

第 3 章 蓝牙 4.0 BLE 硬件开发平台搭建

本章首先介绍蓝牙 4.0 BLE 开发板及相关电气接口,再手把手教大家如何快速搭建硬件实验平台,因为软、硬件开发环境都已经说明,最后再一步步讲解如何通过“狸猫换太子”,利用协议栈原有项目实现一个简单项目——点亮 LED 灯,并讲解调试和下载过程。本章内容是今后蓝牙协议栈开发的基础。

3.1 选择世嵌开发板的理由

作为蓝牙 4.0 BLE 的开发,硬件用的是 TI 的 CC2540,软件用的是与 TI 配套的蓝牙协议栈,自然而然,就会想用与 TI 配套的 CC2540 开发板,也就是 CC2540DK。如果这样当然最好,协议栈的硬件驱动完全基于对应的 CC2540DK 开发板,软、硬件完全兼容,不需要更改任何底层驱动即可开发应用程序。好虽好,但 299 美元的价格似乎有点贵。而且如果用的是其他厂商的开发板,我们正好可以针对其不同的硬件配置开发相应的驱动程序,从而更能熟悉 CC2540 的驱动开发,而与 TI 配套的开发板则不需要开发驱动,反而对学习不利。正所谓塞翁失马,焉知非福!

目前市场上关于蓝牙开发的有好几个公司的产品,为什么要选世嵌的开发板呢?这里我要说,其实选哪家都不重要,开发板大体上都差不多,若仔细查看硬件设计,其实硬件设计乃至对外接口都差别不大,掌握了一种开发板,在其他的开发板上也是一样开发,毕竟都是基于 TI 的蓝牙协议栈的,开发的软件平台一致,无非稍加修改一些不同的硬件配置而已。但还是要回答为什么选择世嵌。世嵌应该是在 TI 发布了 CC2540 及相应开发板之后,不久就推出了自己的 CC2540 开发板,在 2011 年当我开始第一个蓝牙 4.0 BLE 项目时,遍寻国内的蓝牙 4.0 开发板,仅仅找到了世嵌,世嵌应是国内首家做蓝牙 4.0 BLE 开发板的。而且在我这几年的蓝牙 4.0 开发中,世嵌的产品表现还是比较稳定的。所以,因为稳定,所以选用。且世嵌的开发板拥有丰富的板上资源,有光敏电阻、DS18B20、4 个 LED、4 个按键、蜂鸣、USB 接口,且所有 CC2540 芯片上的能使用的对外接口都引出,这些丰富的接口大大方便了实验教学和学习。开发板的引脚封装与 315/433MHz WS1110 模块以及 2.4GHz WZ2530 ZigBee 模块全面兼容,用户只需要一次硬件设计就可以通过选用不同的射频前端,推出兼容不同标准的无线通信产品,也就是兼容性好。另外,与 TI 的 299 美元相比,世嵌的最小开发套件价格当前仅 300 多元,注意一个是美元,一个是元。世嵌的最小开发套件,全称是“CC2540 蓝牙 4.0 低功耗最简配置开发板套件(2 模块+2 底板+1 仿真器)”,在淘宝的世嵌商城可以买到,也可以进入世嵌的官网 <http://www.worldembed.com/vw/cn/index.asp>,然后单击“世嵌科技淘宝店”,进入其淘宝店铺,如图 3-1 所示。

在“本店搜索”栏输入“CC2540 蓝牙 4.0 低功耗最简配置开发板套件(2 模块+2 底板+1 仿真器)”即可搜到该产品,如图 3-2 所示。我们也不需要那些动辄好几千元的企业版开发



图 3-1 世嵌科技官网

套件,在学习蓝牙4.0 BLE时,以上最简配置已经足够,本书的所有程序开发也是基于这个最简配置的。



图 3-2 CC2540 蓝牙4.0低功耗最简配置开发板套件(2模块+2底板+1仿真器)

按说真正开发蓝牙4.0 BLE,最好硬件开发板也自己亲手做一个,本来 TI 就已经给出了详尽的硬件原理图和布线图,参考 TI 的设计,完全可以自己动手焊出一款属于自己的开发板。2011年的时候,如果找不到世嵌,我还真想这么干。但转念一想,这么做太浪费时间,除了核心板、底板,还有难缠的 CCDBG 仿真器要动手设计,也就买了。当然,如果有时间、有精力,又充满兴趣,我还真建议你们自己动手打造一块属于自己的开发板,你会感到其乐无穷!

3.2 世嵌最简配置开发套件

该套件的两个模块一样,都是 WB2540MVA;两个底板一样,都是 WX2530。之所以底板名取为 WX2530,主要因为这个底板其实就是与以前的 ZigBee 核心板配套的底板。总之,开发套件也就三样不同的东西:

- (1) 核心板 WB2540MVA;
- (2) 底板 WX2530;

(3) 仿真下载工具 CCDBG。

3.2.1 核心板 WB2540MVA

1. 模块介绍

以下来自世嵌对其核心板的介绍。

WB2540MVA 模块是世嵌科技推出的新一代蓝牙 4.0 低功耗标准无线收发模块,基于 TI CC2540 低功耗蓝牙 SoC 芯片,硬件部分彻底免除了客户射频开发的困难,数字接口部分全部引出,并扩展了模块存储空间,即使协议后续再更新,也有足够的存储空间可以使用。软件部分支持业界领先的蓝牙 4.0 低功耗协议栈 BleStack 软件包,方便客户开发出符合蓝牙 4.0 低功耗标准,或其他私有标准的产品。

模块采用 CC2540F256 芯片,配合 5dBi 天线,收发数据户外可视距离最高可达 200m。模块设计精巧,只有 30mm×24mm,而且引脚封装与 315/433MHz WS1110 模块以及 2.4GHz WZ2530 ZigBee 模块全面兼容,用户只需要一次性硬件设计,就可以通过选用不同的射频前端,推出兼容不同标准的无线通信产品。WB2540MVA 核心板如图 3-3 所示。

