

蓝牙 4.0 连接之配对与绑定

www.AmoMcu.com

阿莫单片机 出品

2014-12-17 v1.0

目录

1, 要实现的功能	1
2, 开发环境	2
2.1 硬件	2
2.2 软件	2
3, 源码位置	2
4, 源码分析	3
5, 源码编译	5
6, 下载运行	5
7, 测试	5
8, 联系我们	6

1, 要实现的功能

TI 1.3.2 的 BLE 协议栈中的 SimpleBLEPeripheral 从机工程并没有实现配对与绑定, 本章, 我们就来实现配对与绑定。

2，开发环境

2.1 硬件

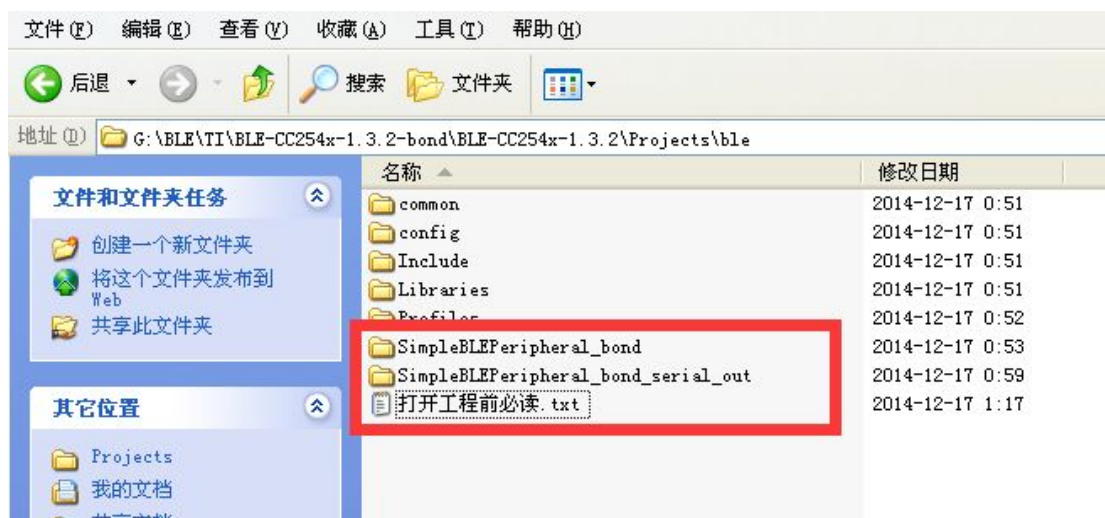
- 1、 SmartRF 系列开发板，CC2540 或者 CC2541
- 2、 CC-Debugger 仿真器
- 3、 MiniUSB 线
- 4、 运行 SimpleBLEPeripheral 从机程序的开发板。

2.2 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、 iphone4s lightblue

3，源码位置

本教案提供两个工程， 如下：



1, SimpleBLEPeripheral_bond 这个工程师会睡眠的，只有 LCD 显示密码，有 oled 显示屏的建议使用这个工程。

2, SimpleBLEPeripheral_bond_serial_out 这个工程师不会睡眠，有 LCD 显示密码，同时串口
[技术支持与项目开发合作\(TEL\) 18588220515](tel:18588220515) [QQ11940507](http://qq.com/11940507) 阿莫

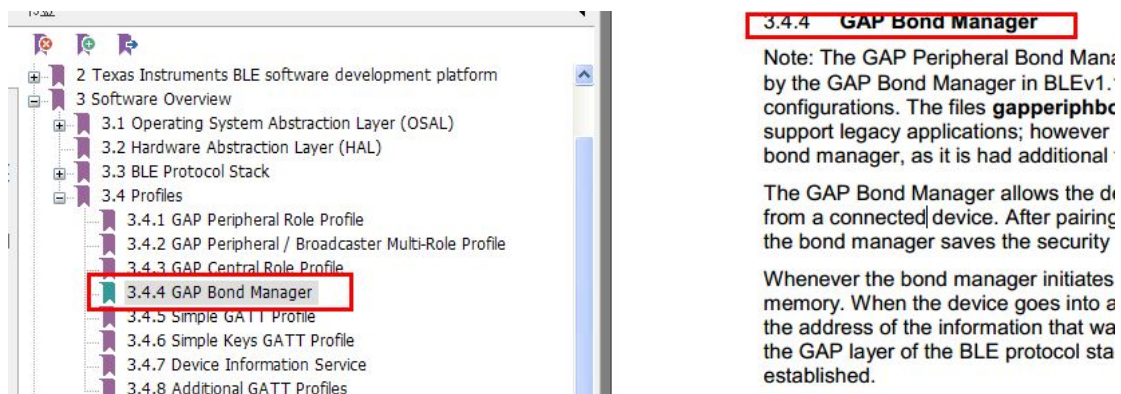
已经打开，并在主机如 ios 的 lightblue 连接他前从串口输出密码，方便观察。想观察串口输出的建议使用此工程。

