AN66271 — CY8C21x34/B

CapSense®设计指南

文档编号：001-78594 版本*D

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

版权所有

赛普拉斯半导体公司，2010-2019 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

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

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

目录

1. 简介 5

1.1 摘要 5
1.2 赛普拉斯的 CapSense 文档体系 5
1.3 CY8C21x34/B CapSense Plus 系列功能 7
1.4 文档规范 8

2. CapSense 技术 9

2.1 CapSense 基本原理 9
2.2 CY8C21x34/B 中的 CapSense 方法 11

3. CapSense 设计工具 14

3.1 概述 14
3.2 用户模块概述 16
3.3 CapSense 用户模块全局数组 17
3.4 CSD 用户模块配置 18
3.5 CSD 用户模块参数 18
3.6 CSD 用户模块高级参数 19
3.7 CSD 用户模块低级参数 21
3.8 CSDADC 用户模块配置 23
3.9 CSDADC 用户模块参数 24
3.10 低级参数 24
3.11 SmartSense 用户模块参数 25
3.12 低级参数 26

4. 使用用户模块对 CapSense 性能进行调校 27

4.1 基本注意事项 27
4.2 调校 CSD 和 CSDADC 用户模块 30
4.3 配置 SmartSense 用户模块 33

5. 设计的注意事项 35

5.1 覆盖层选择 35
5.2 ESD 保护 36
5.3 电磁兼容性（EMC）的注意事项 37
5.4 软件过滤 38
5.5 功耗 39

5.6 引脚分配 40
5.7 PCB 布局指南 41

6. 防水功能 42

6.1 屏蔽电极和保护传感器 42
6.2 设计建议 48

7. 接近感应 49

7.1 接近感应传感器类型 49
7.2 设计建议 50

8. 低功耗设计注意事项 51

8.1 其它节能技术 51
8.2 唤醒后执行序列 53

9. 资源 54

9.1 网站 54
9.2 数据手册 54
9.3 技术参考手册 54
9.4 开发套件 54
9.5 PSoC Programmer 55
9.6 MultiChart 55
9.7 PSoC Designer 55
9.8 代码示例 55
9.9 设计支持 56

术语表 57

修订记录 62

文档修订记录 62

1. 简介

1.1 摘要

本文档提供有关使用 CapSense Plus™控制器 CY8C21x34/B 系列电容式触摸感应（CapSense®）功能的设计指南。本指南包含以下主题：

- CapSense Plus 控制器 CY8C21x34/B 系列的功能
- CapSense 的操作原理
- CapSense 设计工具简介
- 调试 CapSense 触摸感应系统以获得最佳性能的详细指南
- 防水和接近感应检测等各项高级功能
- 电气和机械系统设计注意事项
- 在系统中设计 CapSense 所使用的附加资源和支持

1.2 赛普拉斯的 CapSense 文档体系

图 1-1 和表 1-1 汇总了赛普拉斯 CapSense 文档体系。利用这些资源，可快速获得所需的信息来完成 CapSense 产品设计。

图 1-1 显示了利用电容式感应进行产品设计的典型流程；本指南中与该主题密切相关的信息用绿色文字高亮显示。表 1-1 提供了与图 1-1 中所列出的每个已编号任务相对应的文档链接。

图 1-1. CapSense 产品的典型设计流程

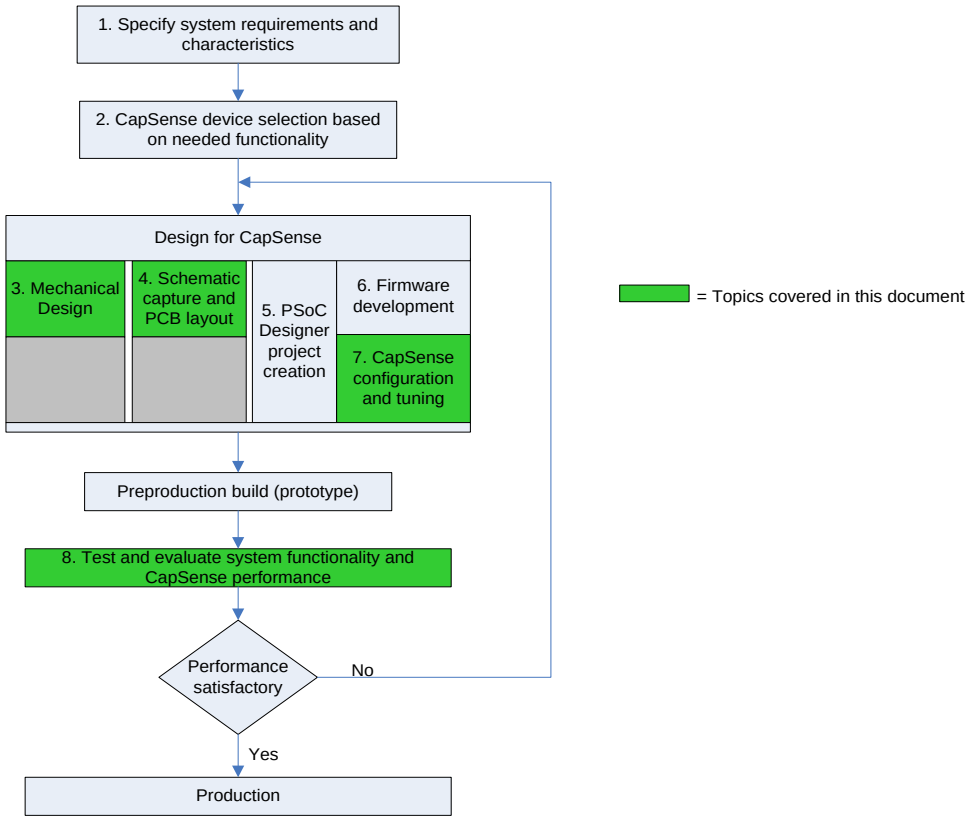


表 1-1. 图 1-1 中每个已编号的任务相对应的赛普拉斯文档

图 1-1 中已编号的设计任务	赛普拉斯的 CapSense 文档
1	Capsense 入门 PSoC 1 入门
2	Capsense 入门 CY8C21x34/B CapSense 器件数据手册
3	Capsense 入门 PSoC 系列特定的 CapSense 设计指南（本文档） CapSense 应用笔记
4	Capsense 入门 PSoC 系列特定的 CapSense®设计指南（本文档） CapSense 应用笔记
5	PSoC Designer™用户指南
6	CapSense 应用笔记 CapSense 代码示例 PSoC 系列特定的技术参考手册（适用于 CY8C21x34/B） AN2014 — PSoC® 1 编程的基础知识 AN2015 — PSoC® 1：闪存和 E2PROM 入门 AN44168 — 使用外部微控制器（HSSP）编程 PSoC® 1 器件 PSoC® 1 的 ISSP 编程规范

图 1-1 中已编号的设计任务	赛普拉斯的 CapSense 文档
7	PSoC 系列特定的 CapSense 设计指南（本文档） PSoC 系列特定的 CapSense 用户模块数据手册（CSD、CSDADC 和 SmartSense） PSoC 系列特定的技术参考手册（适用于 CY8C21x34/B） CapSense 控制器代码示例设计指南
8	PSoC 系列特定的 CapSense 设计指南（本文档） CapSense 代码示例

1.3 CY8C21x34/B CapSense Plus 系列功能

赛普拉斯 CY8C21x34/B 是一个低功耗、高性能、可编程的 CapSense 控制器系列，其功能如下：

1.3.1 支持高级触摸式感应功能

- 可编程电容式感应元件
  - 使用 CSD 和 CSDADC 电容式感应技术支持 CapSense 按键、滑条和接近感应传感器的组合
  - 支持 SmartSense™自动调校（CY8C21x34B）
  - 包含用来实施按键和滑条的集成式 API
  - 支持 24 个电容式按键和 4 个滑条
  - 支持 5 cm 的接近感应系统（使用 PCB 板上引线）
  - 通过屏蔽电极提供防水功能

1.3.2 器件特性

- 高性能、低功耗 M8C Harvard 架构处理器
  - M8C 处理器的速度最高可达 24 MHz
- 灵活的片上存储器
  - 高达 8 KB 闪存和 512 字节的 SRAM
  - 仿真型 EEPROM
- 高精度的可编程时钟
  - 内部主振荡器（IMO）：24/48 MHz ± 2.5%
  - 32 kHz 的内部低速振荡器（ILO），用于为看门狗和在睡眠模式的操作提供脉冲
- 增强型 GPIO 功能
  - 高达 28 个 GPIO，具有可编程引脚配置
  - 所有 GPIO 上均具有 25 mA 的灌电流能力
  - 所有 GPIO 上均支持内部电阻上拉、高阻态、开漏驱动和强驱动模式
- 外设功能
  - 计数器、定时器、脉宽调制器（PWM）和脉宽鉴别器（PWD）和由硬件驱动的 7 段显示屏
  - 可将 I^2C 配置为主设备、从设备和多主设备来进行通信
  - 支持 SPI、UART、IRDA 和单线通信协议
- 工作条件
  - 工作电压：2.4 V ~ 5.25 V，使用片上开关电压泵时工作电压可低至 1.0 V
  - 温度范围：–40 ˚C ~ +85 ˚C
- 封装
  - 16-SOIC（150 mil）到 32-QFN（5 x 5 mm）
  - AEC 合格的汽车级器件：CY8C21334 和 CY8C21534

