



# **MPLAB<sup>®</sup> IDE**

## **快速入门指南**

---

---

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

---

提供本文档的中文版本仅为为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、Accuron、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PRO MATE、rPIC 和 SmartShunt 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

AmpLab、FilterLab、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL、SmartSensor 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、In-Circuit Serial Programming、ICSP、ICEPIC、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、PICkit、PICDEM、PICDEM.net、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、REAL ICE、rLAB、Select Mode、Total Endurance、UNI/O、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2008, Microchip Technology Inc. 版权所有。

QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
== ISO/TS 16949:2002 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

---

---

## 目录

---

---

### 第 1 章 MPLAB® IDE 简介

1.1 嵌入式系统简介 .....	1
1.2 开发周期 .....	6
1.3 项目管理器 .....	7
1.4 语言工具 .....	8
1.5 目标调试 .....	9
1.6 对器件编程 .....	10
1.7 MPLAB IDE 的组件 .....	10
1.8 MPLAB IDE 文档 .....	11
1.9 MPLAB IDE 在线帮助 .....	11
1.10 MPLAB IDE 更新和版本编号 .....	14

### 第 2 章 MPLAB IDE 基础教程

2.1 简介 .....	15
2.2 MPLAB IDE 的功能和安装 .....	16
2.3 教程概述 .....	18
2.4 选择器件 .....	19
2.5 创建项目 .....	20
2.6 设置语言工具 .....	21
2.7 为项目命名 .....	22
2.8 将文件添加到项目 .....	23
2.9 编译项目 .....	26
2.10 创建代码 .....	27
2.11 再次编译项目 .....	29
2.12 使用软件模拟器测试代码 .....	30
2.13 教程总结 .....	37

全球销售及服务网点 .....	40
-----------------	----

注：

---

## 第 1 章 MPLAB® IDE 简介

---

### 1.1 嵌入式系统简介

MPLAB IDE 是一种在个人计算机（PC）上运行的 Windows® 操作系统（Operating System, OS）软件程序，用来为 Microchip 单片机和数字信号控制器开发应用程序。由于它提供了一种单一的集成“环境”为嵌入式单片机开发代码，因此被称为集成开发环境或 IDE。经验丰富的嵌入式系统设计人员可以跳过前面章节转至第 1.7 节“MPLAB IDE 的组件”。在此还建议阅读第 1.9 节“MPLAB IDE 在线帮助”和第 1.10 节“MPLAB IDE 更新和版本编号”。本章的其余部分简要地介绍了嵌入式系统开发以及使用 MPLAB IDE 的方法。

#### 1.1.1 “嵌入式系统”说明

嵌入式系统通常为一种利用小型单片机（如 Microchip 的 PIC® MCU 或 dsPIC® 数字信号控制器（Digital Signal Controller, DSC））功能的设计。这些单片机将微处理器（类似于台式计算机中的 CPU）与某些称为“外设”的附加电路相结合，加上同一芯片上的其他电路就构成了一个需要极少其他外部器件的小型控制模块。这种单个器件可以被嵌入到其他电子和机械设备中，以实现低成本的数字控制。

#### 1.1.2 嵌入式控制器和 PC 之间的区别

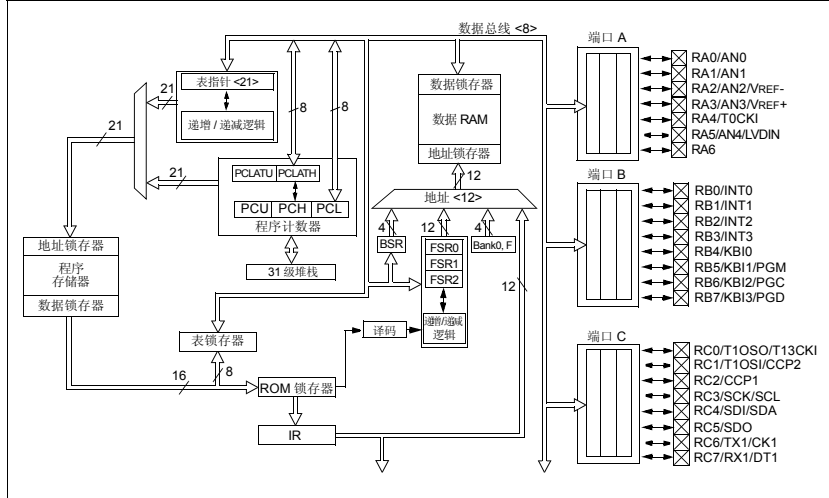
嵌入式控制器与 PC 之间的主要区别在于：嵌入式控制器专用于某项特定任务，或某些任务。而 PC 则设计为能够运行多种不同类型的程序，并能够连接到多种不同的外部设备。嵌入式控制器只有一个程序，因此成本低廉，只要能够保证处理专项任务所需的计算能力和硬件即可。而 PC 的核心具有比较昂贵的通用中央处理单元（CPU），它还包含了很多其他外部设备（内存、磁盘驱动器、视频控制器和网络接口电路等）。嵌入式系统采用低成本的微控制器单元（MCU）作为其智能核心，在同一芯片上还有很多外设电路而外部设备则相对较少。通常，嵌入式系统属于不可见部件，或者是其他产品的子模块，例如电池式钻孔机、冰箱或车库开门器。此类产品中的控制器只执行整个设备的一小部分功能。控制器给这些设备中的一些关键子系统增添了低成本的智能控制。

举例来说，烟雾检测器就是一种嵌入式系统。它的功能是检测传感器输出的信号，如果信号指示有烟雾存在，则发出警报。可以使烟雾检测器中的小程序执行无限循环，不停地对烟雾传感器输出的信号进行采样，也可以使烟雾检测器处于低功耗的“休眠”模式，由传感器的输出信号将其唤醒。唤醒后，烟雾检测器中的小程序就会发出警报。该程序可能还具有一些其他功能，例如用户测试功能和电池欠压报警。虽然配备传感器和音频输出设备的 PC 通过编程也可以实现上述功能，但这并不是一种节约成本的解决方案（而且它也不可能依靠 9 伏的电池供电、以无人照管方式常年运作！）。嵌入式设计使用廉价的单片机，为我们的日常生活环境提供智能化处理，如烟检测器、相机、手机、家用电器、汽车、智能卡以及安防系统。

## 1.1.3 单片机的组件

PIC MCU 具有用于储存固件或编码指令的程序存储器，以运行程序。它还具有“文件寄存器”存储器，用来储存程序计算所需的临时变量。单片机芯片本身还具有很多外设。其中某些外设被称为 I/O 端口。I/O 端口是单片机上的引脚，可以将其驱动为高电平或低电平以发送信号、闪烁指示灯或驱动扬声器——发送任何能够通过线路传输的信息。这些引脚通常为双向引脚，可以配置为输入，以使程序能够对外部开关或传感器作出响应，或是与某些外部设备通信。

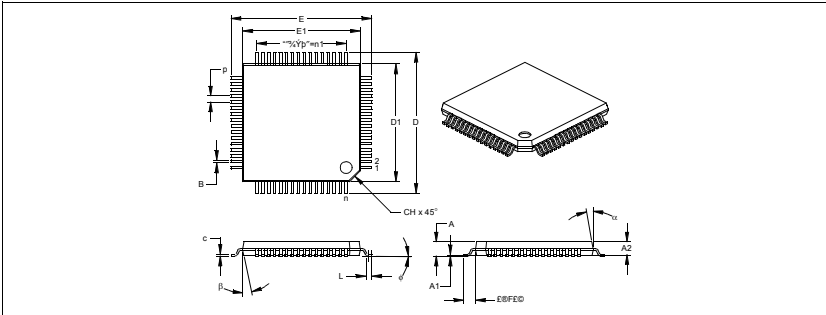
图 1-1: PIC® MCU 数据手册——框图（摘录）



为了设计这种系统，必须确定应用需要哪些外设。模数转换器（Analog-to-Digital Converters, ADC）允许单片机连接到传感器并接收变化的电平。串行通信外设可以通过几根线以串行方式与另一台单片机、局域网或因特网通信。PIC MCU 上称为“定时器”的外设可以精确地测量信号事件、生成和捕捉通信信号以及产生精确的波形，甚至可以在单片机挂起或因电源毛刺或硬件故障导致单片机工作异常时自动将其复位。其他外设可以检测外部电源是否降到了危险电平之下，以便让单片机能够及时存储重要信息，从而在完全掉电之前安全关闭。

应用使用哪种 PIC MCU 在很大程度上取决于应用运行程序所需的外设和存储器大小。其他因素可能包括单片机的功耗及其“外形因素”，即安装到目标设计中的物理封装尺寸和特性。

图 1-2: PIC® MCU 器件封装示例



1.1.4 使用 MPLAB IDE 实现嵌入式系统设计

嵌入式控制器开发系统是在台式计算机上运行的程序系统，它有助于编写、编辑、调试程序代码（嵌入式系统应用的灵魂），并将其烧写到单片机中。MPLAB IDE 在 PC 上运行，包含设计和部署嵌入式系统应用所需的全部组件。

开发嵌入式控制器应用的典型任务有：

1. 创建高端设计。根据所需的功能和性能，决定最适用于应用的 PIC MCU 或 dsPIC DSC 器件，然后设计相关的硬件电路。在决定由哪些外设和引脚控制硬件之后，编写固件（控制嵌入式应用中硬件的软件）。可以使用语言工具（如汇编器（可以直接将汇编代码转换为机器码）或编译器（允许使用更通用的语言创建程序））编写和编辑代码。汇编器和编译器允许使用功能标号来标识代码程序，变量名可与其用途相关联，并采用有助于在可维护的结构中组织代码的程序结构，从而使代码易于理解。

图 1-3: PIC® MCU 数据手册——时序（摘录）

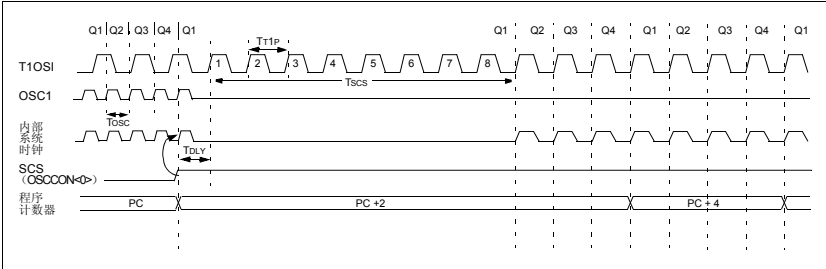


图 1-4: PIC® MCU 数据手册——指令（摘录）

RRNCF

Rotate Right f (no carry)

Syntax:

[label] RRNCF f[,a][,n]

Operands:

0 ≤ f < 255  
d = [0,1]  
a = [0,1]

Operation:

(R←R) ⇒ dest←R-1\*,  
(R[Rd]) ⇒ dest←R>

Status Affected:

N, Z

Encoding:

01000100a0000000ffff

Description:

The contents of register f are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register f (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).

register f

Words:

1

Cycles:

1

O Cycle Activity:

Q1Q2Q3Q4

DecodeRead register fProcess DataWrite to destination

Example 1:

RRNCF REG, 1, 0

Before Instruction  
REG = 1101 0111

After Instruction  
REG = 1110 1011

Example 2:

RRNCF REG, 0, 0

Before Instruction  
W = ?  
REG = 1101 0111

After Instruction  
W = 1110 1011  
REG = 1101 0111

- 2. 使用汇编器和 / 或编译器以及链接器汇编、编译和链接软件以将代码转换为 “0 和 1 序列” —— 可被 PIC MCU 识别的机器码。机器码最终将变为固件（编程到单片机中的代码）。
- 3. 测试代码。通常，复杂的程序不一定会按照预期运行，要得到正确的结果，还需要去除设计中的“错误”（bug）。可通过调试器观察与所编写的带有符号和函数名的源代码相对应的机器码中“0 和 1 序列”的执行。在调试过程中，可测试代码以观察变量在程序执行过程中各个点的值、进行“what if”检查、更改变量值和单步调试程序。
- 4. 将代码“烧写”到单片机中，验证其在最终应用中是否能正确执行。

