**优微电子科技**

**CC2530 基础实验**

**IAR工程创建与配置**

# 一、实验介绍

嵌入式**IAR Embedded Workbench for 8051**提供一个框架，任何可用的工具都可以完整地嵌入其中，这些工具包括：

* 高度优化的IAR 8051 C/C++编译器；
* 8051 IAR汇编器；
* 通用IAR XLINK Linker;
* IAR XAR库创建器和 IAR XLIB Librarian;
* 一个强大的编辑器、工程管理器；

嵌入式IAR Embedded Workbench for 8051适用于大量8位、16位以及32位的微处理器和微控制器，使用户在开发新的项目时也能在所熟悉的开发环境中进行。它为用户提供一个易学和具有最大量代码继承能力的开发环境，以及对大多数和特殊目标的支持。嵌入式IAR Embedded Workbench for 8051有效提高用户的工作效率，通过IAR工具，用户可以大大节省工作时间。我们称这个理念为：“不同架构，同一解决方案”。

# 实验目的

通过本实验了解并掌握**IAR Embedded Workbench for 8051**使用，掌握**CC2530**基础实验工程的创建以及配置。

# 实验设备

## 3.1、硬件设备

（1）ZIGBEE开发板

（2）SmartRF04EB仿真器

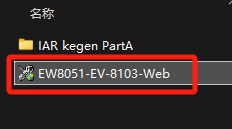
## 3.2、软件环境

（1）IAR Embedded Workbench for 8051集成开发环境

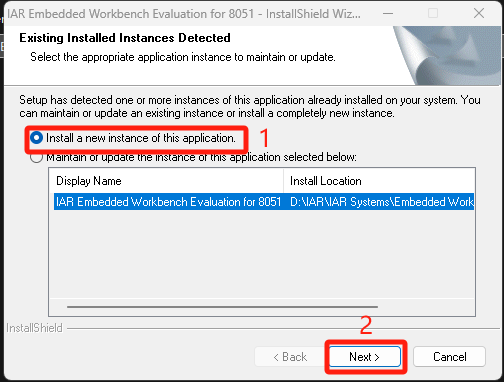
# 四、实验原理

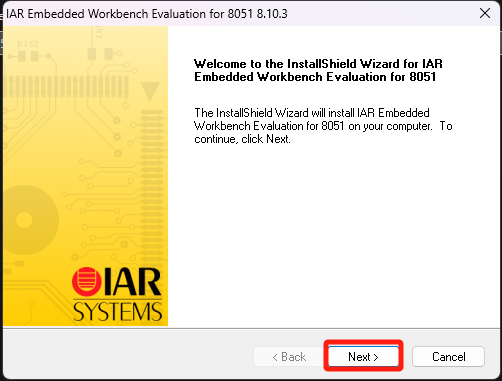
## 4.1、IAR安装

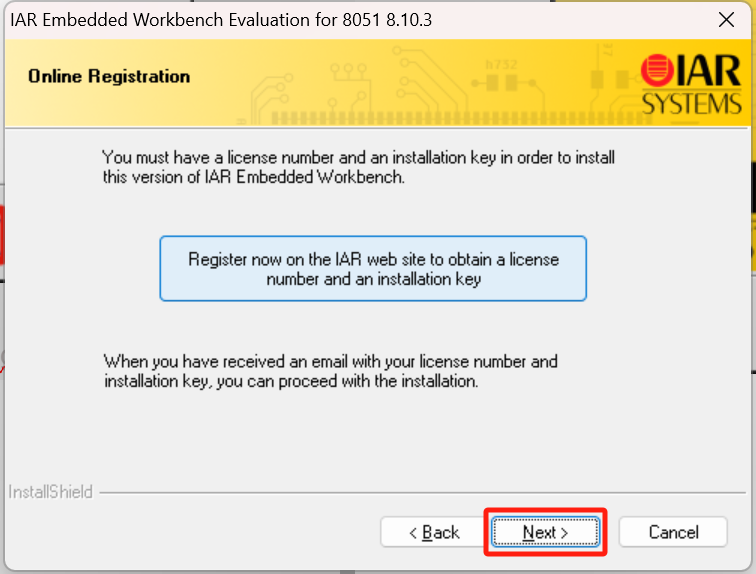
打开**ZIGBEE资料包**里的**开发环境**，并**以管理员身份运行EW8051-EV-8103-Web**，

