

1.GPIO输出控制实验-控制Led亮灭

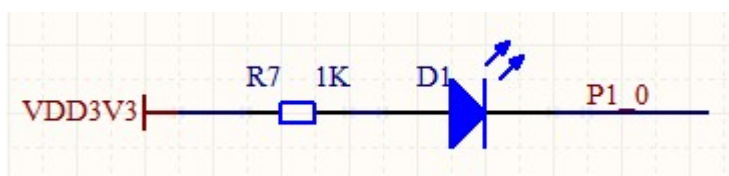
1. 实验目的

- 1) 通过实验掌握 CC2530 芯片 GPIO 的配置方法，带你一步步走进嵌入式大门
- 2) 掌握 Led 驱动电路及开关 Led 的原理
- 3) 试着看数据手册 [cc2530-datasheet.pdf](#)

2. 实验设备

- 硬件：PC 机 一台
EB2530（底板、核心板、仿真器、USB线） 一套
- 软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

3. 实验相关电路图



由于发光二极管单向导电特性，即只有在正向电压（二极管的正极接正，负极接负）下才能导通发光。P1.0 引脚接发光二极管(D1)的负极,所以 P1.0 引脚输出低电平 D1 亮，P1.0 引脚输出高电平 D1 熄灭。

提示：TI 官方的开发板是高电平点亮 LED 哦，我们的板子是低电平亮，更符合国人习惯，后面协议栈也是一样的。

更多电路图请看“EB2530\相关资料与软件\原理图和参考 PCB”下的原理图

4. 实验相关寄存器

操作P1.0我们需要掌握相关寄存器的作用和配置方法。如下表所示（更详细的介绍请参考“EB2530\相关资料与软件\cc2530-datasheet.pdf, 英文不好的可参考CC2530中文数据手册完全版.pdf，翻译有250页并非官方中文，有个别地方有错，请以英文为主哦”）：

寄存器	作用	描述
P1 (0x90)	端口 1	端口 1。通用 I / O 端口。可以从 SFR 位寻址。
P1SEL (0xF4)	端口 1 功能选择	P1.7 到 P0.0 功能选择 0: 通用 I / O 1: 外设功能
P1DIR (0xFE)	端口 1 方向	P1.7 到 P1.0 的 I/O 方向 0: 输入 1: 输出
P1INP (0xF6)	端口 1 输入模式	P1.7 到 P1.2 的 I/O 输入模式。由于 P1.0 和 P1.1 没有上拉/下拉功能，P1INP 暂时不需要配置，了解一下为后面的实验打下基础 0: 上拉/下拉(见 P2INP (0xF7) - 端口 2 输入模式) 1: 三态

按照表格寄存器的内容，对 P1.0 口进行配置，当 P1.0 输出低电平时 D1 被点亮。所以配置如下：

```
P1SEL &= 0x01; //配置P1.0为通用IO口
```

```
P1DIR |= 0x01; //配置 P1.0 为输出
```

由于 CC2530 寄存器初始化时默认值为（详细说明请参考 CC2530 数据手册（中文）.pdf）：

```
P1SEL = 0x00;
```

```
P1DIR = 0x00;
```

```
P1INP = 0x00;
```

所以 IO 口初始化我们可以简化初始化指令：

```
P1DIR |= 0x01; //配置 P1.0 为输出
```

也许有人会问手册怎么看啊？那么多页。我们看手册是并非一章章看下去，那样太浪费时间，恐怕看两个星期也看不完。现在教大家一个方法，比如这个实现是讲 GPIO 我们就直接看 CC2530 中文数据手册完全版.pdf 7. IO 端口那一章，再不行找一个代码中使用的寄存器搜索，这方法高效吧，多做一些实验，你看的也多了基本上也就学会了，方法很重要。

