CapSense®控制器代码示例

文档编号：001-85848 版本*A

赛普拉斯半导体公司
198 Champion Court
San Jose, CA 95134-1709
电话（美国）：800.858.1810
电话（国际）：+1.408.943.2600
http://www.cypress.com

版权所有

©赛普拉斯半导体公司，2012-2016。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它电路的使用承担任何责任。也不会以明示或暗示的方式授予任何专利许可或其他权利。除非与赛普拉斯签订了明确的书面协议，否则赛普拉斯不能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 Designer™和 Programmable System-on-Chip™是赛普拉斯半导体公司的商标且 PSoC®是赛普拉斯半导体公司的注册商标。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

赛普拉斯不针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。对于发生故障（包括运转异常）或失效可能会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。若将赛普拉斯产品使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，并确保赛普拉斯不会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

简介 8

概述 8

代码示例 1　使用 CY8C20xx6A CapSense 控制器来实现基于 SmartSense 技术的按键和滑条 10

1.1 项目名称 10
1.2 概况 10
1.3 硬件设置 10
1.3.1 要求 10
1.3.2 摘要 10
1.3.3 组装电路板 11
1.4 原理图 12
1.5 软件设置 12
1.5.1 所需工具 12
1.5.2 用户模块列表 12
1.5.3 用户模块参数、全局资源 13
1.6 操作说明 13
1.7 执行代码示例 14
1.8 通过 I²C 读取 CapSense 数据 14
1.8.1 加载 Bridge Control Panel 工具 14
1.8.2 读取 BTN0 的原始计数、基准线和计数差值 15
1.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置 15

代码示例 2　通过 I²C 使 CY8C20xx6A CapSense 控制器与主机通信 16

2.1 项目名称 16
2.2 概述 16
2.3 硬件设置 16
2.3.1 要求 16
2.3.2 摘要 16
2.3.3 设置电路板 17
2.4 原理图 17
2.5 软件设置 18
2.5.1 所需工具 18

2.5.2 用户模块列表 18
2.5.3 用户模块参数、全局资源 18
2.6 操作说明 18
2.7 执行代码示例 19
2.8 通过 I²C 读取 CapSense 数据 19
2.8.1 加载 Bridge Control Panel 工具 19
2.8.2 读取 BTN0 的原始计数、基准线和计数差值 20
2.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置 20

代码示例 3 通过 UART 与 CY8C20xx6A CapSense 控制器进行数据传输 21

3.1 项目名称 21
3.2 概述 21
3.3 硬件设置 21
3.3.1 要求 21
3.3.2 摘要 21
3.3.3 组装电路板 22
3.4 原理图 23
3.5 软件设置 24
3.5.1 所需的工具 24
3.5.2 用户模块列表 24
3.5.3 用户模块参数，全局资源 24
3.6 操作说明 24
3.7 执行代码示例 25
3.8 使用 MultiChart 将 CapSense 数据绘制成图表 26

代码示例 4 使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容 28

4.1 项目名称 28
4.2 概况 28
4.3 硬件设置 28
4.3.1 要求 28
4.3.2 摘要 28
4.3.3 组装电路板 29
4.4 原理图 30
4.5 软件设置 30
4.5.1 所需工具 30
4.5.2 用户模块列表 31
4.5.3 用户模块参数、全局资源 31
4.6 操作说明 31

代码示例 5 使用 CY8C21x34/B CapSense 控制器测量传感器绝对电容 38

5.1 项目名称 38
5.2 概述 38
5.3 硬件设置 38
5.3.1 要求 38

5.3.2 摘要 38
5.3.3 组装电路板 39
5.4 原理图 40
5.5 软件设置 40
5.5.1 所需的工具 40
5.5.2 用户模块列表 41
5.5.3 用户模块参数、全局资源 41
5.6 操作说明 41
5.7 执行代码示例 42

代码示例 6 使用 CY8C20x34 CapSense 控制器测量传感器的绝对电容 47

6.1 项目名称 47
6.2 概述 47
6.3 硬件设置 47
6.3.1 要求 47
6.3.2 摘要 47
6.3.3 组装电路板 48
6.4 原理图 48
6.5 软件设置 49
6.5.1 需要的工具 49
6.5.2 用户模块列表 49
6.5.3 用户模块参数、全局资源 49
6.6 操作说明 49
6.7 执行代码示例 50

代码示例 7 使用 CY8C20xx6A CapSense 控制器调试 CSD 53

7.1 项目名称 53
7.2 概况 53
7.3 硬件设置 53
7.3.1 要求 53
7.3.2 摘要 53
7.3.3 组装电路板 54
7.4 原理图 54
7.5 软件设置 55
7.5.1 所需工具 55
7.5.2 用户模块列表和放置 55
7.5.3 用户模块参数、全局资源 55
7.6 操作说明 55
7.7 执行代码示例 56
7.8 通过 I^2C 读取 CapSense 数据 56
7.8.1 加载桥接控制面板工具 56
7.8.2 读取 BTN0 的原始计数、基准线和计数差值 57
7.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置 57
7.9 调校 CSD 用户模块 58

7.9.1 配置 CSD UM 设置 58

代码示例 8 通过 CY8C21x34/B CapSense 控制器并使用反馈电阻来调校 CSD 61

8.1 项目名称 61
8.2 概述 61
8.3 硬件设置 61
8.3.1 要求 61
8.3.2 摘要 61
8.3.3 组装电路板 62
8.4 原理图 63
8.5 软件设置 63
8.5.1 需要的工具 63
8.5.2 用户模块列表和放置 64
8.5.3 用户模块参数、全局资源 64
8.6 操作说明 64
8.7 执行代码示例 65
8.8 通过 I²C 读取 CapSense 数据 65
8.8.1 加载桥接控制面板工具 65
8.8.2 读取 BTN0 的原始计数、基准线和计数差值 66
8.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置： 66
8.9 调校 CSD 用户模块 67
8.9.1 配置 CSD UM 设置 67

代码示例 9 使用 CY8C20x34 CapSense 控制器调校 CSA_EMC 70

9.1 项目名称 70
9.2 概述 70
9.3 硬件设置 70
9.3.1 相关硬件 70
9.3.2 摘要 70
9.3.3 组装电路板 71
9.4 原理图 71
9.5 软件设置 72
9.5.1 需要的工具 72
9.5.2 用户模块列表 72
9.5.3 用户模块参数、全局资源 72
9.6 操作说明 72
9.7 执行代码示例 73
9.8 通过 I²C 读取 CapSense 数据 73
9.8.1 读取 BTN0 的原始计数、基准线和计数差值 74
9.8.2 读取 SLD0 的原始计数、基准线、计数差值和滑条位置 74
9.9 调校 CSA_EMC 用户模块 75
9.9.1 配置 CSD UM 设置 75

代码示例 10　通过使用 CY8C20xx6A 上的 SmartSense，可对一个 CapSense 按键进行功耗优化 78

10.1　项目名称 78
10.2　概况 78
10.3　硬件设置 78
10.3.1　相关硬件 78
10.3.2　摘要 78
10.4　组装电路板 79
10.5　原理图 79
10.6　软件设置 80
10.6.1　所需的工具 80
10.6.2　用户模块列表 80
10.6.3　用户模块参数、全局资源 80
10.7　操作说明 80
10.8　执行代码示例 81

附录 83

配置 CSD 用户模块 83
将代码示例移植到其他 CapSense 器件中 88
SmartSense 88
使用 CapSense 控制器进行通信 88
测量传感器的绝对电容 89
相关文档 89
设计指南 89
数据手册 89
应用笔记 89
套件指南 89
缩略语 90
文档修订记录 90

简介

赛普拉斯 CapSense®解决方案能够为您的设计提供精致、可靠且易用的电容式触摸感应功能。我们的电容式触摸感应解决方案已经取代了超过 40 亿个机械按键。同时，CapSense 还改变了移动电话、个人电脑、消费类电子产品和白色家电等产品的工业设计的外观。赛普拉斯强大的 CapSense 解决方案利用了灵活的可编程片上系统（PSoC®）架构，从而可加快产品上市时间、集成关键系统功能，并降低了材料清单（BOM）成本。

概述

本文档详细描述了 10 个 CapSense 代码示例，以帮助客户体验和评估赛普拉斯 CapSense 控制器、CapSense 套件和 CapSense 用户模块的各项功能。本文档还说明了如何通过 I^2C 和 UART 接口来回读和绘制传感器数据。阅读本文档之前，您应先熟悉 CapSense 感应技术。要想了解更多有关 CapSense 常规理论和操作的信息，请参考 CapSense 入门手册。

注意： 本应用笔记中的代码示例是本文档中所引用套件的特定代码示例。请参考 CapSense 入门，了解关键的设计注意事项以及布局的最佳实践，以确保能成功将这些代码示例移植到自定义的电路板上。如果本文档未满足您的要求或不能回答您的所有问题，或者您想要提供反馈信息，请联系 capsense@cypress.com。

为了方便您进行评估，我们按照复杂性递增的顺序介绍这些代码示例：

- **简介**：对于初次体验者来说，我们建议您先从 CapSense 触摸传感器的运行方式的演示着手。*代码示例 1：使用 CY8C20xx6A CapSense 控制器来实现基于 SmartSense 技术的按键和滑条*展现了如何通过使用 SmartSense 自动调校电容式感应算法轻松快速地实现触摸按键和滑条传感器。
- **通信**：掌握了 CapSense 传感器的运行方式后，您需要了解如何才能使其与 CapSense 控制器进行通信，以发送和接收数据。要想调校 CapSense 传感器，以及将 CapSense 控制器连接到主处理器上，必须先完成该步骤。*代码示例 2：通过 I2C 使 CY8C20xx6A CapSense 控制器与主机通信*和*代码示例 3：通过 UART 与 CY8C20xx6A CapSense 控制器进行数据传输*详细说明了如何与 CapSense 控制器进行通信。
- **调校**：下一步是调校 CapSense 传感器，以获得稳健的操作。要想完成该调试流程，您首先要知道 PCB 板上所有电容式传感器的寄生电容。*代码示例 4： 使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容*，*代码示例 5：使用 CY8C21x34/B CapSense 控制器测量传感器的绝对电容*，以及*代码示例 6：使用 CY8C20x34* CapSense 控制器测量传感器的绝对电容描述了如何测量不同 CapSense 器件中每个电容传感器的寄生电容。*代码示例 7：使用 CY8C20xx6A CapSense 控制器调试 CSD*，*代码示例 8：通过 CY8C21x34/B CapSense 控制器并使用反馈电阻来调校 CSD*，以及*代码示例 9：使用 CY8C20x34 CapSense 控制器调校 CSA_EMC* 详细说明了调校 CapSense 控制器的各个步骤。
- **优化功耗**：最后，*代码示例 10： 通过使用 CY8C20xx6A 上的 SmartSense，可对一个 CapSense 按键进行功耗优化*说明了如何优化 CapSense 控制器的功耗，以确保终端应用的电池的使用寿命更长。

注意： 本设计指南仅涉及四种 CapSense 器件：CY8C20xx6A、CY8C20x34、CY8C21x34/B 以及 CY8C24x94。请参考本文档附录，以了解如何将这些代码示例移植给其他 CapSense 器件。附录部分还提供了配置 CSD 用户模块的指导，您可以使用该 CSD 用户模块来创建或修改示例代码。

表 1. 代码示例列表

代码示例	代码示例	目标套件	说明
1	使用 CY8C20xx6A CapSense 控制器来实现基于 SmartSense 技术的按键和滑条	CY3280-20x66	演示了如何使用 SmartSense 自动调校的用户模块来实现 CapSense 按键和滑条传感器。
2	通过 I2C 使 CY8C20xx6A CapSense 控制器与主机通信	CY3280-20x66	说明了如何通过 I²C 将 CapSense 数据传入/传出 CapSense 控制器。此外，它还说明了如何使用 Bridge Control Panel 工具绘制回读数据。
3	通过 UART 与 CY8C20xx6A CapSense 控制器进行数据传输	CY3280-20x66	演示了如何用 RS232 数据包格式传输 CapSense 数据。此外，它还说明了如何使用 MultiChart 工具来绘制回读数据。
4	使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容	CY3280-20x66	寄生电容（C_P）高达 60 pF（准确度在 1 pF 以内）时，测量传感器绝对电容值， 然后使用 Windows HyperTerminal 或 Bridge Control Panel 工具将测量结果显示在个人电脑上。
5	使用 CY8C21x34/B CapSense 控制器测量传感器的绝对电容	CY3280-21x34	寄生电容（C_P）高达 60 pF（准确度在 1 pF 以内）时，测量传感器绝对电容值， 然后使用 Windows HyperTerminal 或 Bridge Control Panel 工具将测量结果显示在个人电脑上。
6	使用 CY8C20x34 CapSense 控制器测量传感器的绝对电容	CY3280-20x34	寄生电容（C_P）高达 60 pF（准确度在 1 pF 以内）时，测量传感器绝对电容值。
7	使用 CY8C20xx6A CapSense 控制器调试 CSD	CY3280-20x66	调校 CY8C20xx6A 中的 CSD 用户模块。
8	通过 CY8C21x34/B CapSense 控制器并使用反馈电阻来调校 CSD	CY3280-21x34	调校 CY8C21x34/B 中的 CSD 用户模块。
9	使用 CY8C20x34 CapSense 控制器调校 CSA_EMC	CY3280-20x34	调校 CY8C20x34 中的 CSA_EMC 用户模块。
10	通过使用 CY8C20xx6A 上的 SmartSense，可对一个 CapSense 按键进行功耗优化	CY3280-20x66	提供了使用 SmartSense 用户模块将传感器功耗降至 50 µA 的方法。

表 2. 按产品系列分类的代码示例

预期功能	产品系列			
	CY8C20xx6A	CY8C20x34	CY8C21x34/B	CY8C24x94
使用 SmartSense 实现按键和滑条传感器	代码示例 1	N/A [1]	移植后[2]	N/A[1]
通过 I²C 与主机进行通信	代码示例 2	对接后[2]	对接后[2]	对接后[2]
通过 UART 进行数据传输	代码示例 3	移植后[2]	移植后[2]	移植后[2]
测量传感器的绝对电容值	代码示例 4	代码示例 5	代码示例 6	移植后[2]
调校 CSD	代码示例 7	N/A [3]	不支持	N/A [3]
使用反馈电阻调校 CSD	N/A [3]	N/A [3]	代码示例 8	N/A [3]
调校 CSA_EMC	N/A [3]	代码示例 9	N/A [3]	N/A [3]
通过 SmartSense 实现功耗优化	代码示例 10	不支持	不支持	不支持

[1] 这些器件不支持 SmartSense 用户模块。
[2] 将本指南中所提供的项目移植到相应的器件内后，该代码示例才开始生效（如附录部分中“将代码示例移植到其他 CapSense 器件中”一节的内容所述）
[3] 这种调校方法仅适用于该器件和 UM。

代码示例 1　使用 CY8C20xx6A CapSense 控制器来实现基于 SmartSense 技术的按键和滑条

1.1　项目名称

CE_1_ButtonsandSliders_withSmartSense_withCY8C20xx6A

1.2　概况

为了确保电容式传感器能够正常运行，您需要根据寄生电容（C_P）的大小手动调试几个参数。SmartSense™用户模块改进并扩展了 CSD UM 的基本架构。其中，一个关键性的改进就是 SmartSense 自动调校功能。SmartSense UM 在运行期间能够根据每个传感器的 C_P 优化寄生电容的操作参数；在使用 SmartSense 时不需要进行手动调校。该代码示例将演示了使用 SmartSense 用户模块的方法。我们使用 EzI2C 用户模块来将 CapSense®参数（如各个按键的原始计数、计数差值、基准线，以及滑条的质心位置）发送给 I^2C 主设备。

1.3　硬件设置

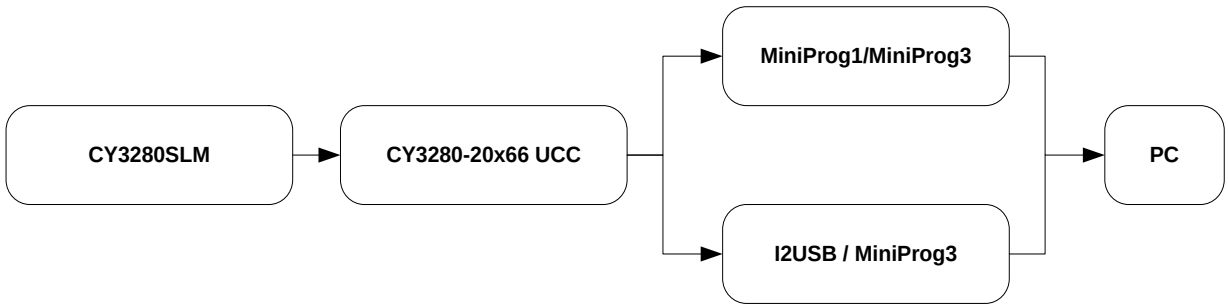
1.3.1　要求

- CY3280-20x66 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 装有 Windows XP 或更高版本的 PC

1.3.2　摘要

图 1-1 演示了硬件的设置方式。在该图中，通过一个 22x2_RA_Receptacle 将 CY3280-20x66UCC 套件和 CY3280-SLM 模块连接起来。然后，使用 MiniProg1/MiniProg3 或 I2USB 桥接器/MiniProg3 将它们连接到套件的 ISSP 插座上。该设置使用 MiniProg1/MiniProg3 进行编程，并使用 I2USB 桥接器/Miniprog3 将数据发送给 PC。最后，通过 USB 线缆将它们连接到 PC 上。

图 1-1. 硬件设置框图

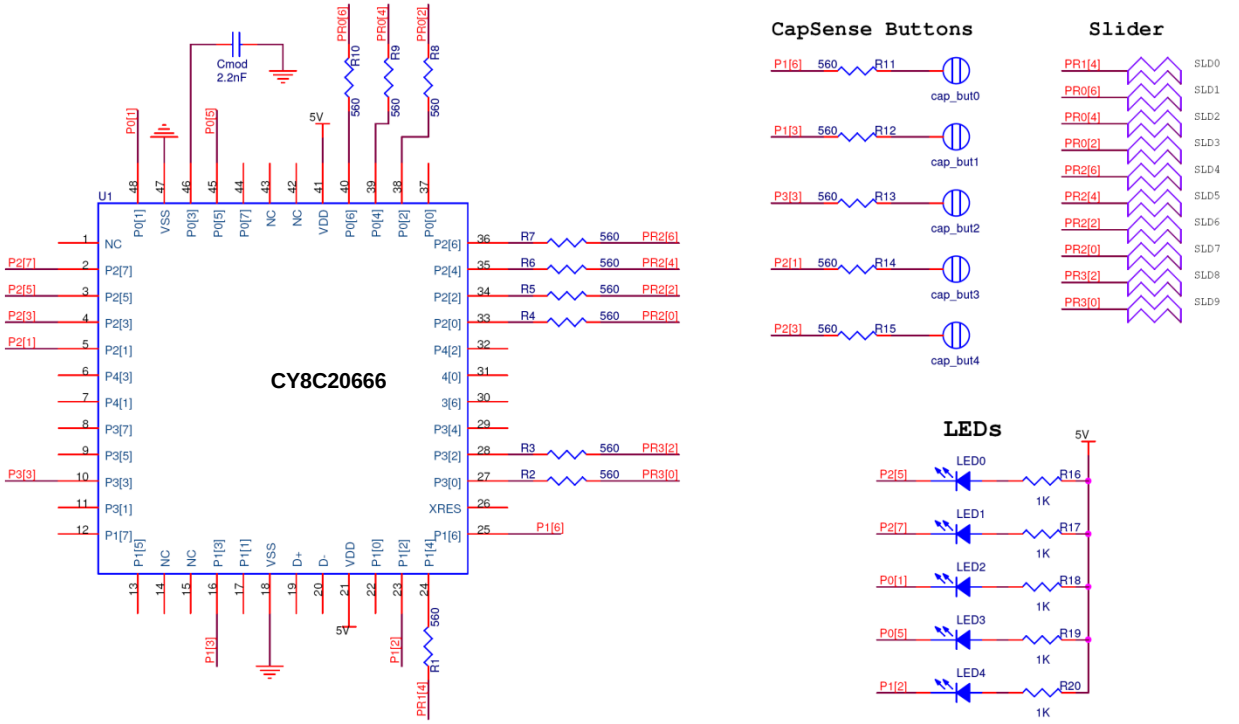


1.3.3 组装电路板

需要进行以下的硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 连接到 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）上。
- 在插座 J7 上放置一个跳线器，将 UCC 电路板的引脚 V_{CC} 和 V_{CC_PROG} 短接。该设置允许您通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上放置一个跳线器，将 CY3280-SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。通过该设置，可将 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 UCC 电路板的 ISSP 插座 J3 上。仅当使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。通过 Bridge Control Panel 软件读取 CapSense 数据时，需要使用 I2USB 桥接器/MiniProg3 代替 MiniProg1/MiniProg3 进行连接。
- 使用 USB A 转 Mini B 线缆将 MiniProg1（或 I2USB 桥接器/MiniProg3）的另一端连接至 PC 上。

1.4 原理图



将大小为 2.2 nF 的调制器电容（C_{MOD}）连接到 P0[3]引脚上。使用一个 560 Ω 的电阻将各个 CapSense 按键串联起来，以降低射频干扰。将 LED 配置为低电平有效，并将它与 1 kΩ 的电阻串联起来。该原理图中共有五个 LED、五个按键和一个拥有 10 个可用段的滑条。

表 1-1. LED、按键和滑条段的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

1.5 软件设置

1.5.1 所需工具

- PSoC Designer™（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel 工具

1.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块（UM）以及各个 UM 所占用的硬件资源。

用户模块	硬件资源
SmartSense	CapSense 和比较器、定时器 1
EzI2C	I²C/SPI 模块

1.5.3 用户模块参数、全局资源

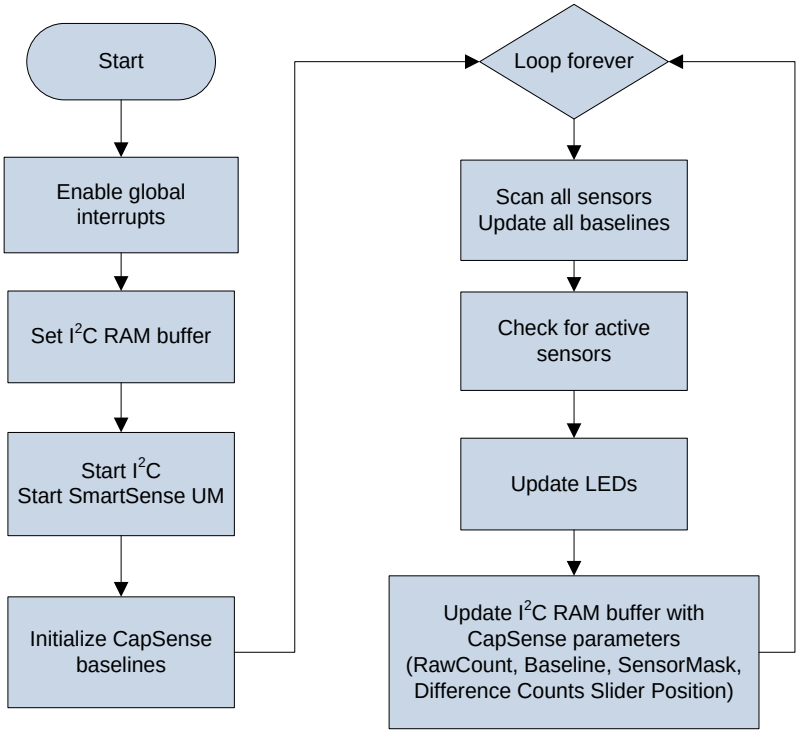
项目中的 ReadMe.txt 文件说明了该代码示例所用各用户模块的参数设置。此外，还提供了一个全局资源列表。

1.6 操作说明

复位时，固件将执行以下操作：

- 定义一个结构体 MyI2C_Regs 以存储按键编号、原始计数、计数差值、基准线、中心位置以及该按键编号所对应的 CapSense 按键的状态。
- 使能全局中断，然后启动 SmartSense 用户模块。
- 启动 EzI2C UM，然后将结构体 MyI2C_Regs 设置为 I²C RAM 缓冲区。
- 无限循环地执行下面操作：
  - 持续扫描所有传感器，并为 MyI2C_Regs 结构体更新原始计数、计数差值、基准线、滑条中心位置以及选定 CapSense 按键的状态。I²C 主设备通过将按键编号写入到 EzI2C 从设备 I²C 缓冲区的第一个字节中，依次请求某个特殊按键的 CapSense 数据。
  - 当固件检测到某个按键触摸时，它会将相应的 LED 状态切换为“ON”（打开）。释放按键时，LED 的状态将切换为“OFF”（关闭）。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”（打开），以指出触摸位置。

图 1-2. 代码示例 1 的功能流程



1.7 执行代码示例

使用项目来编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-20x66 套件指南中的第五章节。

1. 使用 MiniProg1/MiniProg3 或 CY3280-20x66 UCC 套件指南中所介绍的任意一个电源为该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。CY3280-SLM 电路板上相应的 LED 会点亮。
3. 触摸某个按键。CY3280-SLM 模块电路板上相应的 LED 会被点亮。您可以同时触摸多个按键，也可以同时使用线性滑条和按键。

1.8 通过 I²C 读取 CapSense 数据

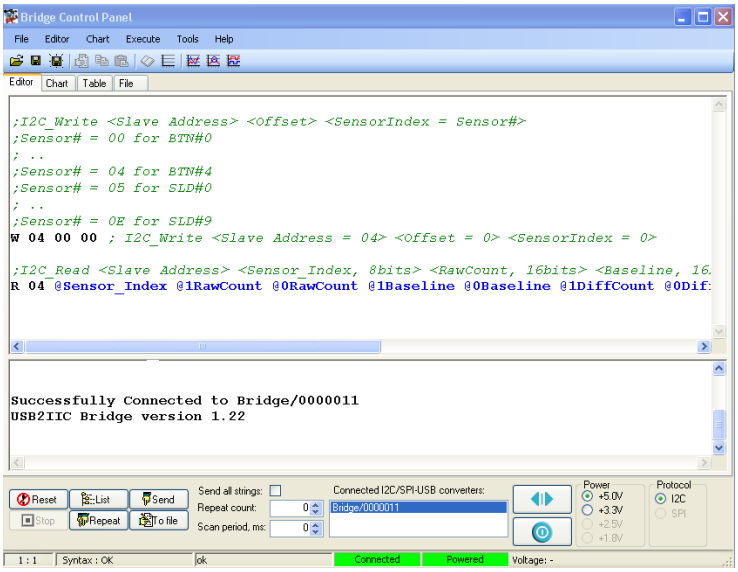
按照该程序执行代码示例，并读取 Bridge Control Panel 工具上的 CapSense 数据。有关 Bridge Control Panel 的更多详细信息，请参见 AN2397 — “CapSense®数据查看工具”。

1.8.1 加载 Bridge Control Panel 工具

1. 使用 I2USB 桥接器/MiniProg3 和 USB A 转 Mini B 线缆将您的计算机与 CY3280-20x66 通用 CapSense 控制器电路板的 ISSP 连接器 J3 连接在一起。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。

   Bridge Control Panel 是在 PSoC Programmer 安装过程中安装的一个组件。
3. 从端口选择窗口中选择您需要的器件。
4. 为 CY3280-20x66 CapSense 控制器电路板提供 5 V 的电源。
5. 从 Bridge Control Panel 窗口中依次选择 **File** > **Open**。加载代码示例项目文件夹 *BCP Configuration Files* 文件夹中加载 `BCP.iic` 文件。

6. 依次选择 **Charts > Variable Settings**。从代码示例项目文件夹中的 *BCP Configuration Files* 文件夹中加载 `BCP.ini` 文件。



7. 单击 **OK**，返回主窗口。

1.8.2 读取 BTN0 的原始计数、基准线和计数差值

1. 发送 I²C 写指令 W 04 00 00 一次。
2. 点击 **Repeat** 按键，以连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1Diffcount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，查看 BTN0 的原始计数、基准线和计数差值。

1.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置

1. 发送 I²C 写指令 W 04 00 05 一次。
2. 点击 **Repeat** 按键，以连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1Diffcount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，以查看 SLD0 的原始计数、基准线、计数差值和滑条位置。

代码示例 2　通过 I²C 使 CY8C20xx6A CapSense 控制器与主机通信

2.1　项目名称

CE_2_HostCommunication_viaI2C_withCY8C20xx6A

2.2　概述

该代码示例使用 I²C 通信协议将五个 CapSense®按键和一个十段式滑条的数据（原始计数、计数差值、基准线和状态）发送给 Bridge Control Panel 工具，便于查看。CSD 算法激活了 CY8C20xx6A CapSense 控制器上的这些按键。通过指定按键编号，可以获得特定按键的具体数据。

