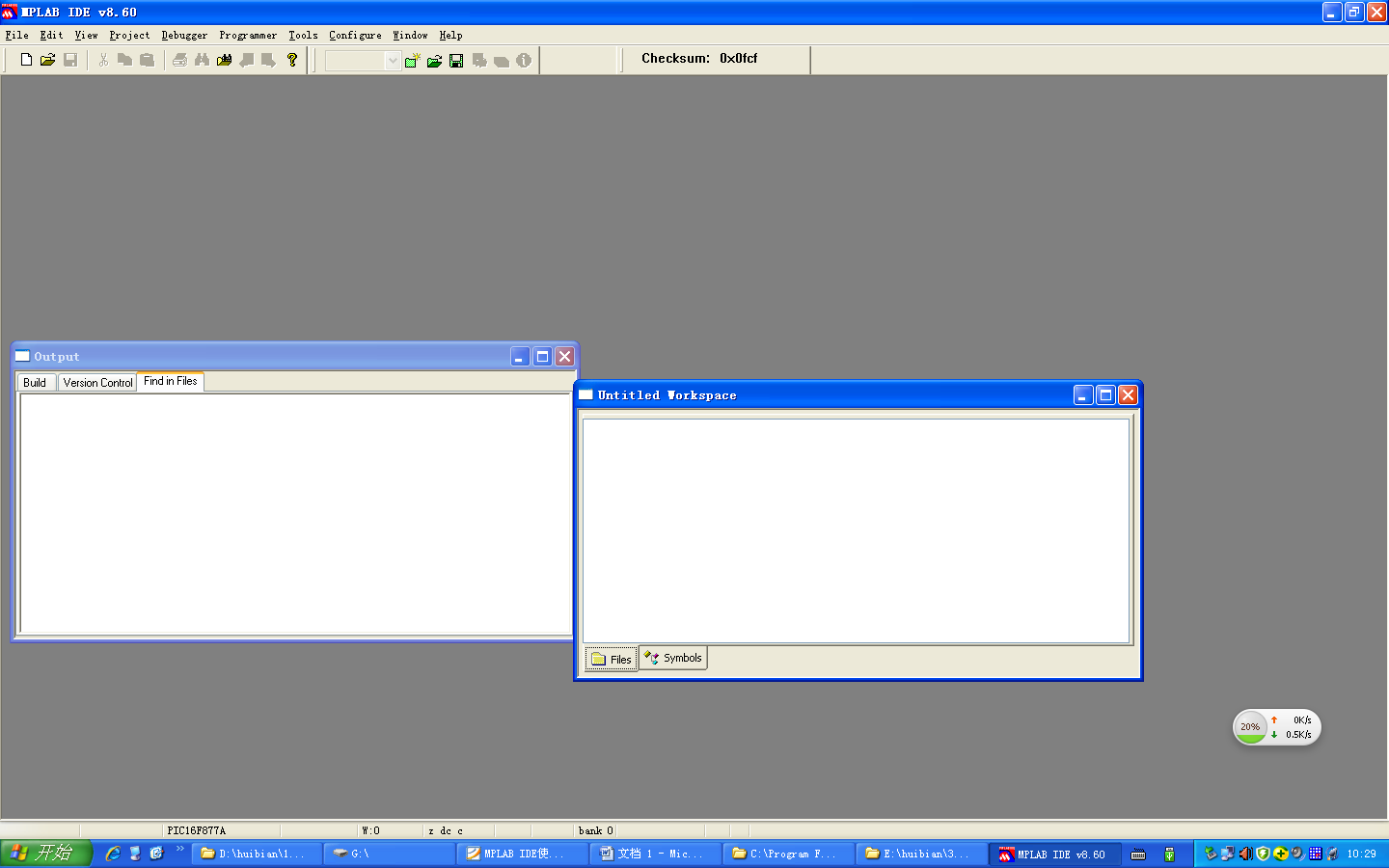
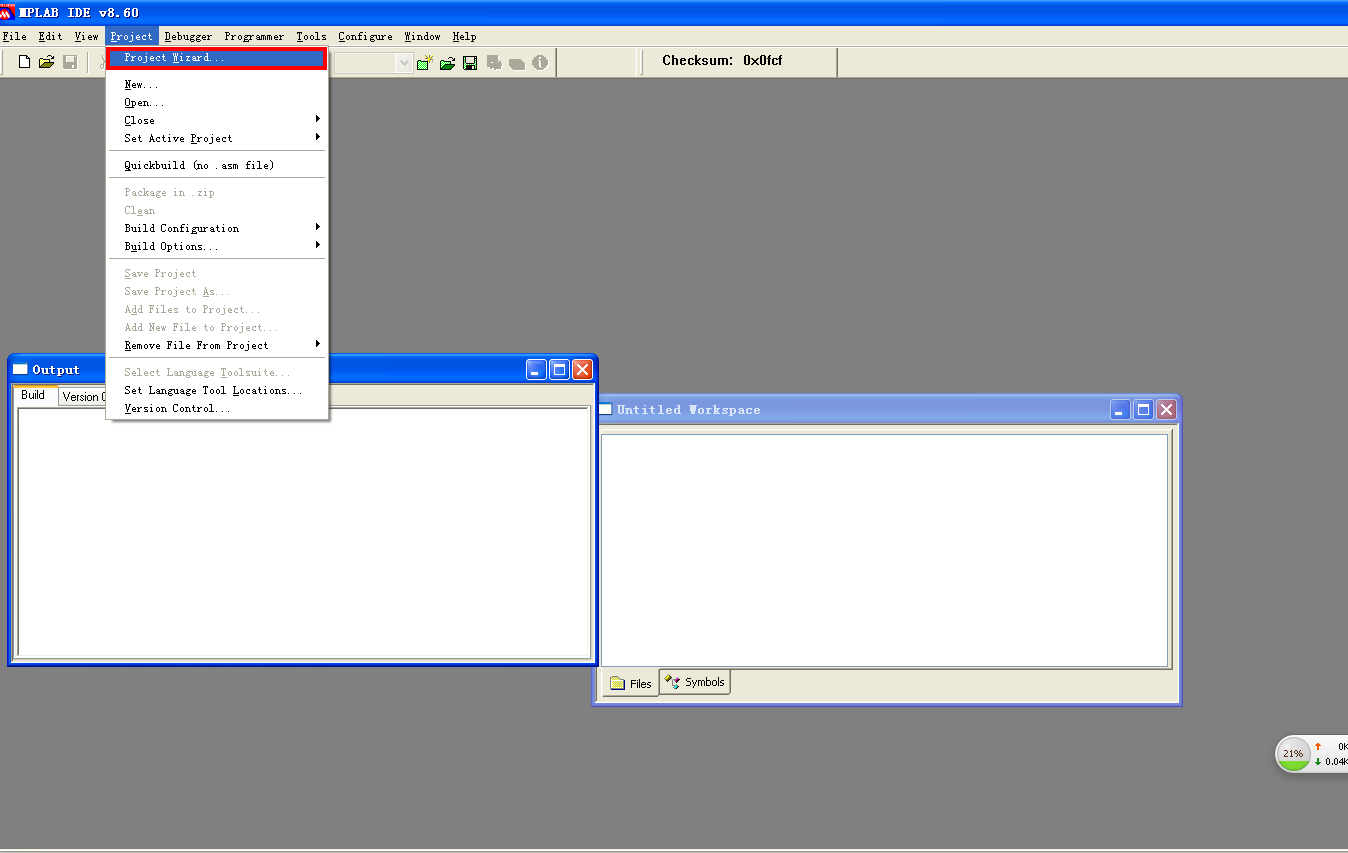
**MPLAB IDE V8.60 使用教程**