当然，其中的每个步骤都可能非常复杂。重要的是必须关注设计中的细节，并依靠 MPLAB IDE 及其组件来完成每个步骤，这样就不会不断地将时间浪费在学习上。



虽然可使用 **MPLAB IDE** 对电路和代码建模，以便做出关键的设计决定，但步骤 1 仍需由设计人员完成。

**MPLAB IDE** 真正起帮助作用的是步骤 2 到步骤 4。它的程序编辑器有助于使用选定的语言工具编写正确的代码。编辑器可以识别汇编器和编译器的编程语法结构，从而自动将源代码以不同颜色区分，这有助于确保代码在语法上的正确性。项目管理器有助于组织应用程序中使用的各种文件：源文件、处理器描述头文件以及库文件。编译了代码之后，还可以控制编译器以何种程度优化代码大小或执行速度，以及将在器件中的哪些部分存储各个变量及程序数据。也可以指定“存储模型”以使应用能最佳地利用单片机的存储器。如果在编译应用程序时语言工具报错，则会显示出错的行，“双击”它即可转到对应的源文件，以便立即编辑。编辑后，可以按“**build**”（编译）按钮再次尝试编译。由于要编写和测试很多子程序段，因此复杂的代码通常会经过许多次这样的编写—编译—修正过程。**MPLAB IDE** 会以最快的速度执行这一过程，从而使您能够尽快转入下一个步骤。

代码编译没有错误之后，还需要对其进行测试。**MPLAB IDE** 具有用于 **PIC MCU** 和 **dsPIC DSC** 器件、称为“调试器”的组件和免费的软件模拟器，以帮助测试代码。即使硬件还没有完成，也可以使用软件模拟器开始测试代码。软件模拟器就是一种模拟单片机执行的软件程序。软件模拟器可以接收模拟的输入（激励信号），以便模拟固件对外部信号的响应。软件模拟器可以测量代码执行时间、单步调试代码以观察变量和外设，并跟踪代码以生成详细的程序运行记录。

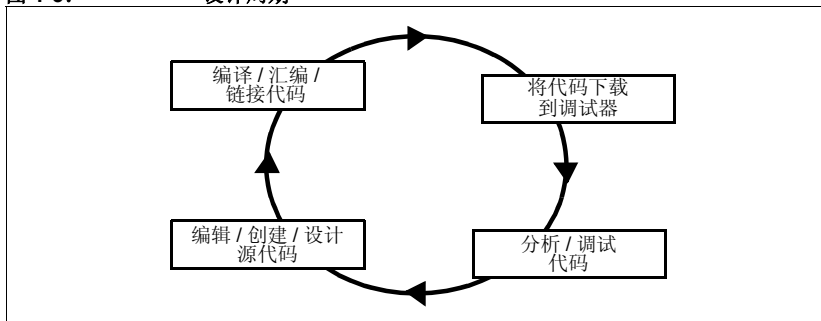
一旦硬件进入样机阶段，就可以使用诸如 **MPLAB ICE 2000** 在线仿真器、**MPLAB REAL ICE™** 在线仿真器或 **MPLAB ICD 2** 在线调试器的硬件调试器。这些调试器在实际的应用中实时运行代码。**MPLAB ICE 2000** 仿真器实际上取代了目标板上的单片机，它使用高速仿真头对设计中的硬件进行完全控制。**MPLAB REAL ICE** 在线仿真器和 **MPLAB ICD 2** 调试器则使用在许多带有闪存的 **Microchip MCU** 中内置的特殊电路，并且能“检查”目标单片机中的程序和数据存储器。这两类调试器可以停止和开始执行程序，使您可以直接使用应用板上的单片机测试代码。

应用程序正确运行之后，就可以使用 **Microchip** 的某种器件编程器来对单片机编程了，如 **PICSTART® Plus** 或 **MPLAB PM3**。这些编程器可以验证最终代码是否按设计运行。**MPLAB IDE** 支持 **PIC MCU** 和 **dsPIC** 数字信号控制器。

## 1.2 开发周期

编写应用程序的过程通常被称为开发周期——因为第一次就可以完成从设计到实现的所有步骤而不出任何差错是很少的。通常，编写代码之后要进行测试和修改才能生成正确执行的应用程序。集成开发环境让嵌入式系统设计工程师能够顺利完成这种开发周期，而不会因为要在各种工具之间切换而分心。使用 **MPLAB IDE**，所有功能都集成在一起，工程师就可以专心完成应用程序，而不会因为要切换不同的工具和操作模式而中断开发。

图 1-5: 设计周期

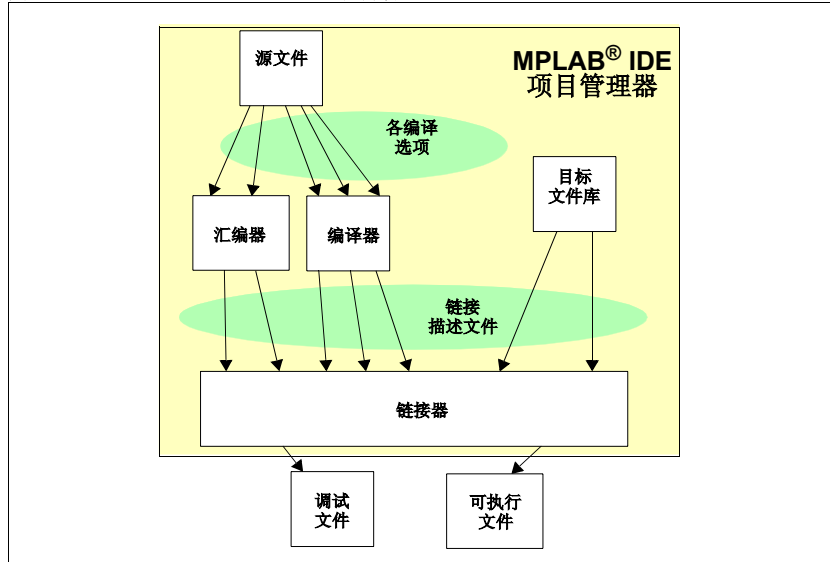


MPLAB IDE 是一种“包装器”（wrapper），它协调同一图形用户界面的所有工具——这一过程通常是自动完成的。例如，一旦代码编写完成，就可以将其转换为可执行指令，并下载到单片机上以观察它的运行。这一过程需要多种工具：编写代码的编辑器、管理文件及设置的项目管理器、将源代码转换为机器码的编译器或汇编器，以及多种连接到目标单片机的硬件或用来模拟单片机运行的软件。

### 1.3 项目管理器

项目管理器管理要编辑的文件和其他相关文件，以便将这些文件送至语言工具进行汇编或编译，并最终送至链接器。链接器的任务是将来自汇编器、编译器和库的目标代码片段存放到嵌入式控制器的恰当存储区，并确保各个模块之间相互作用（或“链接在一起”）。从汇编、编译到链接的整个操作过程被称为项目“build”。使用 MPLAB IDE 项目管理器，每个文件可以根据需要调用不同的语言工具，并且“build”过程会将所有的语言工具操作集成到一起。

图 1-6: MPLAB® IDE 项目管理器



源文件是遵从汇编器或编译器规则编写的文本文件。汇编器和编译器将源文件转换为中间机器码模块和占位符，以作为函数和数据存储的参考。链接器解析这些占位符，并将所有模块合并为一个可执行的机器码文件。链接器还会生成一个调试文件，允许 MPLAB IDE 将正在执行的机器码与源文件相关联。

文本编辑器用于编写代码。这不是普通的文本编辑器，而是为 Microchip MCU 编写代码而专门设计的编辑器。它可以识别文本中的语法结构，并采用彩色编码来区分各种元素，如指令助记符、C 语言语法结构和注释。编辑器支持编写源代码常用的操作，如查找 C 语言中的匹配花括号、注释掉和取消注释掉代码段、在多个文件中查找文本以及添加特殊书签等。编写好代码之后，编辑器可以配合其他工具，在调试器中显示代码的执行。可以在编辑器中设置断点，而且将鼠标指针悬停在变量名上方还可以查看变量的值。可以将源代码窗口中的变量名拖拽到 Watch（观察）窗口中。

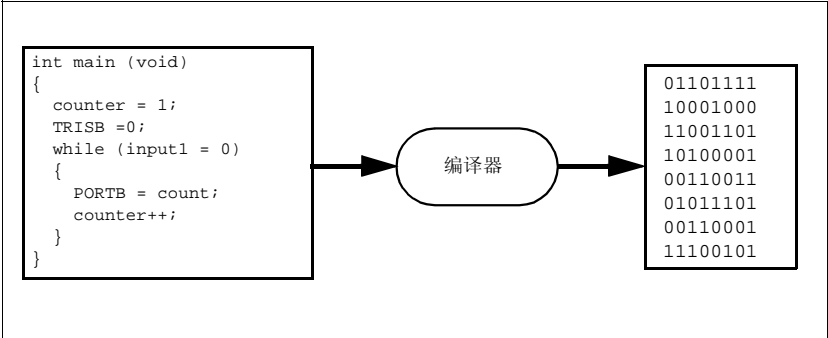
## 1.4 语言工具

语言工具就是诸如交叉汇编器和交叉编译器的程序。大多数人都比较熟悉在 PC 上运行的语言工具，比如 **Visual Basic** 或 **C** 编译器。当使用嵌入式系统的语言工具时，就会使用“交叉汇编器”或“交叉编译器”。这些工具与常见编译器的区别在于，它们在 PC 上运行，但生成的代码却在其他单片机上运行，即它们为单片机交叉编译代码，单片机使用与 PC 完全不同的指令集。

语言工具还会生成调试文件，MPLAB IDE 使用这个文件将机器指令和存储单元与源代码相关联。这种集成让 MPLAB IDE 编辑器能够设置断点，允许在 **watch** 窗口中查看变量的内容，并允许单步调试源代码，观察应用程序的执行。

嵌入式系统语言工具与在 PC 上运行和执行的编译器还有一点不同，那就是对空间非常敏感。生成的代码量越小越好，因为这样可以使用存储容量尽可能小的目标器件，从而降低成本。这就意味着需要一种使用特定于机器的知识来优化和增强代码的技术。对于 PC，复杂程度适中的程序通常就会大到数兆字节。而简单的嵌入式系统程序则可以小至几千字节，甚至更小。中等规模的嵌入式系统可能需要 **32K** 或 **64K** 字节的代码，以实现相对复杂的功能。某些嵌入式系统会使用数兆字节的空间，以存储大型表、用户文本消息或数据日志。

图 1-7: 编译器将源代码转换为机器指令



## 1.5 目标调试

在开发环境中，代码的执行是在调试器中测试的。调试器可以是软件程序，用来模拟单片机的操作以便进行测试，也可以是特殊的工具，用来分析程序在应用中的执行。

MPLAB IDE 中内置了软件模拟器，因此不需要任何其他硬件就可以对程序进行测试。软件模拟器是一种软件调试器，软件模拟器的调试功能与硬件调试器的功能几乎完全相同，学习起来非常容易。由于软件模拟器使用 PC 中的 CPU 来模拟单片机的操作，因此它通常比实际单片机运行要慢一些。MPLAB IDE 为所有的 PIC MCU 和 dsPIC DSC 处理器提供了许多软件模拟器。

在 MPLAB IDE 中，可以使用两种类型的硬件：编程器和硬件调试器。编程器直接将机器码从 PC 上烧写到目标单片机的片内存储器中。然后就可以将单片机插入到应用中，并希望单片机中的程序能够按设计运行。

但是，代码通常都不会完全按照预期运行，工程师需要检查代码及其在应用中的运行状况，以决定如何修改原始源代码，使之能够按预期运行。这个过程被称为调试。如前所述，软件模拟器可以用于测试代码的运行，但是一旦向单片机烧写了固件，软件模拟器之外的很多因素就发挥作用了。仅使用编程器，虽然可以更改代码，并将其重新烧写到单片机，然后插入目标板重新测试，但如果代码比较复杂，这个过程就非常费时费力，而且很难弄清楚硬件到底出了什么问题。

