

## 开发工具客户须知

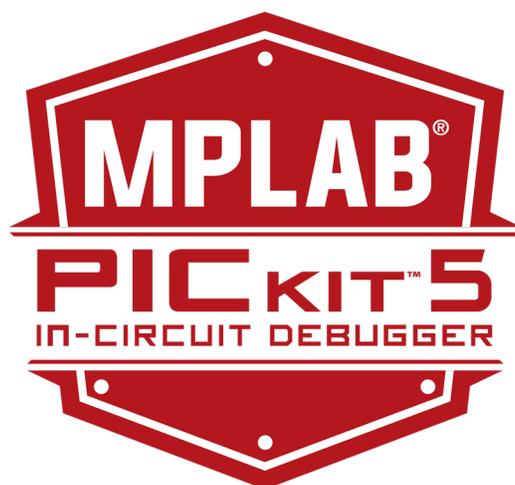


### 重要:

开发工具手册如同所有其他文档一样具有时效性。我们不断改进工具和文档以满足客户的需求，因此实际使用中有些对话框和/或工具说明可能与本文档所述之内容有所不同。请访问我们的网站 ([www.microchip.com/](http://www.microchip.com/)) 获取最新版本的 PDF 文档。

文档每页的底部均标有 DS 编号。DS 格式为 DS<文档编号><版本>\_CN，其中<文档编号>为 8 位数字，<版本>为大写字母。

有关最新信息，请访问 [onlinedocs.microchip.com/](http://onlinedocs.microchip.com/) 查看您所使用的工具的帮助信息。



# 目录

|  |    |
|--|----|
| 开发工具客户须知.....  | 1  |
| 1. 前言.....   | 4  |
| 1.1. 本指南使用的约定.....   | 4  |
| 1.2. 推荐读物.....   | 4  |
| 2. 关于调试器.....  | 6  |
| 2.1. 调试器的优势.....   | 6  |
| 2.2. 调试器组件.....  | 7  |
| 2.3. 调试器框图.....  | 9  |
| 2.4. MPLAB PICKit 5 在线调试器与 MPLAB X IDE 和 MPLAB IPE 配合使用..... | 9  |
| 3. 连接.....   | 11 |
| 3.1. 电源连接.....   | 11 |
| 3.2. PC 与智能手机连接.....   | 13 |
| 3.3. 目标板连接.....  | 14 |
| 4. 工作原理.....   | 27 |
| 4.1. 调试/编程快速参考.....  | 27 |
| 4.2. 操作概述.....   | 27 |
| 4.3. 关于高电压.....  | 28 |
| 4.4. SAM 和 PIC32C Arm <sup>®</sup> 器件——片上调试.....             | 29 |
| 4.5. AVR 器件——片上调试 (OCD) .....                                | 29 |
| 4.6. PIC32M MCU——片上调试.....                                   | 36 |
| 4.7. PIC MCU/dsPIC DSC——片上调试.....                            | 36 |
| 5. 脱机编程.....   | 43 |
| 5.1. 脱机编程的电源要求.....  | 44 |
| 5.2. 脱机编程的限制.....  | 44 |
| 5.3. 将 PICKit 5 设置为脱机编程模式.....                               | 45 |
| 5.4. 使用脱机编程.....   | 48 |
| 5.5. 退出脱机编程模式.....   | 49 |
| 5.6. 使用 BLE 设备上的 PTG 应用程序.....                               | 50 |
| 6. 疑难解答.....   | 54 |
| 6.1. 要首先回答的一些问题.....   | 54 |
| 6.2. 无法调试的首要原因.....  | 54 |
| 6.3. 通用.....   | 54 |
| 6.4. 如何使用硬件工具紧急引导固件恢复实用程序.....                               | 55 |
| 7. 常见问题解答.....   | 57 |
| 7.1. 工作原理.....   | 57 |
| 7.2. 出现的问题.....  | 57 |
| 8. 错误消息.....   | 59 |

|                                 |    |
|---------------------------------|----|
| 8.1. 错误消息类型.....                | 59 |
| 8.2. 常规纠正措施.....                | 66 |
| 9. 调试器功能汇总.....                 | 68 |
| 9.1. 调试器选择与切换.....              | 68 |
| 9.2. 调试器选项选择.....               | 68 |
| 10. 硬件规范.....                   | 73 |
| 10.1. USB 连接器规范.....            | 73 |
| 10.2. MPLAB PICKit 5 在线调试器..... | 73 |
| 10.3. 通信硬件.....                 | 75 |
| 10.4. 目标板注意事项.....              | 76 |
| 11. 版本历史.....                   | 78 |
| 12. 支持.....                     | 79 |
| 12.1. 保修登记.....                 | 79 |
| 12.2. myMicrochip 个性化通知服务.....  | 79 |
| 13. 术语表.....                    | 80 |
| Microchip 信息.....               | 96 |
| 商标.....                         | 96 |
| 法律声明.....                       | 96 |
| Microchip 器件代码保护功能.....         | 96 |

## 1. 前言

本章包含使用 MPLAB® PICKit™ 5 在线调试器前需要了解的一般信息。

### 1.1. 本指南使用的约定

本指南采用以下文档约定。在大多数情况下，格式符合 [OASIS Darwin Information Typing Architecture \(DITA\) Version 1.3 Part 3: All-Inclusive Edition](#) (2018 年 6 月 19 日)。

表 1-1. 文档约定

| 说明                                       | 实现                | 示例   |
|--|-------------------|--|
| 参考书目                                     | DITA: 引用          | <i>MPLAB® PICKit™ 5 在线调试器用户指南。</i>                                     |
| 需强调的文字                                   | 斜体字               | <i>...仅有的编译器...</i>  |
| 窗口、窗格或对话框名称                              | DITA: 窗口标题        | <b>Output</b> 窗口。<br><b>New Watch</b> 对话框。                             |
| 窗口或对话框中的字段名                              | DITA: 用户界面控件      | 选择 <b>Optimizations</b> 选项类别。  |
| 菜单名称或菜单项                                 | DITA: 用户界面控件      | 选择 <b>File</b> 菜单，然后选择 <b>Save</b> 。                                   |
| 菜单路径                                     | DITA: 级联菜单，用户界面控件 | <b>File &gt; Save</b>  |
| 选项卡                                      | DITA: 用户界面控件      | 单击 <b>Power</b> 选项卡。   |
| 软件按钮                                     | DITA: 用户界面控件      | 单击 <b>OK</b> 按钮。   |
| 键盘上的按键                                   | DITA: 用户界面控件      | 按下 <b>F1</b> 键。  |
| 文件名和路径                                   | DITA: 文件路径        | C:/Users/User1/Projects  |
| 源代码：嵌入                                   | DITA: 代码片段        | 切记要在代码的开头添加#define START。  |
| 源代码：代码块                                  | DITA: 代码块         | 示例如下：<br><pre>#include &lt;xc.h&gt; main(void) {     while(1); }</pre> |
| 用户输入的数据                                  | DITA: 用户输入        | 键入器件名称，例如 PIC18F47Q10。   |
| 关键字                                      | DITA: 代码片段        | static, auto, extern   |
| 命令行选项                                    | DITA: 代码片段        | -Opa+, -Opa-   |
| 二进制位值                                    | DITA: 代码片段        | 0, 1   |
| 常量                                       | DITA: 代码片段        | 0xFF, 'A'  |
| 可变参数                                     | DITA: 代码片段 + 选项   | file.o, 其中 file 可以为任何有效的文件名。   |
| 可选参数                                     | 方括号[]             | xc8 [options] files  |
| 选择互斥参数：“或”选择                             | 花括号和竖线: {   }     | errorlevel {0 1}   |
| 代替重复文字                                   | 省略号...            | var_name [, var_name...]   |
| 表示由用户提供的代码                               | 省略号...            | void main (void) { ... }   |
| verilog 格式的数字，其中 N 为总位数，R 为基数，n 为其中一位。   | N'Rnnnn           | 4'b0010, 2'hF1   |
| 器件相关标志。指定并非所有器件上均支持的特性。有关详细信息，请参见器件数据手册。 | [DD]              | 汇编器特殊操作符   |

### 1.2. 推荐读物

本用户指南介绍了如何使用 MPLAB PICKit 5 在线调试器。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。

《多工具设计忠告》(DS51764C\_CN)

**请先阅读本文档！** 该文档包含有关在使用 MPLAB PICKit 5 在线调试器进行目标设计时应考虑的操作问题的重要信息。

### **MPLAB X IDE 在线帮助**

这是在使用任何 **Microchip** 硬件工具时的必备文档。

这是 **MPLAB X IDE** 的全面帮助文件。它包括嵌入式系统的概述、安装要求、教程以及创建新项目、设置编译属性、调试代码、设置配置位、设置断点以及编程器件等方面的详细信息。此帮助文件通常比可以从 **MPLAB X IDE 网页** 免费下载的用户指南 (DS50002027) 的 PDF 内容更新。

### **MPLAB PICKit 5 的发行说明**

有关使用 MPLAB PICKit 5 在线调试器的最新信息，可访问 MPLAB X IDE 中 **Help > Release Notes** (帮助 > 发行说明) 下的发行说明。发行说明包含本用户指南中可能未提供的更新信息和已知问题。

### **MPLAB PICKit 5 In-Circuit Debugger Quick Start Guide (DS50003478)**

该参考材料介绍了如何为 MPLAB PICKit 5 在线调试器连接硬件和安装软件。

## 2. 关于调试器

MPLAB PICKit 5 在线调试器 (PG164150) 允许利用 MPLAB X IDE 集成开发环境或 MPLAB IPE (集成编程环境) 功能强大的图形用户界面快速、轻松地对 Microchip 器件进行调试和编程。支持的器件包括:

- PIC<sup>®</sup>和 AVR<sup>®</sup>单片机 (MCU)
- dsPIC<sup>®</sup>数字信号控制器 (Digital Signal Controller, DSC)
- SAM (基于 Arm<sup>®</sup> Cortex<sup>®</sup> 内核) MCU 和微处理器 (MPU)
- CEC (基于 Arm Cortex 内核) MCU

有关 PICKit 5 对特定器件的支持, 请参见[器件支持列表](#)。

MPLAB PICKit 5 通过 USB Type-C<sup>®</sup> 接口连接到设计工程师的计算机, 并可通过 Microchip 8 引脚单列直插式 (Single In-Line, SIL) 调试连接器连接到目标板。附加 microSDHC 卡插槽和通过目标板自供电的能力意味着您可以随身“携带”代码进行脱机编程。

除了更宽的目标电压外, MPLAB PICKit 5 还支持高级接口 (如 4 线 JTAG、串行线调试 (Serial Wire Debug, SWD) 和流数据网关), 同时向后兼容使用 2 线 JTAG 和 ICSP 的演示板和目标系统。由于添加了 microSDHC 卡插槽用于存放项目代码以及可通过目标板供电, MPLAB PICKit 5 还具有脱机编程 (Programmer-To-Go, PTG) 功能。此外, 还可以使用 MPLAB PTG 应用程序来选择和管理 microSDHC 卡上的代码, 以及将代码编程到目标器件。

由于该调试器系统使用带有内置仿真电路的器件 (而非特殊调试器芯片), 因此代码的执行与实际器件相似。通过该调试器系统可以交互方式访问给定器件的所有可用功能, 且可通过 MPLAB X IDE 界面设置和修改这些功能。

MPLAB PICKit 5 在线调试器兼容以下任意平台:

- Microsoft Windows<sup>®</sup> 操作系统
- Linux<sup>®</sup> 操作系统
- macOS<sup>®</sup>

有关支持的版本, 请参见发行说明。

MPLAB PICKit 5 在线调试器专为调试嵌入式处理器而开发, 具有丰富的调试功能, 嵌入式处理器在以下方面与传统系统处理器有所不同:

- 处理器以最大速度运行。
- 能够集成 I/O 端口数据输入。
- 高级主机通信接口 (Windows、macOS 和 Linux)。
- 高级通信介质和协议。

除了调试器功能外, MPLAB PICKit 5 在线调试器系统也可用作器件生产编程器。

### 2.1. 调试器的优势

MPLAB PICKit 5 在线调试器系统具有以下优势:

#### 特性/功能:

- 通过 USB Type-C 线缆连接到计算机。
- 通过 USB 线缆或目标器件供电, 并可选择为目标器件供电 (最高 150 mA)。
- 8 引脚 SIL 编程连接器, 可选择使用各种接口。

- 使用 MPLAB X IDE 或 MPLAB IPE 编程器件。
- 支持脱机编程（PTG）以现场编程器件。
- 支持 MPLAB PTG iOS/Android 应用程序，用于通过 *Bluetooth*<sup>®</sup> 无线技术选择和管理 PTG 程序映像。
- 支持虚拟通信端口（VCOM）。
- 支持多个硬件和软件断点、跑表以及源代码文件调试。
- 在您自己的硬件上实时调试应用程序。
- 在器件全速工作的情况下进行调试。
- 基于内部事件设置断点。
- 监视内部文件寄存器。
- 配置引脚驱动器。
- 通过在 MPLAB X IDE 或 MPLAB IPE 中新增/更新软件包支持新器件和新特性。
- 通过指示灯带指示调试器状态。
- 可在 0-70°C 的温度范围内工作。

### 性能/速度:

- 实时操作系统（Real-Time Operating System, RTOS）。
- 更换器件时不会发生固件下载延迟。
- 以 300 MHz 频率运行的 32 位 MCU。

### 安全性:

- 当目标板需要外部电源时，从调试器接收反馈。
- 对于低电压编程模式，支持的目标电源电压为 1.2V 到 5.0V；对于高电压编程模式，支持的目标电源电压为 1.8V 到 5.0V。
- 在探针驱动器中添加保护电路，以便免受来自目标板的电源浪涌的危害。
- $V_{DD}$  和  $V_{PP}$  电压监视器用于过压保护/所有线路均具备过流保护。
- 可设置编程/调试引脚的电阻的值的范围和方向（上拉、下拉或不存在）。
- 可控编程速度提供了灵活性，能够克服目标板设计问题。
- 符合 CE 和 RoHS 标准——符合行业标准。

## 2.2. 调试器组件

下图展示了 MPLAB PICKit 5 在线调试器系统的组件。

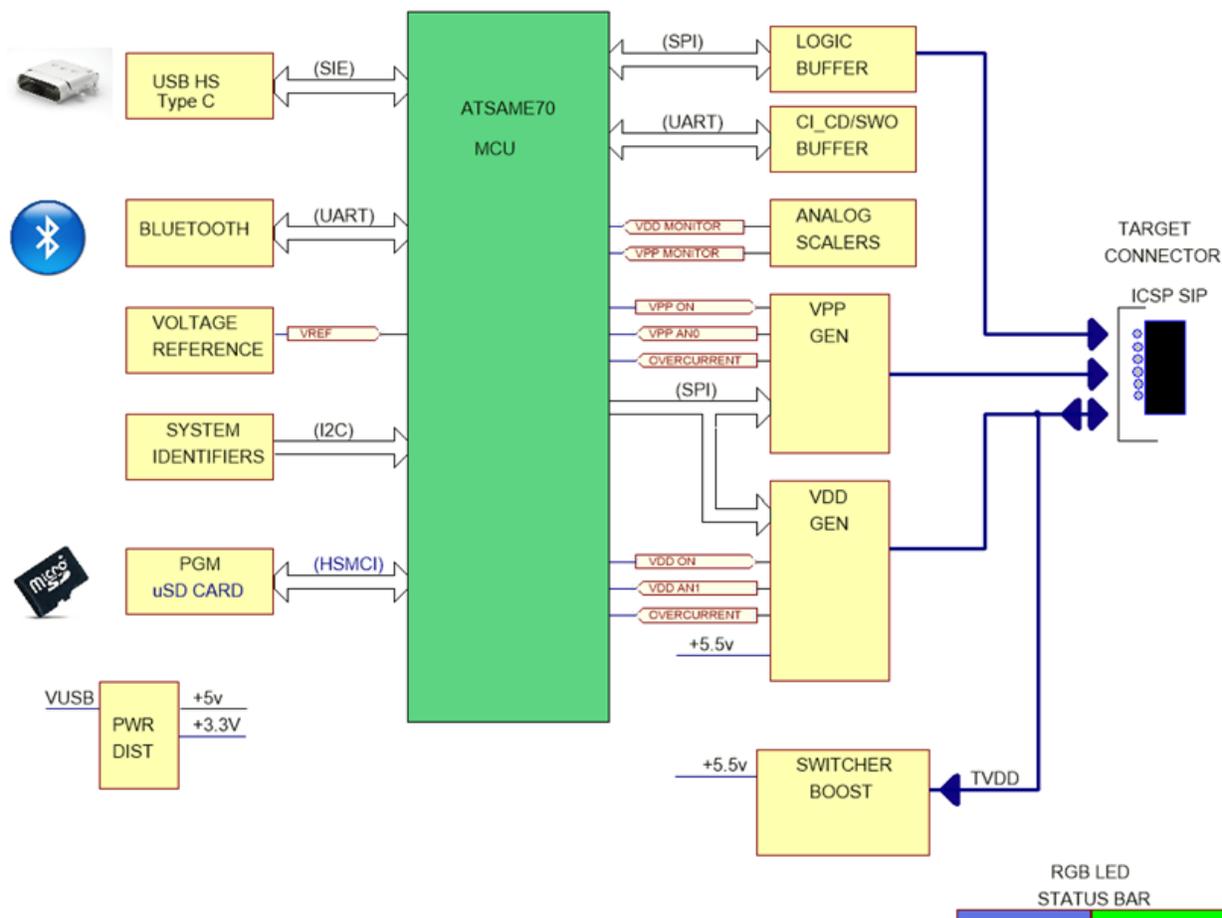


1. 随附的 USB Type-C 线缆，用于提供调试器与计算机之间的通信以及向调试器供电。
2. MPLAB PICKit 5 主机，其采用正面为拉丝金属的长方形耐用黑色塑料外壳，带有指示灯带（通电时亮起）。下面列出了主机特性的概述。有关详细信息，请参见 [MPLAB PICKit 5 在线调试器](#)。
  - USB Type-C 连接器（顶部）
  - 8 引脚单列直插式（SIL）连接器（底部）
  - MicroSD 卡插槽（左侧）
  - 紧急恢复按钮（顶部）
  - 挂绳连接器（顶部）
  - 脱机编程按钮（正面拉丝金属盾形徽标下方）
  - （内部）经认证的蓝牙 5.0 模块，用于 MPLAB PTG 应用程序与 MPLAB PICKit 5 之间的通信。
3. MPLAB PICKit 5 贴纸。

其他硬件和附件可以从 [Microchip 直销网站](#) 单独订购，例如：

- 调试器适配器板（部件编号 [AC002015](#)）——一款支持 JTAG、SWD 和 ICSP 协议的连接板，适用于使用 MPLAB PICKit 5 调试传统 AVR<sup>®</sup> 单片机。

## 2.3. 调试器框图



## 2.4. MPLAB PICKIT 5 在线调试器与 MPLAB X IDE 和 MPLAB IPE 配合使用

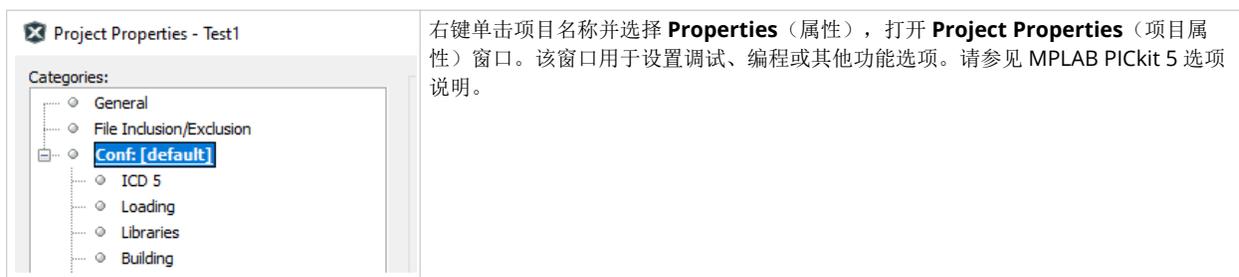
从 [MPLAB X IDE](#) 网页下载最新版本的 MPLAB<sup>®</sup> X IDE 并进行安装。MPLAB X IDE 安装程序将安装 MPLAB X IDE 和/或 MPLAB IPE。

### MPLAB<sup>®</sup> PICKIT<sup>™</sup> 5 与 MPLAB X IDE 配合使用

MPLAB PICKIT 5 在线调试器与 MPLAB X IDE 配合使用来开发目标应用程序。[MPLAB X IDE](#) 网页中提供 *MPLAB<sup>®</sup> X IDE User's Guide* (DS-50002027) 和其他文档。或者，也可参见 [MPLAB X IDE](#) 在线帮助。

表 2-1. MPLAB X IDE 概述

|   |                                       |
|---|---------------------------------------|
|  | 使用桌面图标启动 IDE。                         |
|  | 新建项目或打开现有项目。选择 MPLAB PICKIT 5 作为硬件工具。 |



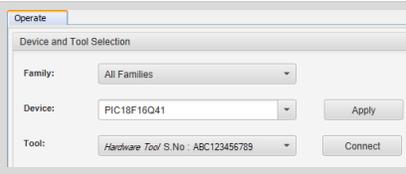
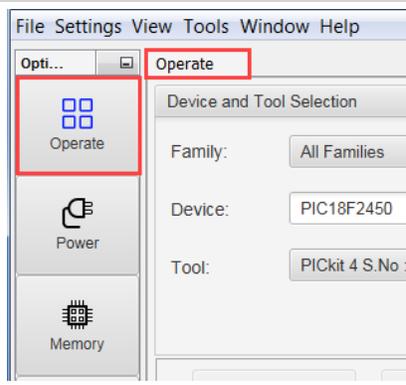
右键单击项目名称并选择 **Properties**（属性），打开 **Project Properties**（项目属性）窗口。该窗口用于设置调试、编程或其他功能选项。请参见 [MPLAB PICKit 5 选项说明](#)。

## MPLAB PICKit 5 与 MPLAB IPE 配合使用

MPLAB PICKit 5 在线调试器与 MPLAB IPE 配合使用来作为编程器。[MPLAB X IDE](#) 网页中提供 *MPLAB IPE User's Guide*（DS-50002227）用户指南和其他文档。或者，也可参见 [MPLAB IPE](#) 在线帮助。

此外，还提供命令行 IPE 工具选项。请查看 [MPLAB X IDE](#) 安装文件夹下的 docs 子文件夹。

表 2-2. MPLAB IPE 概述

|   |  |
|---|--|
|    | 使用桌面图标启动 IPE。  |
|   | 选择要编程的器件，然后选择 MPLAB PICKit 5 作为工具。   |
|  | 可选择单击 <b>Program</b> （编程）、 <b>Erase</b> （擦除）、 <b>Read</b> （读取）、 <b>Verify</b> （校验）或 <b>Blank Check</b> （空白检查）按钮。有关 <a href="#">MPLAB IPE</a> （包括高级模式）的更多信息，请参见 <i>MPLAB IPE User's Guide</i> 。 |

### 3. 连接

进行 MPLAB® PICKit™ 5 在线调试器的硬件设置时，首先为调试器连接电源、通信和目标板。有关详细信息，请参见以下章节。

#### 3.1. 电源连接

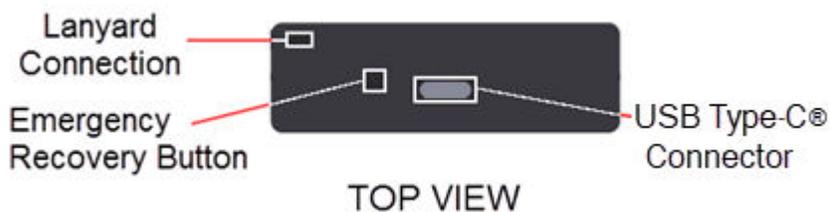
MPLAB PICKit 5 在线调试器通过其 USB Type-C® 连接器供电。另请参见 [USB 连接器规范](#)。

或者，MPLAB PICKit 5 也可通过 PTG 的目标板供电。请参见[脱机编程的电源要求](#)。

图 3-1. 调试器主机顶部的连接器

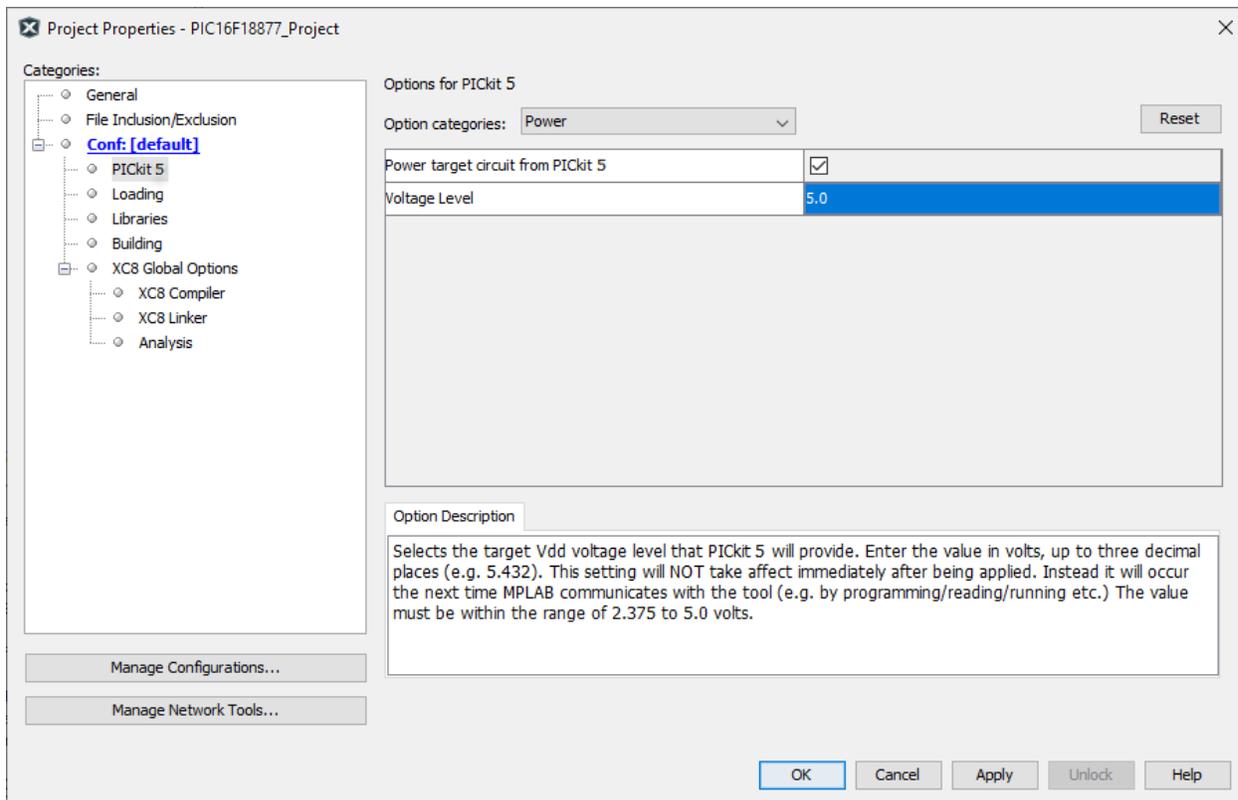


图 3-2. USB 电源和通信



目标板由自身电源供电。或者，也可由调试器为目标板供电，但需满足以下两个条件：(1)通过外部供电集线器连接调试器；(2)目标板的电流消耗不超过 150 mA。在 **Project Properties** 窗口中，选中 **Power target from PICKit 5**（通过 PICKit 5 为目标板供电）（见下图）。选择 Voltage Level（电压大小）时，不要超过器件 VDD 额定值。请参见[目标板由调试器供电](#)。

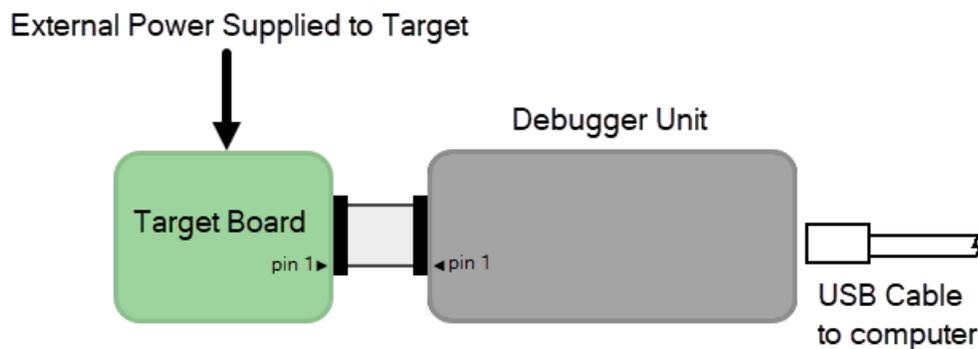
图 3-3. 选择由调试器为目标板供电



### 3.1.1. 目标板由外部供电

如果目标板由外部供电（见下图），调试器将检测目标  $V_{DD}$  以进行电压转换来实现目标低电压操作。如果调试器在其  $V_{DD}$  线（接口连接器的引脚 2）上未检测到电压，则它将无法工作。

图 3-4. 通过外部电源为目标板供电



注：在脱机编程操作期间，由外部电源供电的目标板可为调试器供电。

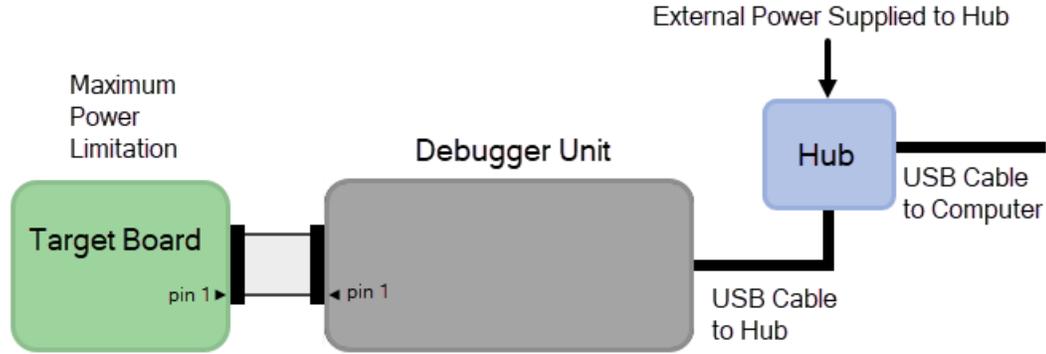
#### 相关信息

[目标板连接](#)

### 3.1.2. 目标板由调试器供电

如果目标板通过调试器供电，则必须按如下所示来使用外部供电集线器。目标板的最大可用电流限制为 150 mA。

图 3-5. 通过自供电集线器为目标板供电



为目标器件供电时，请确保目标器件不会暴露在高于器件 VDD 额定值的电压下。不得超过器件 VDD 的绝对最大值。器件在最大额定值条件下工作一段时间，其可靠性可能受到影响。我们建议不要使器件工作在超过器件数据手册规范标示的参数范围的条件下。有关所需的器件电压大小和最大额定值，请参见器件数据手册。

并非所有的器件都有  $AV_{DD}$  和  $AV_{SS}$  线，但是如果目标器件上有这两根线，为了调试器能够正常工作，必须将它们连接到适当的电平。它们不能处于悬空状态。

此外，应将具有  $V_{CAP}$  线的器件（例如 PIC18FXXJ）连接到适当的电容或电平。

#### 相关信息

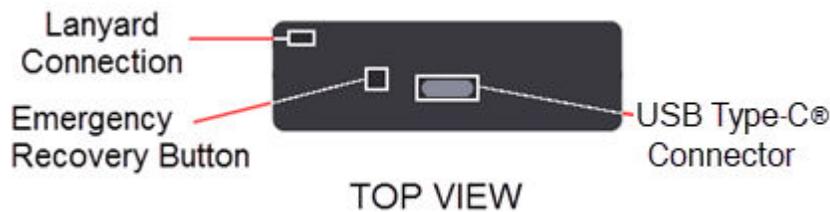
[目标板连接](#)

