



MICROCHIP

MPLAB[®] X IDE

用户指南

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 safest 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, 是您自身应尽的责任。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、适用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、dsPIC、FlashFlex、flexPWR、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、MediaLB、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、PICSTART、PIC³² 徽标、RightTouch、SpyNIC、SST、SST 徽标、SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

The Embedded Control Solutions Company 和 mTouch 为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、ECAN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、KleerNet、KleerNet 徽标、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、RightTouch 徽标、REAL ICE、SQI、Serial Quad I/O、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

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

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2013-2014, Microchip Technology Inc. 版权所有。

ISBN: 978-1-63276-761-5

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

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-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

目录

前言	7
第 1 章 什么是 MPLAB X IDE?	
1.1 简介	11
1.2 嵌入式系统概述	12
1.3 开发周期	19
1.4 项目管理器	20
1.5 语言工具	21
1.6 目标调试	22
1.7 对器件编程	23
1.8 MPLAB X IDE 的组件	24
1.9 MPLAB X IDE 在线帮助	25
1.10 其他 MPLAB X IDE 文档	26
1.11 网站	28
1.12 MPLAB X IDE 更新	28
第 2 章 开始之前	
2.1 简介	29
2.2 安装 JRE 和 MPLAB X IDE	29
2.3 安装 USB 设备驱动程序（对于硬件工具）	30
2.4 连接目标（对于硬件工具）	34
2.5 安装语言工具	34
2.6 启动 IDE	35
2.7 启动 IDE 的多个实例	38
第 3 章 教程	
3.1 简介	41
3.2 设置硬件和软件	42
3.3 创建和设置项目	43
3.4 运行和调试代码	59
第 4 章 基本任务	
4.1 处理 MPLAB X IDE 项目	67
4.2 创建新项目	68
4.3 查看桌面上的变化	76
4.4 查看或更改项目属性	77
4.5 设置调试器、编程器或语言工具的选项	78
4.6 设置语言工具位置	80
4.7 设置其他工具选项	81
4.8 创建新文件	82
4.9 向项目中添加现有文件	84
4.10 编辑器用法	85

4.11 添加、设置库和目标文件	86
4.12 设置文件和文件夹属性	88
4.13 设置编译属性	90
4.14 编译项目	93
4.15 运行代码	94
4.16 调试运行代码	95
4.17 使用断点控制程序执行	97
4.18 单步执行代码	100
4.19 观察符号值变化	101
4.20 观察局部变量值的变化	103
4.21 查看 / 更改器件存储器（包括配置位）	104
4.22 查看调用堆栈	108
4.23 对器件编程	108

第 5 章 附加任务

5.1 执行附加任务	111
5.2 导入 MPLAB 旧项目	112
5.3 预编译项目	114
5.4 可装入项目和文件	115
5.5 可装入项目和文件：自举程序	118
5.6 库项目	119
5.7 其他嵌入式项目	120
5.8 示例项目	120
5.9 处理其他类型的文件	120
5.10 修改或创建代码模板	121
5.11 切换硬件或语言工具	122
5.12 修改项目文件夹和编码	122
5.13 加快编译速度	123
5.14 使用跑表	124
5.15 查看 Disassembly 窗口	124
5.16 查看调用图	125
5.17 查看仪表盘显示	126
5.18 改善您的代码	128
5.19 控制源代码	129
5.20 在代码开发和错误跟踪方面进行协作	131
5.21 添加插件工具	132

第 6 章 高级任务

6.1 简介	135
6.2 加快 MPLAB X IDE 速度	135
6.3 处理多个项目	137
6.4 处理多个配置	139
6.5 创建用户 MakeFile 项目	142
6.6 日志数据	144
6.7 定制工具条	145

第 7 章 编辑器

7.1 简介	153
7.2 编辑器用法	153
7.3 编辑器选项	155
7.4 代码折叠	157
7.5 C 代码重构	160

第 8 章 项目文件和文件夹

8.1 简介	163
8.2 Projects 窗口视图	163
8.3 Files 窗口视图	164
8.4 Favorites 窗口视图	166
8.5 Classes 窗口视图	166
8.6 查看用户配置数据	166
8.7 导入 MPLAB IDE v8 项目 —— 相对路径	167
8.8 移动、复制或重命名项目	167
8.9 删除项目	167

第 9 章 疑难解答

9.1 简介	169
9.2 USB 驱动程序安装问题	169
9.3 跨平台问题	169
9.4 MPLAB X IDE 问题	170
9.5 NetBeans 平台问题	171
9.6 错误	171
9.7 论坛	172

第 10 章 MPLAB X IDE 与 MPLAB IDE v8

10.1 简介	173
10.2 主要差异	173
10.3 菜单差异	176
10.4 工具支持差异	182

第 11 章 桌面参考

11.1 简介	183
11.2 菜单	184
11.3 工具条	193
11.4 状态条	196
11.5 灰显或缺失的项和按钮	196

第 12 章 MPLAB X IDE 窗口和对话框

12.1 简介	197
12.2 MPLAB X IDE 窗口管理	197
12.3 MPLAB X IDE 窗口与相关菜单和对话框	198
12.4 Breakpoints 窗口	198
12.5 Customize Toolbars 窗口	202
12.6 Licenses 窗口	203
12.7 Dashboard 窗口	203
12.8 存储器窗口	203

12.9 Output 窗口	213
12.10 Project Properties 窗口	214
12.11 Projects 窗口	214
12.12 Tools Options Embedded 窗口	219
12.13 Trace 窗口	223
12.14 Watches 窗口	224
12.15 向导窗口	226
第 13 章 NetBeans 窗口和对话框	
13.1 简介	227
13.2 NetBeans 特定窗口和窗口菜单	227
13.3 NetBeans 特定对话框	227
附录 A 配置设置汇总	229
A.1 简介	229
A.2 MPASM 工具链	229
A.3 HI-TECH® PICC™ 工具链	231
A.4 HI-TECH® PICC-18™ 工具链	231
A.5 C18 工具链	232
A.6 ASM30 工具链	232
A.7 C30 工具链	233
A.8 C32 工具链	235
A.9 XC 工具链	235
附录 B 在 IDE 之外工作	237
B.1 简介	237
B.2 在 MPLAB X IDE 之外编译项目	238
B.3 针对在 MPLAB X IDE 之外进行调试而编译	239
附录 C 版本历史	241
支持	245
术语表	249
索引	269
全球销售及服务网点	274

前言

简介

本章包含使用 MPLAB® X IDE 前需要了解的一般信息。讨论的内容包括：

- 文档编排
- 本指南使用的约定
- 推荐读物

文档编排

本文档介绍了如何使用 MPLAB X IDE。本手册的内容编排如下：

- **第 1 章 “什么是 MPLAB X IDE?”** —— 概述什么是 MPLAB X IDE，以及可以从何处获取帮助。
- **第 2 章 “开始之前”** —— 介绍如何安装硬件工具的 USB 驱动程序和用于编译 / 汇编代码的语言工具套件。
- **第 3 章 “教程”** —— 为使用 MPLAB X IDE 提供关于功能的分步说明。
- **第 4 章 “基本任务”** —— 介绍如何使用 MPLAB X IDE 的基本功能。它类似于“教程”一章，但提供更多的详细信息。
- **第 5 章 “附加任务”** —— 介绍如何使用 MPLAB X IDE 的附加功能，如导入 MPLAB IDE v8 项目或使用跑表。
- **第 6 章 “高级任务”** —— 介绍如何使用 MPLAB X IDE 的高级功能，如处理多个项目和项目配置。
- **第 7 章 “编辑器”** —— 概述了编辑器功能。关于编辑器的更多详细信息，请参见 NetBeans 帮助。
- **第 8 章 “项目文件和文件夹”** —— 介绍用于查看文件和文件夹的 MPLAB X IDE 窗口以及有关对这些文件进行操作的方法的信息。
- **第 9 章 “疑难解答”** —— 介绍疑难问题的解答。
- **第 10 章 “MPLAB X IDE 与 MPLAB IDE v8”** —— 说明 MPLAB X IDE 和 MPLAB IDE v8 之间主要的功能、菜单和工具支持差异。
- **第 11 章 “桌面参考”** —— 提供关于 MPLAB X IDE 桌面项（包括菜单、工具条和状态条）的参考信息。
- **第 12 章 “MPLAB X IDE 窗口和对话框”** —— 介绍 MPLAB X IDE 所独有的窗口和对话框。
- **第 13 章 “NetBeans 窗口和对话框”** —— 介绍 MPLAB X IDE 中可用的 NetBeans™ 窗口和对话框的参考信息。
- **附录 A “配置设置汇总”** —— 说明如何在代码中为所支持的语言工具设置配置位。这在 MPLAB X IDE 中是必需的，因为 Configurations Settings（配置设置）窗口只会为调试临时设置配置位。

- 附录 B “在 IDE 之外工作” —— 说明如何在 MPLAB X IDE 中创建可在 IDE 外使用的文件以及如何将在 IDE 外创建的文件导入到 MPLAB X IDE。
- 附录 C “版本历史” —— 列出文档的不同版本所做的更改。

本指南使用的约定

本指南采用以下文档约定：

文档约定

说明	表示	示例
Arial 字体：		
斜体字	参考书目	<i>MPLAB® IDE User's Guide</i>
	需强调的文字 为仅有的编译器
首字母大写	窗口	Output 窗口
	对话框	Settings 对话框
	菜单选择	选择 Enable Programmer
引用	窗口或对话框中的字段名	“Save project before build”
带右尖括号且有下列划线的斜体文字	菜单路径	<i>File>Save</i>
粗体字	对话框按钮	单击 OK
	选项卡	单击 Power 选项卡
N'Rnnnn	verilog 格式的数字，其中 N 为总位数，R 为基数，n 为其中一位。	4'b0010, 2'hF1
尖括号 <> 括起的文字	键盘上的按键	按下 <Enter>, <F1>
Courier New 字体：		
常规 Courier New	源代码示例	#define START
	文件名	autoexec.bat
	文件路径	c:\mcc18\h
	关键字	_asm, _endasm, static
	命令行选项	-Opa+, -Opa-
	二进制位值	0, 1
	常量	0xFF, 'A'
斜体 Courier New	可变参数	<i>file.o</i> , 其中 <i>file</i> 可以是任一有效文件名
方括号 []	可选参数	mcc18 [选项] <i>file</i> [选项]
花括号和竖线: {}	选择互斥参数；“或”选择	errorlevel {0 1}
省略号 ...	代替重复文字	var_name [, var_name...]
	表示由用户提供的代码	void main (void) { ... }

推荐读物

本用户指南介绍了如何使用 MPLAB X IDE。下面列出了其他有用的文档。以下 Microchip 文档均已提供，并建议读者作为补充参考资料。

MPLAB IDE 的自述文件

关于使用 MPLAB X IDE 的最新信息，请阅读 Start page（起始页）上“Release Notes and Support Documentation”（版本说明和支持文档）标题下的版本说明。版本说明包含了本用户指南中可能未提供的更新信息和已知问题。

自述文件

有关使用其他工具的最新信息，请阅读相应工具的自述文件，自述文件位于 MPLAB X IDE 安装目录的 Readmes 子目录下。自述文件包含了本用户指南中可能未提供的更新信息和已知问题。

在线帮助文件

在线帮助提供了全面的 MPLAB X IDE、MPLAB 编辑器和 MPLAB SIM 软件模拟器的帮助文件。包括教程、功能说明和参考资料。

器件数据手册和系列参考手册

请访问 Microchip 网站（<http://www.microchip.com>），以获取 PIC[®] MCU 和 dsPIC[®] DSC 数据手册和相关器件系列参考手册的完整已更新版本。

注:

第 1 章 什么是 MPLAB X IDE?

1.1 简介

MPLAB® X IDE 是一款软件程序，用于为 Microchip 单片机和数字信号控制器开发应用程序。（经验丰富的嵌入式系统设计人员可以跳过这章而直接阅读下一章。）

该开发工具称为集成开发环境或 IDE，因为它提供了单一的集成“环境”来开发用于嵌入式单片机的代码。

本章将介绍嵌入式系统的开发，并简要说明如何在该过程中使用来自 Microchip 的 MPLAB X IDE。

本章讨论以下主题：

- 嵌入式系统概述
- 开发周期
- 项目管理器
- 语言工具
- 目标调试
- 对器件编程
- MPLAB X IDE 的组件
- MPLAB X IDE 在线帮助
- 其他 MPLAB X IDE 文档
- 网站
- MPLAB X IDE 更新

1.2 嵌入式系统概述

嵌入式系统通常为一种利用小型单片机（如 Microchip PIC® 单片机（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单元（类似于个人计算机中的 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。这种单一的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。

1.2.1 嵌入式控制器和个人计算机之间的区别

嵌入式控制器和个人计算机之间的主要区别在于：嵌入式控制器专用于某项特定任务，或某些特定任务。而个人计算机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。

而个人计算机的核心具有比较昂贵的通用中央处理单元（Central Processing Unit, 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）。嵌入式系统具有低成本智能型单片机（MCU），在同一芯片上还有很多外设电路，而外部设备则相对很少。

通常，嵌入式系统属于隐藏部件，或者是其他产品（如充电电钻机、冰箱或车库开门器）的子模块。这些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关键子系统增添了低成本智能。

举例来说，烟雾探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。可以使烟雾探测器中的小程序无限循环，不停地对烟雾传感器输出的信号进行采样；也可以使烟雾探测器中的小程序处于低功耗的“休眠”模式，由传感器的输出信号将它唤醒，唤醒后烟雾探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，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个人计算机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 伏的电池以无人照管方式运行多年！）。嵌入式设计使用廉价的单片机，为我们日常生活环境中的方方面面提供智能化处理，如烟雾探测器、相机、手机、家用电器、汽车、智能卡以及安防系统。

1.2.2 单片机的组件

PIC MCU 具有片上程序存储器（图 1-1），用于储存固件或编码指令（图 1-2）以运行程序。程序计数器（Program Counter, PC）用于对程序存储器（包括复位和中断地址）进行寻址。硬件堆栈用于代码中的调用和返回指令，所以它与程序存储器配合工作，但不是后者的一部分。器件数据手册中介绍了程序存储器操作、向量和堆栈的详细信息。

图 1-1: PIC® MCU 数据手册 —— 程序存储器和堆栈

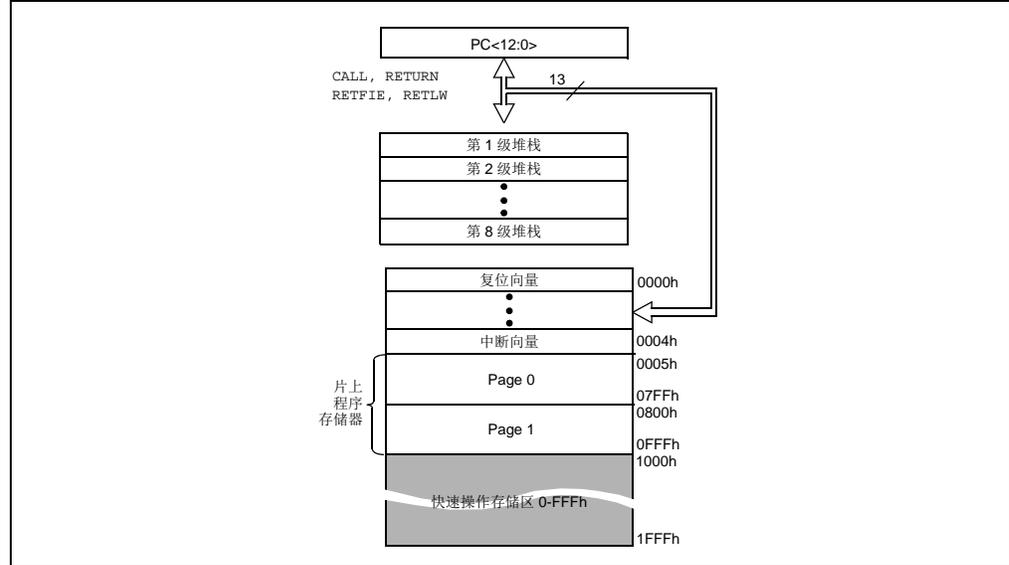


图 1-2: PIC® MCU 数据手册 —— 指令 (摘录)

RRNCF	Rotate Right f (no carry)								
Syntax:	[(abef)] RRNCF f [d [a]								
Operands:	0 ≤ f ≤ 255 d ∈ {0,1} a ∈ {0,1}								
Operation:	(f<n>) → dest<n-1> (f<0>) → dest<7>								
Status Affected:	N, Z								
Encoding:	0100 00da tfff tfff								
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).								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register f</td> <td>Process Data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register f	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register f	Process Data	Write 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								

单片机还具有数据或“文件寄存器”存储器。该存储器包含特殊功能寄存器（Special Function Register, SFR）和通用寄存器（General Purpose Register, GPR），如图 1-4 所示。SFR 是 CPU 和外设功能用来控制所需器件操作的寄存器。GPR 用于存储程序进行计算或临时存储所需的变量。一些单片机具有额外的数据 EEPROM 存储器。与程序存储器一样，器件数据手册中介绍了数据存储器和使用的详细信息。

图 1-3: PIC® MCU 数据手册 —— 文件寄存器

文件地址	文件地址	文件地址	文件地址
间接地址 (1)	间接地址 (1)	间接地址 (1)	间接地址 (1)
TMR0	OPTION_REG	TMR0	OPTION_REG
PCL	PCL	PCL	PCL
STATUS	STATUS	STATUS	STATUS
FSR	FSR	FSR	FSR
PORTA	TRISA	PORTA	TRISA
PORTB	TRISB	PORTB	TRISB
PORTC	TRISC	PORTC	TRISC
PCLATH	PCLATH	PCLATH	PCLATH
INTCON	INTCON	INTCON	INTCON
PIR1	PIE1	EEDAT	EECON1
PIR2	PIE2	EEADR	EECON2(1)
TMR1L	PCON	EEDATH	
TMR1H	OSCCON	EEADRH	
T1CON	OSCTUNE		
TMR2			
T2CON	PR2		
SSPBUF	SSPADD(2)		
SSPCON	SSPSTAT		
CCPR1L	WPUA	WPUB	
CCPR1H	IOCA	IOCB	
CCP1CON	WDTCON		
RCSTA	TXSTA	VRCON	
TXREG	SPBRG	CM1CON0	
RCREG	SPBRGH	CM2CON0	
	BAUDCTL	CM2CON1	
PWM1CON			
ECCPAS			
ADRESH	ADRESL	ANSEL	PSTRCON
ADCON0	ADCON1	ANSELH	SRCON
通用寄存器 96 字节	通用寄存器 80 字节	通用寄存器 80 字节	
	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh
Bank 0	Bank 1	Bank 2	Bank 3

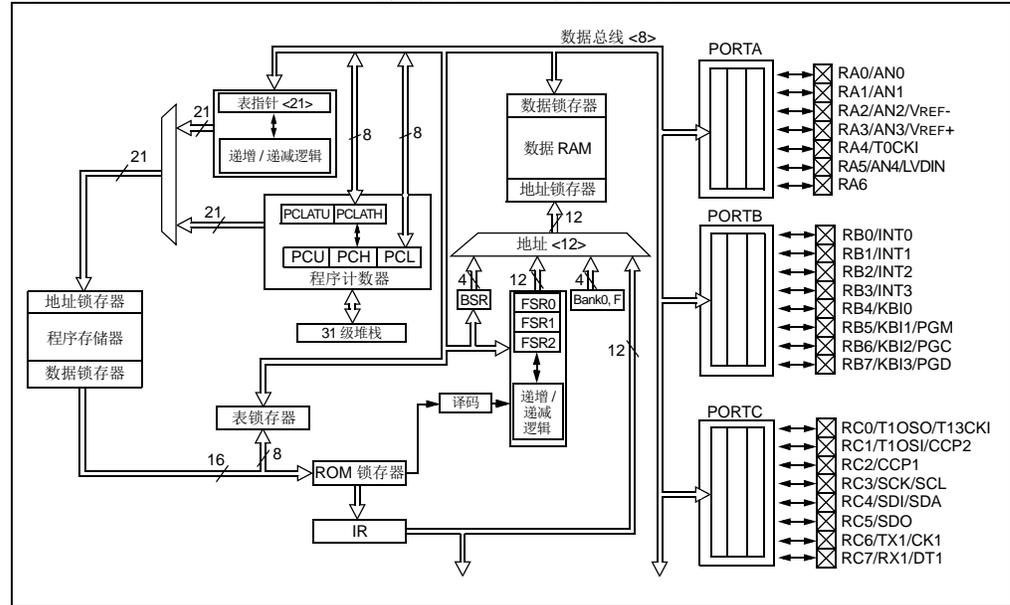
未实现的数据存储单元，读为 0。

注 1: 不是实际存在的寄存器。

注 2: 在某些情况下地址 93h 也可以访问 SSP 屏蔽 (SSPMSK) 寄存器。

除了存储器之外，单片机在同一芯片上还具有一定数量的外设电路（图 1-4）。一些外设称为输入/输出（input/output, I/O）端口。I/O 端口是单片机上的引脚，可以用作输出，并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，也可以配置为输入，以使程序能够对外部开关或传感器做出响应，或是与某些外部器件通信。

图 1-4: PIC® MCU 数据手册——框图（摘录）



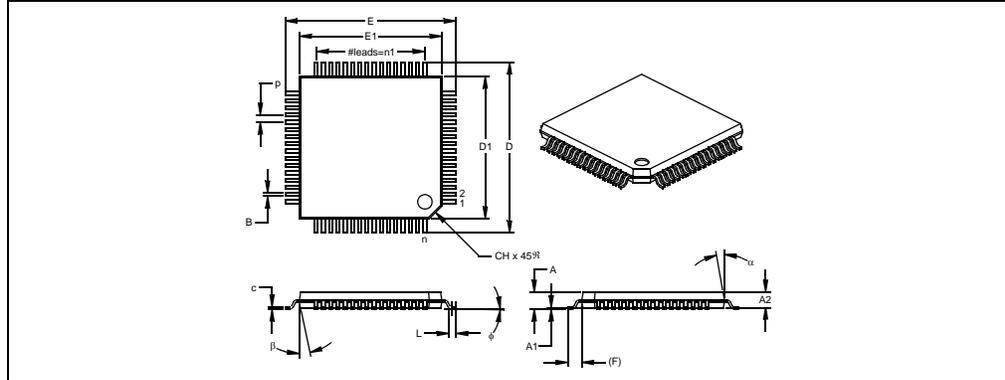
为设计这种系统，选择应用所需的外设。

下面列出了常用的外设：

- 模数转换器（Analog-to-Digital Converter, ADC）允许单片机连接到传感器并接收变化的电压。
- 串行通信外设可以通过几根线缆以串行方式与另一台单片机、局域网或互联网通信。
- PIC MCU 上称为“定时器”的外设可以精确地测量信号事件，并生成和捕捉通信信号以及输出精确的波形，甚至可以在单片机“挂起”或者因电源故障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何时降到了危险等级之下，以便让单片机能够及时存储关键信息，从而在完全掉电之前安全关闭。

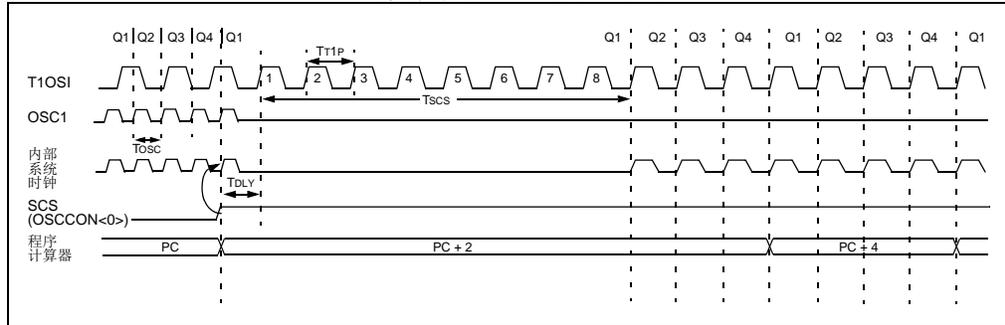
应该使用哪种 PIC MCU 在很大程度上取决于应用运行程序所需的外设和存储器大小。其他因素可能包括单片机的功耗及其“外形因素”，即安装到目标设计中的物理封装尺寸和特性（图 1-5）。

图 1-5: PIC® MCU 器件封装示例



当上电且振荡器开始产生时钟信号时，单片机会变为工作状态（图 1-6）。根据不同的单片机型号，可能会有几种内部和外部振荡器工作模式。

图 1-6: PIC® MCU 数据手册——时序（摘录）



1.2.3 使用 MPLAB X IDE 实现嵌入式系统设计

嵌入式控制器开发系统是在计算机上运行的程序系统，它有助于编写、编辑和调试程序代码（嵌入式系统应用的灵魂），并将其烧写到单片机中。MPLAB X IDE 就是这样一种系统；它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. 创建高阶设计。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（控制嵌入式应用中的硬件的软件）。可以使用语言工具编写和编辑代码，这些语言工具有汇编器（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）。汇编器和编译器允许使用功能标号来标识代码子程序，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。
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您的代码转换为“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，还需要除去设计中的“错误”（bug）。您可以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，您可以测试代码，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，验证其在最终的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 X 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 X IDE 对电路和代码进行建模，以便做出关键的设计决定，步骤 1 仍需由设计人员完成。

MPLAB X IDE 真正起帮助作用的是在步骤 2 至步骤 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，您还可以控制编译器以何种程度优化代码大小或执行速度，以及将在器件中的什么位置存储各个变量及程序数据。您也可以指定“存储器模型”以使您的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，可以尝试重新编译您的应用程序。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 X 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 X IDE 具有称为“调试器”的组件和免费的软件模拟器，以帮助所有的 PIC MCU 和 dsPIC DSC 器件测试代码。即使当硬件还没有完成时，您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。软件模拟器可以接收模拟输入（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，就可以使用诸如在线仿真器或在线调试器的硬件调试器了。这些调试工具使用在许多带有闪存程序存储器的器件中内置的特殊电路，在实际的应用中实时运行代码。它们可以“检查”目标单片机中的程序和数据存储器，并且可以停止和开始执行程序，使您可以直接使用安装到应用电路板上的单片机测试代码。

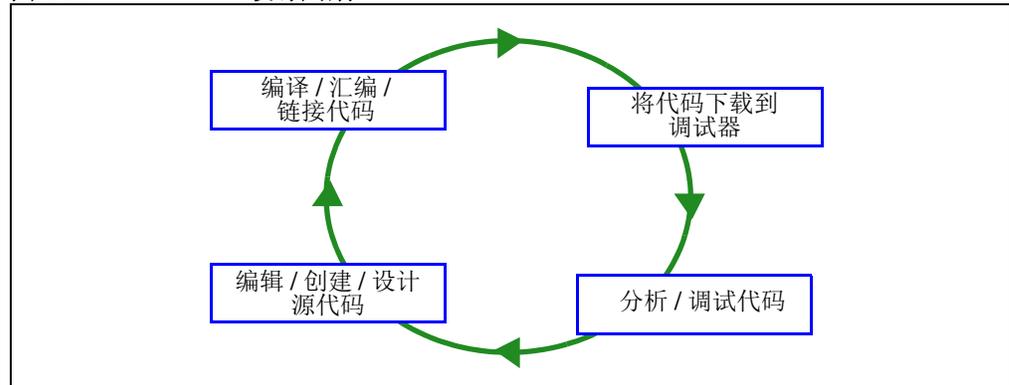
应用程序正确运行之后，就可以使用某一 Microchip 器件或开发编程器来对单片机编程了。这些编程器可以验证最终代码的运行和所设计的是否一致。MPLAB X IDE 支持大多数 PIC MCU 和所有的 dsPIC DSC。

1.3 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 X 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工作模式而中断开发。

图 1-7: 设计周期



MPLAB X IDE 就是一种“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，并下载到单片机中观察它的运行。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，以及某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。

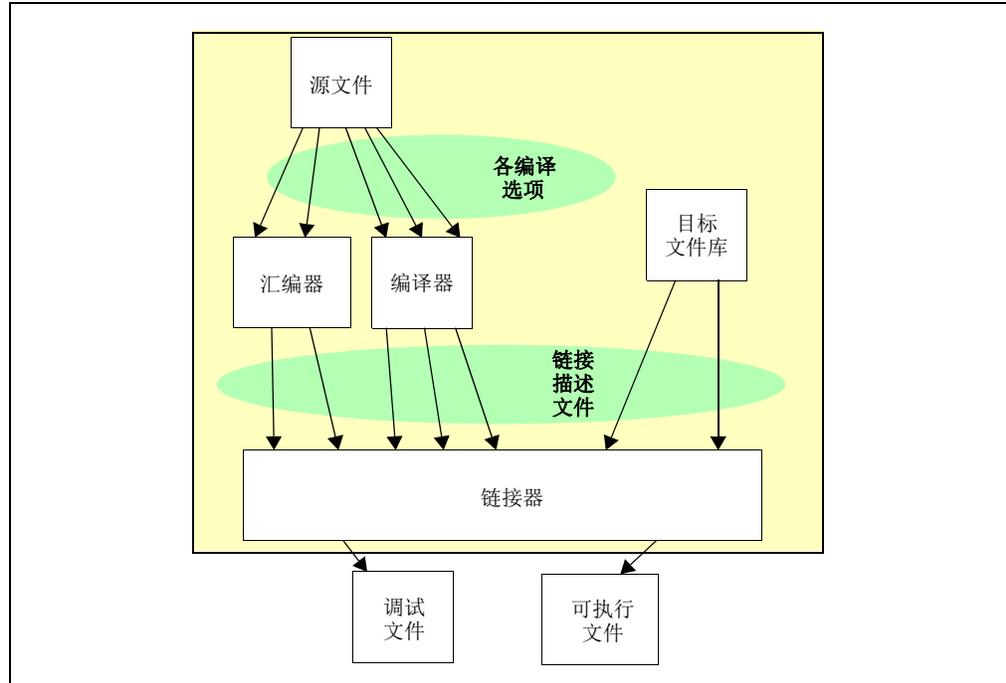
1.4 项目管理器

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可以相互作用（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”（编译）。根据需要，对于每个文件，对为语言工具所指定属性的调用可以不同，并且编译过程会将所有的语言工具操作集成到一起。

图 1-8: MPLAB® X 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 X 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。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 语言结构和注释。编辑器支持编写源代码常用的操作。编写好代码之后，编辑器可以配合其他工具，显示调试器中代码的执行。可以在编辑器中设置断点（停止或“暂停”代码执行），而且将鼠标指针悬停在变量名上还可以查看变量的值。可以将变量的名称从源代码文本窗口拖放到 **Watches**（观察）窗口中，然后可以在其中观察它们在每个断点之后或在代码执行过程中的变化值。

1.5 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计算机上运行的语言工具，比如 **Visual Basic** 或 **C** 编译器。

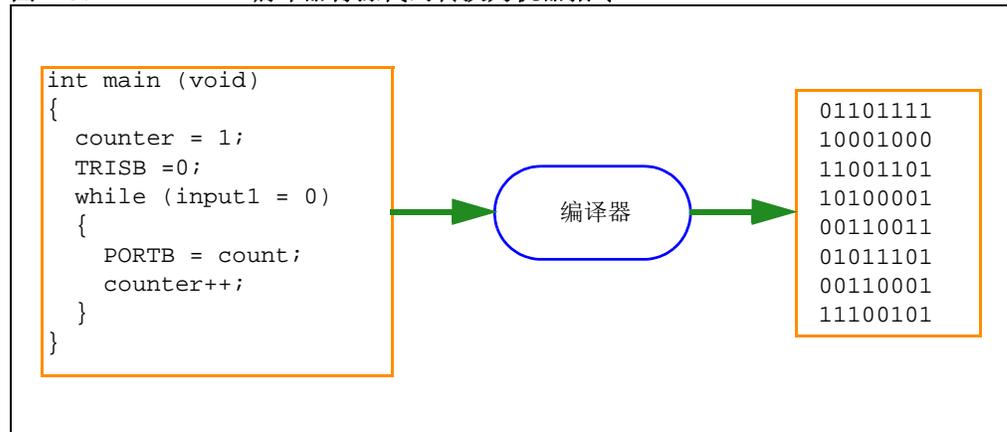
当使用嵌入式系统的语言工具时，就会使用“交叉汇编器”或“交叉编译器”。这些工具与常见编译器的区别在于，它们在计算机上运行，但生成的代码却在其他微处理器（或单片机）上运行。

语言工具还会生成调试文件，**MPLAB X 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 X IDE** 编辑器能够设置断点，允许在 **Watches** 窗口中查看变量的内容，并允许您单步调试源代码，观察应用程序的执行。

嵌入式系统语言工具与在计算机上运行和执行的编译器还有一点不同，那就是对存储空间非常敏感。生成的代码量越小越好，因为这样可以尽可能地减少目标对存储器的占用，从而降低成本。这就意味着需要特定于机器的知识来优化和增强代码的技术。

对于计算机，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** 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 1-9: 编译器将源代码转换为机器指令



1.6 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；也可以是硬件工具，用来在应用中执行程序从而对其进行分析。

1.6.1 软件调试器

MPLAB X 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计算机中的 CPU 来模拟单片机的操作，因此它通常比实际单片机运行得慢一些。

1.6.2 硬件调试器

在 MPLAB X IDE 中，您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 烧写到目标单片机的内部存储器中。然后就可以将单片机插入到应用中了，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，然后插入目标进行重新测试，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中究竟发生了什么问题。

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或在线调试器，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，允许工程师检查当代码执行到各个点时的变量，单步调试程序以在硬件与专用电路交互时检查指令。

1.6.3 集成开发环境

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 X 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。

1.7 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。可以使用在线仿真器、在线调试器、开发编程器或器件编程器对器件进行编程。MPLAB X IDE 可以设置为编程器功能，从而为器件烧写程序。目标应用程序现在可视为接近完成的状态。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 PCB 上之后进行编程。使用在线串行编程（In-Circuit Serial Programming™, ICSP™），可以在生产过程中将固件编程到应用中，从而能够在嵌入式应用开发的末期将更新的版本编程到嵌入式应用中。支持在线调试的器件甚至可以在生产完成后重新插入到在线调试器中，以进行质量检测和下一代固件的开发。

生产编程可以使用生产编程器和 MPLAB IPE（随 MPLAB X IDE 一起安装）实现。

1.8 MPLAB X IDE 的组件

MPLAB X IDE 包括：

- 功能全面的程序文本编辑器，它还可以作为调试器的窗口使用。
- 项目管理器（以 **Projects**（项目）窗口的形式显示），可提供 IDE 和语言工具之间的集成和通信。
- 许多汇编器 / 链接器套件，用于为项目所使用器件开发固件。
- 提供断点、单步、**Watches** 窗口和现代调试器所有功能的调试器引擎。该调试器与调试工具（包括软件和硬件）配合工作。
- 适用于所有 PIC MCU 和 dsPIC DSC 器件的软件模拟器。该模拟器实际上由几个特定于器件的模拟器可执行文件组成。MPLAB X IDE 将基于项目所使用器件来决定使用哪一个模拟器可执行文件。

可以获取或购买一些可选的组件，与 MPLAB X IDE 配合工作：

• 编译器语言工具

Microchip 的 MPLAB XC C 编译器可以为 PIC MCU 和 dsPIC DSC 提供高度集成的优化代码。MPLAB X IDE 项目管理器可以调用这些编译器以及 **microEngineering Labs**、**CCS** 和 **SDCC** 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 ICD 3 在线调试器、MPLAB REAL ICE™ 在线仿真器和 MPLAB PM3 编程器都能够用于在生产编程过程中将代码写入目标器件。PICKit™ 3 在线调试器能够用于在开发编程过程中将代码写入目标器件。

所有这些工具都可以与 MPLAB X IDE 配合使用，用于控制代码和数据编程，以及对配置位进行设置来设定目标单片机或数字信号控制器的各种工作模式。

此外，所有这些工具还可以与 MPLAB IPE 配合使用来进行代码、数据和配置位编程。MPLAB IPE 设计为更多地用于生产编程，它的界面也简化为仅执行该功能。

• 在线调试器和仿真器

PICKit 3 和 MPLAB ICD 3 在线调试器以及 MPLAB REAL ICE 在线仿真器都可用于在目标器件上调试应用程序代码。通过使用部分片上资源，这些工具可以将代码下载到插入到应用中的目标单片机中，并设置断点、单步调试以及监视寄存器和变量。仿真器包含了额外的调试功能，例如跟踪。

• 插件工具

提供了几个可增加 MPLAB X IDE 的功能的插件。例如，数据监视与控制界面（**Data Monitor and Control Interface, DMCI**）提供了一种机制来查看和控制代码中的变量，以及实时更改其值。它还允许您以图形格式查看输出数据。

关于支持的插件工具的更多信息，请参见第 5.21 节“添加插件工具”。

1.9 MPLAB X IDE 在线帮助

MPLAB X IDE 是基于 NetBeans 平台构建的。因此，许多 NetBeans 功能现在也成为 MPLAB X IDE 功能。

要完全了解 MPLAB X IDE 行为，请参见所有帮助文件。要启动帮助，请选择 [Help \(帮助\)](#) > [Help Contents \(帮助内容\)](#)。这会将所有帮助文件合并为一个，所以可能需要多一点的时间才会打开。

关于 NetBeans 的信息，请参见目录中“NetBeans Help”（NetBeans 帮助）下的联机帮助文件。关于所有 MPLAB X IDE 开发工具信息，请参见目录中“MPLAB X IDE Help”（MPLAB X IDE 帮助）下的联机帮助文件。

此外，您还可以在 [Help>Tool Help Contents \(工具帮助内容\)](#) 下查看选定工具的独立帮助文件。启动独立帮助文件的速度会较快，并且提供的主题搜索范围会较小。

关于 MPLAB X IDE 和 MPLAB IDE v8 的比较，请参见第 10 章“MPLAB X IDE 与 MPLAB IDE v8”。

1.10 其他 MPLAB X IDE 文档

除了帮助之外，Start Page（图 1-10）上还给出了其他文档、视频、论坛和 Wiki 的链接。

MPLAB X IDE Wiki 可供您查找关于高级功能的提示（图 1-11）。

图 1-10: MPLAB® X IDE 起始页

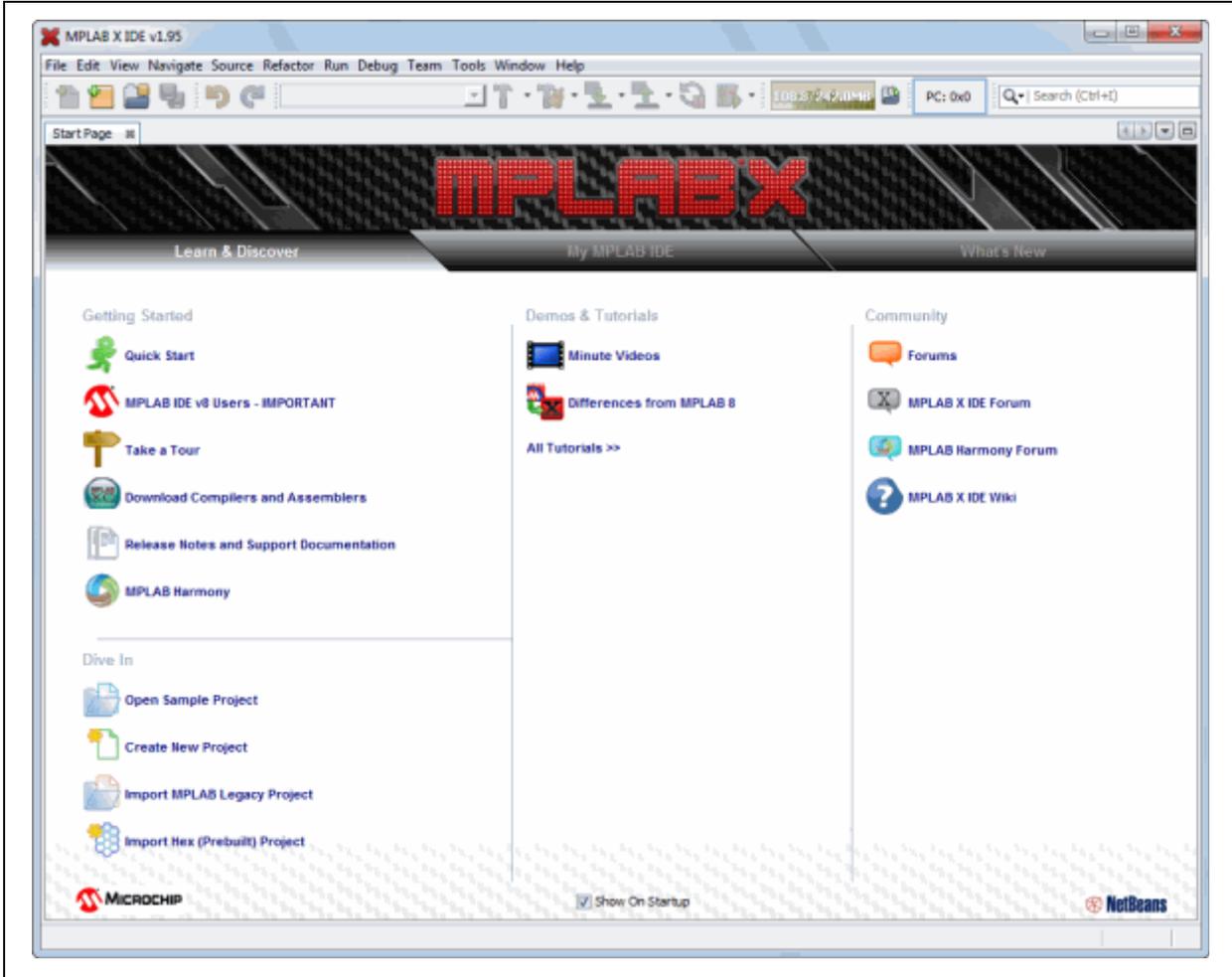


图 1-11: MPLAB® X IDE WIKI



1.11 网站

Microchip 通过我公司网站提供在线支持：

<http://www.microchip.com/devtools>

客户可通过该网站方便地获取文件和信息。更多详细信息，请参见**支持**。

1.12 MPLAB X IDE 更新

MPLAB X IDE 是一个不断改进的程序，具有成千上万的用户。Microchip Technology 在不断地设计具有新功能的新型单片机。许多新的 MPLAB X 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型单片机促使 MPLAB X IDE 将不断改进。

MPLAB X IDE 计划大约每几个月更新一次版本，以便增加新的器件支持和新的功能。

对于在 MPLAB X IDE 新版本发布时正处于开发中途的项目，“最佳做法”是不要更新为新版本，除非有令人信服的理由需要这样做，例如，新版本包含针对阻碍当前工作的错误进行的错误修复。新项目开始时是更新为新版本的最佳时机。

MPLAB X 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 X IDE 中任何问题的最好参考。

要获得关于 MPLAB X IDE 及其组件的更新通知，请在 <http://www.microchip.com/pcn> 上 myMICROCHIP Personalized Notification Service（myMICROCHIP 个性化通知服务）的 Development Tools（开发工具）部分进行订阅。

更多详细信息，请参见**支持**。

第 2 章 开始之前

2.1 简介

在使用 MPLAB X IDE 之前，必须执行以下操作：

- 安装 JRE 和 MPLAB X IDE
- 安装 USB 设备驱动程序（对于硬件工具）
- 连接目标（对于硬件工具）
- 安装语言工具
- 启动 IDE
- 启动 IDE 的多个实例

2.2 安装 JRE 和 MPLAB X IDE

在安装 MPLAB X IDE（基于 NetBeans 平台）时，需要安装对应于 Windows 操作系统（OS）或 Linux OS 的正确 Java 运行时环境（JRE）。对于 Mac 操作系统，如果已经安装了正确的 JRE，MPLAB X IDE 安装将继续进行。如果未安装，则将通过一个对话框提示您获取正确的版本。按照说明进行操作，安装 JRE，然后再安装 MPLAB X IDE。

2.3 安装 USB 设备驱动程序（对于硬件工具）

为了进行正确的工具操作，可能需要安装 USB 驱动程序。

2.3.1 Mac 或 Linux 操作系统的 USB 驱动程序安装

在 Mac 或 Linux 计算机上安装 MPLAB X IDE 时，安装程序会为您安装 USB 驱动程序。您不需要执行任何操作。

2.3.2 Windows® XP/7/8 操作系统的 USB 驱动程序安装

如果在使用 Windows OS 的个人计算机上安装 MPLAB X IDE，需要按照下面的说明来正确安装 USB 驱动程序。（MPLAB IDE v8.xx 的 USB 硬件工具驱动程序与 MPLAB X IDE 的不同。）

这些说明适用于以下工具：

- MPLAB REAL ICE 在线仿真器
- MPLAB ICD 3 在线调试器
- MPLAB PM3 器件编程器
- PIC32 入门工具包

对于 PICKit 2、PICKit 3 或其他 MPLAB 入门工具包，不需要执行任何操作。

按照下面的说明来确定您的安装方法。

2.3.2.1 在安装驱动程序之前

无论是使用 Switcher 实用程序还是激活预安装程序来安装驱动程序（都在后面几节中介绍），都需要注意，如果系统的 WinUSB 驱动程序版本早于 Switcher 或预安装程序的版本，则将会替换它们。如果希望保留您的 WinUSB 驱动程序版本，请在安装任何 Microchip 设备驱动程序之前重命名这些文件。

WinUSB 驱动程序文件位于以下位置：

32 位操作系统

C:\Windows\system32\WinUSB.dll （32 位）

C:\Windows\system32\drivers\WinUSB.sys （64 位）

64 位操作系统

C:\Windows\SysWOW64\WinUSB.dll （32 位）

C:\Windows\system32\WinUSB.dll （64 位）

C:\Windows\system32\drivers\WinUSB.sys （64 位）

2.3.2.2 如果具有 WINDOWS XP 64，请手动切换

如果使用的是 Windows XP 64 位 OS，则必须手动切换设备驱动程序。请参见第 2.3.2.6 节“如果需要手动安装驱动程序”。

2.3.2.3 如果具有 WINDOWS 7 或 8，请使用管理员模式

如果使用 Switcher 可执行文件来安装设备驱动程序，则在 Windows 7 或 8 上，必须在管理员模式下运行该程序。

要以管理员身份运行 Device Driver Switcher GUI（设备驱动程序切换器 GUI）应用程序，请右键单击可执行文件（MPDDSwitch.exe 或 MPDDSwitch64.exe），并选择“以管理员身份运行”。

建议先使用 GUI 应用程序来切换驱动程序。如果这发生问题，可以使用命令行应用程序来切换驱动程序。

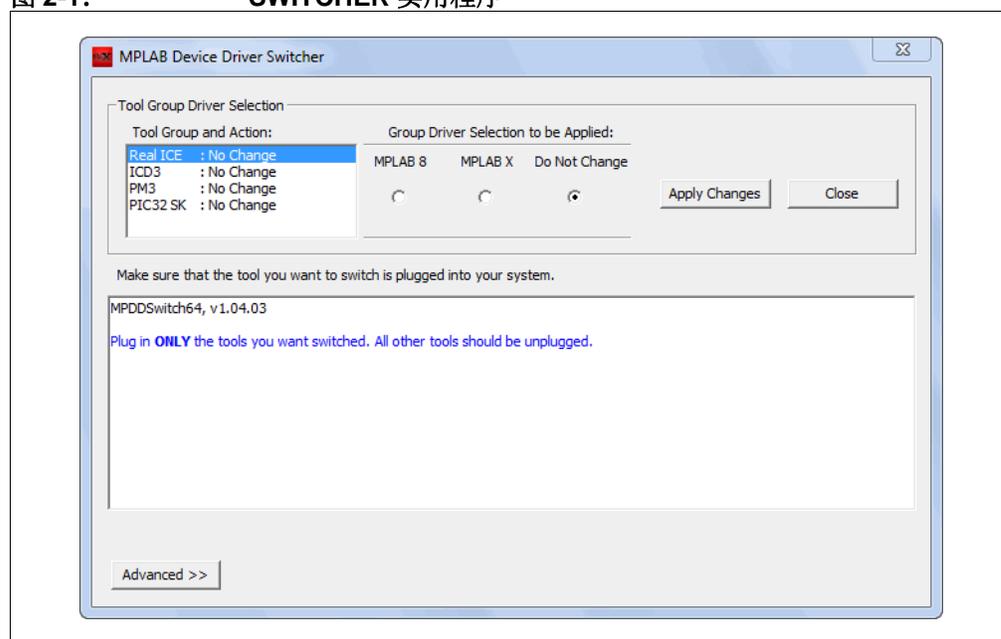
要以管理员身份运行命令行应用程序（mchpdds32.exe 或 mchpdds64.exe），先以管理员模式打开命令提示符：*开始 > 所有程序 > 附件 > 命令提示符*，单击右键并选择“以管理员身份运行”。这将打开“Administrator: Command Prompt”。在此之后，可以按照 ReadMe32.txt 或 ReadMe64.txt 文件中提供的说明执行操作，完成驱动程序切换。

2.3.2.4 如果系统上已经安装了 MPLAB IDE v8.XX

如果计算机上已经安装了 MPLAB IDE v8.xx 或更早版本，则需要运行 Switcher 程序从 MPLAB IDE v8 驱动程序切换到 MPLAB X IDE 驱动程序，后者将在首次进行切换时安装。

Switcher 程序是称为 MPDDSwitch（32 位 OS）或 MPDDSwitch64（64 位 OS）的 GUI 应用程序。提供 MPLAB Driver Switcher 的桌面图标。

图 2-1: SWITCHER 实用程序



1. 将所需工具插入个人计算机上的 USB 连接器。
2. 打开计算机的“设备管理器”窗口。例如，在运行 Windows XP 的个人计算机上，右键单击“我的电脑”图标并选择“属性”。在“系统属性”对话框中，单击**硬件**选项卡，然后单击**设备管理器**按钮。
3. 展开“Microchip Tools”（Microchip 工具）部分来查看工具的当前驱动程序。名称将为以下形式：**Microchip 工具名称**。
4. 转至 MPLAB X IDE 安装文件夹，并查找 Switcher 文件夹，默认情况下位于：
 - a) 32 位操作系统：C:\Program Files\Microchip\MPLABX\Switcher
 - b) 64 位操作系统：C:\Program Files (x86)\Microchip\MPLABX\Switcher
5. 在 Switcher 文件夹下，转至对应于操作系统的文件夹：32Bit 或 64Bit。
6. 启动 MPDDSwitch.exe（32 位操作系统）或 MPDDSwitch64.exe（64 位操作系统）。
7. 如果 MPLAB IDE v8 或 MPLAB X IDE 未安装到默认目录中，请单击**高级**来指定驱动程序文件的位置。
8. 要安装或切换 USB 驱动程序：
 - a) 在“Tool Group and Action”（工具组和操作）下单击，选择要为之切换驱动程序的**已连接**工具。
 - b) 单击“MPLAB 8”或“MPLAB X”的单选按钮。
 - c) 单击**Apply All（全部应用）**。大文本窗口中将显示切换进度。这可能需要一些时间。

注： 如果在 Switcher 运行时工具未连接，将不会为相应工具安装或切换驱动程序。

9. 如果 GUI 无法安装驱动程序，则通过单击**高级**检查驱动程序文件的路径。再次运行 Switcher。
10. 如果 GUI 仍然无法安装驱动程序，则需要手动安装驱动程序。关于说明和驱动程序位置，请参见第 2.3.2.6 节“**如果需要手动安装驱动程序**”。
11. 当程序/批处理完成时，在“设备管理器”窗口中查看驱动程序的名称。应为“Microchip WinUSB Device”。

安装 MPLAB X IDE 驱动程序之后，可以在 MPLAB IDE v8.xx 和 MPLAB X IDE 之间来回切换驱动程序。

2.3.2.5 如果系统上未安装 MPLAB IDE V8.XX

不需要执行任何操作；在安装 MPLAB X IDE 时，将预装 USB 驱动程序。将工具插入计算机 USB 端口时，将显示“发现新硬件”通知。然后，安装将自动继续，或者您必须按照向导执行操作并选择“自动选择驱动程序”。但是，如果该过程无法安装驱动程序，则需要手动安装驱动程序。关于说明和驱动程序位置，请参见第 2.3.2.6 节“**如果需要手动安装驱动程序**”。

2.3.2.6 如果需要手动安装驱动程序

如果需要手动安装驱动程序：

1. 打开“设备管理器”（“控制面板”下）。在“Microchip Tools”下查找您的工具，如果无法在其中找到它，则在“其他设备”下查找“未知设备”。
2. 右键单击您的工具名称或“未知设备”，并选择“更新驱动程序”。
3. 在“更新驱动程序”对话框中，选择“浏览计算机以查找驱动程序软件”。

注： 不要选择“自动搜索更新的驱动程序软件”。这将会安装错误的设备驱动程序。如果无意中选择了该项，则后退或退出，并重复这些步骤来安装正确的驱动程序。

4. 查找对应于系统的正确设备驱动程序。设备驱动程序的默认位置为：

C:\Program Files\Microchip\MPLABX\Switcher\32Bit\winusb\
x86\MCHPWinUSBDevice.inf

或

C:\Program Files\Microchip\MPLABX\Switcher\64Bit\winusb\
amd64\MCHPWinUSBDevice.inf

对于 Windows 8，驱动程序位于 x86_Windows8 或 amd64_Windows8 子文件夹中。

5. 如果弹出“Windows 安全”对话框，请选择“安装此驱动程序软件”继续安装驱动程序。

2.3.2.7 工具通信问题

1. 如果使用的是扩展坞或集线器，并在插入工具之后发生问题，则可能需要将该工具直接插入计算机上的 USB 端口。这是 WinUSB 驱动程序的一个已知问题。
2. 如果需要手动重新安装驱动程序，则需要指向 32Bit 或 64Bit 文件夹中的 INF 文件。详情请参见第 2.3.2.6 节“如果需要手动安装驱动程序”。

2.4 连接目标（对于硬件工具）

对于在线调试器和仿真器，请参考以下资料来确定如何将硬件工具与目标连接：

- 开发工具设计建议
- 仿真头规范（如果使用仿真头）
- 工具文档

对于专用编程器，请参见工具文档来了解连接信息。

如果使用 Microchip 演示板、评估工具包或参考设计作为目标，请参见随附文档来了解设置信息。

2.5 安装语言工具

安装 MPLAB X IDE 时，也会安装以下语言工具：MPASM 工具链（不再包含 ASM30 工具链）。

当前存在几个可用于 MPLAB X IDE 的 C 编译器工具套件（编译器、汇编器和链接器等）。请转至 Microchip 网站：

<http://www.microchip.com/xc>

其中可以找到免费编译器（免费版和评估版）和经过代码优化的全功能编译器（标准版和专业版）。

要选择编译器工具套件，请考虑好希望使用的器件，然后选择支持该器件的工具套件。

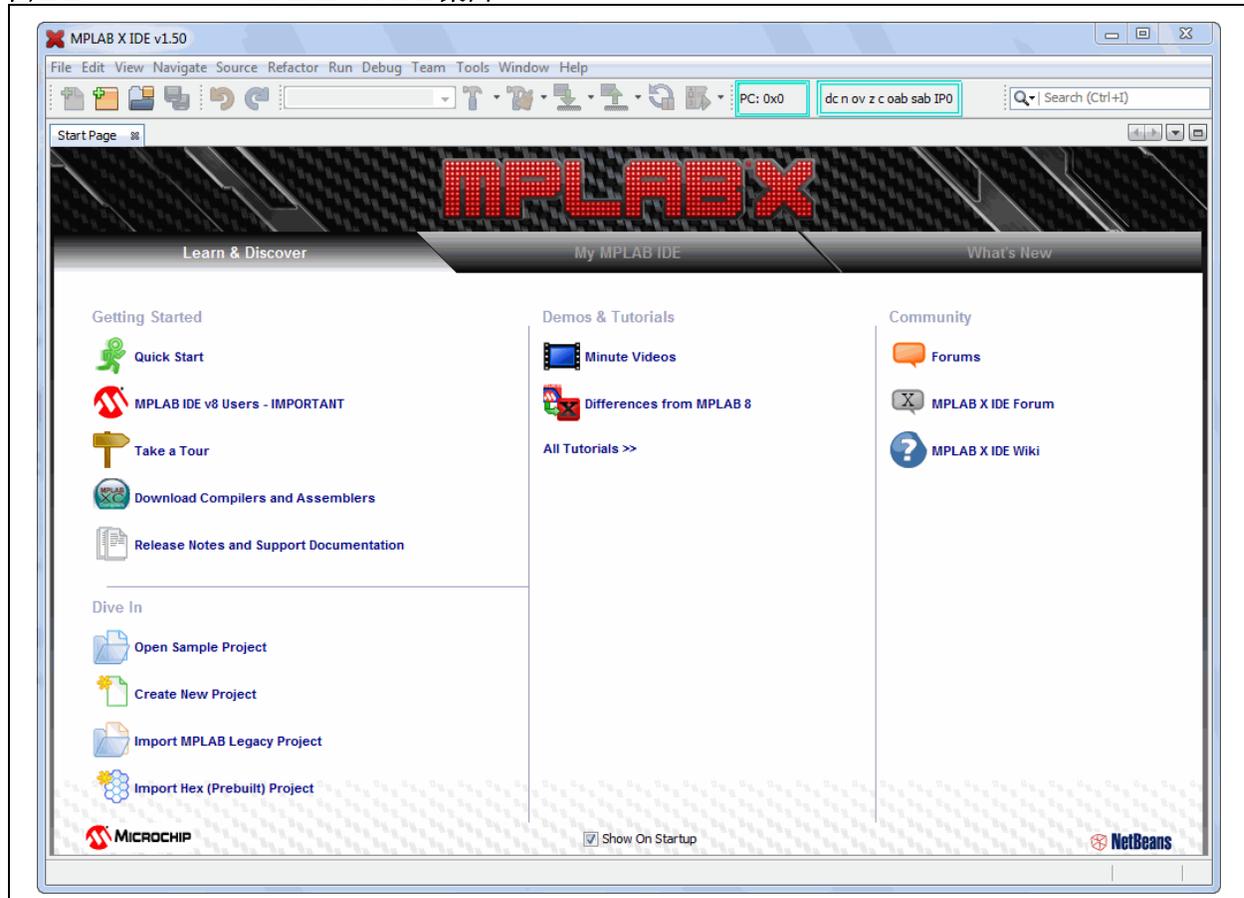
要安装所需的编译器并进行许可授权，请查看文档 *Installing and Licensing MPLAB XC C Compilers*（DS50002059）。MPLAB X IDE 包含了对安装的编译器进行许可授权以及检入和检出网络许可证的选项。请参见第 11.2.10 节“工具菜单”下的“Licenses”（许可证）选项。

2.6 启动 IDE

双击 MPLAB X IDE 图标来启动程序。

MPLAB X IDE 是基于 NetBeans 平台构建的。如果您熟悉 NetBeans IDE，则 MPLAB X IDE 桌面看起来会很熟悉。

图 2-2: MPLAB® X IDE 桌面



在 **Start Page** 中，存在 3 个含有链接的选项卡。每个选项卡上的项定义如下。

表 2-1: LEARN AND DISCOVER (学习和发现)

Getting Started (入门)	
Quick Start (快速开始)	通过创建和设置项目快速开始。
MPLAB® IDE v8 Users IMPORTANT (MPLAB® IDE v8 用户 —— 重要信息)	安装用于硬件工具的 WinUSB 设备驱动程序。
Take a Tour (导览)	MPLAB X IDE 操作的导览。
MPLAB® X IDE Limitations (MPLAB® X IDE 限制)	该版本当前支持和不支持的范围。
Dive In (深入)	
Open Sample Project (打开示例项目)	打开一个现有项目作为示例。
Create New Project (创建新项目)	创建一个新的 MPLAB X IDE 项目。建议在创建第一个项目之前先查看 “Quick Start”。
Import Legacy Project (导入旧项目)	导入现有的 MPLAB IDE v8 项目。建议在使用导入项目之前先查看 “Quick Start”。
Import (Hex) Prebuilt Project (导入 (十六进制) 预编译项目)	从另一个工具的预编译项目中导入十六进制文件。
Demos and Tutorials (演示和教程)	
Minute Videos (小视频)	查看讲解 MPLAB X IDE 操作的短视频。
Difference from MPLAB v8 (与 MPLAB v8 的差异)	查看 MPLAB X IDE 和 MPLAB IDE v8 之间的差异。
All Tutorials (所有教程)	查看所有可用的教程。
Community (社区)	
Forums (论坛)	转至 Microchip 论坛网页。
MPLAB X IDE Forum (MPLAB X IDE 论坛)	注册 MPLAB X IDE 论坛。
MPLAB X IDE Wiki	打开 MPLAB X IDE 开发人员帮助中心。

表 2-2: MY MPLAB X IDE (我的 MPLAB X IDE)

Recent Projects (最近的项目)	
MyProject.c:	最近打开的项目的列表。
Extend MPLAB (扩展 MPLAB)	
Selecting Simple or Full Featured Menus (选择简单或全功能菜单)	在初次启动时，MPLAB X IDE 会显示简单菜单。关于更多功能，请按照这些说明操作。
Install More Plug-Ins (安装更多插件)	打开 Plugins (插件) 对话框。
Notes and Newsletters (笔记和简报)	
ANxxxx, TBxxxx	精选的应用笔记和技术摘要。
microSOLUTIONS E-newsletter (microSOLUTIONS 电子简报)	精选的简报。
All App Notes/ Newsletters (所有应用笔记 / 简报)	查看所有可用应用笔记 / 简报。
References and Featured Links (参考资料和精选链接)	
Data Sheets, etc. (数据手册等)	单击链接可转至所描述的项。

表 2-3: WHAT'S NEW (最新发布)

Data Sheets and Errata (数据手册和勘误表)	
Data Sheet, Silicon Errata (数据手册和硅片勘误表)	精选数据手册和勘误表的列表。要查看所有这些文档的列表，只需单击“ All Data Sheets ”(所有数据手册)或“ All Errata ”(所有勘误表)。
Reference Manuals and Programming Spec (参考手册和编程规范)	
Family Reference Manual, Device Programming Spec (系列参考手册和器件编程规范)	精选参考手册和编程规范的列表。要查看所有这些文档的列表，只需单击“ All Reference Manuals ”(所有参考手册)或“ All Programming Specs ”(所有编程规范)。
Recently Released Software (最近发布的软件)	
Source code (源代码)	支持 Microchip 器件开发的最新软件的列表。要查看所有这些文档的列表，只需单击“ All Recently Released Software ”(所有最近发布的软件)。
Product and Corporate News (产品和公司新闻)	
New stuff, new news (新资料和新新闻)	精选 Microchip 产品和新闻的列表。要查看所有这些文档的列表，只需单击“ All News ”(所有新闻)。

2.7 启动 IDE 的多个实例

在将硬件工具（PICkit 3 等）与 MPLAB X IDE 实例配合使用之前，需要进行一些设置。完成任何硬件工具设置之后，可以从 IDE 实例自己的目录中调用 IDE 实例。

- 设置硬件工具使之与多个实例配合工作
- 调用 IDE 的实例

2.7.1 设置硬件工具使之与多个实例配合工作

默认情况下，最多可以与 5 个 IDE 实例配合工作。如果希望具有更多的实例，则需要手动修改“mchpdefport”文件。

2.7.1.1 “MCHPDEFPORT”文件的使用和格式

“mchpdefport”文件提供让工具可热插拔用于 IDE 和低级 USB 库（DLL、so 或 dylib 文件）所必需的信息。该文件的格式如下：

```
localhost
30000
30002
30004
30006
30008
```

第一行指示 IDE 运行时使用的主机名。

其他行代表低级库与上层 IDE 进行通信所使用的端口或套接字编号。每个 IDE 实例将分配到一个不同的端口或套接字。IDE 实例之间的所有通信都是对用户隐藏的。

最多只有 5 个 MPLAB X IDE 实例时，不需要更改该文件。如果需要 5 个以上的实例，可以通过编辑该文件来添加更多的端口或套接字编号。

2.7.1.2 “MCHPDEFPORT”文件的位置

在 IDE 默认安装中，“mchpdefport”文件位于以下位置：

操作系统	位置
Windows (64 位)	C:\Windows\system32 和 C:\Windows\SysWOW64 (两处的“mchpdefport”文件都必须修改。)
Windows (32 位)	C:\Windows\system32
Linux	/etc/.mplab_ide
Mac (OS X)	/etc/.mplab_ide

2.7.2 调用 IDE 的实例

MPLAB X IDE 要求每个实例都具有自己的用户目录。因此，在一个实例中设置的首选项或添加的插件将不会在另一个中反映。

要调用多个实例，需要使用 `--userdir` 选项指定一个目录来启动 IDE。

2.7.2.1 WINDOWS OS

创建带有 `--userdir` 选项的快捷方式。例如：

1. 在桌面上单击右键并选择 **New (新建) >Shortcut (快捷方式)**。
2. 浏览至已安装的 MPLAB X IDE 可执行文件，默认情况下位于：
"C:\Program Files\Microchip\MPLABX\mplab_ide\bin\
mplab_ide.exe"
3. 在行尾输入：
`--userdir "C:\Documents and Settings\MyFiles\
ApplicationData\mplab_ide\dev\beta7Instance2"`
4. 单击 **OK (确定)**。

2.7.2.2 LINUX OS

不带任何参数运行已安装版本（在桌面图标上单击）将使用 `$(HOME)/.mplab_ide` 用户目录运行。要更改用户目录，请运行 `$InstallationDir/mplab_ide/bin/mplab_ide shell` 脚本，向其传递参数 `--userid anydir`。例如，要在两个不同实例中运行 MPLAB X IDE：

```
$ /opt/microchip/mplabx/mplab_ide/bin/mplab_ide --userdir ~/.anydir1 &  
$ /opt/microchip/mplabx/mplab_ide/bin/mplab_ide --userdir ~/.anydir2 &
```

您可以创建嵌入用户 ID 的桌面图标。

2.7.2.3 MAC OS

打开一个 Shell 窗口，并输入以下命令行在备用用户目录中执行 MPLAB X IDE 安装（示例：Beta 7.12）：

```
$/bin/sh /Applications/microchip/mplabx/712/mplab_ide.app/Contents/  
Resources/mplab_ide/bin/mplab_ide --userdir "${HOME}/Library/  
Application Support/mplab_ide/dev/beta7.12"
```

注:

第 3 章 教程

3.1 简介

本教程提供关于处理 MPLAB X IDE 项目的指导示例。

- 设置硬件和软件
 - 教程工具
 - 安装和设置
- 创建和设置项目
 - 创建新项目
 - 查看桌面上的变化
 - 查看或更改项目属性
 - 设置调试器、编程器或语言工具的选项
 - 设置语言工具位置
 - 向项目中添加现有文件
 - 编辑器用法
 - 配置位
- 运行和调试代码
 - 编译项目
 - 运行代码
 - 调试运行代码
 - 使用断点控制程序执行
 - 单步执行代码
 - 观察符号值变化
 - 查看器件存储器（包括配置位）
 - 对器件编程

3.2 设置硬件和软件

以下信息介绍开始使用 MPLAB X IDE 之前的准备工作。

3.2.1 教程工具

本教程中使用的产品包括：

工具	网页	订购编号
MPLAB® X IDE	http://www.microchip.com/mplabx	免费
MPLAB® XC32 C 编译器 *	http://www.microchip.com/xc	SW006023-1 (标准版本)
MPLAB® REAL ICE™ 在线仿真器	http://www.microchip.com/realice	DV244005
Explorer 16 开发板	http://www.microchip.com/explorer16	DM240001
PIC32MX360F512L PIM	http://www.microchipdirect.com/product-search.aspx?Keywords=MA320001	MA320001

* 您可以从 microchip 网站获取该编译器的免费版或评估版。下载编译器，并且在安装时不要输入许可证编号。

3.2.2 安装和设置

要安装 MPLAB X IDE、设置仿真器（安装 USB 驱动程序并正确地与目标连接），以及安装 32 位语言工具，请参见第 2 章“开始之前”。然后，启动 MPLAB X IDE 并开始本教程。

3.3 创建和设置项目

以下信息介绍在 MPLAB X IDE 中建立和设置项目，这是开发应用程序代码所必需的。

3.3.1 创建新项目

MPLAB X IDE 是基于项目的，所以要开发您的应用程序，必须建立一个项目。

可以通过选择以下任一方式创建新项目：

- **Start** 页面，“**Learn and Discover**”选项卡，“Dive In”部分，“Create New Project”链接
- **File** (文件) > **New Project** (新建项目) (或 Ctrl+Shift+N)