在这种情况下，硬件调试器就非常有用。硬件调试器可以是在线仿真器，它使用专用硬件来代替实际的目标单片机。也可以是在线调试器，使用具有特殊内置调试功能的单片机。硬件调试器和软件模拟器一样，允许工程师检查代码执行到各个点时的变量值，并在硬件与其专用电路交互时单步调试指令。

在项目设计周期的末期，调试通常会变得紧迫。随着交付期的临近，让应用程序按最初设计运行是部署产品之前的最后一步，这通常是使产品交付延期的最大因素。这就是集成开发环境最重要之处。调试和修改代码、重新编译、下载和测试——所有都需要时间。在一个环境中使用所有工具可以减少该“周期”的时间。能否在最后这几个步骤中找出关键错误对嵌入式系统设计人员是一种考验。使用正确的工具可以节省时间。使用 MPLAB IDE 可以选择多种工具，不过这些工具的界面都是类似的，因此从软件模拟器到低成本的在线调试器再到强大的在线仿真器的学习过程也会比较简单。

## 1.6 对器件编程

应用程序经过调试并在开发环境中运行后，还需要对其进行测试。可以使用在线调试器或器件编程器对器件进行编程。MPLAB IDE 可以设置为编程器功能，从而烧写器件。目标应用程序现在可视为接近完成的状态。工程样机开发编程器可以快速制作和评估样机。某些应用程序可以在器件焊接到目标 PC 板上之后进行编程。使用在线串行编程（In-Circuit Serial Programming™，ICSP™），可以在生产时将固件烧写到应用中，从而能够在嵌入式应用开发周期的末期将更新的版本烧写到嵌入式应用中。支持在线调试的器件甚至可以在生产完成后重新插入到 MPLAB ICD 2 中，以进行质量检测和下一代固件的开发。

## 1.7 MPLAB IDE 的组件

MPLAB IDE 具有内置组件和插件模块来为系统配置各种软件和硬件工具。

### 1.7.1 MPLAB IDE 内置组件

内置组件包括：

- **项目管理器**

项目管理器提供 IDE 和语言工具之间的集成和通信。

- **编辑器**

编辑器是功能全面的程序文本编辑器，它还可以作为调试器的关联窗口使用。

- **汇编器 / 链接器和语言工具**

汇编器可以单独使用，以汇编单个文件，也可以与链接器配合使用，以从不同的源文件、库和重新编译的目标文件生成一个项目。链接器负责将编译后的代码分配到目标单片机的存储区中。

- **调试器**

Microchip 的调试器允许使用断点、单步调试、watch 窗口以及目前 MPLAB IDE 调试器中的所有功能。将它与编辑器配合使用，可以将正在调试的目标中的信息与源代码相关联。

- **执行引擎**

MPLAB IDE 中具有适用于所有 PIC MCU 和 dsPIC DSC 器件的软件模拟器。这些软件模拟器使用 PC 来模拟 PIC MCU 和 dsPIC DSC 器件的指令和某些外设功能。还可以使用可选的在线仿真器和在线调试器通过应用硬件中运行代码来测试代码。

## 1.7.2 MPLAB IDE 的其他可选组件

可选购以下组件并将它们添加到 MPLAB IDE 中：

- **编译器语言工具**

Microchip 的 MPLAB C18 和 MPLAB C30 C 编译器可以提供高度集成的优化代码。MPLAB IDE 项目管理器可以调用这些编译器以及 HI-TECH、IAR、microEngineering Labs、CCS 和 Byte Craft 生产的编译器，来编译自动载入到目标调试器中的代码，进行即时测试和校验。

- **编程器**

MPLAB PM3、PICSTART® Plus、PICKit™ 1 和 2、以及 MPLAB ICD 2 调试器和 MPLAB REAL ICE 在线仿真器都可以将代码烧写到目标单片机。MPLAB IDE 提供了对代码和数据编程的完全控制，还可以使用 MPLAB IDE 对配置位进行设置来设定目标单片机的各种操作模式。

- **在线仿真器**

MPLAB REAL ICE 和 MPLAB ICE 2000 系统是 PIC MCU 和 dsPIC DSC 器件的在线仿真器。它们通过 I/O 端口与 PC 相连，可以对目标应用中的单片机操作进行完全控制。

- **在线调试器**

MPLAB ICD 2 和 PICKit 2 提供了一种取代仿真器的经济方案。通过使用部分片上资源，MPLAB ICD 2 可以将代码下载到被插入到应用中的目标单片机中、设置断点、单步调试和监视寄存器及变量。

## 1.8 MPLAB IDE 文档

现有以下文档可帮助您使用 MPLAB IDE：

- “MPLAB® IDE Quick Chart”（DS51410）
- 《MPLAB® IDE 快速入门指南》（DS51281G\_CN）——用户指南的第一章和第二章。
- 《MPLAB® IDE 用户指南》（DS51519A\_CN）

与 MPLAB IDE 配合使用的各种 Microchip 软件和硬件工具也存在相应文档。请访问 Microchip 网站获取这些文档的可下载 PDF 版本。

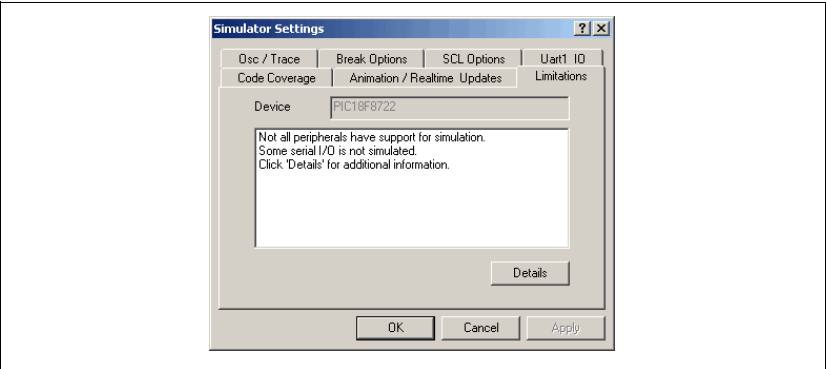
## 1.9 MPLAB IDE 在线帮助

由于 MPLAB IDE 还在不断地更改中（请参阅第 1.10 节“MPLAB IDE 更新和版本号”），本文中提到的部分细节可能会有所改变。对话框的外观可能与本手册中介绍的不完全一致，菜单列表的顺序也可能不同，或是增加了新的项。因此，在线帮助是您当前使用的 MPLAB IDE 版本的最佳参考。

MPLAB IDE 具有大量的在线帮助，处在不断更新之中。如果在使用 MPLAB IDE 时遇到了问题，一定要查看在线帮助寻找答案。最重要的是，在线帮助列出了特定工具在支持特定器件时可能存在的任何限制。在使用新的器件 / 工具组合时，请始终尝试查看这部分内容。

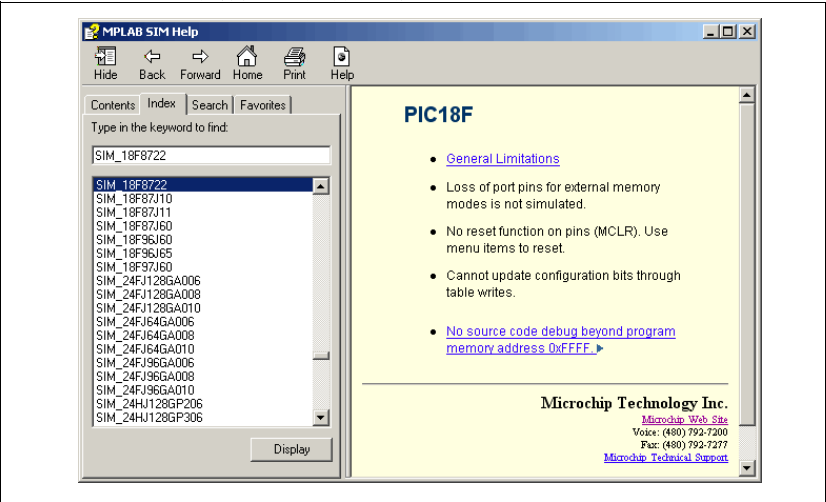
*Debugger>Settings* 对话框的 “Limitations”（限制）选项卡给出了软件模拟器、仿真器或在线调试器与正在被模拟的实际器件相比可能存在的任何限制。文本区域给出了一般限制。

图 1-8: *DEBUGGER>SETTINGS, LIMITATIONS* 选项卡



按下 **Details**（详情）按钮，显示与要被调试的器件相关的具体限制。从这里显示的信息中还可以看到与调试器相关的一般限制。

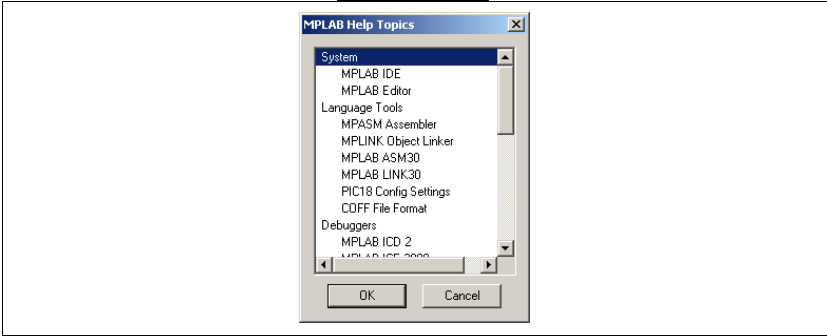
图 1-9: 软件模拟器限制详细信息





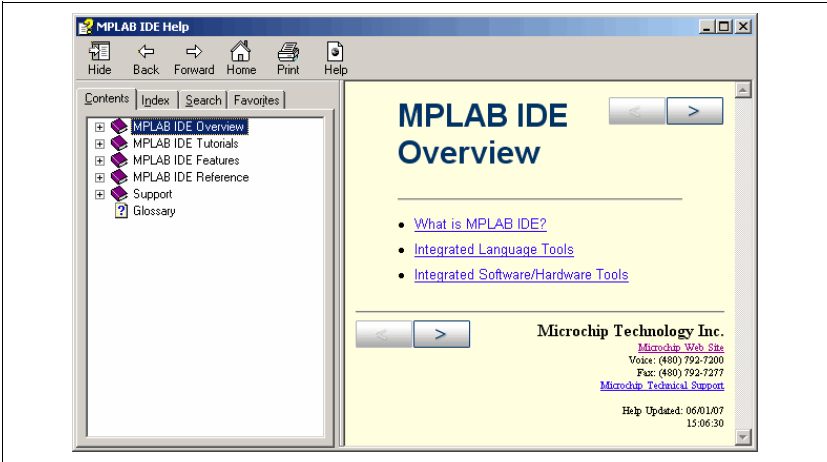
从 MPLAB IDE 的 Help（帮助）菜单，选择 Help>Topics，获取 MPLAB IDE 及其所有组件的帮助列表。

图 1-10: MPLAB® IDE Help>Topics 菜单



MPLAB IDE Help 涵盖了 MPLAB IDE 的各个方面以及所有的 Microchip 工具。可以按目录或索引方式离线查看它，也可以使用搜索程序查找有关任何 MPLAB IDE 主题的帮助。它也会告诉用户如何得到其他形式的帮助，例如 Microchip 的“客户变更通知”（Customer Change Notification）系统。

图 1-11: MPLAB® IDE HELP 对话框



## 1.10 MPLAB IDE 更新和版本编号

MPLAB IDE 是一个不断改进的程序，具有成千上万的用户。Microchip 在不断地设计具有新功能的新型单片机。许多新的 MPLAB IDE 功能都源自客户的需求和内部使用。不断开发的新设计和发布的新单片机促使 MPLAB IDE 将不断改进。