### 3.2. PC 与智能手机连接

MPLAB<sup>®</sup> PICKIT<sup>™</sup> 5 在线调试器可使用 USB Type-C 线缆（见下图）连接到 PC（以及 MPLAB X IDE 或 MPLAB IPE）。建议使用工具包随附的线缆，以避免发生通信问题。

此外，还可将 MPLAB PICKIT 5 连接到装有相关应用程序的 BLE Android/iOS 智能手机或平板电脑以支持 PTG。

图 3-6. USB 电源和通信



| 连接类型                      | 连接详细信息     | 编程 <sup>1</sup> | 调试 <sup>1</sup> | MPLAB <sup>®</sup> 数据可视化器 |
|---------------------------|------------|-----------------|-----------------|---------------------------|
| USB Type-C                | HS USB 2.0 | 支持              | 支持              | 支持                        |
| Bluetooth <sup>®</sup> 技术 | 无线         | 支持 <sup>2</sup> | 不支持             | 不支持                       |

1. 有关速度的信息，请参见[电路板规范](#)。  
2. 智能手机/平板电脑连接。

#### 相关信息

[使用 BLE 设备上的 PTG 应用程序](#)

### 3.3. 目标板连接

MPLAB<sup>®</sup> PICKit<sup>™</sup> 5 在线调试器通过 8 引脚单列直插式（SIL）连接器连接到目标板。确保将调试器上的引脚 1（▼）与目标板上的引脚 1 对齐（见下图）。对于其他目标板连接，我们有相应的适配器板可供选购。有关连接器和适配器板的引脚分配，请参见以下章节。

图 3-7. 调试器主机底部的连接器



图 3-8. 8 引脚单列直插式连接器

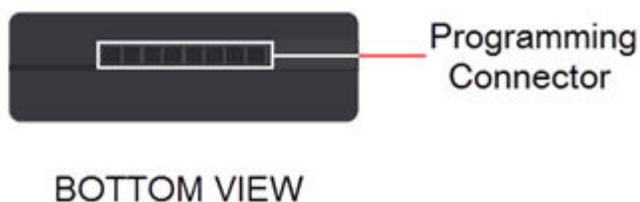
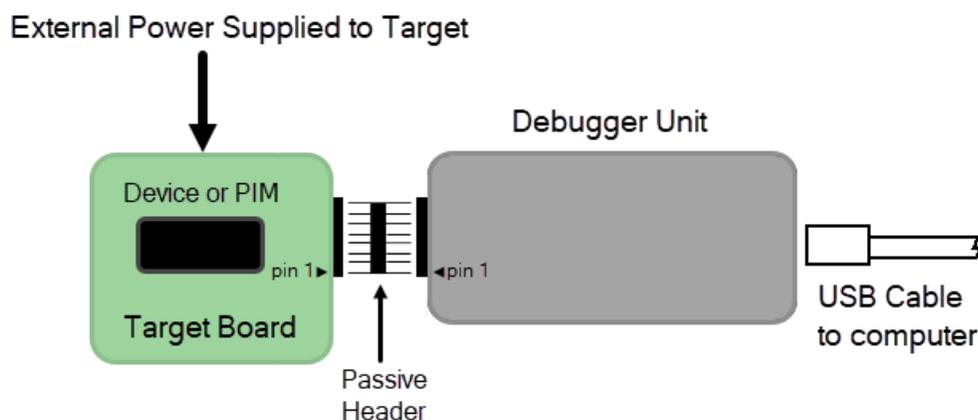


图 3-9. 与目标板的连接



#### 3.3.1. 目标板连接引脚分配

对于不同的器件和接口，编程连接器引脚功能有所不同。有关调试和数据流接口，请参见以下引脚分配表。

**注：**有关更多信息和图，请参见所用器件的数据手册以及特定接口的应用笔记。

表 3-1. 调试接口的引脚分配

| MPLAB <sup>®</sup><br>PICkit <sup>™</sup> 5 在线<br>调试器连接器 |          | 调试接口                        |                            |                            |                          |                     |                    |            |         |         | 目标 <sup>5</sup> 连接器 |              |
|--|----------|-----------------------------|----------------------------|----------------------------|--------------------------|---------------------|--------------------|------------|---------|---------|---------------------|--------------|
| 8 引脚 SIL <sup>1</sup>                                    |          | ICSP <sup>™</sup><br>(MCHP) | MIPS <sup>®</sup><br>EJTAG | Cortex <sup>®</sup><br>SWD | AVR <sup>®</sup><br>JTAG | AVR dW <sup>3</sup> | AVR<br>UPDI        | AVR<br>PDI | AVR ISP | AVR TPI | 8 引<br>脚 SIL        | 6 引<br>脚 SIL |
| 引脚<br>编号   | 引脚名<br>称 |                             |                            |                            |                          |                     |                    |            |         |         | 引脚<br>编号            | 引脚<br>编号     |
| 1  | TVPP     | MCLR/<br>VPP                | MCLR                       | RESET                      |                          |                     | RESET <sup>4</sup> |            |         |         | 1                   | 1            |
| 2  | TVDD     | VDD                         | VDD/<br>VDDIO              | VDD                        | VTG                      | VTG                 | VTG                | VTG        | VTG     | VTG     | 2                   | 2            |
| 3  | GND      | GND                         | GND                        | GND                        | GND                      | GND                 | GND                | GND        | GND     | GND     | 3                   | 3            |
| 4  | PGD      | DAT                         | TDO                        | SWO <sup>2</sup>           | TDO                      |                     | DAT <sup>4</sup>   | DAT        | MISO    | DAT     | 4                   | 4            |
| 5  | PGC      | CLK                         | TCK                        | SWCLK                      | TCK                      |                     |                    |            | SCK     | CLK     | 5                   | 5            |
| 6  | TAUX     |                             |                            |                            | RESET                    | RESET/<br>dW        |                    | CLK        | RESET   | RESET   | 6                   | 6            |
| 7  | TTDI     |                             | TDI                        |                            | TDI                      |                     |                    |            | MOSI    |         | 7                   |              |
| 8  | TTMS     |                             | TMS                        | SWDIO <sup>2</sup>         | TMS                      |                     |                    |            |         |         | 8                   |              |

1. 使用 6 引脚插座将导致引脚 7 和引脚 8 的功能丧失，进而影响 EJTAG、JTAG、SWD 和 ISP。

2. SWO 用于跟踪。SWDIO 用于调试。

3. dW = debugWIRE

4. 在一些器件中，该引脚可用于通过高电压脉冲重新激活统一编程和调试接口（Unified Program and Debug Interface, UPDI）功能。有关详细信息，请参见器件数据手册。

5. 这些是假定与调试器主机（SIL）相似的目标连接器示例。

表 3-2. 数据流接口的引脚分配

| MPLAB <sup>®</sup> PICkit <sup>™</sup> 5 连接器 |  | 数据流                                    |  | 目标 <sup>1</sup> 连接器 |
|--|--|--|--|---------------------|
| 8 引脚 SIL <sup>2</sup>                        |  | PIC <sup>®</sup> 和 AVR <sup>®</sup> 器件 | Arm <sup>®</sup> Cortex <sup>®</sup> 器件 <sup>3</sup> | 8 引脚 SIL            |
| 引脚编号   |  | 虚拟 UART/CDC <sup>4</sup>               | 虚拟 UART/CDC <sup>4</sup>                             | 引脚编号                |
| 1  |  |  |  | 1                   |
| 2  |  | VTG                                    | VTG  | 2                   |
| 3  |  | GND                                    | GND  | 3                   |
| 4  |  |  | TX（目标）   | 4                   |
| 5  |  |  |  | 5                   |
| 6  |  |  |  | 6                   |
| 7  |  | TX（目标）                                 | RX（目标）   | 7                   |
| 8  |  | RX（目标）                                 |  | 8                   |

1. 这是假定与调试器主机（SIL）相似的目标连接器示例。

2. 数据流传输需要使用 8 引脚连接器。使用 6 引脚连接器将导致引脚 7 和引脚 8 的功能丧失。

3. 为了适应其他接口的接线，需要移动 RX 引脚与 TX 引脚的位置。

4. 使用 JTAG 的器件无法使用 UART。

### 3.3.2. 调试器适配器板

对于传统连接，可使用[调试器适配器板](#)（AC102015）。

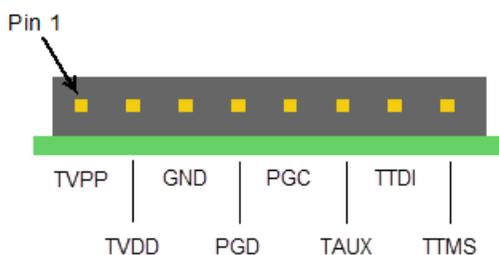
可按照下面部分所示将 MPLAB PICkit 5 连接到调试器适配器板。然后，可使用适配器板上的各种连接来连接具体目标板（见以下主题）。

或者，也可使用 [AVR 编程适配器板](#)（AC31S18A），该板适用于 MPLAB PICkit 5、PICkit 4 和 Snap。

### 使用 8 引脚直插式连接器

使用单列直插式（SIL）连接器时，工具直接连接到适配器板（确保工具上的引脚 1 与适配器板上的引脚 1 对齐）。如果使用 6 引脚 SIL 插座，则不连接引脚 7（TMS）和引脚 8（TDI）。

图 3-10. 调试器适配器板上的 SIL 连接

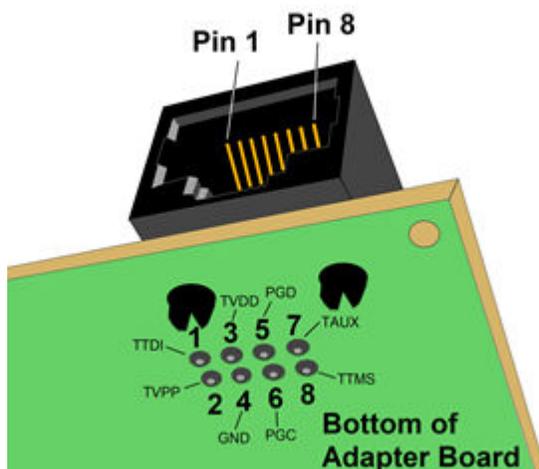


### 使用 8 引脚模块化连接器

对于 MPLAB PICkit 5，建议使用上述 8 引脚直插式连接器。

使用 AC164110（RJ-11 转 ICSP 适配器）时，可以使用 6 引脚模块化线缆将调试器连接到模块化连接器，此时将导致适配器板上的引脚 8（TMS）和引脚 1（TDI）断开连接。

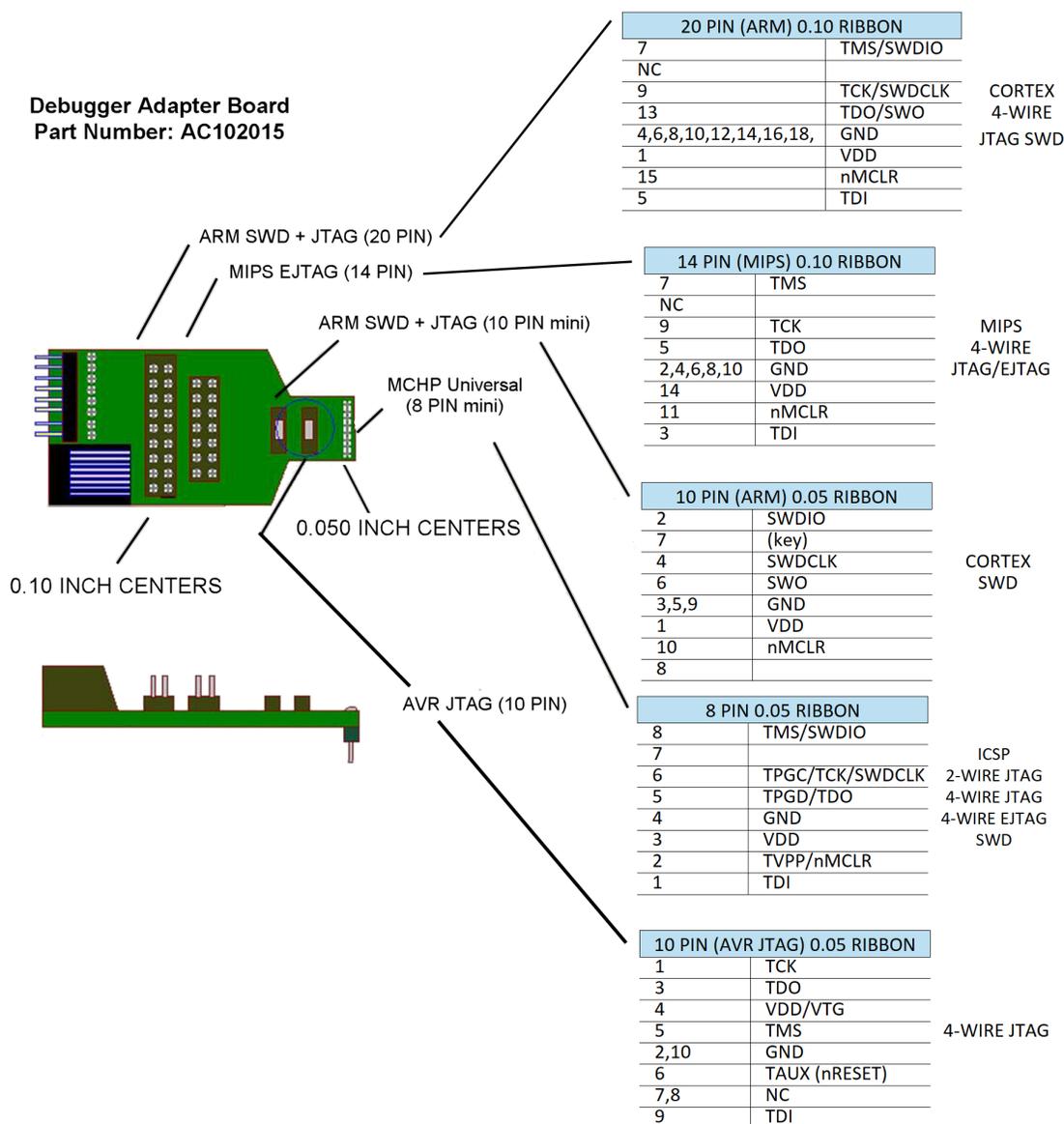
图 3-11. 调试器适配器板上的 RJ-45



#### 3.3.2.1. 适配器板引脚分配

此连接板支持 JTAG、SWD、ICSP 和 AVR 协议。

图 3-12. MPLAB® PICKIT™ 5 适配器板 (AC102015) 引脚分配



### 3.3.3. SAM/PIC32C MCU——JTAG/SWD 连接

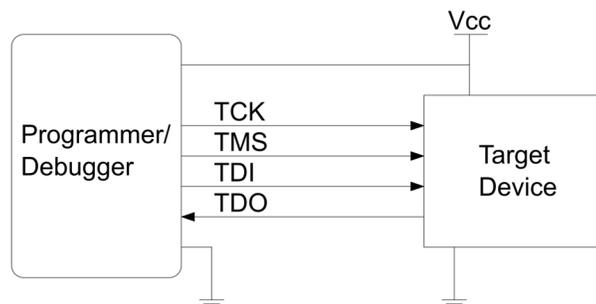
所有基于 Arm 的 SAM/PIC32C 器件均具有用于编程和调试的串行线调试 (SWD) 接口。此外，一些器件还具有用于编程和调试的 JTAG 接口。查看器件数据手册可以确定具体器件支持的接口。

#### 3.3.3.1. JTAG 物理接口

JTAG 接口包含符合 IEEE® 1149.1 标准的四线测试访问端口 (TAP) 控制器。制定 IEEE 标准的目的是提供一种行业标准方式来高效测试电路板连接 (边界扫描)。Microchip AVR 和 SAM 器件已将该功能扩展为包含全面编程和片上调试支持。

要将该目标接口与 MPLAB X IDE 配合使用，请依次打开 **Project Properties** 窗口，**PICKIT 5** 类别，**Communications** 选项类别，然后选择 **4-wire JTAG**。

图 3-13. JTAG 接口基本框图



### 3.3.3.1.1. 连接到 SAM/PIC32C JTAG 目标板

MPLAB PICKit 5 可直接连接新设计，或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接和传统 20 引脚 100 mil JTAG 连接。

相关信息

[适配器板引脚分配](#)

### 3.3.3.1.2. SAM/PIC32C JTAG 引脚分配（Cortex-M 调试连接器）

下表列出了 SAM/PIC32C JTAG 引脚名称及说明。其中分别给出了 MPLAB PICKit 5 直接连接以及调试器适配器板 10 引脚和 20 引脚连接的引脚编号。

表 3-3. SAM/PIC32C JTAG 引脚及说明

| MPLAB PICKit 5 引脚 | 适配器板 (10 引脚) 引脚 | 适配器板 (20 引脚) 引脚         | 名称       | 说明  |
|-------------------|-----------------|-------------------------|----------|---|
| 8                 | 2               | 7                       | TMS      | 测试模式选择（从 MPLAB PICKit 5 发送到目标器件的控制信号）。  |
| 6                 | 7               | 2,3,9,11,17,19          | NC/AUX   | 建议不要连接。   |
| 5                 | 4               | 9                       | TCK      | 测试时钟（从 MPLAB PICKit 5 发送到目标器件的时钟信号）。  |
| 4                 | 8               | 5                       | TDO      | 测试数据输出（从目标器件发送到 MPLAB PICKit 5 的数据）。  |
| 3                 | 3, 5, 9         | 4,6,8,10,12,14,16,18,20 | GND      | 地。必须全部连接以确保 MPLAB PICKit 5 与目标器件共用相同的参考地。   |
| 2                 | 1               | 1                       | VDD\VTG* | VDD: MPLAB PICKit 5 为目标板供电（可选）或目标板为 MPLAB PICKit 5 供电（PTG）。<br>VTG: 目标参考电压。MPLAB PICKit 5 采样该引脚上的目标电压来为电压转换器正确供电。在该模式下，MPLAB PICKit 5 从该引脚消耗的电流不到 3 mA。 |
| 1                 | 10              | 15                      | MCLR     | 复位（可选）。用于复位目标器件。建议连接该引脚，因为这样 MPLAB PICKit 5 便可使目标器件保持复位状态，这对于某些情况下的调试是必不可少的。  |
| 7                 | 6               | 13                      | TDI      | 测试数据输入（从 MPLAB PICKit 5 发送到目标器件的数据）。  |

\* 切记要在该引脚与 GND 之间连接去耦电容。请参见 [AN4451: SAMA5D2 Hardware Design Considerations](#)。

### 3.3.3.2. SAM/PIC32C SWD 接口

Arm SWD 接口是 JTAG 接口的子集，使用 TCK 和 TMS 引脚。

#### 3.3.3.2.1. 连接到 SAM/PIC32C SWD 目标板

MPLAB PICKit 5 可直接连接新设计，或者使用适配器板进行传统 10 引脚 50 mil SWD 连接和传统 20 引脚 100 mil SWD 连接。

## 相关信息

### 适配器板引脚分配

#### 3.3.3.2.2. SAM/PIC32C SWD 引脚分配

下表列出了 SAM/PIC32C SWD 引脚名称及说明。其中分别给出了 MPLAB PICKit 5 直接连接以及调试器适配器板 10 引脚和 20 引脚连接的引脚编号。

表 3-4. SAM/PIC32C SWD 引脚及说明

| MPLAB PICKit 5 引脚 | 适配器板 (10 引脚) 引脚 | 适配器板 (20 引脚) 引脚         | 名称       | 说明  |
|-------------------|-----------------|-------------------------|----------|---|
| 8                 | 2               | 7                       | SWDIO    | 串行线调试数据输入/输出。   |
| 5                 | 4               | 9                       | SWDCLK   | 串行线调试时钟。  |
| 4                 | 8               | 5                       | SWO      | 串行线输出 (与 ITM 配合使用——并非所有器件上均实现)。   |
| 3                 | 3, 5, 9         | 4,6,8,10,12,14,16,18,20 | GND      | 地   |
| 2                 | 1               | 1                       | VDD\VTG* | VDD: MPLAB PICKit 5 为目标板供电 (可选) 或目标板为 MPLAB PICKit 5 供电 (PTG)。<br>VTG: 目标参考电压。MPLAB PICKit 5 采样该引脚上的目标电压来为电压转换器正确供电。在该模式下, MPLAB PICKit 5 从该引脚消耗的电流不到 1 mA。 |
| 1                 | 10              | 15                      | MCLR     | 复位 (可选)。用于复位目标器件。建议连接该引脚, 因为这样 MPLAB PICKit 5 便可使目标器件保持复位状态, 这对于某些情况下的调试是必不可少的。   |

\* 切记要在该引脚与 GND 之间连接去耦电容。请参见 [AN4451: SAMA5D2 Hardware Design Considerations](#)。

#### 3.3.4. AVR MCU——各种连接

AVR 器件具有多种编程和调试接口。查看器件数据手册可以确定具体器件支持的接口。

##### 3.3.4.1. AVR MCU——JTAG 连接

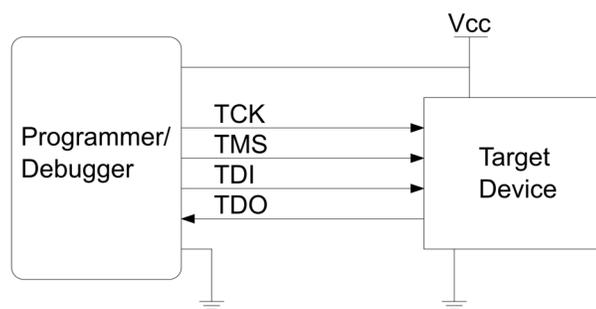
一些 AVR 器件具有用于编程和调试的 JTAG 接口。查看器件数据手册可以确定具体器件支持的接口。

###### 3.3.4.1.1. JTAG 物理接口

JTAG 接口包含符合 IEEE® 1149.1 标准的四线测试访问端口 (TAP) 控制器。制定 IEEE 标准的目的是提供一种行业标准方式来高效测试电路板连接 (边界扫描)。Microchip AVR 和 SAM 器件已将该功能扩展为包含全面编程和片上调试支持。

要将该目标接口与 MPLAB X IDE 配合使用, 请依次打开 **Project Properties** 窗口, **PICKit 5** 类别, **Communications** 选项类别, 然后选择 **4-wire JTAG**。

图 3-14. JTAG 接口基本框图



#### 连接到 AVR JTAG 目标板

MPLAB PICKit 5 可直接连接新设计, 或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接。

## 相关信息

### 适配器板引脚分配

## AVR JTAG 引脚分配

下表列出了 AVR JTAG 引脚名称及说明。其中分别给出了 MPLAB PICkit 5 直接连接和调试器适配器板 10 引脚连接的引脚编号。

表 3-5. AVR® JTAG 引脚及说明

| MPLAB PICkit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 名称         | 说明  |
|-------------------|-------------------|------------|---|
| 1                 | 7, 8              | NC         | 未连接。  |
| 2                 | 4                 | VDD\VTG*   | VDD: MPLAB PICkit 5 为目标板供电（可选）或目标板为 MPLAB PICkit 5 供电（PTG）。<br>VTG: 目标参考电压。MPLAB PICkit 5 采样该引脚上的目标电压来为电压转换器正确供电。在该模式下，MPLAB PICkit 5 从该引脚消耗的电流不到 3 mA。 |
| 3                 | 2, 10             | GND        | 地。必须全部连接以确保 MPLAB PICkit 5 与目标器件共用相同的参考地。   |
| 4                 | 3                 | TDO        | 测试数据输出（从目标器件发送到 MPLAB PICkit 5 的数据）。  |
| 5                 | 1                 | TCK        | 测试时钟（从 MPLAB PICkit 5 发送到目标器件的时钟信号）。  |
| 6                 | 6                 | RESET/TAUX | 复位（可选）。用于复位目标器件。建议连接该引脚，因为这样 MPLAB PICkit 5 便可使目标器件保持复位状态，这对于某些情况下的调试是必不可少的。  |
| 7                 | 9                 | TDI        | 测试数据输入（从 MPLAB PICkit 5 发送到目标器件的数据）。  |
| 8                 | 5                 | TMS        | 测试模式选择（从 MPLAB PICkit 5 发送到目标器件的控制信号）。  |

\* 切记要在该引脚与 GND 之间连接去耦电容。请参见 [AVR042: AVR Hardware Design Consideration](#)。

### 3.3.4.2. AVR SPI 接口

在线编程使用目标 AVR 的内部串行外设接口（Serial Peripheral Interface, SPI）将代码下载到闪存和 EEPROM 存储器中。该接口并非调试接口。

#### 3.3.4.2.1. 连接到 AVR SPI 目标板

MPLAB PICkit 5 可直接连接新设计，或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接。



#### 重要:

编程 debugWIRE 使能（DWEN）熔丝时实际上会禁止 SPI 接口，即使 SPIEN 熔丝也已编程时也不例外。要重新使能 SPI 接口，必须在处于 debugWIRE 调试会话中时发出“disable debugWIRE”（禁止 debugWIRE）命令。以这种方式禁止 debugWIRE 时要求 SPIEN 熔丝已编程。如果 MPLAB X IDE 无法禁止 debugWIRE，可能是因为 SPIEN 熔丝尚未编程。此时，需要使用[高电压编程](#)接口来编程 SPIEN 熔丝。



#### 信息:

SPI 接口通常称为“ISP”，因为它是 Microchip AVR 产品上的第一个在线编程接口。现在可使用其他接口进行在线编程。

## 相关信息

### 适配器板引脚分配

#### 3.3.4.2.2. AVR SPI 引脚分配

对于传统应用 PCB（包含带 SPI 接口的 AVR 器件），可能使用了下图所示的引脚分配。

图 3-15. 传统 SPI 插座引脚分配

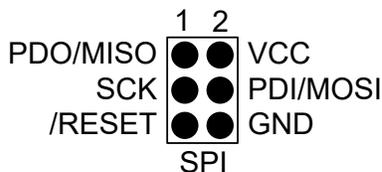


表 3-6. SPI 引脚映射

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 传统 SPI 引脚 | 名称      | 说明          |
|-------------------|-------------------|-----------|---------|-------------|
| 1                 | 7 或 8 (NC)        |           |         |             |
| 2                 | 4 (VDD/VTG)       | 2         | VCC/VTG | 目标电压 (参考电压) |
| 3                 | 2 或 10 (GND)      | 6         | GND     | 地           |
| 4                 | 3 (TDO)           | 1         | MISO    | 主输入从输出      |
| 5                 | 1 (TCK)           | 3         | SCK     | SPI 时钟      |
| 6                 | 6 (RESET/TAUX)    | 5         | RESET   |             |
| 7                 | 9 (TDI)           | 4         | MOSI    | 主输出从输入      |
| 8                 | 5 (TMS)           |           |         |             |

### 3.3.4.3. AVR UPDI 接口

统一编程和调试接口 (UPDI) 为 Microchip 专有接口, 用于器件的外部编程和片上调试。它是编程和调试接口 (PDI) 双线物理接口的后继产品, 在所有 AVR XMEGA® 器件上均有提供。UPDI 是一个单线接口, 可与目标器件进行双向半双工异步通信, 从而实现编程和调试。

UPDI 目前有三种配置:

- UPDI 功能与  $\overline{\text{RESET}}$  或 GPIO 共用一个引脚。当已选择  $\overline{\text{RESET}}$  或 GPIO 功能时, 如果需要 UPDI 功能, 则需在该引脚上施加高电压 (HV) 脉冲以重新激活 UPDI 功能。  
旧款 AVR 器件 (ATtiny) 采用该配置, 所需的 HV 脉冲为 12V。
- UPDI 功能使用专用引脚, 因此始终可用。 $\overline{\text{RESET}}$  与 GPIO 共用一个引脚。  
新款 AVR 器件 (ATmega0 和 AVR DA/DB 等) 采用该配置, 无需 HV 脉冲。
- UPDI 功能与 GPIO 共用一个引脚。 $\overline{\text{RESET}}$  使用专用引脚。当已选择 GPIO 功能时, 如果需要 UPDI 功能, 则需在  $\overline{\text{RESET}}$  引脚上施加 HV 脉冲以重新激活 UPDI 功能。  
新款 AVR 器件 (AVR DD) 采用该配置, 所需的 HV 脉冲约为  $V_{DD} + 2V$ 。有关实际电压值范围, 请参见器件数据手册。

有关所用器件的配置, 请参见该器件的数据手册。

#### 3.3.4.3.1. 连接到 AVR UPDI 目标板

MPLAB PICKit 5 可直接连接新设计, 或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接 (对于 6 引脚 UPDI 接口)。

相关信息

[适配器板引脚分配](#)

#### 3.3.4.3.2. AVR UPDI 引脚分配

对于传统应用 PCB (包含带 UPDI 接口的 AVR 器件), 可能使用了下图所示的引脚分配 (适用于所有配置)。

图 3-16. 传统 UPDI 插座引脚分配

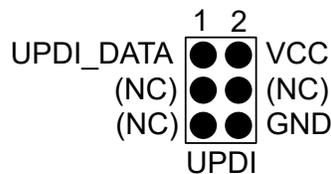


表 3-7. 传统 UPDI 引脚映射

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 传统 UPDI 引脚 | 名称         | 说明                             |
|-------------------|-------------------|------------|------------|--------------------------------|
| 1                 | 7 或 8 (NC)        |            |            |                                |
| 2                 | 4 (VDD/VTG)       | 2          | VCC/VTG    | 目标电压 (参考电压)。                   |
| 3                 | 2 或 10 (GND)      | 6          | GND        | 地                              |
| 4                 | 3 (TDO)           | 1          | UPDI_DATA* | UPDI 数据。该引脚可能有其他复用功能, 具体取决于器件。 |
| 5                 | 1 (TCK)           |            |            |                                |
| 6                 | 6 (RESET/TAUX)    |            |            |                                |
| 7                 | 9 (TDI)           |            |            |                                |
| 8                 | 5 (TMS)           |            |            |                                |

\* 若要使用 UPDI 功能, 可能需在该引脚上施加高电压 (HV) 脉冲。有关 UPDI 配置, 请参见器件数据手册。

### 3.3.4.4. AVR PDI 接口

编程和调试接口 (PDI) 为 Microchip 专有接口, 用于器件的外部编程和片上调试。PDI 物理接口是一个 2 引脚接口, 可与目标器件进行双向半双工同步通信。

#### 3.3.4.4.1. 连接到 AVR PDI 目标板

MPLAB PICKit 5 可直接连接新设计, 或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接 (对于 6 引脚 PDI 接口)。

#### 相关信息

[适配器板引脚分配](#)

#### 3.3.4.4.2. AVR PDI 引脚分配

对于传统应用 PCB (包含带 PDI 接口的 AVR 器件), 可能使用了下图所示的引脚分配。

图 3-17. 传统 PDI 插座引脚分配

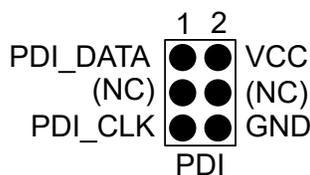


表 3-8. PDI 引脚映射

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 传统 PDI 引脚 | 名称       | 说明           |
|-------------------|-------------------|-----------|----------|--------------|
| 1                 | 7 或 8 (NC)        |           |          |              |
| 2                 | 4 (VDD/VTG)       | 2         | VCC/VTG  | 目标电压 (参考电压)。 |
| 3                 | 2 或 10 (GND)      | 6         | GND      | 地            |
| 4                 | 3 (TDO)           | 1         | PDI_DATA | PDI 数据       |
| 5                 | 1 (TCK)           |           |          |              |
| 6                 | 6 (RESET/TAUX)    | 5         | PDI_CLK  | PDI 时钟       |
| 7                 | 9 (TDI)           |           |          |              |