1.4 文档规范

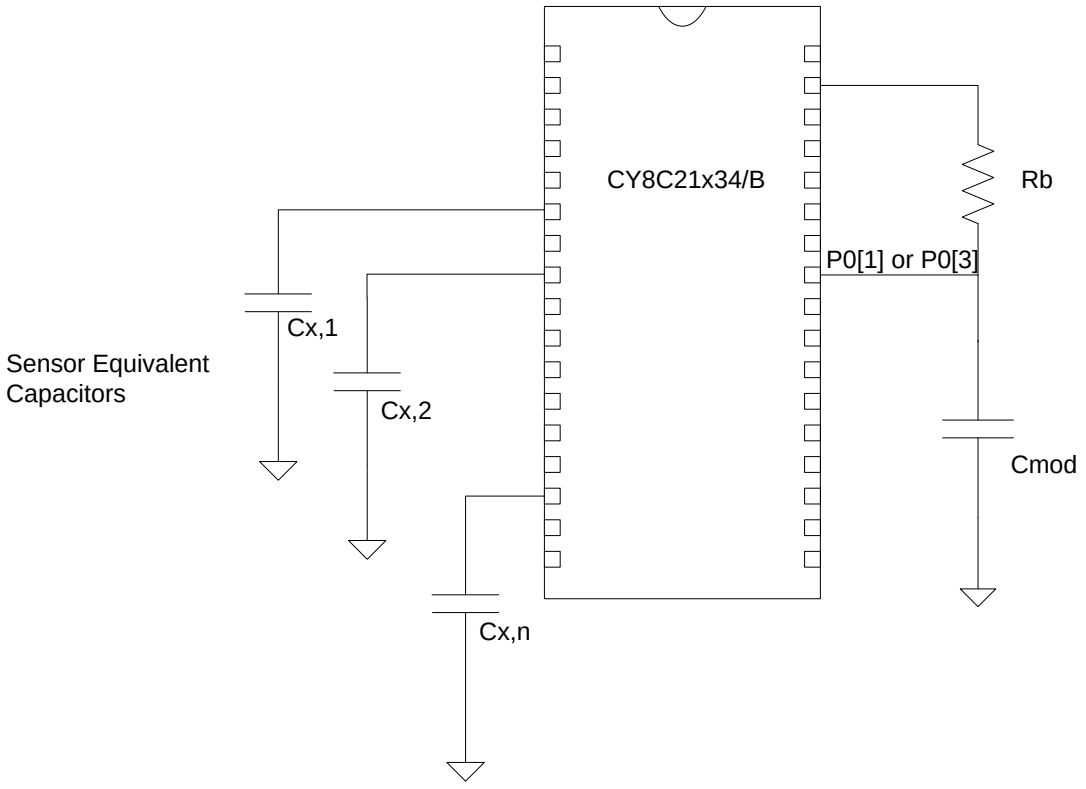
规范	使用说明
Courier New 字体	显示文件位置、用户输入的文本和源代码： C:\ ...cd\icc\
斜体字	用于显示文件名称和参考文档： 请阅读 *PSoC Designer 用户指南*中的 *sourcefile.hex* 文件。
[方括号、粗体]	用于显示程序中的键盘指令： **[Enter]**或**[Ctrl] [C]**
File > Open	表示菜单路径： File > Open > New Project
粗体字	用于显示操作过程中的各条指令、菜单路径和图标名称： 请点击 **File** 图标，然后点击 **Open**。
Times New Roman 字体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

2. CapSense 技术

2.1 CapSense 基本原理

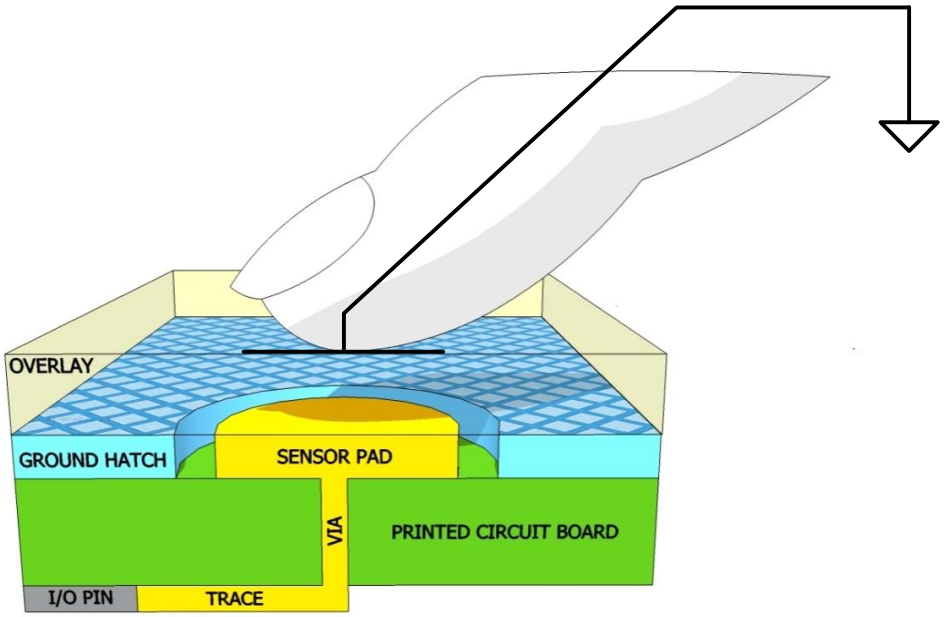
CapSense 是一种触摸式感应技术，其工作方式是使用 CapSense 控制器测量每个被预先指定为传感器的 I/O 引脚上的电容。如图 2-1 所示，每个传感器引脚的总电容可以模拟为从 $C_{x,1}$ 到 $C_{x,n}$ 的等效集总电容（其中，'n'是设计中的传感器数量）。与 CY8C21x34/B 器件连接的内部电路将每个 C_x 数量级转换成数字代码，然后进行存储以备后期处理。其它组件（即 R_b 和 C_{MOD}）由 CapSense 控制器内部电路使用。

图 2-1. 在 CY8C21x34/B PSoC 器件中实施 CapSense



如图 2-2 所示，每个传感器 I/O 引脚均通过走线、过孔或在必要时同时使用二者连接至传感器垫片。 覆盖层是传感器垫片上面的非导电性盖板，这构成了产品的触摸式接口。当手指与覆盖层接触时，人体组织的导电性会产生一个与传感器垫片并行的接地导电层，如图 2-2 所示。该样会构成一个平行板电容器；其容值可通过公式 1 计算得到：

图 2-2. 典型的 CapSense PCB 与通过手指激活的传感器之间的横截面图



$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{D}$$ 公式 1

其中：

C_F = 手指与传感器覆盖层接触时所产生的电容值

ε_0 = 空气介电常数

ε_r = 覆盖层的绝缘常数（相对介电常数）

A = 手指与传感器垫片覆盖层的接触面积

D = 覆盖层的厚度

除了平行板电容值外，手指与覆盖层的接触也会在其自身与附近其它导体之间产生边缘电场。与平行板电容相比，这些边缘电场的影响通常很轻微，所以通常可以忽略它们。

即使手指未触摸覆盖层，传感器 I/O 引脚也有一些寄生电容（C_P）。C_P 是由 CapSense 控制器内部寄生组件与耦合电场共同产生，其中后者是系统中的传感器垫片、走线、过孔以及其它导体之间耦合产生的。这些导体可以是接地层、其它走线、产品的机箱或外壳中的任何金属，等等。CapSense 控制器可测量连接到传感器引脚的总电容（C_x）。

手指未接触传感器时：

$$C_X = C_P$$ 公式 2

手指触摸传感器垫片时：

$$C_X = C_P + C_F$$ 公式 3

通常，C_P 比 C_F 大一个数量级以上。C_P 的取值范围通常为 6 ~ 15 pF，但在极端情况下它的值可以高达 50 pF。C_F 的取值范围通常为 0.1 pF - 0.4 pF。调整 CapSense 系统时，C_P 的数量级至关重要，该内容将在使用用户模块对 CapSense 性能进行调校中进行讨论。