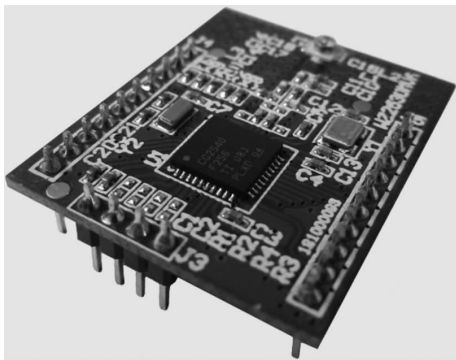


图 3-3 WB2540MVA 核心板

2. 模块特点

- (1) 工作频带: 2402~2480MHz。
- (2) 支持 Bluetooth 4.0 Low Energy 标准。
- (3) 支持 TI bleStack 协议栈软件包。
- (4) 主控芯片: CC2540F256。
- (5) 网络拓扑结构: 星形、树状。
- (6) 调制方式: GFSK。
- (7) 数据传输速率: 1Mb/s。
- (8) 天线模式: 外置天线。
- (9) 通信范围: 150~200m
- (10) 接收灵敏度: -97dBm。
- (11) 发射电流: 27mA。
- (12) 接收电流: 24mA。
- (13) 工作温度: -40~85℃。
- (14) 电源: 2.0~3.6V。
- (15) 模块外形尺寸: 30mm x 24mm。

3. 接口说明

通常开发人员更为关注的就是其接口定义了。接口说明包含接口电气定义和机械尺寸

描述。WB2540MVA 用户接口采用 3 组 2.00mm 单列直插式连接器,机械尺寸说明见图 3-4。3 组 2.00mm 单列直插式连接器对应的接口电气定义见表 3-1~表 3-3,核心板将所有好用的 CC2540 的 IO 都引出来了,方便开发人员的开发。注意引脚编号都是从对应直插的丝印编号处开始。

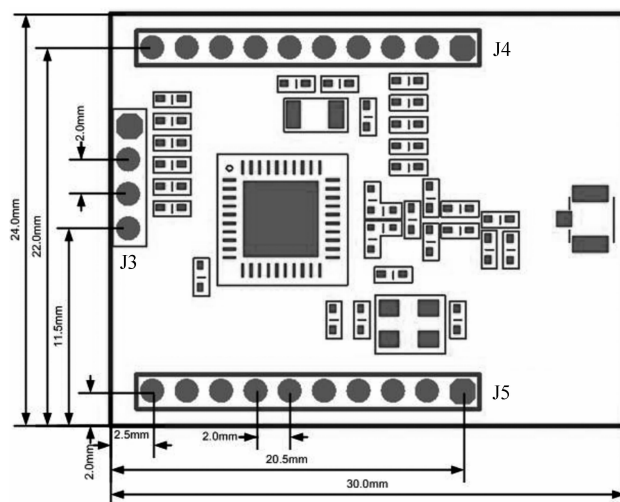


图 3-4 WB2540MVA 核心板机械尺寸和元件布局

表 3-1 核心板 J3 直插引脚定义

引脚编号	信号名称	CC2540 对应引脚	备注
1	VDD_3P3V	DVDD_USB	Power Supply, 3.3V DC
2	DN	USB_N	USB Data-
3	DP	USB_P	USB Data+
4	GND	DGND_USB	Power Ground

表 3-2 核心板 J4 直插引脚定义

引脚编号	信号名称	CC2540 对应引脚	备注
1	GND	GND	Power Ground
2	VDD_3P3V	VDD_3P3V	Power Supply, 3.3V DC
3	P2_2	P2_2	Debug Clock
4	P2_1	P2_1	Debug Data
5	P1_4	P1_4	GPIO, Input/Output
6	P1_5	P1_5	GPIO, Input/Output
7	NRESET	NRESET	Chip reset, active low.
8	P1_6	P1_6	GPIO, Input/Output
9	P2_0	P2_0	GPIO, Input/Output
10	P1_7	P1_7	GPIO, Input/Output

表 3-3 核心板 J5 直插引脚定义

引脚编号	信号名称	CC2540 对应引脚	备 注
1	P0_7	P0_7	GPIO, Input/Output
2	P0_6	P0_6	GPIO, Input/Output
3	P0_5	P0_5	GPIO, Input/Output
4	P0_4	P0_4	GPIO, Input/Output
5	P0_3	P0_3	GPIO, Input/Output
6	P0_2	P0_2	GPIO, Input/Output
7	P0_1	P0_1	GPIO, Input/Output
8	P0_0	P0_0	GPIO, Input/Output
9	P1_0	P1_0	GPIO, Input/Output
10	P1_1	P1_1	GPIO, Input/Output

3.2.2 底板 WX2530

大家或许对这个底板名字感到很奇怪,为什么不是 WX2540 呢? 世嵌在做蓝牙之前应该是做 ZigBee 的,所以开发了针对 CC2530 的底板,而 CC2540 与 CC2530 的引脚兼容,实际上两块芯片的引脚定义一模一样! 其核心板接口也是一模一样,所以 WX2530 底板也就直接用到 CC2540 的底板来了。由此也不得不佩服 TI 对产品的兼容性做得很好! 一块 ZigBee 的板子换上 CC2540,再烧写相应程序,就能直接做蓝牙应用了。这无疑有利于产品的升级和推广。当然 TI 还不止这点能耐,其软件的兼容性也做得非常强悍! 在今后学习蓝牙协议栈的开发中将会看到。

因为绝大多数实验都要用到底板的各功能模块,下面详细介绍底板功能。

1. 模块介绍

以下来自世嵌对其底板的介绍,从中可知该底板来自世嵌对 TI 早期无线产品的开发。虽然里面没有涉及 CC2540,但实际上是完全兼容 CC2540。

WX2530 底板是世嵌科技为其 CC1110、CC2530 和 CC2531 智能收发模块所专门开发出的通用传感器板,可以配合不同的无线收发模块,实现对不同频段(覆盖 315MHz、433MHz、2.4GHz 频点)和不同通信协议标准(包括 SimplicTI、IEEE 802.15.4、ZigBee 2006/2007/2007Pro 等)的支持。

该传感器板可以通过 USB 接口,或者直流电池接口供电,板上集成了多路传感器、4 个功能按键、4 个状态指示灯,用户可以使用此传感器板实现完整的终端节点功能。

如图 3-5 所示,WX2530 底板+CC2540 核心板,其中白色虚线框住的是 CC2540 核心板。

2. 模块特点

- (1) 支持世嵌科技 CC1110、CC2530 或 CC2531 模块。
- (2) 可与世嵌科技仿真器和协议分析仪配合使用。
- (3) 硬件组成:

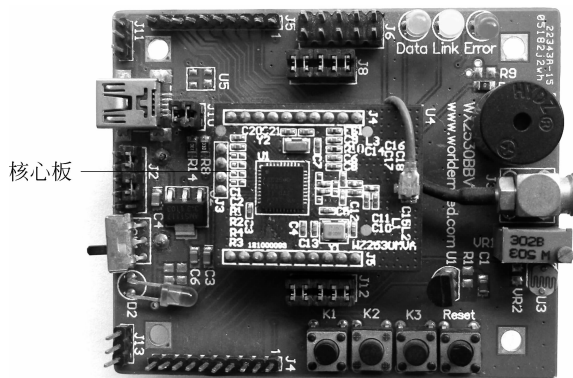


图 3-5 WX2530 底板(绿色)+CC2540 核心板(蓝色)

- ① USB 接口供电。
 - ② 4 个 LED 指示灯。
 - ③ 2 节 5 号电池供电单元。
 - ④ 支持蜂鸣器报警提醒。
 - ⑤ 仿真器接口供电。
 - ⑥ 一个高精度温度传感器。
 - ⑦ 电源开关及指示灯。
 - ⑧ 一个高精度光照传感器。
 - ⑨ 4 个功能按键。
- (4) 软件特性：
- ① 远程维护管理。
 - ② 兼容 TI SmartRF 系列编程, 仿真工具及协议分析软件。
- (5) 外观尺寸: 68mm×56mm。

3. 开发板整体架构

整个底板+核心板的功能模块设计架构如图 3-6 所示。

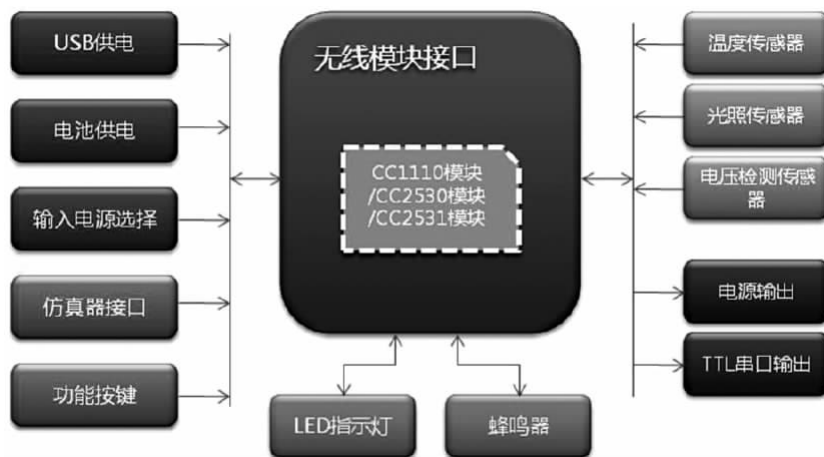


图 3-6 开发板整体架构

4. 模块输入接口

CC1110、CC2530、CC2531、CC2540 模块从核心板输入的接口如图 3-7 所示。

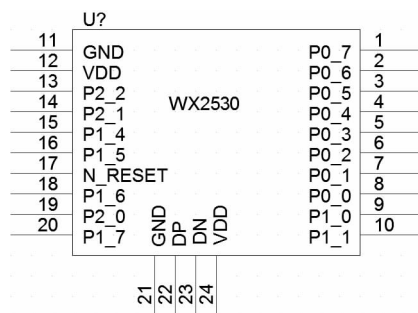


图 3-7 WX2530 底板输入接口

5. 各功能模块接口说明

需注意的是,表 3-4~表 3-6 中的绝大部分常用的跳线设置,产品出厂前世嵌已经设置好了,并不需要配置。

(1) LED 接口见表 3-4。

表 3-4 LED 接口

LED 编号	LED 颜色	信号名称	备注
D0	红色	VDD_3.3V	电源 3.3V
D1 (Data)	绿色	P1_0	输出低电平点亮
D2 (Link)	黄色	P1_1	输出低电平点亮
D3 (Error)	红色	P2_0	输出低电平点亮

(2) 按键接口见表 3-5。

表 3-5 按键接口

按键编号	信号名称	备注
K1	P0_0	需短路 J12 上 9 和 10 脚,按下对应输入低电平
K2	P0_1	需短路 J12 上 7 和 8 脚,按下对应输入低电平
K3	P0_4	
Reset	NRESET	按键触发节点复位,按下对应输入低电平

(3) 传感器接口见表 3-6。

(4) USB。现在无论是 PC 还是手机等移动设备几乎统统都集成了 USB 口,USB 接口的产品可谓琳琅满目。CC2540 也集成了 USB 接口,要使用该接口,先拔出核心板,需要设置 3 根跳线。注意按照图 3-8 中跳线方向,不可搞错。本跳线设置,我遍查世嵌手册未得,自己摸索的,实践证明是成功的。

表 3-6 传感器接口

传感器名称	信号名称	备注
温度传感器	P0_6	短路 J12 上 3 和 4 脚打开温度检测
光敏传感器/电压检测	P0_7	短路 J12 上 1 和 2 脚打开光敏/电压检测
蜂鸣器	P0_5	短路 J12 上 5 和 6 脚打开蜂鸣器报警提醒,输出高电平蜂鸣器响

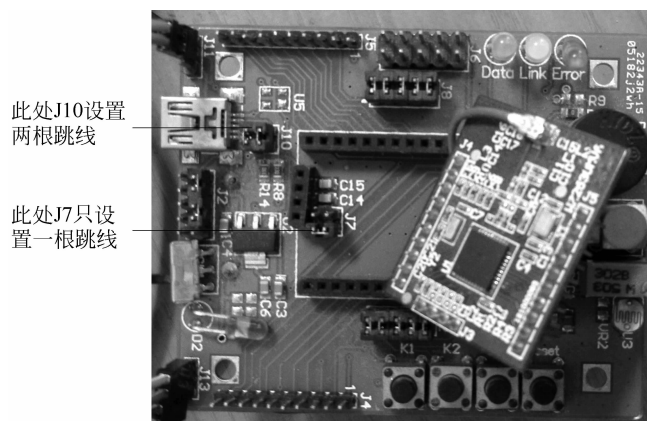


图 3-8 跳线使能 USB 接口

(5) UART 接口见表 3-7。

(6) 电源接口。底板也能向外提供或者由外部提供电源。接口如表 3-8 所示。注意与接口配置的电源,其电压大小和方向要匹配;否则后果不堪设想!

