

Tiva C Series LaunchPad 连接AY-CC2564EVM蓝牙模块操作手册

杭州艾研信息技术有限公司
2014 年 11 月

申明

杭州艾研信息技术有限公司保留随时对其产品进行修正、改进和完善的权利，同时也保留在不作任何通告的情况下，终止其任何一款产品的供应的权利。用户在下订单前应及时获取相关信息的最新版本，并验证这些信息是当前的和完整的。

可通过如下方式获取最新信息、技术资料和技术支持：

技术支持电话：0571-86134572

技术支持邮箱：support@hpati.com

产品&资料下载中心：<http://www.hpati.com/products/>

互动论坛：<http://www.hpati.com/bbs/forum.php>

公司地址：浙江省杭州市西湖区留和路16号新峰商务楼B306

Tiva C Series LaunchPad 连接AY-CC2564EVM蓝牙模块操作手册

版本 <1.0>

修订历史记录

日期	版本	说明	作者
<26/07/2014>	<1.0>	<创建文档>	<张恺>

目录

1. 简介	4
1.1 目的	4
1.2 范围	4
1.3 定义、首字母缩写词和缩略语	4
1.3.1 Tiva™ C 系列 LaunchPad 评估套件	4
1.3.2 AY-CC2564EVM 蓝牙模块:	4
1.3.3 CCS (Code Composer Studio) :	4
1.3.4 Bluetopia SDK:	5
1.3.5 串口调试助手 (sscom42 v 4.2)	5
1.3.6 TivaWare	5
1.4 参考资料	5
2. 环境搭建	6
2.1 硬件平台搭建	6
2.1.1 模块插装:	6
2.1.2 连接电脑:	6
2.2 软件环境创建	7
2.2.1 CCS 安装:	7
2.2.2 Bluetopia SDK 安装:	7
2.2.3 Tivaware 安装	7
3. 蓝牙程序的运行	7
3.1 SPPDemo 工作空间的创建:	7
3.2 SPPDemo 工程的导入	9
3.3 TivaWare 环境配置	10
3.4 SPPDemo 工程的移植修改	11
3.4.1 工程属性打开	11
3.4.2 添加 CC256XB 的预编译宏定义	11
3.5 SPPDemo 的代码修改	12
3.5.1 创建属于 Cortex M4 的硬件引用库文件	13
3.5.2 修改 HALCFG.h	13
3.6 SPPDemo 的验证测试	15
3.6.1 硬件接线和链接	15
3.6.2 打开串口调试助手	15
3.6.3 手机端 APP 安装	18
3.6.4 打开蓝牙模块服务	18
3.6.5 开启手机蓝牙	19
3.6.6 蓝牙配对连接	20
3.6.7 蓝牙无线连接	22
3.6.8 数据收发测试	23

EK-TM4C123GXL 移植AY-CC2564EVM 模块操作手册

1. 简介

本手册简要描述的如何从 TI 官方提供的基于 Tiva C Series 系列 MCU 和 AY-CC2564EVM 蓝牙模块的蓝牙通讯例程，包括模块间通讯和模块与手机间通讯的实现。文档内容包含了硬件环境的搭建，软件的安装和配置以及工程的导入和修改、联机调试等多个环节的内容。

1.1 目的

帮助刚刚开始着手从事无线或者物联网开发的工程师掌握最基本的蓝牙通讯 Demo 例程的安装调试；

1.2 范围

仅适用于由 TI 提供的 Tiva C 系列 ARM Cortex-M4 MCU 开发板和 TI 的第三方合作伙伴--杭州艾研信息技术有限公司所提供的 AY-CC2564EVM 蓝牙模块上使用。

1.3 定义、首字母缩写词和缩略语

1.3.1 Tiva™ C 系列 LaunchPad 评估套件

Tiva C 系列 TM4C123G LaunchPad 评估套件是用于德州仪器 (TI) 基于 ARM® Cortex™-M4F 的微控制器的低成本评估平台。TM4C123G LaunchPad 的设计亮点为配有 USB 2.0 器件接口和休眠模块的 TM4C123GH6PM 微控制器。

EK-TM4C123GXL 还具有用于自定义应用的可编程用户按钮和 RGB LED。Tiva C 系列 TM4C123G LaunchPad BoosterPack XL 接口采用可堆叠接头，这样在使用德州仪器 (TI) 的 MCU BoosterPack 连接到其他外设时可以轻松简单地扩展 TM4C123G LaunchPad 的功能。

产品链接：

<http://www.ti.com.cn/tool/cn/ek-tm4c123gx1?keyMatch=tiva%20launchpad&tisearch=Search-CN>

1.3.2 AY-CC2564EVM 蓝牙模块：

AY-CC2564EVM 是基于 TI CC256x 的双模式蓝牙 4.0 的评估板。TI 的电源技术和软件算法使得 CC256X 在蓝牙 BR/EDR/LE 多种模式都比同类产品更节能。评估板采用板载 PCB 天线，无障碍通信距离不小于 10 米；评估板提供标准的 BoosterPack 接口，可以与 TI 的 MSP430 LaunchPad 或 TIVA LaunchPad 直接互联；评估板同时提供非 BoosterPack 标准的连接接口，方便用户将评估板连接到自制的 MCU 系统上。板上有关蓝牙芯片的应用设计可以直接拷贝到用户自己的电路板上以减少产品的开发时间。可应用于无线传感、移动设备扩展附件、工业控制、医疗保健设备和简单的音频设备。

产品链接：

http://www.hpati.com/ay_bluetooth/product_32.html

1.3.3 CCS (Code Composer Studio) :

Code Composer Studio 是一种集成开发环境 (IDE)，支持 TI 的微控制器和嵌入式处理器产品系列。Code Composer Studio 包含一整套用于开发和调试嵌入式应用的工具。它包含了用于优化的 C/C++ 编译器、源码编辑器、项目构建环境、调试器、描述器以及多种其他功能。直观的 IDE 提供了单个用户界面，可帮助您完成应用开发流程的每个步骤。熟悉的工具和界面使用户能够比以前更快地入手。

Code Composer Studio 将 Eclipse 软件框架的优点和 TI 先进的嵌入式调试功能相结合，为嵌入式开发人员提供了一个引人注目、功能丰富的开发环境。

下载链接:

<http://www.ti.com.cn/tool/cn/ccstudio?intc=searchrecs&keyMatch=CCS&tisearch=Search-CN>

1.3.4 Bluetopia SDK:

Stonestreet One 的 Bluetopia 最新版专门针对 TI 的 eXpressDSP 参考框架 3 (RF3) 而精心设计和优化。TI 的参考框架能适应客户的各种应用，为开发商消除初始低层次设计决策的许多麻烦，并使之将更多精力集中在特色产品的开发上。该版本 Bluetopia 构建于 Stonestreet One 性能卓越的 DSP 蓝牙协议栈之上，后者已被众多客户及合作伙伴广泛采用。Bluetopia 适用于蓝牙版本 1.1，具有灵活的可扩展内存空间，并包括范围广泛的 API。众多对基于 DSP 产品开发至关重要的耳机与免提产品的配置文件，以及其它核心和最新发布配置文件也都包括在内。Bluetopia 几乎可适用于任何操作系统及平台。

下载链接:

<http://www.ti.com.cn/tool/cn/stonestreetone-bt-sdk?keyMatch=bluetopia&tisearch=Search-CN>

1.3.5 串口调试助手 (sscom42 v 4.2)

SSCOM v4.2 绿色最新版是一款专业的串口调试软件，能够支持 110-256000bps 波特率，设置数据位 (5678)，校验 (odd, even, mark, space)、停止位 (1, 1.5, 2)，并发送任意的字符串。对于 dtr, rts 信号线也能自由控制输出状态。软件功能:

- 1.接收从串口进来的数据并在窗口显示.显示流畅,可以保持接收大量数据不死机.
- 2.所接收到的数据数据显示方式可以选择为字符方式或者 HEX 方式
- 3.中文显示无乱码,且不影响速度
- 4.串口波特率可以选择为 110bps-256000bps.(波特率>115200 时需要硬件支持)
- 5.可以即时显示存在的串口号.如果您增加了 usb 转串口等设备,串口号也会在列表中出现.
- 6.可以选择“5、6、7、8”四种数据长度.
- 7.可以选择为“1、1.5、2”三种停止位. (1.5 停止位需要硬件支持)
- 8.可以自由选择校验方式.
- 9.可以自由选择流控方式.(某些计算机不能选择硬流控)
- 10.串口设置和字符串操作等设置在程序关闭时自动保存, 打开时自动载入.
- 11.可以在接收窗口按键即发送该键值.

