请注意赛普拉斯已正式并入英飞凌科技公司。
此封面页之后的文件标注有“赛普拉斯”的文件即该产品为此公司最初开发的。请注意作为英飞凌产品组合的部分，英飞凌将继续为新的及现有客户提供该产品。

文件内容的连续性
事实是英飞凌提供如下产品作为英飞凌产品组合的部分不会带来对于此文件的任何变更。未来的变更将在恰当的时候发生，且任何变更将在历史页面记录。

订购零件编号的连续性
英飞凌继续支持现有零件编号的使用。下单时请继续使用数据表中的订购零件编号。

PSoC® Creator™用户指南

文档编号：001-93523 版本*C

赛普拉斯半导体公司
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709
www.infineon.com

目录

1 PSoC Creator 介绍

PSoC Creator 帮助您对赛普拉斯 PSoC 器件的模拟外设和数字外设进行配置和编程。通过使用 PSoC Creator，您可以选择和放置组件、编写 C/汇编源码，并调试和烧写项目/器件。与相关的硬件一起使用时，通过这个动态的软件-硬件组合，您可以在硬件环境下测试项目，同时在软件环境下检查和调试器件。

注意：本文档选择了围绕 PSoC 4 器件（还附加了其他器件）对 PSoC Creator 的功能进行讲解。这里提到的 PSoC 4 指的是 PSoC 4 和 PSoC 4 BLE（蓝牙低能耗）器件。

这份 PSoC Creator 指南主要包括以下章节的内容：

入门	指导您如何开始使用 PSoC Creator。
了解 PSoC Creator	提供了加深了解 PSoC Creator 的更多信息和用法。
使用设计输入工具	图形设计输入工具的任务管理和接口说明。
PSoC Creator 项目编译	讲解如何配置和编译 PSoC Creator 项目。
将设计移植到第三方 IDE	讲解如何将 PSoC Creator 设计快速移植到第三方 IDE (Keil µVision, IAR, Eclipse)。
调试器的使用	讲解如何使用调试器。
完成项目	讲解设计后期如何在 PSoC Creator 中完成一个项目。
参考文档	第三方工具链的文档和其他参考文档。

修订记录

文档标题：Doc. No. 001-93523 Rev*B PSoC® Creator™用户指南 **文档编号：001-93523**		
版本	**日期**	**变更说明**
**	09/03/2014	本文档版本号为 Rev**，译自英文版 001-93417 Rev*A。
*A	02/06/2015	本文档版本号为 Rev*A，译自英文版 001-93417 Rev*B。
*B	11/27/2017	Minor datasheet edits.
*C	08/23/2022	Updated to PSoC™ Automotive Multitouch guidelines

2 PSoC Creator 帮助

本主题提供了有关如何使用 PSoC Creator 帮助的信息。

如何获得 Help：

- 在任意一个 PSoC Creator 窗口中按**[F1]**按键，打开有关该窗口的“帮助”主题。
- 从 **Help** 菜单中选择 **Topics**，打开 PSoC Creator 帮助文档。

如何使用“Help”：

- 使用 **Contents**（内容）选项卡查看结构表中所有“help”主题的内容。
- 使用 **Index** 选项卡以字母顺序查找和查看关键主题。
- 使用 **Search** 选项卡，以通过键入关键词找到特定的主题。
- 点击蓝色文字，跳转到互联网或帮助主题中的其他主题。

3 入门

本节包含了两组指导您如何开始使用 PSoC Creator 的教程。

- 设计指南 — 指导您逐步快速创建设计。
- 操作指南 — 指导您如何使用 PSoC Creator 提高工作效率。

设计指南

本节包括以下指导内容，这些内容可以帮助您使用 PSoC Creator 开始创建设计。这些指导的示例项目保存在 Find Example Project（查找示例项目）对话框内，可以 PSoC Creator 起始页中找到该对话框。这些指导是对使用 PSoC Creator 的入门级用户提供的快速介绍。有关更多信息，请参见下面内容

- PSoC 3：AN54181：www.cypress.com/go/PSoC3GettingStarted
- PSoC 4：AN79953：www.cypress.com/go/PSoC4GettingStarted
- PSoC 4 BLE：AN91267：www.cypress.com/go/AN91267
- PSoC 5LP：AN77759：www.cypress.com/go/PSoC5GettingStarted
- PRoC BLE：AN94020：www.cypress.com/go/AN94020
- PSoC Creator 培训：www.cypress.com/go/creatorstart/creatortraining

赛普拉斯提供了多种开发套件，您可以使用套件来配合学习如何使用 PSoC Creator。安装开发套件软件包时，这些套件相关的文件链接将出现在 PSoC Creator 起始页中。

初级教程：

- 我的第一个设计 — “Hello World Blinky”
- 入门设计

中级教程：

- 基本设计
- 引脚的使用
- 时钟的使用
- 中断的使用

- 调试设计

高级教程：

- 组件库项目
- 基本层级设计

另外，请参考：

- 查找示例项目
- 如何使用...

初级教程：

我的第一个设计 — “Hello World Blinky”

本节介绍了 PSoC Creator 和开发设计过程。设计过程包括：

- 创建新的项目
- 添加/配置组件
- 编写 C 代码
- 编程器件

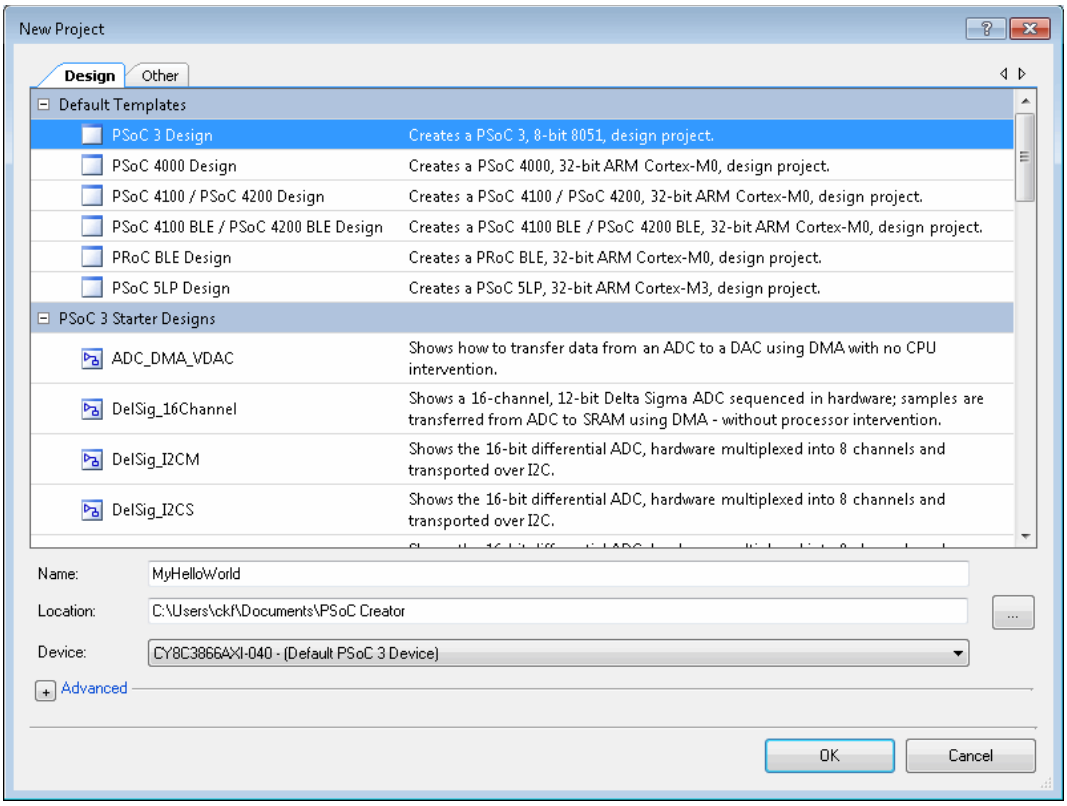
这是该 PSoC Creator 帮助文件提供的最初几个设计指导该设计向您介绍了如何闪烁一个 LED。然后您会添加其他组件，并在某块 LCD 屏上显示“Hello World”。

注意：如果您不想创建新的空项目，则可以通过使用 Find Example Project 对话框打开本节的完整示例项目，即“HelloWorld_Blinky”。该对话框的链接显示在 PSoC Creator “Start Page”（起始页）上。另外，还有一些入门设计可以从 New Project 对话框中创建。

创建一个新项目：

创建基础设计项目是创建设计的第一步。

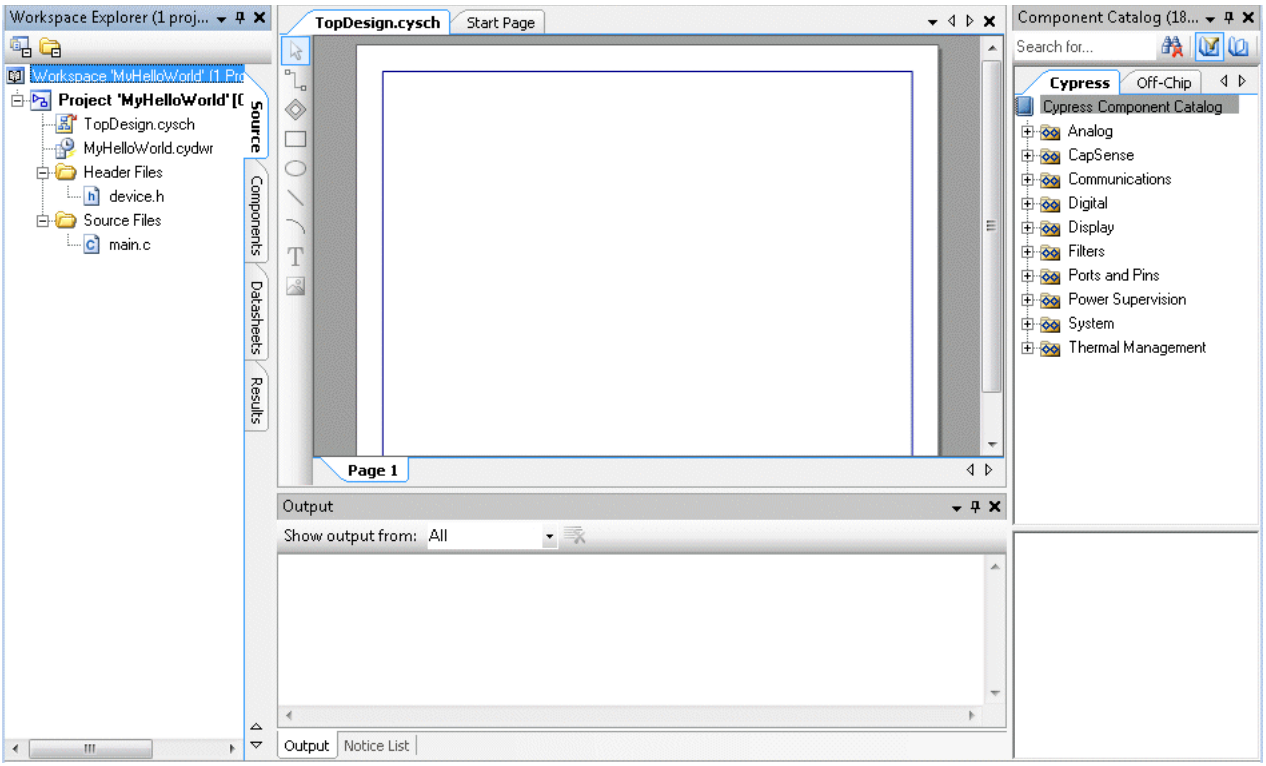
1. 从 **File**（文件）菜单中，依次选择 **New > Project** 或点击打开 New Project（新项目）对话框。



2. 在 **Design**（设计）选项卡下，点击 **PSoC 3 Design** 模板。
3. 在 **Name**（名称）字段中，输入您的项目名称，例如：“MyHelloWorld”。
4. 在 **Location**（地址）字段中，输入您想存储该项目的路径，或直接点击**[...]**导航到相应的目录。
5. 在 **Device**（器件）字段，将其设置为默认的器件，或您需要的具体器件。

   对于本项目，我们使用了默认 PSoC 3 器件 CY8C3866AXI-040。该器件默认被选中。如果您选择的是其他器件，则可能需要相应调整引脚设置。
6. 点击 **OK**。

默认情况下，PSoC Creator 创建一个新工作区来包含了新项目。项目创建后，PSoC Creator 会在 **Source** 选项卡的 Workspace Explorer（工作区浏览器）中添加必要的文件和文件夹。



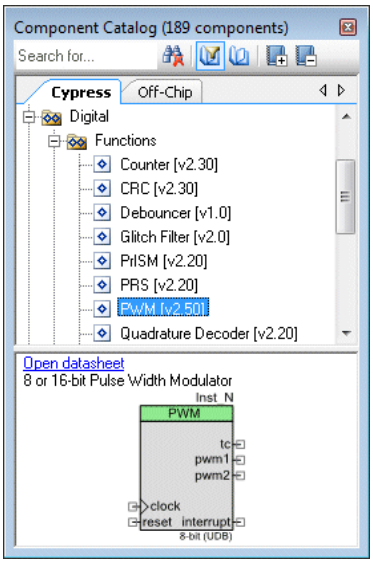
在原理图编辑器中，顶层原理图文件（*TopDesign.cysch*）作为一个文档窗口显示，Component Catalog（组件目录）被打开，显示设计中使用的组件列表。

注意：如果您在同一个 Workspace（工作区）里添加了多个设计项目，请通过选择**项目**项中的 **Set as Active Project** 来激活当前需要操作的那个项目。

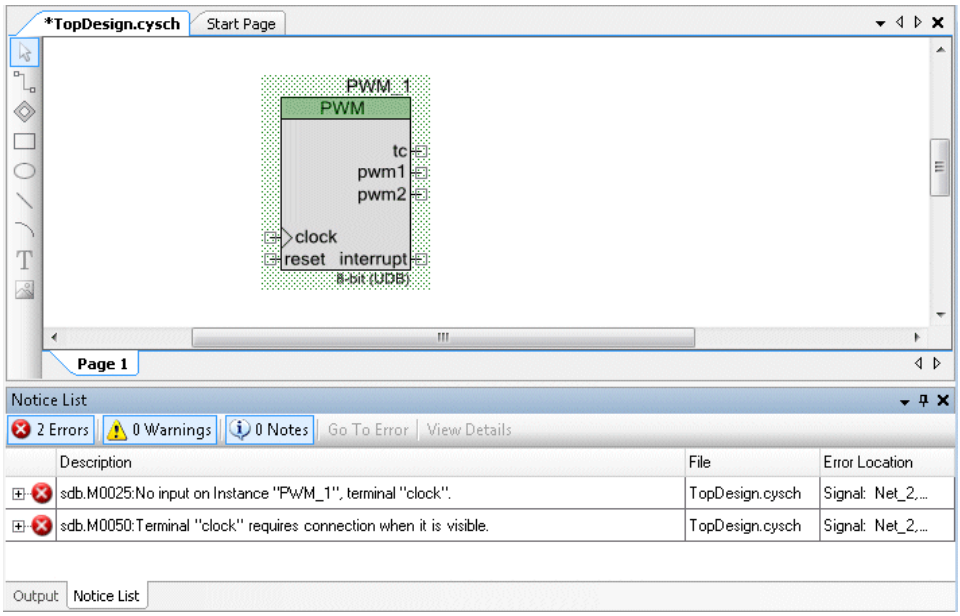
添加/配置组件：

创建新项目后，将组件添加到原理图图纸中，并配置它们。

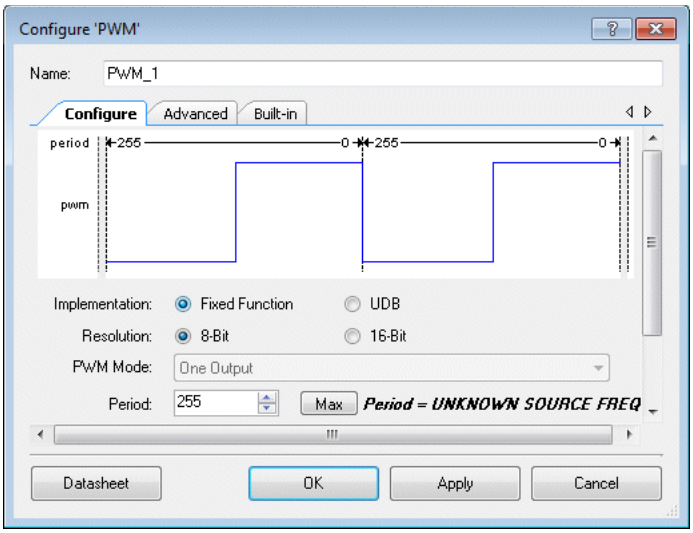
1. 在组件目录中，展开“Digital > Functions”文件夹，并将 PWM 组件拖放到设计中。



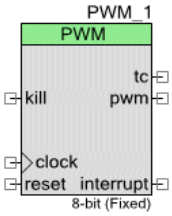
“Notice List”（注意列表）窗口显示的是连接错误。



2. 双击 PWM 组件以打开 Configure（配置）对话框，将 **Implementation** 设置为“Fixed Function”，并点击 **OK**



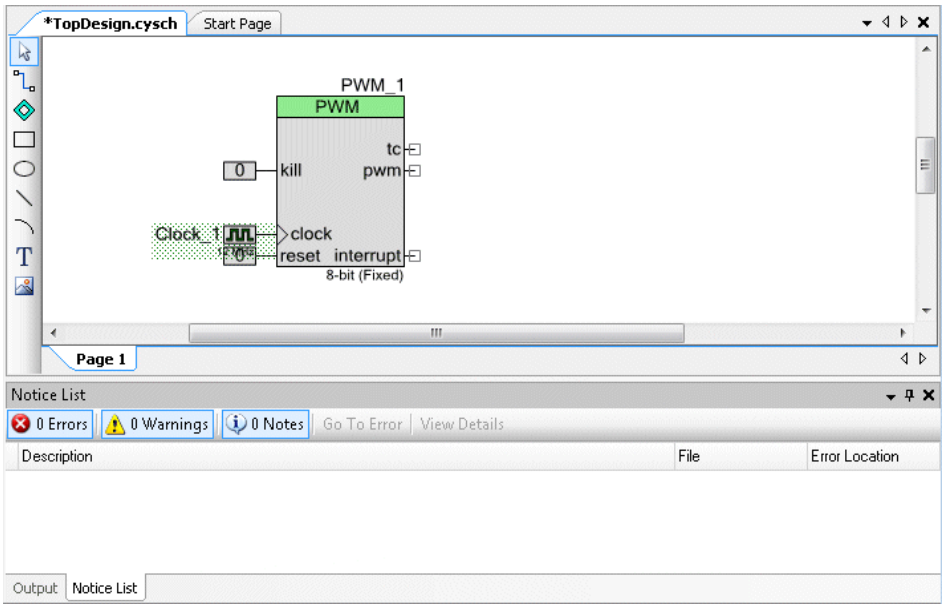
注意：配置后组件的外观发生了变化：添加了 **kill** 输入，一个 **pwm** 输出，另外更新了标签。



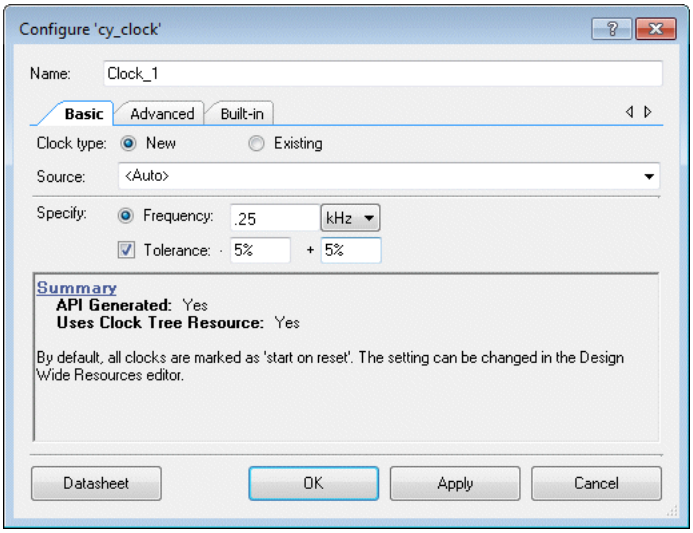
3. 在组件目录中，展开“Digital > Logic”文件夹，拖放 Logic Low '0' 组件到设计中，并将其连接到 PWM 中的 **kill** 终端。拖放另外一个 Logic Low '0' 组件，并将其连接到 **reset** 终端。

4. 在组件目录中，展开“System”文件夹，将时钟组件拖放到设计中，并将它连接到 PWM 的 **clock** 终端。

   “Notice List”（注意列表）窗口被清空。

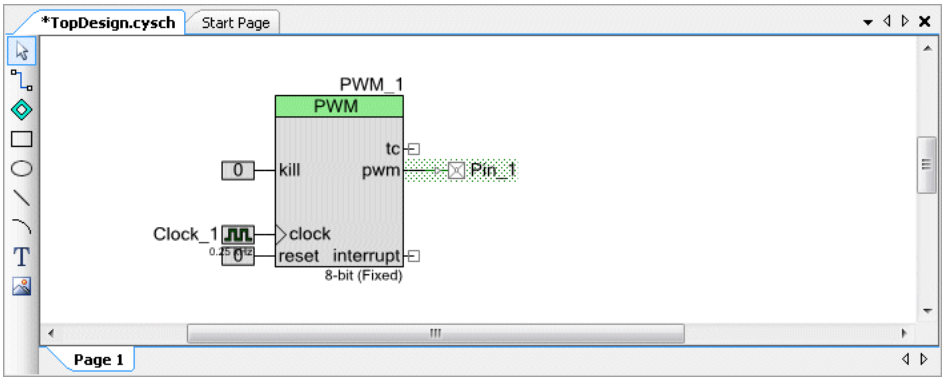


5. 双击“Clock”以打开 Configure 对话框，将 **Desired Frequency** 的值改为 0.25 kHz，然后点击 **OK**。



6. 在组件目录中，展开“Ports and Pins”文件夹，将数字输出引脚组件拖放到设计中，并将其连接到 PWM 中的 **pwm** 终端。

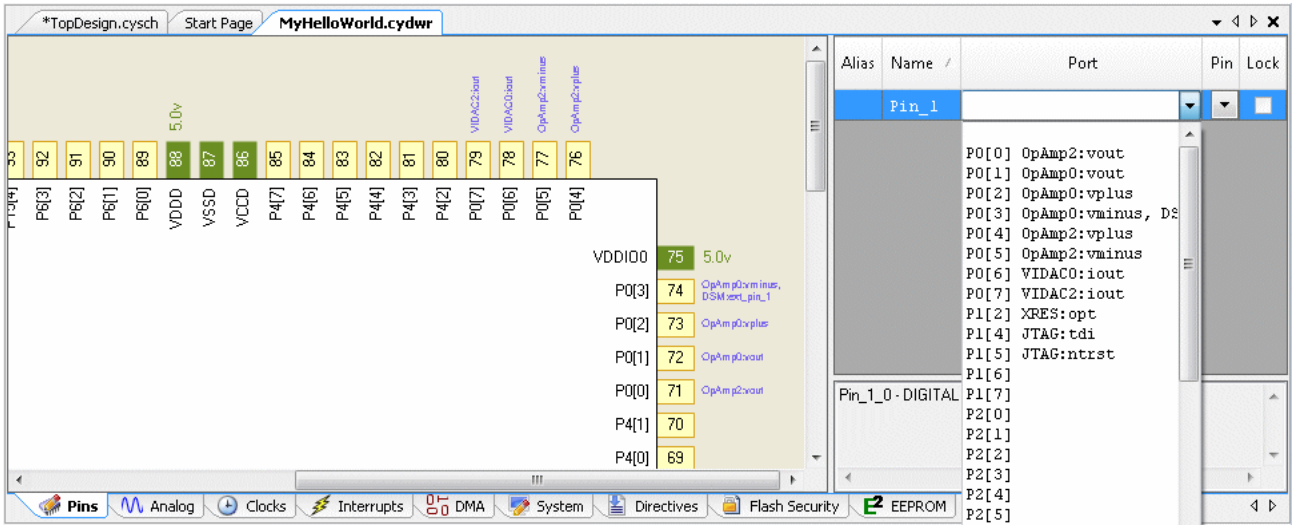
   您的设计应同下面的原理图相同。



分配引脚：

PSoC Creator 会自动将引脚分配给器件上的物理端口/引脚。要指定具体的引脚，请使用 Pin Editor（引脚编辑器）。

1. 在工作区浏览器中，双击 *HelloWorld.cydwr* 文件以打开“设计范围资源引脚编辑器”。



2. 在 **Port（端口）**或 **Pin（引脚）**列选中下拉菜单，并根据您拥有的套件分配 Pin_1 到下面的引脚：
   - 对于 PSoC 3 FirstTouch 初学者套件（CY8CKIT-003），使用 P4[1]（或引脚 70）。
   - 对于 PSoC 开发套件（CY8CKIT-001），使用 P0[0]（或引脚 71）。

编写 C 代码：

1. 在工作区浏览器中，双击 *main.c* 文件可打开它。
2. 将下面的函数添加到 main()内：

```
PWM_1_Start();
```

编程器件：

1. 将 MiniProg3 连接到您的套件，并使用一根 USB 电缆将其连接到 PC。
2. 如果您正在使用 PSoC 开发套件（CY8CKIT-001），则使用一跟连线将 P0[0]引脚连接到其中一个 LED。有关更多信息，请参考套件的相应文档。
3. 点击 **Program**（编程）。
4. 如果显示 Select Debug Target 对话框，则选择您的器件，然后点击 **Connect** 和 **OK**。

PSoC Creator 将构建您的设计、生成代码和编程器件。当编程操作完成时，电路板上选定的 LED 将闪烁发光；如果需要，请按一下 **Reset** 按键。

扩展设计：

如果您正在使用 PSoC 开发套件（CY8CKIT-001），您可以扩展设计，在 LCD 上显示“Hello World”。

1. 在组件目录中，展开“Display”文件夹，将 Character LCD 组件拖放到原理图中。
2. 打开引脚编辑器，并将 LCD_Char_1:LCDPort 分配给 P2[6:0]。
3. 打开 *main.c* 文件，并编辑它来将下面的函数添加到 main()中：

```
LCD_Char_1_Start();
LCD_Char_1_PrintString("Hello World");
```

4. 点击 **Program**（编程）。

PSoC Creator 将构建您的设计、生成代码和编程器件。当编程操作完成时，LCD 会显示“Hello World”字符。

从现在开始，您可以根据自己的想法修改设计来想要的功能。

另外，请参考：

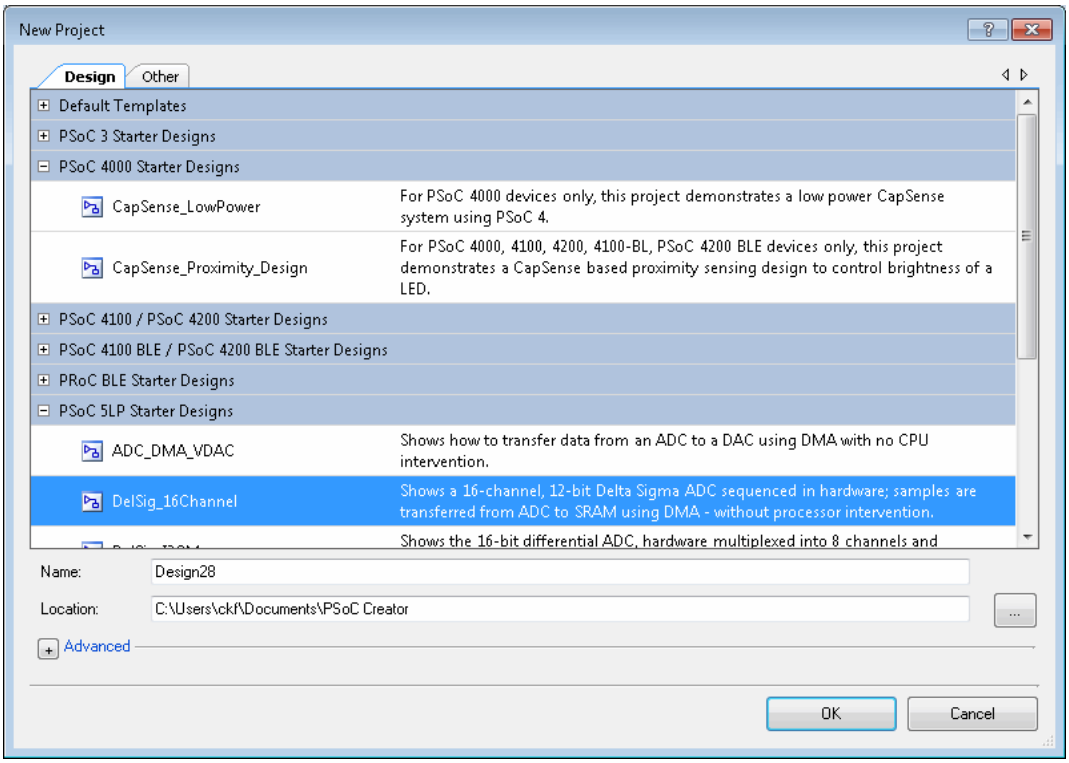
- 入门设计
- 查找示例项目
- 工作区浏览器
- 原理图编辑器
- 组件目录
- 引脚编辑器

入门设计

PSoC Creator 提供了几个入门设计。这些设计突出了 PSoC 器件的独特功能。通过这些功能，您可以使用各种组件和已提供的代码创建设计，而不用创建新的空设计。

要使用入门设计，请按照下面步骤进行操作：

1. 从 **File**（文件）菜单中，依次选择 **New > Project** 或直接点击打开 New Project 对话框。
2. 在 **Design** 选项卡下，向下滚动到**入门设计**章节。



3. 选择所需的入门设计。每个器件都有一些可选的入门设计。

   注意：PRoC BLE 器件不提供入门设计。但提供了它们的自定义设计原理图，如创建一个新的项目一节所示。

4. 在 **Name**（名称）字段中，输入您的项目名称。

5. 在 **Location**（地址）字段中，输入您想存储该项目的路径，或点击**[...]**导航到相应的目录。

6. 如果需要，请展开 **Advanced** 项，然后选择创建新工作区并指定工作区的名称或将已有的项目添加到现有的工作区。

7. 点击 **OK**。

   在工作区浏览器中创建选定的项目并显示文件。项目准备就绪进行构建和编程器件。有关更多信息，请参阅设计中包含的文档。如果不存在文档，请在 Options 对话框中选择 **Include Starter Design documentation...**选项，并创建项目。

另外，请参考：

- 创建新项目
- 工作区浏览器
- 我的第一个设计 — “Hello World Blinky”
- Options 对话框

中级教程：

基本设计

本节提供了更多有关组件配置和设计范围资源（DWR）设置的详细信息。基本设计重点介绍不同基本组件之间不同的配置方法。本节没有对设计过程中的每一步进行详细说明。却介绍了与设计各部分相关的概述。在所有设计中您都可以使用相同的原则。有关创建设计的指导，请参考“Hello World Blinky”节。

注意：如果您不想创建新的项目，则可以通过使用 Find Example Project 对话框打开本节中完整的示例项目，即“BasicDesign”。该对话框的链接显示在 PSoC Creator “Start Page”（起始页）上。

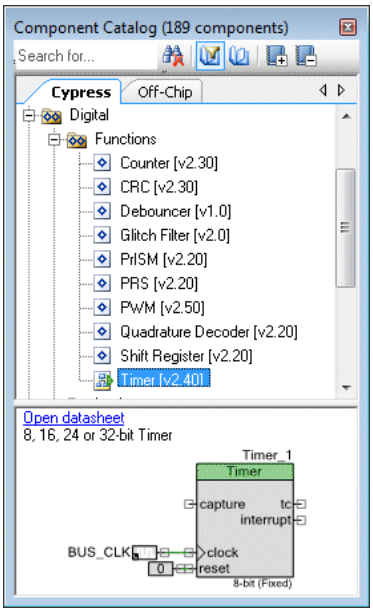
创建新项目：

如“Hello World”一节中显示的内容，设计的第一步是创建一个新的项目。如果需要，可以参照上一节的指导。您也可以参考创建新的项目一节。

选择和配置数字组件：

对于本节，主要的数字组件为定时器。然后，您将为设计添加支持组件，配置它们，并将它们连接至定时器。

1. 在组件目录中，展开 Digital > Functions 文件夹，点击 Timer component（定时器组件），然后将其拖放到设计图纸中。



   注意：定时器已经连接到时钟组件和逻辑低组件。因为定时器就是原理图宏中的示例。

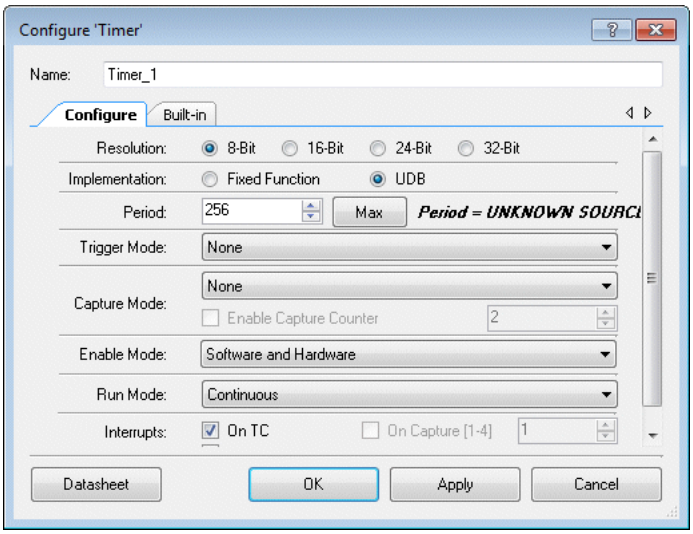
2. 删除时钟组件和逻辑低组件，因为该示例将使用其他组件。

3. 双击定时器组件，以打开 Configure 对话框。

   默认情况下，定时器被配置为：8-bit、Fixed Function 以及 Software Only enable（仅能通过软件使能）。有关定时器组件的详细信息，请参考其数据手册。

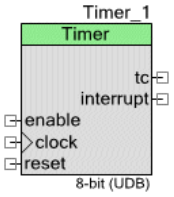
更改下面各参数：

- 将 Implementation 更改为 UDB
- 将 **Enable Mode**（使能模式）更改为“Software and Hardware”（硬软件）
- 将“**Interrupt**”（中断）选择为“**on TC**”选框



4. 点击 **OK**。

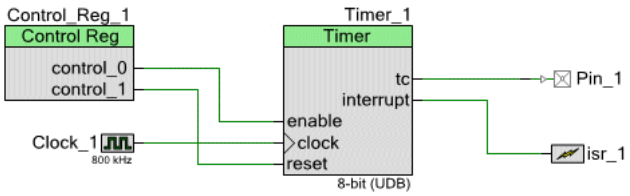
注意：在定时器中会显示“enable”终端。



5. 从组件目录中，将下面各个组件添加到设计中：
   - 控制寄存器（属于 Digital > Registers 文件夹） — 用于使能受软件控制的定时器，并为定时器提供“reset”信号。
   - 数字输出引脚（属于 Ports and Pins 文件夹） — 用于为设计中的其他组件提供分频时钟。分频时钟的频率为 3.1 KHz。
   - 时钟（属于 System 文件夹） — 用于提供输入时钟。在此处，时钟的频率为 800 KHz。
   - 中断（属于 System 文件夹） — 作为设计的 heartbeat（心跳），中断可用于软件的定时功能。
6. 根据下述内容配置下面各组件实例：

实例	参数
Control_Reg_1	NumOutputs：2
Clock_1	所需频率：800 Khz

7. 对设计中的各组件进行排序，并使用 Design Elements Palette（设计元素控制板）的 **Wire**（线）工具相互连接各组件，如下所示。有关更多信息，请参阅 Wire 的用法一节。

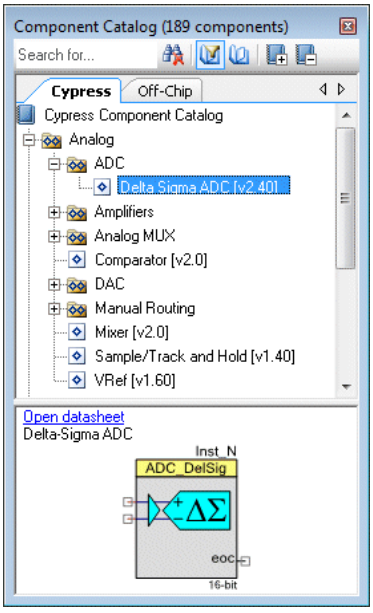


完成时，设计应该不存在 DRC 警告和错误。

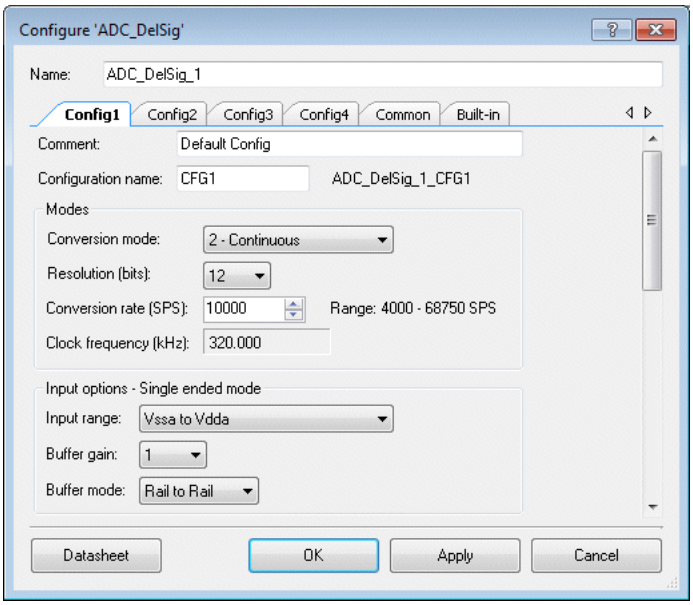
选择和配置模拟组件：

设计的下一部分将包含 Delta Sigma 模数转换器组件和对应的支持组件。

1. 从组件目录中，将 Delta Sigma 模数转换器组件（属于 Analog > ADC 文件夹）拖放到设计中。



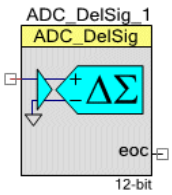
2. 双击组件，以打开 Configure 对话框。



默认情况下，Delta Sigma 模数转换器被配置为 16 位，其输入范围为+/- 1.024 V。有关 Delta Sigma 模数转换器组件的详细信息，请参阅其数据手册。

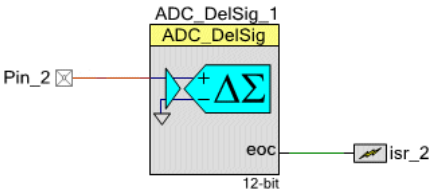
3. 按照下面的内容更改各参数：
   - 在 **Common** 选项卡下，将 **Input Mode**（输入模式）设置为“Single ended”（单端）。
   - 在 **Config1** 选项卡下，将 **Resolution**（分辨率）参数更改为“12”，并将 **Input Range**（输入范围）修改为“Vssa to Vdda”。
   - 点击 **OK**。

   注意：组件下面的标签是指分辨率为 12 位。



4. 从组件目录中，将下面各个组件添加到设计中：
   - 模拟引脚（属于 Ports and Pins 文件夹） -- 用于将输入信号连接到模数转换器。
   - 中断（属于 System 文件夹） -- 用于触发从模数转换器中读取转换结果的中断。

5. 对设计中的各组件进行排序，并使用 Design Elements Palette（设计元素控制板）的 **Wire**（线）工具相互连接各组件，如下所示。



注意：

- 第一，模拟引脚到模数转换器的连线默认为棕红色，但模数转换器到中断的连线默认为绿色。PSoC Creator 使用这些颜色区别模拟和数字信号。通过使用 Options 对话框可以更改这些颜色。
- 第二，注意互相连接各组件时可以无需连线。通过连接终端可以直接连接它们。本节使用的不同颜色的连线显示不同类型的信号。

完成时，设计应该不存在 DRC 警告和错误。

编辑源代码：

1. 通过编译项目生成组件的源文件。

   完成时，Workspace Explorer（工作区浏览器）会展开 Generated_Source 文件夹，以便显示设计中多个组件的所有.c 文件和.h 文件。

2. 打开头文件以查看（和复制）在 *main.c* 文件中需要输入的函数原型。

3. 复制和粘贴下面代码，并替代 *main.c* 文件中的所有代码。

   注意： 下面代码的注释内容仅供参考。

```
/*******************************************************************************
* File: main.c
*
* Version: 2.0
*
* Description:
*   This is a basic design source code.
*
********************************************************************************/
#define ADC_NUMBER_SAMPLES (15)

/* Initialize array elements to zero. */
uint16 ADC_Samples[15] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

/* Defining and initializing the index */
int8 ADC_Sample_Index = 0;

/* Initiialize the average result */
uint32 ADC_Sample_Average = 0;

/* Sample read from ADC */
int16 ADC_Current_Sample = 0;

/* Indicator for when sample is available */
int8 ADC_Sample_Available = 0;
/*******************************************************************************
* Function Name: InterruptHandler
********************************************************************************
* Summary:
*   The Interrupt Service Routine for isr_1.
*
* Parameters:
*   None.
*
* Return:
*   void.
*
*******************************************************************************/
```

```
CY_ISR(InterruptHandler)
{
    Timer_1_ReadStatusRegister(); /* Read the Status Register */
}

/*******************************************************************************
* Function Name: main
********************************************************************************
* Summary:
* Main function performs following functions:
* 1: Start the clock
* 2: Start the Timer
* 3: Start the interrupts
* 4: Start ADC DelSig and its interrupts
* 5: Testing for sample available from ADC
* 6: Storing the sample into the array
* 7: Comparing the samples
*
* Parameters:
*  None.
*
* Return:
*  None.
*
*******************************************************************************/

void main()
{
    int8 i;

    CyGlobalIntEnable;
    /* Start the Interrupt */
    isr_1_Start();
    isr_1_Disable();
    isr_1_SetVector(InterruptHandler);
    isr_1_Enable();

    /* Enable the Timer; reset disabled */
    Control_Reg_1_Write(0x01);

    /* Start the LCD */
    LCD_Start();
    /* Start the Timer */
    Timer_1_Start();

    /* Start the ADC */
    ADC_DelSig_1_Start();

    /* Start the ADC conversion */
    ADC_DelSig_1_StartConvert();

    LCD_Position(0, 0);
    LCD_PrintString("ADC OUTPUT:");
    for(;;)
    {
        /* Check whether ADC conversion complete or not */
        if (ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT))
        {
            /* Get the result */
            ADC_Current_Sample = ADC_DelSig_1_GetResult16();
            ADC_Sample_Available = 1;
            /* Print the ADC result on LCD */
            LCD_Position(0, 11);
```

```
            LCD_PrintInt16(ADC_Current_Sample);
        }
        /* Testing for sample available from the ADC */
        if (ADC_Sample_Available)
        {
            ADC_Sample_Available = 0;

            /* storing the sample into the array, based on the index */
            ADC_Samples[ADC_Sample_Index++] = ADC_Current_Sample;

            /* comparison */
            if (ADC_Sample_Index == ADC_NUMBER_SAMPLES)
            {
                ADC_Sample_Average = 0;
                for (i = 0; i < ADC_NUMBER_SAMPLES; i++) ADC_Sample_Average +=
ADC_Samples[i];
                ADC_Sample_Average /= ADC_NUMBER_SAMPLES;
                ADC_Sample_Index = 0;
            }
        }
    }
}
/* [] END OF FILE */
```

编译项目。

点击 **Build**。PSoC Creator 将编译项目，并显示相应信息。根据需要，解决所有错误或警告。

如果您拥有带 LCD 的 PSoC 开发套件，那么可以点击 **Program**进行编程器件并观察 LCD 上的输出。

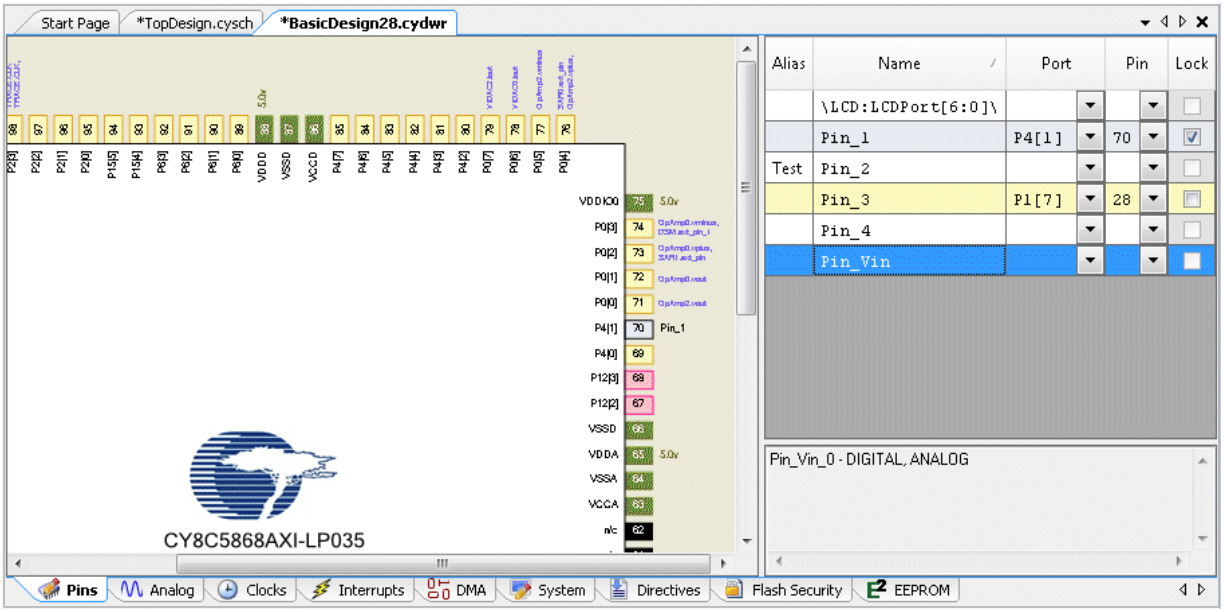
下一步：

本节介绍了设计中配置组件的基本过程。为了继续进行该设计，您需要获得硬件以编程器件，同时使用调试器的功能。与此同时，您可以参考下面各相关主题：

- 引脚的使用
- 时钟的使用
- 中断的使用
- 调试设计

引脚的使用

在 PSoC Creator 中，您需要通过使用引脚组件来访问器件上的物理引脚。你可以从原理图或软件中可以访问引脚组件。使用引脚组件，您还也可以将一组引脚作为一个个体来管理。PSoC Creator 通过设计范围资源引脚编辑器来管理设备引脚。



本节包括以下各主题：

- 引脚
- 引脚映射
- 软件访问

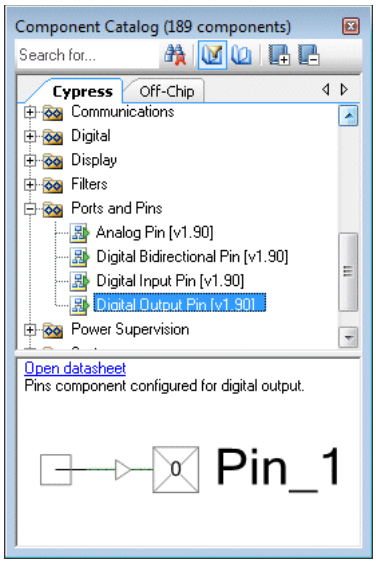
打开引脚编辑器：

在 **Source** 选项卡的 Workspace Explorer（工作区浏览器）的中双击*<Project>.cydwr* 文件。

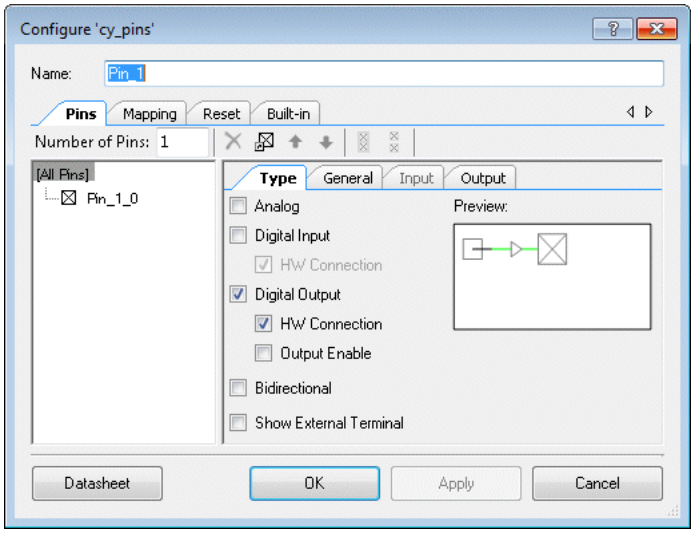
在工作区内，该文件显示为选项卡式文档，这样便于您访问项目中的各种设计范围资源。默认情况下，引脚编辑器（**Pins** 选项卡）在显示在最上方。如果当前显示的是其他编辑器，请点击 **Pins** 选项卡，以将它置于顶层。

引脚：

引脚是一种组件，通过它们您可以在设计中给物理引脚设置复杂的接口。您可以从 Ports and Pins 文件夹下的组件目录中选中它们，然后拖放到设计中。



组件目录中提供了 4 种预先配置的引脚组件：模拟引脚、数字双向引脚、数字输入引脚和数字输出引脚。您可以任意使用这些组件类配置单一的引脚或总线。Configure 对话框提供了多个管理功能参数。



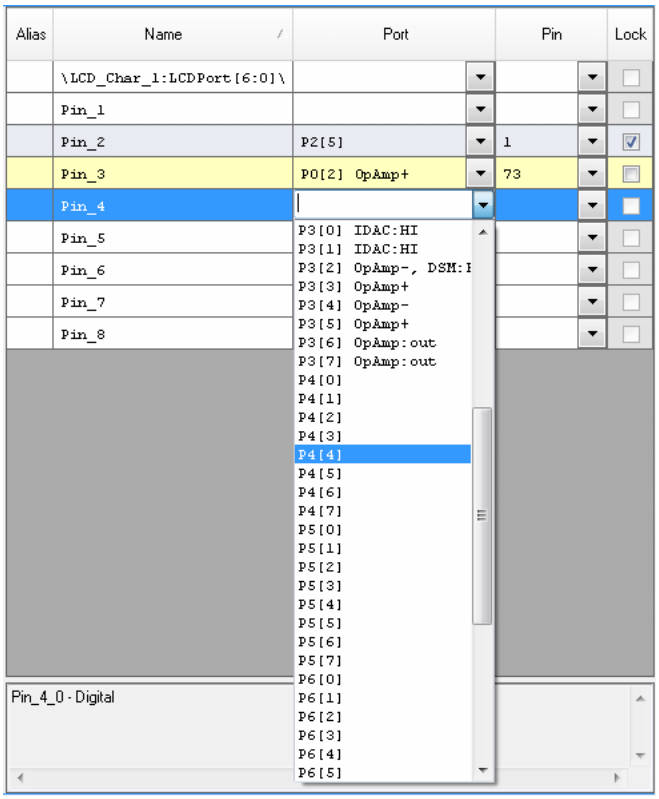
使用多种设置，您还可以通过配置组件来使用 SIO 引脚、Vref 和/或中断。有关每个配置的指定功能的详细信息，请参阅引脚组件的数据手册。在组件目录中，或通过在 Configure 对话框中点击 **Datasheet** 按键能找到组件数据手册。

引脚的电气特性（如驱动模式和转换速率）可以在 Configure 对话框中单独设置。您可以为所有引脚组件创建别名，该别名可以显示在引脚编辑器中和并用于生成特定的 APIs。不同的别名可以方便用户记住不同引脚的功能并方便在软件中对其进行访问。

引脚映射:

PSoC Creator 可以根据引脚自身的限制和用户设置的限制来自动为引脚组件映射物理引脚。配置包含多个引脚的组件可将这些引脚指定为连续或者非连续映射。当被指定为连续映射，该组件包含的多个引脚只能映射到单一物理端口的相邻引脚。在这种情况下，引脚的分配不能跨越多个物理端口，因此引脚数量最多限制为 8。引脚编辑器会强制将连续映射的引脚放置在相邻的引脚上。对于非连续映射的引脚，您可以单独放置各个引脚到制定的物理端口。

对于每一个引脚组件，您可以自己配置映射地址或者让 PSoC Creator 自动帮您完成。您可以通过引脚编辑器中 Port 或者 Pin 选项的下拉菜单来为各个引脚组件分配映射物理引脚。



软件访问：

通过生成的 API 您可以管理各个引脚。更多有关 API 的信息，请参考数据手册。

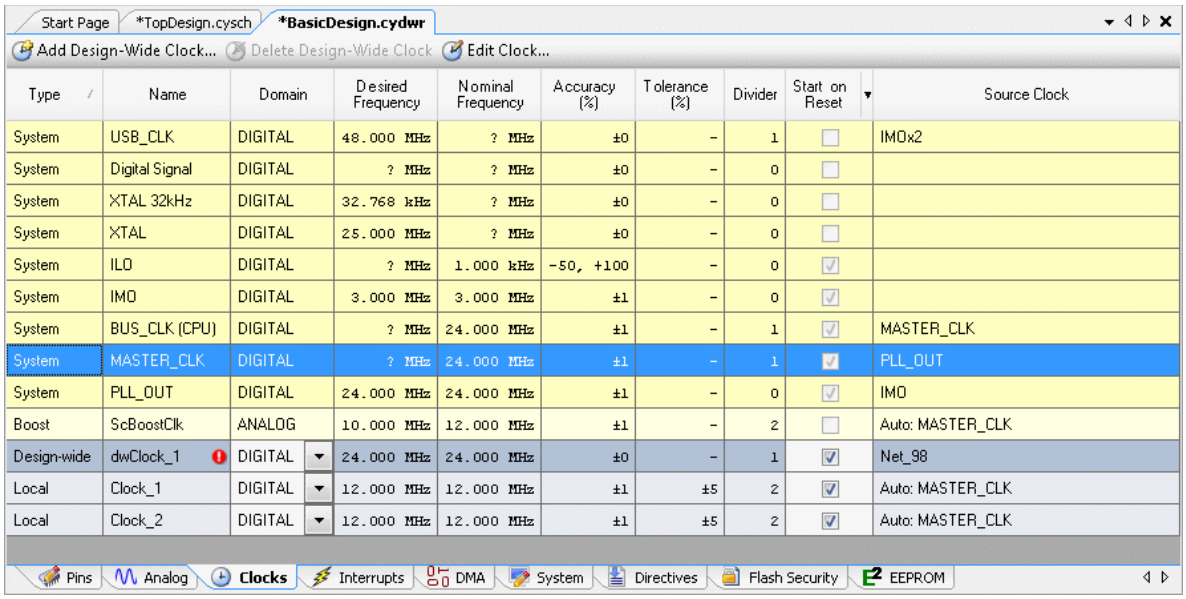
另外，请参考：

- Design-Wide Resources（设计范围资源）
- 引脚编辑器
- 组件目录

时钟的使用

PSoC 器件为您的设计提供了一系列精密且灵活的时钟资源。PSoC Creator 允许您在各种情况下使用这些时钟资源，以配合您的应用。您还可以将时钟源设置为内部振荡器、外部时钟或晶体。这样您就可以根据自己的应灵活配置时钟源和始终组件，得到您需要的时钟。

PSoC Creator 列出了设计范围资源时钟编辑器中的所有时钟。



Type	Name	Domain	Desired Frequency	Nominal Frequency	Accuracy (%)	Tolerance (%)	Divider	Start on Reset	Source Clock
System	USB_CLK	DIGITAL	48.000 MHz	? MHz	±0	-	1	☐	IMOx2
System	Digital Signal	DIGITAL	? MHz	? MHz	±0	-	0	☐	
System	XTAL 32kHz	DIGITAL	32.768 kHz	? MHz	±0	-	0	☐	
System	XTAL	DIGITAL	25.000 MHz	? MHz	±0	-	0	☐	
System	ILO	DIGITAL	? MHz	1.000 kHz	-50, +100	-	0	☑	
System	IMO	DIGITAL	3.000 MHz	3.000 MHz	±1	-	0	☑	
System	BUS_CLK (CPU)	DIGITAL	? MHz	24.000 MHz	±1	-	1	☑	MASTER_CLK
System	MASTER_CLK	DIGITAL	? MHz	24.000 MHz	±1	-	1	☑	PLL_OUT
System	PLL_OUT	DIGITAL	24.000 MHz	24.000 MHz	±1	-	0	☑	IMO
Boost	ScBoostClk	ANALOG	10.000 MHz	12.000 MHz	±1	-	2	☐	Auto: MASTER_CLK
Design-wide	dwClock_1	DIGITAL	24.000 MHz	24.000 MHz	±0	-	1	☑	Net_98
Local	Clock_1	DIGITAL	12.000 MHz	12.000 MHz	±1	±5	2	☑	Auto: MASTER_CLK
Local	Clock_2	DIGITAL	12.000 MHz	12.000 MHz	±1	±5	2	☑	Auto: MASTER_CLK

几乎在所有设计中，您都可以使用该编辑器优化应用程序的时钟设置。

时钟有三种类型：系统时钟、本地时钟和设计范围时钟。以下内容介绍的是这些时钟的用法。

- 配置系统时钟资源
- 在原理图中使用本地时钟
- 创建新设计范围时钟资源
- 使用软件管理时钟

欲了解如何设置时钟，请参考时钟编辑器。另外，您也可以参考组件目录中的时钟组件数据手册。

欲了解如何在一个示例应用中配置时钟组件，请参考基本设计。

要想打开时钟编辑器，请执行下面的操作：

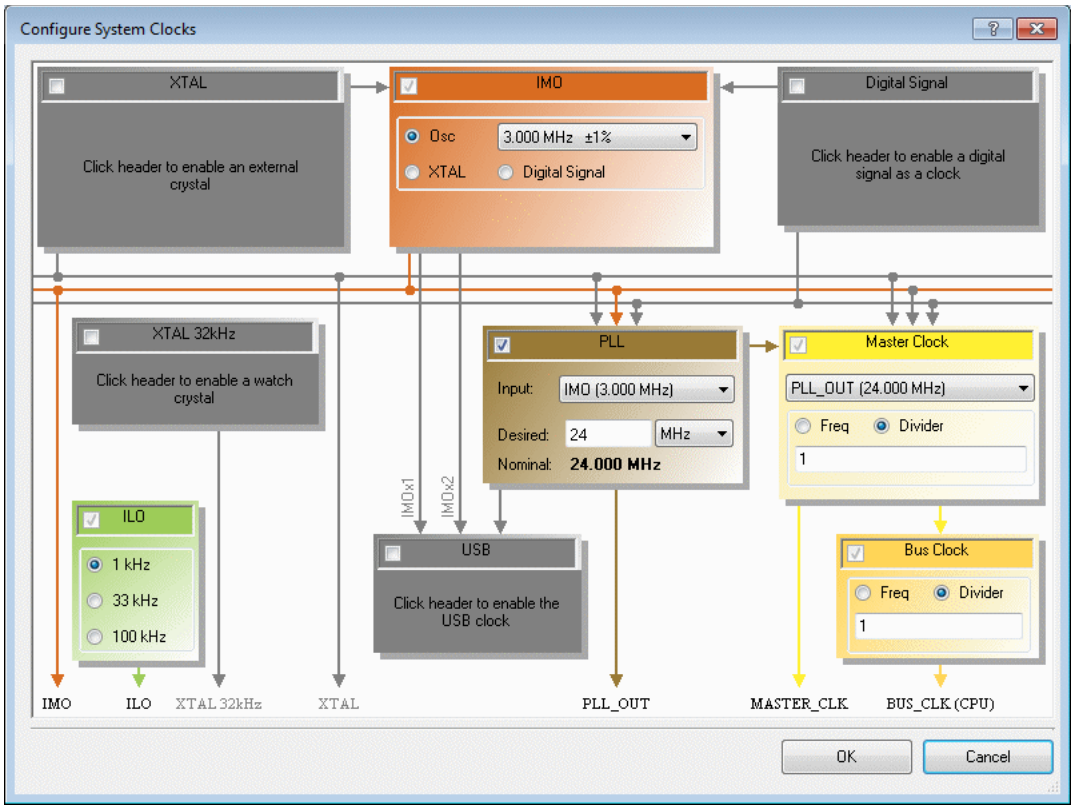
双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

在工作区内，该文件显示为选项卡式文档，这样便于您访问项目中的各种设计范围资源。点击 **Clocks**（时钟）选项卡以访问时钟编辑器。

配置系统时钟资源：

PSoC Creator 自动列出时钟编辑器中的各系统时钟。这些时钟被设置为典型的默认值，因此在早期设计过程中，您不需要对其进行过多更改。

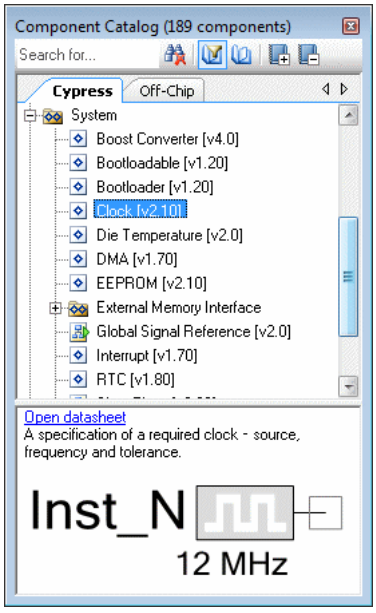
点击 **Edit Clock**（编辑时钟）打开 Configure System Clocks（配置系统时钟）对话框，然后查看各时钟间的关系，并按照您的要求进行设置。



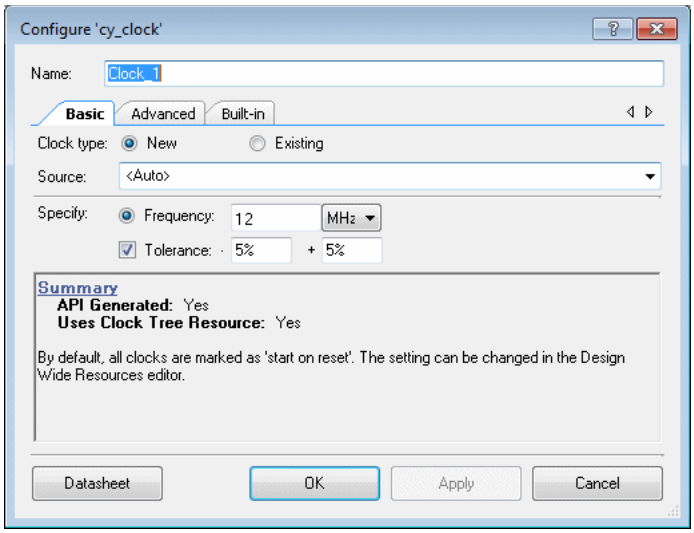
更多有关信息，请参考配置系统时钟中介绍的内容。

在原理图中使用本地时钟

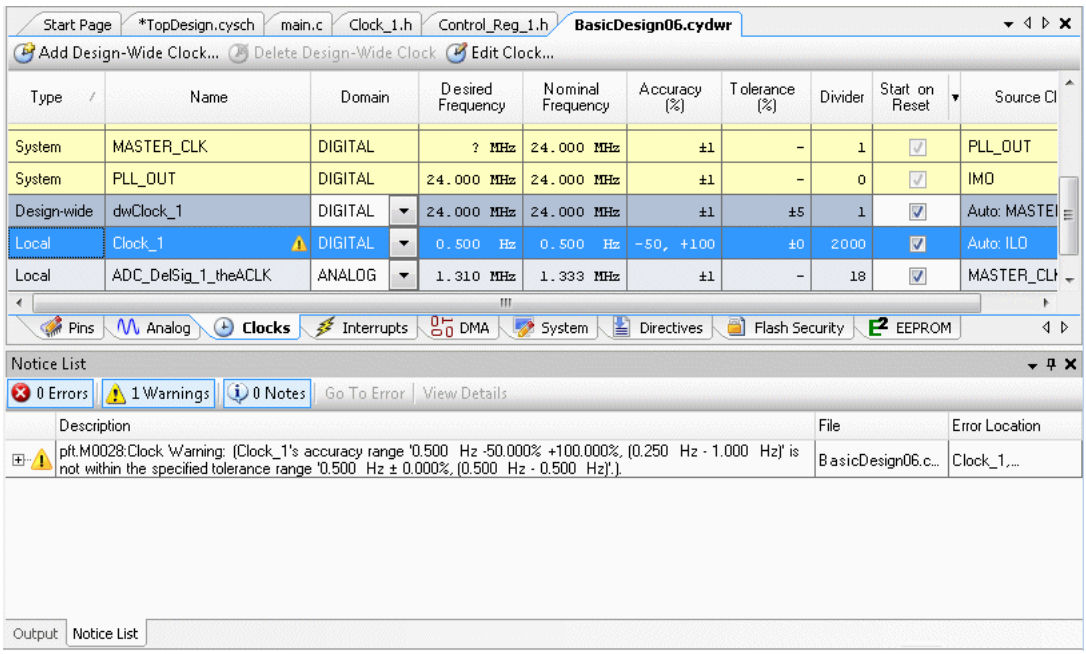
当您将组件目录中的时钟组件拖放到原理图中时，将创建本地时钟。



通过双击该时钟打开 Configure Clock（配置时钟）对话框，这样您便可以指定本地时钟的各项特性。



在时钟编辑器系统时钟下面会列出设计中添加的本地时钟。检查时钟的实际派生频率是一个很好的设计实践经验。如果不能得到所需要的频率（包含容差值），则工具将在 Notice List（注意列表）窗口中产生一条警告信息。



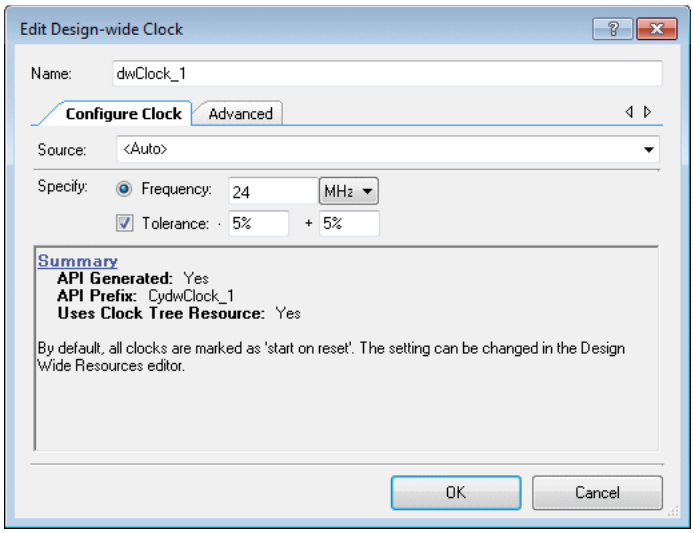
更多有关信息，请参考配置本地时钟中介绍的内容。另请参见时钟组件数据手册。

创建新设计范围时钟源：

在设计中的任何地方您都可以随时使用设计范围时钟。这些时钟源由系统时钟（或其他设计范围时钟）派生得到。

一般情况下，您需要创建自己的时钟资源（这些时钟资源有特殊的频率和容差），以便可以在您的设计中多次使用这些时钟资源。请注意，用于设计范围时钟的 API 前缀为“Cy”。

要想创建设计范围时钟，请点击时钟编辑器上的 **Add Clock**（添加时钟）按钮。Add/Edit Clock 对话框与本地时钟的配置对话框基本相同。



同本地时钟相似，设计范围时钟也罗列在系统时钟下的时钟编辑器内。检查时钟的实际派生频率是一个很好的设计实践。如果不能找到所需要的频率（包含容差值），那么工具将在 Notices List（注意列表）窗口中生成一条警告信息。

更多有关信息，请参考添加/编辑设计范围时钟。

使用软件管理时钟：

可以使用编译过程生成的软件 API 函数控制所有‘新’时钟。时钟组件数据手册中详细介绍了这些 API 函数。最重要的 API 函数是_Start()和_Stop()。各个时钟在默认情况下是使能状态。如果勾选了 **Start in Reset** 选项，那么在进入 main 函数前设置时钟，将自动调用_Start()函数。

欲了解如何控制系统时钟，请参考 PSoC 器件的*《技术参考手册》*。

另外，请参考：

- 设计范围资源
- 时钟编辑器

中断的使用

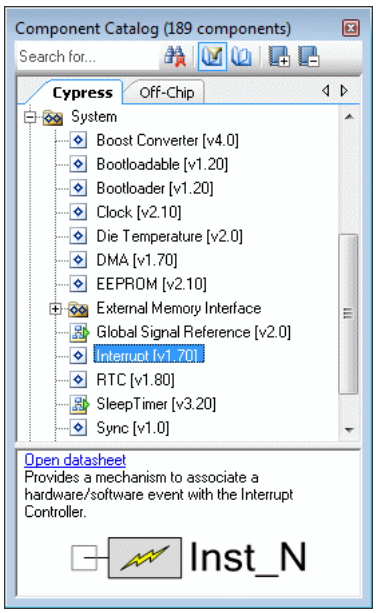
使用中断组件表明了在您的设计中有一个硬件触发中断。通过使用中断组件并写入处理程序代码，您可以在原理图中创建中断。另外，您可以使用设计范围资源中的中断编辑器调整中断优先级。

- 在原理图中使用中断
- 调整中断优先级
- 编写中断处理程序

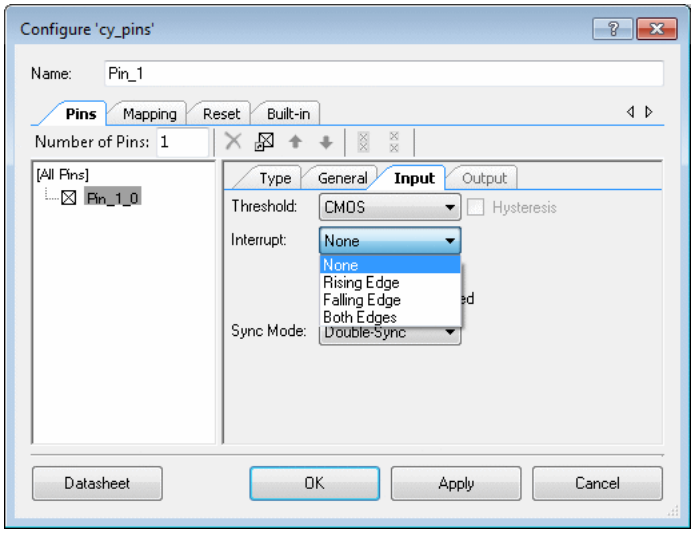
有关中断的普通信息，请参考中断编辑器中介绍的内容。另外，您也可以参考组件目录中的中断组件数据手册。

在原理图中使用中断

中断是用于填充向量表和设置软件处理程序的组件。您可以在系统文件夹下的组件目录中选择中断组件，然后将它们拖放到您的原理图上。



这些中断组件通常连接到其他组件的中断输出端口上（如 SIO 引脚、定时器和计数器、DMA、ADC 等）。中断触发模式是由中断信号的生成组件决定的。在某些组件中（如 PWM），不能更改该模式。在其他组件中（如 SIO 引脚），该模式作为引脚配置对话框中的可选项。



当一个中断与原理图中的数字逻辑（如时钟或逻辑门）相连时，触发始终发生在上升沿上。

要想打开中断编辑器，请执行下面的操作：

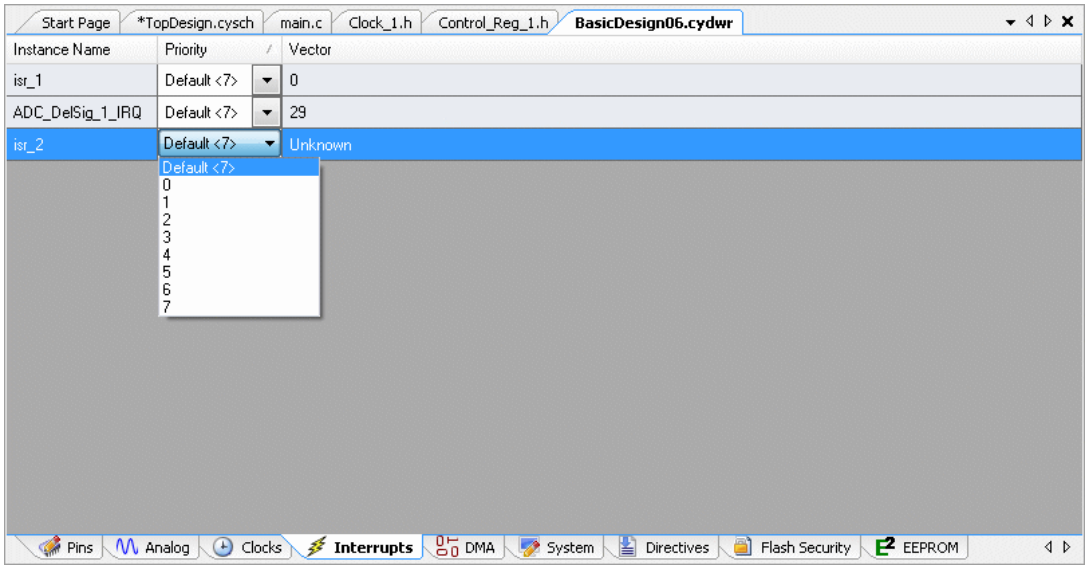
双击 **Source**（源）选项卡的 Workspace Explorer（工作区浏览器）中的“.cydwr”文件。

在工作区内，该文件显示为选项卡式文档，这样您可以访问项目中的各种设计范围资源。点击 **Interrupts**（中断选项卡）访问中断编辑器。

注意：如果该设计中没有使用任何中断，则中断编辑器将显示一条信息。

要想调整中断优先级，请执行下述操作：

PSoC Creator 管理设计范围资源中断编辑器中的各个中断。



通过下拉菜单可以为每个中断选择优先级。

1. 打开设计范围资源文件，然后点击 **Interrupts**（中断）选项卡。
2. 对于特殊的中断，请使用 **Priority**（优先级）列中的下拉菜单，并选择所需值。
3. 点击 **Save** 按钮。

编写中断处理程序

中断组件可以生成中断组件数据手册中介绍的多个 API 函数。API 文件位于工作区浏览器窗口中 **Source**（源）选项卡下的 Generated_Source 目录内。

除了通用的 Start/Stop 函数以及用于管理向量和 ISR 优先级的函数外，还有处理程序代码。该处理程序定义在中断源文件中，它包含各标签，以便在各编译程序之间保护您的代码。编译程序不能覆盖`#START name`和`#END`文本围绕的源代码。

```
CY_ISR(tick_100ms_Interrupt)
{
    /*  Place your Interrupt code here. */
    /* `#START tick_100ms_Interrupt` */
    /* `#END` */
}
```

注意：`CY_ISR()`宏用于定义函数名称和各参数。该宏为中断函数处理编译器的特殊 C 语言扩展，并确保该代码保持可移植到其他 PSoC 架构内的性能。

调试设计

本节将使用基本设计一节中创建的设计来介绍了某些调试概念，如设置断点、添加观察窗口和跳过代码行。这个设计将显示在不同执行点上变量数组的内容。另外它还显示了特定变量的总值和平均值。

打开示例设计：

如果您按照“基本设计”一节中指导 i 创建了一个设计，那么请使用该设计。欲了解如何打开所需要的设计，请参考打开现有项目。

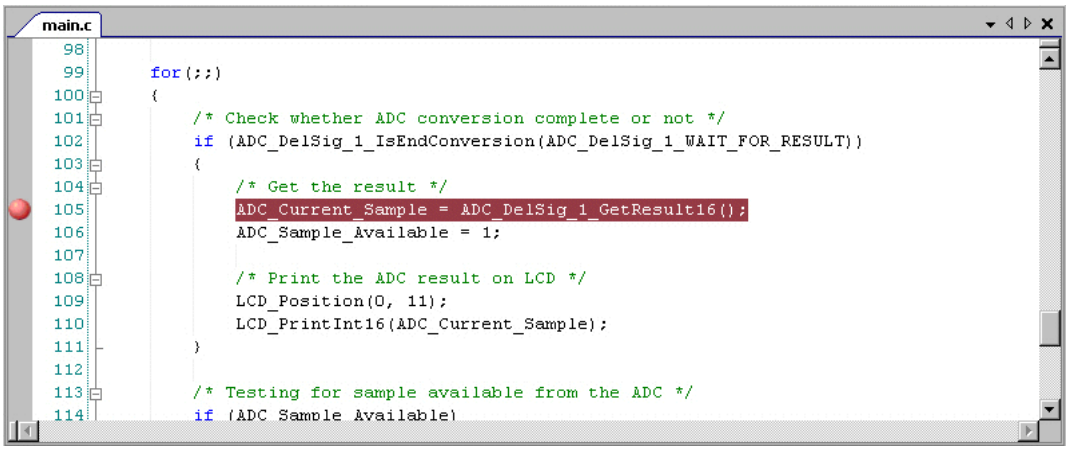
通过 Find Example Project（查找示例项目）对话框，您也可以打开用于本节的完整示例项目，即“BasicDesign”。该对话框的链接显示在 PSoC Creator “Start Page”（起始页）上。

设置断点与单步调试：

在本节中，您将设置两个断点并运行调试器来验证该代码是否按要求被执行。另外，您还将使用 Step 功能来显示查看当前的堆栈。更多有关断点的信息，请参考断点窗口中介绍的内容。更多有关步进信息，请参考调试器工具栏指令中介绍的内容。

1. 在工作区浏览器中，双击 *main.c* 文件可打开它。
2. 滚动到 for 循环部分并点击第 105 行的边缘，这样可以将一个断点插入到该代码行内。

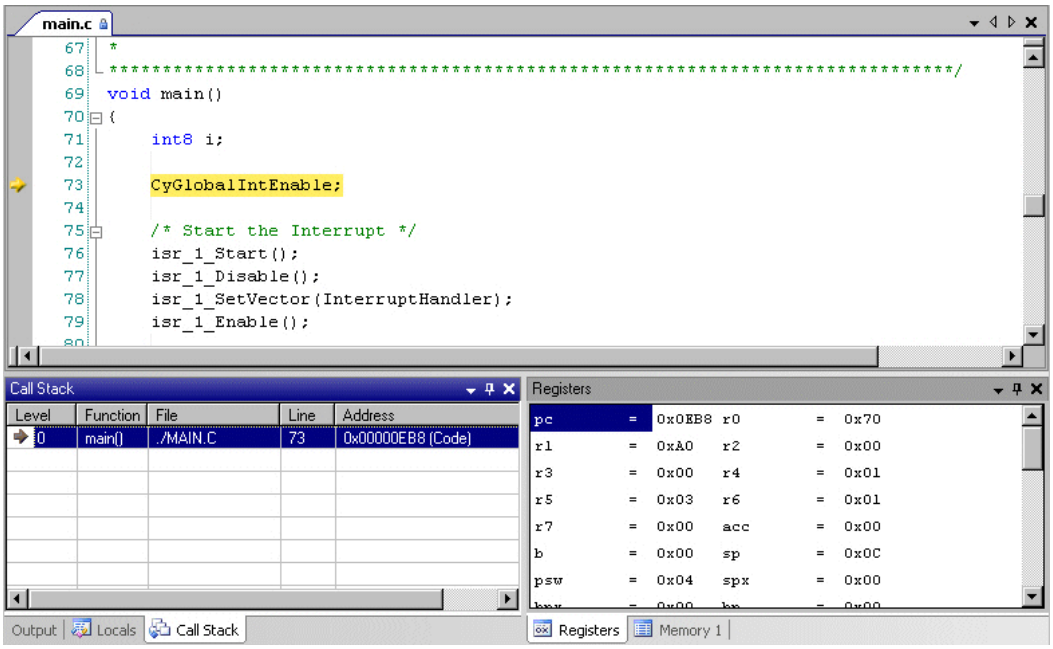
   断点指示将出现在所需代码行附近的边缘内。



3. 点击 **Debug** 启动该调试器。

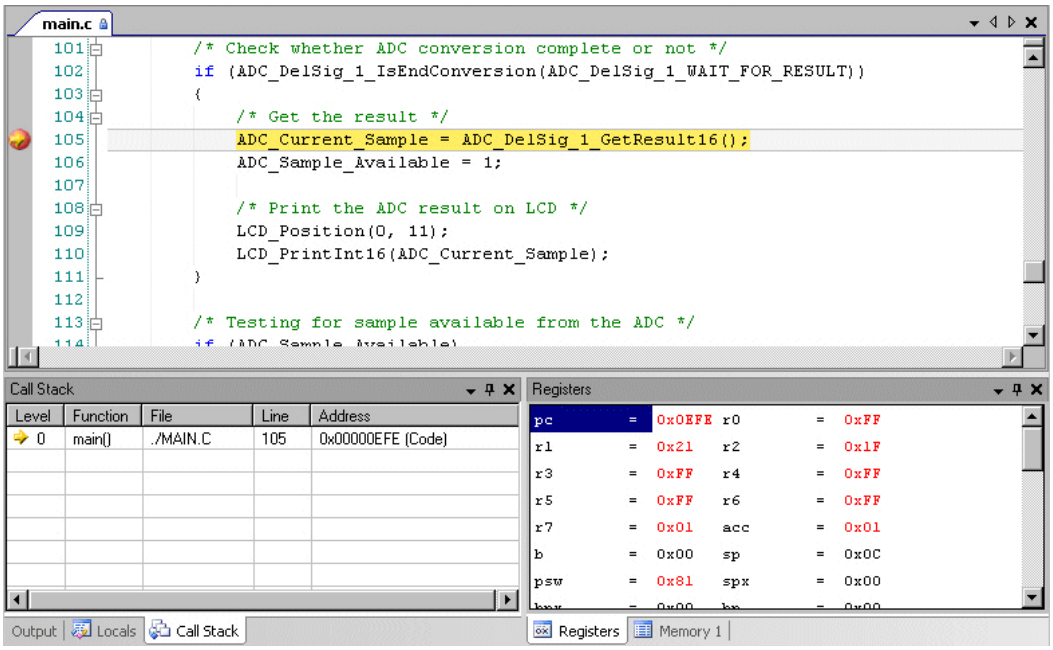
   根据调试器选项设置，该调试器运行到 Main 函数。

4. 从 **Debug > Windows** 菜单栏中打开 Call Stack 窗口。请注意，当前行指示位于 main()函数内。



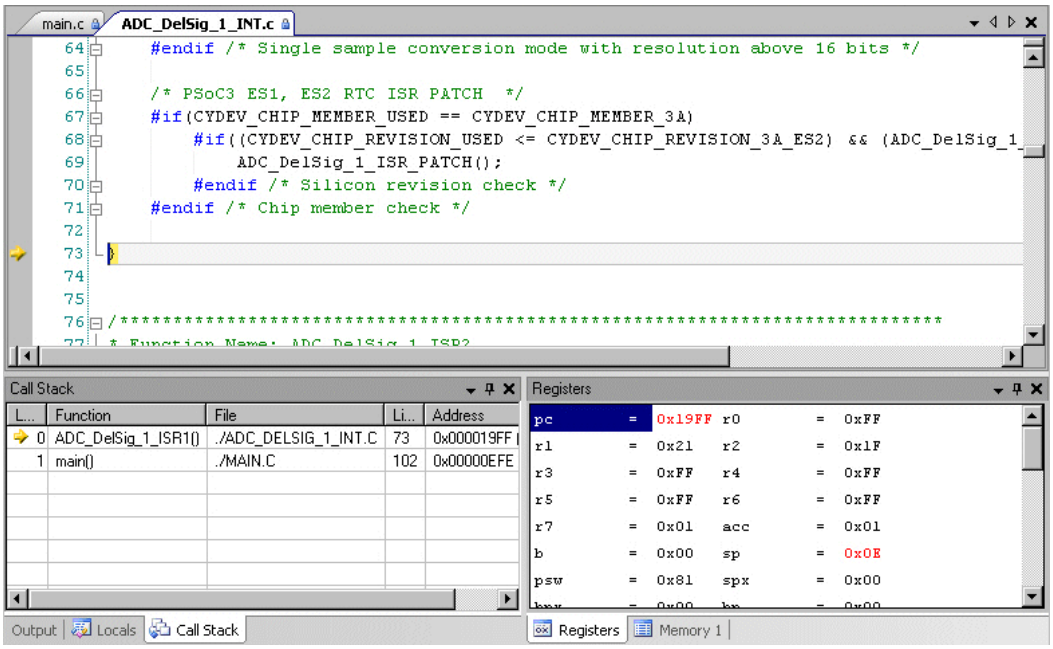
5. 点击 **Continue**（继续）。

该调试器将转移到您设置的断点处。请注意，断点图标上的当前行指示以及 Call Stack 窗口中的当前行指示。



6. 点击 **Step Into**。

该调试器打开 *ADC_DelSig_1_INT.c* 文件并停留在第 73 行。再次注意 Call Stack 窗口中的当前行指示。



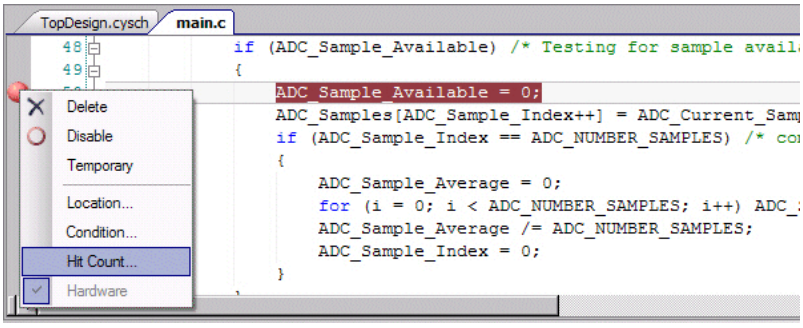
7. 通过点击停止该调试器。

现在，您确认该调试器在所需要的断点上停止后，可以通过右击边缘中的图标并选择 **Disable**（禁用）禁用该断点。

设置触发计数和变量观察点：

在本节中，您将通过设置触发计数断点和变量观察点来监控阵列的值。

1. 打开 *main.c* 文件并滚动到第 50 行。
2. 点击该边缘进行设置断点，然后右击该断点图标并选择 **Hit Count...**



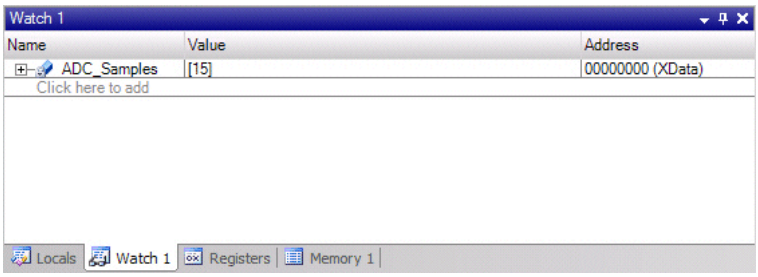
打开 Breakpoint Hit Count（断点触发计数）窗口。

3. 输入触发计数值“5”，然后点击 **OK**。
4. 点击 **Debug** 启动该调试器。

根据调试器选项设置，该调试器运行到 Main 函数。

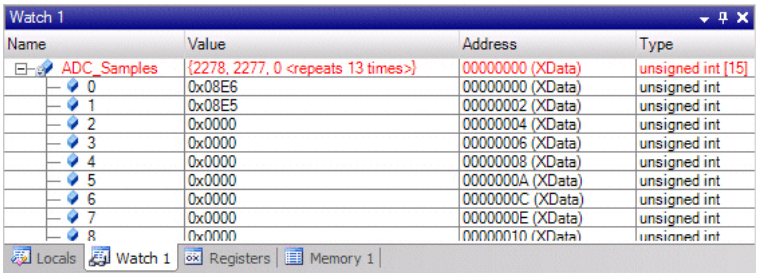
5. 右击第 51 行中的 ADC_Samples 阵列，并选择 **Add Watch**（添加查看器）。

Watch 1 窗口被打开，并显示 ADC_Samples 数组。展开该列表查看所有数组值。



6. 点击 **Continue**（继续）。

   该调试器停止在第 50 行，而且 Watch 窗口将显示带有两个值的 ADC_Samples 数组。



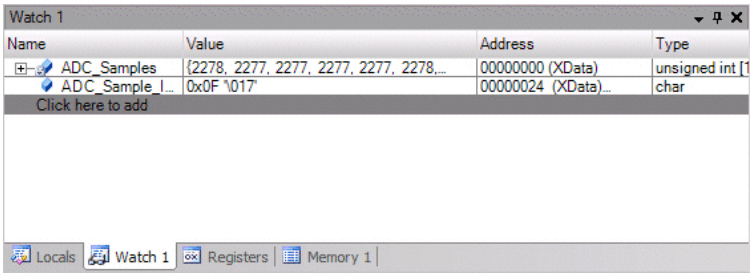
   注意：使能中断时，各断点被执行两次。这是因为该断点代码被执行前生成了一个中断并已经被处理。该中断执行完成时，处理器将返回原始代码行。这样会再次执行该断点代码。

7. 禁用第 50 行中的断点。

8. 在 *main.c* 文件中第 54 行上的 ADC_Sample_Average 内添加另一个查看器。

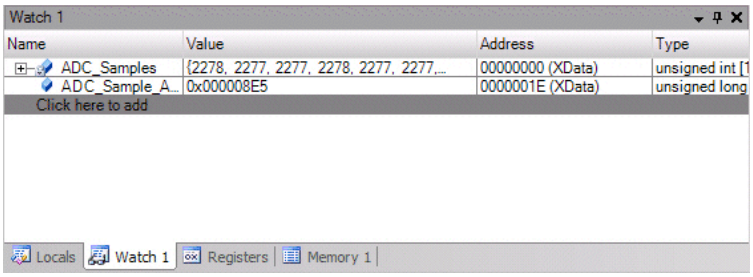
9. 右击第 56 行并选择 **Run to Cursor**。

10. 在 Watch 窗口中，请注意现在 ADC_Samples 阵列中显示了各个值，而且 ADC_Sample_Average 则是这些数值的总和。



11. 右击第 57 行并选择 **Run to Cursor**。

12. 在 Watch 窗口中，请注意 ADC_Sample_Average 现在是平均值。



更多有关 PSoC Creator 调试器的信息，请参考本文档中使用调试器部分中的各主题。

高级教程

组件库项目

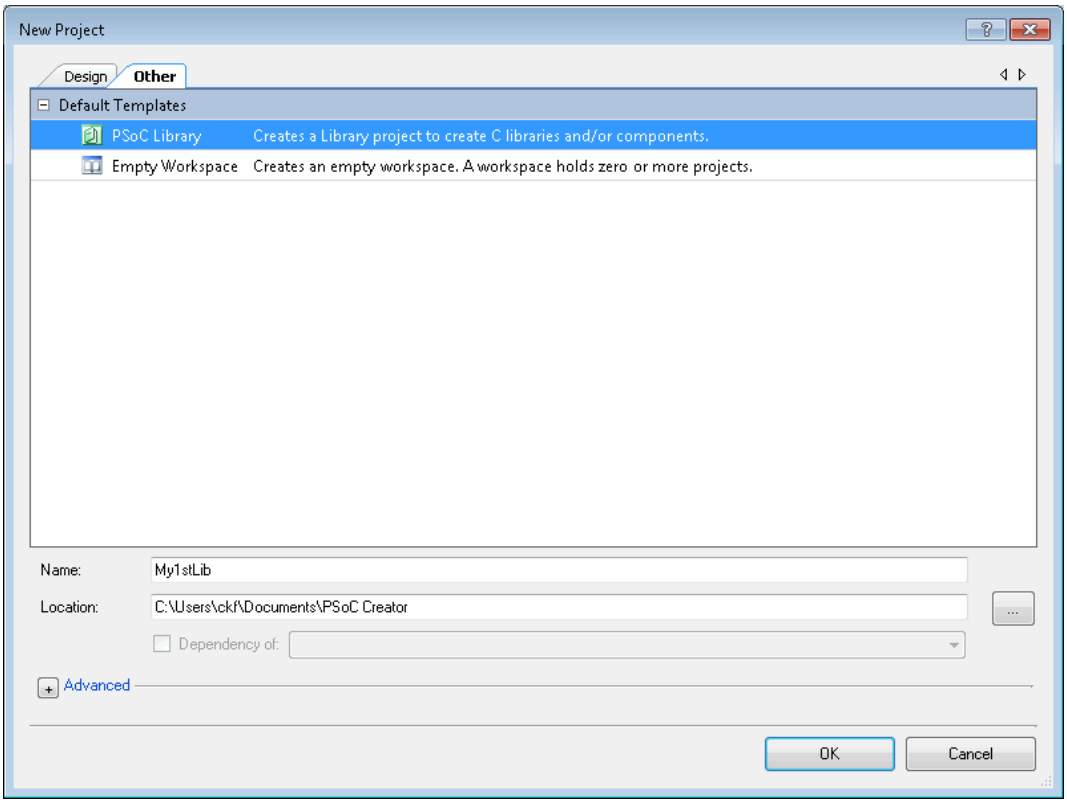
默认情况下，PSoC Creator 提供了您的设计可使用的组件库。在某些情况下，您希望创建自己的库。本节通过介绍如何创建一个 4 位移位器说明创建库的基本流程。这**只是一个示例**，用于向您介绍如何创建基本组件。更多有关创建组件的详细指导，请参考组件创建指南中介绍的内容。

注意：如果您不想创建新的项目，则可以通过使用 Find Example Project（查找示例项目）对话框打开用于本节的完整示例项目，即“LibraryComponent”。该对话框的链接显示在 PSoC Creator “Start Page”（起始页）上。该对话框打开后，请选择工作区浏览器中的 **Components**（组件）选项卡来查看组件文件。

创建新项目：

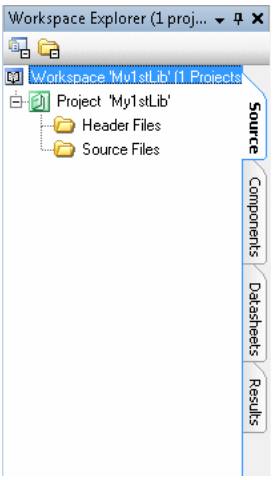
第一步是创建基本库项目：

1. 如果当前打开了一个项目或工作区，请选择 **File**（文件）菜单中的 **Close Workspace**（关闭工作区）。
2. 从 **File**（文件）菜单中，依次选择 **New > Project** 或直接点击打开 New Project（新项目）对话框。
3. 在 **Other** 选项卡下，点击 **PSoC Library** 模板。



4. 在 **Name**（名称）字段中，输入您的项目名称，例如：“My1stLib”。
5. 在 **Location**（地址）字段中，输入您想存储该项目的路径，或直接点击**[...]**导航到相应的目录。
6. 点击 **OK**。

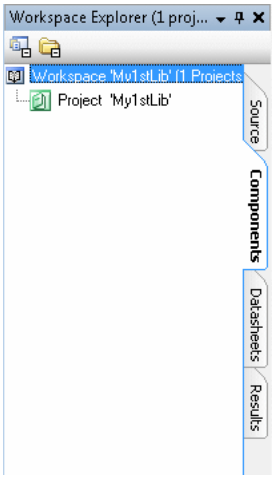
PSoC Creator 创建您的项目并将各文件和文件夹添加到 **Source** 选项卡下的工作区浏览器内。更多信息，请参考工作区浏览器部分。



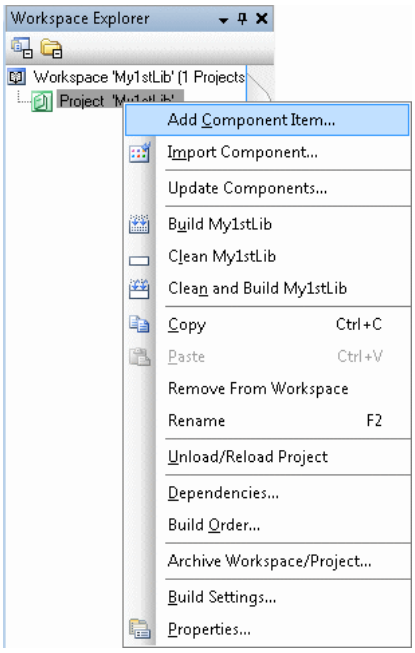
添加移位器组件：

基本库项目是创建一个组件。为了添加一个组件，您必须先添加一个符号，这是因为每个组件必须拥有一个唯一的符号：

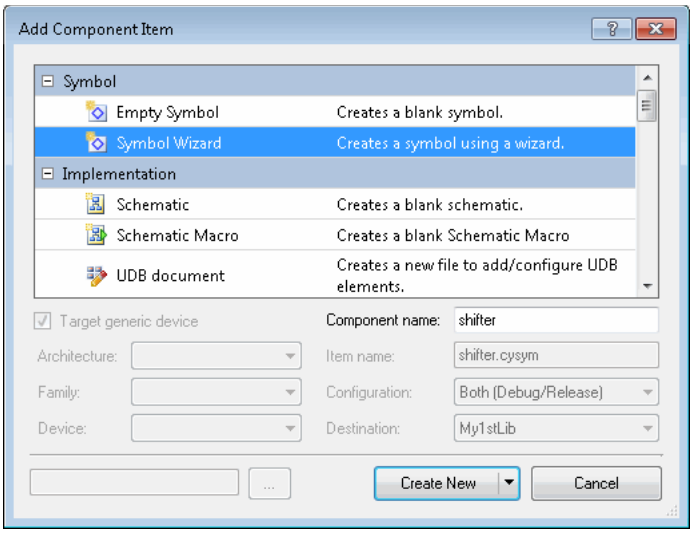
1. 点击 Workplace Explorer（工作区浏览器）窗口中的 **Components** 选项卡。



2. 在 Workspace Explorer （工作区浏览器）中，右击项目并选择 **Add Component Item...**



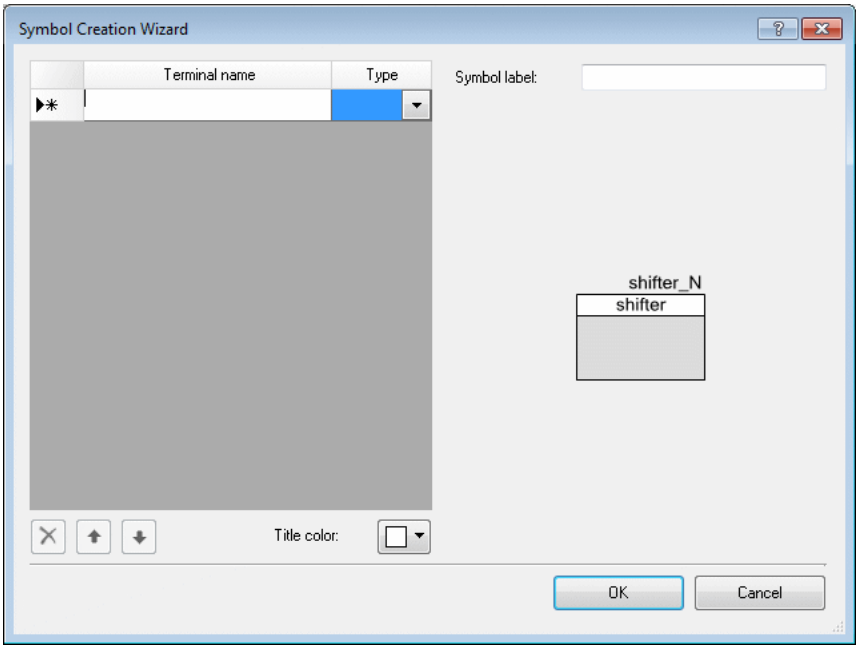
将出现 Add Component Item（添加组件项）对话框。



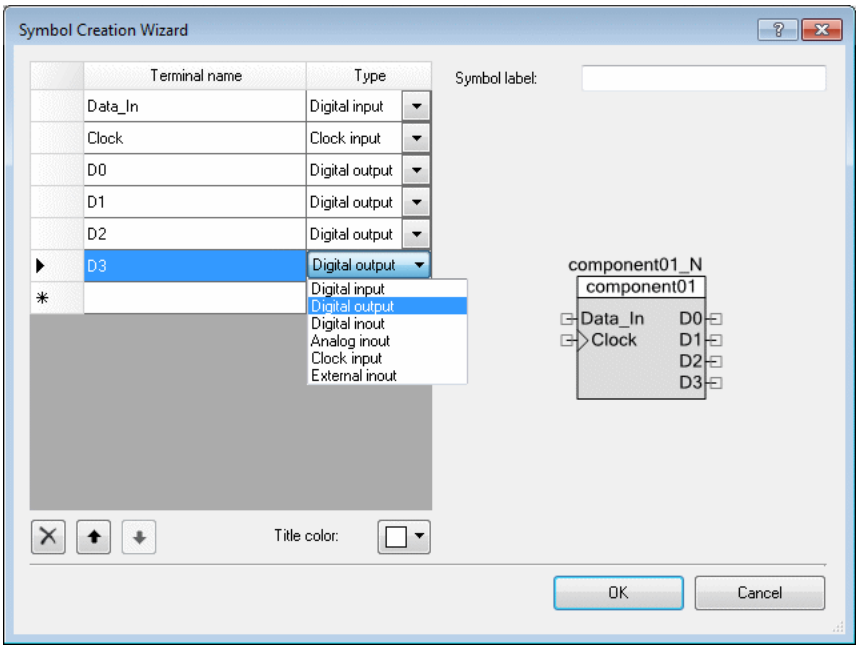
3. 在 **Symbol**（符号）项下，点击 Symbol Wizard 模板文件。
4. 在 **Component name**（组件名称）项下，输入您组件的名称，例如：“shifter”。

   注意：该符号将继续使用该组件名称，而且该名称中不能存在空格。
5. 点击 Create New。

   Symbol Creation Wizard（符号创建向导）窗口出现。



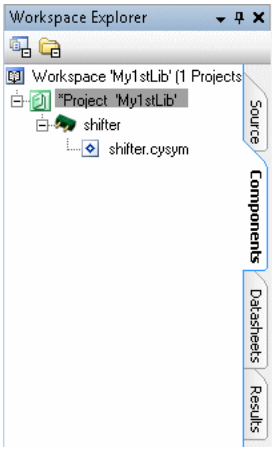
6. 通过使用该窗口中的 **Terminal name**（终端名称）和 **Type**（类型）字段创建两个输入终端（名称分别为 Data_In 和 Clock）和四个数字输出终端（名称为 D0、D1、D2、D3）。



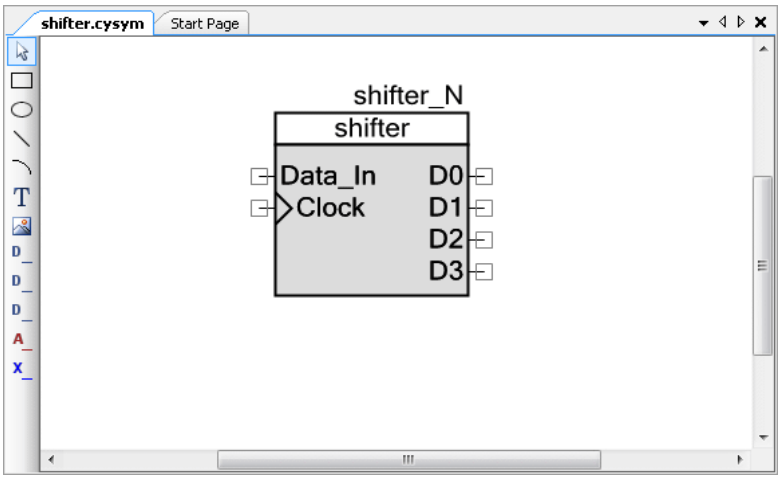
更多详细信息，请参考 Symbol Wizard（符号向导）。

7. 点击 **OK**。

您的项目显示了该组件以及添加到您项目的符号。新的组件显示在 Workspace Explorer（工作区浏览器）树内，其中符号文件（.cysym）作为唯一的组件项显示。



Symbol Editor（符号编辑器）也会打开.cysym 文件并显示已创建的符号。

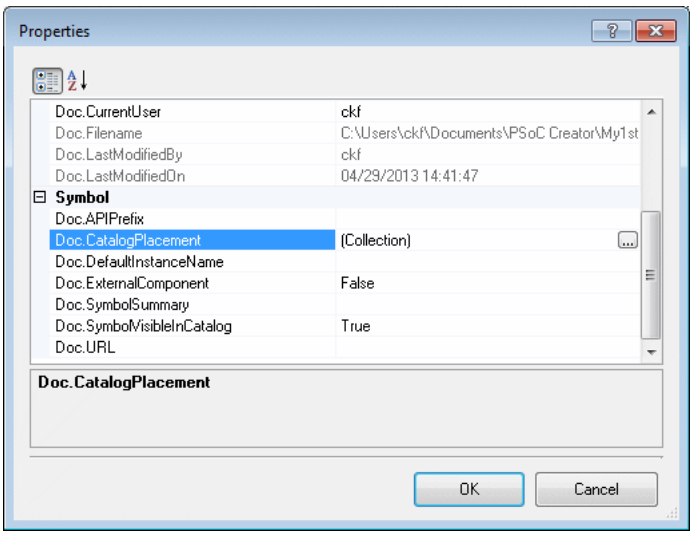


请注意，该符号上有一个 shifter_N 文本框。这是一个实例文本标签，当您在一个设计中实例化该组件时，您可以命名组件。更多有关信息，请参考文本的用法。通过使用 Properties 对话框中的 Doc.DefaultInstanceName 符号文档属性，您可以更改默认的实例名称。

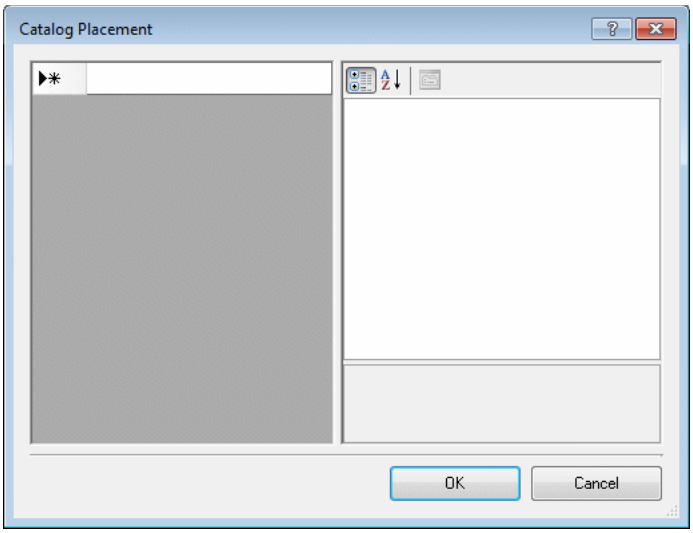
指定组件目录中的位置

完成后，各组件将显示在组件目录内。若需要，您可以控制它们的显示方式。

1. 右击符号图纸并选择 **Properties** 以打开 Properties 对话框。
2. 点击 **Doc.CatalogPlacement** 字段来显示省略符号**[...]**按钮。



3. 点击省略符号**[...]**按钮来打开 Catalog Placement（目录）对话框。



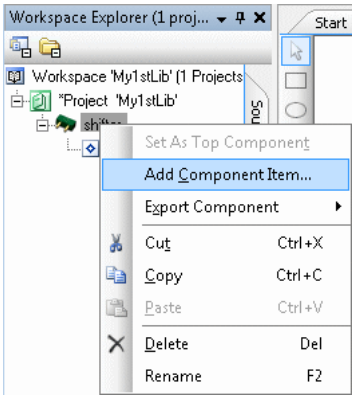
4. 点击 旁边第一行的右列，并输入：
   Example/LogicCircuits/
5. 点击 **OK** 以关闭 Catalog Placement（目录位置）对话框。
6. 点击 **OK** 以关闭 Properties 对话框。
7. 点击 **Save** 来保存您的符号文件。

更多有关该对话框的信息，请参考定义目录位置中介绍的内容。

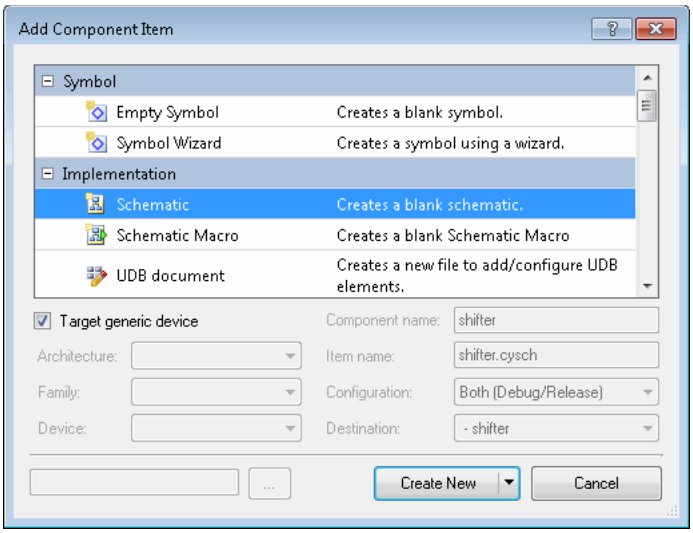
添加 Shifter Implementation

为了使您的符号显示在组件目录内，您必须为组件添加 Implementation。在本节中，我们将添加一个原理图。

1. 在 Workspace Explorer （工作区浏览器）中，右击组件并选择 **Add Component Item...**



出现 Add Component Item（添加组件项）对话框。

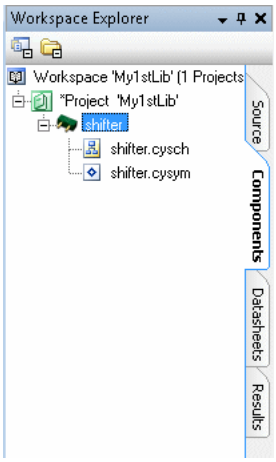


2. 在 **Implementation** 项下，点击 Schematic 图标。
3. 在 **Target** 项下，选择 **Generic Device**。

   注意：**该原理图将继续使用组件名称，不能修改该名称。**
4. 点击 Create New。

   注意：将出现 Select Sheet Template（选择工作表模板）对话框，这样您可以选择图纸模板。如果显示了该对话框，请点击需要的模板，然后点击 **OK**。

   您的项目显示了已添加的原理图。该组件显示在 Workspace Explorer（工作区浏览器）树形结构内，并且包含了原理图文件（.cysch）和符号文件。

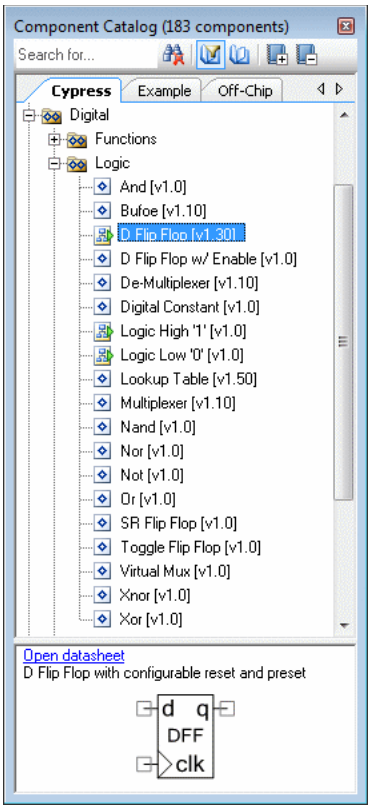


   原理图编辑器会打开.cysch 文件，并显示一个空白图纸和组件目录。更多有关信息，请参考原理图编辑器中的内容。

完成移位器原理图

现在您有一个空白的图纸，这时需要绘制该原理图来实现符号。

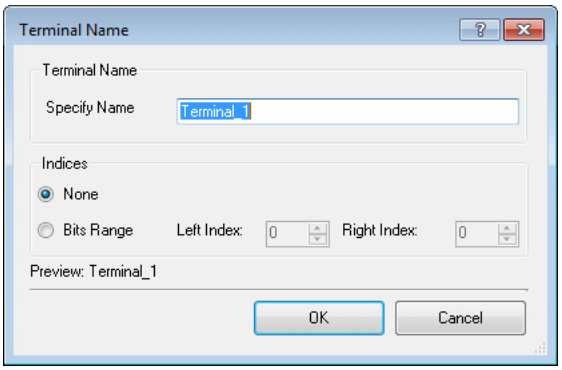
1. 在 Component Catalog（组件目录）窗口中，展开 Digital > Logic 树形结构。



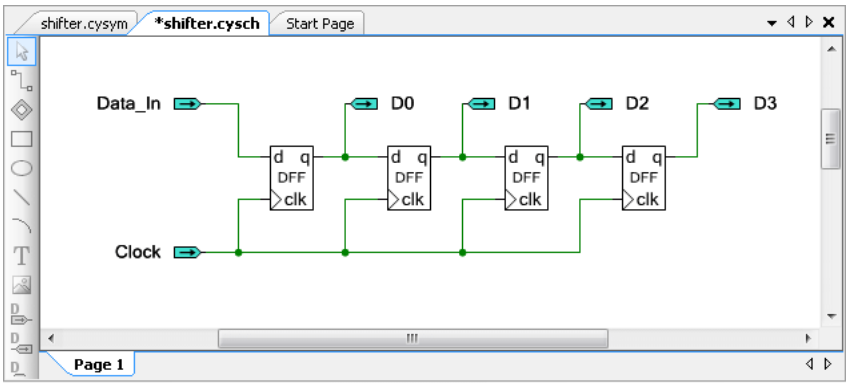
- □ 点击 D Flip Flop 并将其绘制到您的图纸上。
- □ 添加四个 D Flip Flop 组件。

2. 从 Design Elements Palette（设计元素控制板）中选择 **Digital Input**（数字输入）终端，然后点击图纸来放置该终端。

   出现 Terminal Name（终端名称）对话框。另请参考原理图终端的用法。



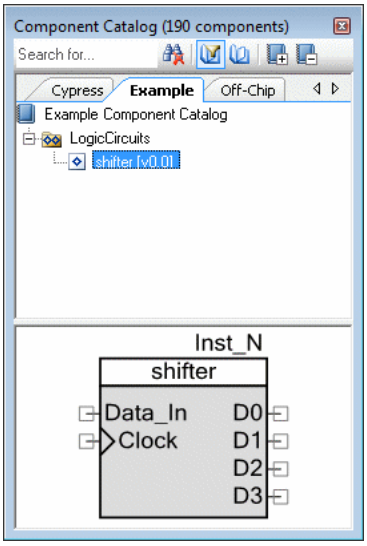
3. 在 **Specify Name**（指定名称）字段中，输入“Data_In”，然后点击 **OK**。
4. 通过重复该过程来添加另一个 **Digital Input**（数字输入）和四个 **Digital Output**（数字输出）；将这些终端分别命名为 Clock、D0、D1、D2 和 D3。
5. 点击 **Draw Wire**（绘制线）工具。
6. 按照下面的图像将各终端连接至逻辑门：



请参考连线的用法。

7. 点击 **Save** 保存您的原理图文件。

该过程完成后，该组件将显示在 Component Catalog（组件目录）窗口中的 Example（示例）选项卡内。



使用库：

您一旦创建了带一个或多个组件的库，就可以将这些组件作为您设计项目的库元素使用。欲了解如何在设计中使用库，请参考基本层级设计节中介绍的内容。

基本层级设计

通过 PSoC Creator，您可以创建并重新使用各组件。这是层级设计的基本定义；组件和设计成为其他组件和设计的构建模块。

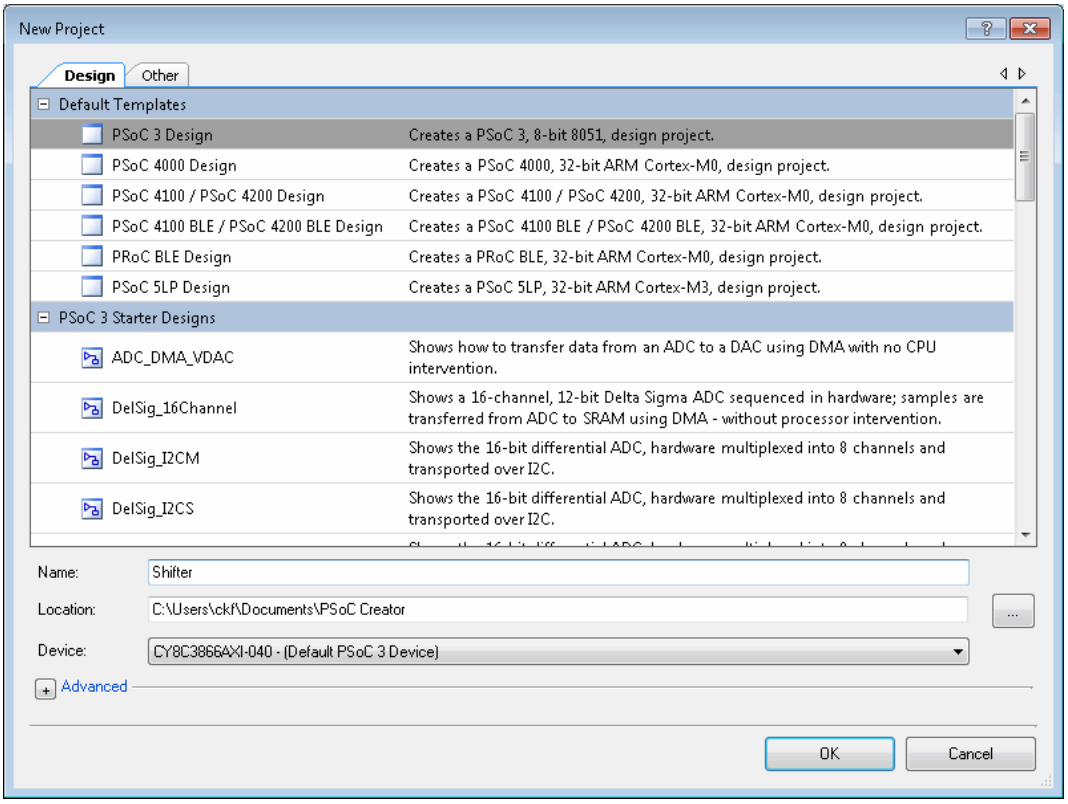
本节介绍的是如何使用前一节中创建的 LogicCircuit 库来创建 8 位移位器。它**只是一个示例**，用于向您介绍如何在一个层级设计中使用您所创建的组件。更多有关创建组件的详细指导，请参考组件创建指南中介绍的内容。

注意：如果您不想创建新的项目，那么可以通过使用 Find Example Project（查找示例项目）对话框打开本节的完整示例项目，即“Shifter”。该对话框的链接显示在 PSoC Creator “Start Page”（起始页）上。

创建新项目：

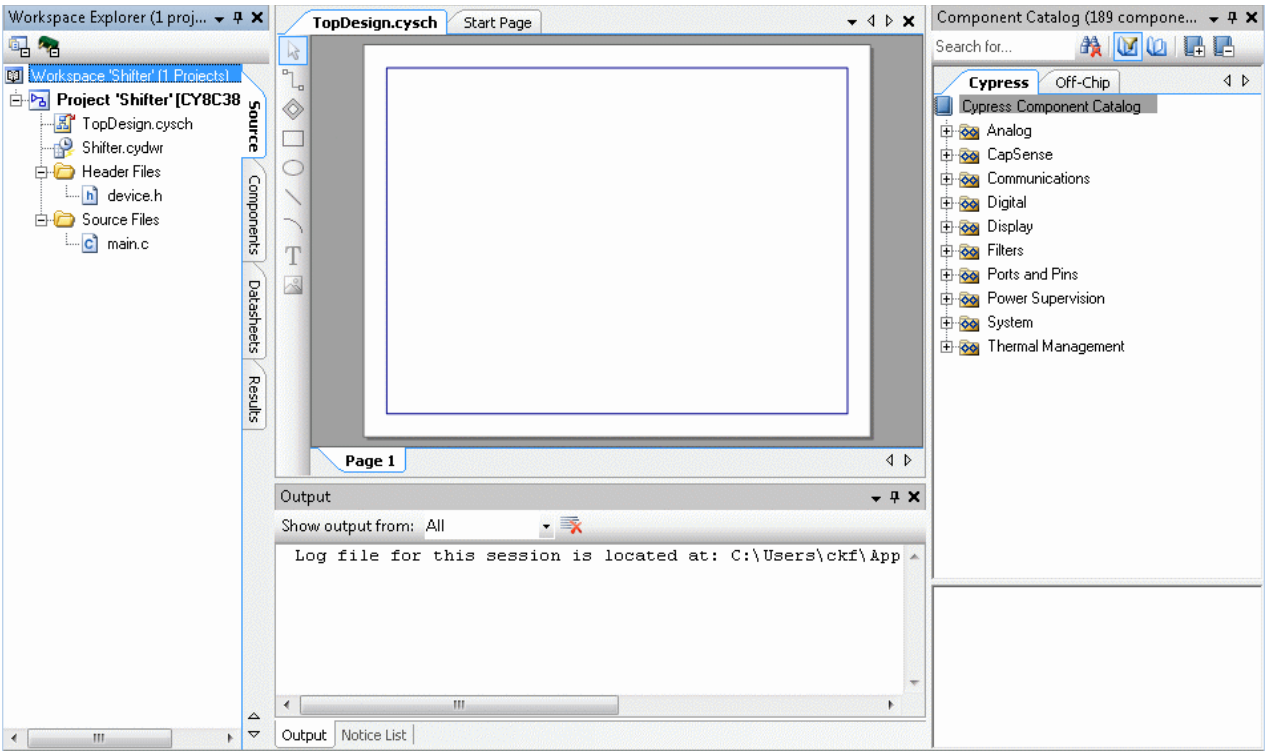
第一步是创建基本的设计项目：

1. 如果当前已经打开了一个工作区，那么请选择 **File**（文件）菜单中的 **Close Workspace**（关闭工作区）。
2. 从 **File**（文件）菜单中，依次选择 **New > Project** 或点击打开 New Project（新项目）对话框。



3. 在 Empty Templates（空模板）项下，点击 Empty PSoC 3 Design 模板。
4. 在 **Name**（名称）字段中，输入您项目的名称，例如：“Shifter”。
5. 在 **Location**（地址）字段中，输入您想存储该项目的路径，或直接点击**[...]**导航到相应的目录。
6. 点击 **OK**。

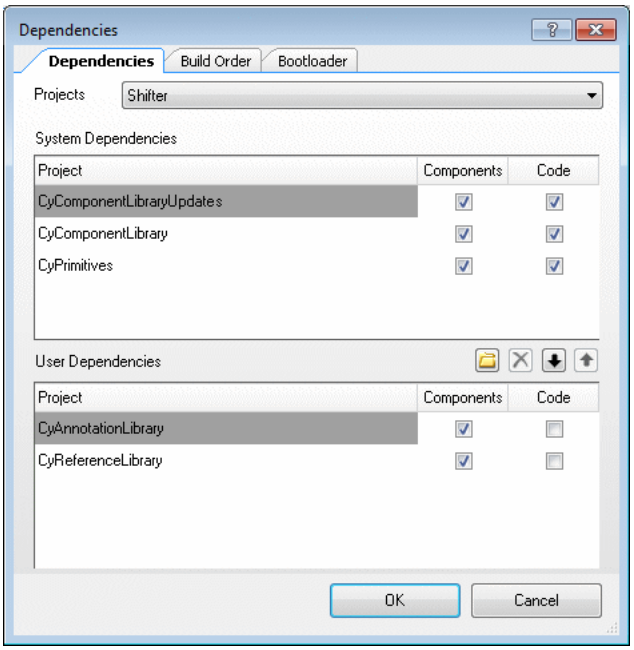
PSoC Creator 创建您的项目并将各文件和文件夹添加到 **Source** 选项卡下的工作区浏览器内。在原理图编辑器中，顶层原理图文件以文档窗口格式显示，另外 Component Catalog（组件目录）显示了您设计中使用的组件列表。



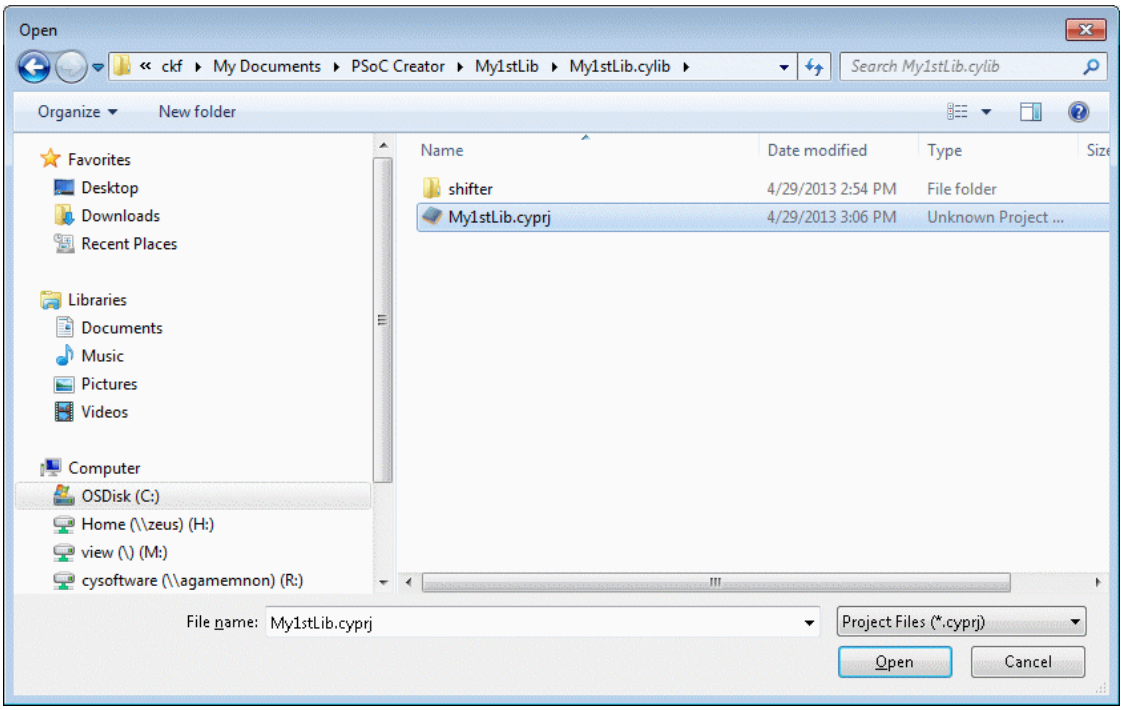
添加逻辑电路库：

默认情况下，所有新设计都有一个预定义的组件库。这些组件显示在 Component Catalog（组件目录）窗口中。要想添加逻辑电路库，您需要将库项目添加到库搜索路径内。

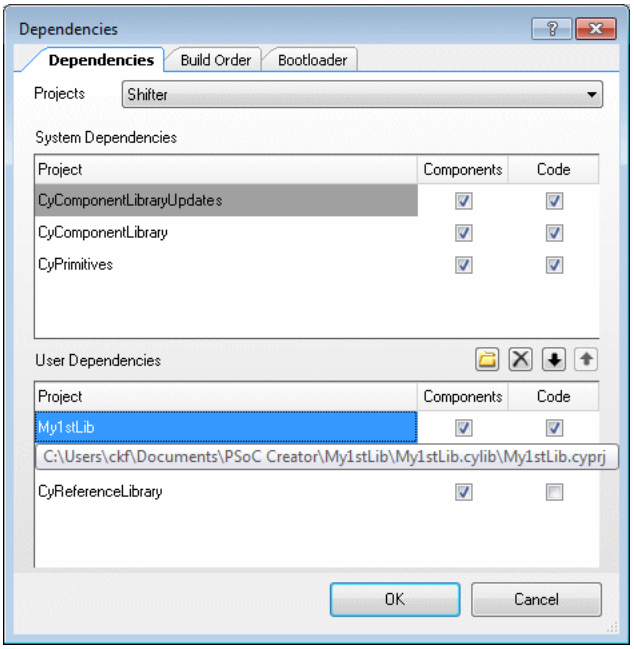
1. 从 **Project**（项目）菜单中，选择 **Dependencies...**以打开 Dependencies（依赖属性）对话框。



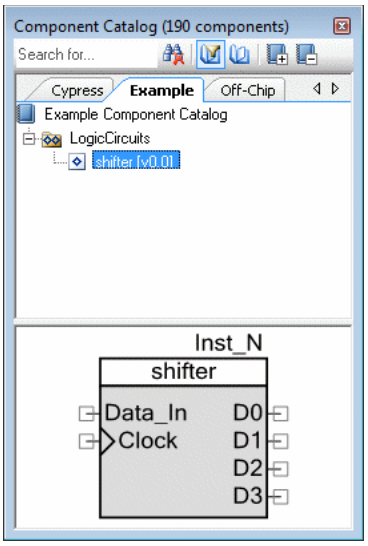
2. 在 **User Dependencies** 项下，点击 **New Entry** 以打开文件浏览器对话框，导航并寻找该库中的 *My1stLib.cyprj* 文件，然后点击 **Open**。