5. 源码分析

打开 “EB2530\第3章 基础实验\1.GPIO 输出控制实验-控制 Led 亮灭\Led\Led.eww” 工程，也可以在文件夹双击 “Led.eww”。我们先分析一下源码后再调试程序

```
/******
```

```
* 文件 名: main.c
```

```
* 作 者: Andy
```

```
* 修 订: 2013-01-08
```

```
* 版 本: 1.0
```

```
* 描 述: GPIO 输出控制实验 1 操作 IO 口控制 LED 灯的亮和灭
```

```
*****/
```

```
#include <ioCC2530.h>
```

```
typedef unsigned char uchar;
```

```
typedef unsigned int uint;
```

```
#define LED1 P1_0 //定义 P1.0 口为 LED1 控制端
```

```
/******
```

```
* 名 称: DelayMS()
```

```
* 功 能: 以毫秒为单位延时 16M 时大约为 530, 系统时钟不修改默认为 16M
```

```
* 入口参数: msec 延时参数，值越大，延时越久
```

```
* 出口参数: 无
```

```
*****/
```

```
void DelayMS(uint msec)
```

```

{
    uint i, j;

    for (i=0; i<msec; i++)
        for (j=0; j<530; j++);
}

/*****
* 名    称: InitLed()
* 功    能: 设置 LED 灯相应的 IO 口
* 入口参数: 无
* 出口参数: 无
*****/
void InitLed(void)
{
    P1DIR |= 0x01;          //P1.0 定义为输出口
}

/*****
* 程序入口函数
*****/
void main(void)
{
    InitLed();              //设置 LED 灯相应的 IO 口

    while(1)                //死循环
    {
        LED1 = 0;           //点亮 LED1
        DelayMS(1000);       //延时 1 秒
        LED1 = 1;           //LED1 熄灭
        DelayMS(1000);       //延时 1 秒
    }
}