New Project 向导将启动，指导您完成新项目设置。

步骤 1

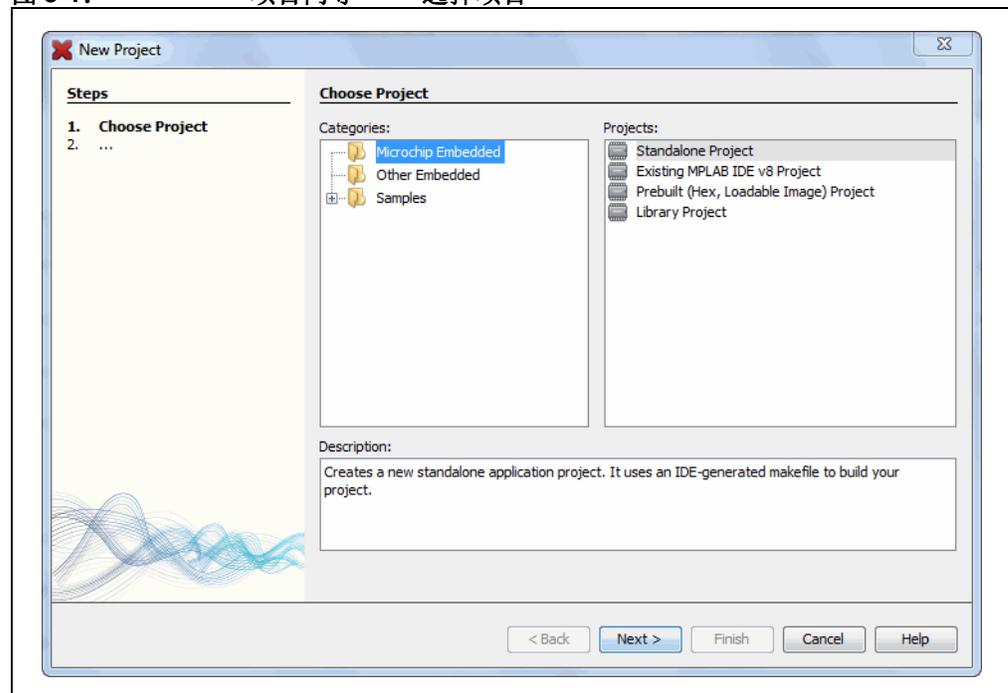
步骤 1 会要求您选择一个项目类别。

这是一个 NetBeans 对话框。要使用 Microchip 产品，请选择“Microchip Embedded”（Microchip 嵌入式）。

接着，选择项目类型。对于本教程，请选择“Standalone Project”（独立项目）。

单击 **Next** (下一步) > 转至下一个对话框。

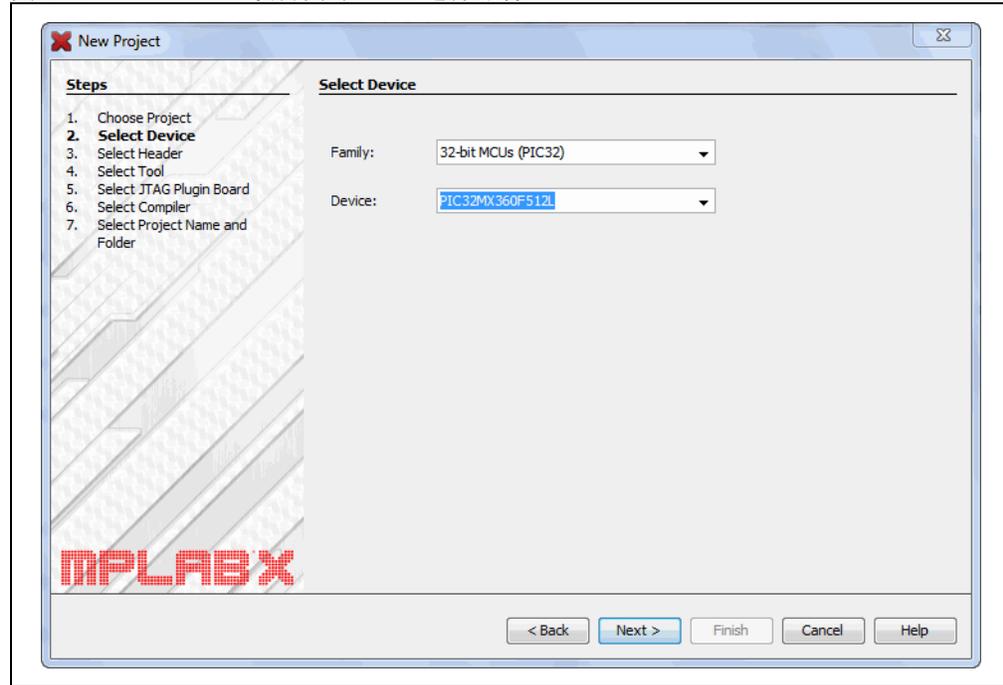
图 3-1: 项目向导 —— 选择项目



步骤 2

步骤 2 选择您使用的器件，在此例中为 PIC32MX360F512L。选择完后单击 **Next>**。

图 3-2: 项目向导 —— 选择器件



步骤 3

只有存在可用于选定器件的仿真头时，才会出现步骤 3。由于不存在用于 PIC32MX360F512L 器件的仿真头，MPLAB X IDE 知道要跳过该步骤。

步骤 4

步骤 4 选择工具。

选定器件的工具支持通过工具名称前的彩色圆形（指示灯）指示。如果看不到颜色，可将鼠标移至指示灯上来弹出关于支持的文本。

指示灯	颜色	支持
●	绿色	完全（已实现并已经过完全测试）
●	黄色	Beta（已实现但未经过完全测试）
●	红色	无

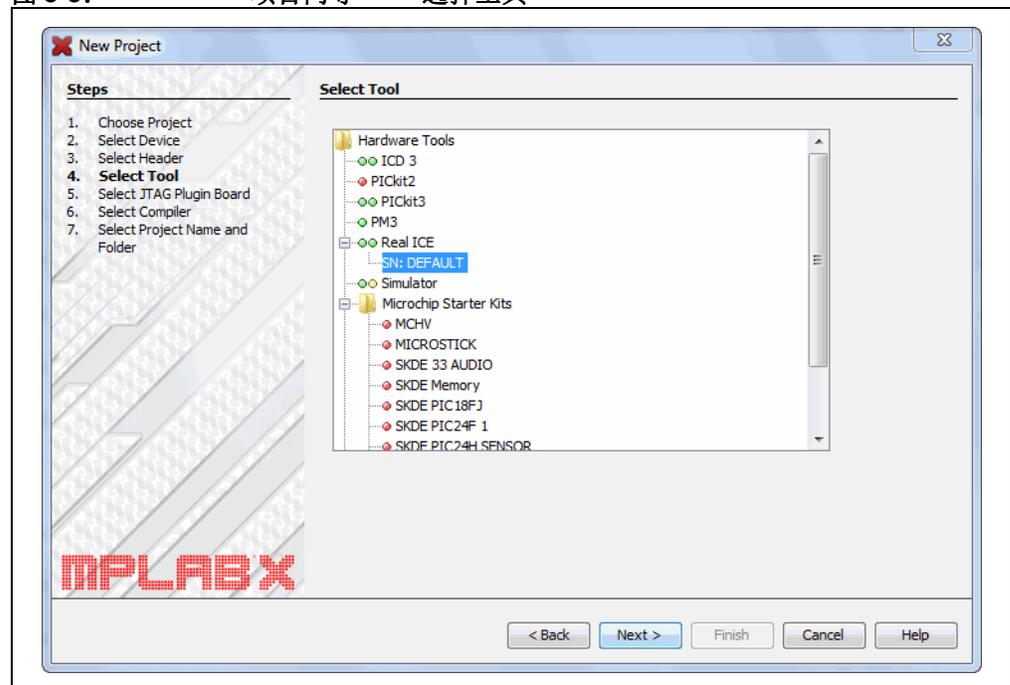
对于一些工具，工具名称旁会有两个指示灯，其中的第一个指示灯是最左边的指示灯，第二个指示灯在第一个指示灯的右边。

指示灯编号	调试工具	软件模拟器
1	调试器支持	内核（指令集）支持
2	编程器支持	外设支持

对于硬件工具，可以注意到在已连接到计算机的所有工具下面都会指定一个序列号（SN）。这使您可以从几个已连接的硬件工具中进行选择。

选择工具，然后单击 **Next>**。

图 3-3: 项目向导 —— 选择工具



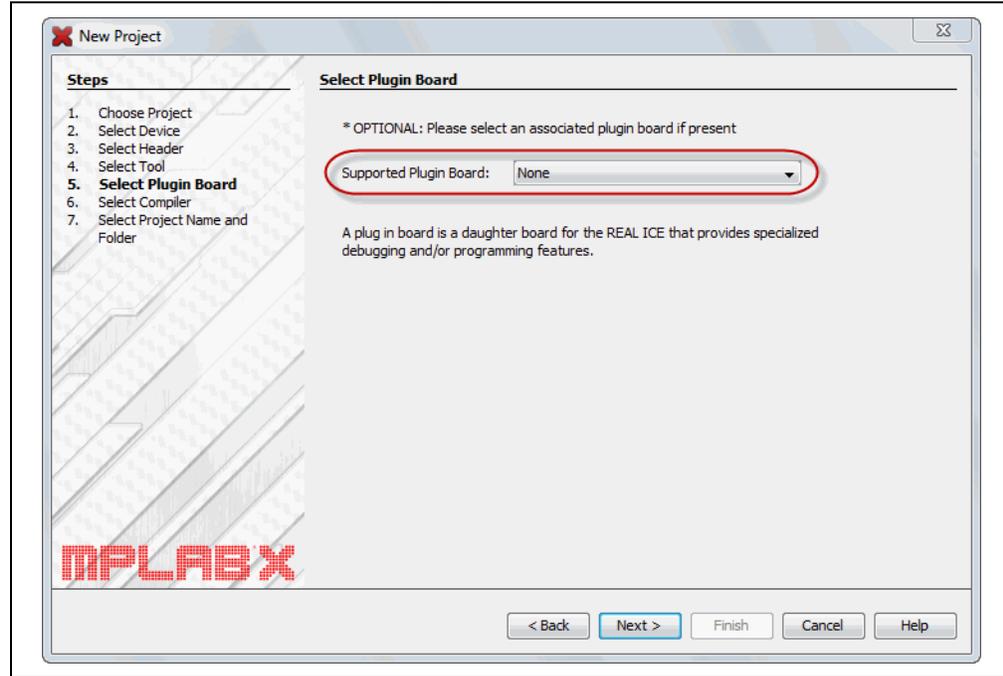
步骤 5

只有选择 MPLAB REAL ICE 在线仿真器作为工具时，才会出现步骤 5。

对于 MPLAB REAL ICE 在线仿真器，可以指定要使用的接插板。接插板是插入到仿真器的驱动板插槽的电路板。由于 Explorer 16 电路板与标准或高速通信驱动板配合工作，所以将“Supported Plugin Board”（支持的接插板）保留为“None”（无）。

选择工具，然后单击 **Next>**。

图 3-4: 项目向导 —— 选择接插板



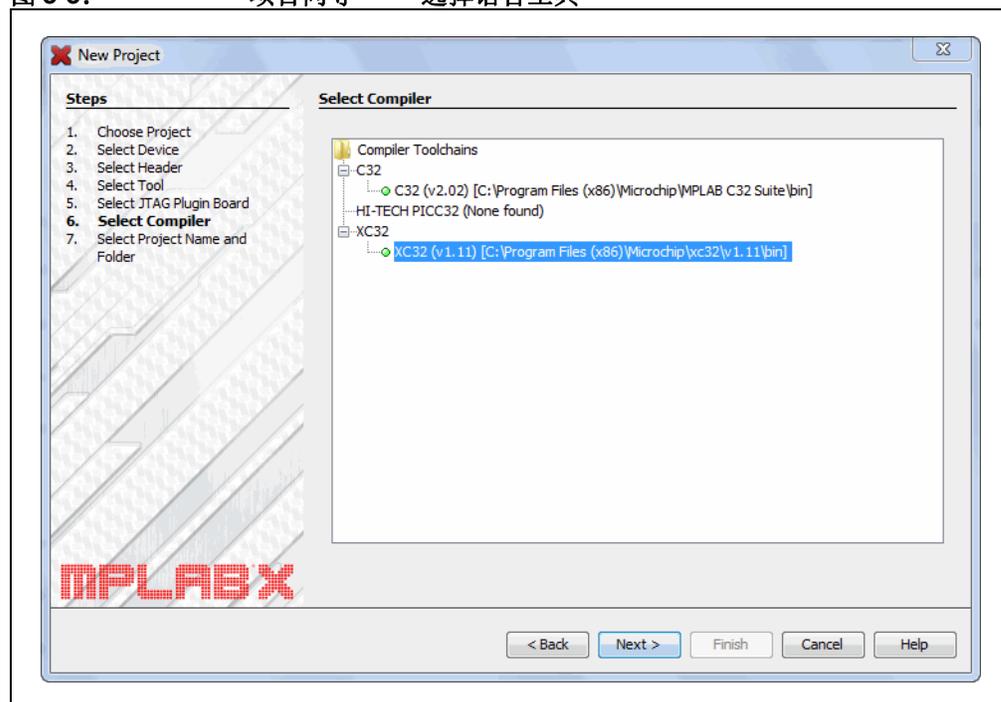
步骤 6

步骤 6 将选择语言工具，即 C 编译器或汇编器。同样，编译器名称前的彩色圆形（指示灯）指示器件支持级别。将鼠标移到其上可显示文本。

语言工具的版本和安装位置将显示在工具下面。这使您可以从几个已安装的语言工具中进行选择。

选择工具，然后单击 **Next>**。

图 3-5: 项目向导 —— 选择语言工具



步骤 7

步骤 7 将选择项目名称、位置和其他项目选项。

输入项目名称 MyProject。

默认情况下，项目将放置在：

- Windows XP——C:\Documents and Settings\UserName\MPLABXProject
- Windows 7/8——C:\Users\UserName\MPLABXProjects
- Linux——/home/UserName/MPLABXProjects
- Mac——/Users/UserName/MPLABXProjects

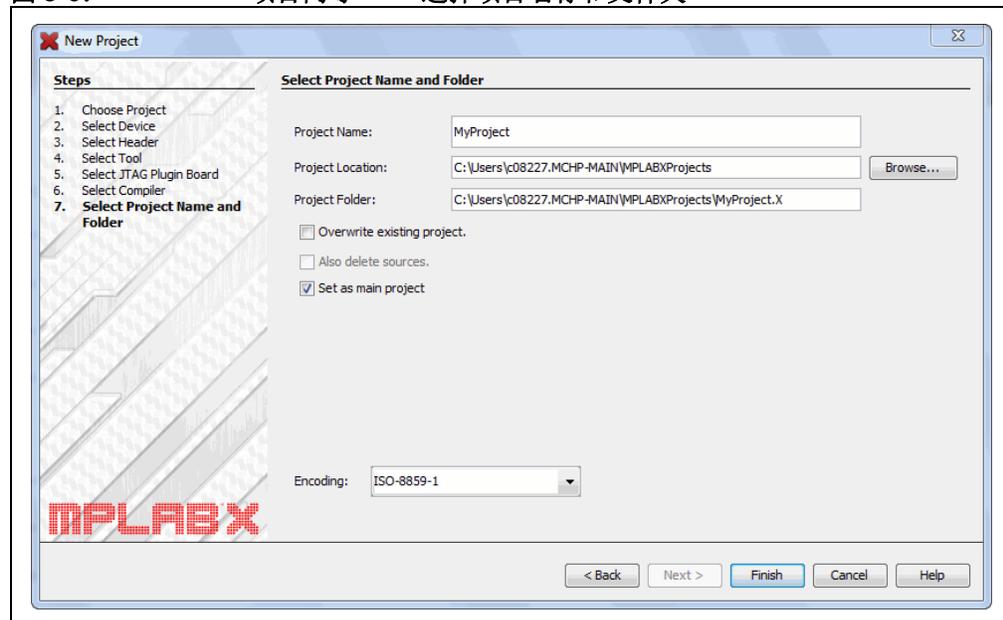
如果项目位置和上面所列不同，则浏览至相应的位置。

选中“Set as main project”（设置为主项目）将该项目设为主项目。

本教程是在编码设置为 ISO-8859-1（Latin 1）的情况下生成的，因此不需要更改该设置。

完成后，选择 **Finish**（完成）以完成新项目创建。

图 3-6: 项目向导 —— 选择项目名称和文件夹

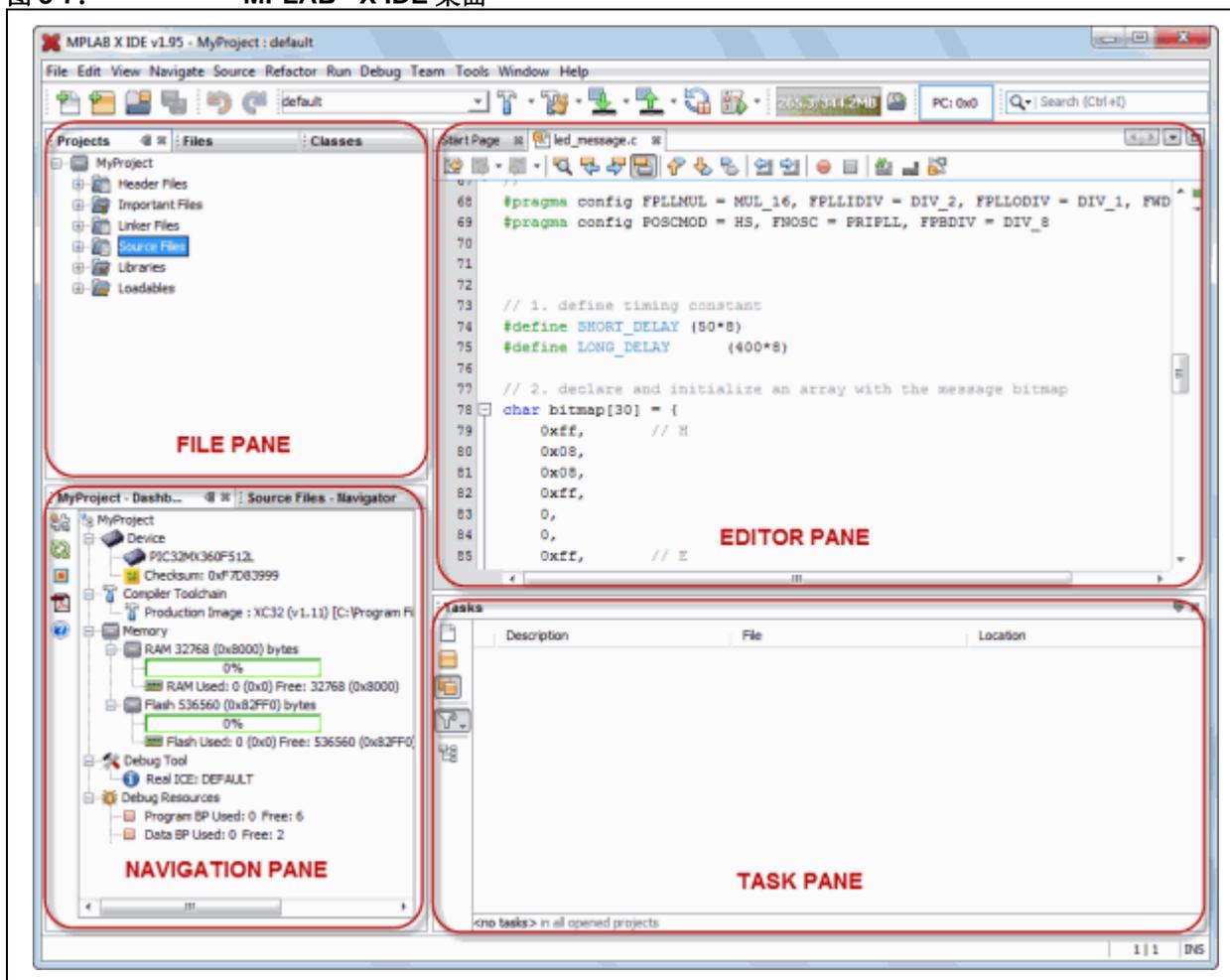


3.3.2 查看桌面上的变化

创建项目之后，将会在 IDE 中打开几个窗格。

- **File（文件）窗格** —— 带有 4 个选项卡式窗口的窗格：**Projects（项目）**、**Files（文件）**、**Classes（类）**和 **Services（服务）**（不自动显示）窗口。在本教程中，将主要关注 **Projects** 窗口，其中会显示文件按类别分组的项目树。
- **Navigation（导航）窗格** —— 显示关于选定文件或项目信息的窗格。对于项目，项目环境将显示关于项目的详细信息；对于文件，将显示符号和变量。
- **Editor（编辑器）窗格** —— 用于查看和编辑项目文件的窗格。**Start Page** 也在此处显示。
- **Task（任务）窗格** —— 显示编译、调试或运行应用程序产生的任务输出的窗格。

图 3-7: MPLAB® X IDE 桌面



如果在文件窗格中双击任何文件名，将会在 **Editor** 窗格中，在 **Start Page** 旁边的选项卡下打开相关文件。要关闭选项卡，请单击文件名旁边的“x”。

在文件窗格的 **Projects** 窗口中右键单击项目名称，可查看弹出（上下文）菜单。对项目子文件夹可执行相同的操作。

3.3.3 查看或更改项目属性

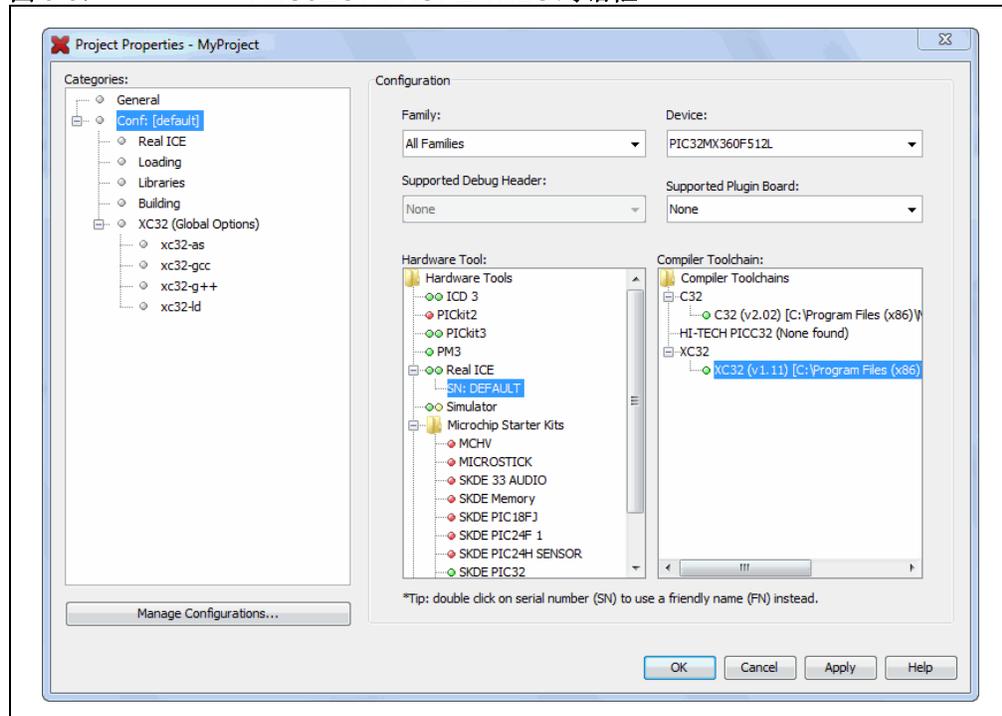
创建项目之后，可以在 **Project Properties**（项目属性）对话框中查看或更改项目属性。通过以下任一方式访问该对话框：

- 在 **Projects** 窗口中右键单击项目名称，然后选择 “**Properties**”（属性）。
- 在 **Projects** 窗口中单击项目名称，然后选择 **File>Project Properties**。

单击 “**Conf: [default]**” 类别可显示一般项目配置，例如项目器件、相关的调试 / 编程器工具和语言工具。对于本教程，除非在前面部分进行了错误的选择，否则不要更改其中任何项。

然后在该对话框中进行更新，并单击 **Apply**（应用）。

图 3-8: **PROJECT PROPERTIES** 对话框



3.3.4 设置调试器、编程器或语言工具的选项

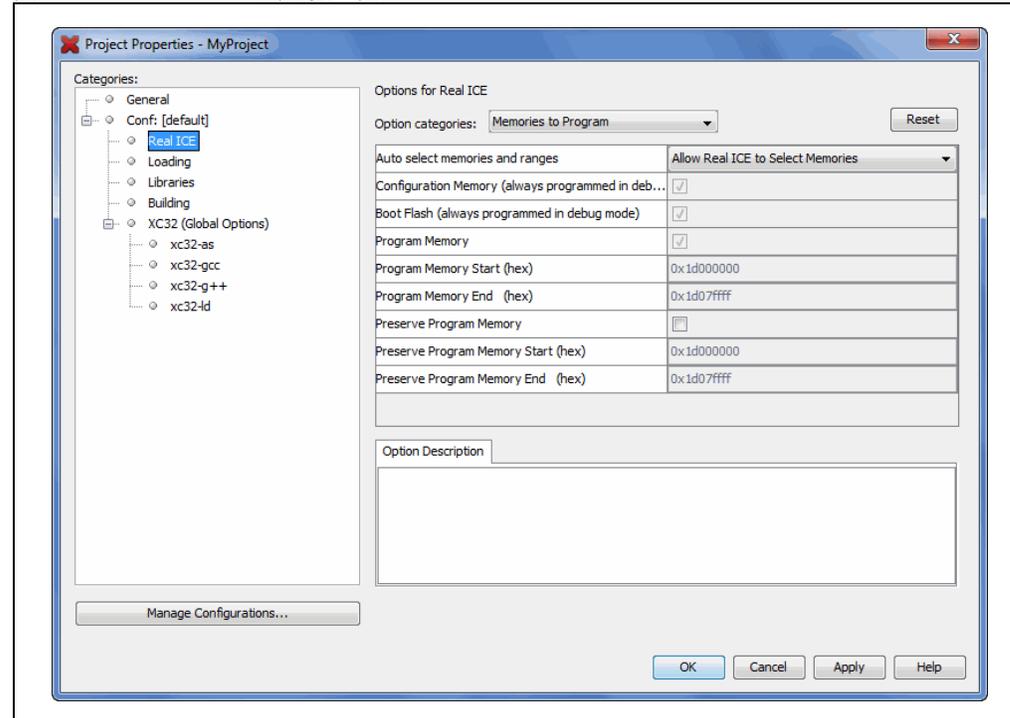
此外，还可以在 Project Properties 对话框中设置工具的选项。

要设置或更改调试器 / 编程器工具选项：

- 单击 REAL ICE 可查看相关的设置选项。关于这些选项含义的更多信息，请参见仿真器文档。

对于本教程，不要进行任何更改。

图 3-9: 工具设置页

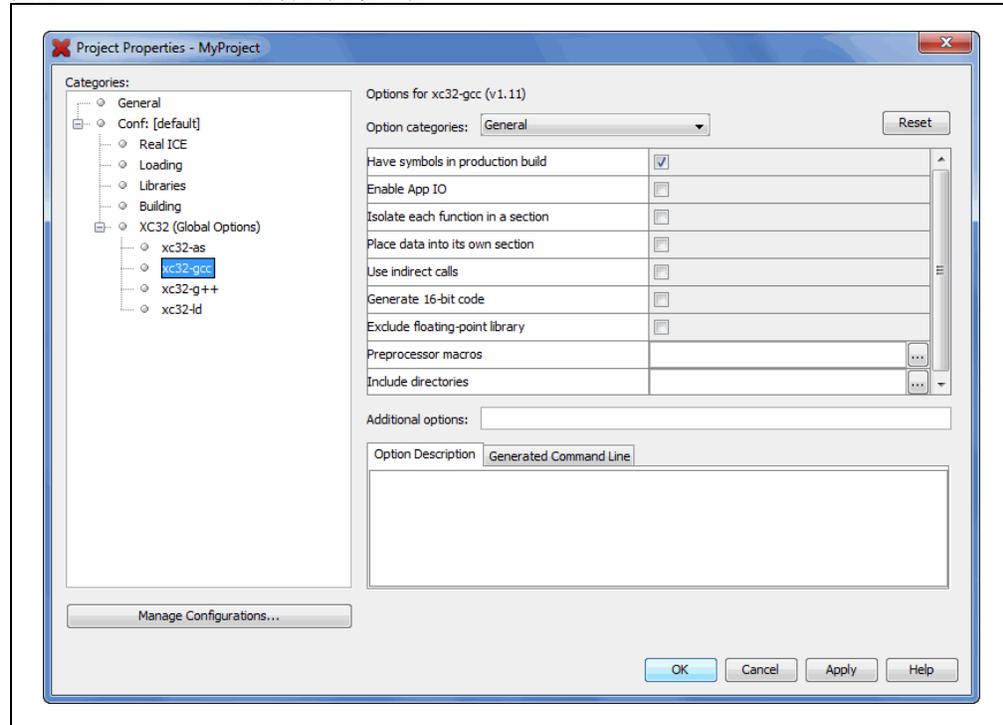


要设置或更改语言工具选项：

- 单击您的语言工具可查看相关的设置选项。关于这些选项含义的更多信息，请参见语言工具文档。

对于本教程，不要进行任何更改。

图 3-10: 语言工具设置页



3.3.5 设置语言工具位置

要查看哪些语言工具可用于 MPLAB X IDE，以及查看或更改它们的路径：

- 对于 **Mac OS**：通过主菜单条中的 *mplab ide*>*Preferences* (首选项) > *Embedded* (已安装工具) > *Build Tools* (编译工具) 访问编译工具。
- 对于其他 **OS**：通过 *Tools* (工具) > *Options* (选项) > *Embedded*>*Build Tools* 访问编译工具。

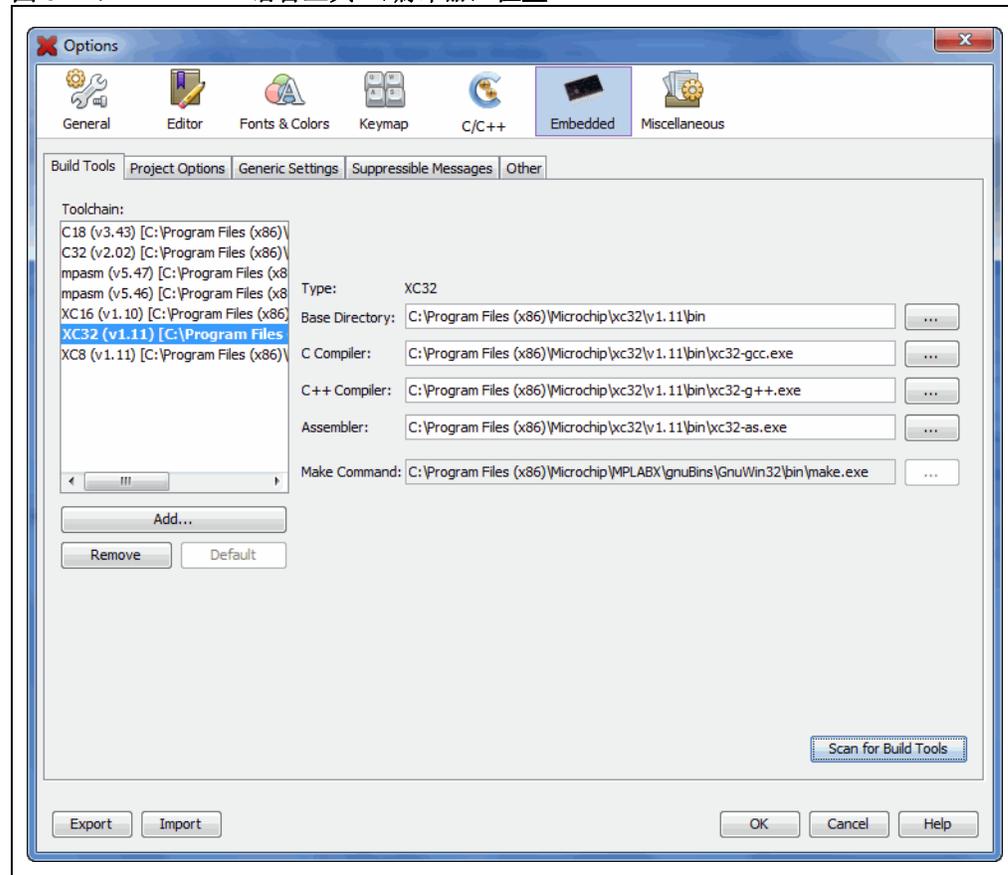
窗口将自动填充所有已安装的工具链。如果发现其中未列出您的工具，请尝试以下操作：

- **Scan for Build Tools** (扫描编译工具) —— 扫描环境路径并列出生计算机上已安装的语言工具。
- **Add** (添加) —— 通过输入包含工具可执行文件的目录的路径（即基目录），将工具手动添加到列表中。通常，它是工具安装目录中的 bin 子目录。

如果有多个编译器版本可用，则从列表中进行选择。

确保为本教程选择 XC32 工具链。

图 3-11: 语言工具（编译器）位置



3.3.6 向项目中添加现有文件

对于本教程，您将使用现有的代码示例：

- 转至 Microchip 网站的 Explorer 16 开发板网页：
<http://www.microchip.com/explorer16>
- 单击“PIC32 Explorer 16 LED Example Application”（PIC32 Explorer 16 LED 示例应用程序）下载包含示例代码的 ZIP 文件。
- 下载之后，解压缩该项目。
- 将文件 `led_message.c` 移至项目目录（`MyXProject.X`）。

现有文件可以通过执行以下操作之一添加到项目中：

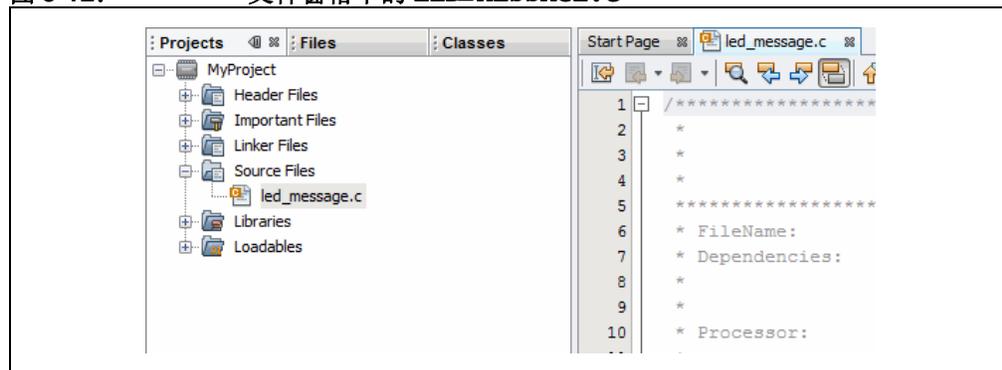
- 在 **Project/File** 窗口中右键单击项目，并选择“Add Existing Item”（添加现有项）
- 在 **Project/File** 窗口中右键单击逻辑文件夹（例如，**Source Files**（源文件）），并选择“Add Existing Item”

添加文件时，可以选择它的添加方式：

- **Auto**（自动）—— 让 MPLAB X IDE 决定定位文件的最佳方式。
- **Relative**（相对）—— 指定相对于项目的文件位置。（建议：相对路径有助于将项目移至另一台计算机时的可移植性。）
- **Absolute**（绝对）—— 通过绝对路径指定文件位置。
- **Copy**（复制）—— 将指定文件复制到项目文件夹中。

该文件将显示在文件窗格中的指定项目下，**Editor** 窗格中将会显示具有该文件名称的选项卡。

图 3-12: 文件窗格中的 `LED_MESSAGE.C`



led_message.c 的代码清单:

```

/*****
 *
 *           Message via 8 LEDs
 *
 *****/
 * FileName:      led_message.c
 * Dependencies:p32xxxx.h
 *
 *
 * Processor:     PIC32
 *
 * Compiler:      MPLAB C32
 *                MPLAB IDE v8.0+
 * Company:       Microchip Technology, Inc.
 *
 * Software License Agreement
 *
 * The software supplied herewith by Microchip Technology Incorporated
 * (the "Company") for its PIC32 Microcontroller is intended
 * and supplied to you, the Company's customer, for use solely and
 * exclusively on Microchip PIC32 Microcontroller products.
 * The software is owned by the Company and/or its supplier, and is
 * protected under applicable copyright laws.All rights are reserved.
 * Any use in violation of the foregoing restrictions may subject the
 * user to criminal sanctions under applicable laws, as well as to
 * civil liability for the breach of the terms and conditions of this
 * license.
 *
 * THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION.NO WARRANTIES,
 * WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED
 * TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.THE COMPANY SHALL NOT,
 * IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR
 * CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 *
 * $Id:led_message.c 5898 2007-10-23 19:39:48Z rajbhartin $
 *
 *****/

/*
** Message in a bottle
**
** Explorer16 version (long delays)
**
** Thanks to Lucio DiJasio for letting us use this example.
**
** Run this example on Explorer-16 board with PIC32MX PIM.
** Hold the board vertically from the PICTail connector size
** and wave the board back-and-forth to see message "HELLO" on LEDs
*/

#include <p32xxxx.h>

// Config settings
// POSCMOD = HS, FNOSC = PRIPLL, FWDTEN = OFF
// PLLIDIV = DIV_2, PLLMUL = MUL_16
// PBDIV = 8 (default)
// Main clock = 8MHz /2 * 16    = 64MHz
// Peripheral clock = 64MHz /8  = 8MHz

```

```
// Configuration Bit settings
// SYSCLK = 64 MHz (8MHz Crystal/ FPLLIDIV * FPLLMUL / FPLLODIV)
// PBCLK = 8 MHz
// Primary Osc w/PLL (XT+,HS+,EC+PLL)
// WDT OFF
// Other options are don't care
//
#pragma config FPLLMUL = MUL_16, FPLLIDIV = DIV_2, FPLLODIV = DIV_1,
FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL, FPBDIV = DIV_8

// 1. define timing constant
#define SHORT_DELAY (50*8)
#define LONG_DELAY(400*8)

// 2. declare and initialize an array with the message bitmap
char bitmap[30] = {
    0xff, // H
    0x08,
    0x08,
    0xff,
    0,
    0,
    0xff, // E
    0x89,
    0x89,
    0x81,
    0,
    0,
    0xff, // L
    0x80,
    0x80,
    0x80,
    0,
    0,
    0xff, // L
    0x80,
    0x80,
    0x80,
    0,
    0,
    0x7e, // O
    0x81,
    0x81,
    0x7e,
    0,
    0
};

// 3. the main program
main()
{
    // disable JTAG port
    DDPCONbits.JTAGEN = 0;

    // 3.1 variable declarations
    int i;           // i will serve as the index
}
```

```
// 3.2 initialization
TRISA = 0;          // all PORTA as output
T1CON = 0x8030;    // TMR1 on, prescale 1:256 PB

// 3.3 the main loop
while( 1)
{
    // 3.3.1 display loop, hand moving to the right
    for( i=0; i<30; i++)
    { // 3.3.1.1 update the LEDs
        PORTA = bitmap[i];

        // 3.3.1.2 short pause
        TMR1 = 0;
        while ( TMR1 < SHORT_DELAY)
        {
        }
    } // for i

    // 3.3.2 long pause, hand moving back to the left
    PORTA = 0;      // turn LEDs off
    TMR1 = 0;
    while ( TMR1 < LONG_DELAY)
    {
    }
} // main loop
} // main
```

3.3.7 编辑器用法

示例代码不需要进行编辑。但是，在需要编辑代码时，需要使用 NetBeans 编辑器。NetBeans 帮助主题 [IDE Basics \(IDE 基础知识\)](#) > [Basic File Features \(基本文件功能\)](#) 下的目录中提供了关于该编辑器的一般信息。NetBeans 帮助主题 [C/C++/Fortran Development \(C/C++/Fortran 开发\)](#) > [C/C++/Fortran Project Basics \(C/C++/Fortran 项目基础知识\)](#) > [Navigating and Editing C/C++/Fortran Source Files \(浏览和编辑 C/C++/Fortran 源文件\)](#) 下的目录中提供了关于该编辑器的 C 编译器信息。第 7 章“编辑器”下给出了功能的列表。

要使用编辑器功能，请转至：

- Edit (编辑) 菜单 (见第 11.2.2 节“编辑菜单”)。
- 位于每个文件的 Editor 窗口顶部的编辑器工具条。
- 窗口右键 (上下文) 菜单。

图 3-13: 编辑器工具条



3.3.8 配置位

在示例代码中，配置位已经设置。对于您自己的应用程序，您必须设置配置位。

您必须在代码中设置配置位。但在调试会话期间，您可以在 Configuration Bits (配置位) 窗口中临时更改配置位 ([Window \(窗口\)](#) > [PIC Memory Views \(PIC 存储器视图\)](#) > [Configuration Bits](#))。然后，可以通过在窗口中单击右键，并选择“Generate Source Code to Output” (生成源代码到输出) 来导出设置。然后，可以将该代码从 Output (输出) 窗口中复制到代码中。

关于不同器件的配置位设置的总结，请参见附录 A “配置设置汇总”。

3.4 运行和调试代码

以下信息介绍如何使用 MPLAB X IDE 运行或调试代码。

3.4.1 编译项目

对于 MPLAB X IDE，不需要先编译项目再运行或调试。编译属于运行和调试过程的一部分。但是，对于初始开发或重要更改，可能需要确保在尝试运行或调试之前先进行项目编译。

要编译项目：

- 在 Projects 窗口中，右键单击项目名称，并选择“Build”（编译）。您还可以选择“Clean and Build”（清除并编译）在编译之前删除中间文件。
- 单击“Build Project”（编译项目）或“Clean and Build Project”（清除并编译项目）工具条图标。



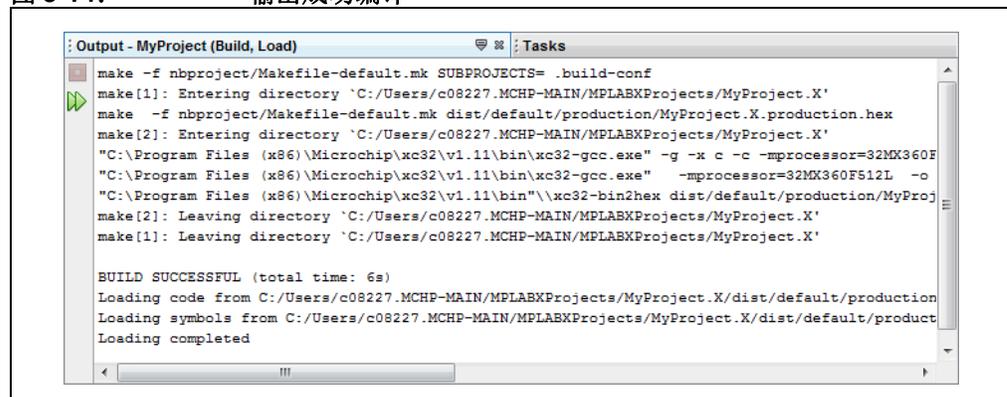
Build 图标



Clean and Build 图标

Output 窗口中将会显示编译进度（桌面的右下角）。对于本教程，代码将会成功编译。

图 3-14: 输出成功编译



```
: Output - MyProject (Build, Load) Tasks
make -f nbproject/Makefile-default.mk SUBPROJECTS= .build-conf
make[1]: Entering directory `C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X'
make -f nbproject/Makefile-default.mk dist/default/production/MyProject.X.production.hex
make[2]: Entering directory `C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X'
"C:\Program Files (x86)\Microchip\xc32\v1.11\bin\xc32-gcc.exe" -g -x c -c -mprocessor=32MK360F
"C:\Program Files (x86)\Microchip\xc32\v1.11\bin\xc32-gcc.exe" -mprocessor=32MK360F512L -o
"C:\Program Files (x86)\Microchip\xc32\v1.11\bin"\xc32-bin2hex dist/default/production/MyProj
make[2]: Leaving directory `C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X'
make[1]: Leaving directory `C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X'

BUILD SUCCESSFUL (total time: 6s)
Loading code from C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X/dist/default/production
Loading symbols from C:/Users/c08227.MCHP-MAIN/MPLABXProjects/MyProject.X/dist/default/product
Loading completed
```

要查看校验和信息：

如果 Dashboard（仪表板）窗口尚未打开，则打开它（[Window>Dashboard](#)）来查看编译后的校验和。

3.4.2 运行代码

成功编译代码之后，可以尝试运行应用程序。单击“**Make and Program Device Project**”（**Make** 项目并对器件编程）图标（或选择 **Run (运行) > Run Project (运行项目)**）来运行程序。



Make and Program Device Project 图标

演示板上的指示灯将会闪烁。来回摆动演示板可看到“Hello”一词。

Output 窗口中也会显示运行进度。

使用 **Hold in Reset (保持复位)** 按钮可使器件在复位和运行之间切换。



Hold in Reset 图标

如果需要，可以向工具条中添加“Run Project”图标（**View (视图) > Toolbars (工具条) > Customize (定制)**）。



Run 图标

3.4.3 调试运行代码

对于本教程，所使用的代码已经过测试并运行。但是，在开发应用程序时，您自己的代码可能需要进行调试。

要调试运行教程代码，请单击“**Debug Project**”（调试项目）图标（或选择 **Debug (调试) > Debug Project** 或 **Debug > Step Into**）开始调试会话。



Debug Run (调试运行) 图标

Output 窗口中将会显示调试运行进度。

要暂停应用程序代码：

单击“**Pause**”（暂停）图标（或选择 **Debug > Pause**）暂停程序执行。

要再次运行代码：

单击“**Continue**”（继续）图标（或选择 **Debug > Continue**）再次启动程序执行。

要结束代码的执行：

单击“**Finish Debugger Session**”（完成调试器会话）图标（或选择 **Debug > Finish Debugger Session**）结束程序执行。

关于调试 C 代码项目的更多详细信息，请参见 NetBeans 帮助主题 **C/C++/Fortran Development > Debugging C/C++/Fortran Applications with gdb (使用 gdb 调试 C/C++/Fortran 应用程序)**。

在使用调试功能时（从第 3.4.4 节“使用断点控制程序执行”开始），Run 和 Debug Run 之间的差异将变得很明显。

3.4.4 使用断点控制程序执行

在调试代码时，通过在代码中特定位置暂停执行来检查变量值的功能会很有用。要执行该操作，请使用断点。

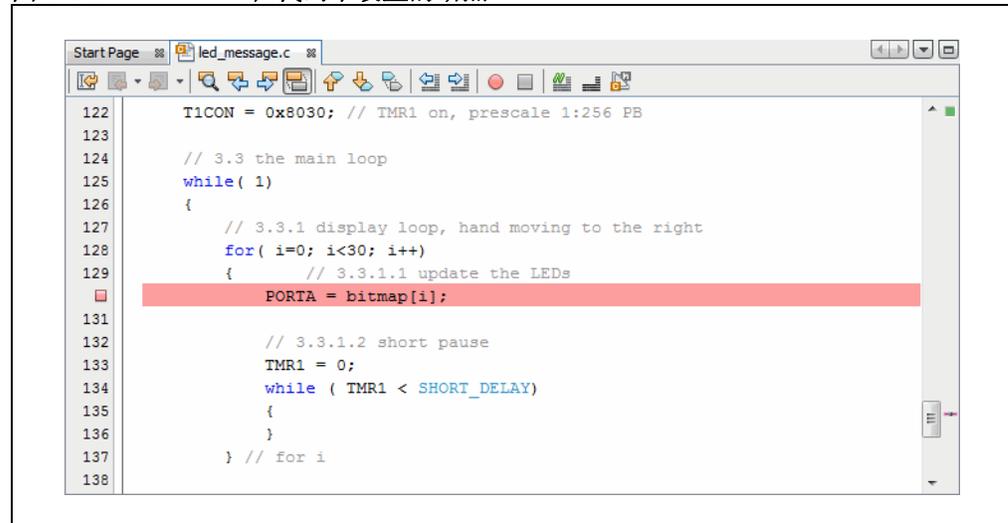
在以下代码行上设置断点：

```
PORTA = bitmap[i];
```

要在某个代码行上设置断点，请执行以下操作之一：

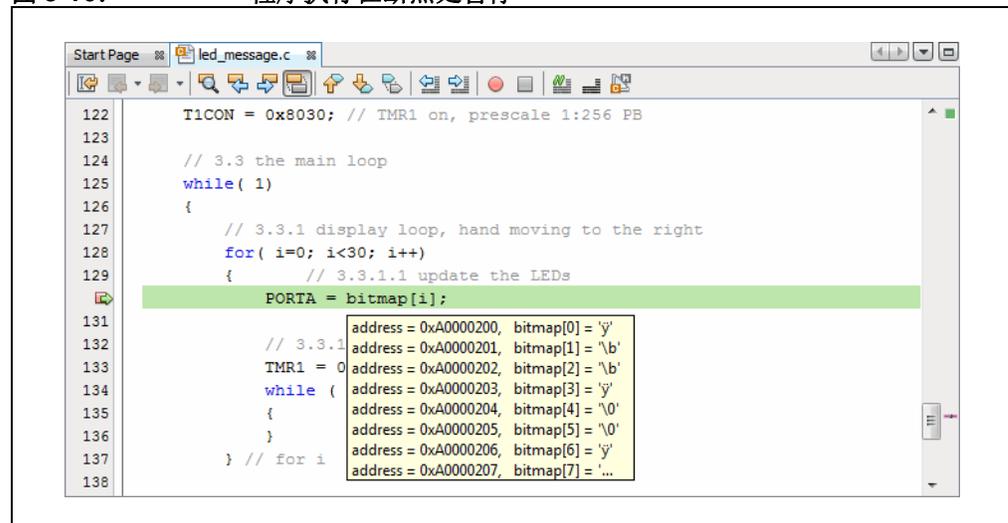
- 在源代码编辑器中单击代码行的左边缘
- 按下 Ctrl-F8

图 3-15: 在代码中设置的断点



再次调试运行教程程序。程序将在断点处暂停。将鼠标悬停在 `bitmap[]` 变量上可查看其值。

图 3-16: 程序执行在断点处暂停



要清除断点，请执行以下操作之一：

- 重复设置断点的步骤
- 选择 *Debug>Toggle Breakpoint* (翻转断点)。

关于断点的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb>Setting C/C++/Fortran Breakpoints](#) (设置 C/C++/Fortran 断点)。

3.4.5 单步执行代码

使用 *Debug* 菜单和 *Debug* 工具条上的单步功能之一，可从代码起始处或断点处暂停之后逐步执行代码。检查变量值的变化（见下一节）或确定程序流是否正确。

有几种方法可以单步执行代码：

- **Step Over**—— 执行程序的一行源代码。如果该行是一个函数调用，则执行整个函数，然后停止。
- **Step Into**—— 执行程序的一行源代码。如果该行是一个函数调用，则程序执行到该函数的第一条语句，然后停止。
- **Step Out**—— 执行程序的一行源代码。如果该行是一个函数调用，则执行函数，并将控制返回给调用方。
- **Run to Cursor** (运行至光标位置) —— 运行当前项目，直到文件中的光标位置，并停止程序执行。

关于单步的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb>C/C++/Fortran Debugging Sessions \(C/C++/Fortran 调试会话\) >Stepping Through Your C/C++/Fortran Program](#) (单步执行 C/C++/Fortran 程序)。

3.4.6 观察符号值变化

在 **Watches**（观察）窗口中观察选择更改的符号的值。在程序执行期间确定这些值是否为预期值可以帮助您调试代码。

要创建新的观察：

1. 选择 **Debug>New Watch**（新建观察）。此时将打开 **New Watch** 对话框（见下图）。
2. 输入观察表达式（在此例中为 **PORTA**），然后单击 **OK**。现在，**Watches** 窗口将显示在桌面上，其中会列出符号。

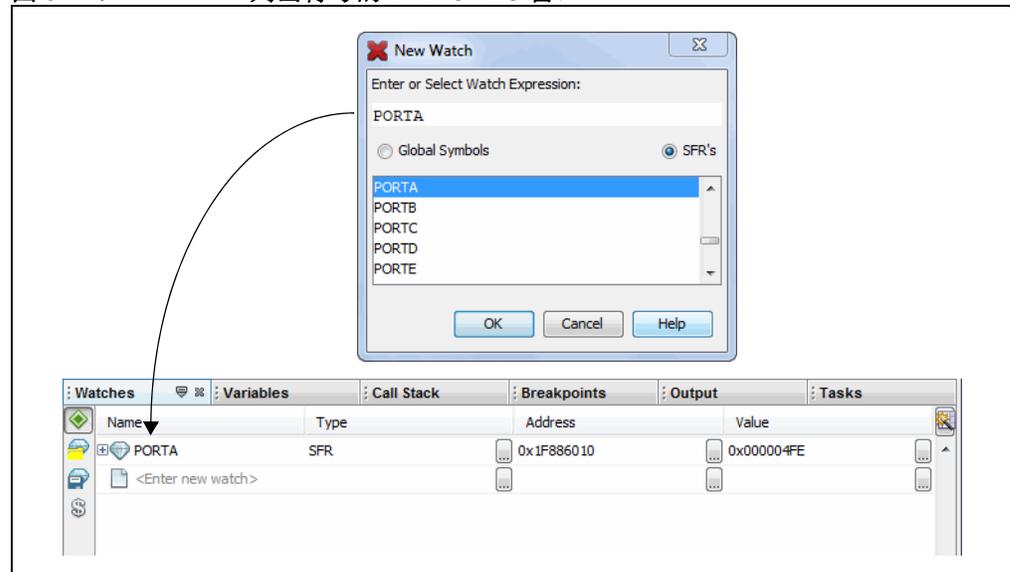
或

1. 在代码中右键单击 **PORTA** 并选择“**New Watch**”。
2. 此时将打开 **Watches** 窗口，其中会列出 **PORTA**。

或

1. 选择 **Window>Debugging**（调试）>**Watches**。
2. 将 **PORTA** 从编辑器窗口拖放到 **Watches** 窗口中。

图 3-17: 列出符号的 **WATCHES** 窗口



要查看符号变化：

1. 调试运行，然后暂停程序。
2. 单击 **Watches** 选项卡可查看窗口和符号值。（红色文本表示发生了变化）。

关于观察的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb>Viewing C/C++/Fortran Program Information](#)（查看 C/C++/Fortran 程序信息）>[Creating a C/C++/Fortran Watch](#)（创建 C/C++/Fortran 观察）。

3.4.7 查看器件存储器（包括配置位）

MPLAB X IDE 具有灵活的、抽象化的存储器窗口，它们提供了不同类型器件存储器的可定制视图。从存储器窗口中，可以在下拉框中选择存储器类型和存储器格式。

通过查看闪存存储器开始：

1. 选择 **Window>PIC Memory Views>Execution Memory**（**执行存储器**）。
2. 此时将打开 Execution Memory 窗口，显示上次暂停位置。

图 3-18: 存储器窗口内容

Line	Address	Opcode	Label	DisAssy
8265	9D00...	34038030		ORI V1, ZERO, -32720
8266	9D00...	AC430600		SW V1, 1536(V0)
8267	9D00...	AFC00000		SW ZERO, 0(S8)
8268	9D00...	0B40005E		J 0x9D000178
8269	9D00...	00000000		NOP
8270	9D00...	3C02A000		LUI V0, -24576
8271	9D00...	24430200		ADDIU V1, V0, 512
8272	9D00...	8FC20000		LW V0, 0(S8)
8273	9D00...	00621021		ADDU V0, V1, V0
8274	9D00...	80420000		LB V0, 0(V0)
8275	9D00...	00401821		ADDU V1, V0, ZERO

通过以下操作更改存储器视图：

- 从 **Window>PIC Memory Views** 列表中选择另一个窗口。
- 使用窗口中的“Memory”下拉菜单。

要设置存储器窗口选项：

在存储器窗口中单击右键将弹出一个菜单，其中具有各种选项，例如显示选项、填充存储器、表导入 / 导出和输出到文件。关于该菜单的更多信息，请参见第 12.8.13 节“存储器窗口菜单”。

要刷新闪存存储器窗口：

1. 暂停程序（Finish Debugger Session）。
2. 单击名为“Read Device Memory”（读取器件存储器）的图标。



Read Device Memory 图标

3.4.8 对器件编程

调试完代码之后，可以将它编程到目标器件上。

首先，在 **Project Properties** 窗口中检查编程选项。对于本教程，不更改任何选项。

其次，以如下两种方式之一编程器件：

- 单击 **Run**：对项目进行编译（如需要）并对器件进行编程。程序将在编程完成后立即开始执行。
- 单击 **Make and Program Device**：对项目进行编译（如需要）并对器件进行编程。程序将不会在编程完成后立即开始执行。在应用程序运行之前，需要先将硬件工具与目标电路板断开。

其他与编程相关的功能包括：

- **Hold in Reset**：使器件在复位和运行之间切换。
- **Read Device Memory**：将目标存储器中的内容传输到 MPLAB X IDE。

与编程相关的图标包括：



Run 图标



Hold In Reset 图标



Make and Program Device 图标



Read Device Memory 图标

注： 不是所有编程功能都包含在 MPLAB X IDE 中。关于更多编程支持，请参见 MPLAB X IDE 安装随附的 MPLAB IPE。

注:

第 4 章 基本任务

4.1 处理 MPLAB X IDE 项目

以下步骤说明如何在 MPLAB X IDE 中处理项目

1
准备工作

1. 开始之前，先安装 MPLAB X IDE，设置所有硬件工具（安装 USB 驱动程序并正确地与目标连接），并安装对应于选定器件的语言工具。然后，启动 MPLAB X IDE 并开始本教程。

2
创建并编译项目

1. 通过使用 **New Project** 向导创建新项目。然后，查看桌面上的变化。
2. 在 **Project Properties** 对话框中查看或更改项目属性。此外，在同一对话框中设置调试器、编程器或语言工具的选项。
3. 在 **Tools Options**（工具选项）对话框中设置语言工具位置和设置其他工具选项。
4. 创建新文件以将其添加到项目中或向项目中添加现有文件。在 **File** 窗口中输入或编辑应用程序代码。
5. 研究编辑器用法的其他功能。
6. 添加、设置库和目标文件。
7. 设置文件和文件夹属性以从编译中保留或排除各个文件或整个文件夹。
8. 为编译前和编译后步骤以及在编译时装入替代的十六进制文件设置编译属性。
9. 编译项目。

3
执行代码

1. 使用 **Run** 菜单运行代码。
2. 使用 **Debug** 菜单调试运行代码。

4
调试代码

1. 使用断点控制程序执行。在行内或通过 **Breakpoint**（断点）窗口设置断点。
2. 在程序执行时单步执行代码。
3. 在 **Watches** 和 **Variables**（变量）窗口中观察符号值变化。
4. 查看 / 更改器件存储器（包括配置位）。存储器类型取决于选定的器件。
5. 使用查看调用堆栈来浏览函数调用。

5
对器件编程

1. 使用简单的工具条按钮对器件编程。

4.2 创建新项目

MPLAB X IDE 是基于项目的，所以要开发您的应用程序，必须建立一个项目。可以通过如下两种方式之一来创建新项目：

- Start Page, “**Learn and Discover**” 选项卡, “Dive In”, “Create New Project”
- *File>New Project* (或 Ctrl+Shift+N)

New Project 向导将启动，指导您完成新项目建立。

4.2.1 步骤 1：选择项目

步骤 1 会先要求您选择一个项目类别。在大多数情况下，都是从 “Microchip Embedded” 选择一个项目类型：

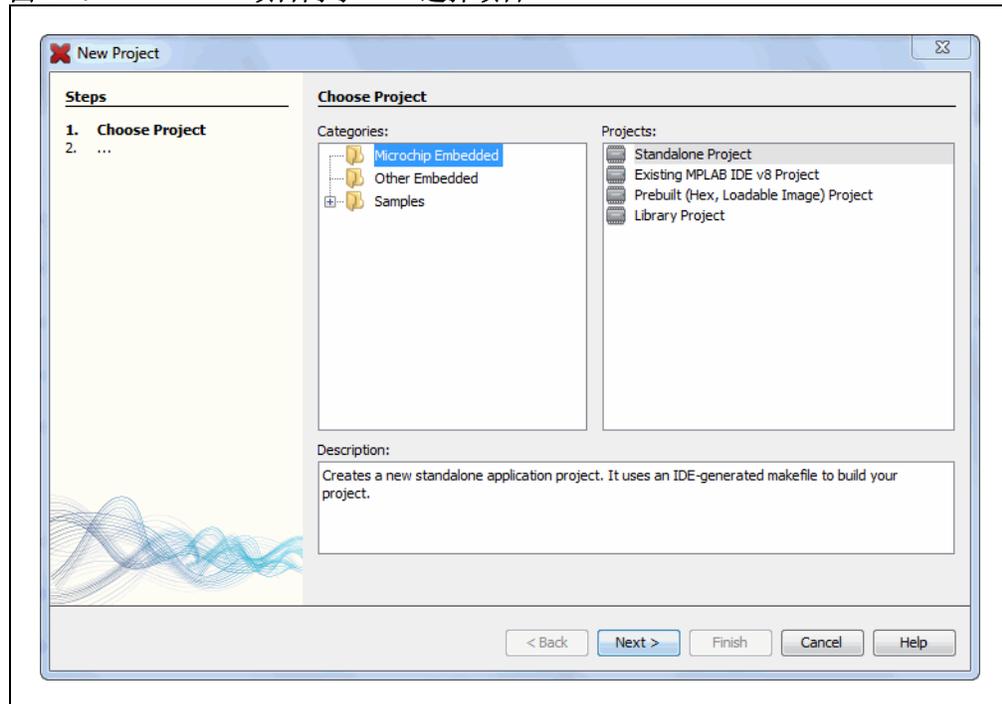
- **Standalone Project**—— 创建新的 C 和 / 或汇编代码项目。这种类型的项目显示在该部分中。
- **Existing MPLAB IDE v8 Project** (现有 MPLAB IDE v8 项目) —— 将现有 MPLAB IDE v8 项目转换为 MPLAB X IDE 项目。详情请参见第 5.2 节 “**导入 MPLAB 旧项目**”。
- **Prebuilt (Hex, Loadable Image) Project** (预编译 (十六进制可装入映像) 项目) —— 将现有项目映像装入 MPLAB X IDE 中。详情请参见第 5.3 节 “**预编译项目**”。
- **Library Project** (库项目) —— 创建将编译为库而不是可执行十六进制文件的新的 C 和 / 或汇编代码项目。详情请参见第 5.6 节 “**库项目**”。

其他选项：

- 其他嵌入式项目 —— 来自其他供应商的项目。
- 示例项目 —— 包含用于不同器件系列的现成项目，以及用于不同器件系列的项目模板。

进行选择之后，单击 **Next>** 移至下一个对话框。

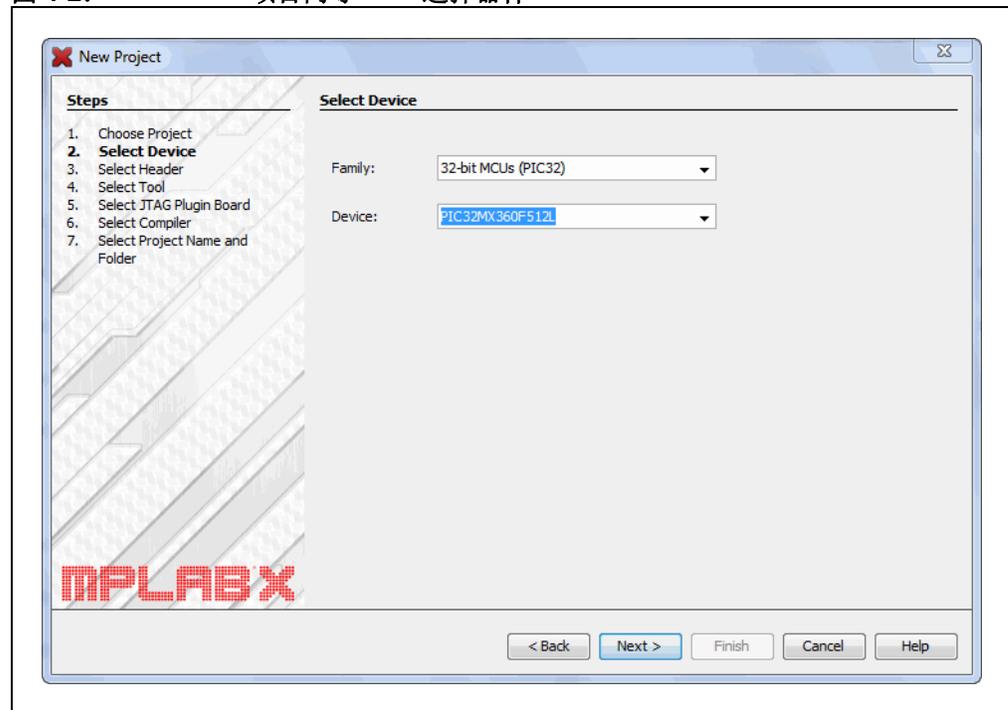
图 4-1： 项目向导 —— 选择项目



4.2.2 步骤 2: 选择器件

在步骤 2 中，从“Device”（器件）下拉列表中选择将在应用中使用的器件。要缩小选择列表，请先选择“Family”（系列）。单击 **Next>**。

图 4-2: 项目向导 —— 选择器件

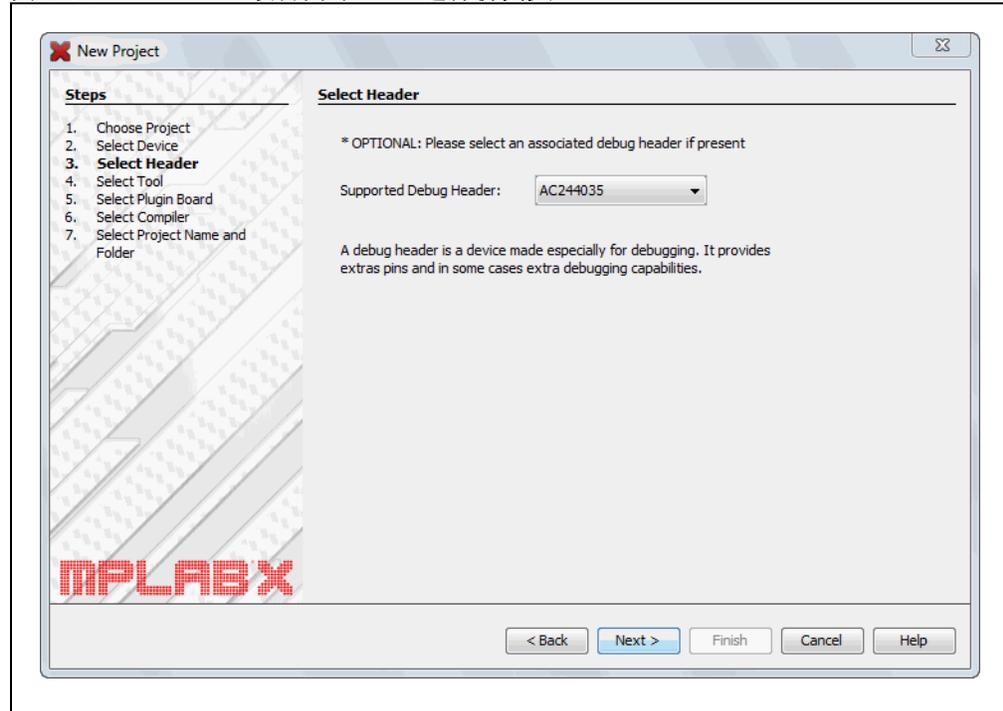


4.2.3 步骤 3：选择仿真头

如果存在可用于选定器件的仿真头，则会出现**步骤 3**。要确定调试是否需要仿真头，或器件是否具有片内调试电路，请参见“*Processor Extension Pak and Header Specification*”（DS51292 或联机帮助）。然后选择是否使用仿真头。完成后，单击**Next>**。

注： 您可以稍后使用 Project Properties 窗口选择仿真头（如可用）。

图 4-3： 项目向导 —— 选择仿真头



4.2.4 步骤 4：选择工具

步骤 4 涉及到选择工具。

选定器件的工具支持通过工具名称前的彩色圆形（指示灯）指示。如果看不到颜色，可将鼠标移至指示灯上来弹出关于支持的文本。

指示灯	颜色	支持
●	绿色	完全（已实现并已经过完全测试）
●	黄色	Beta（已实现但未经过完全测试）
●	红色	无

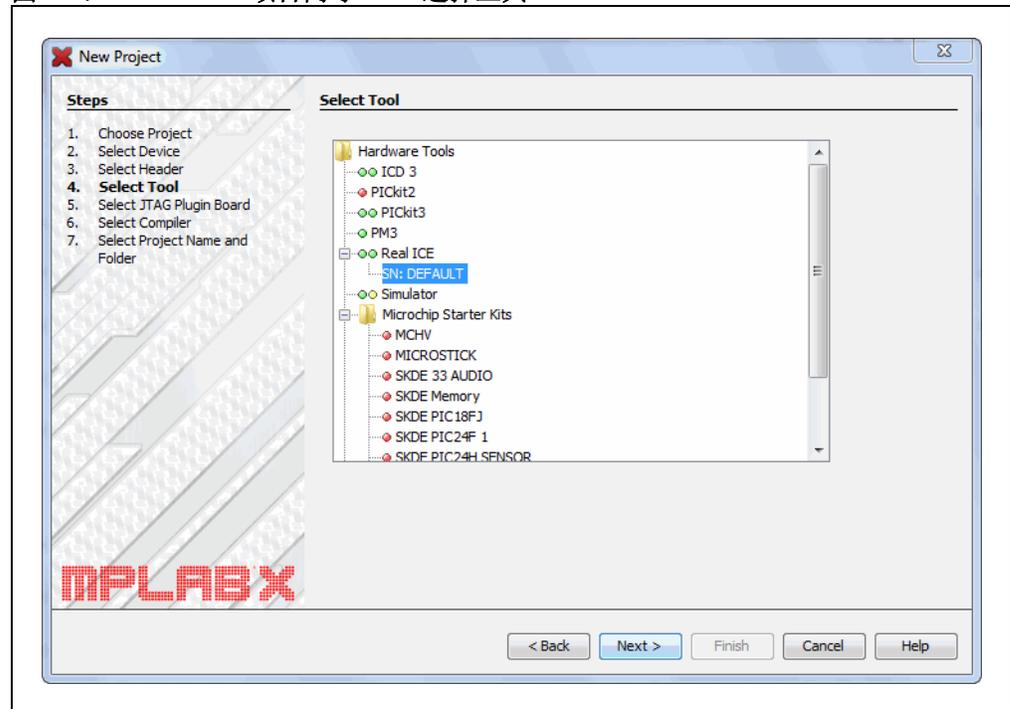
对于一些工具，工具名称旁会有两个指示灯，其中的第一个指示灯是最左边的指示灯，第二个指示灯在第一个指示灯的右边。

指示灯编号	调试工具	软件模拟器
1	调试器支持	内核（指令集）支持
2	编程器支持	外设支持

对于硬件工具，可以注意到在已连接到计算机的所有工具下面都会指定一个序列号（SN）。这使您可以从几个已连接的硬件工具中进行选择。

选择工具，然后单击 **Next>**。

图 4-4：项目向导 —— 选择工具



4.2.5 步骤 5: 选择接插板

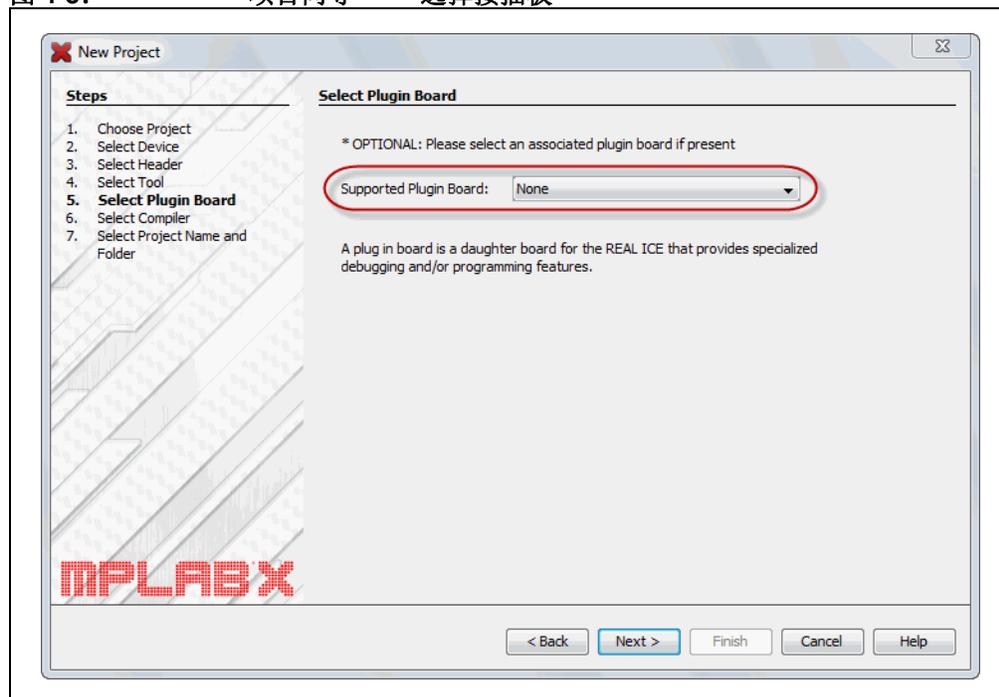
对于 MPLAB REAL ICE 在线仿真器，可以指定要使用的接插板。接插板是插入到仿真器的驱动板插槽的电路板。