**工程建立和编译**

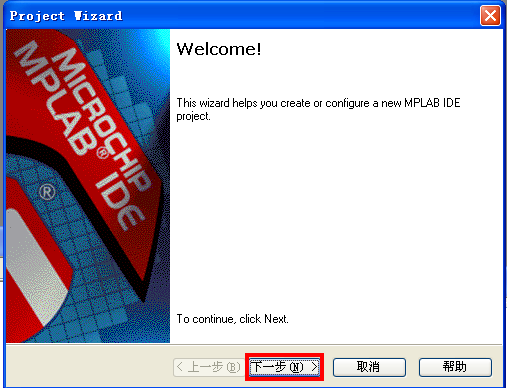
1. **首先找到自己桌面上的MPLAB IDE V8.60快捷图标****并双击，打开后如图所示**

****

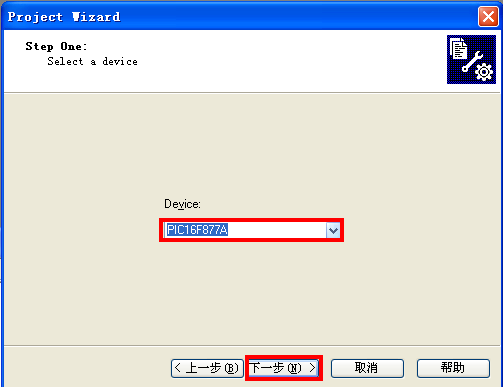
1. **选择Project->Project Wizard菜单项，利用安装向导来完成整个项目的建立**

****

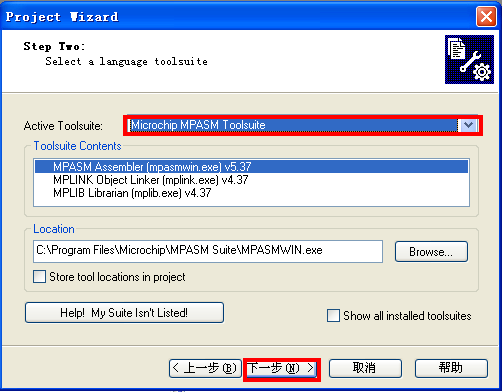
1. **出现安装向导界面，点击下一步（N）**

****

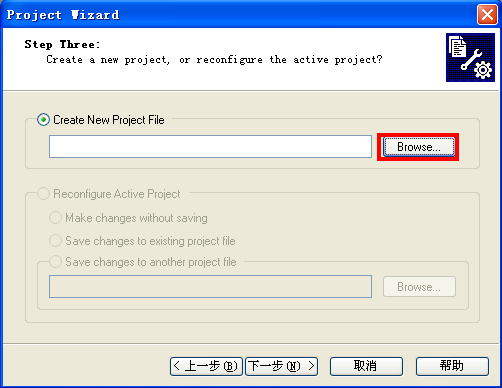
1. **出现第一步（Step One），选择一款芯片（Select a Device），在芯片列表中我们选择PIC16F877A，然后点击下一步（N）**

