CY8CKIT-040

# PSoC® 4000 Pioneer套件指南

文档编号：001-92737版本*B

赛普拉斯半导体
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

## 版权所有

© 赛普拉斯半导体公司， 2014-2020 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC （“ **赛普拉斯** ”）的财产。本文件，包括其包含或引用的任何软件或固件 （“ **软件** ”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可 （无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权 （一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和 （2）在被软件 （由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

**在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证**。没有任何电子设备是绝对安全的。因此，尽管赛普拉斯在其硬件和软件产品中采取了必要的安全措施，但是赛普拉斯并不承担任何由于使用赛普拉斯产品而引起的安全问题及安全漏洞的责任，例如未经授权的访问或使用赛普拉斯产品。此外，本材料中所介绍的赛普拉斯产品有可能存在设计缺陷或设计错误，从而导致产品的性能与公布的规 格不一致。（如果发现此类问题，赛普拉斯会提供勘误表）赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途 （“ **非预期用途** ”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion徽标，及上述项目的组合，WICED，及PSoC、CapSense、EZ-USB、F-RAM和Traveo应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问cypress.com获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。

# 目录


安全信息 5

1. 简介 7
   - 1.1 套件内容 ....7
   - 1.2 PSoC Creator ....9
     - 1.2.1 PSoC Creator中的代码示例 ....10
     - 1.2.2 套件代码示例 ....11
     - 1.2.3 PSoC Creator帮助 ....12
   - 1.3 入门 ....13
   - 1.4 其他参考资源 ....13
   - 1.5 技术支持 ....14
   - 1.6 文档规范 ....14

2. 软件安装 15
   - 2.1 准备 ....15
   - 2.2 安装软件 ....15
   - 2.3 安装硬件 ....17
   - 2.4 卸载软件 ....18

3. 套件操作 19
   - 3.1 套件概述 ....19
   - 3.2 套件USB接口 ....21
   - 3.3 编程与调试PSoC 4000 ....22
     - 3.3.1 使用板载PSoC 5LP的编程器和调试器 ....22
     - 3.3.2 使用CY8CKIT-002 MiniProg3编程器和调试器 ....23
   - 3.4 USB-I2C桥接器 ....25
   - 3.5 USB-UART桥接器 ....26
   - 3.6 更新编程器的固件 ....27

4. 硬件 29
   - 4.1 电路板的详细信息 ....29
   - 4.2 框图 ....32
   - 4.3 套件组件的详细信息 ....33
     - 4.3.1 CY8CKIT-040主板组件 ....33
     - 4.3.2 CY8CKIT-040 CapSense触控板扩展电路板 ....47

5. 代码示例 49
   - 5.1 概述 ....49
     - 5.1.1 编程示例项目 ....49
   - 5.2 项目：使LED闪烁发光 ....54
     - 5.2.1 项目概述 ....54

5.2.2 项目说明 ..... 55
5.2.3 验证输出 ..... 56
5.3 项目：CapSense接近感应和UART ..... 57
5.3.1 项目概述 ..... 57
5.3.2 项目说明 ..... 58
5.3.3 验证输出 ..... 67
5.4 项目：使用I2C调谐器的CapSense触摸板 ..... 74
5.4.1 项目概述 ..... 74
5.4.2 项目说明 ..... 76
5.4.3 验证输出 ..... 79
5.5 项目：调色板（Color Palette） ..... 84
5.5.1 项目概述 ..... 84
5.5.2 项目说明 ..... 84
5.5.3 验证输出 ..... 101
5.6 PSoC 4000中的ADC ..... 102
5.6.1 使用PSoC Creator代码示例：ADC_VoltageInput ..... 102

**6. 高级主题 105**

6.1 将PSoC 5LP作为USB-I2C桥接器使用 ..... 105
6.2 使用FM24W256 F-RAM ..... 113
6.2.1 地址选择 ..... 113
6.2.2 读/写操作 ..... 114
6.2.3 示例固件 ..... 115
6.3 将PSoC 5LP作为USB-UART桥接器使用 ..... 118
6.4 开发PSoC 5LP的应用 ..... 130
6.4.1 为PSoC 5LP编译Bootloadable项目 ..... 131
6.4.2 编译PSoC 5LP的普通项目 ..... 139
6.5 PSoC 5LP出厂设置恢复说明 ..... 141
6.5.1 使用Bootloadable应用编程过的PSoC 5LP ..... 141
6.5.2 使用标准的应用编程过的PSoC 5LP ..... 147
6.6 使用µC/Probe工具 ..... 149

**A. 附录 158**

A.1 CY8CKIT-040原理图 ..... 158
A.2 引脚分配表 ..... 163
A.3 编程和调试插座 ..... 165
A.4 0 Ω电阻和空载的使用 ..... 166
A.5 KitProg的LED状态 ..... 167
A.6 材料清单 ..... 168
A.7 触控板标签详情 ..... 171
A.8 法规遵从信息 ..... 171
A.9 在不同的Pioneer系列套件间移植项目 ..... 172

**修订记录 176**

文档修订记录 ..... 176

# 安全信息

## 法规遵从

CY8CKIT-040 PSoC® 4000 Pioneer 套件旨在用作实验室环境中硬件和软件的开发平台。该电路板是一个开放系统设计，没有外壳保护。因此，在很近距离接触中，可能会对其他电器和电气设备造成干扰。在室内环境中，该产品可能会导致无线干扰。这时，用户需要采取适当的预防措施。同样，不应该在医疗设备或 RF 器件附近使用该电路板。

增加连接到该产品的连线或修改该产品的默认设置可能会影响它的性能，并对周围的其他设备造成干扰。如果检测到该类干扰，应采取适当措施降低干扰。

所有的 CY8CKIT-040 出厂前都经过验证确保能满足 CE 对 A 类产品的要求。

CY8CKIT-040 包含静电放电 （ESD）敏感器件。静电电荷很容易会在人体或所有设备上累积，并且瞬间释放。对于经常发生高能量放电的设备，则可能造成永久性损害。建议采取适当的 ESD 预防措施，以避免降低其性能或丧失功能。CY8CKIT-040 电路板应该被存放在保护性运输包装里。

有效期 / 产品回收

从盒子背面显示的制造日期算起，该套件的有效期为五年。请联系离您最近的回收商来处理套件。

# 通用安全说明

## ESD 保护

ESD 会损坏电路板和相关组件。赛普拉斯建议用户只在 ESD 工作站环境中操作。如果 ESD 工作站不可用，那么对器件进行操作时，请采取适当的 ESD 保护措施：戴上一个连接到电路板上机架接地（任何未上漆的金属平面）的防静电手环。

## 操作电路板

CY8CKIT-040 电路板对 ESD 十分敏感。只能握着电路板的边缘。从盒子里取出电路板后，请将其放置在一个接地且无静电的平面上。如果可能，请使用一个导电泡沫衬垫。请勿将该电路板在任何表面上滑动。

# 1. 简介

感谢您对 PSoC® 4000 套件的关注。该套件是简单易用且经济的开发套件，它展示了 PSoC 4000 架构独特的灵活性。以灵活性为设计目标，本套件提供了与某些第三方的 Arduino™ 扩展板相兼容的插座。另外，该电路板还集成了 RGB LED、板载 USB 编程器 / 调试器、编程 / 调试插座、USB-UART/ $I^2C$ 桥接器、接近感应插座以及一个与 Arduino 相兼容的 CapSense® 触控扩展板。该套件支持 5 V 或 3.3 V 的电源电压。

PSoC 4000 Pioneer 套件基于 PSoC 4000 系列器件为许多嵌入式应用程序提供可编程的平台。PSoC 4000 是 PSoC 4 平台的最小成员，它支持 CapSense、定时器 / 计数器 / 脉宽调制器 （TCPWM）、$I^2C$ 主设备或从设备，并支持多达 20 个 GPIO。PSoC 4000 是一个低成本的入门级 PSoC4 器件，用于代替过时的和 / 或专有的 8 位和 16 位 MCU。带有 ARM Cortex-M0 内核的 PSoC 4000 提供了 32 个可编程外设，包括 CapSense 在内。

## 1.1 套件内容

PSoC 4000 Pioneer 套件包括 （请查看图 1-1）：

- PSoC 4000 Pioneer 套件电路板
- 带有与一个调色板贴纸的触控板扩展板
- 快速入门指南
- 标准 USB-A 转 Mini-B 型线缆
- 六条跳线

**注意：**本文档中的触控板和触摸板表示相同的含义可以互换使用。

图 1-1. 套件内容

如果发现缺少任何部件，请联系距离您最近的赛普拉斯销售网点以获得帮助：www.cypress.com/go/support。

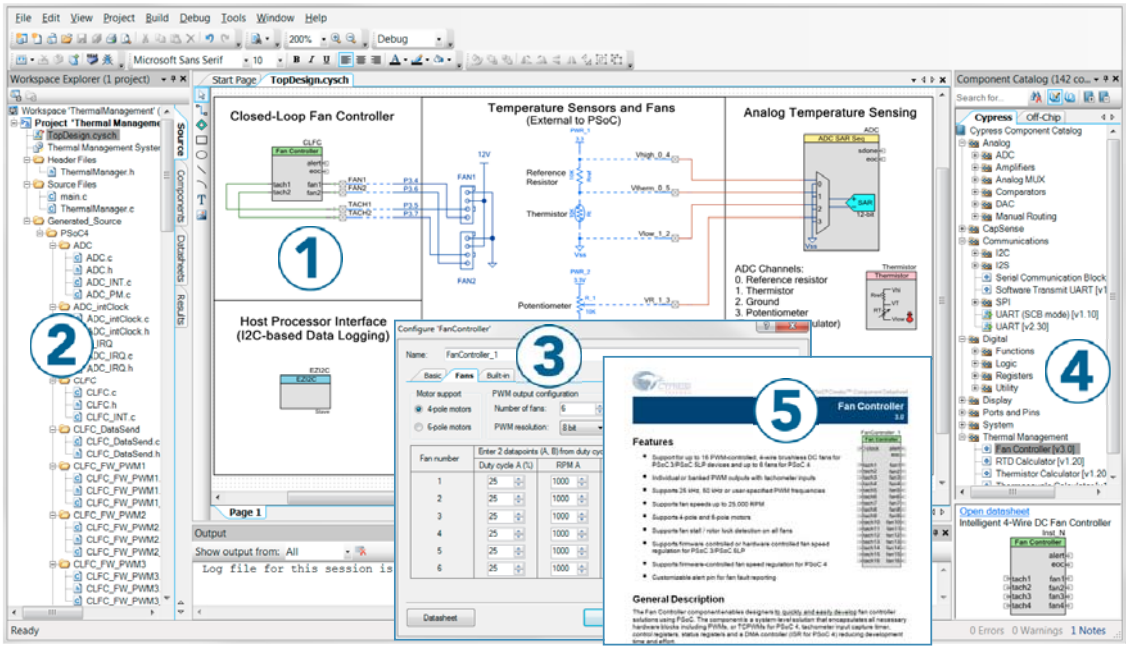
您可以从 www.cypress.com/CY8CKIT-040 下载该套件相关文件的最新版本。

## 1.2 PSoC Creator

PSoC Creator™ 是一个先进、简单易用的集成开发环境 （IDE）。它包含了大量预验证和预配置的 PSoC 组件，从而能够进行革命性的硬件和软件协同设计。通过使用 PSoC Creator，可以执行以下操作：

1. 将组件图标拖放到主要设计工作区中，以进行您的硬件系统设计
2. 对您的应用固件和 PSoC 硬件进行协同设计。
3. 使用配置工具配置各组件。
4. 了解包含一百多个组件的库。
5. 查看组件数据手册。

图 1-2. PSoC Creator 特性




PSoC Creator 为您提供了完备开发工具的系统，包括适用于 PSoC 器件的集成编译器链接和量产编程器。

更多有关信息，请访问 www.cypress.com/psoccreator。有关学习和使用 PSoC Creator 的视频向导，请参考 PSoC Creator 培训网页。

### 1.2.1 PSoC Creator 中的代码示例

PSoC Creator 为您提供了大量代码示例。可以从 PSoC Creator 的 “Start Page” （起始页）上获取这些示例，如图 1-3 所示。

通过向您提供完整的设计 （并非一个空白设计），代码示例可以加快您的设计进程。代码示例还介绍了如何将 PSoC Creator 组件用于不同的应用中。另外，同时提供了代码示例和文档，如第 11 页上的图 1-4 所示。

在第 11 页上的图 1-4 所示的 **Find Example Project**（查找示例项目）对话框中，您可以选择以下选项：

- 根据 architecture（架构）或 device family（器件系列）（例如：PSoC 3、PSoC 4 或 PSoC 5LP）、project name （项目名称）或 keyword （关键词）等选项对示例进行筛选。
- 从 **Filter Options** 的示例菜单中进行选择。
- 查看示例项目的描述 （在 **Documentation** 选项卡上）。
- 查看 **Sample Code** 选项卡中的代码。您可以复制该窗口中的代码，然后将其粘贴在您的项目内，从而加快代码开发过程。
- 根据已选工程创建一个新的工程 （若需要可添加新的工作区） 。通过为您提供一个完整的基本设计，它可以加快您的设计过程。然后，您可以根据自己的应用要求来调整该设计。

图 1-3. PSoC Creator 中的代码示例

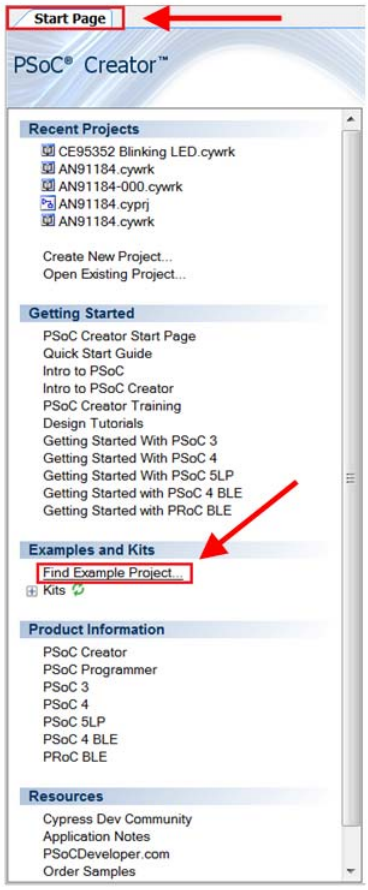
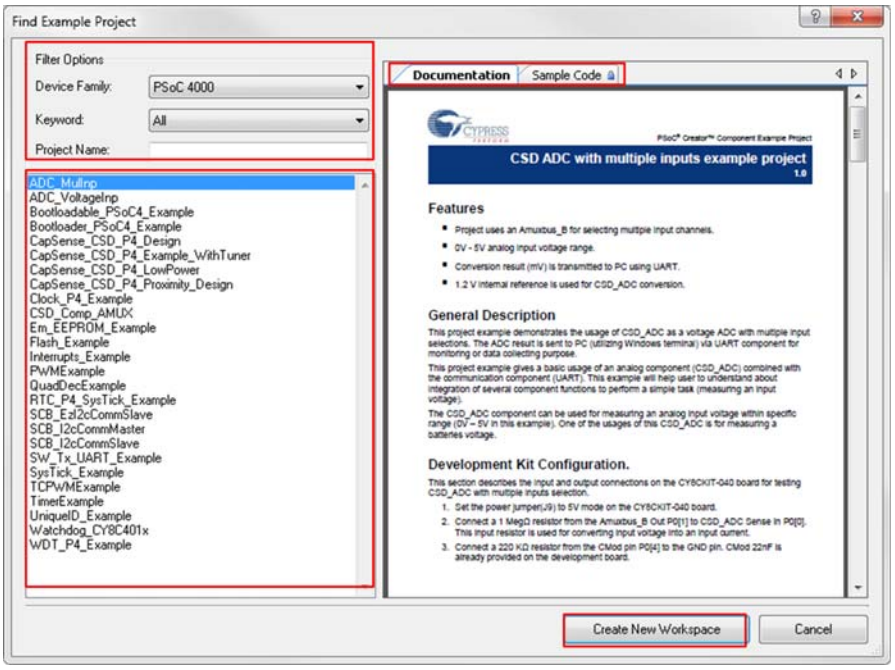


图 1-4. 带样本代码的代码示例项目




## 1.2.2 套件代码示例

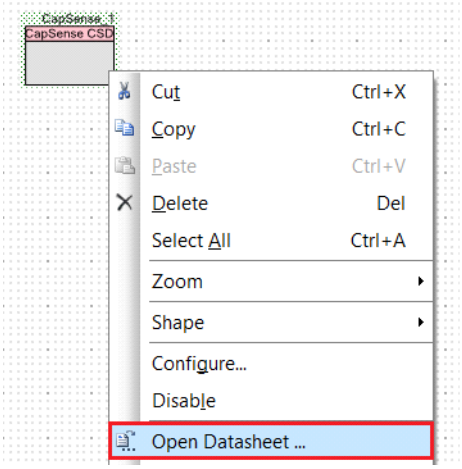
除了 PSoC Creator 内置的示例外，该套件还包含一个简单示例，可用于快速评估该套件的功能。该示例在第 49 页上的代码示例章节中描述。此外，本章节还包括一个部分，通过举例说明如何使用该套件配有的 PSoC Creator 代码示例。

### 1.2.3 PSoC Creator 帮助

请访问 PSoC Creator 主页以下载 PSoC Creator 的最新版本。然后，启动 PSoC Creator，并导航到以下各项：

- **快速入门指南**：依次选择 **Help > Documentation > Quick Start Guide**。本指南提供了有关开发 PSoC Creator 项目的基本知识。
- **简单组件的示例项目**：依次选择 **File > Example project...**。这些示例项目展示了如何配置和使用 PSoC Creator 组件。
- **入门设计**：依次选择 **File > New > Project > PSoC 4000 Starter Designs**。这些入门设计展示了 PSoC 4 的独特特性。
- **系统参考指南**：依次选择 **Help > System Reference Guides**。该指南列出并描述了 PSoC Creator 提供的系统功能。
- **组件数据手册**：右击组件，然后选择 **Open Datasheet** 项，如第 12 页上的图 1-5所示。请访问 PSoC 4 组件数据手册网页，以下载所有 PSoC 4 组件的数据手册列表。
- **文档管理工具**：PSoC Creator 提供了一款文档管理工具，便于查找和参考文档资源。要想打开文档管理工具，请选择菜单项：**Help > Document Manager**。

图 1-5. 打开组件数据手册



## 1.3 入门

本指南帮您熟悉 PSoC 4000 Pioneer 套件。

- 第 15 页上的软件安装章节介绍了该套件软件的安装情况。
- 第 19 页上的套件操作章节详细说明了如何使用编程器和调试器（板载 PSoC 5LP 或外部 MiniProg3（CY8CKIT-002））对 PSoC 4 进行编程。
- 第 29 页上的硬件章节 详细说明了硬件的操作。
- 第 49 页上的代码示例章节 描述了该套件中提供的示例项目。
- 第 104 页上的高级主题章节介绍了以下内容：构建 PSoC5 LP 的项目、如何使用板载 F-RAM、USB-UART 功能以及 PSoC5 LP 的 USB-$I^2C$ 功能。
- 第 157 页上的附录 提供了各个原理图、引脚分布、零欧姆电阻、故障排除以及材料表（BOM）的详细信息。

## 1.4 其他参考资源

在赛普拉斯的 www.cypress.com 网站上提供了大量资料，有助于选择符合您设计的 PSoC 器件，并能够快速有效地将该器件集成到您的设计中。有关完整的资源列表，请参考 KBA86521 — 如何使用 PSoC 3、 PSoC 4 和 PSoC 5LP 进行设计。下面提供了 PSoC 4 的简要列表：

- 概况：PSoC 产品系列和 PSoC 蓝图。
- 产品选型：PSoC 1、PSoC 3、PSoC 4 或 PSoC 5LP。此外，PSoC Creator 还包含了一个器件选择工具。
- 数据手册：描述并提供了适用于 PSoC 4000 器件系列的电气规格。
- CapSense 设计指南：了解如何使用 PSoC 4 器件系列设计电容式触摸感应应用。
- 应用笔记和代码示例：包括了从基本到高级的广泛主题。许多应用笔记包括了代码示例。要想获取 PSoC Creator 所有可用代码示例的完整列表，请访问 PSoC 3/4/5 代码示例。要想访问 PSoC Creator 中的代码示例，请查看第 10 页上的 PSoC Creator 中的代码示例的内容。
- 技术参考手册 （TRM）：对每个 PSoC 4 器件系列中所用的架构和寄存器进行了详细说明。
- 开发套件：
  - CY8CKIT-040、CY8CKIT-042 和 CY8CKIT-044 均为易于使用且廉价的开发平台。这些套件包括用于 Arduino 兼容屏蔽和 Digilent Pmod 外设模块的连接器。
  - CY8CKIT-049 和 CY8CKIT-043 都是成本非常低的原型平台，用于选择 PSoC 4 器件。
  - MiniProg3 套件提供了一个用于进行闪存编程和调试的接口。
- 知识库文章 （KBA）：提供了专家针对使用器件所提出的设计和应用技巧。
- PSoC Creator 培训：请访问 www.cypress.com/go/creatorstart/creatortraining 网站以获取 PSoC Creator 培训视频的完整列表。
- 从同事学习：请访问 www.cypress.com/forums 网站，以参加赛普拉斯开发者社区论坛上 PSoC 开发爱好者针对下一代嵌入式系统进行的讨论。

## 1.5 技术支持

如果您有任何问题，请在赛普拉斯技术支持页面上创建技术请求。

若果您是在美国，则可以拨打我们以下免费电话，直接与技术支持团队联系：+1-800-541-4736。在提示符处选择第 2 项。若果您不在美国，则可以拨打我们以下免费电话与技术支持团队联系：+1 (408) 943-2600 Ext. 2.

若想获得快速支持，您同样也可以使用下面的支持资源。

- 自助
- 所在地销售办事处

## 1.6 文档规范

表 1-1. 指南文档规范

| 规范 | 使用说明 |
|---|---|
| `Courier New` 字体 | 用于显示文件位置、用户输入的文本和源代码：<br>`C:\ ...cd\icc\` |
| 斜体字 | 用于显示文件名称和参考文档：<br>请阅读 *PSoC Creator 用户指南*文档中的 *sourcefile.hex* 文件。 |
| **[ 方括号、粗体字 ]** | 用于显示程序中的键盘命令：<br>**[Enter]** 或 **[Ctrl] [C]** |
| File > Open | 表示菜单路径：<br>File > Open > New Project |
| **粗体字** | 用于显示操作过程中的各条命令、菜单路径和图标名称：<br>请点击 **File** 图标，然后点击 **Open**。 |
| Times New Roman 字体 | 用于显示公式：<br>2 + 2 = 4 |
| 灰色框中的文本 | 用于说明警告或产品的独特功能。 |

# 2. 软件安装

本章节介绍了 CY8CKIT-040 PSoC 4000 Pioneer 套件软件的安装信息及其对操作系统的要求。

## 2.1 准备

所有赛普拉斯的软件安装都需要管理员特权。但是之前所安装完毕的软件不再有这样的要求。安装套件软件前，需要关闭所有其他正在运行的赛普拉斯软件。

## 2.2 安装软件

请按照下面步骤安装 CY8CKIT-040 PSoC 4000 Pioneer 套件软件：

1. 下载 CY8CKIT-040 软件。

   CY8CKIT-040 软件提供了三种不同的格式用以下载：

   a. **CY8CKIT-040 Kit Only**：这个可执行文件仅安装套件的内容，包括套件代码示例、硬件文件以及用户文档。如果第 5 步所列出的所有必要软件已经在你电脑上安装，则可使用该包装。

   b. **CY8CKIT-040 Kit Setup**：该安装包包含所有套件的内容以及 PSoC Creator、赛普拉斯文档管理器和 PSoC 编程器。但不包括 Windows 安装程序或 Microsoft .NET 框架包。如果您的电脑上没有这些软件包，安装程序将引导您从网上下载并安装它们。

   c. **CY8CKIT-040 CD ISO**：该文件是一个以 CD-ROM 镜像格式存放的完整安装包，您可以使用它来创建 CD 或通过 ISO 解压程序（如 WinRAR）对其进行解压。该文件也可以使用虚拟驱动程序（如 Virtual CloneDrive 或 MagicISO）像虚拟 CD 一样安装。该文件包含了所有所需的软件、工具、驱动程序、硬件文件和用户文档。

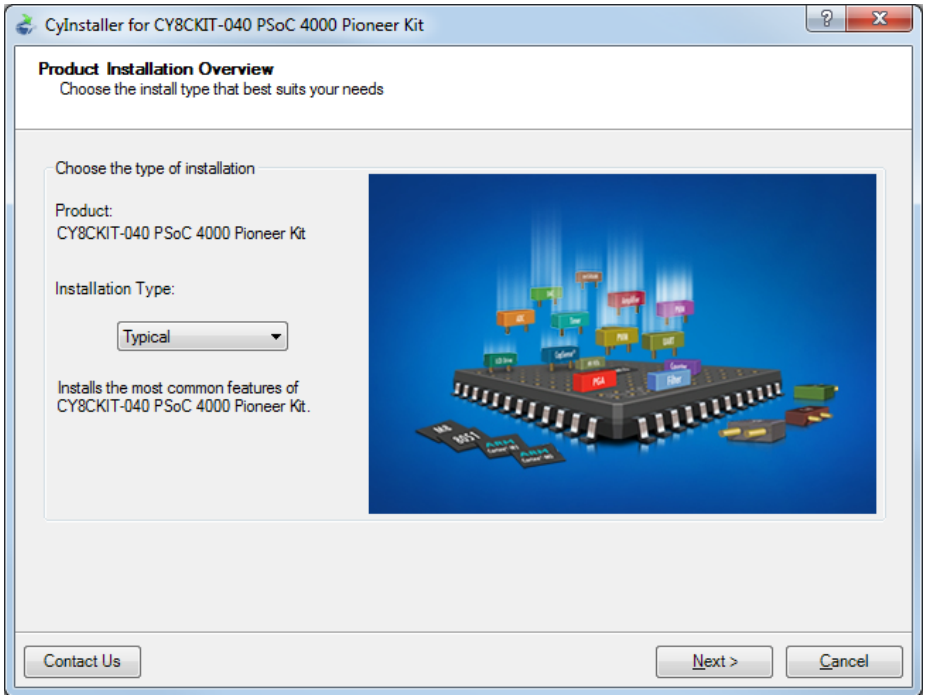
2. 如果您已经下载了 ISO 文件，请将其加载到一个虚拟光驱程序中。如果没有虚拟光驱程序，则要对 ISO 内容进行解压。如果电脑上未使能 ‘Autorun from CD/DVD’ 功能，请双击所解压内容或所加载到 ISO 内容的根目录中的 *cyautorun.exe*。此时会自动出现如图 2-1 显示的安装窗口。**注意：**如果您正在使用 ‘Kit Setup’ 或 ‘Kit Only’ 文件，请转到第 6 步进行安装。
3. 单击 **Install CY8CKIT-040**，以开始进行安装套件，如图 2-1 所示。

图 2-1. 套件安装程序启动屏幕

4. 选择您想安装与 CY8CKIT-040 套件相关文件的文件夹。选择该目录并点击 **Next** 按键。
5. 点击 **Next** 按键时，如果电脑上没有安装必备的软件， CY8CKIT-040 ISO 安装程序会自动安装。所需的软件包括：
   a. PSoC Creator 3.2 Service Pack 1 或更高版本：请在 www.cypress.com/psoccreator 网站上下载最新版本。
   b. PSoC Programmer 3.23.1 或更高版本：请在 www.cypress.com/programmer 网站上下载最新版本。
   c. 赛普拉斯文档管理器 1.0 Service Pack 1 或更高版本：请在 www.cypress.com/cypressdocumentmanager 网站上下载最新版本。

6. 在 Product Installation Overview（产品安装概述）窗口中，选择 **Typical/Custom/Complete** 的安装类型，如图 2-2 所示。选择安装类型后，请点击 **Next** 按键。

图 2-2. Product Installation Overview 窗口




7. 阅读并接受“End-User License Agreement”（终端用户许可协议），然后点击 **Next** 以进行安装。
8. 当开始进行安装时，软件安装包列表会显示在安装页面上。安装成功后，每个软件安装包后面会显示一个绿色的勾。
9. 请输入您的联系信息或勾选 **Continue Without Contact Information**（不提供联系信息并继续）选框。单击 **Finish** 按键完成 CY8CKIT-040 套件的安装。
10. 安装完成后，套件内容位于以下位置：

    `<Install_Directory>\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>`

    默认位置：

    在 Windows 7（64 位）中：

    `C:\Program Files (x86)\Cypress\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>`

    在 Windows 7（32 位）中：

    `C:\Program Files\Cypress\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>`

    **注意：**对于使用 Windows 7/8/8.1 的用户，所安装的文件和文件夹是只读的。要想更改属性，右键单击文件夹并依次选择 **Properties > Attributes**，然后取消 **Read-only** 选框的勾选。依次点击 **Apply** 和 **OK** 按键，关闭该窗口。

## 2.3 安装硬件

该套件不要求安装任何其他的硬件。

## 2.4 卸载软件

通过使用下面的任意方法，都可以卸载 CY8CKIT-040 PSoC 4000 Pioneer 套件软件：

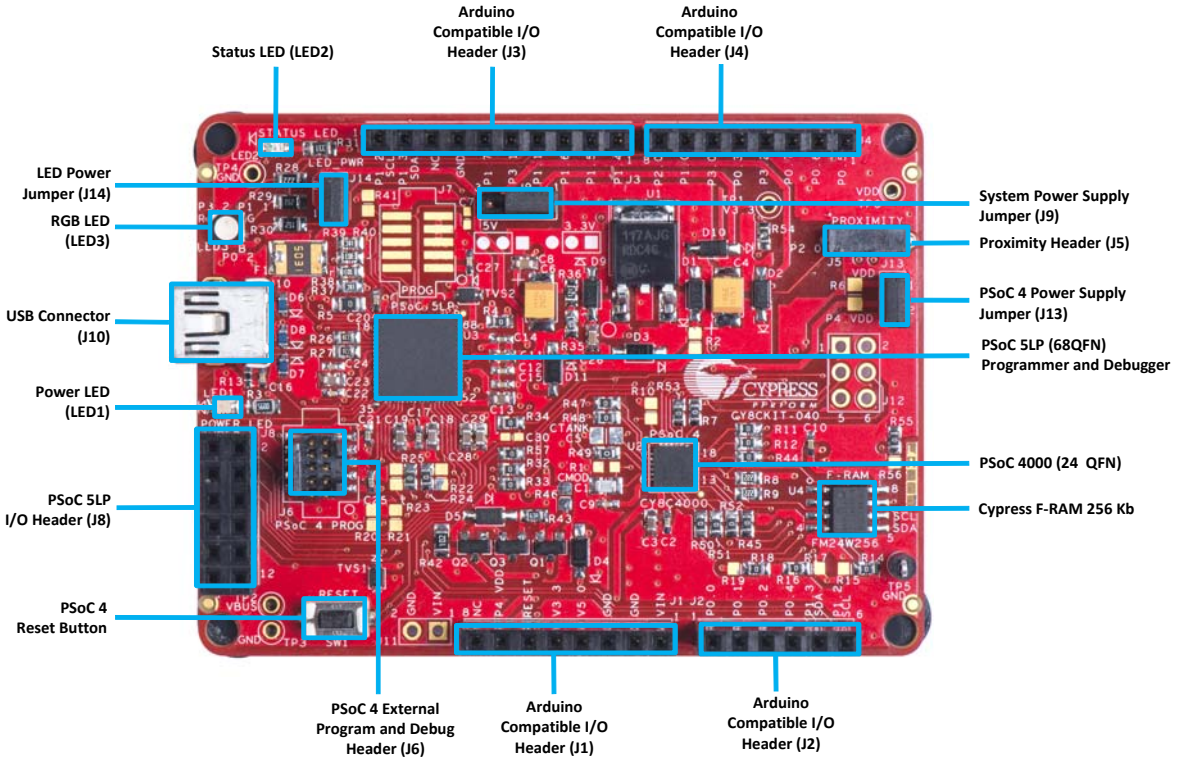
- 依次选择 **Start > All Programs > Cypress > Cypress Update Manager > Cypress Update Manager**。选择与套件软件相应的 **Uninstall** 按键。
- 依次选择 **Start > Control Panel > Programs and Features** （对于 Windows XP 则选择 **Add/Remove Programs**）。选择与套件软件相应的 **Uninstall/Change** 按键。

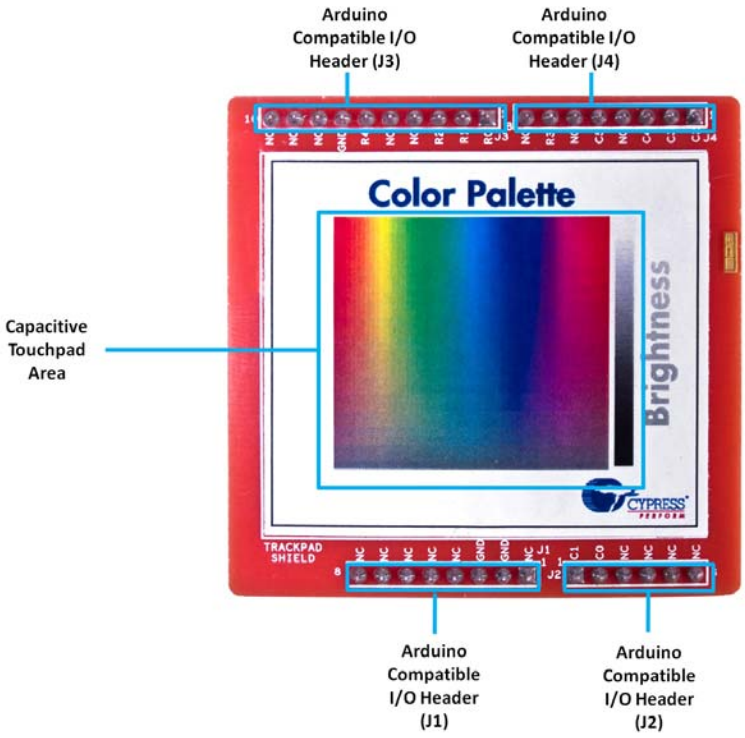
# 3. 套件操作

## 3.1 套件概述

PSoC 4000 Pioneer 套件可以用来开发 PSoC 4000 系列器件的各种应用。该套件包括两个电路板：和 Arduino 兼容的主板和基于 CapSense 的触控扩展板。图 3-1 是 PSoC 4000 Pioneer 套件主板和触控扩展板的图片，图片中标记出了板上的每个组件。

图 3-1. CY8CKIT-040 套件的详细信息




Arduino Compatible I/O Header (J3)
Arduino Compatible I/O Header (J4)
Color Palette
Brightness
Capacitive Touchpad Area
TRACKPAD SHIELD
Arduino Compatible I/O Header (J1)
Arduino Compatible I/O Header (J2)

## 3.2 套件 USB 接口

PSoC 4000 Pioneer 套件通过 USB 接口连接到 PC （请查看图 3-2）。该套件作为复合设备枚举，它的三个独立设备在 Windows 操作系统中的 **Device Manager** （设备管理器）窗口内显示。请参见表 3-1 和图 3-3。

图 3-2. 套件 USB 接口

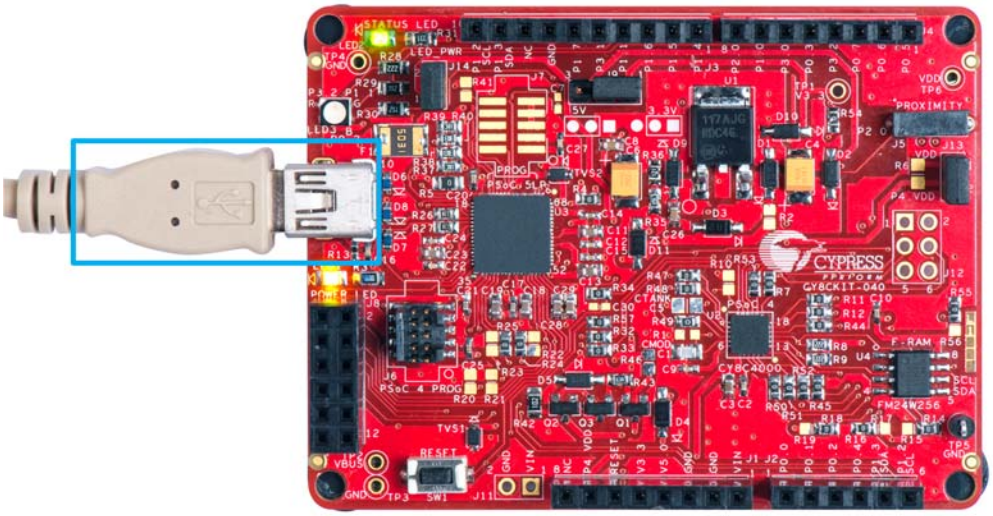
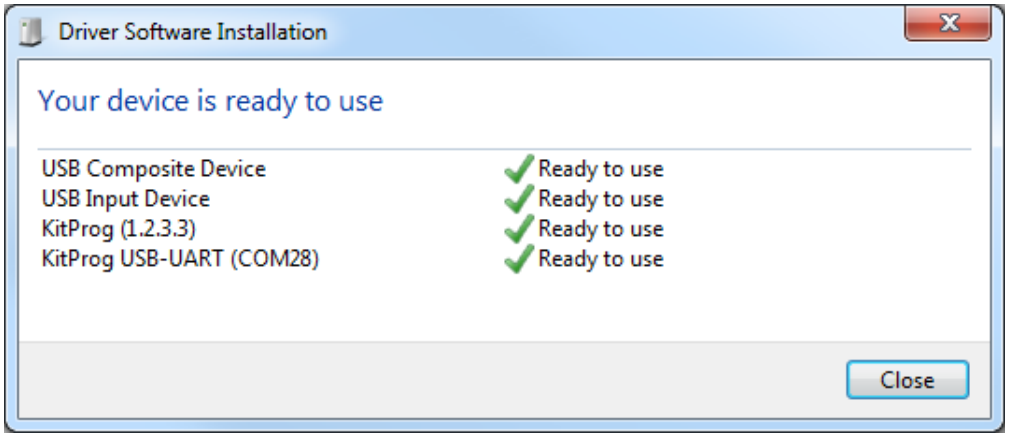


表 3-1. 枚举后设备管理器中显示的 PSoC 4000 Pioneer 套件

| 端口 | 说明 |
|---|---|
| USB 复合器件 | 复合器件 |
| USB 输入器件 | USB-I$^2$C 桥接器， Kitprog 命令接口 |
| KitProg | USB-I$^2$C 桥接器、编程器和调试器 |
| KitProg USB-UART | USB-UART 桥接器 （显示为 COM# 端口） |

图 3-3. KitProg 驱动程序安装完成



## 3.3 编程与调试 PSoC 4000

该套件允许在两种模式下对 PSoC 4 器件进行编程与调试：

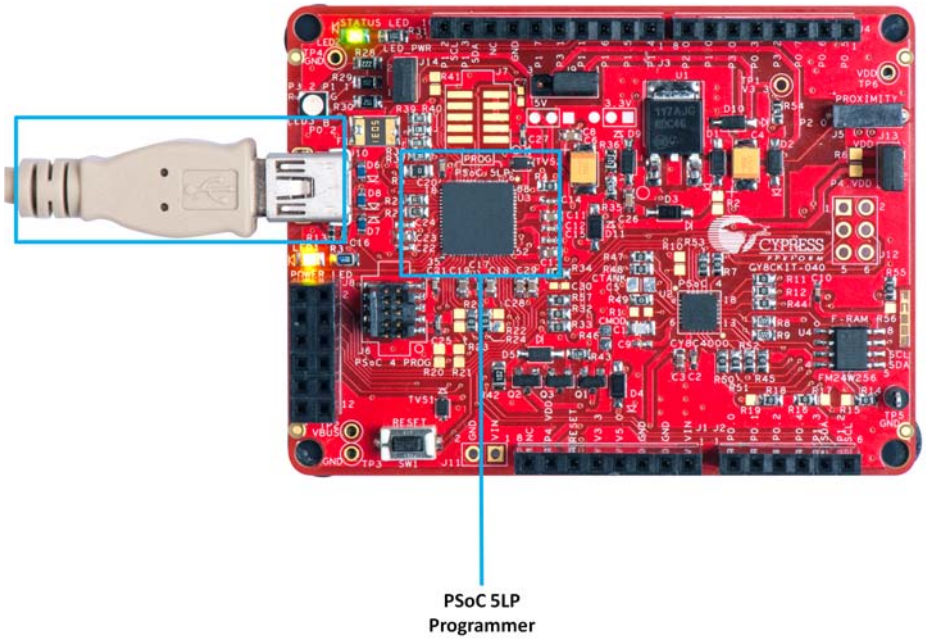
- 3.3.1 使用板载 PSoC 5LP 的编程器和调试器
- 3.3.2 使用 CY8CKIT-002 MiniProg3 编程器和调试器

### 3.3.1 使用板载 PSoC 5LP 的编程器和调试器

套件的默认编程接口是基于 USB 的板载编程接口。对 PSoC 4 进行编程前，必须安装 PSoC Creator 和 PSoC Programmer。请查看第 15 页上的安装软件，了解如何安装套件软件的信息。

1. 要想编程该器件，请将 USB 线缆插入到编程 USB 连接器 J10 上，如图 3-4 所示。套件将被枚举成复合设备。更多详细信息，请参考第 21 页上的套件 USB 接口。

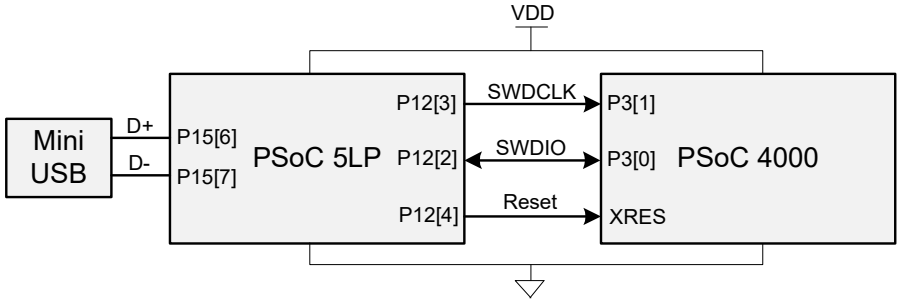
图 3-4. 将 USB 线缆连接到 J10




2. 板载 PSoC 5LP 使用了串行线调试接口 （SWD）对 PSoC 4 器件进行编程。请参见图 3-5。

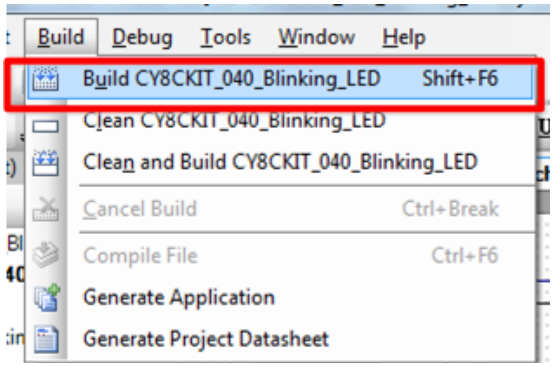
   **注意：**图 3-5 仅供参考，所有连接均在板上通过固定连线实现。

图 3-5. PSoC 5LP 使用 SWD 对 PSoC 4000 进行编程



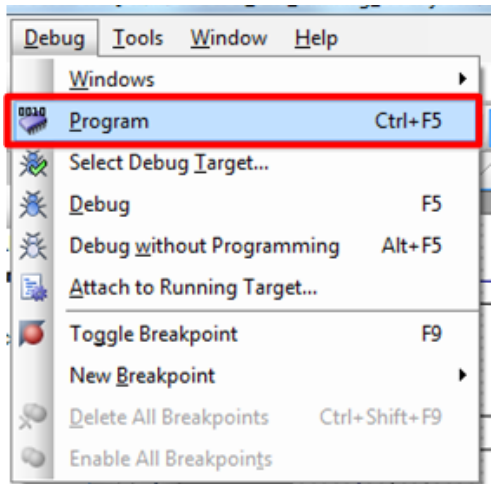
3. 套件的板载编程器在 PC 和软件工具中枚举为 **KitProg**。打开 PSoC Creator 中的示例项目 （如第 54 页上的项目：使 LED 闪烁发光），并通过选择 **Build > Build Project** 或按下 **[Shift] [F6]**，开始创建项目。请参见图 3-6。

图 3-6. 在 PSoC Creator 中创建项目




4. 正常编译项目后，如果没有错误和警告，依次选择 **Debug > Program** 或按组合**键 [Ctrl] [F5]** 来编程该器件。请参见图 3-7 的内容。

图 3-7. 使用 PSoC Creator 编程器件




板载编程器仅支持 RESET 编程模式。使用板载编程器时，电路板可以由 USB （VBUS）或外部电源（如 Arduino 扩展板）供电 （请参见第 36 页上的供电系统）。如果已经使用了另外的电源给电路板供电，插入 USB 编程器并不会损坏电路板。

### 3.3.2 使用 CY8CKIT-002 MiniProg3 编程器和调试器

可以使用 MiniProg3 （CY8CKIT-002）对套件上的 PSoC 4 进行编程。此时，需要使用开发板上的连接器 J6，如图 3-8 所示。 MiniProg3 的编程操作和板载编程器的操作相类似；只是，它被枚举为 MiniProg3，而不是 KitProg。

电路板也可以由 MiniProg3 供电。要想实现该操作，请在 PSoC Creator 中依次选择 **Tool > Options**。在 Options 窗口中，扩展 **Program/Debug > Port Configuration** ； 点击 **MiniProg3** 并选择图 3-9 中所示的设置内容。依次选择 **Debug > Program**，对电路板进行编程与供电。

**注意：**CY8CKIT-002 MiniProg3 并不包含在 PSoC 4000 Pioneer 套件中。您可以从 赛普拉斯在线商店购买。

图 3-8. 使用 MiniProg3 对 PSoC 4 进行编程 / 调试

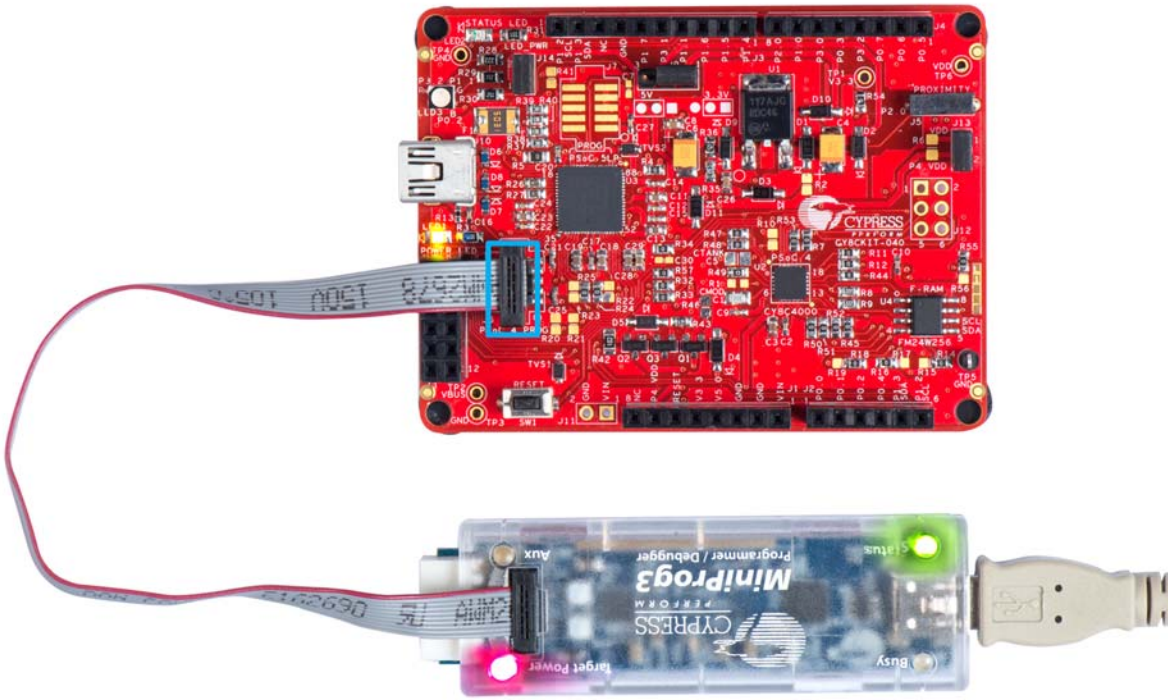
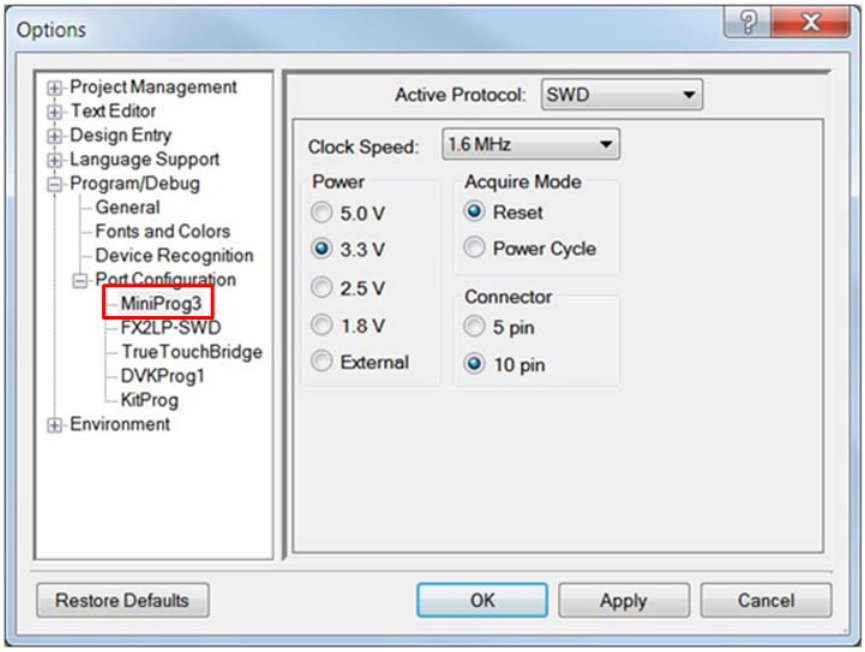



图 3-9. PSoC Creator 中的 MiniProg3 配置



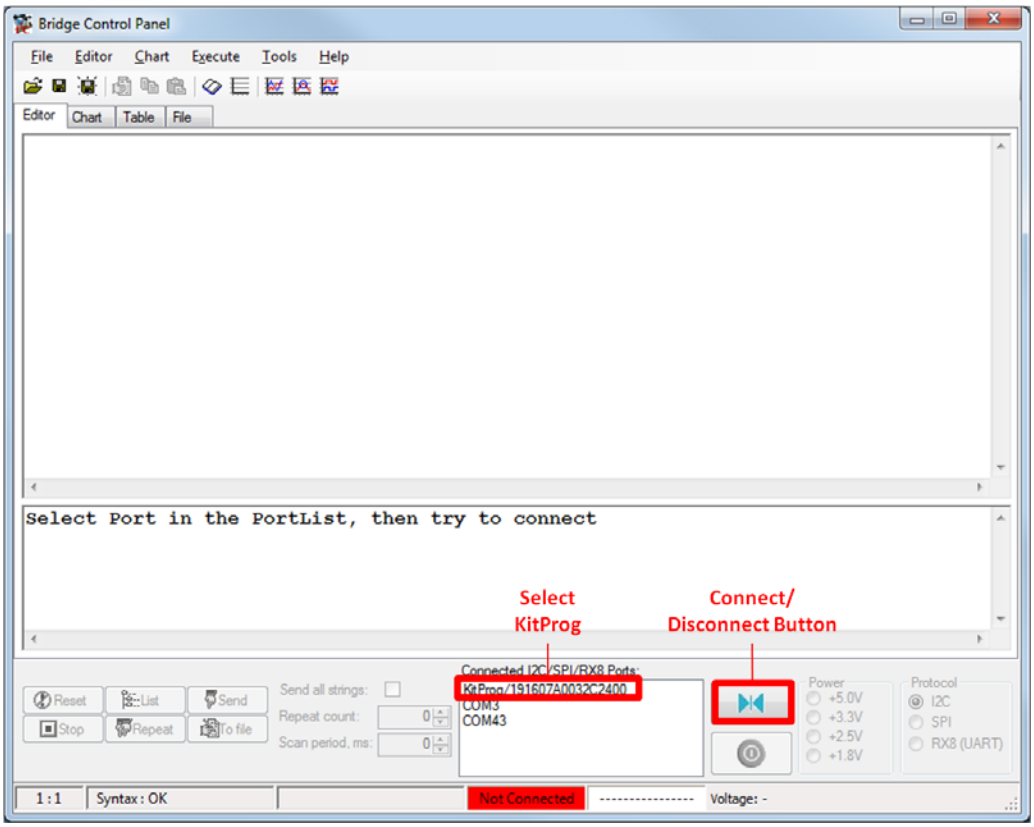

**注意：**J6 插座上的 MiniProg3 （无论带电或不带电）和 KitProg 不能同时连接到板上的 PSoC 4。否则，上述两种方式都会获取设备失败。

## 3.4 USB-I²C 桥接器

PSoC 5LP 还可以作为 USB-I²C 桥接器使用。 PSoC 4 通过 I²C 接口与 PSoC 5LP 进行通信，然后 PSoC 5LP 通过 USB 线缆将数据传输给 PC 上的 USB-I²C 软件工具，即桥接控制面板 （BCP）。

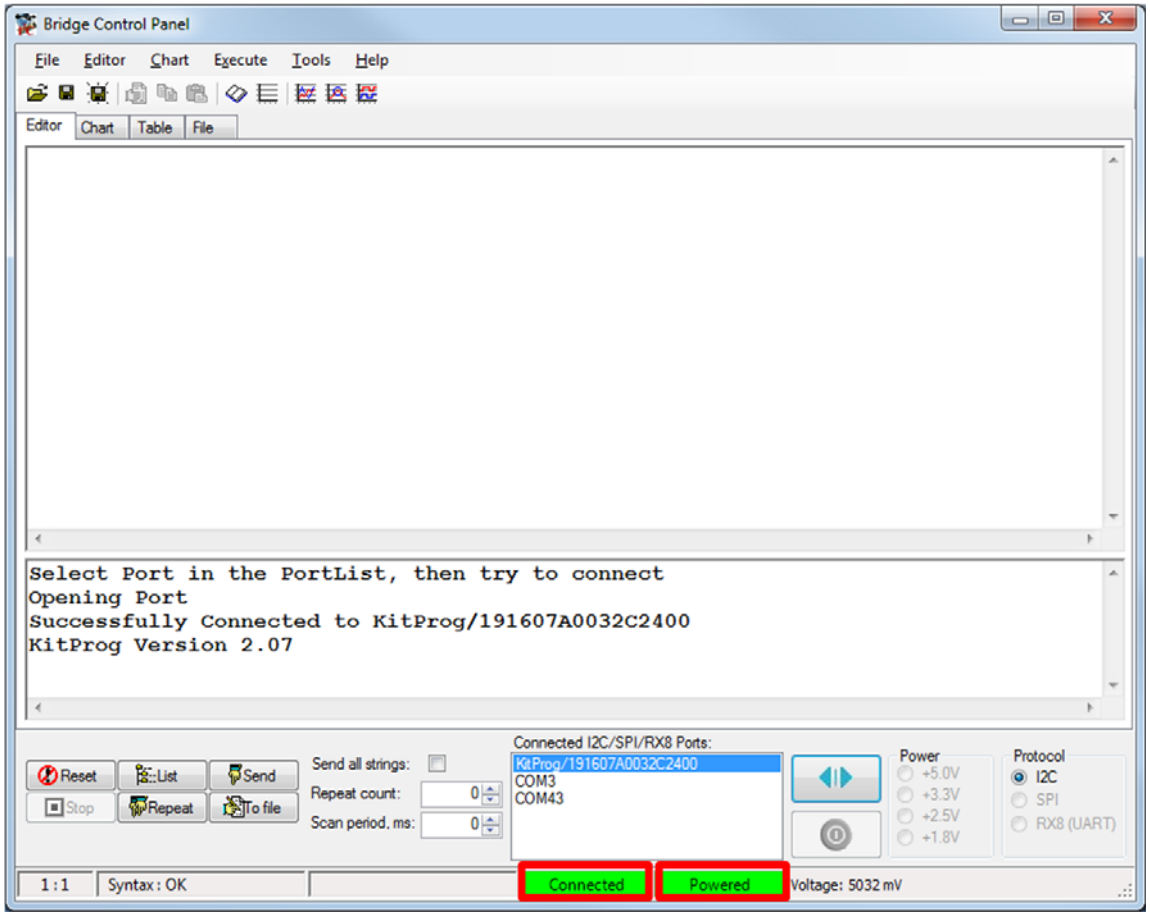
安装 PSoC Programmer 时，BCP 也一起被安装。可以使用该软件进行发送或接收 PSoC 5LP 的 USB-I²C 数据。 USB Mini-B 线缆连接到 PSoC 4000 套件上的插座 J10 时， **KitProg USB-I²C** 器件显示在 BCP 的 **Connected I2C/SPI/RX8 Ports** （已连接的 I2C/SPI/RX8 端口）中，如图 3-10 所示。

图 3-10. 桥接控制面板



要使用 USB-I²C 功能，选择 BCP 中的 **KitProg USB-I²C**（如图 3-10 所示）。成功连接后，**Connected** 和 **Powered** 状态框将显示为绿色 （如图 3-11 所示）。

图 3-11. 桥接器控制面板中 KitProg USB-I² 器件的连接情况




USB-I²C 由 PSoC 5LP 中的 USB 和 I²C 组件来实现。PSoC 5LP 的 SCL（P12_0）和 SDA（P12_1）线分别连接到 PSoC 4 中 I²C 的 SCL （P1_2）和 SDA （P1_3）线。目前， USB-I²C 桥接器支持以下 I²C 速率：50 kHz、 100 kHz、 400 kHz 和 1 MHz。

请参见第 104 页上的将 PSoC 5LP 作为 USB-I2C 桥接器使用，了解如何创建使用 USB-I²C 桥接功能的项目。

## 3.5 USB-UART 桥接器

板载 PSoC 5LP 能够作为一个 USB-UART 桥接器，通过串口 （COM）终端软件从 PSoC 4 器件至 PC 进行传输和接收数据。USB Mini-B 线缆连接到 PSoC 4000 Pioneer 套件的 J10 时，**KitProg USB-UART** 器件将在设备管理器中的**端口（COM & LPT）**内显示。更多有关 USB-UART 功能的信息，请参见第 117 页上的将 PSoC 5LP 作为 USB-UART 桥接器使用。

为了能在串口终端软件中使用 USB-UART 功能，需要 选择相应的 COM 端口作为通信端口，用于传送和接收 COM 终端软件中的数据。

PSoC 5LP 中的 UART 线被连接到插座 J8 的 P12[6] (J8_9) 和 P12[7] (J8_10) 引脚。可以通过将插座 J8 上的 RX 和 TX 引脚连接到其它任何设备的 UART 口以使用该桥接器发送或接收 UART 数据。

**注意：**套件电路板中安装的 PSoC 4000 系列器件不支持全双工 UART ；它仅支持在任意引脚上进行基于软件 UART 的发送操作。在电路板上， PSoC 4000 器件的 P3[0] 通过零欧姆电阻 R57 硬连接到 UART 桥接器的 RX 线。

表 3-2 列出了 USB-UART 桥接器支持的协议。

表 3-2. USB-UART 桥接器所支持的协议

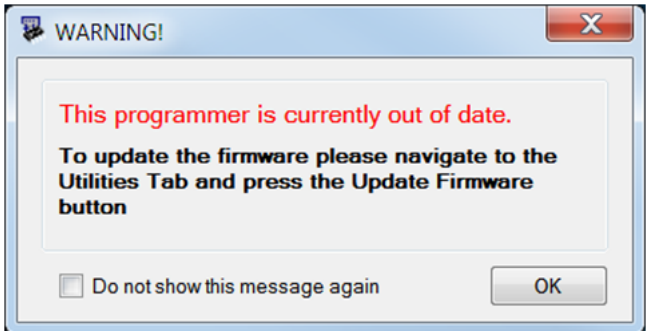
| 参数 | 支持的值 |
|---|---|
| 波特率 | 1200、 2400、 4800、 9600、 19200、 38400、 57600 以及 115200 |
| 数据位 | 8 |
| 奇偶校验 | 无 |
| 停止位 | 1 |
| 流控 | 无 |
| 支持的文件传输协议 | Xmodem、1K Xmodem、Ymodem、Kermit 以及 Zmodem（仅适用于超过 2400 波特率的速度）。 |

## 3.6 更新编程器的固件

可以使用 PSoC 编程器来更新板载编程器和调试器 （KitProg）的固件，即 PSoC 5LP 固件。当有新的固件或当 KitProg 固件被损坏时 （请参见第 166 页上的 KitProg 的 LED 状态）， PSoC 编程器将显示一个警报，表示新的固件可用。

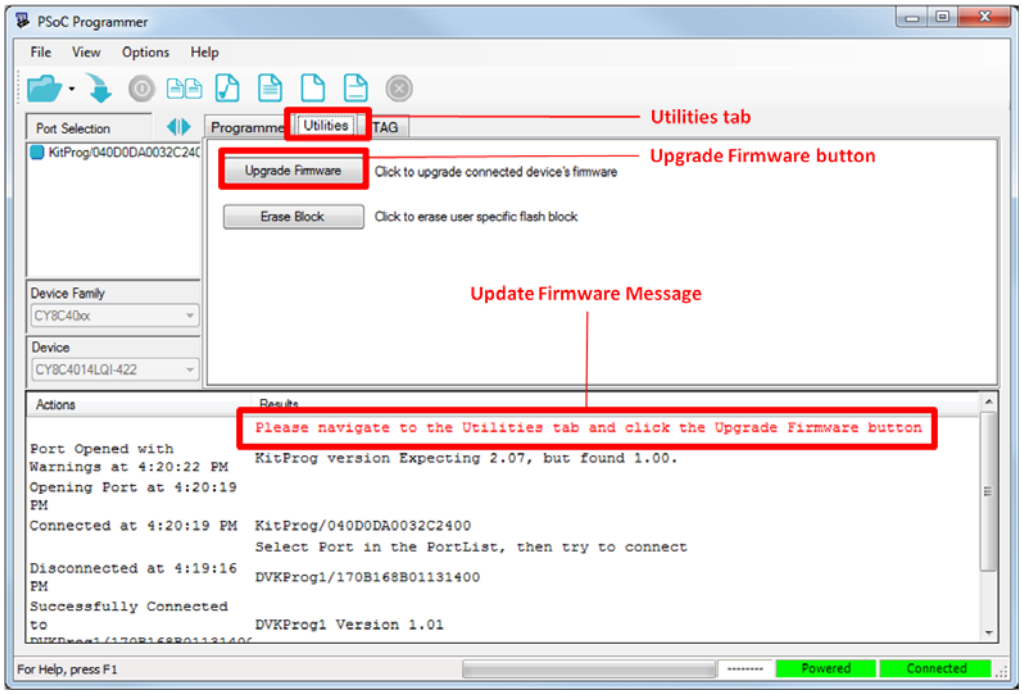
依次选择 **Start > All Programs > Cypress > PSoC Programmer<version>**，打开 PSoC 编程器。PSoC 编程器被打开时，将弹出 WARNING! 窗口，通知当前编程器固件需更新，如图 3-12 所示。

图 3-12. 固件更新警报



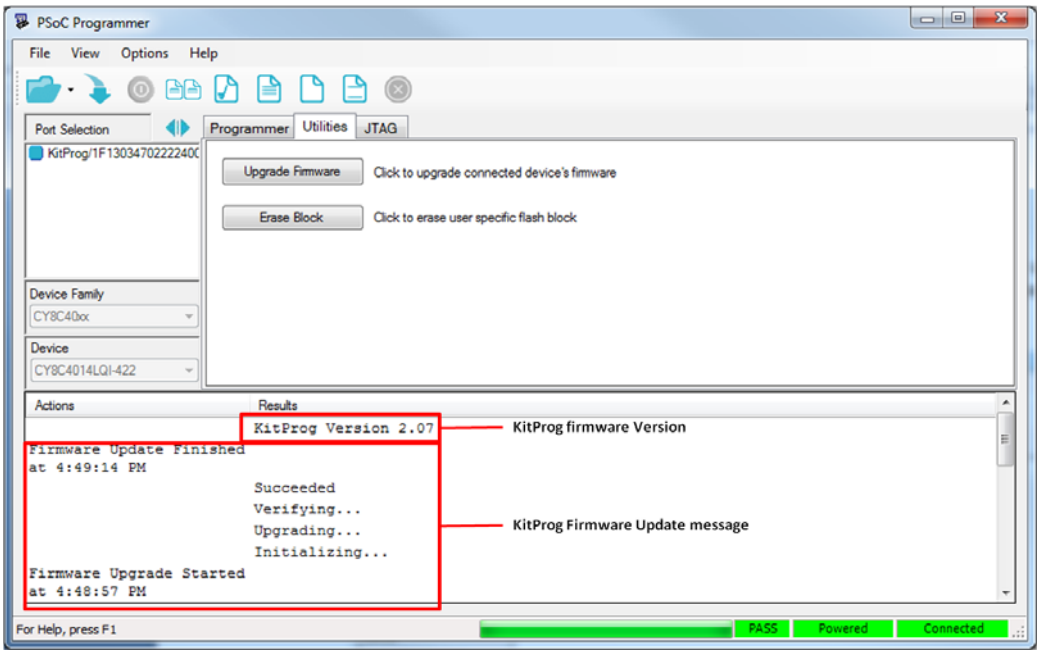
点击 **OK**，关闭该窗口。关闭警告窗口后，**Actions** 和 **Results** 窗口将显示：**Please navigate to the Utilities tab and click the Upgrade Firmware button**，如图 3-13 所示。

图 3-13. PSoC 编程器中的升级固件信息




点击 **Utilities** 选项卡，然后点击 **Upgarde Firmware** 按键。成功升级后，**Actions** 和 **Results** 窗口将显示包含了 KitProg 版本在内的固件更新信息，如图 3-14 所示。

图 3-14. 在 PSoC 编程器中更新的固件



# 4. 硬件

## 4.1 电路板的详细信息

PSoC 4000 Pioneer 套件包含以下各模块：

- CY8CKIT-040 主板 （请参考 图 4-1）
  - PSoC 4 （4000 系列）
  - PSoC 5LP
  - 供电系统
  - 纽扣式电池座 （BT1）
  - 编程接口 （J6 和 J10）
  - 与 Arduino 兼容的插座 （J1、 J2、 J3、 J4 和 J12）
  - PSoC 5LP GPIO 插座 （J8）
  - 接近感应插座 （J5）
  - Pioneer 电路板 LED
  - 按键 （复位按键）
  - 赛普拉斯铁电存储体 （F-RAM）
- CY8CKIT-040 CapSense 触控扩展板 （请参考图 4-2）