表 4-1: 仿真器接插板

支持的接插板	电路板说明
无	标准通信驱动板
无	高速通信驱动板
JTAG 驱动板	JTAG 适配器板
电源监视器板	电源监视器板（也插入逻辑探针连接器）

选择工具，然后单击 **Next>**。

图 4-5: 项目向导 —— 选择接插板



4.2.6 步骤 6: 选择编译器

步骤 6 涉及到选择语言工具，即 C 编译器或汇编器。同样，编译器名称前的彩色圆形（指示灯）指示器件支持级别。将鼠标移到其上可显示文本。

注： 如果发现其中未列出您的语言工具，可以在 *Tools>Options*（对于 Mac OS X 为 *mplab_ide>Preferences*），**Embedded** 按钮，**Build Tools** 选项卡下查找，以确保 MPLAB X IDE 可以找到该工具。如果列出了您的工具，则说明您的工具可能不支持您的项目所使用的器件。请考虑选择或安装支持该器件的其他语言工具。

表 4-2: MICROCHIP 语言工具

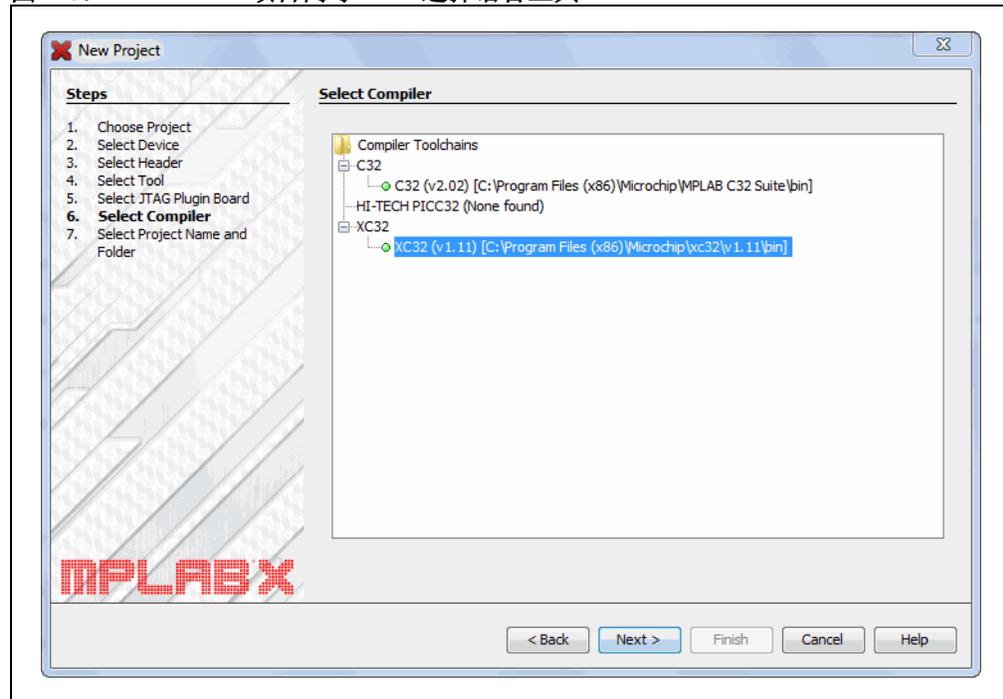
工具链	全名
8 位器件语言工具	
MPASM*	MPASM 汇编器、MPLINK 目标链接器和实用程序
C18*	用于 PIC18 MCU 的 MPLAB [®] C 编译器
HI-TECH PICC	用于 PIC10/12/16 MCU 的 HI-TECH C 编译器
HI-TECH PICC18	用于 PIC18 MCU 的 HI-TECH C 编译器
XC8	MPLAB XC8 C 编译器
16 位器件语言工具	
ASM30**	用于 PIC24 MCU 和 dsPIC [®] DSC 的 MPLAB 汇编器、目标链接器和实用程序
C30	用于 PIC24 MCU 和 dsPIC DSC 的 MPLAB C 编译器
C24	用于 PIC24 MCU 的 MPLAB C 编译器（C30 的子集）
dsPIC	用于 dsPIC DSC 的 MPLAB C 编译器（C30 的子集）
HI-TECH DSPICC	用于 PIC24 MCU 和 dsPIC DSC 的 HI-TECH C 编译器
XC16	MPLAB XC16 C 编译器
32 位器件语言工具	
C32	用于 PIC32 MCU 的 MPLAB C 编译器
HI-TECH PICC32	用于 PIC32 MCU 的 HI-TECH C 编译器
XC32	MPLAB XC32 C 编译器 *
* 大多数编译器都随附汇编器、链接器和实用程序。MPLAB C18 由 MPASM 提供支持。	
** 在 v1.30 中，不再随 MPLAB X IDE 提供。请使用任一 16 位编译器随附的汇编器。	

关于每种语言工具的更多信息，请参见语言工具文档。

对于第三方语言工具链（CCS 等），请参见 Start Page，“Release Notes and Support Documentation”下的“Readme for Third Party Tools.htm”文件。

选择工具，然后单击 **Next>**。

图 4-6: 项目向导 —— 选择语言工具



4.2.7 步骤 7：选择项目名称和文件夹

步骤 7 涉及到选择项目名称、位置和其他项目功能。完成后，选择 **Finish** 以完成新项目创建。

项目名称、位置和文件夹

输入项目名称。默认情况下，名称末尾将带有 `.x`。这不是必需的，只是一项约定。您可以根据需要在“Project Folder”（项目文件夹）文本框中删除它。

浏览至文件夹位置。如果需要，您可以创建一个新的项目文件夹。默认情况下，项目将放置在：

- Windows XP——`C:\Documents and Settings\UserName\MPLABXProject`
- Windows 7/8——`C:\Users\UserName\MPLABXProjects`
- Linux——`/home/UserName/MPLABXProjects`
- Mac——`/Users/UserName/MPLABXProjects`

但是，您可以选择将它们放置在您自己选择的位置。

主项目

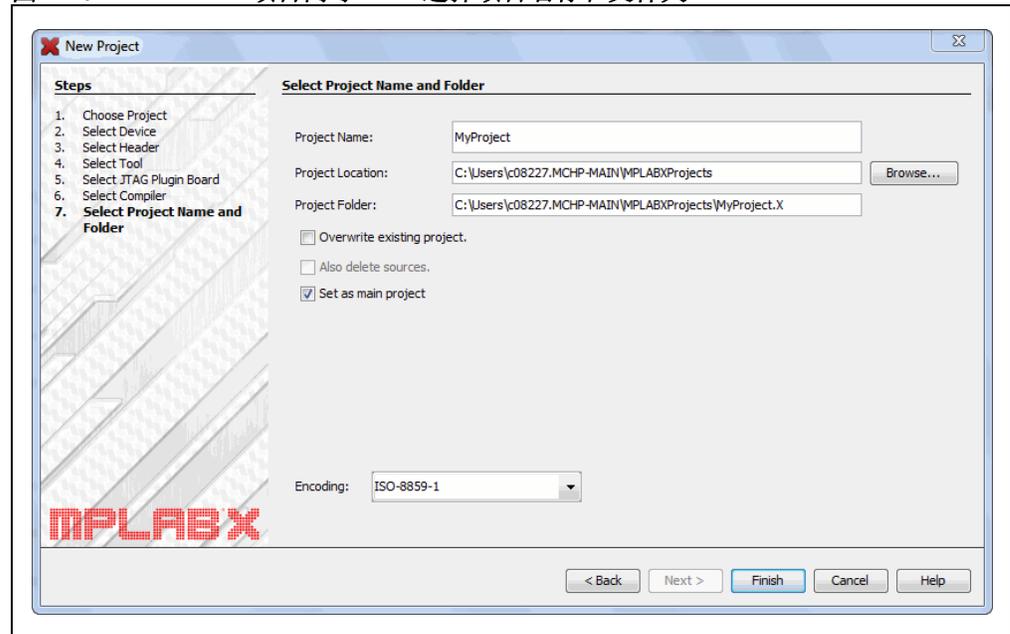
选中“Set as main project”可将该项目指定为主项目。

编码

选择项目的编码。默认为 ISO-8859-1（Latin 1）字符集。

该选择将指定代码的语法着色设置，它可以在 [Tools>Options](#)（对于 Mac OS X 为 [mplab_ide>Preferences](#)），**Fonts and Colors**（字体和颜色）按钮，**Syntax**（语法）选项卡下进行编辑。

图 4-7： 项目向导 —— 选择项目名称和文件夹

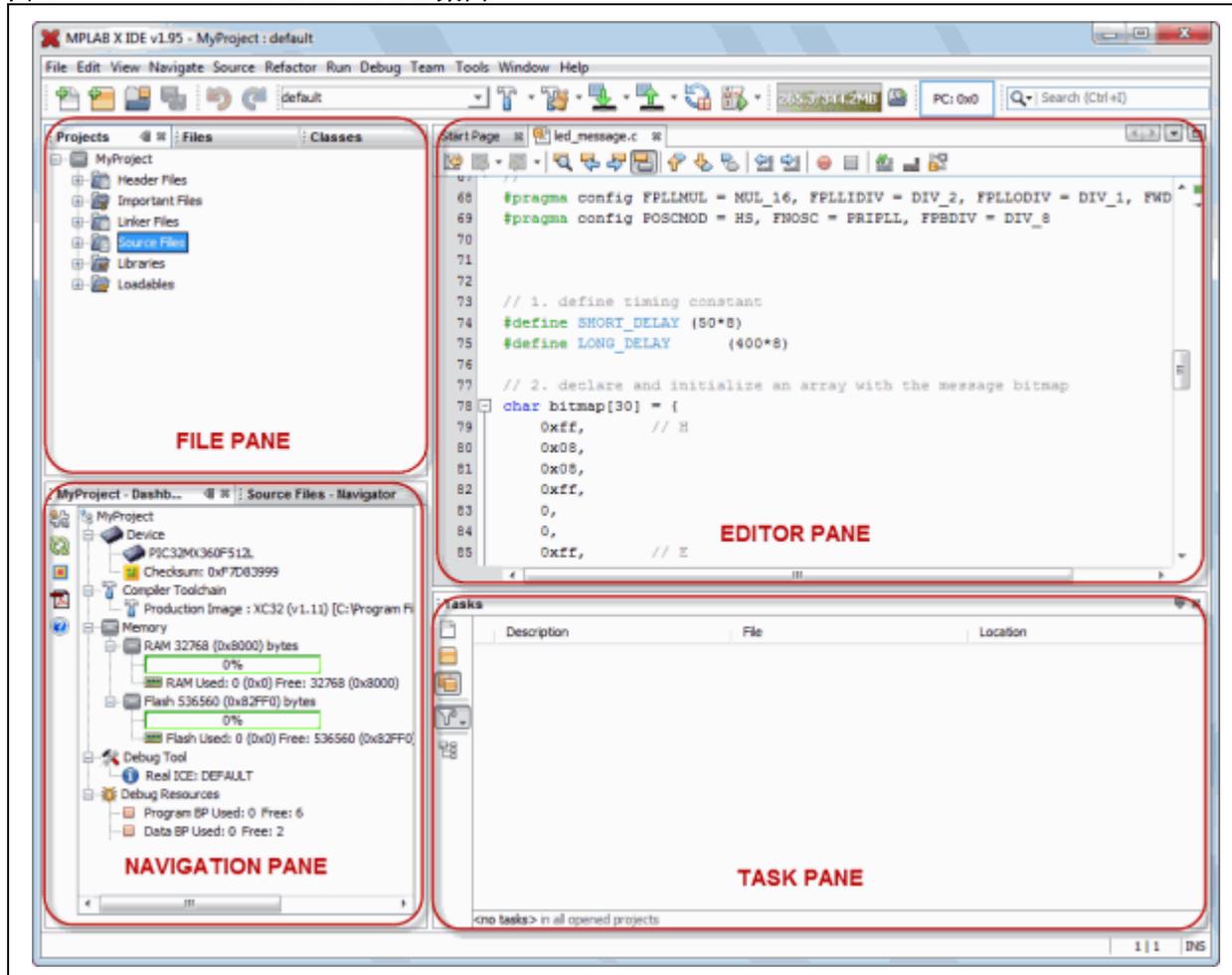


4.3 查看桌面上的变化

创建项目之后，将会在 IDE 中打开几个窗格。

- **File 窗格** —— 具有 4 个选项卡式窗口的窗格：
 - Projects 窗口将显示文件按类别进行分组的项目树。
 - Files 窗口将根据计算机上的文件夹组织来显示项目文件。
 - Classes 窗口将显示代码中的所有类及其函数、变量和常量。双击某个项可查看其声明。
 - Services 窗口将显示可用于代码开发的所有服务。
- **Navigator 窗格** —— 显示关于在文件窗格中选定的文件中的符号和变量的信息。
- **Editor 窗格** —— 用于查看和编辑项目文件。Start Page 也在此处显示。
- **Tasks 窗格** —— 显示编译、调试或运行应用程序产生的任务输出。

图 4-8: MPLAB® X IDE 桌面



如果在文件窗格中双击任何文件名，将会在 Editor 窗格中，在 Start Page 旁边的选项卡下打开相关文件。要关闭选项卡，请单击文件名旁边的“x”。

在文件窗格的 Projects 窗口中右键单击项目名称，可查看弹出（上下文）菜单。对项目子文件夹可执行相同的操作。

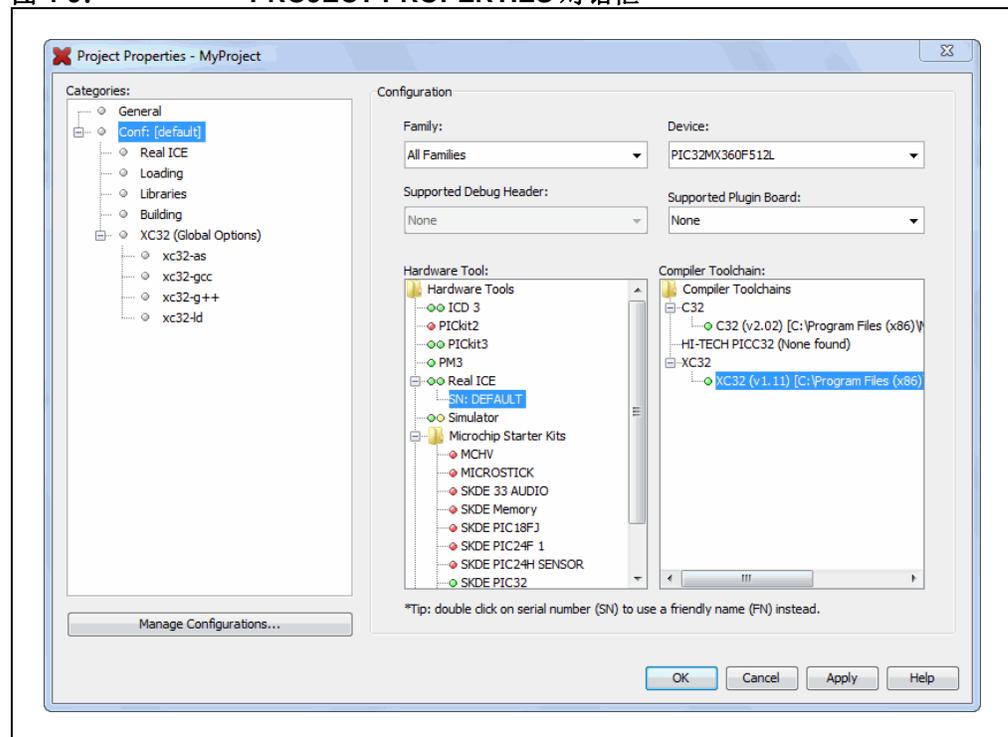
4.4 查看或更改项目属性

创建项目之后，可以在 Project Properties 对话框中查看或更改项目属性。通过以下任一方式访问该对话框：

- 在 Projects 窗口中右键单击项目名称，然后选择“Properties”。
- 在 Projects 窗口中单击项目名称，然后选择 *File>Project Properties*。

单击“Conf: [default]”类别可显示一般项目配置，例如项目器件、相关的调试 / 编程器工具和语言工具。要更改其中的任意项，请参见第 4.2 节“创建新项目”的步骤 4 和 5。

图 4-9: PROJECT PROPERTIES 对话框



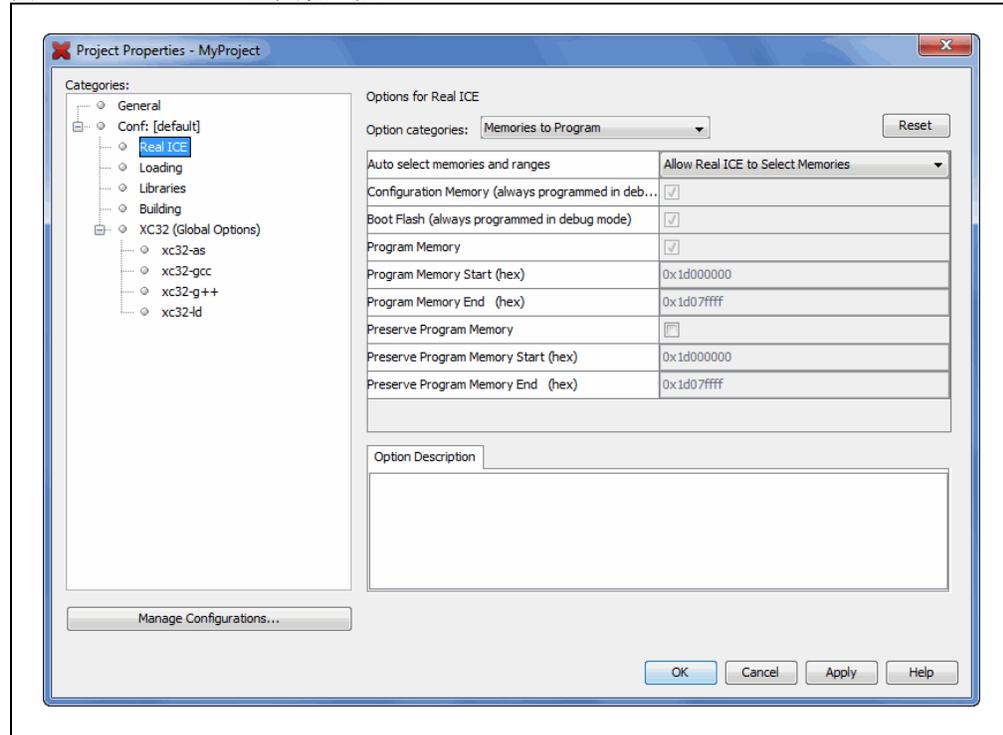
4.5 设置调试器、编程器或语言工具的选项

此外，还可以在 Project Properties 对话框中设置工具的选项。

要设置或更改调试器 / 编程器工具选项：

单击硬件工具或软件模拟器（在 Conf:[default] 下方）可查看相关的设置选项。关于这些选项含义的更多信息，请参见工具文档。

图 4-10: 工具设置页

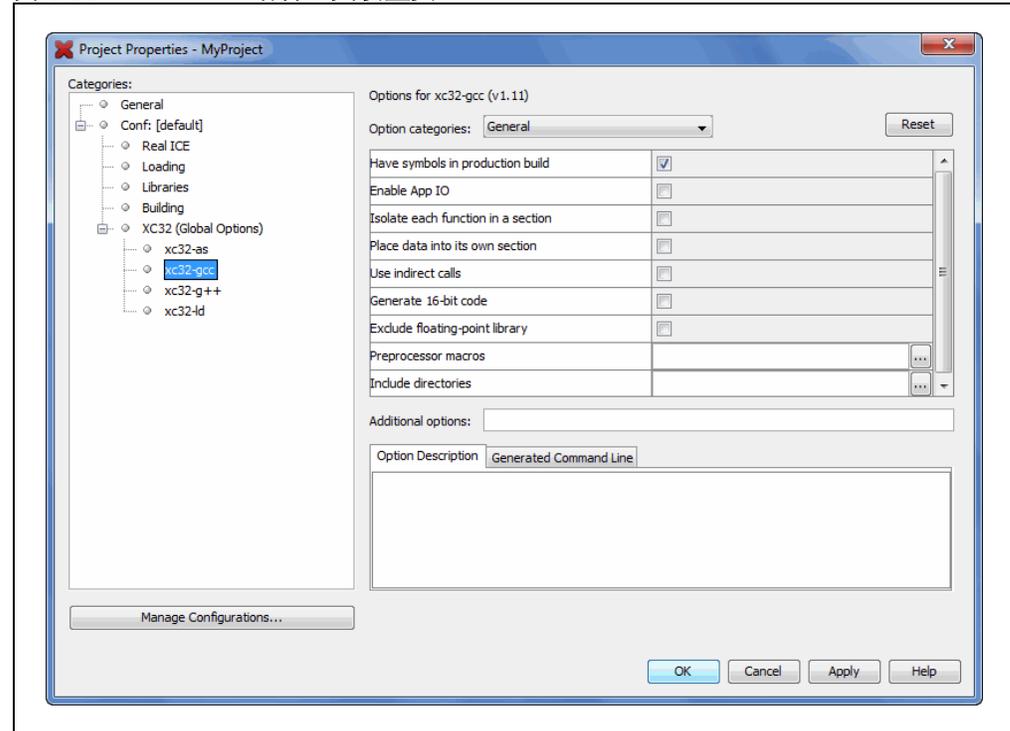


要设置或更改语言工具选项：

单击您的语言工具可查看相关的设置选项。关于这些选项含义的更多信息，请参见语言工具文档。

关于更多帮助，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Working with C/C++/Fortran Projects \(处理 C/C++/Fortran 项目\)](#) > [Setting Project Properties \(设置项目属性\)](#)。

图 4-11: 语言工具设置页



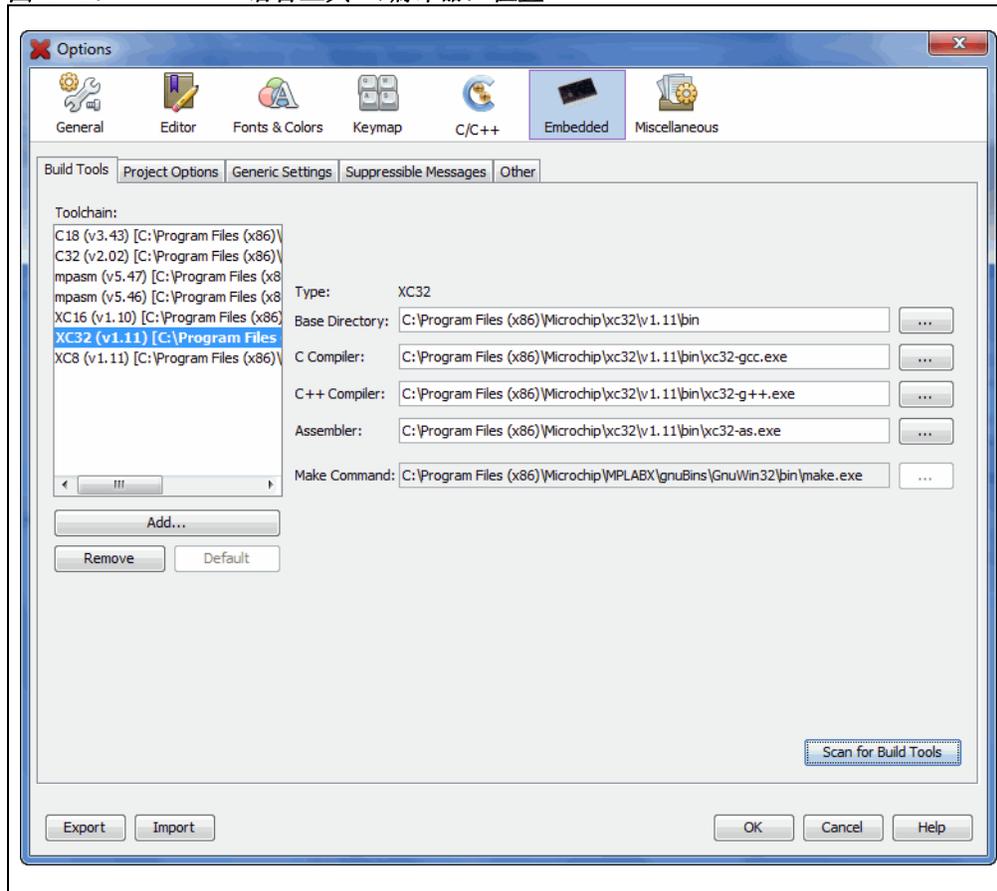
4.6 设置语言工具位置

要查看哪些语言工具可用于 MPLAB X IDE，以及查看或更改它们的路径：

- 对于 **Mac OS X**——通过主菜单条中的 *mplab ide>Preferences>Embedded>Build Tools* 访问编译工具。
- 对于**其他 OS**——通过 *Tools>Options>Embedded>Build Tools* 访问编译工具。

窗口将自动填充所有已安装的工具链。

图 4-12: 语言工具（编译器）位置



4.6.1 添加或更改工具链

如果发现 Toolchain 下未列出您要使用的工具，请尝试以下操作：

- **Scan for Build Tools**——扫描环境路径并列出生计算机上已安装的语言工具。
- **Add**——通过输入包含工具可执行文件的目录的路径（即基目录），将工具手动添加到列表中。通常，它是工具安装目录中的 `bin` 子目录。

如果有多个编译器版本可用，则从列表中进行选择。

要更改工具的路径，请输入新路径或浏览设定它。

4.6.2 关于工具链路径

MPLAB X IDE 会在默认安装路径和 PATH 环境变量下搜索工具链。MPLAB XC16 在 Windows 64 位操作系统下的示例默认路径：

```
C:\Program Files (x86)\Microchip\xc16\bin
```

在安装编译器时，您可以选择：

1. 安装在默认位置或其他位置
2. 将该位置添加到 PATH 中

如果安装在不同于默认位置的其他位置，应将该位置添加到 PATH 中。

如果不选择让安装程序将新位置添加到 PATH 中，可以在 *Tools (工具) > Options (选项)*，**Embedded** 按钮，**Build Tools (编译工具)** 选项卡中通过单击 **Add (添加)** 为 MPLAB X IDE 指明编译器位置。

最后，您可以手动将新位置添加到 PATH 变量中。

4.7 设置其他工具选项

除了编译路径之外，还可以设置其他选项。在 Options 窗口 Embedded 类别中从以下选项卡中进行选择：

- **Project Options (项目选项)** —— 设置项目相关选项，例如 make 选项，以及路径是默认设置为相对还是绝对路径（第 12.12.2 节 “**Project Options 选项卡**”）。
- **Generic Settings (通用设置)** —— 设置日志文件和其他项目功能（第 12.12.3 节 “**Generic Settings 选项卡**”）。
- **Suppressible Messages (可禁止的消息)** —— 选择要禁止的消息（第 12.12.4 节 “**Suppressible Messages 选项卡**”）
- **Other (其他)** —— 编辑 C 源文件和头文件的可接受文件扩展名的列表（第 12.12.6 节 “**Other 选项卡**”）。

4.8 创建新文件

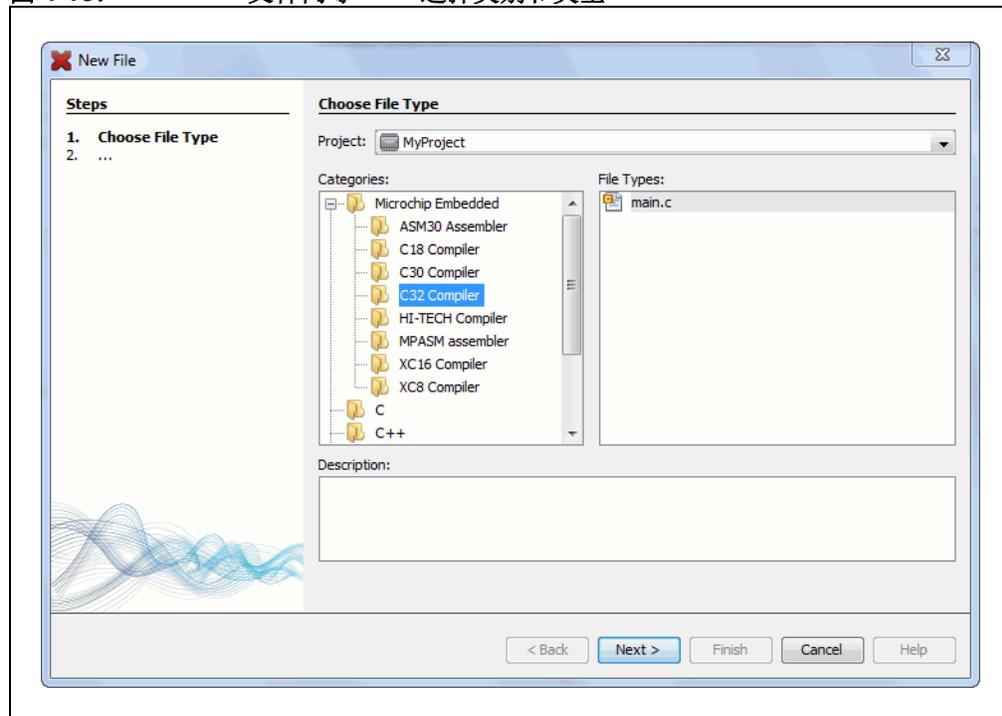
可以通过执行以下操作之一来创建新文件：

- 选择 **File>New File** (*新建文件*) (或 **Ctrl+N**)
- 在项目 / 文件窗口中右键单击项目，并选择 **New** (*新建*) **>Other** (*其他*)
- 在项目 / 文件窗口中右键单击逻辑文件夹 (例如，**Source Files**)，并选择 **New>Other**

New File 向导将启动，指导您完成新文件设置。

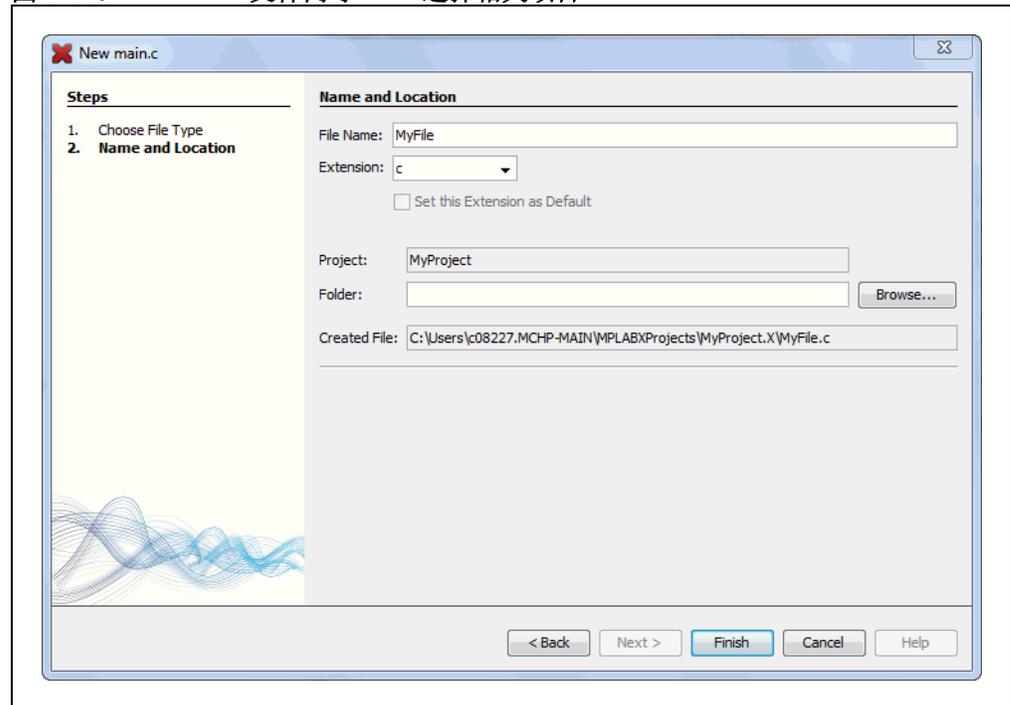
步骤 1. 选择文件类型：通过展开“Microchip Embedded”查找相应选择来选择文件类别。然后选择文件类型。

图 4-13: 文件向导 —— 选择类别和类型



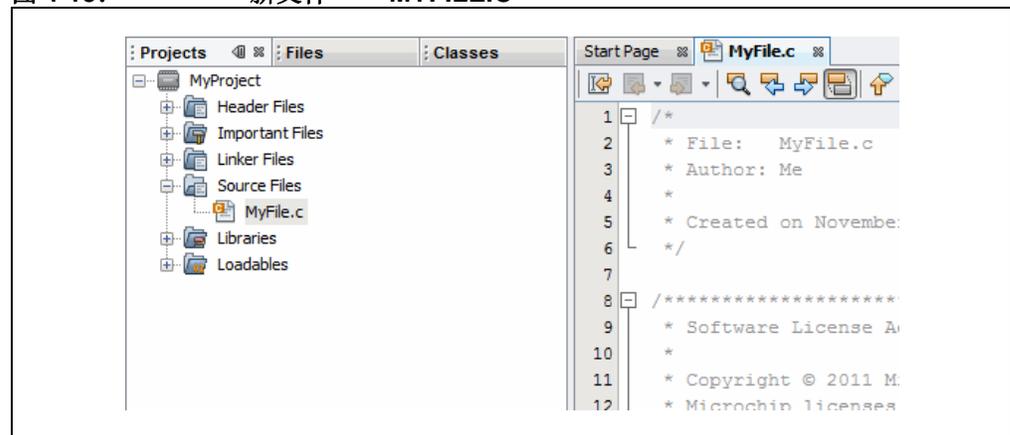
步骤 2. 名称和位置: 命名文件并将其放置在项目文件夹中。

图 4-14: 文件向导 —— 选择相关项目



该文件将显示在文件窗格中的指定项目下，Editor 窗格中将会显示具有该文件名称的选项卡。在该选项卡下输入文件内容。文件中的文本将基于其文件类型进行语法着色。

图 4-15: 新文件 ——MYFILE.C



4.9 向项目中添加现有文件

现有文件可以通过执行以下操作之一添加到项目中：

- 在项目 / 文件窗口中右键单击项目，并选择 “Add Existing Item”
- 在项目 / 文件窗口中右键单击逻辑文件夹（例如，Source Files），并选择 “Add Existing Item”

4.9.1 项目文件夹中的文件

添加文件时，可以选择它的添加方式：

- **Auto**—— 让 MPLAB X IDE 决定定位文件的最佳方式。
- **Relative**—— 指定相对于项目的文件位置（对于大多数可移植项目）。
- **Absolute**—— 通过绝对路径指定文件位置。
- **Copy**—— 将指定文件复制到项目文件夹中。

该文件将显示在文件窗格中的指定项目下，Editor 窗格中将会显示具有该文件名称的选项卡。

4.9.2 项目文件夹以外的文件

添加不处于项目文件夹中的源文件时，请以 “Relative” 方式添加文件。这将创建一个外部文件夹（_ext），从而使项目可以在编译时找到该文件。

要使用导航功能（例如 //TODO 和文件上下文菜单），必须告知项目文件的位置。为此：

1. 在 Projects 窗口中，右键单击项目名称，并选择 “Properties”。
2. 在 “Categories”（类别）下单击 “General”（常规）。
3. 在 “Source Folders”（源文件夹）旁，单击 **Add**。
4. 浏览至已添加到项目中的外部文件的路径。选择 **Select**（选择）。
5. 单击 **Apply** 或 **OK**。然后重新编译项目。

导入 MPLAB IDE v8 项目时，源文件不处于项目文件夹中。使用 *File>Import (导入)>MPLAB IDE v8 Project (MPLAB IDE v8 项目)* 来自动导入文件。

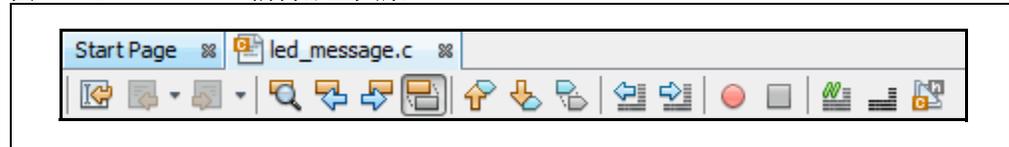
4.10 编辑器用法

要创建新代码或更改现有代码，请使用 NetBeans 编辑器。NetBeans 帮助主题 [IDE Basics \(IDE 基础知识\)](#) > [Basic File Features \(基本文件功能\)](#) 下的目录中提供了关于该编辑器的一般信息。NetBeans 帮助主题 [C/C++/Fortran Development > C/C++/Fortran Project Basics > Navigating and Editing C/C++/Fortran Source Files](#) 提供了关于该编辑器的 C 编译器信息。第 7 章“编辑器”下给出了功能的列表。

要使用编辑器功能，请转至：

- 编辑菜单（见第 11.2.2 节“编辑菜单”）
- 位于每个文件的编辑器窗口顶部的编辑器工具条

图 4-16: 编辑器工具条



- 窗口右键（上下文）菜单。

要设置编辑器属性：

1. 选择 [Tools > Options](#)（对于 Mac OS X 为 [mplab_ide > Preferences](#)）打开 Options 对话框。
2. 单击 **Editor** 按钮。然后单击选项卡来设置编辑器功能。
3. 单击 **Fonts and Colors** 按钮。然后单击选项卡来设置颜色选项。

要进行文件导航和格式化代码：

1. 从 **Navigate**（导航）菜单中选择菜单项，从而在文件内部和外部进行导航。
2. 从 **Source**（源代码）菜单中选择菜单项，从而对代码进行格式化、注释和自动补全。

要使用代码折叠：

关于代码折叠的详细信息，请参见第 7.4 节“代码折叠”。

4.11 添加、设置库和目标文件

您可以引用以下位置中要由链接器使用的库文件：

- Projects 窗口中的 Libraries（库）文件夹
- Project Properties 对话框

在语言工具库管理器中可进行额外的库文件设置。

4.11.1 Libraries 文件夹

在 Projects 窗口中，右键单击 Libraries 文件夹可看到以下选项：

- **Add Library Project（添加库项目）**

创建一个生成您的项目所需的库作为其输出的项目，即建立依赖关系。关于该操作的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development > C/C++/Fortran Project Basics > Building C/C++/Fortran Projects（编译 C/C++/Fortran 项目） > Creating Dependencies Between C/C++/Fortran Projects（在 C/C++/Fortran 项目之间创建依赖关系）](#)。

- **Add Library/Object File（添加库 / 目标文件）**

向项目中添加现有的库文件或预编译（目标）文件。

- **Properties**

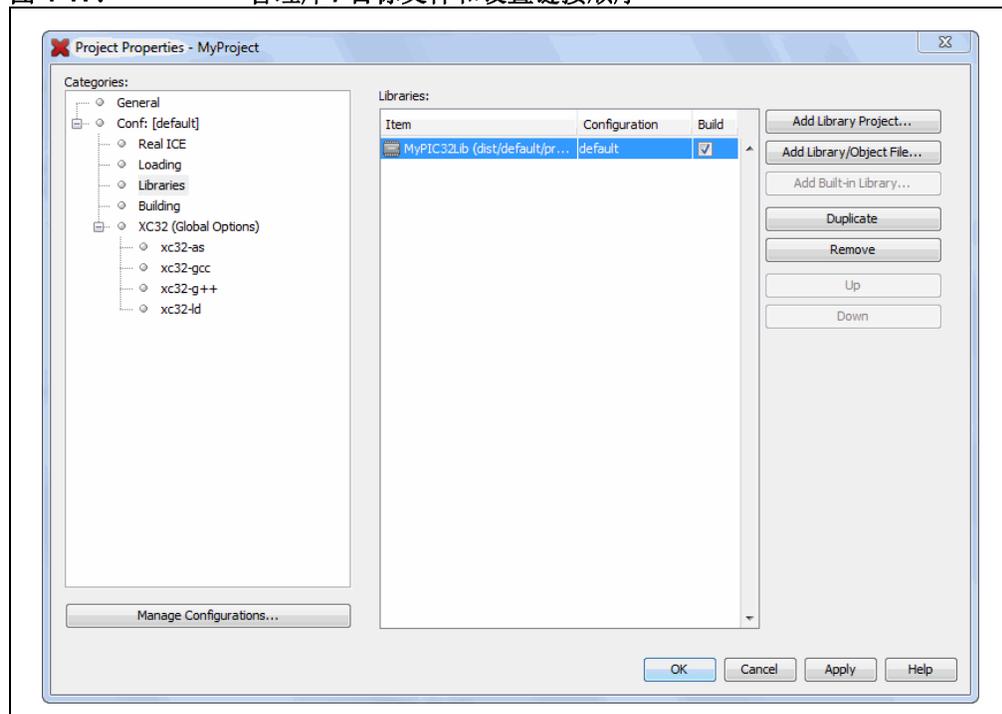
打开项目的 Properties 窗口，以选择更多选项（请参见下一节）。

4.11.2 Project Properties 窗口：Libraries 类别

打开 Project Properties 窗口，然后单击“Libraries”类别。此处将列出在 Projects 窗口中向 Libraries 文件夹添加的所有文件。您可以添加更多文件，并使用右侧的按钮管理这些文件。

此处的文件顺序将决定它们的链接顺序。

图 4-17: 管理库 / 目标文件和设置链接顺序

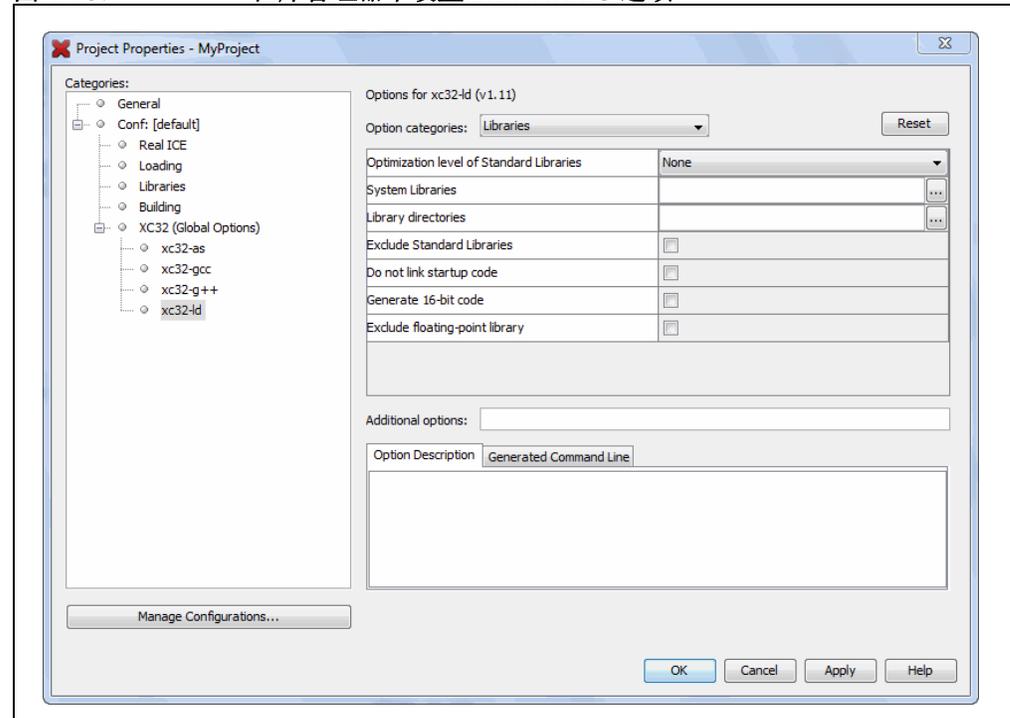


4.11.3 Project Properties 窗口: Librarian (库管理器) 类别

打开 Project Properties 窗口，然后单击语言工具类别下的库管理器。要确定库管理器的可执行文件名，请参见语言工具文档。

在左窗格中，单击链接器的名称；然后在右窗格中，选择“Libraries”选项。从该列表中选择库选项。

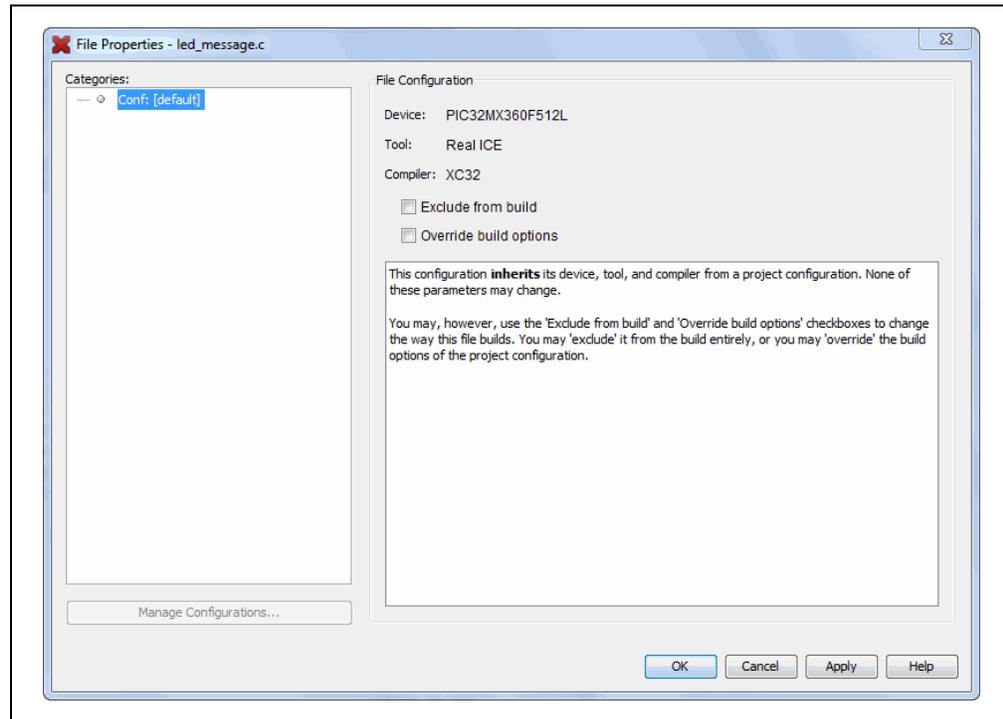
图 4-18: 在库管理器中设置 LIBRARIES 选项



4.12 设置文件和文件夹属性

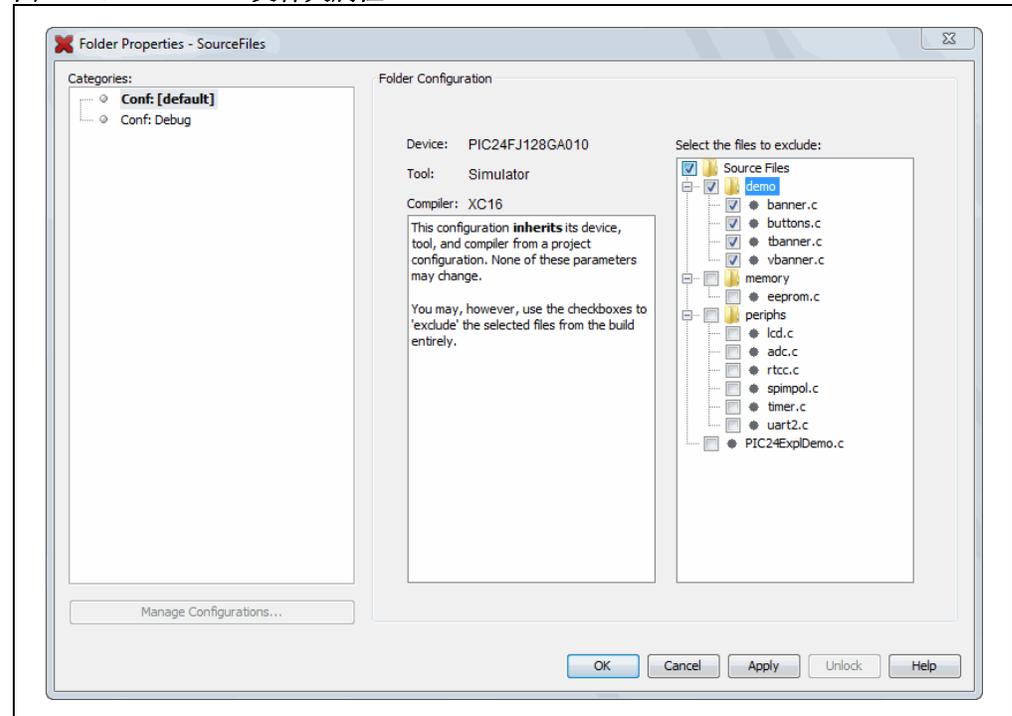
通过使用 **File Properties**（文件属性）对话框，一个项目文件可以使用与项目中其他文件不同的方式进行编译。通过在 **Projects** 窗口中右键单击文件名，然后选择 **“Properties”** 来访问该对话框。选择从项目编译中排除该文件，或使用在此处选择的选项覆盖项目编译选项。要覆盖编译选项，请选中复选框，然后单击 **“Apply”** 以查看 **“Categories”** 下出现的选择选项。

图 4-19: FILE PROPERTIES



通过在 **Projects** 窗口中右键单击文件夹名称，然后选择“**Properties**”（属性），可以从编译过程中排除整个（虚拟）文件夹。通过选择从编译中排除文件夹或各个文件。选择处于其他文件夹之上的某个文件夹时，将选择该文件夹中的全部内容。然后，您可以根据需要取消选择文件或其他文件夹。

图 4-20: 文件夹属性

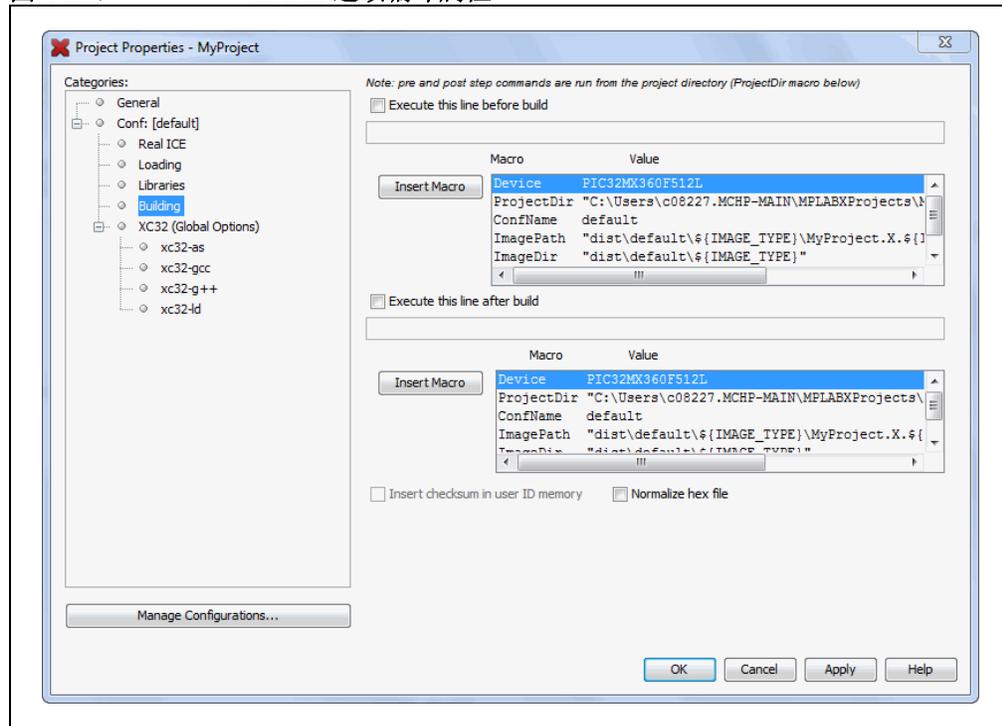


4.13 设置编译属性

在编译项目之前，您可能希望设置额外的编译属性：

- Execute this line before/after the build （在编译之前 / 之后执行该行）
- Insert checksum in user ID memory （在用户 ID 存储器中插入校验和）
- Normalize hex file （标准化十六进制文件）

图 4-21: MAKE 选项编译属性



4.13.1 Execute this line before/after the build (在编译之前 / 之后执行该行)

输入要在编译过程最初或最后执行的命令。这些命令会被插入 nbproject/Makefile- $\$$ CONF.mk 文件，使您可以定制编译过程。如果需要在要调用的脚本或程序中引用一些与项目相关的项（如映像名称），请使用所提供的宏。

您可以自己输入宏，也可以单击 **Insert Macro**（插入宏）将宏名称复制到编辑框中的当前位置。命令将在 **make** 过程中运行，而当前目录设置为 MPLAB X IDE 项目目录。项目目录定义为包含 nbproject 文件夹的项目。

表 4-3: 宏

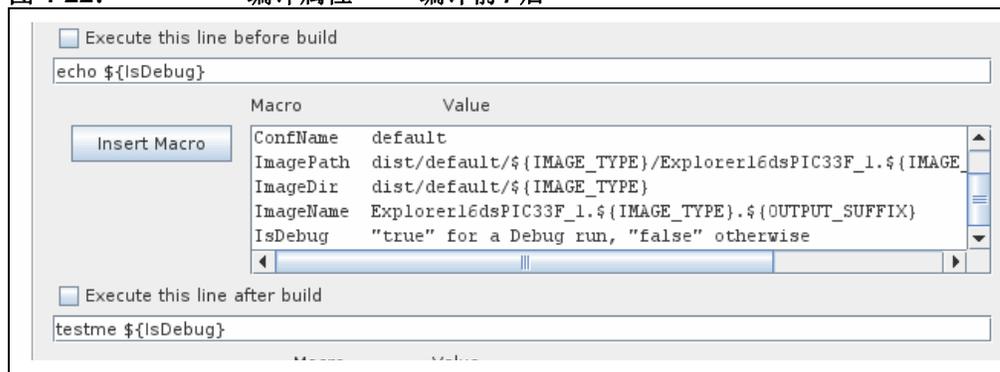
名称 *	功能
Device	当前选定项目配置的器件。
IsDebug	对于调试运行行为“true”；否则为“false”。
ProjectDir	项目文件在 PC 上的位置。
ConfName	当前选定项目配置的名称。
ImagePath	编译映像的路径。
ImageDir	包含编译映像的目录。
ImageName	编译映像的名称。

* 可能会有其他宏可用，具体取决于所用的编译器。

例 4-1: 仅在调试期间执行某个过程

在下面所示的窗口中，单击 IsDebug 宏来选择它，然后单击 **Insert Macro** 将其插入脚本。

图 4-22: 编译属性 —— 编译前 / 后



检查运行的脚本中的 $\$1$ （Linux 或 Mac 操作系统）或 $\%1$ （Windows 操作系统）的值。

Bash 代码示例

```
#!/bin/bash
echo is $1
if [ "$1" == "true" ]; then
    echo We are in debug
else
    echo We are in production
fi
```

批处理代码示例

```
@echo off
if "%1" == "true" goto debug
:production
@echo We are in production
goto end
:debug
@echo We are in debug
:end
```

4.13.2 Insert checksum in user ID memory（在用户 ID 存储器中插入校验和）

如果器件支持，则选中时将使用编译后生成的校验和数字作为用户 ID。

4.13.3 Normalize hex file（标准化十六进制文件）

当行地址按递增顺序放置，且数据尽可能缓冲为一种大小（16 字节）时，即表示十六进制文件进行了标准化处理。

例如，以下行（记录）来自文件 `myfile.hex`，该文件是 MPLAB XC16 C 编译器 / 链接器的输出：

```
:0801f800361e0000361e000057
:020000040000fa
:10020800361e0000361e0000361e0000361e000096
```

当该文件进行标准化之后，文件数据的形式如下：

```
:10022800361E0000361E0000361E0000361E000076
:10023800361E0000361E0000361E0000361E000066
:10024800361E0000361E0000361E0000361E000056
```

其格式为：

```
:bbaaaarrdd...ddcc
```

其中：

:	起始记录
bb	字节计数
aaaa	地址
rr	记录类型
dd	数据
cc	校验和

在 MPLAB X IDE 中，使用应用程序 HEXMATE 来对 Intel 十六进制文件进行标准化处理。选中“Normalize hex file”选项时，在编译后会调用以下命令：

```
hexmate myfile.hex -omyfile.hex
```

实际上，HEXMATE 会将整个十六进制文件解包，并将数据放置在由十六进制文件指定的地址处。然后，它会将产生的存储器映像重新打包为一个新的十六进制文件。在产生的文件中，所有数据都按升序放置，并且是连续的。如果地址中存在空隙，则输出文件中也会存在空隙（不会填充未使用的地址）。

关于使用 HEXMATE 的更多信息，请参见所安装的 MPLAB XC8 C 编译器的 docs 文件夹中的 `manual.pdf`。

标准化的十六进制文件对于通过串行连接进行代码编程（如自举程序）非常有用，因为这最大程度减小了记录字节的差异。

4.14 编译项目

对于 MPLAB X IDE，不需要先编译项目再运行或调试。编译属于运行和调试过程的一部分。但是，对于初始开发或重要更改，可能需要确保在尝试运行或调试之前先进行项目编译。

要编译项目：

- 在 **Projects** 窗口中，右键单击项目名称，并选择 “**Build**”。您还可以选择 “**Clean and Build**” 在编译之前删除中间文件。
- 单击 “**Build Project**” 或 “**Clean and Build Project**” 工具条图标。

Output 窗口中将会显示编译进度。

可用的编译功能包括：

表 4-4: 工具条按钮上的编译选项

按钮图标	功能	说明
	编译项目	make 项目中的所有文件。
	针对调试而编译	make 项目中的所有文件，并在编译映像中添加调试执行程序。
	清除并编译	删除先前的编译文件，并 make 项目中的所有文件。
	针对调试进行清除并编译	删除先前的编译文件，并 make 项目中的所有文件。向编译映像添加调试执行程序。

要在 **Output** 窗口中查看错误：

1. 在 **Output** 窗口中单击右键并选择 “**Filter**”（过滤器）。
2. 在 **Filter** 对话框中，选中 “**Match（匹配）**” 并输入 “**: error**”，从而在 **Output** 窗口中仅显示导致停止编译的错误。
3. 使用 **<Ctrl>-<G>** 可开启和关闭过滤器。

关于错误的讨论，请参见语言工具文档。

要查看校验和信息：

- 打开 **Dashboard** 窗口（见第 5.17 节 “**查看仪表板显示**”）可查看编译后的校验和。

关于编译项目的更多信息，另请参见 NetBeans 帮助主题 [C/C++/Fortran Development>C/C++/Fortran Project Basics>Building C/C++/Fortran Projects](#)。

4.15 运行代码

成功编译代码之后，可以尝试运行应用程序。

4.15.1 运行如何工作

单击 **Run Project**（运行项目）工具条按钮时：

1. 如果 **make** 过程确定需要进行 **build**，则进行 **build**。
2. 对于在线调试器 / 仿真器和编程器，将自动使用映像（无需调试执行程序）对目标器件进行编程，然后将释放器件使之运行，即，将不会使能任何断点或其他调试功能。要在编程后将器件保持在复位状态，请单击“**Hold in Reset**”（保持复位）而不是“**Run Project**”。



Hold in Reset

3. 对于软件模拟器，应用程序将只是执行，没有调试功能。

Output 窗口中将会显示运行进度。

4.15.2 运行注意事项

运行程序时，需要注意：

1. MPLAB X IDE 操作会在运行时连接到硬件工具（Run 或 Debug Run）。这意味着在 MPLAB X IDE 中进行的设置只会在运行时被传递给该工具。只有运行时再次启动时，才会更新在暂停期间更改的设置。

要总是连接到硬件工具（如同 MPLAB IDE v8），请参见 [Tools>Options](#)（对于 Mac OS X 为 [mplab_ide>Preferences](#)），**Embedded** 按钮，**Generic Settings** 选项卡，“Maintain active connection to hardware tool”（保持硬件工具的连接处于活动状态）复选框。

2. 请参见第 4.23 节“对器件编程”。

4.15.3 运行应用程序代码

1. 在 **Projects** 窗口中，选择项目或将其设为主项目（右键单击项目并选择“**Set as main project**”）。
2. 单击“**Make and Program Device**”图标（或选择 [Run>Run Project](#)）来运行程序。



Make and Program Device

Output 窗口中将会显示运行进度。

4.16 调试运行代码

如果代码未成功运行，则需要调试它。在调试模式下运行代码称为 Debug Run（调试运行）。

4.16.1 调试运行如何工作

单击“Debug Project”工具条按钮时：

1. 如果 make 过程确定需要进行 build，则进行 build。
2. 对于在线调试器 / 仿真器，将自动使用映像（包括调试执行程序）对目标器件或仿真头进行编程并将启动调试会话。
3. 对于软件模拟器，将会启动调试会话。

Output 窗口中将会显示调试运行进度。

4.16.2 生成的调试宏

MPLAB X IDE 会生成一些调试宏，供 Microchip 语言工具使用。表 4-5 列出了传递给 Microchip 编译器和汇编器的宏。

表 4-5: MICROCHIP 工具调试宏

宏名称	关联工具	功能
__DEBUG	所有	指定这是调试编译。
__MPLAB_REAL_ICE__ __MPLAB_ICD3__ __MPLAB_PK3__ __MPLAB_PICKIT2__	XC8	指定使用的硬件调试工具。 格式为 __MPLAB_xxx__，其中的 xxx 代表硬件工具说明符。
__MPLAB_DEBUGGER_REAL_ICE__ __MPLAB_DEBUGGER_ICD3__ __MPLAB_DEBUGGER_PK3__ __MPLAB_DEBUGGER_PICKIT2__	XC16、 XC32 和 MPASM	指定使用的硬件调试工具。 格式为 __MPLAB_DEBUGGER_xxx__，其中的 xxx 代表硬件工具说明符。
__MPLAB_DEBUGGER_PIC32MXSK	XC32	指定使用的入门工具包。

您可以在自己的代码中使用这些宏。例如：

```
#ifdef __DEBUG
    fprintf(stderr, "This is a debugging message\n");
#endif
```

4.16.3 调试注意事项

调试代码时，需要注意：

1. 只有处于调试会话（调试模式）下时，才能激活许多调试功能。例如，在观察或存储器窗口中查看变量的值。
2. MPLAB X IDE 操作会在运行时连接到硬件工具（Run 或 Debug Run）。这意味着在 MPLAB X IDE 中进行的设置只会在运行时被传递给该工具。只有运行时再次启动时，才会更新在暂停期间更改的设置。

要总是连接到硬件工具（如同 MPLAB IDE v8），请参见 [Tools>Options](#)（对于 Mac OS X 为 [mplab_ide>Preferences](#)），**Embedded** 按钮，**Generic Settings** 选项卡，“Maintain active connection to hardware tool”复选框。

3. 对于某些应用，可能需要分解一些调试步骤，独立执行调试步骤。为此，请使用 [Debug>Discrete Debugger Operation](#)（[分离式调试器操作](#)）下的步骤。

4.16.4 调试应用程序代码

要调试应用程序代码：

1. 在 **Projects** 窗口中，选择项目或将其设为主项目（右键单击项目并选择“**Set as main**”）。
2. 单击“**Debug Project**”图标（或选择 *Debug>Debug Project* 或 *Debug>Step Into*）开始调试运行。



Debug Project

要暂停应用程序代码：

- 单击“**Pause**”图标（或选择 *Debug>Pause*）暂停程序执行。

要再次运行代码：

- 单击“**Continue**”图标（或选择 *Debug>Continue*）再次启动程序执行。

要结束代码的执行：

- 单击“**Finish Debugger Session**”图标（或选择 *Debug>Finish Debugger Session*）结束程序执行。

在使用调试功能时（从第 4.17 节“使用断点控制程序执行”开始），Run 和 Debug Run 之间的差异将变得很明显。

要启动调试器：

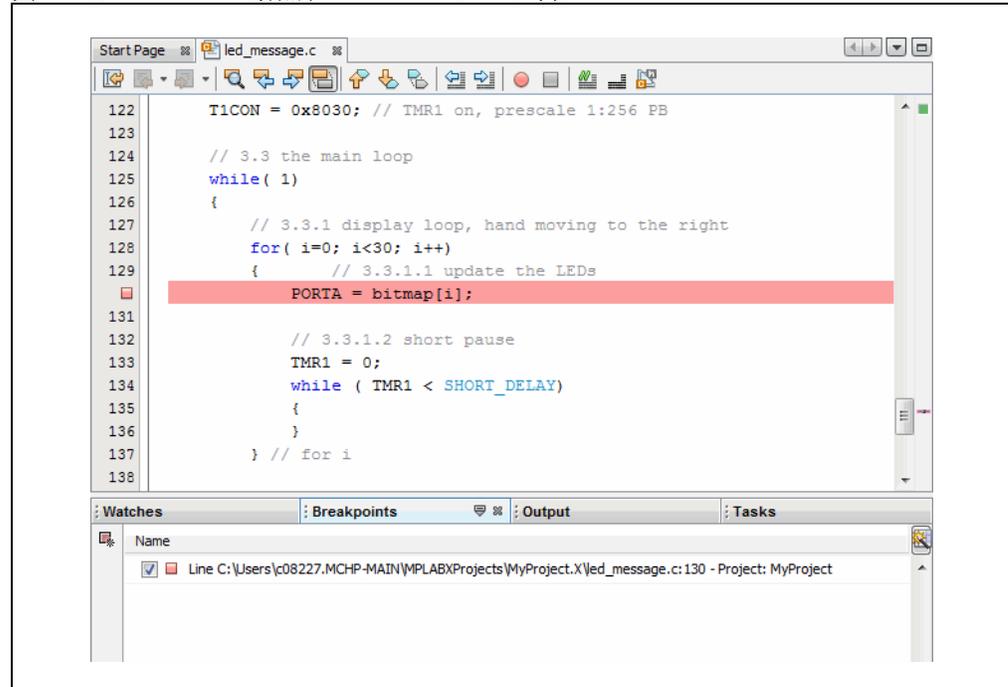
如果代码是针对调试而编译的，并且只是希望启动调试工具，则可以通过选择“**Debug Project**”图标旁边的向下箭头并选择“**Launch Debugger**”（启动调试器）来实现。

4.17 使用断点控制程序执行

在调试代码时，通过在代码中特定位置暂停执行来检查变量值的功能会很有用。要执行该操作，请使用断点。

关于断点的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development > Debugging C/C++/Fortran Applications with gdb > Setting C/C++/Fortran Breakpoints](#)。

图 4-23: 断点和 BREAKPOINTS 窗口



4.17.1 设置或清除简单断点

要在某个代码行上设置断点，请执行以下操作之一：

- 在源代码编辑器中单击代码行的左边缘
- 按下 **Ctrl+F8**

要清除断点，请执行以下操作之一：

- 重复设置断点的步骤
- 选择 **Debug > Toggle Breakpoint**。

4.17.2 使用 Breakpoint 对话框设置断点

要使用 New Breakpoint（新建断点）对话框设置断点：

1. 选择 *Debug>New Breakpoint*。
2. 在 New Breakpoint 对话框中选择断点类型和其他选项。

将在第 12.4.1 节“New Breakpoint 对话框”中详细介绍每个断点类型所具有的选项。

4.17.3 在 Breakpoints 窗口中设置 / 清除断点

要在 Breakpoints 窗口中查看和翻转断点：

1. 选择 *Window>Debugging>Breakpoint (断点)*。
2. 通过选中 / 取消选中复选框来翻转断点。

要使用 New Breakpoint 对话框设置断点：

1. 单击窗口左上角的图标。

4.17.4 设置断点序列（依赖于器件）

断点序列是断点的列表，它们会一直执行直到执行最后一个断点之后才会暂停执行。当存在到特定指令的多个执行路径，并且只希望使用一个特定路径时，序列化断点会很有用。

要创建断点序列：

1. 右键单击某个现有断点或按 Shift 键并单击选择一组现有断点，然后右键单击该组。
2. 从弹出式菜单中，转至“Complex Breakpoint”（复杂断点）并选择“Add a New Sequence”（添加新序列）。
3. 在对话框中输入序列的名称并单击 **OK**。
4. 断点会出现在新序列下。
5. 要向序列中添加更多现有断点，请右键单击断点并选择 *Complex Breakpoint>Add to Name (添加至 Name)*，其中的 *Name* 是序列的名称。
6. 要向序列中添加新断点，请右键单击序列并选择“New Breakpoint”。

要选择序列顺序：

1. 展开序列来查看所有项。
2. 右键单击某项，然后选择 *Complex Breakpoints>Move Up (上移)* 或 *Complex Breakpoints>Move Down (下移)*。断点的执行顺序为从下到上；序列中的最后一个断点最先执行。

要删除序列或断点：

1. 右键单击相应的项并选择“Disable（禁止）”，可临时删除该项。
2. 右键单击相应的项并选择“Delete（删除）”，可永久删除该项。

4.17.5 设置断点元组（依赖于器件）

对于 MPLAB X IDE，一个元组代表断点的逻辑与列表。如果某个变量会在多个位置发生修改，而您只需在变量于一个特定位置发生修改时产生中断，逻辑与断点会很有用。只能对两个断点进行逻辑与运算，并且它们必须包括一个程序存储器断点和一个数据存储器断点。断点 1 和断点 2 必须同时发生，程序才会暂停。

要创建断点元组：

1. 单击 **Breakpoints** 窗口左上角的图标来打开 **New Breakpoint** 对话框。
2. 创建地址断点。单击 **OK** 将其添加到 **Breakpoints** 窗口中。
3. 重复步骤 1 和步骤 2 来创建一个数据断点。
4. 右键单击一个断点，然后选择 **Complex Breakpoint>Add to New Tuple**（添加至新元组）。
5. 在对话框中输入元组名称并单击 **OK**。
6. 断点会出现在新元组下。
7. 在另一个断点上单击右键，并选择 **Complex Breakpoint>Move to Name**（移至 Name），其中的 **Name** 是元组的名称。

要删除元组或断点：

1. 右键单击相应的项并选择“Disable”，可临时删除该项。
2. 右键单击相应的项并选择“Delete”，可永久删除该项。

4.17.6 断点应用

要确定断点之间的时间：

- 使用跑表（见第 5.14 节“使用跑表”）。

要确定断点资源：

- 打开 **Dashboard** 窗口（见第 5.17 节“查看仪表板显示”）可查看可用断点和已用断点的数量，以及是否支持软件断点。

4.17.7 断点使用注意事项

入门工具包、在线调试器（包括 PICKit 2 和 3）和 MPLAB REAL ICE 在线仿真器支持有限数量的断点。可用断点的数量取决于选定的器件。要查看可用断点的数量和跟踪已用的数量，请参见 **Dashboard** 窗口（第 5.17 节“查看仪表板显示”）。