2.3　硬件设置

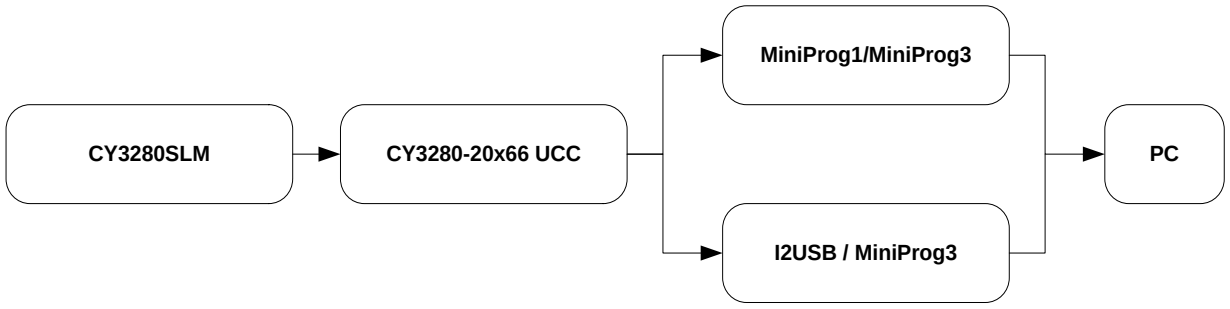
2.3.1　要求

- CY3280-20x66 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆

2.3.2　摘要

图 2-1 演示了硬件的设置方式。使用一个 22x2_RA_Receptacle 将 CY3280-20x66 UCC 套件和 CY3280-SLM 模块连接起来。然后，将 MiniProg1/MiniProg3 或 I2USB bridge/MiniProg3 连接至套件的 ISSP 插座上。该设置使用 MiniProg1/MiniProg3 进行编程，并使用 I2USB 桥接器/Miniprog3 将数据发送给 PC。最后，使用 USB 线缆将它们连接到 PC 上。

图 2-1. 硬件设置框图

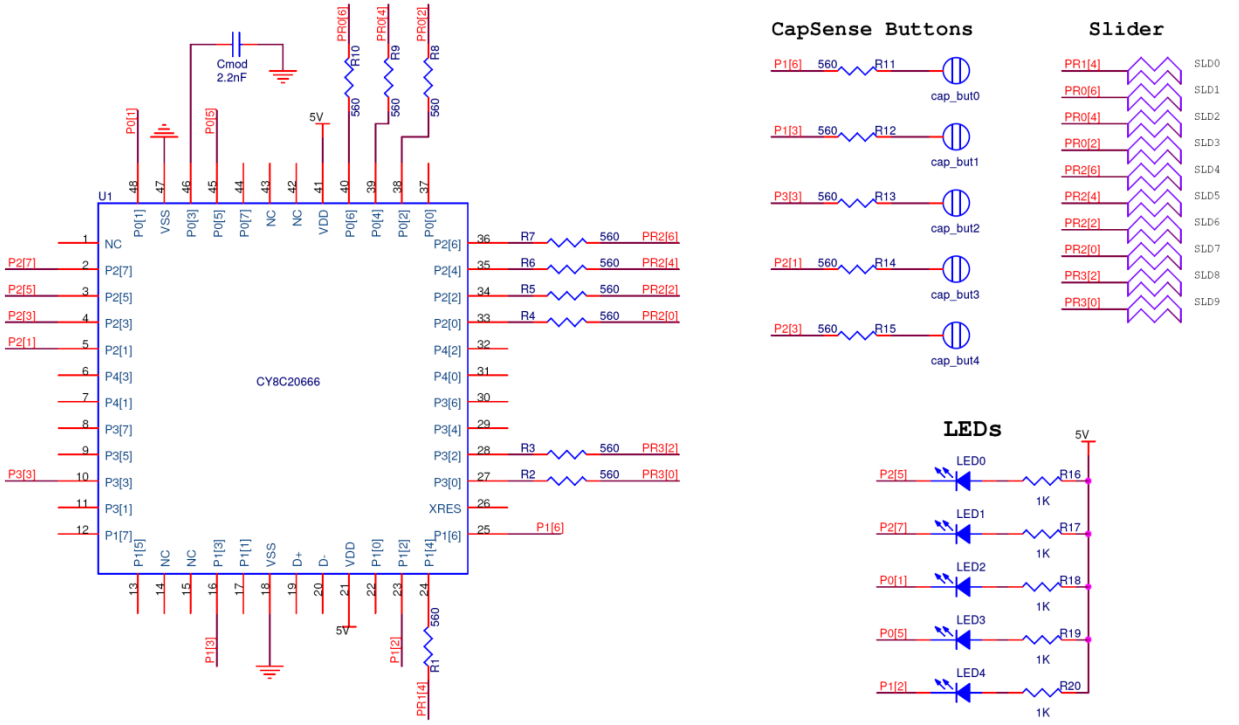


2.3.3 设置电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡上的插座 J1 和 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）连接在一起。
- 在插座 J7 上放置一个跳线器，用于将 UCC 电路板的引脚 V_{CC} 和 V_{CC_PROG} 短接。借助于该设置，您可以通过 ISSP 连接器为 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3 上。
- 在插座 J2 上放置一个跳线器，将 CY3280-SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。通过该设置，可将 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 UCC 电路板的 ISSP 插座 J3 上。只有使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。使用 Bridge Control Panel 软件读取 CapSense 数据时，需要使用 I2USB 桥接器/MiniProg3 代替 MiniProg1/MiniProg3 进行连接。
- 使用 USB A 转 Mini B 线缆将 MiniProg1（或 I2USB 桥接器/MiniProg3）的另一端连接到 PC 上。

2.4 原理图



将大小为 2.2 nF 的调制器电容（C_{MOD}）连接到 P0[3]引脚上。使用一个 560 Ω 的电阻将各个 CapSense 按键串联起来，以减少射频干扰。将 LED 配置为低电平有效，并将其串联一个 1 kΩ 的电阻。该原理图中共有五个 LED、五个按键以及一个包含 10 个可用段的滑条。

表 2-1. LED、按键和滑条段的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

2.5 软件设置

2.5.1 所需工具

- PSoC Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel 工具

2.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块（UM）和各个 UM 所占用的硬件资源。

用户模块	硬件资源
CSD	CapSense 模块、定时器 1（默认）
EzI2C	I²C/SPI 模块

2.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例所用各用户模块的参数设置。此外，它还提供了一个全局资源列表。

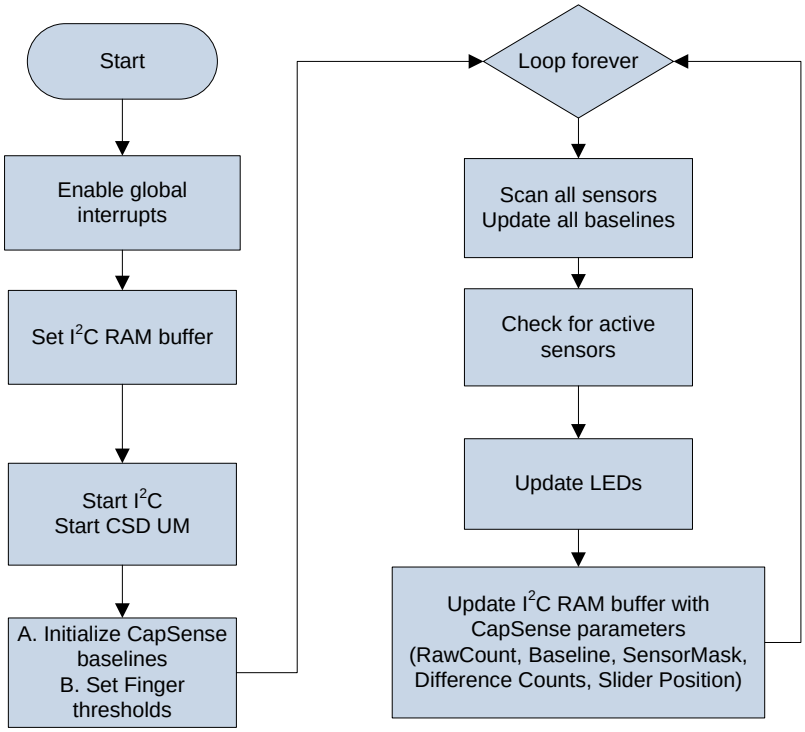
注意： 我们对该代码示例进行了调校，使之符合 CY3280-SLM 套件中 1.5 mm 厚的亚克力覆盖层。要想使用覆盖层更厚的代码示例，请按照*代码示例 7：使用 CY8C20xx6A CapSense 控制器调试 CSD* 所述的内容调校项目。

2.6 操作说明

复位时，固件将执行以下操作：

- 定义一个结构体 MyI2C_Regs 以存储按键编号、原始计数、计数差值、基准线、中心位置以及同该按键编号相应的 CapSense 按键的状态。
- 使能全局中断，然后启动 CSDUM。
- 启动 EzI2C UM，然后将 MyI2C_Regs 结构体设置为 I²C RAM 缓冲区。
- 无限循环地执行下面操作：
  - 继续扫描所有传感器，并为 MyI2C_Regs 结构体更新原始计数、计数差值、基准线、滑条中心位置以及选定 CapSense 按键的状态。通过将按键编号写入到 EzI2C 从设备中 I²C 缓冲区的第一个字节内，I²C 可以请求某个特定按键的 CapSense 数据。
  - 当固件检测到按键触摸时，它会将相应的 LED 状态切换为“ON”（打开）。释放按键时，LED 状态将恢复为“OFF”（关闭）。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”（打开），以表示触摸位置。

图 2-2. 代码示例 2 的功能流程



2.7 执行代码示例

使用项目编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-20x66 套件指南的第五章节。

1. 使用 MiniProg1/MiniProg3 或 CY3280-20x66 UCC 套件指南中所介绍的任意一个电源为该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。
   CY3280-SLM 电路板上相应的 LED 被点亮。
3. 触摸某个按键。
   CY3280-SLM 模块电路板上相应的 LED 被点亮。您可以同时触摸多个按键，也可以同时使用线性滑条和按键。

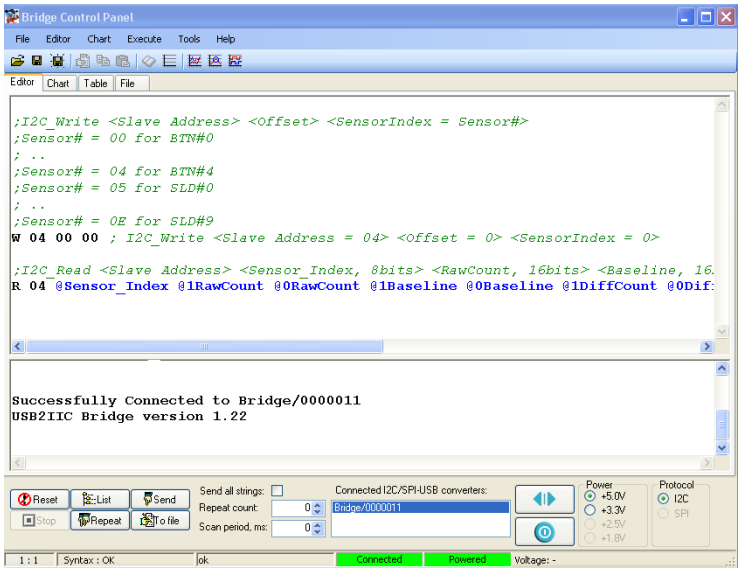
2.8 通过 I²C 读取 CapSense 数据

按照该流程执行代码示例并读取 Bridge Control Panel 工具上的 CapSense 数据。更多有关桥接控制面板工具的详细信息，请参考 AN2397 — “CapSense®数据查看工具”。

2.8.1 加载 Bridge Control Panel 工具

1. 使用 I2USB 桥接器/MiniProg3 和 USB A 转 Mini B 线缆，将您的计算机和 CY3280-20x66 通用 CapSense 控制器电路板的 ISSP 连接器 J3 连接起来。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。
   Bridge Control Panel 是在 PSoC Programmer 安装过程中安装的一个组件。
3. 从端口选择窗口中选择您需要的器件。
4. 为 CY3280-20x66 CapSense 控制器电路板提供 5 V 的电源。
5. 从 Bridge Control Panel 窗口中依次选择 **File** >**Open**。加载代码示例项目文件夹中 *BCP Configuration Files* 文件夹中的 `BCP.iic` 文件。

6. 依次选择 **Charts** > **Variable Settings**。从代码示例项目文件夹中的 *BCP Configuration Files* 文件夹内加载 `BCP.ini` 文件。



7. 单击 **OK**，返回到主窗口。

2.8.2 读取 BTN0 的原始计数、基准线和计数差值

1. 发送 I²C 写指令 W 04 00 00 一次。
2. 点击 **Repeat** 按键，以连续发送下面的 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，查看 BTN0 的原始计数、基准线和计数差值

2.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置

1. 发送 I²C 写指令 W 04 00 05 一次。
2. 点击 **Repeat** 按键，以连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 单击 **Chart** 选项卡，以查看 SLD0 的原始计数、基准线、计数差值和滑条位置。

代码示例 3　通过 UART 与 CY8C20xx6A CapSense 控制器进行数据传输

3.1　项目名称

CE_3_DataTransmission_viaUART_withCY8C20xx6A

3.2　概述

该代码示例使用 UART 通信协议以 MultiChart 工具可读取的格式将 CapSense®数据发送给 PC。我们使用 MultiChart 工具以图形的形式查看下面各 CapSense 参数：原始计数、计数差值（信号）、基准线和滑条位置。SmartSense™用户模块（UM）在 CY3280-SLM 上持续扫描所有按键和滑条段，而 UART UM 会将 CapSense 参数发送给 TX8 输出引脚。外部电平转换器将 TX8 数据转换为 RS232 电压电平，然后通过标准的 RS232 线缆将其发送给 PC。在 PC 上，我们使用由 AN2397 提供的 MultiChart 工具监控数据。代码示例还会打开并关闭 CY3280-SLM 上的 LED，用于表明按键状态和滑条位置。

3.3　硬件设置

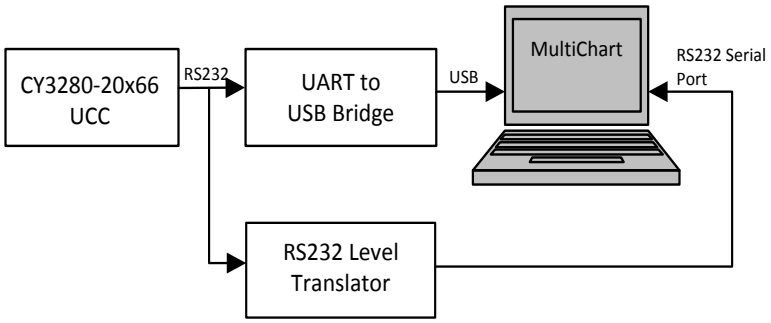
3.3.1　要求

- CY3280-20x66 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 外部 RS232 电平转换器
- 标准的 9 引脚 RS232 串行线缆
- 带有串行（9 引脚 RS232）端口的 PC

3.3.2　摘要

图 3-1 演示了硬件的设置方式。在该图中，使用一个 22x2_RA_Receptacle 将 CY3280-20x66 UCC 套件和 CY3280-SLM 模块连接起来。将 MiniProg1/MiniProg3 连接到插座 J3 上，从而为该套件供电并对其进行编程。然后，将连接器 P3 上的 P3[5]连接至外部电平转换器，该转换器通过 UART 线缆连接至 PC。

图 3-1. 硬件设置框图



注意： CY3280-20x66 UCC 可以通过以下两种方法将数据发送给 PC，即：使用 UART 到 USB 桥接器或者 RS232 电平转换器（需要一个带有 RS232 串行端口的 PC）。

3.3.3 组装电路板

需要进行以下各硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 和 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）连接在一起。
- 在插座 J7 上放置一个跳线器，从而将 UCC 电路板的引脚 V_{CC} 和 V_{CC_PROG} 短接。完成该设置，您便可以通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上放置一个跳线器，将 CY3280-SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。通过该设置，可将 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接至 CY3280-20x66 UCC 的插座 J3 上。
- 将 CY3280-20x66 UCC 连接器 P3 的 P3[5]同外部电平转换器连接起来。有关原理图示例，请参见图 3-2。
- 将 CY3280-20x66 UCC 上连接器 P3 的 GND 同外部电平转换器连接起来。
- 为外部电平转换器提供电源。可以使用来自 CY3280-20x66 UCC 上连接器 P3 的 V_{CC} 为外部电平转换器提供 5 V 的电源。
- 将外部电平转换器输出的 RS232 线缆连接至 PC 上。

3.4 原理图

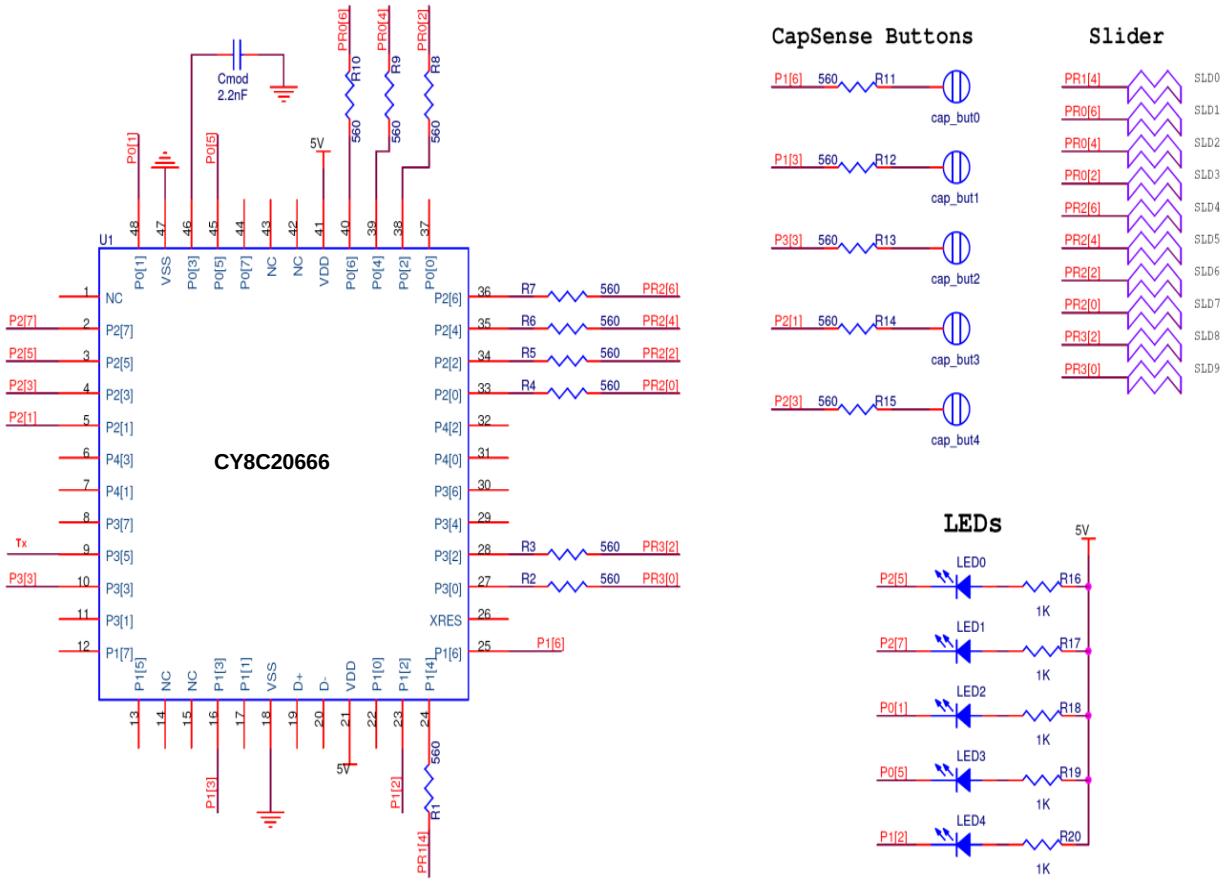
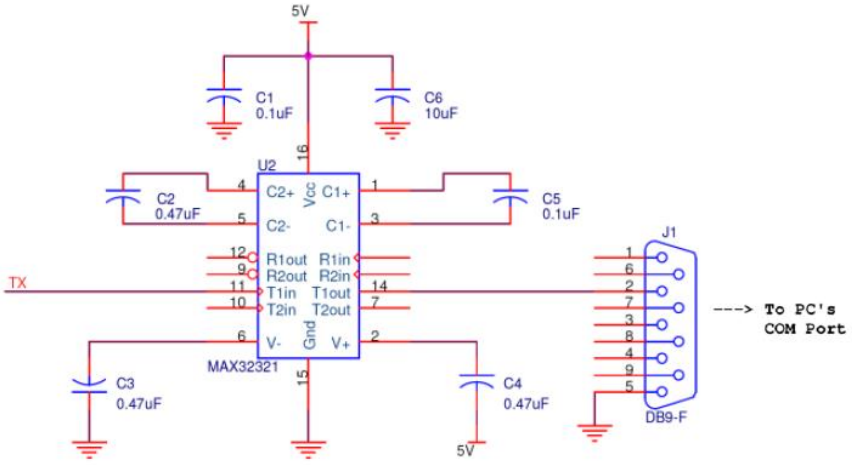


图 3-2. 外部电平转换器



将大小为 2.2 nF 的调制器电容（C_{MOD}）连接到 P0[3]引脚上。使用一个 560 Ω 的电阻串联各个 CapSense 按键，以降低射频干扰。将 LED 配置为低电平有效，并将其串连一个 1 kΩ 的电阻。该原理图中共有五个 LED、五个按键以及一个包含 10 个可用段的滑条。

表 3-1. LED、按键和滑条段的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

3.5 软件设置

3.5.1 所需的工具

- PSoC Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- MultiChart 工具（由 AN2397 提供）

3.5.2 用户模块列表

下表列出了该项目中所使用的用户模块和各个用户模块所占用的硬件资源。

用户模块	硬件资源
SmartSense	CapSense 模块、定时器 1（默认）
UART	未占用模块（软件实现）

3.5.3 用户模块参数，全局资源

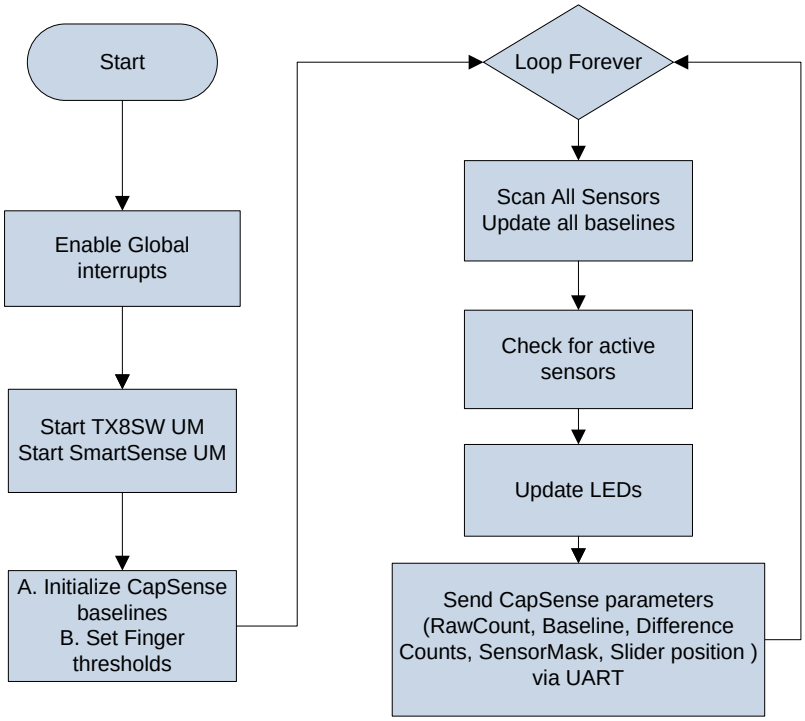
项目中的 ReadMe.txt 文件说明了该代码示例中所使用的用户模块的参数设置。此外，还提供了一个全局资源列表。

3.6 操作说明

复位时，程序会将器件配置中所有硬件设置加载到器件中，并执行 *main.c*。然后，固件将执行以下操作：

- 使能全局中断，然后启动 SmartSense 和 UART 用户模块。
- 初始化传感器基准线。
- 无限循环地执行下面操作：
  - 持续扫描所有传感器，并使用 UART 用户模块将原始计数、差值计数、基准线和滑条位置发送到 TX8 引脚。
  - 当固件检测到按键触摸时，它会将相应的 LED 状态切换为“ON”（打开）。释放按键时，LED 状态会恢复为“OFF”（关闭）。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”（打开），以表示触摸位置。

图 3-3. 代码示例 3 的功能流程



3.7 执行代码示例

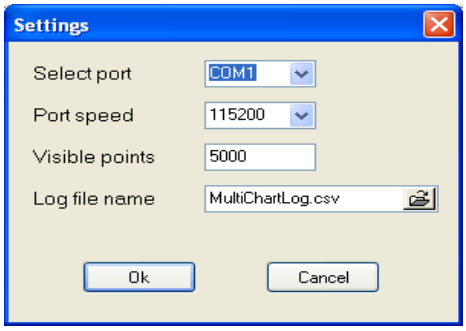
使用项目编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-20x66 套件指南中的第五章。

1. 使用 MiniProg1/MiniProg3 或 CY3280-20x66 UCC 套件指南中介绍的任意源给该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。
   CY3280-SLM 电路板上相应的 LED 将被点亮。
3. 触摸某个按键。
   CY3280-SLM 模块电路板上相应的 LED 将被点亮。您可以同时触摸多个按键，也可以同时使用线性滑条和按键。

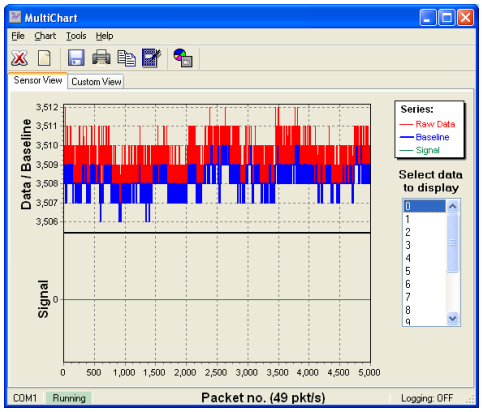
3.8 使用 MultiChart 将 CapSense 数据绘制成图表

按照该程序执行代码示例，并使用 MultiChart 工具查看绘制好的数据。要想获取更多信息和下载 MultiChart 工具，请参考 AN2397 — “CapSense®数据查看工具”

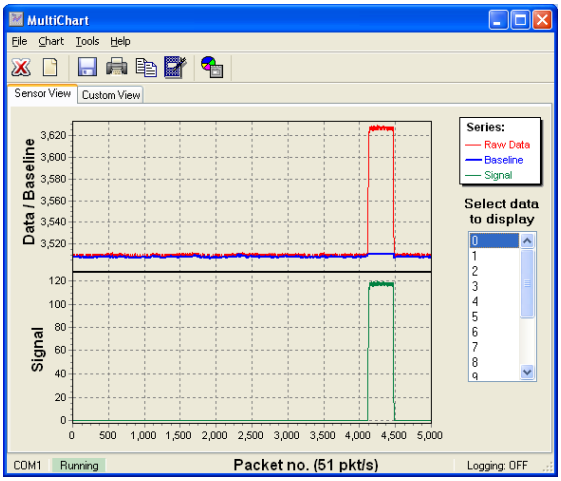
1. 打开 MultiChart 工具。
2. 在 MultiChart 工具的窗口中，依次选择 **Tools** > **Settings**。
   将出现“Settings”对话框。



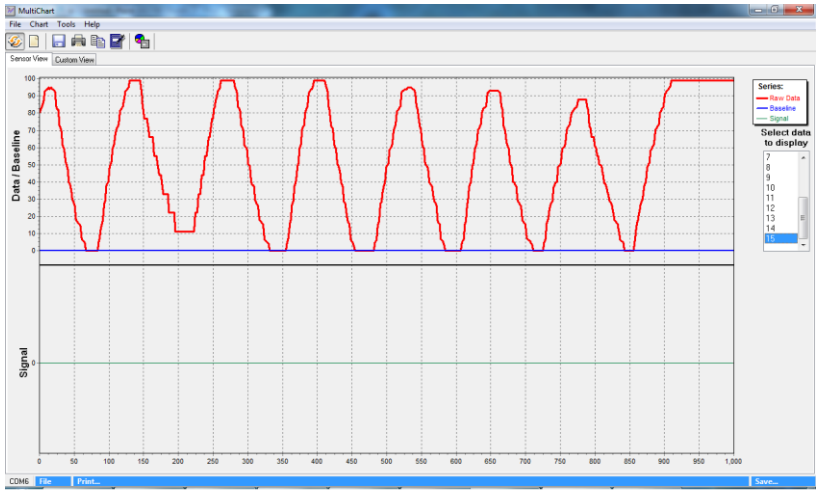
3. 使用 **Select port** 的下拉列表选择可连接到电路板和计算机的 COM 端口。
   大部分用户使用 COM1 端口进行连接。但是，您的系统可能不一样。
4. 使用 **Port speed** 的下拉表选择“115200”，然后点击 **OK**。
   原始计数和基准线分别显示为红色和蓝色。



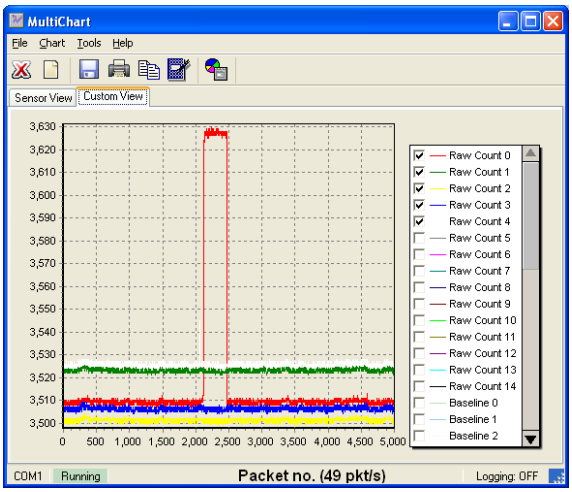
5. 触摸按键“0”以观察相应的信号线（计数差值）。



6. 系列 15 的‘Raw Count 15’（计数差值 15）变量表示滑条位置（其范围为 0 到 100），如下图所示。如果您看不到滑条的位置，请使用放大功能，或依次通过选择 Chart > Edit...>Axis，从而将各轴设为“Auto”（自动）。