表 3-8. PDI 引脚映射 (续)

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 传统 PDI 引脚 | 名称 | 说明 |
|-------------------|-------------------|-----------|----|----|
| 8                 | 5 (TMS)           |           |    |    |

### 3.3.4.5. AVR TPI 接口

在一些 tinyAVR® 器件上，Tiny 编程接口 (TPI) 为仅编程接口，并非调试接口，这些器件没有片上调试 (On-Chip Debugging, OCD) 功能。

#### 3.3.4.5.1. 连接到 AVR TPI 目标板

MPLAB PICKit 5 可直接连接新设计，或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接 (对于 6 引脚 TPI 接口)。

相关信息

[适配器板引脚分配](#)

#### 3.3.4.5.2. AVR TPI 引脚分配

对于传统应用 PCB (包含带 TPI 接口的 AVR 器件)，可能使用了下图所示的引脚分配。

图 3-18. 传统 TPI 插座引脚分配

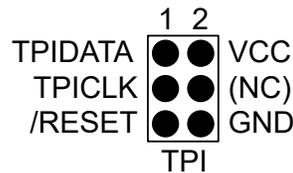


表 3-9. TPI 引脚映射

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚 | 传统 TPI 引脚 | 名称      | 说明           |
|-------------------|-------------------|-----------|---------|--------------|
| 1                 | 7 或 8 (NC)        |           |         |              |
| 2                 | 4 (VDD/VTG)       | 2         | VCC/VTG | 目标电压 (参考电压)。 |
| 3                 | 2 或 10 (GND)      | 6         | GND     | 地            |
| 4                 | 3 (TDO)           | 1         | DATA    | TPI 数据       |
| 5                 | 1 (TCK)           | 3         | CLOCK   | TPI 时钟       |
| 6                 | 6 (RESET/TAUX)    | 5         | RESET   | 复位器件。        |
| 7                 | 9 (TDI)           |           |         |              |
| 8                 | 5 (TMS)           |           |         |              |

### 3.3.4.6. AVR debugWIRE 接口

debugWIRE 接口适用于低引脚数器件。与使用 4 个引脚的 JTAG 接口不同的是，debugWIRE 仅使用一个引脚 (RESET) 即可与调试器工具进行双向半双工异步通信。

注：

debugWIRE 接口无法用作编程接口。这意味着，必须额外使用 SPI 接口 (如 [AVR SPI 接口](#) 所示) 对目标器件进行编程。

使用 debugWIRE 启动调试会话时，将使用 debugWIRE 接口对闪存进行编程。这并非出厂编程时可以考虑的一种选项。

当 debugWIRE 使能 (DWEN) 熔丝已编程而锁定位尚未编程时，将激活目标器件内的 debugWIRE 系统。RESET 引脚将配置为使能上拉的线与 (漏极开路) 双向 I/O 引脚，并成为目标器件与调试器之间的通信网关。

### 3.3.4.6.1. 连接到 AVR debugWIRE 目标板

MPLAB PICKit 5 可直接连接新设计，或者使用适配器板进行传统 10 引脚 50 mil JTAG 连接（对于 6 引脚 debugWIRE/SPI 接口）。

尽管 debugWIRE 接口只需要使用一根信号线（ $\overline{\text{RESET}}$ ）、VCC 和 GND 即可正常工作，但仍建议使用完整的 SPI 连接器，以便可通过 SPI 编程使能和禁止 debugWIRE 接口。

使能 DWEN 熔丝时，将在内部改写 SPI 接口，以便 OCD 模块能够控制  $\overline{\text{RESET}}$  引脚。debugWIRE OCD 可暂时禁止自身，从而释放对  $\overline{\text{RESET}}$  线的控制。SPI 接口随后将再次可用（仅当编程 SPIEN 熔丝后），以允许使用 SPI 接口取消对 DWEN 熔丝的编程。如果在 DWEN 熔丝取消编程之前重启电源，debugWIRE 模块将再次控制  $\overline{\text{RESET}}$  引脚。通常，MPLAB X IDE 或 Microchip Studio 将自动切换接口，但也可以使用 Microchip Studio 的属性对话框中的调试选项卡上的按钮手动切换。

**注：**强烈建议由 MPLAB X IDE 或 Microchip Studio 来负责将 DWEN 熔丝置 1 和清零。

如果已编程目标 AVR 器件的锁定位，则无法使用 debugWIRE 接口。始终确保在编程 DWEN 熔丝之前清零锁定位，并且在 DWEN 熔丝编程期间切勿将锁定位置 1。如果 debugWIRE 使能（DWEN）熔丝和锁定位同时置 1，可以使用高电压编程进行全片擦除，从而清零锁定位。当锁定位清零后，将重新使能 debugWIRE 接口。仅当取消对 DWEN 熔丝的编程时，SPI 接口才能读取熔丝、读取签名和执行全片擦除。

#### 相关信息

[适配器板引脚分配](#)

### 3.3.4.6.2. AVR debugWIRE 引脚分配

对于传统应用 PCB（包含带 debugWire 接口的 AVR 器件），可能使用了下图所示的引脚分配。

图 3-19. 传统 debugWIRE（SPI）插座引脚分配

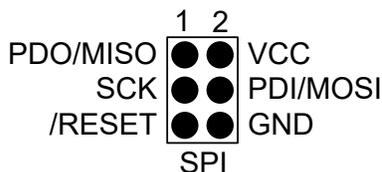


表 3-10. debugWIRE 引脚映射

| MPLAB PICKit 5 引脚 | 适配器板 AVR® JTAG 引脚                    | 传统 debugWIRE 引脚 | 名称                                    | 说明                         |
|-------------------|--------------------------------------|-----------------|---------------------------------------|----------------------------|
| 1                 | 7 或 8 (NC)                           |                 |                                       |                            |
| 2                 | 4 (VDD/VTG)                          | 2               | VCC/TTG                               | 目标电压（参考电压）。                |
| 3                 | 2 或 10 (GND)                         | 6               | GND                                   | 地                          |
| 4                 | 3 (TDO)                              |                 |                                       |                            |
| 5                 | 1 (TCK)                              |                 |                                       |                            |
| 6                 | 6 ( $\overline{\text{RESET}}$ /TAUX) | 5               | $\overline{\text{RESET}}$ 或 debugWIRE | 该引脚用于复位器件或发送 debugWIRE 数据。 |
| 7                 | 9 (TDI)                              |                 |                                       |                            |
| 8                 | 5 (TMS)                              |                 |                                       |                            |

### 3.3.5. PIC32M 连接

基于 MIPS 的 PIC32M 器件使用 EJTAG 进行调试和编程。

#### 3.3.5.1. 连接到 PIC32M EJTAG 目标板

MPLAB PICKit 5 可直接连接新设计，或者使用适配器板进行传统 14 引脚 10 mil JTAG/EJTAG 连接。

### 3.3.5.2. PIC32M EJTAG 引脚分配——4 线 JTAG

下表列出了 PIC32M EJTAG 引脚名称及说明。其中分别给出了 MPLAB PICKit 5 直接连接和调试器适配器板 14 引脚连接的引脚编号。

表 3-11. PIC32M JTAG 连接器 14 引脚说明

| MPLAB PICKit 5 引脚 | 适配器板 (14 引脚) 引脚 | 名称   | 说明  |
|-------------------|-----------------|------|---|
| 1                 | 11              | MCLR | 复位 (可选)。用于复位目标器件。建议连接该引脚, 因为这样 MPLAB PICKit 5 便可使目标器件保持复位状态, 这对于某些情况下的调试是必不可少的。 |
| 2                 | 14              | VDD  | MPLAB PICKit 5 为目标板供电 (可选) 或目标板为 MPLAB PICKit 5 供电 (PTG)。                       |
| 3                 | 2, 4, 6, 8, 10  | GND  | 地。必须全部连接以确保 MPLAB PICKit 5 与目标器件共用相同的参考地。                                       |
| 4                 | 3               | TDO  | 测试数据输出 (从目标器件发送到 MPLAB PICKit 5 的数据)。   |
| 5                 | 9               | TCK  | 测试时钟 (从 MPLAB PICKit 5 发送到目标器件的时钟信号)。   |
| 6                 | 1               | NC   | 未连接。  |
| 7                 | 5               | TDI  | 测试数据输入 (从 MPLAB PICKit 5 发送到目标器件的数据)。   |
| 8                 | 7               | TMS  | 测试模式选择 (从 MPLAB PICKit 5 发送到目标器件的控制信号)。   |

### 3.3.6. PIC MCU——ICSP 连接

MPLAB® PICKit™ 5 在线调试器通过 ICSP (在线串行编程) 连接支持 PIC 单片机 (MCU) 和 dsPIC 数字信号控制器 (DSC) 的调试和编程。PICKit 5 SIL 连接器使用两个器件 I/O 引脚和复位线来实现在线调试和 ICSP。

#### 3.3.6.1. ICSP™ 目标板连接

在大多数 MPLAB 调试工具上, 使用 ICSP 模块化连接器或直插式连接器将调试器直接连接到 PIC MCU 目标板。调试器适配器板的连接与目标板的连接相同。

如果调试器与目标板采用不同的连接方式 (模块化与直插式, 或直插式与模块化), 可购买一个小型适配器以确保正确连接: “RJ11 转 ICSP 适配器” (AC164110)。

图 3-20. 6 引脚 RJ11 转 ICSP™ 适配器

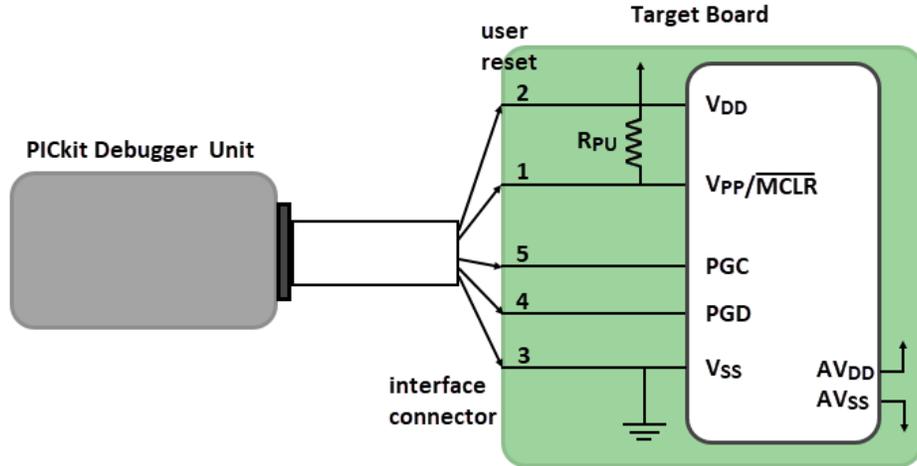


此外, MPLAB PICKit 5 还可使用适配器板提供一种 8 引脚 50 mil Microchip 通用连接, 对 6 引脚和 8 引脚 ICSP 接口均适用。

#### 3.3.6.2. ICSP™ 目标板连接电路

下图显示了 MPLAB PICKit 5 在线调试器与目标板上 ICSP 连接器的互连。该图还显示了从连接器到目标 PCB 上器件的连线。建议在  $V_{PP}/\overline{MCLR}$  线与  $V_{DD}$  之间连接一个上拉电阻  $R_{PU}$ , 以便将该线选通为低电平以复位器件。

图 3-21. ICSP 目标板连接



其中  $R_{pu} = 10\text{ k}\Omega - 50\text{ k}\Omega$

## 4. 工作原理

本章简要概述了 MPLAB PICKit 5 在线调试器系统的工作原理，旨在提供足够的信息，帮助用户设计与调试器兼容的目标板，以进行调试和编程操作。本章还讨论了在线调试和编程的基本原理，以使用户能够快速解决遇到的问题。

### 4.1. 调试/编程快速参考

下表给出了将 MPLAB PICKit 5 在线调试器用作调试或编程工具时的快速参考。

 **注意：** MPLAB PICKit 5 不支持调试头。

表 4-1. 调试与编程操作

| 项目   | 调试  | 编程   |
|--|---|--|
| 所需硬件   | 计算机、目标板（Microchip 演示板或您自己设计的电路板）、线缆，以及用于将计算机连接到目标板的可选适配器板。<br>调试工具、USB 线缆、供电集线器或目标电源。<br>带片上调试电路的器件。  | 器件（带或不带片上调试电路），以及用于 PTG 的智能手机或平板电脑。  |
| 所需软件   | MPLAB X IDE<br>应用程序代码（示例代码（即，来自 MPLAB Discover）或您自己编写的代码）。  | MPLAB X IDE 或 MPLAB IPE<br>预编译的代码——Hex 文件，以及智能手机或平板电脑的 PTG 应用程序。   |
| MPLAB X IDE 选择                               | <b>Project Properties, Hardware Tool</b> （硬件工具）<br><b>Debug Main Project</b> （调试主项目）图标<br> | 选择 <b>PICKit 5</b> 。<br><b>Make and Program Device</b> （编译并编程器件）<br>图标<br> |
| 编程操作   | 将应用程序代码烧写到器件中。这可以是程序存储器的任何范围，具体取决于 <b>Project Properties</b> 对话框上的选择。此外，可能还有一个小的调试执行程序位于程序存储器中，其他调试资源可能保留。  | 将应用程序代码烧写到器件中。这可以是程序存储器的任何范围，具体取决于 <b>Project Properties</b> 对话框上的选择。  |
| 可用的调试功能                                      | 取决于器件——断点等。   | N/A  |
| 序列号快速编程（Serial Quick-Time Programming, SQTP） | N/A   | 使用 MPLAB IPE 生成 SQTP 文件。   |
| 命令行操作  | 使用 MDB 命令行实用程序，默认路径为：<br>C:\Program Files\Microchip\<br>MPLABX\vx.xx\mplab_platform\bin\mdb.bat。  | 使用 IPECMD，默认路径为：C:\Program<br>Files\Microchip\<br>MPLABX\vx.xx\mplab_platform\mplab_ipe\<br>ipecmd.exe。  |
| 脱机编程（PTG）                                    | N/A   | 使用 PTG 应用程序将 MPLAB PICKit 5 中插入的 microSDHC 卡所存储的应用程序代码编程到器件中。  |

### 4.2. 操作概述

**注：** 确保在完成第 5 步之前已阅读 [电源连接](#)、[PC 与智能手机连接](#) 和 [目标板连接](#)。

要使用 MPLAB PICKit 5 在线调试器来调试和编程代码：

- 从 [MPLAB X IDE](#) 网页下载最新的 MPLAB X IDE 并进行安装。MPLAB IPE 包含在 MPLAB X IDE 安装程序中。有关如何创建项目来开发应用程序代码以及如何调试代码的信息，请参见 [MPLAB X IDE 文档](#)。
- 在 [MPLAB Discover](#) 中查找代码示例，或者在 GitHub 上搜索 Microchip 内容，例如 [Microchip PIC 和 AVR 示例](#)。

3. 在 [MPLAB XC 编译器](#) 网页上查找适用于您应用中的器件的编译器。
4. 购买 MPLAB PICKit 5，可另购调试器适配板。
5. 启动 [MPLAB X IDE](#)。使用 USB 线缆将 MPLAB PICKit 5 插入计算机。确保目标板正确连接。
6. 在 MPLAB X IDE 中打开您的项目或示例项目。右键单击 **Projects**（项目）选项卡中的项目名称，然后选择 **Properties**。在 **Project Properties** 对话框中，确保在 **Connected Hardware Tool**（已连接的硬件工具）下选择了 **PICKit 5**。然后，在配置和设置选项下从 **Option Categories**（选项类别）列表中选择 **PICKit 5** 类别。  
**注：** 如果需要，可在此处选择目标板的电源。
7. 有关针对您应用中所用器件的具体操作的详细信息，请阅读以下章节。
8. 有关器件的调试器限制的完整列表，请参见 MPLAB X IDE 中的在线帮助文件（**Help > Help Contents > Hardware Tool Reference Help > Limitations - Emulators and Debuggers**（帮助 > 帮助内容 > 硬件工具参考帮助 > 限制——仿真器和调试器））。

### 4.3. 关于高电压

“高电压”一词多义。下面将讨论说明该术语在过去和现在的不同定义。

#### AVR 器件

##### 高电压编程——HVSP 和 HVPP

旧款 AVR 器件具有一种称为高电压编程的编程接口，该接口支持串行编程（HVSP）和并行编程（HVPP）两种方式。通常，在高电压编程过程中，需要在  $\overline{\text{RESET}}$  引脚上施加 12V 电压。

当配置位（熔丝）被错误置 1 或清零时，有时需要通过高电压编程才能恢复。例如：

- DWEN 和锁定位均置 1：无法使用 debugWIRE。
- DWEN 置 1 且 SPIEN 清零：困在 debugWIRE 模式，无法使用 SPI。
- JTAGEN 清零：无法使用 JTAG。

**工具支持：** 目前所有 MPLAB 硬件工具均不支持该编程方法。因此，务必注意有关上述问题的所有警告并遵循相关说明。

##### 高电压脉冲——UPDI 引脚

根据所用的器件，目前有两种配置需要在引脚上施加高压（HV）脉冲才能使能 UPDI 功能：

1. UPDI 功能与  $\overline{\text{RESET}}$  或 GPIO 共用一个引脚。当已选择  $\overline{\text{RESET}}$  或 GPIO 功能时，如果需要 UPDI 功能，则需在该引脚上施加高电压（HV）脉冲以重新激活 UPDI 功能。  
旧款 AVR 器件采用该配置，所需的 HV 脉冲为 12V。
2. UPDI 功能与 GPIO 共用一个引脚。 $\overline{\text{RESET}}$  使用专用引脚。当已选择 GPIO 功能时，如果需要 UPDI 功能，则需在  $\overline{\text{RESET}}$  引脚上施加 HV 脉冲以重新激活 UPDI 功能。  
新款 AVR 器件采用该配置，所需的 HV 脉冲约为  $V_{DD} + 2V$ 。有关实际电压值范围，请参见器件数据手册。

**工具支持：** 除了 MPLAB Snap（不支持高电压）之外，目前所有 MPLAB 硬件工具均支持 HV 脉冲。

#### PIC 器件

##### 高电压编程与低电压编程

对于高电压编程，旧款 PIC 器件需要在较高的电压（12V）下进行编程，但新款器件可使用较低的电压（在  $V_{PP}$  引脚上施加超过 9V 的电压）。

对于低电压编程，编程电压  $V_{PP}$  不会超过  $V_{DD}$ 。

**工具支持：** 除了以下工具之外，目前所有 MPLAB 硬件工具均支持高电压编程和低电压编程：

- MPLAB Snap（不支持高电压）。
- MPLAB PICKit 4（仅在器件  $V_{DD}$  电压不低于 2.8V 时支持高电压编程。问题已于 10-10094-R6 及之后版本修正；请查看调试器主机背面的标签）。

#### 4.4. SAM 和 PIC32C Arm<sup>®</sup> 器件——片上调试

SAM 和 PIC32C 单片机均基于 Arm Cortex-M 内核。可用的调试功能取决于内核类型（见下表）。调试连接器支持 SWD 和 JTAG。

有关哪些器件具有哪些内核的更多信息，请参见 [32 位 PIC<sup>®</sup>](#) 和 [SAM 单片机](#) 或您所用器件的数据手册。另请参见由 Arm 提供的 [CoreSight 文档](#)。

表 4-2. Cortex<sup>®</sup>-M 调试和跟踪支持汇总

| Cortex <sup>®</sup> -M 类型 | 调试支持   |
|---------------------------|--|
| Cortex-M0+                | 调试可选：基本调试功能包括处理器暂停、单步执行、处理器内核寄存器访问、复位和硬故障向量捕捉、无限软件断点、系统存储器完全访问，以及 1/2/3/4 断点和 1/2 观察点功能。       |
| Cortex-M23                | 调试可选：基本调试功能包括处理器暂停、单步执行、处理器内核寄存器访问、复位和硬故障向量捕捉、无限软件断点、系统存储器完全访问，以及 1/2/3/4 断点和 1/2/3/4 观察点功能。   |
| Cortex-M4 和 M4F           | 调试可选：基本调试功能包括处理器暂停、单步执行、处理器内核寄存器访问、向量捕捉、无限软件断点、系统存储器完全访问，以及各种断点和 1/4 观察点功能。                    |
| Cortex-M7                 | Cortex-M7 调试功能包括处理器暂停、单步执行、处理器内核寄存器访问、向量捕捉、无限软件断点和系统存储器完全访问。处理器还支持实现期间配置的 4/8 个硬件断点和 2/4 个观察点。 |

#### 4.5. AVR 器件——片上调试（OCD）

片上调试模块是一种系统，允许开发人员从外部开发平台监视和控制器件中的执行情况，通常通过称为 *调试器* 或 *调试适配器的* 设备实现。

凭借 OCD 系统，执行应用程序时可在目标系统中保持准确的电气和时序特性，以便能够依条件或手动停止执行应用程序以检查程序流和存储器。

##### 运行模式

处于运行模式时，代码执行完全独立于 MPLAB PICKit 5。MPLAB PICKit 5 将持续监视目标器件是否出现暂停条件。出现暂停条件时，OCD 系统将通过其调试接口查询器件，以便用户查看器件的内部状态。

##### 停止模式

到达断点时，程序执行将暂停，但一些 I/O 可能如同未出现断点一样继续运行。例如，假定到达断点时，USART 刚刚开始发送。此时，即使内核处于停止模式，USART 也将继续全速运行，完成发送。

##### 硬件断点

目标 OCD 模块包含多个通过硬件实现的程序计数器比较器。当程序计数器与比较器寄存器之一中存储的值匹配时，OCD 进入停止模式。由于硬件断点需要使用 OCD 模块上的专用硬件，因此可用断点数取决于目标器件上实现的 OCD 模块的大小。通常，调试器会“保留”一个此类硬件比较器供内部使用。

##### 软件断点

软件断点是位于目标器件上的程序存储器中的 BREAK 指令。加载该指令后，程序执行将暂停，OCD 进入停止模式。要继续执行，OCD 必须发出“启动”命令。并非所有 Microchip 器件均配备支持 BREAK 指令的 OCD 模块。

##### 4.5.1. AVR 器件接口

**注：**如果在编程和调试使用 UPDI/PDI/TPI 接口的 AVR 单片机器件时出现问题，请查看所用工具的 [工程技术说明（ETN）](#)。

AVR 器件具有多种编程和调试接口。查看器件数据手册可以确定具体器件支持的接口。

- 所有 AVR E/D 器件和部分新款 tinyAVR 和 megaAVR<sup>®</sup> 器件具有用于编程和调试的 UPDI 接口。
- 一些 tinyAVR 器件具有 TPI 接口。TPI 只能用于编程器件。这些器件完全不具备片上调试功能。
- 一些 tinyAVR 器件和一些 megaAVR 器件具有 debugWIRE 接口，该接口连接到称为 tinyOCD 的片上调试系统。所有具有 debugWIRE 的器件还具有用于在线编程的 SPI 接口。
- 一些 megaAVR 器件具有用于编程和调试的 JTAG 接口以及片上调试系统（也称为 megaOCD）。所有具有 JTAG 的器件还具有 SPI 接口，该接口用作在线编程的替代接口。
- 所有 AVR XMEGA 器件均具有用于编程和调试的 PDI 接口。一些 AVR XMEGA 器件还具有用于编程和调试的 JTAG 接口。

表 4-3. 编程和调试接口汇总

| AVR <sup>®</sup> 器件系列  | UPDI | TPI  | SPI  | debugWIRE | JTAG | PDI  |
|------------------------|------|------|------|-----------|------|------|
| AVR E/D                | 新器件  |      |      |           |      |      |
| tinyAVR <sup>®</sup>   | 新器件  | 一些器件 | 一些器件 | 一些器件      |      |      |
| megaAVR <sup>®</sup>   | 新器件  |      | 一些器件 | 一些器件      | 一些器件 |      |
| AVR XMEGA <sup>®</sup> |      |      |      |           | 一些器件 | 所有器件 |

#### 4.5.1.1. UPDI OCD 特性

新款 tinyAVR 和 megaAVR 器件以及 AVR E/D 器件的 UPDI OCD 基于 UPDI 物理接口，这是一个单引脚的编程和调试接口。

tinyAVR 和 megaAVR 器件的其他特性包括：

- 针对器件地址空间（NVM、RAM 和 I/O）的存储器映射访问
- 对器件时钟频率没有限制
- 用户程序断点的数量不限
- 两个硬件断点
- 支持高级 OCD 特性
- 无需访问系统寄存器即可进行非侵入式运行时芯片监视
- 用于读取被锁定器件上闪存的 CRC 校验结果的接口

AVR E/D 器件的其他特性包括：

- 两个硬件断点
- 程序流、中断和软件断点有变化
- 运行时读取堆栈指针（Stack Pointer, SP）寄存器、程序计数器（Program Counter, PC）和状态寄存器（SREG）
- 在停止模式下可读写寄存器文件

如果器件的 UPDI 引脚具有复用功能，则可将该引脚重新配置为 GPIO 或/RESET 引脚。有关详细信息，请参见 [UPDI 高电压激活信息](#)。

旧款器件支持 debugWIRE OCD。有关 OCD 功能的更多信息，请参见 [debugWIRE OCD 功能](#)。

##### 4.5.1.1.1. UPDI OCD 特殊注意事项

UPDI 数据引脚（UPDI\_DATA）既可以是专用引脚，也可以是共用引脚，具体取决于目标 AVR 器件。共用的 UPDI 引脚需要使用 UPDI 或 RESET 引脚（具体取决于器件）上的高电压（HV）脉冲来激活。有关详细信息，请参见器件数据手册。

如果器件上包含循环冗余校验存储器扫描（Cyclic Redundancy Check Memory Scan, CRCSCAN）模块，则在调试期间，不得在连续后台模式下使用该模块。OCD 模块的硬件断点比较器资源有限，因此在需

要更多断点时，甚至在源代码单步执行期间，可将 BREAK 指令插入闪存（软件断点）。CRC 模块可能会将该断点误检为闪存内容损坏。

CRCSCAN 模块将配置为在引导前执行 CRC 扫描。如果 CRC 不匹配，器件不会进行引导并处于锁定状态。将器件从该状态恢复的惟一方法是执行全片擦除，然后编程有效闪存映像或禁止引导前 CRCSCAN（如果只进行全片擦除，将导致闪存清空且 CRC 无效，因此器件仍不会进行引导）。在该状态下对器件进行全片擦除时，软件前端将自动禁止 CRCSCAN 熔丝。

如果设计的目标应用 PCB 将使用 UPDI 接口，必须考虑以下注意事项才能确保正常工作：

- UPDI 线上的上拉电阻不得小于 10 kΩ。不得使用下拉电阻，使用 UPDI 时应将其移除。UPDI 物理接口支持推挽功能，因此只需使用弱上拉电阻即可防止在线路空闲时误触发启动位。
- 如果 UPDI 引脚将用作  $\overline{\text{RESET}}$  引脚，使用 UPDI 时必须断开所有稳定电容，因为这些电容将干扰接口的正常工作。
- 如果 UPDI 引脚用作  $\overline{\text{RESET}}$  或 GPIO 引脚，编程或调试期间必须断开线路上的所有外部驱动器，因为这些驱动器可能干扰接口的正常工作。

#### 4.5.1.2. debugWIRE OCD 特性

debugWIRE OCD 是一个具有有限功能集的专用 OCD 模块，专为低引脚数 AVR 器件设计。它具有以下特性：

- 完整程序流控制
- 可完全访问所有寄存器和存储区
- 用户程序断点的数量不限（使用 BREAK 指令）
- 基于目标时钟自动配置波特率

##### 4.5.1.2.1. debugWIRE 特殊注意事项

debugWIRE 通信引脚（dW）在物理上与外部复位（RESET）位于同一引脚上。因此，使能 debugWIRE 接口时，不支持外部复位源。

必须将目标器件上的 debugWIRE 使能（DWEN）熔丝置 1，debugWIRE 接口才能正常工作。当 Microchip AVR 器件出厂时，该熔丝默认处于未编程状态。debugWIRE 接口本身无法用于将该熔丝置 1。要将 DWEN 熔丝置 1，必须使用 SPI 模式。如果连接了所需的 SPI 引脚，软件前端将自动进行此处理。此外，也可在软件前端中使用 SPI 编程手动将其置 1。

**方法 1:** 尝试在带 debugWIRE 接口的器件上启动调试会话。如果未使能 debugWIRE 接口，软件前端将提示重试或尝试使用 SPI 编程使能 debugWIRE。如果已连接整个 SPI 插座，则将使能 debugWIRE，系统将要求重启目标板上的电源。必须满足这一条件才能使熔丝更改生效。

**方法 2:** 在 SPI 模式下打开 Microchip Studio 中的编程对话框，然后验证签名是否与正确的器件匹配。检查用于使能 debugWIRE 的 DWEN 熔丝。



#### 重要：

务必使 SPIEN 熔丝保持已编程状态，RSTDISBL 熔丝保持未编程状态！如果无法满足上述条件，器件将困在 debugWIRE 模式，需要通过高电压编程才能恢复 DWEN 设置。

要禁止 debugWIRE 接口，应使用高电压编程取消对 DWEN 熔丝的编程。此外，也可使用 debugWIRE 接口本身暂时禁止该接口，从而实现 SPI 编程，但前提是 SPIEN 熔丝已置 1。

**重要：**

如果 SPIEN 熔丝尚未编程，软件前端将无法完成此操作，必须使用高电压编程。

在 MPLAB X IDE 中，如果使能目标器件上的 debugWIRE 并尝试 SPI 编程会话，MPLAB X IDE 将提示先禁止 debugWIRE。在 Microchip Studio 中，这一操作必须在调试会话期间手动完成，具体方法是从“Debug”（调试）菜单中选中“Disable debugWIRE and Close”（禁止 debugWIRE 并关闭）菜单选项。DebugWIRE 将暂时禁止，软件前端将使用 SPI 编程来取消对 DWEN 熔丝的编程。

编程 DWEN 熔丝可使时钟系统的某些部分在所有休眠模式下运行。这将增加 AVR 器件在休眠模式下的功耗。因此，不使用 debugWIRE 时，应始终禁止 DWEN 熔丝。

如果设计的目标应用 PCB 将使用 debugWIRE，必须考虑以下注意事项才能确保正常工作：

- dW/（RESET）线上的上拉电阻不得小于 10 kΩ。debugWIRE 功能无需上拉电阻，因为调试器工具中会提供。
- 使用 debugWIRE 时必须断开 RESET 引脚上连接的所有稳定电容，因为这些电容将干扰接口的正常工作。
- 必须断开 RESET 线上的所有外部复位源或其他有效驱动器，因为它们可能干扰接口的正常工作。

切勿编程目标器件上的锁定位。debugWIRE 接口需要锁定位保持清零才能正常工作。

#### 4.5.1.2.2. debugWIRE 软件断点

与 Microchip megaAVR（JTAG）OCD 相比，debugWIRE OCD 的规模大幅缩小。这意味着，debugWIRE OCD 没有可供用户进行调试的程序计数器断点比较器。其中有一个比较器可用于运行到光标和单步执行操作，但其他用户断点不受硬件支持。

因此，调试器必须转为使用 AVR BREAK 指令。该指令可置于闪存中，加载执行时将导致 AVR CPU 进入停止模式。要在调试期间支持断点，调试器必须在用户请求断点时将 BREAK 指令插入闪存。原始指令必须进行高速缓存处理，以便后续替换。当单步跳过一条 BREAK 指令时，调试器必须执行高速缓存的原始指令来保留程序行为。在极端情况下，必须从闪存中删除 BREAK 指令，后续进行替换。上述所有情况均会导致在从断点单步执行时出现明显延时，如果目标时钟频率非常低，延时问题会更加明显。