以下 MPLAB X IDE 功能使用断点来完成其功能：

- Step Over
- Step Out
- Run to Cursor
- Reset to Main（复位至主程序）

如果在没有断点可用时尝试使用这些功能之一，将会显示一个对话框，指示所有资源都已被使用。

4.18 单步执行代码

使用 **Debug** 菜单和 **Debug** 工具条上的单步功能之一，可从代码起始处或断点处暂停之后逐步执行代码。检查变量值的变化（见下一节）或确定程序流是否正确。

有几种方法可以单步执行代码：

- **Step Over**—— 执行程序的一行源代码。如果该行是一个函数调用，则执行整个函数，然后停止。
- **Step Into**—— 执行程序的一行源代码。如果该行是一个函数调用，则程序执行到该函数的第一条语句，然后停止。
- **Step Out**—— 执行程序的一行源代码。如果该行是一个函数调用，则执行函数，并将控制返回给调用方。
- **Run to Cursor**—— 运行当前项目，直到文件中的光标位置，并停止程序执行。

除了 **Editor** 窗口之外，还可以在 **Disassembly**（反汇编）窗口（**第 5.15 节 “查看 Disassembly 窗口”**）和存储器窗口的程序存储器中单步执行代码。

关于单步的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development > Debugging C/C++/Fortran Applications with gdb > C/C++/Fortran Debugging Sessions > Stepping Through Your C/C++/Fortran Program](#)。

4.19 观察符号值变化

在 **Watches**（观察）窗口中观察选择更改的符号的值。在程序执行期间确定这些值是否为预期值可以帮助您调试代码。

可以添加到 **Watches** 窗口中的符号包括：

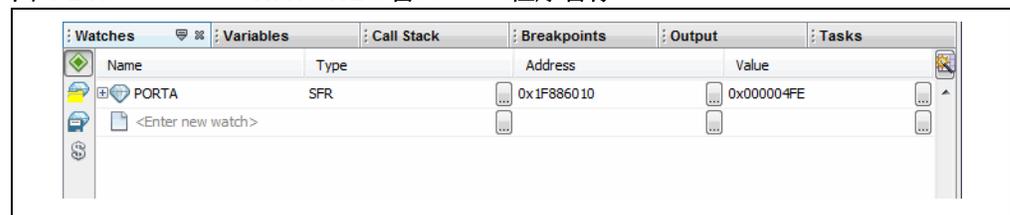
- 全局符号 —— 在编译后可见
- **SFR** —— 特殊功能寄存器（依赖于器件）
- 绝对寻址

一般情况下，只有在从调试运行中暂停时，才能看到更新后的值。但是，一些工具支持运行时更新（您可以在程序运行时看到值的变化）。要确定您的工具是否支持该功能，请参阅工具文档。

对于除 **PIC32 MCU** 之外的所有器件，在运行时观察中使用的符号必须调整长度，使之与器件存储器匹配。也即，使用 **8 位** 器件时需要 **8 位** 符号。

对于 **C** 语言枚举类型，可以在窗口中输入枚举标号（文本）或整数值。对于标号，请注意它们是区分大小写的。

图 4-24: WATCHES 窗口 —— 程序暂停



要查看 **Watches** 窗口，请执行以下操作之一：

- 选择 **Window>Debugging>Watches** 打开窗口。
- 如果 **Output** 窗口已打开，则在窗口中单击 **Watches** 选项卡。

要直接创建新的观察：

可以通过以下操作之一，向 **Watches** 窗口直接添加符号：

- 双击名称列，并输入一个全局符号、**SFR** 或绝对地址（**0x300**）。
- 在 **Editor** 窗口中右键单击某个全局符号或 **SFR**，并选择“**New Watch**”。
- 在 **Editor** 窗口选择全局符号或 **SFR**，并将其拖放到 **Watches** 窗口中。

使用 **New Watch** 对话框创建新的观察：

通过使用 **New Watch** 对话框，可以向 **Watches** 窗口中添加符号或 **SFR**：

- 在 **Watches** 窗口中单击右键并选择“**New Watch**”或选择 **Tools>New Watch**。单击选择按钮可查看 **Global Symbols**（全局符号）或 **SFR**。在列表中单击某个名称，然后单击 **OK**。
- 在 **Editor** 窗口中选择符号或 **SFR** 名称，然后从右键菜单中选择“**New Watch**”。该名称将填充到窗口中。单击 **OK**（确定）。

要创建新的运行时观察：

向 **Watches** 窗口中添加运行时观察之前，需要先设置时钟：

1. 右键单击项目名称，并选择 “**Properties**”。
2. 单击调试工具名称（例如，**Real ICE**），并选择选项类别 “**Clock**”（时钟）。
3. 设置运行时指令速度。

要添加全局符号或 **SFR** 作为运行时观察，请按照 “使用 **New Watch** 对话框创建新的观察” 下的说明操作，只是此时选择 “**New Runtime Watch**”（新建运行时时钟）而不是 “**New Watch**”。

对于除 **PIC32 MCU** 外的所有器件，在运行时观察中使用的符号必须调整长度，使之与器件存储器匹配。也即，使用 8 位器件时需要 8 位符号。

要查看符号变化：

1. 执行调试运行，然后暂停程序。
2. 单击 **Watches** 选项卡，使窗口处于活动状态。
3. 对于观察符号，继续调试运行并暂停可查看值的变化。对于运行时观察符号，继续调试运行并在程序执行时观察值的变化。

只有处于调试会话中，才能看到符号（全局符号、**SFR**、数组和寄存器位域等）的值。

要更改观察符号的基数：

- 右键单击该符号所在的行，并选择 “**Display Value As**”（值的显示方式）。

要执行其他任务：

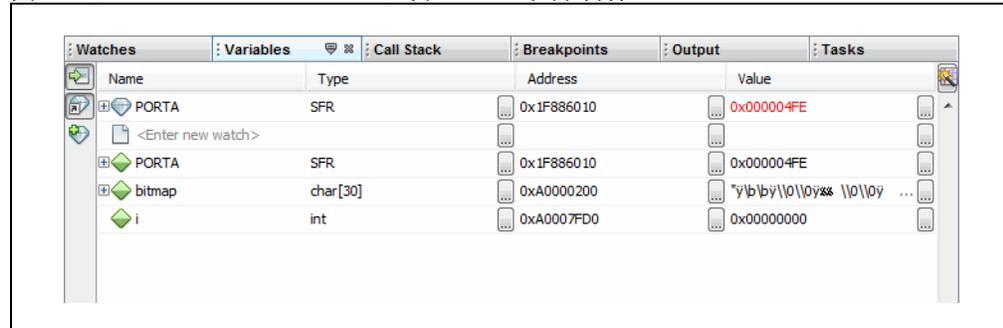
关于观察的更多信息，请参见第 12.14 节 “**Watches 窗口**”。

4.20 观察局部变量值的变化

在 **Variables**（变量）窗口中观察局部变量值的变化。在程序执行期间确定这些值是否为预期值可以帮助您调试代码。

一般情况下，只有在从调试运行中暂停时，才能看到更新后的值。但是，一些工具允许运行时更新。要确定您的工具是否支持该功能，请参阅工具文档。

图 4-25: **VARIABLES** 窗口 —— 程序暂停



要查看 **Variables** 窗口，请执行以下操作之一：

- 选择 **Window>Debugging>Variables** 打开窗口。
- 如果 **Output** 窗口已打开，则在窗口中单击 **Variables** 选项卡。

要查看变量变化：

1. 执行调试运行，然后暂停程序。
2. 单击 **Variables** 选项卡可查看窗口和局部变量值。

要更改变量的基数：

- 右键单击该变量所在的行，并选择 “**Display Value As**”。

要执行其他任务：

关于变量的更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb>Viewing C/C++/Fortran Program Information>C and C++ Variables and Expressions in the IDE](#) (*IDE 中的 C 和 C++ 变量和表达式*)。

4.21 查看 / 更改器件存储器（包括配置位）

MPLAB X IDE 具有灵活的、抽象化的存储器窗口，它们可在调试期间提供不同类型器件存储器的定制视图。只有在从调试运行中暂停时，才能在该窗口中看到更新后的值。

4.21.1 查看器件存储器

1. 在窗格中单击某个窗口，使窗格处于活动状态。存储器窗口将在该窗格中打开。
2. 从 *Window>PIC Memory Views* 中选择一个存储器视图。下表中说明了可用的选项：

表 4-6: 存储器视图——8 位和 16 位器件

类型	说明
Program Memory（程序存储器）	器件上的所有程序存储器（ROM）
File Registers（文件寄存器）	器件上的所有文件寄存器（RAM）存储器
SFR	所有特殊功能寄存器（SFR）
Peripherals（外设）	与外设有关的所有 SFR
Configuration Bits（配置位）	所有配置寄存器
EE Data Memory（EE 数据存储器）	器件上的所有 EE 数据存储器
User ID Memory（用户 ID 存储器）	用户 ID 存储器

表 4-7: 存储器视图——32 位器件

类型	说明
Execution Memory（执行存储器）	器件上的所有闪存存储器
Data Memory（数据存储器）	器件上的所有 RAM
Peripherals	所有特殊功能寄存器（SFR）
Configuration Bits	所有配置寄存器
CPU Memory（CPU 存储器）	所有 CPU 存储器
User ID Memory	用户 ID 存储器

3. 存储器窗口打开之后，可以通过在下拉框中选择存储器类型和存储器格式来进一步修改视图。

图 4-26: 存储器和存储器格式选择



表 4-8: 存储器窗口选项——8 位和 16 位器件

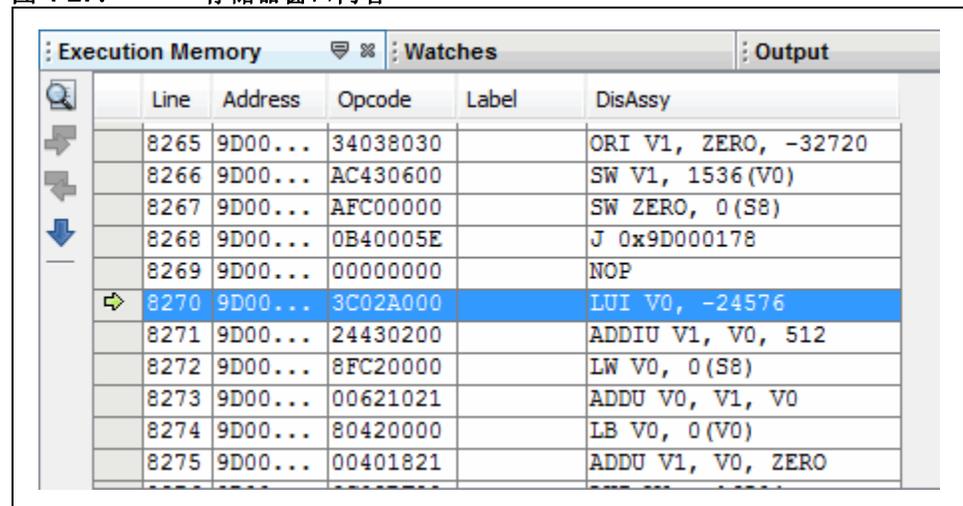
选项	值	说明
Memory	File Registers	器件上的所有文件寄存器
	Program	器件上的所有程序存储器
	SFR	所有特殊功能寄存器（SFR）
	Configuration Bits	所有配置寄存器
	EEPROM	所有 EEPROM 存储器
Format（格式）	Data（数据）	数据存储器（RAM）
	Code（代码）	程序存储器（ROM）

表 4-9: 存储器窗口选项——32 位器件

选项	值	说明
Memory	RAM Memory (RAM 存储器) Flash Memory (闪存存储器) Peripheral Configuration Bits CPU Memory Memory User ID	器件上的所有 RAM 器件上的所有闪存存储器 所有特殊功能寄存器 (SFR) 所有配置寄存器 所有 CPU 存储器 所有存储器 用户 ID 存储器
Format	Data Code	数据存储器 (RAM) 程序存储器 (ROM)

4. 在调试运行并暂停之后，窗口中将填入所选的存储器。
5. 通过单击该窗口选项卡上的“x”关闭该窗口。

图 4-27: 存储器窗口内容



4.21.2 更改器件存储器

要更改存储器值，您必须调试运行您的代码。在运行期间，您无法更改存储器。

注： 数据将仅在调试运行过程中发生更改。应用程序代码不会发生更改。

要更改存储器值：

- 在存储器窗口中，通过在相应列中单击，并选择或输入新数据来更改值。对于一些窗口，文本将使用红色来说明发生了更改。
- 大多数存储器窗口的上下文（右键）菜单上都具有 **Fill memory**（填充存储器）功能。
- 对于程序存储器，必须重新编译才能看到更改。使用 **Debug>Discrete Debugger Operation** 可使用更改后的数据对目标编程并启动调试器。

4.21.3 通过上下文菜单设置存储器窗口选项

在存储器窗口中单击右键将弹出一个上下文菜单，其中具有各种选项，例如显示选项、填充存储器、表导入 / 导出和输出到文件。关于该菜单的更多信息，请参见第 12.8.13 节“存储器窗口菜单”。

4.21.4 设置配置位

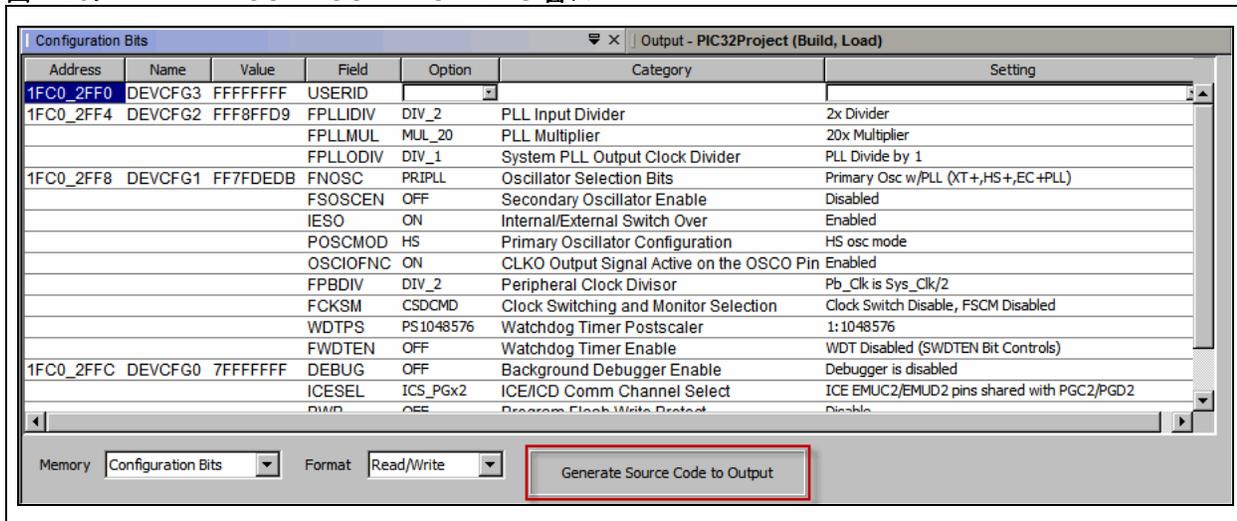
您必须在代码中设置配置位。为了帮助您开发配置位设置，您可以使用 Configuration Bits 窗口（见第 4.21.1 节“查看器件存储器”）。

您可以在调试会话过程中在 Configuration Bits 窗口中临时更改配置位（见第 4.21.2 节“更改器件存储器”）。完成所需的设置之后，请选择“Generate Source Code to Output”。然后，可以将该代码从 Output 窗口中复制到代码中。

如果不是处于调试会话中，则无法编辑配置位。此外，如果进行重新编译，Configuration Bits 窗口中的所有更改都会丢失。

关于不同器件的配置位设置的总结，请参见附录 A “配置设置汇总”。

图 4-28: CONFIGURATION BITS 窗口



4.21.5 刷新选定的存储器窗口

对于显示程序存储器、EEPROM、用户 ID 或配置位存储器的存储器窗口，可以通过执行以下操作刷新视图：

1. 如果正在进行调试，则除非工具和器件支持调试读取否则需要暂停您的程序（见下文）。
2. 单击名为“Read Device Memory”的图标。



Read Device Memory 图标

调试读取

对于大多数器件，必须先暂停您的程序（Finish Debugger Session），之后才能读取器件存储器。对于某些器件，可以在调试模式下进行读取（调试读取）此时您将知道它是可用的，因为“Read Device Memory”图标将在调试时不灰显。

当前，MPLAB REAL ICE 在线仿真器和 MPLAB ICD 3 支持调试读取。

调试读取将以目标振荡器的速度进行，所以如果目标正以极低速度运行，则读取可能需要很长时间。您可以强制进行快速 ICSP 读取，方法是先完成调试会话再执行读取，因为当不处于调试会话中时，将总是进行 ICSP 读取。

4.22 查看调用堆栈

对于 16 位和 32 位器件，可以通过软件 **Call Stack**（调用堆栈）窗口查看正在执行的 C 代码中的 **CALL** 和 **GOTO** 指令。该窗口不适用于汇编代码。（使用调用堆栈时，建议关闭代码优化。）

Call Stack 窗口将显示函数及其参数，按照它们在正执行的程序中的调用顺序列出。

要查看调用堆栈：

1. 调试运行，然后暂停程序。
2. 选择 **Window>Debugging>Call Stack**。此时将打开 **Call Stack** 窗口。

关于调用堆栈的更多信息，请参见 NetBeans 的帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb](#)（使用 gdb 调试 C/C++/Fortran 应用程序）>[Using the C/C++/Fortran Call Stack](#)（使用 C/C++/Fortran 调用堆栈）。

4.23 对器件编程

调试完代码之后，可以将它编程到目标器件上。

4.23.1 设置项目编程属性

在 **Project Properties**（项目属性）窗口中设置编程选项：

1. 在 **Projects** 窗口中右键单击项目名称，然后选择“**Properties**”（属性）。
2. 在“**Categories**”（类别）下，单击将用于编程代码的硬件工具，例如 **PM3**。
3. 复查“**Memories to Program**”（要编程的存储器）选项类别下的设置。如果希望使用 **Preserve Memory**（保护存储器）选项，请确保代码未进行代码保护。代码通过以下方式保护：编程器读取它需要保存的部分、执行器件批量擦除、重编程器件，然后使用先前保存的内容重写保护的区域。
4. 复查“**Program Options**”（编程选项）选项类别下的设置。
5. 根据您使用的硬件工具，可能还会有其他编程选项类别。复查每个类别，确保设置对于您的项目是正确的。

关于编程选项的更多信息，请参见硬件工具文档。

按照需要设置编程选项后，您可以继续对器件编程。

4.23.2 执行编程

要使用调试后的代码对目标器件编程，请单击工具条按钮 **Make and Program Device Project**。

表 4-10 列出了其他与编程相关的功能。第一个功能通过单击按钮来激活。对于其他功能，请单击按钮图标旁边的向下箭头。

表 4-10: 工具条按钮上的编程功能

按钮图标	功能	详细信息
	Make and Program Device	对项目进行编译（如需要）并对器件进行编程。程序将在编程完成后立即开始执行。
	Program Device for Debugging (针对调试对器件编程)	将使用调试映像对器件进行编程。程序将在编程完成后立即开始执行。
	Program Device for Production (针对生产对器件编程)	将使用生产映像对器件进行编程。程序将在编程完成后立即开始执行。
	Programmer to Go PICkit 3 (使用 PICkit 3 脱机编程)	使用 PICkit 3 的脱机编程（Programmer to Go）功能。
	Read Device Memory	将目标存储器中的内容传输到 MPLAB X IDE。
	Read Device Memory to File (将器件存储器读取至文件)	将目标存储器的内容传输到指定文件中。
	Read EE/Flash Data Memory to a File (将 EE/ 闪存数据存储器读取至文件)	将目标数据存储器中的内容传输到指定文件中。
	Hold In Reset	使器件在复位和运行之间切换。

注: 不是所有编程功能都包含在 MPLAB X IDE 中。关于更多编程支持，请参见 MPLAB X IDE 安装随附的 MPLAB IPE。

注:

第 5 章 附加任务

5.1 执行附加任务

以下步骤说明如何在 MPLAB X IDE 中执行更多的任务。

1
处理项目

1. 在 MPLAB X IDE 中，通过使用导入 MPLAB 旧项目向导打开 MPLAB IDE v8 项目。
2. 使用预编译项目导入向导打开预编译的映像（Hex、COF 或 ELF）。
3. 使用可装入项目和文件合并或替换项目十六进制文件。一种常见的应用是使用可装入对象来合并自举程序和应用程序代码，如可装入项目和文件：自举程序中所述。
4. 创建库项目以将其输出编译为库。
5. 从其他嵌入式项目或示例项目创建项目。
6. 处理其他类型的文件，而不只是 Microchip 文件。此外，修改或创建代码模板来更改在项目中使用的默认文件模板。
7. 在项目中切换硬件或语言工具。
8. 为现有项目修改项目文件夹和编码。
9. 使用并行 **make** 加快编译速度。

2
调试代码

1. 使用跑表来确定断点之间的时间。
2. 查看 **Disassembly** 窗口来查看反汇编的代码。
3. 使用查看调用图来浏览函数调用。
4. 查看仪表盘显示来查看项目信息，例如断点资源、校验和和存储器使用情况。

3
管理代码

1. 通过使用重构和性能分析工具改善您的代码*。
2. 通过使用内置的文件历史记录或版本控制系统控制源代码。
3. 通过使用团队服务器和问题跟踪系统在代码开发和错误跟踪方面进行协作*。

4
添加功能

1. 添加插件工具来协助代码开发。

* 要查看该功能，请参见 **Start Page, My MPLAB X IDE** 选项卡，“Extend MPLAB”部分，“Selecting Simple or Full-Featured Menus”（选择简单或全功能菜单）主题。

5.2 导入 MPLAB 旧项目

Import Legacy Project 向导可以将 MPLAB IDE v8 项目导入 MPLAB X IDE 项目，但需要注意以下几点：

- 在 MPLAB IDE v8 的工作区中保存的设置（例如工具设置）将不会转移到新的 MPLAB X IDE 项目中。关于工作区中所存储内容的信息，请参见 MPLAB IDE v8 帮助（[MPLAB IDE Reference \(MPLAB IDE 参考\)](#) > [Operational Reference \(操作参考\)](#) > [Saved Information \(保存的信息\)](#))。项目设置（如编译器、链接器和汇编器选项）将传输到新的 MPLAB X IDE 项目中。
- 对于使用用于 PIC24 MCU 和 dsPIC DSC 的 MPLAB C 编译器（亦称 MPLAB C30）和 COFF 调试文件格式的 MPLAB IDE v8 项目：
 - MPLAB X IDE 项目将转换为 ELF/DWARF 调试文件格式，除非该项目使用了 COFF 库，后面这种情况下项目格式将继续使用 COFF 格式。
 - MPLAB IDE v8 对于文件扩展名不区分大小写，例如 .c 和 .C。但是，MPLAB X IDE 是区分大小写的，并将 .c 与 C 代码文件关联，将 .C 与 C++ 代码文件关联。因此，如果导入 C 代码文件指定为 .C 的 MPLAB IDE v8 项目，则 MPLAB X IDE 将 .C 文件重命名为 .c，以避免错误的编译器行为。

5.2.1 打开向导

有两种方法可以打开向导——Start Page 选项和 New Project 选项。

5.2.1.1 START PAGE 选项

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Import MPLAB Legacy Project”（导入 MPLAB 旧项目）链接。或选择 [File>Import>MPLAB IDE v8 Project](#)。
- 此时将打开 “Import Legacy Project” 向导。

5.2.1.2 NEW PROJECT 选项

要打开 New Project 向导，请执行以下操作之一：

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Create New Project” 链接。
- [File>New Project](#)（或 Ctrl+Shift+N）

向导将启动，指导您完成新项目设置。

- **步骤 1. 选择项目：**选择 “Microchip Embedded” 类别，然后选择项目类型 “Existing MPLAB IDE v8 Project”。
- 此时将打开 “Import Legacy Project” 向导。

5.2.2 Import Legacy Project 向导

按照以下步骤来导入 MPLAB IDE v8 项目。单击 **Next>** 移至下一步。

- **步骤 1 或 2. 导入旧项目：**输入或浏览至旧项目。
- **步骤 2 或 3. 选择器件：**从 “Device” 下拉列表中选择将在应用中使用的器件。要缩小选择列表，请先选择 “Family”。

- **步骤 3 或 4. 选择仿真头：** 如果存在可用于选定器件的仿真头，则会出现该步骤。要确定调试是否需要仿真头，或器件是否具有片内调试电路，请参见 *Processor Extension Pak and Debug Header Specification*（DS51292 或联机帮助）。然后选择是否使用仿真头。
- **步骤 4 或 5. 选择工具：** 从列表中选择将用于开发应用程序的开发工具。工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示尚不支持。
- **步骤 5 或 6. 选择编译器：** 从列表中选择将用于开发应用程序的语言工具（编译器）。工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示不支持。
- **步骤 6 或 7. 选择项目名称和文件夹：** 建议不要更改默认名称和位置，以保持两个项目的可维护性。

文件位置：

新项目不会将源文件复制到其文件夹中，而是引用 v8 文件夹中的文件位置。要创建一个独立的 MPLAB X IDE 项目，需要创建一个新项目，并将 MPLAB IDE v8 源文件复制到其中。

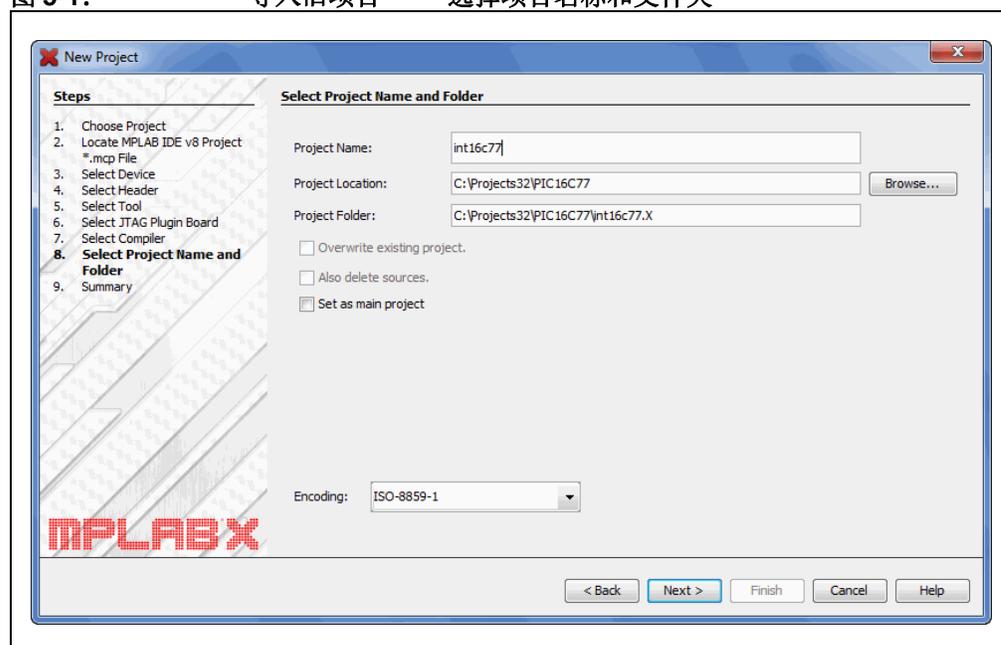
主项目：

选中复选框可使该项目在导入后成为主项目。

文件格式：

从 MPLAB IDE v8 导入项目时，“ISO-8859-1”是默认使用的字符编码。应选择与所导入项目中所用编码匹配的编码。例如，如果 MPLAB IDE v8 格式为“950 (ANSI/OEM Traditional Chinese Big5)”（950 (ANSI/OEM 繁体中文 Big5)），则从下拉列表中选择“Big5”。

图 5-1: 导入旧项目 —— 选择项目名称和文件夹



- **步骤 7 或 8. 摘要：** 在单击 **Finish** 之前查看摘要。如果有任何内容不正确，请使用 **Back**（后退）按钮后退并更改它。

旧项目将在 **Projects** 窗口中打开。

5.3 预编译项目

通过使用 **Import Image File**（导入映像文件）向导，基于预编译的可装入映像（Hex、COF 或 ELF 文件）创建项目。

注： 要将预编译映像编程到器件中，需要单击 **Make and Program Device** 按钮，虽然它将仅对器件进行编程（无 **make** 操作）。

有两种方法可以打开向导——**Start Page** 选项和 **New Project** 选项。

5.3.1 Start Page 选项

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Import Hex (Prebuilt) Project”（导入十六进制（预编译）项目）链接。或选择 **File>Import>Hex/ELF (Prebuilt) File (Hex/ELF (预编译) 文件)**。
- 此时将打开 “Import Image File” 向导。

5.3.2 New Project 选项

要打开 **New Project** 向导，请执行以下操作之一：

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Create New Project” 链接。
- File>New Project**（或 **Ctrl+Shift+N**）

向导将启动，指导您完成新项目设置。

- 步骤 1. 选择项目：** 选择 “Microchip Embedded” 类别，然后选择项目类型 “Prebuilt (Hex, Loadable Image) Project”（预编译（十六进制可装入映像）项目）。
- 此时将打开 “Import Image File” 向导。

5.3.3 Import Image File 向导

按照以下步骤来导入映像文件。单击 **Next>** 移至下一步。

- 步骤 1 或 2. 导入映像文件：** 选择映像文件的名称和位置。您可以浏览至一个位置。
- 步骤 2 或 3. 选择器件：** 从 “Device” 下拉列表中选择将在应用中使用的器件。要缩小选择列表，请先选择 “Family”。
- 步骤 3 或 4. 选择仿真头：** 如果存在可用于选定器件的仿真头，则会出现该步骤。要确定调试是否需要仿真头，或器件是否具有片内调试电路，请参见 *Processor Extension Pak and Debug Header Specification*（DS51292 或联机帮助）。然后选择是否使用仿真头。
- 步骤 4 或 5. 选择工具：** 从列表中选择将用于开发应用程序的开发工具。工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示尚不支持。
- 步骤 6 或 7. 选择项目名称和文件夹：** 选择新项目的名称和位置。您可以浏览至一个位置。
- 步骤 7 或 8. 摘要：** 在单击 **Finish** 之前查看摘要。如果有任何内容不正确，请使用 **Back** 按钮后退并更改它。

新项目将在 **Projects** 窗口中打开。

关于以十六进制形式导出项目的信息，请参见第 12.11.2 节 “**Projects** 窗口——项目菜单”。

5.4 可装入项目和文件

使用可装入项目和文件来合并项目、合并十六进制文件、合并项目与十六进制文件或替换项目十六进制文件。**hexmate** 应用程序用于将项目或装入的十六进制文件合并为一个文件。（关于该应用程序的详细信息，请参见所安装的 **MPLAB XC8** 编译器的 docs 文件夹中的支持文档。）

可装入项目或文件对于创建合并的自举程序和应用程序代码很有用。请参见第 5.5 节“可装入项目和文件：自举程序”。

下面列出了当前项目和可装入对象的组合。

表 5-1: 可装入对象组合

当前项目	可装入对象	注意事项
独立 现有 MPLAB IDE v8 库	独立	无
	十六进制文件	无
	COF/ELF 文件	• 可以调试，但不能编译。Output（输出）窗口中将会显示错误。
预编译（十六进制）	十六进制文件	• 不会自动检查是否有重叠的存储器区域。 • 可以调试，但不能 build。Build 按钮将被禁止。
	COF/ELF 文件	
预编译（COF/ELF）	十六进制文件	
	COF/ELF 文件	

下面列出了这些选项。

表 5-2: 可装入对象选项

Add Loadable Project(s) (添加可装入项目)	将一个或多个现有项目装入当前项目中。在编译当前项目时，将编译所有项目并将十六进制文件合并为一个。所有调试文件也将进行合并（COFF 或 ELF）。
Add Loadable File(s) (添加可装入文件)	将一个或多个现有十六进制文件装入当前项目中。在编译当前项目时，十六进制文件将与其他十六进制文件合并为一个文件。 注： 您将无法再调试包含十六进制文件的项目。请使用可装入项目进行调试。
Add Alternate File（添加备用文件）	装入要使用的备用十六进制文件。 该选项用于提供一个编译后步骤，您可以在该步骤中将项目十六进制文件复制或移动到另一个位置，使用诸如 hexmate 之类的工具来将您的文件和另一个十六进制文件合并，然后将文件装回到 IDE 中。

要设置和使用可装入对象，请参见：

- Project 窗口 —— 可装入对象设置
- Project Properties 窗口 —— 装入设置
- 使用可装入对象的首选方法

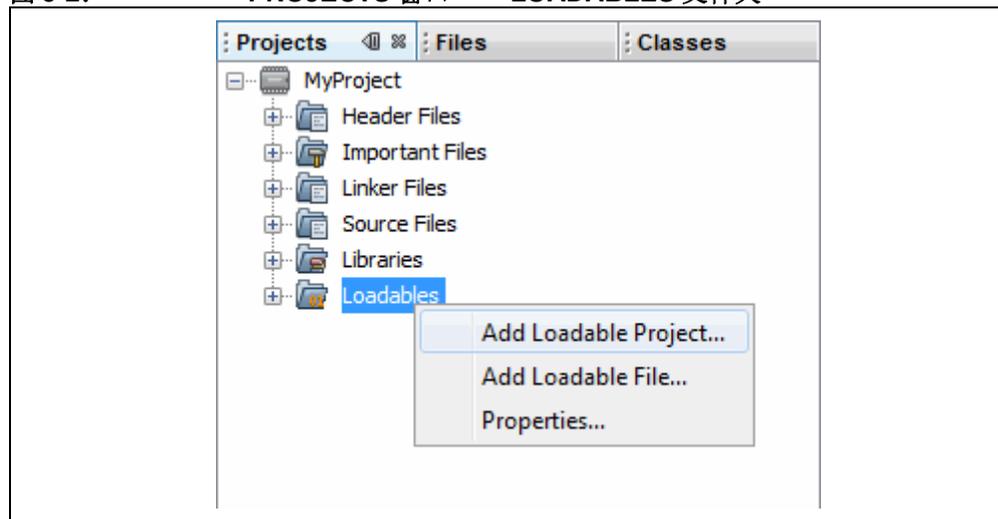
5.4.1 Project 窗口 —— 可装入对象设置

右键单击 Projects 窗口中的“Loadables”（可装入对象）文件夹（图 5-2）并选择一个选项：

- **Add Loadable Project**—— 选择将现有项目添加到当前项目中。重复该步骤来添加更多项目。
- **Add Loadable Files**—— 选择将现有十六进制文件添加到当前项目中。重复该步骤来添加更多十六进制文件。
- **Properties**—— 打开 Project Properties 窗口来进行装入。请参见第 5.4.2 节“Project Properties 窗口 —— 装入设置”。

编译当前项目以编译所有项目，并将十六进制文件合并为一个。所有调试文件也将进行合并。

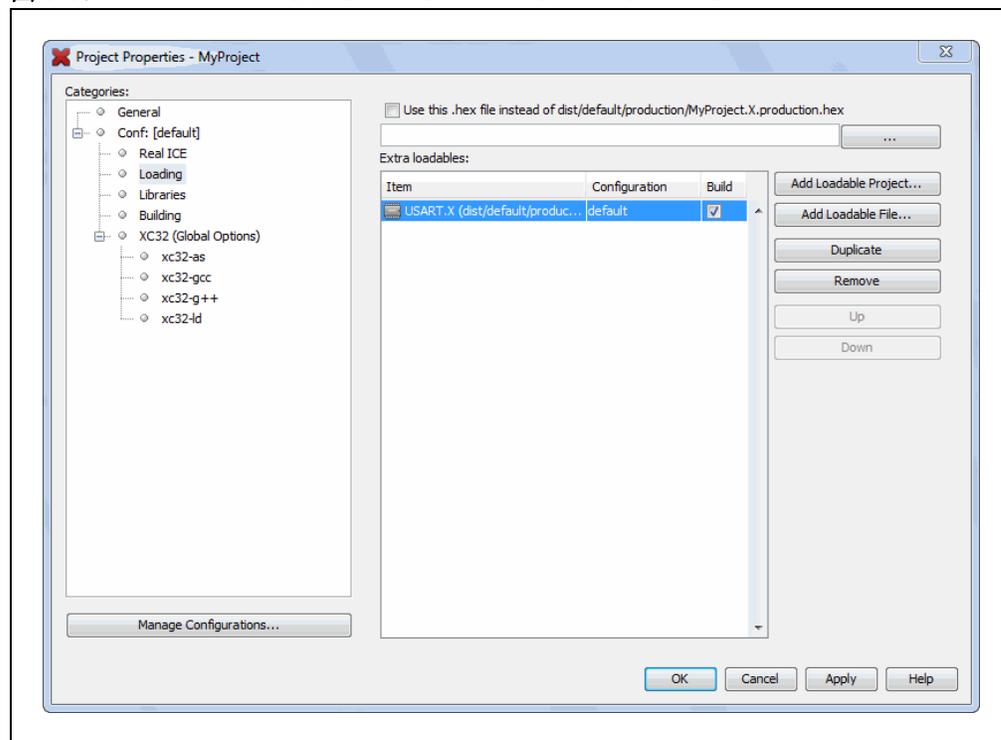
图 5-2: PROJECTS 窗口 ——LOADABLES 文件夹



5.4.2 Project Properties 窗口 —— 装入设置

打开 Project Properties 窗口 (*File>Project Properties*)，然后单击 “Loading” (装入)。

图 5-3: PROJECT PROPERTIES——LOADING



要将当前项目与其他项目合并

1. 单击 **Add Loadable Project**。浏览至一个项目，选择它并单击 **Add**。
2. 在 “Configuration” (配置) 下的下拉框中选择要用于该项目的编译配置。如果尚未向该项目添加其他配置，则将只会看到 “default”。
3. 如果希望在编译当前项目时编译该项目，请确保选中 “Build” 复选框。
4. 如果已添加了多个项目，则此处显示的顺序将决定十六进制文件添加到当前项目的十六进制文件中的顺序。使用 **Up** (向上) 和 **Down** (向下) 可更改顺序。
5. 单击 **Apply** 或 **OK** 接受更改。

下次编译当前项目时，也会编译此处列出的项目 (如果选中了 “Build”)，它们的十六进制文件将与当前项目的十六进制文件合并，以创建单个输出十六进制文件。所有调试文件也将进行合并。

要将当前项目十六进制文件与其他十六进制文件合并

1. 单击 **Add Loadable File**。浏览至一个十六进制文件，选择它并单击 **Add**。
2. 如果已添加了多个文件，则此处显示的顺序将决定十六进制文件添加到当前项目的十六进制文件中的顺序。使用 **Up** 和 **Down** 可更改顺序。
3. 单击 **Apply** 或 **OK** 接受更改。

下次编译当前项目时，此处列出的十六进制文件将与当前项目的十六进制文件合并，以创建单个输出十六进制文件。

要装入备用十六进制文件：

1. 单击以选中 “Load this hex file on run or program builds only (debug not affected)”（仅在运行或程序编译时装入该十六进制文件（调试不受影响））。
2. 浏览至所需的十六进制文件并选择它。选择引用文件的方式（Auto、Relative 和 Absolute），然后单击 **Add**。

下次编译当前项目时，备用十六进制文件将在编译完成时装入。

5.4.3 使用可装入对象的首选方法

使用可装入对象的建议方法为：

1. 选择设置了器件配置位的项目并将其设为主项目（单击右键并选择 “Set as Main Project”）。可装入对象不应具有器件配置设置，因为这会与主项目发生冲突。
2. 将其他项目作为可装入对象添加到该主项目中。如果正装入的项目具有多个项目配置，请确保在装入时指定配置。

将包含可装入对象的项目指定为主项目时，可以确保在按下 **Build** 按钮时，编译操作可以获得所有可装入对象中的更改。

5.5 可装入项目和文件：自举程序

要将自举程序与应用程序代码合并：

1. 为应用程序创建一个项目，并为自举程序创建一个项目。
2. 将自举程序项目或十六进制文件装入应用程序项目中。关于如何执行该操作的信息，请参见第 5.4.1 节 “Project 窗口 —— 可装入对象设置” 和第 5.4.2 节 “Project Properties 窗口 —— 装入设置”。

下次编译应用程序项目时，所产生的十六进制文件将为一个合并的自举程序 / 应用程序十六进制文件。所有调试文件也将进行合并。

关于编译错误，请参见第 5.4.3 节 “使用可装入对象的首选方法” 或以下几节。

注意事项 1：用于 PIC18 MCU 的 MPLAB C 编译器（MPLAB C18）

该编译器提供了应用程序启动代码（c018x.o），它从复位向量（地址 0）处开始，用于初始化软件堆栈，以及可选地初始化 idata 段，然后跳转到 main()。如果该启动代码保留在应用程序中，则总是会与自举程序代码复位发生冲突，您会收到一个关于数据冲突的链接器错误消息。

解决方法是对启动代码进行编辑，使之从地址 0 之外的位置开始。

注意事项 2：MPLAB XC8——PIC18 MCU 示例

以下 Microchip 网络讲座详细介绍了如何使用 MPLAB XC8 和 MPLAB X IDE 合并用于 PIC18 MCU 的引导程序与应用程序代码：
“Linking PIC18 Bootloaders & Applications”

5.6 库项目

创建新的库项目，该库项目使用 IDE 生成的 makefile 将项目编译为库文件而不是可执行文件。

图 5-4: 库项目示例

```

Output - MyPIC32Lib (Build)
make -f nbproject/Makefile-default.mk SUBPROJECTS= .build-conf
make[1]: Entering directory `~/cygdrive/c/Documents and Settings/c08227/MPLABXProjects/MyPIC32Lib'
make -f nbproject/Makefile-default.mk dist/default/mypic32lib.a
make[2]: Entering directory `~/cygdrive/c/Documents and Settings/c08227/MPLABXProjects/MyPIC32Lib'
mkdir -p build/default/production
rm -f build/default/production/library32.o.d
C:\Program Files\Microchip\MPLAB C32\bin\pic32-gcc.exe -x c -c -mprocessor=32MK360F512L -MMD -MF build/de
mkdir -p dist/default
C:\Program Files\Microchip\MPLAB C32\bin\pic32-ar.exe r dist/default/mypic32lib.a build/default/production
make[2]: Leaving directory `~/cygdrive/c/Documents and Settings/c08227/MPLABXProjects/MyPIC32Lib'
C:\Program Files\Microchip\MPLAB C32\bin\pic32-ar.exe: creating dist/default/mypic32lib.a
make[1]: Leaving directory `~/cygdrive/c/Documents and Settings/c08227/MPLABXProjects/MyPIC32Lib'

BUILD SUCCESSFUL (total time: 938ms)

```

要开始操作，请通过以下操作之一打开 New Project 向导：

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Create New Project” 链接。
- **File>New Project** (或 Ctrl+Shift+N)

向导将启动，指导您完成新项目设置。单击 **Next>** 移至下一步。

- **步骤 1. 选择项目：**选择 “Microchip Embedded” 类别，然后选择项目类型 “Library Project”。
- **步骤 2. 选择器件：**从 “Device” 下拉列表中选择将在应用中将使用的器件。要缩小选择列表，请先选择 “Family”。
- **步骤 3. 选择仿真头：**如果存在可用于选定器件的仿真头，则会出现该步骤。要确定调试是否需要仿真头，或器件是否具有片内调试电路，请参见 *Processor Extension Pak and Debug Header Specification* (DS51292 或联机帮助)。然后选择是否使用仿真头。
- **步骤 4. 选择工具：**从列表中选择将用于开发应用程序的开发工具。工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示尚不支持。
- **步骤 5. 选择编译器：**从列表中选择将用于开发应用程序的语言工具（编译器）。工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示不支持。
- **步骤 6. 选择项目名称和文件夹：**选择新项目的名称和位置。您可以浏览至一个位置。

新项目将在 **Projects** 窗口中打开。

5.7 其他嵌入式项目

MPLAB X IDE 可以基于选定的其他嵌入式项目创建项目。

1. 选择 *File>New Project*。
2. 单击 “Categories” 下的 “Other Embedded”（其他嵌入式），并从可用嵌入式项目的列表中进行选择。
3. 继续创建一个 MPLAB X IDE 项目。

该功能会将现有文件导入 MPLAB X IDE 项目。尚不提供其他嵌入式项目设置或代码的转换。

关于如何使用 MPLAB X IDE 的信息，请参见：

- 第 3 章 “教程”
- 第 4 章 “基本任务”

关于可用编译器的信息，请参见：

<http://www.microchip.com/xc>

5.8 示例项目

创建示例项目可以帮助您了解关于 Microchip 器件、工具和 MPLAB X IDE 的信息。

1. 选择 *File>New Project*。
2. 在 “Categories” 下单击 “Samples（示例）>Microchip Embedded”，并从可用嵌入式项目（闪烁演示板指示灯的项目）或模板项目的列表中进行选择。关于更多信息，请参见 “说明”。

演示板的部件编号如下：

Explorer 16 演示板：DM240001

PICDEM™ 2 Plus：DM163022-1

5.9 处理其他类型的文件

选择 *File>New* 时，会为您列出许多类型的文件。先前已介绍了使用 Microchip 编译器文件。但是，您还可以根据项目语言工具或需要选择其他类型的文件，以创建一个特定文件。

表 5-3: 文件类型

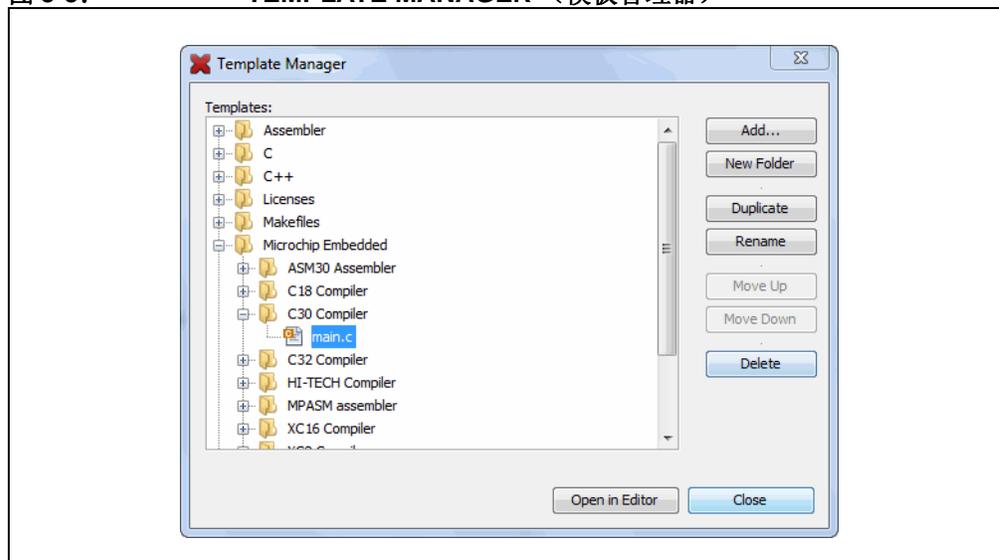
类别	文件类型
Microchip Embedded	用于 MPLAB X IDE 中支持的语言工具的文件 选择您的编译器文件夹可查看可用的文件类型。
C	通用 C 文件
C++	通用 C++ 文件
Assembler（汇编器）	通用汇编文件
Shell Scripts（Shell 脚本）	Shell 脚本文件：Bash、C 和 Korn 等
Makefiles	Makefile 文件
XML	XML 文件
其他	其他类型的文件，例如 HTML 和 JavaScript 等 如果发现此处未列出所需的文件类型，请选择 “Empty File”（空文件）。在下一个窗口中，使用所需扩展名来命名文件。

5.10 修改或创建代码模板

创建要添加到项目中的文件时（第 4.8 节“创建新文件”），将会对新文件使用模板。要更改该模板，请选择 **Tools>Templates**（模板），然后选择“Open in Editor”（在编辑器中打开）来编辑模板。您也可以使用该对话框中的“Add”或“Duplicate”（复制）来创建新模板。

可以使用“New Folder”（新建文件夹）来创建用以存放模板的新文件夹。请注意，MPLAB X IDE 会过滤掉除 Microchip Embedded、Shell Scripts、Makefiles 和 Other 之外的所有文件夹，因此文件或文件夹应创建在这些文件夹下。

图 5-5: TEMPLATE MANAGER（模板管理器）



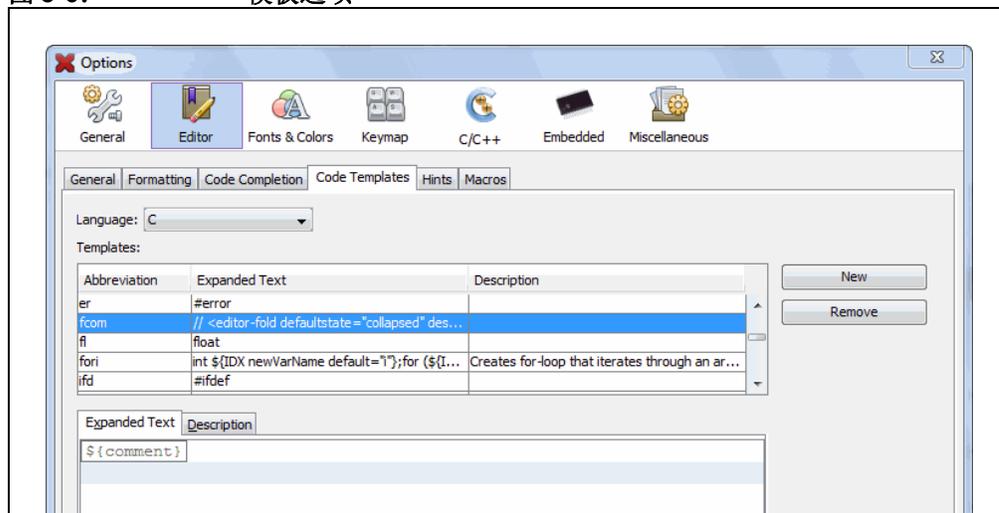
您可以通过选择 **Tools>Options**（对于 Mac OS X 为 *mplab_ide>Preferences*），**Editor** 按钮，**Code Templates**（代码模板）选项卡来设置模板选项（见下图）。

对于 C 代码，需要注意“fcom”选项。在 Editor 窗口中输入“fcom”，然后按“Tab”键可在源代码中插入以下文本：

```
// <editor-fold defaultstate="collapsed" desc="comment" >
// </editor-fold>
```

该选项使您可以隐藏 / 查看代码段。

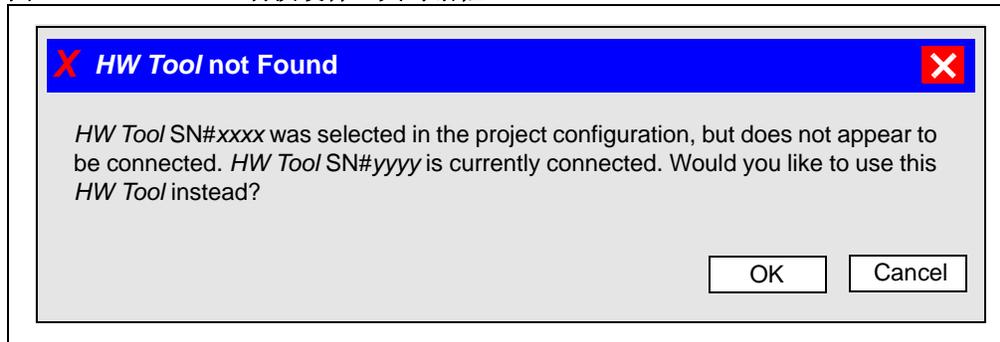
图 5-6: 模板选项



5.11 切换硬件或语言工具

当打开一个现有项目，并且已连接的硬件工具与在项目中指定的不同时，MPLAB X IDE 会弹出一个对话框，询问是否要将新的硬件工具设为项目工具。

图 5-7: 切换硬件工具对话框



您还可以插上两个或更多硬件工具，并在 **Project Properties** 对话框 ([File>Project Properties](#)) 中切换它们。

要在编译器工具链（语言工具）的不同版本之间切换，请再次使用 **Project Properties** 对话框。

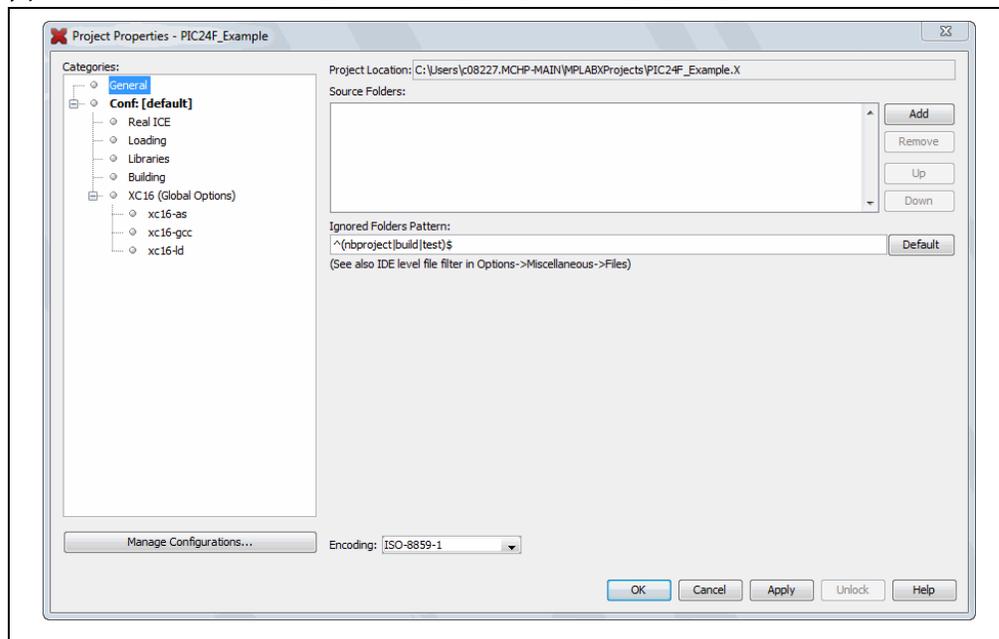
5.12 修改项目文件夹和编码

当您创建项目时，需要指定项目文件夹和编码。创建项目之后，您可以使用 **Project Properties** 窗口中的“**General**”（常规）类别添加或忽略项目文件夹和更改项目编码。

表 5-4: PROJECT PROPERTIES——GENERAL 类别

选项	说明
Project Location (项目位置)	查看当前项目位置。 要更改项目位置，请参见第 8.8 节“移动、复制或重命名项目”。
Source Folders (源文件夹)	添加供 MPLAB X IDE 通过搜索来查找项目文件的文件夹。 注： 添加项目文件夹之外的文件可能会使项目的可移植性降低。
Ignored Folders Pattern (忽略文件夹模式)	根据指定的正则表达式模式忽略项目文件夹中的文件夹。
Encoding (编码)	更改项目编码。 该选择将指定代码的语法着色设置，它可以在 Tools>Options （对于 Mac OS X 为 mplab ide>Preferences ）， Fonts and Colors （字体和颜色）按钮， Syntax （语法）选项卡下进行编辑。

图 5-8: PROJECT PROPERTIES—GENERAL



5.13 加快编译速度

根据您的计算机的配置，您可能可以使用并行 `make`（见第 12.12.2 节“**Project Options 选项卡**”）来加快项目编译速度。并不是所有语言工具都支持并行 `make`。

另一种选项是考虑您的操作系统（OS）。一些操作系统的文件访问速度更快。MPLAB X IDE 支持 Windows、Linux 和 Mac 操作系统。研究一下哪一种操作系统可能适合您。

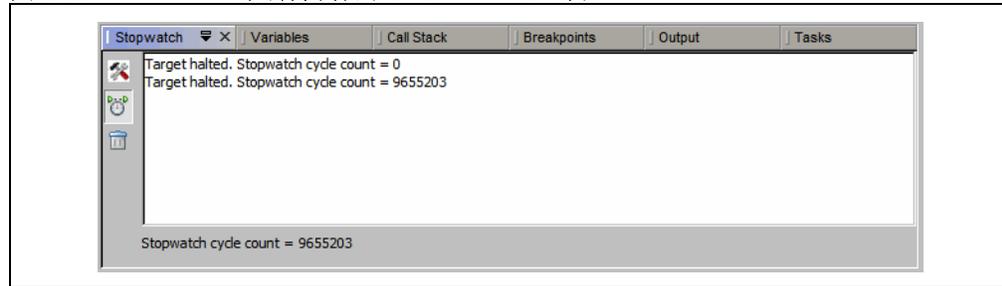
5.14 使用跑表

使用跑表来确定两个断点之间的时间。

要使用跑表：

1. 在要启动跑表的位置添加一个断点。
2. 在要停止跑表的位置添加另一个断点。
3. 选择 **Window>Debugging>Stopwatch (跑表)**。单击窗口左侧的 Properties 图标，并选择开始和停止断点。
4. 再次调试运行程序，以获取跑表计时结果。

图 5-9: 具有内容的 STOPWATCH 窗口



跑表在窗口左侧具有以下图标：

图标	说明
Properties	设置跑表属性。选择一个当前断点或触发信号来启动跑表，另一个断点或触发信号用于停止跑表。
Reset Stopwatch on Run (在运行时复位跑表)	在开始运行时将跑表时间复位为零。
Clear History (清除历史记录)	清除跑表窗口。
Clear Stopwatch (清除跑表)	(仅对于软件模拟器) 在复位器件之后复位跑表。

5.15 查看 DISASSEMBLY 窗口

在该窗口中查看反汇编的代码。选择 **Window>Output>Disassembly Listing File (反汇编列表文件)** 打开窗口。

在链接器生成的列表文件中也可以找到该信息。通过选择 **File>Open File (打开文件)** 并浏览查找 `ProjectName.lst` 打开该文件。

查看整个反汇编文件的一种快速方法是在 Disassembly 窗口中单击右键并选择“Disassembly Listing File”。

注： 反汇编窗口将反汇编每一条指令，但没有与指令相关联的存储区历史记录。因此，在窗口中显示的 SFR 名称将为对应于存储区 0 的名称。

5.16 查看调用图

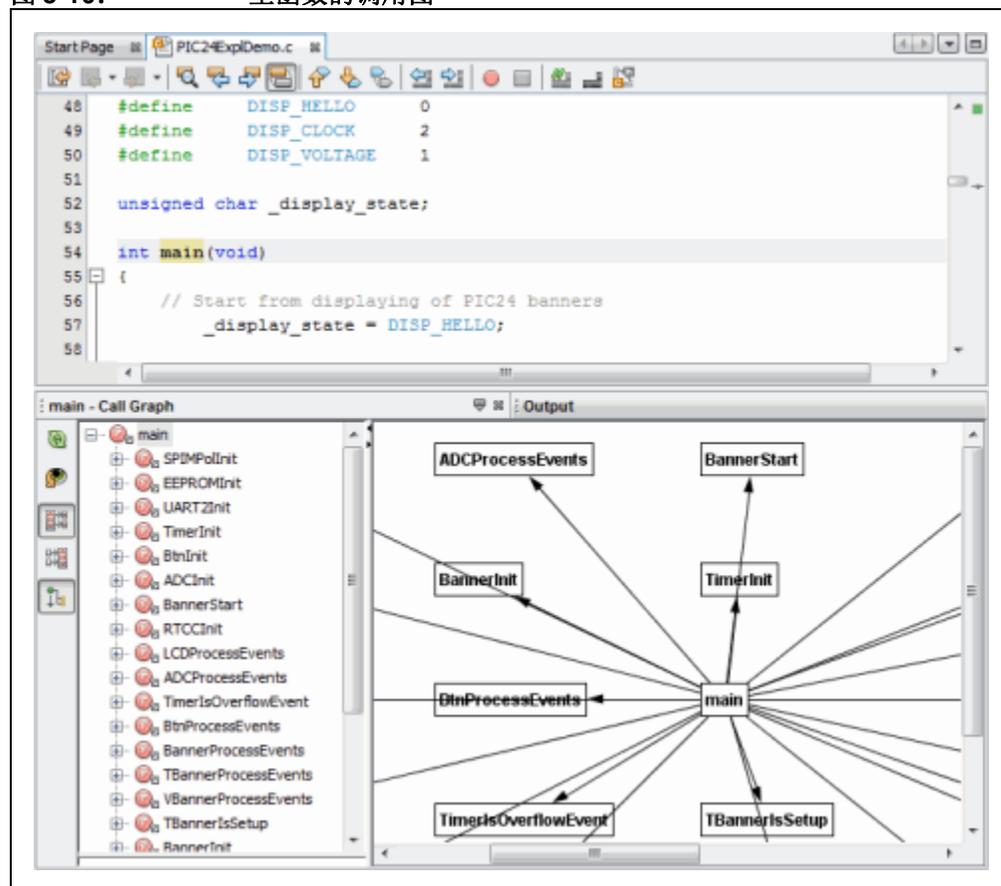
Call Graph（调用图）窗口会显示从选定函数调用的函数或调用该函数的函数的树视图。

要查看调用图：

右键单击一个函数并从下拉菜单中选择“Show Call Graph”（显示调用图）。

更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Working With C/C++/Fortran Projects>Navigating Source Files and Projects](#)（浏览源文件和项目）>[Using the Call Graph](#)（使用调用图）。

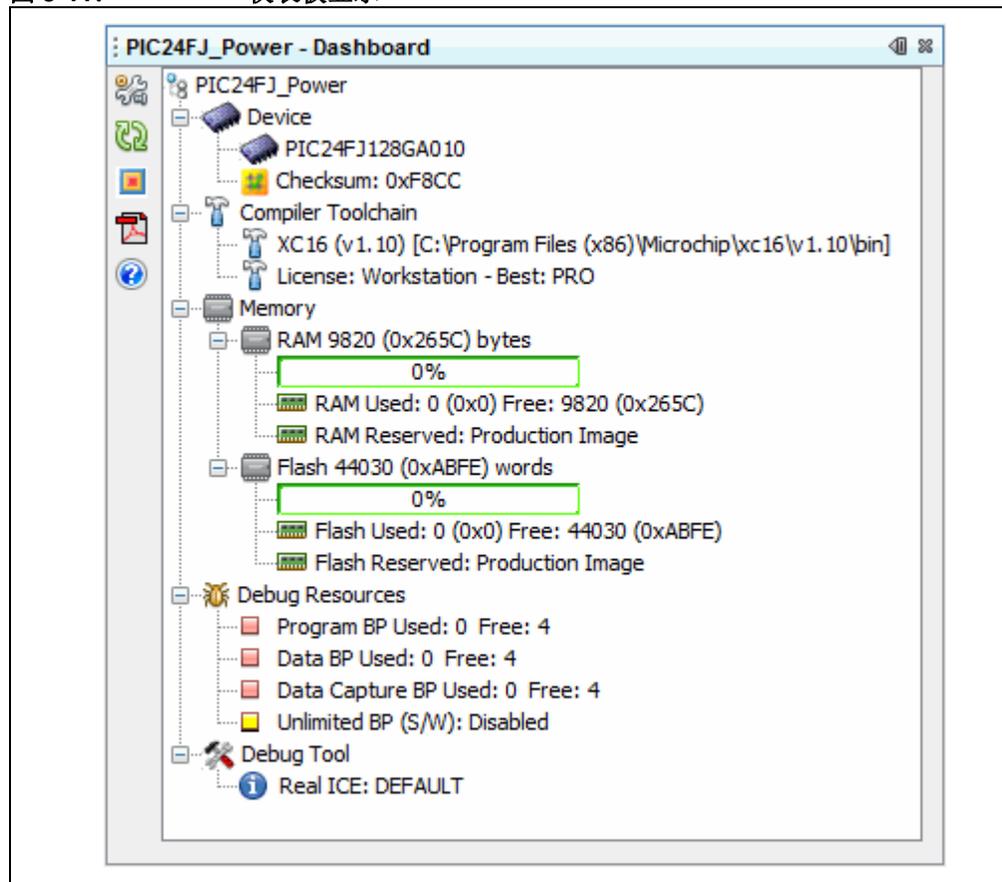
图 5-10: 主函数的调用图



5.17 查看仪表板显示

选择 *Window>Dashboard* 可显示项目信息。

图 5-11: 仪表板显示



显示的项目将为:

- 如果未选择主项目 (*Run>Set Main Project (设置主项目) >None*), 则显示 Projects 窗口中的活动项目。在 Projects 窗口单击某个项目, 可使其成为活动项目。
- 主项目。将不显示任何其他项目 (活动或非活动)。

下表列出了仪表板窗口的功能。

表 5-5: 仪表板组

组	定义和内容
Device	<ul style="list-style-type: none"> • 项目器件 • 在运行或调试运行期间生成的任何状态标志 • 已编译程序的校验和, 即, 必须进行编译才能看到它
Compiler Toolchain (编译器工具链)	<ul style="list-style-type: none"> • 项目工具链 (编译器、汇编器和链接器等) 的名称包括工具版本号和可执行文件的路径。 • 工具链许可类型: PRO (专业版)、STD (标准版) 或 Free (免费版) 关于这些许可类型的更多信息, 请参见语言工具的文档。

表 5-5: 仪表板组 (续)

组	定义和内容
Memory	项目使用的存储器（数据和程序）的类型和大小，以及为调试保留的存储器。 Memory Used （已用存储器）可指示剩余的存储容量。编译器会输出存储器摘要，详细列出程序空间、配置位，ID 单元和 EEPROM（如果处于器件上）的使用情况。这些存储空间的总和（考虑到字大小）应与 Dashboard Flash Used （仪表板已用闪存）一致。如果存储器紧缺，则应检查映射文件。
Debug Resources (调试资源) (断点)	可用于项目器件的硬件断点数量。 当前有多少断点在使用。 项目器件上是否支持软件断点。
Debug Tool (调试工具)	调试工具的连接状态。 硬件调试工具 连接仅在调试运行、运行或编程期间为活动状态。否则，它会处于非活动状态。要使工具连接总是保持活动状态，请转至 Tools>Options （对于 Mac OS X 为 mplab_ide>Preferences ）， Embedded 按钮， Generic Settings 选项卡，并选中“Maintain active connection to hardware tool”。 单击“Refresh Debug Tool Status”（刷新调试工具状态）按钮可查看硬件调试工具固件版本和当前电压。 Simulator（模拟器） 当选择模拟器作为项目调试工具时，会有一个标题为“Click for Simulated Peripherals”（单击查看模拟的外设）的额外条目。单击该项可查看支持哪些外设。

表 5-6: 侧边栏图标

图标	功能
	Project Properties 显示 Project Properties 对话框。
	Refresh Debug Tool Status（刷新调试工具状态） 单击它可查看硬件调试工具详细信息。
	Toggle Software Breakpoint - Enabled/Disabled（切换软件断点 —— 使能 / 禁止） 单击它可使能或禁止软件断点。 图标中会显示该功能的状态： 红色中心：禁止 绿色中心：使能 黑色中心：不支持 —— 调试工具和器件功能的组合
	Open Device Data sheets（打开器件数据手册） 从 Microchip 网站（ http://www.microchip.com/ ）获取器件数据手册。单击可打开已保存的本地数据手册或打开浏览器以转至 Microchip 网站来搜索数据手册。
	Compiler Help（编译器帮助） 单击可打开编译器安装目录的 docs 文件夹中文档的主索引（如可用）。

5.18 改善您的代码

通过使用代码重构和 / 或性能分析来改善您的代码。

注： 要查看该功能，请参见 **Start Page, My MPLAB X IDE** 选项卡，“Extend MPLAB”，“Selecting Simple or Full-Featured Menus” 主题。

重构代码是一种使代码变得简单而无需改变其功能的方法。当前可以对 C 代码执行以下操作：

- 查找整个文件中的函数使用实例
- 重命名整个文件中的函数和参数

更多信息，请参见第 7.5 节 “C 代码重构”。

代码性能分析是在程序运行时 CPU Usage（CPU 使用）、Memory Usage（存储器使用）和 Thread Usage（线程使用）工具进行检查。每当运行 C 项目时，都会自动运行性能分析工具。

5.19 控制源代码

MPLAB X IDE 具有一个内置的本地文件历史记录功能，这是 NetBeans 平台的补充功能。该功能为本地项目和文件提供内置的版本控制支持，类似于传统的版本控制系统。可用工具包括本地 DIFF 和文件恢复。在项目或文件窗口右键单击一个文件可以查看本地历史记录选项。

要查看文件的本地历史记录：

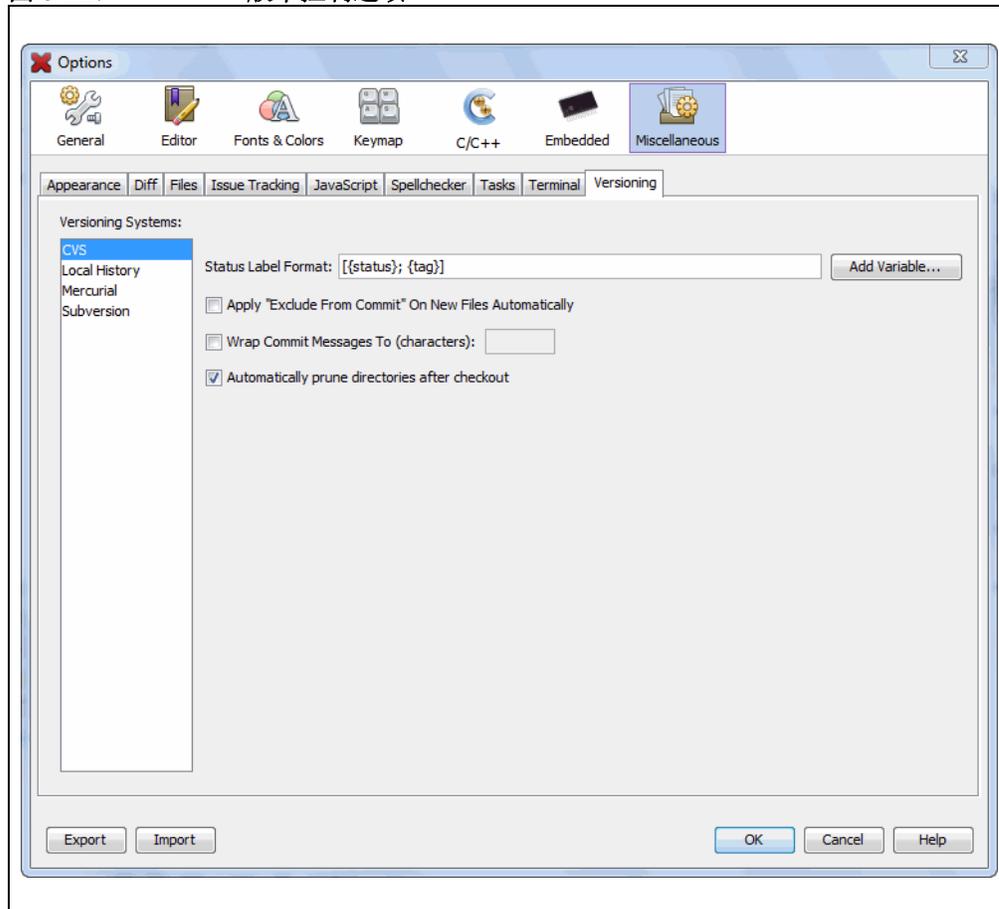
- 在项目或文件窗口中右键单击文件，并选择 **Local History (本地历史记录) > Show Local History (显示本地历史记录)**。此处将会列出过去对文件进行的所有更改。
- 在项目或文件窗口中右键单击文件，并选择 **Local History > Revert to (还原为)**。此时将打开 Revert to 对话框，列出文档的所有先前版本。选择一个并单击 **OK** 可还原为该版本。