下载链接:

<http://www.uzzf.com/soft/53912.html>

1.3.6 TivaWare

TivaWare 软件包是为了达到简单快速开发基于 Tiva C Series MCU 的的外部软件工具套件包。所有的 TivaWare 都可以免费的无需授权的使用。TivaWare 是为了支持 C Series 软件的开发所有完全使用 C 语言开发，这使得整个开发过程更加的容易和高效、

下载链接:

<http://www.ti.com.cn/tool/cn/sw-tm4c-grl>

1.4 参考资料

- 1、Tiva C Series TM4C123G LaunchPad Evaluation Kit User's Manual.pdf
- 2、Hardware Porting Guidelines.pdf

- 3、Code Composer Studio Development Tools v3.3 Getting Started Guide.pdf
- 4、http://processors.wiki.ti.com/index.php/CC256x_MSP430_Bluetopia_Basic_SPPDemo_APP

2. 环境搭建

环境的搭建包涵了硬件和软件两个方面的准备工作。

2.1 硬件平台搭建

硬件部分包含了 Tiva C Series Launchpad 开发板和双模蓝牙 4.0 评估板(AY-CC2564EVM Module)。如下图所示：左侧的为 Tiva C Series Launchpad，右侧的为双模蓝牙 4.0 评估板。



图 1、MSP430F5529 与 AY-CC2564EVM Module 模块展示

其中值得特别关注的是：在 AY-CC2564EVM 蓝牙模块上有一个拨码开关（上右图），**需要将 UART 档位拨至 ON 而关闭 I2S 连接**。也就是说靠下侧的拨码开关拨向左侧，上面的拨码开发拨向右侧。

2.1.1 模块插装：

两个模块上面都有一组四排 40 个插针的 BoostPack 通用接口，其中 LaunchPad 的背面有一组 BoostPack 的母口，而 CC2564EVM Module 上有一组 BoostPack 的公口。将 CC2564EVM Module 插在 LaunchPad 的背面（**注：两个模块的字体方向须一致！**）插上后如下图所示：



图 2、模块插装效果展示

2.1.2 连接电脑：

如图 2，LaunchPad 拥有一个 Micro USB 接口可以直接连接上电脑的 USB 口。整套模块会由 USB 供电后而自动上电，插上电脑后 LaunchPad 和 CC2564 模块上的指示灯会亮起。

2.2 软件环境创建

2.2.1 CCS 安装:

CCS 的安装参考 CCS 的用户手册即可以。值得强调的是 CCS 是一个强大的集成平台，它能够做很多 MCU、DSP 等不同平台的开发工作，根据用户的使用倾向可以选择安装自己所针对的开发平台。在本说明文档主要针对的是 ARM Cortex M4 MCU 的开发，所以在安装过程中勾选上 Tiva Cortex M4 系列的选项。

2.2.2 Bluetopia SDK 安装:

进入 TI 官网的 SDK 安装连接后，实际下载的是一个叫 CC256XM4BTBLESW-v1.1-Setup.zip 安装压缩包文件，其含义就是基于 M4 MCU 平台用于 CC256X 系列蓝牙的 Bluetooth Lower Energy Software。安装包解压后按照正常的安装一步一步执行即可。默认安装模式会在 C:盘下的 ti 文件下生成一个 CC256x M4 Bluetopia SDK 文件夹，其中包涵了 M4 平台蓝牙开发所需要的各种驱动库文件、文档说明、例程等诸多资源文件。

2.2.3 Tivaware 安装

根据上面的下载链接，下载完成后直接解压默认安装即可。如果按照默认方式安装，完成安装后会在 C:\ti\TivaWare_C_Series-2.1.0.12573 路径上生成一个文件夹。这便是 TivaWare 整个包的位置。而且 TivaWare 在 TI 的官网会不断地有更新包。使用最新的软件包就好。

3. 蓝牙程序的运行

3.1 SPPDemo 工作空间的创建:

建立专属于 Bluetooth 的 CCS 工作空间 (WorkSpace)，为了防止导入工程变得复杂化建议用户直接使用 Cortex M4 自带的 Sample 所在的路径作为工作空间。如果开发者前期的安装都是使用的默认安装那么这个工作空间的位置应该是(C:\ti\Connectivity\CC256X BT\CC256x M4 Bluetopia SDK\v1.1\Cortex_M4\Sample)。

操作方法:

- 1、选中 CCS 菜单: File->Switch Workspace->Other...