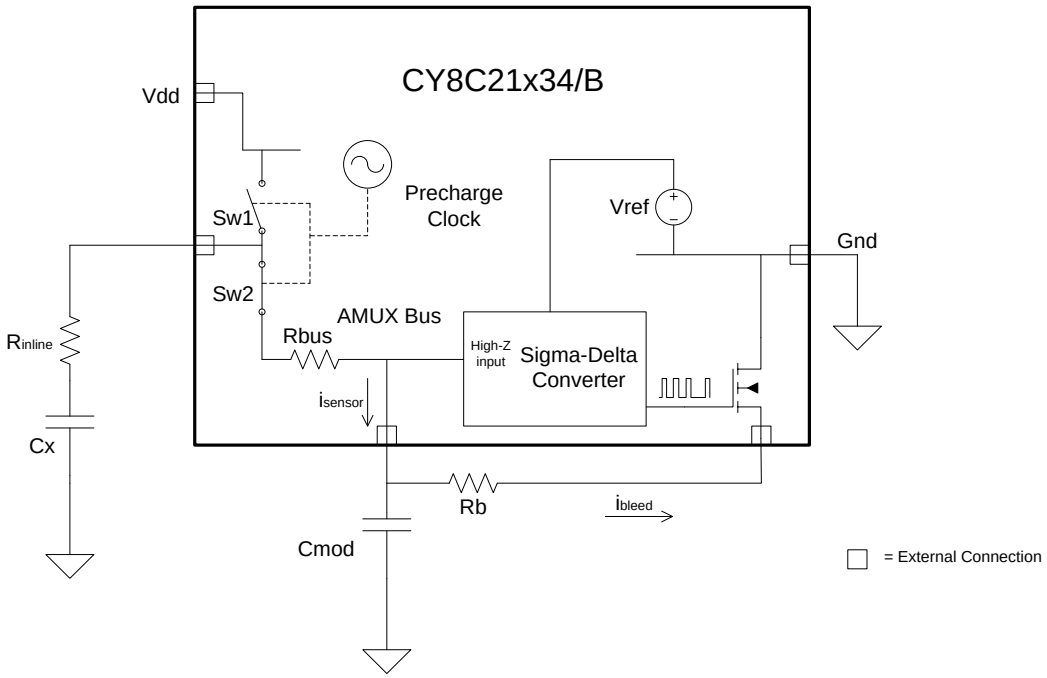
2.2 CY8C21x34/B 中的 CapSense 方法

CY8C21x34/B 器件支持多个用来将传感器电容（C_x）转换成数字代码的 CapSense 方法。这些方法是 CapSense Sigma Delta（CSD）、支持 ADC 的 CSD（CSDADC）和 SmartSense。这些方法使用 PSoC Designer 用户模块来实现，本设计指南的第 3 节中介绍了详情。

2.2.1 CapSense Sigma-Delta（CSD）

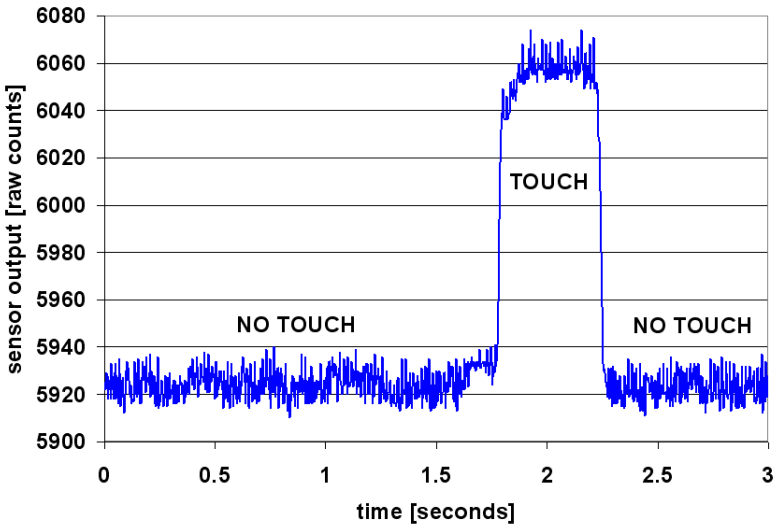
CY8C21x34/B 器件中的 CSD 方法将 C_x 整合至开关电容电路中，如图 2-3 所示。该传感器（C_x）分别通过负遮盖的开关 Sw1 和 Sw2 交替连接到 V_{DD} 和模拟复用器（AMUX）总线。Sw1 和 Sw2 由预充电时钟驱动，用于生成输入 AMUX 总线的电流（I_{SENSOR}）。I_{SENSOR} 的数量级与 C_x 的数量级成正比。Sigma-Delta 转换器采集 AMUX 总线电压样本并生成一个用于控制恒流源（I_{DAC}）的调制位元流，该恒流源对 AMUX 充电使 AMUX 总线的平均电压便保持为 V_{REF}。传感器释放调制电容器（C_{MOD}）中的电荷 I_{SENSOR}。C_{MOD} 与 Rbus 共同构成一个低通滤波器，该滤波器消弱了 Sigma-Delta 转换器输入的预充电开关跃变电压。

图 2-3. CSD 框图



将 AMUX 平均电压保持为稳态值（V_{REF}）时，Sigma-Delta 转换器通过控制位元流占空比的方式使平均的泄漏电流（I_{BLEED}）与 I_{SENSOR} 相互匹配。Sigma-Delta 转换器在传感器扫描期间内存储位元流，其累积结果为数字输出值，称为原始计数，该值与 C_x 成正比。该原始计数由高级算法进行解析，以便求得传感器的状态。手指触摸然后释放传感器的过程中，得到若干连续扫描结果，由此绘制出 CSD 原始计数图，如图 2-4 所示。根据 CapSense 基本原理中的解释，手指触摸传感器导致 C_x 增加 C_F，进而导致原始计数按比例增加。通过比较稳态下原始计数水平到预定阈值转变，高级算法能够确定传感器是处于 ON（触摸）还是 OFF（离开）状态。

图 2-4. 手指触摸期间的 CSD 原始计数



2.2.2 支持 ADC 功能的 CSD（CSDADC）

CSDADC CapSense 方法的功能与 CSD 基本相同，但该方法除具有电容测量功能外，还扩增了 ADC 功能。

2.2.2.1ADC 功能

- 支持两种在运行时可进行切换的绝对和比例 ADC 输入模式，这样能够轻松扫描不同类型的传感器。
- 支持用于绝对电压输入模式的单斜 ADC
- 单斜 ADC 采用了内置校准机制
- 用于比例输入模式的积分增量型 ADC

2.2.3 SmartSense 自动调校

调试触摸感应用户界面是确保系统正常运行和保持良好用户体验的重要步骤。典型设计流程包括：在初始设计阶段和系统整合过程中调校传感器界面，并在投入量产前进行最后的生产微调。调校是一个重复过程，可能非常耗时。SmartSense 自动调校性能用于简化用户界面开发周期。此外，该技术还很简单易用，通过去除从原型到批量生产的整个产品开发周期中的调试过程，来大幅缩短设计周期时间。SmartSense 在上电时自动调校每个 CapSense 传感器，然后在运行期间负责监控和维持传感器的最佳性能。SmartSense 自动调校技术适用于解决 PCB、覆盖层和噪声生成器（如 LCD 反相器、交流电路噪声和开关模式电源）的制造性差异，并自动将其修正。SmartSense 自动调校要求每个传感器上要有一个介于 2.1 KB ~ 3.7 KB 之间的闪存（根据配置）和 29 B RAM。

2.2.3.1过程差异

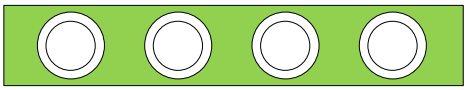
CY8C21x34B SmartSense 用户模块（UM）能够与寄生电容在 5 ~ 45 pF 范围的传感器配合使用（典型的传感器的寄生电容值 C_P 介于 10 ~ 20 pF 范围之间）。各个传感器的灵敏度参数会根据特定传感器的特性自动设置。由于在指定范围 5 ~ 41 pF 内，无论传感器之间的 C_P 如何变动，都保持每个传感器的响应一致，因此提高了批量生产的效率。个别传感器的寄生电容可能由于 PCB 布局、PCB 制造流程各不相同或多源供应链中的不同 PCB 供应商，有所变化。传感器的灵敏度取决于寄生电容的大小；C_P 值越高，传感器灵敏度就越低，进而导致手指触摸信号振幅降低。在某些情况下，C_P 值的变化会解调系统，使传感器无法达到最佳性能（过于灵敏或不够灵敏），或在最坏的情况下，传感器不能正常工作。在上述任一情况下，您都必须重新调试系统；在某些情况下，还需要重新验证 UI 子系统。SmartSense 自动调校可以解决这些问题。