7. 传感器视图一次只能显示一个传感器。在该视图中，‘Raw Count 15’（计数差值 15）表示滑条的位置。要想同时观察多个传感器，请点击“Custom View”（自定义视图）选项卡。



代码示例 4　使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容

4.1　项目名称

CE_4_MeasuringAbsSensorCap_withCY8C20xx6A

4.2　概况

该代码示例演示了如何使用 CY8C20xx6A CapSense®控制器来计算五个传感器的绝对电容，并通过使用 Windows HyperTerminal 软件或 Bridge Control Panel 工具在计算机上显示相应的结果。测量到的绝对电容值的精度为 1 pF。

4.3　硬件设置

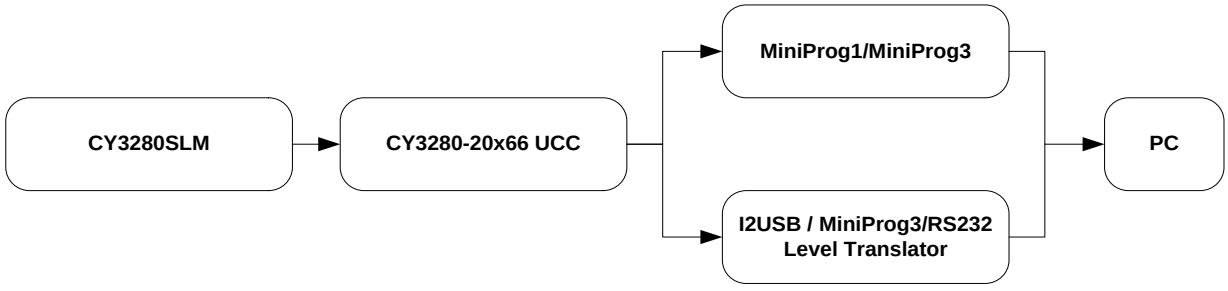
4.3.1　要求

- CY3280-20x66 通用 CapSense 控制器
- CY3280-SLM 通用 CapSense 线性滑条模块
- RS232 电平转换器模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 运行装有 Windows XP 或更高版本的 PC

4.3.2　摘要

下图显示了硬件的设置方式。使用一个 22x2_RA_Receptacle 将 CY3280-20x66 UCC 套件和 CY3280-SLM 模块连接起来。将 MiniProg1/MiniProg3 连接到插座 J3 上，从而为该套件供电并对其进行编程。

图 4-1. 硬件设置框图



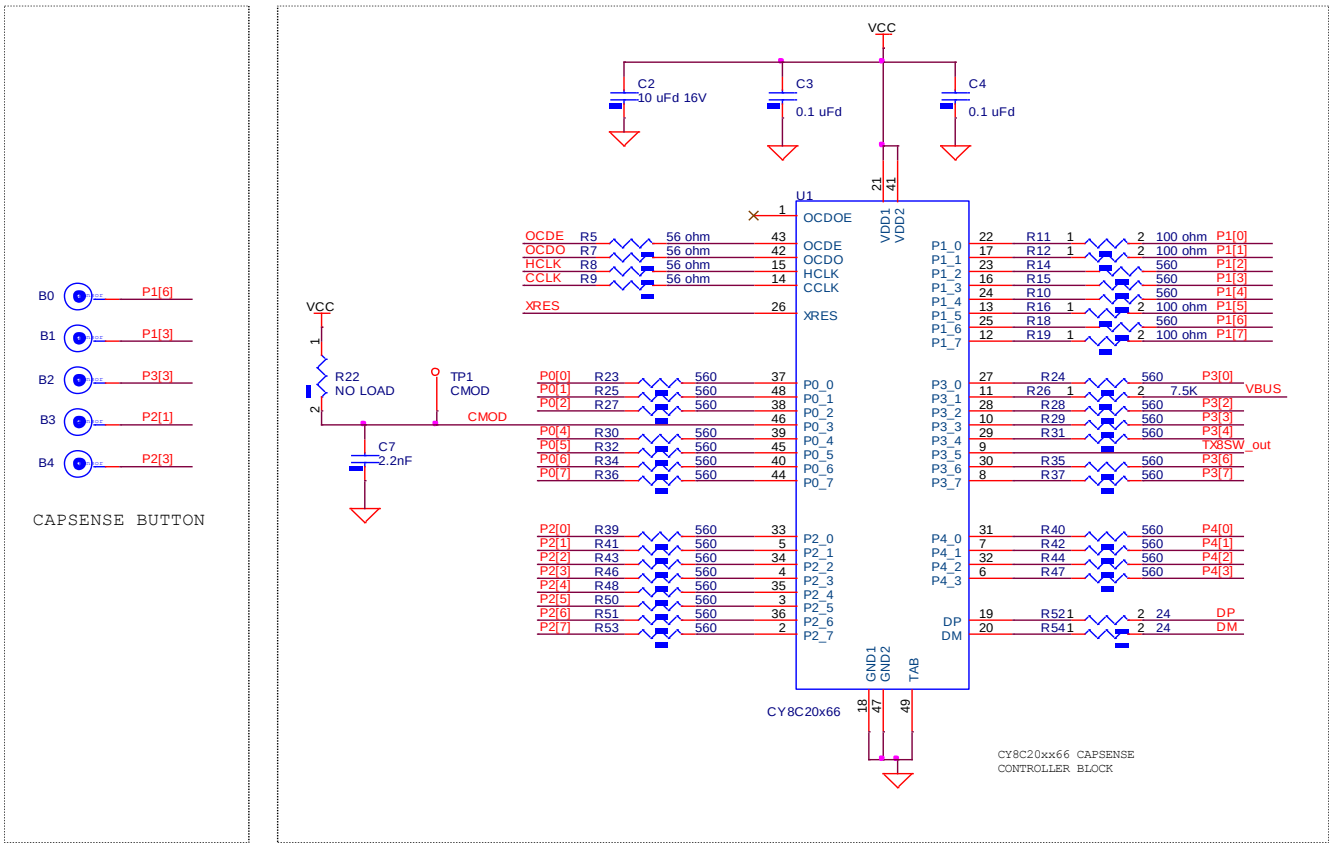
注意： 该代码示例使用的是外部电平转换器。但是，您也可以使用 UART 转 USB 的桥接器查看 PC 上的数据。您可以将 UCC 套件中的 CY3240-I2USB 桥接器转换为 UART 转 USB 的桥接器。更多有关信息，请参考 AN2397 — “CapSense®数据查看工具”。使用 Bridge Control Panel 软件并通过 I2C-USB 桥接器或 MiniProg3，可以在 PC 上查看传感器 Cp。通过 HyperTerminal（超级终端）并使用 RS232 通信协议，可以在 PC 上查看数据。

4.3.3 组装电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 和 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）连接在一起。
- 在插座 J7 上放置一个跳线器，将 UCC 电路板上的引脚 V_{CC} 和 V_{CC_PROG} 短接。该设置便于您通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置将 CapSense 控制器的 XRES 引脚连接到 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上连接一个跳线器，将 CY3280SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。该设置会使 CY3280SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 CY3280-20x66 UCC 的 J3 插座上，以编程器件。
- 如果通过 I2C 通信协议在桥接控制面板上查看传感器 Cp，请将 CY3280-20x66A 的 J3 插座连接至 I2C-USB 桥接器/MiniProg3，并使用 USB A 转 mini B 线缆将该桥接器/MiniProg3 的另一端连接到计算机上。通过点击 Bridge control panel 上的“Toggle power”按键，可以给器件供电。
- 如果通过 UART 在 HyperTerminal 上查看传感器 Cp，请将 RS232 电平转换器电路板上的 TX 引脚连接到 UCC 的 P1[7]。将电平转换器的 Vcc 和 GND 引脚分别连接至 UCC 电路板上的 Vcc 和 GND 引脚。RS232 电平转换器的一个普遍示例是 MAX 232 串行电平转换器。
- 使用串行线缆将电平转换器的 RS232 端口连接到计算机上。

4.4 原理图



调制器电容（C_{MOD}）指的是 P0[3]引脚上连接的 2.2 nF 大小的电容。在各个 CapSense 按键上串联一个 560 Ω 的电阻，以减少射频干扰。

表 4-1. CapSense 按键的引脚分配

按键	引脚
0	P1[6]
1	P1[3]
2	P3[3]
3	P2[1]
4	P2[3]

4.5 软件设置

4.5.1 所需工具

- PSoC® Designer™（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel 工具
- Windows HyperTerminal 软件

4.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块（UM）和各个 UM 所占用的硬件资源。

用户模块	硬件资源
CSD	CapSense 模块，定时器 1（默认）
EzI2C	I2C 模块
TX8SW	未占用模块（软件实现）

4.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例中所使用的各用户模块的参数设置。此外，它还提供了一个全局资源的列表。

4.6 操作说明

该程序采用了以下公式来计算传感器的原始计数：

$$raw_count = \left(\frac{(2^n - 1) \cdot V_{ref} \cdot \overline{f_s}}{i_{DAC}} \right) \cdot C_s$$

其中：
n = 分辨率
Vref = 参考电压
fs = 预充电时钟的开关电压
Cs = 传感器电容
iDAC = iDAC 电流
根据 CSD UM 参数，可确定 n、f 和 Vref 的值。

根据下面公式，可计算出 iDAC 的值：

$$i_{DAC} = IDAC_D \cdot iDAC_{gain} \cdot iDAC_{range}$$

IDAC_D 指的是控制寄存器的设置，且 iDACrange 表示 CSD UM 参数设置中的“Idac Range”。在设计时这两个参数已知。计算 IDAC 的公式中唯一一个未知的参数是 iDACgain。iDACgain 指的是 iDAC 的强度（单位为 A/bit），该参数会根据各种器件中 PVT 值的不同而变化。在 iDAC = 4X 的范围内，iDACgain 的测试结果要满足：230nA/bit < iDACgain < 270nA/bit。

Krypton ATE 程序（生产时预编程）用于测量 4x 范围内的 iDAC 电流（IDAC_D = 0xC0 = 192）。如果 iDAC = 247nA/bit，则该电流将为 247nA/bit；192 * 4 = 190 uA。ATE 测量实际的电流值，并使用符号和大小的格式将该值与 190 uA 的差值记录下来， 然后存储为 bIDACComp。以符号格式演示 MSB 位（mask = 0x80），其中：1 表示负极，0 表示正极。例如，对于包含已存储的 0x8D 值的器件，其 iDACgain = (190uA – 13uA) / (4 * 192) = 230nA/bit。

注意：

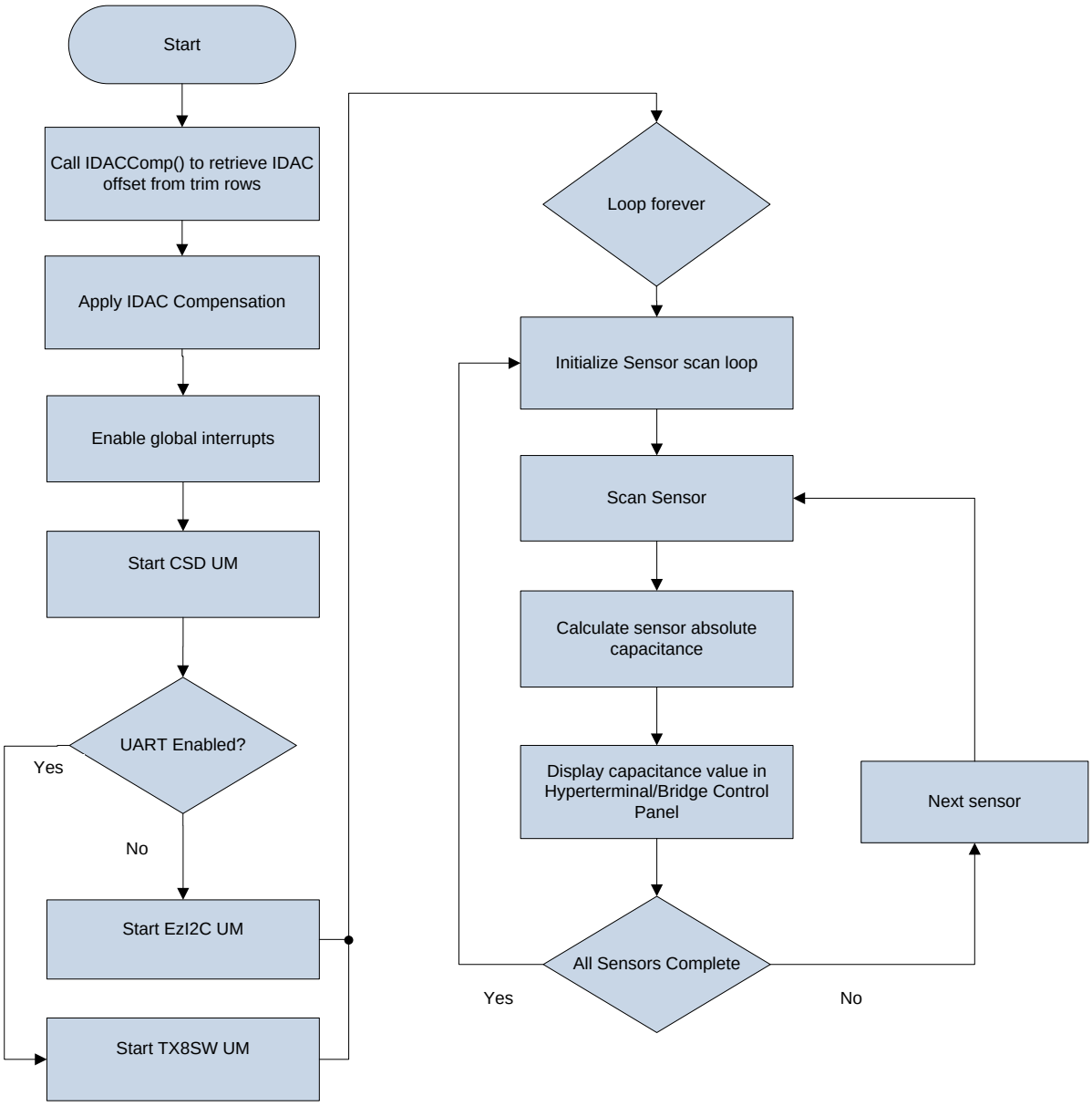
$$\text{uSensitivity} = \text{Cs/RawCount} = \frac{(2^n - 1) * V_{ref} * f_s}{i_{DAC}}$$

该程序按照下面流程将绝对电容显示在屏幕上：

1. 计算传感器的灵敏度，并将获得的结果存储在变量 uSensitivity 中。GetIDACComp()是 *GetIDACComp.asm* 文件中定义的自定义函数，用于检索 IDAC 中各部分差异的 IDAC 补偿。
2. 注意：由于 IDAC 会随着温度的变化而变化，因此，该代码示例仅在室温条件下有效。

3. 使能全局中断，并启动用户模块。
4. 按顺序扫描每个传感器，并从 CSD_waSnsResult[]数组中获取原始计数。
5. 通过使用原始计数与灵敏度的比例来计算电容的绝对值。
6. 将结果发送给计算机，并根据用户选择，将它们显示在 Bridge Control Panel 或 HyperTerminal 中。

图 4-2. 代码示例 4 的功能流程



4.7 执行代码示例

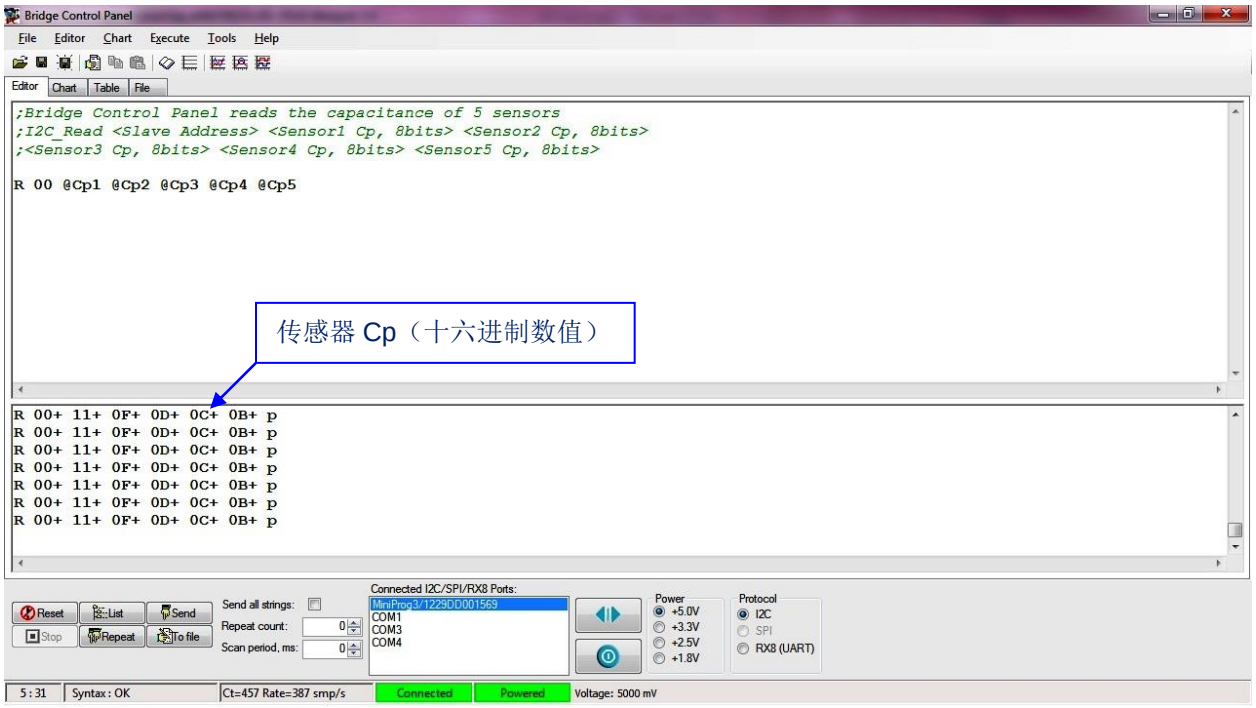
编程UCC电路板

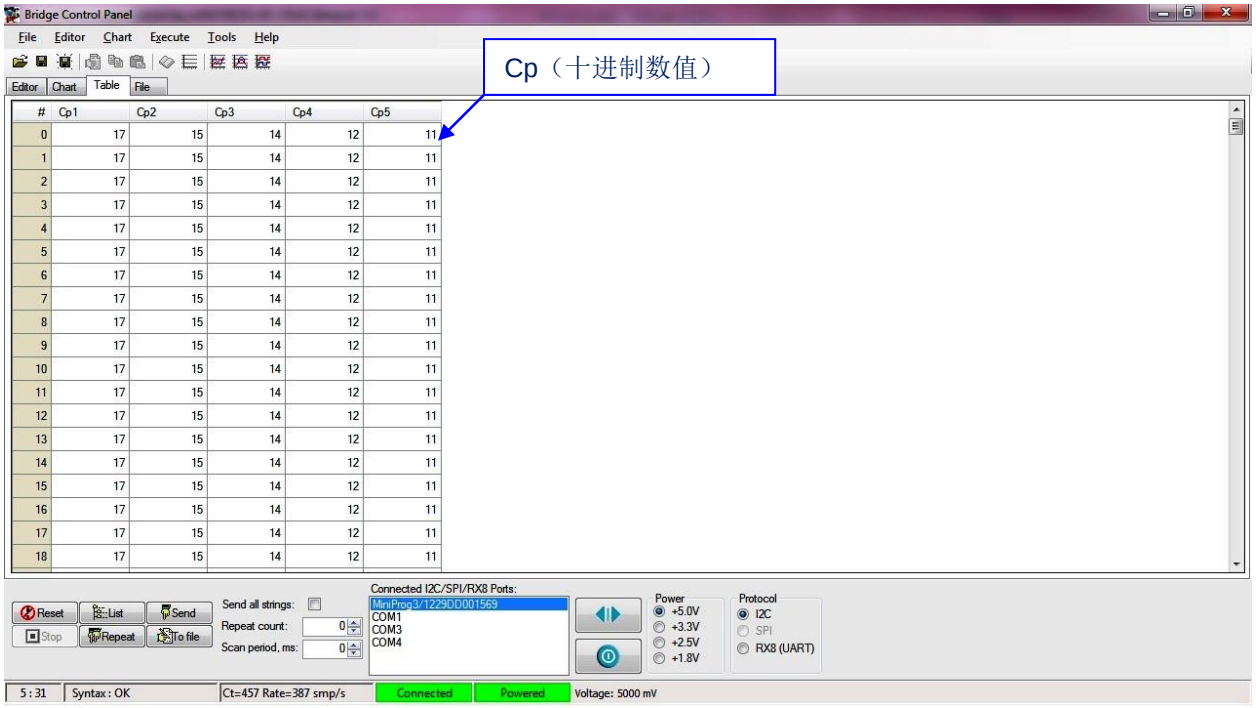
请确保在 *main.c* 文件中注释了“#define UART”语句，如下面屏幕截图所示。然后，生成项目（按下 F6 键），并对电路板进行编程。有关如何编程 UCC 电路板的详细信息，请参考 CY3280-20x66 套件指南中的第五章节。

```
74
75 /* Comment this while using I2C;Uncomment this while using UART */
76  //#define UART
77
```

使用 BCP 监控传感器数据

1. 将 CY3280-20x66A 的 J3 插座连接到 I2C-USB 桥接器或 MiniProg3 上，并使用 USB A 转 Mini B 线缆将 I2C-USB 桥接器/MiniProg3 的另一端连接到计算机上。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel（版本）** > **Bridge Control Panel（版本）**。
3. 从 Port selection 窗口中选择所需器件（I2C-USB 桥接器或 MiniProg3）。
4. 通过点击 Bridge Control Panel 上的“Toggle power”按键，为器件供电。
5. 从 Bridge Control Panel 窗口中依次选择 File > Open。从项目文件夹中的 BCP Configuration Files 文件夹内加载 *BCP.iic* 文件。
6. 依次选择 **Charts** > **Variable Settings**。从项目文件夹中的 BCP Configuration Files 文件夹内加载 *BCP.ini* 文件。
7. 点击“send”按键，将 I²C 指令发送给 CY8C20xx6 控制器，从而能在底部窗口中获取传感器的 Cp 值（十六进制），如下图所示。





编程UCC电路板

取消“main.c”中的注释语句“#define UART”（如下面截图所示），然后生成项目并编程 UCC 电路板。

```
74
75 /* Comment this while using I2C;Uncomment this while using UART */
76  #define UART
77
```

使用 HyperTerminal 监控传感器数据

1. 使用 Miniprog1 或 Miniprog3 给电路板供电。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Accessories > Communication > HyperTerminal**。
3. 为该连接键入名称。
4. 点击 **OK**。



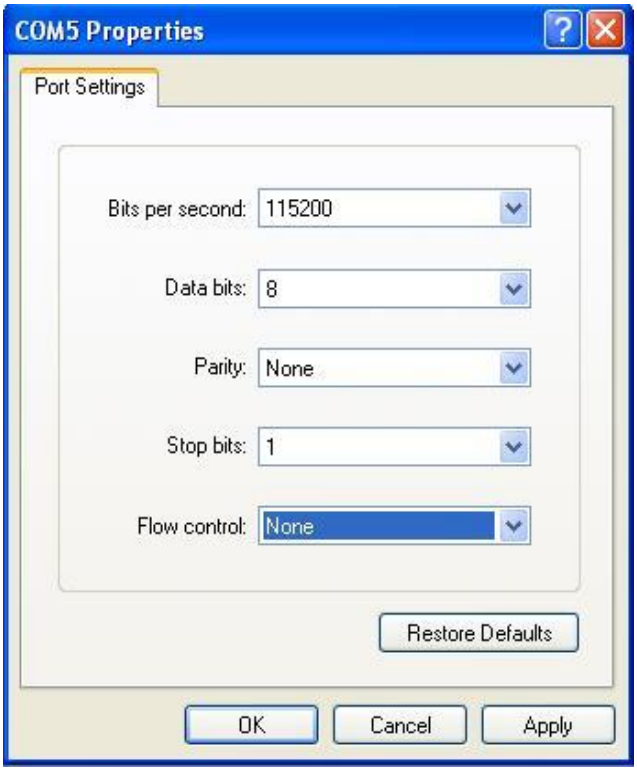
5. 通过“**Connect using**”框中的下拉列表，选择用于接收 PSoC 数据的串行端口。
6. 点击 **OK**。



7. 在 **COM Properties** 对话框中，配置以下参数：

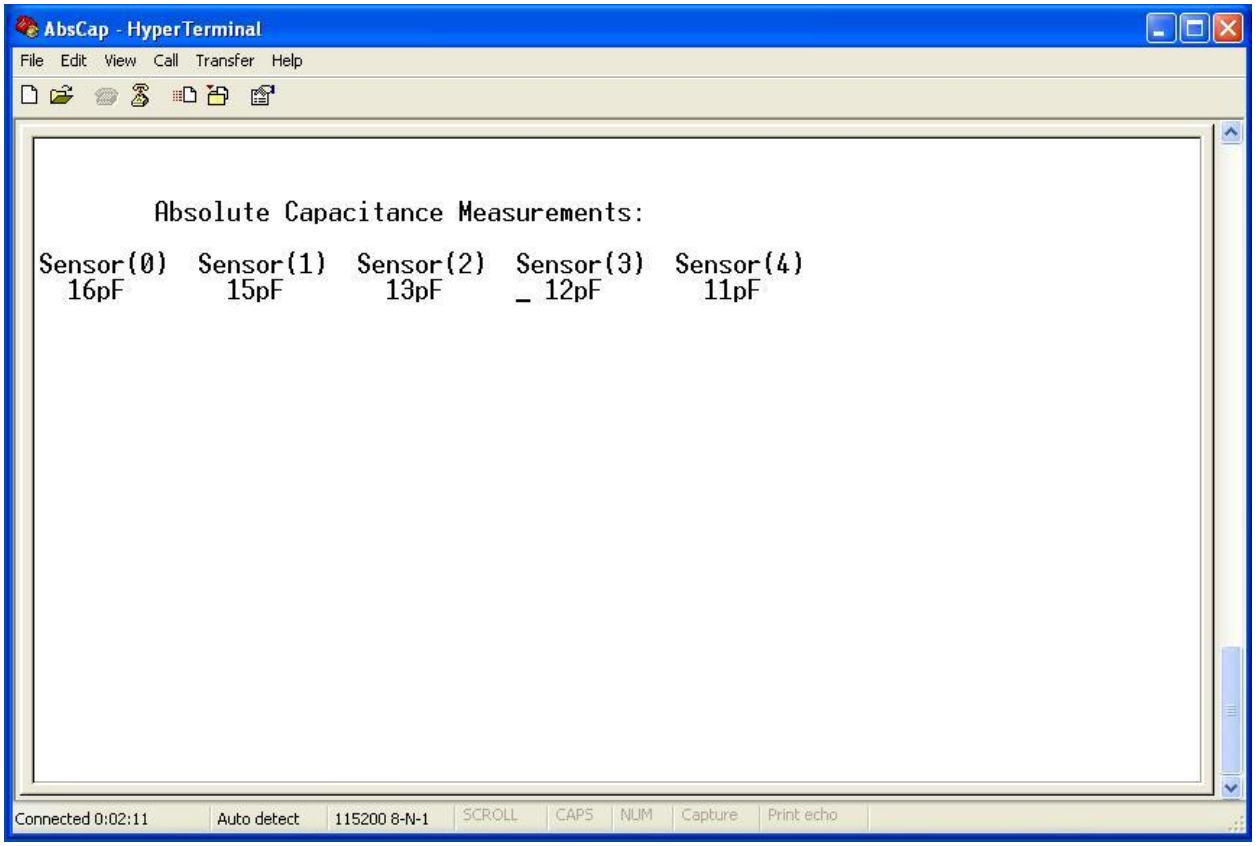
- Bits per second （每秒位数） = 115200
- Data bits（数据位数） = 8
- Parity（奇偶性） = None（无）
- Stop bits（停止位数） = 1
- Flow control（流量控制） = None（无）

8. 点击 **OK**。HyperTerminal 连接着特定的 COM 端口，因此您随时可以使用。



9. 通电并复位 UCC 电路板。

这时，获得的结果将显示在 HyperTerminal 窗口中。


AbsCap - HyperTerminal
File Edit View Call Transfer Help
Absolute Capacitance Measurements:
Sensor(0) Sensor(1) Sensor(2) Sensor(3) Sensor(4)
16pF 15pF 13pF _ 12pF 11pF
Connected 0:02:11 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

代码示例 5　使用 CY8C21x34/B CapSense 控制器测量传感器的绝对电容

5.1　项目名称

CE_5_MeasuringAbsSensorCap_withCY8C21x34

5.2　概述

该代码示例使用 CY8C21x34/B CapSense®控制器计算 5 个传感器的绝对电容，并通过 Windows HyperTerminal 或 Bridge control panel 软件将其显示在计算机上。测量到的绝对电容的精度为 1 pF。

5.3　硬件设置

5.3.1　要求

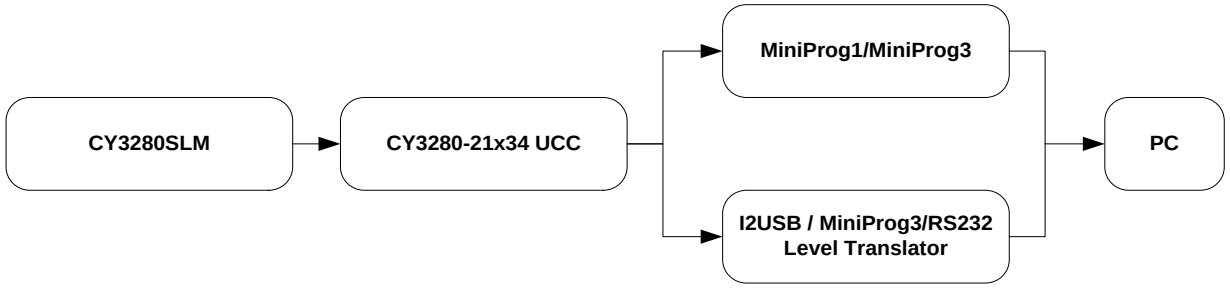
- CY3280-21x34 通用 CapSense 控制器
- CY3280-SLM 通用 CapSense 线性滑条模块
- RS232 电平转换器模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 运行装有 Windows XP 或更高版本的 PC