3. 请注意，该库项目显示在 **User Dependencies > Project** 下，点击 **OK** 会关闭 Dependencies 对话框。



现在，Component Catalog（组件目录）窗口拥有一个包含移位器组件的 Example（示例）选项卡。



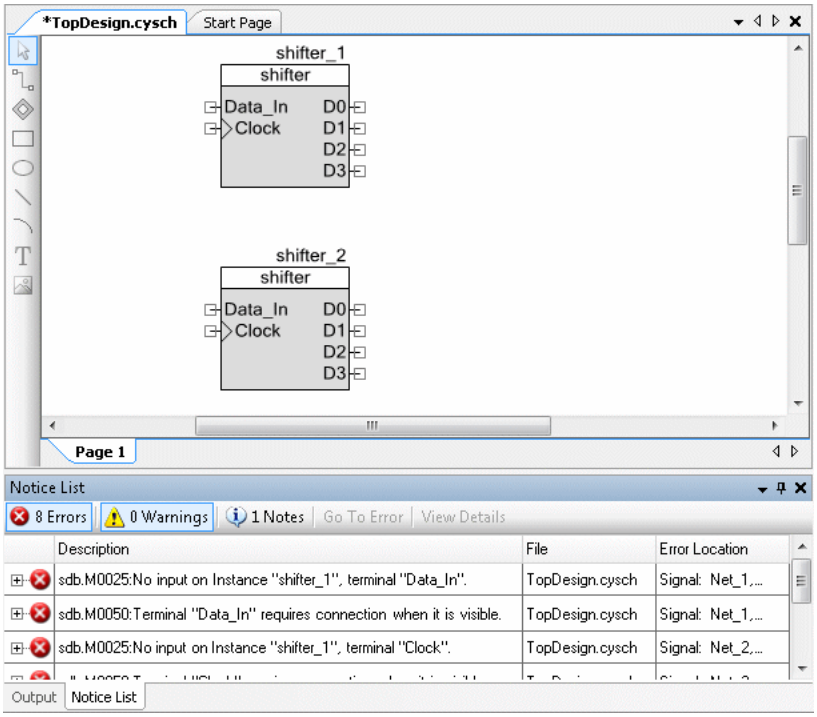
完成该设计：

您已经添加了库组件，现在您可以使用这些组件来创建进位脉动加法器的设计。

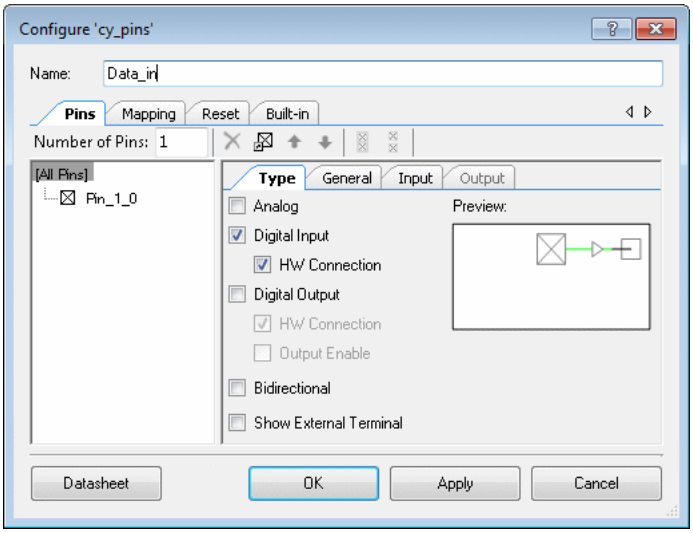
1. 在 Component Catalog 对话框中，点击 **Example** 选项卡，展开“LocicCircuits”文件夹，然后点击移位器组件并将其拖放到您的原理图图纸上；添加第二个移位器组件。

   请注意，当您放置各个组件时，Notice List（注释列表）窗口显示连接错误列表。 这些属于动态规则检察（DRC）系统的组成部分，用于告诉您设计中存在错误。完成该设计后，您应纠正该错误。更多有关该窗口的信息，请参考 Notice List（注释列表）窗口中介绍的内容。

   请注意各组件被命名为“shifter_1”和“shifter_2”。



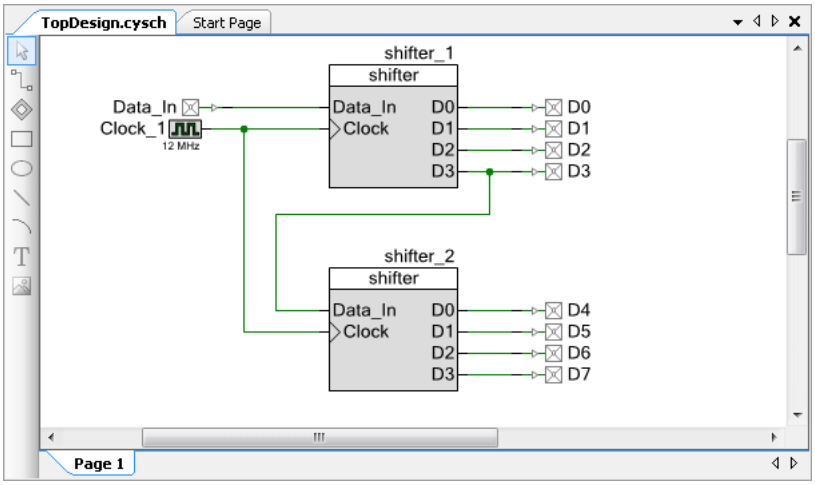
2. 在 Component Catalog（组件目录）窗口中，点击 **Cypress** 选项卡，展开“Ports and Pins”文件夹并将 Digital Input Pin（数字输入引脚）拖放到您的原理图图纸上；另外，添加八个数字输出引脚。
3. 双击 Digital Input Pin（数字输入引脚）打开 Configure（配置）对话框，并将 **Name**（名称）更改为 Data_In。



4. 点击 **OK** 关闭该对话框。
5. 将各个数字输出引脚重新命名为 D0 至 D7。
6. 在组件目录中，展开“System”文件夹，并将 Clock（时钟）拖放到原理图图纸中。
7. 对图纸上的各个组件和终端进行相应安排。
8. 点击 **Draw Wire**（绘制线）工具。
9. 将各引脚和时钟连接至移位器。

   更多有关不同技术的指导，请参考连线的用法、绘制总线及连线标签与名称中介绍的内容。

   当您完成所有连接后，您的原理图将如下图所示：



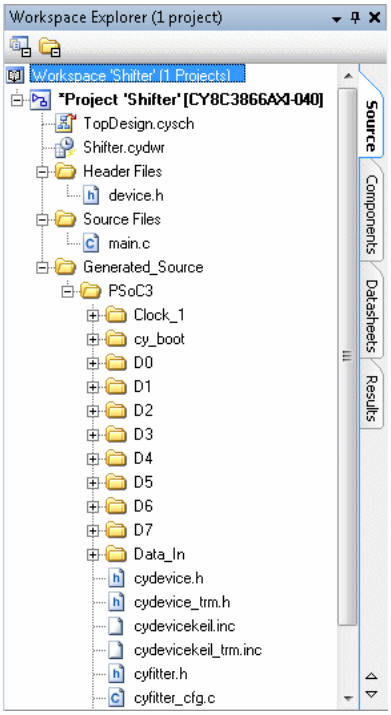
   请注意在 Notice List（注释列表）窗口中所有错误已经清除了。

提示：您可以将相同的连线复制并粘贴在不同的位置；因此，如果您在该设计的各项目之间设置相同的空格，那么复制它们将会变得更容易。

10. 从 Build（编译）菜单中，选择 Build Shifter。

    Output 窗口将出现一个信息，表示已成功进行编译。

    Workspace Explorer（工作区浏览器）显示的是一个包含生成设计文件的 Generated Files 文件夹。



关闭设计：

从基本的角度来看，该 8 位移位器是一个层级设计示例。该设计包含了两个 4 位移位器，这些移位器由各 D Flip Flop 构成。您可以使用该移位器来创建用于另一设计的组件，而且可将该设计用于另一个设计，另外还包括一些其他内容。对于使用 PSoC Creator 来创建的所有设计，您将使用相同的基本原则和工具。

要想关闭该设计（以及它的项目和工作区），请选择 **File**（文件）菜单中的 **Close Workspace** （关闭工作区）项。

如何使用 Psoc Creator

本节介绍的是各种“如何...”主题，目的是为帮助您了解如何充分利用 PSoC Creator。 本节分为多个目录，具体如下：

PSoC Creator 的通用任务：

- 创建新项目
- 打开现有的项目
- 存档工作区/项目（绑定）
- 生成项目数据手册
- 创建新文件
- 打开某个现有的文件
- 创建文件夹
- 复制项目
- 选择默认编译器
- 创建新文件
- 打开现有的文件

原理图编辑器任务：

- 创建新的原理图
- 如何使用组件目录
- 配置组件
- 连线的用法
- 绘制总线
- 原理图终端的用法
- 添加原理图页面
- 更新组件
- 导入组件

设计范围资源的任务：

- 引脚的使用
- 时钟的使用
- 中断的使用

设计输入的通用任务：

- 设计输入选项
- 文本的用法
- 如何使用文本替换
- Line（画线）的用法
- 形状的用法
- 缩放
- 滚动
- 选择设备
- 如何使用 Find Example Project （查找示例项目）对话框
- 如何使用 Notice List （注释列表）窗口

文本编辑器任务：

- 如何使用文本编辑器
- 如何使用 Find Replace（查找替换）
- 如何使用 Wildcard（通配符）
- 如何使用普通表达式
- 如何使用 Go To（转到）

调试器的任务：

- 如何使用调试器
- 如何使用编程/调试选项
- 如何使用断电窗口
- 如何使用调试器菜单指令

框架：

- 使工具窗口浮动
- 使工具窗口停驻
- 使用选项卡式文件
- 如何移动工具窗口
- 自动隐藏工具窗口

符号编辑器任务：

- 创建新符号
- 使用符号向导
- 添加组件项
- 组件终端的用法
- 创建符号参数
- 定义目录放置
- 导出组件

4 了解 PSoC Creator

PSoC Creator 提供了一个 PSoC 硬件/软件的协同设计环境，包括软件开发工具、图形设计编辑器、器件选型器以及用于项目管理等各种特性。PSoC Creator 有多个您可能不太熟悉的概念。本节可帮助您进一步了解 PSoC Creator。它包括以下各主题：

- 基本概念
- 普通任务
- PSoC Creator 框架

概念

为了进一步了解 PSoC Creator，您应该熟悉以下各术语和概念：

- 工作区/项目
- 项目类型
- 组件/实例

工作区/项目

PSoC Creator 集成开发环境（IDE）提供了两个容器（工作区和项目），用于管理您设计中的各个项目。

- **工作区** — 工作区是 PSoC Creator 中的顶层容器，它包含了您可以打开、关闭或保存的一个或多个项目。所有 PSoC Creator 工作区文件（.cywrk）中只能包含一个工作区。
- **项目** — 一个项目包含许多项，代表您的设计，如原理图、设计范围资源、源代码和十六进制文件。各项的类型会根据项目类型而发生变化。项目始终属于工作区的组成部分。您可以在现有的工作区内创建项目或者创建一个全新的项目（附带创建新的工作区）

PSoC Creator 提供了工作区文件夹，用于将相关项目分为各组，然后对这些项目组进行操作。您可以使用工作区浏览器来查看和管理您的工作区、项目和相关项目项。通过一个工作区和各项目，您能够以下列方式使用 IDE：

- 管理您工作区的设置（即各项目的设置）或管理单独项目的设置。
- 如果您只关心项目和设计，使用工作区资源管理器来处理文件管理的细节。

- 将有用的项添加到工作区而不需要为该工作区的每个项目单独添加。
- 对独立于工作区或项目的其他文件进行操作。

当您创建多个项目工作区时，被创建的第一个设计项目将默认作为启动项目。在工作区浏览器中，启动项目以粗体字显示，而且点击 Debug（调试）菜单上的 Start 按钮时，将运行该项目。在工作区内，您可以编译单个项目或多个项目。另外，您还可以指定不需要执行编译过程的工作区项目。更多有关信息，请参考编译 PSoC Creator 项目中介绍的内容。

另外，请参考：

- 项目类型
- Workspace Explorer（工作区浏览器）
- 编译 PSoC Creator 项目

项目类型

一个 PSoC Creator 项目包含了许多项，如原理图、组件、设计范围资源、源代码和十六进制文件。PSoC Creator 提供了两种类型的项目：设计项目和库项目。

- **设计项目 —** 设计项目用于创建和修改设计。对于一个设计项目，在原理图中选择并配置各组件。接下来，设置设计范围资源，如时钟和中断等。然后，为您的应用程序编写 C 代码。最后，编译（与调试）该项目，以生成十六进制的文件并编程该器件。当您创建设计项目时，PSoC Creator 将创建项目/工作区文件、目录结构、顶层原理图、*main.c* 文件和一个设计范围资源文件（.cydwr）。
- **库项目 —** 库文件是一个或多个组件和相应源代码的集合。凭借库项目，您可以开发构成一个设计且可以在许多设计中重新使用的各组件。另外，库项目还可以用于创建链接至设计的静态库。每一个库可提供这两个功能中的一个（或同时提供两个功能）。组件开发包括创建图形符号、定义各参数和指定验证要求。当您创建库项目时，PSoC Creator 将创建项目/工作区文件和目录结构。库项目包含在 PSoC Creator 的 Dependencies 选项卡中，通过它可以确定组件目录中的哪个组件适用于您的设计。

这些项目名称的扩展名为“.cyprj”，如 *ProjectName.cyprj*。另外，这些项目文件必须始终位于一个名为 *ProjectName.cydsn*（设计）或 *ProjectName.cylib*（库）的目录内。

另外，请参考：

- 组件
- 组件目录
- Dependencies（依赖属性）

组件/实例

一个组件是各文件（如符号、原理图、API 和文档）的集合，它定义了 PSoC Device 中的功能。例如：定时器、计数器和复用器。实例是从组件目录中选择且用于设计的组件。某个设计中可以包含许多组件复制或实例，这是因为已选的器件能够支持该功能。您也可以创建或重新使用自己的组件/实例。

构成组件/实例的文件包括以下各项：

- **符号 —** 一个符号包含了组件的基本定义。它包含了组件目录中的顶层图片以及参数定义。一个组件中可能只有一个符号。
- **原理图 —** 原理图定义了如何实现一个组件。原理图可适用于所有 PSoC 器件或仅特定于架构、器件系列和/或器件。
- **API —** 应用编程接口。API 定义了如何使用 C 代码与组件交互。这些 API 可适用于所有 PSoC 器件或仅特定于架构、器件系列和/或器件。
- **Verilog —** 可以使用 Verilog 来定义组件的功能（该组件在 Verilog 文件中实现）。在任何级别的组件中仅包含一个 Verilog 文件。可以在不同级别的组件（如架构、产品系列和器件级别）中找到 Verilog 文件，这些文件不相互参照。
- **控制文件 —** 控制文件包含了代码生成模块的指示。更多有关信息，请参考控制文件和 Directives（指令）。
- 文档 — **组件的文档通常是它自己的数据手册。**
- **CyPrimitive** — CyPrimitive 是一个基本组件项，如逻辑门、中断或 DMA。

  注意：不要求符号是一个基元 — 它可以是某些器件的基元，但对于其他器件，该符号由逻辑和软件实现。

另外，请参考：

- 组件目录
- 控制文件
- Directives（指令）

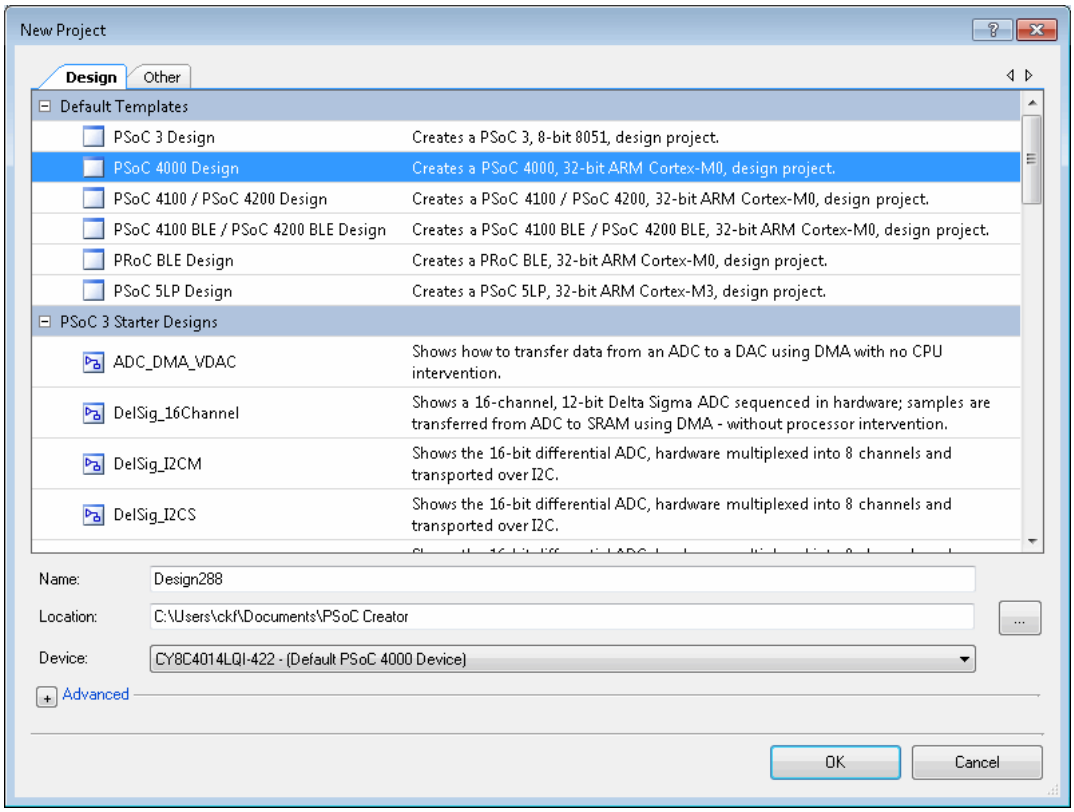
普通任务

下面显示的内容是您要在 PSoC Creator 中执行的普通任务：

- 创建新项目
- 打开现有的项目
- 添加新工作区/项目项
- 添加新的组件项
- 存档工作区/项目
- 另存项目
- 复制项目
- 创建新文件
- 打开某个现有的文件
- 创建文件夹

创建一个新项目

New Project（新项目）对话框用于创建新 PSoC Creator 项目。



使用该对话框进行下面的操作：

- 创建默认的设计或入门设计
- 选择将要创建的项目类型；请参考项目类型
- 指定项目名称和位置
- 选择创建新的工作区还是将项目添加到现有的工作区内；请参考工作区/项目
- （为设计项目）选择器件和工作表模板

要打开该对话框，请执行下面的操作：

从 **File**（文件）菜单中，依次选择 **New > Project...** 或者直接点击 PSoC Creator Start（起始）页面上的 **Create New Project**（创建新项目）项。

要创建一个新项目：

1. 选择合适的项目类型。
   - 点击 **Design**（设计）或 **Other**（其他）选项卡。
   - 选择项目类型：设计、库或空工作区。

每个模板旁边都有一个简要的说明。

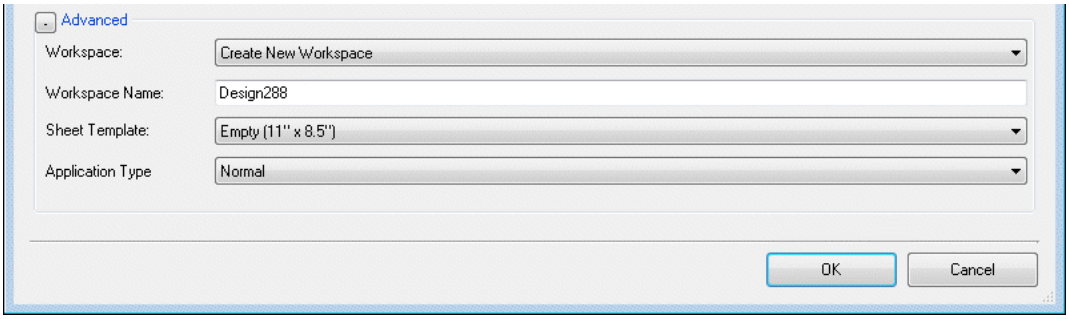
2. 输入项目**名称**和**位置**。

   注意：对于库项目，**Location** 项下有一个 **Dependency of** 复选框。如果当前工作区内存有任何项目，则该复选框将被使能。勾选该复选框时，会使能当前工作区中所有项目的下拉列表。新创建的库文件将成为已选项目的一个相关项。更多有关信息，请参考依赖属性。

3. 对于 **Default Design Template**（默认设计模板），请选择一个**器件**将其用于您的设计项目中。各选项包括：
   - 选择最后使用的器件
   - 为已选架构选择默认器件
   - 启动器件选型器

   注意：Device（器件）选项不适用于入门设计。

4. 若可以，请点击 **Advanced** [+]按钮以访问其他选项。



   Workspace（工作区）和 **Workspace Name**（工作区名称） — 选择创建新的工作区还是将该项目添加到现有的工作区内，并指定工作区名称。

   对于 **Default Templates only**（仅默认模板）：
   - **工作表模板**（设计项目） — 为您的原理图图纸选择需要的模板。参考选择工作表模板
   - **应用类型** — 允许您创建某个项目，如普通、Bootloader、Multi Bootloader 或 Bootloadable 项目。更多有关 Bootloader 项目的信息，请参考 Bootloader/Bootloadable 组件数据手册。

5. 点击 **OK**。
   - 如果您创建一个设计项目，则 PSoC Creator 将创建各个文件并默认打开原理图编辑器。
     - **除了** PRoC BLE 外，所有器件的原理图图纸都是空白的，这样用户可以添加并配置组件，最终完成设计。
     - 对于 **PRoC BLE** 器件，每个器件有一个自定义的原理图图纸，该图纸包含了放置并配置好的组件。这样，可以根据已选的特定 PRoC BLE 器件非常容易地进行设计配置。
   - 如果您创建一个库项目或空的工作区，那么 PSoC Creator 将在工作区浏览器中创建库或空工作区基础设施。

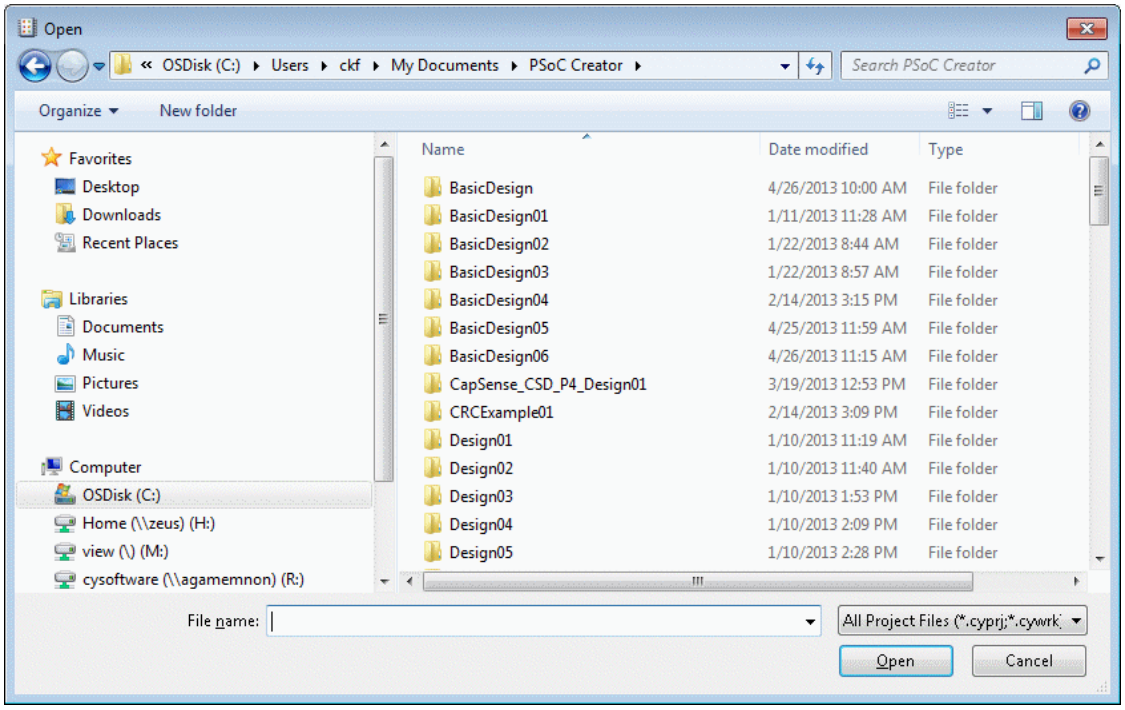
另外，请参考：

- 项目类型
- 工作区/项目
- 器件选型器

- 选择工作表模板
- 原理图编辑器
- 工作区浏览器
- 打开现有的项目
- Dependencies（依赖属性）

打开现有的项目

通过 Open Project 对话框，可以打开现有的 PSoC Creator 工作区/项目。



使用该对话框浏览并选择想要打开的工作区文件(.cywrk)或项目文件(.cyprj)。您可以打开自己创建的项目，也可以通过 PSoC Creator 打开某一个示例项目。

打开该对话框：

从 **File** 菜单中，依次选择 **Open > Project/Workspace…** 或者点击直接 PSoC Creator 起始页上的 **Open Existing Project** 项。

打开某个项目：

1. 导航到需要打开的项目所在的相应目录。
2. 选择所需项目，并点击 **Open**。

这样，可以在 PSoC Creator 中打开该项目，并且该项目会显示在 Workspace Explorer 中。

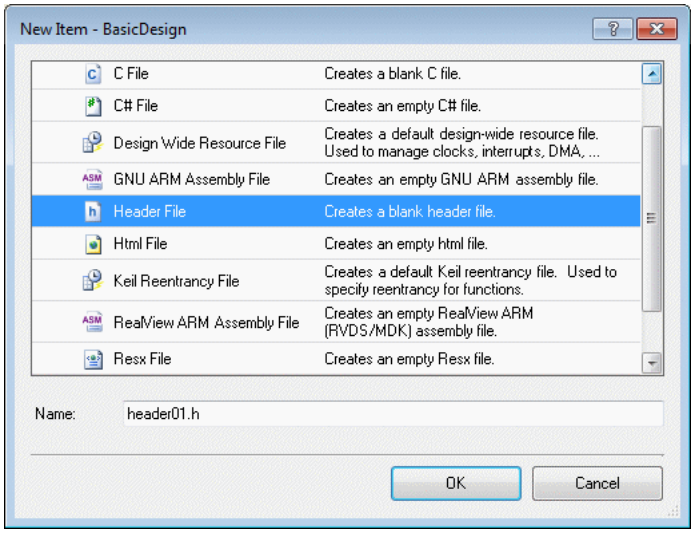
注意：通过使用起始页或 **File** 菜单上的 **Recent Projects** 项，同样能够打开最近项目，而不需要通过该对话框。

另外，请参考：

- 工作区/项目
- 工作区浏览器
- 创建新项目
- 查找示例项目

添加新工作区/项目项目：

通过 New Item（新项目）对话框，您可以将某个新项目添加到您的工作区/项目内。



该对话框提供了您能够添加的各个项的类型的模板。构建过程中，将编译各个源文件，并忽略非源文件。

注意：您不能通过该对话框创建新的组件项目。而要使用 Add Component Item（添加组件项目）对话框。

要打开该对话框：

1. 在 **Source** 选项卡下的 Workspace Explore 中，选择您想要添加的是工作区、项目还是文件夹。
2. 您可以从 **Project** 菜单中，选择 **New Item...** 项。

   或者通过右键点击某个工作区、项目或文件夹，然后从右键菜单中依次选择 **Add > New Item**。

添加项目：

1. 在 **Templates** 区域中，选择您想要添加到工作区/项目中的新项目类型的图标。当前可用的模板包括：
   - **8051 Keil Assembly File**（8051 Keil 汇编文件）—用于创建一个汇编文件，当您选择 Keil 工具链时，该文件将被编译。
   - **C File**（C 文件）— 用于创建一个标准的 C 源文件。
   - **C# File**（C#文件）— 用于选择标准的 C sharp 源文件。

- **Design Wide Resource File**（设计范围资源文件）— 创建用于编辑设计级资源（如中断、时钟，等等）的文件。
- **GNU ARM Assembly File（GNU ARM 汇编文件）**— 用于创建一个汇编文件，当您选择 GNU ARM 工具链时，将编译该文件。
- **Header File**（头文件）— 用于创建一个标准的 C 头文件。
- **HTML File**（HTML 文件）— 创建一个 HTML 文件用于文档目的。
- **Keil Reentrancy File（Keil 可重入文件）**— 用于将 API 标记为可重入函数。请查看 PSoC 3 中的可重入代码。
- **RealView ARM Assemly File**（RealView ARM 汇编文件）—用于创建一个汇编文件，当您选择 RealView ARM 工具链时，将编译该文件。
- **Resx File**（Resx 文件）— 用于创建一个资源文件。
- **Text File**（文本文件）— 用于创建一个空白的文本文件。
- **XML File**（XML 文件）— 创建一个空白 XML 文件用于任意目的。

可通过该对话框右上角的两个按键更改图标大小。**Templates**（模板）区域下方的文本框显示了每个组件的简要说明。

2. 请在 **Name** 字段中指定该项目的文件名。
3. 点击 **OK**。

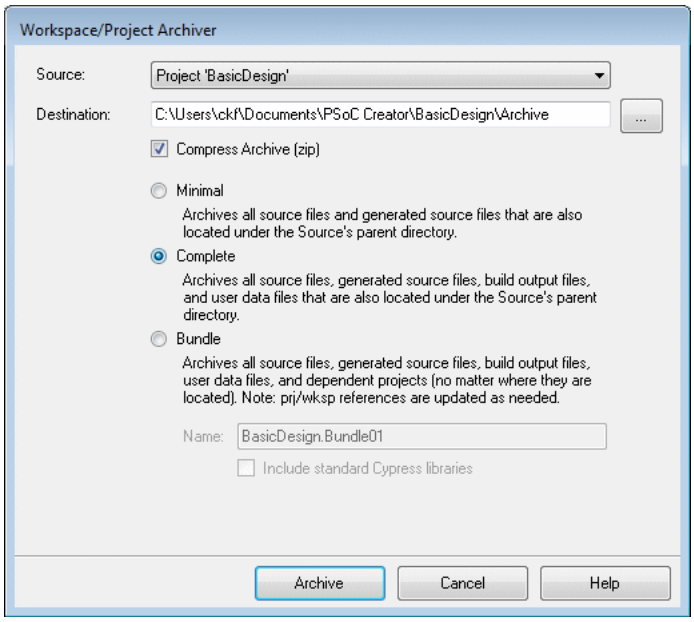
该项被添加到您在 Workspace Explorer 中所选择的位置。同时，该项中相应的空白文件将以选项卡式文档的形式打开在 Text Editor（文档编辑器）中。

另外，请参考：

- 工作区/项目
- 添加新的组件项目
- 工作区浏览器
- 文本编辑器

存档工作区/项目

通过 Workspace/Project Archiver（工作区/项目存档）对话框，您可以存档整个工作区或当前工作区中的某个项目。此外，您还可以使用该对话框来捆绑整个工作区，包括相关项目在内。可以压缩/不压缩存档项目。只能存档工作区/项目中已经注册了的文件和由 PSoC Creator 生成的文件。



要打开该对话框：

1. 打开您要存档的 PSoC Creator 工作区/项目。
2. 使用下列某种方法：
   - 从 Project 菜单中选择 Archive Workspace/Project...项。
   - 右键点击 Workspace Explorer 中 **Source** 选项卡下的某个工作区或项目，然后选择 **Archive Workspace/Project...**项
   - 选择 File 菜单中的 Create Workspace Bundle…项。

注意：如果在工作区中存在某个已被修改的文件，在该对话框打开前，会提示您保存该文件中所更改的内容。

存档某个项目/工作区：

1. 选择要存档的 **Source**（源）工作区或项目。
2. 指定项目存档所在的 **Destination**（目的地）。可以键入该项目相应的路径，或点击**[...]**，然后浏览到相应的目录。
3. 勾选 **Compress Archive**（压缩存档）选框，以创建一个包含了该存档项目的压缩文件；若不想压缩该存档项目，请取消勾选该选框。
4. 存在下面三个存档等级以供选择：
   - **Minimal**（最小）— 该等级包含了项目源文件和所生成的源文件。只有非外部文件被存档。非外部文件是指位于硬盘上存档源父目录中的文件。
   - **Complete**（全部）— 该等级包括项目源文件、所生成的源文件、所有派生文件（构建输出文件）以及用户数据文件。只有非外部文件被存档。非外部文件是指位于硬盘上存档源父目录中的文件。
   - **Bundle**（项目组）— 该等级包括项目源文件、所生成的源文件、所有派生文件（构建输出文件）、用户数据文件以及附属项目。该级别**包含**外部文件。此外，项目/工作区文件的依赖成分/链接将被更新，以作为项目/文件存档副本的参考使用。这样是为了可以在任何地方打开某组项目，并且该组项目具备所有性能（即包含了它的所有的参考内容）。

     如果您选择了“项目组”选项，那么相应的工作区始终被存档。如果选择某个工作区将其作为源使用，

它将被存档。但如果选定了某个项目，那么只会创建包含该项目的新工作区，并且只有该工作区被存档。这是因为该工作区中存储了某些必要的项目相关内容。

- **Name**（名称）— 输入某个名称，以重命名所存档的工作区。如果该文件被压缩，那么输入的文件名将作为压缩文件的名称。
- **Include standard Cypress libraries**（包含标准赛普拉斯库）— 勾选该选框，会存档所有赛普拉斯库（CyPrimitives 和 CyComponentLibrary）的副本。这样，各相关内容将被添加到所存档的副本中。在使用机械上所安装的标准赛普拉斯库前，将使用这些相关内容。

5. 点击 **Archive**。该对话框显示了存档过程，并报告该过程是否成功。
   - 如果操作成功，将出现 **Open archive in Windows Explorer**（在 Windows Explorer 中打开存档）选框，以便在点击 **OK** 后，可以打开存档项目所在的位置。
   - 如果操作失败，将显示一条错误信息，说明失败的原因。
6. 点击 **OK**，关闭该对话框。

另外，请参考：

- 工作区/项目
- 将项目存储为
- 复制项目

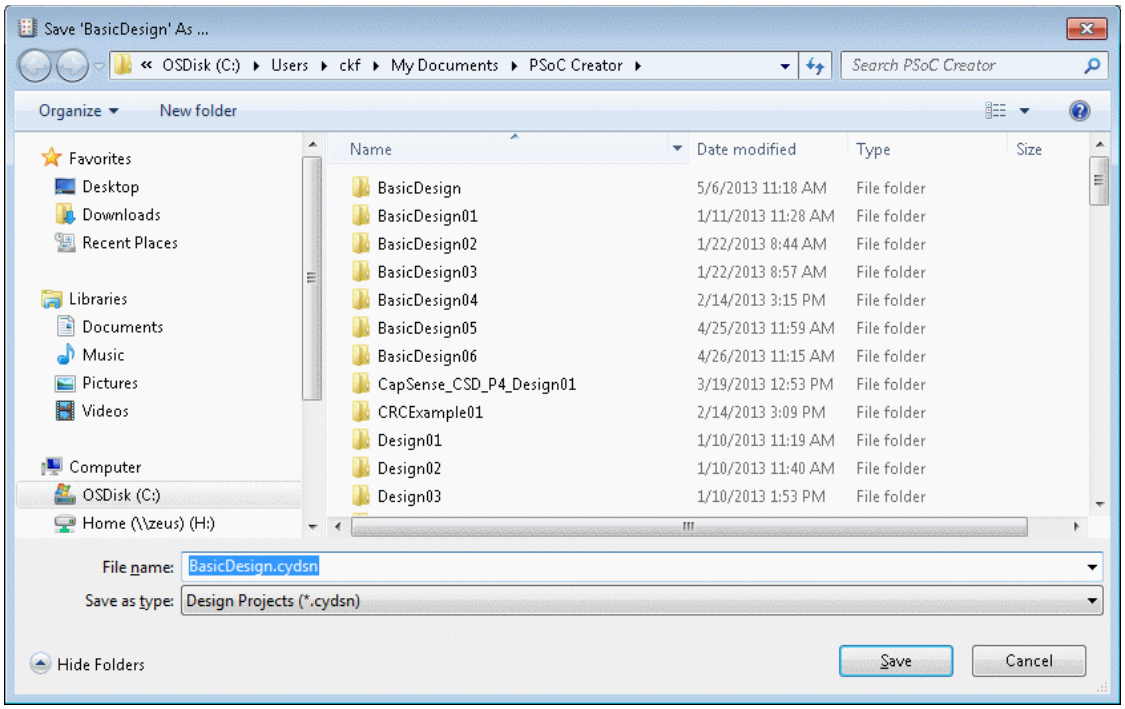
将项目存储为

使用 PSoC Creator，您可以使用别的名称在同一个位置内存储某个项目。您也可以在其他位置以任意名称存储该项目。如果您需要备用某个项目，但不想使用 Archiving tool（存档工具），那么该性能非常有用。

将项目另存为：

选择某个项目，然后选择 **File** 菜单栏中的 **Save <project> As...**项。

将出现标准的 Windows Save As 对话框。您可以在该对话框中选择想要保存的项目名称及保存的位置。



另存为：

- 存档项目
- 复制项目

生成一个项目数据手册

项目数据手册是指由 PSoC Creator 项目生成的 PDF 文档。该手册包括：

- 所选定的 PSoC 芯片架构概况
- 所使用的芯片资源列表
- 引脚
- 设计范围资源设置（时钟、中断、DMA、闪存安全性等）
- 原理图说明书
- 组件和参数设置

项目数据手册是您完整设计的快捷简介。该手册是指负责配置 PSoC 设计的硬件部分的项目团队和编程该设计的固件团队内部使用的文档。如果固件团队使用的是 IDE，而不是 PSoC Creator，并且不能直接访问 PSoC 设计配置数据，那么该手册非常有用。

生成项目数据手册。

1. 用于生成数据手册的项目必须是一个有效的项目。根据需要，右键点击 Workspace Explorer 中 Source 选项卡下的项目，然后选择 Set as Active Project（设置为有效项目）项。

2. 如果未编译项目，或自从最后一次编译后，该项目发生了更改，那么，这时候要重新编译该项目。
3. 点击 Build 菜单，然后选择 Generate Project Datasheet（生成项目数据手册）项。

   如果您在更新编译情况前尝试生成项目数据手册，则 PSoC Creator 将显示一条信息，通知您可以重新编译该项目。

新建的数据手册将显示在 Workspace Explorer 中的 Datasheets 选项卡下。要想打开该文档，请双击它。

另外，请参考：

- 工作区浏览器
- 编译 PSoC Creator 项目

复制项目

使用 PSoC Creator 时，您可以在工作区内复制某个项目，也可以将该项目从某个工作区复制到另一个工作区内。

复制项目：

在 Workspace Explorer 中，右键点击所需项目，然后选择 **Copy**。

粘贴项目：

在 Workspace Explorer 中，右键点击某个项目或工作区，然后选择 **Paste** 项。

复制项目将被添加到选定的工作区内，并且其名称后缀为“_Copy_01”。

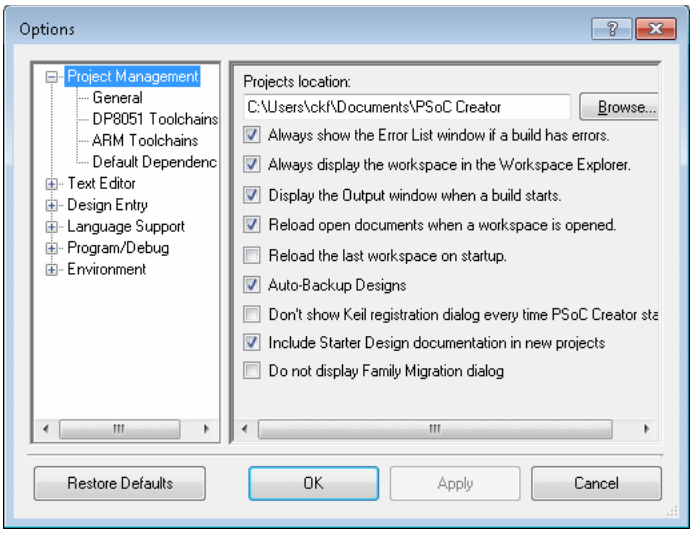
另外，请参考：

- 存档项目
- 将项目存储为

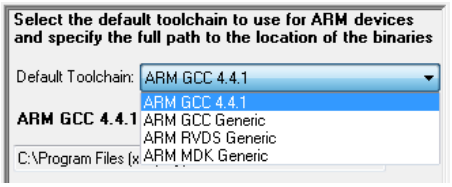
选择默认的编译器

对于所有目标器件，PSoC Creator 支持多个工具链。当某个系列优先使用了某项默认设置时，根据具体项目并通过使用 Build Settings（编译设置）下的选项，您可以更改所使用的工具。但如果您安装了外部编译器，那么可以通过 Project Management Options（项目管理选项）为任意新建项目指定所需的默认设置。选择默认的编译器：

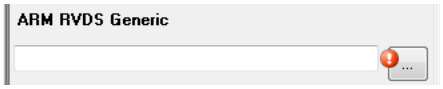
1. 选择 **Tools** 菜单中的 **Options** 项。



2. 扩展 Project Management（项目管理），然后选择 DP8051 工具链或 ARM 工具链。
3. 选择 **Default Toolchain**（默认工具链）下拉菜单，然后选择相应的选项。



将显示针对所选工具链的一个警告指示符。



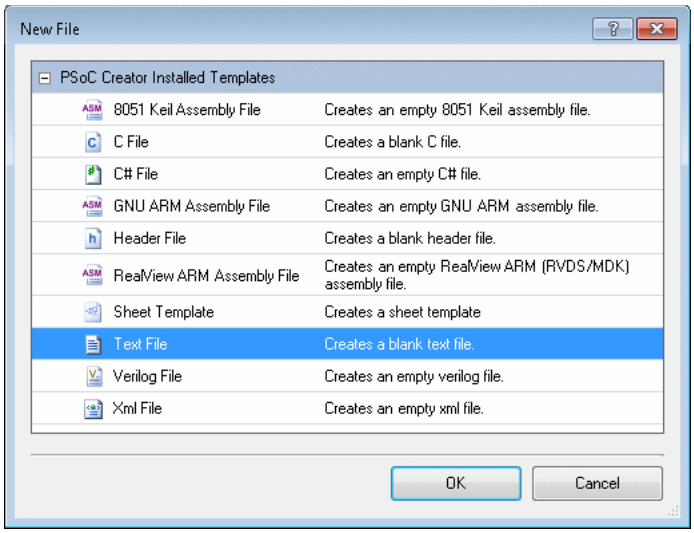
4. 点击 **Browse [...]**按键，然后导航到二进制文件所在的位置，选择该文件，然后点击 **OK**。
5. 点击 **OK**，关闭 Options 对话框。

另外，请参考：

- Options 对话框
- 编译设置

创建一个新文件

在 PSoC Creator 中，可通过 New File 对话框创建一个新文件。该对话框提供了您想要添加的各个文件类型的模板。您可以根据不同目的（如用于复制并粘贴代码的文件）创建一个文件。



创建一个新文件与将某个文件添加到某个项目内或添加某个组件**不同**。该过程仅针对指定类型创建一个空白文件。如要将一个新文件添加到您的项目内，请查看添加新的工作区/项目部分讲述的内容。要将某个组件添加到您的项目内，请参阅添加组件

打开该对话框：

从 **File** 菜单中依次选择 **New > File...** 。

创建新文件：

选择要创建的文件类型的图标，然后点击 **OK**。

新创建的空白文件将以选项卡式文档形式在 Text Editor 中打开，并且该文件使用的是它的默认名称。

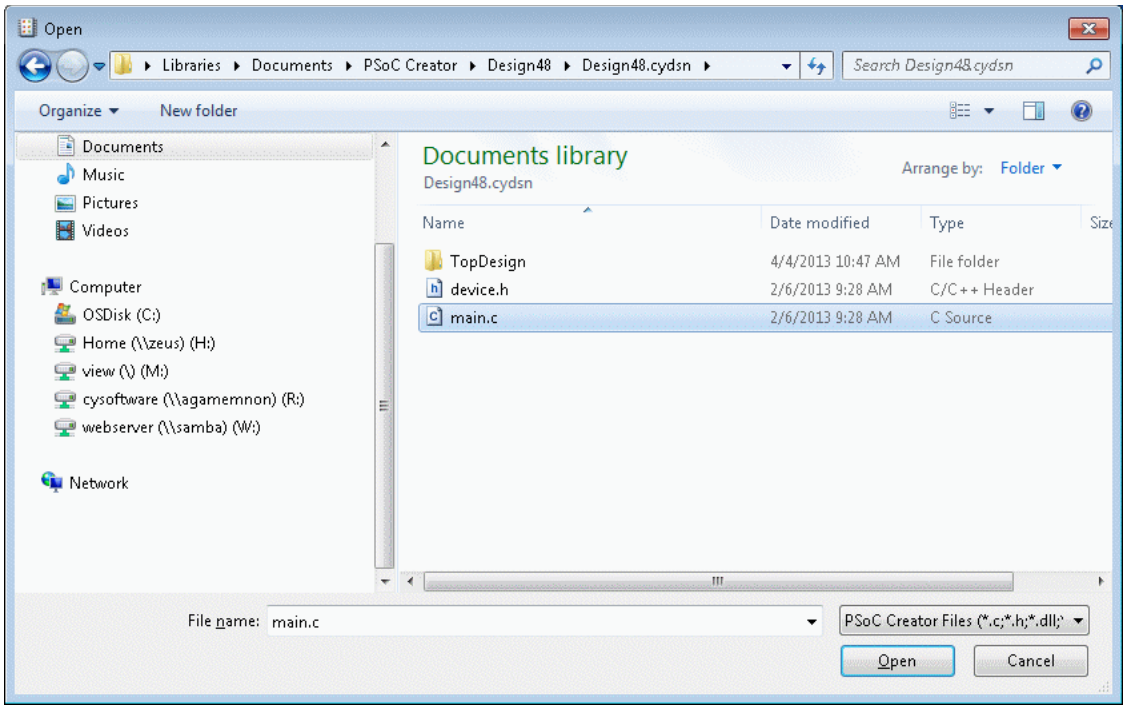
注意：在您指定该文件名称及其存储位置前，请不要将其存储到硬盘上。

另外，请参考：

- ‘文本编辑器’
- 工作区/项目

打开现有文件

在 PSoC Creator 中，通过 Open File 对话框，可以打开某个现有的文件。您可以根据不同目的打开某个文件，如用于复制和粘贴代码的文件。



使用该对话框浏览并选择要打开的文件，如源代码和文本文件。通过使用该对话框，您仅能打开 PSoC Creator 中的文件。该文件并不属于您先前打开的任何项目。如果要将某个文件添加到您的项目内，请查看添加新的工作区/项目部分讲述的内容。

打开该对话框：

从 **File** 菜单中依次选择 **Open > File...** 。

打开文件：

1. 导航到需要打开的文件所在的目录。
2. 选择所需文件，并点击 **Open**。

该文件将以选项卡式文档的形式打开在 Text Editor 中。

另外，请参考：

- ‘文本编辑器’
- 工作区/项目
- 创建一个新文件
- 打开某个现有的项目

创建文件夹

在 PSoC Creator 中，存在两种不同的文件夹：物理的文件夹和虚拟的文件夹。物理文件夹位于硬盘上。在 PSoC Creator 创建项目、组件以及器件/系列/架构的同时，也会创建物理文件。虚拟文件夹并不位于硬盘上。您可以在工作区和项目中创建虚拟文件夹，以根据要求组织各个项目，但您不能通过 Windows Explorer 查看这些文件夹。

滤波器：

每个文件夹均有一组相应的滤波器。当您将某个新的文件添加到包含了一个或多个文件夹的项目内时，该文件将被添加到带有与其扩展名相匹配的滤波器的文件夹内。您只有将某个文件直接添加到某个项目时，才会发生上述情况。

创建物理文件夹：

通过使用 PSoC Creator 接口，可以创建一个新项目、添加组件或在编译过程中生成文件。PSoC Creator 会为您创建这些文件夹。

要创建虚拟文件夹，请进行下述操作：

1. 右键单击 **Source** 选项卡下 Workspace Explorer 中的某个工作区、项目或文件夹，然后依次选择 **Add > New Folder** 或 **Add > New Workspace Folder**。

   带默认名称的新文件夹将被添加到选定的项目中。

2. 键入该文件夹的名称，然后敲击**[Enter]**键。

另外，请参考：

- 工作区浏览器
- 创建新项目
- 创建一个新符号
- PSoC Creator 项目编译

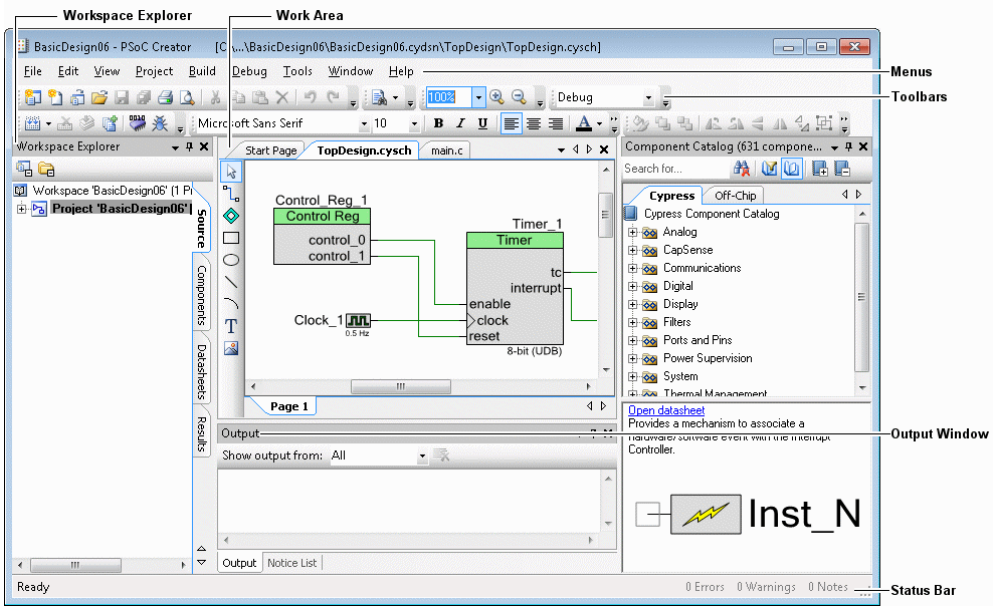
PSoC Creator 框架

本节包括以下各主题：

- 框架说明
- 窗口类型
- 框架接口组件
- 自定义框架

框架说明

PSoC Creator 框架所提供的丰富性能可帮助您组织您的设计并缩短项目的完成时间。



首次打开 PSoC Creator 时，框架将显示 Workspace Explorer（工作区浏览器）、文档工作区域和 Output 窗口。该框架还包含了一个菜单和一个状态栏，以及各种工具栏（根据您正在操作的文件类型，显示的工具栏会存在差异）。

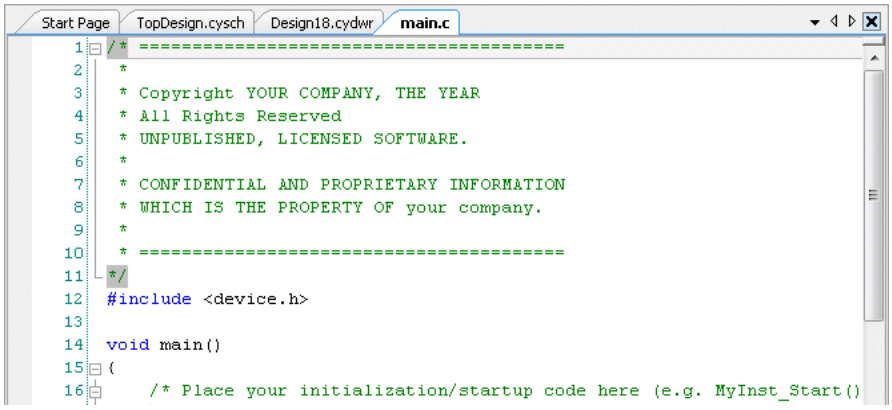
Workspace Explorer（工作区浏览器）

Workspace Explorer 是一种停靠工具窗口，它以 Windows Explorer 的方式显示您的项目文件。如果您在 Workspace Explorer 中双击某个文件，则该文件将显示在工作区内。更多信息，请参考 Workspace Explorer 部分。

文档工作区域：

文档工作区域包含了您在项目中打开的各个文件。根据所显示的文件类型，可用的工具和指令会存在差异。例如，如果您打开了源代码文件，您将看到用于编辑源代码的工具栏指令；如果您打开某个原理图文件（如用于将某个项目添加到您的原理图内的组件目录），则会显示用于编辑原理图的指令。

在工作区域内，所有文件都以选项卡式文档的形式显示。其中，有效文档显示在上面，其他文档位于下面。更多信息，请查看文档窗口。



起始页

起始页是一个通用的选项卡式文档窗口，它并不属于任何项目。它提供用于创建新项目、打开现有项目的链接，同时也提供有关 PSoC 器件的信息和新闻的链接。该页面显示在同一个工作区域内。但其他文档窗口需要在您的设计中打开。

通过配置，您可以选择是否在启动时显示起始页。请查看环境选项部分讲述的内容。

输出窗口：

Output 窗口是指显示各种系统信息的停靠工具窗口。更多信息，请查看输出窗口。

另外，请参考：

- 窗口类型
- 文档窗口
- 工具窗口
- 框架接口组件

窗口类型

窗口类型

PSoC Creator 提供了两种窗口：工具窗口和文档窗口：这两种窗口的运行方式稍微不同。

工具窗口：

可在 **View** 菜单上找到 Tool 窗口。该窗口由当前应用及其附加项定义。该窗口包括 Workspace Explorer、Output Window 和 Component Catalog。您可以配置工具窗口，以进行下面操作：

- 自动显示或隐藏
- 以选项卡形式与其他工具窗口链接
- 停靠在框架的边缘上
- 处于浮动状态

更多信息，请参见工具窗口部分。

文档窗口：

当您打开/创建文件或其他项目时，将随之动态创建文档窗口。可以在 **Window** 菜单上查找打开的文档窗口列表，先列出顶层窗口。这些窗口可能包括源代码文件、原理图文件、符号文件以及设计范围资源文件。更多信息，请查看文档窗口部分。

排列窗口：

通过排列各个窗口，您可以扩大用于查看和编辑代码的空间。您可以按照下面几种方式排列窗口：

- 选项卡式停靠文档窗口

- 停靠工具窗口到框架的边缘
- 沿着框架边缘最小化工具窗口
- 平铺文档窗口

更多信息，请参阅自定义框架部分。

另外，请参考：

- 工具窗口
- 文档窗口

工具窗口

可以在 **View** 菜单中查找工作窗口，其中包括：

- 工作区浏览器
- 输出窗口
- 组件目录

在 PSoC Creator 中，这些窗口会提供不同的功能，但能够在框架中以相同的方式排列它们。

工具窗口工具栏：

每个工具窗口均包含下面各条工具栏指令：

- **Window Mode**（窗口模式）— 通过该下拉菜单，您可以在不同模式中切换该窗口：
- **Floating**（浮动）— 将工具窗口设置为一个浮动的窗口，而不与 PSoC Creator 框架相连。
- **Dockable**（可停靠）— 将工具窗口设置为 PSoC Creator 框架的可停靠窗口。
- **Tabbed Document**（选项卡式文档）— 将工具窗口设置为选项卡式文档窗口。
- **Auto-Hide**（自动隐藏）— 如果未使用工具窗口，可将工具窗口移动到框架边缘。
- **Hide**（隐藏）— 关闭工具窗口。

  当右键点击任意工具窗口的右键菜单时，均能找到上述指令。
- **Auto-Hide Pushpin**（自动隐藏图钉）— 该指令可打开/关闭‘自动隐藏’功能。
- **Close**（关闭）— 该指令可关闭工具窗口。

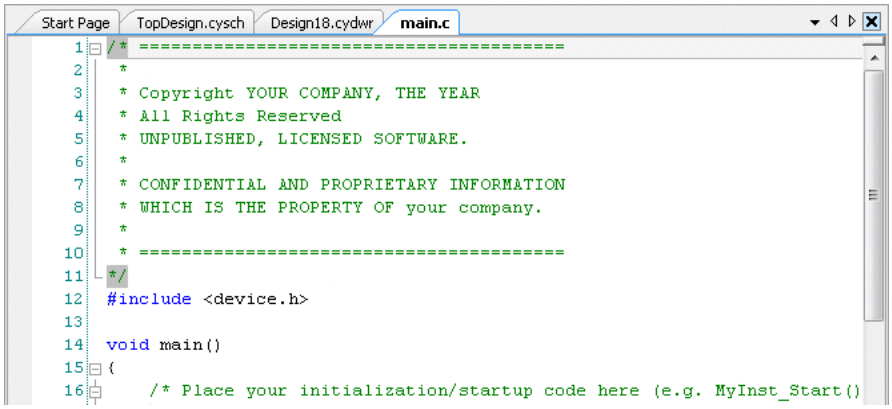
另外，请参考：

- 自动隐藏工具窗口
- 移动工具窗口

- 如何使工具窗口停靠
- 使工具窗口浮动

文档窗口

当您打开/创建文件或其他项目时，将随之动态创建文档窗口。在工作区域内，所有文件都以选项卡式文档的形式显示。其中，有效文档显示在上面，其他文档位于下面。



下面各节介绍了显示和运行工作窗口的各个方面。

文档工作区域工具栏：

文档工作区域左上方的工具栏包括下面各条指令：

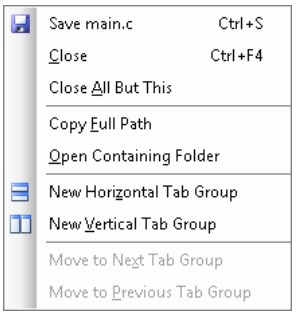
- **Select Document**（选择文档）—根据选项卡的顺序，该下拉菜单显示了工作区域内打开文档窗口的列表。您可以使用该菜单选择想要显示的另一个打开的文档。
- **Scroll Left/Right**（向左/向右滚动）— 如果在可见的工作区中存在几个需要向左/向右显示的打开文档，那么可以使用向左/向右滚动箭头实现该操作。
- Close（关闭）— **该指令可关闭有效文档**

文档选项卡右键菜单：

您可以右键点击某个选项卡，以显示右键菜单。存在两种右键菜单：一种用于项目文件，另一种用于非项目文件。

项目文件右键菜单

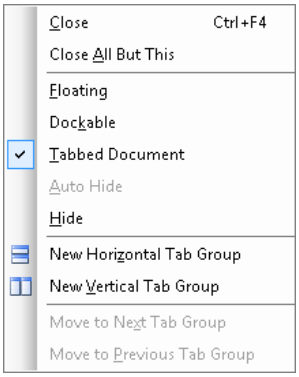
在文档工作区中，项目文件（如源代码文件或原理图文件）只能以选项卡式文档的形式显示。当您右键点击某个项目文件选项卡时，根据所选项目文件的上下文，会显示下面各条指令：



- **Save**（保存）— 保存所选文件。
- **Close**（关闭）— 关闭所选文件。
- **Close All But This**（关闭除该文件外的所有文件）— 关闭所有打开的选项卡式文档，但所选文件除外。
- **Copy Full Path**（复制整个路径）— 将文件的整个路径复制到剪贴板内，用以将其粘贴到一个文本文件中。
- **Open Containing Folder**（打开包含文件）— 打开该文件所在的文件夹。
- **New Horizontal Tab Group**（新的水平选项卡组）— 创建一组新的水平选项卡，并将所选文件移到该组内。
- **New Vertical Tab Group**（新的垂直选项卡组）— 创建一组新的垂直选项卡，并将所选文件移到该组内。
- **Move to Next Tab Group**（移到新一组选项卡）— 将所选文件移到新一组选项卡。
- **Move to Previous Tab Group**（移到前一组选项卡）— 将所选文件移到前一组选项卡内。

非项目文件右键菜单

可将非项目文件（如起始页）显示为选项卡式文档或浮动/停靠工具窗口。这些文件类型的右键菜单包括下面各条指令：



- **Close**（关闭）— 关闭所选文件。
- **Close All But This**（关闭除所选文件外的所有文件）— 关闭所有打开的选项卡式文档，但所选文件除外。
- **Floating/Dockable/Tabbed Document**（浮动/可停靠/选项卡式文档）— 可将所选文件显示为浮动窗口、可停靠窗口或选项卡式文档。
- **Auto-Hide**（自动隐藏）— 仅适用于可停靠窗口。通过该指令，可将窗口缩放到框架边缘。
- **Hide**（隐藏）— 关闭所选的文件/窗口。
- **New Horizontal Tab Group**（新的水平选项卡组）— 创建一组新的水平选项卡，并将所选文件移到该组内。
- **New Vertical Tab Group**（新的垂直选项卡组）— 创建一组新的垂直选项卡，并将所选文件移到该组内。
- **Move to Next Tab Group**（移到新一组选项卡）— 将所选文件移到新一组选项卡。
- **Move to Previous Tab Group**（移到前一组选项卡）— 将所选文件移到前一组选项卡内。

选择文档窗口：

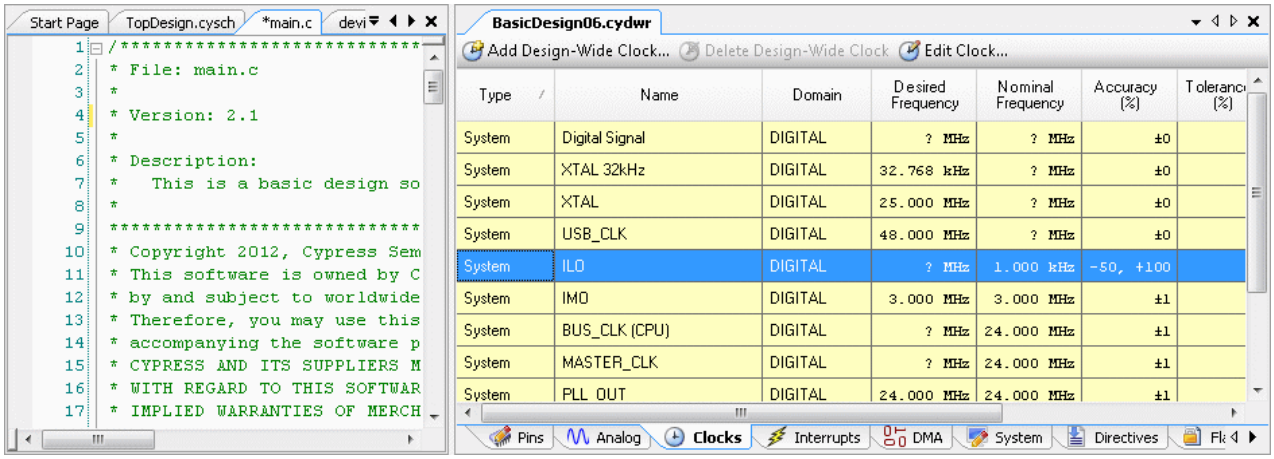
当存在打开的选项卡式文档时，文档窗口将并行排列在选项卡式面板上。这样，您可以轻松实现轮流使用您正在编辑的文档。

根据使用顺序，可以在各个打开的文档中进行切换：

- 按下组合键**[Ctrl] + [Tab]**，将按照最近触摸的顺序依次激活打开的文档。
- 按下组合键**[Ctrl] + [Shift] + [Tab]**，将以相反的顺序激活打开的各个文档。

选项卡组：

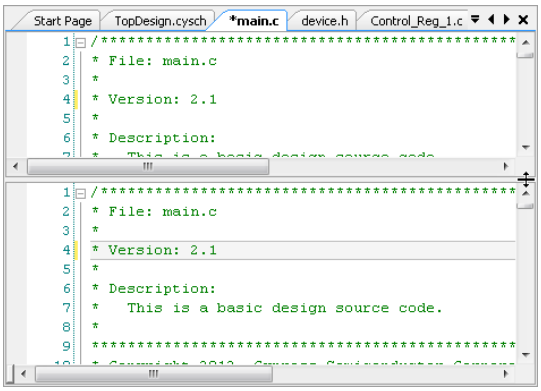
当对两个或更多个打开的文档进行操作时，可以通过选项卡组管理受限制的工作区。你可以将多个文档窗口组织成垂直或水平的选项卡组，并轻松地将某些文档从某个选项卡组中随机切换到另一个选项卡组。



分割窗口：

对于某些文件类型（如源代码文件），您可以同时查看并编辑同一个文件的两个不同位置。

- 如要将某个文档分成两个独立的滚动部分，请选择位于滚动栏上方的 **Split** 图标，并将其拖放到所需位置。



- 如果要取消分割功能，那么请将该图标托放回滚动栏上的位置。

另外，请参考：

- 工具窗口

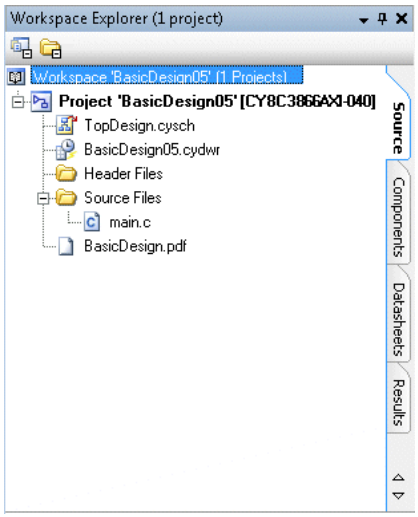
框架接口组件

主要的 PSoC Creator 架构组件包括：

- 工作区浏览器
- 输出窗口
- 注释列表窗口
- 文件菜单
- Edit（编辑）菜单
- 视图菜单
- 调试器菜单
- 项目菜单
- 编译菜单
- 工具菜单
- 帮助菜单
- 标准工具栏
- 键盘快捷方式

工作区浏览器

Workspace Explorer 是 PSoC Creator 进行显示某个工作区内容的方法。



与 Windows Explorer 相同，内容也以树形目录的形式显示。此外，工具窗口边缘有不同的选项卡，通过这些选项卡您可以根据目录查看不同的文件类型。工具栏指令：

Workspace Explorer 包含两条指令：

- **Collapse to Project**（折叠项目）— 将选定的项目树形目录折叠成顶级项目。

- **Collapse to Parent**（折叠父目录）— 将选定的树形目录折叠成顶级节点。

打开文件：

双击 Workspace Explorer 中的某个文件，以便在文档工作区中以选项卡式文档的形式打开该文件。

源选项卡：

如果未选定任何选项卡，将显示 **Source** 选项卡。PSoC Creator 仅显示了您所添加的文件或 PSoC Creator 编译过程中生成代码时所生成的文件。每当编译完成或侧重于 PSoC Creator 时，该视图将被更新。

在 **Source** 选项卡中，您只能向项目内添加新的文件夹或文件，或创建新的项目。却不能对组件进行任何操作。要想将新项目添加到组件内，或添加整个新的组件，必须使用 **Components** 选项卡。

生成的源文件

在某个设计项目中，Generated Source 目录包含了来自代码生成步骤的源。Generated Source 目录下方显示的是先前构建的每一个架构的子目录。每个架构目录被进一步分为每个组件实例（提供一个 API）的虚拟文件夹。除了实例目录外，架构目录还包含了其他生成文件（如 *project.h*、启动文件，等等），其中，每个项目作为树形目录的一个节点。工作区中的每一个项目将显示其所有的内容。请参阅生成文件部分。

组件选项卡：

Components 选项卡筛选工作区中的内容，从而使它仅显示工作区中属于每个项目的组件，以及所有组件的文件和文件夹中的内容。在该选项卡中，您可以添加新的组件，或将新项目添加到某个组件中，当然也可以添加新的项目。您可能在组件范围外添加文件或文件夹，但在这种情况下，需要使用 **Source** 选项卡。更多有关组件的信息，请参阅组件作者指南。

数据手册选项卡：

通过 **Datasheets** 选项卡，您可以快速访问设计中当前使用的器件和组件数据手册。当您添加或移除某个组件，或更改某个器件时，现有的文档也相应被更新。

当您双击某个文件时，它会在默认的 PDF 阅读器工具中打开。

结果选项卡：

Results 选项卡显示了最近构建的每个平台所需要的文件动态列表。某个平台将选择要显示的构建文件子集。根据具体的平台，所显示的内容可能不一样。

如果未构建某个平台或配置，或者构建过程中删除了所创建的文件夹，那么将不再显示该平台的文件。此外，如果找到某个平台或配置目录，但所需的文件被丢失，那么，该文件将显示为灰色，这样便表示在硬盘上未找到该文件。

每个设计构建需要显示的最小文件集为：

- 编程文件
- 调试文件（若与编程文件相异）
- 代码生成报告文件（若运行代码生成）
- 器件文件（当被执行时）

此外，可能包括额外的文件，如列表文件和映射文件。

对于库构建，结果库文件将被显示。

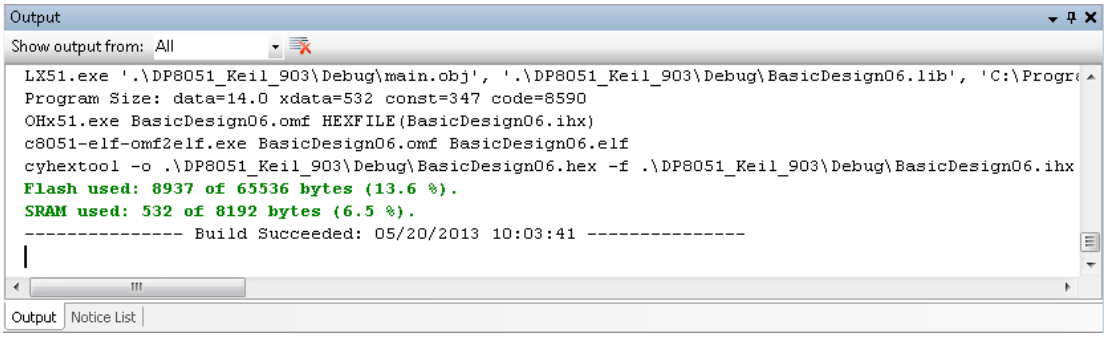
通过双击 **Results** 选项卡中的文件，可以打开该文件的相应编辑器，如报告文件的文本编辑器。根据需要，可以进行其他操作，如编程某个器件或进行调试。不能以任何方式修改 **Results** 选项卡中显示的项目。该选项卡仅用来显示构建过程得到的结果。

另外，请参考：

- 工作区/项目
- 框架说明
- 编译 PSoC Creator 项目
- 生成的文件

输出窗口

Output 窗口显示了 PSoC Creator 中各种功能的状态信息，包括构建器和调试器。该窗口通常位于 PSoC Creator 框架的底部。该窗口一般与注释列表窗口属于同一组窗口。



注意：当构建某个 Bootloader 项目时，闪存大小表示的是存储在闪存中的字节数量（与用于普通项目的闪存的工作方式相同）。当构建某个 Bootloadable 项目时，Bootloader 大小表示的是该 Bootloader 所占用的闪存行大小（它是该 Bootloader 被填充到行大小的整数倍）。

显示 Output 窗口：

当您启动 PSoC Creator 时，通常默认显示 Output 窗口。如果该窗口被关闭，则通过点击 **View** 菜单上的 **Output Window** 项，您可以重新打开该窗口。

工具栏：

根据您当前使用的工具，Output 窗口可能显示不同的工具栏指令。最常用的指令包括：

显示输出自

显示一个或多个输出窗格。根据使用 Output 窗口发送信息的具体工具，可用的信息窗格可能不一样。

全部清除

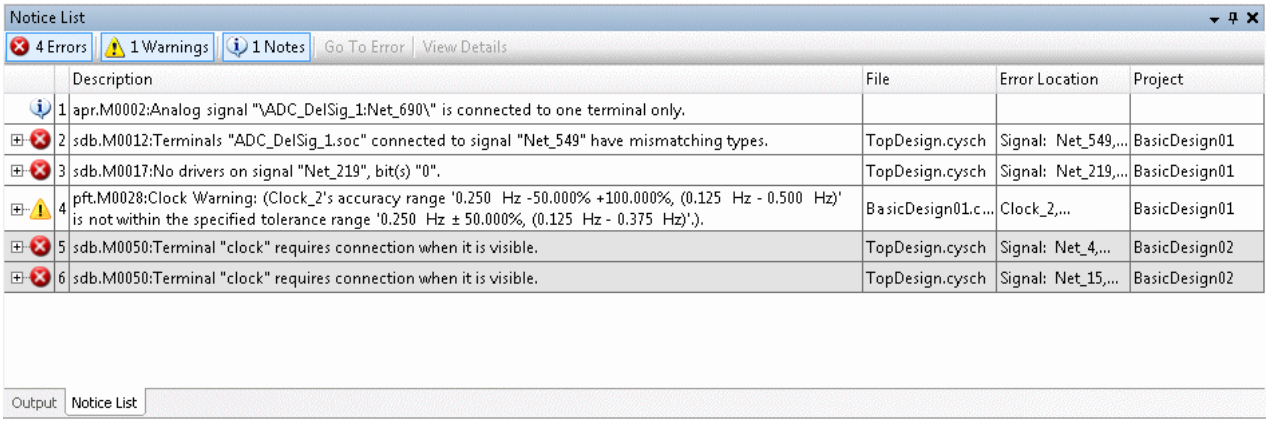
清除 Output 窗格中的所有文本。

另外，请参考：

- 框架说明
- 注释列表窗口

注释列表窗口

Notice List 窗口将分散在不同地方的注释（如错误、警告和注解）聚集到一个列表中。如果显示了某个文件和/或错误位置，您可以通过双击该条目来显示错误或警报。此外，您还可以使用 **Go To Error**（查看错误）或 **View Details**（查看详情）等按键。该窗口通常位于 PSoC Creator 框架的底部。该窗口一般与输出窗口属于同一组窗口。



错误、警报和注解等信息也被显示在 PSoC Creator 状态栏上。

- **错误信息**指示：构建项目前，至少有一个必须解决的问题。典型的错误可能包括：编译器编译错误、原理图中的动态连接错误以及设计原则检查(DRC)错误。构建过程中产生的错误一直被列出，直到进行下一个构建为止。
- **警报信息**指示：可能表示某个问题的异常情况，但一般在构建应用时可以忽略该信息。
- **注意**：信息指示：已发生状况的系统信息。

Notice List 窗口包括下面各列：

- **Icon**（图标）—显示错误、警报或注解的图标。 特定行可能包括了具有整个信息独立部分的树形控制结构。
- **Number**（数量） — 显示注释的数量。
- **Description**（说明）—显示注释的简要说明。
- **File**（文件）—显示注释所在的文件名称。
- **Error Location**（错误位置）— 如果可行，将显示信息的具体行号或其他位置。

- **Project**（项目） — 显示包含注释的项目名称。白色行为活动的项目；灰色行则是非活动的项目。

设计规则检查（DRC）：

DRC 根据项目数据库中预定义的规则集合评估您的设计。DRC 指出了您的项目中可能引起问题的潜在错误或违反“规则”的行为。该工具在 Notice List 窗口中显示各项信息。

当您更改自己的设计时，某些连接和动态错误将立即被更改，但其他错误要通过加载和保存操作才能更新。

打开 Notice List 窗口：

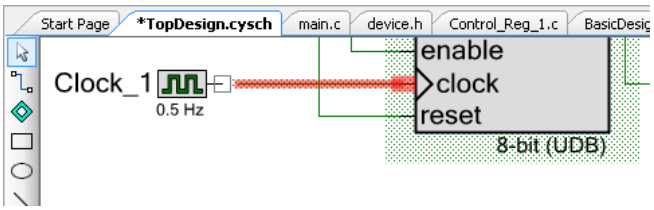
默认显示 Notice List 窗口。如果您关闭了该窗口，可通过依次选择 **View > Other Windows > Notice List** 重新打开它。

另外，通过依次选择 **Windows > Reset Layout**，您也可以再次打开该窗口。但该选项会将所有窗口恢复到其默认的位置。

在各工具中显示错误：

点击列表上的某个注释。如果存在错误的链接，请点击 **Go To Error** 按键。您也可以通过双击 Notice List 窗口中的任意警告或错误信息，以在合适的工具中显示该信息。如果您双击了该注释，那么可以通过使用一个指令导航到该错误。

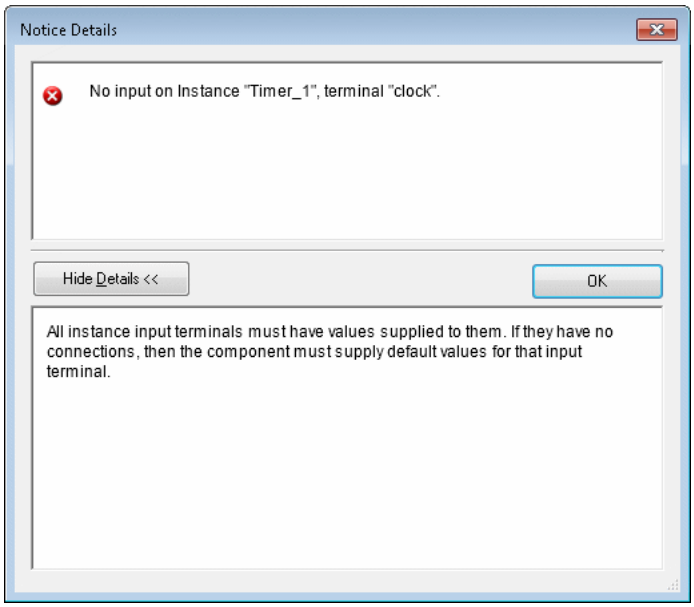
- 在原理图中，该错误的源将以半透明的红色高亮显示。通过在注释上右键点击菜单，您可以打开或关闭高亮显示。



- 在代码文件内，带错误或警告信息的特定代码行将被高亮显示。

查看信息说明：

点击信息旁边的按键，或点击 **View Details** 按键以打开 Notice Details 对话框。如果您双击了该注释，您可以通过一条指令查看详细信息。Notice Details 对话框将显示整条信息，以及额外信息（如果存在）。



另外，请参考：

- 框架说明
- 输出窗口
- 注释列表错误信息
- 公告详情

文件菜单

File 菜单包括下面指令：

菜单项目		图标	快捷方式	说明	另外，请查看
New >	Project			显示一个对话框，用于创建新的项目。	创建新项目
	文件			显示一个对话框，用于创建新的文件。	创建文件
Open（打开） >	Project/Workspace（项目/工作区）			显示个对话框，用于打开现有的项目/工作区。	打开现有的项目
	File（文件）		[Ctrl] + [O]	将显示一个对话框，通过该对话框可以打开现有的文件。	打开某个现有的文件
Example Project（示例项目）				显示一个对话框，用于打开示例项目。	查找示例项目
Add（添加） >	New Project（新项目）			向工作区添加新的项目。	
	Existing Project（现有项目）			向工作区添加现有的项目。	
Close（关闭）			[Ctrl] + [F4]	关闭活动文档窗口。	窗口类型
Close Workspace（关闭工作区）				关闭工作区。	

菜单项目	图标	快捷方式	说明	另外，请查看
Save（保存）		[Ctrl] + [S]	保存活动文档。	
Save <Project / File> As（另存 <项目/文件>）			将显示一个对话框，通过该对话框可以将所选的项目或文件保存为另一个项目或文件。	另存项目
Save All（保存所有）		[Ctrl]+ [shift] + [S]	保存工作区的所有文件。	
Create Workspace Bundle（创建工作区组）			打开工作区/项目存档向导，创建工作区组。	存档工作区/项目
Page Setup（页面设置）			打开 Page Setup 对话框，以选择打印布局选项。	
Print（打印）		[Ctrl] + [P]	打印活动文档。	
Print Preview（打印预览）			打开所选文件的 Print Preview（打印预览）对话框。	Print Preview（打印预览）
Recent Projects（最近的项目）			访问之前打开的项目。	
Recent Files（最近的文件）			访问之前打开的文件。	
Exit（退出）			退出 PSoC Creator。	

Edit（编辑）菜单

Edit 菜单包括下面指令：

菜单项	图标	快捷方式	说明	参考
Undo（撤销操作）		[Ctrl] + [Z]	撤消最后一步操作；可以反复使用该操作，以撤销多步操作。	
Redo（恢复操作）		[Ctrl] + [Y]	恢复最后一个操作。	
Cut（剪切）		[Ctrl] + [X]	剪切选定的元素。	
Copy（复制）		[Ctrl] + [C]	复制选定的元素。	
Paste（粘贴）		[Ctrl] + [V]	从剪贴板粘贴剪切或复制的元素。	
Delete（删除）		[Delete]	删除选定的元素。	
Select All（选择所有）		[Ctrl] + [A]	选择活动窗口中的所有元素。	
Find and Replace（查找与替换） >				
Find（查找）		[Ctrl] + [F]	打开一个对话框，查找所需文本。	查找与替换
Replace（替换）		[Crtl] + [H]	打开一个对话框，查找并替换所需文本。	查找与替换
Batch Find（批量查找）		[Ctrl] + [Shift] + [F]	打开一个对话框，在多个文件中查找文本。	批查找
Batch Replace（批量替换）		[Ctrl] + [Shift] + [H]	打开一个对话框，在多个文件中查找并替换文本。	批替换
Go To（跳转到）		[Ctrl] + [G]	打开 Go To（跳转到）对话框。	跳转到行编号

菜单项	图标	快捷方式	说明	参考
Advanced（高级选项） >				文本编辑器
Tabify Selected Text（跳格选定文本）			将选定文本的空格转换为跳格符。在文本编辑器选项对话框中定义了每个跳格符的空格数量。	
UnTabify Selected Text（取消选定文本的跳隔符）			将选定文本的跳格符转换为空格。	
Make Uppercase（转为大写字母）		[Ctrl] + [Shift] + [U]	将选定文本修改为大写字体。	
Make Lowercase（转为小写字母）		[Ctrl] + [U]	将选定的文本修改为小写字母。	
View White Space（查看空白符）		[Ctrl] + [E], [S]	启用或关闭显示隐藏字符的性能，如空格或标签。按住**[Ctrl]**和**[E]**键，然后释放，这时再按下**[S]**键。	
Incremental Search（增量搜索）		[Ctrl] + [I]	按快捷键后，查找第一个匹配实例。按下**[Ctrl]** + **[I]**组合键，将显示图标。然后输入要查找的字母。	
Comment Selection（注释选择）		[Ctrl] + [E], [C]	注释选定的文本行。在行的开头插入//。按住**[Ctrl]**和**[E]**键，然后释放，这时再按下**[C]**键。	
Uncomment Selection（取消注释选择）		[Ctrl] + [E], [U]	取消选定文本行的注释。移除行开头处的//。按住**[Ctrl]**和**[E]**键，然后释放，这时再按下**[U]**键。	
Increase Line Indent（增加行间距）			缩进选定的文本行。	
Decrease Line Indent（减少行间距）			伸出选定的文本行。	
Toggle Bookmark（切换书签）		[Ctrl] + [K], [Ctrl] + [K]	在光标所在行旁边的指示器边距，插入或删除书签。按住**[Ctrl]**和**[K]**键，然后释放，这时再按下**[Ctrl]**和**[K]**键。	
Outlining（轮廓） >				文本编辑器
Toggle Outlining Expansion（切换轮廓的收缩/展开）		[Ctrl] + [M], [M]	如果启用了 Start Automatic Outlining（启动自动轮廓），则该指令将展开或收缩选定代码段的轮廓。按住**[Ctrl]**和**[M]**键，然后释放，这时再按下**[M]**键。	
Toggle All Outlining（切换所有轮廓的收缩/展开状态）		[Ctrl] + [M], [L]	如果启用了 Start Automatic Outlining（启动自动轮廓），该指令将扩展或收缩文件中的所有轮廓。按住**[Ctrl]**和**[M]**键，然后释放，这时再按下**[L]**键。	
Stop Automatic Outlining（停止自动轮廓）		[Ctrl] + [M], [P]	禁用自动轮廓。按住**[Ctrl]**和**[M]**键，然后释放，这时再按下**[P]**键。	
Start Automatic Outlining（启动自动轮廓）			使能自动轮廓。	

View（视图）菜单

View 菜单包括下面指令：

菜单项	图标	快捷方式	说明	参考
Output Window（输出窗口）			打开或显示 Output 窗口。	输出窗口
Workspace Explorer（工作区浏览器）			打开或显示 Workspace Explorer 窗口。	工作区浏览器
Code Explorer（代码浏览器）			打开代码编辑器的 Code Explorer 窗口。	代码浏览器窗口
Component Catalog（组件目录）			打开原理图编辑器的组件目录。	组件目录
Other Windows > Start Page（其它窗口 > 起始页面）			打开或显示 Start 页面。	框架说明
Other Windows > Notice List（其它窗口 > 注意列表）			打开或显示 Notice List（注意列表）窗口。	注意列表窗口
Find Results（查找结果）			打开或显示 Find Results（查找结果）窗口。	查找与替换
Zoom In/Out/To（放大/缩小/放大至所需指定）		[Ctrl] + [+]或 Ctrl + [–]	按照选项调整页面的尺寸	

Project（项目）菜单

Project 菜单包括下面指令：

菜单项	图标	说明	参考
New Item（新项）		打开 New Item（新项目）对话框，选择添加到项目/工作区的新项。	New Item（新项）
Add Component Item（添加组件项）		打开 Add Component Item（添加组件项）对话框，增加新的组件项。	添加组件项
Existing Item（现有的项）		打开 Open（打开）对话框，向您的项目/工作区内添加现有的项。	打开现有的文件
Import Component（导入组件）		打开 Import Component（导入组件）对话框。	导入组件
Update Components（更新组件）		打开 Update Component Tool（更新组件工具）对话框，更新选定的组件。	
Remove from Workspace（从工作区内移除）		从项目/工作区中移除选定的文件/项目。	
Unload/Reload Project（卸载/重加载项目）		交替进行卸载和重加载工作区中选定的项目。	
New Folder（新文件）		在工作区浏览器中创建虚拟文件夹，并非在磁盘上。	
Show All Files（显示所有文件）		打开目录中的所有文件。	
Set As Active Project（设置为活动项目）		将选定项目设置为活动项目。	

菜单项	图标	说明	参考
Set As Top Component（设置为顶级组件）		将选定组件设置为顶级组件。	
Dependencies（依赖属性）		打开 Dependencies 对话框。	Dependencies（依赖属性）
编译顺序		打开 Dependencies 对话框中的 Build Order 选项卡。	Dependencies（依赖属性）
Device Selector（器件选型器）		打开器件选型器。	器件选型器
Archive Workspace/Project（存档工作区/项目）		打开 Archive Project 对话框，用于存档工作区/项目。	存档工作区/项目
Export to IDE（移植到 IDE）		打开将 PSoC Creator 设计移植到外部 IDE 的对话框。	将设计移植到第三方 IDE 内
<Project> Resources（<项目>资源）		打开项目的 Design-Wide Resources 文件。	Design-Wide Resources
Build Settings（编译设置）		打开 Build Settings 对话框。	编译设置
Properties（属性）		打开 Properties 对话框。	属性

Build（编译）菜单

Build 菜单包括下面各条指令：

菜单项	图标	快捷方式	说明	另外，请参考
Build All Projects（编译所有项目）		[F6]	编译工作区内的所有项目。	PSoC Creator 项目编译
清理所有项目			清理工作区中的所有项目。 **注意：**在清理期过程中，PSoC Creator 仅会清理项目中各文件的编译输出。不属于项目的源文件（从项目中删除或移除）生成的输出文件不会被清理掉。	
Clean and Build All Projects（清理和编译所有项目）			清理和编译工作区中的所有项目。	
Build (Named) Project（编译（名称）项目）		[Shift] + [F6]	编译选定的项目。	
Clean (Named) Project（清理（名称）项目）			清理已选定的项目。	
Clean and Build (Named) Project（清理和编译（名称）项目）			重编译选定的项目。	
Cancel Build （取消编译）		[Ctrl] + [Break]	取消编译过程。	

菜单项	图标	快捷方式	说明	另外，请参考
Compile File（编译文件）		[Ctrl] + [F6]	编译选定的文件。	
Generate Application（生成应用）			生成 API 源代码文件。	Generated Files（生成的文件）
Generate Project Datasheet（生成项目的数据手册）			生成项目的数据手册。	生成项目的数据手册

另外，请参考：

- 编译工具栏指令

调试器菜单指令

通过 Debug（调试）菜单，可以访问当前调试器中所有可用功能和信息。调试菜单有两种模式：非活动模式和活动模式。

非活动模式的调试菜单：

调试器未运行时，调试菜单将处于非活动状态。在该状态中，只有必要的功能可用。例如，在非活动状态中，没必要使用暂停功能。在活动模式下，通过 Debug 窗口子菜单只能访问一定的可用调试窗口子集。

指令	图标	快捷方式	说明
Windows（窗口） >			用于访问各种调试器窗口。请参阅 Debugger 窗口。
Breakpoints（断点）		[Ctrl] + [D], [B]	打开 Breakpoints 窗口，显示设置在工作区中的所有断点。
Output（输出）		[Ctrl] + [D], [O]	打开 Output 窗口
Program（编写）		[Ctrl] + [F5]	单击该图标，将由选定项目生成的代码编写到选定的调试目标内。 如果该项目已经过期，将自动启动编译过程 更新状态栏的信息，以显示正在进行编程操作。 后台启动 PSoC Programmer，以进行编程操作。
Select Debug Target（选择调试目标）			打开 Select Debug Target 对话框，手动选择需要使用的调试目标。
Debug（调试）		[F5]	启动调试器。
Debug without Programming（仅调试，不编程）		[Alt] + [F5]	启动调试器，不需要对器件进行编程。
Attach to Running Target（连接到运行目标）			打开 Attach to Target 对话框，连接到一个已经编程好的目标器件。仅适用于 PSoC 3 和 PSoC 5LP。
Toggle Breakpoint（切换断点）		[F9]	交替执行插入和移除当前代码行的断点。该项的作用与点击代码编辑器中的 **Indicator Margin**（指示器边距）的作用相同。该指令的快捷键为[**F9**]。
New Breakpoint（新断点） >			访问下面的断点窗口。请参阅断点窗口中的内容。
Address Breakpoint（地址断点）		[Ctrl] + [D], [A]	打开 Address Breakpoint 窗口。

指令	图标	快捷方式	说明
File Line Breakpoint（文件行断点）		[Ctrl] + [D], [F]	打开 File/Line Breakpoint 窗口。
Function Breakpoint（函数断点）		[Ctrl] + [D], [U]	打开 Function Breakpoint 窗口。
Variable Watchpoint（变量观察点）		[Ctrl] + [D], [V]	打开 Variable Watchpoints 窗口。
Memory Watchpoint（存储器观察点）		[Ctrl] + [D], [E]	打开 Memory Watchpoint 窗口。
Delete All Breakpoints（删除所有断点）		[Ctrl] + [Shift] + [F9]	删除工作区内的所有断点，并非逐个移除每一个断点。如果您设置了多个断点，但您想处理器保持运行，那么该指令便非常有用。
Enable All Breakpoints（启用所有断点）			启用所有断点

活动调试窗口的菜单：

活动调试模式表示您已经启动了调试会话；有效的 PSoC Creator 子系统便是调试器。在活动模式下，调试器正在运行，您可以调用所有可用的功能；通过调试窗口中的子菜单，可以访问所有可用的调试窗口。

指令	图标	快捷方式	说明
Windows（窗口） >			用于访问各种调试器窗口。请参阅 Debugger 窗口。
Breakpoints（断点）		[Ctrl] + [D], [B]	打开 Breakpoints 窗口，显示设置在工作区中的所有断点。
Output（输出）		[Ctrl] + [D], [O]	打开 Output 窗口
Watch（观察） >			
1		[Ctrl] + [D], [W]	打开四个 Watch 窗口中的一个，进行评估和显示变量、寄存器或表达式。
2		[Ctrl] + [Alt] + [W], [2]	
3		[Ctrl] + [Alt] + [W], [3]	
4		[Ctrl] + [Alt] + [W], [4]	
Locals（局部）		[Ctrl] + [D], [L]	打开 Locals 窗口，查看并修改当前调试框架中所有局部变量。
Components（组件）		[Ctrl] + [D], [P]	打开 Component Debug 窗口，查看设计中有关组件的调试信息。
Call Stack（调用堆栈）		[Ctrl] + [D], [C]	打开 Call Stack 窗口，跟踪目标程序调用函数的顺序。
Memory（存储器）>			
1		[Ctrl] + [D], [M]	打开四个 Memory 窗口中的一个，用于显示处理器内存中保存的值。
2		[Ctrl] + [Alt] + [M], [2]	
3		[Ctrl] + [Alt] + [M], [3]	
4		[Ctrl] + [Alt] + [M], [4]	

指令	图标	快捷方式	说明
Disassembly（反汇编）		[Ctrl] + [Alt] + [D]	打开 Disassembly 窗口，显示源代码中创建的基本指令。
Registers（寄存器）		[Ctrl] + [R], [R]	打开 Registers 窗口，显示内核 CPU 寄存器以及它们的值
Program（编写）		[Ctrl] + [F5]	单击该图标，将由选定项目生成的代码编写到选定的调试目标内。 如果该项目已经过期，将自动启动编译过程 更新状态栏的信息，以显示正在进行编程操作。 后台启动 PSoC Programmer，以进行编程操作。
Select Debug Target（选择调试目标）			打开 Select Debug Target 对话框，手动选择需要使用的调试目标。
Show Current Line（显示当前行）			显示正在执行或将要执行的代码行。
Resume Execution（恢复执行）		[F5]	进行执行调试器。暂停或执行断点后，将重启调试目标。使用该函数继续执行程序，直到下一个断点到来为止。
Halt Execution（暂停执行）		[Ctrl] + [Alt] + [Break]	暂停调试器。
Stop Debugging（停止调试）		[Shift] + [F5]	停止调试会话。
Rebuild and Run（重新编译及运行）		[Ctrl] + [Shift] + [F5]	重新编译程序，并重启调试器。
Reset（复位）		[Ctrl] + [Alt] + [F5]	复位调试器。
Step Into（单步执行）		[F11]	执行单一代码行。如果代码行是函数调用，则调试器将在函数的第一条指令处执行断点。如果代码行不是函数调用，调试器将在下一行执行断点。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。
Step Over（单步跳过）		[F10]	执行单一代码行。调试器将在下一个代码行上执行断点。如果当前的代码行是一个函数调用，那么不需要停止调试器也可以执行函数。函数调用后，调试器将停止在下一个行中。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。
Step Out（单步跳出）		[Shift] + [F11]	执行完当前函数。处理器将继续运行，直到完成当前的函数为止。调用函数后，它将暂停在第一条指令处。使用该函数退出当前的函数，并返回调用函数。该函数暂时允许处理器运行，直到处理完当前函数中的各条指令为止。
Toggle Breakpoint（切换断点）		[F9]	交替执行插入和移除当前代码行的断点。该项的作用与点击代码编辑器中的 **Indicator Margin**（指示器边距）的作用相同。该指令的快捷键为**[F9]**。
New Breakpoint（新断点） >			访问下面的断点窗口。请参阅断点窗口中的内容。
Address Breakpoint（地址断点）		[Ctrl] + [D], [A]	打开 Address Breakpoint 窗口。

指令	图标	快捷方式	说明
File Line Breakpoint（文件行断点）		[Ctrl] + [D], [F]	打开 File/Line Breakpoint 窗口。
Function Breakpoint（函数断点）		[Ctrl] + [D], [U]	打开 Function Breakpoint 窗口。
Variable Watchpoint（变量观察点）		[Ctrl] + [D], [V]	打开 Variable Watchpoints 窗口。
Memory Watchpoint（存储器观察点）		[Ctrl] + [D], [E]	打开 Memory Watchpoint 窗口。
Delete All Breakpoints（删除所有断点）		[Ctrl] + [Shift] + [F9]	删除工作区内的所有断点，并非逐个移除每一个断点。如果您设置了多个断点，但您想处理器保持运行，那么该指令便非常有用。
Enable All Breakpoints（使能所有断点）			使能所有断点。
Refresh（刷新）			刷新调试器。
Enable/Disable Global Interrupt（使能/禁用全局中断）			根据当前的状态使能或禁用全局中断

另外，请参考：

- 调试器的使用
- 调试器工具栏指令
- 文本编辑器的右键菜单指令

Tools（工具）菜单

Tools 菜单包括下面各条指令：

菜单项目	说明	参考
Install drivers for µVision（安装 µVision 的驱动程序）	打开 Installation 向导，安装 µVision IDE 的 MiniProg3 驱动程序。	安装 MiniProg3 驱动程序
Datapath Config Tool（Datapath 配置工具）	启动 Datapath 配置工具，进行创建通过 Verilog 执行的组件。	组件创建指南
DMA Wizard（DMA 向导）	打开 DMA 向导。	DMA 向导
Component Tuners（组件调谐器）	如果设计中包含组件调谐器，那么能够直接访问调谐器对话框。	调试器通信设置
Bootloader Host...（Bootloader 主机）	启动单独的 Bootloader 主机应用。	Bootloader 主机帮助文件（在应用中按**[F1]**键。）

菜单项目	说明	参考
Options...（选项）	打开 Options 对话框。	Options 对话框

Help（帮助）菜单

Help 菜单包括下面指令：

菜单项目	说明
Topics（主题）	打开 Help 中的 Welcome 主题。
Document Manager（文档管理器）	打开文档管理器，以访问与 PSoC Creator 相关的各个文档。
System Reference（系统参考指南）	系统参考指南 — 无论您的设计使用的是哪个 cy_boot 版本，都会打开最新版本的系统参考指南。 打开网页 — 打开访问旧版本和翻译后版本的系统参考指南页面。
Update Manager（更新管理器）	启动赛普拉斯的更新管理器，以检查并采用程序的更新内容。
PSoC Creator Training（PSoC Creator 培训）	打开赛普拉斯 PSoC Creator 的培训网页。
Cypress Dev Community（赛普拉斯开发者社区）	打开赛普拉斯开发者社区网页。
Documentation（文档） >	快速入门指南 — 打开 PSoC Creator 快速入门文档。 发布说明 — 打开发布说明文档。 已知问题及解决方案 — 打开赛普拉斯网页，以访问已知问题及解决方案文档。 移植指南 — 打开赛普拉斯网页，访问各种移植指南文档。 器件数据手册 — 打开赛普拉斯网页，访问 PSoC 器件数据手册。 PSoC 技术参考手册 — 打开赛普拉斯网页，访问 PSoC TRM 文档。
	Japanese Documentation（日文文档） > — 组件数据手册 — 打开访问各种翻译版本的组件数据手册的网页。
	Chinese Documentation（中文文档） > — 组件数据手册 — 打开访问各种翻译版本的组件数据手册的网页。
	组件创建指南 定制 API 参考指南 Warp Verilog 参考指南 调谐器 API 参考指南
	GCC & Keil 文档
Register（注册）	PSoC Creator Keil
Support（支持） >	知识库 — 打开赛普拉斯知识库网页。 创建支持案例 — 打开赛普拉斯支持网页，以创建一个新的支持案例。 查看您的支持案例 — 打开赛普拉斯支持网页的主页。

菜单项目	说明
Order Samples（订购样品）	打开赛普拉斯网上商店网页，以订购样品、套件、电缆等。
Contact Us（联系我们）	打开 Help 主题，查找赛普拉斯的联系方式。
About（关于 PSoC Creator）	打开关于 PSoC Creator 对话框，获取有关编译和插件的信息。

标准工具栏

PSoC Creator 标准工具栏包含以下指令：

- New Project/Workspace（新的项目/工作区）
- New File（新建文件）
- Open Project/Workspace（打开项目/工作区）
- Open File（打开文件）
- Save（保存）
- Save All（保存所有）
- Print（打印）
- Print Preview（打印预览）
- Cut（剪切）
- Copy（复制）
- Paste（粘贴）
- Delete（删除）
- Undo（撤销操作）
- Redo（恢复操作）

键盘快捷方式

下表列出了 PSoC Creator 中各种可用的键盘快捷方式。

注意：这些都是默认的键盘快捷方式。您可以通过 Customize 对话框进行更改。

菜单位置	指令	快捷方式
File（文件）菜单	Open（打开）	[Ctrl] + [O]
	Close active file（关闭活动文件）	[Ctrl] + [F4]

菜单位置	指令	快捷方式
	Save（保存）	[Ctrl] + [S]
	Save All（保存所有）	[Ctrl] + [Shift] + [s]
	Print（打印）	[Ctrl] + [P]
Edit（编辑）菜单	Undo（撤销操作）	[Ctrl] + [Z]
	Redo（恢复操作）	[Ctrl] + [Y]
	Cut（剪切）	[Ctrl] + [X]
	Copy（复制）	[Ctrl] + [C]
	Paste（粘贴）	[Ctrl] + [V]
	Delete（删除）	[Del]
	Select All（选择所有）	[Ctrl] + [A]
	Find（查找）	[Ctrl] + [F]
	Replace（替换）	[Ctrl] + [H]
	Batch Find（批查找）	[Ctrl] + [Shift] + [F]
	Batch Replace（批替换）	[Ctrl] + [Shift] + [H]
	Go To（跳转到）	[Ctrl] + [G]
	Make Uppercase（转为大写字母）	[Ctrl] + [Shift] + [U]
	Make Lowercase（转为小写字母）	[Ctrl] + [U]
	View White Space（查看空白符）	[Ctrl] + [E], [S]
	Incremental Search（增量搜索）	[Ctrl] + [I]
	Comment Selection（注释选择）	[Ctrl] + [E], [C]
	Uncomment Selection（取消注释选择）	[Ctrl] + [E], [U]
	Toggle Bookmark（切换书签）	[Ctrl] + [K], [Ctrl] + [K]
	Toggle All Outlining（切换所有程式码区块的摺叠/展开状态）	[Ctrl] + [M], [L]
	Toggle Outlining Expansion（切换当前程式码区块的收缩/展开状态）	[Ctrl] + [M], [M]
	Start Automatic Outlining（启动程式码区块自动摺叠/展开）	[Ctrl] + [M], [O]
	Stop Automatic Outlining（停止程式码区块自动摺叠/展开）	[Ctrl] + [M], [P]
View（视图）菜单	放大	[Ctrl] + [+]
	缩小	[Ctrl] + [–]
Debug（调试）菜单	Breakpoints（断点）	[Ctrl] + [D], [B]
	Output（输出）	[Ctrl] + [D], [O]
	Watch 1（观察窗口 1）	[Ctrl] + [D], [W]
	Watch 2（观察窗口 2）	[Ctrl] + [Alt] + [W], [2]

菜单位置	指令	快捷方式
	Watch 3（观察窗口 3）	[Ctrl] + [Alt] + [W], [3]
	Watch 4（观察窗口 4）	[Ctrl] + [Alt] + [W], [4]
	Locals（局部）	[Ctrl] + [D], [L]
	Components （组件）	[Ctrl] + [D], [P]
	Call Stack（调用堆栈）	[Ctrl] + [D], [C]
	Memory 1（内存窗口 1）	[Ctrl] + [D], [M]
	Memory 2（内存窗口 2）	[Ctrl] + [Alt] + [M], [2]
	Memory 3（内存窗口 3）	[Ctrl] + [Alt] + [M], [3]
	Memory 4（内存窗口 4）	[Ctrl] + [Alt] + [M], [2]
	Disassembly（反汇编）	[Ctrl] + [D], [D]
	Registers（寄存器）	[Ctrl] + [D], [R]
	Program（编写）	[Ctrl] + [F5]
	Execute Code (Debug) / Resume Execution（执行代码（调试）/恢复执行）	[F5]
	Debug without Programming（仅调试，不编程）	[Alt] + [F5]
	Halt Execution（暂停执行）	[Ctrl] + [Alt] + [Break]
	Stop Debugging（停止调试）	[Shift] + [F5]
	Rebuild and Run（重新编译并运行）	[Ctrl] + [Shift] + [F5]
	Reset（复位）	[Ctrl] + [Alt] + [F5]
	Step Into（单步执行）	[F11]
	Step Over（单步跳过）	[F10]
	Step Out（单步跳出）	[Shift] + [F11]
	Toggle Breakpoint（切换断点）	[F9]
	Address Breakpoint（地址断点）	[Ctrl] + [D], [A]
	File Line Breakpoint（文件行断点）	[Ctrl] + [D], [F]
	Function Breakpoint（函数断点）	[Ctrl] + [D], [U]
	Variable Watchpoint（变量观察点）	[Ctrl] + [D], [V]
	Memory Watchpoint（存储器观察点）	[Ctrl] + [D], [E]
	Delete All Breakpoints（删除所有断点）	[Ctrl] + [Shift] + F9
Build （编译）菜单	Build All Projects（编译所有项目）	[F6]
	Build Active Project（编译活动项目）	[Shift] + [F6]
	Cancel Build （取消编译）	[Ctrl] + [Break]
	Compile File（编译文件）	[Ctrl] + [F6]
Help （帮助）菜单	Help Topics（帮助主题）	[F1]

菜单位置	指令	快捷方式
设计元素控制板	Cancel current operation（取消当前进行的操作）	[Esc]
	矩形工具	[r]
	Ellipse 工具	[e]
	直线工具	[l]
	文本工具	[t]
	圆弧工具	[c]
	图片工具	[m]
	连线工具	[w]
	数据表连接器（网络标签）	[s]
	数字输入	[i]
	数字输出	[o]
	数字输入/输出	[b]
	模拟终端	[a]
	外部终端	[x]
数据表/页面	按一个网格单元移动或滚动	[Up]、[Down]、[Left]或[Right] *
	将所选的模型移动一个网格单元	[Shift] + [Up]、[Down]、[Left]或[Right]
	从网格中移除所选组件**	[Shift] + 左键点击（保持按下[Shift]键）+ 拖动
	禁用/使能弹性连接	[Ctrl] + 左键点击 （保持按下[Ctrl]键）一个或多个对象 + 拖动，或[Ctrl] + [Up]、[Down]、[Left]或[Right]
	放大到所需尺寸	[Ctrl] + 左键点击（保持）空格 + 拖动
	选择多项	[Ctrl] + 左键点击（连续操作）
	平移	[Alt] + 左键点击 + 拖动
	向上/下滚动	将鼠标滚轮向上/下移动
	放大/缩小	[Ctrl] + 将鼠标滚轮向上/下移动
	向左/右滚动	[Shift] + 将鼠标滚轮向上/下滚动
	向上/下远距离滚动	[Page Up] / [Page Down]
	向左/右远距离滚动	[Shift] + [Page Up] / [Page Down]
切换	在各个原理图页面间切换。	[Ctrl] + [F6]（如果使用[Shift]键，则按反方向执行。）
	打开项目文件或打开选项卡式文档。	[Ctrl] + [Tab]（如果使用[Shift]键，则按反方向执行。）
	关闭选项卡式文档。	[Ctrl] + [w]
	返回代码比较器中光标先前所在的位置。	[Ctrl] + [-] （如果使用[Shift]键，则按反方向执行。）
代码编辑器	自动完成	[Ctrl] + [Space]
	查找引用	[Ctrl] + [Shift] + [R]
	跳转至声明位置	[F12]
	跳转至定义位置	[Ctrl] + [F12]

* 如果选定了各个项目，使用箭头键按指定的方向将项目移动一个网格单元。如果选定了数据表/页面，则使用箭头键进行滚动。

** 组件、线和其它特定于设计的对象总是同网格对齐。其它不属于实际设计的模型可自由移动。

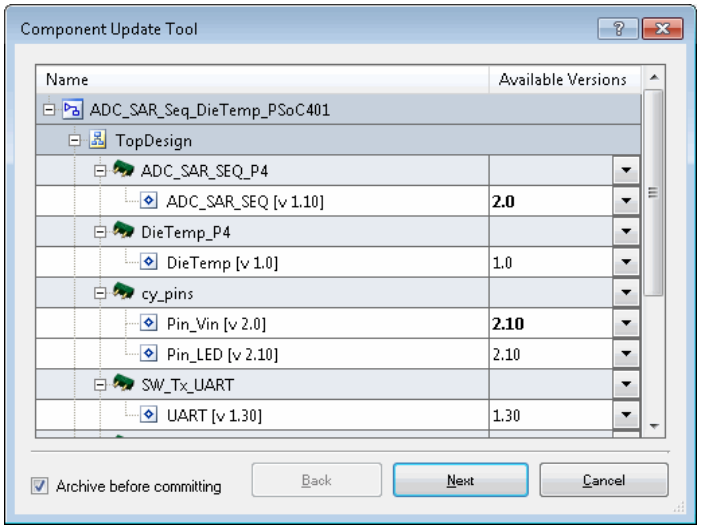
另外，请参考：

- File（文件）菜单
- Edit（编辑）菜单
- 调试器菜单指令
- Build（编译）菜单
- 设计元素控制板
- 自定义对话框

对话框

组件更新

通过组件更新工具对话框，您可以更新设计中的组件版本。如果您在新版本的 PSoC Creator 中打开了旧版本的设计，将自动弹出该对话框。但可以选择关闭自动检查功能。



在 **Available Versions**（可用版本）项下，黑体显示的项目表示的是可以更新的组件。默认情况下，PSoC Creator 会将组件更新为最新版本。

组件更新操作完成后，该工具将生成 *ComponentUpdateLog.txt* 文件，该文件显示了详细的更新内容。该文件被保存在相应的项目文件夹中。

如何打开对话框：

1. 打开具有一个或多个组件的 PSoC Creator 设计项目。

2. 从项目菜单中选择 Update Components 项。

   注意：

   - 如果您打开的设计包含旧版本的组件，并选择了更新这些组件的选项，将自动打开该对话框。
   - 您也可以在 Workspace Explorer 下的 **Source** 选项卡中右键点击项目，并选择 **Update Components** 项。

组件版本状态：

组件更新工具中的 **Available Versions** 列报告了各种组件版本的状态。

- **Obsolete（过时） —** 过时的组件版本需要更新为新版本。PSoC Creator 会将有关该组件版本的警报或出错信息显示在注意列表窗口内。
- **Prototype**（原型） — 试验版本组件，通常只是一个工作示例。它没有经过标准化，也没有经过回归测试。PSoC Creator 常将有关该组件版本的注意显示在注意列表窗口内。在某些情况下，它会显示警报信息。
- **Production**（量产） — 量产版本的组件是满足生产质量的组件；在您的设计中应该使用该版本，而不要使用已经过时或正在试验的版本。如果使用该版本的组件，将不会显示任何信息。

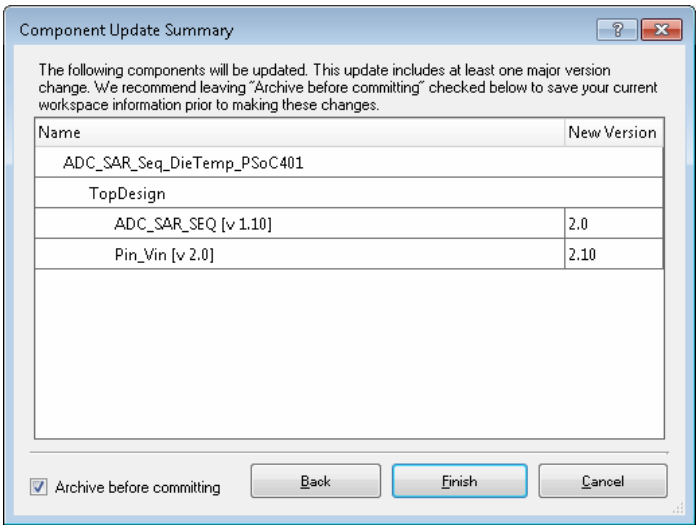
查看数据手册：

在每套组件实例的顶部，点击 **Name**（名称）列中的组件链接，可以显示 Available Versions（可用版本）列中所列版本组件的数据手册。

更新组件：

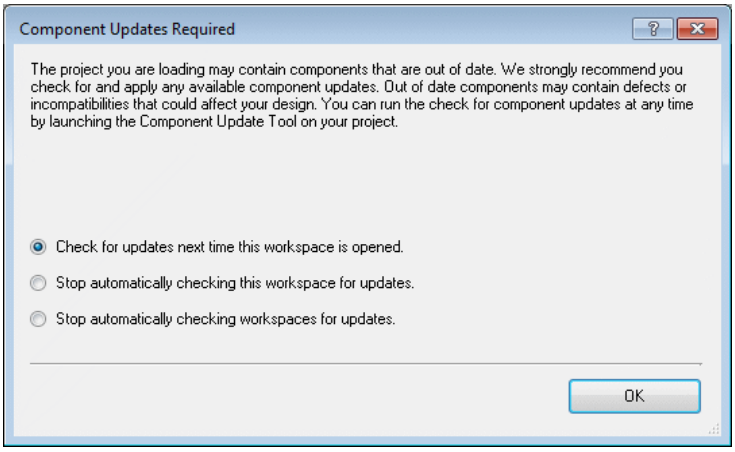
要更新的组件被自动选择，并且在 **Available Versions** 列下以黑体字显示。

1. 若需要，下拉 **Available Versions** 列中每一个组件的菜单，选择您需要的版本。例如，您可以使用该下拉菜单将组件返回到以前的版本。
2. **Archive before committing** 选框将会创建项目的存档。保持选择或取消选择以跳过存档步骤。
3. 点击 **Next** 按键。
4. 查看 Component Update Summary（组件更新摘要）对话框中的信息。



5. 点击 **Finish** 以更新组件或点击 **Cancel** 来取消操作。

   如果您在自动更新期间取消了操作，将显示以下对话框，该对话框为您提了各种选项：



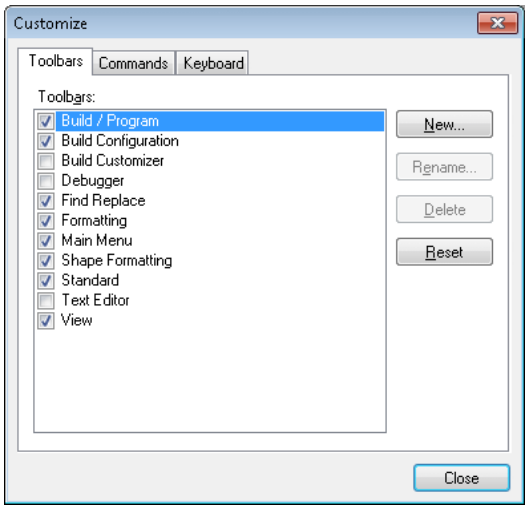
   选择合适的选项，然后点击 **OK**。

另外，请参考：

- 组件/实例
- 工作区浏览器
- 注意列表窗口
- 环境选项

Customize（自定义）对话框

Customize 对话框用于显示不同的工具栏，并自定义工具栏上显示的指令。您也可以创建所有指令的键盘快捷方式。



该对话框共有三个选项卡：

- 工具栏
- 指令
- 键盘

要打开该对话框：

请右键点击 PSoC Creator 菜单/工具栏，然后选择 **Customize**。

Toolbars（工具栏）选项卡

通过 **Toolbars** 选项卡，您可以创建、重命名、移除以及复位工具栏。它显示了可用的工具栏和您创建的工具栏。当出现某个工具栏时，那么该对话框的左边将出现复选框。

- New（新的） — **显示新的工具栏对话框，通过该对话框您可以创建和命名自定义工具栏。**
- Rename（重命名） — **显示重命名工具栏对话框，仅用于更改自定义工具栏的名称。**
- Delete（删除） — **用于删除工具栏列表中选定的自定义工具栏。**
- **Reset**（复位） — 移除在工具栏列表中选定的预定义工具栏中的所有更改，并将其复位为原始状态。只有选择一个内置工具栏时，该项才可用。

注意：您只能对所创建的工具栏进行删除和重命名操作。

Commands（指令）选项卡

通过 **Commands** 选项卡，您可以添加和删除工具栏和菜单中的指令。

- **Categories**（类别） — 指定显示在指令列表框中的指令集。指令类别是根据当前支持环境的工具和设计程序来提供的菜单标题。这个菜单标题的列表是动态的，可以根据工具和设计程序以及其自定义内容来更改类别顺序和菜单标题。因此，可能会有两个来自不同设计程序的菜单具有相同的名称，所以，同一标题可以重复两次，但是它指出的是不同的指令集。

- **Commands**（指令）— 根据您选定的类别显示指令和指令镜像。您可以将指令拖放到想要自定义的工具栏内。
- **Modify Selection**（更改选择） — 显示了一个指令列表，该列表用于自定义工具栏上显示的按键。例如，您可以修改图像或快捷键，以及指定显示指令的文本或图像。当您选择想要自定义的工具栏上的指令按键时，该按键可用。

Keyboard（键盘）选项卡

通过 **Keyboard** 选项卡，可以指定指令的快捷键组合。

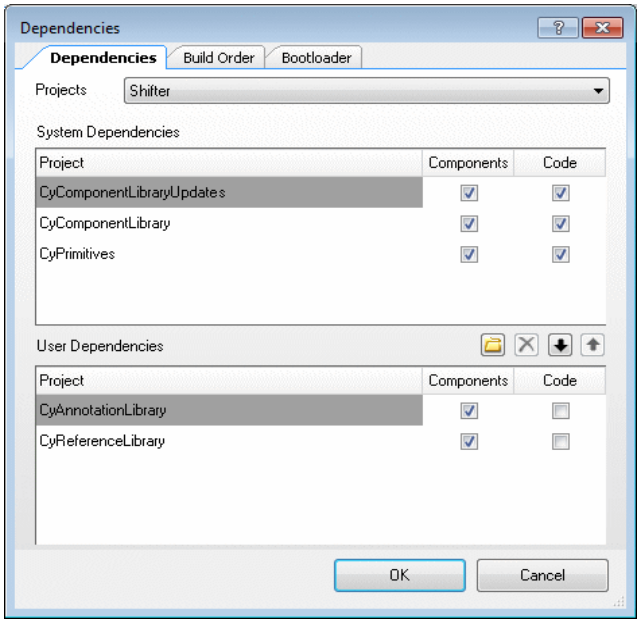
- **Show Commands containing**（显示指令） — 使用该项可对滚动列表中显示的指令进行滤波。
- **Shortcut(s) for Selected Command**（选定指令的快捷方式） — 显示指令的当前快捷方式。若适用，您可以删除**快捷方式**。
- **Use new shortcut in**（使用新的快捷方式） — 下拉菜单，根据特定子系统或全局来指定快捷方式。
- **Press shortcut key(s)**（按快捷键） — 将光标放置在该字段内，然后按下需要使用的快捷键，再点击 Assign。
- **Shortcut currently used by**（快捷方式当前由...使用） — 该字段显示的内容表明当前是否使用了快捷方式，并且表明由哪个函数使用。

另外，请参考：

- 框架说明
- 标准工具栏

Dependencies（依赖属性）

通过 Dependencies 对话框，可以显示并选择一个或多个用户级项目的依赖属性。依赖属性的顺序决定了 PSoC Creator 在搜索组件和代码时使用的秩序。



要打开该对话框，请执行下述操作：

从 **Project** 菜单中选择 **Dependencies...**。

项目：

通过该下拉菜单，您可以选择某个项目使之具有依赖属性。只有在工作区中具有多个项目时，该项才可用。

系统依赖属性：

该区域显示的是您的项目的系统级关系。

每个依赖属性都具有两个选框：**Components**（组件）和 **Code**（代码），它们用于指定搜索路径和/或代码是否具有依赖关系。

用户依赖属性：

通过该区域，您可以添加/删除您项目的其它依赖属性。该区域包括四个按键：**Add**（添加）、**Remove**（删除）、**Move Up**（向上移动）以及 **Move Down**（向下移动）。

注意：如果 Default Dependencies 下的 Project Management Options（项目管理选项）对话框中具有任何默认的依赖属性，则添加依赖属性后，这些依赖属性在该区域中显示为创建项目的依赖属性。

Components/Code（组件/代码）选框：

每个依赖属性均具有两个选框：**Components**（组件）和 **Code**（代码），用于指定搜索路径和/或代码是否具有依赖属性。

- 如果您勾选了 **Components** 选框，它将包含项目中各组件的所有文件。
- 如果您勾选了 **Code** 选框，那么它将包含组件所在项目中的所有代码。此代码并非特定于组件，它还可以包含 C 语言的静态库和宏等。一般情况下，不需要选择该项。

Build Order（编译顺序）选项卡：

Build Order 选项卡是一个只读列表，它会按照您构建项目的顺序进行排列。

Bootloader 选项卡：

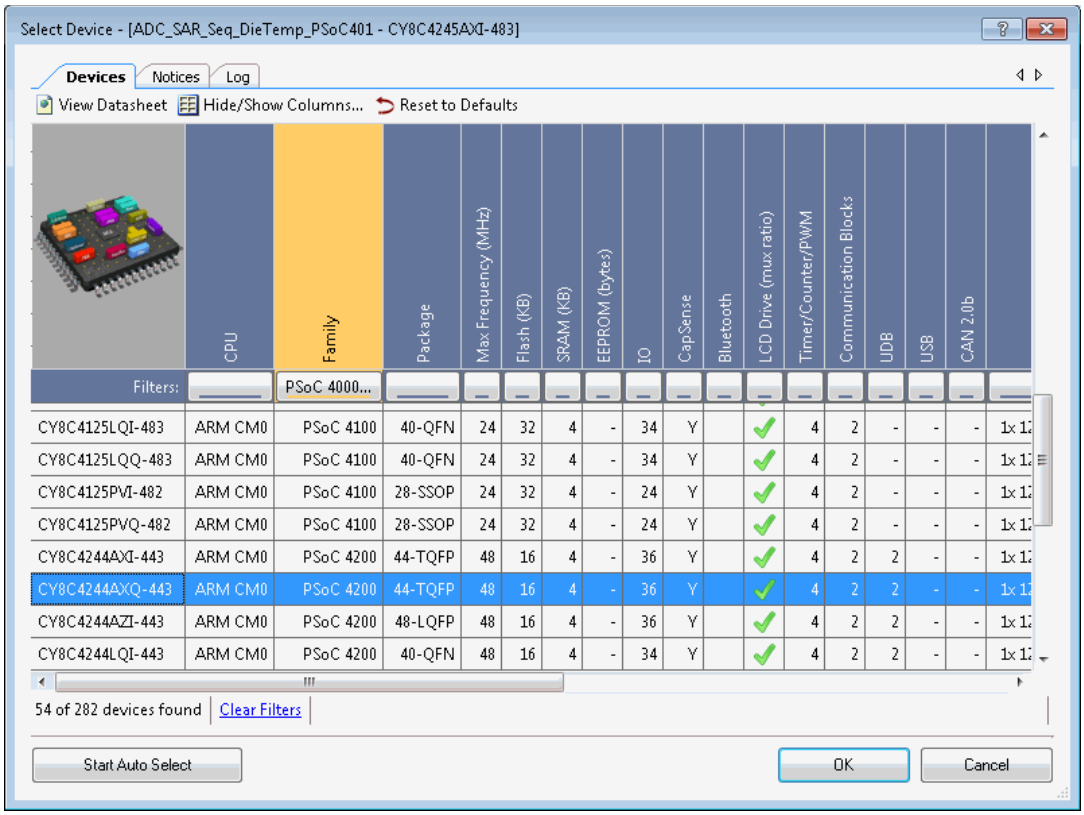
通过 **Bootloader** 选项卡，您可以将某个 Bootloader 项目与某个 Bootloadable 项目关联起来。因此 Bootloadable 项目的编译可以正确计算出它在存储器中的位置。该选项卡只适用于 Bootloadable 项目。关于 Bootloader 项目的更多信息，请参考 Bootloader 和 Bootloadable 组件数据手册。

另外，请参考：

- 基本层级设计
- 项目管理选项

Device Selector（器件选型器）

通过 Device Selector 对话框选择您设计中需要使用的器件。



该对话框提供了可用器件的列表以及每个器件具有的各种特性。使用该对话框时，您可以进行下面各项操作：

- 在设计过程中随时更改器件选择
- 自定义列表中需要显示的器件资源选项
- 使用滤波器来选择只符合特定资源要求的器件（比如支持 USB），隐藏其他器件。

Device Selector 对话框包括下面各选项卡：

- **Devices tab**（器件选项卡）— 显示了供您选择的器件。
- **Notices tab**（注释选项卡）— 显示了运行自动器件选择时生成的所有注释。
- **Log tab**（日志选项卡）— 显示自动器件选择过程中的日志文件。

注意：

- 对话框标题栏显示的是当前的项目名称以及项目使用的器件的名称。
- 器件选型器的布局会保存至下一次打开该工具。当您重新打开某个特定项目的器件选型器时，它将保留上一次使用时的布局和设置。
- 器件选型器将使某个目标项目同特定的芯片版本对应起来。这样，工程编译时所生成的代码只能烧写到已选择器件。

打开对话框：

在 Workspace Explorer 中选择需要更改器件的设计项目，然后执行下面各项操作：

- 依次选择 Project > Device Selector…，或者
- 右键点击所需项目，然后选择 **Device Selector…**

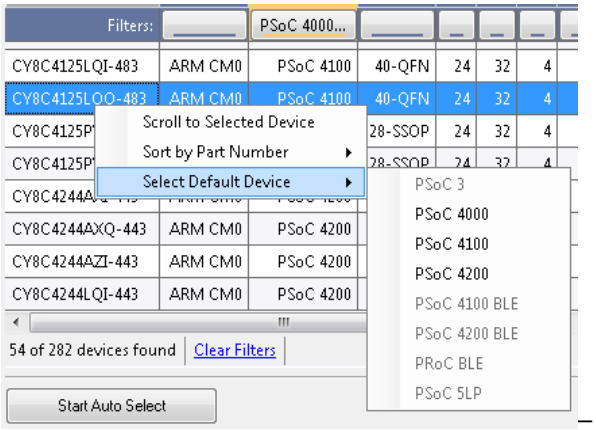
注意：当创建某个新的设计项目时，您还可以通过 New Project 对话框中的 **Device** 下拉菜单打开器件选型器。这两种方法的主要区别是，当打开新设计的器件选型器时，不能使用任何自动选择功能，您只能看到所选项目类型的相应器件。

选择器件：

要想选择某个器件，请执行下面的操作：

- 双击器件表中的某个器件行，或者
- 点击某个器件行，然后点击 **OK**。

如要快速选择特定架构的默认器件，请右键点击器件选型器表中的任意位置，然后选择 **Select Default Device**（选择默认器件）项，再选择您所需要的架构。



对器件目录进行分类：

通过点击某个列的顶部，您可以按递增/递减的顺序排列该列。再次点击该列的顶部，可以翻转排列顺序。对某个列进行排序时，该列顶部文本下方会出现一个箭头。

某些情况下，器件选型器中的部件型号列被排序。如要重新排列部件型号，请右键点击器件选型器表中的任意位置，并选择 **Sort by Part Number**（按部件型号排列）项，然后选择您所需要的排序方式。通过该菜单，还可以按任意的形式自动滚动到当前选择的器件。

查看器件数据手册

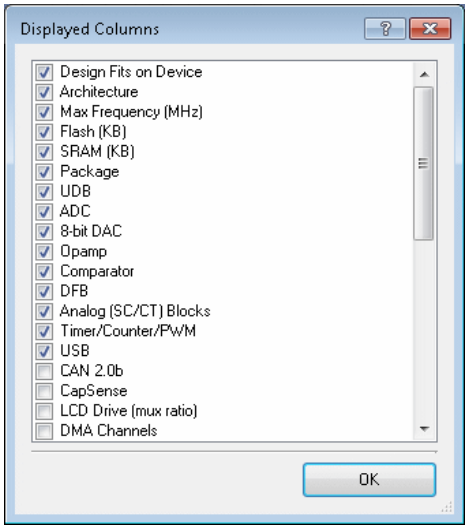
点击 **View Datasheet**（查看数据手册）。

将打开一个网页，该网页显示了所选器件的数据手册。

显示/隐藏各列：

点击 **Hide/Show Columns**（显示/隐藏各个列）。

该选项用于控制器件表中的隐藏/显示功能。通过该项，可以打开罗列了所有可用列的 Display Columns（显示各个列）对话框。



通过勾选/取消勾选某一列，您可以更改器件表中所显示的内容。

注意：您更改各个选框的状态后，更改内容会立即生效。不能隐藏应用滤波器的列。在该对话框中，这些列被禁用，并且在其后面附上了“- Filtered”（被滤波）这个字符，以说明为何不能隐藏它们。

器件表上面显示的是状态信息，以表明被隐藏的列数。如果没有任何列被隐藏，则也不会显示状态信息。

滤波器件表：

器件选型器中的每一列都有一个下拉按键，用于对该列进行滤波。当按下该按键时，将出现一个显示选框列表的下拉窗口，该列中的每个可能值均有相应的选框。根据需要，您可以任意取消选择各个值，并从所显示的各个器件中移除带有该值的器件。在您关闭窗口或按下**[Enter]**键前，这些更改不会生效。（若按下**[Esc]**键，该窗口将被关闭，但更改内容不会生效）。默认情况下，将选中所有值，表示尚未进行任何滤波操作。

