



MPLAB[®] C18

C 编译器

入门

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更安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 **Microchip** 数据手册中规定的操作规范来使用 **Microchip** 产品的。这样做的人极可能侵犯了知识产权。
- **Microchip** 愿与那些注重代码完整性的客户合作。
- **Microchip** 或任何其它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它受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

本出版物中所述的器件应用信息及其它类似内容仅为建议, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。**Microchip Technology Inc.** 不会就这些信息的准确性或使用方式作出任何陈述或保证, 也不会对因使用或以其它方式处理这些信息而引发的侵犯专利或其它知识产权的行为承担任何责任。未经 **Microchip** 书面批准, 不得将 **Microchip** 的产品用作生命维持系统中的关键组件。在知识产权保护下, 不得暗中以其它方式转让任何许可证。

商标

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

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

Analog-for-the-Digital Age、Application Maestro、dsPICDEM、dsPICDEM.net、dsPICworks、ECAN、ECONOMONITOR、FanSense、FlexROM、fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、Migratable Memory、MPASM、MPLIB、MPLINK、MPSIM、PICKit、PICDEM、PICDEM.net、PICLAB、PICKtail、PowerCal、PowerInfo、PowerMate、PowerTool、rfLAB、rfPICDEM、Select Mode、Smart Serial、SmartTel 和 Total Endurance 均为 **Microchip Technology Inc.** 在美国和其它国家或地区的商标。

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

在此提及的所有其它商标均为各持有公司所有。

© 2004, **Microchip Technology Inc.** 版权所有。

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

***Microchip** 位于美国亚利桑那州Chandler和Tempe及位于加利福尼亚州Mountain View的全球总部、设计中心和晶圆生产厂均于2003年10月通过了ISO/TS-16949:2002质量体系认证。公司在PICmicro® 8位单片机、KEELOQ® 跳码器件、串行EEPROM、单片机外设、非易失性存储器 and 模拟产品方面的质量体系流程均符合ISO/TS-16949:2002。此外, **Microchip** 在开发系统的设计和生产方面的质量体系也已通过了ISO 9001:2000 认证。*

目录

前言	1
第 1 章 概述	
1.1 简介	7
1.2 主要内容	7
1.3 系统要求	7
1.4 快速目录浏览	8
1.5 关于语言工具	9
第 2 章 安装	
2.1 简介	11
2.2 主要内容	11
2.3 安装 MPLAB C18	11
2.4 卸载 MPLAB C18	17
第 3 章 使用示例	
3.1 简介	19
3.2 主要内容	19
3.3 示例 1	20
3.4 示例 2	39
3.5 示例 3	43
3.6 示例 4	47
3.7 示例 5	51
术语表	53
索引	59
全球销售及服务网点	60

注:

前言

概述

本文档旨在帮助用户学习和使用 Microchip 的 MPLAB[®] C18 C 编译器。

主要内容

本章主要论述以下内容：

- 关于本指南
- 推荐读物
- Microchip 网站
- 开发系统客户通知服务
- 客户支持

关于本指南

文档内容编排

本文档讲述了如何安装 / 卸载 MPLAB C18，并提供了为 PICmicro[®] 单片机应用编写 C 代码的一些示例。若需要详细了解 MPLAB IDE v6.xx 的基本功能，请参考 MPLAB IDE 在线帮助文件。

本文档包括：

- **第 1 章：概述** — 介绍编译器的系统要求，并对所安装程序和安装过程中所创建的目录进行了简要描述。
- **第 2 章：安装** — 介绍了编译器的安装及卸载。
- **第 3 章：使用示例** — 以教程的形式讲解如何有效地使用 MPLAB C18 C 编译器。所有示例中均使用 MPLAB IDE v6.xx，所选择器件为 PIC18F452 或 PIC18F4620，并以 MPLAB SIM 软件模拟器作为调试工具。部分示例还使用了 MPLAB ICD 2 在线调试器和 PICDEM[™] 2 Plus 演示板。
 - **示例 1** 说明如何建立并编译一个项目，如何运行、单步运行及在代码中设置断点，以及调试代码。
 - **示例 2** 说明如何使用 MPLAB C18 外设函数库和 C 标准函数库，以及变量在程序存储器中的分配。
 - **示例 3** 说明扩展模式和非扩展模式的一些差别。
 - **示例 4** 说明变量在存取 RAM 中的分配。
 - **示例 5** 说明如何在 MPLAB C18 中使用中断服务程序，并举例说明了 MPLAB C18 外设函数库的使用。

- **术语表** — 本指南中使用的术语。
- **索引** — 本文档中出现的术语、要素以及各个章节的交叉引用列表。

本指南使用的约定

本指南使用如下文档约定：

文档约定

描述	表示	示例
代码（Courier 字体）：		
Courier 字体	示例源代码	<code>distance -= time * speed;</code>
	文件名及路径	<code>c:\mcc18\h</code>
	关键字	<code>_asm, _endasm, static</code>
	命令行选项	<code>-Opa+, -Opa-</code>
斜体 Courier 字体	可变参数	<i>file.o</i> , 其中 <i>file</i> 可以是任何有效的文件名
方括号 []	可选参数	<code>mcc18 [options] file [options]</code>
省略号 ...	代替重复的文本	<code>var_name [, var_name...]</code>
	表示用户提供的代码	<code>void main (void)</code> <code>{ ...</code> <code>}</code>
<code>0xnnnn</code>	十六进制数 其中 <i>n</i> 代表十六进制位	<code>0xFFFF, 0x007A</code>
文档（Arial 字体）		
斜体字符	参考书籍	<i>MPLAB User's Guide</i>

文档更新

所有的文档都有过时的时候，本指南也不例外。为满足客户的需要，MPLAB IDE、MPLAB C18 及其它 Microchip 工具都在不断发展之中，本文档中对于对话框和 / 或工具的描述可能与实际有所差别。请登录我公司网站获得最新的文档。

文档命名约定

文档用“DS”号编号。DS 号位于每页页脚的页码之前。DS 号的命名约定为 DSXXXXXA 或 DSXXXXXA_CN，其中：

XXXXX = 文档号。
A = 文档的版本。
_CN = 文档为中文版。

推荐读物

要了解更多关于编译器的函数库和预编译目标文件、MPLAB IDE 及其它工具使用方面的信息，请阅读以下推荐读物。

readme.c18

关于使用 MPLAB C18 C 编译器的最新信息，请阅读本软件自带的 readme.c18 文件（ASCII 文本）。此 readme 文件包含了本文档可能未提供的更新信息。

readme.xxx

需要其它 Microchip 工具的最新信息（MPLAB IDE 和 MPLINK™ 链接器等），请阅读软件自带的相关 readme 文件（ASCII 文本文件）。

MPLAB® C18 C 编译器用户指南（DS51288C_CN）

一个综合指南，讲述了针对 PIC18 器件设计的 Microchip MPLAB C18 C 编译器的使用及特征。

MPLAB® IDE V6.XX 快速入门指南（DS51281C_CN）

介绍如何安装 MPLAB IDE 软件及如何使用 IDE 创建项目并烧写器件。

MPASM™ User's Guide with MPLINK™ Linker and MPLIB™ Librarian （DS33014）

这个用户指南描述了如何使用 Microchip 的 PICmicro 单片机汇编器（MPASM）、链接器（MPLINK）和库管理器（MPLIB）。

PICmicro® 18C 单片机系列参考手册（DS39500A_CN）

重点介绍增强型单片机系列。说明了增强型单片机系列架构和外设模块的工作原理，但没有涉及到每个器件的具体细节。

PIC18 Device Data Sheets and Application Notes

讲述 PIC18 器件工作和电气特性的数据手册。应用笔记介绍了如何使用 PIC18 器件。

要获得上述任何文档，请访问 Microchip 的网站（www.microchip.com），获得 Adobe Acrobat（.pdf）格式的文档。

MICROCHIP 网站

Microchip 在其网站上为您提供在线支持。客户很容易从 Microchip 网站上获得文件和信息。要访问此网站，您必须能访问互联网并要安装 Netscape Navigator[®] 或 Microsoft[®] Internet Explorer 等网络浏览器。

使用您喜欢的互联网浏览器，可以访问 Microchip 的网站：

<http://www.microchip.com>

网站提供多种服务。用户从网站上可以下载到最新开发工具的文件、数据手册、应用笔记、用户指南、文章和示例程序。也可以获得关于 Microchip 业务的具体信息，包括销售办事处、分销商和工厂代表的列表。

技术支持

- 常见问题（FAQ）
- 在线讨论组 — 关于产品、开发系统、技术信息及其它方面的讨论会。
- Microchip 顾问计划成员列表
- 到其它与 Microchip 产品相关的有用网站的链接

工程师工具箱

- 设计技巧
- 器件勘误表

其它信息

- 最新 Microchip 新闻稿
- 研讨会和活动列表
- 招聘职位

开发系统客户通知服务

Microchip 启动了客户通知服务，来帮助客户轻松获得关于 Microchip 产品的最新信息。订阅此项服务后，每当您指定的产品系列或感兴趣的开发工具有更改、更新、改进或有勘误时，您都会收到电子邮件通知。

登录 Microchip 网站 (<http://www.microchip.com>)，点击“客户变更通知”。按照指示注册。

开发系统产品组分类如下：

- 编译器
- 仿真器
- 在线调试器
- MPLAB IDE
- 编程器

下面是对这些类别的描述：

编译器 — 关于 Microchip C 编译器和其它语言工具的最新信息。这些工具包括 MPLAB[®] C17、MPLAB C18 和 MPLAB C30 C 编译器；MPASM[™] 和 MPLAB ASM30 汇编器；MPLINK[™] 和 MPLAB LINK30 目标链接器；MPLIB[™] 和 MPLAB LIB30 目标库管理器。

仿真器 — 关于 Microchip 在线仿真器的最新信息。包括 MPLAB ICE 2000 和 MPLAB ICE 4000。

在线调试器 — 关于 Microchip 在线调试器的最新信息，包括 MPLAB ICD 2。

MPLAB IDE — 关于 Microchip MPLAB[®] IDE 的最新信息，它是开发系统工具的 Windows[®] 集成开发环境。重点介绍 MPLAB IDE、MPLAB SIM 软件仿真器、MPLAB IDE 项目管理器以及一般的编辑和调试功能。

编程器 — 关于 Microchip 器件编程器的最新信息。编程器包括 MPLAB PM3 和 PRO MATE[®] II 器件编程器，以及 PICSTART[®] Plus 开发编程器。

客户支持

Microchip 产品的用户可通过下列渠道获得支持：

- 分销商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 应用工程师（CAE）
- 热线

客户可以致电其分销商、代表或应用工程师（FAE）寻求支持。请查看本手册最后一页的销售办事处及地址列表。

也可致电应用工程师（CAE）获得技术支持，美国地区请拨打 (480) 792-7627。中国大陆地区请拨打 800-820-6247。

此外还有系统信息和更新热线。此热线为系统用户提供开发系统软件产品最新版本的列表。此热线还提供有关客户如何获得目前更新工具包的信息。

热线号码为：

美国和加拿大大部分地区，请拨打 1-800-755-2345。

全球其它国家或地区，请拨打 1-480-792-7302。

第 1 章 概述

1.1 简介

本文档旨在帮助用户快速学会使用 Microchip 的 MPLAB[®] C18 C 编译器。使用 MPLAB C18 C 编译器、PIC18 PICmicro[®] 单片机、MPLINK[™] 链接器和 MPLAB IDE，用户就能轻松开发出 PICmicro 单片机应用程序。关于本文所提到功能的详细描述，请参阅《MPLAB[®] C18 C 编译器用户指南》（DS51288C_CN）。

1.2 主要内容

本章所包含信息：

- 系统要求
- 快速目录浏览
- 关于语言工具

1.3 系统要求

使用 MPLAB C18 和 MPLINK 链接器的最小系统要求为：

- 25 MB 硬盘空间（推荐 50 MB）
- Microsoft[®] Windows[®] 操作系统（95 或更新版本）

1.4 快速目录浏览

MPLAB C18 安装目录包含编译器的 `readme` 文件 (`readme.c18`) 和链接器的 `readme` 文件 (`readme.lkr`)，以及一些子目录。表 1-1 对此进行了详细描述。

表 1-1: MPLAB C18 子目录描述

目录	描述
bin	包含编译器和链接器的可执行程序。将在下一节中对这些可执行程序进行更详细的描述。
cpp	包含 MPLAB C18 C 预处理器的源代码，供有兴趣的用户参考。
doc	包含 MPLAB C18 的电子文档，有关 MPLAB C18 的问题可参考这些文档。
example	包含示例应用程序，帮助用户开始学习使用 MPLAB C18，其中包括本文档中使用的示例。
h	包含标准 C 函数库的头文件和所支持 PICmicro 单片机的特定处理器函数库的头文件。
lib	包含标准 C 函数库 (<code>clib.lib</code> 或 <code>clib_e.lib</code>)、特定处理器的函数库 (<code>p18xxxx.lib</code> 或 <code>p18xxxx_e.lib</code> ，其中 <code>xxxx</code> 是具体的器件型号) 和启动模块 (<code>c018.o</code> 、 <code>c018_e.o</code> 、 <code>c018i.o</code> 、 <code>c018i_e.o</code> 、 <code>c018iz.o</code> 和 <code>c018iz_e.o</code>)。
lkr	包含链接器描述文件。
mpasm	包含 MPASM 汇编器的命令行版本，MPLAB C18 所支持器件的汇编头文件 (<code>p18xxxx.inc</code>)，以及函数库所使用的汇编头文件。
src	包含标准 C 函数库、特定处理器函数库和启动模块的 C 和汇编源代码文件。

1.5 关于语言工具

MPLAB C18 编译器安装目录下的 `bin` 和 `mpasm` 子目录包含 MPLAB C18、MPASM 汇编器和 MPLINK 链接器的可执行程序。这些程序的简要描述参见表 1-2。

表 1-2: MPLAB C18、MPASM 汇编器和 MPLINK 链接器可执行程序

可执行程序	描述
<code>mcc18.exe</code>	编译器 <code>shell</code> 。它以 C 文件（即 <code>file.c</code> ）作为输入，调用扩展模式和非扩展模式编译器可执行程序。
<code>mcc18-extended.exe</code>	扩展模式编译器可执行程序，在扩展模式编译时由编译器 <code>shell</code> 调用。它调用预处理器 <code>cpp18.exe</code> 来预处理 C 文件，然后编译预处理过的输出文件，并生成一个 COFF 文件（如 <code>file.o</code> ）传递给链接器。
<code>mcc18-traditional.exe</code>	这是非扩展模式编译器可执行程序，在非扩展模式编译时由编译器 <code>shell</code> 调用。它调用预处理器 <code>cpp18.exe</code> 来预处理 C 文件，然后编译预处理过的输出文件，并生成一个 COFF 文件（如 <code>file.o</code> ）传递给链接器。
<code>cpp18.exe</code>	C 预处理器。
<code>mplink.exe</code>	链接器的驱动程序。它以链接器描述文件、目标文件和库文件作为输入，并把这些文件传递给 <code>_mplink.exe</code> 。然后它会把 <code>_mplink.exe</code> 输出的 COFF 文件传递给 <code>mp2cod.exe</code> 和 <code>mp2hex.exe</code> 。
<code>_mplink.exe</code>	链接器。它输入链接器描述文件（如 <code>p18f452.lkr</code> ）、目标文件和库文件，输出 COFF 可执行文件（如 <code>file.out</code> 或 <code>file.cof</code> ）。COFF 文件是对输入目标文件及由函数库引用的目标文件的数据和代码进行地址分配所产生的。 <code>_mplink.exe</code> 也可以选择生成一个映射文件（如 <code>file.map</code> ），此映射文件包含关于数据和代码分配的详细信息。
<code>mp2cod.exe</code>	把 COFF 文件变为 COD 文件的文件转换器。COD 文件是 MPLAB IDE v5.xx 使用的符号调试文件格式。 <code>mp2cod.exe</code> 的输入是由 <code>_mplink.exe</code> 生成的 COFF 文件，输出 COD 文件（如 <code>file.cod</code> ）。它也创建列表文件（如 <code>file.lst</code> ），列表文件列出源代码与机器代码之间的对应关系。
<code>mp2hex.exe</code>	把 COFF 文件转化为 HEX 文件的文件转换器。HEX 文件是 PICmicro 单片机编程器（如 PICSTART Plus 或 PROMATE II）可读的文件格式。 <code>mp2hex.exe</code> 的输入是由 <code>_mplink.exe</code> 生成的 COFF 文件，输出是 HEX 文件（如 <code>file.hex</code> ）。
<code>mplib.exe</code>	库管理器。它允许创建和管理库文件（如 <code>file.lib</code> ），而库文件则充当目标文件的存档文件。库文件用于将目标文件组织成可重用的代码库。

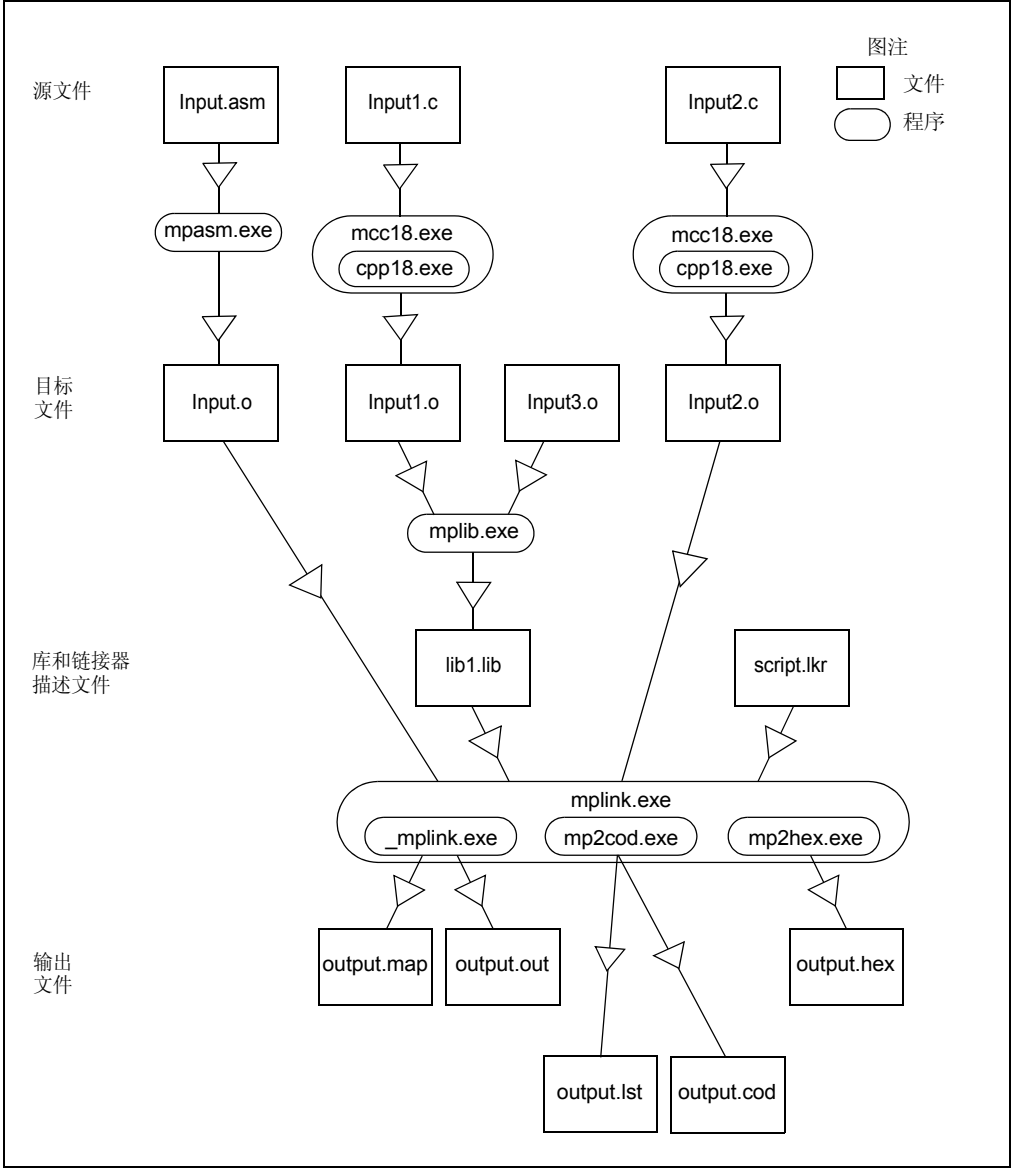
表 1-2: MPLAB C18、MPASM 汇编器和 MPLINK 链接器可执行程序 (续)

mpasm.exe	命令行汇编器。它的输入是汇编源文件 (如 file.asm), 输出是 COFF 文件 (如 file.o), 或 HEX 文件和 COD 文件 (如 file.hex 和 file.cod)。它也生成列表文件 (如 file.lst) 和错误文件 (如 file.err), 错误文件包含在汇编过程中发出的所有错误或警告。汇编源文件可能包含汇编头文件 (如 p18f452.inc), 汇编头文件也包含汇编源代码。
-----------	--

