GAINST_CC2430 用户手册

本手册内容包括 GAINST_CC2430 软件集成开发环境 IAR 介绍，Z-Stack 介绍，节点硬件平台介绍，仿真器使用方法，如何编译调试程序，如何新建并修改代码工程，如何使用 TI 提供的辅助软件，共 7 个部分。

第一章 软件集成开发环境 IAR 介绍

软件集成开发环境为 **IAR Embedded Workbench™ for 8051** ,具体版本是 EW8051-EV-720H，其基本特征如下：

支持的芯片

有 8051, 8052 和扩展结构的所有芯片。IAR 支持 CC1110/CC2510，他们拥有 4 个配置文件，对于 CC2510，这些文件已经随着 IAR Workbench 安装了，CC1110 的可以从网上下载，配置文件的位置如下表所示：

File name	Location	Description
CCxx10.i51	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config\derivatives\chipcon	Setup some parts of an IAR Project.
CCxx10.ddf	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config\derivatives\chipcon	Describe each register used in the debugger.
lnk51ew_ccxx10.xcl	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config	Set up the linker to reflect CCxx10.
ioCCxx10.h	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\inc	Header file for CCxx10.

IAR 也支持 CC2430，它的 5 个配置文件已随 IAR Workbench 安装，所在位置如下表所示：

File Name	Location	Description
CC2430.i51	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config\derivatives\chipcon	Setup some parts of an IAR Project.
CC2430.ddf	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config\derivatives\chipcon	Describe each register used in the debugger.
lnk51ew_cc2430.xcl	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config	Set up the linker to reflect CC2430.

lnk51ew_cc2430b.xcl	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\config	Set up the linker to reflect CC2430 with banked code model.
ioCC2430.h	C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\inc	Header file for CC2430.

集成开发环境(IDE)

一个标准的和扩展的 IDE 在 Windows 98/ME/NT4/2000/XP 下运行.

在一个无缝集成的开发环境里创建工程, 编辑文件, 编译, 汇编, 连接和调试你的应用.

在全部源文件,一组源文件和单个源文件的级别上都有可以选择的工具配置。

同一个工作区里可以有多个工程.

分层工程表示法显示了所有不同的源文件和输出文件并给出他们配置的总体概貌.

XML 架构工程文件.

多字节编辑器.

在构造过程中容易集成外部工具。

C 编译器

高度优化的 ISO/ANSI 标准 C 编译器.

完全支持经典的和扩展的 8051 结构, 像 Dallas DS80C400

普通的和高级的 8051 都可以有针对性的优化.

多种级别的代码大小和执行速度优化.

高效率的 32 比特与 IEEE 一致的浮点算术运算.

直接用 c 进行容易和快速的中断处理.

用户控制寄存器的使用以得到最佳的性能.

针对 8051 的特殊语言扩展,例如特殊函数类型, 扩展关键字, 以及完全的配置以适应嵌入式开发.看实例表格。

支持 DATA, IDATA, XDATA, PDATA 和 BDATA.

在编译器和库里支持多 DPTR.

特殊功能寄存器的比特寻址.

可以用尽 32 个虚拟寄存器.

高度优化重入代码原型使工程在不同的目标板上方便移植.

仿真和调试

有恢复功能的复杂代码和数据断点.

指令精确仿真执行

C 有参数调用堆栈.

即使在高度优化级别上，完全的支持堆栈展开.

函数调用级别的小颗粒单步.

终端端口,外围和中断仿真.

存储器配置和确认.

通用的监视 CPU/外围器件 寄存器, 结构, 调用链, 局部和全局变量.

函数级别的代码压缩和覆盖分析.
针对实时操作系统的调试.

软件的具体使用方法请参考“IAR IDE user Manual”文档。

第二章 Z-Stack 介绍

Z-Stack是TI提供的符合Zigbee规范的免费协议栈，完全可以运行在GAINST_CC2430节点上，利用Z-Stack，用户能够简单快速的开发出适合自己的Zigbee应用。

Application 设计

用户可以为每个 Application Object 创建一个 Task，下面是一些考虑：

一个 OSAL Task 对应 many Application Object

一对多设计的优点和缺点如下：
-优点：当收到一个高级任务事件(按 switch 或者 serial port)的处理较简单
-优点：节省多个 OSAL Task 结构体所需堆栈空间
-缺点：当收到一个发来的 AF message 或者 AF data confirmation 的处理较复杂，多路接收应用对象在负担在一个单独的用户任务中。

一个 OSAL Task 对应一个 Application Object

一对一设计的优点和缺点和一对多的正好相反
-优点：一个发来的 AF message 或者一个 AF data confirmation 已经被协议栈的较低层分成多路，所以接收的 Application Object 是计划中的接收。
-缺点：多个 OSAL Task 结构体所需堆栈空间较多
-缺点：如果两个或多个 Application Object 使用同一个资源，当接收到一个高级的任务事件处理较复杂。

强制性的办法

所有 OSAL Task 必须实行 2 种方法，一是执行 task 初始化，二是执行 task event。
执行初始化的回调函数一般都这样命名：“Application Name”_Init(例如 SampleApp_Init)
执行任务事件的回调函数一般这样命名：“Application

Name”_ProcessEvent(例如 SampleApp_ProcessEvent())。所有的 OSAL Task 都能定义包括 mandatory event(强制命令事件)在内的 15 个事件。

Mandatory Events(强制命令事件)

SYS_EVENT_MSG(0x8000)在设计 OSAL Task 时被保留。

SYS_EVENT_MSG(0x8000)

全局系统消息发送通过 SYS_EVENT_MSG，其定义在 ZComDef.h 文件中。此任务事件处理下面的系统消息。

AF_DATA_CONFIRM_CMD

这是一个针对每个 data request(AF_DataRequest())成功发起而 indecation 结果的任务。ZSuccess confirms 通过 OAT 成功发送 data request。

AF_INCOMING_MSG_CMD

这是针对一个 incoming 的 AF message 的 indication。

KEY_CHANGE

这是一个 key press 行为的 indication。

ADO_NEW_DSTADDR

这是一个匹配描述请求(Match Descriptor Request)response 的 indication。

ZDO_STATE_CHANGE

这是一个网络状态改变的 indication。

Z-Stack协议栈具体使用方法请参考“Z-Stack User’s Guide For CC2430ZDK/CC2431ZDK”文档。

第三章 节点硬件平台介绍

CC2430 芯片介绍

CC2430 芯片采用0.18 μm CMOS 工艺生产，工作时的电流损耗为27 mA；在接收和发射模式下，电流损耗分别低于27 mA 或25 mA。CC2430具备多种休眠模式和转换到主动模式时间超短的特性，特别适合那些要求长电池寿命的应用。

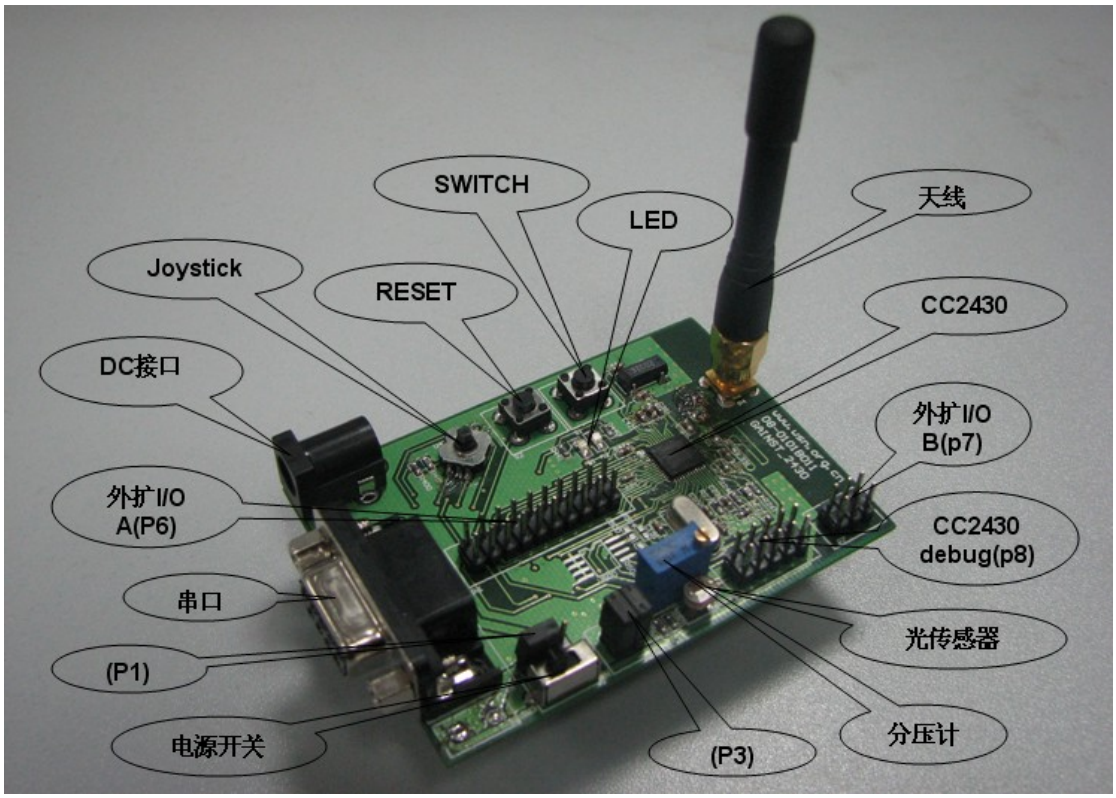
CC2430 芯片的主要特点如下：

- 高性能和低功耗的8051微控制器核。
- 集成符合IEEE802.15.4标准的2.4 GHz的RF无线电收发机。
- 优良的无线接收灵敏度和强大的抗干扰性。
- 在休眠模式时仅0.9μA的流耗，外部的中断或RTC 能唤醒系统;在待机模式时少于0.6μA的流耗，外部的中断能唤醒系统。
- 硬件支持CSMA/CA 功能。
- 较宽的电压范围（2.0～3.6 V）。
- 数字化的RSSI/LQI 支持和强大的DMA功能。
- 具有电池监测和温度感测功能。
- 集成了14 位模数转换的ADC。
- 集成AES 安全协处理器。
- 带有2路USART，以及1个符合IEEE 802.15.4 规范的MAC计时器，1个常规的16位计时器和2个8位计时器。
- 配合强大且灵活的开发工具。

