



## CY15FRAMKIT-001

# 串行 F-RAM™ 开发套件指南

文档编号：001-97577 版本\*\*

赛普拉斯半导体公司

198 Champion Court

San Jose, CA 95134 USA

电话（美国）：+1.800.858.1810

电话（国际）：+1.408.943.2600

[www.cypress.com](http://www.cypress.com)

## 版权所有

©赛普拉斯半导体公司，2015。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯不保证产品能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

## 商标

PSoC 是赛普拉斯半导体公司的注册商标且 PSoC Creator 和 F-RAM 是赛普拉斯半导体公司的商标。此处引用的所有其他商标或注册商标均归其各自所有者所有。

## 源代码

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途外，未经赛普拉斯明确的书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

## 免责声明

赛普拉斯不对此材料提供任何类型的明示或暗示保证，包括但不限于针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障，并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用的赛普拉斯软件许可协议限制。

# 目录



安全信息 .....	5
法规遵从 .....	5
通用安全说明 .....	6
1. 简介 .....	7
1.1 套件内容 .....	7
1.2 PSoC Creator™ .....	8
1.3 入门指南 .....	8
1.4 其他参考资源 .....	8
1.5 技术支持 .....	8
1.6 文档规范 .....	8
2. 套件安装 .....	9
2.1 必备软件 .....	9
2.2 安装套件软件 .....	9
2.3 卸载软件 .....	11
3. 套件概述 .....	12
3.1 CY15FRAMKIT-001 串行 F-RAM 开发套件 概述 .....	12
3.2 套件操作与配置指南 .....	13
3.2.1 电源跳线器 .....	13
3.2.2 F-RAM 器件 .....	13
3.2.3 用于 I <sup>2</sup> C F-RAM 从设备地址选择的 DIP 开关 .....	13
3.2.4 用于连接到 CY8CKIT-042/Arduino UNO R3 电路板的连接器 .....	14
3.2.5 调试插座 .....	15
3.2.6 测试点 .....	15
3.3 CY15FRAMKIT-001 与 PSoC 4 Pioneer 套件 .....	15
3.4 CY15FRAMKIT-001 与 Arduino UNO R3 套件 .....	16
4. 硬件 .....	17
4.1 电路板的详细信息 .....	17
4.2 操作原理 .....	19
4.3 功能说明 .....	19
4.3.1 SPI F-RAM 器件 (256 Kbit FM25W256) .....	19
4.3.2 I <sup>2</sup> C F-RAM 器件 (256 Kbit FM24W256) .....	22
5. 示例项目 .....	25

5.1	编程 PSoC 4 Pioneer 套件 .....	25
5.2	UART 设置 .....	28
5.2.1	HyperTerminal 设置 .....	30
5.2.2	PuTTY 设置 .....	32
5.3	项目: PSoC 4 F-RAM SPI .....	34
5.3.1	项目简介 .....	34
5.3.2	硬件连接 .....	34
5.3.3	固件流程 .....	35
5.3.4	验证输出 .....	35
5.3.5	修改项目 .....	36
5.3.6	API.....	38
5.4	项目: PSoC 4 F-RAM I <sup>2</sup> C .....	39
5.4.1	项目简介 .....	39
5.4.2	硬件连接 .....	39
5.4.3	固件流程 .....	40
5.4.4	验证输出 .....	40
5.4.5	修改项目 .....	41
5.4.6	API.....	42
5.5	编程 Arduino UNO 套件 .....	44
5.6	项目: Arduino F-RAM SPI .....	45
5.6.1	项目简介 .....	45
5.6.2	硬件连接 .....	45
5.6.3	固件流程 .....	46
5.6.4	验证输出 .....	46
5.6.5	API.....	47
5.7	项目: Arduino F-RAM I <sup>2</sup> C .....	48
5.7.1	项目简介 .....	48
5.7.2	硬件连接 .....	48
5.7.3	固件流程 .....	48
5.7.4	验证输出 .....	49
5.7.5	API.....	49
附录.....		50
A.1	CY15FRAMKIT-001 原理图.....	50
A.2	引脚分配表 .....	53
A.3	调试插座 I/O .....	54
A.4	零欧姆电阻及无负载使用说明.....	55
A.5	30 欧姆大小的电阻及无负载使用说明 .....	55
A.6	材料表 (BOM) .....	56

# 安全信息



## 法规遵从

CY15FRAMKIT-001 串行 F-RAM™ 开发套件旨在用作实验室环境中硬件和软件的串行存储器开发平台。该电路板是一个开放系统设计，没有外壳保护。因此，在近距离接触中，它可能会对其他电气和电子设备造成干扰。

在该产品上连接多余的导线或修改其默认设置会影响它们的性能，并会对周围其他机械造成干扰。如果检测到此类干扰，应采取适当的措施以降低干扰。

CY15FRAMKIT-001 的出厂设置情况已被验证，为了能作为 A 类产品满足 CE 的要求。



CY15FRAMKIT-001 包含 ESD 敏感器件。静电电荷很容易会在人体或所有设备上累积，并且瞬间放电。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。应将未使用的 CY15FRAMKIT-001 电路板储存在受保护的运输包装里。



过期时间/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。赛普拉斯建议用户在 ESD 工作站环境中操作。如果没有 ESD 工作站可用，在对器件进行操作时，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。

### 处理电路板

CY15FRAMKIT-001 电路板对 ESD 十分敏感。因此，只能握着电路板的边缘。从盒子里取出电路板后，请将其放在一个接地的无静电平面上。如果可能，请使用一个导电泡沫垫。请勿将该电路板在任何表面上滑动。

# 1. 简介



感谢您对CY15FRAMKIT-001串行F-RAM开发套件（DVK）的关注。该套件（扩展板）设计为一款易于使用且廉价的开发套件，在这里展示了赛普拉斯串行铁电随机存取存储器（F-RAM）的特性。为便于使用，该套件与CY8CKIT-042 PSoC<sup>®</sup> 4 Pioneer套件、CY8CKIT-040、CY8CKIT-042 BLE、第三方Arduino<sup>™</sup> UNO R3以及某些其他Arduino套件在引脚上相兼容。该电路板具有一个256 Kbit SPI F-RAM、一个256 Kbit I<sup>2</sup>C F-RAM、用于控制I<sup>2</sup>C F-RAM器件选择引脚的三极DIP开关、Arduino UNO兼容的插座以及两个调试插座。该套件支持5.0 V或3.3 V的供电电压。

CY15FRAMKIT-001是用于赛普拉斯串行F-RAM存储器的开发套件。赛普拉斯F-RAM使用铁电技术构建，它集成了非易失性数据存储和高性能RAM的特点。串行F-RAM以总线全速提供快速写操作。它们并不存在任何写入延迟，并且数据是即时非易失性的。另外，它们消耗的工作电流可低至300  $\mu$ A，待机电流为6  $\mu$ A。由于写入速度很快，串行F-RAM仅在短的时间内保持活动状态，因而能耗非常低。串行F-RAM提供了一百万亿次的读/写周期，几乎为无限次数的耐久性。

## 1.1 套件内容

CY15FRAMKIT-001 串行 F-RAM DVK 包含以下内容，如图 1-1 所示：

- CY15FRAMKIT-001 串行 F-RAM DVK 电路板
- 快速入门指南

图 1-1. CY15FRAMKIT-001 串行 F-RAM DVK 内容



请访问 [www.cypress.com/shop](http://www.cypress.com/shop) 网站，了解更多信息。检查该套件的内容时，如果发现缺少任何器件，请联系离您最近的赛普拉斯销售办事处以获得帮助。

## 1.2 PSoC Creator™

PSoC Creator 是最先进、简单易用的集成开发环境（IDE）。它引进了革命性的硬件和软件协同设计理念，它有一个预验证和预配置的 PSoC Components™ 库。

使用 PSoC Creator，可以执行以下操作：

- 拖放各 PSoC 组件，以构建您自定义设计的原理图
- 自动放置和布线各组件，并配置 GPIO
- 使用所包含的组件 API 开发并调试固件

PSoC Creator 通过其集成的编译器工具链和量产编程器，为您提供了完备的开发工具生态系统。有关更多信息，请访问网址 [www.cypress.com/Creator](http://www.cypress.com/Creator)。

## 1.3 入门指南

通过本指南，用户可以了解串行 F-RAM DVK。“套件安装”章节介绍了该套件软件的安装情况。“套件概况”章节介绍了该套件特性。“硬件”章节详细说明了硬件的操作。“示例项目”章节介绍了代码示例。“附录”部分提供了各个原理图、引脚分配、零欧姆和 30 欧姆电阻的使用说明以及材料表（BOM）。

## 1.4 其他参考资源

有关数据手册和应用笔记等其他参考资源的信息，请访问 [www.cypress.com/go/F-RAM](http://www.cypress.com/go/F-RAM) 网址。下面的链接提供了基板相关资源。

- [CY8CKIT-042 PSoC 4 Pioneer 套件](#)
- [Arduino UNO R3](#)

## 1.5 技术支持

有关技术支持，请访问 [www.cypress.com/support](http://www.cypress.com/support) 网址，或致电我们公司的客户支持：+1 (800) 541-4736 选择 2 号键（美国），或 +1 (408) 943-2600 选择 2 号键（国际）。

## 1.6 文档规范

表 1-1. 用户指南的文档规范

规范	使用说明
Courier New 字体	显示文件位置、用户输入的文本和源代码： C:\...cd\icc\
斜体	显示文件名和参考文档： 阅读 <i>PSoC Designer 用户指南</i> 中的 <i>sourcefile.hex</i> 文件。
[方括号、粗体]	显示程序中的键盘指令： [Enter] 或 [Ctrl] [C]
File（文件）> Open（打开）	表示菜单路径： File（文件）> Open（打开）> New Project（新建项目）
粗体	显示程序中的指令、菜单路径和图标名称： 单击 <b>File</b> 图标，然后再单击 <b>Open</b> 。
Times New Roman 字体	显示公式： $2 + 2 = 4$
文本显示在灰色框中	说明警告或产品的独特功能。



## 2. 套件安装



本节介绍了 CY15FRAMKIT-001 串行 F-RAM DVK 软件的安装和必备软件。

### 2.1 必备软件

使用该套件前，用户必须安装 PSoC Creator IDE 或 Arduino IDE。安装任何赛普拉斯软件时都需要管理员权限。安装软件后，不再需要管理员权限。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。

要下载 PSoC Creator IDE，请输入 [www.cypress.com/psoccreator](http://www.cypress.com/psoccreator)。

要想下载 Arduino IDE，请输入 [arduino.cc/en/Main/Software](http://arduino.cc/en/Main/Software)。

### 2.2 安装套件软件

请按照下面的步骤安装 CY15FRAMKIT-001 串行 F-RAM DVK 软件：

1. 在 [www.cypress.com/go/CY15FRAMKIT-001](http://www.cypress.com/go/CY15FRAMKIT-001) 网址上下载 CY15FRAMKIT-001 DVK 软件。使用三种不同的格式下载 CY15FRAMKIT-001 软件（请参见图 2-1）：

图 2-1. 使用于下载串行 F-RAM DVK 软件的可用格式

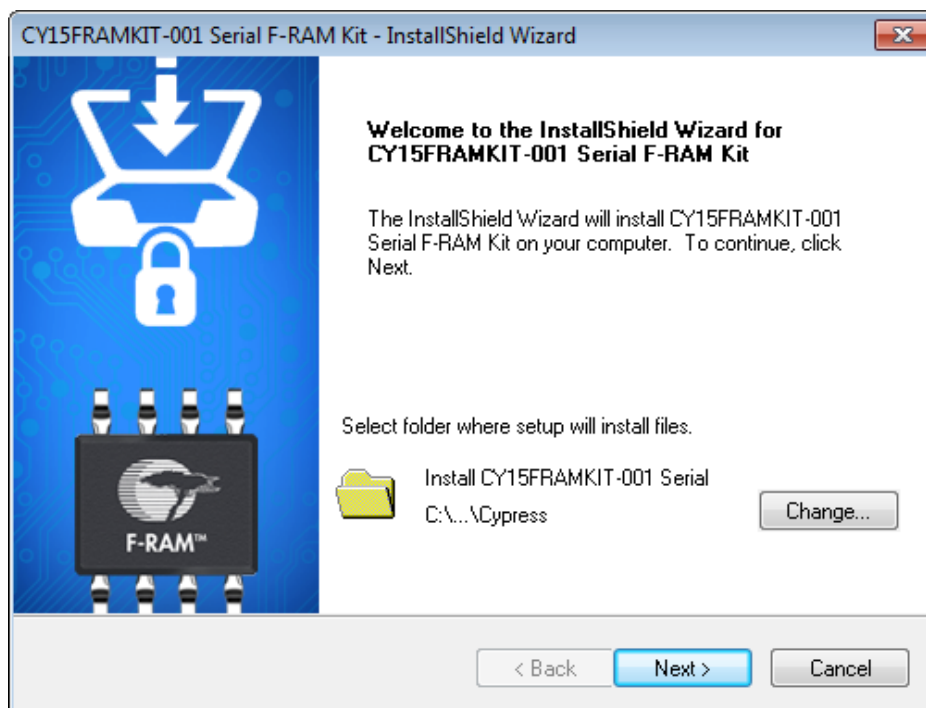


- **CY15FRAMKIT-001 Kit Setup:** 这个可执行文件仅安装套件的内容，包括套件代码示例、硬件文件以及用户文档。
  - **CY15FRAMKIT-001 CD ISO:** 该文件是一个以 CD-ROM 镜像格式存放的完整安装包，您可以使用它来创建 CD 或通过 ISO 解压程序（如 WinZip 或 WinRAR）对其进行解压。也可以使用虚拟光驱程序（如 Virtual CloneDrive 和 MagicISO）来安装该文件。该文件包含了所需的全部固件、硬件文件以及用户文档。
2. 如果您已经下载了 CD ISO 文件，请将其加载到一个虚拟光驱程序中。如果没有用于下载的虚拟光驱程序，则要对 ISO 内容进行解压。如果电脑上尚未启用‘Autorun from CD/DVD’功能，请双击解压内容或所加载的 ISO 文件根目录中的 **cyautorun.exe**。这时，会自动出现安装窗口，如图 2-2 所示。

**注意：**如果您正在使用“套件设置”文件，则转到第 3 步进行安装。

3. 图 2-2 显示的是套件启动屏幕。默认安装位置位于 **Program Files** 文件夹下的 **Cypress** 中。

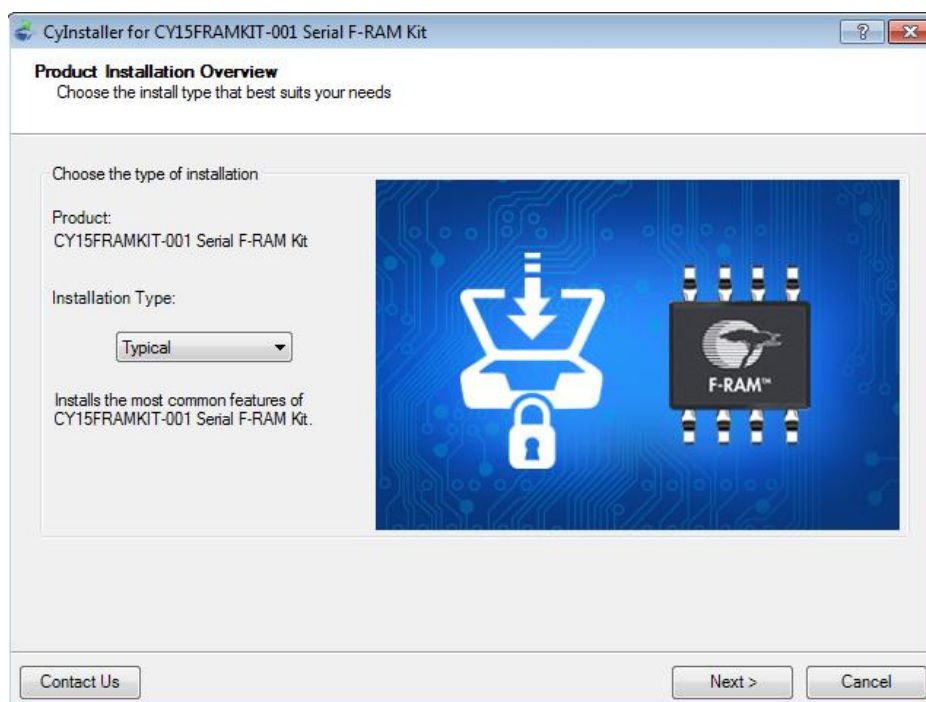
图 2-2. 套件安装程序启动屏幕



选择您想要安装 CY15FRAMKIT-001 套件相关文件所在的文件夹。选择该目录并点击 **Next** 按钮。

4. 在 **Product Installation Overview**（产品安装概述）窗口中，请选择安装类型为 **Typical installation**（典型安装），如图 2-3 所示。然后点击 **Next** 按钮。

图 2-3. Product Installation Overview 窗口



当开始进行安装时，安装包列表会显示在安装页面上。成功安装后，在安装包后面都会显示一个绿色对勾。

5. 如果这是第一个安装的赛普拉斯产品，请输入您的联系方式，或选择 **Continue without Contact Information** 选项。点击 **Finish**（完成）来完成安装 CY15FRAMKIT-001 套件。

安装成功后，可以在<Install\_Directory>\CY15FRAMKIT-001 Serial F-RAM Kit\1.0 中找到该套件内容。默认位置如下：

- Windows 7 (64 位) : C:\Program Files (x86)\Cypress\CY15FRAMKIT-001 Serial F-RAM Kit\1.0
- Windows 7 (32 位) : C:\Program Files\Cypress\CY15FRAMKIT-001 Serial F-RAM Kit\1.0

**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 取消选择 **Read-only** 选项。点击 **Apply** 和 **OK** 按键，以关闭窗口。

完成安装后，在电脑上安装以下内容：

- 套件文档
  - 快速入门指南
  - 套件指南
  - 发布说明
- 固件
  - PSoC 4 F-RAM SPI 示例项目
  - PSoC 4 F-RAM I2C 示例项目
  - Arduino UNO F-RAM SPI 示例项目
  - Arduino UNO F-RAM I2C 示例项目
- 硬件
  - 原理图
  - 布局
  - Gerber
  - PCB 装备图
  - 材料表 (BOM)

## 2.3 卸载软件

通过使用以下方法中的某一个可以卸载软件：

- 依次选择 **Start > Control Panel > Programs and Features**，选择相应的软件安装包，并点击 **Uninstall** 按键。
- 依次选择 **Start > All Programs > Cypress > Cypress Update Manager > Cypress Update Manager**。
  - 选择 “CY15FRAMKIT-001 Serial F-RAM Kit 1.0 Rev \*\*” 行，然后点击 **Uninstall**。在 **Product Installation Overview** 窗口中，选择 **Installation Type**（安装类型）下拉菜单中的 **Remove**。按照说明来卸载软件。