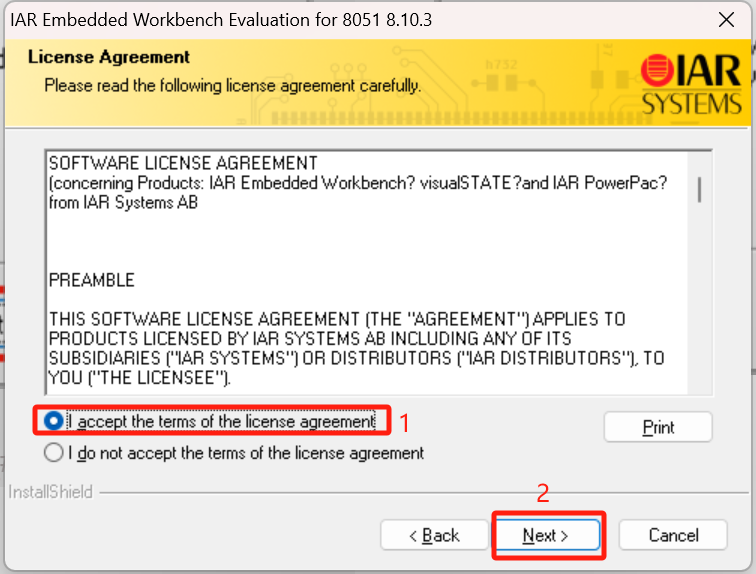


选择下载程序并持续点击**Next**。

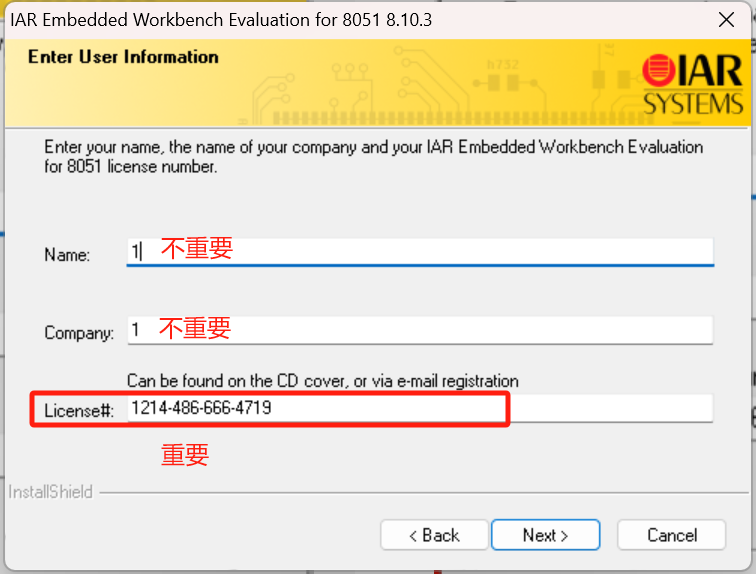




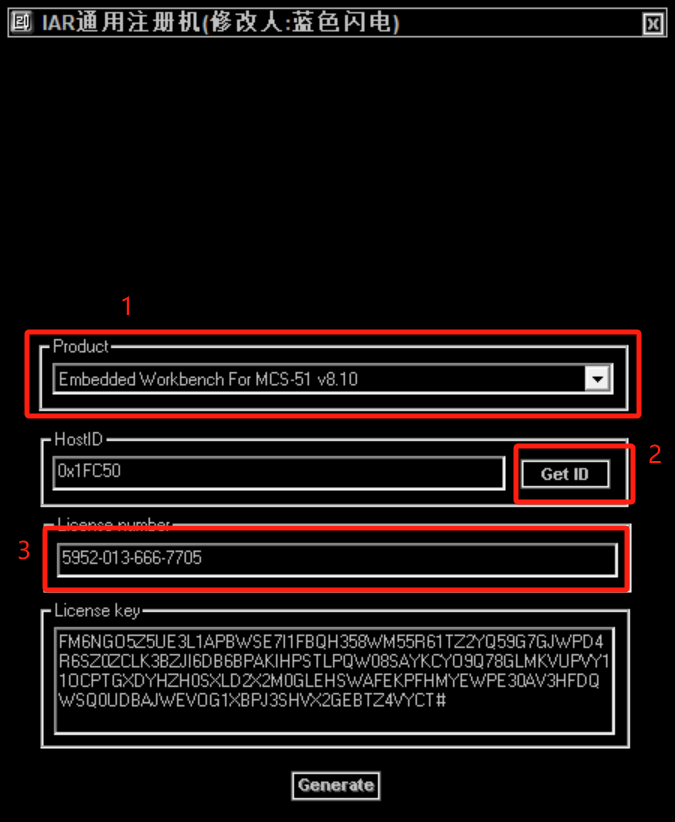


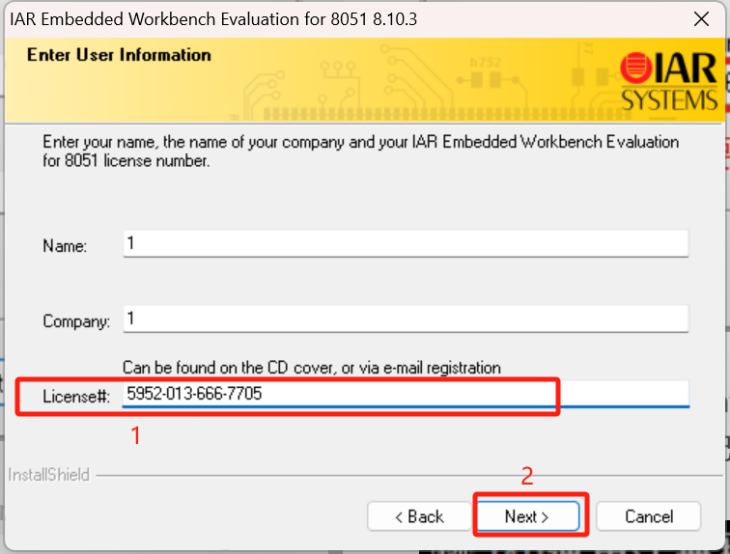


以上的**Name**和**Company**并不重要，可以自行填写，重要的是**License#**。