GAINST_CC2430 节点介绍

GAINST_CC2430节点基于TI SOC芯片CC2430，在功耗、集成度、通信距离、功能及特性方面均很显著。

GAINST_CC2430节点各硬件组成部分如下图所示：



GAINST_CC2430 节点硬件组成部分介绍

- 板载光敏传感器，用于监测光强度
- 板载手柄方向键，用于外设输入控制
- 板载分压计，用于监测电压
- 提供开关控制2个，分别用于硬件复位和外设中断触发
- 提供LED指示灯2个，用于程序调试和节点状态指示
- 提供USB接口，用于flash 编程、在线调试
- 提供RS232接口一个，用于跟后台通信或者连接其他的串口设备
- 可以选择电池供电（两节AA电池）或DC直流供电
- 提供丰富的IO扩展口，方便功能扩展

GAINST_CC2430 节点功能部件介绍

P1接口介绍

Pin 1-2脚，跳线冒连上，电池供电

Pin 2-3脚，跳线冒连上，DC供电

P3接口介绍

Pin 1-2脚，跳线冒连上，正常供电；跳线冒不连，接电流计，可以测试电流

Pin 3-4脚，跳线冒连上，SW control 供电。

P8接口(仿真器接口)介绍

引脚	功能

1	GND
2	VCC
3	P2.2
4	P2.1
5	P1.4
6	P1.5
7	RESET
8	P1.6
9	空
10	P1.7

P6 I/O A扩展接口介绍

引脚	功能
1	VDD
2	VDD_SW_CONTROLLED1
3	P0_0/LDR
4	RESET_N
5	P0_1/BUTTON PUSH
6	P1_7/SO/MISO/UART_RD
7	P0_2/EE_SDA
8	P1_6/SI/MOSI/UART_TD
9	P0_3/EE_SCL
10	P1_5/SCLK/RTS
11	P2_1/DD
12	P1_4/CSN/SS/CTS
13	P2_2/DC
14	P2_0/JOY PUSH
15	P0_6/JOY
16	P1_2/VDD_SW_CTRL
17	P0_7/POT
18	P1_1/LED2
19	P1_0/LED1
20	GND

P7 I/O B扩展接口介绍

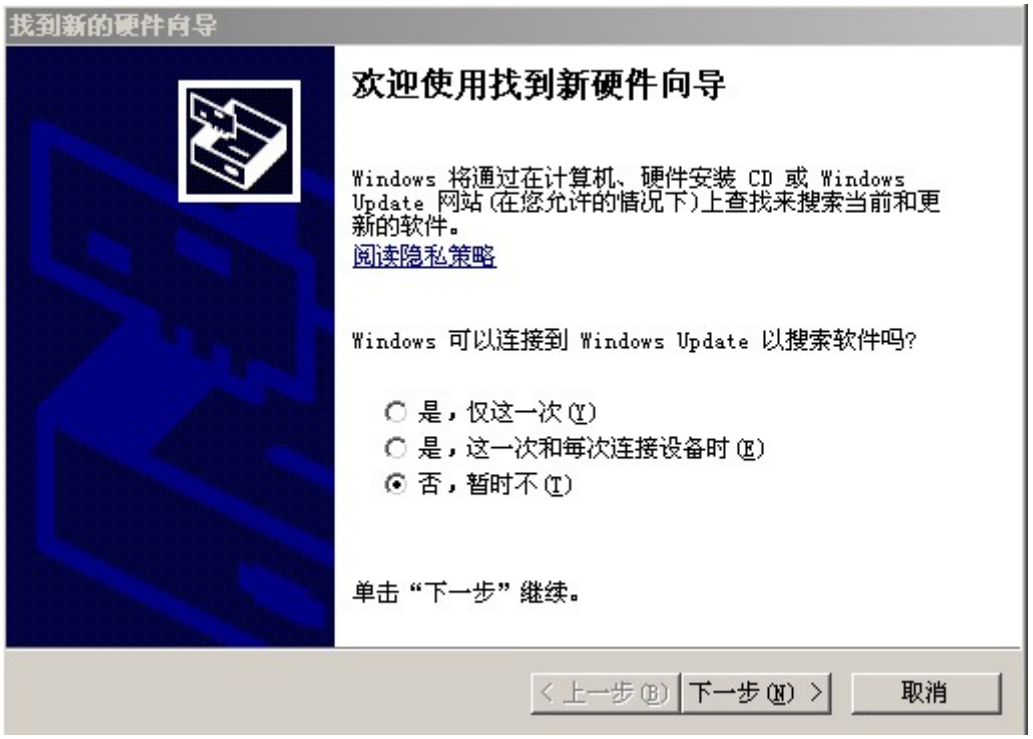
引脚	功能
1	P1_3/GPIO
2	DC_JACK_PWR
3	P0_4
4	VDD
5	P0_5
6	GND