表 3-7 UART 接口(J13)

接口编号	信号名称	电气特征	备注
1	P0_2(RXD)	TTL 电平	J13
2	GND		J13
3	P0_3(TXD)	TTL 电平	J13

表 3-8 电源接口(J11)

接口编号	信号名称	备注
1	VDD_3.3V	与模块的 3.3V 电压相通,J11
2	GND	电源参考地,J11
3	VDD_5V	与 USB 接口的 VBUS 相通,J11

(7) J4、J5 接口。底板将所有核心板接口引出,J3 通过跳线设置(图 3-8)连接到 MiniUSB,其中底板 J4 的各引脚一一对应核心板 J5(表 3-9),底板 J5 的各引脚一一对应核心板的 J4(表 3-10)。

表 3-9 底板 J4 直插引脚定义

引脚编号	信号名称	CC2540 对应引脚	备注
1	P0_7	P0_7	GPIO, Input/Output
2	P0_6	P0_6	GPIO, Input/Output
3	P0_5	P0_5	GPIO, Input/Output
4	P0_4	P0_4	GPIO, Input/Output

续表

引脚编号	信号名称	CC2540 对应引脚	备 注
5	P0_3	P0_3	GPIO, Input/Output
6	P0_2	P0_2	GPIO, Input/Output
7	P0_1	P0_1	GPIO, Input/Output
8	P0_0	P0_0	GPIO, Input/Output
9	P1_0	P1_0	GPIO, Input/Output
10	P1_1	P1_1	GPIO, Input/Output

表 3-10 底板 J5 直插引脚定义

引脚编号	信号名称	CC2540 对应引脚	备 注
1	GND	GND	Power Ground
2	VDD_3P3V	VDD_3P3V	Power Supply, 3.3V DC
3	P2_2	P2_2	Debug Clock
4	P2_1	P2_1	Debug Data
5	P1_4	P1_4	GPIO, Input/Output
6	P1_5	P1_5	GPIO, Input/Output
7	NRESET	NRESET	Chip reset, active low.
8	P1_6	P1_6	GPIO, Input/Output
9	P2_0	P2_0	GPIO, Input/Output
10	P1_7	P1_7	GPIO, Input/Output

(8) 仿真器接口。图 3-9 为底板仿真器接口的封装。理论上应该采用该简易牛角座封装,但由于底板面积有限,元件太多,所以世嵌没有做牛角座,这也容易使仿真器灰排线插反。在这里要注意封装图中开口方向在底板上是朝外的。所以仿真器灰排的母座也要朝外插在底板上,如图 3-10 所示。

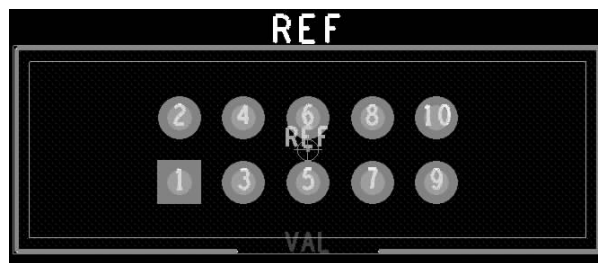


图 3-9 仿真器接口封装

对应的接口定义如表 3-11 所示。

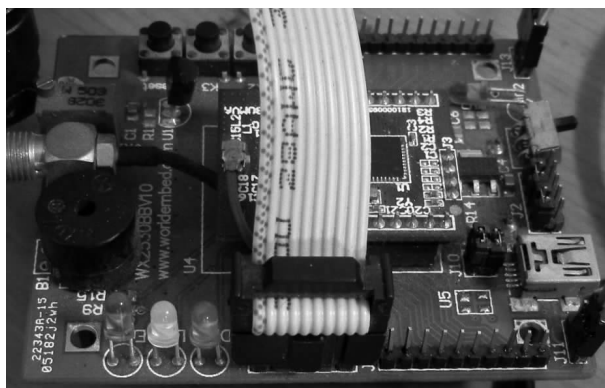


图 3-10 仿真器灰排母座应朝板子外侧插入仿真接口

表 3-11 底板仿真接口定义

引脚编号	信号名称	CC2540 对应引脚	备注
1	GND	GND	Power Ground
2	VDD_3P3V	VDD_3P3V	Power Supply, 3.3V DC
3	DC	P2_2	Debug Clock
4	DD	P2_1	Debug Data
5	P1_4	P1_4	GPIO, Input/Output
6	P1_5	P1_5	GPIO, Input/Output
7	NRESET	NRESET	Chip reset, active low
8	P1_6	P1_6	GPIO, Input/Output
9	VCC	VCC	VCC Power from target
10	P1_7	P1_7	GPIO, Input/Output

3.2.3 仿真器 CCDBG

要对 CC2540 编程和调试,仿真器必不可少,世嵌提供了开发板的配套仿真工具——CCDBG。其实仿真器的电路原理 TI 也给出了,有兴趣的朋友完全可以自己做一个出来。

1. 仿真器简介

CCDBG 仿真器是世嵌科技推出的第二代仿真器开发工具,支持 TI 公司 CC 系列低功耗无线收发器和片上系统芯片。软件部分可以配合 TI-MAC、SimpliciTI、Z-Stack 2.3.0、RemoTI、BleStack 等软件包,实现断点、单步等联机调试功能,并可以通过 Flash Programmer 下载符合不同 RF SoC 要求的固件程序。还可以配合世嵌科技 CC1110、CC2430、CC2530、CC2531、CC2540 等无线模块,实现完整的针对 315/433MHz 和 2.4GHz 频段,不同网络标准及协议(SimpliciTI、RemoTI、IEEE 802.15.4 和 ZigBee 2004/2006/2007 以及蓝牙 4.0 BLE 标准)的支持,是用户开发 WSN 和 ZigBee 应用的必备助手。

整个 CCDBG 仿真器非常轻便小巧,这样可以极大方便客户内部测试和部署现场调试。CCDBG 仿真器如图 3-11 所示。仿真器支持 TI 原装仿真器提供的标准 2×5 排针仿真器接口,可以和基于 CC1101、CC1110、CC1111、CC2430、CC2431、CC2510、CC2511、CC2530、CC2531、CC2540 等 CC 系列无线芯片的模块和底板无缝对接,让客户非常容易就可以使用 TI 公司 CC 系列高性能无线收发产品。



