

通过 USB 控制蓝牙

基于 USBdongle 的 AT 命令示例

Ghostyu.com

2014/1/14

[在此处键入文档的摘要。摘要通常是对文档内容的简短总结。在此处键入文档的摘要。
摘要通常是对文档内容的简短总结。]

目录

1 前言.....	2
2 必要条件	2
3 文件预览	2
4 源码包解压	3
5 打开 IAR 工程.....	3
6 编译下载.....	6
6.1 编译	6
6.2 下载	7
6.3 驱动安装.....	8
7 测试.....	8

1 前言

在前面几个实践中，我们讲解了通过 CC2540 的 UART 通信，以及实现的简单的 AT 命令接口，单价大家有没有想过如何通过 CC2540 的 USB，直接与 PC 通信，而非使用 CC2540 的串口，今天，我们就带领大家实现基于 CC2540USBdongle 的 USB 通信实例。

该 demo 和 TI 提供的 HostTestApp 协议栈 demo 很类似，当 USBdongle 插到电脑上会被识别成一个虚拟串口，安装标准的 CDC 驱动后，就可以在 PC 上打开这个 USBdongle 虚拟出来的串口，然后与 dongle 通信。然后通过发送的简单地 AT 命令来控制 USBdongle。这个 demo 非常有实际意义。

2 必要条件

A 硬件

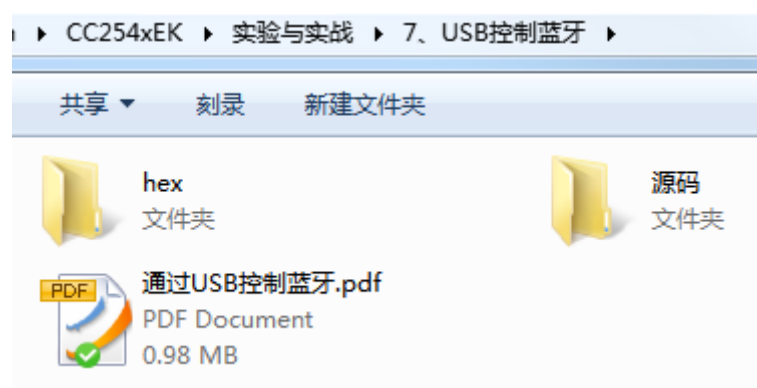
- 1、CC2540USBdongle
- 2、CC-Debugger 仿真器（以及转接板，需要连接 USBdongle 烧写程序）
- 3、运行 SimpleBLEPeripheral 从机程序的开发板。用来连接测试。

B 软件

- 1、ble 协议栈，版本：1.3.2
- 2、IAR for 8051 开发环境，版本：8.10
- 3、Flash Programmer 固件烧写软件。
- 4、串口调试助手。

3 文件预览

本文档的所有相关源码、说明均位于【CC254xEK\实验与实战\7、USB 控制蓝牙】目录下，如下图：



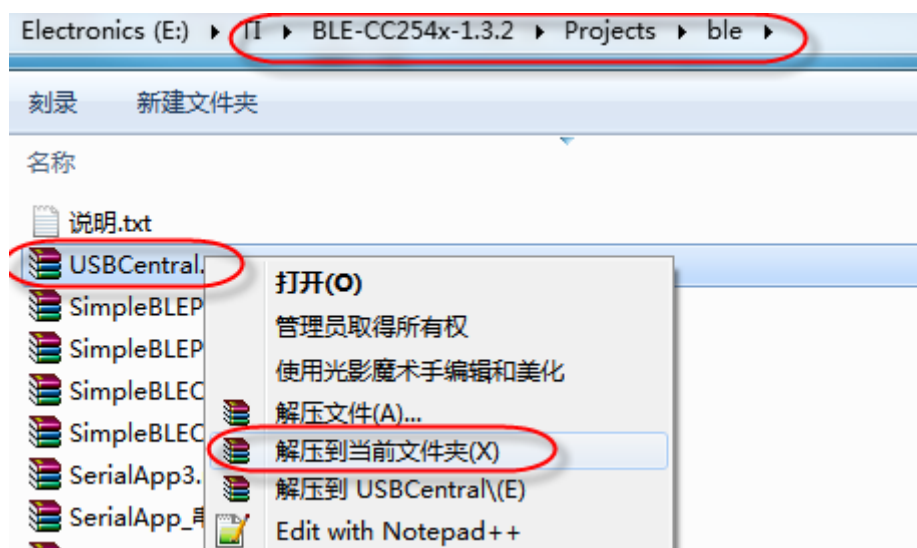
【Hex】文件夹存放我们预先编译 OK 的固件，可以直接下载到 SmartRF 系列开发板中测试运行。