**注意：**默认情况下不会填充编程插座 J7。

图 4-1. CY8CKIT-040 — 主板的详细信息

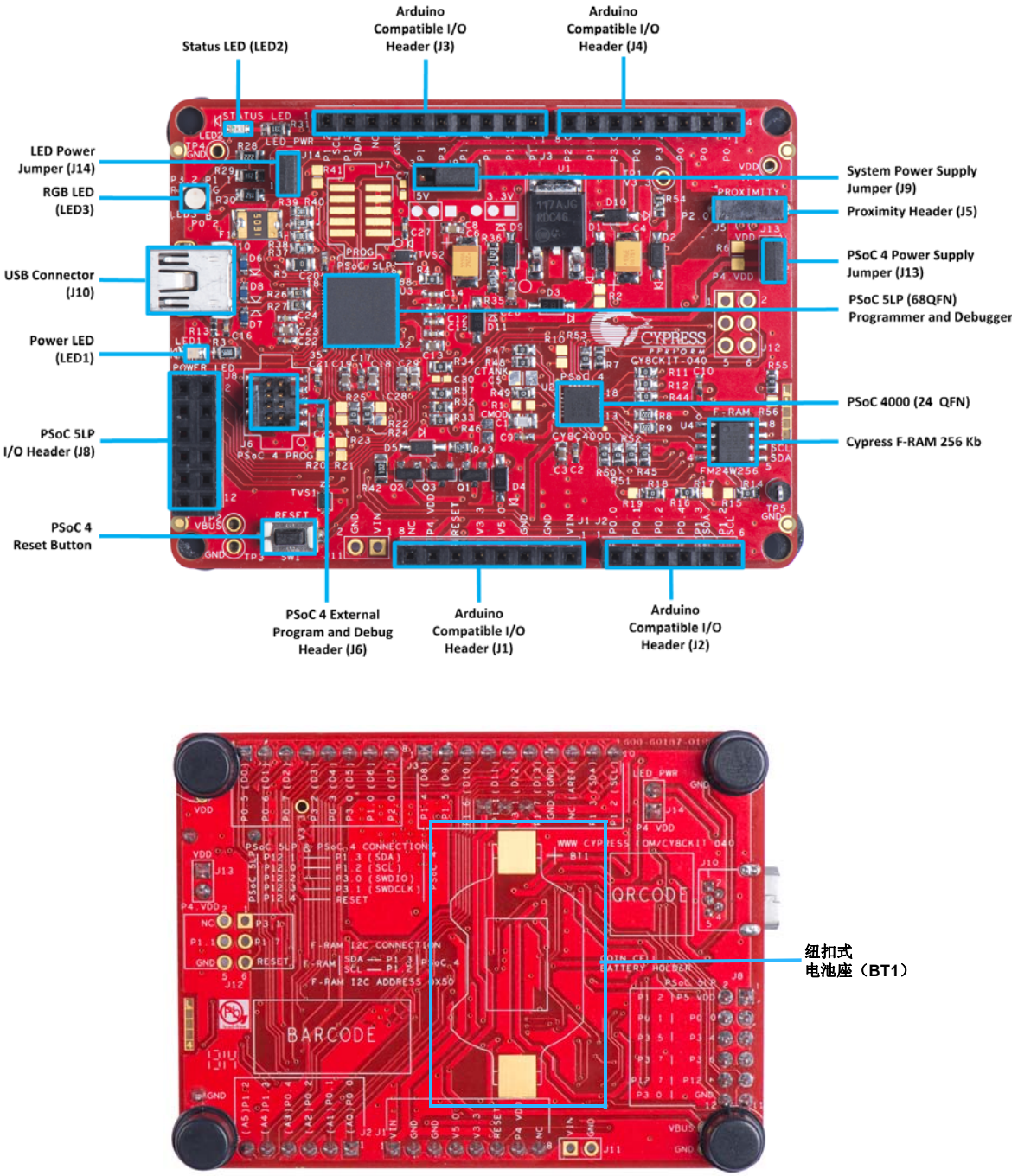


图 4-2. CY8CKIT-040 CapSense 触控扩展板的详细信息

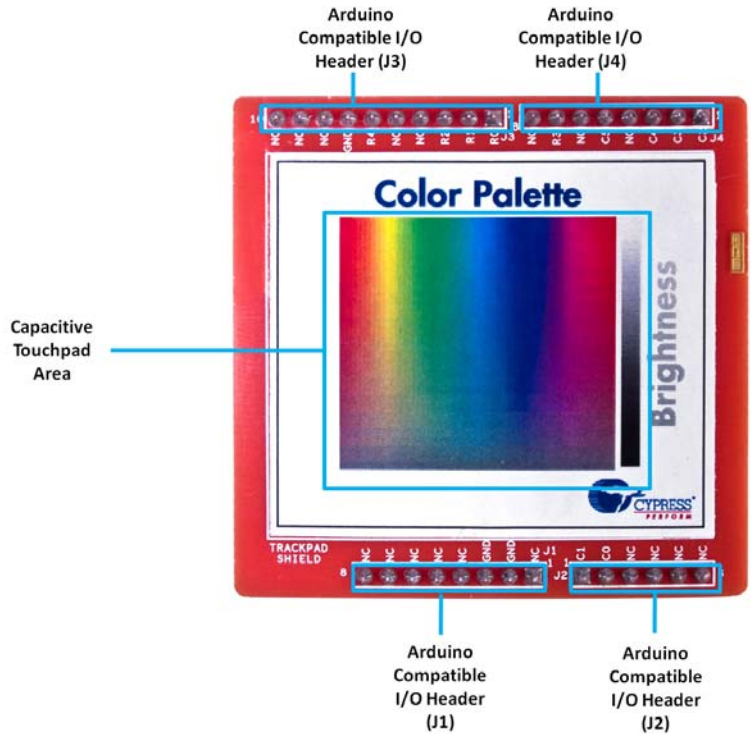
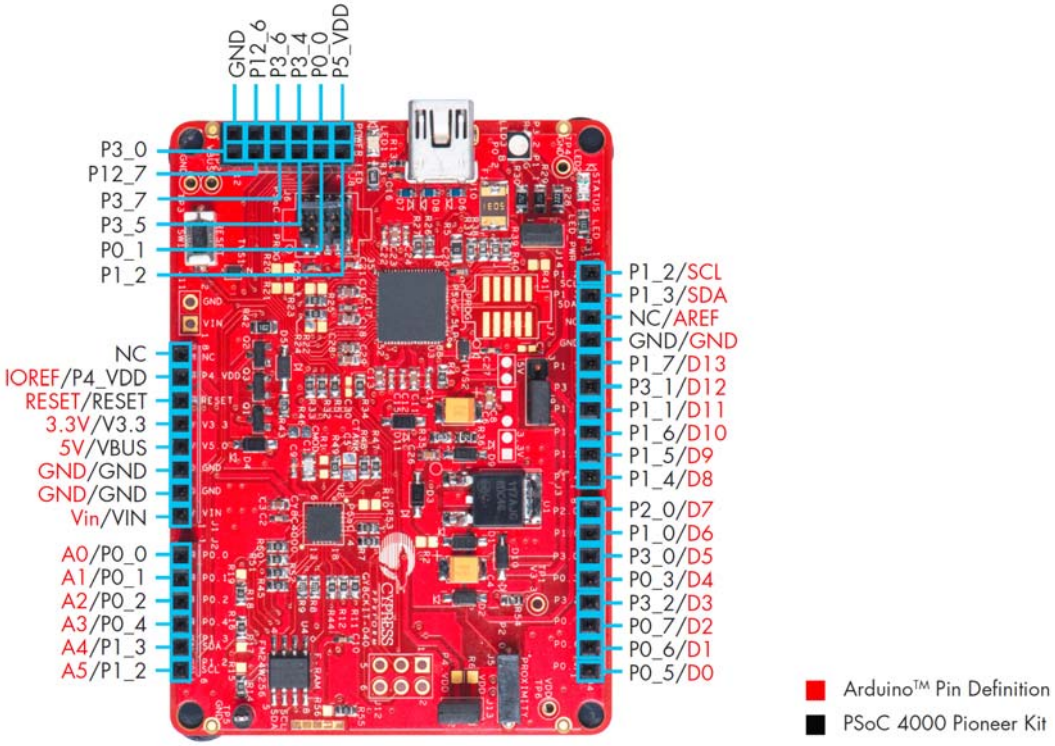


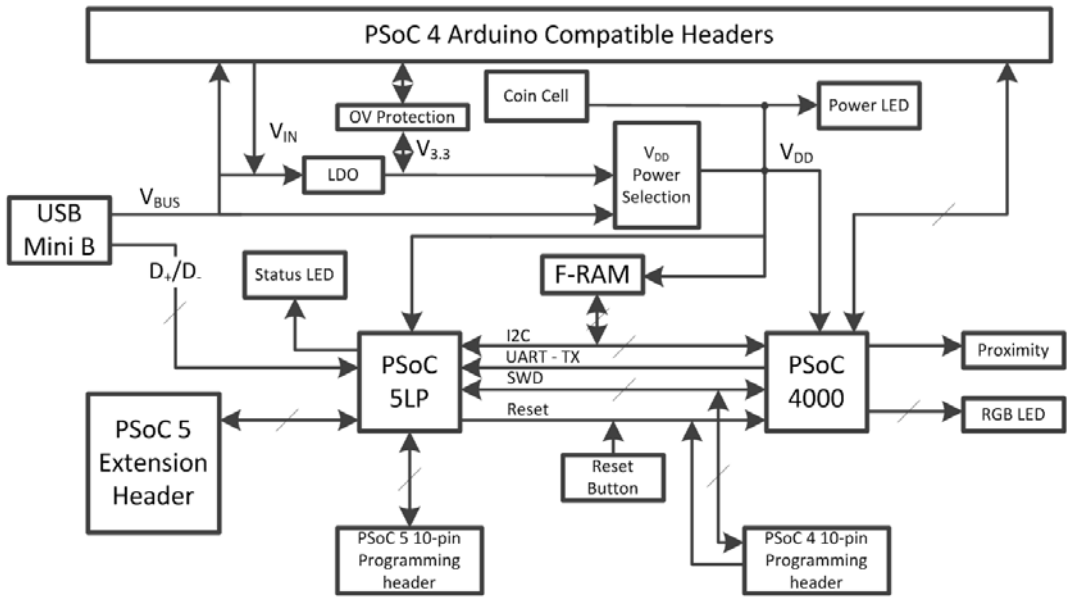

图 4-3. PSoC 4000 Pioneer 套件上引脚的映射情况



## 4.2 框图

本章节介绍的是 PSoC 4000 Pioneer 套件的架构说明，如图 4-4 中所示。

图 4-4. 框图




PSoC 4 是赛普拉斯的新一代可编程片上系统器件，用于嵌入式应用。 PSoC 4000 系列是 PSoC 4 平台中最小的成员，可支持 CapSense、TCPWM、I²C 主设备或从设备，并支持多达 20 个 GPIO。PSoC 4000 是一个低成本的入门级 PSoC4 器件，用于代替过时的和 / 或专有的 8 位和 16 位 MCU。带有 ARM Cortex-M0 内核的 PSoC 4000 提供了 32 个可编程外设，包括 CapSense 在内。

该套件集成了板载 PSoC 5LP 器件。通过 USB 进行通信，该器件使用 SWD 对 PSoC 4 进行编程和调试。 PSoC 5LP 还可以作为 USB-I²C 桥接器和 USB-UART 桥接器运行，也可以用于开发基于 PSoC 5LP 的应用 （请参考 第 129 页上的开发 PSoC 5LP 的应用 ）。

该套件包括一个 RGB 三色 LED、一个状态 LED 和一个电源 LED。RGB LED 连接在 PSoC 4 上，状态 LED 则连接到 PSoC 5LP。该套件还包括一个连接到 PSoC 4 XRES 的复位按键、一个 CapSense 接近感应插座，和一个具有 6x5 矩阵感应单元的触控电路板 （该电路板是与 Arduino 兼容的扩展板，可用于开发基于触摸的应用）。各个 PSoC 4 引脚被连接至该套件中的 J1 ~ J4 插座上，以支持 Arduino 扩展板。 PSoC 5LP 引脚则连接至插座 J8，以允许使用板载 PSoC 5LP 开发自定义应用。

PSoC 4000 Pioneer 套件可通过 USB Mini-B、Arduino 兼容插座、外部电源或可选的纽扣式电池供电。低压差 （LDO）电压调节器将输入电压调节为 3.3 V。通过适当地将跳线器插入到电压选择插座 J9 上，您可以选择使用 VBUS （5 V）或 3.3 V。 VDD 也可以由放置在 BT1 纽扣式电池座的纽扣式电池供电。纽扣式电池所提供的电压会通过二极管 （D11）直接连接到 VDD 导线。请确保 VDD 的电压不能超过器件的 VDD 电压规范 （1.8 V 至 5.5 V）。

## 4.3 套件组件的详细信息

### 4.3.1 CY8CKIT-040 主板组件

#### *4.3.1.1 PSoC 4*

该套件使用了 PSoC 4000 系列器件。PSoC 4 是基于 ARM Cortex-M0 CPU 的可编程嵌入式系统控制器的架构。PSoC4 为嵌入式应用提供了一个可编程的平台。CY8C40 系列是 PSoC 4 器件系列中最小的成员，并与 PSoC 4 中更大的成员向上兼容。

更多有关信息，请参考 PSoC 4 网页和 PSoC 4000 系列的数据手册。

特性

- **32 位 MCU 子系统**
  - 16 MHz ARM Cortex-M0 CPU
  - 高达 16 KB 并带有读取加速器的闪存
  - 可达 2 KB 的 SRAM 容量
- **可编程模拟资源**
  - 用于通用目的或电容式感应应用的两个电流 DAC （IDAC）
  - 带有内部参考电压的低功耗比较器
- **低功耗操作模式的电压范围：1.71 V ~ 5.5 V**
  - 通过中断和 I2C 地址检测唤醒的深度睡眠模式
- **电容式感应**
  - 赛普拉斯的电容式 Sigma-Delta（CSD）电容感应技术提供了一流的信噪比（SNR）和防水功能
  - 通过赛普拉斯提供的软件组件可以更容易地实现电容式感应设计
  - 自动硬件调校 （SmartSense™）
- **串行通信**
  - 多主设备 I2C 模块，在深度睡眠模式下可以进行地址匹配检查，并在匹配时唤醒设备。
- **计时和脉宽调制**
  - 一个 16 位定时 / 计数 / 脉宽调制器 （TCPWM）模块
  - 支持中心对齐、边沿对齐和伪随机模式
  - 基于比较器触发的停止 （Kill）信号可用于电机驱动以及其它高可靠性的数字逻辑应用
- **多达 20 个可编程的 GPIO 引脚**
  - 封装：24 引脚 QFN、 16 引脚 SOIC、 16 引脚 QFN 和 8 引脚 SOIC
  - 端口 0、 1 和 2 上的 GPIO 引脚均可用作 CapSense 或其他功能
  - 可对驱动模式、强度和转换速率进行编程
- **PSoC Creator 设计环境**
  - 集成设计环境 （IDE）提供了原理图设计输入和编译 （包括模拟和数字自动布线）
  - 应用编程接口 （API 组件）可用于所有固定功能和可编程的外设

- **行业标准工具的兼容性**
  - 输入原理图后，可以使用基于 ARM 的行业标准软件开发工具进行开发

#### 4.3.1.2 *PSoC 5LP*

使用板载 PSoC 5LP（CY8C5868LTI-LP039）对 PSoC 4 进行编程及调试。PSoC 5LP 通过 USB Mini-B 连接到 PC 上的 USB 端口， PSoC 5LP 还连接到 PSoC 4 的 SWD 接口。

PSoC 5LP 是一个系统级解决方案，可在单芯片中提供微控制器单元 （MCU）、存储器、模拟和数字外设等功能。 CY8C58LPxx 系列提供了一种新型的信号采集、信号处理和控制方法，并且具有高精度、高带宽和高灵活性等特点。其模拟功能涵盖了从热电偶信号 （类似直流电压）到超声波信号广泛的信号范围。更多有关信息，请参考 PSoC 5LP 网站。

特性

- 32 位 ARM Cortex-M3 CPU 内核
  - 工作频率高达 DC 至 67 MHz
  - 闪存程序存储器空间最高可达256 KB，具有10万次的写循环，并且具有20年保留时间以及多种安全特性
  - 高达 32 KB 的纠错码 （ECC）或系统配置存储空间
  - SRAM 容量可达 64 KB
  - 容量为2 KB的电可擦除可编程只读存储器（EEPROM），具有 100 万次的擦除次数以及 20 年的保留时间的特性
  - 24 通道的直接存储访问 （DMA）支持通过多层 AHB 总线访问数据

  可编程链式描述符和优先级

  支持高带宽 32 位传输
- 低电压、超低功耗
  - 较宽的工作电压范围：0.5 V 至 5.5 V
  - 高效升压调节器 （0.5 V 输入， 1.8 V ~ 5.0 V 输出）
  - 频率为 6 MHz 时，电流为 3.1 mA （电压为 2.7 ~ 5.5 V）
  - 低功耗模式包括：

  2 µA 睡眠模式，提供了实时时钟 （RTC）和低压检测 （LVD）中断

  在休眠模式下，电流为 300 nA （保留 RAM 数据）
- 多功能 I/O 系统
  - 28 至 72 个 I/O （62 个 GPIO、 8 个 SIO 和 2 个 USBIO）
  - 可从任意的 GPIO 连接到任意一个数字或模拟外设
  - 所有 GPIO 均具有 LCD 直接驱动功能，最多能够驱动 46×16 段
  - 所有 GPIO 均提供 CapSense 支持
  - 1.2 V 至 5.5 V 的 I/O 接口电压，多达 4 个电压域
  - 所有引脚或端口支持可屏蔽的独立 IRQ
  - 带施密特触发器的 TTL 电平输入
  - 所有的 GPIO 均可配置为开漏高 / 低电平、电阻上拉 / 下拉、高阻态或者强驱动输出
  - 上电复位 （POR）时可对 GPIO 引脚的状态进行配置
  - SIO 具有 25 mA 的灌电流能力

- 数字外设
  - 20 至 24 个基于可编程逻辑器件 （PLD）的通用数字模块 （UDB）
  - 全速 CAN 2.0b 接口， 16 个 RX 缓冲区， 8 个 TX 缓冲区
  - 12 Mbps 的全速 （FS） USB 2.0 （采用内部振荡器）
  - 4 个 16 位可配置定时器、计数器和 PWM 模块
  - 67 MHz、24 位固定点的数字滤波器模块（DFB），用于实现有限脉冲响应（FIR）和无限脉冲响应 （IIR）滤波器
  - 标准外设库

  8、 16、 24 和 32 位定时器、计数器和 PWM

  串行外设接口 （SPI）、通用异步发送接收器 （UART）以及 $I^2C$

  除了这些，还有其它更多外设包含在 PSoC Creator 集成开发环境的组件库中。
  - 高级外设库

  循环冗余校验 （CRC）

  伪随机序列 （PRS）发生器

  本地互连网络 （LIN）总线 2.0

  正交解码器
- 模拟外设 （1.71 V ≤ VDDA ≤ 5.5 V）
  - 在 –40 °C 到 +85 °C 的温度范围内，内部参考电压为 1.024 V ±0.1%
  - 具有 8 至 20 位分辨率的可配置 Delta-Sigma 模数转换器 （ADC）

  采样率最高可达 192 ksps

  可编程增益级：×0.25 至 ×16

  12 位模式、 192 ksps、 66 dB 信噪比和失真比 （SINAD）、 ±1 位 INL/DNL

  16 位模式、 48 ksps、 84 dB SNR、 ±2 位 INL、 ±1 位 DNL
  - 最多支持两个 SAR ADC，分辨率为 12 位，最高采样率为 1 Msps
  - 四个 8 位 8 Msps 电流 IDAC 或 1 Msps 电压 VDAC
  - 四个响应时间为 95 ns 的比较器
  - 四个驱动能力为 25 mA 的无约束运算放大器
  - 四个可配置多功能模拟模块；典型配置包括可编程增益放大器 （PGA）、互阻放大器（TIA）、混频器，以及采样和保持
  - 支持 CapSense
- 编程、调试和跟踪
  - JTAG （4 线）接口， SWD （2 线）接口，单线跟踪 （SWV），以及 TRACEPORT 接口
  - Cortex-M3 闪存修补和断点 （FPB）模块
  - Cortex-M3 Embedded Trace Macrocell™ （ETM™）可生成指令跟踪流
  - Cortex-M3 数据观察点和跟踪 （DWT）可生成数据跟踪信息
  - Cortex-M3 Instrumentation Trace Macrocell （ITM）可用于 printf-style 调试
  - DWT、 ETM 和 ITM 模块可通过 SWV 或 TRACEPORT 同片外调试与跟踪系统进行通信
  - 支持通过 $I^2C$、 SPI、 UART、 USB 以及其他接口进行 Bootloader 编程

- 高精度、可编程时钟
  - 在整个温度和电压范围内，内部振荡器的频率范围为 3 至 62 MHz
  - 频率范围为 4 至 25 MHz 的晶振，能够实现晶振 PPM 精度
  - 能够生成高达 67 MHz 的内部 PLL 时钟
  - 32.768 kHz 时钟晶振
  - 频率分别为 1 kHz、 33 kHz 和 100 kHz 的低功耗内部振荡器

有关详细信息，请参考 CY8C58LPxxx 系列数据手册。

#### 4.3.1.3 *供电系统*

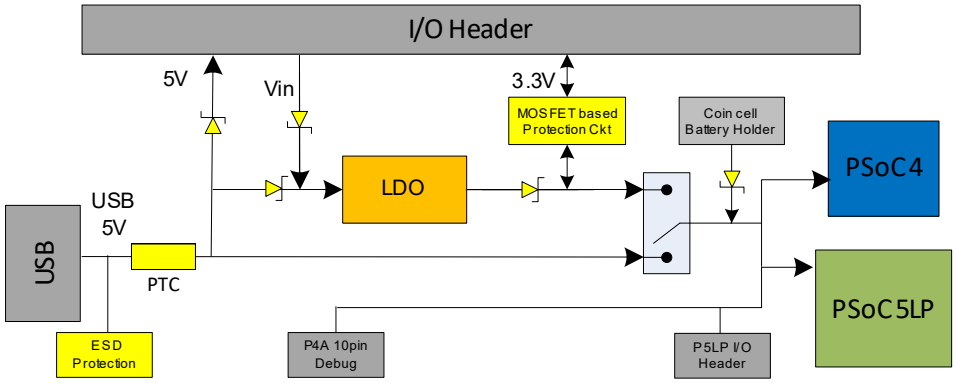
该电路板支持多种形式的电源供电，输入的电源可以来自：

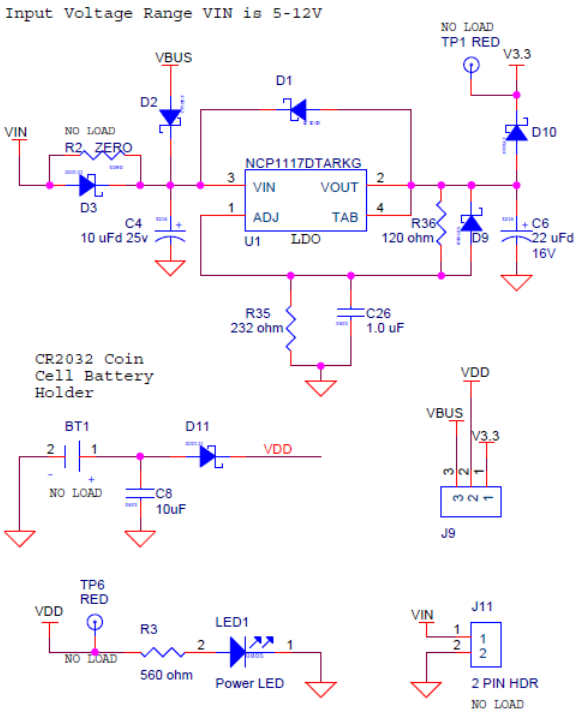
- 使用板载 USB 编程插座 J10 的 5 V 电源
- 使用插座 J1_01 （J1 上的 VIN）的 Arduino 扩展板的 5 V 至 12 V 电源
- 使用插座 J6 或 J7 的板载 SWD 编程的电源 — VTARG
- VIN （J11）（默认情况下未安装）
- 纽扣式电池 （BT1）（默认情况下未安装）

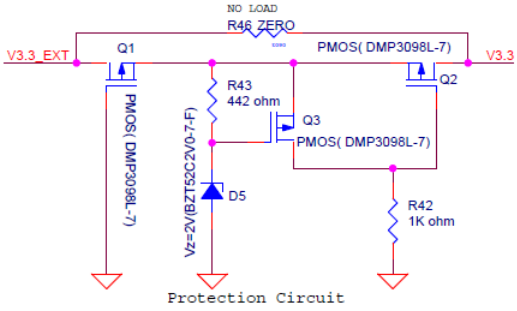
PSoC 4 和 PSoC 5LP 由电压为 3.3 V 或 5 V 的电源供电。通过 J9 跳线器，可以选用 3.3 V 或 5 V 的电源。该电路板可以为输入 / 输出插座提供 3.3 V 和 5 V 的电压，并且接收来自输入 / 输出插座的 3.3 V 电压 （J9 应为其选择 3.3 V）。外部电源可通过插座 VIN （J11）给该电路板供电；其中 VIN 的电压范围为 5 V 到 12 V。 LDO 电压调节器将 VIN 值调制为 3.3 V。图 4-5 显示的是电源框图和电路。此外，纽扣式电池座 （BT1）同样可以直接给 VDD 线供电。纽扣式电池座所支持的电压范围是 1.8 V 到 5.5 V （PSoC 4000 系列的 VDD 规范）。在电路板上，BT1 纽扣式电池座默认不被安装。BT1 可使用 Keystone 电子公司的 BU2032SM-BT-GTR。该器件支持 CR2032 型的纽扣式电池。请参考 第 167 页上的材料清单，了解有关可用于 BT1 的其它器件的详细信息。

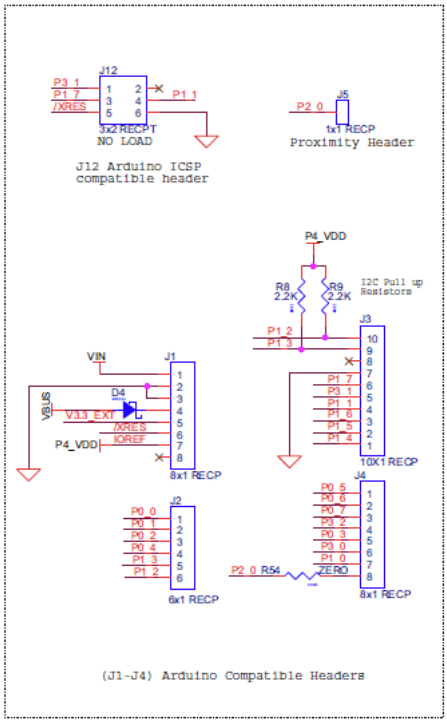
**注意：**5 V 电域直接由 USB （VBUS）供电。因此，该电域的电压未经调节。

图 4-5. 具有保护电路的电源框图和原理图




Input Voltage Range VIN is 5-12V
NO LOAD
TP1 RED
V3.3
VBUS
D1
D2
D10
VIN
NO LOAD
R2 ZERO
NCP1117DTARKG
VIN
VOUT
ADJ
TAB
LDO
U1
D3
C4
10 uFd 25v
R36
120 ohm
D9
C6
22 uFd
16V
R35
232 ohm
C26
1.0 uF
CR2032 Coin
Cell Battery
Holder
BT1
D11
VDD
NO LOAD
C8
10uF
VDD
VBUS
V3.3
J9
TP6
RED
VDD
R3
LED1
NO LOAD
560 ohm
Power LED
VIN
J11
2 PIN HDR
NO LOAD
Power Supply


NO LOAD
R46 ZERO
V3.3_EXT
Q1
PMOS( DMP3098L-7)
V3.3
Q2
R43
442 ohm
Q3
PMOS( DMP3098L-7)
PMOS( DMP3098L-7)
D5
Vz=2V(BZT52C2V0-7-F)
R42
1K ohm
Protection Circuit


J12
P3_1
P1_7
/XRES
P1_1
3x2 RECPT
NO LOAD
J12 Arduino ICSP
compatible header
P2_0
J5
1x1 RECP
Proximity Header
P4_VDD
R8
2.2K
R9
2.2K
I2C Pull up
Resistors
P1_2
P1_3
J3
10X1 RECP
VIN
J1
D4
VBUS
V3.3_EXT
/XRES
IOREF
P4_VDD
8x1 RECP
J2
6x1 RECP
J4
P2_0 R54
ZERO
8x1 RECP
(J1-J4) Arduino Compatible Headers

### 保护电路

该电路具有反向电压、过电压、短路和过电流保护等功能，如图 4-5 所示。

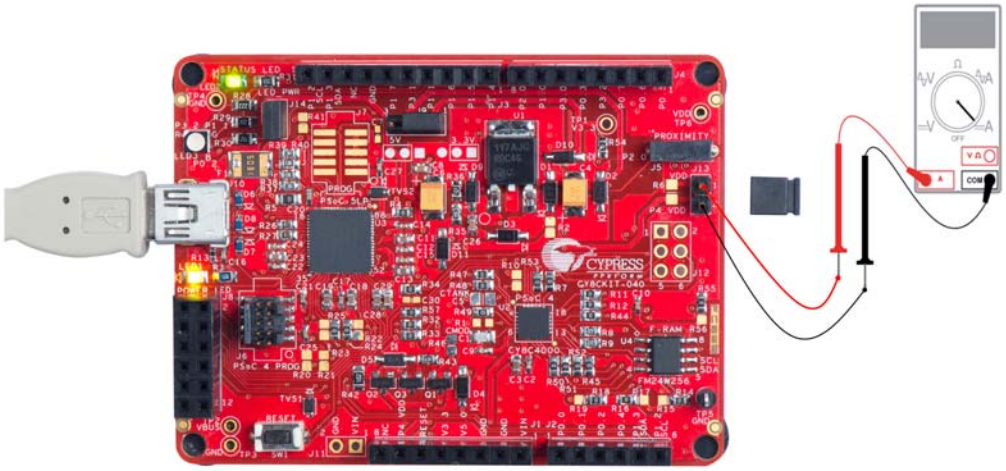
- 通过肖特基二极管 （D4）可确保不会通过输入 / 输出插座供电给 5 V 电域。
- 通过串接保护二极管 （D3）可确保不会通过 VIN （来自输入 / 输出插座的电源）供电给 USB。
- 通过肖特基二极管 （D10）可确保不会通过输入 / 输出插座中的 3.3 V 电源供电给 LDO。
- 串接保护二极管 （D3）可确保不能从 VIN 为电压调节器输入提供反向电压。
- 通过与 PTC 恢复式保险丝相连接，可以避免计算机的 USB 端口发生短路和过电流情况。
- 基于 MOSFET 的保护电路为 3.3 V 的电源提供过电压和反向电压保护。PMOS Q1 可使电路板组件免于发生反向电压情况。PMOS Q2 可使 PSoC 免于发生过电压情况。当电压超过 3.6 V 时，PMOS Q2 将被关闭以保护 PSoC 4。
- 调整 LDO 的输出电压，使其考虑在肖特基二极管上的电压降，仍能提供 3.3 V 的电压。
- 使用大小为 0 Ω 的电阻焊接到 R46 可旁路基于 MOSFET 的保护电路。

### 测量 PSoC 4 电流消耗的流程

下列两种方法都用于测量 PSoC 4 器件的电流消耗。

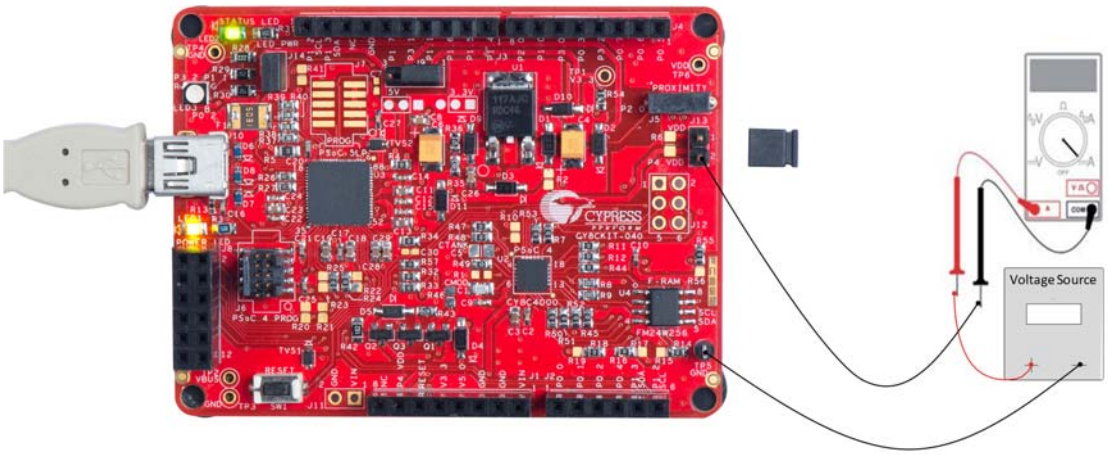
- 当该电路板由 USB 端口 （J10）供电时，移除跳线器 J13 并连接一个电流表，如图 4-6 所示。

图 4-6. 由 USB 端口供电给 PSoC 4 时的电流测量



- 当同时使用了 PSoC 4 的独立电源和 USB 来供电时 （USB 上的电压调节器输出必须在独立电源 ±0.5 V 的范围内），需要移除跳线器 J13。将外部电源的正端连接到电流表的正端，并将该电流表的负端连接到跳线器 J13 的下面的引脚 （P4.VDD）。图 4-7 显示了所需的连接。

图 4-7. 单独供电给 PSoC 4 时进行的电流测量




**注意：**RGB 三色 LED 仅由 PSoC 4 VDD 供电。移除跳线器 J14，以测量 PSoC 4 单独消耗的功率。

#### 4.3.1.4 *编程接口*

该套件允许在两种模式下对 PSoC 4 进行编程及调试：

- 第 22 页上的使用板载 PSoC 5LP 的编程器和调试器
- 第 23 页上的使用 CY8CKIT-002 MiniProg3 编程器和调试器

### 4.3.1.5 *Arduino 兼容的插座（J1、 J2、 J3、 J4 及 J12）*

该套件具有 5 个 Arduino 兼容的插座（J1、J2、J3、J4 及 J12）可以根据 Arduino 扩展板的硬件来开发各种应用。该套件还提供了一个和 Arduino 电路板兼容的触控板。

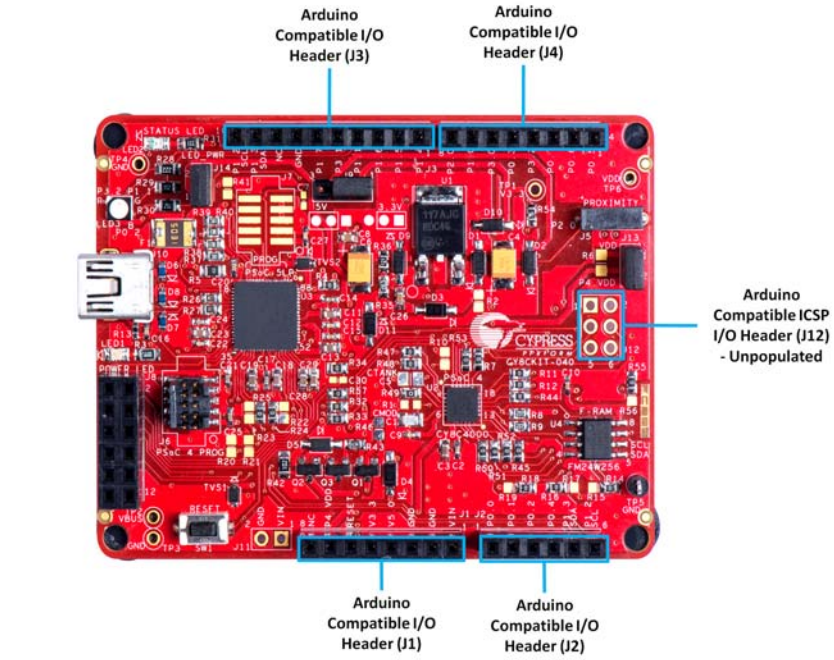
图 4-8. Arduino 插座

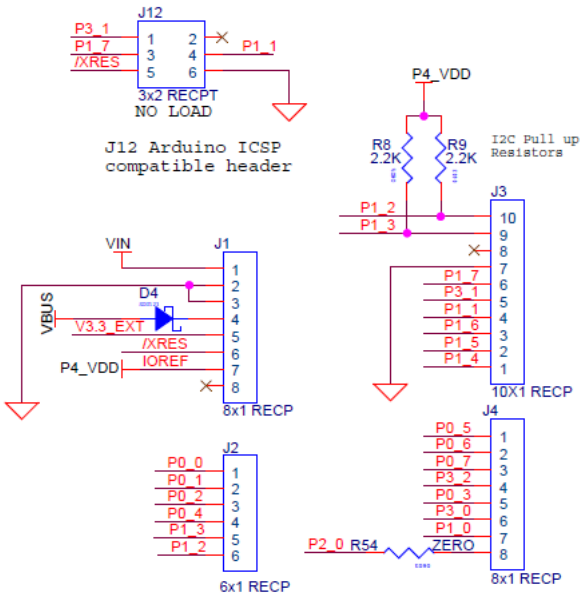
**注意：**图 4-8 中的图形 LCD 扩展板仅供参考，它不是该套件的一部分。

- J1 插座包含用于复位和电源线的输入 / 输出引脚。
- J2 插座是一个模拟端口。由于 PSoC 4000 系列中的模拟支持有限，因此它仅包含了通用的数字 I/O 引脚。
- J3 插座主要作为一个数字端口使用。它包含用于 PWM、 $I^2C$ 及通用数字的 I/O 引脚。
- J4 插座也是一个数字端口。
- J12 是一个用于 SPI 接口的 Arduino ICSP 兼容插座。该插座未被组装。有关插座器件编号的详细信息，请参考物料清单章节中的 “ 空载组件 ” 部分中的内容。

**注意：**PSoC 4000 系列不支持硬件 SPI，但可以使用固件 bit banging 在所有的引脚上实现 SPI 主设备。

图 4-9. Arduino 兼容的插座




(J1-J4) Arduino Compatible Headers

## 未被组装的插座 J12 的功能

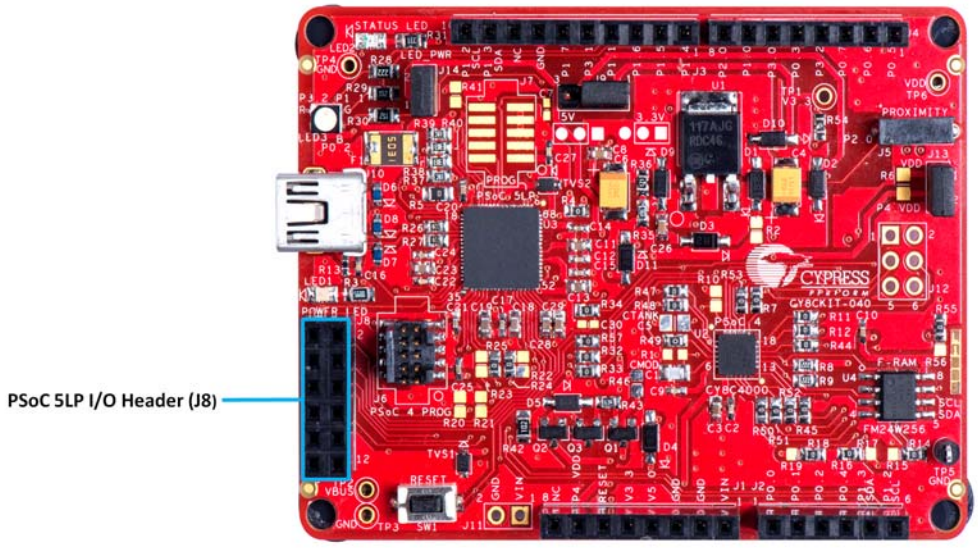
J12 插座是一个支持 Arduino 扩展板的 2 x 3 插座。它用于一小部分的扩展板，在 PSoC 4000 Pioneer 套件上这个插座默认并未安装。

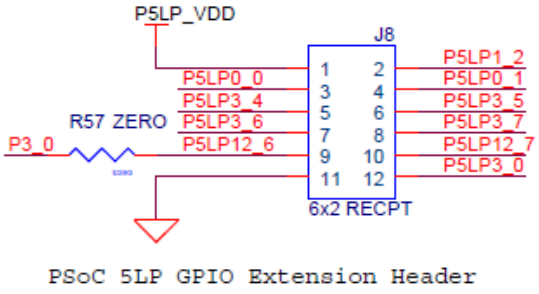
**注意：**J12 插座只能在 5 V 模式下工作。为了保证扩展板的正常功能，需要确保电源跳线器 （J9）设置在 5 V 模式下。

### 4.3.1.6 PSoC 5LP GPIO 插座 （J8）

PSoC 5LP 引脚中的一部分被连接到该插座。有关如何开发自定义应用的详细信息，请参考第 129 页上的开发 PSoC 5LP 的应用 。有关引脚的详细信息，请参考 第 162 页上的引脚分配表 。

图 4-10. PSoC 5LP GPIO 插座 （J8）



### 4.3.1.7 CapSense 电路

该主板包含一个用于连接 CapSense 接近感应导线的插座（J5）（请参考图 4-12）。大小为 2.2 nF 的电容 （C1）位于 CMOD 引脚 （P0[4]）上，用于 CapSense 操作。可安装一个可选电阻 R1，用以将 IDAC 的电流输出转换为电压输出，以提供给非 CapSense 的设计使用。

该电路板可以选择性地支持需要防水功能的 CapSense 设计。可将器件中任意支持 CapSense 的引脚配置为屏蔽信号，以使能防水功能。然而，如果该设计中需要使用屏蔽水库电容，那么需要安装一个大小合适的电容 C5（CTANK），并移除用于将蓝色 LED 连接到 P0_2 的电阻 R30。更多有关 CapSense 的详细信息，请参考 CapSense 设计指南。

**注意：**由于实现触控板和 RGB LED 后 I/O 资源有限，该套件附带的触控扩展板没有演示防水功能，但可以设计一个自定义的扩展电路板，来实现防水功能。

图 4-11. 主板 CapSense 电路

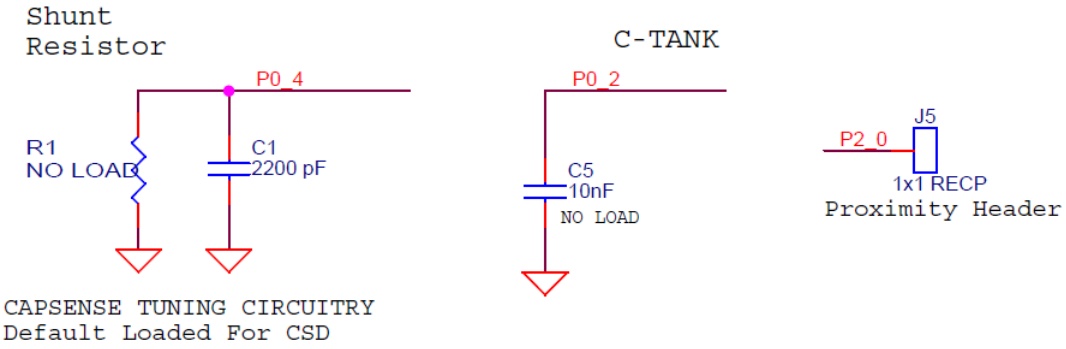
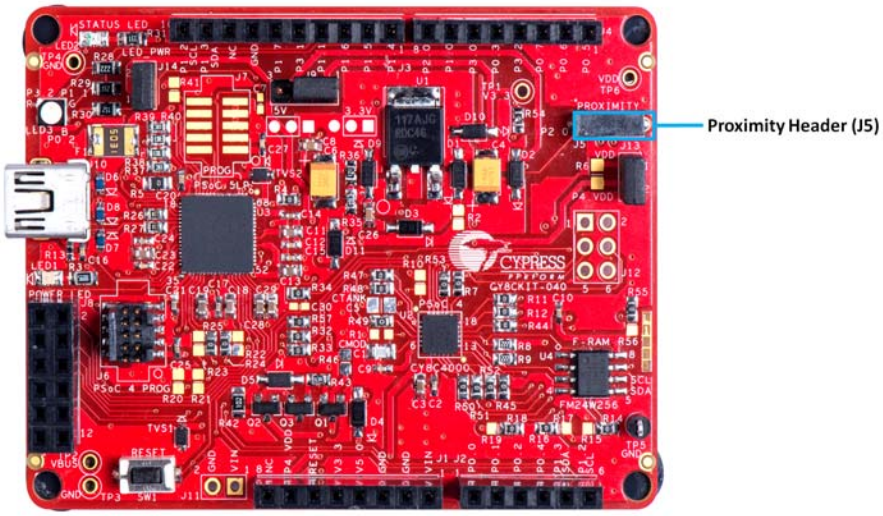



图 4-12. 接近感应插座 （J5）



### 4.3.1.8 电路板上的 LED

PSoC 4000 Pioneer 套件具有 3 个 LED。绿色 LED （LED2）用于指示编程器的状态。有关 LED 指示列表的详细信息，请参考第 166 页上的 KitProg 的 LED 状态 。琥珀色的 LED （LED1）用于指示电路板的电源供电状态。该套件还有一个用户应用可使用的 RGB 三色 LED （LED3），它被连接至 PSoC 4 上的特定引脚。可通过使用跳线器 J14 来使能 / 禁用 RGB LED（LED3）的电源。RGB LED 由 PSoC 4 VDD 供电，因此如果需要准确测量 PSoC 4 的功耗必须移除跳线器 J14，以避免漏电和 LED 供电对测量的影响。

图 4-13 显示了这些 LED 在电路板上的位置。图 4-14 和图 4-15 介绍了有关 LED 原理图的详情。

图 4-13. 电路板上的 LED

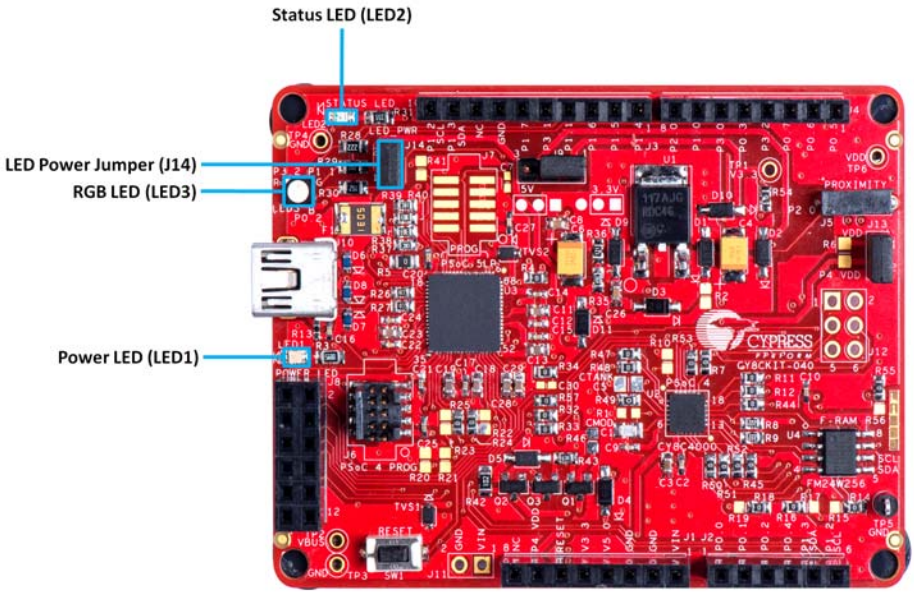



图 4-14. 电源 LED

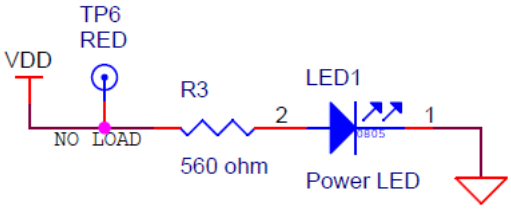
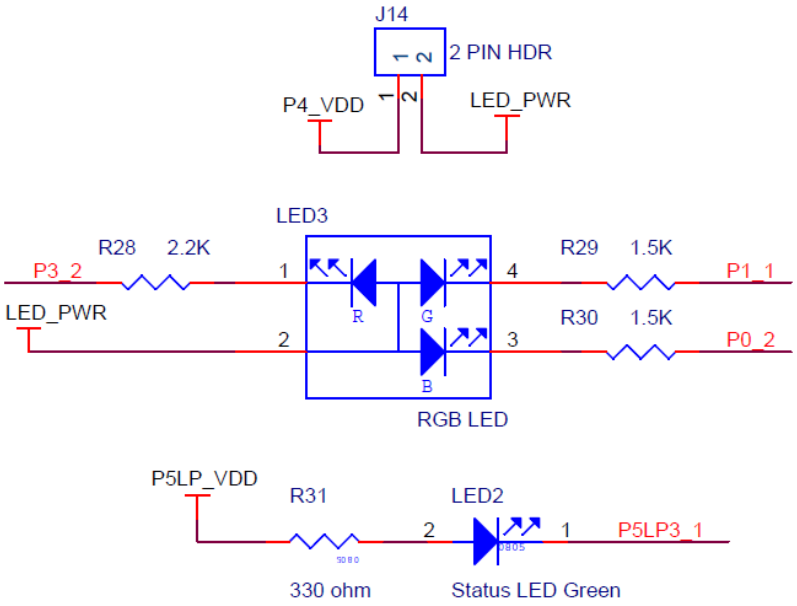


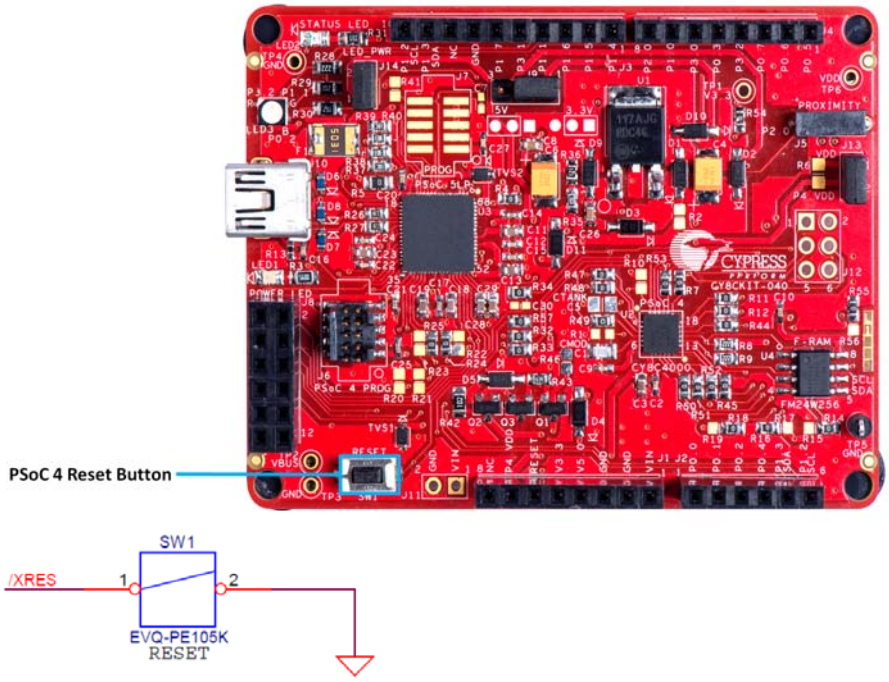
图 4-15. 状态 LED 和 RGB LED




### 4.3.1.9 *按键*

该套件只有一个复位按键，如图 4-16 所示。复位按键被连接到 PSoC 4 上的 XRES 引脚，用于复位 PSoC 4 的板载器件。该按键按下时接地 （低电平有效）。

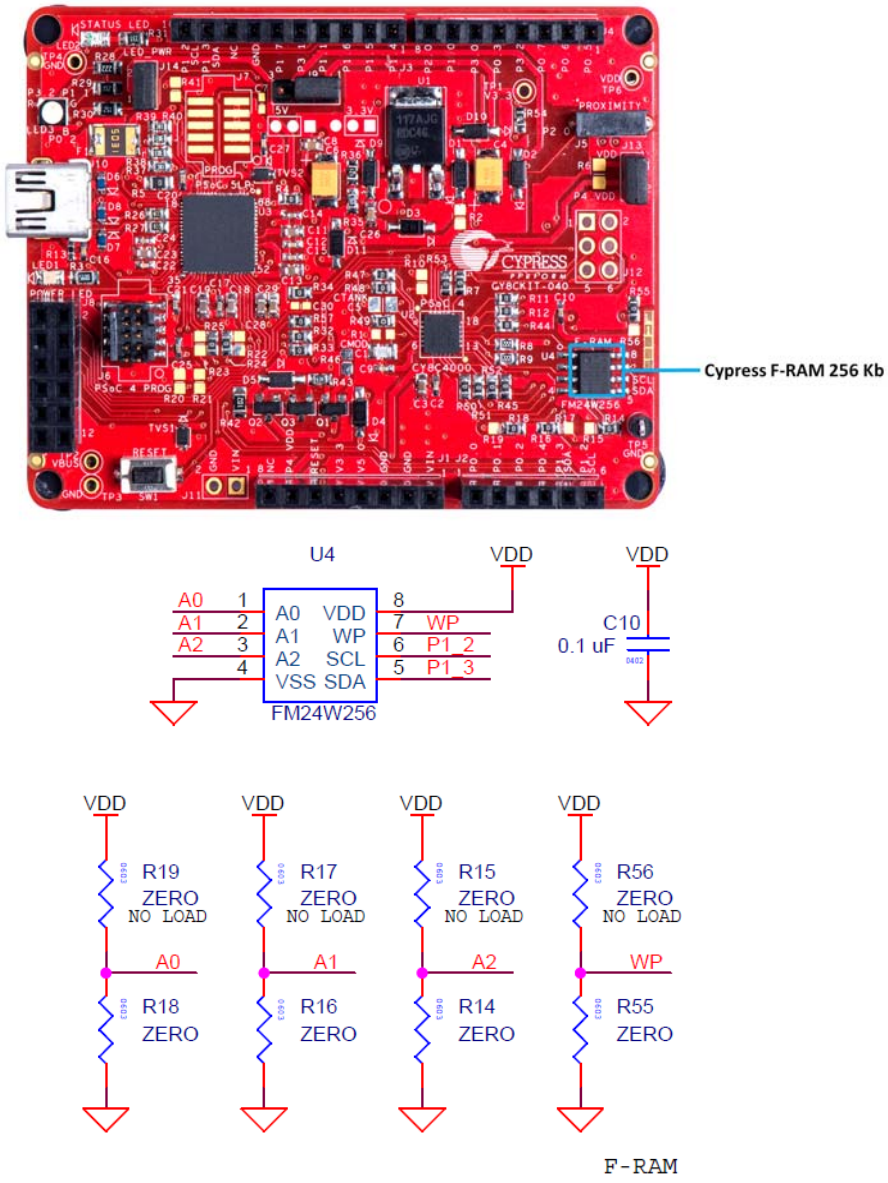
图 4-16. 复位按键



### 4.3.1.10 *赛普拉斯铁电 RAM （F-RAM）*

该主板包含一个 F-RAM 器件（FM24W256）；请参考图 4-17，PSoC 器件（PSoC 5 LP 或 PSoC 4，或这两者）都可以通过 I²C 线访问该器件。该 F-RAM 容量为 256 Kb（32 KB），I²C 速度高达 1 Mbps。F-RAM 器件的 I²C 从设备地址为 7 位宽，并且三个最低有效位 （LSB）可通过物理引脚来配置，在本电路板上被固定连接到 000。电路板上所使用的 F-RAM 器件的地址默认为 0x50。修改 R19/R18、R17/R16 及 R15/R14 电阻对，便可以改变该地址。有关如何使用这三个电阻对来修改 F-RAM 地址的详细信息，请参考第 165 页上的 0 Ω 电阻和空载的使用 。第 112 页上的使用 FM24W256 F-RAM 一节中提供了一个示例实现，用于展示 PSoC 4 如何使用该 F-RAM 器件，以及如何通过 PSoC 5LP USB-I²C 桥接器与桥接控制面板进行通信。

图 4-17. 赛普拉斯所提供的 F-RAM



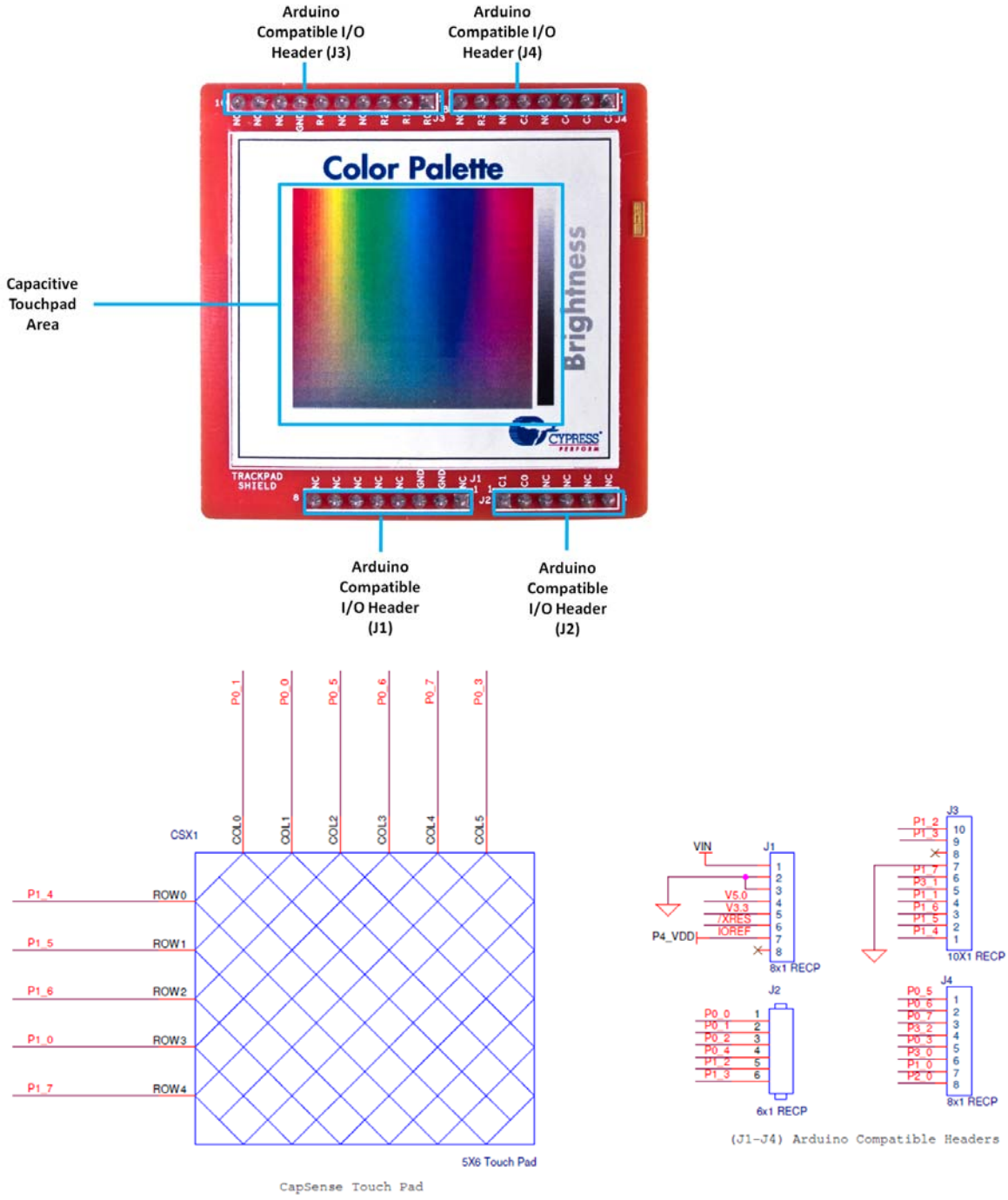
### 4.3.2 CY8CKIT-040 CapSense 触控板扩展电路板

该套件还包含了一个受 Arduino 兼容的 CapSense 触控板扩展电路板。该套件中的触控板是一个具有 6x5 个元素的电容式传感阵列。图 4-18 显示了触控板的引脚映射情况。

CapSense 的调制电容（Cmod）连接到引脚 P0[4]，并且可以选择和一个泄放电阻（R1）进行连接。触控扩展板与 Arduino 相兼容，并且也可使用于 PSoC 4 Pioneer 套件 （CY8CKIT-042）。可以重新设计触控板上的标签，并可将该标签贴在触控板的顶层，用于实现所有的自定义、特定应用的 UI。第 170 页上的触控板标签详情 提供了标签模板。

更多有关 CapSense 的详细信息，请参考 CapSense 设计指南。

图 4-18. CapSense 触控扩展板



# 5. 代码示例

## 5.1 概述

通过本章节中描述的示例项目来介绍 PSoC 4000 器件和板上组件的功能。要访问这些示例，请从套件网页上下载并安装 CD ISO 图像或设置文件，如第 15 页上的安装软件 所述。安装后，示例项目会位于 < 安装目录 >\CY8CKIT-040 PSoC 4000 Pioneer Kit\< 版本号 >\Firmware\ 内。

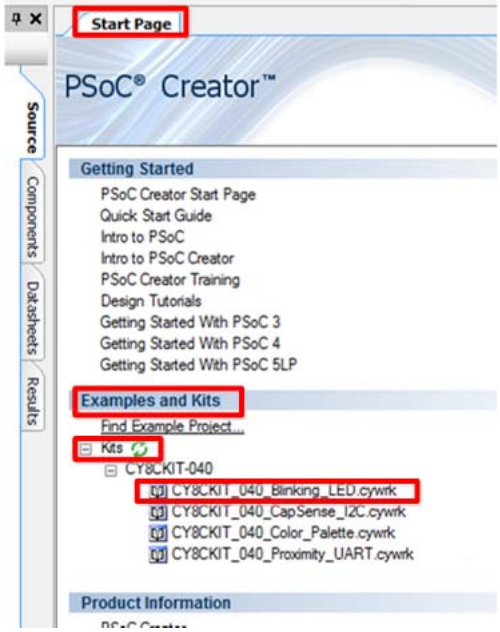
更多有关代码示例的信息，请访问 PSoC 3、PSoC 4 和 PSoC 5 代码示例网页。该网页列出了各个应用笔记、套件和 PSoC Creator 里面的所有 PSoC Creator 代码示例。

### 5.1.1 编程示例项目

可参考本部分，以在电路板上将任何示例项目编程到 PSoC 4。套件附带的示例项目的说明来自于第 54 页上的项目：使 LED 闪烁发光 。请按照下面各步骤来打开和编程示例项目：

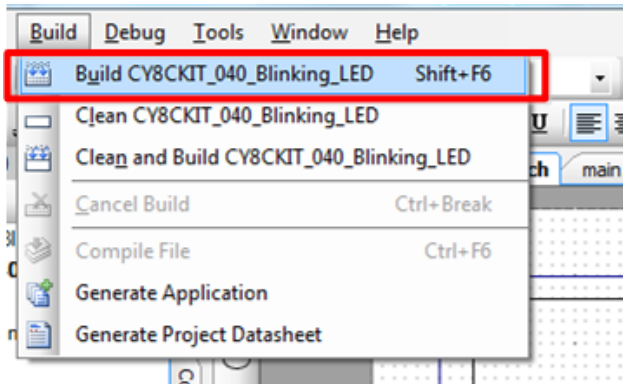
1. 从 “Windows Start” 菜单启动 PSoC Creator （**Start > All Programs > Cypress > PSoC Creator< 版本号 > > PSoC Creator< 版本号 >**）。
2. 通过点击 **Examples and Kits > Kits > CY8CKIT-040** 下面的 **< *项目名* >.cywrk** 来打开示例项目，如图 5-1 所示。在这里，我们以项目 *CY8CKIT_040_Blinking_LED.cywrk* 为例。

图 5-1. 从 PSoC Creator 打开代码示例



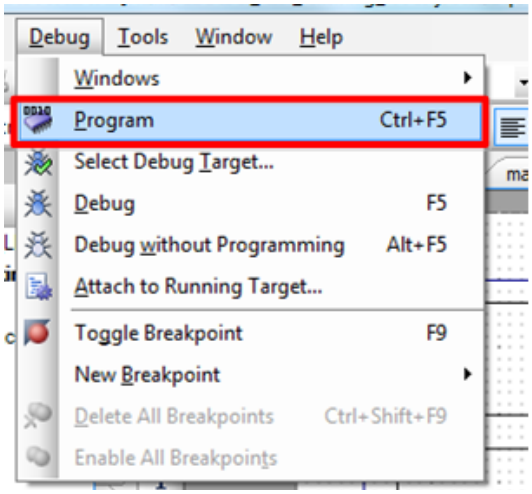

3. 通过选择 **Build > Build CY8CKIT_040_Blinking_LED.cywrk** 可以构建代码示例，这样会生成十六进制文件，如图 5-2 所示。

图 5-2. 在 PSoC Creator 中创建项目



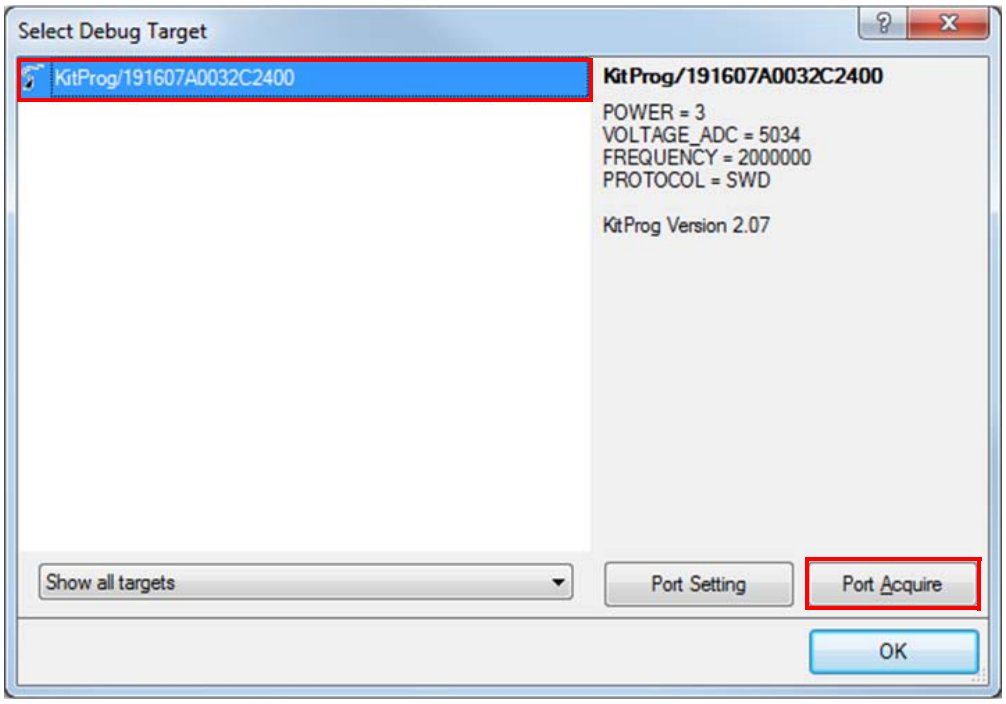

4. 使用连接至端口 J10 的 USB 线将电路板连接至电脑以进行编程，如第 21 页上的套件 USB 接口 所示。该电路板被识别为 KitProg。
5. 从 PSoC Creator，依次选择 **Debug > Program**，如图 5-3 所示。

图 5-3. 使用 PSoC Creator 对器件进行编程



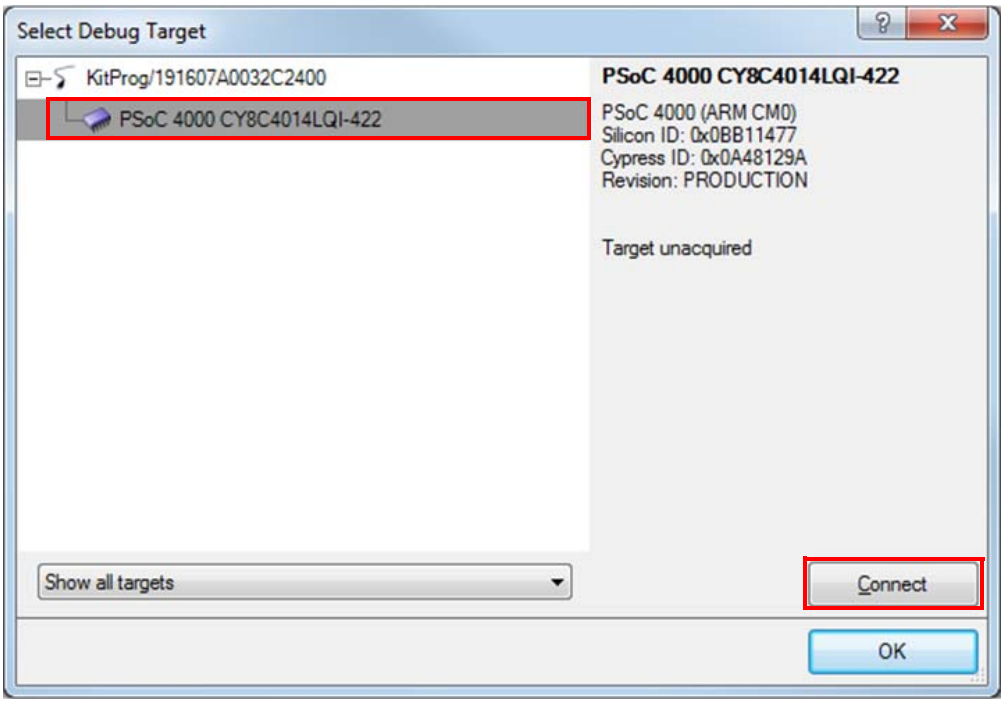
6. 如果尚未获得器件，则 PSoC Creator 将打开 **Select Debug Target** （选择调试目标）窗口。选择 **KitProg/<ID>**，并点击 **Port Acquire** 按键，如图 5-4 所示。