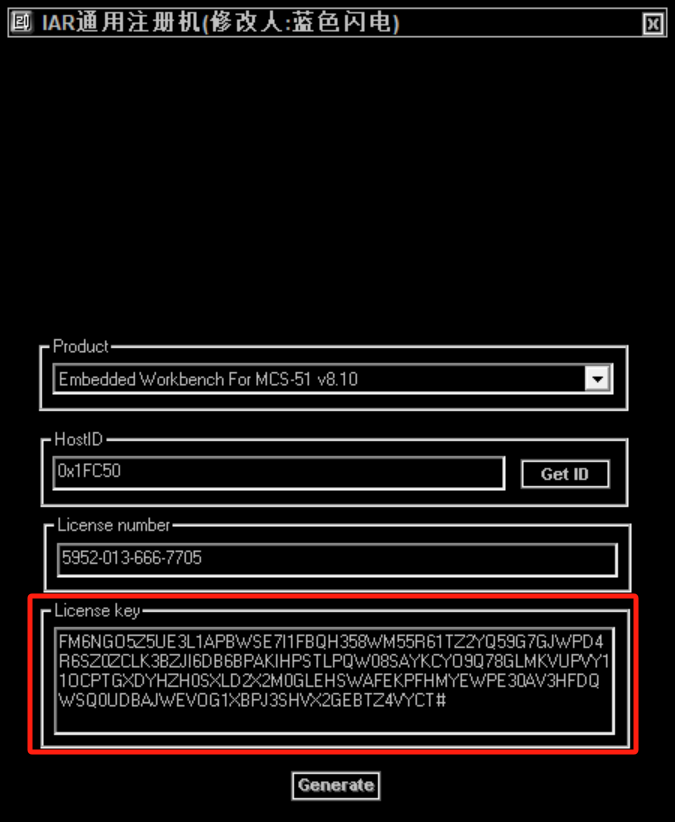


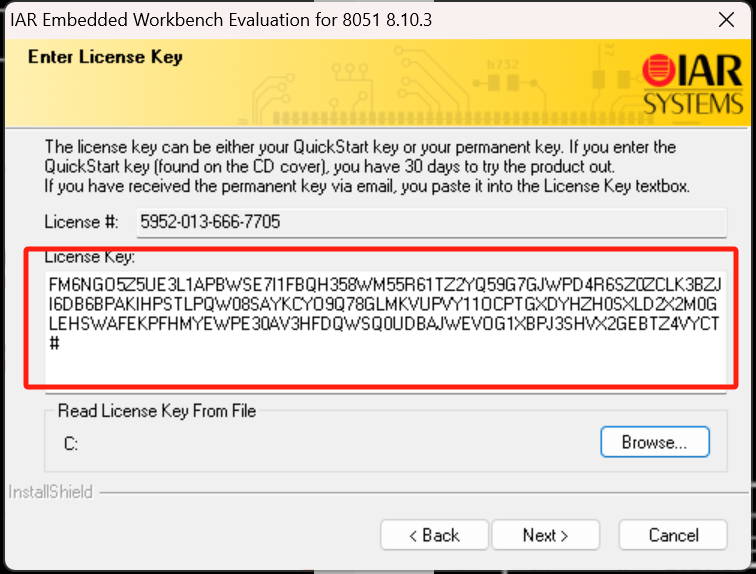
打开**开发环境**的**IAR kegen PartA**，记得要**以管理员身份运行**，打开前建议先把声音调低。选择**MCS-51**，点击**Get ID**，然后复制**License number**，粘贴到**License#**内，依旧是点击**Next**。