5.3.2　摘要

图 5-1 演示了硬件的设置方式。使用 22x2_RA_Receptacle 将 CY3280-21x34 UCC 套件连接到 CY3280-SLM 模块上。将 MiniProg1/MiniProg3 连接到插座 J3 上，从而给该套件供电并对其进行编程。

使用Bridge Control Panel软件并通过I2C-USB桥接器或MiniProg3，可以在PC上查看传感器Cp。
通过 HyperTerminal 并使用 RS232 通信协议，可以在 PC 上查看数据。

图 5-1. 硬件设置框图



5.3.3 组装电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 连接到 CY3280-21x34 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）上。
- 在插座 J1 上连接一个跳线器，将 UCC 电路板的引脚 V_{CC} 和 5 V 短接。该设置能够使 ISSP 连接器为 CapSense 控制器供电。
- 在插座 J4 上连接一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3 上。
- 在插座 J2 上连接一个跳线器，将 CY3280SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。该设置会使 CY3280SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 CY3280-21x34 UCC 的 J3 插座上，以编程器件。
- 如果通过 I2C 通信协议在桥接控制面板上查看传感器 Cp，请将 CY3280-21x34 的 J3 插座连接至 I2C-USB 桥接器/MiniProg3，并通过 USB A 转 Mini B 线缆将该桥接器的另一端连接至计算机。通过点击 Bridge control panel 上的“Toggle power”按键给器件供电。
- 如果通过 UART 在 HyperTerminal 上查看传感器 Cp，请将 RS232 电平转换器电路板上的 TX 引脚连接到 UCC 的 P1[7]上。分别将电平转换器的 Vcc 和 GND 引脚连接到 UCC 的 Vcc 和 GND 引脚上。RS232 电平转换器的一个良好示例是 MAX 232 串行电平转换器。
- 使用串行线缆将电平转换器电路板的 RS232 端口连接到计算机上。

5.4 原理图

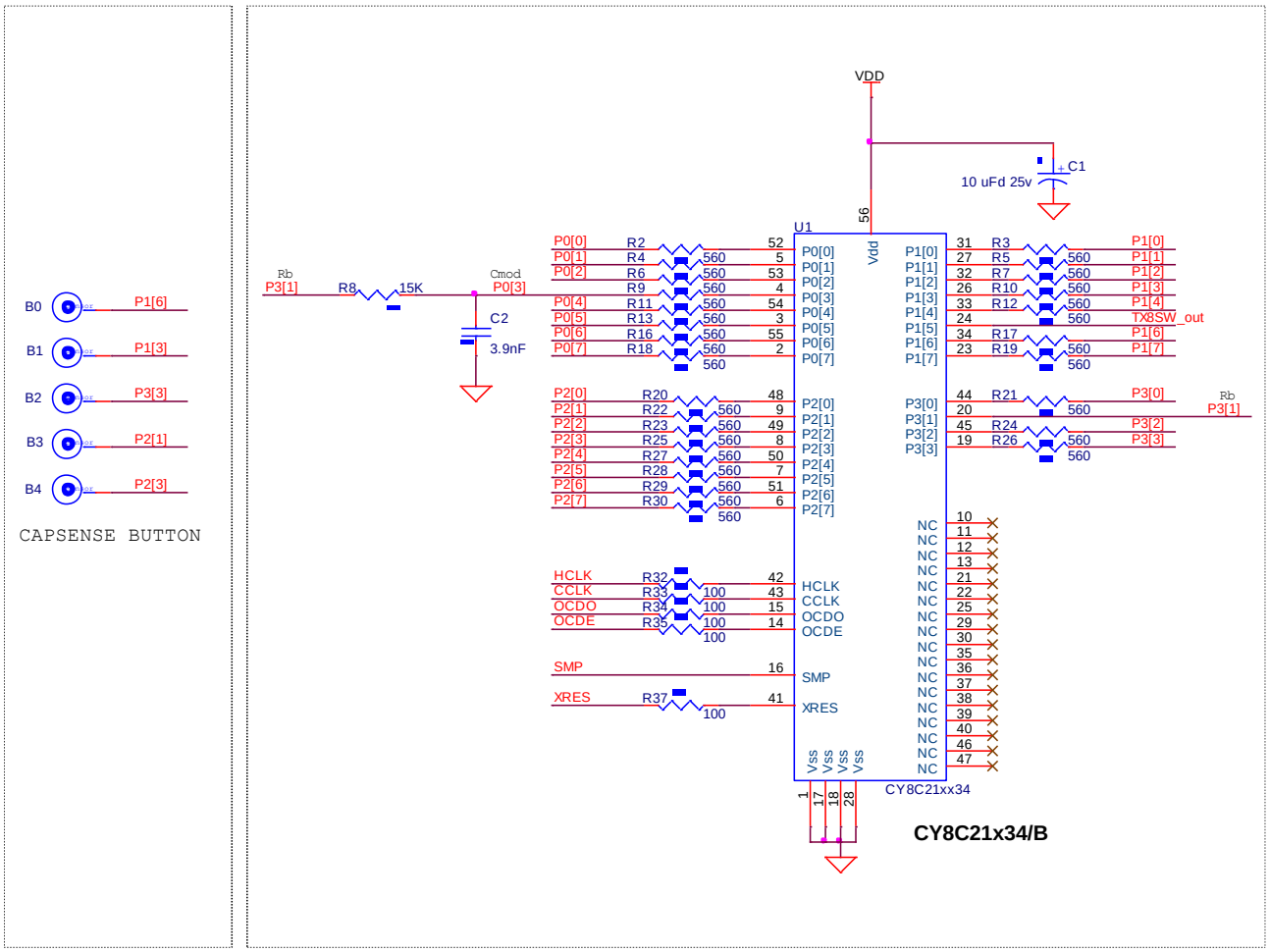


表 5-1. 代码示例 5 的引脚分配

对象	引脚
按键 0	P1[6]
按键 1	P1[3]
按键 2	P3[3]
按键 3	P2[1]
C_{MOD}	P0[3]
Rb	P3[1]

5.5 软件设置

5.5.1 所需的工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel 软件
- Windows HyperTerminal 软件

5.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块（UM）以及各个 UM 所占用的硬件资源。

用户模块	硬件资源
CSD	CapSense 模块，定时器 1（默认）
EzI2C	I2C 模块
TX8SW	未占用模块（软件实现）

5.5.3 用户模块参数、全局资源

项目中的 `ReadMe.txt` 文件说明了该代码示例所使用的各用户模块的参数设置。此外，它还提供了一个全局资源的列表。

5.6 操作说明

该程序采用了以下公式来计算传感器的原始计数：

$$\text{Raw Counts} = C_p \,.\, \text{Sensitivity}$$

其中，C_P 是传感器电容

因此，我们通过使用原始计数与灵敏度之间的比例来计算每个传感器的绝对电容。

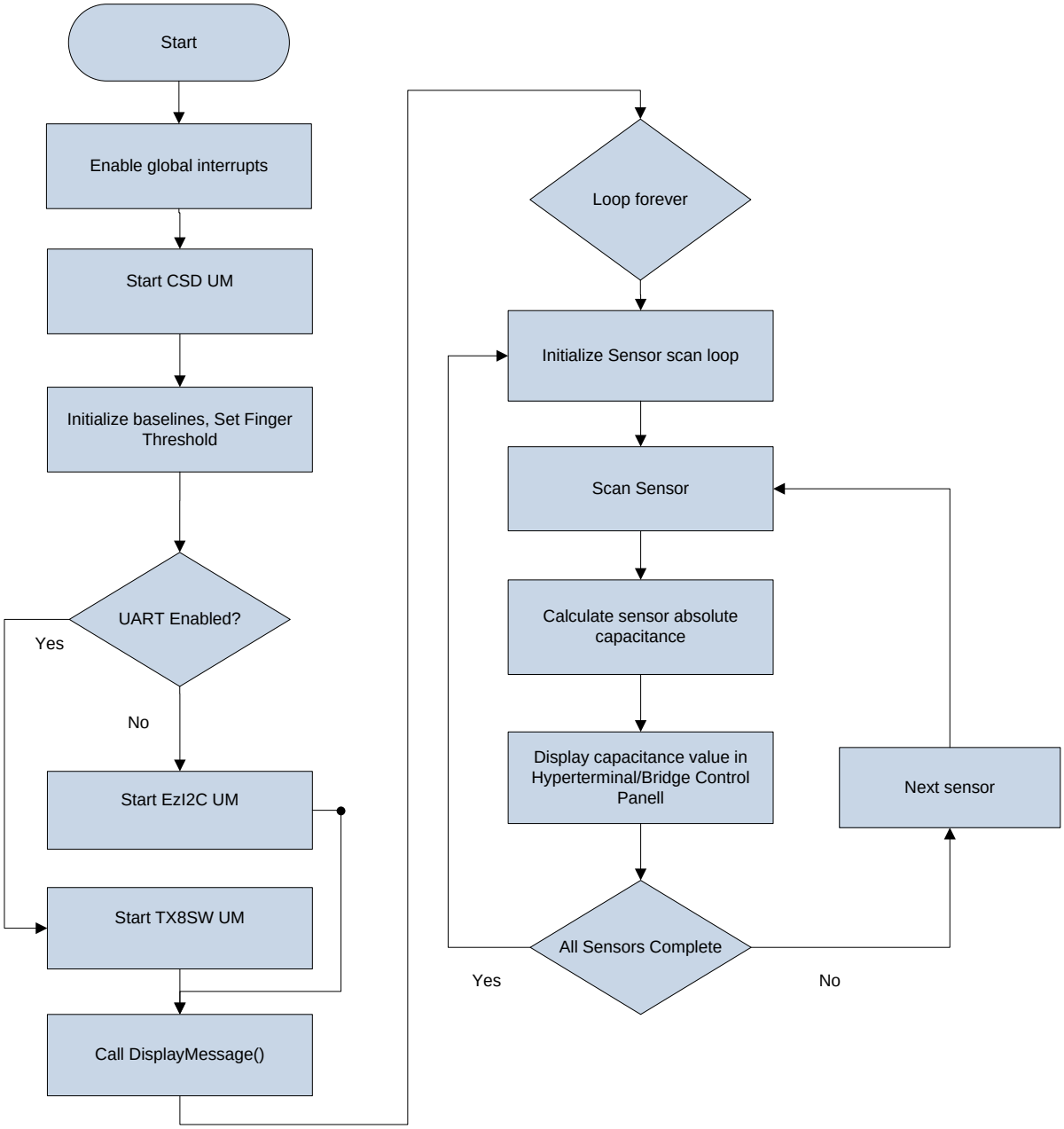
该项目包括一个名为 `CY8C21x34/B_CalculSensitivity.xls` 的文件。并且包含了灵敏度值的计算。下面的“Ref Value”（参考值）和“Prescaler”（预分频器）参数可使精度最好：

- Ref Value = 5
- Prescaler = 5

该程序按照下面流程将绝对电容显示在屏幕上：

1. 使用该配置时，计算出的灵敏度等于 465 计数/pF。请参考 *CY8C21x34/B*_CalculSensitivity.*xls*。
   该“.xls”文件是项目的一部分，它被存储在项目文件夹中。
2. 使能全局中断，并启动用户模块。
3. 按顺序对每个传感器进行扫描，并从 CSD_waSnsResult[]数组中获取原始计数。
4. 通过使用原始计数与灵敏度的比例来计算电容的绝对值。
5. 将结果发送给计算机，并根据用户的选择，将其显示在 Bridge Control Panel 或 HyperTerminal 中。

图 5-2. 代码示例 5 的功能流程



5.7 执行代码示例

编程电路板

请确保在 *main.c* 文件中添加了注释语句“#define UART ”（如下屏截图所示），然后生成项目（按下 F6 键）并对电路板进行编程。有关如何编程 UCC 电路板的详细信息，请参考 CY3280-21x34 套件指南中的第五章节。

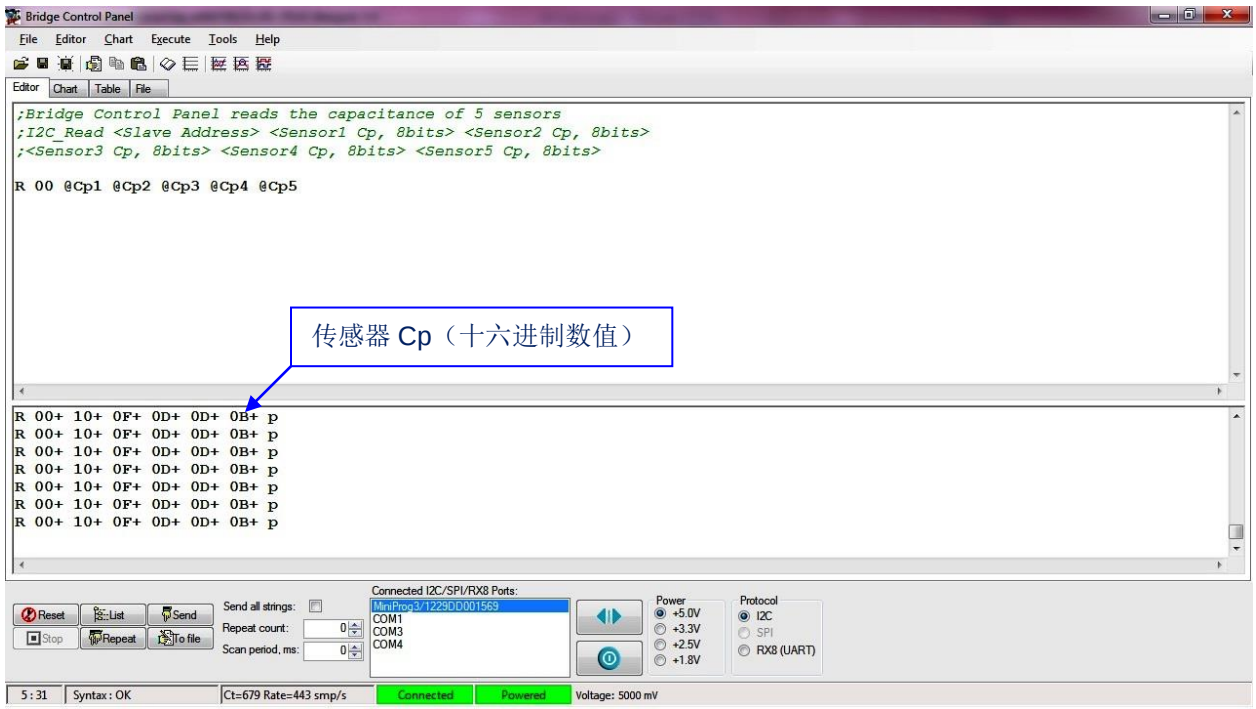
```
/* comment this while using I2C;Uncomment this while using UART */

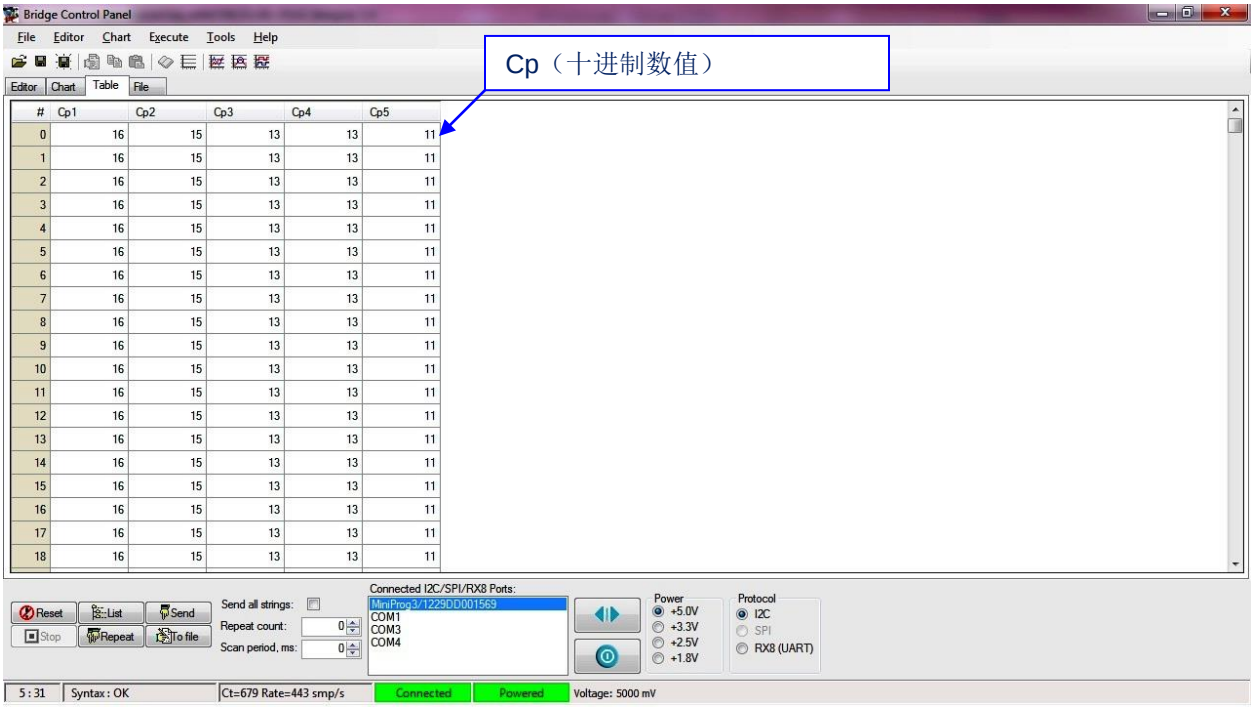
 //#define UART

```

使用 BCP 监控传感器数据

1) 将 CY3280-21x34 的 J3 插座连接到 I2C-USB 桥接器或 MiniProg3 上，并通过 USB A 转 Mini B 线缆将 I2C-USB桥接器/MiniProg3的另一端连接到计算机上。
2) 在电脑桌面上，依次选择**Start** > **All Programs** > **Cypress** > **Bridge Control Panel 版本）** > **Bridge Control Panel（版本）**。
3) 从端口选择窗口中选择相应设备。
4) 为CY3280-21x34 UCC套件电路板提供5 V的电源。
5) 在Bridge Control Panel窗口中，依次选择**File** > **Open**。从项目文件夹中的BCP Configuration Files文件夹内加载BCP.iic文件。
6) 依次选择**Charts** > **Variable Settings**。从项目文件夹中的BCP Configuration Files文件夹内加载*BCP.ini*文件。
7) 点击“send”按键，将I²C指令发送给CY8C21x34控制器，从而在底部窗口中获取传感器C_p的值（十六进制），如下屏截图所示。





编程电路板

在“main.c”中取消注释语句“#define UART”（如下图所示），并编程电路板。

```
76 /* comment this while using I2C;Uncomment this while using UART */
77
78  #define UART
79
```

使用 HyperTerminal 监控传感器数据。

1. 使用Miniprog1或Miniprog3给电路板供电。
2. 在电脑桌面上，依次选择**Start** > **All Programs** > **Accessories** > **Communication** > **HyperTerminal**。
3. 为该连接键入名称。
4. 点击**OK**按键。

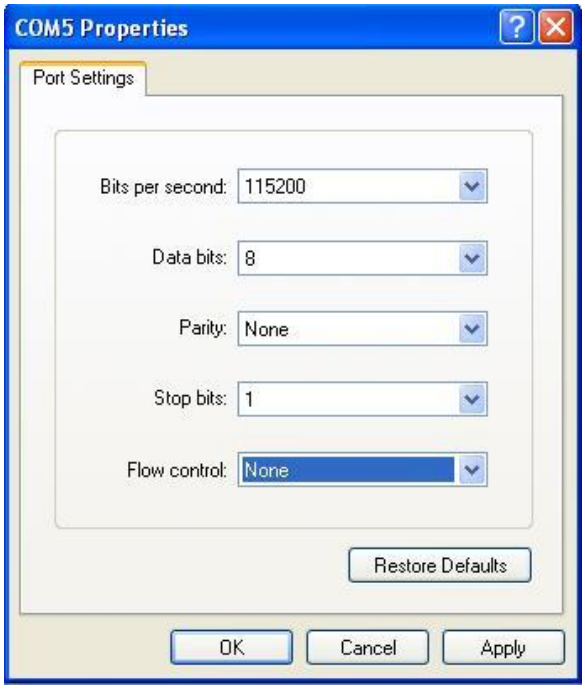


5. 通过“Connect using”框中的下拉列表，选择用于接收PSoC数据的串行端口。该设置因PC配置的不同而异。在我们的示例中，我们使用的是COM5。
6. 单击“OK”。

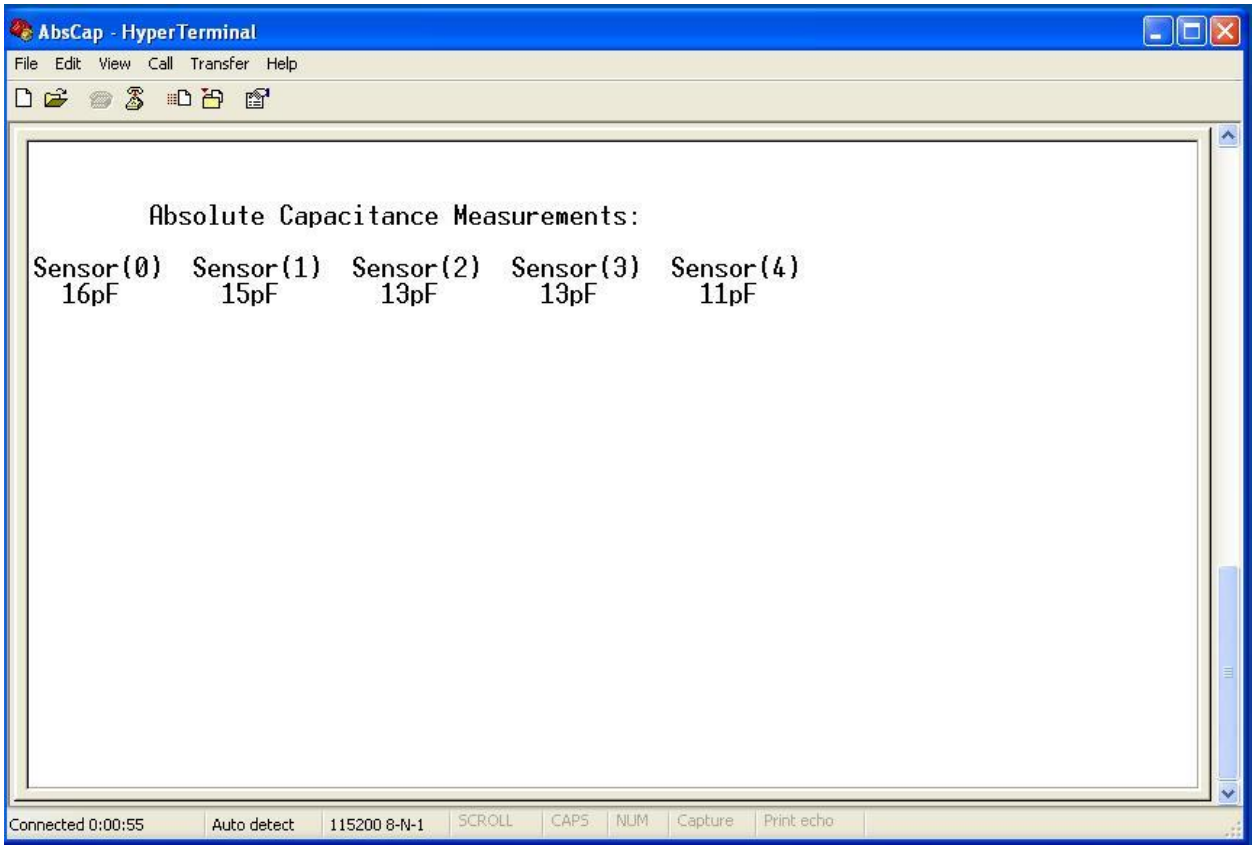


6. 在“COM 属性”对话框中，配置以下参数：
   - Bits per second（每秒位数） = 115200
   - Data bits（数据位数） = 8
   - Parity（奇偶性） = None（无）
   - Stop bits（停止位数） = 1
   - Flow control（流量控制） = None（无）
7. 点击 **OK**。

   HyperTerminal 连接着特定的 COM 端口，因此您随时可以使用它。



8. 通电并复位 UCC 电路板。
   这时，获得的结果将显示在 HyperTerminal 窗口中。



代码示例 6　使用 CY8C20x34 CapSense 控制器测量传感器的绝对电容

6.1　项目名称

CE_6_MeasuringAbsSensorCap_withCY8C20x34

6.2　概述

该代码示例使用 CY3280-20x34 CapSense®控制器来计算五个传感器的绝对电容。测量的绝对电容精度为 1 pF。

6.3　硬件设置

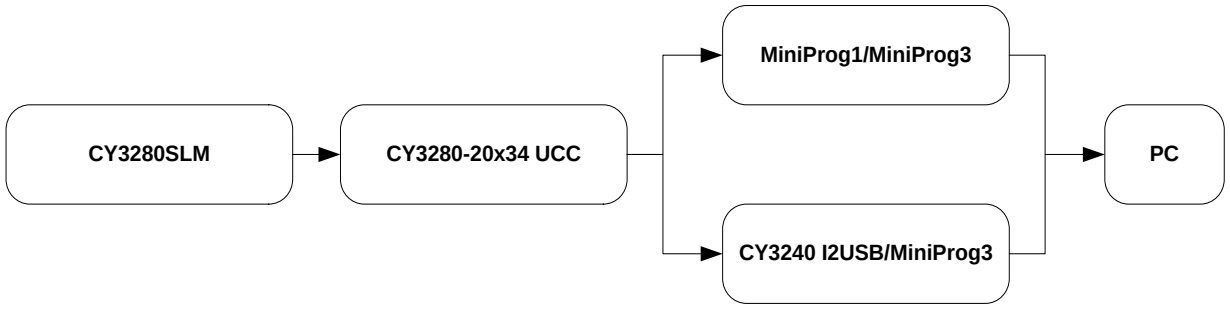
6.3.1　要求

- CY3280-20x34 通用 CapSense 控制器
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3217 – MiniProg1 编程器或 CY8CKIT-002 Miniprog3
- CY3240 – I2USB 桥接器或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 装有 Windows XP 或更高版本的 PC
- 万用表

6.3.2　摘要

图 6-1 演示了硬件的设置方式。CY3280-20x34 UCC 套件通过 22x2_RA_Receptacle 连接到 CY3280-SLM。将 MiniProg1/Miniprog3 连接至插座 J3，从而给该套件供电并对其进行编程。

图 6-1. 硬件设置框图

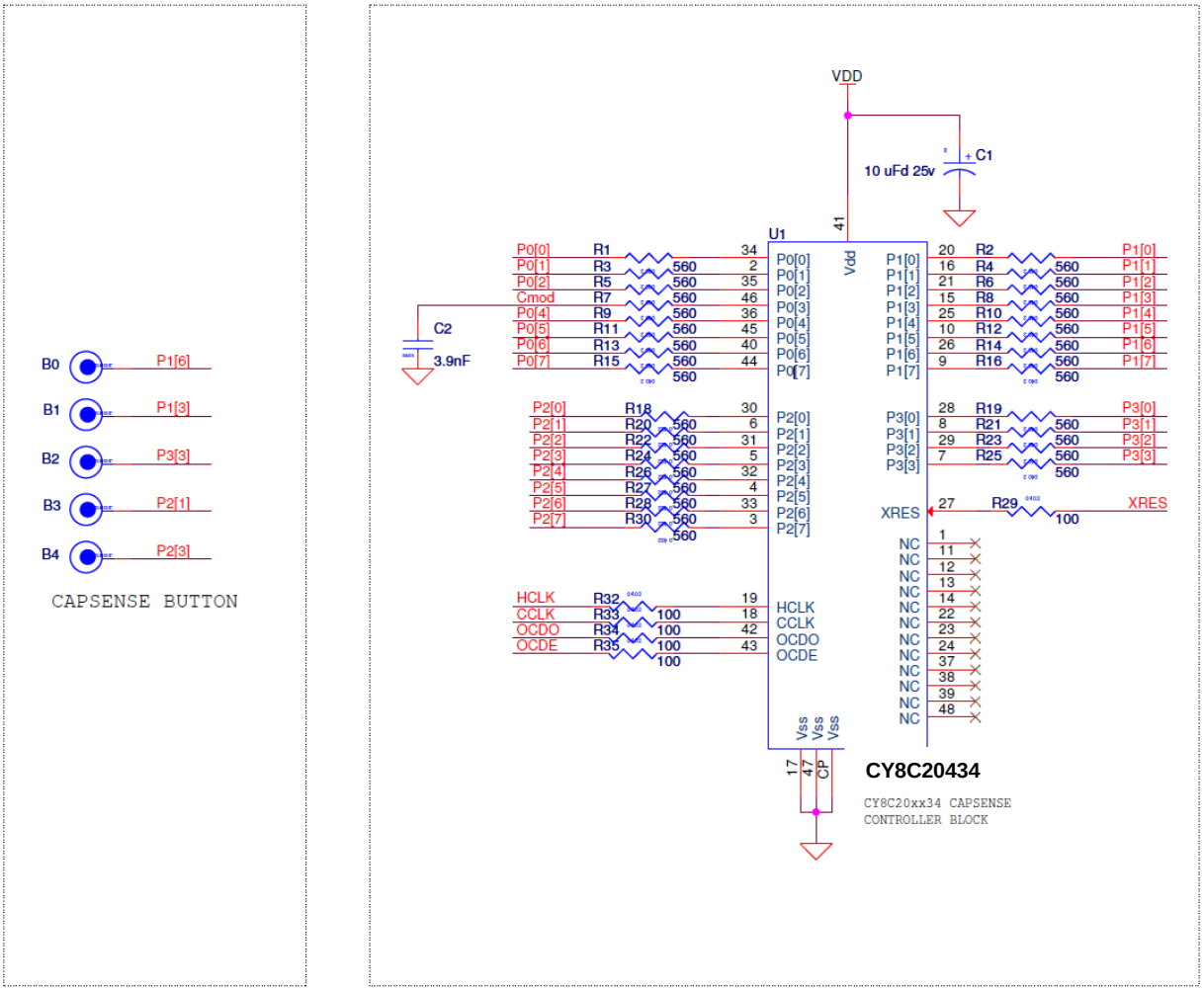


6.3.3 组装电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 连接到 CY3280-20x34 UCC 电路板的 22x2_RA_Receptacle（连接器 P2）上。
- 在插座 J1 上连接一个跳线器，将 UCC 电路板的引脚 V_{CC} 和 5 V 短接。该设置能够使 ISSP 连接器为 CapSense 控制器供电。
- 在插座 J4 上连接一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上连接一个跳线器，将 CY3280SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。该设置会使 CY3280SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接至 UCC 电路板的 ISSP 插座 J3 上。只有使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。要想通过 Bridge Control Panel 软件中取 CapSense 数据，那么需要使用 I2USB 代替 MiniProg1/Minirpog3 进行连接。
- 使用 USB A 转 Mini B 线缆将 MiniProg1/Miniprog3（或 I2USB 桥接器）的另一端连接至 PC 上。