因此建议尽可能遵循以下准则：

- 调试期间始终以尽可能高的频率运行目标器件。debugWIRE 物理接口由目标时钟驱动。
- 尝试最大程度减少添加和删除断点的次数，因为每次都需要替换目标器件上的闪存页。
- 尝试每次添加或删除少量断点，以最大程度减少闪存页写操作次数。
- 如有可能，避免在双字指令上放置断点。

#### 4.5.1.2.3. 了解 debugWIRE 和 DWEN 熔丝

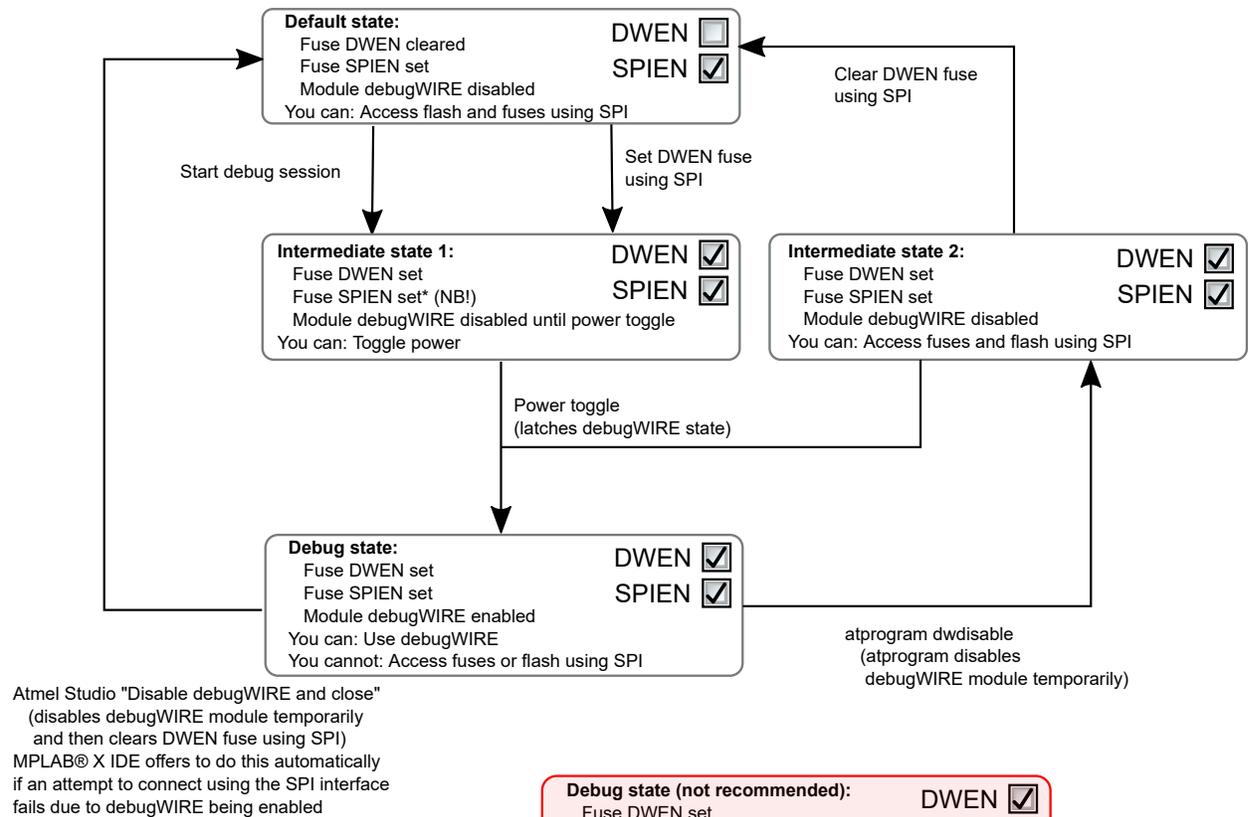
使能 debugWIRE 接口时，它将控制器件的/RESET 引脚，使其与也需要该引脚的 SPI 接口互斥。使能和禁止 debugWIRE 模块时，请遵循以下两种方法之一：

- 交给软件前端进行处理（推荐）
- 手动将 DWEN 置 1 和清零（需小心谨慎，仅限高级用户操作！）



**重要：** 手动处理 DWEN 时，必须使 SPIEN 熔丝保持置 1，以免必须使用高电压编程。

图 4-1. 了解 debugWIRE 和 DWEN 熔丝



#### 4.5.1.3. megaAVR OCD 特性

megaAVR OCD 基于 JTAG 物理接口。它具有以下特性：

- 完整程序流控制。
- 可完全访问所有寄存器和存储区。
- 四个程序存储器（硬件）断点（其中一个保留）。
- 硬件断点可组合成数据断点。
- 程序断点的数量不限（使用 BREAK）（ATmega128[A]除外）。

#### 4.5.1.4. AVR XMEGA OCD 特性

AVR XMEGA OCD 也称为 PDI（编程和调试接口）。JTAG 和 PDI 这两个物理接口访问器件内的同一个 OCD 实现。它具有以下特性：

- 完整程序流控制。
- 可完全访问所有寄存器和存储区。
- 一个专用程序地址比较器或符号断点（保留）。
- 4 个硬件比较器。
- 用户程序断点的数量不限（使用 BREAK 指令）。
- 对系统时钟频率没有限制。

#### 4.5.1.4.1. AVR XMEGA 特殊注意事项

##### OCD 和时钟

当 MCU 进入停止模式时，OCD 时钟用作 MCU 时钟。OCD 时钟为 JTAG TCK（使用 JTAG 接口时）或 PDI\_CLK（使用 PDI 接口时）。

##### 停止模式下的 I/O 模块

与早期 Microchip megaAVR 器件有所不同，XMEGA 器件中的 I/O 模块在停止模式下停止工作。这意味着，USART 发送将中断，定时器（和 PWM）将停止工作。

##### 硬件断点

硬件断点比较器共有四个——两个地址比较器和两个值比较器。这些比较器存在特定限制：

- 所有断点必须为相同类型（程序或数据）。
- 所有数据断点必须位于同一存储区（I/O、SRAM 或 XRAM）。
- 如果使用地址范围，则只能有一个断点。

可以设置以下不同组合：

- 两个单数据或程序地址断点。
- 一个数据或程序地址范围断点。
- 两个单数据地址断点（使用单值比较）。
- 一个具有地址范围和/或值范围的数据断点。

无法设置断点时，MPLAB X IDE 和 Microchip Studio 将发出通知并告知原因。如果软件断点可用，则数据断点的优先级高于程序断点。

##### JTAGEN 熔丝

JTAG 接口使用 JTAGEN 熔丝使能，该熔丝默认处于已编程状态。这样便可访问 JTAG 编程接口。



**重要：** 如果 JTAGEN 熔丝意外禁止，只能使用 PDI 物理接口重新使能。

如果 JTAGEN 熔丝已编程，仍可在固件中禁止 JTAG 接口，具体方法是将 MCU 控制寄存器中的 JTAG 禁止位置 1。这将导致代码不可调试，因此在尝试调试会话时不应该这样做。如果启动调试会话时，此代码已在 Microchip AVR 器件上执行，MPLAB PICKIT 5 会在连接时将 RESET 线置为有效。如果该 RESET 线接线正确，将强制目标 AVR 器件进入复位状态，从而实现 JTAG 连接。

如果使能 JTAG 接口，JTAG 引脚将无法用于备用引脚功能。这些引脚将保持用作专用 JTAG 引脚，直到通过以下方法禁止 JTAG 接口为止：通过程序代码将 JTAG 禁止位置 1 或通过编程接口将 JTAGEN 熔丝清零。



**提示：**

确保在 Microchip Studio 中的编程对话框和调试选项对话框中选中“use external reset”（使用外部复位）复选框，以便 MPLAB PICKIT 5 将 RESET 线置为有效，如果器件上运行的代码禁止了 JTAG 接口，还可以通过将 JTAG 禁止位置 1 来重新使能 JTAG 接口。

##### 在 ATxmegaA1 版本 H 及更早版本处于休眠模式时进行调试

ATxmegaA1 器件的早期版本存在缺陷，会在器件处于特定休眠模式下时阻止 OCD 使能。可通过以下两种变通方法重新使能 OCD：

- 进入 MPLAB PICkit 5。在 Tools（工具）菜单的选项中，使能“Always activate external Reset when reprogramming device”（重新编程器件时始终激活外部复位）。
- 执行全片擦除。

触发该缺陷的休眠模式如下：

- 掉电
- 节能
- 待机
- 扩展待机

#### 4.5.1.5. 高级调试（AVR JTAG/debugWIRE 器件）

##### I/O 外设

即使程序执行因断点停止，大多数 I/O 外设也将继续运行。例如：如果在 UART 发送操作期间到达断点，将会完成该发送操作，并且相应位将置 1。TXC（发送完成）标志将置 1，在下次单步执行代码时即可使用（在实际器件中通常稍后即会发生）。

在停止模式下，除了以下两者之外的所有 I/O 模块都将继续运行：

- 定时器/计数器（可使用软件前端进行配置）
- 看门狗定时器（始终停止工作以防止在调试期间复位）

##### 单步执行 I/O 访问

由于 I/O 在停止模式下继续运行，因此应注意避免某些时序问题。例如以下代码：

```
OUT PORTB, 0xAA
IN TEMP, PINB
```

正常运行该代码时，TEMP 寄存器不会读回 0xAA，因为该数据被 IN 操作采样时尚未被物理锁存到引脚。必须在 OUT 与 IN 指令之间放置一条 NOP 指令，以确保 PIN 寄存器中的值正确无误。

但通过 OCD 单步执行该函数时，该代码会始终将 0xAA 置于 PIN 寄存器中，因为即使内核在单步执行期间停止工作，I/O 也会全速运行。

##### 单步执行和时序

某些寄存器需要在使能控制信号后一定数量的周期内进行读/写。由于 I/O 时钟和外设在停止模式下继续全速运行，因此单步执行此类代码不能满足时序要求。在两次单步执行之间，I/O 时钟可能已运行数以百万个周期。要成功按此时序要求读/写寄存器，应按原子操作执行整个读/写序列，从而全速运行器件。要实现此目的，可以在调试环境中使用宏或函数调用来执行代码或使用运行到光标功能。

##### 访问 16 位寄存器

Microchip AVR 外设通常包含多个 16 位寄存器，这些寄存器可通过 8 位数据总线进行访问（例如 16 位定时器的 TCNTn）。16 位寄存器必须使用两次读/写操作进行字节访问。在 16 位访问过程中暂停或单步执行这种情况可能导致错误值。

##### 受限 I/O 寄存器访问

某些寄存器无法在不影响其内容的情况下进行读取。此类寄存器包括读取时会清零相应标志的寄存器或者缓冲数据寄存器（如 UDR）。软件前端将阻止在停止模式下读取这些寄存器，以保留 OCD 调试的预期非侵入特性。此外，一些寄存器无法在不产生副作用的情况下安全写入。这些寄存器是只读的。例如：

- 通过向任何位写入 1 清零标志的标志寄存器。这些寄存器是只读的。
- UDR 和 SPDR 寄存器无法在不影响模块状态的情况下进行读取。这些寄存器无法访问。

## 4.6. PIC32M MCU——片上调试

PIC32M MCU 器件支持两种类型的调试：(1)使用 PGECx 和 PGEDx 引脚的在线串行编程（ICSP）和调试；(2) 4 线 MIPS® 增强型 JTAG。

MIPS32 M4K 处理器内核为应用程序和内核代码的软件调试提供了一个增强型 JTAG（EJTAG）接口。除了标准的 JTAG 指令之外，EJTAG 规范中定义的特殊指令还指定选择的寄存器及其使用方式。有关该接口的详细信息，请参见器件数据手册。

此外，还有程序断点和复杂数据断点。有关特定 PIC32M 器件的调试功能的详细信息，请参见器件数据手册。

## 4.7. PIC MCU/dsPIC DSC——片上调试

片上调试模块是一种系统，允许开发人员从外部开发平台监视和控制器件中的执行情况，通常通过称为 *调试器* 或 *调试适配器* 的设备实现。凭借 OCD 系统，执行应用程序时可在目标系统中保持准确的电气和时序特性（与软件模拟器不同）。该系统能够依条件或手动停止执行应用程序，以检查程序流和存储器。

对于 PIC 单片机（MCU）或 dsPIC 数字信号控制器（DSC），可能需要保留一些器件资源用于调试。

### 4.7.1. 仿真器基本功能

MPLAB PICKit 5 在线调试器具有以下基本调试功能。

#### 4.7.1.1. 启动和停止仿真

要在 MPLAB X IDE 中调试应用程序，必须创建一个包含源代码的项目，以便编译代码、将代码烧写到器件中并按照如下所述执行代码：

|   |   |
|---|---|
|   | 在调试模式下调试或执行项目代码。  |
|  | 暂停代码执行。   |
|  | 暂停后继续代码执行。  |
|  | 对于暂停的代码，单步进入或执行一条指令。注意不要单步进入 Sleep 指令，否则必须执行处理器复位以恢复仿真。 |
|  | 对于暂停的代码，单步跳过一条指令。                                       |
|  | 完成调试会话，结束代码执行。  |
|  | 执行处理器复位。根据器件的不同，可能有其他复位（如 POR/BOR、MCLR 和系统复位）。          |

#### 4.7.1.2. 查看处理器存储器和文件

MPLAB X IDE 提供了多个窗口来查看调试信息和各种处理器存储器信息，可在 Window（窗口）菜单中进行选择。有关使用这些窗口的帮助信息，请参见 MPLAB X IDE 在线帮助。

- **Window > Target Memory Views**（窗口 > 目标存储器视图）——查看不同类型的器件存储器。存储器类型包括程序存储器、文件寄存器和配置存储器等，具体取决于所选的器件。
- **Window > Debugging**（窗口 > 调试）——查看调试信息。从变量、观察、调用堆栈、断点、跑表和跟踪中进行选择。

要查看源代码，在 **Projects** 窗口中找到要查看的源代码文件并双击以在 **Files**（文件）窗口中打开。该窗口中的代码根据所选处理器和编译工具采用不同颜色标识。要更改颜色标识的样式，请选择 **Tools > Options > Fonts & Colors > Syntax**（工具 > 选项 > 字体和颜色 > 语法）。

有关编辑器的更多信息，请参见 MPLAB X IDE 在线帮助的 Editor（编辑器）部分。

### 4.7.1.3. 使用断点

使用断点在代码中的指定行暂停代码执行。

#### 4.7.1.3.1. 断点资源

对于 16 位 PIC 和 dsPIC DSC 器件，断点、数据捕捉和运行时观察使用相同的资源。因此，可用的断点数实际上是可用的断点和触发的总数。

对于 32 位 PIC32M 器件，断点使用的资源与数据捕捉和运行时观察不同。因此，可用的断点数与可用的触发数无关。

可用和/或已用的硬件和软件断点数在 **Dashboard**（仪表板）（**Window > Dashboard**（窗口 > 仪表板））窗口中显示。有关该功能的更多信息，请参见 MPLAB X IDE 文档。并非所有器件都有软件断点。

有关断点操作的限制（包括每个器件的硬件断点的常规数目以及硬件断点的断点滑移（skidding）大小），请参见 [调试限制——PIC® MCU](#)。

#### 4.7.1.3.2. 硬件或软件断点选择

要选择硬件或软件断点：

1. 在 **Projects** 窗口中选择项目，然后单击右键以选择 **Properties**。
2. 在 **Project Properties** 中，选择 **ICE4 > Debug Options**（ICE4 > 调试选项）。
3. 选中 **Use software breakpoints**（使用软件断点）以使用软件断点。取消选中则使用硬件断点。

**注：**使用软件断点进行调试会影响器件的耐用性。因此，建议不要将以这种方式使用过的器件用作生产器件。

为帮助您确定使用哪类断点（硬件或软件），下表对每种断点的特性进行了比较。

表 4-4. 硬件断点与软件断点

| 特性                        | 硬件断点  | 软件断点                  |
|---------------------------|---|-----------------------|
| 断点数                       | 有限  | 无限                    |
| 断点写入到*                    | 内部调试寄存器   | 闪存程序存储器               |
| 断点应用于**                   | 程序存储器/数据存储器                                       | 仅程序存储器                |
| 设置断点的时间                   | 极短  | 取决于振荡器速度、编程闪存的时间以及页大小 |
| 断点滑移（skidding）            | 大多数器件。有关详细信息，请参见在线帮助中的 <b>Limitations</b> （限制）部分。 | 无                     |
| *器件中写入断点信息的位置。            |   |                       |
| **适用于断点的器件功能类型。这是设置断点的位置。 |   |                       |

#### 4.7.1.4. 使用跑表

使用跑表来确定两个断点之间的时间。

**注：**跑表使用断点资源。

**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**（窗口 > 调试 > 跑表）。单击窗口左侧的 **Properties** 图标，并选择启动和停止断点。
4. 再次调试程序，以获取跑表计时结果。

图 4-2. 跑表设置

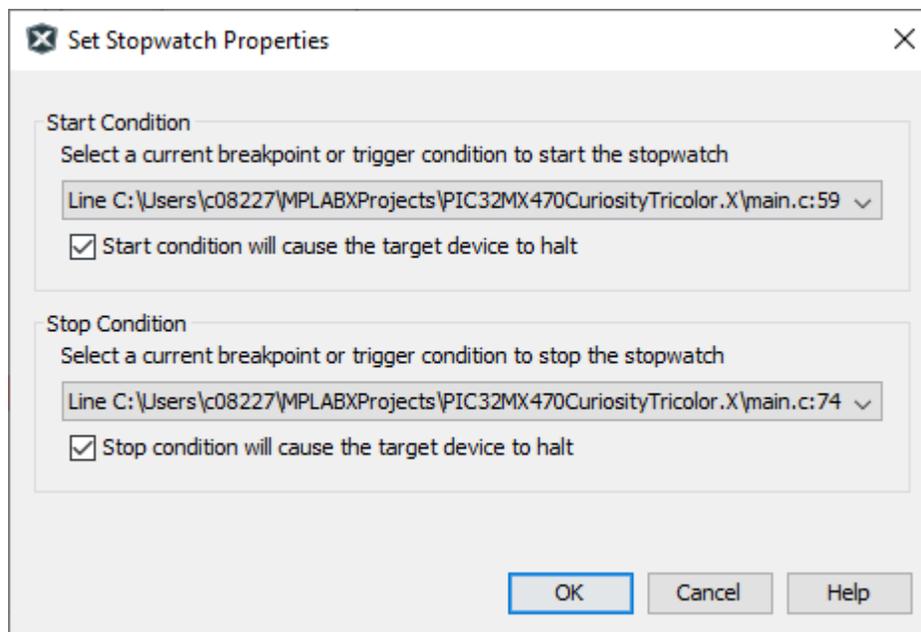
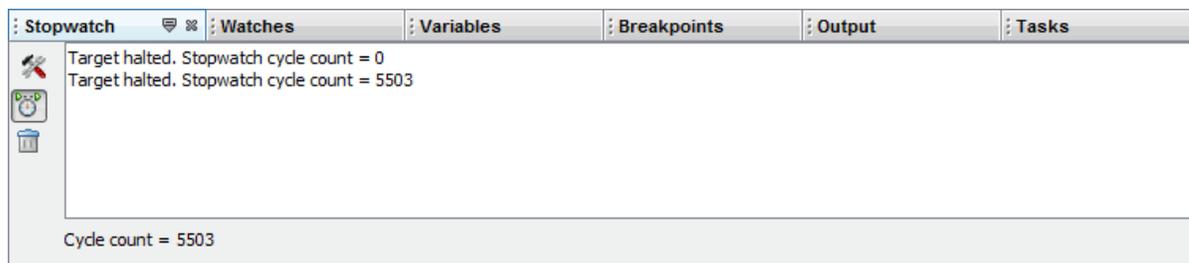


图 4-3. 具有内容的 Stopwatch（跑表）窗口



跑表在窗口左侧具有以下图标：

表 4-5. 跑表图标

| 图标 | 图标文本                             | 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，选择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。                        |

#### 4.7.1.5. 设置冻结外设

选择 **Freeze on Halt**（暂停时冻结）可以在暂停时冻结所选外设。有关这些功能的更多信息，请参见[冻结外设](#)。

## 4.7.2. ICSP 调试

使用 MPLAB® PICKit™ 5 在线调试器作为调试器有两个步骤。第一步要求将应用程序烧写到目标器件中（可使用 MPLAB PICKit 5 来进行）。第二步是使用目标闪存器件的内部在线调试硬件来运行和测试应用程序。这两步与 MPLAB X IDE 操作直接相关：

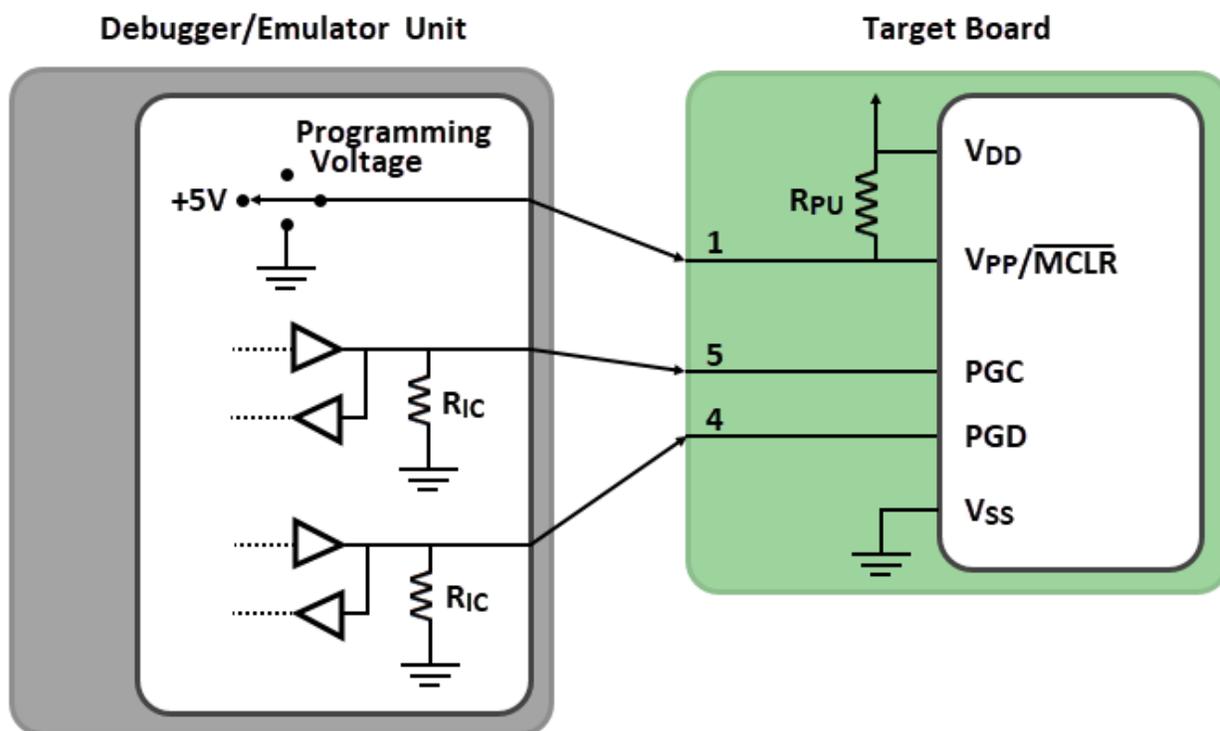
1. 将代码烧写到目标器件并激活具体的调试功能（有关详细信息，请参见下一节）。
2. 使用调试器设置断点并运行。

有关更多信息，请参见 MPLAB X IDE 在线帮助。

如果目标器件不能被正确编程，MPLAB PICKit 5 将无法进行调试。

下面给出了 MPLAB PICKit 5 的部分内部接口电路的简化图。

图 4-4. 正确的 ICSP™ 编程连接



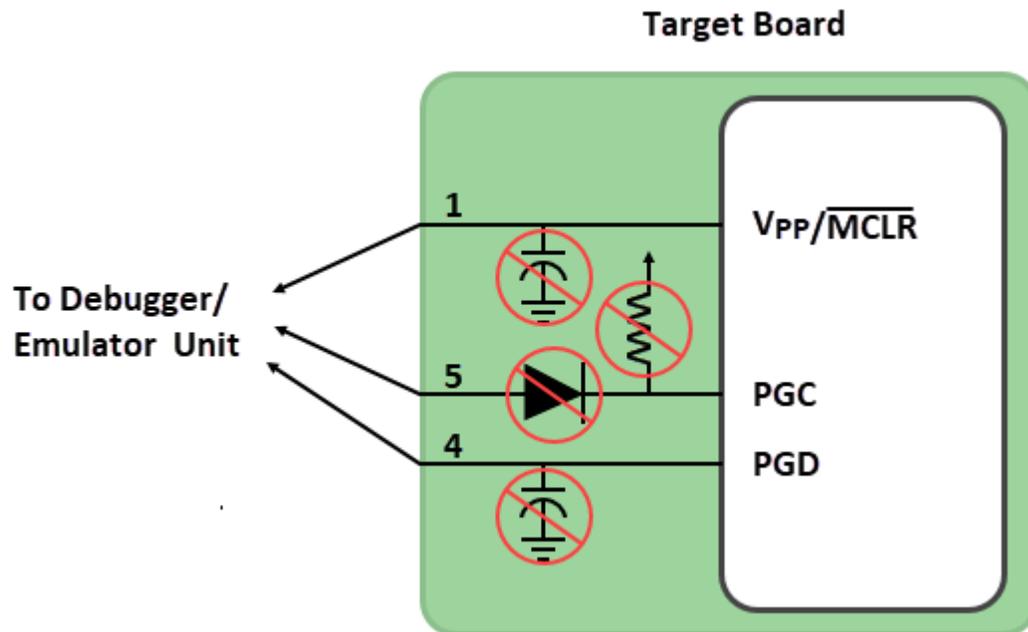
其中  $R_{ic} = 4.7\text{ k}\Omega$ ， $R_{pu} = 10\text{ k}\Omega$ （典型值）

编程时，目标器件不需要时钟，但必须为其提供电源。编程时，调试器将编程电压置于  $V_{PP}/\overline{MCLR}$  上，通过 PGC 发送时钟脉冲，通过 PGD 发送串行数据。要校验器件是否已被正确编程，可以向 PGC 发送时钟，并通过 PGD 读回数据。此操作可确认调试器与器件之间是否正确通信。

### 4.7.2.1. 阻止仿真器正常工作的电路

下图给出了调试器的三根有效线上会阻止 MPLAB PICKit 5 在线调试器正常工作的一些元件。

图 4-5. 不正确的电路元件



尤其是必须遵循以下准则：

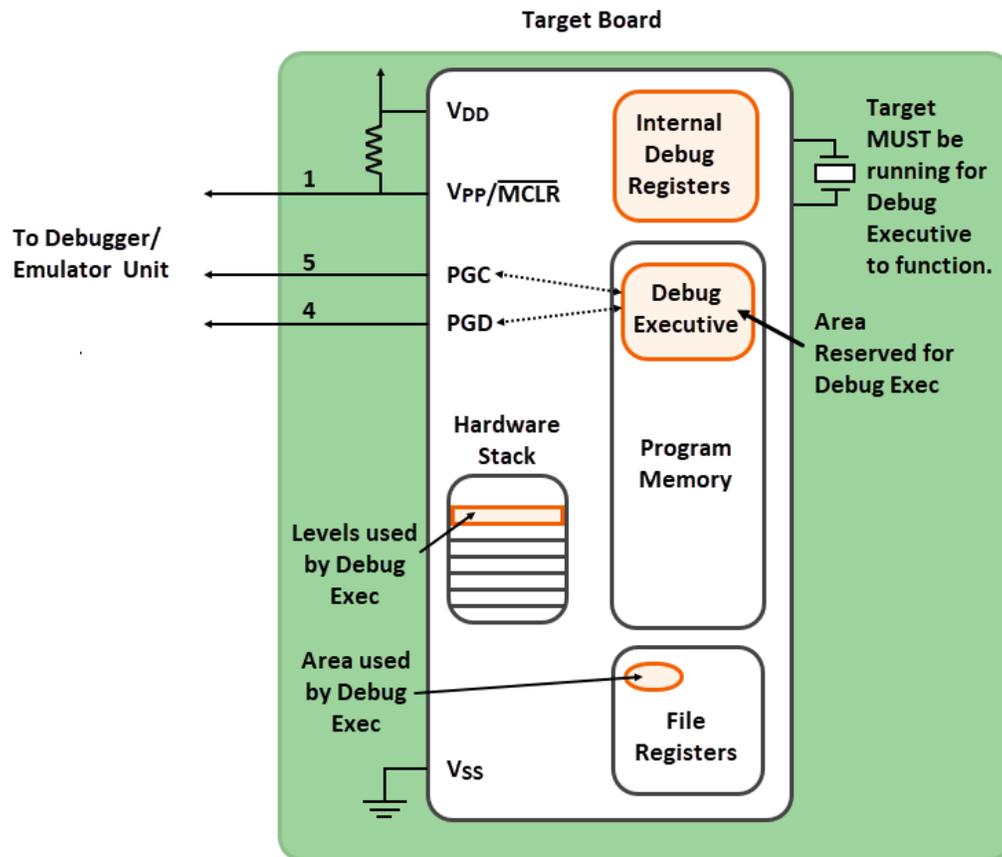
- 请勿在 PGC/PGD 上使用上拉——它们可能会破坏电压。
- 请勿在 PGC/PGD 上使用电容——它们将阻止数据线和时钟线在编程和调试通信期间快速变化并且会延长编程时间。
- 请勿在 MCLR 上使用电容——它们将阻止 VPP 快速变化。通常一个简单的上拉电阻就已足够。
- 请勿在 PGC/PGD 上使用二极管——它们将阻止调试器和目标器件之间进行双向通信。

#### 4.7.2.2. 进入调试模式的操作序列

假定满足[调试要求](#)，将 MPLAB PICKit 5 在线调试器设置为 MPLAB X IDE 中的当前工具。右键单击项目名称并选择 **Properties** 以打开 **Project Properties** 对话框，然后在 **Connected Hardware Tool** 下选择 **MPLAB PICKit 5**。如果连接了多个工具，请根据工具名称旁的序列号进行选择。

现在可以执行以下操作：

- 选择 **Debug > Debug Project**（调试 > 调试项目）时，应用程序代码将通过 ICSP 协议编程到器件的存储器中，如本章开头所述。
- 一个小的“调试执行程序”将加载到目标器件的存储器中。由于某些架构要求调试执行程序必须驻留在程序存储器中，因此应用程序不能使用此保留的空间。有些器件为调试执行程序提供了专门的存储区。有关详细信息，请参见器件数据手册。
- 目标器件中的特殊“在线调试”寄存器通过 MPLAB X IDE 使能。这些寄存器允许调试器激活调试执行程序。有关器件保留资源的更多信息，请参见[调试器使用的资源](#)。
- 目标器件在调试模式下运行。为确保目标器件正常运行，请查看[无法调试的首要原因](#)，例如振荡器问题。

图 4-6. PIC<sup>®</sup> MCU 使用保留资源进行调试

### 4.7.2.3. 调试详细信息

在确保 MPLAB PICKit 5 与目标板正确连接并排除错误电路后，调试器即可开始调试。

为判断应用程序能否正确运行，通常可在程序代码中预先设置断点。当通过 MPLAB X IDE 的用户界面设置断点时，断点的地址存储在目标器件的特殊内部调试寄存器中。PGC 和 PGD 上的命令直接传送到这些寄存器来设置断点地址。