通过 SmartSense 自动调校，可以实现平台设计。设想在笔记本电脑上设置电容式多媒体触摸感应键；两个按键之间的间距大小取决于笔记本电脑尺寸和键盘布局。在本示例中，宽屏机型要比标准屏机型的按键间距大得多。两个按键之间的间距越大，传感器与 CapSense 控制器之间的走线就越长，进而导致传感器寄生电容越高。这意味着在相同平台上设计的不同机型，CapSense 按键的寄生电容可能不同。虽然对于所有笔记本机型而言这些按键的功能均是相同的，但是，传感器必须针对每种机型进行调校。使用推荐的最佳实践（参见 CapSense 入门中的 PCB 布局），SmartSense 可以帮助您实现平台设计，调校功能会自动和有效地完成。

图 2-5. 21 英寸笔记本电脑的多媒体按键设计



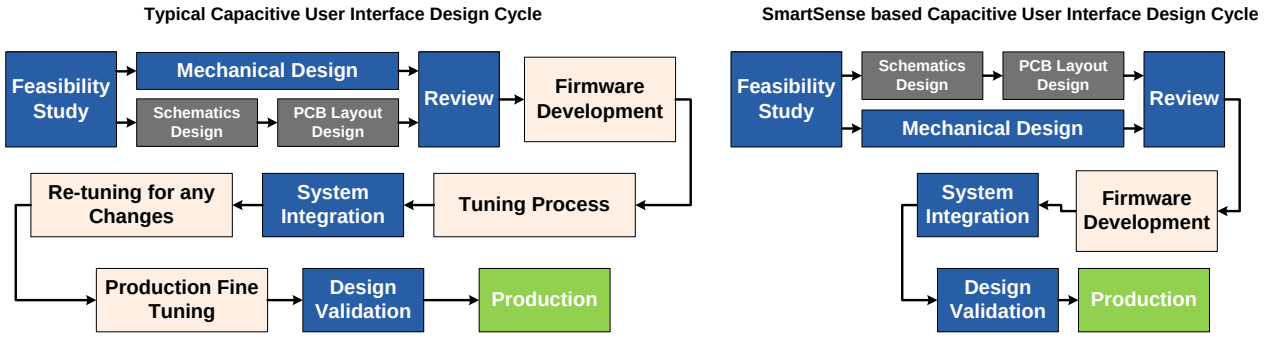
图 2-6. 15 英寸笔记本电脑机型的多媒体按键设计（该机型与 21 英寸机型有相同的功能和按键大小）



2.2.3.2 缩短了设计时间

通常，电容式传感器接口设计最耗时的任务是固件开发和传感器调校。使用典型触摸感应控制器时，如果将相同设计移植到了不同的模型或者 PCB 或传感器 PCB 布局的机械尺寸发生了变化，便需要重新调校传感器。SmartSense 设计解决了这些挑战，因为 SmartSense 需要较少的固件开发工程，无须调校和重新调校过程。这样明显加快了常规设计周期的进程。图 2-7 比较了典型的触摸感应式控制器和基于 SmartSense 设计流程。

图 2-7. 典型电容式界面设计流程的比较



3. CapSense 设计工具

3.1 概述

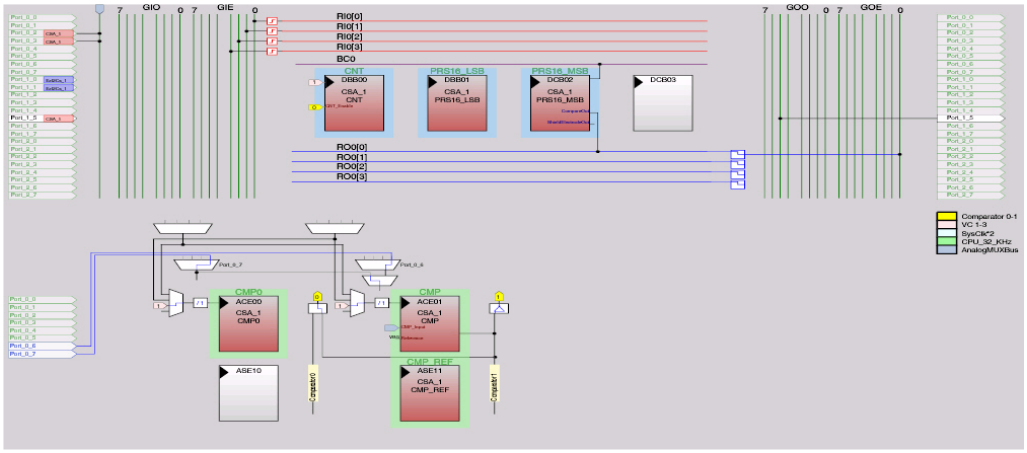
赛普拉斯提供用来开发 CapSense 电容式触摸感应应用的全套硬件和软件开发工具。CY8C21x34/B 系列的基本开发系统包括本章节所讨论的组件。有关订购信息，请参见资源部分。

3.1.1 PSoC Designer 和用户模块

赛普拉斯独有的集成设计环境（PSoC Designer）支持配置模拟和数据模块、开发固件、调校和调试设计。在拖放式设计环境中使用用户模块库开发这些应用程序。要配置用户模块，可通过器件编辑器 GUI 实现或通过固件写入特定的寄存器内来完成。PSoC Designer 具有内置的 C 语言编译器和嵌入式编程器。专业版编译器可用于复杂设计。

CSD、CSDADC 和 SmartSense 用户模块使用开关式电容器电路、模拟复用器、电压比较器、数字计数功能和高级软件子程序（API）来实现电容式触摸传感器。其它模拟和数字外设的用户模块可用于实现其它功能，例如，I²C、SPI、TX8、定时器和 PWM。

图 3-1. PSoC Designer 器件编辑器



3.1.1.1 CapSense 用户模块入门

要想在 PSoC Designer 中新建 CY8C21x34/B 项目：

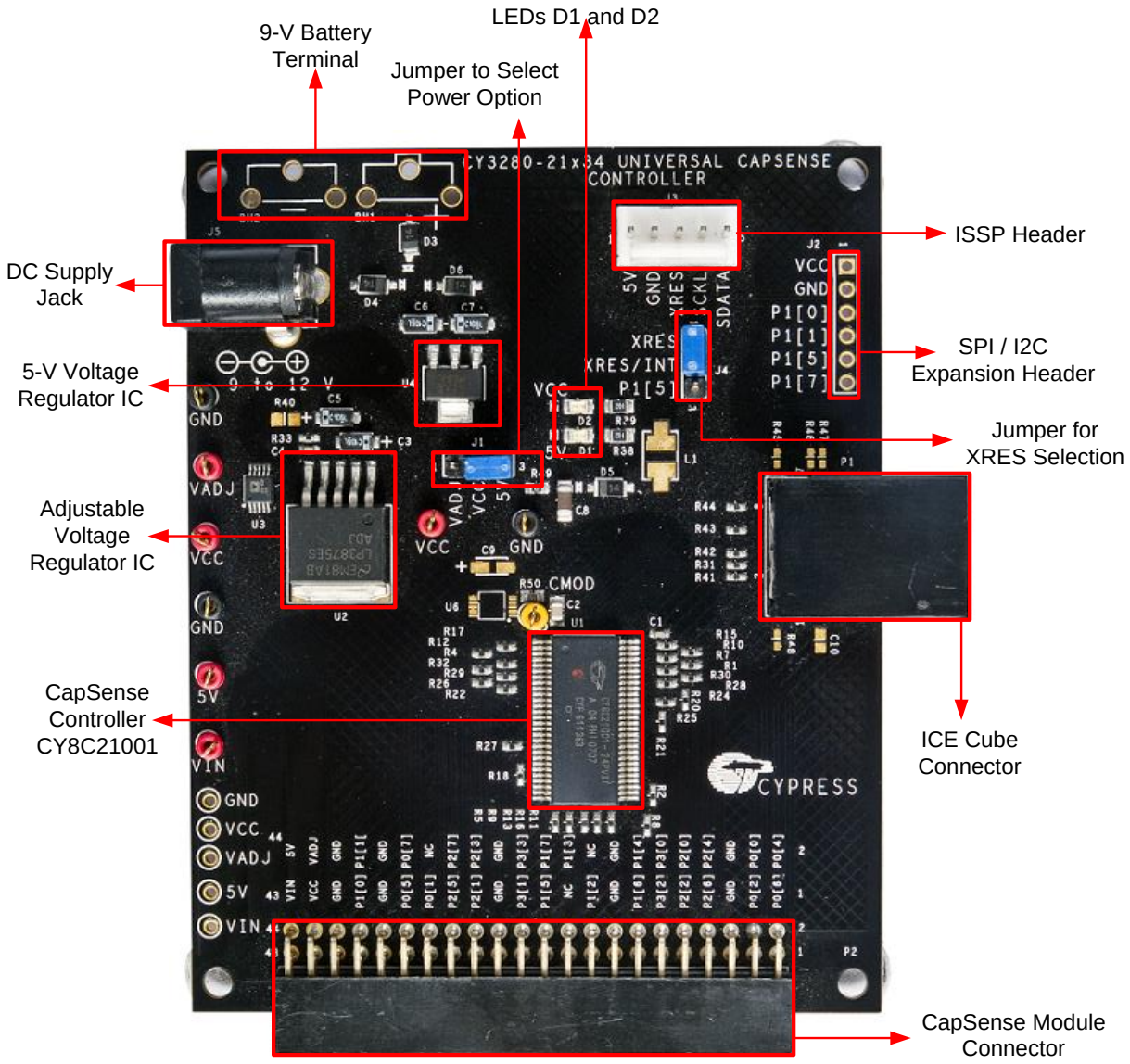
1. 新建 CY8C21x34/B PSoC Designer 项目。
2. 选择和放置需要专用引脚的任何用户模块（例如，I²C 或 LCD），并分配端口和引脚。
3. 选择和放置 CSD、CSDADC 或 SmartSense 用户模块。
4. 右键单击用户模块，访问用户模块向导。
5. 设置按键传感器数量、滑条配置、引脚分配及相关项。
6. 设置引脚和全局用户模块参数。
7. 生成应用，并切换到应用编辑器。