**注意：**这种方法只能卸载套件软件，并不会卸载随套件软件安装的其他所有软件。

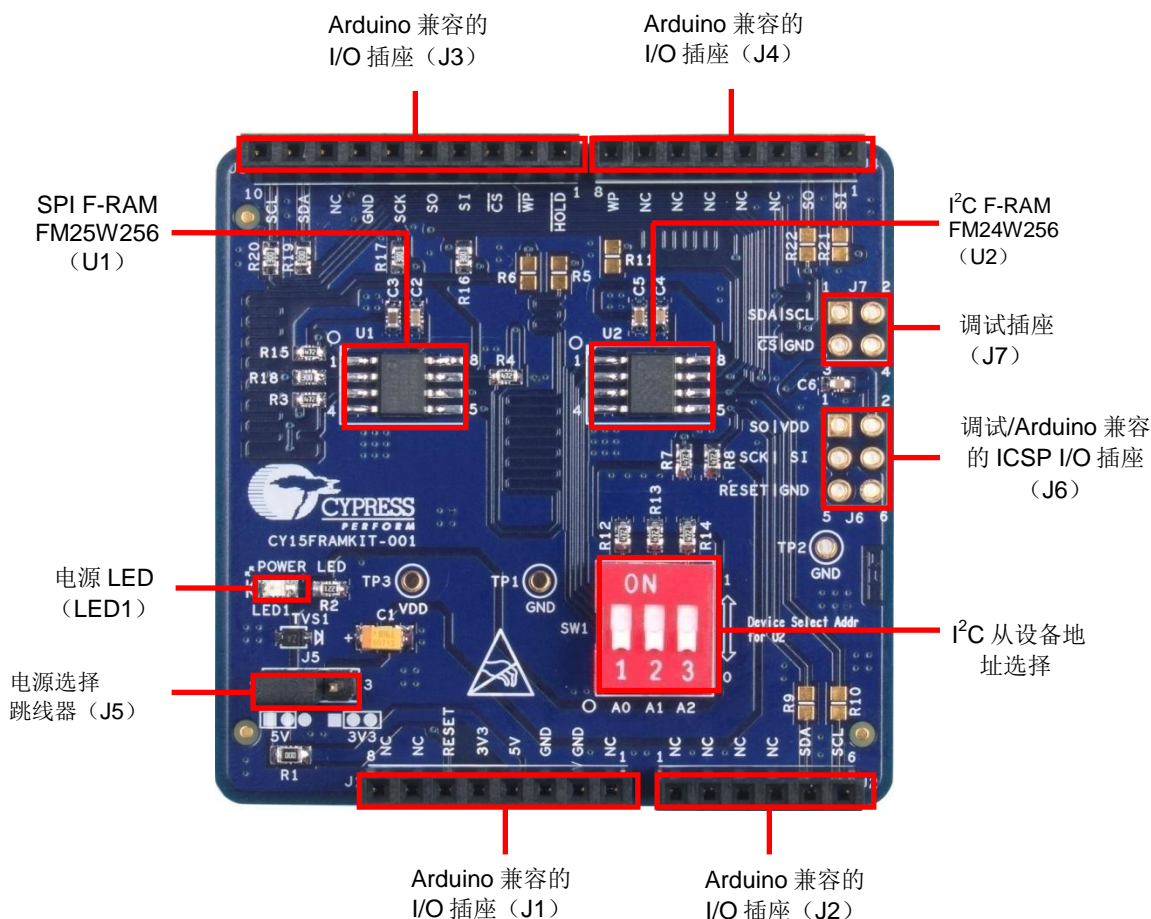
## 3. 套件概述



### 3.1 CY15FRAMKIT-001 串行 F-RAM 开发套件 概述

通过 CY15FRAMKIT-001 串行 F-RAM DVK 可以了解串行 F-RAM (SPI 和 I<sup>2</sup>C) 的特性。DVK 是一个扩展板, 它包含 256 Kbit SPI F-RAM 和 256 Kbit I<sup>2</sup>C F-RAM。它共有四个连接器, 这些连接器是 Arduino UNO 可兼容的, 并连接到赛普拉斯 PSoC 4 Pioneer 套件或 Arduino UNO R3 电路板。这些连接器都是可堆叠的, 因此所有其他 Arduino 兼容的扩展板都可以堆叠在它上面。该套件可以在 3.3 V 或 5.0 V 的电压下运行。Figure 3-1 显示的是 DVK 电路板及其概述。

Figure 3-1.DVK 电路板概述



请参考 A.5 30 欧姆大小的电阻及无负载使用说明部分, 以便通过安装两个 30  $\Omega$  大小的电阻来修改 CY15FRAMKIT-001 硬件, 从而使其能够与 Arduino UNO R2 相兼容。

## 3.2 套件操作与配置指南

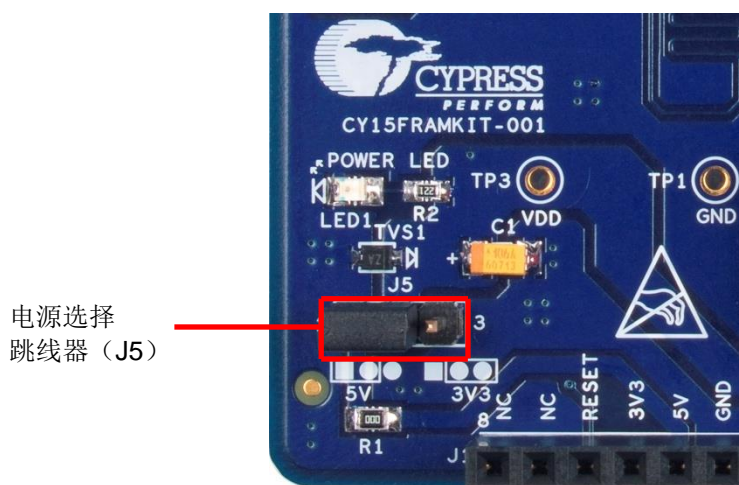
### 3.2.1 电源跳线器

CY15FRAMKIT-001 可以在 3.3 V 或 5.0 V 的电压下工作，通过跳线器 J5 可以选择该工作电压，如图 3-2 所示。跳线器的出厂默认设置为 5.0 V（短接引脚 1 和 2）。Arduino UNO R3 电路板在电压为 5.0 V 下工作，并使用跳线器 J5 的默认设置。CY8CKIT-042 Pioneer 套件的工作电压为 3.3 V 或 5.0 V。Pioneer 套件的默认设置为 3.3 V。因此，根据母板上的电压设置，将跳线器 J5 设置为 5.0 V（短接引脚 1 和 2）或设置为 3.3 V（短接引脚 2 和 3）。

#### 警告

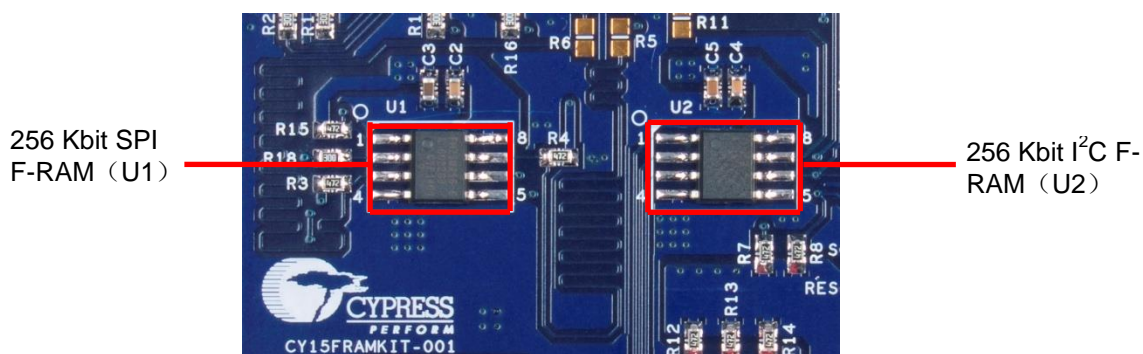
1. 请不要通过外部电源为 CY15FRAMKIT-001 电路板供电。母板会为该电路板供电。
2. CY15FRAMKIT-001 的工作电压范围为 2.7 V 到 5.5 V。如果电压超过最大限制（5.5 V）会损坏电路板。
3. 要使电路板正常工作，请确保通过设置跳线器 J5 来匹配母板的工作电压。

图 3-2. 电源选择跳线器



### 3.2.2 F-RAM 器件

图 3-3. 电路板上的赛普拉斯串行 F-RAM 器件

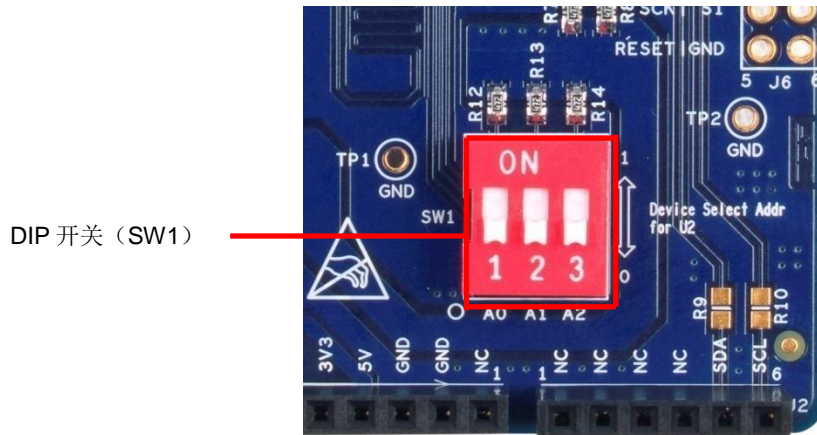


### 3.2.3 用于 I²C F-RAM 从设备地址选择的 DIP 开关

DVK 提供了一个 DIP 开关选项（如图 3-4 所示），该选项用于通过将其单独从设备选择地址引脚（A2-A0）分配为逻辑低‘0’或逻辑高‘1’来配置 I²C F-RAM 的不同从设备地址。DIP 开关位于 ON 状态时，通过使用一个大小为 4.7 kΩ 的上拉电阻来保持相应从设备选择地址引脚为逻辑高。DIP 开关不在 ON 状态时，通过一个内部弱下拉电阻，F-RAM 器件会保持相应从设备选择地址引脚为逻辑低。DIP 开关出厂时被默认设置为不处于 ON 状态，这样 A2、A1 和 A0 的值则被设置为逻辑低‘0’。



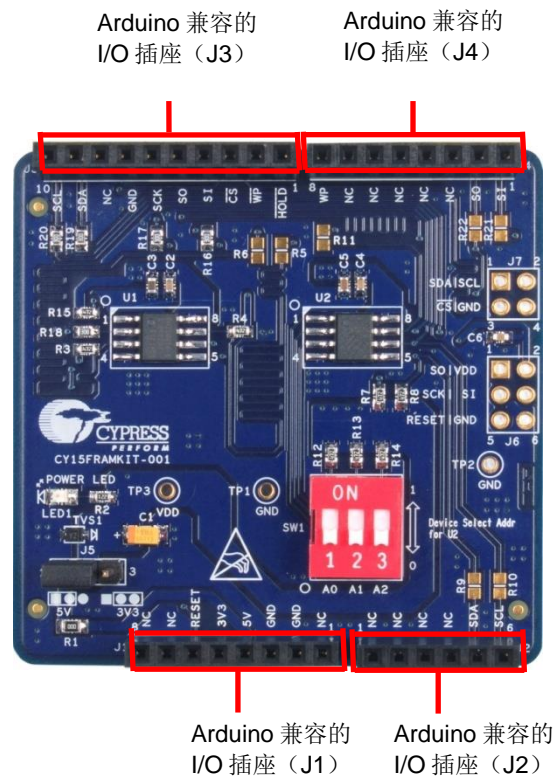
图 3-4. 用于 I<sup>2</sup>C F-RAM A0、A1 和 A2 配置的 DIP 开关



### 3.2.4 用于连接到 CY8CKIT-042/Arduino UNO R3 电路板的连接器

图 3-5 显示的是连接到 CY8CKIT-042/Arduino UNO R3 电路板的 Arduino 兼容的连接器：J1、J2、J3 和 J4。您可以使用这些连接器将 DVK 电路板插到 CY8CKIT-042/Arduino UNO R3 电路板上。有关原理图的详细信息，请参考第 17 页上的硬件章节。

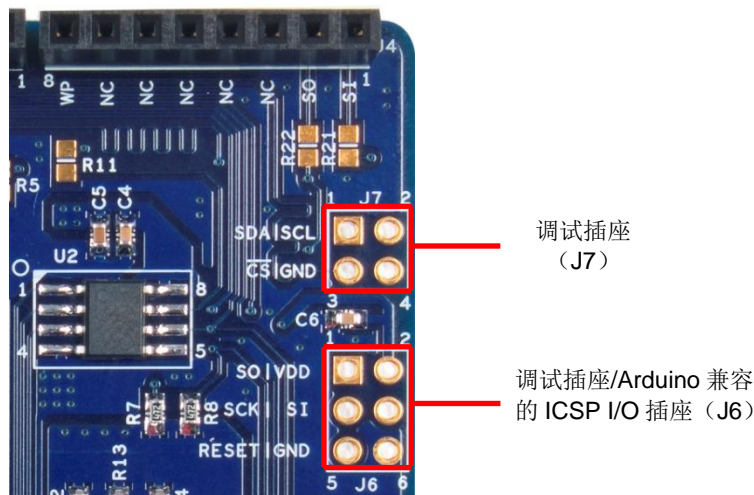
图 3-5. 用于连接到 CY8CKIT-042/Arduino UNO R3 电路板的连接器



### 3.2.5 调试插座

提供调试插座是为了易于访问 SPI 和 I<sup>2</sup>C 通信引脚。如图 3-6 所示，跳线器 J6 采用了电路内串行编程（ICSP）插座引脚布局，并提供对 SPI 信号进行访问。跳线器 J7 提供了访问 SPI F-RAM 器件的“芯片选择”信号和 I<sup>2</sup>C 通信信号。

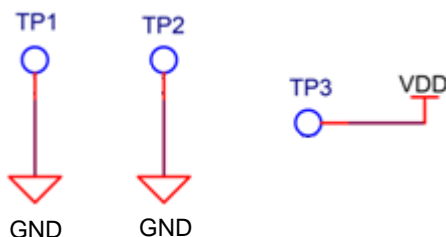
图 3-6. 调试插座



### 3.2.6 测试点

DVK 电路板提供了 V<sub>DD</sub> (TP3) 和 GND (TP1 与 TP2) 测试点，如图 3-7 所示。

图 3-7. 测试点



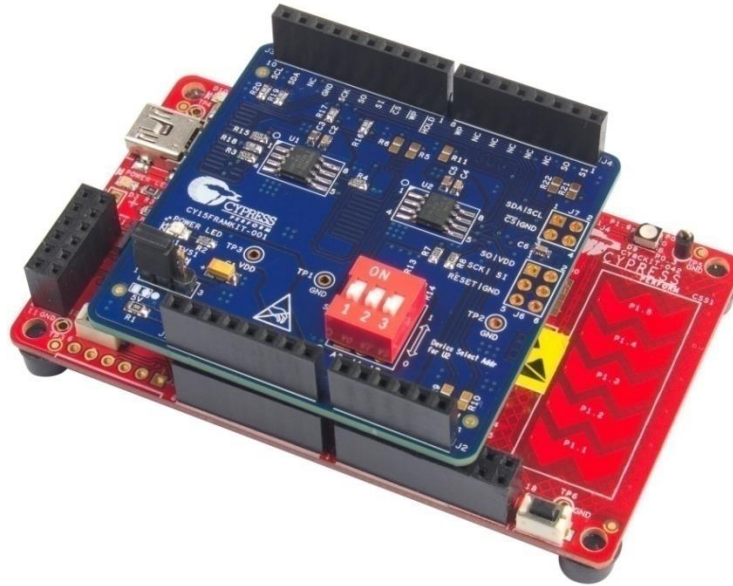
**警告** 这些测试点仅用于探测和测量目的。请不要通过这些测试点为该套件供电，从而避免损坏电路板。

## 3.3 CY15FRAMKIT-001 与 PSoC 4 Pioneer 套件

通过四个连接器（J1、J2、J3 以及 J4）将 CY15FRAMKIT-001 连接到 PSoC 4 Pioneer 套件（CY8CKIT-042）上。串行 F-RAM 套件的默认工作电压为 5.0 V。要与串行 F-RAM 套件相匹配，需要将 PSoC 4 pioneer 套件上的电压更改为 5.0 V。另外，通过将 PSoC 4 pioneer 套件和串行 F-RAM 套件上的电压都选为 3.3 V，那么在电压为 3.3 V 的条件下仍可运行。

**注意：** CY15FRAMKIT-001 上的 J2 是一个 6 引脚的单排跳线器。在 PSoC 4 Pioneer 套件中，J2 是一个 9x2 的插座。因此，将 CY15FRAMKIT-001 的 J2 插入到 Pioneer 电路板上 J2 插座左侧靠外一排的六个引脚上。很容易便能观察到该事件，因为 CY15FRAMKIT-001 只能按照一个方向插入。

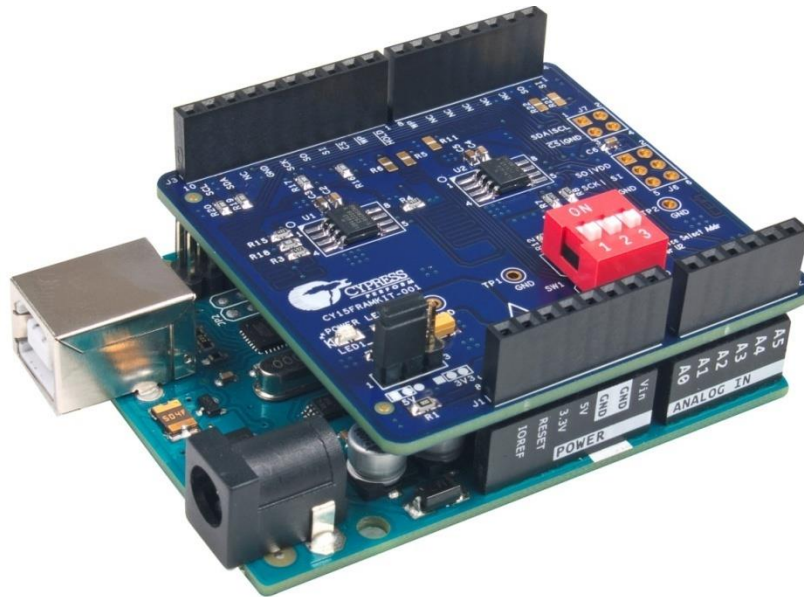
图 3-8. 在 PSoC 4 Pioneer 套件上安装的 CY15FRAMKIT-001



### 3.4 CY15FRAMKIT-001 与 Arduino UNO R3 套件

通过四个连接器（J1、J2、J3 以及 J4）将 CY15FRAMKIT-001 连接到 Arduino UNO R3 套件。一对一映射这些连接器。

图 3-9. 在 Arduino UNO R3 套件上安装的 CY15FRAMKIT-001





## 4. 硬件



### 4.1 电路板的详细信息

CY15FRAMKIT-001 串行 F-RAM DVK 包含以下模块，如图 4-1 所示。

- 256 Kbit SPI F-RAM (U1)
- 256 Kbit I2C F-RAM (U2)
- 电源选择跳线器 (J5)
- 电源 LED (LED1)
- 用于器件地址选择的 DIP 开关 (SW1)
- SPI/I2C 信号的调试插座 (J6、J7)
- Arduino 兼容的连接器 (J1、J2、J3、J4)

