
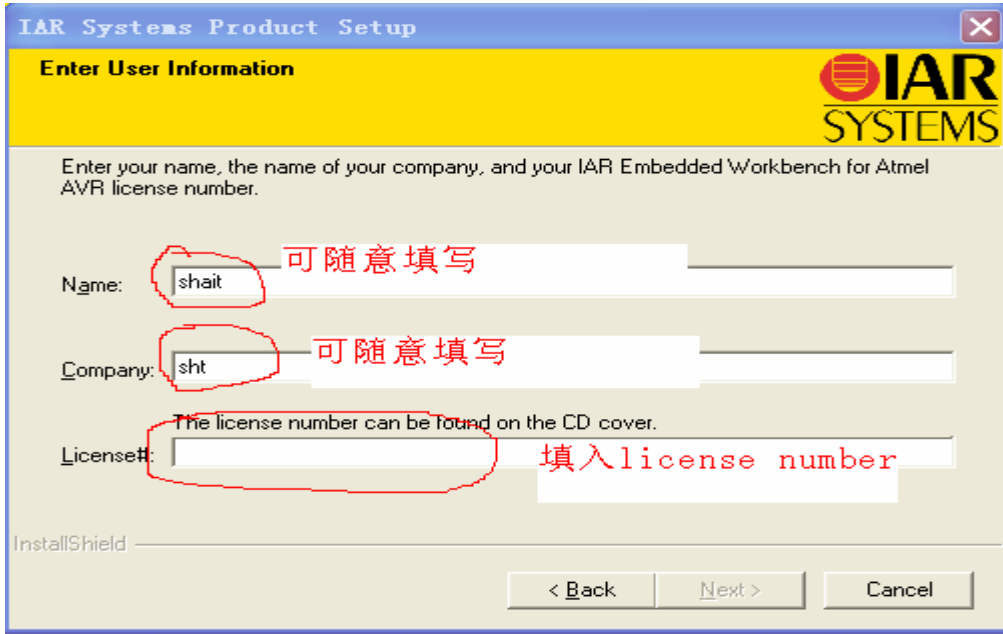


IAR for AVR 安装及入门使用手册

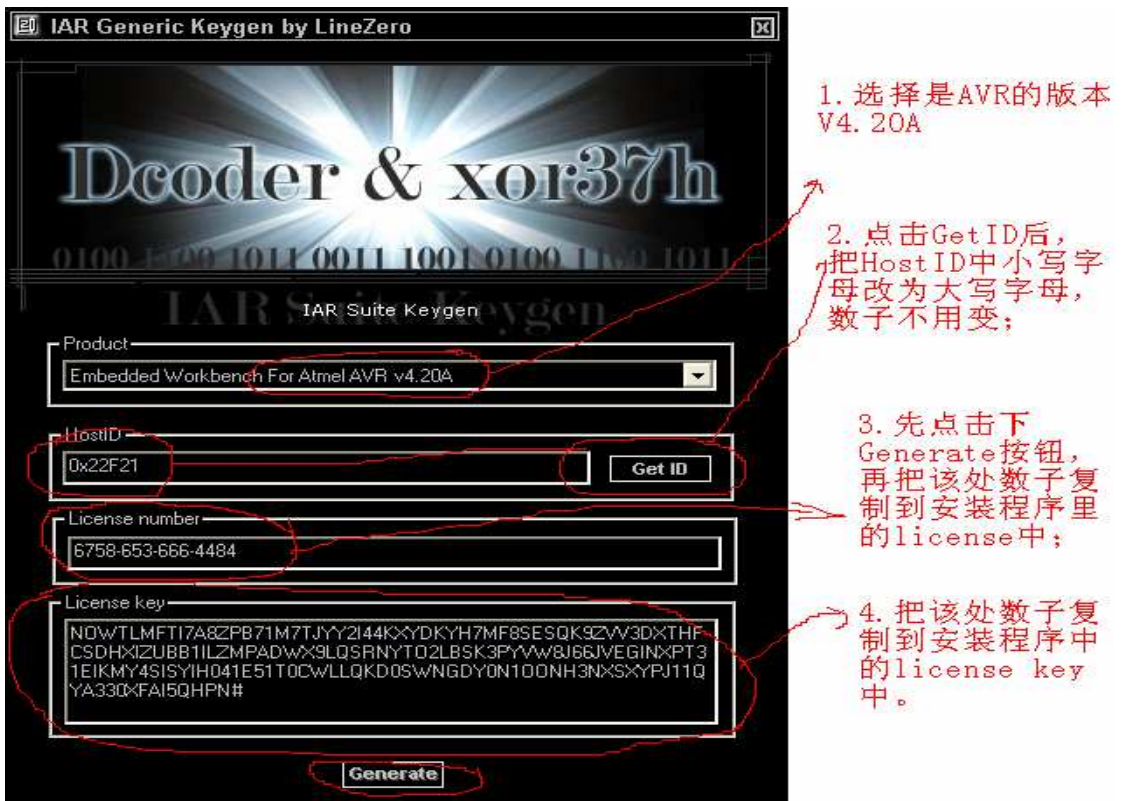
一、安装 IAR for AVR 4.20A 版本

- 1、点击  EWA VR-EV-420A.exe Evaluation versi... IAR Systems 程序，开始安装程序；
- 2、一路点击“next”，直到出现下图(红笔处为说明，以下同)：



- 3、点击安装破解程序  i arkg.exe ;

- 4、出现界面时选择参数如下图：



- 5、把上图的 license number 中的数字复制到下图中，后点击下一步；

The license number can be found on the CD cover.