【源码】文件夹存放的是该实践相关的源码程序

【通过 USB 控制蓝牙.pdf】也就是本文档，在进行任何操作前请务必先仔细阅读。

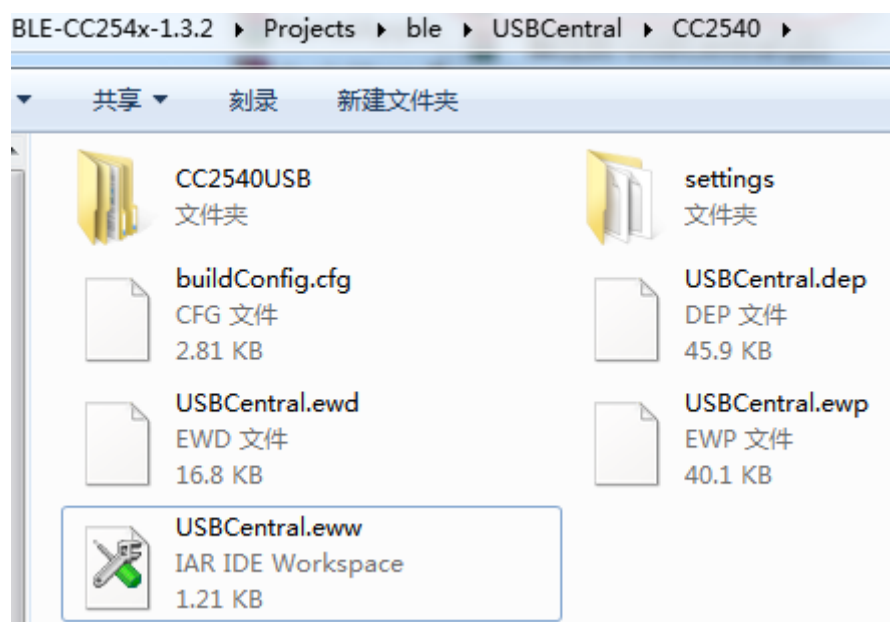
4 源码包解压

将【\实验与实战\7、USB 控制蓝牙\源码\CC254x】下的压缩包，复制到 1.3.2 版本的协议栈 projects 目录下，然后右击选择“解压到当前文件夹”，如下图所示，务必注意，请勿“解压到 xxx”，否则会多一级目录，造成源码编译不通过。



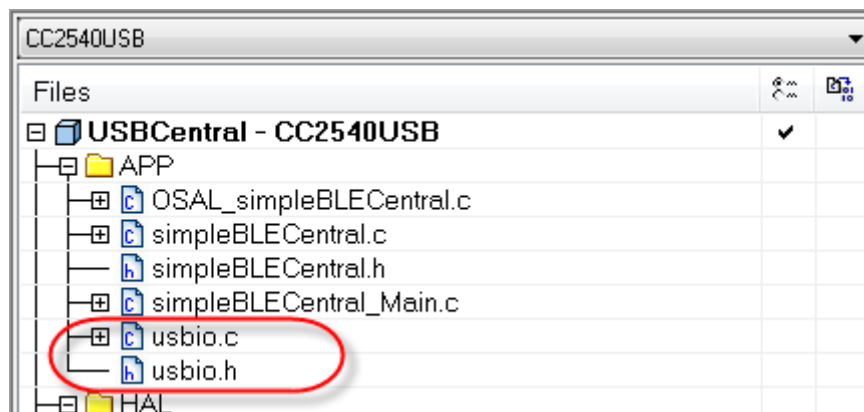
5 打开 IAR 工程

接下来我们打开 USBCentral 工程，进入【BLE-CC254x-1.3.2\Projects\ble\USBCentral\CC2540】，打开 IAR 工程，如下图，如果你使用的芯片是 CC2541，进入 CC2541 文件夹打开工程。