关于语言工具的更详细信息, 包括其命令行用法, 请参考《MPLAB® C18 C 编译器用户指南》(DS51288C_CN) 和 MPASM™ User's Guide with MPLINK™ and MPLIB™ (DS33014)。

图 1-1 说明了语言工具的执行流程。

图 1-1: 语言工具执行流程



第 2 章 安装

2.1 简介

本章详细讲述如何安装 MPLAB C18。有些情况下需要卸载软件，为此也提供了如何卸载软件的指示说明。

2.2 主要内容

本章提供的信息包括：

- 安装 MPLAB C18
- 卸载 MPLAB C18

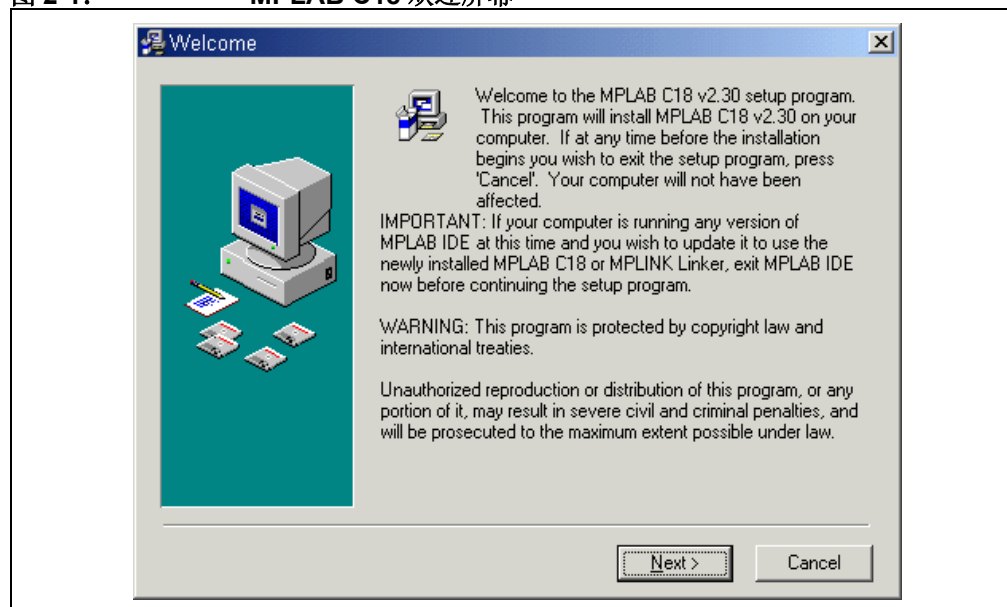
2.3 安装 MPLAB C18

要安装 MPLAB C18，请运行 CD-ROM 中的安装程序。如果要升级 MPLAB C18，请运行从 Microchip 网站上下载的升级安装程序。在整个安装过程中，一系列的对话框将一步一步地指导您如何进行安装。

2.3.1 欢迎

首先会出现一个欢迎屏幕，告知安装程序要安装的 MPLAB C18 的版本号，如图 2-1 所示。

图 2-1: MPLAB C18 欢迎屏幕

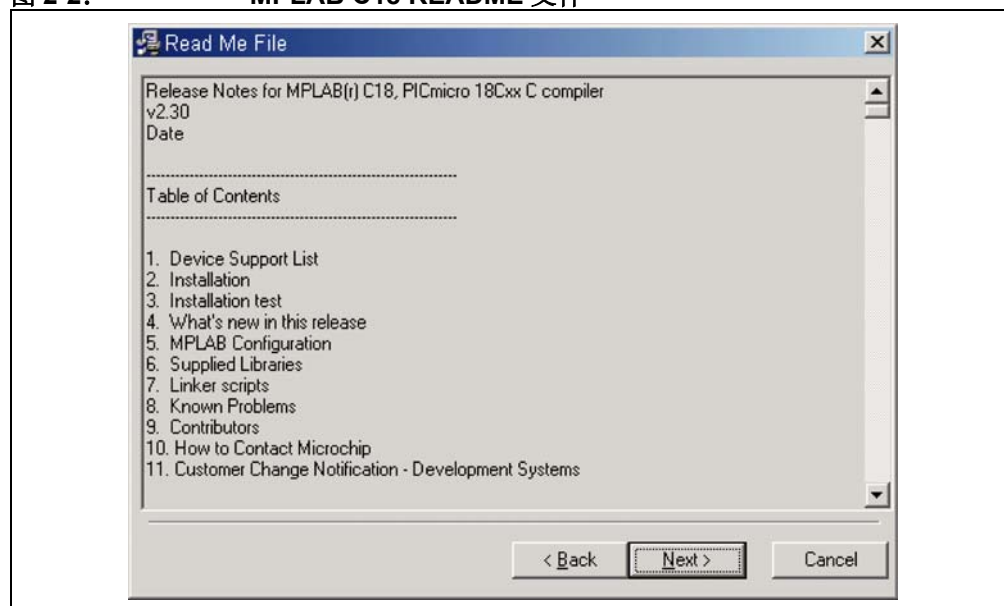


点击 **Next** 继续安装。

2.3.2 Readme 文件

随后将显示 MPLAB C18 的 readme 文件。这个文件包含关于此版本 MPLAB C18 的重要信息，如已知的缺陷，如图 2-2 所示。

图 2-2: MPLAB C18 README 文件



看完后，点击 **Next** 继续安装。

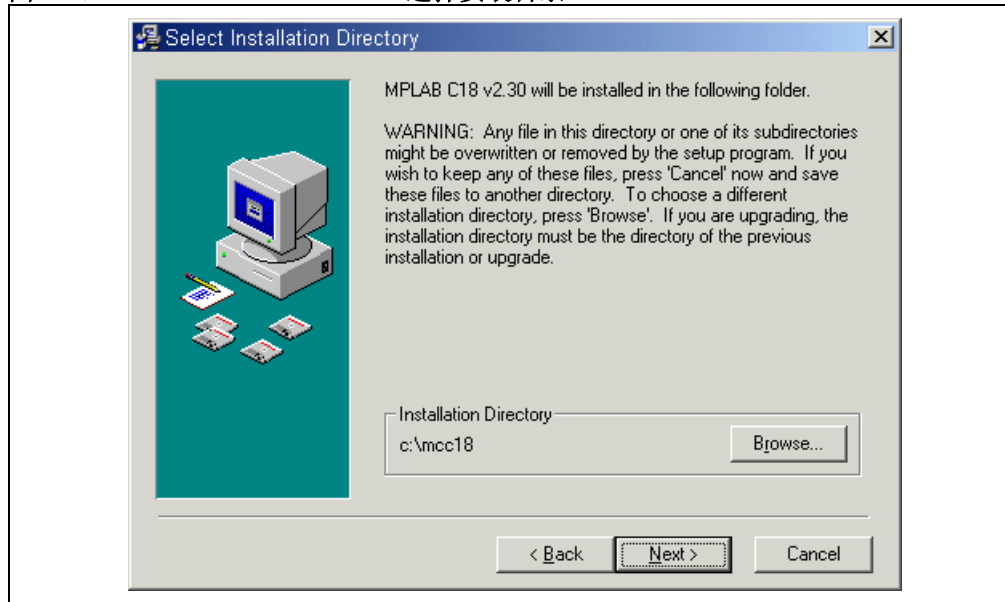
2.3.3 选择安装目录

这一步要求用户选择安装 MPLAB C18 的目录。当第一次安装 MPLAB C18 时，默认的安装目录是 C:\mcc18，如图 2-3 所示。

如果是安装升级程序，安装程序则会把默认安装目录设置为上次安装时的目录。在升级时，所选的安装目录必须是上次安装或升级时的安装目录。

注： 在安装过程中，安装目录和其子目录中的文件可能会被覆盖或删除。若要保存任何文件，例如要保存上次安装修改过的链接器描述文件或库源代码，则可在继续安装前把这些文件复制到安装目录外的目录中去。

图 2-3: MPLAB C18 选择安装目录



指定目录后请点击 **Next**。

2.3.4 选择组件

选中相应的复选框，选择需要安装的组件，参见图 2-4。

图 2-4: MPLAB C18 选择组件

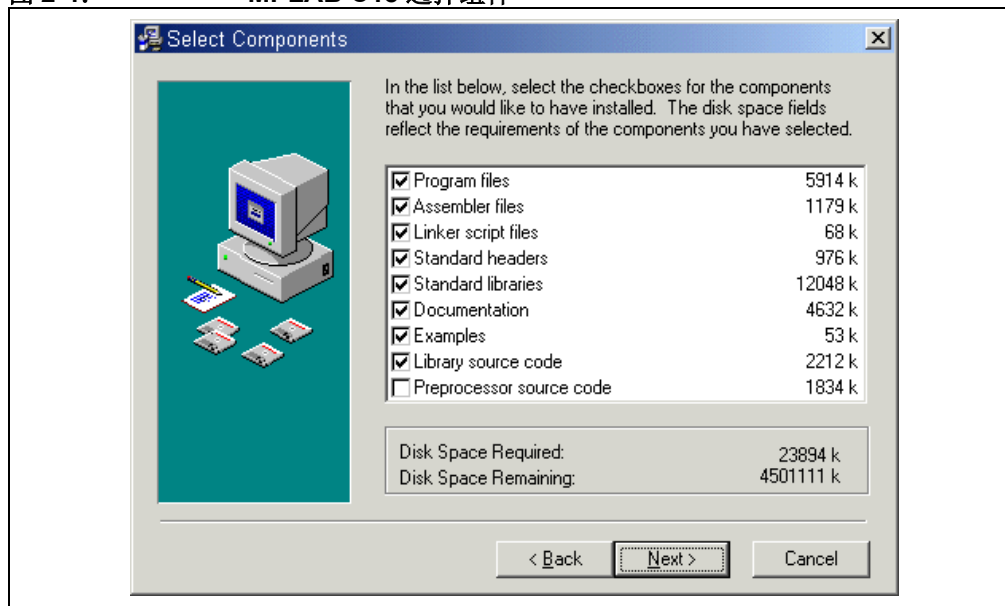


表 2-1 给出了所提供组件的详细描述。

表 2-1: MPLAB C18 软件组件

组件	描述
程序文件	编译器和链接器的可执行程序。用户应该选择安装这个组件，除非是升级安装，希望继续使用上次安装版本的可执行程序。
汇编器文件	包括 MPASM 汇编器的命令行版本（mpasm.exe）、由 MPLAB C18 所支持器件的汇编头文件（p18xxxx.inc）和函数库使用的汇编头文件。
链接器描述文件	这些文件供 MPLINK 链接器使用。每个支持的 PICmicro 单片机都有一个这样的文件。此文件为处理器提供了一个默认的存储配置，并指引链接器在处理器的存储器中分配代码和数据。 这些链接器描述文件有别于 MPLAB IDE 自带的链接器描述文件，是专门为 MPLAB C18 设计的。由于 MPLINK 链接器需要链接器描述文件，所以用户应该安装这个组件，除非他们打算创建自己的链接器描述文件。
标准头文件	这些是标准 C 函数库和特定处理器函数库的头文件。如果用户选择安装标准函数库，那么也会安装这些头文件。
标准库	这个组件包含标准 C 函数库、特定处理器函数库和启动模块。请参阅《MPLAB® C18 C 编译器函数库》（DS51297C_CN）和《MPLAB® C18 C 编译器用户指南》（DS51288C_CN），以获取更多关于函数库和启动模块的信息。由于大多数典型的程序都使用函数库和启动模块，推荐用户安装这个组件。
文档	这是 MPLAB C18 的电子文档。
示例	这些是示例应用程序，其中包括本文档中讲述的示例，用来帮助用户学习使用 MPLAB C18。
库源代码	这是标准 C 函数库和特定处理器函数库的源代码。如果用户打算重建函数库，则应该安装这个组件。
预处理器源代码	这是预处理器的源代码，供有兴趣的用户参考。

选择需要安装的组件，点击 **Next**。

2.3.5 配置选项

接下来的对话屏幕允许用户为其系统选择具体的 MPLAB C18 配置选项，如图 2-5 所示：

图 2-5: MPLAB C18 配置选项

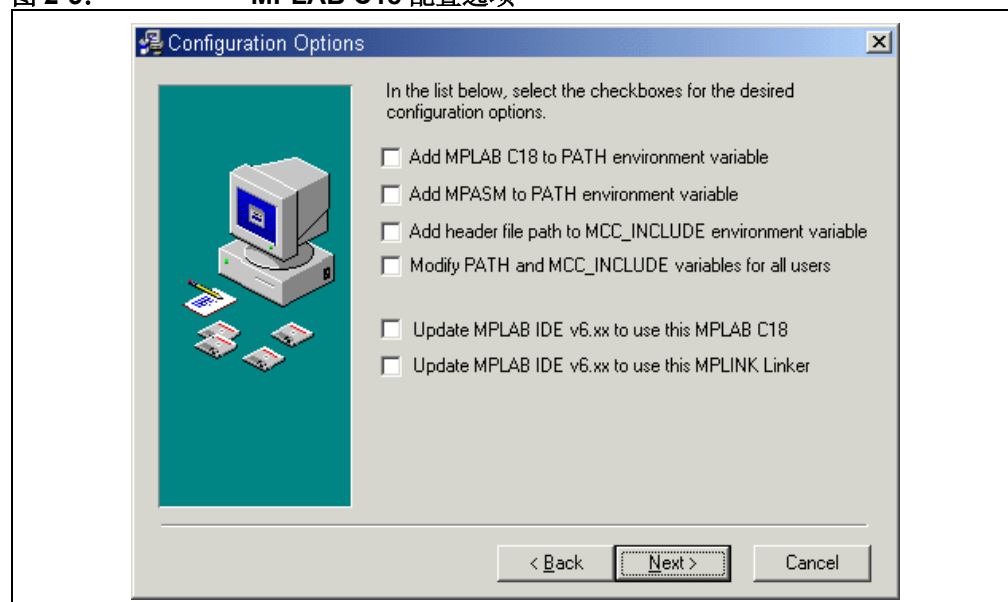


表 2-2 对各配置选项进行了详细描述。

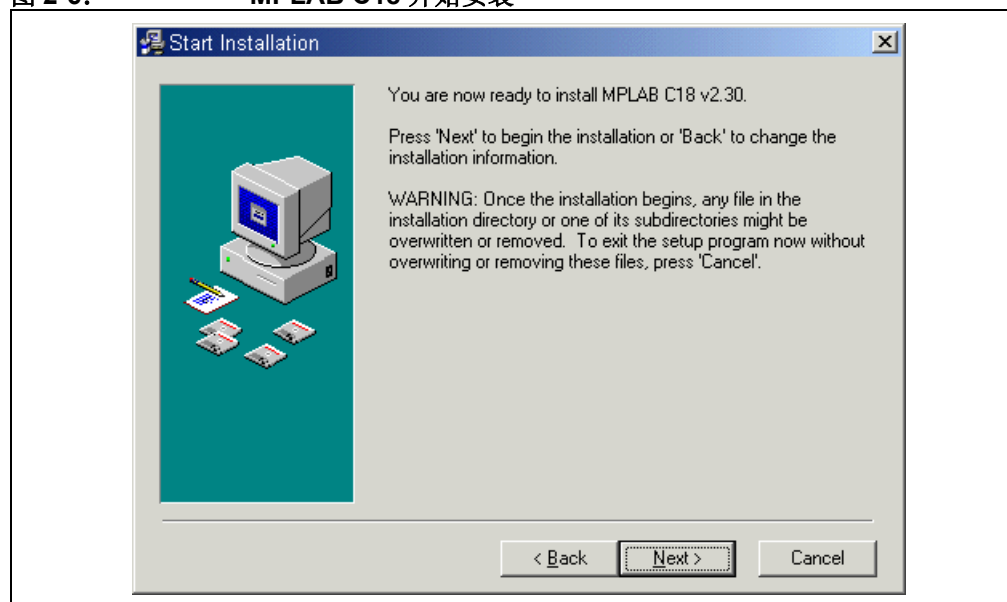
表 2-2: MPLAB C18 配置选项

配置	描述
Add MPLAB C18 to PATH environment variable	把 MPLAB C18 可执行程序（ <code>mcc18.exe</code> ）和 MPLINK 链接器可执行程序（ <code>mplink.exe</code> ）的路径加到 PATH 环境变量的前面。这样用户就能从任何目录、在命令 shell 提示符下启动新安装的 MPLAB C18 和 MPLINK 链接器。
Add MPASM to PATH environment variable	把 MPASM 可执行程序（ <code>mpasm.exe</code> ）的路径加到 PATH 环境变量的前面。这样用户就能从任何目录、在命令 shell 提示符下启动新安装的 MPASM 汇编器。
Add header file path to MCC_INCLUDE environment variable	把 MPLAB C18 头文件目录的路径加到 MCC_INCLUDE 环境变量的前面。如果此变量不存在，会创建它。MCC_INCLUDE 是一个由分号隔开的目录列表。如果 MPLAB C18 在由 -I 命令行选项指定的目录列表中找不到头文件，那么它将在 MCC_INCLUDE 目录列表中搜索头文件。选择该配置选项表明在包含标准头文件时，用户不必使用 -I 命令行选项。
Modify PATH and MCC_INCLUDE variables for all users	此选项只在用户以管理员身份登录 Windows NT 或 Windows 2000 计算机时出现。如果选择这个配置，则对前三个选项中指定的变量所做的修改会影响到所有的用户，否则只有当前用户的变量会受影响。
Update MPLAB IDE v6.xx to use this MPLAB C18	只有当系统安装有 MPLAB IDE v6.xx 时才会出现此选项。选择此选项将把 MPLAB IDE v6.xx 配置为使用新安装的 MPLAB C18。这包括在 MPLAB IDE v6.xx 中，使用 MPLAB C18 库目录作为 MPLAB C18 项目的默认库路径。
Update MPLAB IDE v6.xx to use this MPLINK linker	只有当系统安装有 MPLAB IDE v6.xx 时才会出现此选项。选择此选项将 MPLAB IDE v6.xx 配置为使用新安装的 MPLINK 链接器。

2.3.6 开始安装

接下来的对话屏幕将开始安装，如图 2-6 所示。一旦按下了 **Next** 按钮，安装目录及其子目录中的所有文件就会被覆盖或删除。

图 2-6: MPLAB C18 开始安装



2.3.7 完成安装

现在 MPLAB C18 已安装成功。在 “Installation Complete” 对话框中，点击 **Finish**。

为使 MPLAB C18 能正常运行，可能需要重新启动计算机。如果出现了 “Restart Computer” 对话框，可选择 **Yes** 立即重新启动计算机，或选择 **No**，以后再重新启动计算机。