License#:

- 6、接着出现要求输入 license key 的图，如下图，把第 4 步中生成的 license key 复制到下图的 license key 处；

IAR Systems Product Setup

Enter License Key

The license key can be either your QuickStart key or your permanent key.
If you enter the QuickStart key (found on the CD cover), you have 30 days to try the product out.
If you have received the permanent key via email, you paste it into the License Key textbox.

License #:

License Key:

Read License Key From File

C:

InstallShield

< Back Next > Cancel

- 7、然后一路按‘next’，直到安装完成。点击完成按钮后，你已成功安装了 IAR for AVR 的编程软件，你可以在 windows 的开始菜单中找到该软件，发送一个快捷方式到桌

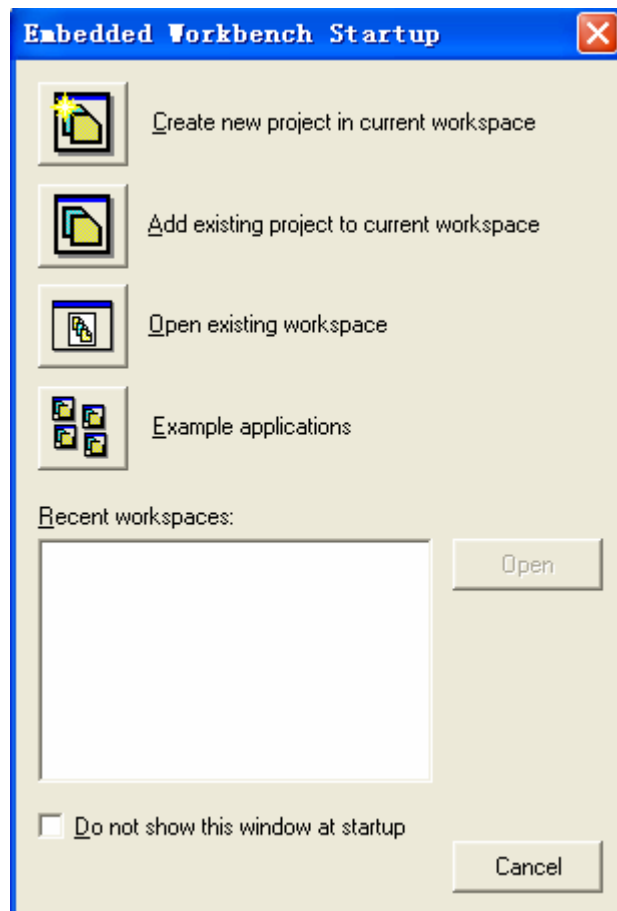


面。下一步将可以在 IAR 上编写程序了。

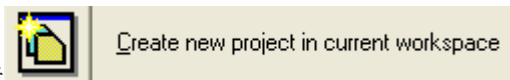
二、IAR for AVR 使用入门



1、 点击图标

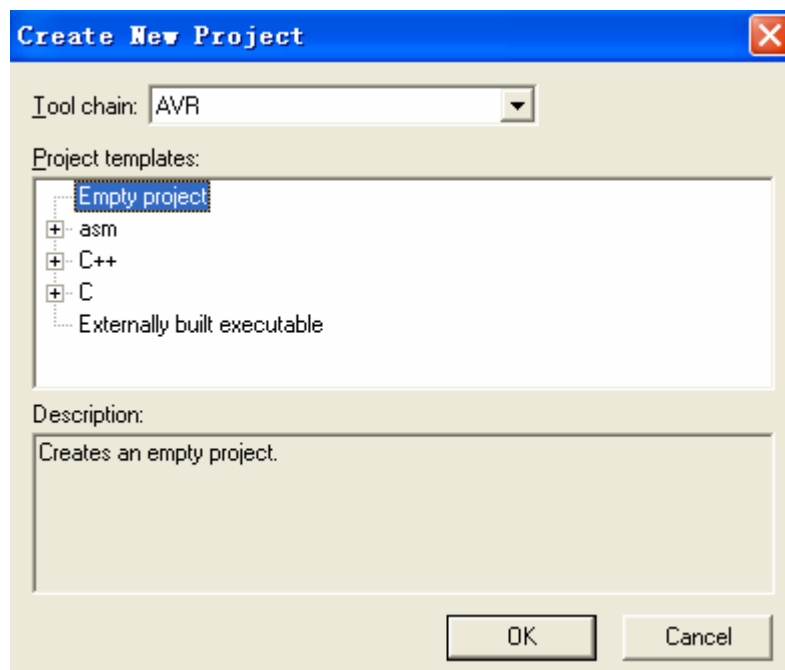


2、 这里我们要新建一个工程，点击

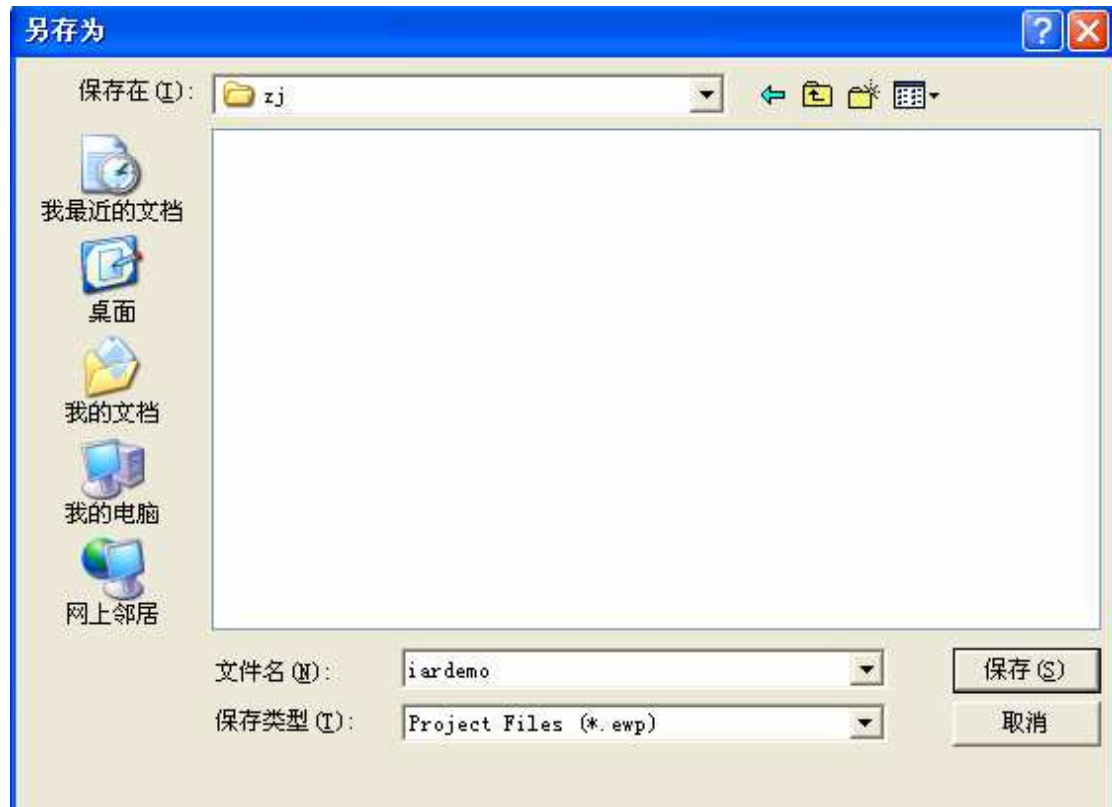


图标，出现下

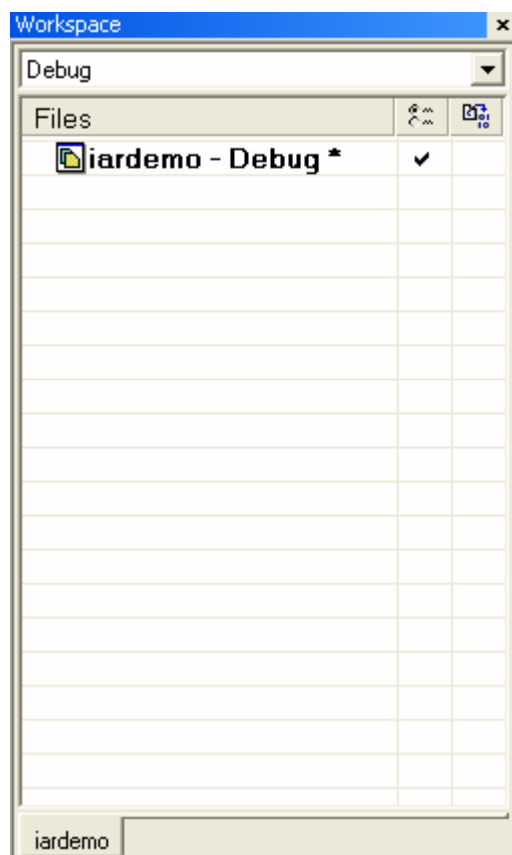
图：




- 3、在 Tool chain 中选择 AVR，Project templates 中选 Empty project，创建一个空的工程；我们命名为 iardemo，保存该工程；