当对某一列进行滤波器时，该列的顶部状态（和按键的强调色）将发生变化，以明确表示应用了滤波器的列。该按键还显示了状态为“打开”的值。如果多个值均处于“打开”状态，那么只有第一个值显示在该按键上。该值后面带有“…”，表示当前显示的还有其他值。在这种情况下，通过在该按键上悬停鼠标，可以使用一个工具提示来显示当前打开的所有值。

器件表下面有一个状态栏，它指示当前显示的器件数量（即为通过所有滤波器的器件数量）以及现有的器件数量。此处还提供了一个链接，用于清除该表中所有的滤波器，从而使各个器件可见。

恢复默认设置：

点击 **Reset to Defaults** 项。

通过该项，可以将所有设置项恢复为其默认状态（即为移除滤波器、重新排列各个列、恢复显示的滤波器等）。

自动选择器件：

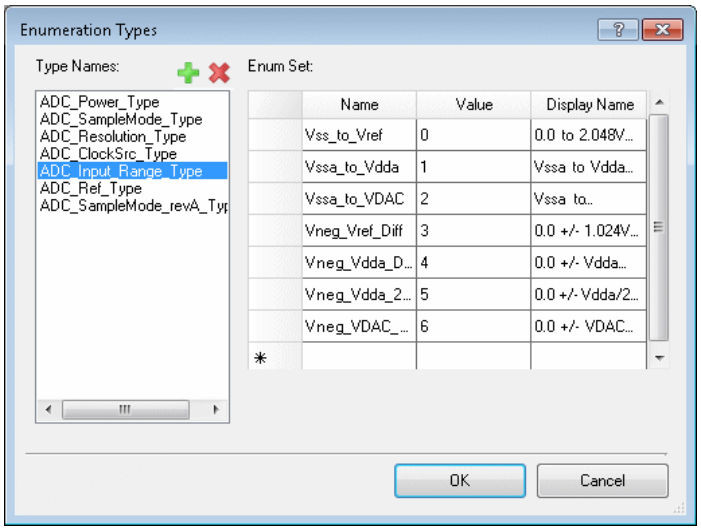
自动器件选择功能可帮助您确定要选择的器件。当您点击 **Start Auto Select**（启动自动给选择）按键，PSoC Creator 将检查器件选型器表中的每一个可见器件，以确定当前的设计是否工作于该器件。当选中每个器件时，**Design Fits on Device**（设计符合于器件）列上的值可以为 Unknown（未知）、Yes（是）或 No（否）。您可以使用这些值进行排列和滤波。在自动器件选择过程中，**Notices** 和 **Log** 选项卡被相应更新。这些选项卡仅适用于自动器件选择过程。

另外，请参考：

- 工作区浏览器
- 创建新项目

枚举类型

通过 Enumeration Types 对话框，您可以为您的组件创建并编辑枚举值（或用户定义类型）。在创建自定义组件时，该对话框的与定义组件参数 j 配合使用。创建自定义组件的过程比较复杂。关于如何设置该对话框，您可以按下**[F1]**键，从 Help 文件中找到相关信息。更多有关创建组件的详情，请参阅组件作者指南。



一个组件可能包含多个枚举类型，但每个类型只能有唯一的一个名称，并且各个类型中的每个键(key)必须是唯一的。已给项目中每个组件所定义的类型将被集中到该项目（项目类型缓存区）中。设计搜索路径中来自所有项目的项目类型缓存又被集中到设计类型缓存区内。

- 如果通过符号识别类型和类型键，可以通过使用这些类型和键的“短名称”访问它们。
- 如果不能通过符号识别类型和类型键，那么请使用这些类型和键的“长名称”（即为组件名称加上实际类型名称和键名称（两者使用“__”来分开））。

当评估某个表达式的类型或按键时，评估系统会先检查标识符的内附符号。如果找不到该符号，则评估系统将检查设计类型缓存区。如果设计类型缓存区不能识别标识符，该标识符将被识别为未知的标识符。

打开对话框：

点击 Parameters Definition 对话框中的 **Types...**按键。

添加枚举类型：

1. 点击 **Type Names** 旁边的 **Add** 键，以创建新的枚举类型。
2. 使用您想要的名称替换“EnumType_1”，然后按下**[Enter]**键。
3. 在 **Enum Set** 下方，点击 **Name** 下的第一行，并键入枚举类型的第一个名称/值对的名称；在 **Value** 下面键入一个数值或接受默认的设置。
4. 此外，还可以在 **Display Name** 中输入一条字符串。这样，该参数的字符串将显示在组件 Configure（配置）对话框的下拉菜单中。
5. 根据需要输入枚举组，然后点击 **OK**，以关闭 Enumeration Types（枚举类型）对话框。

删除枚举类型：

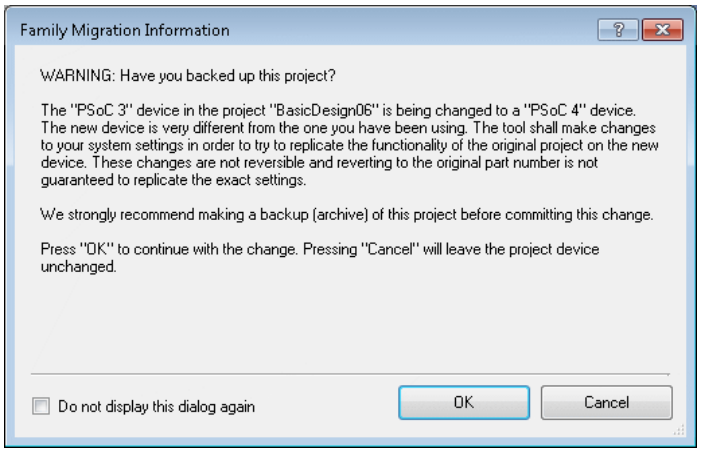
高亮显示要删除的类型，然后点击 **Delete** 。

另外，请参考：

- 符号编辑器
- 参数定义对话框
- 组件创建指南

系列移植信息

当您尝试将 PSoC 4/PRoC BLE 项目更改为其他系列或进行相反操作时，将出现 Family Migration Information 对话框。该对话框提示您，更改器件可能要求手动编辑设计。



打开对话框：

当您尝试使用 Device Selector（器件选型器）将您的项目更改成/更改自 PSoC 4/PRoC BLE 器件时，该对话框将出现。

使用对话框：

点击 **OK**，关闭该对话框，并更改器件。

点击 **Cancel**，关闭对话框，并保持项目器件的设置不变。

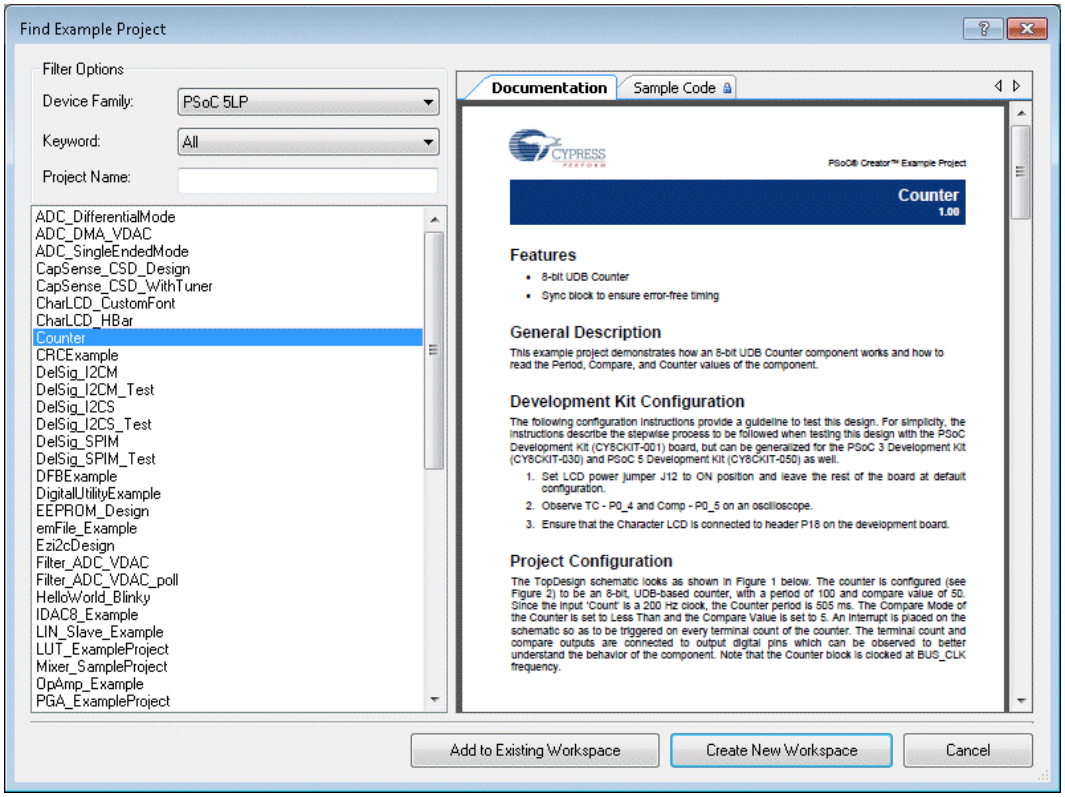
勾选 **Do not display this dialog again**（下次请勿显示该对话框）选框，这样可防止将来出现该对话框。通过使用 Options 对话框中的设置，可以重新启用该功能。

另外，请参考：

- 器件选型器
- Options 对话框

查找示例项目

通过 Find Example Project 对话框，您可以查找并打开 PSoC Creator 中所包含的示例项目。示例项目说明了如何配置各种组件，并包含了相同的代码，以更好地了解如何使用某个组件。



大多数项目都附带了包含设置和配置信息的说明文档。当您从列表中选择某个项目时，相应文档将显示在 **Documentation** 选项卡下。此外，您可以在 **Code Sample** 选项卡下查找可用的示例代码。

当您打开某个示例项目后，可以通过双击 Workspace Explorer 中的文件来打开它们。

打开 Find Example Project 对话框：

选择起始页上的 **Find Example Project...**项，或选择 **File** 菜单上的 **Example Project...**。

要想打开包含了与相应组件相关联的项目列表的对话框，请右键点击 Component Catalog 中的某个组件，或右键点击设计中的某个示例，然后选择 **Find Example Project...**项。

选择项目：

1. 根据要求，在 **Filter Options** 下面选择各个滤波器，以缩小可用的项目列表。
   - 项目的 **Device Family**（器件系列）选项分别为：PSoC 3、PSoC 4000、PSoC 4200 BLE、PSoC 5LP、PRoC BLE、All（所有）等。
   - 如果需要，也可以从下拉菜单选择一个 **Keyword**（关键词）。该关键词可以是某个组件的名称，也可以是示例项目开发者区分各个项目时所使用的不同关键词。
   - 键入项目名称。
2. 从 **Filter Options** 下面窗口中的列表选择某个项目。

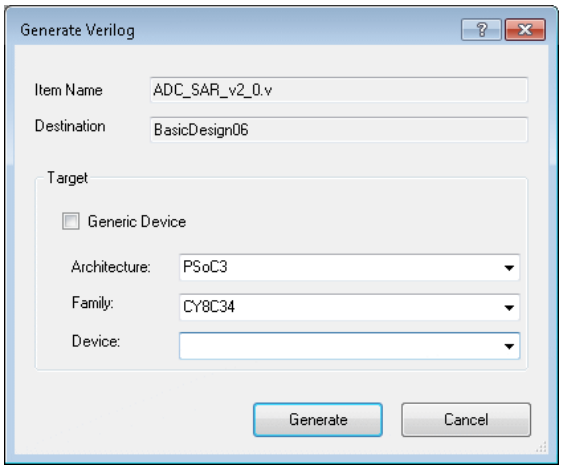
   该区域显示的是满足已给选择条件的项目列表。
3. 可以在相应的选项卡下查看项目的 **Documentation** 和/或 **code**。
4. 根据需要，请点击 Add to Existing Workspace（添加到现有工作区中）或 Create New Workspace（创建新的工作区）。
   - 如果尚未打开任何项目，那么只有 **Create New Workspace** 选项可用。通过该选项可以在 PSoC Creator 的当前实例中打开项目。
   - 如果您已经打开某个项目，但仍选择 **Create New Workspace**，这时，该新的项目将在 PSoC Creator 独立的实例中打开。
   - 如果您选择的是 **Create New Workspace**，那么将打开“Browse”窗口，这样您可以选择用于复制示例项目的位置。
   - 如果您选择的是 **Add to Existing Workspace**，则该示例将作为另一个项目并被添加到当前工作区中。

另外，请参考：

- 工作区/项目
- 组件/实例
- 打开现有的项目

Generate Verilog 对话框

通过 Generate Verilog 对话框，您可以选择为您组件符号生成的 Verilog 文件的架构、系列和/或器件。创建某个组件时，要使用该对话框。更多有关创建组件的详情，请参阅组件作者指南。



打开对话框：

当您从 Symbol canvas context menu（符号画布右键菜单）中选择 **Generate Verilog** 指令时，将自动打开该对话框。

选择架构/系列/器件：

通过 **Target** 选项，可以指定 Verilog 文件所使用的具体架构、系列和/或器件。

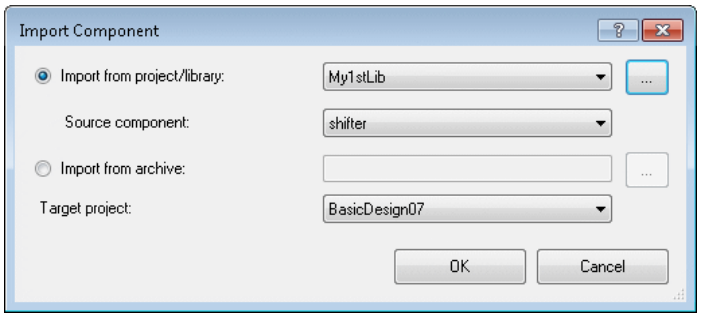
- 选中 **Generic Device** 默认设置，以将 Verilog 文件适用于所有器件。取消选择该选框，可以启用下拉菜单中的 **Architecture**、**Family** 和 **Device**。
- 选择 **Architecture**，可以在子文件夹中创建一个架构的 Verilog 文件。
- 选择器件 **Family**，这样可以在一个子文件夹内创建器件系列的 Verilog 文件。
- 选择特定的 **Device** 部件型号，这样可以在一个子文件夹内创建特定器件的 Verilog 文件。

另外，请参考：

- 组件创建指南
- 符号编辑器右键菜单
- 添加组件

导入组件

通过 Import Component 对话框，可以从某个项目中导入一个组件。使用该对话框还可以从某个组件存档文件中导入一个或多个组件。



注意：所导入的组件保留了原始组件的所有符号和组件属性。它也包含了 Component Catalog Placement（组件目录放置）。如果您的项目中相同的组件的 Dependencies（依赖属性）不一样，那么 Component Catalog 将按照优先级最高的依赖属性显示组件。

该对话框包括下面各选项：

- **Import from project/library**（从项目/库导入）— 用于选择包含了想要导入组件的项目/库。
  - **Source Component**（源组件）— 当从某个项目/库导入时，通过该项，可以指定该项目/库中要导入的组件。
- **Import from archive**（从存档中导入）— 通过该项，可以选择包含需要导入组件的组件存档。更多信息，请参阅导出组件部分。存档中所有的组件均被导入。
- **Target Project**（目标项目）— 用于选择导入组件的具体项目。

打开对话框：

1. 选择 Workspace Explorer 中的 **Components** 选项卡。
2. 右键点击某个项目，并选择 **Import Component** 项。

导入组件：

1. 选择 Import from project/library 项或 Import from archive 项。
   - 如果您选中了 **Import from project/library**，请选择包含需要导入的组件的项目。如果需要，点击浏览按键**[...]**，并导航到包含要导入的组件的文件夹。在 **Source Component** 中，选择要从项目/库导入的组件。
   - 如果您选中了 **Import from archive**，请选择相应的组件存档(.cycomp file)。如果需要，点击浏览按键**[...]**，并导航到包含要组件存档的文件夹。
2. 在 **Target Project** 中，选择放置组件的项目。
3. 点击 **OK**。

PSoC Creator 会将（各个）导入组件添加到选定的项目。

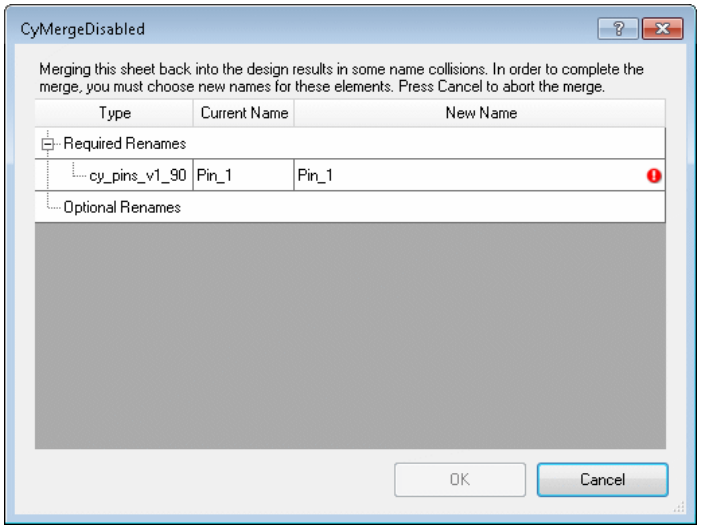
另外，请参考：

- 依赖属性
- 组件目录
- 定义目录放置

- 导出组件
- 工作区浏览器

Merge 对话框

当您尝试启用某个 Disabled Schematic page（被禁用的原理图页面），并且一个或多个组件和/或导线实例的名称相同时，将出现 Merge（合并）对话框。



使用对话框：

在 **New Name** 字段中选择某个实例，给该实例键入其他名称，然后点击 **OK**。

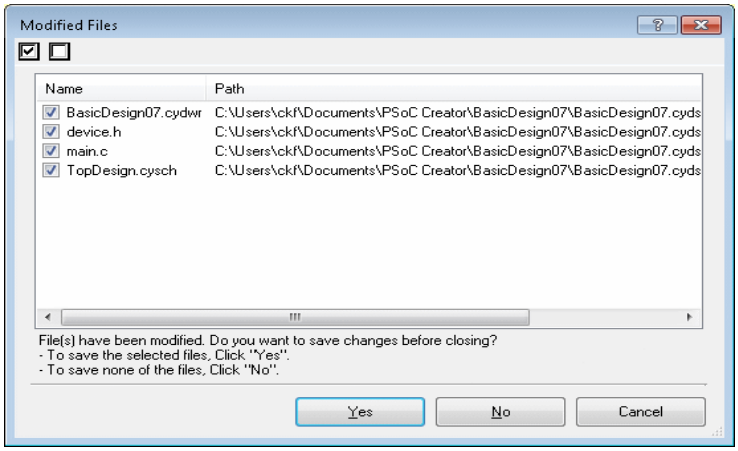
注意：必须重命名组件，以解决名称冲突的问题。但某些连线和实例无需重命名。 在所有必要冲突得到解决前，OK 按键无效。

另外，请参考：

使用被禁用的原理图页面

修改文件

当关闭某个项目或工作区时，通过 Modified Files（更改文件）对话框，你可以选择性地保存文件。



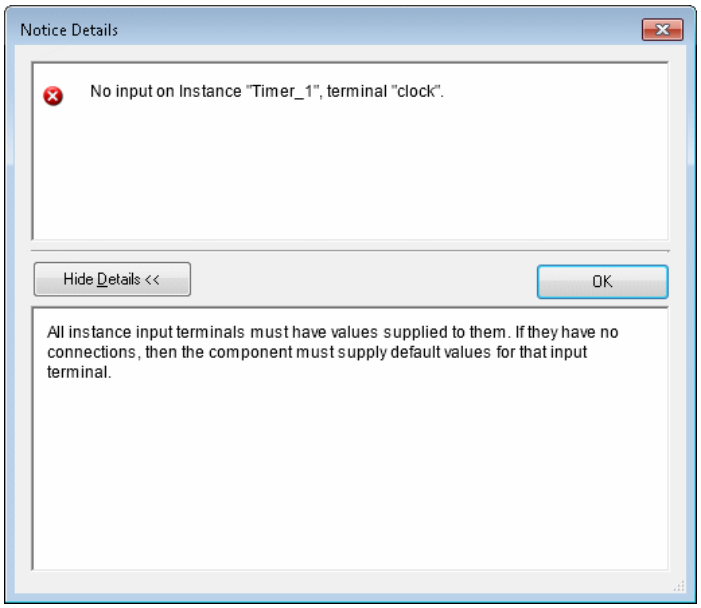
如果您尝试关闭某个需要保存的文件或工作区/项目，将自动出现该对话框。

使用对话框：

- 点击 **Yes** 以保存选定的文件。
- 点击 **No**，则不会保存任何文件。
- 点击 **Cancel**，关闭该对话框，但保持各个文件的打开状态。

公告详情

Notice Details（公告详情）对话框显示了 Notice List 窗口中各条信息的扩展信息。Notice Details 对话框将显示整条信息，以及附加信息（若有）。



打开对话框：

在 Notice List 窗口中，点击该条信息旁边的按键，或点击 **View Details** 按键。

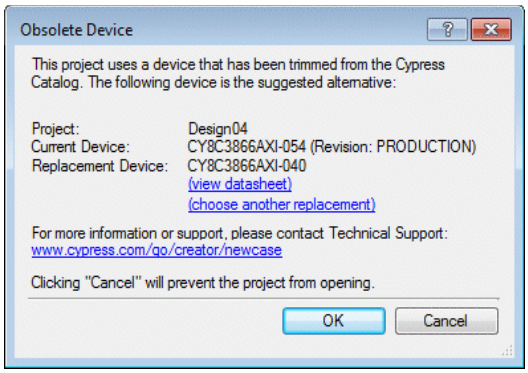
如果您双击了该注释，那么您可以使用一个指令来查看详细信息。

另外，请参考：

- 注意列表窗口

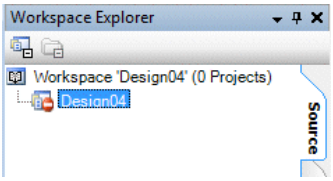
停产器件

当您打开从赛普拉斯器件目录中已经移除的器件项目时，将出现 Obsolete Device 对话框。该对话框推荐了最适合您器件的的下一代产品，并提供了有关推荐器件数据手册的更多信息的链接。



该对话框包含三个选项：

- 点击 **OK** 以在推荐器件中打开您的项目。
- 点击 **choose another replacement**（选择别的替代），打开 Device Selector，然后选择其他器件。
- 点击 **Cancel** 以打开未加载项目的工作区。



注意：如果您选择了器件选型器，但没有选择任何器件，那么该项目仍为未被加载的项目。

如果您重新加载未加载的项目，那么 PSoC Creator 将为您的项目选择默认器件。如果需要，您可以使用器件选型器更改该设置。

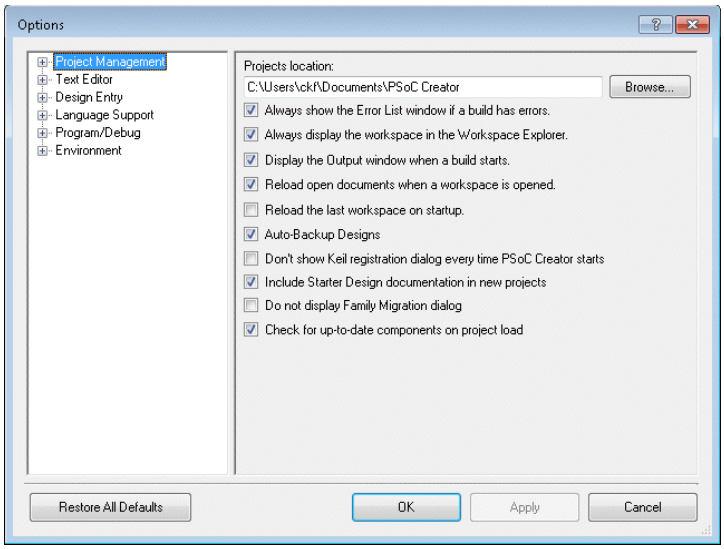
另外，请参考：

- 器件选型器
- 打开现有的项目

Options 对话框

Options 对话框

通过 Options 对话框，您可以指定 PSoC Creator 的各种设置，如存储项目的位置或导线的颜色。



该对话框包括下面各目录：

- Project Management（项目管理）（本主题）
- 文本编辑器
- 设计入口
- 语言支持
- 编程/调试
- 环境

打开 Options 对话框：

从 **Tools** 菜单中选择 **Options** 项。

恢复默认设置：

点击 **Restore Defaults** 项。

所有选项将被恢复为默认配置。

Project Management Options（项目管理选项）：

可通过 Options 对话框的 Project Management 目录，进行设置各种项目管理选项。

该目录包括下面的内容：

General（通用）：

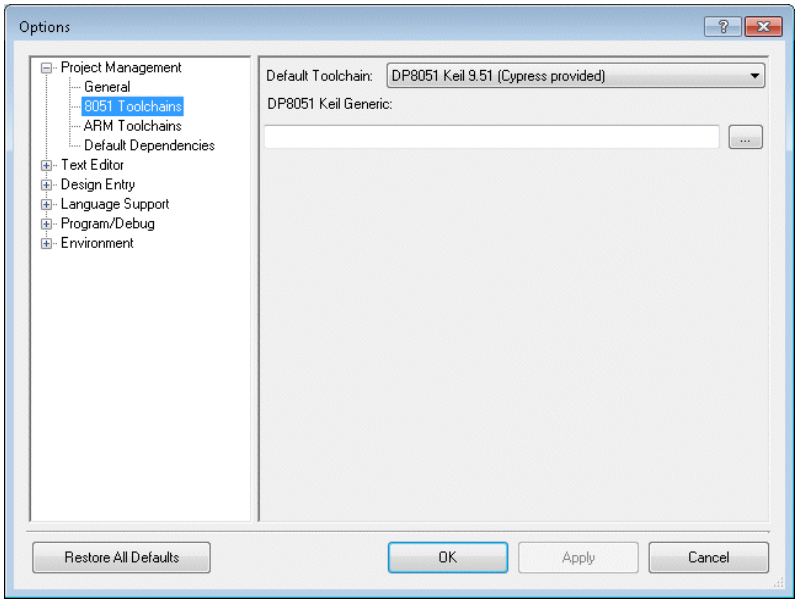
在 General 区域中，您可以指定保存您项目的位置。此外，各种选项还有很多与其相对应的选框，具体包括：

- Always show the Error List window if a build has errors（如果编译发生错误，则始终显示错误列表窗口）— 是（默认）或否。
- Always display the workspace in the Workspace Explorer（始终在工作区浏览器中显示工作区）— 是（默认）或否。

- Display the Output window when a build starts（开始编译时显示输出窗口）— 是（默认）或否。
- Reload open documents when a workspace is opened（打开某个工作区时，重新加载打开的文档）— 是（默认）或否。
- Reload the last workspace on startup（启动时重新加载最后的工作区）— 是或否（默认）。
- **Auto-Backup Designs**（自动备用设计）— 是（默认）或否。如果选中了该项，当您打开使用了上一个 PSoC Creator 版本创建的某个设计时，旧版本的设计将被保存到某个存档中（位于[Workspace Folder]/Backup 目录下）。这个备用“副本”文件的名称中包含了最后一次保存该文件的工具版本（即为可用于打开该旧版本的工具）。
- **Don't show Keil registration ...**（不用显示 Keil 注册）— 是或否（默认）。如果选中该项，则每次您启动 PSoC Creator 时，都不会显示 Keil Compiler Registration 对话框。
- **Include Starter Design documentation in new projects**（新项目中包含启动设计文档）— 是（默认）或否。如果选中该项，在 Starter Designs 中创建的新项目将包含设计的文档。
- **Do not display Family Migration**（不显示系列移植）对话框 — 是或否（默认）。如果选中该项，在移植到 PSoC 4/PRoC BLE 器件项目或者从 PSoC 4/PRoC BLE 器件项目移植出时，将不再显示 Family Migration Information（系列移植信息）。
- **'Component not supported for the toolchain' message is an error**（‘工具链的组件不受支持’是一条错误信息 — 是或否（默认）。若果选中该项，则在 Notice List 窗口中，已给工具链的组件不受支持将显示为一条错误信息。
- **Check for up-to-date components on project load**（检查项目加载上的最新组件）— 是（默认）或否。当加载某个旧项目时，通过该字段可以检查该组件是否是最新的。如果检测到旧版本的组件，那么 Component Update（组件更新）工具将启动项目启动。

8051 工具链：

该部分内容讲解了设置默认的 8051 工具链。



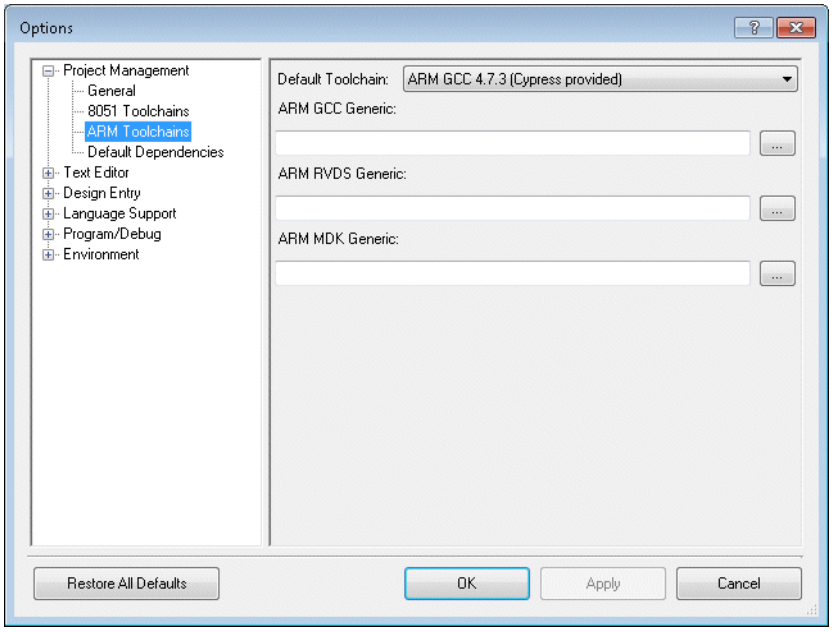
通过该选项，您可以指定每个特定工具链的二进制文件的路径。

- **DP8051 Keil Generic**（DP8051 Keil 通用）— 其默认安装位置为 C:\Keil\C51\BIN\

  位于所选文件夹的示例文件包括：*A51.EXE*、*C51.EXE*

ARM 工具链：

该部分用于设置默认的 ARM 工具链。



通过该选项，您可以指定每个特定工具链的二进制文件的路径。

- **ARM GCC Generic**（ARM GCC 通用）— 没有默认的安装位置。它可以是您解压文件的任意位置。

  应该位于所选文件夹中的示例文件包括：*arm-none-eabi-addr2line.exe*、*arm-none-eabi.ar.exe*

- **ARM RVDS Generic**（ARM RVDS 通用）默认的安装位置相同于 C:\Program Files\ARM\RVCT\Programs\4.0\529\win_32-pentium\（根据具体的版本和构建，这些数字可能不一样）

  应该位于所选文件夹中的示例文件包括：*armar.exe*，*armasm.exe*

- **ARM MDK Generic**（ARM MDK 通用）— 默认的安装位置为 C:\Keil\ARM\BIN40\

  应该位于所选文件夹中的示例文件包括：*armar.exe*，*armasm.exe*

注意：某些工具链（如 RVDS）要求使用环境变量，以寻找它的 bin、include 和库路径。这些变量均通过工具链安装程序设置。为使用这些新的环境变量，您可能要重新启动 PSoC Creator。

默认的依赖属性：

通过该项，您可以为所有新建项目指定默认的库。在 Dependencies 对话框中，这些库显示为用户依赖属性。默认的依赖属性会根据具体用户适用于所有新项目。因此，您所有的新项目中都将包含这些依赖属性。

注意：对于在 PSoC Creator 1.0 中创建的，并在当前 PSoC Creator 版本中首次打开的项目，它们将被更新，以包含当时定义的所有默认依赖属性。当前 PSoC Creator 版本的现有项目**不**被更新，因此，它们不会包含将来任何默认的依赖属性。

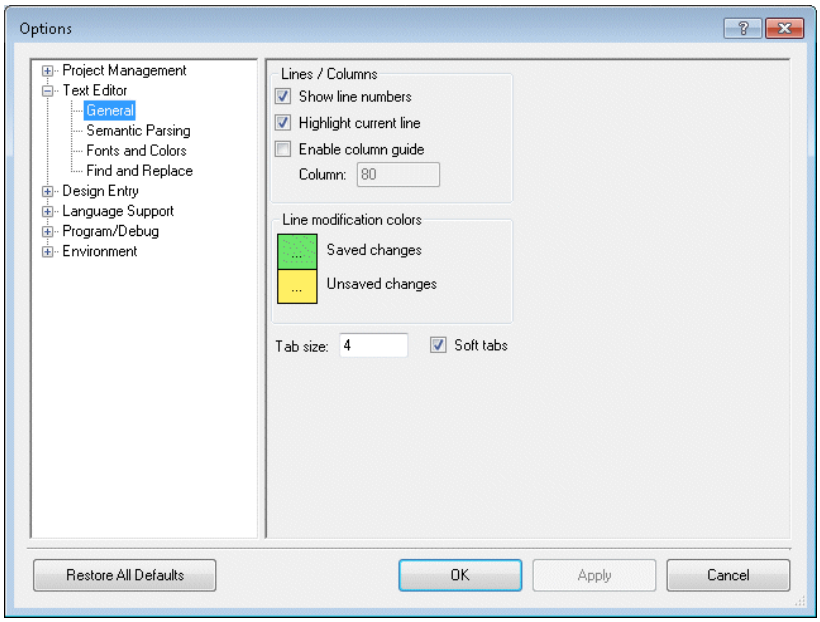
该区域包括四个按键：**Add**（添加）、**Remove**（删除）、**Move Up**（向上移动）以及 **Move Down**（向下移动）。每个依赖属性都具有两个选框：**Components**（组件）和 **Code**（代码），它们用于指定搜索路径和/或代码是否具有依赖关系。

另外，请参考：

- Keil 编译器
- Dependencies（依赖属性）
- 工具链文档

文本编辑器选项

Options 对话框的 Text Editor 目录用于设置代码编辑器的各种选项。



该目录包括下面的内容：

- General（通用）
- Semantic Parsing（语义分析）
- Fonts and Colors（字体与颜色）
- Find and Replace（查找与替换）

General（通用）：

该部分介绍了用于更改各种编辑性能的各选项：

Lines/Columns（行/列）

- **Show Line Numbers**（显示行数）该选框用于控制是否显示行数。
- **Highlight Current Line**（高亮显示当前行）— 通过该选框，可以启用或禁用以灰色显示当前行的当前行指示器。

- **Enable Column Guide**（启用列指南）— 通过该选框，可以启用列长度指南。若果使能了该项，那么可以指定列的长度（单位为字符）。

Line Modification Colors（行更改颜色）

通过这些选项，您可以更改 **Saved Changes**（保存更改） 和 **Unsaved Changes**（未保存更改）边缘显示的颜色。

点击色彩框，以显示 Color selection 对话框。

Tab size 字段

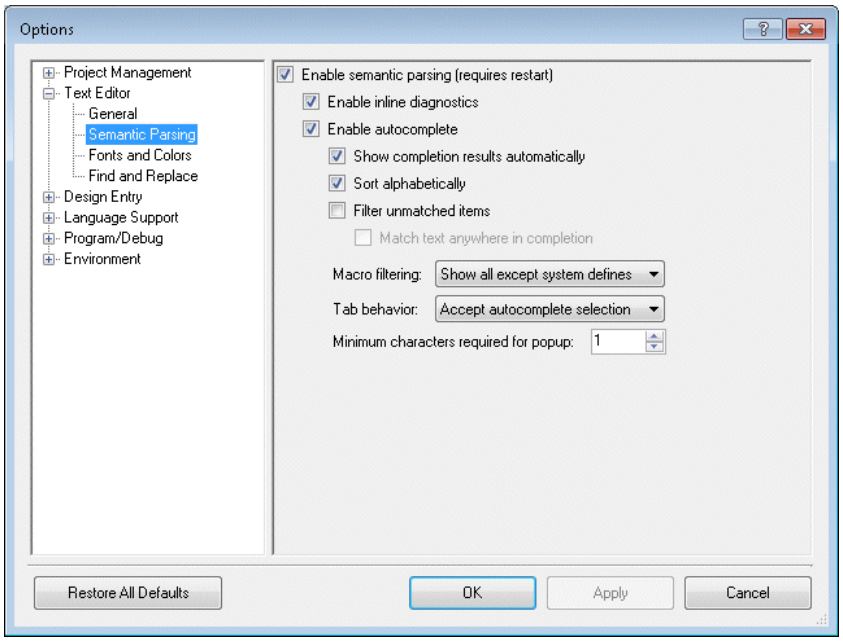
通过 **Tab Size**（跳格大小）字段，您可以指定跳格的长度（单位为字符）。

- 若果您启用了 **Soft Tabs**（软跳格）功能，并按下**[Tab]**键，那么文本编辑器将使用指定的空格数量，而不是一个跳格键点击。

  注意：打开或关闭 **Soft Tabs** 功能并不会影响先前插入的跳格。

Semantic Parsing（语义分析）：

该部分介绍了用于更改自动完成和内联诊断设置的各选项：



Enable semantic parsing（启用语义分析）

通过该项，可以启用或禁用高级的编辑器功能，如自动完成、内联诊断、导航到定义、以及代码浏览器工具窗口等。

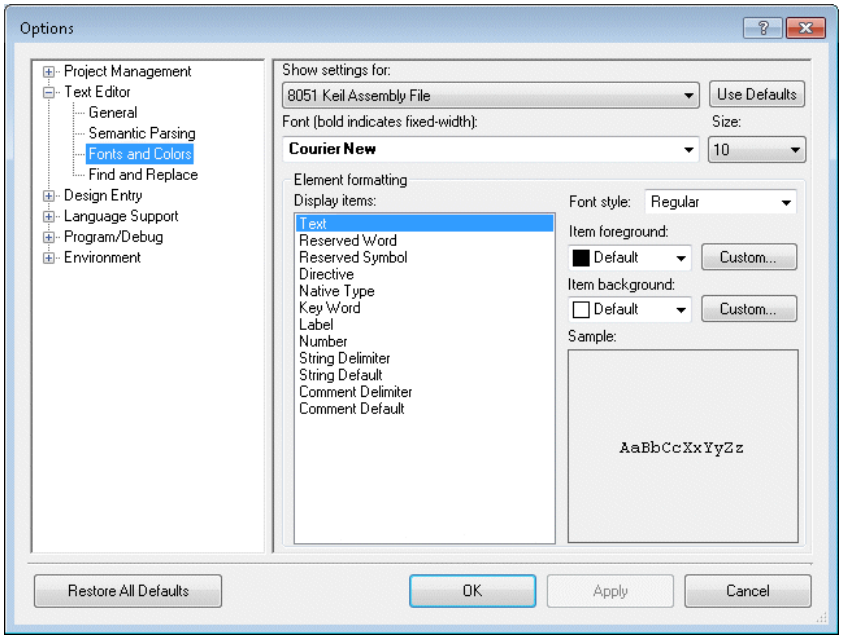
- **Enable inline diagnostics**（启用内联诊断）如果选中 **Enable semantic parsing** 选框，该选框将被启用，以打开和关闭 inline diagnostic 功能。
- **Enable autocomplete**（启用自动完成）如果选中了上述 **Enable semantic parsing** 选框，该选框将被启用，以打开和关闭 Autocomplete 功能。
  - **Show completion results automatically**（自动显示完成结果）— 如果选中了 **Enable autocomplete** 选框，则该选框将被使能，以显示完成结果。如果该选框被选中，将显示各项内容；否则，不会显示任何内容。

- **Sort alphabetically**（按字母顺序排列）— 如果选中 **Enable autocomplete** 选框，该选框将被启用，以打开和关闭自动完成功能的排列机制。如果该选框被选中，将按照字母顺序排列各项。否则，将按照预期值的顺序排列这些内容。
- **Filter unmatched items**（滤波不匹配项）— 通过该项，可以从弹出式菜单中移除不匹配的字符串。这样，各个选项将根据您键入的内容而减少。默认情况下，此选项未被启用。
  - 如果选中了 **Filter unmatched items** 选框，所有搜索操作均能使用 **Match text anywhere in completion**（在完成中的任意位置使文本匹配）选框，以便能够在字符串中间返回匹配结果，而不仅是从字符串的开头返回。例如，“PWM”可能是“PWM_1”和“MyPWM”的匹配结果。默认情况下，此选项未被启用。
- **Macro Filtering**（宏滤波）— 通过使用该项所包含的下面各选项，可以从弹出式列表中移除各个宏 (#define)：
  - Show all (can be slow)（全部显示）（可为低速）
  - Show all (except system defines)（全部显示）（系统 define 除外）（默认）
  - 仅显示宏函数
  - 不显示任何内容
- **Tab behavior**（跳格键性能）— 当显示弹出式菜单时，通过该项可以确定如何使用**[Tab]**键。
  - Accept autocomplete（接受自动完成）（默认）
  - Close autocomplete window（关闭自动完成窗口）
  - Perform UNIX-style Tab-completion（执行 UNIX 型跳格键完成）（**[Tab]**键在所键入的字中添加尽可能多的字符，以便在完成前结果仍是匹配组中的某一情况。
- **Minimum characters required for popup**（弹出需要的最小字符） — 该选项用于指定在自动完成弹出显示前需要输入的字符数量。有效范围为 1 至 10。默认值为 1。

  注意：已经显示弹出窗口时，该选项将被忽略。

字体与颜色：

本节介绍的是用于更改字体和颜色的各个选项：



- **Show settings for**（显示设置） — 下拉该项以选择编辑器将使用的各种文件类型。
- **Use defaults**（使用默认设置） — 选择该项后当前选择的文件类型将返回 PSoC Creator 所提供的原始值。
- **Font**（字体）—用于选择 **Display items**（显示项）的字体类型的下拉菜单。
- **Size**（大小） — 用于选择 **Display items**（显示项）的字体大小的下拉菜单。
- **Display items**（显示项） — 用于选择编辑器中可更改的各种项的列表。
- **Font style**（字体格式） — 用于选择字体的不同格式的下拉菜单，各选项包括：粗体字、斜体字、删除线和下划线。
- **Item foreground/background**（项目前景/背景） — 用于选择 **Display items**（显示项）和背景的颜色的下拉菜单。
  - Custom（自定义）— 它是用于打开自定义颜色选择器的按钮。

查找与替换：

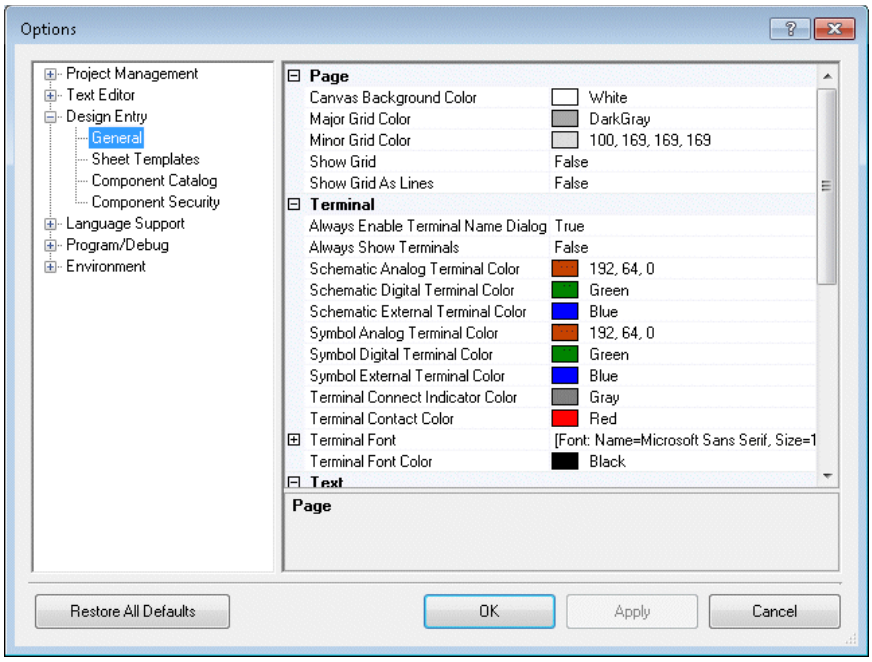
- **Display informational messages**（显示信息） — 选择该复选框后，将显示与 PSoC Creator 中的查找与替换相关的信息。
- **Automatically populate Find What ...**（自动填充 Find What 字段） — 选择该复选框后，文本编辑器中的高亮显示字将显示在 Find What 字段内。

另外，请参考：

- Options 对话框
- 代码编辑器
- 查找与替换

设计输入选项

通过 Options 对话框的 Design Entry 目录可设置设计输入工具的各种选项。



该目录包括以下内容：

General（通用）：

General 部分包含多个字段，这些字段用于指定设计输入工具中几乎所有物体的颜色、宽度、大小和字体。该部分包括以下各项：

- **Page**（页面） — 图纸显示的各项设置，包括网格类型和颜色。
- **Terminal**（终端）— 终端的各设置项，包括颜色和字体。
- **Text**（文本） — 标签的各设置项，包括显示表达式以及颜色还是所有者行。
- **Wire**（连线） — 连线的各设置项，包括颜色、宽度和字体。另外还有一个用于弹性连接的设置项。使能该选项时，如果您移动了某个物体，将自动重绘连线。如果禁用了弹性连接，将不再重绘连线，并会断开连接。当移动各物体时，通过按下**[Space Bar]**可以分别禁用或使能弹性连接。

工作表模板：

通过该部分，您可以指定 PSoC Creator 寻找工作表模板文件的位置，以设计输入工具。更多有关创建工作表模板的信息，请参考工作表模板编辑器中介绍的内容。

使用 **Add** 按钮将更多路径添加到工作表模板的位置中；使用 **Remove** 按钮移除这些路径。

组件目录：

该部分包括以下组件项：

- **Show Hidden Components**（显示隐藏组件） — 勾选该复选框后将显示组件目录中所有隐藏的组件，包括基元。基元即基本组件，在目录的典型组件中，这些基元作为构建模块使用。
- **Enable Param Edit Views**（使能参数编辑视图） — 勾选该复选框后，通过右击 Configure 对话框中的不同选项卡，您可以查看 Expression View（表达式视图）。

Warning（警告）切换到表达式视图是一个高级特性。它需要全面的组件有效参数设置知识。通过使用该视图可以创建无效参数的组合。因此，如果您找不到有效的参数，您需要取消所有更改并重新启动该过程。

- **Remember Dialog Sizes**（记住对话框大小） — 勾选该复选框后，组件的 Configure 对话框将根据组件基础保存/恢复它们的大小。取消勾选该复选框框后，各对话框将以默认的大小打开。

  注意：如果已保存的大小超过了当前屏幕的尺寸，那么该大小将自动调整以适合屏幕显示。

- **Reset Sizes**（恢复尺寸） — 点击该按钮后，当前保存的所有对话框大小将复位到组件的默认大小。该操作无法复原。

组件安全性：

您可以添加或清除连接至第三方定制器的路径（可以在 PSoC Creator 中使用该定制器）。对于本部分不包含的第三方定制器，PSoC Creator 会生成错误。当您尝试打开外部项目时，PSoC Creator 将显示以下信息：

> *“项目 XXX 包含了带特殊代码的组件，可以在您的机器上运行该代码。您是否相信该项目的来源，并且是否想使用该项目组件。”*

如果选择“是”，则该项目的路径将被添加到相应设置。然后，每当您打开该特殊项目时，将不再有信息显示而且该项目将正常加载和编译。

如果您选择“否”，那么该项目将显示一个错误信息：

> *“组件 YYY 不可用。项目 XXX 的组件安全性被禁用。如果您相信该项目的来源，并希望允许该组件在设计过程中在计算机上运行代码，请依次选择* ***Tools > Options > Design Entry > Component Security****，然后将项目文件夹添加到批准列表内。”*

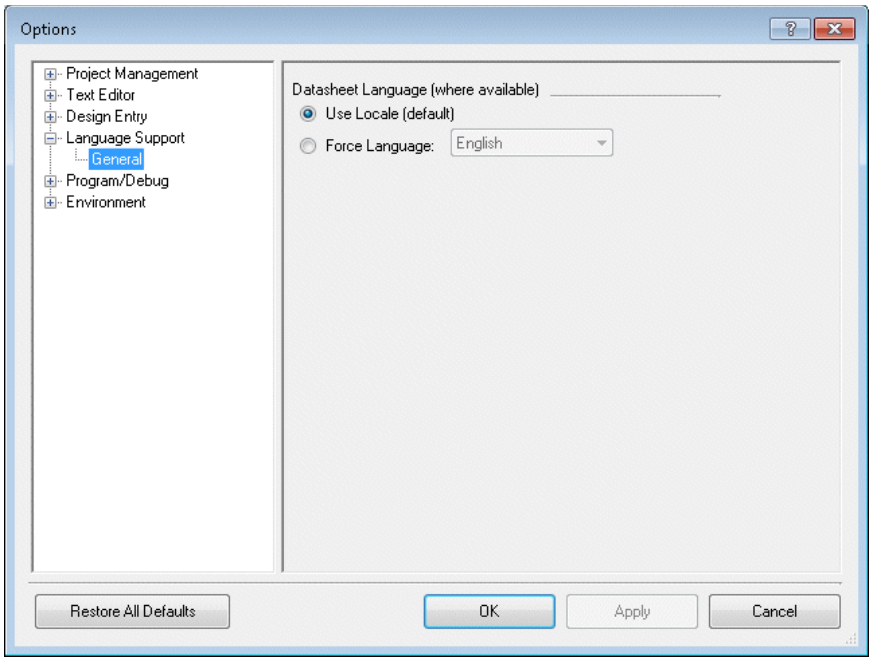
这样，每次您打开该项目都会显示该错误信息，直到您添加了路径为止。

另外，请参考：

- Options 对话框
- 使用设计输入工具
- 工作表模板编辑器
- 组件目录
- 配置对话框

语言支持选项

Options 对话框中的 Language Support（语言支持）项包含用于安装数据手册语言包的设置。



General（通用）选项：

General（默认）部分包含以下选项：

Datasheet Language（数据手册语言）

指定 **Use Locale**（使用语言环境）或 **Force Language**（强制语言）。如果安装了数据手册语言包，则 **Force Language**（强制语言）项将使能一个包含不同语言的下拉菜单。

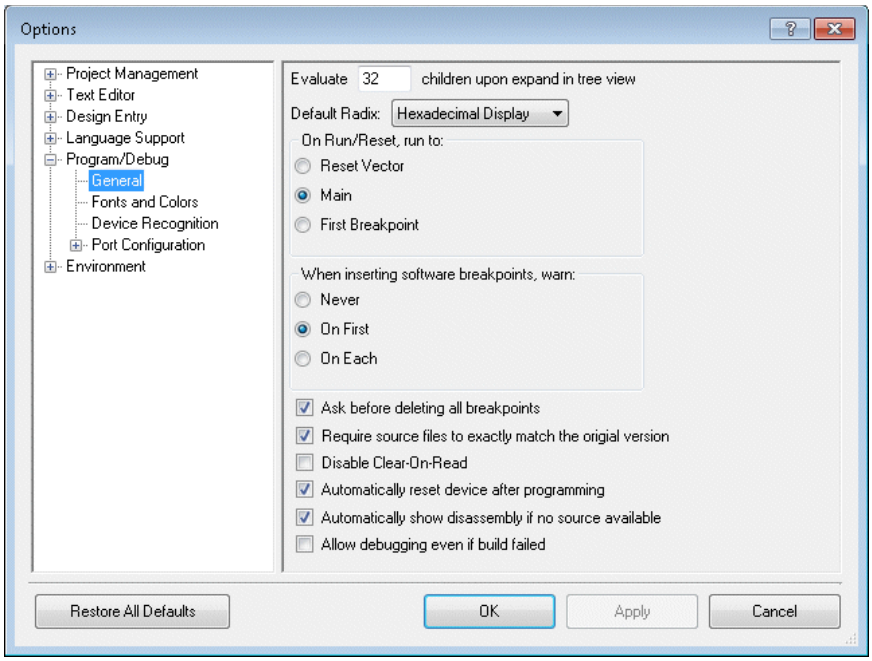
当打开数据手册时，如果特殊语言及版本不可用，将显示 Select Datasheet（选择数据手册）对话框，以选择合适的数据手册。

另外，请参考：

- Options 对话框
- 选择数据手册

Program/Debug（编程/调试）选项

Options 对话框中的 Program/Debug（编程/调试）部分包含了多个选项，用于配置器件、调试器、MiniProg3 和套件。



General（通用）：

在 General 目录下，您可以配置以下各项：

- **Evaluate**（评估） — 指定了子对象数量的文本框（即在变量视图中每次可以检索到的子对象数量）。
- **Default Radix**（默认基数） — 指定了各窗口中默认基数显示的下拉菜单。
- **On Run/Reset, run to**（运行/复位时，将转到） — 选择“转到”点（复位向量、Main 函数或第一个断点）的单选选项。
- **When inserting software breakpoints, warn** –（插入软件断点时出现警告信息） — 选择在插入软件断点时出现警告信息的单选选项。
- **Ask before deleting all breakpoints**（在清除所有断电前询问） — 该复选框指出了 PSoC Creator 是否在清除断点前询问您。
- **Require source files to exactly match the original version**（要求源文件与原始版本完全匹配） — 该复选框指出了源文件是否与原始版本相匹配。通过勾选该项，您能够在调试时编辑代码。

  注意：在您重新进行编译前，各更改内容将不会产生任何效果。
- **Disable Clear-On-Read**（禁用读取清除） — 通过该项，调试器能够读取 CLR 寄存器而不需要真正清除数据。
- **Automatically reset device after programming**（编程后自动复位器件） — 编程后自动进行复位，以执行新程序。
- **Automatically show disassembly if no source**（如果没有源代码，将自动显示 Disassembly 窗口） — 如果没有找到当前调试行的源代码，将自动打开 Disassembly 窗口。
- **Allow debugging even if build failed**（允许进行调试即使编译失败） — 如果编译失败，但前编译的输出仍然存在，则将出现一个对话框以启动旧编译的调试过程。

注意：当调试旧编译时，被调试的代码可能不与编辑器中的源代码相匹配。

Fonts and Colors（字体与颜色）：

在字体与颜色目录下，您可以调整 PSoC Creator 中不同调试器窗口上的字体格式和大小，以及各种颜色和属性。

- 在 **Show settings for** 下拉菜单中选择窗口或控制。请点击 **Use Defaults**，将设置回复为默认情况。
- 选择**字体**类型。 （以粗体字显示的各项表示固定宽度的字体。）
- 选择字体**大小**。
- 在 **Display items**（显示项）中，选择需要更改的项，并从 **Item foreground**（项目前景）**/Item background**（项目背景）下拉菜单中选择一个颜色或点击相应的 **Custom...**按钮来自定义颜色。
- 如果可用，请选择**字体格式**，然后将文本格式更改为粗体字、斜体字、删除线和/或下划线。

Device Recognition（器件识别）：

器件识别用于配置 PSoC Creator，以识别第三方器件。这样，Select Debug Target（选择调试目标）对话框便会列出与计算机相连的器件的正确信息。

请注意，当该配置允许 PSoC Creator 识别第三方器件时，不能将这些器件用于调试目的。器件配置的一种使用情况是用于配置指令寄存器和数据寄存器的大小。这些寄存器属于附加在 JTAG 链路中的第三方器件。

请注意，可以通过两种方式访问器件配置：通过 Options 对话框或通过右击 Select Debug Target（选择调试目标）对话框中的节点。

Port Configuration（端口配置）：

该项用于配置相应端口。

MiniProg3：

- **Active Protocol**（有效协议） — 选择用于与目标器件通信的协议。
- **Clock Speed**（时钟速度） — 选择 MiniProg3 尝试与目标器件通信的频率。该速度不能超过目标器件的总线时钟速度的 1/3。另外，虽然更低的速度也可用于调试，但该速度要求最小为 1 MHz，这样才能用于编程器件。6 MHz 是 PSoC 器件的最高建议时钟频率。
- **Power**（电源） — 指定 MiniProg3 提供给目标器件的电压量或指定该电源是否来自外部源。
- **Acquire Mode**（获取模式） — 选择用于复位器件的机制，以便能够进行调试。
- **Connector**（连接器） — 在 MiniProg3 上，选择用于发送数据的连接器。

FX2LP-SWD （First Touch 套件 3 / First Touch 套件 5 / DVK3 / DVK5）：

- **Active Protocol**（有效协议） — 显示用于与目标器件通信的协议。
- **Power**（电源） — 指定 MiniProg3 提供给目标器件的电压量或指定该电源是否来自外部源。
- **Acquire Mode**（获取模式） — 选择用于复位器件的机制，以便可以进行调试。

Touch Tuning 桥接器：

- **Active Protocol**（有效协议） — 显示用于与目标器件通信的协议。

- **Power**（电源） — 指定电压来自内部源还是外部源。
- **Acquire Mode**（获取模式） — 选择用于复位器件的机制，以便能够进行调试。

DVKProg1:

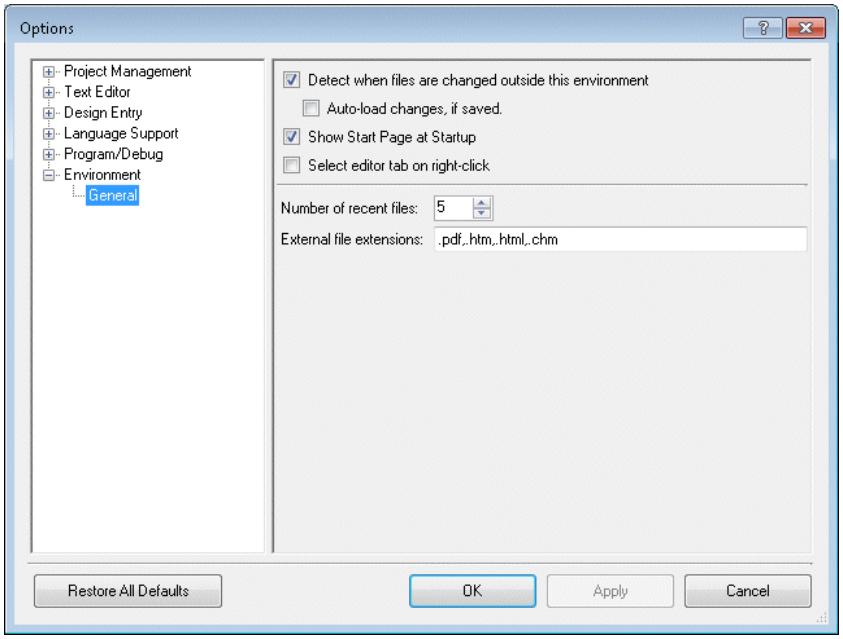
- **Power**（电源） — 指定电压来自内部源还是外部源。
- **Acquire Mode**（获取模式） — 选择用于复位器件的机制，以便能够进行调试。

另外，请参考：

- Options 对话框
- 如何使用调试器
- 选择调试目标
- 器件配置
- 调试器窗口

环境选项

Options 对话框中的 Environment 部分包含多项设置，用于指定内容在 PSoC Creator 中如何显示。



通用环境选项：

General（默认）部分包含以下环境选项：

Detect when files are changed outside this environment（检测各文件在该环境外更改）

该复选框指定了 PSoC Creator 是否检测在 PSoC Creator 外进行的更改。如果勾选该项，您也可以指定是否勾选 **Auto-load changes, if saved**（如果已保存，自动加载更改）。

Show Start Page at Startup（启动时显示起始页）

通过该复选框，您能够在启动时显示或隐藏 Start page（起始页）。如果您选择启动时不显示 Start page（起始页），可以从 **View** 菜单打开它。

Select Editor Tab on Right-Click（通过右击选择编辑器选项卡）

如果勾选该项，右击文档窗口选项卡时，将显示该文档以及一个菜单。如果未选择该项，右击时仅显示一个菜单。

Number of Recent Files（最近的文件数量）

通过该文本框，您能够指定文件的数量，这些文件显示在 **File** 菜单下的 Recent Files（最近文件）项内。

External File Extensions（外部文件扩展名）

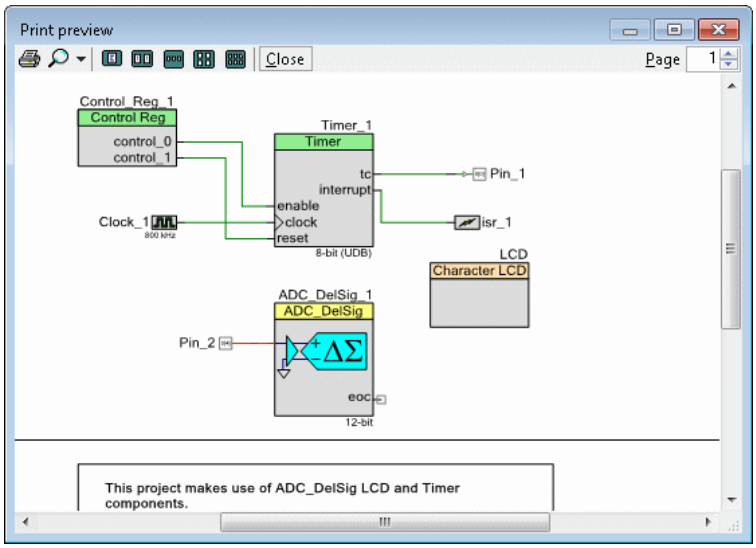
在该字段内输入文件扩展名，以便可以使用其他应用（而不是 PSoC Creator）打开这些文件。使用逗号分开这些扩展名。

另外，请参考：

- Options 对话框
- 文件菜单
- 视图菜单
- 组件更新
- 窗口类型

Print Preview（打印预览）

通过 Print Preview 对话框，您可以在打印某个文件之前预览该文件。



您可以预览的 PSoC Creator 文档类型包括：原理图、符号、源代码文件和某些设计范围资源文件。

要打开 Print Preview 对话框：

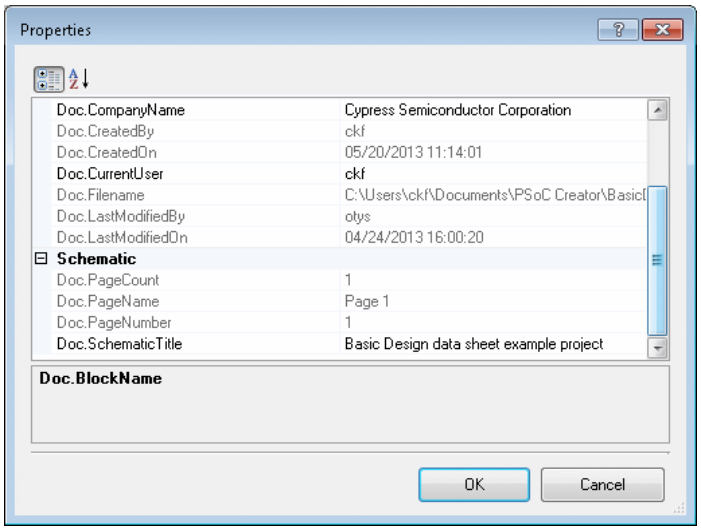
点击 **File** 菜单，并选择 **Print Preview**。

要使用 Print Preview 对话框：

- 点击 **Print** 打印该文档。
- 使用 Zoom 下拉菜单，以不同大小查看该预览。
- 对于有多页的文件，请使用 Various Page View（查看多页）选项。使用 **Page** 框转到指定的一页视图。
- 点击 **Close** 关闭 Print Preview 对话框。

Properties（属性）

Properties 对话框提供了其他窗口中已选项目（如工作区浏览器、原理图编辑器、符号编辑器，等）的相关信息。



在多种情况下，您可以更改已选项目的不同值/属性。除了项目级属性外，项目中的每个文件都有各自的属性。通过这些属性，您能够将各种选项设置为文件级别。

根据您打开该对话框的方式，它会包含不同的属性目录，例如：

- **General**（通用）— 文件类型、路径、名称
- **Document**（文档） — 创建日期、当前用户
- **Schematic**（原理图）— 页面计数、标题
- **Symbol**（符号） — 目录放置、摘要

要打开该对话框：

右击 PSoC Creator 中的某个项目（例如，设计图纸或文件），然后选择 **Properties**。

另外，请参考：

- 文本的使用

- 定义目录放置

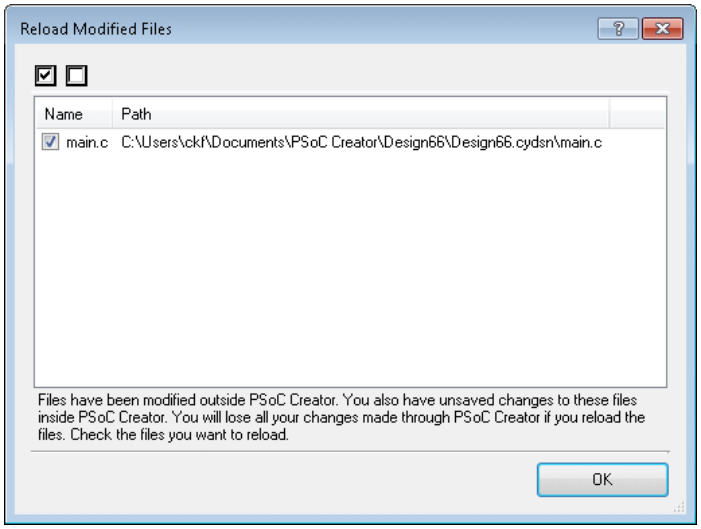
重新加载文件

如果您在 PSoC Creator 应用程序范围外对 PSoC Creator 文件进行操作 — 当它恢复聚焦时 — 它将检查打开文件在磁盘上是否被更改。如果有文件被更改，则 PSoC Creator 将显示两个提示以访问应该重新加载的文件。如果两种文档类型都存在，将显示两个提示。

要想禁用 PSoC Creator 尝试重新加载在该应用范围外修改的文件，您需要取消选定环境选项中的 **Detect when files are changed outside this environment**（检测在该环境外更改的文件）项。

重新加载已修改的文件：

如果需要重新加载的文件拥有在 PSoC Creator 中进行的未保存更改，将显示 Reload Modified Files（重新加载已修改的文件）对话框。

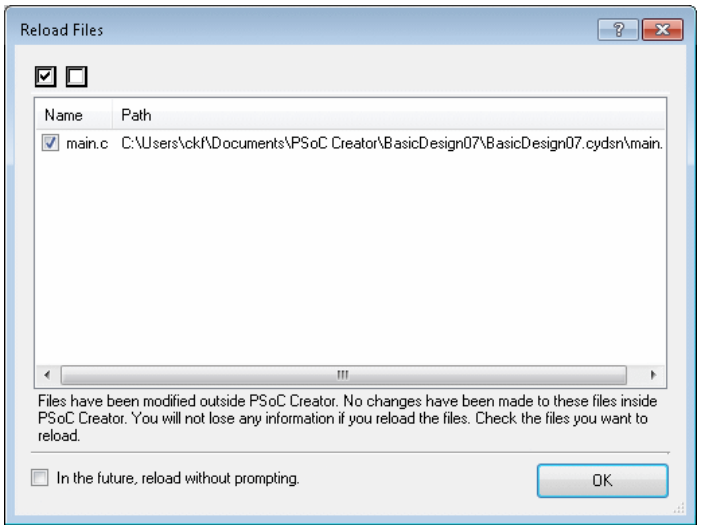


如果重新加载了任意的修改文件，则所有未保存的更改将被丢失。

- **Check All**（勾选全部） — 使用该选项可勾选所有文件，表示需要重新加载所有文件。
- **Uncheck All**（取消勾选全部） — 使用该选项可取消勾选所有文件，表示没有文件需要重新加载。

重新加载已修改的文件：

如果需要重新加载的文件拥有在 PSoC Creator 中进行的未保存更改，将显示 Reload Modified Files（重新加载已修改的文件）对话框。



在该对话框中，会自动重新加载未修改的文件，另外可设置没有提示的选项。如果该项被设置为真，将不再显示该对话框。

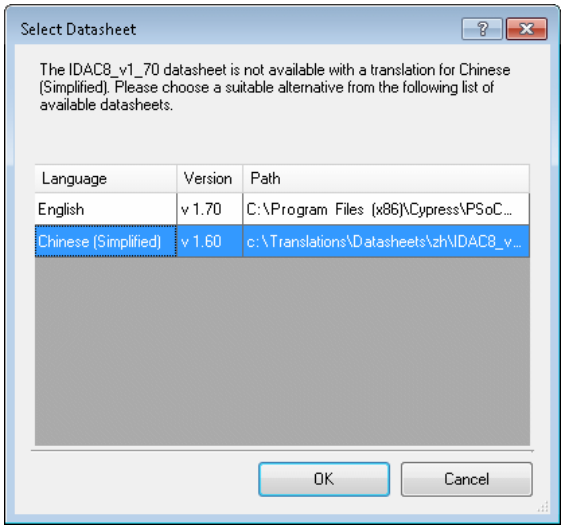
注意：可以从 Environment（环境）选项：**Auto load changes, if saved**（自动加载更改，若被保存）更改该项。

另外，请参考：

- 更改文件
- 环境选项

选择数据手册

Select Datasheet（选择数据手册）对话框用于选择需要打开的合适的数据手册。当您尝试打开某个数据手册时，将显示该对话框，但该数据手册的特定语言和版本不可用。



要想使用该对话框，请执行下述操作：

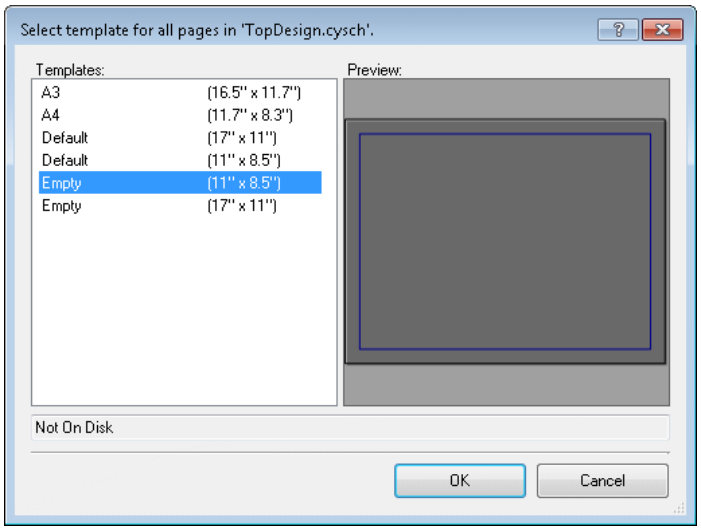
选择需要打开的合适的数据手册并点击 **OK**。

另外，请参考：

- 语言支持选项

选择工作表模板

通过 Select Sheet Template（选择工作表模板）对话框，可以选择您设计的工作表模板。



要打开该对话框，请执行下述操作：

- 创建新项目时：
  - 选择 New Project 对话框上的 **Advanced** [+]按钮。
  - 然后，下拉 Select Template（选择模板）菜单，并选择<Launch Sheet Template Dialog>（启动工作表模板对话框）。
- 对于现有项目，请右击您设计图纸并选择 **Change Template**（更改模板）。

Templates（模板）：

该项列出了可选用于您设计的工作表模板。点击某个模板来选择它。

Preview（预览）：

该项列出了已选模板的预览。

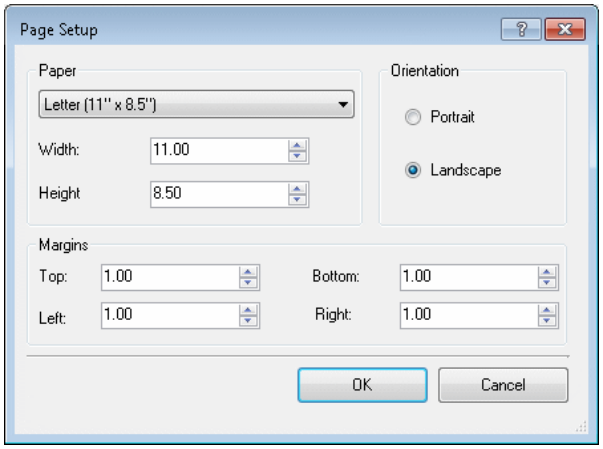
另外，请参考：

- 创建新项目
- 原理图编辑器

- 工作表模板编辑器

工作表模板的页面设置

Sheet Template Page Setup （工作表模板的页面设置）对话框用于指定您工作表模板的各种属性。该对话框与用于打印目的的 Windows Page Setup 对话框不同。



只有在您创建新的数据手册模板时，才会显示该对话框。更多有关信息，请参考 Sheet Template Editor（工作表模板编辑器）。

Paper（纸张）：

从下拉菜单中选择纸张大小，或指定自定义的 **Width**（宽度）和 **Height**（高度）测量（英寸）。

Orientation（方向）：

指定 **Portrait**（纵向）或 **Landscape**（横向）。

Margins（页边距）：

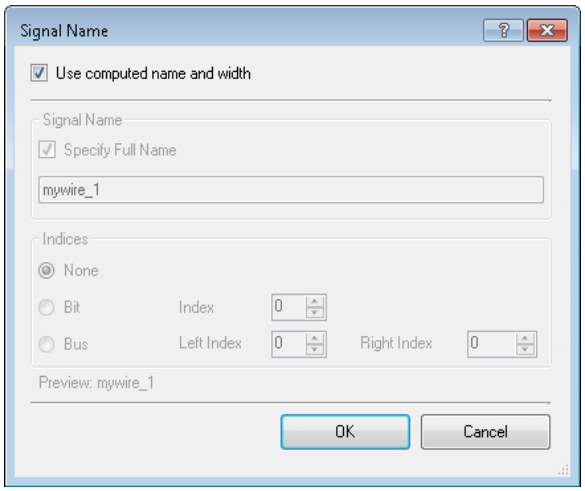
指定 **Top**（上）、**Bottom**（下）、**Left**（左）和 **Right**（右）页边距（英寸）。

另外，请参考：

- 工作表模板编辑器

信号名称

Signal Name（信号名称）对话框用于给信号命名。



一个信号的有效名称如下所示：

- mywire_1 — 仅基本名
- mywire_1[0] — 基本名和位指数
- mywire_1[1:0] — 基本名和总线指数
- [0] — 仅位指数
- [1:0] — 仅总线指数

要打开该对话框，请执行下述操作：

可以通过两种方式打开该对话框：

- 对于无标签的连线，请双击该连线或右击它并选择 **Edit Name and Width**（编辑名称和宽度）。
- 对于带标签的连线，请双击该连线的标签。

Use Computed Name and Width（使用计算名称和宽度）：

勾选该复选框后，PSoC Creator 会自动指定名称和宽度；如果取消勾选，则需要手动执行该操作。

Specify Full Name（指定全名）：

1. 为了指定一个名称，您必须先取消选择 **Use Computed Name and Width**（使用计算名称和宽度）复选框。
2. 然后，勾选 **Specify Full Name**（指定全名）复选框来使能相应的文本框。
3. 在该文本框中输入信号的基本名称。

Indices（指数）：

通过该字段可以选择以下指数选项中的某一项。当您取消选择 **Use Computed Name and Width**（使用计算名称和宽度）复选框时，将可以使用该项。

- **None**（无） — 如果未指定基本名称（例如，my_wire），则勾选该项。如果您取消选择 **Specify Full Name**（指定全名）复选框，**None**（无）选项将被禁用。

- **Bit**（位） — 勾选该项后可以为该信号设置位指数（例如，[0]）。
- **Bus**（总线） — 勾选该项后可以为该信号设置总线指数（例如，[1:0]）。

注意：最大总线宽度为 1024。

Preview（预览）：

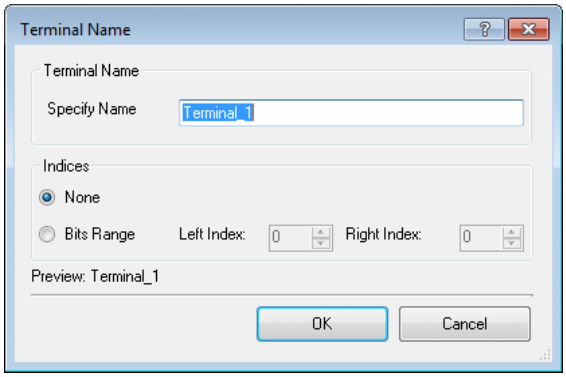
通过该字段，您可以了解信号名称是如何显示在原理图上的。

另外，请参考：

- 连线标签和名称
- 连线的使用
- 绘制总线

Terminal Name（终端名称）

Terminal Name（终端名称）对话框用于给终端命名。



一个终端的有效名称如下所示：

- 只有基本名称（Terminal_1） — 一个终端必须拥有以字母开头的基本名称。
- 带总线指数的基本名称（Terminal_1[1:0]） — 若需要，您可以指定左和右指数。

要打开该对话框，请执行下述操作：

可以通过两种方式打开该对话框：

- 将新终端拖放到您的设计上，或
- 双击该标签或终端。

Specify Name（指定名称）：

Specify Name（指定名称）字段用于输入信号的基本名称。

Indices（指数）：

通过该字段可以选择以下指数选项中的某一个：

- **None**（无） — 如果基本名称没有指数（例如，Terminal_1），则勾选该项。
- **Bits Range**（位范围） — 勾选该项后可以为该信号设置总线指数（例如，[1:0]）。

注意：最大总线宽度为 1024。

Preview（预览）：

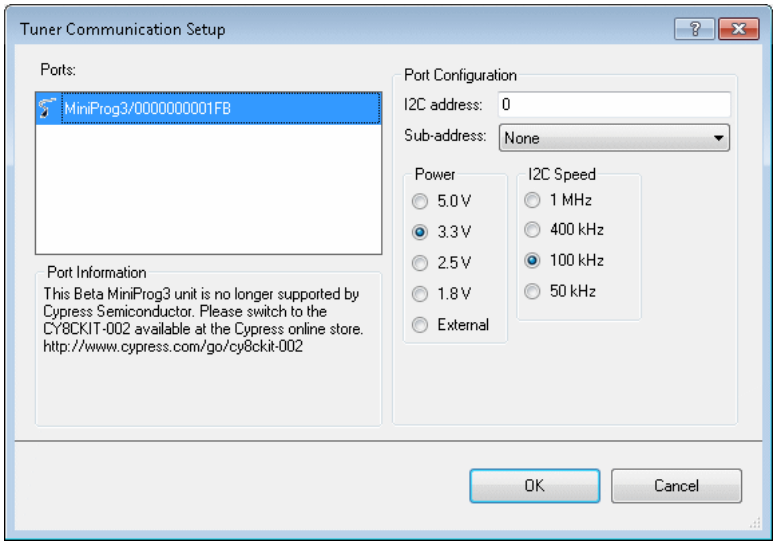
通过该字段，您可以了解信号名称是如何显示在原理图上的。

另外，请参考：

- 信号名称
- 原理图终端的使用
- 组件终端的使用

调试器通信设置

通过 Tuner Communication Setup（调试器通信设置）对话框，您可以配置各项，以实现从 PSoC 器件中回读参数。通过该端口可以选择通信端口和各属性。



目前，调试器仅支持通过 I²C 进行的通信。通过使用 EZ I²C 组件或标准的 I²C 组件实现该操作。

要打开该对话框，请执行下述操作：

从相应的调试器应用中打开该对话框。

目前，只有调试器应用拥有 CapSense® CSD 组件。请参考组件目录中的组件数据手册。

接口说明：

Ports（端口）

与计算机相连的所有端口可用于与可调式组件通信。根据已选项目，而端口配置可以使用不同的选项。

端口配置

该部分允许配置接口的特定选项，为了能与组件通信。需要确保组件和调试器的配置相同。

对于 I²C 信息，需要四种信息：

- 端口需要给器件提供电压（若有）。
- 数据通过时钟控制输入到目标器件的速度。
- 将要进行通信的从设备地址。
- 辅助地址的大小，用于指出需要读取的数据块。

Port Information（端口信息）

该部分显示了有关当前选择端口的信息。

另外，请参考：

- CapSense® CSD 组件数据手册（在组件目录中提供）

自定义框架

自定义框架

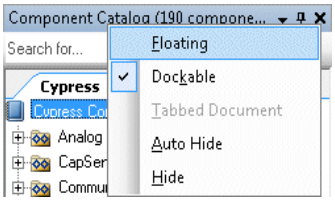
该部分包含许多“如何执行”主题，允许您自定义 PSoC Creator 框架：

- 如何使工具窗口浮动
- 使工具窗口停驻
- 使用选项卡式文件
- 如何移动工具窗口
- 自动隐藏工具窗口

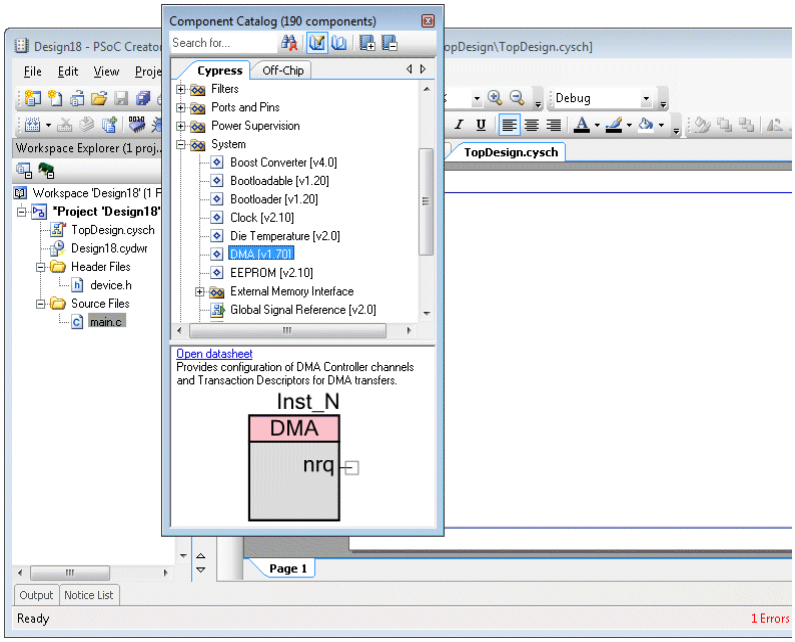
如何使工具窗口浮动

在该框架外，您可以浮动各种 PSoC Creator 窗口。浮动即为窗口不与框架相连。

1. 选择您想浮动的窗口并从 Window mode（窗口模式）菜单中选择 **Floating**。



请注意移动到 PSoC Creator 框架外的窗口。



2. 将该窗口从当前位置移动到您屏幕的任何位置。

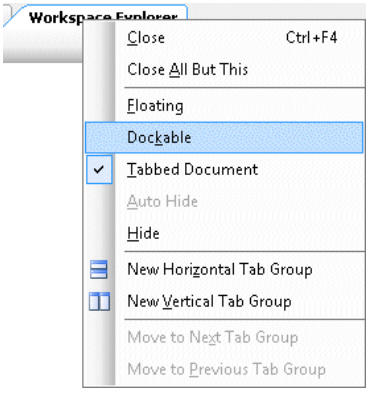
另外，请参考：

- 工具窗口
- 框架说明
- 如何移动工具窗口
- 如何使工具窗口停靠

如何使工具窗口停靠

您可以将各窗口停靠在 PSoC Creator 框架中不同的位置内。停靠即为将窗口连接至框架。

1. 选择您想移动的窗口并从 Window mode（窗口模式）菜单中选择 **Dockable**。



2. 将该窗口从当前位置拖动到框架中其他位置上。

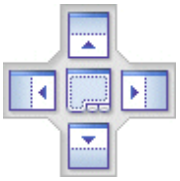
   当您拖动该窗口时，请注意停靠指南显示在不同的位置内。

3. 当该窗口位于您想停靠它的位置时，则将鼠标放置在该指南的相应部分。

   该窗口的外形将显示在设计区域内。

4. 放开鼠标便将该窗口停靠在已指定的位置。

可以将工具窗口停靠在框架边缘或其他现有边缘以及其他窗口中的选项卡内。下表显示的是不同停靠指南以及它们的含义：

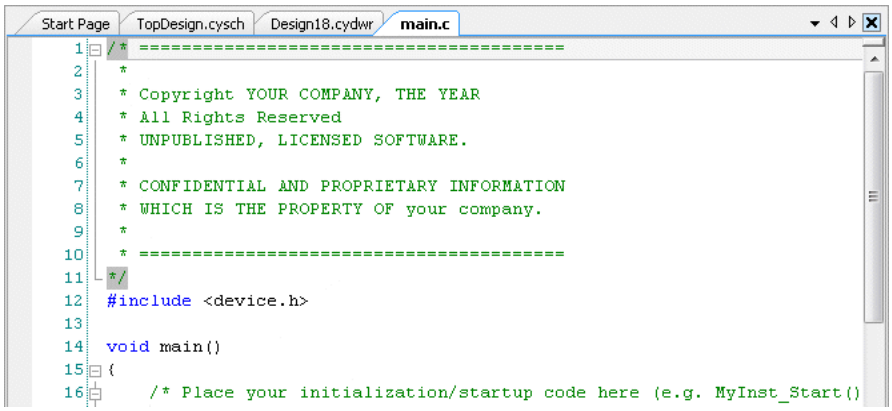
	停靠在左边缘。
	停靠在右边缘。
	停靠在上边缘。
	停靠在下边缘。
	停靠在左边缘、右边缘、上边缘、下边缘或作为选项卡停靠在窗口中。

另外，请参考：

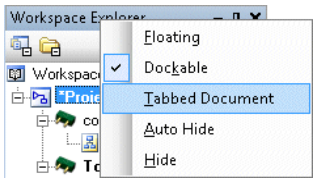
- 工具窗口
- 框架说明
- 如何移动工具窗口
- 如何使工具窗口浮动

如何使用选项卡式文件

如窗口类型中所述，可将某些窗口停靠在框架内或作为选项卡式文档使用。选项卡式文档并行排列在工具区域内：

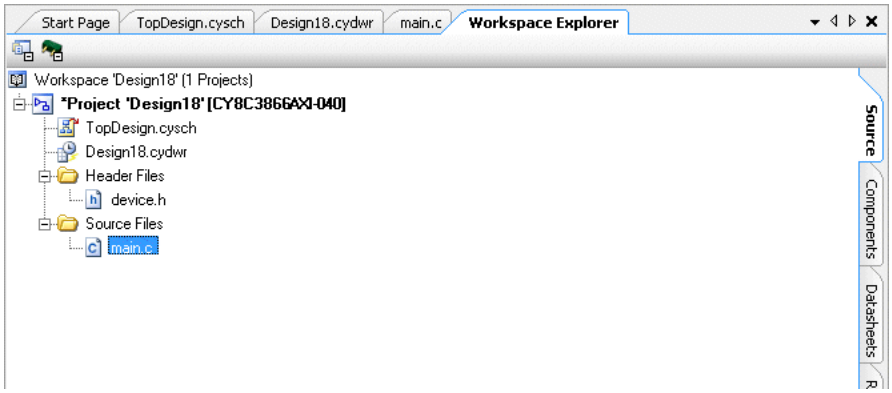


要想将一个窗口更改为选项卡式文档（例如，工作区浏览器），请右击该窗口标头并选择**选项卡式文档**。



注意：如果不能将窗口更改为选项卡式文档（如组件目录），菜单项将被禁用（以灰色显示）。

选择该菜单项后，该窗口将作为选项卡式文档并与其他打开文档一起显示。



另外，请参考：

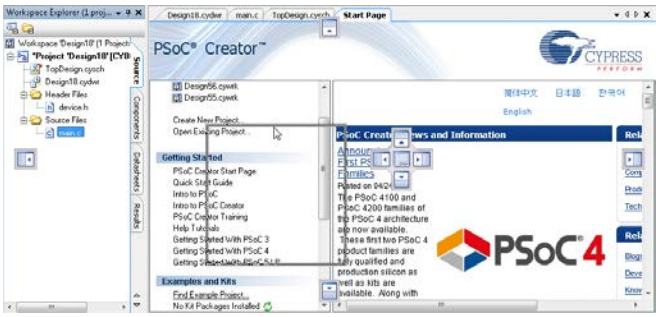
- 窗口类型
- 工具窗口
- 文档窗口

如何移动工具窗口

您可以将各种窗口移动到 PSoC Creator 框架中不同的位置，甚至可以移动到该框架的范围外。

1. 点击您想移动的窗口的标题栏。
2. 将该窗口从当前位置拖动到其他位置。

   当您拖动该窗口时，请注意停靠指南显示在不同的位置内。



3. 将该窗口移动到所需要的位置上，并释放鼠标。

另外，请参考：

- 如何使工具窗口停靠
- 如何使工具窗口浮动
- 框架说明
- 工具窗口

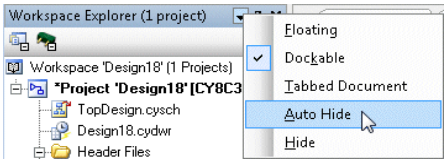
如何自动隐藏工具窗口：

自动隐藏性能允许您通过最小化框架边缘上的工作窗口（当不使用时）查看 PSoC Creator 框架的更多部分。

如何打开自动隐藏性能：

请使用以下方法打开自动隐藏性能：

- 从工具窗口工具栏中选择 **Auto-Hide**（自动隐藏）或右击右键菜单。

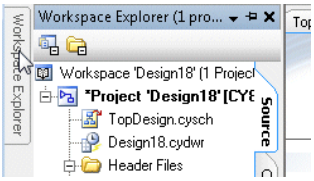


- 点击 Auto-Hide Pushpin 图标。

该窗口自动滑动到框架的边缘上。它的名称显示在该框架边缘上的选项卡内。

如何显示隐藏窗口：

将光标放置在选项卡上，以将工具窗口滑动到该位置内。



如何关闭自动隐藏：

请使用以下方法关闭自动隐藏性能：

- 从工具窗口工具栏中选择 **Dockable**（可停靠）或右击右键菜单，或
- 再次点击 **Auto-Hide Pushpin** 图标。

另外，请参考：

- 框架说明
- 工具窗口
- 如何移动工具窗口
- 如何使工具窗口停靠
- 如何使工具窗口浮动

5 使用设计输入工具

通过 PSoC Creator 设计输入工具，您可以通过使用抽象符号创建设计，这些工具集中到系统内，而不是低级平器件的详细内容。

该部分包括以下目录：

- 原理图编辑器 — 用于创建设计的主要工具
- 代码编辑器 — 用于编辑源代码的工具
- 设计范围资源 — 用于配置整个设计设置的工具
- 符号编辑器 — 用于创建组件的工具
- UDB 编辑器 — 用于配置 UDB 实现的工具
- 其他工具
  - 原理图宏编辑器
  - 工作表模板编辑器
  - 格式形状

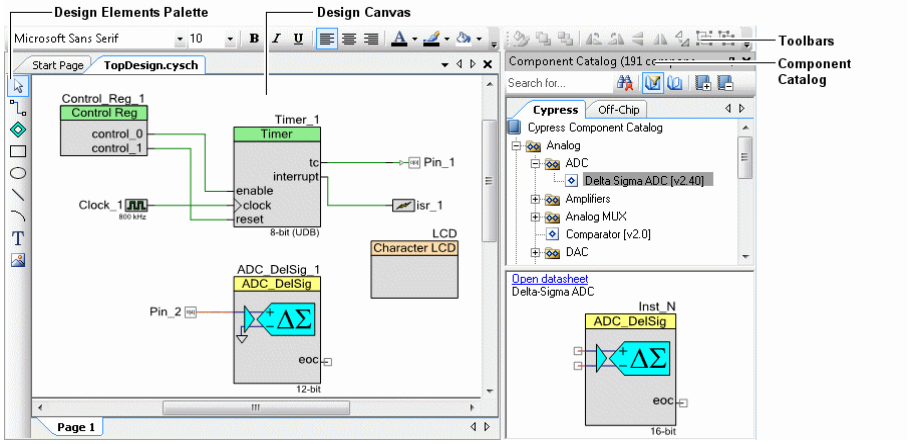
另外，这两个编辑器都包括下面各主题：

- 普通工具栏
- 设计元素控制板
- 文本的使用
- 线路的使用
- 形状的使用
- 设计输入的保留字

原理图编辑器

原理图编辑器

通过原理图编辑器您可以创建与调试设计原理图和符号实现。



原理图编辑器的主要组件包括：

- 设计图纸 — 您绘制设计的图纸
- 设计元素控制板
- 普通设计输入工具栏 — 设计输入工具常用的各条指令
- 右键菜单 — 通过右击即可使用的各指令
- 组件目录 — 用于您原理图的组件库

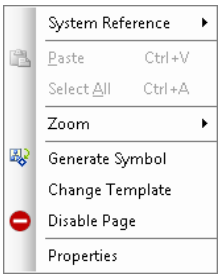
本部分介绍了与原理图编辑器使用有个的各个主题：

- 创建新的原理图
- 配置组件参数
- 连线的使用
- 弹性连接
- 绘制总线
- 连线标签和名称
- 使用多页和连接器
- 禁用/使能原理图页
- 原理图终端的使用

原理图编辑器的右键菜单指令

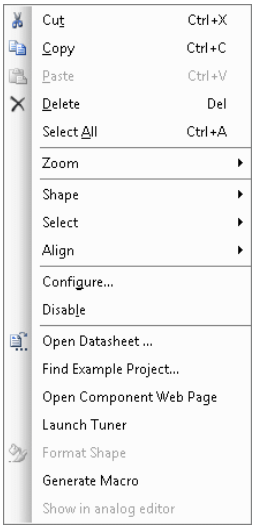
原理图编辑器的右键菜单（右击后即可使用）包含各种指令。根据您右击的地方 — 图纸或组件/元素，可用的指令会存在差异。下面是可用的指令：

在图纸上：



- **System Reference**（系统参考） — 通过该项可以访问磁盘上最新的系统参考指南，以及其他版本或文档的翻译版本（若有）。
- **Paste**（粘贴） — 与 Standard Toolbar（标准工具栏）中的指令相同。
- **Select All**（全部选择） — 选择图纸上的所有对象。
- **Zoom**（缩放）— 与 View 菜单的缩放指令相同。
- **Generate Symbol**（通用符号） — 创建一个拥有自动生成默认形状的新符号；该符号自动包含了各终端，用于代表源原理图的原理图终端。
- **Change Template**（更改模板） — 打开 Select Sheet Template（选定工作表模板）对话框，以选择/更改您设计使用的工作表模板。
- **Disable Page**（禁用页面） — 禁用所选页面。请参见禁用/使能原理图页。
- **Properties**（属性） — 打开 Properties 对话框。

在选定的对象上：



- **Cut, Copy, Paste, Delete**（剪切、复制、粘贴、删除）— 与标准工作栏的指令相同。
- **Select All**（全部选择） — 选择图纸上的所有对象。
- **Zoom**（缩放）— 与 View 菜单的缩放指令相同。
- **Shape**（形状）— 与通用的设计输入工作栏的造型指令相同。
- **Select**（选择）— 当两个或更多的对象被绘制在彼此的顶部上时，通过该指令可以选择具体的对象。
- **Align**（对齐）— 选中两个或更多对象时，通过该指令您可以调整所选的形状：左、右中心、顶部、中间和底部。
- **Configure**（配置）— 打开配置组件参数对话框，以编辑组件实例的各参数。
- **Enable/Disable**（使能/禁用） — 使能或禁用所选定的组件。如果禁用某个组件，那么 PSoC Creator 会忽略它；这样会将 Built-In 选项卡下 Configure 对话框中的 CY_REMOVE 标志设置为 true。
- **Open Datasheet**（打开数据手册）— 打开选定组件的数据手册。
- **Find Example Project**（查找示例项目）— 打开选定组件的 Find Example Project 对话框。
- **Open Component Web Page**（打开组件网页）— 如果该指令可用，它可打开组件的网页，从而您可以使用各种语言来查阅数据手册。
- **Launch Tuner**（启动调谐器）— 如果选定组件有一个调谐器应用，则该指令会启动调谐器。

- **Format Shape**（格式形状）— 打开 Format Shape 对话框以更改选中形状的各种特性。
- **Generate Macro**（生成宏）— 通过选定的元件创建一个原理图宏。
- **Show in analog editor**（显示在模拟编辑器中）— 在设计范围资源的模拟路由编辑器中显示选定的组件。

另外，请参考：

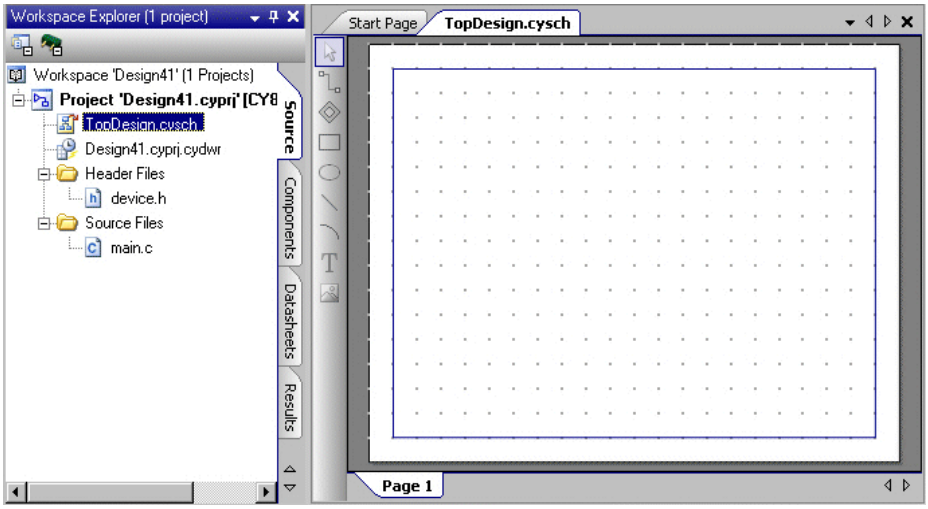
- 设计元素控制板
- 标准工具栏
- 设计输入的通用工作栏

创建新的原理图

根据您在尝试实现的原理图内容，创建新原理图的过程会存在差异。在一个项目范围内，原理图可能是顶层设计的原理图，也可能是一个组件实现原理图。在每个具体情况下，该过程会存在差异。

顶层设计原理图：

如果您创建新的设计项目，将在设计中创建顶层设计原理图以及初始文件的其余部分。该原理图时顶层设计文档，其中您可以通过使用图形组件和连接来表达您的设计。有关创建设计项目的更多信息，请参考我的第一个设计“Hello World Blinky”和创建新项目部分。

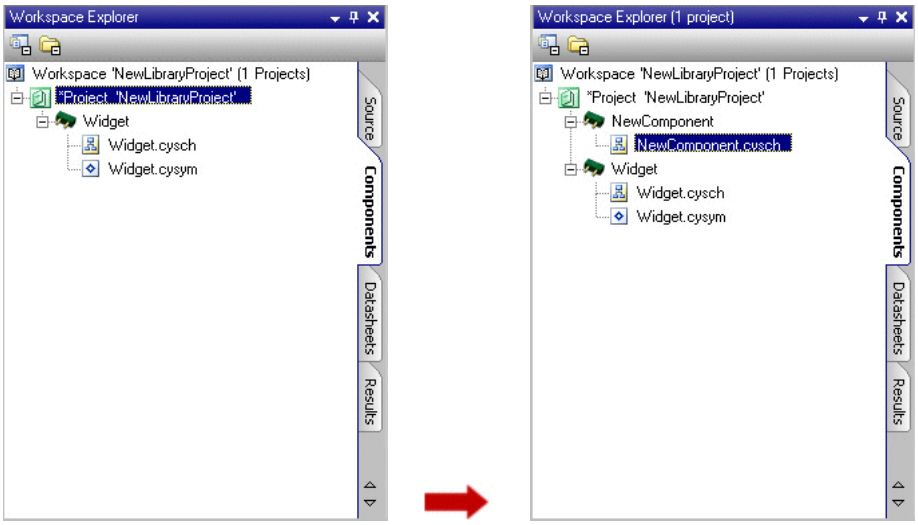


组件实现原理图：

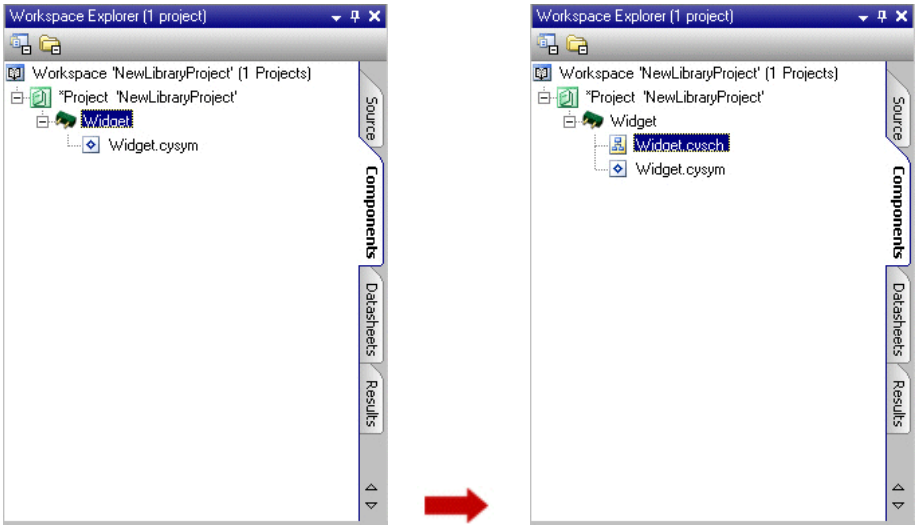
如果您正在使用一个库项目和组件，原理图可能是组件的实现。更多有关创建组件的信息，请参阅组件创建指南。

您必须向组件内手动添加原理图。您可以添加项目级别或组件级别的原理图。无论是哪种级别，您都需要添加原理图类型的组件项。更多有关信息，请参考添加组件项部分的内容。

- 当添加**项目级别的原理图**时，您需要创建新的原理图类组件项：



- 当您添加**组件级别的原理图**时，也是正在将新的原理图添加到现有的组件中：



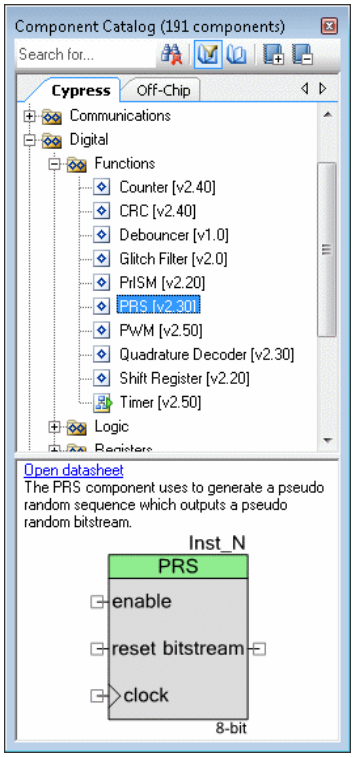
注意：每个组件可能只会有一个顶层（通用器件）原理图。您的组件中每个架构、系列和器件级别也可能只有一个原理图。

另外，请参考：

- 创建新项目
- 组件创建指南
- 添加组件项
- 原理图编辑器
- 工作区浏览器

组件目录

组件目录是一个原理图编辑器工具窗口，该窗口包含了设计中各可使用的组件集。各组件被分成各个类别。每个类别包含一个组件树。



注意：在目录中，组件的位置和序列是通过它们不同的属性决定的，而不一定反映其中它们被存储的实际库。另外，并不是所有组件都适用于每个器件。

下面各节介绍了以下组件：

- **Open Datasheet**（打开数据手册）— 打开选定组件的数据手册。
- **Description**（说明） — 组件的简短摘要。
- **Component Preview**（组件预览）— 选定组件符号的预览。

工具栏：

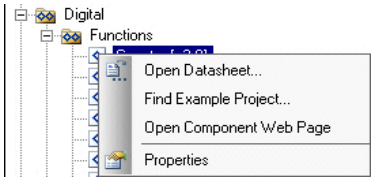
组建目录包含了一个工具集，具体如下：

- **Search**（搜索）— 查找与输入到搜索字段中的文体相匹配的组件。
- **Clear Results**（删除结果）— 清除搜索结果。
- **Show All Versions**（显示所有版本）— 显示组件的所有版本及其编号。如果只有一个版本，不会显示任何编号。
- **Show Latest Versions**（显示最新版本）— 只显示组件的最新版本及其编号。如果只有一个版本，则不显示任何编号。

- **Expand All**（扩展全部）— 扩展所有节点，以显示所有可见组件。
- **Collapse All**（折叠全部）— 折叠所有节点，只显示类别文件夹。

右键菜单指令：

如果右键单击树视图中的某个组件上，您可以访问以下指令：



- **Open Datasheet**（打开数据手册）— 通过选择该选项，可以打开组件的数据手册。
- **Find Example Project**（查找示例项目）— 通过选择该选项，可以打开 Find Example Project，然后打开所选组件特定的示例项目。
- **Open Component Web Page**（打开组件网页）— 如果该选项可用，可以通过它打开组件的网页，这样您便能够使用各种语言来查阅数据手册。
- **Properties**（属性）— 选择该选项来查看组件的基本属性。属性包括组件库的路径。

查看组件数据手册：

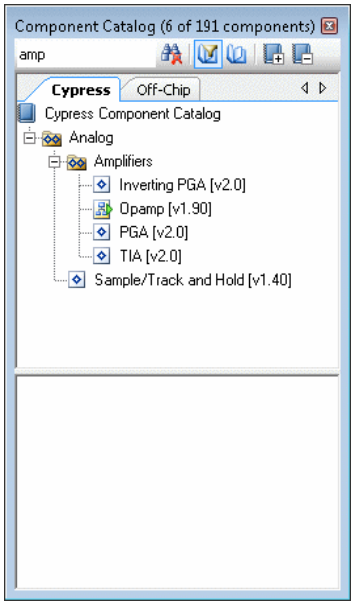
请单击 **Open Datasheet**（打开数据手册）链接，或从右键菜单选择选项。这样可以打开组件的数据手册。

将组件添加到原理图中，需要进行下面的操作：

请单击目录中的某个组件，并将它拖放到原理图编辑器画布内。

搜索某个组件：

请在搜索字段中键入文本：开始进入字符时，会按照您所输入的内容过滤出可用的组件。



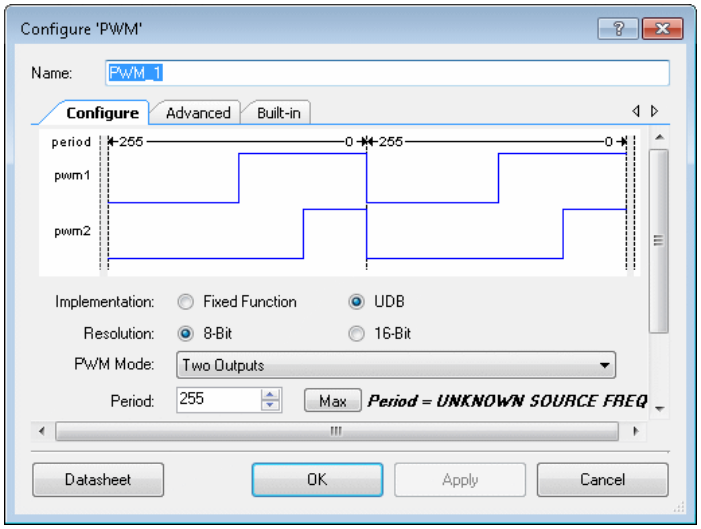
单击 **Clear Results**（删除结果）以清除搜索结果，并恢复组件目录以查看所有组件。

另外，请参考：

- 原理图编辑器
- 工具窗口
- 组件/实例
- 查找示例项目

配置组件参数

当将某个组件实例拖放到原理图上时，您可以通过 Configure 对话框来配置它。



注意：该帮助主题是通用的。很多组件都有自定义参数配置的用户界面。具体的信息，请参考组件数据手册。

要打开该对话框，请执行下面的操作：

1. 双击某个组件实例，或右键单击组件实例上并选择 **Configure**（配置）。
2. 编辑适当的参数，然后点击 **OK**。

如何能打开组件数据手册：

请单击 **Datasheet**（数据手册）按键。

如果组件有其数据手册，会在单独的窗口中打开。

如何重名组件实例：

默认情况下，组件实例的名称为“INSTANCE_1”，其中 INSTANCE 是组件名，如 PGA、计数器等等。

在 **Name** 字段中，请输入您需要的实例名称。

注意：一旦生成了组件的源代码，在后续代码生成的更新中，将忽略名称更改中大小写的区别。欲了解更多有关生成代码的信息，请参考构建 PSoC Creator 项目和生成文件部分。

内置参数：

每个组件均包含了一个选项卡带有下面各内置参数：

- CY_MAJOR_VERSION — 该参数显示了组件的主要版本号。
- CY_MINOR_VERSION — 该参数显示了组件的次要版本号。
- CY_REMOVE — 通过该参数，可以禁用组件，并在构建过程中从生成的网表移除实例。默认值为“假”。
- CY_SUPPRESS_API_GEN — 该参数用于防止 PSoC Creator 为具体的实例生成 API。默认值为“假”。
- CY_VERSION — 该参数显示了 PSoC Creator 组件的版本和构建信息。

更多有关内置参数的信息，请参考组件创建指南。

另外，请参考：

- 组件目录
- 原理图编辑器

连线的用法

通过使用原理图编辑器，您可以绘制各条导线，用以将组件连接到导线。本部分包含了在使用导线工作时可采用的某些技术：

- 绘制导线
- 连接到终端
- 绘制多点的导线
- 连接到另一条导线
- 选择导线/网格

另请参见导线标签和名称和绘制总线。

绘制导线：

1. 请点击 **Draw Wire**（绘制导线）工具。 
2. 请点击设计画布上并将它拖放到所需的位置。
3. 左键单击以便按其他方向连续绘制导线。
4. 双击以完成导线绘制。

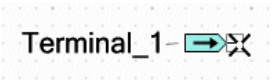
注意：当导线连接到数字组件或不连接到任何组件时，原理图编辑器将绘制数字（绿色）导线。要想绘制出模拟（红色）导线，您必须将其连接到模拟组件上。

要连接到某个终端：

1. 若想绘制导线，先要在理图上创建一个输入和输出组件。

2. 点击 **Draw Wire**（绘制导线）工具。 
3. 将鼠标指针朝着某个终端的接触点方向移动。

   请注意，一旦靠近接触点，鼠标指针会变成黑色的 X：

4. 点击并释放鼠标按键，然后移动指针以开始绘制导线。

   注意：指针会变成+。

   当靠近接触点时，鼠标指针会再次变成黑色的 X：

5. 在第二个终端接触点上，请点击鼠标左键以建立连接。

   选择周围带有虚拟框的导线：

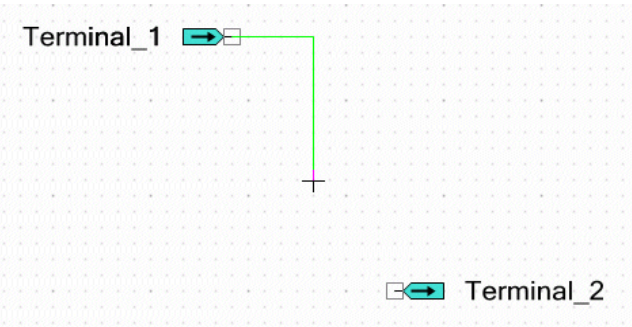
要绘制多点的导线：

1. 使用第一个示例，并从原理图往下移动第二个终端：

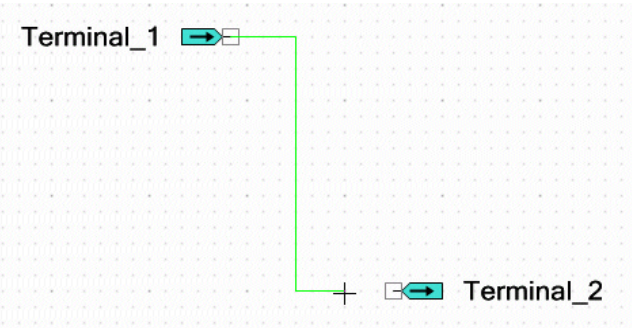
2. 点击 **Draw Wire**（绘制导线）工具。 
3. 点击并释放鼠标按键，然后移动指针以开始绘制导线。
4. 将指针往第二个终端的方向转移。

   注意，导线的形状会变为一个倒 L 字母。



5. 将指针往下移动到与终端扯平的位置，点击并释放鼠标按键，然后将指针左向终端位置。

   请注意，导线获得一个额外的圆点。



   在原理图上，通过在不同的圆点上重复点击鼠标，您可以添加所需的点数。

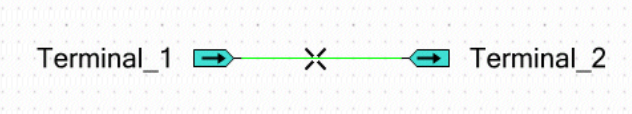
6. 在第二个终端接触点，请点击鼠标左键以建立连接。

   选择周围带有虚拟框的导线。

要连接到另一条导线，请执行下面的操作：

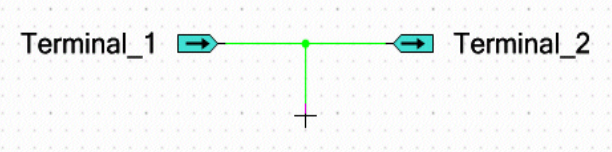
1. 请再次使用第一个示例，并在原理图上添加其他的输出端。
2. 点击 **Draw Wire**（绘制导线）工具。 
3. 将指针向现有导线的方向转移。

   请注意，一旦靠近接触点，鼠标指针会变成 **X**：



4. 点击并释放鼠标按键，然后移动指针以开始绘制导线。

   注意：导线上出现的一个点表示一个连接点。



5. 在第三个终端接触点，请点击鼠标左键以建立连接。

   选择周围带有虚拟框的导线。

要选择导线/网格：

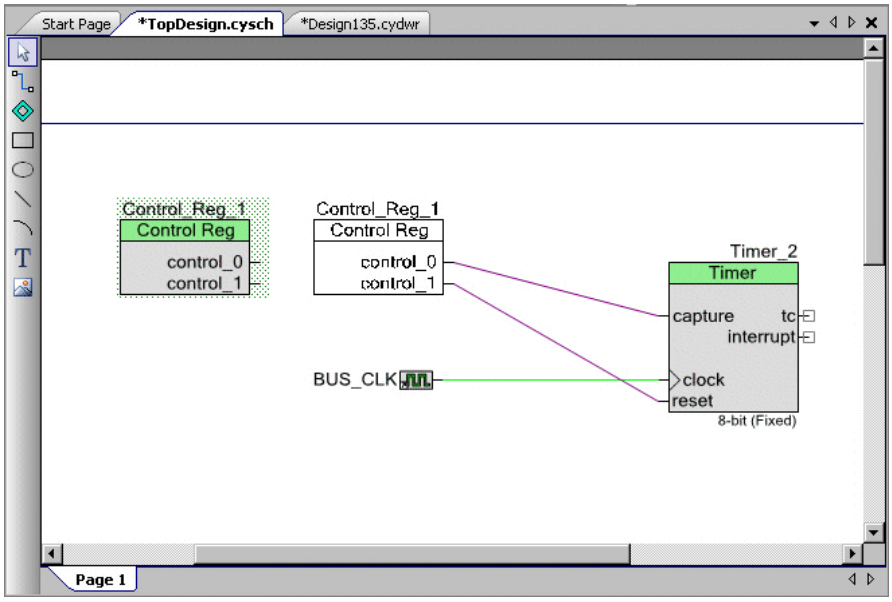
- 若想选择某条导线，请左键单击该导线。
- 若想选择线网中两个点间的某一段，请右键单击导线并选择 **Select Wire Segment**（选择导线段）。
- 要想选择整个线网，请右键单击导线并选择 **Find Wire Trace**。

另外，请参考：

- 使用设计输入工具
- 原理图编辑器
- 设计元素控制板
- 信号名称

弹性连接

在设计原理图上移动物体时，弹性连接特性会自动重新绘制导线。默认情况下，该特性被使能。



如何使用弹性连接特性：

请在原理图上选择一个或多个对象，并将它们转移到屏幕上其他位置。请注意，当您移动个物体时，导线也会延展移动。通过拖动鼠标或使用**[箭头]**键，可以移动各项。

如果暂时停止移动鼠标，则 PSoC Creator 会显示一个重绘导线方法的预览。使用**[箭头]**键不会显示该预览。

如何纠正弹性连接的问题：

在某些情况下，PSoC Creator 不能重绘导线。此时，无法移动。光标将显示为“无”  符号，组件不能返回先前的位置。状态栏将显示一条信息，表示不允许移动。

尝试选择更少的物体或暂时禁用弹性连接特性。您也可以选择更多的物体，并尝试将它们作为一个单元移动。

要暂时禁用弹性连接特性：

使能弹性连接特性时，请按下**[Ctrl]**键，同时移动组件。移动过程中，不会显示导线预览。如果您释放了**[Ctrl]**键，将重新出现导线预览。

如果释放鼠标按键同时按住**[Ctrl]**键，将不会为该移动绘制导线，连接将被断开。

要想关闭弹性连接特性，请进行下述操作：

1. 从 **Tools**（工具）菜单打开 Options（选项）对话框。
2. 在 Design Entry（设计输入）类别中，向下滚动到 **Wires**（导线）部分。
3. 查找弹性连接选项，并选择 **Off**（关闭）。
4. 点击 **OK**，关闭该对话框。

现在，您在移动某个组件时，导线不再被默认重绘。

要暂时使能弹性连接特性：

禁用弹性连接特性时，请按下**[Ctrl]**键，同时移动组件。

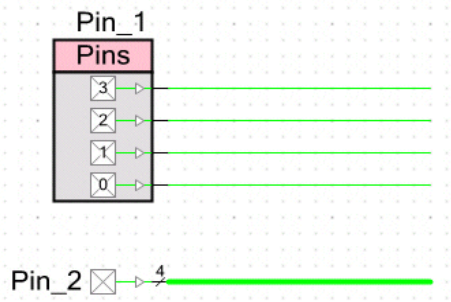
另外，请参考：

- 设计输入选项
- 键盘快捷方式

绘制总线

为了能够绘制总线，您只要将一条导线连接到一个多位连接即可。导线会自动沿用总线的宽度。通过使用 Signal Name（信号名称）对话框，您同样可以指定导线名称的宽度。这样，即使总线没有连接到任何对象，仍能够创建总线。您还可以将来自总线的信号连接到宽度更小的信号。

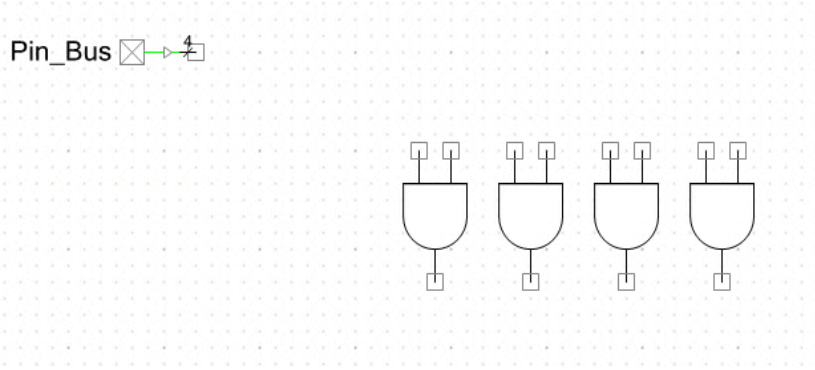
下面框图显示的是两引脚组件的示例，其中一个组件被配置为连接至导线的单独 1 位终端，另一个被配置为 4 位总线。



如何将总线连接到宽度更小的信号（撕信号）：

您不能将总线直接连接到宽度更小的信号（不接收警告列表中的 DRC 错误）。但您必须建立一条多点导线，并标注连接到宽度更小的终端的导线位置，如下面示例所示：

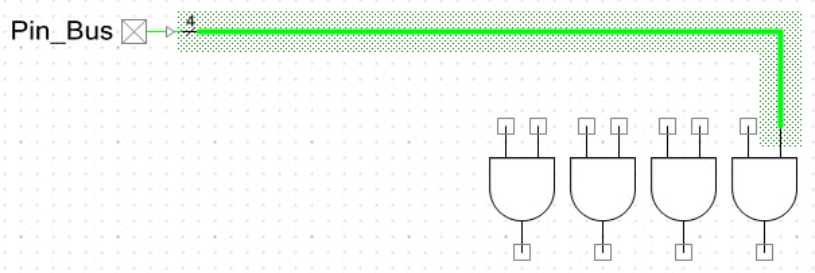
1. 建立一个 4 位引脚组件和四个 AND 组件（每个 AND 组件的终端宽度为 1 位）。



如果需要，请参考配置组件参数、 组件目录和形状的用法。

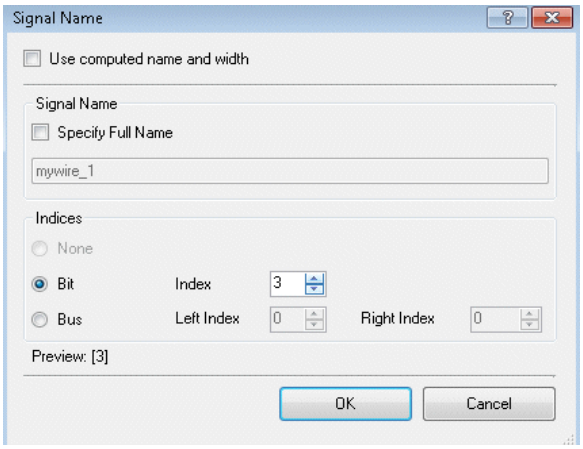
2. 使用 **Draw Wire**（绘制导线）工具并从 4 位端口到最远的 1 位组件终端建立多点连接。

建立好连接后，（在其他的连接错误间）会立即显示出宽度不一致的错误。



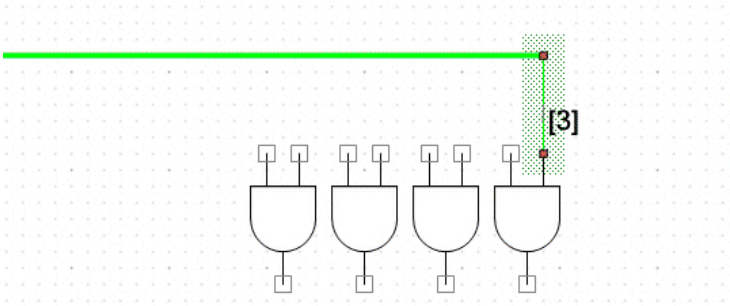
3. 单击画布以取消选择整条导线。
4. 双击连接到 1 位终端的导线段，

将打开 Signal Name（信号名称）对话框，给导线命名。相关详细信息，请参见 信号名称部分。

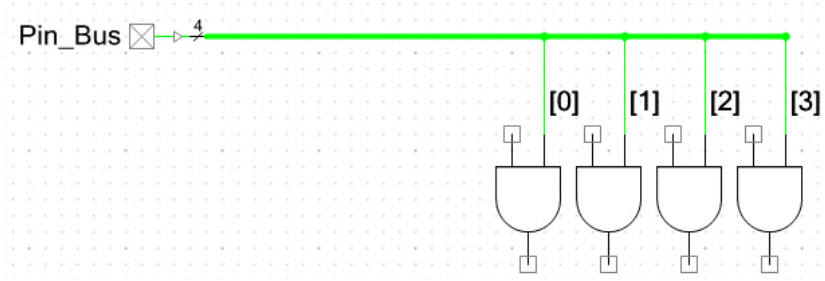


5. 取消选择“Use computed name and width”（使用计算的名称和宽度）以及“Specify Full Name”（指定全名）等选框。
6. 选择 **Bit** 选项和 **Index** [3]。
7. 点击 **OK**，关闭该对话框。

   请注意，标签会显示，且连接到终端的导线会变薄。



8. 将总线连接到其他每一个一位终端。使用上述的相同流程将它们分别标注为[0]、[1]和[2]。



提示：您可以复制和粘贴已完成的导线，然后双击标签以编辑信号名称。

注意：通过使用相同的流程来连接各个宽度不同的信号，但并非需要使用所有输入引脚的位。但是必须驱动输出总线上的所有位。

另外，请参考：

- Signal Name（信号名称）对话框

- 连线的用法
- 导线标签和名称
- 注意列表窗口

导线标签和名称

本节详细介绍了有关在原理图编辑器中显示的及编辑导线标签的内容。导线会有明确的名称，并且每跟（有效的）导线都会有自己的有效名称。您可以在原理图画布上（若显示）或在 Properties 对话框中查看这些名称。

- **Wire Label**（导线标签） — 导线标签是导线的显示名。显示导线标签是可选的，并且只有导线的名称才会被显示。
- **User Name**（名称） — 名称是指通过 Signal Name 对话框指定给导线的标签。不是每一条导线都有名称。
- **Effective Name**（有效名称） — 这是连接子系统计算的名称。例如，名称来自连接到输入原理图终端的原理图中的源。每个（有效）导线都有自己的有效名称，并且始终不为空。

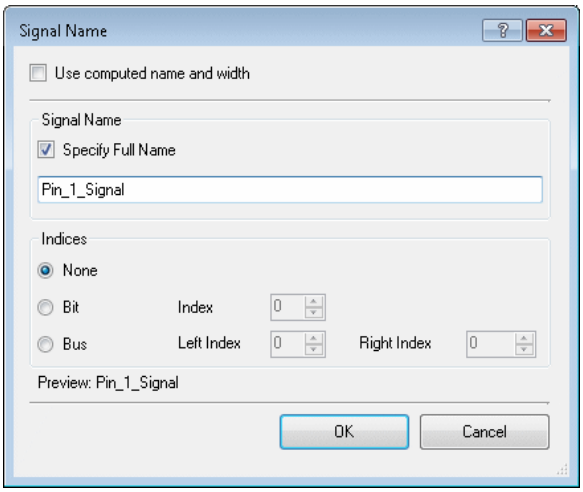
导线标签在原理图编辑器中的显示原理：

- 如果您提供了一个名称，PSoC Creator 将以名称显示标签。
- 如果导线没有名称，则 PSoC Creator 将不会显示标签。
- 编辑名称时，会设置名称。但不会设置有效名称。
- Properties 对话框中的导线属性显示的是有效名称和名称。

注意：如果您复制和粘贴导线的名称，虽然 PSoC Creator 将重新命名其他的所粘贴元素（如组件和终端），但不会重新命名导线。在某些情况下，复制及粘贴连接到终端和组件的导线名称可能会导致 DRC 错误，您需要是当地重名导线或清除标签。

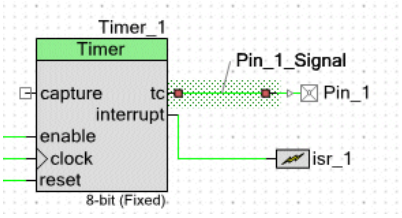
如何定义导线的名称：

1. 双击导线，以打开 Signal Name 对话框。



2. 在该对话框中，请取消选择“Use computed name and width”（使用计算的名称和宽度），并选择“Specify Full Name”（指定全名）选框。
3. 请在该字段中输入所需名称，并点击 **OK**。

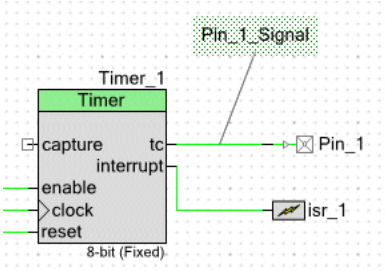
   指定的名称将挨着导线显示。



注意：如果导线已经有了名称，您可以双击其标签来打开信号名称对话框。

要想移动导线标签：

请单击导线标签来选择它，并将其拖放到其他位置。导线锚将附着导线段的中间始终显示。



另外，请参考：

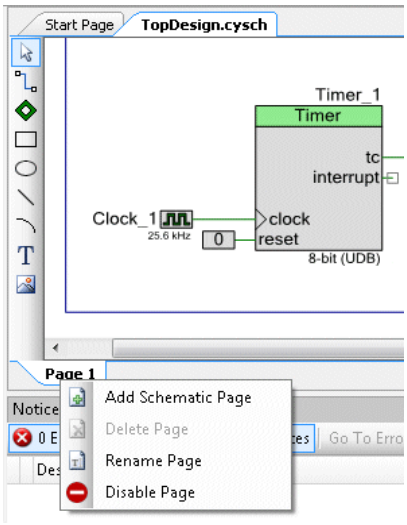
- 原理图编辑器
- 属性
- 信号名称
- 连线的用法

使用多页和链接器

如果在一页上您的设计过大，或如果您想显示设计的各个部分以便于阅读，您可以将多个工作表添加给设计。通过使用工作表链接器，可以连接在不同页上的元素本部分介绍了如何将一页添加到原理图，以及连接不同页上的元素的方法。