GAINST_CC2430 开发板的原理图见光盘

第四章 仿真器使用方法

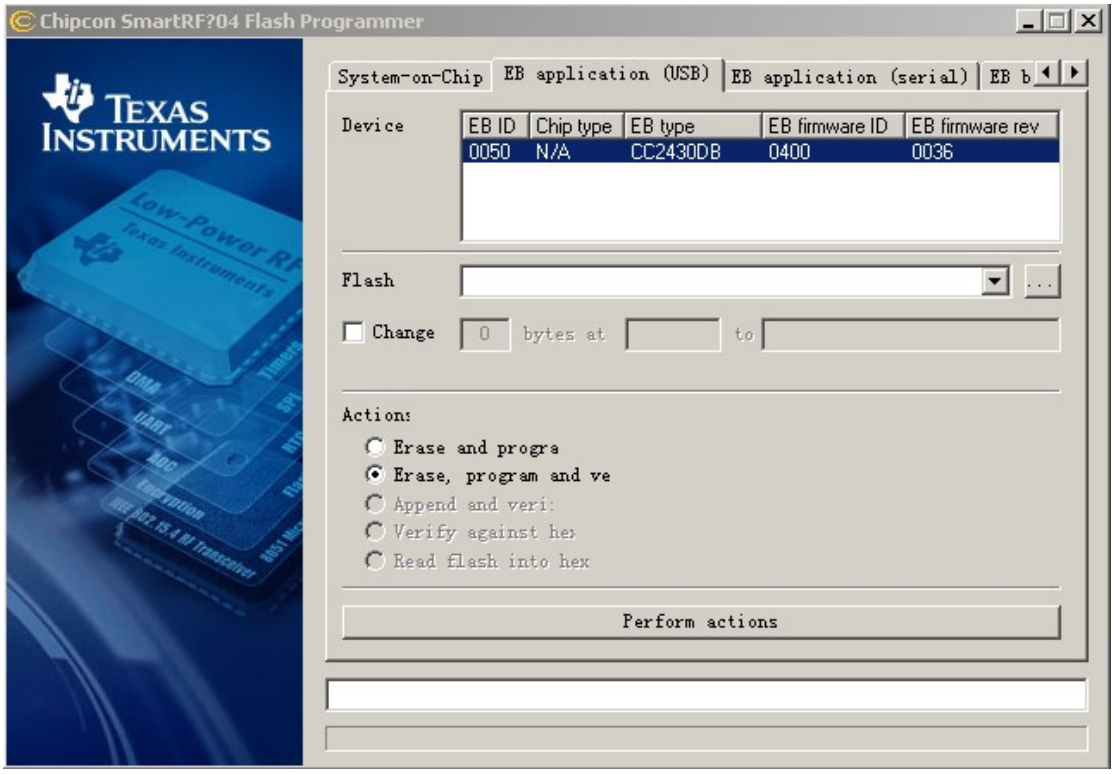
GAINST_CC2430 烧写 flash 和调试代码都需要使用仿真器。

将仿真器 USB 端口与 PC 连接，弹出发现新硬件的提示，选择“否，暂时不”，点击“下一步”，如下图所示：

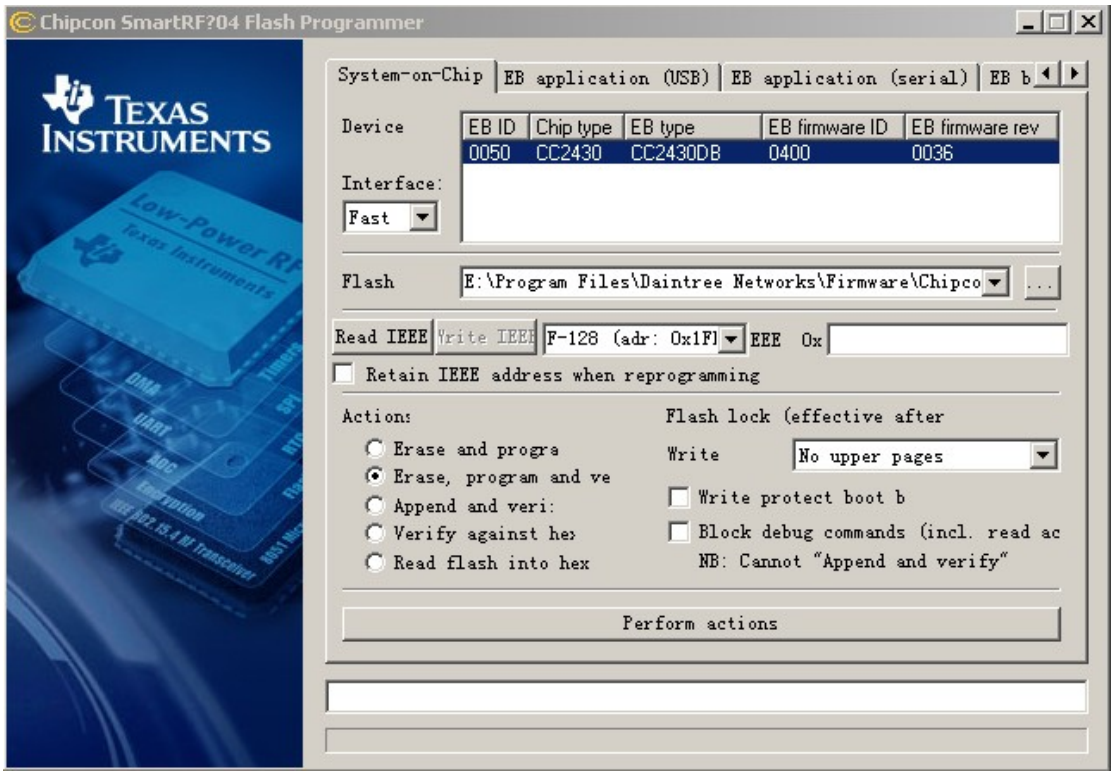


若之前安装过 Z-Stack，则可以识别出硬件型号，选择“自动安装软件(推荐)”，点击“下一步”，如下图所示，则可完成仿真器驱动安装。

当安装完仿真器驱动后，在 TI 提供的辅助软件-“SmartRF04 Flash Programmer”的“EB application(USB)”栏可以发现此设备，如下图所示：



将仿真器另一端的接口与 GAINST 节点 P8 连接，连接无误后在“SmartRF04 Flash Programmer”软件的“System-on-Chip”栏可以发现节点设备，如下图所示：



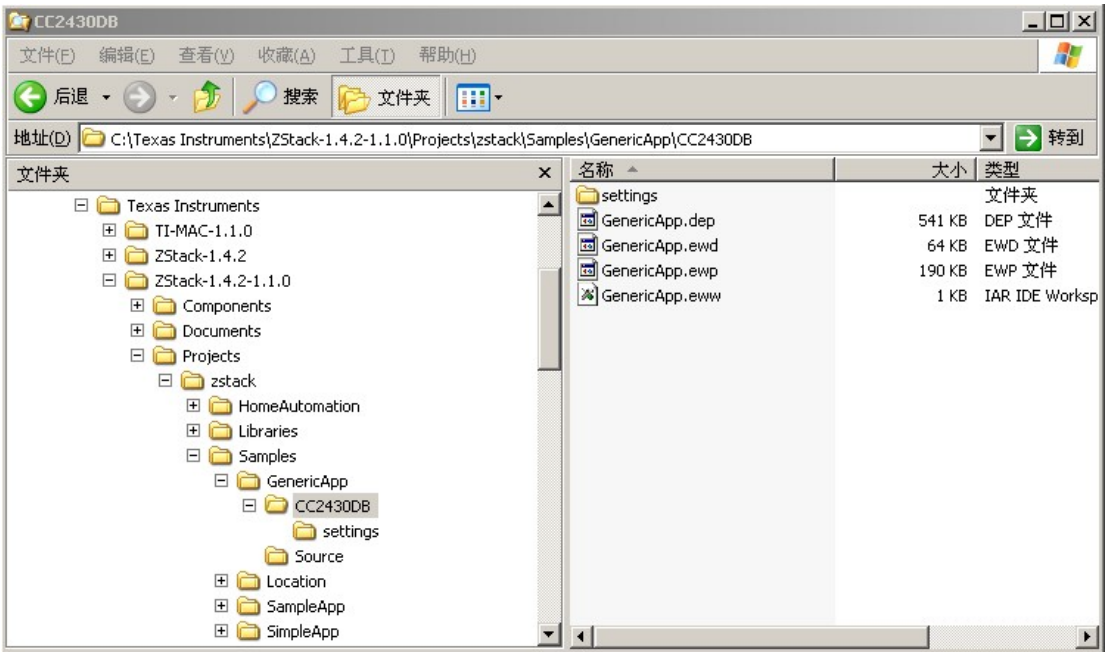
软件的具体使用方法请参考参考“Flash Programmer User Manual ”文档。

第五章 如何编译调试程序

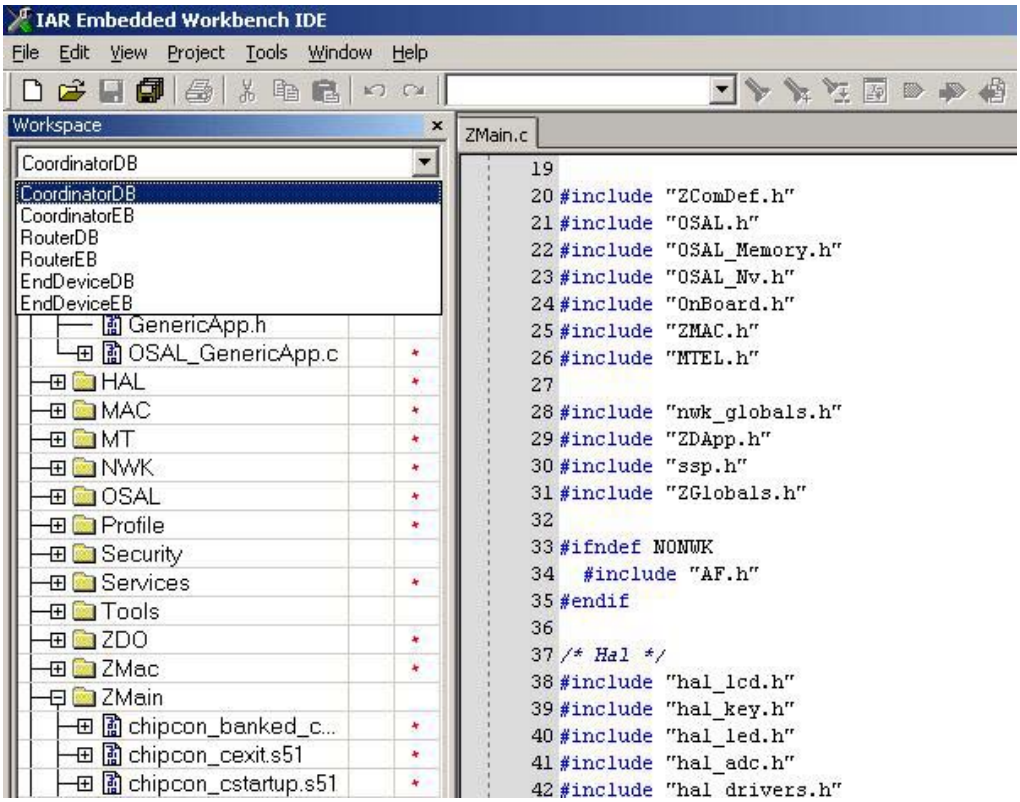
安装完本文档提到的 IAR 集成开发环境和 Z-Stack 软件，且仿真器工作正常后，便可以开始对代码工程的操作。在安装完 Z-Stack 后，在“C:\Texas Instruments\ZStack-1.4.2-1.1.0\Projects\zstack”目录下可以发现许多工程文件目录，现在以“Samples\GenericApp\CC2430DB”目录下的“GenericApp”为例子进行讲解。

1，通过仿真器将PC和节点连接，若需要安装驱动，则到“C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\drivers\chipcon”目录下查找相应文件。