MPLAB IDE 计划大约每四个月更新一次，以便增加新的器件支持和新的功能。此外，在两个主要的发行版之间，还会发布另外的“临时”发行版。MPLAB IDE 的版本编号方案可以反映出当前版本属于主要产品发行版还是临时发行版。如果版本号以零结束，如 MPLAB IDE v7.00、v7.10 或 v7.20，则表示该版本是主要产品发行版。如果版本号以零以外的数字结束，如 v7.11、v7.22 或 v7.55，则表示该版本是临时发行版。临时发行版主要是为了提供对新器件或组件的早期试用，或进行快速重要修正及新功能的预演。这种临时发行版虽然是基于经过完整测试的产品发行版，但可能有一些组件并没有经过完整测试，因此建议不要在严格的设计中使用这种版本。

我们建议在开发过程中使用产品发行版，除非是在使用了新器件或组件，或是遇到了临时发行版中已修正的某个问题时，才使用临时发行版以有效利用 MPLAB IDE。还有，对于在发布新版本的 MPLAB IDE 时正在开发的项目，“最好”不要更新为最新发布版本，除非存在颇具说服力的理由，例如错误修复——修复当前版本中存在的错误。开始新项目时是更新版本的最好时机。

MPLAB IDE 软件的每一次新的发布都实现了新的功能，因此印刷文档的内容肯定会比在线帮助“滞后”。在线帮助是解决有关 MPLAB IDE 中任何问题的最好参考。

要在 MPLAB IDE 及其组件更新时得到通知，请在 [www.microchip.com](http://www.microchip.com) 网站上订阅“客户变更通知”服务的“Development Tools”部分。

---

## 第 2 章 MPLAB IDE 基础教程

---

### 2.1 简介

MPLAB 集成开发环境（IDE）是一个综合的编辑器、项目管理器和设计平台，适用于使用 Microchip PIC MCU 和 dsPIC DSC 进行嵌入式设计的应用开发。

本章讲述了 MPLAB IDE 的初步使用。本简短教程将讲述如何创建项目、编辑代码和测试应用程序。学习完本教程后，就可以快速了解项目管理器、编辑器和调试器的基本概念。在后面的章节中还会讲述 MPLAB IDE 的完整功能。

本节将详细说明 MPLAB IDE 的安装和卸载。接下来是介绍创建项目和说明 MPLAB IDE 基本调试功能的简单逐步教程。学习完教程后，MPLAB IDE 的新手将对使用此系统开发应用有一个基本的了解。无需前面的知识，以及 MPLAB IDE 及其组件的综合技术细节，就可形成使用 MPLAB IDE 的基本框架。

本教程将涉及到以下基本步骤：

- MPLAB IDE 的功能和安装
- 教程概述
- 选择器件
- 创建项目
- 设置语言工具
- 为项目命名
- 将文件添加到项目
- 编译项目
- 创建代码
- 再次编译项目
- 使用软件模拟器测试代码
- 教程总结

## 2.2 MPLAB IDE 的功能和安装

MPLAB IDE 是基于 Windows® 操作系统（OS）的集成开发环境，适用于使用 PIC MCU 系列和 dsPIC 数字信号控制器进行的开发。MPLAB IDE 提供以下功能：

- 使用内置的编辑器创建和编辑源代码。
- 汇编、编译和链接源代码。
- 通过使用内置的软件模拟器观察程序流程，或者使用在线仿真器或在线调试器以实时方式观察程序流程来调试可执行逻辑。
- 用软件模拟器或仿真器进行时序测量。
- 查看 Watch 窗口中的变量。
- 使用器件编程器将固件烧写到器件中（欲知详细信息，请查阅具体的器件编程器的用户手册）。

**注：** MPLAB IDE 也支持部分第三方工具。详情请查阅发布说明或自述文件。

### 2.2.1 安装 / 卸载 MPLAB IDE

要在系统上安装 MPLAB IDE：

**注：** 对于某些 Windows 操作系统，需要管理员权限才能在 PC 上安装软件。

- 若想从光盘安装，请将安装光盘放入光驱。遵循屏幕上的菜单来安装 MPLAB IDE。若未出现屏幕菜单，请使用“Windows 资源管理器”进行查找并执行光盘中的可执行文件 menu.exe。
- 若想从 Microchip 网站（[www.microchip.com](http://www.microchip.com)）下载 MPLAB IDE，请找到下载（.zip）文件，选择该文件然后将它保存到 PC。将该文件解压缩并执行解压缩后的文件来进行安装。

要卸载 MPLAB IDE：

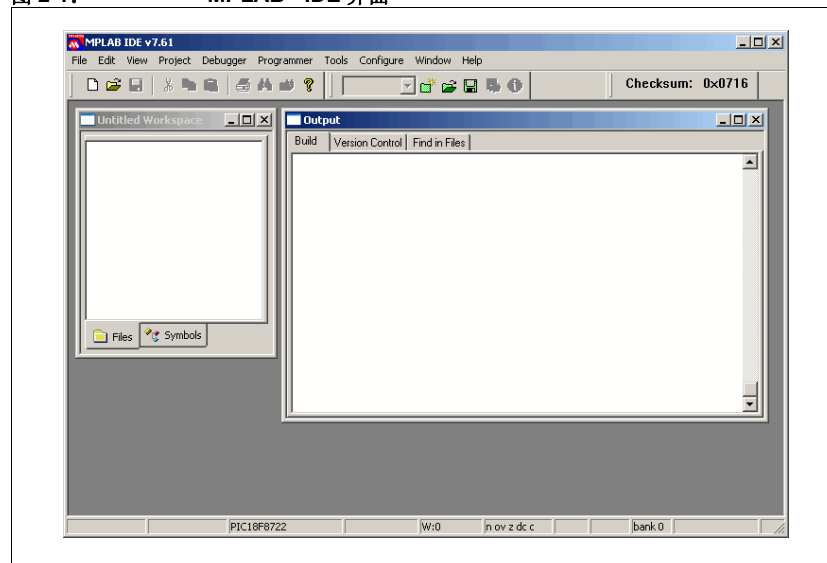
- 选择 **开始 > 设置 > 控制面板**，打开控制面板。
- 双击“添加 / 删除程序”。在列表中找到 MPLAB IDE 并单击它。
- 单击 **更改 / 删除** 将该程序从系统中删除。

## 2.2.2 运行 MPLAB IDE

软件安装后，要启动 MPLAB IDE，双击桌面上的图标或选择

开始 > 程序 > Microchip > MPLAB IDE vx.xx > MPLAB IDE。屏幕会显示 MPLAB IDE 徽标并随后显示 MPLAB IDE 界面（图 2-1）。

图 2-1: MPLAB® IDE 界面



## 2.3 教程概述

要创建目标 PIC MCU 的可执行代码，需要将源文件添加到项目中，然后可以使用选定的语言工具（汇编器、编译器和链接器等）将代码编译为可执行代码。在 MPLAB IDE 中，由项目管理器控制这一过程。

所有项目均具有以下基本步骤：

- 选择器件

MPLAB IDE 的功能根据所选择的器件而有所不同。应该在开始项目之前完成器件选择。

- 创建项目

将使用 MPLAB IDE 的 Project Wizard（项目向导）来创建项目。

- 选择语言工具

将在 Project Wizard 中选择语言工具。就本教程而言，将使用内置的汇编器和链接器。而在其他项目中，也许需要选择 Microchip 的某一编译器或其他第三方工具。

- 将文件添加到项目

将在项目中添加两个文件，一个模板文件和一个链接描述文件。这两个文件都位于 MPLAB IDE 文件夹的子文件夹中。使用这两个文件，开始项目就容易多了。

- 创建代码

将向模板文件添加一些代码，代码的功能是从 I/O 端口发送出一个递增的值。

- 编译项目

将编译项目——使源文件被汇编和链接成可以在选定的 PIC MCU 上运行的机器码。

- 使用软件模拟器测试代码

最后，将使用软件模拟器测试代码。

Project Wizard 将轻松地指导我们完成大部分的步骤。

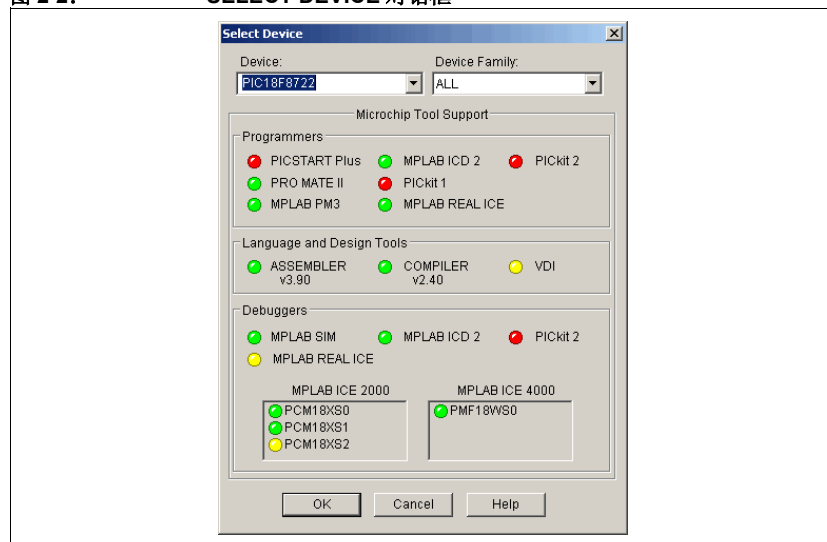
**注：** 在今后的产品发布中，用户界面中的某些部分可能会有所更改，并且本教程中的屏幕截图可能也会与以后发布的 MPLAB IDE 界面不完全相符。随着更多器件的发布，将会添加新的功能。本教程中描述的任何功能均不会被删除，但可能添加更多的功能。在线帮助是当前版本的 MPLAB IDE 最新的参考资料。

## 2.4 选择器件

为了在本文档中显示菜单选择的过程，MPLAB IDE 中顶行菜单中的菜单项将会显示在菜单名之后，如 *MenuName>MenuItem*。要在 *Configure*（配置）菜单中选择 *Select Device*（选择器件），将写作 *Configure>Select Device*。

请选择 *Configure>Select Device*。如果还未选择器件，请从 *Device*（器件）对话框的下拉列表中选择 **PIC18F8722**。

图 2-2: SELECT DEVICE 对话框



“指示灯”表示支持此器件的 MPLAB IDE 组件。

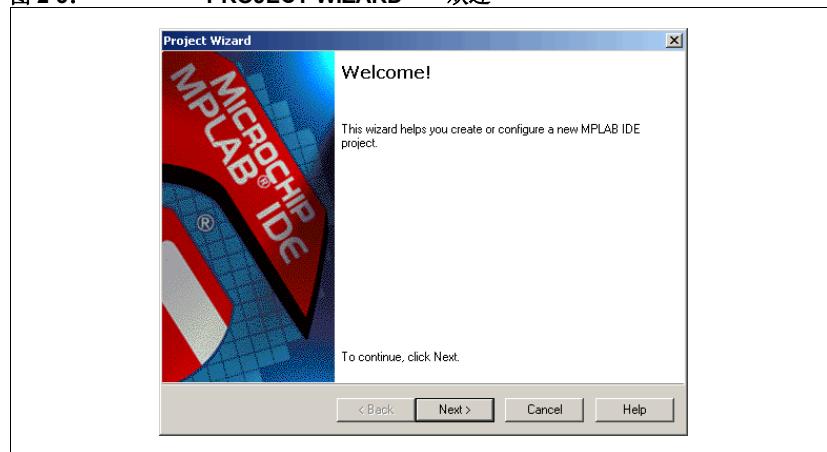
- 绿灯表示完全支持。
- 黄灯表示特定的 MPLAB IDE 工具组件对即将发布的器件的初步支持。有黄灯而不是绿灯的组件通常是针对新器件的早期采用者，他们需要这些器件的支持并明白某些操作和功能是不可用的。
- 红灯表示不支持这种器件。将来可能会提供支持或者该工具根本不适合该器件，例如，MPLAB ICE 2000 不支持 dsPIC DSC 器件。