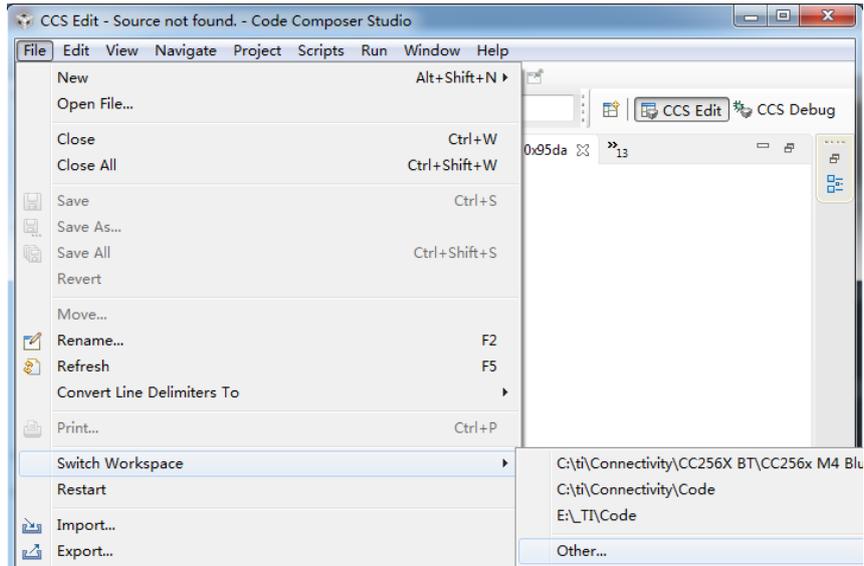


图 3、创建新的 Workspace

2、弹出一个工作空间载入器（Workspace Launcher），点击 Browse...按钮；

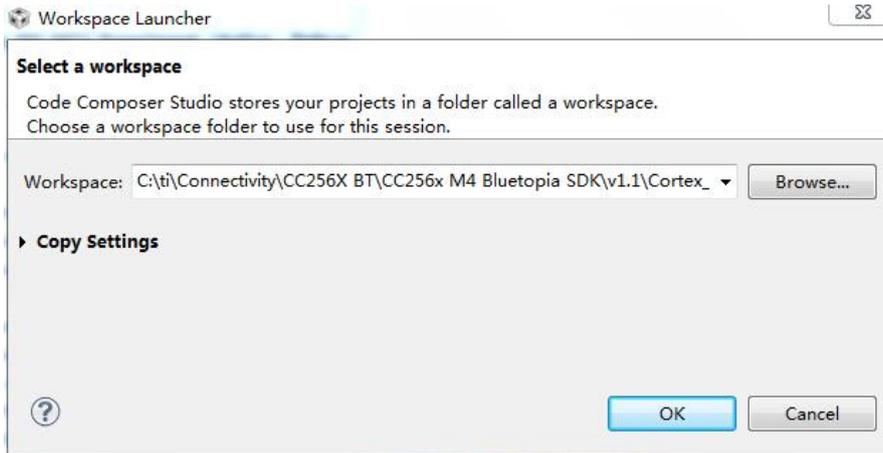


图 4、Workspace 选择提示框

3、找到 (C:\ti\Connectivity\CC256X BT\CC256x M4 Bluetopia SDK\v1.1\Cortex_M4\Sample) 所在的位置选中 Sample 确认即可。或者直接将绝对路径复制到 Workspace 后的输入栏中点击 OK 确认选中都可以。

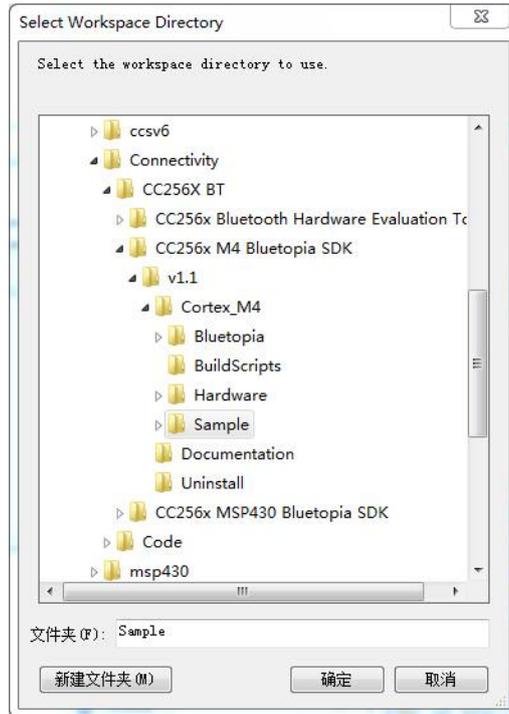


图 5、Workspace 空间选择

完成了上述步骤后，可以从 Sample 所在的系统文件夹中找到类似.metadata 的文件夹。这些都是 CCS 在创建完成 Workspace 后自动生成的文件系统。

3.2 SPPDemo 工程的导入

在 CCS 中导入一个已有的 project 有很多种方式，这里我介绍一种最常用的方式：

- 1、CCS 菜单中选择 Project->Import CCS Project...

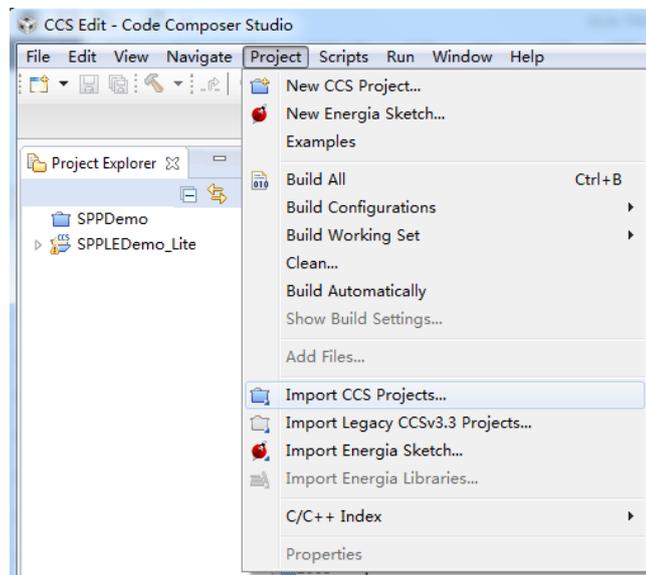


图 5、导入 Project

- 2、点击 Browse...按钮，打开文件系统；

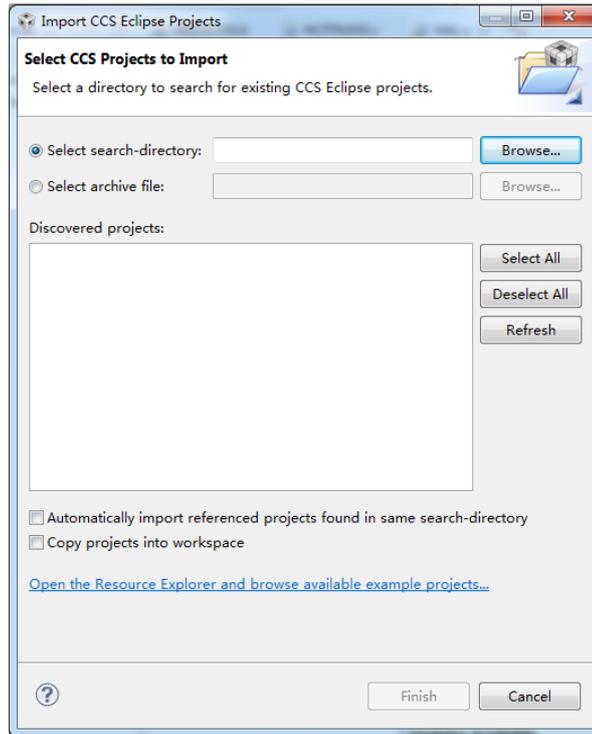


图 6、导入 CCS Project 提示框

2、选择 SPPDemo 文件夹，确认返回；即可完成例程的导入工作。

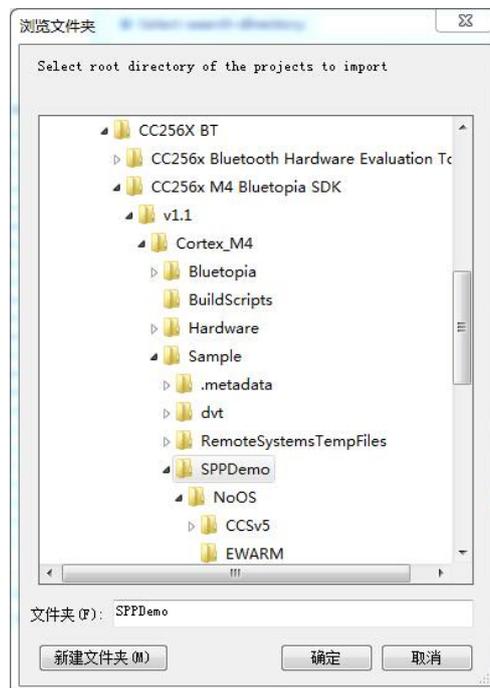


图 7、选择 SPPDemo 例程

3.3 TivaWare 环境配置

在工程中，Project 会使用到 TivaWare 内部的 API，所以在这里需要将 TivaWare API 的环境配置到 CCS 的环境中。打开 C:\ti\Connectivity\CC256X BT\CC256x M4 Bluetopia SDK\v1.1\Cortex_M4\BuildScripts 文件夹里面的 TivaWarePath_DK_TM4C123G_CCS.txt 文本文件，修改里面所有的 TivaWare 的路径信息。将路径修改为开发者本地的安装路径中，默认为 C:\ti\路径下。例