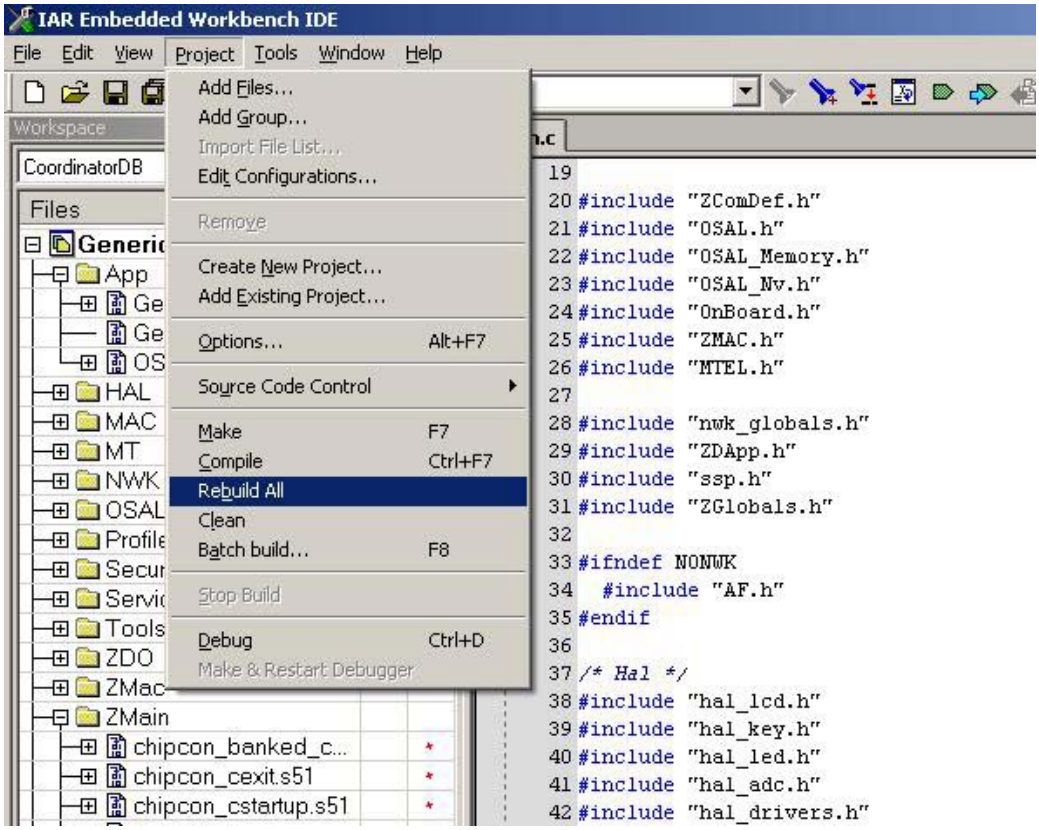
2，打开“GenericApp.eww”工程，如下图所示：



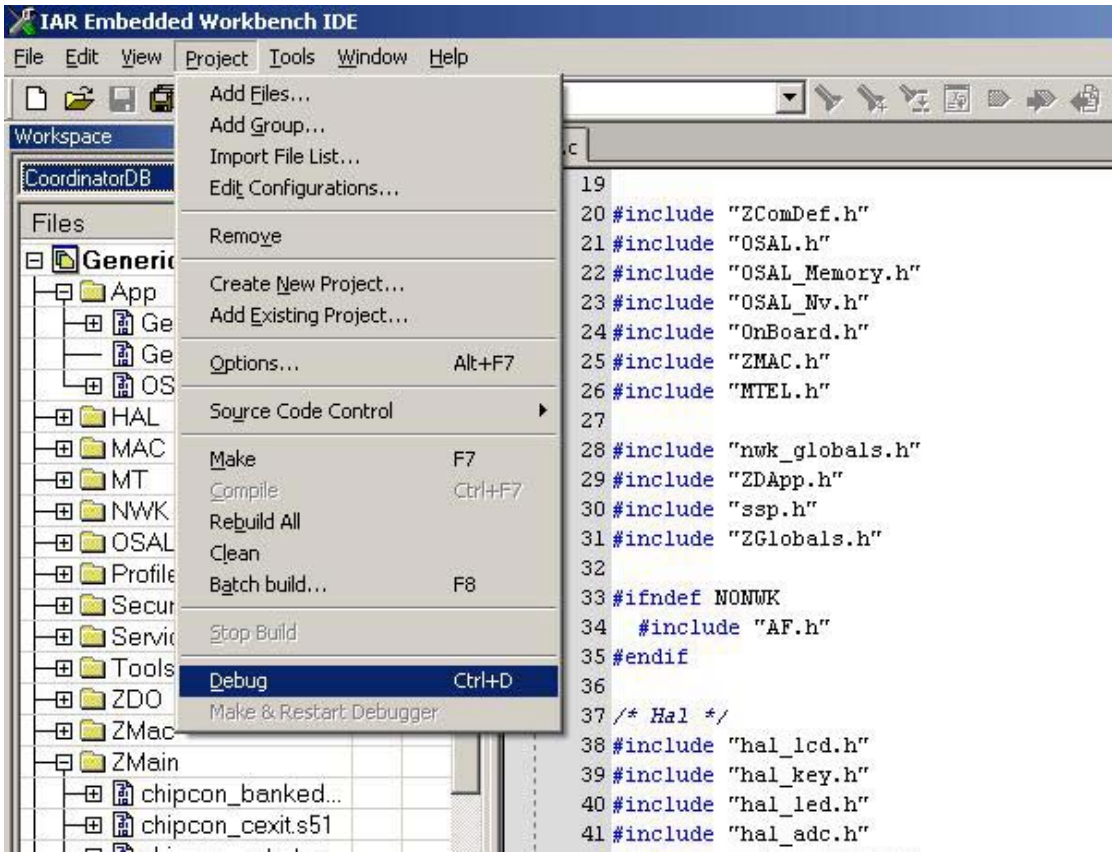
3，在 Workspace 下拉框里选择“CoordinatorDB”，



4，点击“Project”菜单，选择“Rebuild All”，编译后可生成 Zigbee 网络中必需的 Coordinator 设备文件，如下图所示：

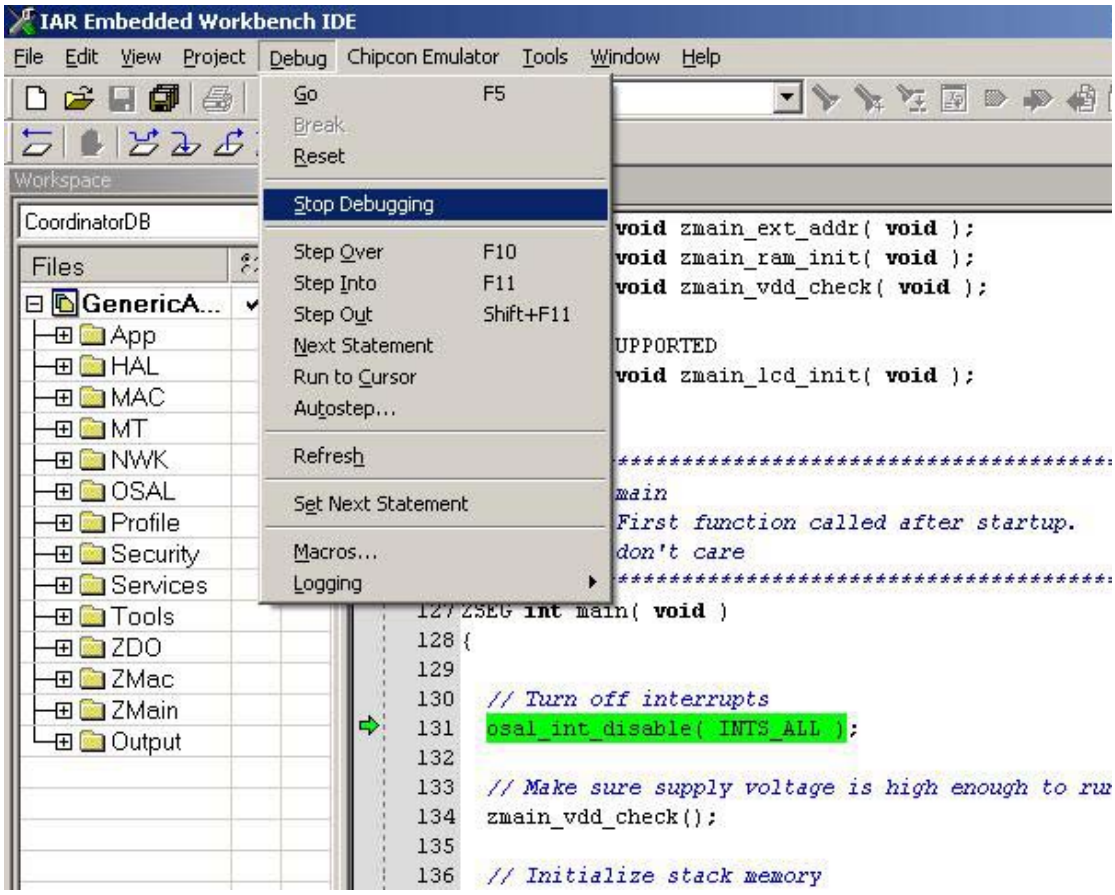


5，点击“Project”菜单，选择“Debug”，可将应用程序下载到节点设备中，如下图所示：



6，当下载过程结束后，通过点击“Debug”菜单，选择“Stop Debugging”退出

Debug 过程，如下图所示，若不点击“Stop Debugging”，则继续保持调试状态。



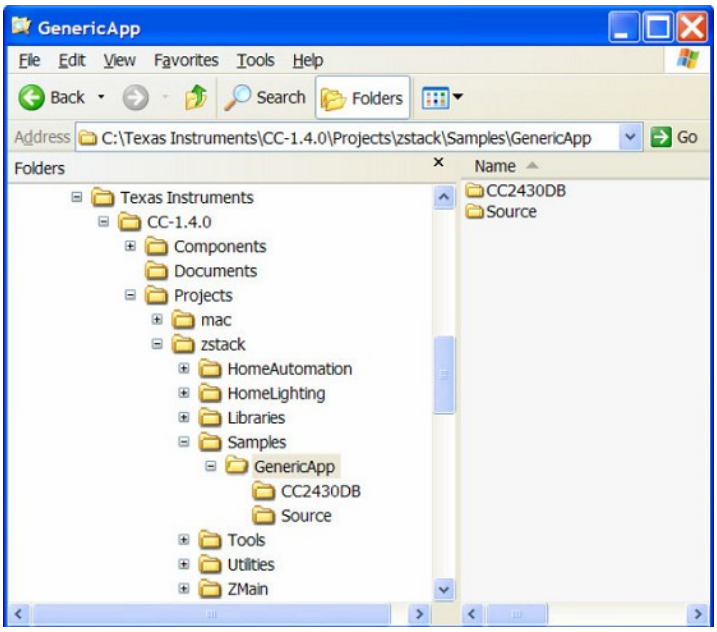
7，断开仿真器和节点的连接，重复上面 3-6 步，选择 EndDeviceDB，可以生成 Zigbee 网络中的 Enddevice 设备应用程序。