图 4-1. CY15FRAMKIT-001 串行 F-RAM DVK 详情

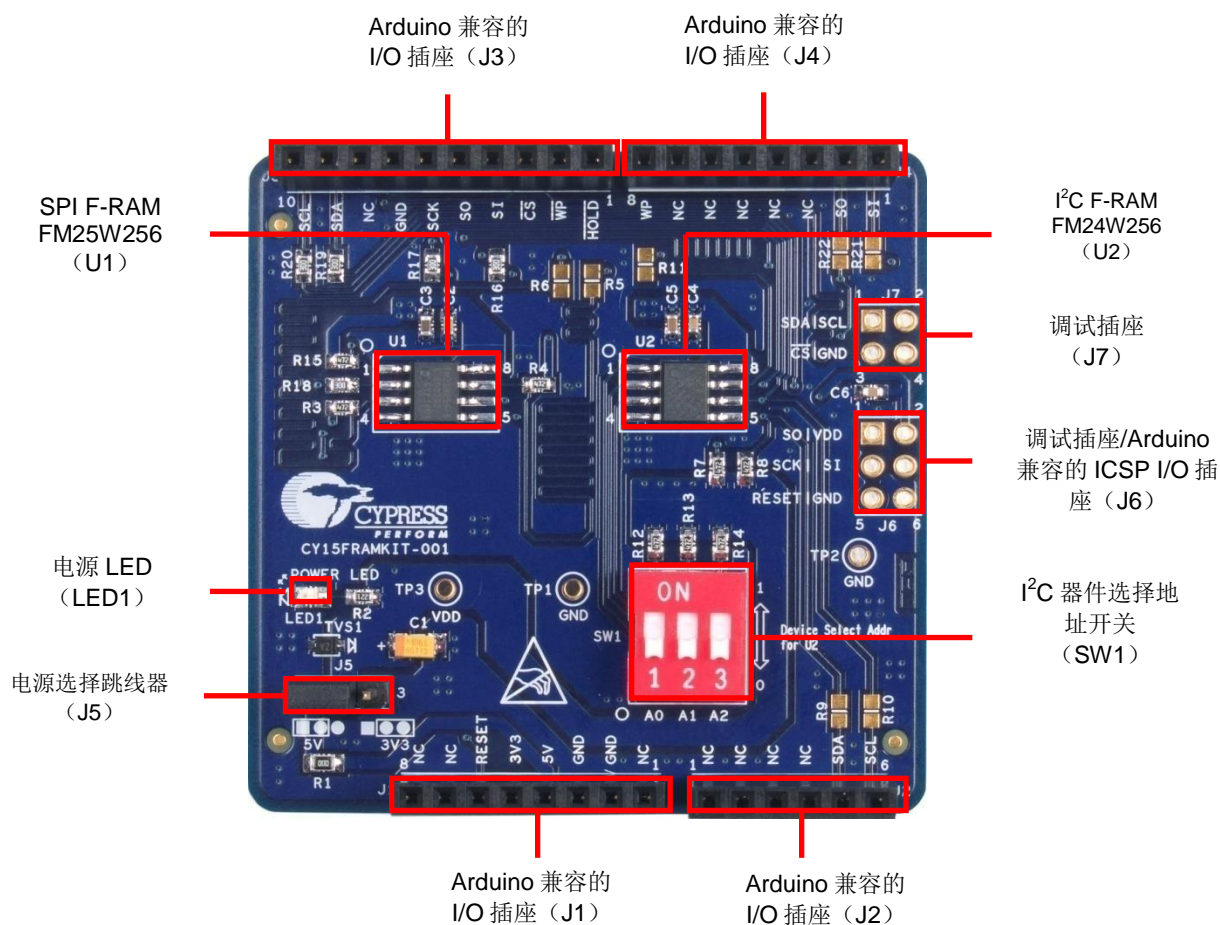
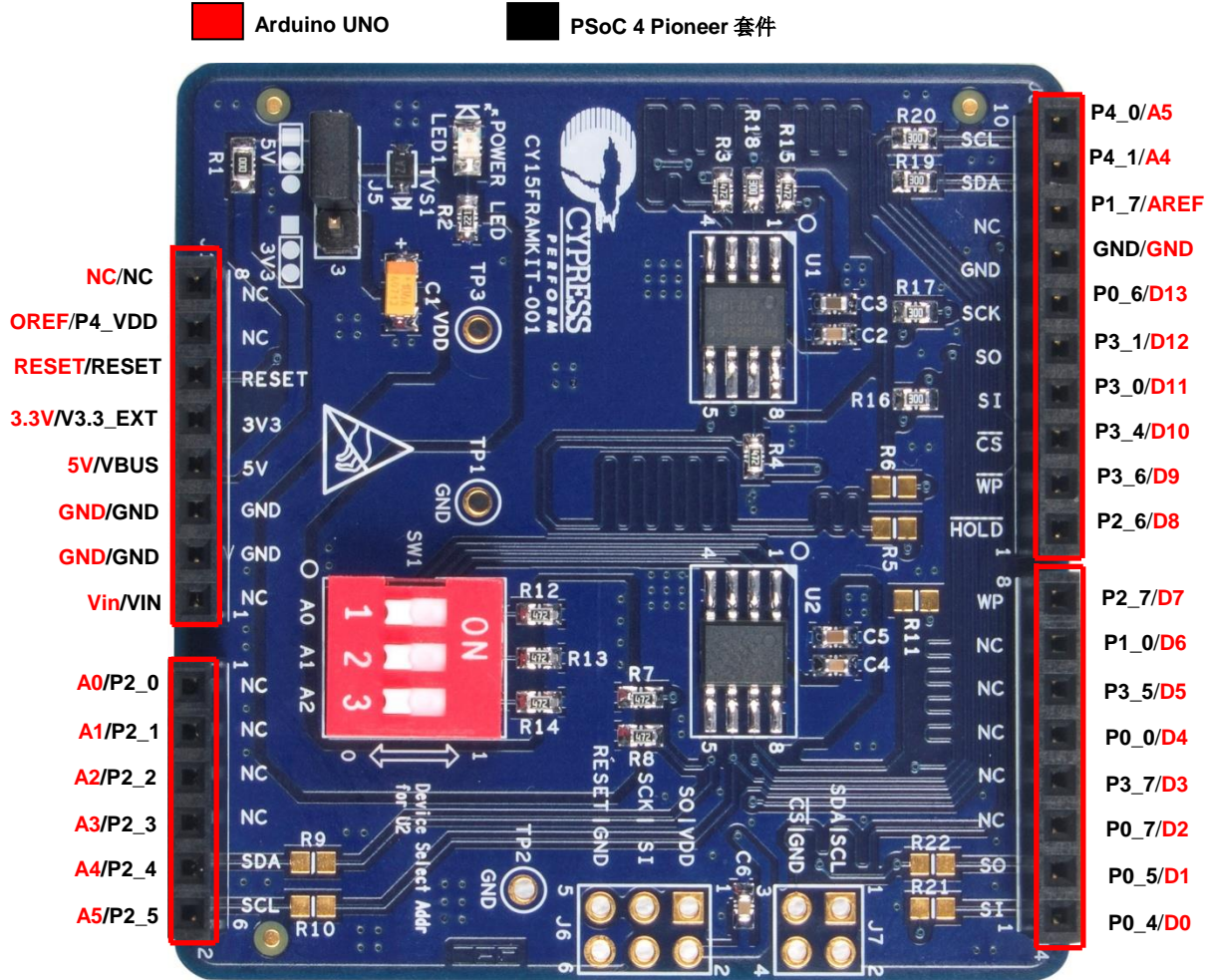


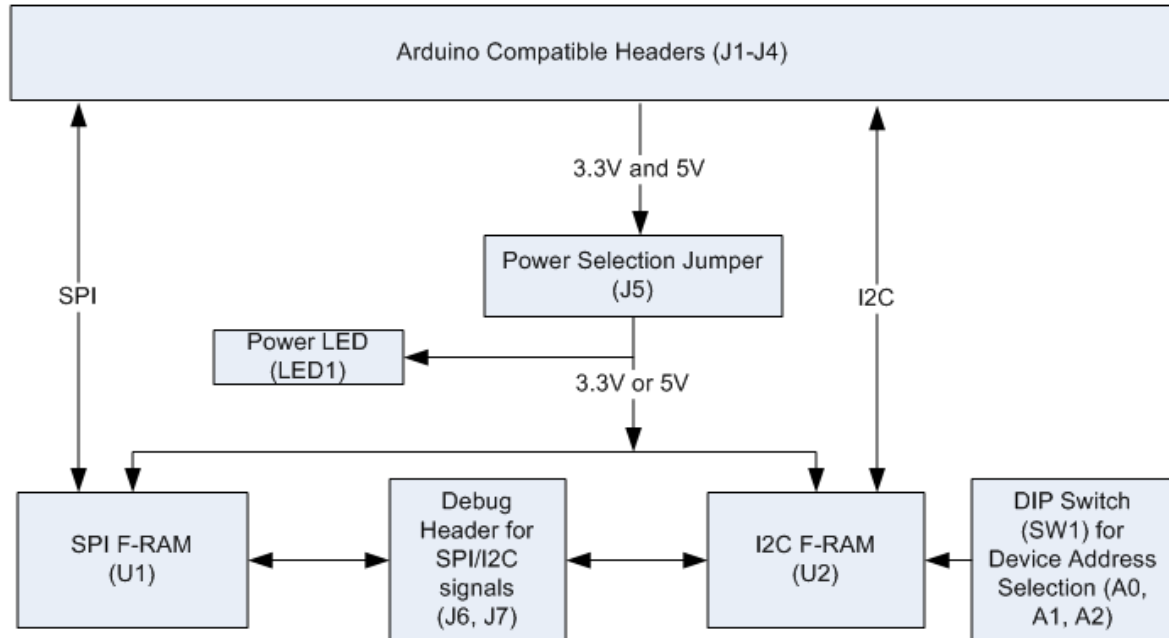
图 4-2. 针对 PSoC 4 Pioneer/Arduino UNO R3 套件进行的 CY15FRAMKIT-001 串行 F-RAM DVK 引脚映射



## 4.2 操作原理

本章节介绍的是 PSoC 4 Pioneer 套件的架构说明。

图 4-3. 框图



CY15FRAMKIT-001 串行 F-RAM DVK 提供了 256 Kbit SPI 和 256 Kbit I<sup>2</sup>C F-RAM 器件。该 DVK 是一个扩展板，可以将该板插入到 PSoC 4 Pioneer 套件或 Arduino UNO R3 套件上。

CY15FRAMKIT-001 串行 F-RAM DVK 具有一个电源 LED（LED1），用于指示电源接通状态。该套件的工作电压为 3.3 V 或 5 V，具体电压通过跳线器 J5 选择。另外，该套件还有一个用于修改 I<sup>2</sup>C F-RAM 的器件地址的选项。该套件也为 SPI/I<sup>2</sup>C 信号提供了调试插座。

## 4.3 功能说明

### 4.3.1 SPI F-RAM 器件（256 Kbit FM25W256）

CY15FRAMKIT-001 串行 F-RAM DVK 使用赛普拉斯 256 Kbit SPI F-RAM 器件。器件引脚通过 Arduino 兼容的插座进行读写访问。

#### 特性

- 256 Mbit 铁电性随机存储器（F-RAM）被逻辑组织为 32K × 8
  - 高耐久性：100 万亿（10<sup>14</sup>）次写/读
  - 151 年数据保留时间
  - NoDelay™ 写操作
  - 高级高可靠性的铁电工艺
- 快速串行外设接口（SPI）
  - 工作频率高达 20 MHz
  - 串行闪存和 EEPROM 的硬件直接替代
  - 支持 SPI 模式 0（0,0）和模式 3（1,1）
- 精密的写入保护方案

- 使用写保护 ( $\overline{WP}$ ) 引脚提供硬件保护
- 使用写禁用指令提供软件保护
- 可对 1/4、1/2 或整个阵列进行软件模块保护
- 低功耗
  - 频率为 1 MHz 时，有效电流为 250  $\mu$ A
  - 待机电流为 15  $\mu$ A（典型值）
- 工作电压：VDD = 2.7 V ~ 5.5 V
- 工业温度范围：-40  $^{\circ}$ C ~ +85  $^{\circ}$ C
- 8 引脚小型塑封集成电路（SOIC）封装
- 符合有害物质限制标准（RoHS）

### 存储器写入操作

对 SPI F-RAM 进行的所有写操作都是以 WREN（写使能）操作码（0x06）开始的。WRITE 操作码（0x02）后面是一个两字节的地址，该地址包含了将写入到存储器的第一个数据字节的 15 位地址（A14-A0）。两字节地址的高位被忽略。这两个地址字节后面紧跟的是连续写入的数据字节。如果总线主设备继续发送时钟并保持  $\overline{CS}$  为低电平，则各地址将内部递增。如果达到最后地址 7FFFh，计数器将翻转到 0000h。先写入数据的最高有效位。  $\overline{CS}$  的上升沿终止写操作。下图显示了 SPI F-RAM 写操作。

图 4-4. SPI F-RAM 存储器写入

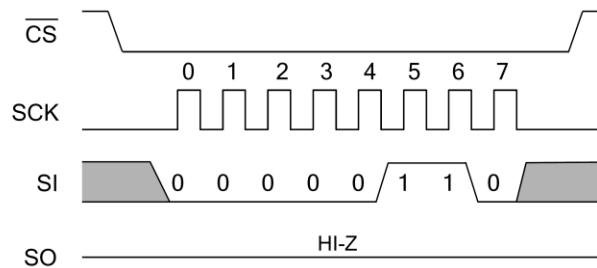
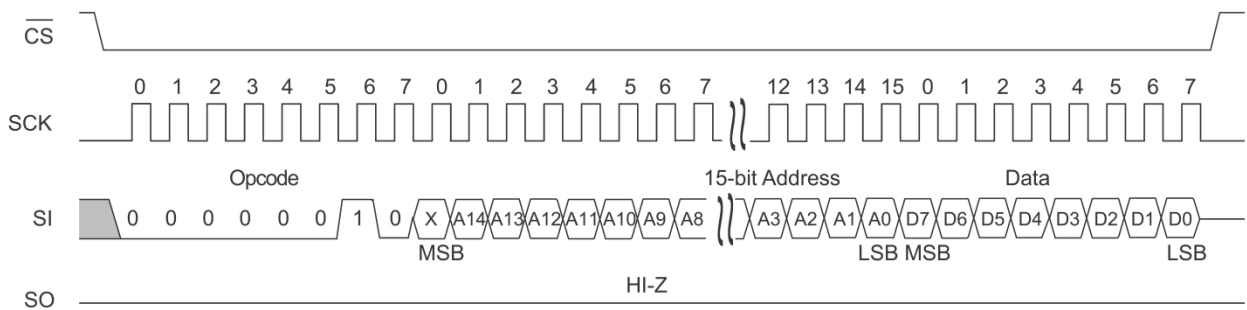


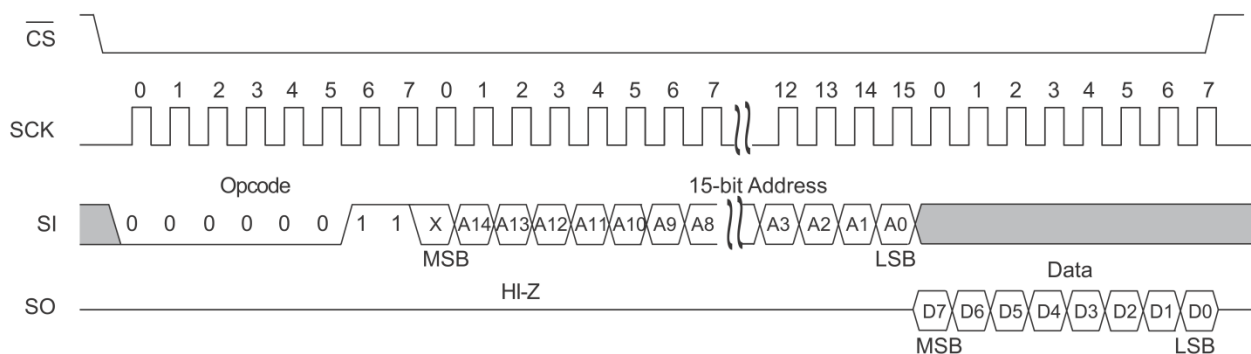
图 4-5. SPI F-RAM 存储器写入



### 存储器读取操作

$\overline{CS}$  的下降沿后，总线主设备将发送一个 READ 操作码（0x03）。READ 指令后面是一个两字节地址，该地址包含读操作第一个字节的 15 位地址（A14-A0）。地址的高位被忽略。发送读操作码和两个地址字节后，在 SO 引脚上的后续八个时钟内器件将输出读数据。在驱动读取数据字节期间，SI 输入被忽略。后续字节是连续读取的数据字节。只要总线主设备继续发送时钟并  $\overline{CS}$  保持为低电平，地址将内部递增。如果达到最后地址 7FFFh，计数器将翻转到 0000h。先读取数据的最高有效位。  $\overline{CS}$  的上升沿终止读操作并使 SO 引脚处于三态。下图显示了 SPI F-RAM 读操作。

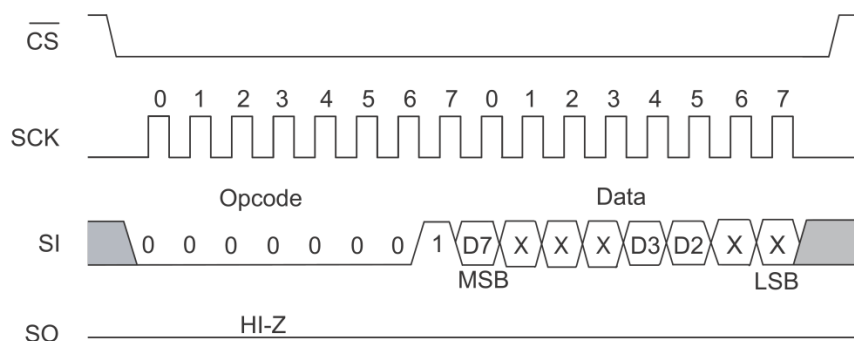
图 4-6. SPI F-RAM 存储器读取



### 状态寄存器写入操作

通过写状态寄存器 WRSR 指令 (0x01)，SPI 总线主设备可以写入到 SPI F-RAM 的状态寄存器中。

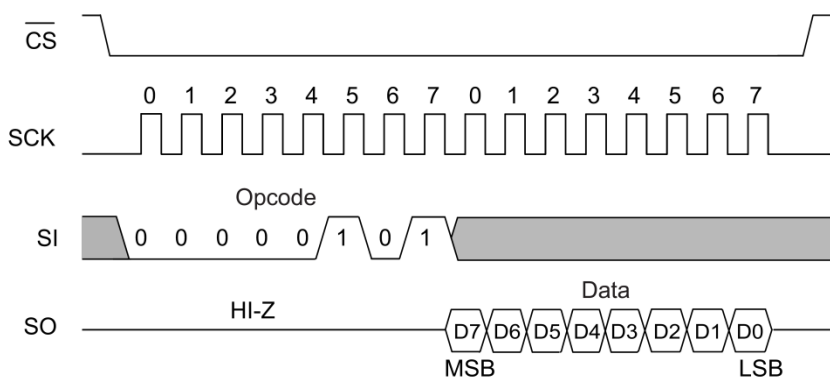
图 4-7. SPI F-RAM 状态寄存器写操作



### 状态寄存器读取操作

通过读状态寄存器 RDSR 指令 (0x05)，总线主设备能验证 SPI F-RAM 的状态寄存器数据。读取状态寄存器后可以了解写保护特性的当前状态。执行 RDSR 操作码后，SPI F-RAM 器件将返回一个字节，该字节包含状态寄存器的内容。

图 4-8. SPI F-RAM 状态寄存器读取操作



更多有关 F-RAM 器件的信息，请参考 [FM25W256 数据手册](#) 中的内容。



### 4.3.2 I<sup>2</sup>C F-RAM 器件（256 Kbit FM24W256）

CY15FRAMKIT-001 串行 F-RAM DVK 使用 256 Kbit I<sup>2</sup>C F-RAM 器件。器件通过 Arduino 兼容的插座执行读写访问。

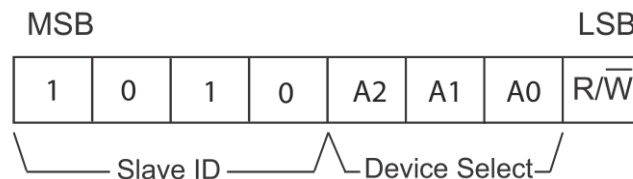
#### 特性

- 256 Kbit F-RAM 被逻辑组织为 32K × 8
  - 高耐久性：100 万亿（10<sup>14</sup>）次写/读
  - 151 年数据保留时间
  - NoDelay™ 写操作
  - 高级高可靠性的铁电工艺
- 快速双线串行接口（I<sup>2</sup>C）
  - 工作频率高达 1 MHz
  - 使用硬件直接替代串行（I2C）EEPROM
  - 支持在 100 kHz 和 400 kHz 频率时的旧版计时器
- 低功耗
  - 频率为 100 kHz 时，工作电流为 100 μA
  - 待机电流为 15 μA（典型值）
- 工作电压：VDD = 2.7 V ~ 5.5 V
- 工业温度范围：-40 °C ~ +85 °C
- 8 引脚小型塑封集成电路（SOIC）封装
- 符合有害物质限制标准（RoHS）

#### 从设备地址

发出 START 条件后，FM24W256 所期望接收的第一个字节是从设备地址。如图 4-9 所示，从设备地址包括器件类型（从设备 ID）、器件选择地址位以及一个用于指定读/写操作的位。对于 FM24W256，位 7-4 是器件类型（从设备 ID），并应该将其设置为 1010b。通过这些位，可以将处于相同地址范围的其他功能类型放置在该 I<sup>2</sup>C 总线上。位 3-1 是器件选择地址位。为选择所需的器件，它们要符合外部地址引脚上的相应值。通过分别给每个器件分配另一个地址，可在同一个 I<sup>2</sup>C 总线上放置八个 FM24W256 器件。位 0 是读/写位（R/W）。R/W = ‘1’ 指示一个读操作以及 R/W = ‘0’ 指示一个写操作。

图 4-9. I<sup>2</sup>C F-RAM 从设备地址寄存器

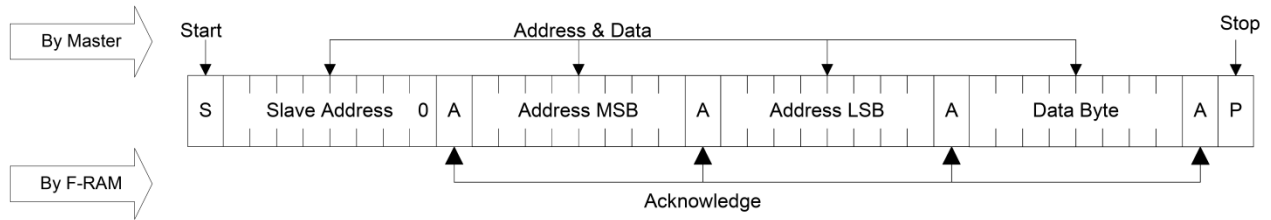


#### 存储器写入操作

所有写操作都是以一个从设备地址开始的，后面是一个存储器地址。总线主设备通过将设备地址的 LSB（R/W 位）设置为 ‘0’ 来指示一个写操作。寻址后，总线主设备将每个数据字节发送到存储器，然后存储器做出相应的应答。可以写入任意连续字节数量。如果内部达到地址范围的最后地址，则地址计数器将从 7FFFh 翻转为 0000h。

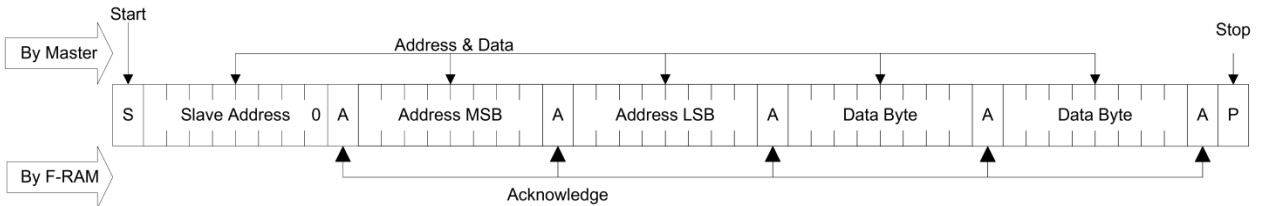
单字节写入:

图 4-10. I<sup>2</sup>C F-RAM 单字节存储器写入操作



多字节写入:

图 4-11. I<sup>2</sup>C F-RAM 多系统存储器写入操作



### 存储器读取操作

读操作有两种基本的类型：当前地址读取和选择性（随机）地址读取。对于当前地址读取，FM24W256 使用内部地址锁存器来提供地址。对于选择性读取，用户通过执行一个流程，以将地址设置为一个特定值。

#### 当前地址读取和连续读取

FM24W256 使用内部锁存为读操作提供地址。进行读操作时，当前地址读取会将地址锁存器中现有的值作为起始地址使用。系统将从紧跟着最后访问位置的地址进行读取。

为执行当前地址读取，总线主设备将提供一个从设备地址，其中 LSB 被设置为 ‘1’。这表示要求执行读操作。接收完整从设备地址后，FM24W256 将在下一个时钟周期内从当前的地址移出数据。当前地址是内部地址锁存器中所保留的值。

从当前的地址开始执行，总线主设备可以读取任意数量的字节。因此，一个连续读取是指多字节传输的当前地址读取。传输完每个字节后，内部地址计数器将被递增。

图 4-12. I<sup>2</sup>C F-RAM 单字节当前地址读取

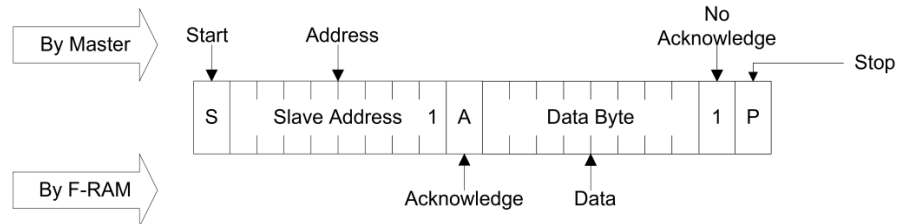
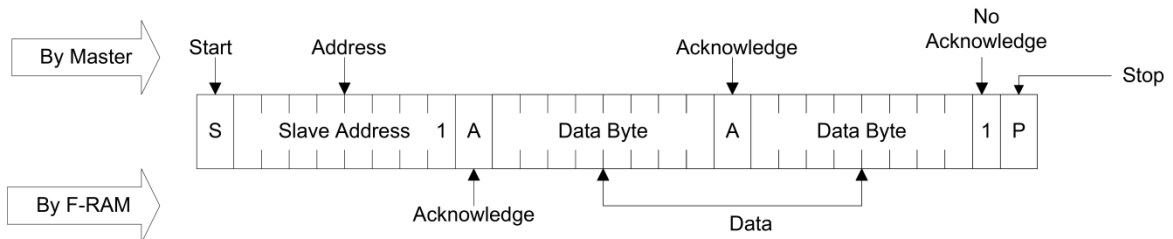


图 4-13. I<sup>2</sup>C F-RAM 多系统连续读取

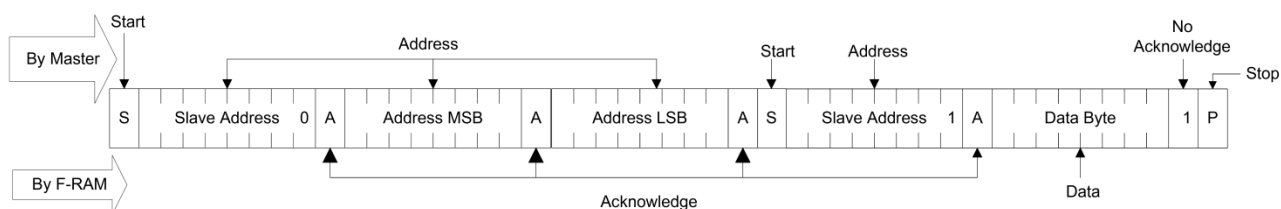


## 选择性（随机）读取

通过一种简单的技术，用户可以选择一个随机地址位置作为读操作的起始地址。该技术使用写操作的前三个字节来设置后续读操作随后的内部地址。

为执行一个选择性的读取操作，总线主设备将发送一个从设备地址，其中 **LSB (R/W)** 被设置为 ‘0’，以表示要求执行一个写操作。根据写协议，总线主设备会发送被加载到内部地址锁存器中的地址字节。**FM24W256** 应答该地址后，总线主设备将发送一个 **START** 条件。这样会终止写操作，同时发送一个读指令，其中从设备地址中的 **LSB** 被设置为 ‘1’。现在要执行当前地址读取操作。

图 4-14. F-RAM I<sup>2</sup>C 选择性（随机）读取





## 5. 示例项目



赛普拉斯提供 CY15FRAMKIT-001 串行 F-RAM DVK 封装时会附带四个示例项目。通过这些示例项目，您可以了解串行 F-RAM 接口和微控制器。这些示例项目都是以 PSoC 4 Pioneer 套件和 Arduino UNO R3 套件为基础的。

从 [www.cypress.com/go/CY15FRAMKIT-001](http://www.cypress.com/go/CY15FRAMKIT-001) 网站上下载这些示例项目。安装程序将示例项目放置在 <Install\_Directory>\CY15FRAMKIT-001 Serial F-RAM Kit\1.0\Firmware\文件夹中。

下面的示例项目是基于 PSoC 4 的，并需要使用 PSoC Creator 软件和 PSoC 4 Pioneer 套件。

- PSoC4\_FRAM\_SPI
- PSoC4\_FRAM\_I2C

下面的示例项目是基于 Arduino 的，并需要使用 Arduino IDE 和 Arduino UNO R3 套件。

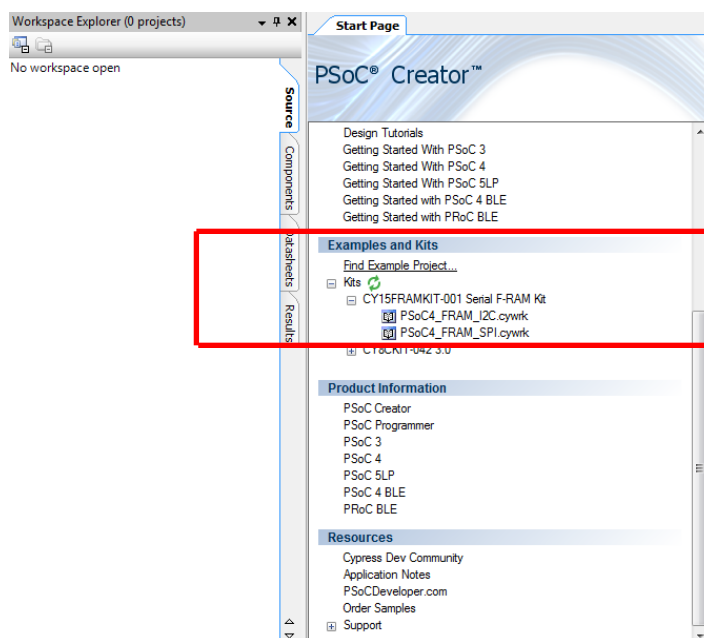
- Arduino\_FRAM\_SPI
- Arduino\_FRAM\_I2C

### 5.1 编程 PSoC 4 Pioneer 套件

按照下面的步骤来编程 PSoC 4 示例项目（PSoC4\_FRAM\_SPI 项目在这显示）：

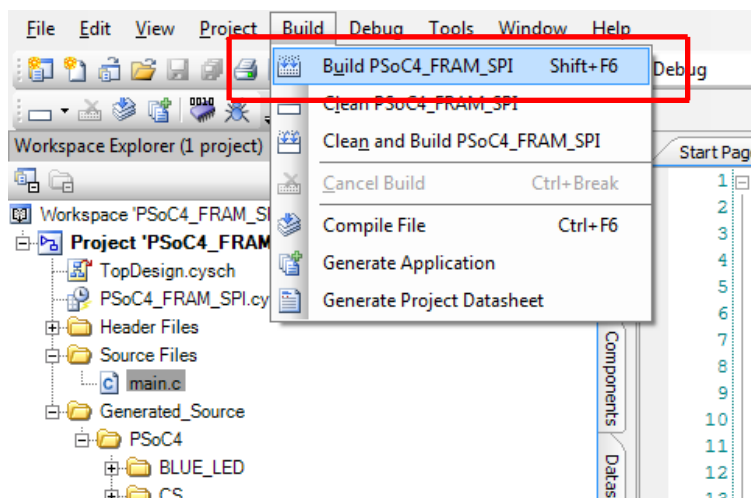
1. 依次点击 **Start > All Programs > Cypress > PSoC Creator 3.1 > PSoC Creator 3.1** 的选项来启动 PSoC Creator。如果未安装 PSoC Creator，通过 [www.cypress.com/psoccreator](http://www.cypress.com/psoccreator) 网页下载它。
2. 打开 Start Page，然后选择 **Examples and Kits** 下面的 **Kits > CY15FRAMKIT-001**。这时会出现示例项目列表，如图 5-1 所示。点击您需要的示例项目。

图 5-1. 打开 PSoC Creator 中的 ‘Example Project’ 选项



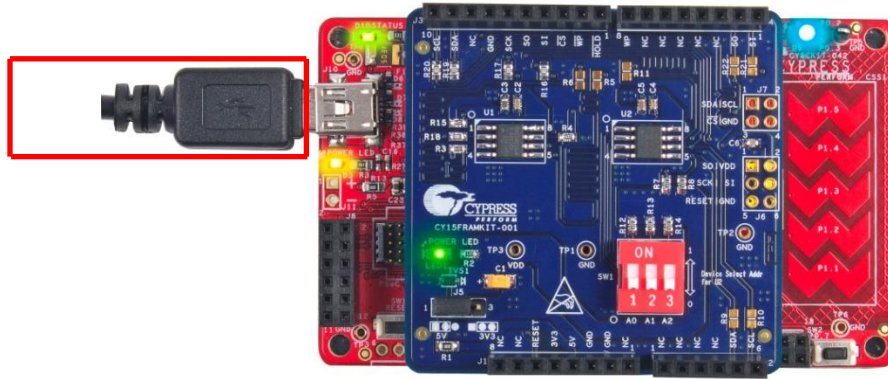
3. 选择用于存储项目的文件夹，然后点击 **OK**。
4. 依次选择 **Build > Build <Project name>** 来编译代码示例，并生成 hex 文件。

图 5-2. 在 PSoC Creator 中编译项目



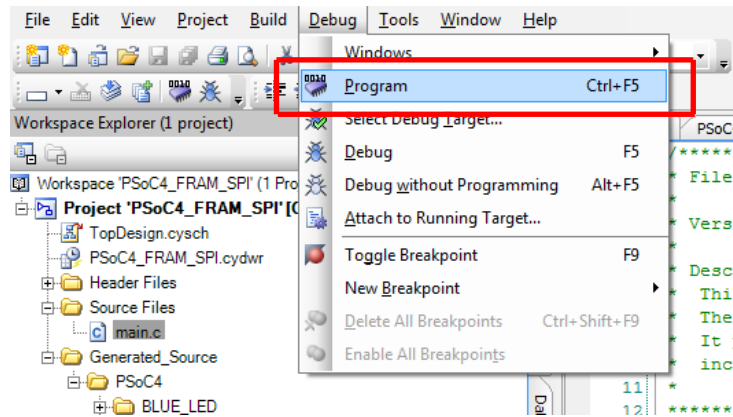
5. 选择工作电压。CY15FRAMKIT-001 可以在 3.3 V 或 5.0 V 的电压下工作，具体工作电压是通过跳线器 J5 选择的。跳线器出厂时默认被设置为 5.0 V（短接引脚 1 和 2）。Arduino UNO R3 电路板的工作电压为 5.0 V，并使用跳线器 J5 的默认设置情况。CY8CKIT-042 Pioneer 套件的工作电压为 3.3 V 或 5.0 V。Pioneer 套件默认被设置为 3.3 V。因此，根据母板上的电压设置，将跳线器 J5 设置为 5.0 V（短接引脚 1 和 2）或 3.3 V（短接引脚 2 和 3）。通过一个连接到 USB Mini-B 连接器的 USB 线缆，将 Pioneer 电路板连接至某个 PC 上，从而编程 PSoC 4，如图 5-3 所示。

图 5-3. 通过 USB 将 PSoC 4 Pioneer 套件连接到 PC 上



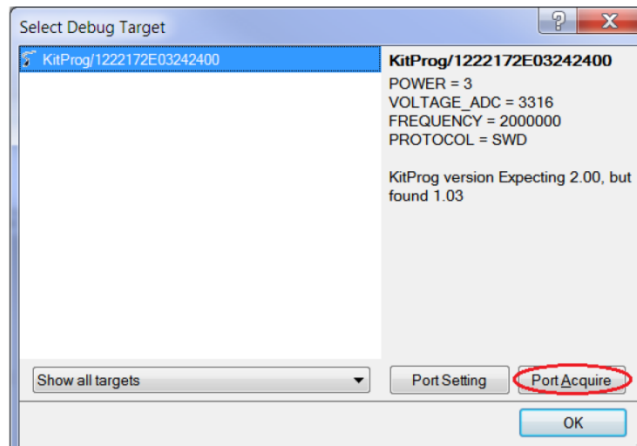
- 在 PSoC Creator 中，点击 **Debug > Program**。

图 5-4. 在 PSoC Creator 中对器件进行编程



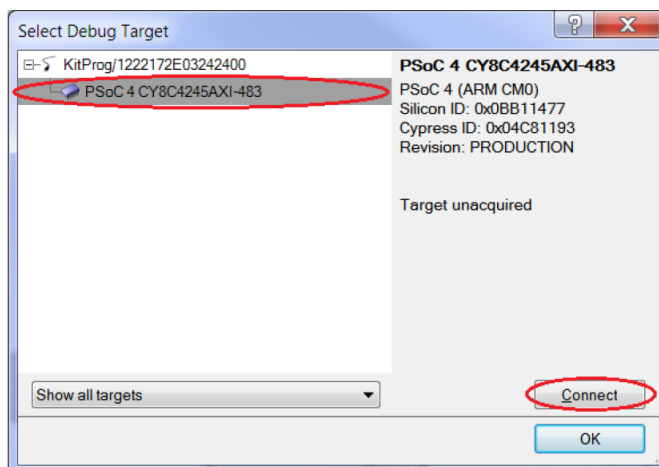
- 如果尚未选择器件，则 PSoC Creator 将打开编程窗口。选择 **KitProg** 并点击 **Port Acquire** 按钮。

图 5-5. 在 PSoC Creator 中获取器件



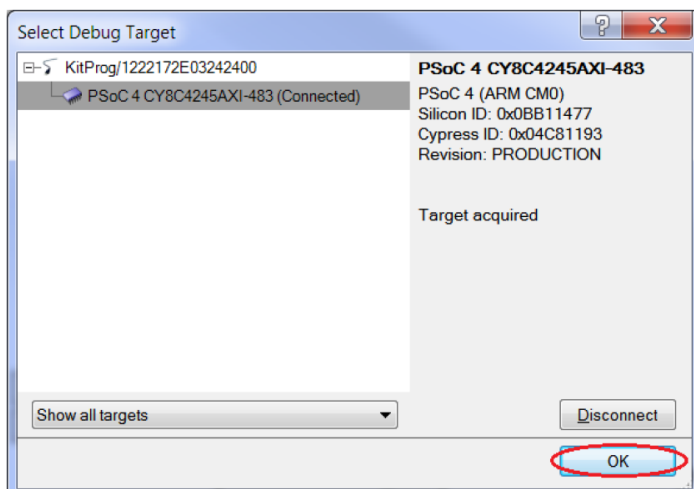
- 拿到器件后，它将显示在 **KitProg** 下的树形结构中。现在，点击 **Connect** 按钮。

图 5-6. 在 PSoC Creator 中连接器件



9. 点击 **OK**，退出该窗口并开始编程。

图 5-7. 在 PSoC Creator 中对器件进行编程

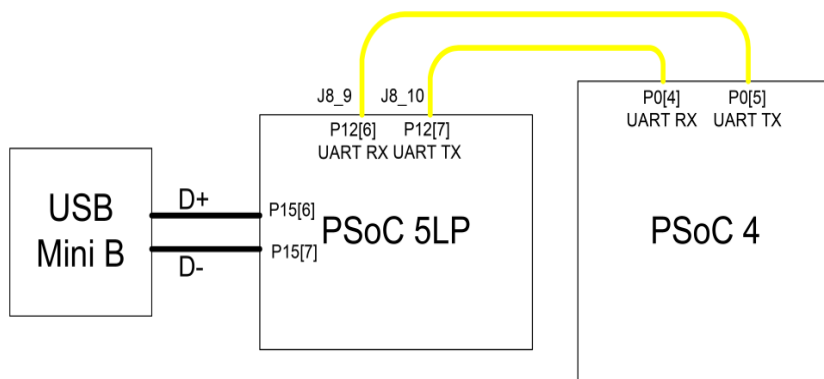


## 5.2 UART 设置

本节介绍了 UART 接口的设置情况，以便通过使用 HyperTerminal 或 PuTTY（PuTTY 可以代替 HyperTerminal，用于支持串行 COM 连接），将显示 PC 上 PSoC 4 示例项目输出。

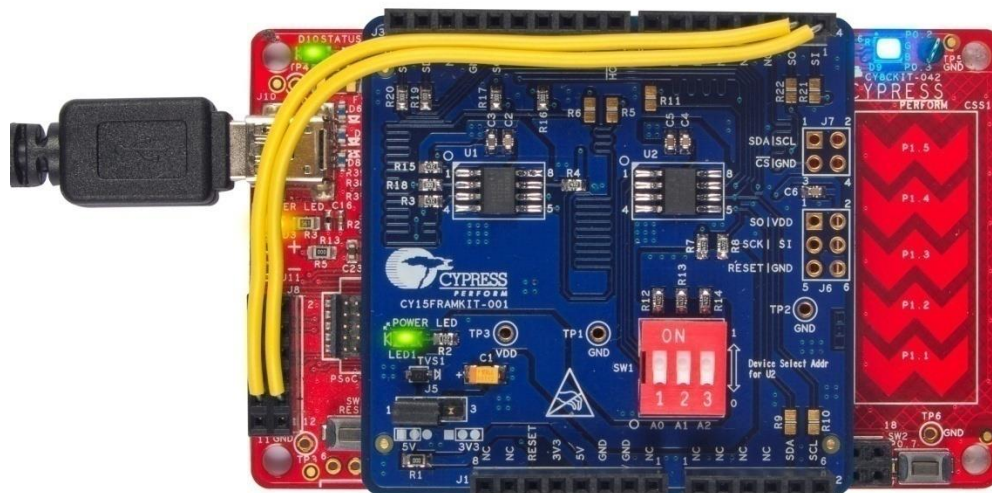
PSoC 4 Pioneer 套件上的 PSoC 5LP 器件可被配置为“USB 到 UART”桥接器，如图 5-8 所示。这两个 PSoC 4 示例项目包含用于通过 USB 端口传输结果到 PC 的 UART 通信。更多有关硬件设置和架构的信息，请参考 PSoC 4 Pioneer 套件网页：<http://www.cypress.com/?rid=77780>。

图 5-8. PSoC 4与PSoC 5LP之间使用UART进行连接的框图



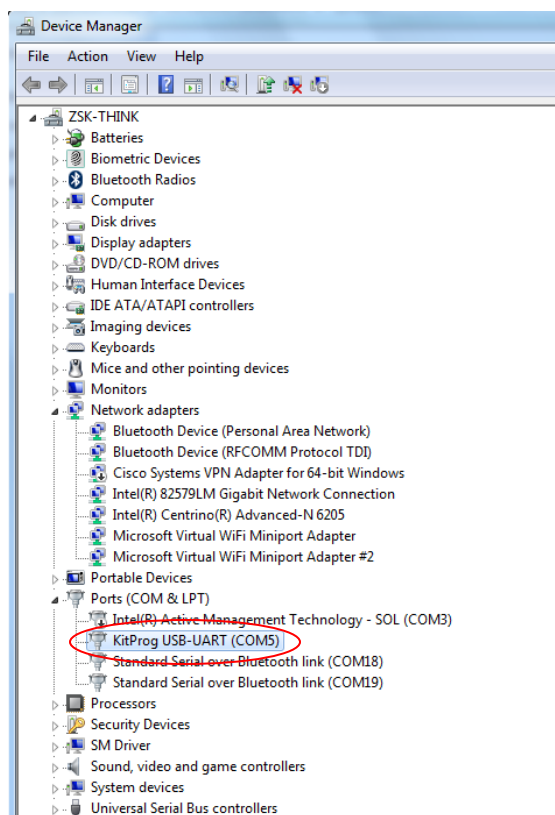
要连接PSoC 4的RX端口，可通过CY15FRAMKIT-001套件上的J4\_1（SI）连接到PSoC 4上的J8\_10（TX）。同样，连接PSoC 4的TX端口，可通过将CY15FRAMKIT-001套件上的J4\_2（SO）连接到J8\_9，如图5-9所示。