2.4 卸载 MPLAB C18

要卸载 MPLAB C18，请打开 Windows 控制面板并运行 “Add/Remove Programs”（“添加 / 删除程序”）。在程序列表中选择安装的 MPLAB C18 程序，并按照指示来删除程序。这样就会从计算机中删除 MPLAB C18 目录及其中的内容。

注： 如果卸载 MPLAB C18 的升级版本，则全部安装内容都会被删除；MPLAB C18 不能恢复到升级前的版本。

注:

第 3 章 使用示例

3.1 简介

本章将举例说明如何有效地使用 MPLB C18，包括如何创建、编译项目和调试程序。这些示例都假定用户系统已经安装了 MPLAB C18 和 MPLAB IDE v6.xx。部分示例还假定用户系统已安装了 MPLAB ICD 2，并连接到了装有 PIC18F452 器件的 PICDEM[™] 2 Plus 演示板上。关于特定处理器的特殊功能寄存器、指令集和中断逻辑等方面的详细信息，请参阅《PIC18FXX2 数据手册》(DS39564B_CN)。

3.2 主要内容

本章中有关使用 MPLAB C18 的示例包括：

- **示例 1** 说明如何建立和编译项目，如何运行、单步运行示例代码和在示例代码中设置断点，以及如何调试代码。
- **示例 2** 说明如何使用 MPLAB C18 外设函数库和 C 标准函数库，并说明了变量在程序存储器中的分配。
- **示例 3** 说明扩展模式和非扩展模式的一些差别。
- **示例 4** 说明变量在存取 RAM 中的分配。
- **示例 5** 说明如何在 MPLAB C18 中使用中断服务程序，并提供了关于如何使用 MPLAB C18 外设函数库的示例。

3.3 示例 1

本示例是为使用 MPLAB IDE v6.xx、MPLAB SIM 软件模拟器和 PIC18F452 单片机而设计的。它说明了如何在 MPLAB IDE 中建立 MPLAB C18 项目，以及如何使用 MPLAB SIM 软件模拟器来编译项目和调试源代码。另外还示范了如何使用 MPLAB ICD 2 和 PICDEM 2 Plus 演示板来运行程序。此示例假定 c:\mcc18 是 MPLAB C18 的安装目录。

以下是此示例的源代码：

```
#include <p18cxxx.h>    /* for TRISB and PORTB
                        declarations */
/* Set configuration bits for use with ICD2 / PICDEM2 PLUS Demo Board:
 * - set HS oscillator
 * - disable watchdog timer
 * - disable low voltage programming
 */

#pragma romdata CONFIG
_CONFIG_DECL(_CONFIG1H_DEFAULT & _OSC_HS_1H,
             _CONFIG2L_DEFAULT,
             _CONFIG2H_DEFAULT & _WDT_OFF_2H,
             _CONFIG3H_DEFAULT,
             _CONFIG4L_DEFAULT & _LVP_OFF_4L,
             _CONFIG5L_DEFAULT,
             _CONFIG5H_DEFAULT,
             _CONFIG6L_DEFAULT,
             _CONFIG6H_DEFAULT,
             _CONFIG7L_DEFAULT,
             _CONFIG7H_DEFAULT);
#pragma romdata

int counter;
void main (void)
{
    counter = 1;
    TRISB = 0;          /* configure PORTB for output */
    while (counter <= 15)
    {
        PORTB = counter; /* display value of 'counter'
                           on the LEDs */
        counter++;
    }
}
```

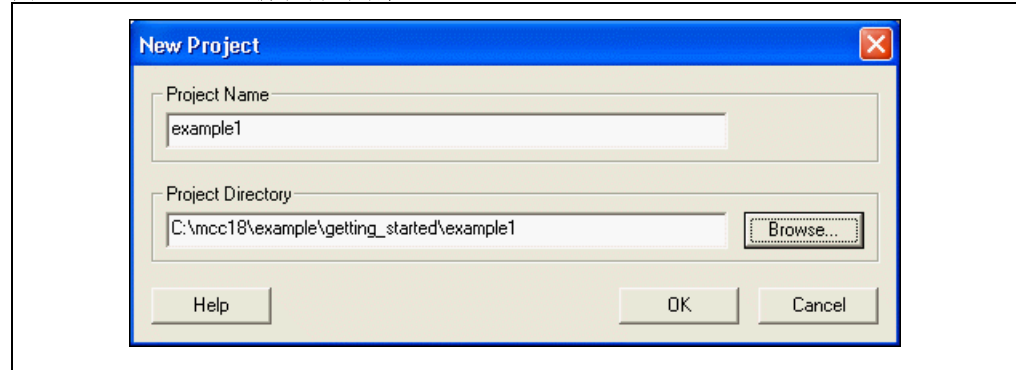
- TRISB 和 PORTB 是 PIC18F452 单片机的特殊功能寄存器。PORTB 的各引脚连接到 PICDEM 2 演示板上的各发光二极管；TRISB 引脚将 PORTB 引脚配置为输入（1）或输出（0）。
- 配置位需要正确设置；这是通过调用宏 _CONFIG_DECL 设置每个配置字节实现的。这包括指定 PICDEM 2 Plus 演示板上使用的振荡器；本示例中使用晶振（OSC_HS_1H）。此外，MPLAB ICD 2 要求禁止看门狗定时器和低电压编程（分别为 _WDT_OFF_2H 和 _LVP_OFF_4L）。设置配置位的宏在特定处理器的头文件中定义；具体细节请参阅《MPLAB[®] C18 C 编译器用户指南》（DS51288C_CN）。

3.3.1 建立项目

选择 **Project>New** 来创建一个新项目。然后在所显示的对话框中输入项目名及其路径，再点击 **OK**，如图 3-1 所示。

如果已经安装了 MPLAB C18 自带的示例，那么 MPLAB C18 安装目录下的 `example\getting_started\example1` 子目录中已包含本示例的源文件。

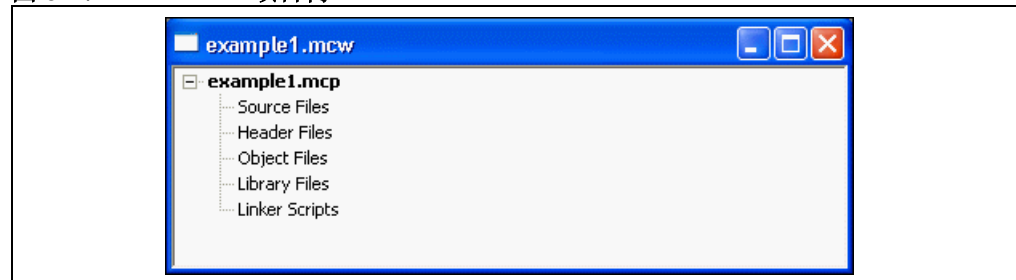
图 3-1: 新项目对话框



注：项目名不必和目录名相同。

现在会显示出项目树，每种类型项目文件对应一个分支。

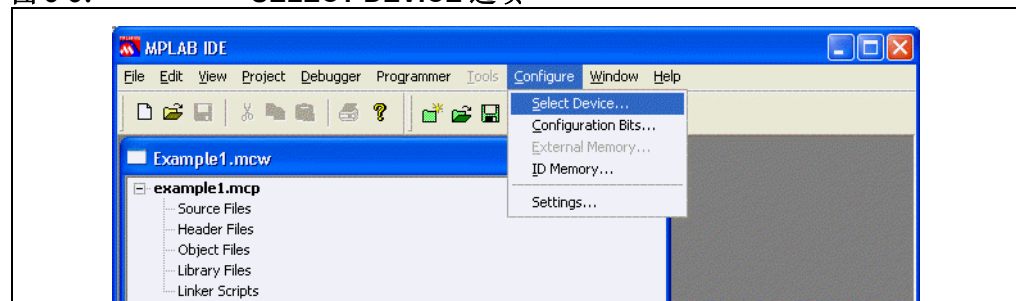
图 3-2: 项目树



3.3.2 选择目标处理器

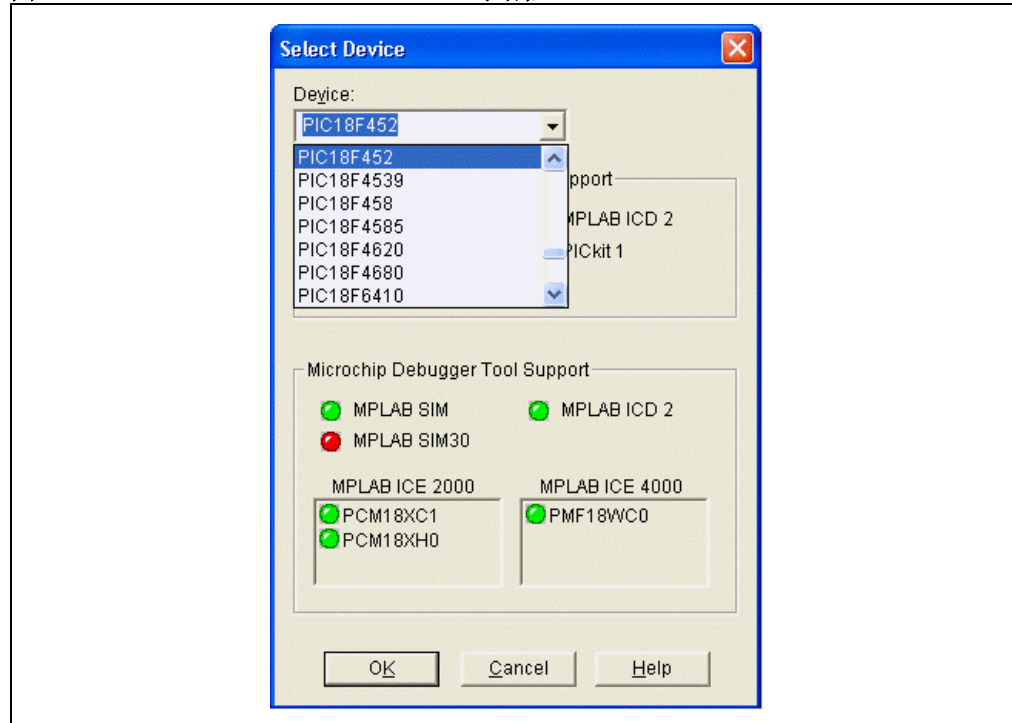
在对项目进行任何操作之前必须选择目标处理器。这可通过选择 **Configure>Select Device** 来完成。

图 3-3: SELECT DEVICE 选项



本例中使用的是 PIC18F452 单片机。选择要使用的器件并点击 **OK**。

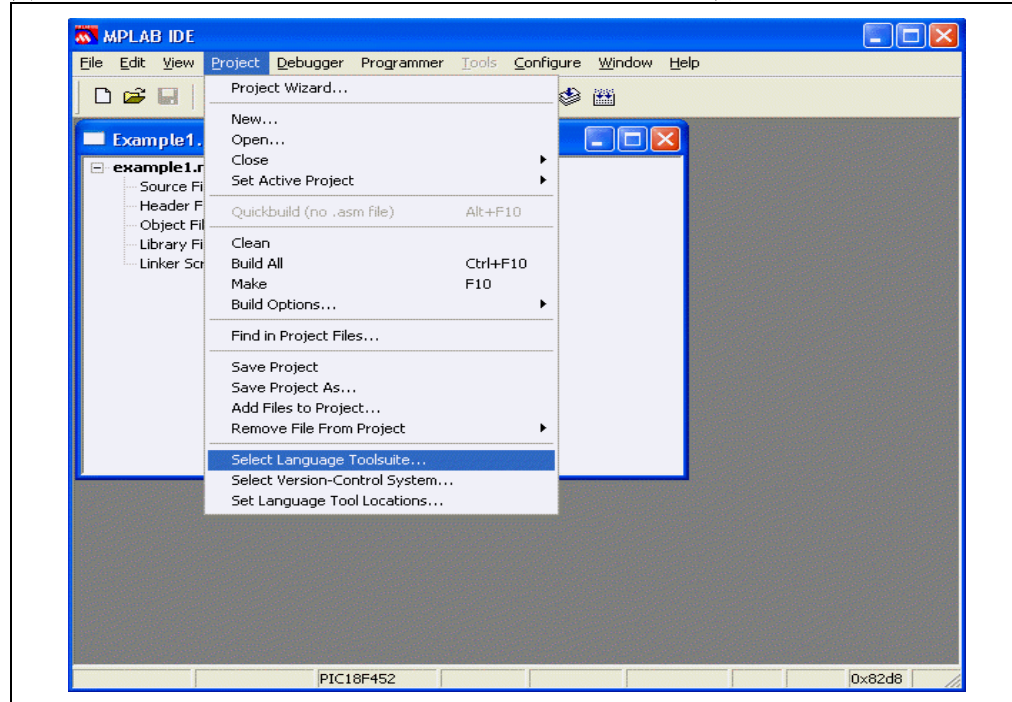
图 3-4: **SELECT DEVICE 对话框**



3.3.3 选择项目设置

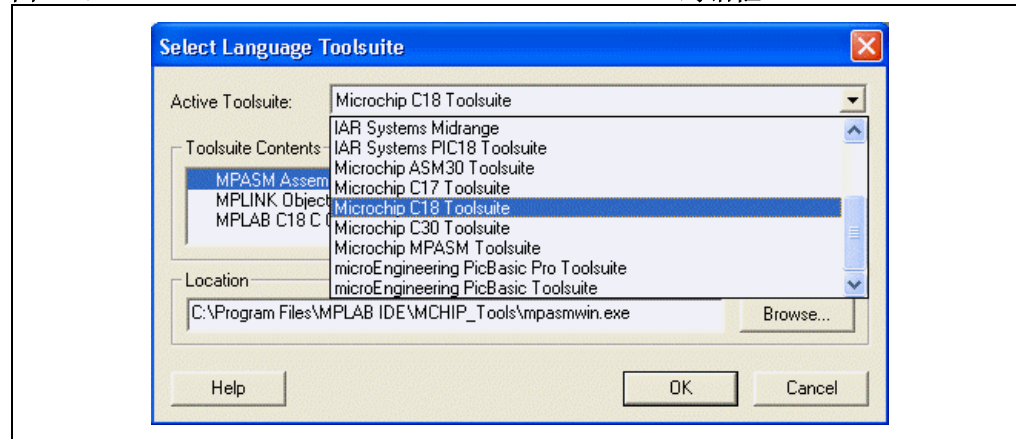
MPLAB IDE 需要知道使用哪个编译器和链接器。要选择 MPLAB C18 和 MPLINK 链接器，首先选择 **Project>Select Language Toolsuite**。

图 3-5: **SELECT LANGUAGE TOOLSUITE 选项**



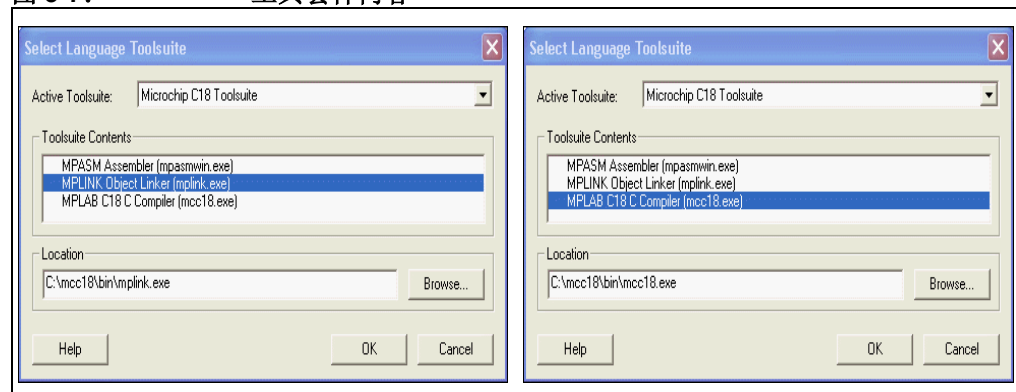
接下来出现选择语言工具套件的对话框。若要使用包括 MPLAB C18 和 MPLINK 链接器的语言工具，选择“Microchip C18 Toolsuite”作为有效工具套件。

图 3-6: SELECT LANGUAGE TOOLSUITE 对话框



然后，选择“MPLAB C18 C Compiler”和“MPLINK Object Linker”，并确保其路径分别指向新安装的可执行程序 `mcc18.exe` 和 `mplink.exe`。点击 **OK**。

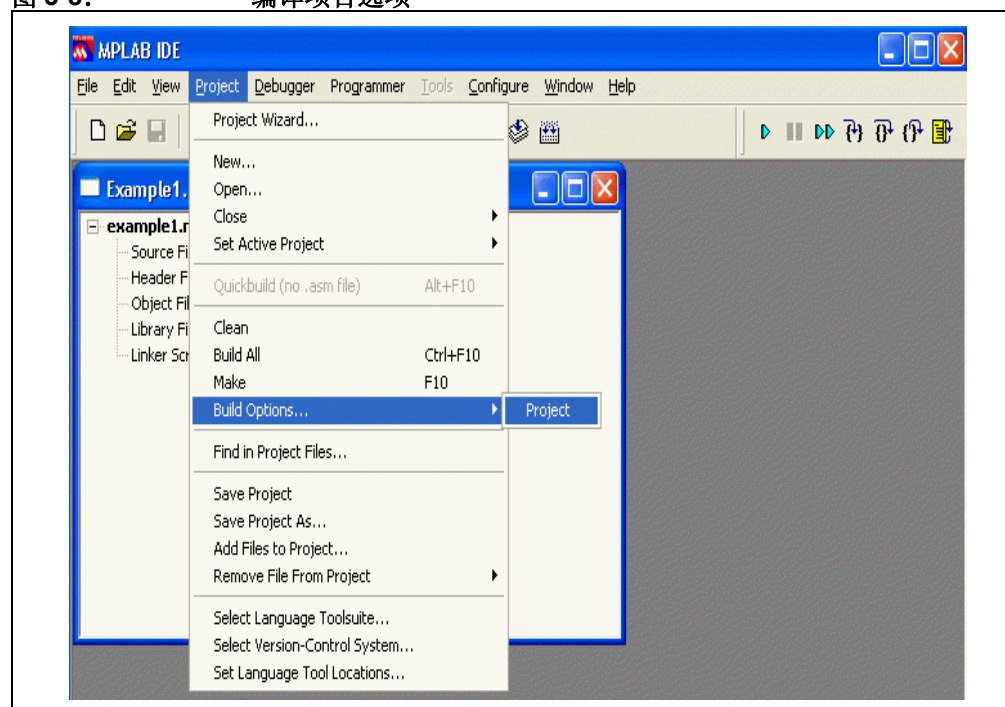
图 3-7: 工具套件内容



注： 如果用户选择升级 MPLAB IDE 6.xx，使用 MPLAB C18 安装程序中新安装的编译器和链接器，那么应该已正确设置了这些路径。

下一步是为编译器和链接器设定命令行选项。选择 **Project>Build Options>Project**。

图 3-8: 编译项目选项

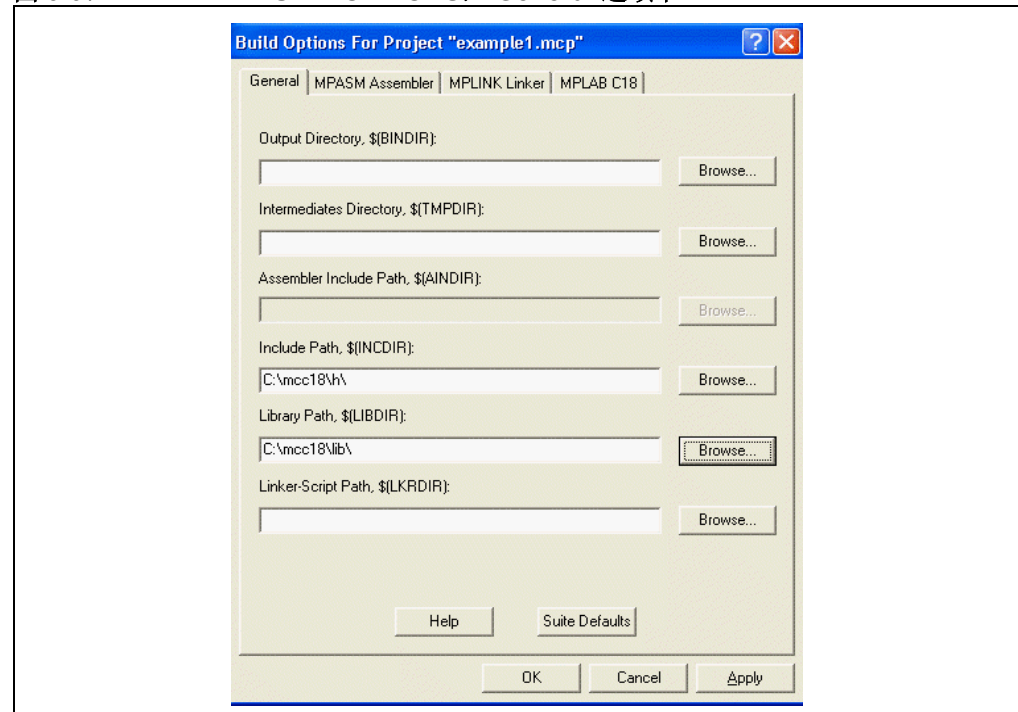


在图 3-9 所示的 “General” 选项卡中输入 MPLAB C18 安装目录中头文件和函数库子目录的路径。可以直接输入路径或点击 **Browse** 指定路径。MPLAB C18 会在指定的头文件目录下搜索包含的 .h 文件。MPLINK 链接器会在函数库目录下搜索目标文件和库文件，其中包括在链接器描述文件中指定的文件。“Output Directory” 中存放最终生成的 .cod、.cof 和 .hex 文件，这些文件只有在项目通过了完整的编译后才会生成。保持 “Output Directory” 域为空白，因此输出文件（example1.cof）将保存在项目目录中。