8. 根据用户模块数据手册：（CY8C21x34）调整示例代码，使按键或滑条生效。

3.1.2 通用的 CapSense 控制器套件

通用的 CY3280-BK1 CapSense 控制器套件带有预定义的控制电路和插入式硬件，用来简化原型设计和调试系统。该套件还包含了用于调校和数据采集的编程硬件和 I²C 至 USB 桥接器。

图 3-2. CY3280-21x34 CapSense 控制器套件



3.1.3 通用 CapSense 控制器模块电路板

赛普拉斯模块电路板具有一系列传感器、LED 和接口，以满足您的应用需求。

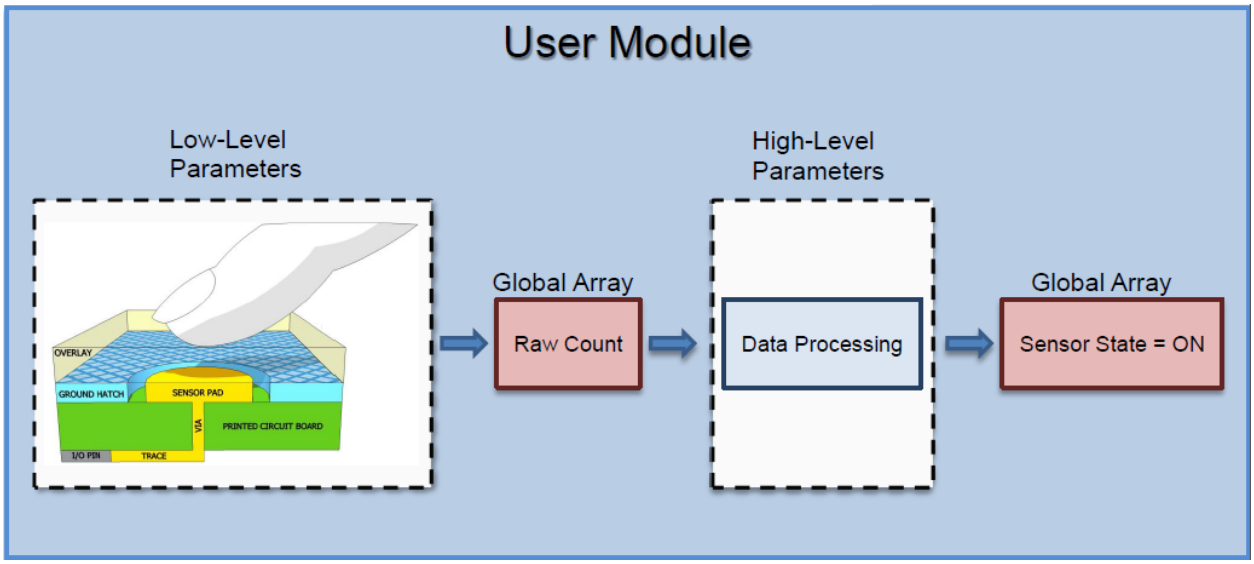
- CY3280-BSM 的简单按键模块
- CY3280-BMM 矩阵按键模块
- CY3280-SLM 线性滑条模块
- CY3280-SRM 辐射滑条模块
- CY3280-BBM 通用 CapSense 原型设计模块

3.1.4 CapSense 数据查看工具

在 CapSense 设计过程中，您需要监测用于调校和调试的 CapSense 数据（原始计数、基准线、计数差值等）。通过两种 CapSense 数据查看工具（即 MultiChart 和桥接控制面板（BCP））来实现该操作。应用笔记：AN2397 — CapSense 数据查看工具会详细介绍这些工具。

3.2 用户模块概述

图 3-3. 用户模块框图



用户模拟包含整个 CapSense 系统，即从物理感应到数据处理。使用各种参数定义用户模块的行为。这些参数影响感应系统的不同器件，可分为低级参数和高级参数，在这些参数之间使用全局数组进行通信。

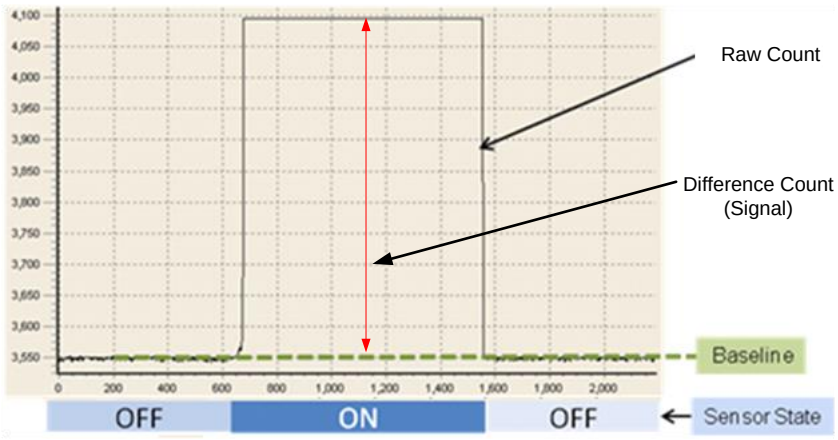
低级参数（如扫描传感器的速度和分辨率）用于定义感应方法在物理层上的行为，并与电容到原始计数的转换相互关联。每种感应方法具有独特的低级参数，相关描述请参见 CSDADC 用户模块配置、CSDADC 用户模块参数和 SmartSense 用户模块参数等部分。

高级参数（如去抖动计数和噪声阈值）用于定义如何处理原始计数，以产生相关信息，例如：传感器 ON/OFF（触摸/离开）状态和手指在滑条上的大致位置。这些参数对于所有感应方法而言均是相同的，相关描述请参见 CSD 用户模块高级参数部分。

3.3 CapSense 用户模块全局数组

学习 CapSense 用户模块参数之前，您必须熟悉 CapSense 系统所使用的某些全局数组。这些数组不应通过手动方式来改变，但可以针对调试这一目的进行测量。

图 3-4. 原始计数、基准线、计数差值和传感器状态



3.3.1 原始计数

CapSense 控制器中的硬件电路用于测量传感器的电容。调用用户模块 API *UMname*_ScanSensor()（其中 *UMname* 可能是 CSD、CSDADC 或 SmartSense）时，该 API 以数字格式存储此结果（又称原始计数）。

传感器的原始计数与其传感器电容成比例。当传感器电容值增加时，原始计数也随之增加。传感器的原始计数值存储在 *UMname*_waSnsResult[]整数型数组中。该数组在头文件 *UMname.h* 中得到定义。

3.3.2 基准线

渐变的环境改变（如温度和湿度）会影响传感器的原始计数，这会导致计数差异。

用户模块使用复杂的基准线算法来补偿这些差异。该算法使用基准线变量来完成该功能。基准线变量跟踪原始计数值的任何渐变差异。通常，输入原始计数值馈送到数字低通滤波器中，基准线变量记录该滤波器的输出。通过调用用户模块 API *UMname*_UpdateSensorBaseline（其中 *UMname* 可以是 CSD、CSDADC 或 SmartSense），可执行基准线算法。

传感器的基准线值存储在 *UMname*_waSnsBaseline[]整数型数组。该数组在头文件 *UMname.h* 中得到定义。

3.3.3 计数差值（信号）

计数差值也称为传感器信号，被定义为传感器原始计数和基准线之间的计数差值。当传感器处于非活动状态时，计数差值为零。通过触摸激活传感器时，计数差值会变为正数。

传感器的计数差值值存储在 *UMname*_waSnsDiff[] 整数型数组，其中：*UMname* 可以是 CSD、SmartSense 或 CSA_EMC。在头文件 *UMname.h* 中定义该数组。计数差值变量可通过用户模块 API *UMname*_UpdateSensorBaseline() 更新。