要想添加页面：

请右键单击原理图底部的原理图页面选项卡，然后选择 **Add Schematic Page**（添加原理图页面）。

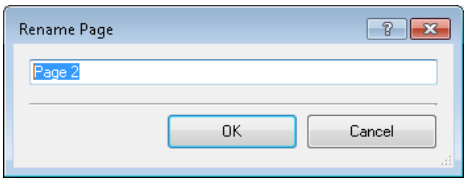


新的页面会作为选项卡式的文档添加。

要重名页面：

1. 请右键单击原理图底部的原理图页面选项卡，然后选择 **Rename Page**（重名页面）。

   这时会显示 Rename Page 对话框。



2. 请输入页面名称并点击 **OK**。

   页面选项卡将显示您输入的名称。

要删除某个页面：

请右键单击原理图底部的原理图页面选项卡，然后选择 **Delete Page**（清除页面）。

这样，选定的页面选项卡将被移除。

注意：如果只有一个页面选项卡，您将无法删除它。

如何使用工作表链接器：

工作表链接器连接到多个工作表上的导线或总线，其连接点位于图形的中心。它们是无名的，并且尚未定义流方向。

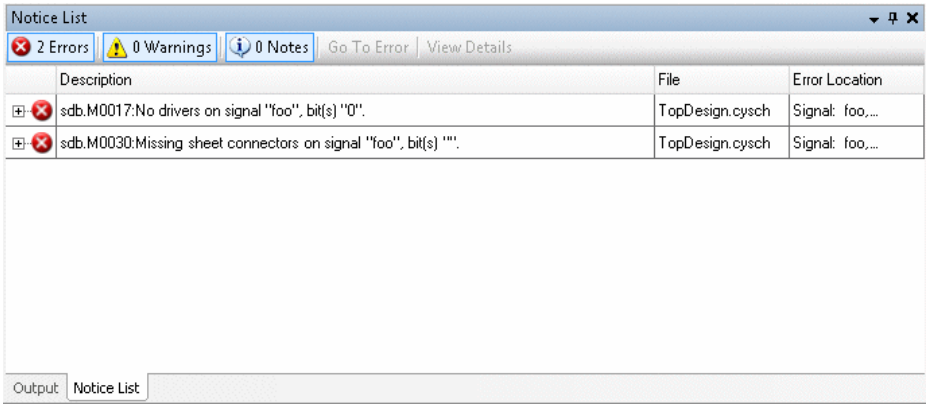
在多个页面上显示的所有导线简单名称以及所有总线的位都要连接到所使用的每个工作表上的链接器。当为各个输入端和总线使用工作表链接器时，请注意输入端中使用某一基本名称时必须定义整个信号。因此，在单独页面上使用输入端未定义的信号位会违反现有的规则。请参见导线标签和名称和绘制总线部分。

注意：即使不使用工作表链接器，您仍能在原理图的同一页上连接导线的名称。

1. 单击 **Wire** 图标，并在原理图中两个不同页面上绘制导线。

2. 双击每一条导线，并给它们起相同的名称（比如“foo”）。

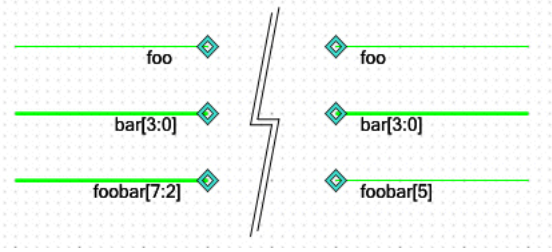
   请注意，注意列表窗口会显示工作表链接器的错误。



3. 点击 **Sheet Connector** 图标，并在每一页中的每一条导线上放置一个链接器。

   一旦连接了所有导线，注意列表窗口将被清除。

下图显示的是一些有效工作表连接的示例：



另外，请参考：

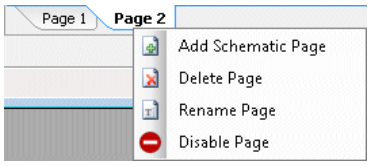
- 连线的用法
- 导线标签和名称
- 终端和导线

禁用/使能原理图页

通过禁用原理图页，您可以指定构建过程中不需要设计的内容。在原理图设计中，您可以禁用（并重新使能）一个或多个页面。这样能够移除各部分，以进行测试和调试，并提供各种不同的配置。

如何禁用页面：

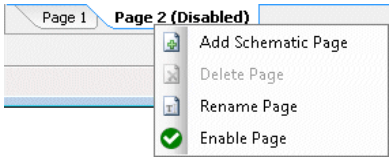
请右键单击原理图底部的原理图页面选项卡，然后选择 **Disable Page**（禁用页面）。



原理图页将被禁用，并且在原理图和选项卡上会出现“Disabled”字符。

如何使能一个被禁用的页面：

请右键单击原理图底部的原理图页面选项卡，然后选择 **Enable Page**（使能页面）。



这样可再次激活原理图页。

注意：如果原理图中其他的激活页面的组件实例名称和您正在尝试使能的页面上的实例名称相同，则会显示 Merge Dialog（合并对话框）以解决冲突问题。

另外，请参考：

- 原理图编辑器
- Merge 对话框

原理图终端的用法

通过原理图的 Design Elements Palette 中的终端工具，您可以绘制数字输入、输出、输入输出以及模拟和外部终端。

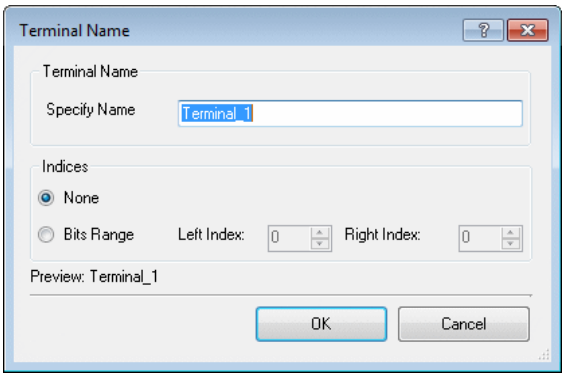
使用原理图实现组件时，请使用原理图终端来表示芯片内的组件的外部连接。更多有关创建组件的信息，请参阅组件创建指南。

注意：当您编辑顶层原理图时，在设计元素控制板中将隐藏原理图终端；仅在组件实现原理图中他们才可见。更多有关信息，请参见创建新的原理图。

如何放置一个单终端：

1. 点击设计元素控制板中的相应 **Terminal**（终端）工具，然后点击设计画布。

   显示 Terminal Name（终端名称）对话框。



2. 适当地指定 **Terminal Name**（中断名称）和 **Index**（索引）信息。
3. 点击 **OK**。

如何放置多个终端：

1. 双击设计元素控制板中的相应 **Terminal**（终端）工具，然后点击设计画布。
2. 重复点击画布，以放置多个终端。

   这称为“粘滞模式”。
3. 按下[**Esc**]键或单击“Select”  工具，可推出粘滞模式。

如何重新命名一个终端：

1. 双击标签，打开 Terminal Name 对话框。
2. 请输入相应的 **Terminal Name**（终端名称）。
3. 点击 **OK**。

注意：如果您复制终端并将其粘贴到原理图上，PSoC Creator 会将“_n”附加到该名称（其中，‘n’是下一个可用的编号），这样重新给所粘贴的组件命名。

如何清除一个终端：

请选择终端并按下[**Delete**]键或点击  。

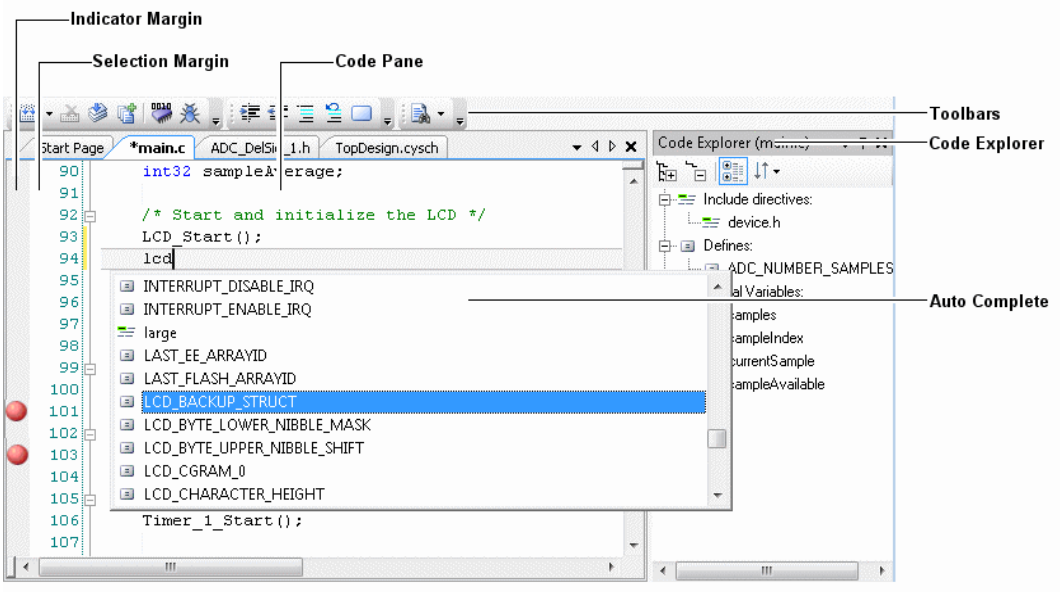
另外，请参考：

- 原理图编辑器
- 设计元素控制板
- Terminal Name（终端名称）

代码编辑器

代码编辑器

通过代码编辑器或文本编辑器，您可以查看并编辑 PSoC Creator 中的源文件。您可以打开选项卡式的文档窗口中的多个文件，然后复制并粘贴文件。



下面内容介绍了文本编辑器的各个不同部分：

- **Code Pane**（代码面板） — 是指显示代码或文本以进行编辑的区域。它为您开发的语言提供了 Autocomplete 语句完成。分别通过使用**[Ctrl]+[-]**和**[Ctrl]+[Shift]+[-]**组合键，您可以返回光标先前所在的位置。
- **Indicator Margin**（指示器边距） — 是指代码编辑器左侧上的一列灰色区域，其中显示了断点、书签、快捷键以及内联代码诊断等指示器。点击该区域会设置相应的代码行上的断点。更多有关信息，请查看调试器部分。
- **Selection Margin**（选择边距） — 是指指示器边距和代码面板之间的一列，您可以通过点击该区域来选择代码行。该区域显示了行编号。此外，当选择 Options 选项卡中 **Text Editor > General** 菜单下的 **Track Changes**，可以跟踪代码的变化。
- 工具栏和指令
- 代码浏览器窗口
- 自动完成

如何打开文本脚编辑器：

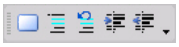
可通过下面各种方式打开文本编辑器中的文件：

- 工作区浏览器 — 双击某个文件。
- **File** 菜单 — 选择 **New** 或 **Open** 以打开文件。

另外，请参考：

- 文档窗口
- 工作区浏览器
- 代码编辑器工具栏
- 代码编辑器的右键菜单指令
- 自动完成
- 代码外形工具窗口
- 查找所有引用
- 内联代码诊断
- 查找与替换
- Options 对话框 >Text Editor 选项
- 跳转到行编号
- 调试器的使用

代码编辑器工具栏



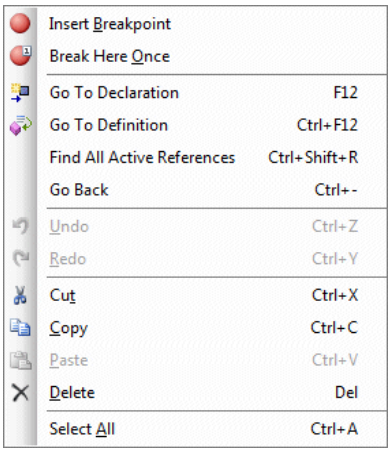
文本编辑器工具栏包含了以下各条指令：

- **Toggle Bookmark**（切换书签） — 添加/删除当前行边距中的书签。
- **Comment Selection**（注释选择）— 将所选的文本更改为注释。
- **Uncomment Selection**（取消注释选择）— 删除选定文本的注释。
- **Increase Line Indent**（增加行间距）— 将选定的文本缩进到下一个制表位。
- **Decrease Line Indent**（减少行间距）— 将选定的文本减少缩进到上一个制表位。

代码编辑器的右键菜单指令

文本编辑器的右键菜单（右键点击时出现的菜单）包含各条指令。当编辑源文件或运行调试器时，可用的指令会发生变化。下面是可用的指令：

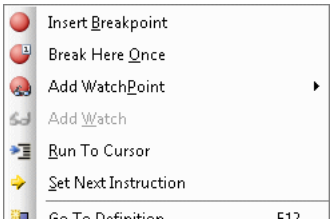
在编辑模式中：



- **Insert Breakpoint**（插入断点）— 将一个新的文件/行断点添加到光标所在的行。请参阅断点窗口中的内容。
- **Break Here Once** — 将一个新的临时文件/行断点添加到光标所在的行。击中该断点一次后，会自动删除它。
- **Go To Declaration**（转至声明） — 与“Go To Definition”（转至定义）相似，该选项会跳转到符号的实现部分。但与 Go To Definition 不同的是，选择 Go To Declaration 会跳转到实际的声明/实现部分。大多数情况下，它会直接跳转到声明函数的源文件。只在未找到声明时（比如，在一个包含库中声明），才不会发生这种情况。
- **Go To Definition**（转至定义） — 如果工具提示信息不足以回答您任何问题，该选项会跳转到讯顶符号的定义。如果在不同的源文件中定义了符号，那么它会跳转到所包含的头文件。如果符号是在当前的原文件中声明的，那么它会跳转到此位置。
- **Find All Active References**（查找所有可用的引用） — 允许搜索符号的所有引用。请参见查找所有引用部分。
- **Go Back** — 将光标重新移动到先前选定的位置。
- **Undo**（撤销）— 撤销上次对文件进行的编辑。
- **Redo**（重做） — 恢复上次撤销的操作
- **Cut**（剪切）— 剪切选定的文本
- **Copy**（复制）— 复制选定的文本
- **Paste**（粘贴）— 将当前剪贴板上的文本粘贴到文件中
- **Delete**（删除）— 删除选定的文本内容
- Select All（全选）— 选择整个文档中的文本

在调试模式中：

右键菜单包含了与编辑模式下相同的指令，另外还包含下面各条指令：



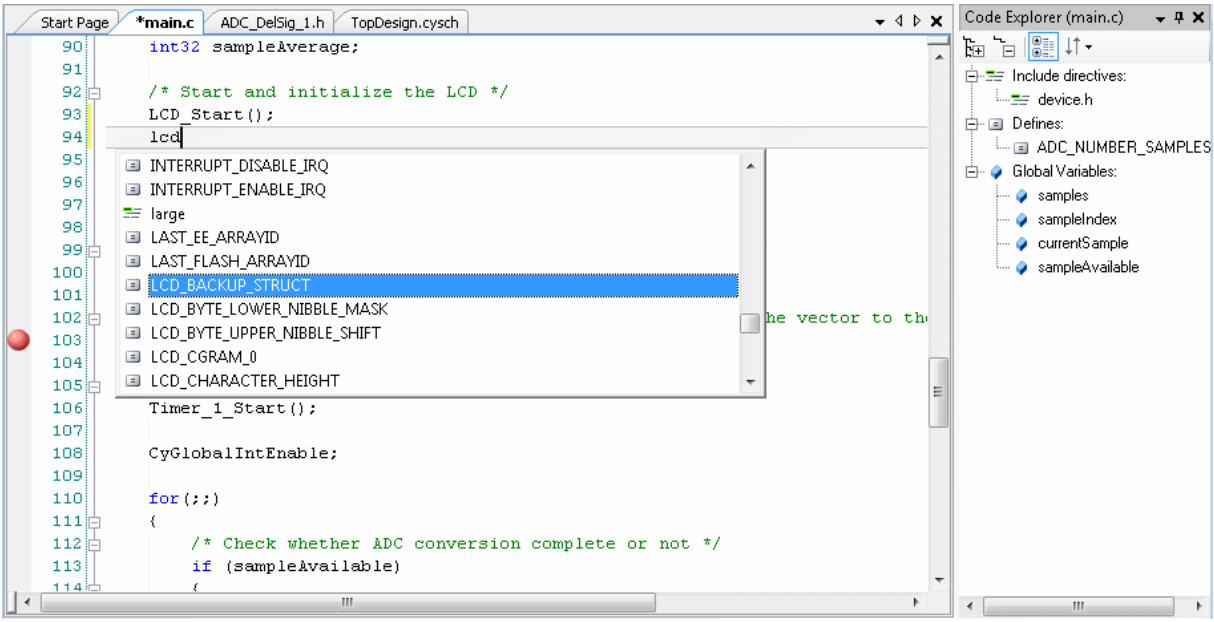
- **Add Watchpoint**（添加观察点）— 将新的观察点添加到所选变量；请参考变量观察点部分。
- **Add Watch**（添加监视）— 将变量或选定的文本添加到监视窗口中；请参考监视窗口部分讲述的内容
- **Run to Cursor**（运行至光标处）— 恢复执行代码，直至到达该行
- **Set Next Instruction**（设置下一条指令）— 使程序跳转到当前的代码行

另外，请参考：

- 文本编辑器
- 文本编辑器工具栏
- 调试器的使用
- 调试器的工具栏指令
- 调试器菜单指令
- 变量观察点
- 监视窗口

自动完成

自动完成功能提供的上下文识别的下拉列表可能与您键入的关键字、 类型、 变量、 宏和函数有关。



通过该功能，您可以快速编写代码。它还作为文档源提供了巨大的价值。您不用再经常返回到数据手册或源文件也能查找所需函数。只要开始输入，即能看到可用内容。

注意：构建项目前，完成特性所创建的列表是受限制的。

除提供了可用项名单外，代码编辑器还显示了选定条目的工具提示，用以提供完整的识别标志。对于各函数，它显示了函数的返回值和名称，以及所有参数变量的类型和名称。

要使用该功能：

1. 如果未使能自动完成特性，请通过 Text Editor Options（文本编辑器选项）对话框使能该特性。该特性在默认情况下为使能状态。
2. 正常构建您的设计，并点击 **Generate Application**（生成应用）按键  来允许 PSoC Creator 生成或更新各个 API 文件。您同样可以通过组合键**[Ctrl] + [Space]**初始化该特性。

3. 打开代码编辑器并开始输入，注意当找到匹配的条目时，会打开下拉菜单。
4. 滚动浏览整个列表以找出所需要的项，或者继续输入内容，直到突出显示所需条目为止。
5. 一旦在列表中选中所需项目，请按下**[Space]**、**[Tab]**、**[Enter]**或**[;]**键以自动完成该字。按下**[Esc]**键会取消自动完成操作。

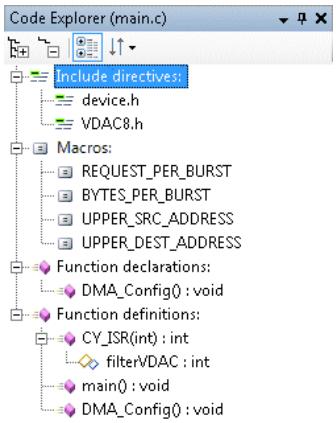
插入选中的条目并调整大小写的写法，使之与您所完成的实际函数名称相匹配。

另外，请参考：

- 代码编辑器

代码浏览器窗口

通过代码浏览器特性，您可以快速查看源文件的整个结构。您可以查看所定义的内容和位置。另外还可以使用它跳转到代码中指定的位置。



要想显示代码浏览器工具窗口：

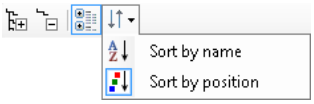
当您打开代码编辑器时，会自动显示该工具窗口。它位于工具窗口的右侧。关闭了该工具窗口后，要想重新打开它，可以通过选择 **View** 菜单中的 **Code Explorer** 实现。

要想跳转到指定的位置：

双击某个条目，或在突出显示条目时按下**[Enter]**键。

代码编辑器会立即滚动到该符号。

工具栏：



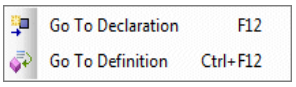
代码浏览器工具栏包含了下面各格式选项，用以更改显示方式。

- **Expand All / Collapse All**（全部展开/全部折叠）— 各个按键用于展开或折叠整个目录树中所有节点。
- **Show in Groups**（组显示）— 通过该按钮可切换各种符号的显示方式：分组显示和单独显示 。

- **Sort Order**（排序）— 用于更改文件中名称或位置的前后顺序的按键和下拉菜单。

右键菜单：

当右键单击目录树上某个节点项时，下面各指令可用。另外，请参考代码编辑器的右键菜单指令部分讲述的内容。



- **Go To Declaration**（转至声明）— 该选项用以跳转到选定符号的声明。
- **Go To Definition**（转至定义）— 该选项用以跳转到选定符号的定义。

标签：

显示在目录树中的每一项都有一个标签。下面列出的是各标签的含义：

- – 包括方向
- – 宏
- – 结构
- – 类型定义
- – 联盟
- – 枚举
- – 枚举常量
- – 函数
- – 参数变量
- – 变量

另外，请参考：

- 代码编辑器
- 代码编辑器的右键菜单指令

查找所有引用

通过 The Find All References（查找所有引用）性能，您可以找出正在使用特殊符号的所有项目（如函数、宏、变量、类型）。

要想使用该功能：

请右键单击代码编辑器中的某个符号，并从右键菜单中选择 **Find All Active References**，或按下**[Ctrl]** + **[Shift]** +**[R]** 组合键。

PSoC Creator 将扫描整个项目以找出符号的所有用途，并将结果显示在 Find Results（查找结果）窗口中。

双击 Find Results 窗口中的某一项，以转到它在源代码中的位置。

另外，请参考：

- 代码编辑器
- 查找结果

内联代码诊断

通过该特性，可在编辑器中直接显示编译器诊断信息。这样，即使未完成构建项目，您仍能看到代码中存在的问题。

编辑器中显示的诊断信息可能与构建项目时所生成的错误/警告不相对应。这是因为编辑器使用了一个通用的构建框架，该框架中不存在和有效工具链相同的检查器。因此，在注意列表中没有显示任何诊断信息。注意列表中显示的错误/警告受当前设计配置生成的内容的严格限制。

注意：如果显示了某个有关丢失‘包括文件’的错误，将不会显示本文档中的任何错误。这是因为丢失‘包括’（include）文件被视为非常严重的错误，限制了额外的处理。如果无法找到重要的包含文件，很可能会使文件中几乎所有行都被视为错误，这样掩盖实际的问题（丢失‘包括’文件）。

要想使用该功能：

请保存当前文件。代码中的所有问题都将显示为带有一条波形线。

将鼠标悬停在边距指示器或波形线上，以查看解决该问题的工具提示。

如果该问题是一个警告或错误，则会显示左侧的边距。

另外，请参考：

- 代码编辑器
- 参考工具提示

参考工具提示

根据代码编辑器所提供的参考工具提示，可以更容易读取和了解代码（比如，代码模块正在执行的操作，或用于执行函数的各个参数）。

```
    /* Place your initialization/startup code here (e.g. MyIn
    PWM_1_SetCaptureMode(1);
    void PWM_1_SetCaptureMode ( uint32 triggerMode )
    /* CyGlobalIntEnable; */ /* Uncomment this line to enable
    for(;;)
```

```
19 #if(I2C_1_SCB_MODE_I2C_INC)
20     I2C_1_SCB_MODE_I2C_INC ( 0u != (I2C_1_SCB_MODE_I2C & I2C_1_SCB_MODE))
21 #endif /* (I2C_1_SCB_MODE_I2C_INC) */
22
23 #if(I2C_1_SCB_MODE_SPI_INC || I2C_1_SCB_MODE_UART_INC)
24     #include "I2C_1_SPI_UART_PVT.h"
25 #endif /* (I2C_1_SCB_MODE_SPI_INC || I2C_1_SCB_MODE_UART_INC) */
```

要使用该功能：

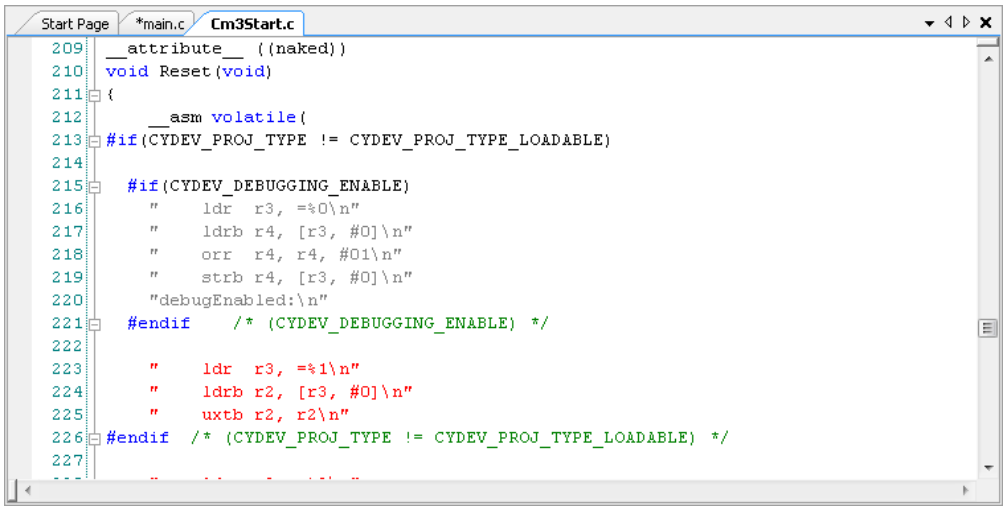
将鼠标悬停在任何参考类型（变量、函数或宏）上可以观察该项的声明签名。

另外，请参考：

- 代码编辑器

禁用的代码

代码编辑器使用灰色来标识被禁用的代码。这样有助于解决因代码拥挤而难读的问题。



设计中使用的 PSoC 器件和组件都是高度可配置的，并且 PSoC Creator 还支持几种不同的器件和工具链。这样要求使用整个固件代码中使用的#ifdef 语句的有效数量，使得读取和调试代码非常困难。

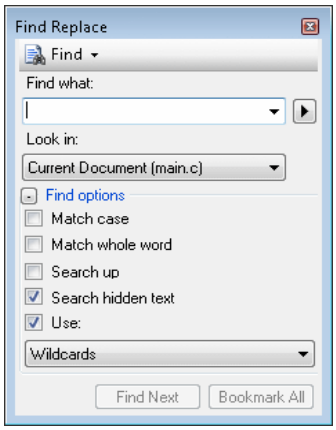
禁用代码特性可明显提高代码的可读性和可理解性。这样也有助于提高调试经验。您可以清楚地看到大量代码模块都被禁用，并很快能够明白对它们进行调试的原因。

会自动支持该功能而无需用户干预。

查找与替换

查找与替换

Find Replace（查找与替换）对话框用于定位文件中的文本，并可以选择替换它。

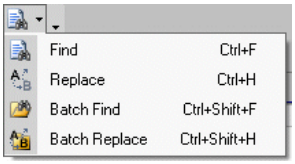


根据打开的方式，该对话框略有不同。Batch Find（批量查找）和 Batch Replace（批量替换）具有几个帮助选项。

要打开 Find Replace 对话框：

可以使用下面任何一种方式打开该对话框：

- 按下**[Ctrl]+[F]**（以查找）或**[Ctrl]+[H]**组合键（进行查找和替换）。
- 在 **Edit** 菜单中，请选择 **Find and Replace**，然后选择相应的查找/替换指令。
- 点击所显示的查找/替换按键标签，或从下拉菜单中选择一个查找/替换选项。



要使用 Find Replace 对话框：

Find what（查找内容）

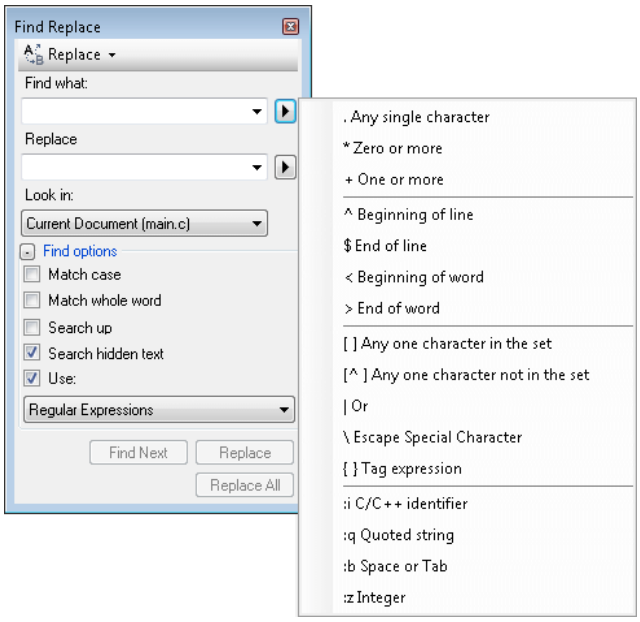
请使用该字段来指定需要查找的字符串或表达式。通过在该下拉菜单中选择，您可以重新使用最后 20 个搜索字符串；或者键入新文本字符串或表达式进行查找。

Replace with（替换为）

使用该字段可将 **Find what** 字符串的实例替换为其他字符串。要删除 **Find what** 字符串的实例，请保留该字段为空白。

Expression Builder（表达式生成器）

选中 **Find options** 下的 **Use** 选框时，请使用 **Find what** 和 **Replace with** 字段旁边的三角按键。



根据所选的 **Use** 选项，点击该按键可显示通配符或正则表达式的列表。从该列表中任选一项，并将其添加到 **Find what** 或 **Replace with** 字符串内。

Look in（查找的位置）

使用该下拉菜单来选择 **Current Document**（当前文档）或 **All Open Documents**（所有打开的文档）。

Find options（查找选项）

您可以展开或折叠 Find Options 选项。可以选择或清除下面各选项：

- **Match case**（形式匹配）— 勾选该选项时，Find Results 窗口只显示内容和表现形式均匹配的“Find what”字符串的实例。例如，当勾选“MatchCase”时，如果搜索“MyObject”，将返回“MyObject”，而不是“myobject”或“MYOBJECT”。
- **Match whole word**（全字匹配）— 勾选该选项时，Find Results 窗口只会显示符合全字匹配的“Find what”字符串的实例。例如，搜索“MyObject”时，将返回“MyObject”而不是“CMyobject”或“MyobjectC”。
- **Search up**（向上搜索）— 勾选该选项时，会从插入点开始搜索到文件头。
- **Search hidden text**（搜索隐藏的文本）— 勾选该选项时，将搜索被隐藏和折叠的文本，如设计时间控件的元数据、概述文档的隐藏部分或者被折叠的类别或方法。
- **Use**（使用）— 该选项指出了如何理解在 **Find what** 或 **Replace with** 字段中键入的特殊字符。选项包括：
- **Wildcards**（通配符）— 特殊字符（如星号（*）和问号（？））分别代表一个或多个字符。请参考通配符部分。
- **Regular Expressions**（正则表达式）— 特殊符号定义了匹配文本的模式。请参考正则表达式部分。

Buttons（按键）

请点击相应按键，具体如下：

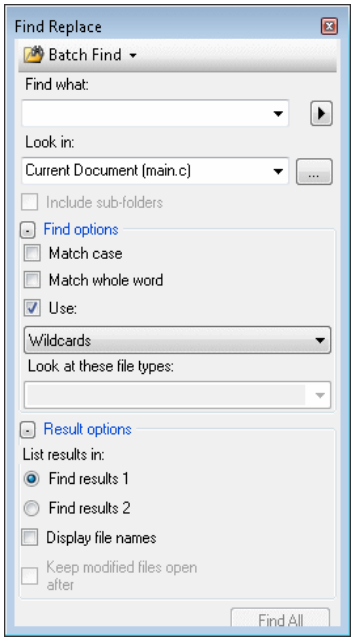
- **Find Next**（查找下一个） — 点击该按键可以在 Look in 选择的搜索范围中查找下一个 **Find what** 字符串实例。
- **Bookmark All**（全部书签）— 点击该按键会在文本编辑器的左边缘显示书签，指出发生 **Find what** 字符串实例的每一行。
- **Replace**（替换）— 通过点击该按键，可以将 **Find what** 字符串的当前实例替换为 **Replace with** 字符串的实例，并在 **Look in** 范围内查找下一个实例。
- **Replace All**（全部替换）— 通过点击该按键，可以将 **Look in** 范围内所有文件中 **Find what** 字符串的所有实例替换为 **Replace with** 字符串的实例。

另外，请参考：

- 文本编辑器
- 批量查找
- 批量替换
- 正则表达式
- 通配符

Batch Find（批量查找）

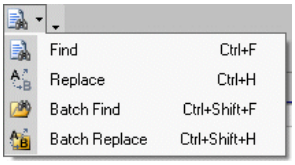
通过 Batch Find 对话框，您可以搜索字符串或表达式中特定文件组的代码。**Result options** 下所选的 Find Results 窗口中将显示查找的匹配和所采取的方式。



要打开 Batch Find 对话框：

可以使用下面任何一种方式来打开该对话框：

- 按下[**Ctrl**]+[**Shift**]+[**F**]组合键。
- 在 **Edit** 菜单中，请选择 **Find and Replace**，然后选择 **Batch Find**。
- 从下拉菜单选择 **Batch Find**。



要使用 Batch Find 对话框：

Find what（查找内容）

请使用该字段来指定需要查找的字符串或表达式。通过在该下拉菜单中选择，您可以重新使用最后 20 个搜索字符串；或者键入新文本字符串或表达式进行查找。

Expression Builder（表达式生成器）

当勾选了 **Find options** 中的 **Use** 选框时，请使用 **Find what** 字段旁边的三角按键。

根据所选的 **Use** 选项，点击该按键可显示通配符或正则表达式的列表。从该列表中选择任意一项，并将其添加到 **Find what** 字符串内。

Look in（查找的位置）

使用该下拉菜单来选择 **Current Document**（当前文档）或 **All Open Documents**（所有打开的文档）。

点击**[...]**按键以选择需要搜索的目录。您也可以勾选 **Include subfolders**（包括子文件夹）选框来搜索特定搜索目录的子文件夹。

Find options（查找选项）

您可以展开或折叠 Find Options 选项。可以选择或清除下面各选项：

- **Match case**（形式匹配）— 勾选该选项时，Find Results 窗口只显示内容和表现形式均匹配的“Find what”字符串的实例。例如，当勾选“MatchCase”时，如果搜索“MyObject”，将返回“MyObject”，而不是“myobject”或“MYOBJECT”。
- **Match whole word**（全字匹配）— 勾选该选项时，Find Results 窗口只会显示符合全字匹配的“Find what”字符串的实例。例如，搜索“MyObject”时，将返回“MyObject”而不是“CMyobject”或“MyobjectC”。
- **Use**（使用）— 该选项指出了如何理解在 **Find what** 或 **Replace with** 字段中键入的特殊字符。选项包括：
  - **Wildcards**（通配符）— 特殊字符（如星号（*）和问号（？））分别代表一个或多个字符。请参考通配符部分。
  - **Regular Expressions**（正则表达式）— 特殊符号定义了匹配文本的模式。请参考正则表达式部分。
- **Look at these file types**（观看这些文件类型）— 该列表指出需要在 **Look in** 目录中搜索的文件类型。如果该字段为空白，将搜索 **Look in** 目录中的所以文件。
  - 如果将该列表中的任意项目输入到预配置的搜索字符串，则会查找相应特殊类型的文件。
  - 要查找下拉列表中不可用的文件类型，请在文件名称输入一个星号（*）通配符，然后是一点（.）最后是

所需的文件扩展名。如果想查找多种文件类型，请输入多个文件扩展名，并使用分号（;)隔开。

Result options（结果选项）

您可以展开或折叠各结果选项。可以选择或清除下面各选项：

- **Find Results 1 window**（查找结果 1 窗口）— 选择该选项可以在 Find Results 1 窗口中显示当前搜索的结果。自动打开该窗口，以显示您查找的结果。要想手动打开该窗口，请在 **View** 菜单中选择 **Find Results**，然后再选择 **Find Results 1**。
- **Find Results 2 window**（查找结果 2 窗口）— 选择该选项可以在 Find Results 2 窗口中显示当前搜索的结果。自动打开该窗口，以显示您查找的结果。要想手动打开该窗口，请从 **View** 菜单中选择 **Find Results**，然后选择 **Find Results 2**。
- **Display file names**（显示文件名称）— 通过选中该复选框，可以显示包含了搜索匹配的文件列表，并不显示搜索匹配项本身。

Buttons（按键）

请点击相应按键，具体如下：

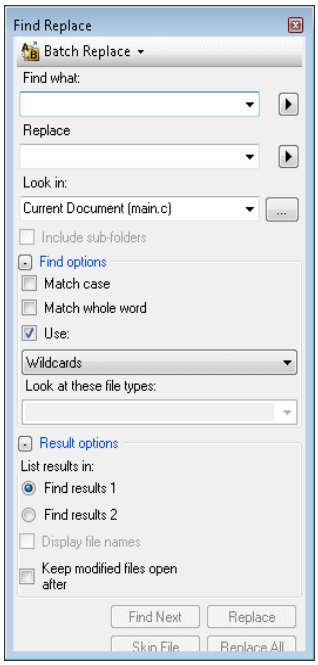
- **Find All**（全部查找）— 点击该按键以在 Look in 选择的搜索范围中查找 Find what 字符串的所有实例。这些结果会显示在 Result options 下选择的 Results 窗口中。

另外，请参考：

- 批量替换
- 查找与替换
- 查找结果
- 正则表达式
- 通配符

Batch Replace（批量替换）

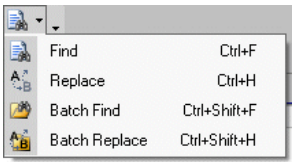
通过 Batch Replace 对话框，您可以搜索字符串或表达式中特定文件组的代码，并可以更改部分或所有找到的匹配结果。**Result Options** 下所选的 Find Results 窗口中将显示查找匹配和所采取的方式。



要打开 Batch Replace 对话框：

可以使用下面任何一种方式打开该对话框：

- 按下**[Ctrl]**+**[Shift]**+**[H]**组合键。
- 在 **Edit** 菜单中，鼠标指向 **Find and Replace**，然后点击 **Batch Replace**。
- 从下拉菜单中选择 **Batch Replace**。



要使用 Batch Replace 对话框：

Find what（查找内容）

请使用该字段来指定需要查找的字符串或表达式。通过在该下拉菜单中选择，您可以重新使用最后 20 个搜索字符串；或者键入新文本字符串或表达式进行查找。

Expression Builder（表达式生成器）

当勾选了 **Find options** 中的 **Use** 选框时，请使用 **Find what** 字段旁边的三角按键。

根据所选的 **Use** 选项，点击该按键可显示通配符或正则表达式的列表。从该列表中选择任意一项，并将其添加到 **Find what** 字符串内。

Look in（查找的位置）

使用该下拉菜单来选择 **Current Document**（当前文档）或 **All Open Documents**（所有打开的文档）。

点击**[...]**按键以选择需要搜索的目录。您也可以勾选 **Include subfolders**（包括子文件夹）选框来搜索特定搜索目录的子文件夹。

Find options（查找选项）

您可以展开或折叠 Find Options 选项。可以选择或清除下面各选项：

- **Match case**（形式匹配）— 勾选该选项时，Find Results 窗口只显示内容和表现形式均匹配的“Find what”字符串的实例。例如，当勾选“MatchCase”时，如果搜索“MyObject”，将返回“MyObject”，而不是“myobject”或“MYOBJECT”。
- **Match whole word**（全字匹配）— 勾选该选项时，Find Results 窗口只会显示符合全字匹配的“Find what”字符串的实例。例如，搜索“MyObject”时，将返回“MyObject”而不是“CMyobject”或“MyobjectC”。
- **Use**（使用）— 该选项指出了如何理解在 **Find what** 或 **Replace with** 字段中键入的特殊字符。选项包括：
  - **Wildcards**（通配符）— 特殊字符（如星号（*）和问号（？））分别代表一个或多个字符。请参考通配符部分。
  - **Regular Expressions**（正则表达式）— 特殊符号定义了匹配文本的模式。请参考正则表达式部分。
- **Look at these file types**（观看这些文件类型）— 该列表指出需要在 **Look in** 目录中搜索的文件类型。如果该字段为空白，将搜索 **Look in** 目录中的所以文件。
  - 如果将该列表中的任意项目输入到预配置的搜索字符串，则会查找相应特殊类型的文件。
  - 要查找下拉列表中不可用的文件类型，请在文件名称输入一个星号（*）通配符，然后是一点（.）最后是所需的文件扩展名。如果想查找多种文件类型，请输入多个文件扩展名，并使用分号（;)隔开。

Result options（结果选项）

您可以展开或折叠各结果选项。可以选择或清除下面各选项：

- **Find Results 1 window**（查找结果 1 窗口）— 选择该选项可以在 Find Results 1 窗口中显示当前搜索的结果。自动打开该窗口，以显示您查找的结果。要想手动打开该窗口，请在 **View** 菜单中选择 **Find Results**，然后再选择 **Find Results 1**。
- **Find Results 2 window**（查找结果 2 窗口）— 选择该选项可以在 Find Results 2 窗口中显示当前搜索的结果。自动打开该窗口，以显示您查找的结果。要想手动打开该窗口，请从 **View** 菜单中选择 **Find Results**，然后选择 **Find Results 2**。
- **Keep modified file open after**（在下列操作后保持所修改的文件打开）— 勾选该复选框可以保持在批量替换过程中所修改的文件。

Buttons（按键）

请点击相应按键，具体如下：

- **Find Next**（查找下一个） — 点击该按键可以在 Look in 选择的搜索范围中查找下一个 **Find what** 字符串实例。
- **Replace**（替换）— 通过点击该按键，可以将“Find what”字符串的当前实例替换为“Replace with”字符串的实例，并在“Look in”范围内查找下一个实例。
- **Replace All**（全部替换）— 通过点击该按键，可以在“Look in”范围内所有文件中将“Find what”字符串的所有实例替换为“Replace with”字符串的实例。

- **Skip File**（跳过文件）— 当“Look in”列表中包含多个文件时，该按键可用。如果您不想搜索或更改当前文件，请点击该按键。继续在“Look in”列表上的下一个文件中搜索。

另外，请参考：

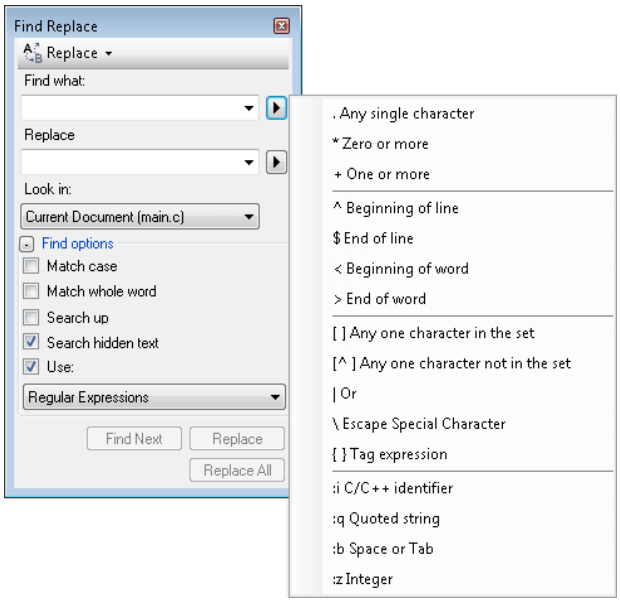
- 批量查找
- 查找与替换
- 查找结果
- 正则表达式
- 通配符

Regular Expressions（正则表达式）

正则表达式是一个简单灵活的符号，用以查找和替换文本的模式。特殊的正则表达式组可用在 Find Replace 窗口的 **Find what** 字段中。

当进行查找和替换操作时，为了能够在 **Find what** 字段中使用正则表达式，请在 **Find Options** 下依次选择 **Use > Regular Expressions**。

点击 **Find what** 字段旁边的三角按键会显示最常用的正则表达式列表。



从 Expression Builder（表达式生成器）中选择任何项目时，都会将其插入到 **Find what** 字符串中。

注意： 可用于 Find what 字符串中的正则表达式和可用于‘.NET’框架编程中的正则表达式在语法上不一样。例如，在 Find Replace 中，大括号{}是用于标记的表达式。所以表达式 zo{1}匹配于 zo 后面带有标记 1 的所有出现，如 Alonzo1 和 Gonzo1 中。但是，在‘.NET’框架中，大括号{}则用于确定数量。因此，表达式 zo{1}匹配于“z”后面只有一个“o”的所有出现，如匹配于“zone”（而不是“zoo”）。

用于查找和替换的正则表达式：

下面显的是参考列表中可用的正则表达式。

表达式	语法	说明
Any character（任意字符）	.	除换行符外，匹配于任意单字符。
Zero or more（零或多个）	*	不匹配，或匹配于上述多个表达式的事件，使全部都得到匹配。
One or more（一个或多个）	+	至少匹配于上述一个表达式的事件。
Beginning of line（行开头）	^	将匹配字符串稳固地连接到某一行的开头部分。
End of line（行结尾）	$	将匹配字符串稳固地连接到某一行的结尾。
Beginning of word（字开头）	<	只有某个字从文本中该点开始时，才会匹配。
End of word（字结尾）	>	只有某个字在文本中的该点结束时，才会匹配。
Line break（换行符）	\n	匹配于一个独立于平台的换行符。在替换表达式中，请插入一个换行符。
Any one character in the set（一组中的任意字符）	[]	匹配于[]中的任意字符。要确定字符的范围，请使用破折号（-）分别列出开始字符和结束字符，如在[a-z]中。
Any one character in the set（不属于某组的任意字符）	[^...]	同后面是^.的字符组中的任意字符不相匹配。
Or（或）	|	匹配于 OR 符号(|)之前或之后的表达式。大部分用于同一组中。例如，(sponge|mud) bath 匹配于“sponge bath”和“mud bath”。
Escape（避开）	\	匹配于一个作为文字跟随反斜线（\）的字符。这样，您可以查找到用于正则表达式的字符，如{和^。例如，\^用于搜索^字符。
C/C++ Identifier（C/C++标识符）	:i	匹配于表达式（[a-zA-Z_$][a-zA-Z0-9_$]*）
Quoted string（带引号的字符串）	:q	匹配于表达式(("[^"]*")|('[^']*'))。
Space or Tab（空格或制表符）	:b	匹配于空格字符或制表符。
Integer（整数）	:z	匹配于表达式([0-9]+)。

其他正则表达式

用于查找和替换操作中的所有正则表达式列表比可显示在参考列表中的表达式列表更长。您也可以将下面任何正则表达式插入到 **Find what** 字符串中：

表达式	语法	说明
Minimal zero or more（最小为零或更多）	@	不匹配，或匹配于上述多个表达式的事件，以尽可能匹配于最少的字符。
Minimal zero or more（最小为一个或更多）	#	匹配于一个或多个上述表达式的事件，以尽可能匹配于最少的字符。
Repeat n times（重复 n 次）	^n	匹配于上述表达式的 n 个出现。例如，[0-9]^4 匹配于任意的 4 位数字序列。
Grouping（分组）	()	将某个子表达式进行分组。
nth tagged text（第 n 个标记文本）	\n	在查找和替换表达式中，它表示与第 n 个标记表达式相匹配的文本，其中 n 的范围为 1 到 9。 在替换表达式中，\0 会插入整个匹配文本。

表达式	语法	说明
Right-justified field（右对齐字段）	\(w,n)	在替换表达式中，以至少为 w 个字符的宽度来对字段中的第 n 个标记表达式进行右对齐。
Right-justified field（左对齐字段）	\(-w,n)	在替换表达式中，以至少为 w 个字符的宽度来对字段中的第 n 个标记表达式进行左对齐。
Prevent match（防止匹配）	~(X)	当 X 出现在表达式中的某一位置，可防止发生该处的匹配。例如，real~(ity) 匹配于“realty”中的“real”，而不匹配于“reality”中的“real”。
Alphanumeric character（字母数字字符）	:a	匹配于表达式([a-zA-Z0-9])。
Alphabetic character（字母字符）	:c	匹配于表达式([a-zA-Z])。
Decimal digit（十进制数字）	:d	匹配于表达式([0-9])。
Hexadecimal digit（十六进制数字）	:h	匹配于表达式([0-9a-fA-F]+)。
Rational number（有理数）	:n	匹配于表达式(([0-9]+.[0-9]*)|([0-9]*.[0-9]+)|([0-9]+))。
Alphabetic string（字母字符串）	:w	匹配于表达式([a-zA-Z]+)。
Escape（逃逸）	\e	Unicode U+001B。
Bell（报警)	\g	Unicode U+0007。
Backspace（退格键）	\h	Unicode U+0008。
Tab（制表键）	\t	匹配于制表符，Unicode U+0009。
Unicode character（Unicode 字符）	\x#### 或 \u####	匹配于 Unicode 值所提供的字符，其中####是十六进制数字。您可以指定基本多语言平面范围外的某个字符（即代替字符），该字符有一个符合 ISO 10646 标准的码位或有两个提供代替字符对的值的 Unicode 码位。

标准 Unicode 字符的属性

下表列出的是匹配于标准 Unicode 字符属性的语法。两个字母的缩写和 Unicode 字符属性数据库中列出的缩写相同。它们可能被指定作为字符组的一部分。例如，表达式[:Nd:Nl:No]匹配于任何类型的数字。

表达式	语法	说明
Uppercase letter（大写字母）	:Lu	匹配于任意一个大写字母。例如，:Luhe 匹配于“The”，不匹配于“the”。
Lowercase letter（小写字母）	:Ll	匹配于任意一个小写字母。例如，:Llhe 匹配于“the”，不匹配于“The”。
Title case letter（标题大小写字母）	:Lt	匹配于一个大写字母和一个小写字母相结合的字符组，如 Nj 和 Dz。
Modifier letter（修正符字母）	:Lm	匹配于字母或标点符号（比如：逗号、 交叉重音符和双引号），用于表示对前面字母的修改。
Other letter（其他字母）	:Lo	匹配于其他字母，如哥特体字母 ahsa。
Decimal digit（十进制数字）	:Nd	匹配于十进制数字，如 0-9 以及它们的全宽度相等的字符。
Letter digit（字母数字）	:Nl	匹配于字母数字，如罗马数字和表意数字零等
Other digit（其他数字）	:No	匹配于其他数字，如旧的斜体数字一。
Open punctuation（标点符号插入）	:Ps	匹配于标点符号插入，如方括号和大括号的左半边。
Close punctuation（标点符号结束）	:Pe	匹配于标点符号结束，如方括号和大括号的右半边。
Initial quote punctuation（起引号）	:Pi	匹配于开始的双引号。

表达式	语法	说明
Final quote punctuation（引回号）	:Pf	匹配于单引号和双引号的引回号。
Dash punctuation（破折号标点符号）	:Pd	匹配于破折号的标点符号
Connector punctuation（连接标点符）	:Pc	与下划线标记相匹配。
其他标点符号	:Po	匹配于（,）、?、"、!、@、#、%、&、*、\、（:）、（;）、' 和 /。
Space separator（空格分隔符）	:Zs	匹配于空格。
Line separator（行分隔符）	:Zl	匹配于 Unicode 字符 U+2028。
Paragraph separator（段落分隔符）	:Zp	匹配于 Unicode 字符 U+2029。
Non-spacing mark（非空格标记）	:Mn	匹配于非空格标记。
Combining mark（组合字符）	:Mc	匹配于组合字符
Enclosing mark（环绕标记）	:Me	匹配于环绕标记。
Math symbol（算术符号）	:Sm	匹配于+、=、~、|、< 和 >。
Currency symbol（货币符号）	:Sc	匹配于$和其他货币符号。
Modifier symbol（修正符号）	:Sk	匹配于修正符号，如音调符号、重音符和长音符。
Other symbol（其它符号）	:So	匹配于其他符号，如版权符号、段落符号标志以及和度数符号。
Other control（其他控制）	:Cc	匹配于如 TAB（制表符）和 NEWLINE（换行符）等 Unicode 控制字符。
Other format（其他格式）	:Cf	格式控制字符，例如双向控制字符。
Surrogate(代替)	:Cs	匹配于代替对的一半。
Other private-use（其他专用情况）	:Co	匹配于来自专用区域的任意字符。
Other not assigned（其他无符号情况）	:Cn	匹配于未映射到 Unicode 字符的字符。

其他属性

除了标准的 Unicode 字符属性外，下面各项属性可能被指定作为字符组的一部分。

表达式	语法	说明
Alpha	:Al	匹配于任意一个字符。例如，:Alhe 匹配于如“The”、“then”和“reached”等个字。
Numeric（数字）	:Nu	匹配于任意一个数字。
Punctuation（标点符号）	:Pu	匹配于任意一个标点符号，如?、 @、' 等。
White space（空格）	:Wh	匹配于空格的所有类型，包括印刷和表意文字中的空格。
Bidi	:Bi	匹配于从右到左脚本的字符，如阿拉伯语和希伯来语。
Hangul（朝鲜语字符）	:Ha	匹配于朝鲜语字符和组合字母。
Hiragana（平假名）	:Hi	匹配于平假名字符。
Katakana（片假名）	:Ka	匹配于片假名字符。
Ideographic/Han/Kanji（表意字符/汉字/日本汉字）	:Id	匹配于如汉字或日本汉字等表意字符。

另外，请参考：

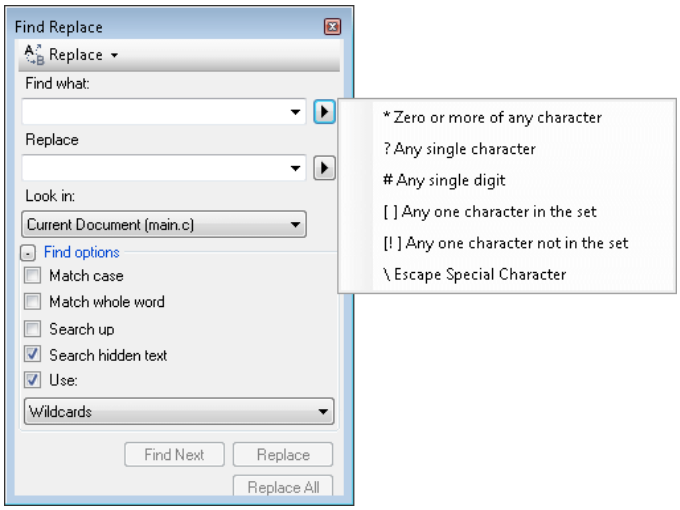
- 通配符
- 查找与替换
- 批量查找
- 批量替换

通配符

通过下面的表达式，可以替换 Find and Replace 窗口中“Find what”字段的字符或数字。

当进行查找和替换操作时，为了能够在 **Find what** 字段中使用正则表达式，请在 **Find Options** 下依次选择 **Use > Wildcards**。

点击**Find what**字段旁边的三角按键会显示可用的通配符列表。从参考列表中选择任何项目时，都会将其插入到**Find what**字符串中。



用于查找和替换的通配符：

下面显的是可用于参考列表的通配符。

表达式	语法	说明
Any single character（任意单字符）	?	匹配于任意单字符。
Any single digit（任意单数字）	#	匹配于任意单数字。例如，7#匹配于 7 后面为另一个数字的数值（如 71），但不匹配 17。
Characters not in set（非字符组中的字符）	[!]	匹配于不属于字符组中指定的任意字符。
Escape（逃逸）	\	匹配于一个作为文字跟随反斜线（\）的字符。这样，您可以查找到通配符表达式中使用的字符，如*和#。
One or more characters（一个或多个字符）	*	匹配于任意一个或多个字符。例如，new*匹配于包含“new”的任何文本，如 newfile.txt。

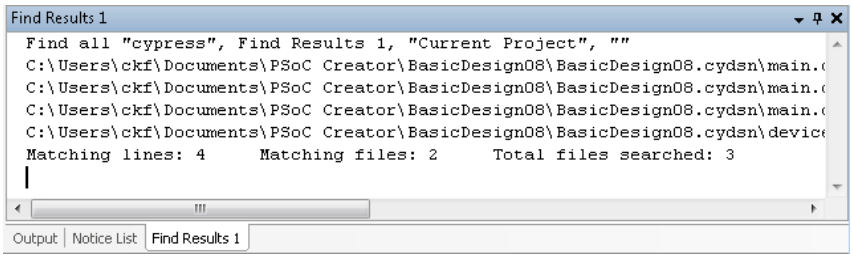
表达式	语法	说明
Set of characters（字符组）	[ ]	匹配于字符组中指定的任意字符。

另外，请参考：

- 正则表达式
- 查找与替换
- 批量查找
- 批量替换

Find Results（查找结果）

当使用 Batch Find 和 Batch Replace 对话框时，Find Results 窗口将显示所有查找到的匹配项。



共有两个 Find Results 窗口。通过 **Result（结果）选项**，您可以选择将列出匹配结果的 Find Results 窗口。当找到匹配项时，将自动打开选定的 Find Results 窗口。

要手动显示 Find Results 窗口：

请从 **View** 菜单中选择 **Find Results**，然后选择 **Find Results 1** 或 **Find Results 2**。

要想选择匹配：

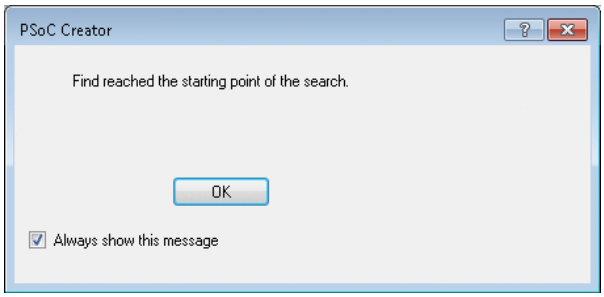
请双击结果列表中的任意行。源文件显示在文本编辑器中，匹配文本的起始位置处有一个插入点。编辑器的指示区出现一个符号，用于标志包含匹配的行；状态栏将显示它的完整文本。

另外，请参考：

- 文本编辑器
- 批量查找
- 批量替换

Search Result （搜索结果）

Search Result 对话框显示搜索结果的信息，它是 Find 的一部分。



仅在选择 Text Editor 选项的子选项时，才会显示该对话框。

通过取消选择 **Always show this message** 选框，可以禁用该对话框。

该对话框可能显示的信息包括：

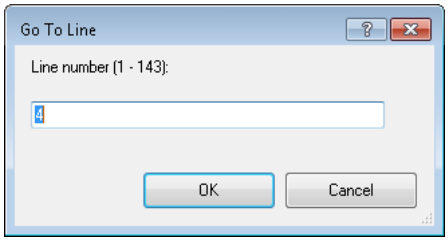
- 未找到下面指定的文本：xxxx
- 返回到搜索的起始点。
- 在指定文件中未找到其它结果。
- 已代替#。

另外，请参考：

- 文本编辑器
- 查找与替换
- 文本编辑器选项

Go To Line（跳转到行编号）

Go To Line 对话框用于跳转到特定的代码行。



要打开该对话框：

请按下**[Ctrl]** + **[G]**或从 **Edit** 菜单中选择 **Go To** 项。

要跳转到特定的某一行：

请输入行编号，然后点击 **OK**。

光标将转移到指定的行编号。

另外，请参考：

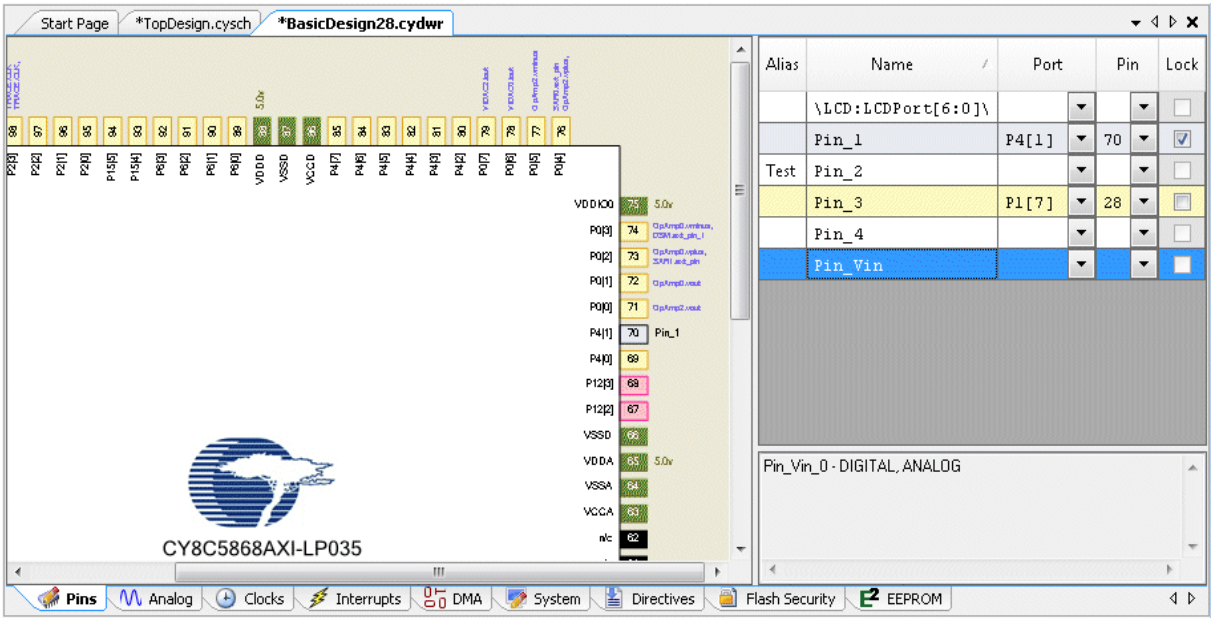
- 文本编辑器

Design-Wide Resources（设计范围资源）

Design-Wide Resources（设计范围资源）

通过 PSoC Creator 的 Design-Wide Resources（DWR）系统，可以管理设计中的所有资源，如引脚、时钟、中断、DMA 等等。每个新的设计项目都会提供默认的 Design-wide Resources 文件（.cydwr），它的名称与项目名一样。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



只有设计项目才有“.cydwr”文件，并且每个设计项目只有一个文件。对 DWR 信息进行的所有更改均被保存在该文件中。该设计级信息使用特定方式进行保存，使其能够在各器件间转移。

注意：选择不同的器件的过程中，如果继续执行该选择将发生错误，在更改器件选择前，您会收到提示信息。这样，您可以取消器件的更改过程，或继续进行。如果继续进行，将生成相应的错误。

要打开“.cydwr”文件：

请双击 Workspace Explorer 中 **Source** 选项卡下的文件。

在工作区内，该文件显示为选项卡式文档，这样便于您访问项目内的各种设计范围资源。默认情况下，引脚编辑器（**Pins** 选项卡）显示在最上方。

选择资源编辑器：

通过页面选项卡选择相应的编辑器。选择后，将显示相应的资源编辑器。

通过点击相应的选项卡，可以切换不同的资源，但是每次您只能编辑一个资源。

要向设计项目中添加“.cydwr”文件：

您无法从 PSoC Creator 中直接复制或剪切“.cydwr”文件；但您可以将该文件添加到其他设计项目内，作为现有文件使用。您也可以向设计项目中添加一个新的“.cydwr”文件。

- **Add Existing Item**（添加现有项目） — 向设计项目添加某个现有的“.cydwr”文件。该文件从选定的位置被复制到设计项目的文件夹内。它将被重命名，以匹配于项目名。
- **Add New Item**（添加新项目） — 向设计项目添加一个新的“.cydwr”文件。如果已经存在某个文件，将显示一条信息提示您是否要覆盖该文件。

要想删除/排除“.cydwr”文件：

您可以从您的项目中删除或排除“.cydwr”文件。如果项目中没有任何“.cydwr”文件，那么您只能使用默认值，并且不能编辑这些值。

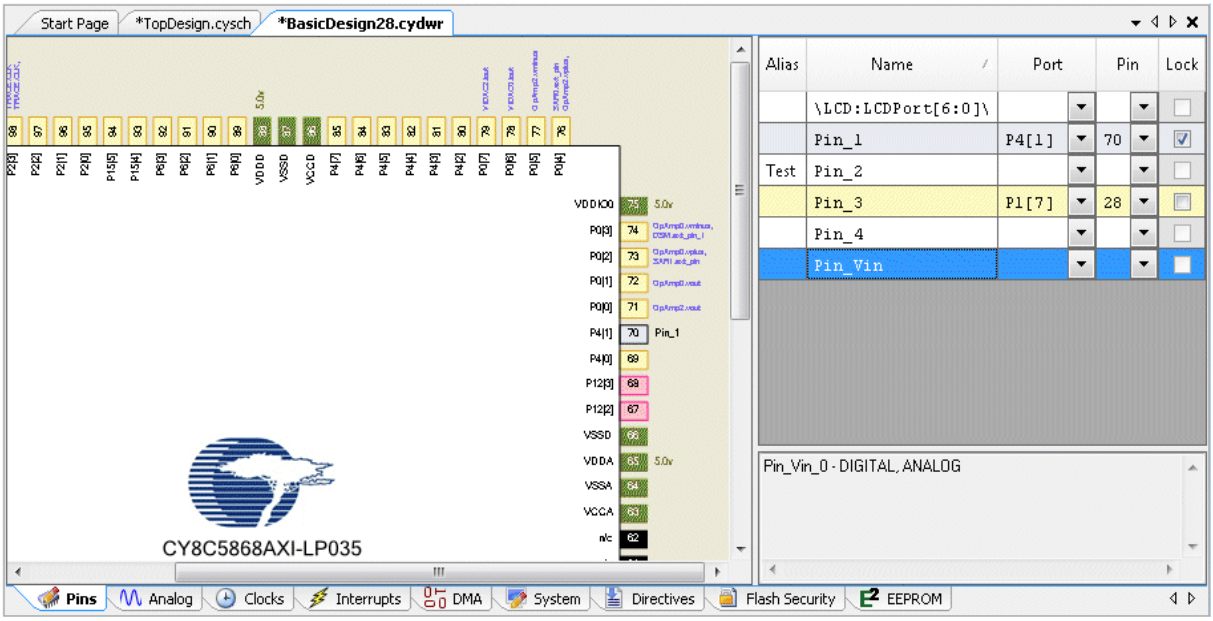
另外，请参考：

- 引脚编辑器
- 模拟路由编辑器
- 时钟编辑器
- 中断编辑器
- DMA 编辑器
- 系统编辑器
- 指令编辑器
- Flash 安全编辑器
- EEPROM 编辑器

Pin Editor（引脚编辑器）

通过引脚编辑器，您可以在 PSoC Creator 的构建过程中执行放置和布线操作前，手动指定和/或锁存器件中的引脚。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



锁定的引脚被限定在指定的引脚位置。在构建过程中，所有未分配和未锁定的引脚都将被分配。

引脚编辑器包含选定器件的交互图片和设计中的可用信号表。

要想打开引脚编辑器：

在 Workspace Explorer（工作区浏览器）的 **Source**（源）选项卡中双击“.cydwr”文件。

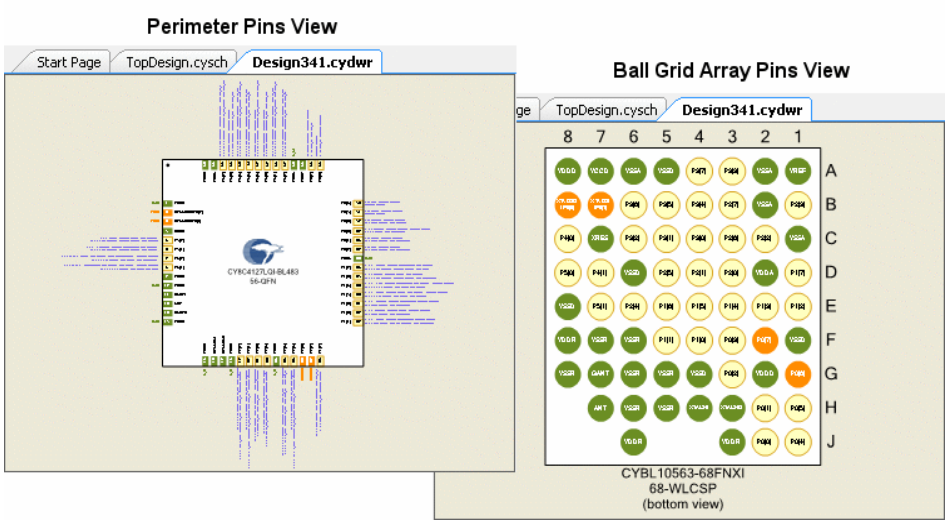
在工作区内，该文件将显示为选项卡式文档，这样您便能够访问项目中的各种设计范围资源。默认情况下，引脚编辑器（**Pins** 选项卡）在显示在最上方。如果显示的是其他编辑器，请点击 **Pins** 选项卡，将它置于最上方。

器件图片：

选定的器件图片显示了各种不同的引脚和端口。每个引脚均具有相应的引脚编号。每个引脚的功能（如 Vcc、未连接，等等）或端口名称都显示在引脚的旁边或内部。将鼠标悬停在一个特定的引脚上，这时将显示该引脚的所有功能。

外围与球栅阵列

对于 TQFP、QFN 和 SSOP/SOIC 器件，引脚编辑器会通过使用外围视图来显示引脚。对于 BGA/CSP 器件，它会使用球栅阵列视图来显示各引脚。下图显示的是每个视图的示例。



对于这两个视图，它们的引脚颜色、端口/引脚编号以及标签都是相同的。同时，分配和锁定引脚的过程也完全一样。但是，球栅阵列视图不包括有关器件外侧图像的描述，而外围视图则包含了这些内容。另外对于外围视图，任意已分配的信号名称都将显示在引脚旁边，而对于球栅阵列视图则不一样。

引脚的颜色和形式

引脚的表达格式有助于识别引脚的特性，如下表所示：

引脚的表达格式	说明
文字颜色	黑底白字表示未连接引脚。
文字颜色	深绿色底白色字表示电源引脚。
文字颜色	橘底白字表示保留引脚。在引脚旁边用橘色显示了保留引脚的原因（例如，用于调试、用于外部晶振等等）。
文字颜色	无边框的粉底黑字表示未分配的 SIO 端口引脚。如果带有黑色边框，则表示已分配的未锁定引脚
文字颜色	无边框的黄底黑字表示未分配的端口引脚。如果带有黑色边框，则表示已分配的未锁定引脚。
文字颜色	带边框的灰底黑字表示有效的分配和锁定端口引脚。
文字颜色	黄底黑字表示鼠标在端口引脚上面悬停。同一端口中的所有引脚都以这种格式显示。 该颜色也表示在表格中选中的引脚。
文字颜色	粉底黑字表示鼠标在 SIO 端口引脚上面悬停。同一端口中的所有 SIO 引脚都以这种格式显示。
文字颜色（或文字颜色）	红底黑字或白字表示无效的引脚分配。黑色文字表示如果重分配给其它端口，该引脚可能有效；白色文字表示该引脚为未连接引脚或电源引脚。实现某些引脚分配后又切换到选定的引脚时，可能会出现这种状态。每次发生无效分配时，都会向注意列表添加一个错误。
文字颜色	拖动某个信号分配给引脚时，将显示为淡绿色底黑色字。如果鼠标悬停在有效的引脚分配上，则被指定的引脚将使用这种格式。
	引脚（或引脚组）中的符号 X 表示您正在进行的分配会取消现有的分配。
	已分配给信号的引脚带有黑色的边框。

信号表：

信号表包括以下列：

列	说明
别名	若适用，该列将显示引脚的备用名称。
名称	为引脚定义的信号名称。
端口	以端口格式显示的器件引脚。通过选择该字段的下拉列表选择所需端口[引脚]，也可以实现引脚分配操作。空白的单元格表示未分配。
引脚	分配给该信号的器件引脚编号。通过选择该字段的下拉列表选择所需引脚编号也可以实现引脚分配。空白的单元格表示未分配。
锁存	指定信号的分配是否被锁定（例如，在构建期间不能移动）。只能编辑已分配的引脚。

信号的显示具有下面几种状态：

- 已分配和锁定 — 已分配且锁定给某特定引脚的信号在该表格中以灰色显示。
- 已分配和未锁定 — 已分配但未锁定的信号在表格中以黄色显示。
- 未分配 — 未分配的信号以白色显示。

如果进行了非法分配，该信号则在**引脚**列中将出现出错图标。

注意：双击该表中的某一列将显示包含了相应引脚组件的设计，同时会打开其配置对话框。

说明：

该表下面的说明区显示的是选定信号的名称和类型（数字、模拟）。

分配某个引脚：

请按照下面的方式分配该引脚：

- 点击信号表中的某个信号或点击器件中某个已被分配的引脚，并将其拖放到器件图像中所需的位置。

  注意：拖动某个引脚时，光标将表示当前位置是否有效。如果分配了某个无效的引脚，则状态栏上将显示无效的原因。

- 从信号表**引脚**列的下拉菜单中，选择某个分配。您也可以在该字段中输入引脚分配。输入时，会根据当前输入的内容，将显示合法的分配。输入值的格式为：
  - P#[#] — 指定一个地址。其中，第一个#表示端口编号，第二个#表示端口的偏移量。
  - P#[#:#] — 指定地址范围。其中，第一个#表示端口编号，第二个#表示信号的 MSB 所在的端口偏移量，最后一个#表示信号的 LSB 所在的端口偏移量。
  - 可以使用上面两个格式的组合（用逗号分隔）指定地址范围。

取消分配引脚：

请按照下面介绍的方式取消分配引脚：

- 右键点击器件图像中的分配引脚，并选择 **Unassign <signal>**（取消分配<信号>）。
- 从信号表**引脚**列的下拉菜单中，选择空行。

取消分配所有引脚：

请右键点击引脚编辑器中器件图像上任何位置，然后选择 **Unassign All**（取消分配所有引脚）。

锁存引脚：

请按照下面方式进行锁存引脚：

- 手动分配引脚时，它被默认锁存。
- 如果已经分配了引脚但尚未被锁定，请右键点击器件图片中的引脚，并选择 **Lock <signal>**（锁存<信号>）或勾选表格中所需信号的 **Lock**（锁存）选框。

右键点击菜单，**Lock All**（锁存所有）项将锁存所有已分配的引脚。

要取消锁存引脚：

请右键点击器件图片中的引脚，并选择 **Unlock <signal>**（取消选择<信号>）或取消选择表格中所需信号的 **Lock** 选框。

右键点击菜单，选择 **Unlock All**（取消锁存所有）项将会取消锁存所有被锁定的引脚。

滚动、平移或放大操作：

如果器件图片过大，则通过滚动栏，您可以查看器件的其它区域。需要适当使用这些操作：

- 要想自动滚动，将信号拖动到器件图片的边缘。
- 使用鼠标滚轮向上/下移动；按下**[Shift]**并滚动鼠标滚轮可实现向左/右移动。
- 要想平移图片，请按下组合键**[Alt]** + 左键点击，并拖动。
- 要想放大图片，请按下**[Ctrl]**然后滚动鼠标滚轮。

另外，请参考：

- 映射器、放置器、路由器
- Design-Wide Resources（设计范围资源）
- 引脚的使用

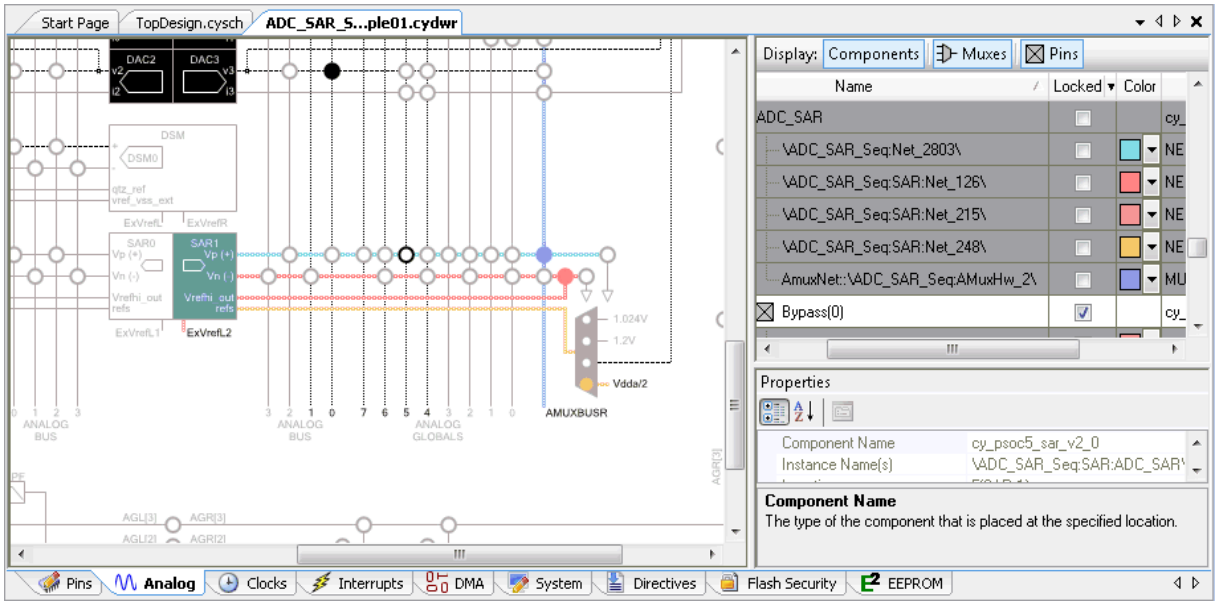
模拟路由编辑器

模拟路由编辑器

模拟路由编辑器提供了包含 PSoC 器件与特定设计的放置和路由结果的互联视图。通过该编辑器，您还可以手动进行放置和路由，同时锁存所有或某些结果。

该编辑器有两个单独的运行模式：

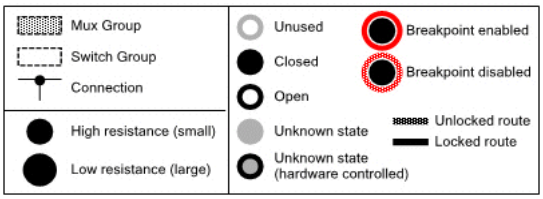
- **设计模式** — 显示设计模式下的编译结果（它是工具的默认模式）
- **调试模式** — 调试过程中可以查看器件当前的状态（请参阅模拟路由编辑器的调试）



模拟路由编辑器包括三个主要部分：模拟互联图、设计信息表以及属性。

模拟互联图：

模拟互联图显示的是如何布置器件的模拟组件、连线、开关和引脚，以及它们间如何连接。未使用的资源以灰色显示。如果可用的资源若未被选择，这些资源将显示为黑色。如果可用的组件或引脚被选中，它将显示为蓝色。选定的路由和开关以表格中选择的颜色显示。下图显示的是互联图的图标。



如果您将鼠标悬停在使用资源上，工具提示将显示相关信息。在选定资源时，将在 **Properties** 区域中显示同样的信息。

Wires（连线）

可以锁定连线（编译过程不应移动连线）。该连线包括 MARS 组件和控制文件锁定的资源（就算通过编辑器仍可以覆盖这些位置）。锁定的线是实线;未锁定的线是虚线。

模拟路由编辑器中锁定的线也可以被解锁。必须通过这些源解锁由其他源锁定的线。

开关

开关被显示为圆圈。填充了颜色的圆圈表示开关被关闭。白色圆圈表示开关被打开。灰色表示状态未知。

- 在设计模式下，**Properties** 部分中复用器的选框用于控制运行中可变开关的显示。
- 在调试模式下，状态反应的是实际寄存器的值，可使用 GUI 进行编辑。如果带有开关的复用器由硬件控制，在调试过程中，它显示为灰色。

注意：对软件控制的模拟复用器进行 DMA 访问时，调速器可能显示错误状态。

开关断点由包围着开关的虚线表示。如果断点被启用，则虚线为红色，否则，它显示为琥珀色。

器件中的开关组由圆点虚线框包围起来显示。这些组允许要么只有一个活动终端，要么有任意多个活动终端。在具有多个活动终端中，用灰色填充边框来表示。

引脚与组件

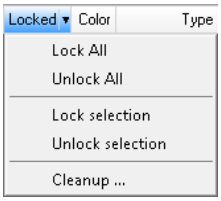
锁定的引脚和组件具有一个小的挂锁图标。

设计中数字引脚被显示为黑底白字，提示则显示为青色。但并没有显示数字引脚的路由信息。

设计信息表：

包括以下各列。

- 名称**是指实例名称。**
- **锁存**是一个复选框，通过它可强迫项目使用相应资源。使用 **Lock** 列的下拉菜单，可以锁存或解锁所有列出的项目，如下所示：



  - **Lock All**（锁存所有） — 锁存设计中的所有路由线。
  - **Unlock All**（解锁所有） — 对设计中的所有路由线进行解锁。
  - **Lock selection**（锁存选定项目） — 对选定路由线进行锁存/解锁。锁定的路由线为实线;未锁定的路由线为虚线。
  - **Unlock selection**（解锁选定项目） — 将选定线标志为未使用。请参阅路由编辑。
  - **Cleanup...**（清理） — 打开 Locked Route Cleanup（锁定路由线清理）对话框，移除不再适用于当前设计的锁定网信息。

- **颜色**，通过该项用户可以通过每个路由资源的下拉菜单选择在互联面板中显示的颜色。注意，每个资源通常多次显示（例如，当网络将引脚连接到顶层原理图的模拟资源时），所以需要后台修改资源中所有地方的颜色。
- **类型**是指组件名称（如“PGA_v1_70”）或资源类型（如“MUX”或“NET”）。

该表的条目按实例名称的字母顺序排列，只能通过该列进行修改条目的顺序。

每个条目（组件、引脚和复用器）均能被展开/收缩，以显示组件层次的更低层次以及它们所连接的资源。

表格的顶部具有用于使能或禁用查看选定项目的按键：**Components**（组件）、**Muxes**（复用器）以及 **Pins**（引脚）。

Properties（属性）：

Properties 区显示的是基于表格或框图中选择的信息。如果您选择了多个项目，该区域并不会显示所有信息。选择引脚/组件或选择复用器时，该区域显示的信息会不一样。

组件/引脚视图

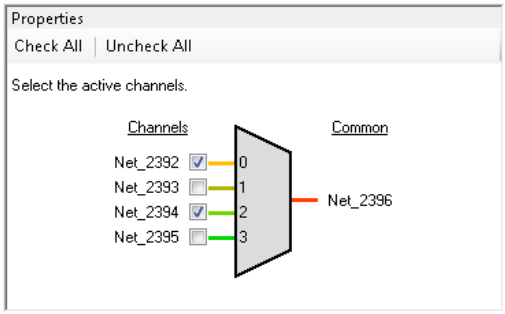
对于组件和引脚，该区域会根据所选项目显示各种只读属性。可显示的属性包括：

- 网络名称（仅适用于网络类型）
  如果在原理图中已经命名了某条线，则工具将显示其名称。如果未命名原理图中的线，工具将显示机械生成的网络名称。
- 驱动模式（仅适用于引脚）
- 组件名称
- 资源名称
- 锁存状态

使用面板上的按键，按字母的前后顺序或按类别对属性进行排序。

复用器视图

对于复用器，**Properties** 区域显示选定复用器的可编辑图像，用于选择互联图和欧姆表中的活动通道。如果目标是一个差分复用器，那么激活某个通道的同时也会对连接产生影响。



框图中具有选择复用器的活动通道的复选框。它还显示了与每个通道相应的网络名称和通用终端。每个网络的颜色同表中的选择相同。当您选择或取消选择通道，更改的内容将反映在互联图和欧姆表中。

使用面板上的 **Check All**（勾选所有）和 **Uncheck All**（取消勾选所有）按键，可分别选择或取消选择所有通道。

注意：如果将 **AtMostOneActive** 参数设置为“真”，那么您只能选择一个通道，**Check All** 按键将被禁用。

另外，请参考：

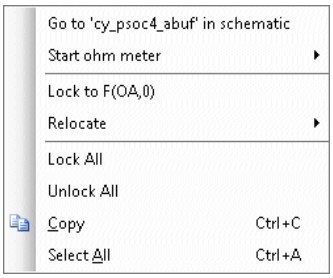
- 模拟路由编辑器的右键菜单
- 欧姆表

- 手动放置
- 路由编辑
- 模拟路由编辑器的调试

模拟路由编辑器的右键菜单

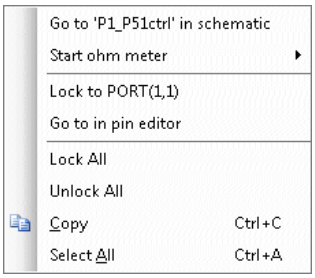
模拟路由编辑器的右键菜单（右键点击时出现的菜单）包含了各种指令。根据您右键点击的位置（引脚、组件或信号），可用的指令会存在差异。您可以右键点击互联图和表中的项目。下面是可用的指令：

对于组件：



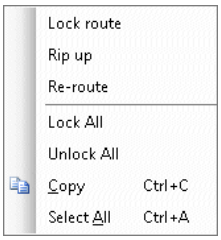
- **Go to <Component> in schematic**（转到原理图中的<组件>） — 打开原理图编辑器中的选定组件。
- **Start Ohm meter**（启动欧姆表） — 打开选定信号的欧姆表。
- **Lock to/Unlock From <Location>**（锁存组件到<地址>/取消锁存组件到<地址>） — 锁存/取消锁存组件到某个地址。
- **Relocate**（重定位） — 允许您将选定的基本组件转移到其他地址上。如果没有有效的地址，将禁用该选项。请参考手动放置。
- **Lock All**（锁存所有） — 锁存设计中的所有路由线。
- **Unlock All**（解锁所有） —解锁设计中的所有路由线。
- **Copy**（复制） — 在相应的编辑器中，将互联图复制到您能够粘贴位图的文件内。
- **Select All**（选择所有） — 选择互联图中的所有对象。

对于引脚：



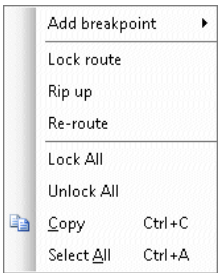
- **Go to <Pin> in schematic**（转到原理图中的<引脚>） — 打开原理图编辑器中选定的引脚。
- **Start Ohm meter**（启动欧姆表） — 打开选定引脚的欧姆表。
- **Lock to/Unlock From <Location>**（锁存引脚到<地址>/取消锁存引脚到<地址>） — 锁存/取消锁存引脚到某个现有的地址。
- **Go to in pin editor**（转到引脚编辑器） — 打开引脚编辑器并选择相同的引脚。
- **Lock All**（锁存所有） — 锁存设计中的所有路由线。
- **Unlock All**（解锁所有） —解锁设计中的所有路由线。
- **Copy**（复制） — 在相应的编辑器中，将互联图复制到您能够粘贴位图的文件内。
- **Select All**（选择所有） — 选择互联图中的所有对象。

对于连线：



- **Lock route**（锁存路由线） — 对选定的路由线进行锁存/解锁操作。锁定的路由线为实线;未锁定的路由线为虚线。
- **Rip up**（取消选定项目） — 将选定的线标志为未使用。请参阅路由编辑。
- **重新路由** — 将模拟路由编辑器切换到手动路由模式。请参阅路由编辑。
- **Remove obsolete "Entire Net" route data**（移除整个网络中已过时的路由数据） — 移除在修改原理图时被锁定作为整个网络一部分的网络。
- **Lock All**（锁存所有） — 锁存设计中的所有路由线。
- **Unlock All**（解锁所有） — 解锁设计中的所有路由线。
- **Copy（复制）** — 在相应的编辑器中，将互联图复制到您能够粘贴位图的文件内。
- **Select All**（选择所有） — 选择互联图中的所有对象。

对于开关：



- **Add/Edit breakpoint**（添加/编辑断点） — 用于添加或编辑断点。请参阅模拟路由编辑器的调试。
- **Lock route**（锁存路由线） — 对选定的路由线进行锁存/解锁。锁定的路由线为实线;未锁定的路由线为虚线。
- **Rip up**（取消选定项目） — 将选定的线标志为未使用。请参阅路由编辑。
- **Re-route**（重新路由） — 将模拟路由编辑器切换到手动路由模式。请参阅路由编辑。
- **Lock All**（锁存所有） — 锁存设计中的所有路由线。
- **Unlock All**（解锁所有） —解锁设计中的所有路由线。
- **Copy**（复制） — 在相应的编辑器中，将互联图复制到您能够粘贴位图的文件内。
- **Select All**（选择所有） — 选择互联图中的所有对象。

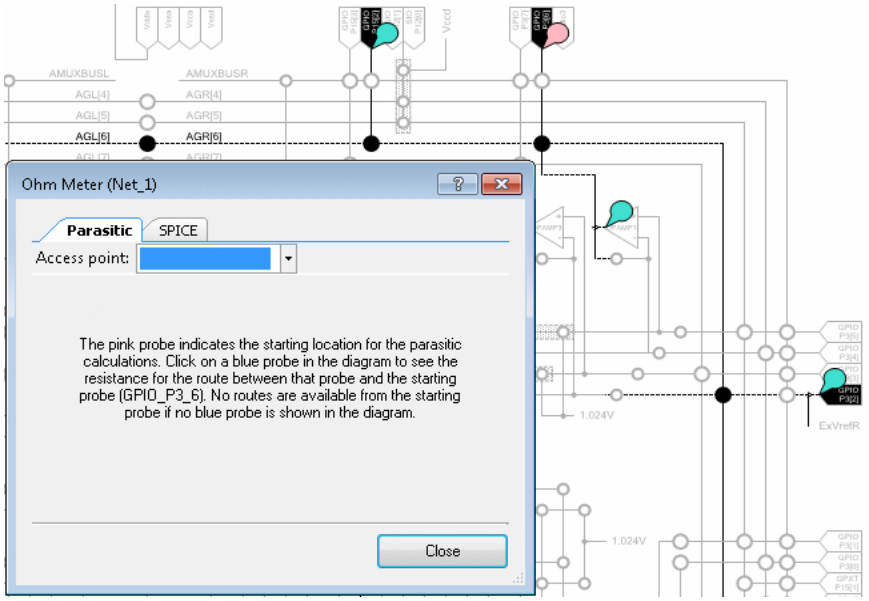
另外，请参考：

- 模拟路由编辑器
- 原理图编辑器
- 引脚编辑器
- 欧姆表
- 手动放置
- 路由编辑

- 模拟路由编辑器的调试

欧姆表

欧姆表显示模拟路由编辑器的互联视图中两点间（引脚和/或组件，并非为连线或开关）的电容。通过该对话框，您可以更改图中的探头点并查看寄生电容。该对话框还包含 SPICE 的路由数据。



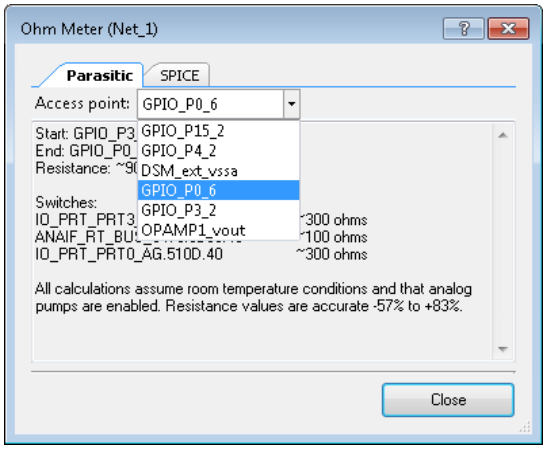
互联图中的探头与引脚相连。第一个放置的引脚以粉色显示，将其放置后，所有合法的目标探头点会自动显示为蓝色。选择某个蓝色引脚，查看它的布线电容。重新选择探头起始点会清除先前的探头、清除对话框结果，并刷新合法的目标（蓝色）探头。

要打开欧姆表，请执行下述操作：

右键点击图中或表中的引脚或组件，然后选择 **Start Ohm Meter >**。这时会指向起始的访问点。

Parasitic（寄生电容）选项卡：

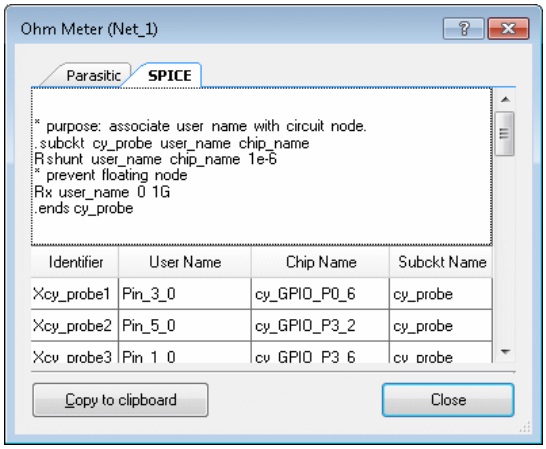
该选项卡会显示在点击图中的蓝色引脚时两点之间的电容。它显示了路由的总电容和每个开关的电容。



通过 **Access Point**（访问点）的下拉菜单，您可以点击其它蓝色引脚，更改第二个选定的引脚。

SPICE 选项卡：

该选项卡显示了路由的 SPICE 网络列表。它是一个只读列表。



使用 **Copy to Clipboard** 按键将数据复制到选定的仿真器。

另外，请参考：

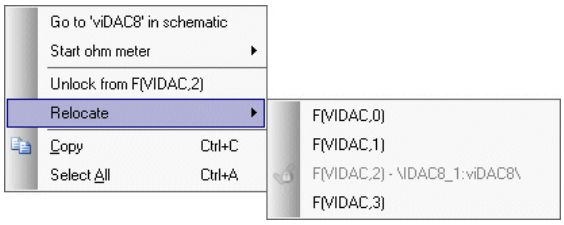
- 模拟路由编辑器

手动放置

通过模拟路由编辑器的手动放置功能，您可以指定执行编译时放置模拟组件的位置。您不能使用该功能放置引脚，需要使用引脚编辑器才能放置引脚。

要想手动放置组件：

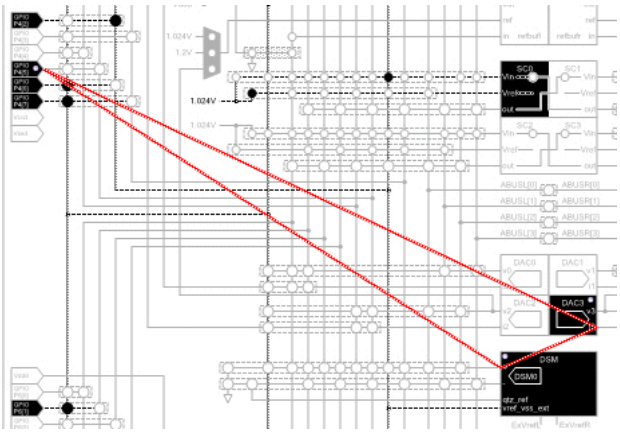
右键点击组件，选择 **Relocate >**。然后指向所需位置。该菜单仅适用于有效的移动。如果并不存在有效的移动，则该菜单项将被禁用。



您可能会将组件放置到当前已被占用的位置。通过随后的编译步骤，现有的组件将被放置在新的位置上。

信号路由

选中新的位置后，将暂时重绘制路由信号，它还被称为鼠迹网。仅在选定了路由时，才会显示该项。



在编译过程中，路由会根据分配的信号被更新。

资源锁存

所有组件的移动都会锁定资源。组件显示为带有一个锁定符号。

要解锁资源，请右键点击组件并选择 **Unlock from <location>**。

如果您在编译前解锁了资源，则组件将被返回到原始位置。

如果您在编译后对资源解锁，将保持该图标，直到您进行其它编译。这是因为编辑器认为该锁存位于外部源。

另外，请参考：

- 模拟路由编辑器
- 引脚编辑器
- 右键菜单
- 路由编辑

路由编辑

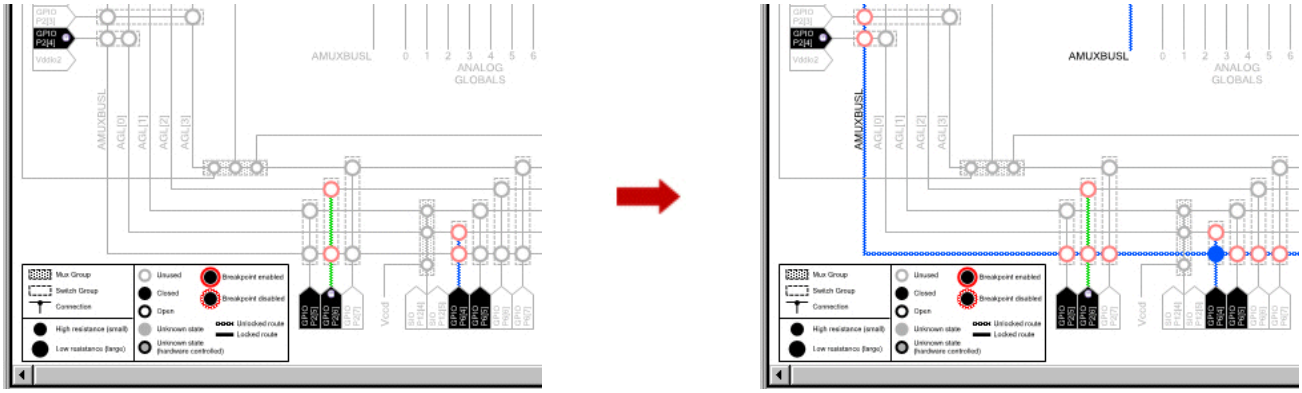
路由编辑包括取消路由和重新路由信号，以及手动选择路由信号的开关。右键菜单包括各种用于路由编辑的元素。

剥离

- **Nets**（网络） — 当网络被剥离时，路由的连线和开关被标志为未使用
- **Net-Ties/Net-Joins** —当选中 Net-Tie 或 Net-Join 项时，您可以剥离整个网络或构成整个网络的子网。任何子网的剥离与整个网络的剥离是一样的。剥离整个网络会使所有单独的子网和执行 net-tie 与 net-join 组件被剥离。
- **Muxes**（复用器） — 复用器剥离的应用范围为整个复用器。当对复用器进行剥离时，用于路由复用器的所有连线和开关均被标志为未使用。

重新路由

在选中 **Re-Route** 指令时，模拟路由编辑器将进入手动编辑模式。在该模式下，您可以点击开关从而打开或关闭它，以实现手动路由信号。可用于路由的开关以默认的颜色被高亮显示；不可用的开关显示为灰色。当您选择某个开关进行路由信号时，其它开关也变为可选状态。



打开状态的开关显示为空心圆圈；关闭状态的开关显示为实心圆圈。

编辑时，状态栏将显示正在进行的操作。

两个可用的按键分别是：**Commit Edit**（保存编辑）或 **Cancel Edit**（取消编辑）。“Commit”按键将保存并锁定路由；“Cancel”按键将模拟路由编辑器返回到编辑前的状态。

复用器的路由

编辑某个复用器的分支同编辑信号网络相类似，它们的差异是：

- 要连接的端点是网络，并非引脚。
- 在开始编辑复用器时，复用器所使用的全部资源（用于所有 arm）被临时标志为未使用，用户可以按最大的灵活性重新使用与其它分支共享的资源。

Net-Ties 和 Net-Joins

net-ties 和 net-joins 的编辑类似于复用器的编辑该编辑会影响任意数量的子网，并非结合一对单网。

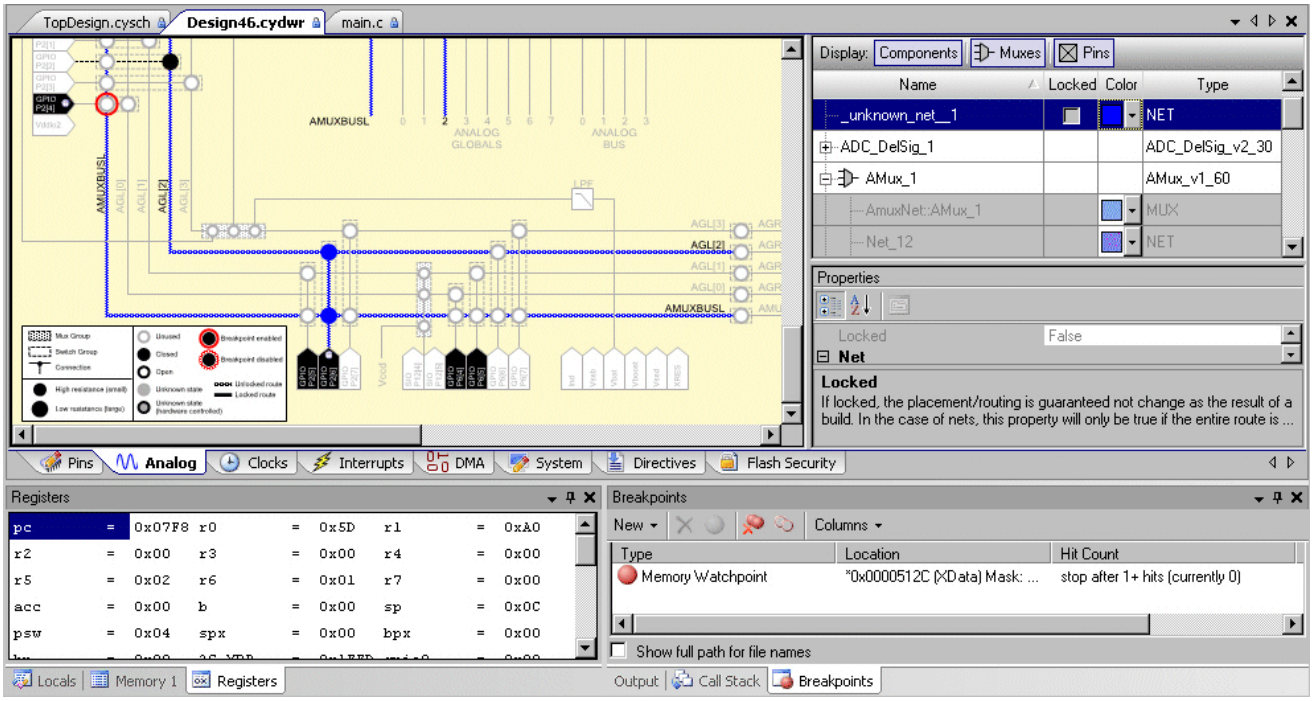
- 要连接的端点是网络，并非引脚（同复用器一样）。
- 由于没有共享的资源，所以需要单独执行剥离和编辑操作。

另外，请参考：

- 模拟路由编辑器
- 模拟路由编辑器的右键菜单
- 路由编辑

模拟路由编辑器的调试

通过模拟路由编辑器中的调试视图，用户可以查看/修改芯片上模拟路由的当前状态。开始进行调试时，调试视图会自动变为活动状态。在调试模式下，互联图的背景为黄色，右下方的标记表示它处于调试模式。



当调试会话停止时，模拟路由编辑器会自动返回设计视图。

打开/关闭开关：

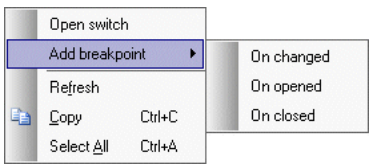
在调试视图中，您可以使用右键菜单来打开或关闭任何非 DSI 控制的开关。由 DSI 控制的开关用中心为灰色的圆圈显示，表示“使用”状态；但 PSoC Creator 不知道开关的当前状态。

注意：即使 CPU 不运行，但仍能够更改模拟路由编辑器的状态，因此其状态可能已经超时。右键点击互联图并选择 Refresh，**以更新所有可见的调试窗口。**

开关断点：

通过片上地址断点，能够执行各个断点。断点应用于整个寄存器，用于控制多个开关和无关元素。调试器会确定是否发生所需的开关断点，并忽略（继续执行）其它变化。

要想设置某个断点，请右键点击开关。您可以将执行断点的条件设置为：发生变化、打开或关闭。



一旦选中了断点，该断点将显示在存储器观察点窗口中。

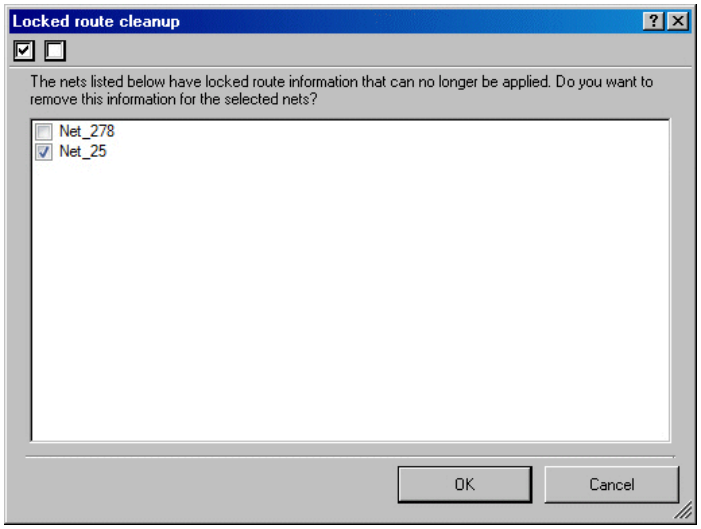
您可以在一个信号寄存器中设置来自同一个断点资源的多个断点。如果已经达到最大数量的断点（无论其使用情况如何），但您还想要设置一个新的断点，PSoC Creator 将提示无法设置任何断点，直到删除或禁用了另一个断点为止。

另外，请参考：

- 模拟路由编辑器
- 模拟路由编辑器的右键菜单
- 存储器观察点

锁定路由清理

通过锁定路由清理对话框，您可以清理网络中的锁定路由信息，由于这些网络不再存在于原理图中或者由于原理图变化不再锁定这些网络。您不必进行解锁操作也能够选择性的删除该类数据。



要打开该对话框，请执行下面的操作：

从模拟路由编辑器表格中 **Locked** 列的下拉菜单中选择 **Cleanup...**项。

如何使用该对话框：

选择需要移除锁定路由信息的一个或多个网络，并点击 **OK**。

您也可以使用 **Select All** 或 **Deselect All** 按键实现全选或全部取消选择。

另外，请参考：

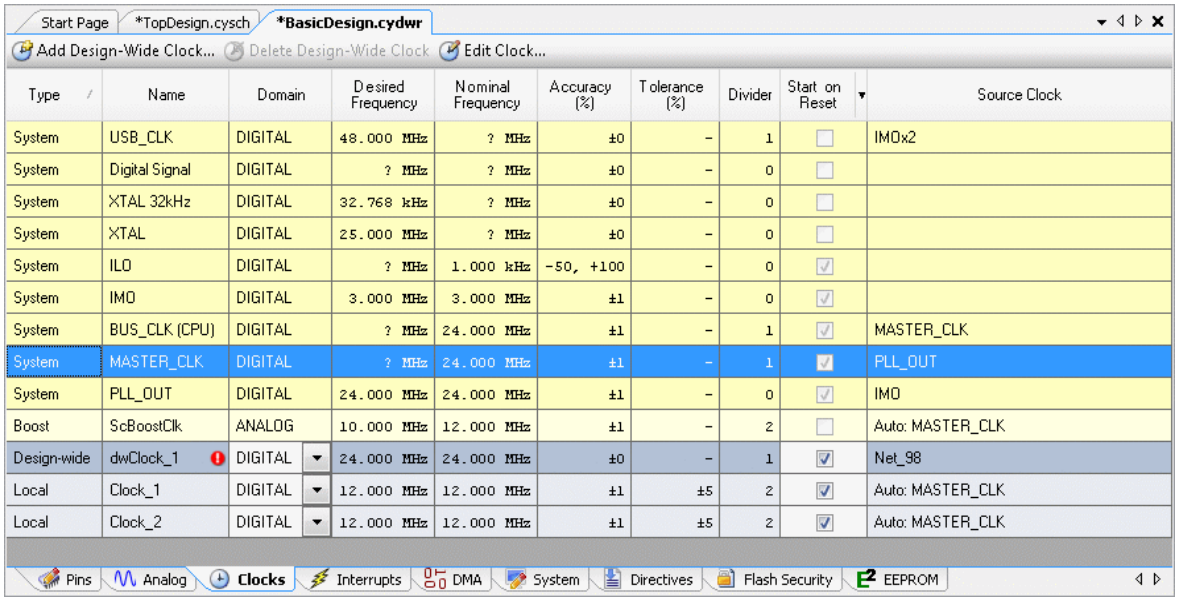
- 模拟路由编辑器

时钟编辑器

时钟编辑器

时钟编辑器是创建和编辑时钟的 Design-wide Resources 工具。使用该工具，您可以查看所有时钟、添加和删除设计范围的时钟，并编辑设计范围和系统时钟。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



要想打开时钟编辑器，请执行下面的操作：

双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

在工作区内，该文件显示为选项卡式文档，这样便于您访问项目中的各种设计范围资源。点击 **Clocks**（时钟）选项卡以访问时钟编辑器。

时钟编辑器工具栏：

工具栏包含以下指令：

- **Add Design-Wide Clock**（添加设计范围时钟） — 使用该指令，您能够将设计范围时钟添加到设计中。请参阅 Add/Edit Design-Wide Clock。
- **Delete Design-Wide Clock**（删除设计范围时钟） — 使用该指令，您可以删除任何已经添加的设计范围时钟。
- **Edit Clock**（编辑时钟） — 使用该指令，您能够编辑设计中的设计范围时钟。请参阅 Add/Edit Design-Wide Clock。

时钟列表：

时钟编辑器使用一个列表显示所有时钟。默认情况下，时钟按**类型**排列。通过点击某列的标头，您可以按递增/递减顺序排列所需列中的时钟。

注意：一般总是**按额定频率**进行排序。

列	说明
类型	时钟类型： 系统 — 在您的设计中可以使用的内部和外部时钟源 设计范围 — 在时钟编辑器中声明并能够在整个设计中共享的时钟 本地 — 通过各组件被添加到原理图中的时钟 升压 — 在系统编辑器中选定变量 Vdda 项时将显示的低电压模拟升压时钟。
名称	时钟的名称。 如果时钟处于错误状态，则图标将显示时钟的名称。鼠标悬停在该图标上时，将显示描述错误的工具提示。该错误也会显示在注意列表窗口中。
区域	模拟或数字。该信息由 PSoC Creator 决定。
所需频率	时钟的所需频率。如果未使用，将显示‘? MHz’。
额定频率	是指时钟的额定频率。该频率由使用时钟的系统确定。如果尚未计算得出，将显示‘? MHz’。
准确度	显示时钟的准确度，单位为百分比。
容差	显示的是时钟范围的容差，单位为百分比。如果未指定容差，将显示‘-’。 注意：**只能为自动时钟指定容差值。**
分频值	显示时钟的分频值。已经指定了该值，或在处理时钟时计算得到。
复位时启动	如果勾选该项，则进入 main 函数前设置时钟的操作会调用_Start()函数（该项被默认选择）。您可以为所有“新”本机时钟和所有设计范围时钟选择该项。 列标头旁边有一个系列菜单，通过该菜单您可以同时选择或取消选择所有被应用的选框。 系统时钟的选框是只读的。它的大小依赖于配置系统时钟对话框中“Start on Reset”的使能状态。“现有”时钟的字段也为只读状态。它显示了源时钟的“Start on Reset”值。
源时钟	使用它（若适用）创建时钟。未明确指定输入时钟（例如，选定<自动>项）的时钟被显示为：“Auto: Clock the solver picked”（自动：处理程序选择的时钟）。

数字和模拟时钟：

PSoC Creator 会自动识别用户创建的时钟是数字时钟还是模拟时钟。它会查看整个设计的 Fanout，如果时钟驱动的是模拟基元，它便是模拟时钟。如果时钟未连接到模拟基元，该时钟被假设为数字时钟。如果时钟连接到数字和模拟基元的混合体，则会生成 DRC 错误。

容差支持：

PSoC Creator 将支持有关 DRC 的时钟准确度和容差的强大系统。

本机和设计范围时钟均能够添加所需容差（+X%、-Y%或+X ppm、-Y ppm）。

器件的内部系统时钟显示的是它的准确度信息。器件的外部系统时钟（例如 XTAL、XTAL 32kHz 和 Dig Sig）允许您输入准确度数值（+X%、-Y%或+X ppm、-Y ppm）。默认值为+0%和-0%。

PSoC Creator 的时钟系统将参考时钟的准确度和实际得到的频率，并比较容差和所需频率。如果得到的频率/准确度小于所需范围，将生成 DRC 错误。

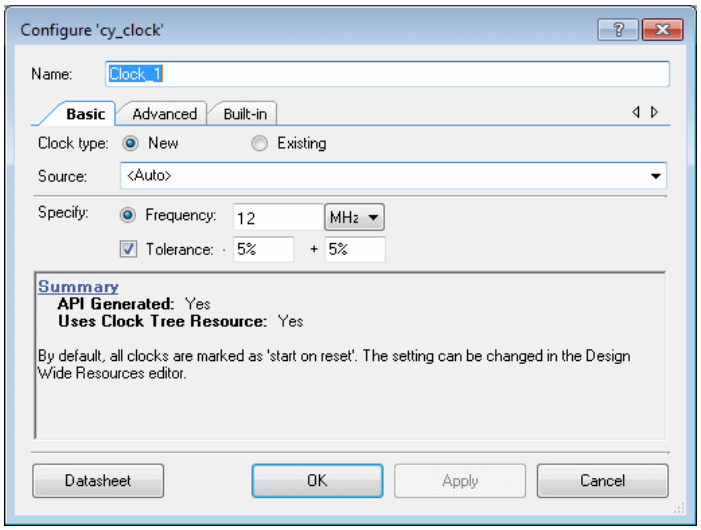
另外，请参考：

- 配置本机时钟

- 配置系统时钟
- 添加/编辑设计范围时钟
- 选择源时钟
- 时钟的使用
- 时钟组件数据手册（位于组件目录中）
- 系统参考指南
- 设计范围资源

配置本机时钟

通过配置时钟对话框，您可以配置放置在原理图中的本机时钟的各种特性。请参阅组件目录中的时钟组件数据手册。



该对话框包括下面各主要项：

- **Clock Name**（时钟名称） —可以为本机时钟键入一个名称。
- **Configuration**（配置） —指定时钟的特性。
- **Summary**（总结） — 显示有关正在创建的时钟和所使用的源时钟（如果 **Source** 项并非<Auto>）的信息。

要想创建本机时钟，请执行下面的操作：

通过拖动原理图中组件目录下的时钟组件，可以创建本机时钟。

如何打开该对话框：

双击时钟组件，以打开配置时钟对话框。

如何配置本机时钟：

根据下面的内容配置本机时钟：

1. 在 **Name** 框中，输入时钟名称或使用默认的名称。
2. 在 **Clock Type** 字段中，选择 **New** 或 **Existing** 项。新时钟将使用器件资源，并具有为之生成的 API。现有的时钟不使用硬件资源，并且没有特定的 API；它们只是已定义时钟的别名。
3. 指定 **Source** 的配置，如下所述：
   - **Clock Type: New / Source: <Auto>** — 输入 **Frequency**（频率）并选择一个 **Tolerance**（容差）范围。PSoC Creator 会指出如何执行它。
   - **Clock Type: New / Source: Specified** — 从下拉菜单选择特定的源时钟，以对其进行分频。然后输入所需频率或明确指定分频值。
     - 如果您指定了某个 **Desired Frequency**（所需频率），那么 PSoC Creator 会自动计算出分频值。
     - 如果您指定了某个 **Divider**（分频值），PSoC Creator 会直接使用该值。
   - Clock Type: Existing / Source: Specified — **为给定的源时钟创建一个别名。**

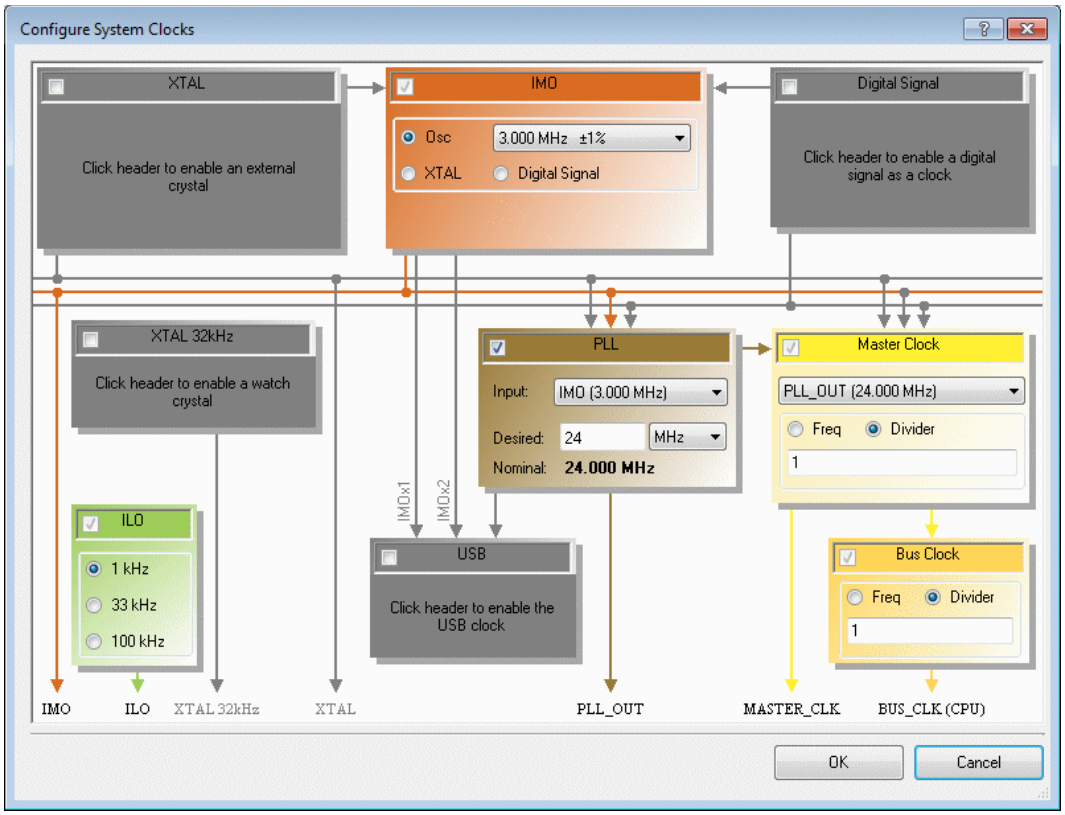
您可以使用 Design-wide Resources 时钟编辑器查看各种时钟的特性。

另外，请参考：

- 设计范围资源
- 时钟编辑器
- 时钟的使用
- 组件目录
- 时钟组件数据手册（从组件目录中打开）

配置系统时钟

配置系统时钟对话框提供的图形框图显示了您的设计中选定器件的各种系统时钟以及它们之间的关系。



通过该对话框，您可以指定系统时钟的各种特性。

- 勾选标记表示该时钟被使能。时钟被禁用时，它显示为灰色，并且它的内容被替换为介绍如何使能它的文本。有些时钟无法禁用。
- 线条表示时钟信号传输的路径。如果线条显示为灰色，则表示当前的时钟配置未使用该路径。
- 错误图标表示没有正确配置时钟。将鼠标悬停在图标上，可显示错误内容的信息。

注意：根据设计选定的器件，该对话将包含不同的时钟和选项。对于某些器件，该对话框针对高频率和低频率时钟被分为多个选项卡。

更多有关系统时钟的配置选项的信息，请参阅相应的*技术参考手册*。

数字信号：

可以将数字信号配置为设计中任意的路由数字信号。将来自数字引脚的输出作为数字信号使用时，您要确保引脚的输入是非同步的。在引脚组件配置对话框中的 **Input** 选项卡内，通过取消选择“Input Synchronized”（输入同步）选项来实现。更多信息，请参阅引脚组件数据手册。

要打开该对话框，请执行下面的操作：

在 Design-Wide Resources 的时钟编辑器中：

- 双击系统时钟，或者

- 选择一个系统时钟，并点击 **Edit Clock**。

要想使能/禁用系统时钟，请执行下面的操作：

点击相应的时钟选框，以使能或禁用它。

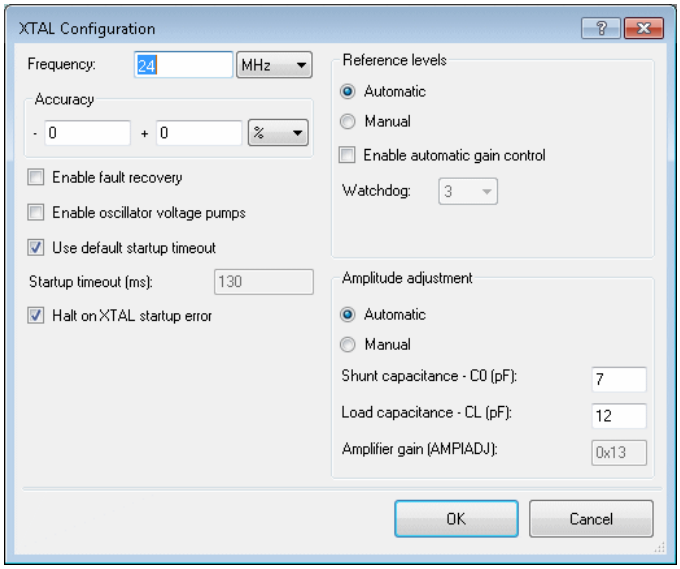
注意：有些时钟无法被禁用。它们的选框变暗。

另外，请参考：

- 设计范围资源
- 时钟编辑器
- 时钟的使用
- 时钟组件数据手册（从组件目录中打开）
- 引脚组件数据手册（从组件目录中打开）

XTAL 配置

通过 XTAL 配置对话框，可以配置外部晶振的特性。



要打开该对话框，请执行下述操作：

使能系统时钟编辑器中的 XTAL，并点击**配置**按键。

如何配置 XTAL：

选择/输入相应的字段，并点击 **OK**。

字段：

使用下面的字段根据要求配置 XTAL。

频率/准确度：

使用这些字段指定 XTAL 的频率和准确度。

Enable fault recovery（使能故障恢复性能）：

该选框可使能故障恢复性能。通过故障恢复性能，在 XTAL 被检测为准确地停止工作时，XTAL 的输出可以从 XTAL 时钟切换到 IMO。它使用 XTAL 内部的看门狗/错误信号。

注意：当该字段被使能，并且 XTAL 作为 IMO 的时钟源时，该字段将生成错误信息：“当 XTAL 也被 IMO 使用时，XTAL 时钟不能使用故障恢复性能”。

Enable oscillator voltage pumps（使能振荡器电压泵）：

勾选该选框时，可以通过 FASTCLK_XMHZ_CSR 寄存器中的位 3（xpump_dis）控制振荡器电压泵。这样可以节省功耗并降低抖动。默认情况下，没有选择该选框，因此该性能处于未使能状态。

Use default timeout（使用默认超时）：

该选框在默认情况下被选择。它指出在启动系统时使用的是默认的超时还是用户自定义的超时。

- **启动超时（ms）** —使用默认超时时，该字段是只读的，并显示默认的超时值。如果您取消选择 **Use default timeout** 选框，那么可通过该字段根据特定 XTAL 的要求指定超时。

Halt on XTAL startup error（发生 XTAL 启动错误时将暂停）：

该选框会使能 ECO 错误处理程序中的“while(1);”代码。默认情况下，该选框被使能，并在应用进入 main()函数前被暂停。如果取消选择该选框，则代码会继续运行到 main 函数（您要处理相应的时钟错误）。

注意：XTAL 不提供总线时钟时，该字段被忽略。如果发生这种情况，将不会显示任何信息。

Reference levels（参考电压）：

- **Automatic**（自动） — 默认情况下该选项被选中。通过给定的 XTAL 频率计算 **Reference levels** 的值。初始化 XTAL 时钟时，将使用这些值来设置 CFG1 寄存器。
- **Manual**（手动） — 该选项适用于高级用户。使用该选项可使能 **Feedback**（反馈）和 **Watchdog**（看门狗）字段，用于微调参考电压的配置。初始化 XTAL 时钟时，将使用 CFG1 寄存器中的值。
  - **Enable automatic gain control**（使能自动增益控制） — 通过该选框，可以使能自动增益控制（AGC）。AGC 将测量振荡幅度，并该值同参考值进行比较。如果该值过高或过低，将对 XTAL 进行内部调整，以便增加或降低幅度。这样可以降低驱动电平，满足晶振的要求。并非所有情况都要进行这样的操作。在默认情况下未选择该选框。当选择该选框时，**Watchdog** 下的 **Feedback** 字段可见。
  - **Watchdog**（看门狗） — 看门狗作为一个电路（XERR/错误检测）它会测量振荡幅度并将测得的值同参考值进行比较。如果测得的值过低，将确认被传送到状态寄存器位的错误信号。该位可通过软件检查，也将它用于控制执行“故障恢复”的复用器。启动 XTAL 时，通过看门狗可以确定振荡达到可接受幅度的时间，并能够确定是否将 XTAL 作为整个器件的时钟源使用。看门狗参考电压是与振荡幅度进行比较的电压，使用该电压可以确定振荡幅度是否可接受。
  - **Feedback**（反馈） —反馈参考电压是与振荡幅度进行比较的电压。

幅度调整：

- **Automatic**（自动） — 默认情况下该选项被选中。通过给定的 XTAL 频率可以计算**放大器增益（AMPIADJ）**字段的值。通过该选项，您可以指定晶振的 **Shunt capacitance**（并联电容）和 **Load capacitance**（负载电容）。通过这些值和 XTAL 频率，可以计算出 AMPIADJ 值。初始化 XTAL 时钟时，会使用该值进行设置 CFG0 寄存器。
- **Manual**（手动） — 该选项适用于高级用户。通过该项能够向 **Amplifier Gain（AMPIADJ）**字段手动输入指定的 AMPIADJ 值。初始化 XTAL 时钟时，将使用该值来设置 CFG0 寄存器。

另外，请参考：

- 时钟编辑器
- 配置系统时钟

系统时钟 API

一起提供了系统时钟 API 与 cy_boot 生成的 API 文件。系统参考指南中列出并描述了系统时钟 API。

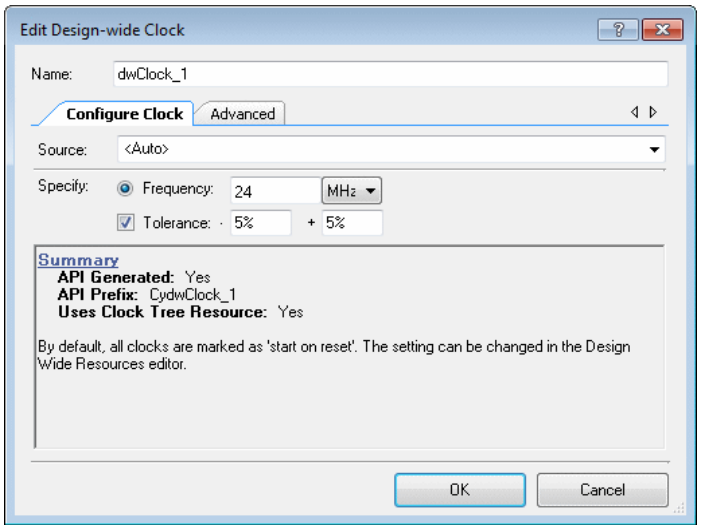
成功编译后，所生成的文件位于工作区浏览器中名称为“Generated_Source/<Architecture Name>/cy_boot”的文件夹内。更多有关编译过程的信息，请参阅生成的文件。

另外，请参考：

- 系统参考指南
- 时钟的使用
- 工作区浏览器
- 生成的文件

添加/编辑设计范围时钟

通过添加/编辑设计范围时钟对话框，您可以添加和编辑设计中各种设计范围内的时钟。



注意：设计范围的时钟使用器件的资源并具有特定的 API，其 API 带有前缀“Cy”。

如何打开对话框：

从时钟编辑器中打开该对话框。

- 请点击 **Add Design-Wide Clock**（添加设计范围时钟），打开用于创建新的设计范围时钟的对话框。
- 双击现有的设计范围时钟，打开编辑对话框。

要想配置设计范围的时钟，请执行下面操作：

根据下述内容配置设计范围的时钟：

1. 在 **Name** 框中，输入时钟名称或使用默认的名称。
2. 在 **Clock Type**（仅适用于 PSoC 4/PRoC BLE）中，选择“New”或“Existing”项。
3. 在 **Source** 中，从下拉菜单选择源时钟的类型。
   - 选择“<Auto>”后，您可以指定 **Desired Frequency**（所需频率）的值，并能够选择 **Tolerance**（容差）项。然后，PSoC Creator 将计算源和分配值。
   - 选择“<Select Signal...>”后，将打开 Select Source Clock（选择源时钟）对话框，该对话框提供了一个可用信号的列表供您选择。
   - 选择“<Select Pin...>”（仅适用于 PSoC 4/PRoC BLE 器件）后，将打开 Select Source Clock (from Pin)对话框，该对话框提供了一个可用引脚的列表供您选择。
   - 选择其他任何特定的基准时钟，您可以指定 **Desired Frequency**（所需频率）或 **Divider** 的分频值，如下所述：
     - 如果您指定了 **Desired Frequency**（所需频率），则 PSoC Creator 会自动计算分频值。
     - 如果您指定了 **Divider**（分频值），PSoC Creator 将直接使用该值。