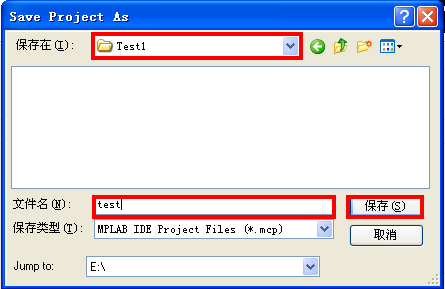
****

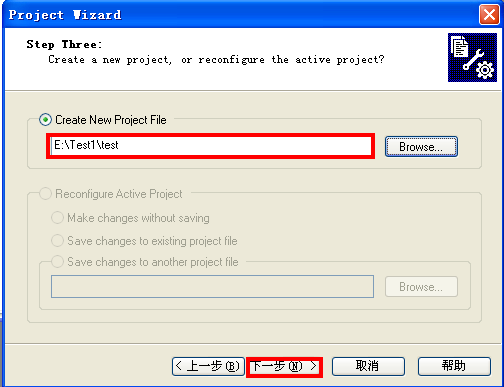
1. **进入下一步（Step Two），选择编译语言（Select a language toolsuite）,由于我们编程使用汇编语言，这里我们在Active Toolsuite中选择Microchip MPASM Toolsuite，点击下一步（N）**

****

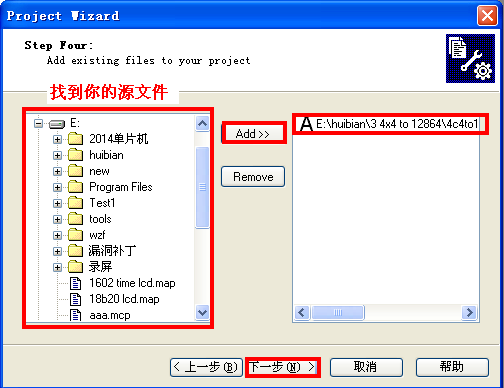
1. **进入第三步（Step Three），建立工程并选择工程存放路径，找到自己存放的路径，并为工程起名，点击下一步（N）（注意路径应为英文路径，不能有中文），点击保存（S），点击下一步（N）**

****

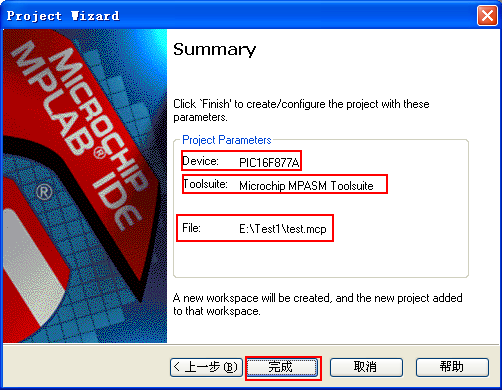
****

****

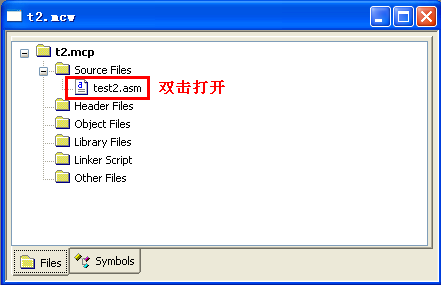
1. **第四步（Step four），添加已经存在的源文件到工程中（Add existing files to your project）,如果有已经写好的源文件，请在左侧的文件列表中找到，并点击ADD添加，添加后右侧会出现该文件的路径和名称，点击下一步（N），如果还没有写源程序，则直接点击下一步（N）（如果未写源程序，则不需要添加（ADD），直接点击下一步）**

****

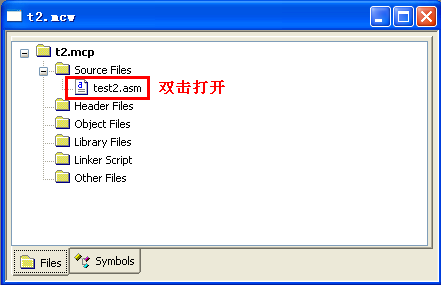
1. **进入最后出现的工程概况，包括芯片型号，编译器类型，文件存放路径，点击完成**