```

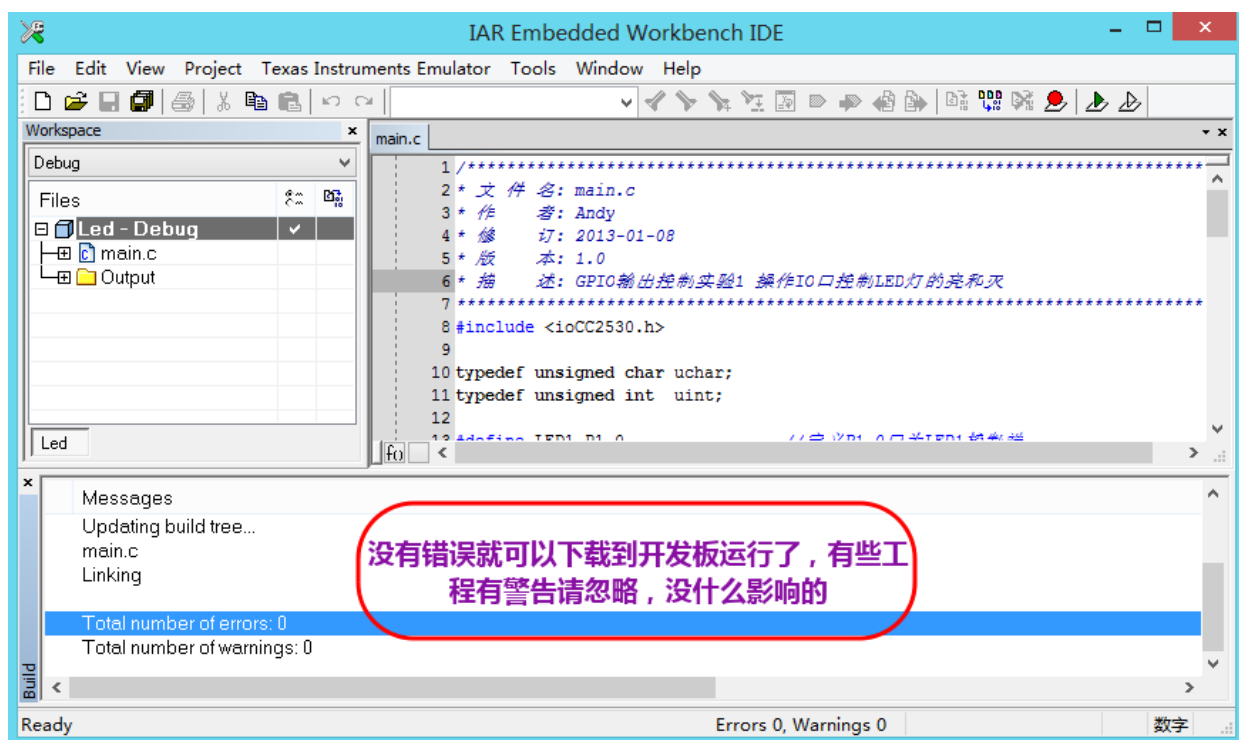
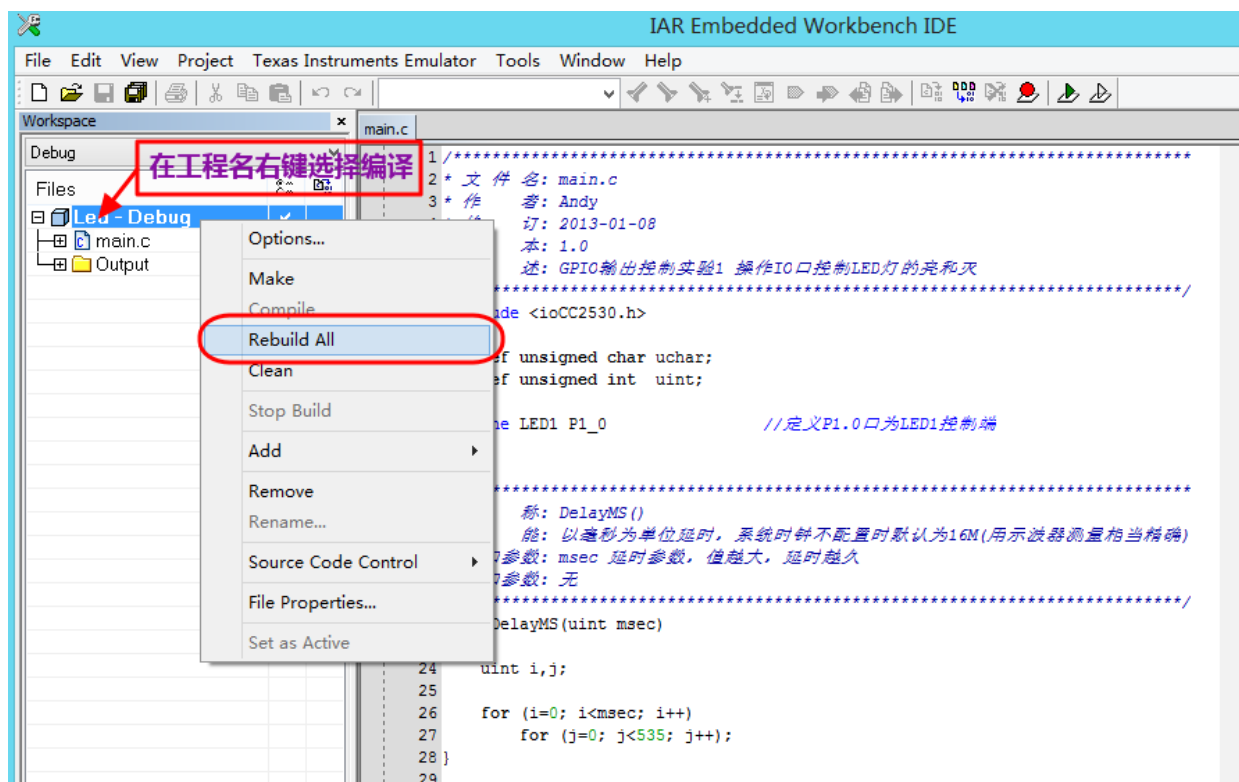
6. 仿真下载程序、调试（**根据自己的仿真器选择以下一种连接方式**）

1) SmartRF04EB 仿真器连接如下图:

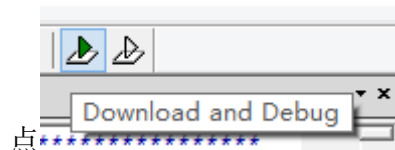


2) CC Debugger 连接如下图：

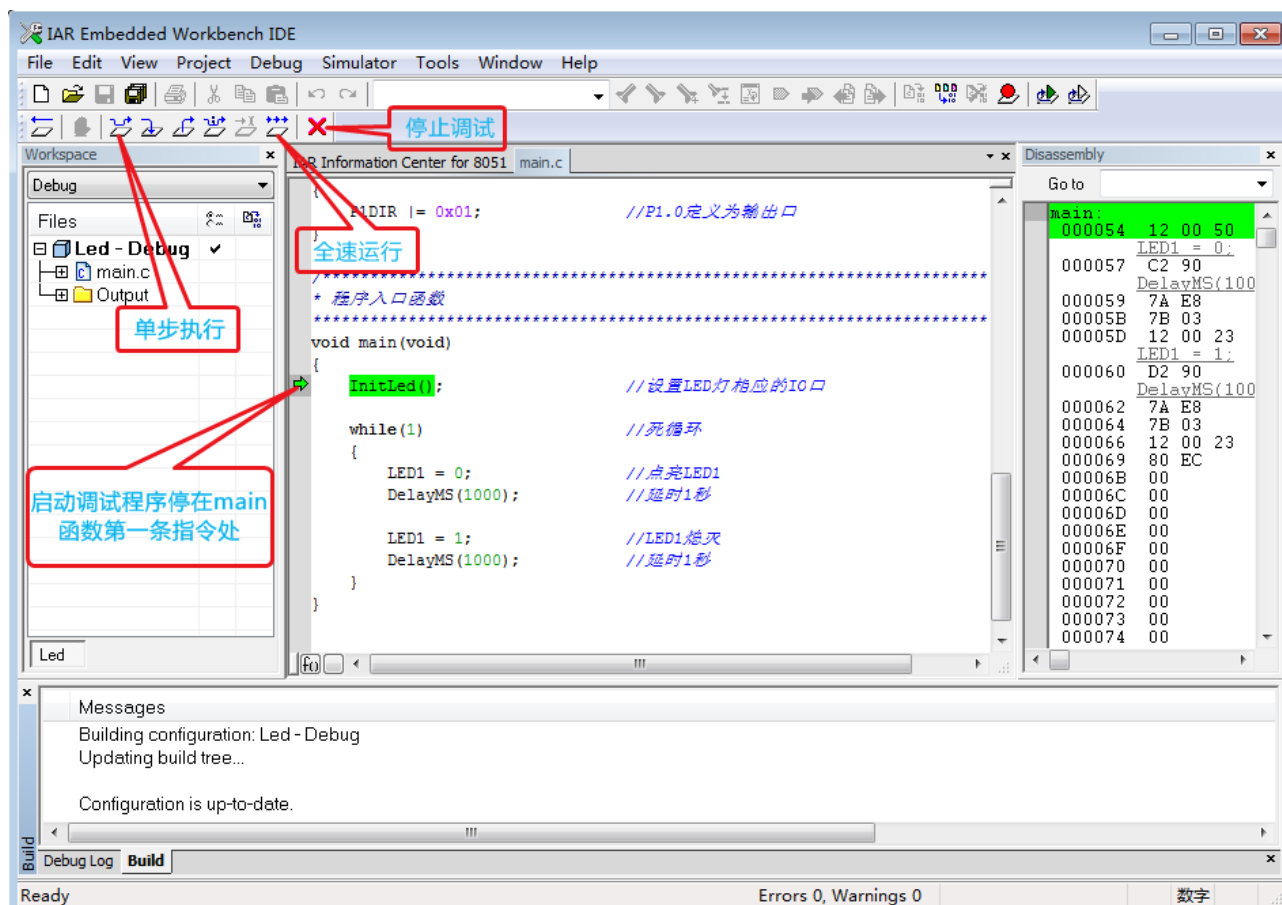