   您可以使用 Design-wide Resources 时钟编辑器查看各时钟的特性。

注意：对于 PSoC 4/PRoC BLE 器件，还存在一个 **Use fractional divider**（使用分数分频值）选框。如果您指定了 **Frequency** 项，勾选 **Use fractional divider** 选框则表示 PSoC Creator 将计算分数。如果您指定的是 **Divider** 选项，该选框将提供用于指定分数的手动工具。

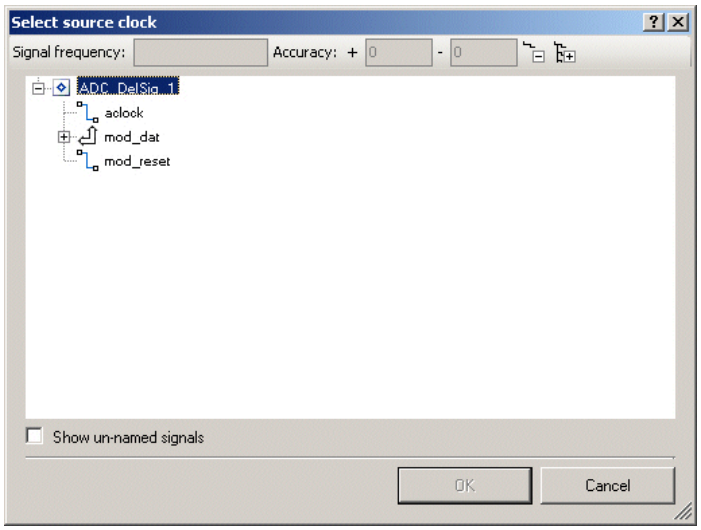
4. 通过 **Advanced** 选项卡，可以指定该时钟是否与 MASTER_CLK 同步（默认情况为同步）。该选项卡不适用于 PSoC 4/PRoC BLE 器件。

另外，请参考：

- 设计范围资源
- 时钟编辑器
- 时钟的使用
- 选择源时钟
- 选择源时钟（引脚）

选择源时钟

选择源时钟对话框提供了一个显示了设计中各种信号（按字母顺序排列）的树形视图或表格视图。通过该视图，您可以选择给定时钟的输入信号。



要想把它作为时钟输入使用，信号必须满足下面所有要求：

- 必须是数字信号。
- 如果信号来自某个终端，该终端必须处于原理图的顶层。
- 否则，信号不能被连接到原理图终端。

要打开该对话框，请执行下述操作：

按照下面内容打开该对话框：

- 在配置系统时钟对话框中的数字信号时钟部分，点击省略号按钮**[...]**。
- 在添加/编辑设计范围时钟对话框中的 **Source** 下，选择下拉菜单中的“<Select Signal...>”项。（对于 PSoC 4/PRoC BLE 器件，您必须将 **Clock Type** 选择为“Existing”。）

信号频率：

该字段用于指定选定信号的频率。

准确度：

这些字段用于指定选定信号的准确度。该值常显示为一个百分数，但您可以输入%或 ppm 的数值形式。

工具栏：

工具栏中有展开和收缩指令，用以显示或隐藏整个信号树。

Show Un-named Signals（显示未命名信号）：

选择该选框后，可以显示设计中的所有信号；如果取消选择该选框，则只显示被命名的信号。

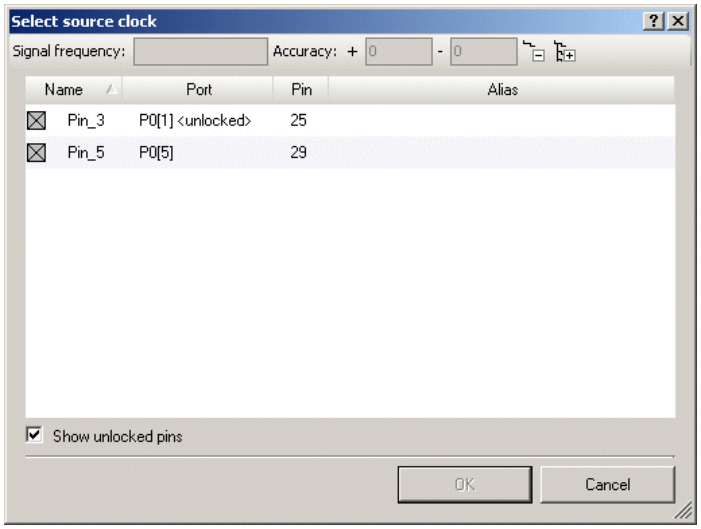
另外，请参考：

- 设计范围资源
- 时钟编辑器
- 添加/编辑设计范围时钟
- 配置系统时钟

选择源时钟（引脚）

选择源时钟对话框提供了一个显示设计中各种信号（按字母顺序排列）的表格视图。通过该视图您可以选择给定时钟的输入信号。

该对话框仅适用于 PSoC 4/PRoC BLE 器件。



要想把它作为时钟输入使用，信号必须传输给数字输入引脚。

要打开该对话框，请执行下述操作：

从添加/编辑设计范围时钟对话框中打开该对话框。

1. 在 **Clock Type** 字段中，选择“Existing”项。
2. 在 **Source** 字段中，从下拉菜单选择“<Select Pin...>”项。

信号频率：

该字段用于指定选定信号的频率。

准确度：

这些字段用于指定选定信号的准确度。该值常显示为一个百分数，但您可以输入%或 ppm 的数值形式。

工具栏：

工具栏中有展开和收缩指令，用以显示或隐藏整个信号树。

表格字段：

信号表包括以下列：

列	说明
名称	为引脚定义的信号名称。
端口	以端口格式显示的器件引脚。该列还显示了引脚是否被解锁。
引脚	该信号被分配给的器件引脚编号。
别名	若适用，则该列将显示引脚的备用名称。

Show Unlocked Pins（显示解锁引脚）：

通过选择该选框，可以显示设计中的所有解锁引脚。

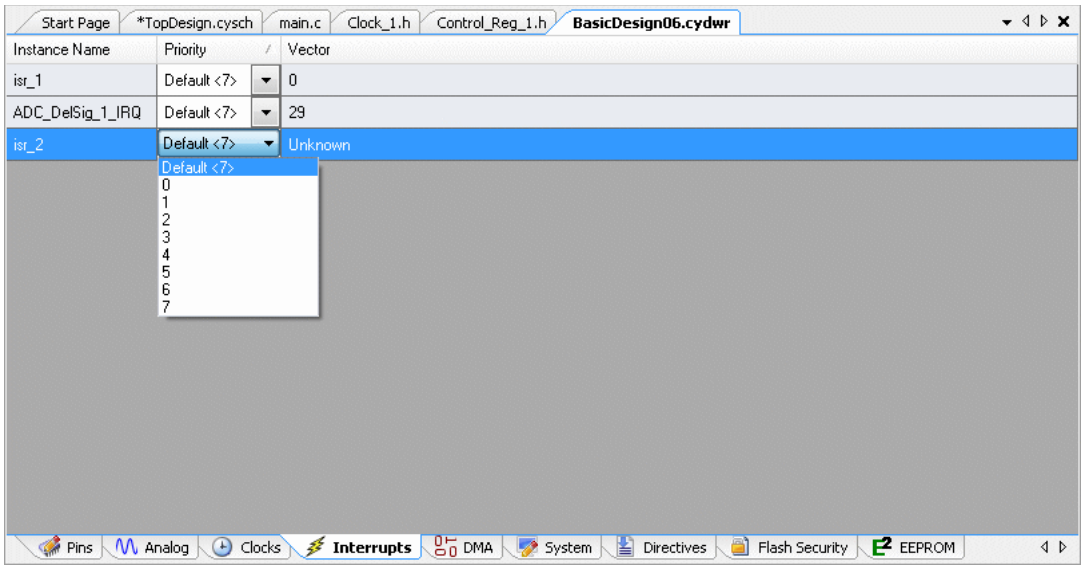
另外，请参考：

- 设计范围资源
- 时钟编辑器
- 添加/编辑设计范围时钟

中断编辑器

通过中断编辑器，您可以修改设计中各中断服务子程序（ISR）的优先级。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



该编辑器是 Design Wide Resources 文件的一部分，该文件还包括其它资源，如时钟编辑器。

中断编辑器包括的一个列表包含下面各列：

- **Instance Name**（实例名称） — 实例名称也作为 ISR 的名称使用。该列是只读的。
- **优先级** — 通过其下拉菜单，可以选择中断的优先级。该列有一个默认值，描述为“Default <#>”，因此可以区分默认值的选择。器件更改时，只有默认值被更新，并不更新非默认值。

  如果更改器件导致了非法值，则单元格中将显示一个错误图标。注意列表窗口中也显示了该错误。
- **向量** — 仅在完成了放置和路由操作后，该信息（中断编号）才可用。该列是只读的。

要想打开中断编辑器，请执行下面的操作：

双击 Workspace Explorer 的 **Source** 选项卡中的*<project>.cydwr* 文件。

然后点击 **Interrupts** 选项卡。

要想修改优先级，请执行下面操作：

从下拉菜单选择相应中断的值。

另外，请参考：

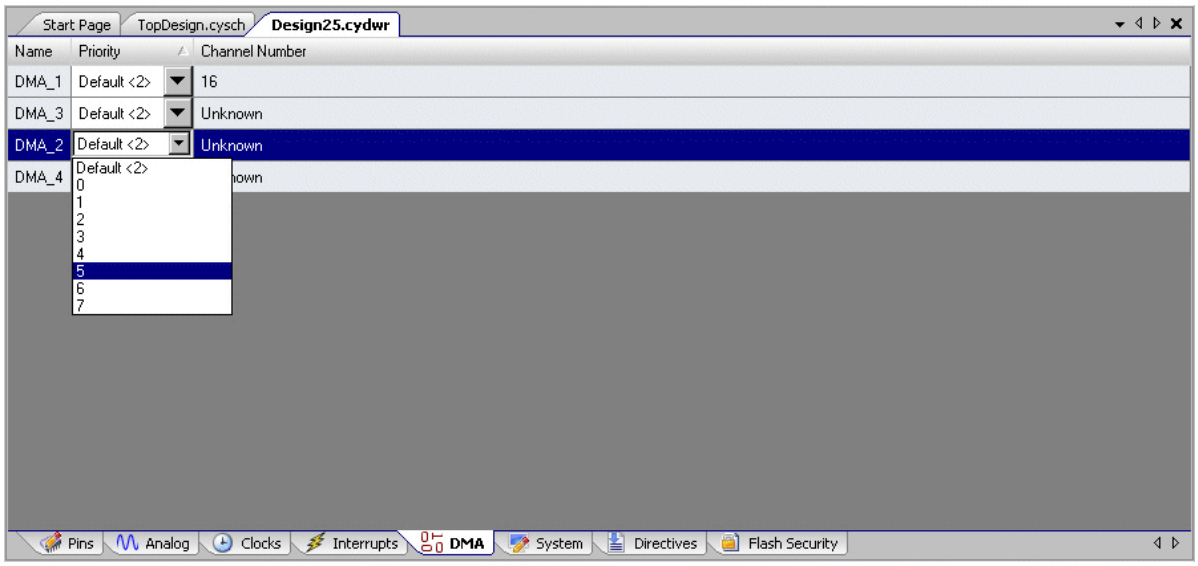
- 设计范围资源
- 中断的工作原理
- 中断组件数据手册（从组件目录中打开）

DMA 编辑器

DMA 编辑器将显示设计中直接放置的所有直接存储器存取（DMA）组件以及所有 DMA 组件中内部放置的组件。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。

DMA 编辑器不适用于 PSoC 4/PRoC BLE 器件。



DMA 编辑器包含的表格，每一行表示一个 DMA。该表包括下面各列：

- **名称** — DMA 所属的实例的层级名。该列是只读的。
- **优先级** — 指定 DMA 的优先级。

  注意：该列有一个默认值，被描述为“Default <#>”，因此可以区分默认值的选择。器件更改时，只有默认值被更新，并不更新非默认值。如果更改器件导致了非法值，这时单元格中将显示一个错误图标。注意列表窗口中也显示了该错误。
- **通道编号** — 仅在完成了放置和路由操作后，该信息才可用。该列是只读的。

注意：如果设计中没有使用 DMA 资源，则 DMA 编辑器将显示一条信息显示它的影响。

要想打开 DMA 编辑器，请执行下面操作：

双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

然后点击 **DMA** 选项卡。

要想修改优先级，请执行下面操作：

点击菜单中的向下箭头，并从下拉菜单中选择相应的 **Priority** 值。

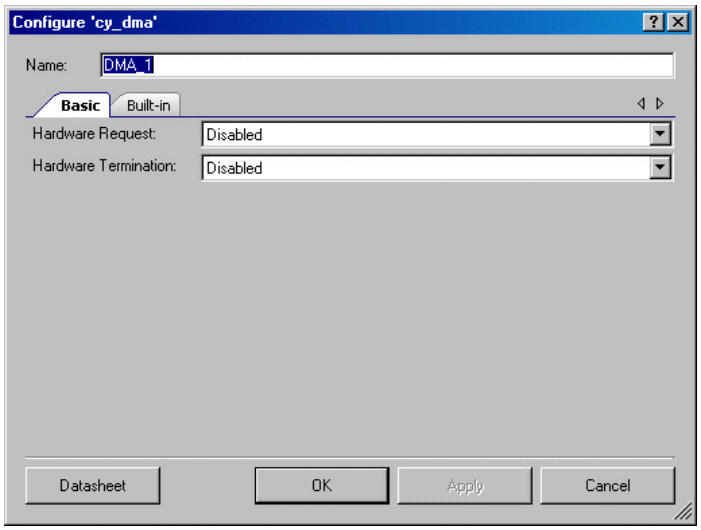
要想排列 DMA 编辑器中的表格，请执行下面操作：

默认情况下，DMA 编辑器的表格是按 **Priority** 列的升序排列的，次要的排列顺序是按通道编号列进行排序。

要想按其它列排序，请点击相应的列表头。次要的排列顺序一般都按通道编号列。

要想选择/编辑 DMA，请执行下面操作：

双击表格中的入口；PSoC Creator 将打开原理图文件并高亮显示相应的 DMA 实例，然后再打开配置 DMA 对话框。



您可以编辑下面各参数：

- 硬件请求 — 如果该参数被使能，将添加 drq 终端，这样才能通过来自硬件的 DMA 请求。
- 硬件终止 — 如果使能该参数，将添加 trq 终端，这样才能终止来自硬件的 DMA 请求。

更多信息，请参考 DMA 组件数据手册。

另外，请参考：

- 设计范围资源
- 中断的工作原理
- DMA 向导
- 编辑组件参数
- DMA 组件数据手册（从组件目录中打开）

DMA 向导

DMA 向导

通过 DMA 向导，可以快速正确地开发使用 DMA 的应用。向导会逐步指导您定义数据操作的描述符。它也会生成必要的 C 语言代码，您可以将这些代码复制并粘贴到您的应用中。DMA 向导仅适用于至少包含了一个 DMA 组件的设计项目。如果设计中不存在 DMA 组件，则向导将显示一个信息表示它的影响。

DMA 向导包括下面各步骤：

- 开始（本主题）
- 全局设置
- 数据操作描述符
- 生成代码

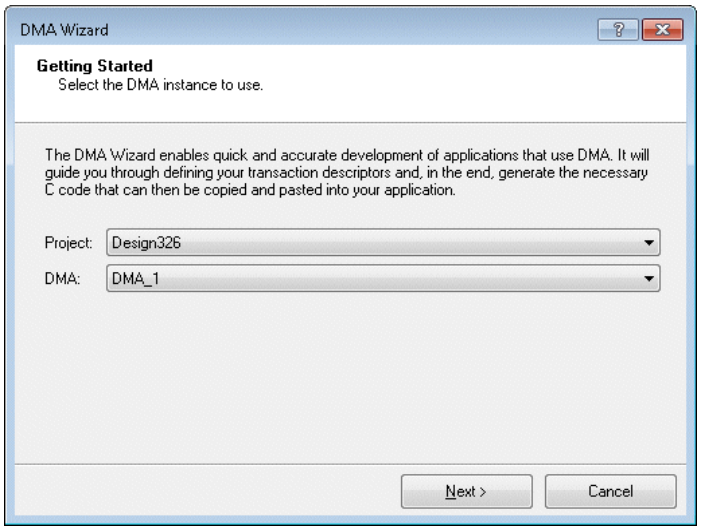
要想打开 DMA 向导，请执行下面操作：

从 **Tools** 菜单中选择 **DMA Wizard** 项。

系统将分析项目中的 DMA 组件，并打开向导。如果在您的设计中不存在 DMA，将显示一个信息提示您需要打开带有 DMA 组件的项目。

Getting Started（开始）：

在开始步骤中，您可以选择需要配置的项目和 DMA 实例。



该步骤包括下面字段：

- **Project**（项目） — 该字段列出了当前打开的工作区中包含 DMA 组件的所有设计项目。如果“活动”项目满足上述要求，将默认选中它。
- **DMA** — 用于选择所需的通道（DMA 组件）。

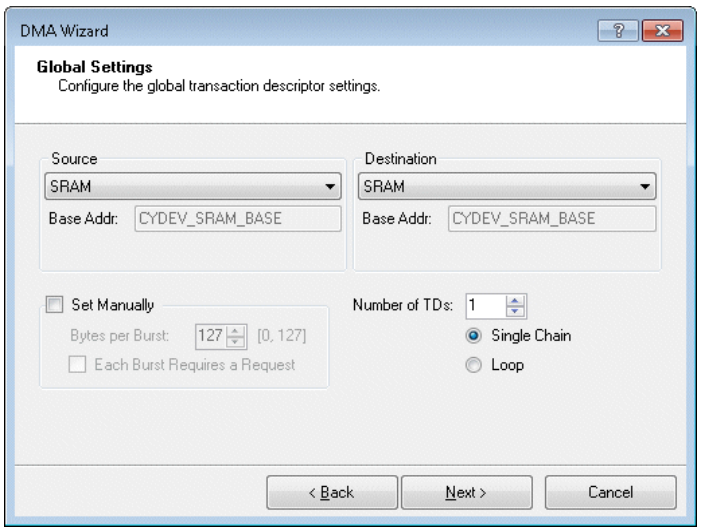
选择合适的项目和实例后，请点击 **Next >**转到下一步。

另外，请参考：

- DMA 向导全局设置
- DMA 向导数据操作描述符
- DMA 向导生成的代码
- DMA 编辑器

DMA 向导全局设置

在全局设置步骤中，您将配置全局数据操作描述符（TD）的设置



该步骤包括下面的设置内容：

Source（源）：

- **Source**（源） — 可以是 SRAM、Flash，或者是 EEPROM。它也可以是适用于 DMA 向导的组件。如果选定的组件具有多个源，则可以通过次要的下拉菜单实现更多的特定选择。
- **Base Addr**（基地址） — 根据所使用的 PSoC 器件，可以自动或手动提供基地址字段的内容。手动输入这些内容时，需要使用 C 语言表达式。如果没有填入该字段，将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。

Destination（目标）：

- **Destination**（目标） — 它可以是 SRAM 或适用于 DMA 向导的组件。如果选定的组件具有多个目标，则可以通过次要的下拉菜单实现更多特定的选择。
- **Base Addr**（基地址） — 根据所使用的 PSoC 器件，可以自动或手动提供基地址字段的内容。手动输入这些内容时，需要使用 C 语言表达式。如果没有填入该字段，将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。

Set Manually（手动设置）：

下面字段被自动定义。要手动设置这些字段，请选择 **Set Manually** 选框。

- **Bytes per Burst**（每传输的字节数量） — 通过该字段，您可以设置每次传输中将要传输的字节数量。该值是源和目标都支持的最大值。该字段右侧显示的是当前源和目标的合法值范围。如果您输入一个无效值或者不存在合法值，那么字段旁会出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。
- **Each Burst Requires a Request**（每次传输要求一个请求） — 通过该字段，您可以设置每次传输传输前是否要求一个请求。根据所选的源和目标自动计算出该值。

数据操作描述符：

- **Number of TDs**（TD 数量） — 指定所创建数据操作描述符的数量（有效范围为 1 到 128）。
- **Single Chain or Loop**（单链或循环） — 该字段将确定最终 TD 后的 Next TD（下一个 TD）。如果是单链，则 Next TD 表示结束。如果是循环，将返回到第一个 TD。

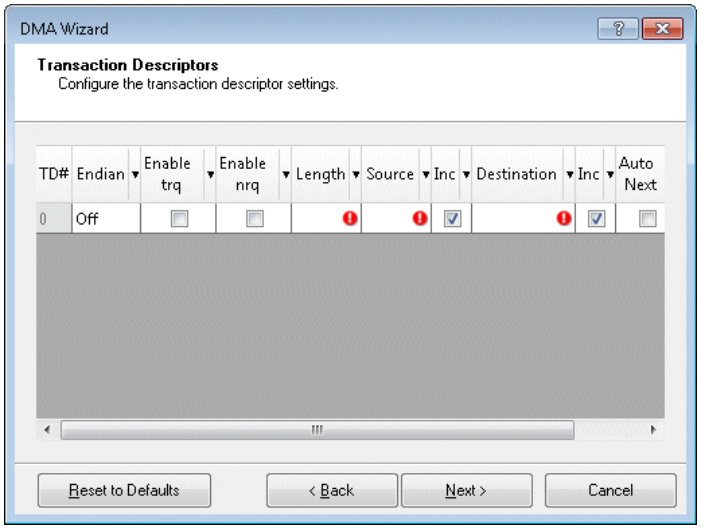
配置合适的设置内容后，点击 **Next >**以转到下一步。

另外，请参考：

- DMA 向导
- DMA 向导数据操作描述符
- DMA 向导生成的代码
- DMA 编辑器

DMA 向导数据操作描述符

在数据操作描述符步骤中，您将配置每个单独数据操作描述符（TD）的相关设置。



该步骤包括下面各字段：

字段	说明
TD#	它显示了数据操作描述符的逻辑编号。它与 Next TD 结合使用。
Endian（字节序）	允许按 2 或 4 字节的字节序列进行交换。当该字段被设置为 2 或 4 时，必须将**每次传输的字节数量**设置为该值的整数倍。否则，该单元格将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。
Enable trq（使能 trq）	允许在 trq 信号的上升沿上终止该 TD（TD 传输过程中有效）。
Enable nrq（使能 nrq）	允许在完成 TD 时生成 nrq 信号。
长度	表示 TD 的长度，单位为字节（有效范围为 0 到 4095）。该字段显示的是运行时计算的 C 表达式。如果使能了字节序交换，则该长度必须是字节序交换尺寸的整数倍。否则，该单元格将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。
Source（源）	DMA 传输的源地址。该字段显示的是运行时计算的 C 表达式。如果某个组件被选定为源，该字段可能包括下拉的地址列表。在所有情况中，都能够编辑该单元格。如果该字段为空白，它将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。
Inc（源）	允许在 DMA 传输指定的字节数量后递增源地址。
Destination（目标）	DMA 传输的目标地址。该字段显示的是运行时计算出的 C 表达式。如果某个组件被选定为源，则该字段可能包括下拉的地址列表。在所有情况中，都能够编辑该单元格。如果该字段为空白，它将出现错误图标。它不会阻止您继续按向导进行操作。它只会生成未编译的代码。
Inc（目标）	允许在 DMA 传输指定的字节数量后递增目标地址。
Auto Next（自动执行下一个）	指定完成该 TD 后是否自动执行下一个 TD，而不需要其它请求。
Next TD（下一个 TD）	指定 TD 链中的下一个逻辑 TD。如果 TD 链结束于这个 TD，请将该字段设置为 END。

Reset to Defaults（复位到默认设置）按键将表中的所有值复位为其默认值或计算得到的值。

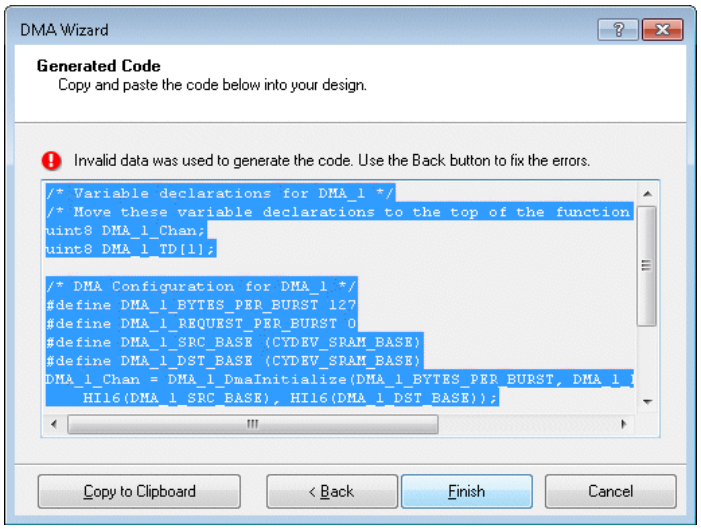
配置合适的设置内容后，请点击 **Next >**以转到下一步。

另外，请参考：

- DMA 向导
- DMA 向导全局设置
- DMA 向导生成的代码
- DMA 编辑器

DMA 向导生成的代码

在生成的代码步骤中，将显示代码，供您复制并粘贴到设计中。



通过 **Copy to Clipboard**（复制到剪贴板）按键，可以将代码添加到剪贴板上。如果需要，您也可以使用标准的键盘快捷键和右键点击菜单。

注意：如果在上述各步骤中忽略了所有错误，那么将显示一个错误图标，提示您代码不会工作。

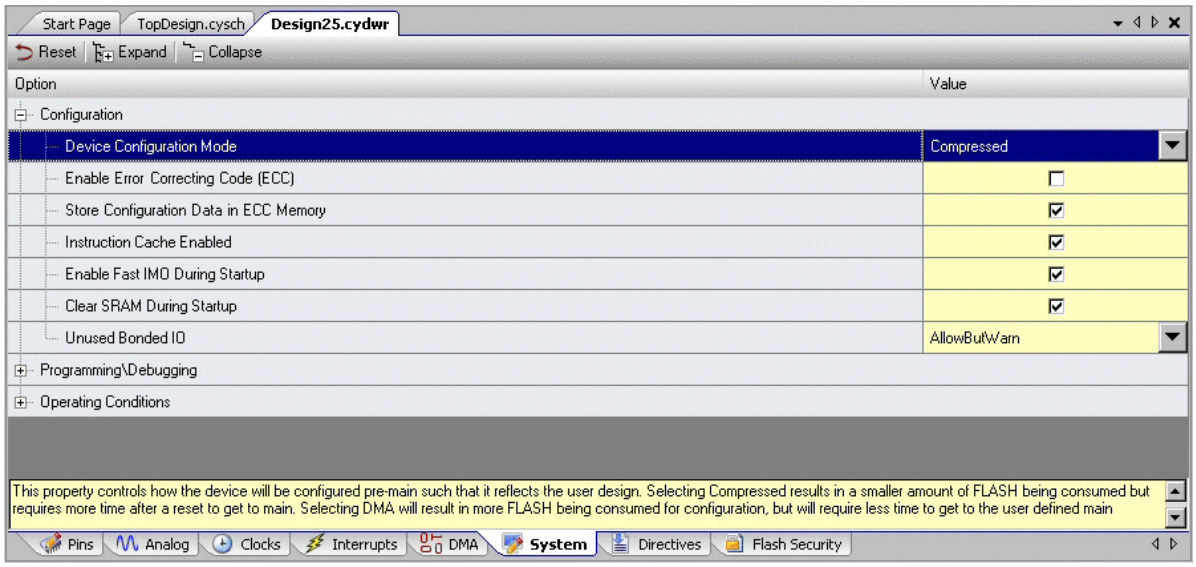
另外，请参考：

- DMA 向导
- DMA 向导全局设置
- DMA 向导数据操作描述符
- DMA 编辑器

系统编辑器

系统编辑器用于编辑各种系统属性。该编辑器中的表格包含了不同的属性目录，如配置、编程/调试以及工作条件等。根据您的设计，可用的目录会不一样。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



要想打开系统编辑器，请执行下面的操作：

1. 双击 Workspace Explorer 的 **Source** 选项卡中的*<project>.cydwr* 文件。
2. 然后点击 Design-Wide Resources 对话框中的 **System** 选项卡。

要想编辑属性，请执行下面操作：

请点击所需编辑的属性中的 **Value** 列。

不同的属性会有不同的编辑方式。有的属性带有可切换使能或禁用状态的选框，有的属性带有下拉菜单供您选择，有的属性则带有输入某一数值的文本字段。

如果您输入的值无效，将显示错误图标，表示无效的值和纠正问题的方法。

属性描述：

在每个目录下，均有一个或多个可编辑的属性列。当您高亮显示特定属性时，编辑器下端的文本框将显示它的说明内容。该表包括下面各列：

- **Option**（选项） — 该列显示层次结构中每一列/层次的名称。
- **Value**（值） — 该列显示设置的当前值，并且您能够修改它（若可用）。

该表包括下面各选项：

选项	说明	注释
配置		
器件配置模式	该属性控制着在进入 main 函数前如何配置器件，比如它反映了设计情况。可用的选项为 Compressed（压缩）、UnCompressed（解压缩）和 DMA。（DMA 选项不适用于 PSoC 4/PRoC BLE。） 如果选择了 Compressed 项，则 FLASH 中的消耗空间较小，但是复位后需要更长的时间才能进入 main 函数。 UnCompressed 使用的 FLASH 空间与 DMA 项的一样，但是它会对配置数据进行软件复制。 DMA 使用更多的 FLASH 空间，但只要经过较短时间便可用进入用户定义的 main 函数。	
使能纠错码（ECC）	如果 ECC 被使能，它将检测和纠正 FLASH 存储器中的错误。选择该选项会隐藏“Store Configuration Data in ECC Memory”（将配置数据存储到 ECC 存储器中）项。 **警报：** 小心在开发过程中更改该设置。对该设置进行过多的重编程可能会导致意外的结果。	仅适用于 PSoC 3 和 PSoC 5LP 器件。
将配置数据存储到 ECC 存储器中	如果该选项被使能，则器件配置数据被存储到 ECC 存储器内，以减少对主 FLASH 存储器的使用。当使能该选项时，可能不使用错误纠正功能。	仅适用于 PSoC 3 和 PSoC 5LP 器件。
指令缓存使能	如果该选项被使能，器件将来自 FLASH 的数据写入指令缓存 SRAM 中。	不适用于 PSoC 4/PRoC BLE 器件。
在启动期间启用快速 IMO	若快速 IMO 被使能，它将被使用。该配置有助于均衡快速启动和配置的需求与高功耗间的问题。在器件启动期间，IMO 的运行速度为 48 MHz，并不是 12 MHz。 该配置有助于均衡快速启动和配置的需求与高功耗间的问题。在器件启动期间和在 CyPmSaveClocks()以及 CyPmRestoreClocks()函数的调用期间，IMO 的运行速度为 48 MHz，并不是 12 MHz。更多有关这些函数的信息，请参阅系统参考指南。 **警报：** 小心在开发过程中更改该设置。对该设置进行过多的重编程可能会导致意外的结果。	仅适用于 PSoC 3 和 PSoC 5LP 器件。
在启动期间清除 SRAM	若被使能，初始化变量前向 SRAM 写入 0。若被禁用，没有明确的初始化程序的变量不被初始化为 0。只有应用要求更短的启动时间，并且代码或库中不包含任何依赖于未初始化变量的值时，才会禁用该选项。	仅适用于 PSoC 3 器件。
读加快使能	若该特性被使能，则器件将使用被缓存的值来加快读操作。	仅适用于 PSoC 4/PRoC BLE 器件。
未使用的接合 IO	该选项控制如何对未使用的接合引脚进行内部模拟放置和路由。 “Allow but warn”选项允许模拟路由器在当前的设计中使用未使用引脚的开关，但会引发有关相应开关所在引脚的警报。 “Allow with info”选项允许模拟路由器在当前的设计中使用未使用引脚的开关，但会引发有关相应开关所在引脚的注意。 “Disallowed”选项不允许模拟路由器在当前设计中使用任何未使用引脚的开关。	
堆尺寸	定义为堆空间保留的 SRAM 的字节数量。	适用于所有基于 ARM 的器件（如 PSoC 4、PRoC BLE 以及 PSoC 5LP）。

选项	说明	注释
栈尺寸	定义为栈空间保留的 SRAM 的字节数量。	适用于所有基于 ARM 的器件（如PSoC 4、PRoC BLE 以及 PSoC 5LP）。
包括 CMSIS 内核外设库文件	CMSIS（Cortex 微控制器软件接口标准）内核外设库包含了用于访问内核寄存器和外设的 API。选中该选项后，将包括项目中标准当前版本的 API。目前版本被捆绑在项目中的 cy_boot 组件版本中。更多有关信息，请参阅系统参考指南。如果需要使用不同的标准版本，应取消选择该选项，并手动添加所需文件。	适用于所有基于 ARM 的器件（如PSoC 4、PRoC BLE 以及 PSoC 5LP）。
编程/调试		
芯片保护	根据不同的芯片保护等级，CPU 和调试器可访问的资源也不尽相同： Open：可以访问所有资源 Protected：不能调试 Kill：器件不能再重新编程 注意： 使用 PSoC Creator 编程 PSoC 4/PRoC BLE 器件时，无论器件的芯片保护等级如何，都会将其设置为“Open”模式。 要想将芯片保护等级设置为“Protected”或“Kill”模式，请在 PSoC Creator 中选择相应的模式，使能 PSoC Programmer 中 **Options > Programmer Options** 路径下的“芯片锁存”性能，然后再使用 PSoC Programmer 编程 PSoC 4/PRoC BLE 器件。	仅适用于 PSoC 4/PRoC BLE 器件。
调试选择	设置端口 1 首选的编程/调试接口（JTAG 或 SWD）；启动或复位后，芯片将默认使能该接口。 JTAG 接口不适用于 PSoC 4/PRoC BLE 器件。 设置 GPIO 的目的是将引脚作为 GPIO 使用，单不完全禁用针对保护 Flash 目的的调试接口。为实现该目的，必须设置“Enable Device Protection”（使能器件保护），或必须将“Chip Protection”（芯片保护）设置为 Open 模式。 更多有关编程和调试选项的信息，请参阅器件数据手册或技术参考手册（TRM）。 **警报：** 小心在开发过程中更改该设置。对该设置进行过多的重编程可能会导致意外的结果。	PSoC 3 和 PSoC 5LP 选项包括： 5 线 JTAG 4 线 JTAG SWD（串行线调试） GPIO PSoC 4/PRoC BLE 选项包括： SWD（串行线调试） GPIO
使能器件保护	完成一个生产设计时，您应该选择 Enable Device Protection（使能器件保护）功能。使能该功能后，器件将在运行期间禁止调试。器件可以与编程器相连，但是会禁用调试功能。不建议对多器件 JTAG 链使能该功能，因为它会破坏该链。 **注意：**该设置不会影响闪存保护。只是使用它来禁用 PSoC 3 或 PSoC 5LP 器件的调试访问。	不适用于 PSoC 4/PRoC BLE 器件。
嵌入式走线（ETM）	如果使能了 Embedded Trace (ETM)功能，则在处理器运行期间，可以输出实时调试的信息。 该功能将走线引脚保留为调试引脚。不作为走线使用时，PSoC Creator 将该引脚作为 GPIO 引脚使用。	仅适用于 PSoC 5LP 器件。

选项	说明	注释
使用可选的 XRES	通过可选的 XRES 引脚（P1[2]）使能硬件复位。除了专用引脚以外，该可选 XRES 引脚是器件的辅助引脚。 **警报：** 小心在开发过程中更改该设置。过多的重新编程该设置（超过 1000 次）可能会使之成为永久性选择。	适用于具有 48 个引脚以上的 PSoC 3 和 PSoC 5LP 器件。
使能 XRES	通过可选的 XRES 引脚（P1[2]）使能硬件复位。该器件没有专用的 XRES 引脚。如果禁用了该项，经无法对没有 Power Cycle 的器件进行编程，或无法对器件进行硬复位。 默认情况下，通过切换 XRES 引脚来实现复位。Power Cycle 的编程方法高度依赖于电路板的设计，它可能不会工作。如果 XRES 引脚不可用，并且 Power Cycle 也不工作，则不能重新编程器件。 为确保能够对器件进行编程，需要使能该选项。 **警报：** 小心在开发过程中更改该设置。过多的重新编程该设置（超过 1000 次）可能会使之成为永久性选择。	适用于不超过 48 个引脚的 PSoC 3 器件。

运行条件

这些设置适用于指定 PSoC 器件的各种电压和温度范围。PSoC 器件中的各种组件将这些值作为配置信息使用，所以您需要使用正确的数值。这些设置通过多种方式影响您的设计，包括：

- USB_Start 函数可以使用 Vddd，以设置枚举的内部 USB 电压调压器。
- ADC_CountsTo_Volts() API 将输入电压范围设置为 Vssa ~ Vdda。
- SAR_ADC 组件使用 Vdda 电压设置来配置输入范围和参考电压。
- 在 Vdda 低于 2.7 V 时，SC 模块组件（TIA、PGA、PGA_Inv、Sample_Hold 和 Mixer）将使能升压时钟。
- VDAC8 组件会生成一个注意信息，显示 VDAC 被限制在 0 到 Vdda 的范围内。
- 如果 Vddd >= 1.8 V（并且温度范围为 0 °C 到 85 °C），则静态时序分析的时序延迟将低于 UDB 中的时序延迟。如果电压（VddioN）等于或大于 3.3 V，则 IO 的延迟会更短。

选项	说明	注释
Vdda (V)	输入值表示 Vdda 引脚上的输入电压	
变量 VDDA	当 Vdda 等于或大于 2.7 V 时，会显示选框，用以创建低电压的模拟升压时钟 ScBoostClk。更多有关信息，请参阅系统参考指南。	
Vddd (V)	输入值表示 Vddd 引脚上的输入电压	
Vddio0 (V)	输入值表示 Vddio0 引脚上的输入电压	不适用于 PSoC 4/PRoC BLE 器件。
Vddio1 (V)	输入值表示 Vddio1 引脚上的输入电压	
Vddio2 (V)	输入值表示 Vddio2 引脚上的输入电压	
Vddio3 (V)	输入值表示 Vddio3 引脚上的输入电压	
温度范围	指定器件的工作温度范围。 PSoC 器件能够在较广的温度范围下可靠运行。如果您的设计仅在一般室温下工作，那么更容易满足它的时序约束。选择较小的温度范围有助于查找符合时序的路由解决方案。	可用的选择有： 从 0C 至 85C（仅适用于 PSoC 3 和 PSoC 5LP） 从-40C 至 85C（适用于所有的器件）

注意：这里输入的电压值用于确定的 API（例如 ADC_CountsTo_Volts）以及组建配置对话框（例如，如果 Vref 被选定为 Vdda/4 或 Vdda/3，ADC_DelSig 组建将该 Vdda 信息作为输入范围使用）。尽管建议更新 DWR 系统编辑器中的实际 Vddx 电压，但是就算提供的是其它的有效供电电压（根据数据手册中），也不会破坏器件。

Bootloader/Bootloadable 设置

这些设置适用于使用 cy_boot_v2.40 或更早版本的项目。对于 cy_boot_v3.0 或更新的版本，会使用单独的 Bootloader 和 Bootloadable 组件。有关详细信息，请参考相应的组件数据手册。

选项	说明	注释
Bootloader		
IO 组件	IO 组件支持 Bootloader IO 与 Bootloader 主机间进行通信。可用值依赖于设计中 Bootloader 使能的 IO 组件的设置。	只有 Bootloader 项目的设置内容会被显示。
数据包校验和类型	在 Bootloader 主机和固件间发送数据包时使用的效验和类型。 “基本求和”对数据包中所有字节的总和进行 2 进制补码。 “CRC-16”将使用 CCITT 子程序来计算数据包中所有字节的 16 位 CRC。	
等待指令	复位时，如果 Bootloader 检测到应用的效验和有效，则 Bootloader 会等待开始加载新项目的指令。如果 Bootloader 将等待，请使能该项。	
等待指令时间（10 ms）	器件复位后，Bootloader 要等待一条指令后才能开始加载新的应用。等待指令的时间就是启动现有应用之前所等待的时间。如果“等待指令的时间”被禁用，那么该选项无效。所有时间的单位均为毫秒。	
版本	以两字节的十六进制格式定义 Bootloader 的版本编号。	
Bootloadable		
版本	以两字节的十六进制格式定义 Bootloadable 应用的版本编号。	只有 Bootloadable 项目的设置内容会被显示。
应用 ID	定义用于识别应用的两字节应用 ID。	
自定义 ID	定义应用中某种属性的 4 字节自定义 ID。	

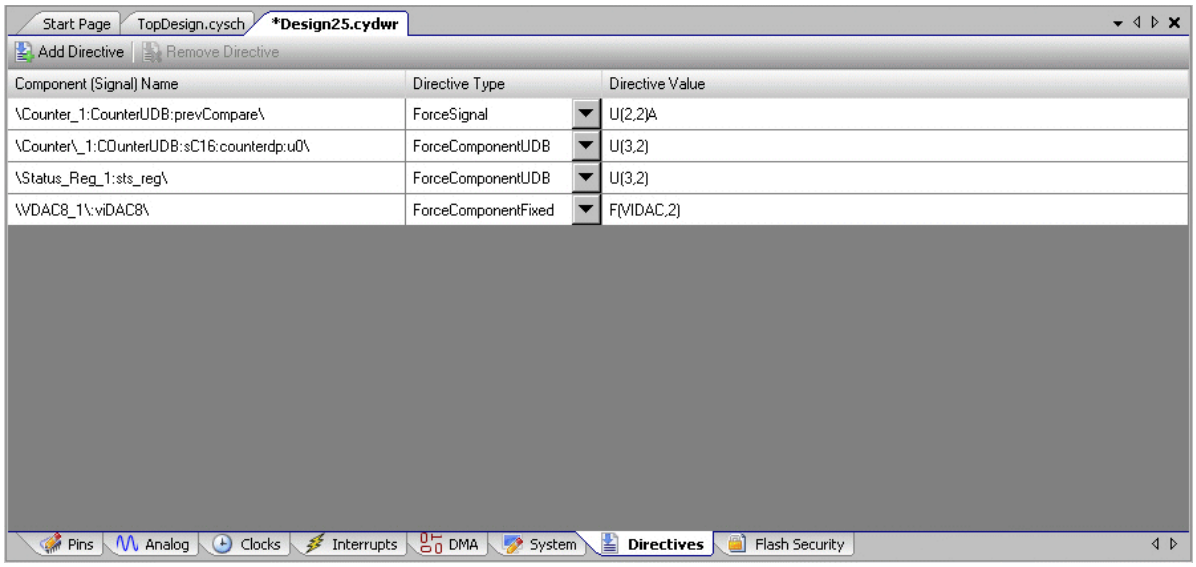
另外，请参考：

- 设计范围资源
- 系统参考指南
- Bootloader 和 Bootloadable 组件数据手册（位于组件目录中）

指令编辑器

通过指令编辑器，可以添加、移除和编辑指令。更多有关 PSoC Creator 中可用的指令的信息，请参阅指令中的内容。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



要想打开指令编辑器，请执行下面操作：

双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

点击 **Directives** 选项卡。

如何添加指令：

请点击编辑器左上端上的 **Add Directive** 按键。

如何编辑指令：

在相应的列中，输入**组件/信号名称**、**指令类型**以及**指令值**。

- **组件（信号）名称** — 输入信号或组件的名称。当前不能检查该列的有效值，除非在“指令值”列中输入了某个值时也向该列输入某一值。

  注意： 该字段中的组件名称是成功编译后在 *<project>.rpt* 文件中指定的全名（例如，“\Counter_1:CounterUDB:sC8:counterdp:u0\”）。您可以在 Workspace Explorer 中 **Output** 选项卡下找到*<project>.rpt* 文件。

- **指令类型** — 通过该列的下拉列表，选择有效的指令类型。该列的初始值为 INVALID（无效）。如果您在指令值列中输入某一值，必须将指令类型列的内容改为相应的类型。有效类型包括：
  - **ForceSignal**：映射为“placement_force”指令格式，用于任意逻辑。通过这种类型可以将 UDB PLD 位置分配给某个逻辑模块的输出信号，以通知放置器将该输出放置在 UDB PLD 中。这样会生成一个规则形式：

    attribute placement_force of [signal name] : signal is "[UDB spec]"
    其中：[UDB spec]：U(x,y)[A|B]或 U(x,y,[A|B])i

  - **ForceComponentUDB**：映射为 UDB 组件的“placement_force”指令格式。将所列出的 UDB 组件位置分配给指定的位置。这样会生成一个规则形式：

    attribute placement_force of [component name] : label is "[UDB spec]"
    其中，[UDB spec]：U(3,2)

- **ForceComponentFixed:**映射到固定功能模块的“placement_force”指定格式。将所列出的固定模块位置分配到指定的位置。这样会生成一个规则形式：

  attribute placement_force of [component name] : label is "[fixed spec]"
  其中，[fixed spec]：F([fixed block],i)
  其中：[fixed block]：CAN、Comparator、I2C、SC、Timer、VIDAC、...

- **Group**：映射为“placement_group”指令格式。将指定的信号名称组合在指定的组中。这样会生成一个规则形式：

  attribute placement_group of [signal name] : signal is "[group name]"

- **指令值** — 根据选定的指令类型，使用该列指定指令值。每种类型的值都符合一定的规则。如果指令值不遵循类型规则，则该列旁边将出现错误图标（请参考图中的 signal_5 实例）。将鼠标悬停在错误图标上，会显示发生错误的原因。

如何删除指令：

点击表中的某一行，然后点击 **Delete Directive** 按键。

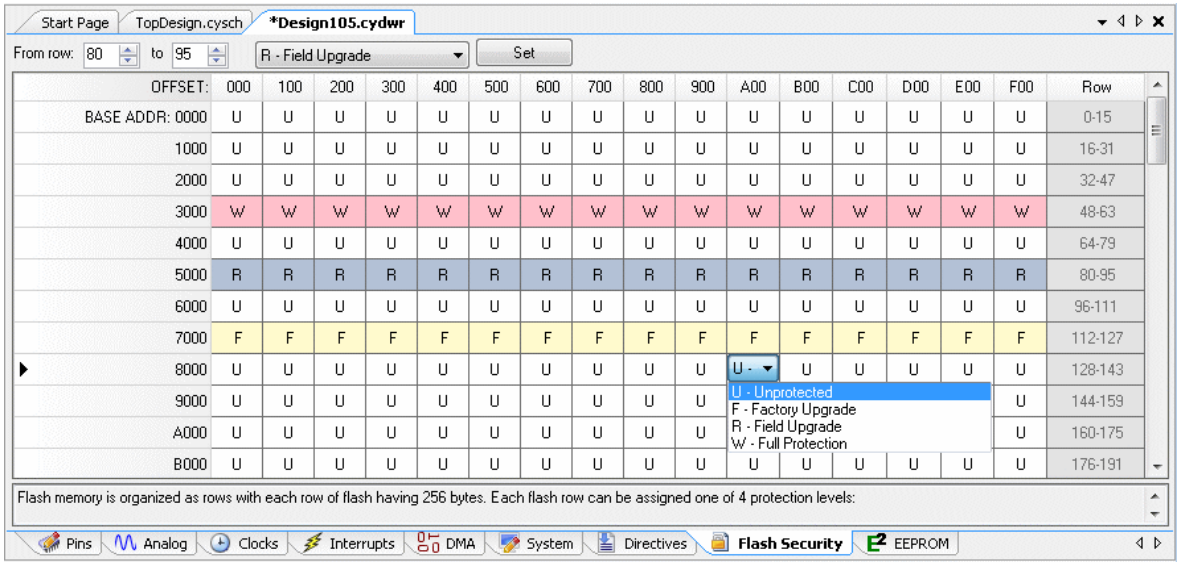
另外，请参考：

- Design-Wide Resources（设计范围资源）
- 指令

Flash 安全编辑器

通过 Flash 安全编辑器，您可以控制对器件存储器进行的读/写访问。Flash 行以表格形式显示，表格中的每个可编辑单元均代表 Flash 的一行（256 个字节）。可以单独设置每一 Flash 行的保护等级。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



如何打开 Flash 安全编辑器：

双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

在工作区内，该文件显示为选项卡式文档，这样便于您访问项目中的各种设计范围资源。点击 **Flash Security** 选项卡，访问 Flash 安全编辑器。

保护等级：

该工具支持如下四个保护等级：

- 无保护（U）— 没有任何保护。
- **工厂升级（F）** — 读保护。所有的外部器件即使使用 SPC 读指令也无法读取受读保护的 Flash 模块。 只有处理器和 PHUB 才能访问受读保护的 Flash 模块。（该项不适用于 PSoC 4/PRoC BLE 器件。）
- **现场升级（R）** — 外部写保护。所有的外部器件均不能擦除或写入受外部写保护的 Flash，包括读保护限制。Bootloader 运行于该保护等级。（该项不适用于 PSoC 4/PRoC BLE 器件。）
- **全保护（W）** — 全面保护。处理器不能擦除或写入受全保护的 Flash 模块。包括所有更低保护等级的 Flash 数据保护。不再允许内部或外部器件修改 Flash 模块时，会使用该保护等级。

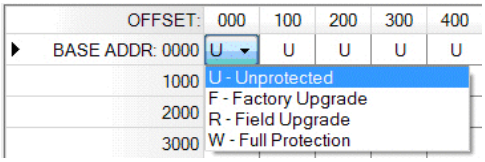
  注意：该保护等级不适用于 Flash/Bootloader/Bootloadable 项目的最后一行，因为这一行作为元数据，并要求执行内部写访问。

生成的 Hex 文件：

有两个包含 Flash 安全等级生成的 hex 文件。在编译期间，通过 CyFlashSecurityModel 收集 Flash 保护的数据，然后输出一个 hex 文件（位于 *Generated_Source/[Architecture]/protect.hex* 路径下）。该数据被组合成项目的综合 hex 文件（位于*[Platform (DP8051-Keil_Generic/Debug, ...)]/[Configuration (Debug, Release)]/[Prj Name].hex* 路径下）。

如何修改单一的 Flash 行：

要想修改单一 Flash 行的保护等级，请点击表格中相应的单元格，并从下拉列表中选择所需项。



如何修改多个 Flash 行：

要想同时修改多个连续的 Flash 行的保护等级，请执行下面的操作：

1. 通过编辑器顶部提供的字段，选择起始和结束 Flash 行的编号。

   默认的起始和结束值包括所有 Flash 行。
2. 从上述字段右侧的下拉列表中选择所需保护等级。
3. 点击 **Set** 按键。

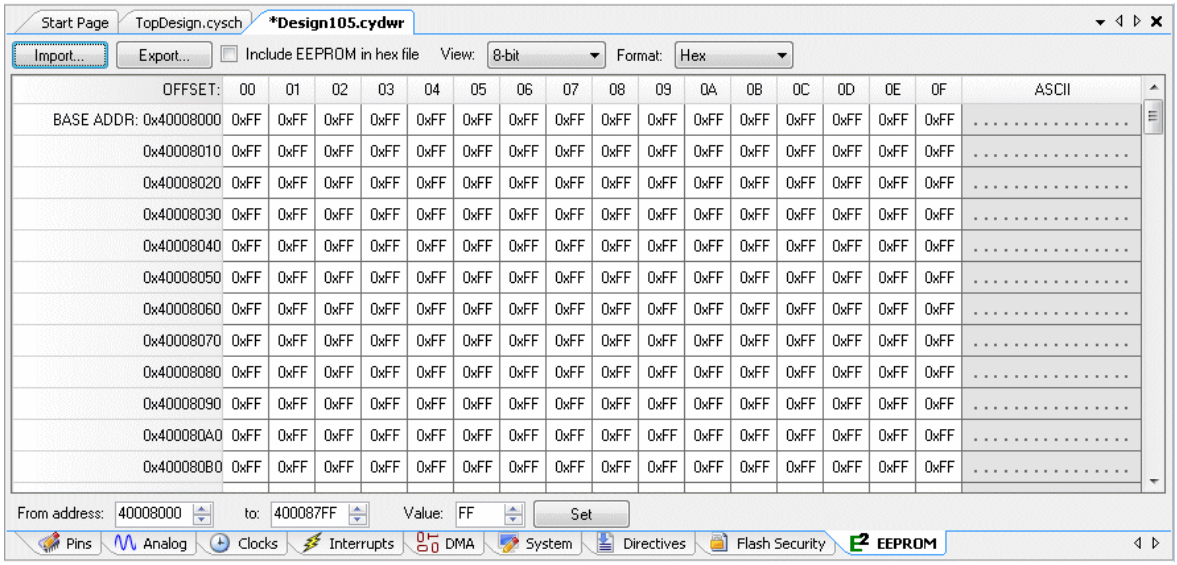
另外，请参考：

- Design-Wide Resources（设计范围资源）

EEPROM 编辑器

通过 EEPROM 编辑器，您可以从 PSoC Creator 中设置 EEPROM 数据，不必使用任何代码来运行 PSoC 应用。根据所显示的项，以网格形式显示 EEPROM 存储器。每个单元格均可编辑，其默认值为 0xFF。

注意：PSoC Creator 包含了动态的 DWR 信息。根据设计的复杂性，可能需要几秒钟的时间来更新 DWR 信息。



左侧列显示的是基地址。最上面一行显示的是每一列的偏移。在右侧显示的 ASCII 码，同存储器窗口中的相同。

如何打开 EEPROM 编辑器：

双击 Workspace Explorer（工作区浏览器）内 **Source**（源）选项卡中的“.cydwr”文件。

在工作区内，该文件显示为选项卡式文档，这样便于您访问项目中的各种设计范围资源。点击 **EEPROM** 选项卡，以访问 EEPROM 编辑器。

如何修改单一的单元值：

要想修改单一的单元值，请点击然后输入所需值，或从菜单列表中选择所需的值。

如何修改多个单元值：

要想修改多个单元值，请使用网格下面的字段。在 **From address**、**to** 和 **Value** 等字段中输入或选择相应的值。然后点击 **Set**。

如何导入/导出数据：

您可以在外部编辑器内用逗号分隔值（CSV）的格式写入一个字节序列，然后保存为 CSV 文件，并将其导入 PSoC Creator 中。点击 **Import**，指出 CSV 文件的地址，选择它，然后点击 **Open**。更改内容被应用在 EEPROM 编辑器中。

您也可以在 EEPROM 编辑器中进行更改，导出 CSV 文件，然后在外部编辑器中浏览该文件。点击 **Export**，指出保存文件的地址，然后点击 **Save**。

如何使 Hex 文件中包含 EEPROM 镜像：

选择该选框，以便在 hex 文件中包含 EEPROM 镜像。如果包含镜像，会增加编程时间。默认情况下，未选择该选框。

注意：**如果未选择该选框，则项目的功能没有发生变化。**

如何修改显示：

下面的设置内容被保存在用户配置文件中，并且它会更改所有用户项目的显示。这些值只会更改显示内容，不会修改数据的实际值。

- 使用 **View** 下拉菜单更改在编辑器中显示的数据大小：8 位、16 位或 32 位。
- 使用 **Format** 的下拉菜单更改在编辑器中显示的数据格式：Hex（十六进制）、Signed Int（带符号的整数）和 Unsigned Int（无符号的整数）。

Bootloader 支持：

EEPROM 编辑器适用于所有项目类型。但是，不能同时用于 Bootloader 和 Bootloadable。否则，PSoC Creator 将生成一个错误。

对于具有多个应用的 Bootloader，EEPROM 被分为多个相等的部分，用于所有 Bootloadable。

规则：

1. 如果您将 EEPROM 使用于 Bootloader/多应用的 Bootloader 项目，那么使用了 Bootloader 的 bootloadable 项目不能再使用 EEPROM（Bootloadable 项目中的 DWR 不允许使能用于该项目的 EEPROM）。
2. 如果 Bootloader 没有使用 EEPROM，并且不是多应用 Bootloader，任何使用 Bootloader 的 Bootloadable 项目可以在 DWR 中使能 EEPROM，并能够使用 EEPROM 的整个存储器区域。
3. 如果您有一个多应用的 Bootloader，并且它不使用 EEPROM，那么使用 Bootloader 的 Bootloadable 项目能够使用 EEPROM。EEPROM 被分为相同大小的容量分配到各个 bootloadable .cyacd 文件（例如，*_1.cyacd 占有前一半 EEPROM 而*_2.cyacd 占有后一半 EEPROM 数据）。

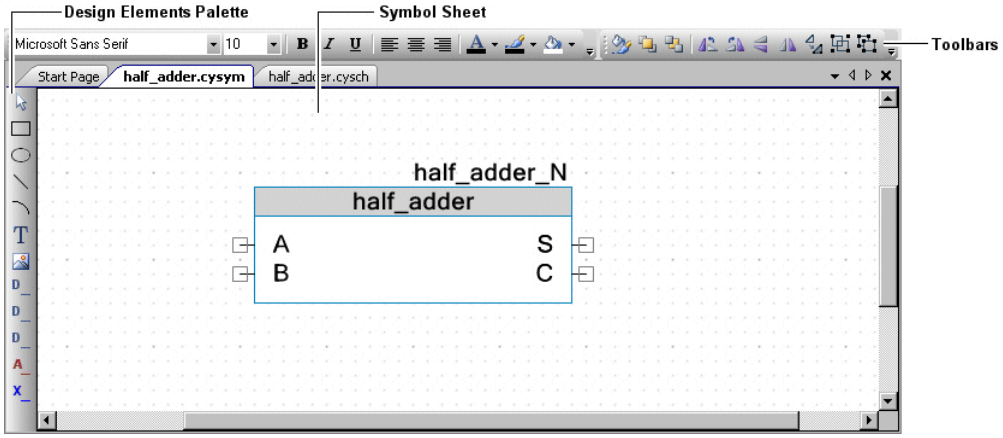
另外，请参考：

- 设计范围资源
- 存储器窗口
- Bootloader/Bootloadable 组件数据手册（在组件目录中提供）

符号编辑器

符号编辑器

使用符号编辑器您可以创建并编辑设计中使用的各组件。



创建组件的过程可能很复杂。遇到任何对话框时您都可以按下**[F1]**按键，这些主题将作为帮助内容提供给您。更多有关创建组件的详情，请参阅组件作者指南。

符号编辑器的主要主题包括：

- 符号表 — 您绘制各组件的图纸
- 设计元素控制板
- 符号编辑器右键菜单 — 符号编辑器的专用指令
- 普通工具栏 — 适用于设计输入工具的各条指令

本部分也介绍了有关符号编辑器的使用各主题：

- 创建新符号
- 符号向导
- 符号编辑器右键菜单
- 组件终端的使用
- 定义目录放置
- 创建符号参数
- 创建参数验证程序

创建新符号

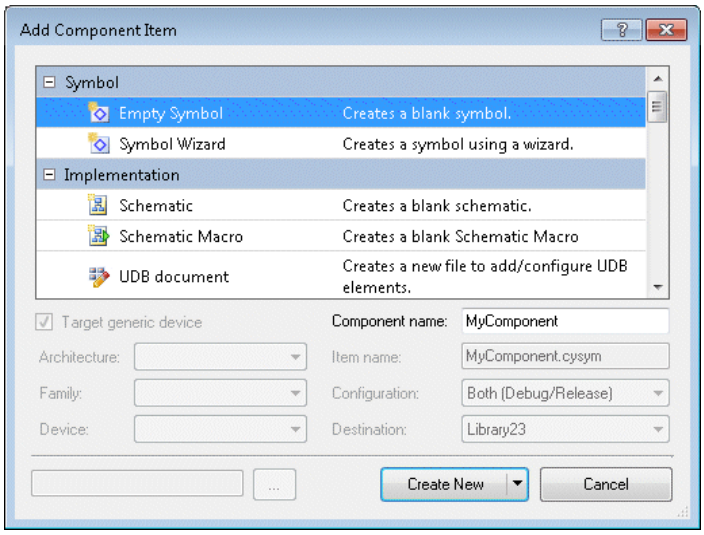
创建符号只是创建组件的一个方面。创建组件的过程可能很复杂。更多有关创建组件的详情，请参阅组件作者指南。

如果在一个项目中创建符号。首相要创建新项目或打开现有项目。

1. 打开项目后，请选择 Workspace Explorer（工作区浏览器）中的 **Components**（组件）选项卡。
2. 右击某个组件或项目，并选择 **Add Component Item...**

   另外，您也可以从 **Project**（项目）菜单中选择该项。

   这时会出现 Add Component Item（添加组件项）对话框。



3. 选择 Empty Symbol（空符号）图标。
4. 输入组件名称。
5. 指定 **Destination**（目标）。

   注意：只有从 **Project** 菜单中打开该对话框时，该选项才有效。
6. 点击 Create New。

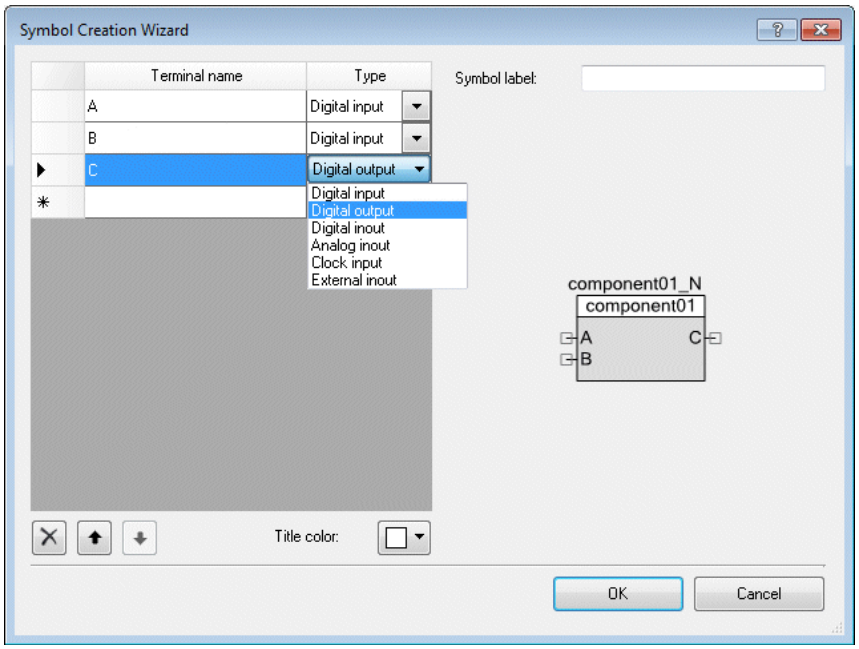
该符号将被添加到现有组件内，或者新组件将包含新符号。

另外，请参考：

- 组件创建指南
- 库组件项目
- 添加组件项
- 符号向导
- 从原理图创建符号

符号向导

使用符号向导您可以创建基本符号并指定各终端的名称和类型。该向导也提供了一个符号预览以及用于更改标题颜色的选项。



创建符号只是创建组件的一个方面。创建组件的过程可能很复杂。更多有关创建组件的详情，请参阅组件作者指南。

如何打开符号向导：

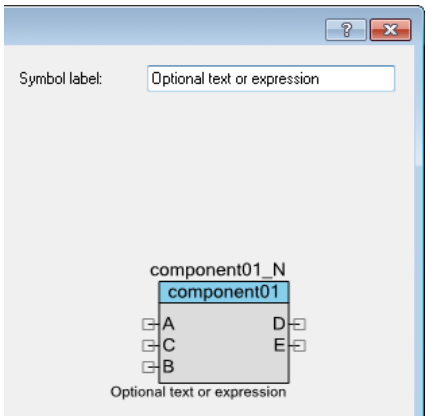
1. 请打开一个项目（或创建新的项目），然后选择 Workspace Explorer（工作区浏览器）中的 **Components**（组件）选项卡。
2. 右击某个组件或项目，并选择 **Add Component Item...**以打开 Add Component Item 对话框。
3. 选择符号向导图标并点击 **Create New**。

如何添加新终端：

输入**终端名称**，并为每个组件终端选择**类型**；使用不同行创建单独终端。

如何添加符号标签：

使用 **Symbol label**（符号标签）文本框添加将要显示在符号下面的可选文本或表达式。



注意：该字段通常用于输入一个表达式，以显示符号中的组件的单参数值（例如，参数名称 = `=$ParamName`）。更多有关各表达式的信息，请参考使用文本替换。

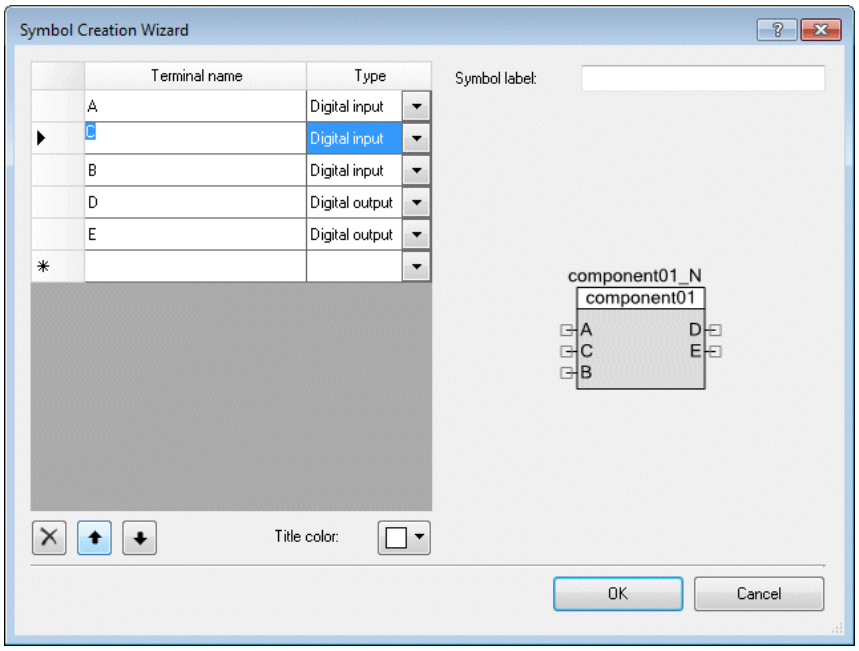
如何清除一个终端：

双击终端的行标头单元格。将出现一个对话框，用以确认清除操作。

您也可以选择一行，然后点击 **Delete** 按钮。

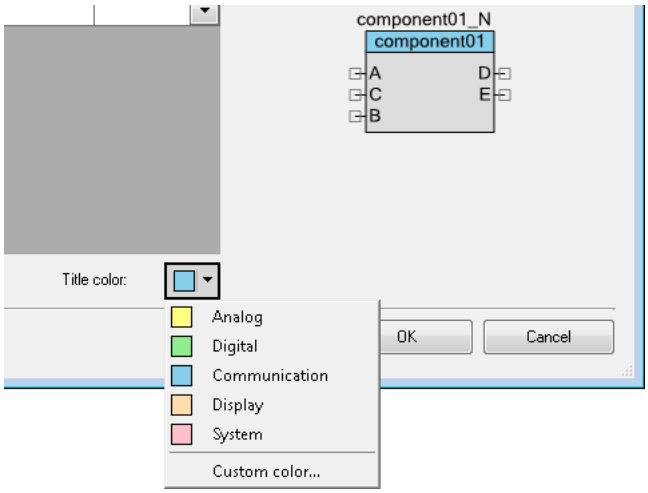
如何更改终端顺序：

选择一行，并使用 **Up Arrow**（向上箭头）和 **Down Arrow**（向下箭头）按钮重新排列该表中的终端顺序。这样会影响到各终端显示在符号上的情况，如预览所示。



如何更改标题颜色：

要想更改符号标题的颜色，请从 **Title color**（标题颜色）下拉菜单中选择一种颜色。有多种预定义的赛普拉斯颜色。您也可以选择一个自定义的颜色。



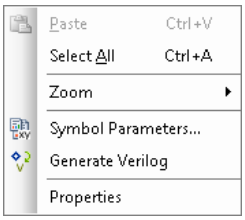
另外，请参考：

- 组件创建指南
- 库组件项目
- 添加组件项
- 创建新符号
- 从原理图创建符号
- 如何使用文本替换

符号编辑器右键菜单

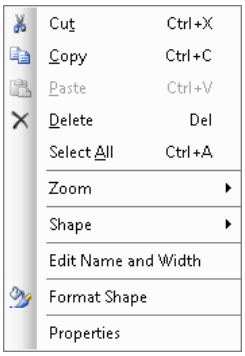
符号编辑器右键菜单（右击后出现）包含各种指令。根据您右击的位置 — 图纸或组件/元素，可用的指令会存在区别。下面是可用的指令：

在图纸上：



- **Paste**（粘贴） — 与 Standard Toolbar（标准工具栏）中的指令相同。
- **Select All**（全部选择） — 选择图纸上的所有对象。
- **Zoom**（缩放）— 与 View 菜单中的缩放指令相同。
- **Symbol Parameters**（符号参数） — 打开 Parameters Definition（参数定义）对话框。请查看创建符号参数.
- **Generate Verilog**（生成 Verilog 文件） — 根据符号定义生成 Verilog 文件。请查看生成 Verilog 文件。
- **Properties**（属性） — 打开 Properties 对话框。

在选定的对象上：



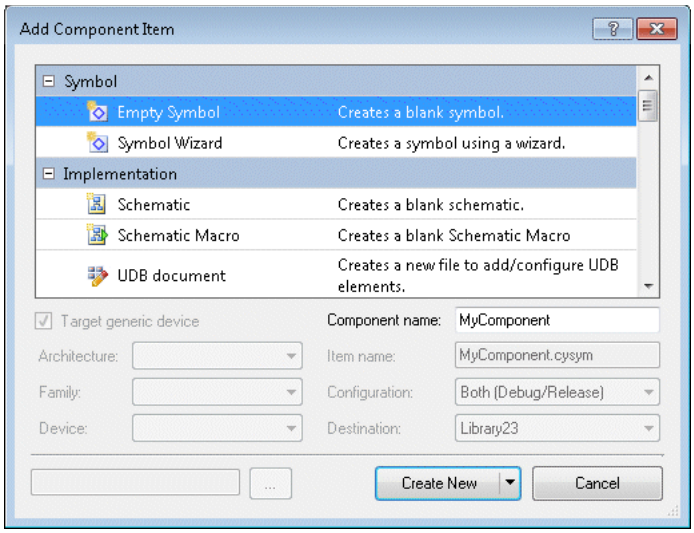
- **Cut, Copy, Paste, Delete**（剪切、复制、粘贴、删除）— 与标准工作栏的指令相同。
- **Select All**（全部选择） — 选择图纸上的所有对象。
- **Zoom**（缩放）— 与 View 菜单的缩放指令相同。
- **Shape**（形状）— 与通用的设计输入工作栏的造型指令相同。
- **Select**（选择）— 当在彼此的顶部上绘制两个或更多的对象时，通过该指令可以选择具体的对象。
- **Align**（对齐）— 选中两个或更多对象时，使用该指令您可以调整所选的形状：左、右、中心、顶部、中间和底部。
- **Edit Name and Width**（编辑名称和宽度） — 仅适用于终端，用于打开 Terminal Name 对话框。
- **Format Shape**（格式形状）— 打开 Format Shape 对话框来更改选中形状的各种特性。
- **Properties**（属性） — 仅适用于终端，用于打开 Properties 对话框。

另外，请参考：

- 符号编辑器
- 通用的设计输入工作栏
- 设计元素控制板
- 标准工具栏

添加组件项

“Add Component Item”对话框的主要功能是将新项目添加到某个组件内。通过使用该对话框可以在一个库项目内创建新组件。添加组件项和创建各组件使用的是各指令和程序的复合集。该主题会作为该对话框的帮组内容。更多有关创建组件的详情，请参阅组件作者指南。



该对话框提供了您能够添加到组件的各项类型的模板。根据您选择的各种类型以及想要将项目添加到某个组件还是创建新组件，可用选项会不一样。

要打开该对话框，请执行下述操作：

在 Workspace Explorer 的 **Components** 选项卡下，右键单击某个组件或项目，或直接选择 **Add Component Item...** 。

- 选择一个组件以将新的组件项添加到该组件内。
- 选择某个项目以在该项目内创建新组件。

您也可以通过使用 **Project** 菜单打开该对话框；但该菜单与右键菜单的区别在于它不会自动选定 **Destination** 字段。

要想使用该对话框，请执行下述操作：

Add Component Item 对话框包括以下各项：

Templates（模板）

Templates 字段显示的是可用组件项中不同类型的图标。当前可用的模板包括：

- **Symbol**（符号） — 包含符号模板，用于创建空符号或使用符号向导。另请查看创建符号。
- **Implementation**（实现） — 包含实现模板。请参考创建新模板。
- **API** — 包含 API 模板，用于创建 C 文件、头文件以及汇编文件。
- **Library**（库） — 包含一个库模板集，允许组件创造者添加各库，以用于不同的编译器工具链和配置（调试/释放）。
- **Misc** — 包含其他模板，用于创建文档文件、控制文件、XML 文件或其他文件类型。

除模板外，每个组件项还有一个简要说明。

Target generic device（通用的目标器件）

Target generic device 复选框用于禁用/使能 **Architecture**（架构）、**Family**（系列）和 **Device**（器件）下拉菜单。通过这些选项可以确定将新项目存储在项目层次中的位置。该字段仅适用于某些组件项类型，如实现、API 文件等。该复选框根会据您选择的不同模板项而进入有效状态。

- 选择 **Target generic device** 复选框后，可以创建组件级的组件项目并禁用其他选项；取消选定该复选框后会使能它们。
- 选择一个 **Architecture**，这样可以在子文件夹内创建架构（PSoC 3、PSoC 4 或 PSoC5，等）的组件项。
- 选择一个器件 **Family**，这样可以在子文件夹内创建一个器件系列（CY8C32、PSoC 4000、PSoC 4200 BLE、CY8C52LP 等）的组件项。
- 选择特定的 **Device** 器件型号，这样可以在一个子文件夹内为某个特定器件创建组件项。

Component Name（组件名称）

当您为某项目创建的新组件时，可以在 **Component Name** 字段中指定该组件的名称。当添加组件级的组件项时，该字段无效。

Item Name（项名称）

您可以使用 **Item Name** 字段指定某些组件项的名称。下列各组件项的名称均来自于 **Component Name**：

- 符号
- 原理图
- 控制文件
- Verilog 文件

配置

Configuration 字段仅适用于库模板文件。通过该字段，您可以将库文件指定为调试模式、释放模式还是两种模式。

Destination（目标）

只有使用 **Project** 菜单中的 **Add Component Item** 指令打开该对话框时，**Destination** 字段才有效。该字段是一个下拉菜单，通过它您可以选择将要添加组件项的项目或组件。请注意，如果您选择了某个项目，那么 **Component Name** 字段有效。但如果您选择了某个组件，**Component Name** 字段将无效。

Create New 和 Add Existing

默认情况下，显示 **Create New** 按钮。如果您点击了该按钮，那么 PSoC Creator 将创建属于已选类型的新文件，并将该文件添加到该组件内。

该按钮包含一个下拉切换  ，通过它可切换到 **Add Existing** 按钮。如果您进行切换存在，导航按钮**[...]**进入有效状态，用以选定属于已选类型的现有文件。然后，当您点击 **Add Existing** 按钮时，已选文件将被复制到组件文件夹，被被添加到该组件内。可以在复制过程中重命名各文件，以确保它们的名称符合某些文件类型对文件名的要求。

另外，请参考：

- 组件创建指南
- 库组件项目
- 符号向导
- 从原理图创建符号
- 工作区浏览器
- 创建新的原理图
- 创建新符号

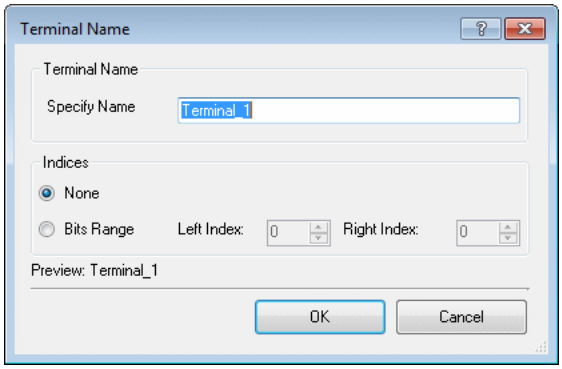
组件终端的使用

通过原理图的设计元素控制板中的终端工具，您可以绘制数字输入、输出、输入输出以及模拟和外部终端。

在创建组件过程中，当您在符号编辑器中创建某个符号时可以使用组件终端。更多有关创建组件的信息，请参阅组件创建指南。

要想放置一个终端，请进行下述操作：

从设计元素控制板中选择相应的 **Terminal**（终端）工具，然后点击图纸。这时会出现 Terminal Name（终端名称）对话框。



要想重新命名一个终端：

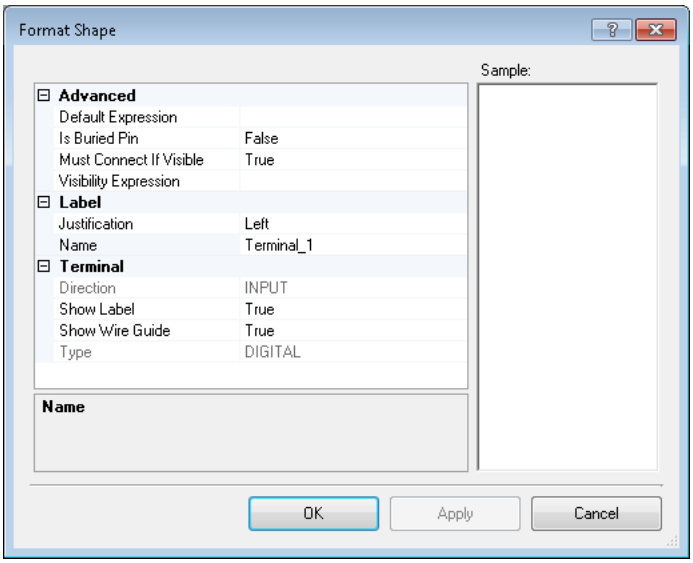
右击某个终端并选择 **Edit Terminal Name**（编辑终端名称）。

这时会出现 Terminal Name（终端名称）对话框出现。

使用该对话框指定相应终端的名称和/或指数。

要想显示/隐藏一个终端标签，请进行下述操作：

1. 右击该终端并选择 **Format Shape**（格式形状）以打开 Format Shape 对话框。



2. 通过将 **Show Label**（显示标签）属性分别更改为 true/false（真/假）可显示/隐藏该标签。
3. 点击 **OK**。

要想清除某个终端，请进行下述操作：

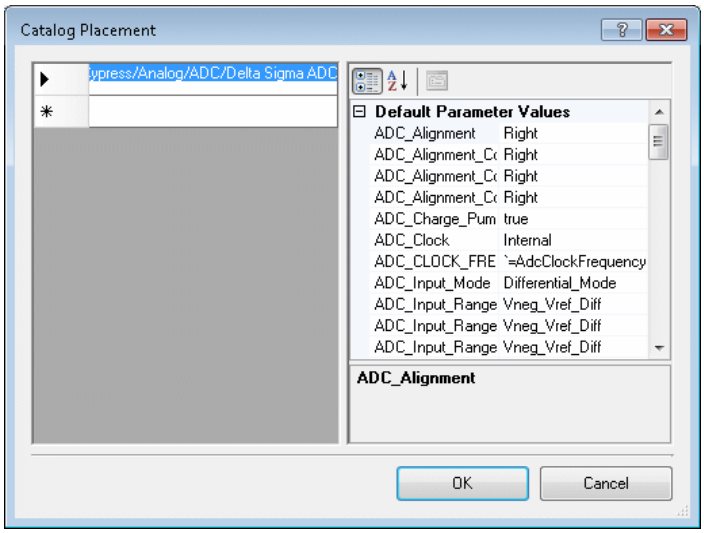
请选择终端并按下**[Delete]**键或点击。

另外，请参考：

- 组件创建指南
- 符号编辑器
- 设计元素控制板
- Terminal Name 对话框
- Format Shape 对话框

定义目录放置

创建组件时，您可以使用 Catalog Placement（目录放置）对话框来定义符号在组件目录中各种树和选项卡中显示的情况。

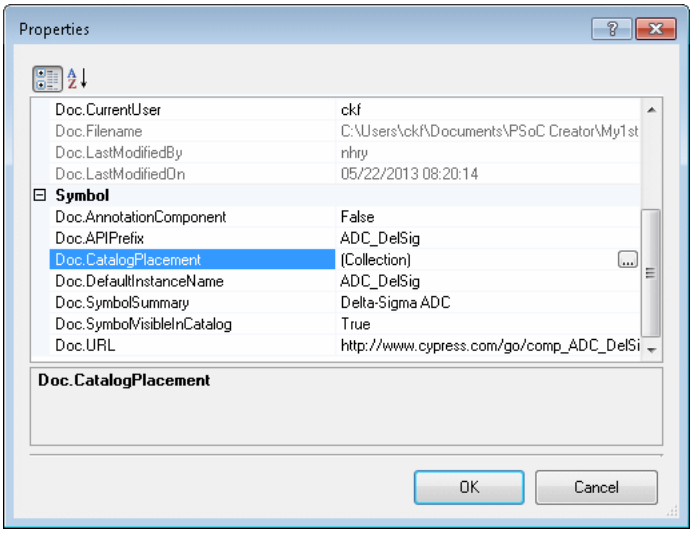


创建组件的过程可能很复杂。关于如何设置该对话框，您可以按下**[F1]**键，从 Help 文件中找到相关信息。更多有关创建组件的详情，请参阅组件作者指南。

要想打开 Catalog Placement 对话框，请进行下述操作：

1. 右击符号图纸并选择 **Properties**。

   Properties 对话框出现。



2. 点击 **Doc.CatalogPlacement** 字段中的**[...]**按钮，以打开 Catalog Placement 对话框。

要想定义目录放置，请执行下述操作：

使用以下语法：

```
t/x/x/x/x/d
```

该语法中每部分的含义如下：

- t — 选项卡名称。显示在 Component Catalog 对话框中的选项卡，它们是按字母（不分大小写）排列的。

- x — 选项卡下树的节点。至少要有一个节点。
- d — 符号的显示名称（可选）。

如果您未指定显示名称，则使用符号名称；但您必须使用 t/x/语法。

如果您未定义已给符号的 **Doc.CatalogPlacement** 属性，它将默认显示在 **Default** 选项卡下。

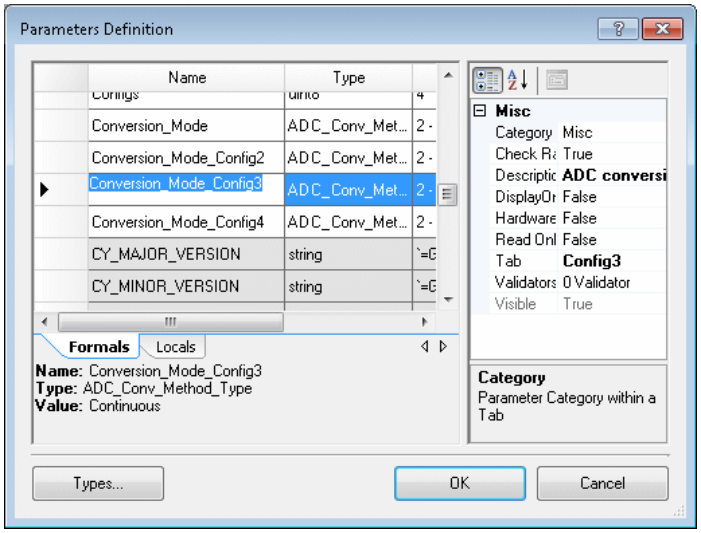
如果您想在多目录树中显示该符号，请将单独的语法字符串输入到该对话框中不同的行内。

另外，请参考：

- 组件创建指南
- 库组件项目
- 组件目录
- 属性对话框

创建符号参数

创建组件时，您可以使用 Parameters Definition 对话框来创建符号的一个或多个参数集。



创建组件的过程可能很复杂。关于如何设置该对话框，您可以按下[**F1**]键，从 Help 文件中找到相关信息。更多有关创建组件的详情，请参阅组件作者指南。

要打开该对话框：

右击符号图纸并选择 **Symbol Parameters**（符号参数）图标。

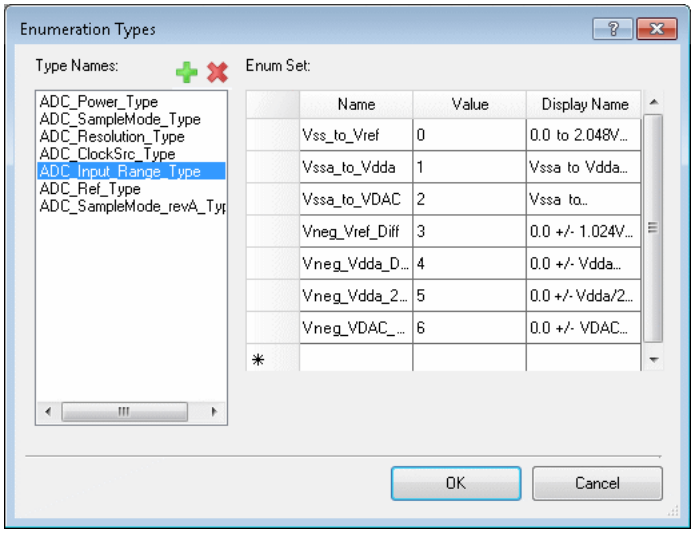
要想创建参数，请执行下述操作：

您可以为每个符号创建任意多个参数。

1. 在对话框左侧的参数表内，定义某个参数的 **Name**（名称）、**Type**（类型）和 **Value**（数值）。
2. 在对话框右侧定义每个参数的属性。
3. 要想添加其他参数，请输入到该参数表的下一行内。

要想创建枚举类型，请进行下述操作：

点击 **Types...**按键，打开 Enumeration Types（枚举类型）对话框。



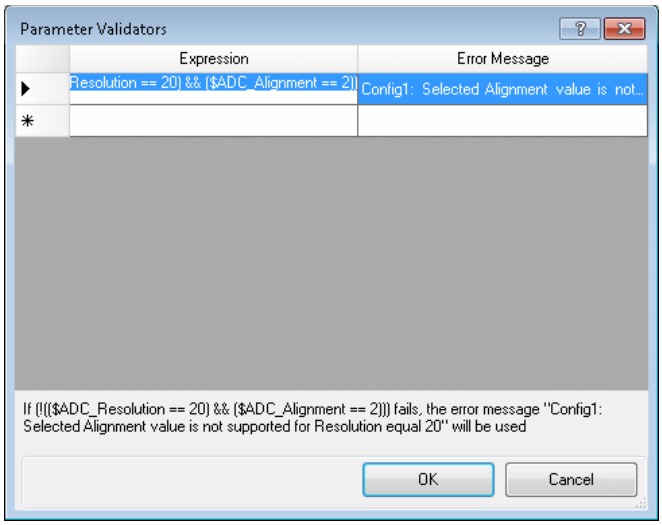
更多有关信息，请参考枚举类型。

另外，请参考：

- 组件创建指南
- 符号编辑器
- 枚举类型

创建参数验证程序

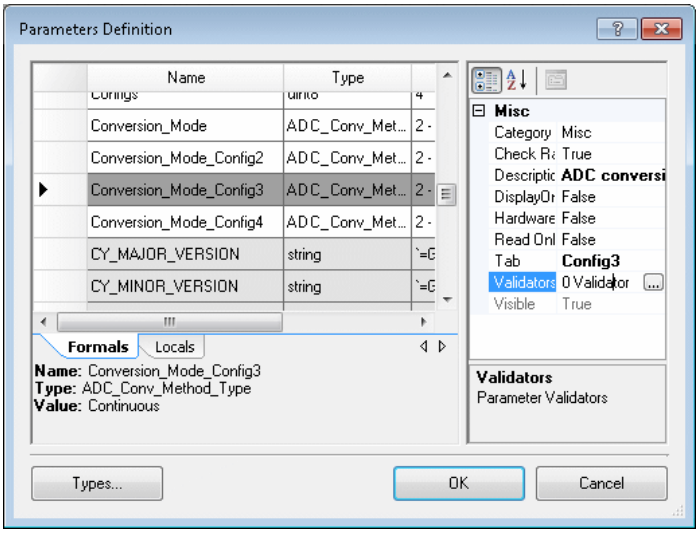
在创建组件时，您可以使用 Parameter Validators 对话框来创建一个或多个参数验证程序。



创建组件的过程可能很复杂。关于如何设置该对话框，您可以按下**[F1]**键，从 Help 文件中找到相关信息。更多有关创建组件的详情，请参阅组件作者指南。

要想打开 Parameter Validators 对话框，请进行下述操作：

1. 在 Parameters Definition 对话框上，选择需要添加验证程序的参数。
2. 然后选择 **Misc** 下的 **Validators** 字段，并点击省略符号按钮。



要想添加一个验证程序：

1. 请在 **Expression** 字段中，将表达式输入到验证程序表中的第一行内。请参考组件创建指南。
2. 在 **Error message**（错误信息）字段中，输入一条信息以显示是否尚未满足验证检查。
3. 要想添加其他验证程序，请输入到该验证表的下一行内。
4. 点击 **OK**，关闭该对话框。

另外，请参考：

- 组件创建指南
- 符号编辑器
- 创建符号参数

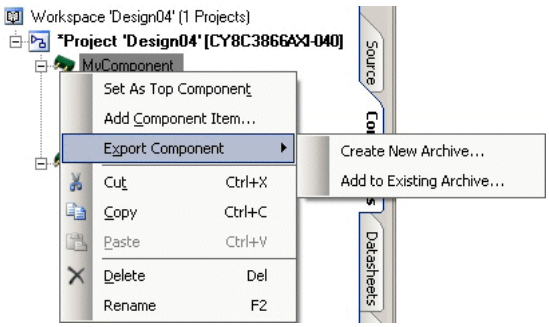
导出组件

当您完成开发一个或多个组件后，您可以导出这些组件，并不需要提供整个项目或整个库。这样您可以执行独立于组件库的分配方法，并能够更加容易控制您自己的库和组件。

如果您想将该组件添加到某个项目/库内，请参考存档工作区/项目。

要想导出一个组件，请进行下述操作：

1. 在工作区浏览器对话框中的 **Components** 选项卡下，右击该组件以显示右键菜单，然后选择 **Export Component**（导出组件）。



2. 再选择 **Create New Archive...**（创建新存档）以将其导出作为一个新的组件存档。
   - 该选项打开 Save As 对话框，将组件命名为 *comp_archiveXX* 并按默认的.cycomp 文件存储，其中 XX 表示您每次导出组件时递增的数值。
   - 重新命名该文件并选择所需位置。
   - 点击 **Save** 按钮。
3. 选择 **Add to Existing Archive...**，以更新现有的导出组件存档。使用该选项，您可以更新现有的组件，另外还能够将其他组件添加到现有的.cycomp 文件内。
   - 该选项打开 Save As 对话框。
   - 导航到现有的需要更新的.cycomp 文件所在的位置。
   - 然后选择.cycomp 文件并点击 **Save**。

.cycomp 文件包含了已选组件所带有的所有文件，而在导入组件过程中，该文件被 PSoC Creator 使用。

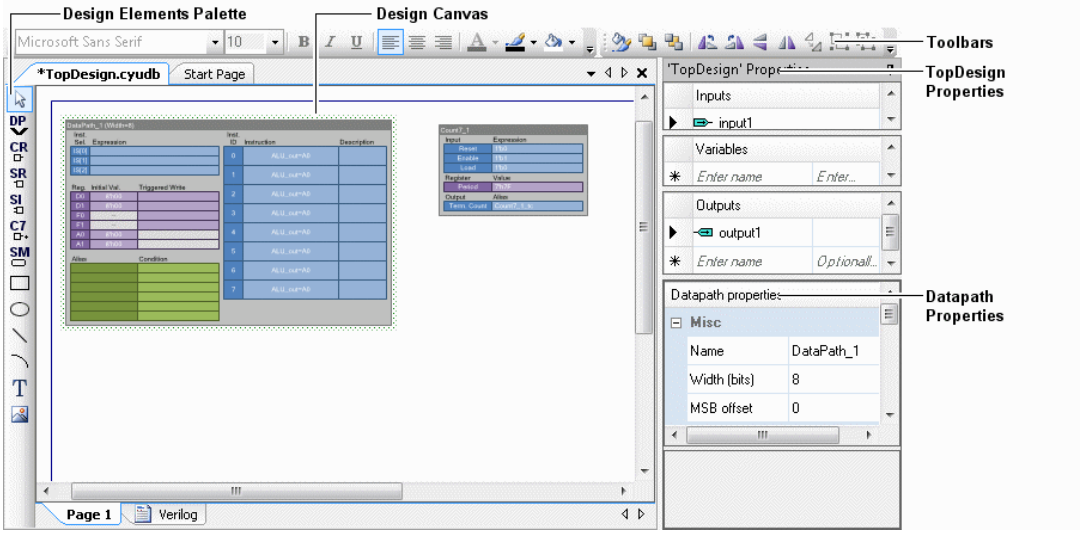
另外，请参考：

- 存档工作区/项目
- 工作区浏览器
- 导入组件

UDB 编辑器

UDB 编辑器

通用数字模块（UDB）编辑器是一个用于创建 PSoC 组件的图形工具。简单易用的编辑器提供了一个实现并配置 UDB 资源的普通方法。



注意：UDB 编辑器不会使能 100%的 UDB 功能，或显著的大部分的当前设计。但它能够提高您使用 UDB 资源的能力。该编辑器不会取代数据路径配置工具（对 UDB 进行操作时，该工具不单单是一个专家级的工具）。更多有关数据路径配置工具和 UDB 资源的信息，请参考组件创建指南。更多有关 UDB 编辑器的信息，请参考 UDB 编辑器指南。

UDB 编辑器的主要组件包括：

- 设计图纸 — 您绘制设计的图纸
- Verilog — 显示了设计图纸所生成的 Verilog 代码。目前，UDB 编辑器仅支持 Verilog；将来它会支持其他语言。所有表达式都必须遵循 Verilog 语法。
- UDB 设计元素控制板
- UDB 属性
- 普通设计输入工具栏 — 适用于设计输入工具的各项指令
- 右键菜单 — 通过右击便可使用各项指令

UDB 编辑器将下列 UDB 功能作为可编辑的模块：

- 数据路径（DP）
- 控制寄存器
- 状态寄存器
- 状态中断寄存器

- Count7
- 状态机

另外，请参考：

- 组件创建指南

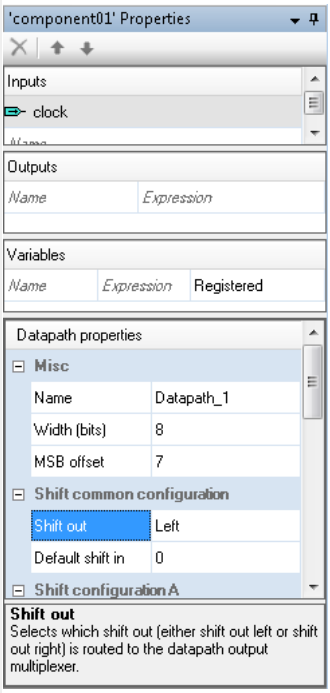
UDB 设计元素控制板

垂直工具栏与其他设计输入工具使用的设计元素控制板相同。它们之间的主要区别是该工具栏包含了将要放置在 UDB 编辑器上的 UDB 资源。各按钮如下所示：

- – UDB 数据路径
- – UDB 控制寄存器
- – UDB 状态寄存器
- – UDB 状态寄存器中断
- – UDB Count7
- – UDB 状态机

UDB 属性

UDB 编辑器的右侧显示了属性的几部分。



- **Inputs**（输入） — 它是组件的输入列表（相当于原理图输入终端）。始终存在时钟输入，并且不能对其进行修改。但可以对所有其他输入进行需要的添加、清除或重新命名等操作。
- **Outputs**（输出） — 它是与组件相连的输出列表（相当于原理图输出终端）。这些输出与输入的操作完全相同，但它们没有固定/不变的默认设置。
- **Variables**（变量） — 通过使用这些变量，您可以创建被命名且可重复使用的组合逻辑和注册逻辑。单独的图标用于指出该变量的类型为组合的还是注册的。
- **Datapath Properties**（数据路径属性） — 这些属性对数据路径的所有指令（而不是一个特殊指令）产生影响。这些属性包括：
  - 其他配置
  - 移位普通配置
  - 移位配置 A/配置 B
  - 可配置的比较器输入
  - 掩码位
  - FIFO

工具栏：

工具栏包含以下指令：

- **Delete Row**（清除行） — 清除某个属性表中的已选行。
- **Move Up/Down**（上移/下移） — 在表中上移或下移所选行。

如何添加一个输入/输出/变量：

点击显示为“Enter Name”（输入名称）的单元，输入名称并按下**[Enter]**按键。

如何清除一个输入/输出/变量：

点击某行或按下**[Delete]**按键或点击**[Delete Row]**按钮。

如何编辑数据路径属性：

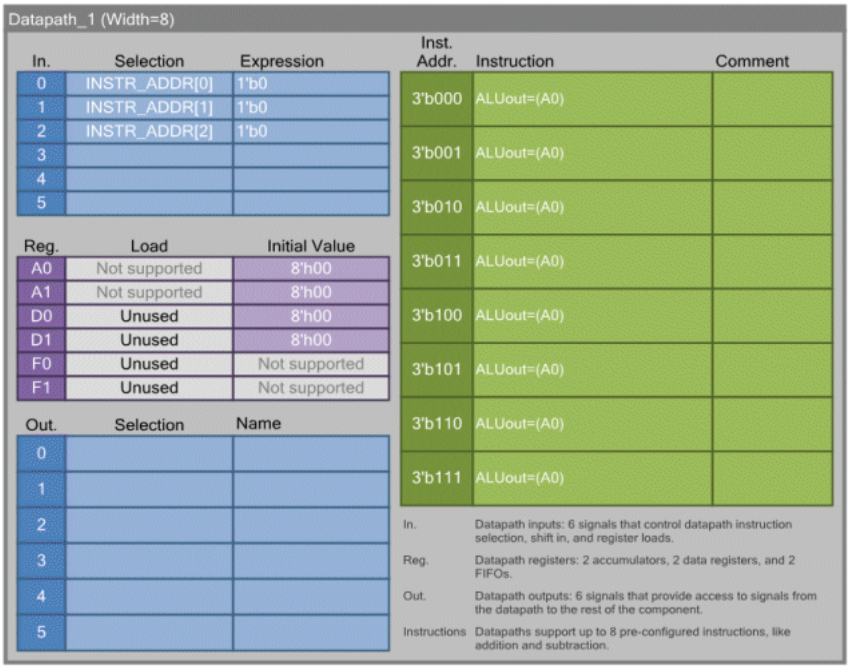
1. 您必须将数据路径资源放置在图纸上。选择它可使能各属性。
2. 点击某个单元以显示它的特殊属性，并从下拉菜单中选择某个值或向该字段中输入某个值。

另外，请参考：

- 组件创建指南
- UDB 编辑器
- UDB 数据路径
- 原理图终端的使用

UDB 数据路径

UDB 数据路径元素用于配置 PSoC 器件中的数据路径资源。该元素包含了某些表格，用于编辑各输入、寄存器、输出和指令。



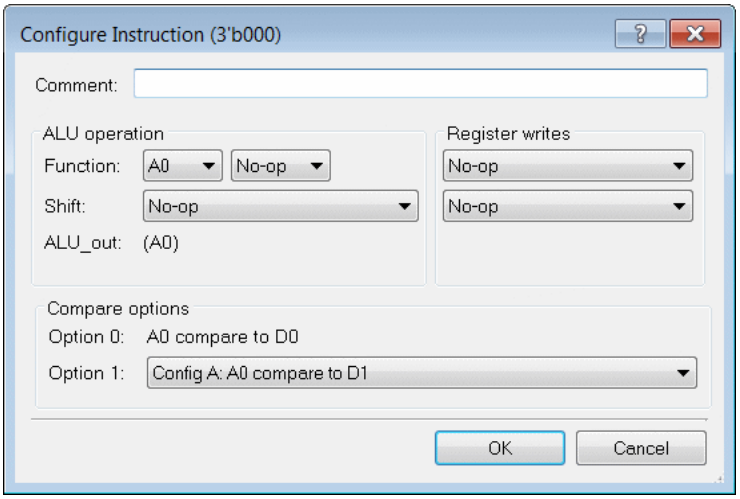
更多有关 UDB 数据路径的信息，请参阅组件创建指南。

如何放置一个数据路径：

点击 UDB 设计元素控制板中的数据路径图标，然后将该实例拖放到图纸上。

如何配置各输入、寄存器、输出和指令：

1. 双击数据路径实例中的某个区域，以打开该区域的 Configure 对话框。



注意：数据路径的每个区域都包含不同的配置对话框。更多有关信息，请参考配置对话框的说明。

2. 从下拉菜单中选择一个合适的值或在该字段中输入某个值。
3. 点击 **OK**，关闭该对话框。

如何配置通用数据路径属性：

要想更改数据路径名称和其他属性，请确保已经选择了数据路径实例，这样才能使能设计图纸右侧的 UDB 属性项。

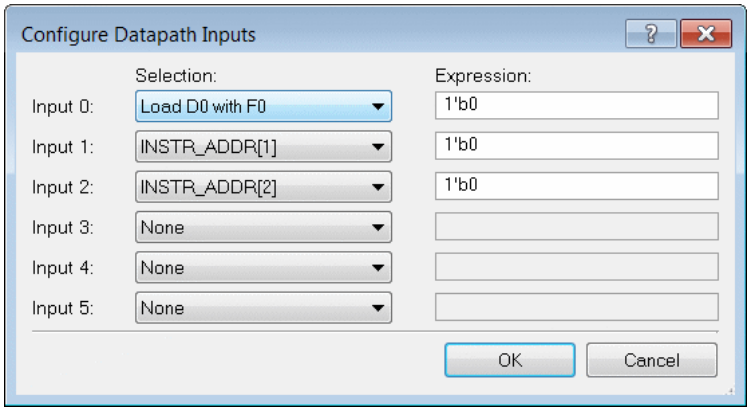
在此处，您可以选择或输入某个值来编辑各项属性。

配置对话框的说明：

数据路径元素包含了以下配置对话框：

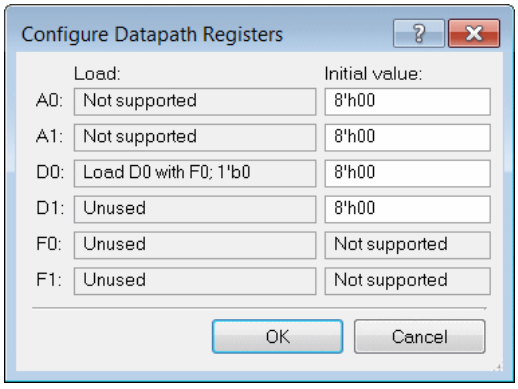
输入

该对话框包含的各字段用于选择数据路径输入和分配一个输入表达式。每个数据路径共有六个输入。通过使用该对话框，可以同时输入所有受支持的值。



寄存器

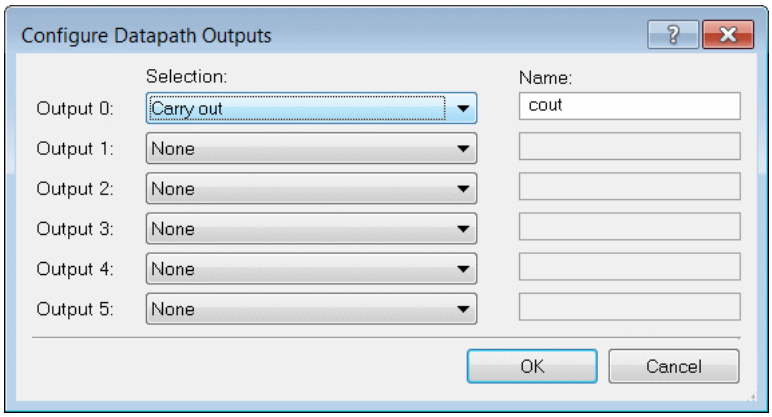
该对话框包含了许多字段，用于输入加载寄存器初始值。通过使用该对话框，可以同时输入所有受支持的值。



注意：Load 列中的值来自输入对话框的各选项；它们是仅供参考的只读字段。

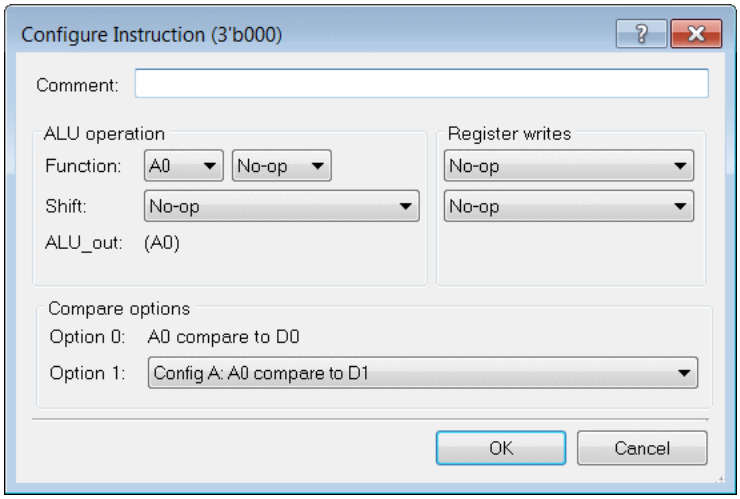
输出

该对话框包含的多个字段用于输入数据路径输出和分配条件。每个数据路径共有六个输出。通过使用该对话框，可以同时输入所有受支持的值。



指令

该对话框包含的多个字段用于输入各种数据路径指令。每个数据路径共有八条指令，可以单独配置每条指令。



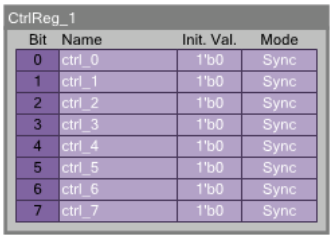
每条指令分为三部分：ALU 运算、寄存器写操作和比较选项。ALU 运算用于确定该指令周期进行的算术运算符或 Boolean 运算。寄存器写操作用于将下一条指令周期所使用的值加载到 A0 和 A1 内。比较选项用于设置比较器 0 和比较器 1 所采用的比较运算符。更多有关信息，请参考《组件创建指南》。

另外，请参考：

- 组件创建指南
- UDB 编辑器

UDB 控制寄存器

CPU 使用控制寄存器将各条指令发送给组件。每个控制寄存器共有八个可用位。可以在该设计中使用这些位来控制组件操作的特性。



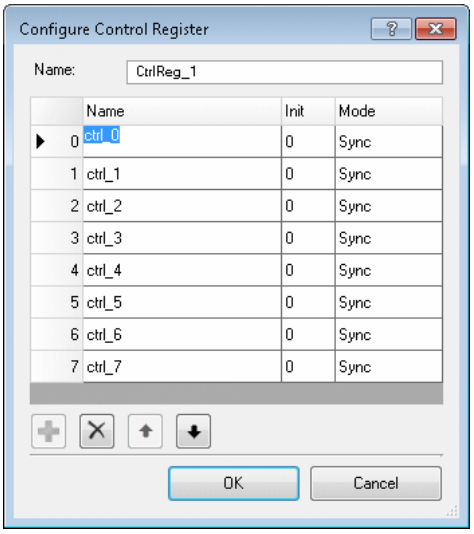
更多有关控制寄存器和 UDB 编辑器的信息，请参考组件创建指南。

如何放置一个控制寄存器：

点击 UDB 设计元素控制板中的控制寄存器图标，然后将该实例拖放到图纸上。

如何配置各位：

1. 双击控制寄存器实例，以打开 Configure 对话框。



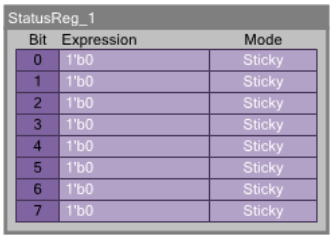
2. 输入 **Name**（名称）字段，并选择 **Init** 和 **Mode** 的值。
3. 点击 **OK**，关闭该对话框。

另外，请参考：

- 组件创建指南
- UDB 编辑器
- 控制寄存器组件数据手册（在组件目录中提供）

UDB 状态寄存器

CPU 通过使用状态寄存器可以从组件中读取硬件信号。单状态寄存器中提供的八位可用，并且 CPU 能够查看组件中的信号。



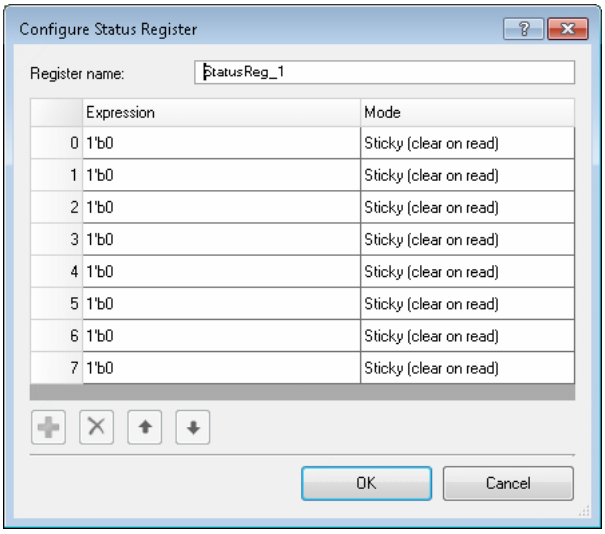
更多有关状态寄存器和 UDB 编辑器的信息，请参考组件创建指南。

如何放置一个状态寄存器：

点击 UDB 设计元素控制板中的状态寄存器图标，然后将该实例拖放到图纸上。

如何配置各位：

1. 双击状态寄存器实例，以打开 Configure 对话框。



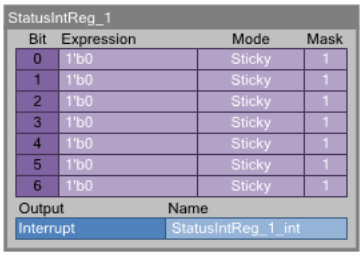
2. 输入一个 **Expression**（表达式），然后选择 **Mode** 值。
3. 点击 **OK**，关闭该对话框。

另外，请参考：

- 组件创建指南
- UDB 编辑器
- 状态寄存器组件数据手册（在组件目录中提供）

UDB 状态中断寄存器

通过使用状态中断寄存器可以从状态位生成可屏蔽中断。其中的七位可以作为输入使用，另外一位可以作为中断输出使用。



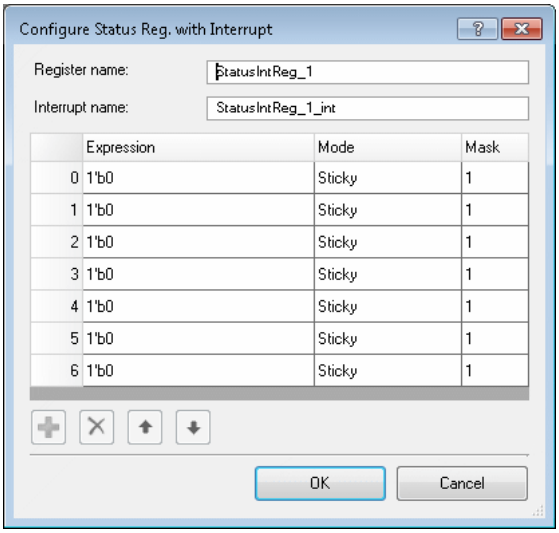
更多有关状态中断寄存器和 UDB 编辑器的信息，请参考组件创建指南。

如何放置一个状态中断寄存器：

点击 UDB 设计元素控制板中的状态中断寄存器图标，然后将该实例拖放到图纸上。

如何配置各位：

1. 双击状态中断寄存器实例，以打开 Configure 对话框。



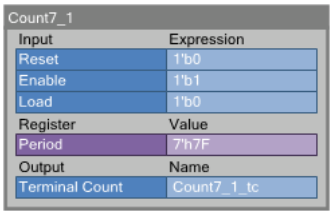
2. 输入一个 **Expression**（表达式），然后选择 **Mode** 和 **Mask** 值。
3. 点击 **OK**，关闭该对话框。

另外，请参考：

- 组件创建指南
- UDB 编辑器
- 状态寄存器组件数据手册（在组件目录中提供）

UDB Count7

Count7 计数器是一个七位的递减计数器，当需要一个包含三至七位的计数器时，应该使用该计数器。与基于 PLD 或数据路径的计数器设计相比，该计数器可节省更多的资源。



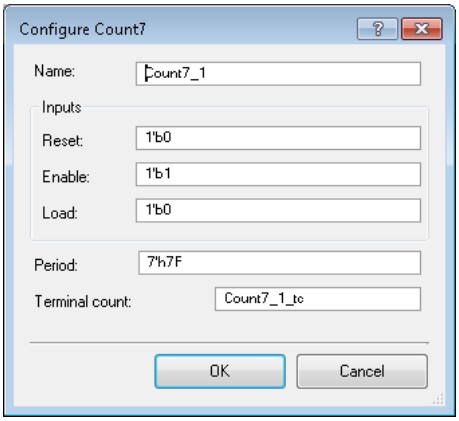
更多有关 Count7 和 UDB 编辑器的信息，请参考组件创建指南。

如何放置 Count7：

点击 UDB 设计元素控制板中的 Count7 图标，然后将该实例拖放到图纸上。

如何配置各位：

1. 双击 Count7 实例，以打开 Configure 对话框。



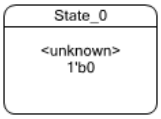
2. 输入 **Name**（名称）、**Inputs**（输入）、**Period**（周期）和 **Terminal**（终端）的计数值。
3. 点击 **OK**，关闭该对话框。

另外，请参考：

- 组件创建指南
- UDB 编辑器
- Count7 组件数据手册（在组件目录中提供）

UDB 状态机

状态机是使用 PLD 实现的控制逻辑。通过该状态机，可以将控制信号发送到您设计中的元素内，并能够跟踪硬件中进行的各种操作。



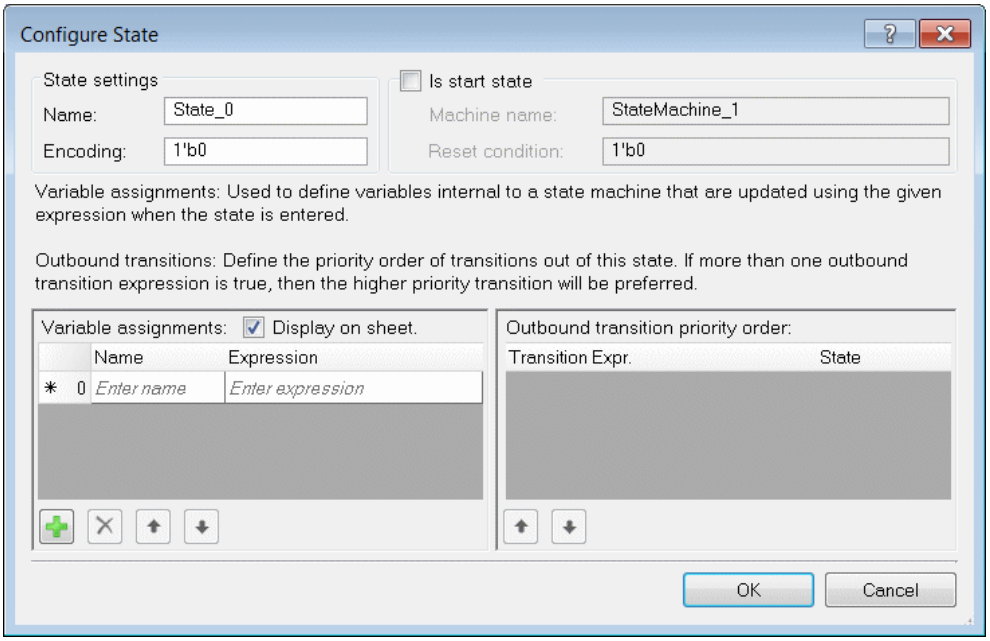
更多有关状态机和 UDB 编辑器的信息，请参考组件创建指南。

如何放置状态机：

点击 UDB 设计元素控制板中的状态机图标，然后将该实例拖放到图纸上。

如何配置状态机：

1. 双击状态机实例，以打开 Configure 对话框。



2. 向各字段中输入信息。

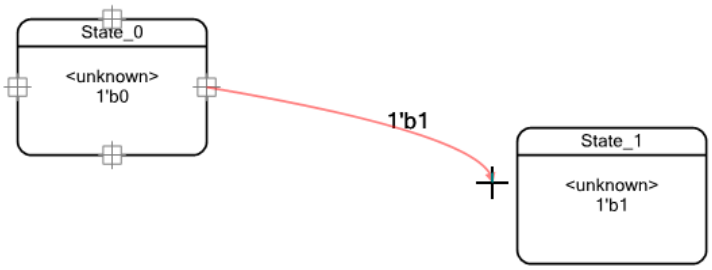
   有关这些字段的详细说明，请参考组件创建指南。注意：

   - 状态 **Name**（名称）必须是全局唯一的。
   - 在状态机内 **Encoding**（编码）必须是唯一的。
   - **Machine name**（机器名称）必须是全局唯一的；当状态机没有启动状态时，该字段被自动填充。

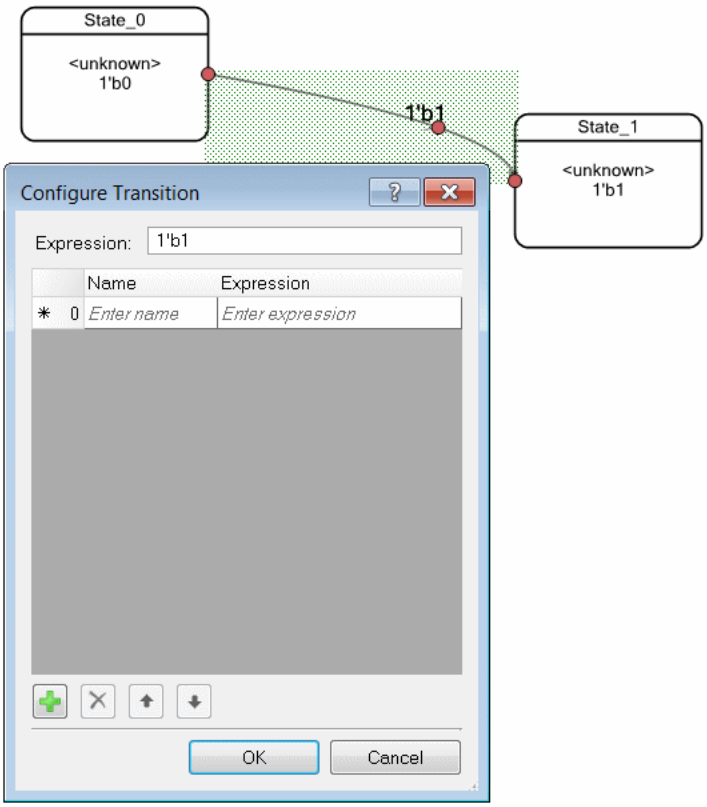
3. 点击 **OK**，关闭该对话框。

如何添加状态转换：

1. 将光标悬停在其中一个状态上，可以查看转换定位点。
2. 点击一个定位点，并拖放以开始绘制状态转换。



3. 继续拖放到其他状态以实现连接。当您释放鼠标时，将出现 Configure 对话框。

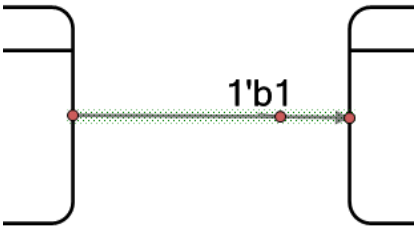


4. 输入表达式并按要求添加更多表达式。
5. 点击 **OK**，关闭该对话框。

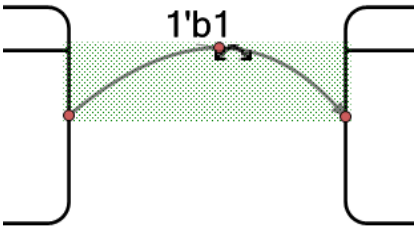
   要想再次打开该对话框，需要双击转换线。

如何调整转换圆弧：

1. 点击转换线以查看圆弧点。



2. 拖放圆弧的中间点，进行调整该圆弧。



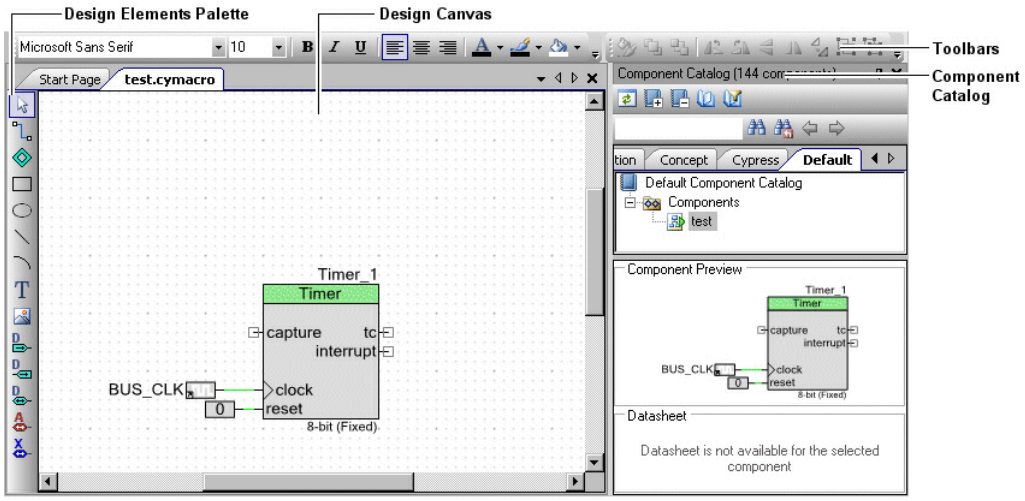
另外，请参考：

- 组件创建指南
- UDB 编辑器

其他工具

原理图宏编辑器

在使用该编辑器访问组件目录和其他相关的原理图工具，原理图宏编辑器与原理图编辑器是相同的，以此创建原理图。另外，通过使用这两种编辑器，您还可以 gde_symbol_editor_htm 创建显示在组件目录中的组件宏。



原理图宏通常由组件的作者创建，以简化编译组件的使用情况。组件的典型用法以得到准备并通过各宏实现。最终用户使用这些宏而不是使用基本组件。

原理图宏是微型的原理图。一个组件可以拥有多个宏。这些宏的等级可以为：普通级、架构级、产品系列级和器件级。可以将这些宏从组件目录拖放到原理图内。宏可以含有各个实例（包括用于该宏的组件）、终端和连线。

原理图宏的主要组件包括：

- 设计图纸 — 您绘制设计的图纸
- 设计元素控制板
- 普通设计输入工具栏 — 设计输入工具常用的各条指令
- 组件目录 — 用于您原理图的组件库

如何创建原理图宏：

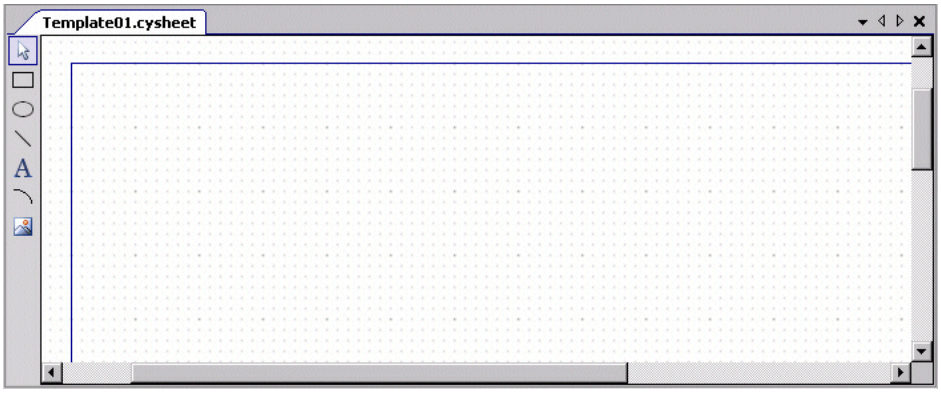
使用添加组件项对话框将原理图宏添加到组件内。

另外，请参考：

- 原理图编辑器
- 符号编辑器

工作表模板编辑器

通过工作表模板编辑器，您可以创建、编辑和保存原理图的工作表模板。



通过创建自己的工作表模板，您可以将标志、公司信息等信息添加到您的模板文件内，并能够在设计中显示这些内容。

要创建工作表模板：

1. 请依次点击 **File >** New File 。
2. 在 New File 对话框中，选择 Sheet Template 图标，然后点击 **OK**。

   Sheet Template Page Setup 对话框出现
3. 选择页面的大小和方向，输入页边距然后点击 **OK**。

   空工作表模板文件（.cysheet）被打开。

如何设计模板：

您可以使用所有绘制工具来设计满足您的规范的设计。

- 使用 Import Graphic（导入图形）工具可导入徽标或其他图像。
- 使用 Text（文本）工具输入公司名称、地址和其他重要信息。
- 使用形状工具创建图标、帧信息或您需要的任何其他形状。

如何使用模板属性：

Properties 对话框包含了有关模板信息的各种字段。右击该图纸并选择 **Properties** 以打开该对话框。

- 可以使用地址字段（Addr1、Addr2、Addr3）输入您公司的地址。
- 可以使用名称字段来输入您公司的名称。
- 可以使用当前用户字段来输入您的名称或公司 ID。
- 您可以向 DisplayName 字段内输入该模板的名称。另外，该名称还是您创建可执行新项目时显示在工作表目录中的名称。

如何保存模板：

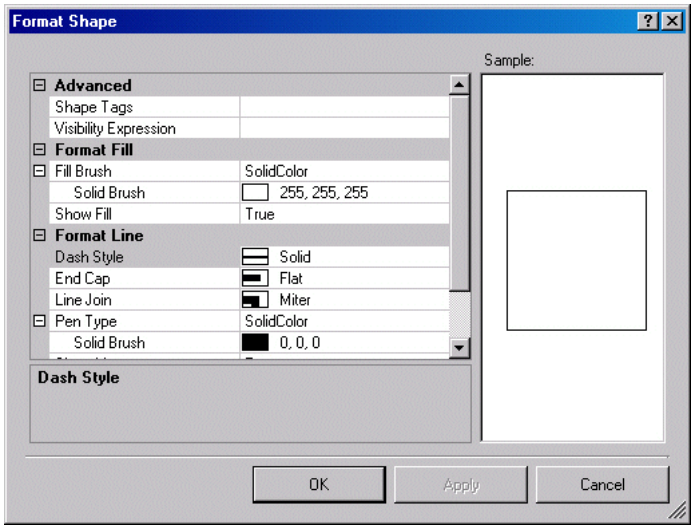
您可以将模板保存在 PC 的任何位置上。默认情况下，PSoC Creator 将您的模板定位在 <INSTALL_PATH>\templates\sheets 内。如果您将该模板保存在其他位置内，请使用工作表模板下的 PSoC Creator 设计输入选项指定该位置。

另外，请参考：

- 形状的使用
- 文本的使用
- 设计输入选项
- 属性

格式形状

通过 Format Shape 对话框，您可以更改图纸上任何形状（或形状集）的属性。



可用的属性类型会根据所选定的（各）形状而发生改变。例如，文本提供了字体、颜色、大小等信息；而画线则提供了宽度、笔型、后盖等信息。

普通形状属性：

在该对话框中，几乎所有普通形状的属性都是非常明显的，它们与您在文字处理或绘图程序中看到的形状格式相同。对于大多数属性而言，请从下拉菜单中选择一个值。

高级形状属性：

某些形状（如实例终端）具有以下各高级属性：

- **Default Expression**（默认表达式） — 将形状的默认值定义为<size>'b<value>，其中<size> = 位数，<value> = 二进制值。

- **Shape Tags**（形状标签） — 为一个或多个形状定义一个或多个标签，以用于形状自定义代码；请参考组件创建指南。
- **Visibility Expression**（可视性表达式） — 定义一个表达式，用于评估是否显示已选的形状。例如，在 UART 组件中，该属性通过 rts_n 和 cts_n 终端的变量$FlowControl 定义。如果$FlowControl 被设置为真，那么这些终端将显示在该原理图内；如果该变量被设置为假，这些终端将被隐藏。

另外，请参考：

- 组件创建指南
- 形状的用法
- Line（画线）的用法
- 通用的设计输入工作栏

通用的设计输入工作栏

有两种通用工作栏适用于图形设计输入编辑器（如原理图编辑器和符号编辑器）：格式和形状格式。

注意：在原理图编辑器右键菜单和符号编辑器右键菜单中也提供了很多指令。

格式：

下表列出并说明了设计输入的各种指令格式。

注意：这些指令仅适用于文本标签。要想调整代码编辑器的文本属性，请参考 Options 对话框。

图标	指令	说明
--	Font Style	用于选择字体系列和大小。
--	Font Size	用于选择字体大小。
	Bold	用于将已选文本设置为粗体字。
	Italic	用于将已选文本设置为斜体字。
	Underline	用于为已选文本添加下划线。
	Align Left	左对齐已选文本。
	Align Center	中心对齐已选文本。
	Align Right	右对齐已选文本。
	Font Color	为已选文本选择颜色。
	Line Color	为已选内容选择轮廓线颜色（或无颜色）。
	Fill Color	为已选内容选择填充颜色（或无颜色）。

