC_DEBUG 及 TTC_DRIVER_UART；

(2) 程序默认为堆栈分配 3 Kbyte RAM，如宏定义 INT_HEAP_LEN=3072，这 3Kbyte RAM 作为已用 RAM 计算。根据实际需求，可适当调整。

1.5.1. 从机示例空间

OAD	测试程序	ROM(KByte)			RAM(KByte)		
		总量	用量	剩余	总量	用量	剩余
无	无	256	115.4	140.6	8	6.5	1.5
	有		121.3	134.7		6.9	1.1
片内大容量	无	157	114	43	8	6.7	1.3
	有		119.9	37.1		7.2	0.8
片内小容量	无	128	113.9	14.1	8	6.7	1.3
	有		119.8	8.2		7.2	0.8

1.5.3. 主机示例空间

OAD	测试程序	ROM(KByte)			RAM(KByte)		
		总量	用量	剩余	总量	用量	剩余
无	无	256	121	135	8	7.2	0.8

1.6. SDK 功耗

说明：以下功耗测试，均未使用 UART 等驱动。

1. 从机角色

蓝牙状态	设置参数	实际间隔时间 (ms)	平均电流 (uA)
关闭广播	\	\	0.42
广播间隔	32	20	2190
	160	100	463.5
	800	500	98.5
	1600	1000	57.2
连接间隔	16	20	1320
	24	30	843.3
	80	100	269.4
	160	200	145.0
	400	500	
	800	1000	

2. 工具使用说明

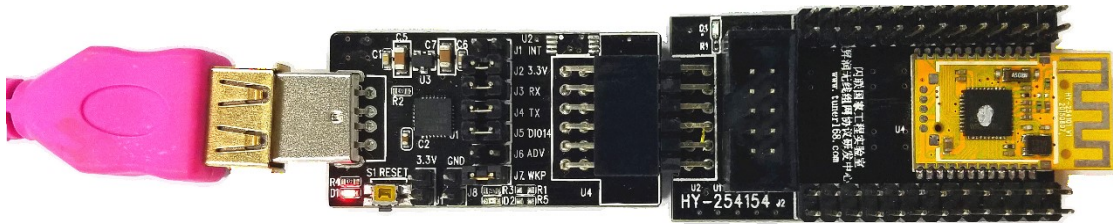
手机 APP (TTC-BLE) 下载二维码方式见[附录 A](#)。

2.1. 模组连接

SDK 主机 demo 相关功能是使用 AT 指令的方式实现的，若需要演示 dmeo 功能，则需要使用到 USB 转串口工具，以下介绍两种连接方式。

➤ 方式一：使用 USB 转接板

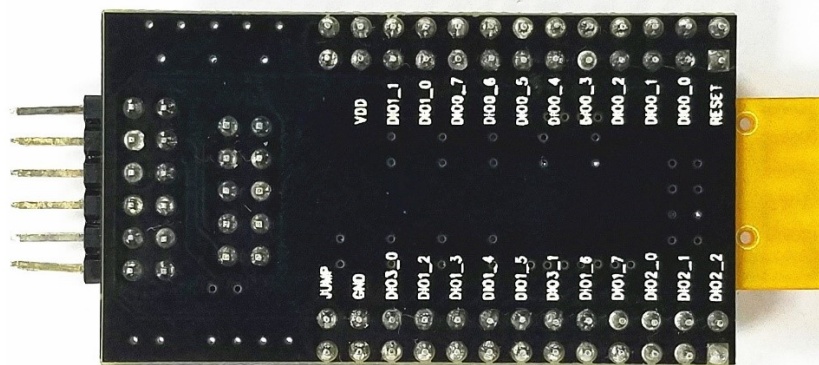
CC2541 模组通过 USB 转接板（需用跳线帽短接 J2-3.3V、J3-RX、J4-TX、J7-WKP），连接至电脑 USB 接口，S1 (RESET) 为复位按钮，如下图。从机上电后，demo 默认广播名称为”TTC_CC2541_SDK”。



➤ 方式二：使用其他 USB 转串口工具

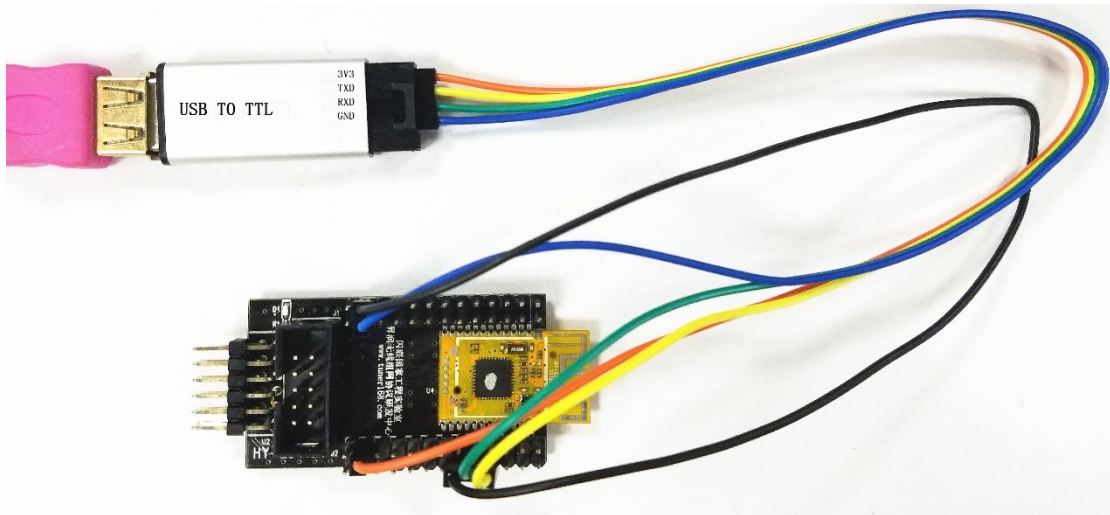
如果没有以上的 USB 转接板，则可以使用常见的 [USB 转 UART 的小工具](#) 实现连接。需注意，模块供电范围是 2.0V~3.8V，可以使用 3.3V 供电，RX/TX 信号的高电平也应对应的 3.3V。模组与小工具需要 5 条杜邦线连接，分别是 VCC/GND/WAKEUP/RX/TX。