图 5-4. 从 PSoC Creator 中获得器件



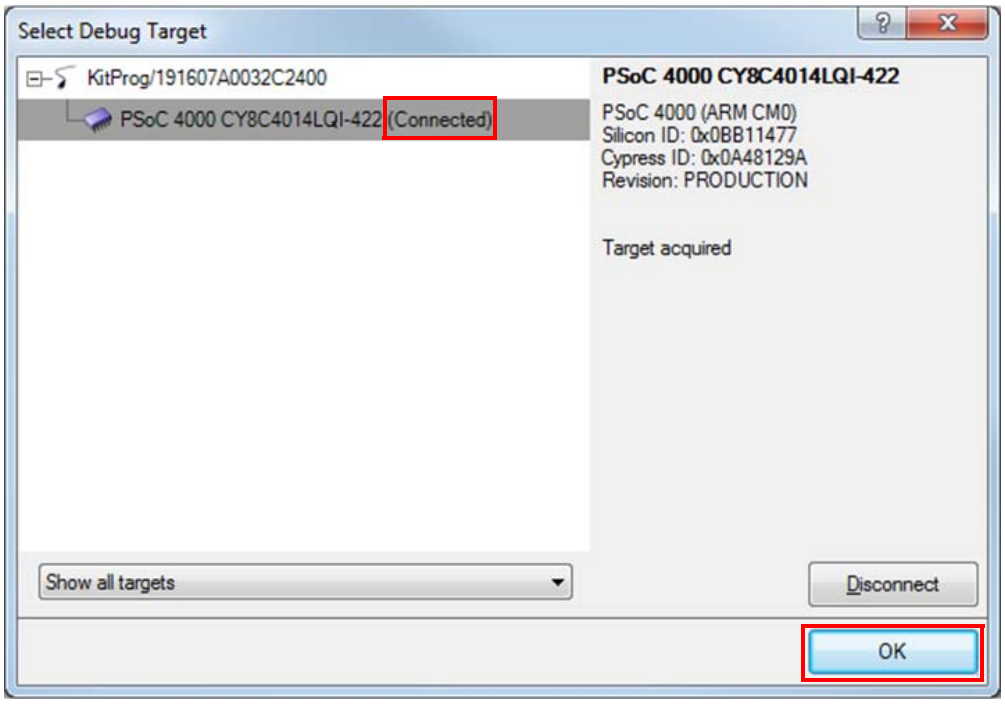

7. 获取器件后，在 KitProg 下的树形结构中将显示该器件 （请参见图 5-5）。点击 **Connect** 按键。

图 5-5. 从 PSoC Creator 中连接器件



8. 点击 **OK** 以退出窗口，并开始进行编程，如图 5-6 所示。

图 5-6. 使用 PSoC Creator 对器件进行编程




**备注：**

- 默认情况下，在各示例项目中的其中一个项目（第 57 页上的项目：CapSense 接近感应和 UART ）禁用了调试端口，因为它使用 P3[0] （SWDIO）引脚作为软件 TX 输出。如果需要进行调试，请将 *cydwr* 文件中的 **Debug Select** 设置项修改为 SWD，如图 5-8 所示。要想禁用项目中的软件 TX，请注释掉 *main.h* 文件中的 TX_ENABLE 宏。第 84 页上的项目：调色板 （Color Palette） 示例项目中包含一个软件 TX，但 TX 端口被默认禁用。要想使能 TX 端口，请在 *cydwr* 文件中将 **Debug Select** 设置项更改为 **GPIO**，并取消 *main.h* 文件中的 TX_ENABLE 宏的注释。如果同时需要 TX 和 SWD 调试，请按照以下步骤进行操作：
  a. 通过对项目的 main.h 文件所提供的 TX_PORT/TX_PIN 宏进行修改，可以将 TX 引脚路由到任何其他有效的引脚。
  b. 拆下电路板上的电阻 R57 （图 5-7）。
  c. 将第一步所选定的 TX 引脚路由到引脚 J8_9 （PSoC 5LP 的 P12[6]/RX 线连接至 J8）。
- 当在项目中使用 SmartSense 自动调试时，第一次插入 USB 线缆后（如果正在安装套件驱动器，则等到驱动器安装完成后），需要复位该器件。这是因为在上电过程中 SmartSense 会调试传感器，并且由于在插入 USB 过程中发生手触摸和电源波动，这些原因都将影响调整算法；它可能会使触摸传感器数据卡住或变得不灵敏。
- 默认情况下，第一次打开示例项目时，会在 *main.c* 或 *main.h* 文件中的‘#include <project.h>’行处有内联错误。该错误是暂时的，在构建项目时会被去除。因为 *project.h* 文件仅在构建项目时才会生成 project.h 文件，因此在构建项目前会显示这样的错误。

图 5-7. 电路板上的 R57 位置

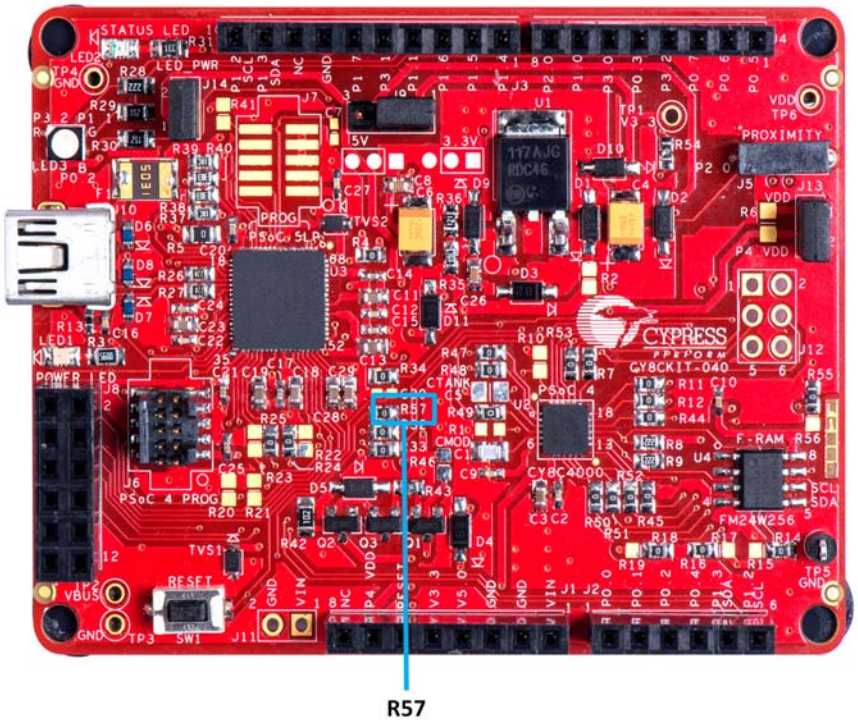
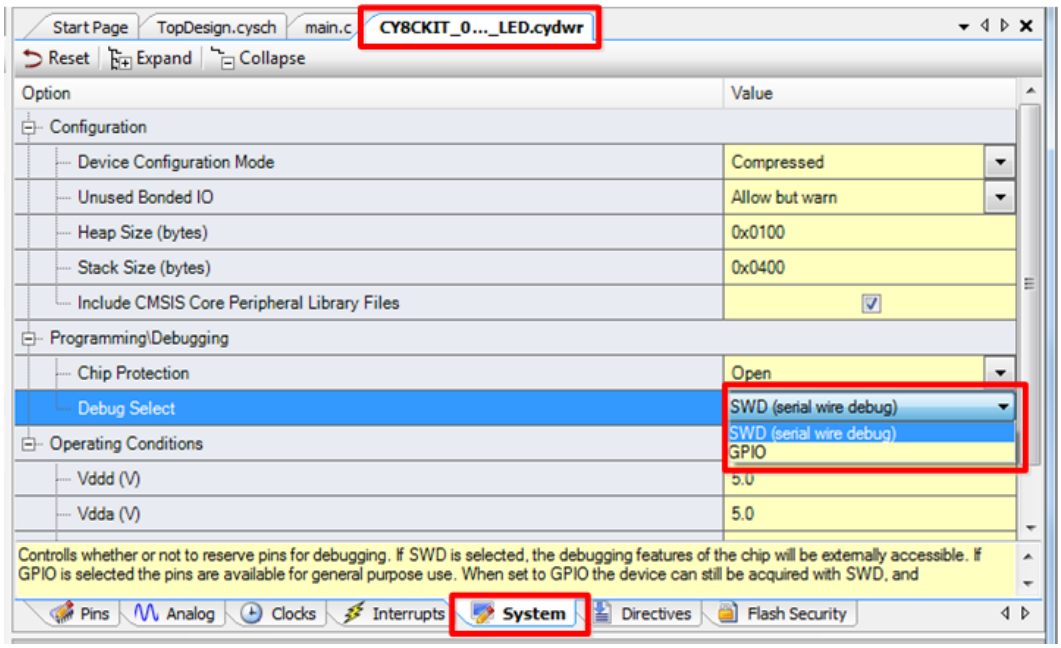



图 5-8. 选择调试端口引脚功能

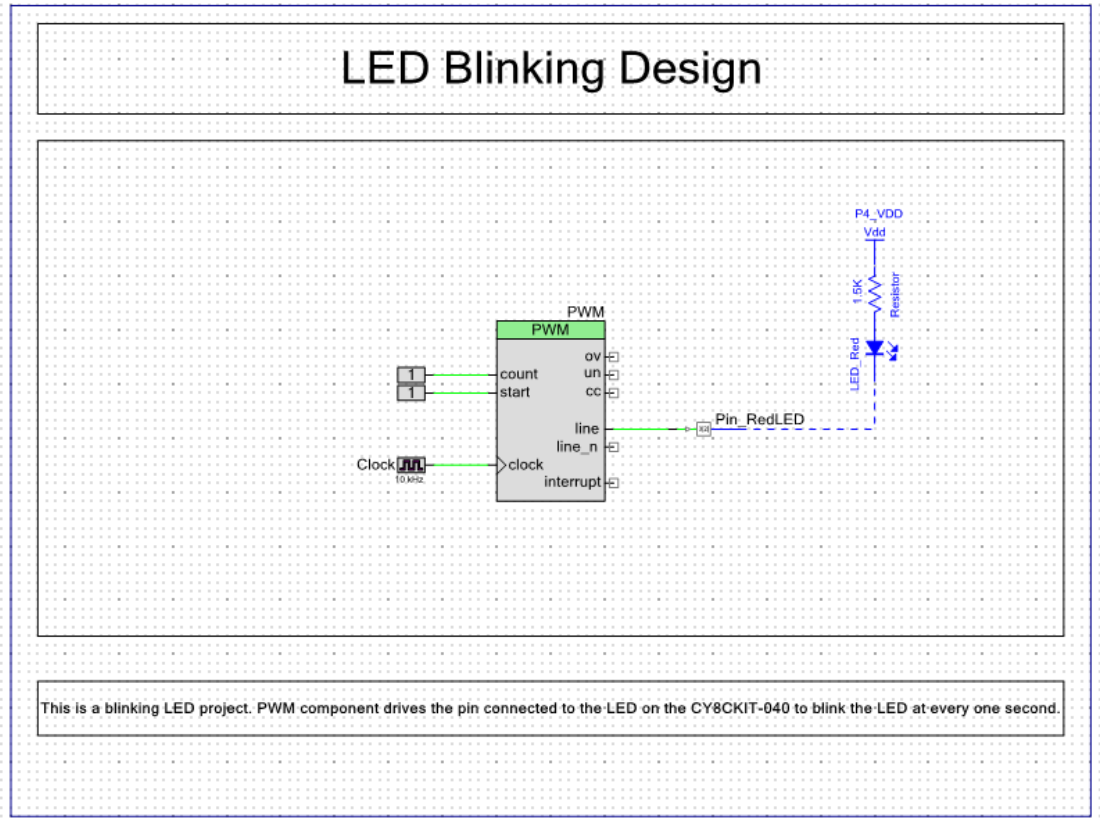


## 5.2 项目：使 LED 闪烁发光

### 5.2.1 项目概述

*CY8CKIT_040_Blinking_LED.cyprj* 示例使用了一个 PWM 模块来使 RGB LED 中的红色 LED 发亮，如图 5-9 所示。将 PWM 输出连接至 RGB LED 的引脚 P3_2 （红色）。 PWM 模块被配置为频率为 1 Hz 的数字时钟信号发生器。通过修改 PWM 的比较值和周期值，可以改变闪烁频率。

图 5-9. 闪烁 LED 项目的 PSoC Creator 原理图设计



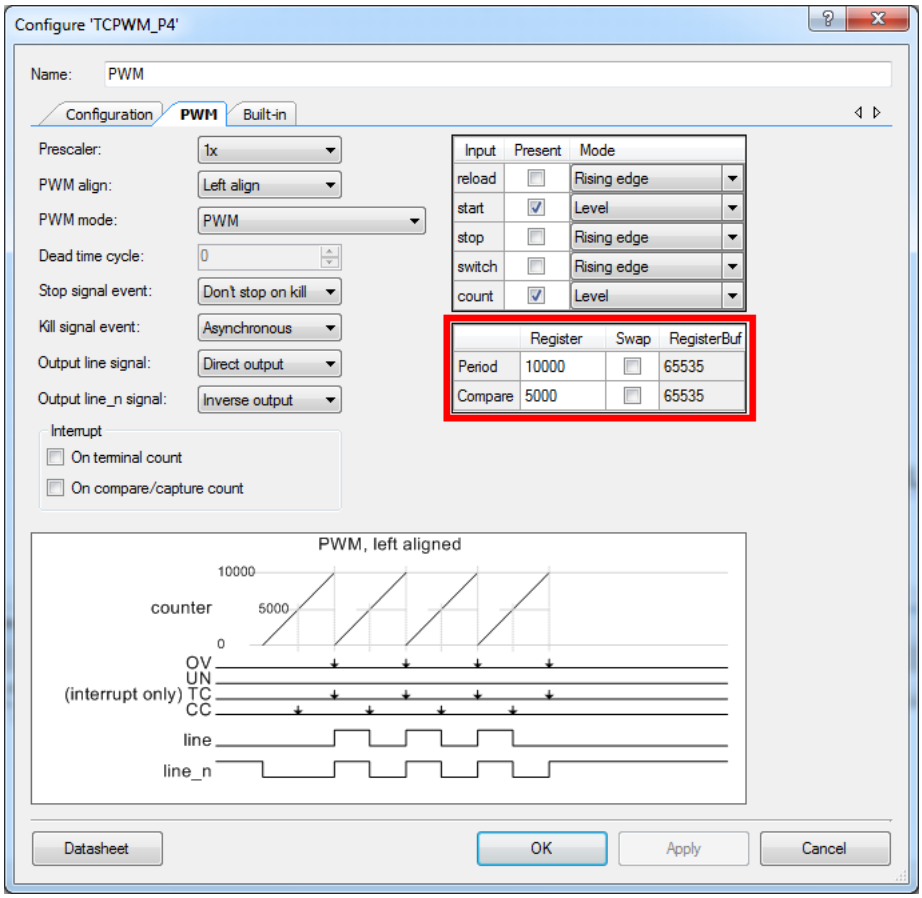
## 5.2.2 项目说明

### *5.2.2.1 PSoC Creator 组件配置*

**PWM （TCPWM 模式）**

TCPWM 组件被配置为 PWM ；该 PWM 参数配置如图 5-10 中所示。

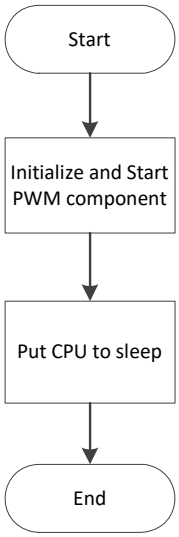
图 5-10. TCPWM 组件参数



#### 5.2.2.2 *固件的详细信息*

图 5-11 显示的是 main.c 中执行代码的流程图。

图 5-11. 闪烁 LED 项目的流程图



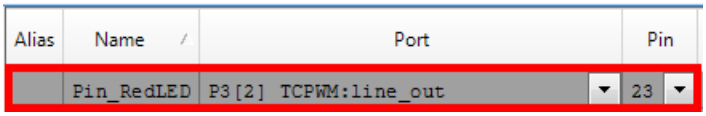

#### 5.2.2.3 *硬件连接*

由于所有连接均为电路板上的固定连线，所以该项目不要求进行任何额外的硬件连接。在 **Workspace Explorer**（工作区浏览器）中的 **Source** 垂直选项卡下，打开 *CY8CKIT_040_Blinking_LED.cydwr*，并选择合适的引脚，如图 5-12 所示。

表 5-1. 引脚连接

| 引脚名称 | 端口名称 |
|---|---|
| PWM | P3_2（红色） |

图 5-12. 为闪烁 LED 项目选择引脚




### 5.2.3 验证输出

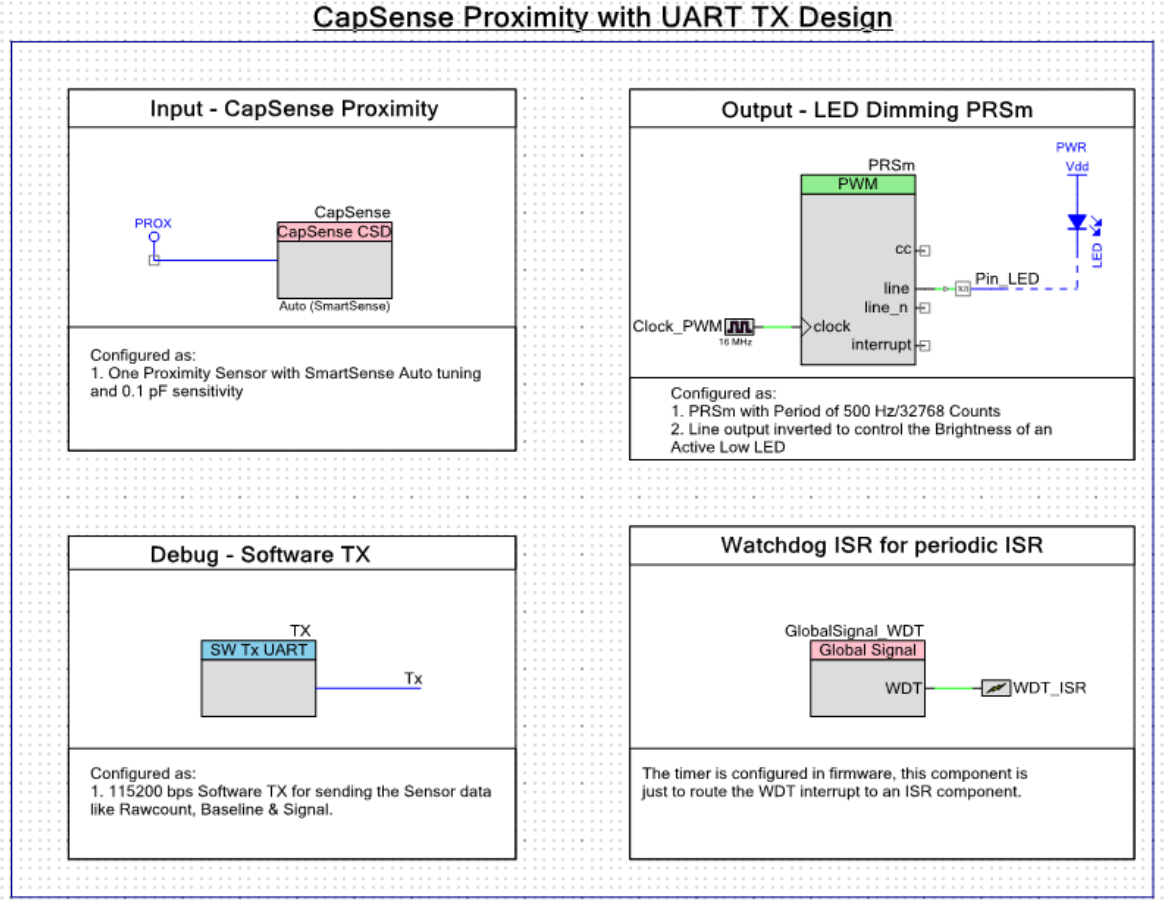
编译并烧写示例目标代码到器件中。观察闪烁 LED 的频率和占空比。在 PWM 组件内更改周期值和比较值，如图 5-10 所示。重新编译并重新编程器件，以改变闪烁频率。

## 5.3 项目：CapSense 接近感应和 UART

### 5.3.1 项目概述

通过 *CY8CKIT_040_Proximity_UART.cyprj* 项目，电容式接近感应传感器会控制 LED 的亮度。项目将传感器配置为具有 SmartSense 自动调试功能的 CapSense 接近感应 widget。第 61 页上的固件的详细信息 介绍了固件流程，并对固件模块进行了详细说明。

图 5-13. PWM 项目的 PSoC Creator 原理图设计



## 5.3.2 项目说明

### *5.3.2.1 PSoC Creator 组件配置*

#### CapSense

在设计中使用一个接近感应传感器时，在 SmartSense 自动调试模式下使用表 5-2 所示的各个参数对 CapSense 组件进行配置。

表 5-2. CapSense 组件参数

| 参数 | 选项卡显示 | 数值 | 说明 |
|---|---|---|---|
| Tuning method（调试方法） | General（通用） | 自动（SmartSense） | 针对不同的系统环境对灵敏度进行自动调整。 |
| Threshold mode（阈值模式） | | 自动 | 实时计算阈值以保证信噪比达到 5:1 |
| Raw data noise filter（原始数据噪声滤波器） | | 一阶 IIR 1/4 滤波 | 过滤掉噪声或原始计数中不必要的毛刺。可以根据需求进行调整该设置。 |
| ProximitySensor0 | Widgets Config（Widget 配置） | - | 要添加接近感应传感器，请点击 **Proximity Sensors**，然后点击 **Add Proximity Sensor**。在该选项卡中，debounce（去抖动）是唯一能修改的参数。根据系统要求可以设置或调整该参数。 |
| Analog Switch Drive Source（模拟开关驱动源） | Advanced（高级） | PRS-12b | 降低 EMI 辐射，并增强 EMC 的抗干扰能力。 |
| Sensor auto-reset（传感器自动复位） | | 禁用 | 在设计中不需要传感器自动复位。如果应用需要，则添加该特性。 |
| Low Baseline Reset（低基线复位） | | 5 | 取决于系统的数值。根据用户的需求可以配置该项。 |
| Inactive Sensor Connection（无效传感器连接） | | 接地 | 接近感应回路不扫描时不会进行充电 |
| Shield （屏蔽） | | 禁用 | 在设计中不使用该参数。 |
| Guard Sensor（保护传感器） | | 禁用 | 在设计中不使用该参数。 |
| Cmod precharge（Cmod 预充电） | | 通过 Vref 缓冲进行预充电 | 因为只有一个传感器，所以进行预充电时，Vref 足够用。GPIO 快速预充电时，Cmod 电压不会降得太低。 |
| Sensitivity（灵敏度） | Scan Order（扫描顺序） | 1 | 使用 SmartSense 时，可以获得尽可能高的灵敏度。通过该参数控制扫描时间，因此对于灵敏度较低的设置，扫描率反而会更高。根据响应速率和所需要的近距范围来调整该参数。 |
| Enable tuner helper（使能调谐器助手） | Tuner helper（调谐器助手） | 未选中 | 未使用调谐器工具。 |

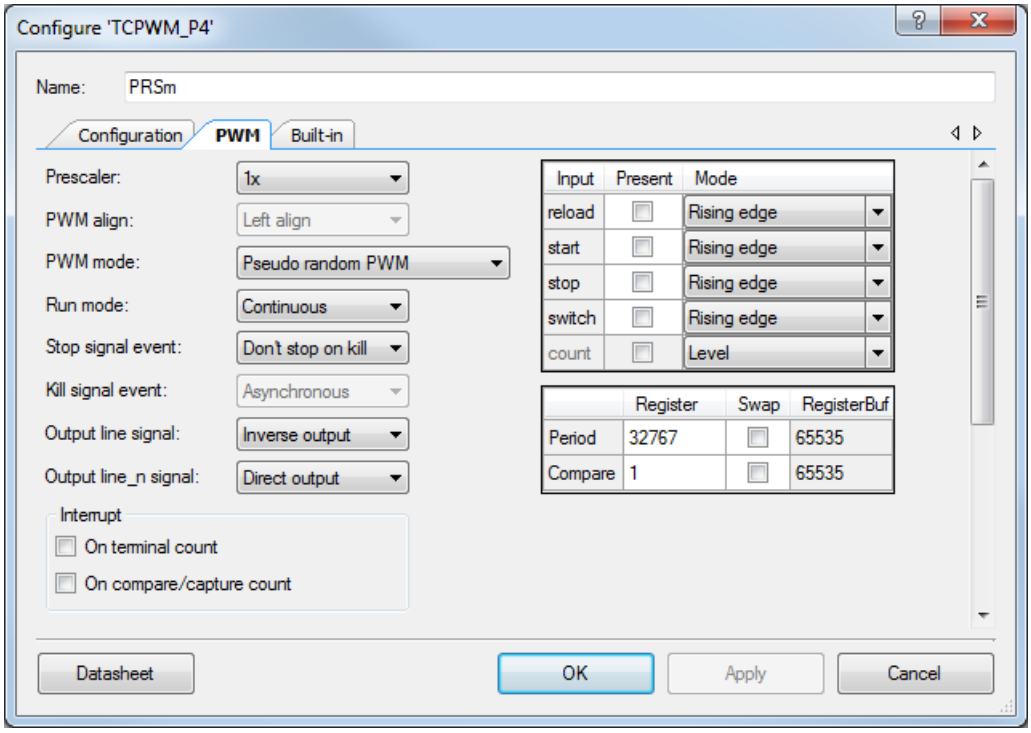
#### PRSm （TCPWM 模式）

TCPWM 组件用于控制 LED 亮度。 CapSense 接近感应传感器输出的信号用于降低 LED 的亮度。图 5-14 显示了 TCPWM 组件的参数。在伪随机序列调制器 （PRSm）模式中将 TCPWM 模块配置为 PWM。其分辨率为 15 位（由 TCPWM 模块架构设置成固定值）。该 15 位分辨率的 PRSm 和频率为 16 MHz 的输入时钟可生成一个频率为 500 Hz （PRS 重复周期）的周期。输出反相，以驱动低电平

有效 LED。在组件中将周期值设置为 32767，以为 15 位 PRSm 生成合适的周期宏。虽然 PRSm 的输出端有可变的频率，并且最大频率为 8 MHz （16 MHZ/2），但在该情况下， PRSm 的重复率仍被视为周期。

**备注：**最小的比较值应为 ‘1’ ；如果比较值为 ‘0’ ，则 LED 会亮起。

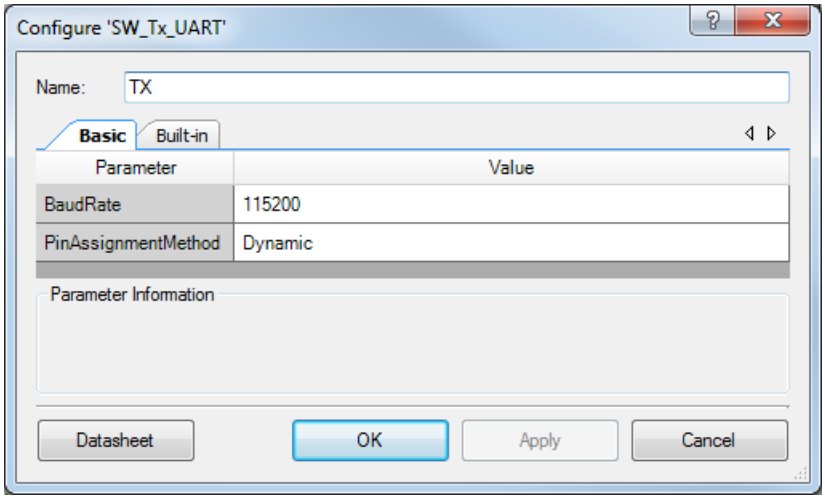
图 5-14. TCPWM 组件配置 — ‘PWM’ 选项卡



### TX （软件 TX UART）

软件传输 TX 用于发送与接近感应传感器相关的数据，以进行调试。图 5-15 显示了组件的配置情况。通过 *main.h* 文件中定义的 TX_PORT/TX_PIN 宏，可在固件中选用 TX 引脚。通过使用 RS232 连接器（与电脑间电平转换）、或 CY8CKIT-040 PSoC 5 LP UART 桥接中提供的 USB-UART 桥接、或配置为 UART 桥接的 CY3240 桥接 （详见 AN2397），可以将 SW TX 数据发送到电脑。

图 5-15. 软件 UART TX 组件参数




### Pin_LED （数字输出引脚）

数字输出引脚用于驱动 LED 的 PWM 输出。这是一个设置为标准强驱动的输出引脚。

### Clock_PWM (Cy_Clock)

Clock_PWM 提供一个用于驱动 PWM 模块的时钟。将时钟配置为尽可能大的值或允许值（16 MHz），以便使 PRSm 的重复率尽可能高，这样可降低 LED 闪烁。

### GlobalSignal_WDT （ISR 的全局信号参考）

用于将 WDT ISR 路由到 ISR 组件的组件。然后，在固件中配置该 ISR，用以在 Sleep_Scan 模式中通过使用 WDT 生成周期性的唤醒信号。

### 5.3.2.2 *固件的详细信息*

#### 固件结构

固件开发以模块化方式进行，以将该固件所具有的不同功能作为单独的功能介绍，这样便于理解。头文件提供了一系列的宏，用于根据用户的要求配置项目关键内容。头文件的注释内容提供了宏的详细信息。

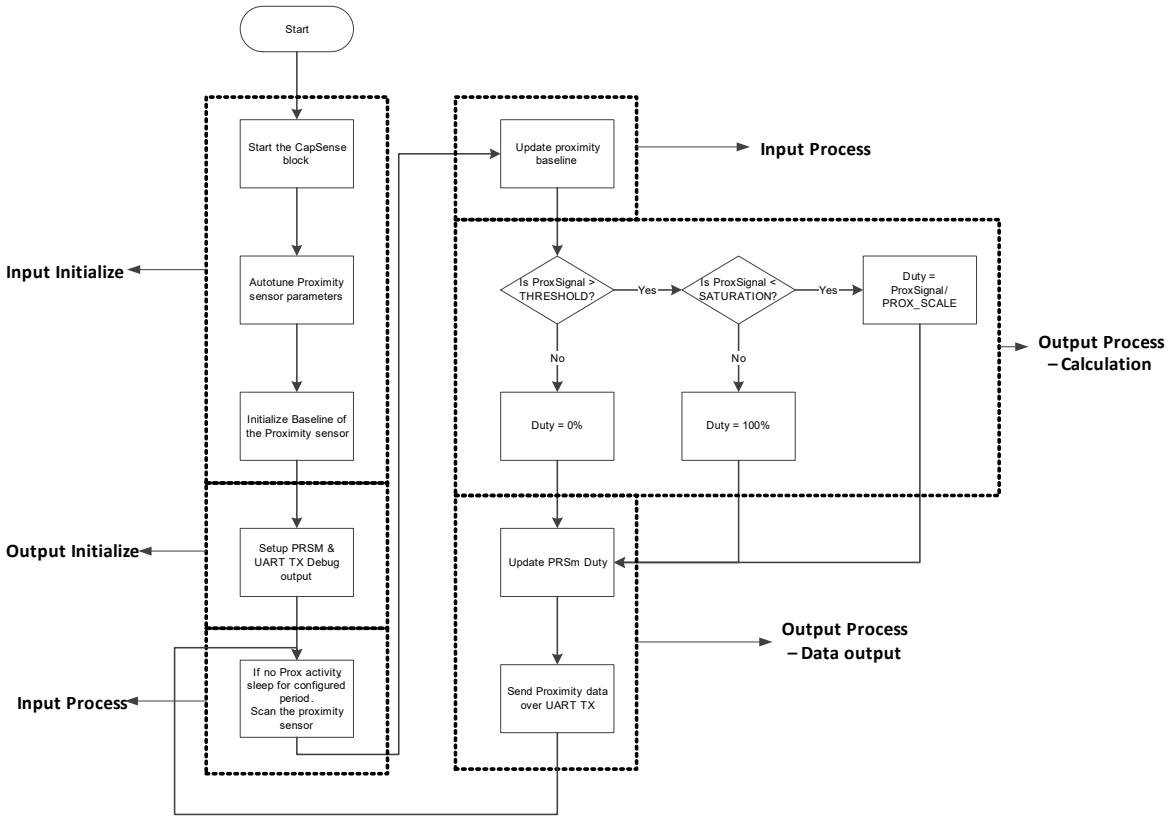
表 5-3. CapSense 接近感应项目中的源文件和头文件

| 文件名称 | 用途 |
|---|---|
| main.c | 该文件包含了固件中所使用的所有函数定义。 |
| main.h | 该文件包含了固件中所使用的所有宏。可以在每个宏的注释中找到该宏及其用法的详细信息。 |

#### 固件流程图

图 5-16 显示的是在 main.c 中执行代码的流程图。

图 5-16. CapSense 接近感应和 UART 项目的流程图



固件执行下面的操作：

- CapSense/ 输入的初始化代码部分使用 SmartSense 来调整 CapSense 系统参数。
- 输出的初始化代码部分配置 PWM 和软件 UART TX 输出。
- 无限循环内代码分为两个阶段：输入过程和输出过程。
  - 输入处理会扫描接近感应传感器，并处理传感器信号，如：应用滤波器、计算基线和信号。
  - 输出处理又可分为两个小阶段：数据计算和数据输出等子阶段。

    i. 数据计算子阶段对接近信号与手接近定义的最小阈值和最大阈值进行比较。然后，会根据传感器相对于阈值的信号值计算 LED 的亮度。最小阈值会使 LED 亮度最低，并且最大阈值会使 LED 亮度最高。

    ii. 数据输出子阶段将 PWM 比较值改为计算得出的亮度值。通过 UART TX 线发送系统数据（如：传感器原始数据、基线、信号和计算得到的 LED 亮度）。

器件对接近感应传感器进行监控，在不发生任何操作时，即手未进入接近感应的范围，则器件会进入睡眠扫描模式。进入睡眠扫描模式前，器件需要经过 5 秒的时间来检查传感器上是否发生操作，在项目中，该时长是可配置的（*main.h* 中的 ENTER_SLEEP_COUNTS 宏）。在睡眠扫描模式下，每经过 100 毫秒，器件将被唤醒一次，然后检查接近感应传感器是否发生任何操作。通过修改 *main.h* 文件中的 WATCHDOG_TIMER_COUNT 宏，可配置唤醒率。

**备注：**在 *main.h* 中定义了一些宏，用于根据用户的要求配置项目。每个宏定义上面的注释内容都提供该宏的详细信息。

### 常用的 CapSense 组件函数 / 变量

表 5-4、表 5-5 和表 5-6 分别提供了某些常用的变量、宏定义和组件 API 的细节。此外，也可以在 CapSense 组件数据手册中查看这些信息。

**备注：**对 API/ 变量 / 宏的所有引用中，都假设 CapSense 组件实例的名称为 “CapSense”。

表 5-4 为 CapSense 组件的某些关键变量 / 数组提供快速参考。

表 5-4. CapSense 组件关键变量

| 变量 / 数组名称 | 说明 | 用法 |
|---|---|---|
| uint16 CapSense_SensorRaw[] | 此数组包含了每个传感器的原始计数。数组大小等于传感器总数 (CapSense_TOTAL_SENSOR_COUNT)。 CapSense_SensorRaw [ ] 数据通过以下函数更新：<br>• CapSense_ScanSensor()<br>• CapSense_ScanEnabledWidgets()<br>• CapSense_InitializeSensorBaseline()<br>• CapSense_InitializeAllBaselines()<br>• CapSense_UpdateEnabledBaselines() | 通过使用 ‘.c’ 或 ‘.h’ 文件中的 ‘extern uint16 CapSense_SensorRaw[];’ 导入变量，可以在任何文件中访问该变量。<br>建议不要手动改变数组。 |
| uint16 CapSense_SensorBaseline[] | 此数组存储了每个传感器中的基线数据。针对每个传感器独立计算得出的历史计数值被称为这个传感器的基线。数组大小等于传感器总数。通过以下各函数更新 CapSense_SensorBaseline[ ] 数组：<br>• CapSense_InitializeSensorBaseline()<br>• CapSense_InitializeAllBaselines()<br>• CapSense_UpdateSensorBaseline()<br>• CapSense_UpdateEnabledBaselines()。 | 通过使用 ‘.c’ 或 ‘.h’ 文件中的 ‘extern uint16 CapSense_SensorBaseline[];’ 导入变量，可以在任何文件中访问该变量。<br>建议不要手动改变数组。 |
| uint8 CapSense_SensorSignal[] | 此数组存储通过从每个传感器的当前计数中减去以前的基准线计算而来的传感器信号数据。只有在数值超过传感器的噪声阈值时，与传感器相应的每个数组元素才会有相应的差值。否则，它将为 0。数组大小等于传感器总数。 CapSense_SensorSignal[ ] 数组通过以下各函数更新：<br>• CapSense_InitializeSensorBaseline()<br>• CapSense_InitializeAllBaselines()<br>• CapSense_UpdateSensorBaseline()<br>• CapSense_UpdateEnabledBaselines()。 | 通过使用 ‘.c’ 或 ‘.h’ 文件中的 ‘extern uint8 CapSense_SensorSignal[];’ 导入变量，可以在任何文件中访问该变量。<br>建议不要手动改变数组。 |

表 5-5 为 CapSense 组件的某些关键宏定义提供快速参考。

表 5-5. CapSense 组件宏

| 宏格式 | 样本 | 说明 |
|---|---|---|
| CapSense_TOTAL_SENSOR_COUNT | – | 定义CapSense组件内的传感器总数。 |
| CapSense_SENSOR_"WIDGET_NAME"_<*ELEMENT+ELEMENT_NUMBER*>__"WIDGET_TYPE" | • CapSense_SENSOR_TP1_ROW0__TP<br>• CapSense_SENSOR_LS0_E0__LS<br>• CapSense_Sensor_BTN1__BTN | 常量表示 CapSense 模块中某个传感器的编号。<br>WIDGET_TYPE:<br>BTN — 按键<br>LS — 线性滑条<br>RS — 辐射滑条<br>TP — 触摸板<br>PROX — 接近感应传感器<br>MB — 矩阵按键<br>GEN — 通用传感器<br>GRD — 保护传感器 |
| CapSense_"WIDGET_NAME"__"WIDGET_TYPE" | • CapSense_TP1__TP<br>• CapSense_LS0__LS<br>• CapSense_BTN1__BTN | 常量表示 CapSense 模块中一个 widget 的编号。 |

表 5-6 为 CapSense 组件中的一些关键 API 及其用途提供快速参考。

表 5-6. CapSense 组件 API

| API | 说明 / 用法 |
|---|---|
| void CapSense_EnableWidget(uint32 widget) | 通过 API，可以使选中的 Widget 传感器成为扫描过程的一部分。<br>该组件默认禁用了接近感应 widget；用户需要以该接近感应 widget 编号为参数调用该 API，以在扫描过程中使能它们。 |
| void CapSense_Start(void) | 如果使用了 SmartSense 或自动校准，则该 API 将使能 CapSense 模块，并对传感器进行校准。<br>使用 CapSense 模块工作前，应调用该函数。 |
| void CapSense_InitializeAllBaselines(void) | 该 API 将 CapSense_sensorBaseline[] 数组初始化为扫描所有传感器得到的值。<br>启动CapSense模块后，开始扫描过程前，应调用该函数，以便检测触摸。 |
| void CapSense_InitializeSensorBaseline(uint32 sensor) | 该 API 将 CapSense_sensorBaseline[sensor] 数组元素初始化为扫描所选传感器获得的值。<br>可以使用它来单独初始化每个基线。 |
| void CapSense_ScanEnabledWidgets(void) | API 开始扫描已使能的 Widget 的传感器。 ISR 持续对传感器进行扫描，直至扫描完所有已使能的 Widget 为止。使用 ISR 可确保该函数无阻塞。完成扫描每个传感器后， ISR 会将已测量的传感器原始计数复制到 CapSense_SensorRaw[] 数组中。<br>如果设计中有多个 widget，则优先选择这种扫描方法。 |
| void CapSense_ScanWidget(uint32 widget) | API为所选的widget进行CapSense模块设置，并开始对widget进行扫描。<br>如果只需要扫描单独的 widget，那么可以使用该函数。 |

表 5-6. CapSense 组件 API

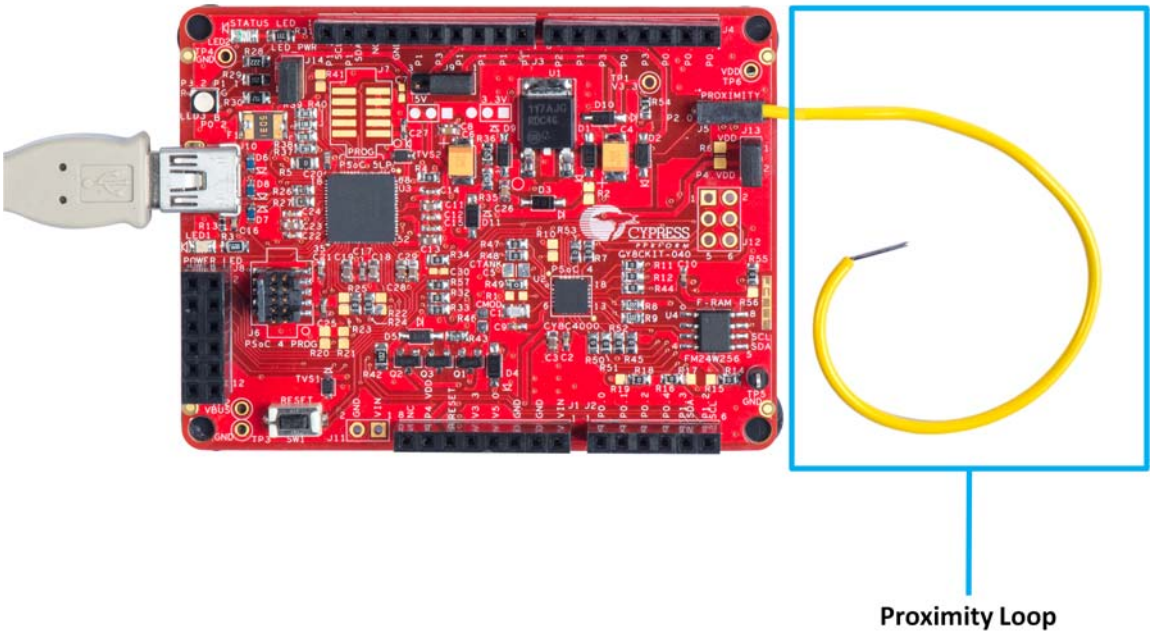
| API | 说明 / 用法 |
|---|---|
| void CapSense_ScanSensor(uint32 sensor) | API 扫描所选的传感器。完成扫描后，ISR 会将已测量的传感器原始数据复制到 CapSense_SensorRaw[] 数组元素中。<br>可以用它来执行单一的传感器扫描。 |
| uint32 CapSense_IsBusy(void) | API 返回传感器扫描状态。<br>调用任何扫描 API 后，可使用该 API 检查扫描是否完成。<br>返回<br>如果正在进行扫描，将会返回数值 ‘1’。<br>如果扫描完成，将会返回数值 ‘0’。 |
| void CapSense_UpdateSensor-Baseline(uint32 sensor) | API 使用组件中选中的滤波器来处理 CapSense_SensorRaw[sensor] 元素。滤波得到 CapSense_SensorRaw[sensor] 值后，使用 “k = 256” 的低通滤波器将它们更新到 CapSense_SensorBaseline[sensor] 中。<br>ScanSensor() API 完成后，在对传感器上的任意操作进行检查前，应调用该 API。 |
| void CapSense_UpdateEnabled-Baselines(void) | API 将所选的滤波器应用于 CapSense_SensorRaw[] 数组，并更新使能 widget 内所有传感器的 CapSense_SensorBaseline[] 数组。<br>ScanEnabledWidgets() API 完成后，在对任何传感器上的任意操作进行检查前，应调用该 API。 |
| uint32 CapSense_CheckIsWidget-Active(uint32 widget) | API 将选定的传感器 CapSense_Signal[ ] 数组值与其手指阈值进行比较。通过迟滞和去抖动性能判断选定的widget中的传感器是否处于活动状态。<br>调用 UpdateSensorBaseline() 或 UpdateEnabledBaselines() API 后，应调用该 API，以检查 widget 中的传感器是否活动。<br>返回<br>如果 widget 内的一个或多个传感器为活动状态，则返回 ‘1’<br>如果 widget 内的所有传感器都处于非活动状态，则返回 ‘0’ |
| uint32 CapSense_CheckIsAnyWid-getActive(void) | API 对所有使能的 widget 执行与 CapSense_CheckIsWidgetActive() 相同的任务。<br>调用 UpdateEnabledBaselines() API 后或更新所有使能传感器 /widget 的基线后，应调用该 API，以进行检查传感器在所有使能的 widget 中是否为活动状态。<br>返回<br>如果任意 widget 处于有效状态，将返回数值 ‘1’<br>如有所有 widget 都处于无效状态，则返回 ‘0’ |

### 5.3.2.3 *硬件连接*

将环状的导线连接到跳线 J5 (P2_0)，如 图 5-17 所示。要使能 UART TX 与 PSoC 5LP USB-UART 桥接间的连接，请确保电路板上安装好 R57 （默认情况下，在电路板上已经安装了 R57）。该项目不需要任何其他硬件连接。其他所有连接都被硬件连接在电路板上。

**备注：**接近距离取决于导线环的直径。直径越大，则距离越大。因为大的环更易接纳噪声，所以设计环路时应该更加留意。如果套件附带的导线（长度为 4 英寸）形成直径为 1 至 2 英寸的环，快速靠近的手的接近感应范围将与环的直径大致相同。为得到更大的范围，请使用更长的导线 / 更大的环。另外，不要在对器件进行编程或供电后插入导线环，因为固件在复位过程中调校接近感应传感器。复位后，如果插入导线环，会被识别为电容的变化，这样会使 LED 始终亮起。如果已编程完器件，则将导线插入导线环后始终进行一次复位。

图 5-17. CapSense 接近感应示例 — 硬件设置




在**工作区浏览器**中打开 *CY8CKIT_040_Proximity_UART.cydwr*，然后选择适合的引脚，如 图 5-18 所示。

表 5-7. 引脚连接

| 引脚名称 | 端口名称 |
|---|---|
| 接近感应引脚 | P2_0 |
| CMOD 引脚 | P0_4 |
| LED 引脚 | P3_2 （红色） |
| UART TX 引脚 | P3_0[1] |

1. 在固件中选定

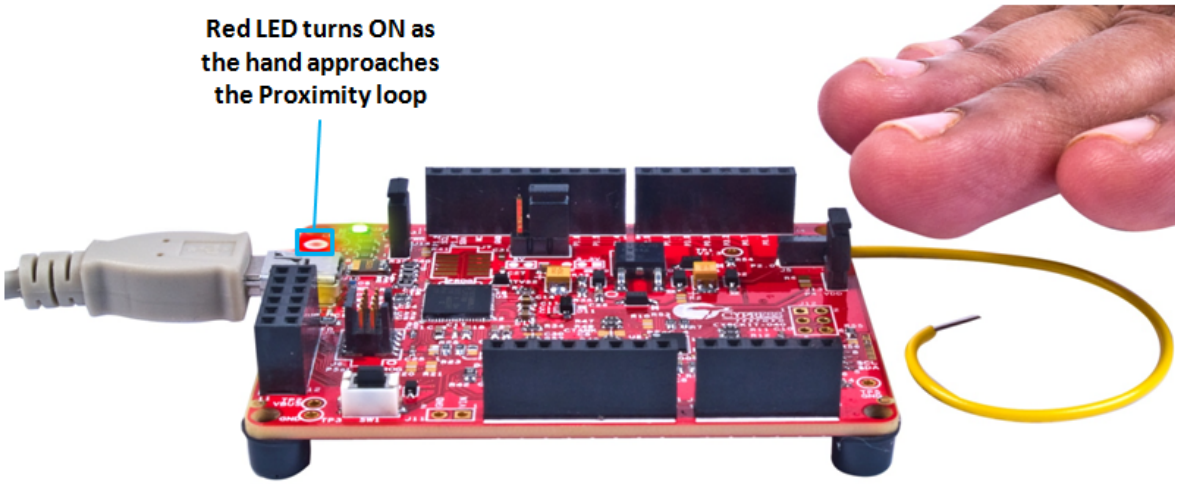
图 5-18. 为接近感应项目选择引脚

| Alias | Name | Port |
|---|---|---|
| Cmod | \CapSense:Cmod\ | P0[4] CSD:c_mod, SRSS:ext_clk, TCPWM:tr in[4], CSD:comp |
| ProximitySensor0_0_PROX | \CapSense:Sns\ | P2[0] |
| | Pin_LED | P3[2] TCPWM:line_out |

## 5.3.3 验证输出

创建和编程代码示例，并复位器件。当您把手掌移到接近导线环时，请观察红色 LED 亮度的变化。通过 BCP 可以查看 UART TX 数据，如 5.3.3.1 查看 UART 数据章节介绍。

图 5-19. CapSense 接近感应示例输出




### *5.3.3.1 查看 UART 数据*

一个 UART 数据包的大小为 13 个字节，其中包含 8 个字节的数据、 2 个字节的包头和 3 个字节的包尾。2 个字节的包头位于数据字节前面；在设计中其位置为 0x0D 和 0x0A。3 个字节的包尾位于数据字节后面。在设计中其位置包括 0x00、0xFF 和 0xFF。数据字节包括接近感应传感器原始计数（RC）、基线（BL）、信号（SIG）与计算好的 PWM 占空比（DUTY）。表 5-8 显示了 UART TX 数据包的结构。

表 5-8. UART TX 数据包结构

| **包头** | | **数据** | | | | |
|---|---|---|---|---|---|---|
| 字节 0 | 字节 1 | 字节 2 | 字节 3 | 字节 4 | 字节 5 | 字节 6 |
| 0x0D | 0x0A | RC_MSB | RC_LSB | BL_MSB | BL_LSB | SIG_MSB |
| **数据** | | **包尾** | | | | |
| 字节 7 | 字节 8 | 字节 9 | 字节 10 | 字节 11 | 字节 12 | |
| SIG_LSB | DUTY_MSB | DUTY_LSB | 0x00 | 0xFF | 0xFF | |

按照这些步骤设置 BCP，以查看数据：

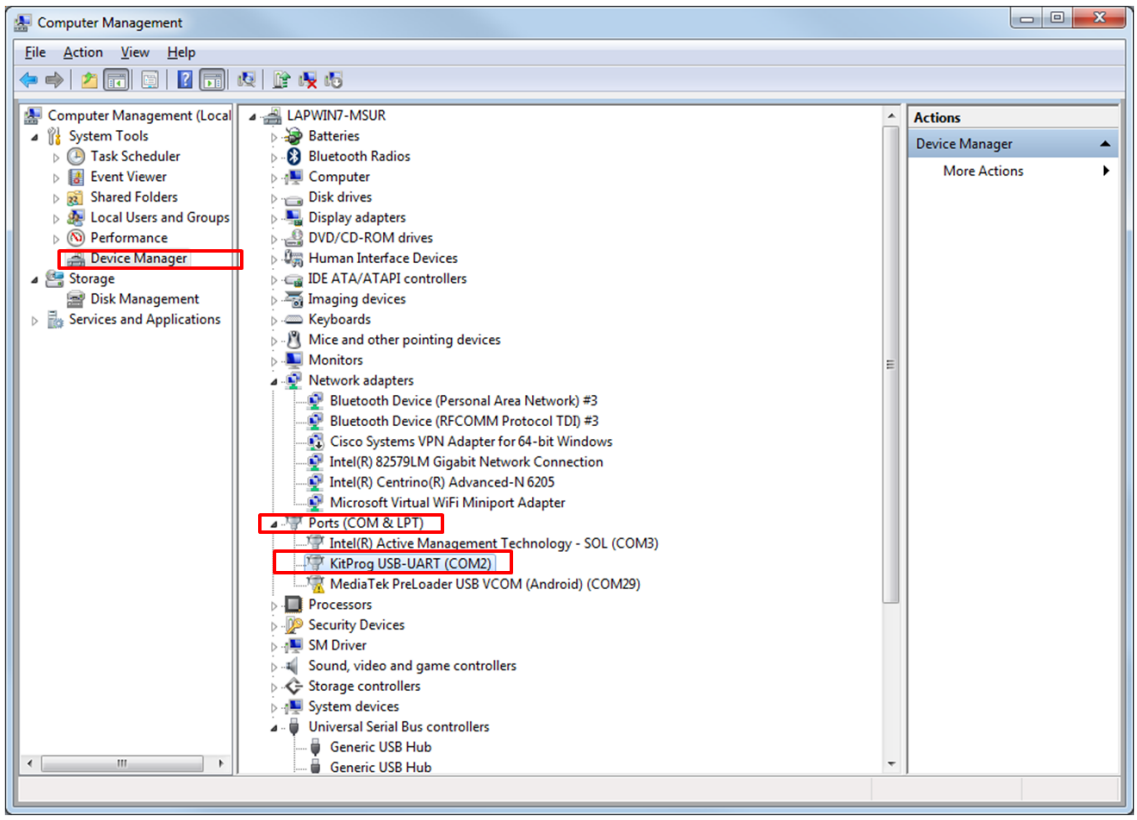
1. 依次选择 **All Programs > Cypress > Bridge Control Panel < 版本 > > Bridge Control Panel < 版本 >**，打开 BCP 软件。

2. 将器件的 TX 引脚路由到任何可连接到 PC COM 端口上的 RX。 CY3240 （请参考 AN2397）或 CY8CKIT-040 中的 KitProg 可用于该目的。

   **备注：**在 CY8CKIT-040 中，通过零欧姆的电阻将 Pin 3_0 直接路由到 PSoC 5 LP USB-UART 桥接的 RX 引脚上。Pin 3_0 可作为 TX 使用，并且只有通过禁用芯片的调试功能，Pin 3_0 才有效。如果同时需要 TX 和调试功能，则要移除 TX 和 RX 之间连接的零欧姆电阻。可将任何空闲的 GPIO 作为器件的 TX 使用，并且可以将其外部路由到 PSoC 5 LP 的 RX 引脚（请参考第 49 页上的编程示例项目 的最后步骤，了解详细信息）

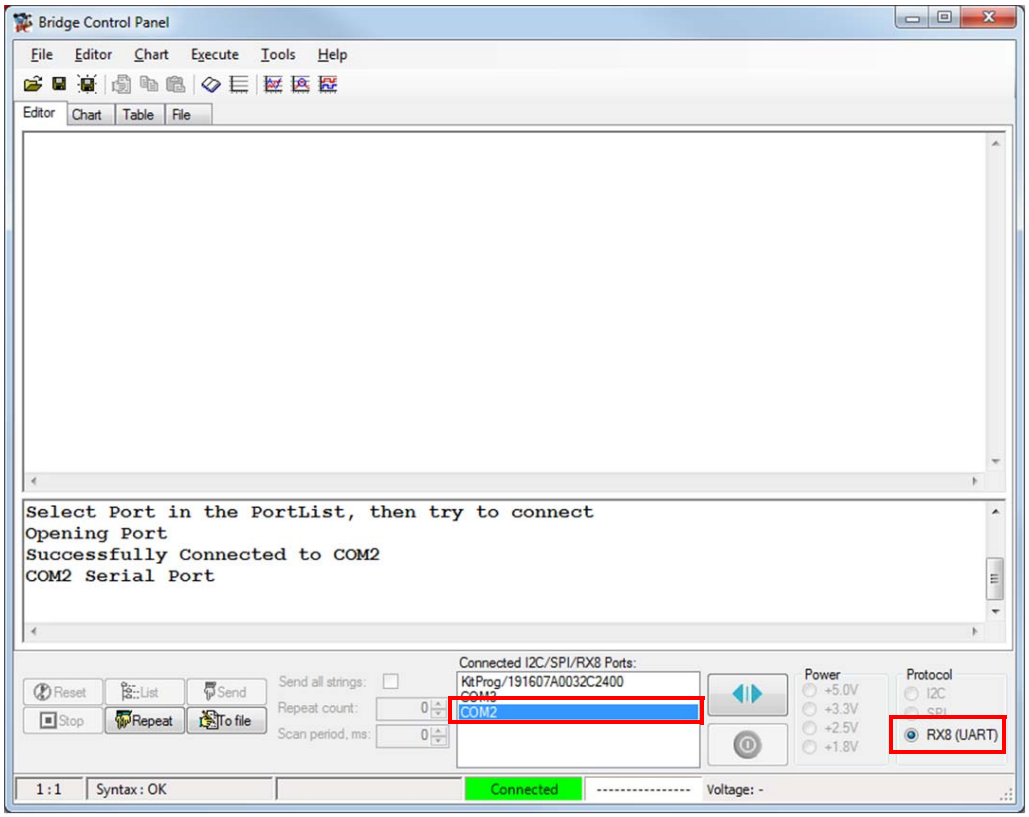
3. 在 BCP 软件中，点击已经连接到数据的 COM 端口。在该情况下，该端口为 KitProg 的 COM，如图 5-20 所示。

图 5-20. 器件管理中的 KitProg COM 端口



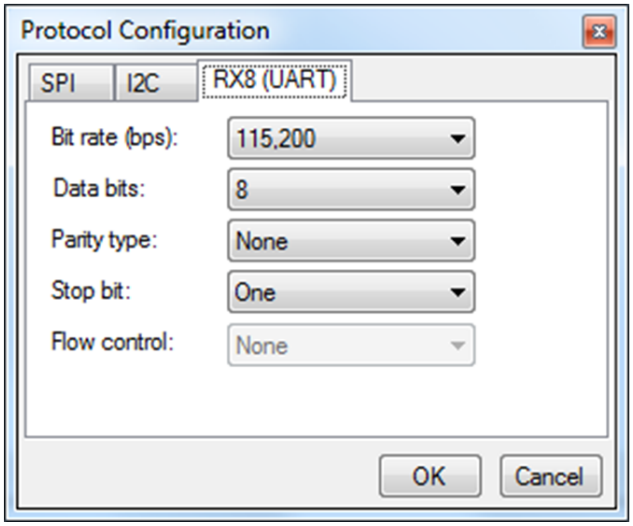
4. 选择 **COM** 端口，并选择 **RX8** 协议，如图 5-21 所示。

图 5-21. Bridge Control Panel （桥接控制面板） — 选择 COM 端口和协议



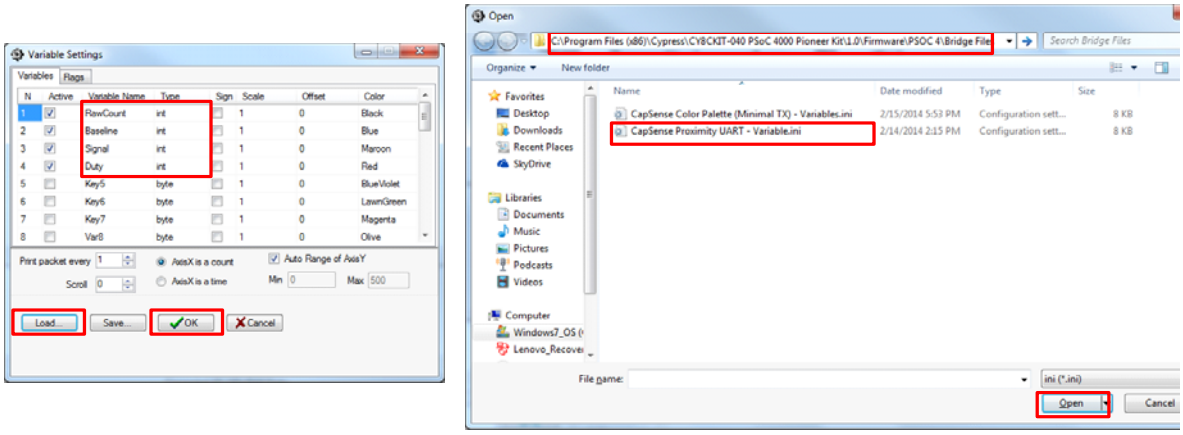

5. 依次选择**Tools > Protocol Configuration**或按下**[F7]**按键，并配置RX8协议参数，如图5-22所示。

图 5-22. RX8 协议配置



6. 依次选择 **Chart > Variable Settings**，然后设置变量名称和类型，如图 5-23 所示。或者点击 **Load** 按键，然后在显示的 **Open** 窗口中选择项目所提供的 *CapSense Proximity UART - Variable.ini* 文件 （...\Firmware\PSoC 4\Bridge Files\）。点击 **OK** 按键，以退出该窗口。

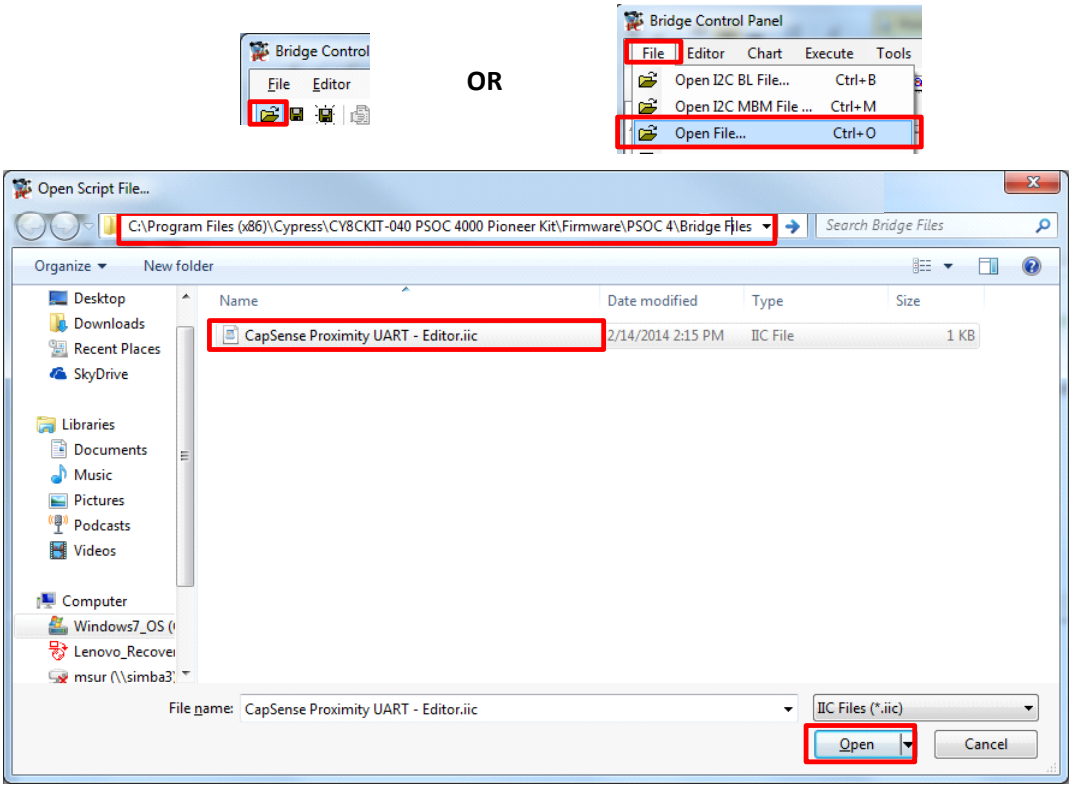
图 5-23. Bridge Control Panel （桥接控制面板）— Variable Settings （变量设置）选项卡




7. 依次选择 **File > Open File** ，然后从 ...\Firmware\PSoC 4\Bridge Files\ 文件夹中选择 *CapSense Proximity UART - Editor.iic* 文件（如图 5-24 所示）。另外，也可以进入 **Editor** ，然后键入或复制下面指令：

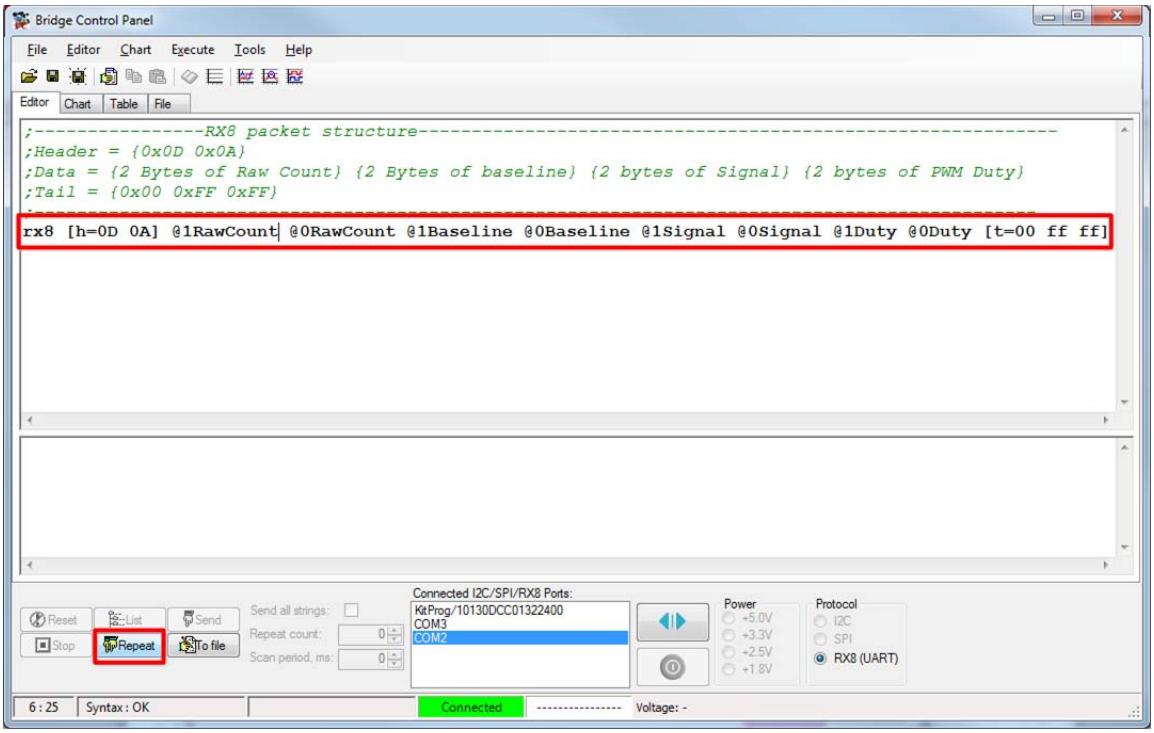
```
rx8 [h=0D 0A] @1RawCount @0RawCount @1Baseline @0Baseline @1Signal @0Signal @1Duty @0Duty [t=00 ff ff]
```

图 5-24. 在 Bridge Control Panel 中打开 “*.iic” 文件



8. 点击指令行，然后点击 **Repeat**，如图 5-25 所示，以开始接收数据包（请确保您已经给器件供电，并已经使用固件编程项目； TX 被连接至 COM 的 RX 线以及在 BCP 中选中了 COM 端口）。