当 2 个应用程序下载到不同的节点中后，先后给 Coordinator 和 Enddevice 上电，若发现 D1 不停闪烁，则说明此设备 IEEE 地址为全 F，需要重新写入 IEEE 地址，此时可以通过“SmartRF04 Flash Programmer”软件写入地址，(注意：当 IEEE 地址不为全 F 时，是不能写入其他地址的)，或者向下(Center)按节点上的导航键，可以为设备分配一个临时的 IEEE 地址。当 LED2 亮起，表示设备开始正常运行。若 Enddevice 的 LED2 闪烁表示 Coordinator 和 Enddevice 超出通信范围。

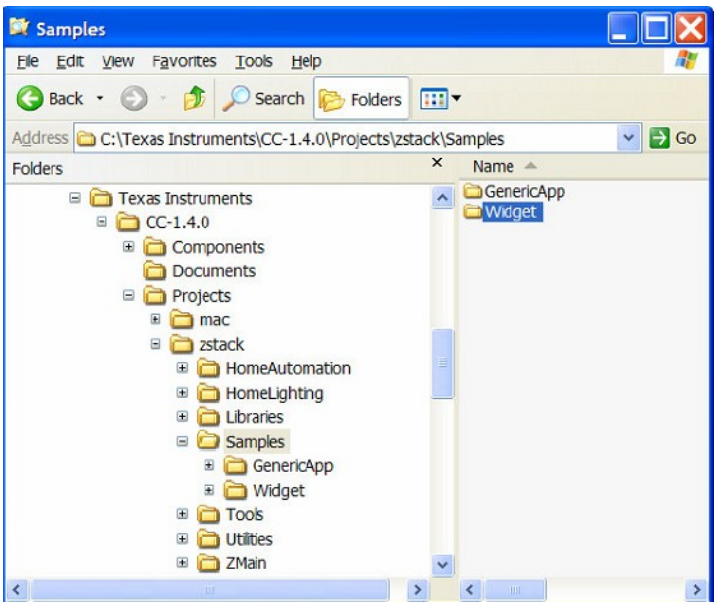
第六章 如何新建并修改代码工程

新建工程可以在 IAR 中进行，但添加源文件或者组比较麻烦，所以推荐修改现有工程而达到新建工程的目的，下面是根据 GenericApp 工程修改而得到新工程 Widget 的例子。