如：

```
-IC:\ti\TivaWare_C_Series-2.1.0.12573  
-IC:\ti\TivaWare_C_Series-2.1.0.12573\third_party  
-IC:\ti\TivaWare_C_Series-2.1.0.12573/examples/boards/dk-tm4c123g  
-iC:\ti\TivaWare_C_Series-2.1.0.12573\driverlib\ccs\Debug  
-iC:\ti\TivaWare_C_Series-2.1.0.12573\gplib\ccs\Debug
```

3.4 SPPDemo 工程的移植修改

3.4.1 工程属性打开

打开工程属性框。右键工程名称弹出列表选中下方 Properties 选项

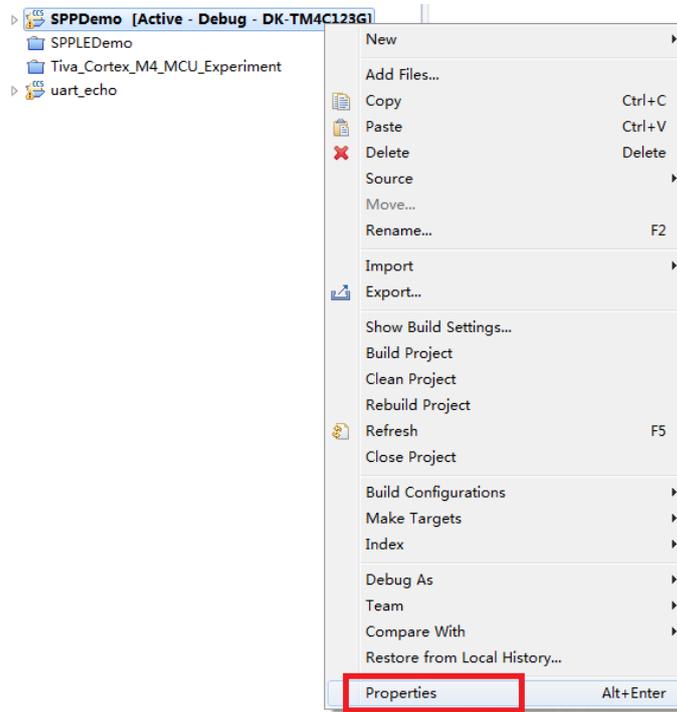


图 8、打开 Project 工程属性

3.4.2 添加 CC256XB 的预编译宏定义

在 Build->MSP430 Compiler->Advanced Options->Predefined Symbols 下添加“`__SUPPORT_CC256XB_PATCH__`”预编译宏定义名。原因在于我们使用的 CC256XB 系列。

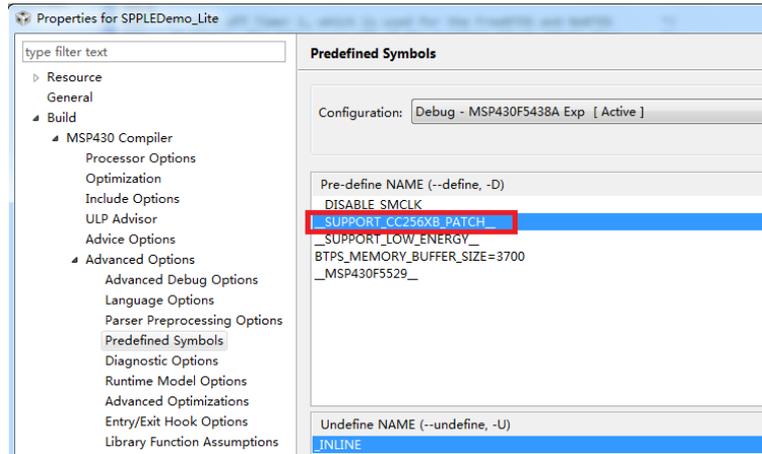


图 12、添加 CC256XB 系列预编译宏定义

3.5 SPPDemo 的代码修改

代码的修改主要是针对 M4 LaunchPad 的管脚做响应的调整。在这里我们抽取出了它的主要内容加以简单描述，便于快速调整开发。

首先我们先将 AY-CC2564EVM Module 使用的端口罗列出来：

CC256x 引脚信号	LaunchPad				CC256x 引脚信号		CC256x 引脚信号	LaunchPad				CC256x 引脚信号
	定义	LP1	LP1X	定义				定义	LP2X	LP2	定义	
+3.3V	+3.3V	○	○	+5V				PWM OUT/GPIO	○	○	GND	GND
	Analog In	○	○	GND				PWM OUT/GPIO	○	○	PWM OUT/GPIO	
HCI_TX	UART_RX	○	○	Analog In				PWM OUT/GPIO	○	○	SPI_CS	
HCI_RX	UART_TX	○	○	Analog In			HCI_CTS	PWM OUT/GPIO	○	○	GPIO	
nSHUTD	GPIO	○	○	Analog In			HCI_RTS	TimerCap/GPIO	○	○	RST	
	Analog In	○	○	Analog In				TimerCap/GPIO	○	○	SPI_MOSI	
	SPI_CLK	○	○	Analog In				GPIO	○	○	SPI_MISO	
	GPIO	○	○	Analog In				GPIO	○	○	GPIO	
	IIC_SCL	○	○	Reserved				GPIO	○	○	GPIO	
	IIC_SDA	○	○	Reserved				GPIO	○	○	GPIO	

图 13、AY-CC2564EVM Module 端口对应表

其中能够看到橙黄色标注的是 3.3V、GND、HCI_TX、HCI_RX、nSHUTD、HCI_CTS、HCI_RTS 七个端口。由于都是使用了 BoostPack 的通用接口，CC2564 和 Cortex M4 的对应关系如下表所示：

表 1、Cortex M4 与 CC2564 BoostPack 管脚对应表

Cortex M4 LP	CC2564
3.3V	3.3V
GND	GND
PB0	HCI_TX
PB1	HCI_RX
PE4	nSHUTD
PC4	HCI_CTS
PC5	HCI_RTS

3.5.1 创建属于 Cortex M4 的硬件引用库文件

我们必须根据上面分析，结合 Cortex M4 LP 特定的接口管脚配置硬件端口初始化。首先需要创建属于 M4 LP 自己的硬件初始化函数库。

3.5.2 修改 HALCFG.h

下面代码展示了原有的 HALCFG.h 里面的管脚定义，它是原程序硬件配置头文件。结合上面的列表修改配置代码：

```

/* Here is the default allocation of pins to connect to the Bluetooth*/
/* device, using UART1. */
/* */
/* Function Port/Pin */
/* ----- */
/* RX PC4 */
/* TX PC5 */
/* RTS PF0 */
/* CTS PF1 */
/* RESET PF2 */
/* */

```

3.5.2.1 RESET 定义修改

将源代码中关于 HCI_RESET_PERIPH、HCI_RESET_BASE、HCI_RESET_PIN 修改为如下的代码。

```

/* Define the GPIO ports and pins that are used for the Bluetooth */
/* RESET signal. */
#define HCI_RESET_PERIPH SYSCTL_PERIPH_GPIOE
#define HCI_RESET_BASE GPIO_PORTE_BASE
#define HCI_RESET_PIN GPIO_PIN_4