复制**License key**内容，粘贴到**License Key**内。如果需要修改安装路径的话点击**Browse**，默认是在**C盘**。

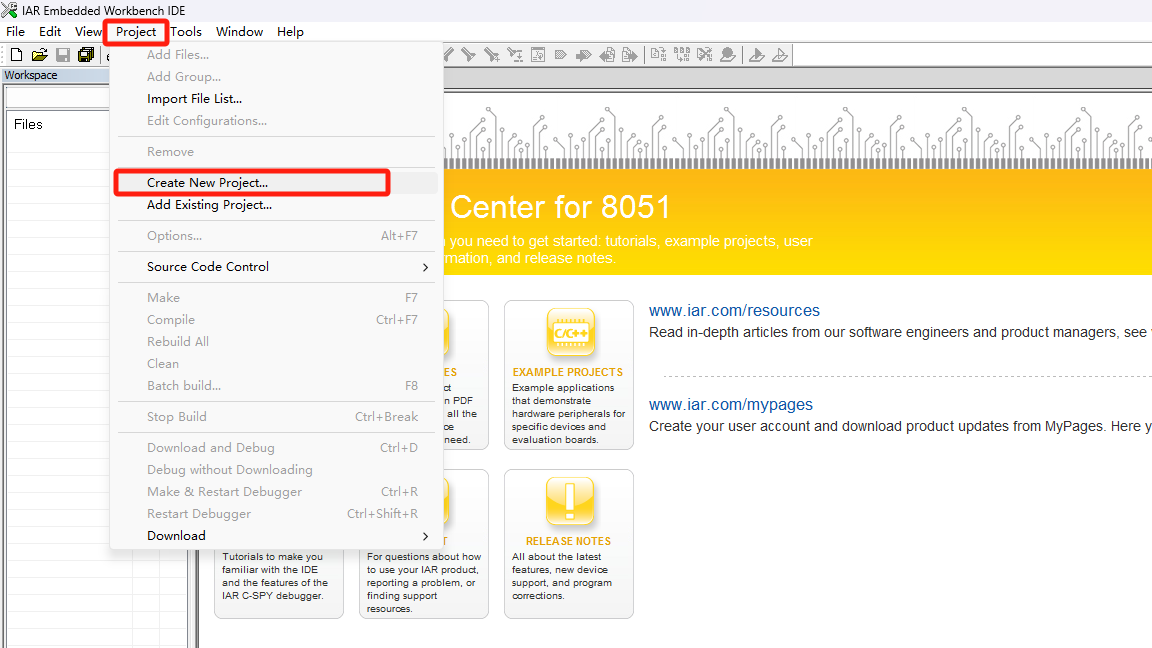




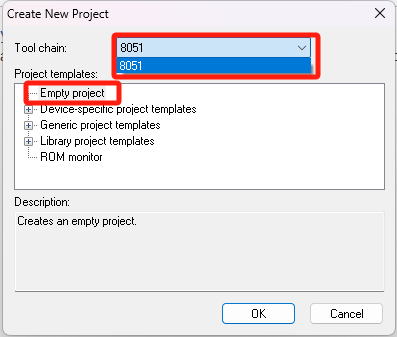
后续只需点击**Next**即可，最后点击**Intall**，等待一些时间就下载好了**IAR Embedded Workbench for 8051。**

## 4.2、建立工程

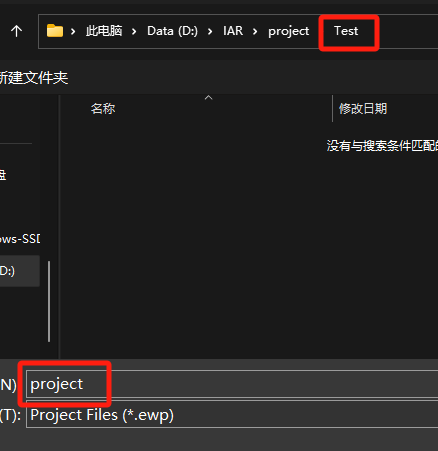
打开**IAR Embedded Workbench for 8051**，点击**Project**，选择**Create New Projec**t生成空的工程。

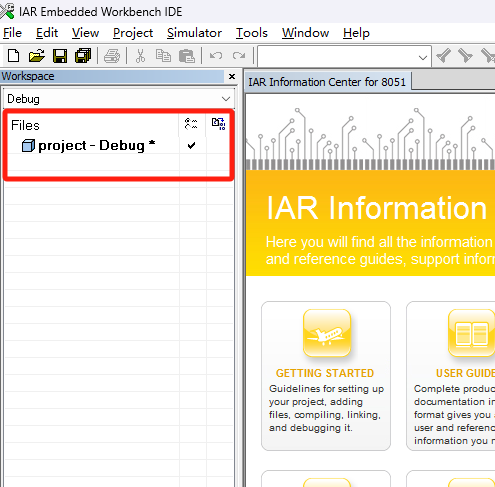


选择**8051**和**Emply project**，

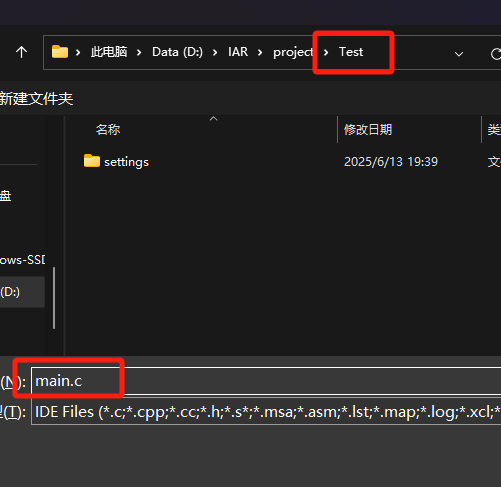


由于建立工程会生成不少工程文件，所以先选择一个为创建一个文件夹用于放置该工程，在这我创建一个名为****Test****的文件夹存放名为****project****的工程。