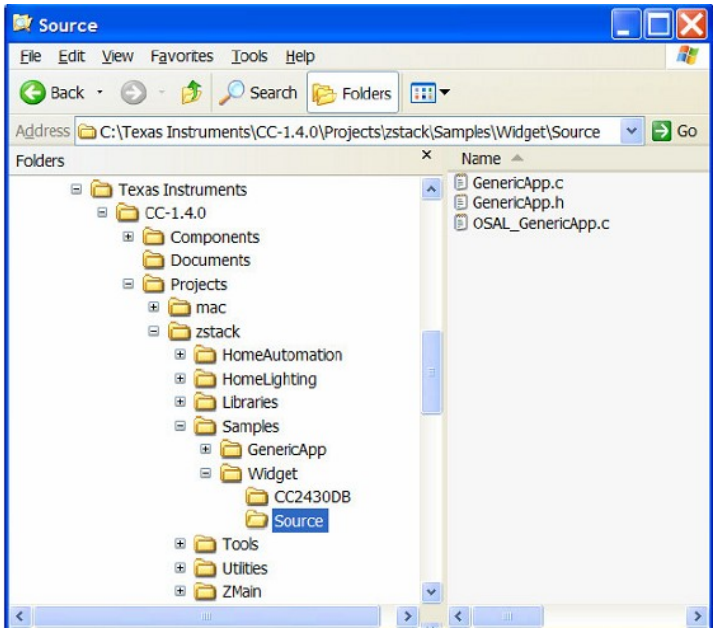
1，复制且重命名文件和文件夹



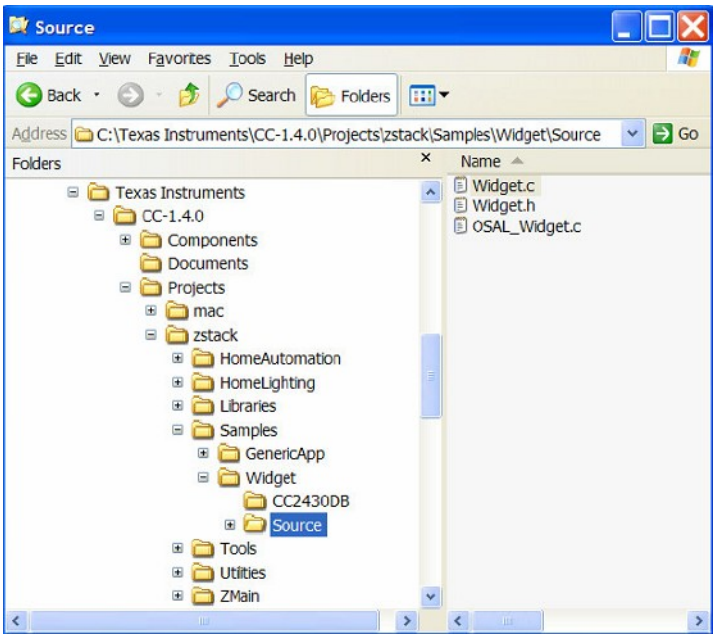
2，复制 GenericApp 文件夹且重命名为 Widget



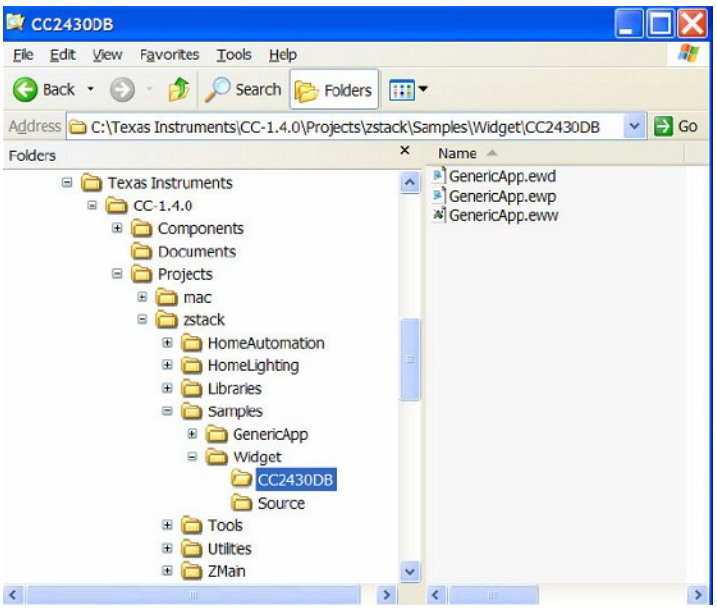
3，打开 Widget 文件夹下的 source 文件夹



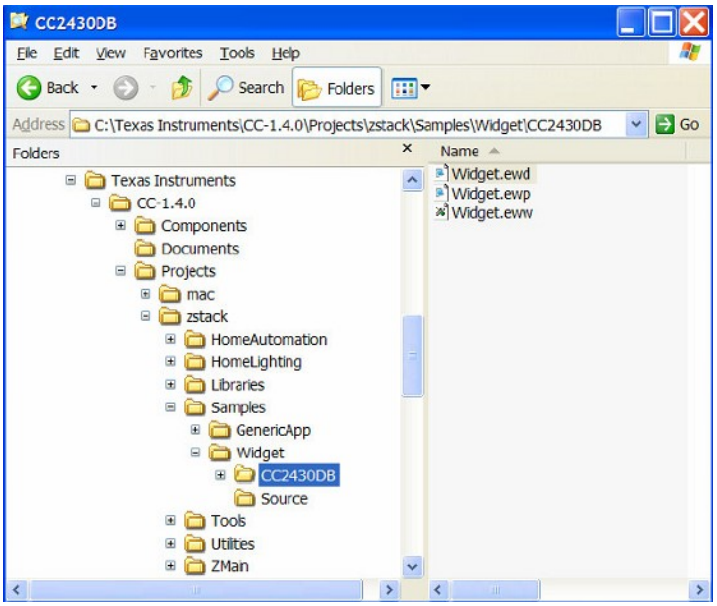
4，重命名所有文件，用 Widget 代替 GenericApp



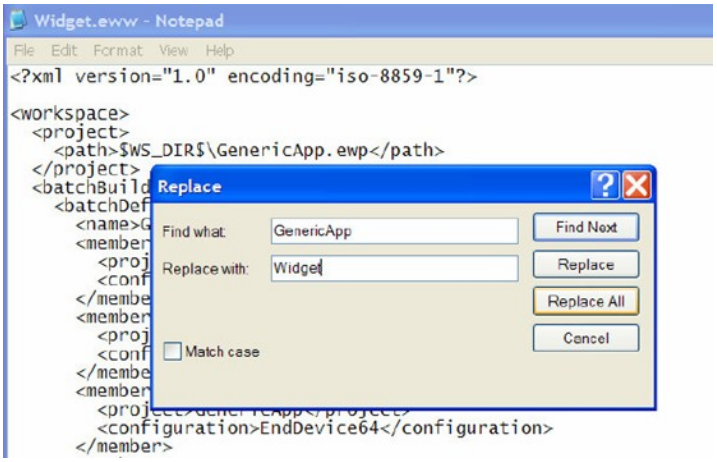
5，打开 Widget 文件夹下的 CC2430DB 文件夹



6，重命名所有的工程文件，将 Widget 代替 GenericApp

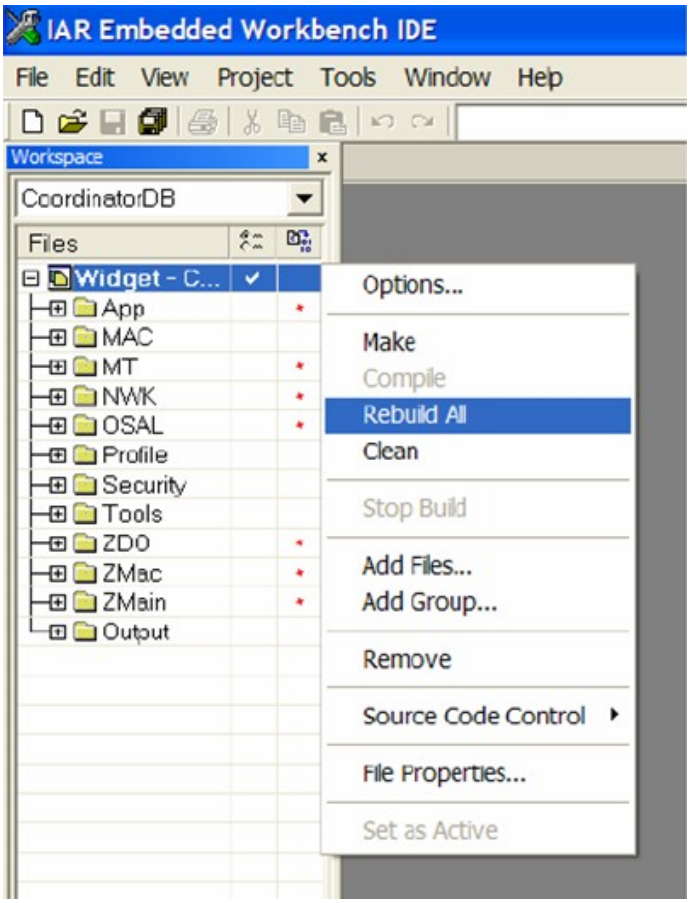


7，在…\Widget\CC2430DB 文件夹中，用编辑器编辑 Widget.eww、Widget.ewp，将所有 GenericApp 替换成 Widget。



8，在…\Widget\Source 文件夹中，用编辑器编辑 Widget.c、Widget.h、OSAL_Widget.c，将所有的 GenericApp 替换成 Widget。

完成对工程文件和源文件的修改后，就可以 building 工程，使用 IAR 打开 Widget.eww。



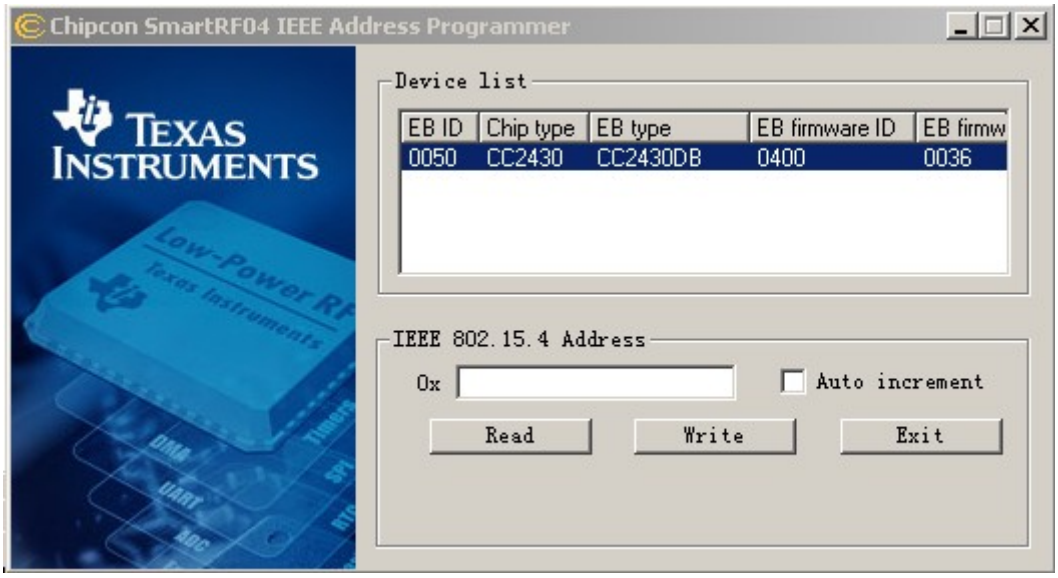
一般情况下，用户可以根据自己应用的需要修改 Widget.c 和 Widget.h，其他跟 ZStack 有关的源文件也可以根据需要修改。

第七章 如何使用 TI 提供的辅助软件

TI 提供了相当多的辅助软件，包括“IEEE_Address_Prog”、“SmartRF04Progr”、“Packet_Sniffer”、“SmartRF_Studio”，下面分别做简要介绍。