图 5-9. PSoC 4 Pioneer 套件上的 UART 连接



将 USB 微型 B 连接至 J10。该套件将枚举为 **KitProg USB-UART**，并显示在 **Device Manager > Ports (COM & LPT)**中。并将一个通信端口分配给 KitProg USB-UART。

图5-10. Device Manager（器件管理程序）中的KitProg USB-UART

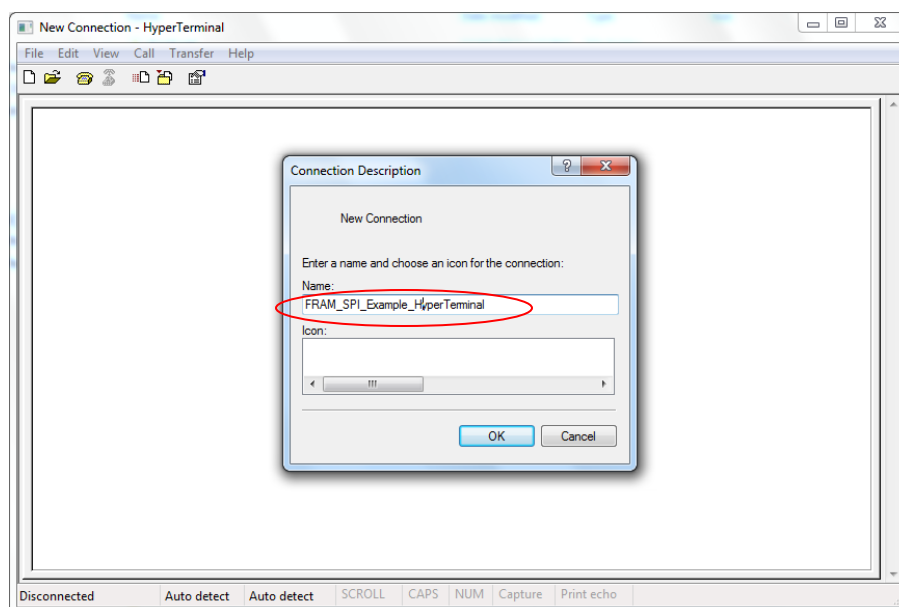


下面部分介绍了通过 PC 上的 HyperTerminal 或 PuTTY 设置串行连接的流程，从而能够实现与 PSoC 4 Pioneer 套件进行通信。

## 5.2.1 HyperTerminal 设置

打开 HyperTerminal，并依次选择 **File > New Connection**，然后输入新连接的“名称”，最后点击 **OK**。

图 5-11. 打开新连接



1. 一个新窗口被打开，在这里可以选择通信端口。

在 **Connect using** 字段中选择相应的通信端口“COMX”（或分配到 KitProg USB-UART 的特定通信端口），然后点击 **OK**，从而转到 KitProg USB-UART 的下一个设置。

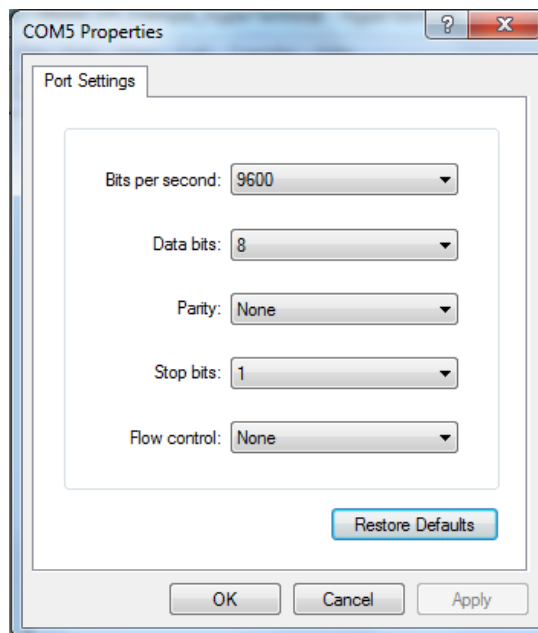
该代码示例使用 **COM5**。为您的设置验证 COM 设置，并选择相应 COMX。

图 5-12. 选择通信端口



2. 打开 **COMX Properties** 窗口会显示 **Port Settings**（端口设置）选项卡。点击 **Restore Defaults**，然后配置 ‘Bits per second’、‘Data bits’、‘Parity’、‘Stop bits’ 以及 ‘Flow control’，如图 5-13 显示，最后点击 **OK**。如果您在示例项目中更改了 PSoC 4 UART 端口设置，请确保在 HyperTerminal 中使用相同的设置。点击 **OK** 以启动通信过程。

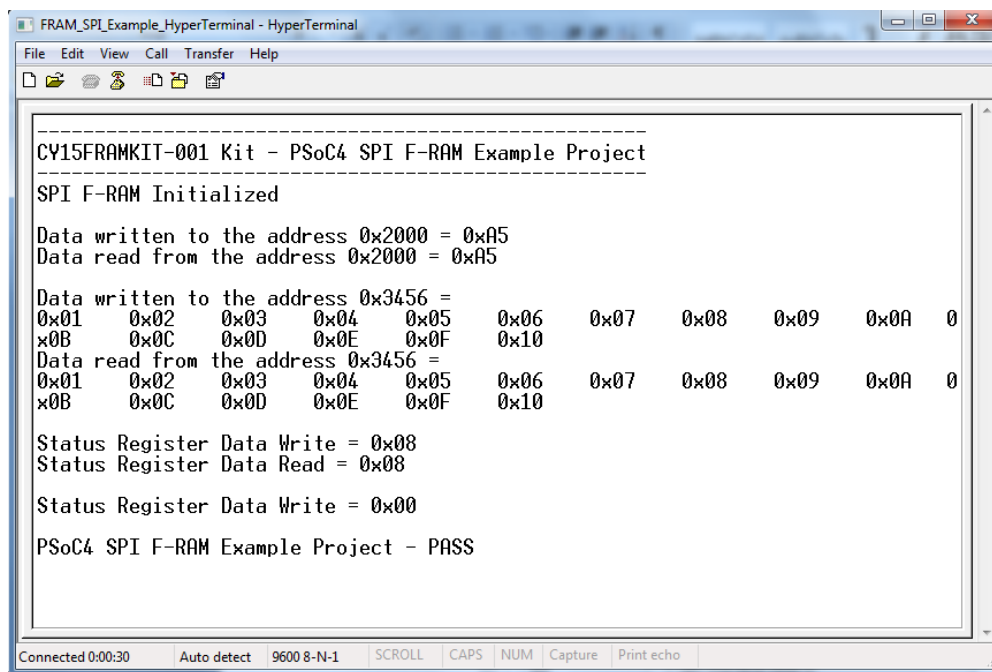
图 5-13. 配置通信端口



3. HyperTerminal 窗口被打开，以显示 PSoC 4 示例项目输出。



图 5-14. 配置通信端口

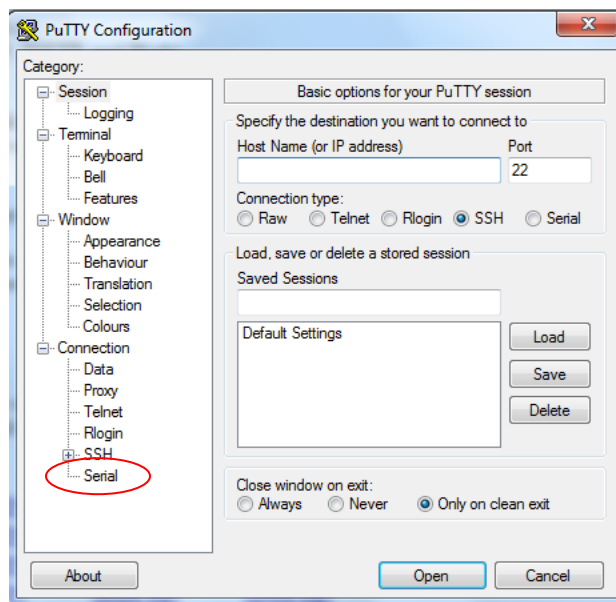


## 5.2.2 PuTTY 设置

PuTTY 是 Windows 的免费 SSH 和 Telnet 客户端。您可以从 [www.putty.org/](http://www.putty.org/) 网页上下载 PuTTY。

1. 双击 Putty 图标，并选择 **Connection** 下方的 **Serial**。

图 5-15. 打开新连接

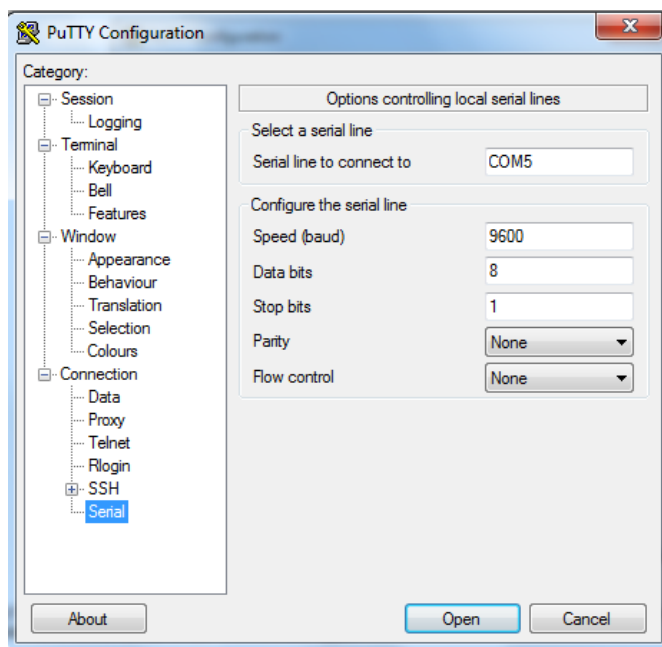


2. 将打开一个新窗口，在这里可以选择通信端口。在 **Serial line to connect to** 字段中键入 COMX。该代码示例使用 **COM5**。为您的安装内容验证 COM 设置，并选择相应的 COMX。



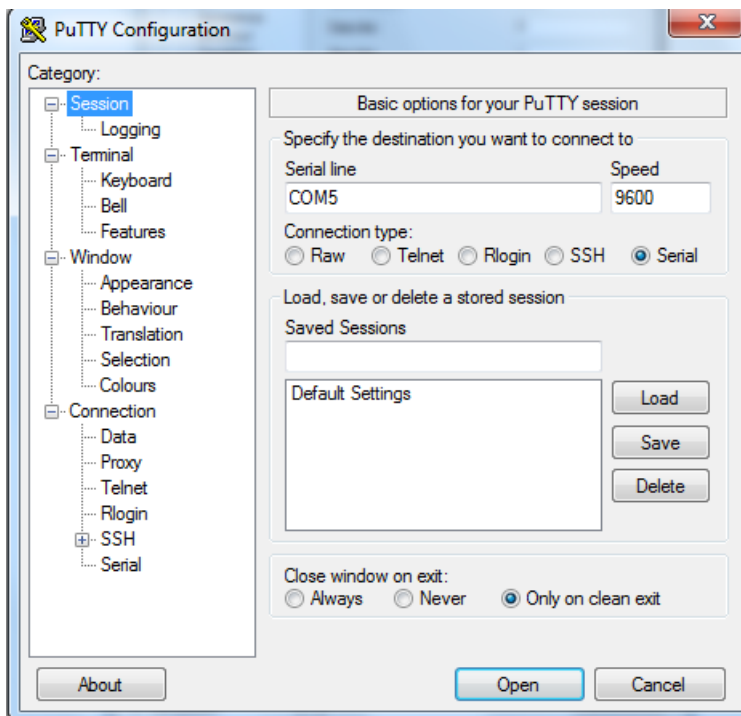
在 **Configure the serial line** 下面，选择 *Speed (baud)* 速度（波特）、*Data bits*（数据位）、*Stop bits*（停止位）、*Parity*（奇偶校验）以及 *Flow control*（流量控制）的值。在 **Connection type**（连接类型）下点击 **Session** 并选择 **Serial** 项。

图 5-16. 选择并配置通信端口



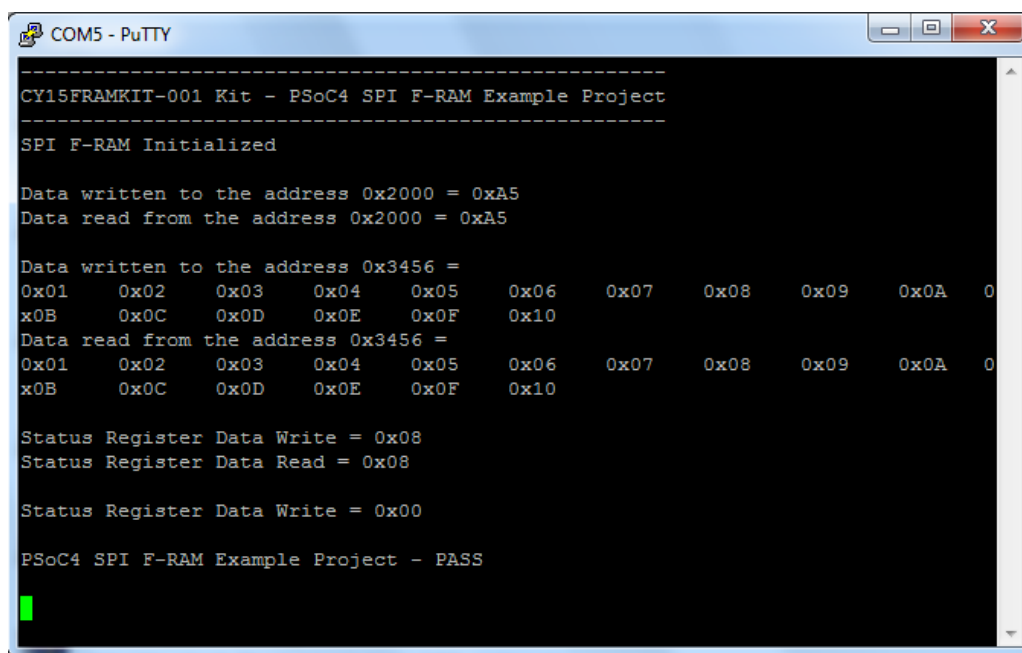
3. 选择 **Category** 下方的 **Session**，并将通信类型设置为“Serial”。然后点击 **Open**。

图 5-17. 在 PuTTY 中选择通信类型



4. COM 终端软件将显示 PSoC 4 的数据。例如，SPI 示例项目的初始屏幕，如图 5-18 所示。

图 5-18. PuTTY 上所显示的数据



```

-----
CY15FRAMKIT-001 Kit - PSoC4 SPI F-RAM Example Project
-----
SPI F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0
0x0B  0x0C  0x0D  0x0E  0x0F  0x10
Data read from the address 0x3456 =
0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0
0x0B  0x0C  0x0D  0x0E  0x0F  0x10

Status Register Data Write = 0x08
Status Register Data Read = 0x08

Status Register Data Write = 0x00

PSoC4 SPI F-RAM Example Project - PASS

```

## 5.3 项目：PSoC 4 F-RAM SPI

### 5.3.1 项目简介

该示例项目运行于 CY8CKIT-042 PSoC 4 Pioneer 套件上，并提供 SPI F-RAM 组件和 API，以便对 F-RAM 存储器/状态寄存器进行写/读操作。示例项目将数据写入到 F-RAM 内以及从 F-RAM 中读取数据，并通过以蓝色照明 PSoC 4 上的 RGB LED（D9）来指示 PASS（读写操作成功）。输出通过 UART 显示在 HyperTerminal 中。

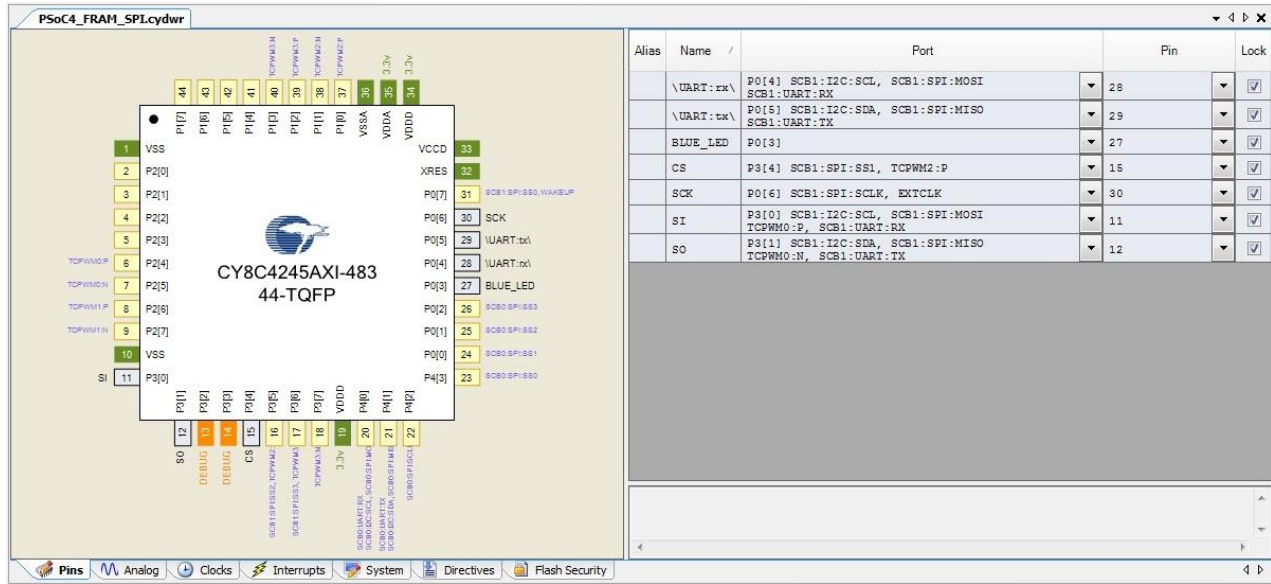
### 5.3.2 硬件连接

有关 UART 设置的信息，请参考 5.2 部分。除此之外，其他所有连接被硬件连接在电路板上。在 Workspace Explorer 中打开 *PSoC4\_FRAM\_SPI.cydwr*，并选择适合的引脚。

表 5-1. PSoC 4 F-RAM SPI I/O 引脚分配情况

F-RAM I/O	PSoC 4 引脚
CS#	P3_4
SI	P3_0
SO	P3_1
SCK	P0_6

图 5-19. PSoC 4 F-RAM SPI 示例项目引脚配置



### 5.3.3 固件流程

示例项目固件的流程如下：

- 初始化 F-RAM SPI 和 USB-UART 模块。
- 等待 (HyperTerminal / PuTTY) 显示屏出现 “Enter” 按键。
- 将 EXAMPLE\_DATA\_BYTE (0xA5) 写入到 EXAMPLE\_ADDR\_1 (0x2000) 地址中。
- 从地址 EXAMPLE\_ADDR\_1 (0x2000) 中读取字节，并将它存储在 fram\_rd\_byte 中。
- 将 EXAMPLE\_DATA\_BYTE (0xA5) 写入到 EXAMPLE\_ADDR\_1 (0x2000) 地址中。
- 从地址 EXAMPLE\_ADDR\_1 (0x2000) 中读取字节，并将它存储在 fram\_rd\_byte 中。
- 将从 data\_bytes 阵列 (1、2、3、...、16) 中的 16 个数据字节写入到 EXAMPLE\_ADDR\_2 (0x3456) 地址。
- 通过读功能从 EXAMPLE\_ADDR\_2 (0x3456) 地址中读取 16 个数据字节。
- 将 EXAMPLE\_STS\_REG\_VALUE (0x08) 写入到状态寄存器中。
- 读取状态寄存器的数据，并将其存储在 ‘statug\_reg’ 变量中。
- 对读取到的数据和写入的数据进行比较。如果操作成功，蓝色 LED 将发光。
- 清除状态寄存器的内容。