形状格式：

下表列出并说明了设计输入形状的各种格式指令：

图标	指令	说明
	格式形状	打开 Format Shape 对话框以定义已选内容的各种特性。
	Bring to Front	将已选内容拖动到图纸顶部。
	Send to Back	将已选内容拖动到图纸底部。
	Rotate Left	将已选内容向左旋转。
	Rotate Right	将已选内容向右旋转。
	Flip Vertical	将所选对象按垂直方向翻转。
	水平翻转	将所选内容按水平方向翻转。
	转换为封闭的形状	将所选的直线和弧线连接成一个封闭的形状。
	组合	将所选的对象组合成一个单独的形状。
	取消组合	取消对先前组合的对象集进行的组合操作。

另外，请参考：

- 原理图编辑器
- 符号编辑器
- Line（画线工具）的用法
- 形状的使用
- 设计元素控制板

设计元素控制板

设计元素控制板是一个垂直的工作栏，位于图形设计输入编辑器的图纸的左侧，比如原理图编辑器和符号编辑器。控制板包含可放置在图纸中的多种设计元素。

普通元素：

下表列出并说明了在原理图编辑器和符号编辑器中均有效的通用设计输入形状指令：

图标	指令	快捷键**	说明
	选择	[Esc]	默认；允许选择各项。并能够退出其他工具。
	绘制矩形	[**R**]	用于绘制正方形和长方形。
	绘制椭圆	[**E**]	用于绘制圆形和椭圆形。
	绘制直线	[**L**]	用于绘制直线。

图标	指令	快捷键**	说明
	绘制弧线	[C]	用于绘制曲线。
	添加文本	[T]	用于绘制文本。
	插入影像	[M]	用于将一个影像插入到图纸内。允许影像格式为：BMP、GIF、EXIF、png、PNG 和 TIFF。

** 只有工作表图纸文档有效时，形状绘制工具的键盘快捷方式才可用。

原理图编辑器元素：

原理图编辑器控制板包括以下元素：

- **Terminals（终端）** — 用于绘制数字输入、输出、输入输出以及模拟和外部原理图终端。请参考原理图终端的用法和键盘快捷方式一节中介绍的内容。

  注意： 您正在编辑顶层原理图时，DEP 中不会显示原理图终端。只有针对组件创建原理图的实现时，才能找到原理图终端。

- **Wire（连线）** — 用于在原理图中绘制连线；快捷键是**[W]**。请参考连线的使用一节的内容。
- **Sheet Connector（工作表连接器）** — 用于在多个工作表中常被命名的连线间绘制连接器；快捷键是**[C]**。请参考使用多页和连接器一节的内容。

符号编辑器元素：

- **Terminals（终端）** — 用于绘制数字输入、输出、输入输出以及模拟和外部组件终端。请参考组件终端的用法和键盘快捷方式一节中介绍的内容。

正常模式与粘滞模式：

控制板上的所有指令（除 **Select**（选择）指令外）均有两种模式：正常模式和粘滞模式。在正常模式下，您只能将元素在图纸上放置一次，但在粘滞模式下，您可以多次将元素放置在图纸上。

- 要想激活正常模式，请单击某个元素；要激活粘滞模式，请双击某个元素。
- 通过按下**[Esc]**键或单击“**Select**”指令，可退出粘滞模式。

另外，请参考：

- 原理图编辑器
- 符号编辑器
- 键盘快捷方式

文本的使用

使用“文本”工具您可以在符号和原理图文档中输入文本。对于基本的形状（比如：矩形，圆形和直线等），您可以通过插入文本来标签它们。本节介绍了有关创建一个文本的基本内容，并介绍了更多先进的技术来使用文本替换。

注意： 连线和终端也可以有文本（被称为标签）。通过使用具体的对话框，可以控制这些连线和连线的标签。更多有关的信息，请参考信号名称、连线标签和名称以及终端名称节中的内容。

如何创建文本：

1. 在设计元素控制板中选择**文本工具**。 
2. 请点击原理图或符号图纸。

   文本框如下所示：

3. 键入文本，并点击图纸。

   在您键入文本时会显示它（用虚线框周围文本）。

如何编辑文本：

4. 双击文本标签，以对其进行编辑。

   使用创建新文本时的方法选择文本。

5. 编辑文本，并点击图纸。

   您编辑文本时会显示它（用虚线框周围文本）。

另外，请参考：

- 原理图编辑器
- 符号编辑器
- 设计元素控制板
- 连线标签和名称
- 原理图终端的使用
- 组件终端的使用
- 如何使用文本替换

如何使用文本替换

在符号和原理图中创建和编辑文本框（请参考文本的用法的内容）时，您可以使用文本替换，将键入的文本串替换为不同类型的信息。您可以输入多个表达式（例如，`=1+1`），这样会扩大相应值。

如何使用文本替换：

1. 创建一个文本框。
2. 使用下面的格式在框中键入：
   ```
   `=<expression>`
   ```
   该格式要求使用一个向后的撇号 [`]、等号[=]、表达式，并使用另一个向后的撇号[`]来结束。如果表达式包含一个参数，请将美元符号[$]添加到参数名称的前面。
   ```
   `=$<parameter>`
   ```

文档中现有的替换：

在文本中，根据您的需要，会存在多个替换点。您可以访问各参数、文档属性和实例名。另外，您还可以访问枚举的变量，并能够使用全表达语言。更多有关表达的信息，请参考组件创建指南中的内容。

下面是各可用参数，在多个文档中均能找到这些参数：

- 从参数和函数进行的文本访问，确实只使用在符号中。但在某些特殊情况中，可以将其使用在原理图中。不允许在宏中访问参数和函数。
- 符号中的文本能够清楚符号中所定义的所有参数。这些参数可以是内置的（比如：$INSTANCE_NAME），也可以是用户定义的。
- 原理图中的文本可以确定原理图符号中所定义的全部参数。例如，实现 UART_v1_20 组件的原理图可以使用 `=$PARAM` 扩展 UART 参数。
- 文本替换不会对设计产生语义影响。仅当用户可以读取它时，它才可用。
- 原理图编辑器中的文本仅能访问默认的参数值。
- 不存在符号的原理图中的文本没有任何参数。您仅能使用简单的表达式，例如 `=1+1`。
- 原理图宏中的文本没有权限访问任何参数。
- 文本表达式通过 ICyExprEval_v1 或 ICyExprEval_v2 定制器 API 可以访问特定组件内已创建的函数（请参考定制器 API 参考指南的内容）。函数的使用规则与参数的使用规则相同，具体如下：
  - 在符号中可以调用函数，这些函数由当前定义的组件定义
  - 在原理图中可以调用函数，这些函数由正在实现的组件定义
  - 在原理图宏中不能调用任何函数

替换示例：

下面各节为变量提供了可使用的不同示例。

符号参数

在符号中，您可以参考所定义的所有参数（但不能得到默认值）。符号被实例化到原理图中时，您会看到用户设置给该实例的参数。所有参数必须使用$符号。例如，参数 p1，类型 = int，默认值 = 42，则输入：

```
`=$p1`
```

此标签的显示文本为 42。

文档属性

在符号和原理图中，您可以获得文档属性。例如：

```
`=$Doc.CurrentUser`
```

此标签显示的文本是当前用户的名称或 ID。更多有关有效的不同属性的信息，请参考属性对话框。

实例名称

在符号中您可以获得实例名称。您常常能够在符号文档中获得“Inst_N”，但拖放到原理图中时它常为实例的正确名称。

```
`=$INSTANCE_NAME`
```

注意： Design Entry Options 对话框中有一个用来显示或隐藏未计算表达式的设置

枚举

在符号和原理图中，您可以获得任何枚举。例如，如果您定义了下面的枚举类型：

```
enum Foo
{
     Foo_VAL_1 = 1,
     Foo_VAL_2 = 2
};
```

您应该为该标签输入的变量如下：

```
`=Foo_VAL_1`
```

该标签的文本将显示为 Foo_VAL_1。

注意：枚举被转换成为文本串，并默认的文本串转换使用了枚举的文本名称。如果您想显示数字值，可以执行下面两种方法中的一种：

```
`=Foo_VAL_1 + 0`
```

或

```
`=cast(int, Foo_VAL_1)`
```

第一种方法利用+运算符会强制左侧和右侧为一个数字，并且 0 为加法单位元。第二种方法使用明确的转换操作，并强制枚举转换为 int。

复杂的表达式

您还可以执行复杂的表达式。例如：

```
`="p1=" . $p1 . " Current User=" . $Doc.CurrentUser . " Foo_VAL_1=" . cast(int, Foo_VAL_1)`
```

此标签显示的文本应该为：p1=42 Current User=xxx Foo_VAL_1=1

本实例中使用转换和文本串联结。您也可以在文本中嵌入尽可能多的替换文本串。例如：

```
Life is short, so eat `=$1` donuts and `=Foo_VAL_1 + 0` fig newtons.
```

此标签的文本会显示为：Life is short, so eat 42 donuts and 1 fig newtons。

另外，请参考：

- 原理图编辑器

- 符号编辑器
- 文本的使用
- 组件创建指南
- 定制器 API 参考指南
- 属性对话框
- 设计输入选项

Line（画线工具）的使用

原理图编辑器和符号编辑器都提供了 Line 工具用于绘制连线。本节介绍了多种绘制和使用连线的方法，包括：

- 绘制单条连线
- 绘制多条连线
- 移动某条连线
- 调整某条连线

请参考形状的使用。

如何绘制单条连线：

1. 请点击 **Draw Line**（绘制连线）工具。 
2. 将光标移动到连线的起始位置。
3. 点击并**按住**鼠标按键，将光标拖放到连线的终点。
4. 然后，释放鼠标按键。

如何绘制多条连线：

1. 双击 **Draw Line** 工具  以进入粘滞模式。
2. 将光标移动到连线的起始位置。
3. 点击并**按住**鼠标按键，将光标拖放到第一条连线的终点。
4. 然后，释放鼠标按键。
5. 将光标移动到第二条连线的起始位置。
6. 点击并**按住**鼠标按键，将光标拖放到第二个连线的终点。
7. 继续单击然后拖动，直到您完成所有连线为止。
8. 按下**[Esc]**或点击 **Select**  工具，可退出粘滞模式。

如何移动连线：

1. 点击 **Select** 工具。 
2. 点击并在鼠标位于连线上时按住鼠标按键然后移动，并将它拖放到目的位置。

   当鼠标位于连线上时，光标会变成一个手指形状。

如何调整连线：

1. 点击 **Select** 工具。 
2. 用鼠标按键点击连线，以进行调整。

   请注意，这时会显示句柄。

   - 要调整宽度或高度，请拖放某一边的句柄。
   - 要调整宽度和高度，请拖放角落句柄。

另外，请参考：

- 使用设计输入工具
- 设计元素控制板
- 形状的使用

形状的使用

使用符号编辑器或原理图编辑器时，您能够绘制多种不同的形状。形状和连线间有很多相似点；但连线上存在一些区别。本节介绍了形状的基本使用，包括：

- 绘制一个形状
- 移动一个形状
- 调整一个形状

请参考连线的使用。

如何绘制形状：

1. 点击形状工具，进行绘制。参见通用的设计输入工作栏。
2. 将光标移动到形状的起始位置。
3. 点击并**按住**鼠标按键，将光标拖放到形状的终点。
4. 然后，释放鼠标按键。

如何移动一个形状：

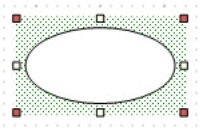
1. 点击 **Select** 工具。 
2. 点击并在鼠标位于形状上时按住鼠标按键然后移动，并将它拖放到目的位置。

   当鼠标位于连线上时，光标会变成一个手指形状。

如何调整形状：

1. 点击 **Select** 工具。 
2. 用鼠标点击形状，以进行调整。

   请注意，这时会显示句柄。



   - 要调整宽度或高度，请拖放某一边的句柄。
   - 要调整宽度和高度，请拖放角落句柄。

另外，请参考：

- 使用设计输入工具
- 设计元素控制板
- Line（画线工具）的用法

缩放

您可通过不同的窗口（比如：原理图编辑器和符号编辑器）实现放大和缩小操作。这几类窗口具有您想使用的缩放功能：

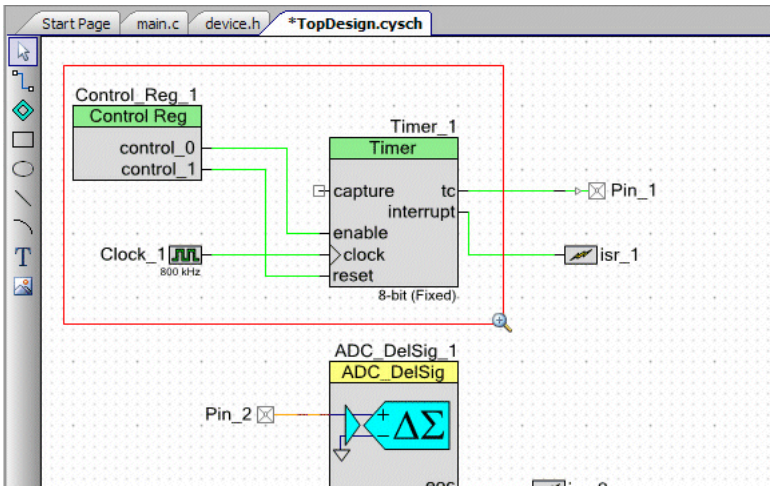
- 工具栏
- [Ctrl]键
- 右键单击

工具栏的用法：

- 要使用某个特定的缩放比例，请单击 **Zoom** 下拉菜单并选择您想要的缩放等级。您可以手动键入任何比例值；有效的缩放范围为 1% 至 2038%。
- 为了逐渐缩放，请点击 **Zoom In**（放大）或 **Zoom out**（缩小）。

[Ctrl]键的使用：

- 按住**[Ctrl]**键，并将一个框拖放到您想放大的区域内。在您拖放鼠标时，该流程会绘出一个红色的矩形。



释放鼠标后，图纸会立即缩放为选定的区域。

- 按住**[Ctrl]**键，然后使用鼠标上的滚轮逐步放大和缩小。
- 按住**[Ctrl]**键，并按下**[+]**键进行放大，使用**[–]**键进行缩小。

右键点击的使用

右键点击图纸，然后选择相应的**缩放**选项。

另外，请参考：

- 标准工具栏
- 键盘快捷方式

滚动

多个窗口提供了滚动条用于查看更多信息。有几种滚动方法：

- 需要时，拖放垂直或水平的滚动条。
- 使用鼠标滚轮，可以进行向上/向下滚动。
- 使用鼠标滚轮按住**[Shift]**键，可以进行左右滚动。
- 使用左箭头和右箭头进行垂直滚动；使用上箭头和下箭头进行水平滚动。

设计输入的保留字

下面内容由 PSoC Creator 设计输入工具保留。不能将其作为设计元素的名称，比如：连线、终端、实例、参数等。

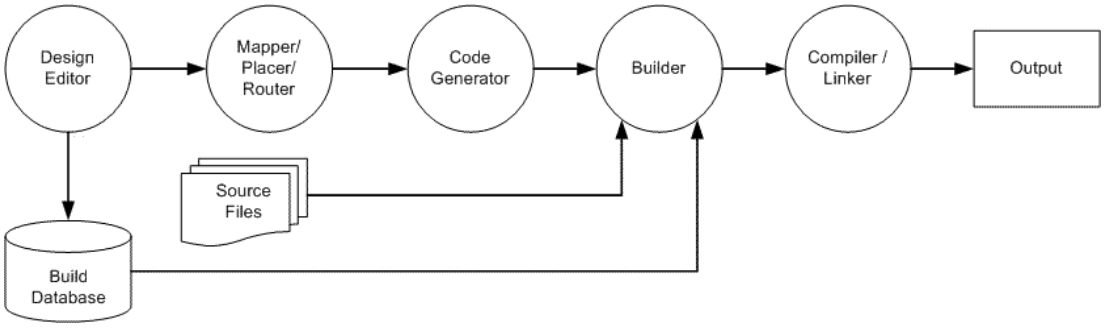
C/C++	Verilog	VHDL
asm	always	abs
auto	and	access
bool	assign	after
break	attribute	alias
case	begin	all
catch	buf	and
char	bufif0	architecture
class	bufif1	array
const	case	assert
const_cast	casex	attribute
continue	casez	begin
default	cmos	block
delete	deassign	body
do	default	buffer
double	defparam	bus
dynamic_cast	disable	case
else	edge	component
enum	else	configuration
explicit	endattribute	constant
export	endcase	disconnect
extern	endfunction	downto
false	endmodule	else
float	endprimitive	elsif
for	endspecify	end
friend	endtable	entity
goto	endtask	exit
if	event	file
inline	for	for
int	force	function
long	forever	generate
mutable	fork	generic
namespace	function	group
new	highz0	guarded
operator	highz1	if
private	if	impure

C/C++	Verilog	VHDL
protected	ifnone	in
public	initial	inertial
register	inout	inout
reinterpret_cast	input	is
restrict	integer	label
return	join	library
short	medium	linkage
signed	module	literal
sizeof	large	loop
static	macromodule	map
static_cast	nand	mod
struct	negedge	nand
input	nmos	new
template	nor	next
this	not	nor
throw	notif0	not
true	notif1	null
try	or	of
typedef	output	on
typeid	parameter	open
typename	pmos	or
union	posedge	others
unsigned	primitive	out
using	pull0	package
virtual	pull1	port
void	pulldown	postponed
volatile	pullup	procedure
while	rcmos	process
wchar_t	real	pure
_Bool	realtime	range
_Complex	reg	record
_Imaginary	release	register
	repeat	reject
	rnmos	rem
	rpmos	report
	rtran	return
	rtranif0	rol
	rtranif1	ror
	scalared	Select

C/C++	Verilog	VHDL
	signed	severity
	small	signal
	specify	shared
	specparam	sla
	strength	sll
	strong0	sra
	strong1	srl
	supply0	subtype
	supply1	then
	table	to
	task	transport
	time	type
	tran	unaffected
	tranif0	units
	tranif1	until
	tri	use
	tri0	variable
	tri1	wait
	triand	when
	trior	while
	trireg	with
	unsigned	xnor
	vectored	XOR
	WAIT	
	wand	
	weak0	
	weak1	
	while	
	wire	
	wor	
	xnor	
	XOR	

6 PSoC Creator 项目编译

在您编译项目时，PSoC Creator 会执行多个流程，以便生产输出文件。对于某个设计项目，输出是一个用于编程器件的 hex 文件。而对于库项目，输出则是一个用于指定组件信息的库文件。下面的原理图概述了设计项目的流程。库项目不一定要包括所有这些流程。



创建一个项目时，PSoC Creator 会设置用于编译项目创建输出文件的工具链。工具链是各个工具的组合（代码生成器、编译器、汇编器和连接器等），它能够将项目的内容转为项目类的相应的输出。

编译配置：

PSoC Creator 为这些工具链提供 Debug（调试）和 Release（释放）配置。编译配置过程中的更改对开发和测试设计非常有用。例如，初次进行开发代码时使用“Debug”配置是最简单的情况，因为它通常进行的优化较低，并会生成较多的调试信息。这样可以跟踪引起问题的原因。因为代码已经完善，并准备好进行释放，所以在使用额外的优化时会优先使用“Release”配置。这些优化有助于缩小程序，并能够提高它的运行速度。

如果发生了需要检查的问题，使用多个可用的配置能够在‘Debug’模式和‘Release’模式间快速切换。通过主工具栏上的编译配置，可以在不同的配置间进行更改。如果该选项是不可视的，请右键点击工具栏区域，然后选择编译配置。另外，通过使用工具链编译设置 对话框可以调整任何配置的编译选项。

该部分介绍的主题包括：

本部分介绍了 PSoC Creator 构建系统的各方面内容。它包括下面各主题：

- 编译工具栏指令
- 编译菜单
- 编译设置
- 映射器、放置器、路由器

- 控制文件
- 指令
- 生成的文件
- 源代码控制
- 静态时序分析
- CyPrjMgr 指令行工具
- Keil C51 编译器
- PSoC 3 中的可重入代码
- 注意列表错误信息

编译工具栏指令

编译工具栏包含了构建设计时需要使用的多条常用指令：

通过下拉菜单可以访问多个构建和清理选择，如下表所示。

编译工具栏包含以下指令：

菜单项	图标	快捷方式	说明	另外，请参考
Build (Named) Project（编译（名称）项目）		[Shift]+[F6]	编译选定的项目。	PSoC Creator 项目编译
Clean and Build (Named) Project（清理和编译（名称）项目）			清理和编译译选定的项目。 **注意**：在清理期过程中，PSoC Creator 仅会清理项目中各文件的编译输出。不属于项目的源文件（从项目中删除或移除）生成的输出文件不会被清理掉。	
Clean (Named) Project（清理（名称）项目）			清理已选定的项目。	
Build All Projects（编译所有项目）		[F6]	编译工作区中的所有项目。	
Clean and Build All Projects（清理和编译所有项目）			清理和编译工作区中的所有项目。	
清理所有项目			清理工作区中的所有项目。	
Cancel Build （取消编译）		[Ctrl]+[Break]	取消编译过程。	
Compile File（编译文件）		[Ctrl]+[F6]	编译选定的文件。	

菜单项	图标	快捷方式	说明	另外，请参考
Generate Application（生成应用）			生成 API 源代码文件。	Generated Files（生成的文件）
Program（程序）		[Ctrl]+[F5]	将选定项目生成的代码编写到选定的器件内。	
Debug（调试）		[F5]	启动调试器。	调试器的使用

另外，请参考：

- Build 菜单指令

Build 菜单

Build 菜单包括下面各条指令：

菜单项	图标	快捷方式	说明	另外，请参考
Build All Projects（编译所有项目）		[F6]	编译工作区内的所有项目。	PSoC Creator 项目编译
清理所有项目			清理工作区中的所有项目。 **注意**：在清理期过程中，PSoC Creator 仅会清理项目中各文件的编译输出。不属于项目的源文件（从项目中删除或移除）生成的输出文件不会被清理掉。	
Clean and Build All Projects（清理和编译所有项目）			清理和编译工作区中的所有项目。	
Build (Named) Project（编译（名称）项目）		[Shift] + [F6]	编译选定的项目。	
Clean (Named) Project（清理（名称）项目）			清理已选定的项目。	
Clean and Build (Named) Project（清理和编译（名称）项目）			重编译选定的项目。	
Cancel Build （取消编译）		[Ctrl] + [Break]	取消编译过程。	
Compile File（编译文件）		[Ctrl] + [F6]	编译选定的文件。	
Generate Application（生成应用）			生成 API 源代码文件。	Generated Files（生成的文件）
Generate Project Datasheet（生成项目的数据手册）			生成项目的数据手册。	生成项目的数据手册

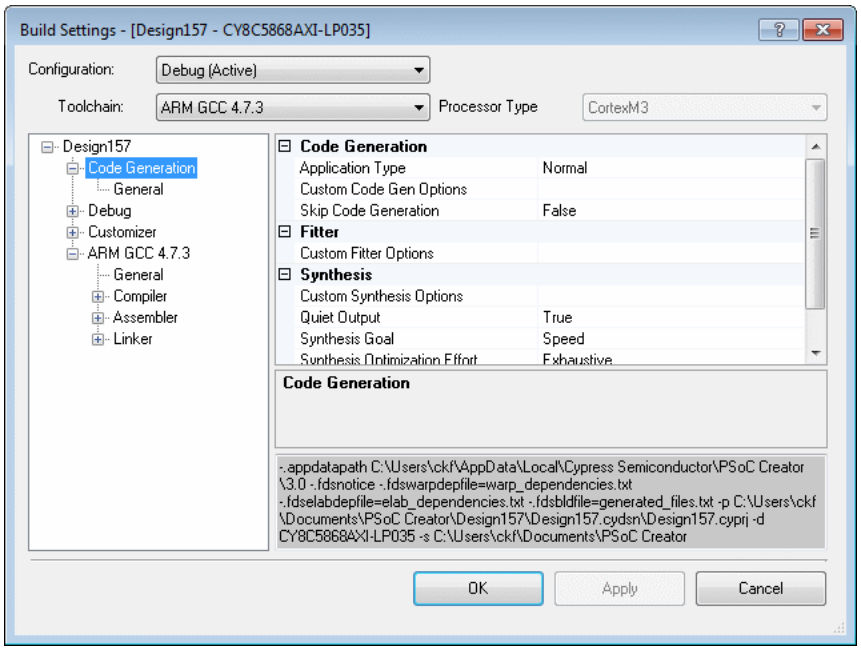
另外，请参考：

- 编译工具栏指令

编译设置

编译设置

您可以通过 Build Settings（编译设置）对话框在每个项目基础上定义各种编译设置。您还可以为选定的项目选择不同的工具链，并依次为编译器、汇编器和链接器设置不同的内容。



编译设置类别：

根据指定的 PSoC 器件和项目类型，该对话框会包括几个不同类别的设置项。这些类别包括：

- 代码生成（本主题）
- 调试
- 定制器
- 工具链
  - 编译器
  - 汇编器
  - 连接器
  - 文件库生成

要打开该对话框：

请右键单击您的项目，然后选择 **Build Settings**。

设置选项：

对话框的顶部包含下列各下拉菜单：

- **Configuration**（配置） — 通过使用该菜单可以为不同类型的编译设置编译设置首选项：Debug（调试）或者 Release（释放）。调试模式添加了日志信息功能（用于调试目的）；而释放模式则没有。

注意： 该选项不会改变项目的编译类型。它仅用于设置首选项。当前有效的编译类型显示在括号里。要修改有效的编译类型，请关闭 Build Settings 对话框，并使用主工具栏中的 **Configuration** 下拉菜单。

- **Toolchain**（工具链） — 使用该项选择选定项目的工具链。为给所有新项目指定默认的工具链，请参考选择默认的编译器。
- **Processor Type**（处理器类型） — 该字段显示设计项目的处理器类型。对于库项目，使用该字段进行选择处理器类型。

代码生成类别：

代码生成类别被默认显示。通过本节您能够指定多个代码生成选项。

代码生成：

- **应用类型** — 常规、标准 Bootloader、多应用 Bootloader 以及 Bootloadable。
- **自定义代码生成选项** — 自定义参数，以控制 API 的代码生成器。此时，用户看不到任何参数。该字段仅在赛普拉斯内部使用。
- 跳过代码生成 — 该选项下有两种情况，其状态为 true（使能）或 false（禁用）。

注意：跳过代码生成可有效地锁定设计，将来在修改原理图、设计范围资源时不会更改编译过程的输出。

Fitter（钳工）：

- **Custom Fitter Options**（自定义 Fitter 选项） — 指定自定义参数进行控制在 PSoC 器件中实现设计的方法。这些选项包括：

`-q` :	`-q`（“quiet”）选项禁止了在编译过程中状态信息的印刷。这样汇编和合成操作完成时会使屏幕更加简洁。
`-xor2` :	`-xor2` 选项会将设计中找到的所有 XOR 运算符传输到 Fitter，以当它认为合适时进行实现。
`-f(O|D|T` :	`-f` 选项使能专用的全局 Fitter 选项。`-f` 的后面（没有中间空格）必须是一个参数 `O`、`D` 或 `T`。该选项定义了 UDB 需要使用的触发器类型（d 型或 t 型）。`O` 代表优化，允许 warp 给器件选择最佳的触发器类型。
`-f(P|K)` :	`-fK` 选项会强制 Fitter 保留所有输出的用户指定极性。该选项与 `-fP` 选项相反，`-fP` 选项会优化最佳的极性。不推荐将 `-fK` 选项使用在大部分设计中，但在某些情况下，用户可以通过它确定所有信号的正确极性。
`-m` :	`-m` 选项使能项目 Verilog 文件的智能编译功能。通常，没有该选项时，Warp 会编译所有文件。指定该选项时，Warp 仅会对最新编译后被修改的文件进行编译。
`-w#` :	`-w` 选项指定了在 Warp 退出前作为单一 Warp 来显示的警报最多次数。
`-e#` :	`-e` 选项指定了在 Warp 退出前单一的 Warp 运行时发生非致命错误的最大数量。
`-yg(a|s|c)` :	`-yg` 选项会引起 Warp 合成设计，以便优化 UDB 组件的面积（`a`）、速度（`s`）或组合恒等式（`c`）。
`-yv#`	`-yv` 选项控制着在报告文件中报告的信息数量。`-yv` 选项后面是一个数字。默认情况下为 `0`。比 `0` 大的数字会生成更多

：　　有助于调试的详细报告文件。默认情况（数值 0）下，报告文件仅显示合成过程的主要事件。

-v#　　-v 选项中有一个数字参数用于控制虚拟替换算法的积极性。有效的数字范围为 0 到 11，其中数值 0 不执行任意虚拟替换，

：　　数值 11 执行虚拟替换，甚至对抗算法的更好判断，以隔离大的组合和在 UDB 中结合它们。

合成：

- **自定义合成选项 —** 指定自定义参数来控制 HDL 合成。请参考指令编辑器。

```
-fl -- no_factor
-o  -- opt_level
```

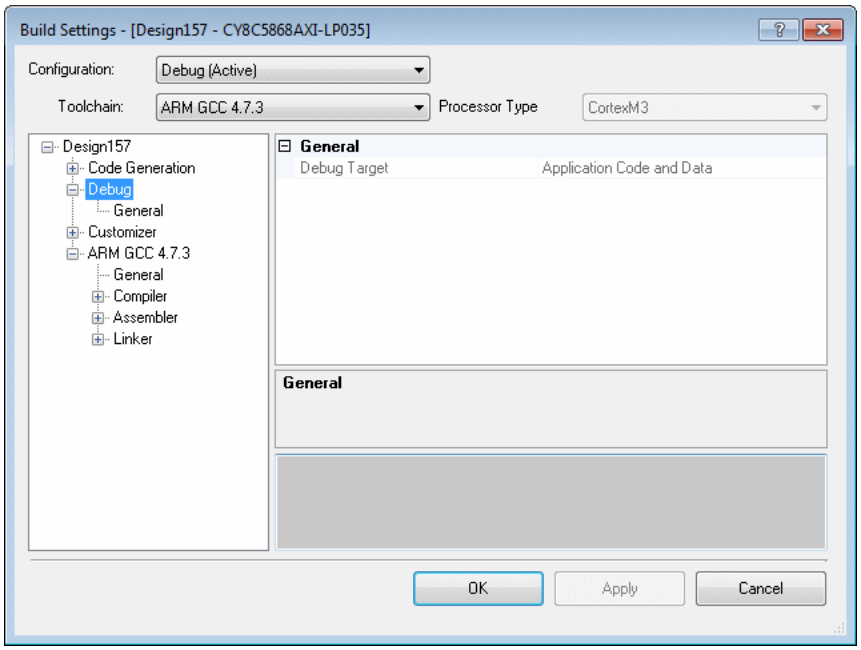
- **Quiet Output**（安静输出） — 控制合成工具的输出等级，其状态为 true（使能）或 false（禁用）。
- **Synthesis Goal**（合成目标） — 选择合成器的效率：速度或面积。
- **Synthesis Optimization Effort**（合成优化尝试） — 选择合成器投入到优化设计的程度：none（无）、normal（正常）或 exhaustive（全部）。
- **Virtual Node Substitution Level**（虚拟节点替换等级） — 用于优化设计的大小和速度：0-11。

另外，请参考：

- 工作区/项目
- 项目类型
- 指令编辑器

调试编译设置

使用“Build Settings”对话框的 Debug 部分能够选择影响调试经验的项目设置。



调试目标：

它控制着设计中被调试的项。在多数情况下，它仅支持应用代码和数据不会被修改。但如果项目类型为 Bootloadable，则会选择调试 Bootloader 还是 Bootloadable 的应用代码和数据。如果项目参照到一个多应用的 Bootloader，则会有第三个选择使用于应用代码和数据 2。

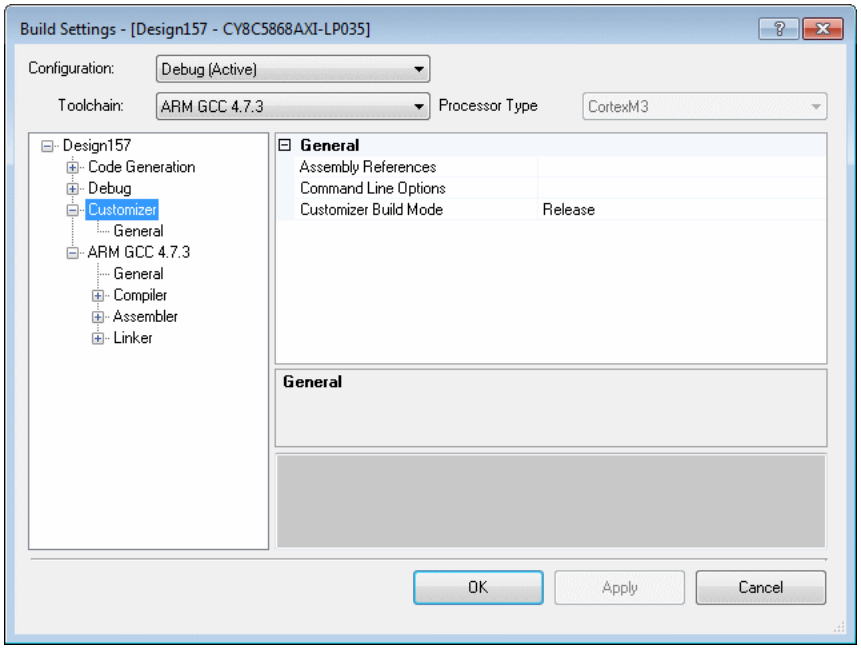
注意： 因为 Bootloader 执行了软件复位以启动可引导加载应用，所以无法从标准的 Execute Code（执行代码）菜单选项中调试 Bootloadable 应用。但仍建议您在将类型修正为 Bootloadable 前，调试应用。如果使您的应用成为可引导加载的程序，您仍然可以通过烧写设计到电路板上得到大多数调试功能。然后，按周期给器件复位/供电，并使用 Attach to Target（连接到目标）将调试器连接到 Bootloadable 应用。

另外，请参考：

- 编译设置
- 系统参考指南
- Attach to Target（连接到目标）

定制器的编译设置

“Build Settings”对话框中的 Customizer（定制器）部分用于控制多个选项，这些选项可确定如何为选定的项目构建基于源数据的定制器。



General（通用）：

- **Assembly References**（汇编参考） — 指定在为选定项目编译定制器代码的过程中需要增加的任何其他.NET 汇编程序。
- **Command Line Options**（指令行选项） — 指定在为选定项目编译定制器的过程中提供给编译器的任何其他指令行。默认的指令行选项只定义了“TRACE”常量，这些常量在调试过程中非常有用。

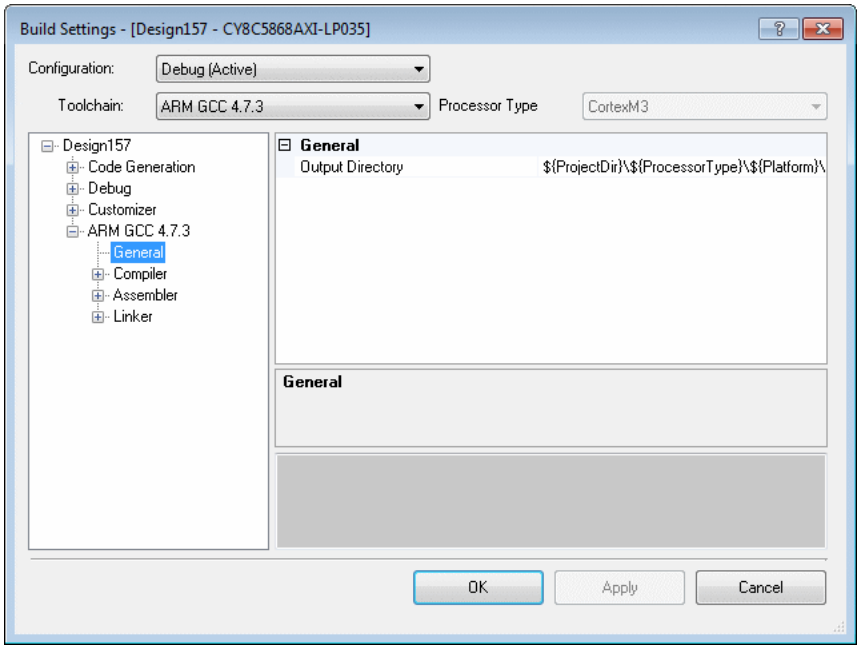
- **Customizer Build Mode**（定制器编译模式） — 指定该项目的定制器是在“Debug”（调试）模式下还是在“Release”（释放）模式下进行编译。在调试模式中，定制器会使用调试信息进行编译。调试定制器时应该选择该模式。

另外，请参考：

- 编译设置

工具链编译设置

Build Settings 对话框中的 Toolchain 部分提供了特定工具链的各项属性用于定义编译器、汇编器和链接器选项。这些设置只适用于选定的 **Configuration**、**Toolchain** 以及 **Processor Type**，并能够为每个选项设置不同的内容。



General（通用）类别：

在 General 下，这种类别的顶层仅包含一个选项：**Output Directory**。使用该选项您可以指定输出文件目录的相关路径。默认情况下为：

```
$(ProjectDir)\$(ProcessorType)\$(Platform)\$(Config)
```

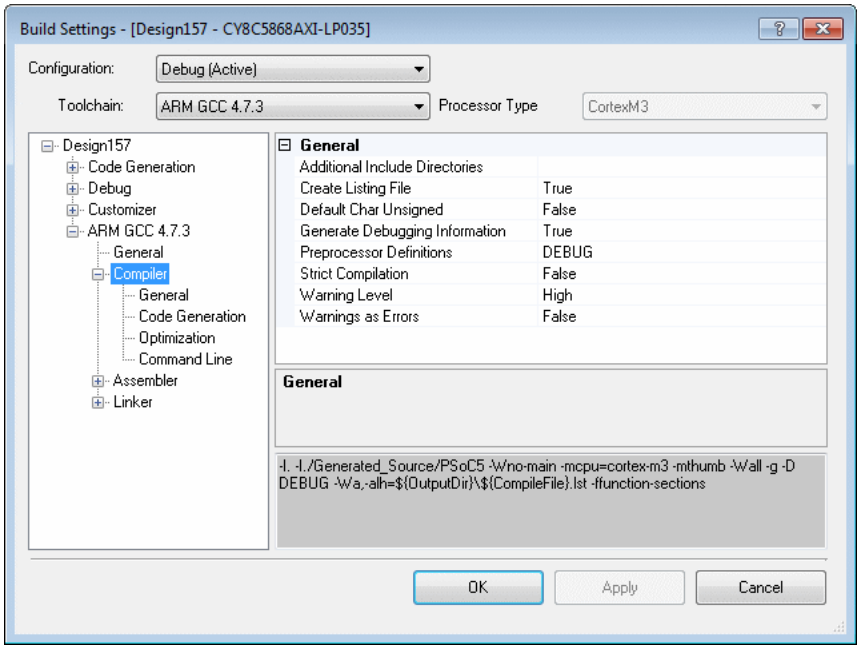
该路径定义了项目目录、处理器类型、平台和配置。

另外，请参考：

- 编译设置

编译器的编译设置

Build Settings 对话框中的 Compiler 部分用于控制根据 CPU 和编译器相异的多个选项。



Keil 选项：

Code Generation（代码生成）：

- **Inline Assembly**（内联汇编） — 用于使能内联汇编，其状态为 true（使能）或 false（禁用）。
- **Integer Promotion**（整型提升） — 用于使能整型提升，其状态为 true（使能）或 false（禁用）。

General（普通）

- **Additional Include Directories**（附加包含目录） — 指定额外的目录，这些额外目录用于编译器的 include 路径。如果您想指定多项，请用分号将其隔开。
- **Browse Information**（浏览器信息） — 包括生成的目标模块中的浏览器信息，其状态为 true（使能）或 false（禁用）。
- **Float Fuzzy** — 用于进行比较浮点数运算前指定被四舍五入的位数量，其范围为 0-7。
- **Generate Debugging Information**（生成调试信息） — 包括目标文件中的调试信息，其状态为 true（使能）或 false（禁用）。
- **Preprocessor Definitions**（预处理器定义） — 打开 Preprocessor Definitions 对话框，添加影响 C 源代码编译方法的指令定义。
- **Warning Level（警报等级）** — 确定所使用的警报等级：0、1、2。

列表文件

- **汇编代码表** — 添加一个汇编助记符表到列表文件内，其状态为 true（使能）或 false（禁用）

- **创建列表文件** — 创建文件包含高级源和汇编，其状态为 true（使能）或 false（禁用）。
- **列出 Include 文件** — 打印列表文件中的多个#include 文件，其状态为 true（使能）或 false（禁用）。

优化

- **链接器代码封装** — 包括关于链接器等级的程序优化的目标文件中的信息，其状态为 true（使能）或 false（禁用）。
- **优化重点** — 表示由编译器执行的优化侧重内容：none、size 或 speed。
- **Optimization Level**（优化等级） — 代码优化选项： 0-11。

Command Line（指令行）

- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

ARM 选项：

General（普通）

- **Additional Include Directories**（附加包含目录） — 指定额外的目录，这些额外目录用于编译器的 include 路径。如果您想指定多项，请用分号将其隔开。
- **Create Listing File**（创建列表文件） — 创建包含高级源和汇编的文件，其状态为 true（使能）或 false（禁用）。
- **Default Char Unsigned**（默认 Char 无符号） — 将 char 类型设置为无符号，其状态为 true（使能）或 false（禁用）。
- **Generate Debugging Information**（生成调试信息） — 生成使用 GDB 的调试信息，其状态为 true（使能）或 false（禁用）。
- **Preprocessor Definitions**（预处理器定义） — 打开 Preprocessor Definitions 对话框，添加影响 C 源代码编译方法的指令定义。
- **Strict Compilation**（严格的编译） — 使能严格的 ISO C/C++编译，其状态为 true（使能）或 false（禁用）。
- **Warning Level**（警报等级） — 确定所使用的警报等级：0、1、2。
- **Warnings as errors**（将警报作为错误） — 将所有警报报告为错误，其状态为 true（使能）或 false（禁用）。

Code Generation（代码生成）：

- **Struct Return Method** — 指定用于返回短结构/方法的方式：系统默认、寄存器或存储器。
- **Verbose Asm**（详细 Asm） — 能够在生成的汇编代码中加入额外的注释信息，使读取它更加容易，其状态为 true（使能）或 false（禁用）。

优化

- Create Function Sections（创建功能段） — 有两个可选情况，其状态为 true（使能）或 false（禁用）。

- **Inline Functions**（内联函数） — 编译时使能函数内联，其状态为 true（使能）或 false（禁用）。
- **Optimization Level**（优化等级） — 代码优化的选项：none、size、speed。

Command Line（指令行）

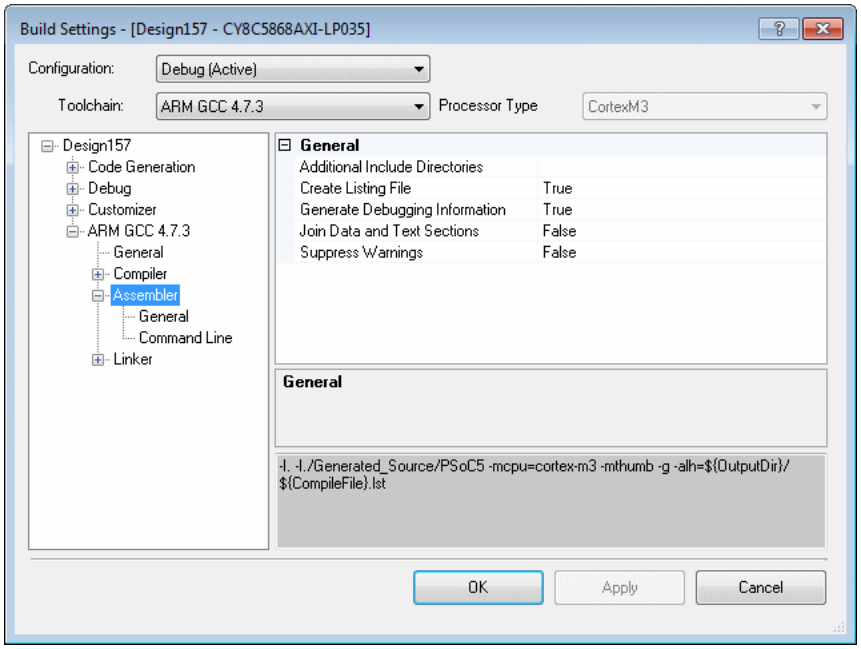
- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

另外，请参考：

- 编译设置

汇编器的编译设置

使用 Build Settings 对话框中的 Assembler 部分可以控制基于 CPU 和编译器的多个选项。



Keil 选项：

General（普通）

- **Additional Include Directories**（附加包含目录） — 指定额外的目录，这些额外目录用于编译器的 include 路径。如果您想指定多项，请用分号将其隔开。
- **Generate Debugging Information**（生成调试信息） — 生成使用 GDB 的调试信息，其状态为 true（使能）或 false（禁用）。
- **Macro Expansion**（宏扩展） — 使能宏扩展，其状态为 true（使能）或 false（禁用）。

- **Preprocessor Definitions**（预处理器定义） — 打开 Preprocessor Definitions 对话框，将定义指令添加到源代码内。

列表文件

- **Conditional** — 包括条件汇编程序源代码。该选项有两个选择，其状态为 true（使能）或 false（禁用）。
- **Create Listing File**（创建列表文件） — 创建文件包含高级源和汇编，其状态为 true（使能）或 false（禁用）。
- **Macros**（宏） — 包括列表文件中的所有宏扩展，其状态为 true（使能）或 false（禁用）。

Command Line（指令行）

- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

ARM 选项：

General（普通）

- **Additional Include Directories**（附加包含目录） — 指定额外的目录，这些额外目录用于编译器的 include 路径。如果您想指定多项，请用分号将其隔开。
- **Create Listing File**（创建列表文件） — 创建包含高级源和汇编的文件，其状态为 true（使能）或 false（禁用）。
- **Difference Tables**（差异表） — 当汇编器更改由指令生成的代码时，允许发布警报，其状态为 true（使能）或 false（禁用）。
- **Generate Debugging Information**（生成调试信息） — 生成使用 GDB 的调试信息，其状态为 true（使能）或 false（禁用）。
- **Join Data and Text Sections** — 使能该选项可生成短地址位移，其状态为 true（使能）或 false（禁用）。
- **Suppress Warnings（抑制警报）** — 使能该选项可抑制所有警报，其状态为 true（使能）或 false（禁用）。

Command Line（指令行）

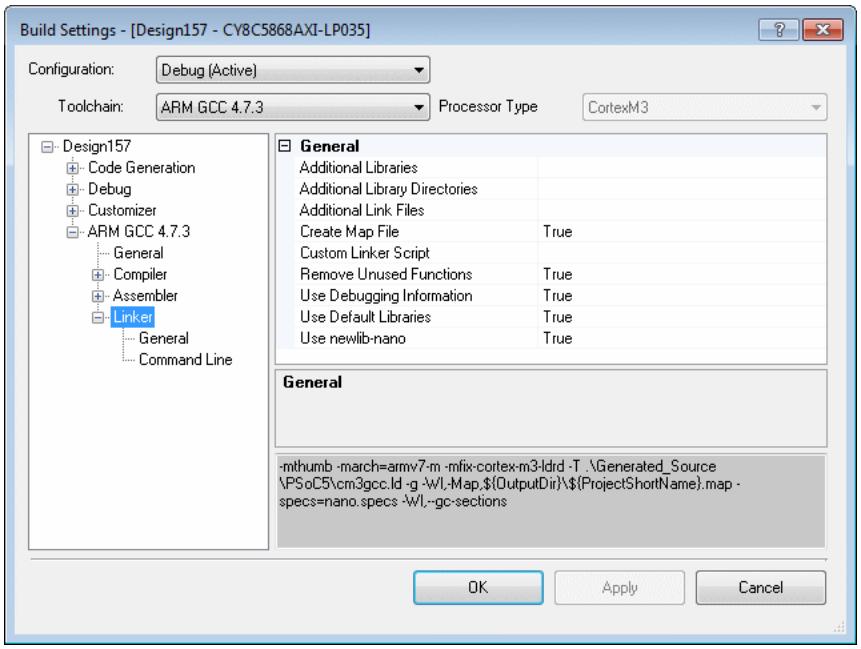
- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

另外，请参考：

- 编译设置

链接器的编译设置

使用 Build Settings 对话框中的 Linker 部分可控制基于 CPU 和编译器的多个选项。



Keil 选项：

General（普通）

- **Additional Link Files**（其他链接文件）— 指定将附加文件链接到可执行文件。如果存在多个目录，请使用分号将其隔开。
- **Create Code Listing**（创建代码列表） — 创建包含编程源/汇编的代码列表文件，其状态为 true（使能）或 false（禁用）。
- **Disable Unreferenced Segments Warnings**（禁用未参考的代码段警报） — 禁用关于未参考代码段的链接器警报，其状态为 true（使能）或 false（禁用）。
- **Recursions**（递归） — 控制在发生某个中止前，连接器中允许递归的数量，并要求为整数。
- 移除未使用的代码段 — 有两个选项，其状态为 true（使能）或 false（禁用）
- **Warning Level**（警报等级） — 确定所使用的警报等级：0-2。

列表文件

- **Create Map File**（创建映射文件）— 用于生成来自重新定位地址的更新文件以及来自链接器的数据，其状态为 true（使能）或 false（禁用）。
- **Cross Reference Report**（交叉参考报告） — 包括列表文件中的交叉参考报告，其状态为 true（使能）或 false（禁用）。
- **Generate Memory Map**（生成存储器映射） — 生成列表文件中的存储器映射和覆盖层映射，其状态为 true（使能）或 false（禁用）。

调试

- **Generate Debug Lines**（生成调试行） — 包括链接器输出和映射文件中的行号信息，其状态为 true（使能）或 false（禁用）。如果该项被设置为“false”，则源代码等级调试不可用。
- **Generate Local Symbols**（生成局部符号） — 包括链接器输出和映射文件中的局部符号信息，其状态为 true（使能）或 false（禁用）。如果该项被设置为“false”，则源代码等级调试不可用。
- **Generate Public Symbols**（生成公共符号） — 包括链接器输出和映射文件中公共符号的信息，其状态为 true（使能）或 false（禁用）。如果该项被设置为“false”，则源代码等级调试不可用。
- **Symbol Types**（符号类型） — 包含输出文件中的符号类型，其状态为 true（使能）或 false（禁用）。

Command Line（指令行）

- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

ARM 选项：

General（普通）

- **Additional Libraries**（其他库）— 用于指定链接到可执行文件的其他库。如果存在多个库，请使用分号将其隔开。
- **Additional Librarie Directories**（其他库目录）— 用于指定添加到链接器的库路径的其他库。如果存在多个目录，请使用分号将其隔开。
- **Additional Link Files**（其他链接文件）— 指定将附加文件链接到可执行文件。如果存在多个文件，请使用分号将其隔开。
- **Create Map File**（创建映射文件）— 用于生成来自重新定位地址的更新文件以及来自链接器的数据，其状态为 true（使能）或 false（禁止）。
- **Custom Linker Script**（自定义链接器脚本）— 用于指定构建项目时使用的自定义链接器脚本的路径，而不使用 cy_boot 组件提供的默认脚本。
- Remove Unused Functions（移除未使用的函数）— 其状态为 true（使能）或 false（禁止）
- **Use Debugging Information**（使用调试信息）— 允许使用由 gcc 在编译过程中生成的源代码调制信息，其状态为 true（使能）或 false（禁止）。
- Use Default Libraries（使用默认的库）— 其状态为 true（使能）或 false（禁止）

Command Line（指令行）

- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

printf、sprintf 和浮点值

默认情况下，与 ARM GCC 工具链结合使用的 newlib 纳米 C 库不包含同%f 格式字符串相同浮点的支持。这样可以节省存储器的空间。如果需要使用浮点值，可以要求 newlib 纳米库通过将自定义指令行参数添加到连接器中来支持格式字符串。在 **Linker > Command Line** 字段中，请输入下面的内容：

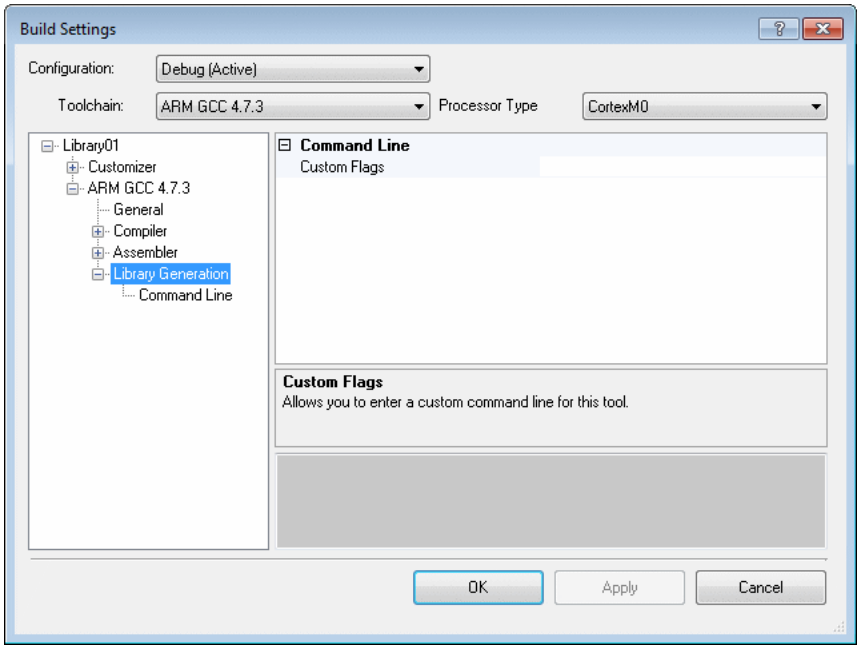
```
-u _printf_float
```

另外，请参考：

- 编译设置

文件库生成的编译设置

Build Settings（编译设置）对话框的文件库生成部分只适用于库项目。



这时，它只有一个属性：

- **Custom Flags**（自定义标志）— 您能够输入编译器可识别的任何标志。请参考有关编译器的相应文档。根据其他部分的选项，将这些标志添加到由 PSoC Creator 生成的标志上。

另外，请参考：

- 编译设置

映射器、放置器、路由器

使用 **Build** 功能时，PSoC Creator 将设计的原理图和 Verilog 规范转换为配置信息。此信息可用于对器件的数字和模拟组件进行的编程。转换过程的第一个阶段被称为映射程序。一旦映射过程完成，就进入放置和路由阶段。

映射阶段是指 PSoC Creator 将设计中所描述的逻辑映射到代表 PSoC 器件的功能模块组件中的过程。放置阶段会将功能模块放置在所选器件中可用的位置上。根据设计中的连接以及您所指定的位置限制，放置器会尝试优化各不同组件的放置。确定好所有组件的位置后，路由阶段将通过切换器件的结构计算各组件间最佳的信号走线。如果资源不足以支持设计的必要组件，或者如果 PSoC Creator 确定此走线不可用，则会生成错误，且构建操作失败。

注意：由于器件的各区域之间的关系，PSoC Creator 会尝试先对所有模拟组件进行放置和布线。完成后，它会再次尝试对设计的数字组件进行相同的操作。

一旦设计被映射、放置、布线完毕，将生成用于创建此配置的设置，并将其写到构建整个项目应用时使用的生成文件文件夹中。通过指定设计中放置组件的位置，您可以改变放置和布线阶段的结果。更多有关信息，请参考设计范围资源中引脚编辑器和指令编辑器部分，以及指令一节中介绍的内容。

另外，请参考：

- Design-Wide Resources（设计范围资源）
- 引脚编辑器
- 指令编辑器
- Directives（指令）
- 生成的文件

控制文件

控制文件

控制文件是一个可选文件，它为给定的设计提供了一个用于设置全局指令的通用位置。当维持某个器件和独立于供应商的硬件描述语言（HDL）源文件时，控制文件会提供组合的多方面详细控制。使用控制文件，用户可以通过属性机制添加各条指令，该文件为这些（扩展的）属性提供了 Warp 的具体 HDL 语法，以便能够剪切和粘贴 HDL 源文件和控制文件间的这些指令。拟合、放置和布线过程结束后，该文件还能用于自动将引脚分布和放置信息反向注解到设计（或 HDL 文件）中。

在合成、优化和因式分解过程中，Warp 获得很多新的信号和节点名称，以进行设计。例如，Warp 将总线分隔为各个单独的信号。即使各种对象（如总线）使 HDL 设计条目变得更加简单，但也不会存在任何部分能够基于 VHDL 语言的方式将各属性分配给总线。在其他情况下，Warp 会生成全新的信号名称，这些信号在设计中与任意一个对象都没有任何直接的关系。在进行因式分解过程中（其中各因子是通过全局检查设计生成的），会发生这种情况。

在每个设计中只有一个控制文件，并且该文件的基本名称必须与顶层设计文件名称相似。例如，如果顶层设计文件的名称为 *mydesign.vhd* 或 *mydesign.v*，那么控制文件必须为 *mydesign.ctl*。

控制文件的构建和编辑是一个重复过程，通常进行这些操作可以调整、改进或限制合成的结果。

在语法分析、合成和拟合过程中采用了控制文件。语法分析过程中创建的名称不合格。因此，如果控制文件模式与名称相互匹配，才会使用属性。这些发生在设计层次的所有等级中。

例如，考虑被称为'c'的顶层信号。会将 'c' 采用的属性应用于顶层信号，也可以将它应用于名称为 'c' 的任何内部信号，其中包括 Warp 库中的信号。因此，如果控制文件显示为：

```
attribute placement_force of c : signal is "U(1,2)A";
```

那么它会尝试将所有名称为 'c' 的信号强制放置在可编程逻辑模块中。

可能有过多的信号，此时放置器会给出提示。有关工作是将顶层信号重新命名为某个独特的名称（如 top_level_c），或使用原文件中的属性（而不使用控制文件中的属性）。

要想创建控制文件，请执行下列操作：

1. 点击 Workplace Explorer（工作区浏览器）中的 **Components** 选项卡。
2. 右键单击有效项目的 **TopDesign** 组件，并选择 **Add Component Item...**项。

   出现‘Add Component Item’对话框。
3. 向下滑动到 **Misc** 组，然后选择 **Control File**，并点击 **OK**。

   一个 *TopDesign.ctl* 文件将被创建并添加到‘Workspace Explorer’窗口内。
4. 双击它以打开文件。

另外，请参考：

- 指令
- Attribute（属性）、CSAttribute（CS 属性）和 FixedAttribute （固定属性）
- 控制文件的格式
- 控制文件模式的匹配

Attribute（属性）、CSAttribute（CS 属性）和 FixedAttribute （固定属性）

Attribute（属性）：

将“Attribute”关键字应用到大部分属性的应用。

对于并非以反斜线符号开始的模式，属性关键字识别所有模式时不分大小写。对于以反斜线符号开始的模式，识别时它会区分大小写（类似于 CSAttribute）。在 VHDL 设计的文本中同样允许属性关键字，它遵循如 VHDL 语言中所定义的语法和语义。但是，除了采用通配符模式外，属性的含义仍保持不变。

CSAttribute（CS 属性）：

在罕见的情况下，当多个对象具有相同的标识符，但只有大小写不同时（比如：Fred_Astaire 和 fred_astaire），则可以使用 CSAttribute（区分大小写属性）来选择与已给模式名称相同的对象。

CSAttribute 关键字以区分大小写的方式识别所有模式。

FixedAttribute（固定属性）：

FixedAttribute 是由工具使用的。FixedAttribute 行上的模式实际上是一个区分大小写的名称，其中任何元字符都被视为标识符中的实际字符。

FixedAttribute 关键字接受标识符而不接受模式，并与大小写区分一致。

有效性检查：

对于控制文件中的 CSAttribute 和 Attribute 指令，必须为基于 Verilog 和 VHDL 的设计来验证属性名称的有效性和数值的合格性。对于字符串值属性，实际上是由属性的预定客户端进行检查的（比如：只有相应的钳工才能检查 placement_force 值）。下面示例显示的是 Attribute、CSAttribute 和 FixedAttribute 如何与 Verilog 和 VHDL 源代码相匹配。

模式	源代码标识符	与VDHL源代码匹配	与Verilog源代码匹配
`attribute ff_type of mysig : signal is ff_d`	mysig Mysig MYSIG mySig	有 有 有 有	有 有 有 有
`csattribute ff_type of mysig : signal is ff_d`	mysig Mysig MYSIG mySig	有 有 有 有	有 不支持 无 无
`fixedattribute ff_type of mysig : signal is ff_d`	mysig Mysig MYSIG mySig	有 有 有 有	有 不支持 无 无

另外，请参考：

- 控制文件

控制文件的格式

控制文件的格式如下：

- 一个注释以“--”开始，并结束在行尾。
- 控制文件中的行包含 attribute、csattribute 或 fixedattribute 语句。
- 行必须以下面某个关键字开始：attribute、csattribute 或 fixedattribute。

控制文件中 attribute、csattribute 和 fixedattribute 语句的语法如下所示：

```
attribute attribute_name [of] pattern [:] [object-class] [is] value ;
csattribute attribute_name [of] pattern [:] [object-class] [is] value ;
fixedattribute attribute_name [of] identifier [:] [object-class] [is] value ;
```

除模式和标识符外，上述各行中的所有字都按照区分大小写方式处理。模式则根据控制文件模式的匹配部分所解释的情况。默认的对象级是 SIGNAL（信号）。但是，对象级可以是：

- 标签
- 实体
- 模块
- 架构
- 信号

在 VHDL 设计中，attribute 语句可以出现在控制文件和源文件中。控制文件的属性优先于源文件的属性。如果源文件的特定属性与控制文件中的属性行相匹配，则优先控制文件中的属性。

具体应用属性优先于层次应用属性。

Attribute_name（属性名称）：

Attribute_name 是指令名称。

模式/标识符：

能够将可放入指令的对象名称指定为一个标识符（简单、扩展/转义），或将其作为一个模式。某个阵列的单独位由括号内的整数表示。

在控制文件中，Warp 包含 VHDL 语言扩展标识符和 Verilog 有限版本的转义标识符。扩展标识符或转义标识符通常以反斜线开始。该标识符常以第一个非转义的反斜线（若有）结束，也可以以第一个遇到的空格（若没有尾部的反斜线终止符）结束。通过在 VHDL 扩展标识符中的反斜线的前面放置另一个反斜线，就可退出该标识符。

对于 Verilog 转义标识符，控制文件中不允许任何嵌入式反斜线。例如，即使‘\foo\bar’在 Verilog 中是一个有效的转义标识符，但在控制文件中仍无效。添加转义标识符的有限版本是为了便于用户使用，并且推荐使用了 VHDL 类型的扩展标识符。

注意：需要将带有[]的名称替换为通配符。例如：

```
attribute placement_force of \Sync:genblk1[0]:INST\ : label is "U(0,0)2"
```

被代替为：

```
attribute placement_force of \Sync:genblk1?0?:INST\ : label is "U(0,0)2"
```

可选关键词：

“of”和“is”均为可选的关键词，并被忽略。

对象类别：

对象类别是指 HDL 对象的类型。如果未指定对象类别，将假定一个信号。对于 VHDL，有效的类别包括实体、架构、组件和标签。可将标签类别指定用于组件实例化的指令（directive）。对于 Verilog，有效的类别包括模块、标签和信号。

指令（directive）终止：

指令的终止可由新的一行、分号或一条注释来表示。

数值：

是指指令的数值。

控制文件模式匹配

控制文件中属性规范的对象名称部分不仅仅是一个名称。实际上它还代表了一种格式。只要合理（相应类型）的对象名称的格式，该对象就能使用该属性。格式由普通的字符组成。根据属性类型，这些字符必须完全符合区分大小写或不区分大小写的形式。下表列出了附加的通配符结构：

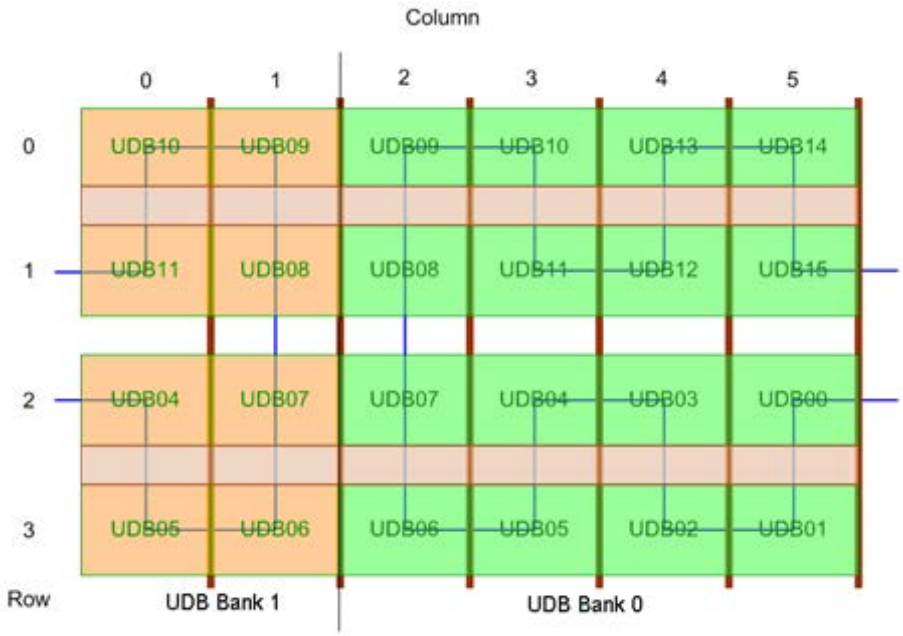
字符/结构	定义
*	表示匹配于零或更多个字符
?	表示匹配于一个字符
[c-c]	表示 ASCII 字符范围中某个必须匹配的字符。
[0-9]	表示 ASCII 字符范围中某个必须匹配的字符。
[ccc]	表示需要匹配的字符列表。

当构建格式时，区分各个字符的大小写。匹配情况还取决于属性指令类型（Attribute、 CSAttribute）。当格式匹配字符和通配符结构均位于[ ]内时，它们被视为字符文字。这样，该格式要完全得到匹配。

PSoC Creator 中的 PSoC UDB

PSoC Creator 从报告文件（请参考 生成文件）和指令编辑器中 UDB 网格内的位置（行、列格式[UDB=(0,2)]）引用通用数字模块（UDB）。但器件的技术参考手册（TRM)却以其他方式引用 UDB（组号、组中的 Udb，例如，Bank 0, Udb 11）。

PSoC Creator 报告文件中 UDB 元件的命名规范参考与 TRM 中的不一样。 所用的符号是 U（行，列），其中行和列均为 UDB 阵列左上角基于网格的 0,0。而基于网格到基于 TRM 的命名规范的映射情况如下所示:



Directives（指令）

可使用指令对设计的实现产生影响。能以迭代方式使用它们，以完善，改进或限制合成结果。可将指令应用于已在原理图中进行实例化的组件，也可以将其用于由合成器通过 Verilog HDL 代码推断的组件。

指令格式汇总：

下表总结了 PSoC Creator 中所使用的指令。

指令	目标	格式	位置值
放置指令			
placement_force	任意的逻辑	attribute placement_force of signal_name : signal is "string";	U(1,2)A 或 U(1,2,A)3
placement_force	固定功能	attribute placement_force of component_name : label is "string";	F(Timer,3) 或 F(DFB,0)
placement_force	UDB 组件	attribute placement_force of component_name : label is "string";	U(3,2)
port_location	IO 端口	attribute port_location of bit_name : label is "string";	PORT(2,3)
placement_group	任意的逻辑	attribute placement_group of signal-name : signal is "string" ;	group_1
synchronization_needed	IO 端口	attribute synchronization_needed of signal_name : signal is "string";	AUTO 或 SYNC 或 NOSYNC
合成指令			
no_factor	任意的逻辑	attribute no_factor of signal_name : signal is value;	
opt_level	任意的逻辑	attribute opt_level of signal_name : signal is integer;	2 或 1 或 0
synthesis_off	任意逻辑	attribute synthesis_off of signal_name : signal is value;	false 或 true

placement_force:

placement_force 指令有助于将信号和相关组件锁存到滤波器中特定的位置。一般情况下，滤波器将组件放置在器件上最好的位置，但您可能需要限制滤波器，在特定的位置上放置组件。

```
attribute placement_force of signal_name : signal is "string";
attribute placement_force of component_name : label is "string";
```

字符串是可编程逻辑模块、宏单元、UDB 元件或固定功能模块的位置描述符。根据寻址对象，每一种均有自己的格式和含义。

- 对于 UDB 的某个元件（Datapath、状态寄存器，等等），可以根据具体的 UDB 确定相应的字符串。 该字符串的格式为 `U(x,y)`，其中 `x,y` 分别是 UDB 的行和列。为识别属性，必须使用带标签关键词的属性版本将组件分配给组件，这样才能使属性生效。 例如：

  ```
  attribute placement_force of \test_control:ctrl_reg : label is "U(3,5)";
  ```

  注意： 当指定某个datapath链的位置时，必须将属性分配给该链中的第一个datapath，这样才能使该属性生效。

- 对于某组随机逻辑，该字符串会受特定 UDB 中具体 PLD 的限制。该字符串的格式为 `U(x,y)l`，其中 `x,y` 分别为 UDB 的行和列，`l` 为 PLD 描述符（使用 A 或 B 表示 UDB 中两个 PLD 的其中一个）。 这样，滤波器仍可以选择最适合设计布局的宏单元列。为识别属性，需要将其分配给来自宏单元的输出信号，这样才能使该属性生效。 例如：

  ```
  attribute placement_force of out_1 : signal is "U(2,4)A";
  ```

- 另外，在特定 UDB 中的具体 PLD 内，可以使一组随机逻辑受特殊宏单元列的限制。该字符串的格式为 `U(x,y,l)i`，其中 `x,y` 分别为 UDB 的行和列，`l` 是 PLD 描述符(使用 A 或 B 表示 UDB 中两个 PLD 的其中一个)，`i` 是 PLD 中的宏单元索引。为识别属性，需要将其分配给来自随机逻辑组的输出信号，这样才能使该属性生效。 例如：

  ```
  attribute placement_force of out_2 : signal is "U(1,3,A)2";
  ```

- 通过使用标签关键词既名称（而不是信号）参考固定功能组件，可将属性直接适用于这些组件。 标签的格式为 F(block_type,index)，其中 block_type 是模块的关键词描述符，index 是所使用的模块实例。 有效的关键词组为 CAN、Comparator、 DFB、DSM、Decimator、EMIF、I2C、SC、Timer、USB、VIDAC、Abuf、Lpf：

  ```
  attribute placement_force of \Timer_1:TimerHW\ : label is "F(Timer,1)";
  ```

port_location:

port_location 指令根据引脚在器件（而不是在封装）中的位置将设计的外部信号映射到目标器件的引脚上。 该指令的语法为：

```
attribute port_location of bit_name : label is "string";
```

用于指定位置的字符串格式为 PORT(port_index, pin_index)，其中 port_index 表示器件中端口的编码，pin_index 表示端口的引脚。

该指令示例为：

```
attribute port_location of Terminal_1 : label is "PORT(2,3)";
```

请注意，当使用该指令纠正逻辑端口的位置时，必须将纠正内容应用在端口的第一个引脚上。

placement_group:

可使用 placement_group 指令将某组信号（与任意逻辑相关联，但与固定功能组件无关）组合在一起，并要求放置和布线工具将这些信号放在一起。放置和布线工具尝试以最小的 PLD 数量将属于相同组标识符的所有信号放在一起。 该指令的格式为：

```
attribute placement_group of signal-name : signal is "string" ;
```

该字符串可以是用户定义的任意字符串，并用于唯一标识某组信号（请参见下面示例）。组标识符不分大小写。 下面的示例尝试将逻辑驱动 signal2 和 signal3 放在一起：

```
attribute placement_group of signal2: signal is "group1" ;
attribute placement_group of signal3: signal is "group1" ;
```

synchronization_needed:

synchronization_needed 指令可用于修改器件中 IO 的默认同步化性能。默认情况下，对 IO 输入进行同步，但不对 IO 输出进行该操作。

```
attribute synchronization_needed of signal-name : signal is "string" ;
attribute synchronization_needed of logical-port-name(pin-index) : label is "string" ;
```

synchronization_needed 属性可以使用某个字符串的值。该字符串的值可以是 AUTO、SYNC 或 NOSYNC 中的某一个。 示例：

```
attribute synchronization_needed of Terminal_1 : signal is "SYNC" ;
attribute synchronization_needed of dport_1(2) : label is "NOSYNC" ;
```

no_factor:

使用 no_factor 指令可以阻止 Warp 合成引擎中的逻辑系数分解，从而可以阻止分开所述节点。

```
attribute no_factor of signal_name : signal is value;
```

在优化阶段，Warp 合成引擎会对具有相同驱动程序（公式）的信号定义一个别名。使用该指令时，会使方程式忽略这两个操作。如果因设计的限制而重复某些相同的逻辑，或者逻辑系数分解算法过于积极，那么该性能便会非常有用。

示例：

该示例可阻止为 my_signal 信号定义别名或阻止对其进行系数分解。

```
attribute no_factor of my_signal : signal is true;
```

在 Verilog 设计中，可将属性放置在某个模块中的所有信号上，具体如下：

```
attribute no_factor of my_module : module is true;
```

该示例可阻止为 my_module 的所有信号定义别名或阻止对其进行系数分解。

opt_level:

opt_level 指令向 Warp 声明优化某些信号时需要进行的尝试。

```
attribute opt_level of signal_name : signal is integer;
```

整数表示需要进行的尝试。当前有三个尝试等级（0、1 和 2）。0 等级 opt_level 指示 Warp 关闭所述信号的所有优化操作。PLD/CPLD 滤波器可以使用该指令进行同样的操作。1 等级 opt_level 指示 Warp 对各个公式进行简单快速的优化。2 等级 opt_level 指示 Warp 执行最高的可用优化等级。推荐在所有设计中执行 2 等级 opt_level。

示例：

该指令会禁用 my_signal 信号上的所有优化操作。

```
attribute opt_level of my_signal : signal is 0;
```

synthesis_off:

synthesis_off 指令控制服务各信号的表达式的扁平化和系数分解。对于这些信号，该指令被设置为 true。该指令会使某个信号成为逻辑公式中的一个系数分解点。这样，在优化过程中，不会代替该信号。

```
attribute synthesis_off of signal_name : signal is value;
```

synthesis_off 指令只适用于信号。对于某个给定的信号，synthesis_off 指令的默认设置为‘false’。通过该指令，用户可以控制分解成节点（即将它们分配给某个物理路由路径）的公式或子表达式。

- 当针对已给信号将该指令被设置为‘true’时，synthesis_off 将使该信号变成一个节点（即逻辑公式的系数分解点），以供目标技术使用。这样，在优化过程中，可以避免替换掉该信号。在下面的情况中，该指令将非常有用：当执行取代操作严重延长优化阶段时长（由于要求成倍使用 CPU 和存储器）或使用资源过多时。
- 等式放置节点，会强制信号在阵列中进行额外的传输，从而会降低性能，但会使设计变得更好。
- synthesis_off 指令只适用于组合等式。注册的公式本身便是系数分解点；因此，在这类公式中使用 synthesis_off 会导致多余的系数分解。

示例：

在下面示例中，sig1 信号的 synthesis_off 指令被设置为 true。

```
attribute synthesis_off of sig1:signal is true;
```

另外，请参考：

- 指令编辑器
- 控制文件

生成的文件

构建成功后，PSoC Creator 将生成各种文件，这些文件会成为您设计的一部分。这些文件位于 **Source** 选项卡下的工作区浏览器录内。它们专用于器件架构（PSoC 3、PSoC 5LP 或 PSoC 4/PRoC BLE）和选定的编译器。下面列出并说明了构建后所生成的各个文件。

（各）文件	说明
cy_boot（另见系统参考指南）。	
CyBootAsmKeil.a51	为对时间要求比较严格的子程序提供汇编时间（该 Keil 文件与 CyBootAsmGnu.s 和 CyBootAsmRv.s 等效）。

CyBootAsmIar.s	为对时间要求比较严格的子程序提供汇编实现（该 IAR 文件与 CyBootAsmGnu.s、CyBootAsmKeil.a51 和 CyBootAsmRv.s 等效）。
CyDmac.c /.h	使用 DMA 控制器时所需要的软件 API。
CyFlash.c /.h	用于写入闪存的软件 API。
CyLib.c /.h	软件 API 用于进行电源管理、字符串/字符子程序、存储器操作以及启用/禁用 PSoC 器件的选定部分。
cypins.h	包含用于端口/引脚访问和控制的函数原型和常数。
cyPm.c/.h	提供电源管理 API 的函数定义。
CySpc.c /.h	用于写入到系统性能控制器的软件 API。
cytypes.h	提供宏和定义，以使代码作为写入工具链，处理器对此并不清楚。
cyutils.c	实现用于提供工具链/处理器无关函数的低级工具函数。显示在 cytypes.h 文件中。
cymem.a51	Keil 启动（boot-up）的专用存储器子程序。
KeilStart.a51	使用 Keil 工具的 PSoC 3 芯片的启动（bootup）代码。
Cm3Start.c/Cm0Start.c	ARM CM3/CM0 的启动（startup）代码。
PSoC3_8051.h /.inc	PSoC 3 架构的 8051 寄存器定义。
cm3gcc.ld/cm0gcc.ld	GCC 工具链的链接器脚本
Cm3RealView.scat/Cm0RealView.scat	RealView & MDK 工具链的分散文件
core_cm0.h 或 core_cm3.h core_cmFunc.h 和 core_cmInstr.h	Cortex-M 处理器系列的 CMSIS 标准库：用于 PSoC 4 的 core_cm0.h/用于 PSoC 5LP 的 core_cm3.h。 PSoC 4/PRoC BLE 和 PSoC 5LP 均包含这两个文件。
core_cm0_psoc4.h 或 core_cm3_psoc5.h	CMSIS 库的 PSoC 4/PRoC BLE 或 PSoC 5LP 特定中断信息。
CyBootAsmGnu.s	为对时间要求比较严格的子程序提供汇编实现（该 GNU 文件与 CyBootAsmKeil.a51 和 CyBootAsmRv.s 等效）。
CyBootAsmRv.s	为对时间要求比较严格的子程序提供汇编实现（该 RealView 文件与 CyBootAsmKeil.a51 和 CyBootAsmGnu.s 等效）。
General（普通）	
cydevice_trm.h	定义器件中配置空间的所有地址。这些地址并不包含与您设计中各实例相关的含义信息。不应直接使用这些地址。
cydevice.h	cydevice_trm.h 的弃用版本。
cydevicekeil_trm.inc（PSoC 3）	针对 Keil 汇编器（AX51），对器件中配置空间的所有地址进行定义。这些地址并不包含与您设计中各实例相关的含义信息。不应直接使用这些地址。
cydevicegnu_trm.inc （ PSoC 5LP / PSoC 4 / PRoC BLE GCC）	针对 GNU 汇编器（gas）定义器件中配置空间的所有地址。这些地址并不包含与您设计中各实例相关的含义信息。不应直接使用这些地址。
cydevicerv_trm.inc （ PSoC 5LP / PSoC 4 / PRoC BLE Real View）	针对 Real View 汇编器定义器件中配置空间的所有地址。这些地址并不包含与您设计中各实例相关的含义信息。不应直接使用这些地址。
cydevicekeil.inc（PSoC 3） cydevicegnu.inc（PSoC 5LP / PSoC 4 / PRoC BLE GCC）	cydevicekeil_trm.inc、cydevicegnu_trm.inc 或 cydevicerv_trm.inc 的弃用版本。

cydevicerv.inc（PSoC 5LP / PSoC 4 / PRoC BLE Real View）	
cyfitter.h	定义在代码生成步骤中计算出的所有实例的特定地址。该文件主要用于实例 API 实现，但高级用户能够从该文件中找出有意思的信息。
cyfitter_cfg.c	用于实现在执行 main 代码前配置器件所需要的方法和逻辑。您不应该使用该文件中所实现的任何内容。
cyfitter_cfg.h	包含引导（boot）固件时所使用的定义，用于在执行 main 代码前配置器件。您不应该使用该文件中所定义的任何内容。
cyfitterkeil.inc（PSoC 3） cyfittergnu.inc（PSoC 5LP / PSoC 4 / PRoC BLE GCC） cyfitterrv.inc（PSoC 5LP / PSoC 4 / PRoC BLE Real View）	该汇编文件与 cyfitter.h 等效。
project.h	该文件包含了该目录及其子目录下的其他所有头文件。它的存在是为了方便，您只需要一条 # include 语句便能包含所生成的所有头文件。
<project>.cyversion	由组件作者创建的文件。打开某个设计项目时，可使用该文件检查组件的更新。更多信息，请参阅组件创作者指南。

引导（boot）组件：

所有设计项目都包含了“boot”组件，这样是便于为所有引导固件文件（如 CyDma*、CyFlash*，等等）提供版本控制。组件目录默认隐藏该组件，并且不能将该组件放置在设计中。编译成功后，引导组件的文件位于 Workspace Explorer 中 Generated_Source/<Architecture Name>/cy_boot 文件夹内。

使用 PSoC Creator 早期的版本创建的设计会包含引导组件的早期版本。它们已经包含了所有 API 文件。如果需要更新 API 文件，可以使用组件更新工具将引导组件升级到最新版本。

更多有关信息，请参考系统参考指南中的内容。

组件 API：

所生成的源文件会包含实例化组件的 API（例如：counter_1.c、counter_1INT.c、counter_1.h），器件指定的文件夹中会列出这些 API。如果您不想为指定实例所生成的 API，可使用内置式参数 CY_SUPPRESS_API_GEN。更多有关信息，请参考配置组件参数节的内容。

结果文件：

Workspace Explorer（工作区浏览器）还包含了 **Results**（结果）选项卡，它包含以下文件：

- <project>.cycdx — 本文件包含了指定到组件调试窗口的 XML 信息。调试器将使用本文件来确定设计应显示的组件。请勿打开或修改本文件。更多有关本文件的信息，请参阅组件创建指南。
- <project>.rpt — 它是项目报告的文件。它包含了如何编程器件的信息，其中包括了如何利用目标器件资源的部分。高级用户可以查看本文件中提到的信息，从而确定是否存在更好的配置设计方法。
- <project>_timing.html — 这是静态时序分析报告的文件。更多有关信息，请参阅静态时序分析。

Results 选项卡还可以包含下面的多个其他文件：

- <project>.cyfit — 本文件是内部数据库，以 PSoC Creator 保留代码生成的结果数据。用户不能直接交换本文件。每次编译项目时都会生成本文件。
- <project>.elf — 本文件包含了 GCC 工具链的调试信息。用户不能直接交换本文件。
- <project>.ihx — 通过进行编译项目会生成 Intel HEX 文件，该文件只包含已编译的设计。用户不能直接交换本文件。
- <project>.hex — 通过将<project>.ihx 文件与选定的保护、配置和初始化设置结合使用可以生成 Intel HEX 文件。用户不能直接交换本文件。
- <project>.map — 本文件是由连接器生成的。根据工具链，它包含了如何使用器件存储器的详细信息、函数和变量被放置的位置以及其他详细信息。用户不能直接交换本文件。
- <file>.lst — 代码列表文件显示初始的 c 代码和生成的汇编程序。用户不能直接交换本文件。
- <project>.omf — 本文件包含了 Keil 工具链生成的调试信息。用户不能直接交换本文件。

另外，请参考：

- 工作区浏览器
- 组件目录
- 组件更新
- 静态时序分析

源代码控制

PSoC Creator 中没有任何源代码控制或版本控制系统中的集成。但您可以使用任何源代码控制系统，这样方便在工作时将电脑中的 PSoC Creator 文件更新到源控制，并从源控制中读取该文件。

本主题介绍了各种被发送到源代码控制系统中的文件和文件夹。您应该一直保存已创建或编辑的所有文件。如果将来其中的一个或多个文件不能被取出，则该项目可以不被正确加载，PSoC Creator 生成了错误或丢失了用户指定的代码。

另外，请参考赛普拉斯知识库文章：*PSoC® Creator™ 项目的版本控制 — KBA86358*（http://www.cypress.com/?id=4&rID=76644）

注意： 源控制中包含了更多的文件，一次可能要取出更多的文件，这样才能进行更新。这些文件不应处于“只读”状态，因为在这种状态下会限制 PSoC Creator 的功能。

项目文件/文件夹

下面是源代码控制中包含的项目文件和文件夹。有两种项目类型：设计项目和库项目。设计项目文件位于名称为 *<project>.cydsn* 的文件夹内。库项目文件位于名称为*<project>.cylib* 的文件夹内。

如下面列表所示，某些文件都用于两种项目类型，但是其他文件只指定于设计项目。通常，源代码控制中包含了所有这些文件和子文件夹。

: ': '

静态时序分析

作为成功构建的一部分，PSoC Creator 会自动执行设计中的静态时序分析（STA），以确定各时序内容，比如：

- 传输延迟：从输入端到输出端的完全组合延迟。
- 时钟到输出的延迟：从时钟源通过寄存器到输出端的延迟。
- 建立时间：是指寄存器的时钟信号在时钟引脚上被激活前数据必须传输到输入引脚所需的最短时间。
- 寄存器到寄存器的延迟：从寄存器输出端到寄存器输入端的延迟。如果两个寄存器使用同一个时钟，则静态时序分析仪会计算出最大频率。

更多有关设计实践和如何最好地使用 STA 报告的策略的信息，请参考应用笔记：AN81623。

静态时序分析确定设计的数字逻辑中的延迟，并计算每个时钟的最大频率。静态时序分析的报告介绍了设计中限制时钟频率的重要路径。如果实际的时钟频率超过了计算出的最大频率，则报告会显示设计中的时序违规。

静态时序分析只有在构建过程中才能访问设计，因此它不具备设计元素如何使用或动态进行的任何更改的信息（例如：更改时钟频率的固件）。由于这些限制，静态时序分析可以发出警报实际没问题的路径（由于采用设计的方式）。如果您已验证有问题的路径不被使用，您可以放心地忽略这些警报。例如，如果一个引脚组件被配置为“数字输入和数字输出”，则静态时序分析会发出警报表示有关路径正在被输出，然后回到了同一个引脚组件的输入端。如果没有配置使该路径得到使用，可以放心忽略该警报。

注意： 不应该忽略这些警报。需要时，请查看注意列表窗口中的警报和解决方案。

静态时序分析不具有信号如何产生或 PSoC 器件外所使用的信息。它可以显示有关这样信号的延迟，但不能自动查找时序违规。

要打开 STA 报告，请执行下面各操作：

成功构建设计后，STA 报告将作为工作区浏览器窗口下 **Results** 选项卡中的独立 HTML 报告。

双击*<project_name>.html* 文件，打开系统默认网页浏览器的报告。

报告的布局：

STA 报告包含标题、项目信息、“报告节”中所介绍的各种部分以及用于扩展/折叠这些部分中的信息的链接。

项目信息

每个 STA 报告包括以下项目信息：

- 项目名称和路径

- 本报告的构建时间
- 所使用的器件
- 器件版本
- 用于分析的温度
- 用于分析的电压
- 四个 I/O 区域中每一个的电压

扩展/折叠链接

这些链接包括：

- 扩展全部 — 此链接会扩展报告中的所有内容，使信息可见。
- 折叠全部 — 此链接会折叠报告中的所有内容，只能使部分标题可见。
- 显示所有路径 — 此链接扩展可应用的部分表，以显示一个完整路径时序的多视图。
- 隐藏所有路径 — 此链接可折叠可应用的部分表，只显示时序路径的某一行。

报告段落：

报告中可以包括下面各部分内容。除“时序违规”外，剩下的所有空段落将不会保存在报告中。

- 时序违规
- 时钟总结
- 寄存器到寄存器
- 异步时钟交叉
- 输入引脚到输出引脚
- 输入引脚到时钟元素
- 时钟元素到输出引脚
- 输入到输出使能
- 时钟元素到输出使能

时序违规部分

如果没有时序违规，本部分只显示“没有时序违规”文本。

如果存在时序违规，并且 DWR 工作范围被设置为整个范围，将显示一条注意信息。注意信息显示，更改工作范围会减小时序警报的数量。在注意指示信息后，会出现一行条目表，该表是违规类型、源时钟和目标时钟的组合。

此表被分成三个违规类型：Setup（设置）、Hold（保持）和 Asynchronous（异步），如下图所示。

违规	源时钟	目标时钟	余量（ns）
建立：			
	Clock_1	Clock_2	-2.501
	Clock_1	Clock_3	-3.458
	Clock_2	Clock_3	-2.344
保持：			
	Clock_1	Clock_2	-0.127
异步：			
	Pin_3	Clock_1	
	Pin_3	Clock_2	

每个违反源/目标时钟对仅包含一个条目。后面各节会详细介绍每个存在问题的路径。在 STA 报告中，您可以点击表中的条目，以跳转至具体的说明。

如果特定的违规类型并不存在于设计中，那么该标题也不会存在于列表内。每个违反时钟对的一行条目都会包括以下字段：

- 源时钟：违反路径的源时钟
- 目标时钟：违反路径的目标时钟
- 余量：故障中的余量时间（slack time）对于设置违规或保持违规，余量时间总是负数（指示一个违规）。对于异步时钟交叉违规，该字段为空白。

时钟总结章节

本节是一个简短的概述，介绍了设计当前实现的时钟频率要求和可达到的频率。下面是一个示例：

时钟	区域	额定频率	需要的频率	最大频率	违规（Violation）
BUS_CLK	CyBUS_CLK	48.000 MHz	48.000 MHz	无限制	
Clock_1	CyBUS_CLK	24.000 MHz	24.000 MHz	21.845 MHz	频率
Clock_2	CyBUS_CLK	12.000 kHz	12.000 kHz	22.387 MHz	
Pin_3	Pin_3	18.000 MHz	18.000 MHz	45.239 MHz	
Pin_5	Pin_5	未知	未知	21.764 MHz	未知

系统中每个时钟的一行条目均具有以下字段：

- 时钟：本时钟的名称。第一个条目始终为 BUS_CLK。其余条目按字母顺序显示。
- 区域：这是本时钟属于的时钟区域。同一区域中的时钟相互同步。
- 额定频率：根据设计的所需频率，工具会计算出该频率。它是由分频器设置分频的源时钟的直接值。由于与 MASTER_CLK 同步，所以它不包括精度或抖动的调整。在不可确定时钟频率（比如：引脚的时钟）中，频率被显示为“Unknown”（未知）。
- 需要的频率：是指使用该频率运行的路径必须满足时序的频率。本时钟脉冲频率是额定时钟脉冲频率与最大同步抖动的和。所需频率（MHz）：是设计中指定的时钟频率。如果它是一个没有时钟频率属性的异步时钟，将被显示为“未知”。

- 最大频率：是使该时钟可以安全运行的频率。根据设计中影响该时钟的最慢路径可计算出该频率。如果时钟不受限制，则“N/A”将出现在此列中。
- 违规：有两种违规情况，即：“Frequency”（频率）或“Unknown”（未知）。当最大频率小于所需频率时，会显示“Frequency”。当所需频率为未知时，会显示“Unknown”。这些违规均以红色显示。如果没有违规，则该字段显示为空白。

寄存器到寄存器章节

本节介绍了寄存器到寄存器的时序路径，其中源时钟和目标时钟是相同的，或者它们互相同步。“异步时钟交叉”一节介绍了所有异步时钟交叉。但本节介绍的是异步时钟和它本身间的路径。

本节有两个主要的主题，即：建立和保持。

“建立”主题

本主题首先介绍源时钟，然后介绍目标时钟。每一个时钟都按字母顺序排列。如果使用了时钟的下降沿，则将其视为独特的时钟，并且下降沿同时钟名称一同被指出。

源时钟标题中列出了源时钟名称和所需的频率。例如：

```
Source clock: Clock_1 (Required Freq. 24.000 MHz)
```

目标时钟标题中列出了目标时钟名称和所需的频率。它还包括了用于路径延迟的要求。它是取决于源时钟和目标时钟的组合。使用相对时钟边沿或同步到 BUS_CLK 的其他时钟会影响路径延迟。例如，目标时钟与源时钟不同，但它们都被同步到 BUS_CLK（在这个例子中，它以频率为 48 MHz 运行）。

```
Destination clock: Clock_2 (Required Freq. 12.000 MHz)
Path Delay Requirement: 20.833ns (48 MHz)
```

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标时钟	最大频率（MHz）	延迟（单位为ns）	余量（ns）	违规（Violation）
dff_reg1:macrocell.mc_q	Net_5:macrocell.mc_d	8.000	12.500	-2.500	SETUP（建立）
dff_reg1:macrocell.mc_q	Net_6:macrocell.mc_d	11.000	9.091	0.909	

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
宏单元	U(3,1)	1	dff_reg1	.cr_clk	.q	1.250
路由		3	dff_reg1	.q	.main_0	3.608
宏单元	U(3,2)	1	Net_5	.main_0	.mc_d	2.810
宏单元	U(3,2)	1	Net_5		Setup	0.875
时钟					Skew	0.548

dff_reg1:macrocell.mc_q	Net_7:macrocell.mc_d	11.500	8.696	1.304	

每个路径的一行条目均具有以下各字段：

- 源：是指路径的第一个寄存器
- 目标：是指路径的目标寄存器
- 最大频率（MHz）：是基于特定路径的最大频率，单位为 MHz（等于 1/延迟）。
- 延迟（单位为 ns）： 是整个路径的延迟（单位为 ns），包括建立时间和任何时钟的时滞。
- 余量（ns）：余量是（路径延迟要求 - 实际的延迟）。用红色显示小于 0 的所有余量时间。
- 违规： 只有一种类型的违规，既“SETUP”（建立）。负余量显示为红色的“SETUP”，其他字段为空白。

条目按从最小的余量到最大的余量的顺序排列。该报告包含了违规建立时间的所有路径。然后，每一对时钟显示多达 10 个条目。

可以扩展每一个条目来显示详细的完整路径。每个条目均包含以下各字段：

- Type（类型）：是指所涉及的函数类型
- Route（路由）：用于所有布线
- Macrocell（宏单元）
- Datapath（数据路径）
- Control（控制）：表示控制寄存器
- Status（状态）：表示状态寄存器
- Clock（时钟）：用于时钟时滞条目
- IO：表示器件的引脚
- Location（位置）：是指“type”字段中所指出的单元位置，它显示了所有类型（除 Route 和 Clock 外）。
- U(x,y)：是 UDB 阵列中所有单元的格式（x、y 是 UDB 的坐标）
- Pi[j]：引脚的格式（i 是端口编号，j 是端口内的引脚编号）
- Fanout：表示信号的 Fanout 数量。Fanout 数量通常为 1（除 Route 类型外，因为在这种情况下它会表示同一个信号驱动的目标数量）。
- 实例/网络：是指与这个路线有关的实例名或网络。

  注意： 宏单元的名称可能与组件的原始名称不匹配。fitter 可以将宏单元与其他网络和宏单元结合起来。该过程会导致一些名称信息被丢失。已经与线网结合的宏单元可以继续使用网名，比如：“Net_73”。
- Source and Dest（源和目标）：是单元中的源引脚和目标引脚位于布线的该部分的两端。目标字段值（源为空时）也适用于一些特殊情况。
- Setup（设置）：特殊情况下，每次建立路径结束后存在该目标字段，以表示所需的寄存器建立。
- Skew（时滞）：是指特殊情况下仅适用于路由的时钟（全局时钟没有时滞条目）。

- Delay（延迟）：是指为整个路径的行条目指定的部分增加延迟，单位为 ns。所有增加的延迟输入总和要等于某行延迟的总和。

“保持”主题

本主题首先介绍了源时钟，然后介绍目标时钟。至少有时钟中的一个被作为路由的时钟时，才存在源/目标对。本主题的命名和排序与“建立”主题的一样，但除时钟频率不包括源和目标标题外。例如：

```
Source clock: Clock_1
Destination clock: Clock_2
```

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	余量（ns）	违规（Violation）
dff_reg1:macrocell.mc_q	Net_5:macrocell.mc_d	-0.541	HOLD （保持）
dff_reg1:macrocell.mc_q	Net_6:macrocell.mc_d	2.965	

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
宏单元	U(3,1)	1	dff_reg1	.cr_clk	.q	1.000
路由		3	dff_reg1	.q	.main_0	3.122
宏单元	U(3,2)	1	Net_5	.main_0	.mc_d	2.523
宏单元	U(3,2)	1	Net_5		Hold	-0.120
时钟					Skew	-3.560

dff_reg1:macrocell.mc_q	Net_7:macrocell.mc_d	5.234	

每一个路径的一行条目的格式与“建立”主题条目的格式相同，但具有以下变化：

- “保持”不包括 FMax 和延迟内容
- 本主题中 Slack 的计算方法同“建立”中的延迟计算方法相同。它是具体路径中存在的所有延迟条目的总和。
- 对于“保持”违规，违规类型为“HOLD”。

条目按从最小的余量到最大的余量的顺序排列。该主题包含了违规保持时间的所有路径。此后，每一对时钟显示多达 10 个条目。

可以展开每一个条目以显示详细的完整路径。每个条目与“建立”主题中的条目相同，但有存在下面的变化：

- 根据最好情况下的路径而不是最坏情况下的路径来计算延迟
- 目标的特殊条目是：
- Hold（保持）：特殊情况下，每个保持路径结束后会存在该内容，表示所需的寄存器保持。正保持要使用负数显示，以便使增加延迟的总和等于余量时间。
- Skew（时滞）：时滞始终存在，因为在没有时钟时滞时，不会发生该架构中的保持时间违规。

- Delay（延迟）： 增加延迟的总和等于余量时间。总和的运算必须计算“保持”和“时钟”时滞条目的符号，这样这些条目总和的运算才正确。

异步时钟交叉章节

本节介绍了时钟区域之间的所有路径，其中源时钟和目标时钟互相不同步。

本节被分为源时钟和目标时钟两个子节。只有这些时钟彼此异步，并且它们均不与 BUS_CLK 同步时，才能存在源/目标对。本主题的命名和排序同“建立”主题中的一样，但除了时钟频率不包括源和目标标题外。例如：

```
Source clock: Clock_1
Destination clock: Clock_2
```

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	延迟（单位为ns）
dff_reg1:macrocell.mc_q	Net_5:macrocell.mc_d	12.500
dff_reg1:macrocell.mc_q	Net_6:macrocell.mc_d	9.091

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
宏单元	U(3,1)	1	dff_reg1	.cr_clk	.q	1.250
路由		3	dff_reg1	.q	.main_0	3.608
宏单元	U(3,2)	1	Net_5	.main_0	.mc_d	2.810
宏单元	U(3,2)	1	Net_5		Setup	0.875
时钟					Skew	0.548

dff_reg1:macrocell.mc_q	Net_7:macrocell.mc_d	8.696

每一个路径的一行条目的格式与“设置”条目中的格式相同，但存在下面的变化：

- 时钟交叉不存在 FMax 和 Slack
- 进行计算延迟，以便提供信息给用户，但计算目的不是为了查看是否满足时序要求。
- 不存在 Violation 字段。

条目按从最长的延迟到最短的延迟的顺序排列。每一对时钟显示多达 10 个条目。

可以展开每一个条目以显示详细的完整路径。路径的详细条目类似于建立中存在的条目。

输入到输出章节

本节介绍了通过器件组合路径。它包含每个源/目标对中最长的结合路径的一个单行条目。这些条目按从长延迟到短延迟的顺序排列。

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	延迟（单位为ns）
Pin_3(0):iocell._fb	Pin_4(0):iocell.pad_out	56.823
Pin_3(0):iocell._fb	Pin_6(0):iocell.pad_out	54.113

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
引脚	P5[6]	1	Pin_3(0)	.pad_in	.fb	15.258
路由		2	Net_3	.fb	.main_0	5.714
宏单元	U(3,2)	1	Net_4	.main_0	.q	3.350
路由		1	Net_4	.q	.input	6.297
引脚	P5[2]	1	Pin_6(0)	.input	.pad_out	23.495

Pin_5(0):iocell._fb	Pin_6(0):iocell.pad_out	42.696

每个路径的一行条目均具有以下各字段：

- 源： 该结合路径的开始
- 目标： 该结合路径最后的目标
- 延迟（单位为 ns）： 沿着路径的延迟，单位为 ns。

可以展开每一个条目来显示详细的完整路径。路径的详细条目同建立条目中的一样，但除从来都不存在的时钟时滞和建立的特殊条目外。

“从输入引脚到时钟元素”章节

本节介绍了进入器件内终止在时钟元素中的路径。

本节按基于目标时钟，被分为多个子章节。这些子节以字母顺序排列，并使用了时钟的名称标签。例如：

```
Destination clock: Clock_1
```

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	延迟（单位为ns）
Pin_3(0):iocell._fb	Net_7:macrocell.mc_d	31.763
Pin_3(0):iocell._fb	Net_6:macrocell.mc_d	24.657

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
引脚	P5[6]	1	Pin_3(0)	.pad_in	.fb	15.258
路由		2	Net_5	.fb	.main_0	5.714
宏单元	U(3,2)	1	Net_6	.main_0	.mc_d	2.810
宏单元	U(3,2)	1	Net_6		Setup	0.875

Pin_5(0):iocell._fb	Net_6:macrocell.mc_d	22.376

从输入到时钟的最长路径以单行条目显示。这些条目按从长路径到短路径的顺序排列。这些条目与“从输入引脚到输出引脚”一节中的条目相同，但除了每个路径有建立条目外。

“从时钟元素到输出引脚”章节

本节介绍从时钟元素到器件输出引脚的路径。

本节根据源时钟被分为多个子节。这些子节以字母顺序排列，并使用了时钟的名称标签。例如：

```
Source clock: Clock_1
```

下面是三个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	延迟（单位为ns）
dff_reg1:macrocell.mc_q	Pin_4(0):iocell.pad_out	43.873
dff_reg1:macrocell.mc_q	Pin_6(0):iocell.pad_out	30.459

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
宏单元	U(3,1)	1	dff_reg1	.cr_clk	.q	1.250
路由		2	dff_reg1	.q	.input	5.714
引脚	P5[2]	1	Pin_6(0)	.input	.pad_out	23.495

dff_reg2:macrocell.mc_q	Pin_6(0):iocell.pad_out	29.745

从时钟元素到每个输出引脚的最长路径以单行条目来显示。这些条目按从长路径到短路径的顺序排列。这些条目都具有相同的扩展能力和相同的字段，这些内容与“从输入引脚到输出引脚”节中的相同。

“从输入到输出使能”章节

本节与“从输入引脚到输出引脚”章节相同，但除了目标路径是输出端口的“输出使能”信号而不是输出端口的数据外。除了每个路径均有两个条目（一个用于打开输出信号（TURNON），另一个用于关闭输出信号（TURNOFF））外，处理报告的方法都是相同的。即使延迟相同，但这两个情况仍被处理为两个不同的条目。

下面是两个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	类型	延迟（单位为ns）
Pin_9(0):iocell._fb	Pin_6(0):iocell.pad_out	TURNON	49.625
Pin_9(0):iocell._fb	Pin_6(0):iocell.pad_out	TURNOFF	49.625

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
引脚	P3[5]	1	Pin_9(0)	.pad_in	.fb	15.258
路由		1	Net_26	.fb	.main_0	5.714
宏单元	U(2,1)	1	tmpOE__Pin6_net_0	.main_0	.q	3.350
路由		1	tmpOE__Pin6_net_0	.q	.oe	6.331
引脚	P5[2]	1	Pin_6(0)	.oe	.pad_out	18.972

为了区别这两个条目，请将另一个字段添加到某一行条目中：

- 类型：TURNON 或 TURNOFF 都取决于使能还是禁止输出端。

“从时钟元素到输出使能”章节

本节与“从时钟元素到输出引脚”章节相同，但除了目标路径是输出端口的“输出使能”信号而不是输出端口的数据外。除了每个路径均有两个条目（一个用于打开输出信号（TURNON），另一个用于关闭输出信号（TURNOFF））外，处理报告的方法都是相同的。即使延迟相同，但这两个情况仍被处理为两个不同的条目。

下面是两个时序路径的示例，其中第二个条目可以扩展，这样能够显示完整的路径。

源	目标	类型	延迟（单位为ns）
dff_reg3:macrocell.mc_q	Pin_6(0):iocell.pad_out	TURNON	22.869
dff_reg3:macrocell.mc_q	Pin_6(0):iocell.pad_out	TURNOFF	22.869

类型	位置	Fanout	实例/网络	源	目标	延迟（单位为ns）
宏单元	U(1,1)	1	dff_reg3	.cr_clk	.q	1.250
路由		1	dff_reg3	.q	.oe	5.714
引脚	P5[2]	1	Pin_6(0)	.oe	.pad_out	15.905

为了区别这两个条目，请将另一个字段添加到某一行条目中：

- 类型：TURNON 或 TURNOFF 都取决于使能还是禁止输出端。

另外，请参考：

- 生成的文件
- Workspace Explorer（工作区浏览器）

CyPrjMgr 指令行工具

cyprjmgr.exe 是一各指令行工具，通过使用该指令行工具可以使用项目的通用管理功能。可以从指令行调用该工具，在工作区/项目内可以使用该工具执行多项功能。另外，它还具有灵活性，通过该指令行工具能够设置编译配置。

语法：

```
cyprjmgr
[-h]
[-ver]
[-wrk <workspace_name>]
[-clean]
[-build]
[-rebuild]
[-archive <archive_level> <archive_as_zip>]
[-t <toolchain>]
[-c <config>]
[-p <TopProject>]
[-n <TopDesign>]
[-d <selectedDev>]
[-m <paramsFile>]
```

```
[-import <Source_Project> <Source_Component>]
[-rename <component_name> <new_name>]
[-delete <component_name>]
[-exclude <component_name>]
[-list][-l <NewPrjName>]
[-e]
[-def]
[-s]
[-v <visibility>]
[-prj <Target_Project>]
[-cmp <Target_Component>]
[-addprj <prj_path>]
[-cp <path>]
[-con <Target_Project>]
[-batch <file_name>]
[-updateComp <source_project> <source_component>]
[-updatePrj <source_project>]
[-updateInst] [-updateDWInst]
[-forceWrite]
[-noCustBuild] [-noRefresh]
[-ol <compiler optimization level>]
[-warn <High|Low|None>]
[-buildPreCompCust <Project>]
[-hex <path>]
[-updateInstIfNeeded]
[ignoreDepsWarning]
[-export <IDE>]
```

该工具预期各参数以指定的格式传输。

- 第一个参数必须为`-w`，然后是工作区的名称。
- 所有其他参数均是可选的，并且可能被用于设置构建环境。所有可选的参数将适用于后面是`-w` 参数的工作区。

选项：

下表列出并说明了多个可选参数：

参数	说明
`-wrk、-w`	指定使用的工作区
`-prj`	指定目标项目，在这里所有指令行选项都将被应用。在没有目标项目被指定时，工作区的顶级项目会成为目标项目
`-cmp, -o`	指定目标组件，在这里所有选项库将被应用。如果没有目标组件被指定时，目标项目的顶级模块便会成为目标组件
下面四个选项均被应用于整个工作区，除非目标项目通过使用`-prj` 参数被设置：	
`-clean`	清除工作区/项目
`-build`	编译工作区/项目
`-rebuild`	重新编译工作区/项目
`-archive`	以不同的存档等级（完全/典型）将工作区/项目存档到 zip 或非 zip 文件中
`-h`	显示帮助信息
`-ver`	显示 cyprjmgr 的版本编号和编译编号

-t	设置需要使用的工具链（Keil、ARM，等等）
-c	设置需要使用的配置（调试、释放）
-ol	设置编译器的优化等级
-warn	设置编译器的警报等级
-p	设置工作区中的顶级项目
-n	设置顶级项目中的顶层设计
-d	设置目标项目中会被编译的选定器件
-m	用于在顶级项目的顶层设计中覆盖掉默认的原理图参数值的参数文件
-import	将源项目中的源组件导入到工作区中的目标项目
-rename	对工作区中目标项目的组件名称（component_name）重命名（new_name）
-delete	删除磁盘上的组件
-exclude	从工作区的目标项目中排除组件
-list	列出目标项目中的所有组件
*L	向工作区内添加名称为<NewPrjName>的空白库项目
*E	列出目标项目中符号所在的目录位置
-def	列出目标项目中所有符号的默认参数值
-s	列出目标项目的外部依赖属性
-v	使能或禁用目标组件的可见属性
-addprj	向工作区内添加现有的项目<prj_path>
-cp	将全部工作区复制到<path>指定的位置，在复制的工作区中执行所有指令行选项
-con	检查工作区中目标项目的一致性
-batch	读取包含一系列指令的文件，每一行包括一条指令。逐一执行指令。使用批量选项时，所有其它可选的选项均被忽略
-hex	以十六进制的格式设置 Bootloadable 项目的路径
-updateComp	根据源项目更新目标项目中的组件
-updatePrj	根据源项目更新目标项目
-updateInst	将原理图中的实例更新为最新组件
-forceWrite	使只读文件变成可写内容，并进行修改
-noCustBuild	延迟自定义程序 DLL 的编译，直到结束项目的编译过程为止（例如，在导入期间）
-noRefresh	禁用刷新管理器的更新性能（使用时，要特别注意）
-buildPreCompCust	编译项目的自定义程序
-updateDWInst	将设计范围实例更新为最新组件
-updateInstIfNeeded	将实例更新为最低的有效版本
-ignoreDepsWarning	编译基元期间抑制 SystemDepNotFoundOnDisk 警报
-export	将项目移植到目标 IDE。有效的目标 IDE 是 EWA、Eclipse 和 uVision

在用户工作区文件中保存的参数包括：

- 工作区浏览器的状态（选中的选项卡、节点的展开/收缩状态）
- 断点
- 浏览文档
- 配置（释放和调试）

在用户项目文件中保存的参数包括：

- 编译的依赖属性（编译过程依赖于的非项目文件）
- 最后的编译指令行（适用于所有工具）
- 器件选型器的状态（隐藏列、滤波状态）

cyprjmgr.exe 的必要参数：

- `-w <workspace path>`（无默认值）
- `-build/clean/rebuild`（无默认值）
- `-t <toolchain name>`（默认值是每个项目的选定工具链）
- `-c <config name>`（默认值是在工作区用户文件中指定的值，如果未指定该值，默认将其设置为 Debug）
- `-prj <project name>`（默认值是活动项目）

其它选项：

其它选项都是非必须的，因为在进行调试时，这些信息默认可用于工作区/项目，但是您可以通过指令行控制它们。这些选项包括：

- `-ol` — 设置编译器的优化等级
- `-warn` — 设置编译器的警报等级
- `-p` — 设置工作区中的顶级项目
- `-n` — 设置顶级项目中的顶层设计
- `-d` — 设置目标项目中将被编译的选定器件

使用场合和示例

–clean / -build / -rebuild

```
cyprjmgr.exe –w workspace –clean
cyprjmgr.exe –w workspace –build
cyprjmgr.exe –w workspace –rebuild
```

该选项分别对工作区进行清除/编译/重新编译操作。每次运行工具，只能执行清除、编译或重新编译等三个操作中的一个。

-h

```
cyprjmgr.exe –h
```

该项将显示工具的帮助信息

–t

```
cyprjmgr.exe –w workspace –build –t “ARM CM3-GCC 4.2.1”
```

使用被指定的工具链编译工作区。如果用户未指定任何工具链，便会使用上一次编译工作区时所使用的工具链。

–c

```
cyprjmgr.exe –w workspace –build –c Release
```

使用被指定的编译配置（在该示例中为释放配置）来编译工作区

–p

```
cyprjmgr.exe –w workspace -build –p Design01
```

该项将工作区中的 *Design01* 项目设置为顶级项目，并编译工作区

–n

```
cyprjmgr.exe –w workspace –build –n component01
```

将顶级项目中的 *component01* 组件设置为顶级组件，并编译工作区

–d

```
cyprjmgr.exe –w workspace –build –d CY8C3866AXI-040
```

该项将设置工作区中所有项目的器件，并进行编译。

```
cyprjmgr.exe –w workspace –build –d CY8C3866AXI-040 –prj Design01
```

该项将设置目标项目的器件，并进行编译。

```
cyprjmgr.exe –w workspace –build –d CY8C3866AXI-040 –t “DP8051-Keil Generic” –prj Design01
```

该项用于设置目标项目‘Design01.cyprj’的器件和工具链，并进行编译

-rev

```
cyprjmgr.exe –w workspace –rev ES1
```

该项用于设置工作区中所有项目的选定器件版本。

```
cyprjmgr.exe –w workspace –build  –d CY8C3866AXI-040  –rev ES1 –prj Design01
```

该项用于设置目标项目‘Design01.cyprj’ 的‘ES1’器件版本，并进行编译。

–m

```
cyprjmgr.exe –w workspace –build –m params_file
```

该项会在文本文件中读取参数及其值，并覆盖原理图中的值。

参数文件的格式为：

参数文件中的参数和其值的定义格式为 name=value。定义实例名称的格式为 *inst_name=value*。

例如：

inst_name=and_1
NumTerminals=8
TerminalWidth=4

inst_name=Counter_1
Resolution=16

实例 and_1 *中的* NumTerminals *和* TerminalWidth *参数值和* Counter_1.*实例中的* Resolution *参数值将被更改。*

–prj

```
cyprjmgr.exe –w workspace –prj Project01
```

该项将工作区中的 Project01 设置为目标项目。同一条指令行上的所有库选项均在 Project01 上进行。如果未选定任何目标项目，则将工作区的顶级项目作为目标项目使用。

```
cyprjmgr.exe –w workspace –build –prj Project01
```

通过该项，可以只编译 *Project01*（如果 *Project01* 位于*工作区*中）和它的依赖属性，而不编译全部工作区。该项等于选择性地编译 GUI 中的某个项目。

–o, -cmp

```
cyprjmgr.exe –w workspace –o Component01
cyprjmgr.exe –w workspace -cmp Component01
```

该项将 Component01 设置为目标项目中的目标组件。针对组件的同一条指令行上的所有库选项将操作于目标组件。如果未选定目标组件，便会将目标项目中的顶级组件作为目标组件使用。

–import

```
cyprjmgr.exe –w workspace –import SourceProject SourceComponent
```

该项将源项目中的源组件导入到工作区中的顶级项目内。

```
cyprjmgr.exe –w workspace –import SourceProject SourceComponent –prj TargetProject
```

该项将源项目中的源组件导入到工作区中的目标项目内。

–rename

```
cyprjmgr.exe –w workspace –rename ComponentName NewComponentName
```

该项将顶级项目中的顶级组件的名称从 ComponentName *变更为* NewComponentName。

```
cyprjmgr.exe –w workspace –rename ComponentName NewComponentName –prj Project01
```

该项将重命名工作区中 Project01 的顶级组件

```
cyprjmgr.exe –w workspace –rename ComponentName NewComponentName –prj –Project01 –o
Component02
```

该项将重命名工作区中 Project01 的 Component02

–delete

```
cyprjmgr.exe –w workspace –delete Component01
```

该项将从磁盘上删除工作区中顶级项目的 Component01。

```
cyprjmgr.exe –w workspace –delete Component01 –prj Project01
```

该项将从磁盘上删除工作区中 Project01 项目的 Component01。

–exclude

```
cyprjmgr.exe –w workspace –exclude Component01
```

该选项可从工作区中的顶级项目中删除 Component01 实例。

```
cyprjmgr.exe –w workspace –exclude Component01 –prj Project01
```

该项可从工作区中的 Project01 项目中删除 Component01 实例。

–list

```
cyprjmgr.exe –w workspace –list
```

该项列出了工作区中顶级项目的所有组件。

```
cyprjmgr.exe –w workspace –list –prj Project01
```

该项列出了工作区中 Project01 项目的所有组件。

–l

```
cyprjmgr.exe –w workspace –l LibraryName
```

该项将新的空白库项目添加到工作区内。

–e

```
cyprjmgr.exe –w workspace –e
```

该项列出了工作区中顶级项目的所有符号所在的目录位置。

```
cyprjmgr.exe –w workspace –e –prj Project01
```

该项列出了工作区中 Project01 项目的所有符号所在的目录位置。

```
cyprjmgr.exe –w workspace –e –prj Project01 –o Component01
```

该项列出了工作区中 Project01 项目的 Component01 符号所在的目录位置。

–def

```
cyprjmgr.exe –w workspace –def
```

该项列出了工作区中顶级项目的符号内的参数值。

```
cyprjmgr.exe –w workspace –def –prj Project01
```

该项列出了工作区中 Project01 项目的符号内的参数值。

```
cyprjmgr.exe –w workspace –def –prj Project01 –o Component01
```

该项列出了工作区中 Project01 项目的 Component01 内的参数值。

–s

```
cyprjmgr.exe –w workspace –s
```

该项列出了工作区中顶级项目的外部依赖属性。

```
cyprjmgr.exe –w workspace –s –prj Project01
```

该项列出了工作区中 Project 01 项目的外部依赖属性。

–v

```
cyprjmgr.exe –w workspace –v false –o Component01
```

该项将禁用工作区中顶级项目的符号 Component01 的可见性。

```
cyprjmgr.exe –w workspace –v false –o Component01 -prj Project01
```

该项将禁用工作区中 Project01 项目的符号 Component01 的可见性。

–addprj

```
cyprjmgr.exe –w workspace –addprj Project01
```

该项将现有的 CyDesigner 项目（Project01）添加到工作区中。

–con

```
cyprjmgr.exe –w workspace –con
```

该项将检查工作区中顶级项目的一致性。

5. 该项目能够识别到目标项目文件夹中的所有文件。

6. 项目识别到的所有文件均位于“Canonical Name”（规范名称）属性指定的位置中。

```
cyprjmgr.exe –w workspace –con –prj Project01
```

该项将检查工作区中 Project01 项目的一致性。

–cp

```
cyprjmgr.exe –w workspace –cp NewLocation
```

该项将整个工作区文件夹复制到 NewLocation 中。当前，在同一条指令行中的所有其它选项将在 NewLocation Workspace 中进行。

-ver

```
cyprjmgr.exe -ver
```

该项将显示应用的版本编号。

-batch

```
cyprjmgr.exe –w workspace –batch file_name
```

该项将读取包含一系列 CyPrjMgr 指令（每行一条指令）的文本文件。CyPrjMgr 工具将读取每一行，然后执行指令，如果执行成功便继续执行下一条指令。如果执行失败，该工具将退出，同时发出故障信息。

使用–batch 时，所有其它可选的选项均被忽略

-updateComp

```
cyprjmgr.exe –w workspace -updateComp source_project source_component
```

该项将更新工作区中顶级项目的组件。

```
cyprjmgr.exe –w workspace –prj Project01 - updateComp source_project source_component
```

该项将更新工作区中 Project01 项目的组件。

组件的更新应遵守下面规则：

- 工具将寻找目标组件中具有与源组件的基名称相同的组件。如果未找到，便将该组件导入到目标项目中。
- 如果找到了目标组件，将源组件中的文件与目标组件中的相应文件进行比较：
  - 如果源组件中包含在目标组件中不存在的文件，将进行 UPDATECOMP。
  - 如果目标组件中包含与源文件中具有同样的名称的文件，将对两个文件的内容进行比较。如果他们的内容不一样，将进行 UPDATECOMP。
- 如果所有文件均相同，将不再更新组件。

UPDATECOMP：从目标项目中删除目标组件。将源组件导入到目标项目内。

-updatePrj

```
cyprjmgr.exe –w workspace –updatePrj source_project
```

该项将更新工作区中的顶级项目。

```
cyprjmgr.exe –w workspace –prj Project01 –updatePrj source_project
```

该项将更新工作区中的 Project01 项目。

项目的更新遵守下面规则：

- 比较源项目和目标项目的工具链。如果它们间存在差异，将进行 UPDATEPRJ。
- 如果两个工具链相同，将源项目的工具链设置与目标项目的配对进行比较。如果它们间存在差异，将进行 UPDATEPRJ。
- 将源项目中的非组件文件与目标项目中的（例如，main.c、.cydwr 和.cyprj 文件）进行比较。如果存在差异，则进行 UPDATEPRJ。
- 完成上述步骤后，便不再进行更新。

UPDATEPRJ：从工作区中删除目标项目和它所在的目录。将源项目和他的所有文件/文件夹/组件复制到工作区中目标项目的位置。

-archive

```
cyprjmgr.exe –w workspace –archive complete zip
```

该项将全部工作区存档到 zip 文件中。

```
cyprjmgr.exe –w workspace –archive typical nozip –prj Design01
```

该项以典型的存档等级对 Design01 项目进行存档。存档项目是源项目的原样复制（在上面的示例中，Design01 是源项目）。

工作区/项目的存档应遵守下面的规则：

- 如果未指定目标项目，它将存档整个工作区，否则它只存档指定的项目。
- 有两个存档等级，即“complete”（完整）和“typical”（典型）。使用‘complete’存档等级时，将存档整个项目/工作区，反之仅使用‘typical’存档等级。
- 通过所提供的‘zip’或‘nozip’存档选项，用户能够使用存档到 zip 文件的内容或复制所需内容。

-updateInst

```
cyprjmgr.exe –w workspace –updateInst
```

该项为指定工作区中的所有项目将原理图中的所有组件实例（包括项目的引导组件）更新为可用的最新版本组件。

```
cyprjmgr.exe –w workspace –updateInst –prj Design01
```

该项为目标项目“Design01”中将原理图中的所有组件实例（包括引导组件）更新为可用的最新版本组件。

工作区/项目的更新需要遵守下面的规则：

- 该项将原理图中的所有实例更新为最新的组件实例。
- 如果指定了目标项目，它只会更新指定项目中的组件实例。如果未指定项目，它会将工作区中的所有项目更新为最新的组件实例。

-forceWrite

```
cyprjmgr.exe –w workspace –d CY8C3866AXI-040 –forceWrite
```

该项为指定工作区中的所有项目设置其器件，甚至工作区和项目处于只读状态。

强制对工作区/项目的编写需要遵守下面的规则：

- 如果工作区和项目文件处于只读状态，那么它们被设置为可写状态
- 向‘cyprjmgr’工具传入任何将会更改文件的其它参数时，必须使用该选项保存只读文件中的更改内容。
- 如果不使用该项，则不会保存对只读文件进行的更改。

-noCustBuild

```
cyprjmgr.exe –w workspace -build –noCustBuild
```

该项将延迟自定义程序 DLL 的编译，直到完成项目的编译为止。

-noRefresh

```
cyprjmgr.exe –w workspace -build –noRefresh
```

该项在编译项目期间会禁用刷新管理程序。

-buildPreCompCust

```
cyprjmgr.exe –w workspace -build –buildPreCompCust project
```

该项将编译特定项目的定制器。

-updateDWInst

```
cyprjmgr.exe –w workspace -build –updateDWInst
```

该项会将原理图中的实例更新为最新版本

-ol

```
cyprjmgr.exe –w workspace -build –ol level
```

该项会设置编译器的优化等级。它的等级取决于编译器。

-warn

```
cyprjmgr.exe –w workspace -build –warn[High|Low|None]
```

该项将设置编译器的警报等级。

-hex

```
cyprjmgr.exe –w workspace -build –hex path
```

该项将设置 Bootloadable 目标的十六进制路径

-updateInstIfNeeded

```
cyprjmgr.exe –w workspace -build –updateInstIfNeeded
```

该项将组件更新为最低的有效等级。

-ignoreDepsWarning

```
cyprjmgr.exe –w workspace -build –ignoreDepsWarning
```

该项在编译基元期间抑制 SystemDepNotFoundOnDisk 警报。

-export

```
cyprjmgr.exe –w workspace -export uVision
```

该项将项目移植到 uVision。

Keil 编译器

PSoC Creator 包含 Keil 编译器。它是功能齐全的 C 语言编译器，它的优化等级被限为等级 5。如果您需要更高等级的优化，可以联系 Keil 以升级编译器。

Keil 编译器在安装后的 30 天内正常运行，此时间到期后，它的代码大小将受限。因此它不能链接到大小超过 2k 的程序。要解决代码大小限制的问题，您的编译器必须注册。您只要填写网上的表格就可以免费注册。

如何注册编译器：

1. 请点击 PSoC Creator **Help** 菜单，然后选择 **Register > Keil...**

   将出现一个 GUI 界面，显示有关各种 Keil 安装的信息。

2. 在该 GUI 中，请点击 **Get LIC via Internet** 按键。

   将打开用于注册编译器的网页。该网页包括一个字段组，通过它您可以输入有关 Keil 安装的数据。必须填入所有粗体字段的标题。

   - 第一个字段由 PSoC Creator 自动填写。

- 第二个字段也会被自动填写为 IKA1P-M6Q0E-8W7ST 代码。

3. 输入其他要求的值，然后按下 **Submit** 按键。
4. 您会通过电子邮件收到一个数字，要将这个数字填入到 Keil 注册对话框中的 LIC 字段内。打开对话框，将新的 LIC 值粘贴到“**New License ID Code (LIC)**”字段并按下“**Add LIC**”按键。
5. LIC 被添加，这时您将拥有完全注册版本的 Keil 编译器。

如果您对上述注册流程存在任何问题，请使用 Keil µVision3 应用。在 **File** 菜单下，选择 **License Management** 项，这样便会打开注册对话框。

另外，请参考：

- PSoC 3 中的可重入代码
- Keil 文档

PSoC 3 中的可重入代码

由于堆栈数量和 RAM 可用空间的限制，以及性能的原因，使用 Keil 编译器编译的函数默认情况下并不是可重入代码。 因此，根据参数的数量和类型以及局部变量的使用状况，在多数情况下，不能同时多次调用同一个函数。同一个函数被两个不同的中断或被同一个中断和主程序调用时，将发生函数的同时调用情况。虽然默认设置是不可重入的，但函数支持可重入性。

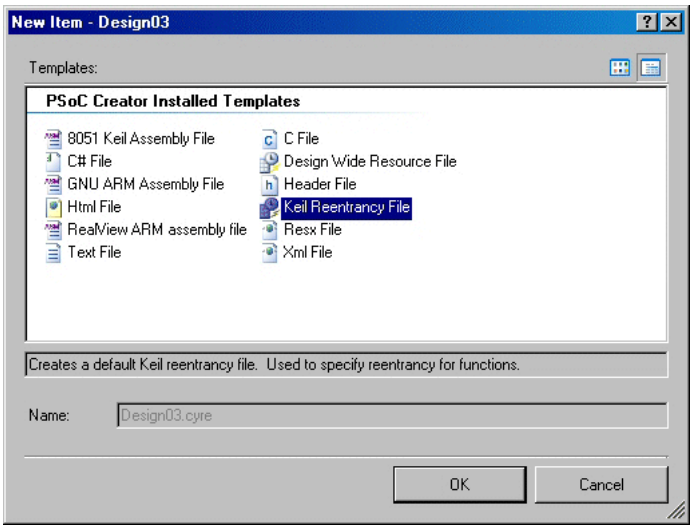
将所生成的 API 函数设置为可重入函数：

PSoC Creator 在系统中生成各组件的函数。PSoC Creator 提供了“可重入性文件”（*.cyre），通过它您能够根据具体情况设置可重入的函数。该文件明确指定了应设置为可重入的函数。文件中的每一行都必须是单一的函数名称。在编译期间，文件包含的所有备选函数自动被标志，以支持可重入性。

API 文件中的大多数函数都能被设置为可重入函数。组件 API 源文件中的注释表示不建议使用相应函数。

如何将.cyre* *文件添加到项目中：*

1. 右键点击 Workspace Explorer 中的项目，并选择 **Add > New Item**。
2. 在 New Item（新项）对话框中，选择“Keil Reentrancy File”并点击 **OK**。



可在代码编辑器中浏览*.cyre 文件。文件名依赖于项目名，所以想要更改文件名，必须先更改项目名（类似于 DWR 文件）。

如何输入可重入函数：

1. 在每一行中输入一个函数名，例如：

```
ADC_Start
PWM_Start
```

2. 完成后，请保存*.cyre 文件。

将用户应用代码设置为可重入：

要想支持自定义应用代码的可重入性，需要将 *cytypes.h* 文件中的“CYREENTRANT” #define 用于函数原型和函数定义。对于 Kei 工具链，它将参考“reentrant”关键字，而对于其他工具链，则没有关键字。这样您能够在多个工具链和多个器件架构中正确执行各函数。

原有的函数：

```
void Foo(void);
```

更改后的函数：

```
void Foo(void) CYREENTRANT;
```

使自定义组件 API 支持可重入性：

当创建一个可能需要支持可重入性的自定义组件时，请使用“ReentrantKeil”编译表达式来声明函数。“.h”文件中的函数声明和“.c”文件中的函数定义必须包含该表达式。这允许该函数的运行方式与 PSoC Creator 随附的其他组件的运行方式相同。默认情况下，它是标准的函数；但，如果您将函数名添加到可重入性文件中，它将被标志为可重入函数。

原有的函数：

```
void `$INSTANCE_NAME`_Foo(void);
```

更改后的函数：

```
void `$INSTANCE_NAME`_Foo(void) `=ReentrantKeil($INSTANCE_NAME . "_Foo")`;
```

确定可重入函数:

Keil 编译器可以帮助在编译期间确定需要标志为可重入的函数。当将优化等级设置为 2 或更高，并且编译过程已经完成时，Keil 编译器将针对每个被同时调用但未标记为可重入的函数生成一个编译警报信息。

仅在 Keil 编译器为某个函数分配 RAM 空间并需要并发调用该函数时，您才需要将它标志为可重入函数。它依赖于函数参数的数量和类型、局部变量的用法以及函数计算的复杂性。Keil 编译器的警报信息用于确定需要标志为可重入的函数。下面显示的是 Keil 链接器的示例输出：

```
Warning: L15 MULTIPLE CALL TO FUNCTION NAME: _MYFUNC/MAIN CALLER1: ?C_C51STARTUP
CALLER2: ISR_1_INTERRUPT/ISR_1
```

在该示例中，*main.c* 文件中的 `MyFunc` 函数被两个不同的并发执行流程调用。第一个调用程序名称为`?C_C51STARTUP`，它是来自 `main()`函数的主执行流。第二个调用程序是 *isr_1.c* 文件中的 `ISR_1` 中断。

另外，请参考：

- Keil 编译器

7 将设计移植到第三方 IDE 内

IDE Export Wizard（IDE 移植向导）对话框支持在第三方开发环境中开发应用固件。一旦移植完成后，您可以在所选环境中编写、调试和测试固件。

移植过程支持生成新的项目，并更新现有的项目。通过更新选项，您能够在最后一刻对硬件进行更改，快速更新您的项目。

如何打开该对话框：

1. 正常开发您的 PSoC Creator 设计。
2. 完成后，点击 **Project** 菜单，并选择 **Export to IDE...**

执行移植操作：

根据所选的器件，您可以将设计移植到下列第三方 IDE 中。选择合适的选项，并参阅相关内容，以了解更多信息。

注意：所有使用了第三方编程器的器件（使用 MP3 的 µVision 除外），都需要导出到相应 IDE 的项目需要一个任意的 System Editor **Debug Select** 值（而不是 GPIO）。如果具有 **Debug Select** 设置为 GPIO 的项目被导出，那么该项目只能进行一次编程。通过第三方程序进行编程后进行尝试会失败。这里是标准 ARM 采集序列的限制内容，指的是没意识到赛普拉斯器件所使用的特殊采集序列。

- 将设计移植到 Eclipse IDE 内
- 将设计移植到 IAR IDE 内
- 将设计移植到 Keil µVision IDE 内

注意：

- 菜单项仅适用于受移植过程支持的工具链。
- 如果未编译项目，将提示您对其进行编译。
- 如果项目已过时，将提示您重新对其进行编译。

将设计移植到 Keil µVision IDE 内

将设计移植到 Keil µVision IDE 内

使用 PSoC Creator 开发 PSoC 的硬件设计。然后，使用 PSoC Creator 或 Keil 的 µVision IDE 开发固件。要想使用 µVision IDE，您必须在 PSoC Creator 中编译设计，然后使用 Export to IDE（移植到 IDE）功能。

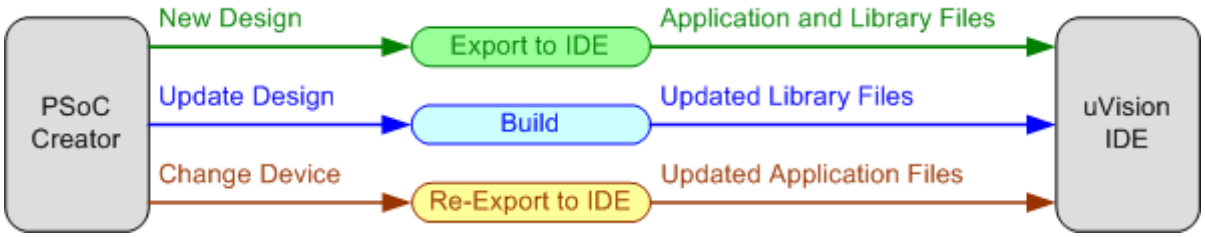
注意：PSoC Creator 支持 Keil µVision 中的 Advanced System Viewer（高级系统视图）。请确保您的 µVision 版本支持 CMSIS-SVD。

注意：编译 µVision 的项目时，所有输出文件都被保存在项目目录中的 UV4Build 子目录下。

PSoC 3 流程

对于 PSoC 3，编译和移植过程会在 µVision 中创建两个项目：一个包含设计 API 和源文件的库项目和一个包含启动文件和固件模板文件的应用项目。

下面框图显示的是概况的工作流程，它描述了如何创建新的 PSoC Creator 设计，并通过移植过程生成 µVision 的项目和文件。它也显示了更新 PSoC Creator 设计的工作流程。更多有关项目和文件的信息，请参阅主要的 IDE 移植文件/项目。



下面的内容描述了每个流程：

- 移植一个新的 PSoC Creator 设计
- 更新一个 PSoC Creator 设计
- 更改 PSoC 器件

  注意：并不支持下列操作：创建和移植 PSoC 3 项目，更改器件为 PSoC 4/PSoC 5LP，然后重新移植。如果您需要这些操作，重新移植前必须手动删除 µVision 项目（*.uvproj）。

PSoC 4/PRoC BLE/PSoC 5 流程

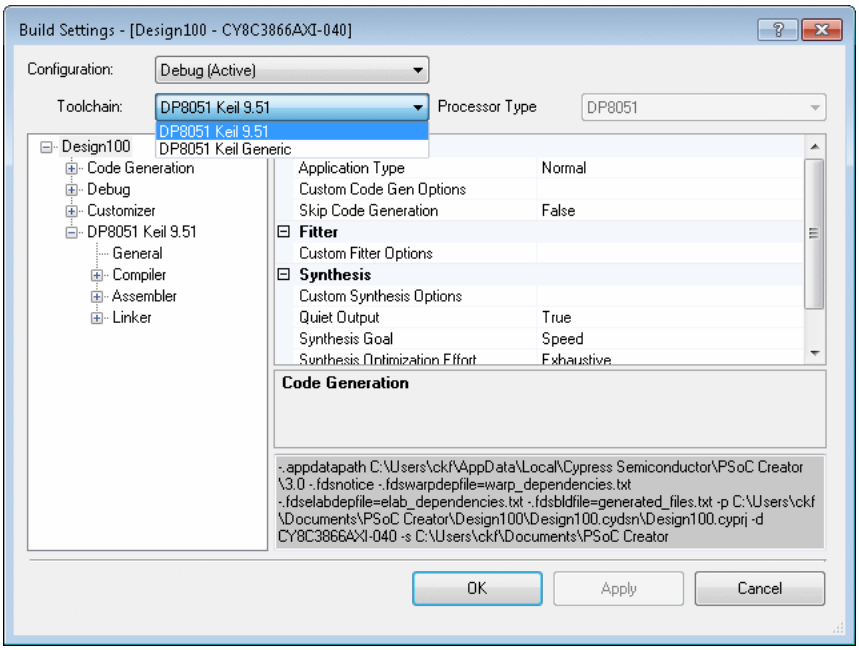
对于 4/PRoC BLE/PSoC 5LP，新项目的编译和移植过程与更新项目的过程一样。

注意：并不支持下列操作：创建和移植 PSoC 4/PRoC BLE/PSoC 5LP 项目，更改器件为 PSoC 3，然后重新移植。如果您需要这些操作，重新移植前必须手动删除 µVision 项目（*.uvproj）。

移植一个新的 PSoC Creator 设计：

最初流程是创建 PSoC Creator 中的设计，并将设计移植到 µVision IDE 内。

1. 在 PSoC Creator 中手动创建您的设计。
2. 对于 **PSoC 3**，打开编译设置对话框，并为您的项目选择相应的工具链。更多信息，请参阅 PSoC Creator 工具链设置。



3. 使用 **Project > Export to IDE** 菜单选项打开 IDE Export Wizard 对话框。

   注意：PSoC Creator 不支持在 µVision 中开发的基于 PSoC 3 的 Bootloader/Bootloadable 应用。PSoC 4/PRoC BLE/PSoC 5LP 的 Bootloader/Bootloadable 项目被所有 IDE 支持。

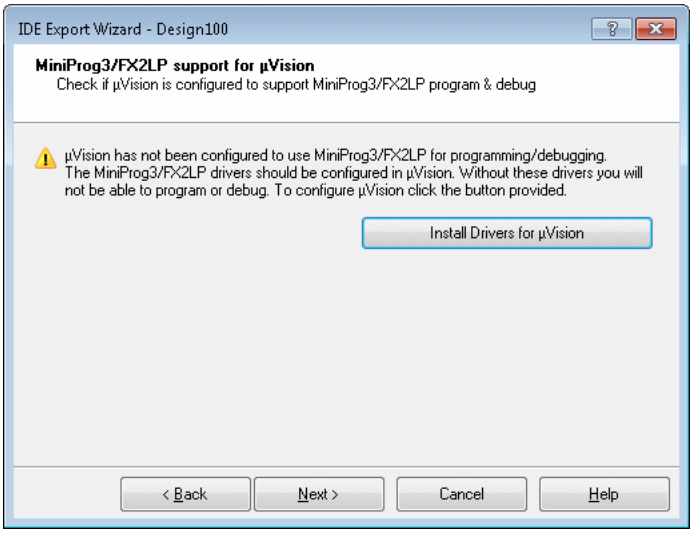
4. 如果您未编译设计，此时将出现一条提示您进行编译的信息。

5. 点击 **Yes**，进行编译。如果点击了 **No** 按钮，移植过程被撤销。

   成功编译后，将打开 IDE Export Wizard 对话框。

6. 如果未选择，请选择 µVision 项。单击 **Next >**按键，进行后续操作。

   PSoC Creator 检查 MiniProg3/FX2LP 驱动器是否注册了 µVision。如果尚未注册，MiniProg3/FX2LP 仅支持打开 µVision。但是，如果 PSoC Creator 找到支持 MiniProg3/FX2LP 性能的 µVision 版本，它将跳过这个步骤。转到 第 7 步。



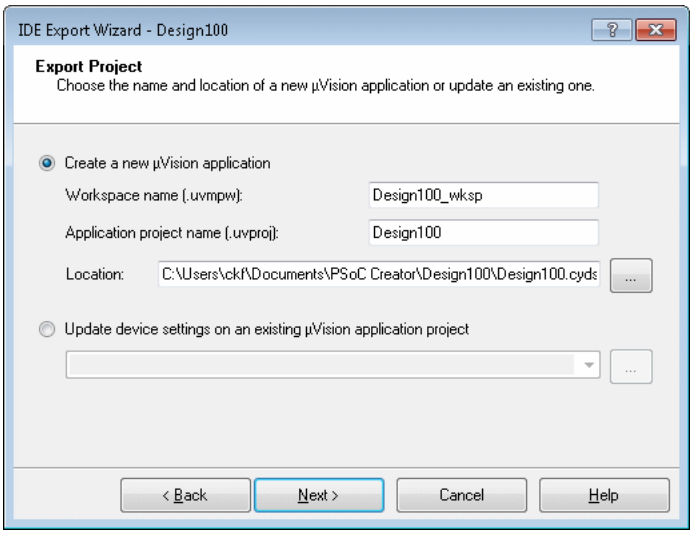
注意：PSoC Creator 仅在 µVision 的第一次安装中检查 MiniProg3/FX2LP 驱动器是否正确注册。如果您的电脑上存在多个 µVision 的复制版本，自动检测过程可能不正确。如果您确定驱动器已经注册过，可以跳过向导中的这一步。

7. 如果您需要注册驱动器，请点击 **Install Drivers for µVision**（安装 µVision 驱动器）并按照安装向导中的指示进行操作。只需执行一次即可。更多选项信息，请参阅注册 MiniProg3/FX2LP 驱动器。

8. 如果您要注册其它 µVision 驱动器，请转到 PSoC Creator 的**工具**菜单，然后选择 **Install drivers for µVision**，启动安装向导。

9. 完成这一步后，请点击 **Next >**，进行后面的步骤。

   对于 PSoC 3，将转到 Export Project（移植项目）步骤。



10. 选择 Create a new µVision application（创建新的 µVision 应用）选项。

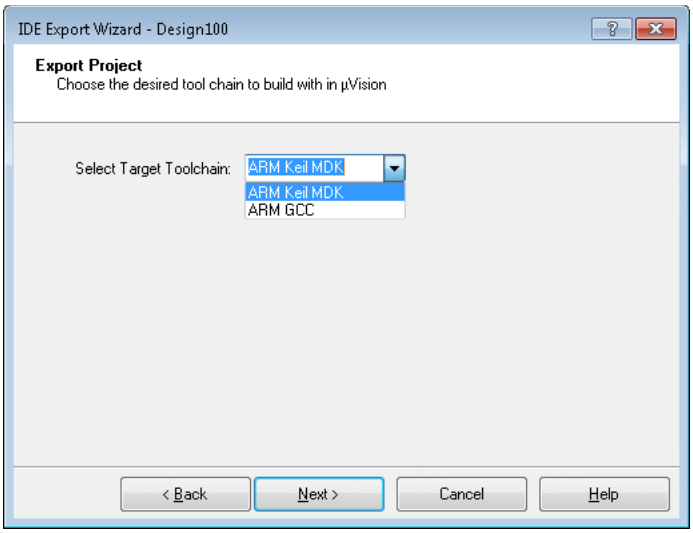
    第一次移植设计时，需要使用该项，用以生成新的 µVision 应用项目。如果您需要更新已移植的设计，请参阅更新 µVision 项目。

    - 可选修改 µVision 工作区和项目的名称。默认情况下，他们的名称以被移植的 PSoC Creator 项目的名称为依据。
    - 您也可以指定保存移植项目的位置。

      注意：向导将不会覆盖现有的项目或工作区文件。如果您想替换掉现有的 µVision 项目，必须先通过 Windows Explorer 删除它。

11. 继续单击 **Next >**按键。

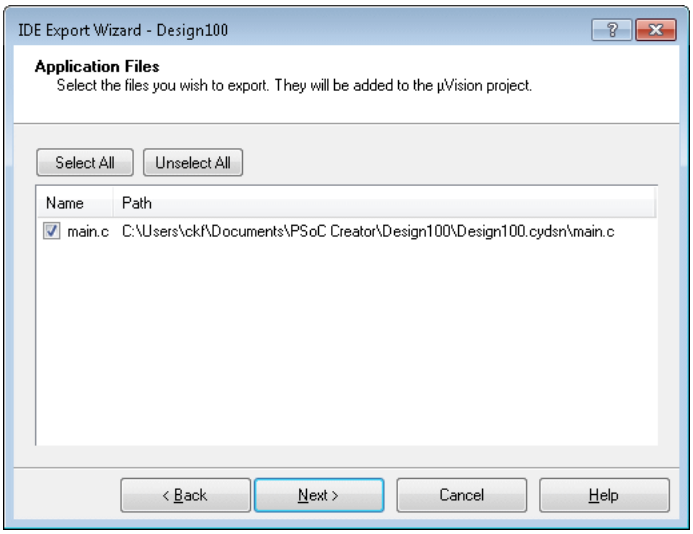
    对于 PSoC 4/PRoC BLE 和 PSoC 5LP，将转到 Select Toolchain（选择工具链）这一步。



12. 选择相应的工具链，并点击 **Next >**，进行后面的步骤。

    注意：如果您选择了 GCC 工具链，请参阅 µVision 中的 GCC 设置。

    对于所有器件，将打开 Application Files（应用文件）步骤。

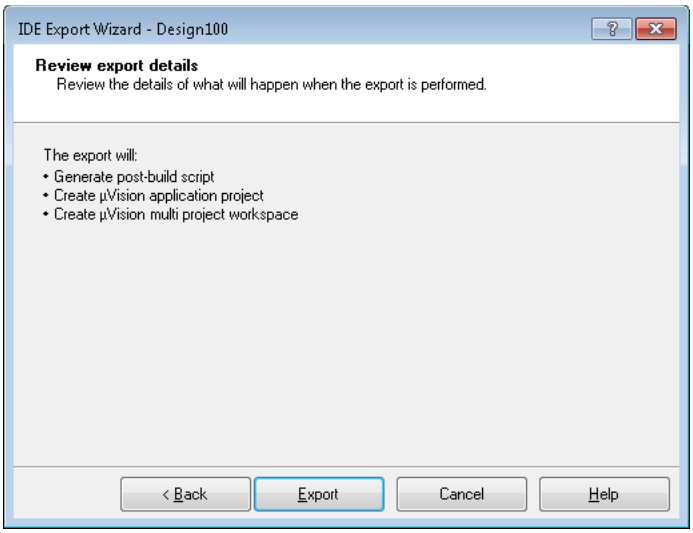


- 选择想要添加到新 µVision 应用项目的文件。

  注意：向导在移植期间不会复制这些文件。它只会将现有文件的参考添加到 µVision 应用项目中。
- 要想开始使用空白的 µVision 应用项目，请单击 **Unselect All**（取消全选）按键。
- 一旦完成第一步移植，那么必须在 µVision 环境中编译设置并管理该文件。例如，如果您想添加一个新的源文件，请在 µVision 中选择 **File > New**。

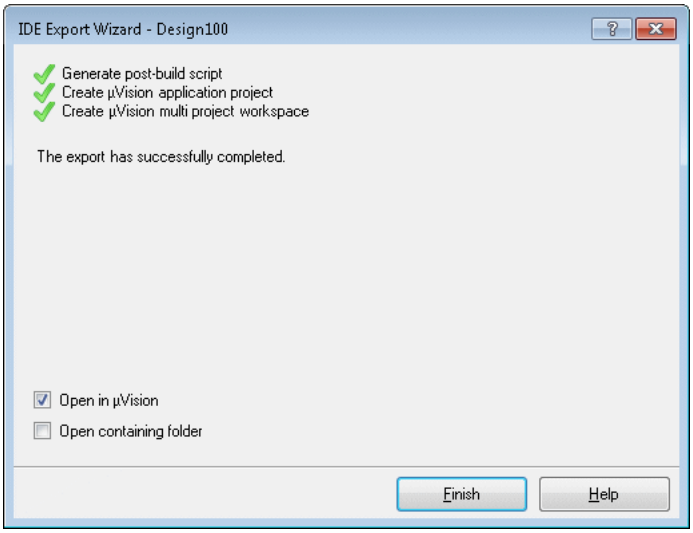
13. 单击 **Next >**按键，进行后续操作。

    此时，将打开“Review export details”（检查移植文件的详情）对话框。



14. 检查移植文件并点击 **Export**。

    最后一个对话框将打开，并显示移植过程已完成。



    移植过程完成时，可以选择执行下面的操作：

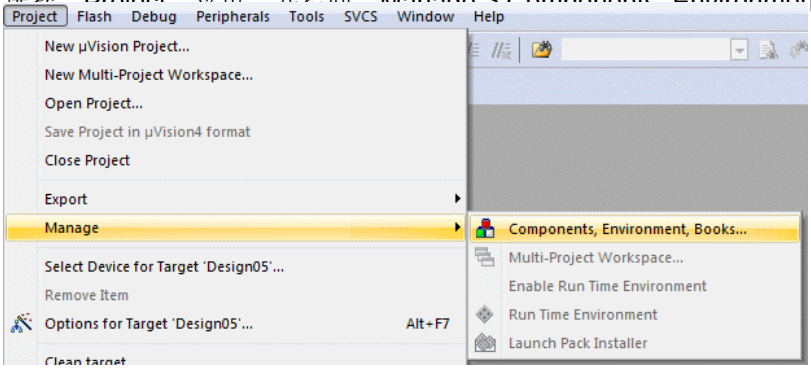
    - 在 µVision 中打开项目
    - 打开包含项目文件的文件夹

15. 请点击 **Finish** 按键关闭向导。

µVision 中的 GCC 设置

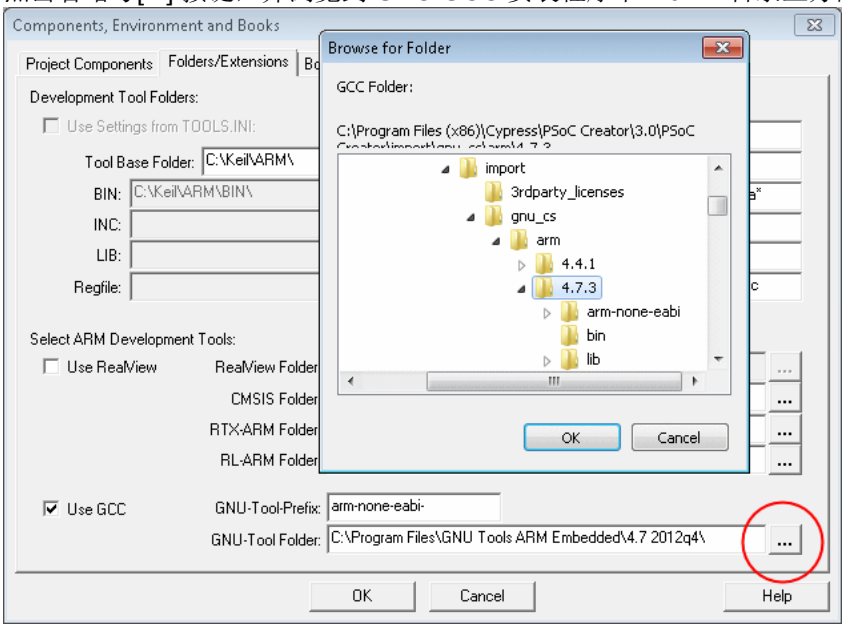
对于使用 GCC 工具链的项目，打开项目后必须在 µVision 中设置一些选项。

1. 选择“Project”菜单，并指向“Manage > Components, Environments, Books...”



2. 在“Components, Environment, and Books”对话框上，选择 **Folders/Extensions** 选项卡，并确保已经选中了 **Use GCC** 复选框。

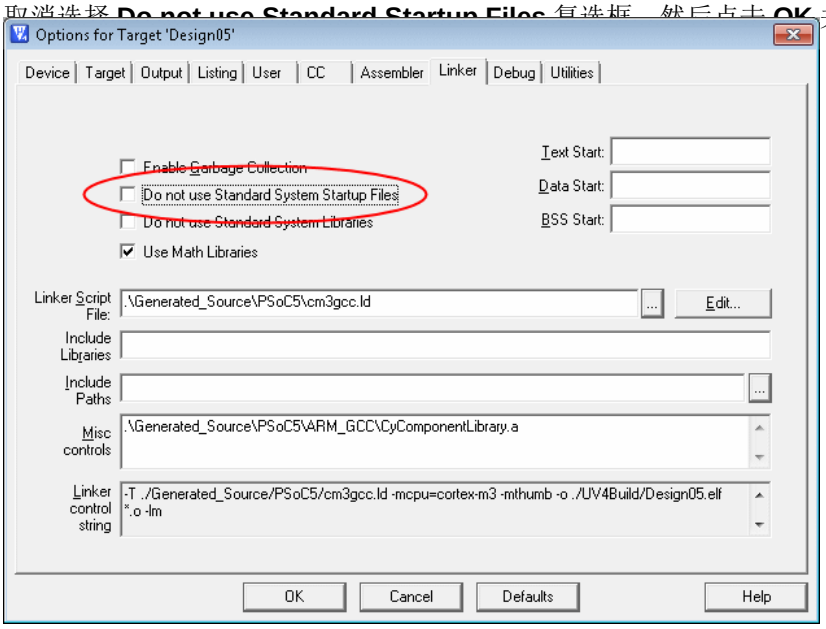
3. 点击省略号**[...]** 按键，并浏览到 GNU GCC 安装程序中“bin”目录上方的目录。



4. 点击 **OK** 关闭“Browse”（浏览）对话框，然后再点击 **OK** 关闭“Components, Environment, and Books”对话框。

5. 右键单击项目，并依次选择 **Options** > **Linker** 选项卡。

6. 取消选择 **Do not use Standard Startup Files** 复选框，然后点击 **OK** 关闭该对话框。



另外，请参考：

- 主要的 IDE 移植文件/项目
- 在 µVision 中打开项目
- 更新 µVision 项目
- 使用 µVision 对闪存进行编程/调试
- PSoC Creator 工具链设置
- 寄存 MiniProg3/FX2LP 驱动程序
- 移植的其他注意事项

MDK 版本 5.01

如果您正在使用的是 Keil 微控制器开发套件（MDK）版本 v5.01 或更新的版本，请注意：它使用了一个新器件和软件包系统。这时，这个新系统并不包含赛普拉斯器件。为了使用赛普拉斯的器件，您需要下载并安装各个随增的器件包：

http://www2.keil.com/mdk5/legacy

更多有关安装的信息和帮助，请参考：

http://www.keil.com/support/man/docs/license/license_sul_install.htm

主要的 IDE 移植文件/项目

移植到 IDE 的性能会生成一些重要的文件。这些信息包括：

路径、文件名称	用途	详细信息
<DesignName>.cydsn/ <DesignName>_<Arch>lib.uvproj （仅适用于 PSoC 3）	µVision 库项目。它是一个静态库文件，允许调用 PSoC Creator 组件 API。	该文件对应于 PSoC Creator 设计项目中“Generated Source”（生成的源文件）文件夹中的内容。 该文件是在与相应的 PSoC Creator 设计项目相同的目录中构建的。不能移动或重命名它。 如果需要，可通过 PSoC Creator 编译过程生成该文件。构建该文件时，会复制 PSoC Creator 的编译设置。 如果已存在该文件，则 PSoC Creator 编译过程会更新文件列表，但不会影响其他设置。这样能够确保在 PSoC 设计发生变化时，固件开发者能获得已更新了的组件 API。
<DesignName>.cydsn/ <DesignName>_<Arch>.uvopt	µVision 库的选项文件。µVision 使用它存储一些设置，如断点位置。	该文件与库项目文件同时生成。 构建该文件后，PSoC Creator 不能再更改它。
<DesignName>.cydsn/ Generated_Source/<Arch>/ post_link.bat	项目构建后的脚本文件。构建应用项目后， µVision 将运行该文件。	可将该文件作为所有 PSoC Creator 构建过程的一部分使用。 它为最终的 HEX 文件添加重要的信息。 进行检查以确保 PSoC Creator 和 µVision 应用项目之间的选定器件、SRAM/闪存地址和大小等设置是同步的。
<User_Selected_Path>/ <DesignName>.uvproj	µVision 应用项目。该项目连接到库项目，以创建一个所设计的最终 HEX 文件。	只有通过 PSoC Creator 才能创建/更新该项目，可以将它作为“Export to IDE”（移植到 IDE）向导的一部分。可以给它命名，或将其保存到任意合理的位置。 构建该文件时，文件列表、编译设置以及器件的重要信息（如器件名称、SRAM/闪存地址和大小）需要与 PSoC Creator 同步。 更新该文件时，PSoC Creator 仅更新器件的重要信息（如器件名称、SRAM/闪存地址和大小）和‘包括’路径中 *Generated_Source/<Arch>*文件夹的参考。 更新该文件时，PSoC Creator 会保留所有其他设置（包括在‘包括’路径中找到的呵呵其他目录）。 可以随时编辑 µVision 中“Firmware”（固件）组的 API 文件。 “Start”（开始）组中的 API 文件将是由 PSoC Creator 生成的，并在 PSoC Creator 构建过程中被覆盖。这些文件包含了启动器件的重要代码。 通过 DebugLib（调试库）和 ReleaseLib（施放库）中的文件， 可确

		保能够准确地链接各个所生成的库项目输出文件，以进行调试和释放。
<User_Selected_Path>/<DesignName>.uvopt	µVision 应用选项文件。µVision 使用它存储一些设置，如断点位置。	该文件与应用项目文件同时生成。 构建该文件后，PSoC Creator 不能再更改它。
<User_Selected_Path>/postlink.bat	项目构建后的脚本文件。构建应用项目后， µVision 将运行该文件。	该文件与应用项目文件同时生成。 构建该文件后，PSoC Creator 不能再更改它。 一个简单的包装器脚本会调用 PSoC Creator *Generated_Source/<Arch>*目录中的 post_link.bat 脚本。
<User_Selected_Path>/<DesignName>_wksp.uvmpw （仅适用于 PSoC 3）	µVision 多应用工作区它是一个 µVision 多应用项目文件，包含了库项目和应用项目的参考。	该文件与应用项目文件同时生成。 构建该文件后，PSoC Creator 不能再更改它。
<DesignName>.cydsn/<DesignName>_<Arch>.sfr （仅适用于 PSoC 4/PRoC BLE 和 PSoC 5LP）	µVision 系统浏览器文件。该文件包含了组件寄存器的详细信息。µVision 使用它进行调试外设寄存器。这是 Keil µVision 所需的文件格式，以能够通过其系统浏览器调试外设寄存器。	通过转换符合 ARM CMSIS 标准的 SVD 文件，可获得该文件（SVD 文件是由 PSoC Creator 使用 ARM-MDK 所提供的 SVDConv.exe 工具来生成的）。 该文件与 µVision 库文件同时生成；一旦设计的寄存器映射发生变化，该文件也会得到更新。

另外，请参考：

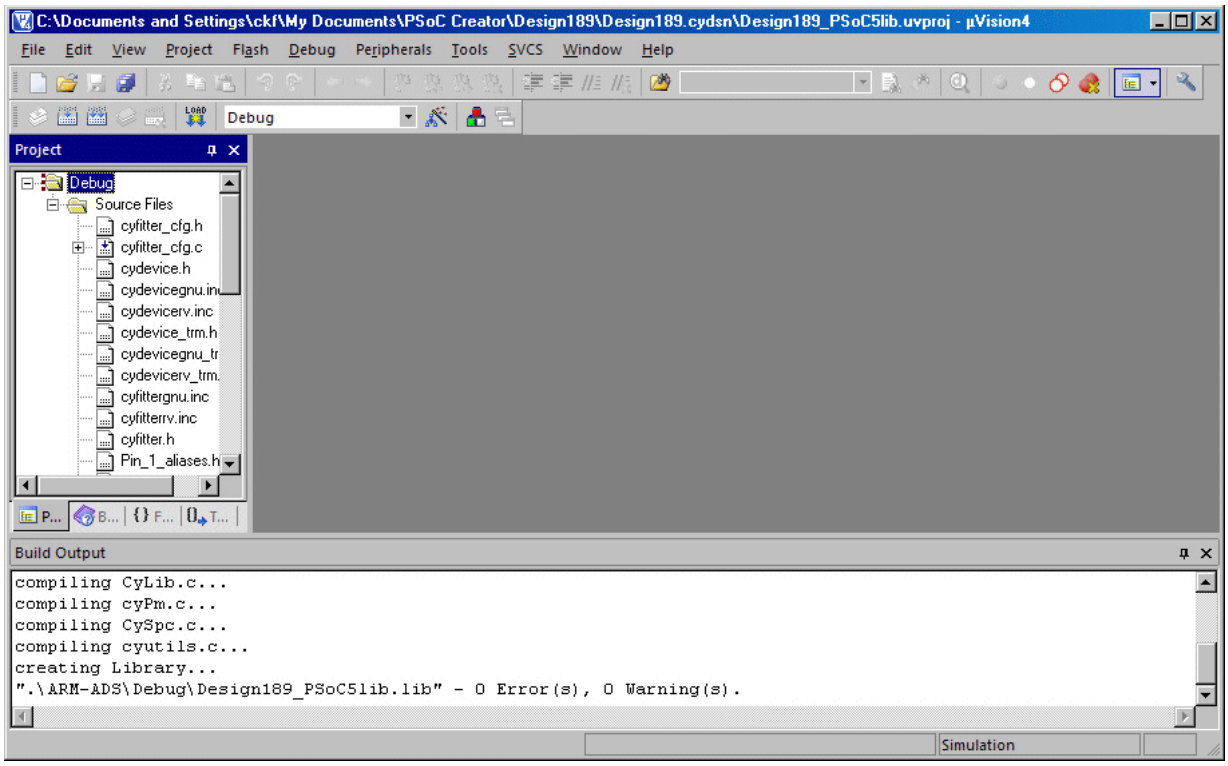
- 将设计移植到 Keil µVision IDE 内
- 更新 µVision 项目

在 µVision 中打开项目

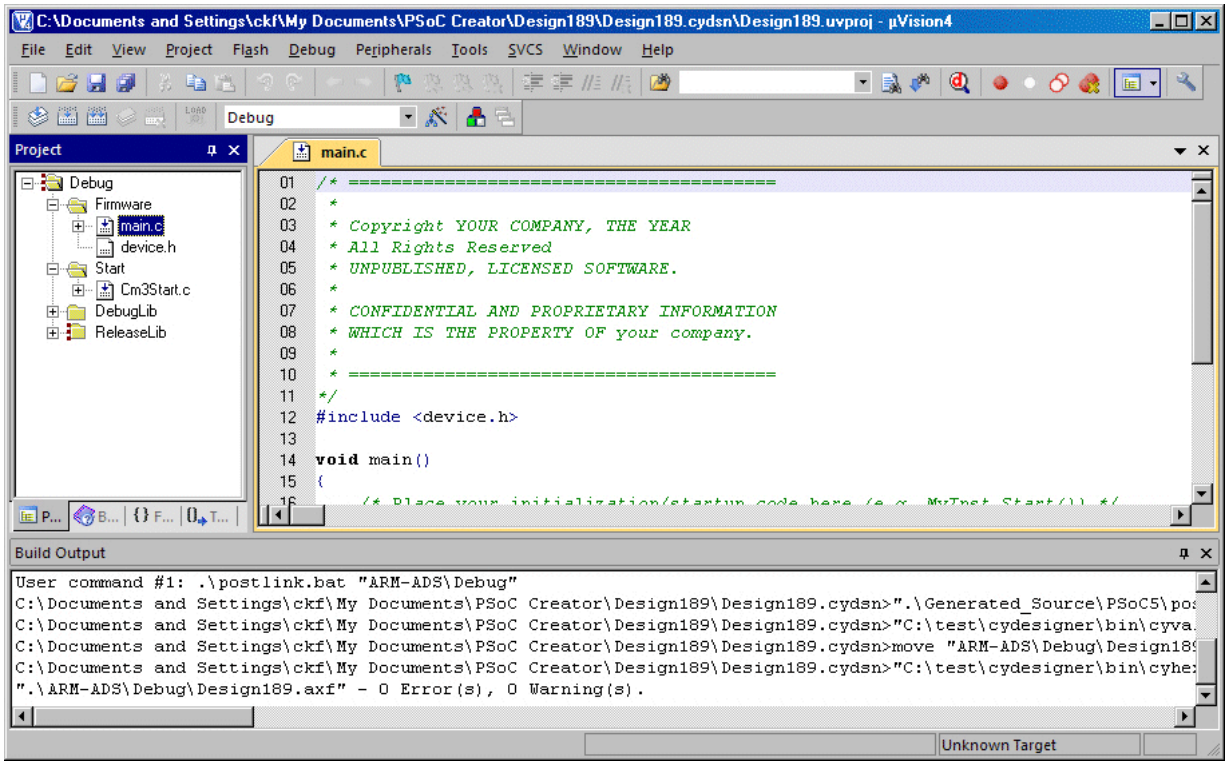
正如主要的 IDE 移植文件/项目部分中所述，导出过程中会生成一些 µVision 文件和项目。您可以分别打开库项目和应用项目，也可以打开包含这两者的多项目工作区。

注意：必须更新库项目，以便能成功构建应用。如果因无法找到库而不能构建应用项目，很可能是因为需要在 µVision 环境中构建该项目。请打开库项目或使用多项目工作区中的批量构建选项解决这种情况。

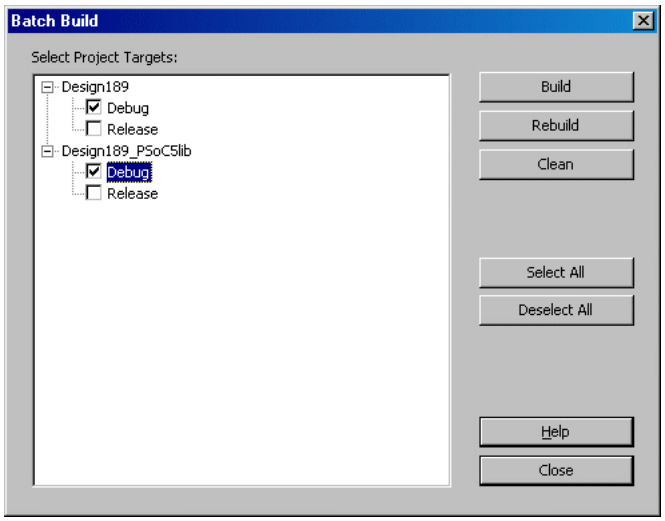
下面是在 µVision 环境中打开和构建库项目的截图。



下面是在 µVision 环境中打开和构建应用项目的快照。



可以在 µVision 工作区*<ProjectName.uvmpw>*中同时打开这两个项目， 并使用下图所示的 **Project > Batch Build** 构建它们。



调试和释放构建

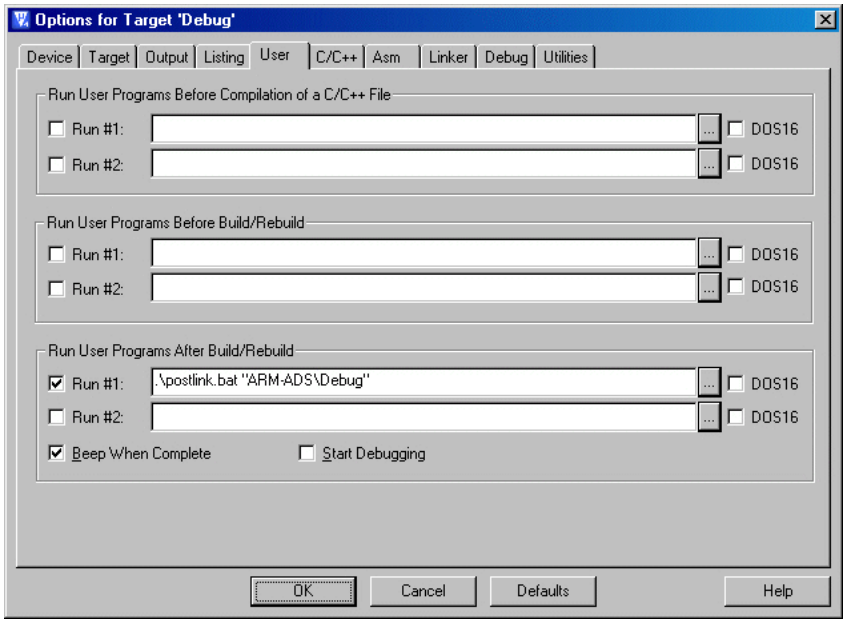
在 PSoC Creator 中，可执行构建项目的配置共有两种，即：调试和释放。项目构建的调试是出于调试目的，而构建释放则用于为用户提供项目。两种配置之间的主要区别正是构建构成中所使用的选项。

在 µVision 环境中，通过创造两个对象（一个用于“调试”，另一个用于“释放”）来处理这种情况。µVision 提供了一个下拉菜单用于切换这两个对象。

设置 µVision 中的选项

下面是在 µVision 环境中用于指定构建后的指令的应用项目的截图。

注意：PSoC 4/PRoC BLE/PSoC 5LP 的移植过程只提供基本的选项。所有器件的项目移植完成后，用户能够进行全面的设置。



另外，请参考：

- 将设计移植到 Keil µVision IDE 内
- 主要的 IDE 移植文件/项目

设置 ULink2/ULink Pro 和 Segger J-Link 调试器探针

通过这些步骤可设置 ULink2/ULink Pro 调试器探针。

要在 µVision 4.72a 中完全编程和调试 PSoC 4/PRoC BLE/PSoC 5LP 器件，需要下载 µVision 的“add-on”（插件）。

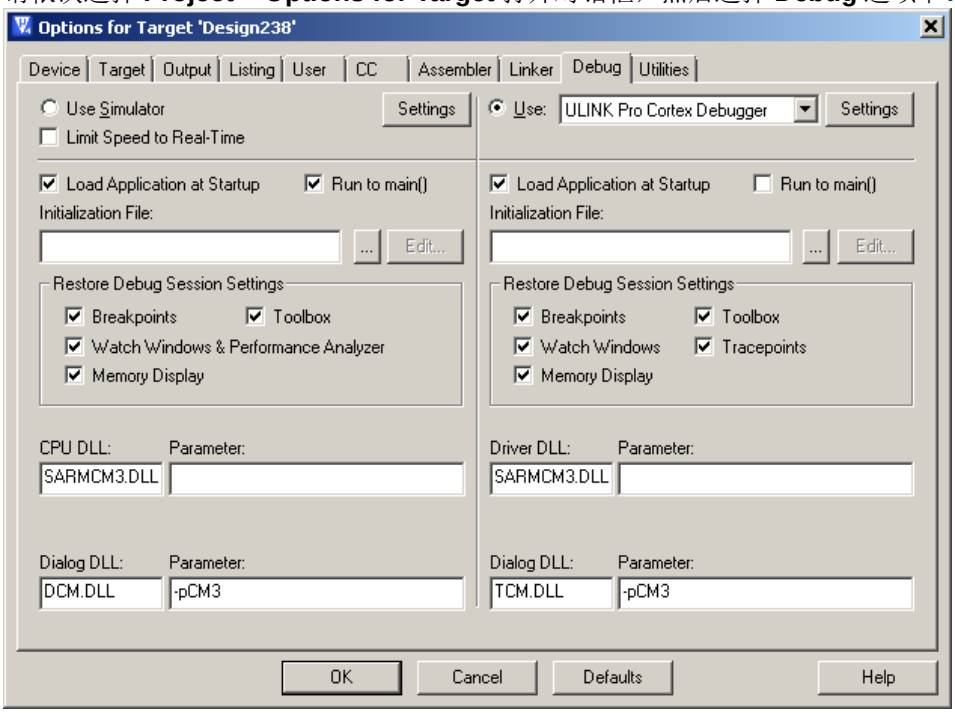
注意：如果您正在使用比 4.72a 更新的版本，则不需要再下载插件。

可以在下面网站下载插件：http://www.cypress.com/go/creator/uvisionimportdownload。

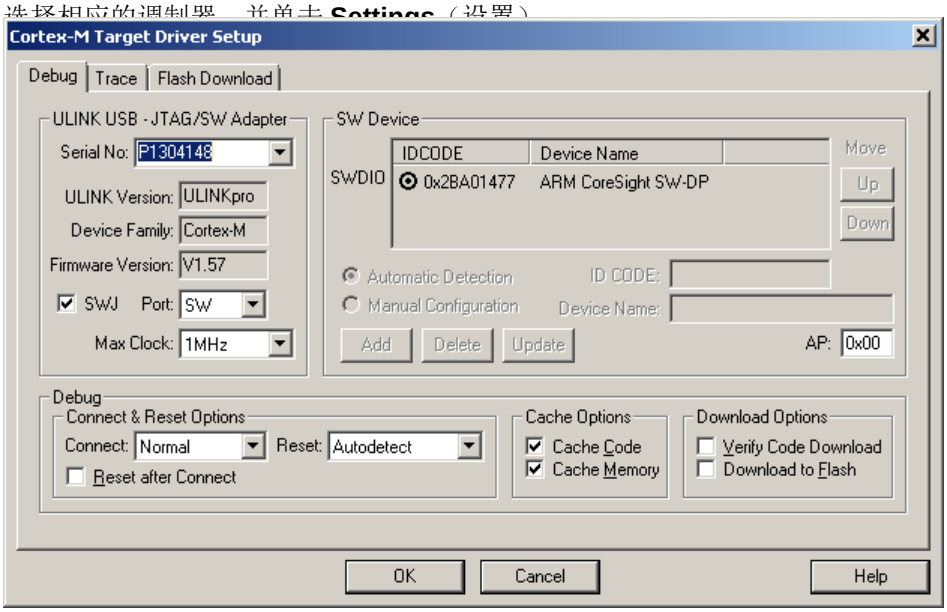
下载插件后，请解压归档文件，运行可执行文件。

然后运行 µVision 并遵循下列过程来安装它：

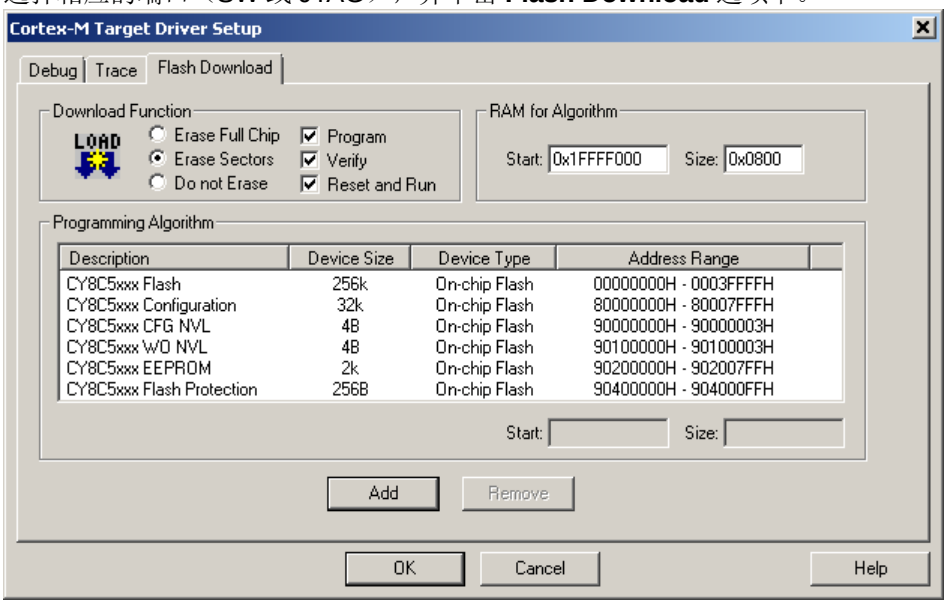
1. 构建 PSoC 5LP 器件的设计并将它移植到 µVision 内前，请跳转到项目的设计范围资源系统页，并取消选择“Store Configuration Data in ECC Memory”（将配置数据存储在 ECC 存储器中）参数。“Storing configuration data in ECC memory”是 PSoC Creator 项目的默认设置，但将设计移植到 uVision 内时必须进行更改。
2. 请依次选择 **Project > Options for Target** 打开对话框，然后选择 **Debug** 选项卡。



3. 选择相应的调制器，并单击 **Settings**（设置）。



4. 选择相应的端口（SW 或 JTAG），并单击 **Flash Download** 选项卡。



5. 选择 **Erase Full Chip**，并设置“RAM for Algorithm”（算法的 RAM）的大小，如下表所示。

Keil ULink 和 Segger J-Link 调试器中 PSoC 4 和 PSoC 5LP “算法的 RAM”值

PSoC器件系列	算法的RAM		编程算法 *
	启动（Start）	大小	
PSoC 4100/4200	0x20000300	0x0D00	CY8C42xx 1 MACRO（32kB）闪存
PSoC 4100 BLE/PSoC 4200 BLE、PRoC BLE	0x20000200	0x3E00	CY8C42xx/41xx-BL、CYBL10xx（128）kB 闪存
PSoC 4000	0x20000200	0x0600	CY8C40xx（16kB） 闪存

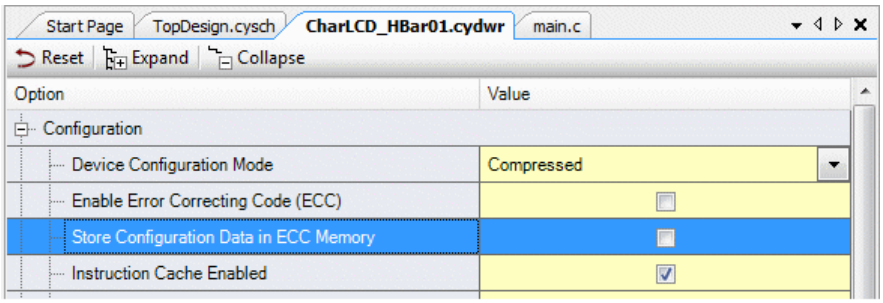
PSoC器件系列	算法的RAM		编程算法 *
	启动（Start）	大小	
PSoC 5LP	0x20000400	0x0C00	CY8C5xxxx 闪存 CY8C5xxxx 配置 CY8C5xxxx CFG NVL CY8C5xxxx WO NVL CY8C5xxxx EEPROM CY8C5xxxx 闪存保护

***** 编程算法应存储在 *C:\Keil\ARM\flash*.FLM* 中。如果 uVision 中没有包含任何赛普拉斯闪存加载程序，请从 PSoC Programmer 文件夹（*C:\Program Files (x86)\Cypress\Programmer\ 3rd_Party_Configuration_Files*）中复制它。

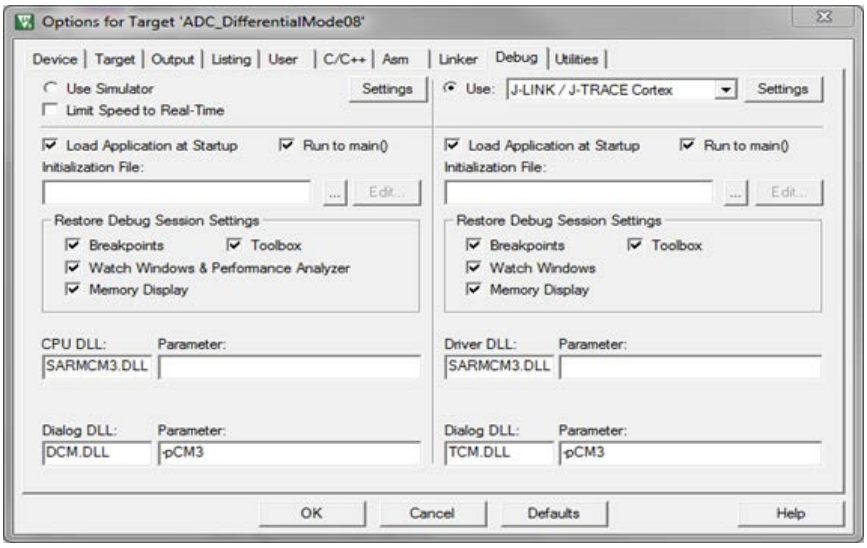
设置 Segger J-Link/J-Trace 调试器

请按照下面流程在 µVision 中设置 Segger J-Link，以编程和调试 PSoC 5LP 器件系列。

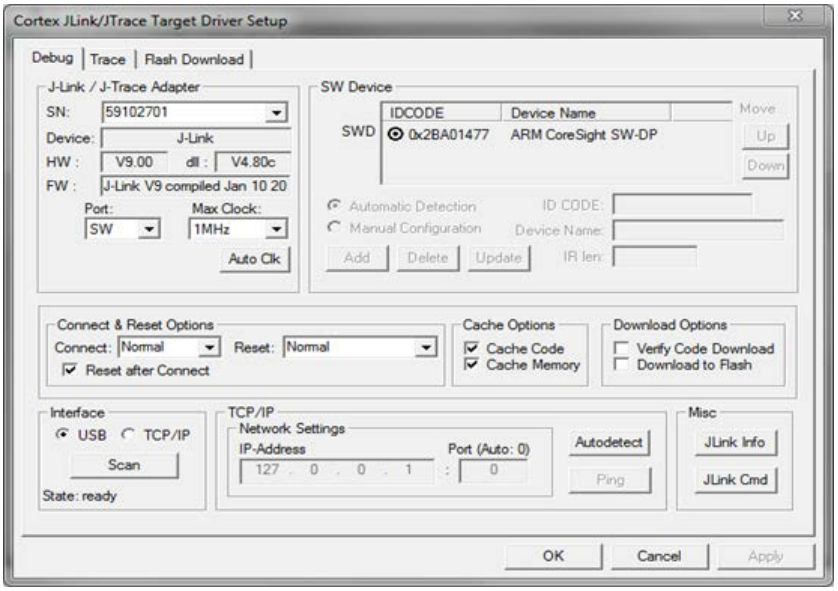
1. 构建 PSoC 5LP 器件的设计并将它移植到 uVision 内之前，请跳转到项目的设计范围资源系统编辑器页，并取消选择“Store Configuration Data in ECC Memory”（将配置数据存储在 ECC 存储器中）参数。“Storing configuration data in ECC memory”是 PSoC Creator 项目的默认设置，但将设计移植到 µVision 内时必须进行更改。



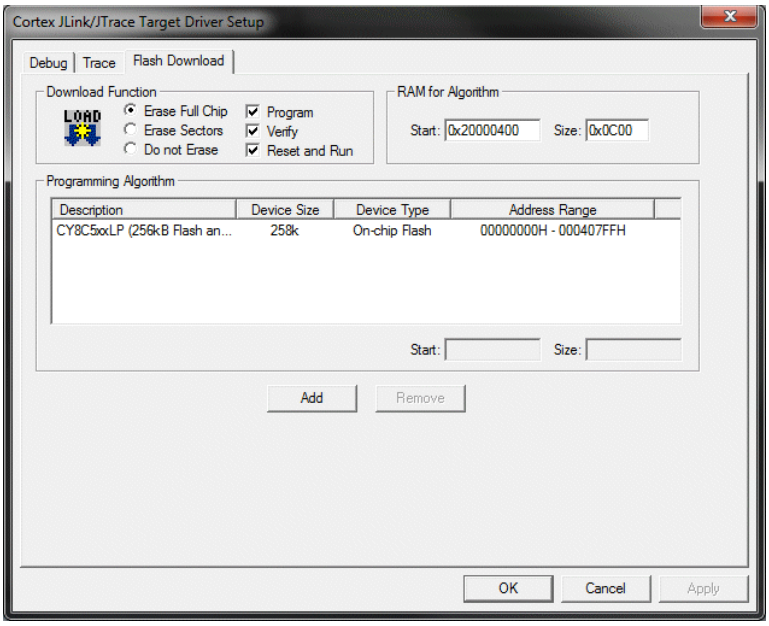
2. 启动 µVision IDE，并依次选择 **Project > Options for Target** 打开对话框，然后选择 **Debug** 选项卡。



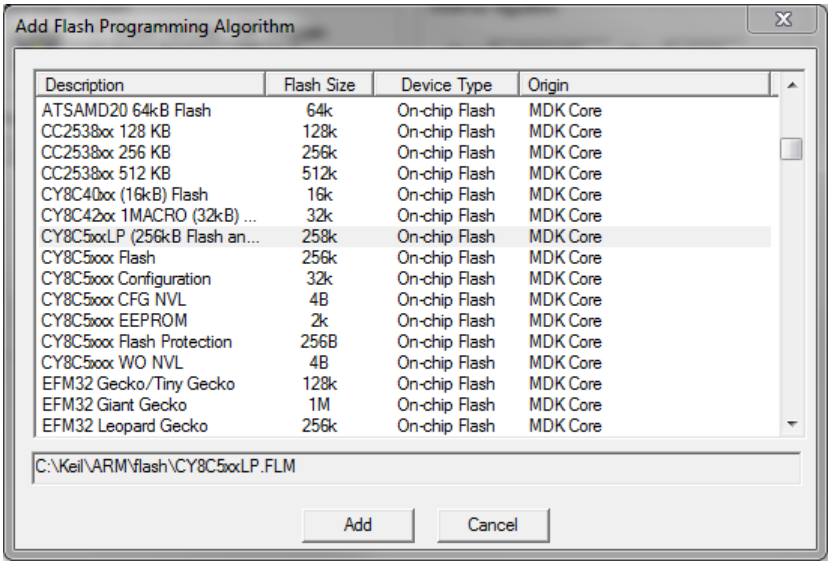
3. 选择相应的调制器，并单击 **Settings** 打开“Driver Setup”（驱动程序设置）对话框。



4. 在 **Debug** 选项卡中，请选择相应的端口（SW 或 JTAG）。然后单击 **Flash Download**（闪存下载）选项卡。



5. 如果在“Programming Algorithm”（编程算法）区域内未显示 CY8C5xxLP，请单击 **Add**（添加）按键。



6. 如果 CY8C5xxLP 未显示在列表中，请在下面的网站上下载 µVision 的插件：
http://www.cypress.com/go/creator/uvisionimportdownload。

通过下列操作，您可以从 PSoC 编程示例项目手动添加闪存加载程序。访问下面路径：

C:\Program Files (x86)\Cypress\Programmer\3rd_Party_Configuration_Files\CY8C5xxLP\Prog_Algorith

将这些文件复制到：

C:\Keil\ARM\Flash

更新 µVision 项目

本部分仅针对 PSoC 3 器件。对于 PSoC 4/PRoC BLE 和 PSoC 5LP，除了代码更改外，设计中进行任何其他更改都需要进行重新移植项目。

在大部分情况下，当更新 PSoC Creator 中的设计（如添加组件或配置引脚）时，只要在 PSoC Creator 重建您的设计即可。µVision 互项目中的文件列表会自动更新。在 µVision 中轻松重建库项目，以便能够调用各个新的组件 API。但是，如果在 PSoC Creator 中选择不同的 PSoC 器件，则需要重新移植设计，以更新 µVision 应用项目。

如何更新库项目（更改 PSoC Creator 设计）：

要想更新库项目：

1. 请在 PSoC Creator 中进行必要的更改，如将某个引脚添加到您的设计中。
2. 在 PSoC Creator 中保存并构建设计，以生成已更改的 µVision 库项目。
3. 打开 µVision 环境中的库项目，并重建它。
4. 大多情况下，您不需要重新移植应用项目。

被更新的内容包括：

- 器件名称（与应用项目相同）
- CPU 信息（与应用项目相同）
- 输出文件名称（<输出名称>关键字）
- 创建库和可执行文件（<CreateLib>和<CreateExecutable>关键字）
- SRAM/闪存的起始地址和空间大小与应用项目相同
- 项目文件中‘源文件’组的文件列表

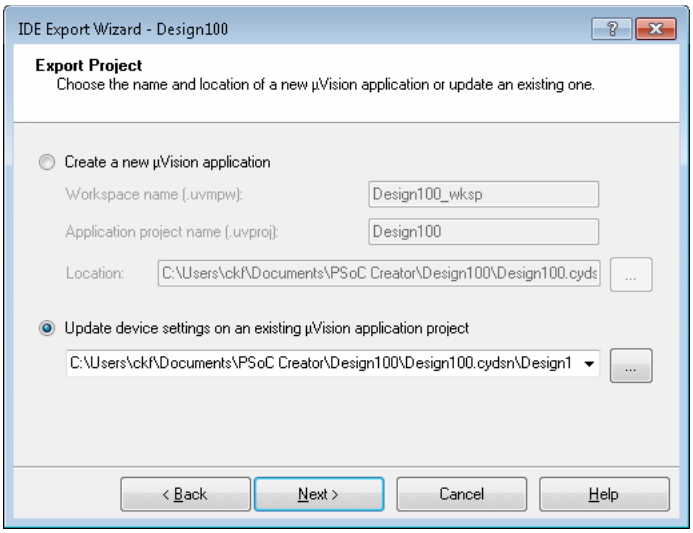
如何更新应用项目（更改 PSoC 器件）：

如果您先前已经移植了设计，并且在 PSoC Creator 中更改了 PSoC 器件，则 µVision 输出窗口会显示一条错误信息清晰地指出问题所在。请使用 IDE 移植向导更新器件的重要信息，用以清除错误。

注意：µVision IDE 环境不允许使用项目构建后的脚本报告错误。您需要手动检查输出窗口的文本来确定这些错误。

要想更新应用项目，请执行下面的操作：

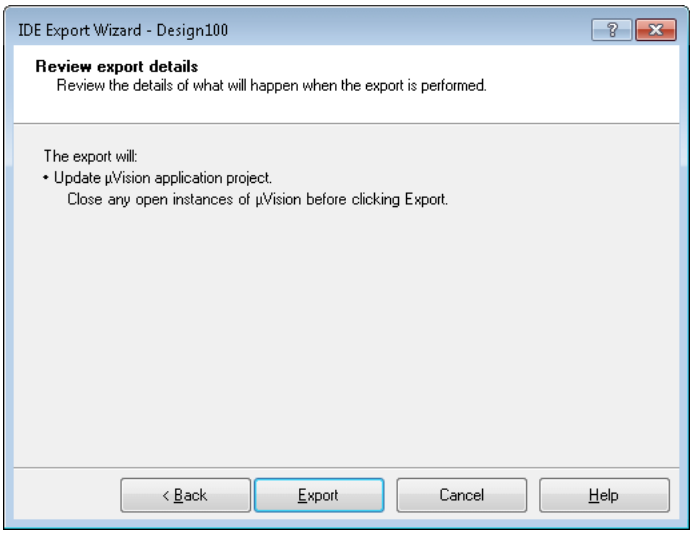
1. 保存并构建 PSoC Creator 中的设计。
2. 然后依次选择 **Project > Export to IDE** 菜单选项来打开 IDE 移植向导对话框。



3. 选中 **Update device settings...**（更新器件设置…）选项，并导航到存储应用项目的位置。

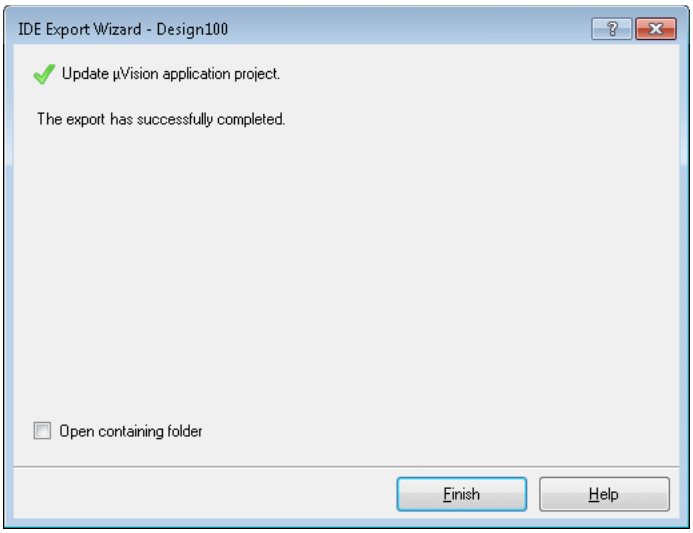
   该选项将更新必要的应用级文件，但**不会**更新任何固件文件。

4. 点击 **Next >**按键。向导将显示应用项目的更改内容。



   注意：如果应用项目是在 µVision IDE 环境中打开的，那么赛普拉斯推荐您从 PSoC Creator 重新移植前请先关闭该项目。

5. 点击 **Export** 按键。向导会显示更改的内容。



您可以勾选选项以打开所包含的文件夹。

6. 点击 **Finish** 按键以完成从 PSoC Creator 的移植过程。然后重新打开 µVision 环境中的项目，并重建它。

   注意：Update device settings...选项只会更改应用项目， 并不会更改主要的 IDE 移植文件/项目中所述的其他任何文件。

被更新的内容包括：

- 器件名称（项目文件中的<器件>关键字）
- CPU 信息（项目文件中的<Cpu>关键字）
- 闪存的起始地址（<片上存储器><IRO>部分中的<起始地址>关键字）
- 闪存的空间大小（<片上存储器><IRO>部分中的<大小>关键字）
- SRAM 的起始地址（<片上存储器><Ocr1>部分中的<起始地址>关键字）
- SRAM 的大小（<片上存储器><Ocr1>部分中的<大小>关键字）
- 指向 PSoC Creator Generated_Source 目录的所有‘包括’路径条目（所有<包括路径>关键字）

另外，请参考：

- 将设计移植到 Keil µVision IDE 内
- 移植文件/项目

使用 MiniProg3 对闪存进行编程/调试

通过使用 MiniProg3 或任何一个 FX2LP 器件（如 FTK-3、FTK-5、DVK-030、DVK-050 等等）（硬件支持），都可以在 µVision 环境中对闪存进行编程/调试。可以在 µVision 中选择驱动程序实现该目的。

MiniProg3/FX2LP 驱动程序会实现 µVision AGDI 接口（高级的常见调试器接口），该接口被直接连接到 Keil µVision 编程器/调制器。

注意：对于 PSoC 5LP，赛普拉斯提供 Keil ULink2 调试器探针的闪存加载程序。通过闪存加载程序，Keil µVision 能够使用 Keil ULink2 来对 PSoC 5LP 器件进行编程和调试。但是，闪存加载程序并不支持该器件的非易失性锁存器（NVL），因此您需要使用 MiniProg3 来对该器件进行编程，以便设置相应值。

对闪存进行编程时，支持下面各项功能：

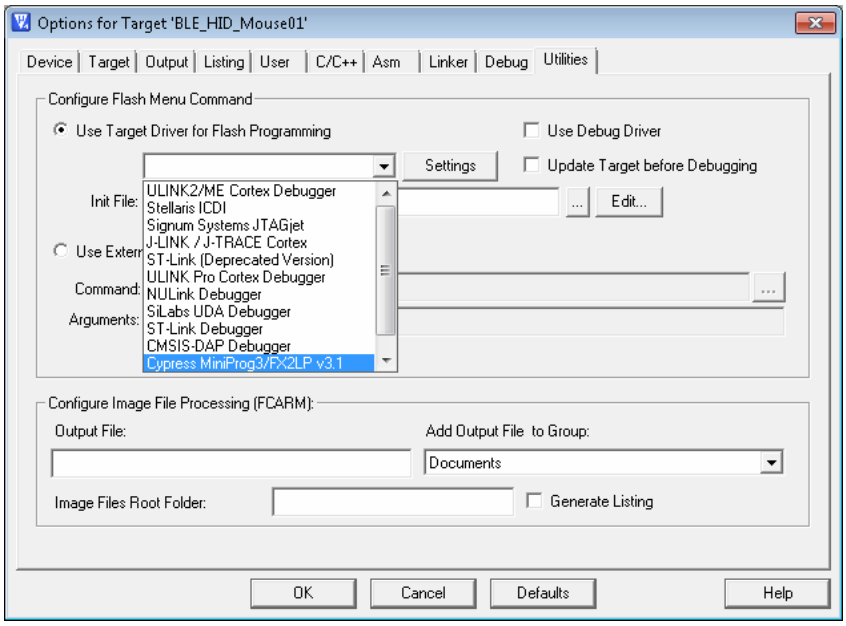
- 全部檫除
- 下载闪存，NVL 及保护
- 验证闪存。NVL 及保护
- 编程后复位器件

调试时，支持以下功能：

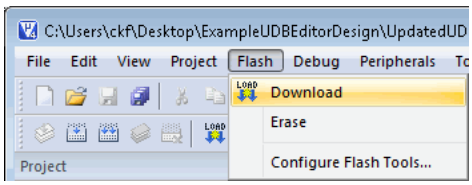
- 启动/停止/复位/单步调试/单步跳出/运行到光标
- 9800 插入/移除断点
- 使能/禁用断点
- 禁用所有断点
- 停止（Kill）所有断点
- 读取/写入寄存器
- 读取/写入存储器

在 µVision 环境中对闪存进行编程时，开始下载闪存前，请从下面显示的下拉菜单中选择“Cypress MiniProg3/FX2LP vX.Y”驱动程序。点击“Settings”（设置）按键可打开一个对话框，然后在点击“LOAD”按键时可以选择执行不同的操作。

- 檫除闪存
- 编程闪存
- 验证闪存
- 复位和运行

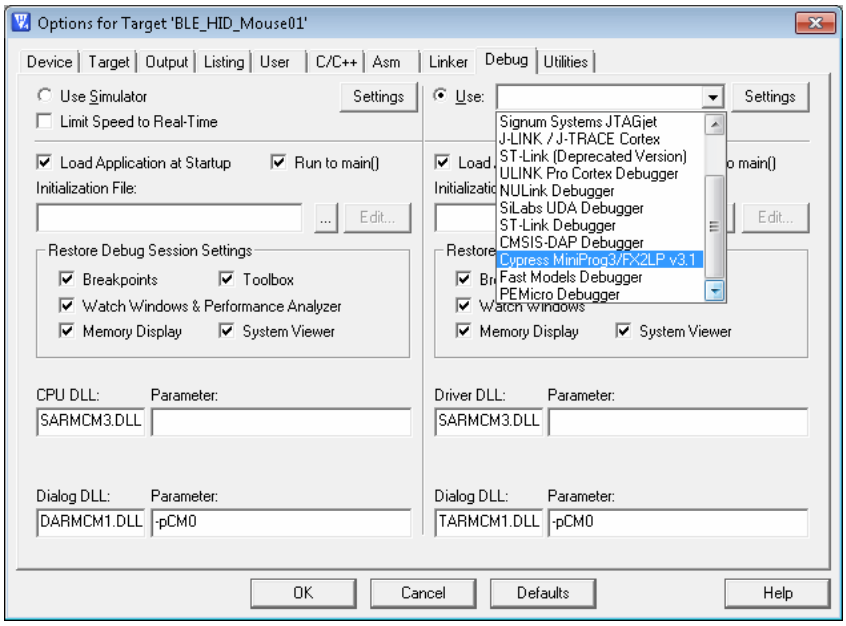


依次选择 **Flash > Download** 菜单选项，以初始化闪存的编程操作，如下图所示。



通过选择 **Debug** 选项卡中的“Cypress Miniprog3/FX2LP vX.Y”驱动程序，可以将 µVision 调试器配置为使用 MiniProg3/FX2LP 调试驱动程序。点击“Settings”按键会打开包含下列设置信息的对话框：

- 协议：SWD/JTAG
- 目标电压
- 时钟速度
- 编程模式：复位/循环供电
- 有效端口：10 引脚/5 引脚



在 **Debug** 菜单中点击“Start/Stop Debug session”（启动/停止调试会话），以初始化调试操作。点击 **Save All**（保存所有）按键，保存 µVision IDE 中的这些调试设置。

请参考 µVision IDE 文档。

另外，请参考：

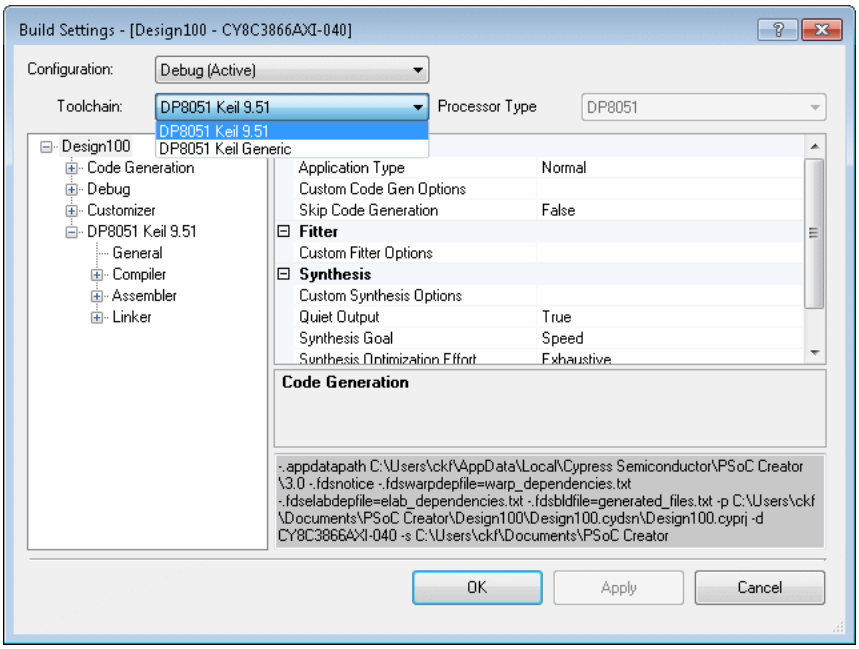
- 将设计移植到 Keil µVision IDE 内

PSoC Creator 工具链设置

如果想在 PSoC Creator 项目中使用 ARM MDK 通用工具链或 DP8051 Keil 通用工具链，您需要指定相应的工具链设置。

1. 对于所有 PSoC Creator 项目，请执行下面的一次性操作：打开 Options 对话框，并导航到 **Project Management > Generic Toolchains**。
2. 指定 ARM MDK 通用工具链（PSoC 5LP）/ DP8051 Keil 通用工具链（PSoC 3）的相应二进制目录。

   这些二进制文件都是作为工具链应用的一部分安装的。
3. 对于每一个 PSoC Creator 项目而言，请打开 Build Settings（编译设置）对话框并选用相应的通用工具链。



另外，请参考：

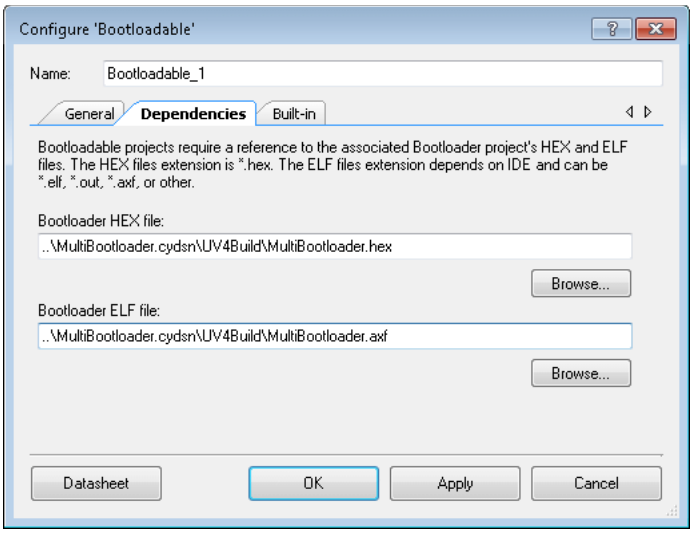
- 将设计移植到 Keil µVision IDE 内
- 项目管理选项
- 编译设置

µVision 多应用 Bootloader 移植支持

本部分介绍的是如何移植 PSoC Creator 多应用 bootloader 项目，以便能在 µVision IDE 环境中使用它。

1. 正常移植多应用 bootloader 项目。请参考将设计移植到 Keil µVision IDE 内部分。
2. 在 µVision 中编译 bootloader 项目。

   在 PSoC Creator 中，配置 Bootloadable（可引导加载）项目的 Bootloadable 组件，以指向 µVision 输出目录中多应用 bootloader 的 hex/axf 文件。

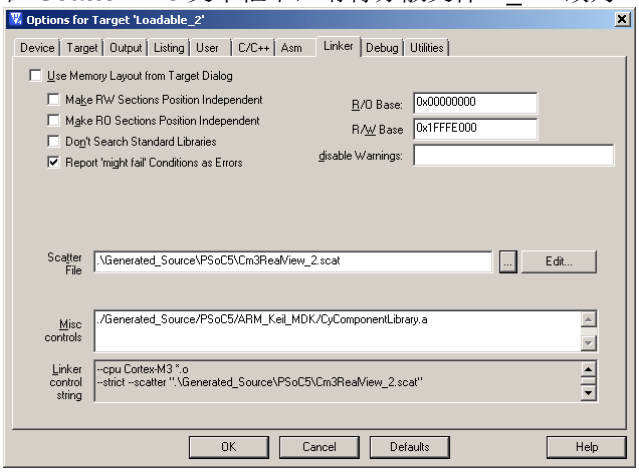


3. 正常移植 Bootloadable 项目。这样会将一个名称为{ProjectName}_1.uvproj 的项目放置在磁盘上。
4. 在 µVision 中编译(ProjectName}_1.uvproj 项目。
5. 然后手动移植项目并给它起一个新的名称，如下所示：
6. 打开某条指令行并导航到 Bootloadable 项目的‘.cydsn’目录中。
7. 使用 µVision 指令行来移植{ProjectName}.xml 项目。

   该指令如下所示：

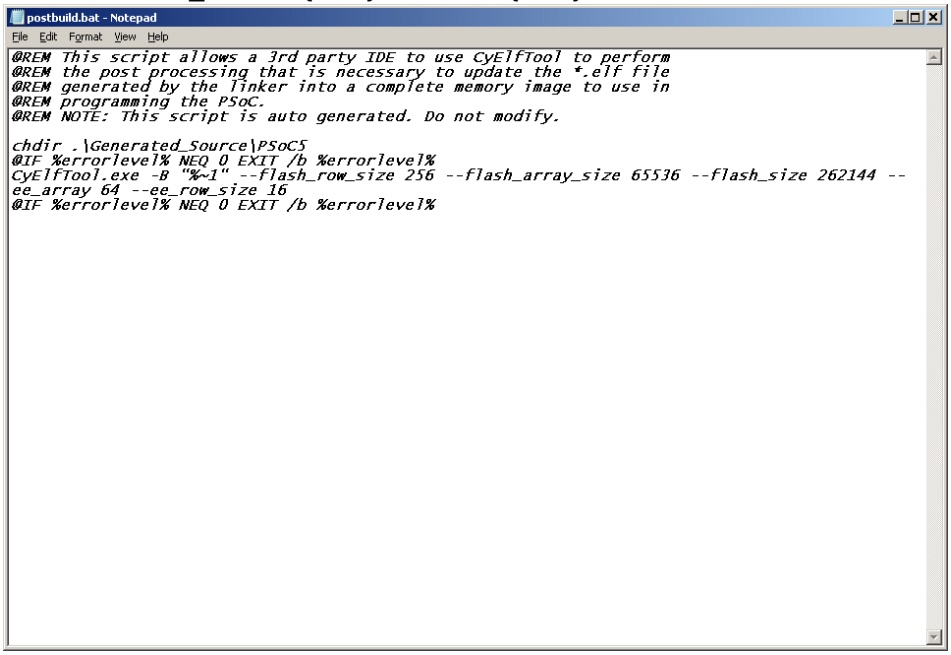
```
C:\Keil\UV4\Uv4.exe {ProjectName}_2.uvproj -t {ProjectName}_2 -i
".\Generated_Source\{ProjectName}.xml" -c
"{PSoCCreatorInstallDir}\cydesigner\bin\CyPSoC.cdb"
```

8. 在 Keil uVision 环境中打开{ProjectName}_2.uvproj。
9. 请依次选择 Project > Options for Target 打开对话框，然后选择 **Linker** 选项卡。
10. 在 **Scatter File** 文本框中，请将分散文件‘_1’改为‘_2’。



11. 编译项目。

12. 然后使用 CyElfTool.exe 工具构建一个包含 bootloader 和两个 Bootloadable 映射文件的组合文件。
13. 打开 Generated_Sourcet\{Arch}目录（其中{Arch}是 PSoC4 或 PSoC5）中找到的 *postbuild.bat*。



14. 复制 `flash_row_size, flash_array_size, flash_size` 指令行参数。如果出现`-ee_array` 和`-ee_row_size` 参数，也需要复制它们。
15. 在 Windows 指令行中，通过 cd 指令进入 Bootloadable 项目的‘.cydsn’目录中。
16. 使用下面指令运行 *CyElfTool.exe*：

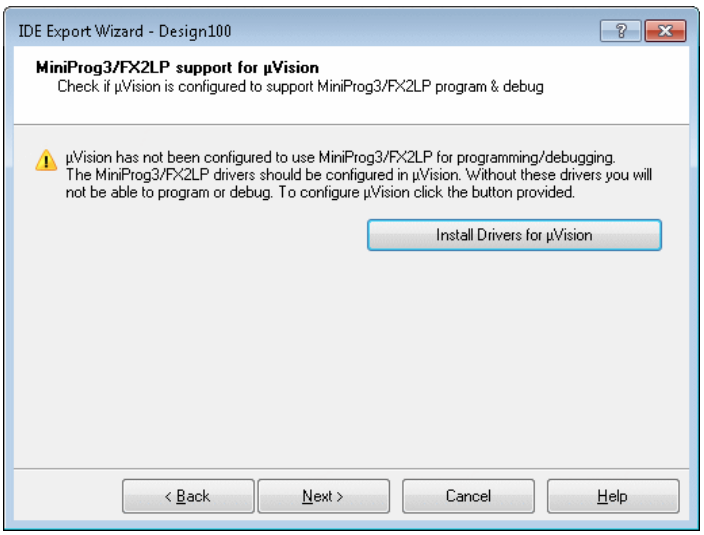
    ```
    \Generated_Source\{Arch}\CyElfTool.exe -M UV4Build\{ProjectName}_1.axf
    UV4Build\{ProjectName}_2.axf UV4Build\{ProjectName}.hex
    ```

    （复制 *postbuild.bat* 中的参数，如：`--flash_size 262144 --flash_row_size 256 --ee_array 64 --ee_row_size16`）
17. 现在，在 UVBuild 目录中存在两个‘.axf’文件和一个 hex 文件。根据要求，您可以编程并调试 Bootloader、Bootloadable 或组合文件。
18. 如果您对某个 Bootloadable 项目进行了更改，请确保也对另一个 Bootloadable 项目进行了同样的更改。
19. 如果您从 PSoC Creator 中重新导出某个 Bootloadable 项目，那么请确保通过上述的手动步骤重新生成第二个 Bootloadable 项目。

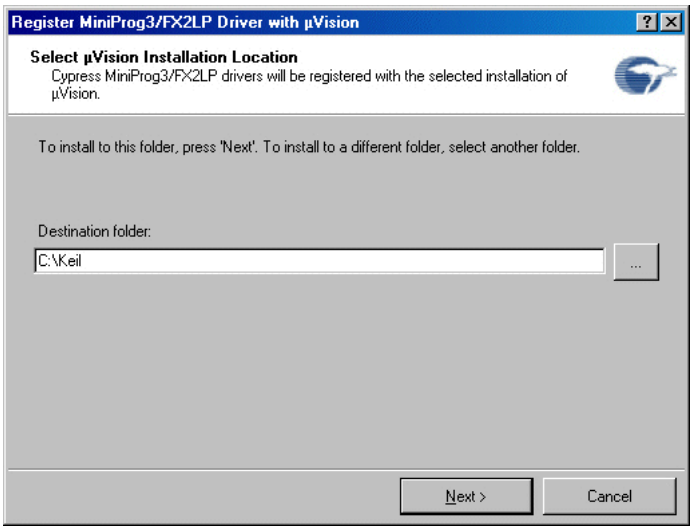
寄存 MiniProg3/FX2LP 驱动程序

通过 MiniProg3/FX2LP 驱动程序，µVision 编程/调试接口可以使用 MiniProg3 或任何基于 FX2LP 的硬件与器件进行通信。该配置过程更新了 µVision *tools.ini* 文件，即将各条目添加到`[C51]`、 `[ARM]`和 `[ARMADS]`部分中，这些条目包含链接到 PSoC Creator 安装目录中 DLL 文件的路径。

1. 第一次运行 IDE 移植向导时，PSoC Creator 会提示安装驱动程序。



2. 请点击 **Install Drivers for µVision** （安装 µVision 的驱动程序）按键，将打开注册窗口。

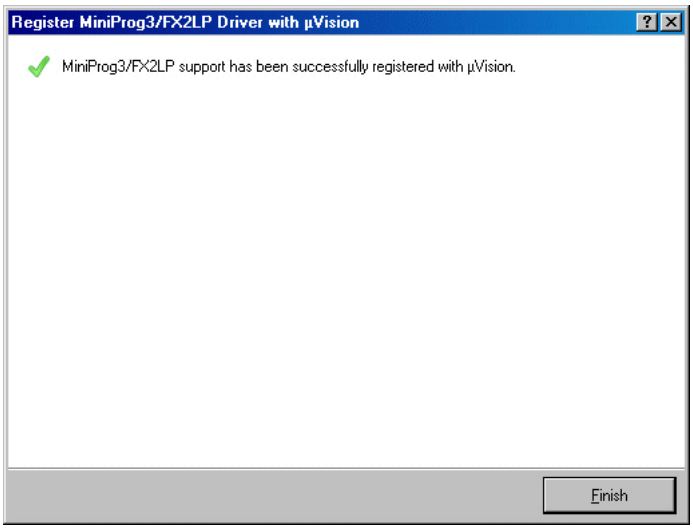


   您同样可以在 PSoC Creator **Tools** 菜单中打开该对话框。

3. 输入路径，或点击**[...]**按键并导航到 Keil 安装程序所在的路径。这是到保存 µVision *tools.ini* 文件的目录的路径。

4. 点击 **Next >**按键。

   确认屏幕将显示，表示驱动程序已成功注册。



注意：如果选定的目录中没有任何 *tools.ini* 文件，将显示一个错误信息。请确保选择的是正确的目录。

5. 完成时，请点击 **Finish** 按键。

另外，请参考：

- 将设计移植到 Keil µVision IDE 内

移植的其他注意事项

本部分重点介绍了一些关键内容：

- 不支持通过使用多个连接到 USB 端口的 MiniProg3/FX2LP 器件来同时进行编程/调试。
- 要确保 MiniProg3/FX2LP 驱动程序成功安装，请验证 *Tools.ini* 文件已经包含了“C51”、“ARM”和“ARMADS”部分中到 MiniProg3 驱动程序的路径。下述示例介绍的是 PSoC Creator 的默认安装的 *Tools.ini* 文件：

```
[ARM]
TDRV12="<INSTALL_DIR>\PSoC
Creator\export\ide\uVision\4.x\driver\cyuvdriver_8051.dll"("Cypress Miniprog3/FX2LP
v<version>")
[ARMADS]
TDRV12="<INSTALL_DIR>\PSoC
Creator\export\ide\uVision\4.x\driver\cyuvdriver_arm.dll"("Cypress Miniprog3/FX2LP
v<version>")
[C51]
TDRV9="<INSTALL_DIR>\PSoC
Creator\export\ide\uVision\4.x\driver\cyuvdriver_arm.dll"("Cypress Miniprog3/FX2LP
v<version>")
```

  “TDRV”后面的数字会因 µVision 的安装不同而存在变化。同样，驱动程序的路径和名称中的版本号也随着 PSoC Creator 的安装版本的不同而不同。

- 如果未使用 MiniProg3/FX2LP 驱动程序的路径更新 *Tools.ini* 文件，则可以使用相应的路径手动更新该文件。
- 首次创建 µVision 应用项目前，在 PSoC Creator 中编写的固件代码将被同步化。一旦创建好 µVision 应用项目，构建设置和文件都被保留在 µVision GUI 中。

- 导致 IDE 特性只支持 PSoC Creator 的下面各种项目：
  - PSoC 3 项目会使用“DP8051 Keil 9.51”或“DP8051 Keil Generic”工具链。
  - PSoC 5LP 项目会使用“ARM MDK Generic”、“GCC 4.7.3”或“GCC Generic”工具链。
  - 所有项目必须为“Normal”（正常）或“Bootloadable”（可引导加载）状态。通过查看“Build Settings -> Code Generation”（构建设置 -> 代码生成）并检查“Application Type”（应用类型）设置，可以验证您项目的类型。

    注意：支持普通的 Bootloadable 项目，但针对多应用 Bootloader 的 Bootloadable 项目并不受支持。
- µVision 应用项目中“Start”（启动）组的代码是由 PSoC Creator 自动重新生成的。
- µVision 构建后的步骤无法按影响最终错误/警报计数的方式来报告错误。您需要检查 µVision 输出窗口以确保构建后的脚本不报告任何错误。
- 从 UNC 格式路径（\\server\path\to\script.bat）运行时，不能稳定地操作 Windows BAT 文件。尤其是 BAT 文件处理器可能会报告它不能将当前的工作目录设置为\\server\path。通过使用 Windows “Map Network Drive”（映射网络驱动）功能将一个驱动字母分配给共享的服务器目录，您可以解决该问题。

另外，请参考：

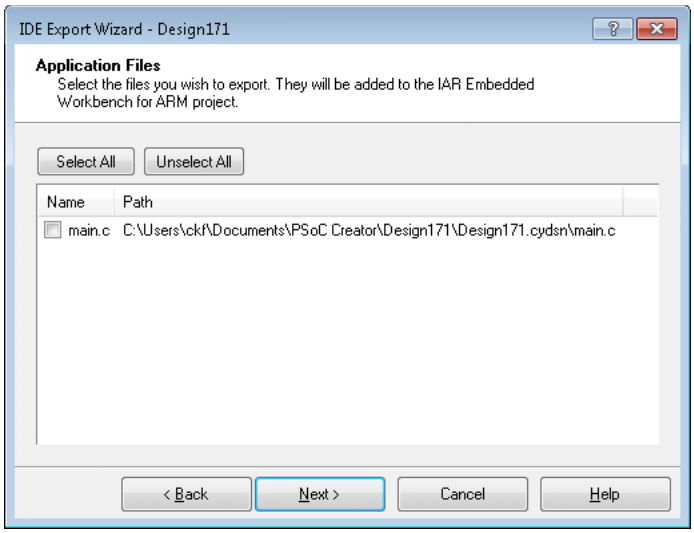
- 将设计移植到 Keil µVision IDE 内

将设计移植到 IAR IDE 内

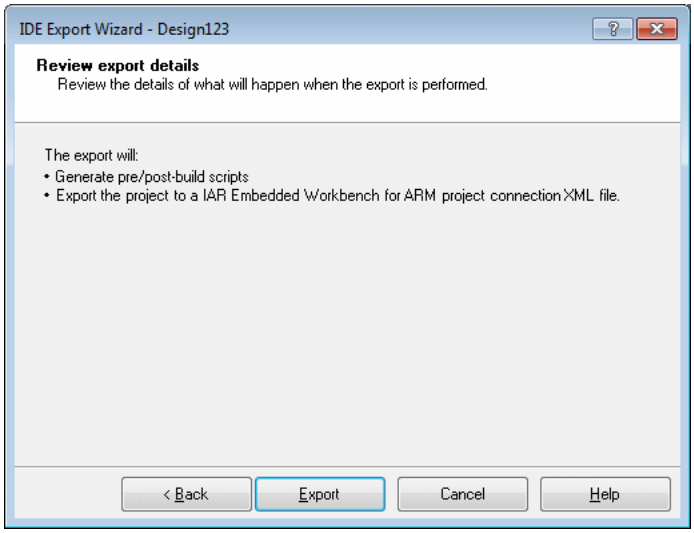
将设计移植到 IAR IDE 内

要想将某个 PSoC Creator 设计移植到 IAR IDE 环境中，请选择“Export IDE”对话框中的“IAR”项。

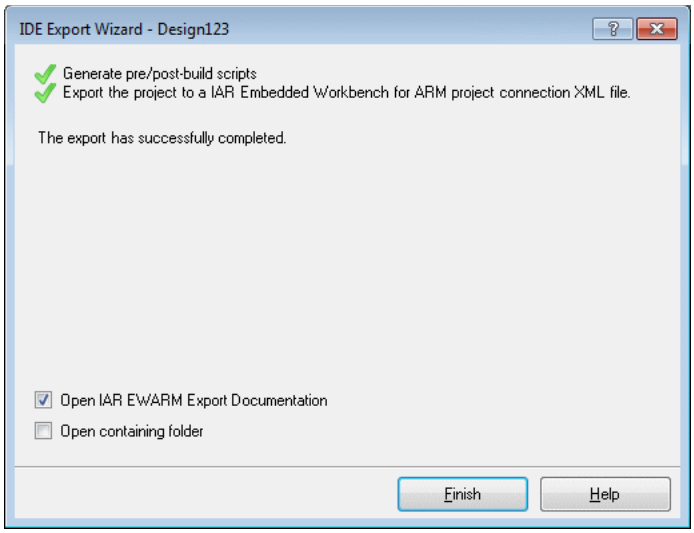
点击 **Next >**按键转到 Application Files（应用文件）页面，以选择未生成代码的条目，从而导出到项目：



点击 **Next >**转到向导的下一页以显示将进行的操作（比如：移植选定项目的 XML 文件）。



点击 **Export** 按键。向导会显示移植操作是否成功，并提供有关用户当前需要进行事项的扩展文档连接。



移植过程完成时，可以选择执行下面的操作：

- 打开 IAR EWARM 移植文档。请参考设置 IAR 项目的流程部分。
- 打开包含项目文件的文件夹

请点击 **Finish** 按键关团向导。

另外，请参考：

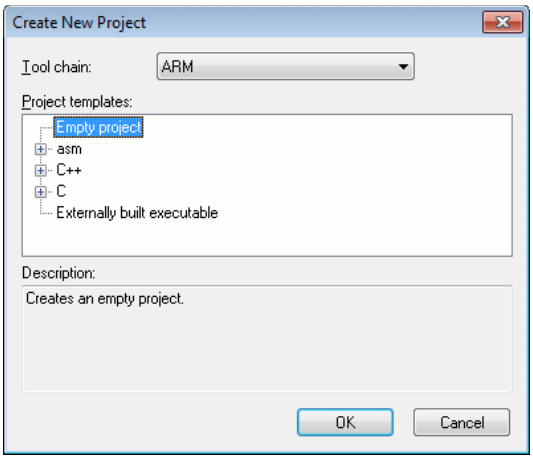
- 将设计移植到第三方 IDE 内
- 设置 IAR 项目的流程

设置 IAR 项目的流程

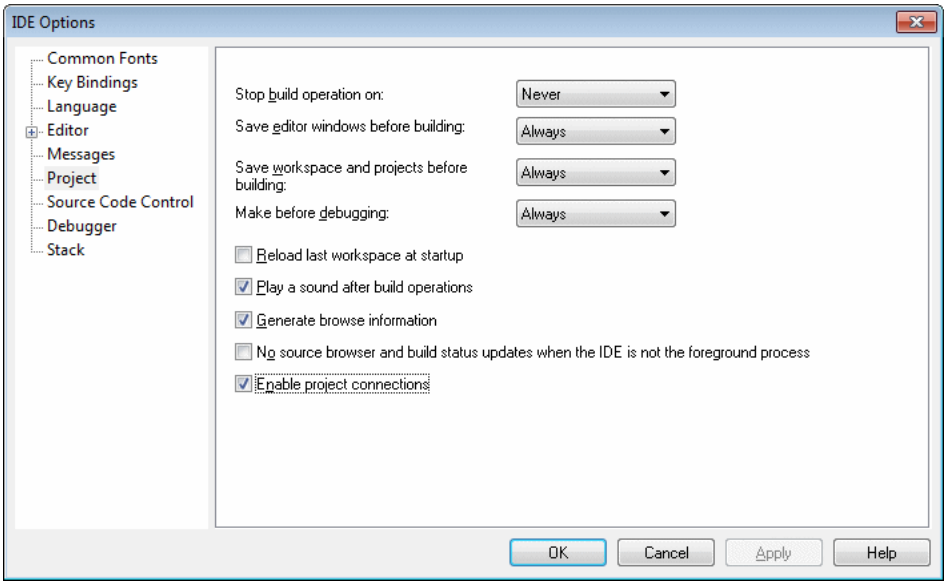
移植 PSoC Creator 项目，使之能在 IAR IDE 环境中使用后，您需要遵循一系列步骤来设置项目。这是因为 IAR 不提供移植一些需要用于设置项目的重要信息（如：链接器文件、工具链指令行设置）的方法，或将第三方的库添加到项目链接器的其他库列表中的方法。

具体的流程如下：

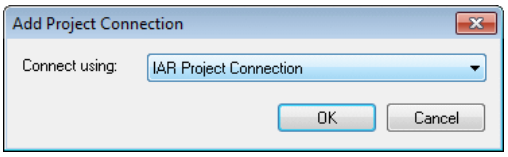
1. 使用移植向导将 PSoC Creator 项目移植到 IAR 中。
2. 启动 ARM 的 IAR 嵌入式工作台（EWARM）。
3. 创建一个新的空白项目（**Project > Create New Project**），并将其保存在所移植的 PSoC Creator 项目的‘.cydsn’目录中。



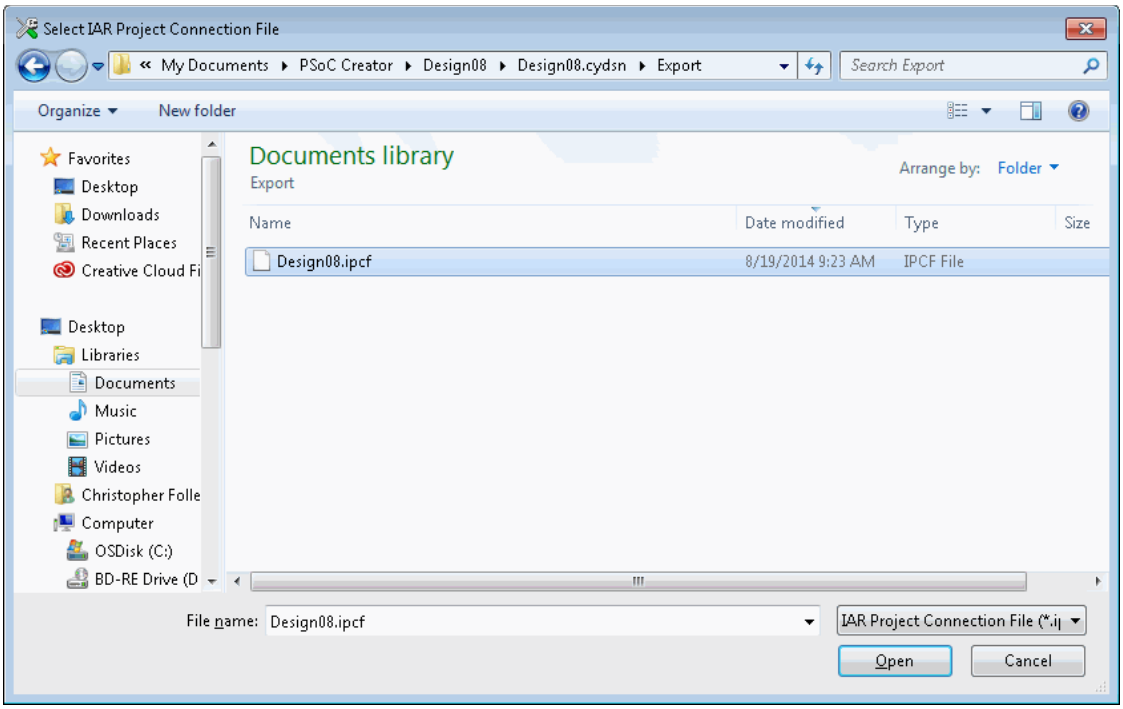
4. 打开 IDE Options 对话框（**Tools > Options**），并选择 **Project**。使能 **Add Project Connection**，并点击 **OK**。



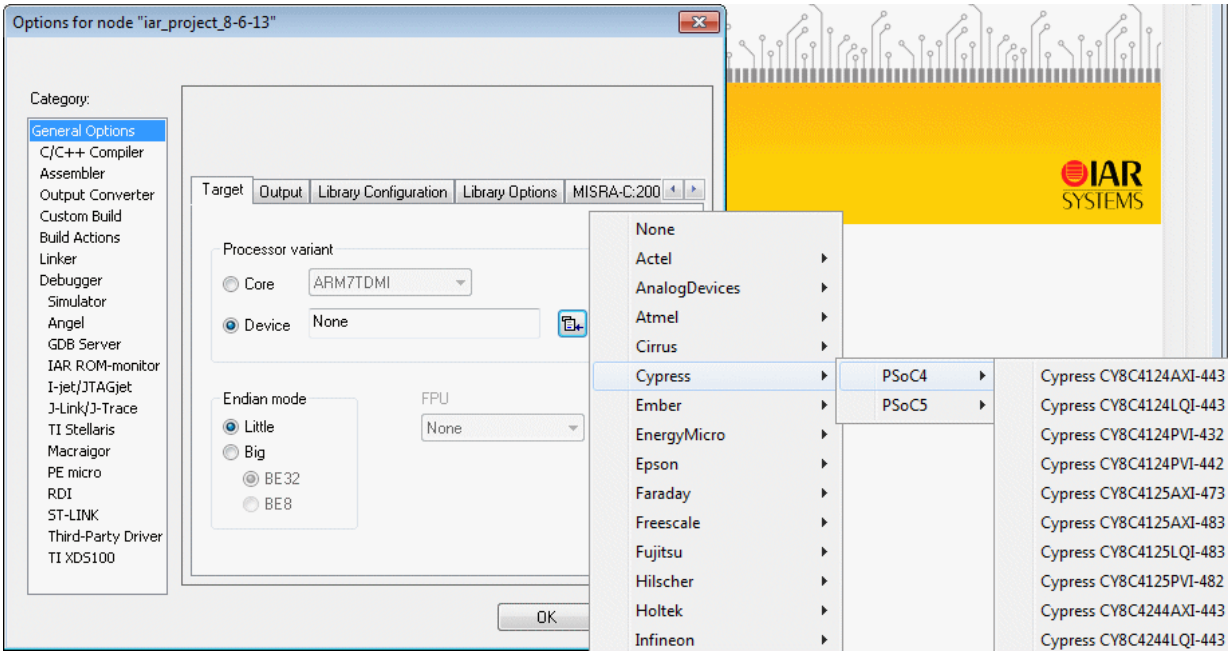
5. 打开 Add Project Connection（添加项目连接）对话框（**Project > Add Project Connection**），选择 **IAR Project Connection** 并点击 **OK**。



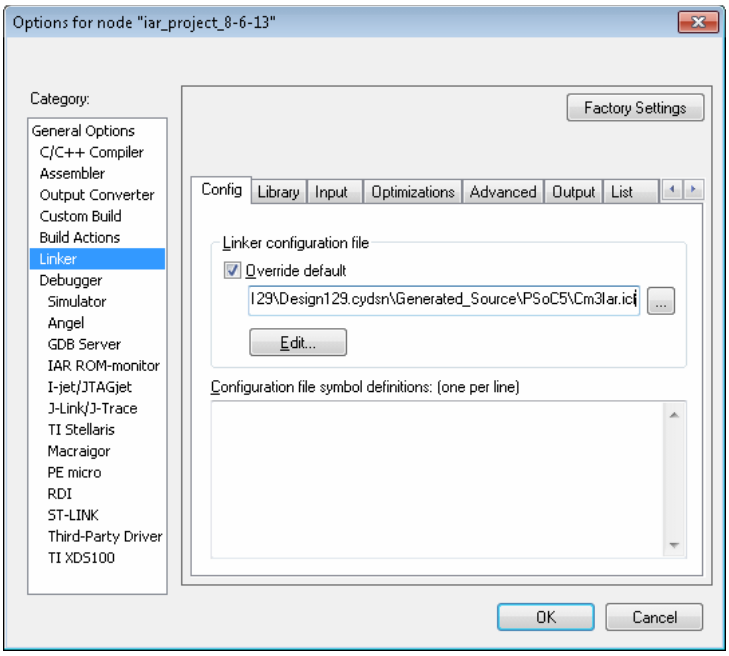
6. 在 Select IAR Project Connection File（选择 IAR 项目连接文件）对话框中，请浏览 PSoC Creator Export 目录，选择‘.ipcf’文件，然后点击 **Open**。



7. 在 EWARM Workspace 窗口中，请右键单击项目，并选择 **Options...**来打开 Options 对话框。
   - 在 **General Options** 页面上，选中‘Processor variant’下面的 **Device** 项。然后点击 **Select Device** 按键，并选择相应的赛普拉斯器件。



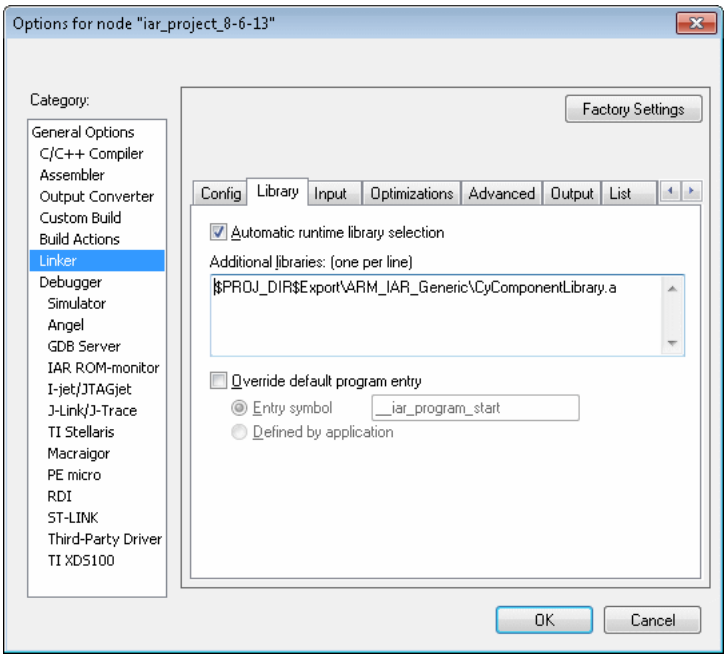
   - 在 **Linker** 页面上，请选择 **Config** 选项卡，然后勾选链接器配置文件的 **Override default** 项。导航到您的项目目录下，然后转到 Generated_Source/{ARCH}目录，查找您项目中的‘.icf’文件：



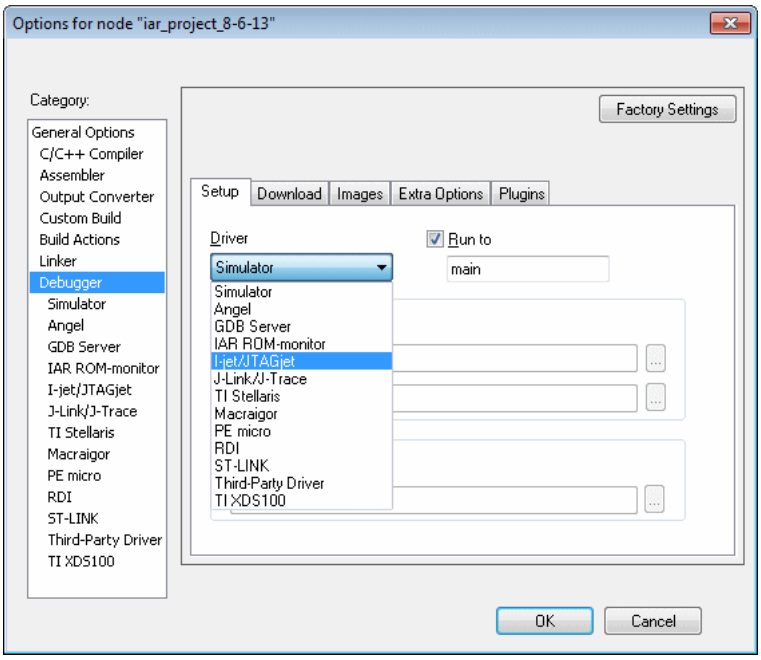
□ 在 **Linker** 页面上，请选择 **Library** 选项卡。将路径添加到“Export\ARM_IAR_Generic”目录中存在的每一个库内。

将 IAR 指令$PROJ_DIR$添加到每个库的面前，使 IAR 能在项目目录中找到它们。这可以确保您的程序将会与器件的正确库链接。例如：

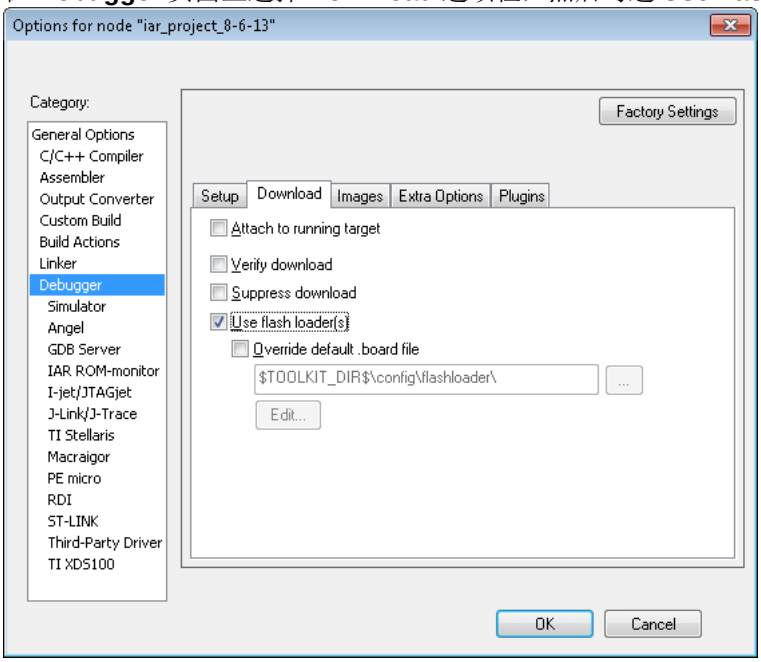
$PROJ_DIR$\Export\ARM_IAR_Generic\CyComponentLibrary.a



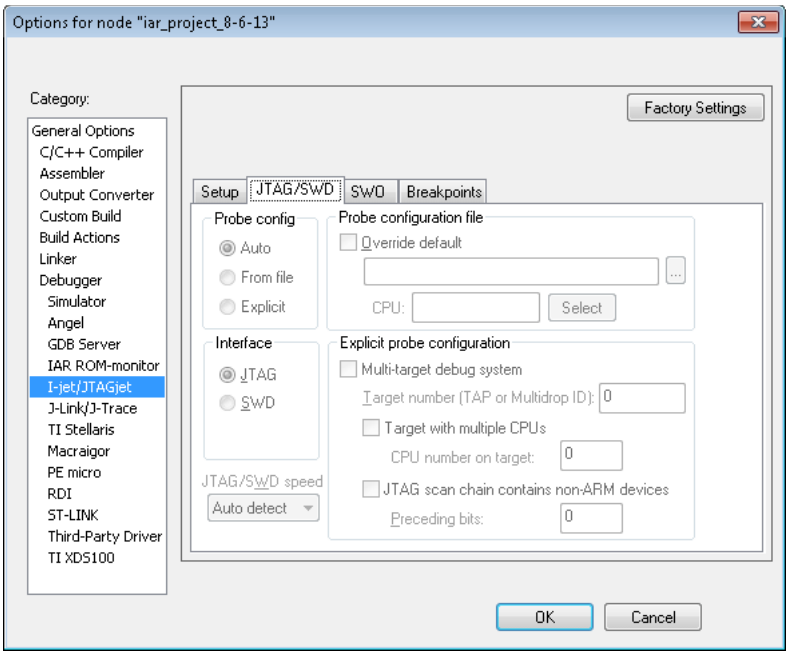
□ 在 **Debugger** 页面上，请单击 **Driver** 下拉菜单，并选择相应的调试器探针（目前支持 i-Jet 和 J-Link）。



□ 在 **Debugger** 页面上选择 **Download** 选项栏，然后勾选 **Use flash loader(s)**项。



□ 选择调试器的节点，并选择 **JTAG/SWD** 选项卡。然后选择器件上相应的接口。



8. 点击 **OK**，关闭 Options 对话框。

现在，您可以构建、编程和调试您的 EWARM 项目了。如果需要，请参考 IAR 文档。

如何设置 i-Jet 编程和调试：

对于 **PSoC 5 LP**，将带有 10 引脚适配器的 i-Jet 连接到 CY8CKIT-001 套件的 PSoC 5 LP 模块上的 PROG 10 引脚连接器。

对于 **PSoC 4/PRoC BLE**，将带有 10 引脚适配器的 i-Jet 同 CY8CKIT-042（Pioneer 套件）的“PSoC 4/PRoC BLE Prog” 10 引脚连接器连接起来。

有关如何使用 IAR i-Jet/调试器系统的信息，请参考 IAR IDE。在 Help 菜单中，打开名称为 *C-SPY Debugging Guide*（C-SPY 调试指南）和 *i-Jet User Guide*（i-Jet 用户指南）的文档。

另外，请参考：

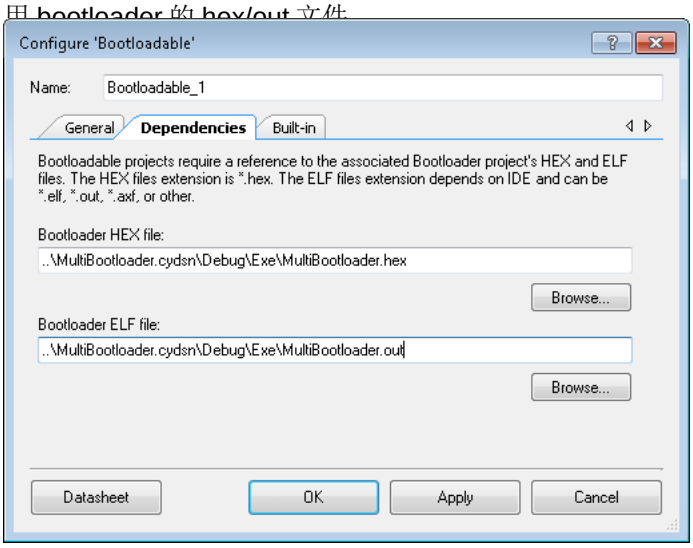
- 将设计移植到第三方 IDE 内
- 将设计移植到 IAR IDE 内

IAR 多应用 Bootloader 移植支持

本部分介绍的是如何移植 PSoC Creator 多应用 Bootloader 项目，以便能在 IAR IDE 环境中使用它。

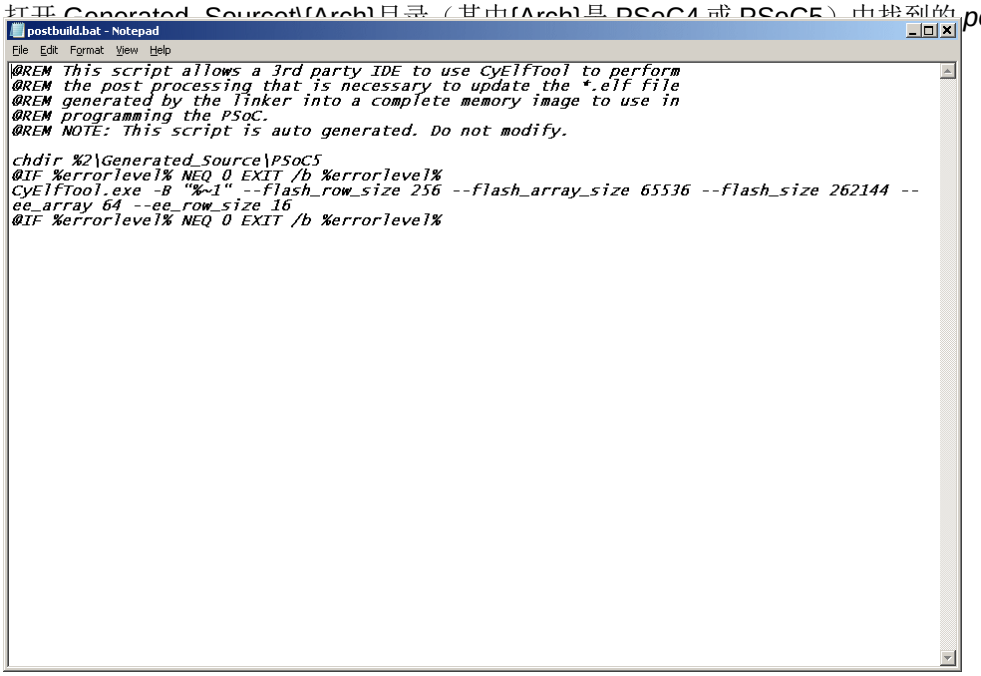
1. 正常移植多应用 bootloader 项目。请参考将设计移植到 IAR IDE 内部分。
2. 在 IAR 中编译 bootloader 项目。

3. 在 PSoC Creator 中，配置 Bootloadable（可引导加载）项目的 Bootloadable 组件，以指向 IAR 输出目录中多应用 bootloader 的 hex/out 文件。



4. **除**发生以下变化外，均需要正常导出您的 Bootloadable 项目：
5. 当给 Bootloadable 项目命名时，将“_1”添加到该名称的后面。
6. 当选择链接器配置文件时，请确保选择其名称中包含“_1”的文件。
7. 正常编译项目。
8. 在 Bootloadable 项目‘.cydsn’目录中创建第二个 IAR 项目，具体如下：
9. 当给新项目命名时，将“_2”添加到该名称的后面。
10. 当选择链接器配置文件时，请选择其名称中包含“_2”的‘.icf’文件。
11. 正常编译新项目。
12. 然后使用 *CyElfTool.exe* 工具构建一个包含了 bootloader 和两个 Bootloadable 映射文件的组合文件。

13. 打开 Generated_Source\{Arch}目录（其中{Arch}是 PSoC4 或 PSoC5）中找到的 *postbuild.bat*。



14. 复制 `-flash_row_size`、`flash_array_size`、`flash_size` 指令行参数。如果出现 `-ee_array` 和 `-ee_row_size` 参数，也需要复制它们。

15. 在 Windows 指令行中，通过 cd 指令进入 Bootloadable 项目的‘.cydsn’目录中。

16. 使用下面的指令运行 *CyElfTool.exe*：

    ```
    \Generated_Source\{Arch}\CyElfTool.exe -M Debug\Exe\{ProjectName}_1.out
    Debug\Exe\{ProjectName}_2.out Debug\Exe\{ProjectName}.hex
    ```

    （复制来自 *postbuild.bat* 的参数，如：`--flash_size 262144 --flash_row_size 256 --ee_array 64 --ee_row_size16`）

    注意：如果您使用释放配置，请确保将上述指令行中的所有 Debug（调试）实例更改为 Release（释放）。

17. 现在，在 UVBuild 目录中存在两个‘.out’文件和一个‘hex’文件。根据要求，您可以编程并调试 Bootloader、Bootloadable 或组合文件。

18. 如果您修改了某个 Bootloadable 项目，请确保也对另一个 Bootloadable 项目进行同样的更改。如果您从 PSoC Creator 中重新导出某个 Bootloadable 项目，那么请确保通过上述步骤重新生成第二个 Bootloadable。

将设计移植到 Eclipse IDE 内

如要将某个设计从 PSoC Creator 移植到 Eclipse IDE 内，需要执行下面的高级操作：

- PSoC Creator Export Wizard（PSoC Creator 移植向导）
- 使用 J-Link 探头将 PSoC 5LP 设计保存到闪存内
- 配置注意事项

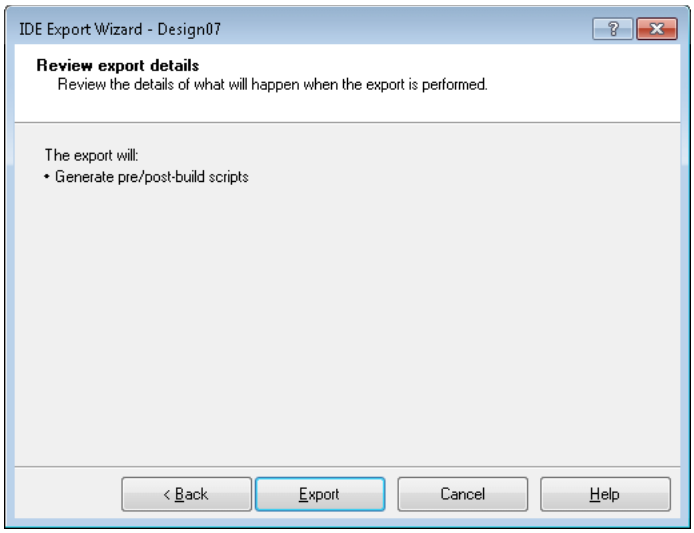
- 创建新的 Eclipse CDT 固件应用项目
- 在未安装 Cygwin/make 的情况下进行编译
- PSoC Creator 和 Eclipse 之间的设计迭代
- Eclipse 中的调试流程
- 项目编译设置
- 项目自定义
- 编译自定义
- 多应用 Bootloader 支持

PSoC Creator Export Wizard（PSoC Creator 移植向导）

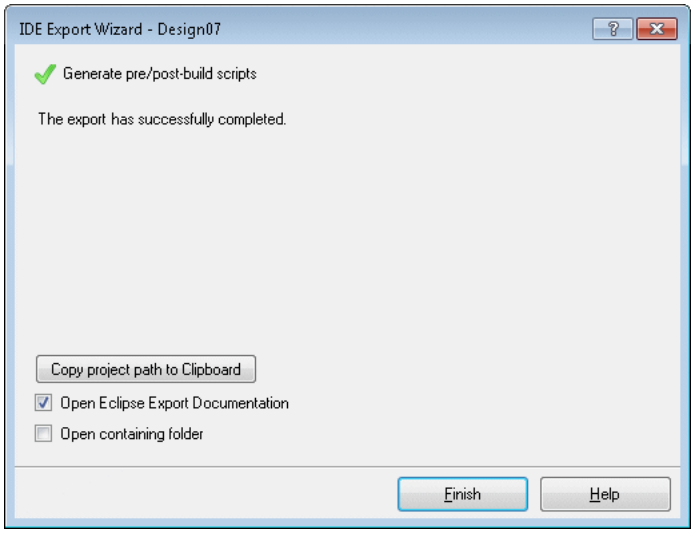
要想将某个 PSoC Creator 设计移植到 Eclipse IDE 环境中，请选择“Export IDE”对话框中的“Eclipse”项。

注意：对于 PSoC 3 项目，Eclipse IDE 选项无效。

导出过程比较简单。下一步进行的是确认您是否要导出该设计。点击 **Export**。



移植完成后，将显示最后步骤。



- 使用 **Copy project path to Clipboard**（将项目路径复制到剪贴板）按键复制项目文件夹的路径。在 Eclipse New Project 对话框中粘贴该路径，将其作为位置路径。
- 选择 **Open Eclipse Export Documentation**（打开 Eclipse 导出文档）项，以显示所需文档。
- 选择 **Open containing folder**（打开包含文件夹）项，以打开一个文件夹显示已被导出的所有文件。

选择相应选项，并点击 **Finish**，以完成导出过程。

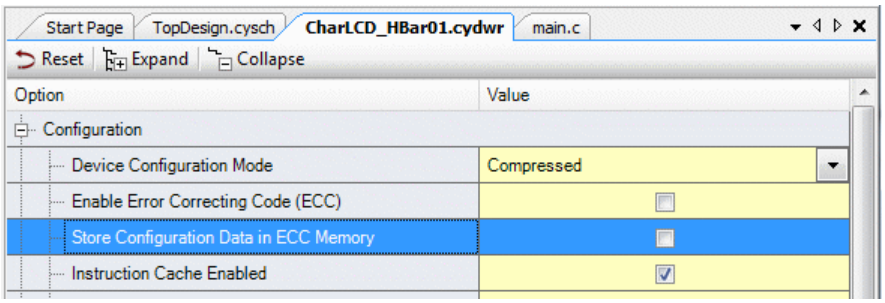
注意：

- Eclipse 中创建的项目文件夹与原始的 PSoC Creator 项目共享了文件系统文件夹。这样，在后续更改中，可以在 PSoC Creator 中查看固件开发者在 Eclipse 内进行的更改（用于合并任意组件文件的区域），反过来也一样。
- 当删除 Eclipse 中的某个项目时,请勿选择 **Delete project contents on disk**（删除硬盘上的项目内容）选框。这是因为文件系统文件夹的内容会随之被移除，这样在 PSoC Creator 中，这些内容将变为无效。

使用 J-Link 探头将 PSoC 5LP 设计保存到闪存内

如果您将某个基于 PSoC 5LP 的设计导入到 Eclipse 内，您必须了解下面的限制内容。对于 PSoC 5LP 器件，这些限制将在 Segger J-Link Flashloader 的以后版本中被移除。

- 在编译某个基于 PSoC 5LP 的设计并将其移植到 Eclipse 中前，请访问您项目的 Design Wide Resources System（设计范围资源系统）页面，然后取消选择 Store Configuration Data in ECC（将配置数据保存到 ECC 中）选框。对于 PSoC Creator 项目，配置数据被默认存储到 ECC 存储器内。但如果将这些设计导入到 Eclipse 内，则必须更改它们的设置，以在将它们保存到闪存后可以正常运行它们。



- 不能在您的设计中使用 EEPROM 存储器（带有任何通过 PSoC Creator EEPROM 编辑器设置的初始值）。带有 Segger 的 PSoC 5LP Flashloader 尚未编程该存储器空间。
- 同样，您的设计也不能使用 ECC 存储器，因为带有 Segger 的 PSoC 5LP Flashloader 尚未编程该存储器空间。在您项目的 DWR 系统设置中，应该取消选择 Enable Error Correcting Code (ECC)（使能纠错码（ECC））选框，如上所示。
- 带有 Segger 的 PSoC 5LP Flashloader 尚未编程 NVL 存储器空间。如果在 PSoC Creator 中更改了您项目中的下述设置，请先通过赛普拉斯 MiniProg3 探头在 PsoC Creator 中编译并下载您的设计。然后，可将该设计导入到 Eclipse，并通过 J-Link 探头对其进行编程：
  - DWR 系统设置，用于在启动期间启用快速 IMO
  - DWR 系统设置，用于使能器件保护
  - DWR 系统设置，用于调试选择
  - DWR 系统设置，用于使用可选的 XRES
  - 各个引脚上的初始状态设置

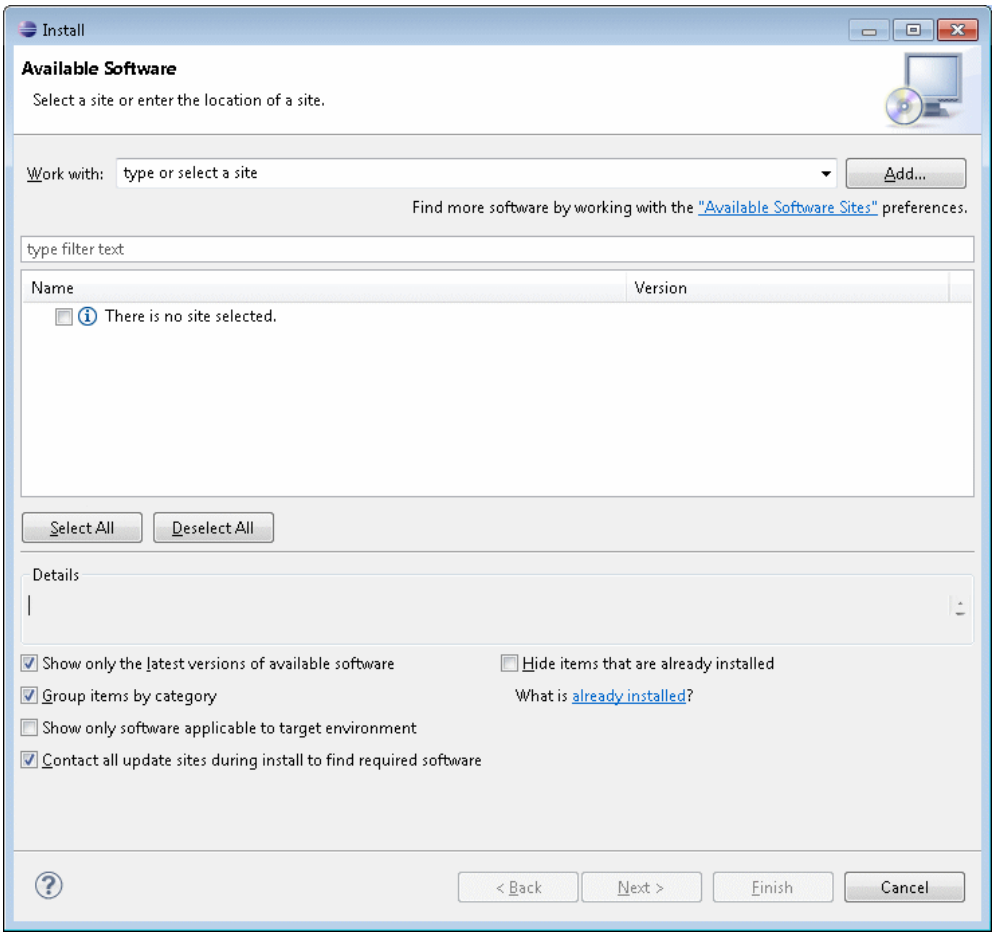
配置注意事项

- 使用 Eclipse IDE 时，您不需要在固件开发过程中进行整个 PSoC Creator 安装流程。
- 但您必须安装好 Java JRE 或 JDK（Java 版本 6 或更高版本）。
- 如果您尚未安装 cygwin 或 make，请参考在未安装 Cygwin/make 的情况下进行编译。
- 你还要安装好一个 GCC ARM 编译器。通过访问 https://launchpad.net/gcc-arm-embedded 网站，可以获取 GNU 工具 ARM 嵌入式工具集。
- 您必须从 http://www.eclipse.org/cdt 安装 Eclipse CDT。在该网站上寻找“Eclipse C/C++ IDE”下载封装。Juno SR2、Kepler 和 Luna 版本均支持赛普拉斯所提供的 Eclipse 性能。
  - 如果您正在运行一个 32 位操作系统，请下载 32 位 Eclipse 版本。
  - 如果您正在运行一个 64 位操作系统，那么要求您下载的 Eclipse 版本符合已安装的 Java 运行操作，即为

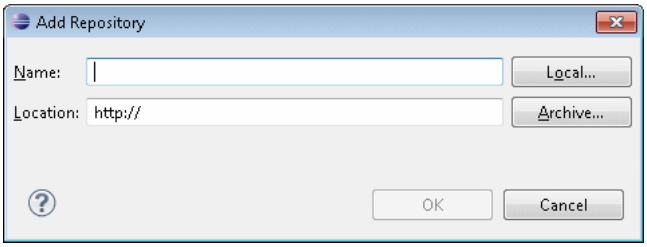
32 位 Java 下载 32 位 Eclipse；为 64 位 Java 下载 64 位 Eclipse。

- 此外，还要安装赛普拉斯所提供的新 Eclipse 性能（com.cypress.psoccreatorimport），具体包括：
  - 加快与 PSoC Creator 设计相关联的 Eclipse 项目的创建过程，其中，使用新的 PSoC Creator 项目类型和预填充的工具链编译选项。
  - 提供了 PSoC Creator 和 Eclipse 之间的项目资源同步化。对 PSoC Creator 中构成某个项目的文件组进行的更改将反映在 Eclipse 的后续编译过程中。