IEEE_Address_Prog

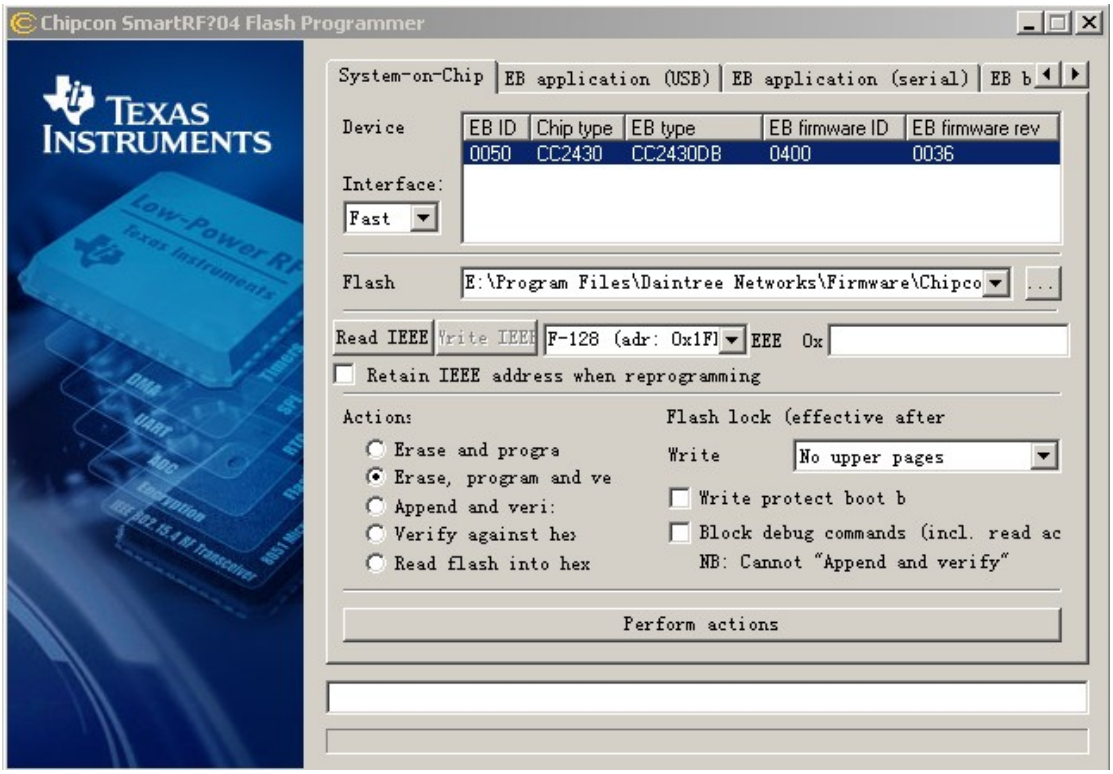
当用仿真器将节点和 PC 正常连接后，此软件可以自动识别设备，如下图所示：



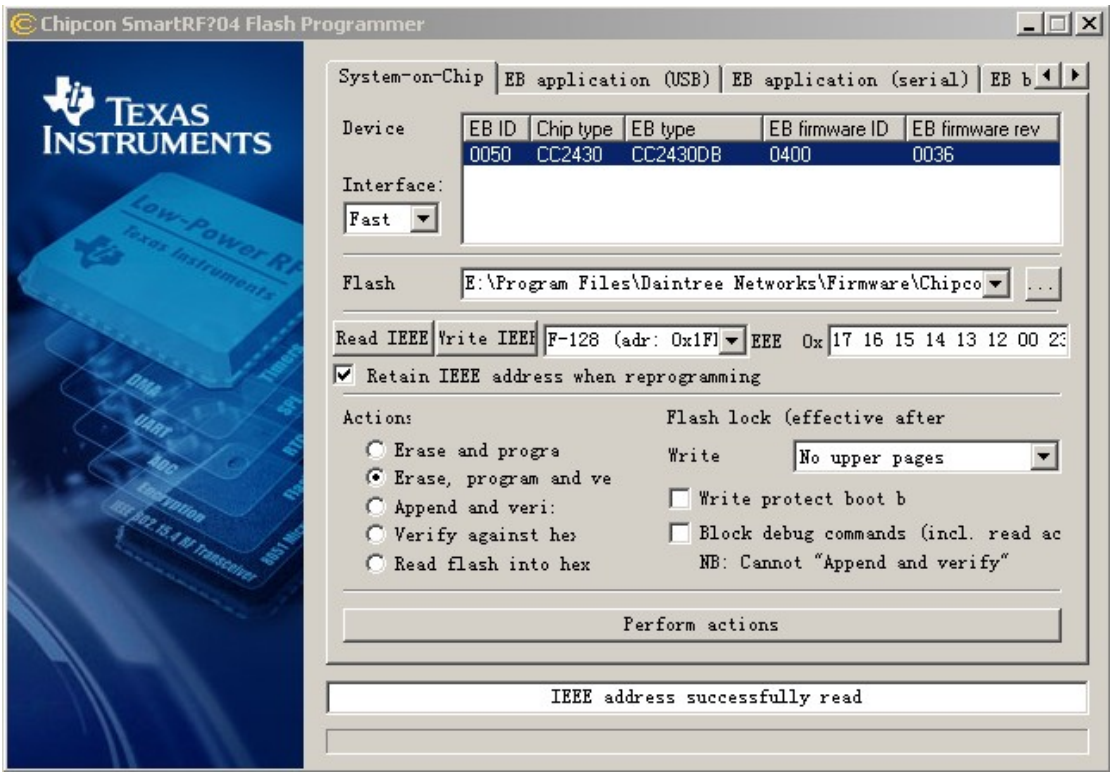
此软件的作用是向节点中写入 IEEE 地址，分别通过“Read”和“Write”按钮可以对 IEEE 地址进行读写操作，勾选“Auto increment”可使地址自动增加 1。软件的具体使用方法请参考“User Manual IEEE Address Programmer.pdf”文档。

SmartRF04Progr

当用仿真器将节点和 PC 正常连接后，此软件可以自动识别设备，如下图所示：



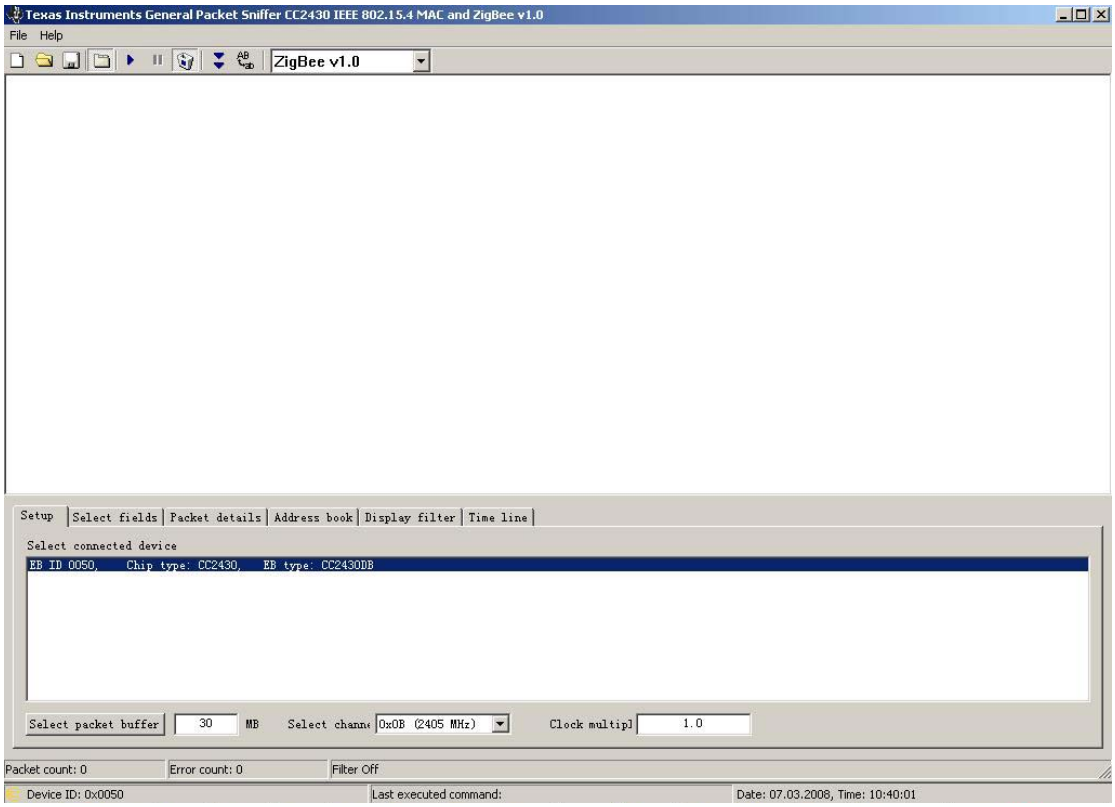
此软件可以烧写节点 Flash，后缀名为 hex，还可以读写 IEEE 地址。勾选“Retain IEEE address when reprogramming”后可以在烧写 flash 后防止 IEEE 地址被擦掉。如下图所示：



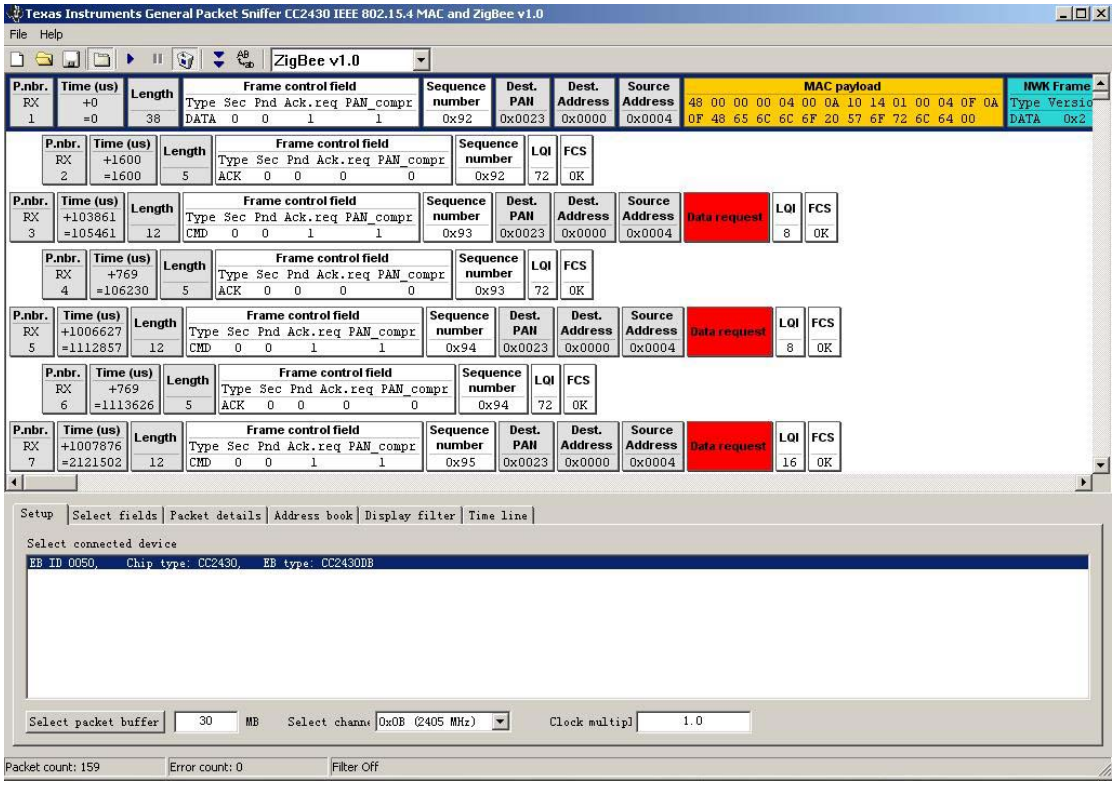
软件的具体使用方法请参考“User Manual Flash Programmer.pdf”文档。

Packet_Sniffer

当用仿真器将节点和 PC 正常连接后，此软件可以自动识别设备，并显示在“Setup”栏，如下图所示，



最下面分别是“接收包缓存”，“信道选择”，“时钟”，信道选择可以选择所监控的信道，默认为信道 11，其余 2 个参数保持默认。点击上面的 play(三角符号)，可以开始捕获，捕获到的数据包显示如下：