6.4 原理图



调制器电容器（C_{MOD}）是一个连接至 P0[3]并且大小为 3.9 nF 的电容。在各个 CapSense 按键上串联一个 560 Ω 的电阻，以减少射频干扰。

表 6-1. 代码示例 6 的引脚分配

按键	引脚
0	P1[6]
1	P1[3]
2	P3[3]
3	P2[1]
4	P2[3]

6.5 软件设置

6.5.1 需要的工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC 编程器（版本 3.13 或更高版本）
- Bridge Control Panel 工具

6.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块（UM）和各个 UM 所占用的硬件资源。

用户模块	硬件资源
CSA_EMC	CapSense 模块，比较器（默认）
EzI2C	I^2C/SPI

6.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例所使用的各用户模块的参数设置。此外，它还提供了一个全局资源列表。

6.6 操作说明

main.c 文件有两个函数。它会计算出每个传感器的 IDAC 代码。然后，它会将均衡的电流量导向端口，用户可以通过该端口物理测量电流值。我们使用以下公式计算每个传感器的绝对电容：

$$C_P = \frac{I_{measured}}{\left(IMO/8\right) \times 1.3}$$

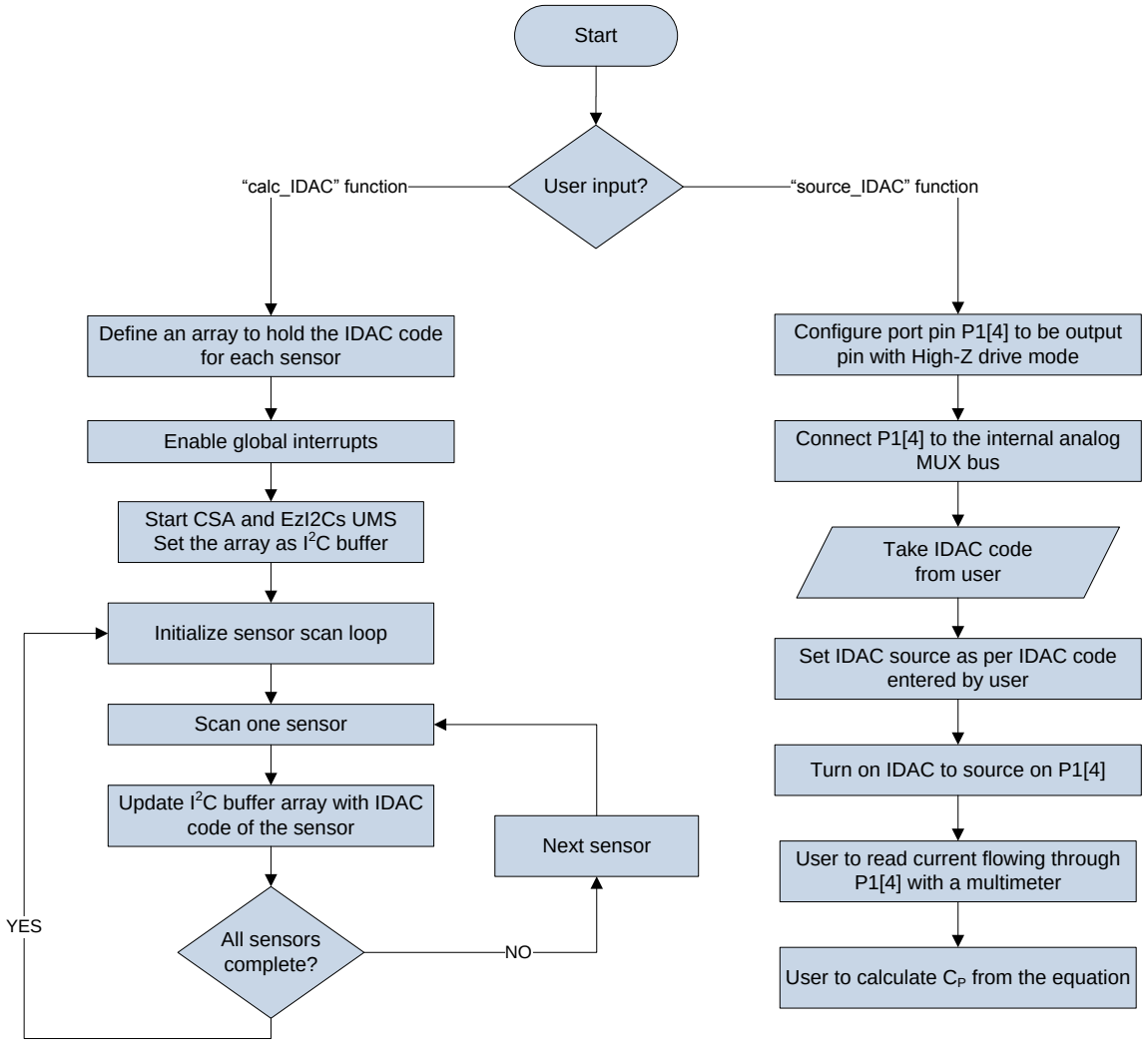
其中，C_P 是传感器电容。

该程序完成了以下步骤：

- 定义两个函数 calc_IDAC 和 source_IDAC，然后根据用户输入值来调用其中一个函数。
- 第一个函数 calc_IDAC 会执行以下操作：
  - 定义一个阵列 baIDAC，以根据传感器编号来存储每个 CapSense 传感器的 IDAC 代码。
  - 将该阵列设置为 I^2C 读取缓冲区。
  - 使能全局中断，然后启动用户模块。
  - 在无限循环中，依次扫描每个传感器，并获取存储在阵列中的 IDAC 代码。此外，它还会更新 I^2C 缓冲区。
- 第二个函数 source_IDAC 会执行以下操作：

- 将端脚 P1[4]配置为 High-Z 驱动的输出引脚。
- 将引脚 P1[4]连接到内部模拟复用器总线上。
- 读取用户输入的 IDAC 代码。
- 在引脚 P1[4]上提供相应的电流。

图 6-2. 代码示例 6 的功能流程



6.7 执行代码示例

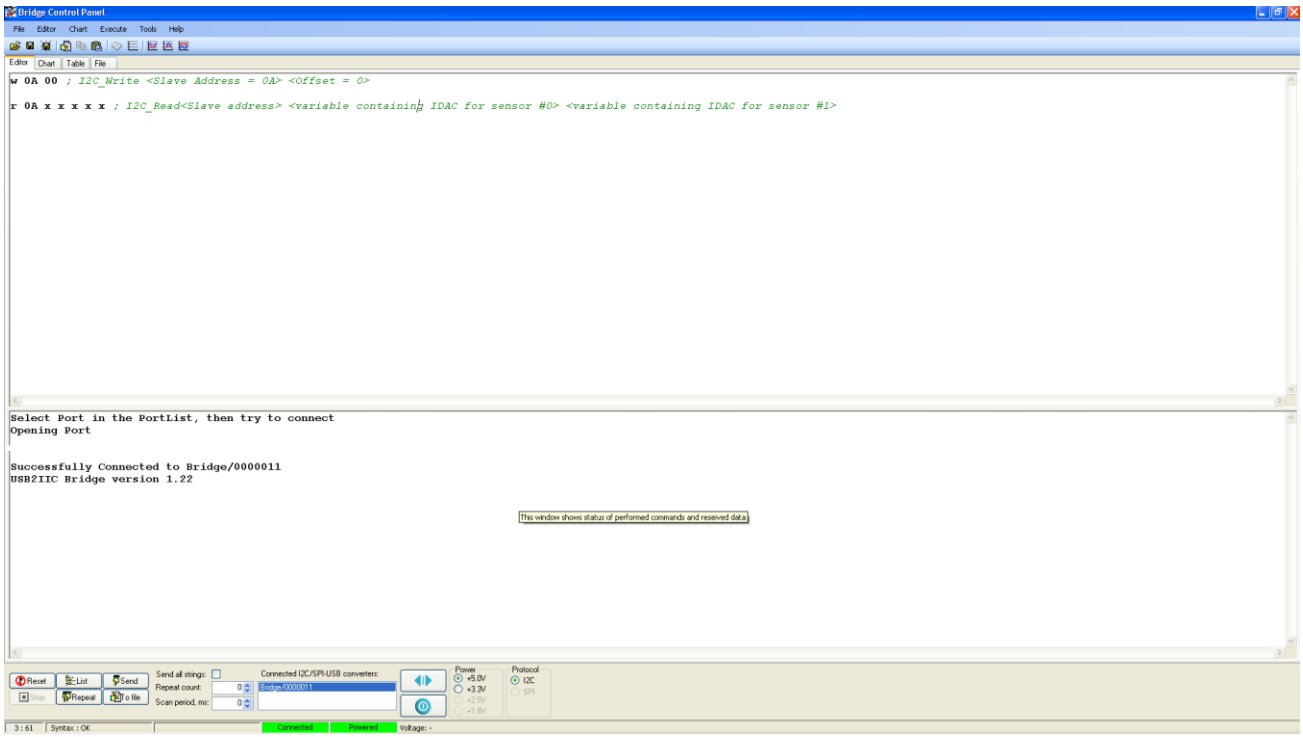
使用项目编程电路板，然后按照该程序执行代码示例。有关编程 UCC 电路板的详细信息，请参考 CY3280-20x34 套件指南中的第五章节。

1. 使用 I2USB 桥接器和 USB A 转 Mini B 线缆，将您的计算机和 CY3280-20x34 通用 CapSense 控制器电路板上的 ISSP 连接器 J3 连接起来。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。

   Bridge Control Panel 是 PSoC Programmer 安装的一个组件。
3. 从 Port selection 窗口中选择相应的设备。

4. 为 CY3280-20x34 CapSense 控制器电路板提供 5 V 的电源。
5. 在 Bridge Control Panel 窗口中，依次选择 **Tools** > **Protocol Configuration**。在 **I2C** 选项卡中，将 **I2C Speed** 设置为 100 kHz。
6. 依次选择 **File** > **Open**。从代码示例项目文件夹的 *BCP Configuration files* 文件夹中加载 `BCP.iic` 文件。

图 6-3. 在状态窗口中显示从器件地址



7. 按下[**Enter**]键，执行 Bridge Control Panel 中的第二个指令。
   状态窗口将显示所有传感器的 IDAC 代码。
8. 获取五个传感器专用的 IDAC 代码后，打开项目，取消对语句 *source_IDAC()*的注释，并对语句 *calc_IDAC()*添加注释，以搜寻端脚 P1[4]上相应电流的来源。
9. 在功能 *source_IDAC()*中，通过给 IDAC_D 变量分配适当的十六进制数值，将 IDAC 代码设置为任意一个传感器在 Bridge Control Panel 中已获得的代码。
10. 使用 PSoC Designer 编译项目。依次选择 **Build** > **Generate** 或 **CE_6_MeasuringAbsSensorCap_withCY8C20x34** 项目。确保在编译过程中没有发生任何错误或警报。
11. 要想使用所生成的十六进制文件来编程 UCC，请依次选择 **Program** > **Program Part**。
12. 在 Program Part 窗口中，确认已经连接了 MiniProg1/Miniprog3。如果未连接，请点击 **Connect** 按键。
13. 点击 **Program** 按键。
14. 断开 CY3280-SLM 电路板和 UCC 的连接。
15. 点击电源按键，为 UCC 和 MiniProg1/Miniprog3 通电。
16. 在引脚与地面间放置一个电流表，测量通过 P1[4]($I_{MEASURED}$)的电流。
17. 使用下面的公式计算 C_P：

$$C_P = \frac{I_{measured}}{\left(IMO/8\right) \times 1.3}$$

    其中，IMO 是系统时钟源。在本项目中，它被设定为 12 MHz。

18. 重复步骤 9 到 17，以获取其他传感器的 C_P 值。

表 6-2 列出了 CapSense 传感器的 C_P 值以及通过 Bridge Control Panel 获得的 IDAC 代码和引脚上的 IDAC 电流。

表 6-2. 测量到的 C_P

CapSense 传感器	IDAC 代码	端口引脚上的 IDAC 电流（µA）	测量到的 C_P 值（pF）
P1[6]	0A	28.0	14.40
P1[3]	09	25.3	12.98
P3[3]	08	22.6	11.58
P2[1]	07	19.3	9.90
P2[3]	07	19.3	9.90

代码示例 7　使用 CY8C20xx6A CapSense 控制器调试 CSD

7.1　项目名称

CE_7_TuningCSD_withCY8C20xx6A

7.2　概况

该代码示例演示了如何调校 CY8C20xx6A 设备的 CapSense® UM 模块。要想调校 CapSense UM，必须查看 CapSense 数据的图形格式。该代码示例使用 EzI2C UM 并通过一条 I²C 总线和一个 I2USB 桥接器来传输所需要的 CapSense 参数。它使用了 PC 上的 Bridge Control Panel 软件来查看数据。

7.3　硬件设置

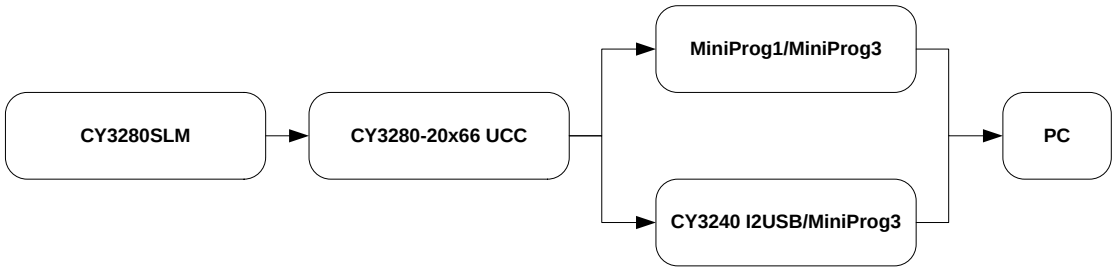
7.3.1　要求

- CY3280-20x66 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 装有 Windows XP 或更高版本的 PC

7.3.2　摘要

图 7-1 演示了硬件的设置方式。使用一个 22x2_RA_Receptacle 将 CY3280-20x66UCC 套件和 CY3280-SLM 模块连接起来。将 MiniProg1/MiniProg3 或 I2USB bridge/MiniProg3 连接到套件的 ISSP 插座上。该设置使用 MiniProg1/MiniProg3 进行编程，并使用 I2USB 桥接器/Miniprog3 将数据发送给 PC。使用 USB 线缆将它们连接到 PC 上。

图 7-1. 硬件设置框图

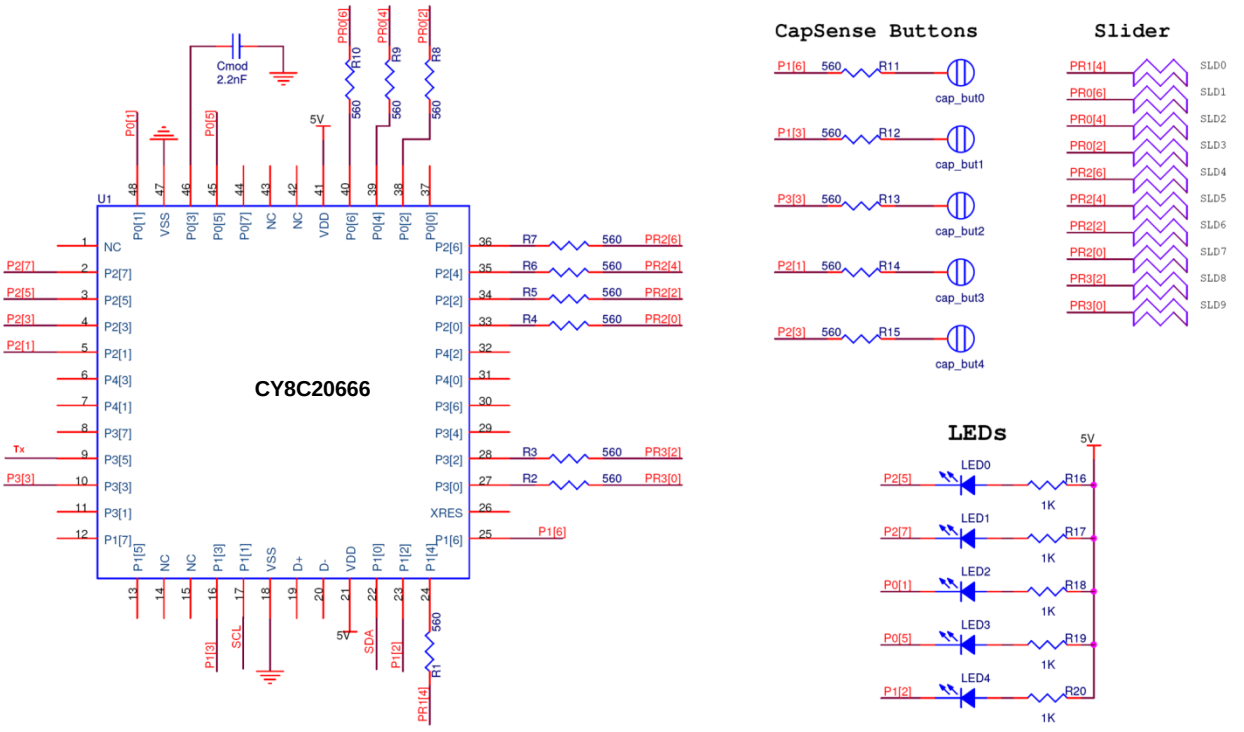


7.3.3 组装电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 和 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）连接起来。
- 在插座 J7 上放置一个跳线器，将 UCC 电路板的引脚 VCC 和 VCC_PROG 短接。该设置能够使 ISSP 连接器为 CapSense 控制器供电。
- 在插座 J4 上连接一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上连接一个跳线器，将 CY3280SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。该设置会使 CY3280SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接至 UCC 电路板的 ISSP 插座 J3 上。仅当使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。要想通过 Bridge Control Panel 软件读取 CapSense 数据，那么需要使用 12USB 桥接器代替 MiniProg1/Miniprog3 进行连接。
- 使用 USB A 转 Mini B 线缆来将 MiniProg1/Miniprog3（或 I2USB 桥接器）的另一端连接到 PC 上。

7.4 原理图



调制器电容（C_{MOD}）指的是 P0[3]引脚上连接的 2.2 nF 大小的电容。使用一个 560 Ω 的电阻串联各个 CapSense 按键，以降低射频干扰。将 LED 配置为低电平有效，并将其与 1 kΩ 的电阻串联起来。该原理图中共有五个 LED、五个按键以及一个包含 10 个可用段的滑条。

表 7-1. LED、按键和滑块的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

7.5 软件设置

7.5.1 所需工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.14 或更高版本）
- Bridge Control Panel（桥接控制面板）

7.5.2 用户模块列表和放置

下表列出了该代码示例中所使用的用户模块以及各个用户模块所占用的硬件资源。

用户模块	放置
CSD	CapSense 模块、定时器 1（默认）
EzI2C	I^2C/SPI 模块

7.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例中所使用的用户模块的参数设置。此外，它还提供了一个全局资源列表。

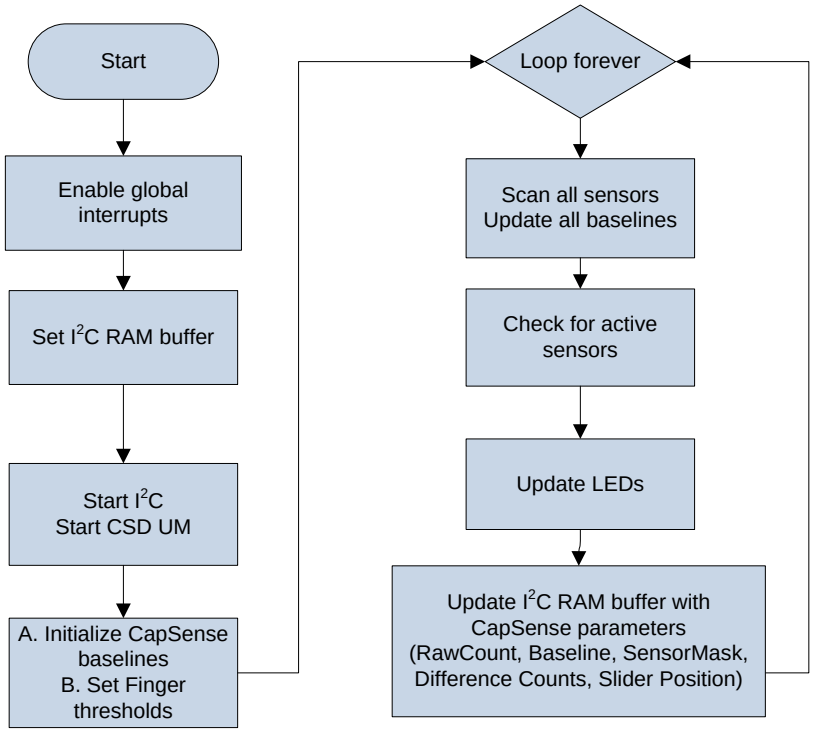
注意： 我们对该代码示例进行了调校，使之符合 CY3280-SLM 套件中 1.5 mm 厚的亚克力覆盖层。要想使用具有更厚覆盖层的代码示例，请按照该代码示例中所述的内容调校项目。

7.6 操作说明

复位时，固件将执行以下操作：

- 定义一个结构体 MyI2C_Regs 以存储按键编号、原始计数、计数差值、基准线、中心位置以及同该按键编号相应的 CapSense 按键的状态。
- 使能全局中断，然后启动 CSD UM。
- 启动 EzI2Cs UM，然后将 MyI2C_Regs 结构体设置为 I^2C RAM 缓冲区。
- 无限循环地执行下面操作：
  - 继续扫描所有传感器，并为 MyI2C_Regs 结构体更新原始计数、计数差值、基准线、滑条中心位置以及选定 CapSense 按键的状态。通过将按键编号写入到 EzI2C 从设备中 I^2C 缓冲区的第一个字节内，I^2C 可以请求某个特定按键的 CapSense 数据。
  - 当固件检测到按键触摸时，它会将相应的 LED 状态切换为“ON”。释放按键时，LED 状态会切换为“OFF”。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”，以表示触摸位置。

图 7-2. 代码示例 7 的功能流程



7.7 执行代码示例

使用项目来编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-20x66 套件指南中第五章的内容。

1. 使用 MiniProg1/MiniProg3 或 CY3280-20x66 UCC 套件指南中介绍的某个源给该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。
   CY3280-SLM 电路板上相应的 LED 会点亮。
3. 触摸某个按键。
   CY3280-SLM 模块电路板上相应的 LED 会点亮。您可以同时触摸多个按键。也可以同时使用线性滑条和按键。

7.8 通过 I²C 读取 CapSense 数据

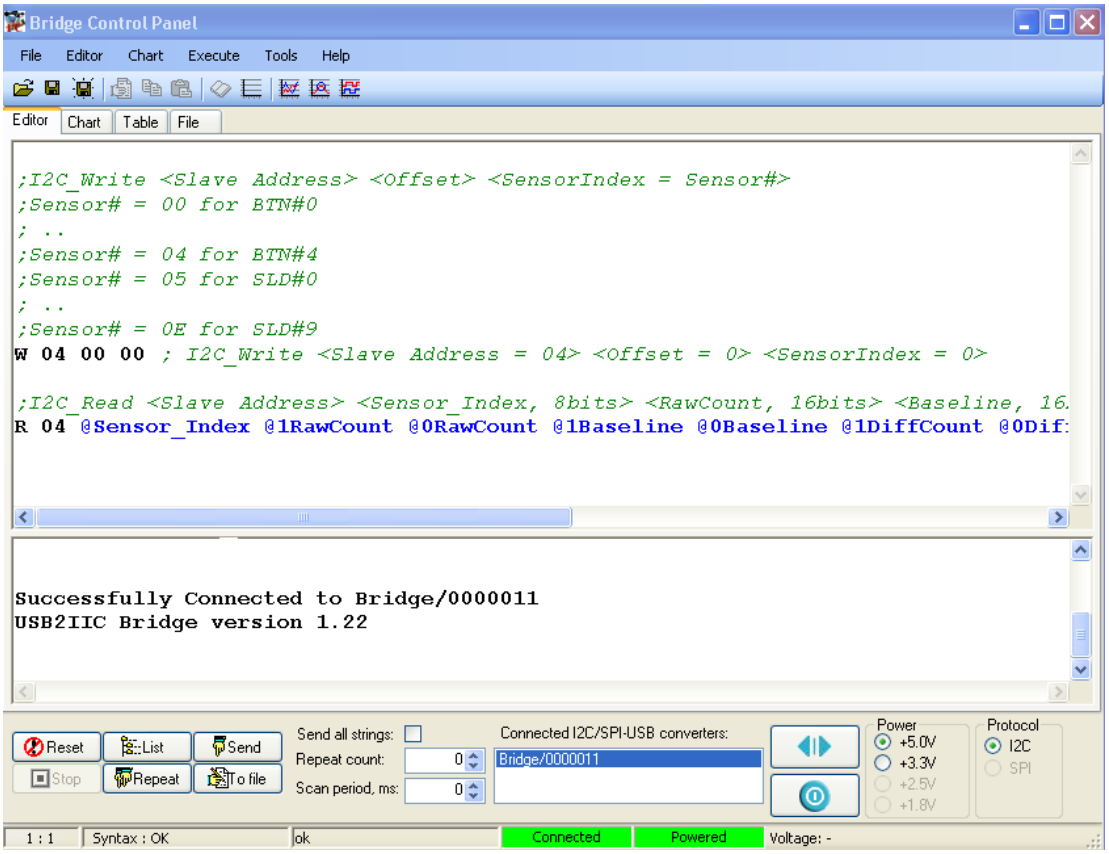
按照该程序执行代码示例并读取 Bridge Control Panel 工具上的 CapSense 数据。更多有关桥接控制面板工具的详细信息，请参考 AN2397，“CapSense®数据查看工具”。

7.8.1 加载桥接控制面板工具

1. 使用 I2USB 桥接器/MiniProg3 和 USB A 转 Mini B 线缆将您的电脑连接到 CY3280-20x66 UCC 电路板的 ISSP 连接器 J3 上。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。
   Bridge Control Panel 是在 PSoC Programmer 安装过程中安装的一个组件。
3. 从端口选择窗口中选择您需要的器件。
4. 为 CY3280-20x66 CapSense 控制器电路板提供 5 V 的电源。

5. 在 Bridge Control Panel 窗口中，依次选择 **File** > **Open**。从代码示例项目文件夹中的 *BCP Configuration files* 文件夹加载 `BCP.iic` 文件。
6. 依次选择 **Charts** > **Variable Settings**。从代码示例项目文件夹中的 *BCP Configuration files* 文件夹加载 `BCP.ini` 文件。单击 **OK**，返回到主窗口。

图 7-3. Bridge Control Panel 视图



7.8.2 读取 BTN0 的原始计数、基准线和计数差值

7. 发送 I²C 写指令 W 04 00 00 一次。
8. 按下 **Repeat** 按键，以连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