如果编译有错误，请参考最后一页图片说明。没有错误请继续

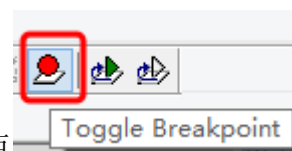


点*****下载调试程序，显示如下图所示：

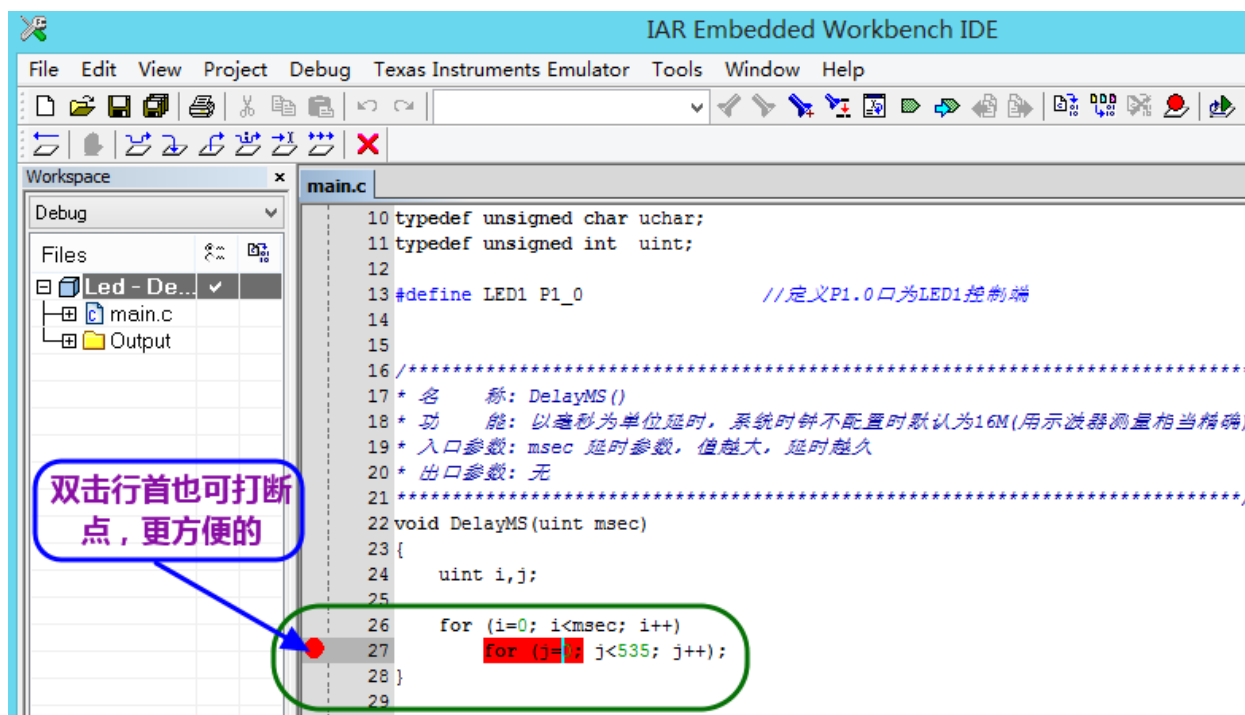