HY-254101 V1 背面标明引脚排序如下：



在测试时，可将透传模组 WAKEUP (DI00_4) 引脚接 GND，使模组一直处于唤醒状态，便于测试。最终连接如下：

USB to TTL 工具	HY-254101 V1 (CC2541 模组)
3V3	VDD
TXD	RX (DI00_2)
RXD	TX (DI00_3)
\	WAKEUP (DI00_4) ←→ GND
GND	GND

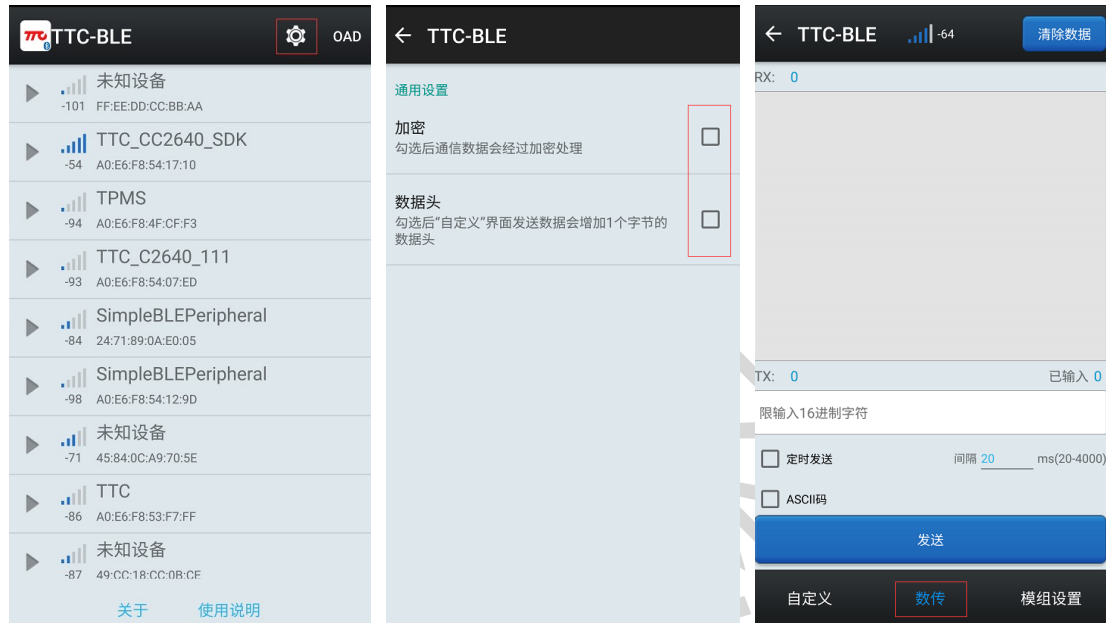


2.2. 手机 APP (TTC-BLE) 使用

安装手机 APP (TTC-BLE)，进入扫描界面，如下图 1；

点击设置，取消勾选“加密”（TTC SDK demo 在传输数据时默认设置为不加密）、“数据头”，如下图 2；

点击相应的蓝牙设备，如“TTC_CC2541_SDK”，即可与设备建立链接，进入“数传”界面，如下图 3。

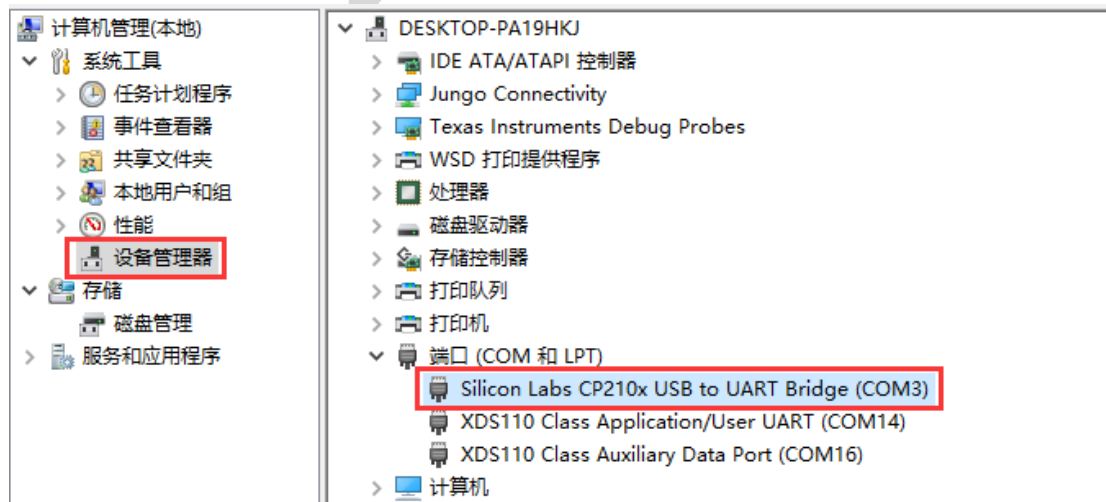


2.3. PC 端软件 (TTC Demo) 使用

(1) 串口设置 (UART)

选择对应的串口号，数据格式为：长度 8bit，无校验位，1bit 停止位，设置波特率为 115200bps (TTC SDK demo 默认波特率)，点击“打开串口”，如下图。

设备管理器查看 UART 端口号，选择 Silicon Labs Cp210x USB to UART Bridge 对应的端口号，请勿选择 XDS110 虚拟出的端口，如下图：



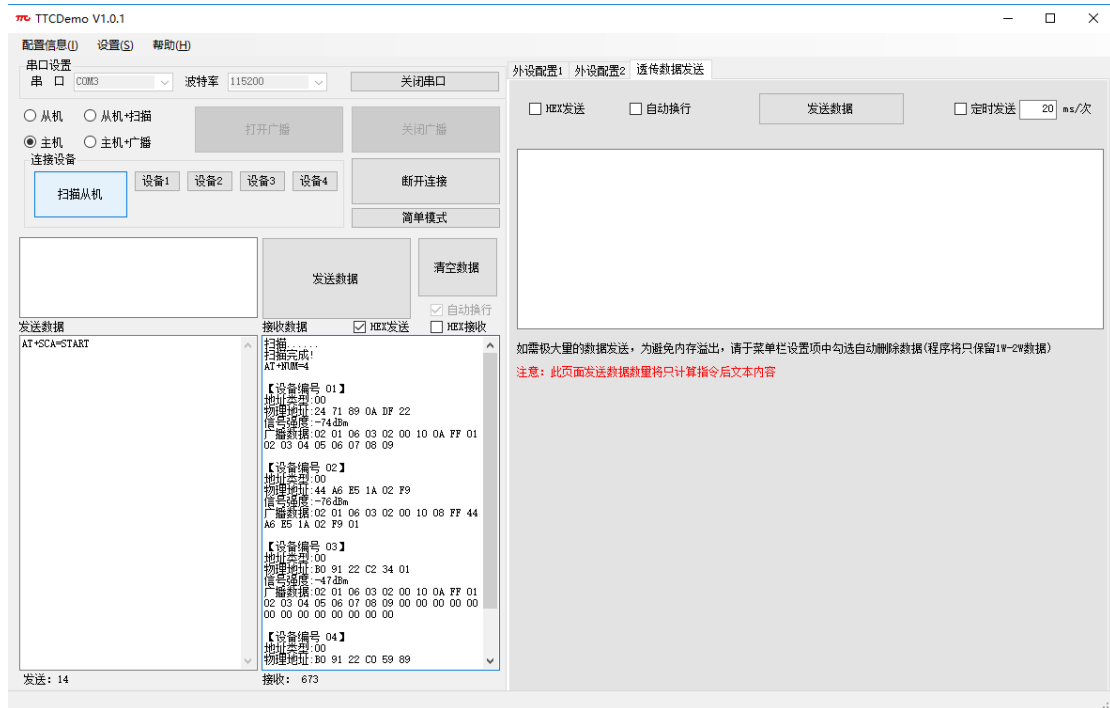
(2) 蓝牙角色选择

根据 demo 程序所实现的蓝牙角色，设置 PC 端软件(TTCDemo)对应的角色。不同角色时，所支持的 AT 指令不一样，灰色为不可使用的指令。SDK 支持从机角色以及主机角色，但不支持通过 AT 指令切换主从机，如下图。

(3) 外设配置

“外设配置 1”包含 GPIO、ADC 及定时器的设置，“外设配置 2”包含 PWM、IIC、SPI 及 UTC 的配置，2541 SDK 不支持此功能，如下图。

(4) 透传数据发送



2.4. IAR 常见操作

2.4.1. IAR 版本

TTC 2541 SDK V1.x 使用 IAR 8051 9.20 及以上版本，开启 IAR 时需要以管理员身份运行。

2.4.2. 工程文件路径

- (1) IAR WorkSpace 文件：SDK.eww
- (2) 所在路径：\Projects\ble\TTCbleSDK_2541\CC2541DB

2.4.3. 宏定义

蓝牙角色 demo 及驱动 demo 均需要调整部分宏定义，操作步骤如下：

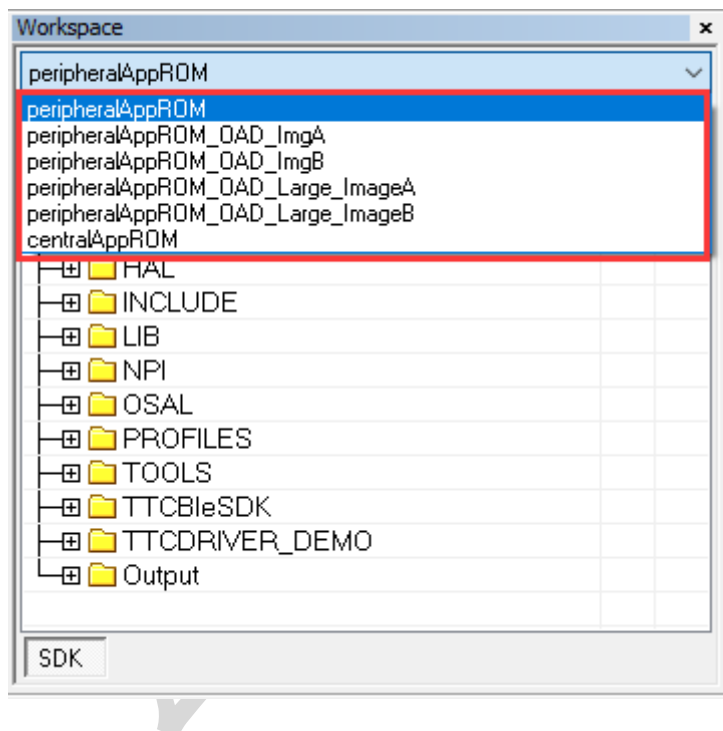
IAR -> Project -> Options -> C/C++ Compiler -> Preprocessor -> Defined symbols 选项中定义，例如 TTCDRIVER_IO 为 IO demo 运行所需要的宏，删除宏如 TTCDRIVER_IOx，以便再次开启。

2.4.4. 工程配置

工程含有 6 个配置：

注意：带 OAD 功能的配置，不支持在线调试仿真，不可直接使用 IAR 下载程序，相关操作请参见[章节 5 空中升级说明（OAD）](#)。

- peripheralAppROM: 从机应用程序
- peripheralAppROM_OAD_ImgA: 从机（片内小容量 OAD imageA）程序
- peripheralAppROM_OAD_ImgB: 从机（片内小容量 OAD imageB）程序
- peripheralAppROM_OAD_Large_ImgA: 从机（片内大容量 OAD imageA）程序
- peripheralAppROM_OAD_Large_ImgB: 从机（片内大容量 OAD imageB）程序
- centralAppROM: 主机应用程序



3. TTC SDK API 说明

3.1. TTC SDK 头文件说明

TTC SDK 共有 5 个头文件

- TTCBleSDKConfig.h:TTC SDK 工程配置头文件
- TTCBleDevInfoService.h:TTC SDK 设备信息服务头文件
- TTCBlePeripheral.h:TTC SDK 从机线程头文件
- TTCBlePeripheralProcess.h:TTC SDK 从机处理头文件
- TTCBleProfile.h:TTC SDK 处理蓝牙数据头文件

注意:

在函数说明上写了“请勿调用该函数，该函数为 SDK 功能。”。请用户不要自行调用或删除该函数。

3.2. 蓝牙初始化 API (TTCBlePeripheralProcess.h)