4，源码分析

4.1 源码是先前的分析

C:\Texas Instruments\BLE-CC254x-1.3.2\Documents\

目录下的《TI_BLE_Software_Developer's_Guide.pdf》文件是 BLE 软件开发的指导文档，这个文档对配对和绑定有详细的讲解，不下图即是讲解的位置：



可见，实际上配对和绑定是在 GAP 层进行的，GAP Bond Manager 就是管理绑定的。

绑定管理是通过下面的两个函数进行的：

【1】GAPBondMgr_SetParameter 设置绑定参数

【2】GAPBondMgr_GetParameter 获取绑定参数

并且，GAPBondMgr_SetParameter 函数需要在从机初始化时进行设置。

有以下的参数需要注意：

- **GAPBOND_PAIRING_MODE** – This parameter tells the bond manager whether pairing is allowed, and if so, whether it should wait for a request from the central device, or initiate pairing on its own. The default setting is to wait for a request from the central device.
- **GAPBOND_MITM_PROTECTION** – This parameter sets whether man-in-the-middle protection is enabled or not. If it is enabled, the pairing request will also authenticate the connection between the slave and master. The profile default value is **FALSE**, though the sample application sets it to **TRUE** during initialization.
- **GAPBOND_IO_CAPABILITIES** – This parameter tells the bond manager the input and output capabilities of the device. This is needed in order to determine whether the device can display and/or enter a passkey. Therefore, the default value is **GAPBOND_IO_CAP_DISPLAY_ONLY**, indicating that the device has a display but no keyboard. Even if the device does not have a physical display, a passkey presented in a user guide (such as this one) can be considered to be a “display”. The default passkey value is a six digit string of zeros: “000000”.
- **GAPBOND_BONDING_ENABLED** – This parameter enables bonding. The profile default value is **FALSE**, though the SimpleBLEPeripheral application sets it to **TRUE** during initialization.

以上就不翻译先了。

4.2 源码实现

在从机程序 SimpleBLEPeripheral 中，要实现密码绑定，首先需要实现 bongmgr 的回调函数。

```
00266: // GAP Bond Manager Callbacks
00267: static gapBondCBs_t simpleBLEPeripheral_BondMgrCBs =
00268: {
00269:     ProcessPasscodeCB,           // 密码回调
00270:     ProcessPairStateCB          // 绑定状态回调
00271: };
```

主要流程是：

【1】主机需要连接从机时，从机的 ProcessPasscodeCB，显示配对密码，并通过 GAPBondMgr_PasscodeRsp 发送密码请求给主机。

【2】然后主机输入密码，确定密码后。。。

【3】从机开始回调 ProcessPairStateCB 函数，该函数需要实现主机发送过来的密码是否对和错的判断。

实际上，主机程序也要实现类似的两个回调函数的，如以下的描述一样：

It is up to the application to provide the actual callback functions. In the case of the SimpleBLECentral sample application (see section 4.4), the passcode callback function is simpleBLECentralPasscodeCB and the pairing state callback function is simpleBLECentralPairStateCB.

Additional details on using the GAP bond manager can be found in [4].

不过，我们的开发板没有键盘可输入密码，因此，我们就不做用开发板做配对实验了，直接用 ios 来测试即可。

具体密码位置在如下：（以 SimpleBLEPeripheral_bond_serial_out 工程为例）

```
00833: //绑定过程中的密码管理回调函数
00834: static void ProcessPasscodeCB(uint8 *deviceAddr,uint16 connectionHandle,uint8 u:
00835: {
00836:     uint32 passcode;
00837:     uint8 str[7];
00838:
00839:     //在这里可以设置存储，保存之前设定的密码，这样就可以动态修改配对密码了。
00840:     // Create random passcode
00841:     #if 0
00842:     LL_Rand( ((uint8 *) &passcode), sizeof( uint32 ));
00843:     passcode %= 1000000;
00844:     #else
00845:     passcode = 123456;           // 连接密码， 固定的 123456
00846:     #endif
00847:     //在lcd上显示当前的密码，这样手机端，根据此密码连接。
00848:     // Display passcode to user
00849:     if ( uiOutputs != 0 )
00850:     {
00851:         HallcdWriteString( "Passcode:", HAL_LCD_LINE_1 );
00852:         HallcdWriteString( (char *) _ltoa(passcode, str, 10), HAL_LCD_LINE_2 );
00853:     }
00854:
00855:     // Send passcode response 发送密码请求给主机
00856:     GAPBondMgr_PasscodeRsp( connectionHandle, SUCCESS, passcode );
00857: } ? end ProcessPasscodeCB ?
```