## 2.5 创建项目

下一步，使用 **Project Wizard** 创建项目。项目是将文件组织起来以便进行编译和汇编的方式。我们将使用一个用于该项目的汇编文件和一个链接描述文件。选择 **Project>Project Wizard**。

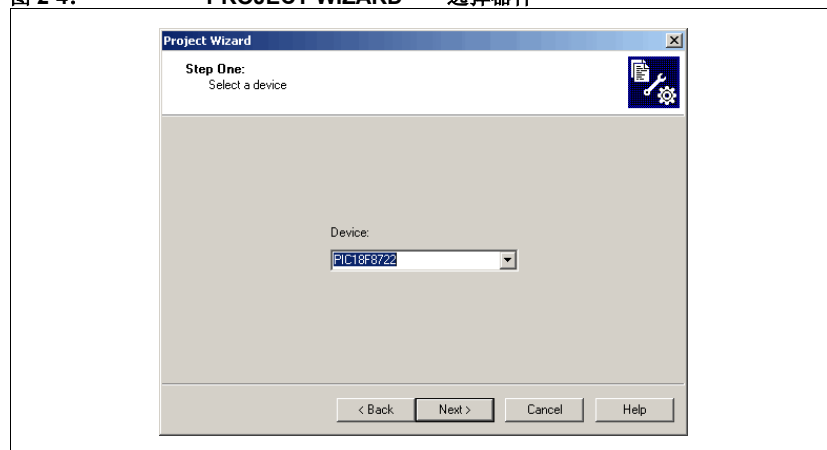
在 **Welcome**（欢迎）对话框中，单击 **Next>**（下一步）继续。

图 2-3: PROJECT WIZARD——欢迎



下一个对话框（**Step One**（第一步））让您选择器件（已选定）。确定所选择的是 **PIC18F8722**。如果不是的话，请在下拉菜单中选择 **PIC18F8722**。单击 **Next>**。

图 2-4: PROJECT WIZARD——选择器件





## 2.6 设置语言工具

Project Wizard 的第二步设置该项目所要使用的语言工具。在 Active Toolsuite（可用工具包）列表框中选择“Microchip MPASM Toolsuite”（Microchip MPASM 工具包）。然后应可在 Toolsuite Contents（工具包内容）框中看到“MPASM”和“MPLINK”。可以单击任一个查看其位置。如果将 MPLAB IDE 安装在默认目录下，则 MPASM™ 汇编器的可执行文件位于：

C:\Program Files\Microchip\MPASM Suite\mpasmwin.exe

MPLINK™ 链接器的可执行文件位于：

C:\Program Files\Microchip\MPASM Suite\\_mplink.exe

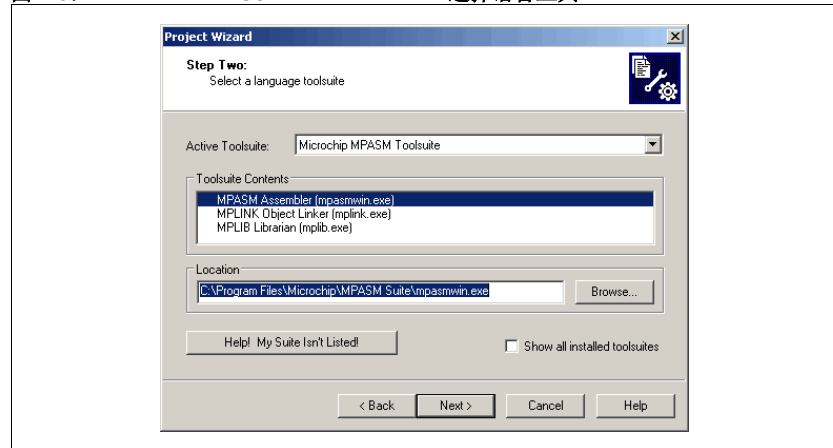
**注：** 还有一个可执行文件包 mplink.exe，它可调用 \_mplink.exe（链接器）、mp2cod.exe（输出转换为 COD 文件）和 mp2hex.exe（输出转换为 Hex 文件）。默认情况下，MPLAB IDE 将单独调用链接器和 Hex 转换器，不生成 COD 文件。

MPLIB™ 库管理器的可执行文件位于：

C:\Program Files\Microchip\MPASM Suite\mplib.exe

如果未正确显示，使用 Browse（浏览）按钮将它们设置为 MPLAB IDE 子文件夹中的相应文件。

图 2-5: PROJECT WIZARD——选择语言工具

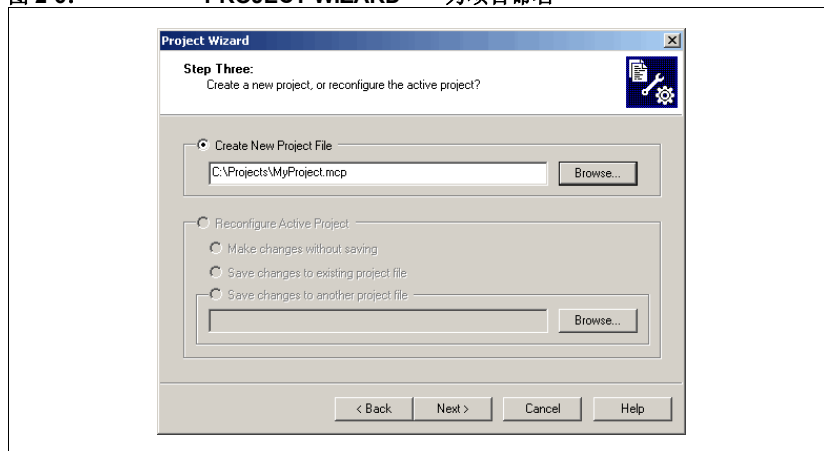


完成后，单击 **Next>**。

## 2.7 为项目命名

向导的第三步让您为项目命名并将它存入文件夹。这个示例项目将被命名为 `MyProject.mcp`，并放在 `C:\Projects` 目录下。单击 **Next>**。

图 2-6: PROJECT WIZARD——为项目命名



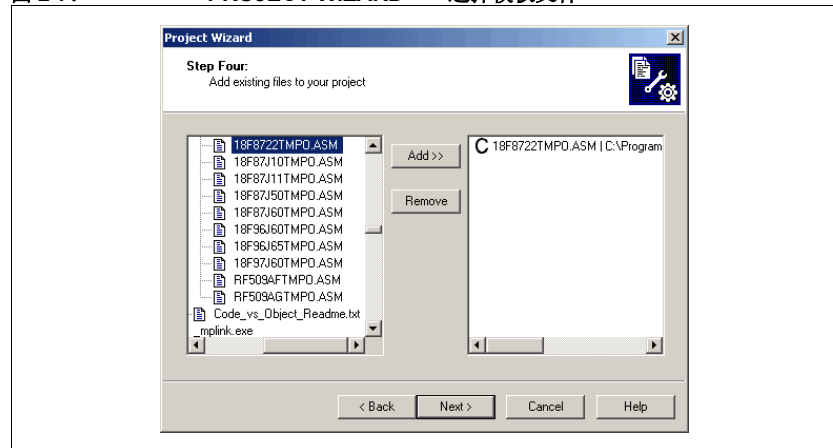
## 2.8 将文件添加到项目

Project Wizard 的第四步允许为项目选择文件。由于还未选择源文件，我们将使用 MPLAB IDE 模板文件。这些模板文件是用于开始项目的简单文件。它们具有任何源文件的基本部分，并且包含可帮助您编写和组织代码的信息。

每个 Microchip PIC MCU 和 dsPIC DSC 器件都有两个模板文件：一个用于 Code 目录下绝对代码（不使用链接器）的开发，另一个用于 Object 目录下可重定位代码（使用链接器）的开发。由于本教程中将使用链接器，请在 Object 目录下选择名为 18F8722TMPO.ASM 的文件。如果 MPLAB IDE 安装在默认位置，则该文件的完整路径为：

```
C:\Program Files\Microchip\MPASM Suite\Template\Object  
\18F8722TMPO.ASM
```

图 2-7: PROJECT WIZARD——选择模板文件



单击 **Add>>**（添加）将文件名移至右边的窗格中。然后单击文件名所在行开始处的“A”三次，直到出现“C”。这样可将此文件复制到项目目录。

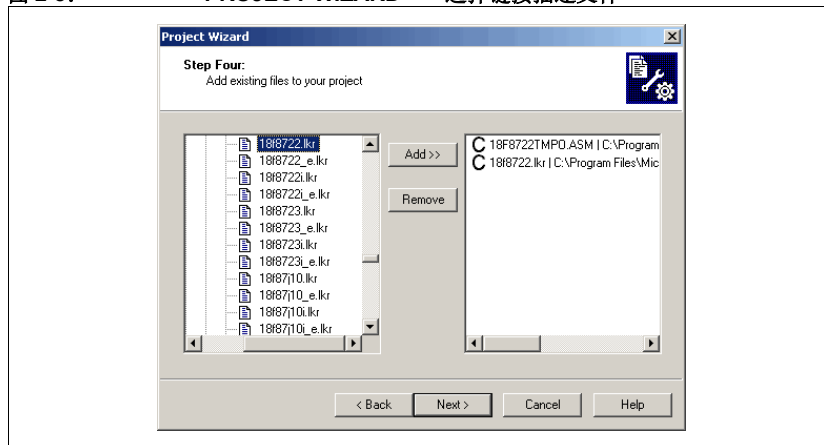
接下来，添加项目的第二个文件：链接描述文件。每个器件都有一个链接描述文件。这些文件定义各个器件的存储器配置和寄存器名称。本项目使用名为 18f8722.lkr 的文件。该文件的完整路径如下：

```
C:\Program Files\Microchip\MPASM Suite\LKR\18f8722.lkr
```

**注：** 还有一个名为 18f8722i.lkr 的链接描述文件，在使用 MPLAB ICD 2（所以名称中有“i”）时，用于此器件。该链接描述文件为 MPLAB ICD 2 保留了存储区。由于本例将使用软件模拟器，因此不需要使用该链接描述文件。  
还有另一个名为 18f8722\_e.lkr 的链接描述文件，用于器件扩展指令集。详细信息请参见相应器件的数据手册。本教程将不会使用扩展指令集。

要将链接描述文件复制到项目中，请单击“A”三次直到出现“C”。

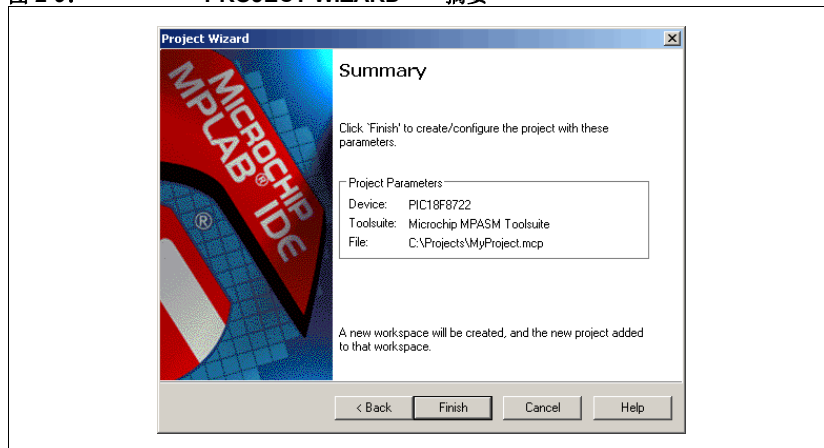
图 2-8: PROJECT WIZARD——选择链接描述文件



确定您的对话框类似于上图，然后单击 **Next>** 完成 Project Wizard。

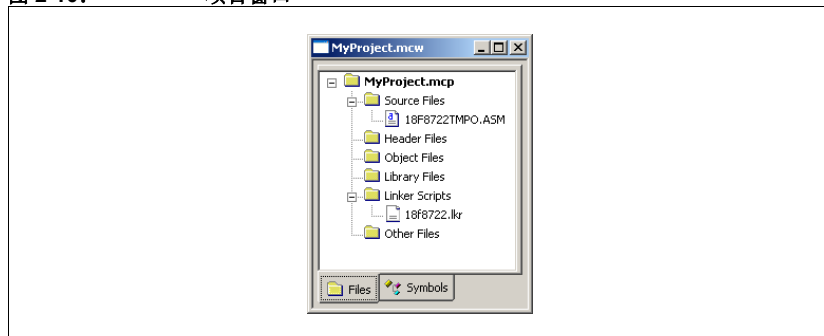
Project Wizard 的结束屏幕是一个摘要，显示了所选择的器件、工具包和新项目文件名。