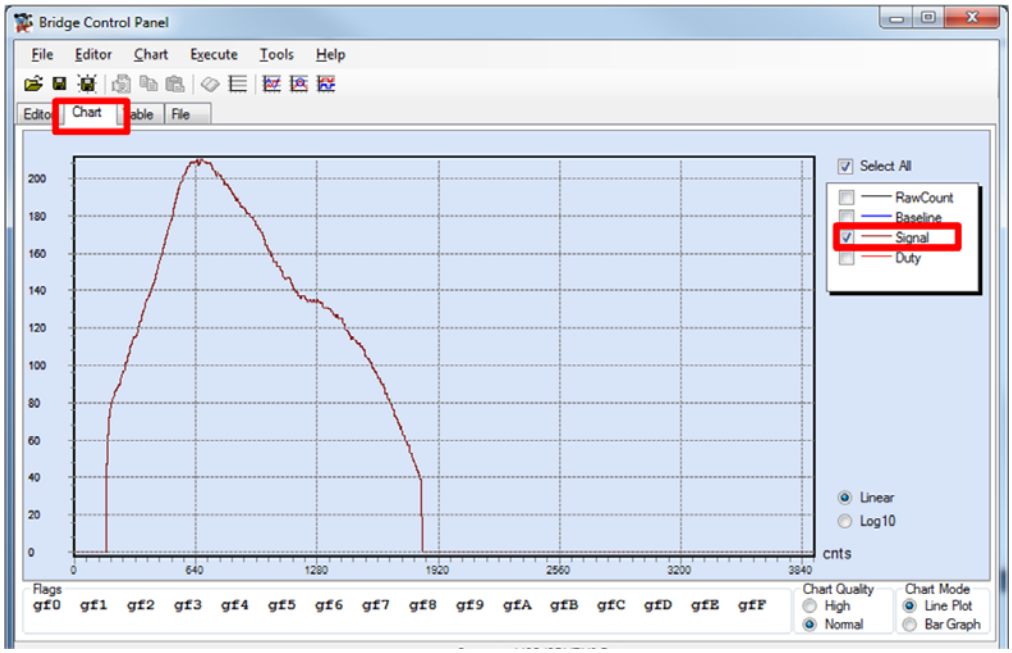
图 5-25. Bridge Control Panel （桥接控制面板） — Protocol Execution （执行协议）



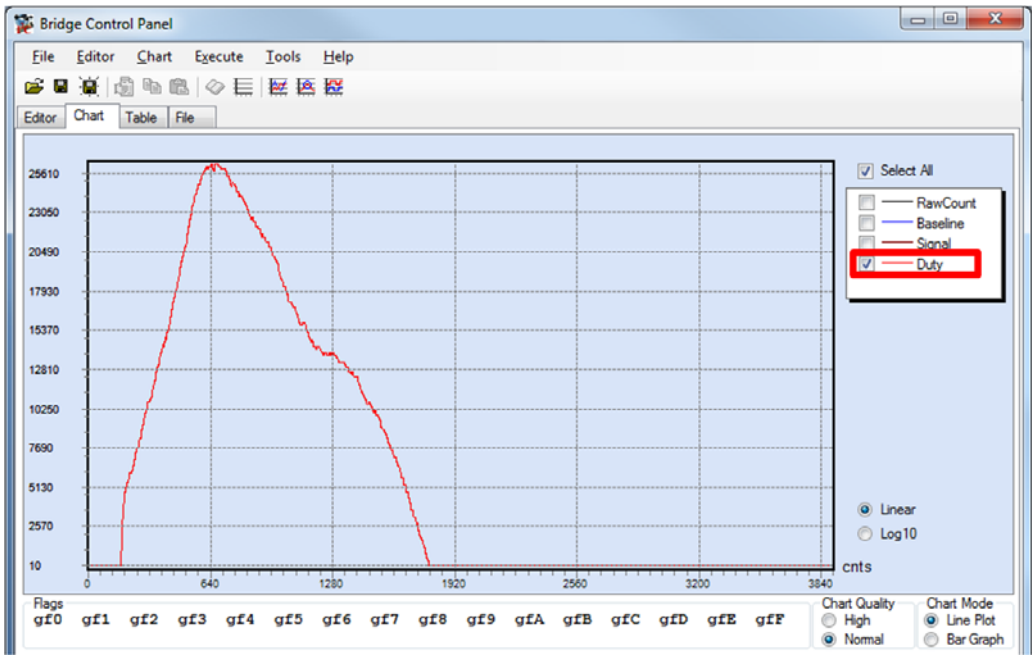
9. 这时应该开始接收数据。点击 **Chart** 选项卡以查看所需图形，如图 5-26 所示。

图 5-26. Bridge Control Panel （桥接控制面板） — 用于查看调试数据的图表

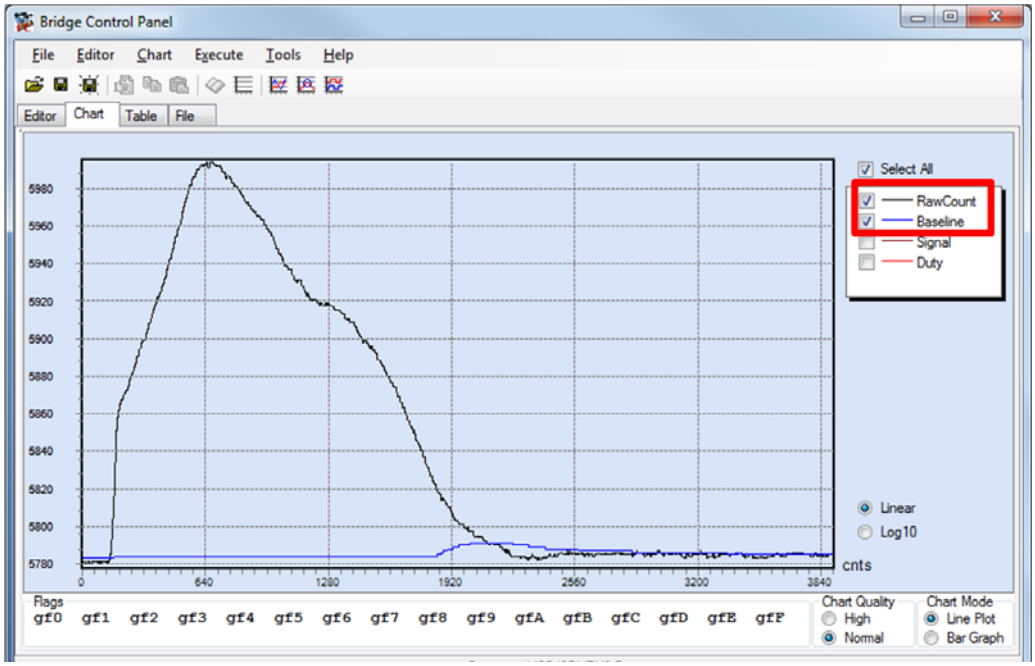
**信号**




**LED亮度（PRSm占空比）**



**原始计数和基线**


Bridge Control Panel
File Editor Chart Execute Tools Help
Editor Chart Table File
Select All
RawCount
Baseline
Signal
Duty
5980
5960
5940
5920
5900
5880
5860
5840
5820
5800
5780
0 640 1280 1920 2560 3200 3840
cnts
Linear
Log10
Flags
gf0 gf1 gf2 gf3 gf4 gf5 gf6 gf7 gf8 gf9 gfA gfB gfC gfD gfE gfF
Chart Quality
High
Normal
Chart Mode
Line Plot
Bar Graph

## 5.4 项目：使用 I²C 调谐器的 CapSense 触摸板

### 5.4.1 项目概述

*CY8CKIT_40_CapSense_I2C.cyprj* 项目使用 SmartSense 来说明 CapSense 触控板的实现，并使用基于 EzI2C 的 CapSense 调谐器窗口查看触摸板的坐标。该项目为使用 SmartSense（最小调试）的简单实现。 PSoC 4000 的 EzI2C 模块通过基于 PSoC 5LP 的 USB-I²C 桥接连接至 PC GUI。该项目采用 SmartSense 功能，通过该功能可以将所有的 CapSense 参数自动设置为最佳值。在 GUI 中可以监控参数设置项，但不能对其进行任何更改。在手动调试方法中，在 GUI 中可以更改参数设置项，并可以看到输出结果（请参考 CapSense 设计指南和 CapSense 组件数据手册中的内容，了解有关手动调试的更详细信息）。

**备注：**该项目要求将触控扩展板插入 PSoC 4000 Pioneer 套件基板上，如图 5-27 所示。

图 5-27. 触控板的套件设置

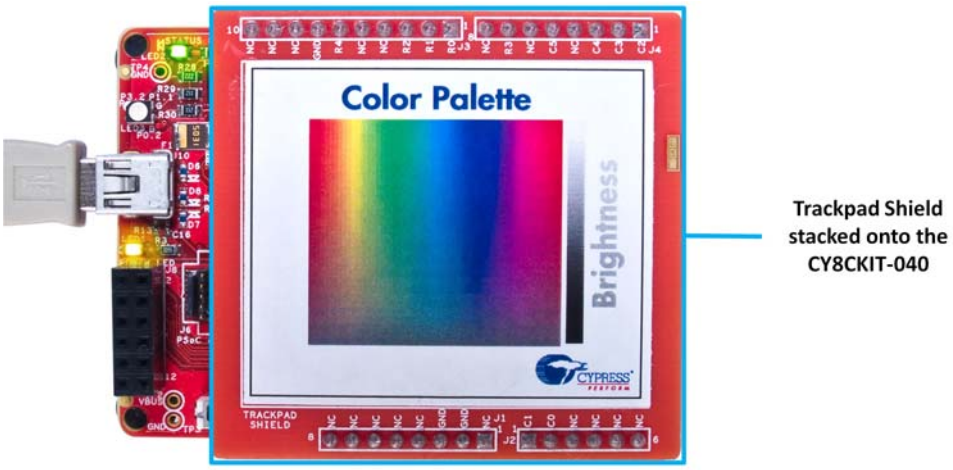
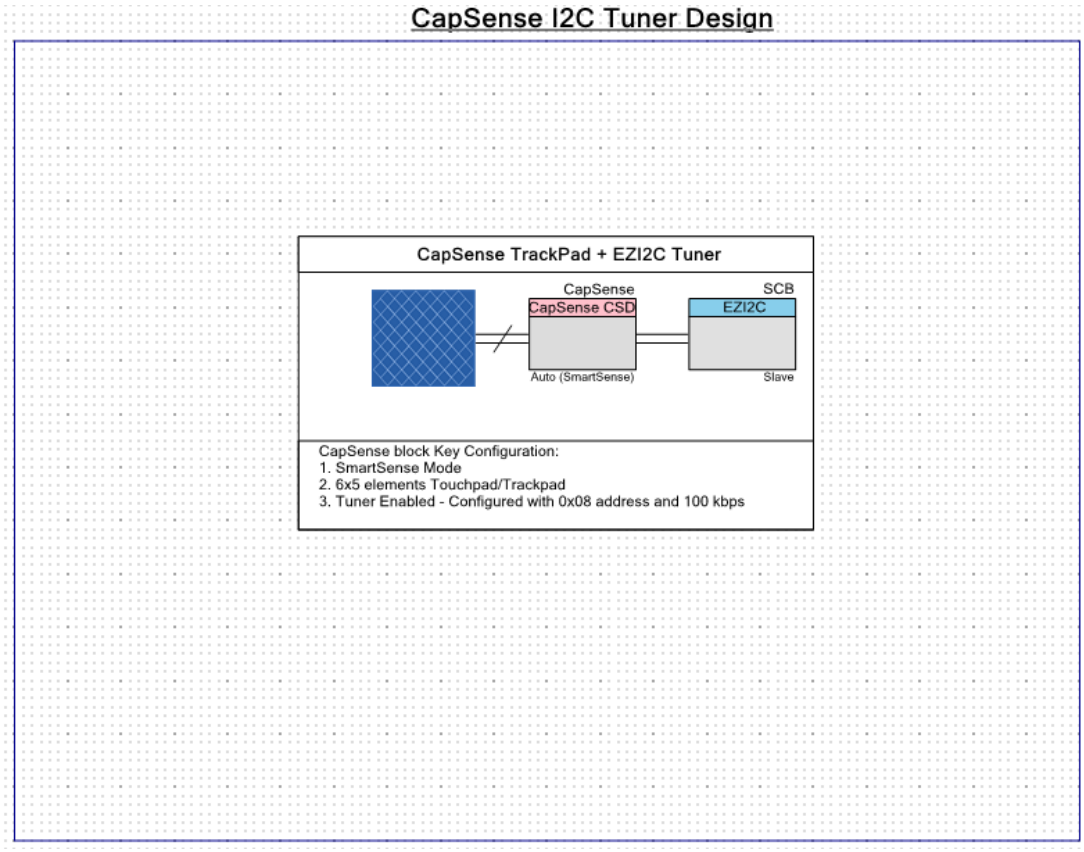


图 5-28. 带 I²C 调谐器的 CapSense 触控板的 PSoC Creator 原理图



## 5.4.2 项目说明

### 5.4.2.1 *PSoC Creator 组件配置*

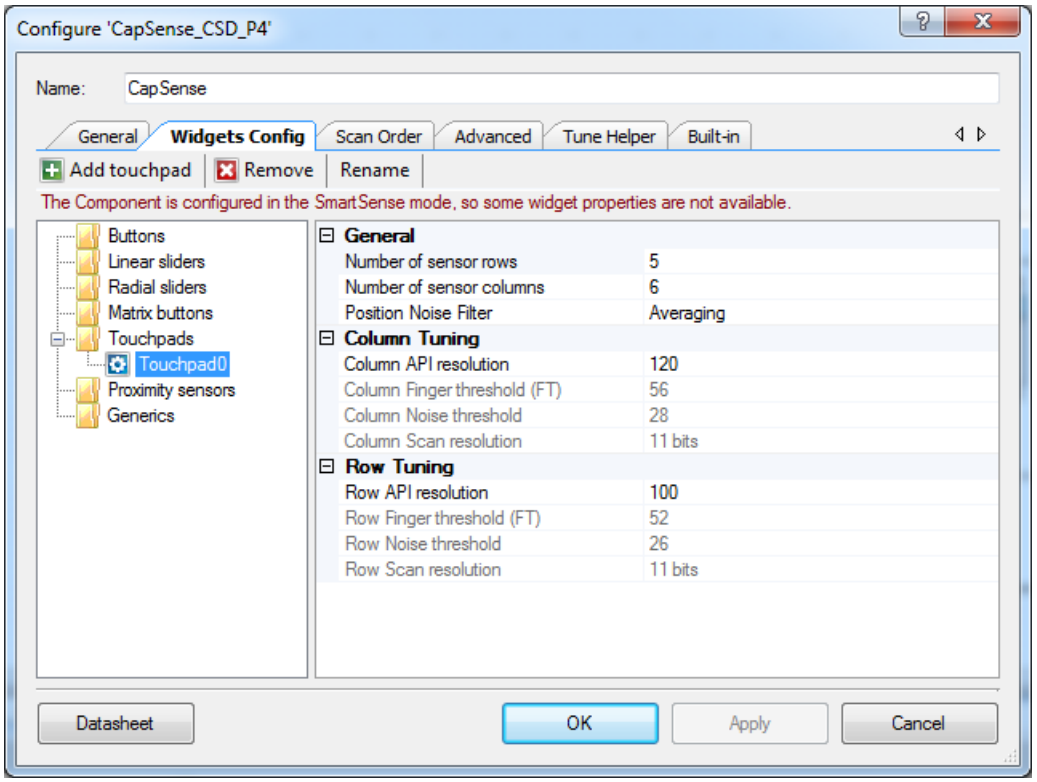
#### CapSense

在 SmartSense 自动调试模式下，使用一个 6x5 CapSense 触摸板 widget 对 CapSense 组件进行配置，配置参数见表 5-9。

表 5-9. CapSense 组件参数

| 参数 | 选项卡显示 | 数值 | 说明 |
|---|---|---|---|
| Tuning method（调试方法） | General（通用） | 自动（SmartSense） | 针对不同的系统环境对灵敏度进行自动调整。 |
| Threshold mode（阈值模式） | | 自动 | 实时计算阈值以保证信噪比达到 5:1 |
| Raw data noise filter （原始数据噪声滤波器） | | 一阶IIR 1/4滤波 | 过滤掉噪声或原始计数中不必要的毛刺。可以根据需求进行调整该设置。 |
| TouchPad0 | Widgets Config（Widget 配置） | - | 添加和配置触摸板，如图 5-29 所示。 |
| Analog Switch Drive Source（模拟开关驱动源） | Advanced（高级） | PRS-Auto | 降低 EMI 辐射，并增强 EMC 的抗干扰能力。 |
| Sensor auto-reset （传感器自动复位） | | 禁用 | 在设计中不需要传感器自动复位。如果应用需要，则添加该特性。 |
| Low Baseline Reset（低基线复位） | | 5 | 取决于系统的数值。根据用户的需求来配置该项。 |
| Inactive Sensor Connection （无效传感器连接） | | 接地 | 接近回路不扫描时不会进行充电。 |
| Shield （屏蔽） | | 禁用 | 在设计中不使用该参数。 |
| Guard Sensor（保护传感器） | | 禁用 | 在设计中不使用该参数。 |
| Cmod precharge（Cmod 预充电） | | 通过Vref缓冲进行预充电 | 因为只有一个传感器，所以具有足够的 Vref 进行预充电。GPIO 快速预充电时， Cmod 电压不会降得太低。 |
| Sensitivity（灵敏度） | Scan Order（扫描顺序） | 5 | 如果要选择所有传感器，可通过按住 **[CTRL]** 或 **[Shift]** 按键并点击所有传感器实现。选中传感器时，在 **Sensitivity** 字段输入一个数值。 |
| Enable Tuner helper （使能调谐器助手） | Tune Helper（调谐助手） | 使能 / 选中 | 名称 = ‘SCB’ |

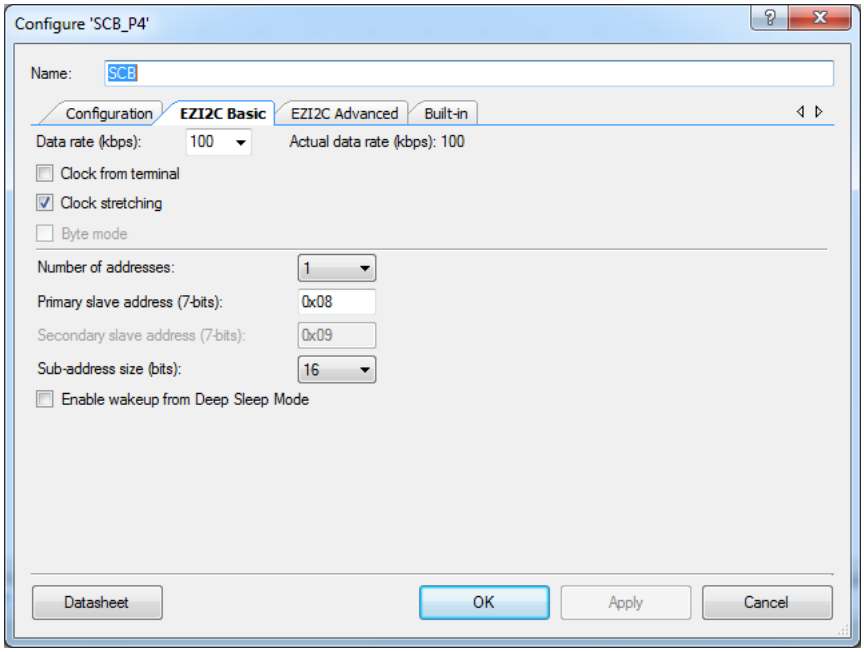
图 5-29. CapSense 触摸板参数




## SCB （EzI2C 模式）

CapSense 调谐器使用 EzI2C 模式中配置的串行通信模块 （SCB）。图 5-30 显示了该组件的参数。

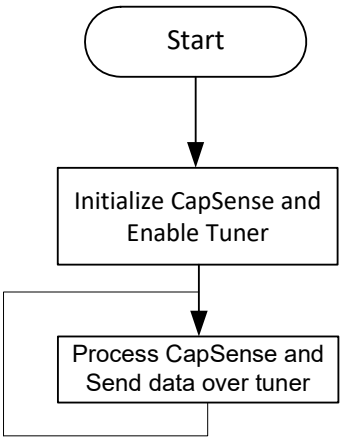
图 5-30. SCB （EZI2C 模式）组件参数



### 5.4.2.2 *固件的详细信息*

图 5-31 显示的是 *main.c* 中执行代码的流程图。

图 5-31. 使用 I²C 调谐器的 CapSense 触摸板项目的流程图




### 5.4.2.3 *硬件连接*

将触控扩展板插入套件的 Arduino 插座上，如图 5-27 所示。在电路板上以硬连线方式实现其它连接。

在**工作区浏览器**中打开 *CY8CKIT_040_CapSense_I2C.cydwr*，然后选择适合的引脚。

表 5-10. 引脚连接[1]

| 引脚名称 | 端口名称 |
|---|---|
| CMOD | P0_4 |
| Trackpad_X0 | P0_3 |
| Trackpad_X1 | P0_7 |
| Trackpad_X2 | P0_6 |
| Trackpad_X3 | P0_5 |
| Trackpad_X4 | P0_0 |
| Trackpad_X5 | P0_1 |
| Trackpad_Y0 | P1_4 |
| Trackpad_Y1 | P1_5 |
| Trackpad_Y2 | P1_6 |
| Trackpad_Y3 | P1_0 |
| Trackpad_Y4 | P1_7 |
| I2C_SCL | P1_2 |
| I2C_SDA | P1_3 |

1. Trackpad_X0（Touchpad0_Col0_TP）至 Trackpad_X5（Touchpad0_Col5_TP）映射到触控板的 COL5 至 COL0，使触控板 x 轴左对齐。

图 5-32. 为 CapSense I²C 项目选择引脚

| Alias | Name | Port | Pin | Lock |
|---|---|---|---|---|
| Cmod | \CapSense:Cmod\ | P0[4] CSD:c_mod, SRSS:ext_clk, TCPWM:tr in[4], CSD:comp | 5 | ☑ |
| Touchpad0_Col0__TP | \CapSense:Sns[0]\ | P0[3] TCPWM:tr_in[3] | 4 | ☑ |
| Touchpad0_Col1__TP | \CapSense:Sns[1]\ | P0[7] | 11 | ☑ |
| Touchpad0_Col2__TP | \CapSense:Sns[2]\ | P0[6] | 10 | ☑ |
| Touchpad0_Col3__TP | \CapSense:Sns[3]\ | P0[5] | 9 | ☑ |
| Touchpad0_Col4__TP | \CapSense:Sns[4]\ | P0[0] TCPWM:tr_in[0] | 1 | ☑ |
| Touchpad0_Col5__TP | \CapSense:Sns[5]\ | P0[1] TCPWM:tr_in[1], CSD:comp | 2 | ☑ |
| Touchpad0_Row0__TP | \CapSense:Sns[6]\ | P1[4] TCPWM:tr_underflow | 16 | ☑ |
| Touchpad0_Row1__TP | \CapSense:Sns[7]\ | P1[5] TCPWM:tr_overflow | 17 | ☑ |
| Touchpad0_Row2__TP | \CapSense:Sns[8]\ | P1[6] TCPWM:tr_overflow, TCPWM:line_out_compl, TCPWM:tr underflow, CSD:comp | 18 | ☑ |
| Touchpad0_Row3__TP | \CapSense:Sns[9]\ | P1[0] | 12 | ☑ |
| Touchpad0_Row4__TP | \CapSense:Sns[10]\ | P1[7] SRSS:ext_clk, TCPWM:tr_compare_match | 19 | ☑ |
| | \SCB:scl\ | P1[2] SCB:i2c_scl | 14 | ☑ |
| | \SCB:sda\ | P1[3] SCB:i2c_sda | 15 | ☑ |

## 5.4.3 验证输出

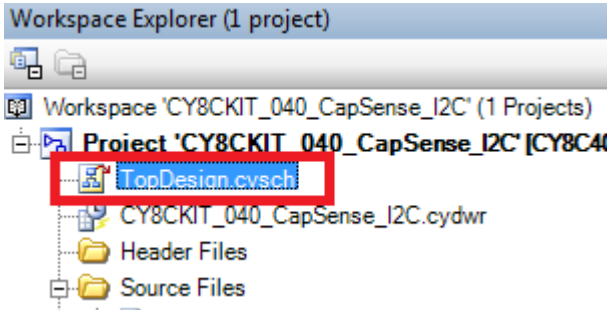
创建和编程代码示例，然后复位器件。按下面介绍的各步骤启动 CapSense Tuner 窗口。

### *5.4.3.1 启动 Tuner （调谐器）窗口*

需要启动和运行 PSoC Creator 中的调试器窗口，以执行代码示例。请按照下面各步骤启动 GUI：

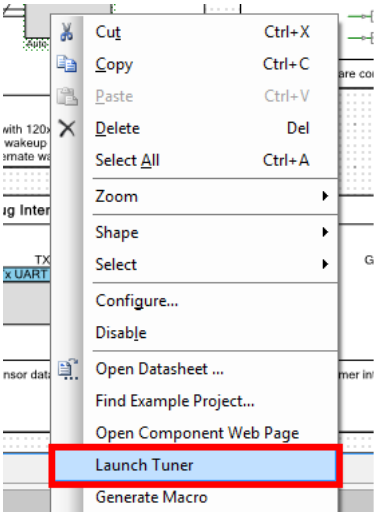
1. 打开项目的 *TopDesign.cysch* 文件，如图 5-33 所示。

图 5-33. 顶层设计文件



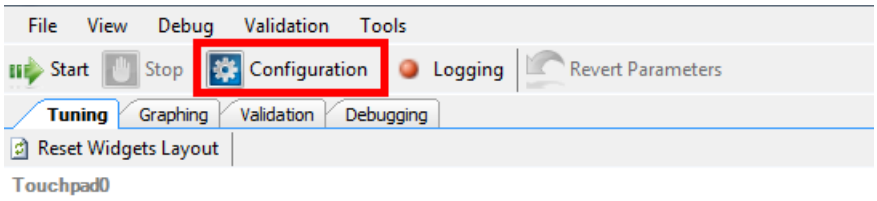
2. 要启动调试器，请右键单击 PSoC Creator 中的 CapSense 组件，然后点击 **Launch Tuner**（启动调谐器），如图 5-34 所示。

图 5-34. Launch Tuner （启动调试器）



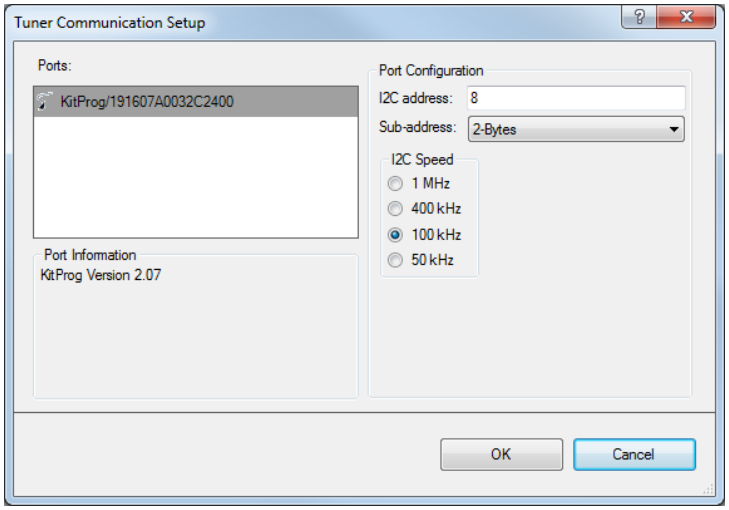

3. 打开调试器窗口。点击 **Configuration** （配置），以打开配置窗口，如图 5-35 所示。

图 5-35. 调试器窗口




4. 设置 I²C 通信参数，如图 5-36 所示。

图 5-36. I²C 通信

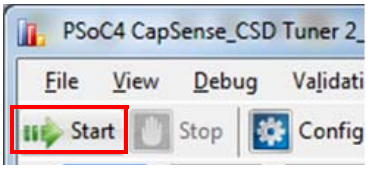



5. 点击 **OK** 按键，以使设置的内容生效。

### 5.4.3.2 验证输出

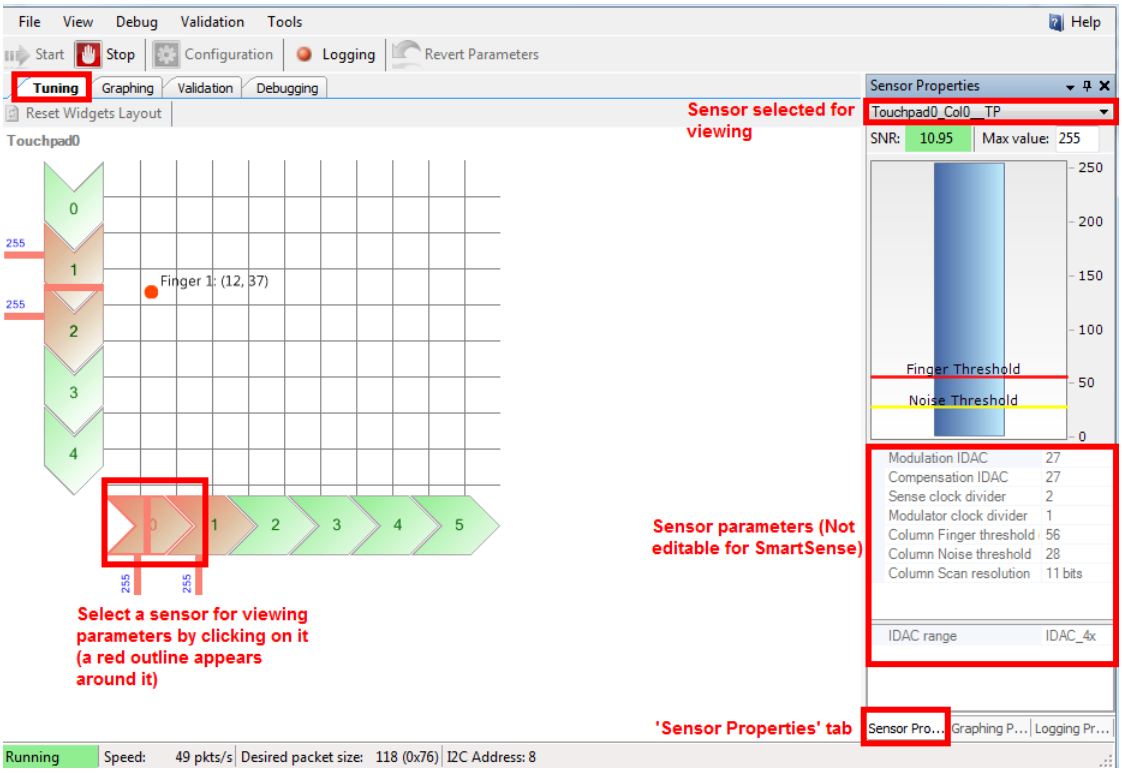
1. 要开始执行扫描和通信，请在调试器窗口中点击 **Start**，如图 5-37 所示。

图 5-37. 开始通信




2. 在 **Tuning** （调试）选项卡中选择一个传感器。选定的传感器有一个红色边框标志。各 CapSense 参数位于右下角。您不能对各项设置进行更改，因为该项目使用了 “ 自动调试 ” 功能；该功能会自动设置所有参数。触摸选定的传感器，并观察 Tuner 窗口的响应。

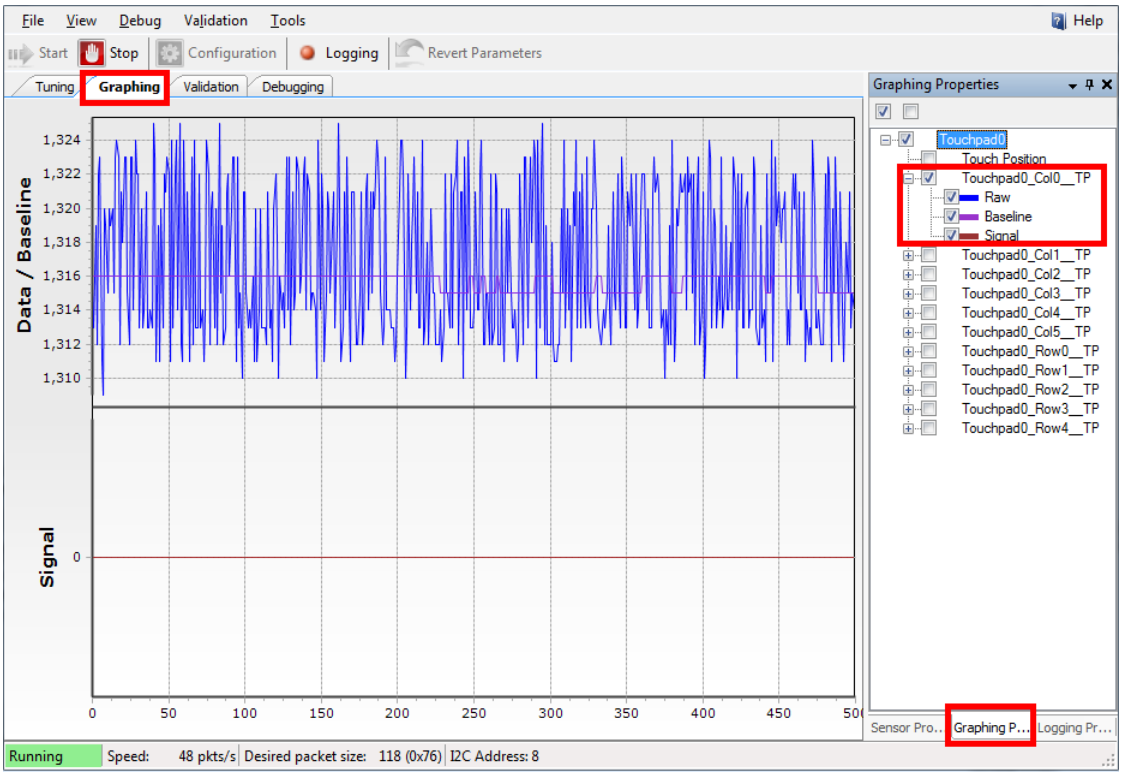
图 5-38. Widget 测试




3. 在 **Graphing** （图形）选项卡中，每个传感器的 CapSense 数据 （原始计数、基线和信号 （差值计数））都是以图形显示的。

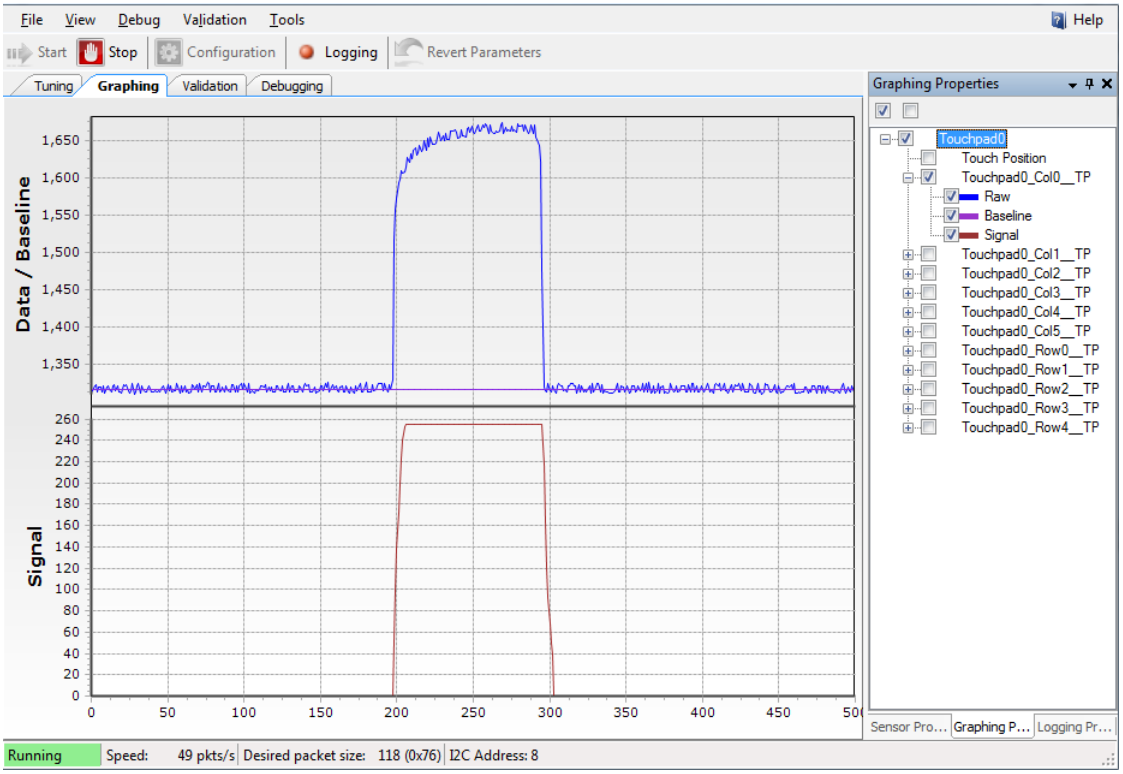
4. 选择需要查看的传感器参数，如图 5-39 所示。选定参数的图形将会出现。

图 5-39. 传感器参数图形



5. 触摸某个传感器或滑条元素，并查看原始计数和信号的增加，如图 5-40 所示。

图 5-40. 原始计数增加

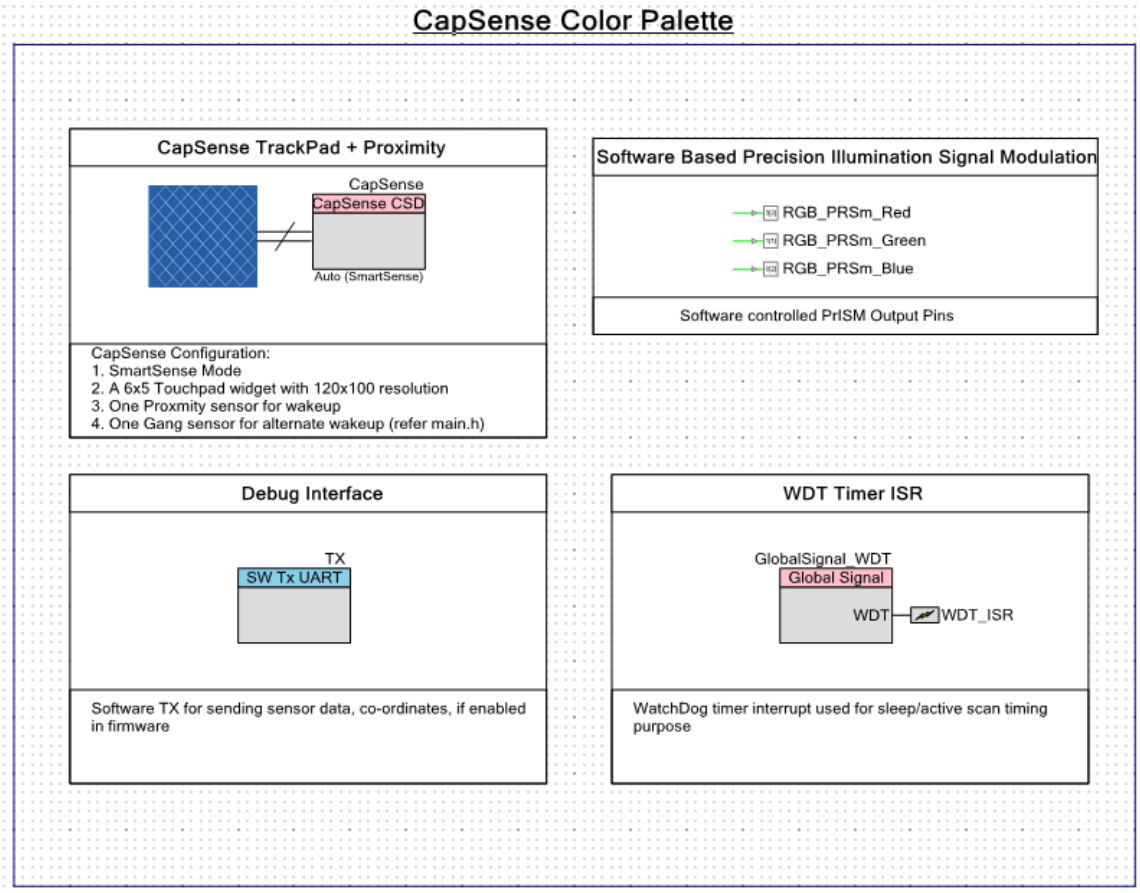


# 5.5 项目：调色板 （Color Palette）

## 5.5.1 项目概述

*CY8CKIT_040_Color_Palette.cyprj* 项目展示的是 PSoC 4000 器件能够与电容式触控板相连接，并根据选定的颜色 （通过在触控板顶部上触摸标签进行选择）控制 RGB LED。该标签也会包含一个滑条区域 （触控板的一部分），用于控制 RGB LED 的颜色亮度。该项目通过使用一条导线来说明该器件的接近感应能力。通过使用软件精度照明信号调制器 （PrISM）控制 LED 亮度。第 89 页上的固件的详细信息 介绍了项目的详细信息。图 5-41 显示的是项目的顶层设计原理图。

图 5-41. 调色板项目的 PSoC Creator 原理图设计




## 5.5.2 项目说明

### *5.5.2.1 PSoC Creator 组件配置*

#### CapSense

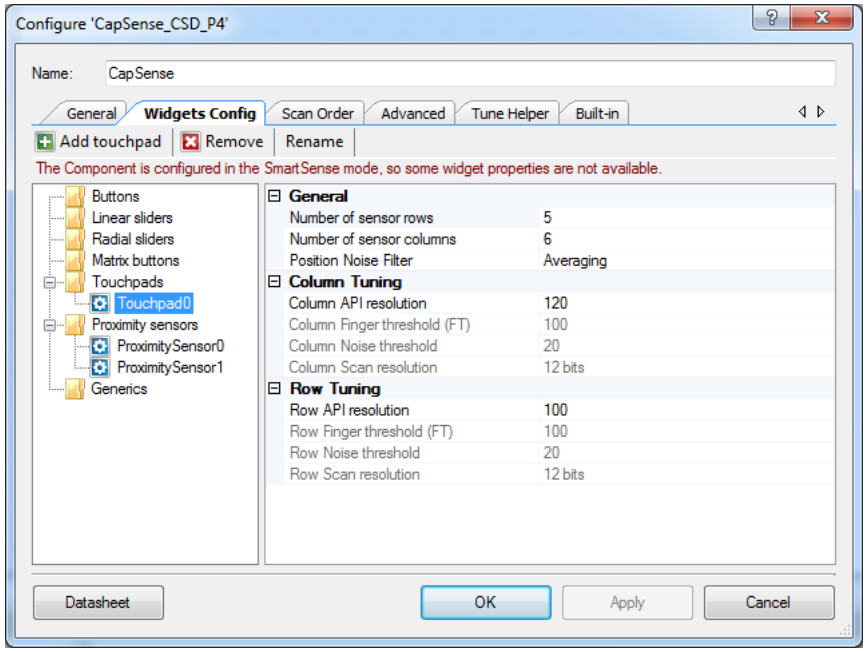
在 SmartSense 自动调试模式下使用表 5-11 中列出的参数对 CapSense 组件进行配置。该示例的设计包含了一个 6x5 触摸板、一个专用的接近感应传感器和一个虚拟 / 组合的接近感应传感器。

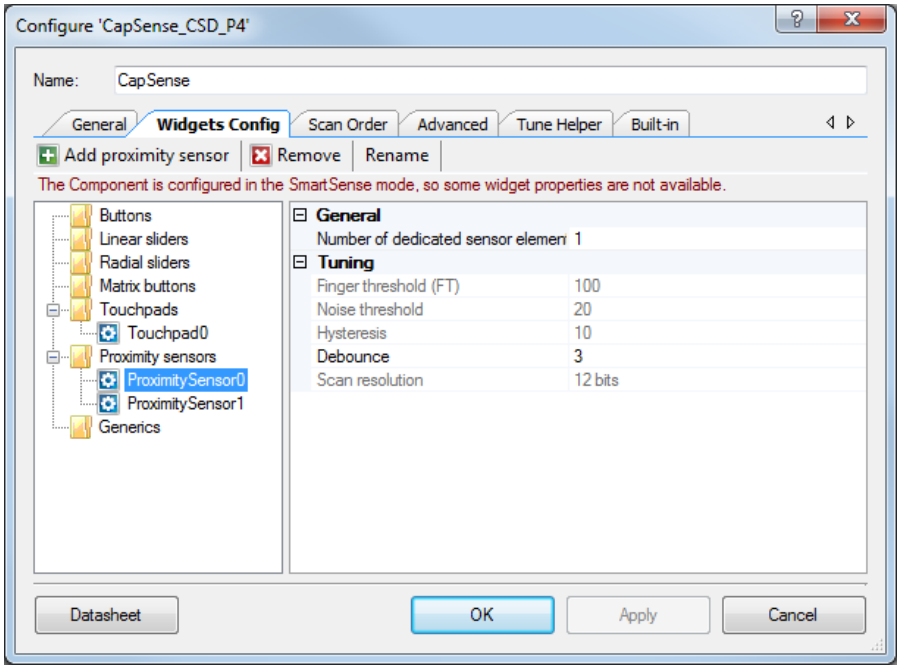
请参考 PSoC 4 CapSense 设计指南或 CapSense 组件数据手册中介绍的内容，了解所使用的各种 widget 和参数。

表 5-11. CapSense 组件参数

| 参数 | 选项卡显示 | 数值 | 说明 |
|---|---|---|---|
| Tuning method（调试方法） | General（通用） | 自动（SmartSense） | 针对不同的系统环境会自动调整灵敏度。 |
| Threshold mode（阈值模式） | | 自动 | 实时计算阈值以保证信噪比达到 5:1 |
| Raw data noise filter （原始数据噪声滤波器） | | 一阶 IIR 1/8 滤波 | 过滤掉噪声或原始计数中不必要的毛刺。可以根据需求进行调整该设置。 |
| TouchPad0 | Widgets Config（Widget 配置） | – | 添加和配置触摸板，如图 5-42 所示。 |
| ProximitySensor0 | | – | 添加和配置接近感应传感器，如图 5-42 所示。 |
| ProximitySensor1 | | – | 添加并配置接近感应传感器（联组），如图 5-42 所示。请留意，该传感器是虚拟的传感器，与触摸板的行元素联组，用以唤醒系统。 |
| Analog Switch Drive Source（模拟开关驱动源） | Advanced（高级） | PRS-Auto | 可用于降低 EMI 辐射，并增强 EMC 抗干扰能力。 |
| Sensor Auto Reset（传感器自动复位） | | 禁用 | 在设计中不需要该特性。如果应用需要，则可以添加该特性。 |
| Low Baseline Reset（低基线复位） | | 5 | 取决于系统的数值。可根据每个用户的需求对其进行配置。 |
| Inactive Sensor Connection（无效传感器连接） | | 接地 | 确保接近回路不扫描时不会进行充电。 |
| Shield （屏蔽） | | 禁用 | 在设计中不使用。 |
| Guard Sensor（保护传感器） | | 禁用 | 在设计中不使用。 |
| Cmod precharge（Cmod 预充电） | | 通过 Vref 缓冲进行预充电 | 因为只有一个传感器，所以对于预充电来说， Vref 足够用； Cmod 电压不会降得太低，以快速进行 GPIO 预充电。 |
| Sensitivity（灵敏度） | Scan Order（扫描顺序） | 4 | 如果要选择所有传感器，可通过按住 **[CTRL]** 或 **[Shift]** 按键，并点击所有触摸板传感器实现。选中后，在 “Sensitivity” 字段中输入数值。 |
| | | 1 | 对于最大范围，请点击 **ProximitySensor0**，并将灵敏度设置为 ‘1’ 。 |
| | | 5 | 请点击 **ProximitySensor1**，并将灵敏度设置为 5，以在扫描处于睡眠状态时，使扫描的分辨率最小，这样能够降低功耗。 |
| Sensor ganging（传感器机械连接） | | - | 将触摸板行单元连接到 ProximitySensor1，如图 5-43 所示。 |
| Enable tuner helper（使能调谐器助手） | Tuner helper（调谐器助手） | 禁用 / 未选中 | 未使能调校工具。 |

图 5-42. CapSense 触摸板参数



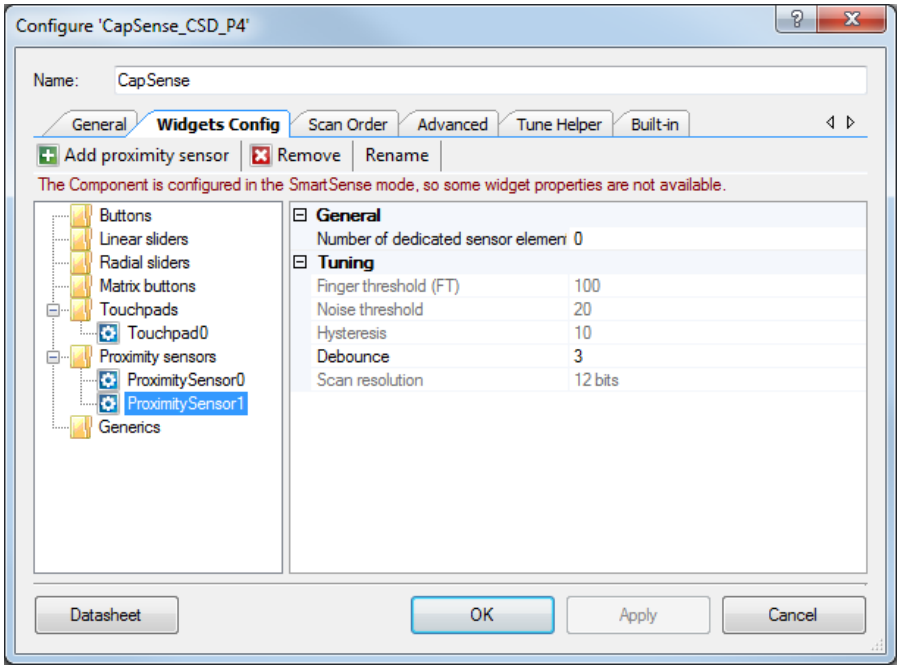
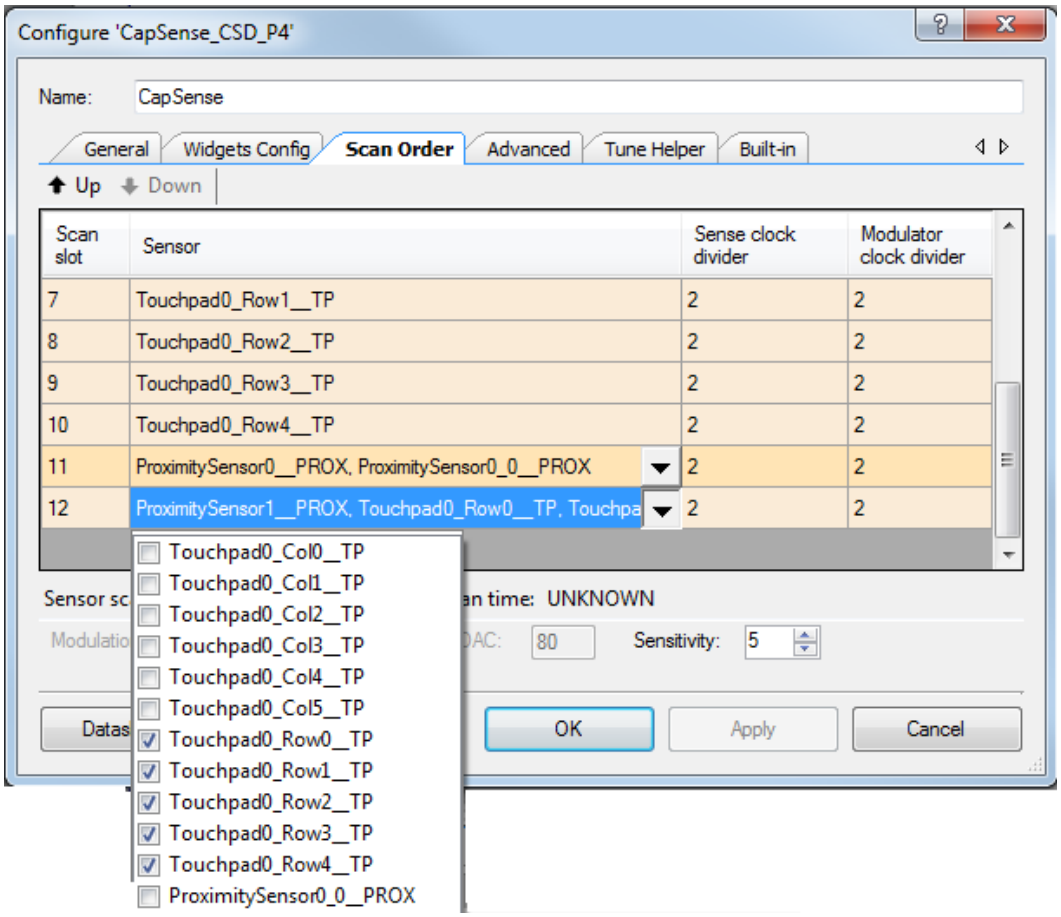



图 5-43. CapSense 组件配置窗口中的 Scan Order （扫描顺序）选项卡

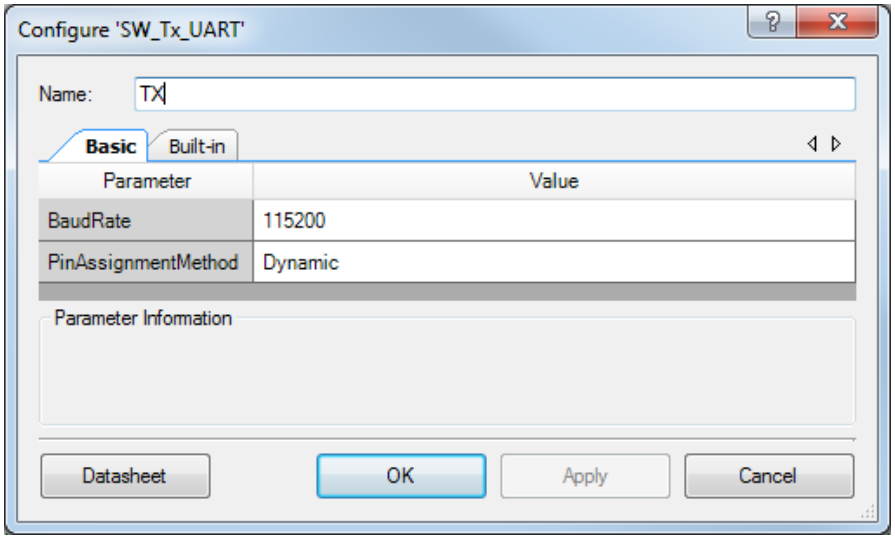


### TX （SW TX UART）

如果在固件中使能了软件传输 TX，那么将使用该 TX 发送传感器数据，以进行调试（请参考项目中的 *main.h* 文件）。图 5-44 显示了组件的配置情况。通过使用 RS-232 连接器 （CapSense 控制器和 PC 之间的电压转换器）、或 CY8CKIT-040 PSoC 5 LP UART 桥接中提供的 USB-UART 桥接、或配置为 UART 桥接的 CY3240 桥接 （如 AN2397 中所述），可以将 SW TX 发送到 PC 内。通过 *main.h* 中提供的 TX_PORT 和 TX_PIN 宏，可以在固件中配置 TX 引脚 （如果已使能该引脚）。

**备注：**默认情况下， P3[0] 作为固件中的 TX 使用，并且在 CY8CKIT-040 中， P3[0] 以硬连线方式连接到 PSoC 5LP USB-UART 桥接的 RX 线。因此，如果 P3[0] 作为 TX 使用，并在固件中使能它，则不能使用 SWD 调试。请参考第 49 页上的编程示例项目 中的最后步骤，了解如何在同一个项目中使用 SWD 调试和 TX。

图 5-44. 软件 UART TX 组件参数




### RGB_PRSm_<Color> （数字输出引脚）

用于将软件 PrISM 输出驱动到相应的 <color> LED。这是一个标准的、由固件控制的强驱动输出引脚。

### GlobalSignal_WDT （ISR 的全局信号参考）

该组件用于将 WDT ISR 路由到 ISR 组件。然后，在固件中配置该 ISR，使它在 Sleep_Scan 模式中使用 WDT 生成周期性的唤醒信号。

### 5.5.2.2 *固件的详细信息*

#### 固件结构

调色板以模块化格式进行编写，在单独的函数、源文件和头文件中提供不同的功能。通过这些内容，用户可以更好地了解固件结构，并能轻松地修改固件，以满足应用需求。通过使用 *main.h* 文件中定义的各种宏，可以修改固件的关键内容，以满足各种应用的要求。

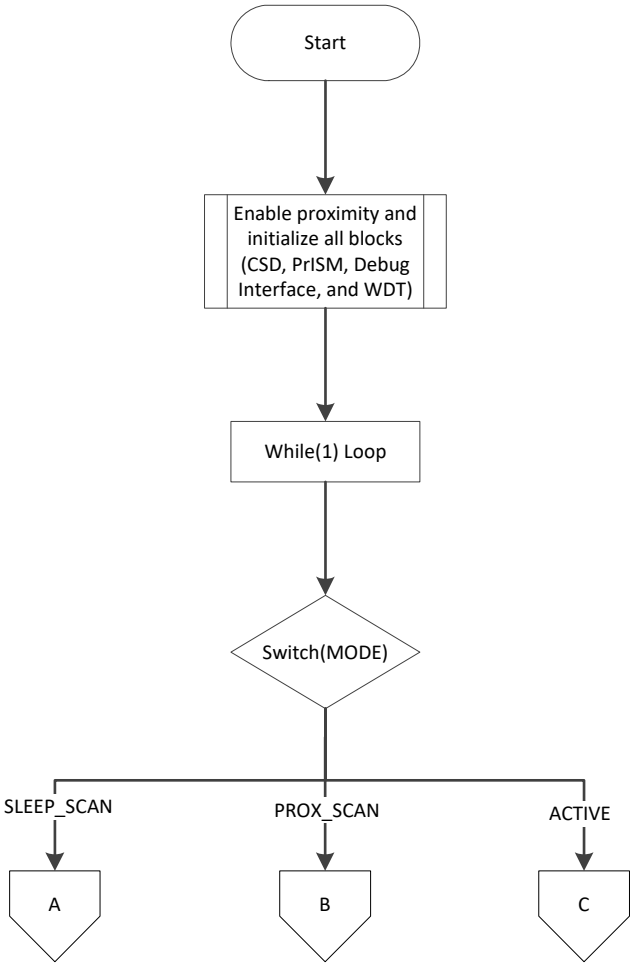
表 5-12. “Color Palette” 项目中的源文件和头文件

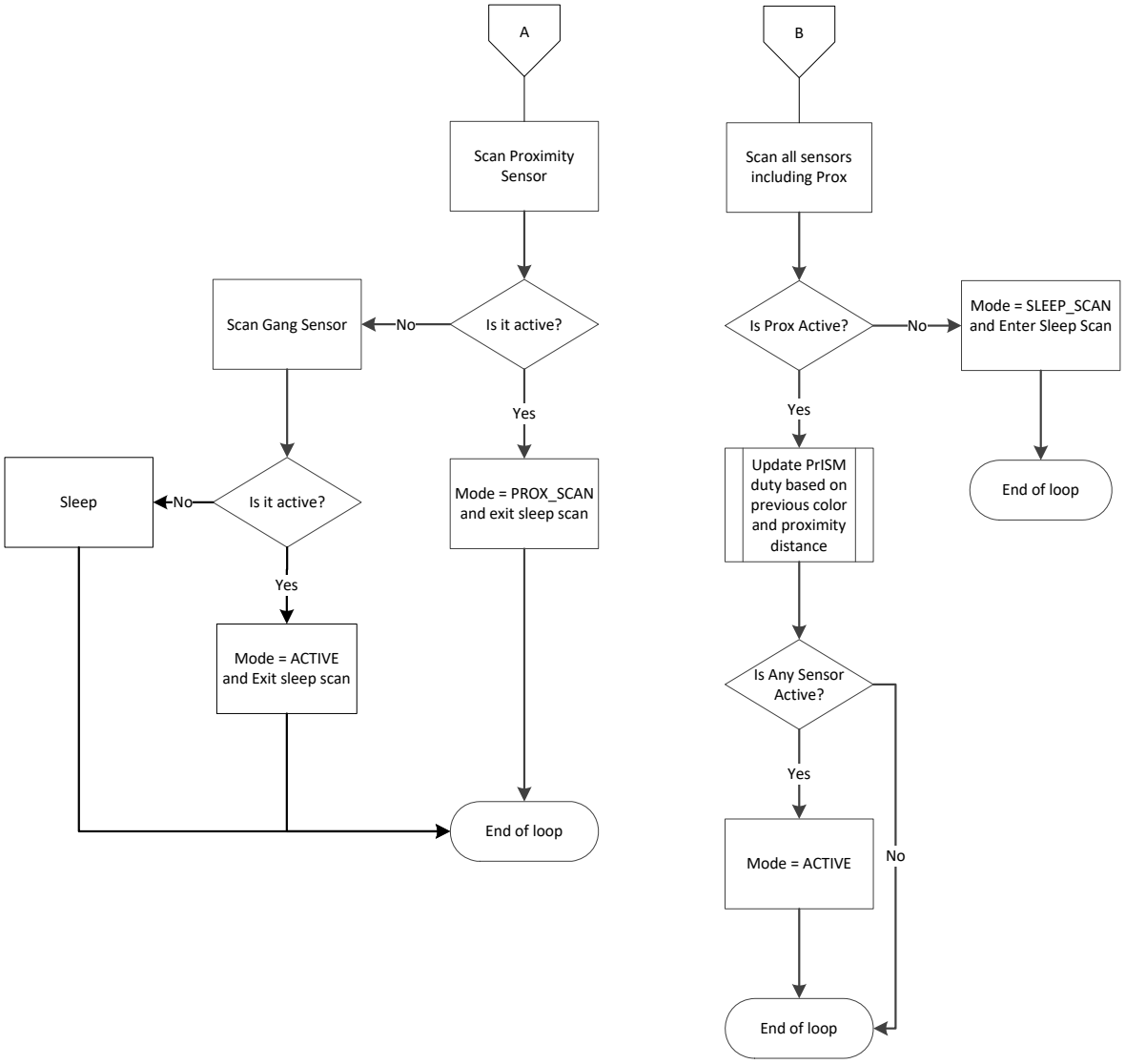
| 文件名称 | 用途 |
|---|---|
| RGB_PRSm.c、RGB_PRSm.h | 这些文件包含了软件 PRS 调制器的实现，通过该调制器可控制 LED 的亮度。 |
| main.c | 该文件包含了固件中所使用的所有函数定义。 |
| main.h | 该文件包含了所有宏定义、导入的变量以及导出的函数声明。通过改变本文件中的宏值，可以修改固件的关键内容。 |

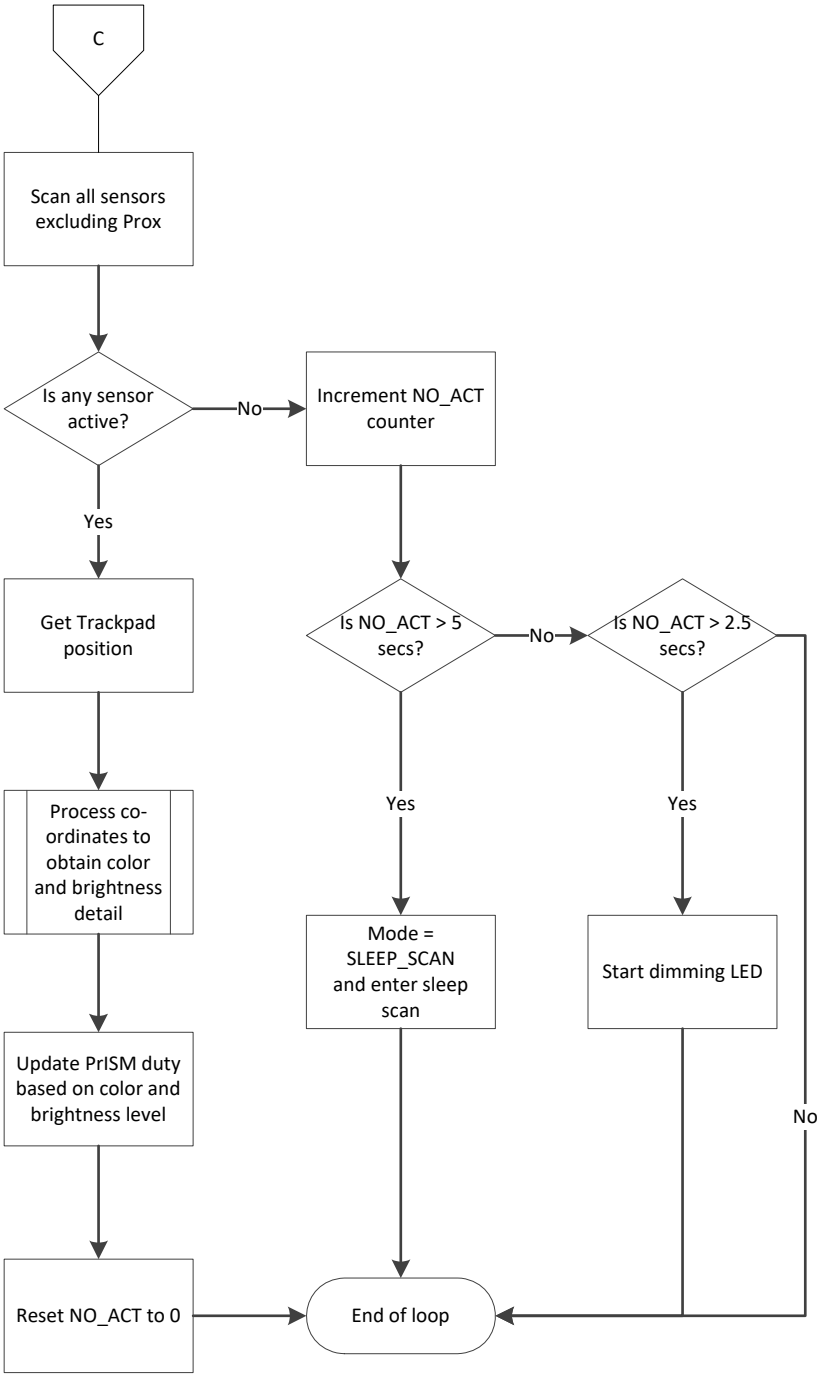
#### 固件流程图

图 5-45 显示的是在 main.c 中执行代码的流程图。

图 5-45. 调色板项目的流程图




A
Scan Proximity Sensor
Is it active?
No
Scan Gang Sensor
Yes
Mode = PROX_SCAN and exit sleep scan
Is it active?
No
Sleep
Yes
Mode = ACTIVE and Exit sleep scan
End of loop
B
Scan all sensors including Prox
Is Prox Active?
No
Mode = SLEEP_SCAN and Enter Sleep Scan
End of loop
Yes
Update PrISM duty based on previous color and proximity distance
Is Any Sensor Active?
Yes
Mode = ACTIVE
No
End of loop


C
Scan all sensors excluding Prox
Is any sensor active?
No
Increment NO_ACT counter
Yes
Get Trackpad position
Is NO_ACT > 5 secs?
No
Is NO_ACT > 2.5 secs?
Process co-ordinates to obtain color and brightness detail
Yes
Yes
Mode = SLEEP_SCAN and enter sleep scan
Start dimming LED
Update PrISM duty based on color and brightness level
No
Reset NO_ACT to 0
End of loop

## 固件实现介绍

固件包括三种不同的模式：活动扫描 （ACTIVE_SCAN）、睡眠扫描 （SLEEP_SCAN）和接近感应扫描 （PROX_SCAN）

- 活动扫描：活动扫描主要执行下面两个任务：
  - 扫描触摸板
  - 根据触摸板的操作更新颜色

扫描触摸板传感器和接近感应传感器 （如果使能了它们），然后依次执行下面两个子任务：

- 保持器件处于活动扫描模式中
- 计算触摸坐标

触摸板或接近感应传感器上的任何操作都会使器件保持其活动扫描状态。如果停止操作，固件将逐渐减弱 RGB 颜色亮度，并使器件进入睡眠扫描模式。从最后一次检测到触摸 / 接近操作到 LED 亮度降低的时间由 LED_DIM_THRESHOLD 宏定义。 LED 调暗的比率则由 LED_DIM_RATE 宏定义。请参考 *main.h* 文件的内容，了解各种宏及其用途。

图 5-46 和图 5-47 显示了项目中实现的两种不同的颜色选择模型。通过使用 *main.h* 中定义的 DO_SATURATION 宏，可以选择两项中的一项。