“Intermediates Directory” 中存放编译器生成的目标文件。保持此域空白，因此目标文件（example1.o）将保存在源文件所在的目录中。

MPLINK 链接器会在 “Linker-Script Path” 指定的目录中搜索链接器描述文件。由于链接器描述文件被添加到项目树时其位置会被指定，所以本例中此项可以为空白。

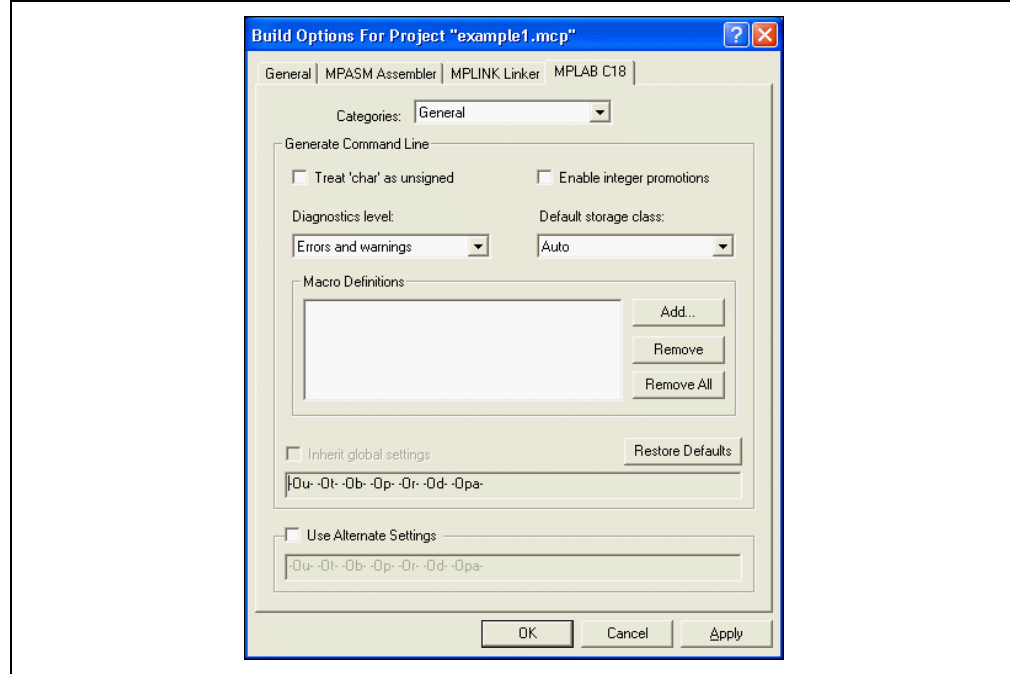
图 3-9: BUILD OPTIONS, General 选项卡



3.3.4 选择编译器和链接器设置

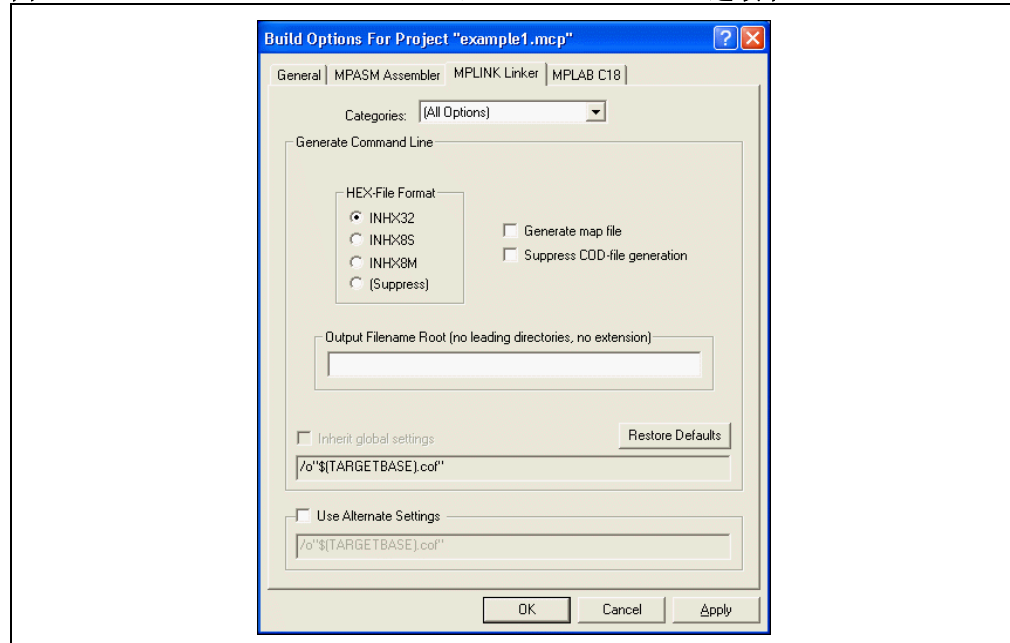
传递到编译器和链接器的各种命令行选项，可以分别在“Build Options”窗口的“MPLAB C18”和“MPLINK Linker”选项卡下设置。本示例接受 MPLAB C18 的默认命令行选项。

图 3-10: BUILD OPTIONS, MPLAB C18 选项卡



默认情况下，从 MPLAB IDE 运行 MPLINK 链接器时，不会生成映射文件（example1.map）。要改变此设置，可以在“MPLINK Linker”选项卡中选择“Generate map file”并点击 **OK**。其它命令行选项使用默认设置。

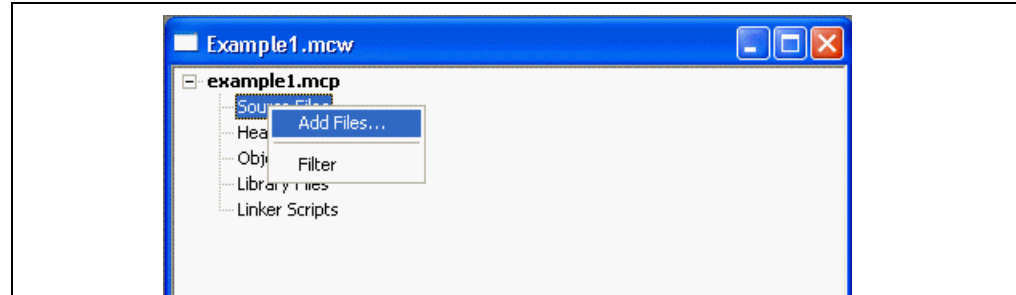
图 3-11: BUILD OPTIONS, MPLINK LINKER 选项卡



3.3.5 把文件添加到项目中

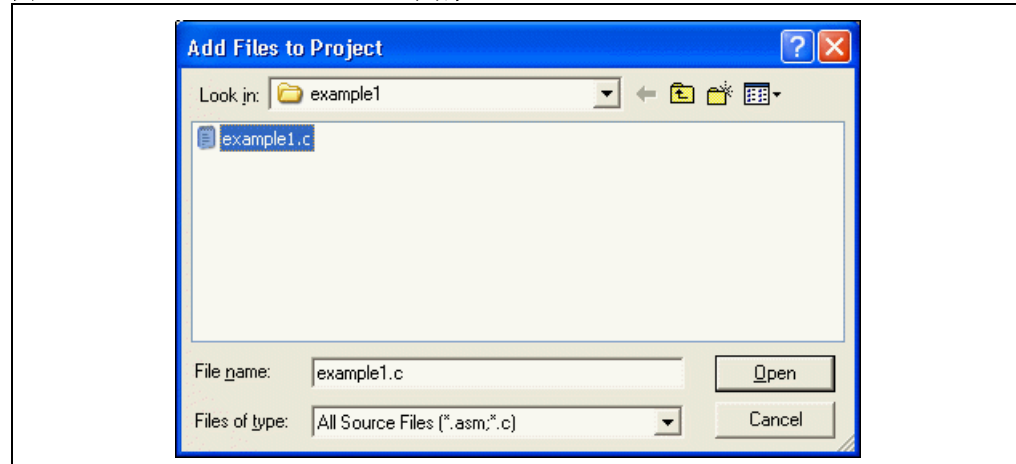
必须把 C 源文件添加到项目中。在项目窗口中的 “Source Files” 上单击鼠标右键，然后选择 “Add Files”。

图 3-12: SOURCE FILES, ADD FILES 选项



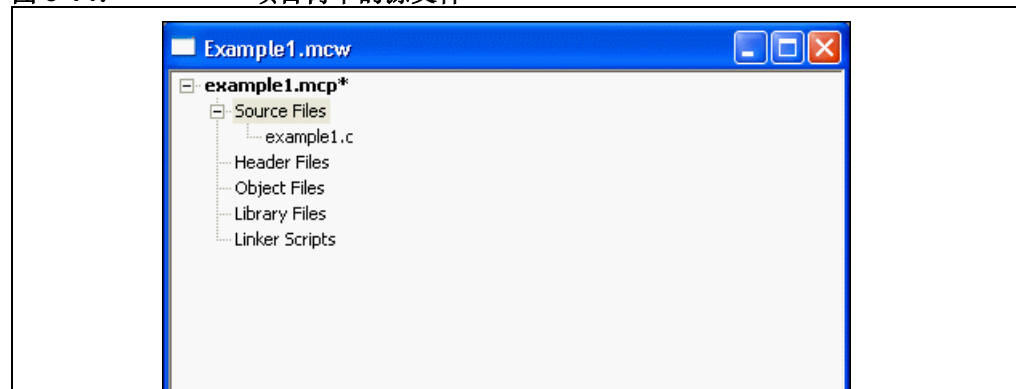
如果选择 `c:\mcc18\example\getting_started\example1` 作为项目目录，则源文件 `example1.c` 已经在此目录中。浏览至此目录并选择文件 `example1.c`。点击 **Open** 将文件添加到项目中。

图 3-13: ADD FILES 对话框



源文件应该出现在项目树中。

图 3-14: 项目树中的源文件

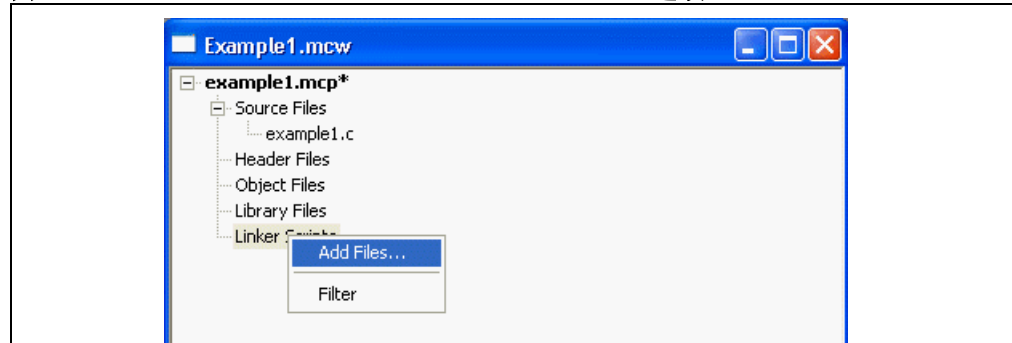


C 源文件中已包含了头文件，因此在项目树的 “Header Files” 中不需要添加任何文件。为方便查看文件，可将头文件添加到 “Header Files” 中，但只需把头文件包含到 C 源代码中就能编译项目。所需要的启动模块、标准函数库和处理器函数库在链接

器描述文件中指定，所以项目树的“Object Files”或“Library Files”中都不需要添加任何文件。如果项目中要链接其它目标文件或库文件，那么应将它们添加到这些分支下。

MPLINK 链接器要求指定一个链接器描述文件。用鼠标右键单击项目窗口中的“Linker Scripts”，并选择 **Add Files**。

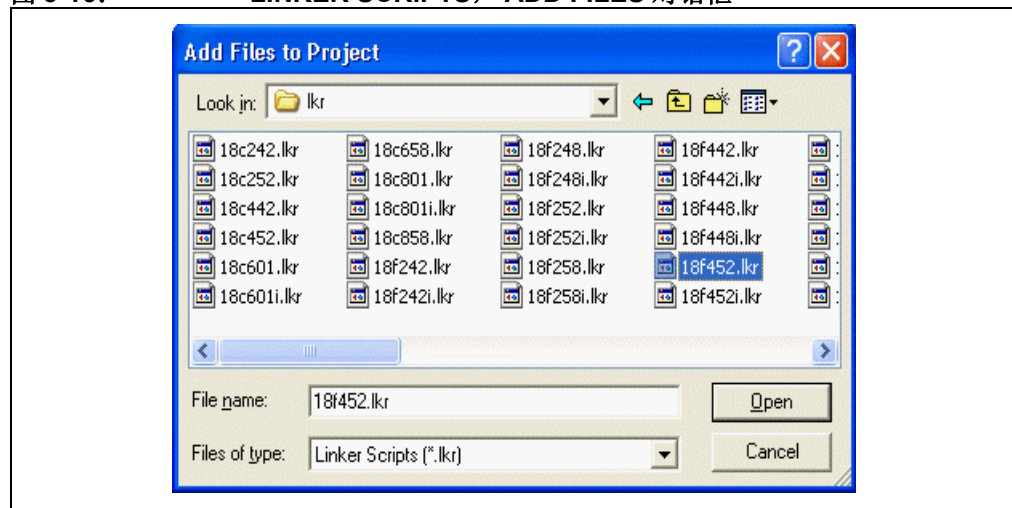
图 3-15: LINKER SCRIPTS, ADD FILES 选项



使用 MPLAB C18 安装目录下 `lkr` 子目录中的链接器描述文件 `18f452.lkr`。此描述文件适用于 PIC18F452 器件。

点击 **Open** 将文件添加到项目树中。

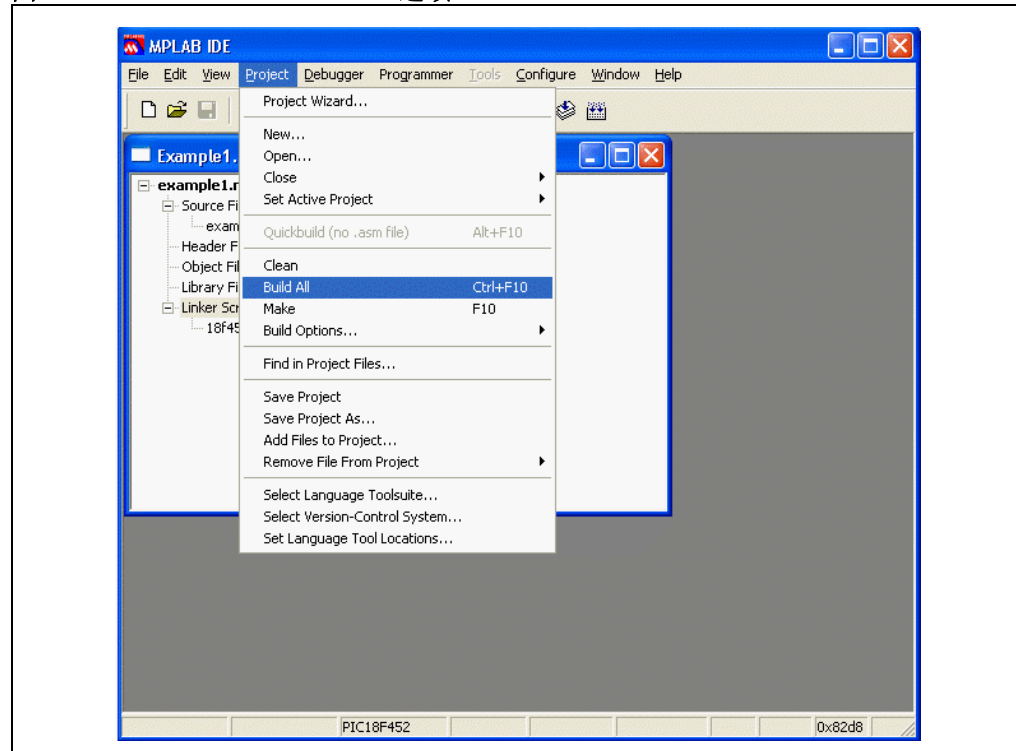
图 3-16: LINKER SCRIPTS, ADD FILES 对话框



3.3.6 编译项目

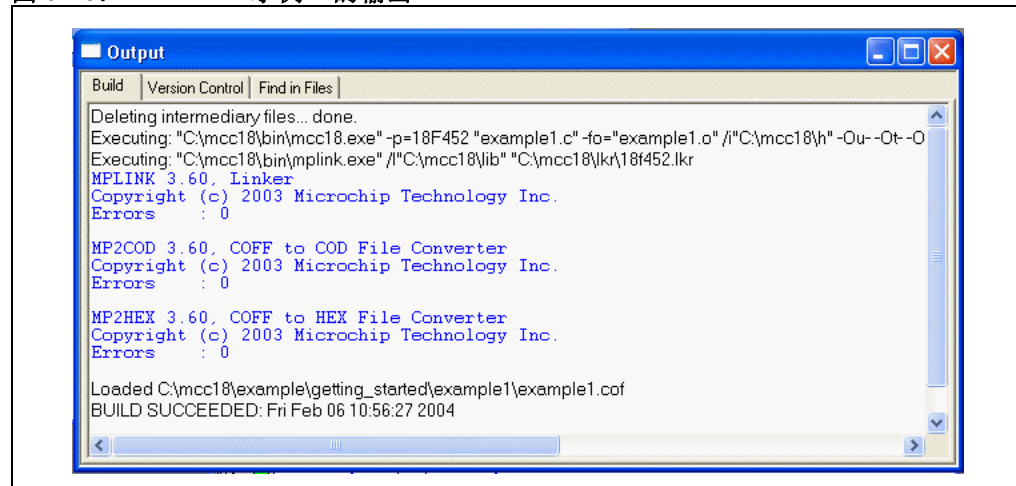
选择 **Project>Build All** 编译和链接项目。若有任何错误或警告消息，将会显示在 output（输出）窗口中。