3.3.4 传感器状态

传感器状态表示物理传感器的活动/非活动状态。当手指触摸传感器时，传感器的状态从 0 变为 1，而手指释放时，传感器的状态返回到 0。

传感器的状态存储在名为 *UMname*_baSnsOnMask[] 的字节数组内，其中：*UMname* 可以是 CSD、CSDADC 或 SmartSense。每个数组元素存储 8 个连续传感器的状态。传感器状态可通过用户模块 API *UMname*_bIsAnySensorActive()更新。

3.3.5 信号

该数组仅适用于 SmartSense 用户模块。对计数差值进行标准化以获得信号。通常，传感器处于非活动状态时，信号值为零。触摸传感器时，原始计数会增大，从而产生正差值信号值。该数组用于阈值模式参数被设置为手动模式时，调试 SmartSense 用户模块。

信号值被存储在 SmartSense_baSnsSignal[]的整数型数组中。

3.4 CSD 用户模块配置

CSD 用户模块有两个可选的时钟配置。这些配置针对 CSD 前端的开关式电容器电路使用了不同信号源，如第 11 页上的图 2-3 所示。

您需要在 PSoC Designer 项目中初次放置 CSD 用户模块时，选择时钟配置。您可以通过右键单击 CSD 用户模块并选择 **User Module Selection Options**（用户模块选择选项）来更改此选择。

3.4.1 CSD 不带有时钟预分频器

在这种配置中，16 位伪随机序列（PRS）用作 CSD 前端开关式电容器电路的信号源。系统时钟（IMO）是 PRS 的时钟源。PRS 扩展时钟的频谱，并提供优异的抗噪能力。该配置需要 3 个数字模块。

3.4.2 CSD 带有时钟预分频器

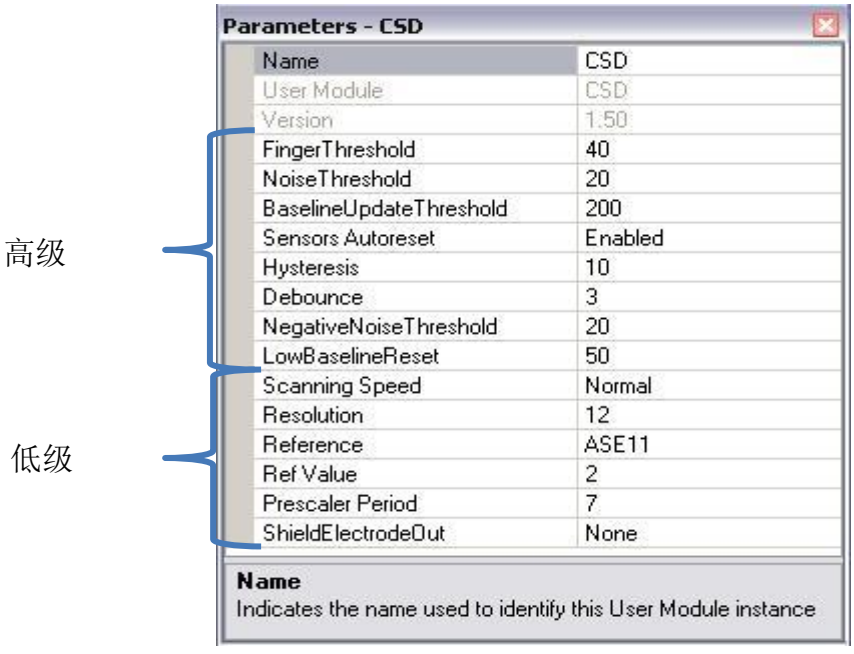
在这种配置中，一个 8 位 PRS 作为 CSD 前端开关式电容器电路的信号源使用。IMO 由可调整式预分频器进行分频，并作为 PRS 的时钟源使用。PRS 扩展时钟的频谱，从而提供抗噪功能。请注意，16 位 PRS 用于不带时钟预分频器配置的 CSD，可提供优异的抗噪功能。这种配置需要 3 个数字模块。

由于它需要较低的开关频率，在具有高 C_P 值的条件下工作时，可以使用此配置。有关 C_P 和开关频率之间的关系，请参阅选择预分频器部分。

3.5 CSD 用户模块参数

CSD 用户模块参数分为高级参数和低级参数两种。有关 CSD 用户模块参数列表和如何分类这些参数的详细内容，请参见图 3-5。

图 3-5. PSoC Designer — CSD 参数窗口



3.6 CSD 用户模块高级参数

3.6.1 手指阈值

用户模块使用此手指阈值参数判断传感器是否为活动/非活动状态。如果传感器的计数差值超过手指阈值，则传感器被判定为活动状态。该定义假设迟滞为 0，去抖动值为 1。

手指阈值的取值范围为 5 ~ 255。

3.6.2 噪声阈值

用户模块使用噪声阈值来了解原始计数中噪声计数的上限。对于单个传感器而言，当原始计数超过基准线且二者之间的差值大于噪声阈值时，基准线更新算法将暂停。

对于滑条传感器而言，当计数差值大于噪声阈值时，质心计算将暂停。

噪声阈值的取值范围为 3 到 255。为了确保用户模块操作正确进行，噪声阈值不能高于手指阈值与迟滞之差。

3.6.3 基准线更新阈值

当原始计数大于当前基准线，且计数差值小于噪声阈值时，当前基准线与原始计数之间的差值被累计到“水桶”中。当“水桶”完全充满时，基准线会增加并清空“水桶”。基准线更新阈值参数设置水桶必须达到以便使基准线增加的阈值。

取值范围为 0 到 255。

3.6.4 传感器自动复位

传感器自动复位参数确定基准线是随时进行更新，还是仅在计数差值小于噪声阈值时才予以更新。

使能传感器自动复位后，会不断地更新基准线。这样会限制传感器的最大持续时间，但可以防止在无任何触摸的情况下，原始计数意外升高而导致传感器处于永久性打开状态。该突发性上升可能由较大的电源电压波动、高能射频噪声源或极快的温度变化引起的。

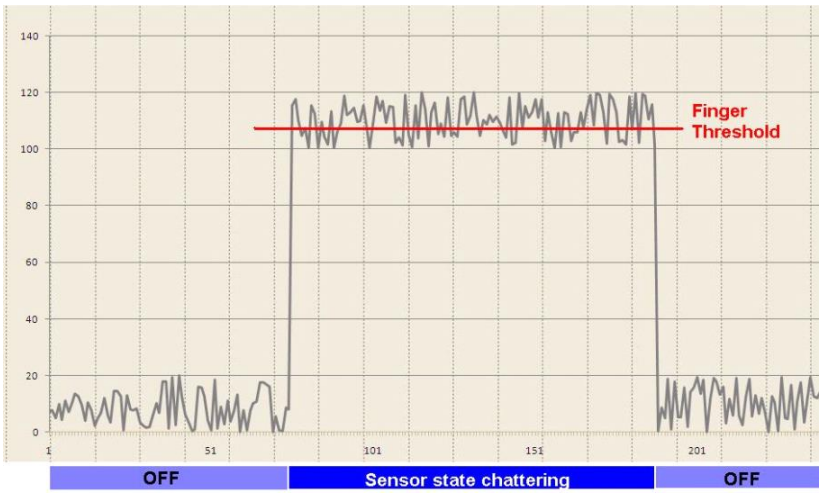
如果禁用了传感器自动复位，仅在计数差值小于噪声阈值参数时才予以更新基准线。

可能值为“使能”和“禁用”。

3.6.5 迟滞

该迟滞设置防止传感器 ON 状态由于系统噪声而发生振动。公式 4 会提供迟滞的功能。该公式假设去抖动值为 1。

图 3-6. 迟滞值为 0 时，传感器状态与计数差值的关系



$$if Difference Count \geq Finger Threshold + Hysteresis, Sensor State = ON$$
$$if Difference Count \leq Finger Threshold - Hysteresis, \ Sensor State = OFF \quad \text{公式 4}$$

3.6.6 去抖动

去抖动参数用于防止原始计数尖峰脉冲将传感器状态从 OFF 更改为 ON。要想将传感器状态从 OFF 转换为 ON，计数差值在指定的采样数内必须大于手指阈值与迟滞之和。

取值范围介于 1 到 255 之间。如果将该参数设置为 1，则不提供去抖动。

3.6.7 负噪声阈值

“负噪声阈值”参数可作为负计数差值阈值使用。对于由低基准线复位参数指定的采样数而言，如果原始计数小于基准线值与负噪声阈值之差，则基准线会被设置为新的原始计数值。

该参数的取值范围为 0 到 255。

3.6.8 低基准线复位

低基准线复位参数与负噪声阈值参数结合使用。它计算复位基准线时所需要的异常的低采样数量。它用来修正启动时手指在传感器上触摸的情况。

该参数的取值范围为 0 到 255。

3.6.9 高级参数建议

下列建议是选择理想参数设置的出发点。

- **手指阈值：**传感器为 ON 时，将该参数设置为计数差值的 75%
- **噪声阈值：**传感器为 OFF 时，将该参数设置为计数差值的 40%
- **基准线更新阈值：**将该参数设置为噪声阈值的两倍
- **传感器自动复位：**根据设计要求
- **迟滞：**传感器为 ON 时，将该参数设置为计数差值的 15%
- **去抖动：**根据设计要求
- **负噪声阈值：**将该参数设置为等于噪声阈值
- **低基准线复位：**设置为 10