  请从 www.cypress.com/go/creator/eclipseimportdownload 网站下载赛普拉斯所提供的 Eclipse 性能，并按照下面各步骤安装所下载的压缩文件：
  - 在 Eclipse 中，依次选择 **Help > Install New Software…**以打开 Install 对话框。



  - 选择该对话框中 Details 部分的 Contact all update sites during install to find required software（安装过程中联系所有更新网站以寻找所需软件）选框。
  - 选择该对话框顶部的 **Add…**按键以打开 Add Repository 对话框。



  - 点击 **Archive…**并浏览到刚下载的赛普拉斯 Eclipse 插件压缩文件所在的位置。选择该文件并点击 **Open**，

然后点击 **OK**。

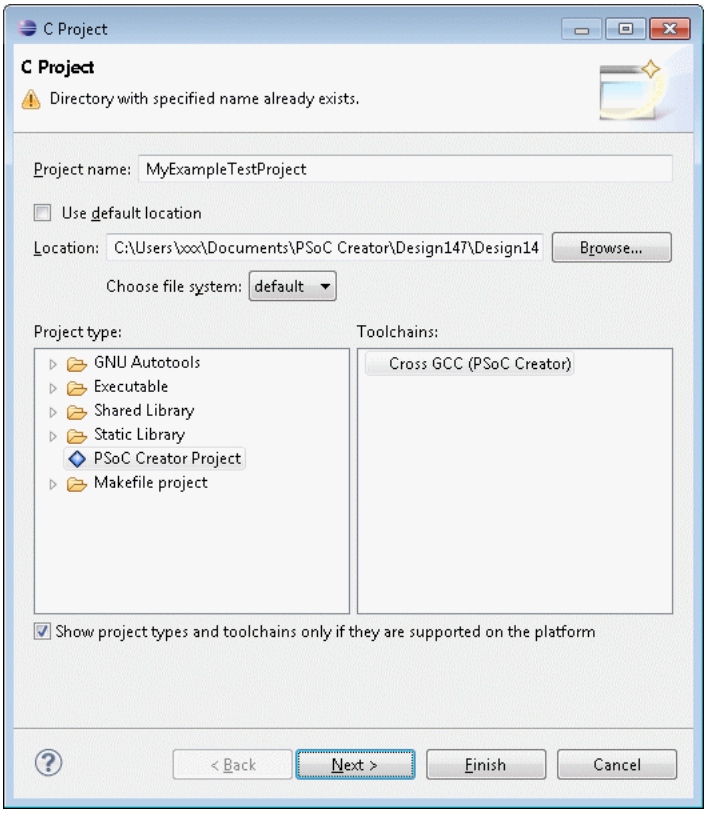
- 返回 Install 对话框，选择 PSoC Creator Import Feature 旁边的选框，然后点击 **Next >**。
- 下一个向导页面将总结将要安装的性能。点击 **Next >**，阅读并接受该性能的许可证。
- 点击 **Finish** 以安装插件。可能出现有关未签订许可证的软件的警报信息；点击 **OK**。
- 按照提示重启 Eclipse。

- 要想使用 Segger J-Link 调试探头，您必须从 http://www.segger.com/jlink-software.html 网站上下载 Segger J-Link 工具集。您应该使用版本 4.74 或更高版本。

创建新的 Eclipse CDT 固件应用项目

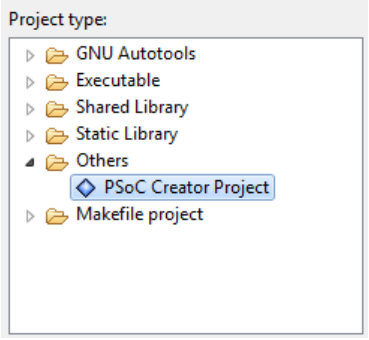
为您的固件创建一个 Eclipse CDT 可执行项目。请勿在 Eclipse 工作区中创建该项目。该项目需要共享您的 PSoC Creator 设计中的源代码文件。请执行下面各操作：

1. 依次选择 **File > New > C Project** 菜单条目以创建一个项目。在相应向导的 C Project 页面中：



- 选择 **Project Type** 下方的“PSoC Creator Project”项。

  注意：在某些 Eclipse 版本中，“PSoC Creator Project”输入将位于“Other”类别内。



- 选择 **Toolchains** 下方的“Cross GCC 项（for PSoC Creator）”。
- 对于项目位置，请取消选择 **Use Default Location** 选框，浏览到并选择您 PSoC Creator 项目的顶级文件夹（结尾为.cydsn 的文件夹）。这样，PSoC Creator 和 Eclipse 可共享相同的源文件。

2. 点击 **Next** 以跳转到 Select Configurations 向导页面。该页面无需任何更改。再次点击 **Next**。
3. 在 **Cross GCC Command** 向导页面上：
   - 默认的前缀为 arm-none-eabi-
   - 导航到您编译器安装目录的 bin 文件夹（包含 *arm-none-eabi-gcc.exe*），以设置所需路径。一般情况下，通过下面路径，可以找到所需位置：

     <your_ARM_tools_install_path>/bin

     （请寻找 *arm-none-eabi-gcc.exe*，而不是 *gcc.exe*。）

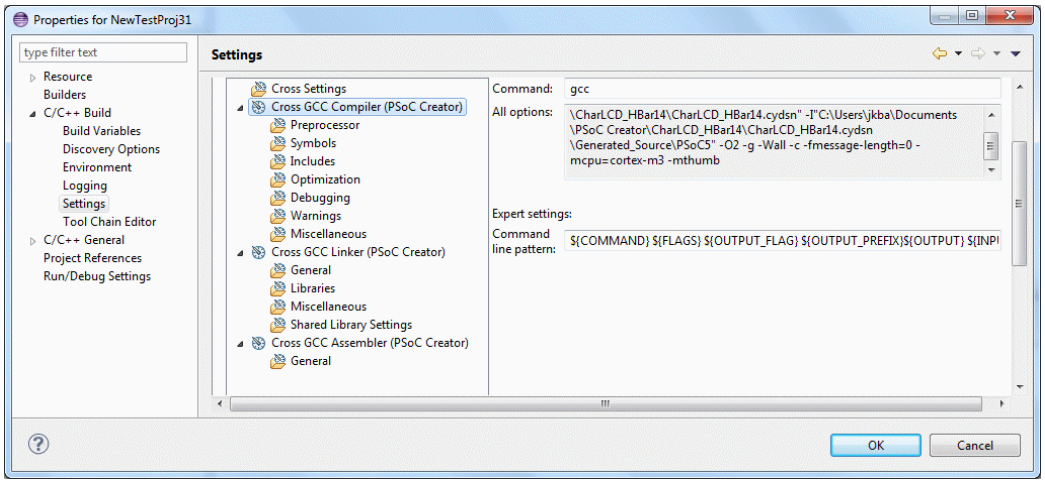
     注意：如果您已经安装了 PSoC Creator，请使用 PSoC Creator 中包含的 ARM GCC 安装程序。其路径为：

     <your_Creator_tools_install_path>/import/gnu_cs/arm/4.7.3/bin

Eclipse 中新创建的 PSoC Creator 项目包含工具链设置。可以通过 Project Explorer 窗格访问该项目。

1. 右击该项目名称，然后依次选择 **Properties > C/C++ Build > Settings**。
2. 选择 **Tool Settings** 选项卡。

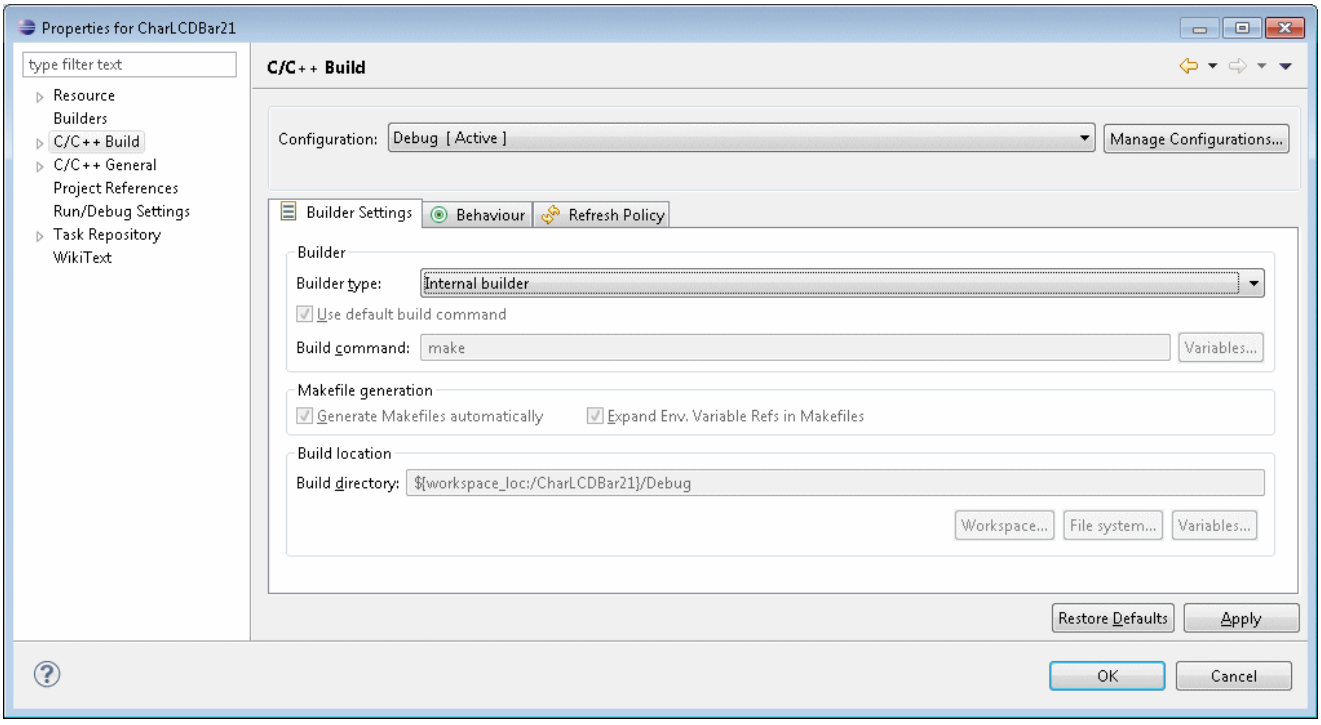
这些设置均由该项目的 PSoC 器件驱动。根据所使用的是 PSoC 4/PRoC BLE 还是 PSoC 5LP 器件，相应项目的可用选项会不一样。(本文档结尾的 Project Build Settings 部分列出了预填充选项设置，以供您更改或以后保存。这些值被设置为默认值，一般情况下，您不需修改它们。)



在未安装 Cygwin/make 的情况下进行编译

默认情况下，Eclipse CDT Builder 假设您的 PATH（路径）中包含 make 工具。通常通过将 cygwin 安装到您的机械上获得该工具。如果没有 make 工具，需要调整您项目的编译设置，以使用 Eclipse 内部编译器（Builder）。

1. 右击该项目名称，然后依次选择 **Properties > C/C++ Build**。
2. 选择 **Builder Settings** 选项卡。
3. 将 **Builder type** 中的值改为 Internal builder（内部编译器）。点击 **OK**。



PSoC Creator 和 Eclipse 之间的设计迭代

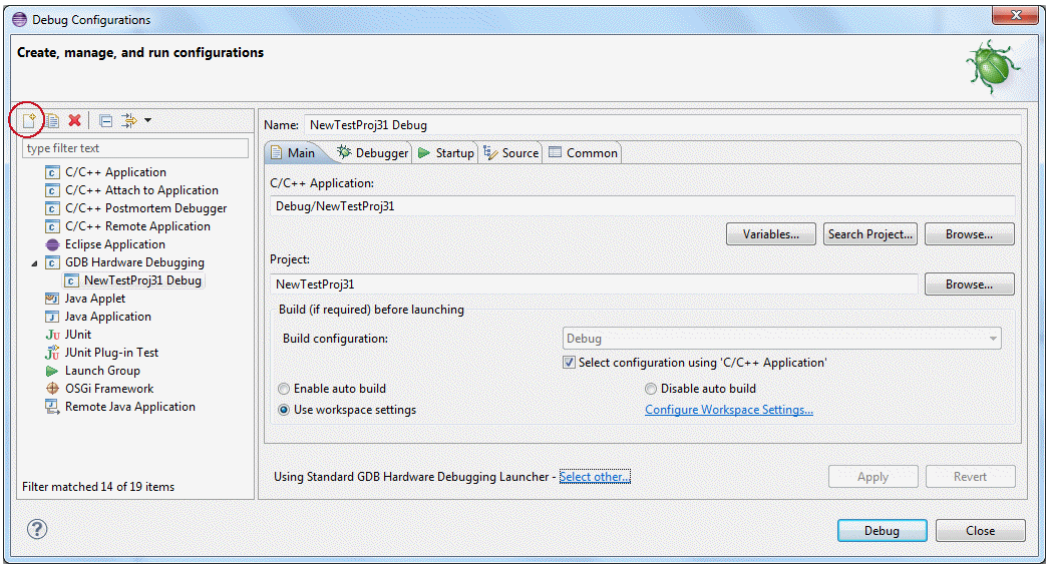
该部分内容指导用户通过使用他们的 PSoC Creator 设计文件夹创建 Eclipse 项目。用户只要在 PSoC Creator 重新构建他们的设计，然后再次运行 IDE Export 向导便可。赛普拉斯所提供的插件将检测对 PSoC Creator 项目所进行的构建更新，并使相应的更改在 Eclipse 项目文件中变为可见。当启动 Eclipse 时，将执行相应的检查和更新操作。

请注意，这是单向同步化过程：可以在 Eclipse 中查看 PSoC Creator 中所进行的更改，但不能在 PSoC Creator 中查看 Eclipse 中文件的添加和删除过程。这样，PSoC Creator 中的上行硬件配置更改将成为 Eclipse 中的下行更改，有助于后续的固件开发。

Eclipse 中的调试流程

按照下面各步骤，并通过使用一个 Segger J-Link 探头在 Eclipse/CDT 中闪存并调试所导入的 PSoC Creator 项目。

1. 给您的 PSoC 硬件供电，然后将一个 J-Link 调试探头连接到您的 PSoC 硬件和主机上。从 All Programs->SEGGER->J-Link ARM 菜单运行 J-Link Commander.exe。这样可以初始化 J-Link 探头。每次连接某个 J-Link 探头，只要执行该操作一次。（如果不执行该步骤，则 J-Link GDB Server 不能正常连接至目标。）退出应用前，请验证下面的条件：
   - 目标器件被设置为您的 PSoC 器件。
   - 目标接口被设置为 JTAG（仅用于 PSoC 5LP）或 SWD（用于 PSoC 4/PRoC BLE 或 PSoC 5LP）
   - 当工具指令提示时，键入‘exit’以退出应用。
2. 在 Eclipse 中，为您设计创建新的调试启动配置。依次选择 **Run > Debug Configurations…**
3. 通过选择 GDB Hardware Debugging 目录并点击 **New** 按键（下图中以圆圈显示），可以创建新的调试启动配置。



4. 可通过下述内容设置调试启动配置：

   Main 选项卡：
   - 提供项目构建时所生成的设计可执行文件的路径。导航到项目的 Release or Debug（发布或调试）文件夹寻找该路径，或仅点击 **Search Project…**按键并选择这个可执行文件。
   - 必须将调试器启动程序（launcher）设置为 Standard GDB Hardware Debugging Launcher（标准 GDB 硬件调试启动程序）。（可在 Main 选项卡底部寻找该项。如要更改该设置，请使用 Select other…链接。）

Debugger 选项卡：

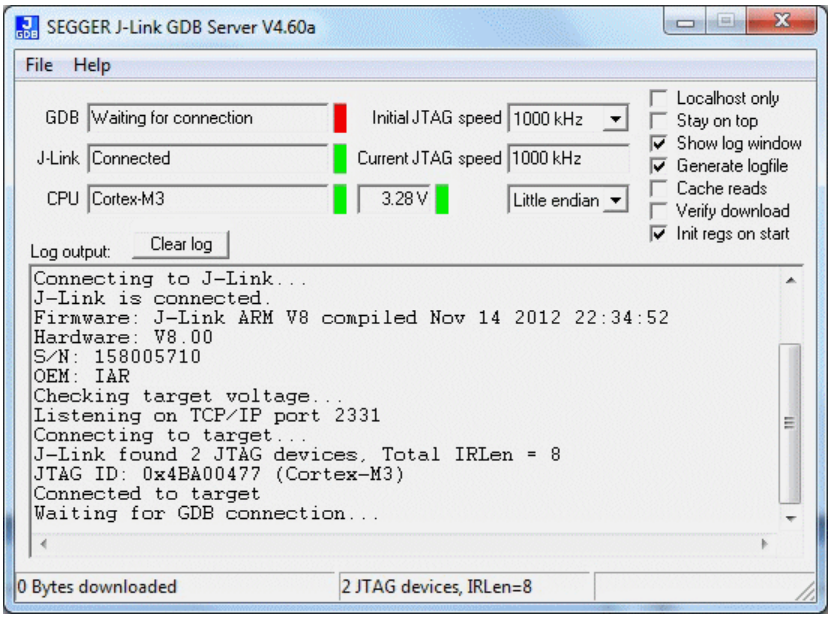
- 通过浏览到要使用的 GDB 可执行文件，可以设置 GDB 指令的路径。通常，该文件名称为 arm-none-eabi-gdb.exe，它与创建 Eclipse 项目时所提供的 Cross compiler path（交叉编译器路径）设置位于同一文件夹内。
- 选中 Use remote target（使用远程目标）项
- JTag 器件被设置为“Generic TCP/IP”
- 主机名称被设置为"localhost"，端口为 "2331"（为默认的 J-Link GDB 服务器端口）。

Startup 选项卡：

- Reset（复位）和 Delay（延迟）被设置为 3 秒
- Halt（暂停）被选中
- 在 Init 指令字段中输入 "monitor reset"：
- 必须选中 Load Image（加载图像）和 Load Symbols（加载符号）部分中的"Use project binary"项。
- Runtime（运行时）选项：选中"Resume"选框。
- 点击 Apply 和 Close，保存该调试启动配置，以供将来使用。

5. 依次选择 All Programs->SEGGER->J-Link ARM 菜单，以运行 J-Link GDB Server，进而启动 Segger J-Link GDB 服务器。验证是否设置了下面各条件：
   - 与 J-Link 的连接被设置为 USB
   - 根据要求，将目标接口设置为 JTAG 或 SWD。
   - 目标器件被设置为您当前使用的赛普拉斯 PSoC 器件。点击"..."按键以浏览到受支持的器件。从 Manufacturers（制造商）下拉菜单选择 Cypress 以罗列赛普拉斯的器件。选择您的器件系列，然后点击 OK。
   - 点击主要对话框窗口的 OK。

此时，服务器会连接至目标，并表示它正在等待 GDB 连接，如下图所示。



6. 在 Eclipse 中，用您的代码设置需要的断点。通过依次选择 **Run > Debug Configuration…**菜单运行上述第二步中所创建的调试启动配置，以便从 GDB 硬件调试标题下的配置定位您的调试启动配置。点击 **Debug** 按键，以启动调试会话。当 Eclipse GDB 会话开始时，GDB Server 将显示额外的输出。
7. 当遇到断点时，Eclipse 会更改它的调试视景并暂停。此处，普通的 Eclipse 调试功能（断点操作、检查/更改变脸和存储器，等等）均有效。

注意：如果在单步调试代码时 PC 不进行跟踪，那么，在调试时，你可以通过关闭您代码中的编译器优化解决该问题。请执行下面各步骤：

1. 右击该项目名称，然后依次选择 Properties > C/C++ Build。
2. 选择 Settings（设置）。
3. 选择 Cross GCC Compiler (PSoC Creator)项。
4. 选择 Optimization（优化）。
5. 将优化等级改为"None"。
6. 重新编译您的设计并再次进行调试。

Project Build Settings（项目编译设置）

下面各部分列出了 PSoC Creator Import Eclipse 功能所提供的默认工具选项。

在 Cross GCC Compiler 下方：

- **Includes** 下方提供了这些项目包含路径的默认值：
  - "${workspace_loc:/${ProjName}}"
  - "${workspace_loc:/${ProjName}/Generated_Source/${GEN_ARCH}}"
- 必须在 **Optimzation** 下方设置该标志：
  - -ffunction-sections
- 在 **Miscellaneous, Other flags** 下方，通过${PROC_FLAGS}变量的设置提供与 ARM 相关的标志默认值：
  - 对于 PSoC 5LP: -mcpu=cortex-m3 -mthumb
  - 对于 PSoC 4/PRoC BLE: -mcpu=cortex-m0 -mthumb
- 在 **Miscellaneous, Other flags** 下方，必须设置这些标志：
  - -c -fmessage-length=0

在 Cross GCC Assembler 下方：

- **Includes** 下方提供了这些项目包含路径的默认值：
  - "${workspace_loc:/${ProjName}}"
  - "${workspace_loc:/${ProjName}/Generated_Source/${GEN_ARCH}}"
- 在 **General, Other flags** 下方，通过${PROC_FLAGS}变量的设置提供与 ARM 相关的标志默认值：
  - 对于 PSoC 5LP: -mcpu=cortex-m3 -mthumb

- 对于 PSoC 4/PRoC BLE: -mcpu=cortex-m0 -mthumb

在 Cross GCC Linker 下方：

- 在 **Miscellaneous, Linker flags** 下方提供了这些标志的值：
  - -T ../Generated_Source/${GEN_ARCH}/${LINKER_FILE} -specs=nano.specs
- 在 **Miscellaneous, Linker flags** 下方，通过 ${PROC_LINK_FLAGS}变量的设置提供与 ARM 相关的标志的默认值：
  - 对于 PSoC 5LP: -march=armv7-m -mthumb -mfix-cortex-m3-ldrd
  - 对于 PSoC 4/PRoC BLE: -march=armv6-m -mthumb
- 在 **Miscellaneous, Linker flags** 下方提供了该默认值： -Xlinker --gc-sections
- 在 **Other Objects** 下方，必须出现下面对象路径：
  - "${workspace_loc:/${ProjName}/${EXPORT_FOLDER_PATH}/${TOOL_NAME}/CyComponentLibrary.a}"
- 对链接步骤指令行进行结构化处理，以便“`-Wl,--start-group`”和“`-Wl,--end-group`”选项包围所有输入文件和库，这样会提供多次链接搜索，用于解决各库之间的循环参考（如需要）。

在 Build Steps 选项卡下：

提供了下面各项默认设置：

- 预编译步骤将读取：
  "../${EXPORT_FOLDER_PATH}/prebuild.bat"
- 编译后步骤将读取：
  "../${EXPORT_FOLDER_PATH}/postbuild.bat"
  "${POSTBUILD_HEXFILE_PATH}/${ConfigName}/${ProjName}"

Project Customization（项目自定义）

编译器优化设置：

有助于降低所生成的闪存映像尺寸的其他优化设置包括：

在 Cross GCC Compiler 下方：

- 在 **Optimizations** 下方，将 Optimization level（优化等级）设置为 Optimize for size (-Os)（大小优化）

链接器脚本变更：

默认的 GCC 链接器脚本的位置为：

- 对于 PSoC 5LP，位置为: Generated_Source/PSoC5/cm3gcc.ld
- 对于 PSoC 4/PRoC BLE，位置为 Generated_Source/PSoC4/cm0gcc.ld

如要自定义可执行映像的链接步骤，请执行下面各步操作：

1. 将位于上述位置的文件复制到您 Eclipse 项目的顶级文件夹内。这样，在 PSoC Creator 的所有后续编译过程中，可以阻止覆盖您对该文件已经进行的更改。

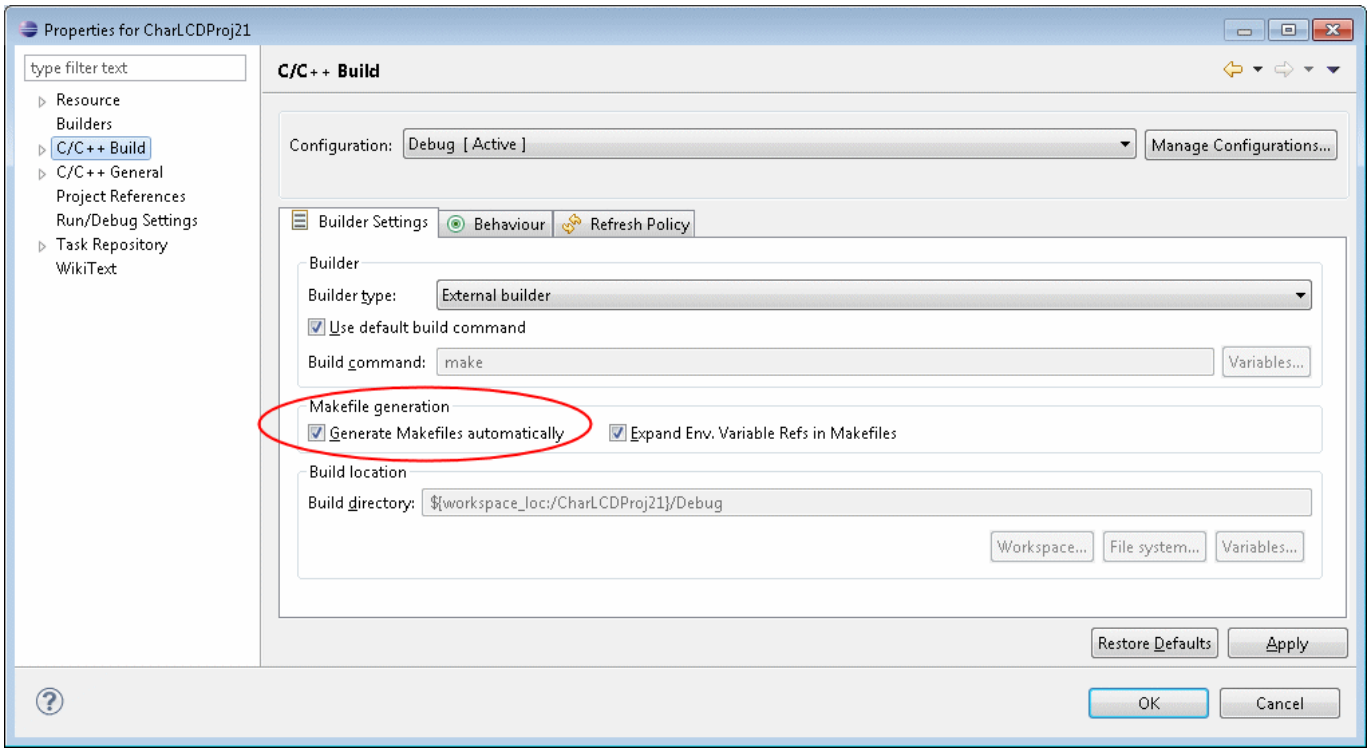
2. 对您刚复制的文件进行所需编辑。
3. 将（**Cross GCC Linker, Miscellaneous, Linker flags** 下方的)Build Settings（编译设置）链接器标志更改为：
   - 对于 PSoC 5LP，它为: -T ../cm3gcc.ld
   - 对于 PSoC 4/PRoC BLE，它为: -T ../cm0gcc.ld

Build Customization（编译自定义）

如果您的设计需要通过特殊步骤才能完成编译（标准 Eclipse CDT 管理编译系统[MBS]的标准预编译/汇编/编译/链接/编译后步骤除外），您可能要考虑自己的管理项目的编译过程。 最简单的方法便是在每个项目编译过程中阻止 MBS 自动生成 Makefile。 这样，根据要求，您可以自定义已生成的 Makefile 以完成对您项目的编译。

要想关闭您项目中 Makefile 的自动生成功能，请执行下面各步骤：

- 在 Eclipse Project Explorer 中，右击项目名称，然后选择 **Properties…**
- 选择右侧窗格中的 **C/C++ Build** 项
- 取消选择 Generate Makefiles automatically 选框

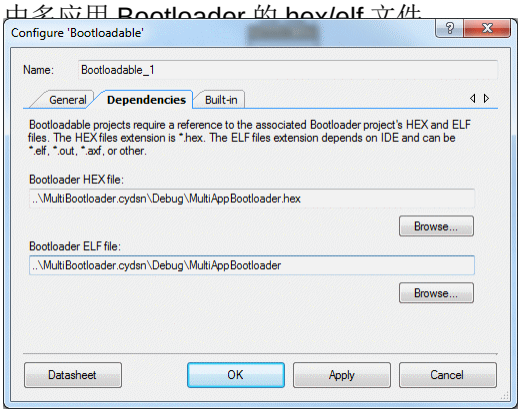


多应用 Bootloader 支持

本部分介绍的是如何导出一个 PSoC Creator 多应用 Bootloader 项目，并将其使用于 Eclipse IDE。

1. 在 PSoC Creator 中，编译并导出所需的多应用 Bootloader 项目。
2. 将该 Bootloader 项目导入 Eclipse 内，并对其进行编译。

3. 在 PSoC Creator 中，配置 Bootloadable（可引导加载）项目的 Bootloadable 组件，使之指向 Eclipse 输出目录内多应用 Bootloader 的 hex/elf 文件。



4. 导出 Bootloadable 项目。这样，将在磁盘上放置一个名称为{ProjectName}_1 的项目。

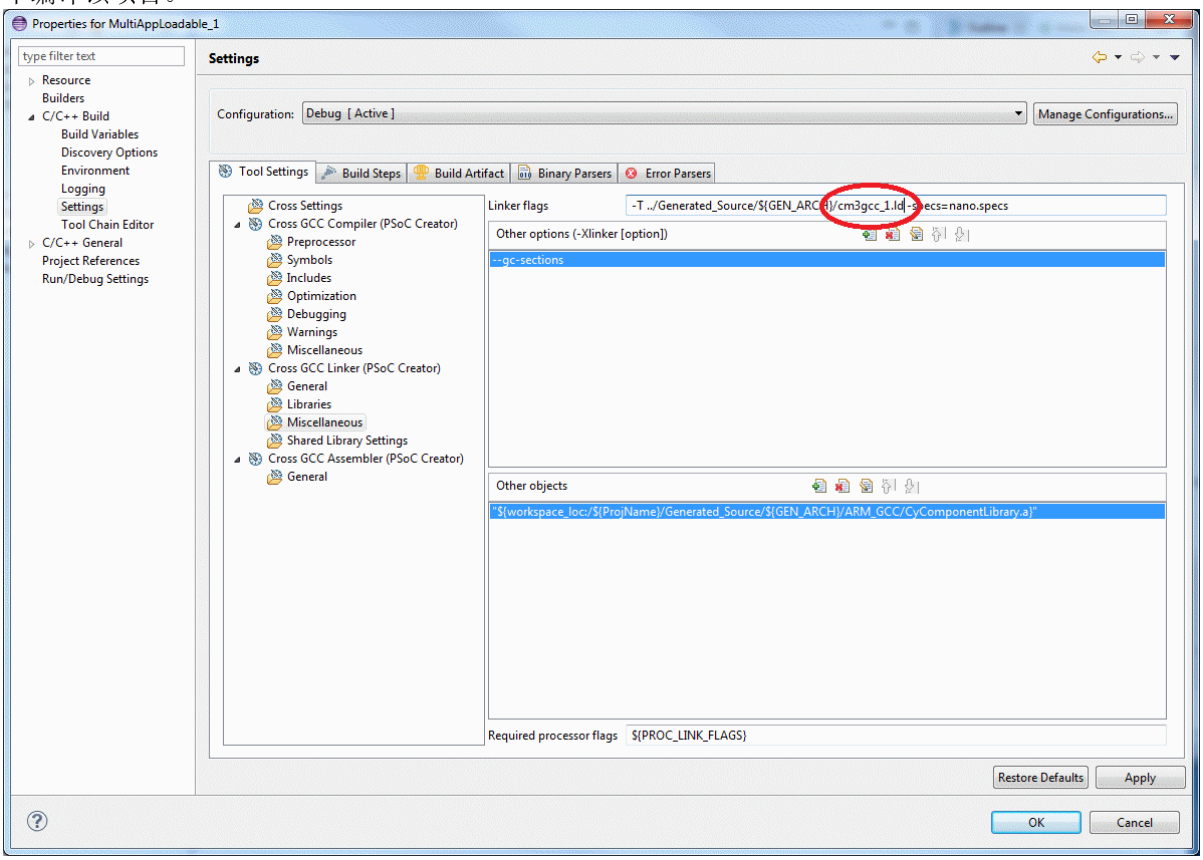
5. 将该项目导入到 Eclipse 内，并对其进行编译。

6. 然后，使用新的名称创建一个 Bootloadable 项目，并将其导入到第二个目录内。

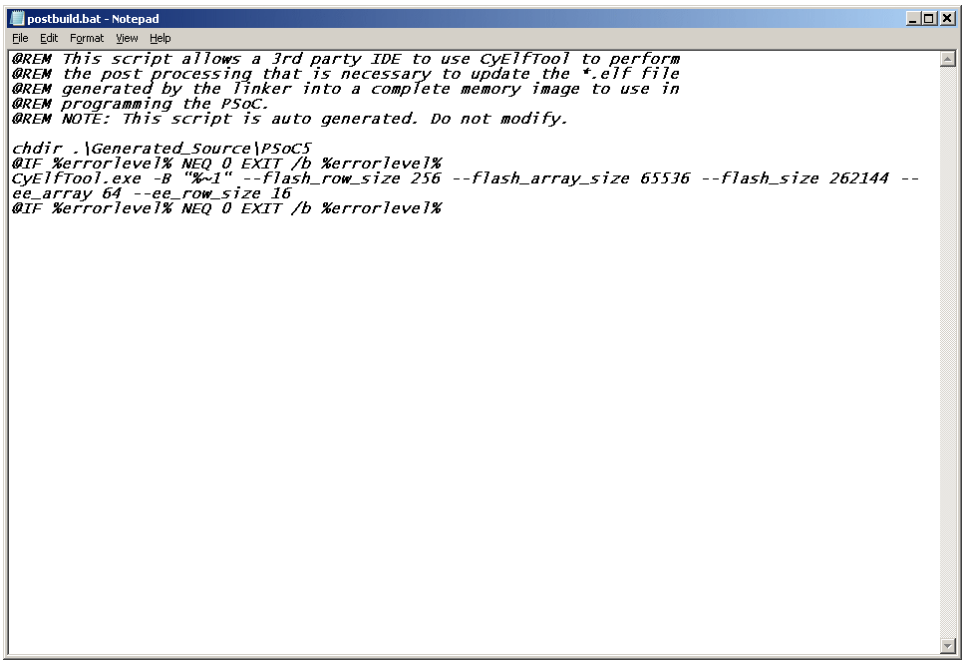
7. 在 PSoC Creator 中，将您的 Bootloadable 项目复制/粘贴为另一个名称，例如{ProjectName}_2。

8. 编译该第二个项目，并将其导入到 Eclipse 内。

9. 将{ProjectName}_2 项目导入到 Eclipse 后，更新该项目的链接器文件，如下面所示。对于{ProjectName}_2，请使用 cm3gcc_2.ld。（注意，对于 PSoC 4/PRoC BLE 项目，链接器文件的名称应为 cm0gcc_2.ld）。在 Eclipse 中编译该项目。



10. 对{ProjectName}_1 重复同样的操作，并使用 cm3gcc_1.ld）（或 cm0gcc_1.ld）。在 Eclipse 中编译该项目。
11. 最后，使用 CyElfTool.exe 创建包含 Bootloader 和两个可加载映像的组合十六进制文件。
12. 打开您 Bootloadable 2 项目中 Generated_Source\{Arch}目录下的 *postbuild.bat* 文件（{Arch}为 PSoC4 或 PSoC5）。



13. 复制`-flash_row_size`、`flash_array_size`、`flash_size` 指令行参数。如果出现`-ee_array` 和`-ee_row_size` 参数，请一同复制它们。
14. 在 Windows 指令行中，cd 到 Bootloadable 2 项目的‘.cydsn’目录中。
15. 使用下面指令或脚本文件运行 *CyElfTool.exe* 工具：

```
\Generated_Source\{Arch}\CyElfTool.exe -M {path to loadable_1
project}\Debug\{ProjectName}_1.cyacd Debug\{ProjectName}_2.cyacd {path to Bootloader
project}\Debug\{ProjectName}.hex
```

（复制来自 *postbuild.bat* 的参数，如：`--flash_size 262144 --flash_row_size 256 --ee_array 64 --ee_row_size16`）

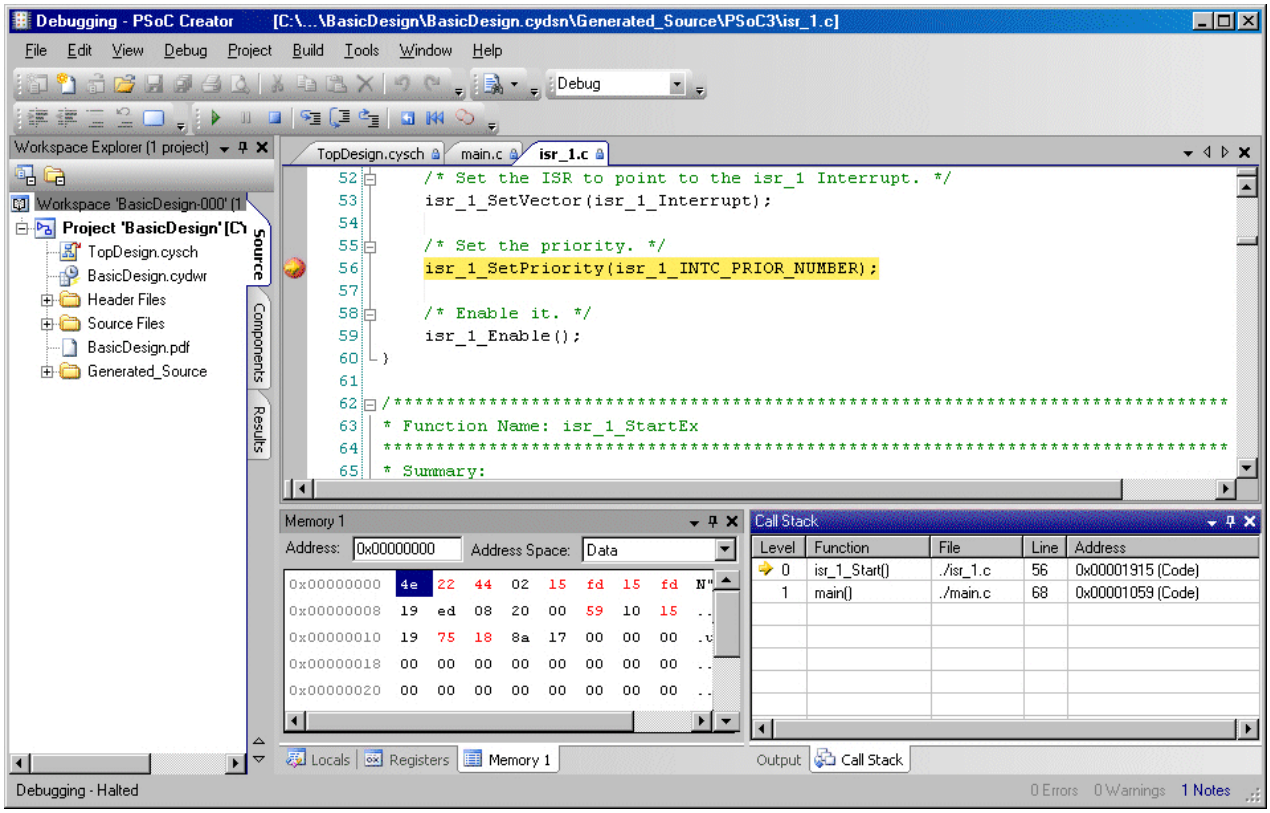
示例：

```
Generated_Source\PSoC5\CyElfTool.exe   -M   ..\Loadable_1.cydsn\Debug\MultiAppLoadable_1
Debug\MultiAppLoadable_2   Debug\MultiAppBootloadable.hex   --flash_row_size   256   --
flash_array_size 65536 --flash_size 262144 --ee_array 64 --ee_row_size 16
```

16. 在上述三个 Eclipse 项目调试目录中，您可以找到两个 Bootloadable .cyacd 文件和一个 Bootloader elf 文件。现在，根据要求，您可以编程并调试 Bootloader、Bootloadable 或它们的组合文件。
17. 如果您对某个 Bootloadable 项目进行了更改，请确保也对另一个 Bootloadable 项目进行了同样的更改。
18. 如果您从 PSoC Creator 重新导出某个 Bootloadable 项目，那么请确保通过上述的手动步骤重新生成第二个 Bootloadable 项目。

8 如何使用调试器

通过 PSoC Creator 调试器，您可以观察程序运行时的行为并确定语义错误的位置。



调试器可以理解被编译为编程语言的功能以及与之相关的库。通过调试器，您可以断开（暂停）程序的执行以检查代码、评估并编辑程序的变量、查看寄存器、查看源代码所创建的指令以及观察您应用所使用的存储器空间。

通过使用调试器，您可以检查程序中各变量的内容，而不需要插入用于输出各个值的额外调用。可以在您的代码中插入某个断点，以在断点处暂停程序的执行。

当程序暂停（在断点模式下）时，您可以使用各工具（如 Watch（查看）窗口和 Memory（存储器）窗口）检查本地变量和其他相关数据。更多信息，请查看 Watch 窗口 或 Memory 窗口。在断点模式下，您不仅能查看内容，而且还能编辑或更改所需的内容。在大多数情况下，您可在源文件（您编写和编辑代码的文件）内设置断点。有时可以在调试器的 Disassembly（反汇编）窗口中设置断点。Disassembly 窗口显示了您源代码所创建的指令。更多信息，请查看 Disassembly 窗口。与 **printf** 或 **MsgBox** 不同，当设置某个断点时，您不需要在源代码中添加额外的功能调用。因此，设置某个断点将不会更改您当前尝试调试的程序的性能。

注意：使用调试器时，您只要通过 PSoC Creator 就能够对您的器件进行编程，而不需启动 PSoC 编程器。此外，当您启动某个调试会话时，该器件自动被重新编程。

PSoC Creator 调试器提供了所需的菜单项、窗口和对话框，以访问其所有的工具。通过选择某一项，并按下[**F1**]键，您可以从任何窗口、对话框或控制项获得帮助。此外，您还可以在各个窗口间拖放并移动调试信息。调试器的主要子部分包括：

- 调试器指令
- 调试器菜单
- 文本编辑器的右键菜单指令
- 调试器窗口
- MiniProg3
- QuickProgrammer
- 错误处理

受支持的调试器：

PSoC Creator 支持 8051 和 ARM Cortex-M3 微处理器。支持的调试器为 GDB。

调试器工具栏指令

调试器工具栏包含了多条调试程序时将要使用的通用指令：

下表介绍了各个工具栏指令：

指令	图标	快捷方式	说明
执行代码/ 继续运行		[F5]	用于启动/继续运行调试器。 如果该项目已过期，将自动启动编译过程 更新状态栏的信息，以指出调试器正在启动 使用项目的最新版本编程所选目标 启动调试会话
暂停执行		[Ctrl]+[Alt]+[Break]	在任意运行阶段中暂停调试目标。当芯片不能正常运行，并且您想要查看实际的工作情况时，该指令非常有用。
停止调试		[Shift]+[F5]	终止调试会话，并将 PSoC Creator 置于标准的视景。当您调试完代码，并准备好进行更改或实现其他任务时，请使用该指令。
单步执行		[F11]	执行单一代码行。如果代码行是函数调用，则调试器将在函数的第一条指令处执行断点。如果代码行不是函数调用，调试器将在下一行执行断点。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。
Step Over （单步跳过）		[F10]	执行单一代码行。调试器将在下一个代码行上执行断点。如果当前的代码行是函数调用，那么不要停止也可以执行函数。那么函数调用后，调试器将停止在下一个行中。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。

指令	图标	快捷方式	说明
单步跳出		[Shift]+[F11]	执行完当前函数。处理器将继续运行，直到完成当前的函数为止。调用函数后，它将暂停在第一条指令处。使用该函数退出当前的函数，并返回调用函数。该函数暂时允许处理器运行，直到完成当前函数中的各条指令为止。
Rebuild and Run（重新编译及运行）		[Ctrl]+[Shift]+[F5]	暂停当前调试会话、重编译项目、使用已更新代码编程器件以及再次启动调试器。这样可以加快测试更改内容，因为每次进行更改时，不再需要启动和停止调试器。
重启		[Ctrl]+[Alt]+[F5]	将程序计数器（PC）复位为零，并将处理器置于运行状态。 这样，您可以启动运行程序而不需执行整个编译和编程序列。
启用/禁用所有断点			交替启用和禁用工作区中的所有断点。使用该指令快速更改所有断点的状态，而无需逐个更改每一个断点的状态。如果您设置了多个断点，但您需要处理器一直运行，那么该指令将非常有用。
启用/禁用全局中断			交替启用/禁用全局中断。如果禁用全局中断，当处理器单步调试 Main 代码时，将不会出现任何中断。

另外，请参考：

- 使用调试器
- 调试器菜单指令
- 文本编辑器的右键菜单指令

调试器菜单指令

通过 Debug（调试）菜单，可以访问当前调试器中所有可用功能和信息。调试菜单有两种模式：非活动模式和活动模式。

非活动模式的调试菜单：

调试器未运行时，调试菜单将处于非活动状态。在该状态中，只有必要的功能可用。例如，在非活动状态中，没必要使用暂停功能。在活动模式下，通过 Debug 窗口子菜单只能访问一定的可用调试窗口子集。

指令	图标	快捷方式	说明
Windows（窗口） >			用于访问各种调试器窗口。请参阅 Debugger 窗口。
Breakpoints（断点）		[Ctrl] + [D], [B]	打开 Breakpoints 窗口，显示设置在工作区中的所有断点。
Output（输出）		[Ctrl] + [D], [O]	打开 Output 窗口
Program（编写）		[Ctrl] + [F5]	单击该图标，将由选定项目生成的代码编写到选定的调试目标内。 如果该项目已经过期，将自动启动编译过程 更新状态栏的信息，以显示正在进行编程操作。 后台启动 PSoC Programmer，以进行编程操作。
Select Debug Target（选择调试目标）			打开 Select Debug Target 对话框，手动选择需要使用的调试目标。
Debug（调试）		[F5]	启动调试器。

指令	图标	快捷方式	说明
Debug without Programming（仅调试，不编程）		[Alt] + [F5]	启动调试器，不需要对器件进行编程。
Attach to Running Target（连接到运行目标）			打开 Attach to Target 对话框，连接到一个已经编程好的目标器件。仅适用于 PSoC 3 和 PSoC 5LP。
Toggle Breakpoint（切换断点）		[F9]	交替执行插入和移除当前代码行的断点。该项的作用与点击代码编辑器中的 **Indicator Margin**（指示器边距）的作用相同。该指令的快捷键为**[F9]**。
New Breakpoint（新断点） >			访问下面的断点窗口。请参阅断点窗口中的内容。
Address Breakpoint（地址断点）		[Ctrl] + [D], [A]	打开 Address Breakpoint 窗口。
File Line Breakpoint（文件行断点）		[Ctrl] + [D], [F]	打开 File/Line Breakpoint 窗口。
Function Breakpoint（函数断点）		[Ctrl] + [D], [U]	打开 Function Breakpoint 窗口。
Variable Watchpoint（变量观察点）		[Ctrl] + [D], [V]	打开 Variable Watchpoints 窗口。
Memory Watchpoint（存储器观察点）		[Ctrl] + [D], [E]	打开 Memory_Watchpoint 窗口。
Delete All Breakpoints（删除所有断点）		[Ctrl] + [Shift] + [F9]	删除工作区内的所有断点，并非逐个移除每一个断点。如果您设置了多个断点，但您想处理器保持运行，那么该指令便非常有用。
Enable All Breakpoints（启用所有断点）			启用所有断点

活动调试窗口的菜单：

活动调试模式表示您已经启动了调试会话；有效的 PSoC Creator 子系统便是调试器。在活动模式下，调试器正在运行，您可以调用所有可用的功能；通过调试窗口中的子菜单，可以访问所有可用的调试窗口。

指令	图标	快捷方式	说明
Windows（窗口） >			用于访问各种调试器窗口。请参阅 Debugger 窗口。
Breakpoints（断点）		[Ctrl] + [D], [B]	打开 Breakpoints 窗口，显示设置在工作区中的所有断点。
Output（输出）		[Ctrl] + [D], [O]	打开 Output 窗口

指令	图标	快捷方式	说明
Watch（观察） >			
1		[Ctrl] + [D], [W]	打开四个 Watch 窗口中的一个，进行评估和显示变量、寄存器或表达式。
2		[Ctrl] + [Alt] + [W], [2]	
3		[Ctrl] + [Alt] + [W], [3]	
4		[Ctrl] + [Alt] + [W], [4]	
Locals（局部）		[Ctrl] + [D], [L]	打开 Locals 窗口，查看并修改当前调试框架中所有局部变量。
组件		[Ctrl] + [D], [P]	打开 Component Debug 窗口，查看设计中有关组件的调试信息。
Call Stack（调用堆栈）		[Ctrl] + [D], [C]	打开 Call Stack 窗口，跟踪目标程序调用函数的顺序。
Memory（存储器）>			
1		[Ctrl] + [D], [M]	打开四个 Memory 窗口中的一个，用于显示处理器内存中保存的值。
2		[Ctrl] + [Alt] + [M], [2]	
3		[Ctrl] + [Alt] + [M], [3]	
4		[Ctrl] + [Alt] + [M], [4]	
Disassembly（反汇编）		[Ctrl] + [Alt] + [D]	打开 Disassembly 窗口，显示源代码中创建的基本指令。
Registers（寄存器）		[Ctrl] + [R], [R]	打开 Registers 窗口，显示内核 CPU 寄存器以及它们的值
Program（编写）		[Ctrl] + [F5]	单击该图标，将由选定项目生成的代码编写到选定的调试目标内。 如果该项目已经过期，将自动启动编译过程 更新状态栏的信息，以显示正在进行编程操作。 后台启动 PSoC Programmer，以进行编程操作。
Select Debug Target（选择调试目标）			打开 Select Debug Target 对话框，手动选择需要使用的调试目标。
Show Current Line（显示当前行）			显示正在执行或将要执行的代码行。
Resume Execution（恢复执行）		[F5]	进行执行调试器。暂停或执行断点后，将重启调试目标。使用该函数继续执行程序，直到下一个断点到来为止。
Halt Execution（暂停执行）		[Ctrl] + [Alt] + [Break]	暂停调试器。
Stop Debugging（停止调试）		[Shift] + [F5]	停止调试会话。

指令	图标	快捷方式	说明
Rebuild and Run（重新编译及运行）		[Ctrl] + [Shift] + [F5]	重新编译程序，并重启调试器。
Reset（复位）		[Ctrl] + [Alt] + [F5]	复位调试器。
Step Into（单步执行）		[F11]	执行单一代码行。如果代码行是函数调用，则调试器将在函数的第一条指令处执行断点。如果代码行不是函数调用，调试器将在下一行执行断点。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。
Step Over（单步跳过）		[F10]	执行单一代码行。调试器将在下一个代码行上执行断点。如果当前的代码行是函数调用，那么不要停止也可以执行函数。那么函数调用后，调试器将停止在下一个行中。使用该函数可验证是否正确执行了代码行。该函数暂时允许处理器运行，直到处理完当前代码行中的指令为止。
Step Out（单步跳出）		[Shift] + [F11]	执行完当前函数。处理器将继续运行，直到完成当前的函数为止。调用函数后，它将暂停在第一条指令处。使用该函数退出当前的函数，并返回调用函数。该函数暂时允许处理器运行，直到处理完当前函数中的各条指令为止。
Toggle Breakpoint（切换断点）		[F9]	交替执行插入和移除当前代码行的断点。该项的作用与点击代码编辑器中的 **Indicator Margin**（指示器边距）的作用相同。该指令的快捷键为**[F9]**。
New Breakpoint（新断点） >			访问下面的断点窗口。请参阅断点窗口中的内容。
Address Breakpoint（地址断点）		[Ctrl] + [D], [A]	打开 Address Breakpoint 窗口。
File Line Breakpoint（文件行断点）		[Ctrl] + [D], [F]	打开 File/Line Breakpoint 窗口。
Function Breakpoint（函数断点）		[Ctrl] + [D], [U]	打开 Function Breakpoint 窗口。
Variable Watchpoint（变量观察点）		[Ctrl] + [D], [V]	打开 Variable Watchpoints 窗口。
Memory Watchpoint（存储器观察点）		[Ctrl] + [D], [E]	打开 Memory_Watchpoint 窗口。
Delete All Breakpoints（删除所有断点）		[Ctrl] + [Shift] + [F9]	删除工作区内的所有断点，并非逐个移除每一个断点。如果您设置了多个断点，但您想处理器保持运行，那么该指令便非常有用。
Enable All Breakpoints（使能所有断点）			使能所有断点。
Refresh（刷新）			刷新调试器。

指令	图标	快捷方式	说明
Enable/Disable Global Interrupt（使能/禁用全局中断）			根据当前的状态使能或禁用全局中断

另外，请参考：

- 调试器的使用
- 调试器工具栏指令
- 文本编辑器的右键菜单指令

调试器的指示符

本部分介绍了在调试器中出现的各个指示符标签：

断点：

下表显示的是您可能会遇到的各种不同的断点符号：

	硬件	软件
永久		
暂时		

主要的图表共有下面 6 个：

- 断点被使能
- 断点被禁用
- 条件/命中次数的断点被使能
- 条件/命中次数的断点被禁用
- 断点因设置错误而被禁用。请查看 Notice List 窗口中的错误。
- 条件/命中次数的断点因设置错误而被禁用。请查看 Notice List 窗口中的错误。

有四个主要的类型：硬件、软件、永久和暂时。红色表示硬件；绿色表示软件。右上角的‘1’表示暂时。

观察点：

当 CPU 进行读取、写入或访问指定的存储器空间时，观察点会暂停程序运行。下面是三种观察点图标：

- 读取的观察点
- 写入的观察点
- 访问的观察点

当前代码行的指示符：

它表示被执行的当前代码行。这是指令在恢复程序时将被执行的指示。

活动的堆栈元素：

如果当前的堆栈元素调用的不是堆栈中最上面的一项，则表示它处于活动状态。它指示调试器不是仅仅关注堆栈最顶层的元素。

堆栈元素：

这里是在一个代码行周围的灰色突出显示，表示它作为调用堆栈的一部分，并指出如何执行代码。

另外，请参考：

- 调试器的使用
- 断点窗口
- 变量观察点
- 调用堆栈窗口
- 软件断点
- 注意列表窗口

调试器状态信息

当使用调试器时，PSoC Creator 状态栏在不同时期会显示不同的信息。下面各节说明的是您会遇到的调试器信息：

- **Starting Debugger**（启动调试器）— 指示您请求开始调试项目。PSoC Creator 正在配置选定的器件以进行调试。
- **Checking Build**（检查构建）— 指示启动调制器过程中实际进行的操作。检查构建过程是否必要会占用一些时间，然后才会开始进行构建。该操作占用了大量 CPU。
- **Programming**（编程）— 表示正在编辑选定的调试对象。完成编程过程需要 3 - 5 秒。这条消息提供了有关实际发生的操作的信息。
- **Debugging - Halted**（调试 - 停止）— 表示目标设备已被停止，用户可以与之互动。调试器有两种状态，即停止和运行。两种状态之间用户可用的选项也有明显的差异。通过该信息，可以对调试器的状态一目了然。
- **Debugging - Running**（调试 - 运行）— 表示目标设备正在运行，用户不能与之互动。

另外，请参考：

- 调试器的使用

调试器窗口

调试器窗口

PSoC Creator 调试器提供了一组窗口，从而为调试环境提供有用的信息。虽然默认显示了某些窗口，但这些窗口的显示都是可选的。

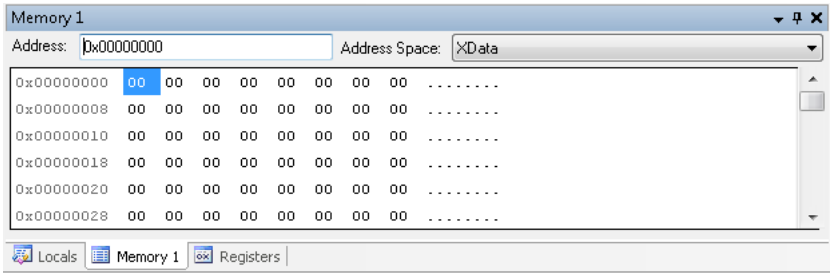
它们全部都是单独可停靠的，也可以是自由浮动的。如果某个窗口自由浮动，可以单独调整它。如果某个窗口被固定显示，则对它进行调整会影响附近的固定窗口。如果多个窗口被固定在同一个位置，则将在窗口的底部显示一组选项卡，列出所有可用的窗口。默认窗口设置的持续以及用于选择默认窗口的用户界面都由 PSoC Creator 框架控制。更多有关排序和调整窗口的信息，请参阅自定义框架部分。

下面列出了各个可用的调试器窗口：

- 存储器窗口
- 监视窗口
- 组件调试窗口
- 断点窗口
- 寄存器窗口
- 调用堆栈窗口
- 局部窗口
- 反汇编窗口

存储器窗口

存储器端口显示的是存储在处理器的存储器中的值。



根据目标架构定义，存储器被分为不同区域。窗口将显示一个包含了相关字节的地址列表以及一个表示被显示字节的 ASCII 字符串。当停止事件发生后，可随时更新该窗口。

某些存储器窗口可以同时存在，并且每个窗口针对存储器中的不同区域。通过该工具窗口，即使没有任何特殊的滤波器或格式，仍能够轻松查看芯片上的原始数据。它允许快速浏览大量数据块，以进行验证。另外它也可能是浏览不显示在其他窗口中的芯片区域的唯一方法。

如何显示存储器窗口：

调试器必须处于运行模式或中断模式。

从 **Debug** 菜单，请依次选择 **Windows > Memory**，然后点击 **1, 2, 3,**或 **4**。

右键菜单：

左键单击该窗口后，可以选择下面各选项：

- **Columns**（列）— 更改显示的数据列数量。默认值为 8。
- **Data View**（数据浏览）— 更改数据的格式。默认值为 1 字节整数。
- **Data Style**（数据类型）— 更改显示基数。默认为十六进制。
- **ASCII Display**（显示 ASCII 码）— 用于切换是否显示 ASCII 码列。

要想跳转到某个特定地址：

请将地址输入到“Address”字段内，并按下[**Enter**]键。

您同样可以跳转到一个具体的函数或变量地址上。

要想改变地址空间：

从下拉菜单选择相应的地址空间。

要想编辑存储器中的内容：

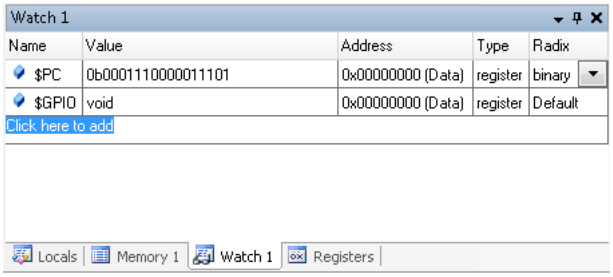
请点击一个条目以使能编辑模式，然后键入所需的值并按下[**Enter**]键。

另外，请参考：

- 调试器的使用

监视窗口

使用监视窗口进行评估和显示各个变量、寄存器或表达式。它只会显示您具体要求的项目。



通过选择文本编辑器中的文本块，或将表达式直接键入到监视窗口中的 **Name**（名称）列内，即可将各项目添加到监视窗口。

如何使用监视窗口进行评估各种表达式：

- **Simple variables**（简单的变量）— 只需要键入变量名称。例如，如果程序包含'`uint16 foo = 3;`'，则只需要键入'`foo`'。
- **Array variables**（数组变量）— 只需要键入索引周围带有括号的变量名称。例如，如果程序包含'`uint8[10] foo;`'，则只需要键入'`foo[2]`'。
- **Struct variables**（结构变量）— 只需要键入带有句号与其他项目隔离的变量名称。例如，如果程序包含'`struct foo { uint16 a; uint8 b }; struct foo temp;`'，请键入'`foo`'来查看整个结构，或键入'`foo.a`'以查看结构的‘一个’成分。
- **Registers**（寄存器）— 将‘$’符号输入到寄存器名称的前面。例如，要查看程序计数器，请输入'$PC'。通过打开寄存器窗口，可以查看所有寄存器名称。
- **Expression**（表达式）— 输入表达式以进行评估。例如，要在某个变量中输入一个数字，请输入'`foo+3`'。
- **Assignments**（分配）— 键入所分配的语句以进行评估。例如，要将 3 分配给变量 foo，请输入'`foo=3`'。

每次发生停止事件时，都会自动使用最新的值来更新监视窗口。停止时，您可以修改各项的值，以查看并更新设计中您感兴趣的任何方面。对于数字值，还存在一个选项可将数字显示为二进制、八进制、十进制或十六进制的值。某些监视窗口可以同时存在，并且每个窗口针对程序的不同方面。每个监视窗口包含以下各列：

- **Name**（名称）— 在该列中，您可以键入任何由调试器识别的有效表达式。
- **Value**（值）— 调试器会评估显示在 Name 列中的表达式，并将结果放置在该列中。
  - 如果表达式是一个变量或寄存器名称，那么您可以编辑该列中的值，以更改该变量或寄存器的值。
  - 您不能编辑常数变量的值。
  - 您可以编辑并显示本机代码所应用的寄存器值。
  - 通过右键点击 Watch 窗口并从快捷菜单选择 Hexadecimal Display（十六进制显示）选项，可以将数字的格式改为十六进制。
- **Address**（地址） — 显示存储器中的变量地址。
- **Type**（类型） — 该列显示的是变量或表达式的数据类型。
- **Radix**（基数）— 指的是 Value 的显示方式。

显示 Watch 窗口：

调试器必须处于运行模式或中断模式。

从 **Debug**（调试）菜单中选择 **Windows > Watch**，并点击 **1**、**2**、**3** 或 **4**。

右键菜单：

右击该窗口后可以选择下面选项：

- **Copy**（复制） — 将选定项复制到剪贴板内。
- **Paste**（粘贴） — 使用剪贴板中的内容，创建一个新输入项。

- **Edit Value**（更改数值） — 用于更改输入值。另外，双击 Value 字段也可以更改此值。
- **Add Watchpoint**（添加观察点） —将新的观察点变量添加到所选变量中
- **Delete Watch**（删除监视项） —删除列表中选定的输入项
- **Select All**（全部选择） —选择窗口中的所有输入项
- **Clear All**（全部清除） — 清除窗口中的所有输入项
- **Radix**（基数） — 更改调试器使用的默认基数。
- **Collapse Parent**（收缩父目录） — 收缩输入的父目录，子节点将被隐藏起来

如何输入监视项：

1. 点击 **Name** 列中的“Click here to add”（点击此处以添加新项）处。
2. 输入所需的监视项，并按下[**Enter**]。

如何修改监视项：

执行下面两个操作的其中一个：

- 通过双击输入项的 **Name** 字段，可以更改被监视的对象。
- 通过双击某个输入项中的 **Value** 字段（若不是只读字段），可以更改它的值。

如何更改基数显示：

要想更改所有行，请右击并选中 **Radix**，然后点击所需的显示类型。

要想更改单行，请双击此行，然后从下拉菜单中选择所需的显示类型。

如何删除监视项：

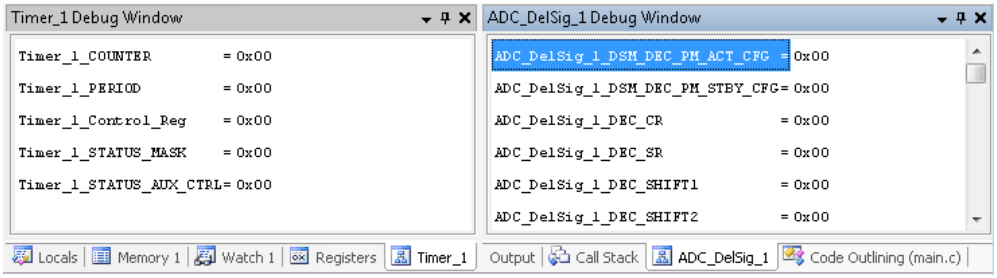
选择所需的项，并按[**Delete**]键。

另外，请参考：

- 调试器的使用
- 文本编辑器
- 寄存器窗口

组件调试窗口

Component Debug（组件调试）窗口用于查看有关设计中的组件的调试信息。



通过该窗口，您很容易便能看到组件的构成数据。这样有助于跟踪自定义组件的问题，并分析组件不正常运行的原因。

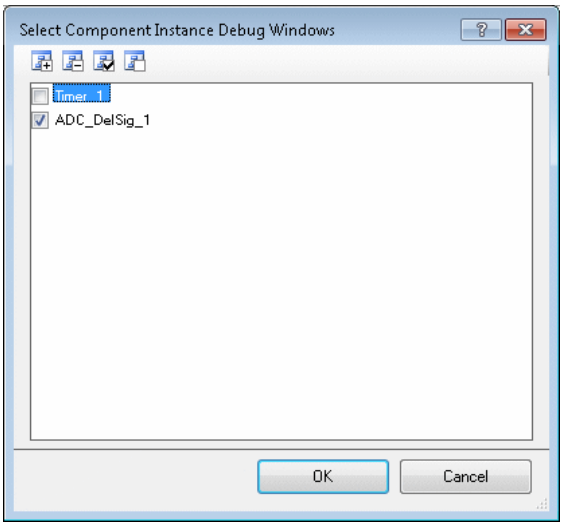
使用选择组件实例的调试窗口为选定项目中每个组件实例生成该窗口。每个组件窗口中列出了实例的存储器和寄存器，以及它们所使用的所有子组件。并不是所有组件都具有上面的三类数据，所以只会显示实际使用的资源。因此在该窗口内可能只存在存储器或寄存器视图。存储器和寄存器窗口的行为同主存储器窗口和寄存器窗口的行完全一样。为了便于访问，这些窗口提供了子组件列表的链接。

要打开该窗口：

调试器必须处于运行模式或中断模式。

1. 从 Debug 菜单中，依次选择 Windows > Components...

   Component（组件）窗口的 Selector（选型器）对话框将显示。



2. 在组件窗口中选择需要查看的组件实例，并点击 **OK**。

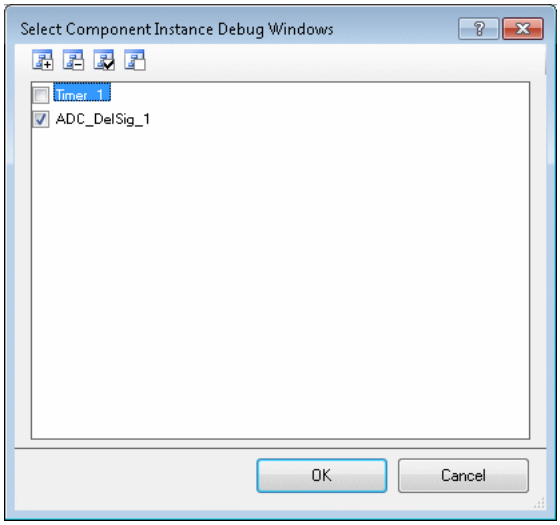
选定的组件调试窗口在调试器框架中显示。您可以根据需要重新安排这些窗口在调试器框架中的显示。

另外，请参考：

- 选择组件实例调试窗口
- 存储器窗口
- 寄存器窗口

选择组件实例调试窗口

通过选择组件实例调试窗口对话框，您可以从包含调试信息的所有组件实例列表中选择某一项。



在调试会话期间，每个选定的实例将显示在不同的组件窗口内。该窗口提供了一个简单的包含所有可用组件的树形视图，它保留了组件的潜入层次。它也清楚地显示了所构成的组件。

如何选择/取消选择组件窗口：

通过选择/取消选择每个输入项旁边的选框，可以分别显示或关闭窗口。

若需要，可以使用 **Select All** 或 **Deselect All**。

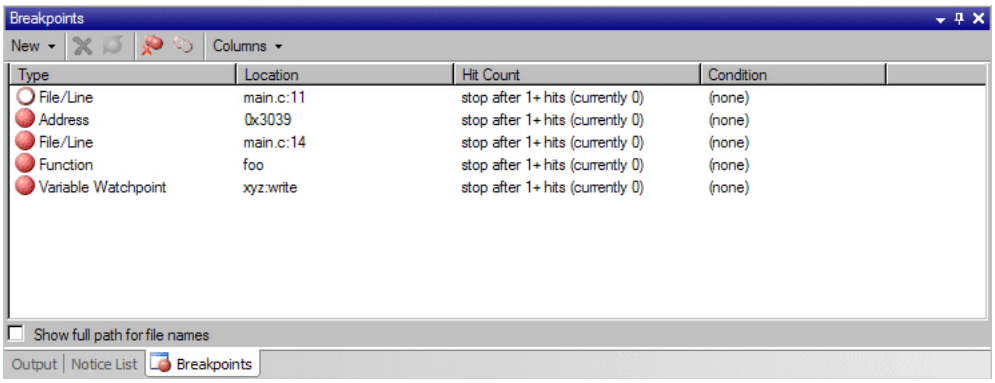
另外，请参考：

- 组件窗口

断点窗口

断点窗口

断点窗口用于显示在工作区中所设置的所有断点，不管断点的状态如何。该窗口显示了包含类型和地址信息的断点列表。如果处理器（调试工具链）支持该窗口，该窗口还会提供有关断点的其它信息，如断点被触发的次数和触发断点的真条件。



断点用于使调试器在到达断点的位置时停止下一次操作。可以对断点进行设置、列举、禁用或删除等操作。可通过不同的方式设置断点，如指定在文件中需要设置断点的代码行编号、要设置断点的函数或要访问的地址。

如果您设置一个硬件断点，则在 **Type**（类型）列中以及与断点所在的行相应的源/反汇编代码的左边缘将出现一个红色小点。禁用的断点处会出现一个红色圆圈。当删除断点时，相应的点/圆圈也被删除。要了解所有调试器指示符的列表和说明，请参阅调试器指示符。

PSoC 器件的可用断点数量是有限的。请参考相应的器件数据手册，了解可用的断点数量。PSoC Creator 保留了一个断点，用于执行其它操作，如单步、跳线和执行到光标处等。当您使用了全部可用的硬件断点数量时，如果想要添加其它断点，调试器将显示一条信息，通知您可用的断点数量已经用完。该信息会提示您使用软件断点。如果当前操作不要求 PSoC Creator 使用硬件断点，则第一个断点将自动使用可用的硬件断点。

注意：使能中断时，各断点被执行两次。这是因为该断点代码被执行前生成了一个中断并已经被处理。该中断执行完成时，处理器将返回原始代码行。这样会再次执行该断点代码。

如何打开断点窗口：

请依次点击 **Debug > Windows > Breakpoints**。

指令：

断点窗口包含下面的工具栏指令：

图标	指令	说明
	New（新）	用于创建不同断点类型的下拉菜单。
	Delete（删除）	删除选定的断点。
	使能/禁用断点	交替使能和禁用已选定的断点。
	Delete All Breakpoints（删除所有断点）	删除工作区内的所有断点，而不要单独移除每一个断点。如果您设置了多个断点，但您想要处理器一直运行，那么该指令将非常有用。
	启用/禁用所有断点	交替启用和禁用工作区中的所有断点。使用该指令快速更改所有断点的状态，而无需逐个更改每一个断点的状态。如果您设置了多个断点，但您想要处理器一直运行，那么该指令将非常有用。
	列	下拉菜单用于显示/隐藏窗口中的各列。
	显示文件名称的整个路径	交替显示/隐藏文件/行断点的全路径名。

如何设置断点：

1. 请打开您想调试的文件。
2. 在打开文件的左边上，请单击目标点来设置断点；再次点击它可以取消设置。

如何修改一个现有的断点：

通过右键点击现有的断点进行访问以下指令，您可以修改断点：

- **Delete**（删除） — 删除已选定的断点。
- **Enable/Disable**（使能/禁用） — 交替使能和禁用已选定的断点。
- **Temporary**（临时） — 设置断点作为临时的。
- **Location**（位置） — 打开断点对话框中的选定类型进行修改位置。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。
- **Hardware**（硬件） — 交替设置断点由硬件处理或者由软件处理。

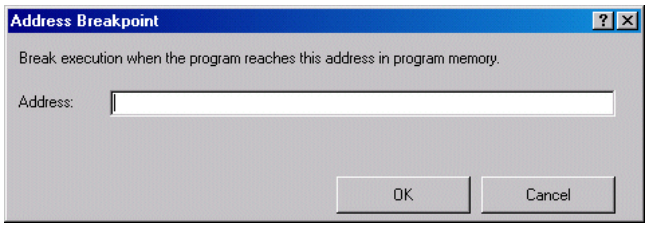
  注意： 软件断点（绿点）的处理方式同硬件断点的处理方式不相同。调试时软件断点位于工作区的任何位置，会导致程序比仅存在硬件断点时运行较慢。有关更多信息，请参考软件断点。

另外，请参考：

- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Memory Watchpoint（存储器观察点）
- Breakpoint Condition（断点条件）
- Breakpoint Hit Count（断点命中计数）
- Software Breakpoints（软件断点）

Address Breakpoint（地址断点）

Address Breakpoint 对话框用于在指定的地址上创建一个新的断点（或修改现有的断点）。



每次程序计数器（PC）与地址相匹配时，会命中该断点。它不需要将有关源代码编译以生成程序数据的任何信息。

只有在调试模式中，您才会看到地址断点被显示在源编辑器的边距指示器中。只有在调试模式下，这些断点类型才被显示在边距内。您可以在 Breakpoints 窗口中查看这些断点。

如何打开对话框：

新断点

对于一个新断点：

- 在 PSoC Creator 调试菜单中，请依次选择 New Breakpoint > Address Breakpoint...
- 在 Breakpoints 窗口中的 **New**（新）菜单下，选择 **Address...**

现有的断点

对于一个现有的断点，在 Breakpoints 窗口中请右键点击断点，然后选择 **Location...**

如何创建新的断点：

将十进制地址或十六进制地址输入到“break on”字段内，并点击 **OK**。

如何修改一个现有的断点：

通过在 Breakpoints 窗口中右键点击现有的断点进行访问以下的指令，您可以修改断点：

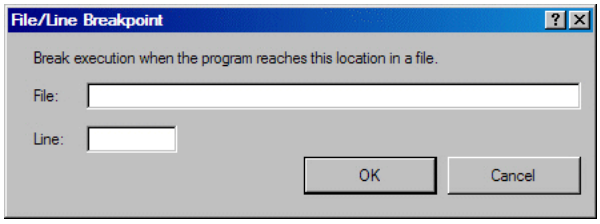
- **Temporary**（临时） — 设置断点作为临时的。
- **Location**（位置） — 打开 Address Breakpoint 对话框，进行修改地址。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。

另外，请参考：

- Breakpoints 窗口
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Condition（断点条件）
- Breakpoint Hit Count（断点命中计数）

File/Line Breakpoint（文件/行断点）

使用 File/Line Breakpoint 对话框在源代码的指定文件行中创建一个新的断点（或修改现有的断点）。



每次执行代码行时，都会命中该断点。这是最方便的方法，但精度也最小（添加一个断点时）。

如何打开对话框：

新断点

对于新断点，在 PSoC Creator **调试**菜单中，依次选择 **New Breakpoint > File/Line Breakpoint...**

现有的断点

对于一个现有的断点，在 Breakpoints 窗口中请右键点击断点，然后选择 **Location...**

如何创建新的断点：

在代码编辑器的 **Indicator Margin**（指示器边距）中，请点击需要创建断点的行边距。

注意：您还能使用这个对话框来输入代码行编号和要放置断点的文件全名，并点击 OK。

如何修改一个现有的断点：

通过在 Breakpoints 窗口中右键点击现有的断点进行访问以下的指令，您可以修改断点：

- **Temporary**（临时） — 设置断点作为临时的。
- **Location**（位置） — 打开 Address Breakpoint 对话框，进行修改地址。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。

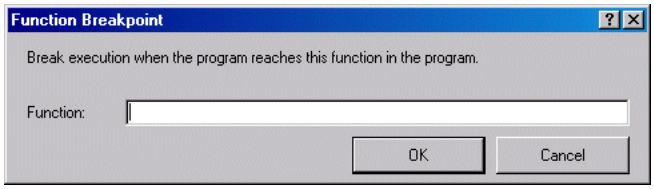
另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Condition（断点条件）

- Breakpoint Hit Count（断点命中计数）

Function Breakpoint（函数断点）

Function Breakpoint 对话框用于在程序中指定的函数上创建一个新断点（或修改现有的断点）。



通过使用函数断点，可以创建断点；函数被调用时，并不取决于代码内部和周围函数的变化，该断点会被命中。

只有在调试模式下，您才会看到函数断点被显示在源编辑器的边距指示器中。只有在调试模式下，这些断点类型才被显示在边距内。您可以在 Breakpoints 窗口中查看这些断点。

注意：函数断点未显示在指示器边距中。函数带有参数时，Keil 预设函数名的前缀为“_”。否则函数断点不起作用。为了解决这个问题，函数名称的前缀必须为‘_’。

要打开该对话框：

新断点

对于一个新断点：

- 在 PSoC Creator 调试菜单中，请依次选择 New Breakpoint > Function Breakpoint...
- 在 Breakpoints 窗口中的 **New**（新）菜单下，选择 **Function...**

现有的断点

对于一个现有的断点，在 Breakpoints 窗口中请右键点击断点，然后选择 **Location...**

如何创建新的断点：

将函数名称输入到“break on”字段，并点击 **OK**。

如何修改一个现有的断点：

通过在 Breakpoints 窗口中右键点击现有的断点进行访问以下的指令，您可以修改断点：

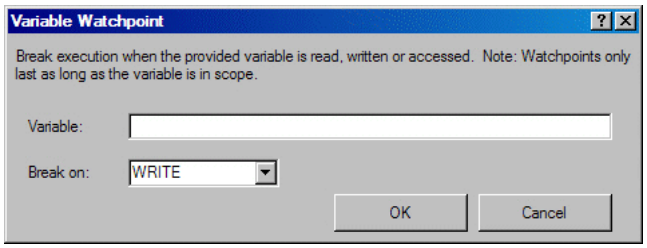
- **Temporary**（临时） — 设置断点作为临时的。
- **Location**（位置） — 打开 Function Breakpoint 对话框，进行修改函数。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。

另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Condition（断点条件）
- Breakpoint Hit Count（断点命中计数）

Variable Watchpoints（变量观察点）

Variable Watchpoint 对话框用于创建指定变量中的观察点（或修改现有的观察点）。



观察点用于设置数据存储器中放置的断点（与标志断点不同的是标志断点用于程序存储器）。不用在一行代码上设置断点，您只要在代码中的变量上设置它便可。

每次对变量所在的地址进行读取、写入或访问（根据 **Break on** 字段中的内容）操作时，则该观察点将被命中。使用它来跟踪指定的变量/地址被读取或写入的时间。这样可以帮助您分析特定的存储器地址不含有预计值的原因。与标准的断点相同的是命中时观察点会使程序停止执行，并且指示器将被显示用于指出用指令触发的观察点。

同标准断点（与代码内具体指令相关）不同的是观察点可以被命中在任意指令内。这完全取决于指令访问的数据。因此，观察点在文本编辑器的左侧边缘上没有显示任何指示器。但，它们仍被列在 Breakpoint 窗口中。

注意：有效的观察点数量取决于正在调试的器件。有关更多信息，请参考适用的器件数据手册。

如何打开对话框：

新的观察点

对于新观察点：

- 在 PSoC Creator 调试菜单中，请依次选择 New Breakpoint > Variable Watchpoint...
- 在 Breakpoints 窗口中的 **New**（新）菜单下，选择 **Watch Variable...**

现有的观察点

对于现有的观察点，在 Breakpoints 窗口中请右键点击观察点，然后选择 **Location...**

要创建新的观察点：

将变量名称输入到“break on”字段，并点击 **OK**。

如何修改一个现有的断点：

通过在 Breakpoints 窗口中右键点击现有的断点进行访问以下的指令，您可以修改断点：

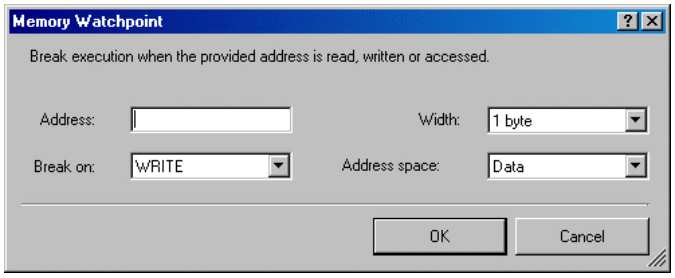
- **Location**（位置） — 打开 Variable Watchpoint 对话框，进行修改地址。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。

另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Memory Watchpoint（存储器观察点）
- Breakpoint Condition（断点条件）
- Breakpoint Hit Count（断点命中计数）
- Software Breakpoints（软件断点）

Memory Watchpoint（存储器观察点）

Memory Watchpoint 对话框允许您选择地址、大小、地址空间及中断类型。



存储器观察点与变量观察点一样，但可以将它们设置在特殊存储器地址上，而不是变量上。模拟路由编辑器使用存储器观察点来检测何时打开或关闭开关。然而，存储器观察点可以用于其它目的，比如：监控堆栈或数据损坏。

该对话框包括下面各字段：

- **Address**（地址） — 地址用于设置断点。（请注意，根据指定的大小，地址的底部 n 位会被忽略）。
- **Width**（大小） — 是用来设置断点的字节数量。通过忽略地址中的底部 n 位实现。

- **Break On** — 是观察点被触发的存储器访问类型。
- **Address Space**（地址空间） — 如果器件有多个存储器空间，将选择该字段。

要打开该对话框：

新的观察点

对于新观察点：

- 在 PSoC Creator 调试菜单中，请依次选择 New Breakpoint > Memory Watchpoint...
- 在 Breakpoints 窗口中的 **New**（新）菜单下，选择 **Watch Memory...**

现有的观察点

对于现有的观察点，在 Breakpoints 窗口中请右键点击观察点，然后选择 **Location...**

要创建新的观察点：

将十进制地址或十六进制地址输入到“break on”字段，并点击 **OK**。

要修改一个现有的断点：

通过在 Breakpoints 窗口中右键点击现有的断点进行访问以下的指令，您可以修改断点：

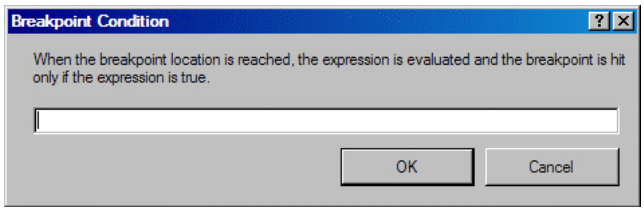
- **Location**（位置） — 打开 Memory Watchpoint 对话框，进行修改地址。
- **Condition**（条件） — 打开 Breakpoint Condition（断点条件）对话框。
- **Hit Count**（命中计数） — 打开 Breakpoint Hit Count（断点命中计数）对话框。

另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Condition（断点条件）
- Breakpoint Hit Count（断点命中计数）
- Software Breakpoints（软件断点）

Breakpoint Condition（断点条件）

Breakpoint Condition 对话框用于添加/修改断点上的条件。



这样您能够控制调试器将实际报告断点已被命中的时间。通过使用断点条件，代码被执行任意次，并不满足条件。它提供识别问题的方法比每次停止然后手动评估条件是否满足的方法更快。

要打开该对话框：

在 Breakpoints 窗口中，请右键点击断点，然后选择 **Condition...**

要设置断点条件：

代码被终止前，请输入必须评估为满足的条件，并点击 **OK**。

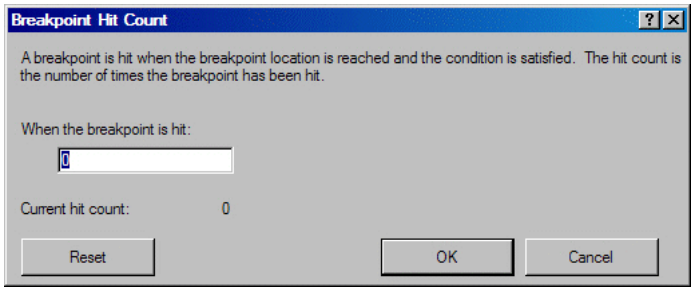
例如：`i>10` 或 `a==b`

另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Hit Count（断点命中计数）

Breakpoint Hit Count（断点命中计数）

Breakpoint Hit Count 对话框用于设置实际停止前必须经历的断点次数。



当运行到断点位置并满足各条件时，将命中断点。 命中计数是指断点被命中的次数。这样您能够控制调试器将实际报告断点已被命中的时间。通过使用它，在实际触发断点（用于用户分析）前代码可以执行几次。这样有助于快速检查边界条件，并确定代码执行的次数是否比预计的多。

要打开该对话框：

在 Breakpoints 窗口中，请右键点击断点，然后选择 **Hit Count...**

要设置断点命中计数：

输入被终止前断点被命中的次数，然后点击 **OK**。

点击 **Reset**（复位），以进行复位断点已经被命中的次数。

另外，请参考：

- Breakpoints 窗口
- Address Breakpoint（地址断点）
- File/Line Breakpoint（文件/行断点）
- Function Breakpoint（函数断点）
- Variable Watchpoints（变量观察点）
- Breakpoint Condition（断点条件）

Software Breakpoints（软件断点）

一旦用完通过目标设备提供的所有硬件断点，则软件断点便作为一个主机 PC 端实现的断点。

软件断点使 PSoC Creator 单步执行处理器，并将 PC 与断点地址进行比较。如果这两者匹配，PSoC Creator 会报告已经命中了一个断点。如果 PC 与断点地址不匹配，它会继续单步调试代码。因为处理器单步调试代码（而不是仅运行），所以它变得非常慢。

如何使用软件断点：

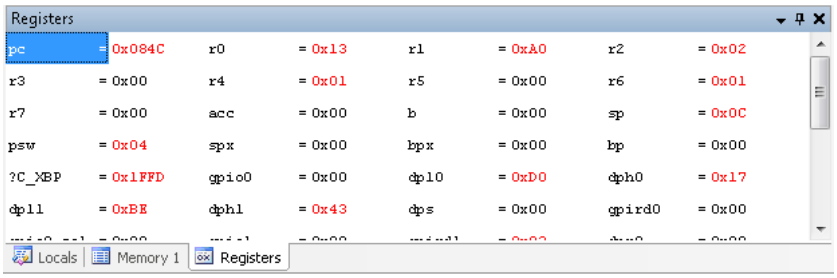
一旦目标器件用完了所有硬件断点，PSoC Creator 便会自动使用软件断点进行触发所有将来的断点。根据 Debugging Options 对话框中的设置，工具可以警报有关使用软件断点的问题。如果删除了一些硬件断点，软件断点会被转换为硬件断点，以提高调试性能。

另外，请参考：

- Breakpoints 窗口
- 调试选项

Registers（寄存器）窗口

Registers 窗口显示内核 CPU 寄存器和它们的值。



它提供快速查看处理器的内核中发生问题的方法。它还允许修改您认为合适的任何寄存器值，以确保处理器正在执行预计的任务。

每次发生终止时，该窗口都会被更新。该窗口中显示的值默认显示为十六进制格式。

要显示 Registers 窗口：

调试器必须处于运行模式或中断模式。

从 **Debug** 菜单中，依次选择 **Windows > Registers**。

右键菜单：

右击该窗口后可以选择下面选项：

- **Copy**（复制） — 将选定的项复制到剪贴板。
- **Select All** （选择全部） — 选择显示屏上的所有寄存器。
- **Radix**（基数） — 更改调试器使用的默认基数。

要编辑寄存器的值：

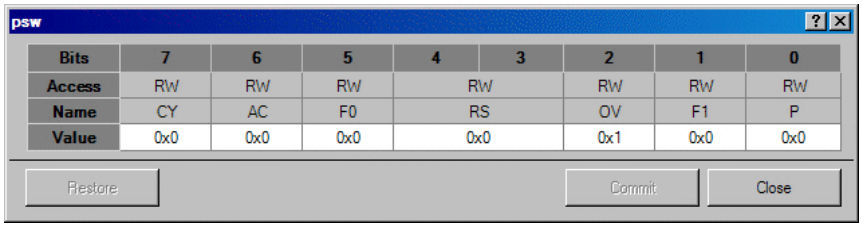
请点击一个条目以使能编辑模式，然后键入所需的值并按一下**[Enter]**键。

另外，请参考：

- 调试器的使用

寄存器的详细信息

使用 Register Details 对话框您可以了解有关指定寄存器的更多信息。 该对话框显示寄存器内的每个字段、访问权限和该字段设置的值。



另外，将鼠标悬停在某个字段，调试器会提供工具提示，该工作提示描述字段并在某些情况下表示具体设置的意义。

要打开对话框：

请在 Registers 窗口中点击寄存器，或右键点击寄存器，然后选择 **Details...**

要修改某个值：

请点击您要修改的字段中的单元值，设置新值，然后单击 **Commit** 按键。

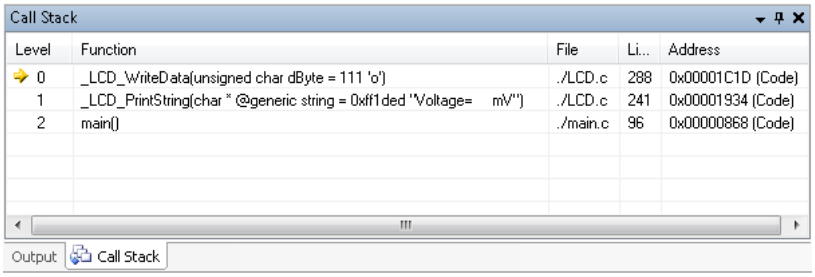
点击 **Restore**（恢复）按键，以恢复当前在芯片上的寄存器的值。

另外，请参考：

- Registers（寄存器）窗口
- Debugger（调试器）窗口

Call Stack（调用堆栈）窗口

Call Stack 窗口用于跟踪目标程序调用不同函数的顺序。



将没完成调用的函数放置在堆栈的顶部。当完成调用函数时，不会再弹出它。此工具有助于查看如何执行代码，并确保一切都以正确的顺序进行。它提供了一种简单方法用于正确跟踪程序如何执行代码的当前行。

该窗口显示了每个函数的名称，以及有可选的信息（比如：代码行编号、字节偏移等）。可以使能或禁用该可选信息的显示功能。

要想显示调用堆栈窗口：

调试器必须处于运行模式或中断模式。

从 **Debug** 菜单中，选择 **Windows** 和点击 **Call Stack**。

右键菜单：

右击该窗口后可以选择下面选项：

- **Copy**（复制） — 将选定的项复制到剪贴板。
- **Select All**（选择全部） — 选择显示屏上的所有寄存器。
- **Switch to Frame**（切换到帧） — 更改可视的项目，同双击一个项一样。
- **Run to Frame**（运行到帧） — 引起代码执行，直到选定的帧得到执行为止。
- **Show Module Name**（显示模块名称） — 显示函数所存在的文件

- **Show Parameter Type**（显示参数类型） — 显示每一个函数参数的类型
- **Show Parameter Name**（显示参数名称） — 显示每一个函数参数的名称
- **Show Parameter Value**（显示参数值） — 显示每一个函数参数的值
- **Show Line Number**（显示代码行编号） — 显示执行调用的函数中的代码行编号
- **Show Byte Offset**（显示字节偏移） — 显示执行调用的指令地址

要修改被显示的可选信息：

右键点击 Call Stack 窗口，然后点击需要的信息，以显示或隐藏快捷菜单中的这些信息。

如何修改活动的调用堆栈项/帧：

双击然后输入或右键点击并选择 **Switch to Frame**。

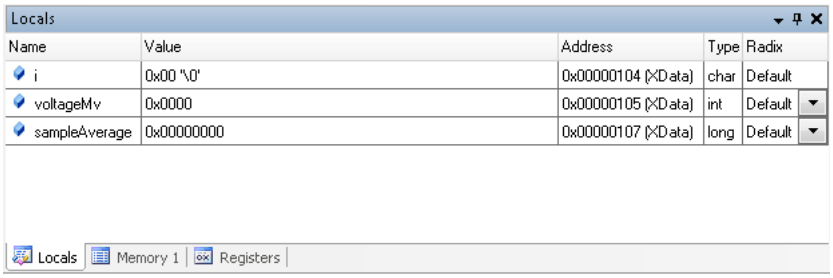
这样将使焦点移位到与点击框架的编辑器相应的代码行。另外，调试器也会更改有效框架，从而使本地窗口更新已选框架中的本地变量。

另外，请参考：

- 如何使用调试器
- Locals（本地）窗口

Locals（本地）窗口

通过本地窗口您可以查看并修改当前调试框架中的所有本地变量。



您可以看到每个变量的值、类型和地址。这样您能够快速查看一个函数是否按预计的情况运行，并能查看是否相应更新了它的变量。

该窗口包含以下各列：

- **Name**（名称）：包含当前范围的所有本地变量名称。结构变量和数组变量具有可用于显示或隐藏元素的树形控制。
- **Value**（值）：显示每个变量的值。默认情况下，整数变量以十六进制格式显示。通过右键点击本地窗口，然后从快捷菜单中选择 **Decimal Display** 项，可以将变量值的格式改变为十进制。
- **Address**（地址）：显示变量在存储器中的位置。
- **Type**（类型）：指出名称列中每个变量的数据类型。

- **Radix**（基数）：指的是 Value（值）的显示方式。

如何显示本地窗口：

调试器必须处于运行模式或中断模式。

从 **Debug**（调试）菜单中，选择 **Windows**，然后点击 **Locals** 项。

默认显示的内容是包含当前执行位置的函数。

右键菜单：

右击该窗口后可以选择下面选项：

- **Copy**（复制） — 将选定的项复制到剪贴板。
- **Edit Value**（更改数值） — 用于更改输入值。另外，双击 Value 字段也可以更改此值。
- **Add Watchpoint**（添加观察点） —将新的变量观察点添加到所选的变量。
- **Select All**（全部选择） —选择窗口中的所有输入项。
- **Radix**（基数） — 更改调试器使用的默认基数。
- **Collapse Parent**（折叠父目录） — 折叠输入父目录，子节点将被隐藏。

如何更改变量：

通过双击某个输入的 **Value** 字段（若不是只读字段）可以更改它的值。

如何更改基数显示：

要想更改所有行，请右击并选中 **Radix**，然后点击所需的显示类型。

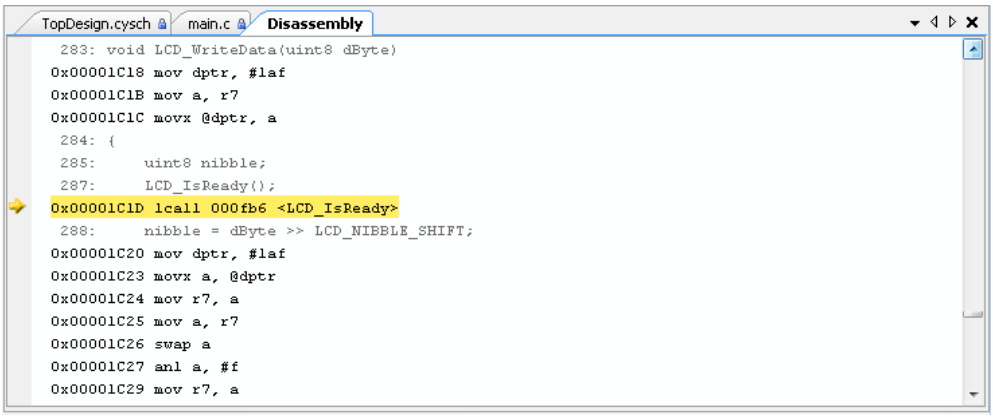
要想更改单行，请双击此行，然后从下拉菜单中选择所需的显示类型。

另外，请参考：

- 如何使用调试器
- Locals（本地）窗口

Disassembly（反汇编）窗口

Disassembly 窗口显示源代码中所创建的基本指令。



为了避免读取二进制或十六进制格式的指令代码，这些指令将被拆解成汇编语言的格式。这样您可以查看和输入由用户源代码组成的断点，并可以通过汇编各指令了解代码正在执行的操作。

使用 Disassembly 窗口时，您可以使用所有与调试 C 源代码时相同的性能进行单步调试汇编代码，设置断点并交互代码。也可以使用混合模式拆解性能，如拆解 C 代码。

汇编语言代码包括用于指令命名的缩略词助记符以及代表变量、寄存器和常量的符号。一个汇编语言助记符代表一个机器语言指令，它的后面是一个或多个变量、寄存器或常量。由于汇编代码主要依赖于处理寄存器（对于受控代码，将依赖于通用语言运行寄存器），Disassembly 窗口与 Registers（寄存器）窗口一起使用将有助于检查寄存器内容。

要显示 Disassembly 窗口：

调试器必须处于运行模式或中断模式。

从 **Debug（调试）**菜单中，依次选择 **Windows > Disassembly**。

右键菜单：

右击该窗口后可以选择下面选项：

- **Insert Breakpoint** — 将地址断点插入到当前地址
- **Break Here Once** — 在当前地址中插入一个临时地址断点
- **Run to Instruction** — 运行处理器直到达到当前地址
- **Go to PC** — 设置窗口的重点地址为当前 PC 地址
- **Go to Address** — 设置窗口的重点地址为所提供的地址
- **Show source lines when available** — 混合用户源 和汇编代码
- **Hide empty source lines** — 隐藏空白行
- **Show code bytes** — 显示指令的代码数据
- **Copy** — 将选定的项复制到剪贴板。

要插入一个断点：

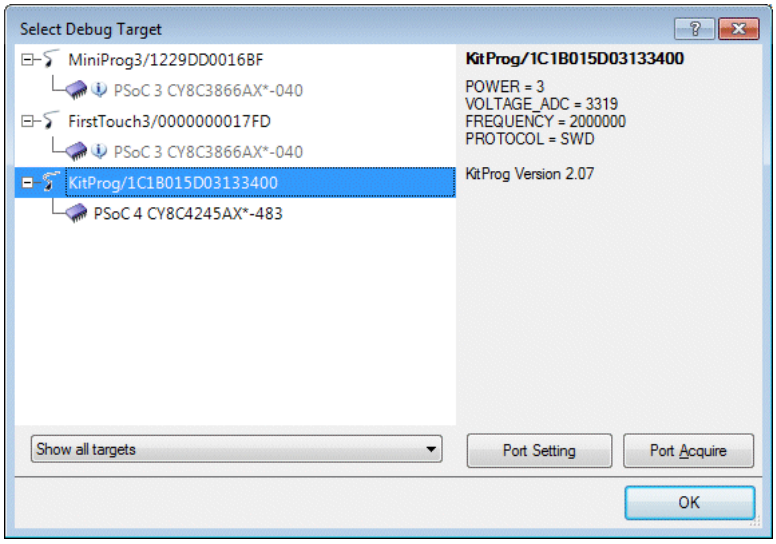
点击代码行旁边的容限。再次点击以删除它。

另外，请参考：

- 调试器的使用
- 寄存器窗口

选择调试目标

通过 Select Debug Target 对话框可以选择您需要编程和调试的连接器件。该对话框还支持配置用于检测器件的设置。



要打开对话框：

打开一个项目，然后从 **Debug** 菜单中选择 **Select Debug Target** 项。

首次点击 **Debug**（调试）项或 **Program**（编程）项时，也会出现该对话框。如果您没有或有多个器件连接到电脑，会显示该对话框。

它使用器件选择器所选择的器件类型作为您可选的调试目标。打开 PSoC Creator 时，调试器将该选择使用于所有后续调试会话。

如果您要选择其他调试目标，依次选择 **Debug > Select Debug Target** 以打开 Select Debug Target 对话框。

进行应用设置：

当没有配置通信通道的设置，或者这些设置被其他应用重新配置时，则使用 **Apply Settings** 按键来应用 Programmer/Debugger Options 对话框的设置。

要复位连接器件：

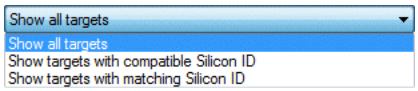
点击连接节点，然后点击 **Port Acquire**（端口获取）按键。

该项用于复位任何连接器件，并保证调试端口处于有效状态。

对器件显示进行滤波：

根据有效项目可以滤波所显示的器件。可以配置对话框显示的下述内容：

- 所有器件
- 与有效项目的所选器件兼容的器件
- 与有效项目的所选器件完全匹配的器件

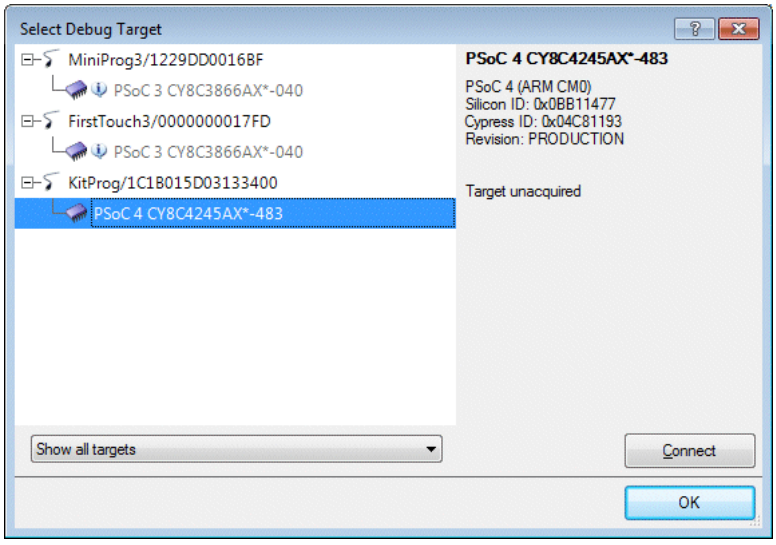


注意：为实现兼容，目标器件必须为相同的系列，并拥有项目器件的所有（或多个）资源。

要想选择器件：

器件可以通过 MiniProg3 或套件进行连接，并能够使用在 PSoC Creator 上。MiniProg3 和套件下面是其所连接的器件列表。

点击列表中的相应器件，然后点击 **Connect**，这样在编程或调试时可以使用该器件。



打开 PSoC Creator 时，调试器将该选择使用于所有后续调试会话。如果您要选择其他调试目标，请重新打开 Select Debug Target 对话框。

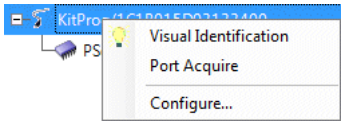
注意：该 Connect 指令会使所选器件只对 PSoC Creator 的此次会话有效。

如果芯片的调试功能无效，点击 **Port Acquire** 会使 PSoC Creator 正常读取芯片版本。该操作会复位器件并使其进入临时调试状态。在该调试状态下，不执行代码。

在运行 **Port Acquire** 后，如果 PSoC Creator 仍不能认识芯片版本，可能是因为 PSoC Creator 的版本太旧了，不能处理这个芯片版本。或者对于 PSoC Creator 的当前设置中，电路板的接线不正确。通过 Programmer/Debugger Options 对话框所选的选项可以验证电路板的接线情况。例如，如果 MiniProg3 可以使用 Reset for the Programming Mode 选项，则必须将 MiniProg3 的 XRES 引脚连接到 PSoC 的 XRES 引脚。

右键菜单：

右键点击树形目录将显示目录中节点可以使用的指令。例如，您可以使用这些指令来配置通信通道设置。

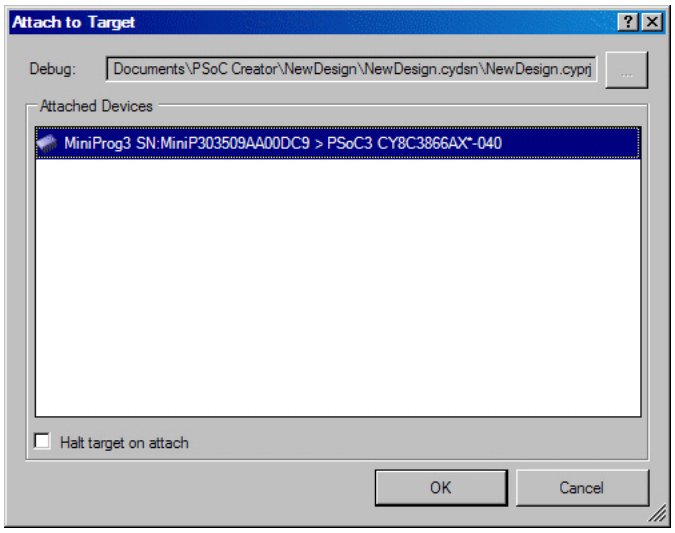


另外，请参考：

- 调试器的使用
- 器件配置
- MiniProg3
- QuickProgrammer
- Programmer/Debugger（编程器/调试器）选项

Attach to Target（连接到目标）

通过 Attach to Target 对话框可以连接到一个已经编程好的目标器件。该对话框仅对 PSoC 3 和 PSoC 5LP 器件可用。



这个对话框适用于调试已经运行一段时间并按预计停止运行的设计。您也可以使用该对话框来调试不存在源代码的设计。与标准调试流程不同，在调试器被打开前，该指令不会重新编程器件。

注意：使用该功能可瞬间中止 PSoC 器件的运行代码。

您可以选择项目文件、工作区文件或 elf 文件，并将其作为调试文件使用。如果选择某个项目或工作区，该对话框将在 PSoC Creator 中打开它。如果在 PSoC Creator 中已经打开了某个项目，则该字段将自动填充这个有效的项目。该字段不要求将调试器连接到目标器件，但是指定了一个文件允许在调试时提供等多的选项。

要打开对话框：

请从 **Debug** 菜单中选择 **Attach to Running Target...**项。

要选择某个文件：

请点击[**...**]可以导航到需要选择的项目、工作区或 ELF 文件。

如果在 PSoC Creator 中已经打开了一个项目，则该字段将自动填充这个有效的项目。浏览键被禁用。如果不存在有效项目，您可以选择某个文件。

该字段不要求将调试器连接到目标器件，但是指定了一个文件允许在调试时提供等多的选项。

要想选择一个器件：

点击连接至电脑的器件列表。从该列表中选择需要连接的器件。

要停止目标连接（Halt Target on Attach）：

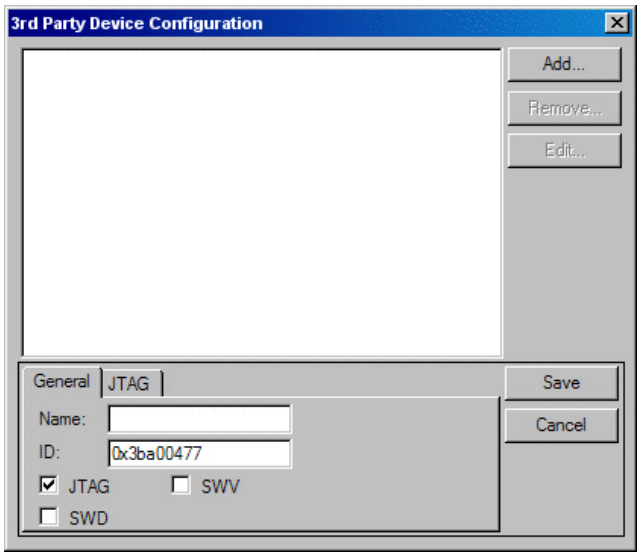
通过点击该复选框，调试器可以停止目标器件，并指出停止的位置。

另外，请参考：

- 调试器菜单指令
- 调试器的使用

Device Configuration（器件配置）

通过 Device Configuration 对话框，PSoC Creator 可以确定第三方器件，并报告器件名称而不仅仅是芯片 ID。



这样，选择调试目标对话框会列出与计算机相连器件的正确信息。

注意：当该配置允许 PSoC Creator 识别第三方器件时，不能将这些器件用于调试目的。Device Configuration 对话框的主要用途是用于配置第三方器件的指令寄存器和数据寄存器的大小。该器件附加在 JTAG 链路中的第三方器件内。

要查看器件配置：

通过下面两种方式都可以查看器件配置：

- 从 **Tools** 菜单中依次选择 **Options > Program/Debug > Device Recognition**。
- 从 Select Debug Target 对话框，右击器件节点并选择 **Configure** 项。

另外，请参考：

- 调试器的使用
- 选择调试目标
- MiniProg3
- 调试选项

MiniProg3

MiniProg3 适用于编成和调试 PSoC 器件。它提供下面各种传输协议：

- SWD
- JTAG
- I2C

如何编成或调试一个 PSoC 器件：

将 MiniProg3 配置为 SWD 模式或 JTAG 模式。

通过在 PSoC Creator 首选项中配置 MiniProg3 的默认配置，您可以配置所有 MiniProg3 器件（Program/Debug Options > MiniProg3）。在 Select Debug Target 对话框中右击连接 MiniProg3 器件可以对其进行配置。这两种配置选项都会影响到所有连接 MiniProg3s 器件。

注意：如果配置了 MiniProg3 的各选项，使 MiniProg3 向器件提供电压，并且该器件已经由外部电源供电，那么各设置将不会与 MiniProg3 正在进行的操作相匹配。

如何使用 MiniProg3：

保证它被正常连接。

1. 使用所提供的 USB 电缆将 MiniProg3 连接到电脑上。
2. 使用 10 引脚带状电缆将 MiniProg3 连接到 DVK 版的处理器模块。

PSoC Creator 将自动检测最新连接的 MiniProg3 和 PSoC 器件。您可以在 Select Debug Target 对话框的 MiniProg3 连接类型下查看这两种器件。

另外，请参考：

- Program/Debug（编程/调试）选项
- 选择调试目标
- 器件配置

QuickProgrammer

QuickProgrammer 是一种允许 PSoC Creator 需要时可以自动调用 PSoC Programmer 的方法。它节省 PSoC Programmer 的手动启动和配置的时间。可以通过两种不同的方法使用 QuickProgrammer。

- 第一种方法是点击 **Program** 按键。 

  点击后将对当前项目进行编译，并且检查所选的器件是否与项目中所指定的器件相兼容。然后，PSoC Creator 会调用 PSoC Programmer 编程器件。

- 第二种方法是使用 **Debug** 按键。 

  除了上面所述的器件编程内容外，该方法还启动了一个新的调试会话。

另外，请参考：

- 调试器的使用
- 调试器的工具栏指令
- 调试器菜单指令

错误处理

本节将详细说明调试模块如何处理各种错误，如 USB 电缆被拔掉的错误。

多个调试器实例：

可以在一个电脑上运行多个 PSoC Creator 实例。通过使用 USB 端口和/或 JTAG DUT 的锁定机制和锁定检测，您能够了解 PSoC Creator 调试器实例拥有专用 USB 器件和/或 JTAG DUT 的时间。

外部中止或复位硬件目标：

在 PSoC Creator 中，可以独立中止或复位硬件目标。PSoC Creator 会通过源代码或反汇编代码中的断点指示器，另外状态栏信息会通知器件被复位/中止。

如果硬件被外部中止，调试器将通知您（就像您按下 **Break** 按钮一样。如果复位，它的运行情况将与断开连接的情况相同。

硬件断开连接：

如果硬件在调试会话过程中断开与电脑的连接，则在下次尝试与芯片进行通信时，IDE 会通过一个对话框通知您。

9 完成项目

下面显示的是完成 PSoC Creator 项目的各步骤：

- 检查器件数据手册
- 优化编译器设置
- 存档项目
- 设置编译配置
- 选择编程协议
- 使能器件保护功能
- 选择可选的复位行
- 选择闪存的安全保护
- 使能一次性写锁存器闪存保护
- 评估通用的编程选项

检查器件数据手册

在进行生产前，要根据 PSoC 器件数据表中的详细布局对应用的引脚选择、PCB 布局以及硬件设计进行最终检查，该过程非常重要。在 www.cypress.com 网站 工作区浏览器下的 **Datasheets** 选项卡上可以找到准确的 PSoC 器件数据手册。

器件数据手册总结了器件的特性、电气特性、引脚布局、器件级规范以及外设电气规范。常见错误包括编成接口、复位行、电源连接以及相关的电压连接布局中发生的错误。更多信息，请参考 AN61290 — PSoC 3 和 PSoC 5LP 硬件计的注意事项。

当您使用支持功能齐全器件的开发套件来开发设计，然后将该设计移动到更小的 PSoC 封装时，您也可以试验固件和硬件的不匹配。该应用适用于在更小的 PSoC 器件上不能使用的引脚或特性。如果该应用是在其他 PSoC 封装上的

另外，请参考：

- 完成项目
- Workspace Explorer（工作区浏览器）

- AN61290

优化编译器设置

当您从开发移植到生产时，项目设计需要高级压缩代码，使大型应用符合 PSoC 闪存空间。您需要检查编译器安装所提供的编译器代码压缩文档。根据所选编译器，导航到以下路径的其中一个：

- <INSTALL_DIR>\PSoC Creator\import\gnu_cs\arm\4.4.1\share\doc\
- <INSTALL_DIR>\PSoC Creator\import\keil\pk51\9.03\C51\hlp

注意：通过 PSoC Creator Help 菜单可以直接访问这些文档。

另外，请参考：

- 完成项目
- 帮助菜单

下载与存档开发工具

当您将设计移植到生产时，赛普拉斯建议您保存 PSoC Creator 和 PSoC Programmer 开发环境的备份。在您必须返回设计环境对生产项目进行修改或更新时，该操作非常有用。通过导航到 PSoC Creator 和 PSoC Programmer 网页并下载下载表中提供的 ISO 图像可以实现此操作。通过这些 ISO 图像，可以将存档软件内容刻录到 CD/DVD 内。

- http://www.cypress.com/go/psoccreator
- http://www.cypress.com/go/psocprogrammer

另外，请参考：

- 完成项目

存档项目

当开发项目移植到生产状态时，您可以使用 PSoC Creator Workspace/Project Archiver 来存档该项目。Archiver 工具支持文件容量、存档范围和压缩的许多选项。

当保存开发环境或将项目放置在源控制下时，该功能非常有用。

另外，请参考：

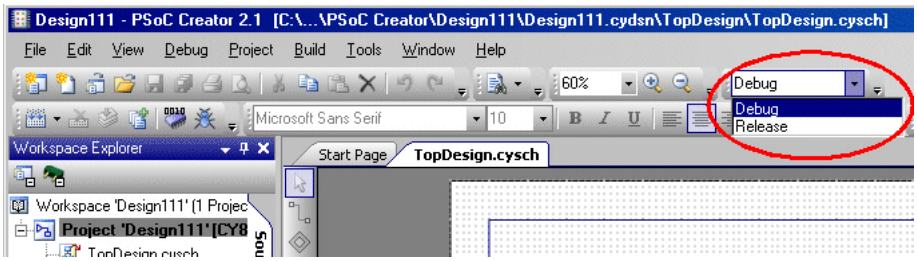
- 完成项目
- 存档工作区/项目

- 源代码控制

设置编译配置

PSoC Creator 为这些编译器工具链提供了 **Debug**（调试）和 **Release**（释放）编译配置。编译配置过程中的更改对开发和测试设计非常有用。例如，初次进行开发代码时使用 **Debug** 配置是最简单的情况，因为它通常进行较小的优化，并产生非常多的调试信息。当您移植到生产状态时，代码会进入稳定状态并准备释放。因此，当需要多次进行额外优化时，优先使用 **Release** 配置。这些优化有助于减少程序的大小，并允许它快速运行。

默认情况下可以使用 **Build Configuration** 工具栏：



如果该工具栏不可用，请参考 Customize 对话框，了解如何添加该工具栏。

另外，请参考：

- 完成项目
- Customize（自定义）

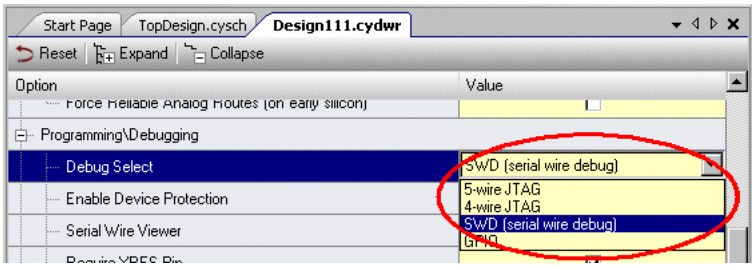
选择编程协议

PSoC Creator 在系统编辑器中支持配置编程端口。当进入生产时，这是非常重要的选择。请确保该选择与您系统的配置和预计相匹配。它不仅适用于初始编程事件，而且还适用于后续编程事件（若需要）。

如何选择该功能：

从工作区浏览器中打开设计范围资源（*<project>.cydwr* 文件。然后选择 **System** 选项卡。

在 **Programming\Debugging** 选项下，使用 **Debug Select** 下拉菜单选择正确的编程协议。



另外，请参考：

- 完成项目
- 系统编辑器
- 工作区浏览器

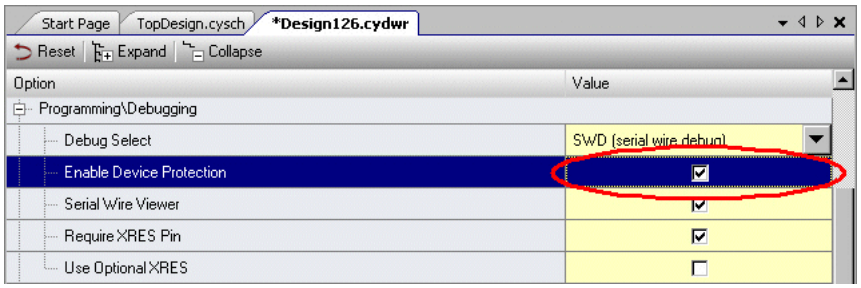
使能器件保护功能

完成一个生产设计时，您应该选择 **Enable Device Protection**（使能器件保护）功能。使能该功能后，器件将执行 Flash 保护设置并在运行期间禁止调试。器件可以与编程器相连，但是会禁用调试功能。

如何选择该功能：

从工作区浏览器中打开设计范围资源（*<project>.cydwr* 文件。然后选择 **System** 选项卡。

在 **Programming\Debugging** 选项下，选择 **Enable Device Protection** 复选框。



另外，请参考：

- 完成项目
- 系统编辑器
- 工作区浏览器

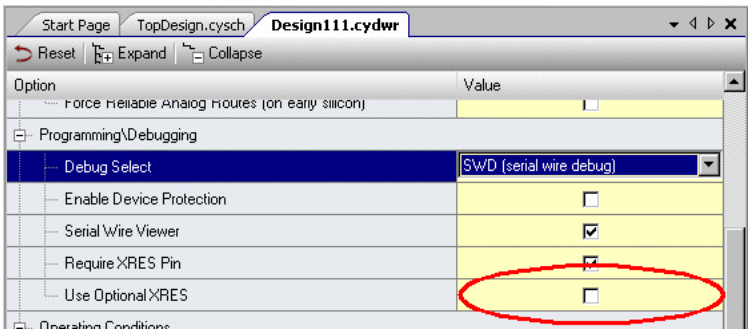
选择可选的复位引脚

PSoC Creator 在系统编辑器中支持可选复位（XRES）引脚的各选项。需要确保根据系统设计使能或禁用该选项。

如何选择该功能：

从工作区浏览器中打开设计范围资源（*<project>.cydwr* 文件。然后选择 **System** 选项卡。

在 **Programming\Debugging** 选项下，按需要选择或取消选择 **Use Optional XRES** 下拉复选框。



注意：如果您选择了可选复位引脚，并要求目标系统进行系统内编程，您必须确保编程解决方案（由分销商、合约制造商或第三方程序供应商）支持 Power Cycle 的编程模式。

另外，请参考：

- 完成项目
- 系统编辑器
- 工作区浏览器

选择闪存的安全保护

完成开发生产设计时，您需要 PSoC 上用户代码的闪存保护。另外，还需要确保外部实体不会访问 PSoC 应用代码。PSoC Creator 支持从设计范围资源 **Flash Security** 选项卡中选择闪存安全性。

如何使用该功能：

从工作区浏览器中打开设计范围资源（*<project>.cydwr* 文件。然后选择 **Flash Security** 选项卡。

按需要设置闪存安全性选项。更多有关信息，请参考闪存安全性编辑器。

另外，请参考：

- 完成项目
- 工作区浏览器
- 闪存安全性编辑器

使能一次性写锁存器闪存保护

完成生产级设计时，除了选择闪存安全性保护中说明的闪存保护外，您应考虑其他闪存保护设置。PSoC 产品支持一次性写锁存器（WOL）编程保护。使能该项时，您可以将 NVL 设置修改为使能 WOL 选项。

使用正确的 NVL 配置编程该器件后，它将被永久锁定。由于使能 WOL 后该器件无法被恢复，因此需要慎重使用该功能。该选项提供在 Programming Options（编程选项）菜单中的 PSoC Programmer 工具内。

要想通过使能生产 HEX 文件使能 WOL 选项，NVL 配置数据必须包含保护代码。更多有关信息，请参考以下网站中提供的相应 PSoC 器件编程规范：

http://www.cypress.com/go/p3_p5_trm

另外，请参考：

- 完成项目
- 选择闪存的安全保护

评估通用的编程选项

当您完成设计时，必须考虑各编程选项和解决方案。目前，赛普拉斯提供了一个普通编程网页，上面介绍很多重要的编程主题。具体而言，您可以通过增值程序或直接通过合格的第三方编程供应商检查合约制造商和分销商的可用制造编程解决方案。

您可以导航到该编程网页进行查看关于重要的编程主题（如原理图要求、编程规格、可用的开发编程器以及合格的第三方生产编程供应商）的详细内容。

http://www.cypress.com/go/programming

另外，请参考：

- 完成项目

10 参考文档

该发行版本还包含以下各种文档：

- 由赛普拉斯提供
  - 系统参考指南
  - 组件创建指南
  - 定制器 API 参考指南
  - 调谐器 API 参考指南
  - Warp Verilog 参考指南

- 由第三方提供

这个安装文档在 PSoC Creator **Help** 菜单上的 **Documentation**（文档）中列出。

系统参考指南

它说明了赛普拉斯提供的各种功能，以便更好地访问芯片资源。虽然这些功能并非来自组件库，但仍能供其使用。您可以通过函数调用可靠地执行所需芯片功能。该文档中介绍的功能包括：

- 标准类型和定义
- 时钟
- 电源管理
- 中断
- 缓存
- 引脚
- 寄存器访问
- 闪存和 EEPROM
- 系统函数
- 启动和连接
- 看门狗定时器

组件创建指南

该指南提供了有助于创建 PSoC Creator 组件的指导和信息。高级用户使用 CAG 来创建复杂的组件，以便最终用户可以使用组件这些与 PSoC Creator 进行交互。然而，CAG 也提供了一些基本原理，这些内容对想要创建自己内容的初学者非常有用。

可以将 CAG 安装为组件开发套件的一部分。您可以在 Windows **Start** 菜单或 PSoC Creator **Help** 菜单上找到该 CAG。

创建组件的过程包括以下高级步骤。更多有关信息，请参考该指南中的其他章节。

- 创建库项目
- 创建组件/符号
- 定义符号信息
- 创建实现
- 仿真硬件
- 创建 API 文件
- 自定义组件
- 添加/创建文档以及其他文件/文档
- 编译、检测和部署库项目

注意：这些章节提供了相关信息的逻辑组合，并介绍了创建组件的一种工作流程（有多种流程）。

定制器 API 参考指南

定制器 API 参考指南提供了用于扩展代码的 API 参考信息，可以使用这些信息自定义各组件。该指南由自定义源代码和注释生成。高级用户使用组件创建指南来创建复杂的组件，以便最终用户可以使用组件这些与 PSoC Creator 进行交互。

可以将 API 参考指南安装为组件开发套件的一部分。您可以在 Windows **Start** 菜单或 PSoC Creator **Help** 菜单上找到该指南。

调谐器 API 参考指南

调谐器 API 参考指南提供了有关调谐器的 API 参考信息，可以使用这些信息自定义组件。该指南由调谐器源代码和注释生成。高级用户使用组件创建指南来创建复杂的组件，以便最终用户可以使用组件这些与 PSoC Creator 进行交互。

可以将 API 参考指南安装为组件开发套件的一部分。您可以在 Windows **Start** 菜单或 PSoC Creator **Help** 菜单上找到该指南。

Warp Verilog 参考指南

Warp Verilog 参考指南提供了在 Warp 中实现的 Verilog 的基本元素。可以将该指南安装为组件开发套件的一部分。您可以在 Windows **Start** 菜单或 PSoC Creator **Help** 菜单上找到该指南。

- 第一节介绍的是 Warp 中支持的 Verilog 语言结构。
- 第二节介绍的是在 Warp 中实现的寄存器和三态合成。
- 最后一节介绍了一些 Verilog 设计示例。
- 附录包含 Verilog 保留字的列表

Warp 合成编译器是一个用于设计 PSoC 器件的 Verilog 编译器。Warp 接收 Verilog 文本输入，然后合成并优化目标硬件的设计。接下来，Warp 将输出一个用于编程器件的文件。Warp 在背景中运行。几乎所有用户都不直接与该程序交互。

另外，请参考：

- 控制文件
- Directives（指令）

第三方参考

PSoC Creator 包含第三方编译器以及相关文档。

该文档位于以下默认安装目录中：

- **GNU Documentation -** <INSTALL_DIR>\import\gnu_cs\arm\4.4.1\share\doc\arm-arm-none-eabi\pdf
- **Keil Compiler Documentation -** <INSTALL_DIR>\import\keil\C51\hlp

您可以从 PSoC Creator Help 菜单访问这些文档。

11 联系我们

感谢您与我们联系。我们重视您的意见和建议，以帮助我们改进 PSoC Creator。

可以使用以下任何方式联系我们：

- 要想获取现场帮助，请致电 1-800-541-4736 并选择 8
- 要想获取在线帮助，请访问 http://www.cypress.com/go/support
- 有关普通注释和建议，请发送电子邮件至 psoc_creator_feedback@cypress.com

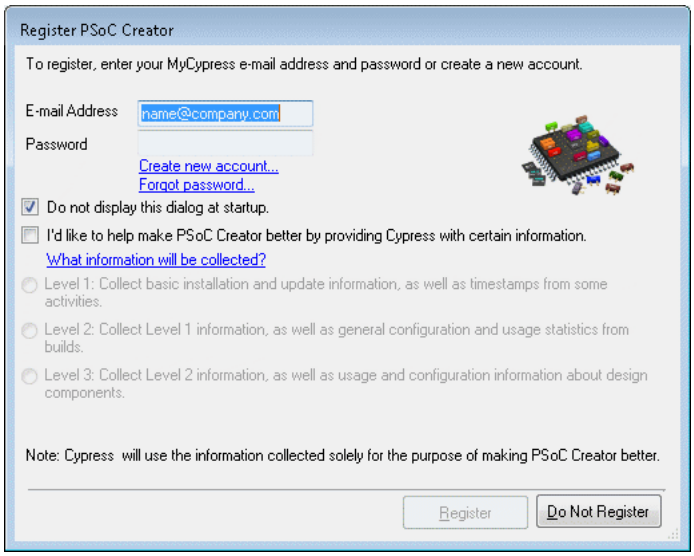
请尽可能包含以下各项。

- 问题的简要说明
- 编译版本[从 PSoC Creator **Help** 菜单选择 **About**。]
- O/S
- 错误信息（若有）
- 发生问题前您正进行的操作
- 问题的严重性（低、中、高）

若可以，请附带一张屏幕截图。

12 注册 PSoC Creator

Register PSoC Creator 对话框在启动时自动显示，除非您已经注册了 PSoC Creator 或指示不用显示该对话框。



通过选择 **Help** 菜单中的 **Register...**，您也可以打开该对话框

要注册 PSoC Creator：

请输入您的 MyCypress 邮件地址和密码，然后点击 **Register**。点击 **Register Later**，关闭该对话框不需要注册。

- 如果注册过程失败，该对话框上将显示一个错误信息。
- 如果您取消注册过程或进行三次尝试后仍不能访问网络，该对话框将显示一个复选框，通过它您可以跳过注册步骤。

如何创建新账户：

如果您还没有 cypress.com 的用户账户，请点击 **Create new account**（创建新账户）链接。这样，网页浏览器将打开 cypress.com 账户创建页面。

如果您忘记了密码：

如果您忘记了密码，请点击 **Forgot Password** 链接。这样，默认网页浏览器将打开 cypress.com 密码忘记页。

所收集的信息：

注册过程将以下配置信息传送到赛普拉斯并将其与您的 cypress.com 账户相连。

- 赛普拉斯软件产品的名称和版本
- Windows 和.NET 的版本
- CPU 的类型和速度
- 存储器大小

如果您想了解如何使用我们产品所收集的信息，请点击 **How Cypress will use my information** 链接，使网页浏览器显示赛普拉斯在注册过程收集到的信息类型，数据收集系统聚集的信息并打开赛普拉斯隐私政策页面的链接。

信息等级：

请选择以下信息级中的一个来发送至赛普拉斯，从而改进 PSoC Creator：

- 等级 1 — 基本信息
- 等级 2 — 等级 1 以及编译的配置信息
- 等级 3 — 等级 2 以及设计组件的用途和配置信息

13 索引

D

E

F

Q

R

S

T

U