如果更偏好使用完整的版本控制系统，也提供了对于 CVS、Subversion 和 Mercurial 的支持。

要使用源代码 / 版本控制：

1. **Team (团队) 菜单** —— 从子菜单中选择版本控制程序并设置该版本控制程序。
2. **Tools > Options (对于 Mac OS X 为 mplab_ide > Preferences)**，**Miscellaneous (杂项)**，**Versioning (版本控制)** —— 设置版本控制选项（见图 5-12）。
3. **Window > Versioning** —— 打开版本控制窗口。

图 5-12: 版本控制选项



需要保存到存储库（repository）中的项目文件：

下表列出了需要或不需要提交到版本控制存储库的项目文件。

表 5-7: 保存到存储库中的项目文件

目录或文件	提交?
项目目录	
Makefile	是
源文件	是
build 目录	否
dist 目录	否
nbproject 目录	
configurations.xml	是
project.properties	是
project.xml	是
Makefile-*	否
Package-*	否
私有目录	否
是：生成项目映像所必需的。 否：这些目录 / 文件会被重新生成，因此不需要保存。	

关于项目结构的更多信息，请参见第 8.3 节“Files 窗口视图”。

关于使用本地文件历史记录和 / 或源代码控制的更多信息，请参见 the NetBeans 帮助主题 *IDE Basics > Version Control and File History*（版本控制和文件历史记录）下的目录。

关于以上源代码控制程序的进一步信息，请参见：

- CVS —— <http://www.nongnu.org/cvs/>
- Subversion —— <http://subversion.tigris.org/>
- Mercurial —— <http://mercurial.selenic.com/>

5.20 在代码开发和错误跟踪方面进行协作

使用 MPLAB X IDE 内支持的团队服务器（例如 Kenai.com）与您所在的小组进行代码开发协作。

注： 要查看该功能，请参见 **Start Page, My MPLAB X IDE** 选项卡，“Extend MPLAB”，“Selecting Simple or Full-Featured Menus” 主题。

支持的菜单项包括：

- **Team**—— 团队服务器主菜单。登录到您的帐户、创建或打开您的项目、分享您的项目、获取资源、发送聊天消息或显示您的联系人列表。
- **File>Open Team Project** (*打开团队项目*)—— 打开现有的团队项目。

通过使用问题跟踪系统（即 Bugzilla™ 和 JIRA®（需要插件）），团队协作跟踪错误。支持的菜单项包括：

- **Windows>Services**—— 右键单击“Issue Tracker”（问题跟踪器）可添加问题跟踪器。
- **Team>Find Issues** (*查找问题*)—— 在 Issue Tracker 窗口中，选择项目的问题跟踪器，选择条件，并单击 **Search**（搜索）。
- **Team>Report Issues** (*报告问题*)—— 在 Issue Tracker 窗口中，选择项目的问题跟踪器，指定问题详细信息，并单击 **Submit**（提交）。

关于团队项目和问题跟踪的更多信息，请参见 NetBeans 帮助主题 **IDE Basics> Collaborative Development** (*协作开发*)。

要了解关于这些工具的更多信息，请参见：

- Kenai——<http://kenai.com/>
- Bugzilla——<http://www.bugzilla.org/>
- JIRA——<http://www.atlassian.com/software/jira/>

5.21 添加插件工具

MPLAB IDE v8 插件工具（例如 DMCI 和 MATLAB）可通过插件管理器（[Tools>Plugins \(插件\)](#)）提供给 MPLAB X IDE 使用。显示在 MPLAB IDE v8 Tools 菜单下的 Macro（宏）控件实际上是一种编辑器功能，所以处于 MPLAB X IDE 的 Edit 菜单下。

- 添加插件
- 升级插件
- 配置更新中心
- 插件代码位置

5.21.1 添加插件

要查看插件和向 MPLAB X IDE 添加插件：

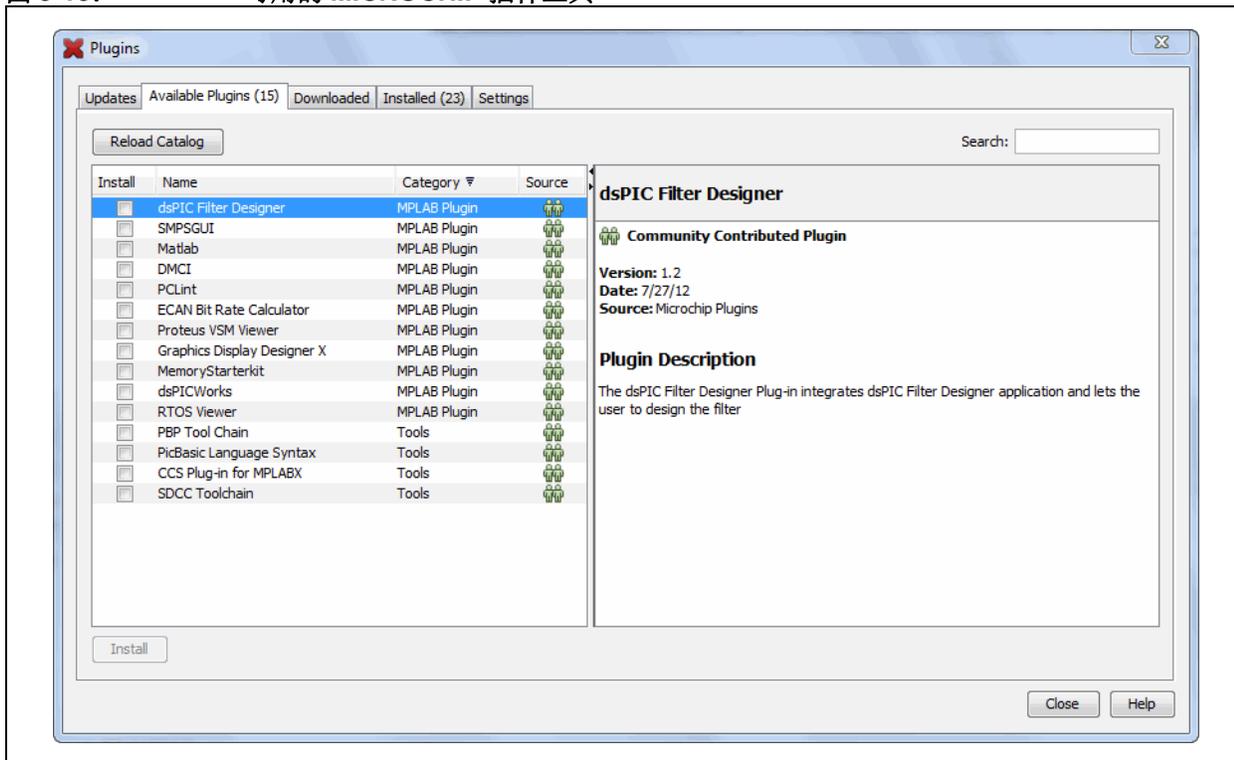
1. 单击 **Available Plugins (可用插件)** 选项卡。
2. 通过选中复选框来选择您的插件。
3. 单击 **Install (安装)**。
4. 按照屏幕说明下载并安装插件。

注： 一些插件可能依赖于其他插件中的模块，以便实现其功能。在这种情况下，插件管理器向您发出警告。

5. 在 [Tools>Embedded](#) 下查找您的工具。如果未看到它，则可能需要关闭并重新打开 MPLAB X IDE。

单击 **Help (帮助)** 按钮可阅读关于安装插件的更多信息。

图 5-13: 可用的 MICROCHIP 插件工具



5.21.2 升级插件

安装 MPLAB X IDE 的新版本将不会更新已安装的插件。插件是针对某个版本的界面进行测试的。不是所有插件都可以在不同版本之间移植，所以不能在新版本中继续使用它们。对于 NetBeans 也是如此。

要将插件升级到更新版本或为 MPLAB X IDE 的更新版本添加它，请使用第 5.21.1 节“添加插件”中的 Plugins 对话框。

5.21.3 配置更新中心

要查看可用插件，必须配置一个或多个更新中心。

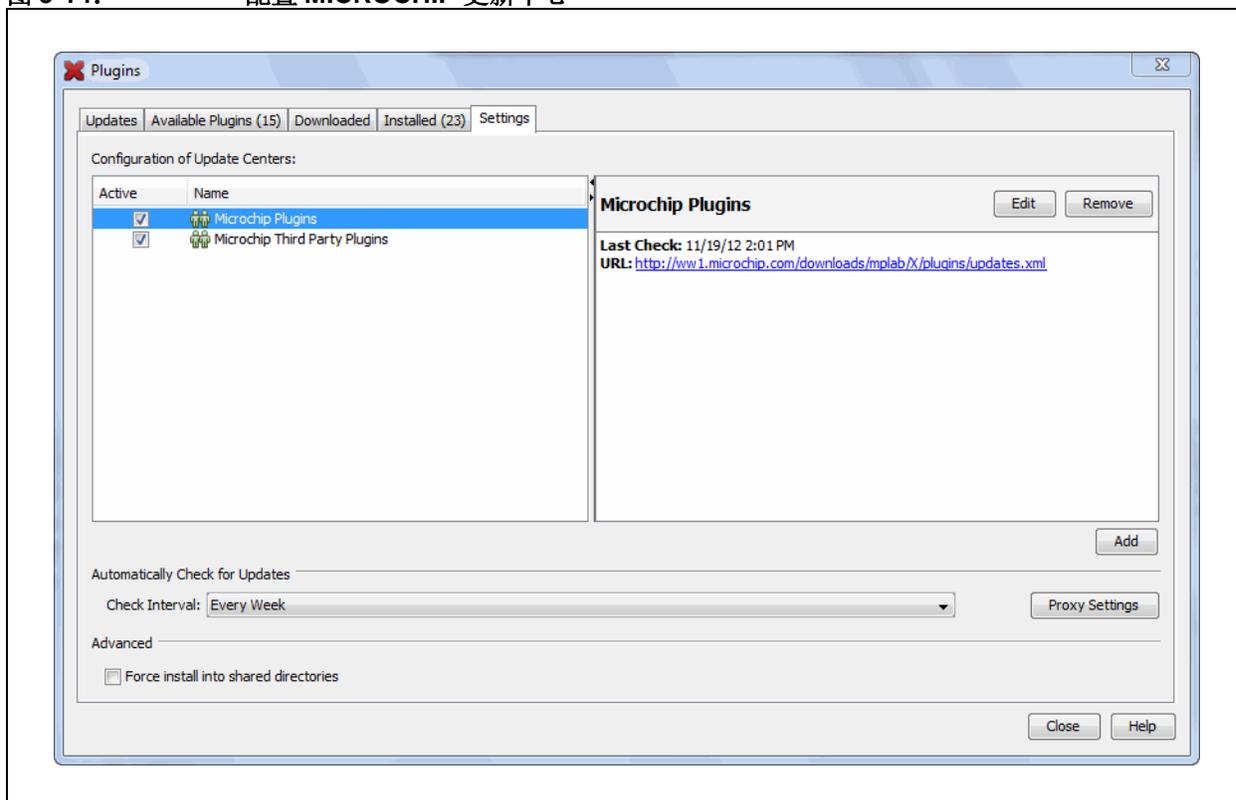
MPLAB X IDE 安装时已配置了两个 Microchip 更新中心：

- <http://ww1.microchip.com/downloads/mplab/X/plugins/updates.xml>
- <http://ww1.microchip.com/downloads/mplab/X/thirdpartyplugins/updates.xml>

要在插件管理器中配置另一个更新中心：

1. 选择 **Tools>Plugins** 并单击 **Settings**（设置）选项卡。
2. 单击 **Add** 按钮，打开 Update Center Customizer（更新中心定制器）对话框。
3. 输入更新中心的名称。
4. 输入更新中心的 URL。
5. 单击 **OK**。

图 5-14: 配置 MICROCHIP 更新中心



5.21.4 插件代码位置

插件代码与 MPLAB X IDE 用户配置数据一起存储。请参见第 8.6 节“查看用户配置数据”。

注:

第 6 章 高级任务

6.1 简介

本章提供关于在 MPLAB X IDE 中执行高级任务的指南。第 4 章“基本任务”和第 5 章“附加任务”介绍了其他功能。

- 加快 MPLAB X IDE 速度
- 处理多个项目
- 处理多个配置
- 创建用户 MakeFile 项目
- 日志数据
- 定制工具条

6.2 加快 MPLAB X IDE 速度

如果 MPLAB X IDE 的工作速度太慢，可以考虑以下方面：

- 增加计算机的堆大小
- 调试工具使用

6.2.1 增加计算机的堆大小

您可以在文件 `mplab_ide.conf` 中修改为 MPLAB X IDE 分配的存储器量。建议您在开始编辑该文件之前先进行备份。如果更改了该文件的内容，所做的更改将在下一次运行 MPLAB X IDE 时生效。

Windows OS 64 位

```
C:\Program Files (x86)\Microchip\MPLABX\mplab_ide\etc
```

Windows OS 32 位

```
C:\Program Files\Microchip\MPLABX\mplab_ide\etc
```

Linux OS

```
/opt/microchip/mplabx/mplab_ide/etc
```

Mac OS X

```
/Applications/microchip/mplabx/Contents/Resources/mplab_ide/etc
```

以下行包含了默认值：

```
default_options="-J-Dnb.FileChooser.useShellFolders=false  
-J-Dcrownking.stream.verbosity=very-quiet -J-Xms256m -J-Xmx512m  
-J-XX:PermSize=128m -J-XX:MaxPermSize=384m -J-XX:+UseConcMarkSweepGC  
-J-XX:+CMSClassUnloadingEnabled"
```

粗体区域为：

-Xms256m 指示 JVM 初始时至少为堆分配 256 MB。

-Xms512m 指示 JVM 最多为堆分配 512 MB，但不超出。

-XX:PermSize=128m 指示 JVM 为跟踪不进入堆中的其他数据所需的内存分配 128 MB。

-XX:MaxPermSize=384m 指示 JVM 为不进入堆中的其他数据最多分配 384 MB。

除非收到错误 `java.lang.OutOfMemoryError: PermGen space`，否则应该不需要修改 `PermSize` 或 `MaxPermSize`。

一般情况下，最重要的方面为 `-Xms512m`；它会限制 MPLAB X IDE 将使用的堆的最大空间量。虽然使用较大的堆可能会有所帮助，但存储器用于 MPLAB X IDE 意味着用于其他应用程序和系统功能的存储器会减少。

您可以通过使能存储器监视器来监视 IDE 使用了多少存储器。右键单击工具条区域中的空白区域并选择存储器，或选择 **View (视图) > Toolbars (工具栏) > Memory (存储器)**。



除非您在 `mplab_ide.conf` 中更改值，否则上界值不会超出 512 MB。建议以 128 MB 为增量更改 `-Xms512m` 设置。如果具有大量的存储空间，则增量可以超出 128 MB，但是需要确保为系统的其余部分留下足够的存储器。

6.2.2 调试工具使用

当使用调试工具导致 MPLAB X IDE 速度下降时：

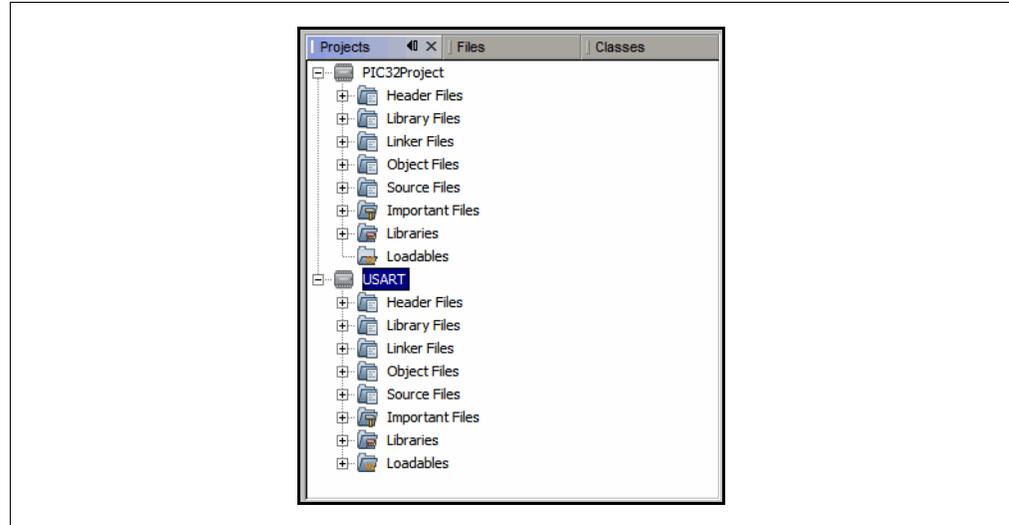
1. 仅显示需要查看的窗口。当使用调试工具时，MPLAB X IDE 调试器会检查每个打开的窗口。
2. 在调试单步运行过程中，减少对堆栈窗口的更新。在 **Tools > Options** 窗口，**Embedded** 按钮，**Generic Settings** 选项卡中，选中“Disable auto refresh for call stack view during debug sessions”（在调试会话期间禁止调用堆栈视图自动刷新）和“On mouse-over structure and array expressions, evaluate integral members only”（鼠标悬停在结构和数组表达式上时，仅计算整数成员的值）。

6.3 处理多个项目

MPLAB X IDE 允许您处理多个项目。

如果需要同时处理多个项目，可以在 MPLAB X IDE 中打开多个项目并在 Projects 窗口中查看。关于该窗口的更多信息，请参见第 12.11 节“Projects 窗口”。

图 6-1: PROJECTS 窗口中的多个项目



活动项目

通过在 Projects 窗口中单击项目，可以将它们设为活动项目。

主项目

一个项目可以通过以下操作之一选择为主项目：

- 右键单击项目名称，然后选择“Set as Main Project”。
- 选择 *Run>Set Main Project*。

然后，主项目名称将以**粗体**显示。

此外，您还可以在不设置活动项目的情况下工作。在这种情况下，IDE 将通过上下文确定您正在处理的项目，即，如果将编辑器焦点设置在某个文件中，则拥有该文件的项目将成为接收操作的项目，例如，当您按下 **Debug Run** 按钮时。是显式地将一个项目设为“主”项目，还是使用编辑器焦点来选择主项目，这可由您选择。

如果您已设置了主项目，并希望将它去除：

1. 在 Projects 视图的空白区域单击鼠标右键。
2. 使用“Set Main Project”更改活动项目或选择无活动项目。

要处理多个项目：

1. 创建 ([File>New Project](#)) 或打开 ([File>Open Project \(打开项目\)](#)) 两个或三个项目。
2. 开始调试每个项目，即，在 **Projects** 窗口中单击某个项目，然后选择 [Debug>Debug Project](#)。
3. 打开 **Sessions** (会话) 窗口 ([Window>Debugging>Sessions](#))，您可以在当前正在运行的任何调试会话之间切换。

从一个调试会话切换到另一个时，**Watches** 窗口和变量以及存储器会切换为显示正在调试的当前选定项目。状态位也会跟随调试项目而变化。**Dashboard** 将跟随最后选定的项目变化，无论它是调试项目还是项目窗口中的项目。这是由设计决定的。

要对多个项目分组：

处理多个项目的另一种方式是进行分组。选择 [File>Project Group \(项目组\)](#)，并选择或创建一个项目组。

关于创建项目组的更多信息，请参见 **NetBeans 帮助主题 Project Group: Create New Group Dialog Box** (项目组：创建新组对话框)。

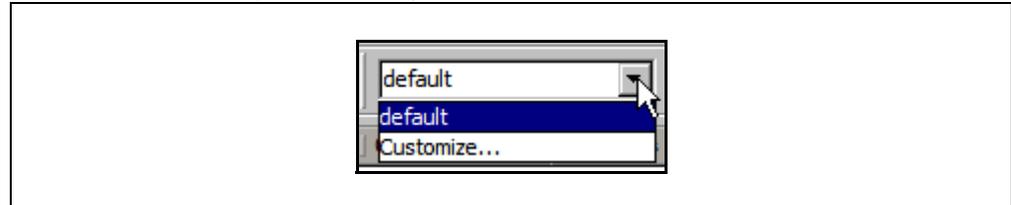
6.4 处理多个配置

MPLAB X IDE 允许对于同一项目使用多个配置。对于可以在多个平台上进行编译的代码（例如 Microchip 应用程序库演示项目），这会很有用。

创建一个新项目时，会创建“default”配置。要创建您自己的配置，可以通过以下操作之一开始：

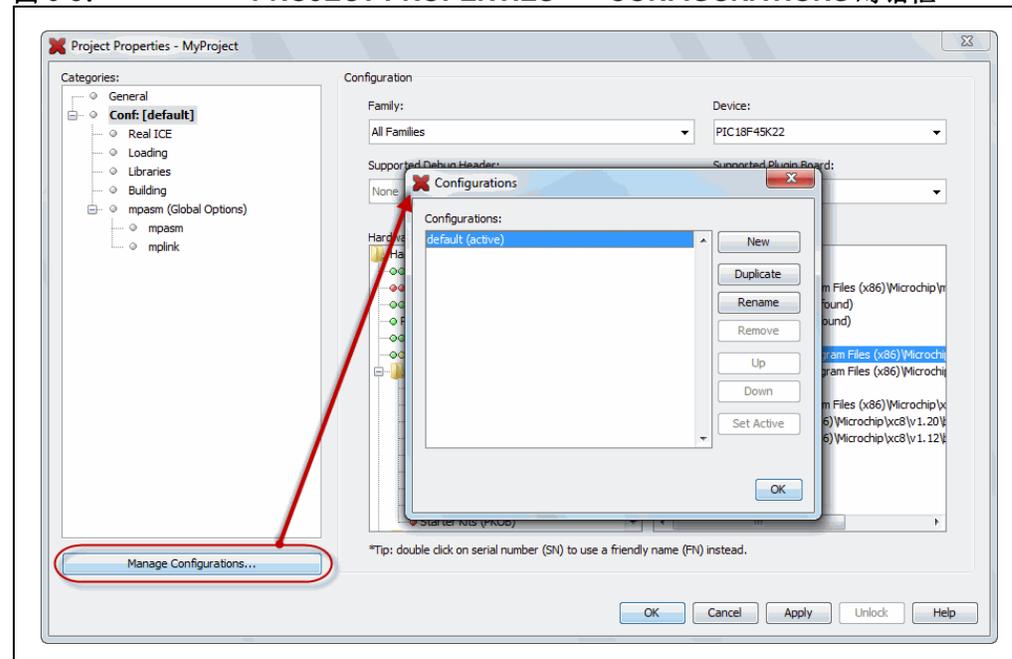
- 使用工具条上的下拉菜单，并选择“Customize”（图 6-2）。此时会打开 Project Properties 对话框。
- 通过右键单击项目名称并选择“Properties”，打开 Project Properties 对话框。

图 6-2: 项目配置下拉框



在 Project Properties 对话框中，单击 **Manage Configurations**（管理配置）打开 Configurations 对话框。可以重命名现有配置，也可以添加或从现有配置复制新的配置。为项目创建多个配置时，可以从 **Manage Configurations** 或从下拉菜单中选择活动配置。

图 6-3: PROJECT PROPERTIES—CONFIGURATIONS 对话框

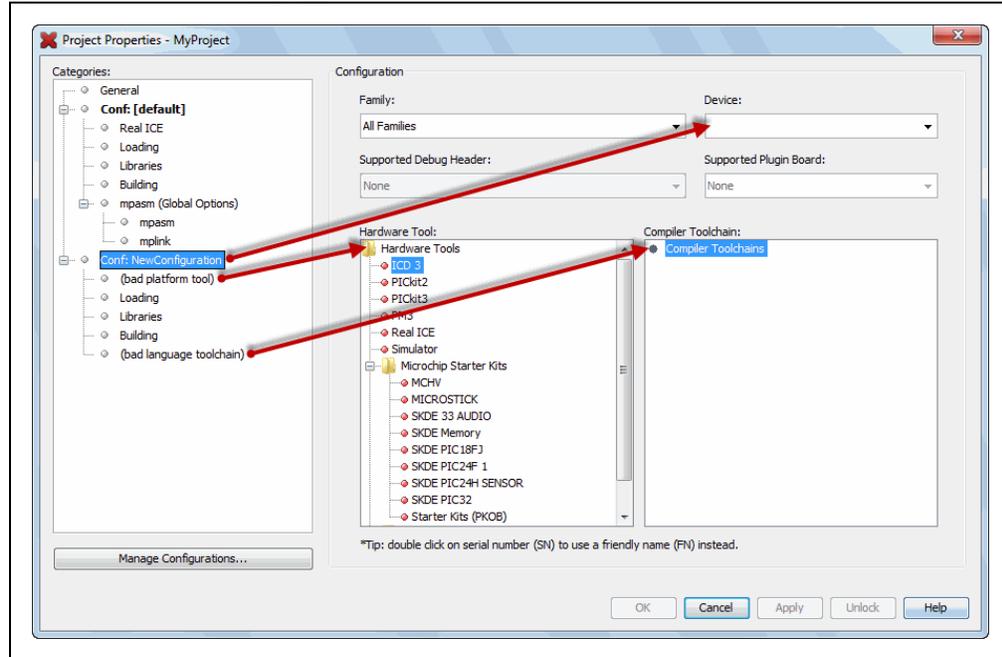


6.4.1 添加新配置

如果添加了一个新配置，则必须在 **Project Properties** 对话框中指定一些项。

- 器件 —— 必须先选择此项，才能看到硬件工具和编译器支持。
- 硬件工具
- 编译器工具链

图 6-4: 新配置



6.4.2 添加副本配置

您可以添加一个现有配置的副本来作为新配置，然后基于它进行编辑。

副本配置的一个很好用途是用于创建您自己的调试配置。虽然 MPLAB X IDE 提供了用于 Microchip 工具的调试宏（第 4.16.2 节“生成的调试宏”），但您可能希望使用自己的调试宏或希望使用第三方工具设置相同的调试功能。

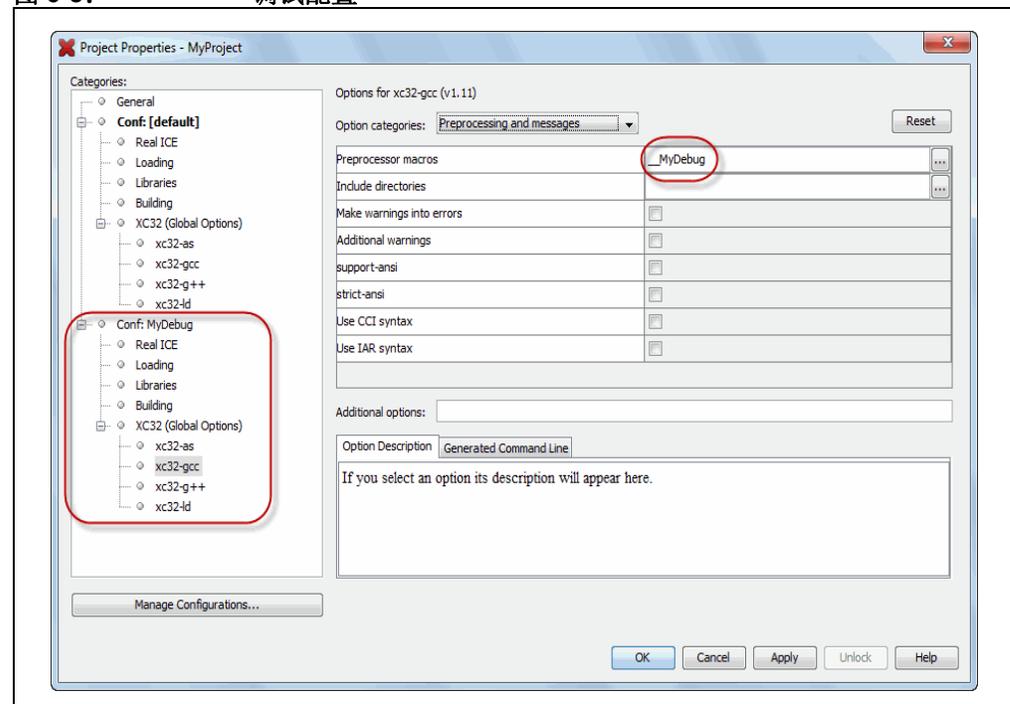
要设置调试配置：

1. 在 Configurations 对话框中，选择项目配置，然后单击 **Duplicate**（复制副本）。
2. 单击 **Rename**（重命名），然后在 New Configuration Name（新配置名称）对话框中输入名称，例如“MyDebug”。
3. 单击 **OK** 两次以返回到 Project Properties 对话框。现在可以看到调试配置（Conf:MyDebug）。
4. 单击工具链中的编译器。在“Preprocessing and messages”（预处理和消息）选项类别下，找到用于定义宏的选项，然后单击关联的文本框。
5. 在弹出对话框中，输入宏名称（如 __MyDebug），并单击 **OK**。

现在，可以在需要调试时切换为调试配置。您可以在条件文本中使用预处理器宏：

```
#ifdef __MyDebug
    fprintf(stderr, "This is a debugging message\n");
#endif
```

图 6-5: 调试配置



6.5 创建用户 MAKEFILE 项目

创建一个使用外部 makefile 的项目。如果您具有一个在 MPLAB X IDE 之外编译的现有项目，但现在希望使用 MPLAB X IDE 进行调试，则这会很有用。

要创建 makefile 项目，请使用 New Project 向导。

6.5.1 新项目设置

要打开 New Project 向导，请执行以下操作之一：

- 在 **Start Page** 中，单击 **Learn & Discover** 选项卡 “Dive In” 部分的 “Create New Project” 链接。
- **File (文件) > New Project (新建项目)** (或 Ctrl+Shift+N)

向导将启动，指导您完成新项目设置。

- **步骤 1. 选择项目：**选择 “Microchip Embedded” 类别，然后选择项目类型 “User Makefile Project” (用户 makefile 项目)。
- **步骤 2. 选择器件：**从 “Device” 下拉列表中选择应用中使用的器件。要缩小选择列表，请先选择 “Family”。
- **步骤 3. 选择仿真头：**如果存在可用于选定器件的仿真头，则会出现该步骤。要确定调试是否需要仿真头，或器件是否具有片内调试电路，请参见 *Processor Extension Pak and Debug Header Specification* (DS51292 或联机帮助)。然后选择是否使用仿真头。
- **步骤 3 或 4. 选择工具：**从列表中选择将用于调试应用程序的开发工具。
工具名称旁边将显示所选器件的工具支持级别。绿色表示完全支持，黄色表示 beta 支持，红色表示尚不支持。
- **步骤 4 或 5. 创建用户 makefile 项目：**输入信息来设置您的 makefile 项目。详情请参见第 6.5.2 节 “用户 makefile 项目设置”。
- **步骤 5 或 6. 选择项目名称和文件夹：**选择新项目的名称和位置。您可以浏览至一个位置。

新项目将在 Projects 窗口中打开。

关于将项目导出为十六进制文件的信息，请参见第 12.11.2 节 “Projects 窗口 —— 项目菜单”。

6.5.2 用户 makefile 项目设置

在 New Project 向导的 “User Makefile Project Settings” (用户 makefile 项目设置) 窗口中选择 makefile 项目设置。在创建项目之后，在 Project Properties 窗口的 “Makefile” 类别中更改 makefile 项目设置。每个窗口具有相同的项。

表 6-1: 用户 MAKEFILE 项目设置

项	说明
Working Directory (工作目录)	指定外部项目的位置。这是编译、调试编译和清除命令的默认位置。
Build command (编译命令)	当在 MPLAB X IDE (图标或菜单) 中请求对应于 Run 的命令时，改为根据该命令执行编译。
Debug build command (调试编译命令)	当在 MPLAB X IDE (图标或菜单) 中请求对应于 Debug Run 的命令时，改为根据该命令执行调试编译。
Clean command (清除命令)	当在 MPLAB X IDE (图标或菜单) 中请求对应于 Clean 的命令时，改为根据该命令执行清除。
Image name (映像名称)	输入生产映像文件 (十六进制) 的路径和名称。
Debug image name (调试映像名称)	输入调试映像文件 (elf 或 cof) 的路径和名称。

数据输入说明:

- 建议在路径中使用正斜杠 (/)。但是，您可以改用反斜杠 (\)。
- 建议使用不带空格的路径名和文件名。但如果具有空格，则当您在窗口中输入路径名和文件名时，请使用转义空格 (空格之前加上反斜杠)。已知 GNU make 会对于空格产生一些奇怪的问题，因为空格是它们的分隔符。
- 工作目录: 建议通过浏览来选择目录，因为它会为您处理格式问题。
- 命令路径 (“Build command”、“Debug build command”和 “Clean command”):
 - 可以是绝对路径或相对于工作目录的相对路径。
 - 对于 Windows 操作系统 (OS)，需要包含在引号 (“) 中。
 - 对于其他操作系统，不需要包含在引号中，但需要对所有空格进行转义。
- 映像路径 (“Image name”和 “Debug image name”)
 - 必须为绝对路径。
 - 需要对所有空格进行转义。

6.6 日志数据

日志文件可用于捕捉程序执行和调试中出现的问题。

当您遇到 MPLAB X IDE 错误或问题，需要与技术支持联系（见“Customer Support”）时，应将数据捕捉到两个日志文件中：MPLAB X IDE 日志文件和 NetBeans 平台日志文件。

6.6.1 MPLAB® X IDE 日志文件

MPLAB X IDE 日志文件基于 Java Logger 类来生成数据。

要设置日志文件：

1. 选择 **Tools>Options**（对于 Mac OS X 为 **mplab_ide>Preferences**），**Embedded** 按钮，**Diagnostics**（诊断）选项卡。
2. 从下拉框中选择一个日志记录级别。详情请参见第 12.12.5 节“Diagnostics 选项卡”。

注： 日志记录级别越高，收集的数据就越多，但应用程序的运行速度就越慢。

3. 选择日志文件的位置（路径）。

要记录数据日志：

1. 使用日志记录级别“Finest”设置日志文件。
2. 记下日志文件的名称和位置。
3. 重复会导致错误或问题的步骤。
4. 找到日志文件并发送给技术支持。

6.6.2 NetBeans 平台日志文件

NetBeans 日志文件会生成关于执行 NetBeans 平台的信息。

要记录数据日志：

5. 通过选择 **View (查看) >IDE log (IDE 日志)** 打开日志文件 Output 窗口。
6. 在窗口中单击右键并选择“Clear”（清除）。
7. 重复会导致错误或问题的步骤。
8. 在窗口中单击右键并选择“Save As”（另存为）将文本保存到一个文件中。
9. 将该文件发送给技术支持。

6.7 定制工具条

您可以使用 **Customize Toolbars**（定制工具条）窗口定制 MPLAB X IDE 工具条。选择 **View>Toolbars>Customize** 打开窗口。

可用图标包括 **Clean Only**（仅清除）、**Run** 和 **Set PC to Cursor**（将 PC 设置为光标位置）等。

向工具条中添加功能：

将一个图标从 **Customize Toolbars** 窗口拖动到工具条上。

从工具条中删除功能：

将一个图标从工具条拖动到 **Customize Toolbars** 窗口上。

添加您自己的工具条：

单击“**New Toolbar**”（新建工具条），并命名新的工具条。

更改工具条图标的大小：

- 选中复选框“**Small Toolbar Icons**”（小工具条图标）可使图标变小。
- 取消选中可使图标变大。

还原为默认工具条：

单击“**Reset Toolbars**”（复位工具条）。

下表列出了可用的功能。

表 6-2: BUILD 功能

功能	详细信息
Make	不进行清除而执行 make（如果项目文件发生了更新，则编译项目文件）
Make Clean (Make 清除)	清除（删除先前编译的项目文件），并对项目执行 make

表 6-3: DEBUG 功能

功能	详细信息
New Data Breakpoint (新建数据断点)	在地址处设置新的数据断点
New Run Time Watch (新建运行时观察)	添加指定符号来观察将在程序运行 / 执行时改变值的符号
New Watch	在 Watches 窗口中输入一个表达式或选择要观察的 SFR
PIC AppIO	打开 PIC AppIO 窗口。您的调试工具和器件必须支持应用程序 I/O 关于详细信息，请参见调试工具文档。
Program Device for Debugging Main Project (为调试主项目而对器件 编程)	通过调试映像对器件编程 程序将在编程完成后立即开始执行。
Disassembly	打开 Disassembly 窗口
Disconnect from Debug Tool (与调试工具断开)	断开 MPLAB [®] X IDE 和调试工具之间的通信 要重新连接，请选择 Run/Debug Run。
Focus Cursor at PC (将 光标焦点设置在 PC 处)	将光标移动到当前 PC 地址处，并使该地址处于窗口的中心
PC Profiling (PC 性能 分析)	打开 PC Sampling (PC 采样) 窗口 关于 PC 采样和性能分析的更多信息，请参见 MPLAB REAL ICE 在 线仿真器文档。
Reset	将器件复位
Set PC at Cursor (将 PC 设置在光标位置)	将程序计数器 (PC) 值设置为光标的行地址

表 6-3: DEBUG 功能 (续)

功能	详细信息
Step Instruction (单步执行指令)	执行一条机器指令。如果该指令是一个函数调用, 则执行函数, 并将控制返回给调用方
Status Toolbar Action (状态工具条操作)	显示状态工具条
Launch Debugger Main Project (为主项目启动调试器)	为主项目启动调试器 这是分离式编译过程的最后一个步骤: 先 Build , 再 Program Target (对目标编程), 最后 Launch Debugger 。这对于在调试和使用入门工具包期间更改存储器窗口设置很有用。
Debug Main Project (调试主项目)	调试主项目
New Watch	在 Watches 窗口中输入要观察的表达式
Attach Debugger (连接调试器)	建立 MPLAB® X IDE 和调试工具之间的通信 调试工具在运行或调试运行期间会自动连接, 并在运行结束时断开连接。要使调试工具总是保持连接, 请选中 Tools>Options (对于 Mac OS X 为 mplab_ide>Preferences), Embedded 按钮, General Settings 选项卡, “Maintain active connection to hardware tool”。
Continue	“Pause” 之后继续调试, 直到达到下一个断点或程序末尾。
Debug File (调试文件)	启动当前选定文件的调试会话。
Debug Test File (调试测试文件)	在 JUnit 中启动文件的调试测试。(Java 相关)
Apply Code Changes (应用代码更改)	将代码中的所有更改应用于正在执行的程序。
Finish Debugger Session	结束调试会话。
Make Callee Current (将被调用方设为当前调用)	将被调用的方法设为当前调用 仅当在 Call Stack 窗口中选定某个调用时可用。
Make Caller Current (将调用方设为当前调用)	将产生调用的方法设为当前调用 仅当在 Call Stack 窗口中选定某个调用时可用。
Pause	暂停调试 使用 “Continue” 可继续。
Run to Cursor	运行当前项目, 直到文件中的光标位置, 并停止程序执行
Step Into	执行程序的一个源代码行 如果该行是一个函数调用, 则程序执行到该函数的第一条语句, 然后停止。
Step Over Expression (单步跳过表达式)	单步跳过表达式, 然后停止调试。
Step Out	执行程序的一个源代码行 如果该行是一个函数调用, 则执行函数, 并将控制返回给调用方。
Step Over	执行程序的一个源代码行 如果该行是一个函数调用, 则执行整个函数, 然后停止。

表 6-4: EDIT 功能

功能	详细信息
Next Bookmark (下一个书签)	循环向前浏览书签
Previous Bookmark (上一个书签)	循环向后浏览书签
Clear Document Bookmarks (清除文档书签)	清除文档中的所有书签
Quick Search (快速搜索)	显示 Quick Search 工具条
Find in Projects (在项目中查找)	在项目内查找指定的文本、对象名称和对象类型
Replace in Projects (在项目中替换)	在项目内替换文本、对象名称和对象类型
Copy (复制)	将当前选定内容复制到剪贴板上
Cut (剪切)	删除当前选定内容并将其放置到剪贴板上
Delete	删除当前选定内容
Find (查找)	查找文本字符串
Paste (粘贴)	将剪贴板的内容粘贴到插入点
Redo (重做)	回退 (一次一个) 一系列 Undo (撤消) 命令
Undo	回退 (一次一个) 除 Save (保存) 之外的一系列编辑器操作

表 6-5: HELP 功能

功能	详细信息
Start Page	显示起始页。
Individual Help Files (独立帮助文件)	显示独立的弹出帮助文件, 例如, MPLAB [®] X IDE 帮助。
Help	显示 Help 窗口。

表 6-6: PROFILE 功能

功能	详细信息
Start Sampling IDE (开始采样 IDE)	开始采样, 来进行 PC 性能分析 / 采样 关于 PC 采样和性能分析的更多信息, 请参见 MPLAB [®] REAL ICE 在线仿真器文档。

表 6-7: PROJECT 功能

功能	详细信息
Programmer to Go PICkit™ 3 Main Project (使用 PICkit 3 对主项目进行脱机编程)	使用 PICkit 3 的脱机编程功能
Erase Device Memory Main Project (擦除主项目器件存储器)	擦除主项目的器件存储器
Hold in Reset	使器件在 Reset 和 Run 之间切换
Make and Program Device Main Project (对器件主项目执行 make 并编程)	对主项目进行编译 (如需要) 并对器件进行编程。程序将在编程完成后立即开始执行。
Read Device Memory Main Project (读取主项目器件存储器)	将目标存储器中的内容传输到 MPLAB® X IDE
Read Device Memory to File (将器件存储器读取至文件)	将目标存储器的内容传输到指定文件中
Build Main Project (编译主项目)	编译主项目的所有文件
Build for Debugging Main Project (为调试主项目而进行编译)	编译主项目的所有文件以进行调试
Clean and Build Main Project (清除并编译主项目)	清除 (删除) 所有已生成的文件, 然后重新编译主项目的所有文件
Build Project (编译项目)	编译选定项目的所有文件
Clean Main Project (清除主项目)	清除 (删除) 主项目的所有已生成文件
Clean Project (清除项目)	清除 (删除) 选定项目的所有已生成文件
Compile File (编译文件)	使用项目编译器编译选定的文件
New File (新建文件)	启动 New File 向导
New Project	启动 New Project 向导
Open Project	打开现有项目
Clean and Build Project (清除并编译项目)	清除 (删除) 所有已生成的文件, 然后重新编译选定项目的所有文件
Run Main Project (运行主项目)	对主项目进行编译 (如需要) 并对器件进行编程。程序将在编程完成后立即开始执行。 等同于 “Make and Program Device Main Project”。
Run Project	对选定项目进行编译 (如需要) 并对器件进行编程。程序将在编程完成后立即开始执行。
Run File (运行文件)	运行当前选定的文件
Test Project (测试项目)	启动当前项目的 JUnit 测试 (Java 相关)
Test File (测试文件)	启动当前文件的 JUnit 测试 (Java 相关)
Open Required Projects (打开所需的项目)	打开依赖的项目
Projects	打开 Projects 窗口
Files	打开 Files 窗口

表 6-8: CVS* 功能

功能	详细信息
Commit (提交)	将文件的本地更改提交到存储库中
Diff	显示存储库版本与本地工作副本之间的文件修订
Show Annotations (显示标注)	在源代码编辑器中, 在已打开文件的左边缘显示提交消息、作者、日期和版本号信息
Revert Modifications (还原修改)	将文件的本地版本还原为存储库中存储的版本
Update (更新)	使用已提交到存储库中的更改来更新文件的本地版本

* CVS 不随 MPLAB® X IDE 提供, 必须单独获取。关于 CVS 的更多信息, 请参见 <http://www.nongnu.org/cvs>。

表 6-9: SYSTEM 功能

功能	详细信息
Next Error (下一个错误)	将源代码编辑器滚动到包含下一个编译错误的行
Previous Error (上一个错误)	将源代码编辑器滚动到包含上一个编译错误的行
Run SQL (运行 SQL)	运行 SQL 语句和脚本 关于 SQL 和数据库支持的更多信息, 请参见: NetBeans 帮助 > IDE Basics > Servers and Databases (服务器和数据库)。
SQL History (SQL 历史记录)	提供先前在 SQL Editor (SQL 编辑器) 中执行的 SQL 语句的列表
Keep Prior Tabs (保留先前选项卡)	进行翻转, 使来自先前执行的 SQL 结果选项卡保持打开 / 关闭
Open File	打开 Open (打开) 对话框
Properties	打开 Project Properties 窗口
Save	保存当前文件
Save All (全部保存)	保存所有打开的文件。如果选择了“Compile on Save”(保存时编译)功能, 则这也将编译项目文件。

表 6-10: TERMINAL 功能

功能	详细信息
org-netbeans-modules-dlight-terminal-action-LocalTerminal(→)	打开 Terminal (终端) 窗口
org-netbeans-modules-dlight-terminal-action-RemoteTerminal(→)	打开 Remote Terminal (远程终端) 窗口

表 6-11: TOOLS 功能

功能	详细信息
Check CVS (检查 CVS)	选择 CVS 版本控制项 关于 CVS 的更多信息, 请参见 http://www.nongnu.org/cvs

表 6-12: VIEW 功能

功能	详细信息
Web Browser (网络浏览器)	在默认浏览器中打开 NetBeans 下载页面
IDE Log (IDE 日志)	显示与 View>IDE Log 相同的信息
Classes (类)	打开 Classes 窗口
Customize Zoom (定制缩放显示)	定制数据的缩放显示
Zoom In (放大显示)	放大数据显示
Zoom Out (缩小显示)	缩小数据显示

表 6-13: WINDOW 功能

功能	详细信息
Debug>Breakpoints	打开 Breakpoints 窗口
Debug>Call Stack	打开 Call Stack 窗口
Debug>Variables	打开 Variables 窗口
Debug>Sessions	打开 Sessions 窗口
Debug>Sources (源)	打开 Sources 窗口
Debug>Threads (线程)	打开 Threads 窗口 该窗口会列出当前调试会话中的所有线程
Debug>Watches	打开 Watches 窗口
Select Document In (选择文档) > Select In Favorites (选择收藏夹中的文档)	打开 Favorites (收藏夹) 窗口, 并选择其中的当前文档
Terminal	打开 Terminal 窗口
Remote Terminal	打开 Remote Terminal 窗口
Terminal (Experimental) (终端 (实验))	打开 Experimental Terminal (实验终端) 窗口
Stopwatch	打开 Stopwatch 窗口
Trace (跟踪)	打开 Trace 窗口
Disassembly Listing File	在编辑器窗口中打开反汇编列表文件
Project Environment (项目环境)	打开项目环境 (Dashboard) 窗口
Analyzer (分析器)	打开 Simulator Analyzer (模拟器分析器) 窗口
Stimulus (激励)	打开 Simulator Stimulus (模拟器激励) 窗口
Services	打开 Services 窗口
Output	打开 Output 窗口
Properties	打开 Project Properties 窗口
Call Graph	打开 Call Graph 窗口
Hierarchy View (层次结构视图)	打开 Include Hierarchy (包含层次结构) 窗口 这使您可以检查所有直接或间接包含在源文件中的头文件和源文件, 或所有直接或间接包含某个头文件的源文件和头文件
Macro Expansion View (宏扩展视图)	打开 Macro Expansion (宏扩展) 窗口 该窗口会展开代码中的宏
Thread Map (线程映射)	打开 Threads 窗口
Reporter Result (报告器结果)	打开 Exception Reporter (异常报告器) 窗口, 其中列出异常断点
Favorites	打开 Favorites 窗口。
Test Results (测试结果)	打开 C/C++ 项目的单元 Test Results 窗口
Chat	打开 Team Chat (团队聊天) 窗口
Team	打开团队项目。关于团队服务器项目的更多信息, 请参见 NetBeans 帮助” IDE Basics>Collaborative Development”。
Navigator	打开 Navigator 窗口

表 6-13: WINDOW 功能 (续)

功能	详细信息
Palette (调色板)	打开 Palette 窗口 (Java 相关)
Find Usages Results (查找使用实例结果)	打开 Find Usages (查找使用实例) 窗口
Refactoring Preview (重构预览)	打开重构结果的 Preview (预览) 窗口
Search Results (搜索结果)	打开 Search 窗口
Tasks	打开 Tasks 窗口

表 6-14: XML 功能

功能	详细信息
Check File (检查文件)	检查 XML 文件的格式正确性
Check DTD (检查 DTD)	打开 DTD 和 XML Schemas Manager (XML 模式管理器)
Validate File (验证文件)	根据 XML 模式或 DTD 检查 XML 文件中的语法错误
XSL Transform (XSL 转换)	使用 XSL 样式表转换 XML 文档

注:

第 7 章 编辑器

7.1 简介

MPLAB X IDE 是基于 NetBeans 平台而构建的，提供了一个内置编辑器来创建新代码或更改现有代码。

NetBeans 帮助主题 [IDE Basics \(IDE 基础知识\) > Basic File Features](#) 下提供了关于该编辑器的一般信息。NetBeans 帮助主题 [C/C++/Fortran Development > Working with C/C++/Fortran Projects > Editing Source Files in C/C++/Fortran Projects \(编辑 C/C++/Fortran 项目中的源文件\) > About Editing C and C++ Files \(关于编辑 C 和 C++ 文件\)](#) 下提供了关于编辑器的 C 编译器信息。下面介绍 MPLAB X IDE 编辑器的功能。

- 编辑器用法
- 编辑器选项
- 代码折叠
- C 代码重构

7.2 编辑器用法

要使用编辑器，可以分别使用 [File > New File](#) 或 [File > Open File](#) 创建文件或打开现有文件。以下几节介绍了附加的控件和功能。

7.2.1 桌面控件

以下桌面项与编辑器关联：

- File 菜单（见第 11.2.1 节“File 菜单”），用以在 Editor（编辑器）窗口中打开文件。
- Edit 菜单（见第 11.2.2 节“编辑菜单”），用以使用编辑命令。
- 编辑器工具条，位于每个文件的编辑器窗口顶部（见第 11.3.9 节“Editor 工具条”），用以访问一些编辑命令。
- 窗口右键（上下文）菜单，用以访问更多的命令。

7.2.2 C 代码中的超链接

超链接导航功能使您可以从函数、变量或常量的调用跳至其声明。

要使用超链接，请执行以下操作之一：

- 将鼠标悬停在某个函数、变量或常量上，同时按下 <Ctrl> 键（Windows 和 Linux）或 <Command> 键（Mac）。此时会显示一个超链接，以及包含有关元素信息的工具提示。单击该超链接，编辑器会跳至声明。按下 <Alt>+<向左箭头> 可跳回到调用。
- 将鼠标悬停在某个标识符上，并按下 <Ctrl>+ 或 <Command>+，编辑器会跳至声明。按下 <Alt>+<向左箭头> 可跳回到调用。

按下 <Alt>+<向左箭头> 和 <Alt>+<向右箭头> 可向后和向前遍历光标位置的历史记录。

此外可参见 NetBeans 帮助主题：

[C/C++/Fortran Development>Working with C/C++/Fortran Projects>Navigating Source Files and Projects>Using Hyperlink Navigation](#) (使用超链接导航)

7.2.3 ASM 代码中的超链接

要导航到汇编源文件中包含的头文件，可以按下 <Ctrl> 键 (Windows 和 Linux) 或 <Command> 键 (Mac)，同时将鼠标光标放置在由 #include 语句引用的文件名上。然后单击鼠标选择按钮，在编辑器中在包含文件自己的文件选项卡中打开包含文件。

7.2.4 编辑器功能说明

下表总结了编辑器的一些更常用的功能。

表 7-1: 编辑器功能

编辑器功能	参考
支持 Unicode。	默认情况下，在 IDE 中新创建的项目将使用 ISO-8859-1 字符编码。要更改该设置： <ul style="list-style-type: none"> 在 Projects 窗口中右键单击项目名称并选择 Properties。 在左侧栏的“Categories”下，选择“General”。 在页面底部，查找“Encoding”并进行更改。
代码基于语法进行着色。	Tools>Options (对于 Mac OS X 为 <i>mplab_ide>Preferences</i>)， Fonts and Colors 按钮， Syntax (语法) 选项卡。 基于项目创建过程中设置的编码。
在输入代码时标记错误。	<i>C/C++/Fortran Development>Working with C/C++/Fortran Projects>Editing Source Files in C/C++/Fortran Projects>Error Highlighting</i> (错误高亮显示)
着色标记可用于快速访问多个符号和错误等。	Tools>Options (对于 Mac OS X 为 <i>mplab_ide>Preferences</i>)， Fonts and Colors 按钮， Annotations (标注) 选项卡。
智能代码补全可提出建议和提供提示。	Tools>Options (对于 Mac OS X 为 <i>mplab_ide>Preferences</i>)， Editor 按钮， Code Completion 选项卡。
可以折叠和展开汇编及 C 代码	第 7.4 节 “代码折叠”
右键单击函数 (delay(x)) 可查找使用实例。这可以限制在函数内进行查找 (例如，局部 i 变量)。	Find Usages 对话框
右键单击函数 (delay(x)) 可显示调用图 调用图一侧具有一些按钮，可用于切换顺序等。	第 5.16 节 “查看调用图”
在源代码中使用任务关键字 (示例：//TODO) 创建注释，将会自动对任务进行扫描并添加到 Tasks 窗口中。	<i>IDE Basics>Basic File Features>Working with Task Lists</i> (使用任务列表)
即使没有版本控制系统，也可以使用文件历史记录查看最近的更改并进行还原。	第 5.19 节 “控制源代码” —— 本地历史记录
导航通过一些功能项进行简化，例如“Go to file” (转至文件)、“Go to type” (转至类型)、“Go to symbol” (转至符号)、“Go to header” (转至头文件) 和“Go to declaration” (转至声明)。	第 11.2.4 节 “Navigate 菜单”
用于改善代码结构的重构选项 (重命名函数和变量以及查找所有函数等)。	第 7.5 节 “C 代码重构”

7.3 编辑器选项

要设置编辑器选项：

1. 选择 **Tools>Options** (对于 Mac OS X 为 **mplab_ide>Preferences**) 打开 Options 对话框。
2. 单击 **Editor** 按钮。然后单击选项卡来设置编辑器功能。

下面列出了每个选项卡及其选项。

表 7-2: GENERAL 选项卡

项	说明
Code Folding (代码折叠)	选中时可启用代码折叠，并选择要折叠的代码类型。 关于 Code Folding 的更多信息，请参见第 7.4 节“代码折叠”。
Camel Case Behavior (驼峰大小写行为)	选中时可启用驼峰大小写导航。

表 7-3: FORMATTING (格式设置) 选项卡

项	说明
Language (语言)	选择将应用格式设置的编程语言。
Category (类别)	选择一个类别，当前只有“Tabs and Indents”(制表符和缩进)。
Expand Tabs to Spaces (将制表符扩展为空格)	选中时可制表符转换为空格。
Number of Spaces per Indent (每个缩进的空格数)	输入每个缩进的等效空格数。
Tab Size (制表符大小)	输入制表符的大小。
Right Margin (右边距)	输入右边距的大小。

表 7-4: CODE COMPLETION (代码补全) 选项卡

项	说明
Language	选择将应用代码补全的编程语言。
Completion options (补全选项)	选中时可启用代码补全选项。关于代码补全的更多信息，请参见 NetBeans 帮助主题 C/C++/Fortran Development>Working with C/C++/Fortran Projects>Editing Source Files in C/C++/Fortran Projects>Using Code Completion (使用代码补全)

表 7-5: CODE TEMPLATES 选项卡

项	说明
Language	选择将应用模板的编程语言。
Templates	输入上面指定的语言的模板信息。 Abbreviation (缩写): 输入要在编辑器中输入的缩写。 Expanded Text (展开的文本): 在输入缩写和“Expand template on”(模板展开条件)字符之后，在编辑器中将缩写展开为该文本。 Description (说明): 添加模板项的可选说明。 关于代码模板的更多信息，请参见 NetBeans 帮助主题 C/C++/Fortran Development>Working with C/C++/Fortran Projects>Editing Source Files in C/C++/Fortran Projects>Using Code Templates (使用代码模板)
Expand template on	选择要在编辑器中输入什么字符以将缩写文本展开为展开文本。

表 7-6: HINTS (提示) 选项卡

项	说明
Language	选择将应用提示的编程语言。
Hint Window (提示窗口)	在 Hint 窗口中, 选择一个事件并指定在发生该事件时希望接收到的“提示”: Error (错误)、Warning (警告) 或 Warning on current line (在当前行上的警告)。

表 7-7: MACROS (宏) 选项卡

项	说明
Macros	添加、删除和定义编辑器宏。 <ul style="list-style-type: none">要创建新的宏, 请单击 New (新建) 按钮, 输入新宏的名称, 然后从列表中选择新宏并在 Macro Code (宏代码) 编辑器中输入代码。要设置宏的键盘快捷键, 可以从列表中选择宏, 单击 Set Shortcut (设置快捷键) 按钮并在对话框中输入快捷键。使用该快捷键方式可运行您的宏。要删除宏, 可以从列表中选择宏并单击 Remove (删除) 按钮。要修改宏代码, 可以从列表中选择宏, 并在 Macro Code 编辑器中编辑代码。
Macro Code (宏代码)	单击上面的宏名称, 然后输入宏代码。 关于宏关键字的列表, 请参见: https://netbeans.org/kb/docs/ide/macro-keywords.html 注: 通常, 通过录制宏来添加新宏会比在 Macro Code 编辑器中手动添加更简便。请使用 <i>Edit>Start/Stop Macro Recording</i> (开始/停止宏录制)。

7.4 代码折叠

代码折叠使您可以隐藏代码的某些部分，从而可以专注于其他部分。可折叠（隐藏）或展开（显示）的 C 或汇编代码部分取决于您选择的选项。

- 代码折叠用法
- 定制代码折叠
- 代码折叠 —— 嵌入汇编和 MPLAB C18 C 编译器
- 展开 / 折叠代码定制折叠块

7.4.1 代码折叠用法

默认情况下会使能代码折叠。对于大多数 C 或汇编代码，可折叠的代码部分通过代码左侧的“-”和“+”图标来指示。

- 使能并设置代码折叠
- 展开 / 折叠代码折叠块
- 代码折叠 —— 嵌入汇编和 MPLAB C18 C 编译器

7.4.1.1 使能并设置代码折叠

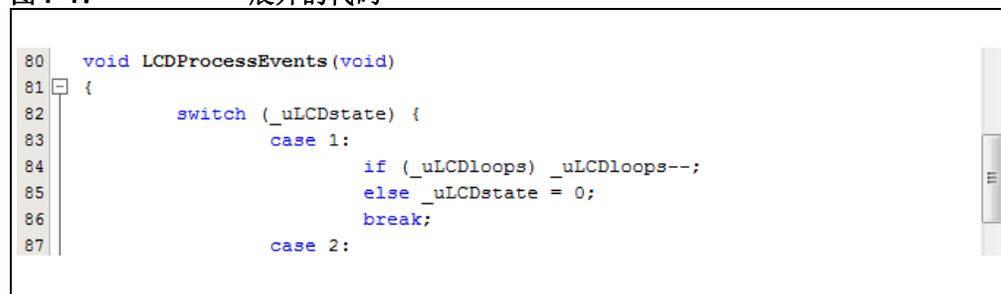
使用以下步骤来使能并设置代码折叠：

1. 选择 **Tools>Options**（对于 Mac OS X 为 **mplab_ide>Preferences**）打开 Options 对话框。
2. 单击 **Editor** 按钮。
3. 单击 **General** 选项卡，并选中“Use code folding”（使用代码折叠）。
4. 选中其他选项来指定要折叠的部分。如果发现其中未列出要折叠的代码的类型，可以进行定制折叠。

7.4.1.2 展开 / 折叠代码折叠块

在编辑器中，单击某个方法旁边的“-”图标，它会发生折叠。单击已折叠方法旁边的“+”图标，它会发生展开。折叠的代码通过省略号框来表示。将光标停留在省略号框上时，IDE 会显示折叠的方法。

图 7-1: 展开的代码



```
80 void LCDProcessEvents(void)
81 {
82     switch (_uLCDstate) {
83         case 1:
84             if (_uLCDloops) _uLCDloops--;
85             else _uLCDstate = 0;
86             break;
87         case 2:
```

图 7-2: 折叠的代码



```
80 void LCDProcessEvents(void)
81 {...}
151 /*****
152  * EOF
153  *****/
```

7.4.1.3 代码折叠 —— 嵌入汇编和 MPLAB C18 C 编译器

对于在 MPLAB C18 C 编译器中采用 `_asm` 和 `_endasm` 伪指令嵌入的汇编代码块不能进行代码折叠。变通方法是此类代码块放到代码 / 文件的末尾附近。

7.4.2 定制代码折叠

要对代码块进行折叠，可以使用定制代码折叠。

- 定制代码折叠 —— C
- 定制代码折叠 —— 汇编
- 展开 / 折叠代码定制折叠块

7.4.2.1 定制代码折叠 —— C

要对 C 代码进行定制折叠，请执行以下操作：

- 在代码前后输入以下注释：

```
// <editor-fold defaultstate="collapsed" desc="user-description">  
要折叠的 C 代码块  
// </editor-fold>
```

或

- 输入 `fcom`，然后按 `<Tab>` 键来自动输入上面的注释文本。

输入该注释之后，就可以为代码定制它：

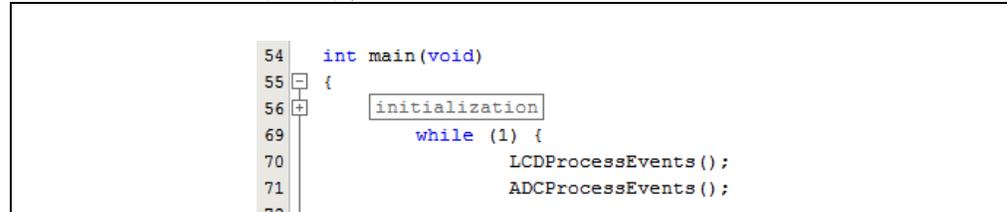
defaultstate	输入 collapsed 或 expanded。
desc	输入代码的说明。在折叠之后，将仍然可以看到该说明，以供参考。

下面给出了一个定制代码折叠的示例。

图 7-3: 展开的代码

```
54 int main(void)  
55 {  
56     // <editor-fold defaultstate="collapsed" desc="initialization">  
57     _display_state = DISP_HELLO; // Start from displaying of PIC24 banners  
58     AD1PCFG = 0xffff; // Setup PortA I/Os as digital  
59     SPIMPolInit(); // Setup SPI to communicate to EEPROM  
60     EEPROMInit(); // Setup EEPROM I/Os  
61     UART2Init(); // Setup the UART  
62     TimerInit(); // Setup the timer  
63     mLCDInit(); // Setup the LCD  
64     BtnInit(); // Setup debounce processing  
65     ADCInit(); // Setup the ADC  
66     BannerStart(); // Setup the banner processing  
67     RTCCInit(); // Setup the RTCC  
68     // </editor-fold>  
69     while (1) {  
70         LCDProcessEvents();  
71         ADCProcessEvents();
```

图 7-4: 折叠的代码



更多信息，请参见 NetBeans 帮助主题 [C/C++/Fortran Development>Working with C/C++/Fortran Projects>Editing Source Files in C/C++/Fortran Projects>Folding Blocks of Code](#) ([折叠代码块](#))。

7.4.2.2 定制代码折叠 —— 汇编

要对汇编代码进行定制折叠，请执行以下操作：

- 对于 MPASM 汇编器或 MPLAB XC16 汇编器代码，在代码前后输入以下注释：
`; <editor-fold defaultstate="collapsed" desc="user-description">`
 要折叠的汇编代码块
`; </editor-fold>`
- 对于 MPLAB XC32 汇编器代码，在代码前后输入以下注释：
`// <editor-fold defaultstate="collapsed" desc="user-description">`
 要折叠的汇编代码块
`// </editor-fold>`

或者

- 对于 MPASM 汇编器或 MPLAB XC16 汇编器代码，输入 `fcom`；然后按 `<Tab>` 键来自动输入以上相关注释文本。
- 对于 MPLAB XC32 汇编器代码，输入 `fcom//`；然后按 `<Tab>` 键来自动输入以上相关注释文本。

汇编代码将按照与 C 代码相同的方式折叠。关于代码折叠的示例，请参见上面的 C 代码示例。

7.4.2.3 展开 / 折叠代码定制折叠块

要展开 / 折叠代码块，请转至 **View** 菜单，或在编辑器窗口中单击右键，并选择如下之一：

1. [Code Folds \(代码折叠\) > Collapse Fold \(折叠\)](#) 可隐藏代码块。
2. [Code Folds > Expand Fold \(展开折叠\)](#) 可显示代码块。
3. [Code Folds > Collapse All \(全部折叠\)](#) 可隐藏所有折叠代码块。
4. [Code Folds > Expand All \(全部展开\)](#) 可显示所有折叠代码块。

关于未折叠（展开）和折叠代码的图示，请参见图 7-3 和图 7-4。

7.5 C 代码重构

注： 要查看该功能，请参见 **Start Page, My MPLAB X IDE** 选项卡，“Extend MPLAB” 部分，“Selecting Simple or Full-Featured Menus” 主题。

重构指的是使用小转换来重构代码，而不改变任何程序行为。正如重构表达式以使之更易于理解或修改，重构代码也是为了使它更易于阅读、更易于理解，以及可更快速地更新。并且，正如重构后的表达式必须产生相同的结果，重构后的程序也必须在功能上等效于原始的源代码。

重构代码的一些常见动机包括：

- 使代码更易于更改或更易于添加新功能
- 降低复杂性，以便更好地理解
- 删除不必要的重复
- 使代码可以用于其他需要或更一般的需求
- 提高代码性能

IDE 的重构功能可简化代码重构，其方式为评估您希望进行的更改、显示应用程序会受到影响的部分，并对您的代码进行所有必要的更改。例如，如果您使用 **Rename** 操作来更改类名，IDE 将在您的代码中查找该名称的所有使用实例，并可为您更改该名称的每一处实例。

- **Refactor**（重构）菜单
- 撤消重构更改
- 查找函数使用实例
- 重命名函数或参数
- 移动、复制和安全删除 C 代码

7.5.1 Refactor（重构）菜单

使用 IDE 的重构操作时，您可以更改代码的结构，并使代码的其余部分更新为反映您所做的更改。

重构操作可通过 **Refactor** 菜单使用（见第 11.2.6 节“**Refactor 菜单**”）。

7.5.2 撤消重构更改

可以通过以下步骤，使用 **Refactor** 菜单中的命令撤消所做的所有更改。在撤消重构时，IDE 将取消受重构影响的所有文件中的所有更改。

要撤消重构命令：

1. 转至主菜单条中的 **Refactor** 菜单。
2. 选择 **Undo**。
IDE 会取消受重构影响的所有文件中的所有更改。

如果在执行重构之后，任何受影响的文件发生了修改，重构 **Undo** 操作将不可用。

7.5.3 查找函数使用实例

您可以使用 Find Usages 命令，确定在项目源代码中使用的每一个函数实例。

要查找项目中使用了函数的位置：

1. 在 Navigator 窗口或源代码编辑器窗口中，右键单击函数名称，并选择 Find Usages (Alt-f7)。
2. Find Usages 命令会显示调用该函数的代码行。在 Find Usages 对话框中，单击 **Find**。

Usages (使用实例) 窗口会显示文件名和在该文件中找到的每个使用实例的代码行。

要跳转至函数的某个特定实例，请执行以下操作之一：

- 在 Usages 窗口中，使用左窗格中的向上和向下箭头按钮，从函数的一个使用实例转至下一个使用实例。
- 双击某个代码行，打开该文件并将光标定位到该代码行上。

其他 IDE 查找机制

可用于搜索项目中使用了特定文本的所有位置的其他 IDE 工具包括：

- **Finding and Replacing Text (查找并替换文本)**。在 Editor 中打开的源文件中搜索使用了特定文本的所有位置。选择 Edit>Find 可打开 Find 对话框，选择 Edit>Replace (替换) 可打开 Replace 对话框。这些命令会查找所有匹配字符串，无论字符串是否为 C 代码元素。
- **Find in Projects**。与 Find 命令一样，Find in Projects 命令也会搜索匹配字符串，不论字符串是否为函数名称。选择 Edit>Find in Projects 可打开 “Find in Projects” 对话框，然后输入要查找的文本字符串。

要在源文件中查找声明函数的位置，可以在 Navigator 窗口中双击函数。如果该函数是在其他源文件中声明的，则右键单击该函数，并从上下文菜单中选择 Navigate>Go To Declaration。

7.5.4 重命名函数或参数

要重命名函数或参数，并更新整个项目中对它的引用：

1. 在源代码编辑器中，右键单击该函数或参数，并从上下文菜单中选择 Refactor>Rename。
2. 在 Rename 对话框中，输入新的名称。
3. (可选) 单击 **Preview**。在源代码编辑器底部的 Refactoring (重构) 窗口中，查看将更改的代码行，并清除不希望更改的任何代码的复选框。
4. 单击 **Do Refactoring (执行重构)** 应用选定的更改。

要进行快速的原地重命名，请将光标放在要重命名的项上，并按下 Ctrl-R 组合键。然后输入新的名称。要完成重命名，请按 **Escape (ESC 键)**。

7.5.5 移动、复制和安全删除 C 代码

这些功能是特定于 C++ 代码的。请参见第 11.2.6 节 “Refactor 菜单”。

注:

第 8 章 项目文件和文件夹

8.1 简介

在 MPLAB X IDE 中，有几种方法可以查看项目文件和文件夹：

- Projects 窗口视图
- Files 窗口视图
- Favorites 窗口视图
- Classes 窗口视图

其他 MPLAB X IDE 文件和文件夹信息包括：

- 查看用户配置数据
- 导入 MPLAB IDE v8 项目 —— 相对路径
- 移动、复制或重命名项目
- 删除项目

8.2 PROJECTS 窗口视图

在 Projects 窗口中查看的项目文件夹是逻辑（虚拟）文件夹。关于该窗口的更多信息，请参见第 12.11 节“Projects 窗口”。

您至少需要添加源文件。MPLAB X IDE 会为您查找许多默认文件（头文件和链接描述文件）。您可以添加库和预编译的目标文件，以及已编辑的头文件和链接描述文件文件。不包含在编译中的文件可以放在“Important Files”下。