图 5-46. 色调和亮度控制实现模型 （默认模型）

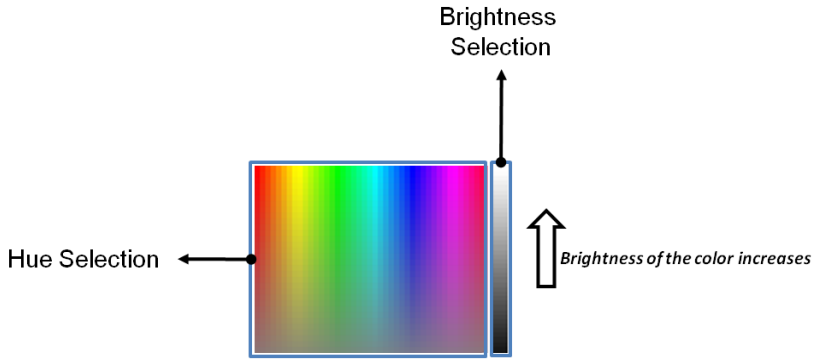



图 5-47. 色调、饱和度和明度值 （HSV）实现模型

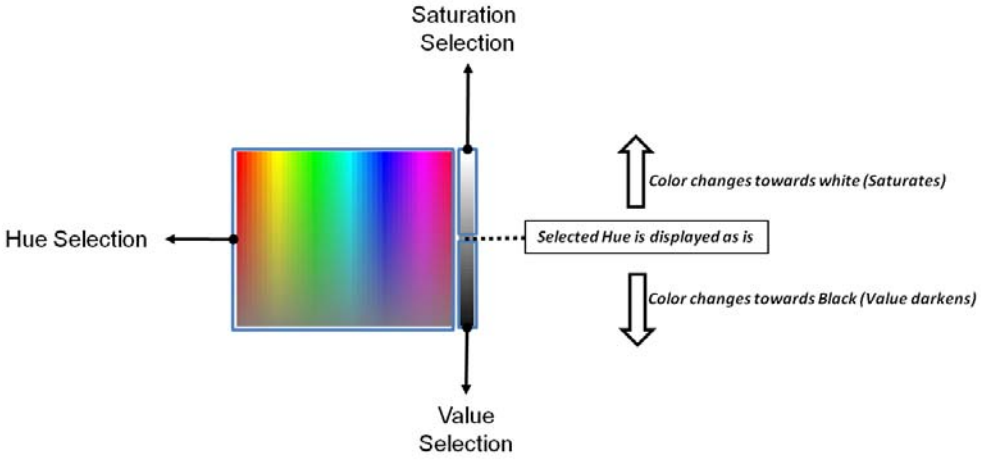


图 5-48 到图 5-51 介绍了为 RGB LED 生成亮度级的过程。调色板和饱和度滑条都是触控板区域的一部分。下面各步骤总结了该流程。

1. 按照图 5-48 中所显示的映射情况取得触控板坐标。图 5-49 显示的是在固件 （main.h）中针对触控板 / 调色板区域使用的各种宏定义。

图 5-48. 选择触控板坐标

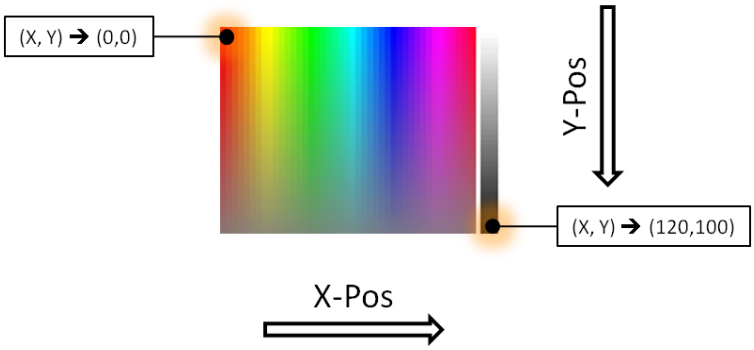
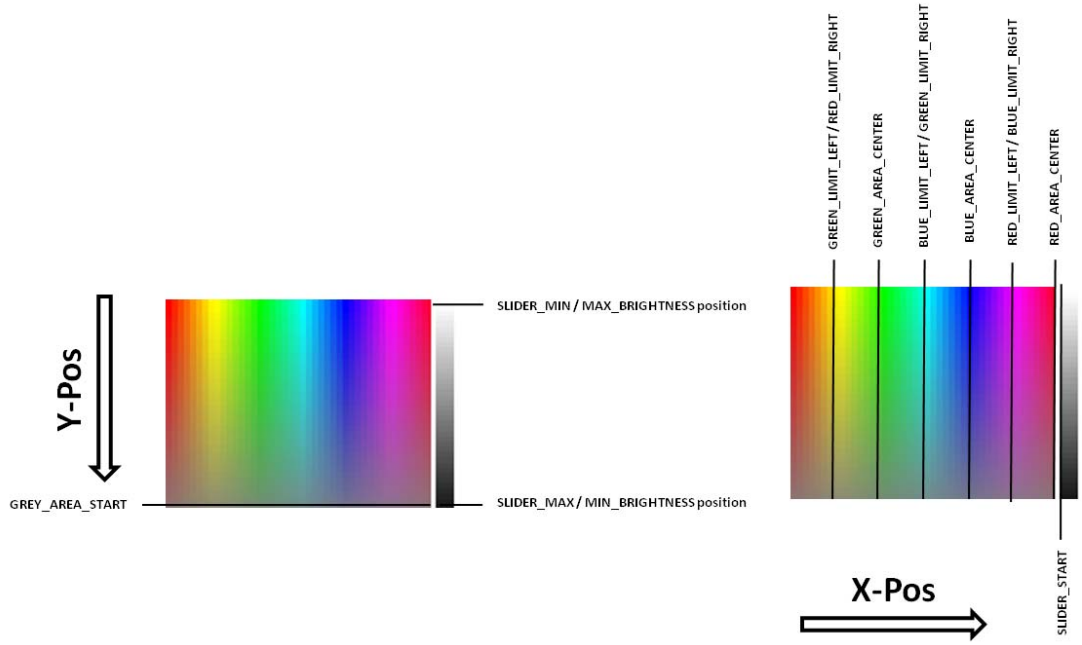


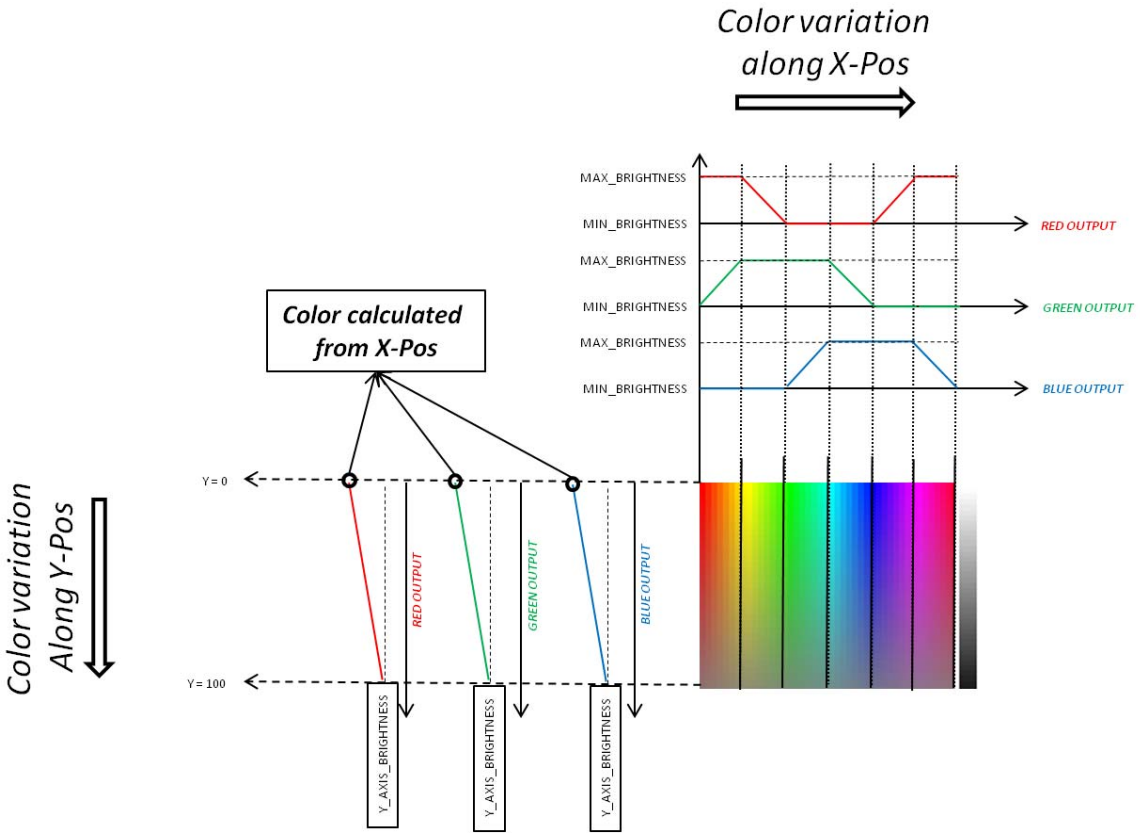

图 5-49. X-Y 轴的宏定义



2. 图 5-50 显示了使用饱和度前 RGB 颜色亮度级的计算情况。在 X 轴上，调色板被分为六个窗口。在每个窗口中，一个 RGB 颜色以最大亮度发光，一个以最低亮度发光，并且一个从最大亮度切换为最低亮度，或反过来切换。图 5-49 中所示的宏标记这些窗口边界。 < 颜色 >_AREA_CENTER 标记<颜色>窗口的中间，该<颜色>窗口的两边会标记<颜色>的最大亮度。<颜色>_LIMIT_LEFT/ < 颜色 >_LIMIT_RIGHT 宏标记每个 < 颜色 > 区域的边沿。该该区域里面， < 颜色 > 的亮度最大。

   在 Y 轴上，每个颜色的亮度会以 X 轴上选定的亮度值 （百分之 x）的一半 （百分之 50）进行切换。先通过窗口规则和 X 坐标在 X 轴上计算颜色级。然后再通过 Y 坐标对所得到的颜色进行处理。

图 5-50. 根据调色板上的 X 和 Y 轴计算亮度

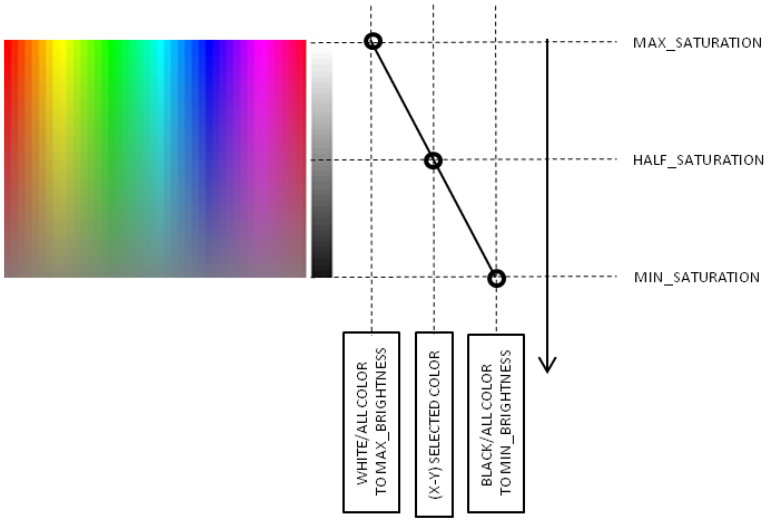


3. 执行第二步骤得到主颜色亮度 （位于调色板上）后，将使用由滑条选择的亮度级或饱和度级。通过使用 *main.h* 中定义的 DO_SATURATION 宏，可以选择亮度级或者饱和度级。

   a. 在禁用 / 注释掉 DO_SATURATION 宏时，颜色的亮度控制将被使用。亮度控制指的是通过滑条输出对三种颜色强度 （从第二步骤计算得出）进行简单的比例分配。在通电时，亮度的默认值为 100％或最大值。

   b. 如果使能 DO_SATURATION 宏，那么饱和度将被使用，如图 5-51 所述。在饱和度控制模式下时，如果滑条输出值等于最大值的一半，则保留该亮度。如果该输出值小于最大值的一半，则所有颜色的亮度都转为最低亮度 （黑 / 更暗）。如果该值大于最大值的一半，则所有颜色的亮度都会转为最高亮度 （白 / 更亮）。上电时，饱和度的默认值为 50% 或 HALF_SATURATION 值。每个颜色变暗或变亮的数值与滑条位置到其中心间的距离成正比。请参见图 5-51。

图 5-51. 根据滑条输出计算饱和度




- 睡眠扫描模式

  在睡眠扫描模式中，固件扫描接近感应传感器或组合的传感器 （触摸板行元素中的传感器被组合在一起）或扫描上述两者。在编译项目过程中，通过注释掉 ENABLE_PROXIMITY 或 ENABLE_GANG 宏，可以配置该选项。如果保持该两个宏，器件将在睡眠模式下扫描两个传感器；如果禁用该两个宏，则使器件始终处于活动扫描模式中。

  - 如果在睡眠扫描模式下使用接近感应传感器，并且器件在传感器上检测到任何接近操作，那么器件将进入接近扫描模式。
  - 如果在睡眠扫描模式下使用组合传感器，并且在扫描过程中器件在触控板上检测到某个操作，那么器件将直接进入活动扫描模式。

  通过更改 *main.h* 中定义的 SLEEP_TIMER_PERIOD 宏，可以配置睡眠扫描模式下器件进行扫描的速率。

- 接近扫描模式：

  在接近扫描模式中，器件主要扫描接近感应传感器，并且触摸板传感器也会得到扫描，以便检测发生的操作。先前选定的 RGB 颜色被打开，其亮度与接近感应信号强度成正比。当触摸触控板元素时，器件将进入活动扫描模式。只要接近感应传感器发生某些操作，则器件会保持在接近扫描模式中。如果将手移出了接近感应范围，则器件将返回睡眠扫描模式。

### 输出接口

- 可选的 TX 接口

  固件还具有一个可选的 UART TX 接口。根据要求可以使能或禁用该接口。要想使能或禁用 TX 接口，只要取消注释或注释 *main.h* 中的 TX_ENABLE 宏即可。此外，固件还提供了一个选项，即在 TX 线上发送的两种数据包类型中选择一种。一种数据包以多图形式保存系统中所有传感器的原始计数、基线和信号数据 （请参阅 AN2397，了解有关多图图形工具的信息）。另一种数据包只发送触控板坐标。要想选择这两种数据包类型中的一种，请使用 MINIMAL_TX 宏。如果该宏被注释掉，那么将发送第一种数据包。如果该宏被使用，则发送第二种数据包。

  **备注：**要想查看MINIMAL_TX数据，可通过...\Firmware\PSoC 4\Bridge Files文件夹内的*CapSense Color Palette (Minimal TX) - Commands.iic*和*CapSense Color Palette (Minimal TX) - Variables.ini* 文件设置 BCP，以进行查看 （如第 67 页上的查看 UART 数据 所述）。针对数据包大小的情况，可以使用 AN2397 中的多图工具查看第一种类型的数据包。

- 用于控制 LED 亮度的软件 PrISM

  固件采用了一个由软件实现的 7 位 PRS，它通过精密照明信号调制 （PrISM）控制 LED 的亮度。实现过程中将使用 Cortex-M0 CPU 子系统中的 SysTick 定时器。SysTick 定时器将生成一个中断，在中断期间计算出 PRS，以便在 LED 引脚的输出端上生成一个伪随机信号。有关该实现的详细信息，请查看项目中的 *RGB_PRSm.h* 和 *RGB_PRSm.c* 文件。用于生成 PRS 的多项式的固定值为 [7,6,5,2]。 PRS 的周期或重复率为 127 个计数。如果 SysTick 定时器以 20 kHz 的频率生成 ISR，那么得到的输出频率将为 ~150 Hz。因为输出信号是一个脉冲宽度不固定的信号，并且它也是一个高频率信号 （其最大频率为 10 kHz，平均频率约为 5 kHz），所以该信号很容易被过滤掉，目的是为了避免发生闪烁。这种闪烁通常发生于与该信号频率相同的 PWM 信号。

### 5.5.2.3 *硬件连接*

该示例要求 CapSense 触控扩展板连接，如第 74 页上的项目：使用 I2C 调谐器的 CapSense 触摸板一节中介绍的内容。在电路板上以硬连线的方式连接 RGB LED 和 CMOD。此外，还可以连接一个接近感应环，如图 5-52 中的第一项设置所示。为了有效地使用固件，可以将接近感应环构建成围绕触控板的一个环路，如图 5-53 所示。使用第二项设置时，当手接近触控板时， LED 将被打开，以显示先前的颜色。

**备注：**对于接近感应环路，可以使套件附带的导线（长度为 4 英寸）弯曲，使之形成一个直径为 1 至 2 英寸的环路；获得的直径范围将与快速接近手指模式的环路直径大约相同。为了得到更大的直径范围，请使用更长的导线 / 更大的循环。

图 5-52. 第一项设置

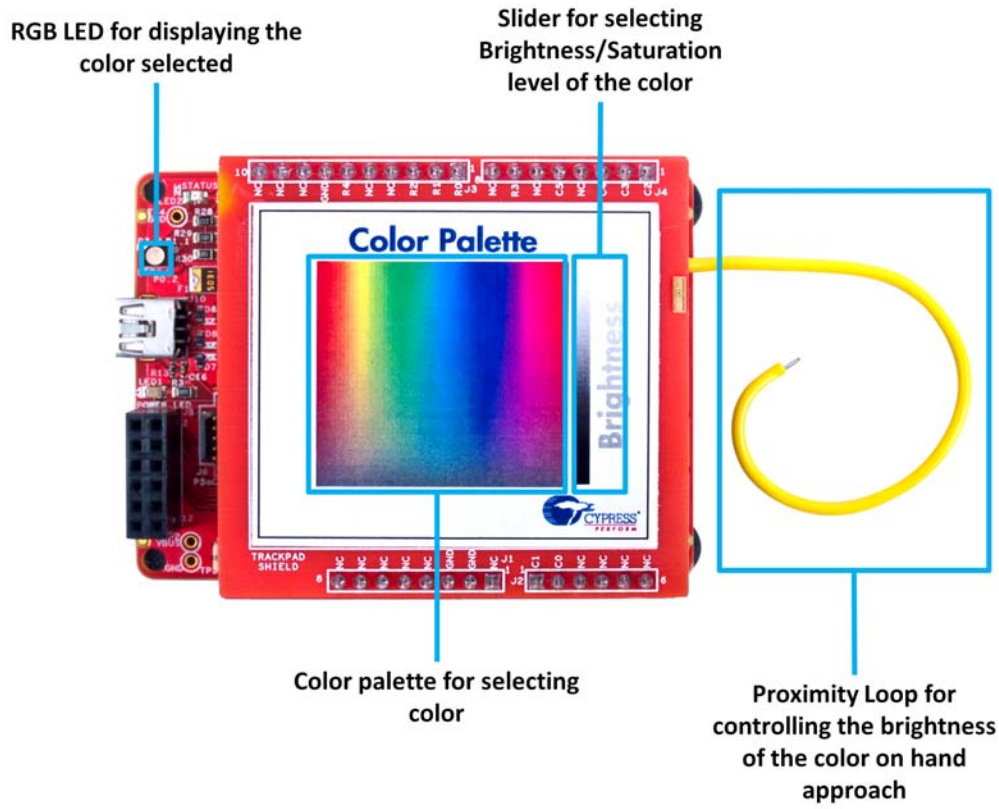
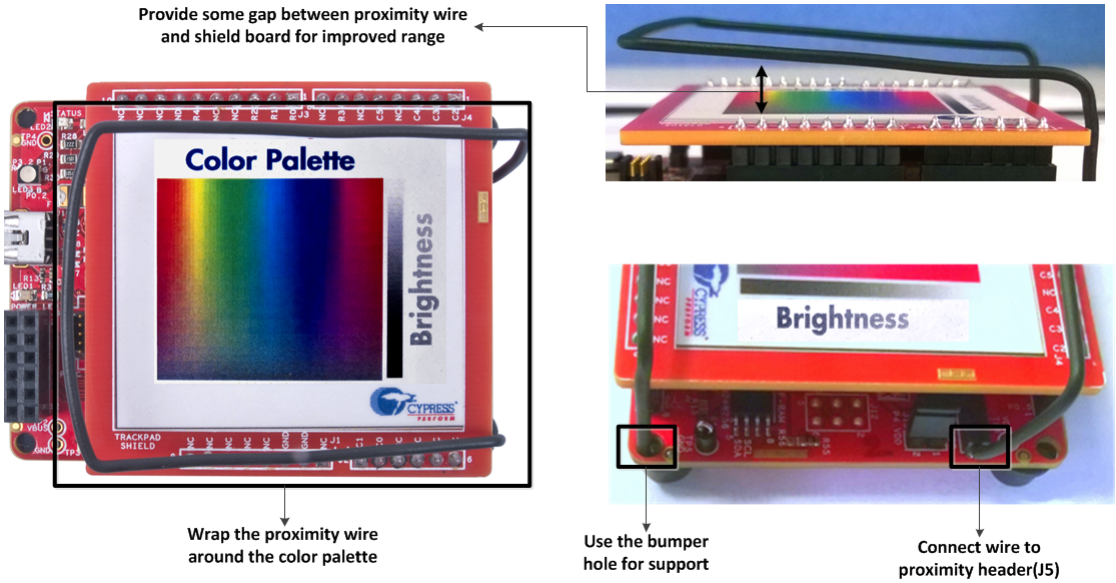


图 5-53. 第二项设置




在 **Workspace Explorer** 中的 **Source** 垂直选项卡下，打开 *CY8CKIT_040_Color_Palette.cydwr*，并选择合适的引脚。

表 5-13. 引脚连接[1]

| 引脚名称 | 端口名称 |
|---|---|
| Trackpad_X0 | P0_3 |
| Trackpad_X1 | P0_7 |
| Trackpad_X2 | P0_6 |
| Trackpad_X3 | P0_5 |
| Trackpad_X4 | P0_0 |
| Trackpad_X5 | P0_1 |
| Trackpad_Y0 | P1_4 |
| Trackpad_Y1 | P1_5 |
| Trackpad_Y2 | P1_6 |
| Trackpad_Y3 | P1_0 |
| Trackpad_Y4 | P1_7 |
| CMOD | P0_4 |
| 接近 | P2_0 |
| 红色 LED | P3_2 |
| 绿色 LED | P1_1 |
| 蓝色 LED | P0_2 |
| UART TX | P3_0 [2] |

1. Trackpad_X0 （Touchpad0_Col0_TP）至 Trackpad_X5 （Touchpad0_Col5_TP）映射到触控板的 COL5 至 COL0，使触控板 x 轴左对齐。
2. 在固件中选中 （请参考 “main.h” 文件）

图 5-54. 为调色板项目选择引脚

| Alias | Name | Port | Pin |
|---|---|---|---|
| Cmod | \CapSense:Cmod\ | P0[4] CSD:c_mod, SRSS:ext_clk, TCPWM:tr in[4], CSD:comp | 5 |
| Touchpad0_Col0__TP | \CapSense:Sns[0]\ | P0[3] TCPWM:tr_in[3] | 4 |
| Touchpad0_Col1__TP | \CapSense:Sns[1]\ | P0[7] | 11 |
| Touchpad0_Col2__TP | \CapSense:Sns[2]\ | P0[6] | 10 |
| Touchpad0_Col3__TP | \CapSense:Sns[3]\ | P0[5] | 9 |
| Touchpad0_Col4__TP | \CapSense:Sns[4]\ | P0[0] TCPWM:tr_in[0] | 1 |
| Touchpad0_Col5__TP | \CapSense:Sns[5]\ | P0[1] TCPWM:tr_in[1], CSD:comp | 2 |
| Touchpad0_Row0__TP | \CapSense:Sns[6]\ | P1[4] TCPWM:tr_underflow | 16 |
| Touchpad0_Row1__TP | \CapSense:Sns[7]\ | P1[5] TCPWM:tr_overflow | 17 |
| Touchpad0_Row2__TP | \CapSense:Sns[8]\ | P1[6] TCPWM:tr_overflow, TCPWM:line_out_compl, TCPWM:tr underflow, CSD:comp | 18 |
| Touchpad0_Row3__TP | \CapSense:Sns[9]\ | P1[0] | 12 |
| Touchpad0_Row4__TP | \CapSense:Sns[10]\ | P1[7] SRSS:ext_clk, TCPWM:tr_compare_match | 19 |
| ProximitySensor0_0__PROX | \CapSense:Sns[11]\ | P2[0] | 20 |
| | RGB_PRSm_Blue | P0[2] CSD:c_sh_tank, TCPWM:tr_in[2] | 3 |
| | RGB_PRSm_Green | P1[1] TCPWM:line_out | 13 |
| | RGB_PRSm_Red | P3[2] TCPWM:line_out | 23 |

### 5.5.3 验证输出

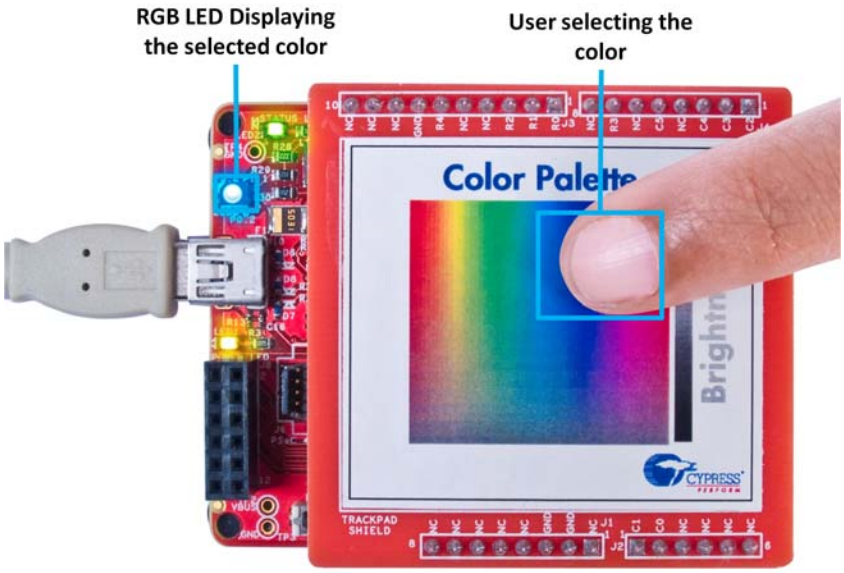
编译并烧写示例目标代码到器件中。 RGB LED 的默认颜色为蓝色。因此，如果您在触摸触控板前先在接近感应传感器上方移动， RGB LED 会变为蓝色。请按照这些步骤验证代码。

1. 触摸触控板上的调色板，如图 5-55 所示。被触摸的颜色会显示在 RGB LED 上。

   **备注：**由于 LED 较大 （与显示屏和纸张上的像素相比），所以可能会在某些点上看到单独的颜色 /LED。要想看到正确的颜色混合，可以将一层额外的扩散物 （如薄纸）放在 RGB LED 上面。另外，根据 LED 上的最大电流以及它们能够生成的最大光度，这些 LED 显示的最大亮度级会有所不同 （亮度 = 电流 × 光度与电流的比例）。例如，当使用一个给定的串行电阻时，红色亮度可能大于蓝色亮度，因为红色的光度与电流的比例更高，且电压下降更低，使得该颜色的电流更高。在这种情况下，可以限制较亮颜色的强度，以确保获得准确的混合色。在示例项目中已实现同样的设置情况，具体是将红色 LED 的强度限制为 ~85%。

   请留意，当亮度滑条被设置为最低 / 零亮度时，在颜色区域上实现的触摸操作不会显示在 LED 上。

图 5-55. 活动扫描模式中的调色板输出



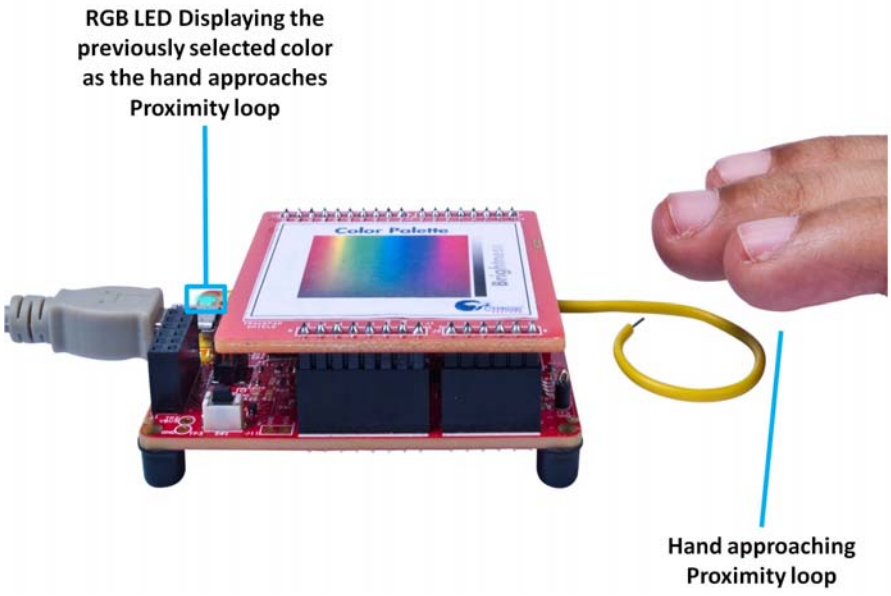

2. 要想调整颜色的亮度，请在触控板右角上的滑条区域内进行移动操作。观察从一端到另一端颜色光度从关闭到全亮的变化。

   **备注：**当从颜色选择区域移动到滑条区域时，为了使触摸区域正常工作，请先从触控板上移除手指，然后再把手指放置在滑条上。当选择原色区域接近于滑条区域时，实现上述操作可以避免意外触摸滑条区域。

3. 从触控板移除手指，然后在 2-3 秒后观察 LED 亮度逐渐降低，以及在大约 5 秒后该 LED 被灭掉。

4. 当 LED 被关闭时，如果将您的手移到接近感应环路，当手进入接近感应范围内的同时，LED 的亮度会逐渐增大；当您的手靠近触控板时，LED 的亮度将达到最大值。LED 的颜色将为先前选定的触控板颜色，如图 5-56 所示。

图 5-56. 接近扫描模式下的调色板输出




## 5.6 PSoC 4000 中的 ADC

从 PSoC Creator 3.2 起，组件目录中将提供一个用于 PSoC 4000 系列的 ADC 组件 （CSD_ADC）。为了开始使用该组件，PSoC Creator 介绍了两个示例。这两个示例都使用 CY8CKIT-040 和某些外部无源组件（电阻和连线）实现所需程序。本节介绍了有关如何访问代码示例以及如何在 CY8CKIT-040 上使用该示例的信息。在本节中，为了进行内容讲解，将使用 **ADC_VoltageInp** 示例作为参考。

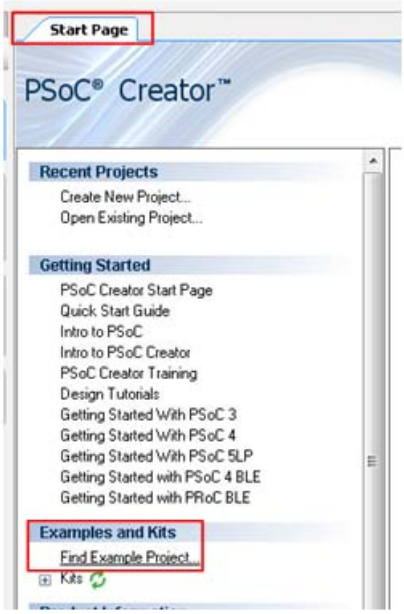
### 5.6.1 使用 PSoC Creator 代码示例：ADC_VoltageInput

请按照下面个步骤打开并使用 PSoC Creator CSD_ADC 代码示例。

1. 依次选择 **Start > All Programs > Cypress > PSoC Creator<version> > PSoC Creator <version>**，启动 PSoC Creator。请留意， PSoC Creator 3.2 或更高版本支持 CSD_ADC 组件。

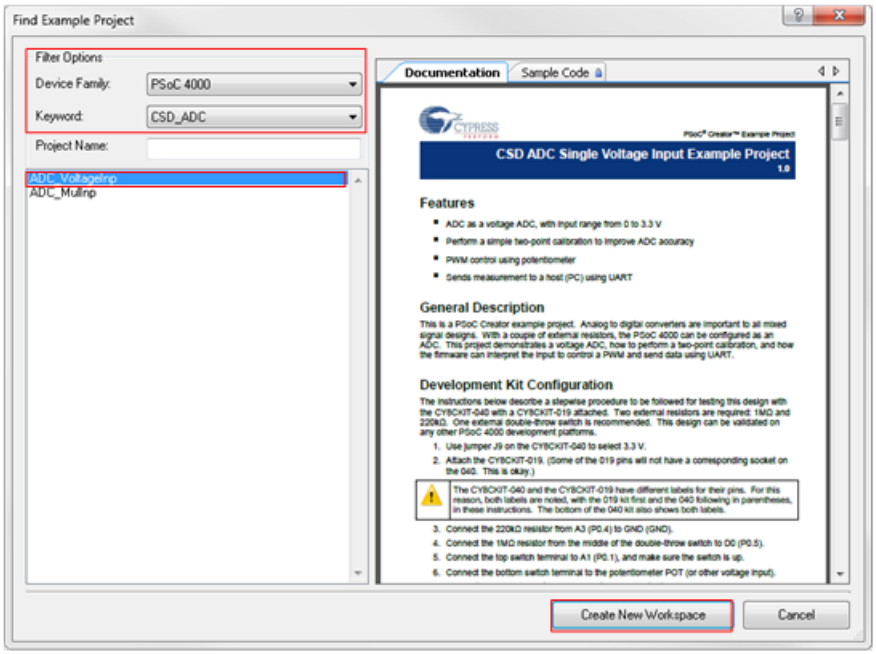
2. 在 **Start page** 上，点击 **Examples and Kits** 下方的 **Find Example Project...**，如 图 5-57 所示。

图 5-57. 从 PSoC Creator 打开代码示例




3. 在 **Find Example Project** 窗口中，将 **Device family** 设置为 **PSoC 4000** 和将 **Keyword** 设为 **CSD_ADC**，如图 5-58 所示。

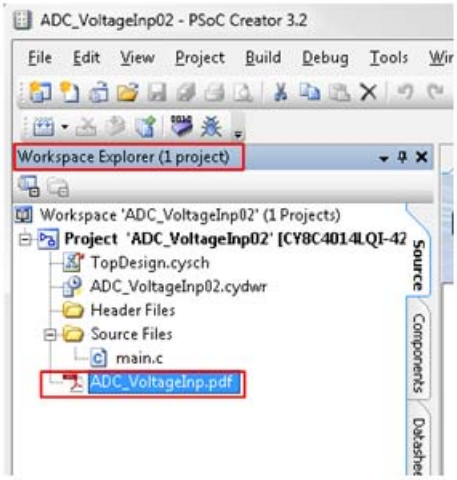
图 5-58. 设置 Find Example 中的 **Filter Options** 项以及选择 **ADC_VoltageInp**




4. 图 5-58 中显示了 CSD_ADC 列表的两个示例。我们将选择 **ADC_VoltageInp** 示例作为参考。
5. 选择 **ADC_VoltageInput**，然后点击 **Create New Workspace** 按键。将工作区保存到所需位置。

6. 该示例介绍了 PSoC 4000 中的一个 ADC 电压实现。该示例还说明了如何执行一个两点校准，以提高准确性。固件根据 ADC 输出控制 PWM 输出，并且 ADC 数据是通过 UART 发送出去。要想进一步了解该项目，请参考创建项目时打开的 **ADC_VoltageInput.pdf** 文件。此外，也可以在 **Workspace Explorer** 中查找该文件，请参见图 5-59。

图 5-59. ADC_VoltageInp 示例的项目数据手册




7. 已在 CY8CKIT-040 上测试过示例项目；要想配置套件，以便在该套件上使用该示例，请遵循 **ADC_VoltageInput.pdf** 文件中**开发套件配置** 部分的内容。请留意，CSD_ADC 组件的当前实现使用了 CapSense 硬件。因此，不能同时使用 CapSense 和 ADC。更多有关信息，请参阅 CSD_ADC 组件数据手册。

# 6. 高级主题

本节介绍了套件的某些高级特性。

**注意：**对于在本部分中创建的所有新示例项目，请重命名组件，如关联的组件图像中所示。这是必需的，因为附带用户指南的 C 文件遵循这些图像中显示的组件名称。

## 6.1 将 PSoC 5LP 作为 USB-I²C 桥接器使用

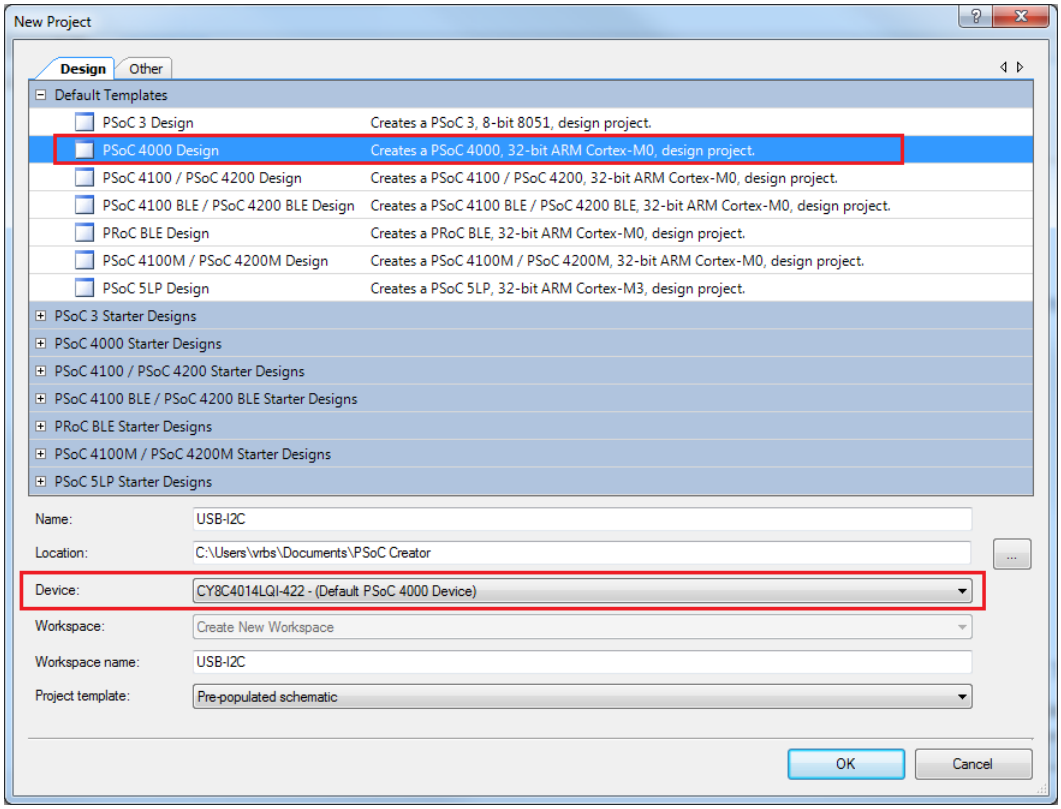
将 PSoC 5LP 作为一个 USB-I²C 桥接器使用，通过该桥接器可以同运行在 PC 上的 USB-I²C 软件进行通信。

**注意：**第 74 页上的项目：使用 I2C 调谐器的 CapSense 触摸板 也通过 CapSense Tuner 窗口使用套件中的 USB-I²C 桥接器。

下述各步骤说明了如何使用 USB-I²C 桥接器在 BCP 与 PSoC 4 器件间进行通信。

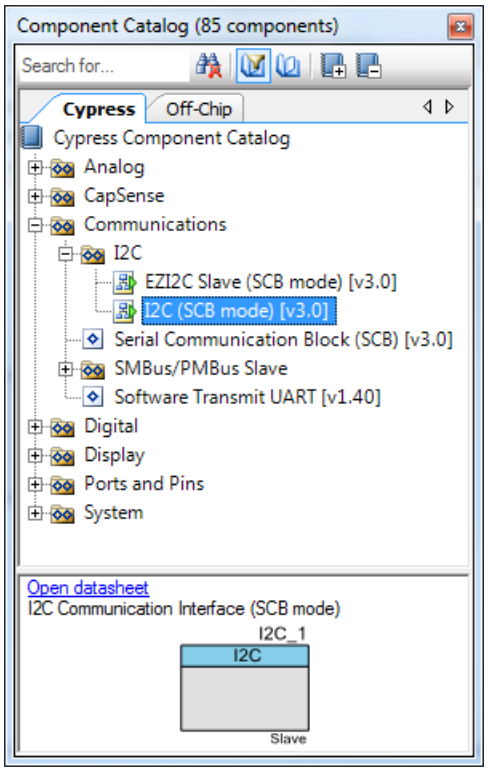
1. 在 PSoC Creator 中创建面向 PSoC 4 的新项目，如图 6-1 所示。

图 6-1. 在 PSoC Creator 中创建新项目



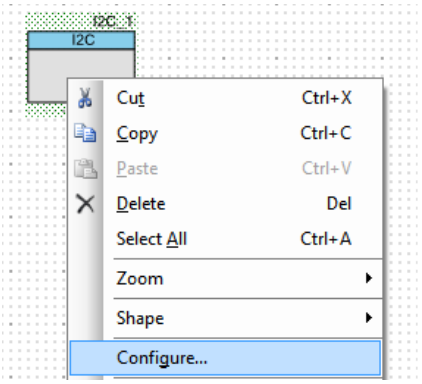
2. 将一个 I²C 组件拖放到顶层设计上，如图 6-2 所示。

图 6-2. 组件目录中的 I²C 组件




3. 要想配置 I²C 组件，请双击该组件或右键点击它并选择 **Configure**，如图 6-3 所示。

图 6-3. 打开 I²C 配置窗口



4. 根据这些设置进行配置 I²C，然后点击 **OK**，如图 6-4 和图 6-5 所示。

图 6-4. I²C “Configuration” 选项卡

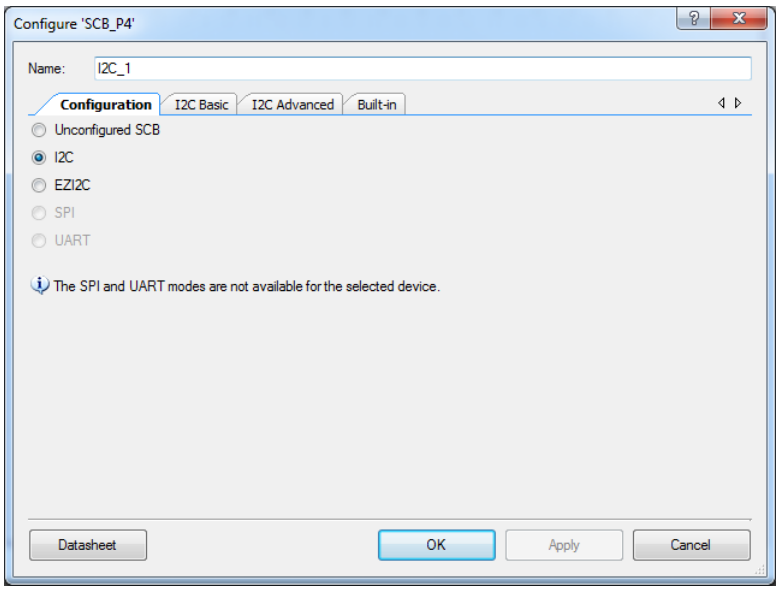
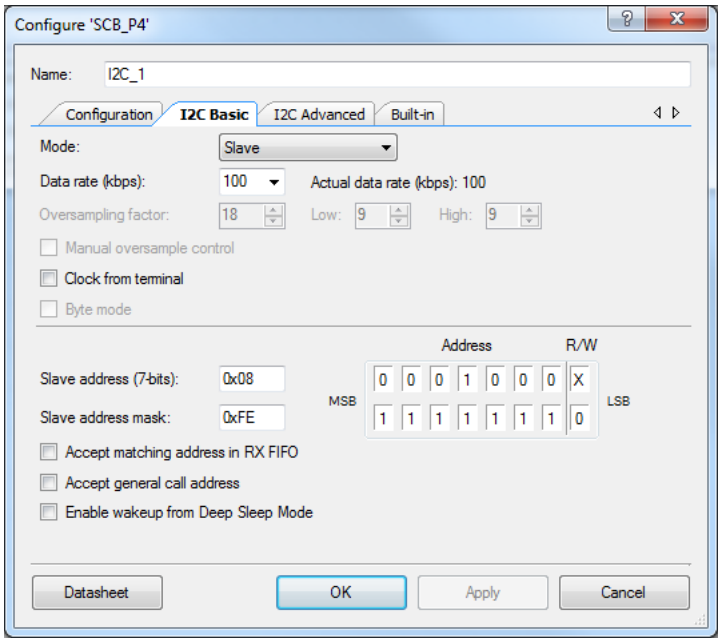
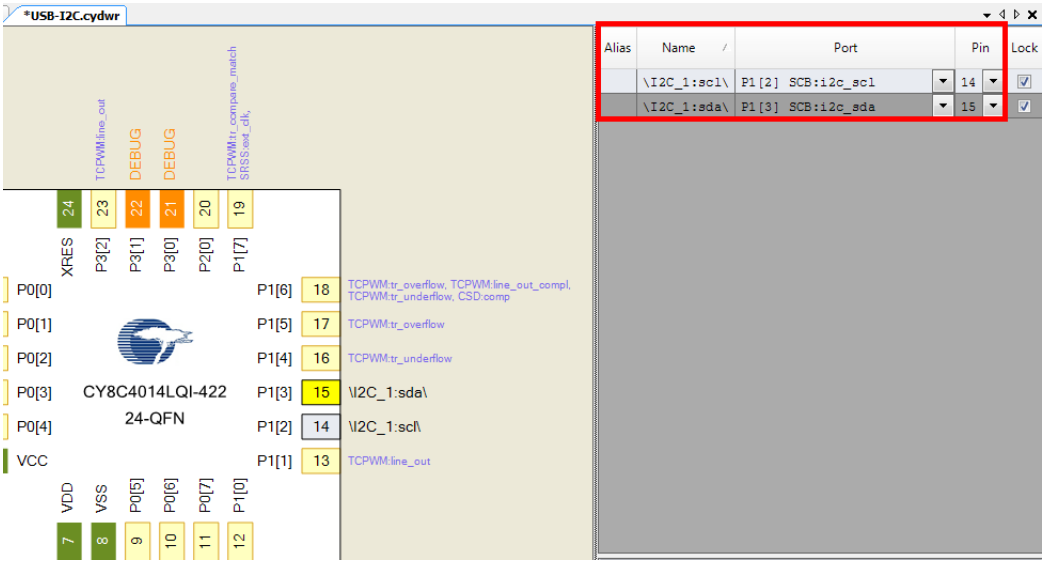



图 6-5. I²C 选项卡



5. 在 <project.cydwr> 的引脚选项卡下为 I²C SCL 选择 P1[2] 引脚，并为 I²C SDA 选择 P1[3] 引脚，如图 6-6 所示。

图 6-6. 引脚选择

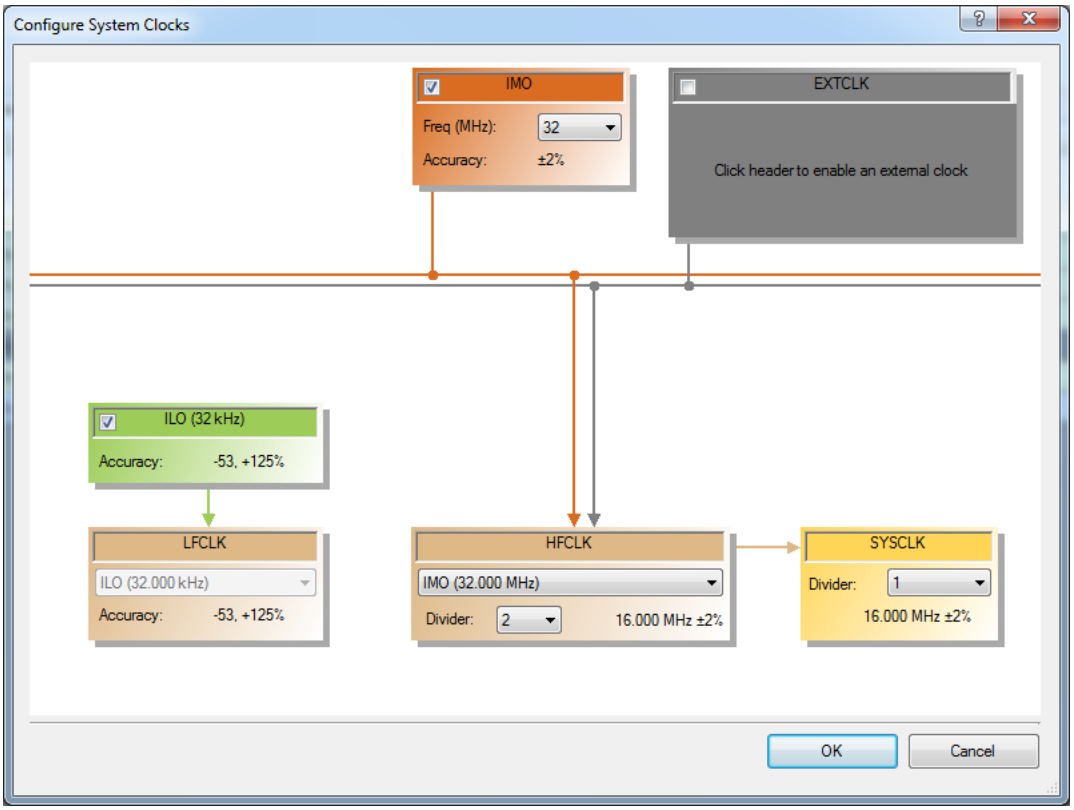



6. 将（本 PDF 文档附带的）*USB_I2C-main.c* 文件内的有效代码放置在 *main.c* 项目文件内。该代码允许在 PSoC 4 器件与 BCP 应用之间发送和接收 I²C 数据。

7. 通过依次点击 **Build > Build Project** 或按组合键 **[Shift] [F6]**，可以构建项目。构建项目无误或没有发生警告后，通过 PSoC 5LP 编程器或 MiniProg3，将该代码编程（通过按组合键 **[Ctrl] [F5]** 实现）到 PSoC 4 内。

   **注意：**I²C 的输入时钟上会显示发生一个警告。这是因为要想生成一个速率为 100 kbps 的 I²C 时钟，该模块需要一个频率为 1.6 MHz 的信号（频率为 12 MHz 的默认 HFCLK 设置无法派生该信号）。要想移除该警报，请依次点击 *<project_name>.cydwr* > **Clocks** 并双击 **HFCLK**。在出现的窗口内，将 **IMO** 设置为 **32 MHz**，并将 **HFCLK 驱动器**设置为 **'2'**（请参考图 6-7）。这样会生成一个频率为 16 MHz 的 HFCLK；再通过 10 分频，可生成 I²C 模块所需要的 1.6 MHz 时钟。

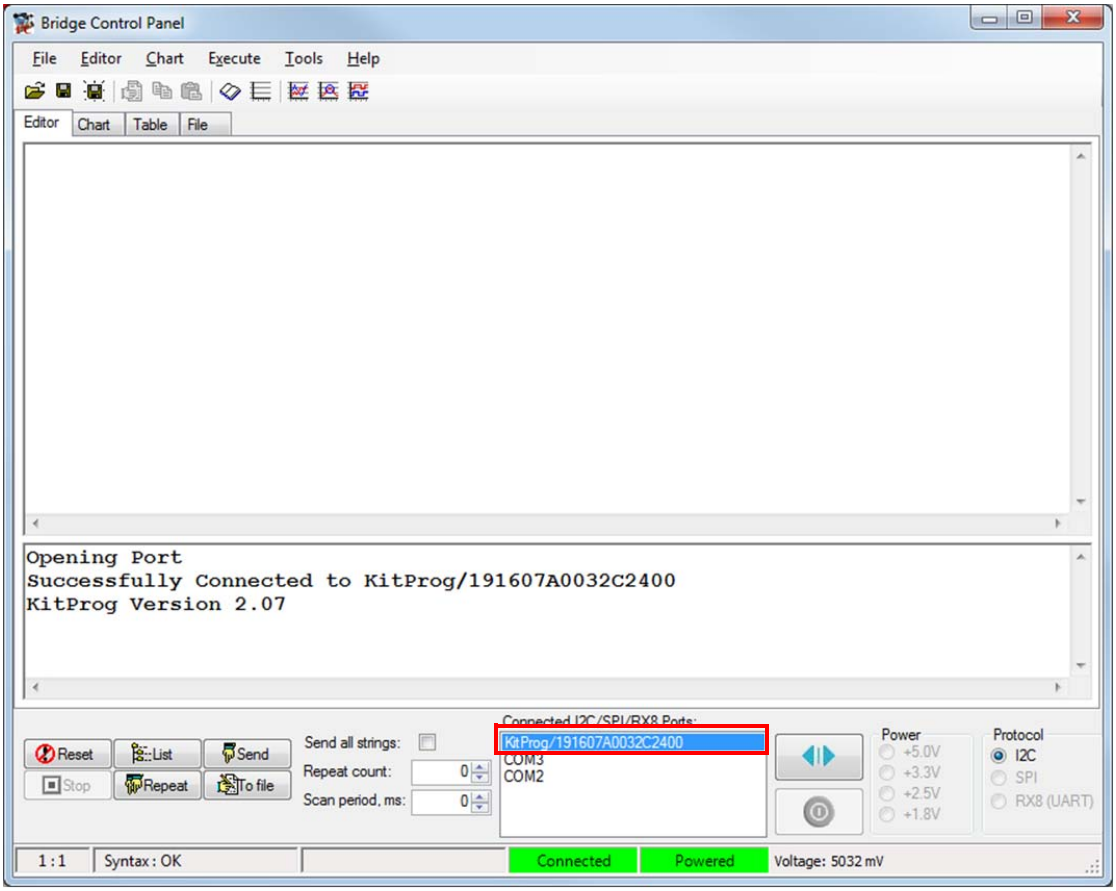
图 6-7. cydwr 文件中的时钟设置




8. 通过依次选择 **Start > All Programs > Cypress > Bridge Control Panel <version number>** 以打开 BCP。

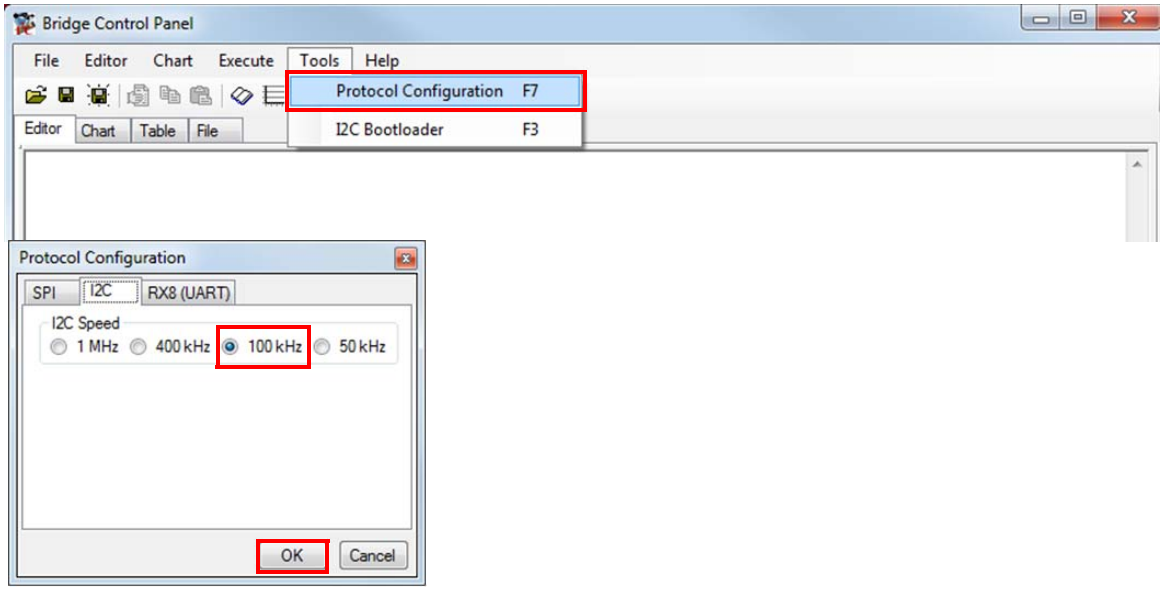
9. 连接到 **Connected I²C/SPI/RX8 Ports** 下的 **KitProg/**，如图 6-8 所示。

图 6-8. 连接到 BCP 中的 KitProg/



10. 从 **Tools** 菜单中打开 **Protocol configuration**，然后选择相应的 **I²C Speed**。请确保该 I²C 速度与 I²C 组件中所配置的速度相同。点击 **OK** 关闭该窗口，如图 6-9 中所示。

图 6-9. 打开 BCP 中的协议配置窗口




11. 要想传输数据，需要输入出现在图 6-10 内的指令，并按下 **[Enter]** 或点击 BCP 中的 **Send** 按键。记录显示了此传输操作是否成功。每个字节后显示的 “+” 表示操作已成功，而 “–” 指出操作失败，如图 6-10 和图 6-11 中所示。

图 6-10. 在 BCP 中输入指令

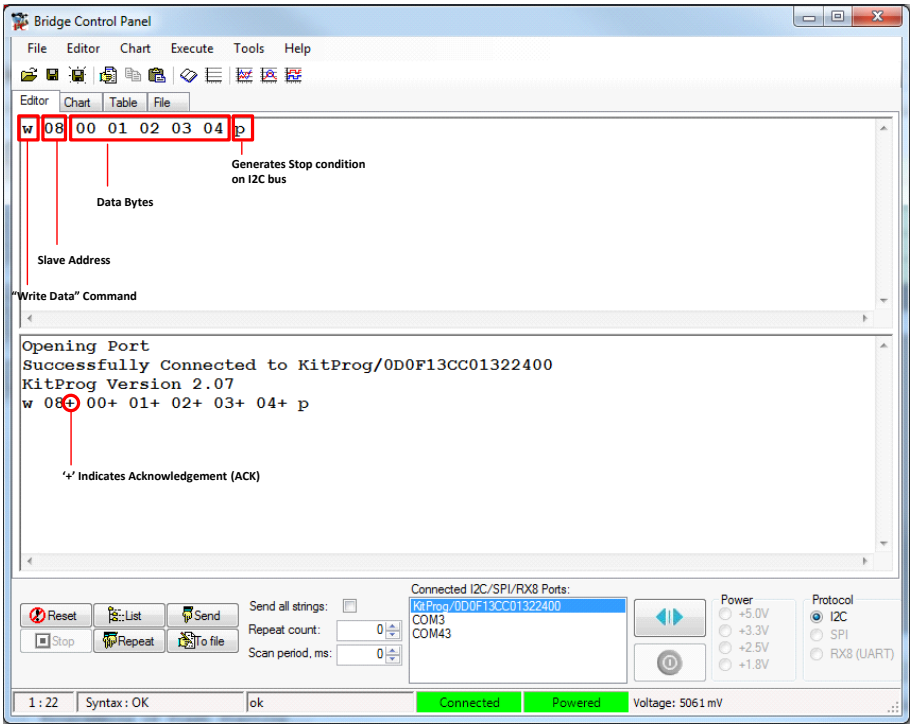
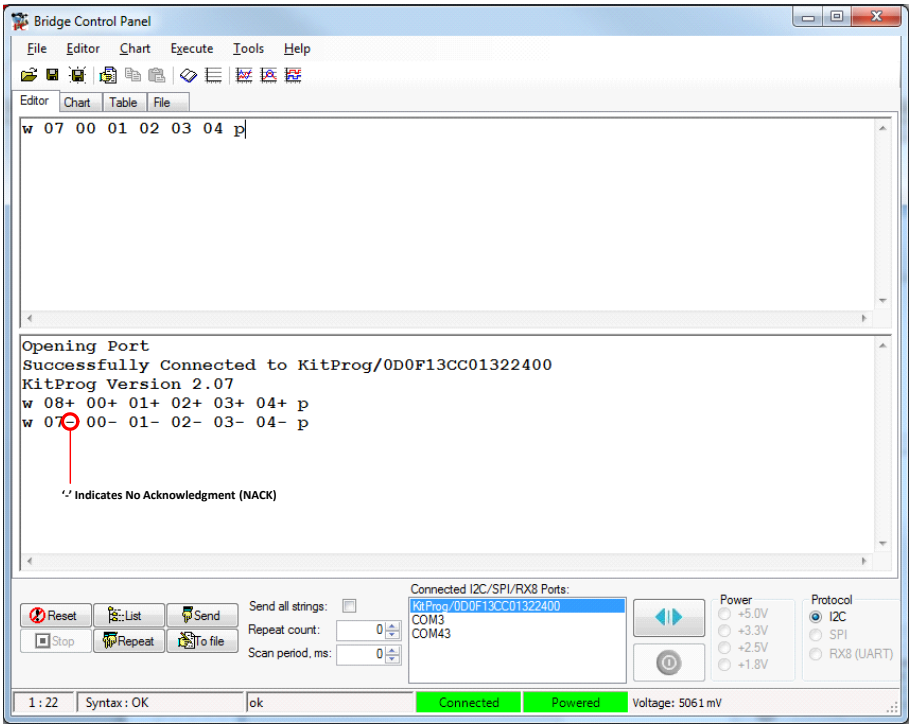
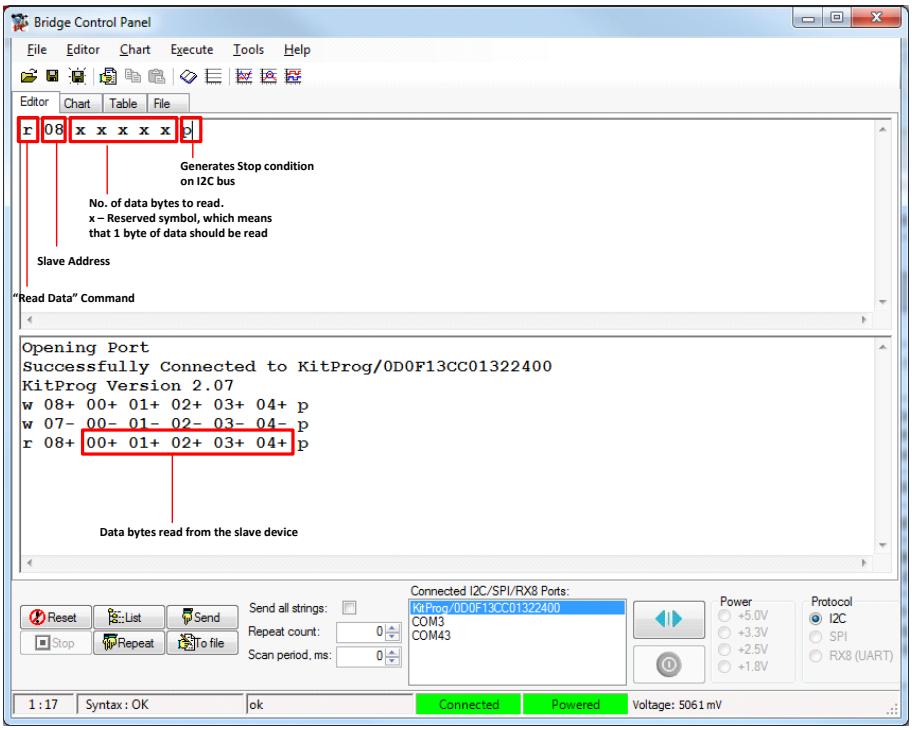


图 6-11. BCP 中的 NACK 指示




12. 从 BCP 读取来自 I²C 从设备的五个数据字节 （从设备地址为 0x08），如图 6-12 中所示。记录显示了此传输操作是否成功。

图 6-12. 从 BCP 中读取数据字节




**注意：**请参见BCP中**Help**（帮助）下的**Help Contents**（帮助内容）或按下**[F1]**，了解I²C指令的细节。

## 6.2 使用 FM24W256 F-RAM

PSoC 4000 Pioneer 套件拥有一个板上铁电存储体 （F-RAM）芯片，它可存储 32 KB 的数据。赛普拉斯 F-RAM 产品将 ROM 的非易失性数据存储功能与 RAM 的优势相结合（这些优势包括数量惊人的读和写周期、高速度的读和写周期以及低功耗）。对要求数据高完整性和超低功耗的应用来说，F-RAM 内核存储器和集成产品是理想的选择。这些产品的目标为汽车、工业级、使能技术以及网络等市场。 F-RAM 本身具有的特性包括高耐久性、快速单周期、对称的读 / 写速度、低功耗、伽马射线耐受性以及抗电磁噪声能力。

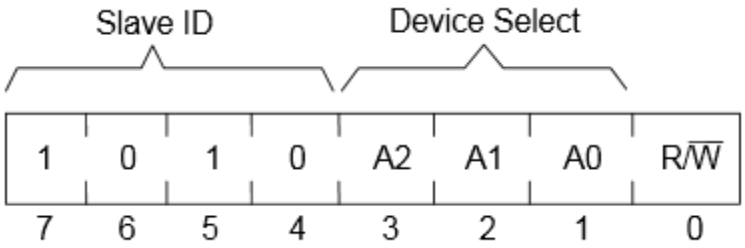
F-RAM 芯片为数据访问提供了 $I^2C$ 通信接口。它被硬连接到 PSoC 4000 $I^2C$ 线 （P1_2 和 P1_3）；同时它还连接到 PSoC 5LP $I^2C$。由于 F-RAM 器件是一个 $I^2C$ 从设备，因此同一总线上的各种 $I^2C$ 主设备可以共享该器件或对其进行访问。更多有关 F-RAM 器件的信息，请参考器件数据手册中的内容。

本节介绍了一个简单的示例，用于说明如何设置 F-RAM 器件的地址，将其用于 $I^2C$ 主（PSoC 4000）设备，并通过 PSoC 5LP USB-$I^2C$ 桥接器与 BCP 共享同一个 RAM。

### 6.2.1 地址选择

F-RAM 器件包括两个部分：从设备 ID 和器件选择，如图 6-13 中所示。从设备 ID 是特殊 F-RAM 器件数据手册中介绍的 F-RAM 系列产品 ID。对于 CY8CKIT-040 （FM24W256-G）所使用的器件，该从设备 ID 为 1010b。通过使用器件中的三个物理引脚 A2-A0 可设置器件选择位。在 CY8CKIT-040 中，这三个引脚的设置由电阻 R19/R18 (A0)、R17/R16 (A1) 和 R15/R14 (A2) 控制。更多有关信息，请参考第 46 页上的赛普拉斯铁电 RAM （F-RAM）。