接下来，通常在 MPLAB X IDE 中选择 Debug  按钮。然后，调试器通知调试执行程序运行。目标器件将从复位向量开始执行，直到程序计数器到达先前存储在内部调试寄存器中的断点地址。

执行断点地址处的指令后，目标器件的在线调试机制会“启动”，并将器件的程序计数器传递给调试执行程序（类似于中断），用户的应用程序将暂停执行。调试器通过 PGC 和 PGD 与调试执行程序通信，获取断点状态信息，并将这些信息发送回 MPLAB X IDE。随后，MPLAB X IDE 向调试器发送一系列查询，以获取关于目标器件的信息，例如文件寄存器内容和 CPU 状态。这些查询由调试执行程序执行。

调试执行程序像程序存储器中的应用程序一样运行。它使用堆栈中的某些地址单元存储临时变量。如果器件没有运行，无论出于何种原因（没有振荡器、电源连接故障和目标板短路等），调试执行程序都将无法与 MPLAB PICKit 5 通信，并且 MPLAB X IDE 将发出一条错误消息。

设置断点的另一种方法是选择 Pause（暂停） 按钮。这会翻转 PGC 和 PGD 线，从而使目标器件的在线调试机制将程序计数器从程序存储器中的用户代码切换到调试执行程序。同样，目标应用程序暂停，MPLAB X IDE 通过调试器与调试执行程序的通信来查询目标器件的状态。

#### 4.7.2.4. 调试要求

要使用 MPLAB PICkit 5 在线调试器系统进行调试（设置断点和查看寄存器等），以下关键因素必须正确：

- 调试器必须上电，必须连接到计算机并且必须与 MPLAB X IDE 软件通信。
- 目标器件必须通电且具有正常运行的振荡器。如果目标器件因任何原因无法运行，MPLAB PICkit 5 在线调试器将无法进行调试。
- 必须正确编程目标器件的配置字。这些配置字可使用代码或 MPLAB X IDE 中的 Configuration Bits（配置位）窗口来设置。
  - 振荡器配置位应对应于目标板上的振荡器类型。
  - 对于某些器件，看门狗定时器默认使能，需要禁止。
  - 不要使能目标器件的任何类型的代码保护功能。
  - 不要使能目标器件的表读保护功能。
- 对于具有多个 PGC/PGD 引脚对的某些器件，需要在器件的配置字设置中选择正确的引脚对。这一点仅针对调试，因为编程可通过任何 PGC/PGD 引脚对工作。

#### 4.7.2.5. 调试器使用的资源

对于一些器件，必须使用器件资源进行调试。如需器件的调试器所使用资源的完整列表，请在 MPLAB X IDE 中选择 **Help > Release Notes**。除了“Release Notes/Readmes”（发行说明/自述文件）部分以外，还有“Reserved Resources”（保留资源）部分。选择“Reserved Resources by Device Family and Tool”（按器件系列和工具保留的资源）或“Reserved Resources by Device for All Tools”（按器件为所有工具保留的资源）。

#### 4.7.2.6. 编程

进行调试时，将 MPLAB PICkit 5 在线调试器设置为 MPLAB X IDE 中的当前工具。右键单击项目名称并选择 **Properties** 以打开 **Project Properties** 对话框，然后在 **Connected Hardware Tool** 下选择 **PICkit 5**。如果连接了多个工具，请根据工具名称旁的序列号进行选择。



- 选择 **Run Project**（运行项目）图标（）。应用程序代码通过 ICSP 协议编程到器件的存储器中。编程时不需要时钟，可以编程所有的处理器模式，包括代码保护、使能看门狗定时器和表读保护。
- 对于某些目标器件，可以将一个小的“编程执行程序”加载到程序存储器的高地址区域中。
- 目标器件中的特殊“在线调试”寄存器以及所有调试功能均通过 MPLAB X IDE 禁止。这意味着不能设置断点，不能查看或更改寄存器内容。
- 目标器件运行在发布模式下。作为编程器，调试器只能通过翻转 MCLR 线来复位和启动目标器件。

## 5. 脱机编程

MPLAB PICKit 5 脱机编程 (PTG) 功能可将器件存储器映像下载到插入 MPLAB PICKit 5 工具的 microSDHC 卡中, 以便之后编程到特定器件中。该映像包含编程算法信息。MPLAB PICKit 5 编程器设置为脱机编程后, 不需要软件和 PC 即可编程器件; 但是, 可使用 BLE 智能手机或平板电脑上的 PTG 应用程序作为控制台来查看 microSDHC 卡上的映像文件并触发操作。可以使用 MPLAB X IDE (见[使用 MPLAB X IDE 设置 PTG 模式](#)) 或 MPLAB IPE (见[使用 MPLAB IPE 设置 PTG 模式](#)) 将 MPLAB PICKit 5 设置为脱机编程模式。

### 注:

在脱机编程模式下, 不能使用任何调试功能。

图 5-1. MPLAB<sup>®</sup> PICKit<sup>™</sup> 5 在线调试器图



### 脱机编程的准备工作

- 必须在计算机上安装 MPLAB X IDE 或 MPLAB IPE (v6.10 或更高版本)。
- MicroSDHC 卡——必须将格式化的 FAT32 兼容 microSDHC 卡正确插入 MPLAB PICKit 5 工具中以使用脱机编程功能。
- 可选择在 BLE 设备上安装 PTG 应用程序, 以控制 PTG 操作并选择要编程的存储器映像。
- 电源——用于将 MPLAB PICKit 5 设置为脱机编程模式以及远程编程器件。有关具体电源要求, 请参见以下章节。

## 相关信息

使用 BLE 设备上的 PTG 应用程序

### 5.1. 脱机编程的电源要求

#### 连接到计算机时

当需要将 MPLAB PICKit 5 连接到计算机（以及 MPLAB X IDE 或 MPLAB IPE 应用程序）时，通过计算机与工具顶部的 USB Type-C® 连接器之间连接的 USB 线缆供电。

#### 使用 PTG 编程器件时

将 MPLAB PICKit 5 连接到远程目标板以使用脱机编程功能对器件进行编程时，目标板可为调试器供电。目标板至少需要为 MPLAB PICKit 5 提供 350 mA 的电流。

如果目标板无法提供足够的电流，则必须通过 MPLAB PICKit 5 工具顶部的 USB Type-C 连接器使用 5V 电源为 MPLAB PICKit 5 供电。可通过以下方法来提供电源：

- 任何可用的 PC USB 端口或 USB 集线器端口（无需 USB 通信；仅用于提供电源）。
- 便携式设备的 USB 主机端口。
- USB 电源适配器或带 USB Type-C 连接器的充电器，连接到汽车电源插孔或交流电源插座。
- 便携式电池充电器或电源，用于手机或其他带 USB Type-C 连接器的便携式设备。
- 为 MPLAB PICKit 5 USB Type-C 连接器提供 5V 稳压电源的定制电池组。

所用 USB 电源应满足以下最低标准：

- 能够为 MPLAB PICKit 5 工具提供最小 350 mA 的电流。
- 提供稳定的 4.5V-5.5V 稳压输出。

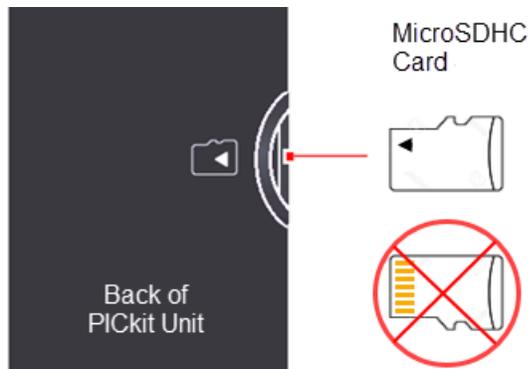
#### 注：

1. 大多数自带电池的便携式充电器/电源设备在其内部电池电压变低、输出跌落到 4.5V 以下时，不会提供指示信息。因此，必须确保设备电池有足够的剩余容量为 MPLAB PICKit 5 提供 4.5V 以上的电源。
2. 在不使用时，任何电池电源都应 与 MPLAB PICKit 5 设备断开连接。否则 MPLAB PICKit 5 设备将耗尽电源电池。

### 5.2. 脱机编程的限制

1. 必须将格式化的 FAT32 兼容 microSDHC 卡正确插入 MPLAB PICKit 5 中以使用脱机编程功能。

图 5-2. 正确插入 MicroSDHC 卡



2. 支持的器件：有关器件支持信息，请参见 MPLAB PICKit 5 发行说明（自述文件）。

3. 在脱机编程模式下，如果目标板正在为 PICkit 5 供电，则它必须能够提供 350 mA 的电流，PICkit 5 工具才能正常运行。如果目标板不能提供足够的电流，则需通过连接电源、计算机或 USB 移动电源的 PICkit 5 USB 端口直接为 PICkit 5 供电。有关建议的电源，请参见[脱机编程的电源要求](#)。
4. 如果不需要高电压（HV）编程，建议在器件支持的情况下使用低电压编程（Low Voltage Programming, LVP）。

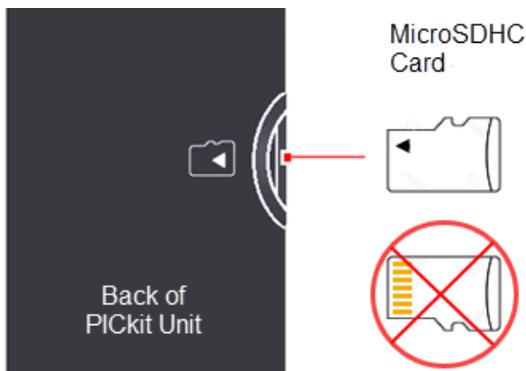
## 5.3. 将 PICkit 5 设置为脱机编程模式

在将存储器映像下载到 MPLAB PICkit 5 进行 PTG 操作之前，应针对脱机编程操作设置 PICkit 5 编程器选项。实际上，强烈建议在下载映像以进行脱机编程之前，首先使用所需选项通过软件编程目标器件进行测试，确保按预期编程器件。之后可以将 PICkit 5 置于 PTG 模式。有关说明，请参见以下章节。

### 5.3.1. 使用 MPLAB X IDE 设置 PTG 模式

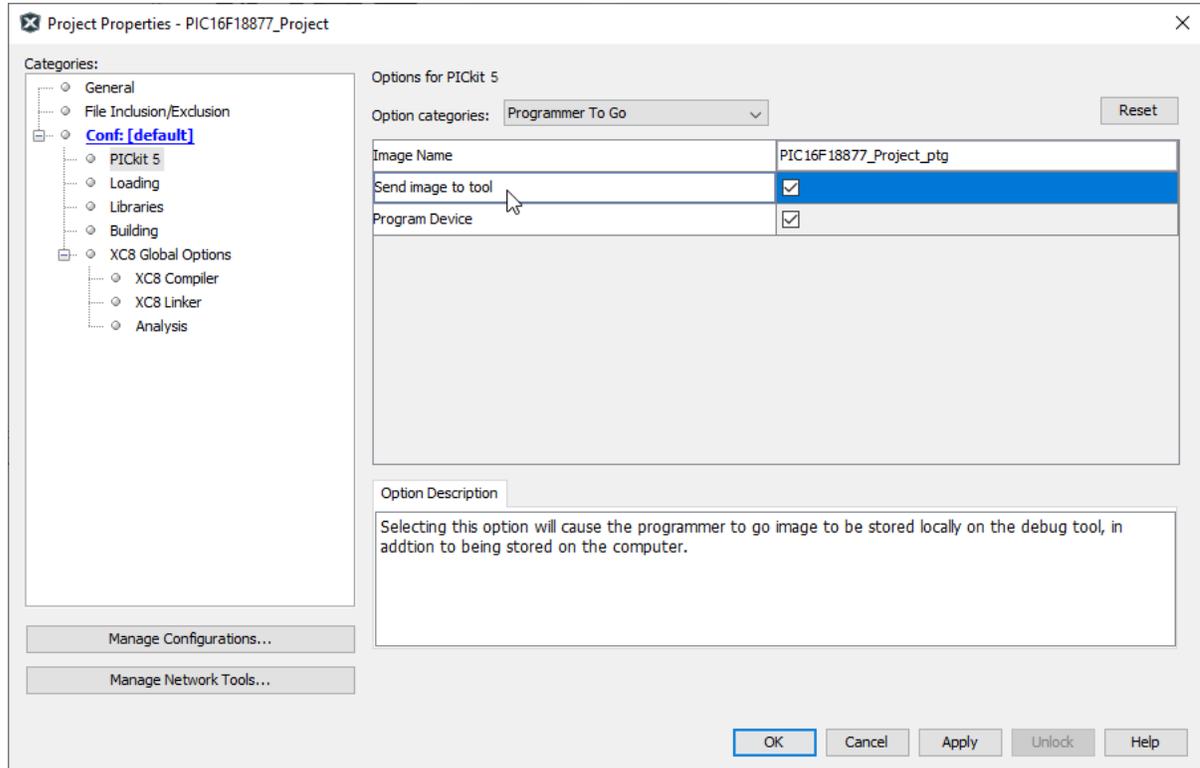
使用 MPLAB X IDE，按照以下步骤将项目文件下载到 MPLAB PICkit 5 中插入的 microSDHC 卡并进入脱机编程模式。

1. 将格式化的 FAT32 兼容 microSDHC 卡插入 PICkit 5。



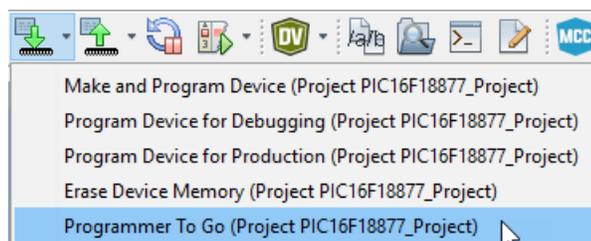
2. 确保已正确连接要进行脱机编程的器件：
  - 通过 USB 线缆将 PICkit 5 连接到计算机。
  - 通过相应的编程接口连接器将 PICkit 5 连接到目标板。
  - 目标板由 PICkit 5 或电源供电，具体取决于 **Project Properties** 中的选择。
3. 在 MPLAB X IDE 中，打开要使用的项目，然后选择 PICkit 5 工具进行编程。
4. 右键单击您的项目名称，打开 **Project Properties** 对话框。然后单击 **Categories**（类别）下的 **PICkit 5**，随即将在屏幕右侧显示 **Options for PICkit 5**（PICkit 5 的选项）。选择 **Programmer-To-Go**（脱机编程）选项类别。

图 5-3. MPLAB® X IDE Programmer-To-Go 选项



5. 在 **Image Name**（映像名称）字段中，默认设置为“<your project name>\_ptg”，但也可根据需要编辑名称。这将是 microSDHC 卡上的文件夹名称，其中包含脱机编程的相关文件。
6. **Send image to tool**（将映像发送到工具）复选框默认处于选中状态。选中该复选框后，将创建 PTG 映像，然后将其发送到所连接的 MPLAB PICKit 5 中插入的 microSDHC 卡。
7. **Program Device**（编程器件）复选框默认处于选中状态。选中该复选框后，将对连接到 MPLAB PICKit 5 的器件进行编程。  
注：如果 *Send image to tool* 和 *Program Device* 复选框均未选中，请参见[在没有存储卡的情况下设置 PTG 模式](#)。
8. 单击 **Apply**（应用），然后单击 **OK**（确定）。使用工具栏上的 **Make and Program Device Main Project**（编译和编程器件主项目）图标并选择 **Programmer-To-Go Project Name**（项目名称）。

图 5-4. MPLAB® X IDE——将映像下载到 MicroSDHC 卡



在此过程中，将对器件进行编程，然后在 microSDHC 卡中装入脱机编程操作的相关文件。当此过程成功完成时，**Output** 窗口将显示状态消息“Programming/Verify complete”（编程/校验完成）。

注：microSDHC 卡上的 PTG 设置与项目中的设置相同（例如，存储器和电源等）。

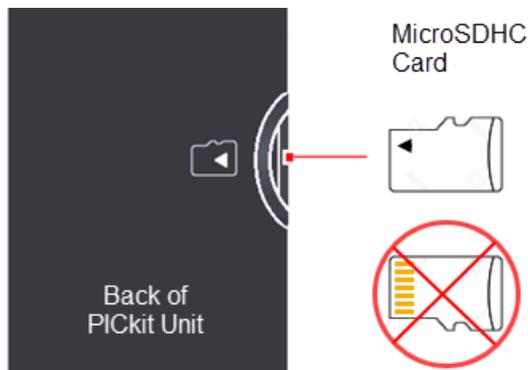
**PICKit 5** 现在处于脱机编程模式。LED 应呈绿色闪烁状态，表示已成功针对脱机编程配置了工具。

9. 将 PICkit 5 断开连接，现在即可使用脱机编程。

### 5.3.2. 使用 MPLAB IPE 设置 PTG 模式

使用 MPLAB IPE，按照以下步骤将项目文件下载到 MPLAB PICkit 5 中插入的 microSDHC 卡并进入脱机编程模式。

1. 将格式化的 FAT32 兼容 microSDHC 卡插入 PICkit 5。

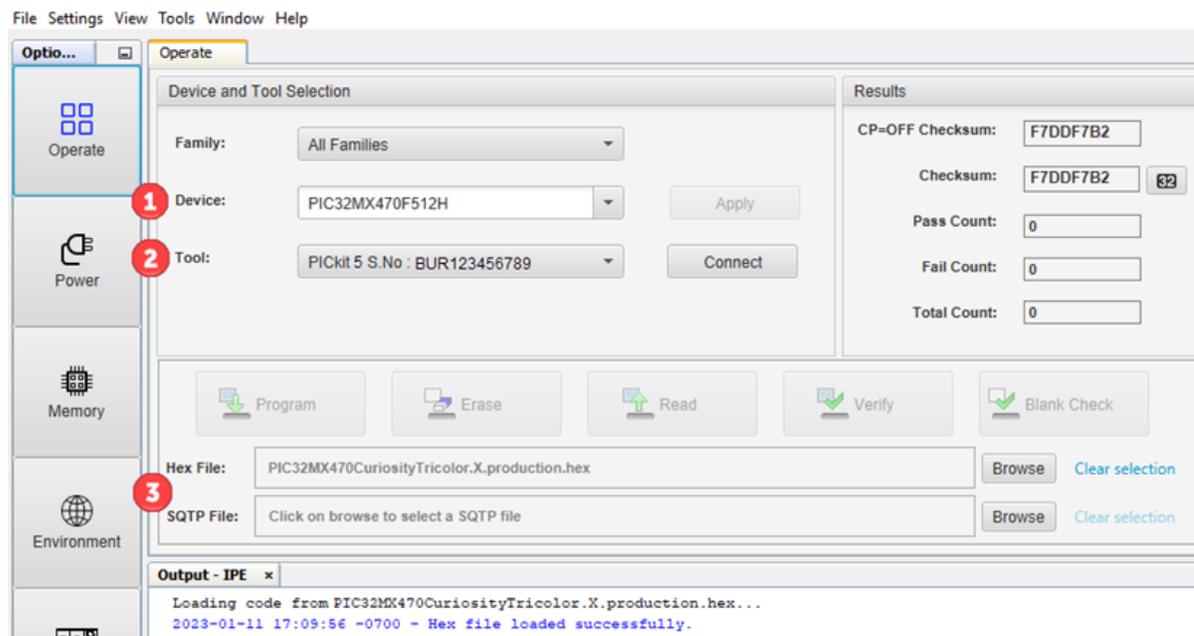


2. 确保已正确连接要进行脱机编程的器件：

- 通过 USB 线缆将 PICkit 5 连接到计算机。
- 通过相应的编程接口连接器将 PICkit 5 连接到目标板。
- 目标板由 PICkit 5 或电源供电，具体取决于 **Project Properties** 中的选择。

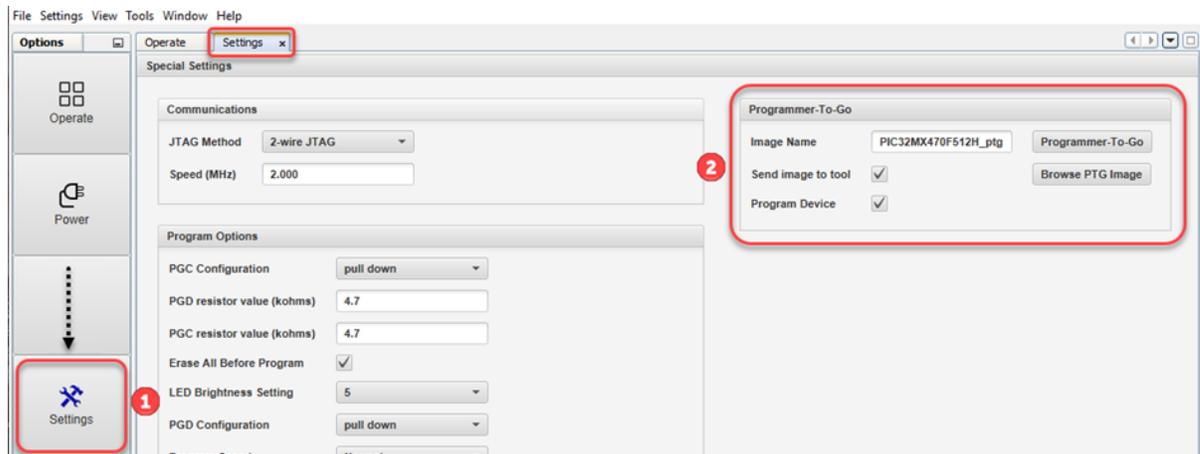
3. 在 MPLAB IPE 菜单中，选择 **Settings > Advance Mode**（设置 > 高级模式），然后输入登录密码。此时应显示/已选中 **Operate**（操作）选项卡；如果未选中，则单击 **Operate** 按钮。然后，按照下图所示，(1)输入器件名称并单击 **Apply**；(2)选择调试工具并单击 **Connect**（连接）；(3)通过浏览选择 Hex 文件或 SQTP 文件。

图 5-5. MPLAB® IPE 操作设置



4. 接下来，按照下图所示，(1)单击 **Settings**；(2)找到 **Programmer-To-Go** 框。

图 5-6. MPLAB® IPE Programmer-To-Go 选项



5. 在 **Image Name** 字段中，默认设置为“<your project name>\_ptg”，但也可根据需要编辑名称。这将是 microSDHC 卡上的文件夹名称，其中包含脱机编程的相关文件。
6. **Send image to tool** 复选框默认处于选中状态。选中该复选框后，将创建 PTG 映像，然后将其发送到所连接的 MPLAB PICKit 5 中插入的 microSDHC 卡。
7. **Program Device** 复选框默认处于选中状态。选中该复选框后，将对连接到 MPLAB PICKit 5 的器件进行编程。  
注：如果 **Send image to tool** 和 **Program Device** 复选框均未选中，请参见[在没有存储卡的情况下设置 PTG 模式](#)。

8. 单击 **Programmer-To-Go** 按钮。

在此过程中，将对器件进行编程，然后在 microSDHC 卡的脱机编程目录中装入脱机编程操作的相关文件。当此过程成功完成时，**Output** 窗口将显示状态消息“Programming/Verify complete”。

注：microSDHC 卡上的 PTG 设置与项目中的设置相同（例如，存储器和电源等）。

**PICKit 5 现在处于脱机编程模式。** LED 应呈绿色闪烁状态，表示已成功针对脱机编程配置了工具。

9. 将 PICKit 5 断开连接，现在即可使用脱机编程。

### 5.3.3. 在没有存储卡的情况下设置 PTG 模式

如果您在设置 PTG 模式时不想发送映像或对器件进行编程，则无需在 PICKit 5 中插入 microSDHC 卡也可将其置于 PTG 模式。此方法可将 PICKit 5 置于 PTG 模式，并假定在使用 PTG 对器件进行编程之前，所需的映像已装入将插入 PICKit 5 中的 microSDHC 卡上。

有些情况下您可能希望这样做，例如，用于对器件进行编程的文件压缩为 zip 文件，并发送到不同的设备（可将其解压缩并下载到 microSDHC 卡，然后将该卡插入已处于 PTG 模式的 PICKit 5 中）。或者，您可能会遇到以下情况：将 PICKit 5 的多个实例置于 PTG 模式下，并使用用于各种器件的 microSDHC 卡对这些器件进行编程。这些情况下，可能会想要在未安装存储卡的情况下设置 PTG 模式。

请参见[使用 MPLAB X IDE 设置 PTG 模式](#)或[使用 MPLAB IPE 设置 PTG 模式](#)中的说明，但不要选中 **Send Image to Tool** 和 **Program Device** 复选框。继续按照上述章节提供的说明进行操作，将 PICKit 5 置于 PTG 模式。

请记住，要使用 PTG，必须将装有所需映像的 microSDHC 卡插入 PICKit 5 中。

## 5.4. 使用脱机编程

当您准备好使用 MPLAB PICKit 5 在 PTG 模式下开始对器件进行编程时，请完成以下步骤：

1. 将已插入 microSDHC 卡的 PICKit 5 工具连接到装有项目中指定器件的目标板。

2. 确保已正确连接要进行脱机编程的器件：
  - 通过目标板上相应的编程接口连接器来连接 PICKit 5。确保目标板上的引脚 1 对准 PICKit 5 上的引脚 1 指示标记。
  - 目标板由 PICKit 5 或电源供电，具体取决于 **Project Properties** 中的选择。

**注：**在脱机编程模式下，如果目标板正在为 MPLAB PICKit 5 供电，则它必须能够提供 350 mA 的电流，PICKit 5 工具才能正常运行。

3. 当 PICKit 5 LED 变为绿色闪烁状态时，即可进行脱机编程。如果 microSDHC 卡上没有映像，则 MPLAB PICKit 5 工具呈红色闪烁状态，表示存在 PTG 错误。
4. 要开始对器件进行编程，可执行以下任一操作：
  - 用力按下（但不按住）**PICKit 5 盾形（徽标）的中心**（位于工具正面）。
  - 使用 BLE 设备上的 PTG 应用程序选择映像并执行编程。

工具检查器件 ID 后，LED 将变为紫色闪烁状态，指示正在对器件进行编程。

5. 编程完成后，LED 将变回绿色闪烁状态，指示编程/校验操作成功。现在可以进行下一项编程操作。

**注：**长按 PICKit 5 按钮可重新初始化工具。该方法可用于在检测到错误后重新初始化 PICKit 5。如果要更换为另一个包含不同脱机编程映像的 microSDHC 卡，也可以采用这种方法。

### LED 状态序列

当 PICKit 5 处于脱机编程模式并正确连接到目标板时，会发生以下序列：

| 状态     | 含义  |
|--------|---|
| 黄色快速闪烁 | 初始化电源设置。  |
| 绿色闪烁   | PICKit™ 5 准备好进行编程或编程成功完成。   |
| 紫色闪烁   | 正在进行编程。   |
| 红色闪烁   | 初始化期间的错误： <ul style="list-style-type: none"> <li>• 未检测到 microSDHC 卡。</li> <li>• microSDHC 卡格式不受支持。当选中 <b>Send image to tool</b> 时，MPLAB X IDE 或 MPLAB IPE 应会报告该错误。</li> <li>• microSDHC 卡上未找到初始化文件。检查 microSDHC 卡中是否存在脱机编程映像。</li> </ul> 编程期间的错误： <ul style="list-style-type: none"> <li>• 电源设置不正确。例如，工具正在供电，但它检测到来自目标板的 VDD。或者，本来应该由目标板供电，但 PICKit 5 工具未检测到来自目标板的 VDD，则 LED 将呈红色闪烁状态，表示发生意外事件。</li> <li>• 存储器校验错误。</li> <li>• 器件 ID 不匹配。</li> <li>• 用于编程的数据文件未找到或已损坏。</li> </ul> |

### 相关信息

使用 BLE 设备上的 PTG 应用程序

## 5.5. 退出脱机编程模式

要退出脱机编程模式，请将 MPLAB PICKit 5 设备插入到 PC USB 端口并连接到 MPLAB X IDE 或 MPLAB IPE。启动任何非 PTG 操作（例如，编程和擦除等），将显示以下消息：

“The PICKit 5 is currently in programmer to go mode.If you continue with this operation, the PICKit 5 will exit programmer to go mode.Would you like to continue?”（PICKit 5 现在处于脱机编程模式。如果继续执行此操作，则 PICKit 5 将退出脱机编程模式。是否要继续？）

选择 **Yes**（是），退出脱机编程模式。

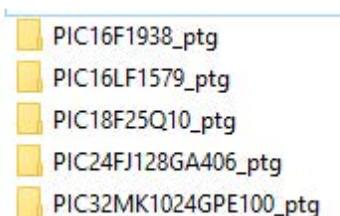
## 5.6. 使用 BLE 设备上的 PTG 应用程序

**注：** 确保 PICkit 5 已使能蓝牙无线通信功能（见 [Bluetooth Options](#)（蓝牙选项）一节）。

使用运行 PTG 应用程序的 Android/iOS 智能手机或平板电脑，可以通过蓝牙连接到 PICkit 5，并且可通过用户友好的图形界面：

- 浏览 SD 卡并获取其中包含的存储器映像的列表。
- 选择并加载所需映像。
- 触发脱机编程（PTG）操作。

图 5-7. SD 卡上的映像列表



应用程序控制台将提供相应的选项，用以查看正在进行的 PTG 操作的状态和操作期间工具传出的编程消息。

### 5.6.1. 打开脱机编程应用程序

使用 MPLAB 脱机编程（PTG）应用程序，可以通过蓝牙无线通信连接到 PICkit 5，从而远程编程下载好的器件存储器映像（该映像存储在插入 PICkit 5 设备的 microSDHC 卡上）。MPLAB PTG 应用程序可下载到 Android/iOS 智能手机和平板电脑上。

可在设备上通过以下途径下载 MPLAB PTG：

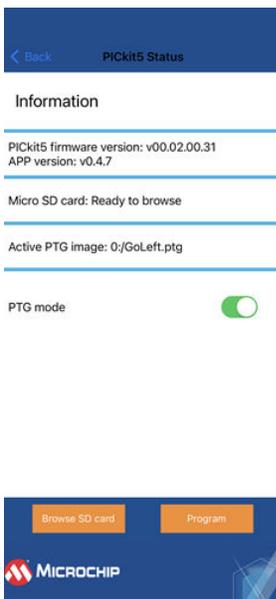
- Google Play（Android 操作系统）
- App Store（iOS）



在设备上安装该应用程序后，请确保设备已启用蓝牙无线技术，以便连接到目标 PICKit 5 设备（见 [Bluetooth Options（蓝牙选项）](#)）。若要打开该应用程序，可从设备的应用程序抽屉或主页中选择它。



打开 PTG 应用程序后，将显示您的设备周围所有可用的 PICKit 5 设备列表。可用设备将通过序列号进行标识。若要选择您的目标设备，可在屏幕上轻击它。

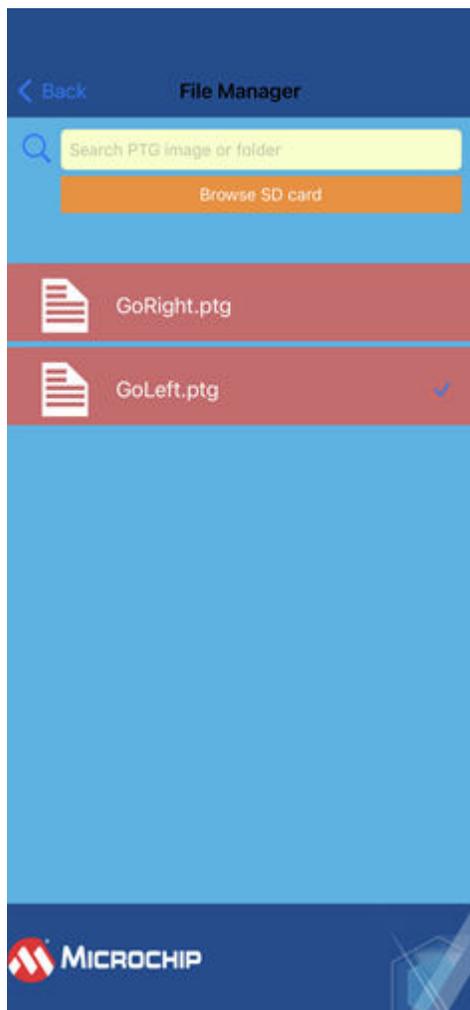


选择目标设备后，系统将显示 PICKit 5 Status（PICKit 5 状态）菜单。在这里，可以查看 PICKit 5 设备的状态：

- **PTG Mode**（PTG 模式）：显示和切换当前的脱机编程模式。
- **MicroSD card**（MicroSD 卡）：显示 SD 卡的当前状态。
- **Active PTG image**（激活的 PTG 映像）：显示当前激活的存储器映像的文件路径和名称。
- **Browse SD card**（浏览 SD 卡）：用于浏览 MicroSD 卡上所有可用的存储器映像。
- **Program**（编程）：用于打开操作菜单，以便编程激活的映像或者查看日志和统计信息。