图 8-1: PROJECTS 窗口 —— 简单 C 代码项目

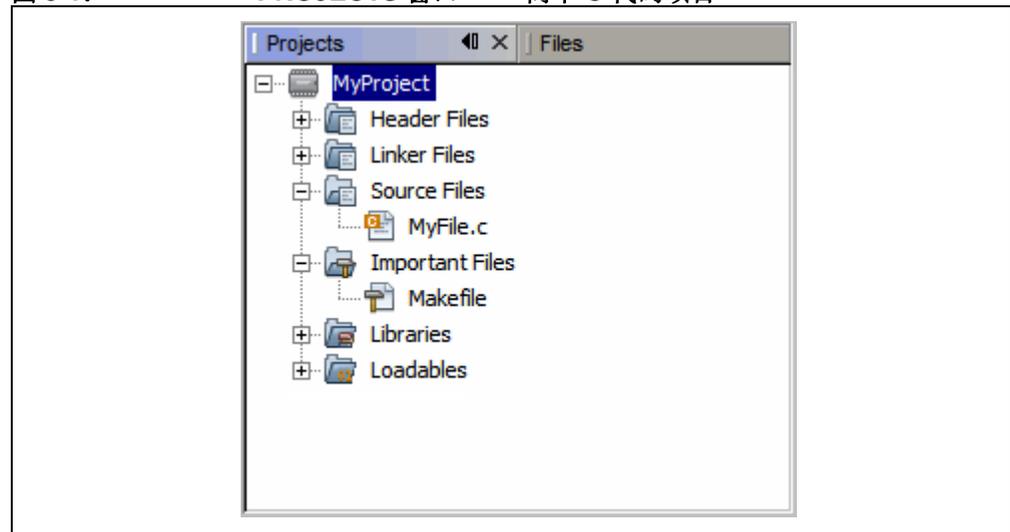


表 8-1: PROJECTS 窗口定义

虚拟文件夹	包含的文件
Header Files	头文件 (.h 或 .inc)
Library Files	库文件 (.a 或 .lib)
Linker Files	链接器文件 (.ld、.gld 或 .lkr)
Object Files	预编译目标文件 (.o)
Source Files	源文件 (.c 或 .asm)
Important Files	重要的文件，例如 makefile。其他文档可以放在此处，例如数据手册 PDF。

8.3 FILES 窗口视图

在 Files 窗口中查看的项目文件夹是实际文件夹（个人计算机上的文件夹）。关于该窗口的更多信息，请参见 NetBeans 帮助主题“Files Window”（Files 窗口）（用于 C 代码项目时）。

图 8-2: FILES 窗口 —— 简单 C 代码项目

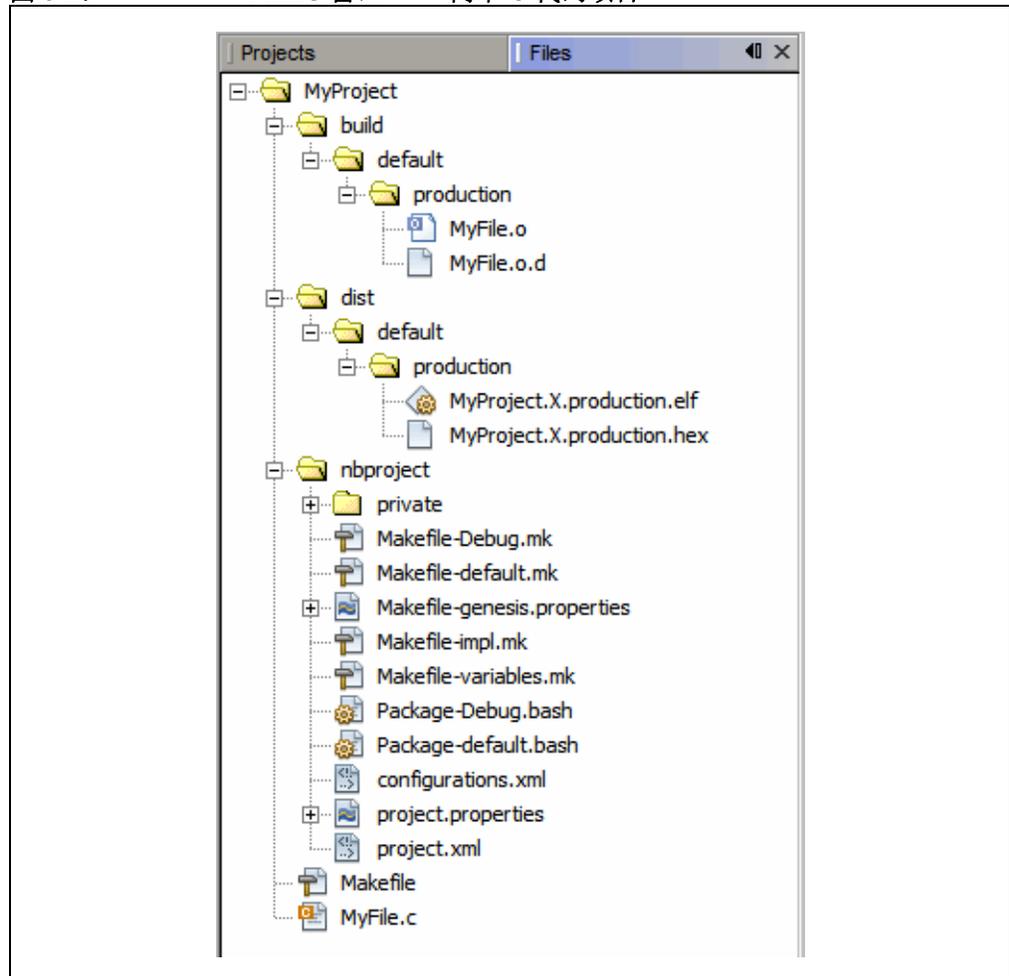


表 8-2: FILES 窗口定义

文件夹	说明
MyProject	项目文件夹，其中包含 Makefile 文件和 C 代码或汇编应用程序文件。Makefile 是项目的主 makefile。该文件在创建项目时生成，此后将不会触及它（不会重新生成它）。如果您熟悉 GNU make，可以对该文件进行更改。但 MPLAB X IDE 提供了添加编译前步骤和编译后步骤的方法（Project Properties），可以使用它们而不修改 Makefile 本身。
build ⁽¹⁾	中间文件的文件夹。文件将根据项目配置、使用情况和位置而包含在子文件夹中。 编译文件包括： <ul style="list-style-type: none"> • 运行文件（.o） • 依赖性文件（.o.d） • HI-TECH[®] 中间文件（.pl）
dist ⁽¹⁾	输出文件的文件夹。文件将根据项目配置、使用情况和位置而包含在子文件夹中。 分发文件包括： <ul style="list-style-type: none"> • 可执行文件（.hex） • ELF 或 COFF 目标文件（.elf 或 .cof） • 库文件（.a 或 .lib）
nbproject	makefile 和元数据文件夹。包含以下文件： <ul style="list-style-type: none"> • 项目 makefile • 特定于配置的 makefile • project.xml IDE 生成的元数据文件 • configurations.xml 元数据文件
default, MyConfig ⁽²⁾	项目配置文件夹。如果开发人员未创建任何配置，所有代码都将位于 default 中。
production, debug ⁽²⁾	生产和调试版本文件夹。
_ext ⁽²⁾	外部项目文件的文件夹。如果引用了项目文件夹之外的某个文件，则它将列在此处。

注 1: 不需要将这些文件夹检入源代码控制。编译过程将会创建它们。另请参见第 5.19 节“控制源代码”。

注 2: 图 8-2 未列出这些项。

8.4 FAVORITES 窗口视图

通过 Favorites 窗口 ([Window>Favorites](#))，您可以访问计算机或网络上的任何文件或文件夹，无论它是否处于一个项目中。

首次打开 Favorites 窗口时，它仅包含计算机的主目录。

- 要添加文件或文件夹，可以在 Favorites 窗口中单击右键并选择“Add to Favorites”（添加到收藏夹）。
- 要添加文件，可以在 Projects 窗口中右键单击文件名，并选择 [Tools>Add to Favorites](#)。

关于该窗口的更多信息，请参见 NetBeans 帮助主题“Favorites Window”（Favorites 窗口）。

8.5 CLASSES 窗口视图

对于支持 C++ 的编译器，可以使用 Classes 窗口 ([Window>Classes](#)) 查看项目中的类文件。

关于该窗口的更多信息，请参见 NetBeans 帮助主题“C/C++/Fortran Development>Working with C/C++/Fortran Projects>Navigating Source Files and Projects>Using the Classes Window（使用类窗口）”用于 C++ 代码项目时的相关窗口信息。

8.6 查看用户配置数据

MPLAB X IDE 用户目录 (`userdir`) 存储用户配置数据（例如窗口布局、编辑器设置、菜单和工具条定制项）和各种模块设置（例如已知编译器的列表）。此外，当您安装插件 ([Tools>Plugins](#)) 时，插件信息（代码）会被存储在 `userdir` 中，而不是 MPLAB X 安装目录中。

要确定用户目录所在的位置，可以选择 [Help>About](#)（关于），并查看 **Userdir** 旁的路径。该信息由 MPLAB X IDE 创建和管理。

8.7 导入 MPLAB IDE V8 项目 —— 相对路径

对于位于以下位置的 MPLAB IDE v8 项目：

```
C:\MyProjects\mplab8project
```

导入 MPLAB X IDE 中时，将得到：

```
C:\MyProjects\mplab8project\mplab8project.X
```

默认情况下，MPLAB X IDE 导入项目将放置在 MPLAB IDE v8 项目下，以保持这两个项目的可维护性。但是，您可以将 MPLAB X IDE 项目文件夹放置在任意位置。此外，项目的名称将设置为 *mplab8project.X*，但如果愿意的话，您可以重命名它。存在一项约定，即 MPLAB X IDE 项目的名称以 *.X* 结尾，但这只是一项约定。

新项目不会将源文件复制到其文件夹中，而是引用 v8 文件夹中的文件位置。要创建一个独立的 MPLAB X IDE 项目，需要创建一个新项目，并将 MPLAB IDE v8 源文件复制到其中。

详情请参见第 5.2 节“导入 MPLAB 旧项目”。

8.8 移动、复制或重命名项目

在创建项目之后，您可能发现需要进行更改。使用上下文（右键单击）项目菜单上的命令，可以从 MPLAB X IDE 内移动、复制或重命名您的项目。此外，还有一个选项用于删除项目。详情请参见 12.11.2 “Projects 窗口 —— 项目菜单”

您也可以使用外部工具；项目文件结构并不要求您使用 MPLAB X IDE。

8.9 删除项目

要在 MPLAB X IDE 中删除项目：

1. 在 Projects 窗口中，右键单击项目名称，并选择“Delete”（删除）。
2. 在“Delete Project”（删除项目）对话框中，您可以从计算机中删除项目文件或删除项目文件与所有源文件。进行选择并单击 **Yes**（是）。
3. 项目将被删除，因而 MPLAB X IDE 将不再可以看到它。但是，一些 makefile 信息会保留在计算机上。

要从计算机中删除所有项目文件：

1. 在 MPLAB X IDE 中删除或关闭项目。
2. 退出 MPLAB X IDE（见下面的注释）。
3. 删除文件。

注： 在关闭时，MPLAB X IDE 并不会释放分配给项目的所有资源。MPLAB X IDE 基于 Java 运行，所以释放资源不会立即发生。在 Java 下，最好让 JRE 决定何时在软件代码内调用垃圾回收（garbage collection, GC）。但是，您可以通过使能存储器工具条并单击它来强制执行 GC。

注:

第 9 章 疑难解答

9.1 简介

本节旨在帮助您对在使用 MPLAB X IDE 时遇到的任何问题、错误或疑问进行疑难解答。如果本信息未能帮助到您，请参见“支持”来了解联系 Microchip Technology 的方法。

- USB 驱动程序安装问题
- 跨平台问题
- MPLAB X IDE 问题
- NetBeans 平台问题
- 错误
- 论坛

9.2 USB 驱动程序安装问题

要安装正确的 USB 驱动程序，请参见第 2.3 节“安装 USB 设备驱动程序（对于硬件工具）”。

要解决出现的错误，请参见第 2.3.2.7 节“工具通信问题”。

9.3 跨平台问题

如果计划在不同平台（Windows、Mac 或 Linux 操作系统）上使用 MPLAB X IDE，请注意以下问题：

- 请在相对路径中使用正斜杠“/”。反斜杠“\”仅在 Windows 操作系统平台上工作。示例：`#include headers/myheader.h`。
- Linux 操作系统区分大小写，所以 `generictypedefs.h` 不同于 `GenericTypeDefs.h`。

9.4 MPLAB X IDE 问题

导入 MPLAB IDE v8 项目 —— 设置

在 MPLAB IDE v8 的工作区中保存的设置（例如工具设置）将不会转移到新的 MPLAB X IDE 项目中。关于工作区中所存储内容的信息，请参见 [MPLAB IDE v8 帮助](#) ([MPLAB IDE Reference](#)>[Operational Reference](#)>[Saved Information](#))。

导入 MPLAB IDE v8 项目 —— 修改的链接描述文件

如果您修改了 MPLAB IDE v8 项目链接描述文件，从而使它包含某个目标文件，则在导入到 MPLAB X IDE 时，由于 MPLAB IDE v8 和 MPLAB X IDE 的编译路径不同，链接器将无法找到该文件。

所以，可能会看到类似以下错误：

```
<install path>ld.exe: cannot find file.o
```

因为在 MPLAB IDE v8 中，与编译有关的所有文件都处于一个目录中，而在 MPLAB X IDE 中编译文件则处于不同子目录中。

您可以编辑您的链接描述文件，通过使用通配符来使之适用于 MPLAB X IDE。

例如，将：

```
/* Global-namespace object initialization - MPLAB v8*/  
.init :  
{  
    KEEP (crti.o(.init))  
    :  
} >kseg0_program_mem
```

更改为：

```
/* Global-namespace object initialization - MPLAB X*/  
.init :  
{  
    KEEP (*crti.o(.init))  
    :  
} >kseg0_program_mem
```

或者，您也可以使用地址属性，使您可以在 C 代码中设置函数的地址。

```
int __attribute__((address(0x9D001000))) myfunction (void) {}
```

这使您可以将函数放置在某个地址处，而无需修改默认的链接描述文件。

9.5 NETBEANS 平台问题

NetBeans 平台可能存在关于用于 MPLAB X IDE 的平台版本的问题。关于更多帮助，请访问 NetBeans 网站 (www.netbeans.org)。

另请参见：

<http://netbeans.org/community/releases/index.html>

9.6 错误

在 IDE 中，错误可能具有多种形式，最常见的形式是窗口中的图标或 Output 窗口中的消息。将鼠标悬停在图标上会弹出可能对问题进行说明的文本。对于文本消息，请参见联机帮助来搜索错误。

下面列出了一些错误。

命令行太长 (Windows XP 及更高版本)

当运行项目的 makefile 时，会将所执行的每个命令传递给本地的本机 shell。对于 Windows XP 和更高版本的操作系统，存在 8191 个字符的限制。因此，对于需要链接数百个 C 文件的项目，可能会超出该限制并显示该错误。

Linux OS 和 Mac OS X 通常可采用非常长的命令行。如果希望知道您的系统能接受的命令行长度，可在命令行中输入 “getconf ARG_MAX”。

响应文件变通方法

如果使用的是 MPLAB XC32 C 编译器或 MPLAB XC8 C 编译器 v1.01 及更高版本，可以在调用链接器时使用响应文件来避开该问题。

对于 MPLAB XC8，在 Project Properties 窗口 ([File>Project Properties](#)) 中，在 “Categories” 中单击 “XC8 linker” (XC8 链接器)。在 “Option categories” (选项类别) 下选择 “Additional Options” (其他选项)。选中 “Use response file to link” (使用响应文件进行链接) 旁的复选框。

对于 MPLAB XC32，在 Project Properties 窗口 ([File>Project Properties](#)) 中，在 “Categories” 中单击 “xc32-ld”。在 “Option categories” 下选择 “General”。选中 “Use response file to link” 旁的复选框。

库变通方法

对于所有其他编译器，可以创建一个 MPLAB X IDE 库项目，并将一些源文件从主项目移至库项目。然后将库项目添加到主项目中。

要创建库项目，请选择 [File>New Project](#)，单击 “Microchip Embedded” 类别，然后选择 “Library project” 作为项目。

要将库添加到主项目中，请打开 Project Properties 窗口 ([File>Project Properties](#))，在 “Categories” 下单击 “Libraries”，然后单击 **Add Library Project** 按钮。

无法为 Cygwin™ 堆保留空间（Windows 7 或 8）

MPLAB X IDE 在其 make 过程中会使用 Cygwin MinGW。在 Windows 7 或 8 中，可能存在虚拟存储器分配问题。

要更改 Virtual Memory（虚拟存储器）设置，请单击“开始”>“控制面板”。然后单击“系统”，并选择“安全”>“系统”>“高级系统设置”或“系统保护”。在高级选项卡上，单击性能设置按钮。在该对话框中单击高级选项卡，然后单击更改按钮。如下输入定制大小值：

- 初始大小（MB）= 当前已分配（显示在底部）
- 最大值（MB）= 建议（显示在底部）

单击设置，单击 OK，并重新启动您的个人计算机。

无法通过外部浏览器访问 URL。请检查浏览器配置

在 MPLAB X IDE 中，选择 *Tools>Options*，General 选项卡。在“Web Browser”下拉列表中，选择您的浏览器。单击 OK。

更多信息，请参见：

Bug 21236 - Ext-Browser: Message "Could not access the URL" to guide from Tools menu (http://netbeans.org/bugzilla/show_bug.cgi?id=21236)

Bug 38211 - A problem with using IE external browser in the IDE (http://netbeans.org/bugzilla/show_bug.cgi?id=38211)

9.7 论坛

如果未在此处看到您的问题，请访问我们的论坛：

<http://www.microchip.com/forums/f238.aspx>

此处可以找到关于问题的讨论和最新发布的解决方法。

第 10 章 MPLAB X IDE 与 MPLAB IDE v8 的差异

10.1 简介

MPLAB X IDE 与 MPLAB IDE v8 和先前版本具有很大的差异。本章中的主题将在移植问题方面为您提供帮助。

- 主要差异
- 菜单差异
- 工具支持差异

10.2 主要差异

由于 MPLAB X IDE 基于 NetBeans 平台，所以许多功能将不同于 MPLAB IDE v8。关于更详细的信息，请参见以下章节：

- 第 2 章 “开始之前”
- 第 3 章 “教程”
- 第 5 章 “附加任务”

下面简要介绍了主要的功能差异：

1. **MPLAB X IDE 基于开源的跨平台 NetBeans 平台。**第三方可以简便地以插件形式添加功能。特定于 Microchip 产品的 MPLAB X IDE 组件仍然是专有的。

MPLAB IDE v8 是专有的，并基于 Windows 操作系统。第三方可以根据来自 MPLAB 开发组的设计信息为 v8 添加功能。

2. **MPLAB X IDE 是基于项目的（无工作区）。**在 MPLAB X IDE 中，必须通过创建项目来开发应用程序。创建项目涉及到选择器件，以及选择和设置语言工具、调试工具、编程工具和其他项目具体信息。这可确保成功开发应用程序所需的所有项都存在。多项目分组通过多个配置进行处理。

MPLAB IDE v8 是基于器件的。虽然总是强烈建议您在 v8 中使用项目来创建应用程序，但这不是必需的。可以使用工作区来包含一些设置信息，包括多项目分组。

由于项目发生了显著变化，所以建议阅读第 3 章 “教程”。

3. **MPLAB X IDE 允许选择多个工具**

例 1：将几个 MPLAB ICD 3 调试器连接到计算机上的几个 USB 端口。然后，通过访问项目属性可在调试器之间简便地进行切换，调试器则通过它们的序列号（SN）标识。

例 2：将一个 MPLAB ICD 3 调试器连接到计算机的一个 USB 端口，将一个 MPLAB PM3 编程器各连接到另一个 USB 端口。然后，通过访问项目属性可在工具之间简便地进行切换。

MPLAB IDE v8 不允许选择多个工具。

4. MPLAB X IDE 允许选择多个语言工具版本。

示例：安装用于 PIC18 MCU 的 MPLAB C 编译器的两个版本。然后，通过访问项目属性可在编译器工具链版本之间简便地进行切换。

MPLAB IDE v8 不允许选择多个语言工具版本。

5. MPLAB X IDE 允许多个调试和编程会话。MPLAB X IDE 允许在一个 IDE 中具有多个活动的调试会话。更多信息，请参见第 6.3 节“处理多个项目”。

MPLAB IDE v8 允许一个调试或编程会话。MPLAB IDE v8 允许在 IDE 中打开多个项目。但是，每次只能对一个项目进行调试或编程。

6. MPLAB X IDE 允许多个编译配置。MPLAB X IDE 允许多个编译配置。更多信息，请参见第 6.4 节“处理多个配置”。

MPLAB IDE v8 仅允许两个配置。MPLAB IDE v8 仅允许从 Build Configuration（编译配置）下拉框中选择“Release”（发布）或“Debug”，以及在您自己的代码中使用 `__DEBUG`。

要在 MPLAB X IDE 中重新创建 MPLAB IDE v8 功能，可以按照第 6.4.2 节“添加副本配置”中的示例，创建您自己的调试配置和 `__DEBUG` 宏。

7. MPLAB X IDE 提供了多步选项来进行调试和编程。MPLAB X IDE 具有一个“Debug Project”图标，只需单击它即可进行编译、使用您的程序和调试执行程序（对于硬件工具）对目标器件编程，并在调试模式下运行您的代码。此外，还提供了“Make and Program”（Make 并编程）图标，只需单击它即可进行编译、对目标器件编程（对于硬件工具）并运行您的代码。如果不希望程序在进行 make 和编程之后运行，可以改为使用“Hold in Reset”图标。

8. MPLAB IDE v8 需要几个手动步骤来进行调试或编程。在调试或运行代码之前，MPLAB IDE v8 需要先完成一个过程：

- a) (1) 选择正确的编译配置（Release 或 Debug）
- b) (2) build/make 代码
- c) (3) 使用代码对目标编程（对于硬件工具）
- d) (4) 运行代码

对于一些工具（例如，MPLAB 入门工具包），仍然需要独立地执行一些步骤。MPLAB X IDE 在 *Debug>Discrete Debugger Operation* 下提供了该功能。

9. MPLAB X IDE 使用在代码中设置的配置位。MPLAB X IDE 要求在代码中设置配置位。但是，处于调试会话中时，可以在 Configuration bits 窗口中临时更改配置位，然后将这些设置保存到一个文件中以便稍后粘贴到代码中（见第 4.21.4 节“设置配置位”）。

MPLAB IDE v8 使用在代码或窗口中设置的配置位。MPLAB IDE v8 允许在代码或 Configuration bits 窗口中设置配置位。但是，在窗口中所做的设置需要手动输入代码中。

10. **MPLAB X IDE 调试工具仅在会话期间连接。** MPLAB X IDE 仅在调试或编程会话期间将调试或编程器工具与目标连接。其他情况下它们不连接。

MPLAB IDE v8 调试工具总是连接。 MPLAB IDE v8 会在选择工具之后立即与调试和编程器工具连接。这种配置不允许多个会话。

要在 MPLAB X IDE 中与 MPLAB IDE v8 一样总是保持该连接，请转至 Tools>Options (对于 Mac OS X 为 mplab_ide>Preferences)，**Embedded** 按钮，**Generic Settings** 选项卡，并选中“Maintain active connection to hardware tool”。

11. **MPLAB X IDE 信息会整合在一处显示。** MPLAB X IDE 具有 Dashboard 窗口，其中包含断点资源、校验和和存储器计量信息。请参见第 5.17 节“查看仪表板显示”。该窗口将这些信息和将来的信息放在同一个地方。

MPLAB IDE v8 信息分散在多处显示。 MPLAB IDE v8 具有断点资源工具条、校验和工具条和存储器计量窗口。每个功能以不同的方式访问。

12. **MPLAB X IDE 具有许多 NetBeans 功能。** MPLAB X IDE 具有许多 NetBeans 编辑和调试功能。(更多详细信息，请参见 NetBeans 帮助)。MPLAB X IDE 会定期更新它所基于的 NetBeans 平台。然后，IDE 将会更新为使用新的 NetBeans 功能。MPLAB X IDE 版本说明将指定每个 MPLAB X IDE 版本所基于的 NetBeans 平台版本。

MPLAB IDE v8 具有它自己的专有功能。 MPLAB IDE v8 是一个专有产品。因此，第三方和社区开发较困难。

10.3 菜单差异

以下表格列出从 MPLAB IDE v8 到 MPLAB X IDE 的菜单变化。由于 MPLAB X IDE 中的变化，所以不是所有 MPLAB IDE v8 菜单项都具有相同的映射。“备注”下介绍了主要的差异。

MPLAB X IDE 中可能会有更多菜单项可用。关于详细信息，请参见 NetBeans 帮助主题“IDE Basics”。

表 10-1: FILE 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
New	File>New File	在文件向导中选择关联的项目。
Add New File to Project (向项目中添加新文件)		
Open	File>Open File	
Close (关闭)	File>Close	
Save	File>Save	
Save As	File>Save As	
Save All	File>Save All	
Open Workspace (打开工作区)	不可用	项目中保存的所有数据
Save Workspace (保存工作区)		
Save Workspace As (工作区另存为)		
Close Workspace (关闭工作区)		
Import	File>Import	对于十六进制导入有用
Export (导出)	在 Projects 选项卡式窗口中右键单击项目中的文件，并选择“Export Hex” (导出十六进制)	
Print (打印)	File>Print File>Print to HTML (打印为 HTML) File>Page Setup (页面设置)	
Recent Files (最近的文件)	File>Open Recent File (打开最近的文件)	
Recent Workspaces (最近的工作区)	不可用	项目中保存的所有数据
Exit (退出)	File>Exit	

MPLAB X IDE 与 MPLAB IDE v8 的差异

表 10-2: EDIT 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Undo	Edit>Undo	
Redo	Edit>Redo	
Cut	Edit>Cut	
Copy	Edit>Copy	
Paste	Edit>Paste Edit>Paste Formatted (粘贴格式化内容)	
Delete	Edit>Delete	
Select All (全选)	Edit>Select All Edit>Select Identifier (选择标识符)	
Find	Edit>Find Edit>Find Selection (查找选定内容)	
Find Next (查找下一处)	Edit>Find Next Edit>Find Previous (查找上一处)	
Find In Files (在文件中查找)	Edit>Find in Projects Edit>Replace in Projects	
Replace	Edit>Replace	
Go To (转至)	Navigate>Go to Line (转至代码行) Navigate>Go to Declaration	
Go To Locator (转至定位器)	Navigate>Go to Symbol	
Go Backward (后退)	Navigate>Back	
Go Forward (前进)	Navigate>Forward (前进)	
External DIFF (外部 DIFF)	Tools>Diff	另请参见 Tools>Options (对于 Mac OS X 为 mplab_ide>Preferences), Miscellaneous, Diff 选项卡
Advanced	Source>Format Source>Shift Left/Right (左移/右移) Source>Move Up/Down (上移/下移) Source>Toggle Comment (翻转注释)	
Bookmarks (书签)	Navigate>Toggle Bookmark (翻转书签) Navigate>Next Bookmark Navigate>Previous Bookmark	
Properties	Tools>Options>Editor 选项卡, Tools>Options>Fonts & Colors	对于 Mac OS X, 请使用 mplab_ide>Preferences, 而不是 Tools>Options。

表 10-3: VIEW 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Project	Window>Projects	
Output	Window>Output>Output	
Toolbars	View>Toolbars	
CPU Registers* (CPU 寄存器)	Window>PIC Memory Views>CPU	
Call Stack	Window>Debugging>Call Stack	
Disassembly Listing (反汇编列表)	Window>Output>Disassembly Listing File	
EEPROM	Window>PIC Memory Views>EE Data Memory	在窗口中定制
File Registers	Window>PIC Memory Views>Data Memory (数据存储)	
Flash Data (闪存数据)	Window>PIC Memory Views>Data Memory	
Hardware Stack (硬件堆栈)	Debug>Stack (堆栈)	
LCD Pixel (LCD 像素)	尚不可用	
Locals (本地)	Window>Debugging>Variables	
Memory* (存储器)	Window>PIC Memory Views>Other Memory	在窗口中定制
Program Memory	Window>PIC Memory Views>Flash Memory	
SFR/Peripherals* (SFR/ 外设)	Window>PIC Memory Views>Peripherals	
Special Function Registers	Window>PIC Memory Views>SFRs	
Watch	Window>Debugging>Watches	另请参见: Debug>New Watch
Memory Usage Gauge (存储器使用情况计量)	Window>Dashboard	存储器使用情况位于 PE 窗口中
Trace	Window>Debugging>Trace	另请参见: File>Project Properties 来使能跟踪

* 仅对于 PIC32 MCU

MPLAB X IDE 与 MPLAB IDE v8 的差异

表 10-4: 项目菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Project Wizard (项目向导)	File>New Project	总是对于新项目调用
New	File>New Project	
Open	File>Open Project File>Open Recent Project (打开最近的项目)	
Close	File>Close Project (关闭项目)	
Set Active Project (设置活动项目)	Run>Set Main Project	
Quickbuild	不支持	所有开发都需要使用项目。
Package in .zip (以 .zip 打包)	在 Projects 选项卡式窗口中右键单击项目, 并选择 “Package” (打包)	
Clean (清除)	在 Project 选项卡式窗口中右键单击项目, 并选择 “Clean and Build”	
Locate Headers (查找头文件)	在 Projects 窗口中右键单击项目名称, 然后选择 “Locate Headers”	
Export Makefile (导出 Makefile)	尚不可用。	
Build All (全部编译)	Run>Run Project	编程器编译 / 运行 调试编译 / 运行
Make	Debug>Debug Project	
Build Configuration	File>Project Group (项目组) Run>Set Project Configuration (设置项目配置)	原先只有两种配置: Release 和 Debug。现在, 您可以决定需要多少配置。
Build Options (编译选项)	File>Project Properties	语言工具设置
Save Project (保存项目)	File>Save File>Save All	
Save Project As (项目另存为)	File>Save As	
Add Files to Project (向项目中添加文件)	在 Projects 选项卡式窗口中右键单击文件夹, 并选择 “Add Existing Item”	
Add New File to Project	File>New File	
Remove File from Project (从项目中删除文件)	在 Projects 选项卡式窗口中右键单击文件, 并选择 “Remove From Project” (从项目中删除)	
Select Language Toolsuite (选择语言工具套件)	File>Project Properties	
Set Language Tool Locations (设置语言工具位置)	Tools>Options, Embedded	对于 Mac OS X, 请使用 mplab_ide>Preferences, 而不是 Tools>Options。
版本控制	Tools>Options, Miscellaneous, Versioning 选项卡	对于 Mac OS X, 请使用 mplab_ide>Preferences, 而不是 Tools>Options。 另请参见 Team 菜单, Window>Versioning

表 10-5: 调试器菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Select Tool (选择工具)	File>New Project File>Project Properties	进行选择 进行更改
Clear Memory (清除存储器)	尚不可用	由编程器使用
Run	Run>Run Project Debug>Debug Project Debug>Continue	Programmer Run (编程器运行) Debug Run Run from Halt (从暂停中运行)
Animate	不支持	
Halt (暂停)	Debug>Pause	
Step Into	Debug>Step Into	
Step Over	Debug>Step Over	
Step Out	Debug>Step Out	
Reset	Debug>Reset	
Breakpoints	Debug>New Breakpoint Debug>Toggle Breakpoint	
Settings	File>Project Properties	
Stopwatch	Window>Debugging>Stopwatch	

表 10-6: 编程器 * 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Select Programmer (选择编程器)	File>New Project File>Project Properties	进行选择 进行更改
Enable Programmer (使能编程器)	不可用	选定之后, 就会使它
Disable Programmer (禁止编程器)		
Program	Program Target Project (编程目标项目)	位于 Run 工具条上
Verify (校验)	不可用	见 MPLAB IPE
Read (读取)	Upload Target Project (上传目标项目)	位于 Run 工具条上
Blank Check All (空白检查全部)	不可用	见 MPLAB IPE
Blank Check OTP (空白检查 OTP)	不可用	见 MPLAB IPE
Erase Flash Device (擦除闪存器件)	位于定制工具条上	见第 6.7 节 “定制工具条”
Reset Program Statistics (将编程统计复位)	不可用	
Download OS (下载操作系统)	File>Project Properties, 调试工具类别, Firmware (固件) 选项	IDE 会自动选择最新固件来进行下载
Settings	File>Project Properties	

* 关于编程器信息, 另请参见 MPLAB IPE。

表 10-7: TOOLS 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Data Monitor and Control Interface (DMCI, 数据监视和控制界面) 和其他插件	Tools>Plugins Tools>Embedded	进行安装 进行使用
MPLAB® Macros	Edit>Start Macro Recording (开始宏录制) Edit>Stop Macro Recording (停止宏录制)	
RTOS Viewer (RTOS 查看器)	Tools>Plugins Tools>Embedded	进行安装 进行使用

MPLAB X IDE 与 MPLAB IDE v8 的差异

表 10-8: 配置菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Select Device (选择器件)	File>New Project File>Project Properties	进行选择 进行更改
Configuration Bits	Window>PIC Memory Views>Configuration Bits	
External Memory (外部存储器)	Window>PIC Memory Views>Other Memory	在窗口中定制
ID Memory (ID 存储器)	Window>PIC Memory Views>Other Memory	在窗口中定制
Settings	Tools>Options	对于 Mac OS X, 请使用 mplab_ide>Preferences, 而不 是 Tools>Options。

表 10-9: WINDOW 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Close All (全部关闭)	Window>Close All Documents (关闭所有文档)	
Cascade (级联)	不支持。	对于文档管理, 请参见 Window>Documents (文档)。
Tile Horizontally (横向平铺)		
Tile Vertically (纵向平铺)		
Arrange Icons (排列图标)		
Window Sets (窗口集)		
Create Window Set (创建窗口集)		
Destroy Window Set (销毁窗口集)		
Recent Windows (最近的窗口)		

表 10-10: HELP 菜单差异

MPLAB® IDE v8	MPLAB® X IDE	备注
Topics (主题)	Help>Contents (内容)	要查看 PDF, 请参见第 8.2 节 “Projects 窗口视图”。
Release Notes (版本说明)	Help>Online Docs and Support (联机文档和支持)	
Driver Installation (驱动程序安装)	Help>Online Docs and Support	
Check for Updates (查看更新)	不可用	请查看 Microchip 网站获取更 新。
Web Links (网络链接)	Help>Online Docs and Support	
About MPLAB® IDE (关于 MPLAB® IDE)	Help>About (关于)	

10.4 工具支持差异

下表列出了可用的 Microchip 开发工具，以及 MPLAB X IDE 是否支持它们。

表 10-11: 工具支持差异

开发工具	在 MPLAB® X IDE 中是否支持	
	是	否
PICKit™ 3	X	
PICKit 2	X	
PICKit 1		X
MPLAB ICD 3	X	
MPLAB ICD 2		X
MPLAB REAL ICE™ 在线仿真器	X	
MPLAB ICE 2000		X
MPLAB ICE 4000		X
MPLAB PM3	X	
PRO MATE II		X
PICSTART® Plus		X
MPLAB VDI		X

表 10-12: 插件支持差异

插件	在 MPLAB® X IDE 中是否支持	
	是	否
DMCI	X	
dsPIC® 滤波器设计	X	
dsPIC30F SMPS 降压转换器	X	
dsPIC33F SMPS 降压 / 升压转换器 *	X	
dsPICworks	X	
图形显示设计器	X	
MATLAB®	X	
存储器入门工具包	X	
PC-Lint	X	
RTOS 查看器	X	
段式显示设计器 *	X	
AN851 自举程序		X
AN901 BLDC 调整界面		X
AN908 ACIM 调整界面		X
KEELOQ® 插件		X

* 开发中

第 11 章 桌面参考

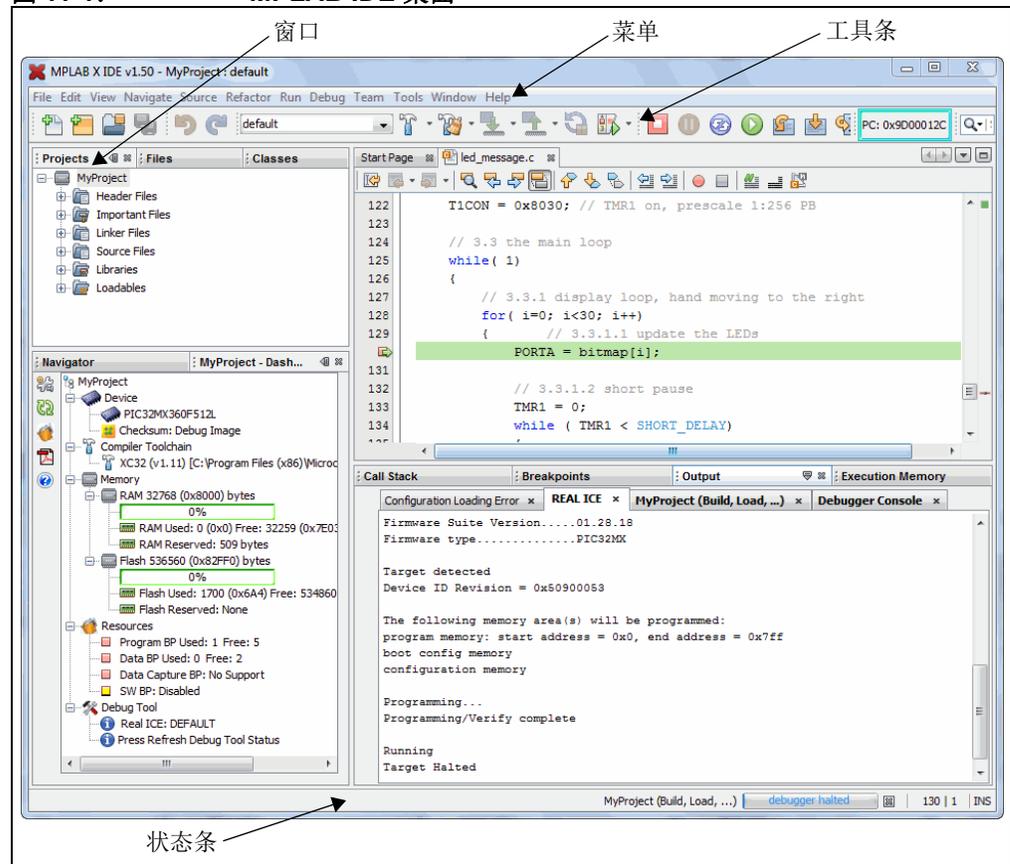
11.1 简介

MPLAB X IDE 桌面是一个可调整大小的窗口，它独立于其他桌面项而工作。桌面包含菜单、工具条、状态条和选项卡式窗口（图 11-1）。本章将介绍菜单、工具条和状态条。窗口和对话框在其各自的章节中介绍。

注： 当 NetBeans 帮助主题谈及某个工作区时，它指的是桌面。它指的**不是** MPLAB IDE v8（及其更早版本）的工作区。

- 菜单
- 工具条
- 状态条
- 灰显或缺失的项和按钮

图 11-1: MPLAB IDE 桌面



11.2 菜单

许多 MPLAB IDE 功能可通过位于桌面顶部的菜单条以菜单项形式进行访问。跟随省略号 (...) 的菜单项将打开一个对话框。关于对话框的更多信息，请参见第 12 章“MPLAB X IDE 窗口和对话框”。

菜单项旁会列出菜单项的快捷键。示例：“New File”的快捷键为 CTRL+N。NetBeans 帮助主题“IDE Basics>Keyboard Shortcuts（键盘快捷方式）”下提供了关于快捷键的更多信息。

菜单项可能由于各种原因而灰显。请参见第 11.5 节“灰显或缺失的项和按钮”。

通过在窗口中单击右键可显示更多的上下文菜单。关于这些菜单的更多信息，请参见第 12.11 节“Projects 窗口”。

下面列出了可用的菜单。

注： 要查看所有可用的菜单和菜单项，请参见 **Start Page**, [My MPLAB IDE>Extend MPLAB>Selecting Simple or Full-Featured Menus](#)。不是所有这些菜单项都适用于嵌入式开发。

- File 菜单
- Edit 菜单
- View 菜单
- Navigate 菜单
- Source 菜单
- Refactor 菜单
- Run 菜单
- Debug 菜单
- Team 菜单
- Tools 菜单
- Window 菜单
- Help 菜单

11.2.1 File 菜单

以下是 File 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)（菜单快捷方式）。

表 11-1: FILE 菜单选项

命令	操作
New Project	使用 New Project 向导创建一个新项目
New File	使用 New File 向导创建一个新文件
Open Project	打开现有项目
Open Recent Project	显示最近打开的所有项目的列表，以供选择
Import	要导入以下对象之一： Hex/ELF...（预编译）文件 —— 使用其他工具编译 MPLAB IDE v8 项目 —— 启动 Import Legacy Project 向导
Open Team Project	打开团队项目 关于团队服务器项目的更多信息，请参见 NetBeans 帮助“IDE Basics>Collaborative Development”。
Close Project	关闭当前项目
Close All Projects (关闭所有项目)	关闭所有已打开的项目

表 11-1: FILE 菜单选项 (续)

命令	操作
Open File	打开现有文件
Open Recent File	显示最近打开的所有文件的列表, 以供选择
Project Group	将当前项目与某个组相关联
Project Properties	打开 Project Properties 对话框
Save	保存当前文件
Save As	使用新的路径和 / 或名称保存当前文件
Save All	保存所有打开的文件 如果选择了“Compile on Save”功能, 则这也将 compile/build 项目文件。
Page Setup	设置页面以进行打印
Print	显示当前文件的打印预览, 并允许进行打印
Print to HTML	将当前文件打印为 HTML 格式的新文件
Exit	退出 MPLAB® IDE

11.2.2 Edit 菜单

以下是 Edit 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式, 请参见 NetBeans 帮助主题 [IDE Basics > Keyboard Shortcuts > Menu Shortcuts](#)。

表 11-2: EDIT 菜单选项

命令	操作
Undo	回退 (一次一个) 除 Save 之外的一系列编辑器操作
Redo	回退 (一次一个) 一系列 Undo 命令
Cut	删除当前选定内容并将其放置到剪贴板上
Copy	将当前选定内容复制到剪贴板中
Paste	将剪贴板的内容粘贴到插入点
Paste Formatted	将格式化的剪贴板内容粘贴到插入点
Delete	删除当前选定内容
Select All	选择当前文档或窗口中的所有内容
Select Identifier	选择距离光标最近的词
Find Selection	查找当前选定内容的实例
Find Next	查找已找到文本的下一个实例
Find Previous	查找已找到文本的上一个实例
Find	查找文本字符串
Replace	查找文本的字符串, 并使用指定字符串替换它
Find Usages	查找选定代码的使用实例和子类型
Find in Projects	在项目内查找指定的文本、对象名称和对象类型
Replace in Projects	在项目内替换文本、对象名称和对象类型
Start Macro Recording	开始录制按键操作
Stop Macro Recording	停止录制按键操作 要管理宏, 请选择 Tools > Options , Editor 按钮, Macros 选项卡。

11.2.3 View 菜单

以下是 View 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics > Keyboard Shortcuts > Menu Shortcuts](#)。

表 11-3: VIEW 菜单选项

命令	操作
Editors	选择一个可用编辑器来查看您的文件
Code Folds>Collapse Fold	如果插入点处于可折叠文本部分，则将这些代码行折叠为一行
Code Folds>Expand Fold	如果源代码编辑器中的当前选定行代表几个已折叠的行，则展开折叠内容来显示所有行
Code Folds>Collapse All	在源代码编辑器中折叠所有可折叠文本部分
Code Folds>Expand All	在源代码编辑器中展开所有可折叠文本部分
Web Browser	打开默认的网络浏览器并转至 NetBeans 主页
IDE Log	在 Output 窗口的选项卡中打开 MPLAB® X IDE 日志文件
Toolbars>File 等	在选择（选中）时显示关联的工具条
Toolbars>Small Toolbar Icons	在选择（选中）时在工具条上使用小图标
Toolbars>Reset Toolbars	复位为默认工具条设置
Toolbars>Customize	定制现有工具条上的项，并允许创建新的项
Show Editor Toolbar (显示编辑器工具条)	在选定（选中）时在 File 选项卡上显示编辑器工具条
Show Line Numbers (显示行号)	在选定（选中）时显示行号
Show Diff Sidebar (显示 Diff 侧边栏)	在选择（选中）时显示 DIFF 侧边栏
Show Versioning Labels (显示版本控制标签)	在选定（选中）时显示版本控制标签
Synchronize Editor with Views (将编辑器与视图同步)	在选定（选中）时会将编辑器与打开的视图同步
Show Profiler Metrics (显示性能分析器度量)	在选定（选中）时显示性能分析器度量
Full Screen (全屏)	将窗口展开到屏幕的完整长度和宽度

11.2.4 Navigate 菜单

以下是 Navigate 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics > Keyboard Shortcuts > Menu Shortcuts](#)。

表 11-4: NAVIGATE 菜单选项

命令	操作
Go to File	查找并打开特定文件
Go to Type	查找并打开特定的类或接口
Go to Symbol	查找指定的符号名称
Go to Previous Document (转至上一个文档)	打开在当前文档之前打开的上一个文档
Go to Source (转至源文件)	显示包含选定类的定义的源文件
Go to Declaration	跳转至光标下的项的声明
Go to Super Implementation (转至超类实现)	跳转至光标下的项的超类实现
Last Edit Location (上一个编辑位置)	将编辑器滚动到上一次发生编辑的位置
Back	后退
Forward	前进

表 11-4: NAVIGATE 菜单选项 (续)

命令	操作
Go to Line	跳转至指定的行
Toggle Bookmark	在代码行上设置书签
Next Bookmark	循环向前浏览书签
Previous Bookmark	循环向后浏览书签
Next Error	将源代码编辑器滚动到包含下一个编译错误的行
Previous Error	将源代码编辑器滚动到包含上一个编译错误的行
Select in Projects (在项目中选择)	打开 Projects 窗口, 并选择其中的当前文档
Select in Files (在文件中选择)	打开 Files 窗口, 并选择其中的当前文档
Select in Classes (在类中选择)	打开 Classes 窗口, 并选择其中的当前文档
Select in Favorites	打开 Favorites 窗口, 并选择其中的当前文档

11.2.5 Source 菜单

以下是 Source 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式, 请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-5: SOURCE 菜单选项

命令	操作
Format (格式化)	格式化选定代码, 如果未选择任何内容, 则格式化整个文件
Remove Trailing Spaces (删除尾随空格)	删除行尾空格
Shift Left	将选定的一行或多行向左移动一个制表符距离
Shift Right	将选定的一行或多行向右移动一个制表符距离
Move Up	将选定的一行或多行上移一行
Move Down	将选定的一行或多行下移一行
Duplicate Up (向上复制)	将选定的一行或多行向上复制一行
Duplicate Down (向下复制)	将选定的一行或多行向下复制一行
Toggle Comment	翻转当前行或选定行的注释
Complete Code (补全代码)	显示代码补全框。
Insert Code (插入代码)	弹出可用于生成构造函数、getter 和 setter 等常见结构的上下文菜单
Fix Code (修复代码)	显示编辑器提示。当有提示时, IDE 会在显示灯泡时通知您。
Show Method Parameters (显示方法参数)	选择下一个参数。 只有选定 (高亮显示) 一个参数时, 该快捷方式才会起作用。
Show Documentation (显示文档)	显示光标下的项的文档
Insert Next Matching Word (插入下一个匹配的词)	当您输入词的起始字符时, 生成下一个已在代码中其他位置使用的词
Insert Previous Matching Word (插入上一个匹配的词)	当您输入词的起始字符时, 生成上一个已在代码中其他位置使用的词
Scan for external changes (扫描外部更改)	扫描文件是否在 MPLAB® X IDE 之外进行了更改

11.2.6 Refactor 菜单

以下是 Refactor 菜单中的菜单项。可看到的项取决于要重构的对象（变量和函数等）的类型。更多信息，请参见第 7.5 节“C 代码重构”。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-6: REFACTOR 菜单选项

命令	操作
Rename	可用于将变量或函数的名称更改为更有意义的形式 此外，它还会更新项目中的所有源代码，使之按新名称引用元素。
Move（移动）	将某个类移至另一个包或另一个类中 此外，项目中的所有源代码也会更新为在新的位置引用该类
Copy	将某个类复制到相同或不同的包中
Safely Delete（安全删除）	检查对某个代码元素的引用，然后如果没有任何其他代码引用它，则自动删除该元素
Change Function Parameter （更改函数参数）	更改选定函数的参数数量和名称
Undo	撤消重构
Redo	重做重构

11.2.7 Run 菜单

以下是 Run 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-7: RUN 菜单选项

命令	操作
Run Project	运行主项目或选定项目
Test Project	启动项目的 JUnit 测试（Java 相关）
Build Project	编译项目中的所有文件
Clean and Build Project	删除（清除）先前生成的项目文件，然后重新编译项目中的文件
Batch Build Project （批量编译项目）	编译项目的多个配置 （仅嵌入式项目可用）
Set Project Configuration	选择项目配置——应为“default”（默认）
Set Main Project	通过从已打开项目的列表中选择来设置主项目
Run File	运行当前选定的文件
Test File	启动当前文件的 JUnit 测试（Java 相关）
Check File	根据标准来检查文件（XML 相关）
Validate File	根据标准来验证文件（XML 相关）
Repeat Build/Run （重复编译 / 运行）	暂停之后再次运行
Stop Build/Run （停止编译 / 运行）	结束运行

11.2.8 Debug 菜单

以下是 Debug 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 [NetBeans 帮助主题 IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-8: DEBUG 菜单选项

命令	操作
Debug Project	调试主项目或选定项目
Debug File	启动当前选定文件的调试会话
Debug Test File (调试测试文件)	在 JUnit 中启动文件的调试测试 (Java 相关)
Discrete Debugger Operation	执行调试操作，一次执行一个步骤 (分离式): Build, Program Target, Launch Debugger 这对于在调试和使用入门工具包期间更改存储器窗口设置很有用。
Finish Debugger Session	结束调试会话
Pause	暂停调试 —— 使用 “Continue” 可继续
Continue	“Pause” 之后继续调试，直到达到下一个断点或程序末尾
Step Over	执行程序的一个源代码行 如果该行是一个函数调用，则执行整个函数，然后停止。
Step Into	执行程序的一个源代码行 如果该行是一个函数调用，则程序执行到该函数的第一条语句，然后停止。
Step Instruction	执行一条机器指令 如果该指令是一个函数调用，则执行函数，并将控制返回给调用方。
Run to Cursor	运行当前项目，直到文件中的光标位置，并停止程序执行
Reset	复位器件
Set PC at cursor	将程序计数器 (PC) 值设置为光标的行地址
Focus Cursor at PC	将光标移至当前 PC 地址，并使该地址在窗口中居中
Stack>Make Callee Current	将被调用的方法设为当前调用 仅当在 Call Stack 窗口中选定某个调用时可用。
Stack>Make Caller Current	将产生调用的方法设为当前调用 仅当在 Call Stack 窗口中选定某个调用时可用。
Stack>Pop Topmost Call (弹出顶部的调用)	弹出堆栈顶部的调用
Toggle Line Breakpoint (翻转行断点)	在程序中的光标位置处添加行断点或删除断点
New Breakpoint	在指定行、异常或方法处设置新的断点
New Watch	添加指定符号来进行观察
New Run Time Watch	添加指定符号来观察将在程序运行 / 执行时改变值的符号
Disconnect from Debug Tool	断开 MPLAB [®] X IDE 和调试工具之间的通信 要重新连接，请选择 Run/Debug Run (运行 / 调试运行)。
Run Debugger/Programmer Self Test (运行调试器 / 编程器自检)	执行调试工具自检 对于支持自检的工具，请按照工具文档来设置硬件，然后运行该测试来确认操作是否正确。

11.2.9 Team 菜单

以下是 Team 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-9: TEAM 菜单选项

命令	操作
CVS, Mercurial, Subversion	显示特定于每个版本管理系统的子菜单 关于子菜单选项的更多信息，请参见产品文档
Local History	显示文件的本地历史记录或将文件还原为历史版本
Find Issues	在版本控制系统中查找问题
Report an Issue (报告问题)	向版本控制系统报告问题
Create Build Job (创建编译作业)	使用版本控制系统创建编译

11.2.10 Tools 菜单

以下是 Tools 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-10: TOOLS 菜单选项

命令	操作
Embedded	如果已添加插件则可见 —— 从子菜单中选择插件。
Licenses (许可证)	<ul style="list-style-type: none"> • 允许检入和检出编译器许可证。 • 允许激活许可证 关于更多信息，请参见编译器安装路径下的 docs 目录中的 XCLM 文档。
Apply Diff Patch (应用 Diff 补丁)	选择 Diff 补丁文件，并应用于您的代码
Diff	比较在 IDE 中选定的两个文件
Add to Favorites (添加到收藏夹)	将选定文件添加到 Favorites 窗口中
Templates	打开 Template Manager
DTDs and XML Schemas (DTD 和 XML 模式)	打开 DTD 和 XML Schemas Manager
Plugins	打开 Plugins Manager 关于详细信息，请参见 NetBeans 的帮助 “IDE Basics>Plugins>About Managing Plugins (关于管理插件)”。
Options	打开 Options 对话框 对于 Mac OS X: 使用 mplab_ide>Preferences 。

11.2.11 Window 菜单

以下是 Window 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式，请参见 NetBeans 帮助主题 *IDE Basics>Keyboard Shortcuts>Menu Shortcuts*。

表 11-11: WINDOW 菜单选项

命令	操作
Projects	打开 Projects 窗口
Files	打开 Files 窗口
Classes	打开 Classes 窗口
Favorites	打开 Favorites 窗口
Services	打开 Services 窗口
Tasks	打开 Task List (任务列表) 窗口
Dashboard	打开 Dashboard 窗口 请参见第 5.15 节“查看仪表板显示”。
Output>Output	打开或将 Output 窗口移至最前
Output>Search Results	打开或将 Search 窗口移至最前
Output>Find Usages Results	在窗口中显示“Find Usages”的结果
Output>Versioning Output (版本控制输出)	在窗口中显示版本控制操作的结果
Output>Refactoring Preview	打开重构结果的预览窗口
Output>Disassembly Listing File (Project Project)	打开项目 Project 的反汇编列表
Output>Call Graph	打开 Call Graph 窗口
Navigating>Navigator	打开 Navigator 窗口
Navigating>Hierarchy (层次结构)	打开 Hierarchy 窗口
Debugging>Variables	打开 Local Variables (局部变量) 调试器窗口
Debugging>Watches	打开 Watches 调试器窗口
Debugging>Call Stack	打开 Call Stack 窗口
Debugging>Breakpoints	打开 Breakpoints 窗口
Debugging>Sessions	打开 Sessions 窗口
Debugging>Sources	打开 Sources 窗口
Debugging>Disassembly	打开 Disassembly 窗口
Debugging>PIC AppIO	打开应用程序输入/输出窗口 关于应用程序 I/O 的更多信息，请参见 MPLAB® REAL ICE™ 在线仿真器文档。
Debugging>Trace	打开 Trace 窗口 关于跟踪的更多信息，请参见 MPLAB REAL ICE 在线仿真器文档。
Debugging>Stopwatch	打开 Stopwatch 窗口
Debugging>PC Sampling	打开 PC Sampling 窗口 关于 PC 采样和性能分析的更多信息，请参见 MPLAB REAL ICE 在线仿真器文档。
Versioning>CVS	选择 CVS 版本控制项 关于 CVS 的更多信息，请参见 http://www.nongnu.org/cvs
Versioning>Subversion	选择 Subversion 版本控制项 关于 Subversion 的更多信息，请参见 http://subversion.tigris.org
Versioning>Mercurial	选择 Mercurial 版本控制项 关于 Mercurial 的更多信息，请参见 http://mercurial.selenic.com
PIC Memory Views>Memory	打开指定的存储器窗 所显示的存储器取决于项目所使用的器件。

表 11-11: WINDOW 菜单选项 (续)

命令	操作
Simulator (模拟器) >Stimulus	打开 Simulator Stimulus 窗口
Simulator>Analyzer	打开 Simulator Analyzer 窗口
Simulator>IOPin	打开 Simulator IO Pin 窗口
Other>Exception Reporter	打开 Exception Reporter 窗口, 其中列出异常断点
Other>CSS Preview (CSS 预览)	打开级联样式表的 Preview 窗口
Other>CSS Style Builder (CSS 样式生成器)	打开 Style Builder (样式生成器) 窗口, 列出级联样式表规则
Other>Macro Expansion	打开 Macro Expansion 窗口, 以查看宏结构
Editor	打开一个空的编辑器窗口
Close Window (关闭窗口)	关闭当前窗口中的当前选项卡 如果窗口没有任何选项卡, 则关闭整个窗口。
Maximize Window (最大化窗口)	最大化源代码编辑器或当前窗口
Undock Window (取消停靠窗口)	将窗口与 IDE 分离
Clone Document (克隆文档)	克隆活动文档
Close All Documents	关闭源代码编辑器中所有已打开的文档
Close Other Documents (关闭其他文档)	关闭除活动文档之外的所有其他已打开文档
Documents	打开 Documents 对话框, 在其中您可以保存并关闭已打开文档的组
Reset Windows (复位窗口)	将窗口复位为其默认设置

11.2.12 Help 菜单

以下是 Help 菜单中的菜单项。

关于其中一些菜单项的键盘快捷方式, 请参见 NetBeans 帮助主题 [IDE Basics>Keyboard Shortcuts>Menu Shortcuts](#)。

表 11-12: HELP 菜单选项

命令	操作
Help Contents	显示 JavaHelp 查看器, 其中会列出所有已安装的帮助集
Online Docs and Support	打开 NetBeans 支持网页
Keyboard Shortcuts Card (键盘快捷方式卡)	显示键盘快捷方式文档
Start Page	打开或将 Start Page 选项卡移至所有其他已打开选项卡之前
About	显示关于 MPLAB® IDE 的窗口

11.3 工具条

MPLAB IDE 会根据您正在使用的功能或工具来显示不同的工具条。这些工具条中的图标提供了日常任务的快捷方式。要在工具条中添加或删除图标，或创建新的工具条，请参见第 6.7 节“定制工具条”。

工具条按钮可能由于各种原因而灰显。请参见第 11.5 节“灰显或缺失的项和按钮”。

可用的工具条

提供了以下基本工具条。

- File 工具条
- Clipboard 工具条
- Status Flags 工具条
- Undo/Redo 工具条
- Run 工具条
- Debug 工具条
- Memory 工具条
- Quick Search 工具条
- Editor 工具条

工具条功能

工具条具有以下功能：

- 将鼠标指针悬停在图标上可弹出图标功能。
- 单击工具条并拖动到工具条区域中的另一个位置。
- 在工具条区域中单击右键，可显示 / 隐藏工具条或更改一些工具条的内容。
- 选择 [View>Toolbars](#) 可显示 / 隐藏工具条、更改一些工具条的内容或创建定制工具条。

11.3.1 File 工具条

File 工具条当前包含以下功能的按钮图标。File 菜单上也提供了这些功能。

- New File——使用 New File 向导创建一个新文件。
- New Project——使用 New Project 向导创建一个新项目。
- Open Project——打开一个现有项目。
- Save All Files（保存所有文件）——保存所有已打开的文件。

11.3.2 Clipboard 工具条

Clipboard 工具条当前包含以下功能的按钮图标。Edit 菜单上也提供了这些功能。

- Cut——删除当前选定内容并将其放置到剪贴板上。
- Copy——将当前选定内容复制到剪贴板中。
- Paste——将剪贴板的内容粘贴到插入点。

11.3.3 Status Flags 工具条

Status Flags 工具条包含：

- PC——程序计数器（PC）当前值。

11.3.4 Undo/Redo 工具条

Undo/Redo 工具条当前包含以下功能的按钮图标。Edit 菜单上也提供了这些功能。

- Undo——回退（一次一个）除 Save 之外的一系列编辑器操作。
- Redo——回退（一次一个）一系列 Undo 命令。

11.3.5 Run 工具条

Run 工具条当前包含以下功能的按钮图标。Run、Debug 和项目上下文菜单也提供了这些功能。

- Set Project Configuration——选择项目配置。选择 “default” 或 “Customize”。
- Build Project——编译所有项目文件。单击向下箭头可显示其他相关选项。
- Clean and Build Project——删除来自先前编译的文件，然后编译所有项目文件。单击向下箭头可显示其他相关选项。
- Make and Program Device Project——编译，对目标编程，并运行选定项目。单击向下箭头可显示其他相关选项。
- Hold in Reset——编译，对目标编程，并使选定项目保持复位状态。
- Read Device Memory——读取目标器件存储器并装入 MPLAB X IDE。单击向下箭头可显示其他相关选项。
- Debug Project——编译，对目标编程，并调试运行选定项目。

11.3.6 Debug 工具条

Debug 工具条当前包含以下功能的按钮图标。Debug 菜单上也提供了这些功能。

- Finish Debugger Session——结束调试会话。
- Pause——暂停调试。使用 “Continue” 可继续。
- Reset——运行当前项目，直到文件中的光标位置，并停止程序执行。
- Continue——继续调试，直到达到下一个断点或程序末尾。
- Step Over——执行程序的一行源代码。如果该行是一个函数调用，则执行整个函数，然后停止。
- Step Over Expression——单步跳过表达式，然后停止调试。
- Step Into——执行程序的一行源代码。如果该行是一个函数调用，则程序执行到该函数的第一条语句，然后停止。
- Step Out——执行程序的一行源代码。如果该行是一个函数调用，则执行函数，并将控制返回给调用方。
- Run to Cursor——运行当前项目，直到文件中的光标位置，并停止程序执行。
- Apply Code Changes——将代码中的所有更改应用于正在执行的程序。
- Set PC at cursor——将程序计数器（PC）值设置为光标的行地址。
- Focus Cursor at PC——将光标移至当前 PC 地址并将当前 PC 地址在窗口中居中。

11.3.7 Memory 工具条

Memory 工具条会显示 MPLAB IDE 的当前 PC 存储器使用情况。单击显示内容可强制进行垃圾回收。

11.3.8 Quick Search 工具条

Quick Search 工具条会显示一个搜索文本框。单击放大镜旁的向下箭头可以选择搜索类型。

11.3.9 Editor 工具条

Editor 工具条当前包含以下功能的按钮图标。Edit 和 Source 菜单上也提供了这些功能。该工具条显示在包含当前文件源代码的选项卡的顶部。

- Last Edited（上次编辑）—— 移至上次进行编辑的行。
- Back—— 后退。
- Forward—— 前进。
- Find Selection—— 查找选定文本的第一个匹配项。
- Find Previous Occurrence（查找上一个匹配项）—— 查找选定文本的上一个匹配项。
- Find Next Occurrence（查找下一个匹配项）—— 查找选定文本的下一个匹配项。
- Toggle Highlight Search（切换高亮显示搜索）—— 开启 / 关闭为搜索选定的文本。
- Previous Bookmark—— 循环向后浏览书签。
- Next Bookmark—— 循环向前浏览书签。
- Toggle Bookmark—— 在代码行上设置书签。
- Shift Left—— 将选定的一行或多行向左移动一个制表符距离。
- Shift Right—— 将选定的一行或多行向右移动一个制表符距离。
- Start Macro Recording—— 开始录制按键操作。
- Stop Macro Recording—— 停止录制按键操作。
- Comment（注释）—— 通过添加 “//” 使选定行变为注释。
- Uncomment（取消注释）—— 通过删除 “//” 使选定注释变为代码行。
- Go to Header/Source（转至头文件 / 源文件）—— 在头文件和相关源代码之间移动。

11.4 状态条

状态条提供关于 MPLAB IDE 会话状态的最新信息。当前仅提供编辑器信息。

11.5 灰显或缺失的项和按钮

菜单项、工具条按钮或状态条项会因为几个原因而灰显（不可用）或缺失：

- 项 / 按钮与选定器件不具有的器件功能相关联，例如，PIC16F877A 不支持外部存储器。
- 项 / 按钮与选定工具不具有的工具功能相关联，例如 “Step Out” 对于 MPLAB ICD 3 不可用。
- 项 / 按钮与项目相关，但尚未选择任何项目，例如，项目编译功能不可用（No Active Project（无活动项目））。
- 项 / 按钮对于选定器件或工具不受支持。
- 项 / 按钮正在执行其功能，所以无法再次选择，例如，当程序正在运行时，“Run Project” 将灰显。
- 项 / 按钮与另一个项互斥，例如，当程序正在运行时，“Pause” 可用，而 “Continue” 则灰显；当程序暂停时，“Continue” 可用，而 “Pause” 则灰显。

第 12 章 MPLAB X IDE 窗口和对话框

12.1 简介

MPLAB X IDE 桌面拆分为一些包含选项卡式窗口的窗格。一些功能仅在选择某种功能之后可见。

例如，首次打开 MPLAB X IDE 时，只会打开 **Start Page**。打开某个项目之后，将打开基本窗口——即左上窗格中的 **Project** 和 **Files** 窗口，左下窗格中的 **Navigator** 窗口，以及右下窗格的 **Output** 窗口。**Start Page** 窗口将移至右上窗格。

MPLAB X IDE 窗口是基本 NetBeans 窗口和 MPLAB X IDE 特定窗口的组合。从菜单项中选择时，将打开对话框。与窗口一样，MPLAB X IDE 对话框是基本 NetBeans 对话框和 MPLAB X IDE 特定对话框的组合。关于 NetBeans 窗口和对话框的信息，请参见第 13 章“**NetBeans 窗口和对话框**”。

12.2 MPLAB X IDE 窗口管理

关于管理 IDE 窗口的信息，请参见 NetBeans 帮助主题 [IDE Basics > Configuring the IDE \(配置 IDE\) > Configuring the IDE's Workspace \(配置 IDE 的工作空间\) > Managing IDE Windows \(管理 IDE 窗口\)](#)。

下面提供了更多窗口信息。

12.2.1 窗口数据更新

打开的窗口在程序暂停时发生更新（闪存存储器窗口除外，这些窗口必须进行读取。）程序暂停包括运行后暂停和单步运行。暂停时的更新会产生以下影响：

- 速度——更新需要一定的时间。要减少更新时间，请关闭所有未用窗口。
- 数据覆盖——暂停时会读取在已打开窗口中显示的文件寄存器的值。关于读取时的寄存器操作，请参见器件数据手册。

12.2.2 窗口数据更改

MPLAB X IDE 窗口数据可以按以下所述进行编辑。如果无法编辑数据，则说明该信息不可供您更改。

- 数据可以“就地”编辑。通过双击选择某个项，然后输入新值，或单击某个项旁边的省略号 (...) 并在弹出窗口中输入新值。
- 在只有某些选项可接受时，可以从下拉列表中选择数据。

12.2.3 窗口焦点

要确保具有一个处于焦点的窗口，不仅可以单击窗口框架，而且可以单击按钮、表格单元格或下拉组合框。

12.3 MPLAB X IDE 窗口与相关菜单和对话框

MPLAB X IDE 按原样使用一些 NetBeans 窗口。但是，其他窗口及其相关菜单则特别针对嵌入式使用而修改或创建。

特定窗口 *	相关菜单
Breakpoints 窗口	Window>Debugging>Breakpoints
Customize Toolbars 窗口	View>Toolbars
Licenses 窗口	Tools>Licenses
Dashboard 窗口	Window
存储器窗口	Window>PIC Memory Views
Output 窗口	Window>Output>Output
Project Properties 窗口	File
Projects 窗口	Window
Tools Options Embedded 窗口	Tools>Options (Windows 和 Linux 操作系统) mplab_ide>Preferences (Mac OS X)
Trace 窗口	Window>Debugging
Watches 窗口	Window>Debugging
向导窗口	File>New Project File>New File

* 根据您已安装的工具，可能可以查看特定于工具的窗口。关于特定窗口的信息，请参见相应工具的文档。

12.4 BREAKPOINTS 窗口

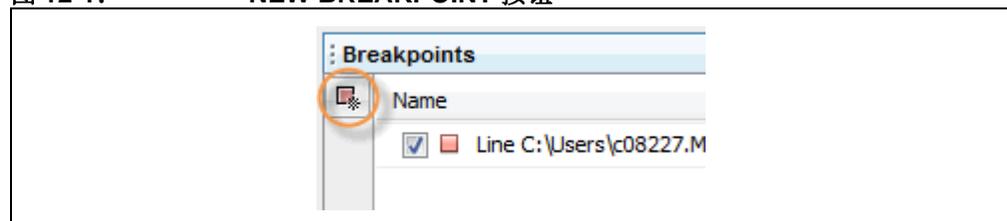
Breakpoints 窗口用于设置和查看代码中的断点。通过选择 **Window>Debugging>Breakpoints** 打开窗口。

关于使用 Breakpoints 窗口的更多信息，请参见第 4.17 节“使用断点控制程序执行”。

12.4.1 New Breakpoint 对话框

单击 Breakpoint 窗口中的 **Create New Breakpoint**（创建新断点）按钮可打开 New Breakpoint 对话框。

图 12-1: NEW BREAKPOINT 按钮



对应于断点的选项由所选断点的类型和所选器件决定（并非所有器件具有所有选项）。

12.4.2 行断点

以下选项可用于在代码行上指定的断点。

表 12-1: 断点类型: 行 —— 设置

项	说明
Settings	通过用鼠标左键单击文件行旁边的编辑器装订线可创建新的行断点。或者，从编辑器上下文菜单中选择“Toggle Line Breakpoint”（切换行断点）。

12.4.3 数据断点

以下选项可用于数据存储器断点。

表 12-2: 断点类型: 数据 —— 设置

项	说明
Project (项目)	从下拉列表中选择一个打开的项目 它是代码中包含了该断点的项目。
Symbols (符号)	输入全局符号或 SFR 的名称，或通过单击 Symbols 浏览选择一个。根据“Breaks on”（中断条件）访问该符号会触发代码执行发生暂停。
Enable Range Address (使能范围地址)	选中时可设置范围断点。 取消选中时可设置单个断点。
Address (地址) Address (Start) (地址 (起始))	根据“Enable Range Address”的选择，该项可能为“Address”或“Address (Start)”。 输入数据存储器中的十六进制地址 根据“Breaks on”访问该地址会触发代码执行发生暂停。
Address (End) (地址 (结束))	如果选中了“Enable Range Address”，则会启用该输入框。 输入数据存储器中的十六进制地址。
Breaks on	Read (读)、 Write (写) 和 Read or Write (读或写)：当上述符号或地址被读取、写入或者读取或写入时，中断代码执行。 Read Specific Value (读特定值)、 Write Specific Value (写特定值) 和 Read or Write Specific Value (读或写特定值)：当上述符号或地址被读取并具有下面指定的值、被写入下面指定的值或者根据下面指定的值发生读取或写入时，中断代码执行。 注： 对于 dsPIC DSC，具有对应于 X 和 Y 总线的读和写选项。
Value (值)	对“Breaks on”选择 Read Specific Value、Write Specific Value 或 Read or Write Specific Value 时，在此处输入一个十六进制值。
Value Comparison (值比较)	<i>仅对于 PIC16F1xxx 器件 ——</i> 与指定的“Value”进行比较： = Value: 等于值 != Value: 不等于值 > Value: 大于值 < Value: 小于值
Data Value Mask (数据值掩码)	<i>仅对于 PIC16F1xxx 器件 ——</i> 在与“Value”比较时使用掩码 输入一个介于范围 0x00 到 0xhh 之间的值，其中： 0x00: 不比较任何位 0xhh: 比较所有位