图 6-13. F-RAM $I^2C$ 地址字节结构



## 6.2.2 读 / 写操作

器件的数据手册详细介绍了如何使用 F-RAM 来执行写 / 读操作。图 6-14 和图 6-15 提供了写 / 读数据包结构的简介作为参考。

图 6-14. F-RAM 单 / 多字节的写数据包结构

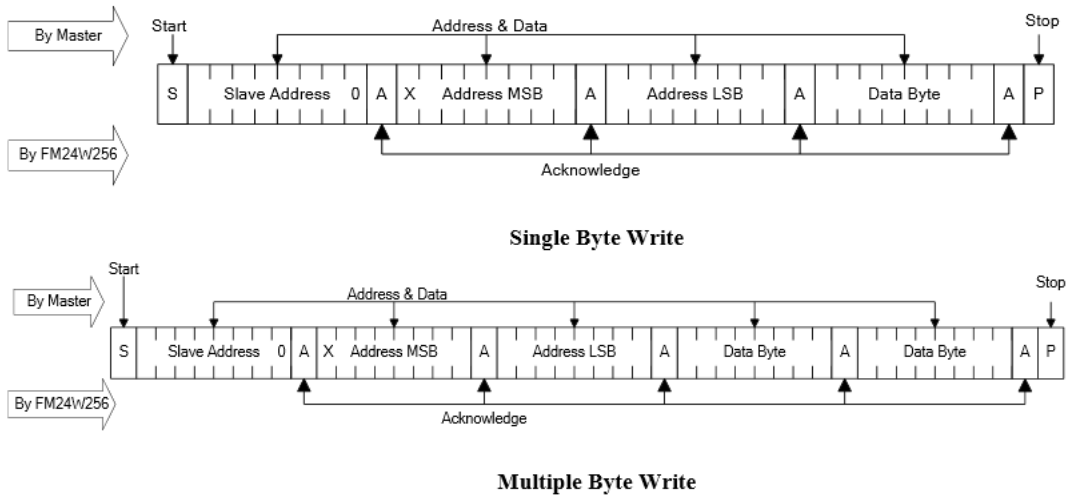
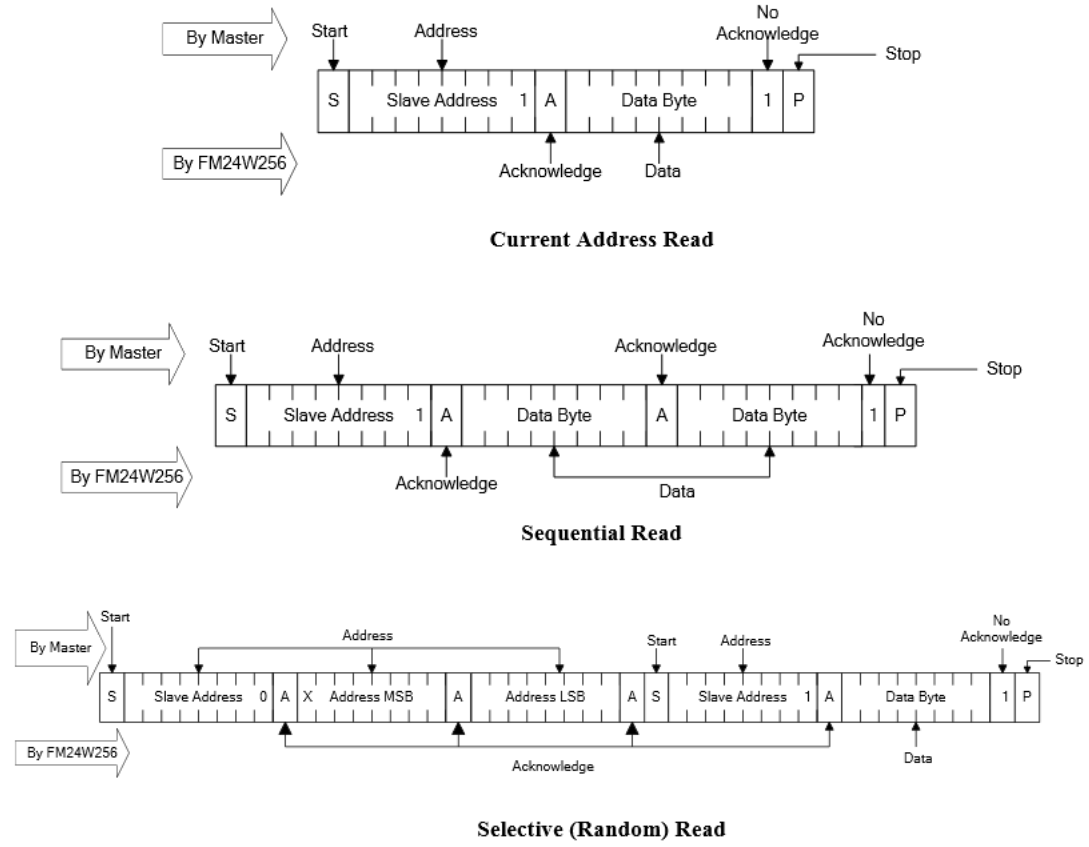



图 6-15. F-RAM 单 / 多字节的读数据包结构



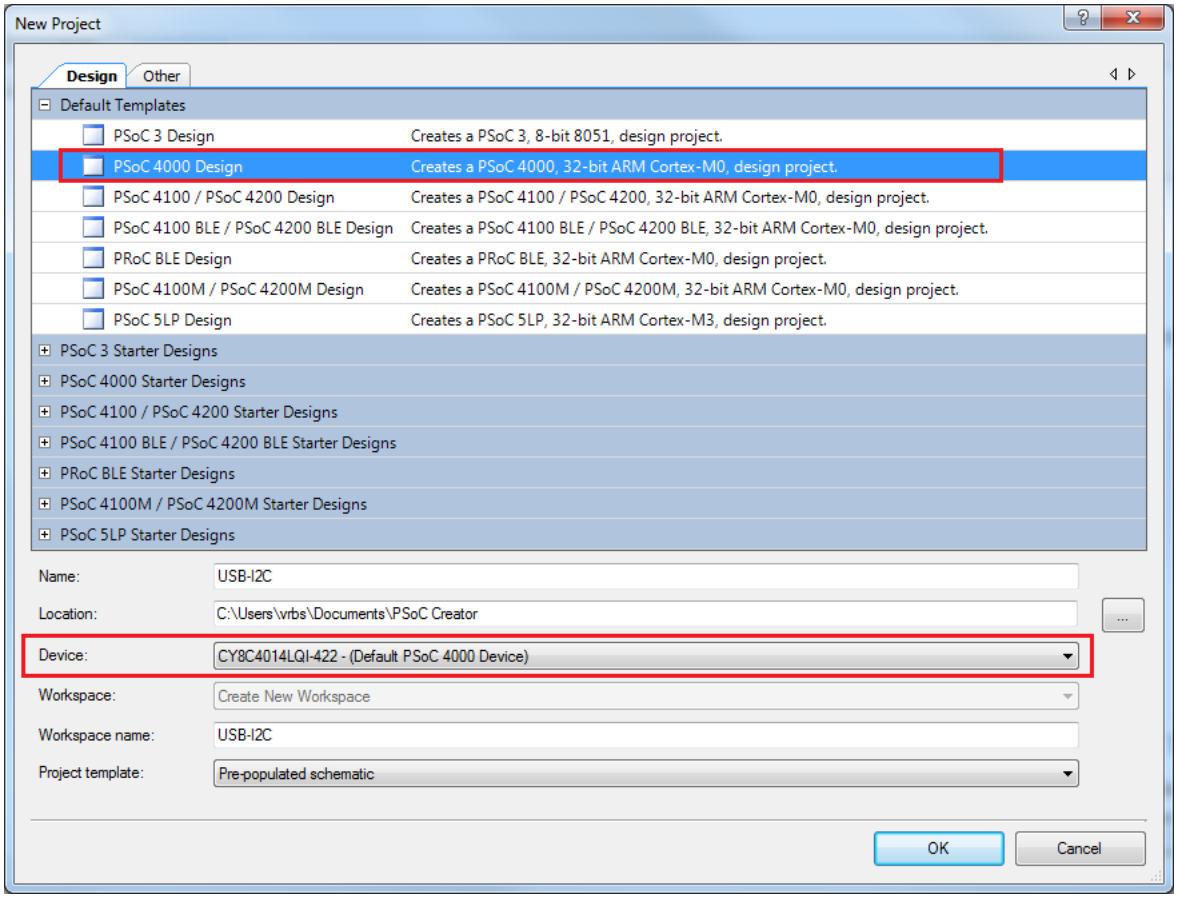
在上图中，“ 从设备地址 ” 表示 F-RAM 从设备的地址。在 F-RAM 中， [ 地址 MSB：地址 LSB] 的形式组成需要访问的存储器位置的 16 位地址，从而执行读 / 写操作。在写操作期间，从设备地址字节后面的两个字节组成的是将要访问的初始存储器地址。从该地址算起，每个访问字节 （读 / 写）将使该地址递增 1，并且该计数将在边界范围内进行翻转 （对于 32 KB 器件，该地址范围为 0x7FFF 到 0x0000）。通过对所需要的存储器地址进行写操作可以随时复位该值。

### 6.2.3 示例固件

下列各步骤说明了如何创建 PSoC 4000 系列来使用板上的 F-RAM，并且与其他 I²C 主设备共享该器件 （BCP 到 PSoC 5LP USB-I²C 桥接器）。

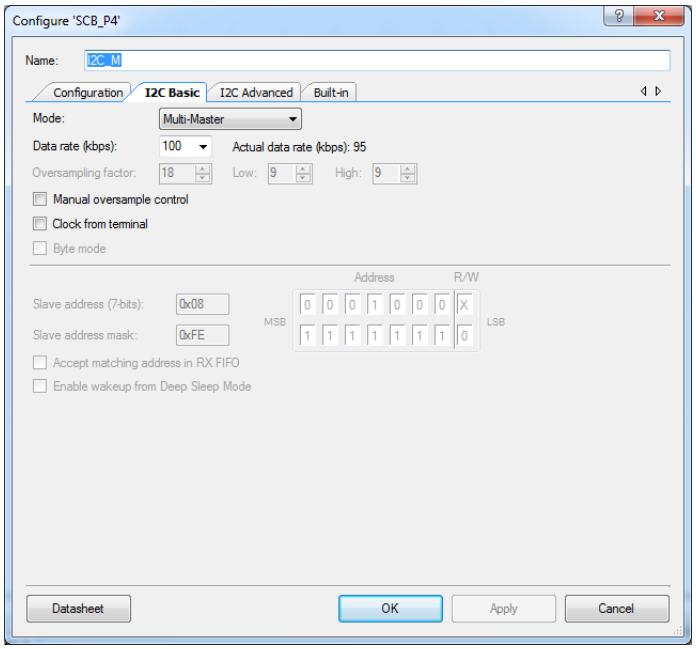
1. 打开 PSoC Creator 3.2 SP1 并创建新的 PSoC 4000 项目，如图 6-16 中所示。

图 6-16. 在 PSoC Creator 中创建新的 PSoC 4000 项目。



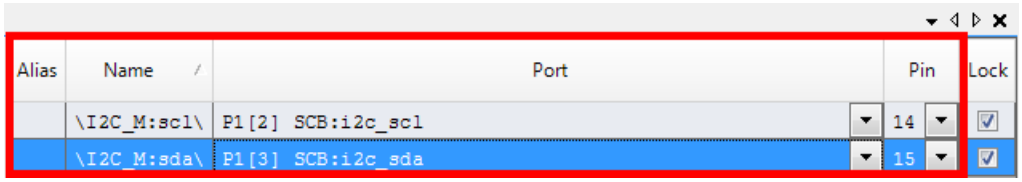
2. 在 **Components Catalog**（组件目录）中，将 **I²C （SCB 模式）**组件放置在 **Communication > I²C > I²C (SCB mode) [<version>]** 的顶层设计内，并通过使用出现在图 6-17 内的参数在 **Configuration**（配置）选项卡中将该组件配置为 I²C。

图 6-17. I²C 主设备的配置




3. 在 *.cydwr* 文件的 **Pins**（引脚）选项卡中，选择引脚 1[2] 和 1[3]，将其作为 I²C 引脚使用，如图 6-18 中所示。

图 6-18. 引脚选择

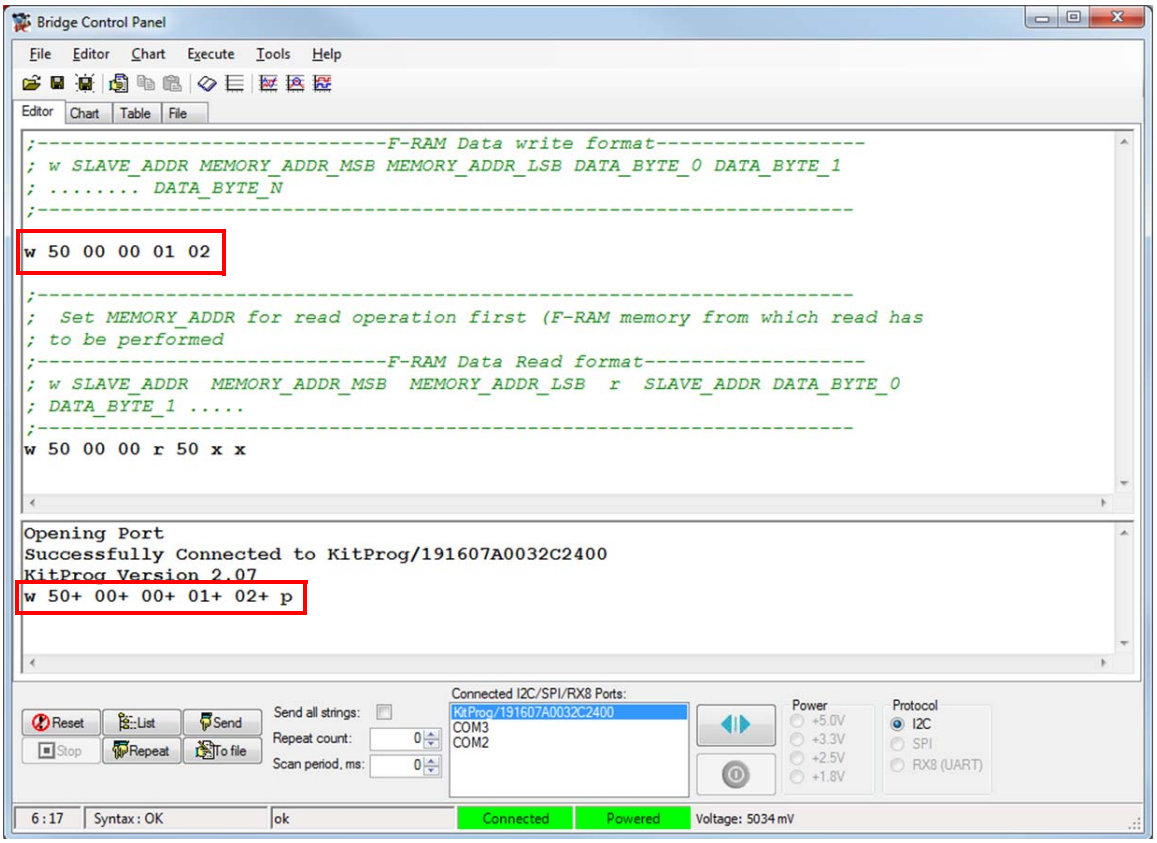



4. 将 （本 PDF 文档附带的） *Example_FRAM-main.c* 文件内的有效代码放置在 *main.c* 文件。
5. 通过依次点击 **Build > Build Project** 或按组合键 **[Shift] F6]**，可以构建项目。如果项目编译没有发生任何错误或警告，按组合键 **[Ctrl]+ [F5]** 可以通过套件中的 MiniProg3 或 PSoC 5LP 编程器编程该器件。

   **注意：**I²C 输入时钟上会显示一个警告。这是因为要想生成一个速率为 100 kbps 的 I²C 时钟，该模块需要一个频率为 1.6 MHz 的信号 （频率为 12 MHz 的默认 HFCLK 设置无法派生该信号）。要想清除此警告，依次点击 *<project_name>.cydwr* > **Clocks** 并双击 **HFCLK**。在出现的窗口内，将 **IMO** 设置为 **32 MHz** 以及 **HFCLK 驱动器**设置为 ‘**2**’ （参考图 6-7）。这样将生成一个频率为 16 MHz 的 HFCLK ；通过 10 分频，可以生成 I²C 模块所需要的 1.6 MHz 时钟。
6. 打开 BCP 并配置 I²C 协议，如第 104 页上的将 PSoC 5LP 作为 USB-I2C 桥接器使用 中的定义。

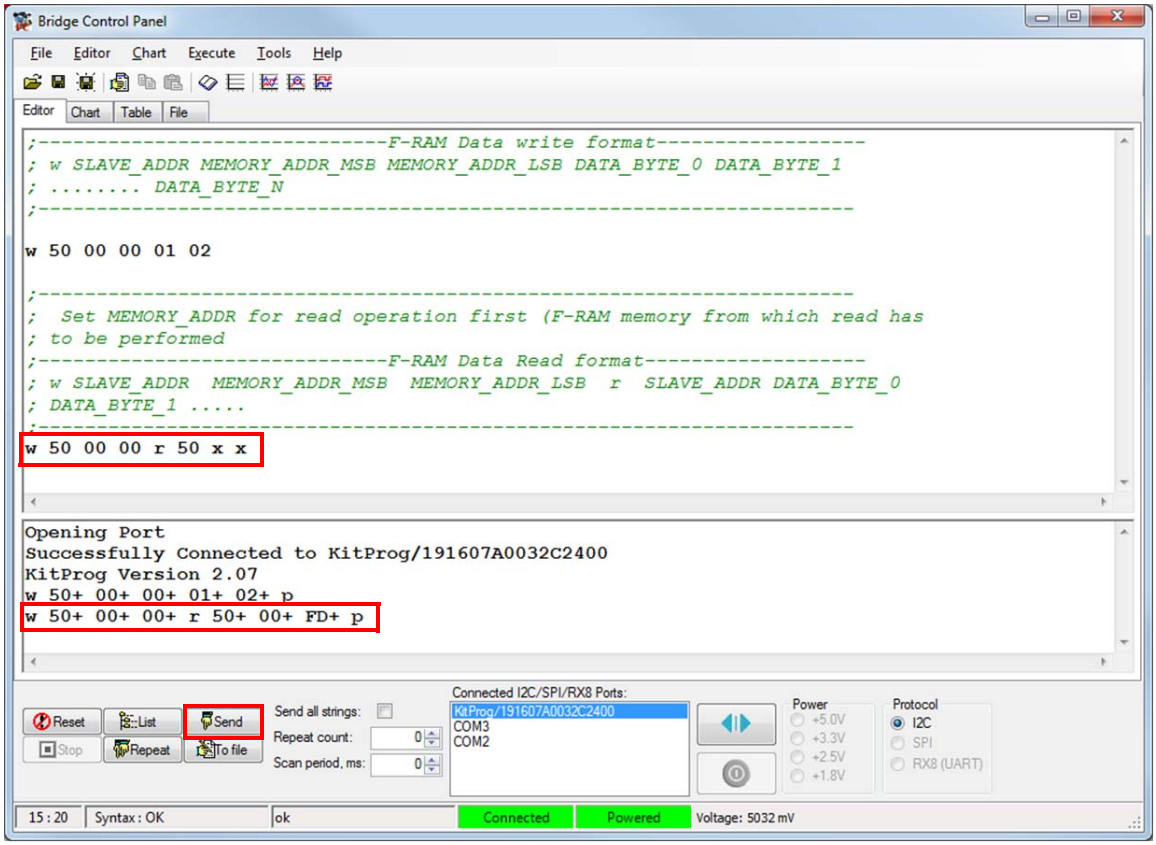
7. 在指令窗口中，复制并粘贴本文档中 *F-RAM_BCP_Commands.txt* 文件内附带的代码。
8. 默认情况下， F-RAM 器件的从设备地址被配置为 0x50。如果已经修改了该值 （如第 112 页上的地址选择 中所述），那么需要修改指令窗口中的从设备地址 （使用十六进制格式的从设备地址替换 “50”）。
9. 通过发送 ‘W 50 00 00 01 02’ 线对 F-RAM 器件进行写操作，从而将 0x01 值写入到存储器地址 0x0000 上，将 0x02 写入到存储器地址 0x0001 上。从设备应准确确认所发送的指令，以确保该传输准确无误，如图 6-19 中所示。

图 6-19. 将数据发送 / 写入到 F-RAM



10. 如果写操作成功，那么通过发送第二行指令向同一个地址传送读指令，从而检查该数据是否准确，如图 6-20 中所示。同一个地址上的读指令会在 0x0000 地址上产生 ‘0’ （标志被清除），并在 0x0001 地址上产生 0xFD （发送了 0x02 的反向值）。这样便表示板载 PSoC 4 器件已经访问了这些字节，并对它们进行了修改。如步骤 9 所述，将任何字节写入到 0x0001 时，如果 0x0000 字节被设置为 ‘1’，则会计算同一个字节的反向值，并且将该值存储到同一个地址内。

图 6-20. 从 F-RAM 读取数据




## 6.3 将 PSoC 5LP 作为 USB-UART 桥接器使用

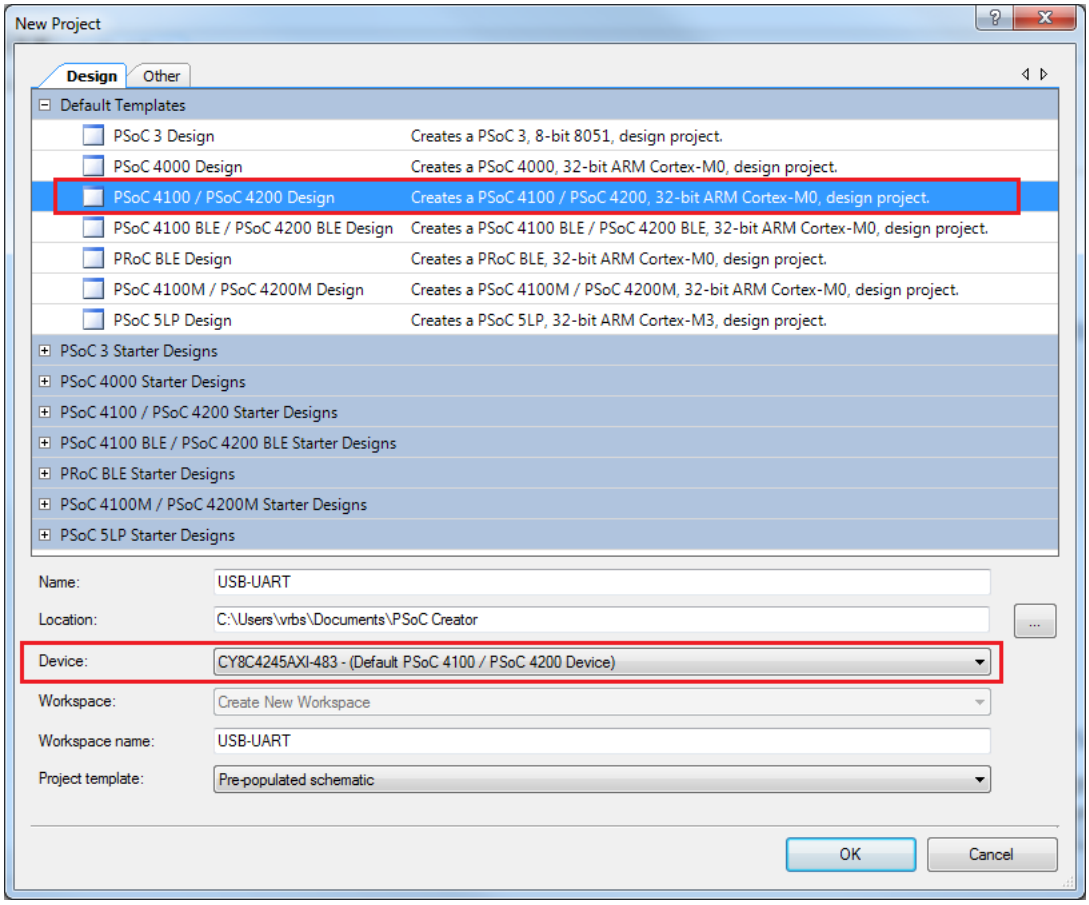
PSoC 5LP 用作为 USB-UART 桥接器可以和 COM 终端软件进行通信。本节介绍了一个示例，用以说明如何通过外部器件或电路板 （带有 CY8CKIT-001 的 PSoC 4200）使用 PSoC 5LP 的 USB-UART 桥接器。

**注意：**此项目说明了如何将 PSoC 5 LP 的 USB-UART 桥接器用于外部 UART 线。在将 USB-UART 桥接器用于电路板上的 PSoC 4000 系列时，可以参考第 57 页上的项目：CapSense 接近感应和 UART中介绍的内容。 PSoC 4000 系列仅支持软件 UART 传送线，如第 57 页上的项目：CapSense 接近感应和 UART 中所述。

如果用户使用的 Windows 操作系统不带超级终端，那么，可采用其它的终端软件，，如 PuTTY。

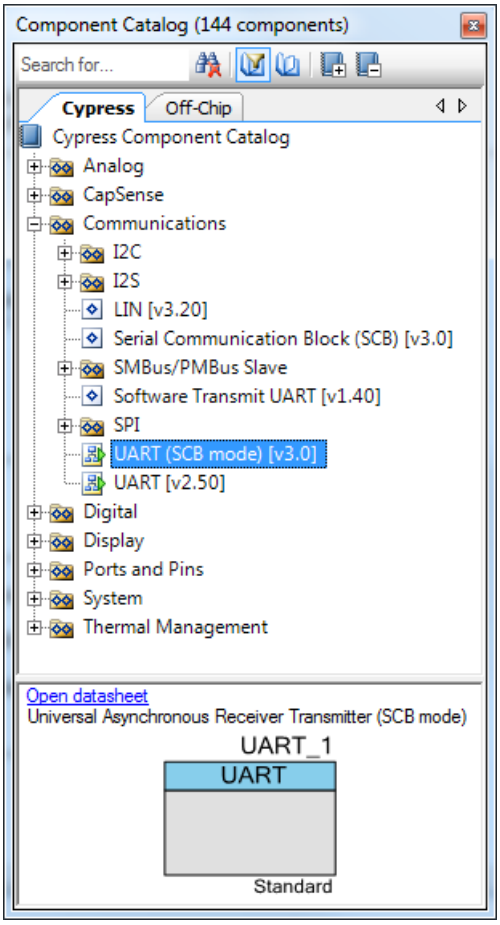
1. 在 PSoC Creator 中创建新的 PSoC 4 项目，如图 6-21 中所示。为您的项目选择合适的位置，并根据需要重新命名它。

图 6-21. 在 PSoC Creator 中创建新的项目。



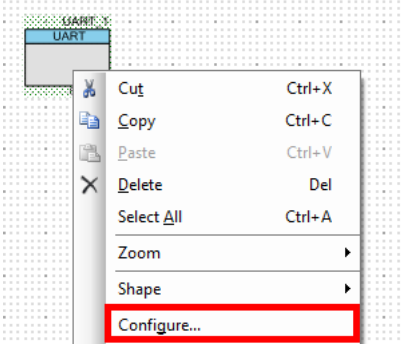
2. 将 **Component Catalog** （组件目录）中 （在图 6-22 中显示）的 UART （SCB）组件拖放到顶层设计上。

图 6-22. 组件目录下的 UART 组件。




3. 要配置 UART，请双击该组件或右键点击它并选择 **Configure**，如图 6-23 中所示。

图 6-23. 打开 UART Configuration （配置）选项表



4. 根据图 6-24、图 6-25 和图 6-26 配置 UART，然后点击 **OK**。

图 6-24. ‘ 配置 ’ 选项卡

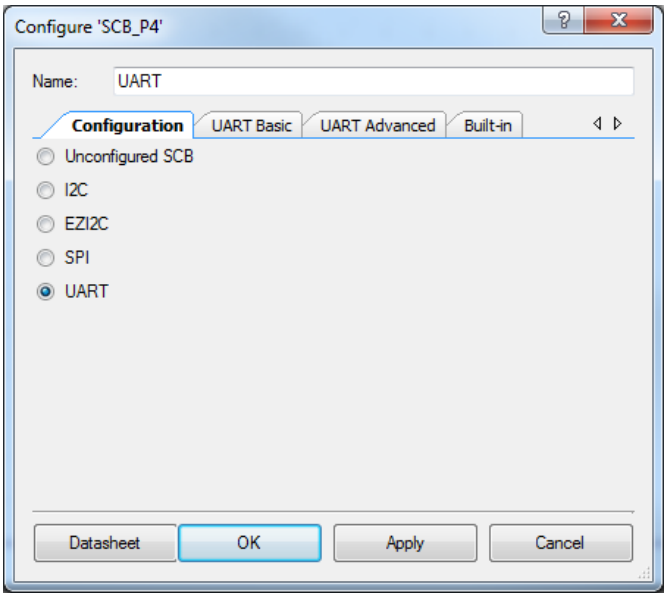



图 6-25. ‘UART 基本 ’ 选项卡

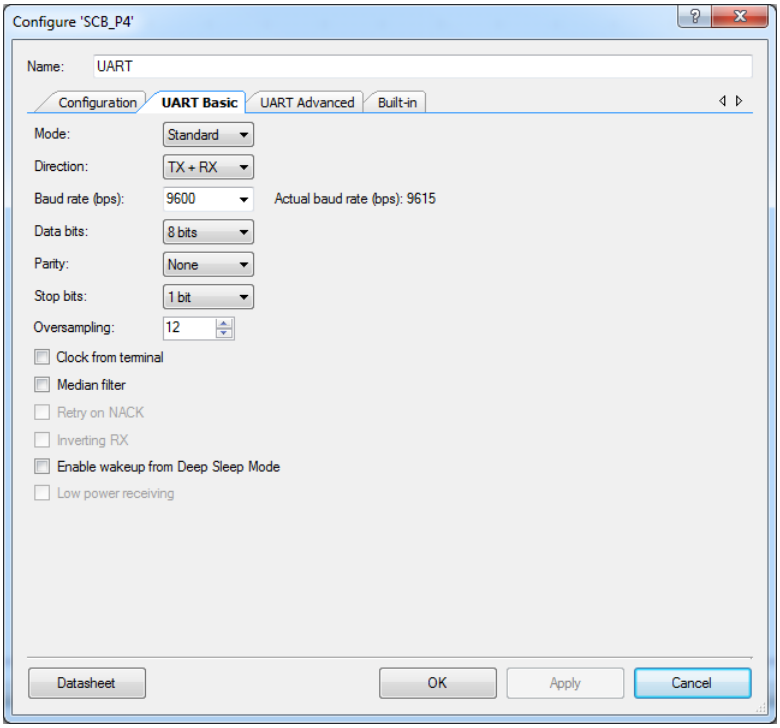
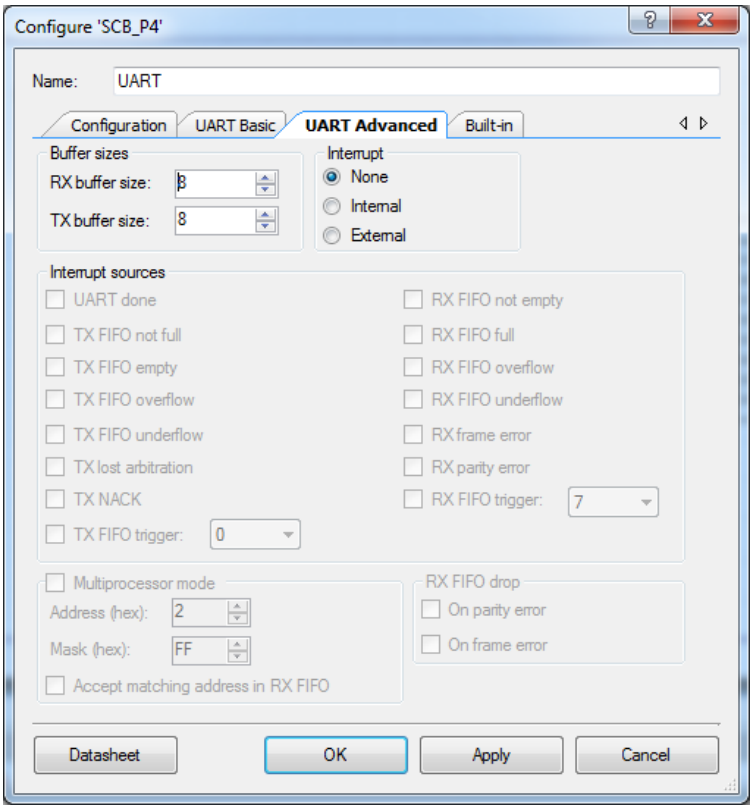
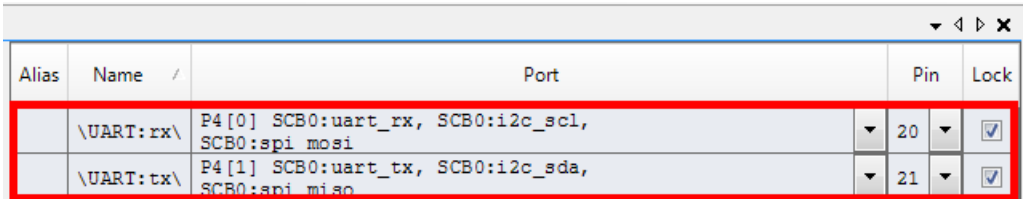


图 6-26. ‘UART 高级’ 选项卡




5. 在 *<Project_Name>.cydwr* 的 **Pins** （引脚）选项卡中，为 UART RX 选择 P4[0]，并为 UART TX 选择 P4[1]，如图 6-27 所示。

图 6-27. 引脚选择




6. 将（本 PDF 文档附带的）*USB_UART-main.c* 文件内的有效代码放置在 *main.c* 项目文件。代码将回送所接收的所有 UART 数据。
7. 通过依次点击 **Build > Build { 项目名称 }** 或按组合键 [Shift] [F6]，可以构建项目。项目建立无误且无警告后，通过 MiniProg3，（依次点击 **Debug > Program**）将该项目编程到 PSoC 4 内。

8. 将 PSoC 4 上的 RX 线连接到 J8_10，并将 PSoC 4 的 TX 线连接到 J8_9，如图 6-28 和图 6-29 所示。

**注意：**

- 连接 RX 线前，需要先移除 R57（该电阻将 PSoC 4000 器件的 P3[0] 引脚同 PSoC 5LP RX 线相连接）。这样可确保当外部桥接器使用 RX 线时，能够对 PSoC 4000 器件进行编程 / 调试。
- 使用USB-UART桥接器与外部UART接口相连时，可以参考所提供的CY8CKIT-001、CT8CKIT-038 和 CY8CKIT-040 的设置情况。

图 6-28. PSoC 4 与 PSoC 5LP 之间使用 UART 进行连接。

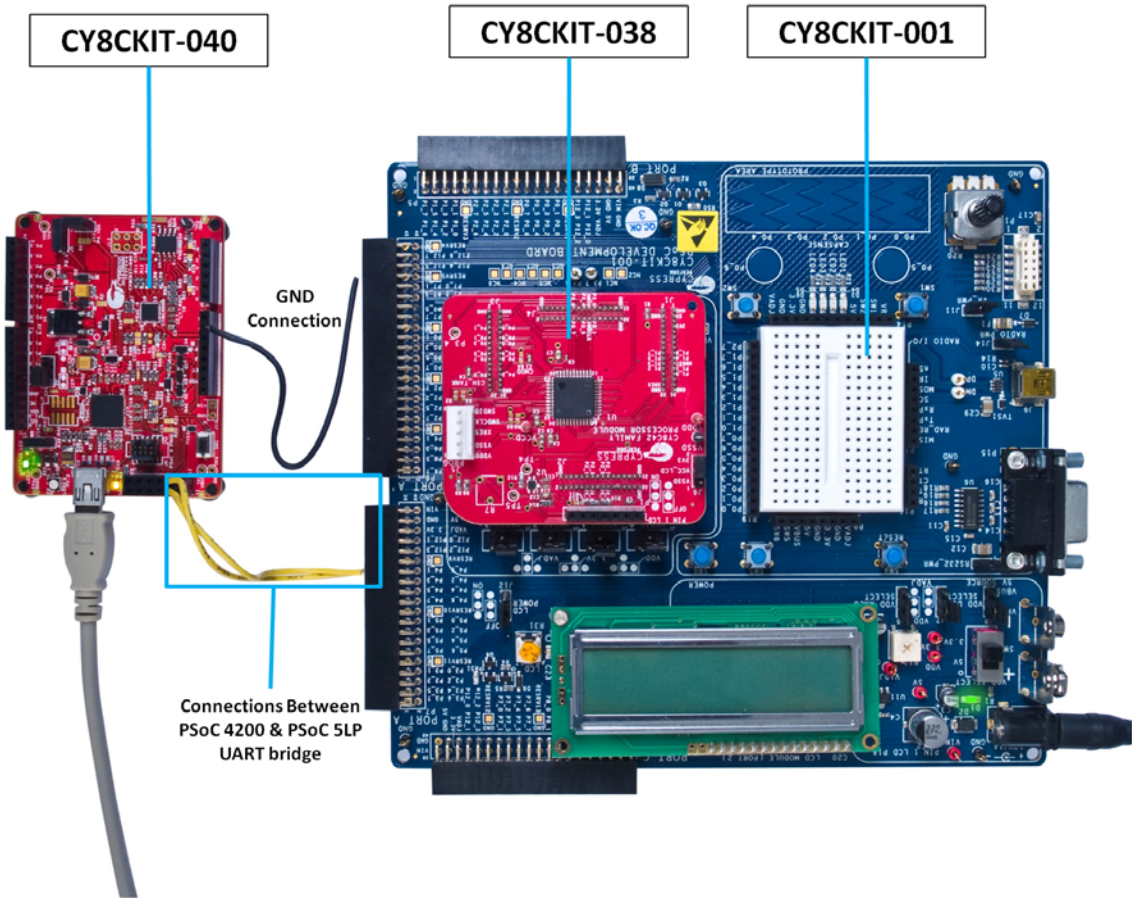
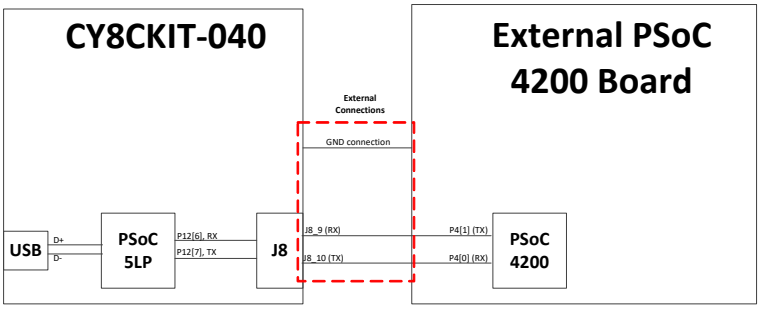



图 6-29. PSoC 4 与 PSoC 5LP 之间使用 UART 进行连接的框图。

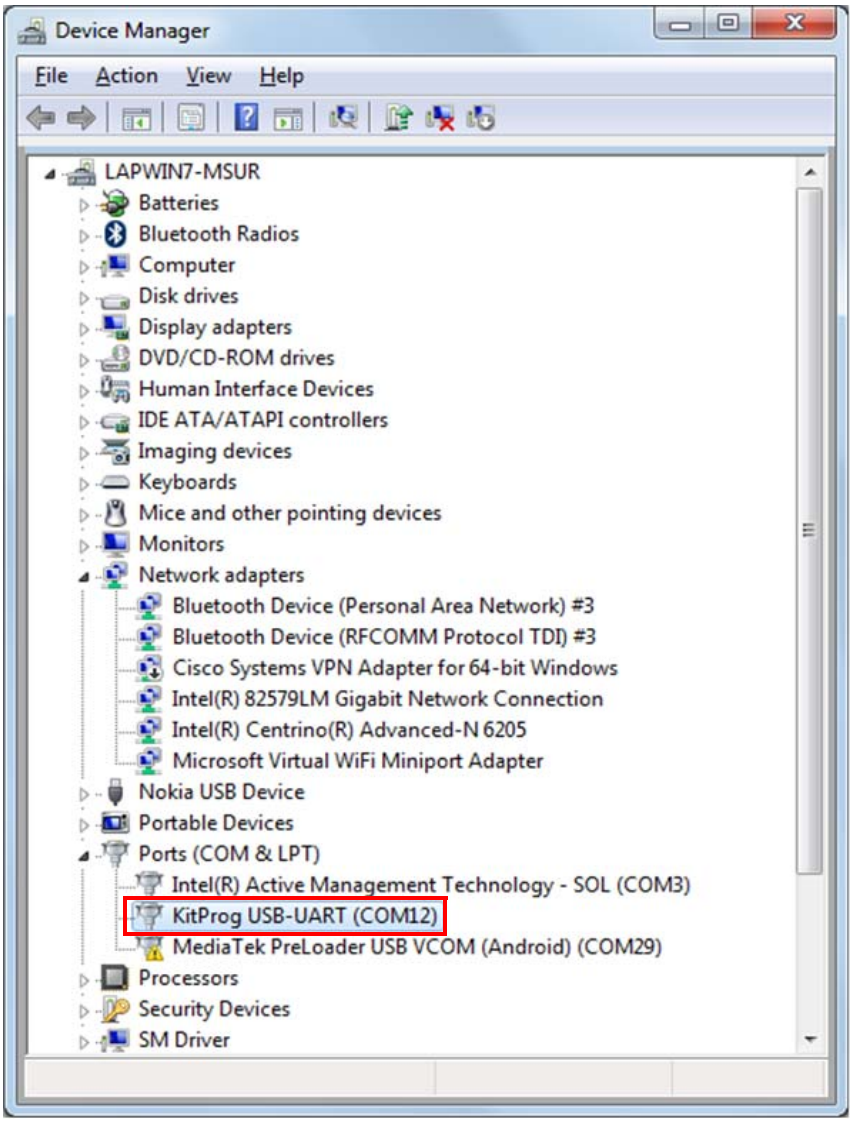


**注意：**根据 UART 组件的配置情况，可将 UART RX 和 UART TX 路由到 PSoC 4 上的任何数字引脚。通过 UART 的 SCB 执行，可将 RX 和 TX 引脚路由到下面任何一个子集：（P0[4]、P0[5]）、（P3[0]、P3[1]）或 （P4[0]、 P4[1]）。

要想从终端软件同 PSoC 4 通信，请按照以下流程进行操作：

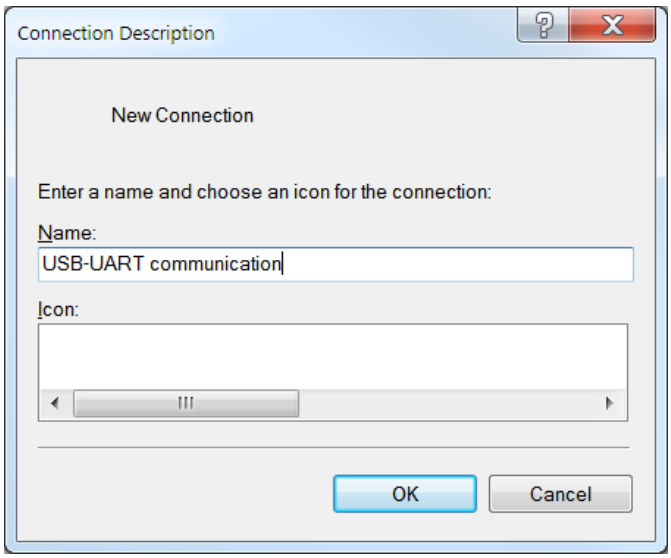
1. 将 USB Mini-B 连接至 J10。该套件将枚举为 KitProg USB-UART，并显示在 **Ports (COM & LPT)** 下的 **Device Manager** （器件管理程序）中。将一个通信端口分配给 KitProg USB-UART，如图 6-30 所示。

图 6-30. 器件管理程序中的 KitProg USB-UART



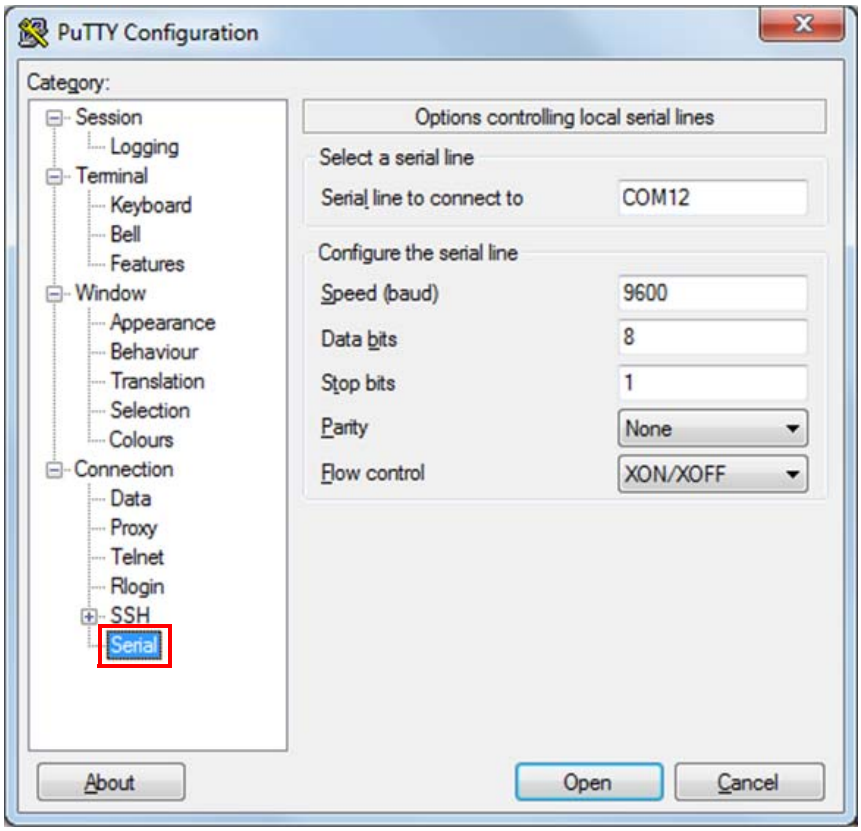
2. 打开超级终端并依次选择 **File > New Connection**，然后输入新连接的名称，最后点击 **OK**，如图 6-31 中所示。

图 6-31. 打开新的超级终端



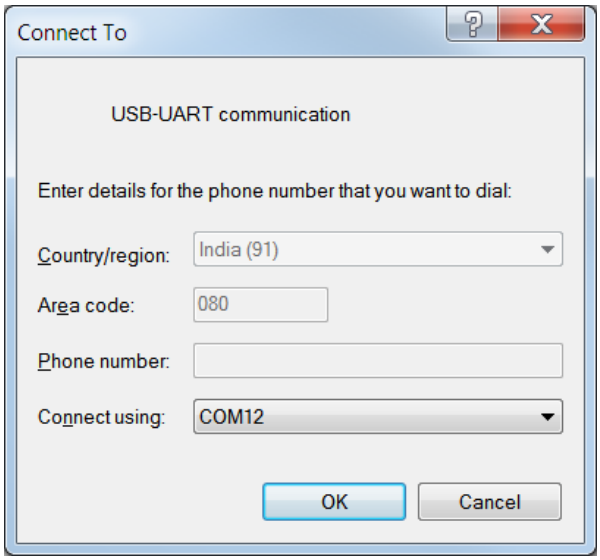

3. 对于 PuTTY，双击 PuTTY 图标并选择 **Connection** 下的 **Serial** 项，如图 6-32 所示。

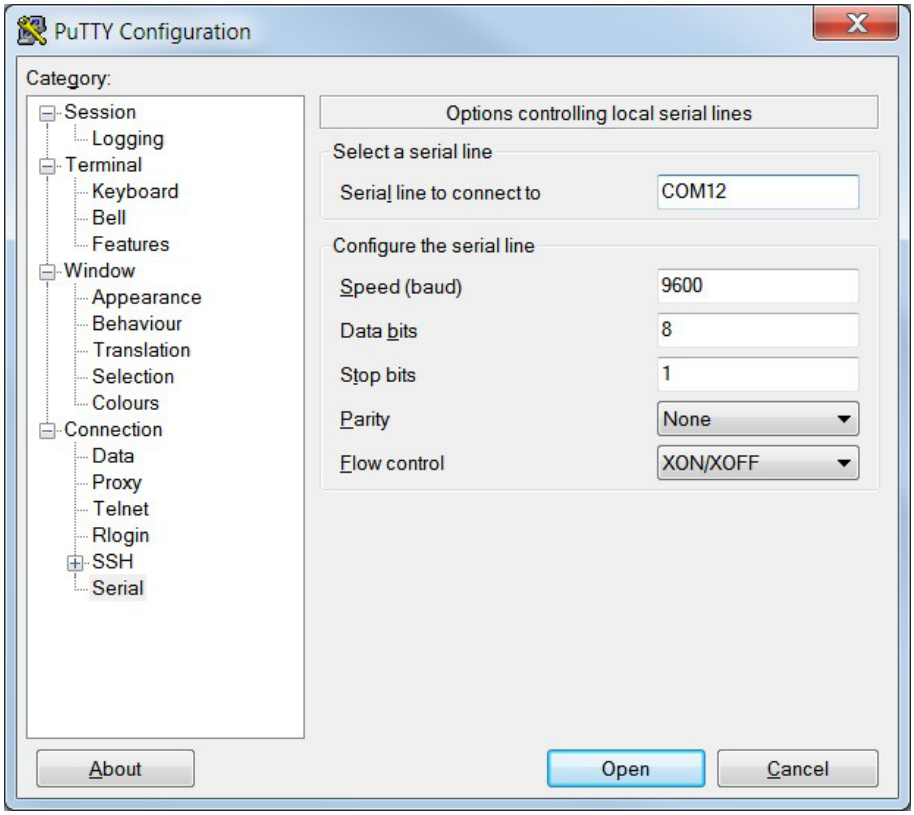
图 6-32. 在 PuTTY 中打开新连接



4. 一个新窗口打开，在这里可以选择通信端口，如图 6-33 所示。
   a. 在超级终端中，选择 **‘Connect using’** 的 COMx（或分配给 KitProg USB-UART 的特定通信端口），然后点击 **OK**。该代码示例使用了 **COM12**。
   b. 在 PuTTY 配置中 **Serial line to connect to** 项旁边的文本框内键入 COMx。该代码示例使用 **COM12**。

图 6-33. 选择通信端口 —— 超级终端和 PuTTY



5. 在超级终端中，在 **Port Settings** （端口设置）下选择 **Bits per second** （每秒位数）、 **Data bits** （数据位）、**Parity**（奇偶校验）、**Stop bits**（停止位）和 **Flow control** （流控），然后点击 **OK**，如图 6-34 中所示。请确保该设置与 PSoC 4 中所配置的 UART 设置相同。

   在 PuTTY 中，在 **Configure the serial line**（配置串行线）下（显示在图 6-33 的第二个图像内）选择 **Speed (baud)**（速度（波特））、**Data bits** （数据位）、**Stop bits** （停止位）、**Parity** （奇偶校验）和 **Flow control** （流量控制）。在 **Connection type** （连接类型）下点击 **Session** 并选择 **Serial** 项，如图 6-35 所示。 **Serial line** 显示的是所选定的通信端口 (COM12)， **Speed** 显示的是所选的波特率。点击 **Open** 以启动通信过程。

图 6-34. 在超级终端中配置通信端口

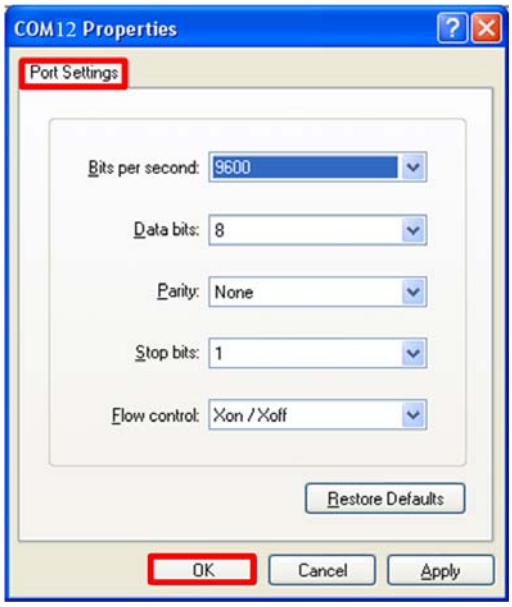
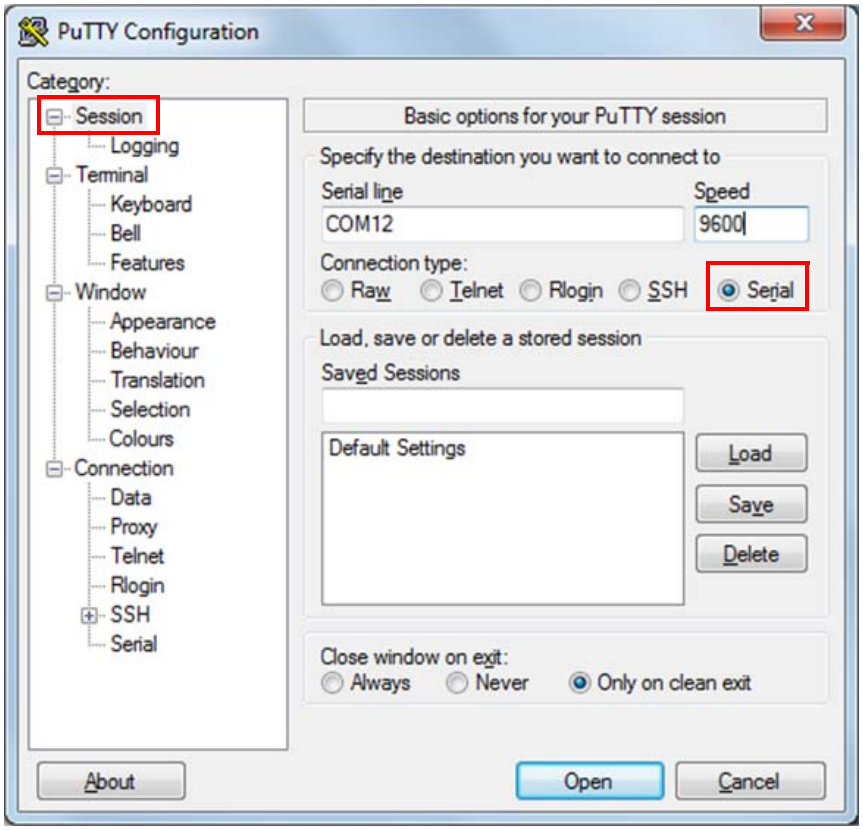


图 6-35. 在 PuTTY 中选择通信类型




6. 依次选择 **File > Properties > Settings > ASCII Setup** 后，通过使能 **Echo typed characters locally**，可以在超级终端中显示所输入的字符，如图 6-36 所示。在 PuTTY 中，使能 **Terminal > Line discipline** 选项下的 **Force on**，可显示在 PuTTY 中所输入的字符，如图 6-37 所示。

图 6-36. 使能在超级终端中所输入字符的回送

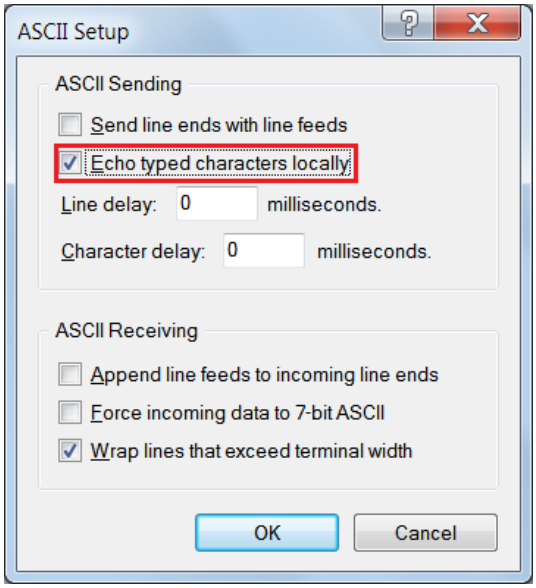
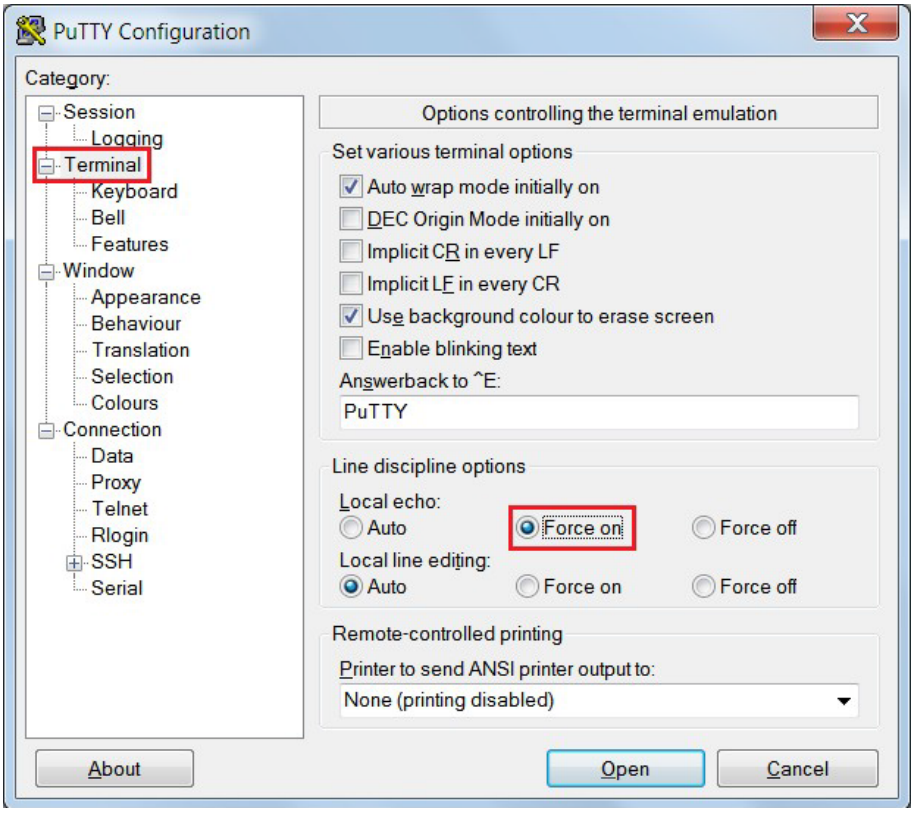


图 6-37. 使能在 PuTTY 中所输入字符的回送




7. COM 终端软件显示了来自 PSoC 4 中 UART 的键入数据和回送数据，如图 6-38 和图 6-39 所示。

图 6-38. 超级终端上所显示的数据

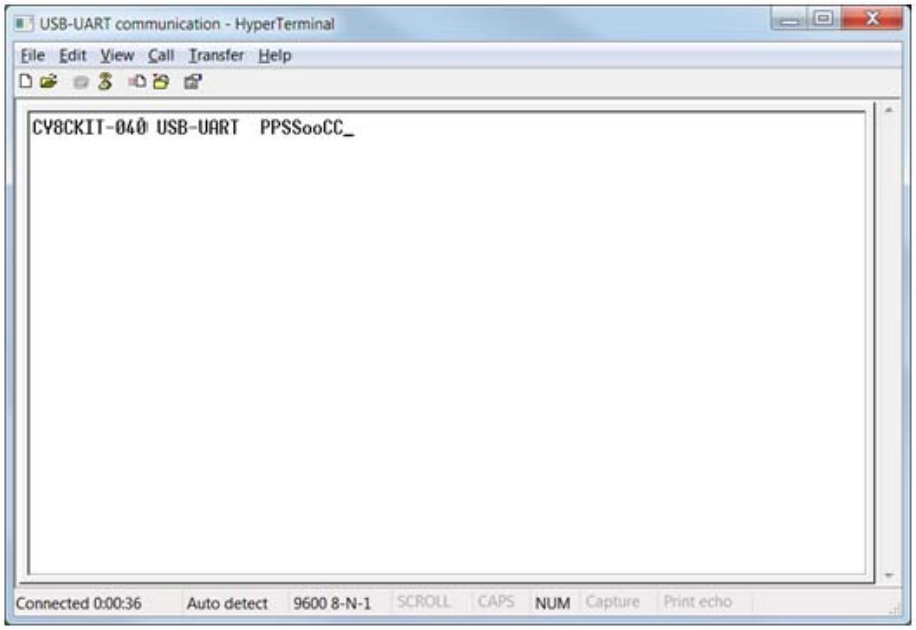
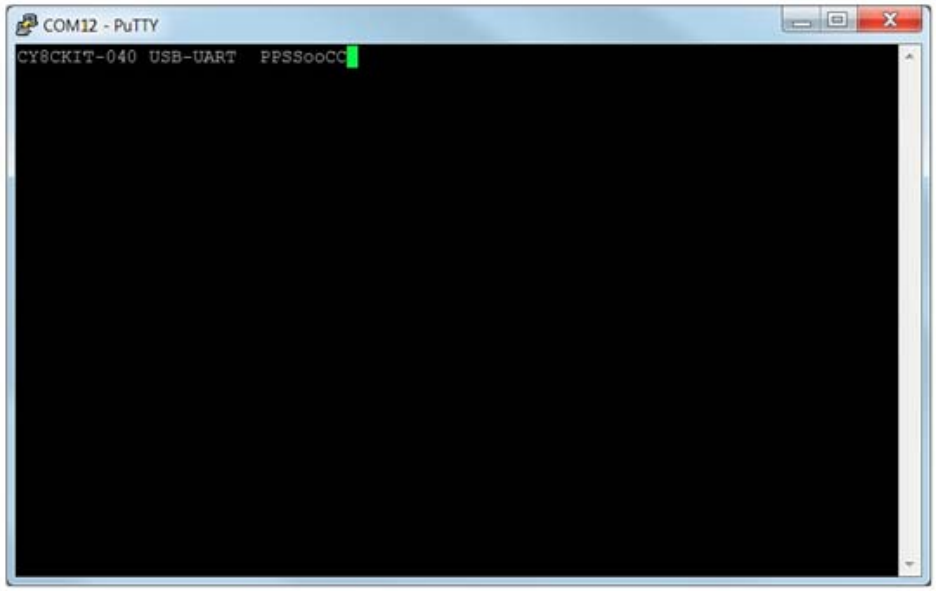


图 6-39. PuTTY 上所显示的数据



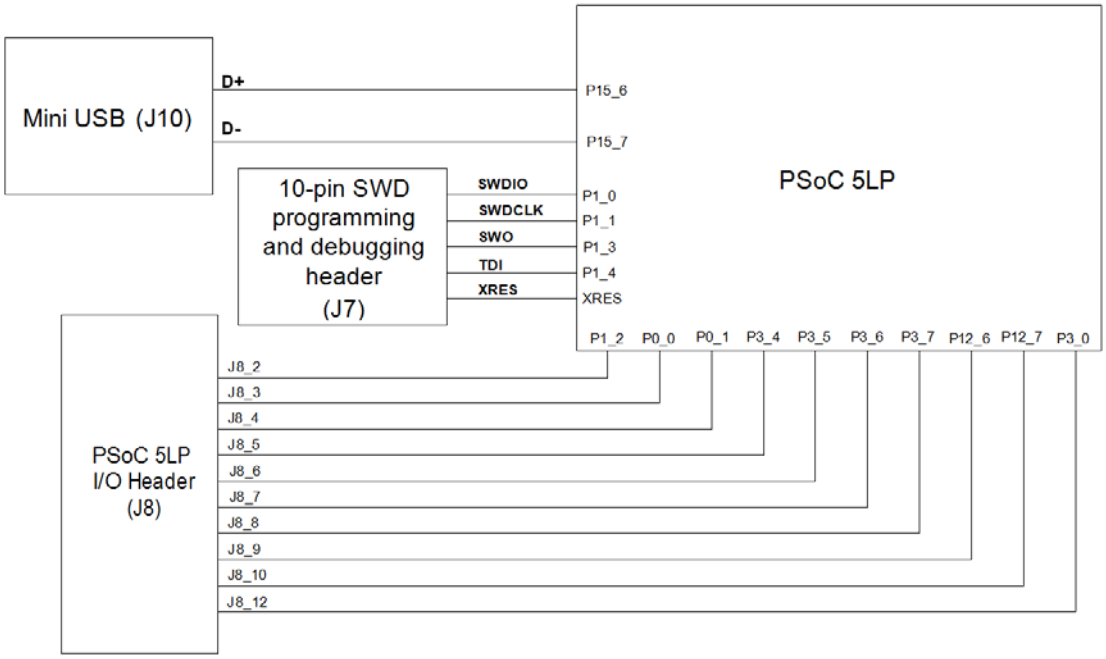

## 6.4 开发 PSoC 5LP 的应用

PSoC 4000 Pioneer 套件具有一个板载 PSoC 5LP，其主要功能是作为编程器和桥接器使用。用户可以使用 PSoC 5LP 来编译普通项目或者 Bootloadable 项目。

图 6-40 汇总了电路板上的 PSoC 5LP 连接。 J8 是 I/O 连接器 （请参考 A.2.2 PSoC 5LP GPIO 插座（J8））。已连接 USB （J10），并作为 PC 接口使用。然而，您仍可以使用该 USB 连接创建自定义 USB 设计。

编程插座（J7）用于独立编程目的。需要自行安装该插座。请参考A.6 材料清单中的“空载组件”章节

图 6-40. PSoC 5LP 框图



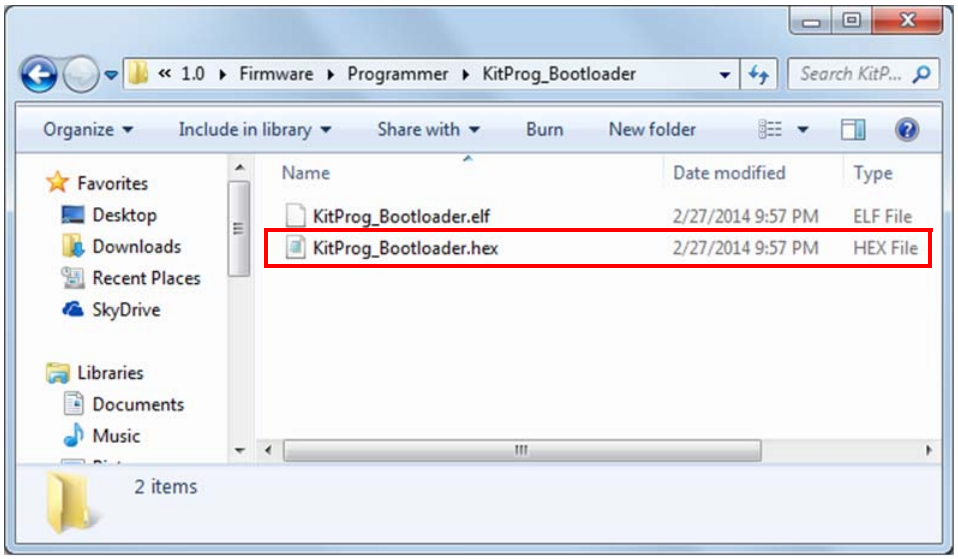

## 6.4.1 为 PSoC 5LP 编译 Bootloadable 项目

所有PSoC 5LP的Bootloadable应用需要基于已经编程在板上的Bootloader十六进制文件。Bootloader的十六进制文件可在套件文件中获得，或可从套件网页下载。

十六进制文件位于以下套件安装程序目录中，如图 6-41 所示：

```
<Install_Directory>\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>\Firmware\Programmer\KitProg_Bootloader
```

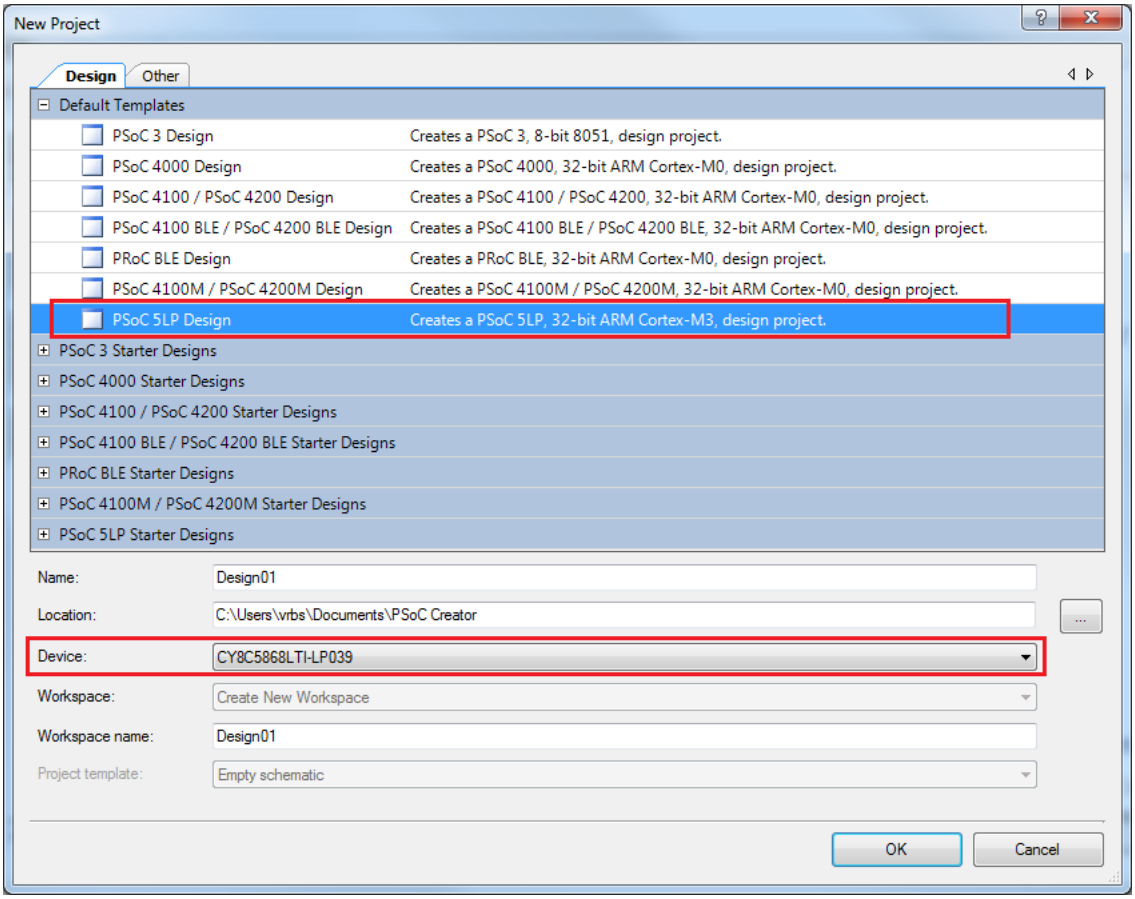
图 6-41. KitProg Bootloader 十六进制文件的位置



按照以下程序为 PSoC 5LP 构建所需要的 Bootloadable 应用：

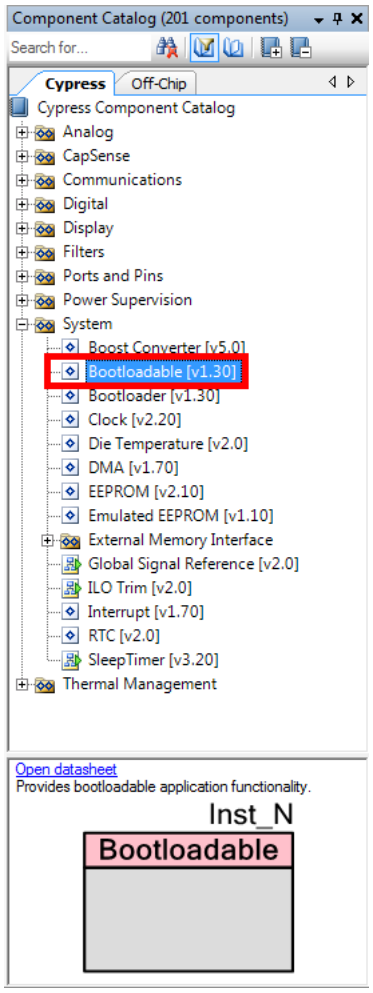
1. 在 PSoC Creator 中，依次选择 **New > Project > PSoC 5LP**；点击 Advanced 旁边的扩展按键并在 **Device** 项选择 **CY8C5868LTI-LP039**，如图 6-42 所示。