### 5.6.2. 映像选择

打开 MPLAB 脱机编程（PTG）应用程序并选择目标 PICKit 5 设备后，导航至 **Browse SD card** 选项。选择应用程序的该选项后将显示所有兼容的存储器映像文件，这些文件远程存储在所连接的 PICKit 5 设备内插入的 microSDHC 卡上。



可以滚动浏览显示的列表，或者直接搜索相关文件名。通过轻击选择所需的存储器映像文件。

### 5.6.3. 对目标 PICKit 5 器件进行编程

通过 File Manager（文件管理器）选择一个兼容的存储器映像文件后，将显示 Operation 菜单。在该菜单中，可以选择 **Program** 选项来立即对目标 PICKit 5 器件进行编程。



## 6. 疑难解答

如果 MPLAB PICKIT 5 在线调试器在工作过程中出现问题，请从此处开始故障诊断。

### 6.1. 要首先回答的一些问题

1. 使用的是哪款器件？要支持新的器件，通常需要将 MPLAB<sup>®</sup> X IDE 或 MPLAB IPE 升级到较新的版本。
2. 您使用的是 **Microchip** 的演示板还是您自己设计的电路板？您是否遵循了用于通信连接的电阻/电容的准则？请参见“工作原理”一章。
3. 您是否已为目标板供电？调试器无法为目标板提供大于 150 mA 的电流。对于需要超过 150 mA 电流的应用，请使用外部电源为目标板供电。
4. 您是否在设置中使用了 **USB 集线器**？它是否已上电？如果仍有问题，请尝试不使用集线器来连接调试器（将调试器直接插入计算机）。
5. 您是否使用了随调试器一起提供的 **USB 线缆**？其他 USB 线缆可能质量差、过长或不支持 USB 通信。

### 6.2. 无法调试的首要原因

1. **振荡器不工作**。检查与振荡器有关的配置位设置。如果使用的是外部振荡器，请尝试使用内部振荡器。如果使用的是内部 PLL，请确保 PLL 的设置正确。
2. **目标板未上电**。检查电源线缆连接。
3. **V<sub>DD</sub> 电压不正确**。V<sub>DD</sub> 电压超出该器件规范。更多详细信息，请参见器件编程规范。
4. **断开物理连接**。调试器已断开与计算机和/或目标板的物理连接。检查通信线缆的连接。
5. **通信中断**。调试器与 PC 的通信已中断。在 MPLAB X IDE 或 MPLAB IPE 中重新连接调试器。
6. **器件未安装到位**。器件未正确安装在目标板上。如果调试器连接正确并且目标板已通电，但器件不存在或未完全插入，则可能会收到以下消息：Target Device ID (0x0) does not match expected Device ID (0x%x)  
(目标器件 ID (0x0) 与预期器件 ID (0x%x) 不匹配)，其中 %x 为预期器件 ID。
7. **器件受到代码保护**。检查配置位设置，确认是否使能了代码保护。
8. **应用程序代码损坏**。目标应用程序损坏或包含错误。尝试重新编译和重新编程目标应用程序。然后对目标板进行上电复位。
9. **编程引脚不正确**。没有在配置位中正确编程 PGC/PGD 引脚对（对于带多个 PGC/PGD 引脚对的器件）。
10. **需要额外的设置**。其他配置设置干扰调试。任何阻止目标器件执行代码的配置设置同样会阻止调试器将代码置于调试模式。
11. **欠压电压不正确**。欠压检测电压大于工作电压 V<sub>DD</sub>。这意味着器件处于复位状态，不能调试。
12. **连接不正确**。请参见“连接”一章中的准则了解正确的通信连接。
13. **请求无效**。调试器无法始终执行请求的操作。例如，如果目标应用程序正在运行，则调试器无法设置断点。

### 6.3. 通用

1. 该错误可能只发生一次。重试操作。
2. 通常，编程可能会出现问题。作为测试，使用  图标切换为运行模式，然后向目标器件烧写一个尽可能简单的应用程序（例如，使 LED 闪烁的程序）。如果程序没有运行，则说明目标设置存在错误。
3. 目标器件可能因某种原因（如过流）遭到损坏。开发环境明显对元器件不利。请考虑尝试使用另一个目标板。Microchip 提供了各种演示板来支持大部分单片机。考虑使用已知可以正常工作的一个应用程序来验证 MPLAB<sup>®</sup> PICKIT<sup>™</sup> 5 在线调试器的功能是否正确。

4. 查看调试器设置以确保应用设置正确。更多信息，请参见“连接”和“工作原理”两章。
5. 可能为您的电路设置了过高的编程速度。在 MPLAB X IDE 中，转到 **File > Project Properties**（文件 > 项目属性），然后依次选择 **PICKit 5** 类别和 *Program Options*（编程选项）选项类别。从 *Program Speed*（编程速度）旁边的下拉菜单中选择一个较慢的速度。默认为 Normal（正常速度）。
6. 在某些情况下，调试器可能无法正常运行，可能需要下载固件或需要重新编程调试器。请参见以下部分以确定其他操作。

## 6.4. 如何使用硬件工具紧急引导固件恢复实用程序

### NOTICE

仅使用此实用程序将硬件工具引导固件恢复为出厂状态。仅在硬件工具不能在任何计算机上运行时使用。

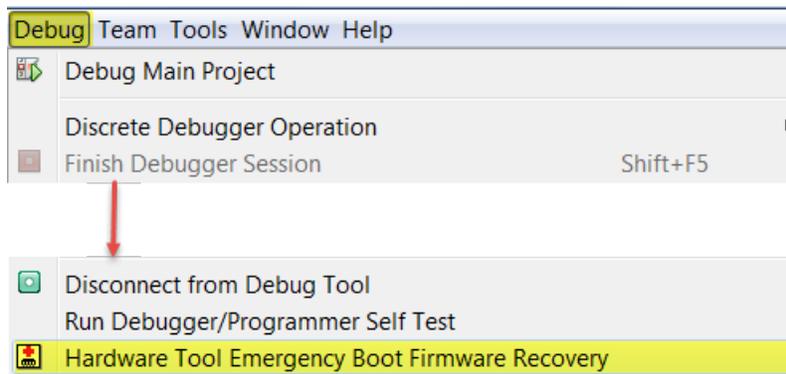
在极少数情况下，可能需要强制调试器进入恢复引导模式（重新编程）；例如，如果在调试器连接到计算机时发生以下任何情况：

- 调试器没有点亮任何 LED。
- 上一节中描述的步骤不成功。

必须使用 **MPLAB X IDE v6.10** 或更高版本才能为 **MPLAB PICKit 5** 使用紧急恢复实用程序。

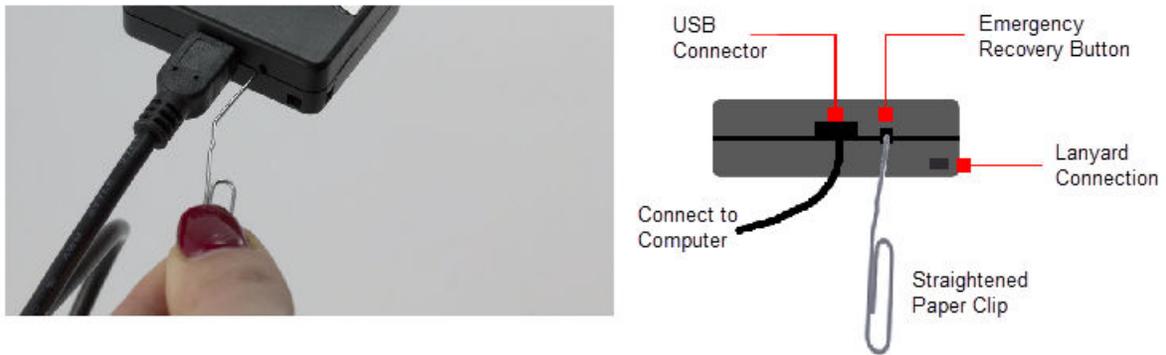
请严格遵循 MPLAB X IDE 中主菜单选项 **Debug > Hardware Tool Emergency Boot Firmware Recovery**（调试 > 硬件工具紧急引导固件恢复）下的说明进行操作。

图 6-1. 选择紧急实用程序



下图给出了紧急恢复按钮在 MPLAB PICKit 5 在线调试器上的位置。

图 6-2. 紧急恢复按钮



如果该过程成功，则恢复向导将显示成功画面。MPLAB PICKit 5 现在可以运行，并且能够与 MPLAB X IDE 通信。

如果该过程失败，请再次尝试。如果第二次仍然失败，请通过 [support.microchip.com](http://support.microchip.com) 与 Microchip 技术支持联系。

## 7. 常见问题解答

本章提供有关 MPLAB PICKit 5 在线调试器系统的常见问题的解答。

### 7.1. 工作原理

芯片中的什么组件允许其与 **MPLAB PICKit 5 在线调试器** 通信？

芯片内的模块支持不同类型的通信接口。MPLAB PICKit 5 在线调试器既可通过 ICSP™ 接口与器件通信，也可借助适配器板通过其他接口与器件通信。如果通信模块支持访问器件内置调试电路，则调试器可访问该电路来执行调试功能。

必须运行调试执行程序对处理器吞吐量会有什么影响？

对于 PIC MCU 器件，调试执行程序在调试会话期间编程到程序存储器或专用存储器中。运行模式下，调试执行程序并不运行，因此在运行代码时不会降低吞吐量，即调试器不会占用目标器件的任何周期。

**MPLAB PICKit 5 在线调试器** 是否具有与其他在线仿真器/调试器一样的复杂断点？

没有。但您可以根据数据存储单元或程序地址中的值来暂停程序执行。

**MPLAB PICKit 5 在线调试器** 是否具有复杂断点？

支持。您可以根据数据存储单元中的值来暂停程序执行。如果在暂停程序执行前有多个事件发生，还可以设置顺序断点序列。但只能设置两个序列。此外，还可以设置 AND 条件以及执行 PASS 计数。

**MPLAB PICKit 5 在线调试器** 是否采用光隔离或电隔离？

未采用。您不能对当前系统施加悬浮电压或高电压（120V）。

**MPLAB PICKit 5 在线调试器** 是否会降低程序的运行速度？

不会。器件将以数据手册中指定的任何器件速度运行。

可以调试以任何速度运行的 **dsPIC DSC** 器件吗？

MPLAB PICKit 5 在线调试器能够以器件数据手册中指定的任何器件速度进行调试。

### 7.2. 出现的问题

考虑以下方面：

编程器件后，执行校验失败。这是编程问题吗？

如果选中 **Run Main Project**（运行主项目）图标（），则器件将在编程后立即自动运行。因此，如果代码对闪存存储器进行了更改，校验可能会失败。要阻止代码在编程后立即运行，请选择 **Hold in Reset**（保持复位）。

我的计算机进入掉电/休眠模式，并且现在调试器无法工作。这是怎么回事？

长时间使用调试器时（尤其是用作调试器时），请确保在计算机操作系统的 **Power Options Dialog**（电源选项对话框）窗口中禁止休眠模式。转到 **Hibernate**（休眠）选项卡，并取消选中 **Enable hibernation**（启用休眠）复选框。这可确保保持所有 USB 子系统组件之间的所有通信。

我已将外设设置为暂停时不冻结，但它却突然发生冻结。这是怎么回事？

对于 dsPIC30F/33F 和 PIC24F/H 器件，外设控制寄存器中的一个保留位（通常为 bit 14 或 bit 5）由调试器用作 **Freeze**（冻结）位。如果您对整个寄存器执行过写操作，那么您可能改写了此位（在调试模式下，用户可访问此位）。

为避免上述问题，请只在应用程序中对想要更改的位执行写操作（通过 **BTS** 和 **BTC**），而不是对整个寄存器执行写操作（通过 **MOV**）。

**使用 16 位器件时，发生了意外复位。如何确定复位的原因？**

考虑以下几个方面：

- 要确定复位源，请检查 RCON 寄存器。
- 在中断服务程序（Interrupt Service Routine, ISR）中处理陷阱/中断。应包括 trap.c 样式的代码，例如

```
void __attribute__((__interrupt__)) _OscillatorFail(void);
:
void __attribute__((__interrupt__)) _AltOscillatorFail(void);
:
void __attribute__((__interrupt__)) _OscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;          //Clear the trap flag
    while (1);
}
:
void __attribute__((__interrupt__)) _AltOscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;
    while (1);
}
:
```

- 使用 ASSERT。例如：ASSERT (IPL==7)

## 8. 错误消息

MPLAB PICkit 5 在线调试器会产生各种不同的错误消息；其中一些错误消息比较特殊，而其他的都可以用常规纠正措施解决。通常，请阅读错误消息下面的所有指示信息。如果无法解决问题或者无指示信息，请参见以下章节。

### 8.1. 错误消息类型

#### 8.1.1. 调试器到目标器件的通信错误

##### **Failed to send database**（无法发送数据库）

如果收到这个错误：

- 尝试重新下载。该错误可能只发生一次。
- 尝试手动下载编号最高的 .jam 文件。

如果无法解决问题或者无指示信息，请参见[调试器与计算机的通信错误纠正措施](#)。

#### 8.1.2. 损坏/过时安装错误

##### **Failed to download firmware**（无法下载固件）

如果存在 hex 文件：

- 重新连接并重试。
- 如果无法解决问题，则表明该文件可能已损坏。重新安装 MPLAB X IDE 或 MPLAB IPE。

如果不存在 hex 文件：

- 重新安装 MPLAB X IDE 或 MPLAB IPE。

##### **Unable to download debug executive**（无法下载调试执行程序）

如果尝试调试时收到此错误：

1. 取消将调试器选作调试工具。
2. 关闭项目，然后关闭 MPLAB X IDE 或 MPLAB IPE。
3. 重新启动 MPLAB X IDE 或 MPLAB IPE，然后重新打开项目。
4. 重新选择调试器作为调试工具并再次尝试编程目标器件。

##### **Unable to download program executive**（无法下载编程执行程序）

如果尝试编程时收到此错误：

1. 取消将调试器选作编程器。
2. 关闭项目，然后关闭 MPLAB X IDE 或 MPLAB IPE。
3. 重新启动 MPLAB X IDE 或 MPLAB IPE，然后重新打开项目。
4. 重新选择调试器作为编程器并再次尝试编程目标器件。

如果无法解决问题或者无指示信息，请参见[安装损坏纠正措施](#)。

#### 8.1.3. 调试失败错误

**The target device is not ready for debugging. Please check your Configuration bit settings and program the device before proceeding.**（目标器件未准备好调试。请检查配置位设置并编程器件，然后继续。）

如果您在对器件进行初次编程之前便尝试运行，将收到此消息。如果在这种情况下收到此消息，或者在编程器件后立即收到此消息，请参见[调试失败纠正措施](#)。

#### 器件受到代码保护。

您尝试进行操作（读、编程、空白检查或校验）的器件受到代码保护，即，无法读或修改代码。检查配置位设置，看是否使能了代码保护（**Windows > Target Memory Views > Configuration Bits**（窗口 > 目标存储器视图 > 配置位））。

要禁止代码保护，请根据器件数据手册，在代码或 **Configuration Bits** 窗口中置 1 或清零相应的配置位。然后擦除并重新编程整个器件。

如果这些操作无法解决问题，请参见[调试器与目标器件的通信错误纠正措施](#)和[调试失败纠正措施](#)。

### 8.1.4. 其他错误

**MPLAB PICKIT 5 is busy. Please wait for the current operation to finish.**（ICD 5 忙。请等待当前操作完成。）

如果在尝试取消将调试器选作调试器或编程器时收到此错误：

1. 等待。使调试器有时间完成任何应用程序任务。然后重新尝试取消选择调试器。
2. 选择 **Finish Debugger Session**（结束调试器会话） 来停止所有运行中的应用程序。然后重新尝试取消选择调试器。
3. 将调试器从计算机上拔下。然后重新尝试取消选择调试器。
4. 关闭 MPLAB X IDE。

### 8.1.5. 错误消息列表

表 8-1. 按字母顺序排列的错误消息列表

|   |
|---|
| AP_VER=算法插件版本   |
| AREAS_TO_PROGRAM=将编程以下存储区：  |
| AREAS_TO_READ=将读取以下存储区：   |
| AREAS_TO_VERIFY=将校验以下存储区：   |
| BLANK_CHECK_COMPLETE=空白检查完成，器件空白。   |
| BLANK_CHECK_FAILED=空白检查失败。器件不是空白的。  |
| BLANK_CHECKING=正在进行空白检查……   |
| BOOT_CONFIG_MEMORY=引导配置存储器  |
| BOOT_VER=引导版本   |
| BOOTFLASH=引导闪存  |
| BP_CANT_B_DELETED_WHEN_RUNNING=目标器件运行时无法移除软件断点。所选断点将在下次目标器件暂停执行时移除。                       |
| CANT_CREATE_CONTROLLER=无法找到工具控制器类。  |
| CANT_FIND_FILE=无法找到文件%s 的位置。  |
| CANT_OP_BELOW_LVPTHRESH=所选电压%f 低于最小擦除电压%f。此电压下无法继续工作。                                     |
| CANT_PGM_USEROTP=调试工具无法编程用户 OTP 存储器，因为它不是空白的。请从要编程的存储器中排除用户 OTP 存储器或切换到具有空白用户 OTP 存储器的器件。 |
| CANT_PRESERVE_PGM_MEM=无法保留程序存储器：无效范围起始地址 = %08x，结束地址 = %08x。                              |
| CANT_READ_REGISTERS=无法读取目标寄存器。  |
| CANT_READ_SERIALNUM=无法读取器件序列号。  |
| CANT_REGISTER_ALTERNATE_PNP=无法针对多个 USB 产品 ID 注册 PNP 事件。                                   |
| CANT_REMOVE_SWPS_BUSY=PICKIT 5 当前繁忙，此时无法移除软件断点。   |

|   |
|---|
| CHECK_4_HIGH_VOLTAGE_VPP=注意：检查 MPLAB® IDE 中选择的器件 (%s) 是否与物理连接到调试工具的器件相同。如果在连接 3.3V 器件时选择 5V 器件，则会导致调试器检查器件 ID 时损坏器件。是否希望继续操作？   |
| CHECK_PGM_SPEED=您已将编程速度设置为%s。电路板上的电路可能要求您减慢速度。请在工具属性中将设置更改为“低”并再次尝试操作。  |
| CHECK_SLAVE_DEBUG=调试可能失败，因为尚未使能主项目的从内核设置中的“Debug”复选框。请确保已使能此设置。   |
| COMM_PROTOCOL_ERROR=调试工具已出现通信错误。工具将复位且应尽快重新枚举。  |
| COMMAND_TIME_OUT=PICKIT 5 等待命令%02x 的响应超时。   |
| CONFIGURATION=配置  |
| CONFIGURATION_MEMORY=配置存储器  |
| CONNECTION_FAILED=连接失败  |
| CORRUPTED_STREAMING_DATA=检测到无效的数据流。运行时观察或跟踪数据可能不再有效。建议重启您的调试会话。   |
| CPM_TO_TARGET_FAILED=ControlPointMediator.ToTarget()期间出现异常。   |
| DATA_FLASH_MEMORY=数据闪存  |
| DATA_FLASH=数据闪存   |
| DEBUG_INFO_PGM_FAILED=无法进入调试模式，因为调试信息编程失败。配置位的无效组合可能会导致这一问题。  |
| DEBUG_READ_INFO=受目标振荡器速度的影响，在调试模式下读取器件可能会花费较长时间。缩小要读取的范围（在 ICD 4 项目属性下）可缓解这一情况。中止操作可用于在必要时终止读操作。  |
| DEVICE_ID_REVISION=器件 ID 版本   |
| DEVICE_ID=器件 ID   |
| DEVICE_INFO_CONFIG_BITS_MASK=地址 = %08x，掩膜 = %08x  |
| DEVICE_INFO_MEMBERS=器件信息： pcAddress = %08x， Vpp = %.2f， useRowEraseIfVoltageIsLow = %s， voltageBelowWhichUseRowErase = %.2f， deviceName = %s， programmerType = %s       |
| DEVICE_INFO_MEMINFO_MEMBERS= 器件信息： mask = %04x， exists = %s， startAddr = %08x， endAddr = %08x， rowSize = %04x， rowEraseSize = %04x， addrInc = %04x， widthProgram = %04x |
| DEVICE_INFO=器件信息： 值：  |
| DEVID_MISMATCH=目标器件 ID (0x%x) 是无效的器件 ID。请检查与目标器件的连接。  |
| DFU_NOT_SUPPORTED=MPLAB X 检测到连接的工具有此版本不支持的功能。请下载最新版本 MPLAB X 以使用此工具。  |
| DISCONNECT_WHILE_BUSY=工具在繁忙时断开。   |
| EEDATA_MEMORY=EEData 存储器  |
| EEDATA=EEData   |
| EMPTY_PROGRAM_RANGES=编程操作未完成，因为未选择任何存储区。  |
| EMULATION_MEMORY_READ_WRITE_ERROR=尝试读/写 MPLAB 的仿真存储器时出现错误： 地址=%08x  |
| END=结束  |
| ENSURE_SELF_TEST_READY=请确保 RJ-11 线缆连接到测试板后再继续。  |
| ENSURE_SELF_TEST_READY=请确保 RJ-11 线缆连接到测试板后再继续。是否要继续？  |
| ENV_ID_GROUP=器件标识   |
| ERASE_COMPLETE=擦除成功   |
| ERASING=正在进行擦除……  |
| FAILED_2_PGM_DEVICE=器件编程失败。   |
| FAILED_CREATING_COM=无法创建通信对象。   |
| FAILED_CREATING_DEBUGGER_MODULES=初始化失败： 创建调试器模块失败。  |
| FAILED_ERASING=器件擦除失败。  |
| FAILED_ESTABLISHING_COMMUNICATION=无法建立工具通信。   |
| FAILED_GETTING_DBG_EXEC=尝试加载调试执行程序时出现问题。  |
| FAILED_GETTING_DEVICE_INFO=初始化失败： 检索器件数据库 (.pic) 信息时失败。   |
| FAILED_GETTING_EMU_INFO=初始化失败： 仿真数据库信息获取失败。   |
| FAILED_GETTING_HEADER_INFO=初始化失败： 仿真头数据库信息获取失败。   |

|  |
|--|
| FAILED_GETTING_PGM_EXEC=尝试加载编程执行程序时出现问题。   |
| FAILED_GETTING_TEX=无法获取 ToolExecMediator   |
| FAILED_GETTING_TOOL_INFO=初始化失败：检索工具数据库（.ri4）信息时失败。   |
| FAILED_INITING_DATABASE=初始化失败：无法初始化工具数据库对象。  |
| FAILED_INITING_DEBUGHANDLER=初始化失败：无法初始化 DebugHandler 对象。   |
| FAILED_PARSING_FILE=固件文件%s 解析失败%s  |
| FAILED_READING_EMULATION_REGS=仿真存储器读取失败。   |
| FAILED_READING_MPLAB_MEMORY=无法读取%s 存储器的%0x08 至%0x08 部分。  |
| FAILED_READING_SECURE_SEGMENT=读取安全段配置位失败。  |
| FAILED_SETTING_PC=无法设置 PC。   |
| FAILED_SETTING_SHADOWS=无法正确设置影子寄存器。  |
| FAILED_SETTING_XMIT_EVENTS=无法同步运行时数据信号量。   |
| FAILED_STEPPING=对目标器件单步执行失败。   |
| FAILED_TO_GET_DEVID=器件 ID 获取失败。请确保目标器件已连接并再次尝试操作。  |
| FAILED_TO_INIT_TOOL=PICKit 5 初始化失败   |
| FAILED_UPDATING_BP=断点更新失败：文件： %s， 地址： %08x   |
| FAILED_UPDATING_FIRMWARE=无法正确更新固件。   |
| FILE_REGISTER=文件寄存器  |
| FIRMWARE_DOWNLOAD_TIMEOUT=固件下载过程期间 PICKit 5 超时。  |
| FLASH_DATA_MEMORY=闪存数据存储   |
| FLASH_DATA=闪存数据  |
| FRCINDEBUG_NEEDS_CLOCKSWITCHING=要在调试模式下使用 FRC，必须使能时钟切换配置位设置。请使能时钟切换并重试所请求的操作。                            |
| FW_DOESNT_SUPPORT_DYNBP=当前 PICKit 5 固件不支持为所选器件设置运行时断点。请下载固件版本%02x.%02x.%02x 或更高版本。                       |
| GOOD_ID_MISMATCH=目标器件 ID（0x%x）是有效器件 ID，但与选择的预期器件 ID（0x%x）不匹配。  |
| HALTING=暂停中……  |
| HIGH=高电平   |
| HOLDMCLR_FAILED=保持复位失败。  |
| IDS_SELF_TEST_BOARD_PASSED=PICKit 5 正常工作。如果您的目标电路仍存在问题，请参见在线帮助中的 Target Board Considerations（目标板注意事项）部分。 |
| IDS_ST_CLKREAD_ERR=测试接口 PGC 时钟线读取失败。   |
| IDS_ST_CLKREAD_NO_TEST=测试接口 PGC 时钟线读取未经测试。   |
| IDS_ST_CLKREAD_SUCCESS=测试接口 PGC 时钟线读取成功。   |
| IDS_ST_CLKWRITE_ERR=测试接口 PGC 时钟线写入失败。请确保测试器连接正确。   |
| IDS_ST_CLKWRITE_NO_TEST=测试接口 PGC 时钟线写入未经测试。  |
| IDS_ST_CLKWRITE_SUCCESS=测试接口 PGC 时钟线写入成功。  |
| IDS_ST_DATREAD_ERR=测试接口 PGD 数据线读取失败。   |
| IDS_ST_DATREAD_NO_TEST=测试接口 PGD 数据线读取未经测试。   |
| IDS_ST_DATREAD_SUCCESS=测试接口 PGD 数据线读取成功。   |
| IDS_ST_DATWRITE_ERR=测试接口 PGD 数据线写入失败。  |
| IDS_ST_DATWRITE_NO_TEST=测试接口 PGD 数据线写入未经测试。  |
| IDS_ST_DATWRITE_SUCCESS=测试接口 PGD 数据线写入成功。  |
| IDS_ST_LVP_ERR=测试接口 LVP 控制线故障。   |
| IDS_ST_LVP_NO_TEST=测试接口 LVP 控制线未经测试。   |
| IDS_ST_LVP_SUCCESS=测试接口 LVP 控制线测试成功。   |
| IDS_ST_MCLR_ERR=测试接口 $\overline{\text{MCLR}}$ 电平故障。  |

|  |
|--|
| IDS_ST_MCLR_NO_TEST=测试接口 $\overline{\text{MCLR}}$ 电平未经测试。  |
| IDS_ST_MCLR_SUCCESS=测试接口 $\overline{\text{MCLR}}$ 电平测试成功。  |
| IDS_TEST_NOT_COMPLETED=接口测试无法完成。请联系您当地的 FAE/CAE 为设备在 SAR 系统中提交请求以进行维修或替换。  |
| INCOMPATIBLE_FW=PICKIT 5 固件与 MPLAB X 软件的当前版本不兼容。   |
| INVALID_ADDRESS=操作无法继续，因为 %s 地址超出器件地址范围 0x%08x-0x%08x。   |
| JTAG_NEEDS_JTAGEN=JTAG 适配器需要开启 JTAG 使能配置位。请在继续之前使能此配置位。  |
| MCLR_HOLD_RESET_NO_MAINTAIN_POWER=警告：您正在通过 PICKIT 5 为目标器件供电，并且未在 PICKIT 5 的 Power（电源）属性页面上选择“Maintain active power”（保持有效供电）选项。如果没有选择此选项，则在当前会话结束后无法保证 MCLR 的状态（保持/从复位释放）。                                    |
| MCLR_OFF_ID_WARNING=如果您正在使用低电压编程并且目标器件上的 MCLRE 配置位设置为 OFF，则可以解释器件 ID 不正确的原因。在这种情况下，请切换到 PICKIT 5 Program Options 属性页面上的“Use high voltage programming mode entry”（使用高电压编程模式）编程模式设置，然后再次尝试操作。                  |
| MCLR_OFF_WARNING=如果要继续将 MCLRE 配置位设置为 OFF，请切换到 PICKIT 5 Program Options 属性页面上的“Use high voltage programming mode entry”编程模式设置。  |
| MEM_INFO=器件信息：存储器信息值：  |
| MEM_RANGE_ERROR_BAD_END_ADDR=收到无效编程范围结束地址 %s。请在调试工具的“Memories to Program”（待编程存储器）属性页面上检查手动编程范围。  |
| MEM_RANGE_ERROR_BAD_START_ADDR=收到无效编程范围起始地址 %s。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| MEM_RANGE_ERROR_END_LESSTHAN_START=收到无效编程范围：结束地址 %s < 起始地址 %s。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。   |
| MEM_RANGE_ERROR_ENDADDR_NOT_ALIGNED=收到无效编程范围：结束地址 %s 未对齐到适当的 0x%x 地址边界。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| MEM_RANGE_ERROR_STARTADDR_NOT_ALIGNED=收到无效编程范围：起始地址 %s 未对齐到适当的 0x%x 地址边界。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| MEM_RANGE_ERROR_UNKNOWN=尝试验证用户输入的存储器范围时出现未知错误。   |
| MEM_RANGE_ERROR_WRONG_DATABASE=无法在验证用户输入的存储器范围时访问数据对象。   |
| MEM_RANGE_OUT_OF_BOUNDS=所选编程范围 %s 未在所选存储区的正确范围内。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。   |
| MEM_RANGE_STRING_MALFORMED=在“Memories to Program”属性页面上输入的存储器范围（%s）未采用正确格式。   |
| MISSING_BOOT_CONFIG_PARAMETER=无法在数据库中找到引导配置起始/结束地址。  |
| MUST_NOT_USE_LVP_WHEN_LVPCFG_OFF=MPLAB 检测到器件上的低电压配置位已关闭，并且您已在调试工具的属性页面上选择了低电压编程选项。如果要使用低电压编程选项，必须首先执行以下操作：1) 关闭调试工具的 Program Options 属性页面上的低电压编程选项，2) 将低电压配置位编程为开启，3) 开启调试工具的 Program Options 属性页面上的低电压编程选项。 |
| MUST_SET_LVPBIT_WITH_LVP=低电压编程功能需要在目标器件上使能 LVP 配置位。请使能此配置位并再次尝试操作。   |
| NEW_FIRMWARE_NO_DEVICE=正在下载固件。   |
| NEW_FIRMWARE=现在为目标器件（%s）下载新固件： %s  |
| NMMR=NMMR  |
| NO_DYNAMIC_BP_SUPPORT_AT_ALL=当前器件不支持在器件运行时设置断点。断点将在下次运行器件之前应用。   |
| NO_PGM_HANDLER=无法编程软件断点。编程处理程序尚未初始化。   |
| NO_PROGRAMMING_ATTEMPTED=MPLAB 的存储器空白，因此未尝试编程操作。   |
| NORMAL=正常  |
| OP_FAILED_FROM_CP=所请求的操作失败，因为器件受到代码保护。   |
| OpenIDE-Module-Name=PICKIT 5   |
| OPERATION_INFO_MEMBERS=操作信息：类型 = %s，掩膜 = %08x，擦除 = %s，生产模式 = %s。   |
| OPERATION_INFO_TRANSFER_INFO_MEMBERS=操作信息：起始地址 = %x，结束地址 = %x，缓冲区长度 = %d，类型 = %s，掩膜 = %08x。  |
| OPERATION_INFO=操作信息：值：   |
| OPERATION_NOT_SUPPORTED=所选器件不支持此操作   |
| OUTPUTWIN_TITLE=PICKIT 5   |