表 12-3: 断点类型: 数据 —— 次数计数

项	说明
Condition (条件)	<p>确定何时发生 “Breaks on” 下指定的中断。</p> <p>Always Break (总是中断): 总是在满足 “Breaks on” 条件时发生中断。</p> <p>Break occurs Count Instructions after Event (事件之后的 Count 条指令后发生中断): 发生事件 (“Breaks on” 条件) 后, 在实际中断之前先执行 Count 条指令。</p> <p>Event must occur Count times (事件必须发生 Count 次): 只有发生 Count 次事件 (“Breaks on” 条件) 之后, 才会实际发生中断。</p>
Count* (计数)	根据指定的条件, 输入事件后的指令计数数量或事件数量。

* 另请参见 “对于某些器件 (PIC16F1xxx MCU), 提供了增强型事件断点操作。”。

表 12-4: 断点类型: 数据

项	说明
Trigger Options (触发选项)	<p>仅对于 PIC16F1xxx 器件 ——</p> <p>选择何时触发:</p> <ul style="list-style-type: none"> • 到达断点时不触发输出 • 到达断点时触发输出
Interrupt Context (中断上下文)	<p>仅对于 PIC16F1xxx 器件 ——</p> <p>地址 / 数据断点的中断上下文限定符。选择如下:</p> <ul style="list-style-type: none"> • 总是中断 (在 ISR 和主代码中均中断) • 仅在主干 (非中断) 上下文中发生中断 —— 仅在主代码中发生中断 • 仅在中断上下文中发生中断 —— 仅在 ISR 代码中发生中断

12.4.4 地址断点

以下选项可用于程序存储器断点。

表 12-5: 断点类型: 地址 —— 设置

项	说明
Project	从下拉列表选择一个打开的项目 它是代码中包含了该断点的项目。
Enable Range Address	选中时可设置范围断点。 取消选中时可设置单个断点。
Address Address (Start)	根据 “Enable Range Address” 的选择, 该项可能为 “Address” 或 “Address (Start)”。 输入程序存储器中的十六进制地址。根据 “Breaks on” 访问该地址会触发代码执行发生暂停。
Address (End)	如果选中了 “Enable Range Address”, 则会启用该输入框。 输入数据存储器中的十六进制地址。
Breaks on	<p>Program Memory Execution (程序存储器执行): 到达上面指定的地址时, 中断代码执行。</p> <p>TBLRD Program Memory (表读程序存储器): 在对以上指定的地址进行表读时, 中断代码执行。</p> <p>TBLWT Program Memory (表写程序存储器): 在对以上指定的地址进行表写时, 中断代码执行。</p>

表 12-6: 断点类型: 地址 —— 次数计数

项	说明
Condition	确定何时发生 “Breaks on” 下指定的中断 Always Break: 总是在满足 “Breaks on” 条件时发生中断。 Break occurs Count Instructions after Event: 发生事件 (“Breaks on” 条件) 后, 在实际中断之前先执行 Count 条指令。 Event must occur Count times: 只有发生 Count 次事件 (“Breaks on” 条件) 之后, 才会实际发生中断。
Count*	根据指定的条件, 输入事件后的指令计数数量或事件数量。

* 另请参见 “对于某些器件 (PIC16F1xxx MCU), 提供了增强型事件断点操作。”。

表 12-7: 断点类型: 地址

项	说明
Trigger Options	仅对于 PIC16F1xxx 器件。 选择何时触发: <ul style="list-style-type: none"> • 到达断点时不触发输出 • 到达断点时触发输出
Interrupt Context	仅对于 PIC16F1xxx 器件。 地址 / 数据断点的中断上下文限定符。选择如下: <ul style="list-style-type: none"> • 总是中断 (在 ISR 和主代码中均中断) • 仅在主干 (非中断) 上下文中发生中断 —— 仅在主代码中发生中断 • 仅在中断上下文中发生中断 —— 仅在 ISR 代码中发生中断

12.4.5 事件断点

以下选项可用于事件断点。

表 12-8: 断点类型: 事件

项	说明
Project	从下拉列表选择一个打开的项目 它是代码中包含了该断点的项目。
Break on clock mode switch (发生时钟模式切换时中断)	当时钟模式切换时发生中断
Break on Reset instruction (发生复位指令时中断)	发生器件复位指令时中断
Break on Sleep (休眠时中断)	进入休眠时中断
Break on stack over/underflow (堆栈上溢 / 下溢时中断)	在堆栈上溢或下溢时中断
Break on wake up (唤醒时中断)	器件从休眠中唤醒时中断
Break when watchdog timer has expired (看门狗定时器计满时中断)	在看门狗定时器周期结束时中断
Break on execution out of bounds (执行超出范围时中断)	在程序试图移出正常程序存储器 / 空间时中断
Break on MCLR Reset (MCLR 复位时中断)	发生主复位 (MCLR) 时中断
Break on trigger in signal (触发输入信号时中断)	检测到触发输入脉冲时中断

* 从列表中选择将导致执行代码暂停 (中断) 的事件。某些事件可能对于您的器件不可用。

对于某些器件（PIC16F1xxx MCU），提供了增强型事件断点操作。

操作	说明
Break（中断）	根据指定的选项中断（暂停）执行
Trigger out（触发输出）	根据指定的选项发出触发脉冲输出
Break and trigger out（中断和触发输出）	根据指定的选项中断（暂停）执行并发出触发输出脉冲

12.4.6 次数计数操作

使用次数计数使您可以延迟中断，直到达到指定的计数。

Break occurs Count Instructions after Event

计数是在断点之后且在停止之前的指令执行次数。

例如：

- 0 表示立即停止执行
- 1 表示在一条额外指令之后停止执行
- 10 表示在十条额外指令之后停止执行

Event must occur Count times

计数是在停止之前执行事件的次数。

例如：

- 0 表示立即停止执行
- 1 表示执行事件一次，然后在下一次（事件第二次发生时）停止
- 10 表示执行事件十次，然后在下一次（事件第十一次发生时）停止

12.5 CUSTOMIZE TOOLBARS 窗口

您可以使用 **Customize Toolbars** 窗口定制 MPLAB X IDE 工具条。选择 **View>Toolbars>Customize** 打开窗口。

关于定制说明，请参见第 6.7 节“定制工具条”。

12.6 LICENSES 窗口

MPLAB X IDE 提供了两个用于管理网络编译器许可证的弹出窗口。选择 **Tools>Licenses** 可打开这些窗口。

注： 许可证管理器不会在尝试连接之前检测是否存在网络服务器连接。如果您使用的是 Windows 操作系统，并尝试在没有网络连接可用时连接网络，则会遇到延时。

表 12-9: NETWORK COMPILER LICENSE (网络编译器许可证) 窗口

Tools>License 菜单项	窗口	说明
Roam Network License (检入 / 检出网络许可证)	Network License Management (网络许可证管理)	检入或检出网络许可证。 注： 对于检出，将会存在 60 分钟的延时，之后服务器许可证才能由其他客户端使用。
Activate Network License (激活网络许可证)	Activate Network License (激活网络许可证)	输入激活密钥来激活网络编译器许可证。

关于网络许可证的详细信息，请参见 “*Installing and Licensing MPLAB® XC C Compilers*” (DS50002059)。

12.7 DASHBOARD 窗口

Dashboard 窗口包含常规项目信息，例如校验和、存储器使用情况和断点资源。详情请参见第 5.17 节 “查看仪表盘显示”。

12.8 存储器窗口

存储器窗口 (**Window>PIC Memory Views**) 会显示许多器件存储器类型，例如 SFR 和配置位。使用 “Memory” 和 “Format” 下拉框可以定制您的窗口。

关于这些控件的更多信息，请参见第 4.12 节 “设置文件和文件夹属性”。

可用的 PIC 存储器视图：8 位和 16 位器件

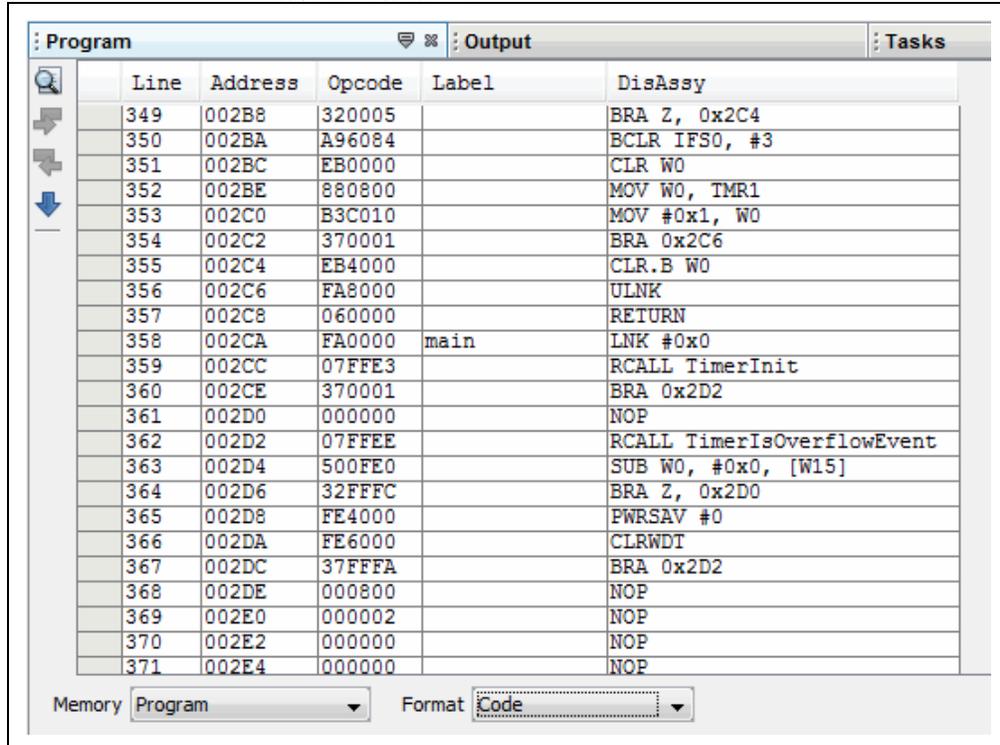
- Program Memory (程序存储器) 窗口
- File Registers (文件寄存器) 窗口
- SFRs 窗口
- Configuration Bits (配置位) 窗口
- EE Data Memory (EE 数据存储器) 窗口
- Other Memory (其他存储器) 窗口

可用的 PIC 存储器视图：32 位器件

- Execution Memory (执行存储器) 窗口
- Data Memory (数据存储器) 窗口
- Peripherals 窗口
- Configuration Bits 窗口
- CPU Registers (CPU 寄存器) 窗口
- User ID Memory (用户 ID 存储器) 窗口

存储器窗口上下文 (右键单击) 菜单: 第 12.8.13 节 “存储器窗口菜单” 提供了关于上下文菜单的信息。

图 12-2: 具有内容的存储器窗口



Line	Address	Opcode	Label	DisAssy
349	002B8	320005		BRA Z, 0x2C4
350	002BA	A96084		BCLR IFS0, #3
351	002BC	EB0000		CLR W0
352	002BE	880800		MOV W0, TMR1
353	002C0	B3C010		MOV #0x1, W0
354	002C2	370001		BRA 0x2C6
355	002C4	EB4000		CLR.B W0
356	002C6	FA8000		ULNK
357	002C8	060000		RETURN
358	002CA	FA0000	main	LNK #0x0
359	002CC	07FFE3		RCALL TimerInit
360	002CE	370001		BRA 0x2D2
361	002D0	000000		NOP
362	002D2	07FFEE		RCALL TimerIsOverflowEvent
363	002D4	500FE0		SUB W0, #0x0, [W15]
364	002D6	32FFFC		BRA Z, 0x2D0
365	002D8	FE4000		PWRSVAV #0
366	002DA	FE6000		CLRWDI
367	002DC	37FFFA		BRA 0x2D2
368	002DE	000800		NOP
369	002E0	000002		NOP
370	002E2	000000		NOP
371	002E4	000000		NOP

12.8.1 Program Memory（程序存储器）窗口

Program Memory 窗口会显示处于当前所选处理器的程序存储器范围内的存储单元。如果所选器件支持并使能了外部程序存储器，则它也会显示在 Program Memory 窗口中。

对于 MPLAB X 软件模拟器，当程序存储器值发生更改或处理器暂停时，Program Memory 窗口中的数据会发生更新。

对于任何 Microchip 硬件调试工具（例如，MPLAB REAL ICE 在线仿真器），当某个程序存储器值发生更改或处理器暂停时，Program Memory 窗口中的数据不会发生更新；您必须对器件存储器执行读操作。

您可以通过单击 Program Memory 窗口底部的下拉框之一来更改操作码在窗口中的显示方式：

- Hex（十六进制）
- Code（代码）
- PSV Mixed（PSV 混合）（仅适用于 dsPIC DSC/PIC24 器件）
- PSV Data（PSV 数据）（仅适用于 dsPIC DSC/PIC24 器件）

12.8.1.1 CODE（代码）

Code 格式会显示反汇编的十六进制代码和符号。窗口将具有以下列：

- **Debug Info**（调试信息）—— 可用于调试的信息
一个指针会显示程序计数器的当前位置。
- **Line**（行）—— 参考行号
- **Address**—— 操作码十六进制地址
- **Opcode**（操作码）—— 十六进制操作码，以 2 或 3 字节块的形式显示
对于大多数 PIC MCU，这些块代表字。
对于 PIC18CXXX 器件，这些块代表 2 字节。
对于 dsPIC DSC 器件，这些块代表 3 字节。
- **Label (Symbolic Only)**（标号（仅符号））—— 以符号格式显示的操作码标号。
- **Disassembly**（反汇编）—— 操作码助记符的反汇编版本。

12.8.1.2 HEX（十六进制）

该格式将以十六进制代码的形式显示程序存储器信息。窗口将具有以下列：

- **Address**—— 下一列中的操作码的十六进制地址
- **Opcode Blocks**（操作码块）—— 十六进制操作码，以 2 或 3 字节块的形式显示
对于大多数 PIC MCU，这些块代表字。对于 PIC18CXXX 器件，这些块代表 2 字节。
对于 dsPIC DSC 器件，这些块代表 3 字节。
高亮显示的操作码块代表程序计数器的当前位置。
- **ASCII**—— 相应操作码行的 ASCII 表示形式

12.8.1.3 PSV MIXED（PSV 混合）（仅适用于 dsPIC DSC/PIC24 器件）

该格式会以操作码和 PSV 区域的形式显示程序存储器（CORCON 寄存器，PSV 位置 1）。窗口将具有以下列：

- **Debug Info**—— 可用于调试的信息。一个指针会显示程序计数器的当前位置。
- **Line**—— 参考行号。
- **Address**—— 操作码十六进制地址。
- **PSV Address**（PSV 地址）—— 操作码的数据空间十六进制地址。
- **Data**—— 格式化为数据形式的操作码。
- **Opcode**—— 十六进制操作码，以 3 字节块的形式显示。
- **Label**（标号）—— 以符号格式显示的操作码标号。
- **Disassembly**—— 操作码助记符的反汇编版本。

更多信息，请参见“*dsPIC30F Family Reference Manual*”（DS70046）。

12.8.1.4 PSV DATA（PSV 数据）（仅适用于 dsPIC DSC/PIC24 器件）

当程序空间在数据空间中可见时，该格式会以文件寄存器的形式显示程序存储器（CORCON 寄存器，PSV 位置 1）。窗口将具有以下列：

- **Address**—— 数据的程序空间十六进制地址
- **PSV Address**—— 数据的数据空间十六进制地址
- **Data Blocks**（数据块）—— 十六进制数据，以 3 字节块的形式显示
高亮显示的数据块代表程序计数器的当前位置。
- **ASCII**—— 相应数据行的 ASCII 表示形式

更多信息，请参见“*dsPIC30F Family Reference Manual*”（DS70046）。

12.8.2 File Registers（文件寄存器）窗口

File Registers 窗口会显示选定器件的所有文件寄存器。当文件寄存器值发生更改，或处理器被查询时，File Registers 窗口中的数据会发生更新。

注： 为了加快使用某些硬件工具进行调试的速度，请关闭该窗口。改为使用 SFRs 窗口或 Watches 窗口。

您可以通过单击 File Registers 窗口底部的按钮之一来更改数据在窗口中的显示方式。

- Hex
- Symbol
- Dual Port（双端口）（仅适用于 dsPIC33FJ DSC/PIC24HJ MCU 器件）
- XY Data（XY 数据）（仅适用于 dsPIC DSC 器件）

12.8.2.1 HEX

该格式将以十六进制数据的形式显示文件寄存器信息。窗口将具有以下列：

- Address—— 下一列中的数据的十六进制地址
- Data Blocks—— 十六进制数据，以 1 字节或 2 字节块的形式显示
- ASCII—— 相应数据行的 ASCII 表示形式

12.8.2.2 SYMBOL

该格式将以符号方式显示每个文件寄存器，并以十六进制、十进制、二进制和字符格式显示相应的数据。窗口将具有以下列：

- Address—— 数据十六进制地址
 - Symbol Name（符号名称）—— 数据的符号名称
 - Radix Information（基数信息）—— Hex、Decimal（十进制）、Binary（二进制）和 Char（字符）
- 这四列中会显示基数信息。十六进制以 1 或 2 字节块的形式显示。

12.8.2.3 DUAL PORT（双端口）（仅适用于 dsPIC33FJ DSC/PIC24HJ MCU 器件）

该格式将以十六进制数据的形式显示文件寄存器信息。窗口将具有以下列：

- Address—— 数据十六进制地址
- DMA Address（DMA 地址）—— 相对于硅片 DMA 地址的偏移量
- Data Blocks—— 十六进制数据，以 1 字节或 2 字节块的形式显示
- ASCII—— 相应数据行的 ASCII 表示形式

关于 dsPIC33F DSC 和 PIC24H MCU 器件的信息，请参见 Microchip 网站上的器件数据手册以及 dsPIC33F 和 PIC24H 参考手册章节。

12.8.2.4 XY DATA（XY 数据）（仅适用于 dsPIC DSC 器件）

该格式将以十六进制数据的形式显示文件寄存器信息。窗口将具有以下列：

- Address—— 数据的 X 总线十六进制地址
- Y Bus（Y 总线）—— 数据的 Y 总线十六进制地址（如支持）
- Data Blocks—— 十六进制数据，以 2 字节块的形式显示
- ASCII—— 相应数据行的 ASCII 表示形式

关于 dsPIC DSC 器件的更多信息，请参见“*dsPIC30F Family Reference Manual*”（DS70046）。

12.8.3 SFRs 窗口

特殊功能寄存器（SFRs）窗口会显示所选处理器的 SFR 的内容。对于查看 SFR，该窗口提供的格式会比正常文件寄存器窗口更有用，因为其中会包含每个 SFR 名称，并提供了几种数字格式。要仅查看几个 SFR，您可能会希望使用 Watches 窗口，该窗口可以帮助解决使用硬件调试工具时的速度问题（即，窗口更新速率更快）。

每次中断执行时，特殊功能寄存器的内容都会发生更新。

可见的寄存器

如果某个数据存储寄存器在器件上未物理实现，则它可能不会出现在 SFR 列表中。某些工具（如软件模拟器）可能会让您看到实际器件上不存在的寄存器，如预分频器。

单步运行

如果选择了“Freeze Peripherals On Halt”（暂停时冻结外设），则在单步运行时 SFR 中的 I/O 端口位或 Watches 窗口不会发生更新。引脚会被修改，但获取新值的读请求会由于冻结而被阻止，无法进行更新，直到下一个单步或运行命令为止。

- Individual（独立）
- Peripheral（外设）

12.8.3.1 INDIVIDUAL（独立）

在这种显示方式中，SFR 按地址列出。

数据显示在以下各列中。

- Address——SFR 十六进制地址
- Name——SFR 的符号名称
- Radix Information——Hex、Decimal 和 Binary
这三列中会显示基数信息。十六进制以 1 字节块的形式显示。

12.8.3.2 PERIPHERAL（外设）

在这种显示方式中，SFR 按其相关的外设分组。

数据显示在以下各列中：

- Address——SFR 十六进制地址
- Name—— 外设的名称或 SFR 的符号名称
- Radix Information——Hex、Decimal、Binary 和 Char
这四列中会显示基数信息。十六进制以 1 字节块的形式显示。

单击 **Select Peripheral**（选择外设）按钮可仅查看选定外设的 SFR。

12.8.4 Configuration Bits（配置位）窗口

第 4.21.4 节“设置配置位”介绍了关于使用 Configuration Bits 窗口的详细信息。数据显示在以下各列中。

表 12-10: 配置位列显示

列标题	定义
Address	配置字 / 字节的地址
Name	配置寄存器的名称
Value	配置字 / 字节的当前值
Field*（字段）	对于代码中设置的配置位，宏的字段部分 例如，WDTE 是宏 _WDTE_OFF 的字段部分。
Option*	对于代码中设置的配置位，宏的选项部分 例如，OFF 是宏 _WDTE_OFF 的选项部分。
Category	相应配置字 / 字节中的配置位的名称
Setting	配置位的当前设置。使用下拉列表可更改设置 配置字 / 字节的值将相应地发生改变。

* 并非所有器件都支持。

12.8.5 EE Data Memory（EE 数据存储）窗口

对于任何具有 EEPROM 数据存储器的单片机器件（如 PIC16F84A），EEPROM 窗口会显示 EEPROM 数据。其中会显示所选器件的数据 / 操作码十六进制信息。

对于 MPLAB X 软件模拟器，当 EEPROM 寄存器值发生改变或处理器暂停时，EEPROM 窗口中的数据会发生更新。

对于任何 Microchip 硬件调试工具（例如，MPLAB REAL ICE 在线仿真器），当某个 EEPROM 寄存器值发生改变或处理器暂停时，EEPROM 窗口中的数据不会发生更新；您必须对器件存储器执行读操作。

要与编程器配合使用，需要指定 EEPROM 数据存储器的起始地址。下表列出了一些通用值，但请查看您选择的器件的编程规范，以确定正确的地址。

表 12-11: 编程器 EEPROM 起始地址

器件	地址
大多数 PIC1X MCU	0x2100
PIC16F19XX MCU	0x1E000
PIC18F MCU	0xF00000
PIC24F MCU	0x7FFE00

这种显示格式会在以下各列中显示数据：

- Address——下一列中的数据十六进制地址
- Data Blocks——十六进制数据，以 1、2 或 4 字节块的形式显示，可从菜单中选择
- ASCII——相应数据行的 ASCII 表示形式

12.8.6 Other Memory（其他存储器）窗口

这是一个灵活的存储器窗口，允许您通过使用窗口底部的 Memory 下拉列表选择其他窗口。

12.8.7 Execution Memory（执行存储器）窗口

Execution Memory 窗口会显示处于当前所选 PIC32MX 器件的程序和 / 或数据存储单元范围内的存储单元。

对于 MPLAB X 软件模拟器，当执行存储器值发生更改或处理器暂停时，Execution Memory 窗口中的数据会发生更新。

对于任何 Microchip 硬件调试工具（例如，MPLAB REAL ICE 在线仿真器），当某个执行存储器值发生更改或处理器暂停时，Execution Memory 窗口中的数据不会发生更新；您必须对器件存储器执行读操作。

您可以通过单击窗口底部的选项卡之一来选择在窗口中显示的存储器类型：

- Code View - Program Memory（代码视图——程序存储器）
- Data View - Data Memory（数据视图——数据存储单元）

12.8.7.1 CODE VIEW - PROGRAM MEMORY（代码视图——程序存储器）

该格式将以十六进制代码的形式显示程序存储器信息。窗口将具有以下列：

- Line——对应于存储器地址的参考行号。
- Address——操作码的物理十六进制地址。
- Opcode——十六进制操作码，以 4 字节块的形式显示。高亮显示的操作码代表程序计数器的当前位置。
- Label——以符号格式显示的操作码标号。
- Disassembly——操作码助记符的反汇编版本。

12.8.7.2 DATA VIEW - DATA MEMORY（数据视图——数据存储单元）

该格式将以十六进制代码的形式显示数据存储单元信息。窗口将具有以下列：

- Address——下一列中的数据十六进制地址。
- Data Blocks——十六进制数据，以 4 字节块的形式显示。
- ASCII——相应数据行的 ASCII 表示形式。

12.8.8 Data Memory（数据存储单元）窗口

Data Memory 窗口会显示处于当前所选 PIC32MX 器件的数据和 / 或程序存储器范围内的存储单元。

您可以通过单击窗口底部的选项卡之一来选择在窗口中显示的存储器类型：

- DATA VIEW - DATA MEMORY（数据视图——数据存储单元）
- CODE VIEW - PROGRAM MEMORY（代码视图——程序存储器）

12.8.8.1 DATA VIEW - DATA MEMORY（数据视图——数据存储单元）

该格式将以十六进制代码的形式显示数据存储单元信息。窗口将具有以下列：

- Address——下一列中的数据十六进制地址。
- Data Blocks——十六进制数据，以 4 字节块的形式显示。
- ASCII——相应数据行的 ASCII 表示形式。

12.8.8.2 CODE VIEW - PROGRAM MEMORY（代码视图——程序存储器）

该格式将以十六进制代码的形式显示程序存储器信息。窗口将具有以下列：

- **Line**—— 对应于存储器地址的参考行号
- **Address**—— 操作码的物理十六进制地址
- **Opcode**—— 十六进制操作码，以 4 字节块的形式显示
高亮显示的操作码代表程序计数器的当前位置。
- **Label**—— 以符号格式显示的操作码标号
- **Disassembly**—— 操作码助记符的反汇编版本

12.8.9 Peripherals 窗口

Peripherals 窗口会显示与器件外设相关的 **SFR** 的内容。要仅查看几个 **SFR**，您可能会希望使用 **Watches** 窗口，该窗口可以帮助解决使用硬件调试工具时的速度问题（即，窗口更新速率更快）。

每次发生中断时，**SFR** 的内容都会发生更新。

可见的寄存器

如果某个数据存储寄存器寄存器在器件上未物理实现，则它可能不会出现在 **SFR** 列表中。某些工具（如软件模拟器）可能会让您看到实际器件上不存在的寄存器，如预分频器。

单步运行

如果选择了“**Freeze Peripherals On Halt**”，则在单步运行时 **SFR** 中的 I/O 端口位或 **Watches** 窗口不会发生更新。引脚会被修改，但获取新值的读请求会由于冻结而被阻止，无法进行更新，直到下一个单步或运行命令为止。

数据显示在以下各列中。

- **Address**——**SFR** 物理十六进制地址。
- **Virtual**（虚拟）—— 由总线矩阵定义的 **SFR** 虚拟十六进制地址。
- **Name**——**SFR** 的符号名称。
- **Radix Information**——**Hex**、**Decimal**、**Binary** 和 **Char**
您可以通过右键单击列标题栏来添加要显示的基数信息。十六进制以 4 字节块的形式显示。

12.8.10 Configuration Bits 窗口

与第 12.8.4 节“**Configuration Bits（配置位）窗口**”相同。

12.8.11 CPU Registers (CPU 寄存器) 窗口

CPU Registers 窗口会显示与器件 CPU 相关的 SFR 的内容。要仅查看几个 CPU SFR，您可能会希望使用 Watches 窗口，该窗口可以帮助解决使用硬件调试工具时的速度问题（即，窗口更新速率更快）。

每次中断执行时，CPU 寄存器的内容都会发生更新。

可见的寄存器

如果某个数据存储寄存器寄存器在器件上未物理实现，则它可能不会出现在 SFR 列表中。某些工具（如软件模拟器）可能会让您看到实际器件上不存在的寄存器，如预分频器。

单步运行

如果选择了“Freeze Peripherals On Halt”，则在单步运行时 SFR 中的 I/O 端口位或 Watches 窗口不会发生更新。引脚会被修改，但获取新值的读请求会由于冻结而被阻止，无法进行更新，直到下一个单步或运行命令为止。

数据显示在以下各列中。

- Address——SFR 物理十六进制地址。
- Name——SFR 的符号名称。
- Radix Information——Hex、Decimal、Binary、Char
您可以通过右键单击列标题栏来添加要显示的基数信息。
- Virtual——由总线矩阵定义的 SFR 虚拟十六进制地址。

12.8.12 User ID Memory (用户 ID 存储器) 窗口

一些器件具有可用于存储校验和或其他代码标识 (ID) 编号的存储单元。在编程 / 校验期间，这些存储单元是可读写的。根据不同的器件，它们可能还可以在正常执行期间通过 TBLRD 和 TBLWT 指令访问。

数据显示在以下各列中。

- Address——用户 ID 十六进制地址。单击右键可查看虚拟地址或物理地址。
- User ID——用户 ID 存储器的内容（以十六进制表示）。

要确定可在此处输入的值，请参见器件编程规范。对于大多数器件，它会设置器件 ID 字的低半字节；高半字节设置为“0”。高半字节只能通过编程方式写入，例如通过使用表写操作。

12.8.13 存储器窗口菜单

在存储器窗口单击右键将显示各种选项，如下所示。并非所有选项在所有窗口上都可用。

表 12-12: 存储器窗口菜单项

项	说明
Virtual Address (虚拟地址) Physical Address (物理地址)	显示在 Address (地址) 列下检查的地址的类型。
Hex Display Width (十六进制显示宽度)	设置十六进制显示宽度。选项取决于选定的器件。 32 位示例: One byte (一字节), 例如 00 01 02 ... 0E 0F Two bytes (二字节), 例如 00 02 04 ... 0C 0E Four bytes (四字节), 例如 00 04 08 0C
Run to Cursor	运行程序到当前光标位置。
Set PC at Cursor	将程序计数器 (PC) 设置为光标位置。
Focus Cursor at PC	将光标移到当前 PC 地址, 并将当前 PC 地址在窗口中居中。
Toggle Breakpoint	翻转 (开启 / 关闭) 现有断点。
Symbolic Mode (符号模式)	显示反汇编的十六进制代码和符号。
Verbose Labels (详细标签)	显示内部编译器标签。
Fill Memory	使用 Data 中的值从 Start Address (起始地址) 到 End Address (结束地址) 填充存储器。在 Fill Memory 对话框中指定其他选项。
Go To	转至在 Go To 对话框中指定的地址 / 函数。
Find	查找在 Find 对话框中指定的文本。
Enable Multiline Rows (使能多行)	允许 Configuration Bits 窗口中存在多行。
Output To File (输出到文件)	将所显示的窗口内容写入一个文本文件中。在 Output to File Range (输出到文件的范围) 对话框指定要输出的数据范围。
Import Table (导入表)	将表格式数据从文件导入存储器窗口。在 Import Table Range (导入表范围) 对话框中指定要导入的数据范围。
Export Table (导出表)	将表格式数据从存储器窗口导出到文件中。在 Export Table Range (导出表范围) 对话框中指定要导出的数据范围。此外, 指定导出是否采用单列格式。
Print	打印该窗口的内容。 注: 对于具有大容量存储器的器件, 打印的页数可能会很多。这种情况下建议将窗口内容打印到一个文件中 (Print 对话框, General 选项卡, “Print to File” (打印到文件) 复选框), 然后从该文件中选择需要打印哪些页。
Adjust Table Columns (调整表中的列)	自动调整列。

右键单击窗口选项卡将显示其他选项, 例如 Close Window、Maximize Window/Minimize Window 以及 Dock Window/Undock Window。

12.9 OUTPUT 窗口

Tasks 窗格中包含许多窗口，一些是从 NetBeans 继承的，一些是特定于 MPLAB X IDE 的。Output 窗口中包含 MPLAB X IDE 输出信息。它显示在窗口内的选项卡上。

表 12-13: OUTPUT 窗口选项卡项

项	说明
Debugger Console (调试器控制台)	显示主要的调试操作，例如 “User program running” (用户程序运行)。
Tool-specific (工具特定信息)	显示工具的固件版本、器件 ID 和操作状态。
Build, Load (编译, 装入)	显示关于编译和程序装入的信息和状态。
Clean, Build, Load (清除, 编译, 装入)	显示关于清除、编译和程序装入的信息和状态。
Peripheral Output (外设输出)	显示使用软件模拟器作为调试工具时来自外设 (例如 UART) 的技术输出。

在 Output 窗口单击右键将显示各种选项，如下所示。

表 12-14: OUTPUT 窗口菜单项

菜单项	说明
Copy	将选定文本从 Output 窗口复制到剪贴板。
Paste	将选定文本从剪贴板复制到 Output 窗口。
Find	在 Output 窗口中查找选定的文本 (或输入要查找的其他文本)。您可以使用正则表达式和匹配大小写。
Find Next	查找 Find 文本的下一个匹配项。
Find Previous	查找 Find 文本的上一个匹配项。
Filter	按文本或正则表达式过滤输出。
Wrap Text (文本折行)	对 Output 窗口中的文本进行折行。
Larger Font (较大的字体)	使字体变大。
Smaller Font (较小的字体)	使字体变小。
Choose Font (选择字体)	选择字体类型、样式和大小。
Save As	将选定文本保存到一个文件中。
Clear	清除 Output 窗口选项卡中的所有文本。
Close	关闭 Output 窗口。

12.10 PROJECT PROPERTIES 窗口

该窗口用于查看或更改项目器件、工具和工具设置。更多信息，请参见：

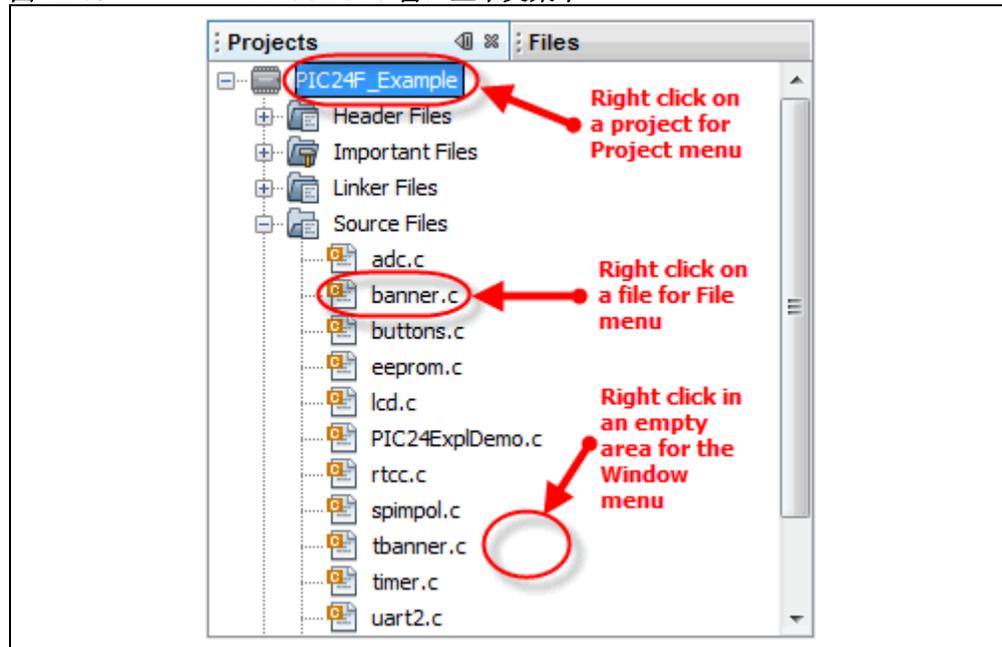
- 第 4.4 节 “查看或更改项目属性”
- 第 4.5 节 “设置调试器、编程器或语言工具的选项”
- 第 5.4 节 “可装入项目和文件”
- 第 4.11 节 “添加、设置库和目标文件”
- 第 4.13 节 “设置编译属性”

12.11 PROJECTS 窗口

Projects 窗口是一个 NetBeans 窗口。但是，它被定制为显示 MPLAB X IDE 项目的相关逻辑（虚拟）文件夹。此外，上下文（右键）菜单也被定制为带有一些 MPLAB X IDE 特定项。

- Projects 窗口 —— 逻辑文件夹
- Projects 窗口 —— 项目菜单
- Projects 窗口 —— 文件菜单
- Projects 窗口 —— 窗口菜单

图 12-3: PROJECTS 窗口上下文菜单



12.11.1 Projects 窗口 —— 逻辑文件夹

添加了适用于 MPLAB X 项目的额外逻辑文件夹。因此，显示的所有文件夹可以归类为如下文件类型之一：

- **Header Files (头文件)** ——MPLAB X IDE 在编译时不使用该类别。可以将它视为记录项目对于头文件的依赖性的方式，以及一种访问这些文件的简便方法。在 Projects 窗口中双击某个文件，可以在编辑器中打开该文件。
- **Linker Files (链接器文件)** —— 您不需要向项目中添加链接描述文件，因为项目语言工具会查找对应于器件的相应通用链接描述文件。因此，该文件夹将为空，除非您自己创建了链接描述文件。
该文件夹中应只有一个文件。如果具有多个链接描述文件，只有第一个会起作用——它是工具在链接步骤中使用的链接描述文件。
- **Source Files (源文件)** —— 它们是工具链唯一接受作为其文件命令输入的文件。
- **Important Files (重要文件)** —— 不属于任何其他类别的所有文件最终会放入该文件夹中。
您可以添加模拟器文件 (SBS、STC 和 SCL) 并双击它们来分别打开 SCL 发生器、激励控制器或编辑器。
您可以在 Projects 窗口中向该位置添加特定于项目的数据手册 (PDF 文件)。然后，您可以双击 PDF 文件来打开数据手册。(这要求已安装 PDF 阅读器。)
- **Libraries**—— 工具链应获取该文件夹中的所有文件，以及目标文件，并将它们包含在最终链接步骤中。更多信息，请参见第 4.11 节“添加、设置库和目标文件”。
- **Loadables**—— 要与项目十六进制文件进行合并或替换项目十六进制文件的项目和文件。更多信息，请参见第 5.4 节“可装入项目和文件”。

12.11.2 Projects 窗口 —— 项目菜单

在 Projects 窗口中右键单击某个项目名称可弹出项目菜单。菜单项如表 12-15 所示。

表 12-15: 项目上下文菜单项

菜单项	说明
New	添加新项。对于 MPLAB® X IDE，提供了嵌入式文件类型。
Add Existing Item	向项目中添加现有文件。文件的路径可以是自动（由 MPLAB X IDE 选择）、绝对或相对的。此外，您还可以选择将文件复制到项目文件夹中。
Add Existing Items from Folders（从文件夹中添加现有项）	添加指定文件夹中包含的所有文件。您可以通过指定模式来忽略某些文件夹。单击 Default （默认）可查看默认模式。另请参见 <i>Tools>Options, Miscellaneous</i> （其他）按钮， Files 选项卡，“Files ignored by the IDE”（被 IDE 忽略的文件）。
New Logical Folder（新建逻辑文件夹）	创建新的逻辑文件夹。要查看实际文件夹，请参见 Files 窗口。
Locate Headers	查找在 C 代码项目文件中调用的所有头文件（.h），并在列表窗口中显示它们，从而可以将它们添加到项目中。由于几种原因，您可能会希望向项目中添加头文件： <ul style="list-style-type: none"> • 可以从 Projects 窗口中的“Header Files”（头文件）下打开这些文件，而不需要在您的计算机上搜寻它们。 • 通过“Save Project As”（将项目另存为）创建的项目将不包含头文件，除非将它们添加到项目中；因此，如果项目需要任何用户生成的头文件，则可能无法编译。 • 通过“Package Project as .zip”（将项目打包为 .zip）创建的压缩项目将不包含头文件，除非将它们添加到项目中；因此，如果解压后的项目需要任何用户生成的头文件，则可能无法编译。该功能能够找到所有 Microchip 工具链的用户生成的头文件。
Add Item to Important Files（将项添加到重要文件中）	向项目中添加现有项。文件的路径可以是自动（由 MPLAB X IDE 选择）、绝对或相对的。此外，您还可以选择将文件复制到项目文件夹中。这对于向项目中添加模拟器文件、数据手册 PDF 和其他文件供以参考会非常有用。
Export Hex	将项目编译导出为十六进制文件。在 MPLAB IDE v8 中，导出是对存储器对象进行提取。在 MPLAB X IDE 中，导出是编译的十六进制文件结果；因此，需要在代码中包含配置和 EEPROM 的所有必需的辅助存储器设置。
Build	根据在 Project Properties 窗口中选择的选项编译项目。 注： 当执行调试或运行时，将会自动编译项目。
Clean and Build	根据在 Project Properties 窗口中选择的选项清除并编译项目。 注： 当执行调试或运行时，将会自动编译项目。
Clean	通过删除先前编译的输出来清除项目。

表 12-15: 项目上下文菜单项 (续)

菜单项	说明
Package (包)	将当前项目打包为 .zip 文件。虽然 MPLAB X IDE 具有打包为 zip 文件的功能，但它无法解压缩它们。所以需要有一个单独的程序来解压缩该项目。 为了打包项目，该功能会检查项目文件，以确定要包含的项目文件的位置。将仅包含在项目文件夹中包含并且使用相对而不是绝对路径的文件。 项目打包文件中包含源代码文件、makefile 和 nbproject 目录。
Set Configuration (设置配置)	打开 Project Properties 对话框，使您可以设置项目配置。
Run	执行项目代码。详情请参见第 4.15 节“运行代码”。
Debug	在调试环境中执行项目代码。详情请参见第 4.16 节“调试运行代码”。
Step In (单步运行)	逐句执行在调试环境中运行的暂停代码。详情请参见第 4.18 节“单步执行代码”。
Make and Program Device	对目标器件编程并保持在复位状态 (不运行)。
Set as Main Project	将该项目设置为主项目。在处理多个项目时非常有用。另请参见第 6.3 节“处理多个项目”。
Open Required Projects	装入运行选定项目所需的所有其他项目。
Close	关闭所选的项目。
Rename	重命名项目。
Move	将项目移动到另一个位置。随项目一起移动项目目录内的源文件。项目目录外的文件不会被移动。但是，项目将仍然保持对项目目录外的文件的引用。这是由设计决定的。
Copy	创建项目的副本。如果源文件处于项目目录内，则源文件也会被复制到新位置。项目目录外的文件不会被复制。但是，项目将仍然保持对项目目录外的文件的引用。这是由设计决定的。
Delete	删除项目文件。项目文件会被删除，但不删除项目目录下的内容。只有在选择“Also delete sources”（同时删除源文件）时才会删除所有源文件。
Code Assistance (代码辅助)	选择在创建代码时的辅助功能：代码折叠、代码补全等。
Find	在该项目的文件中查找指定文本。
Share on Team Server (在团队服务器上共享)	在团队服务器上共享该项目。详情请参见第 5.20 节“在代码开发和错误跟踪方面进行协作”。
Versioning (版本控制)	通过使用版本控制系统来控制该项目的版本。详情请参见第 5.19 节“控制源代码”。
Local History (本地历史记录)	查看或恢复为该项目的本地历史记录。详情请参见第 5.19 节“控制源代码”。
Properties	设置项目属性。对于 MPLAB X IDE，Project Properties 窗口是特定于嵌入式开发的。请参见第 12.10 节“Project Properties 窗口”。

12.11.3 Projects 窗口 —— 文件菜单

在 Projects 窗口中右键单击文件名可弹出文件菜单。表 12-16 列出了特定的菜单项。

表 12-16: 文件上下文菜单项

菜单项	说明
Open	在选项卡形式的编辑器窗口中打开该文件。
Cut	从项目中删除该文件，但将它的副本放到剪贴板上。
Copy	在剪贴板上放置文件的一个副本。
Paste	将文件的剪贴板副本粘贴到项目中。
Compile File	仅编译该文件。
Remove from Project	从项目中移除该文件。该操作不会从 PC 中删除文件。要从项目和计算机中删除文件，请使用 Delete 键。
Rename	重命名文件。
Save as Template (另存为模板)	将该文件保存为模板文件。
Local History	查看或恢复为该文件的本地历史记录。详情请参见第 5.19 节“控制源代码”。
Tools	文件工具包括： <ul style="list-style-type: none"> • Apply Diff Patch: 应用由 Diff 创建的现有补丁。 • Diff to (比较差异): 显示该文件和此处指定的一个文件之间的差异。 • Add to Favorites: 将该文件添加到 Favorites 窗口。
Properties	设置不同于项目属性的文件属性。通过此菜单项可以选择从编译中排除该文件或通过选择不同配置来覆盖项目编译选项。详情请参见第 4.12 节“设置文件和文件夹属性”。

12.11.4 Projects 窗口 —— 窗口菜单

在 Projects 窗口中的空白区域单击右键可弹出窗口菜单。表 12-17 列出了特定的菜单项。

表 12-17: 文件上下文菜单项

菜单项	说明
New Project	启动 New Project 向导。更多信息，请参见第 4.2 节“创建新项目”。
New File	启动 New File 向导。更多信息，请参见第 4.8 节“创建新文件”。
Open Project	打开现有项目。
Open Recent Project	打开最近项目列表中的现有项目。
Open Project Group (打开项目组)	打开项目组列表中的现有项目组。
Run Project	运行主项目。
Set Main Project	从已打开项目的列表中设置主项目。

12.12 TOOLS OPTIONS EMBEDDED 窗口

使用 Tools>Options（对于 Mac OS X 为 mplab_ide>Preferences）打开该窗口。

Options 窗口是一个 NetBeans 窗口。但是，它通过增加 Embedded 按钮针对 MPLAB X IDE 项目进行了定制。单击该按钮之后，以下选项卡和选项将可用。

- Build Tools 选项卡
- Project Options 选项卡
- Generic Settings 选项卡
- Suppressible Messages 选项卡
- Diagnostics 选项卡
- Other 选项卡

12.12.1 Build Tools 选项卡

对于 Mac 计算机，该选项卡上的信息将以不同的方式访问。从主菜单条中的 [mplab_ide>/preferences](#) 访问编译工具。请参见第 3.3.5 节“设置语言工具位置”。

请确保您已安装了语言工具，否则它不会显示在“Toolchain”（工具链）列表上。如果知道已安装了该工具，但它不处于列表上，请单击 **Scan for compilers**（扫描编译器）。如果仍未找到，请单击 **Add** 将该工具添加到列表中。

MPLAB X IDE 随附了以下语言工具：

- MPASM 工具链 —— 包括 MPASM 汇编器、MPLINK 链接器和实用程序。

其他工具可从 Microchip 网站（www.microchip.com）或第三方获取。

表 12-18: BUILD TOOLS 选项卡项

项	说明
Toolchain	显示计算机上已安装的语言工具的列表。为您的项目选择工具并确保（应用于）右侧的路径是正确的。 Base directory（基目录） ：指向工具主文件夹的路径 C compiler（C 编译器） ：指向 C 编译器的路径（如可用） Assembler ：指向汇编器的路径（如可用） Make command（Make 命令） ：由 MPLAB [®] X IDE 生成的 make 命令的名称。
Add	向列表中添加新的语言工具项。此外，还可以考虑使用 Scan for Build Tools 。
Remove	从列表中删除语言工具项。这并不会从计算机中删除语言工具。
Default（默认）	单击某个工具，然后单击 Default 可将该工具设为选定器件的默认编译器 / 汇编器。
Scan for Build Tools	在计算机中的各种默认位置而 不是 整个系统中扫描已安装编译器 / 汇编器。如果安装在其他位置，则请手动添加编译器。

12.12.2 Project Options 选项卡

设置与项目有关的选项。

表 12-19: PROJECT OPTIONS 选项卡项

项	说明
Make Options (Make 选项)	输入在编译项目时使用的 make 选项。这些选项依赖于工具链。请参见语言工具文档。
File Path Mode (文件路径模式)	指定在项目中如何存储文件路径信息。 Auto: 项目文件夹内的文件的路径以相对方式存储; 项目文件夹外的文件的路径以绝对方式存储。 Always Relative (总是相对): 所有路径都以相对于项目文件夹的方式存储。 Always Absolute (总是绝对): 所有路径都以绝对 (完整路径) 方式存储。
Save All Modified Files Before Running Make (运行 Make 之前保存所有修改过的文件)	如果选中, 将在运行 make 之前保存 IDE 中所有未保存的文件。建议将该属性保留为选定, 因为除非先将对 IDE 中文件的修改保存到磁盘中, 否则 make 将无法识别到这些修改。
Show binary files in Project view (在项目视图中显示二进制文件)	如果选中, Projects 视图将显示目录树中包括二进制对象在内的所有文件。与该选项关系最大的是使用现有源文件创建的项目, 这些项目可能会将源文件和二进制文件放在同一位置。取消选择该选项可以更简便地查看 C 源文件和头文件。
Show profiler indicators during run (new projects only) (运行期间显示性能分析器指示器 (仅对于新项目))	如果选中, 性能分析器工具 (例如 CPU Usage 和 Memory Usage) 将设置为在新创建项目运行时在默认情况下运行。所显示的工具由在 Tools>Profiler Tools (性能分析器工具) 中选择的性能分析配置决定。
Use parallel make (使用并行 make) (make -j 2n)	如果选中, make 将每次执行多个进程, 其中的 -j (或 --jobs) 是要并行运行的选项, 2n 是进程数量, 其中的 n 是计算机上可用处理器的数量。如果您的计算机不支持并行处理, 则会禁止并行 make。 如果愿意, 可以通过使用 “Make Options” 指定更多的进程。 示例: -j 10。 注 1: 对于 MPLAB XC16 或 MPLAB C30, 需要关闭过程抽象才能使用并行 make (File>Project Properties , 编译器类别, “Optimizations” (优化) 选项类别: 取消选中 “Unlimited procedural abstraction” (无限的过程抽象))。 注 2: MPASM 无法在并行 make 下运行, 无论是作为工具链还是作为 MPLAB C18 项目的一部分。因此, 在使用 MPASM 工具链的项目中或在使用包含至少一个 .asm 文件的 C18 工具链的项目中, 会忽略并行 make 选项。

12.12.3 Generic Settings 选项卡

设置日志文件和其他项目功能。

表 12-20: GENERIC SETTINGS 选项卡项

项	说明
Projects Folder (项目文件夹)	放置 MPLAB [®] X IDE 项目的文件夹的路径和名称。
Close open source file on project close (在项目关闭时关闭已打开的源文件)	在关闭项目时, 关闭编辑器窗口中所有已打开的源文件。
Clear output window before build (编译之前清除输出窗口)	在编译项目时清除 Output 窗口的内容。
Remove breakpoints upon importing a file (导入文件时删除断点)	向项目中导入某个文件时, 删除所有现有的项目断点。
Halt build on first failure (在第一次失败时暂停编译)	在进行编译时, 在发生第一次失败时暂停该过程。可以通过设置选定项目语言工具来确定产生哪些错误。转到 Project Properties 对话框中, 并在 “Categories” 下选择语言工具。然后, 在 Option Categories (选项类别) 中进行查找, 找到一个允许您设置错误、警告和 / 或消息的类别。
Maintain active connection to hardware tool	如果选中, 则将总是保持与硬件工具连接, 而不只是在运行时 (MPLAB [®] IDE v8 的行为)。在选择该选项的情况下切换项目时 (例如, 在开发自举应用程序时), 请确保工具和器件是相同的, 以避免错误消息。
Clear tool output window on new session (debug, program, upload) (在新会话 (调试、编程和上传) 时清除工具输出窗口)	在开始运行、调试运行或上载时清除 Output 窗口的内容。
Silent build (静默编译)	进行编译, 但不在 Output 窗口中产生消息。
Enable alternate watch list views during debug session (在调试会话期间使能备用观察列表视图)	在 Watches 窗口中显示 3 个观察视图菱形。将观察视图与观察变量相关联。在单击观察视图菱形时, 将仅显示与该视图相关联的变量。所以, 该功能的工作方式类似于一个过滤器。
Reset @	选择复位时的操作。 Main: 复位时在 main() 处停止 Reset Vector (复位向量): 复位时在复位向量处停止
Debug start-up (调试启动)	选择调试启动时的操作。 Run (运行): 立即开始执行 Main: 停止在 main() 处 Reset Vector: 停止在复位向量处
Default Charset (默认字符集)	选择项目的默认字符集。

12.12.4 Suppressible Messages 选项卡

选择要禁止的错误和警告消息。双击可用的文件夹，可以进一步选择可用项来进行禁止。

12.12.5 Diagnostics 选项卡

设置日志文件和其他诊断功能。请参见第 6.6 节“日志数据”。

表 12-21: DIAGNOSTICS 选项卡项

项	说明
Logging Level (日志记录级别)	设置消息的日志记录级别。 OFF (关闭) : 无日志记录 SEVERE (严重) : 仅记录严重 (错误) 消息 WARNING (警告) : 仅记录警告消息 INFO (信息) : 仅记录信息性的消息 CONFIG (配置) : 仅记录配置信息 FINE (略详细) : 记录一些模块间通信 FINER (较详细) : 记录更多的模块间通信 FINEST (最详细) : 记录所有模块间通信
Log File (日志文件)	日志文件的路径和名称。
USB Circular Log (USB 循环日志)	仅适用于 MPLAB REAL ICE 在线仿真器、MPLAB ICD 3 和 PICkit 3。
Start New Logging Session/Pause Logging (启动新的日志记录会话 / 暂停日志记录)	单击按钮可开始一个新的日志记录会话或暂停正在进行的日志记录会话。
Circular USB log file location (循环 USB 日志文件位置)	指定日志文件的路径。
Circular USB log file max size in KBs (以 KB 表示的循环 USB 日志文件最大大小)	指定日志文件的大小。

12.12.6 Other 选项卡

编辑 C/C++ 和汇编源文件以及头文件的可接受文件扩展名的列表。此外，设置每种类型的默认扩展名。

注: MPLAB X IDE 不支持 Fortran 编程。

12.13 TRACE 窗口

跟踪使您可以记录代码的分步执行并检查该记录。跟踪当前可用于以下工具：

- 软件模拟器
- MPLAB REAL ICE 在线仿真器

在窗口中右键单击某个跟踪列可弹出上下文菜单（表 12-22）。根据您正在使用的工具，可能会也可能不会看到所有菜单项。

表 12-23 定义了与跟踪关联的对话框。

表 12-22: TRACE 窗口上下文菜单

菜单项	说明
Symbolic Mode	对于“Instruction”（指令）列，在显示立即数寄存器地址（例如 0x5）和符号寄存器宏（例如 PORTA）之间切换。
Go To	Trigger（触发） ：移至触发行（0）。 Top（顶部） ：移至顶部跟踪行。 Bottom（底部） ：移至底部跟踪行。 Trace Line（跟踪行） ：指定并转至跟踪行位置。打开 Go To 对话框。
Go To Source Line (转至源代码行)	选择跟踪行，然后选择该选项可转至源代码中的相应行。
Display Time (显示时间)	对于“Cycle”（周期）列（如果先前不可见，则将显示）： As Hex Cycle Count（十六进制周期计数） ：以十六进制形式显示周期计数 As Decimal Cycle Count（十进制周期计数） ：以十进制形式显示周期计数 In Seconds Elapsed（经过的秒数） ：以经过的秒数的形式显示周期计数 In Engineering Format（工程格式） ：以相应的工程格式（10 ³ 的幂）显示周期计数。
Clear Trace File (清除跟踪文件)	清除跟踪显示中的数据。
Find	在跟踪显示中查找项。打开 Find 对话框。
Output To File	将跟踪数据保存到一个文件中。打开 Define Range（定义范围）对话框，它将打开 Save 对话框。
Print	打印数据。打开 Print 对话框。
Adjust Table Columns	自动调整跟踪显示中的列，使之适应数据。
Reload View (重新装入视图)	在暂停时重新装入跟踪显示的原始数据视图。

表 12-23: 跟踪对话框

对话框	说明
Go To	指定要转至的跟踪行。
Find	在跟踪显示中查找行号或其他数据。
Output to File Range	指定要输出到文件的行范围。单击 OK 可转至 Save 对话框，从而将数据保存到一个文本文件中。
Save	将数据保存到一个文本文件中。
Print	在打印之前指定打印机、页面设置和外观。

12.14 WATCHES 窗口

使用 Watches 窗口可观察选择更改的符号的值。要打开 Watches 窗口，请选择 [Window>Debugging>Watches](#)。

12.14.1 Watches 操作

关于使用 Watches 窗口的信息，请参见：

- 第 4.19 节 “观察符号值变化”
- NetBeans 帮助主题 [C/C++/Fortran Development>Debugging C/C++/Fortran Applications with gdb>Viewing C/C++/Fortran Program Information>Creating a C/C++/Fortran Watch](#)。

12.14.2 Watches 菜单

右键单击一行可打开 Watches 菜单。可用项将取决于行中是否存在符号或您正在使用的调试工具。

表 12-24: WATCHES 窗口菜单项

菜单项	说明
New Watch	添加要观察的新符号
New Runtime Watch	仅对于 MPLAB REAL ICE 在线仿真器为选定符号添加新的运行时观察。
Run Time Update Interval (运行时更新间隔)	仅对于 MPLAB REAL ICE 在线仿真器指定符号值的更新速率。“No Delay”（无延时）将以个人计算机支持的速率尽可能快地进行更新。如果在运行时数据中发现错误，可以通过增加延时来降低更新速度。
Export All Watches to List File (将所有观察导出到列表文件)	将关于要观察的符号的信息导出到一个文件（.xwatch）中。
Delete All (全部删除)	从 Watches 窗口中删除观察的所有符号。

注： 此外，还可以在代码中右键单击某个符号，将其添加到 New Watch 或 New Runtime Watch 中。

12.14.3 Watches 显示

该显示窗口具有以下特性：

- 图标
- 列
- 操作

12.14.3.1 图标

图标位于名称列中的名称对象的左侧：

表 12-25: 名称列图标

图标	说明
	观察对象
	程序存储器中的观察对象
	对象的静态字段
	对象的非静态字段

12.14.3.2 列

您可以通过在窗口中右键单击某个标题，弹出“Change Visible Columns”（更改可视列）对话框来更改所显示的列。

表 12-26: CHANGE VISIBLE COLUMNS 项

对话框项	如果选中
Name	显示列的名称（总是选中）
Address	显示变量的存储器地址
Binary	显示二进制格式值
Char	显示字符格式值
Decimal	显示十进制格式值
Type（类型）	显示观察变量的类型
Value	显示观察表达式的值（以十六进制表示）

12.14.3.3 操作

操作位于窗口左侧的按钮上：

表 12-27: 名称列图标

按钮	操作
	切换按钮： - 显示成员字段的详细（限定）名称。 - 显示成员字段的简要（相对）名称。
	从列表文件中导入观察。单击右键可查看选项。
	将所有观察导出到列表文件。
	设置 Value 字段的默认数字格式。

12.15 向导窗口

MPLAB X IDE 使用了许多 NetBeans 窗口。但是，一些窗口是特别针对嵌入式使用而修改或创建的。

- New Project 向导 —— 请参见第 4.2 节 “创建新项目”
- New File 向导 —— 请参见第 4.8 节 “创建新文件”
- Import Image File 向导 —— 请参见第 5.3.3 节 “Import Image File 向导”

第 13 章 NetBeans 窗口和对话框

13.1 简介

MPLAB IDE 窗口是基本 NetBeans 窗口和 MPLAB IDE 特定窗口的组合。从菜单项中选择时，将打开对话框。与窗口一样，MPLAB IDE 对话框是基本 NetBeans 对话框和 MPLAB IDE 特定对话框的组合。

本章将介绍 NetBeans 窗口和对话框，并引用帮助中的文档。关于 MPLAB X IDE 特定窗口和对话框的信息，请参见第 12 章 “MPLAB X IDE 窗口和对话框”。

- NetBeans 特定窗口和窗口菜单
- NetBeans 特定对话框

13.2 NETBEANS 特定窗口和窗口菜单

NetBeans 帮助中对 NetBeans 窗口进行了介绍。要打开关于某个窗口的帮助，请单击 **Window** 选项卡选择它，然后按 <F1> 键。如果未找到任何帮助，可以选择 [Help>Help Contents](#) 并单击帮助文件的 **Search** 选项卡来搜索关于该窗口的信息。或者，请参见 NetBeans 帮助主题 “Managing IDE Windows (管理 IDE 窗口)”。

要打开大多数窗口，请参见第 11.2.11 节 “Window 菜单”。

窗口可以进行停靠和取消停靠（右键单击 **Window** 选项卡），并具有特定于窗口的弹出或上下文菜单，菜单中带有 **Fill**（填充）、**Goto**（转至）和 **Edit Cells**（编辑单元格）之类的菜单项。在窗口中或窗口中某个项上单击右键将弹出一个上下文菜单。大多数内容菜单项也会位于桌面菜单条的菜单上（请参见第 11.2 节 “菜单”）。

通过选择 [Tools>Options](#)（对于 Mac OS X 为 [mplab_ide>Preferences](#)），**Miscellaneous** 按钮，**Appearance**（外观）选项卡来设置窗口属性。

13.3 NETBEANS 特定对话框

NetBeans 帮助中对 NetBeans 对话框进行了介绍。要打开关于某个对话框的帮助，请单击对话框上的 **Help** 按钮，或者如果没有 **Help** 按钮，则按 <F1> 键。如果未找到任何帮助，可以选择 [Help>Help Contents](#) 并单击帮助文件的 **Search** 选项卡来搜索关于该对话框的信息。

要打开大多数对话框，请参见第 11.2 节 “菜单”。

注:

附录 A 配置设置汇总

A.1 简介

以下给出了说明如何在代码中为不同语言工具和相关器件设置配置位的示例。关于如何设置配置位的更多信息，请参见语言工具文档。对于一些语言工具，提供了配置设置文档，其中会列出器件的所有配置设置。其他情况下，请参考您的器件头文件，查看其中的宏。

另一个选项是使用 Configuration Memory 窗口来设置配置位，然后单击“Generate Source Code to Output”。请参见第 4.21.4 节“设置配置位”。

- MPASM 工具链
- HI-TECH® PICC™ 工具链
- HI-TECH® PICC-18™ 工具链
- C18 工具链
- ASM30 工具链
- C30 工具链
- C32 工具链
- XC 工具链

A.2 MPASM 工具链

使用两种类型的汇编器伪指令来设置器件配置位：__config 和 config。不要在同一代码中混合使用 __config 和 config。

A.2.1 __config

伪指令 __config 用于 PIC10/12/16 MCU。它可用于 PIC18 MCU（不包括 PIC18FXXJ 器件），但是建议使用 config 伪指令。语法如下所示：

```
__config expr ;For a single configuration word
```

或

```
__config addr, expr ;For multiple configuration word
```

其中：

addr：配置字的地址。可以为立即数，但通常通过一个宏表示。

注：宏必须按寄存器升序列出。

expr：代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件包含文件 (*.inc) 中指定，器件包含文件位于 Windows 操作系统 (OS) 默认目录中：

```
C:\Program Files\Microchip\MPLABX\mpasmx
```

伪指令的大小写无关紧要；__CONFIG 或 __config 均可接受。宏的大小写应与头文件中匹配。

示例——PIC10/12/16 MCU

```
#include p16f877a.inc

;Set oscillator to HS, watchdog time off, low-voltage prog. off
__CONFIG _HS_OSC & _WDT_OFF & _LVP_OFF
```

示例——PIC18 MCU

```
#include p18f452.inc

;Oscillator switch enabled, RC oscillator with OSC2 as I/O pin.
__CONFIG _CONFIG1, _OSCS_OFF_1 & _RCIO_OSC_1

;Watch Dog Timer enable, Watch Dog Timer PostScaler count - 1:128
__CONFIG _CONFIG3, _WDT_ON_3 & _WDTPS_128_3
```

A.2.2 config

伪指令 `config` 用于 PIC18 MCU（包括 PIC18FXXJ 器件）。语法如下所示：

```
config setting=value [, setting=value]
```

其中：

setting: 代表一个或多个配置位的宏。

value: 代表指定配置位设置值的宏。多个设置可以在单个行上定义，使用逗号分隔。单个配置字节的设置也可以在多个行上定义。

宏在器件包含文件 (*.inc) 中指定，器件包含文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\Microchip\MPLABX\mpasmx
```

伪指令的大小写无关紧要；`__CONFIG` 或 `__config` 均可接受。宏的大小写应与头文件中匹配。

示例——PIC18 MCU

```
#include p18f452.inc

;Oscillator switch enabled, RC oscillator with OSC2 as I/O pin.
CONFIG _OSCS=ON, OSC=LP

;Watch Dog Timer enable, Watch Dog Timer PostScaler count - 1:128
CONFIG _WDT=ON, WDTPS=128
```

A.3 HI-TECH® PICC™ 工具链

使用在 `htc.h` 头文件中指定的宏来设置 PIC10/12/16 MCU 器件的配置字：

```
__CONFIG(x);
```

其中：

`x`：代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件头文件（*.h）中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\HI-TECH Software\PICC\version\include
```

其中，`version` 是编译器的版本号。

对于具有多个配置字单元的器件，随后每次调用 `__CONFIG()` 都将按顺序修改下一个配置字。

宏的大小写应与相关头文件中匹配。对于 `htc.c`，`__CONFIG()` 是正确的，但 `__config()` 不正确。

PICC 示例

```
#include <htc.h>

__CONFIG(WDTDIS & XT & UNPROTECT); // Program config. word 1
__CONFIG(FCMEN); // Program config. word 2
```

A.4 HI-TECH® PICC-18™ 工具链

使用在 `htc.h` 头文件中指定的宏来设置 PIC18 MCU 器件的配置字：

```
__CONFIG(n,x);
```

其中：

`n`：配置寄存器编号

`x`：代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件头文件（*.h）中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\HI-TECH Software\PICC\version\include
```

其中，`version` 是编译器的版本号。

宏的大小写应与相关头文件中匹配。对于 `htc.c`，`__CONFIG()` 是正确的，但 `__config()` 不正确。

PICC-18 示例

```
#include <htc.h>

//Oscillator switch enabled, RC oscillator with OSC2 as I/O pin.
__CONFIG(1, LP);

//Watch Dog Timer enable, Watch Dog Timer PostScaler count - 1:128
__CONFIG(2, WDTEN & WDTPS128);
```

A.5 C18 工具链

`#pragma config` 伪指令指定要由应用程序使用的特定于器件的配置设置（即，配置位）：

```
#pragma config setting-list
```

其中：

setting-list: 一个或多个 *setting-name = value-name* 宏对的列表，使用逗号分隔。

宏在器件头文件（*.h）中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\program files\microchip\mplabc18\version\.h
```

`Pragma` 伪指令的大小写无关紧要；`#PRAGMA CONFIG` 或 `#pragma config` 均可接受。宏的大小写应与头文件中匹配。

示例

```
#include <p18cxxx.h>

/*Oscillator switch enabled, RC oscillator with OSC2 as I/O pin.*/
#pragma config OSCS = ON, OSC = LP

/*Watch Dog Timer enable, Watch Dog Timer PostScaler count - 1:128*/
#pragma config WDT = ON, WDTPS = 128
```

A.6 ASM30 工具链

使用在器件包含文件中指定的宏来设置配置位：

```
config __reg, value
```

其中：

__reg: 配置寄存器名称宏。

value: 代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件包含文件（*.inc）中指定，器件包含文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\Microchip\MPLAB ASM30 Suite\Support\device\inc
```

其中，*device* 是选定 16 位器件的缩写，如 PIC24H 或 dsPIC33F。

宏的大小写应与相关头文件中匹配。例如，`config` 是正确的，但 `CONFIG` 不正确。

示例

```
.include "p30fxxxx.inc"

;Clock switching off, Fail-safe clock monitoring off,
; Use External Clock
config __FOSC, CSW_FSCM_OFF & XT_PLL16

;Turn off Watchdog Timer
config __FWDT, WDT_OFF
```

A.7 C30 工具链

使用两种类型的编译器宏来设置器件配置位：一种类型用于 PIC24F MCU，一种类型用于 dsPIC30F 和 dsPIC33F/PIC24H 器件。

A.7.1 PIC24F 配置设置

使用在器件头文件中提供的宏来设置配置位：

```
_confign(value);
```

其中：

n: 配置寄存器编号。

value: 代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件头文件 (*.h) 中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\Microchip\MPLAB C30\support\PIC24F\h
```

宏的大小写应与相关头文件中匹配。例如，_CONFIG1 是正确的，但 _config1 不正确。

示例——PIC24F MCU

```
#include "p24Fxxxx.h"
```

```
//JTAG off, Code Protect off, Write Protect off, COE mode off, WDT off  
_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDTEN_OFF )
```

```
//Clock switching/monitor off, Oscillator (RC15) on,  
// Oscillator in HS mode, Use primary oscillator (no PLL)  
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_ON & POSCMOD_HS & FNOSC_PRI )
```

A.7.2 dsPIC30F/33F/PIC24H 配置设置

使用在器件头文件中提供的宏来设置配置位：

```
_reg(value);
```

其中：

_reg: 配置寄存器名称宏。

value: 代表指定配置位设置值的表达式。可以为立即数，但通常通过一个宏或多个宏的逻辑与表示。

宏在器件头文件 (*.h) 中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\Microchip\MPLAB C30\support\device\h
```

其中，*device* 是选定 16 位器件的缩写，即 PIC24H、dsPIC30F 或 dsPIC33F。

宏的大小写应与相关头文件中匹配。例如，_FOSC 是正确的，但 _fosc 不正确。

示例——dsPIC30F DSC

```
#include "p30fxxxx.h"

//Clock switching and failsafe clock monitoring off,
// Oscillator in HS mode
_FOSC(CSW_FSCM_OFF & HS);

//Watchdog timer off
_FWDT(WDT_OFF);

//Brown-out off, Master clear on
_FBORPOR(PBOR_OFF & MCLR_EN);
```

示例——dsPIC33F/PIC24H 器件

```
#include "p33fxxxx.h"

// Use primary oscillator (no PLL)
_FOSCSEL(FNOSC_PRI);

//Oscillator in HS mode
_FOSC(POSCMD_HS);

//Watchdog timer off
_FWDT(FWDTEN_OFF);

//JTAG off
_FICD(JTAGEN_OFF);
```

A.8 C32 工具链

`#pragma config` 伪指令指定要由应用程序使用的特定于器件的配置设置（即，配置位）：

```
# pragma config setting-list
```

其中：

setting-list: 一个或多个 *setting-name = value-name* 宏对的列表，使用逗号分隔。

宏在器件头文件（*.h）中指定，器件头文件位于 Windows 操作系统默认目录中：