通过以下结构体可以直接设置蓝牙的参数。并将该结构体传递到函数

TTCBlePeripheralInit (TTCBlePeripheralInitConf_t * config) 中。

蓝牙从机参数初始化结构体

```
typedef struct { //蓝牙从机参数初始化结构体
TTCData_t      advData;           //广播数据
TTCData_t      scanRspData;       //扫描回应数据
TTCData_t      attDevName;        //设置通用设备特性名称
u16            maxConnInterval;   //最大连接间隔
u16            minConnInterval;   //最小连接间隔
u16            advOffTime;        //通过设置为零,设备将进入等待状态被发现后 30.72 秒
                                   //30.72 秒,又不会被广告直到设置为 TRUE
u8             advEnable;         //蓝牙广播使能
u8             updateParEnable;   //参数更新使能
u16            updateParDelay;    //参数更新延迟(默认 6)
u16            slaveLatency;      //从机跳过的回应包包数
u16            connTimeout;       //连接超时
u8             txPower;           //TX
u16            advInterval;       //广播间隔
u16            rssiReadPeriod;    //RSSI 刷新周期
u8             encryptEnable;     //加密使能
u8             advNoticeEnable;   //广播回调使能
u8             OverLappFlag;      //快速传输使能位
TTCPeripheralCBFxn_t  CB;
TTCSdkClass_t   *appCB;          //SDK 通用回调类//回调函数
uint8           taskId;          //任务 ID
}TTCBlePeripheralInitConf_t;
```

```

/*****
【函 数】 TTCblePeripheralInit(TTCblePeripheralInitConf_t * config)
【概 述】 蓝牙从机初始化
【入口参数】 config:蓝牙参数配置初始化结构体
【返回参数】 无
【说 明】 无
*****/

```

```

void TTCblePeripheralInit(TTCblePeripheralInitConf_t * config);

```

3.3. 蓝牙参数及相关蓝牙操作 API (TTCblePeripheral.h)

```

/*****
【函 数】 TTCblePeripheralSetParameter(u16 param, uint8_t len, void *pValue)
【概 述】 从机参数设置
【入口参数】 param:参数, 参数内容请参考 GAPROLE_PROFILE_PARAMETERS
               len:参数长度
               pValue:参数内容
【返回参数】 无
【说 明】 无
*****/

```

```

bStatus_t TTCblePeripheralSetParameter(u16 param, uint8_t len, void *pValue);

```

```

/*****
【函 数】 TTCblePeripheralGetParameter(u16 param, void *pValue)
【概 述】 从机获取参数
【入口参数】 param:参数, 参数内容请参考 GAPROLE_PROFILE_PARAMETERS 参数说明
               *pValue:参数内容
【返回参数】 无
【说 明】 无
*****/

```

```

bStatus_t TTCblePeripheralGetParameter(u16 param, void *pValue);

```

```

/*****
【函 数】 TTCblePeripheralGAPRoleTerminateConnection(void)
【概 述】 断开蓝牙连接
【入口参数】 无
【返回参数】 无
【说 明】 无
*****/

```

```

bStatus_t TTCblePeripheralGAPRoleTerminateConnection(void);

```

```

/*****
【函 数】 TTCblePeripheralGAPRoleTerminateConnection(void)
【概 述】 断开蓝牙连接
*****/

```


【入口参数】 无
【返回参数】 无
【说 明】 无

```

*****/
bStatus_t TTCBlePeripheralGAPRoleTerminateConnection(void);

```

```

/*****

```

【函 数】 TTCBlePeripheralGAPRoleSendUpdateParam(u16 minConnInterval,
u16 maxConnInterval,
u16 latency,
u16 connTimeout,
u8 handleFailure);

【概 述】 发送蓝牙参数更新请求

【入口参数】 minConnInterval:最小连接间隔
maxConnInterval:最大连接间隔
latency: 蓝牙从机延迟（允许跳过回应包包数）
connTimeout:连接超时
handleFailure:更新失败操作。请参考 GAPROLE_FAILED_UPDATE 参数说明

【返回参数】 无
【说 明】 无

```

*****/

```

```

bStatus_t TTCBlePeripheralGAPRoleSendUpdateParam(u16 minConnInterval,
u16 maxConnInterval,
u16 latency,
u16 connTimeout,
u8 handleFailure);

```

3.4. 蓝牙数据发送相关 API (TTCBleProfile.h)

```

/*****

```

【函 数】 TTCBleProfileSetParameter(u8 param,u16 len, void *value)

【概 述】 设置蓝牙通道数据

【入口参数】 param:设置设备信息服务参数，填入的参数如下

TTCBLE_PROFILE_CHAR1（暂不开放）
TTCBLE_PROFILE_CHAR2
TTCBLE_PROFILE_CHAR3（暂不开放）
TTCBLE_PROFILE_CHAR4（暂不开放）
TTCBLE_PROFILE_CHAR5（暂不开放）

len:发送数据长度

value:数据

【返回参数】 无

【说明】 无

```

*****/
extern bStatus_t TTCBleProfileSetParameter( u8 param,u16 len, u8 * value );

```

3.5. SDK 线程消息说明

在 TTCBleSDKConfig.h 头文件中定义了如下的消息事件（请勿修改）

```

#define TTCSDK_MSG_GET_BLE_STATE_EVENT  0x0001    //蓝牙状态事件
#define TTCSDK_MSG_GET_BLE_DATA_EVENT   0x0002    //蓝牙数据事件
#define TTCSDK_MSG_REFRESH_RSSI_EVENT   0x0003    //刷新 RSSI 值
#define TTCSDK_MSG_GET_BLE_PARAM_EVENT  0x0004    //获取协商后的蓝牙参数

```

在 TTCBlePeripheralTask.c 中对消息进行接收处理。

```

/*****
【函数】 TTCBlePeripheralTaskProcessAppMsg(TTCMsg_t *pMsg)
【概述】 线程消息处理函数
【入口参数】 pMsg:消息数据
【返回参数】 无
【说明】 请注意内存的释放
*****/
static void TTCBlePeripheralTaskProcessAppMsg(TTCMsg_t *pMsg);

```

3.5.1. TTCSDK_MSG_GET_BLE_STATE_EVENT 蓝牙状态处理

该事件会对蓝牙的状态进行处理，开发人员如需根据蓝牙状态进行应用，可在此 TTCSDK_MSG_GET_BLE_STATE_EVENT 事件中进行处理。

注意：请不要删除 `TTCBlePeripheralProcessStateChangeEvt((gaprole_States_t)pMsg->hdr.state);`;

3.5.2. TTCSDK_MSG_GET_BLE_DATA_EVENT 蓝牙接收数据处理

该事件用于处理蓝牙通道 1 接收到的数据（其他通道暂时不开放），开发人员可在此处理接收到的蓝牙数据。

在 `TTCBlePeripheralTaskGetBleData(TTCMsg_t *TTCMsg)` 中，`TTCMsg->hdr.state` 携带的内容表示蓝牙通道。`TTCData->param` 表示蓝牙数据是否正常。在开启加密的情况下，若加密数据有误，将不会对数据进行解密，并且 `TTCData->param` 携带的参数为 `TTCSDK_ERR_ENCRYPT_DATA`，表示数据错误。其他情况下 `TTCData->param` 携带的参数为 `TTCSDK_NOERR_DATA`。消息传递蓝牙数据结构体为

```

typedef struct{
    u8      len      ;//长度
    u8      param    ;//参数
    u8 *    pValue   ;//数据内容
}TTCData_t;

```

3.5.3. TTCSDK_MSG_REFRESH_RSSI_EVENT 连接状态下 RSSI 值读取

该事件用于处理设备与主机连接后的获取到的 RSSI 值。开发人员可在 `TTCBlePeripheralSetParameter` 函数中对 RSSI 的刷新周期进行设置，设置 RSSI 刷新周期参数为 `GAPROLE_RSSI_READ_RATE`。当设置值为 0 时，将不会刷新 RSSI 值。当设置值不为 0 时，将会以设置值为周期，定期的更新 RSSI 值。消息传递的内容为 `s8(signed char)` 类型数据。

注意：该 RSSI 值只会在连接状态下刷新。

3.5.4. TTCSDK_MSG_GET_BLE_PARAM_EVENT 协商后的蓝牙参数

该事件用于告知开发人员主从机蓝牙参数协商后的蓝牙参数。开发人员可在该事件中获取到最终协商的蓝牙参数。消息传递蓝牙参数的结构体为

```
typedef struct{
    u16    connInterval           ;//连接间隔
    u16    connSlaveLatency       ;//从机跳过回应包包数
    u16    connTimeout           ;//连接超时
}TTCBleParamUpdate_t;
```

3.6. 设备信息服务参数设置 (TTCBleDevInfoService.h)

【函数】 `TTCBleDevInfoSetParameter(u8 param, u8 len, void *value)`

【概述】 设置设备信息服务参数

【入口参数】 `param`: 设置设备信息服务参数，填入的参数如下

```
DEVINFO_SYSTEM_ID
DEVINFO_SERIAL_NUMBER
DEVINFO_FIRMWARE_REV
DEVINFO_HARDWARE_REV
DEVINFO_SOFTWARE_REV
DEVINFO_MANUFACTURER_NAME
DEVINFO_11073_CERT_DATA
DEVINFO_PNP_ID
```

`len`: 数据长度，对应参数的数据长度如下

<code>DEVINFO_SYSTEM_ID_LEN</code>	8 (固定值，长度必须等于 8 否则无效)
<code>DEVINFO_SERIAL_NUMBER_LEN</code>	21
<code>DEVINFO_FIRMWARE_REV_LEN</code>	21
<code>DEVINFO_HARDWARE_REV_LEN</code>	21
<code>DEVINFO_SOFTWARE_REV_LEN</code>	21
<code>DEVINFO_MANUFACTURER_NAME_LEN</code>	21
<code>DEVINFO_11073_CERT_DATA</code>	自定义
<code>DEVINFO_PNP_ID_LEN</code>	7 (固定值，长度必须等于 7 否则无效)