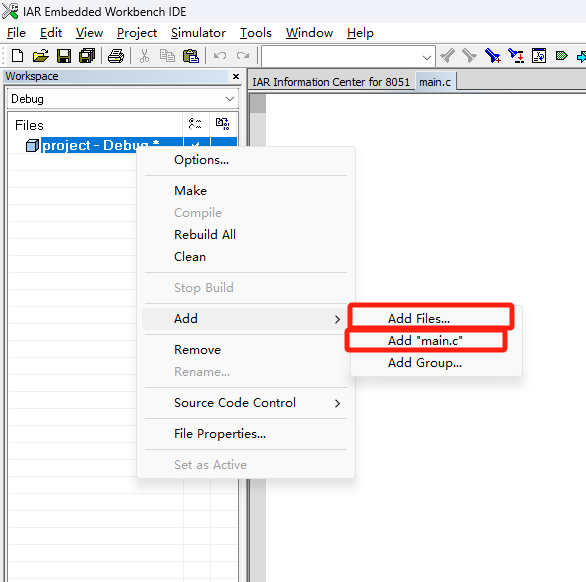




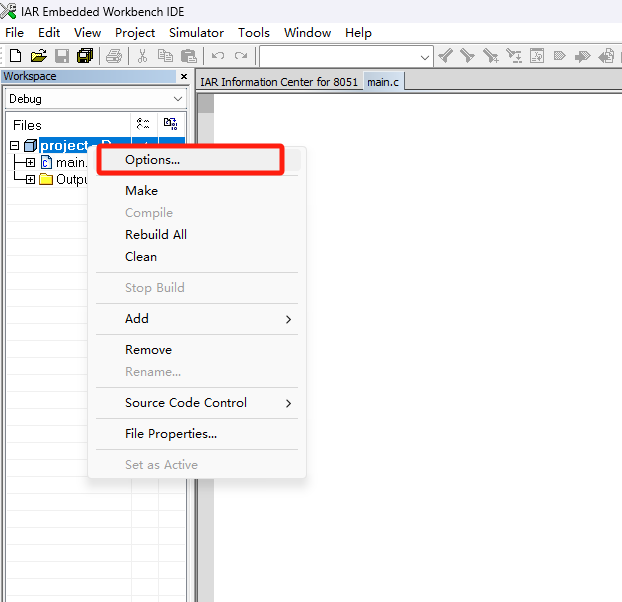
此时工程已创建完成。接下来是创建工程下的C语言文件作为该实验的主文件，****Crtl + N****创建文件，****Crtl + S****保存文件。保存在该工程文件下，在此我命名为****main.c****作为主文件，也可以命名为其他名字，一定要在命名的后面加上**.c**区分为C语言文件。



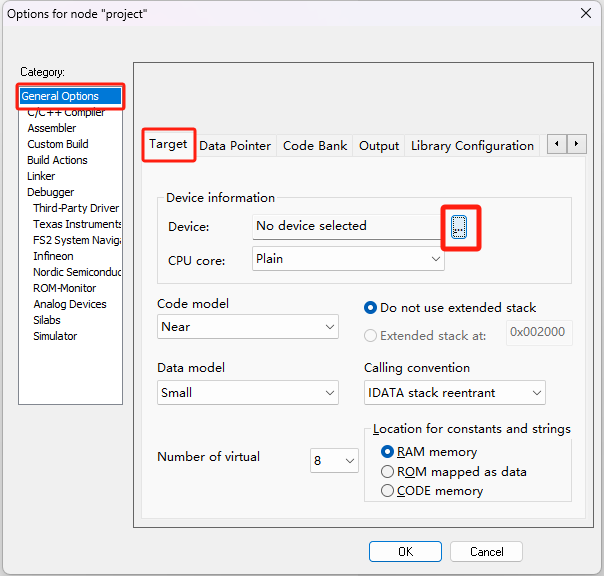
然后把文件添加进来，两种方式都可以。



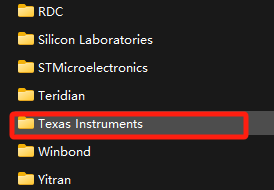
把C语言文件添加进工程后是进行下载配置，



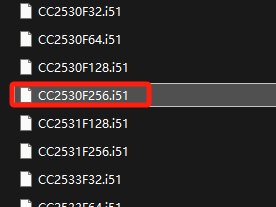
我们开发板的芯片是**CC2530**，选择芯片**CC2530**，



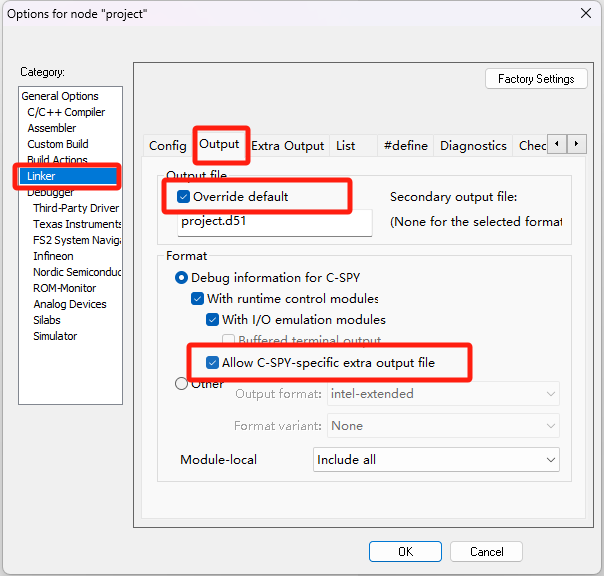
该目录在我们安装IAR的目录下，IAR\IAR Systems\Embedded Workbench 6.0 Evaluation\8051\config\devices\Texas Instruments