在 APP 目录，我们添加了两个文件，用来配置串口通信，另外，在 simpleBLECentral.c

文件的最后，是 USB 数据回调函数，以及 AT 命令字符串的解析函数。



首先打开 `usbio.c` 源文件，USB 通信的相关代码，包括初始化配置，数据发等。配置中波特率为 115200，并且开启了流控制。

```
void USBIO_InitTransport( usbioCBack_t usbioCBack )
{
    halUARTCfg_t uartConfig;

    // configure UART
    uartConfig.configured           = TRUE;
    uartConfig.baudRate             = USBIO_BR;
    uartConfig.flowControl          = USBIO_FC;
    uartConfig.flowControlThreshold = USBIO_FC_THRESHOLD;
    uartConfig.rx.maxBufSize        = USBIO_RX_BUF_SIZE;
    uartConfig.tx.maxBufSize        = USBIO_TX_BUF_SIZE;
    uartConfig.idleTimeout          = USBIO_IDLE_TIMEOUT;
    uartConfig.intEnable            = USBIO_INT_ENABLE;
    uartConfig.callBackFunc         = (halUARTCBack_t)usbioCBack;

    // start UART
    // Note: Assumes no issue opening UART port.
    (void)HalUARTOpen( USBIO_PORT, &uartConfig );

    return;
} ? end USBIO_InitTransport ?
```

在 `simpleBLECentral.c` 文件的最后是 USB 数据回调函数，当硬件接收到数据后会调用该函数。

```
//接收来自usb发来的数据
void usbioCallback ( uint8 port, uint8 event )
{
    uint16 numBytes;
    uint8 pktBuffer[USBIO_RX_BUF_SIZE];
    (void)event;
    (void)port;
    if ( (numBytes = USBIO_RxBufLen()) > 0 ) {
        (void)USBIO_ReadTransport(pktBuffer, numBytes);
        CommandHandle(pktBuffer, numBytes);
    }
}
```

该函数接收全部的串口数据后，调用 `CommandHandle` 函数开始解析 AT 命令。`CommandHandle` 函数位于 `simpleBLECentral.c` 文件中。如下图程序片段，一共可以处理 7 条 AT 命令，大家可以更具需要添加更多的 AT 命令

AT

用于串口测试，如果程序运行并且串口通畅，会返回 OK

AT+ROLE?

获取当前角色，返回 Central

AT+SCAN

扫描从机，发送后 CC254x 开始 Discovery 从机，等待片刻后，返回找到的从机数量。

AT+CON[x]

连接指定的从机，x 为搜索到的从机序号，如果只扫描到一个从机，可以输入：AT+CON1 连接该从机。

AT+RSSI

获取当前 rssi 值，执行该命令后，程序会每个一秒打印一次 RSSI 值，再次发送该命令，停止 RSSI 值打印。

AT+DISCON

断开连接

AT+WRITE[0xXX]