`value`: 设置数据

【返回参数】 无

【说明】 无

```

*****/
extern bStatus_t TTCBleDevInfoSetParameter( u8 param, u8 len, void *value );

```

```

*****/

```

【函数】 TTCBleDevInfoGetParameter(u8 param, void *value)

【概述】 获取设备信息服务参数

【入口参数】 param:设置参数，填入的参数如下

```

DEVINFO_SYSTEM_ID
DEVINFO_SERIAL_NUMBER
DEVINFO_FIRMWARE_REV
DEVINFO_HARDWARE_REV
DEVINFO_SOFTWARE_REV
DEVINFO_MANUFACTURER_NAME
DEVINFO_11073_CERT_DATA
DEVINFO_PNP_ID

```

value:读取数据

【返回参数】 无

【说明】 无

```

*****/
extern bStatus_t TTCBleDevInfoGetParameter( u8 param, void *value );

```

3.7. 应用线程公共回调函数说明

在 TTCBleSDKConfig.h 中声明了公共回调函数（请勿修改）

```

*****/

```

【函数】 TTCBlePeripheralTaskEnqueueMsg(u16 event,
u16 state,
void * pValue)

【概述】 线程置事件

【入口参数】 event:消息携带的事件

state:消息携带的状态

pValue:消息携带的数据

【返回参数】 无

【说明】 该函数封装到 TTCBlePeripheralTaskClass_t 中，为公用函数请勿修改

```

*****/

```

```

Typedef u8 (*TTCBlePeripheralTaskEnqueueMsg_t)(u16 event,
u16 state,
void *pValue);

```

//TTC SDK 从机线程

```

typedef struct {
    TTCBlePeripheralTaskEnqueueMsg_t pfnTTCBlePeripheralTaskEnqueueMsg;
}TTCBlePeripheralTaskClass_t;

```

//TTC SDK 公共函数回调声明

```
extern TTCblePeripheralTaskClass_t *TTCbleRegisterCls;
```

以上回调函数原型在 TTCblePeripheralTask.c 中，开发人员可用回调发送消息。

公开资料

4. 蓝牙角色 Demo 说明

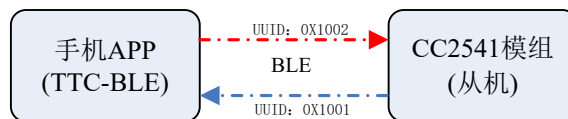
TTC 2541 SDK 支持两种蓝牙角色：主机、从机。其中相关 demo 是通过 AT 指令进行演示的。

4.1. 从机示例

4.1.1 示例功能说明

1. 手机 APP (TTC-BLE) 与 CC2541 模块建立连接后，手机 APP 端通过 BLE 对应的数据下行通道 (UUID: 0X1002) 发送数据给 CC2541 模块，数据传输方向，如下图蓝线所示；

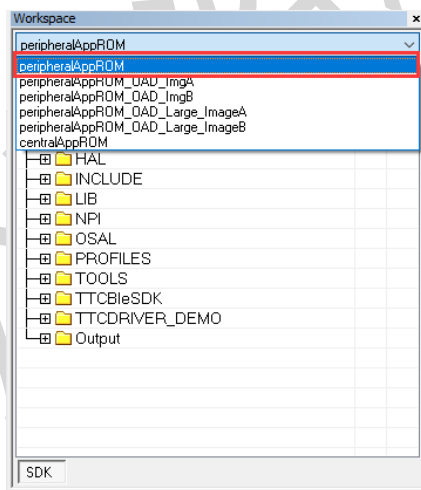
CC2541 模块接收到数据后，通过 BLE 对应的数据上行通道 (UUID: 0X1001) 转发给手机 APP 端；数据流方向，如下图红线所示。



2. 上电后，IO1_1 每 500ms 翻转一次。

4.1.2. 程序编译下载

在配置下拉框中选择“peripheralAppROM”，重新编译并下载程序，如下图：



4.1.3. 相关代码实现

1. 从机收到 APP 下发的数据后，将数据回传给 APP

(1) 从机接收到蓝牙数据、或者蓝牙状态发生改变，底层会发送消息事件，应用程序处理相应的消息事件，如下：

```

static void TTCBlePeripheralTaskProcessAppMsg(osal_event_hdr_t *pMsg)
{
    TTCMsg_t *pTTCMsg = (TTCMsg_t *)pMsg;
    switch ( pMsg->event )
    {
        .....
    }
}
  
```

```

case TTCSDK_MSG_GET_BLE_STATE_EVENT: { //蓝牙状态消息处理
    .....
    TTCBlePeripheralProcessStateChangeEvt((gaprole_States_t)pTTCMsg->hdr.status);
}break;

case TTCSDK_MSG_GET_BLE_DATA_EVENT: { //蓝牙通道数据消息处理
    TTCBlePeripheralTaskGetBleData(pTTCMsg);
}break;
.....
}
}
    
```

收到数据后，将数据回传：

```

static void TTCBlePeripheralTaskGetBleData(TTCMsg_t *pTTCMsg)
{
    TTCData_t *pData = pTTCMsg->data;
    if(pData->param == TTCSDK_NOERR_DATA){
        switch(pTTCMsg->hdr.status) {

            case TTCBLE_PROFILE_CHAR1: //通道 1 AppTx
                //把通道 1 接收的数据经过通道 2 发送回给 App(此地方只做 demo 演示,可删除)
                TTCBleProfileSetParameter(TTCBLE_PROFILE_CHAR2, pData->len, pData->pValue);
            break;

            default: break; //其他
        }
    }
}
    
```

4.1.4 IO 功能演示

上电 P1_1 每隔 500ms 翻转

定义一个定时事件，每次达到定时时间，则置起事件，OSAL 轮询到该事件时则可以进行相关处理。处理完毕后，再次开启定时器。

(1) 启动定时器

```

void TTCBlePeripheralTask_Init(u8 task_id) {
    .....
    osal_start_timerEx(simpleBLEPeripheral_TaskID, LED_DIS_DEMO_EVENT, 500); //演示 demo(可删除)
    .....
}
    
```

(2) 定时器事件处理

```

u16 TTCBlePeripheralTask_ProcessEvent(u8 task_id, u16 events) {
    .....
    if(events & LED_DIS_DEMO_EVENT) { //演示事件 LED 取反处理
        P1_1 = ~P1_1;
        osal_start_timerEx(simpleBLEPeripheral_TaskID, LED_DIS_DEMO_EVENT, 500); //再次启动
        return (events ^ LED_DIS_DEMO_EVENT);
    }
    .....
}
  
```

4.1.5 快速传输功能演示

快速传输功能演示

1. 蓝牙参数初始化时使能 `OverLappFlag`，并把 `OVERLAPPED_DEMO` 宏打开，

```

if(events & OVERLAPPED_DEMO_EVENT) { //演示事件 LED 取反处理
#ifdef OVERLAPPED_DEMO
    static uint16 sendPackPeriod=40;
    static uint32 sendcounttest=0;
    if(TTCBleGapState == GAPROLE_CONNECTED)
    {
        uint8
data_temp[20]={0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x
08,0x09,0x09};

        TTCBlePeripheralGetParameter(GAPROLE_CONN_INTERVAL,&sendPackPeriod);
        sendPackPeriod = sendPackPeriod*5/4;
        if(sendPackPeriod <20 )
        {
            sendPackPeriod = 20;
        }
        for(uint8 j=0; j<4; j++)
        {
            TTCBleProfileSetParameter(TTCBLE_PROFILE_CHAR2, 20, data_temp);
            sendcounttest++;
        }
    }
    osal_start_timerEx(simpleBLEPeripheral_TaskID, OVERLAPPED_DEMO_EVENT, sendPackPeriod);
#endif
    return (events ^ OVERLAPPED_DEMO_EVENT);
}
  
```

从以上 DEMO 可以看到，使能快速发包后，一个连接间隔里面可以实现传输 4 个数据包。

4.1.6 微信功能演示

微信驱动包含了微信的 AirSync 的蓝牙通信协议；自动处理被微信发现、与微信握手连接、数据的蓝牙分包等，用户只需要进行相关的初始化，并调用相应的 API 进行收、发数据即可

1. demo 设置及功能

(1) demo 设置

在从机角色，定义宏 TTCBLE_WECHAT，即可开启 WeChat 相关功能。

(2) 注意事项

以下示例中，微信公众号“深圳市昇润科技有限公司”进入相关界面可以与模组建立链接并进行数据通信。需要注意，SDK 中微信功能所使用的 Mac 地址，是需要经过授权才能正常使用的。示例中仅使用同一个已授权的 Mac 地址及二维码作为测试使用，实际产品生产需使用不同的 Mac 地址，相关授权事宜请与我司联系。

2. 模块与微信公众号绑定连接

(1) 开启微信，扫一扫以下二维码



(2) 如下为弹出界面，点击“下一步”