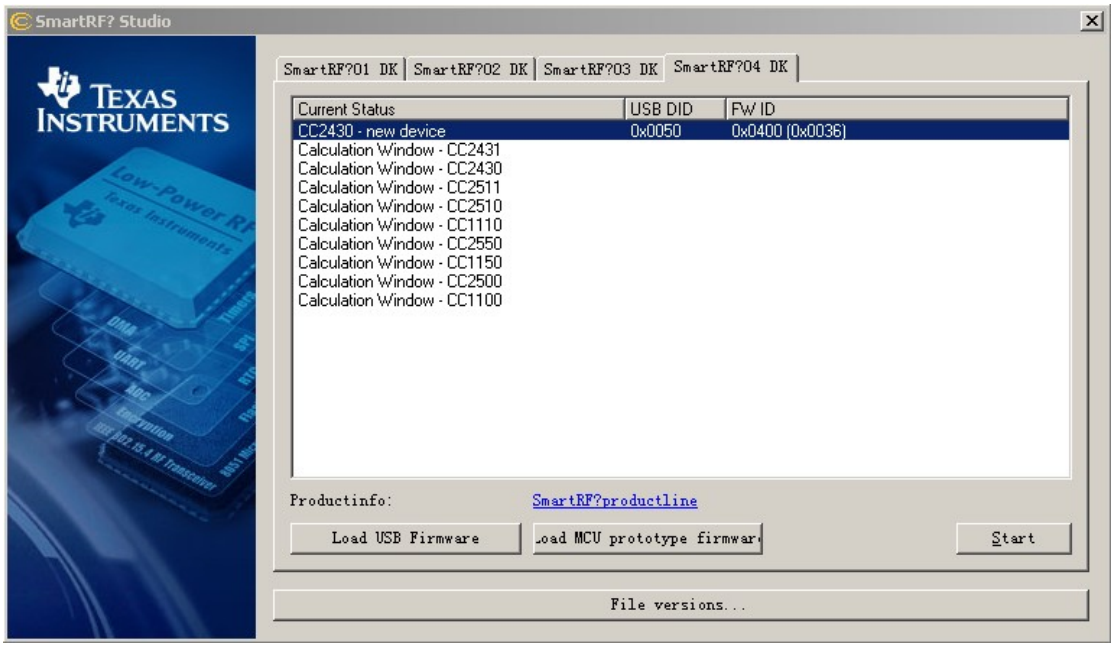


“Select fields”可以选择对捕获包的过滤，“Packet details”显示捕获包的信息，如收到帧的长度、负载数据、RSSI 值等。

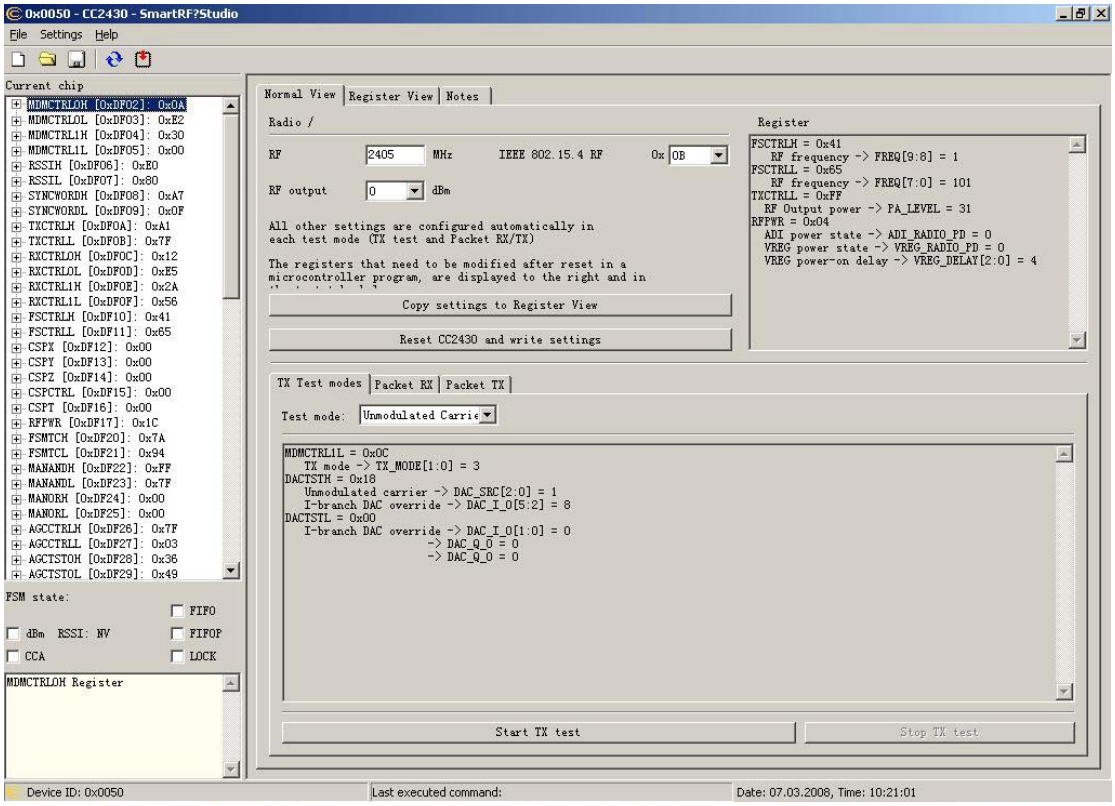
软件的具体使用方法请参考“General Packet Sniffer User Manual.pdf”文档。

SmartRF_Studio

当用仿真器将节点和 PC 正常连接后，此软件可以自动识别设备，并显示在“SmartRF04 DK”栏，如下图所示：



点击“Start”按钮可以打开操作界面，如下图所示，



此软件可以直观的显示芯片各寄存器的值，并在“Register View”栏可以直接修

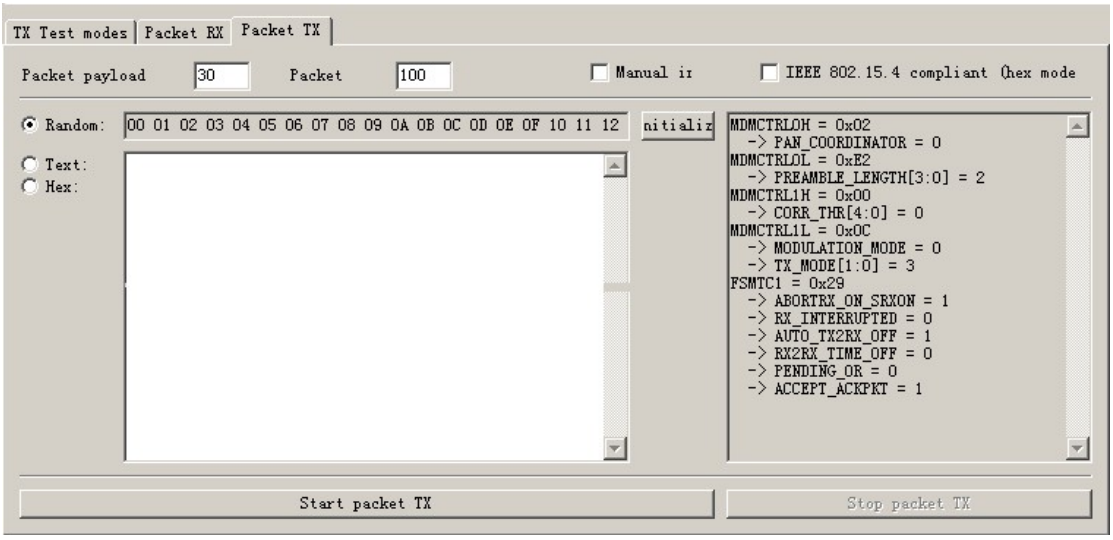
改各寄存器的值。

在软件的“Normal View”栏中可以方便的进行无线收发测试，如下图所示，分别连接 2 个节点到 PC，一个节点作为发送设备，另一个作为接收设备。

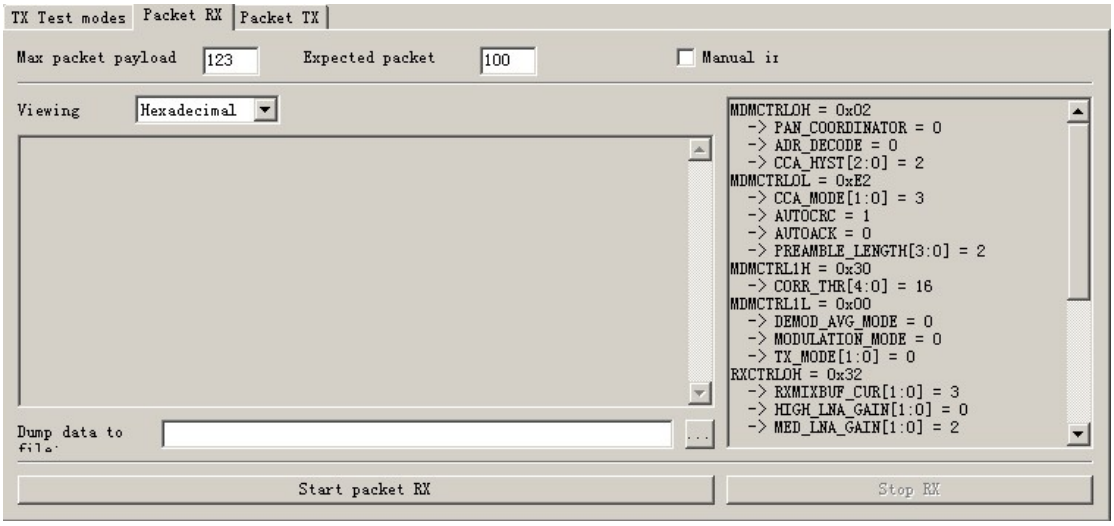
发送端可以选择负载长度、发送包数量，是否手动发送、是否是满足 IEEE 802.15.4 规范的帧，设置完成后点击“Start Packet TX”按钮开发发送。

接收端选择最大包负载个数、期望收到的包数量、是否手动，设置完成后点击“Start Packet RX”按钮开始接收。通常情况下是让接收端处于接收状态后，发送端再开始发送数据。

发送设备：



接收设备：



软件的具体使用方法请参考“SmartRF_Studio_User_Manual_x_y_z.pdf”文档。