```

3.5.2.2 UART RX/TX 定义修改

查找代码中关于下面所示代码的定义将原有 UART 管脚定义修改成如下所示的端口：

```

/* Define the GPIO ports and pins that are used for the UART RX/TX */
/* signals. */
/* * NOTE * See gpio.h for possible values for HCI_PIN_CONFIGURE_ */
/* macros. */
#define HCI_UART_GPIO_PERIPH SYSCTL_PERIPH_GPIOB
#define HCI_UART_GPIO_BASE GPIO_PORTB_BASE
#define HCI_UART_PIN_RX GPIO_PIN_0
#define HCI_UART_PIN_TX GPIO_PIN_1
#define HCI_PIN_CONFIGURE_UART_RX GPIO_PB0_U1RX
#define HCI_PIN_CONFIGURE_UART_TX GPIO_PB1_U1TX

```

3.5.2.3 UART RTS/CTS 修改

同样的方法把 UART RTS/CTS 的管脚定义修改为下列所示的源码：

```

/* Define the GPIO ports and pins that are used for the UART RTS */
/* signal. */
#define HCI_UART_RTS_GPIO_PERIPH SYSCTL_PERIPH_GPIOC
#define HCI_UART_RTS_GPIO_BASE GPIO_PORTC_BASE
#define HCI_UART_PIN_RTS GPIO_PIN_4
#define HCI_PIN_CONFIGURE_UART_RTS GPIO_PC4_U1RTS

```

```

/* Define the GPIO ports and pins that are used for the UART CTS */
/* signal. */
#define HCI_UART_CTS_GPIO_PERIPH      SYSCTL_PERIPH_GPIOC
#define HCI_UART_CTS_GPIO_BASE        GPIO_PORTC_BASE
#define HCI_UART_PIN_CTS                GPIO_PIN_5
#define HCI_PIN_CONFIGURE_UART_CTS     GPIO_PC5_U1CTS

```

到这里，我们完成了 SPPDemo 在 Cortex M4 LP 平台调试运行的所有的工程建立、修改配置、代码调整的内容。接下来可以对整个工程进行编译工作，首先右键点击工程在弹出菜单中选择 Clean Project 清除原有的工程 Debug 过程中生成过程文件，然后重新编译工程 Build Project。

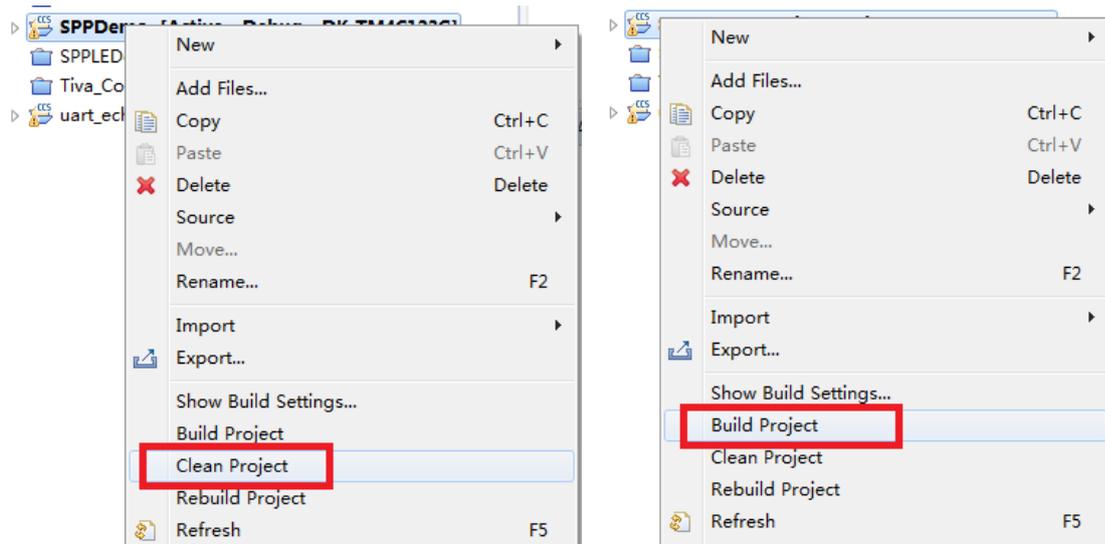


图 18、Project 工程清理和重新编译

如果完成了编译而且未发现 Errors，表示整个 project 编译通过并生成了 SPPDemo.out 的文件。若没有成功，请查找相应的报错信息，并根据上文的配置方式调整配置或者 Project 内容直至编译完成。

最后将 Tiva C Series LP 和 CC2564EVM 模块板链接上 PC 机，点击 CCS 界面中的绿色小虫子 (Debug) 将编译生成的文件仿真烧写到 LP 中。

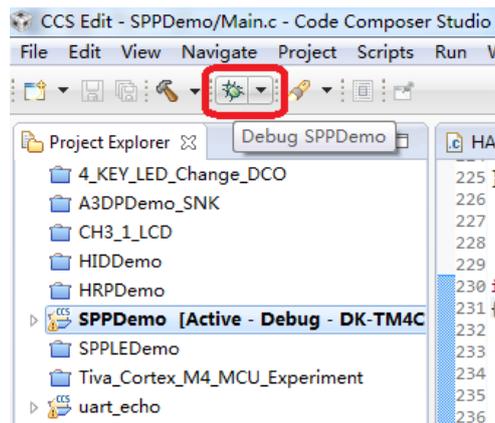


图 19、Project 的仿真烧写到目标板

完成烧写后，程序会自动跳转到程序入口主函数 (Main) 的第一行可执行代码上等待执行命令，接下来点击调试功能框中的绿色三角按钮 (Resume) 执行运行操作，程序就会开始正常的运行起来。如图 20 所示：

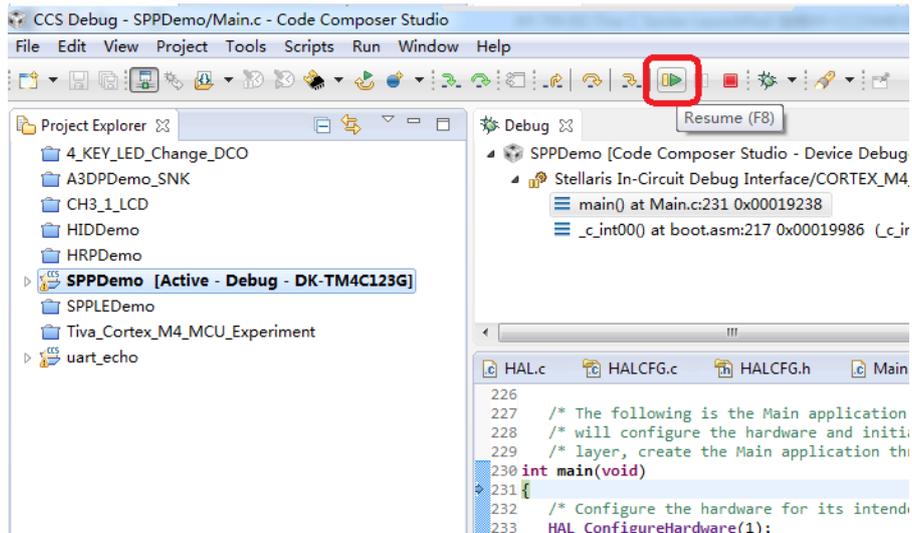


图 20、烧写完成后的结果

3.6 SPPDemo 的验证测试

成功的完成了工程的烧录后，接下来就可以测试无线蓝牙模块是否真正的可以完成无线的链接和数据传输功能。同时开发者也完全可以参考

http://processors.wiki.ti.com/index.php/CC256x_MSP430_Bluetopia_Basic_SPPDemo_APP链接的内容自行学习调试。下文主要基于如上链接，结合 CC2564 蓝牙模块的实际整理出来的 Demo 运行具体步骤。