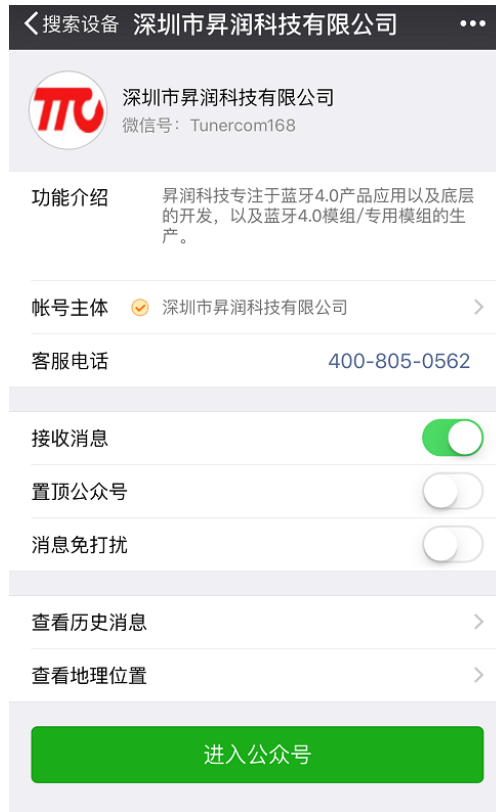
(3) 进入扫描界面，并扫描到设备，如下图，点击设备



(4) 进入公众号，点击绑定设备



(5) 点击进入公众号



(6) 等待与设备建立连接，显示“已连接”；点击“智能硬件”“透传”：

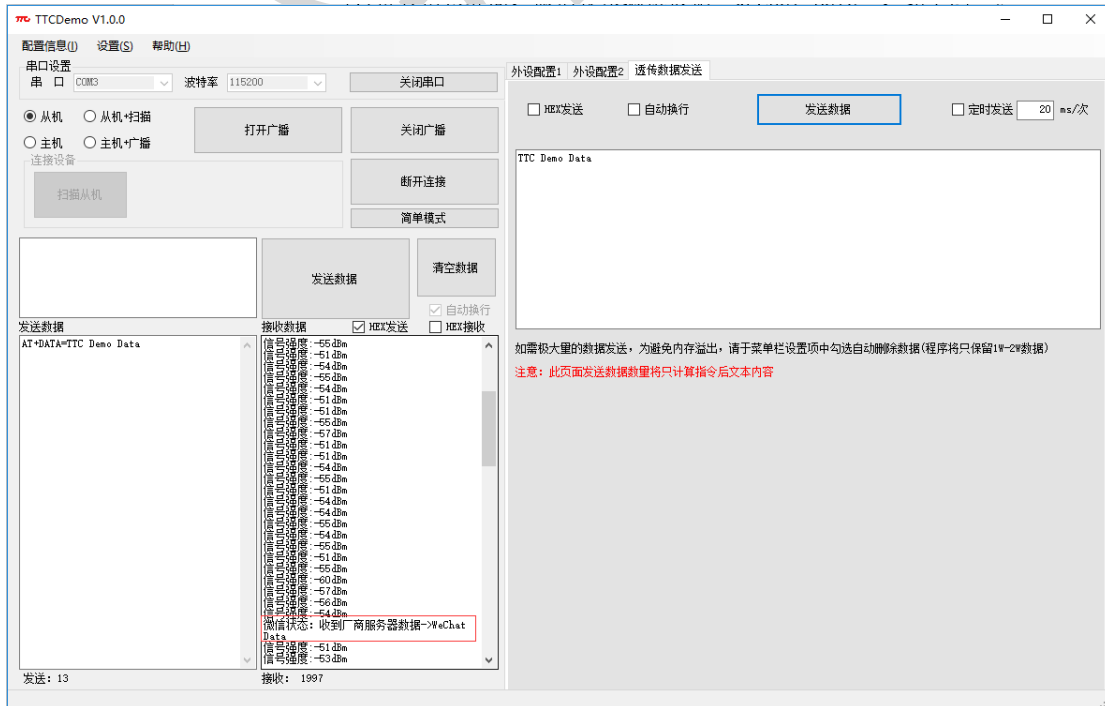


3. 数据收发

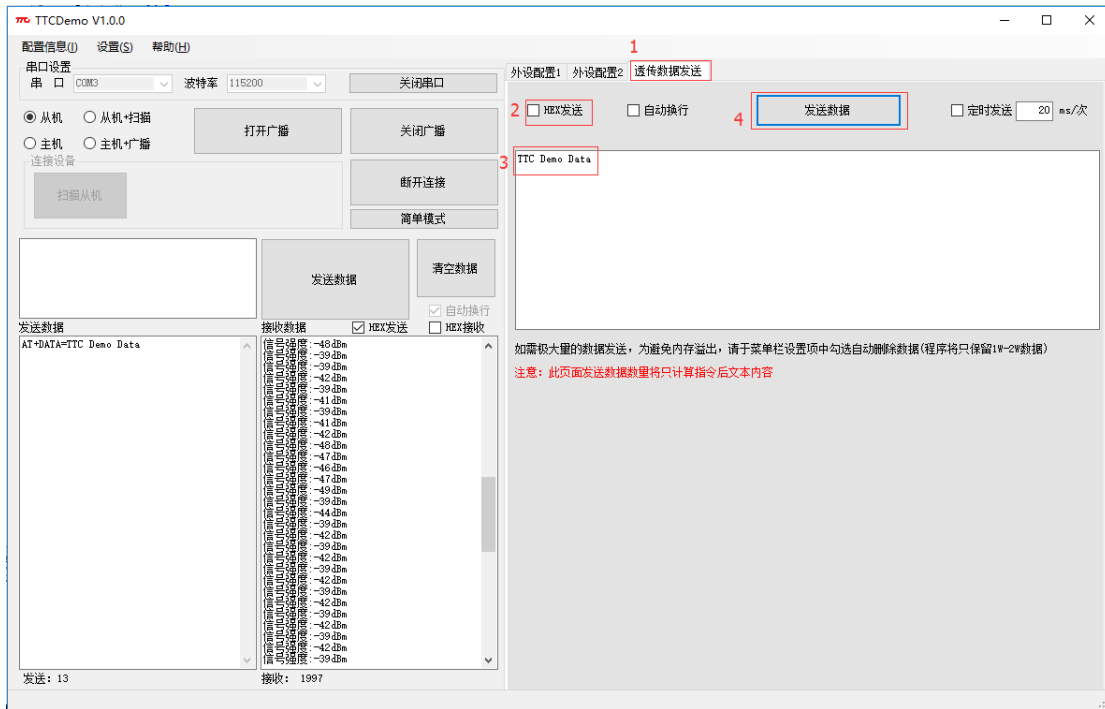
(1) 进入透传界面，微信发送测试数据 “WeChat Data”，如下图：



CC2541 模组收到微信数据后，发送到 PC 端软件 (TTC Demo) 显示如下：



(2) PC 端软件(TTCDemo)发送数据“TTC Demo Data”：



CC2541 模组将数据发送给微信，接收框显示接收到的数据，如下图：



4. API 说明

WeChat 相关 API 见 TTCBleWechat.h

4.2. 主机示例

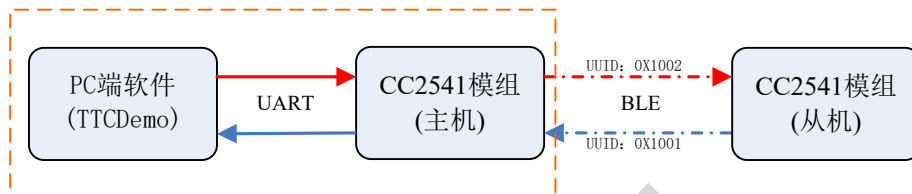
主机 demo 使用 AT 指令的方式，展示相关的功能。默认已经开启宏定义：TTC_DEBUG、TTCDRIVER_UART。

需注意，主机暂不支持测试程序。

4.2.1. 示例功能说明

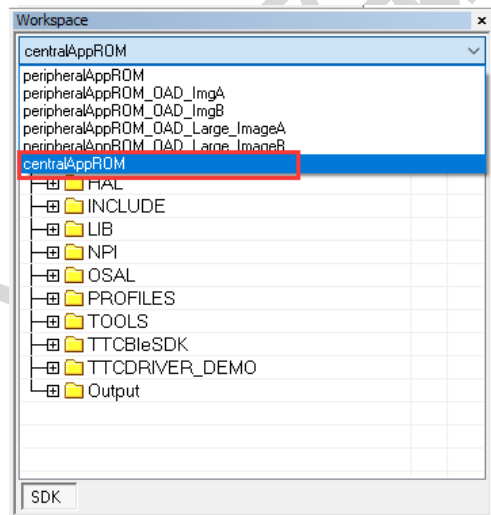
主机扫描从机，建立链接后，PC 端软件(TTCDemo)通过 UART(115200bps)发送 AT 指令实现主机的控制，如主机向从机发送数据，同时接收从机回传的数据。

左侧橘色框中 PC 端软件+2541 主机，实现了扫描从机、向从机发起/断开连接、蓝牙数据收发，功能类似于手机 APP（TTC-BLE）。



4.2.2. 程序编译下载

在配置下拉框中选择“centralAppROM”，重新全编译并下载程序，如下图：



4.2.3. 相关代码实现

1. 串口参数设置并初始化

```

void TTCDebugInit(u8 taskID, TTCSdkClass_t *pCB) {
    .....

    halUARTCfg_t uartcofing;
    uartcofing.configured = TRUE;
    uartcofing.baudRate = HAL_UART_BR_115200;
    uartcofing.flowControl = FALSE;
    uartcofing.callBackFunc = AT_UartRxCallBack;

    state = HalUARTOpen( HAL_UART_PORT_0, &uartcofing );
    .....
}
  