图 3-11 CCDBG 仿真器(上面虽未标明支持 CC2540,但实际完全支持)

2. 仿真器特点

- (1) 支持 TI 公司 CC1110、CC1111 等 1GHz 以下无线收发器 SoC 芯片。
- (2) 支持 TI 公司 CC2430、CC2530、CC2531、CC2533 等 2.4GHz ZigBee 无线收发 SoC 芯片。
- (3) 支持 TI 公司 CC2540 低功耗蓝牙 SoC 芯片。
- (4) 支持 TI 公司 CC2510、CC2511 等 2.4GHz 无线收发 SoC 芯片。
- (5) 支持 TI 公司 Flash Programmer 固件编程软件。
- (6) 支持 TI 公司 Packet Sniffer 无线协议分析仪软件。
- (7) 支持 TI 公司 SmartRF Studio 无线通信配置管理软件。
- (8) 支持 TI 公司 IEEE Address Programmer 无线编址软件。
- (9) 适用于基于 TI-MAC 开发符合 IEEE 802.15.4 无线个人局域网协议的应用。
- (10) 适用于基于 TI ZStack ZigBee 协议栈开发符合 ZigBee 2004、ZigBee2006、ZigBee 2007/Pro 标准协议应用。
- (11) 支持 SimpliciTI 简单无线组网协议。
- (12) 支持 IEEE 802.15.4 TI-MAC、ZigBee 2004/2006/2007、ZigBee Pro 和 ZigBee RF4CE 等标准协议。
- (13) 支持 TI TI-MAC、SimpliciTI 以及 zStack 等软件套件。
- (14) 支持 TI bleStack 蓝牙 4.0 低功耗协议栈软件。
- (15) 仿真器接口：
 - ① USB 接口用于连接 PC,即使不带串口的笔记本电脑也可以使用。
 - ② 兼容 TI 原装仿真器的 2×5 2.54mm 排线接口。
- (16) 支持的 MCU 集成开发环境 IDE 软件：

① Keil C51Development Tools。

② IAR Embedded Workbench for MCS-51 7.20~7.60。

(17) 自动速度识别功能,下载速度高达150kb/s。

(18) 提供 Windows 系统驱动,可以支持最新的 Windows 7 64 位操作系统,一次安装后就可完全即插即用。

(19) 使用 USB 供电,无须再外接直流电源。

(20) 仿真器外观尺寸:70mm×45mm×18.6mm(长×宽×高)。

可以看出该仿真器支持多种 CC 系列产品,包括多种无线产品。实践表明,该仿真器同样能在 IAR Embedded Workbench for MCS-51 8.3 开发环境中正常工作。

仿真器的接口定义见前面表 3-11。

3.3 搭建实验硬件环境

首先我计算机的 CPU 是 I5-2310,内存是 16G,操作系统是 64 位 Windows 7。如果硬件驱动能在我的 64 位 Windows 7 上安装,则在一般的 Windows XP 或者 32 位 Windows 7 上一般也能装。

1. 焊接开发板对外的接口

这些接口都是 2.00mm 间距的直插接口,开发板上留的都是通孔,并未焊上单排针。最好将这些排针接口都焊上排针,以方便杜邦线引出。这些排针接口包括 J4、J5、J11、J13,都是 2.00mm 间距的,所以选用 2.00mm 的单排针将这些直插接口都焊上,焊接技术是基本功,网上有的是教程,这个就不多说了。两块板子都要焊好,然后用杜邦线引出这些接口,说到杜邦线,这里用的可不是一般的 2.54mm 的杜邦线,需要用 2mm 转 2.54mm 的杜邦线,这个淘宝上到处都有,这样做好,就可跟开发板外的各种模块连接了。焊好的排针可参见图 3-10。

2. 搞定串口

做蓝牙开发,至少用到两个开发板,也至少需要两个串口来显示调试信息,虽然有 CCDBG 来调试,但是很多情况下不允许调试中断,尤其在蓝牙协议栈的多任务开发时,等你在调试的时候,很可能蓝牙连接就断了,有时就根本无法调试,这时,串口就能发挥其强大的作用了,通过串口在相应的程序处打印程序执行结果。可是呢,表 3-7 也表明,该 UART 口是 TTL 电平,要进行 RS232 通信,需要电平转换。好在现在这些转换芯片很多。由于需要两个开发板互相通信,也就需要两个串口显示调试信息,我的计算机的华硕主板上有一个串口,现在有串口的计算机不多哦,当初我就是看中这个才买的这款主板。可是这个串口不是标准的 DB9 串口,而且是 10 个针,费了很大的劲才找出了那极为重要的 3 根线,就是 RXD、TXD、GND 这 3 根线,什么 CTS、RTS 等流控制线就统统不管了,毕竟也不是在那种环境很恶劣、干扰很大或者线很长的情况下工作,3 根线传输足矣,事实也证明,3 根线做了很多实验,也没有碰到什么问题。除了这个串口,还需要一个串口,反正计算机的 USB 口很多,弄个 USB 转串口的模块就好了。

3. 第一个串口

就是那从计算机引出的 3 根电线,这个是标准的 RS232 电平,TTL 电平需要转换,我们的 TTL 电平是 3.3V 的,一般单片机都用的是 MAX232,是 TTL 5V 的,笔者借助了一块废弃的神舟 STM32 开发板,这个开发板主控芯片 STM32 已经被拆下,但装不上,因为拆的时候 PCB 板的铜皮都被扯掉了,应该是铜皮不牢所致。除了这个主控芯片没有,其他还好好,正好用上它的 MAX3232,这个是正宗的 3.3V 电平转 RS232 电平。按照神舟的原理图找到对应的串口接口,将杜邦线连接两头,一头是 CC2540 底板的 J13,一头是神舟的 PD5、PD6、GND,最后注意两块板子要共地,总共就 3 根线! 要用该串口,注意要给神舟开发板 5V 供电,或者直接把开发板的 3.3V 电源接口引到神舟 3.3V 的 VCC 上。第一个串口就这么“复杂地”搞定。