- 4、OK，我们已创建了一个工程，如图：



- 5、我们装要向该工程添加程序文件，首先我们编写一个程序；点击左上角图标 ，在右边处将出现一个新的程序编写页面，写上如下程序：

```

/*包含头文件，该文件是编译器自带的，这个头文件功能是根据所选择的芯片型号，
自动选择相应的寄存器地址定义头文件
*/
#include <ioavr.h>

/* 声明延时函数，函数一定要先声明再调用的*/
void delay(unsigned int time);

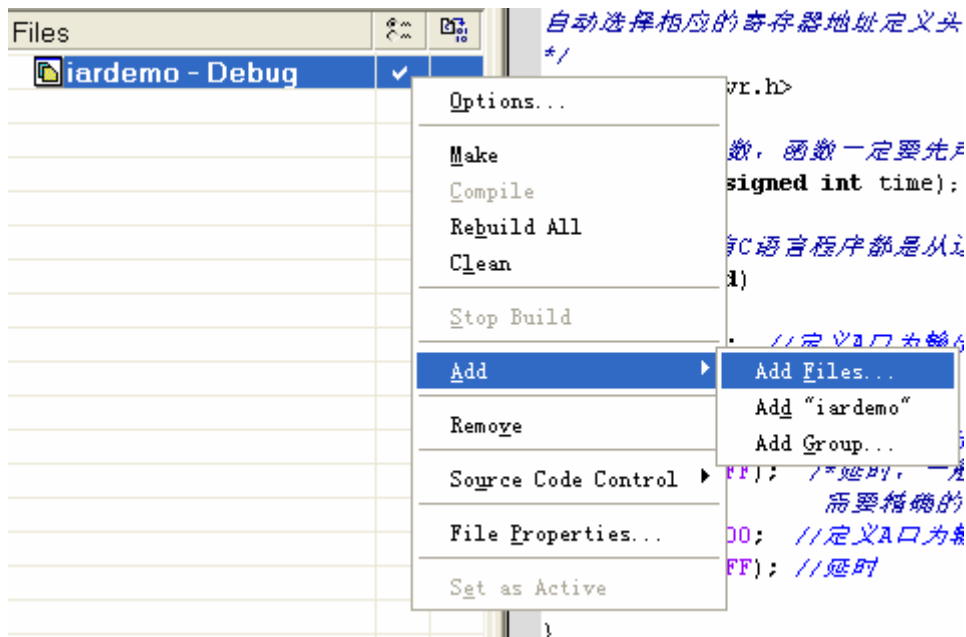
/*主函数，所有 C 语言程序都是从这个 main()函数开始运行的*/
void main(void)
{
    DDRA = 0xFF; //定义 PA 口为输出口；
    while(1) //进入死循环，这是前后面的常规写法
    {
        PORTA = 0xFF; //定义 PA 口输出高电平
        delay(0x07FF); /*延时，一般 C 语言的延时不是很精确的，
                        需要精确的延时时，要用定时器延时*/
        PORTA = 0x00; //定义 PA 口为输出低电平
        delay(0x07FF); //延时
    }
}

/*延时函数*/
void delay(unsigned int time)
{
    while(time--) /*当 time 的值为真时，进行循环，每次运行后 time-1，
                  直到 time=0 时，退出该函数*/
    {
        ;
    }
}

```

编写完成后，保存文件，取名为 iardemo.c

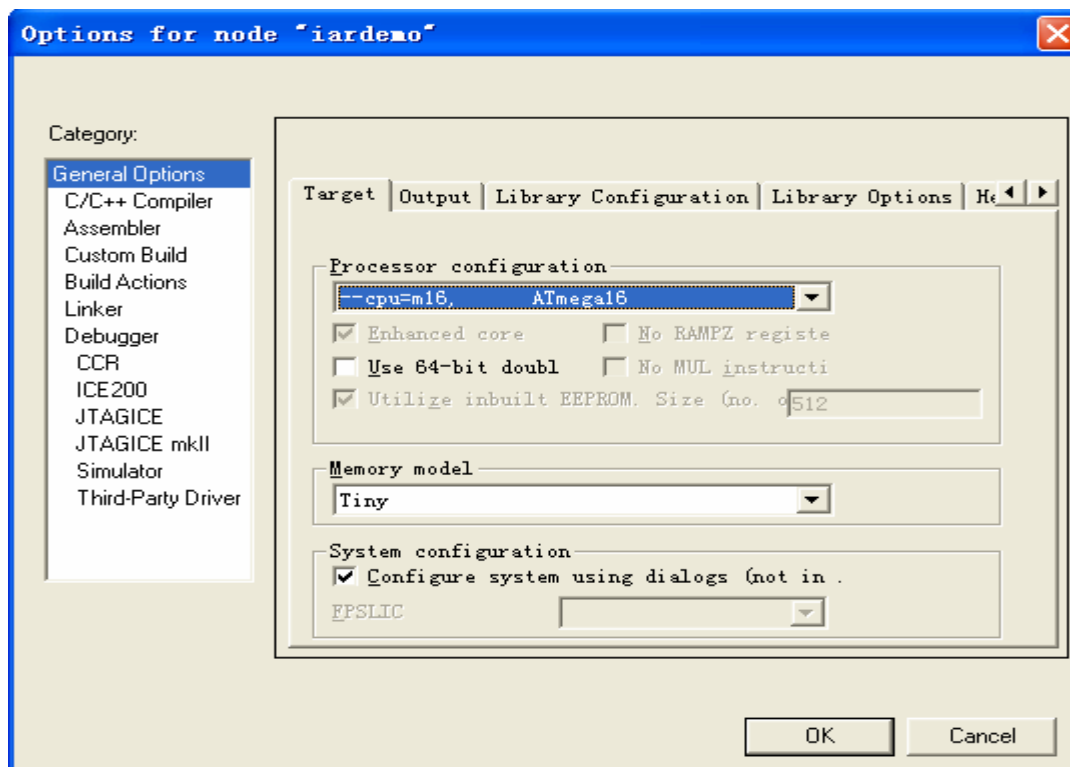
- 6、为刚才建立的工程加入程序文件，把鼠标移到 **Workspace** 框里，点击右键,在出现的菜单中选择 **add**,再选择 **add files**,如下图所示，在弹出的添加文件框里选择刚才编号的 iardemo.c 文件；