### 5.3.4 验证输出

图 5-20 到图 5-22 显示的是数据输出。

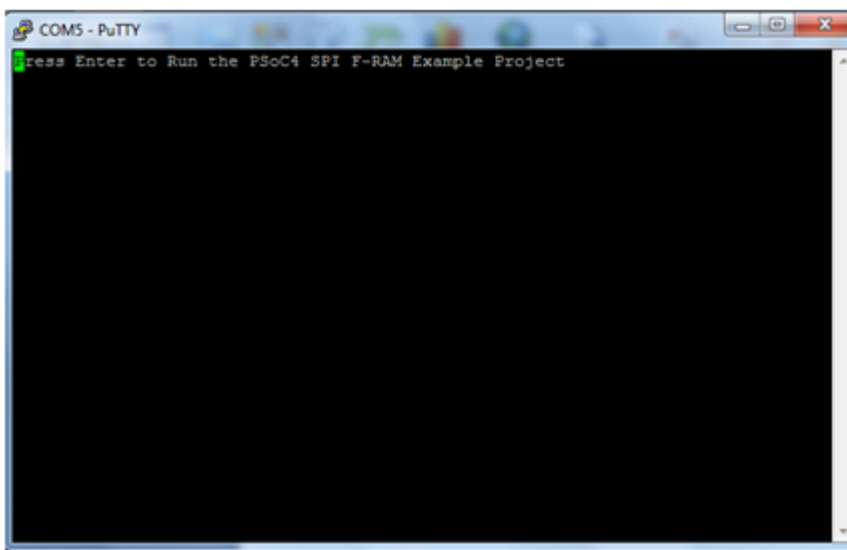
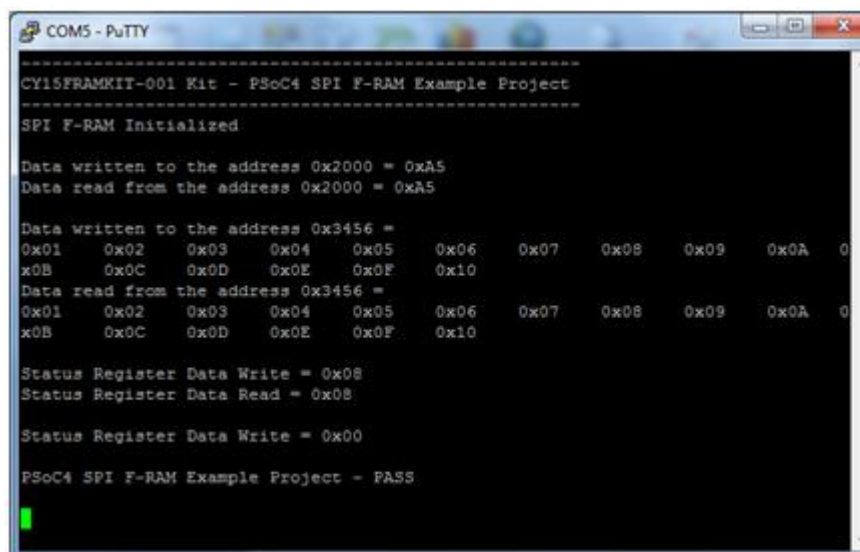
图 5-20. PSoC 4 F-RAM SPI 示例项目输出 — 等待用户按下 **Enter** 按键


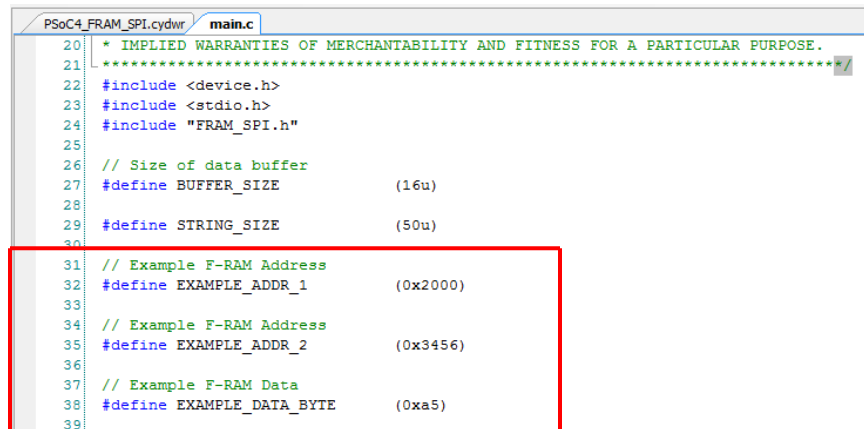
图 5-21. PSoC 4 F-RAM SPI 示例项目输出数据



### 5.3.5 修改项目

在 **Source Files** 下面，打开项目中的 **main.c** 文件，并通过设置图 5-22 中显示的 **EXAMPLE\_ADDR\_1**、**EXAMPLE\_ADDR\_2** 以及 **EXAMPLE\_DATA\_BYTE** 等变量来修改地址和数据字节值。例如，将 **0x2000** 更改为 **0x1200**、将 **0x3456** 改为 **0x1600**、将 **0xA5** 改为 **0x5a**。

图 5-22. Source File 中 **main.c** 的快照



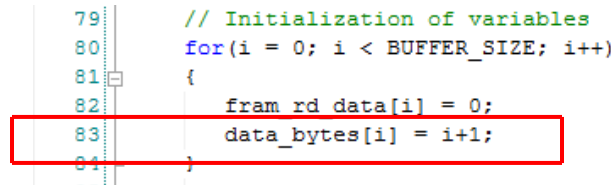
```

20  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
21  *****
22  #include <device.h>
23  #include <stdio.h>
24  #include "FRAM_SPI.h"
25
26  // Size of data buffer
27  #define BUFFER_SIZE      (16u)
28
29  #define STRING_SIZE      (50u)
30
31  // Example F-RAM Address
32  #define EXAMPLE_ADDR_1   (0x2000)
33
34  // Example F-RAM Address
35  #define EXAMPLE_ADDR_2   (0x3456)
36
37  // Example F-RAM Data
38  #define EXAMPLE_DATA_BYTE (0xa5)
39

```

另外，用户可以更改图 5-23 中显示的突发写数据。例如，在第 83 行将 **i+1** 更改为 **i+3**，这样可以初始化 0x03 – 0x13 的输入数据。

图 5-23. 用于初始化输入数据缓冲区的 **main.c** Source file 的快照



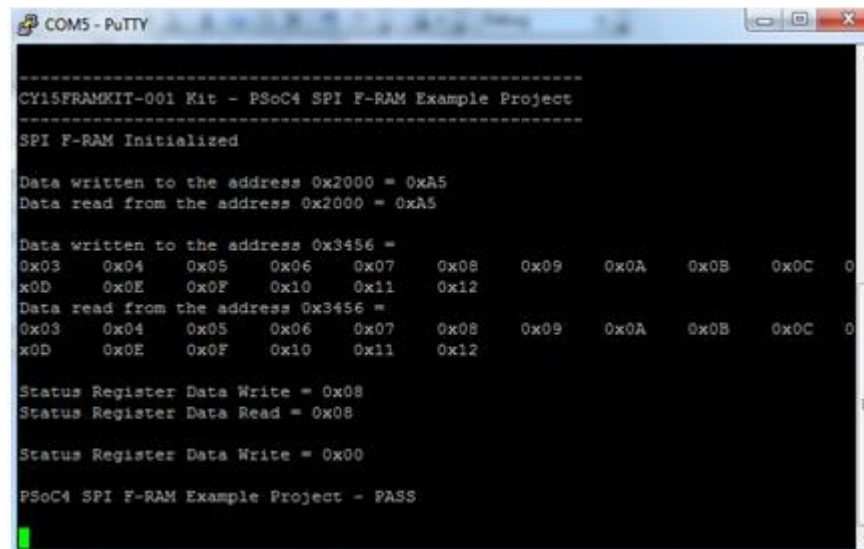
```

79  // Initialization of variables
80  for(i = 0; i < BUFFER_SIZE; i++)
81  {
82      fram_rd data[i] = 0;
83      data_bytes[i] = i+1;
84  }
85

```

使用上述更改来编译项目，并编程 Pioneer 套件。输出将显示写入到地址 **0x1200** 的数据 **0x5a** 以及从 **0x1600** 中写入的数据 **0x03 – 0x13**，如图 5-24 所示。下一节提供的 API 可以用于构建不同的应用。

图 5-24. 更后的 SPI F-RAM 项目输出



```

=====
CY15FRAMKIT-001 Kit - PSoC4 SPI F-RAM Example Project
=====
SPI F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
0xD  0x0E  0x0F  0x10  0x11  0x12
Data read from the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
0xD  0x0E  0x0F  0x10  0x11  0x12

Status Register Data Write = 0x08
Status Register Data Read = 0x08

Status Register Data Write = 0x00

PSoC4 SPI F-RAM Example Project - PASS

```

### 5.3.6 API

#### 5.3.6.1 void FRAM\_SPI\_Init(void)

说明：初始化 SPI 模块。

参数：无

返回值：无

其他影响：无

#### 5.3.6.2 uint8 FRAM\_SPI\_Write(uint32 addr, uint8 data\_write)

说明：将单数据字节写入到 F-RAM 中。

参数：uint32 addr: 用于写操作的 32 位 F-RAM 地址。  
uint8 data\_write: 被写入的数据字节。

返回值：uint8: 0 — 通信错误  
1 — 成功

其他影响：无

#### 5.3.6.3 uint8 FRAM\_SPI\_Read(uint32 addr, uint8 \*data\_read)

说明：读取 F-RAM 中的一个数据字节。

参数：uint32 addr: 用于读操作的 32 位 F-RAM 地址  
uint8 \*data\_read.: 用于保持数据的 8 位指针变量

返回值：uint8: 0 — 通信错误  
1 — 成功

其他影响：无

#### 5.3.6.4 uint8 FRAM\_SPI\_BurstWrite(uint32 addr, uint8 \*data\_write\_ptr, uint32 total\_data\_count)

说明：将 total\_data\_count 字节数据写入到 F-RAM 内。

参数：uint32 addr: 用于写操作的 32 位 F-RAM 地址。  
uint8 \*data\_write\_ptr: 指针到被写入的数据字节阵列。  
uint32 total\_data\_count: 被写入的数据字节数。

返回值：uint8: 0 — 通信错误  
1 — 成功

其他影响：无

#### 5.3.6.5 uint8 FRAM\_SPI\_BurstRead(uint32 addr, uint8 \*data\_read\_ptr, uint32 total\_data\_count)

说明：从 F-RAM 中读取 total\_data\_count 字节数据。

参数：uint32 addr: 用于读操作的 32 位 F-RAM 地址。  
uint8 \*data\_read\_ptr: 指针到一个阵列，以便存储数据字节。

uint32 total\_data\_count: 被读取的数据字节数。

返回值:           uint8: 0 — 通信错误  
                       1 — 成功

其他影响:        无

### 5.3.6.6 uint8 FRAM\_SPI\_Status\_Reg\_Write (uint8 data\_byte)

说明:            F-RAM 状态寄存器写操作  
 参数:            uint8 data\_byte -> 被写入的状态寄存器一字节的数据  
 返回值:           uint8: 0 — 通信错误  
                       1 — 成功

其他影响:        无

### 5.3.6.7 uint8 FRAM\_SPI\_Status\_Reg\_Read (uint8 \* status)

说明:            F-RAM 状态寄存器读取操作  
 参数:            uint8 \* status: 指针到状态寄存器的值。  
 返回值:           uint8: 0 — 通信错误  
                       1 — 成功

其他影响:        无

## 5.4 项目：PSoC 4 F-RAM I<sup>2</sup>C

### 5.4.1 项目简介

该示例项目在 CY8CKIT-042 PSoC 4 Pioneer 套件上运行，并提供 I<sup>2</sup>C F-RAM 组件和 API，以便对 F-RAM 存储器进行写/读操作。示例项目会将数据写入到 F-RAM 内，还会从 F-RAM 中读取数据，并通过使 PSoC 4 上的 RGB LED（D9）发出蓝色光亮来指示 PASS（读写操作成功）。输出通过 UART 显示在 HyperTerminal 中。

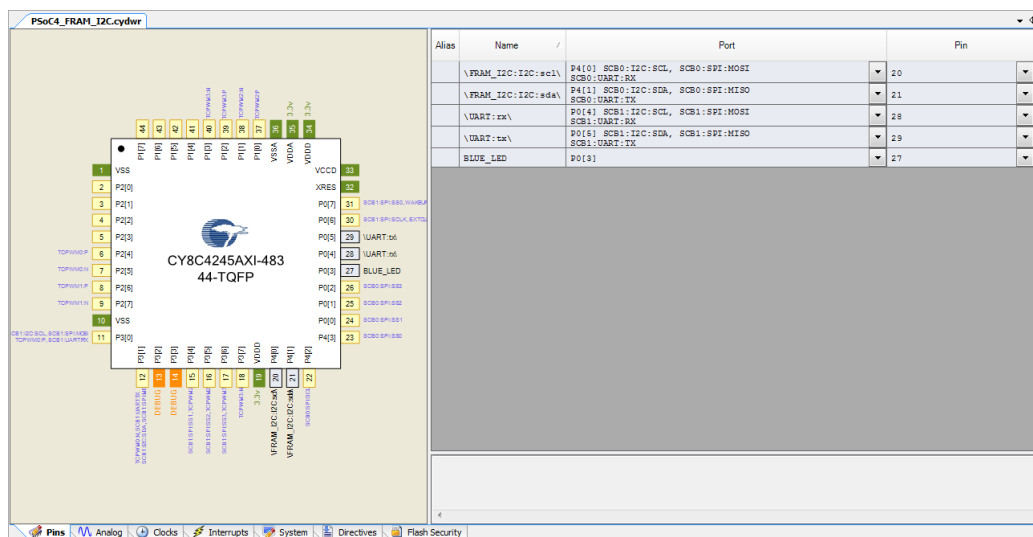
### 5.4.2 硬件连接

有关 UART 设置的信息，请参考 5.2 部分。除此之外，其他所有连接被硬件连接在电路板上。在 Workspace Explorer 中打开 PSoC4\_FRAM\_I2C.cydwr，并选择适合的引脚。

表 5-2. PSoC 4 F-RAM I<sup>2</sup>C I/O 引脚分配情况

F-RAM I/O	PSoC 4 引脚
SDA	P4_1
SCL	P4_0

图 5-25. PSoC 4 F-RAM I<sup>2</sup>C 示例项目引脚配置



### 5.4.3 固件流程

示例项目固件的流程如下：

- 初始化 F-RAM I<sup>2</sup>C 和 USB-UART 模块。
- 将 EXAMPLE\_DATA\_BYTE (0xA5)写入到 EXAMPLE\_ADDR\_1 (0x2000)地址中。
- 从地址 EXAMPLE\_ADDR\_1 (0x2000)中读取字节，并将它存储在 fram\_rd\_byte 中。
- 将从 data\_bytes 阵列（1、2、3、...、16）中的 16 个数据字节写入到 EXAMPLE\_ADDR\_2 (0x3456)地址。
- 通过随机读功能从 EXAMPLE\_ADDR\_2 (0x3456)地址中读取 15 个数据字节。
- 通过当前读功能读取第 16 个字节。
- 比较被读取的数据和被写入的数据；如果读取成功，那么蓝色 LED 会发光。

### 5.4.4 验证输出

如果读操作成功，蓝色 RGB LED 会发光。如果读操作失败，RGB LED 不发光。数据输出如图 5-26 和图 5-27 所示。

图 5-26. PSoC 4 F-RAM I<sup>2</sup>C 示例项目输出 — 等待用户按下 **Enter** 按键

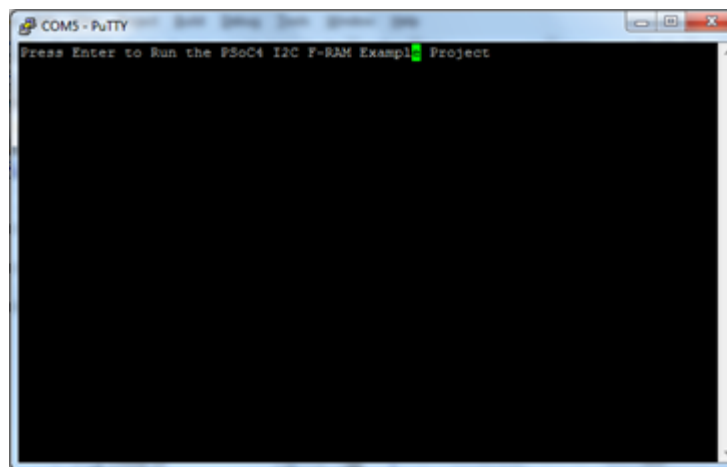
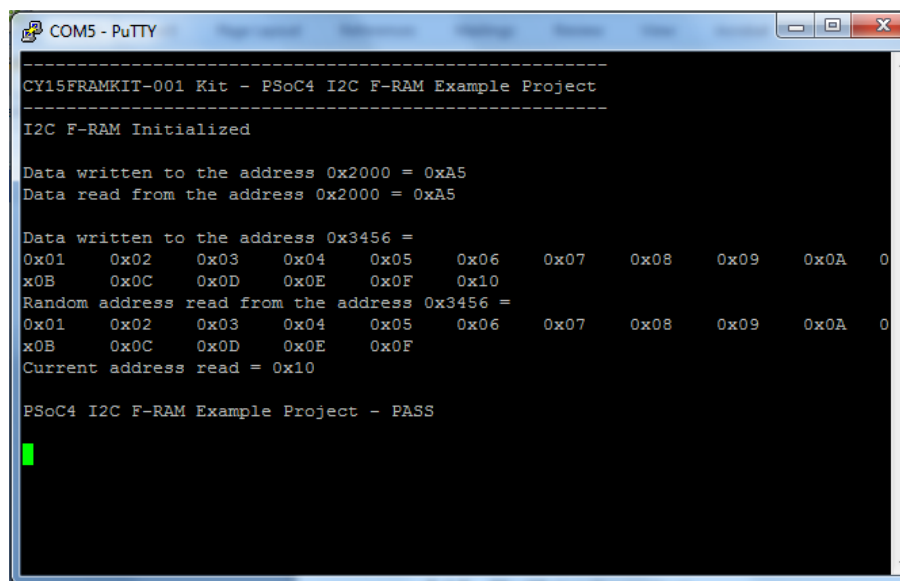




图 5-27. PSoC 4 F-RAM I<sup>2</sup>C 示例项目输出数据



```

COM5 - PuTTY
-----
CY15FRAMKIT-001 Kit - PSoC4 I2C F-RAM Example Project
-----
I2C F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

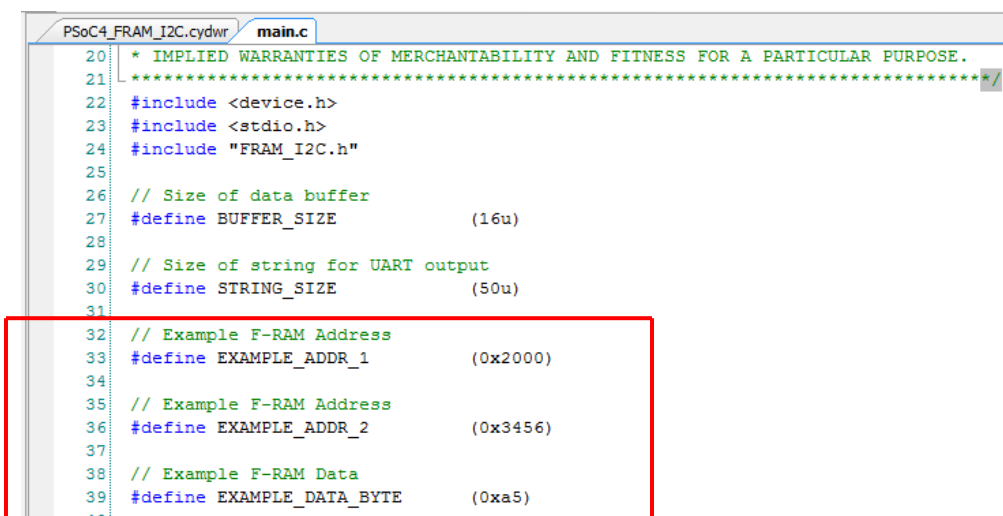
Data written to the address 0x3456 =
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0
x0B 0x0C 0x0D 0x0E 0x0F 0x10
Random address read from the address 0x3456 =
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0
x0B 0x0C 0x0D 0x0E 0x0F
Current address read = 0x10

PSoC4 I2C F-RAM Example Project - PASS
  
```

### 5.4.5 修改项目

在 **Source Files** 下面，打开项目中的 **main.c** 文件，并更改图 5-28 中显示的 EXAMPLE\_ADDR\_1、EXAMPLE\_ADDR\_2 以及 EXAMPLE\_DATA\_BYTE 值。例如，将 **0x2000** 更改为 **0x1200**、将 **0x3456** 改为 **0x1600**，并将 **0xa5** 改为 **0x5a**。