图 2-9: PROJECT WIZARD——摘要



在单击了 **Finish**（完成）按钮之后，复查 MPLAB IDE 界面上的项目窗口。它看起来应该如图 2-10 所示。如果此项目窗口未打开，请选择 View>Project。

**图 2-10:** 项目窗口



**提示:** 可以通过在项目窗口中使用鼠标右键来添加文件和保存项目。发生错误时，可以通过选择文件并使用单击鼠标右键出现的菜单来手动删除文件。

## 2.9 编译项目

使用 **Project**（项目）菜单可以汇编和链接当前文件。虽然这些文件中还没有任何代码，但确保了项目是正确设置的。

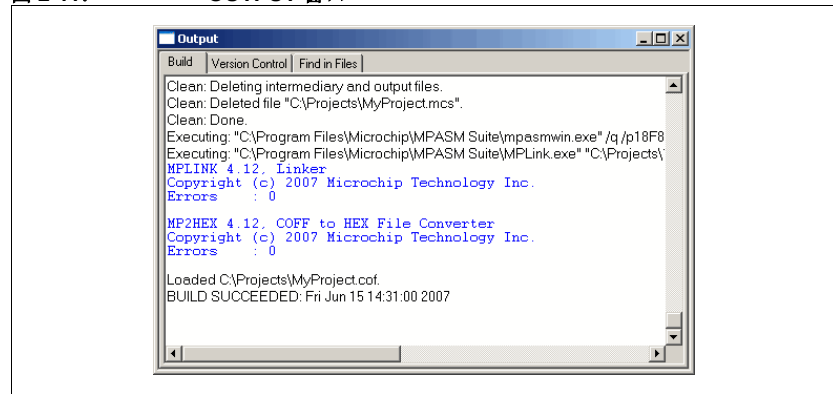
要编译项目，可选择以下任一种方式：

- **Project>Build All**
- 在项目窗口中右击项目名称并选择 **Build All**（编译所有）
- 在项目工具栏中单击 **Build All** 图标。将鼠标悬停在图标上即可看到弹出的文本，显示该图标的含义。

**Output**（输出）窗口将显示编译的结果。在任一步中，应该都没有错误或警告。但是，如果收到错误信息，请返回之前部分并检查项目汇编步骤。错误会阻止项目编译。如果收到警告信息，可以忽略它们，因为警告不会阻止项目编译。要关闭警告显示，请执行以下步骤：

- 选择 **Project>Build Options>Project** 并单击 **MPASM Assembler**（MPASM 汇编器）选项卡。
- 从下拉列表“**Categories**”（类型）中选择“**Output**”（输出）。
- 从下拉列表“**Diagnostic level**”（诊断等级）中选择“**Errors only**”（仅错误）。
- 单击 **OK**（确定）。

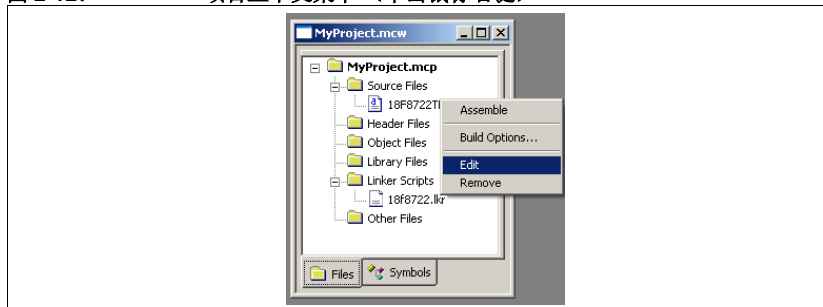
图 2-11: OUTPUT 窗口



## 2.10 创建代码

通过在项目窗口中双击模板文件名，或者通过用光标选择模板文件并用鼠标右键打开上下文菜单，可打开项目中的模板文件：

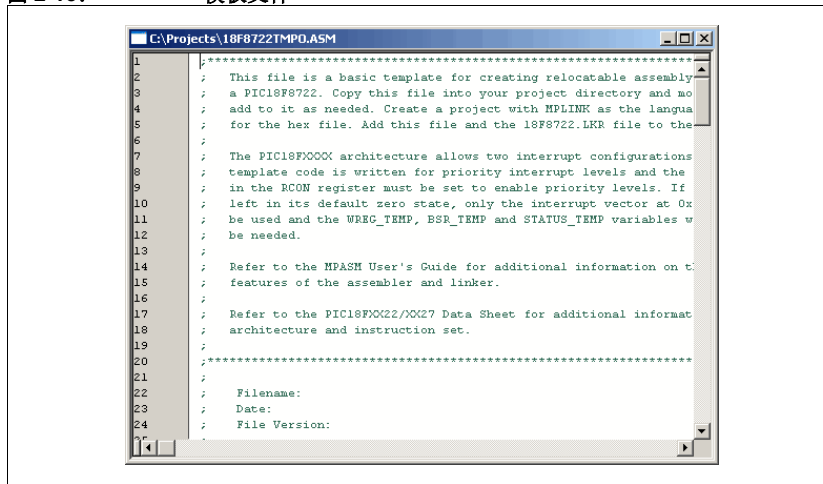
图 2-12: 项目上下文菜单（单击鼠标右键）



在此文件的开头有一些注释，并且此区域可被用作文件的标准注释信息头。现在您可以不管它，但如果这是一个实际的项目，您可以在这里输入与设计有关的信息。

**注：** 此处显示行数。可以通过右击编辑器窗口，选择 **Properties**（属性），然后在 **Editor Options**（编辑器选项）对话框的 '**ASM**' **File Type**（“ASM”文件类型）选项卡中选中 / 清除 **“Line Numbers”**（行数）复选框，来显示或隐藏行数。

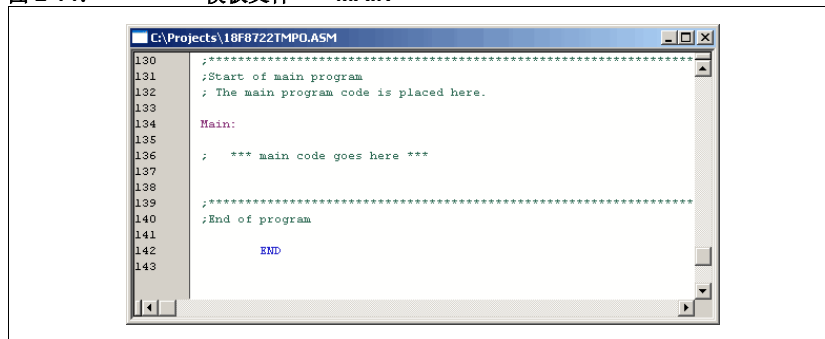
图 2-13: 模板文件



文件第一部分的代码用于较高级的功能，如在最终应用程序中设置中断和配置位。现在可以忽略这些细节而只专注于编写代码。新代码将放在文件中符号 **Main** 之后。

向下滚动到文件底部。

图 2-14: 模板文件——MAIN



打开任何源文件就可以自动进入编辑器。在 Main 下输入以下代码:

```

        clrf      WREG
        movwf     PORTC      ; clear PORTC
        movwf     TRISC      ; configure PORTC as all outputs

Init
        clrf      COUNT,A    ; initialize counter
IncCount
        incf      COUNT,F,A
        movf      COUNT,W,A  ; increase count and
        movwf     PORTC      ; display on PORTC

        call      Delay      ; go to Delay subroutine
        goto      IncCount   ; infinite loop

Delay
        movlw     0x40
        movwf     DVAR2,A    ; set outer delay loop
DelayOuter
        movlw     0xFF
        movwf     DVAR,A     ; set inner delay loop
DelayInner
        decfsz    DVAR,F,A
        goto      DelayInner

        decfsz    DVAR2,F,A
        goto      DelayOuter
        return

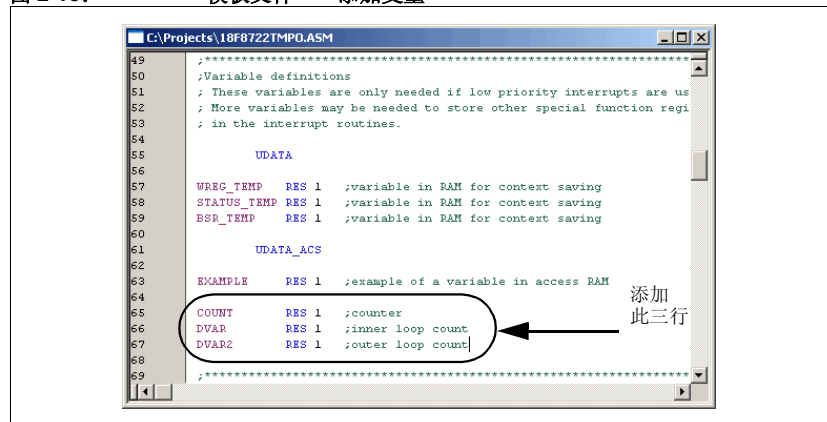
```



在这段简短的代码中，我们使用了三个变量 COUNT、DVAR 和 DVAR2。需要使用快速操作 RAM，在模板文件中名为 UDATA\_ACS 的未初始化数据段中定义这些变量。由于无需跟踪每个变量的存储区（banksel），使用快速操作存储区将简化代码。

在模板文件的此段中已有一个变量，因此使用相同的格式把我们的变量加在后面即可。每个变量都是 8 位的，因此，只需要为每个变量保留 1 个字节。

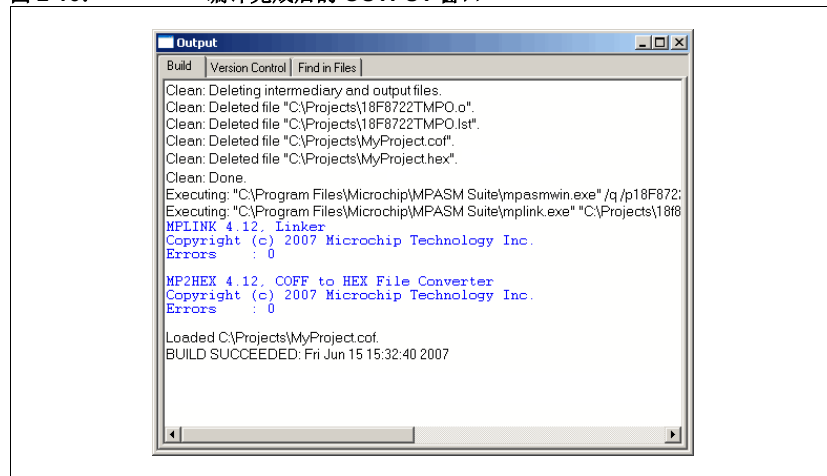
图 2-15: 模板文件——添加变量



## 2.11 再次编译项目

选择 **Project>Build All** 汇编和链接代码。如果代码汇编的过程没有出错，Output 窗口就如图 2-16 所示。

图 2-16: 编译完成后的 OUTPUT 窗口



如果汇编和链接不成功，检查以下各项然后再次编译项目：

- 如果汇编器在 **Output** 窗口中报错，双击错误，MPLAB IDE 就会打开源代码窗口，并在左边空白处用绿色箭头指向源代码中相应的行。
- 检查在编辑器窗口中输入的代码的拼写和格式。确认新变量和特殊功能寄存器 TRISC 和 PORTC 是大写的。
- 检查是否为 PIC MCU 器件使用了正确的汇编器（MPASM 汇编器）和链接器。选择 **Project>Set Language Tool Locations**。单击“+”号框以展开 Microchip MPASM Toolsuite 及其可执行文件。单击 MPASM Assembler (mpasmwin.exe) 然后检查显示的位置。如果位置正确，单击 **Cancel**（取消）。如果不正确，更改位置然后单击 **OK**。默认搜索路径可以为空。