```

2. 串口接收并解析 AT 指令

```
void AT_UartRxCallBack(uint8 port, uint8 event) {
    .....
    TTCDebugReadCB(RxTempBuf, RxTempBuf_len);    //指令解析正确
    .....
}
```

指令解析正确，发送消息，便于应用程序处理：

```
static void TTCDebugReadCB(u8 * buffer, u8 len) {
    .....
    osal_msg_send(Msg_Uart_TTCBleCentralTaskId, (u8 *)pMsg);
    .....
}
```

3. AT 指令相应处理

消息处理：

```
static void TTCBleCentralTaskProcessAppMsg(osal_event_hdr_t *pMsg) {
    TTCMsg_t *pTTCMsg = (TTCMsg_t *)pMsg;
    switch (pMsg->event) {
        .....
        .....
        #ifdef TTC_DEBUG
        case TTCSDK_MSG_DRIVER_UART_EVENT: {
            TTCDriverUartProcess(pTTCMsg);
        }break;
        #endif
    }
    .....
}
```

AT 指令处理：

```
static void TTCDriverUartProcess(TTCMsg_t * TTCMsg) {
    .....
    if(!memcmp(buf, CMD_Scan, SLAVE_SCAN_AT_LEN)) {                //开始扫描
        .....
    }else if(!memcmp(buf, CMD_Connect, CONNECT_AT_LEN)) {        //连接
        .....
    }else if(!memcmp(buf, CMD_Disconnect, DISCONNECT_AT_LEN)) {  //断开连接
        .....
    }else if(!memcmp(buf, CMD_SendData, BLE_SEND_DATA_AT_LEN)) { //发送数据
        .....
    }
}
```

4. 主机接收蓝牙数据

主机接收到蓝牙数据，相应的消息事件处理：

```
static void TTCBleCentralTaskProcessAppMsg(osal_event_hdr_t *pMsg) {
    TTCMsg_t *pTTCMsg = (TTCMsg_t *)pMsg;
```

```
switch (pMsg->event) {
    .....

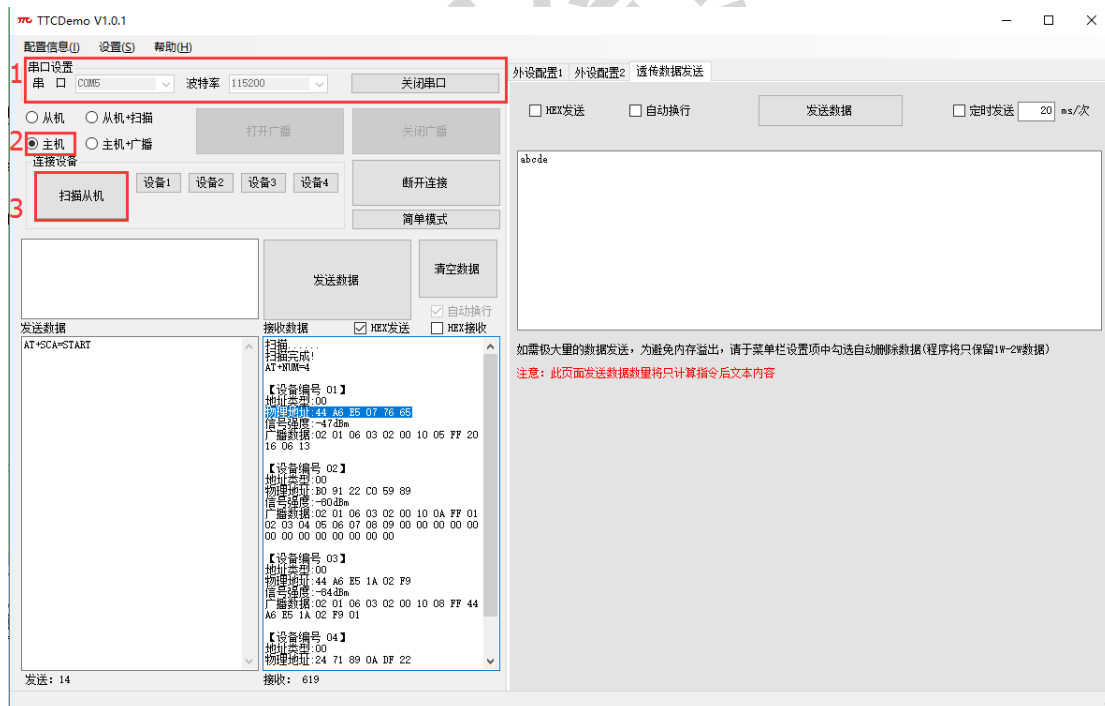
    case TTCSDK_MSG_GET_BLE_DATA_EVENT: {
        TTCBleCentralTaskGetBleData (pTTCMsg);
    }break;
    .....
}
```

将蓝牙数据通过 UART 传输至 PC 端软件（TTCDemo）：

```
static void TTCBleCentralTaskGetBleData(TTCMsg_t * TTCMsg) {
    TTCData_t * bleData = TTCMsg->data;
    TTCDebugLogPrint (bleData->pValue, bleData->len);
}
```

4.2.4. 功能演示

1. 准备两个 2541 模组，分别烧录主机、从机程序；
2. 主机模组通过 USB 转接板，或者 USB 转 UART 的小工具连接至 PC 机；
3. 从机供电即可，即连接 VDD/GND，如连接下载器则正常供电；
4. 开启 PC 端软件(TTCDemo)，选择主机模组对应的串口端口，设置波特率 115200bps, 打开串口；选择“主机”，再点击“扫描从机”，扫描结束后返回扫描结果。如目标从机地址为 0X44A6E5077665，对应设备 1.



5. 点击“设备 1”，则向目标从机发起链接。连接成功后，会有蓝牙参数更新提示，此时可以进行蓝牙数据收发：主机发送数据给从机，从机再将数据回传给主机，如下图：



6. 点击“断开连接”，即可断开与从机的蓝牙连接



5. 空中升级说明 (OAD)

OAD, Over-the-Air Download, 即空中升级。

TTC 2541 SDK 支持两种片内升级方式：小容量 OAD（可编程 flash 空间为 128KByte）、大容量 OAD（可编程 flash 空间为 157KByte）。

(1) 当前支持片内升级的角色：从机

(2) 带 OAD 功能的配置，不支持在线调试仿真，不可直接使用 IAR 下载程序。为方便在线仿真调试，可先使用不带 OAD 的工程进行项目开发，功能完善后，再切换至对应的 OAD 配置，即完成 OAD 功能的添加。

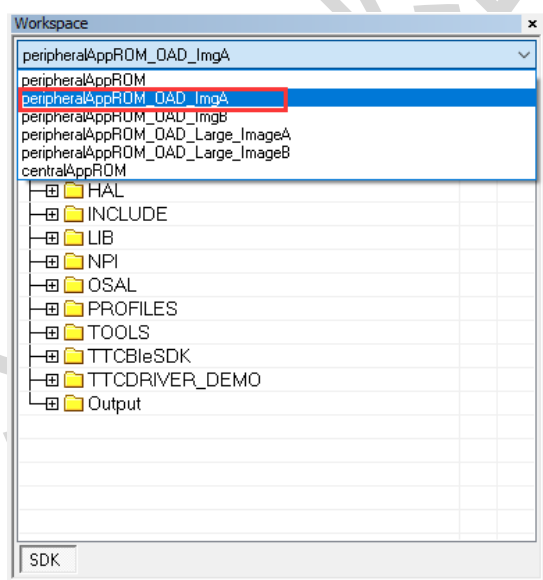
5.1. 小容量 OAD

5.1.1. 首先下载 BIM 工程

路径：Projects\ble\util\BIM\cc254x

5.1.2. image A 生成

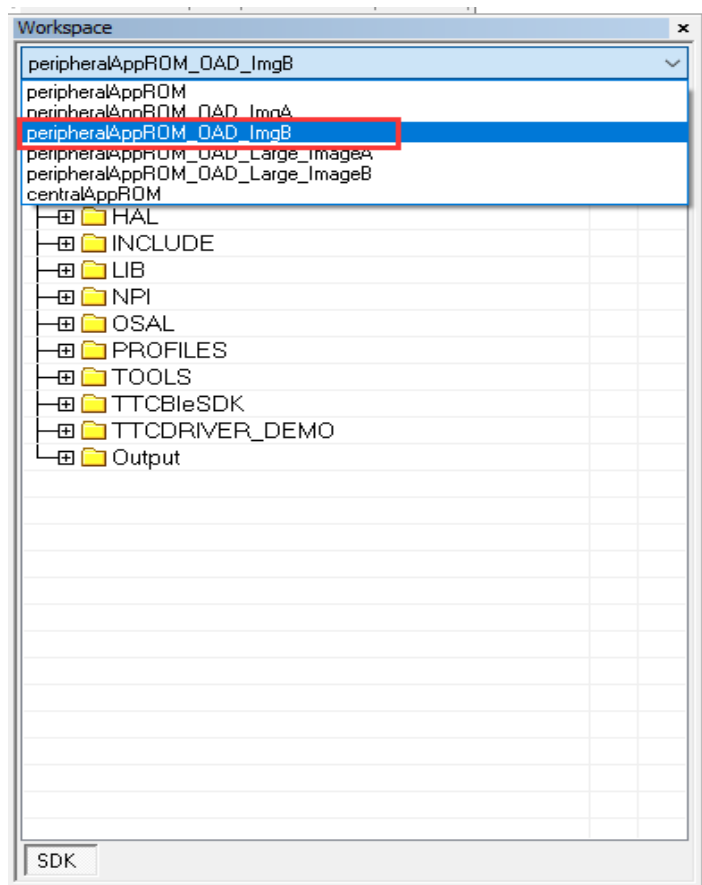
选择 peripheralAppRom_OAD_ImgA 配置，编译并下载程序。