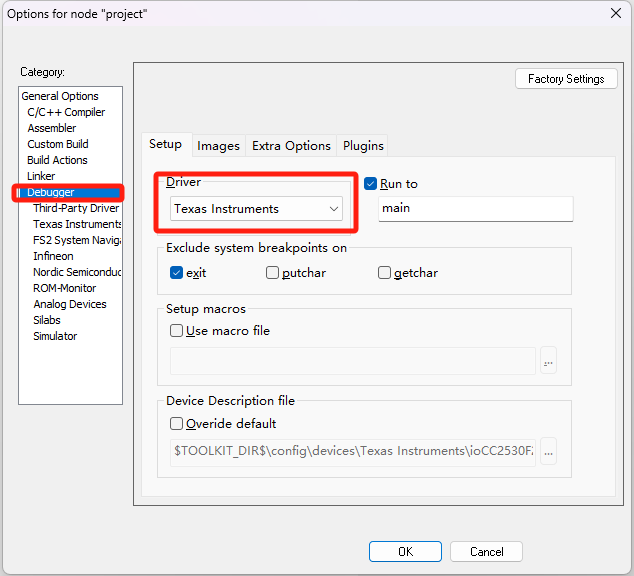
找到并选择****CC2530F256.51****，



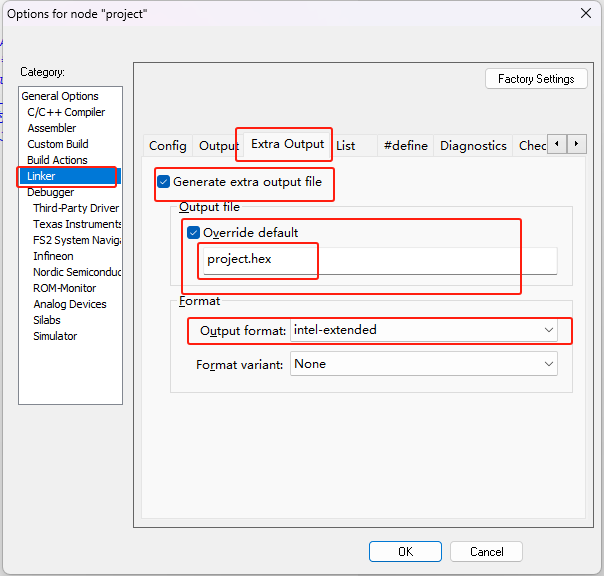
将这两个勾选上，程序便可进行下载和在线仿真，



程序默认为软件仿真，是没法下载到程序内，所以我们要修改一下仿真方式使我们可以正常地把程序下载到我们的开发板上。



接下来是在我们编译时生成对应的****.hex文件****，记得把****.sim****改为****.hex****，

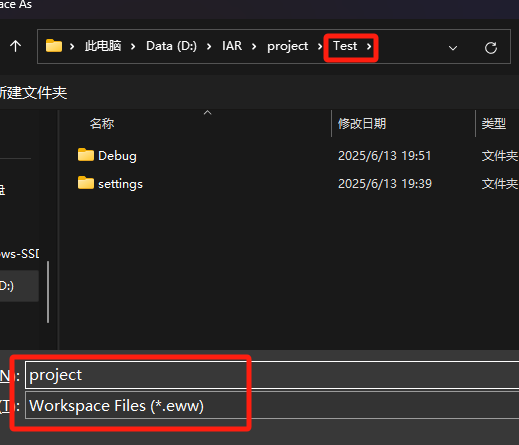


点击****OK****进行保存我们就完成下载配置工作。

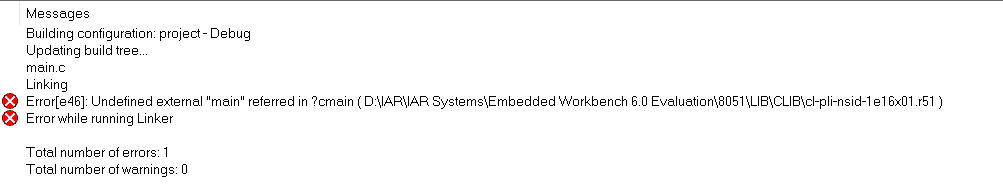
接着我们试着编译，看看会有什么效果，



此时弹出来要我们保存的是以**.eww**为后缀的**工程空间文件**，一个工程空间文件可以包含多个工程文件，工程空间文件只是工程文件的一个载体。我们把该工程文件建立在我们的**Test**文件下，在这里我命名为**project**。