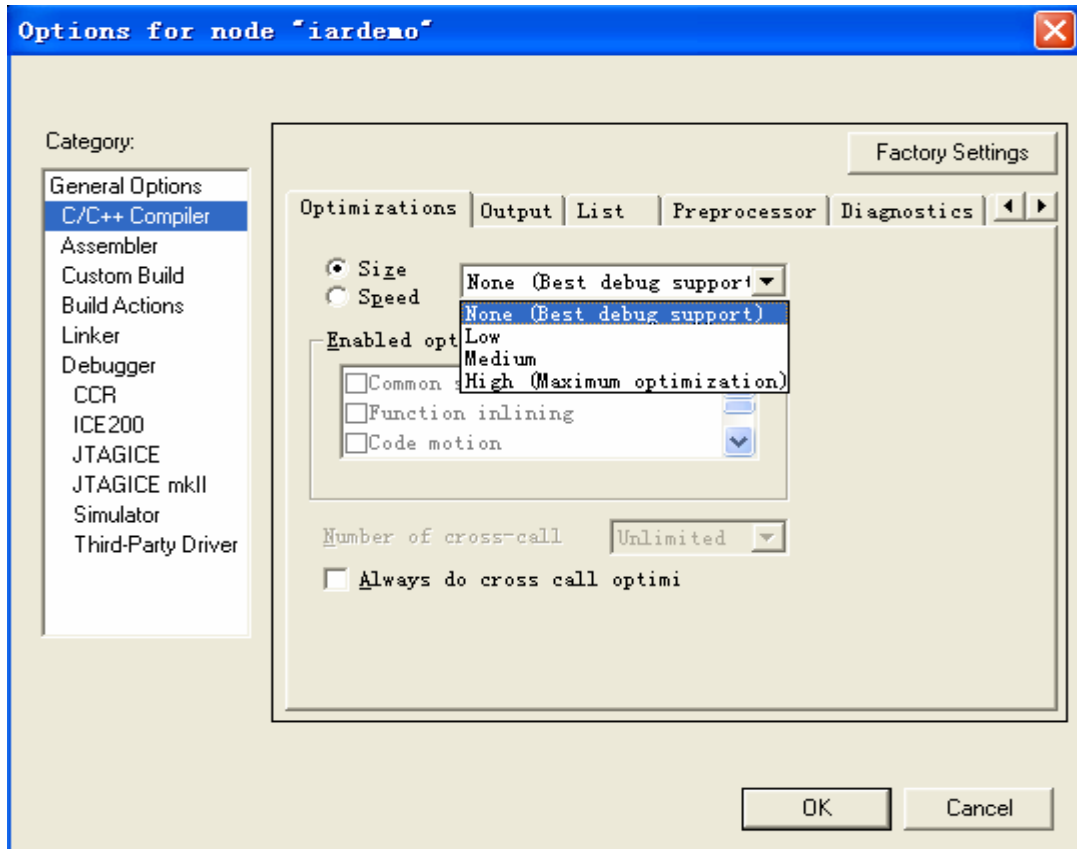
7、OK, 添加文件后, 可以看到如下图的情况; 多了 iardemo.c 和一个 Output 文件, iardemo.c 是我们刚才编号的文件, Output 是用于放置编译后的结果文件的;