****

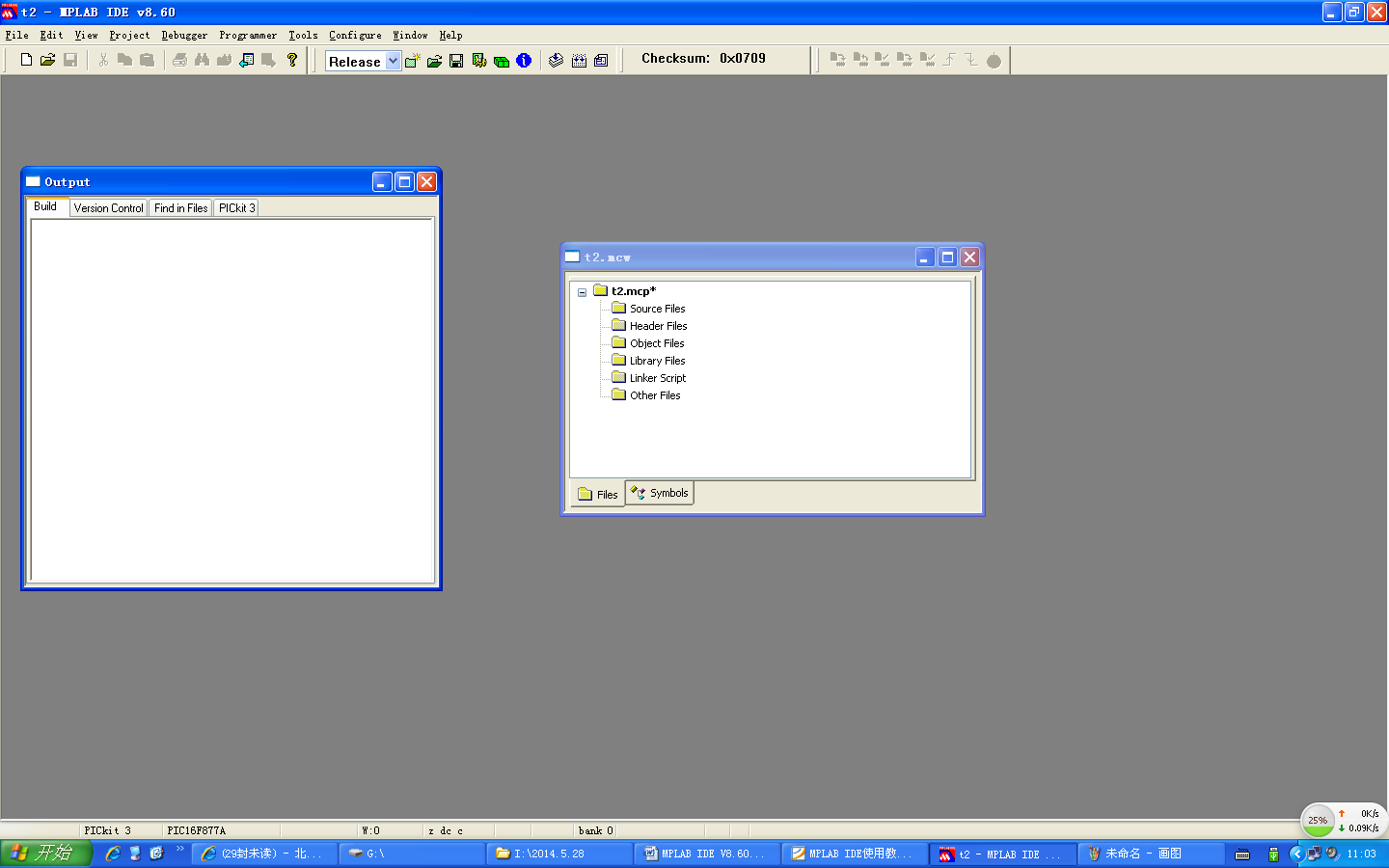
1. **如果第四步（Step four）时已经添加（Add）源文件到工程里，点击完成后会看到源文件已经被添加进去了，双击文件名可以打开源文件**