845 行，我们固定密码为 123456，这个你可以修改的，或者可以吧 841 行的 #if 0 改成 #if 1

那么就是使用一个随机密码了，需要注意的是，例如 ios 手机，一旦跟这个设备连接上

之后，再次连接的话不需要再次输密码的，这就是绑定，已经绑定了，就不需要输密码了哈。

5，源码编译

请看相关章节，或根目录视频

6，下载运行

请看相关章节，或根目录视频

7，测试

请用 ios 的 lightblue 来连接。

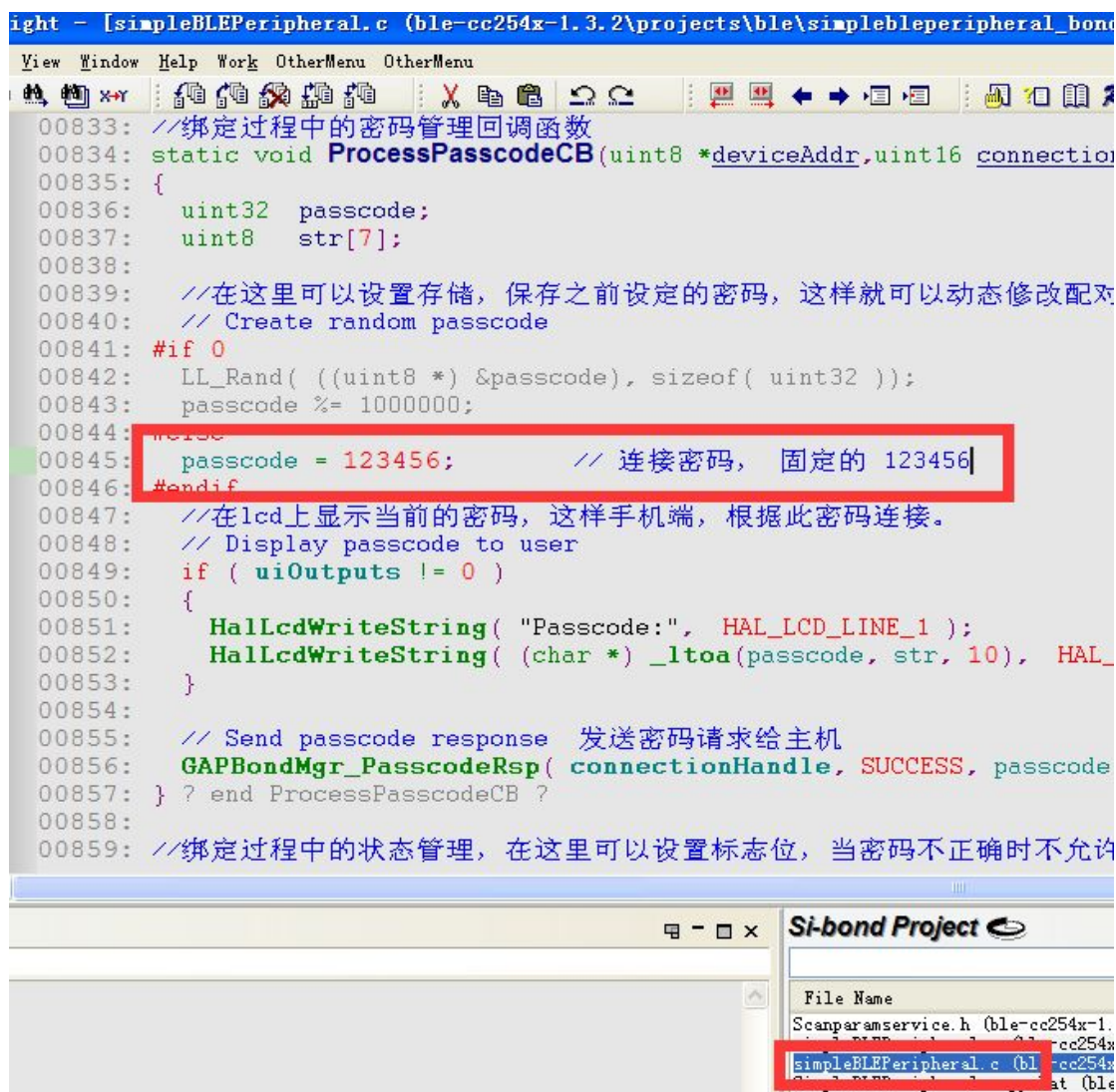
或者

使用 android4.3+ 的手机 用 amomcu.apk （这个是我们提供的）



输入密码 123456 是正确的

源码详见：



8, 联系我们

QQ 群: 257318688

QQ: 11940507

Tel: 18588220515 阿莫

网站支持: www.AmoMcu.com 阿莫单片机社区网

淘宝店铺: <http://amomcu.taobao.com>