操作步骤如下：

3.6.1 硬件接线和链接

如果已经能够完成程序的烧写说明链接方式正确，只要保持烧写时候的状态即可。如下图所示：

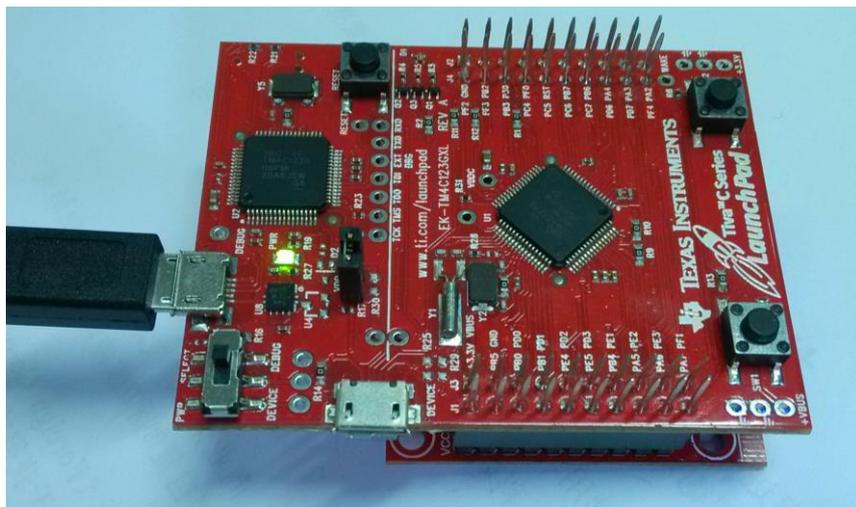


图 21、硬件连接示意图

3.6.2 打开串口调试助手

串口调试助手 (sscom4.2) 可以通过 LP 的 Debug UART 跟踪监控 LP 的运行状态，并将运行状态信息反馈在串口调试助手的控制台信息框中。

完成了硬件连接后，打开电脑设备管理器。如果开发者本地电脑安装过了 Stellaris® ICDI Drivers 的话能够在设备管理器中看到如下驱动显示效果：

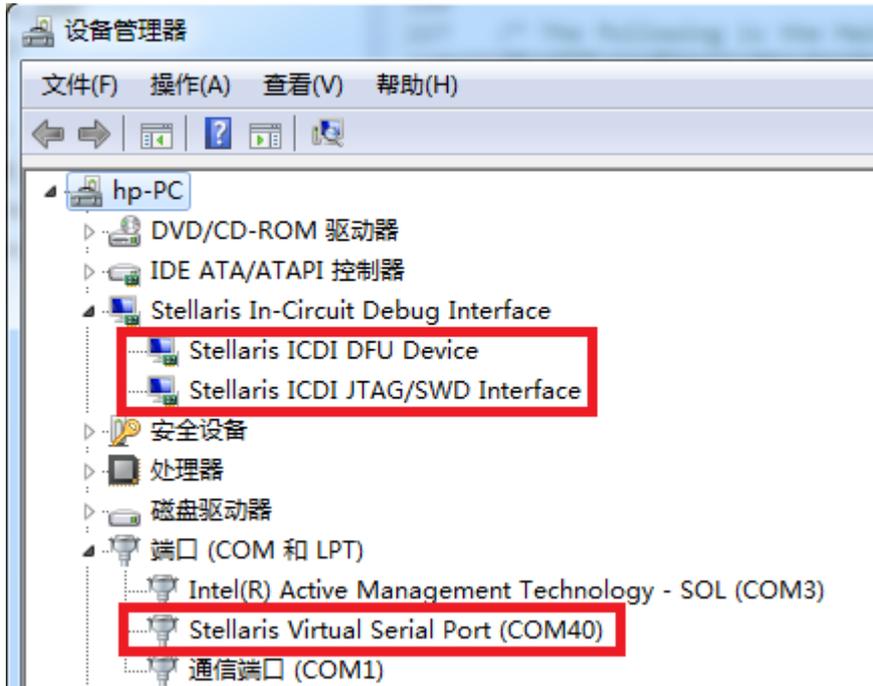


图 22、ICDI 驱动

其中可以看到管理器中多了三个设备，其中 COM 中的 Stellaris Virtual Serial Port (COM40) 就是 LP 的虚拟串口。（关于 ICDI 的安装开发者可以在其他文档中查找，这里不再赘述）。

接下来打开 sscom42 串口调试助手，在串口号下拉栏中找到设备管理器中虚拟串口的端口 COM40。根据设备管理器对于硬件 COM 端口的分配机制不同，不同的电脑上可能端口号的数字略有差异，开发者只要选择和设备管理器显示的虚拟串口一致的端口号即可。