图 6-42. 在 PSoC Creator 中创建新项目。



2. 打开原理图窗口并将 **Bootloadable** 组件拖放到顶层设计内。

图 6-43. 组件目录中的 Bootloadable 组件



3. 通过选择配置窗口中的 **Dependencies** 选项卡并点击 **Browse** 按键，可以设置 Bootloadable 组件的依赖关系，如图 6-44 所示。选择 *KitProg_Bootloader.hex* 和 *KitProg_Bootloader.elf* 文件；然后点击 **Open** （如图 6-45 和图 6-46 所示），最后点击配置窗口上的 **OK**。

图 6-44. Bootloadable 组件的配置窗口

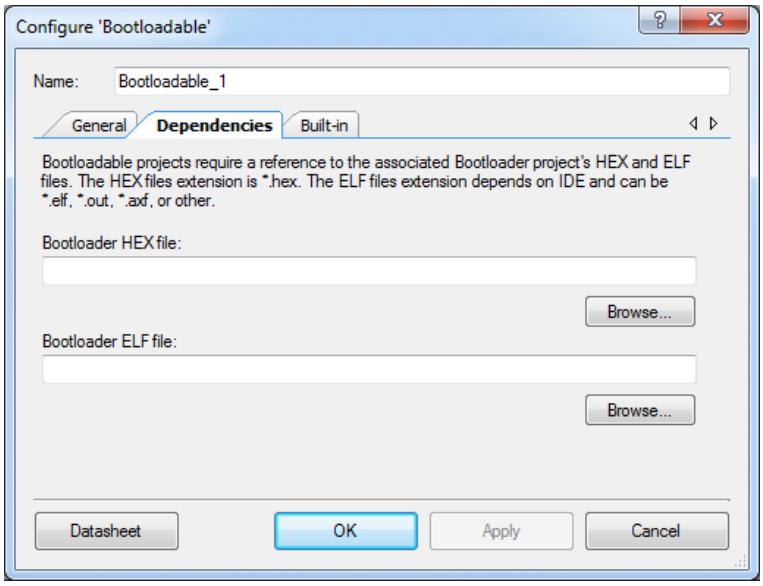


图 6-45. 选择 KitProg Bootloader hex 文件

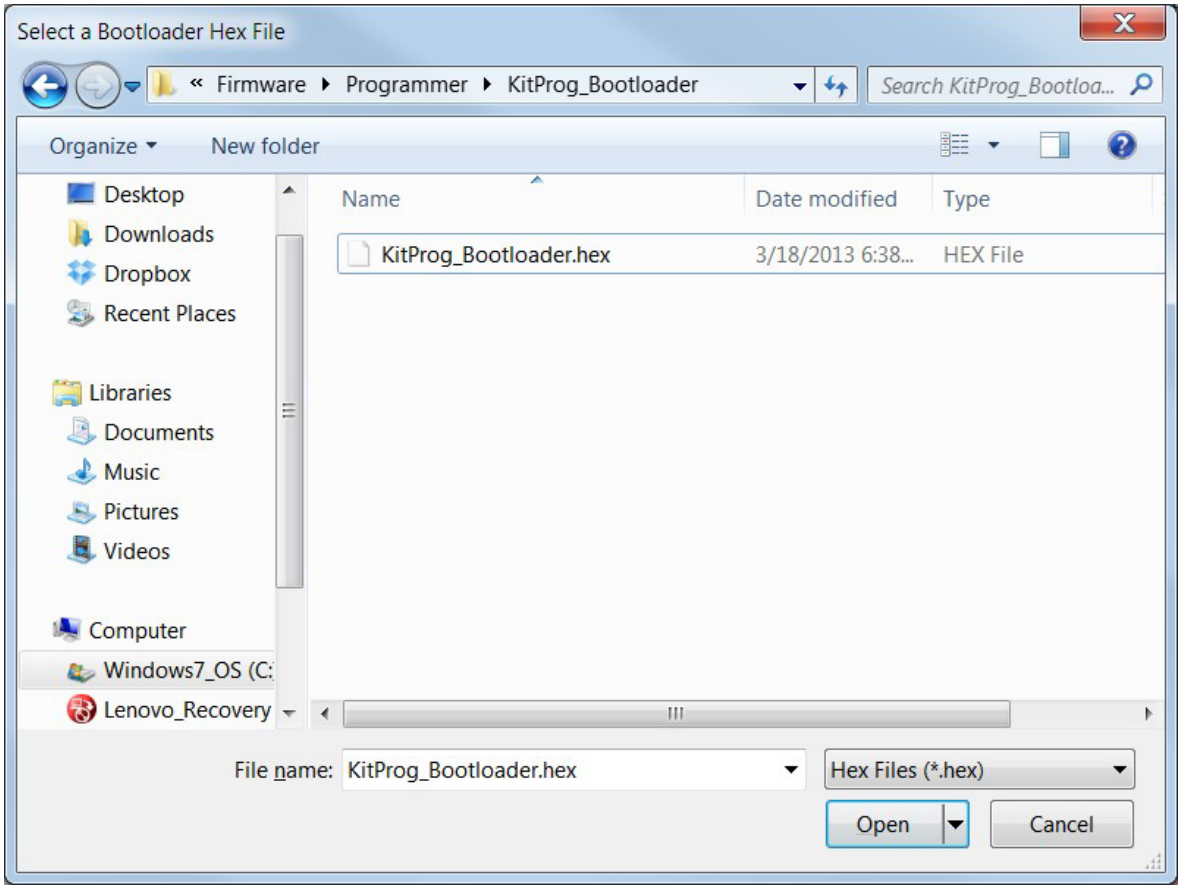
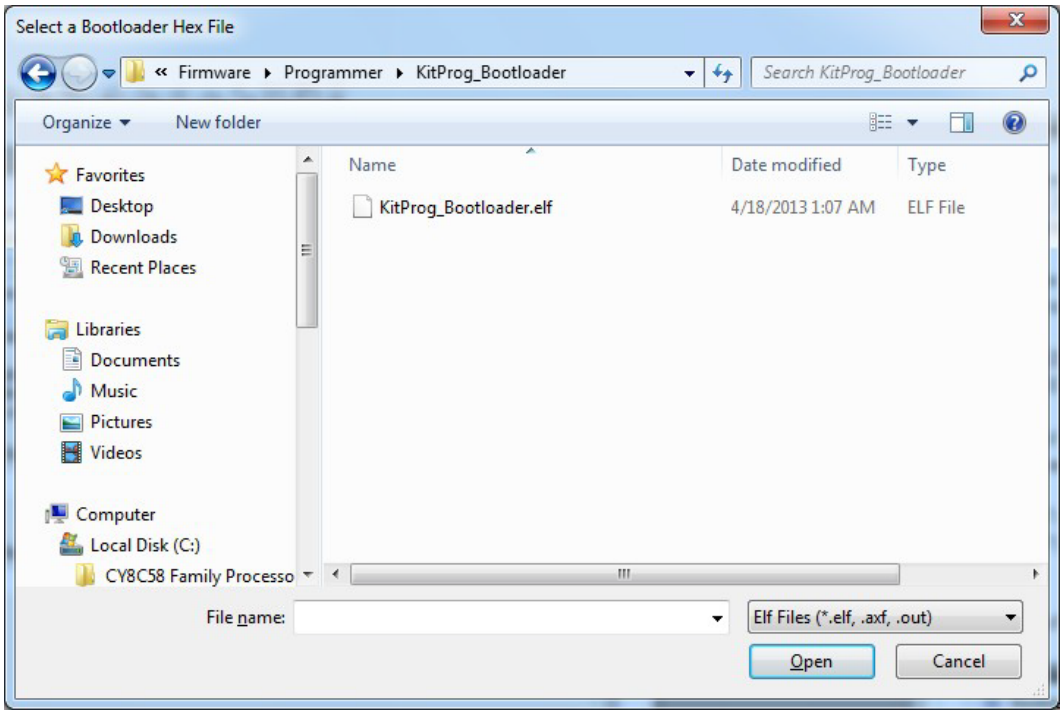
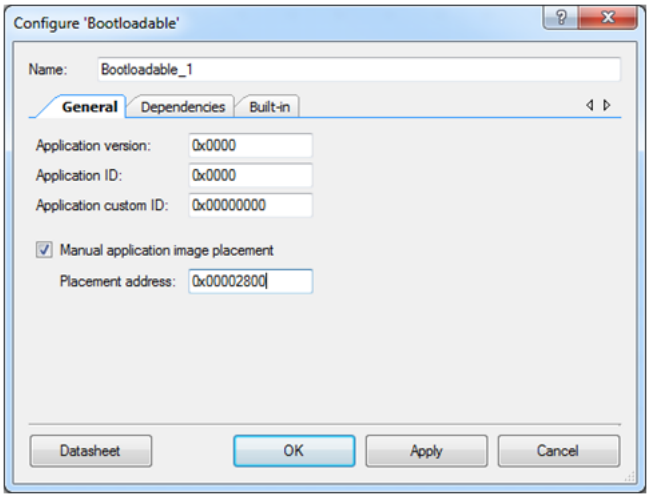



图 6-46. 选择 KitProg Bootloader Elf 文件



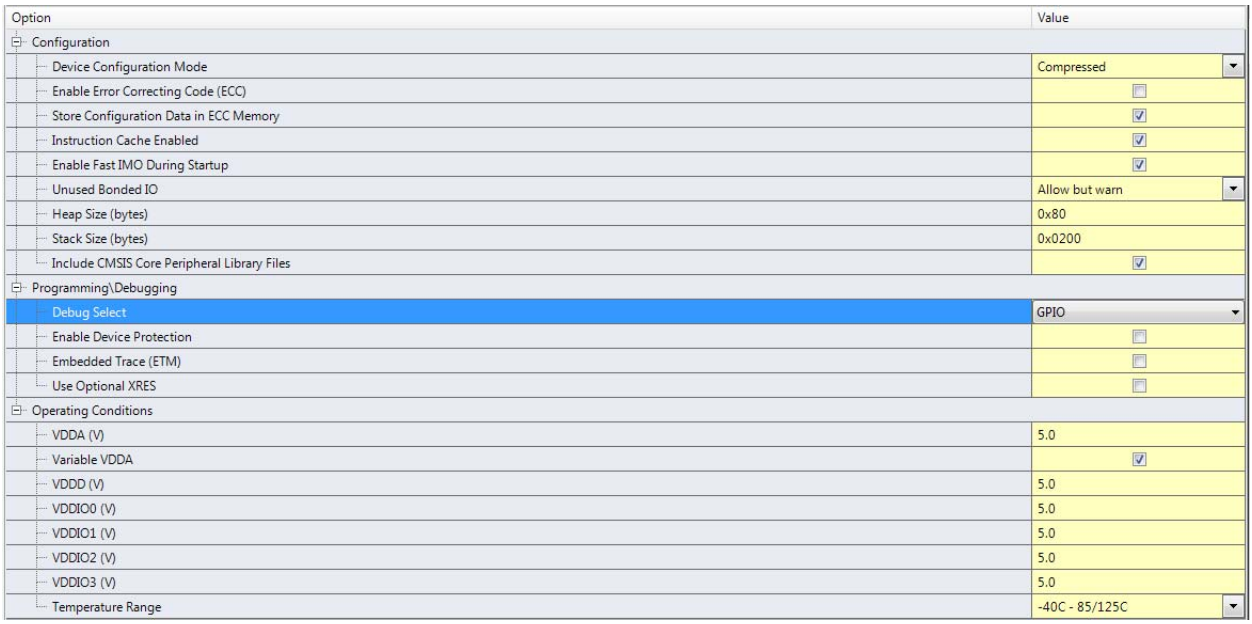
4. 在**General**选项卡中勾选**Manual application image placement**复选框，并将Placement address设置为 “0x00002800”，如图 6-47 所示。

图 6-47. Bootloadable 组件的 “General” 选项卡




5. 开发您的自定义项目。
6. Bootloadable 项目的 NVL 设置必须与 KitProg_Bootloader 项目的相同。图 6-48 显示了 *KitProg_Bootloader.cydwr* 系统设置。

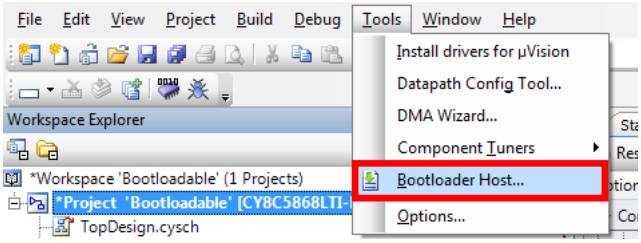
图 6-48. KitProg Bootloader 系统设置




7. 通过依次点击 **Build > Build Project** 或按组合键 **[Shift] [F6]**，可以在 PSoC Creator 中构建项目。
8. 按住复位开关 （SW1），同时插上 USB Mini-B 连接器。若按住开关的时间超过 100 ms， PSoC 5LP 将进入 Bootloader 模式。

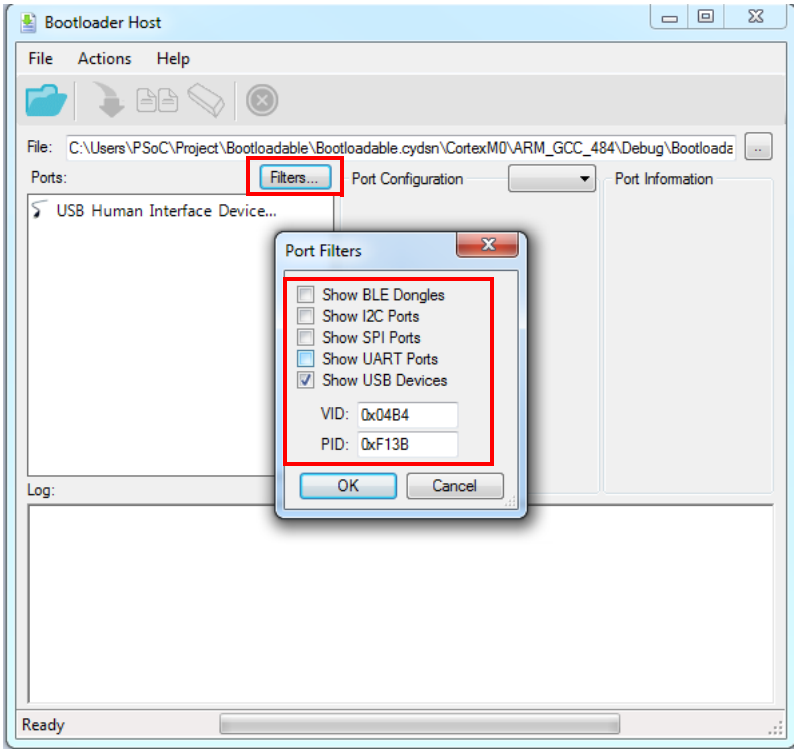
9. 要想将该项目下载到 PSoC 5LP 器件上，请打开 PSoC Creator 中的 Bootloader 主机工具 。依次选择 **Tools > Bootloader Host**，如图 6-49 所示。

图 6-49. 从 PSoC Creator 打开 Bootloader 主机工具




10. 在 Bootloader Host 工具中，点击 **Filters** （滤波器）并添加一个过滤器，以识别 USB 器件。点击 **OK**，如图 6-50 所示。

图 6-50. Bootloader 主机工具中的 Port Filters 选项卡



11. 在 Bootloader 主机工具中点击 **Open File** 按键以浏览到 Bootloadable 文件 （*.cyacd）所在的位置，如图 6-51 所示。

图 6-51. 在 Bootloader 主机工具中打开 Bootloadable 文件

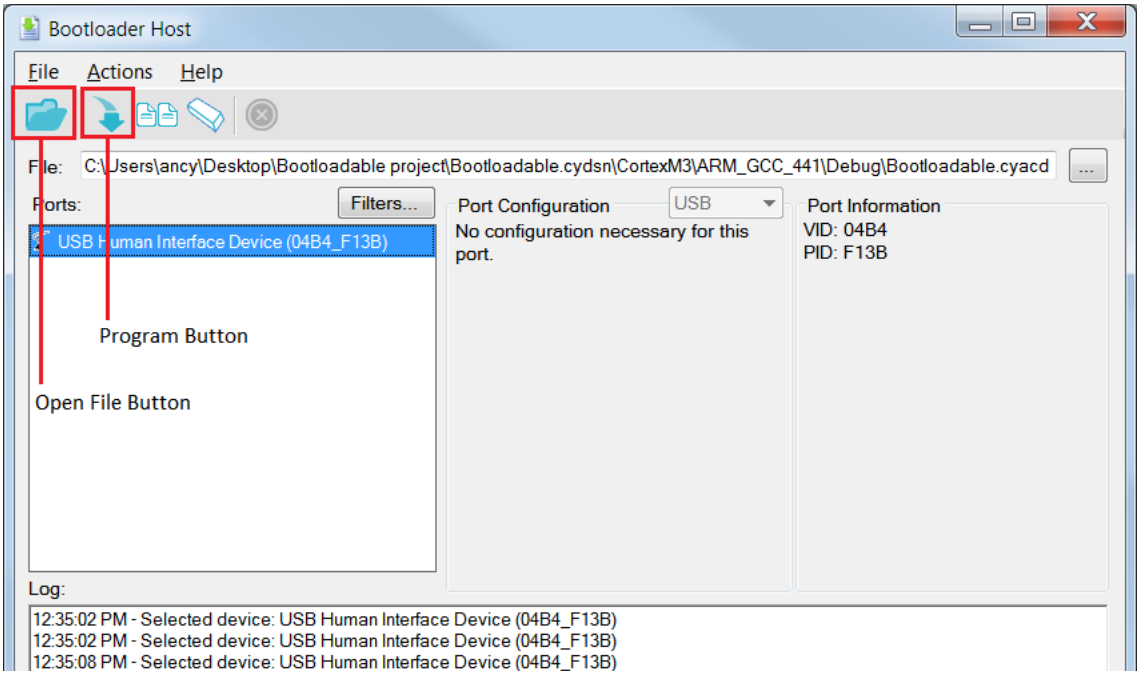
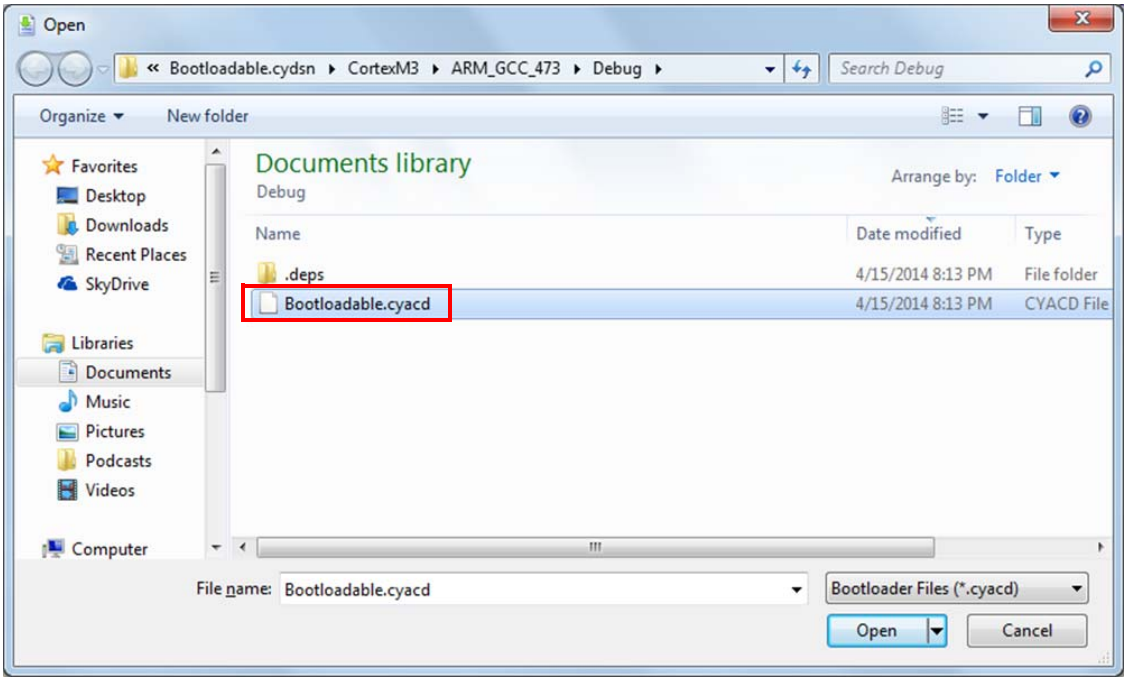



图 6-52. 从 Bootloader 主机工具中选择 Bootloadable .cyacd 文件




12. 点击 Bootloader Host 工具中的 **Program** 按键便可对器件进行加载。
13. 若引导加载成功，工具记录将显示 “Programming Finished Successfully” （成功完成编程）；否则，会显示 “Failed”，并显示一条失败语句。

**注意：**

- PSoC 5LP 的引脚被连接到 PSoC 5LP GPIO 插座（J8）上。这些引脚用于支持高性能的模拟和数字项目。有关引脚的详细信息，请参见 A.2.2 PSoC 5LP GPIO 插座 （J8）。
- 当给自定义应用分配 PSoC 5LP 的引脚时，应格外注意。例如，P2[0]-P2[4] 是专用于编程 PSoC 4 的。在对引脚进行分配前，请参见 A.1 CY8CKIT-040 原理图中的内容。
- 当将某个自定义的 Bootloadable 项目编程到 PSoC 5LP 内时，该 PSoC 5LP 的初始功能（即作为编程器 /USB-UART 桥接器或 USB-I$^2$C 桥接器使用）都将无效。
- 除非自定义项目使用状态 LED，否则该状态 LED 不会工作。

更多有关 Bootloader 信息，请参见 AN73503 — 赛普拉斯 *PSoC3 和 PSoC 5LP 的 USB HID Bootloader* 应用笔记。

### 6.4.2 编译 PSoC 5LP 的普通项目

普通项目是指在 CY8CKIT-040 中为 PSoC 5LP 器件创建的全新项目。在这里，将对 PSoC 5LP 上整个闪存进行编程，包括覆盖所有的 Bootloader 和编程代码。要想恢复编程器，需要通过随套件安装程序附带的出厂设置 *KitProg.hex* 文件重新编程 PSoC 5LP 器件。

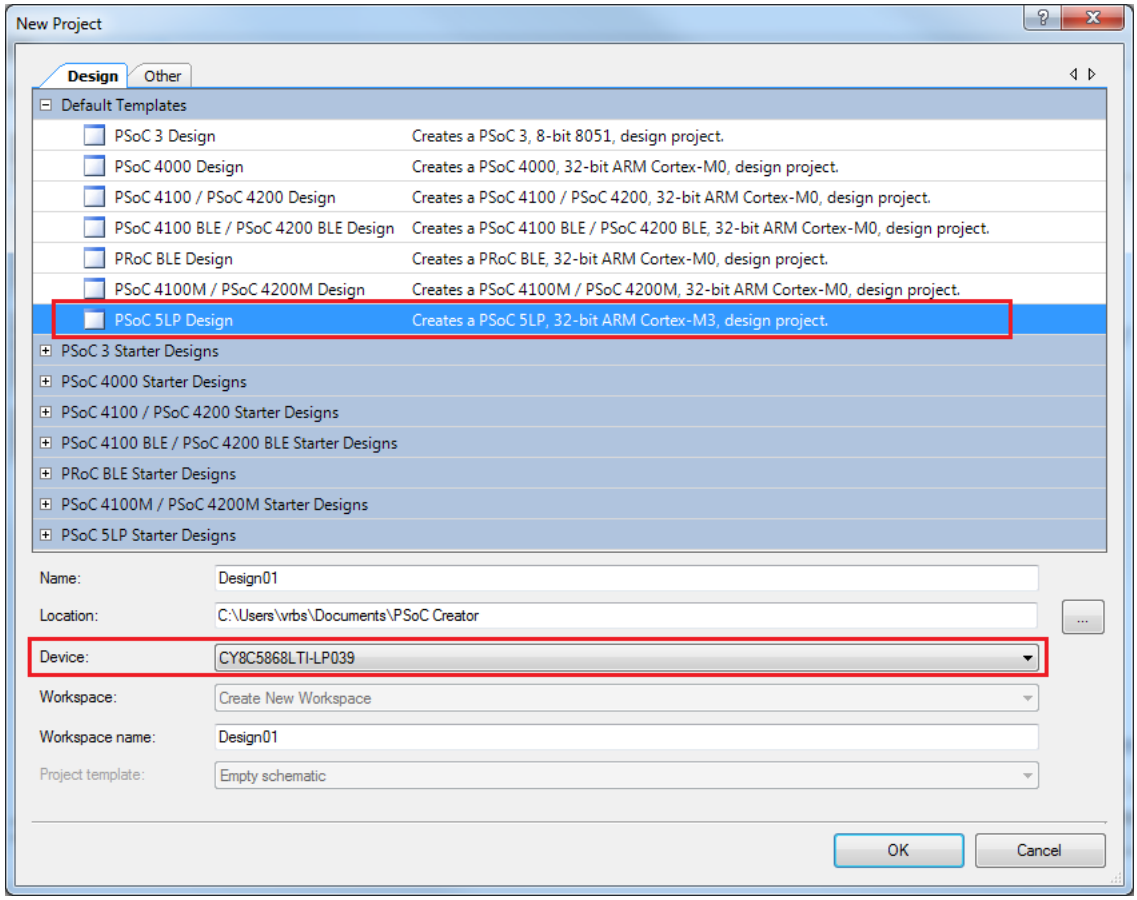
可以在下面的地址中找到 *KitProg.hex* 文件：

```
<Install_Directory>\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>\Firmware\Programmer\KitProg
```

该高级功能需要使用一个不属于此套件的 MiniProg3 编程器。可以在 www.cypress.com/go/CY8CKIT-002 上购买 MiniProg3。要想为 PSoC 5LP 构建一个普通项目，请按照下列步骤进行操作：

1. 在 PSoC Creator 中，依次选择 **New > Project > PSoC 5LP Design**；点击 Advanced 旁边的扩展按键并在 **Device** 项选择 **CY8C5868LTI-LP039**，如图 6-53 所示。

图 6-53. 在 PSoC Creator 中创建新项目。




2. 开发您的自定义项目
3. 通过依次点击 **Build > Build Project** 或按组合键 **[Shift] [F6]**，可以在 PSoC Creator 中构建项目。
4. 将 MiniProg3 的 10 引脚连接器连接到（需要自行安装）板上的 10 引脚 SWD 调试和编程插座 J7。
5. 要想通过 PSoC Creator 编程 PSoC 5LP，请依次点击 **Debug > Program** 或按组合键 **[Ctrl]+[F5]**。编程窗口将显示 MiniProg3 和项目 （CY8C5868LTI-LP039）中所选定的器件。
6. 先点击该器件再点击 **Connect**，以进行编程。

**注意：**

- 板上默认并未焊接 10 脚 SWD 调试和编程插座（J7）。更多有关信息，请参考第 167 页上的材料清单 中的 “ 空载组件 ” 章节。
- PSoC 5LP 的引脚被连接到 PSoC 5LP GPIO 插座（J8）上。这些引脚用于支持高性能的模拟和数字项目。有关引脚的详细信息，请参见第 164 页上的 PSoC 5LP GPIO 插座 （J8）。
- 当给自定义应用分配 PSoC 5LP 的引脚时，应格外注意。例如，P2[0]-P2[4] 是专用于编程 PSoC 4 的。在对引脚进行分配前，请参见第 157 页上的 CY8CKIT-040 原理图 中的内容。

- 当将某个普通项目编程到 PSoC 5LP 内时，该 PSoC 5LP 的初始功能（即作为编程器 /USB-UART 桥接器或 USB-I$^2$C 桥接器使用）都将无效。
- 除非自定义项目使用状态 LED，否则该状态 LED 不会工作。

## 6.5 PSoC 5LP 出厂设置恢复说明

CY8CKIT-040 PSoC 4000 Pioneer 套件包含一个带有出厂设置的 PSoC 5LP 器件。该器件可作为 PSoC 4 器件的板上编程器和调试器使用。

除了为 PSoC 4 器件创建应用外，用户亦可在本套件上创建 PSoC 5LP 器件的自定义应用，如第 129 页上的开发 PSoC 5LP 的应用 中所述。通过新的闪存映像重新编程或引导加载 PSoC 5LP 器件时，出厂设置将被覆盖掉，同时会失去将 PSoC 5LP 器件作为 PSoC 4 编程器 / 调试器使用的能力。要想恢复 PSoC 5LP 的出厂设置并使能编程器 / 调试器的功能，请执行下面各操作。

### 6.5.1 使用 Bootloadable 应用编程过的 PSoC 5LP

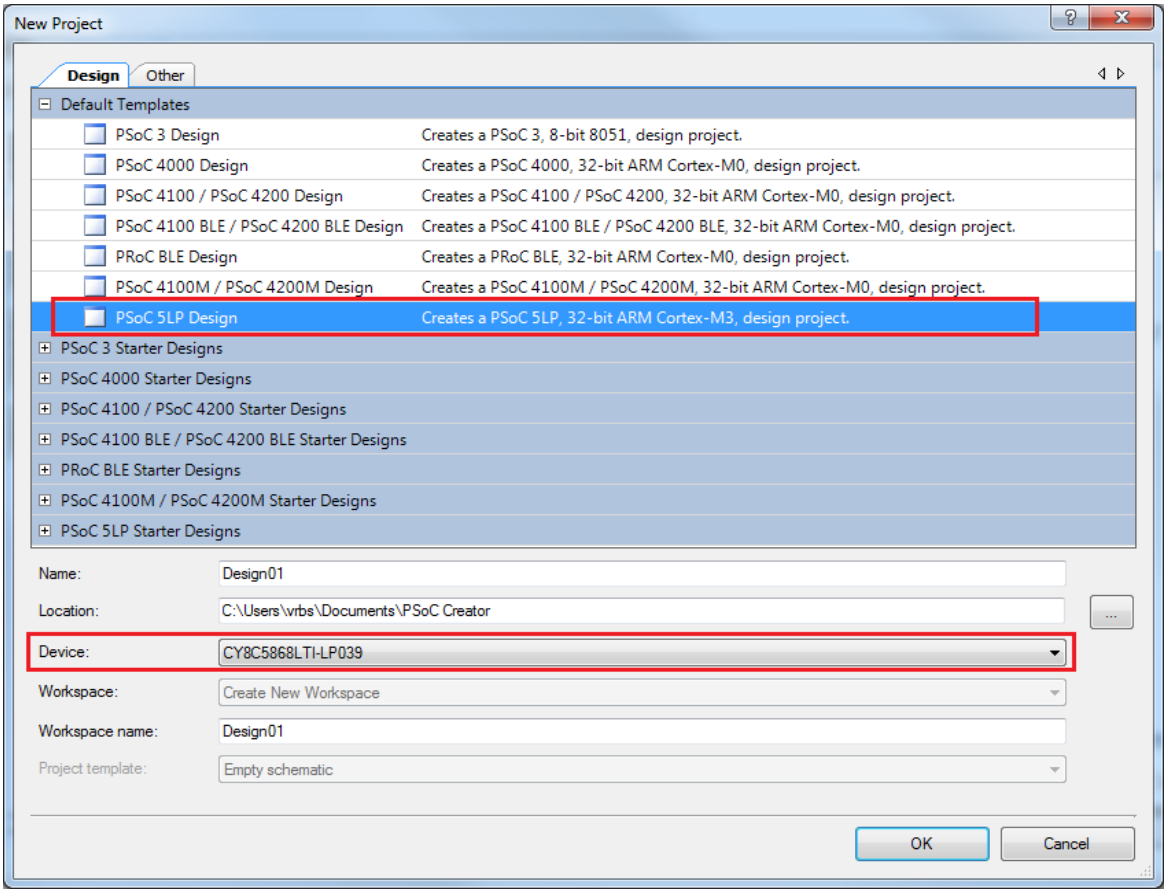
若 PSoC 5LP 已通过 Bootloadable 应用编程过，用户可以使用以下某一个方法恢复出厂设置。

#### *6.5.1.1 使用 PSoC 编程器恢复 PSoC 5LP 出厂设置*

1. 从 **Start > Cypress > PSoC Programmer** 启动 PSoC 编程器 3.23.1 或更高版本。
2. 设置该套件进入服务模式。要进入该模式，当按住复位按键 （SW1 Reset）时，同时使用所包含的 USB 电缆 （USB A 转 Mini-B）将 PSoC 4000 Pioneer 套件连接到计算机。这样的操作会使 PSoC 5LP 进入服务模式，闪烁状态的绿色 LED 表明进入该模式成功。

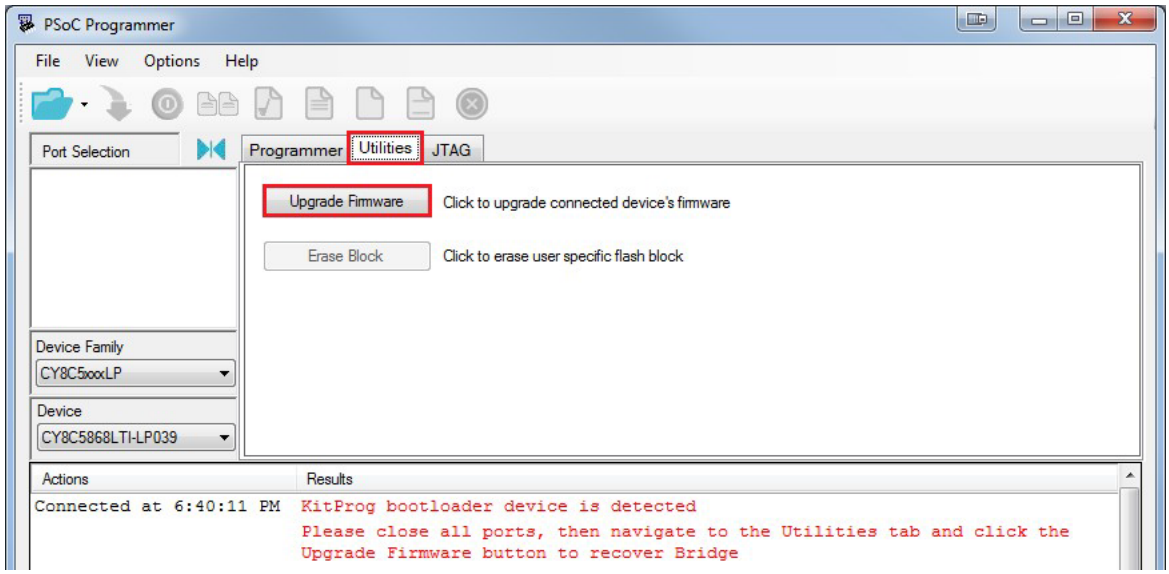
3. 显示在图 6-54 中的以下信息在 PSoC 编程器结果窗口中显示：“KitProg Bootloader device is detected”（已检测到 KitProg Bootloader 器件）。

图 6-54. PSoC 编程器结果窗口



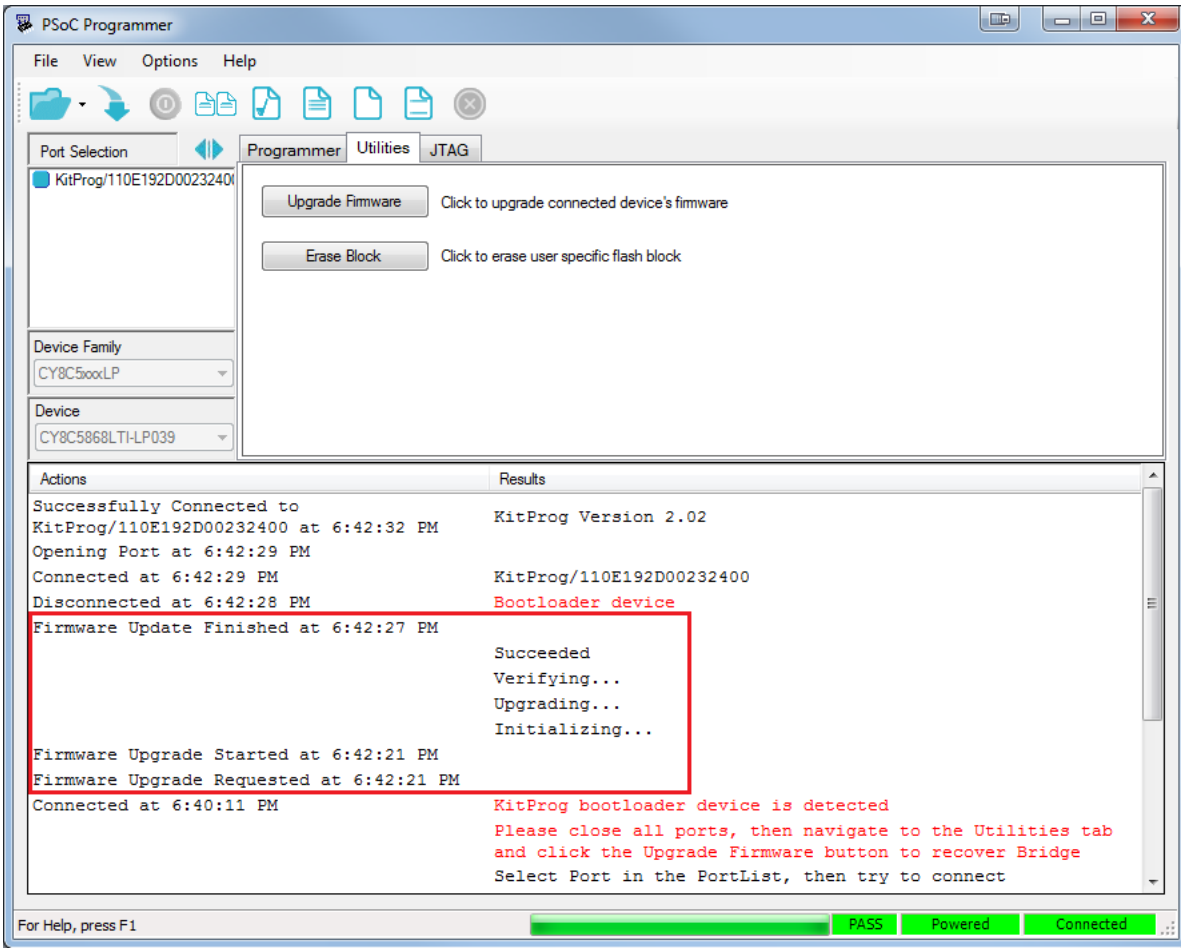
4. 切换到 PSoC 编程器窗口中的 **Utilities** 选项卡，并点击 **Upgrade Firmware** 按键，如图 6-55 中所示。点击 **Upgrade Firmware** 按键前，用户需要在 PC 端拔掉其他所有 PSoC 编程器（如 MiniProg3 和 DVKProg）。

图 6-55. 更新固件



5. 编程完成后，将出现以下信息 “Firmware Update Finished at <time>” （固件升级在 < 时间 > 内完成），如图 6-56 中所示。

图 6-56. 固件更新完成




6. 用户已成功地恢复 PSoC 5LP 的出厂设置。它可作为 PSoC 4 器件上的编程器 / 调试器使用。

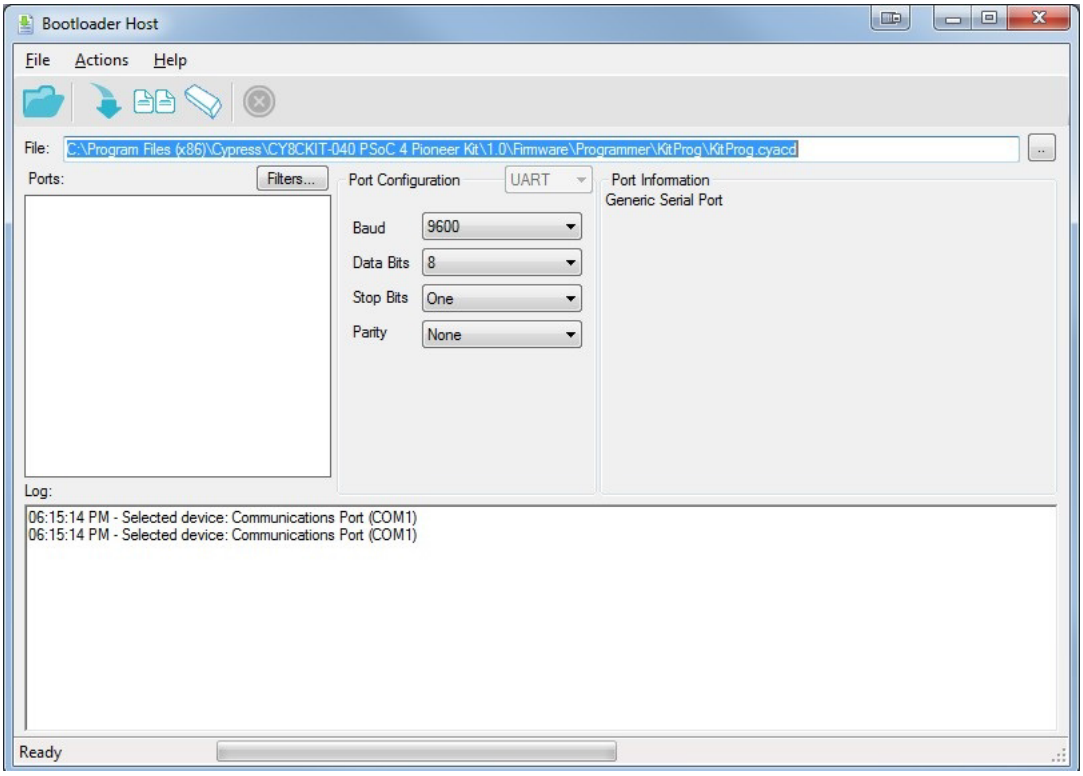
### 6.5.1.2 使用 USB Host 工具恢复 PSoC 5LP 的出厂设置

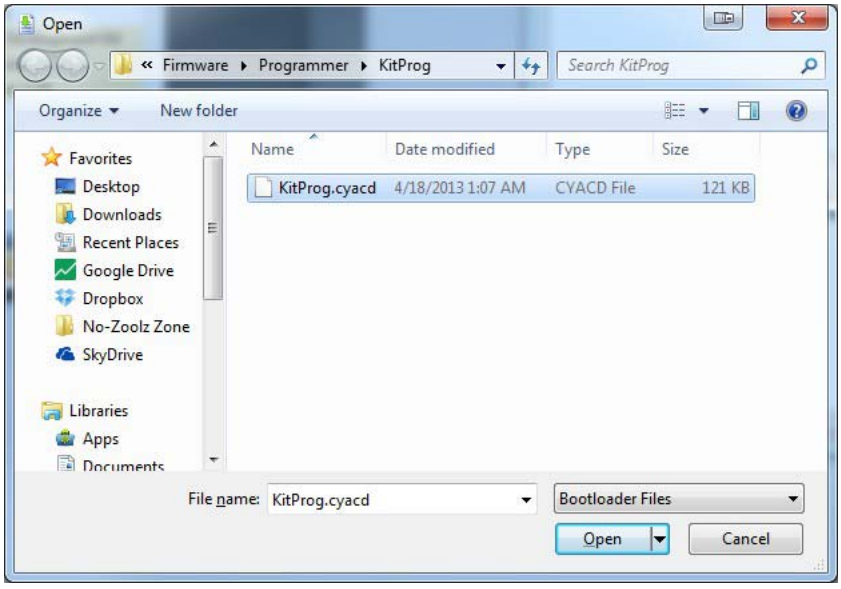
1. 依次选择 **Start > Cypress > PSoC Creator**，启动 Bootloader Host 工具。

   使用 **File > Open** 菜单，加载 *Kit Prog.cyacd* 文件 （该文件通过套件软件安装）。

   该文件的默认位置为：`<Install_Directory>\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>\Firmware\Programmer\KitProg\KitProg.cyacd`。

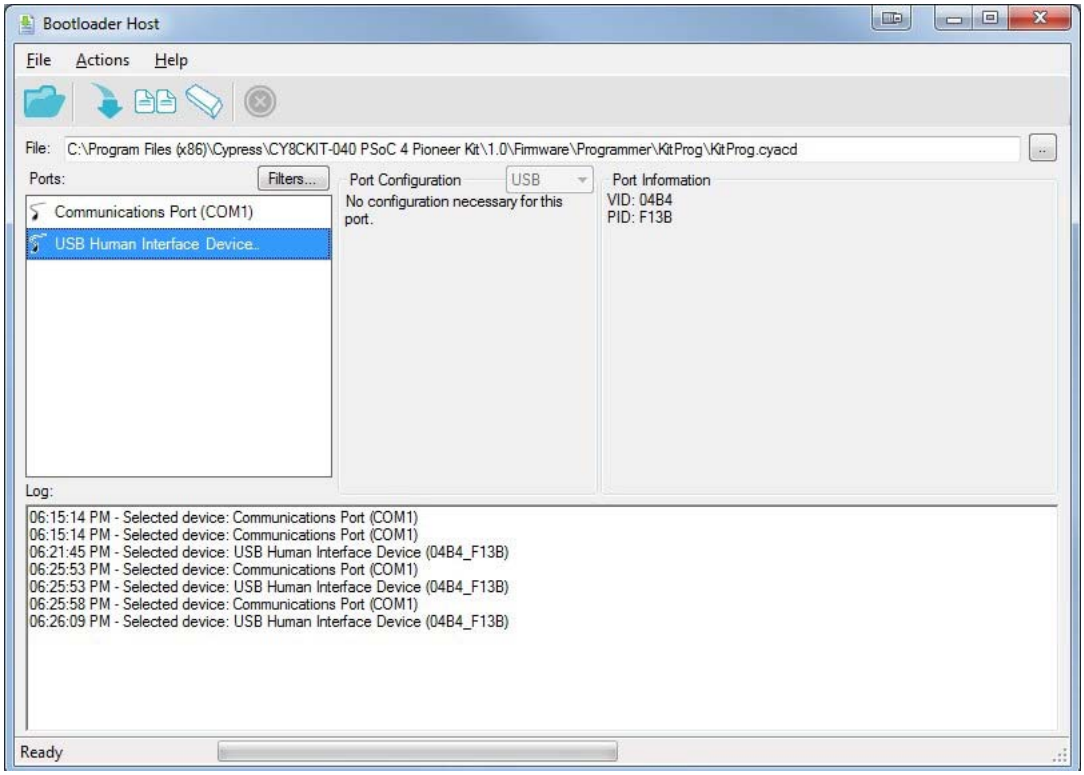
图 6-57. 加载 KitProg.cyacd 文件



2. 设置 Pioneer 套件进入服务模式。要进入该模式，当按住复位按键 （SW1 Reset）时，同时使用所包含的 USB 电缆 （USB A 到 USB 微型 B）将 PSoC 4000 Pioneer 套件连接到计算机。这样的操作会使 PSoC 5LP 进入服务模式，闪烁状态的绿色 LED 表明进入该模式成功。
3. 在 Bootloader Host 工具中，使用 USB 器件过滤，其中 VID 设为 04B4 ； PID 设为 F13B。 **USB Human Interface Device** 端口出现在端口列表中。通过点击该端口来选择。

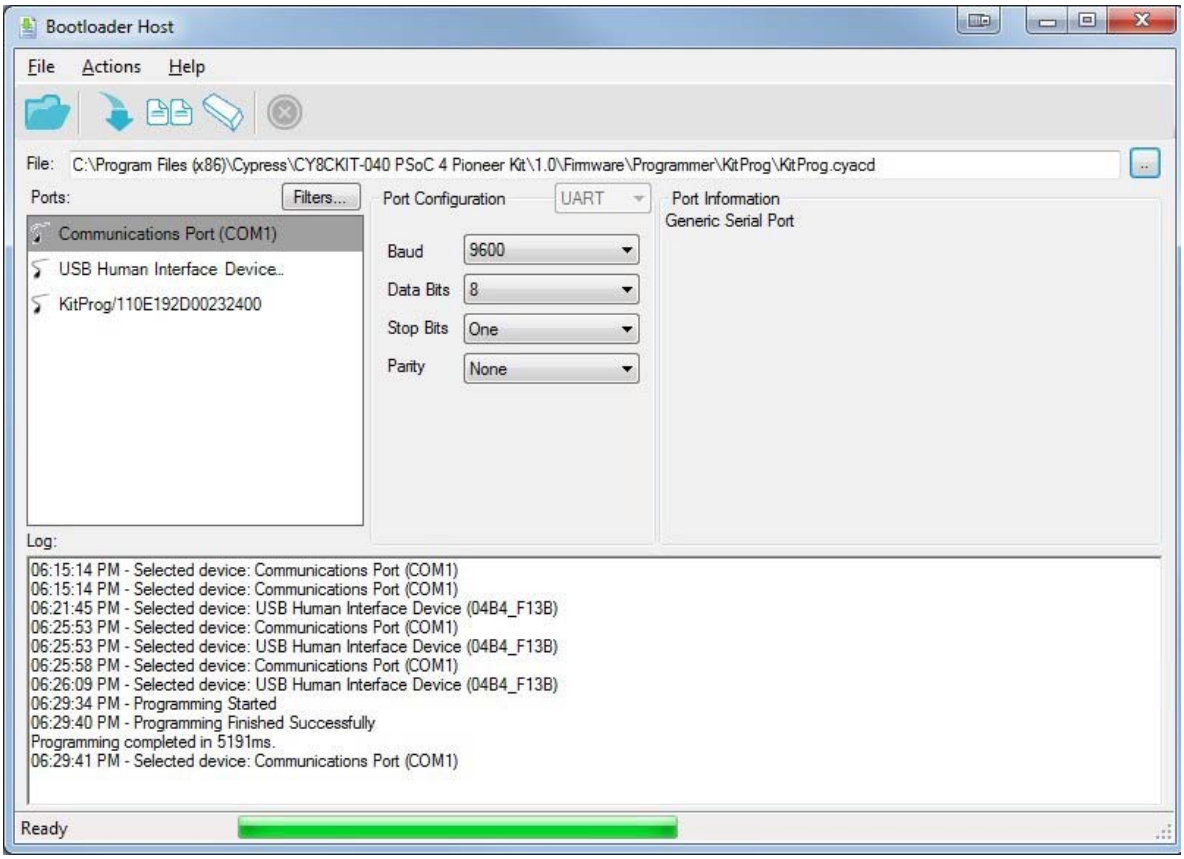
图 6-58. 选择 “USB Human Interface Device”




4. 点击 **Program** 按键（或从菜单项依次选择 **Actions > Program**），通过引导加载到 PSoC 5LP 上以恢复出厂设置。

5. 编程完成后，将出现以下信息 “Programming Finished Successfully”（成功完成编程），如图 6-59 所示。

图 6-59. 成功完成编程




6. PSoC 5LP 已成功地恢复它的出厂设置。它可作为 PSoC 4 器件上的编程器 / 调试器使用。

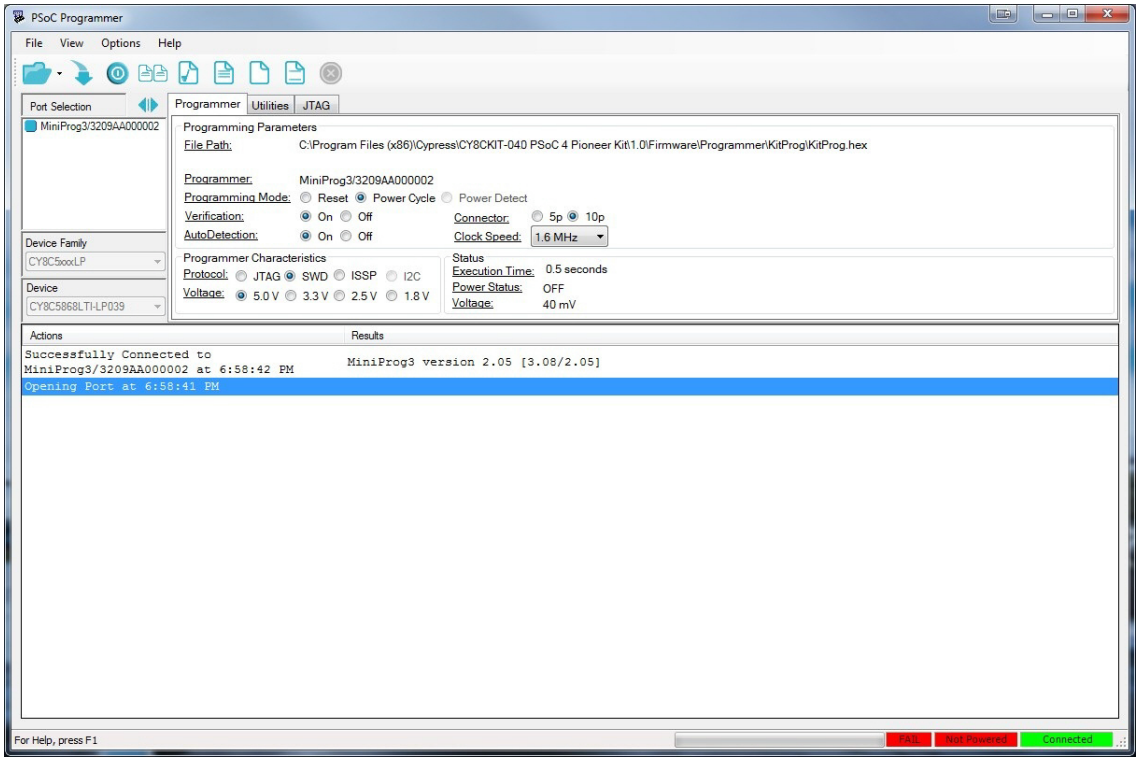
### 6.5.2 使用标准的应用编程过的 PSoC 5LP

若 PSoC 5LP 已使用标准的应用编程过，那么，用户可以通过以下方法恢复出厂设置。

1. 依次选择 **Start > Cypress > PSoC Programmer**，启动 PSoC 编程器 3.23.1 或更高版本。
2. 通过依次选择 **File > Open** 菜单加载随该套件附带的十六进制 *KitProg.hex* 出厂设置文件。该文件的默认位置为：`<Install_Directory>\CY8CKIT-040 PSoC 4000 Pioneer Kit\<version>\Firmware\Programmer\KitProg`。
3. 将 CY8CKIT-002 MiniProg3（单独销售）连接到计算机。将 MiniProg3 的 10 引脚连接器电缆插入到插座 [J7] 内。请注意，插座 J7 默认没有焊接安装。更多信息，请查阅第 167 页上的材料清单 。

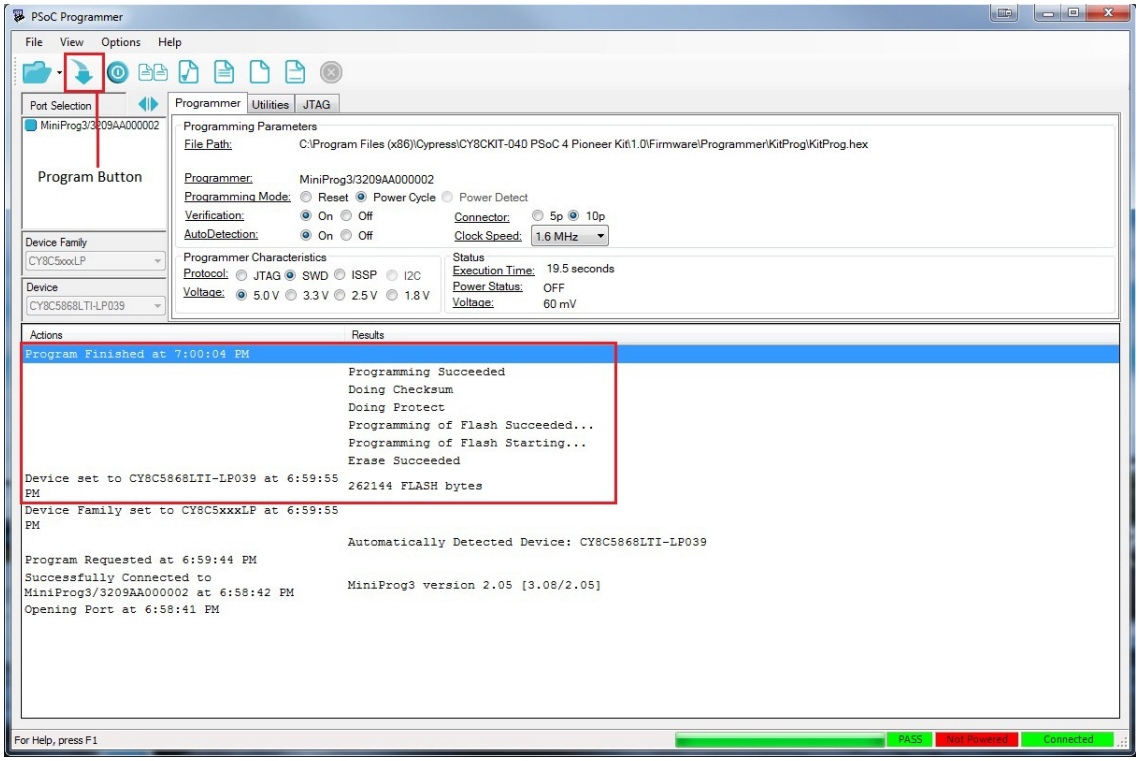
4. 请确保 MiniProg3 是 PSoC 编程器中选定的端口，并且已选中 10 引脚连接器选项 （10p 选项），如图 6-60 所示。若 USB 不为电路板供电，请选择 **Power Cycle** 编程模式。

图 6-60. 选择 MiniProg3



5. 就绪状态下，请按下 **Program** 按键（或依次选择 **File > Program**），以编程 PSoC 5LP 器件，如图 6-61 所示。
6. 编程完成后，将出现以下信息 “Program Finished at <time>” （编程在 < 时间 > 内完成）。

图 6-61. 编程完成




7. PSoC 5LP 已成功地恢复它的出厂设置。它可作为 PSoC 4 器件上的编程器 / 调试器使用。

## 6.6 使用 µC/Probe 工具

Micrium 的 µc/Probe 是一个 Windows 应用，让您可以在运行时对任何嵌入式目标处理器的内存执行读写操作，并将这些值映射到位于图形仪表板上的一组虚拟控件和指示器。

该工具有助于为开发套件示例项目设计一个简单的GUI，而且工作量最小。请注意，Micrium µC/Probe 工具不是运行该套件所需的必备软件，并且不随套件内容一起安装。

使用该工具的所有功能所需的许可证需要单独购买。但您可以在 http://micrium.com/download/ucprobe-3-0-trial-version/ 网址上免费获取该工具的教学用版本。

更多有关 µC/Probe 工具及其使用许可的信息，请访问 http://micrium.com/download/%c2%b5cprobe-3-0-users-manual/ 地址下的 µC/Probe 用户手册。

Micrium µC/Probe 版本 3.3 支持使用赛普拉斯的 KitProg 作为与连接到计算机的目标器件进行通信的工具。编译好PSoC Creator的示例项目时，它会生成HEX、LST、MAP、RPT和ELF格式的输出文件。

ELF 文件列出了所有符号 （变量）、符号类型及其地址。 Micrium µC/Probe 工具读取 ELF 文件并检测代码中所使用的这些符号 （全局变量）。

µC/Probe 工具提供了一系列的图形控制，如滑条、 RGB 调色板、图表以及圆环图。可以将需要的控制类型拖放到 µC/Probe 工作台上，并将 EFL 文件中的符号分配给相应的控制类型。

运行工作台时，可以在计算机上显示与控件关联的符号值 （例如滑块和图形）的更改。

已在套件的 CY8CKIT_040_Color_Palette 和 CY8CKIT_040_Proximity_UART 示例项目中分配了适当的全局变量，以显示 GUI 上的调色板输出和接近输出。

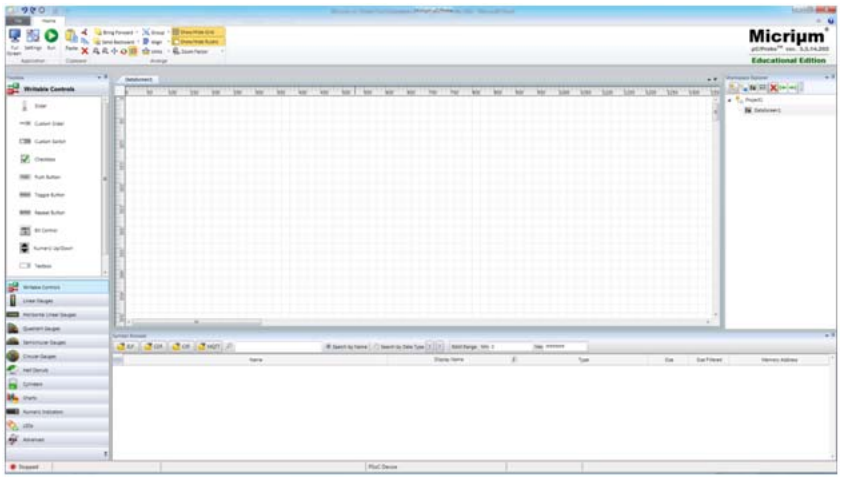
### CY8CKIT_040_Color_Palette 示例项目

要使用 µC/Probe 工具可视化 CY8CKIT_040_Color_Palette 项目的输出，请按照下列步骤操作：

1. 在 PSoC Creator 中打开 CY8CKIT-040_Color_Palette 示例项目。
2. 按照第49 页上的编程示例项目 中介绍的第1步到第8步，在CY8CKIT-040上编程CY8CKIT_040_-Color_Palette 示例项目。

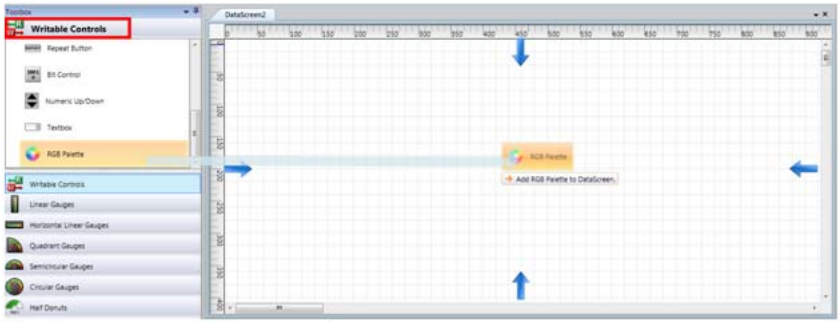
   **请注意：**编程电路板前，请确保将 *.cydwr* 文件中系统设置的 **Debug Select** 设置为 **SWD**，请参考第 53 页上的图 5-8。默认情况下， CY8CKIT_040_Color_Palette 示例项目将 **Debug Select** 设置为 **SWD**。如果需要在项目中启用 TX 端口，则应将其更改为 GPIO。
3. 从 http://micrium.com/download/ucprobe-3-0-trial-version/ 网址上下载并安装 µC/Probe 工具。
4. 依次选择 **Start > All Programs > Micrium > uC-Probe > Micruim uC-Probe**，启动 µC/Probe。

图 6-62. Micrium µC/Probe



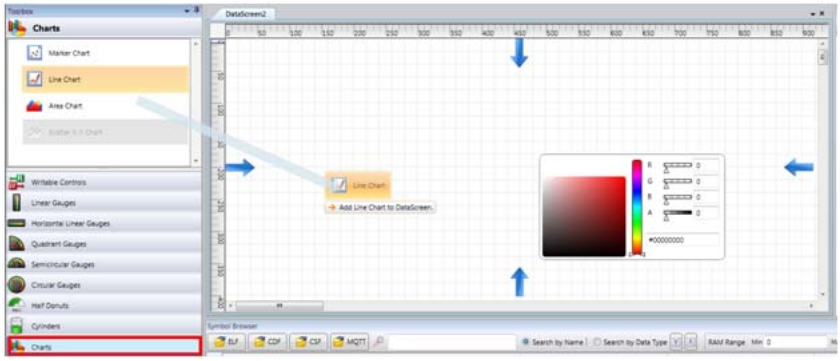
5. 将 RGB 调色板从工具箱中的 **Writable Controls** （可写控件）拖放到 DataScreen 上，以可视化 RGB 输出。

图 6-63. 添加 RGB Palette 控制



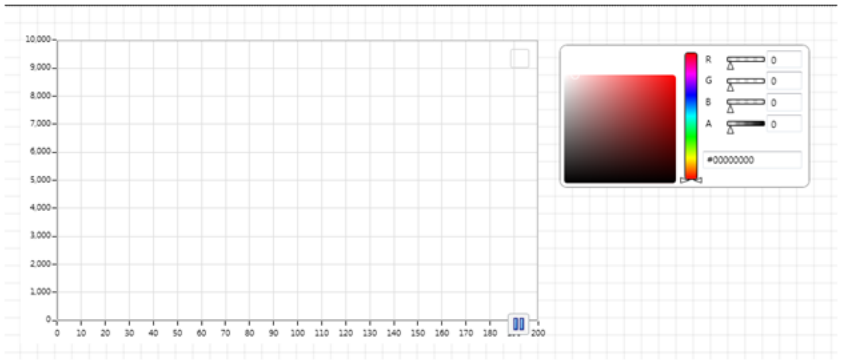

6. 接下来，将工具箱中**Charts**的折线图添加到DataScreen上，以显示手指在触控板上的X和Y位置。

图 6-64. 添加折线图控件



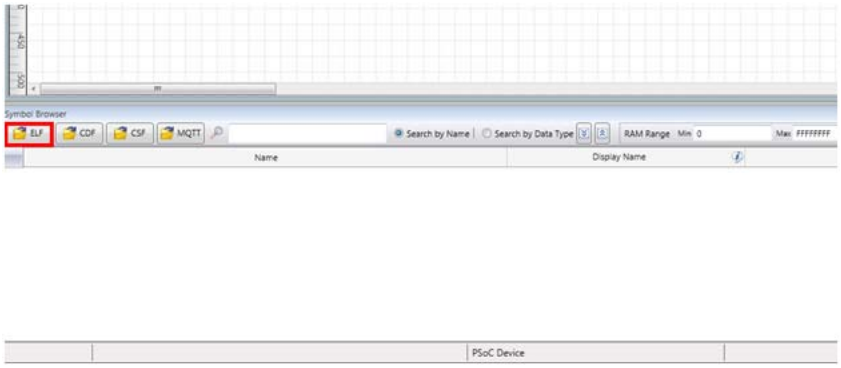

7. 添加两个控件后， DataScreen 看起来如图 6-65 所示。

图 6-65. 带有 RGB 调色板和折线图控件的 DataScreen



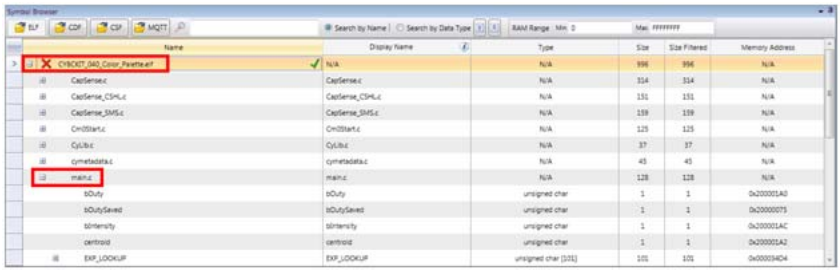
8. 点击 Symbol Browser 窗口中的 **ELF** 按键。

图 6-66. Symbol Browser 中的 EFL 按键




9. 请浏览并指向 *CY8CKIT_040_Color_Palette.ELF* 文件以加载 CapSense 示例项目的符号（全局变量）。等待 ELF 文件加载完成。 EFL 文件默认为折叠状态。要想扩展 EFL 文件，请点击该文件名旁边的 ‘+’ 符号。
10. 展开文件后，Symbol Browser 将显示示例项目中的所有 .C 文件。通过展开每一个文件，您可以看到 C 文件中所定义的全局变量及其名称、类型、尺寸、存储器地址等信息。
11. 现在，请展开 *main.c* 文件，查看该文件中所定义的全局变量。

图 6-67. Symbol Browser 中的全局变量



12. 可以将全局变量 µcARGB 拖放到 RGB Palette 控制面板中，以查看 RGB 输出。同样，将 xPos 和 yPos 拖放到折线图控件上，以便在触控板上查看手指的 X 和 Y 位置。

图 6-68. 将 RGB 输出分配给 RGB 调色板控件

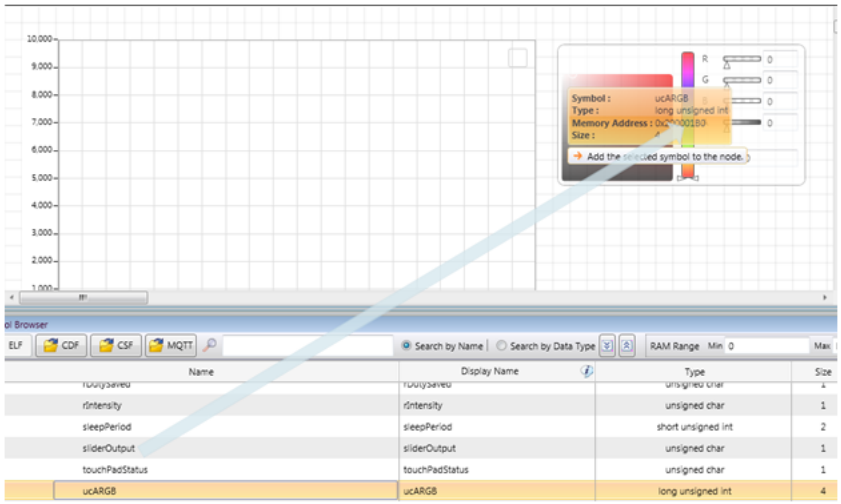



图 6-69. 将 x 位置输出分配给折线图控件

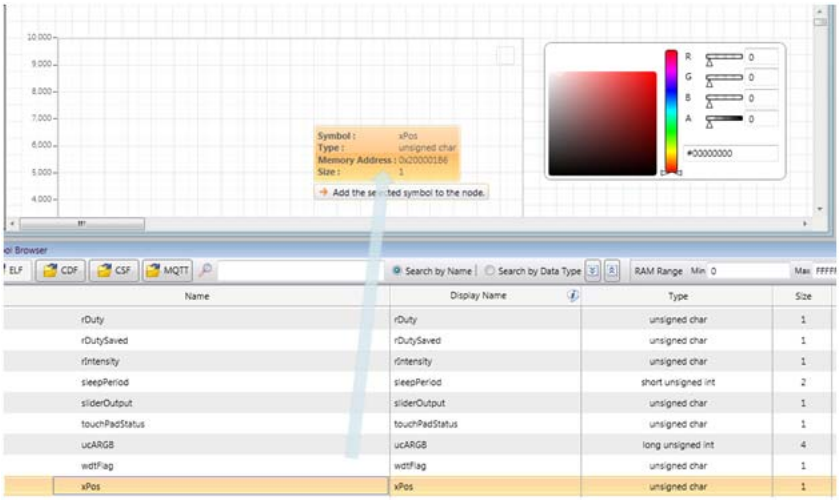
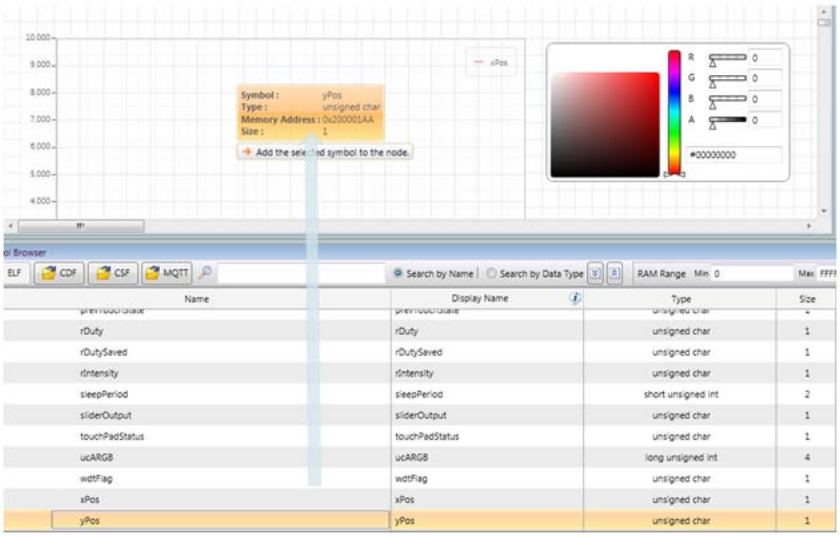


图 6-70. 将 y 位置输出分配给折线图控件




13. 默认情况下，折线图的 Y 轴绘制为 0 到 10000 之间的值。要将 Y 轴刻度更改为 0 到 100，请单击折线图控件。在出现的折线图控件工具栏中，单击 **Properties Editor** 图标。请参见图 6-71 的内容。

图 6-71. 折线图控制工具栏

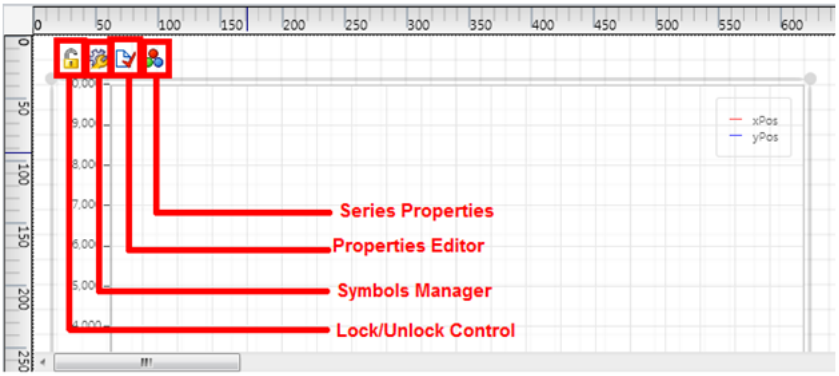
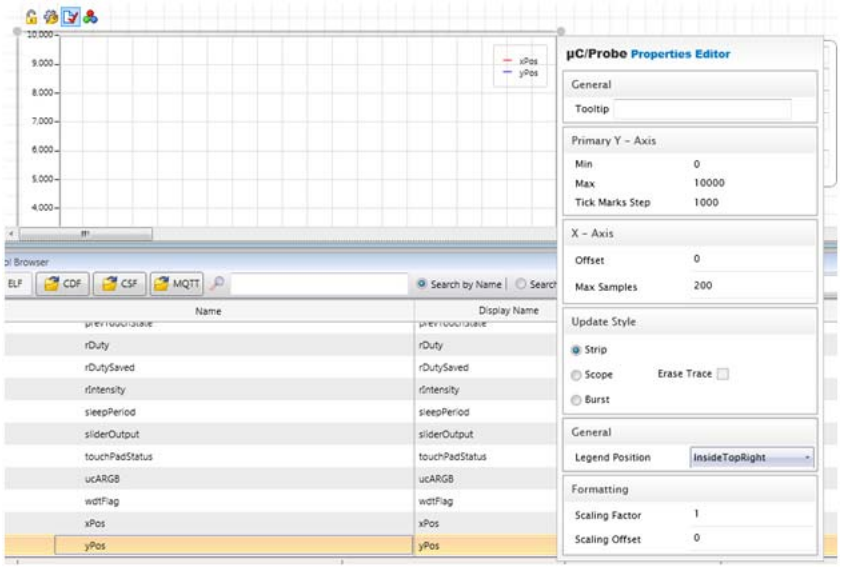


图 6-72. 折线图控制的 Properties Editor




14. 在 Properties Editor 窗口中，将 Y 轴的 **Max** 属性更改为 100.