工程空间文件建立后，底下的信息窗口会显示错误，这是因为我们没有定义主函数，



解决此问题只需在我们建立的文件下写入代码：

**#include<iocc2530.h> //头文件**

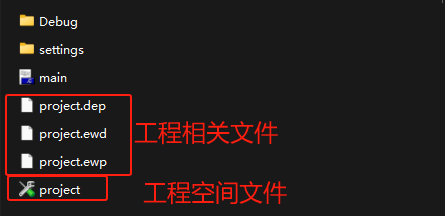
**void main(void)**

**{**

**}**

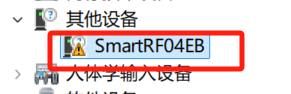
再次编译时就没有错误了。

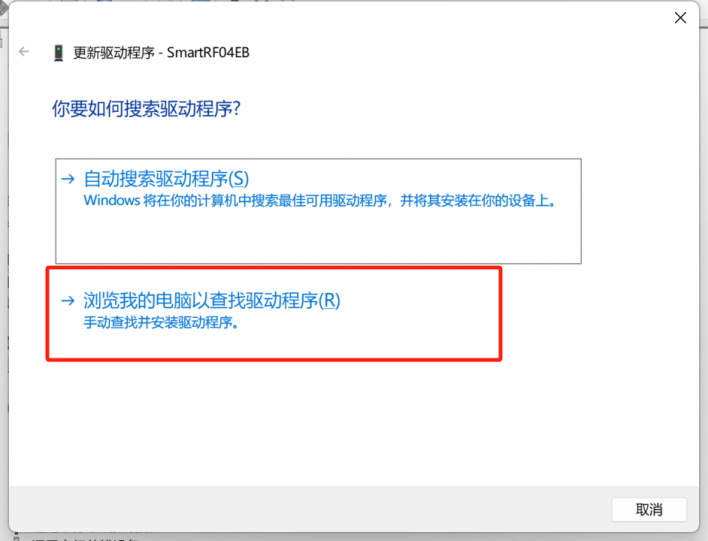
现在打开看一下Test文件夹里会有****工程相关文件****和****工程空间文件****，而main文件是我们添加到工程的一个C语言文件且作为主文件使用，在/Debug/Exe下就有我们的hex文件。