3.7 CSD 用户模块低级参数

除高级参数外，CSD 用户模块还具有多个低级参数。这些参数是 CSD 感应方法特定的参数，用于决定如何从传感器中采集原始计数数据。

3.7.1 扫描速度

该参数用于设置传感器扫描速度。虽然较快的扫描速度可以缩短响应时间，但较慢的扫描速度也具有下列优势：

- 提高信噪比
- 提供更好的抗电源和温度变化的能力
- 少量要求系统中断延迟；可以处理较长的中断

有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。这些值可能为超快、快、正常和慢

3.7.2 分辨率

该参数确定扫描分辨率（单位为“位”）。N 位的扫描分辨率最大原始计数为 $2^N–1$。提高分辨率可以提高灵敏度和 SNR，但会增加扫描时间。有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。该参数的值可能为 9 ~ 16 位。

表 3-1. 在 24 MHz IMO 且不带时钟预分频器（PRS16 配置）的工作条件下的扫描速度和分辨率

分辨率（位）	扫描速度（µs）			
	超快速度	快速	正常速度	慢速
9	75	110	170	300
10	110	170	300	510
11	170	300	510	1010
12	300	510	1010	2030
13	510	1010	2030	4060
14	850	1690	3380	6760
15	1520	3040	6080	12200
16	2880	5720	11500	23200

表 3-2. 在 24 MHz IMO 且带有预分频器（PRS8 配置）的工作条件下的扫描速度和分辨率

分辨率（位）	扫描速度（µs）			
	超快速度	快速	正常速度	慢速
9	60	85	150	255
10	85	150	255	510
11	150	255	510	1020
12	255	510	1020	2040
13	510	1020	2010	4080
14	845	1700	3380	6760
15	1530	3060	6120	12100
16	2880	5800	11500	23000

3.7.3 参考电压

要保证 CSD 正常工作，要求一个电压比较器的输入电压参考值。

可能的值为：

- VBG：内部参考电压 1.3 V 派生于固定带隙参考值。
- ASE11：派生于 PWM 的内部可变参考电压。
- AnalogColumn_InputSelect_1：外部电压参考，如电阻分压器网络或外部滤波的 PWM/PRSPWM 信号。该参考电压可以连接到 CapSense 控制器引脚，并作为电压比较器参考源使用。

3.7.4 参考值

该参数的影响取决于参考电压参数的选择。当参考电压来源于 VBG 或 AnalogColumn_InputSelect_1 时，该值不会产生任何影响。当参考电压来源于 ASE11 时，该参数用于设置参考值。

该参数的取值范围为 0（¼ V_{DD} 最小值）到 8（¾ V_{DD} 最大值）。

3.7.5 Prescaler Period（预分频器周期）

扫描时，传感器电容在两个电位之间持续切换。该参数用于设置预分频器周期寄存器，并确定预充电开关输出频率。该参数仅适用于带有时钟分频器配置的 CSD。

取值范围为 1 到 255。

3.7.6 屏蔽电极输出

要求防水功能的应用应启用 Shield Electrode Out（屏蔽电极输出）。可以从其中一个备用数字行总线（Row_0_Output_1 到 Row_0_Output_3）中选择屏蔽电极信号源。可将每行输出路由到三个引脚中的任意一个。将行 LUT 函数设置为 A。

可能值为“使能”和“禁用”。

3.8 CSDADC 用户模块配置

CSDADC 组合了电容式感应和 ADC 功能。它通过重新使用 CSD 和 ADC 的共同模块来节省代码空间。当您设计同时需要电容式感应和 ADC 功能时，应该使用 CSDADC。仅用到其中一个功能或其它功能的应用程序则应使用 CSD 用户模块，或使用 ADC8 或 ADC10 用户模块。

CSDADC 用户模块有 4 个可选的时钟配置：PRS16、PWM8、PRS8 以及 VC2。这些配置为 CSD 前端开关式电容器电路使用不同的信号源。

当您在 PSoC Designer 项目中首次放置 CSDADC 用户模块时，应选择时钟配置。通过右键单击 CSDADC 用户模块并选择 **User Module Selection Options**（用户模块选择选项），可以更改此选择。

3.8.1 带有 PRS16 时钟源的 CSDADC

在这种配置中，16 位伪随机序列（PRS）用作 CSD 前端开关式电容器电路的信号源。系统时钟（IMO）是 PRS 的时钟源。PRS 扩展时钟的频谱，并提供优异的抗噪能力。这种配置需要 3 个数字模块。

3.8.2 CSDADC 带有 PRS8 时钟源

在这种配置中，8 位 PRS 用作 CSD 前端开关式电容器电路的信号源。系统时钟（IMO）是 PRS 的时钟源。PRS 扩展时钟频谱，并提供抗噪声功能。请注意，16 位 PRS 用于不带时钟预分频器配置的 CSD，可提供优异的抗噪功能。这种配置需要 3 个数字模块。

由于需要较低开关频率，因此在高 C_P 值的环境下工作时，应使用这种配置；有关 C_P 与开关频率之间的关系，请参见使用用户模块对 CapSense 性能进行调校部分。

3.8.3 带有 PWM8 时钟源的 CSDADC

在该配置中，IMO 由可调式预分频器分频，并用于 CSD 前端的开关式电容器电路。该配置对工作频率上的谐波及噪声比较敏感。该配置需要两个数字模块。

该配置适用于高电阻材料（例如：ITO）的感应系统，或适用于由于辐射问题而需要在较低工作频率下工作的情况。

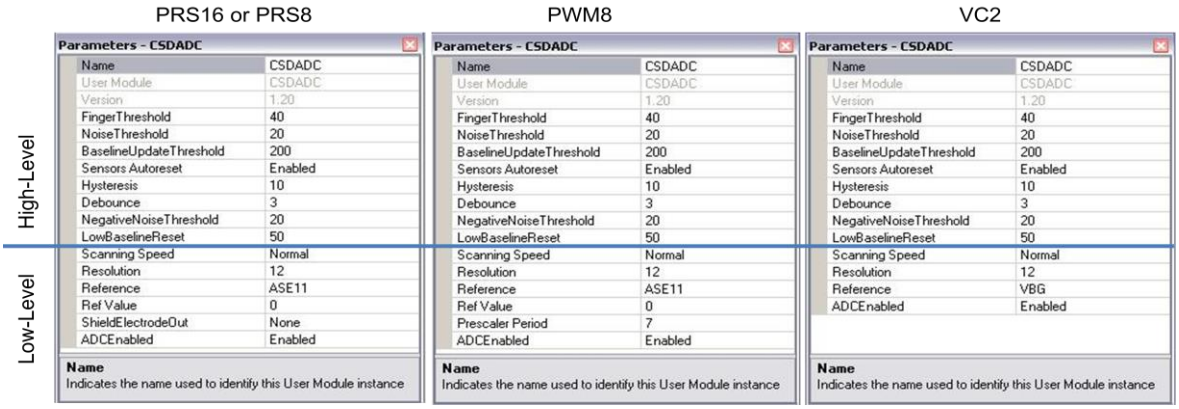
3.8.4 带有 VC2 时钟源的 CSDADC

在此配置中，系统时钟使用可调式预分频器 VC2 进行分频，并用于 CSD 前端的开关式电容器电路。该配置对工作频率上的谐波及噪声比较敏感。该配置需要一个数字模块。

该配置适用于高电阻材料（例如：ITO）的感应系统，或适用于由于辐射问题而需要在较低工作频率下工作的情况。

3.9 CSDADC 用户模块参数

图 3-7. PSoC Designer — CSDADC 参数窗口



3.10 低级参数

3.10.1 Scanning Speed（扫描速度）

该参数用于设置传感器扫描速度。虽然较快的扫描速度可以提供优化的响应时间，但较慢的扫描速度也具有下列优势：

- 提高信噪比
- 提供更好的抗电源和温度变化的能力
- 少量要求系统中断延迟；可以处理较长的中断

有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。可能值为超快、快速、正常和慢速。

3.10.2 Resolution（分辨率）

该参数确定扫描分辨率（单位为“位”）。N 位的扫描分辨率最大原始计数为 2^N–1。提高分辨率可以提高灵敏度和 SNR，但需要缩短扫描时间。有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。