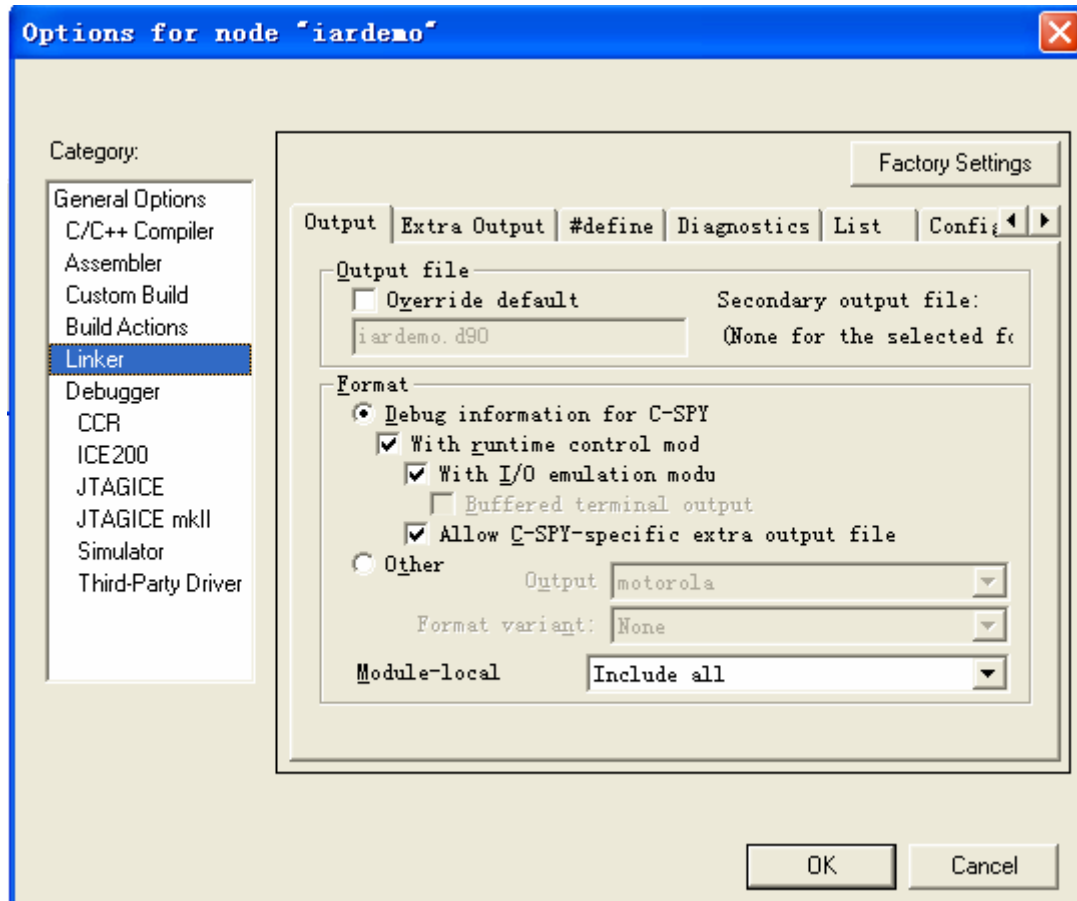
8、接下来我们设置参数, 把鼠标移到 Workspace 框里, 点击右键, 在出现的菜单中选择 Options..., 弹出如下图, 在左边 Category 中选择 General Options 项, 后在右边 Processor configuration 中选择 --cpu=m16, Atmega16;



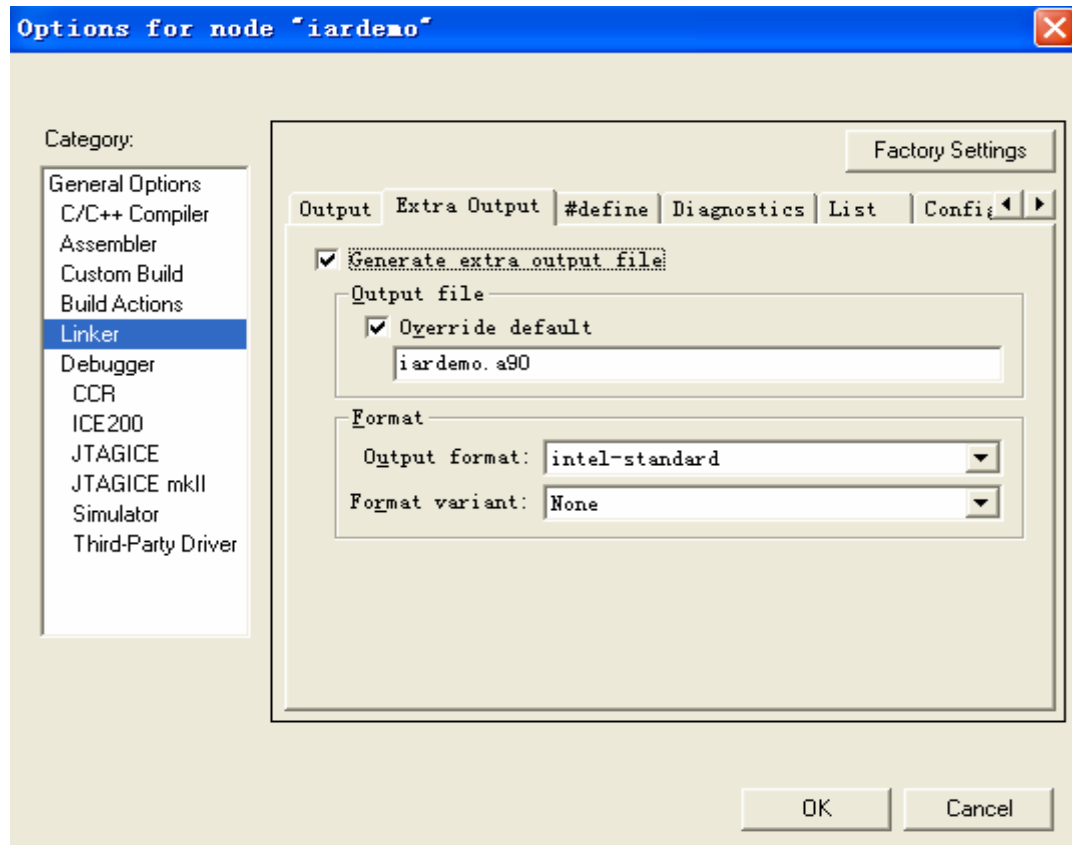
9、在左边 Category 中选择 C/C++ Compiler 项,后在右边 Optimizations 中选择 None(Best debug support),即选择不要代码压缩功能;