|  |
|--|
| PERIPHERAL=外设  |
| POWER_ERROR_NO_POWER_SRC=配置为目标板自行供电，但 V <sub>DD</sub> 上未检测到任何电压。请确保您已为目标板供电并重试。  |
| POWER_ERROR_POWER_SRC_CONFLICT=配置为从工具向目标板供电，但 V <sub>DD</sub> 上已检测到电压。这是一种冲突。请确保目标板未向工具提供电压并重试。  |
| POWER_ERROR_SLOW_DISCHARGE=V <sub>DD</sub> 上似乎有多余的电容，导致系统放电变慢和关闭。考虑最大程度减小总电容负载或使用来自目标板的电源来避免放电延时。  |
| POWER_ERROR_UNKNOWN=发生未知电源错误。  |
| POWER_ERROR_VDD_TOO_HIGH=所需 V <sub>DD</sub> 电压超出范围。它超出 5.5V 的最大电压。   |
| POWER_ERROR_VDD_TOO_LOW=所需 V <sub>DD</sub> 电压超出范围。它低于 1.5V 的最小电压。  |
| POWER_ERROR_VPP_TOO_HIGH=所需 V <sub>PP</sub> 电压超出范围。它超出 14.2V 的最大电压。  |
| POWER_ERROR_VPP_TOO_LOW=所需 V <sub>PP</sub> 电压超出范围。它低于 1.5V 的最小电压。  |
| PRESERVE_MEM_RANGE_ERROR_BAD_END_ADDR=收到无效保留范围结束地址%s。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| PRESERVE_MEM_RANGE_ERROR_BAD_START_ADDR=收到无效保留范围起始地址%s。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| PRESERVE_MEM_RANGE_ERROR_END_LESSTHAN_START=收到无效保留范围：结束地址%s < 起始地址%s。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| PRESERVE_MEM_RANGE_ERROR_ENDADDR_NOT_ALIGNED=收到无效保留范围：结束地址%s 未对齐到适当的 0x%x 地址边界。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| PRESERVE_MEM_RANGE_ERROR_STARTADDR_NOT_ALIGNED=收到无效保留范围：起始地址%s 未对齐到适当的 0x%x 地址边界。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。  |
| PRESERVE_MEM_RANGE_ERROR_UNKNOWN=尝试验证用户输入的保留范围时发生未知错误。   |
| PRESERVE_MEM_RANGE_ERROR_WRONG_DATABASE=无法在验证用户输入的存储器范围时访问数据对象。  |
| PRESERVE_MEM_RANGE_MEM_NOT_SELECTED=您已选择保留一个存储区，但尚未选择编程该存储区。请在调试工具的“Memories to Program”属性页面上检查保留范围，并确保也指定对任何保留的存储区进行编程。   |
| PRESERVE_MEM_RANGE_OUT_OF_BOUNDS=所选保留范围%s 未在所选存储区的正确范围内。请在调试工具的“Memories to Program”属性页面上检查手动编程范围。   |
| PRESERVE_MEM_RANGE_STRING_MALFORMED=在“Memories to Program”属性页面上输入的保留存储器范围（%s）未采用正确格式。  |
| PRESERVE_MEM_RANGE_WONT_BE_PROGRAMMED_AUTO_SELECT=在“Memories to Program”属性页面上输入的保留存储器范围（%s）的部分或全部未处于所选存储器的指定编程范围（%s）内。请取消选中“Memories to Program”属性页面上的“Auto select memories and ranges”（自动选择存储器和范围）选项，切换到手动模式并相应地调整范围。 |
| PRESERVE_MEM_RANGE_WONT_BE_PROGRAMMED=在“Memories to Program”属性页面上输入的保留存储器范围（%s）的部分或全部未处于所选存储器的指定编程范围（%s）内。请在调试工具的“Memories to Program”属性页面上检查保留范围。   |
| PROGRAM_CFG_WARNING=警告：您已选择编程配置存储器。将无效值编程到任何配置字段中可能会导致意外后果。请确保每个配置字段都有一个有效值。如果您不确定，可以先从器件读取配置值，然后仅更改您关注的字段。是否要继续编程？  |
| PROGRAM_COMPLETE=编程/校验完成   |
| PROGRAM_MEMORY=程序存储器   |
| PROGRAM=编程   |
| PROGRAMMING_DID_NOT_COMPLETE=编程未完成。  |
| READ_COMPLETE=读取完成   |
| READ_DID_NOT_COMPLETE=读取未完成。   |
| RELEASEMCLR_FAILED=从复位释放失败。  |
| REMOVING_SWBPS_COMPLETE=软件断点删除完成。  |
| REMOVING_SWBPS=正在删除软件断点……  |
| RESET_FAILED=器件复位失败。   |
| RESETTING=正在复位……   |

|   |
|---|
| RISKY_CFG_RANGE_REMOVED=配置存储器不包含在编程操作中，因为设置了“Exclude configuration memory from programming”（从编程中排除配置存储器）选项。要更改此设置，请转到 <b>Memories to Program</b> 属性页面，取消选中该设置。警告：如果未正确设置所有配置值，则在此器件上编程配置值可能会导致意外后果。建议先从器件读取配置值，然后仅更改您关注的字段。 |
| RUN_INTERRUPT_THREAD_SYNCH_ERROR=发生内部运行错误。建议重启您的调试会话。您可以继续运行，但某些运行时功能可能无法正常工作。  |
| RUN_TARGET_FAILED=无法运行目标器件。   |
| RUNNING=正在运行  |
| SERIAL_NUM=序列号:   |
| SETTING_SWBPS=正在设置软件断点……  |
| STACK=堆栈  |
| START_AND_END_ADDR=起始地址 = 0x%x, 结束地址 = 0x%x   |
| START=起始  |
| TARGET_DETECTED=检测到目标电压   |
| TARGET_FOUND=找到目标器件%s。  |
| TARGET_HALTED=目标器件暂停执行  |
| TARGET_NOT_READY_4_DEBUG=目标器件未准备好调试。请检查配置位设置并编程器件，然后继续。此故障最常见的原因是振荡器和/或 PGC/PGD 设置。   |
| TARGET_VDD=目标 V <sub>DD</sub> :   |
| TEST=测试   |
| TOOL_INFO_MEMBERS=工具信息: speedLevel = %d, PGCResistance = %d, PGDResistance = %d, PGCPullDir = %s, PGDPullDir = %s, ICSPSelected = %s。   |
| TOOL_INFO=工具信息: 值:  |
| TOOL_IS_BUSY=PICKit 5 忙。请等待当前操作完成。  |
| TOOL_SUPPLYING_POWER=PICKit 5 正在为目标板供电 (%.2fV)。   |
| TOOL_VDD=VDD:   |
| TOOL_VPP=V <sub>pp</sub> :  |
| UNABLE_TO_OBTAIN_RESET_VECTOR=PICKit 5 无法检索复位向量地址。这表示未定义_reset 符号，因此可能阻止器件正常启动。   |
| UNKNOWN_MEMTYPE=存储器类型未知   |
| UNLOAD_WHILE_BUSY=PICKit 5 在繁忙时被卸载。在再次使用 PICKit 5 之前，请拔下并重新连接 USB 线缆。   |
| UPDATING_APP=正在更新固件应用程序……   |
| UPDATING_BOOTLOADER=正在更新固件自举程序。   |
| USE_LVP_PROGRAMMING=注：如果要使用低电压编程对此器件进行编程，请在此对话框中选择 <b>Cancel</b> （取消）。然后转到项目属性的 PICKit 5 节点，并选中 <b>Program Options</b> 选项类别窗格的 <b>Enable Low Voltage Programming</b> （使能低电压编程）复选框（低电压编程对调试操作无效）。                            |
| USERID_MEMORY=用户 ID 存储器   |
| USERID=用户 ID  |
| VERIFY_COMPLETE=校验成功  |
| VERIFY_FAILED=校验失败  |
| VERSIONS=版本   |
| VOLTAGE_LEVEL_BAD_VALUE_EX=您已在 PICKit 5 Power 属性页面上为电压大小输入了无效值%s。请在继续操作之前解决此问题。   |
| VOLTAGE_LEVEL_BAD_VALUE=无法解析电压大小%s。请输入有效的电压。  |
| VOLTAGE_LEVEL_OUT_OF_RANGE=您输入的目标电压大小%.3f 超出了器件的范围 (%.3f - %.3f)。   |
| VOLTAGES=电压   |
| WOULD_YOU_LIKE_TO_CONTINUE=是否要继续?   |
| WRONG_PICKIT_5_FLAVOR=您的 PICKit 5 硬件需要更新，请联系 PICKit 5_Update@microchip.com 获取替换件。   |

## 8.2. 常规纠正措施

### 8.2.1. 读/写错误纠正措施

如果收到读或写错误：

1. 是否单击了 **Debug > Reset**（调试 > 复位）？此操作可能产生读/写错误。
2. 重试操作。该错误可能只发生一次。
3. 确保目标板已上电并且器件的电压正确。有关所需器件电压，请参见器件数据手册。
4. 确保调试器到目标器件的连接正确（PGC 和 PGD 已连接）。
5. 对于写错误，确保在 **Project Properties** 窗口中调试器的 **Program Options** 上选中 **Erase all before Program**（编程之前全部擦除）。
6. 确保所使用的线缆的长度正确。

相关信息

[Debug Options](#)（调试选项）

### 8.2.2. 调试器与目标器件的通信错误纠正措施

如果 MPLAB PICKit 5 在线调试器与目标器件之间不能通信：

1. 选择 **Debug > Reset**，然后重试操作。
2. 确保所使用的线缆的长度正确。

### 8.2.3. 调试器与计算机的通信错误纠正措施

如果 MPLAB PICKit 5 在线调试器与 MPLAB X IDE 或 MPLAB IPE 之间不能通信：

1. 从计算机上拔下调试器，然后重新插入。
2. 重新连接调试器。
3. 重试操作。该错误可能只发生一次。
4. 所安装的 MPLAB X IDE 或 MPLAB IPE 版本可能与 MPLAB PICKit 5 在线调试器上加载的固件版本不一致。按[安装损坏纠正措施](#)中概述的步骤操作。
5. 计算机 USB 端口可能存在问题。请参见 [USB 端口通信错误纠正措施](#)。

### 8.2.4. 安装损坏纠正措施

该问题很可能是由于 MPLAB X IDE 或 MPLAB IPE 的安装不完整或者损坏引起的。

1. 卸载计算机上所有的 MPLAB X IDE 或 MPLAB IPE 版本。
2. 重新安装所需的 MPLAB X IDE 或 MPLAB IPE 版本。
3. 如果问题依然存在，请联系 Microchip 技术支持。

### 8.2.5. USB 端口通信错误纠正措施

该问题很可能是由于通信端口故障或者不存在引起的。

1. 重新连接 MPLAB PICKit 5 在线调试器。
2. 确保调试器已物理连接到计算机上适当的 USB 端口。
3. 确保在 **Project Properties** 窗口中的调试器选项中选择了适当的 USB 端口。
4. 确保 USB 端口未被其他设备使用。
5. 如果使用 USB 集线器，请确保已将其上电。
6. 确保已加载 USB 驱动程序。

### 8.2.6. 调试失败纠正措施

MPLAB PICkit 5 在线调试器无法执行调试操作。产生此错误消息的情况有很多，其中几个如下所述。

1. 器件没有时钟源，或者配置位设置中选择的时钟源不工作。
2. 器件上的 PGD（数据）和 PGC（时钟）引脚正在被应用使用。这两个引脚为调试器所需，在调试模式期间不得被应用控制。
3. 许多器件包含多个 PGD/PGC 对。使用配置位设置来选择用于调试的 PGD/PGC。在编程模式下，可以使用连接到调试工具的任何 PGD/PGC 对。对于调试模式，配置位设置必须与物理连接到 MCU 的 PGD/PGC 对相匹配。
4. 当器件尚未编程时，从 MPLAB X IDE 中选择 Run Main Project 将生成此消息。此时，可改用调试模式



此信息基于知识库文章 [Why do I get the following error while trying to debug: "ICD3Err0040: The target device is not ready for debugging"?](#)

#### 相关信息

[疑难解答](#)

### 8.2.7. 内部错误纠正措施

内部错误难以预料并且不应该发生。它们用于 Microchip 内部开发。

最可能的原因是安装损坏（[安装损坏纠正措施](#)）。

另一个可能的原因是系统资源不足。

1. 尝试重新启动系统以释放存储空间。
2. 确保硬盘上有足够的可用空间（而且没有过多的磁盘碎片）。

如果问题依然存在，请联系 Microchip 技术支持。

## 9. 调试器功能汇总

下面列出了 MPLAB PICKit 5 在线调试器功能的汇总。

### 9.1. 调试器选择与切换

使用 Project Properties 对话框可为项目选择或切换调试器。要进行切换，必须有多个调试器与计算机相连。MPLAB X IDE 将通过显示不同序列号来区分各个调试器。

要选择或更改项目使用的调试器：

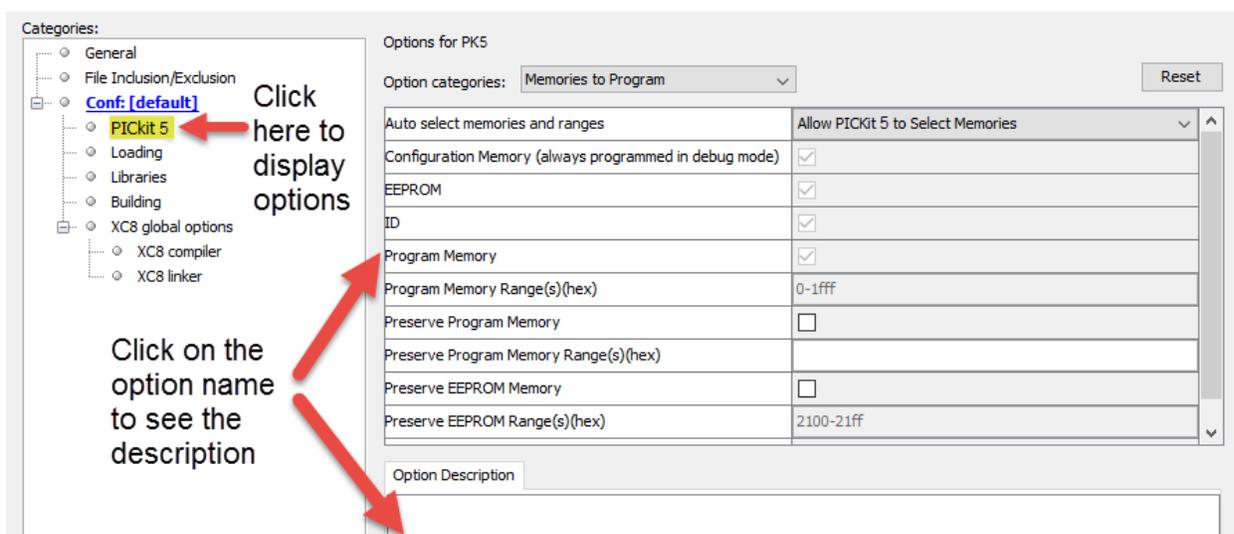
- 通过执行下列一项操作打开 **Project Properties** 对话框：
  - 单击 **Projects** 窗口中的项目名称，然后选择 **File > Project Properties**。
  - 或
  - 右键单击 **Projects** 窗口中的项目名称并选择 **Properties**。
- 在左侧的 **Categories**（类别）下，展开 **Conf:[default]** 以显示 PICKit 5。
- 在 **Hardware Tools**（硬件工具）下，找到 **PICKit 5** 并单击序列号（Serial Number, SN）来选择项目中使用的调试器，然后单击 **Apply**（应用）。

### 9.2. 调试器选项选择

在 Project Properties 对话框中设置调试器选项。在 **Categories** 下单击 **PICKit 5**，显示 **Options for PK5**（见下图）。使用 **Option categories** 下拉列表选择不同的选项。单击某个选项的名称可在下面的 **Option Description**（选项说明）框中查看其说明。单击选项名称的右侧可对其进行选择或更改。

**注：** 可用的选项类别以及这些类别中的选项取决于您选择的器件。

图 9-1. MPLAB® PICKit™ 5 的选项



设置选项后，单击 **Apply** 或 **OK**。还需单击 MPLAB X IDE 仪表盘显示中的 Refresh Debug Tool（刷新调试工具）图标 ，更新所做的任何更改。

#### 9.2.1. Memories to Program

选择目标器件中要编程的存储器。下表列出了所有可能的选项，但只有可用于您所选器件的选项将显示在 MPLAB X IDE 中。

**注：**如果选中 **Erase All Before Program**（如 **Program Options** 中所示），则编程前将擦除所有器件存储器。

**表 9-1. Memories to Program 选项类别**

|  |  |
|--|--|
| Auto select memories and ranges                                  | <b>Allow PICkit 5 to Select Memories</b> （允许 PICkit Basic 选择存储器）——调试器根据选择的器件和默认设置确定要编程的存储器和范围。 <b>Manually select memories and ranges</b> （手动选择存储器和范围）——选择要编程的存储器的类型和范围（见下文）。  |
| Configuration Memory（配置存储器）                                      | 选中时可将 <b>配置存储器</b> 包含在要编程的区域中。始终在调试模式下编程该存储器。  |
| Boot Flash（引导闪存）   | 选中时可将 <b>引导闪存</b> 存储器包含在要编程的区域中。始终在调试模式下编程该存储器。  |
| EEPROM   | 选中时可将 <b>EEPROM</b> 存储器包含在要编程的区域中。   |
| ID   | 选中时可编程用户 ID。   |
| Program Memory（程序存储器）  | 选中时可编程下面指定的目标程序存储器范围。  |
| Program Memory Range(s) (hex)（程序存储器范围（十六进制））                     | 要编程的程序存储器的范围。具体来说，是程序存储器中要编程、读取或校验的起始和结束十六进制地址范围。每个范围必须是两个十六进制数（范围的起始地址和结束地址），这两个数用短划线隔开。多个范围之间必须用逗号隔开（例如 0-ff, 200-2ff）。范围必须对齐到 0x800 地址边界。<br><b>注：</b> 地址范围不适用于擦除功能。擦除功能将擦除器件上的所有数据。  |
| Preserve Program Memory（保留程序存储器）                                 | 使能该项将使器件上的当前程序存储器读入 MPLAB X IDE 的存储区，然后在编程完成后重新编程回目标器件。将保留的程序存储器的范围由下面的 <b>Preserve Program Memory Range(s)</b> （保留程序存储器范围）选项确定。确保代码不受代码保护。  |
| Preserve Program Memory Range(s) (hex)（保留程序存储器范围（十六进制））          | 要保留的程序存储器的范围。每个范围必须是两个十六进制数（表示范围的起始地址和结束地址），这两个数用短划线隔开。多个范围之间必须用逗号隔开（例如 0-ff, 200-2ff）。要保留区域的保留方式是将其读入 MPLAB X IDE，然后在进行编程操作时将其编程回原来的区域。因此，保留区域必须位于将被编程的存储器范围内。  |
| Preserve (Type of) Memory（保留（类型）存储器）                             | 使能该项将使器件上的当前存储器类型读入 MPLAB X IDE 的存储区，然后在编程完成后重新编程回目标器件。选中时会保留 <b>Memory</b> 以对其进行重新编程，其中 <b>Memory</b> 为存储器类型。类型包括：EEPROM 存储器、ID 存储器、引导闪存和辅助存储器。确保代码不受代码保护。  |
| Preserve (Type of) Memory Range(s) (hex)（保留（类型）存储器范围（十六进制））<br>* | 要保留的存储器类型的范围。每个范围必须是两个十六进制数（表示范围的起始地址和结束地址），这两个数用短划线隔开。多个范围之间必须用逗号隔开（例如 0-ff, 200-2ff）。要保留区域的保留方式是将其读入 MPLAB X IDE，然后在进行编程操作时将其编程回原来的区域。因此，保留区域必须位于将被编程的存储器范围内。 <b>Memory</b> 为存储器类型，包括 EEPROM 存储器、ID 存储器、引导闪存和辅助存储器。确保代码不受代码保护。 |
| *如果您收到因不正确范围引起的编程错误，确保范围未超过可用/剩余器件存储器的范围。                        |  |

## 9.2.2. Debug Options（调试选项）

您可以选择使用软件断点（如果此选项可用于项目所使用器件）。

**表 9-2. Debug Options 选项类别**

|                          |               |
|--------------------------|---------------|
| Debug startup（调试启动）      | 在器件启动后开始调试会话。 |
| Debug reset（调试复位）        | 在复位后开始调试会话。   |
| Use Software Breakpoints | 使用软件断点。       |

**表 9-3. 软件断点与硬件断点**

| 特性             | 软件断点              | 硬件断点  |
|----------------|-------------------|-------|
| 断点数            | 无限                | 有限    |
| 断点写入到          | 程序存储器             | 调试寄存器 |
| 设置断点的时间        | 取决于振荡器速度——可能需要数分钟 | 极短    |
| 断点滑移（Skidding） | 不支持               | 支持    |

表 9-3. 软件断点与硬件断点（续）

| 特性   | 软件断点 | 硬件断点 |
|--|------|------|
| <b>注：</b> 使用软件断点进行调试会影响器件的耐用性。因此，建议不要将以这种方式使用过的器件用作生产器件。 |      |      |

### 9.2.3. Program Options

选择在编程前擦除整个存储器还是合并代码。此外，还需专门为 PICkit 5 工具设置选项。

表 9-4. Program Options 选项类别

|   |   |
|---|---|
| Erase All Before Program                                | 使能该选项将导致在从 MPLAB X IDE 编程数据之前擦除整个器件。编程器件时，任何指定为保留的存储区都将在器件被擦除前读取并重新编程到器件。除非编程新器件或已擦除的器件，否则都必须选中此复选框。若未选中此复选框，将不对器件执行擦除操作并且程序代码将与器件中已存在的代码合并。  |
| Programming mode entry（编程模式进入）                          | 该选项指定调试器用于将目标器件置于编程模式的方法。对于低电压编程方法， $V_{PP}$ 不会超过 $V_{DD}$ 电源电压，而会对 $V_{PP}$ 使用测试模式。对于高电压编程方法，将对 $V_{PP}$ 施加超过 9V 的电压。<br><b>注：</b> 高电压编程需要 $V_{DD}$ 高于 2.8V。如果目标电压低于 2.8V，则选择低电压编程。<br>高电压编程模式——2.8V 至 5.0V。<br>仅低电压编程模式——1.2V 至 5.0V。 |
| Programming Method（编程方法）                                | 先施加 $V_{DD}$ 再施加 $V_{PP}$<br>先施加 $V_{PP}$ 再施加 $V_{DD}$  |
| LED Brightness Setting（LED 亮度设置）                        | 选择从 1（最暗）到 10（最亮）的亮度等级。   |
| PGC Configuration（PGC 配置）                               | 此选项确定将应用于 PGC 线的电阻类型（下拉、上拉或无）。默认为下拉。电阻值由下面的 PGC 电阻值选项决定。  |
| PGC resistor value (k $\Omega$ )（PGC 电阻值 (k $\Omega$ )） | 输入 0-50 之间的电阻值。默认值为 4.7 k $\Omega$ 。如果 PGC 配置设置为 none（无），则忽略该值。   |
| PGD Configuration（PGD 配置）                               | 选择 none、pull up（上拉）或 pull down（下拉）。默认为下拉。电阻值由下面的 PGD 电阻值选项决定。   |
| PGD resistor value (k $\Omega$ )（PGD 电阻值 (k $\Omega$ )） | 输入 0-50 之间的电阻值。默认值为 4.7 k $\Omega$ 。如果 PGD 配置设置为 none，则忽略该值。  |
| Program Speed   | 选择调试器编程目标器件时所使用的速度，即 Low（低速）、Normal 或 High（高速）。默认为 Normal。如果编程失败，则使用低速可能会解决问题。  |

### 9.2.4. Bluetooth Options（蓝牙选项）

选择使能或禁止 PICkit 5 使用的蓝牙无线技术通信。

表 9-5. Bluetooth Options 选项类别

|                    |           |
|--------------------|-----------|
| On（开启）             | 使能蓝牙无线技术。 |
| Off（关闭）            | 禁止蓝牙无线技术。 |
| Do not change（不更改） | 不更改默认设置。  |

### 9.2.5. Freeze Peripherals（冻结外设）

从外设列表中选择程序暂停时冻结或不冻结的外设。可用外设取决于器件。

#### PIC12/16/18 MCU 器件

要在暂停时冻结/取消冻结所有器件外设，请选中/取消选中 **Freeze on Halt** 复选框。如果这并未使所需外设暂停，请注意，某些外设没有暂停时冻结功能，因而无法受调试器控制。

#### dsPIC、PIC24 和 PIC32 器件

选中 **Peripherals to Freeze on Halt**（暂停时冻结的外设）列表中某外设的复选框，可在暂停时将其冻结。取消选中外设可使其在程序暂停时运行。如果未在列表中看到某个外设，则选中 **All Other**

**Peripherals**（所有其他外设）。如果这并未使所需外设暂停，请注意，某些外设没有暂停时冻结功能，因而无法受调试器控制。

要选择包括 **All Other Peripherals** 在内的所有外设，则单击 **Check All**（全部选中）。要取消选择包括 **All Other Peripherals** 在内的所有外设，则单击 **Uncheck All**（全部取消选中）。

### 9.2.6. Power

选择电源选项。

表 9-6. Power 选项类别

|  |  |
|--|--|
| Power Target Circuit from PICKit™ 5 (通过 PICKit™ 5 为目标电路供电) | 如果选中该选项，则将允许通过 PICKit 5 为目标电路供电。否则必须使用外部电源（见目标板由调试器供电）。                    |
| Voltage Level  | 如果选中“Power Target Circuit from PICKit 5”复选框，则可选择调试器提供的目标 V <sub>DD</sub> 。 |

### 9.2.7. Programmer-To-Go

选择 Programmer-To-Go 选项。

表 9-7. Programmer-To-Go 选项类别

|                    |  |
|--------------------|--|
| Image Name         | 默认映像名称为“<your project name>_ptg”，但也可根据需要编辑名称。这将是 microSDHC 卡上的文件夹名称，其中包含脱机编程的相关文件。 |
| Send image to tool | 该复选框默认处于选中状态。选中该复选框后，将创建 PTG 映像，然后将其发送到所连接的 MPLAB® PICKit™ 5 中插入的 microSDHC 卡。     |
| Program Device     | 该复选框默认处于选中状态。选中该复选框后，将对连接到 MPLAB PICKit 5 的器件进行编程。                                 |

### 9.2.8. Secure Segment（安全段）

选择和加载调试器固件。

表 9-8. Secure Segment 选项类别

|                                  |   |
|----------------------------------|---|
| Segments to be Programmed（要编程的段） | 选择以下其中一个选项： <ol style="list-style-type: none"> <li>Full Chip Programming（全片编程）（默认）。</li> <li>Boot, Secure and General Segments（引导段、安全段和通用段）。</li> <li>Secure and General Segments（安全段和通用段）。</li> <li>General Segment Only（仅通用段）。</li> </ol> |
|----------------------------------|---|

### 9.2.9. Clock（时钟）

设置此选项可为所选器件使用内部快速 RC（Fast RC，FRC）时钟。

表 9-9. Clock 选项类别

|   |   |
|---|---|
| Use FRC in Debug mode (dsPIC33F and PIC24F/H devices only)<br>(在调试模式下使用 FRC（仅限 dsPIC33F 和 PIC24F/H 器件）) | 调试时，使用器件内部快速 RC（FRC）而不是为应用指定的振荡器来提供时钟。这在应用时钟较慢时十分有用。选中此复选框将使应用以低速运行，但以更快的 FRC 速度进行调试。更改此设置后重新编程。<br><b>注：</b> 在调试时，未冻结的外设将以 FRC 速度运行。 |
|---|---|

### 9.2.10. Tool Pack Selection（工具包选择）

选择和加载调试器固件。

表 9-10. Tool Pack Selection 类别

|   |   |
|---|---|
| Tool pack update options (工具包更新选项)      | 选择 Use Latest installed tool pack (recommended) (使用最新安装的工具包 (推荐)) 或 Use specific tool pack (使用特定工具包)。 |
| Specifically selected version (专门选择的版本) | 单击即可选择要使用的工具包。单击后, 将打开 Select Tool pack (选择工具包) 对话框, 可从中选择所需的版本。                                      |

### 9.2.11. Communication (通信)

设置适用于您的器件和目标通信类型的选项。

表 9-11. Communication 选项类别

|  |  |
|--|--|
| Interface (接口)                         | 根据项目中的器件从可用选项中选择接口。  |
| Speed (MHz) (速度 (MHz))                 | 根据接口适用的速度范围输入速度。   |
| High Voltage Activation Mode (高电压激活模式) | 该选项仅针对具有该选项的 AVR <sup>®</sup> 器件显示。 <b>No High Voltage</b> (无高电压) —— 默认设置。 <b>Simple High Voltage Pulse</b> (简单高电压脉冲) —— 该工具将尝试通过发出高电压脉冲来激活接口。如果引脚配置为输入, 则该过程是安全的。 <b>User Power Toggle</b> (用户电源重启) —— 在该模式下, 系统将提示用户重启目标板的电源。一旦工具检测到电源恢复, 它将在配置目标器件引脚之前发出高电压脉冲, 从而使激活过程尽可能平稳。另请参见 <a href="#">UPDI 高电压激活信息</a> 。 |

### 9.2.12. Event Recorder (事件记录器)

指定事件记录器的选项。有关事件记录器的更多信息, 请参见《[MPLAB<sup>®</sup> X IDE 用户指南](#)》(DS50002027E\_CN) 或 [MPLAB X IDE 在线帮助](#)。

表 9-12. Event Recorder 选项类别

|                      |                   |
|----------------------|-------------------|
| Enable (使能)          | 选中时会使能事件记录器       |
| SCVD Files (SCVD 文件) | 指定项目中要使用的 SCVD 文件 |

## 10. 硬件规范

本章详述了 MPLAB PICKit 5 在线调试器系统的硬件规范和电气规范。

### 10.1. USB 连接器规范

MPLAB PICKit 5 在线调试器通过符合 USB 2.0 版本的 USB Type-C<sup>®</sup> 连接器与主机计算机连接。USB 连接器位于调试器的顶部。

系统能够通过 USB 接口重载固件。

通过 USB 接口为系统供电。根据 USB 规范，该调试器被归类为大功率系统，需要通过 USB 提供略大于 50 mA 的电流，才能使该调试器在所有工作模式（调试器/编程器）下正常工作。