图 3-17: BUILD ALL 选项



在本例中，output 窗口应该显示没有错误，且显示已成功生成输出文件的消息。如果出现任何错误，请查看一下源文件的内容与 3.3 节“示例 1”开头部分所给出的程序文本是否完全一致。

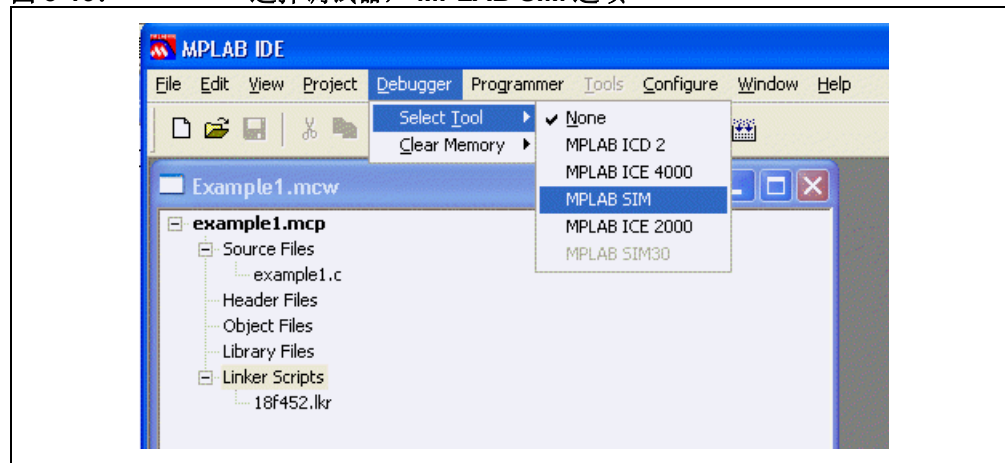
图 3-18: 示例 1 的输出



3.3.7 用 MPLAB SIM 软件模拟器进行调试

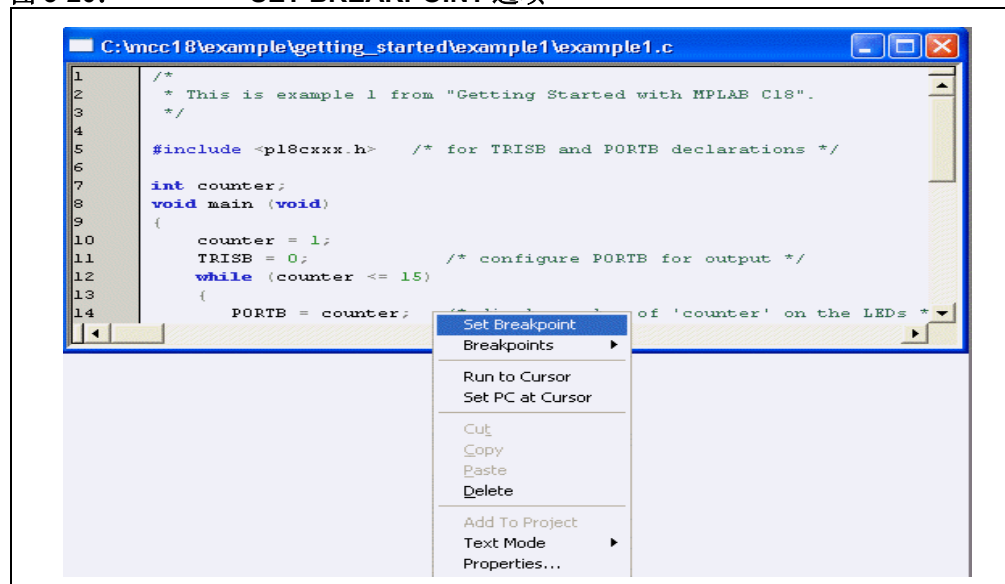
利用 MPLAB SIM 软件模拟器可以在源代码中设置断点，以便在 watch（观察）窗口中查看变量的值。首先，确保已经通过选择 **Debugger>Select Tool>MPLAB SIM** 选择了 MPLAB SIM 软件模拟器作为调试工具。

图 3-19: 选择调试器，MPLAB SIM 选项



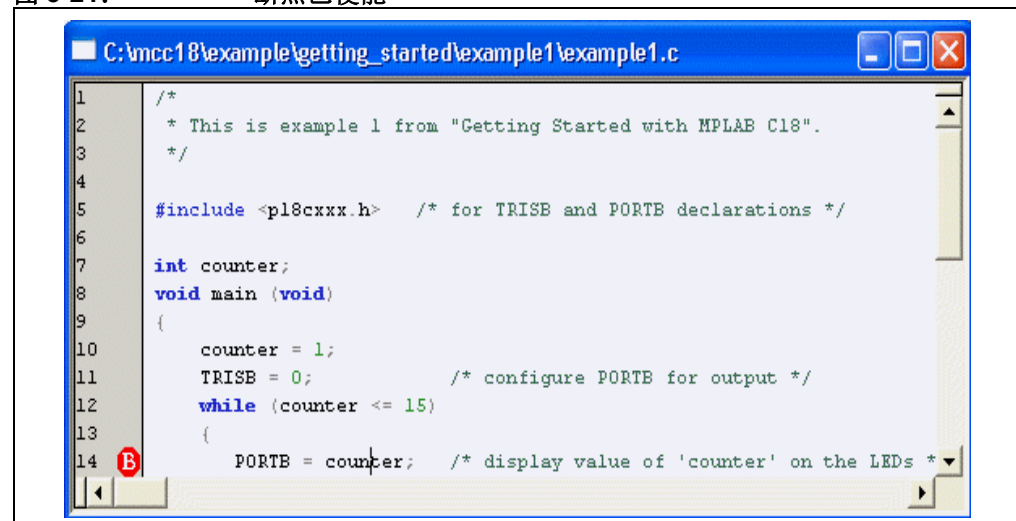
在项目树中双击源文件打开它。在源文件中，把光标放到想要设置断点的代码行，并单击鼠标右键，选择“Set Breakpoint”。

图 3-20: SET BREAKPOINT 选项



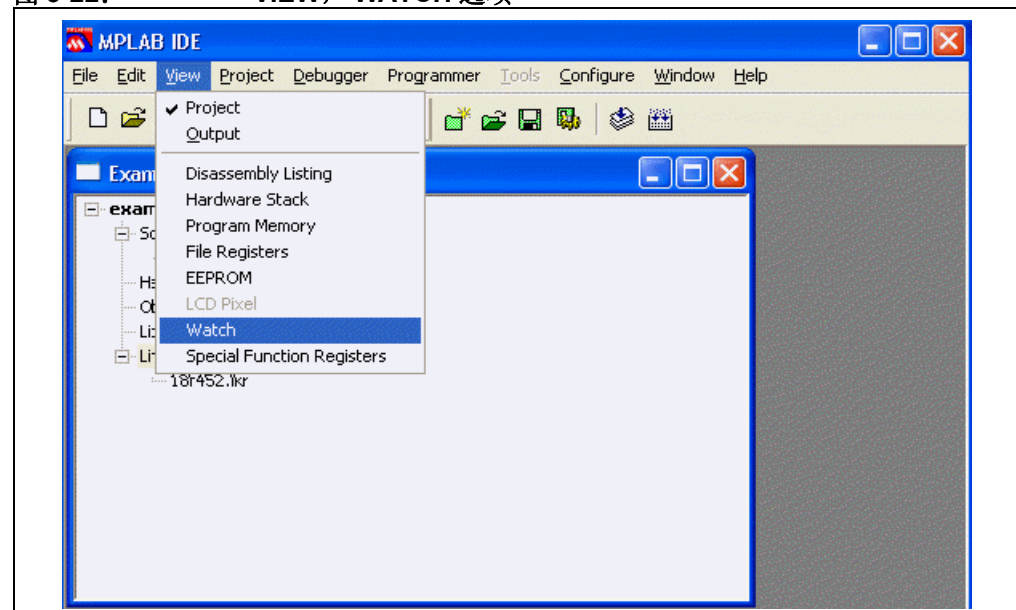
在源代码窗口左侧装订线处有一个红点，表示已设置并使能断点。

图 3-21: 断点已使能



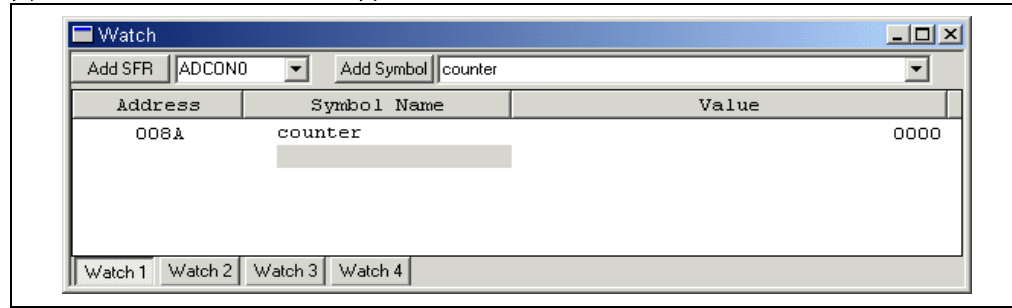
要打开 watch 窗口查看变量 counter，选择 View>Watch。

图 3-22: VIEW, WATCH 选项



从 **Add Symbol** 右侧的下拉菜单中选择 **counter**，并点击 **Add Symbol**。

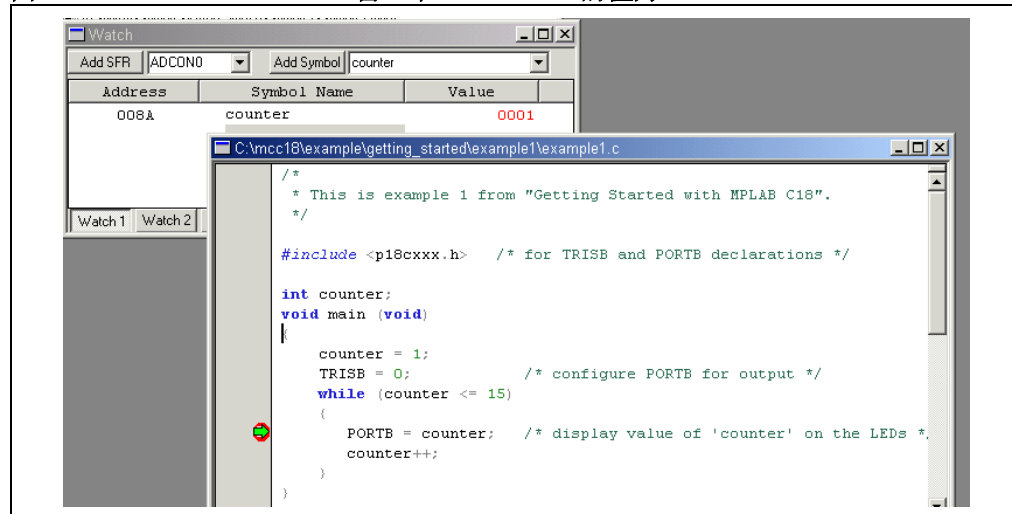
图 3-23: WATCH 窗口



点击工具栏中的 **Run**  运行程序。

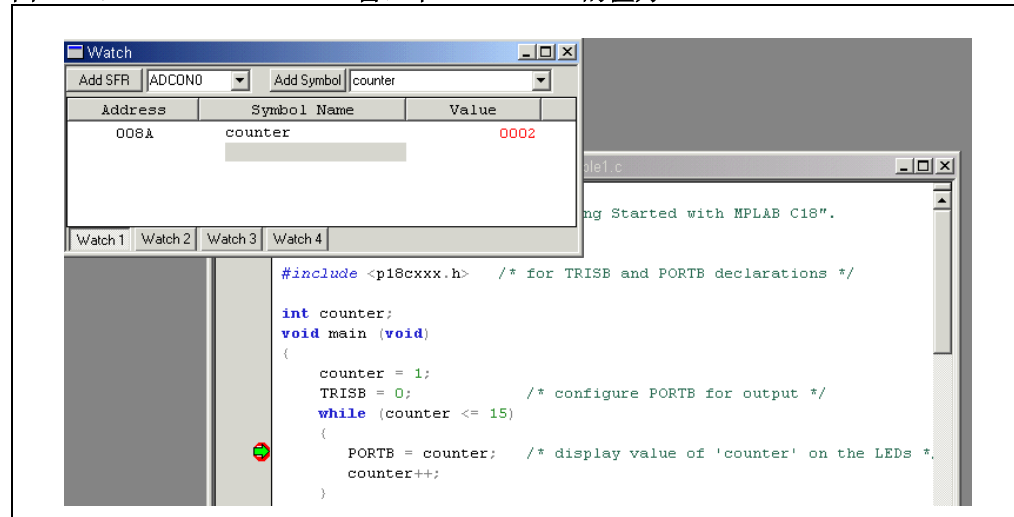
断点前一条语句执行完后，程序将会暂停。源代码窗口左侧装订线处的绿色箭头指向将要执行的下一条语句。watch 窗口中此时应显示 **counter** 的值为 1。

图 3-24: WATCH 窗口中 COUNTER 的值为 1



再次点击 **Run** 继续运行程序。程序再次暂停在断点处，此时 **watch** 窗口中应该显示 **counter** 的值为 2。

图 3-25: WATCH 窗口中 COUNTER 的值为 2



要单步执行源代码，即一次执行一条语句，可以使用工具栏中的 **Step Into**。是随着每条语句的执行，源代码窗口左侧装订线处的绿色箭头会指向要执行的下一条语句。

如果程序正在运行，可以点击工具栏中的 **Halt** 暂停程序的运行。



3.3.8 映射文件和列表文件

映射文件（`example1.map`）和列表文件（`example1.lst`）位于项目目录中，选择 **File>Open**，然后浏览至项目目录即可打开它们。这些文件提供了有助于调试的其它信息，例如变量分配的详细信息以及机器代码与源代码之间的对应关系。例如，映射文件显示了 `counter` 变量已被分配到数据存储器中的地址 `0x80`，且在 `example1.c` 中定义为非静态的全局变量，这样此变量就具备外部链接的功能（即此变量对于其它模块来说是可见的）。

例 3-1：映射文件

Symbols - Sorted by Address			
Name	Address	Location	Storage File
counter	0x00008a	data	extern c:\mcc18\getting_started\example1\example1.c

列表文件给出了为主函数的每一条语句所生成的机器代码。对于每一条指令，列出了其地址、机器代码和汇编指令。

例 3-2：列表文件

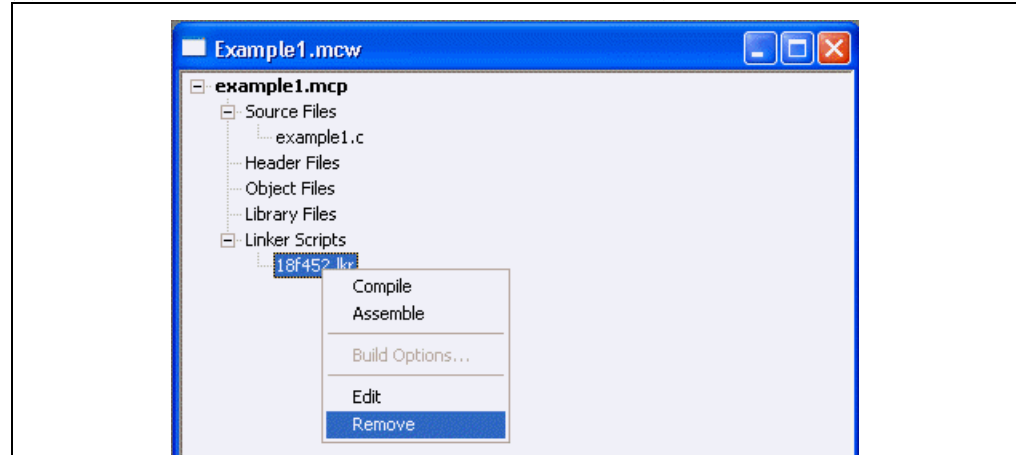
Address	Value	Disassembly	Source

			<code>#include <p18cxxx.h> /* for TRISB and PORTB declarations */</code>
			<code>int counter;</code>
			<code>void main (void)</code>
			<code>{</code>
			<code> counter = 1;</code>
0000e2	0e01	MOVLW 0x1	
0000e4	0100	MOVLB 0x0	
0000e6	6f8a	MOVWF 0x8a,0x1	
0000e8	6b8b	CLRF 0x8b,0x1	
0000ea	6a93	CLRF 0x93,0x0	<code> TRISB = 0; /* configure PORTB for output */</code>
0000ec	518b	MOVF 0x8b,0x0,0x1	<code> while (counter <= 15)</code>
0000ee	0a00	XORLW 0x0	
0000f0	ae88	BTFSS 0xe8,0x7,0x0	
0000f2	d002	BRA 0xf8	
0000f4	358b	RLCF 0x8b,0x0,0x1	
0000f6	d005	BRA 0x102	
0000f8	0e0f	MOVLW 0xf	
0000fa	80d8	BSF 0xd8,0x0,0x0	
0000fc	558a	SUBFWB 0x8a,0x0,0x1	
0000fe	0e00	MOVLW 0x0	
000100	558b	SUBFWB 0x8b,0x0,0x1	
000102	e306	BNC 0x110	
00010e	d7ee	BRA 0xec	
000104	c08a	MOVFF 0x8a,0xf81	<code> { PORTB = counter; /* display 'counter' on the LEDs */</code>
000106	ff81		
000108	2b8a	INCF 0x8a,0x1,0x1	<code> counter++;</code>
00010a	0e00	MOVLW 0x0	
00010c	238b	ADDWFC 0x8b,0x1,0x1	
000110	0012	RETURN 0x0	<code> }</code>

3.3.9 使用 MPLAB ICD 2 进行调试

可以使用 MPLAB ICD 2 来烧写器件和调试应用程序。为此，必须使用为 MPLAB ICD 2 设计的链接器描述文件重新编译项目。在项目窗口中，用鼠标右键单击“Linker Scripts”下面的文件 18f452.lkr，并点击 **Remove**。

图 3-26: LINKER SCRIPTS, REMOVE 选项

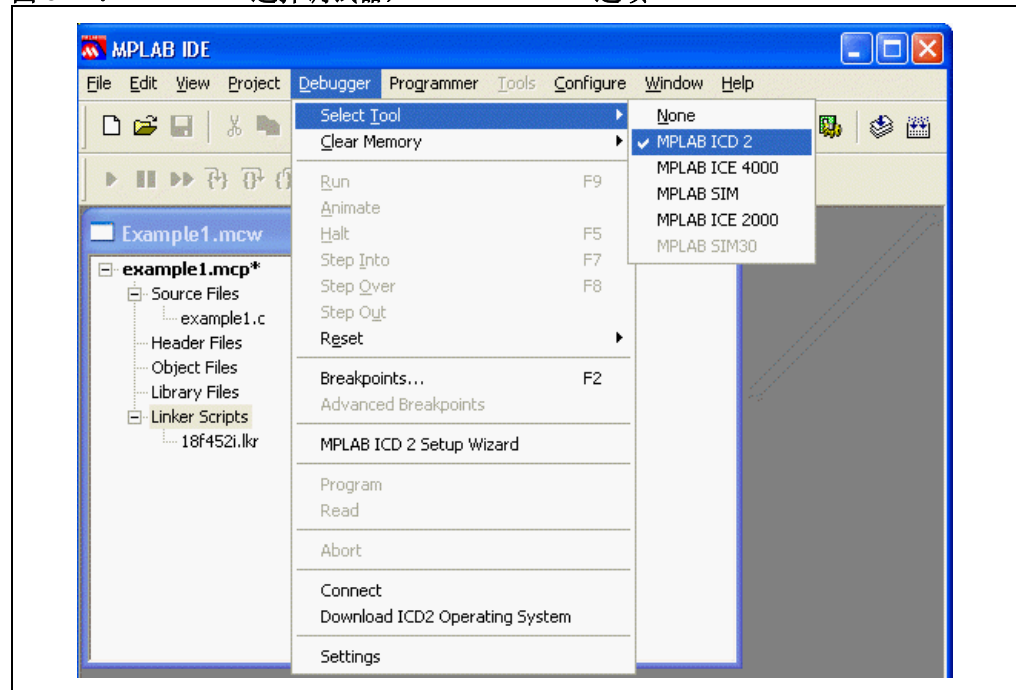


把 MPLAB C18 安装目录下 lkr 子目录中的链接器描述文件 18f452i.lkr，添加到项目树中的“Linker Scripts”下。此链接器描述文件会为 MPLAB ICD 2 使用的资源分配存储空间。文件名中的“i”表示此链接器描述文件用于 MPLAB ICD 2。