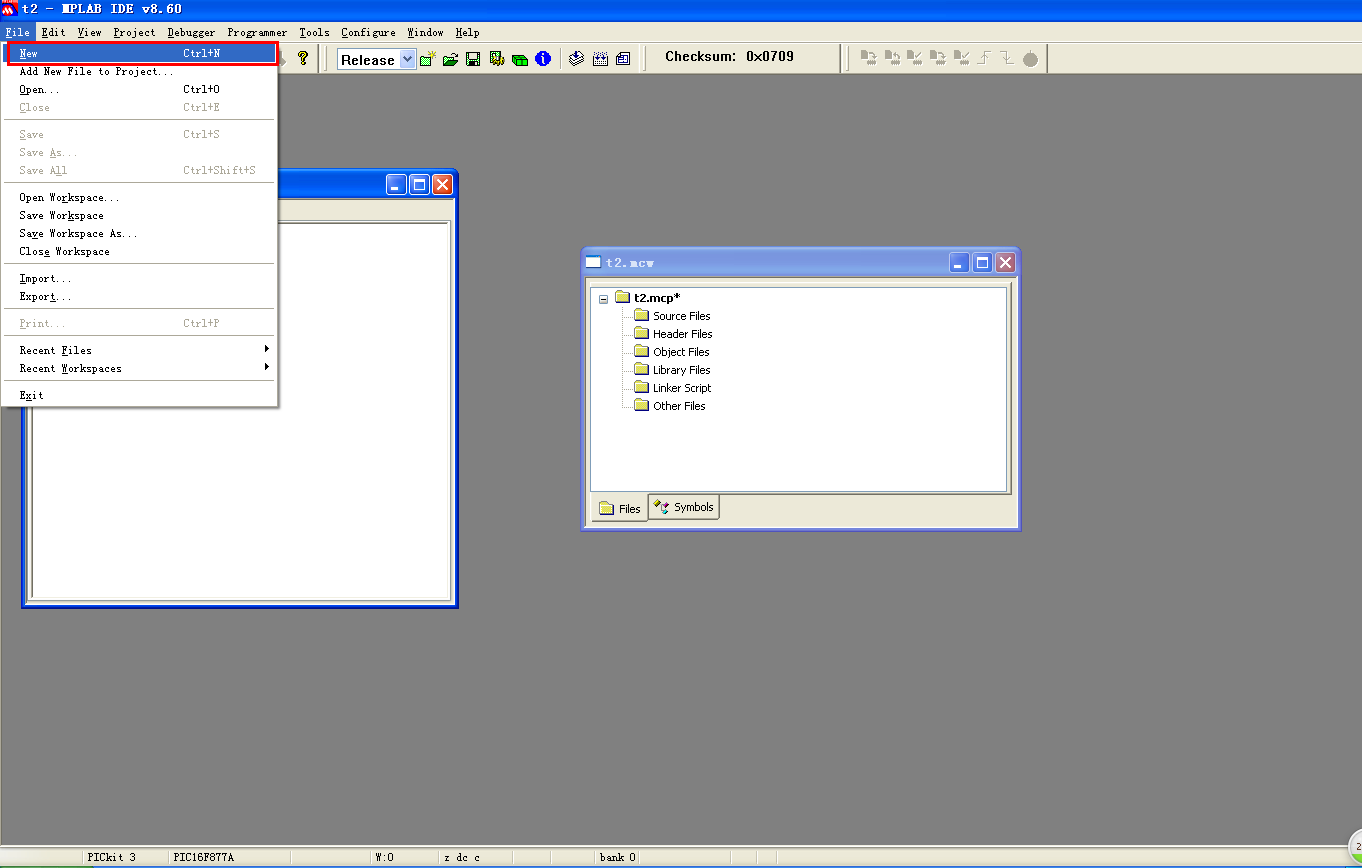
**如果第四步（Step four）时没有添加（Add）源文件到工程里，下面进入源文件建立过程**

****

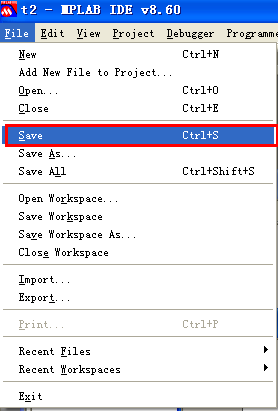
**如果第四步（Step four）时没有添加（Add）源文件到工程里，下面进入源文件建立过程**

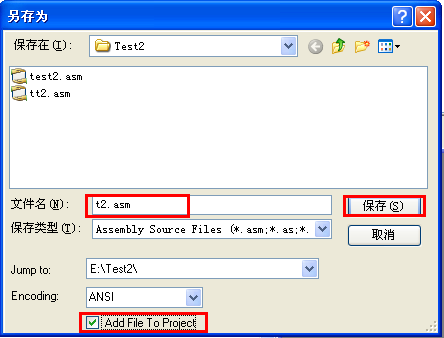
****

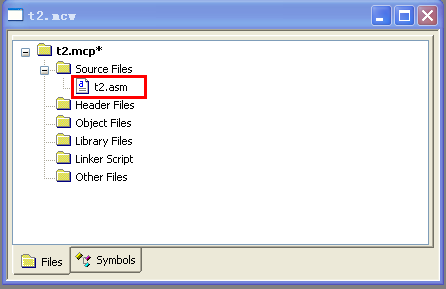
1. **点击File->New或者使用快捷图标开始编写源程序**

****

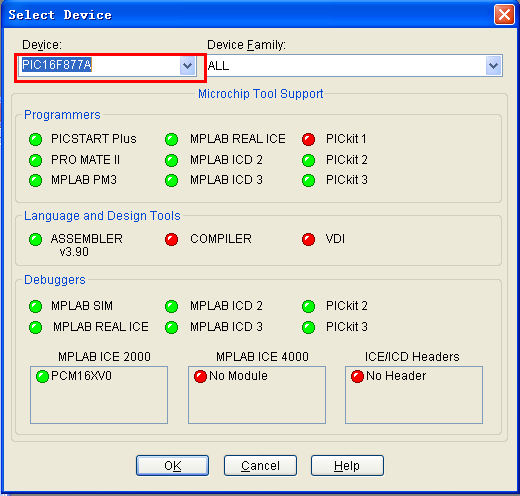
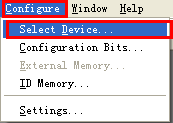
1. **出现编写源文件界面，把自己的源文件写入，选择File->Save或者直接点击快捷图标保存，文件后缀为.asm ,并选择添加文件到工程（Add File to Project），点击保存（S），工程中将会出现你的源文件名称**