4. 第二个串口

第二个串口就没那么麻烦了,直接找个当前最流行的 USB 转 TTL 串口模块。注意 CCDBG 也将占用一个 USB 口、USB 蓝牙应用又将占用一个 USB 口,如果计算机上的 USB 较多,那么你使用两个这种模块也无所谓。关于这个模块,我选的是 PL2303HX,淘宝价格是第一个 2.5 元,以后每个 3.2 元,见图 3-12。



图 3-12 某淘宝店推出的 PL2303HX 模块

虽然这个模块能提供 3.3V 和 5V 电源,但最好不要指望这个 3.3V 电源用在开发板上,经实测,3.3V 电源实测为 3.5V,这用在数据收发上没问题,因为 CC2540 支持最高电压 3.6V,但用于开发板供电就不合适了,因为很多芯片都是在 3.3V 附近正常工作的。但若用该模块引出的 5V 电源直接接开发板的 5V 接口也是可以的,开发板上有 AMS1117-3.3 稳压芯片。为简便起见,就只用这个接口的 3 个针,分别是 RXD、TXD、GND。

这个模块的驱动安装很简单,到商家指定的网站下载,然后按照说明书的说明,三步就能装好,过程详见说明书。

5. 安装 CCDBG 驱动

关于 CCDBG 的安装,虽然有安装说明书,但 64 位 Windows 7 能自动安装,插上后等会就自动装好了。如果你的系统不能自动安装,那就按照说明书一步步安装。

最终的外配硬件连接图见图 3-13。注意：J5 有杜邦线引出到诺基亚 5110 液晶显示屏，还有与 CCDBG 的连接、PL2303HX 串口模块的连接，由于目前尚未用到无线实验，天线已拆去。如果实验尚未用到液晶显示和串口，则这两个模块也可以不要。

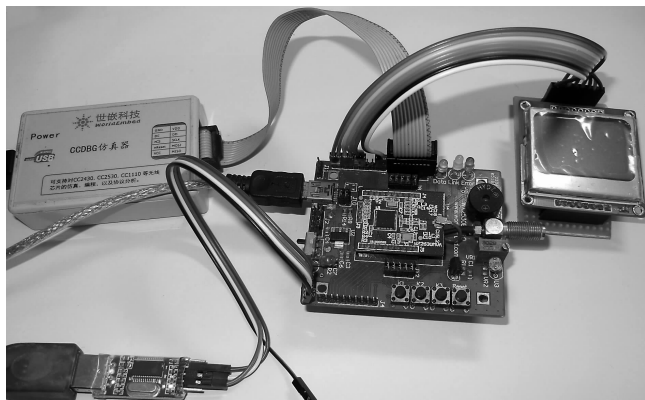


图 3-13 开发板与外配的硬件连接

现在，软硬件环境都搭好了，正所谓万事俱备，只欠东风。下面就让我们创建一个真正的工程，来正式开启我们的物联网工程实践之旅吧！

3.4 创建工程——点亮 LED

学习计算机语言，几乎都是从演示一个“hello world!”之类的程序开始，意思是“世界，我来啦！”，另一方面也大概表示你要进入这门语言的世界了。而从我开始学习单片机开始，单片机的第一个实验又从这个“hello world!”变成了点亮第一盏 LED 灯。于是也从“点灯”开始吧，对着蓝牙 4.0 BLE 的世界大喊一声，“蓝牙，我来也！”。

一般的教科书都是教我们从一个空工程开始，一步步按照要求配置，一旦少了哪个画面或者哪句话，可能就永远配置不好。也可以不参考教科书，就是再用 IAR 打开一个蓝牙协议栈的示例，按照范例一步步地配置，这样配置肯定不会有错。但当我真这么做时，发现太繁杂啦！要配置的窗口界面和参数太多啦！稍不留神就会配置出错。而且有些栏目还呈灰色，不能跟原工程匹配，不如“狸猫换太子”，直接用 TI 的范例配成我们自己的工程，这样所有繁杂的配置 TI 都帮我们做好了，仅仅是将工程改个名字就好了！

因为是本书的第一个项目，所以下将详细讲解我们“处女作”的诞生过程。

首先将安装目录下的蓝牙协议栈复制一个到你的工作目录下，如图 3-14 所示，这是已复制好的，这样就直接在蓝牙协议栈下开发，直接利用协议栈的各软件资源，如果你对协议栈的程序修改有问题，还可以回头看看安装目录下原来好的程序。

进入项目目录，如图 3-15 所示，这是协议栈的一些项目范例。

经过比较，发现较简单的项目是 SimpleBLEBroadcaster，复制该项目得一个副本，如图 3-16 所示。

将副本改个你喜欢的名字，如 mTest1，如图 3-17 所示。这就开始“狸猫换太子啦！”

然后进到项目文件目录，如图 3-18 所示。



图 3-14 复制安装目录下的蓝牙协议栈到你的工作目录



图 3-15 协议栈的项目范例



图 3-16 复制得一个副本



图 3-17 将项目改名



图 3-18 项目文件目录

继续改名,将 SimpleBLEBroadcaster.eww 改为 mTest.eww,将 SimpleBLEBroadcaster.ewp 改为 mTest.ewp,如图 3-19 所示。

这次要打开 mTest.eww 文件进行简单编辑了,随使用什么文本编辑器都可以,如记事本或者 NotePad 都行。如图 3-20 所示,图中光标处将原来的 SimpleBLEBroadcaster 改为 mTest 即可,这在打开该文件时,可自动找到 mTest.ewp 文件。

好了,再双击 mTest.eww 文件,我们创造的“新”工程诞生了!如图 3-21 所示。注意标题栏和工作区都是 mTest 字样。但这只是表面现象,还有几个小地方需要修改。先移除工程中原来的文件,除了 Output 文件夹外,其余文件夹全部选中并右击,弹出快捷菜单,选择 Remove 命令,移除这些文件夹,如图 3-22 所示。注意:只是从工程中移除,并不会真正删除原来的文件。

移除文件之后就是一个空工程了,这下干净了!继续右击 mTest-CC2540,在弹出的快捷菜单中选择 Options 命令,见图 3-23,进入 Gernal Options→Target 选项卡,在 Location for