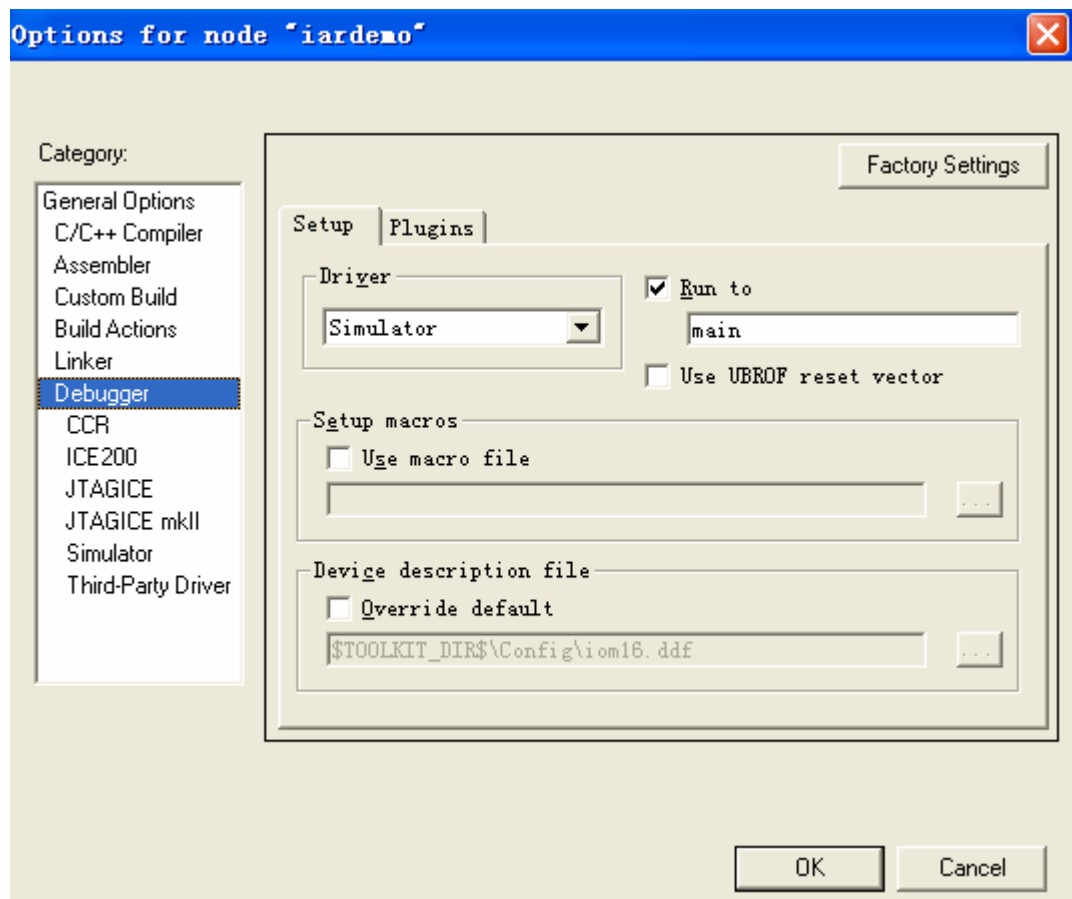
10、在左边 Category 中选择 Linker 项,后在右边 Output 中选择如下图的参数,该设置是生成可以仿真用的文件;





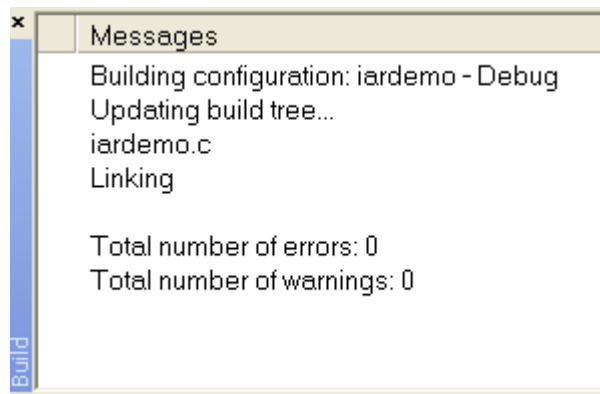
11、在左边 Category 中选择 Linker 项，后在右边 Extra Output 中选择如下图的参数，该设置是生成可以烧写用的文件；