选择 **Project>Build All** 重新编译项目。

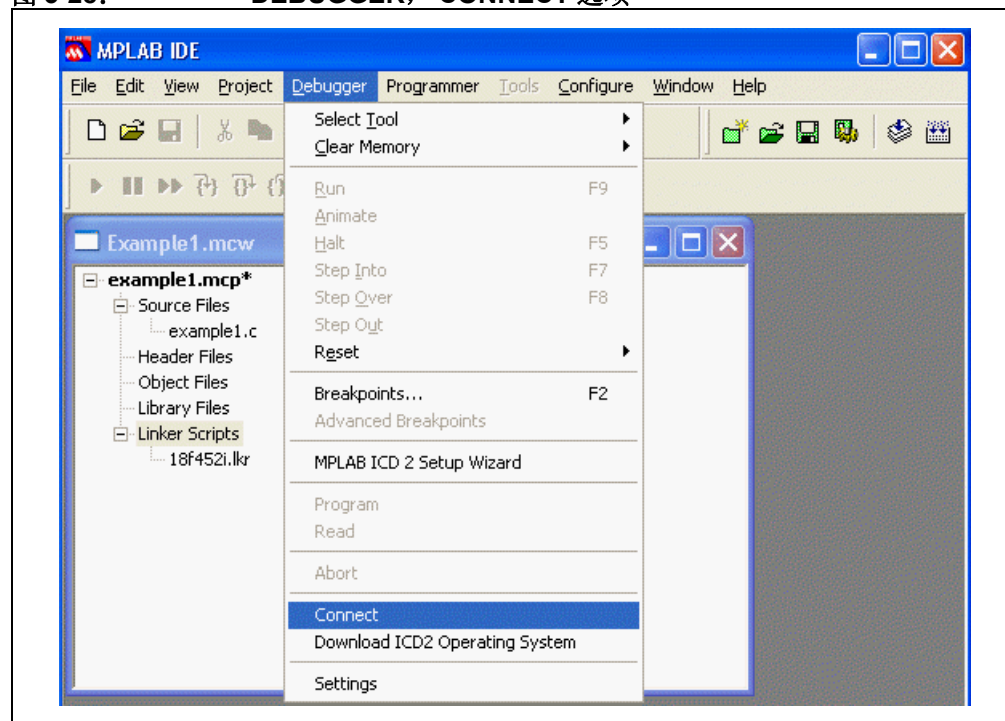
要使用 MPLAB ICD 2，选择 **Debugger>Select Tool>MPLAB ICD 2**。

图 3-27: 选择调试器，MPLAB ICD 2 选项



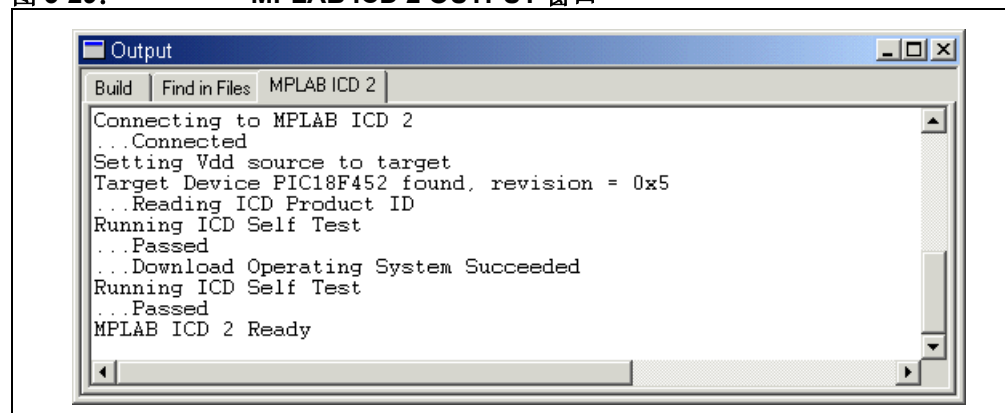
要与 MPLAB ICD 2 建立连接，选择 **Debugger>Connect**。有关如何配置 MPLAB ICD 2 连接设置的信息，请参阅 MPLAB ICD 2 文档。

图 3-28: DEBUGGER, CONNECT 选项



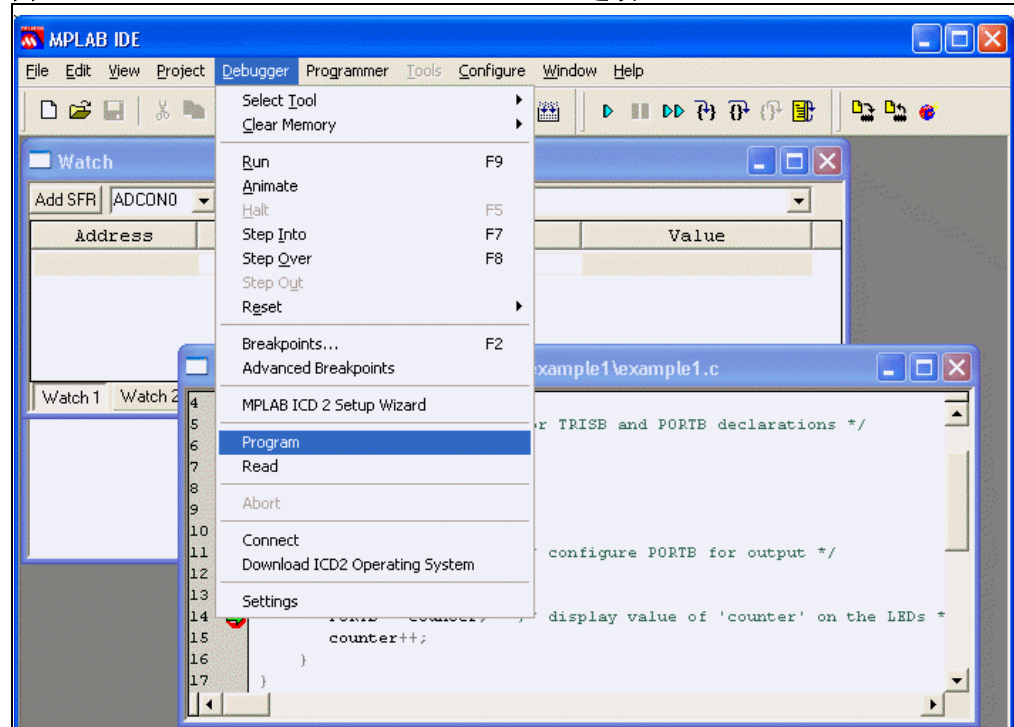
output 窗口应该显示 MPLAB ICD 2 已通过自检，并可以开始烧写器件。如果出现错误，请参阅 MPLAB ICD 2 相关文档。

图 3-29: MPLAB ICD 2 OUTPUT 窗口



要烧写器件，选择 **Debugger>Program**。

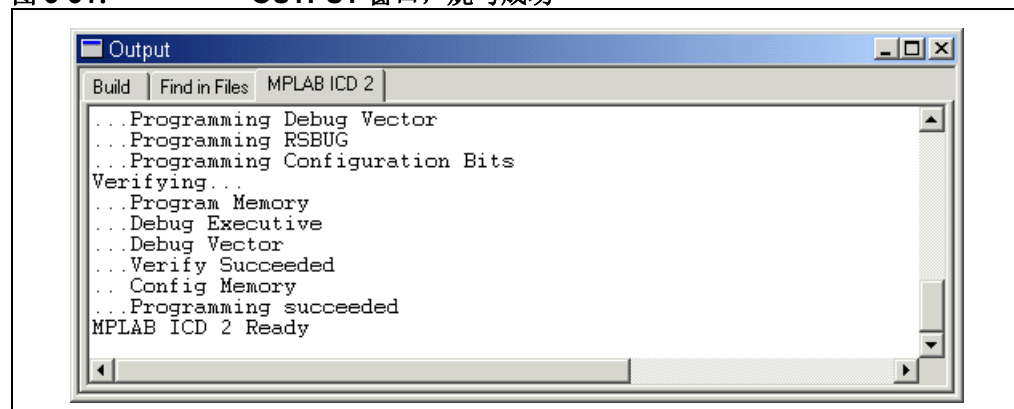
图 3-30: **DEBUGGER, PROGRAM 选项**



注： 考虑到 MPLAB ICD 2 所需要占用的资源，烧写器件时应减少程序存储器空间占用，同时应该禁止低电压编程，禁止看门狗定时器。如果有与这些问题相关的 MPLAB ICD 2 警告对话框出现，只需点击 **OK** 继续操作即可。

output 窗口应该显示烧写操作成功。

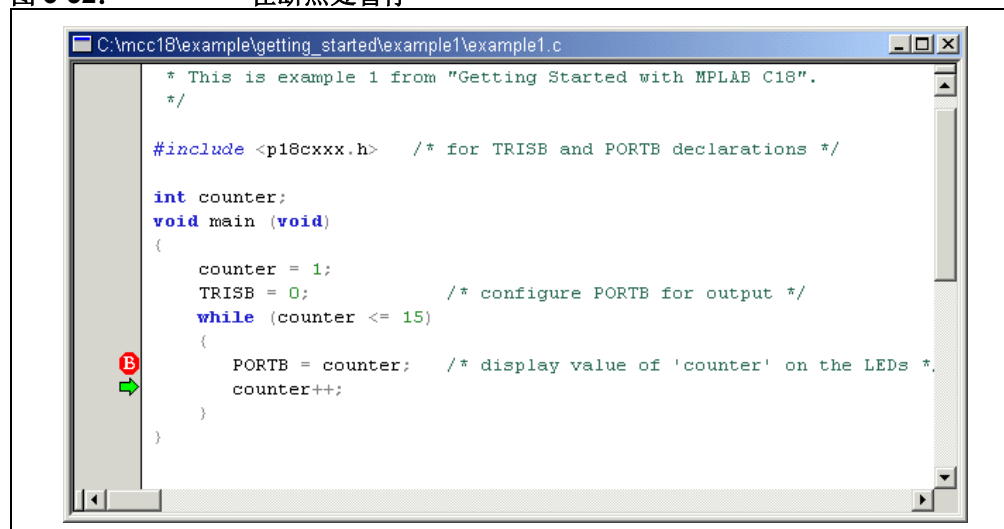
图 3-31: **OUTPUT 窗口，烧写成功**



可以按照前面讲述 MPLAB SIM 软件模拟器时提供的方法在源文件中设置断点。点击工具栏上的 **Run** 按钮，程序执行完断点处语句后会立即暂停。

注： 这与 MPLAB SIM 软件模拟器有所不同，在 MPLAB SIM 软件模拟器中程序会在执行断点处语句前暂停运行。绿色箭头指向下一条要执行的语句。

图 3-32: 在断点处暂停



PORTB 寄存器已经赋了值 1。PICDEM 2 Plus 演示板上的各发光二极管与 PORTB 的各引脚相连，这些发光二极管此时应该显示出以二进制形式表示的值 1。

注： 为了把 PORTB 的各引脚连接到各发光二极管，演示板上的跳线 J6 必须短接。

每次点击工具栏上的 **Run** 时，程序在对 PORTB 赋值后暂停执行，而发光二极管每次都会显示出 counter 递增后的值。

3.4 示例 2

本示例是为使用 MPLAB IDE v6.xx、MPLAB ICD 2、PICDEM 2 Plus 演示板和 PIC18F452 器件而设计的。它说明了如何使用 MPLAB C18 外设函数库和 C 标准函数库，以及变量在程序存储器中的分配。此程序完成一系列字符串的循环，其中每个字符串代表一个 0 到 15 的数字。为了能在发光二极管上显示出来，每个字符串都被转换成整型表示。每显示完一个数字，程序会暂停一会，以方便用户观察发光二极管。为运行此程序，PICDEM 2 Plus 演示板上的跳线 J6 必须短接。

- MPLAB C18 将字符串存放在程序存储器中；因此，在数组 `string_table` 的定义中要有关键字 `rom`。如果仅有关键字 `const`，是数据不能将存放到程序存储器的，故还需要加上关键字 `rom`。若没有额外的专用代码，程序存储器一般是不能被修改的，用关键字 `const` 也是比较恰当的。
- 配置位需要正确设置；这是通过调用宏 `_CONFIG_DECL` 设置每个配置字节实现的。这包括指定 PICDEM 2 Plus 演示板上使用的振荡器：本示例中使用晶振（`OSC_HS_1H`）。此外，MPLAB ICD 2 要求禁止看门狗定时器和低电压编程（分别为 `_WDT_OFF_2H` 和 `_LVP_OFF_4L`）。最后，由于本例使用 MPLAB ICD 2，因此应始终使能后台调试（`_DEBUG_ON_4L`）。设置配置位的宏在特定处理器的头文件中定义；具体细节请参阅《MPLAB® C18 C 编译器用户指南》（DS51288C_CN）。
- 将字符串转换为整型表示的标准 C 库函数 `atoi`，需要一个位于数据存储器中的字符指针。然而，由于字符串位于程序存储器中，故必须先复制到数据存储器。函数 `strcpypgm2ram` 可实现这一功能，此函数是 `strcpy` 在 MPLAB C18 中的变体。
- `PORTB` 寄存器（与 PICDEM 2 演示板上的发光二极管相连接）和 `TRISB` 寄存器（将 `PORTB` 各引脚配置为输入或输出）都是在特定处理器头文件 `p18f452.h` 中定义的。
- MPLAB C18 中的几个函数能提供不同时间长度的延时，例如下面要使用的 `Delay10RTCYx`。具体细节请参阅头文件 `delays.h`。

可以按照示例 1 中的步骤，在 MPLAB IDE v6.xx 中编译和链接此示例，然后使用 MPLAB ICD 2 进行调试。

注： 使用小代码存储模型进行编译时，MPLAB C18 可能会在调用 `strcpypgm2ram` 时发出赋值中类型限定符不匹配的警告。这是因为传递给函数的第二个实参是一个指向 `near` 程序存储器的指针，而函数原型中的参数类型是一个 `far` 程序存储器指针。由于 `near` 指针转换为 `far` 指针总是安全的，所以可以忽略此警告。或者使用大代码存储模型进行编译以消除警告（代价是可能产生更大的代码存储映像）。为此，从“Project”菜单中选择“Build Options”，然后选择“Project”。然后选择“MPLAB C18”并从“Categories”下拉菜单中选择“Memory Model”，最后选择“Large code model”。

图 3-33: MEMORY MODEL 选项

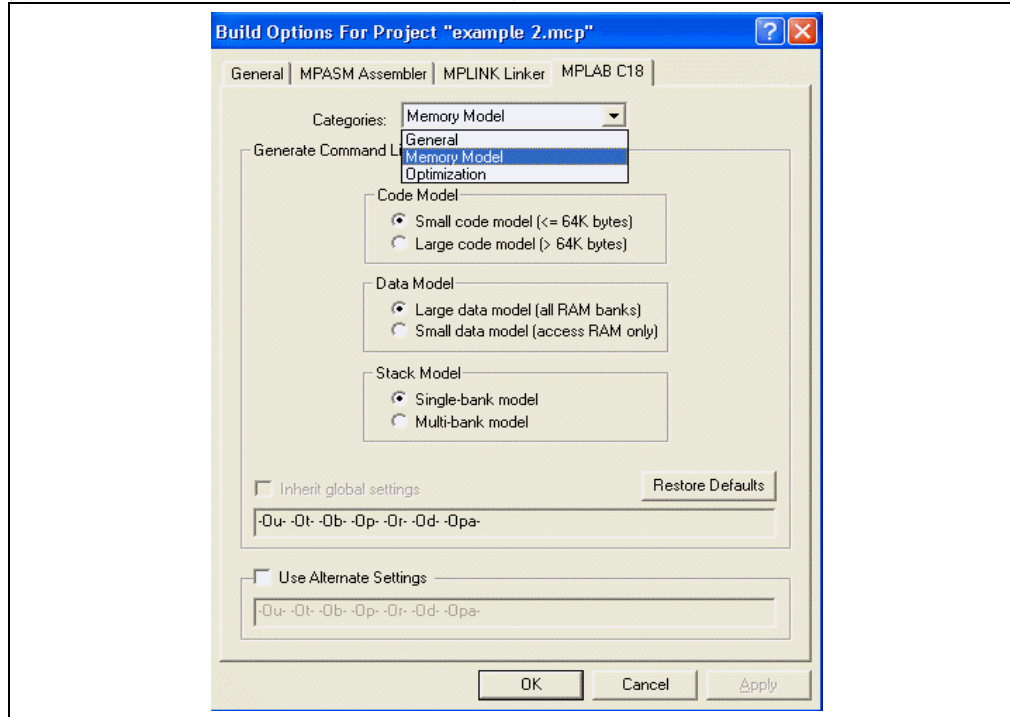
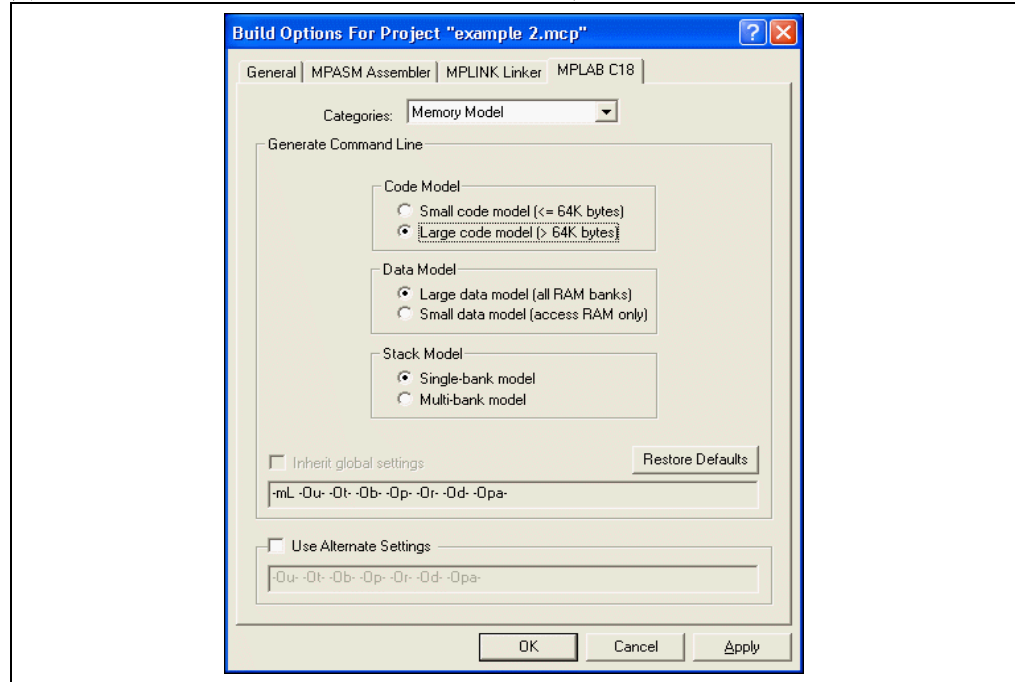