如果程序错误我们怎么跟踪呢？这时我们就要发挥仿真器的作用了，方法如下：

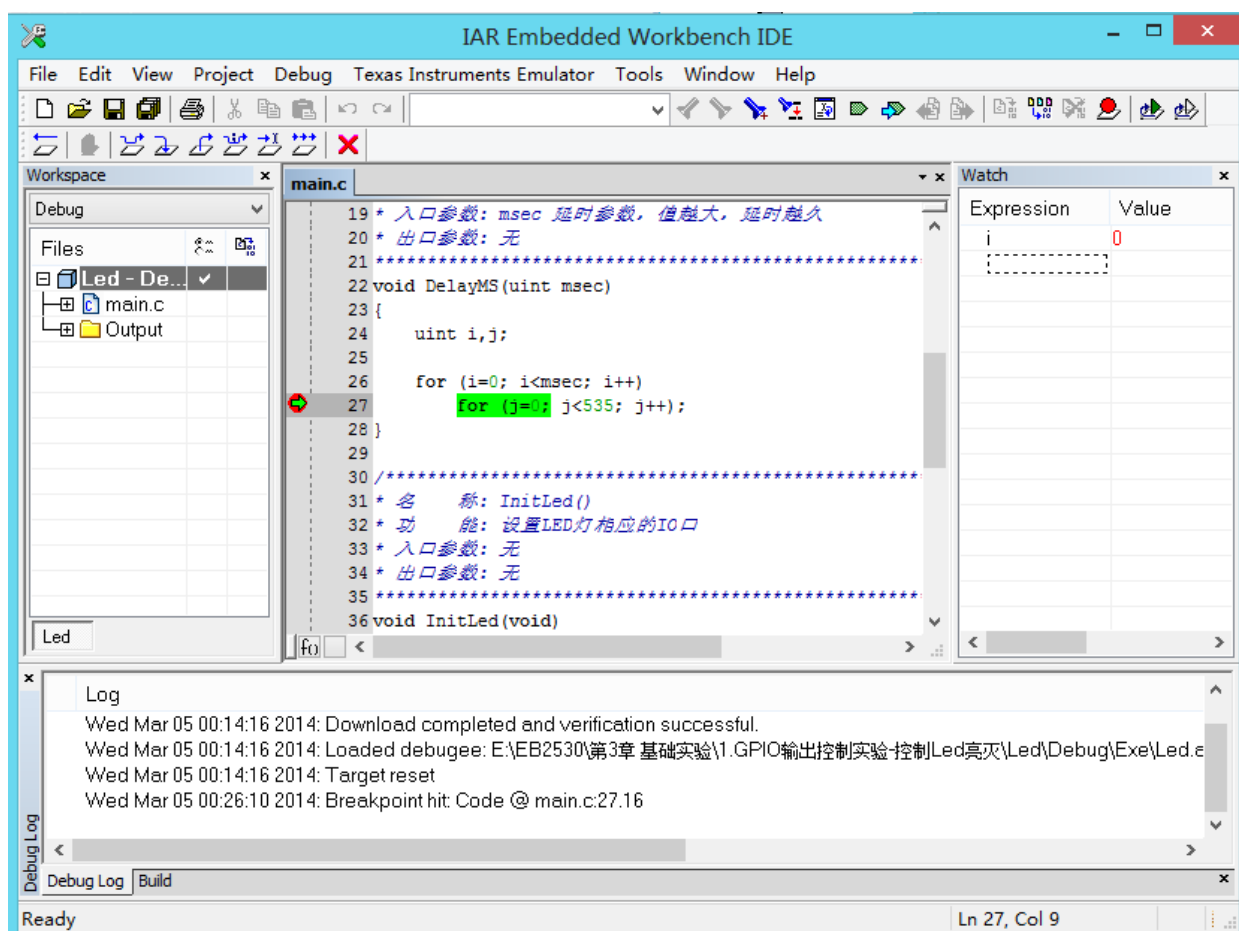
先打断点让程序停在断点处再单步跟踪变量的值。选定变量“Add to watch”即可。