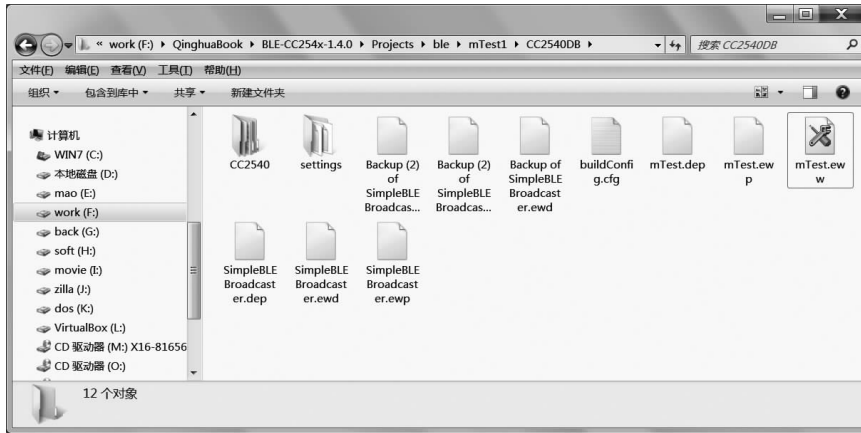


图 3-19 项目文件目录

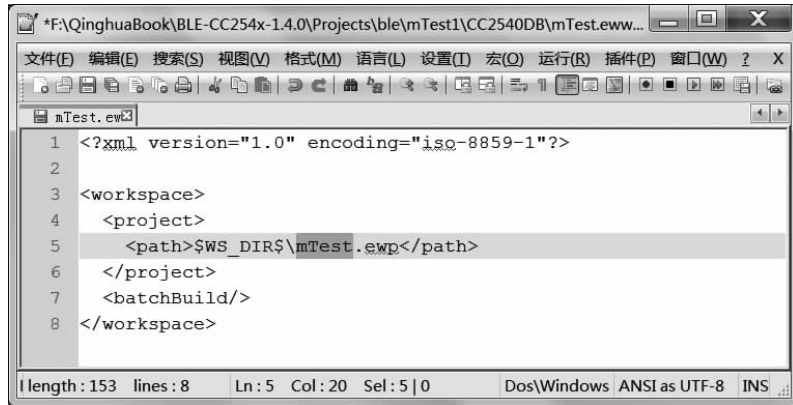


图 3-20 修改 mTest.eww 文件

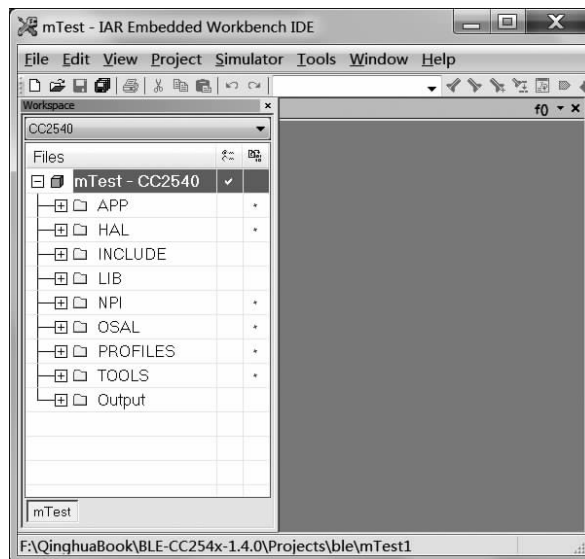


图 3-21 新工程诞生

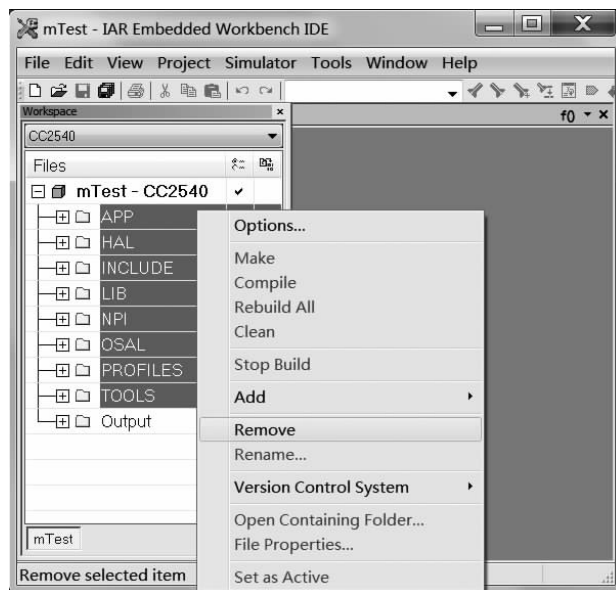


图 3-22 移除原工程文件

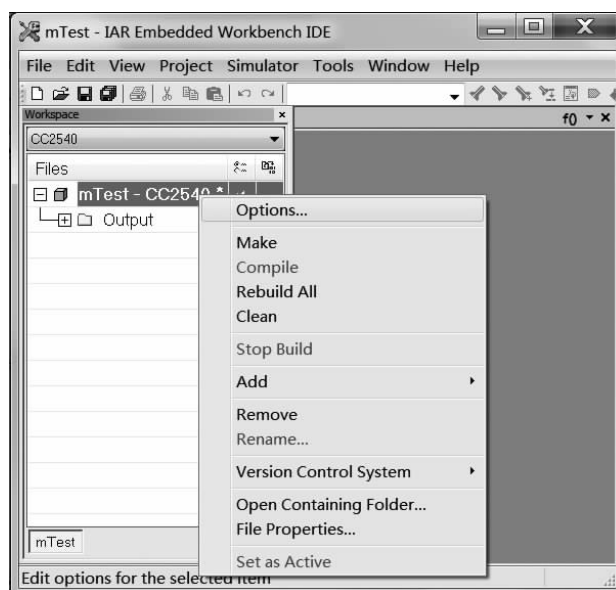


图 3-23 选择 Options 命令

constants and string 选项区域选中 RAM memory 单选按钮,如图 3-24 所示的红色矩形框框住的部分。原项目默认的设置是 ROM mapped as data,这个需要初始化 Flash 相关寄存器,否则需要加上 CODE 关键字定义常量和字符串,且字符串不能直接使用——需要先定义,存在诸多不便。对于简单项目,由于没有初始化 Flash 相关寄存器的代码,所以最好不要使用这个选项,而是将字符串置于 RAM memory,这样定义常量和字符串就无须加上相应的关键字了。

继续在 Options 对话框中进入 Linker → Extra Output 选项卡,将光标处的 SimpleBLEBroadcaster.hex 改为 mTest.hex,如图 3-25 所示。