9. 点击 **Chart**（图表）选项卡，以查看 BTN0 的原始计数、基准线和计数差值。

7.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置

1. 发送 I²C 写指令 W 04 00 05 一次。
2. 按下 **Repeat** 按键，以连续发送以下 I²C 读指令：

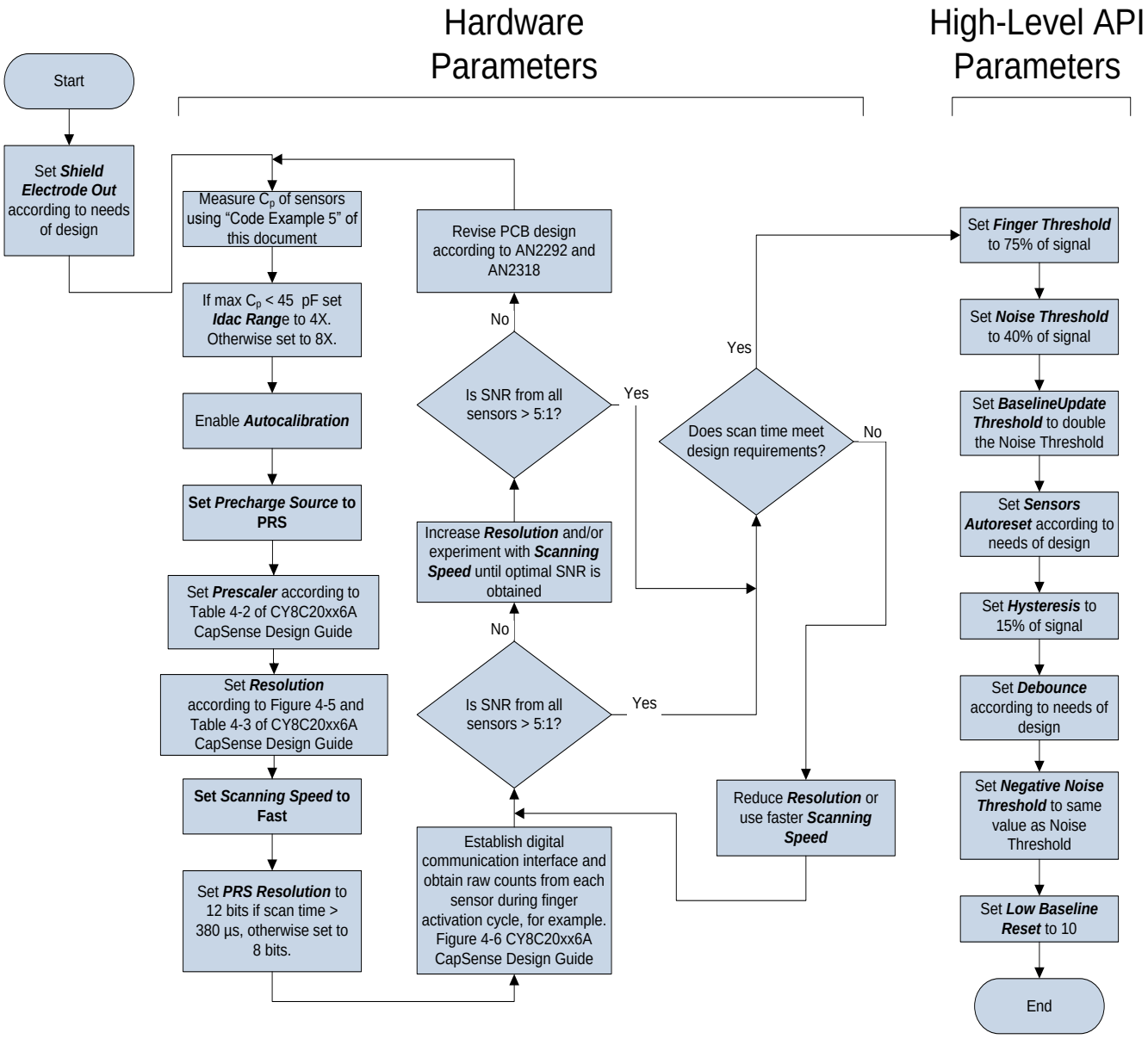
```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，查看 SLD0 的原始计数、基准线、计数差值和滑条位置。

7.9 调校 CSD 用户模块

在本节中，我们按照下面流程图中的程序对用户模块的各个参数进行了调校。更多详细信息，请参见 CY8C20xx6A CapSense 设计指南。

图 7-4. 调校 CSD 用户模块的流程图



7.9.1 配置 CSD UM 设置

使用 PSoC Designer 工具，按照以下要求输入 CSD UM 参数设置：

1. 不需要屏蔽电极。因此，可将 **ShieldElectrodeOut** 参数设置为 **None**。
2. 表 7-2 列出了使用*代码示例 3（使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容）*部分介绍的方法时获得的传感器 C_P。由于所有传感器的 C_P 值均低于 45 pF，因此请将 IDAC 范围设置为 4X。

表 7-2. 所有传感器的 C_P 值

传感器编号	引脚	C_P
BTN0	P1[6]	16 pF
BTN1	P1[3]	14 pF
BTN2	P3[3]	13 pF
BTN3	P2[1]	12 pF
BTN4	P2[3]	11 pF
SLD0	P1[4]	14 pF
SLD1	P0[6]	15 pF
SLD2	P0[4]	14 pF
SLD3	P0[2]	14 pF
SLD4	P2[6]	13 pF
SLD5	P2[4]	12 pF
SLD6	P2[2]	12 pF
SLD7	P2[0]	12 pF
SLD8	P3[2]	11 pF
SLD9	P3[0]	11 pF

3. 使能 **AutoCalibration**（自动校准）参数。
4. 将 **Precharge** 源设置为 **PRS**。
   PRS 源能够提供扩频的操作，并确保对外部噪声源提供良好的免疫性。
5. 根据所有传感器的最高 C_P 值来设置 **Prescaler**（预分频器）。由于我们的示例中 C_P 值最大为 16 pF，因此可将预分频器的值设置为 4。
   更多有关信息，请参见 CSD UM 数据手册中的“基于预充电源、IMO 和 CP 的预分频器设置”列表。
6. 如果一个项目使用了不同 C_P 值的传感器，那么请根据 C_P 值最大的传感器设置**分辨率**。在我们的示例中，最大 C_P 值为 16 pF，并且 Cf 大于 0.2 pF。参考 CSD UM 数据手册中的图“基于覆盖层厚度和圆形传感器直径的手指电容（CF）”和表“基于手指电容和 CP 的分辨率设置”，将分辨率设置为 13。
7. 将 **Scan speed** 设置为 **Fast**。
8. 扫描时间大于 380 µs。参考 CSD UM 数据手册中的“基于分辨率和扫描速度的单个传感器的扫描时间（单位为 µs）”列表，时间值为 720 µs，因此可将 **PRS Resolution** 设置为 8 位。
9. 表 7-3 列出了 **Bridge Control Panel** 上观察到的不同传感器的噪声（即原始计数在无手指触摸时的变量）和信号（手指响应）的值。

表 7-3. 所有传感器的信号

传感器编号	平均原始计数（无手指触摸）	噪声	信号	SNR（取最接近的整数值）
BTN0	7072	24	260	11
BTN1	7017	22	280	13
BTN2	7103	24	280	12
BTN3	7275	22	340	15
BTN4	6953	20	360	18
SLD0	6627	26	165	6

传感器编号	平均原始计数（无手指触摸）	噪声	信号	SNR（取最接近的整数值）
SLD1	7071	24	150	6
SLD2	6697	24	150	6
SLD3	6505	24	145	6
SLD4	6028	22	160	7
SLD5	5840	28	155	5
SLD6	5687	26	160	6
SLD7	5598	24	150	6
SLD8	5362	24	150	6
SLD9	5110	30	160	5

10. **由于所有传感器的 SNR 均大于 5:1，且扫描时间已足够低，手指触摸能够使** LED 立即发光，因此您可以按照表 7-4 **中的内容设置阈值参数。**

    我们根据所有传感器间最低的信号值配置阈值。因此，我们示例中的信号值为 145。

    表 7-4. CSD UM 的高级参数

阈值参数	推荐值	数值	UM 参数设置
手指阈值	信号值的 75%	108.75	108
噪声阈值	信号值的 40%	58	58
基准线更新阈值	2*噪声阈值	116	116
负噪声阈值	噪声阈值	58	58
迟滞	信号值的 15%	21.75	21

11. 由于扫描时间满足要求，并且没有出现虚假的正极报告，因此可将 **Low Baseline Reset**（低基准线复位）参数设置为 10。将 **Debounce**（去抖动）参数设置为 3。

代码示例 8　通过 CY8C21x34/B CapSense 控制器并使用反馈电阻来调校 CSD

8.1　项目名称

CE_8_TuningCSD_withCY8C21x34/B

8.2　概述

该代码示例说明了如何为 CY8C21x34/B 器件调校 CSD 用户模块（UM）。要调试 CapSense UM，您必须能够以图形格式查看 CapSense 数据。该代码示例使用了 EzI2C UM，并通过一个 I²C 总线和一个 I2USB 桥接器来传输所需的 CapSense 参数。它使用了 PC 上的 Bridge Control Panel 软件来查看数据。

8.3　硬件设置

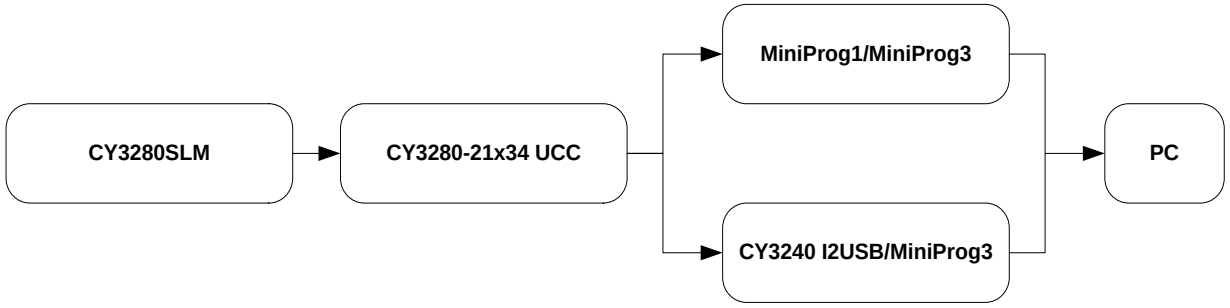
8.3.1　要求

- CY3280-21x34 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 运行装有 Windows XP 或更高版本的 PC

8.3.2　摘要

图 8-1 显示了硬件的设置方式。CY3280-21x34 UCC 套件通过 22x2_RA_Receptacle 连接至 CY3280-SLM。将 MiniProg1/Miniprog3 或 I2USB 桥接器连接到套件的 ISSP 插座上。该设置使用 MiniProg1/Miniprog3 进行编程，并使用 I2USB 桥接器/Miniprog3 将数据发送给 PC。它们通过 USB 电缆连接至 PC 上。

图 8-1. 硬件设置框图

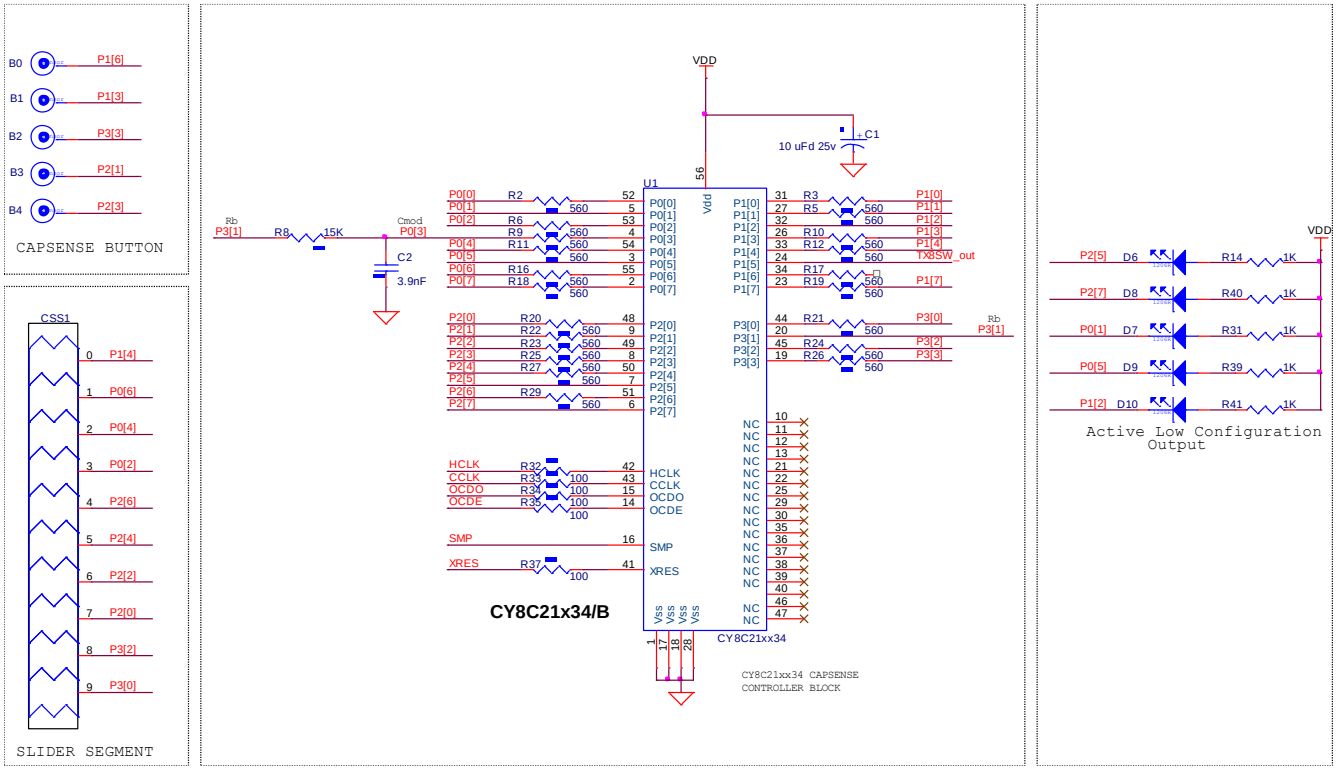


8.3.3 组装电路板

需要进行以下硬件连接：

- 连接 CY3280-SLM 子卡的插座 J1 和 CY3280-21x34 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）。
- 在插座 J7 上放置一个跳线器，以将 UCC 电路板的引脚 5 V 和 VCC_PROG 短接。该设置允许您通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置会将 CapSense 控制器的 XRES 引脚连接至 ISSP 连接器 J3 的引脚 3 上。
- 在插座 J2 上放置一个跳线器，将 CY3280-SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。通过该设置，可将 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 UCC 电路板的 ISSP 插座 J3 上。只有使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。通过 Bridge Control Panel 软件读取 CapSense 数据时，需要使用 I2USB 桥接器/MiniProg3 来代替 MiniProg1/MiniProg3 进行连接。
- 使用 USB A 转 Mini B 线缆来与 MiniProg1（或 I2USB 桥接器/MiniProg3）的另一端连接。

8.4 原理图



调制器电容（C_{MOD}）是在 P0[3]引脚上连接的大小为 2.2 nF 的电容。使用一个 560 Ω 的电阻将各个 CapSense 按键串联起来，以降低射频干扰。将 LED 配置为低电平有效，并将其串连一个 1 kΩ 的电阻。该原理图中共有五个 LED、五个按键以及一个包含 10 个可用段的滑条。

表 8-1. LED、按键和滑块的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

8.5 软件设置

8.5.1 需要的工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel（桥接控制面板）

8.5.2 用户模块列表和放置

下表列出了该代码示例中所使用的用户模块以及各个用户模块所占用的硬件资源。

用户模块	放置
CSD	CapSense 模块、定时器 1（默认）
EzI2C	I²C/SPI 模块

8.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例中所使用的用户模块的参数设置。此外，它还提供了一个全局资源列表。

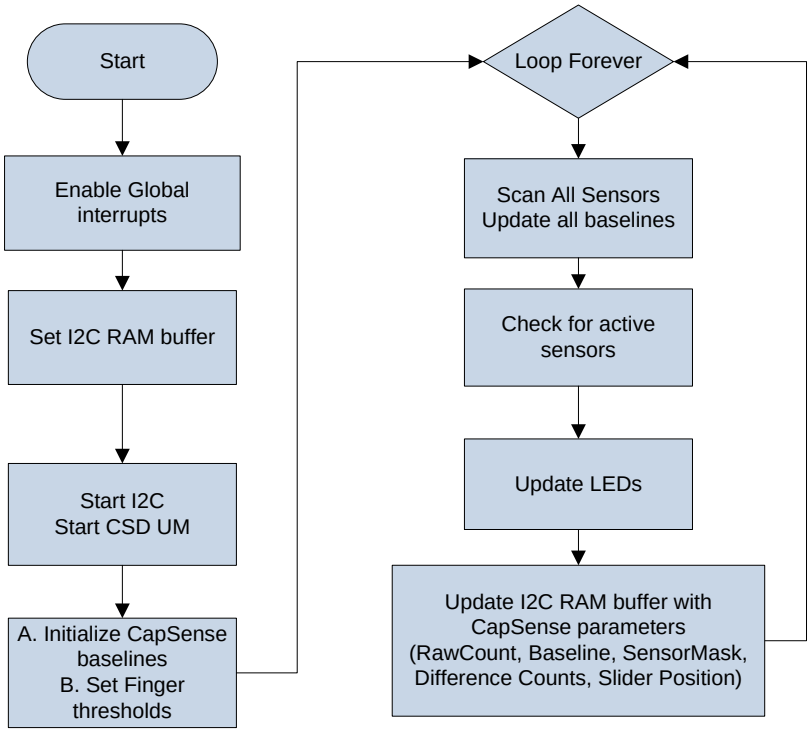
注意： 我们对该代码示例进行了调校，使之符合 CY3280-SLM 套件中 1.5 mm 厚的亚克力覆盖层。要想使用具有更厚覆盖层的代码示例，请按照该代码示例中所述的内容调校项目。

8.6 操作说明

复位时，固件将执行以下操作：

- 定义 MyI2C_Regs 结构体，以存储按键编号、原始计数、计数差值、基准线、中心位置以及给定按键编号相对应的 CapSense 按键的状态。
- 使能全局中断，然后启动 CSD UM 模块。
- 启动 EzI2Cs UM，然后将 MyI2C_Regs 结构体设置为 I²C RAM 缓冲区。
- 无限循环地执行下面操作：
  - 持续扫描所有传感器，并为 MyI2C_Regs 结构体更新原始计数、计数差值、基准线、滑条中心位置以及选定 CapSense 按键的状态。通过将按键编号写入到 EzI2C 从设备中 I²C 缓冲区的第一个字节内，I²C 可以请求某个特定按键的 CapSense 数据。
  - 当固件检测到按键触摸时，它会将相应的 LED 状态切换为“ON”。释放按键时，LED 状态会切换为“OFF”。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”，以表示触摸位置。

图 8-2. 代码示例 8 的功能流程



8.7 执行代码示例

使用项目来编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-21x34 套件指南中的第五章节。

1. 使用 MiniProg1/MiniProg3 或 CY3280-21x34 UCC 套件指南中介绍的某个源给该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。
   CY3280-SLM 电路板上相应的 LED 会点亮。
3. 触摸某个按键。
   CY3280-SLM 模块电路板上相应的 LED 会点亮。您可以同时触摸多个按键。也可以同时使用线性滑条和按键。

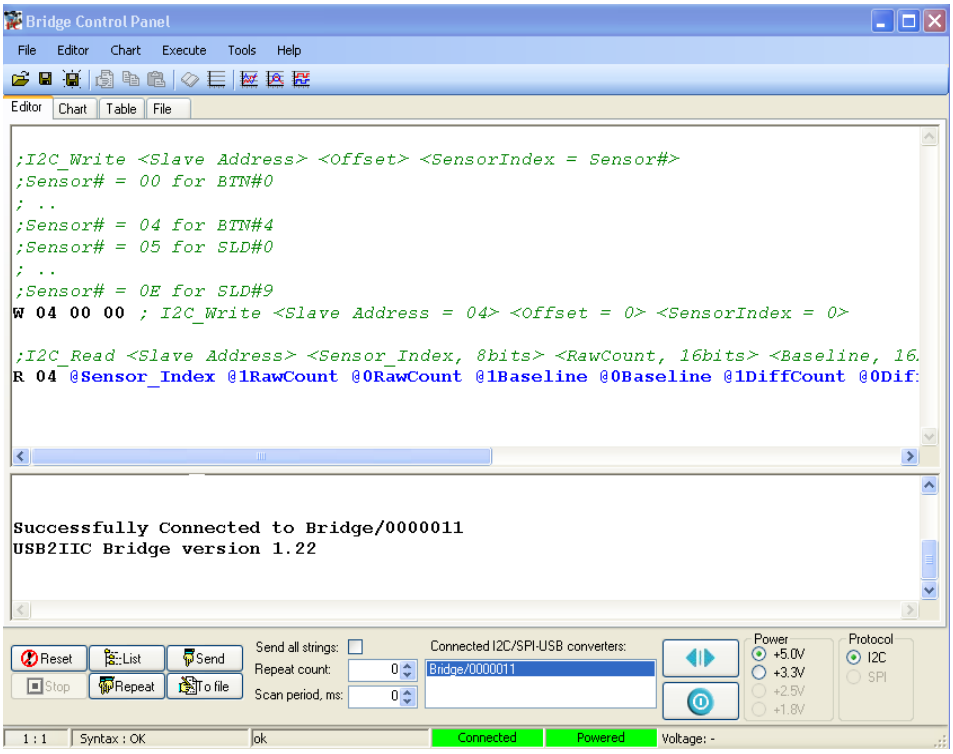
8.8 通过 I²C 读取 CapSense 数据

按照该程序执行代码示例，并读取 Bridge Control Panel 工具上的 CapSense 数据。更多有关桥接控制面板工具的详细信息，请参考 AN2397 — CapSense®数据查看工具。

8.8.1 加载桥接控制面板工具

1. 使用 I2USB 桥接器/MiniProg3 和 USB A 转 Mini B 线缆，将您的计算机连接至 CY3280-21x34 通用 CapSense 控制器电路板的 ISSP 连接器 J3 上。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。
   Bridge Control Panel 是在 PSoC Programmer 安装过程中安装的一个组件。
3. 从端口选择窗口中选择您需要的器件。
4. 为 CY3280-21x34 CapSense 控制器电路板提供 5 V 的电源。

5. 在 Bridge Control Panel 窗口中，依次选择 **File** > **Open**。从代码示例项目文件夹的 *BCP Configuration files* 文件夹中加载 `BCP.iic` 文件。
6. 依次选择 **Charts** > **Variable Settings**。从代码示例项目文件夹中的 *BCP Configuration files* 文件夹加载 `BCP.ini` 文件。单击 **OK** 以返回主窗口。



8.8.2 读取 BTN0 的原始计数、基准线和计数差值

1. 发送 I²C 写入指令 W 04 00 00 一次。
2. 按下 **Repeat** 按键，连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 转到 **Chart** 选项卡，以查看原始计数、基准线和 BTN0 的计数差值。

8.8.3 读取 SLD0 的原始计数、基准线、计数差值和滑条位置：

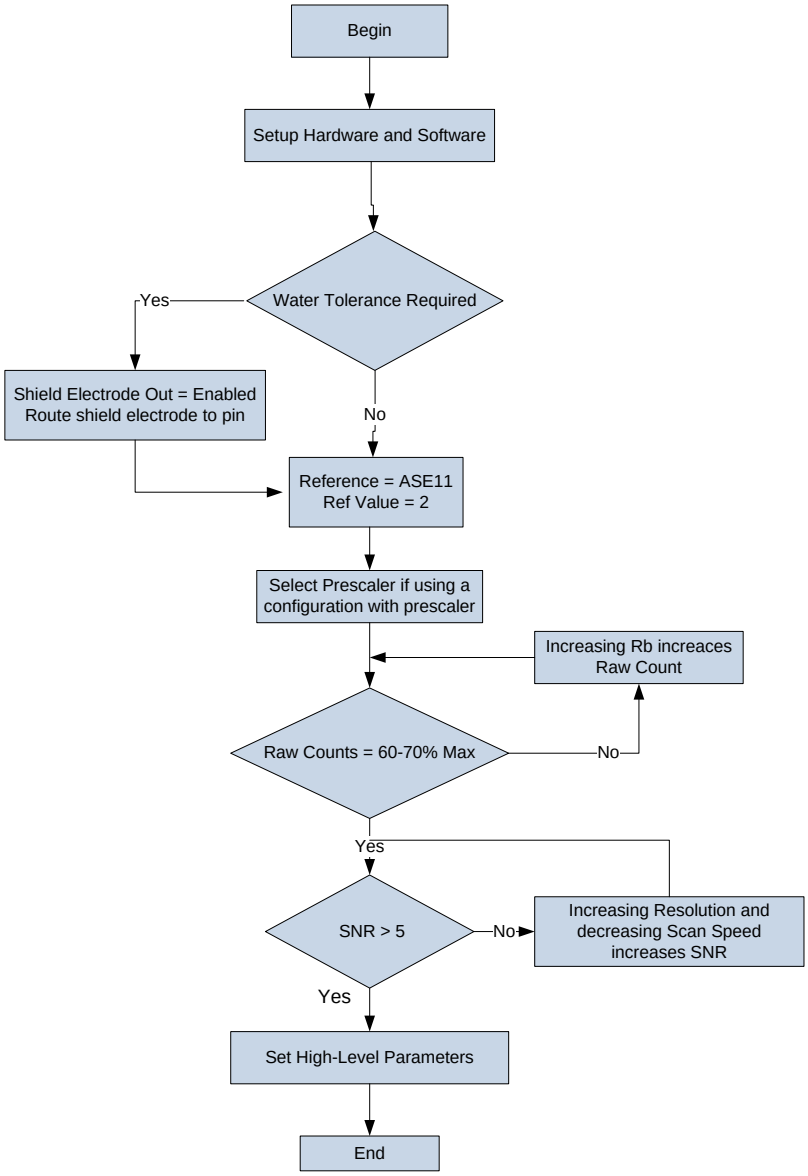
1. 发送 I²C 写指令 W 04 00 05 一次。
2. 按下 **Repeat** 按键，连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，以查看 SLD0 的原始计数、基准线、计数差值和滑条位置。

8.9 调校 CSD 用户模块

在本节中，我们按照下面流程图中的程序对用户模块的各个参数进行了调校。更多详细信息，请参见 CY8C21x34/B CapSense 设计指南。



8.9.1 配置 CSD UM 设置

使用 PSoC Designer 工具并按照以下步骤进行操作，实现对 CSD UM 参数设置进行的配置：

1. 该代码示例中使用了 Rb = 15 kΩ 和 C_{MOD} = 10 nF。CY3280-21x34 UCC 电路板使用了一个固定的 Rb 值。在自定义的电路板设计中，您可以通过修改 Rb 的值来调校用户模块。
   请参见 CY8C21x34/B CapSense 设计指南的第 4.2.3 章节，了解如何选择 Rb 值，从而使 CapSense 传感器的灵敏度最佳。
2. 不需要屏蔽电极。因此，可将 **ShieldElectrodeOut** 参数设置为 **None**。

表 8-2 列出了使用*代码示例 5（使用 CY8C21x34/B CapSense 控制器测量传感器的绝对电容）*中介绍的方法时所获得的传感器 C_P。由于我们示例中 CP 的最大值为 16 pF，因此可将预分频器设置为 4。

表 8-2. 所有传感器的 C_P

传感器	引脚	C_P
BTN0	P1[6]	16 pF
BTN1	P1[3]	14 pF
BTN2	P3[3]	13 pF
BTN3	P2[1]	12 pF
BTN4	P2[3]	11 pF
SLD0	P1[4]	14 pF
SLD1	P0[6]	15 pF
SLD2	P0[4]	14 pF
SLD3	P0[2]	14 pF
SLD4	P2[6]	13 pF
SLD5	P2[4]	12 pF
SLD6	P2[2]	12 pF
SLD7	P2[0]	12 pF
SLD8	P3[2]	11 pF
SLD9	P3[0]	11 pF

3. 将 **Precharge Source** 设置为 **PRS**。
   PRS 源能够实现扩频的操作，并确保对外部噪声源提供良好的免疫性。
4. 根据所有传感器的最大 C_P 值来设置 **Prescaler**（预分频器）。由于我们的示例中，最大 C_P 值为 16 pF，因此可将预分频器设置为 4。
   更多有关信息，请参见 CSD UM 数据手册中的“基于预充电源、IMO 和 CP 的预分频器设置”列表。
5. 如果一个项目使用了不同 C_P 值的传感器，那么请根据 C_P 值最大的传感器设置**分辨率**。在我们的示例中，最大 C_P 值为 16 pF，且 Cf 大于 0.2 pF。参考 CSD UM 数据手册中的图“基于覆盖层厚度和圆形传感器直径的手指电容（CF）”和表“基于手指电容和 CP 的分辨率设置”，将分辨率设置为 13。
6. 选择 **Comparator**（比较器），可供参考的有 ASE11。
   在大多数场合下，建议选择 ASE11。
7. 将 **Scan speed** 设置为 **Normal**。

表 8-3 列出了 **Bridge Control Panel**（桥接控制面板）上观察到的不同传感器的噪声（即原始计数在无手指触摸时的变量）和信号（手指响应）的值。

表 8-3. 所有传感器的信号

传感器	平均原始计数（无手指触摸）	信号	噪声	SNR（取最接近的整数值）
BTN0	4451	179	20	9
BTN1	3921	182	20	9
BTN2	3419	184	21	9
BTN3	3186	179	20	9
BTN4	2652	191	21	9
SLD0	3790	100	20	5
SLD1	4432	101	20	5
SLD2	4026	101	19	5
SLD3	4011	103	19	5
SLD4	3739	107	20	5
SLD5	3456	104	20	5
SLD6	3310	110	21	5
SLD7	3158	99	19	5
SLD8	2900	101	19	5
SLD9	2688	115	20	5

8. 由于所有传感器的 SNR 均大于 5:1，且扫描时间已经足够低，手指触摸能够使 LED 立即发光，因此您可以按照表 8-4 中的信息设置阈值参数。
   我们根据所有传感器间最低的信号值来配置阈值。因此，在我们的示例中信号值为 99。

   表 8-4. CSD UM 的高层参数

阈值参数	推荐值	数值	UM 参数设置
手指阈值	信号值的 75%	75.3	74
噪声阈值	信号值的 40%	39.6	40
基准线更新阈值	2*噪声阈值	80	80
负噪声阈值	噪声阈值	40	40
迟滞	信号值的 15%	14.9	15

9. 由于扫描时间满足要求，并且没有出现虚假的正极报告，因此可将 **Low Baseline Reset（低基准线复位）**参数设置为 10。将 **Debounce（去抖动）**值设置为 3。

代码示例 9　使用 CY8C20x34 CapSense 控制器调校 CSA_EMC

9.1　项目名称

CE_9_TuningCSAEMC_withCY8C20x34

9.2　概述

该代码示例说明了如何为 CY8C20x34 器件调校 CSA_EMC 用户模块（UM）。要调校 CapSense UM，您必须能够查看图形格式的 CapSense 数据。该代码示例使用 EzI2C UM 并通过 I²C 总线和 I²C 将所需的 CapSense 参数传输给 USB 桥接器。它使用了 PC 上的 Bridge Control Panel 软件来查看数据。

9.3　硬件设置

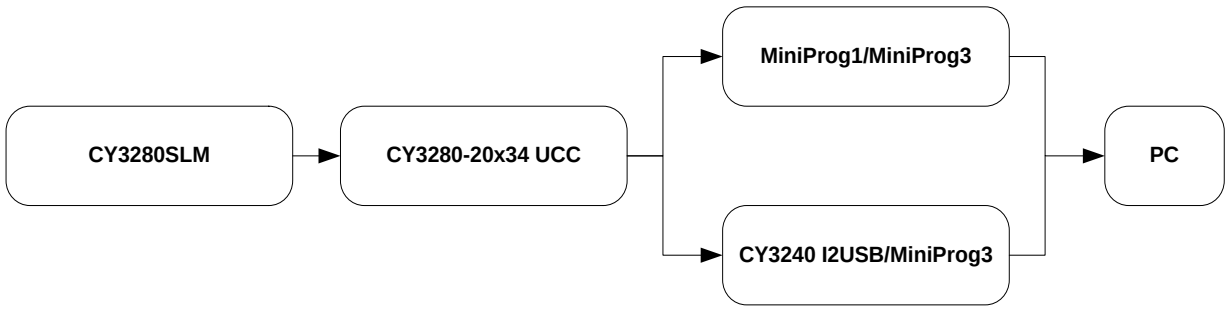
9.3.1　相关硬件

- CY3280-20x34 通用 CapSense 控制器电路板
- CY3280-SLM 通用 CapSense 线性滑条模块
- CY3240-I2USB 桥接器或 CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 带有 USB 端口的 PC

9.3.2　摘要

图 9-1 显示的是硬件设置方式。CY3280-20x34 UCC 套件通过 22x2_RA_Receptacle 连接到 CY3280-SLM。将 MiniProg1/Miniprog3 或 I2USB 桥接器连接到套件的 ISSP 插座上。该设置使用 MiniProg1/Miniprog3 进行编程，并使用 I2USB 桥接器/Miniprog3 将数据发送给 PC。使用 USB 线缆将它们连接到 PC 上。

图 9-1. 硬件设置框图

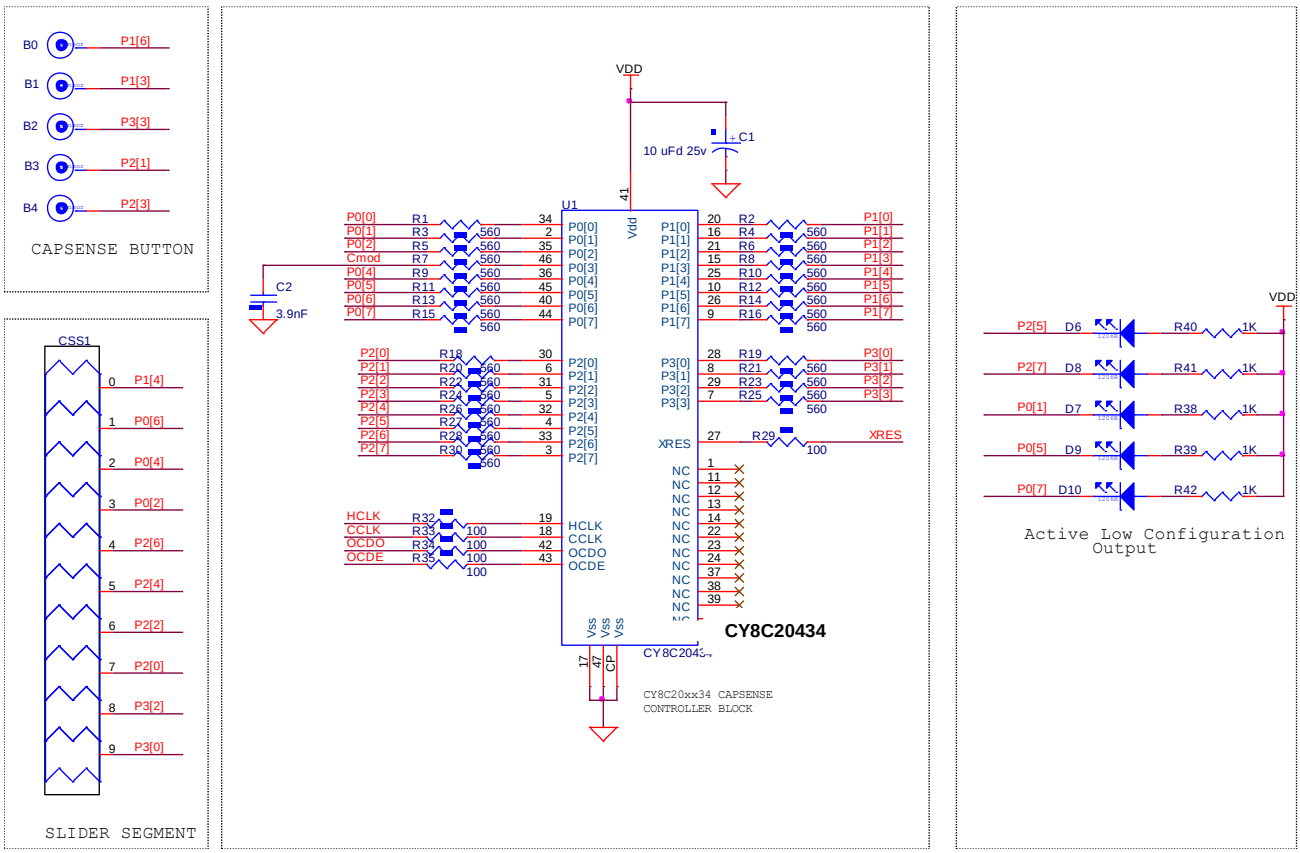


9.3.3 组装电路板

需要进行以下硬件连接：

- 将 CY3280-SLM 子卡的插座 J1 连接到 CY3280-20x34 UCC 电路板的 22x2_RA_Receptacle（连接器 P2）上。
- 在插座 J1 上放置一个跳线器，以将 UCC 电路板的引脚 V_{CC} 和 5 V 短接。该设置允许您通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置将 CapSense 控制器的 XRES 引脚连接到 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上放置一个跳线器，将 CY3280-SLM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。通过该设置，可将 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 UCC 电路板的 ISSP 插座 J3 上。只有使用十六进制文件产生的代码来编程 UCC 时，才需要该连接。使用 Bridge Control Panel 软件读取 CapSense 数据时，需要使用 I2USB 桥接器/MiniProg3 代替 MiniProg1/MiniProg3 进行连接。
- 使用 USB A 转 Mini B 线缆将 MiniProg1/Miniprog3（或 I2USB 桥接器）的另一端连接到 PC 上。

9.4 原理图



调制器电容器（C_{MOD}）是一个连接至 P0[3]的 2.2 nF 大小的电容。使用一个 560 Ω 的电阻将各个 CapSense 按键串联起来，以降低射频干扰。将 LED 配置为低电平有效，并将其串连一个 1 kΩ 的电阻。该原理图中共有五个 LED、五个按键以及一个包含 10 个可用段的滑条。

表 9-1. LED、按键和滑块的引脚分配

LED	按键	滑条段
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

9.5 软件设置

9.5.1 需要的工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）
- Bridge Control Panel（桥接控制面板）

9.5.2 用户模块列表

下表列出了该代码示例中所使用的用户模块以及各个用户模块所占用的硬件资源。

用户模块	硬件资源
CSA_EMC	CapSense 模块
EzI2C	I^2C 模块

9.5.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例中所使用的用户模块的参数设置。此外，它还提供了一个全局资源列表。

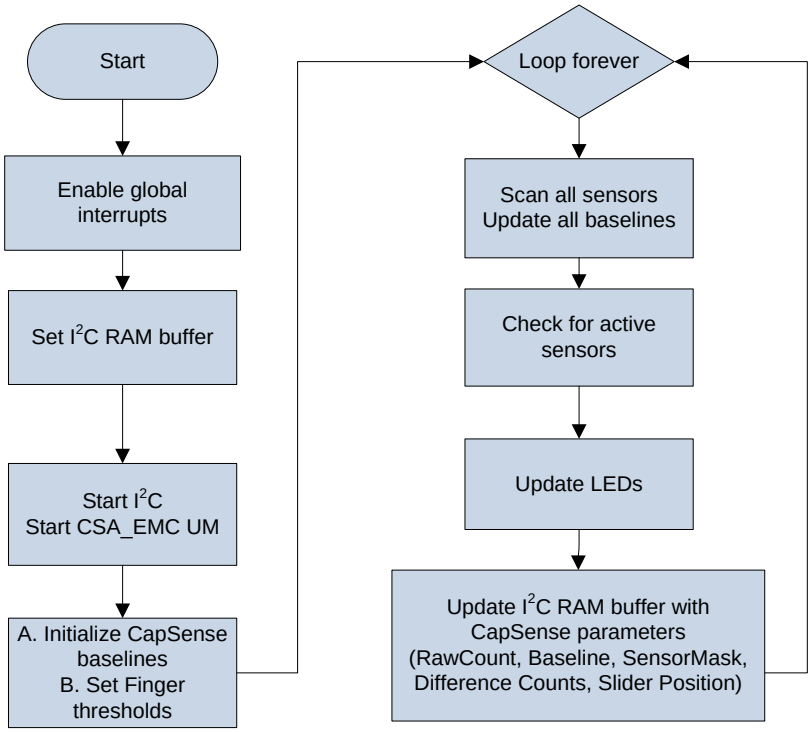
注意： 我们对该代码示例进行了调校，使之符合 CY3280-SLM 套件中 1.5 mm 厚的亚克力覆盖层。要想使用具有更厚覆盖层的代码示例，请按照该代码示例中所述的内容调校项目。

9.6 操作说明

复位时，固件将执行以下操作：

- 定义一个结构体 MyI2C_Regs，以存储按键编号、原始计数、计数差值、基准线、中心位置以及给定按键编号相对应的 CapSense 按键状态。
- 启用全局中断，然后启动 CSA_EMC UM。
- 启动 EzI2Cs UM，然后将 MyI2C_Regs 结构体设置为 I^2C RAM 缓冲区。
- 无限循环地执行下面操作：
  - 持续扫描所有传感器，并为 MyI2C_Regs 结构体更新原始计数、计数差值、基准线、滑条中心位置以及选定 CapSense 按键的状态。通过将按键编号写入到 EzI2C 从设备中 I^2C 缓冲区的第一个字节内，I^2C 可以请求某个特定按键的 CapSense 数据。
  - 当固件检测到按键触摸时，它会将相应的 LED 状态切换为“ON”。释放按键时，LED 状态会切换为“OFF”。
  - 触摸滑条时，固件会将 LED 的状态切换为“ON”，以表示触摸位置。

图 9-2. 代码示例 9 的功能流程



9.7 执行代码示例

使用项目来编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参考 CY3280-20x34 套件指南中的第五章节。

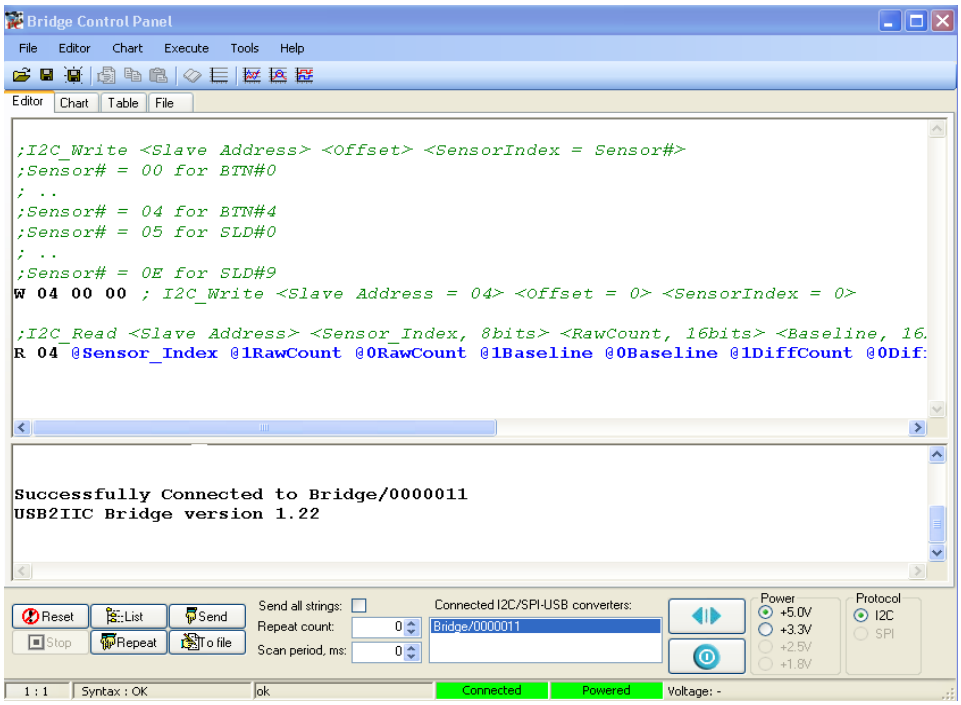
1. 使用 MiniProg1/MiniProg3 或 CY3280-20x34 UCC 套件指南中所介绍的某个源给该电路板提供 5 V 的电源。
2. 触摸 CY3280-SLM 模块电路板上的线性滑条。
   CY3280-SLM 电路板上相应的 LED 会点亮。
3. 触摸某个按键。
   CY3280-SLM 模块电路板上相应的 LED 会点亮。您可以同时触摸多个按键。也可以同时使用线性滑条和按键。

9.8 通过 I²C 读取 CapSense 数据

按照该程序执行代码示例并读取 Bridge Control Panel 工具上的 CapSense 数据。更多有关桥接控制面板工具的详细信息，请参考 AN2397，“CapSense®数据查看工具”。

1. 通过使用 I2USB 桥接器/MiniProg3 和 USB A 转 Mini B 线缆，可以将您的计算机连接到 CY3280-20x34 通用 CapSense 控制器电路板的 ISSP 连接器 J3 上。
2. 在电脑桌面上，依次选择 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *（版本）* > **Bridge Control Panel** *（版本）*。
   Bridge Control Panel 是在 PSoC Programmer 安装过程中安装的一个组件。
3. 从端口选择窗口中选择您需要的器件。
4. 为 CY3280-20x34 CapSense 控制器电路板提供 5 V 的电源。
5. 在 Bridge Control Panel 窗口中，依次选择 **File** > **Open**。从代码示例项目文件夹中的 *BCP Configuration files* 文件夹加载 *BCP.iic* 文件。

6. 依次选择 **Charts** > **Variable Settings**。从代码示例项目文件夹中的 *BCP Configuration files* 文件夹加载 BCP.ini 文件。单击 **OK**，返回主窗口。



9.8.1 读取 BTN0 的原始计数、基准线和计数差值

1. 发送 I²C 写指令 W 04 00 00 一次。
2. 按下 **Repeat** 按键，连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，以查看 BTN0 的原始计数、基准线和计数差值。

9.8.2 读取 SLD0 的原始计数、基准线、计数差值和滑条位置

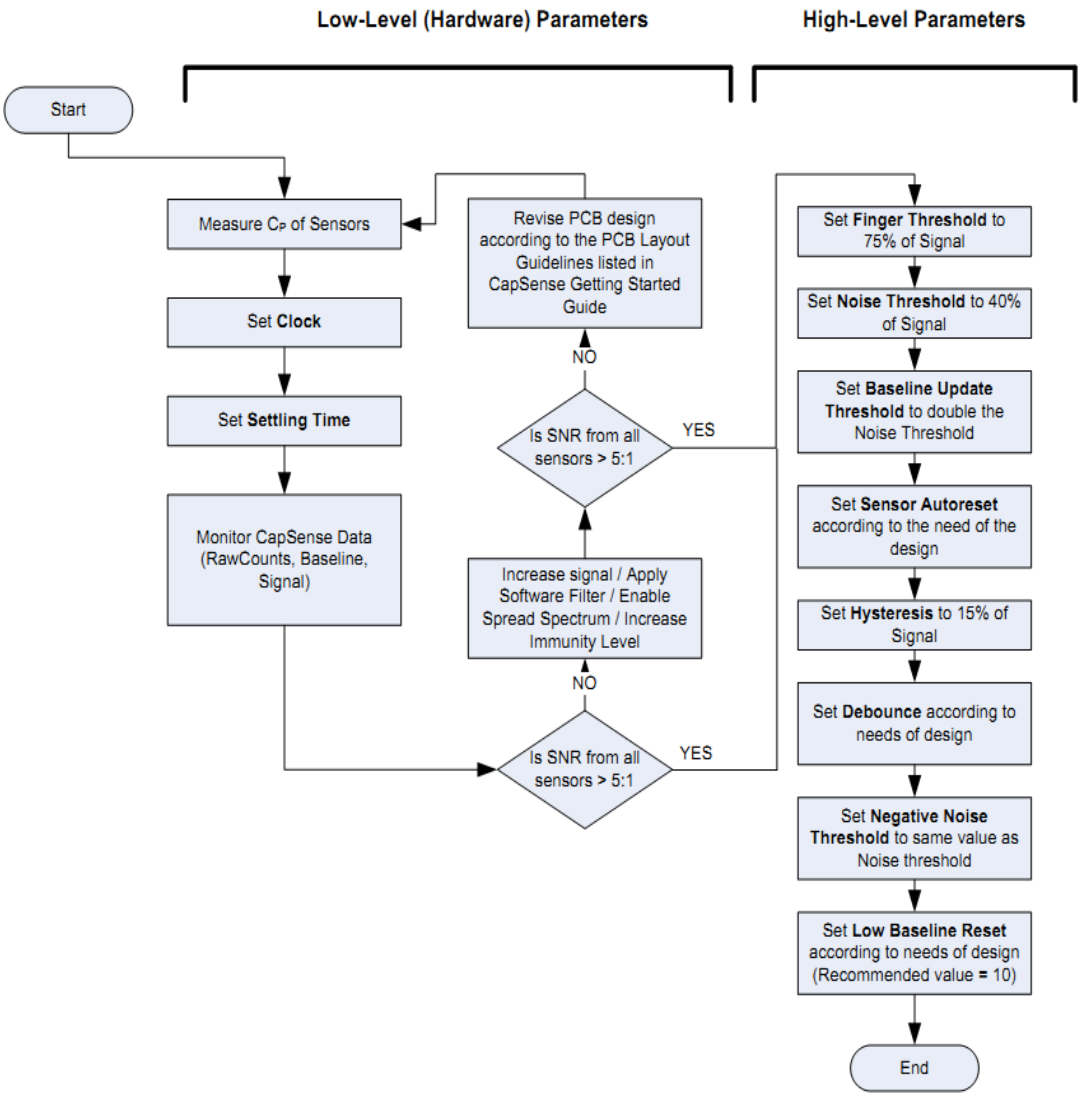
1. 发送 I²C 写指令 W 04 00 05 一次。
2. 按下 **Repeat** 按键，连续发送以下 I²C 读指令：

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount
@0DiffCount @ButtonStatus @SliderPosition
```