**注：**MPLAB PICKit 5 在线调试器通过其 USB Type-C 连接器供电。目标板由自身电源供电。或者，也可由调试器为目标板供电，但仅限目标板消耗的电流小于 150 mA 时。

**线缆长度**——在调试器工具包里随附了正常工作所需长度的计算机-调试器连接线缆。

**供电的集线器**——如果您想使用 USB 集线器，请确保它是自供电的。此外，计算机键盘上的 USB 端口不能为调试器正常工作提供足够的功率。

**计算机休眠/掉电模式**——禁止计算机上的休眠或其他节能模式，以确保与调试器的正常 USB 通信。

### 10.2. MPLAB PICKit 5 在线调试器

调试器包含一个内部主电路板、一个外部 USB Type-C 连接器和一个 8 引脚 SIL 连接器。调试器外壳的正面有一个指示灯带和一个隐藏按钮，按钮位于徽标下面。

图 10-1. MPLAB® PICKIT™ 5 在线调试器



1. 挂绳连接——穿过顶部和侧面的开孔，用于连接挂绳（不提供挂绳）。
2. 紧急恢复按钮——如果需要，此内凹按钮可用于恢复引导模式。
3. USB Type-C 连接器——用于使用随附的 USB 线缆将调试器连接到计算机。
4. 指示灯带——显示调试器的工作模式（见 [指示灯带](#)）。
5. 按钮区域——盾形徽标中心的区域，用于脱机编程选项。
6. 引脚 1 标记——指示引脚 1 的位置，以便正确对齐连接器。
7. 编程连接器——该连接器是一个 8 引脚 SIL 连接器（0.100"间距），用于连接目标器件（见 [目标连接引脚分配](#)）。
8. MicroSDHC 卡插槽——microSDHC 卡插槽支持具有各种速度要求的多种 microSDHC 卡。

### 10.2.1. 电路板规范

主电路板具有以下特性：

- 使用 Arm Cortex-M7 内核的 32 位单片机，其中包括用于存储程序代码映像的存储器。此映像用于编程板上闪存器件。
- 一个 USB 速度最高达 480 Mbps 的 USB 2.0 接口。
- RN4871 蓝牙低功耗模块——蓝牙 v5.0 @ 2.4 GHz。
- 一个 LED 灯带。

## 10.2.2. 指示灯带

MPLAB PICkit 5 调试器预期的启动序列为：

1. 紫色——常亮约 4 秒钟。
2. 蓝色——常亮。调试器就绪。

指示灯带具有如下含义。

**表 10-1.** 指示灯带的典型状态说明

| 颜色 | 说明                                      |
|----|---|
| 蓝色 | 电源已连接；调试器处于待机状态。                        |
| 橙色 | 已选中 Power target circuit from PICkit 5。 |
| 绿色 | 未选中 Power target circuit from PICkit 5。 |
| 红色 | 调试器故障时点亮。                               |

下面的两个表列出了指示灯和自举程序错误的说明。

**表 10-2.** 指示灯带的附加状态说明

| 颜色 | 说明  |
|----|---|
| 蓝色 | 电源已连接；调试器处于待机状态。                                |
| 橙色 | 已选中 Power target circuit from PICkit 5（见表 9-6）。 |
| 绿色 | 未选中 Power target circuit from PICkit 5（见表 9-6）。 |
| 紫色 | 自举程序正在运行。                                       |
| 黄色 | 调试器忙。   |
| 红色 | 操作失败。   |
| 紫色 | 快速闪烁表示强制调试器进入恢复引导模式的时间窗口。                       |

**表 10-3.** 自举程序错误说明

| 自举程序错误  | 说明                   |
|---------|----------------------|
| 红色，慢速闪烁 | 访问调试器的串行 EEPROM 时上电。 |
| 红色，快速闪烁 | 无法处理自举程序 API 命令。     |
| 白色，快速闪烁 | 工具固件出现运行时异常。         |

### 相关信息

[使用脱机编程](#)

## 10.3. 通信硬件

对于调试器与目标板的标准通信，可以直接将调试器连接到目标板。调试器有一个 8 引脚 SIL 连接器。如果目标板具有 6 引脚连接器，请确保正确对齐引脚 1。对于其他连接，可以选择使用适配器板。

此外，还内置蓝牙硬件以支持与移动应用程序通信，从而进行 PTG 代码编程。

### 10.3.1. 标准通信

调试器与目标处理器的接口采用标准通信方式。它包含与高电压（ $V_{PP}$ ）线和  $V_{DD}$  检测线的连接，以及编程和连接目标器件所需的时钟和数据连接。

$V_{PP}$  高电压线可生成电压范围为 0V 至 14V 的可变电压，以满足特定仿真处理器的电压要求。

$V_{DD}$  检测连接从目标处理器汲取极少的电流。实际功率来自 MPLAB PICkit 5 在线调试器系统，因为  $V_{DD}$  检测线仅用作参考来跟踪目标电压。

时钟和数据连接是具有以下特性的接口：

- 时钟和数据信号为高阻模式（即使在未向 MPLAB PICkit 5 在线调试器系统供电时也是如此）。

- 时钟和数据信号不受因故障目标系统或不当连接引起的高电压的影响。
- 时钟和数据信号不受因故障目标系统中出现短路而引起的大电流的影响。

表 10-4. 电气逻辑表

|      |                                     |                        |                       |                       |
|------|-------------------------------------|------------------------|-----------------------|-----------------------|
| 逻辑输入 | $V_{IH} = V_{DD} \times 0.7V$ (最小值) |                        |                       |                       |
|      | $V_{IL} = V_{DD} \times 0.3V$ (最大值) |                        |                       |                       |
| 逻辑输出 | $V_{DD} = 5V$                       | $V_{DD} = 3V$          | $V_{DD} = 2.3V$       | $V_{DD} = 1.4V$       |
|      | $V_{OH} = 3.8V$ (最小值)               | $V_{OH} = 2.4V$ (最小值)  | $V_{OH} = 1.9V$ (最小值) | $V_{OH} = 1.0V$ (最小值) |
|      | $V_{OL} = 0.55V$ (最大值)              | $V_{OL} = 0.55V$ (最大值) | $V_{OL} = 0.3V$ (最大值) | $V_{OL} = 0.1V$ (最大值) |

### 相关信息

[目标板连接引脚分配](#)

## 10.3.2. 其他通信

为了支持其他通信连接，可以使用适配器板。

图 10-2. 调试器适配器板

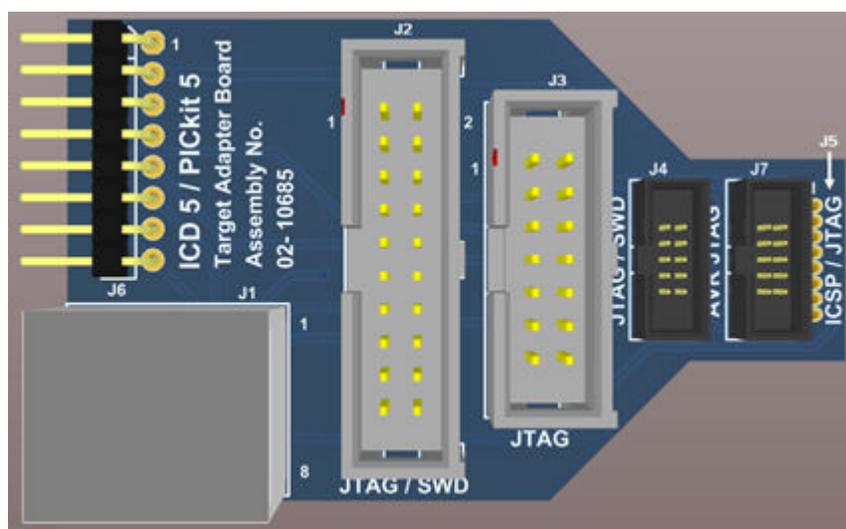


表 10-5. 调试器适配器板——目标板连接

| 连接器 | 在适配器板上的位置 | 说明                        | 兼容性  |
|-----|-----------|---------------------------|--|
| J1  | 顶部        | RJ45/11                   | MPLAB® ICD 4/5 ICSP™/JTAG                      |
| J6  | 顶部        | 1 x 8——100 mil ICSP       | PICKIT 4/5 和 Snap ICSP                         |
| J2  | 顶部        | 2 x 10——100 mil DIP       | SEGGER JLink JTAG SWD                          |
| J3  | 顶部        | 2 x 7——100 mil DIP        | SEGGER JLink EJTAG                             |
| J4  | 顶部        | 2 x 5——50 mil             | Atmel-ICE JTAG SWD                             |
| J7  | 顶部        | 2 x 5——50 mil             | Atmel-ICE AVR® JTAG                            |
| J5  | 底部        | 1 x 6 或 1 x 8——50 mil SIP | MPLAB ICD 4/5 ICSP/JTAG、PICKIT 4/5 和 Snap ICSP |

### 相关信息

[适配器板引脚分配](#)

## 10.4. 目标板注意事项

应根据所选器件和应用的要求向目标板供电。

**注：**如果器件的工作条件超过器件数据手册“电气特性”一章中的“绝对最大值”，可能对器件造成永久性损坏。上述值仅为运行条件最大值，我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其可靠性可能受到影响。

调试器检测目标电压。

根据调试器与目标板之间使用的通信类型，目标板电路有以下注意事项：

- [ICSP 目标板连接电路](#)
- [阻止仿真器正常工作的电路](#)
- [关于高电压](#)

目标板电路的额外注意事项：

- [开发工具设计忠告](#)

## 11. 版本历史

下文列出了本文档不同版本中的内容更改。

**注：**某些版本字母（字母 I 和 O）在使用某些字体时可能会与数字发生混淆，因此不会使用。

### 版本 C（2025 年 3 月）

- **5.6 使用 BLE 设备上的 PTG 应用程序：**增加了有关使能蓝牙功能的注释。
- **5.6.1 打开脱机编程应用程序：**增加了有关使能蓝牙功能的内容。
- **6.4 如何调用自举模式：**删除了本节及其描述的功能。有关详细信息，请参见“MPLAB PICKit 5 的发行说明”（MPLAB X IDE Help > Release Notes）。
- **9.2.4 蓝牙选项：**增加了有关“选择使能或禁止 PICKit 5 使用的蓝牙无线技术通信”的章节。

### 版本 B（2024 年 8 月）

- **3.3.2.1 适配器板引脚分配：**更新了引脚图说明。
- **8.2.6 调试失败纠正措施：**本节增加了更多信息。
- 其他少量更新。

### 版本 A（2023 年 4 月）

本文档的初始版本。

## 12. 支持

### 12.1. 保修登记

请访问 [www.microchip.com/mysoftware](http://www.microchip.com/mysoftware) 在线注册工具。如果您尚未拥有 myMicrochip 帐户，可以单击该链接进行注册。如果您已经拥有一个帐户，请登录并单击 **Register Hardware Tool**（注册硬件工具）。

在线注册工具后，您便可以收到新产品更新信息。可在 Microchip 网站上获得临时软件版本。

### 12.2. myMicrochip 个性化通知服务

Microchip 的个性化通知服务有助于客户了解关于所关注 Microchip 产品的最新信息。注册客户可在指定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。

如需注册并选择接收个性化通知的首选项，请转至：

[www.microchip.com/pcn](http://www.microchip.com/pcn)

该网页提供 FAQ 和注册详细信息。

## 13. 术语表

### Absolute Section（绝对段）

具有链接器不能改变的固定（绝对）地址的 GCC 编译器段。

### Absolute Variable/Function（绝对变量/函数）

使用 OCG 编译器的 @ address 语法放置在绝对地址的变量或函数。

### Access Memory（快速存取存储区）

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（Bank Select Register, BSR）的设置无关。

### Access Entry Point（访问入口点）

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段和安全应用程序段。

### Address（地址）

标识存储器中位置的值。

### Alphabetic Character（字母字符）

字母字符指属于拉丁字母表（a,b, ..., z,A,B, ..., Z）中字母的字符。

### Alphanumeric（字母数字字符）

字母数字字符由字母字符和十进制数字（0,1, ...,9）组成。

### ANDed Breakpoint（逻辑与断点）

为程序暂停设置“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。这只有在数据断点和程序存储器断点同时发生时才能实现。

### Anonymous Structure（匿名结构体）

16 位 C 编译器——未命名的结构体。

PIC18 C 编译器——属于 C 联合体的成员的未命名结构体。匿名结构体成员可以像包含结构体的联合体的成员一样进行访问。例如，在以下代码中，hi 和 lo 是联合体 caster 中的匿名结构体的成员。

```
union castaway
int intval;
struct {
char lo; //accessible as caster.lo
char hi; //accessible as caster.hi
};
} caster;
```

### ANSI

美国国家标准学会，是美国负责制订和批准标准的组织。

### Application（应用）

可由 PIC 单片机控制的一组软硬件。

### Archive/Archiver（归档/归档器）

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。

### 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。

**Assembly/Assembler (汇编语言/汇编器)**

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。

**Assigned Section (已分配段)**

在链接器命令文件中已分配到目标存储区的 GCC 编译器段。

**Asynchronously (异步)**

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。

**Asynchronous Stimulus (异步激励)**

为模拟传递到软件模拟器件的外部输入而生成的数据。

**Attribute (属性)**

C 语言程序中变量或函数的 GCC 特性，用于描述特定于机器的性质。

**Attribute, Section (属性, 段)**

段的 GCC 特性，如“可执行”、“只读”或“数据”，它们可在汇编器 `.section` 伪指令中指定为标志。

**AVR MCU**

AVR 单片机 (MCU) 指 Microchip 的所有 8 位 AVR 单片机系列。

**Binary (二进制)**

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。

**Bookmarks (书签)**

使用书签可轻松地查找文件中的指定行。

在 Editor 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。

**C/C++**

C 语言是一种通用编程语言，具有简练的表达式、现代控制流和数据结构，以及丰富的操作符。C++ 是 C 语言的面向对象版本。

**Calibration Memory (校准存储区)**

用于保存 PIC 单片机内 RC 振荡器或其他外设校准值的特殊功能寄存器或通用寄存器。

**Central Processing Unit (中央处理单元)**

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。

**Clean (清除)**

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。

**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。

**Command Line Interface (命令行接口)**

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。

**Compiled Stack（编译堆栈）**

编译器管理的存储区，用于静态分配变量的空间。当在目标器件上无法高效实现软件或硬件堆栈时，编译堆栈将替代它们。

**Compiler（编译器）**

将用高级语言编写的源文件翻译成机器码的程序。

**Conditional Assembly（条件汇编）**

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。

**Conditional Compilation（条件编译）**

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。

**Configuration Bits（配置位）**

可对其编程来设置 PIC MCU 或 dsPIC DSC 工作模式的专用位。配置位可或不可再编程。

**Constant（常量）**

表示一个立即数的值，例如通过 C 代码 `#define` 伪指令或汇编 `.equ` 伪指令定义。

**Control Directive（控制伪指令）**

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。

**CPU**

请参见 *中央处理单元*。

**Cross Reference File（交叉引用文件）**

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，列出的第一个文件是定义的位置。其他文件包含对符号的引用。

**Data Directive（数据伪指令）**

数据伪指令是那些控制编译器对程序存储器或数据存储器进行分配的指示性语句，它提供了用符号（即有意义的名称）引用数据项的方法。

**Data Memory（数据存储器）**

在 Microchip MCU 和 DSC 器件中，数据存储器（RAM）由通用寄存器（General Purpose Register, GPR）和特殊功能寄存器（Special Function Register, SFR）组成。某些器件还有 EEPROM 数据存储器。

**Debug/Debugger（调试/调试器）**

请参见 *ICE/ICD*。

**Debugging Information（调试信息）**

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。

**Deprecated Feature（弃用功能）**

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。

**Device Programmer（器件编程器）**

用于对电可编程半导体器件（如单片机）进行编程的工具。

**Digital Signal Processing/Digital Signal Processor（数字信号处理/数字信号处理器）**

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。

**Directive (伪指令)**

源代码中提供对语言工具的操作进行控制的语句。

**Download (下载)**

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。

**dsPIC DSC**

dsPIC 数字信号控制器（DSC）指具有数字信号处理能力的 Microchip 单片机系列。

**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。

**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数据。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。

**Emulation/Emulator (仿真/仿真器)**

请参见 *ICE/ICD*。

**Endianness (尾数法)**

多字节对象中的字节存储顺序。

**Environment (环境)**

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。

**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、紧接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。

**Error/Error File (错误/错误文件)**

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。

**Event (事件)**

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）及时间戳的总线周期的描述。事件用于描述触发、断点和中断。

**Executable Code (可执行代码)**

可装入来执行的软件。

**Export (导出)**

以标准化的格式将数据从 MPLAB X IDE 发送出。

**Expressions (表达式)**

用算术或逻辑运算符分隔开的常量和/或符号的组合。

**Extended Microcontroller Mode (扩展单片机模式)**

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。

**Extended Mode (扩展模式) (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。

**External Label (外部标号)**

具有外部链接的标号。

**External Linkage (外部链接)**

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。

**External Symbol (外部符号)**

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。

**External Symbol Resolution (外部符号解析)**

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。

**External Input Line (外部输入线)**

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)。

**External RAM (外部 RAM)**

芯片外部的读/写存储器。

**Fatal Error (致命错误)**

引起编译立即停止的错误。不产生其他消息。

**File Register (文件寄存器)**

片内数据存储寄存器，包括通用寄存器 (GPR) 和特殊功能寄存器 (SFR)。

**Filter (筛选器)**

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。

**Fixup (修正)**

在链接器重定位之后，使用绝对地址替换目标文件符号引用的过程。

**Flash (闪存)**

按块（而不是按字节）写或擦除数据的一种 EEPROM。

**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。

**Frame Pointer (帧指针)**

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。

## Free-Standing（独立）

一种接受任何不使用复杂类型的严格符合程序的实现，而且在这种实现中，对库条款（ANSI 89 标准条款第 7 条）中规定的特性的使用，仅限于标准头文件<float.h>、<iso646.h>、<limits.h>、<stdarg.h>、<stdbool.h>、<stddef.h>和<stdint.h>的内容。

## GPR

通用寄存器。器件数据存储器（RAM）的一部分，作为一般用途。

## Halt（暂停）

停止程序执行。执行 Halt 与在断点处停止相同。

## Heap（堆）

用于动态存储器分配的存储区，其中的存储区块按运行时确定的任意顺序进行分配和释放。

## Hex Code/Hex File（Hex 代码/Hex 文件）

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。

## Hexadecimal（十六进制）

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 10-15（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。

## High Level Language（高级语言）

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。

## 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/调试头板上，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。使用串行通信和最少数量的器件引脚对 Microchip 嵌入式器件进行编程的方法。

## IDE

集成开发环境，如 MPLAB X IDE。

## Identifier（标识符）

函数名或变量名。

## IEEE

电气电子工程师协会。

## Import（导入）

将数据从外部源（如 hex 文件）传送到 MPLAB X IDE 中。

## Initialized Data（已初始化数据）

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。

**Instruction Set（指令集）**

特定处理器理解的机器语言指令的集合。

**Instructions（指令）**

告知中央处理单元执行特定操作、并包含操作中要使用的数据的位序列。

**Internal Linkage（内部链接）**

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。

**International Organization for Standardization（国际标准化组织）**

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。

**Interrupt（中断）**

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。

**Interrupt Handler（中断处理程序）**

发生中断时处理特殊代码的子程序。

**Interrupt Service Request, IRQ（中断服务请求）**

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。

**Interrupt Service Routine, ISR（中断服务程序）**

*语言工具*：处理中断的函数。

*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。

**Interrupt Vector（中断向量）**

中断服务程序或中断处理程序的地址。

**L-value（左值）**

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。

**Latency（响应延时）**

事件与其得到响应之间的延迟时间。

**Library/Librarian（库/库管理器）**

请参见 *归档/归档器*。

**Linker（链接器）**

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。

**Linker Script File（链接描述文件）**

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。

**Listing Directive（列表伪指令）**

列表伪指令是控制汇编器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。

**Listing File（列表文件）**

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。

**Little Endian（小尾数法）**

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。

**Local Label（局部标号）**

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实例。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。

**Machine Code（机器码）**

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。某个特定处理器的所有可用指令的集合称为它的“指令集”。

**Machine Language（机器语言）**

特定中央处理单元的指令集，不需翻译即可供处理器使用。

**Macro（宏）**

宏指令。以缩写形式表示指令序列的指令。

**Macro Directive（宏伪指令）**

控制宏定义体中执行和数据分配的伪指令。

**Makefile**

将用于 Make 项目的指令导出到一个文件中。可使用该文件通过 make 指令在 MPLAB X IDE 外 Make 项目。

**Make Project（Make 项目）**

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。

**Memory Model（存储器模型）**

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。

**Message（消息）**

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。

**Microcontroller（单片机）**

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。

**Microcontroller Mode（单片机模式）**

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。

**Microprocessor Mode（微处理器模式）**

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。

**Mnemonic（助记符）**

可直接翻译成机器码的文本指令。也称为操作码。

**Module（模块）**

源文件在执行预处理器伪指令之后的预处理输出。也称为翻译单元。

**MPLAB® X IDE**

Microchip 的集成开发环境。MPLAB X IDE 随附编辑器、项目管理器和软件模拟器。

**MPLAB X Simulator (MPLAB X 软件模拟器)**

Microchip 的软件模拟器，支持 Microchip MCU、DSC 和 MPU 器件。

**MPLAB XC C Compilers (MPLAB XC C 编译器)**

Microchip 的 C 和 C++ 编译器系列由 MPLAB XC8 C 编译器（支持 8 位 PIC 和 AVR 器件）、MPLAB XC16 C 编译器（支持 16 位 PIC 器件）、MPLAB XC-DSC C 编译器（支持 DSC 器件）和 MPLAB XC32 C/C++ 编译器（支持 32 位 PIC 和 SAM 器件）组成。

**MPLAB Xpress IDE**

Microchip 的云端集成开发环境。MPLAB Xpress 随附编辑器、项目管理器和软件模拟器。

**MPU**

微处理器。microprocessor 的缩写形式。

**MRU**

最近使用的。指可从主下拉菜单选择的文件和窗口。

**Native Data Size (本机数据大小)**

对于本地跟踪，**Watches**（观察）窗口中使用的变量长度必须与所选器件的数据存储器长度相同：对于 PIC18 器件，为字节长度；对于 16 位器件，为字长度。

**Nesting Depth (嵌套深度)**

宏可包含其他宏的最大深度。

**Node (节点)**

项目组件。

**Non-Extended Mode (非扩展模式) (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。

**Non Real Time (非实时)**

指处理器执行到断点或单步执行指令，或运行在软件模拟器模式。

**Non-Volatile Storage (非易失性存储器)**

电源关闭时保存其内容的存储器件。

**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。

**Object Code/Object File (目标代码/目标文件)**

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。

**Object File Directive (目标文件伪指令)**

仅当生成目标文件时使用的伪指令。

**Octal (八进制)**

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。

## Off-Chip Memory（片外存储器）

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**（选项 > 开发模式）访问 **Memory（存储器）** 选项卡可打开 Off-Chip Memory Selection（片外存储器选择）对话框。

## Opcode（操作码）

操作码。请参见 *助记符*。

## Operator（运算符）

加号“+”和减号“-”之类的符号，它们在构成定义明确的表达式时使用。每个运算符都具有指定的优先级，用于确定求值的顺序。

## 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。

## Pass Counter（次数计数器）

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）对话框中的任何顺序执行事件。

## PC

个人计算机或程序计数器。

## PC Host（PC 主机）

运行支持的 Windows 操作系统的任何 PC。

## Persistent Data（持久性数据）

永不清除或初始化的数据。其作用是使应用程序可以在器件复位时保存数据。

## Phantom Byte（虚拟字节）

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。

## PIC MCU

PIC 单片机（MCU）指 Microchip 的所有 8 位、16 位和 32 位 PIC 单片机系列。

## Plug-in（插件）

MPLAB IDE/MPLAB X IDE 使用内置组件和插件模块来为各种软件和硬件工具配置系统。可在 Tools 菜单下找到几个插件工具。

## Pod（主机）

在线仿真器或调试器的外壳。其他名称还有 *Puck*（如果外壳是圆的）和 *Probe*（不要与逻辑探针混淆）。

## Power-on-Reset Emulation（上电复位仿真）

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。

## Pragma（Pragma 伪指令）

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。

## Precedence（优先顺序）

定义表达式中求值顺序的规则。

**Production Programmer (生产编程器)**

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。

**Profile (配置文件)**

对于 MPLAB X 软件模拟器，寄存器已执行激励的汇总列表。

**Program Counter (程序计数器)**

包含正在执行的指令的地址的存储单元。

**Program Counter Unit (程序计数器单元)**

16 位汇编器——程序存储器布局的概念化表示。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。

**Program Memory (程序存储器)**

器件中存储指令的存储区。亦指调试器、仿真器或软件模拟器中包含下载的目标应用固件的存储器。

**Project (项目)**

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。

**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。

**Prototype System (原型系统)**

指用户的目标应用或目标板的术语。

**Psect**

GCC 段的 OCG 等效形式，program section（程序段）的缩写。被链接器视为一个整体的代码块或数据块。

**PWM Signal (PWM 信号)**

脉宽调制信号。某些 PIC MCU 包含 PWM 外设。

**Qualifier (限定符)**

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。

**Radix (基数)**

数基，十六进制或十进制，用于指定一个地址。

**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。

**Raw Data (原始数据)**

与一个段有关的代码或数据的二进制表示。

**Read Only Memory (只读存储器)**

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。

## Real Time（实时）

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。

## Recursive Call（递归调用）

一个直接或间接调用自身的函数。

## Recursion（递归）

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。

## Re-entrant（可重入函数）

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。

## Relaxation（精简）

将某一指令转换为功能相同但大小较小的指令的过程。这对于缩短代码长度非常有用。MPLAB XC16 和 MPLAB XC-DSC 当前知道如何将 CALL 指令精简为 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。

## Relocatable（可重定位）

一个对象，它的地址没有分配到存储器中的固定地址。

## Relocatable Section（可重定位段）

16 位汇编器——地址不固定（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。

## Relocation（重定位）

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。

## ROM

只读存储器（程序存储器）。不能修改的存储器。

## Run（运行）

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。

## Run-time Model（运行时模型）

描述目标架构资源的使用。

## Run-time Watch（运行时观察）

**Watches** 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。

## SAM MCU/MPU

SAM 单片机（MCU）和微处理器（MPU）指 Microchip 的所有 32 位 SAM 单片机和微处理器系列。

## Scenario（场景）

MPLAB X 软件模拟器用于激励控制的一种特定设置。

## Section（段）

OCG psect 的 GCC 等效形式。被链接器视为一个整体的代码块或数据块。

## Section Attribute（段属性）

赋予段的 GCC 特性（例如，access 段）。

## Sequenced Breakpoint（顺序断点）

按在序列中的顺序发生的断点。序列中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。

## Serialized Quick Turn Programming（序列号快速批量编程）

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。

## 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。

## Simulator（软件模拟器）

模仿器件操作的软件程序。

## Single Step（单步执行）

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，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执行。也可单步执行 C 编译器源代码，但不是每次执行一条指令，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，被执行的操作码出现在总线上；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。

## Source Code（源代码）

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源代码，可翻译成机器码或被解释程序执行。

## Source File（源文件）

包含源代码的 ASCII 文本文件。

## Special Function Register, SFR（特殊功能寄存器）

数据存储寄存器（RAM）的一部分，专用于控制 I/O 处理器功能、I/O 状态、定时器或其他模式及外设的寄存器。

## SQTP

请参见 *Serialized Quick Turn Programming（序列号快速批量编程）*。

## Stack, Hardware（堆栈，硬件）

PIC 单片机中调用函数时存储返回地址的存储区。

## Stack, Software（堆栈，软件）

由应用程序用来存储返回地址、函数参数和局部变量的存储区。此存储区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。

### Stack, Compiled（堆栈，已编译）

编译器管理和分配的存储区域，用于静态分配变量的空间。当在目标器件上无法高效实现软件堆栈时，编译堆栈将替代软件堆栈。它会阻止重入。

### Static RAM or SRAM（静态 RAM 或 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。

### Status Bar（状态栏）

**Status Bar** 位于 **MPLAB X IDE** 窗口的底部，指示诸如光标位置、开发模式、器件和有效工具栏之类的当前信息。

### Step Into（单步进入）

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 Over（单步跳过）

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。

### Step Out（单步跳出）

Step Out 使您可以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。

### Stimulus（激励）

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。

### Stopwatch（跑表）

测量执行周期的计数器。

### Storage Class（存储类别）

确定与指定对象相关联存储区的存在时间。

### Storage Qualifier（存储限定符）

指明所声明对象的特殊属性（如 const）。

### Symbol（符号）

符号是描述组成程序的不同部分的一种通用机制。这些部分包括函数名、变量名、段名、文件名以及结构体/枚举/联合体标记名等。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。

### Symbol, Absolute（符号，绝对）

通过将符号置于存储器中的特定地址，可将其设为绝对符号（例如 `int scanMode __at(0x200);`）。

### System Window Control（系统窗口控件）

系统窗口控件位于窗口和某些对话框的左上角。单击该控件时通常会弹出包含“Minimize”（最小化）、“Maximize”（最大化）和“Close”（关闭）项的菜单。

### Target（目标）

指用户硬件。

### Target Application（目标应用程序）

目标板上的软件。

**Target Board（目标板）**

构成目标应用的电路和可编程器件。

**Target Processor（目标处理器）**

目标应用板上的单片机。

**Template（模板）**

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。

**Toolbar（工具栏）**

一行或一列图标，单击这些图标时将执行功能。

**Trace（跟踪）**

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中，该缓冲区的内容将被上传到 trace（跟踪）窗口。

**Trace Memory（跟踪存储区）**

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。

**Trace Macro（跟踪宏）**

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。

**Trigger Output（触发输出）**

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。

**Trigraph（三字母词）**

三字符序列，都以??开头，由 ISO C 定义用于替代单个字符。

**Unassigned Section（未分配段）**

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。

**Uninitialized Data（未初始化数据）**

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。

**Upload（上传）**

上传功能将数据从一个工具（如仿真器或编程器）传送到主机计算机，或将数据从目标板传送到仿真器。

**USB**

通用串行总线。一种外部外设接口标准，用于通过线缆使用双向串行传输在计算机和外部外设之间进行通信。Microchip 硬件工具当前支持的 USB 版本如下所示：

| USB     | 速度描述符 | 最大速度<br>(每秒兆位/千兆位) |
|---------|-------|--------------------|
| USB 2.0 | 高速    | 480 Mbps           |
| USB 3.0 | 超高速   | 5 Gbps             |

**Vector（向量）**

复位或中断发生时应用程序跳转到的存储地址。

**Volatile**

一个变量限定符，它可以防止编译器应用会影响变量在存储器中的访问方式的优化。

**Warning (警告)**

*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。

**Watch Variable (观察变量)**

调试会话期间可在 **Watches** 窗口中监控的变量。

**Watches Window (Watches 窗口)**

**Watches** 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。

**Watchdog Timer, WDT (看门狗定时器)**

PIC 单片机中在一段可选择长度的时间后复位处理器的定时器。使用配置位来使能、禁止和设置 WDT。

**Workbook (工作簿)**

对于 MPLAB X 软件模拟器，是一种用于产生 SCL 激励的设置。

# Microchip 信息

## 商标

“Microchip”的名称和徽标组合、“M”徽标及其他名称、徽标和品牌均为 Microchip Technology Incorporated 或其关联公司和/或子公司在美国和/或其他国家或地区的注册商标或商标（“Microchip 商标”）。有关 Microchip 商标的信息，可访问 <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>。

ISBN: 979-8-3371-1523-8

## 法律声明

提供本文档的中文版本仅为了便于理解。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。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他任何方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。您须自行确保应用符合您的规范。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 [www.microchip.com/en-us/support/design-help/client-support-services](http://www.microchip.com/en-us/support/design-help/client-support-services)。

Microchip “按原样”提供这些信息。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，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或附带的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。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损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

## Microchip 器件代码保护功能

请注意以下有关 Microchip 产品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规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》（Digital Millennium Copyright Act）。
- 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。