向 Char1 写入特征值。如果要向从机 char 发送 0x15，输入发送命令：AT+WRITE0x15

```

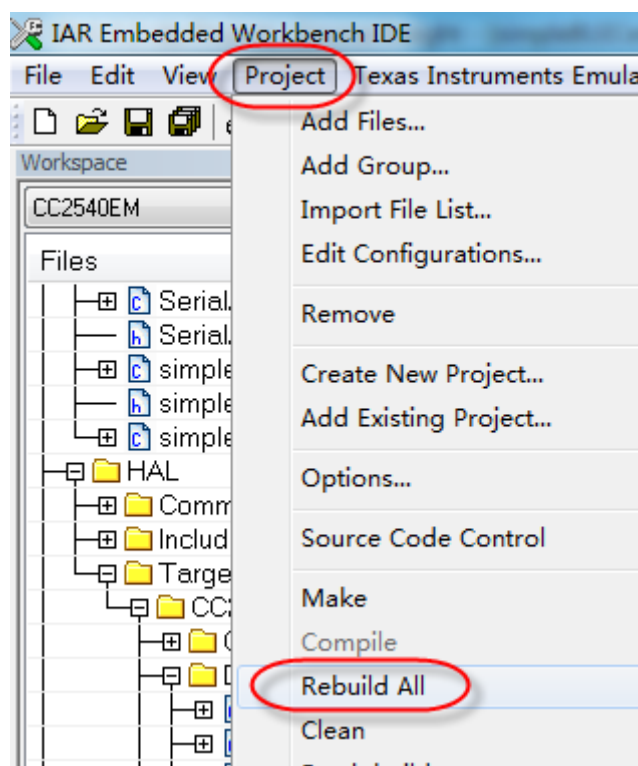
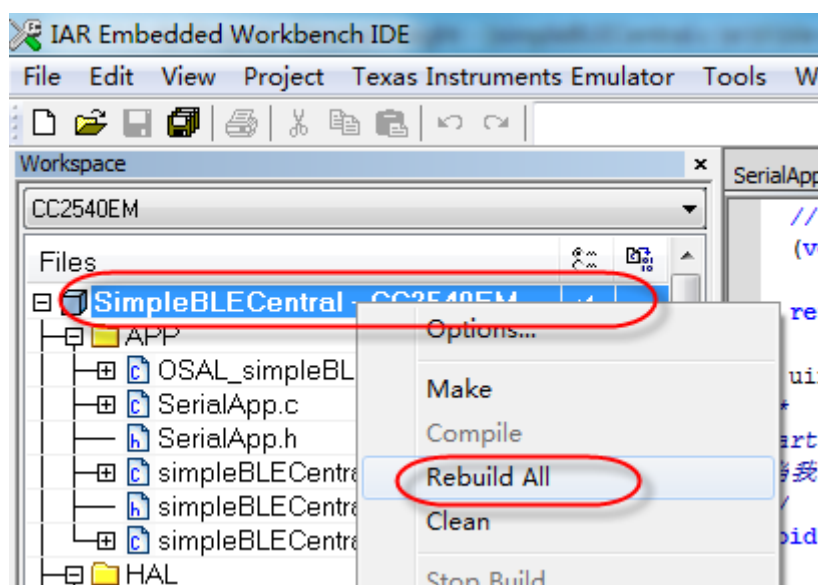
01095: //AT          串口测试, 返回OK
01096: //AT+ROLE?    获取当前角色
01097: //AT+SCAN      扫描从机
01098: //AT+CON[x]    连接指定的从机, x为搜索到的从机序号
01099: //AT+RSSI      获取rssi值
01100: //AT+DISCON    断开连接
01101: //AT+WRITE[0xXX]
01102: void CommondHandle(uint8 *pBuffer, uint16 length)
01103: {
01104:     if(length<2)
01105:         return ;
01106:     if(pBuffer[0]!='A' && pBuffer[1]!='T')
01107:         return ;
01108:     if(length <=4){
01109:         SerialPrintString("OK\r\n");
01110:         return ;
01111:     }
01112:     if(length>=8 && str_cmp(pBuffer+3,"ROLE?",5)==0){
01113:         SerialPrintString("Central\r\n");
01114:         return ;
01115:     }
01116:     if(length>=7 && str_cmp(pBuffer+3,"SCAN",4)==0){
01117:         simpleBLEScanning = TRUE;
01118:         simpleBLEScanRes = 0;
01119:
01120:         LCD_WRITE_STRING( "Discovering...", HAL_LCD_LINE_1 )
01121:         SerialPrintString("Discovering...\r\n");
01122:         LCD_WRITE_STRING( "", HAL_LCD_LINE_2 );
01123:
01124:         GAPCentralRole_StartDiscovery( DEFAULT_DISCOVERY_MO

```

6 编译下载

6.1 编译

在当前 Configuration 上右击, 然后选择 Rebuild All, 重新编译整个工程。或者选择菜单 Project/Rebuild All。效果相同。



如果源码解压的位置正确，并且使用的是 1.3.2 的 ble 协议栈和 8.10 的 IAR 编译器，不会出现任何编译问题。

6.2 下载

连接 CC-Debugger 仿真器和 USBdongle，准备烧写程序，注意，两者的连接需要我们提供的转接板协议栈，详细的连接方法，见【用户手册】目录中的 USBdongle 使用手册。



6.3 驱动安装

该 demo 使用的驱动程序与 HostTestApp 驱动程序完全一样，请参考 USBdongle 使用说明的《2.14 节 安装 HostTestRelease 驱动程序》