在成功编译项目后，会加载语言工具生成的输出文件。此文件包含可被编程到 PIC MCU 的目标代码和调试信息，因此可以调试源代码并在 **Watch** 窗口中以符号的形式查看源变量。

**注：** 在实际应用中，需要将多个文件编译 / 汇编和链接成最终的可执行应用程序时，项目的真正作用非常明显。项目会跟踪所有这些过程。可为每个文件设置 **Bulid Options**（编译选项）来访问语言工具的其他功能，如报告输出和编译器优化。

## 2.12 使用软件模拟器测试代码

要测试代码，就需要用于执行 PIC MCU 指令的软件或硬件。调试执行工具，就是一种硬件或软件工具，用于在程序（在本例中，为 18F8722TMP0.ASM）执行过程中检查代码。硬件工具（如 MPLAB ICE 或 MPLAB ICD 2）可以在实际的器件中执行代码。如果没有硬件调试器，可以使用 MPLAB SIM 软件模拟器来测试代码。本教程使用的就是 MPLAB SIM 软件模拟器。

软件模拟器是在 PC 上运行的软件程序，它用于模拟 PIC MCU 的指令。它并不是“实时”运行的，因为软件模拟器程序的运行速度取决于 PC 的速度、代码的复杂性、操作系统的开销以及正在运行的其他任务的数目。但是，软件模拟器可精确测量代码在应用中实时运行时，所需要的时间。

**注：** 其他调试执行工具包括 MPLAB REAL ICE 在线仿真器、MPLAB ICD 2 在线调试器和 MPLAB ICE 2000 在线仿真器。可以选择这些硬件工具来测试应用板上的代码。使用这些硬件调试工具时，大部分 MPLAB IDE 调试操作和软件模拟器一样，但是与软件模拟器不同的是，这些工具允许目标 PIC MCU 以实际目标应用中的速度全速运行。

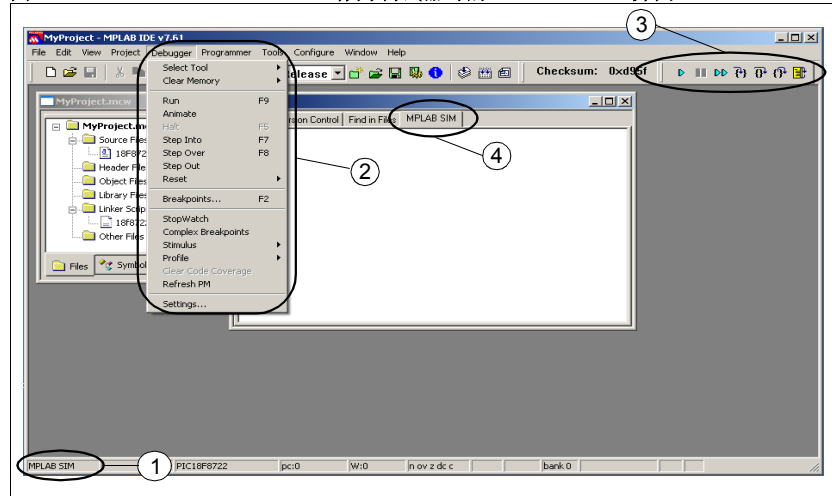
选择软件模拟器作为调试执行工具。这可以通过 **Debugger>Select Tool** 下拉菜单完成。在选择 MPLAB SIM 之后，应可看到以下变化（参见图 2-17 中的相应编号）。

- ① MPLAB IDE 窗口底部的状态栏应该变为“MPLAB SIM”。
- ② 现在 Debugger（调试器）菜单中应该会出现更多菜单项。
- ③ Debug（调试）工具栏中应该会出现更多工具栏图标。

**提示：**若将鼠标指针悬停于工具栏按钮之上，即可看到有关该按钮功能的简单描述。

- ④ Output 窗口添加了 MPLAB SIM 选项卡。

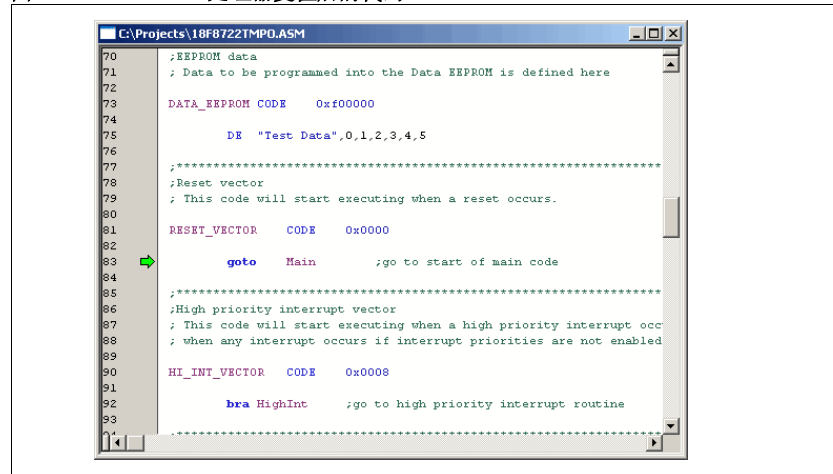
图 2-17: MPLAB SIM 作为调试器时的 MPLAB® IDE 界面



现在已设置了项目和选择了调试工具，应保存工作区设置。选择 **File>Save Workspace**。

接下来，选择 **Debugger>Reset>Processor Reset**，就会在程序开始之处显示一个绿色箭头。这是模板文件的一部分。此时存储器中的第一条指令将跳至名为 Main 的标号处（已在此处插入您自己的代码），即跳过向量区（复位向量和中断向量等）转到程序存储器中的用户存储空间。

**图 2-18: 处理器复位后的代码**



要单步执行应用程序，请选择 **Debugger>Step Into**。这将执行当前指向的代码行，然后箭头将指向要执行的下一行代码。

这些常用功能在 Debug 工具栏中也有快捷方式。

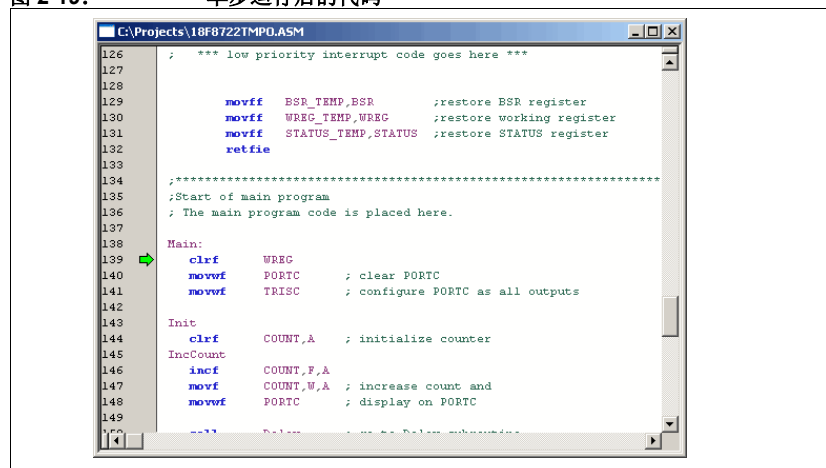
**表 2-1: 调试快捷方式图标**

Debugger 菜单	工具栏按钮	热键
Run（运行）		F9
Halt（暂停）		F5
Animate（单步连续运行）		
Step Into（单步运行）		F7
Step Over（单步跳过）		F8
Step Out（单步跳出）		
Reset（复位）		F6

**提示：**单击工具栏上相应的图标或者使用菜单项旁边显示的热键。这通常是重复操作的最佳方法。

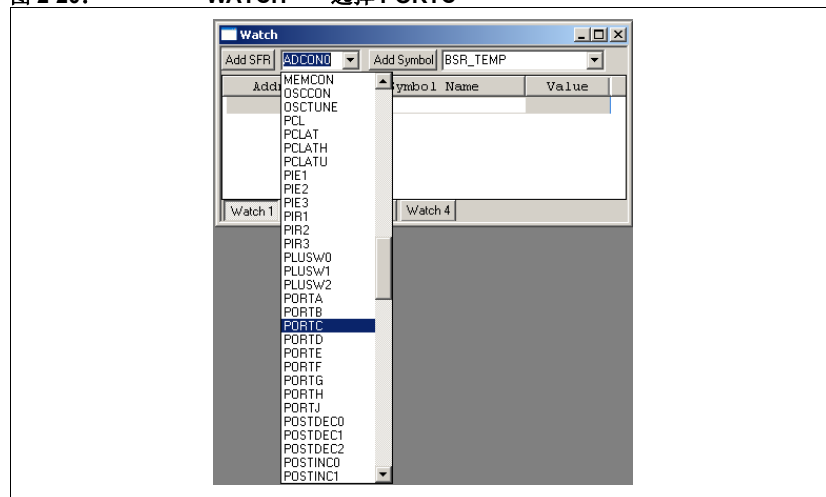
接下来，单击 Step Into 图标或选择 *Debugger>Step Into* 来单步运行 Main 中的代码。

图 2-19: 单步运行后的代码



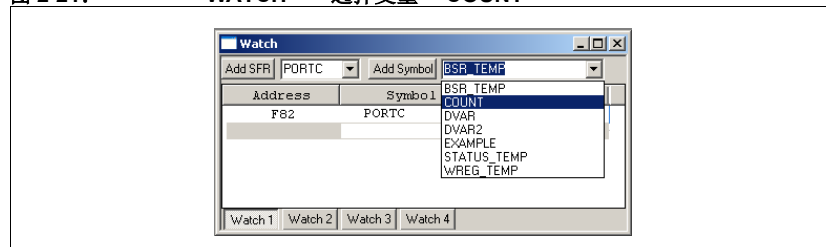
为了查看代码是否按照预期运行，可以向 PORTC 发送递增值，然后观察发送到 PORTC 的值。选择 *View>Watch* 打开一个空的 Watch 窗口。在 Watch 窗口的顶部有两个下拉列表。左边标注为“Add SFR”的下拉列表可用于将特殊功能寄存器 PORTC 添加到 Watch 窗口。从列表中选择 PORTC，然后单击 **Add SFR**（添加特殊功能寄存器）将它添加到窗口。

图 2-20: WATCH——选择 PORTC



右边的下拉列表允许添加程序中的变量。使用此下拉列表将 COUNT 变量添加到 Watch 窗口。从列表中选择 COUNT，然后单击 **Add Symbol**（添加变量）将它添加到窗口。