光标放在要打断点的所在行，我们打在 main.c 的第 27 号，然后点，或者在所在行行首双击，也会出现一个断点，再次双击可以取消此断点。

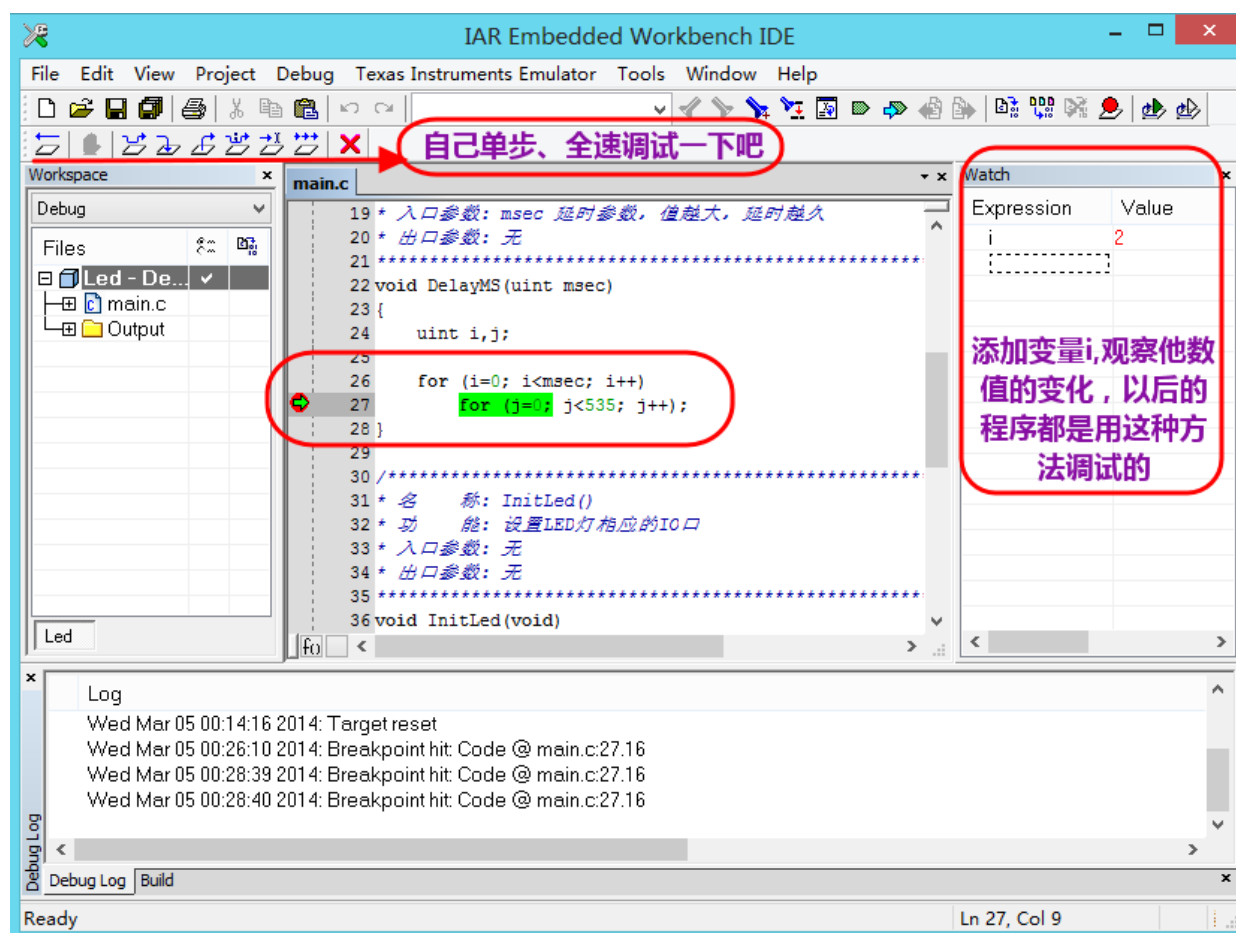


再点全速运行, 如果程序停在断点处时会变成绿色, 如下图所示



双击红色圆点即可取消断点, 再点全速运行程序就运行了, 可以观察实现现象了。

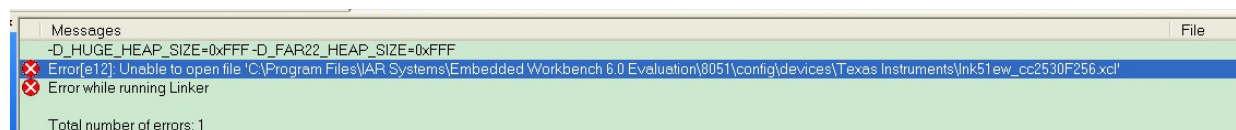
怎么观察变量的值呢？选定变量右键在弹出的窗口选择“Add to watch”即可。