## 4.3、仿真器识别

将仿真器****SmartRF04EB****接入电脑，打开电脑内的****设备管理器****里找到仿真器的名称，右键选择****更新驱动程序****。





点击****浏览****找到资料包内****开发环境****里的****SmartRF04EB仿真器****，根据自己电脑选择32位或者64位，我这里是64位，



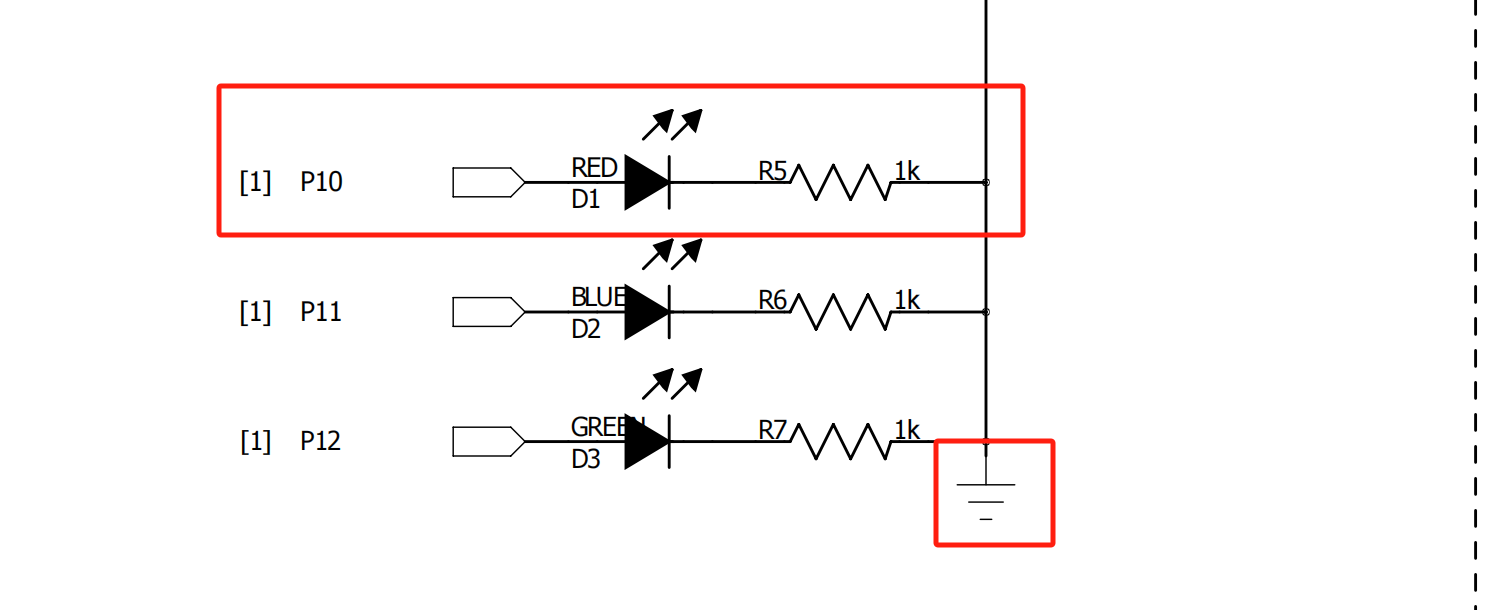
选择完后点击下一步进行安装。



这样我们的仿真驱动就下好了。

# 五、实验演示

接下来我们检测以下我们是否已经搭建好我们的开发环境，首先打开我们的原理图，看到P10上连接的是发光二极管且二极管的负极接的是GND，



回到IAR，在我们建立的main.c文件下输入以下代码：

**#include<iocc2530.h> //头文件**

**void main(void)**

**{**

**P1SEL &= ~0x01;**

**P1DIR |= 0x01;**

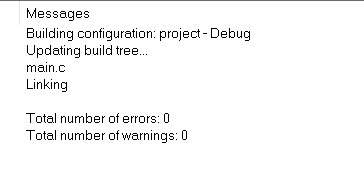
**P1\_0 = 1;**

**}**

这是我们用来检测我们是否已经把开发环境搭建好的代码，之后会解释每行代码的意思，输入完后看一下上方的工具栏，



点击****编译****，下方的信息窗口显示以下就说明我们的代码没有出现错误和警告，



然后点击****下载并仿真****，如果点击完出现错误弹窗，关掉弹窗后需要我们按一下仿真器上的一个****复位键****，此时我们的程序已经下载到开发板上的芯片内，然后工具栏也会多出一行调试工具栏目，下面是对各个功能的解释，要注意的是光标所在的行是指下一次将执行的命令并非正在执行的命令。

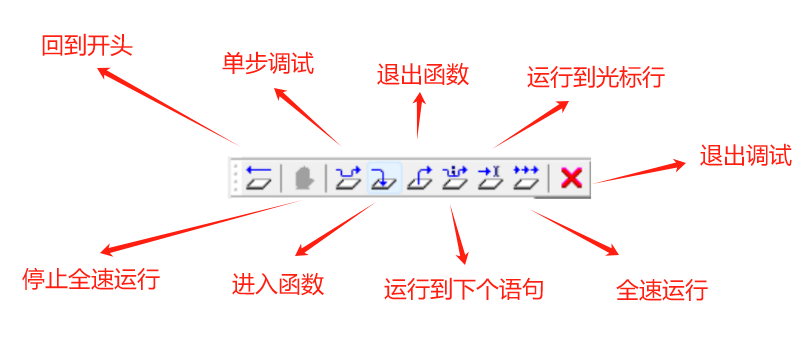
****返回开头：****回到主函数的第一句代码。

****停止全速运行：****是在进入****全速运行****后才激活的，可以停止运行。

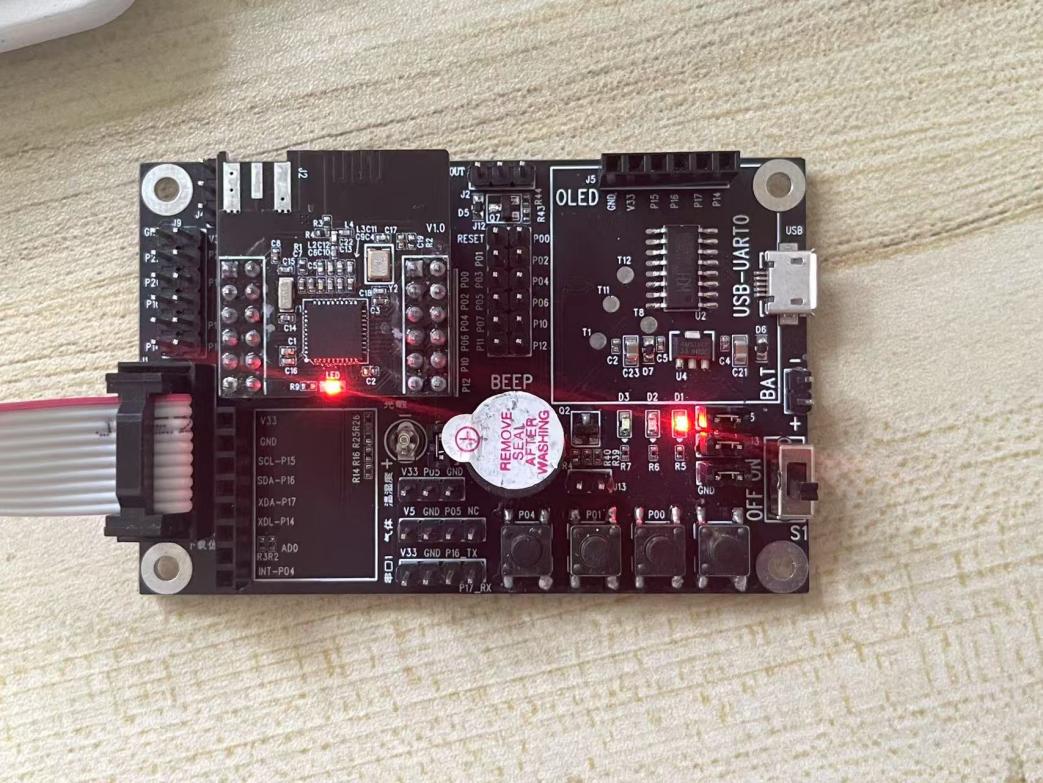
****单步调试：****根据我们的代码点击一次就往下走一步。

****进入函数：****当我们走到一个函数时点击这个就可以进入到函数内部。

****全速运行：****让主函数里的程序直接运行。



点击**全速运行**便可点亮我们的LED，我们来看一下效果，



到这里我们的开发环境搭建就全部完成了。