```
C:\Program Files\Microchip\MPLAB C32\pic32mx\include\proc
```

`Pragma` 伪指令的大小写无关紧要；`#PRAGMA CONFIG` 或 `#pragma config` 均可接受。宏的大小写应与头文件中匹配。

示例

```
#include "p32xxxx.h"

//Enables the Watchdog Timer,
// Sets the Watchdog Postscaler to 1:128
#pragma config FWDTEN = ON, WDTPS = PS128

//Selects the HS Oscillator for the Primary Oscillator
#pragma config POSCMOD = HS
```

A.9 XC 工具链

要创建尽可能便于移植的代码，请参见以下每个文档中的公共 C 接口（CCI）章节的 `config` 宏：

- 《MPLAB[®] XC8 C 编译器用户指南》（DS52053B_CN）或相关帮助文件
- MPLAB[®] XC16 C Compiler User's Guide（DS52071）或相关帮助文件
- 《MPLAB[®] XC32 C 编译器用户指南》（DS51686G_CN）或相关帮助文件

注:

附录 B 在 IDE 之外工作

B.1 简介

MPLAB X IDE 旨在帮助您编写、调试和发布用于嵌入式系统的应用程序。但是，您所在的公司可能存在一些要求，使您需要在 IDE 之外进行代码开发。

您可以按照本章中的说明，在 MPLAB X IDE 之外编译您的代码。如果需要调试该代码，可以将它导入 MPLAB X IDE 项目。更多信息，请参见第 5.3 节“预编译项目”和第 5.4 节“可装入项目和文件”。

或者，您也可以使用调试信息来编译代码，并使用 MPLAB X IDE 的每个版本随附的命令行 Microchip 调试器（MDB）来调试您的代码。关于 MDB 的更多信息，请参见 *Microchip Debugger (MDB) User's Guide* (DS50002102)，它位于 <MPLAB X IDE installation>/docs 目录中。

- 在 MPLAB X IDE 之外编译项目
- 针对在 MPLAB X IDE 之外进行调试而编译

B.2 在 MPLAB X IDE 之外编译项目

MPLAB X IDE 使用 GNU make 作为其编译工具。对于 Linux，将预期您已安装 make。对于 Mac OS X 和 Windows 操作系统，MPLAB X IDE 的安装将提供该程序。其位置为：

- **Windows 32 位操作系统** ——C:\Program Files\Microchip\MPLABX\gnuBins\GnuWin32\bin
- **Windows 64 位操作系统** ——C:\Program Files (x86)\Microchip\MPLABX\gnuBins\GnuWin32\bin
- **Mac OS X** ——/Applications/microchip/mplabx/mplab_ide.app/Contents/Resources/mplab_ide/bin

MPLAB X IDE 自动将包含 make 的目录添加到自己的路径变量中。如果希望在 IDE 之外进行编译，则必须将该目录添加到 PATH 环境变量中。

可以通过直接调用 MPLAB X IDE 项目目录中的 Makefile 来编译默认配置：

命令 —— 在单个行上输入每条命令	说明
\$ make clean	删除所有中间目标和最终映像
\$ make	创建生产映像（Hex 文件）
\$ make TYPE_IMAGE=DEBUG_RUN	创建调试映像（COF/ELF 文件）

如果项目具有多个配置，则：

命令 —— 在单个行上输入每条命令	说明
\$ make -f Makefile CONF=Configuration clean	删除配置 Configuration 的所有中间目标和最终映像
\$ make -f Makefile CONF=Configuration	创建配置 Configuration 的生产映像（Hex 文件）
\$ make -f Makefile CONF=Configuration TYPE_IMAGE=DEBUG_RUN	创建配置 Configuration 的调试映像（COF/ELF 文件）

默认情况下，映像的名称为（相对于 MPLAB X IDE 目录）：

```
dist/$CONF_NAME/production/$PROJ_NAME.production.hex  
dist/$CONF_NAME/debug/$PROJ_NAME.debug.cof（或 elf）
```

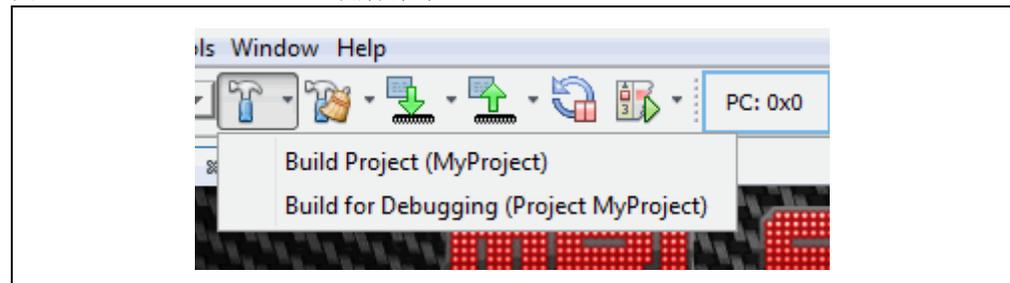
B.3 针对在 MPLAB X IDE 之外进行调试而编译

要针对在 MPLAB X IDE 之外进行调试而编译代码，必须确保向编译器和 / 或链接器传递正确的参数，以保留所使用的特定工具所需的区域。参数值取决于所使用的调试工具和编译器。此外，编译器的版本可能会决定用于保留这些区域的具体机制。

MPLAB X IDE 知道如何确定需要传递给编译器 / 汇编器 / 链接器的参数。所以，确定如何为调试保留区域的最好方法是在 IDE 中使用所需的器件、调试工具和编译器创建一个小项目；请记住，IDE 可以支持同一编译器的多种版本。因此，在 IDE 中创建项目时，请确保选择正确的编译器版本。

在创建小项目之后，针对调试来编译项目。保存输出窗口中说明针对调试的编译步骤的文本。然后，再次编译项目（不用于调试），并保存输出窗口中说明针对生产的编译步骤的文本。比较传递给编译器 / 汇编器 / 链接器的两组指令。

图 B-1: BUILD 图标菜单



输出窗口中的差异的一些示例包括：

- 使用 `__DEBUG` 宏来调用编译器 / 汇编器 / 链接器。
- 使用 `__MPLAB_DEBUG` 宏来调用编译器 / 汇编器 / 链接器。
- 使用 `__ICD2_RAM` 或 `-mreserves` 来调用链接器。这两个参数是互斥的，即使用其中一个调用链接器时，不能使用另一个。
- 使用要使用的调试工具的名称来调用编译器 / 汇编器 / 链接器。

这并不是一个详尽清单。

注:

附录 C 版本历史

版本 A (2011 年 11 月)

- 本文档的初始版本。

版本 B (2012 年 12 月)

- 对于 Windows 和 Linux 操作系统，JRE 现在将自动安装。提供了在 Mac 操作系统中的安装说明，在第 2 章“开始之前”的第 2.2 节“安装 JRE 和 MPLAB X IDE”中进行介绍。
- 更正了第 2 章中关于设备驱动程序名称和路径的信息。
- 在第 3 章“教程”和第 4 章“基本任务”增加了关于调试工具旁的两个指示灯的含义的说明。
- 更新了第 3 章“教程”和第 4 章“基本任务”中的图标定义。
- 在第 4 章“基本任务”，第 4.3 节“创建新项目”的第 4.3.5 节“步骤 5: 选择编译器”中，增加了语言工具链缩写的列表。
- 多个对话框更新为使用新的 MPLAB XC C 编译器显示项目。
- 在第 4 章“基本任务”的第 4.12 节“添加、设置库和目标文件”中增加了关于 Project Properties 对话框中的 Libraries 类别的信息。
- 在第 4 章“基本任务”的第 4.14 节“设置编译属性”中增加了关于标准化十六进制文件的信息。
- 在第 4 章“基本任务”的第 4.20 节“观察符号值变化”中增加了关于如何向 Watches 窗口中添加立即数值的信息。
- 更新了关于 C++ 最新支持的几个章节（当前对于 MPLAB XC32++ 编译器）。
- 在第 5 章“附加任务”的第 5.5 节“可装入项目和文件”中增加了关于可装入项目的章节。
- 在第 5 章“附加任务”的第 5.7 节“其他嵌入式项目”中介绍了如何将嵌入式项目从其他应用程序导入 MPLAB X IDE。
- 在第 5 章“附加任务”的第 5.8 节“示例项目”中增加了关于处理示例项目的信息。
- 在第 5 章“附加任务”的第 5.9 节“处理其他类型的文件”中介绍了如何处理其他文件（例如 XML）。
- 在第 6 章“高级任务”的第 6.4 节“日志数据”中增加了关于如何使用日志文件的信息。
- 在第 6 章“高级任务”的第 6.5 节“定制工具条”中介绍了如何向工具条上添加功能。
- 增加了第 7 章“编辑器”，其中介绍了 MPLAB X IDE 编辑器用法、选项和功能。
- 介绍了在修改或移动默认链接描述文件情况下可能会接收到一条错误消息，以及如何避免这种情况。请参见第 8 章“疑难解答”第 8.4 节“MPLAB X IDE 问题”。
- 将“主要差异”和“功能差异”合并到了第 9 章“MPLAB X IDE 与 MPLAB IDE v8”的第 9.2 节“主要差异”中。说明了 NetBeans 平台是开源的，但 MPLAB X IDE 是专有的。

- 在第 9 章 “MPLAB X IDE 与 MPLAB IDE v8” 中，更新了第 9.3 节 “菜单差异” 以反映 MPLAB X IDE 中的新功能，并更新了第 9.4 节 “工具支持差异” 以反映新的插件支持。
- 在第 10 章 “桌面参考” 中，更新了第 10.2 节 “菜单”，以反映 MPLAB X IDE 中的新功能。
- 将 “窗口和对话框” 拆分到第 11 章 “MPLAB X IDE 窗口和对话框” 和第 12 章 “NetBeans 窗口和对话框” 中，以更好地区分不同的窗口。更新了窗口菜单。
- 更新了第 13 章 “项目文件和文件夹” 的第 13.2 节 “Files 窗口视图” 中的项目文件结构。增加了关于 makefile 的信息。
- 在第 14 章 “配置设置汇总” 的第 14.8 节 “XC 工具链” 中更新了 MPLAB XC C 编译器配置位的介绍，以引用公共 C 接口 (CCI)。

版本 C (2014 年 3 月)

- 总体更改：删除了对旧工具的引用。增加了关于 Windows 8 的信息。更新了屏幕和文本，使之与最新的 GUI 匹配。
- 第 1 章 “什么是 MPLAB X IDE?”：更新了图片引用。以参考资源的形式提及了独立的工具帮助和 Microchip Wiki。
- 第 2 章 “开始之前”：更新了第 2.3.2 节 “Windows® XP/7/8 操作系统的 USB 驱动程序安装”。在第 2.5 节 “安装语言工具” 下增加了编译器许可信息。增加了第 2.7.1 节 “设置硬件工具来处理多个实例”。
- 第 4 章 “基本任务”：删除了第 4.1 节 “简介”。增加了第 4.6.1 节 “添加或更改工具链” 和第 4.6.2 节 “关于工具链路径”。在第 4.12 节 “设置文件和文件夹属性” 下增加了关于如何从编译中排除文件 / 文件夹的信息。在第 4.13 节 “设置编译属性” 下将表 4-3 扩展为一些章节。增加了第 4.15.2 节 “运行注意事项”。增加了第 4.16.2 节 “生成的调试宏” 和第 4.16.3 节 “调试注意事项”。在第 4.19 节 “观察符号值变化” 中增加了符号信息。更新了第 4.21.2 节 “更改器件存储器”。增加了第 4.23.1 节 “设置项目编程属性”。
- 第 5 章 “附加任务”：删除了第 5.1 节 “简介”。更新了第 5.4 节 “可装入项目和文件”，并创建了新章节第 5.5 节 “可装入项目和文件 —— 自举程序”。增加了第 5.12 节 “修改项目文件夹和编码” 和第 5.13 节 “加快编译速度”。更新了文本，使之与第 5.17 节 “查看仪表板显示” 中新的显示选项匹配。在第 5.19 节 “控制源代码” 下，修正了使用版本控制步骤，并更新了需要保存到存储库中的项目文件。更新了第 5.20 节 “在代码开发和错误跟踪方面进行协作” 的选项。重新排列了第 5.21 节 “添加插件工具” 下的章节内容，并增加了第 5.21.4 节 “插件代码位置”。
- 第 6 章 “高级任务”：增加了第 6.1 节 “加快 MPLAB X IDE 速度” 和第 6.5 节 “创建用户 makefile 项目”。在第 6.3 节 “处理多个项目” 下增加了关于在不设置活动项目的情况下工作和对项目进行分组的信息。从第 6.4 节 “处理多个配置” 中删除了调试配置过去需要的语句（现在不需要）。更新了第 6.6 节 “日志文件” 下的日志文件要求。
- 第 7 章 “编辑器”：将 “编辑器功能说明” 移动到第 7.2 节 “编辑器用法” 下的第 7.2.4 节。此外增加了第 7.2.1 节 “桌面控件”、第 7.2.2 节 “C 代码中的超链接” 和第 7.2.3 节 “ASM 代码中的超链接”。扩充了表 7-6 “Macros 选项卡” 中的宏定义。将第 7.4 节 “代码折叠” 重新整理到第 7.4.1 节 “代码折叠用法”（包括 MPLAB C18 和汇编折叠问题）和第 7.4.2 节 “定制代码折叠” 中。

- 第 8 章 “项目文件和文件夹”：这一章移自第 13 章。增加了第 8.4 节 “Favorites 窗口视图”、第 8.5 节 “Classes 窗口视图”、第 8.6 节 “查看用户配置数据” 和第 8.9 节 “删除项目”。将 “移动项目” 更改为第 8.8 节 “移动、复制或重命名项目”。将 “在 MPLAB X IDE 之外编译项目” 移动到附录 B “在 IDE 之外工作”。
- 第 9 章 “疑难解答”：在第 9.6 节 “错误” 中增加了一些信息。
- 第 10 章 “MPLAB X IDE 与 MPLAB IDE v8”：在第 10.2 节 “主要差异” 的第 3 项 “MPLAB X IDE 允许选择多个工具” 和第 9 项 “MPLAB X IDE 使用在代码中设置的配置位” 下进行了阐述。对 MPLAB X IDE 的其他选择进行了其他少量更新。
- 第 11 章 “桌面参考”：第 11.2.10 节 “Tools 菜单” 中更新了关于插件和编译器许可的信息。对 MPLAB X IDE 的其他选择进行了其他少量更新。
- 第 12 章 “MPLAB X IDE 窗口和对话框”：增加了第 12.2 节 “MPLAB X IDE 窗口管理”。在第 12.3 节 “MPLAB X IDE 窗口与相关菜单和对话框” 中，向表中添加了新的窗口，并将窗口小节移动到它们自己的章节中。增加了第 12.4 节 “Breakpoints 窗口” 和第 12.6 节 “Licenses 窗口”。第 12.8 节 “存储器窗口” 下进行了大量更新，增加了对于不同存储器窗口的说明。第 12.11 节 “Projects 窗口” 菜单有更新。第 12.12 节 “Tools Options Embedded 窗口” 选项有更新，增加了第 12.12.5 节 “Diagnostics 选项卡”。第 12.14 节 “Watches 窗口” 内容拆分为三个部分。
- 附录 A: “配置设置”：从第 14 章更改为附录。增加了关于 Configuration Memory 窗口的信息。
- 附录 B: “在 IDE 之外工作”：B.1 节 “在 MPLAB X IDE 之外编译项目” 来自第 8 章 “项目文件和文件夹”。增加了 B.2 节 “针对在 MPLAB X IDE 之外进行调试而编译”。
- 附录 C: “版本历史”：这一章中更新了关于从版本 B 到 C 的主要更改的信息。

注:

支持

简介

关于支持问题，请参考本章提供的内容。

- 保修登记
- myMicrochip 个性化通知客户服务
- Microchip 网站
- Microchip 论坛
- 客户支持
- 联系 Microchip Technology

保修登记

网站: <http://www.microchipdirect.com>

注册您的开发工具，这将使您可以接收到新产品更新。可在 Microchip 网站上获得临时软件版本。

myMICROCHIP 个性化通知客户服务

myMicrochip: <http://www.microchip.com/pcn>

Microchip 的个人通知客户服务有助于客户了解关于所关注 Microchip 产品的最新信息。注册客户可在某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

请访问 myMicrochip 开始注册过程，并选择接收个性化通知的首选项。页面上提供了 FAQ 和注册详细信息，可以通过选择上面的链接打开。

在选择首选项时，选择“Development Systems”（开发系统）将会在列表中填入可用的开发工具。下面列出了主要的工具类别：

- **编译器**——Microchip C 编译器、汇编器、链接器及其他语言工具的最新信息。包括所有 MPLAB C 编译器；所有 MPLAB 汇编器（包括 MPASM™ 汇编器）；所有 MPLAB 链接器（包括 MPLINK™ 目标链接器）；以及所有 MPLAB 库管理器（包括 MPLIB™ 目标库管理器）。
- **仿真器**——Microchip 在线仿真器的最新信息，包括 MPLAB REAL ICE™ 在线仿真器。
- **在线调试器**——Microchip 在线调试器的最新信息，包括 PICKit™ 2、PICKit 3 和 MPLAB ICD 3 在线调试器。
- **MPLAB® IDE**——关于开发系统工具的基于 Windows® 集成开发环境 Microchip MPLAB IDE 的最新信息。主要针对 MPLAB IDE、MPLAB IDE 项目管理器、MPLAB 编辑器、MPLAB SIM 模拟器以及一般编辑和调试功能。
- **编程器**——Microchip 编程器的最新信息。包括器件（生产）编程器 MPLAB REAL ICE 在线仿真器、MPLAB ICD 3 在线调试器和 MPLAB PM3 以及开发（非生产）编程器 PICKit 2 和 3。
- **入门 / 演示板**——它们包括 MPLAB 入门工具包电路板、PICDEM 演示板以及各种其他评估板。

MICROCHIP 网站

网站: <http://www.microchip.com>

Microchip 网站为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动策划表、Microchip 销售办事处、代理商以及工厂代表列表

MICROCHIP 论坛

论坛: <http://www.microchip.com/forums>

Microchip 网上论坛为客户提供更多的在线支持。当前提供的论坛包括:

- 开发工具
- 8 位 PIC MCU
- 16 位 PIC MCU
- 32 位 PIC MCU

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

可通过如下网站获得技术支持:

<http://support.microchip.com>

文档错误或意见可通过电子邮件发送至 CTRC@microchip.com。

联系 MICROCHIP TECHNOLOGY

通过以下号码致电或发送传真到 Microchip 总部:

电话: 1-480-792-7200

传真: 1-480-792-7277

注:

术语表

A

ANSI

美国国家标准学会，是美国负责制订和批准标准的组织。

ASCII

美国信息交换标准码是使用 7 个二进制位来表示每个字符的字符集编码。它包括大写和小写字母、数字、符号以及控制字符。

B

Build

编译并链接一个应用程序的所有源文件。

八进制 (Octal)

使用数字 0-7，以 8 为基数的计数体制。最右边的位表示 1 的倍数，右边第二位表示 8 的倍数，右边第三位表示 $8^2 = 64$ 的倍数，以此类推。

本机数据大小 (Native Data Size)

对于本机跟踪，在 Watches 窗口中使用的变量大小必须等于选定器件的数据存储器的大小：对于 PIC18 器件为字节，对于 16 位器件为字。

编译器 (Compiler)

将用高级语言编写的源文件翻译成机器码的程序。

标识符 (Identifier)

函数名或变量名。

表达式 (Expression)

被算术或逻辑操作符分隔开的常量和 / 或符号的组合。

C

C/C++

C 语言是具有表达式简练、现代控制流程和数据结构，以及操作符丰富等特点的通用编程语言。C++ 是 C 语言的面向对象版本。

COFF

公共目标文件格式。这种格式的目标文件包含机器码、调试及其他信息。

CPU

参见中央处理单元。

操作符 (Operator)

在构成定义明确的表达式时使用的符号，例如加号 “+” 和减号 “-”。每个操作符都具有指定的优先级，用于确定求值的顺序。

操作码 (Opcode)

操作码。参见助记符。

插件 (Plug-in)

MPLAB IDE/MPLAB X IDE 具有内置组件和接插模块，用于针对各种软件和硬件工具来配置系统。可在 Tools (工具) 菜单下找到几个插件工具。

场景 (Scenario)

MPLAB SIM 软件模拟器用于激励控制的一种特定设置。

程序存储器 (Program Memory)

MPLAB IDE/MPLAB X IDE——器件中存储指令的存储区。也是仿真器或软件模拟器中包含已下载目标应用固件的存储器。

16 位汇编器/编译器——器件中存储指令的存储区。

程序计数器 (Program Counter)

包含正在执行的指令的地址的存储单元。

程序计数器单元 (Program Counter Unit)

16 位汇编器——程序存储器布局的概念性表示。对于每个指令字，程序计数器将递增 2。在可执行段中，2 个程序计数器单元相当于 3 个字节。在只读段中，2 个程序计数器单元相当于 2 个字节。

持久性数据 (Persistent Data)

永不清除或初始化的数据。其目的是使应用程序可以在器件复位时保存数据。

重定位 (Relocation)

链接器执行的一个过程，在这个过程中，为可重定位段分配绝对地址，且可重定位段中的所有符号都更新为其新地址。

触发输出 (Trigger Output)

触发输出指可在任意地址或地址范围产生的仿真器输出信号，与跟踪和断点的设置无关。可设置任意个触发输出点。

次数计数器 (Pass Counter)

每次一个事件（如执行特定地址处的一条指令）发生时都会递减 1 的计数器。当次数计数器的值为零时，事件满足。可将次数计数器分配给断点和跟踪逻辑，以及在 complex trigger（复杂触发）对话框中的任何连续事件。

存储类别 (Storage Class)

确定与指定对象相关联存储区的存在时间。

存储器模型 (Memory Model)

对于 C 编译器，指应用程序可使用的存储区的表示。对于 PIC18 C 编译器，指一种描述，它指定指向程序存储器的指针的位数。

存储限定符 (Storage Qualifier)

指明所声明对象的特殊属性（如 const）。

错误 / 错误文件 (Error/Error File)

错误报告使程序不能继续处理的问题。而且，当问题比较明显时，错误还尽可能标识出源文件名和行号。错误文件包含由语言工具生成的错误消息和诊断信息。

D**DWARF**

使用任意记录格式调试。DWARF 是用于 ELF 文件的调试信息格式。

带序列号的快速批量编程 (Serialized Quick Turn Programming)

序列化使您可以将序列号编程到器件编程器进行编程的每个单片机中。该编号可用作记录代码、密码或 ID 号。

单片机 (Microcontroller)

高度集成的芯片，它包括 CPU、RAM、程序存储器、I/O 端口和定时器。

单片机模式 (Microcontroller Mode)

PIC18 单片机的一种程序存储器配置。在单片机模式下，仅允许内部执行。因此，在单片机模式下仅可使用片内程序存储器。

导出 (Export)

以标准的格式将数据发送出 MPLAB IDE/MPLAB X IDE。

导入 (Import)

从外面的源（如 hex 文件）将数据送入 MPLAB IDE/MPLAB X IDE。

地址 (Address)

标识存储器中位置的值。

递归 (Recursion)

已定义的函数或宏可调用自己的概念。当编写递归宏时要特别小心；当递归没有出口时容易陷入无限循环。

递归调用 (Recursive Call)

一个直接或间接调用自身的函数。

段 (Section)

OCG psect 的 GCC 等效形式。被链接器视为整体的代码块或数据块。

段属性 (Section Attribute)

赋予段的 GCC 特性（例如，access 段）。

断点 (Breakpoint)

硬件断点：一个事件，执行这种事件会导致暂停。

软件断点：一个地址，固件会在这个地址处暂停执行。通常由特殊的 break 指令获得。

断点逻辑与 (ANDed Breakpoint)

为中断设置逻辑与条件，即只有断点 1 和断点 2 同时出现时，才会暂停程序。只有数据断点和程序存储器断点同时出现时，才会完成此操作。

堆 (Heap)

用于动态存储器分配的存储区，其中的存储器块按运行时确定的任意顺序进行分配和释放。

堆栈，软件（Stack, Software）

供应用程序用来存储返回地址、函数参数和局部变量的存储器。此存储器由程序中的指令在运行时动态分配。它支持可重入函数调用。

堆栈，已编译（Stack, Compiled）

编译器管理和分配的存储区，在其中为变量静态分配空间。当软件堆栈等机制无法在目标器件上高效实现时，将替换为已编译堆栈。它会阻止重入。

堆栈，硬件（Stack, Hardware）

PIC 单片机中调用函数时存储返回地址的存储区。

E

EEPROM

电可擦除的可编程只读存储器。一种可电擦除的特殊 PROM。一次写或擦除一个字节。EEPROM 即使电源关闭时也能保留内容。

ELF

可执行链接格式。这种格式的目标文件包含机器码。调试和其他信息使用 DWARF 指定。ELF/DWARF 可提供优于 COFF 的优化代码调试。

Epilogue

编译器生成代码的一部分，负责释放堆栈空间、恢复寄存器，以及执行运行时模型中指定的任何其他特定于机器的要求。此代码在给定函数的任何用户代码之后、在函数返回之前执行。

EPROM

可擦除的可编程只读存储器。通常通过紫外线照射来擦除的可编程只读存储器。

二进制（Binary）

使用数字 0 和 1，以 2 为基数的计数体制。最右边的位表示 1 的倍数，右边第二位表示 2 的倍数，右边第三位表示 $2^2 = 4$ 的倍数，以此类推。

F

FNOP

强制空操作。强制 NOP 周期是双周期指令的第二个周期。由于 PIC 单片机的架构是流水线型，在执行当前指令的同时预取物理地址空间中的下一条指令。但是，如果当前指令改变了程序计数器，那么这条预取的指令就被忽略了，导致一个强制 NOP 周期。

Free-Standing

一种实现，它接受任何不使用复杂类型的严格符合程序，而且在这种实现中，对库条款（ANSI '89 标准条款 7）中规定的属性的使用，仅限于标准头文件：<float.h>、<iso646.h>、<limits.h>、<stdarg.h>、<stdbool.h>、<stddef.h> 和 <stdint.h>。

仿真 / 仿真器（Emulation/Emulator）

参见 ICE/ICD。

访问入口点（Access Entry Point）

访问入口点提供了一种方法，可跨段将控制转移到某个可能未在链接时定义的函数。它们支持独立链接引导应用程序段和安全应用程序段。

非扩展模式 (Non-Extended Mode) (PIC18 MCU)

在非扩展模式下，编译器不会使用扩展指令和立即数变址寻址。

非实时 (Non Real Time)

指处理器执行到断点或单步执行指令，或 MPLAB IDE/MPLAB X IDE 运行在软件模拟器模式。

非易失性存储器 (Non-Volatile Storage)

电源关闭时保存其内容的存储器件。

符号 (Symbol)

符号是描述组成程序的各个部分的一种通用机制。这些部分包括函数名、变量名、段名、文件名和结构 / 枚举 / 联合标记名等。MPLAB IDE/MPLAB X IDE 中的符号主要指变量名、函数名和汇编标号。链接后符号的值就是其在存储器中的值。

符号, 绝对 (Symbol, Absolute)

表示一个立即数的值，例如通过汇编 `.equ` 伪指令指定的定义。

G**GPR**

通用寄存器。器件数据存储器 (RAM) 的一部分，作为一般用途。

概要文件 (Profile)

对于 MPLAB SIM 软件模拟器，寄存器已执行激励的汇总列表。

高级语言 (High Level Language)

编写程序的语言，它比汇编语言更不依赖于具体的处理器。

跟踪 (Trace)

记录程序执行的仿真器或软件模拟器功能。仿真器将程序执行记录到其跟踪缓冲区内，并可上载到 MPLAB IDE/MPLAB X IDE 的跟踪窗口中。

跟踪存储区 (Trace Memory)

跟踪存储区包含在仿真器内部。跟踪存储区有时称为跟踪缓冲区。

跟踪宏 (Trace Macro)

一个通过仿真器数据来提供跟踪信息的宏。由于该宏属于软件跟踪，所以必须将它添加到代码中、必须重新编译或重新汇编代码，并且必须使用该代码对目标器件进行编程，之后跟踪才会工作。

工具条 (Tool Bar)

一行或一列图标，单击这些图标时将执行 MPLAB IDE/MPLAB X IDE 功能。

工作簿 (Workbook)

MPLAB SIM 激励器用于生成 SCL 激励的一种设置。

观察变量 (Watch Variable)

调试会话期间可在 Watch 窗口中监控的变量。

归档 / 归档器 (Archive/Archiver)

归档 / 库是可重定位目标模块的集合。由将多个源文件编译 / 汇编为目标文件，然后使用归档器 / 库管理器将目标文件组合为一个归档 / 库文件生成。可将归档 / 库与目标模块和其他归档 / 库链接，生成可执行代码。

国际标准化组织 (International Organization for Standardization)

制订许多行业和技术 (包括计算和通信) 方面的标准的一个组织。也称为 ISO。

过滤器 (Filter)

通过选择确定在跟踪显示或数据文件中包含 / 排除哪些数据。

H

Hex 代码 / Hex 文件 (Hex Code/Hex File)

Hex 代码是以十六进制格式代码存储的可执行指令。Hex 代码包含在 hex 文件中。

宏 (Macro)

宏指令。以缩写形式表示指令序列的指令。

宏伪指令 (Macro Directive)

控制宏定义体中执行和数据分配的伪指令。

环回测试板 (Loop-Back Test Board)

用于测试 MPLAB REAL ICE 在线仿真器的功能。

环境 (Environment)

MPLAB PM3—— 包含关于如何对器件编程的文件的文件夹。该文件夹可以转移到 SD/MMC 卡。

汇编语言 / 汇编器 (Assembly/Assembler)

汇编语言是以符号形式描述二进制机器码的编程语言。汇编器是将汇编源代码翻译成机器码的语言工具。

I

ICE/ICD

在线仿真器 / 在线调试器：用于对目标器件进行调试和编程的硬件工具。仿真器具有比调试器更多的功能，例如跟踪。

在线仿真 / 在线调试：使用在线仿真器或调试器进行仿真或调试的行为。

-ICE/-ICD：带有在线仿真或调试电路的器件 (MCU 或 DSC)。该器件总是安装在仿真头板上，并用于通过在线仿真器或调试器进行调试。

ICSP

在线串行编程。使用串行通信和最低数量的器件引脚对 Microchip 嵌入式器件进行编程的方法。

IDE

集成开发环境，如 MPLAB IDE/MPLAB X IDE。

IEEE

电气电子工程师协会。

J

机盒 (Pod)

在线仿真器或调试器的外壳。其他名称还有“Puck”（如果外壳为圆形）和“Probe”（不要与逻辑探头相混淆）。

机器码 (Machine Code)

处理器实际读和解释的计算机程序的表示。二进制机器码程序由一系列机器指令（可能还包含数据）组成。特定处理器的所有可能指令的集合称为“指令集”。

机器语言 (Machine Language)

特定中央处理单元的指令集，不需翻译即可供处理器使用。

基数 (Radix)

数字基，十六进制或十进制，用于指定一个地址。

激励 (Stimulus)

软件模拟器的输入（即为模拟对外部信号的响应而生成的数据）。通常数据采用文本文件中一系列动作的形式。激励可以是异步的，同步的（引脚），时钟激励和寄存器激励。

交叉引用文件 (Cross Reference File)

引用符号表的一个文件及引用符号的文件的列表。如果定义了符号，则在列出的第一个文件中包含符号的定义。其他文件包含对符号的引用。

校准存储区 (Calibration Memory)

用于保存 PIC 单片机内 RC 振荡器或其他器件外设校准值的特殊功能寄存器或通用寄存器。

节点 (Node)

MPLAB IDE/MPLAB X IDE 项目的组件。

警告 (Warning)

MPLAB IDE/MPLAB X IDE——提醒出现了可能导致器件、软件文件或设备物理损坏的通知。

16 位汇编器 / 编译器 —— 警告报告可能存在问题的条件，但并不暂停处理。在 MPLAB C30 中，警告消息报告源文件名和行号，但包含文本“warning:”来将警告与错误消息区分开。

静态 RAM 或 SRAM (Static RAM or SRAM)

静态随机访问存储器。目标板上可读 / 写且不需要经常刷新的程序存储器。

局部标号 (Local Label)

局部标号是用 LOCAL 伪指令在一个宏内部定义的标号。这些标号特定于宏实例化的一个给定示例。也就是说，声明为 local 的符号和标号在遇到 ENDM 宏后不再可访问。

绝对变量 / 函数 (Absolute Variable/Function)

使用 OCG 编译器的 @ *address* 语法放置在绝对地址的变量或函数。

绝对段 (Absolute Section)

具有链接器不能更改的固定（绝对）地址的 GCC 编译器段。

K

看门狗定时器 (Watchdog Timer, WDT)

PIC 单片机中在一段可选择的时间长度后复位处理器的定时器。使用配置位来使能、禁止和设置 WDT。

可重定位 (Relocatable)

一个目标文件，它的地址没有分配到存储器中的固定地址。

可重定位段 (Relocatable Section)

16 位汇编器——地址非固定（绝对）的段。链接器通过一个称为重定位的过程来为可重定位段分配地址。

可重入函数 (Reentrant)

可以有多个同时运行的实例的函数。在下面两种情况下可能发生函数重入：直接或间接递归调用函数；或者在由函数转入的中断处理过程中又执行此函数。

可执行代码 (Executable Code)

可装入来执行的软件。

控制伪指令 (Control Directive)

汇编语言代码中根据汇编时指定表达式的值包含或忽略代码的伪指令。

库 / 库管理器 (Library/Librarian)

参见归档 / 归档器。

快速存取存储区 (Access Memory)

仅限 PIC18——PIC18 器件中的一些特殊寄存器，对这些寄存器的访问与存储区选择寄存器 (BSR) 的设置无关。

扩展单片机模式 (Extended Microcontroller Mode)

在扩展单片机模式下，既可使用片内程序存储器，也可使用外部存储器。如果程序存储器地址大于 PIC18 器件的内部存储空间，执行自动切换到外部存储器。

扩展模式 (Extended Mode) (PIC18 MCU)

在扩展模式下，编译器将使用扩展指令（即 ADDFSR、ADDULNK、CALLW、MOVSF、MOVSS、PUSHL、SUBFSR 和 SUBULNK）以及立即数变址寻址。

L

LVDS

低电压差分信号。一种通过铜线进行高速（每秒千兆位）数据传输的低噪声、低功耗、低幅值方法。

对于标准 I/O 信号，数据存储依赖于实际电压大小。电压大小会受线路长度影响（线路越长，电阻就越高，这会使电压下降）。但对于 LVDS，数据存储仅通过正负电压值区分，而不是实际电压大小。因此，数据可以传输更长的线路距离，同时保持干净、一致的数据流。

来源：<http://www.webopedia.com/TERM/L/LVDS.html>

链接描述文件 (Linker Script File)

链接描述文件是链接器的命令文件。它们定义链接选项并描述目标平台上的可用存储器。

链接器 (Linker)

将目标文件和库文件组合起来生成可执行代码并解析一个模块对另外一个模块引用的语言工具。

列表伪指令 (Listing Directive)

列表伪指令是控制汇编列表文件格式的伪指令。它们允许指定标题、分页及其他列表控制。

列表文件 (Listing File)

列表文件是列出为每条 C 源语句生成的机器码、源文件中遇到的汇编指令、汇编器伪指令或宏的 ASCII 文本文件。

逻辑探头 (Logic Probe)

Microchip 的某些仿真器最多可连接 14 个逻辑探头。逻辑探头提供外部跟踪输入、触发输出信号、+5V 和公共接地端。

M

Makefile

包含用于 Make 项目的指令的文件。使用该文件可以在 MPLAB IDE/MPLAB X IDE 之外使用 make 编译项目。

Make 项目 (Make Project)

重新编译应用程序的命令，仅编译自上次编译完成后更改了的源文件。

MCU

单片机。microcontroller 的缩写形式。也写作 uC。

MPASM™ 汇编器 (MPASM™ Assembler)

Microchip Technology 用于 PIC 单片机、KeeLoq® 器件及 Microchip 存储器件的可重定位宏汇编器。

MPLAB ICD

Microchip 的在线调试器，与 MPLAB IDE/MPLAB X IDE 配合使用。参见 ICE/ICD。

MPLAB IDE/MPLAB X IDE

Microchip 的集成开发环境。MPLAB IDE/MPLAB X IDE 附带了编辑器、项目管理器和软件模拟器。

MPLAB PM3

Microchip 的器件编程器。用于对 PIC18 单片机和 dsPIC 数字信号控制器进行编程。可与 MPLAB IDE/MPLAB X IDE 配合使用或独立使用。代替 PRO MATE II。

MPLAB REAL ICE™ 在线仿真器 (MPLAB REAL ICE™ In-Circuit Emulator)

Microchip 的新一代在线仿真器，与 MPLAB IDE/MPLAB X IDE 配合使用。参见 ICE/ICD。

MPLAB SIM

Microchip 的软件模拟器，与 MPLAB IDE/MPLAB X IDE 配合使用，支持 PIC MCU 和 dsPIC DSC 器件。

MPLIB™ 目标库管理器 (MPLIB™ Object Librarian)

Microchip 的库管理器，与 MPLAB IDE/MPLAB X IDE 配合使用。MPLIB 库管理器是用于将由 MPASM 汇编器 (mpasm 或 mpasmwin v2.0) 或 MPLAB C18 C C 编译器生成的 COFF 目标模块组合成库文件的目标库管理器。

MPLINK™ 目标链接器 (MPLINK™ Object Linker)

MPLINK 链接器是 Microchip MPASM 汇编器和 Microchip C18 C 编译器的目标链接器。也可将 MPLINK 链接器与 Microchip MPLIB 库管理器配合使用。MPLINK 链接器设计为在 MPLAB IDE/MPLAB X IDE 中使用，尽管它也可独立于 MPLAB IDE/MPLAB X IDE 使用。

MRU

最近使用的。指可从 MPLAB IDE/MPLAB X IDE 主下拉菜单选择的文件和窗口。

命令行接口 (Command Line Interface)

仅基于文本输入和输出，在程序和其用户之间进行通信的一种方式。

模板 (Template)

为以后插入自己的文件中使用而创建的文本行。MPLAB 编辑器将模板存储到模板文件中。

模块 (Module)

执行预处理器伪指令后，源文件的预处理输出。也称为转换单元。

目标 (Target)

指用户硬件。

目标板 (Target Board)

构成目标应用的电路和可编程器件。

目标处理器 (Target Processor)

目标应用板上的单片机。

目标代码 / 目标文件 (Object Code/Object File)

目标代码是由汇编器或编译器生成的机器码。目标文件是包含机器码，也可能包含调试信息的文件。它可以直接执行；或为可重定位的，需要与其他目标文件（如库文件）链接来生成完全可执行的程序。

目标文件伪指令 (Object File Directive)

仅当生成目标文件时使用的伪指令。

目标应用程序 (Target Application)

目标板上的软件。

N**NOP**

空操作。执行该指令时，除了程序计数器加 1 外没有任何其他影响。

内部链接 (Internal Linkage)

如果不能从定义函数或变量的模块外部访问它们，则这样的函数或变量具有内部链接。

匿名结构 (Anonymous Structure)

16 位 C 编译器 —— 未命名的结构。

PIC18 C 编译器 —— 属于 C 联合成员的未命名结构。匿名结构的成员可以像外围联合的成员一样进行访问。例如，在以下代码中，hi 和 lo 是联合 `caster` 中的匿名结构的成员。

```
union castaway
{
    int intval;
    struct {
        char lo; //accessible as caster.lo
        char hi; //accessible as caster.hi
    };
} caster;
```

O**OTP**

一次性编程。非窗口封装的 EPROM 器件。由于 EPROM 需要紫外线照射来擦除其存储内容，因此只有窗口片是可擦除的。

P**PC**

个人计算机或程序计数器。

PC 主机 (PC Host)

运行支持的 Windows 操作系统的任何 PC。

PIC MCU

PIC 单片机 (MCU) 指 Microchip 的所有单片机系列。

PICKit 2 和 3 (PICKit 2 and 3)

Microchip 适用于开发阶段的器件编程器，具有通过 Debug Express 进行调试的功能。要了解每个工具支持哪些器件，请参见每个工具的自述文件。

Pragma 伪指令 (Pragma)

对特定编译器有意义的伪指令。通常一条 pragma 伪指令用于向编译器传达实现定义的信息。MPLAB C30 使用属性来传达这种信息。

Prologue

编译器生成代码的一部分，负责分配堆栈空间、保存寄存器，以及执行运行时模型中指定的任何其他特定于机器的要求。此代码在给定函数的任何用户代码之前执行。

Psect

GCC 段的 OCG 等效形式，程序段的简称。被链接器视为整体的代码块或数据块。

PWM 信号 (PWM Signal)

脉宽调制信号。某些 PIC MCU 包含 PWM 外设。

跑表 (Stopwatch)

测量执行周期的计数器。

配置位 (Configuration Bit)

可对其编程来设置 PIC 单片机工作模式的专用位。配置位可或不可再编程。

片外存储器 (Off-Chip Memory)

片外存储器指 PIC18 器件的一种存储器选择，这种情况下存储器可位于目标板上，或所有程序存储器都由仿真器提供。从 [Options>Development Mode](#) 访问 **Memory** 选项卡可打开 Off-Chip Memory selection（片外存储器选择）对话框。

Q

弃用功能 (Deprecated Feature)

由于历史原因仍然支持但最终将逐步淘汰且不再使用的功能。

器件编程器 (Device Programmer)

用于对电可编程半导体器件（如单片机）进行编程的工具。

器件的 MPLAB 入门工具包 (MPLAB Starter Kit for Device)

Microchip 的入门工具包包含了着手研究指定器件所需的一切。查看工作中的应用程序，然后调试和编程您自己的更改。

器件的 MPLAB 语言工具 (MPLAB Language Tool for Device)

Microchip 用于指定器件的 C 编译器、汇编器和链接器。根据应用中所使用的器件来选择语言工具类型。例如，如果为 PIC18 MCU 编写 C 代码，就选择用于 PIC18 MCU 的 MPLAB C 编译器。

嵌套深度 (Nesting Depth)

宏可包含其他宏的最大深度。

清除 (Clean)

清除会删除活动项目的所有中间项目文件，例如目标文件、hex 文件和调试文件。编译项目时，将基于其他文件重新创建这些文件。

R

RAM

随机访问存储器（数据存储器）。可以以任意顺序访问这种存储器中的信息。

ROM

只读存储器（程序存储器）。不能修改的存储器。

Run

将仿真器从暂停状态释放，允许仿真器实时运行应用代码、实时改变 I/O 状态或实时响应 I/O 的命令。

软件模拟器 (Simulator)

模仿器件操作的软件程序。

S

Shell

MPASM 汇编器 shell 是宏汇编器的提示输入接口。有两个 MPASM 汇编器 shell：一个针对 DOS 版本，一个针对 Windows 操作系统版本。

Single Step

这一命令单步执行代码，一次执行一条指令。执行每条指令后，MPLAB IDE/MPLAB X IDE 更新寄存器窗口、观察变量及状态显示，使您可分析和调试指令。也可单步执行 C 编译器源代码，但不是每次执行一条指令，MPLAB IDE/MPLAB X IDE 将执行一行高级 C 语句生成的所有汇编指令。

Skew

不同时间出现在处理器总线上与指令执行有关的信息。例如，执行前一条指令的过程中取指时，被执行的代码出现在总线上；当实际执行该操作码时，源数据地址及其值以及目标数据地址出现在总线上。当执行下一条指令时，目标数据值出现在总线上。跟踪缓冲区一次捕捉总线上的这些信息。因此，跟踪缓冲区的一条记录将包含三条指令的执行信息。执行一条指令时，从一条信息到另一条信息的捕捉周期数称为 skew。

Skid

当使用硬件断点来暂停处理器时，在处理器暂停前可能再执行一条或多条指令。断点后执行的指令条数称为 skid。

SQTP

参见带序列号的快速批量编程。

Step Into

这一命令与 Single Step 相同。Step Into（与 Step Over 相对）在 CALL 指令后，单步执行子程序。

Step Out

Step Out 使您可以单步跳出当前正在单步执行的子程序。该命令会执行该子程序中的剩余代码，然后在该子程序的返回地址处停止执行。

Step Over

Step Over 允许调试代码时不单步执行子程序。当 step over 一条 CALL 指令时，下一个断点将设置在 CALL 指令后的下一条指令处。如果由于某种原因，子程序陷入无限循环或不正确返回，下一个断点将永远执行不到。除了对 CALL 指令的处理外，Step Over 命令与 Single Step 相同。

三字符组合（Trigraph）

三字符序列，都以 ?? 开头，由 ISO C 定义为单个字符的替代。

闪存存储器（Flash）

按块（而不是按字节）写或擦除数据的一种 EEPROM。

上电复位仿真（Power-on-Reset Emulation）

在应用开始上电时，将随机值写到数据 RAM 区中来模拟 RAM 中的未初始化值的软件随机过程。

上载 (Upload)

上载功能将数据从一个工具（如仿真器或编程器）传送到主机 PC，或将数据从目标板传送到仿真器。

生产编程器 (Production Programmer)

生产编程器是一种编程工具，其中设计了可对器件进行快速编程的资源。它具有在各种电压下进行编程的能力并完全符合编程规范。在生产环境中，应用电路需要在组装线上传送，时间是极其重要的，所以尽可能快地对器件编程至关重要。

十六进制 (Hexadecimal)

使用数字 0-9 以及字母 A-F（或 a-f），以 16 为基数的计数体制。字母 A-F 表示值为 10 至 15（十进制）的十六进制数字。最右边的位表示 1 的倍数，右边第二位表示 16 的倍数，右边第三位表示 $16^2 = 256$ 的倍数，以此类推。

实时 (Real Time)

当在线仿真器或调试器从暂停状态释放时，处理器以实时模式运行且与芯片的正常操作相同。在实时模式下，使能仿真器的实时跟踪缓冲区，并持续捕捉所有选择的周期，使能所有断点逻辑。在在线仿真器或调试器模式下，处理器实时运行，直到有效断点导致暂停，或者直到用户暂停执行。

在软件模拟器模式下，实时仅意味着单片机指令的执行速度与主机 CPU 可模拟的指令速度一样快。

事件 (Event)

对可能包含地址、数据、次数计数、外部输入、周期类型（取指和读/写）及时间标记的总线周期的描述。事件用于描述触发、断点和中断。

书签 (Bookmarks)

使用书签可轻松地查找文件中的指定行。

在 Editor（编辑器）工具栏中选择 Toggle Bookmarks（开关书签）可添加/删除书签。单击该工具栏上的其他图标可移动到下一个或上一个书签。

舒缓 (Relaxation)

将某一指令转换为功能相同但大小较小的指令的过程。这对于节省代码长度非常有用。MPLAB XC16 目前知道如何将 CALL 指令 relax 为 RCALL 指令。当被调用的符号处于当前指令的 +/- 32k 指令字范围内时，将会执行该操作。

属性 (Attribute)

C 程序中变量或函数的 GCC 特性，用于描述特定于机器的性质。

属性, 段 (Attribute, Section)

段的 GCC 特性，可以在汇编器 .section 伪指令中指定为标志，例如“executable”、“read-only”或“data”。

数据存储 (Data Memory)

在 Microchip MCU 和 DSC 器件中，数据存储 (RAM) 由通用寄存器 (GPR) 和特殊功能寄存器 (SFR) 组成。某些器件还有 EEPROM 数据存储。

数据监视与控制界面 (Data Monitor and Control Interface, DMCI)

数据监视与控制界面 (DMCI) 是 MPLAB X IDE 中的一个工具。该界面可以对项目中的应用程序变量提供动态输入控制。

应用程序生成的数据可以使用 4 个可动态分配图形窗口中的任意一个以图形方式进行查看。

数据伪指令 (Data Directive)

数据伪指令是指控制汇编器的程序或数据存储空间分配, 并提供通过符号 (即有意义的名称) 引用数据项的方法的伪指令。

数字信号处理 / 数字信号处理器 (Digital Signal Processing/Digital Signal Processor)

数字信号处理 (DSP) 指数字信号以及已转换为数字形式 (经过采样的) 的一般模拟信号 (声音或图像) 的计算机处理。数字信号处理器是设计为用于数字信号处理的微处理器。

数字信号控制器 (Digital Signal Controller)

数字信号控制器 (DSC) 是具有数字信号处理能力的单片机 (如 Microchip 的 dsPIC DSC 器件)。

顺序断点 (Sequenced Breakpoints)

按顺序发生的断点。断点的执行顺序为从下到上; 序列中的最后一个断点最先发生。

T**特殊功能寄存器 (Special Function Register, SFR)**

数据存储器 (RAM) 的一部分, 专用于控制 I/O 处理器函数、I/O 状态、定时器或其他模式及外设的寄存器。

调试 / 调试器 (Debug/Debugger)

参见 ICE/ICD。

调试信息 (Debugging Information)

编译器和汇编器选项, 在选中时, 它们将提供不同程度的信息来用于调试应用程序代码。关于选择调试选项的详细信息, 请参见编译器或汇编器文档。

条件编译 (Conditional Compilation)

只有当预处理器伪指令指定的某个常量表达式为真时才编译程序段的操作。

条件汇编 (Conditional Assembly)

基于指定表达式在汇编时的值包含或忽略的汇编语言代码。

U**USB**

通用串行总线。一种外部外设接口标准, 用于通过电缆使用双向串行传输在计算机和外部外设之间进行通信。USB 1.0/1.1 支持 12 Mbps 的数据传输速率。USB 2.0 (也称为高速 USB) 支持最高 480 Mbps 的数据传输速率。

V**Volatile**

一种变量限定符, 用于阻止编译器应用会影响存储器中变量访问方式的优化。

W

Watch 窗口 (Watch Window)

Watch 窗口包含一系列观察变量，这些变量在每次执行到断点时更新。

外部 RAM (External RAM)

芯片外部的读 / 写存储器。

外部标号 (External Label)

具有外部链接的标号。

外部符号 (External Symbol)

具有外部链接的标识符符号。这可能是一个引用或一个定义。

外部符号解析 (External Symbol Resolution)

链接器收集所有输入模块的外部符号定义来解析所有外部符号引用的过程。没有相应定义的任何外部符号引用都会导致报告链接器错误。

外部链接 (External Linkage)

如果可以在定义函数或变量的模块外部对函数或变量进行引用，则函数或变量具有外部链接。

外部输入线 (External Input Line)

用于根据外部信号设置事件的外部输入信号逻辑探针线 (TRIGIN)。

微处理器模式 (Microprocessor Mode)

PIC18 单片机的一种程序存储器配置。在微处理器模式下，不使用片内程序存储器。整个程序存储器映射到外部。

伪指令 (Directive)

源代码中控制语言工具操作的语句。

尾数法 (Endianness)

多字节对象中的字节存储顺序。

未初始化数据 (Uninitialized Data)

定义时未指定初始值的数据。在 C 中，

```
int myVar;
```

定义了将存放到未初始化数据段的一个变量。

未分配段 (Unassigned Section)

在链接器命令文件中未分配到特定目标存储区的段。链接器必须找到用于分配未分配段的目标存储区。

文件寄存器 (File Register)

片内数据存储器，包括通用寄存器 (GPR) 和特殊功能寄存器 (SFR)。

X**系统窗口控制 (System Window Control)**

系统窗口控制位于窗口和某些对话框的左上角。单击这一控制通常会弹出包含 Minimize (最小化)、Maximize (最大化) 和 Close (关闭) 等项的菜单。

下载 (Download)

下载是将数据从主机发送到其他设备 (如仿真器、编程器或目标板) 的过程。

限定符 (Qualifier)

次数计数器使用的地址或地址范围, 或用作复杂触发中另一个操作之前的事件。

响应延时 (Latency)

事件与其得到响应之间的延迟时间。

向量 (Vector)

复位或中断发生时应用程序跳转到的存储地址。

项目 (Project)

项目包含编译应用程序所需的文件 (源代码和链接描述文件等), 以及它们与各种编译工具和编译选项的关联。

消息 (Message)

显示出来的文本, 警告在语言工具的操作中可能存在的问题。消息不会停止操作。

小尾数法 (Little Endian)

多字节数据的数据存储顺序机制, 在这种机制中, 低字节存储在较低的地址中。

修正 (Fixup)

链接器重定位后将目标文件符号引用替换为绝对地址的过程。

虚拟字节 (Phantom Byte)

dsPIC 架构中的未实现字节, 在将 24 位指令字视为 32 位指令字时使用。虚拟字节出现在 dsPIC hex 文件中。

Y**样机系统 (Prototype System)**

指用户目标应用或目标板的一个术语。

已编译堆栈 (Compiled Stack)

编译器管理的存储区, 在其中为变量静态分配空间。当软件堆栈或硬件堆栈机制无法在目标器件上高效实现时, 将替换为已编译堆栈。

已初始化数据 (Initialized Data)

用初始值定义的数据。在 C 中,

```
int myVar=5;
```

定义了将存放到已初始化数据段中的一个变量。

已分配段 (Assigned Section)

在链接器命令文件中已分配到目标存储区的 GCC 编译器段。

异步 (Asynchronously)

不同时发生的多个事件。通常用来指可能在处理器执行过程中的任意时刻发生的中断。

异步激励 (Asynchronous Stimulus)

为模拟到所模拟器件的外部输入而生成的数据。

应用 (Application)

可由 PIC® 单片机控制的一组软硬件。

优先顺序 (Precedence)

定义表达式中求值顺序的规则。

原始数据 (Raw Data)

与一个段有关的代码或数据的二进制表示。

源代码 (Source Code)

编程人员编写计算机程序的形式。采用某种正式的编程语言编写的源代码可翻译成机器码或被解释程序执行。

源文件 (Source File)

包含源代码的 ASCII 文本文件。

运行时观察 (Runtime Watch)

Watch (观察) 窗口, 其中的变量会在应用程序运行时更改。要确定如何设置运行时观察, 请参见相应的工具文档。并不是所有工具都支持运行时观察。

运行时模型 (Run-time Model)

描述目标架构资源的使用。

Z

暂停 (Halt)

停止程序执行。执行 Halt 与在断点处停止相同。

帧指针 (Frame Pointer)

引用堆栈中地址, 并将基于堆栈的参数和基于堆栈的局部变量分隔开的指针。为访问当前函数的局部变量和其他值提供了方便。

只读存储器 (Read Only Memory)

存储器硬件, 它允许快速访问其中永久存储的数据, 但不允许添加或修改数据。

指令 (Instruction)

告知中央处理单元执行特定操作, 并可能包含操作中要使用的数据的位序列。

指令集 (Instruction Set)

特定处理器理解的机器语言指令的集合。

致命错误 (Fatal Error)

引起编译立即停止的错误。不产生其他消息。

中断 (Interrupt)

传递到 CPU 的信号, 它使 CPU 暂停执行正在运行的应用程序, 将控制权转交给中断服务程序 (ISR), 以处理事件。在完成 ISR 时, 将恢复应用程序的正常执行。

中断处理程序 (Interrupt Handler)

发生中断时处理特殊代码的子程序。

中断服务程序 (Interrupt Service Routine, ISR)

语言工具 —— 处理中断的函数。

MPLAB IDE/MPLAB X IDE —— 当产生中断时进入的用户生成代码。代码在程序存储器中的位置通常取决于所产生中断的类型。

中断服务请求 (Interrupt Service Request, IRQ)

使处理器暂停正常的指令执行并开始执行中断处理程序的事件。某些处理器有几种中断请求事件，允许具有不同优先级的中断。

中断向量 (Interrupt Vector)

中断服务程序或中断处理程序的地址。

中央处理单元 (Central Processing Unit)

器件的一部分，负责取出要执行的正确指令，对指令进行译码，然后执行指令。如果有必要，它和算术逻辑单元 (Arithmetic Logic Unit, ALU) 一起工作来完成指令的执行。它控制程序存储器的地址总线、数据存储器的地址总线和对堆栈的访问。

助记符 (Mnemonic)

可直接翻译成机器码的文本指令。也称为操作码。

状态条 (Status Bar)

状态条位于 MPLAB IDE/MPLAB X IDE 窗口的底部，指示光标位置、开发模式和器件，以及有效工具条等当前信息。

字母数字字符 (Alphanumeric)

字母数字字符由字母字符和十进制数字 (0, 1, ..., 9) 组成。

字母字符 (Alphabetic Character)

字母字符指属于阿拉伯字母表 (a, b, ..., z, A, B, ..., Z) 中字母的字符。

左值 (L-value)

引用可被检查和 / 或修改的对象的表达式。左值表达式用在赋值的左侧。

注:

索引

符号

__DEBUG 141, 174

A

About 164
 Add to Favorites 162
 Apply Code Changes (应用代码更改) 126
 Apply Diff Patch 162
 ASM30 工具链 69
 Attach Debugger (连接调试器) 126
 安装 MPLAB X IDE 27
 安装语言工具 31

B

Back 158
 Batch Build Project 160
 Build for Debugging Main Project
 (针对调试主项目而进行编译) 128
 Build Main Project (编译主项目) 128
 Build Project 128, 160
 版本控制 113
 保持硬件工具处于活动连接状态 86, 87
 备用项目十六进制文件 103
 本地历史记录 113
 崩溃日志记录 124
 编辑器工具条 54, 80
 编辑器用法 54, 80
 编译后选项 84
 编译配置 146, 151
 编译器支持指示灯 69
 编译前选项 84
 编译属性
 文件 175
 项目 175
 编译项目 55, 85
 编译状态 172
 标准化十六进制文件 84

C

C++ 文件类型 107
 C18 工具链 69
 C24 工具链 69
 C30 工具链 69
 C32 工具链 69
 C 文件类型 107
 C 扩展名 77
 Call Graph 130
 Call Stack 130
 Classes 130
 Clean and Build Main Project
 (清除并编译主项目) 128
 Clean and Build Project 128, 160

Clean Main Project (清除主项目) 128
 Clean Only 图标 125
 Clean Project (清除项目) 128
 Clear Document Bookmarks (清除文档书签) 127
 Clear output window before build 177
 Clear tool output window on new session 177
 Close All Projects 156
 Close open source file on project close 177
 Close Project 156
 Code Folding 158
 Compile File (编译文件) 128
 Complete Code 159
 Configure 菜单变化 153
 Continue 126, 161
 Copy 157
 Copy (复制) 127
 CPU 存储器 95
 CVS 162
 Check CVS (检查 CVS) 129
 Commit 129
 Diff 129
 Revert Modifications (还原修改) 129
 Show Annotations (显示标注) 129
 Update (更新) 129
 Customize Zoom (定制缩放显示) 130
 Cut 157
 Cut (剪切) 127
 菜单 156
 插件 116
 程序存储器 95
 程序和数据存储器 13
 程序计数器 13
 重构 139
 创建新的项目文件 77
 创建新项目 39, 64
 存储器
 程序和数据 13
 存储器, 类型和数量 111
 存储器类型 111
 存储器数量 111
 错误 143

D

__DEBUG 122, 146
 Dashboard 窗口 111, 130
 Debug File 161
 Debug File (调试文件) 126
 Debug Main Project (调试主项目) 126
 Debug Project 161
 Debug Test File 161
 Debug Test File (调试测试文件) 126
 Debug 菜单 161

Delete	127, 157	F	
Diff	162	Favorites 窗口	130
Disassembly Listing File	130	File 菜单	156
Disassembly 窗口	110, 125	File 菜单变化	148
Disconnect from Debug Tool	161	Files	128
Disconnect from Debug Tool (与调试工具断开)	125	Files 窗口视图	184
Discrete Debugger Operation	161	Fill Memory	171
dsPIC 工具链	69	Find	157
DTDs and XML Schemas	162	Find in Projects	157
Duplicate Down	159	Find in Projects (在项目中查找)	127
Duplicate Up (向上复制)	159	Find Next	157
代码重构	112	Find Previous	157
单步执行代码	58, 91	Find Selection	157
导入 MPLAB IDE v8 项目 —— 相对路径	186	Find Usages	157
导入 MPLAB 旧项目	100	Find Usages 窗口	131
电压, 硬件工具	111	Find (查找)	127
调试编译配置	146, 151	Finish Debugger Session	126, 161
调试读取	97	Fix Code	159
调试配置	122	Focus Cursor at PC	161
调试器 / 编程器选项	47, 74	Focus Cursor at PC (将光标焦点设置在 PC 处)	125
调试运行代码	56, 87	Format	159
调用堆栈	98	Forward	159
调用图	110	Full Screen	158
定制工具条	125	发布编译配置	146, 151
读物, 推荐	9	反汇编列表文件	110
断点	57, 88, 130	非活动连接	111
窗口	89	复位时在主程序处停止	177
对话框	89	复位时在复位向量处停止	177
行	88	复位向量	
可用数量	111	复位	177
逻辑与	90	停止	177
时序	90	复杂断点	89
序列	89	G	
元组	90	Go to Declaration	158
支持的软件断点	111	Go to File	158
资源	90	Go to Line	159
断点资源	111	Go to Previous Document	158
堆栈		Go to Source	158
HW	13	Go to Super Implementation	158
Make Callee Current	161	Go to Symbol	158
Make Caller Current	161	Go to Type	158
Pop Topmost Call	161	GPR	13
对器件编程	61, 98	工具条	165
多个配置	121	工具支持指示灯	67
多个项目	119	工作区 (MPLAB v8)	148
E		固件版本	172
Edit 菜单	157	固件版本, 硬件工具	111
Edit 菜单变化	149	H	
EEPROM	13	Halt build on first failure	177
EE 数据存储	95	Help	127, 164
Enable alternate watch list views during debug session	177	Help 菜单	164
Erase Device Memory Main Project (擦除器件存储器主项目)	128	Help 菜单变化	153
Exception Reporter 窗口	130	Hierarchy (层次结构) 窗口	130
Exit	157	HI-TECH DSPIC 工具链	69
Experimental Terminal 窗口	130	HI-TECH PICC18 工具链	69
Export All Watches to a File	179	HI-TECH PICC32 工具链	69
Export Hex	174	HI-TECH PICC 工具链	69
		Hold in Reset (保持复位)	128

-
- 宏扩展 130
宏录制开始 / 停止 157
互联网地址, Microchip 200
汇编器文件类型 107
- I**
- ICSP 23
IDE Log 130, 158
Import 156
Insert Code 159
Insert Next Matching Word 159
Insert Previous Matching Word 159
- J**
- JRE, 安装 27
键盘快捷方式, MPLAB X IDE 164
校验和 111
校验和作为用户 ID 84
- K**
- 看门狗定时器 209
可禁止的消息 77
可装入文件 103
可装入项目 103
客户支持 200
库文件 81
库项目 106
跨平台问题 141
- L**
- Last Edit Location 158
Launch Debugger Main Project
(启动调试器主项目) 126
Licenses 162
Local History 162
Locate Headers 174
Log File 177
Logging Levels 177
立即开始执行 177
连接目标 31
联系 Microchip Technology 201
路径, 相对或绝对 77
- M**
- Maintain active connection to hardware tool 177
main 函数
 复位 177
 停止 177
Make 125
Make and Program Device 175
Make and Program Device Main Project
(对器件主项目执行 make 并编程) 128
Make Callee Current
(将被调用方设为当前调用) 126
Make Caller Current
(将调用方设为当前调用) 126
Make Clean (Make 清除) 125
Make Options 176
Makefile 107
Make 选项 84
make 选项 77
Memory Gauge (存储器计量) 111
- Memory (存储器) 窗口 60, 95
Mercurial 162
Move Down 159
Move Up 159
MPASM 工具链 69
MPLAB IDE v8 项目, 导入 186
MPLAB X IDE 安装和设置 27
MPLAB X IDE 与 MPLAB IDE v8 145
myMICROCHIP 个性化通知客户服务 199
- N**
- Navigate 菜单 158
Navigator 窗口 131
New Breakpoint 161
New Data Breakpoint (新建数据断点) 125
New File 128, 156
New Project 128, 156
New Run Time Watch 161
New Run Time Watch (新建运行时观察) 125
New Runtime Watch 179
New Watch 125, 126, 161, 179
Next Bookmark 159
Next Bookmark (下一个书签) 127
Next Error 159
Next Error (下一个错误) 129
- O**
- Open File 129, 157
Open Project 128, 156
Open Recent File 157
Open Recent Project 156
Open Required Projects (打开所需的项目) 128
Open Team Project 156
Output 窗口 130
- P**
- Package 174
Page Setup 157
Palette 窗口 131
Paste 157
Paste Formatted 157
Paste (粘贴) 127
Pause 126, 161
PC Profiling (PC 性能分析) 125
PDF 153
PIC AppIO 125
Plugins 162
Previous Bookmark 159
Previous Bookmark (上一个书签) 127
Previous Error 159
Previous Error (上一个错误) 129
Print 157
Print to HTML 157
Program Device for Debugging Main Project
(针对调试主项目而对器件编程) 125
Programmer to Go PICkit 3 Main Project
(Programmer to Go PICkit 3 主项目) 128
Programmer 菜单变化 152
Project Group 157
Project Properties 157, 175
Project Properties 窗口 130
-

Project 菜单变化.....	151	Scan for external changes	159
Projects	128	Search 窗口	131
Projects 窗口视图	183	Select All	157
跑表	110	Select Identifier.....	157
配置, 多个	121	Select in Classes (在类中选择).....	159
配置位	96, 189	Select in Favorites	159
存储器窗口	95	Select in Files	159
教程.....	54	Select in Projects.....	159
如何使用.....	96	Services 窗口.....	130
Q		Sessions 窗口.....	130
Quick Search (快速搜索)	127	Set Configuration.....	175
器件 ID.....	172	Set Main Project	160
R		Set PC at Cursor	161
Read Device Memory.....	175	Set PC at Cursor (将 PC 设置在光标位置)	125
Read Device Memory Main Project		Set PC to Cursor	125
(读取器件存储器主项目)	128	Set Project Configuration	160
Read Device Memory to File.....	128	SFR	13, 95
Redo.....	157	Shell 脚本	107
Redo (重做)	127	Shift Left	159
Refactor		Shift Right.....	159
Change Function Parameter	160	Show binary files in Project view	176
Copy.....	160	Show Documentation	159
Move	160	Show Method Parameters.....	159
Redo	160	Show profiler indicators during run	176
Rename	160	Show Profiler Metrics	158
Safely Delete.....	160	Silent build.....	177
Undo	160	Simulator Analyzer 窗口	130
Refactor Preview 窗口	131	Simulator Stimulus	130
Refactor 菜单.....	160	SN (序列号)	67
Remote Terminal 窗口.....	129, 130	Source 菜单	159
Remove breakpoints upon importing a file	177	Sources 窗口	130
Remove Trailing Spaces	159	SQL History	129
Repeat Build/Run	160	SQL, 保留先前选项卡	129
Replace	157	Start Page	32, 127, 164
Replace in Projects	157	Start Sampling IDE (开始采样 IDE)	127
Replace in Projects (在项目中替换)	127	Status Toolbar Action (状态工具条操作)	126
Reset.....	161	Step Instruction	161
Reset Windows	164	Step Instruction (单步执行指令).....	126
Reset (复位)	125	Step Into.....	126, 161
Run Debugger/Programmer Self Test.....	161	Step Over	126, 161
Run File	160	Step Over Expression (Step Over 表达式)	126
Run File (运行文件)	128	Step Out	126
Run Main Project (运行主项目)	128	Stop Build/Run	160
Run Project	128, 160	Stopwatch.....	130
Run SQL	129	Subversion	162
Run Time Update Interval	179	Synchronize Editor with Views	158
Run to Cursor	126, 161	删除所有观察.....	179
Run 菜单.....	160, 162	闪存存储器	95
Run 图标.....	125	设置编译属性.....	84
日志文件.....	124	设置项目文件夹	177
日志文件选项.....	77	十六进制文件递增地址	84
S		数据 EEPROM.....	13
Save	129, 157	数据存储器	95
Save All	157	T	
Save All Modified Files Before Running Make	176	Tasks 窗口.....	131
Save All (全部保存)	129	Team Chat 窗口.....	131
Save As.....	157	Team 菜单	162
Scan for compilers.....	176	Templates.....	162
		Terminal 窗口	129, 130
		Test File.....	160

-
- Test File (测试文件) 128
 Test Project 160
 Test Project (测试项目) 128
 Test Results 窗口 130
 Threads 窗口 130
 Toggle Bookmark 159
 Toggle Comment 159
 Toggle Line Breakpoint 161
 Toolbars 158
 Tools 菜单变化 152
 Trace 窗口 130
 添加工具条按钮 125
 调试器菜单变化 152
 团队项目 131
- U**
- Undo 127, 157
 USB 217
 设备驱动程序, 安装 28
- V**
- Variables (变量) 窗口 59, 94
 Variables 窗口 130
 View 菜单 158
 View 菜单变化 150
- W**
- Watches (观察) 窗口 59, 92
 Watches 窗口 130
 二进制格式 179
 十进制格式 179
 十六进制格式 179
 字符格式 179
 Web Browser 158
 Web Browser (网络浏览器) 130
 Window 菜单 162
 Window 菜单变化 153
 外设 15
 外设存储器 95
 网站, Microchip 200
 文档
 编排 7
 约定 8
 文件寄存器 95
 文件路径选项 176
 文件向导 77
 文件属性 83, 175
 问题日志记录 124
- X**
- XC16 工具链 69
 XC32 工具链 69
 XC8 工具链 69
 XML
 Check DTD (检查 DTD) 131
 Check File 160
 Check File (检查文件) 131
 Validate File 160
 Validate File (验证文件) 131
 XSL Transform (XSL 转换) 131
 XML 文件类型 107
- 显示工具条 158
 相对路径 186
 响应文件 143
 向项目中添加库和其他文件 81
 向项目中添加现有文件 50, 79
 项目, 多个 119
 项目属性 129
 调试器 / 编程器 47, 74
 默认 46, 73
 语言工具 48, 75
 项目向导 39, 64, 77, 78
 协作 115
 行断点 88
 性能分析 112
- Y**
- 移动项目 186
 硬件工具
 活动连接 177
 硬件工具选择 67
 用户 ID 95
 语言工具位置 49, 76
 语言工具选项 48, 75
 预编译项目 102
 元组, 断点 90
 运行代码 56, 86
- Z**
- Zip 项目文件 174
 Zoom In (放大) 130
 Zoom Out (缩小) 130
 在复位向量处停止 177
 在主程序处停止 177
 找不到文件 —— 链接器错误 142
 指示灯
 编译器 69
 硬件和软件工具 67
 状态条 167
 桌面 32, 155
 窗格 45, 72
 组件 155
 自检 161
 自述文件 9
-

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://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

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

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

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

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
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: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

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

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区
Tel: 852-2943-5100
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: 86-755-8864-2200
Fax: 86-755-8203-1760

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

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

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

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

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-3666
Fax: 886-3-5770-955

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

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-3019-1500

日本 Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

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 - Dusseldorf
Tel: 49-2129-3766400

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Pforzheim
Tel: 49-7231-424750

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Venice
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

波兰 Poland - Warsaw
Tel: 48-22-3325737

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820