3. 点击 **Chart** 选项卡，查看 SLD0 的原始计数、基准线、计数差值和滑条位置。

9.9 调校 CSA_EMC 用户模块

在本节中，我们将采用以下流程图中给出的步骤来调试用户模块参数。更多详细信息，请参考 CY8C20x34 CapSense 设计指南。



9.9.1 配置 CSD UM 设置

使用 PSoC Designer 工具并按照以下步骤进行操作来配置 CSD UM 参数设置，具体如下：

1. 在 PSoC Designer 工具的“Global Resources”（全局资源）窗口中，将 CPU_CLK 设为 SysClk/2。根据您的设计需求，在“Global Resources”窗口中配置其他参数。
2. 表 9-2 列出了使用*代码示例 6（使用 CY8C20x34 CapSense 控制器测量传感器的绝对电容）*中的方法所获得的所有传感器 C_P。由于我们示例中最大的 C_P 值为 15 pF，且 IMO 为 12 MHz，因此可将“Clock”（时钟）频率设置为 IMO/4。

   更多有关 IMO 的信息，请参见 CY8C20x34 CapSense 设计指南中表 15 所述的内容。

表 9-2. 所有传感器的 C_P 值

传感器	引脚	C_P
BTN0	P1[6]	15 pF
BTN1	P1[3]	13 pF
BTN2	P3[3]	12 pF
BTN3	P2[1]	11 pF
BTN4	P2[3]	9 pF
SLD0	P1[4]	14 pF
SLD1	P0[6]	14 pF
SLD2	P0[4]	13 pF
SLD3	P0[2]	12 pF
SLD4	P2[6]	12 pF
SLD5	P2[4]	11 pF
SLD6	P2[2]	10 pF
SLD7	P2[0]	10 pF
SLD8	P3[2]	10 pF
SLD9	P3[0]	9 pF

3. 使用下面公式计算“Settling Time”（建立时间）参数：

$$\text{Settling Time} = \frac{5 \times C_{INT}}{\left(\text{Clock} \times C_P \times 25\left(\frac{1}{F_{CPU}}\right)\right)}$$

C_{INT} = 3.9 nF

Clock = IMO/4 = 3 MHz

C_P = 15 pF

F_{CPU} = SysClk/2 = 12 MHz/2 = 6 MHz （在该示例中，将 SysClk 设置为 12 MHz）

将这些值代入上面的计算公式中，可得出：Settling Time = 80

对于新建设置，使用 Bridge Control Panel 来测其 SNR 值。表 9-3 列出了在该代码示例中观察到的值。

表 9-3. 观察到的所有传感器的信号

传感器编号	平均原始计数（无手指触摸）	噪声	信号	SNR（取最接近的整数值）
BTN0	2000	14	130	9
BTN1	2140	15	140	9
BTN2	2210	18	140	8
BTN3	2230	16	140	9
BTN4	1930	20	150	7
SLD0	1985	11	100	9
SLD1	2310	14	95	7

传感器编号	平均原始计数（无手指触摸）	噪声	信号	SNR（取最接近的整数值）
SLD2	2075	14	105	7
SLD3	2160	13	95	7
SLD4	2500	14	100	7
SLD5	2370	15	105	7
SLD6	2200	16	105	7
SLD7	2180	14	110	8
SLD8	1960	14	110	8
SLD9	2050	15	110	7

4. 由于最低 SNR = 7（凑整到最接近的整数）大于 5:1，因此可按照表 9-4 设置各个高级参数。

   表 9-4. UM 的高层参数

阈值参数	推荐值	数值	UM 参数设置
手指阈值	信号值的 75%	71.25	71
噪声阈值	信号值的 40%	38	38
基准线更新阈值	2*噪声阈值	76	76
负噪声阈值	噪声阈值	38	38
迟滞	信号值的 15%	14.25	14

5. 其他参数保留为它们的默认值。

代码示例 10 通过使用 CY8C20xx6A 上的 SmartSense，可对一个 CapSense 按键进行功耗优化

10.1 项目名称

CE_10_PowerOptimization_withCY8C20xx6A

10.2 概况

该代码示例的目标是使用 SmartSense 将一个按键的功耗降到 50 µA 以下。该代码示例采用了 SmartSense 和多个 16 位定时器 UM 模块。该器件在特定时间段内处于睡眠模式，以减少平均电流消耗。我们将 16 位定时器作为扫描定时器使用，以在所配置的睡眠时间后唤醒器件。

10.3 硬件设置

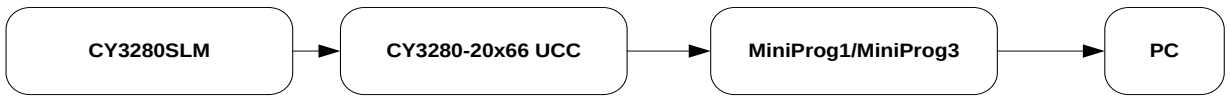
10.3.1 相关硬件

- CY3280-20x66 通用 CapSense 控制器电路板
- CY3280-BSM 通用 CapSense 简单按键模块
- CY3217-MiniProg1 编程器套件或 CY8CKIT-002 Miniprog3
- USB A 转 Mini B 线缆
- 电流表
- 电源

10.3.2 摘要

图 10-1 显示的是硬件设置方式。CY3280-20x66 UCC 套件通过 22x2_RA_Receptacle 连接至 CY3280-BSM 模块。该设置使用了连接到套件 ISSP 插座的 MiniProg1/Miniprog3 进行编程。

图 10-1. 硬件设置框图

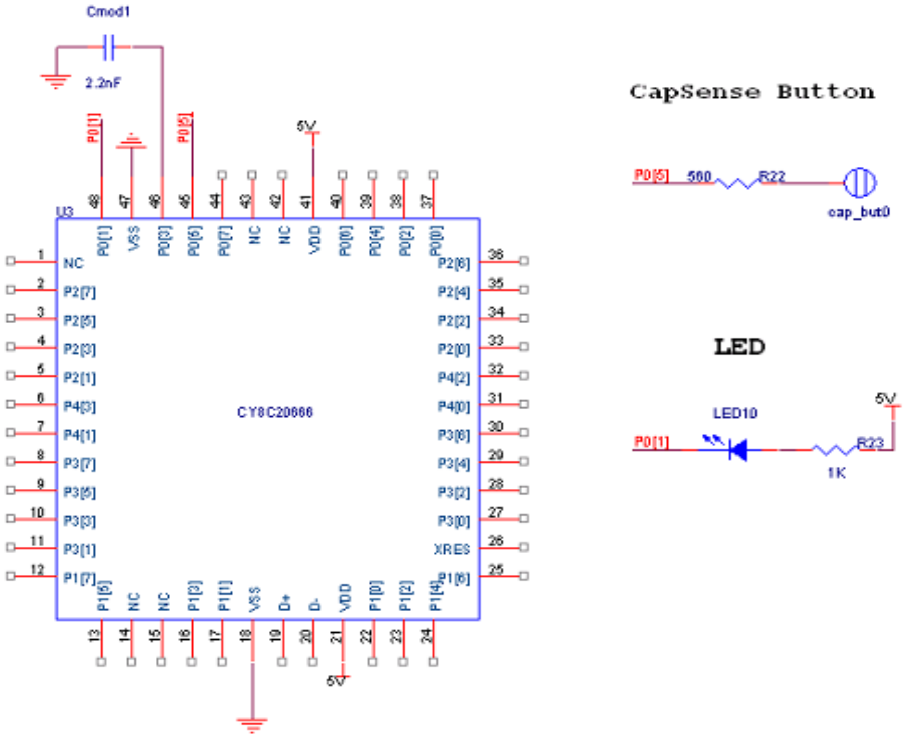


10.4 组装电路板

需要进行以下硬件连接：

- 将 CY3280-BSM 子卡的插座 J1 连接至 CY3280-20x66 UCC 电路板上的 22x2_RA_Receptacle（连接器 P2）。
- 在插座 J7 上放置一个跳线器，将 UCC 电路板上的引脚 V_{CC} 和 V_{CC_PROG} 短接。该设置允许您通过 ISSP 连接器给 CapSense 控制器供电。
- 在插座 J4 上放置一个跳线器，将 UCC 电路板的引脚 XRES 和 XRES/INT（引脚 1 和 2）短接。该设置将 CapSense 控制器的 XRES 引脚连接到 ISSP 连接器 J3 的引脚 3。
- 在插座 J2 上放置一个跳线器，将 CY3280-BSM 电路板的引脚 GND 和 SHIELD（引脚 2 和 3）短接。这样会使 CY3280-SLM 电路板上的网格接地。
- 将 MiniProg1/MiniProg3 连接到 CY3280-20x66 UCC 的 J3 插座上。

10.5 原理图



注意： 测量器件的功耗时，需要移除 CY3280-20x66 通用 CapSense 控制器电路板上的 LED D1 和 D2。带有 SmartSense UM 的器件的功耗取决于系统的 SNR 值。当电路板布局设计遵循 CapSense 入门指南中的指导时，可以使功耗最低。

表 10-1. 代码示例 10 的引脚分配

组件	引脚
按键 0	P0[5]
LED	P0[1]
C_{MOD}	P0[3]

10.6 软件设置

10.6.1 所需的工具

- PSoC® Designer（版本 5.2 或更高版本）
- PSoC Programmer（版本 3.13 或更高版本）

10.6.2 用户模块列表

下表列出了该代码示例中所使用的用户模块以及各个用户模块所占用的硬件资源。

用户模块	硬件资源	用户模块名称
SmartSense	CapSense 模块、定时器 1（默认）	SmartSense
定时器 16	定时器 0	Scan_Timer