5.1.3. image B 生成

选择 peripheralAppRom_OAD_ImgB 配置，编译程序。

路径：\Projects\ble\TTCBleSDK_2541\CC2541DB\imageFile\Exe 中 TTC_SDK_20170825_V1.6_ImageB.bin 即为 APP OAD 升级所需 bin 文件。



5.1.4. 手机 APP (TTC-BLE) 操作说明

(1) 将上小节生成的 bin 文件 TTC_SDK_20170825_V1.6_ImageA.bin 导入手机根目录 Download 目录下；注意：当程序为 ImageA 时，升级文件必须使用 ImageB，当程序为 ImageB 时，升级文件必须使用 ImageA。

(2) 扫描二维码，下载安装 TTC_BLE，并开启手机蓝牙；

(3) 开启 APP，点击 OAD，进入 OAD 模式，如下图 1；

(4) 点击对应设备，与设备建立链接，如下图 2；

(5) 选择 2541 小容量升级，如图 3。

(6) 点击“+”，导入 bin 文件，File Image Type 为 B，如下图 4；

(7) 设置传输间隔为 20ms，点击开始，如下图 10；

(8) 传输至 100%后，不需要操作 APP。设备复位，运行新程序，与 APP 断开链接，如下图 5；

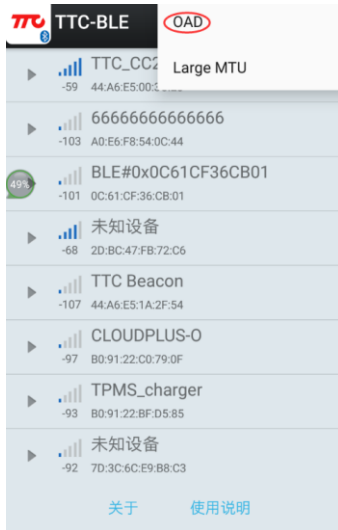


图 1 进入 OAD 界面

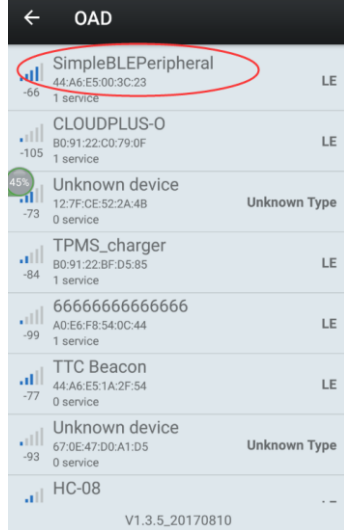


图 2 连接设备

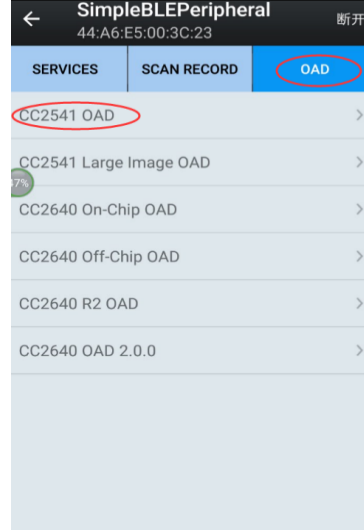


图 3 选择 OAD 方式

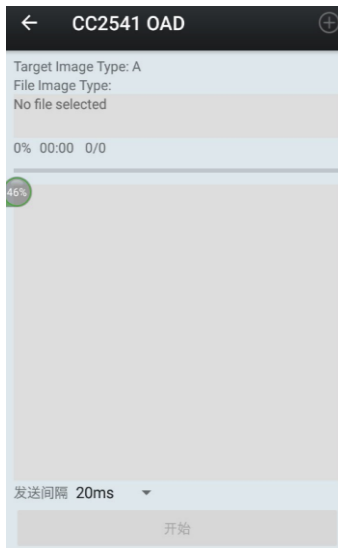


图 4 加载 OAD 文件



图 5 升级完成

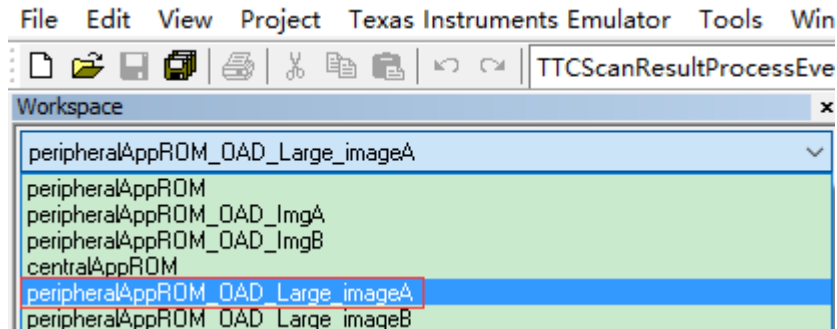
5.2. 大容量 OAD

5.2.1. 首先下载 LARGE BIM 工程

路径: Projects\ble\util_Large_OAD\BIM\cc254x

5.2.2. 编译并下载 Large image A

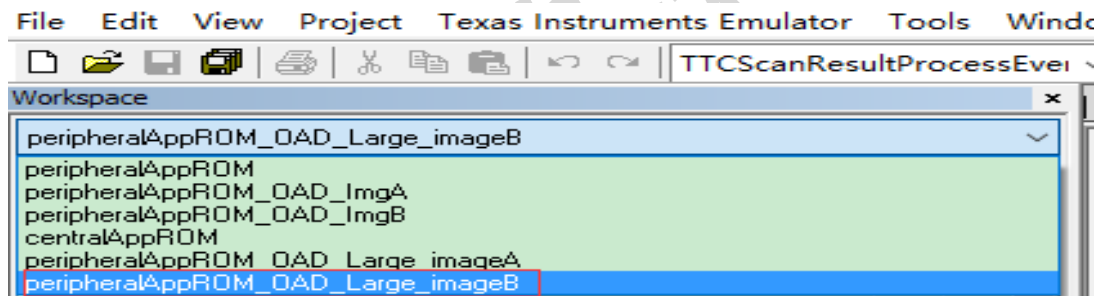
先 clear, 然后再 rebuild all, 最后下载程序。



5.2.3. Large image B 生成

选择 peripheralAppRom_OAD_Large_imageB 配置, 开了哪些功能, 相对应的宏在此工程中也要记得打开。

路径: Projects\ble\TTCBleSDK_2541\CC2541DB\imageFile\Exe



5.2.4. 手机 APP (TTC-BLE) 操作说明

(1) 将上小节生成的 bin 文件 TTC_SDK_20170825_V1.6_Large_imageB.bin 导入手机根目录 Download 目录下; 注意: 先下载 LARGE BIM, 再下载 Large ImageA, 此时的程序不具有用户程序的功能, 仅用于 OAD 升级。

(2) 扫描二维码, 下载安装 TTC_BLE, 并开启手机蓝牙;

(3) 开启 APP, 点击 OAD, 进入 APP 端的 OAD 模式, 如下图 1;

(4) 连接前先确认模块程序在 OAD 模式, 即运行在 Large ImageA 程序, 如果程序是在正常模式, 则需要先通过 1003 通道发 55, AA 指令, 程序会复位进入 OAD 模式, 用户也可以通过其他触发条件让程序进入 OAD 模式。如图 2、3、4 的操作。

(5) 点击对应设备, 与设备建立链接, 如下图 5;

(6) 选择是 2541 大容量升级。如图 6。

(7) 点击“+”, 导入 bin 文件, File Image Type 为 B, 如下图 8、9;

(8) 设置传输间隔为 20ms, 点击开始, 如下图 9;

(9) 传输至 100%后, 不需要操作 APP。设备复位, 运行新程序, 与 APP 断开链接, 如下图 9;