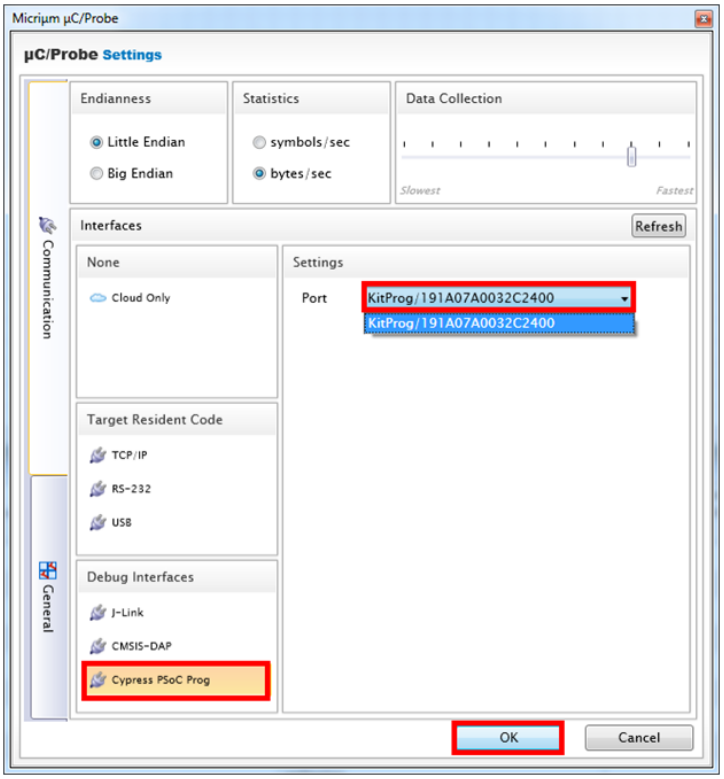
**注意：**您可以使用 Series 属性更改 xPos 和 yPos 的折线图颜色。此外，符号管理器显示与控件关联的全局变量。

15. 现在，将 CY8CKIT-040 连接到计算机上。点击 µC/Probe 工具中的 **Settings** 按键

图 6-73. µC/Probe 中的 Settings 按键

16. 在 µC/Probe 设置窗口中，先选择 **Cypress PSoC Prog**，然后选择 Port 下拉列表中的 **KitProg/<Kit Prog number>**，并点击 **OK** 以启动 CY8CKIT-040 和 µC/Probe 工具间的通信。

图 6-74. µC/Probe 设置

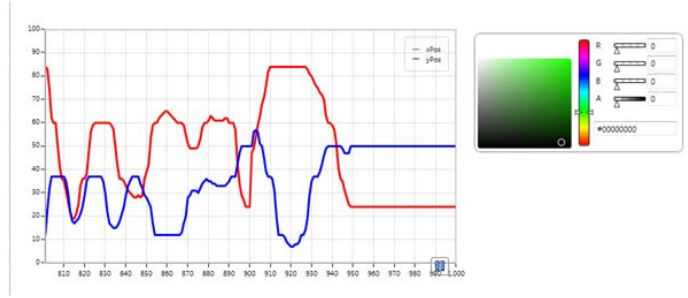



17. 点击 **Run** 按键以开始执行。

图 6-75. Run 按键

18. 现在，将手指移动到套件上的触控板上，并观察 Datascreen 上的折线图和 RGB 调色板控件输出。

图 6-76. 折线图和 RGB 调色板输出



**注意：**如果您正在使用 µC/Probe 工具的教学用版本，在启动 Datascreen 前将显示弹出窗口。单击 **OK** 按键，继续进行操作。另外，使用教学用版本时，Datascreen（输出）会在 1 分钟后超时。

19. 点击 **File** 选项卡，然后选择 **Save** 以保存 µC/Probe 项目。提供合适的名称并选择保存项目的位置。 µC/Probe 项目以扩展名 .WSPX 保存。双击 .WSPX 文件，打开 µC/Probe 工具。

### CY8CKIT_040_Proximity_UART 示例项目

已创建 CY8CKIT_040_Proximity_UART 项目的 µC/ Probe 项目，并随工具包内容一起打包。您可以在安装文件夹 `<Install directory>/ CY8CKIT-040 PSoC 4000 Pioneer Kit/ <version>/uCProbe` 中找到 µC/Probe 项目。

1. 在 PSoC Creator 中打开 CY8CKIT-040_Proximity_UART 示例项目。
2. 在示例项目的 cydwr 文件的系统设置中，将 **Debug Select** 更改为 **SWD** 并构建项目。请参见第 53 页上的图 5-8 的内容。
3. 按照第 49 页上的编程示例项目 中介绍的第 1 步到第 8 步，在 CY8CKIT-040 上编程 CY8CKIT-040_Proximity_UART 示例项目。
4. 双击 *CY8CKIT-040_Proximity_UART.wspx* 文件。
5. 将导线环插入接近传感器。

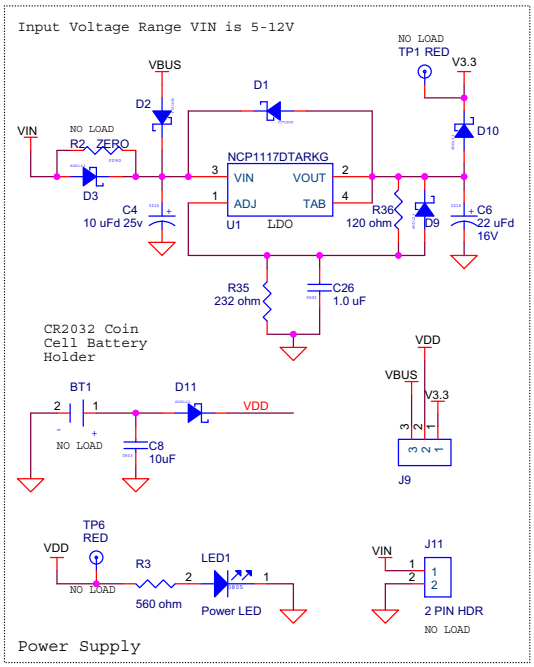
   **注意：**将导线环插入传感器时，请确保电路板未通电。否则，插入导线环会导致基线偏移，接近传感器输出将不正确。
6. 将 CY8CKIT-040 连接到计算机，然后按照 CY8CKIT_040_Color_Palette 示例项目部分中描述的第 14 步到第 16 步开始运行 Datascreen。
7. 将手掌靠近导线环。当接近值小于 100 并且滑条上显示接近值时，LED 处于 OFF 状态。当接近值为 100 时， LED 变为 ON。

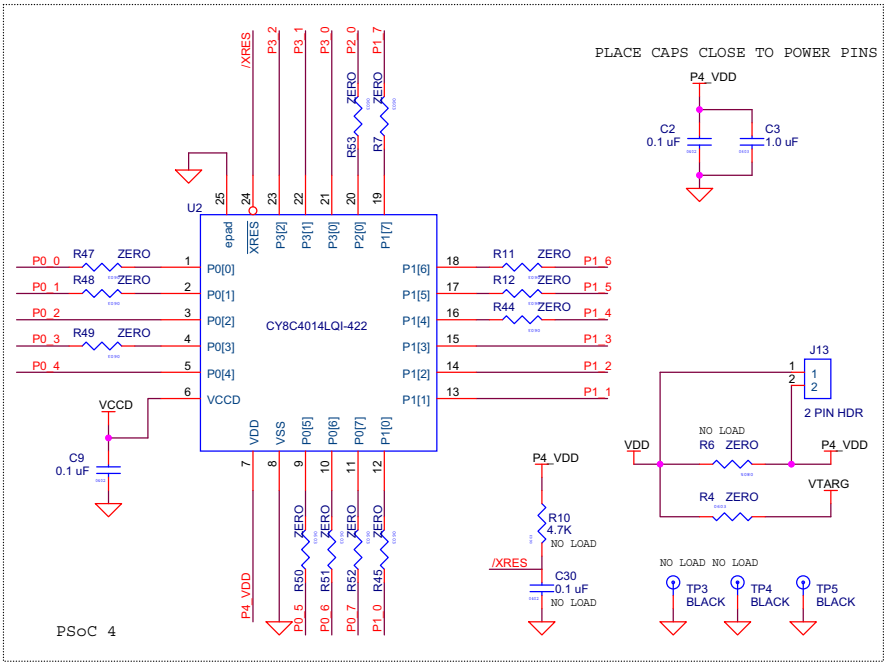
图 6-77. 远离传感器时的 LED 和滑条控制输出

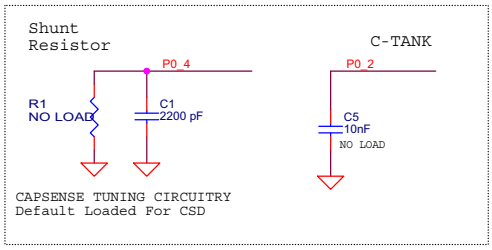
图 6-78. 靠近传感器的 LED 和滑条控制输出

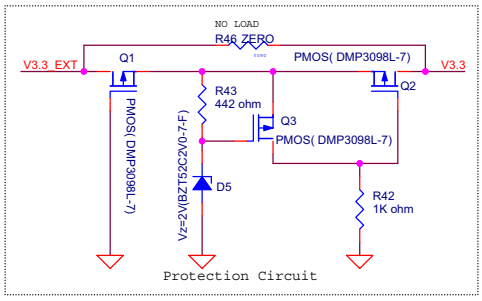
# A. 附录

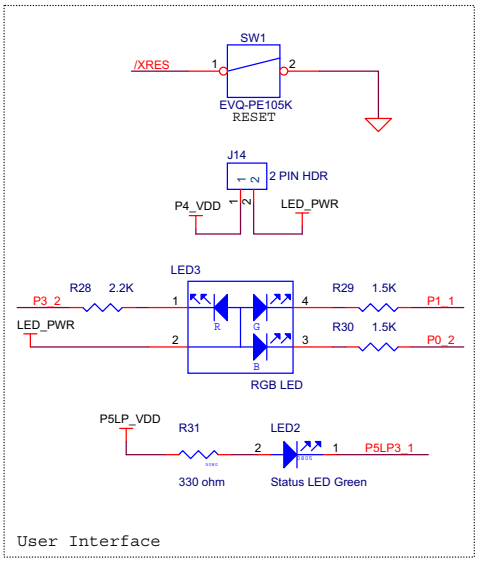
## A.1 CY8CKIT-040 原理图

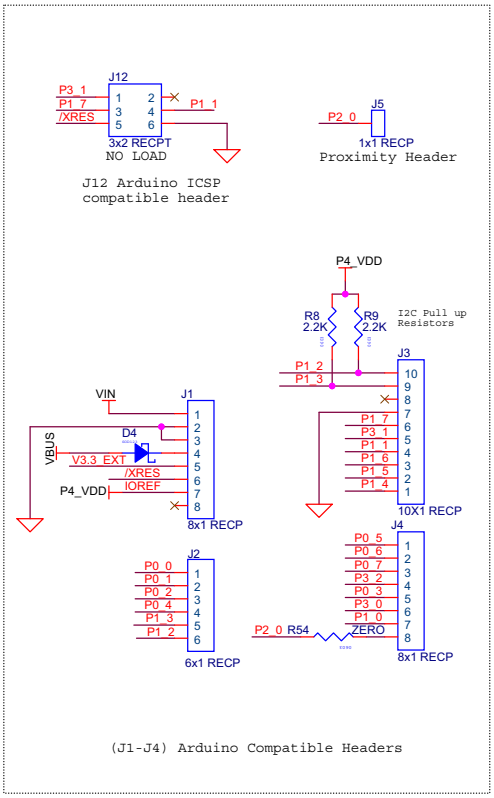


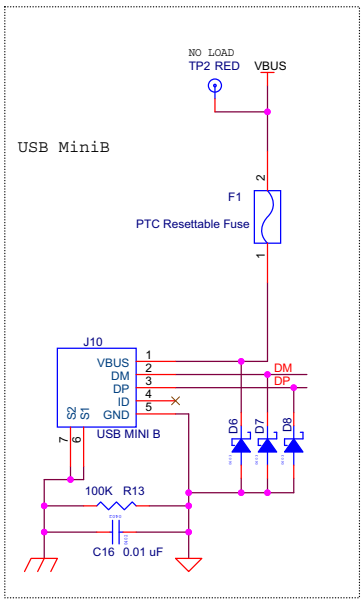

PLACE CAPS CLOSE TO POWER PINS
P4_VDD
C2
0.1 uF
C3
1.0 uF
U2
CY8C4014LQI-422
/XRES
P3_2
P3_1
P3_0
P2_0
P1_7
R53 ZERO
R7 ZERO
P0_0 R47 ZERO
P0_1 R48 ZERO
P0_2
P0_3 R49 ZERO
P0_4
P0[0]
P0[1]
P0[2]
P0[3]
P0[4]
VCCD
P1[6]
P1[5]
P1[4]
P1[3]
P1[2]
P1[1]
R11 ZERO P1_6
R12 ZERO P1_5
R44 ZERO P1_4
P1_3
P1_2
P1_1
VDD
VSS
P0[5]
P0[6]
P0[7]
P1[0]
R50 ZERO P0_5
R51 ZERO P0_6
R52 ZERO P0_7
R45 ZERO P1_0
P4_VDD
C9
0.1 uF
R10
4.7K
NO LOAD
C30
0.1 uF
NO LOAD
J13
2 PIN HDR
NO LOAD
R6 ZERO
R4 ZERO
VTARG
NO LOAD NO LOAD
TP3 BLACK
TP4 BLACK
TP5 BLACK
PSoC 4

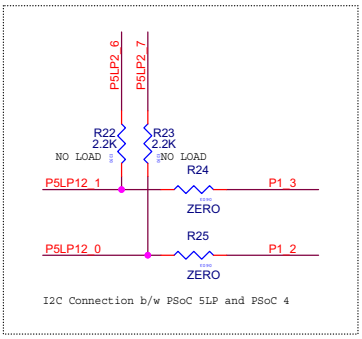

Shunt
Resistor
P0_4
R1
NO LOAD
C1
2200 pF
C-TANK
P0_2
C5
10nF
NO LOAD
CAPSENSE TUNING CIRCUITRY
Default Loaded For CSD

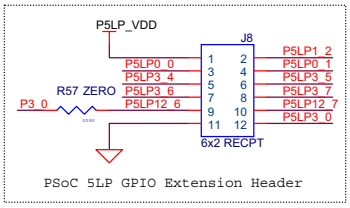

NO LOAD
R46 ZERO
V3.3_EXT
Q1
PMOS( DMP3098L-7)
V3.3
R43
442 ohm
Q2
Q3
PMOS( DMP3098L-7)
PMOS( DMP3098L-7)
D5
Vz=2V(BZT52C2V0-7-F)
R42
1K ohm
Protection Circuit

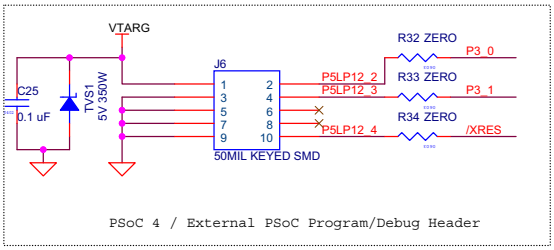

SW1
/XRES
EVQ-PE105K
RESET
J14
2 PIN HDR
P4_VDD
LED_PWR
LED3
R28 2.2K
P3_2
LED_PWR
R29 1.5K
P1_1
R30 1.5K
P0_2
RGB LED
P5LP_VDD
R31
LED2
P5LP3_1
330 ohm
Status LED Green
User Interface

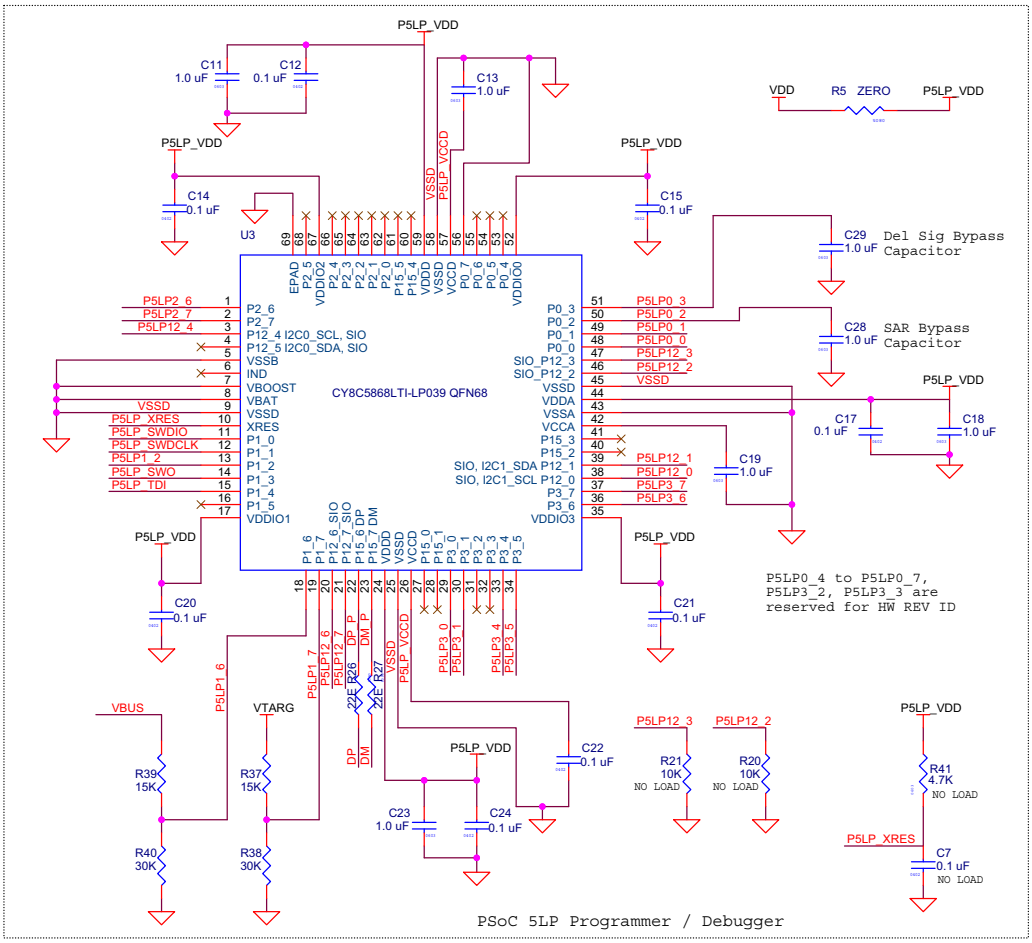

J12
P3_1
P1_7
/XRES
P1_1
3x2 RECPT
NO LOAD
J12 Arduino ICSP
compatible header
P2_0
J5
1x1 RECP
Proximity Header
P4_VDD
R8 2.2K
R9 2.2K
I2C Pull up
Resistors
J3
P1_2
P1_3
P1_7
P3_1
P1_1
P1_6
P1_5
P1_4
10X1 RECP
VIN
J1
VBUS
D4
V3.3_EXT
/XRES
IOREF
P4_VDD
8x1 RECP
J2
P0_0
P0_1
P0_2
P0_4
P1_3
P1_2
6x1 RECP
J4
P0_5
P0_6
P0_7
P3_2
P0_3
P3_0
P1_0
P2_0 R54
ZERO
8x1 RECP
(J1-J4) Arduino Compatible Headers

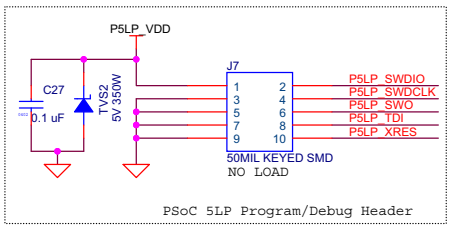

NO LOAD
TP2 RED
VBUS
USB MiniB
F1
PTC Resettable Fuse
J10
VBUS
DM
DP
ID
GND
S2
S1
USB MINI B
DM
DP
D6
D7
D8
100K R13
C16 0.01 uF

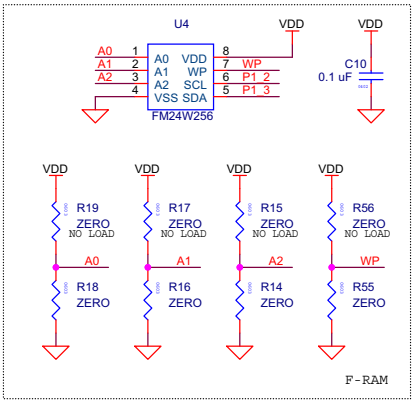

P5LP2_6
P5LP2_7
R22 2.2K
NO LOAD
R23 2.2K
NO LOAD
R24
ZERO
P5LP12_1
P1_3
R25
ZERO
P5LP12_0
P1_2
I2C Connection b/w PSoC 5LP and PSoC 4


P5LP_VDD
J8
P5LP0_0
P5LP3_4
R57 ZERO
P5LP3_6
P3_0
P5LP12_6
P5LP1_2
P5LP0_1
P5LP3_5
P5LP3_7
P5LP12_7
P5LP3_0
6x2 RECPT
PSoC 5LP GPIO Extension Header

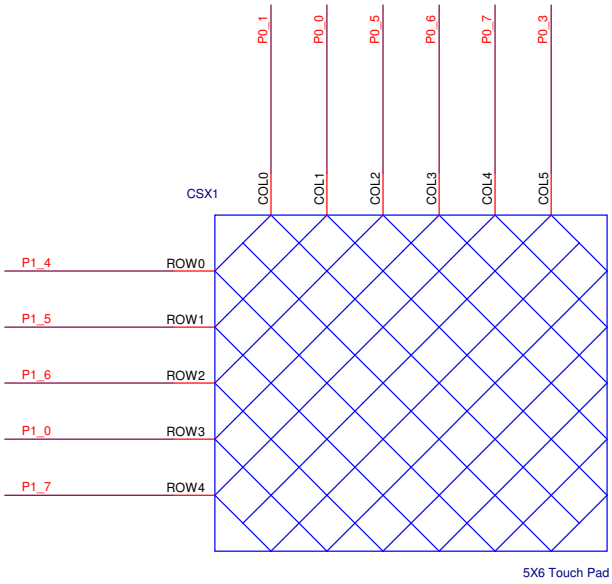

VTARG
C25
0.1 uF
TVS1
5V 350W
J6
P5LP12_2
P5LP12_3
P5LP12_4
R32 ZERO
R33 ZERO
R34 ZERO
P3_0
P3_1
/XRES
50MIL KEYED SMD
PSoC 4 / External PSoC Program/Debug Header


P5LP_VDD
C11
1.0 uF
C12
0.1 uF
C13
1.0 uF
VDD
R5 ZERO
P5LP_VDD
C14
0.1 uF
U3
C15
0.1 uF
C29
1.0 uF
Del Sig Bypass Capacitor
C28
1.0 uF
SAR Bypass Capacitor
CY8C5868LTI-LP039 QFN68
C17
0.1 uF
C18
1.0 uF
C19
1.0 uF
C20
0.1 uF
C21
0.1 uF
P5LP0_4 to P5LP0_7, P5LP3_2, P5LP3_3 are reserved for HW REV ID
VBUS
VTARG
R39
15K
R37
15K
R40
30K
R38
30K
C22
0.1 uF
C23
1.0 uF
C24
0.1 uF
R21
10K
NO LOAD
R20
10K
NO LOAD
R41
4.7K
NO LOAD
C7
0.1 uF
NO LOAD
PSoC 5LP Programmer / Debugger

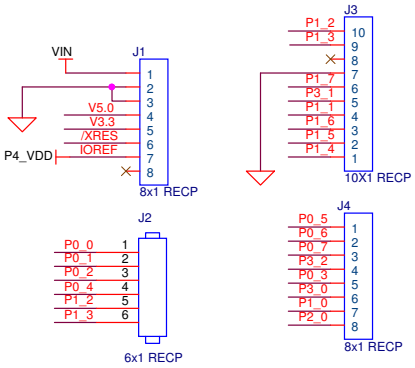

P5LP_VDD
C27
0.1 uF
TVS2
5V 350W
J7
P5LP_SWDIO
P5LP_SWDCLK
P5LP_SWO
P5LP_TDI
P5LP_XRES
50MIL KEYED SMD
NO LOAD
PSoC 5LP Program/Debug Header




## A.1.1 CapSense 触控扩展电路板




CapSense Touch Pad




(J1-J4) Arduino Compatible Headers

## A.2 引脚分配表

该部分提供了插座的引脚映射及其使用情况

### A.2.1 Arduino 兼容的插座 （J1、 J2、 J3、 J4 及 J12）

| 电源连接器 （J1） | | |
|---|---|---|
| 引脚 | 主板信号 | 触控扩展板信号 |
| J1_01 | VIN | NC |
| J1_02 | GND | GND |
| J1_03 | GND | GND |
| J1_04 | V5.0 | NC |
| J1_05 | V3.3 | NC |
| J1_06 | RESET | NC |
| J1_07 | IOREF/P4_VDD | NC |
| J1_08 | NC | NC |

| J2 连接器 | | |
|---|---|---|
| 引脚 | 主板信号 | 触控扩展板信号 |
| J2_01 (A0) | P0[0] | TRACK_COLUMN1 |
| J2_02 (A1) | P0[1] | TRACK_COLUMN0 |
| J2_03 (A2) | P0[2] (TCPWM_LINE/ 蓝色 LED、CTANK) | NC |
| J2_04 (A3) | P0[4] (CMOD) | NC |
| J2_05 (A4) | P1[3] (SDA) | NC |
| J2_06 (A5) | P1[2] (SCL) | NC |

| J3 连接器 | | |
|---|---|---|
| 引脚 | 主板信号 | 触控扩展板信号 |
| J3_01 (D8) | P1[4] | TRACK_ROW0 |
| J3_02 (D9) | P1[5] | TRACK_ROW1 |
| J3_03 (D10) | P1[6] | TRACK_ROW2 |
| J3_04 (D11) | P1[1] (TCPWM_LINE/ 绿色 LED) | NC |
| J3_05 (D12) | P3[1] (SWDCK) | NC |
| J3_06 (D13) | P1[7] | TRACK_ROW4 |
| J3_07 | GND | GND |
| J3_08 | NC | NC |
| J3_09 | P1[3] (SDA) | NC |
| J3_10 | P1[2] (SCL) | NC |

| J4 连接器 | | |
|---|---|---|
| 引脚 | 主板信号 | 触控扩展板信号 |
| J4_01 (D0) | P0[5] | TRACK_COLUMN2 |
| J4_02 (D1) | P0[6] | TRACK_COLUMN3 |
| J4_03 (D2) | P0[7] | TRACK_COLUMN4 |
| J4_04 (D3) | P3[2] (TCPWM_LINE，红色 LED) | NC |
| J4_05 (D4) | P0[3] | TRACK_COLUMN5 |
| J4_06 (D5) | P3[0] (SWDIO) | NC |
| J4_07 (D6) | P1[0] | TRACK_ROW3 |
| J4_08 (D7) | P2[0] (PROX) | NC |

| J12 | | |
|---|---|---|
| 引脚 | 套件信号 | PSoC 4 说明 |
| J12_01 | P3[1] | GPIO |
| J12_02 | NC | NC |
| J12_03 | P1[7] | GPIO |
| J12_04 | P1[1] | GPIO |
| J12_05 | /XRES | PSoC 4 RESET |
| J12_06 | GND | GND |

### A.2.2 PSoC 5LP GPIO 插座 （J8）

J8 是 2 x 6 的插座，用于连接 PSoC 5LP 的各个引脚，以便支持自定义 PSoC5LP 项目的 GPIO 控制。

| J8 | | | | | |
|---|---|---|---|---|---|
| **引脚** | **PSoC 5LP 信号** | **PSoC 5LP 说明** | **引脚** | **PSoC 5LP 信号** | **PSoC 5LP 说明** |
| J8_01 | PSoC 5LP_VDD | VDD | J8_02 | P1[2] | 数字 I/O |
| J8_03 | P0[0] | Delta Sigma ADC + 输入 | J8_04 | P0[1] | Delta Sigma ADC – 输入 |
| J8_05 | P3[4] | SAR – 输入 | J8_06 | P3[5] | SAR + 输入 |
| J8_07 | P3[6] | 缓存 VDAC | J8_08 | P3[7] | 缓存 VDAC |
| J8_09 | P12[6] | UART RX | J8_10 | P12[7] | UART TX |
| J8_11 | GND | GND | J8_12 | P3[0] | IDAC 输出 |

## A.3 编程和调试插座

### A.3.1 PSoC 4 直接编程 / 调试插座 （J6）

| J6 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **引脚** | **PSoC 5LP 信号** | **PSoC 4 信号** | **说明** | **引脚** | **PSoC 5LP 信号** | **PSoC 4 信号** | **说明** |
| J6_01 | VDD | VDD | VCC | J6_02 | P12[2] | P3[0] | TMS/SWDIO |
| J6_03 | GND | GND | GND | J6_04 | P12[3] | P3[1] | TCLK/SWCLK |
| J6_05 | GND | GND | GND | J6_06 | NC | NC | NC |
| J6_07 | GND | GND | GND | J6_08 | NC | NC | NC |
| J6_09 | GND | GND | GND | J6_10 | P12[4] | XRES | RESET |

### A.3.2 PSoC 5LP 直接编程 / 调试插座 （J7）

| J7 | | | | | |
|---|---|---|---|---|---|
| **引脚** | **PSoC 5LP 信号** | **说明** | **引脚** | **PSoC 5LP 信号** | **说明** |
| J7_01 | VDD | VCC | J7_02 | P1[0] | TMS/SWDIO |
| J7_03 | GND | GND | J7_04 | P1[1] | TCLK/SWCLK |
| J7_05 | GND | GND | J7_06 | P1[3] | TDO/SWO |
| J7_07 | GND | GND | J7_08 | P1[4] | TDI |
| J7_09 | GND | GND | J7_10 | XRES | RESET |

## A.4 0 Ω 电阻和空载的使用

| 单元 | 电阻 | 用法 |
|---|---|---|
| 电源 | R2 | 通过焊接大小为 0 Ω 的电阻来获取 VBUS （USB）的电压。 |
| PSoC 5LP 和 PSoC 4 间的 $I^2C$ 连接 | R24 和 R25 | 拆除这些电阻，以便使用 PSoC 5LP 同外部 PSoC 进行通信。移除这两个电阻会禁用 PSoC 4 同 PSoC 5LP 器件间的 $I^2C$ 通信接口。 |
| PSoC 4/ 外部 PSoC 编程 / 调试插座 | R32、R33 和 R34 | 拆除这些电阻，以断开 SWD 线与 PSoC 4 的连接。使用 J6 进行连接并编程外部 PSoC。移除这些电阻时，会禁用由 PSoC 5LP 器件通过 J6 插座对 PSoC 4 进行的编程。 |
| 保护电路 | R46 | 拆除大小为 0 Ω 的电阻，以旁路整个保护电路。 |
| CapSense 调试电路 | R1 | 将 IDAC 输出转换为电压，或将其作为 CMOD 的放电电阻使用。 |
| CapSense 调试电路 / 用户界面 | R30 | 拆除 R30 电阻，这样能够将 P0[2] 连接到蓝色的 LED，以便使能 P0[2] 引脚上的屏蔽水库电容 C5。 |
| PSoC 4 | R4、 R6 | 拆除电阻 R4，以便关闭供给 VTARG 的电源； 焊接大小为 0 Ω 的电阻 R6，可通过使用 VDD （而不是使用 J13）供电给 P4_VDD。 |
| PSoC 5LP 编程器 / 调试器 | R11、 R12、 R14、R15、R16 | 预留以供将来使用 |
| | R5 | 拆除大小为 0 Ω 的电阻，以关闭供给 PSoC 5LP 的 VDD 电源。 |
| | R7 | 保留给将来使用 |
| F-RAM | R14、 R15、 R16、 R17、 R18、 R19 | 选择 F-RAM $I^2C$ 从设备地址的三个低位。 R14-R15 可选择位 2（A2），R16-R17 可选择位 1（A1），R18-R19 可选择位 0（A0）。将所选的各位与 F-RAM 系列的 $I^2C$ 地址（0x50）进行 “OR”（或）运算，用以决定每一位在板上的从设备地址。 |
| | R56、 R55 | 为 R56 焊接一个大小为 0 Ω 的电阻，用以对整个 F-RAM 存储器进行写保护。由于 WP 引脚被内部下拉，因此无需安装 R55。当 WP 引脚保持悬空状态或连接了R55时，将恢复对F-RAM进行的写访问。 |
| PSoC 5LP GPIO 插座 | R57 | 拆除大小为 0 Ω 的电阻，可断开 P3[0] 与 PSoC 5LP RX 线的连接，并使用 P3[0] 来调试 PSoC 4 或将 PSoC 5LP 作为其他器件的 USB-UART 桥接器使用。 |

## A.5 KitProg 的 LED 状态

| | 用户指示 | 方案 | 用户需要执行的操作 |
|---|---|---|---|
| 1 | LED 快速闪烁：<br>频率 = 4.00 Hz | 如果 Bootloadable 文件被损坏，LED 在上电时会闪烁发光。 | 引导加载 *KitProg.cyacd* 文件：在 PSoC Programmer 内，连接至该套件，打开 **Utilities** 选项卡并点击 **Upgrade Firmware**（升级固件）按键。 |
| 2 | LED 慢速闪烁：<br>频率 = 0.67 Hz | 在套件上电期间按下BLE Pioneer 套件 /BLE 收发器 Reset（复位）按键可进入 Bootloader 模式。 | 如果您错误地进入该模式，请放开复位按键并重新接通电源。如果有意进入该模式，请使用 PSoC Creator 所提供的 Bootloader 主机工具下载 new.cyacd 文件。 |
| 3 | LED 高速闪烁：<br>频率 = 15.0 Hz | 正在执行 SWD 操作。<br>所有 I2C 通信。<br>套件的 COM 端口连接 / 断连事件（闪烁一次）。 | 在 PSoC Programmer 中，在日志窗口上监视 SWD 操作的状态信息。在桥接控制面板中，在收到指令请求时，LED 将闪烁。在 BCP 或任何其他串行端口终端程序内，通过在端口连接或断开连接时闪烁 LED 来区分该套件的 COM 端口编号。 |
| 4 | LED 为 ON 状态。 | USB 成功枚举。<br>该套件处于闲置状态等待各条指令。 | 该套件的各项功能可被 PSoC Creator、PSoC Programmer、桥接控制面板和任何串行端口终端程序使用。 |
| 5 | LED 为 OFF 状态。 | 电源 LED 处于 ON 状态。 | 这意味着 USB 枚举失败。如果 USB 主机未给该套件供电或该套件未通过 USB 线缆连接至 USB 主机，将发生这种情况。这时，需要验证 USB 线缆并检查 PC 上是否安装了 PSoC Programmer。 |

**注意：**如果 PSoC 5LP 中正在运行自定义项目，那么 LED 状态不可用。

# A.6 材料清单

## A.6.1 CY8CKIT-040 主板

| 序号 | 数量 | 参考 | 数值 | 说明 | 制造商 | 制造商器件型号 |
|---|---|---|---|---|---|---|
| 1 | | | | PCB， 68.58 mm x 53.34 mm，高熔点（Tg），表面处理采用化学镍金（ENIG），四层，颜色为红色，丝印为白色。 | Cypress | |
| 2 | 1 | C1 | 2200 pFd | CAP CER 2200PF 50V 5% NP0 0805 | Murata | GRM2165C1H222JA01D |
| 3 | 13 | C2、 C9、 C10、 C12、 C14、 C15、 C17、 C20、 C21、 C22、 C24、 C25、 C27 | 0.1 uFd | CAP .1UF 16V CERAMIC Y5V 0402 | Panasonic - ECG | ECJ-0EF1C104Z |
| 4 | 9 | C3、C11、C13、 C18、 C19、 C23、 C26、 C28、 C29 | 1.0 uFd | CAP CER 1.0UF 25V X5R 0603 10% | Taiyo Yuden | TMK107BJ105KA-T |
| 5 | 1 | C4 | 10 uF 25V | CAP TANT 10UF 25V 10% 1210 | AVX Corporation | TPSB106K025R1800 |
| 6 | 1 | C6 | 22 uF 16V | CAP TANT 22UF 16V 10% 1210 | AVX Corporation | TPSB226K016R0600 |
| 7 | 1 | C16 | 0.01 uFd | CAP 10000PF 16V CERAMIC 0402 SMD | Panasonic - ECG | ECJ-0EB1C103K |
| 8 | 1 | C8 | 10 uFd | CAP CER 10UF 6.3 V 20% X5R 0603 | Samsung Electro-Mechanics America, Inc | CL10A106MQ8NNNC |
| 9 | 7 | D1、 D2、 D3、 D4、 D9、 D10、 D11 | MBR05 | DIODE SCHOTTKY 0.5A 20 V SOD-123 | Fairchild Semiconductor | MBR0520L |
| 10 | 1 | LED1 | 琥珀色的 LED | LED 595NM AMB DIFF 0805 SMD | Avago Technologies | HSMA-C170 |
| 11 | 1 | D5 | 2 V 的稳压二极管 | DIODE ZENER 2V 500MW SOD123 | Diodes Inc | BZT52C2V0-7-F |
| 12 | 3 | D6、 D7、 D8 | ESD 二极管 | SUPPRESSOR ESD 5VDC 0603 SMD | Bourns Inc. | CG0603MLC-05LE |
| 13 | 1 | LED3 | RGB LED | LED RED/GREEN/BLUE PLCC4 SMD | Cree, Inc. | CLV1A-FKB-CJ1M1F1BB7R4S3 |
| 14 | 1 | LED2 | 绿色状态 LED | LED GREEN CLEAR 0805 SMD | Chicago Miniature | CMD17-21VGC/TR8 |
| 15 | 1 | F1 | FUSE | PTC 可恢复的熔断器，其电压为 15 V，电流为 100 A | Bourns | MF-MSMF050-2 |
| 16 | 2 | J1、 J4 | 8x1 RECP | CONN HEADER FEMALE 8POS .1" GOLD | Sullins Connector Solutions | PPPC081LFBN-RC |
| 17 | 1 | J2 | 6x1 RECP | CONN HEADER FMAL 6POS.1" GOLD | Sullins Connector Solutions | PPPC061LFBN-RC |
| 18 | 1 | J3 | 10x1 RECP | CONN HEADER FMALE 10POS .1" GOLD | Sullins Connector Solutions | PPPC101LFBN-RC |
| 19 | 1 | J5 | 1X1 RECP | CONN RCPT 1POS .100" SNGL HORZ | Samtec Inc | BCS-101-L-S-HE |
| 20 | 1 | J6 | 50MIL KEYED SMD | CONN HEADER 10 PIN 50MIL KEYED SMD | Samtec | FTSH-105-01-L-DV-K |
| 21 | 1 | J8 | 6x2 RECP | CONN HEADER FMAL 12PS.1" DL GOLD | Sullins Connector Solutions | PPPC062LFBN-RC |

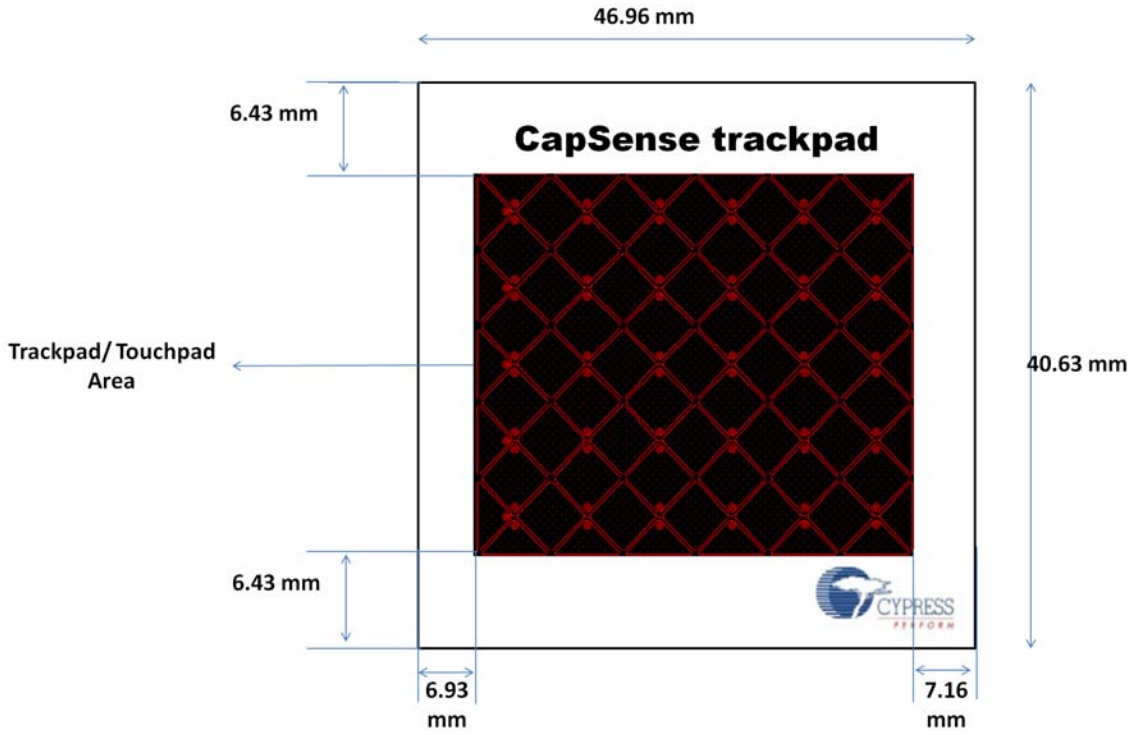
| 序号 | 数量 | 参考 | 数值 | 说明 | 制造商 | 制造商器件型号 |
|---|---|---|---|---|---|---|
| 22 | 1 | J9 | 3p_jumper | CONN HEADER VERT SGL 3POS GOLD | 3M | 961103-6404-AR |
| 23 | 1 | J10 | USB Mini B | CONN USB RECEPTACLE 5POS RT ANG | Molex Inc | 0548190519 |
| 24 | 2 | J13,J14 | 2p_jumper | CONN HEADER VERT SGL 2POS GOLD | 3M | 961102-6404-AR |
| 25 | 3 | Q1、 Q2、 Q3 | PMOS | MOSFET P-CH 30V 3.8A SOT23-3 | Diodes Inc | DMP3098L-7 |
| 26 | 1 | R3 | 560 Ω | RES 560 OHM 1/8W 5% 0805 SMD | Panasonic - ECG | ERJ-6GEYJ561V |
| 27 | 24 | R4、 R7、 R11、 R12、 R14、 R16、 R18、 R24、 R25、 R32、 R33、 R34、 R44、 R45、 R47、 R48、 R49、 R50、 R51、 R52、 R53、 R54、 R55、 R57 | 0 | RES 0.0 OHM 1/10W 0603 SMD | Panasonic-ECG | ERJ-3GEY0R00V |
| 28 | 1 | R5 | 0 | RES 0.0 OHM 1/8W 0805 SMD | Panasonic-ECG | ERJ-6GEY0R00V |
| 29 | 4 | R8、 R9 | 2.2K | RES 2.2K OHM 1/10W 5% 0603 SMD | Panasonic - ECG | ERJ-3GEYJ222V |
| 30 | 1 | R13 | 100K | RES 100K OHM 1/10W 5% 0402 SMD | Panasonic - ECG | ERJ-2GEJ104X |
| 31 | 2 | R26、 R27 | 22E | RES 22 OHM 1/10W 1% 0603 SMD | Panasonic - ECG | ERJ-3EKF22R0V |
| 32 | 1 | R28 | 2.2K | RES 2.2K OHM 1/8W 5% 0805 SMD | Panasonic - ECG | ERJ-6GEYJ222V |
| 33 | 2 | R29、 R30 | 1.5K | RES 1.5K OHM 1/8W 5% 0805 SMD | Panasonic - ECG | ERJ-6GEYJ152V |
| 34 | 1 | R31 | 330 Ω | RES 330 OHM 1/8W 5% 0805 SMD | Panasonic - ECG | ERJ-6GEYJ331V |
| 35 | 1 | R35 | 232 Ω | RES 232 OHM 1/10W 1% 0603 SMD | Panasonic - ECG | ERJ-3EKF2320V |
| 36 | 1 | R36 | 120 Ω | RES 120 OHM 1/10W 1% 0603 SMD | Panasonic - ECG | ERJ-3EKF1200V |
| 37 | 2 | R37、 R39 | 15K | RES 15K OHM 1/10W 5% 0603 SMD | Panasonic - ECG | ERJ-3GEYJ153V |
| 38 | 2 | R38、 R40 | 30K | RES 3.0K OHM 1/10W 5% 0603 SMD | Panasonic - ECG | ERJ-3GEYJ303V |
| 39 | 1 | R42 | 1K | RES 1K OHM 1/8W 5% 0805 SMD | Panasonic - ECG | ERJ-6GEYJ102V |
| 40 | 1 | R43 | 442 Ω | RES 442 OHM 1/10W 1% 0603 SMD | Panasonic - ECG | ERJ-3EKF4420V |
| 41 | 1 | SW1 | SW PUSHBUTTON | SWITCH TACTILE SPST-NO 0.05A 12V | Panasonic - ECG | EVQ-PE105K |
| 42 | 1 | TP5 | BLACK | TEST POINT PC MINI .040"D Black | Keystone Electronics | 5001 |
| 43 | 2 | TVS1、 TVS2 | 5V 350W | TVS UNIDIR 350W 5V SOD-323 | Dioded Inc | SD05-7 |
| 44 | 1 | U1 | NCP1117DTARKG | NCP1117DTARKG | ON Semiconductor | NCP1117DTARKG |
| 45 | 1 | U2 | PSoC 4 S0 (CY8C400) | 24 QFN PSoC4 S0 目标芯片 | Cypress Semiconductor | CY8C4014LQI-422 |
| 46 | 1 | U3 | PSoC 5LP (CY8C5868LTI-LP039) | 用于通过 USB 调试通道和 USB 串行接口进行调试的 68QFN PSoC 5LP 芯片 | Cypress Semiconductor | CY8C5868LTI-LP039 |
| 47 | 1 | U4 | F-RAM | 提供 $I^2C$ 接口的 F-RAM | Cypress Semiconductor | FM24W256-G |
| **空载组件** | | | | | | |
| 48 | 1 | BT1 | 纽扣电池座 | HOLDER CR2032 GOLD LEADS SMD | MPD | BU2032SM-BT-GTR |
| 49 | 1 | C5 | 10000 pFd | CAP CER 10000PF 50V 5% NP0 0805 | Murata | GRM2195C1H103JA01D |

| 序号 | 数量 | 参考 | 数值 | 说明 | 制造商 | 制造商器件型号 |
|---|---|---|---|---|---|---|
| 50 | 2 | C7,C30 | 0.1 uFd | CAP .1UF 16V CERAMIC Y5V 0402 | Panasonic - ECG | ECJ-0EF1C104Z |
| 51 | 1 | J7 | 50MIL KEYED SMD | CONN HEADER 10 PIN 50MIL KEYED SMD | Samtec | FTSH-105-01-L-DV-K |
| 52 | 1 | J11 | 2 PIN HDR | CONN HEADER FEMALE 8POS .1" GOLD | Sullins Connector Solutions | PPPC021LFBN-RC |
| 53 | 1 | J12 | 3X2 RECP | CONN HEADER .100 DUAL STR 12POS | Sullins Connector Solutions | PBC06DFAN |
| 54 | 7 | R1、 R2、 R15、 R17、 R19、 R46、 R56 | 0 | RES 0.0 OHM 1/10W 0603 SMD | Panasonic-ECG | ERJ-3GEY0R00V |
| 55 | 1 | R6 | 0 | RES 0.0 OHM 1/8W 0805 SMD | Panasonic-ECG | ERJ-6GEY0R00V |
| 56 | 2 | R10、 R41 | 4.7K | RES 4.7K OHM 1/10W 5% 0603 SMD | Panasonic-ECG | ERJ-3GEYJ472V |
| 57 | 2 | R20,R21 | 10K | RES 10K OHM 1/10W 5% 0603 SMD | Panasonic - ECG | ERJ-3GEYJ103V |
| 58 | 2 | R22,R23 | 2.2K | RES 2.2K OHM 1/10W 5% 0603 SMD | Panasonic - ECG | ERJ-3GEYJ222V |
| 59 | 3 | TP1、TP2、TP6 | RED | TEST POINT PC MINI .040"D RED | Keystone Electronics | 5000 |
| 60 | 2 | TP3、 TP4 | BLACK | TEST POINT PC MINI .040"D Black | Keystone Electronics | 5001 |
| **按照拐角中字符层来安装 PCB 底部** | | | | | | |
| 61 | 4 | N/A | N/A | BUMPON CYLINDRICAL .312X.215 BLACK | 3 M | SJ61A6 |
| **特殊跳线的安装说明** | | | | | | |
| 62 | 1 | J9 | 安装引脚 1 和 2 的跳线器 | 矩形连接器 MINI JUMPER GF 6.0MM CLOSE TYPE BLACK | Kobiconn | 151-8010-E |
| 63 | 1 | J13 | 安装引脚 1 和 2 的跳线器 | 矩形连接器 MINI JUMPER GF 6.0MM CLOSE TYPE BLACK | Kobiconn | 151-8010-E |
| 64 | 1 | J14 | 安装引脚 1 和 2 的跳线器 | 矩形连接器 MINI JUMPER GF 6.0MM CLOSE TYPE BLACK | Kobiconn | 151-8010-E |

## A.6.2 CY8CKIT-040 触控扩展板

| 序号 | 数量 | 参考 | 数值 | 说明 | 制造商 | 制造商器件型号 |
|---|---|---|---|---|---|---|
| 1 | | | | PCB，53.34 mm x 53.34 mm，高熔点（Tg），表面处理采用化学镍金（ENIG），两层，颜色为红色，丝印为白色。 | 赛普拉斯 | |
| 2 | 2 | J1、 J4 | CON8 | CONN HEADER 8POS .100 STR 30AU | FCI | 68001-108HLF |
| 3 | 1 | J2 | CON6 | CONN HEADER 8POS .100 STR 30AU | FCI | 68001-106HLF |
| 4 | 1 | J3 | CON10 | CONN HEADER 10POS .100 STR 30AU | FCI | 68001-110HLF |

## A.7 触控板标签详情




## A.8 法规遵从信息

CY8CKIT-040 PSoC 4000 已经过测试和验证，符合下列电磁兼容性 （EMC）规定：

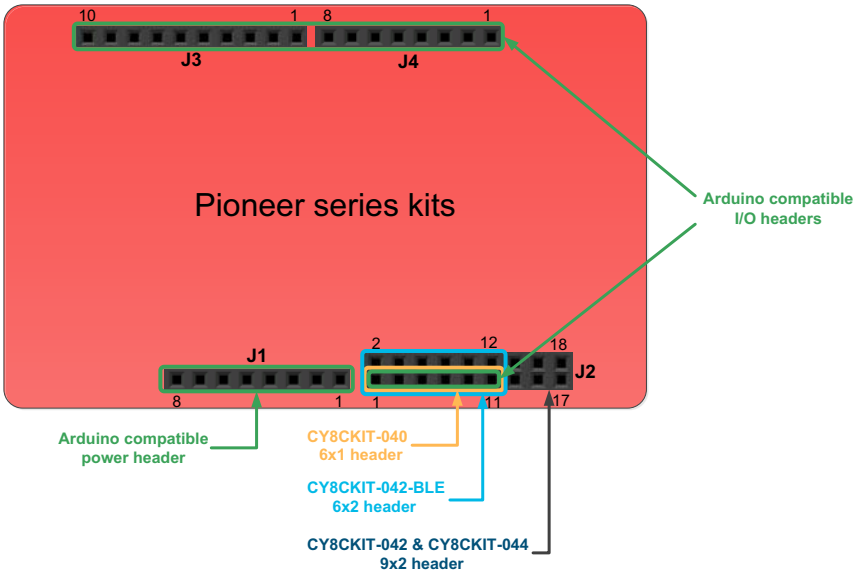
- EN 55022:2010 A 级 — 辐射
- EN 55024:2010 A 级 — 抗干扰

## A.9 在不同的 Pioneer 系列套件间移植项目

赛普拉斯提供的所有 Pioneer 系列套件都是 Arduino Uno 兼容的，并拥有一些通用的板载外设（如 RGB LED、CapSense 和用户开关）。但是由于 PSoC 器件所使用的引脚功能不完全一样，因此各个电路板上的引脚映射情况也有所不同。本指南列出了 Pioneer 系列套件的引脚映射情况，以便在不同的套件上可以轻松进行项目移植。

在某些情况下， Pioneer 套件插座上可用的引脚是标准 Arduino Uno 引脚的超集。例如，在 Arduino Uno 引脚分布上， J2 仅包含一行引脚，而在诸多的 Pioneer 系列套件上，它包含了两行引脚。

图 A-1. Pioneer 系列套件引脚映射情况



### A.9.1 Arduino Uno 兼容的插座

| J1 Arduino 兼容的插座引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | Arduino 引脚 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | VIN | VIN | VIN | VIN | VIN |
| 2 | GND | GND | GND | GND | GND |
| 3 | GND | GND | GND | GND | GND |
| 4 | 5 V | V5.0 | V5.0 | V5.0 | V5.0 |
| 5 | 3.3 V | V3.3 | V3.3 | V3.3 | V3.3 |
| 6 | RESET | RESET | RESET | RESET | RESET |
| 7 | IOREF | P4.VDD | P4.VDD | BLE.VDD | P4.VDD |
| 8 | NC | NC | NC | NC | NC |

| J2 Arduino 兼容的插座引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | Arduino 引脚 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | A0 | P2[0] | P0[0] | P3[0] | P2[0] |
| 2 | – | P0[2]* | – | P2[0] | P2[6]* |
| 3 | A1 | P2[1] | P0[1] | P3[1] | P2[1] |
| 4 | – | P0[3]* | – | P2[1]* | P6[5]* |
| 5 | A2 | P2[2] | P0[2]* | P3[2] | P2[2] |
| 6 | – | P4_VDD | – | P2[2]* | P0[6]* |
| 7 | A3 | P2[3] | P0[4]* | P3[3] | P2[3] |
| 8 | – | P1[5]* | – | P2[3]* | P4[4]* |
| 9 | A4 | P2[4] | P1[3] | P3[4] | P2[4] |
| 10 | – | P1[4]* | – | P2[4]* | P4[5]* |
| 11 | A5 | P2[5] | P1[2] | P3[5] | P2[5] |
| 12 | – | P1[3]* | – | P2[5]* | P4[6]* |
| 13 | – | P0[0] | – | – | P0[0] |
| 14 | – | GND | – | – | GND |
| 15 | – | P0[1] | – | – | P0[1] |
| 16 | – | P1[2]* | – | – | P3[4]* |
| 17 | – | P1[0] | – | – | P0[7]* |
| 18 | – | P1[1]* | – | – | P3[5]* |

这些引脚还可以用于板载外设。更多详细信息，请参考第 174 页上的“板载外设”一节部分中的各表。

| J3 Arduino 兼容的插座引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | Arduino 引脚 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | D8 | P2[6] | P1[4] | P0[5] | P0[2] |
| 2 | D9 | P3[6] | P1[5] | P0[4] | P0[3] |
| 3 | D10 | P3[4] | P1[6] | P0[2] | P2[7] |
| 4 | D11 | P3[0] | P1[1]* | P0[0] | P6[0] |
| 5 | D12 | P3[1] | P3[1] | P0[1] | P6[1] |
| 6 | D13 | P0[6] | P1[7] | P0[3] | P6[2] |
| 7 | GND | GND | GND | GND | GND |
| 8 | AREF | P1[7] | NC | VREF | P1[7] |
| 9 | SDA | P4[1] | P1[3] | P3[4] | P4[1] |
| 10 | SCL | P4[0] | P1[2] | P3[5] | P4[0] |

这些引脚还可以用于板载外设。更多有关连接的详细信息，请参考第 174 页上的“板载外设”一节中各表格的内容。

| J4 Arduino 兼容的插座引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | Arduino 引脚 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | D0 | P0[4] | P0[5] | P1[4] | P3[0] |
| 2 | D1 | P0[5] | P0[6] | P1[5] | P3[1] |
| 3 | D2 | P0[7]* | P0[7] | P1[6] | P1[0] |
| 4 | D3 | P3[7] | P3[2]* | P1[7] | P1[1] |
| 5 | D4 | P0[0] | P0[3] | P1[3] | P1[2] |
| 6 | D5 | P3[5] | P3[0] | P1[2] | P1[3] |
| 7 | D6 | P1[0] | P1[0] | P1[1] | P5[3] |
| 8 | D7 | P2[7] | P2[0]* | P1[0] | P5[5] |

这些引脚还可以用于板载外设。更多有关连接的详细信息，请参考第 174 页上的“板载外设”一节中各表格的内容。

## A.9.2 板载外设

| CapSense 引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | 说明 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042（线性滑条） | CY8CKIT-040 | CY8CKIT-042-BLE（线性滑条） | CY8CKIT-044（手势板） |
| 1 | CSS1 | P1[1] | – | P2[1] | P4[4] |
| 2 | CSS2 | P1[2] | – | P2[2] | P4[5] |
| 3 | CSS3 | P1[3] | – | P2[3] | P4[6] |
| 4 | CSS4 | P1[4] | – | P2[4] | P3[4] |
| 5 | CSS5 | P1[5] | – | P2[5] | P3[5] |
| 6 | CMOD | P4[2] | P0[4] | P4[0] | P4[2] |
| 7 | CTANK | P4[3] | P0[2] | P4[1] | P4[3] |

| 接近感应插座引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | 说明 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | 接近 | – | P2[0] | P2[0] | P3[7] |
| 2 | | – | – | – | P3[6] |

| RGB 三色 LED 引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | 颜色 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | 红色 | P1[6] | P3[2] | P2[6] | P0[6] |
| 2 | 绿色 | P0[2] | P1[1] | P3[6] | P2[6] |
| 3 | 蓝色 | P0[3] | P0[2] | P3[7] | P6[5] |

| 用户开关引脚映射 | | | | | |
|---|---|---|---|---|---|
| 引脚编号 | 说明 | Pioneer 系列套件 | | | |
| | | CY8CKIT-042 | CY8CKIT-040 | CY8CKIT-042-BLE | CY8CKIT-044 |
| 1 | SW2 | P0[7] | – | P2[7] | P0[7] |

# 修订记录

## 文档修订记录

| 文档标题：CY8CKIT-040 PSoC® 4000 Pioneer 套件指南<br>文档编号：001-92737 | | | |
|---|---|---|---|
| **版本** | **ECN 编号** | **发布日期** | **变更说明** |
| ** | 4396040 | 06/02/2014 | 本文档版本号为 Rev**，译自英文版 001-91316 Rev**。 |
| *A | 5898185 | 09/27/2017 | 更新到新模板。<br>完成日落评论。 |
| *B | 6651870 | 04/06/2020 | 本文档版本号为 Rev*B，译自英文版 001-91316 Rev*F。 |