图 3-34: LARGE CODE MODEL 选项



```
#include <string.h>    /* for 'strcpypgm2ram'    */
#include <stdlib.h>    /* for 'atoi'          */
#include <delays.h>    /* for 'Delay10KTCYx'  */
#include <p18cxxx.h>   /* for 'PORTB' and 'TRISB' */

/* MPLAB C18 places string literals in program memory */
#define STRING_TABLE_SIZE 16
const rom char *string_table[STRING_TABLE_SIZE] =
{ "0", "1", "2", "3", "4", "5", "6", "7",
  "8", "9", "10", "11", "12", "13", "14", "15"
};

/* Set configuration bits for use with ICD2 / PICDEM2 PLUS Demo Board:
 * - set HS oscillator
 * - disable watchdog timer
 * - disable low voltage programming
 * - enable background debugging
 */
#pragma romdata CONFIG
_CONFIG_DECL(_CONFIG1H_DEFAULT & _OSC_HS_1H,
             _CONFIG2L_DEFAULT,
             _CONFIG2H_DEFAULT & _WDT_OFF_2H,
             _CONFIG3H_DEFAULT,
             _CONFIG4L_DEFAULT & _LVP_OFF_4L & _DEBUG_ON_4L,
             _CONFIG5L_DEFAULT,
             _CONFIG5H_DEFAULT,
             _CONFIG6L_DEFAULT,
             _CONFIG6H_DEFAULT,
             _CONFIG7L_DEFAULT,
             _CONFIG7H_DEFAULT);
#pragma romdata

void main (void)
```

```
{
    int index;
    int integer;
    char string[3];

    PORTB = 0;
    TRISB = 0;    /* configure all PORTB pins for output */

    for (index = 0; index < STRING_TABLE_SIZE; index++)
    {
        /* copy the string from program memory to data memory for 'atoi' */
        strcpypgm2ram (string, string_table[index]);
        /* get the number's integer representation from its
           string representation */
        integer = atoi (string);

        PORTB = integer;    /* output the value to the LEDs */
        Delay10KTCYx (255); /* pause for a moment (255 * 10,000 cycles) */
    }
}
```

3.5 示例 3

MPLAB C18 从版本 2.30 开始支持 PIC18 器件的扩展模式，旨在减少可重入代码的代码量。然而，并不是所有 PIC18 器件都支持扩展模式。请参阅器件数据手册，以获得详细信息和其它建议。

本示例是为使用 MPLAB IDE v6.xx、MPLAB ICD 2、PICDEM 2 Plus 演示板和 PIC18F4620 器件而设计的。本示例以例 2 为基础，并论述了扩展模式和非扩展模式之间的一些差别。

3.5.1 选择处理器

本示例使用 PIC18F4620 器件作为目标处理器。选择目标处理器的步骤请参见示例 1。

3.5.2 使用扩展模式

默认情况下，MPLAB C18 工作在非扩展模式，其所生成代码可在工作于非扩展模式的器件上运行。默认状态可通过命令行选项 `--extended` 更改。为进行更改，选择 **Project>Build Options>Project** 并点击 **MPLAB C18** 选项卡。然后，选中 **Use Alternate Settings** 复选框来使能此复选框下面的文本域。输入空格和 `--extended`。点击 **OK**。

图 3-35: 编译项目选项

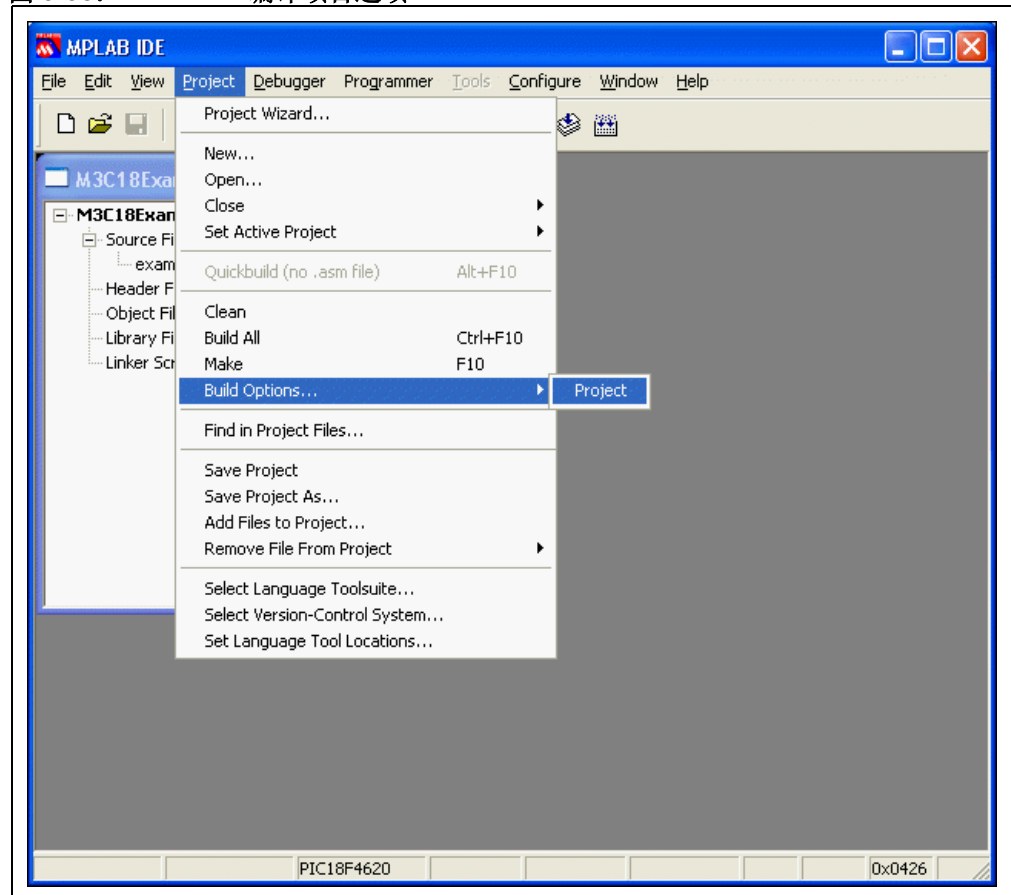
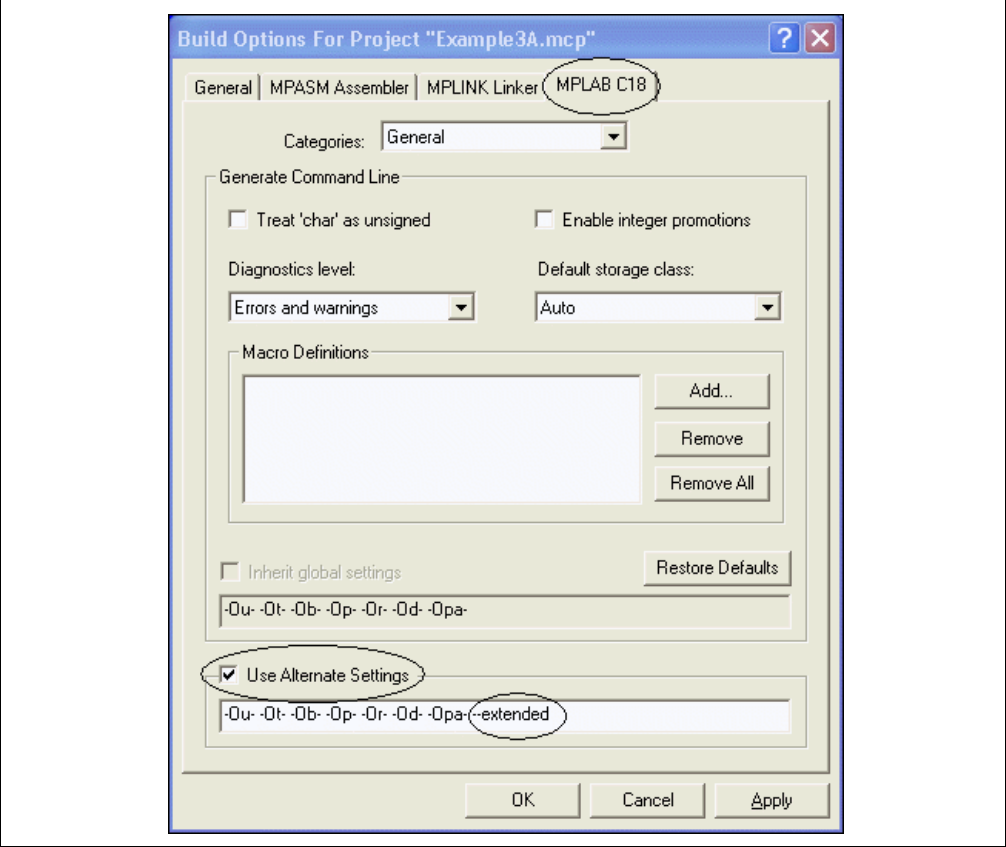


图 3-36: 扩展模式的可选设置



MPLAB C18 C 编译器为每种单片机提供了多达四种类型的链接器描述文件。这些链接器描述文件与 MPLAB IDE 所提供链接器描述文件的不同之处在于，它们在编译器启动代码和库中自动链接，并留出堆栈区。为 PIC18F4620 单片机提供的四个链接器描述文件如下表所示：

18f4620.lkr	供在非扩展模式下编译的应用程序使用。
18f4620i.lkr	供在非扩展模式下编译并使用 MPLAB ICD 2 调试的应用程序使用。“i”表示此链接器描述文件为 MPLAB ICD 2 使用的资源分配存储空间。
18f4620_e.lkr	供在扩展模式编译的应用程序使用。
18f4620i_e.lkr	供在扩展模式下编译并使用 MPLAB ICD 2 调试的应用程序使用。“i”表示此链接器描述文件为 MPLAB ICD 2 使用的资源分配存储空间。

对于本示例的这一部分，18f4620i_e.1kr 必须添加到 MPLAB IDE 项目的项目文件中。关于如何添加文件到项目的信息，参见 3.3 节“示例 1”。

接下来，使用 __EXTENDED18__ 预定义宏来设置配置字。用以下代码替换例 2 中的 CONFIG 段：

```
/* Configuration bit settings for use with ICD2 / PICDEM2 PLUS Demo Board:
 * - set the HS oscillator (for the crystal)
 * - disable watchdog timer and low-voltage programming (0 for disable)
 * - enable background debugging (0 for enable)
 * - for extended mode compilation, enable extended mode (1 for enable)
 */
#define CONFIG1H_DEFAULT 0x07 /* default value for configuration byte 1H */
#define HS_OSC      0x02      /* setting for HS oscillator */

#define CONFIG2H_DEFAULT 0x1F /* default value for configuration byte 2H */
#define WDT_BIT      0x01      /* enables/disables watchdog timer */

#define CONFIG4L_DEFAULT 0x85 /* default value for configuration byte 4L */
#define DEBUG_BIT 0x80      /* disables/enables background debugging */
#define XINST_BIT 0x40      /* enables/disables extended mode */
#define LVP_BIT 0x04      /* enables/disables low-voltage programming */

#pragma romdata CONFIG1H=0x300001
const rom unsigned char config1H = CONFIG1H_DEFAULT & HS_OSC;
#pragma romdata CONFIG2H=0x300003
const rom unsigned char config2H = CONFIG2H_DEFAULT & ~WDT_BIT;
#pragma romdata CONFIG4L=0x300006
#ifdef __EXTENDED18__
const rom unsigned char config4L =
    (CONFIG4L_DEFAULT | XINST_BIT) & ~LVP_BIT & ~DEBUG_BIT;
#else
const rom unsigned char config4L =
    CONFIG4L_DEFAULT & ~LVP_BIT & ~DEBUG_BIT;
#endif
#pragma romdata
```

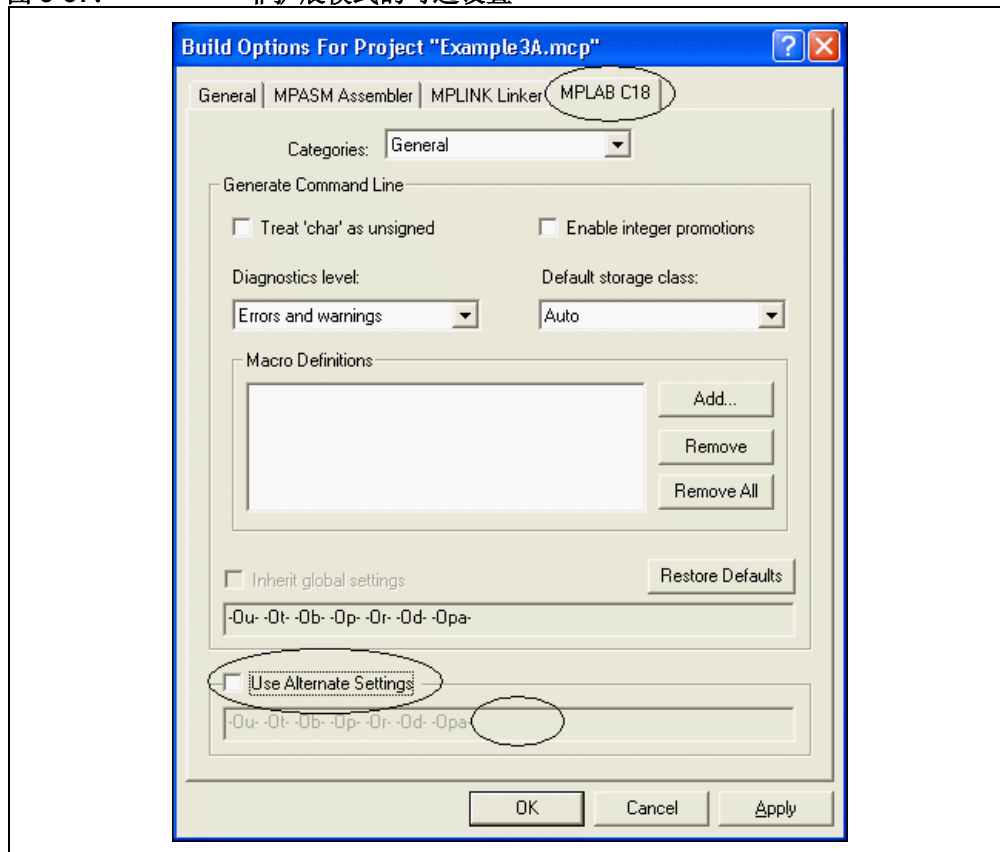
在扩展模式下编译时，以上代码将使能扩展模式指令；在非扩展模式下编译时，以上代码将禁止扩展模式指令。

3.5.3 使用非扩展模式

要将上例从扩展模式更改为非扩展模式，必须遵循下述步骤：

1. 禁止扩展模式。选择 **Project>Build Options>Project** 并点击 **MPLAB C18** 选项卡。然后，如果选中了 **Use Alternate Settings** 复选框，则从可选设置文本框中删除 **--extended**，并通过点击复选框来去掉对它的选择。点击 **OK**。

图 3-37: 非扩展模式的可选设置



2. 从 MPLAB IDE 项目中去除扩展模式的 MPLAB ICD 2 链接器描述文件 (18f4620i_e.lkr)。
3. 将非扩展模式的 MPLAB ICD 2 链接器描述文件 (18f4620i.lkr) 添加到 MPLAB IDE 项目。

3.6 示例 4

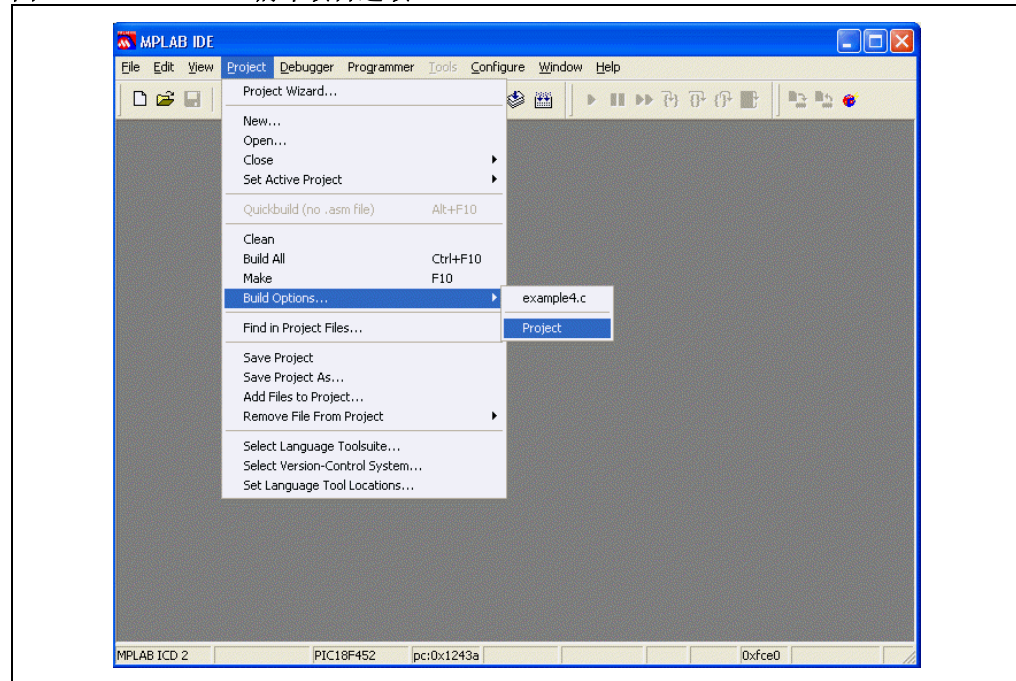
本示例是为使用 MPLAB IDE v6.xx、MPLAB SIM 软件模拟器和 PIC18F452 器件而设计的。在此示例中，介绍了变量在存取 RAM 中的分配。此程序能算出从 0 到 99 中每个值平方根的整数部分，然后再对每个平方根进行平方，以获得小于或等于原值的最大完全平方数。

- 平方根函数是以表的形式实现的，存储在程序存储器中。这样做有几个优点：假如表存储在数据存储器中，则在程序初始化时就需要把表从程序存储器复制到数据存储器；把表分配到程序存储器中还可以节省数据存储空间；此外，当函数定义域较小时，与计算平方根相关的代码在运行时可能会比表占据更多的程序存储空间。
- 位于存取 RAM 中的数据不需要加载 BSR 寄存器，因此所需的指令较少。变量 `root` 和 `square` 被分配到名为 `MY_ACS_DATA` 的存取 RAM 段中。必须使用类型限定符 `near`，以确保 MPLAB C18 知道不需为这些对象选择存储区。