图 1 正常模式

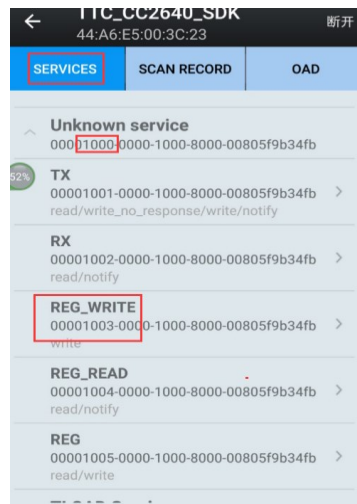


图 2 OAD 模式通道

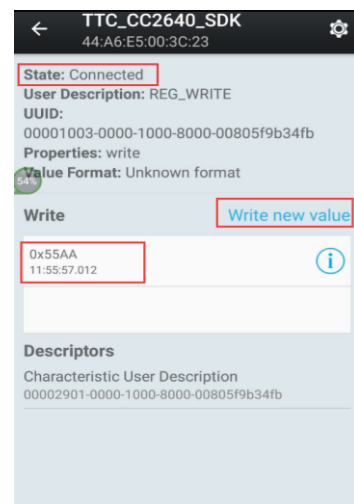


图 3 发指令进行切换

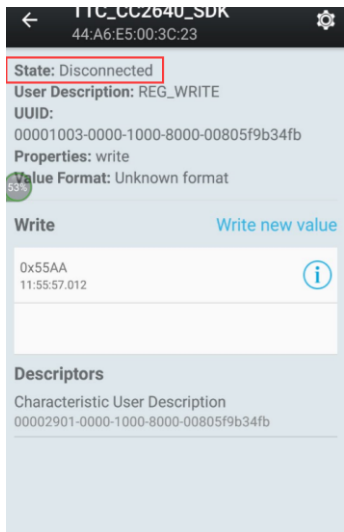


图 4 自动断开连接

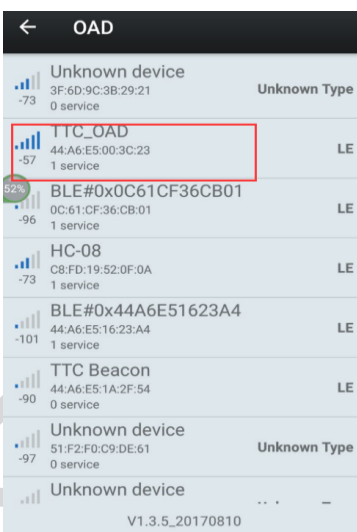


图 5 OAD 模式

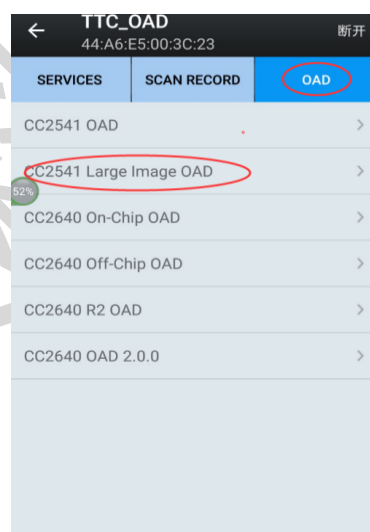


图 6 链接设备

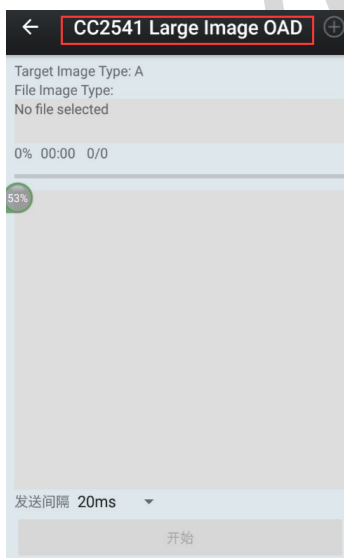


图 7 加载文件

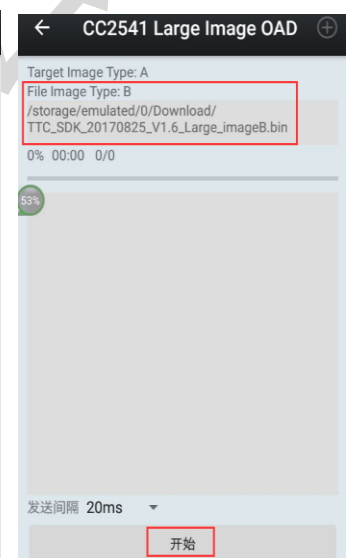


图 8 准备升级

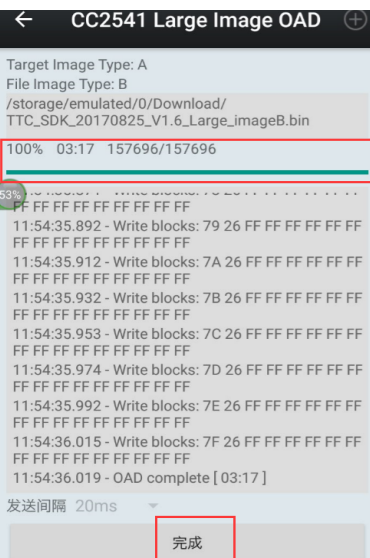


图 9 升级完成

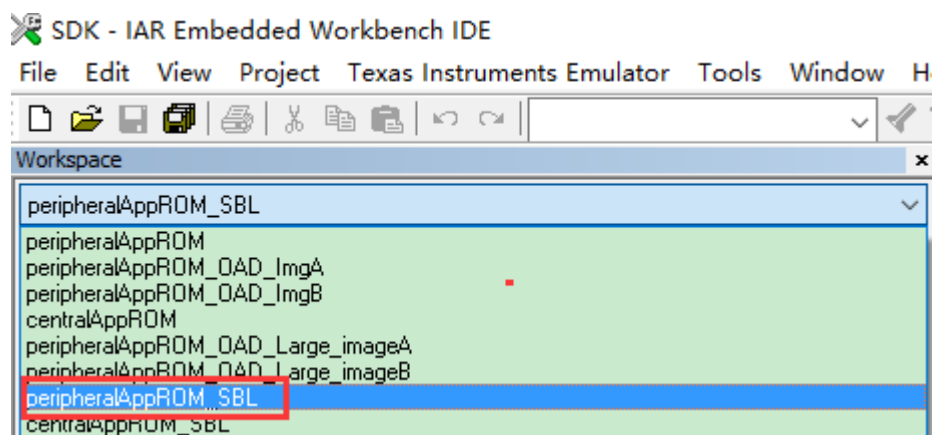
6. 串口升级（SBL）

6.1. 首先下载 SBL 工程

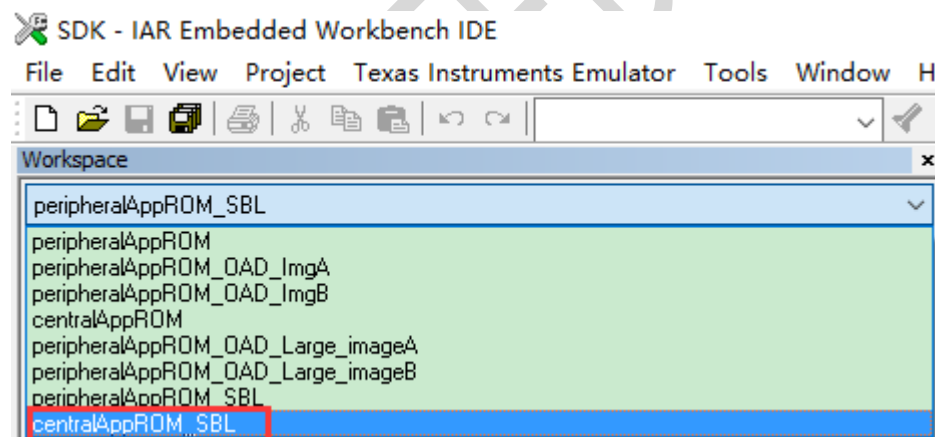
路径：Projects\ble\util\SBL\iar\cc254x

6.2. BIN 文件生成

如果是使用从机开发，选择 peripheralAppRom_SBL 配置，编译生成 BIN 文件。

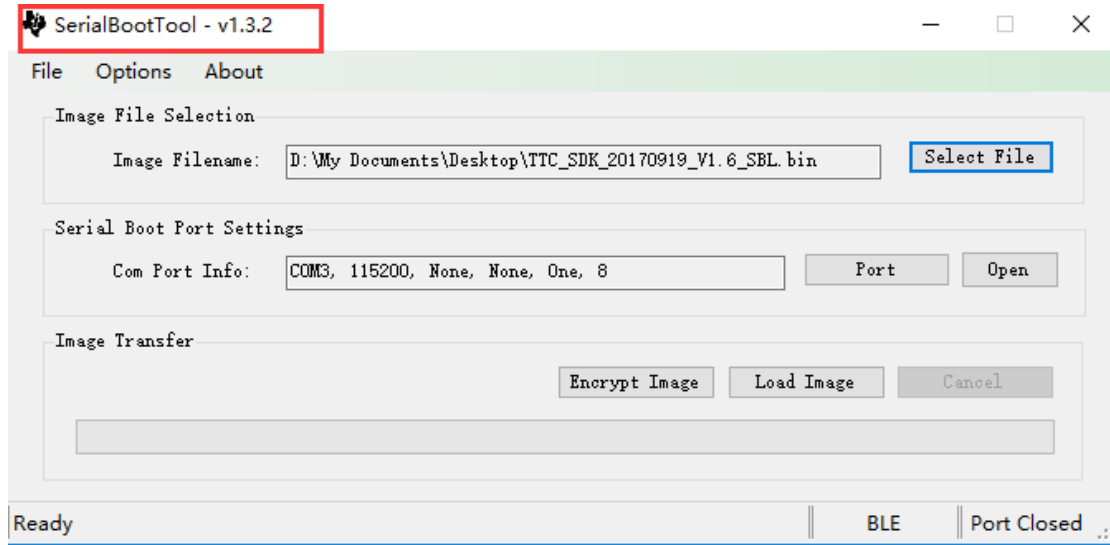


如果是使用主机开发，选择 centralAppRom_SBL 配置，编译生成 BIN 文件。



6.3. 进行串口升级

1. 使用 SerialBootTool 工具首先装载 BIN 文件，然后选择 Load Image 即可。如下图所示：



7. 联系我们

深圳市昇润科技有限公司

ShenZhen ShengRun Technology Co.,Ltd.

Tel: 0755-86233846 Fax: 0755-82970906

官网地址: www.tuner168.com

阿里巴巴网址: <http://shop1439435278127.1688.com>

E-mail: marketing@tuner168.com

地址: 广东省深圳市南山区西丽镇龙珠四路金谷创业园B栋6楼601-602



附录 A. 手机 APP 下载

1. 手机 APP (TTC-BLE) 下载二维码



公开资料