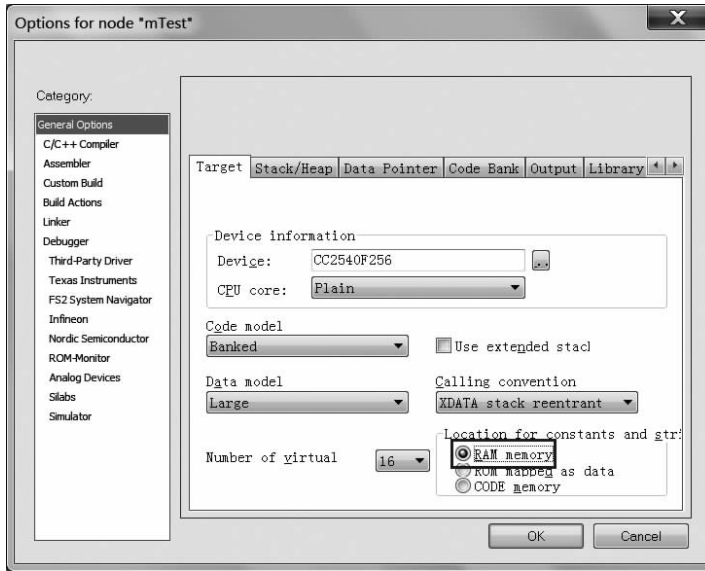


图 3-24 将常量和字符串置于 RAM memory

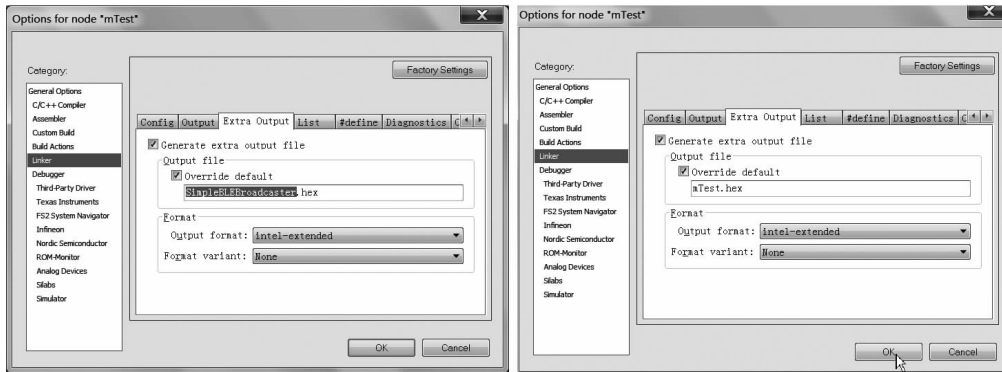


图 3-25 将输出的 hex 文件改为你喜欢的名字

继续选中 Debugger → Setup 选项卡，单击 Simulator 拉下列表框，选中 Texas Instruments 选项，如图 3-26 所示。这个改动非常重要，让你能使用 CCDBG 调试。

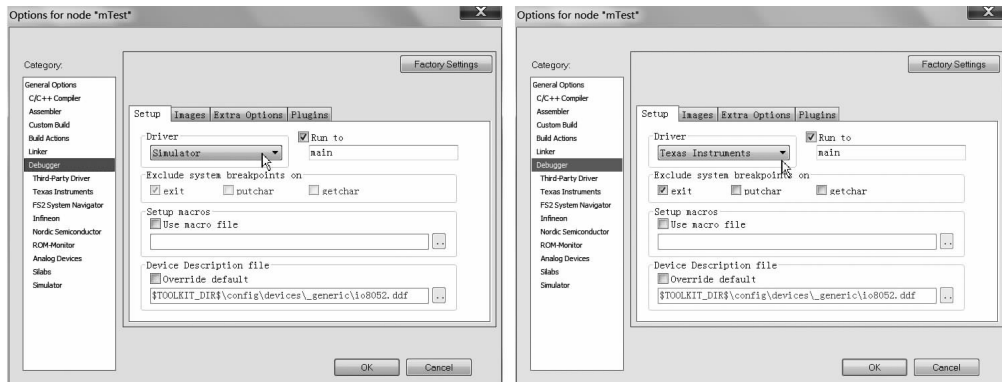


图 3-26 将 Driver 设置为 Texas Instruments

就修改这两个地方,其余按默认设置,复杂的选项就这么简单地搞定了。接下来要为新工程添加文件了。还是右击 mTest 工程,在弹出的快捷菜单中选择 Add→Add Files... 命令,如图 3-27 所示。然后出现 Add Files-mTest 对话框,还是找到这个项目的目录,找到 Source 文件夹,选中 3 个文件,如图 3-28 所示。

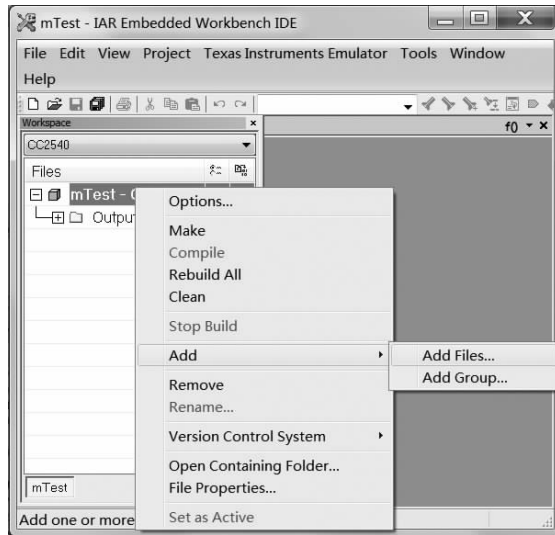


图 3-27 选择 Add→AddFiles 命令



图 3-28 选中 3 个文件

按 Delete 键,将 3 个文件删除,且将 SimpleBLEBroadcaster.c 改名为 mTest.c,如图 3-29 所示。

选中 mTest.c 文件,单击“打开”按钮,将该文件添加到工程中。如图 3-30 所示,mTest.c 文件已经添加进工程了,双击 mTest.c,编辑窗口出现这个文件的代码,按下 Ctrl+A 组合键,全部选中代码,然后按 Delete 键删除,如图 3-31 所示,编辑区已经空空如野,这下终于可编写代码,开始我们的新工程了!“狸猫换太子”成功!