7 测试

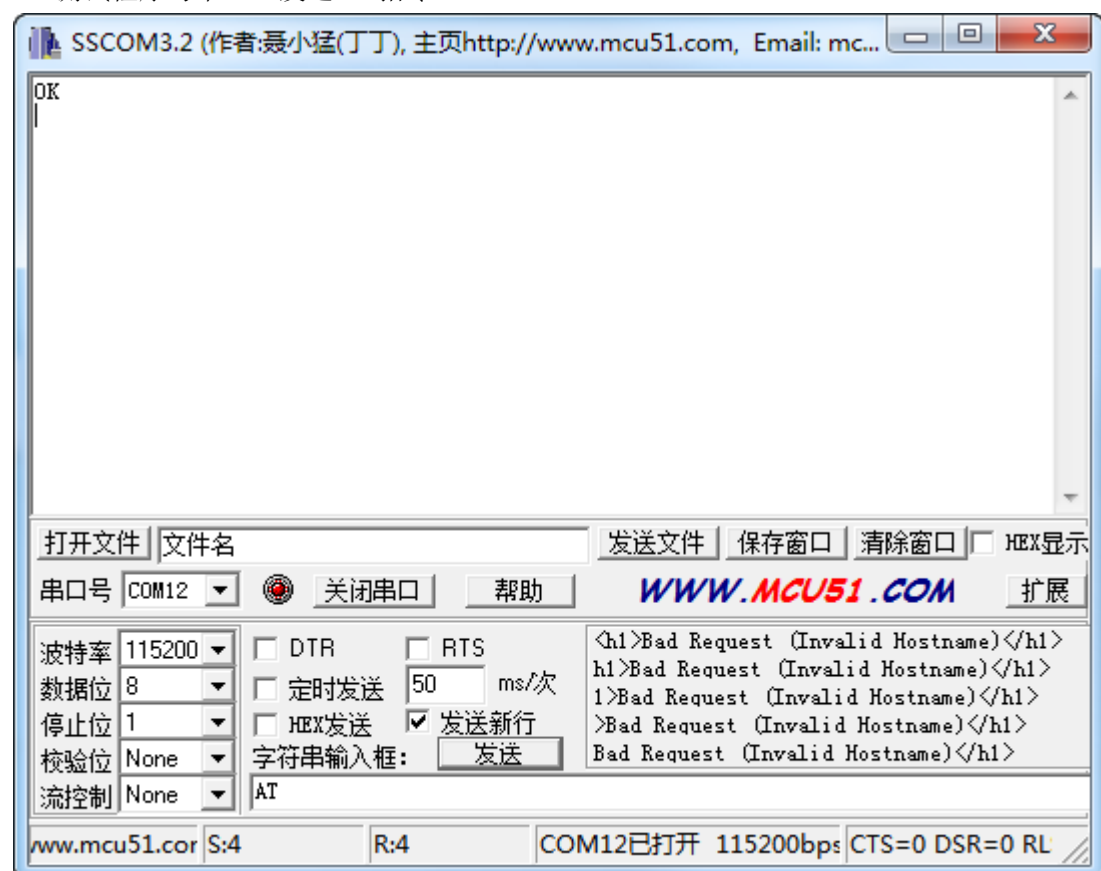
需要一个运行 SimpleBLEPeripheral 从机程序的开发板来辅助测试。请自行准备。

打开串口调试助手，按如下图设置波特率等参数，然后选择与 CC2540USBdongle 开发板匹配的端口号然后打开，注意串口号，开发板连接 PC 的后，需要打开设备管理器，查看 USBdongle 虚拟出来的串口号具体是哪个。我们程序里使用的波特率是 115200，并且开启了 FlowControl 流控制。





1、测试程序与串口，发送 AT 指令。



2、扫描从机，输入：AT+SCAN，点击发送，稍等片刻后会返回扫描结果



3、连接从机，输入：AT+CON1，点击发送，当成功连接后，dongle 会点亮红色 LED



4、获取 RSSI 信号值，输入：AT+RSSI，点击发送，再次发送停止。



5、向 Char1 写入一个数，如 0x5a，输入：AT+WRITE0x5A



6、断开连接，输入：AT+DISCON，断开连接后，dongle 会熄灭红色 LED



Ghostyu.com 谷雨

联系我们:

刘雨 tel:15861666207

网站: <http://www.ghostyu.com>

技术支持: <http://www.ghostyu.com/bbs>

在线文档: <http://www.ghostyu.com/wiki>

官网店铺: <http://ghostyu.taobao.com>