图 5-28. Source File 中 main.c 的快照

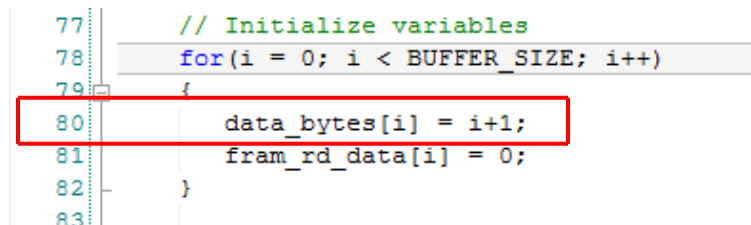


```

PSoC4_FRAM_I2C.cydwr  main.c
20  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
21  *****
22  #include <device.h>
23  #include <stdio.h>
24  #include "FRAM_I2C.h"
25
26  // Size of data buffer
27  #define BUFFER_SIZE          (16u)
28
29  // Size of string for UART output
30  #define STRING_SIZE          (50u)
31
32  // Example F-RAM Address
33  #define EXAMPLE_ADDR_1        (0x2000)
34
35  // Example F-RAM Address
36  #define EXAMPLE_ADDR_2        (0x3456)
37
38  // Example F-RAM Data
39  #define EXAMPLE_DATA_BYTE     (0xa5)
40
  
```

另外，用户可以更改图 5-29 中显示的突发写数据。例如，在第 80 行将 **i+1** 更改为 **i+3**，这样会初始化 0x03 – 0x13 的输入数据。

图 5-29. Source file 中 main.c 的快照，用于初始化输入数据缓冲区

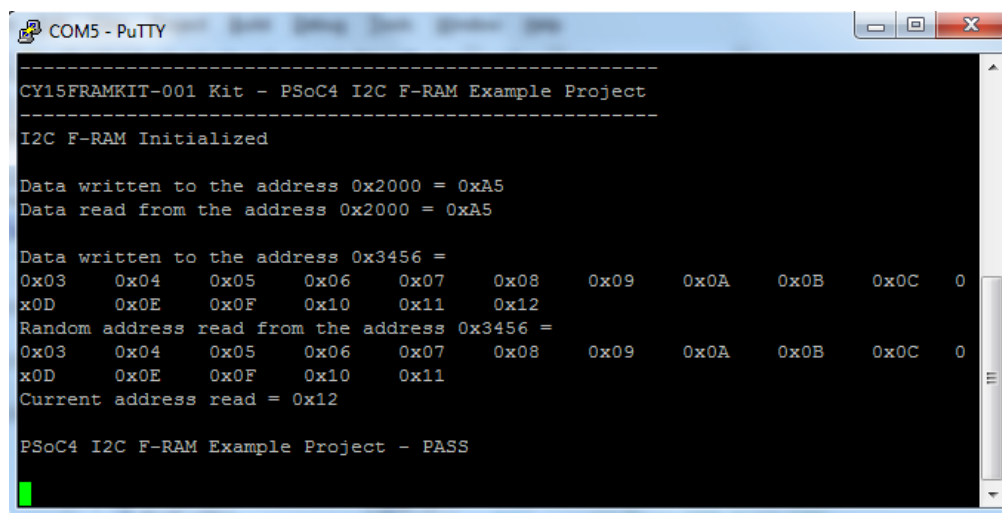


```

77  // Initialize variables
78  for(i = 0; i < BUFFER_SIZE; i++)
79  {
80  data_bytes[i] = i+1;
81  fram_rd_data[i] = 0;
82  }
83
  
```

使用上述更改来编译项目，并编程套件。输出将显示被写入到地址 **0x1200** 的数据 **0x5a** 以及从 **0x1600** 中写入的数据 **0x03 – 0x13**，如图 5-30 所示。下一节提供的 API 可以用于构建不同的应用。

图 5-30. 更后的 I<sup>2</sup>C F-RAM 项目输出



```

COM5 - PuTTY
-----
CY15FRAMKIT-001 Kit - PSoC4 I2C F-RAM Example Project
-----
I2C F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
x0D  0x0E  0x0F  0x10  0x11  0x12
Random address read from the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
x0D  0x0E  0x0F  0x10  0x11
Current address read = 0x12

PSoC4 I2C F-RAM Example Project - PASS

```

## 5.4.6 API

为 PSoC 4 F-RAM I<sup>2</sup>C 定义下面的 API。这些 API 也适用于 Arduino。

### 5.4.6.1 void FRAM\_I2C\_Init(void)

说明： 初始化 I2C 模块。

参数： 无

返回值： 无

其他影响： 无

### 5.4.6.2 uint32 FRAM\_I2C\_Write (uint8 slave\_id, uint32 addr, uint8 \*data\_write\_ptr, uint32 total\_data\_count)

说明： 将 total\_data\_count 字节数据写入到 slave\_id 指定的 F-RAM 内。

参数： uint8: 7 位从设备 ID

uint32 addr: 用于写操作的 32 位 F-RAM 地址。

uint8 \*data\_write\_ptr: 指针到被写入的数据字节阵列。

uint32 total\_data\_count: 被写入的数据字节数。

返回值： 针对 PSoC 4,

uint32: 错误状态

FRAM_I2C_MSTR_NO_ERROR	-- (0x00u)
FRAM_I2C_MSTR_ERR_ARB_LOST	-- (0x01u)
FRAM_I2C_MSTR_ERR_LB_NAK	-- (0x02u)
FRAM_I2C_MSTR_NOT_READY	-- (0x04u)
FRAM_I2C_MSTR_BUS_BUSY	-- (0x08u)

```

    FRAM_I2C_MSTR_ERR_ABORT_START  -- (0x10u)
    FRAM_I2C_MSTR_ERR_BUS_ERR      -- (0x100u)
    FRAM_I2C_MSTR_ERR              -- (0xFFu)
  
```

针对 Arduino UNO,

uint32: 错误状态

```

    成功                                -- (0x00u)
    数据太长, 不适合传输缓冲区          -- (0x01u)
    传输地址时接收的 NACK 信号          -- (0x02u)
    传输数据时所接收的 NACK 信号        -- (0x03u)
    其他错误                            -- (0x04u)
  
```

其他影响: 无

#### 5.4.6.3 uint32 FRAM\_I2C\_Current\_Read (uint8 slave\_id, uint8 \*data\_read\_ptr, uint32 total\_data\_count)

说明: 从 slave\_id 指定的 F-RAM 的当前地址中读取 total\_data\_count 字节数据。

参数: uint8: 7 位从设备 ID

uint8 \*data\_read\_ptr: 指针到一个阵列, 以便存储数据字节。

uint32 total\_data\_count: 被读取的数据字节数。

返回值: 针对 PSoC 4

uint32: 错误状态

```

    FRAM_I2C_MSTR_NO_ERROR          -- (0x00u)
    FRAM_I2C_MSTR_ERR_ARB_LOST      -- (0x01u)
    FRAM_I2C_MSTR_ERR_LB_NAK        -- (0x02u)
    FRAM_I2C_MSTR_NOT_READY         -- (0x04u)
    FRAM_I2C_MSTR_BUS_BUSY          -- (0x08u)
    FRAM_I2C_MSTR_ERR_ABORT_START   -- (0x10u)
    FRAM_I2C_MSTR_ERR_BUS_ERR       -- (0x100u)
    FRAM_I2C_MSTR_ERR               -- (0xFFu)
  
```

针对 Arduino UNO, 返回读取的字节数

其他影响: 无

#### 5.4.6.4 uint32 FRAM\_I2C\_Random\_Read (uint8 slave\_id, uint32 addr, uint8 \*data\_read\_ptr, uint32 total\_data\_count)

说明: 从 slave\_id 指定的 F-RAM 中读取 total\_data\_count 字节数据。

参数: uint8: 7 位从设备 ID

uint32 addr: 用于读操作的 32 位 F-RAM 地址。

uint8 \*data\_read\_ptr: 指针到一个阵列, 以便存储数据字节。

uint32 total\_data\_count: 被读取的数据字节数。

返回值: 针对 PSoC 4

uint32: 错误状态

```

FRAM_I2C_MSTR_NO_ERROR      -- (0x00u)
FRAM_I2C_MSTR_ERR_ARB_LOST -- (0x01u)
FRAM_I2C_MSTR_ERR_LB_NAK   -- (0x02u)
FRAM_I2C_MSTR_NOT_READY    -- (0x04u)
FRAM_I2C_MSTR_BUS_BUSY     -- (0x08u)
FRAM_I2C_MSTR_ERR_ABORT_START -- (0x10u)
FRAM_I2C_MSTR_ERR_BUS_ERR  -- (0x100u)
FRAM_I2C_MSTR_ERR          -- (0xFFu)

```

针对 Arduino UNO，返回读取的字节数

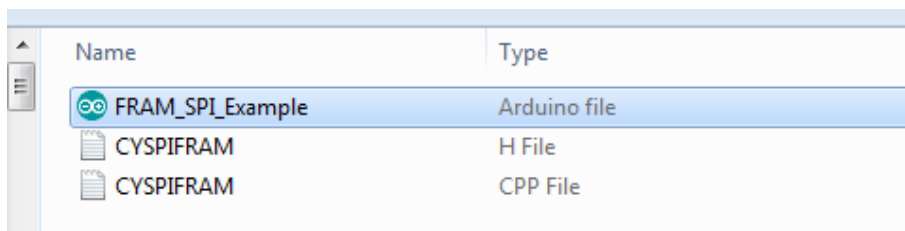
其他影响: 无

## 5.5 编程 Arduino UNO 套件

按照下面的步骤来编程 Arduino 示例项目（Arduino\_FRAM\_SPI 项目在这显示）：

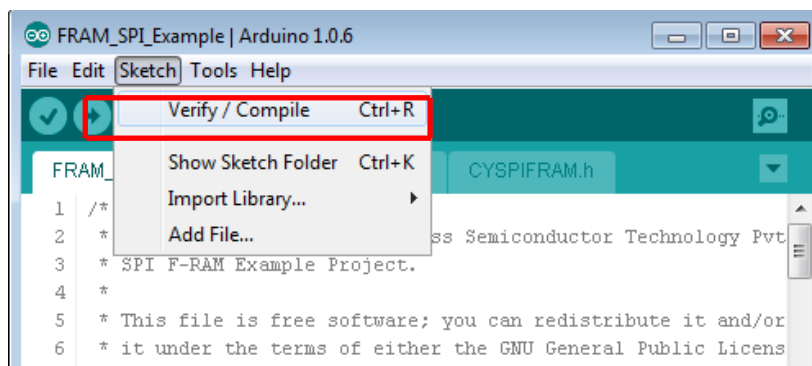
1. 安装 Arduino IDE。可以从 Arduino 网页：<http://arduino.cc/en/Main/Software> 上下载 Arduino IDE。
2. 打开“Arduino\_FRAM\_SPI”项目文件夹，并双击 **FRAM\_SPI\_Example.ino** 文件。

图 5-31. Arduino 项目文件被打开



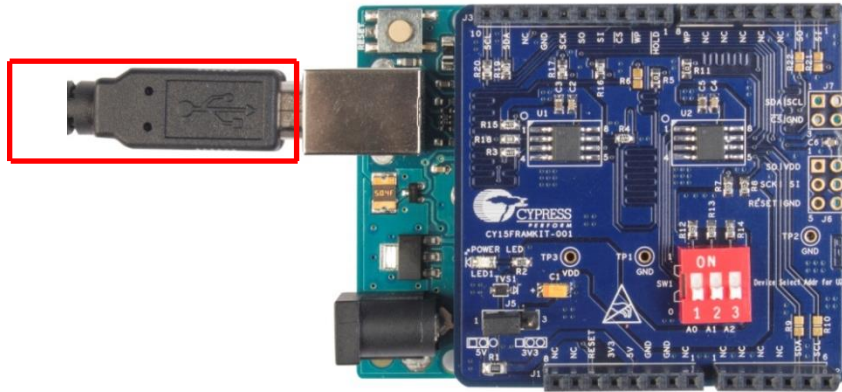
3. 依次点击 **Sketch > Verify / Compile**，从而编译代码示例。编译成功后，IDE 将显示“Done compiling.”（编译完成）状态，并生成二进制草图。

图 5-32. 编译 Arduino 项目



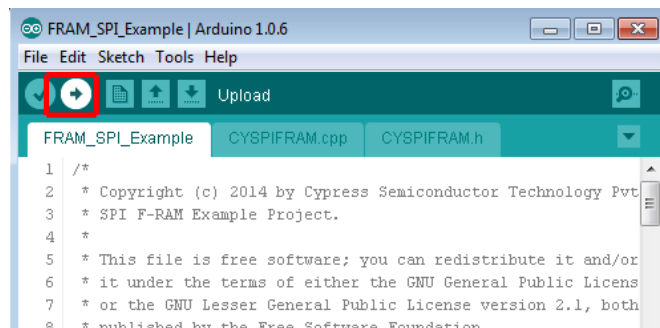
4. 要编程 Arduino UNO 套件，请使用 Arduino UNO 兼容的 USB 线缆通过 USB 端口将 PC 连接好 Arduino 电路板，如图 5-33 所示。因为 Arduino UNO 套件的工作电压为 5 V，所以请确保 CY15FRAMKIT-001 上的电源选择跳线器（J5）被放置在引脚 1 和引脚 2 之间，从而能使该套件在 5 V 电压下运行。

图 5-33. 在 Arduino UNO R3 套件上安装的 CY15FRAMKIT-001



5. 通过点击 **Upload** 按钮来下载二进制草图文件。

图 5-34. 编程 Arduino UNO R3 套件



## 5.6 项目：Arduino F-RAM SPI

### 5.6.1 项目简介

该示例项目在 Arduino UNO 电路板上运行，并提供了 SPI API，用于对 F-RAM 存储器/状态寄存器执行写/读操作。示例项目将数据写入 F-RAM 中以及从 F-RAM 中读取数据，并显示串行输出窗口的结果。

### 5.6.2 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。默认情况下，Arduino SPI 库使用下面各引脚。

表 5-3. Arduino F-RAM SPI I/O 引脚分配情况

F-RAM I/O	Arduino 引脚
CS#	D10
SI	D11
SO	D12
SCK	D13

### 5.6.3 固件流程

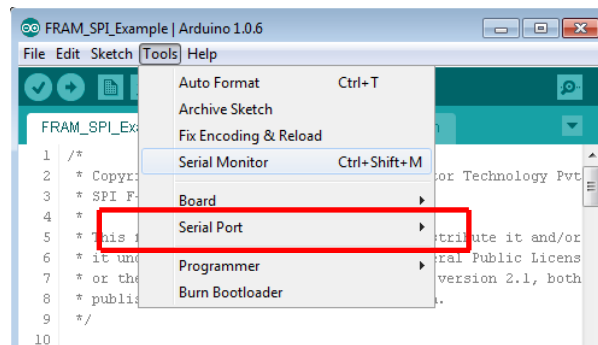
示例项目固件的流程如下：

- 初始化 F-RAM SPI。
- 将 EXAMPLE\_DATA\_BYTE (0xA5)写入到 EXAMPLE\_ADDR\_1 (0x2000)地址中。显示串行输出上被写入的数据。
- 读取 EXAMPLE\_ADDR\_1 地址中的一个字节，以便初始化 F-RAM SPI。
- 将 EXAMPLE\_DATA\_BYTE (0xA5)写入到 EXAMPLE\_ADDR\_1 (0x2000)地址中。显示串行输出上被写入的数据。
- 从地址 EXAMPLE\_ADDR\_1 (0x2000)中读取字节，并将它存储在 fram\_rd\_byte 中。显示串行输出上被读取的数据。
- 比较被写入的数据和被读取的数据，并显示串行输出上的结果（PASS/FAIL）。
- 将从 data\_bytes 阵列（1、2、3、...、16）中的 16 个数据字节写入到 EXAMPLE\_ADDR\_2 (0x3456)地址。
- 通过读功能从 EXAMPLE\_ADDR\_2 (0x3456)地址中读取 16 个数据字节。
- 比较被写入的数据和被读取的数据，并显示串行输出上的结果（PASS/FAIL）。
- 将 EXAMPLE\_STS\_REG\_VALUE (0x08)写入到状态寄存器中。
- 读取状态寄存器的数据，并将其存储在 ‘statug\_reg’ 变量中。
- 比较被写入的数据和被读取的数据，并显示串行输出上的结果（PASS/FAIL）。
- 清除状态寄存器的内容。

### 5.6.4 验证输出

打开 **Tools >Serial Port**，然后为通信选择相应的 COM 端口，如图 5-35 所示。

图 5-35. Arduino IDE 中的串行端口



依次打开 **Tools > Serial Monitor**，以查看串行输出，如图 5-36 和图 5-37 所示。

图 5-36. Arduino IDE 中的串行监控器

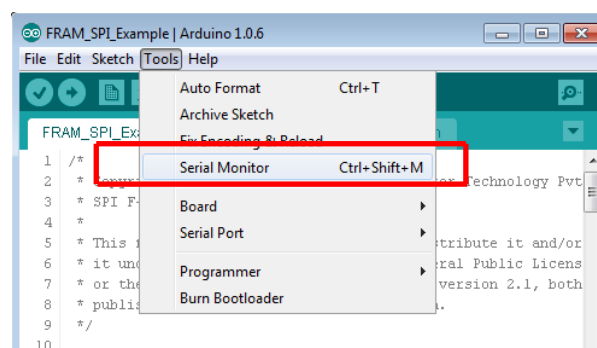


图 5-37. SPI F-RAM 项目的串行监控器输出

```
COM20
F-RAM SPI Example Project Start
1. Write Data A5
2. Read Data A5
Write-Read : PASS
3. Write Data Array
1 2 3 4 5 6 7 8 9 A B C D E F 10
4. Read Data Array
1 2 3 4 5 6 7 8 9 A B C D E F 10
Write-Read : PASS
5. Status Register Write 8
6. Status Register Read 8
Status Register Write-Read : PASS
F-RAM SPI Example Project End
```

## 5.6.5 API

### 5.6.5.1 void FRAM\_SPI\_Init()

说明： 初始化 SPI 库

参数： 无

返回值： 无

其他影响： 无

### 5.6.5.2 void FRAM\_SPI\_Write(uint32 addr, uint8 data\_write)

说明： 将单数据字节写入到 F-RAM 中。

参数： uint32 addr: 用于写操作的 32 位 F-RAM 地址。  
uint8 data\_write: 被写入的数据字节。

返回值： 无

其他影响： 无

### 5.6.5.3 uint8 FRAM\_SPI\_Read(uint32 addr)

说明： 读取 F-RAM 中的一个数据字节。

参数： uint32 addr: 用于读操作的 32 位 F-RAM 地址。

返回值： uint8: 数据读取

其他影响： 无

### 5.6.5.4 void FRAM\_SPI\_BurstWrite(uint32 addr, uint8 \*data\_write\_ptr, uint32 total\_data\_count)

说明： 将 total\_data\_count 字节数据写入到 F-RAM 内。

参数： uint32 addr: 用于写操作的 32 位 F-RAM 地址。  
uint8 \*data\_write\_ptr: 指针到被写入的数据字节阵列。  
uint32 total\_data\_count: 被写入的数据字节数。

返回值： 无

其他影响： 无



### 5.6.5.5 void FRAM\_SPI\_BurstRead(uint32 addr, uint8 \*data\_read\_ptr, uint32 total\_data\_count )

说明: 从 F-RAM 中读取 total\_data\_count 字节数据。

参数: uint32 addr: 用于读操作的 32 位 F-RAM 地址。  
uint8 \*data\_read\_ptr: 指针到一个阵列, 以便存储数据字节。  
uint32 total\_data\_count: 被读取的数据字节数。

返回值: 无

其他影响: 无

### 5.6.5.6 void FRAM\_SPI\_Status\_Reg\_Write (uint8 data\_byte)

说明: F-RAM 状态寄存器写操作

参数: uint8 data\_byte -> 被写入的状态寄存器的一个字节数据

返回值: 无

其他影响: 无

### 5.6.5.7 uint8 FRAM\_SPI\_Status\_Reg\_Read (void)

说明: F-RAM 状态寄存器读取操作

参数: 无

返回值: uint8: 一字节的寄存器数据

其他影响: 无

## 5.7 项目: Arduino F-RAM I<sup>2</sup>C

### 5.7.1 项目简介

该示例项目运行于 Arduino UNO 电路板上, 并提供 I<sup>2</sup>C API, 以便对 F-RAM 存储器进行写/读操作。示例项目将数据写入 F-RAM 中, 从 F-RAM 中读取数据, 并显示串行输出窗口的结果。

### 5.7.2 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。默认情况下, Arduino WIRE 库 (I<sup>2</sup>C) 使用下面引脚。

表 5-4. Arduino F-RAM I<sup>2</sup>C I/O 引脚分配情况

F-RAM I/O	Arduino 引脚
SDA	A4
SCL	A5

### 5.7.3 固件流程

示例项目固件的流程如下:

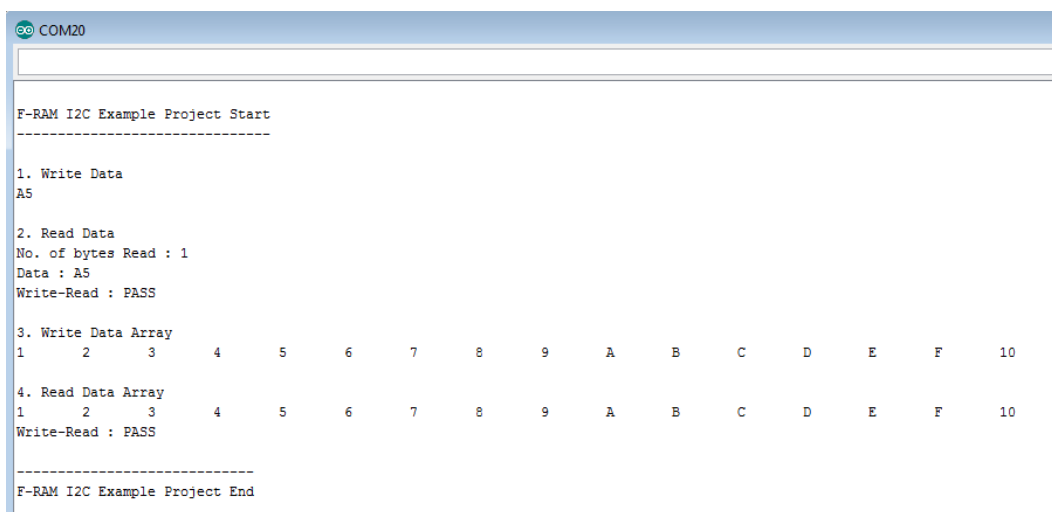
- 初始化 F-RAM I<sup>2</sup>C, 并提供启动延迟。
- 将 EXAMPLE\_DATA\_BYTE (0xA5) 写入到 EXAMPLE\_ADDR\_1 (0x2000) 地址中。

- c. 从地址 EXAMPLE\_ADDR\_1 (0x2000)中读取字节，并将它存储在 fram\_rd\_byte 中。显示串行输出上被读取的数据。
- d. 比较被写入的数据和被读取的数据，并显示串行输出上的结果 (PASS/FAIL)。
- e. 将从 data\_bytes 阵列 (1、2、3、...、16) 中的 16 个数据字节写入到 EXAMPLE\_ADDR\_2 (0x3456)地址。
- f. 通过随机读功能从 EXAMPLE\_ADDR\_2 (0x3456)地址中读取 15 个数据字节。
- g. 通过当前读功能读取第 16 个字节。
- h. 比较被写入的数据和被读取的数据，并显示串行输出上的结果 (PASS/FAIL)。

## 5.7.4 验证输出

依次打开 **Tools** → **Serial Monitor**，以查看串行输出，如图 5-38 所示。

图 5-38. I<sup>2</sup>C F-RAM 项目的串行监控器输出



```

COM20

F-RAM I2C Example Project Start
-----

1. Write Data
A5

2. Read Data
No. of bytes Read : 1
Data : A5
Write-Read : PASS

3. Write Data Array
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10

4. Read Data Array
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10
Write-Read : PASS

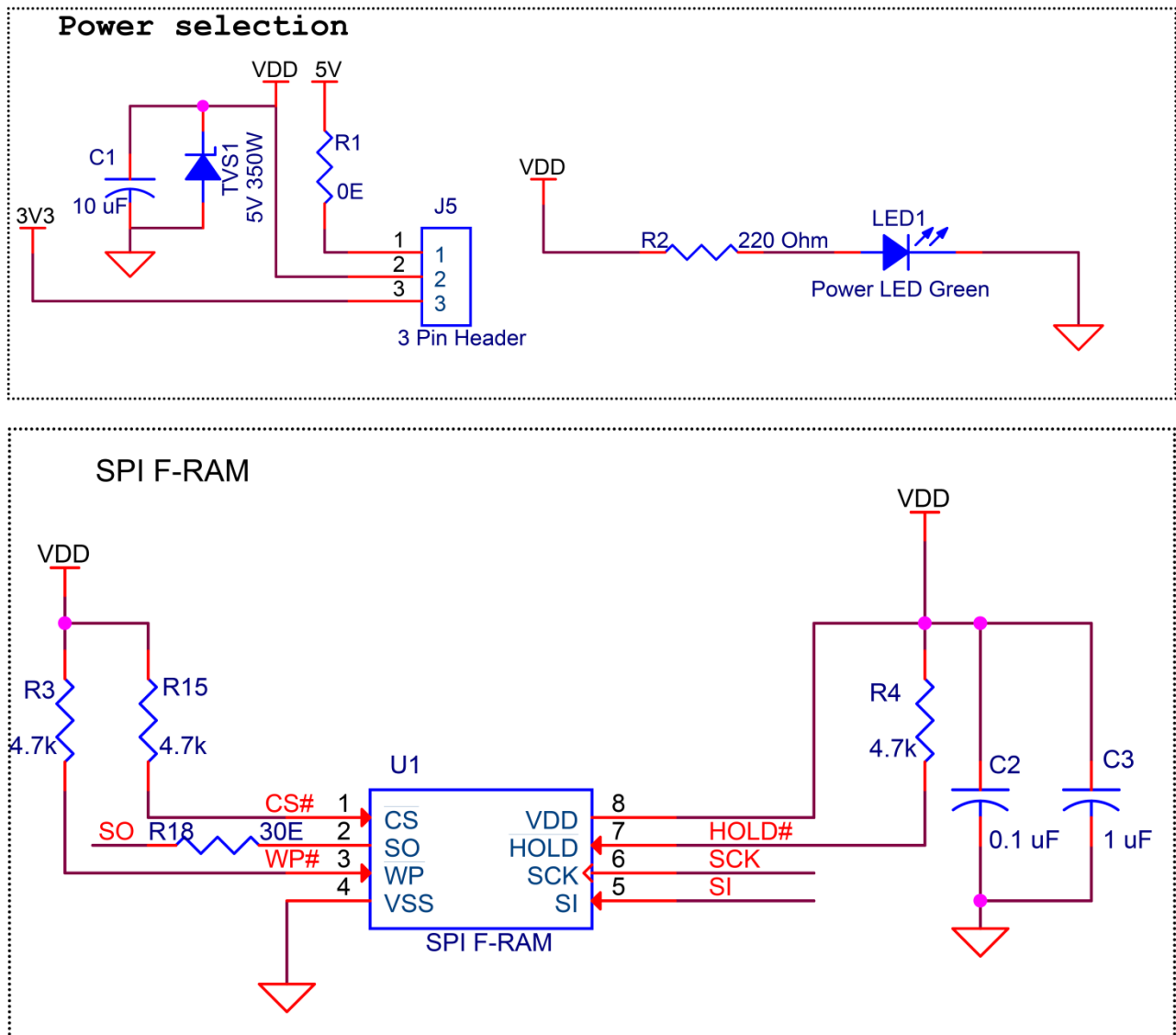
-----
F-RAM I2C Example Project End

```

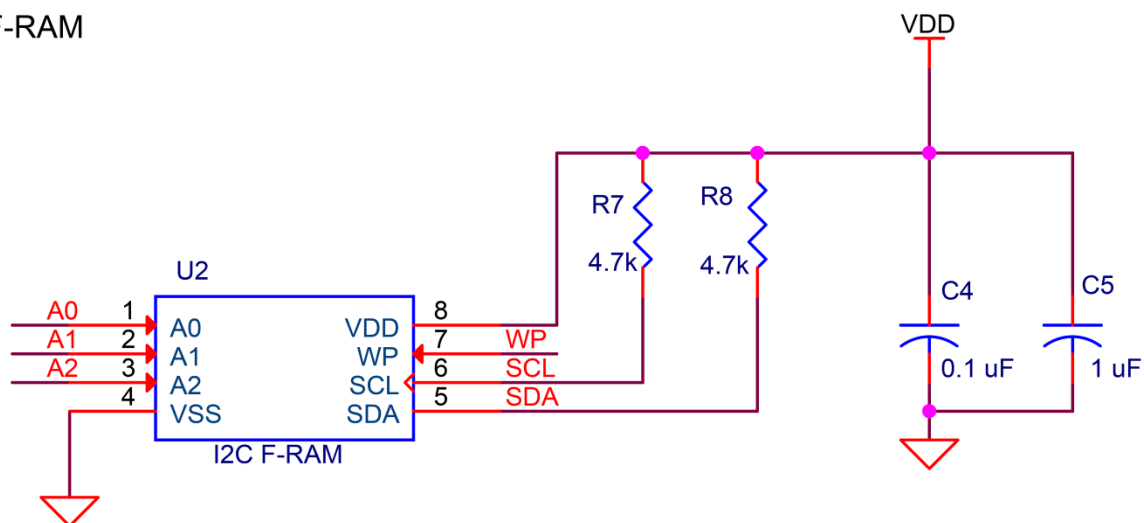
## 5.7.5 API

API 表明同 PSoc 4 I2C F-RAM 的一样。请参考 5.4.6 一节

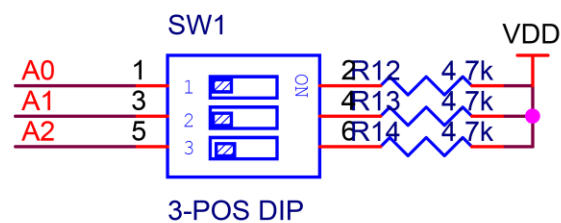
## A.1 CY15FRAMKIT-001 原理图



## I2C F-RAM

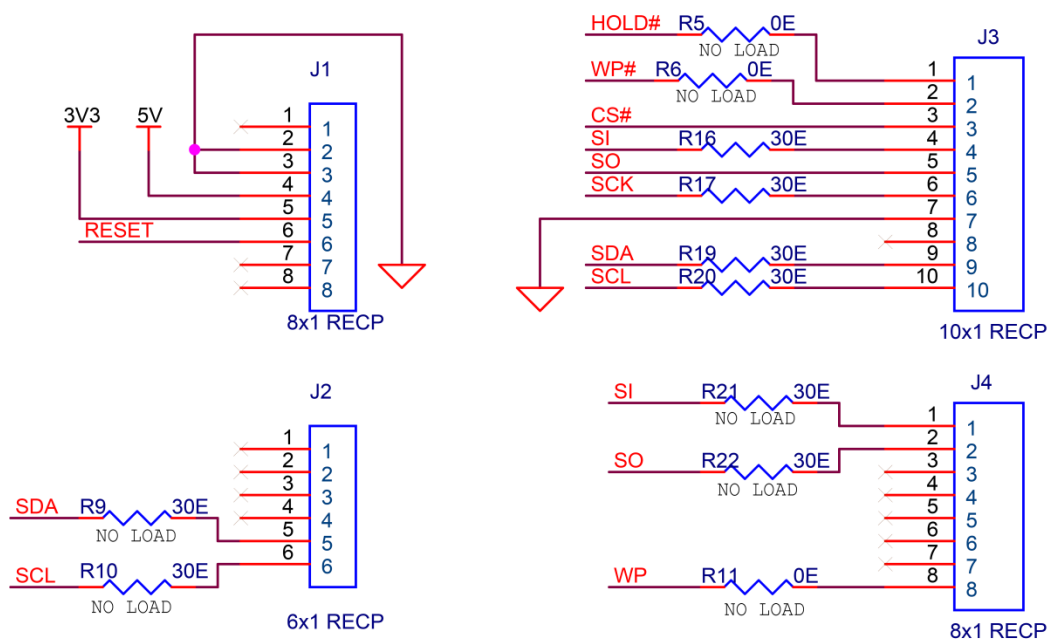


## I2C F-RAM Device Select Address

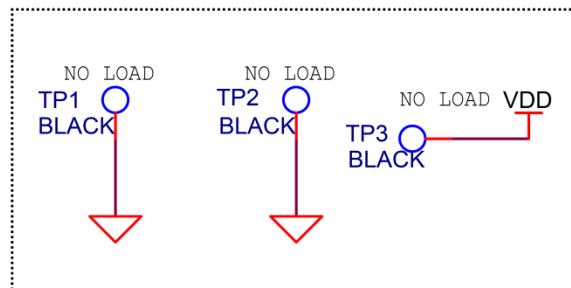
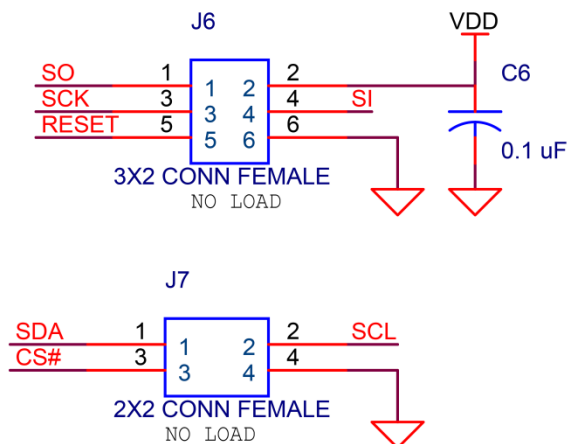


By default, A0, A1, A2 are pulled down internally

### (J1-J4) Arduino Compatible Headers



### Debug Header



## 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

### Arduino 兼容的插座 (J1、J2、J3、J4)

J1			
引脚	Arduino 信号	PSoC 4 Pioneer 套件信号	CY15FRAMKIT-001 套件信号
J1_1	VIN	VIN	NC
J1_2	GND	GND	GND
J1_3	GND	GND	GND
J1_4	5 V	V5.0	5 V
J1_5	3.3 V	V3.3	3V3
J1_6	RESET#	RESET#	RESET#
J1_7	IOREF	P4_VDD	NC
J1_8	NC	NC	NC

J2			
引脚	Arduino 信号	PSoC 4 Pioneer 套件信号	CY15FRAMKIT-001 套件信号
J2_1	A0	P2[0]	NC
J2_2	A1	P2[1]	NC
J2_3	A2	P2[2]	NC
J2_4	A3	P2[3]	NC
J2_5	A4 /SDA (I <sup>2</sup> C)	P2[4]	SDA
J2_6	A5/SCL (I <sup>2</sup> C)	P2[5]	SCL

J3			
引脚	Arduino 信号	PSoC 4 Pioneer 套件信号	CY15FRAMKIT-001 套件信号
J3_1	D8	P2[6]	HOLD# (SPI F-RAM)
J3_2	D9 (PWM)	P3[6]	WP# (SPI F-RAM)
J3_3	D10 (PWM/SS)	P3[4] / CS (SCB1)	CS# (SPI F-RAM)
J3_4	D11 (PWM / MOSI)	P3[0] / MOSI (SCB1)	SI
J3_5	D12 (MISO)	P3[1] / MISO (SCB1)	SO
J3_6	D13 (SCK)	P0[6] / SCK (SCB1)	SCK
J3_7	GND	GND	GND
J3_8	AREF	P1[7]	NC

J3_9	SDA	P4[1] / SDA (SCB0)	SDA
J3_10	SCL	P4[0] / SCL (SCB0)	SCL

J4			
引脚	Arduino 信号	PSoC 4 Pioneer 套件信号	CY15FRAMKIT-001 套件信号
J4_1	D0 (UART RX)	P0[4] / SCL (SCB1)	NC
J4_2	D1 (UART TX)	P0[5] / SDA (SCB1)	NC
J4_3	D2	P0[7]	NC
J4_4	D3 (PWM)	P3[7]	NC
J4_5	D4	P0[0]	NC
J4_6	D5 (PWM)	P3[5]	NC
J4_7	D6 (PWM)	P1[0]	NC
J4_8	D7	P2[7]	WP (I <sup>2</sup> C F-RAM)

### A.3 调试插座 I/O

J6	
引脚	调试信号
J6_1	SO
J6_2	VDD
J6_3	SCK
J6_4	SI
J6_5	RESET#
J6_6	GND

J7	
引脚	调试信号
J7_1	SDA
J7_2	SCL
J7_3	CS#
J7_4	GND



## A.4 零欧姆电阻及无负载使用说明

单位	电阻	使用说明
SPI F-RAM 器件	R5	焊接零欧姆电阻，用以控制控制器的 SPI HOLD#引脚
SPI F-RAM 器件	R6	焊接零欧姆的电阻，用以控制控制器的 SPI WP#引脚
I <sup>2</sup> C F-RAM 器件	R11	焊接零欧姆的电阻，用以控制控制器的 I <sup>2</sup> C WP 引脚

## A.5 30 欧姆大小的电阻及无负载使用说明

单位	电阻	使用说明
I <sup>2</sup> C 通信	R9	焊接一个大小为 30 欧姆的电阻用于访问 SDA，从而进行 Arduino UNO R2 电路板上的 I <sup>2</sup> C 通信
I <sup>2</sup> C 通信	R10	焊接一个大小为 30 欧姆的电阻来访问 SCL，用于 Arduino UNO R2 电路板上的 I <sup>2</sup> C 通信
SPI 通信	R21	焊接一个大小为 30 欧姆的电阻，以便将 J4_1 作为 SPI SI 引脚使用。这样将释放 J3_4 上的 SI，SI 可用于 CY8CKIT-042 Pioneer 套件上 I <sup>2</sup> C-USB 的桥接。
SPI 通信	R22	焊接一个大小为 30 欧姆的电阻，以便将 J4_2 作为 SPI S0 引脚使用。这样将释放 J3_5 上的 S0，S0 可用于 CY8CKIT-042 Pioneer 套件上的 I <sup>2</sup> C-USB 桥接。

## A.6 材料表 (BOM)

序号	数量	参考	数值	说明	制造商	制造商器件型号
1				PCB、52.07 mm X 53.59 mm、两层、Enig Finish、High Tg、蓝色阻焊层、白色丝印	Cypress	
2	1	C1	10 uF	CAP TANT 10UF 10V 20% 1206	AVX Corporation	TPSA106M010R1800
3	3	C2、C4、C6	0.1 uF	CAP CER 0.1UF 50V Y5V 0603	Murata Electronics	GRM188R71C104KA01D
4	2	C3、C5	1 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
5	2	J1、J4	8x1 RECP	CONN RCPT .100" 8PS R/A SGL TIN	Samtec Inc	SSQ-108-03-T-S
6	1	J2	6x1 RECP	CONN RCPT .100" 6POS SNGL TIN	Samtec Inc	SSQ-106-03-T-S
7	1	J3	10x1 RECP	CONN RCPT .100" 10POS SNGL TIN	Samtec Inc	SSQ-110-03-T-S
8	1	J5	3 引脚插座	CONN HEADER VERT SGL 3POS GOLD	3M	961103-6404-AR
9	1	LED1	绿色电源 LED	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
10	1	R1	0E	RES 0.0 OHM 1/8W JUMP 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
11	1	R2	220 Ohm	RES 220 OHM 1/10W 5% 0603 SMD	Panasonic-ECG	ERJ-3GEYJ221V
12	8	R3、R4、R7、R8、R12、R13、R14、R15	4.7k	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic-ECG	ERJ-3GEYJ472V
13	5	R16、R17、R18、R19、R20	30 ohm	RES SMD 30 OHM 5% 1/10W 0603	Panasonic-ECG	ERJ-3GEYJ300V
14	1	SW1	3-POS DIP	SW DIP SLIDE SPST 3 POS UNSEALD	E-Switch	KAS1103E
15	1	TVS1	5 V 350W	TVS DIODE 5VWM 14.5VC SOD323	Diodes Incorporated	SD05-7
16	1	U1	SPI F-RAM	IC FRAM 256-KBIT 40MHZ 8SOIC	Cypress	FM25W256-G
17	1	U2	I2C F-RAM	IC FRAM 256-KBIT 3.4MHZ 8SOIC	Cypress	FM24W256-G
18	1	N/A	跳线器	跳线器	3M	969102-0000-DA
无负载组件						
19	1	J6	3X2 CONN FEMALE	CONN HEADER VERT DUAL 6POS GOLD	3M	961206-6404-AR
20	1	J7	2X2 CONN FEMALE	CONN HEADER VERT DUAL 4POS GOLD	3M	961204-6404-AR
21	3	R5、R6、R11	0E	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEYJ302V
22	4	R9、R10、R21、R22	30E	RES SMD 30 OHM 5% 1/10W 0603	Panasonic - ECG	ERJ-3GEYJ300V
23	3	TP1、TP2、TP3	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001

序号	数量	参考	数值	说明	制造商	制造商器件型号
特殊跳线器的安装说明						
24	1	J5	在引脚 1 和 2 间安装跳线器	Headers & Wire Housings 2.54MM SHUNT	3M	969102-0000-DA
标签						
25	1	N/A	N/A	LBL、PCA 标签、供应商代码、生产日期、序列号 121-60201-01 REV 01 (YYWWVVXXXXXX)	Cypress Semiconductor	
26	1	N/A	N/A	LBL、CY15FRAMKIT-001 QR 代码、12mm X 12mm	赛普拉斯半导体公司	
27	1	N/A	N/A	LBL、防静电报警、ULINE PN S-6516、5/8" x 2"、“注意观察预防措施”	赛普拉斯半导体公司	

# 修订记录



## 文档修订记录

文档标题: CY15FRAMKIT-001 串行 F-RAM™开发套件指南

文档编号: 001-97577

版本	ECN	变更者	提交日期	变更说明
**	4769239	LISZ	06/01/2015	本文档版本号为 Rev**, 译自英文版 001-95689 Rev*A。