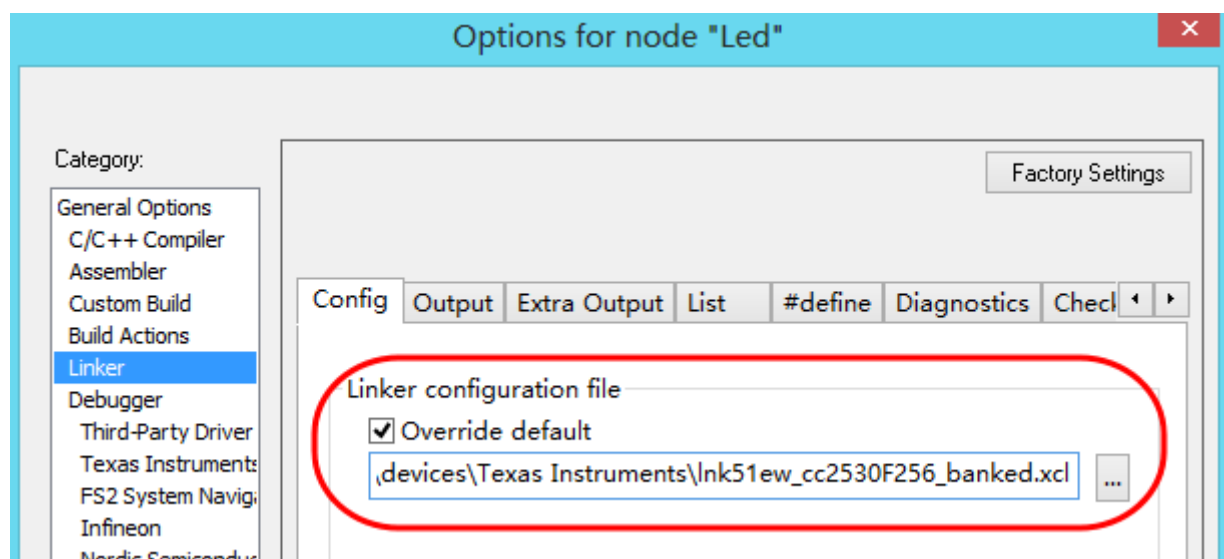
好了，一定要学会仿真调试哦，这是解决问题最常用的方法，也是嵌入式开发中最常用的仿真方法，大部分集成环境调试方法都大同小异的。以后的实验中就不重复讲仿真的方法，灵活使用仿真器会大大提高开发进度。如果你有两个仿真器并且 ID 不同时，可以在一台电脑同时跟踪两种设备，调试无线通讯中问题到底是协调器代码有问题，还是终端代码有问题。此时就能发现两个仿真器是如此强大，效率也大大提高。希望大家掌握！

编译常用错误如图所示：

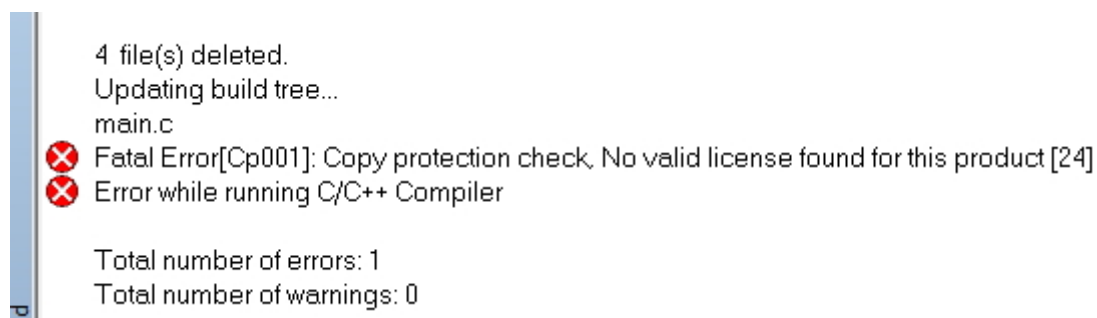
1.



一般是由于 IAR 没有装在 C 盘造成的，修改工程 Linker 项即可，以后每个实验都要修改哦，还是安装在默认路径省事。



2.



软件没有注册，请重新安装 IAR，在第 5.6 步要仔细看。

实验常见问题

问题 1：为什么要用 P1_0 呢，这是硬件决定的，如果想用其它 IO 引脚就要自己跳线了

问题 2：为什么用以下赋值，直接写等于不是更简单？

P1DIR |= 0x01; //P1.0 定义为输出口

P1SEL &=~0x01;

嵌入式中位运算只修改要修改的位，不要影响到其它位，直接等于同时也改变了其它位的状态哦，在基础实验功能单一，感觉不出来，如果在协议栈中就有严重问题了；并且用上述写法一眼就知道操作的位，易懂。