10.6.3 用户模块参数、全局资源

项目中的 ReadMe.txt 文件说明了该代码示例中所使用的用户模块的参数设置。此外，它还提供了一个全局资源列表。

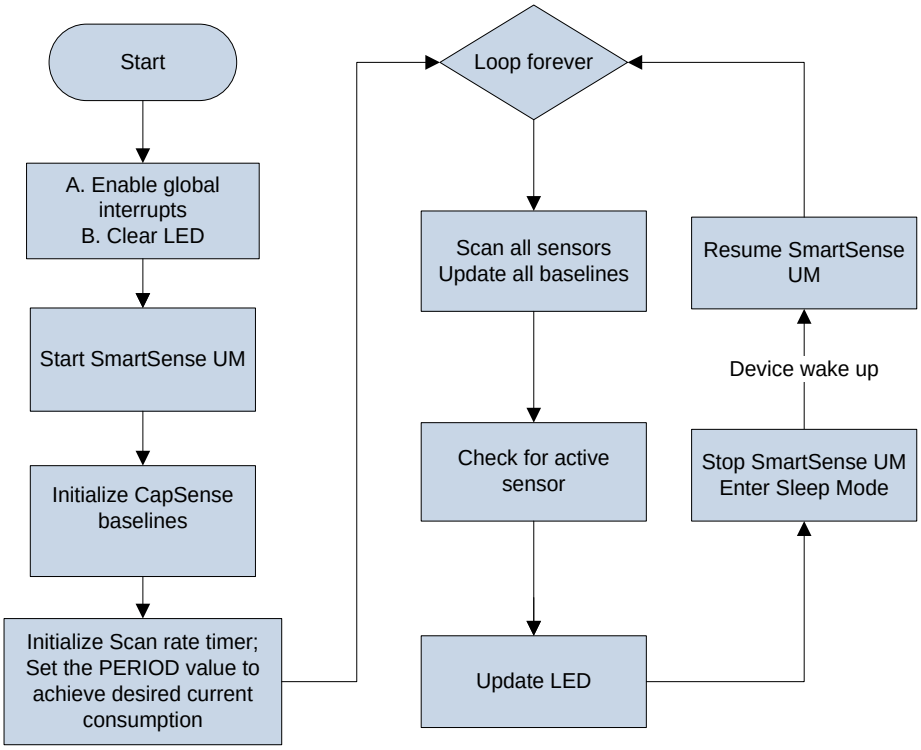
10.7 操作说明

复位时，程序会将器件配置中的所有硬件设置加载到器件中，并执行 *main.c*。然后，固件将执行以下操作：

- 初始化 SmartSense UM、ScanRate_Timer UM 和 LED。
- 使能全局中断，然后启动 SmartSense 和 ScanRate_Timer 用户模块。
- 当固件检测到按键触摸，它会将相应的 LED 状态切换为“ON”。释放按键时，LED 状态会切换为“OFF”。通过该操作，可验证该器件的功能。
- 该器件进入睡眠模式，但固件仍继续扫描传感器。该器件将在 ScanRate_Timer 中断时从睡眠模式中唤醒。

  想要更改扫描速率，需要更改宏 SCANRATE_TIMER_VALUE。该操作可以控制 ScanRate_Timer 的时间。降低扫描速率时，响应速度会提高。

图 10-2. 代码示例 10 的功能流程



10.8 执行代码示例

使用项目来编程电路板，然后按照该程序执行代码示例。有关如何编程 UCC 电路板的详细信息，请参见 CY3280-20x66 套件指南中的第五章节。

1. 要想更改扫描速率，请编辑代码示例中的宏 SCANRATE_TIMER_VALUE。这将更改 ScanRate_Timer 用户模块的参数周期。扫描速率取决于以下关系：

   扫描速率 = 周期/时钟选择频率

   其中，周期 = SCANRATE_TIMER_VALUE

2. 使用电流表测量扫描速率分别为 250、500、750 和 1000 ms 时的电流。以下图形和表格显示了不同工作电压和不同扫描速率下的功耗。

图 10-3. 不同扫描速率和电压时的平均电流

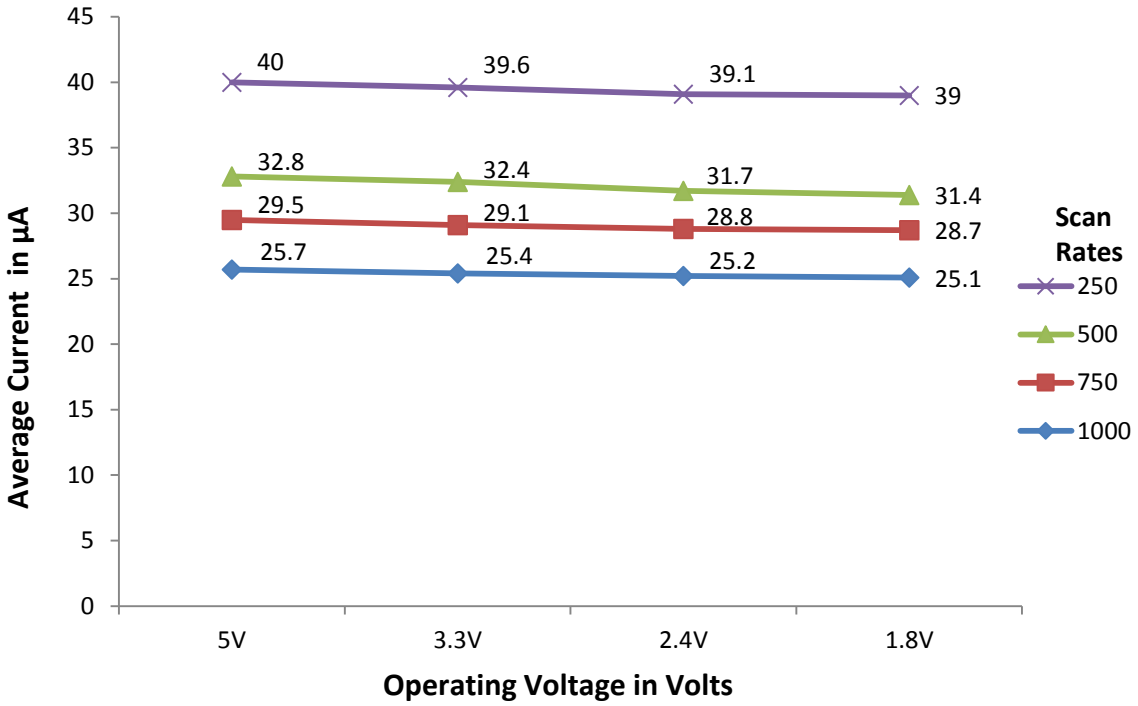


表 10-2. 不同扫描速率和电压时的平均电流

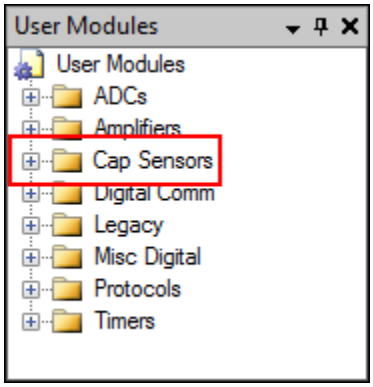
扫描速率（ms）	平均电流（µA）			
	5 V	3.3 V	2.4 V	1.8 V
1000	25.7	25.4	25.2	25.1
750	29.5	29.1	28.8	28.7
500	32.8	32.4	31.7	31.4
250	40	39.6	39.1	39

附录

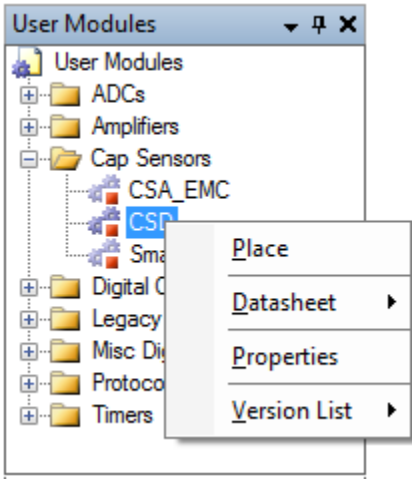
配置 CSD 用户模块

要修改该设计指南中包含的项目，您必须能够配置 PSoC®Designer 软件附带的 CapSense 用户模块（UM）。通过配置 UM 模块，您可以将某个项目从一个器件迁移到其他器件中。所有 CapSense UM 都具有类似的配置流程。在下面的示例中，我们使用一个软件向导来配置 CSD：

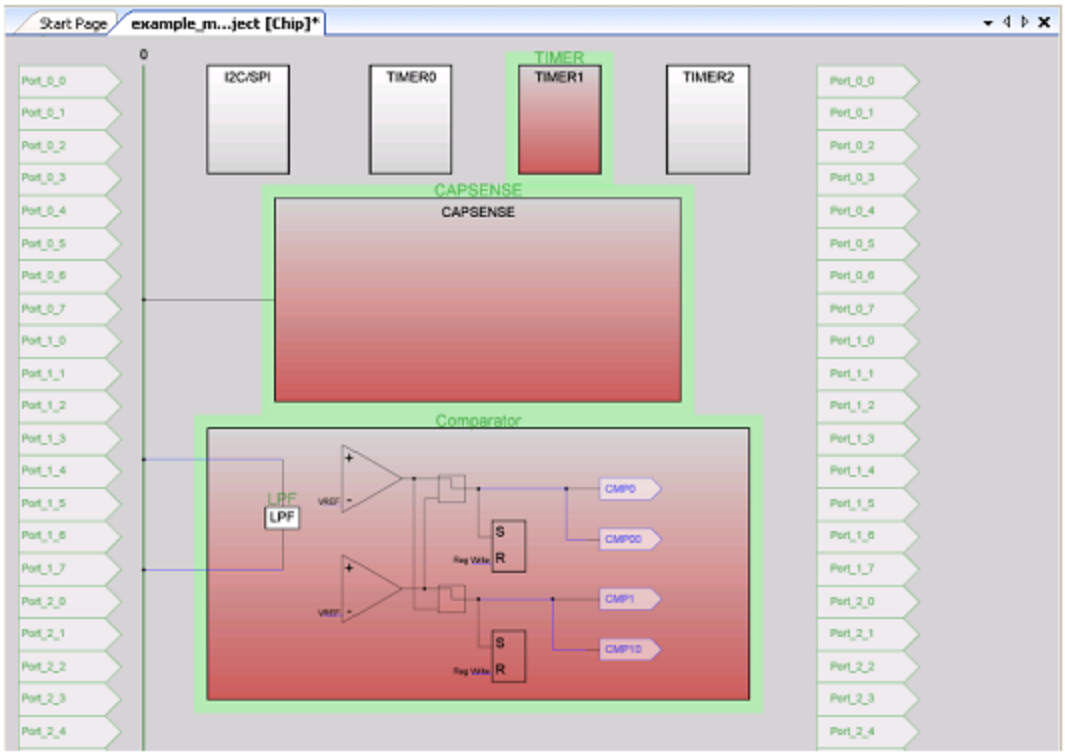
1. 在 PSoC Designer 的 **User Modules** 窗口中，展开 **Cap Sensors** 文件夹。



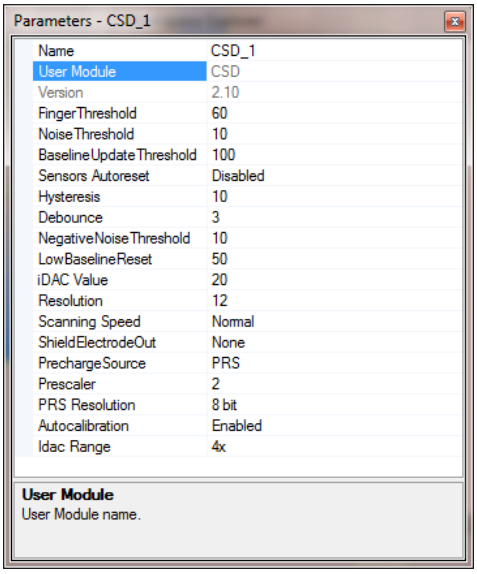
2. 在 **Cap Sensors** 文件夹中，右击 **CSD**，然后选择 **Place** 项。



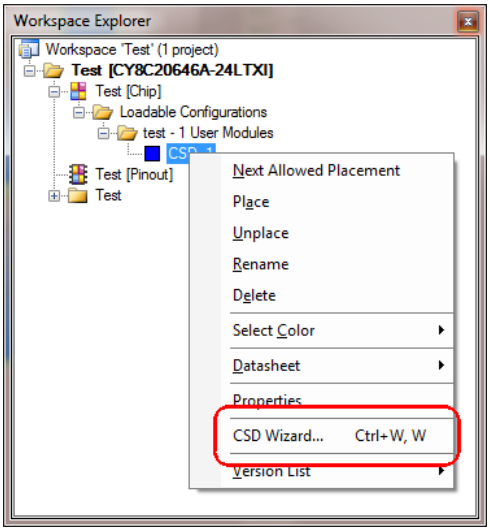
使用 Designer 工具将用户模块放置在芯片编辑器可用的模块中。



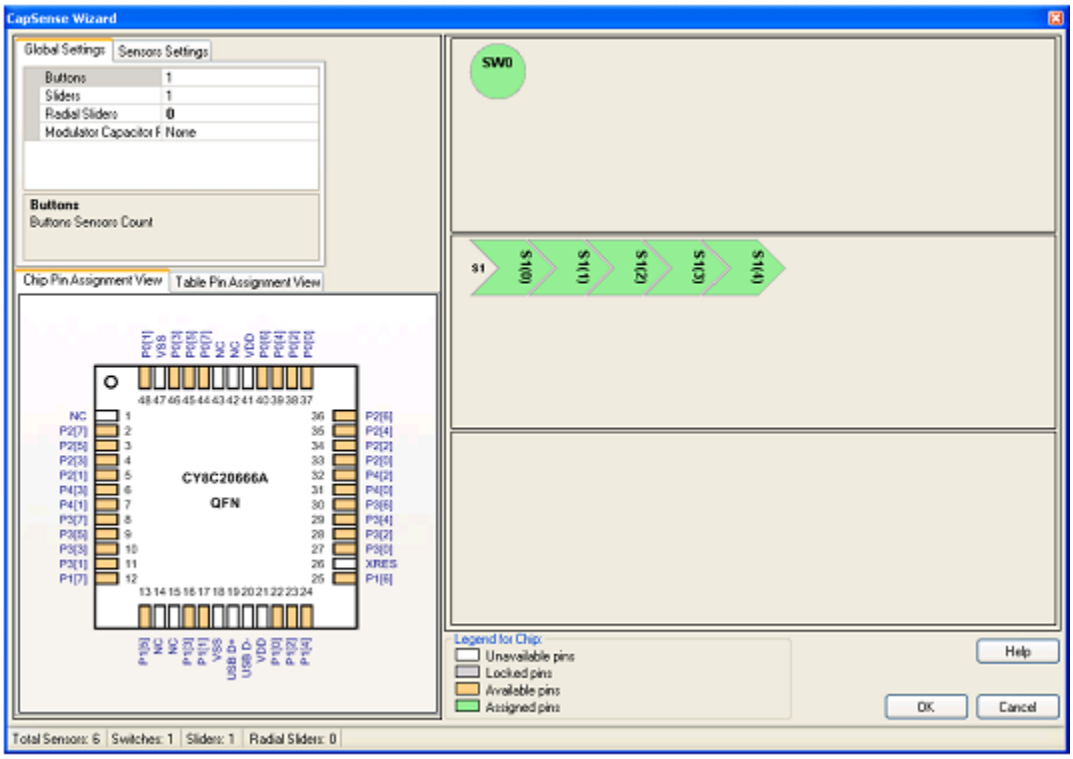
3. 配置 CSD_1 属性，如以下屏幕截图所示：



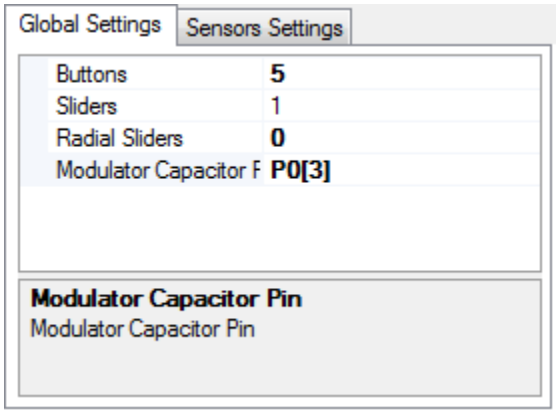
4. 右击 CSD 用户模块，然后选择 **CSD Wizard** 项。
   我们使用该向导将引脚分配给各个传感器。



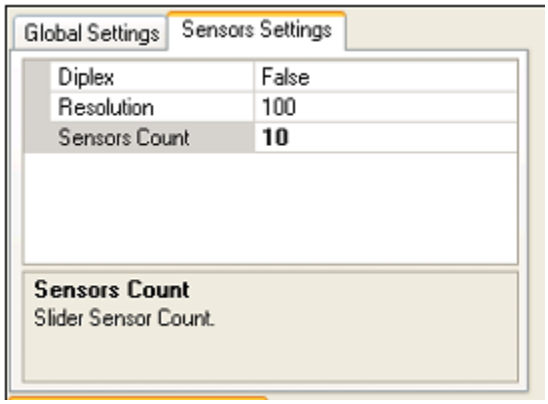
这时，将出现 CapSense Wizard 窗口。



5. 在 **CSD Wizard** 窗口中配置 **Global Settings**。
   该屏幕截图中的设置是一个包含五个按键和一个滑条的项目。使用 **Global Settings** 选项卡配置项目所需要的传感器数量。



6. 单击 **CSD Wizard** 中的滑条，查看传感器设置。配置 **Sensor Settings**，如下图所示。



7. 想要为特定的引脚分配传感器，请将传感器模块中的某个传感器拖放到“Pin Assignment”窗口所需要的引脚上。将 SW0 拖放到引脚 P1 [6]上。您可以按照表格引脚分配视图（图 10-5）或芯片引脚分配视图（图 10-4）中的方式进行分配传感器引脚。

图 10-4. 将传感器分配给引脚 — 芯片引脚分配视图

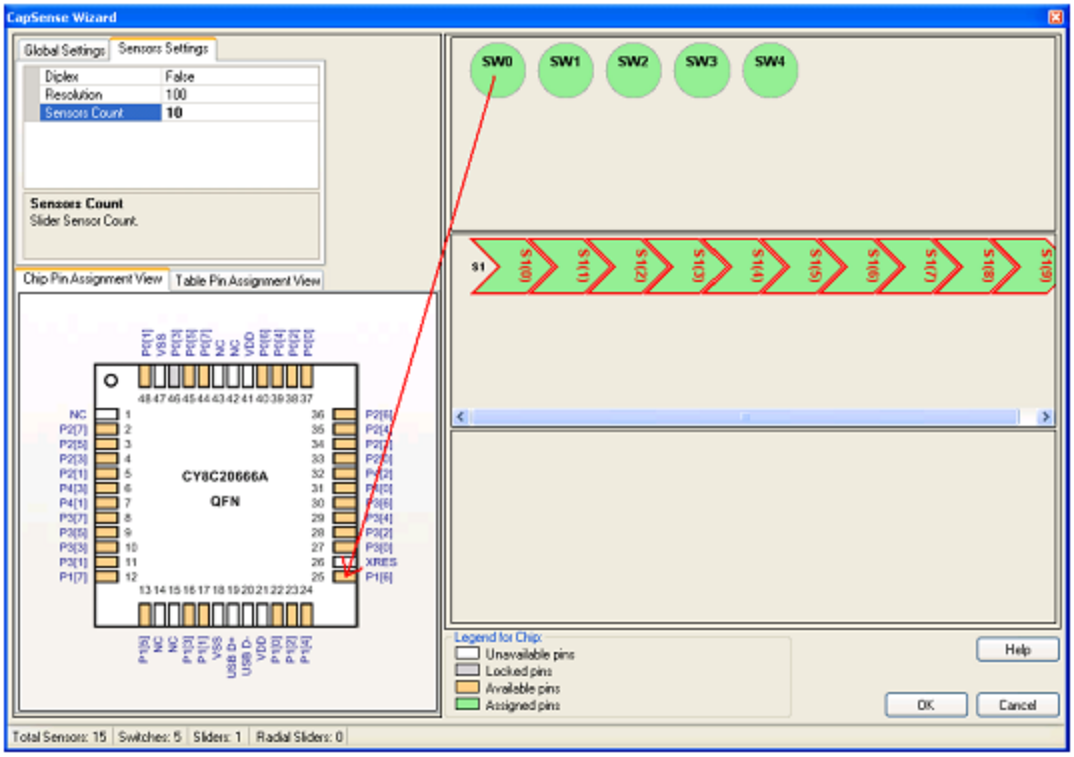
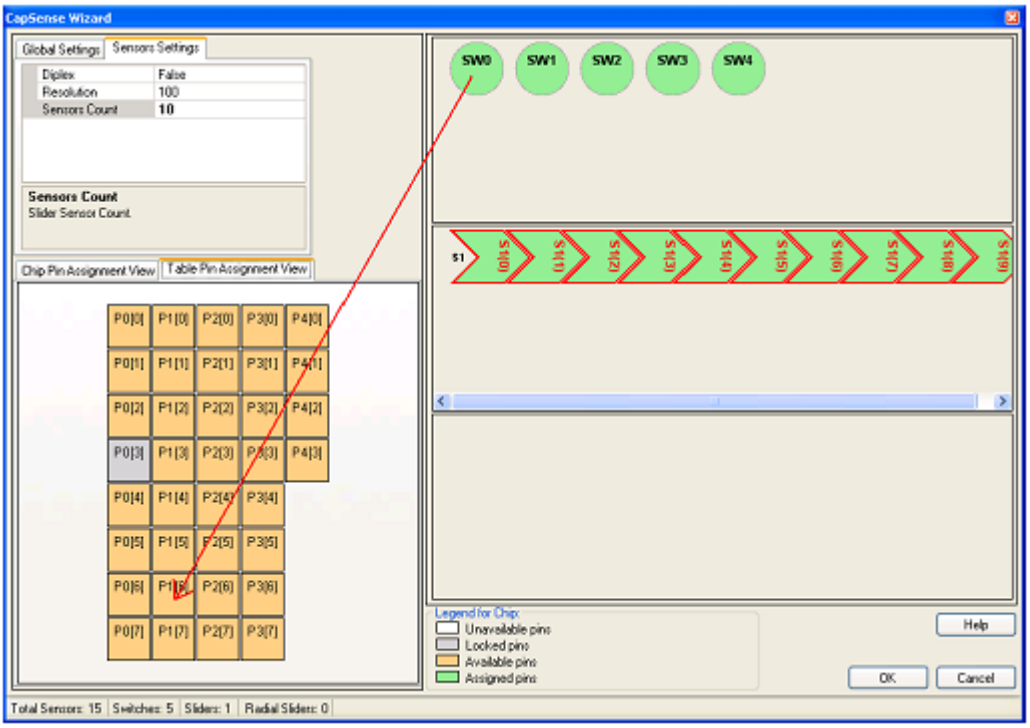


图 10-5. 将传感器分配给引脚 — 表格引脚分配视图



8. 同样，您也可以将其他传感器分配给各个器件引脚。点击 **OK** 以完成该操作。

将代码示例移植到其他 CapSense 器件中

要想为您的终端应用选择正确的器件，请参考 CapSense 入门指南。本文档所提供的所有代码示例都仅适用于特定的器件（目标套件），如表 1 所示。要想将这些代码示例使用于其他 CapSense 器件，需要将该项目移植（或迁移）到所需要的器件内。

移植之前，需要了解每个 CapSense 器件可支持的 CapSense 用户模块。CapSense 入门指南中的*表 5-1* 显示的是 CapSense 产品系列与它所支持的 CapSense UM 的映射关系。

下面各节对器件间的项目移植流程进行了详细的说明。

SmartSense

SmartSense UM 受 CY8C20xx6A 和 CY8C21x34B 系列器件的支持。更多有关信息，请参考*代码示例 1：使用 CY8C20xx6A CapSense 控制器来实现基于 SmartSense 技术的按键和滑条*。目前，虽然我们还没有任何一个专用的套件使用 CY8C21x34B 器件，但您仍可以在您的套件上使用 CY8C21x34B 器件，从而可以使用 SmartSense 技术。如果想索取 CY8C21x34B 套件，请发送邮件到 capsense@cypress.com。

使用 CapSense 控制器进行通信

*代码示例 2（通过 I2C 使 CY8C20xx6A CapSense 控制器与主机通信）*和*代码示例 3（通过 UART 与 CY8C20xx6A CapSense 控制器进行数据传输）*描述了如何使用 I²C 和 UART 协议将 CapSense 控制器 CY8C20xx6A 和主机处理器进行连接。将这些项目移植到其他 CapSense 控制器的操作很简单，并且不需要进行太多修改。

将代码示例 3 导入到其他器件中

1. 使用您需要的目标器件创建一个新项目。
   相应的 UCC 套件指南中提供了用于创建新项目的程序。
2. 将 UART UM 与 CapSense UM（CSD、CSA、CSA_EMC 或 SmartSense）一起放置。
3. 为了确保操作稳定可靠，可根据情况采用代码示例 7、代码示例 8 或代码示例 9 部分介绍的方法调校 CapSense UM。对于 SmartSense，不需要执行该步骤。
4. 根据要求，配置 CapSense 和 UART UM。
   您可以将代码示例 3 中相同的设置使用于 UART UM。
5. 在工作区浏览器中打开现有的 *main.c* 文件。将代码示例 3 项目中的 *main.c* 文件内容代替现有的 *main.c* 文件内容。
6. 将代码示例 3 文件夹中的 *display.c* 和 *display.h* 文件添加到您的项目文件夹内。
7. 完成 *3.7 执行代码示例*，从而执行您的项目。
   请参见*使用 MultiChart 将 CapSense 数据绘制成图表*，以查看 MultiChart 工具中已绘制的数据。

将 I²C 通信使用于 CY8C24x94 器件

I²C 协议可用于本文档中调校代码示例过程中的数据传输操作。请参考代码示例 7，代码示例 8 和代码示例 9 项目，了解如何将 I²C 使用于特定器件。要想将 I²C 通信用于 CY8C24x94 器件，请按照以下步骤进行操作。

1. 创建一个以 CY8C24x94 为目标器件的新项目。
   请参见 CY3280-24x94 UCC 套件指南，了解如何使用该套件创建新项目。
2. 将 I²C UM 与 CSD UM 一起放置在您的项目中。
3. 为了确保稳定可靠的操作，请使用代码示例 7 中介绍的方法调校 CSD UM。
4. 根据需要配置 CapSense 和 I²C UM。
   您可以将代码示例 2 中所使用的设置适用于 I²C UM。
5. 将代码示例 2 项目文件夹中的 *main.c*， *display.c* 和 *display.h* 文件添加到您的项目中。
6. 打开 Workspace Explorer 中现有的 *main.c* 文件。将现有的 *main.c* 内容替换为代码示例 2 项目中的 *main.c* 文件内容。
7. 将代码示例 2 文件夹中的 *display.c* 和 *display.h* 文件添加到您的项目文件夹中。
8. 完成 *2.7 执行代码示例*部分中的步骤，以执行您的项目。

9. 要想通过 I²C 查看 CapSense 数据，请将代码示例 1 项目文件夹中的 *BCP Configuration Files* 文件夹复制到您的项目位置，然后完成 *1.8 通过 I2C 读取 CapSense 数据*。

测量传感器的绝对电容

“CE #4 — 使用 CY8C20xx6A CapSense 控制器测量传感器的绝对电容”、*“CE #5 — 使用 CY8C21x34/B CapSense 控制器测量传感器的绝对电容”*和*“CE #6 量 — 使用 CY8C20x34 CapSense 控制器测量传感器的绝对电容”*测量 CY8C20xx6A、CY8C21x34/B 和 CY8C20x34 器件的传感器的绝对电容。由于 CSD UM 的结构类似性，所以将计算 CY8C21x34/B 传感器电容的方法用于 CY8C24x94 器件。更多有关 CY8C24x94 中 CSD UM 的信息，请参见 CY8C24x94 的 CSD 数据手册。

将代码示例 5 导入到 CY8C24x94 器件内。

1. 创建一个以 CY8C24x94 为目标器件的新项目。
   请参见 CY3280-24x94 UCC 套件指南，了解如何使用该套件创建一个新项目。
2. 将 CSD 和 UART UM 放置在您的项目中。根据需要配置 UM。
   您可以将代码示例 5 中所使用的设置使用到您的项目中。
3. 在工作区浏览器中打开现有的 *main.c* 文件。替换代码示例 5 项目中现有 *main.c* 文件的内容。
4. 请参考 *CY8C21x34/B_Calculsensitivity.xls* 文件，了解如何计算灵敏度。将 *main.c* 文件中的灵敏度值替换为计算出的值。
5. 完成 *6.7 执行代码示例*部分中的步骤，以便在 HyperTerminal 工具上查看相应的绝对电容。
   代码示例 7 到 9 说明了如何调校 CapSense UM 模块，这两个代码示例被特定于该项目中所使用的 UM 模块和器件，其他器件不可用。

相关文档

设计指南

- Capsense 入门
- CY8C20xx6A CapSense 设计指南
- CY8C20x34 CapSense 设计指南
- CY8C21x34/B CapSense 设计指南

数据手册

- CSD 数据手册
- CSA_EMC 数据手册
- SmartSense 数据手册

应用笔记

- AN2397 — CapSense 数据查看工具

套件指南

- CY3280-20x66 通用 CapSense 控制器套件指南
- CY3280-20x34 通用 CapSense 控制器套件指南
- CY3280-21x34 通用 CapSense 控制器套件指南

缩略语

缩略语	定义
AN	应用笔记
API	应用编程接口（API）
BSM	简单按键模块
CE	代码示例
COM	通信
CSA	CapSense 逐次逼近
CSD	CapSense Sigma Delta
EMC	电磁兼容性
PC	个人电脑
SLM	线性滑条模块
UART	通用异步接收器/发送器
UM	用户模块
USB	通用串行总线

文档修订记录

版本	提交日期	变更者	变更说明
**	01/28/2013	ZCLI	本文档版本号为 Rev**，译自英文版 001-74590 Rev*A。
*A	10/14/2016	YLIU	本文档版本号为 Rev*A，译自英文版 001-74590 Rev*E。