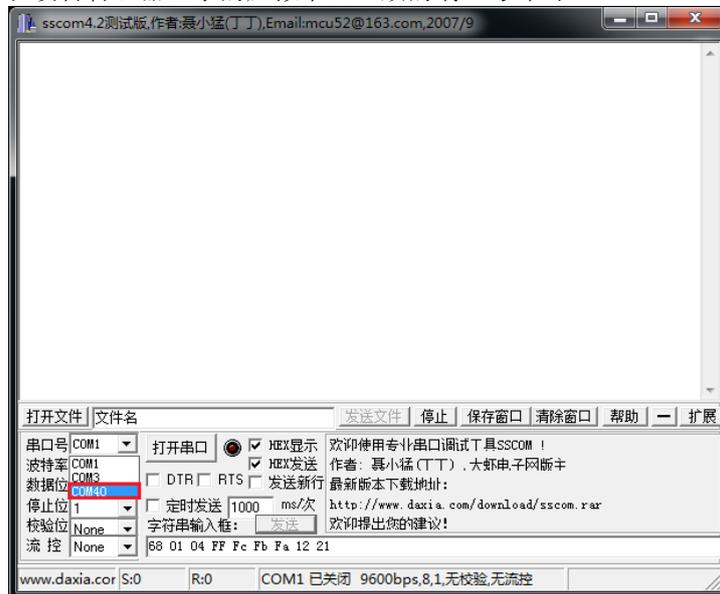


图 23、串口号调整

调整串口通讯波特率，Cortex M4 LP 在和上位机的通讯 UART 设置的是 115200 的通讯波特率，所以在波特率下来菜单中选择 115200。

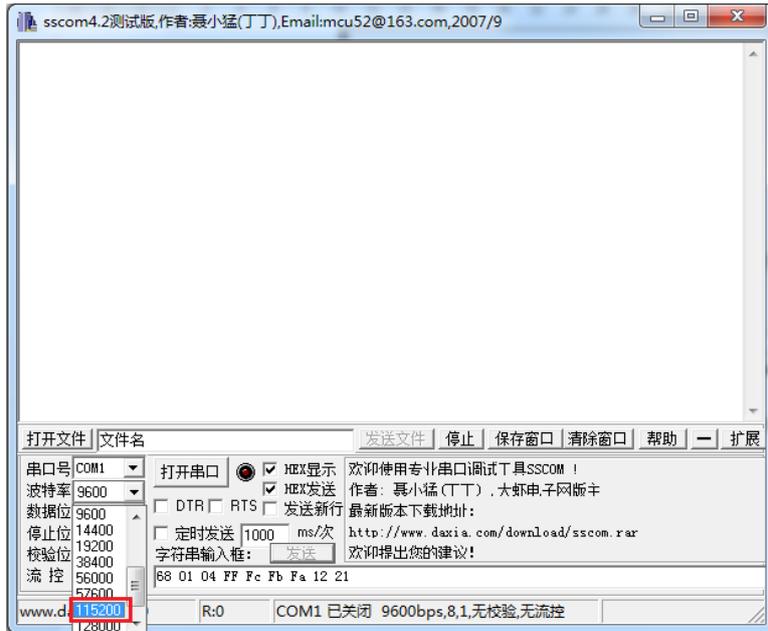


图 24、波特率设置

关闭 HEX 显示，将 HEX 显示和发送的勾选去掉，在这里不需要使用 HEX 显示功能。

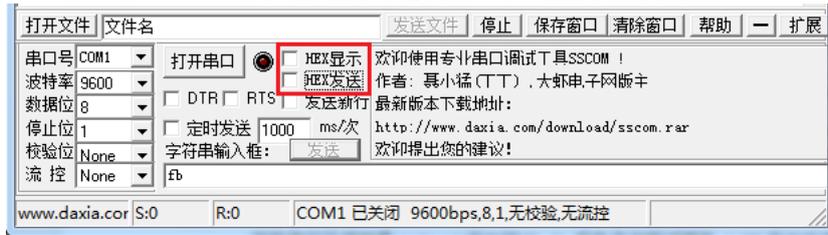


图 25、HEX 显示功能关闭

点击“打开串口”按钮，确认打开串口通讯连接。如果串口通讯已打开，指示灯会由暗变亮。



图 26、打开串口

完成了这一系列操作后，按下 Cortex M4 LP 开发板上的 RESET 按键，使得 LP 复位重启。完成重启后，能够在串口调试助手中显示如下信息：

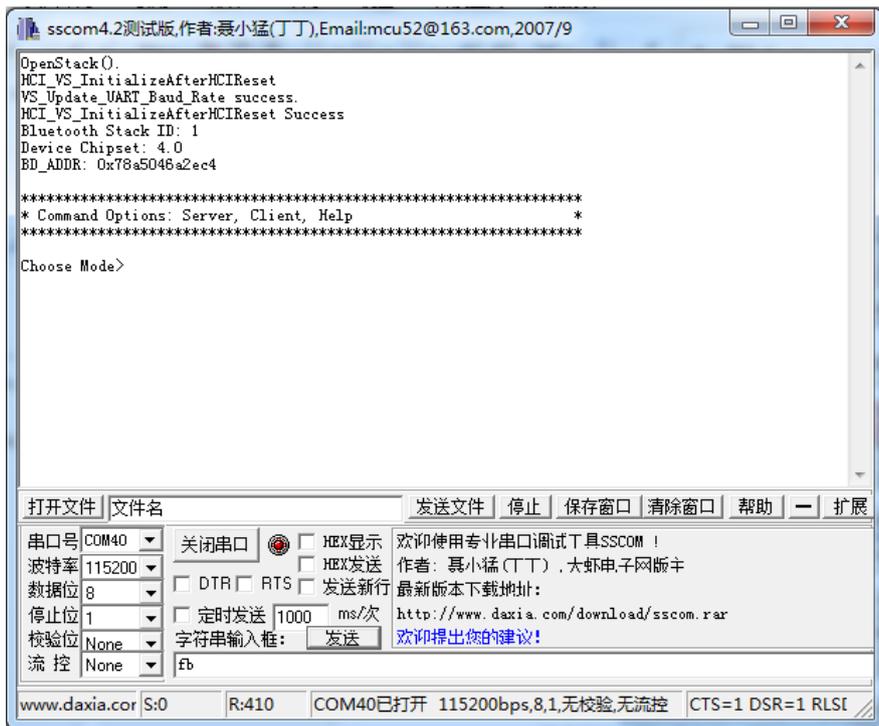


图 27、串口调试助手运行示意图

如果能完成到这一步，表示基本完成了硬件的链接和调试环境搭建工作。

3.6.3 手机端 APP 安装

目前我们采用的是 Android 平台的 Mi 2s 手机作为测试样机（iOS 平台尚未做测试）。在测试手机上安装一个蓝牙串口的 APP 程序。安装方式和普通的手机应用一样，可以从第三方手机应用管理软件中搜索安装也可以打开浏览器直接下载安装。安装完成后，会在手机桌面找到一个蓝牙串口的图标。

下载链接：

<http://apk.gfan.com/Product/App234389.html>



图 28、蓝牙串口 APP 安装后效果

3.6.4 打开蓝牙模块服务

当按下 Cortex M4 LP 上的 RESET 复位键后，串口终端会显示出如下内容，OpenStack () 表示蓝牙通讯协议栈已经成功的打开。而 BD_ADDR 表示的是目前调试的蓝牙模块的设备 ID 号。根据提示信息，蓝牙模块可以工作在服务端、客户端两种模式下，也可以输入 Help 查看支持的命令行操作。本实验采用的是蓝牙模块工作在服务端模式。所以在控制台键入命令“Server”，将蓝牙设置为服务模式。完成输入后，控制单会显示当蓝牙工作在服务模式后能够支持的命令的类型。如下图所示：



图 29、蓝牙设置为服务模式

命令提示行显示内容已经由 Choose Mode>变为了 Server>，现在蓝牙服务已经打开，为了能够使得服务正常运行还需要在 Server>后键入“Open 1”。开发者也可以将 1 替换为 1~30 的任何一个数字。表示的是服务打开了一个通讯端口 1。如下所示：



图 30、打开服务端口

3.6.5 开启手机蓝牙

打开手机设备管理功能，启动蓝牙功能并搜索蓝牙设备。经过一段时间的搜索便能够找到可用设备

78: A5: 04: 6A: 2E: C4。这就是上文中在串口终端控制台看到的蓝牙模块的设备 ID 号，而且这个设备 ID 号是唯一的，每个蓝牙模块都有唯一的编号。



图 31、启动蓝牙发现蓝牙模块

3.6.6 蓝牙配对连接

在可用设备中发现蓝牙设备后,点击这个设备会弹出一个蓝牙配对的对话框提示输入配对所需的 PIN 密码。请输入 0000（密码可以由开发者自行定义，只要保证蓝牙连接双方的 PIN 密码一致即可），完成输入后，点击右下角的确认按钮发送手机蓝牙连接请求。如下如所示：

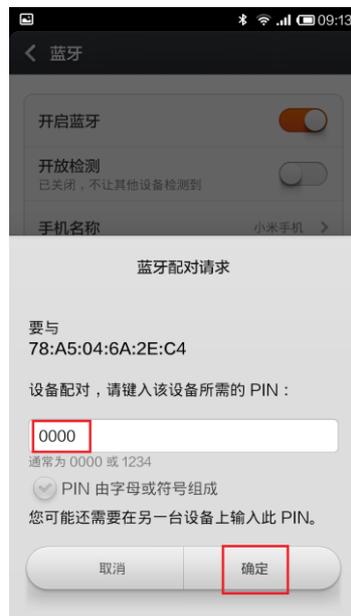


图 32、蓝牙配对 PIN 密码输入

这时观察串口控制台显示信息会发现弹出了一条叫 PINCodeRequest 的消息。后面的 ID 就是手机的设备 ID 号。

```

sscom4.2测试版,作者:聂小猛(丁丁),Email:mcu52@163.com,2007/9
BD_ADDR: 0x78a5046a2ec4
*****
* Command Options: Server, Client, Help
*****
Choose Mode>server
*****
* Command Options: Inquiry, DisplayInquiryList, Pair,
* EndPairing, PINCodeResponse, PassKeyResponse,
* UserConfirmationResponse,
* SetDiscoverabilityMode, SetConnectabilityMode,*
* SetPairabilityMode,
* ChangeSimplePairingParameters,
* GetLocalAddress, GetLocalName, SetLocalName,
* GetClassOfDevice, SetClassOfDevice,
* GetRemoteName, SniffMode, ExitSniffMode,
* Open, Close, Read, Write,
* GetConfigParams, SetConfigParams,
* GetQueueParams, SetQueueParams,
* Loopback, DisplayRawModeData,
* AutomaticReadMode, SetBaudRate, Send
* Help, Quit
*****
Server>open 1
Server Opened: 1.
Server>
atPINCodeRequest: 0xf8a45f0ead89
Respond with: PINCodeResponse