本示例可以按照示例 1 中的步骤，在 MPLAB IDE v6.xx 中编译和链接，并通过 MPLAB SIM 软件模拟器来调试。

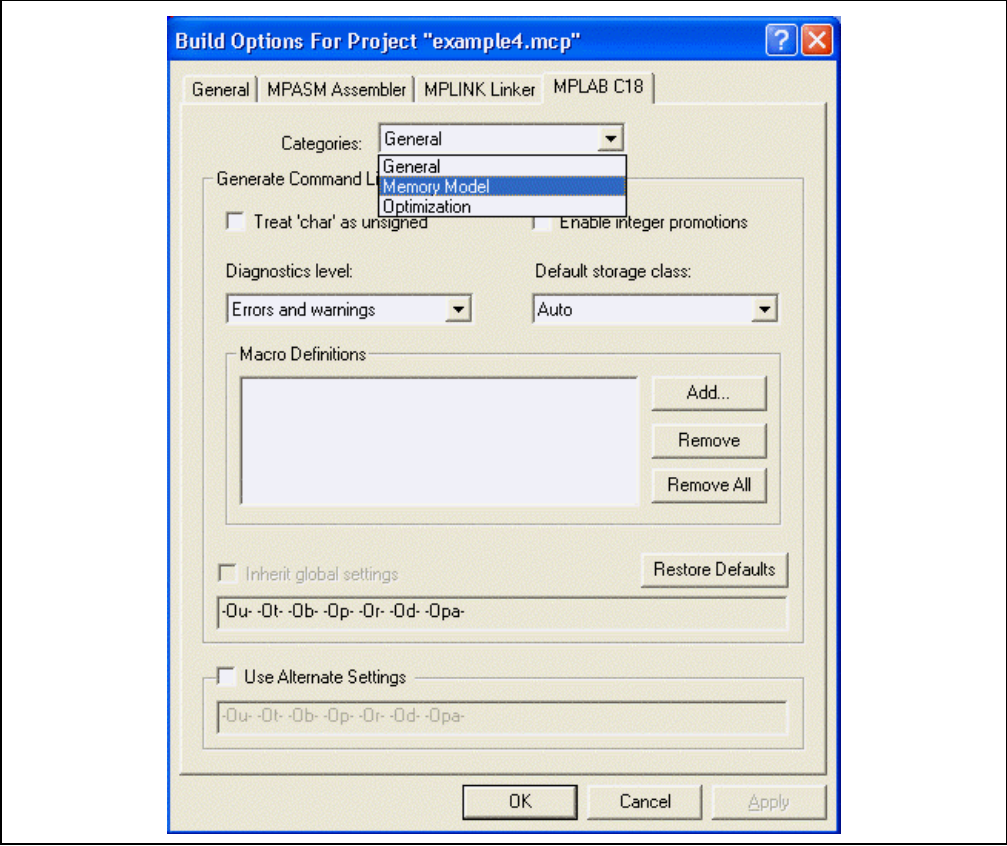
除非用 `near` 类型限定符进行明确定义，否则在默认状态下，MPLAB C18 假设所有静态分配的数据对象都位于分区的 RAM（非存取 RAM）中。默认状态可以通过命令行选项 `-0a+` 来改变。要改变默认设置，可选择 **Project>Build Options>Project**。

图 3-38: 编译项目选项



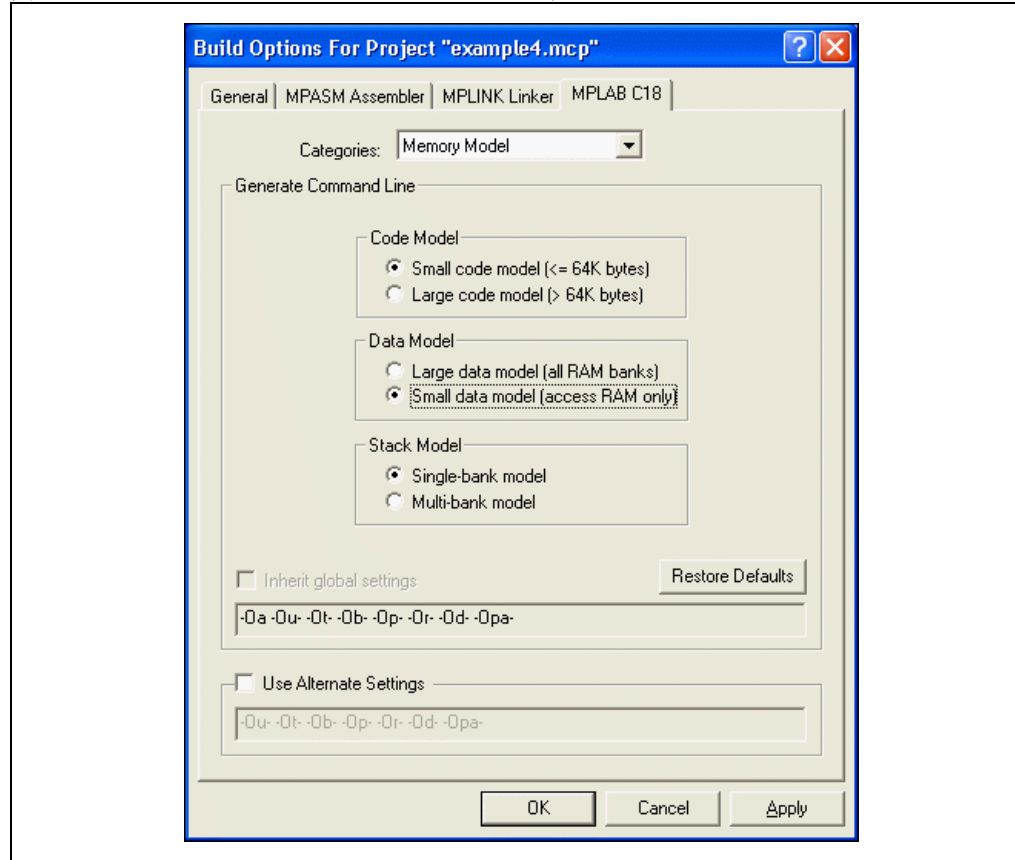
选择 “MPLAB C18” 选项卡并从下拉菜单中选择 “Memory Model”。

图 3-39: MEMORY MODEL 选项



最后，若没有明确的 `near` 或 `far` 限定符，选择 `small data model`（小数据模型）会告知 MPLAB C18，静态分配的数据对象都分配到存取 RAM 中。有关存取 RAM 和分区的（非存取）RAM 的详细信息，请参考《MPLAB® C18 C 编译器用户指南》（DS51288C_CN）。点击 **OK**。

图 3-40: SMALL DATA MODE 选项



```
#include <p18f452.h> /* for 'PRODL' declaration and 'ACCESS' macro */

/*
 * Locate the read-only table in program memory at address 0x1000.
 * 'romdata' is used for data, and 'code' is used for instructions.
 */
#pragma romdata ROOT_TABLE = 0x1000
#define TABLE_SIZE 100
const rom unsigned char roots[TABLE_SIZE] =
{ 0, 1, 1, 1, 2, 2, 2, 2, 3,
  3, 3, 3, 3, 3, 3, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 5, 5, 5, 5, 5,
  5, 5, 5, 5, 5, 5, 6, 6, 6, 6,
  6, 6, 6, 6, 6, 6, 6, 6, 6, 7,
  7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
  7, 7, 7, 7, 8, 8, 8, 8, 8, 8,
  8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
  8, 9, 9, 9, 9, 9, 9, 9, 9, 9,
  9, 9, 9, 9, 9, 9, 9, 9, 9, 9 };

/*
 * Data in access ram does not require banking.
 * When compiled with '-Oa', these pragmas and the 'near' qualifier
 * may be removed.
 */
#pragma udata access MY_ACS_DATA
near unsigned char root, square;
#pragma udata /* continue allocating static data in non-access ram */

/*
 * Returns the truncated root of the value.
 */
unsigned char get_root (int val)
{
    return roots[val];
}

void main (void)
{
    static int val;

    for (val = 0; val < TABLE_SIZE; val++)
    {
        /* 'square' holds the greatest perfect square less than or
           equal to 'val' */
        square = get_root (val);
    }
}
```

注： 当编译该示例时，若默认状态是静态数据在被分配到存取 RAM 中，则与 `square` 和 `root` 声明有关的 `udata pragma` 伪指令和 `near` 类型限定符都可以删除。

3.7 示例 5

本示例是为使用 MPLAB ICD 2、PICDEM 2 Plus 演示板、MPLAB IDE v6.xx 和 PIC18F452 器件而设计的。它示范了如何在 MPLAB C18 中使用中断服务程序，以及如何使用 MPLAB C18 外设函数库。对于这个程序，为了断开 PORTB 引脚和发光二极管间的连接，演示板上的 J6 跳线必须断开。

- 用此程序控制 PICDEM 2 Plus 演示板上的压电式蜂鸣器。用户可以按 S3 按钮来关掉蜂鸣器，再按一下又可以重新激活蜂鸣器。
- S3 按钮与 INT0 引脚相连，此引脚与 INT0 外部中断有关。此中断是一个高优先级中断，所以触发此中断总会跳转到程序存储器的地址 0x8 处。这是中断服务程序 (high_ISR) 的入口地址，此中断服务程序会转到关掉或激活蜂鸣器的函数。
- 配置位需要正确设置；这是通过调用宏 _CONFIG_DECL 设置每个配置字节实现的。这包括指定 PICDEM 2 Plus 演示板上使用的振荡器；本示例中使用晶振 (OSC_HS_1H)。此外，MPLAB ICD 2 要求禁止看门狗定时器和低电压编程（分别为 _WDT_OFF_2H 和 _LVP_OFF_4L）。最后，由于本例使用 MPLAB ICD 2，因此应始终使能后台调试 (_DEBUG_ON_4L)。设置配置位的宏在特定处理器的头文件中定义；具体细节请参阅《MPLAB® C18 C 编译器用户指南》(DS51288C_CN)。
- toggle_buzzer 声明为高优先级中断服务程序。这意味着在进入和退出中断时，WREG、BSR 和 STATUS 寄存器的内容可通过自身影子寄存器来保存和恢复，而无需使用保存和恢复寄存器指令。此外，在中断返回时，INTCON 寄存器中的 GIEH 位会被置位，而在触发中断时此位又会被清零。有关中断逻辑的详细信息，请参考《PIC18FXX2 数据手册》(DS39564B_CN)。

本示例可按照示例 1 的步骤在 MPLAB IDE v6.xx 中编译和链接，并通过 MPLAB ICD 2 来调试。

```
#include <p18f452.h> /* for the special function register declarations */
#include <portb.h> /* for the RB0/INT0 interrupt */

/* Set configuration bits for use with ICD2 / PICDEM2 PLUS Demo Board:
 * - set HS oscillator
 * - disable watchdog timer
 * - disable low voltage programming
 * - enable background debugging
 */
#pragma romdata CONFIG
_CONFIG_DECL(_CONFIG1H_DEFAULT & _OSC_HS_1H,
             _CONFIG2L_DEFAULT,
             _CONFIG2H_DEFAULT & _WDT_OFF_2H,
             _CONFIG3H_DEFAULT,
             _CONFIG4L_DEFAULT & _LVP_OFF_4L & _DEBUG_ON_4L,
             _CONFIG5L_DEFAULT,
             _CONFIG5H_DEFAULT,
             _CONFIG6L_DEFAULT,
             _CONFIG6H_DEFAULT,
             _CONFIG7L_DEFAULT,
             _CONFIG7H_DEFAULT);
#pragma romdata
```

```
/*
 * For high interrupts, control is transferred to address 0x8.
 */
void toggle_buzzer (void); /* prototype needed for 'goto' below */
#pragma code HIGH_INTERRUPT_VECTOR = 0x8
void high_ISR (void)
{
    _asm
        goto toggle_buzzer
    _endasm
}
#pragma code /* allow the linker to locate the remaining code */

/*
 * If the buzzer is on, turn it off.  If it is off, turn it on.
 */
#pragma interrupt toggle_buzzer
void toggle_buzzer (void)
{
    CCP1CON = ~CCP1CON & 0x0F;    /* turn the buzzer off or on */
    INTCONbits.INT0IF = 0;        /* clear flag to avoid another interrupt */
}

void EnableHighInterrupts (void)
{
    RCONbits.IPEN = 1;    /* enable interrupt priority levels */
    INTCONbits.GIEH = 1; /* enable all high priority interrupts */
}

void InitializeBuzzer (void)
{
    T2CON = 0x05;    /* postscale 1:1, Timer2 ON, prescaler 4 */
    TRISCbits.TRISC2 = 0; /* configure the CCP1 module for the buzzer */
    PR2 = 0x80;    /* initialize the PWM period */
    CCP1L = 0x80;    /* initialize the PWM duty cycle */
}

void SoundBuzzer (void)
{
    CCP1CON = 0x0F;    /* turn the buzzer on */
    while (1);    /* wait for the S3 button to be pressed */
}

void main (void)
{
    EnableHighInterrupts ( );
    InitializeBuzzer ( );

    OpenRB0INT (PORTB_CHANGE_INT_ON & /* enable the RB0/INT0 interrupt */
               PORTB_PULLUPS_ON &    /* configure the RB0 pin for input */
               FALLING_EDGE_INT);    /* trigger interrupt upon S3 button
                                     depression */

    SoundBuzzer ( );
}
```

术语表

A

ANSI

美国国家标准学会

B

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，右侧第二位表示 8 的倍数，右侧第三位表示 $8^2 = 64$ 的倍数，以此类推。

编译器 (Compiler)

将用高级语言编写的源文件翻译成机器代码的程序。

C

CPU

中央处理单元

存储类别 (Storage Class)

确定与指定对象相关联存储区的生存时间。

存储模型 (Memory Model)

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)

表明所定义的对象特性（例如 `CONST`）。

存取存储区 (Access Memory)

PIC18 PICMICRO 单片机的一些特殊通用寄存器，对这些寄存器的访问与存储区选择寄存器（BSR）的设置无关。

错误文件 (Error File)

包含 MPLAB C18 所生成的诊断的文件。

D

单片机 (Microcontroller)

高度集成的芯片，它包括 CPU、RAM、某种类型的 ROM、I/O 端口和定时器。

递归函数 (Recursive)

自调用的函数（即调用自己的函数）。参见递归 (*RECURSIVE*)。

地址 (Address)

确定信息在存储器中位置的代码。

段 (Section)

位于特定存储器地址的一段应用程序。

段属性 (Section Attribute)

段的特性 (如 ACCESS 段)。

E

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

Free-standing

一种实现, 它接受任何不使用复杂数据类型的严格符合程序, 而且在这种实现中, 对库条款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>。

非扩展模式 (Non-extended Mode)

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。

G

高级语言 (High-level Language)

编写程序的语言, 与汇编语言相比, 它不依赖于具体的处理器。

H

汇编器 (Assembler)

把汇编源代码翻译成机器代码的语言工具。

汇编语言 (Assembly)

以可读形式描述二进制机器代码的符号语言。

I

ICD

在线调试器

ICE

在线仿真器

IDE

集成开发环境

IEEE

电子和电气工程师协会

ISO

国际标准化组织

ISR

中断服务程序

J**绝对段 (Absolute Section)**

具有链接器不能改变的固定地址的段。

K**可重定位 (Relocatable)**

没有被指定到固定的存储器地址的对象。

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。

库 (Library)

可重定位目标模块的集合。

库管理器 (Librarian)

建立并管理库的程序。

扩展模式 (Extended Mode)

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**链接器 (Linker)**

把目标文件和库文件结合起来生成可执行代码的程序。

M**MPASM 汇编器 (MPASM Assembler)**

MICROCHIP PICMICRO 系列单片机的可重定位宏汇编器。

MPLIB 目标库管理器 (MPLIB Object Librarian)

MICROCHIP PICMICRO 系列单片机的库管理器。

MPLINK 目标链接器 (MPLINK Object Linker)

MICROCHIP PICMICRO 系列单片机的链接器。

目标代码 (Object Code)

由汇编器或编译器生成的机器代码。

目标文件 (Object File)

包含目标代码的文件。它可以直接执行或需要与其它目标代码文件（比如库文件）链接，以生成完全可执行的程序。

N

匿名结构 (Anonymous Structure)

未命名的对象。

P

Pragma

一种伪指令，它对于特定的编译器有意义。

R

RAM

随机访问存储器

ROM

只读存储器

S

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 表示十进制数 10 到 15。最右边的位表示 1 的倍数，右边第二位表示 16 的倍数，第三位表示 $16^2 = 256$ 的倍数，以此类推。

随机访问存储器 (Random Access Memory)

一种存储器，可以在这种存储器中以任意顺序读写信息。

T

特殊功能寄存器 (Special Function Register)

控制 I/O 处理函数、I/O 状态、定时器，或其它模式及外设的寄存器。

条件编译 (Conditional Compilation)

只有当预处理伪指令指定的某个固定表达式为真时才编译程序段的操作。

X

向量 (Vector)

复位或中断发生时，应用程序跳转到的存储器地址。

小尾数法 (Little Endian)

将给定对象的最低有效字节存储在较低的地址。

Y**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。

运行时模型 (Runtime Model)

编译器运行所遵循的各项前提。

Z**帧指针 (Frame Pointer)**

指向堆栈中位置的指针，它用于区分堆栈中的函数参数和局部变量。

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，但不允许添加或更改数据。

致命错误 (Fatal Error)

引起编译立即停止的错误。不产生其它消息。

中断 (Interrupt)

发送到 CPU 的信号，它使 CPU 暂停正在运行的应用程序，把控制权转交给中断服务程序，以处理事件。执行完中断服务程序后，继续正常执行应用程序。

中断服务程序 (Interrupt Service Routine)

处理中断的函数。

中断响应时间 (Latency)

从事件发生到得到响应的的时间。

中央处理单元 (Central Processing Unit)

器件的一部分，其功能是取出要执行的指令，再对指令进行译码，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。

字节存储顺序 (endianness)

多字节对象的字节存储顺序。

注:

索引

符号

__EXTENDED18__ 45

A

安装 MPLAB C18 11

B

把文件添加到项目中 27

编译项目 29

编译选项 26

C

COD 9

参考读物 3

存储模型 48

存取 RAM 1, 19, 47

D

电子文档 14

断点 30, 32, 38

F

非扩展模式 1, 9, 19, 44, 46

G

观察窗口 31–33

H

HEX 9

函数库 3, 8, 14

K

客户通知服务 5

客户支持 6

扩展模式 1, 9, 19, 43, 45

L

链接器描述文件 8, 14, 28, 35

列表文件 34

路径 25

M

MCC_INCLUDE 16

Microchip 网站 4

MPLAB ICD 2 35

MPLAB SIM 软件模拟器 30

目录 12

目录内容 8

R

README 文件 3

Readme 文件 12

S

示例 8, 14, 20, 39, 43, 47, 51

W

文档约定 2

问题解答 4

X

系统要求 7

项目路径 25

项目设置 22

项目树 21

小尾数法 56

卸载 MPLAB C18 17

新项目 21

选择器件 21

选择语言工具套件 22

Y

映射文件 34

语言工具 9, 22

运行时模型 19

Z

中断服务程序 1, 19, 51, 57

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://support.microchip.com>
网址: www.microchip.com

亚特兰大 **Atlanta**

Alpharetta, GA
Tel: 1-770-640-0034
Fax: 1-770-640-0307

波士顿 **Boston**

Westford, MA
Tel: 1-978-692-3848
Fax: 1-978-692-3821

芝加哥 **Chicago**

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

达拉斯 **Dallas**

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

底特律 **Detroit**

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

科科莫 **Kokomo**

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

洛杉矶 **Los Angeles**

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

圣何塞 **San Jose**

Mountain View, CA
Tel: 1-650-215-1444
Fax: 1-650-961-0286

加拿大多伦多 **Toronto**

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

亚太地区

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

中国 - 成都
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

中国 - 福州
Tel: 86-591-750-3506
Fax: 86-591-750-3521

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

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

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

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

中国 - 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

中国 - 青岛
Tel: 86-532-502-7355
Fax: 86-532-502-7205

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

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

印度 **India - Bangalore**
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

印度 **India - New Delhi**
Tel: 91-11-5160-8632
Fax: 91-11-5160-8632

日本 **Japan - Kanagawa**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 **Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

新加坡 **Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

欧洲

奥地利 **Austria - Weis**
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

丹麦 **Denmark - Ballerup**
Tel: 45-4420-9895
Fax: 45-4420-9910

法国 **France - Massy**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Ismaning**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 **Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 **Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

英国 **England - Berkshire**
Tel: 44-118-921-5869
Fax: 44-118-921-5820