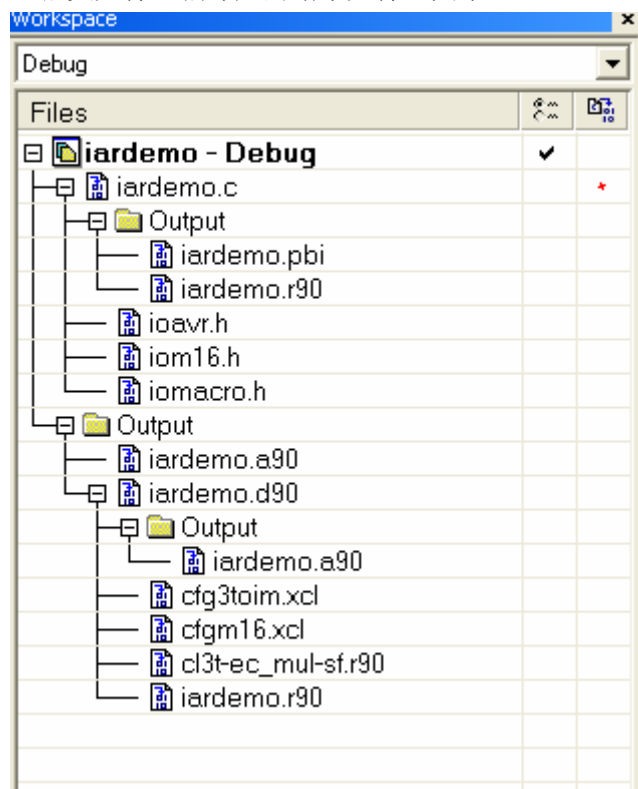
12、在左边 Category 中选择 Debugger 项，后在右边 Setup 中选择如下图的参数，该设置是使用软件仿真 Simulator，当你只想用硬件调试时，可以选择 JTAGICE 等 JTAG 仿真器进行与硬件连接调试用；




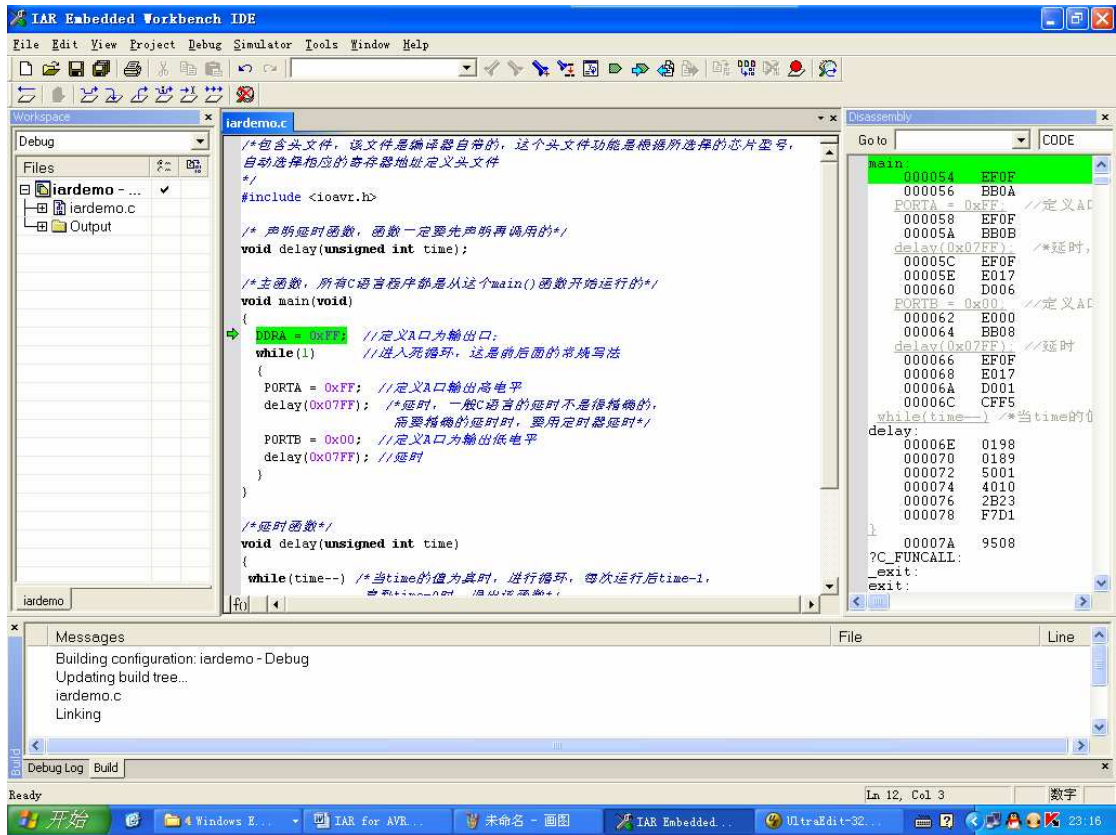
13、可以编译了，按一下工具条的  图标，因为第一次编译，所以需要先保存一下 Workspace，这里输入名 `iardemo`。再按一下  编译，在 Messages 中出现下图，表示成功编译了。



14、这时你打开 workspace 中的+号，会看到里面多了好多文件，这些都是编译时连接文件，包括头文件，编译后的结果文件，其中 `iardemo.a90` 为烧写文件，`iardemo.d90` 不仿真文件；



15、点击工具条上的  图标，即可以进入仿真界面，可以使用 Debug 下的菜单快捷键进行调试了。



你已学会使用 IAR for AVR 了，更详细的使用，可以看帮助文件或相应的使用手册。