```

图 33、蓝牙模块接收到连接请求

接下来响应手机蓝牙连接请求，在控制台命令行 Server>后键入“PINCodeReponse 0000”，表示回复同意手机的链接请求。如果手机端的 PIN 配对密码不是 0000 也可以，只要保持一致都能通过配对连接认证。

```

sscom4.2测试版,作者:聂小猛(丁丁),Email:mcu52@163.com,2007/9
BD_ADDR: 0x78a5046a2ec4
*****
* Command Options: Server, Client, Help
*****
Choose Mode>server
*****
* Command Options: Inquiry, DisplayInquiryList, Pair,
* EndPairing, PINCodeResponse, PassKeyResponse,
* UserConfirmationResponse,
* SetDiscoverabilityMode, SetConnectabilityMode,*
* SetPairabilityMode,
* ChangeSimplePairingParameters,
* GetLocalAddress, GetLocalName, SetLocalName,
* GetClassOfDevice, SetClassOfDevice,
* GetRemoteName, SniffMode, ExitSniffMode,
* Open, Close, Read, Write,
* GetConfigParams, SetConfigParams,
* GetQueueParams, SetQueueParams,
* Loopback, DisplayRawModeData,
* AutomaticReadMode, SetBaudRate, Send
* Help, Quit
*****
Server>open 1
Server Opened: 1.
Server>
atPINCodeRequest: 0xf8a45f0ead89
Respond with: PINCodeResponse
Server>PINCodeResponse_0000
GAP_Authentication_Response(U, Pin Code Response Success.
Server>
atLinkKeyCreation: 0xf8a45f0ead89
Link Key Stored.

```

图 34、响应手机端配对认证

等待片刻后 SPPDemo 便会由可用设备变换为已配对设备。如下图所示：



图 35、完成蓝牙连接配对

完成配对后，尾号 C4 的蓝牙设备在手机蓝牙管理器中有可用设备列表中变换到了已配对设备列表中，完成配对的设备就能够直接使用蓝牙串口 APP 实现链接了。

3.6.7 蓝牙无线连接

完成手机和蓝牙模块的配对后，可以打开蓝牙串口程序（SPP APP）。它的界面比较简单，单击 APP 左下角的连接按钮后就会弹出一个可以链接的蓝牙设备列表其中就包涵我们刚刚配对成功的蓝牙模块。选中后便会直接连接上。



图 36、蓝牙串口连接模块成功示意图

3.6.8 数据收发测试

3.6.8.1 手机端->cc2654 端

直接在 SPP APP 的 text 对话框中输入一些简单的字符串，点击界面右下角的发送（只有真正的连上蓝牙模块才能发送字符串）。便会在界面中央看到由手机发送给蓝牙模块的数据。



图 37、SPP APP 发送数据

从手机端发送数据的同时，观察串口终端控制台显示发现会有收到一个 SPP 数据包的对应提示信息。这便是从手机端发送而来的数据提示。在串口终端控制台 Server>后可以键入“read”后就能在 CC2564 蓝牙端看到从手机端发送过来的“123456e”数据。

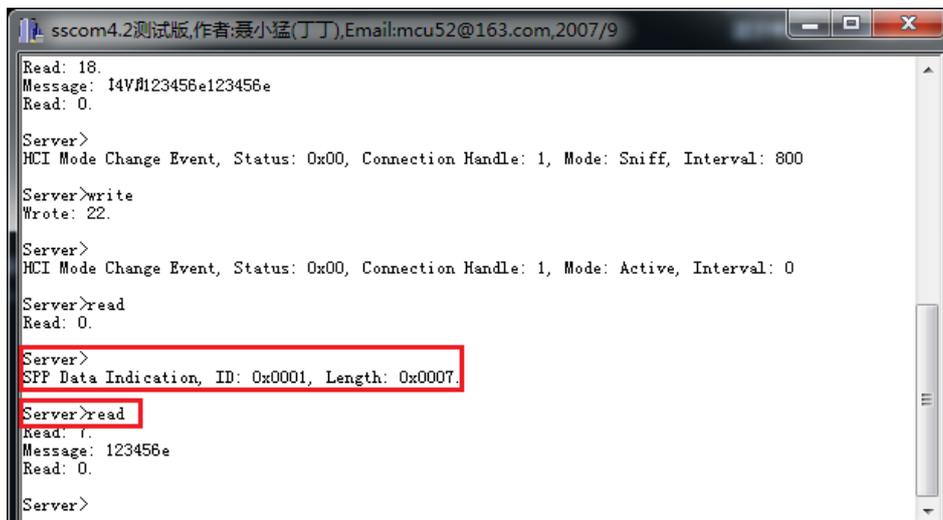


图 38、CC2564 接收数据

3.6.8.2 CC2564 端->手机端

同样我们能通过在控制台键入“Write”命令，表示从 CC2564 蓝牙模块端向手机端发送一串固定的字符串。

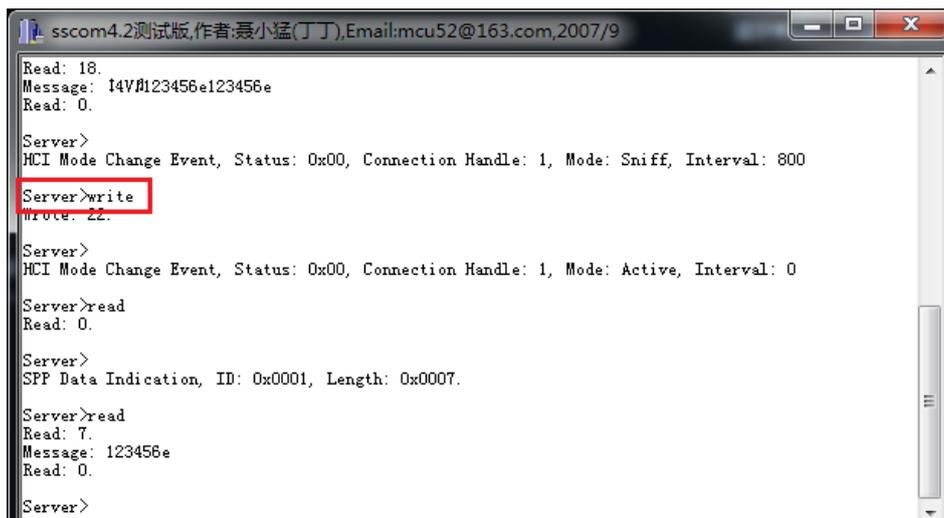


图 39、CC2564 发送数据

而在手机的 SPP APP 的显示框内便能开到由 CC2564 发送上来的一串“**This is a test string.**”的字符串数据。如下图所示：



图 40、手机端接收 CC2564 数据