****

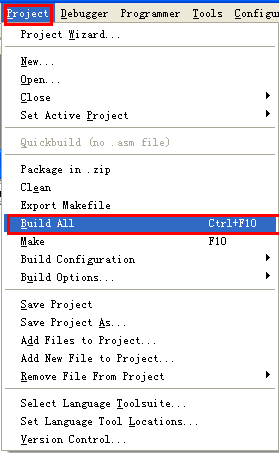
****

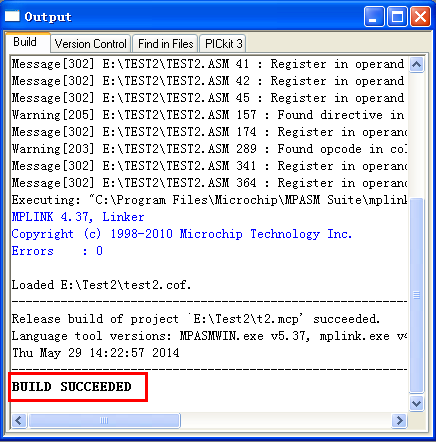
****

1. **如果想要更改芯片型号，可以选择Configure->Select Device更改**

****

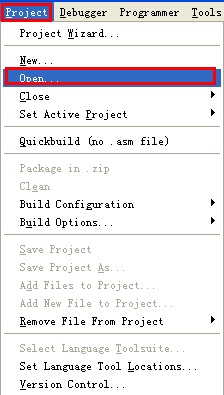
1. **然后选择Project->Build All来进行编译**

****

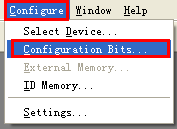
1. **如果你的程序没有问题，将会出现编译成功信息，有问题查找错误，修改后再编译，直到正确为止。**

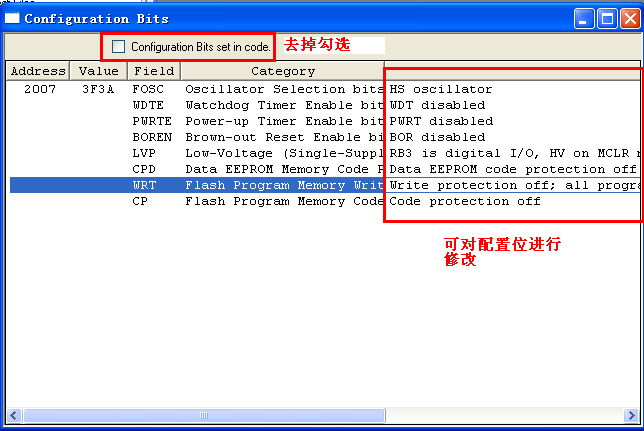
**在线仿真**

1. **首先打开已经建立的工程**

****

1. **选择Configure-> Configuration Bits来更改配置位，改变Configuration Bits set in code 的勾选状态后，才可以对配置位进行修改。**

****

****

**在线调试状态下要做如下设置, 设置好后勾选Configuration Bits set in code进行保存：**

**FOSC（振荡模式）设置为HS oscillator**

**WDTE（看门狗定时器）设置为Disable**

**PWRTE（上电延时定时器）设置为Disable**

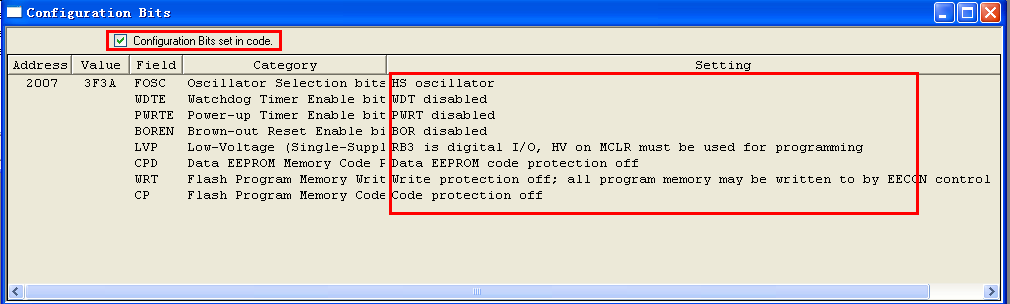
**BOREN（电源跌落检测）设置为Disable**

**LVP（低电压烧写编程）设置为 HV on MCLR must be used**

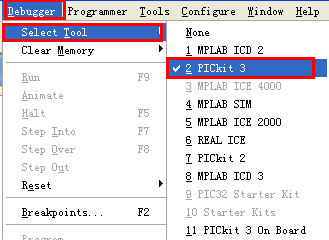
**CPD（保护EEPROM的内容）设置为OFF**

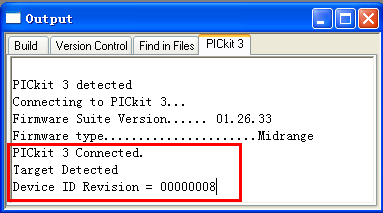
**WRT（FLASH存储器烧写）设置为OFF**

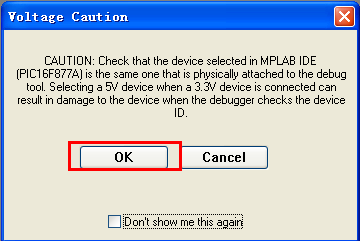
**CP（保护FLASH内的代码）设置为OFF**

****

1. **然后选择Debugger->Select Tool->PICkit 3,如果成功连接PICkit3，Output窗口会出现连接成功提示。如果出现Voltage Caution窗口，点击OK。**