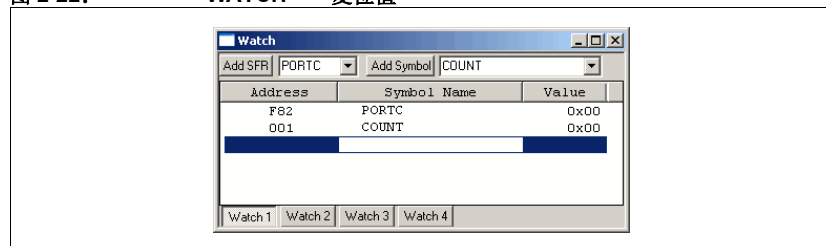
**图 2-21: WATCH——选择变量 “COUNT”**



Watch 窗口现在应显示两个寄存器的地址、值和名称。在程序执行的当前时刻，它们均应为零。

**注：** 也可以从 SFR、File Register（文件寄存器）或编辑器窗口中将这些项拖拽到 Watch 窗口，或者直接在窗口中的 symbol name（符号名称）下方单击，然后输入这些项，从而将它们添加到 Watch 窗口。

**图 2-22: WATCH——复位值**



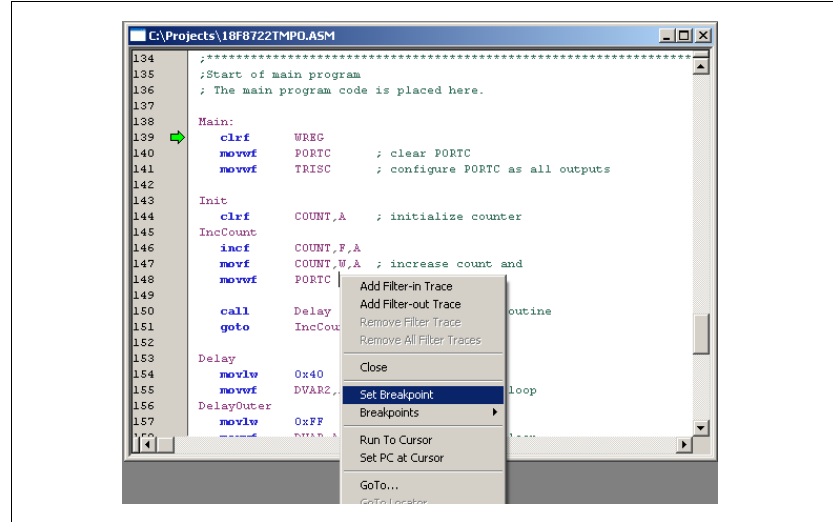
现在已添加了 Watch 窗口，再次使用 **File>Save Workspace** 保存工作区是一个很好的习惯。

可以继续单步执行代码，但是请将断点设置在向 PORTC 发送出第一个值之前。要设置断点，请将光标放在以下行上：

```
movwf    PORTC    ; display COUNT on PORTC
```

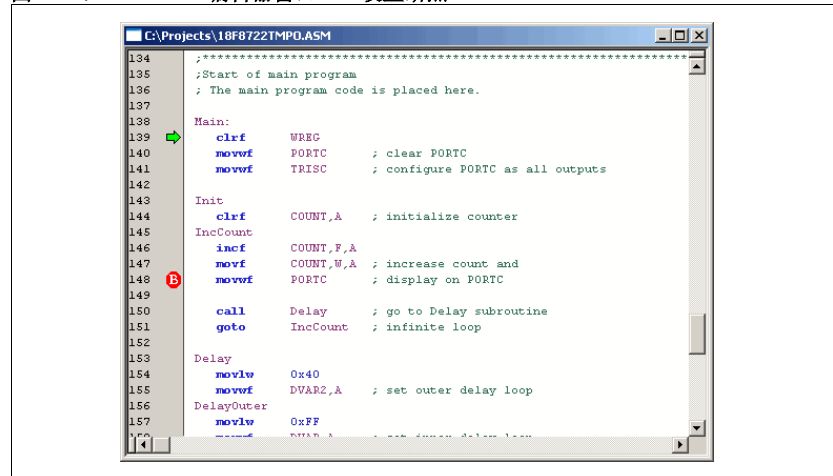
然后右击鼠标。

图 2-23: 调试上下文菜单（在行上右击鼠标）



从上下文菜单中选择 **Set Breakpoint**（设置断点）。该行上会出现一个红色的“B”。（也可以在行上双击鼠标来添加断点。）

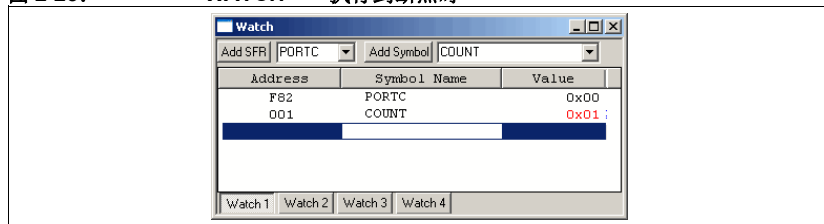
图 2-24: 编辑器窗口——设置断点



选择 **Debugger>Run** 运行应用程序。在应用程序到达在第一个断点之前，状态栏上会短暂地显示文本消息 “Running...”（运行 ...）。

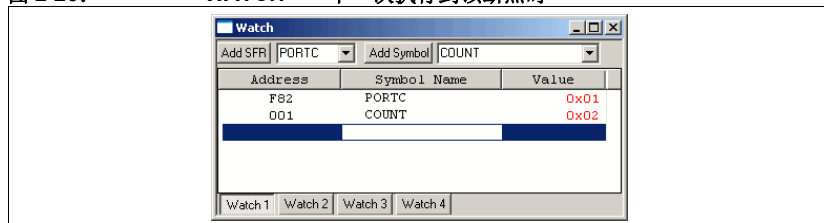
现在 Watch 窗口应显示变量 COUNT 的值已递增 1，但是由于断点位于将值传给 PORTC 的代码执行前的那一行上，因此 PORTC 的值仍为零。

图 2-25: WATCH——执行到断点时



单击 Run 图标执行代码，直到再次到达该断点。现在 Watch 窗口应显示两个值均递增 1。

图 2-26: WATCH——下一次执行到该断点时



这似乎表示程序按照设计正常运行。您可以单步运行代码或者多次运行代码以确认它是否正确执行。如果单步运行进入延时循环，您将需要不断执行几千次单步运行，才能结束循环。要跳出这种延时循环，可以使用 **Debugger>Step Out**。

如果对计算延时时间感兴趣，可以使用数据手册确定运行延时循环中的每条指令所需要的时间并由此得出一个相当精确的数字。您也可以使用 MPLAB IDE Stopwatch（跑表）来测量延时时间。重点应该关注显示 COUNT 的每个新值时的时间。如果像最初那样在从 COUNT 移动到 PORTC 的指令上设置断点，那么就可以运行程序到同一位置的下一个断点来测量该时间。

要使用 Stopwatch，可以通过在行上右击并选择 “Remove Breakpoint”（删除断点）来删除 PORTC 上的断点。然后，右击该行

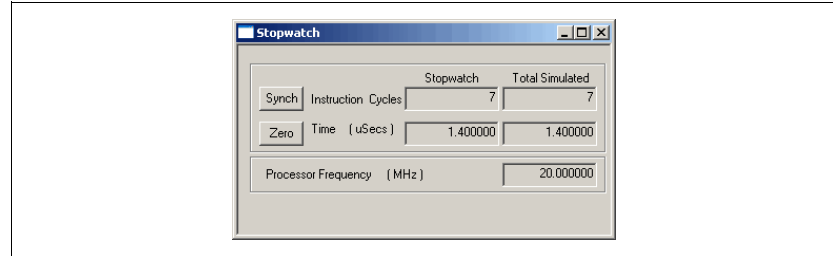
```
movf      COUNT,W,A ; increase count and
```

选择 “Set Breakpoint”。最后，选择 **Debugger>Reset>Processor Reset**。



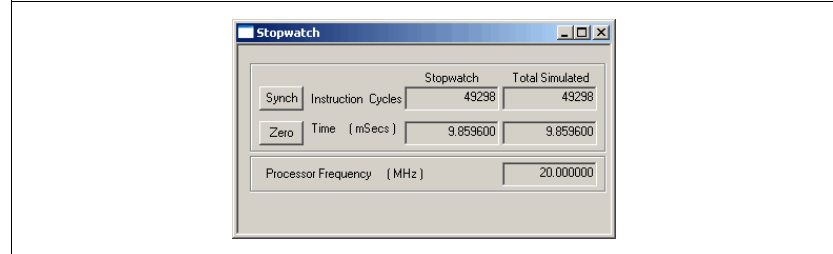
使用 **Debugger>StopWatch** 打开 StopWatch 对话框。单击 **Debugger>Run** 运行然后在断点处暂停。若采用默认的 20 MHz 的处理器频率，StopWatch 将显示到达第一个断点花了 1.4 us。

图 2-27: STOPWATCH——在第一个断点处



再次执行 Run 来运行一次循环，注意 StopWatch 显示它花费了 9.8596 ms。要更改此时间，可以改变延时循环中的值。要更改处理器频率，选择 **Debugger>Settings** 的 **Osc/Trace**（振荡器 / 跟踪）选项卡。

图 2-28: STOPWATCH——在延时之后



## 2.13 教程总结

通过学习本教程，您已经执行了创建、编译和测试简单项目的主要步骤。完成的任务包括：

- 选择器件，即 PIC18F8722。
- 使用 Project Wizard 创建项目，并使用该向导完成以下任务：
  - 选择 MPLAB IDE 中内置的 MPASM 汇编器和 MPLINK 链接器语言工具，
  - 为项目添加文件：用于选定器件的模板文件和用于正确编译项目的链接描述文件。
- 编写一些简单的代码，将变化的值发往 I/O 端口。
- 编译项目。
- 最后，使用软件模拟器测试代码。

这些是开始使用 MPLAB IDE 的基本步骤。现在您已准备就绪，可继续探索 MPLAB IDE 的功能了。

注:

注:

## 全球销售及服务中心

### 美洲

**公司总部 Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277  
技术支持:  
<http://support.microchip.com>  
网址: [www.microchip.com](http://www.microchip.com)

#### 亚特兰大 Atlanta

Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

#### 波士顿 Boston

Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

#### 芝加哥 Chicago

Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

#### 达拉斯 Dallas

Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

#### 底特律 Detroit

Farmington Hills, MI  
Tel: 1-248-538-2250  
Fax: 1-248-538-2260

#### 科科莫 Kokomo

Kokomo, IN  
Tel: 1-765-864-8360  
Fax: 1-765-864-8387

#### 洛杉矶 Los Angeles

Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

#### 圣克拉拉 Santa Clara

Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

#### 加拿大多伦多 Toronto

Mississauga, Ontario,  
Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

**亚太总部 Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

#### 中国 - 北京

Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

#### 中国 - 成都

Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

#### 中国 - 香港特别行政区

Tel: 852-2401-1200  
Fax: 852-2401-3431

#### 中国 - 南京

Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

#### 中国 - 青岛

Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

#### 中国 - 上海

Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

#### 中国 - 沈阳

Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

#### 中国 - 深圳

Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

#### 中国 - 武汉

Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

#### 中国 - 厦门

Tel: 86-592-238-8138  
Fax: 86-592-238-8130

#### 中国 - 西安

Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

#### 中国 - 珠海

Tel: 86-756-321-0040  
Fax: 86-756-321-0049

#### 台湾地区 - 高雄

Tel: 886-7-536-4818  
Fax: 886-7-536-4803

#### 台湾地区 - 台北

Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

#### 台湾地区 - 新竹

Tel: 886-3-572-9526  
Fax: 886-3-572-6459

### 亚太地区

**澳大利亚 Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

#### 印度 India - Bangalore

Tel: 91-80-4182-8400  
Fax: 91-80-4182-8422

#### 印度 India - New Delhi

Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

#### 印度 India - Pune

Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

#### 日本 Japan - Yokohama

Tel: 81-45-471-6166  
Fax: 81-45-471-6122

#### 韩国 Korea - Daegu

Tel: 82-53-744-4301  
Fax: 82-53-744-4302

#### 韩国 Korea - Seoul

Tel: 82-2-554-7200  
Fax: 82-2-558-5932 或  
82-2-558-5934

#### 马来西亚 Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

#### 马来西亚 Malaysia - Penang

Tel: 60-4-227-8870  
Fax: 60-4-227-4068

#### 菲律宾 Philippines - Manila

Tel: 63-2-634-9065  
Fax: 63-2-634-9069

#### 新加坡 Singapore

Tel: 65-6334-8870  
Fax: 65-6334-8850

#### 泰国 Thailand - Bangkok

Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### 欧洲

**奥地利 Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

#### 丹麦 Denmark - Copenhagen

Tel: 45-4450-2828  
Fax: 45-4485-2829

#### 法国 France - Paris

Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

#### 德国 Germany - Munich

Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

#### 意大利 Italy - Milan

Tel: 39-0331-742611  
Fax: 39-0331-466781

#### 荷兰 Netherlands - Drunen

Tel: 31-416-690399  
Fax: 31-416-690340

#### 西班牙 Spain - Madrid

Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

#### 英国 UK - Wokingham

Tel: 44-118-921-5869  
Fax: 44-118-921-5820