****

****

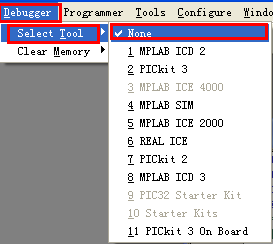
****

1. **然后选择Debugger->Program，将源程序下载到PICkit3仿真器上，选择Debugger->Run或者图标来在线运行程序。**

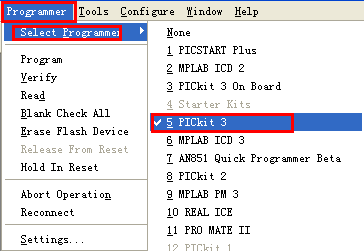
****

**脱机烧写**

1. **首先选择Debugger->Select Tool->None**

****

1. **选择Programmer->Select Programmer->PICkit3**

****

1. **然后选择Programmer->Program，将源文件烧写到芯片上。**

****

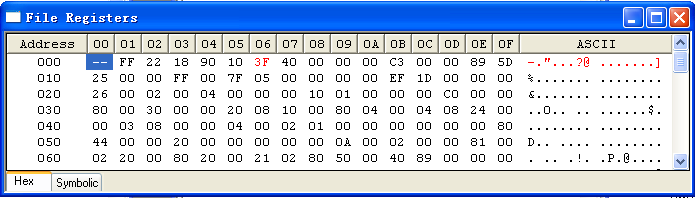
**程序调试**

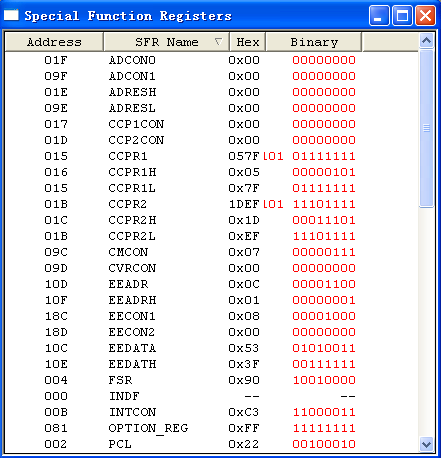
1. **连续运行方式**

**首先选择Debugger->Run或者按动F9键或者点击图标按钮，均可令程序进入连续运行状态。终止程序运行需要选择Debugger->Halt或者按动F5键或者点击按钮。**

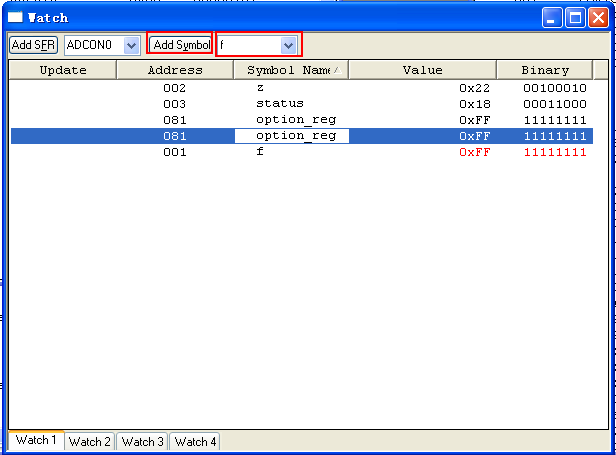
1. **设置观察窗口**

**打开文件寄存器窗口：选择View->File Registers，该窗口可以看到指定单片机的全部RAM数据寄存器的内容（十六进制），以每行16个单元排列。**

**打开特殊功能寄存器窗口：选择View->Special Function Registers，该窗口中以列表形式显示寄存器名称和寄存器内容。寄存器内容以四种格式同时显示：十六进制、十进制、二进制和ASCII码。**



**打开观察寄存器变量窗口和选定被观察寄存器变量：选择View->Watch，出现一个Watch窗口，在下拉列表中选择要观察的变量名或者直接数入要观察的变量名，然后点击Add Symbol按钮，即可在Watch窗口中增加一个待观察的寄存器变量。可以用同样的方法在Watch窗口中添加多个寄存器变量。在该窗口可以查看到变量的物理地址、符号名和数值。**



1. **单步运行方式**

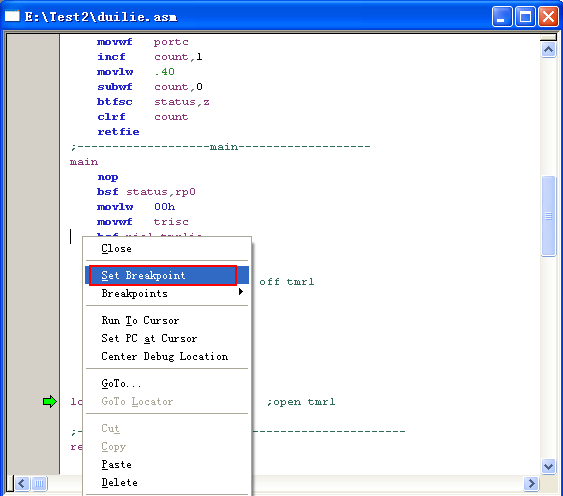
**首先选择Debugger->Reset->Prosessor Reset或者按动F6键或者点击图标按钮使单片机复位。然后选择Debugger->Step into或者按动F7键，均可令程序进入单步运行状态。一次次的点击单步运行按钮的同时，可以看到观察窗口中出现变红的寄存器或变量，并且其中的值会不断的更新。**

1. **自动单步运行方式**

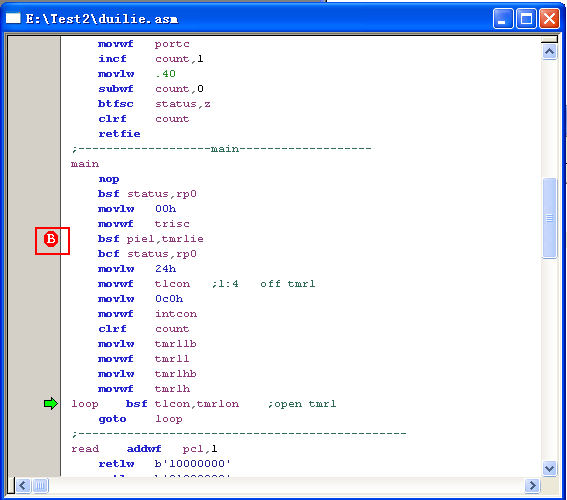
**选择Debugger->Animate或者点击图标按钮可令程序进入自动单步运行状态。**

1. **设置断点运行方式**

**首先确保当前窗口为源程序观察窗，再用鼠标右键点击该窗口中计划设置断点的源程序语句行。这时在弹出的菜单中选择Set Breakpoint选项即可。**

****

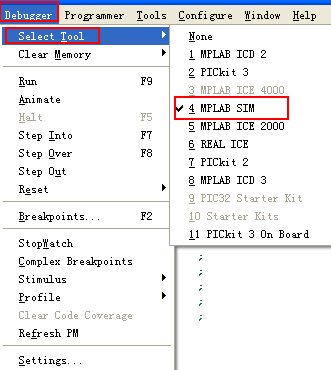
**在源窗口中设有断点的语句行前面会出现一个，以区别于其他语句行。用这种方法可以在同一个程序里设置多个断点。消除某一行语句的断点的操作方法是用右键点击该语句，在弹出的菜单中选择Remove Breakpoint即可**

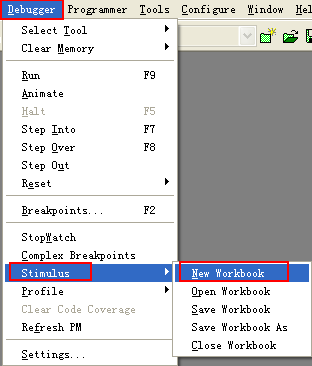
****

**引脚信号异步激励**

**对于MPLAB-SIM软件模拟器而言，当被调试的用户程序，在执行过程中，有时需要检测和接收单片机引脚上的输入信号。**

**首先选择Debugger->Select Tool->MPLAB SIM, 然后选择Debugger->Stimulus->New workbook**

****

****

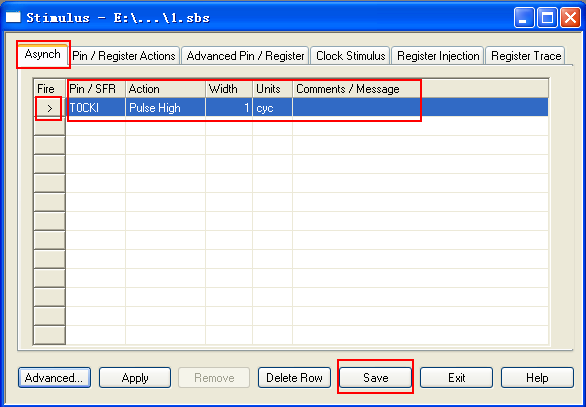
**这时会出现Stimulus窗口, 定义异步按键，完成后单击Save保存引脚激励信号。在程序运行时 点最左边的FIRE 下面的 “ >” 符号就可以了。**

**Pin/SFR选择引脚Rx/TxCLK**

**Action选择动作Plus/ High/ Low /Toggle**

**Width 选择宽度 十进制数值**

**Units单位cyc/ sec /ns/us/ms**

****

**跑表观察窗**

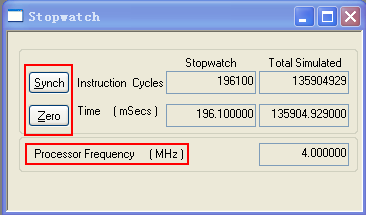
**可以利用该窗口来计算整个子程序的延迟时间。**

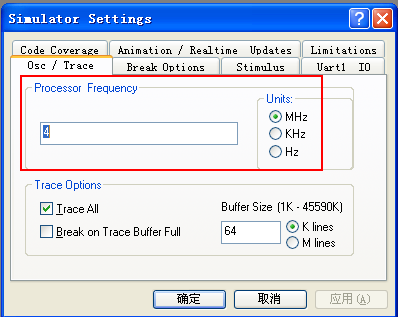
**选择Debugger->Stopwatch,会出现Stopwatch窗口。**

**Synch 将stopwatch的值与total simulated的值同步**

**Zero 将instruction cycles与time的值清零**

**Processor frequency的值需要在Debugger->Settings的Osc/Trace里面修改。**

****

****