该参数的值可能为 9 ~ 16 位。

3.10.3 Reference（参考电压）

要保证 CSD 正常工作，要求一个电压比较器的输入电压参考值。

可能的值为：

- VBG：派生于固定带隙参考的内部参考电压 1.3 V。
- ASE11：派生于 PWM 的内部可变参考电压。
- AnalogColumn_InputSelect_1：外部电压参考，如电阻分压器网络或外部滤波的 PWM/PRSPWM 信号。该参考电压可以连接到 CapSense 控制器引脚，并作为电压比较器参考源使用。

3.10.4 Ref 值

Ref 值的影响取决于 Reference 参数。当 Reference 参数被设置为 ASE11 时，Ref 值用来设置参考电压值。

该参数的取值范围为 0（¼ V_{DD} 最小值）到 8（¾ V_{DD} 最大值）。

提高 Ref 值时，灵敏度会降低，但这样会增加对屏蔽电极产生的影响。如果设计中传感器的电容有明显差异（传感器垫片的尺寸不同），则使用 API 函数来平衡原始计数，方法是为容值较大的传感器设置较高的 Ref 值。

当 Reference 参数被设置为 VBG 或 AnalogColumn_InputSelect_1 时，Ref 值不起任何作用。Ref 值不适用于带有 VC2 时钟源配置的 CSDADC。

3.10.5 Prescaler Period（预分频器周期）

预分频周期确定预充电开关输出频率。

取值范围为 1 到 255。建议值为 $2^N - 1$，这样可以获得最大信噪比（1、3、7、15、31、63、127 或 255）。在低分辨率和高扫描速度的情况下，其它值会导致更多噪声。

预分频器周期仅适用于带有预分频器的配置。

3.10.6 Shield Electrode Out（屏蔽电极输出）

可以从其中一个备用数字行总线（Row_0_Output_1 到 Row_0_Output_3）中选择屏蔽电极信号源。可将每行输出路由到三个引脚中的任意一个。将行 LUT 函数设置为 A。

可能值为“使能”和“禁用”。

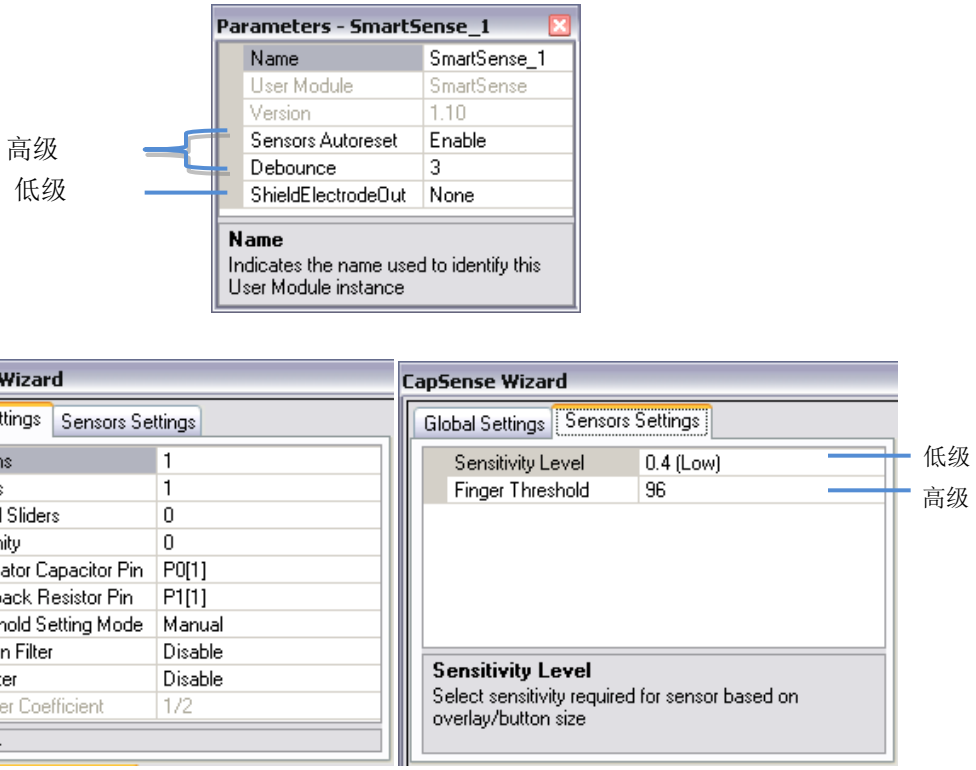
该参数仅适用于带有 PRS8/PRS16 时钟源配置的 CSDADC。在带有 PWM8 时钟源配置的 CSDADC 中，屏蔽电极信号固定连接到 Row_0_Output_0。

3.10.7 ADC Enabled（使能 ADC）

当该参数设置为 Enabled（使能）时，ADC 函数会被嵌入编译代码中，可以从用户代码中调用该程序。当该参数设置为 Disabled（禁用）时，将不包含 ADC 程序。但如果您在设计中不再需要 ADC，可以禁用该参数，以节省闪存。另外，CSDADC 用户模块的行为与 CSD 用户模块相同。

3.11 SmartSense 用户模块参数

图 3-8. PSoC Designer SmartSense 参数



3.12 低级参数

3.12.1 Shield Electrode Out（屏蔽电极输出）

屏蔽电极用于降低寄生电容。可将屏蔽电极信号路由到其中一个备用数字行总线（Row_0_Output_1 到 Row_0_Output_3）。然后，可将每行输出路由到一个容许引脚。将行 LUT 函数设置为 A。

3.12.2 Modulator Capacitor Pin（调制器电容引脚）

该参数设置引脚，以连接外部调制器电容（C_{MOD}）。从可用引脚 P0[1]、P0[3]中选择一个。

3.12.3 Feedback Resistor Pin（反馈电阻引脚）

该参数设置引脚，以将其连接至外部反馈电阻（R_b）。从可用的引脚（P1[1]、P1[5]和 P3[1]）中选择一个。在某些器件封装上，某些引脚不可用。使用引脚 P1[5]或 P3[1]以避免编程问题。

3.12.4 Threshold Setting Mode（阈值设置模式）

使用该参数选择自动阈值设置或手动阈值设置。自动阈值模式使用一种算法，它可以根据系统中发现的噪声来动态调整手指阈值。对于手动阈值模式，您可以提供一个固定的手指阈值。在配置 SmartSense 用户模块一节中介绍的是使用手动阈值设置的情况以及如何确定手指阈值的详细信息。

3.12.5 Sensitivity Level（灵敏度级别）

灵敏度用于增加或减少传感器发出的信号强度。灵敏度值越低（0.1 pF），传感器发出信号越强。为能够准确达到要求，设计的覆盖层越厚，所需的传感器信号也要越强。灵敏度选项有：高（0.1 pF）、中（0.2 pF）、低（0.4 pF）。

要产生更强的传感器信号（即灵敏度为“高”），SmartSense 用户模块需要消耗更多的传感器扫描时间。因此，传感器灵敏度等级设置为 0.1（高）与设置为 0.2 pF（中）相比，前者消耗更多的扫描时间。

3.12.6 Finger Threshold（手指阈值）

仅在将阈值设置模式设置为手动模式的情况下，该参数才可用。建议将该值设置为存储在 SmartSense_baSnSSignal[]数组中传感器信号的 80%。使用 I^2C 或 UART 通信协议，可以轻松监测该数组。

4. 使用用户模块对 CapSense 性能进行调校

理想的用户模块参数设置取决于电路板布局、按键尺寸、覆盖层材料和应用需求。有关这些因素的信息，请参考设计的注意事项。调校是确定最佳参数设置的过程，其目的为保证传感器操作的强健和可靠性。

4.1 基本注意事项

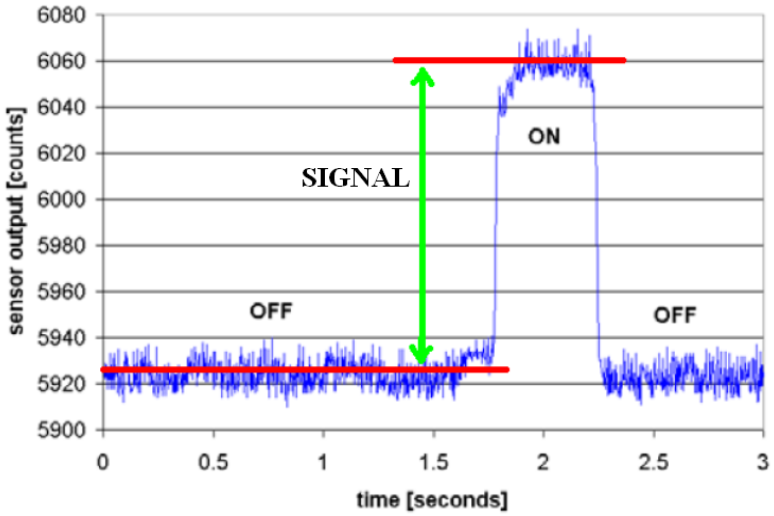
4.1.1 信号、噪声和信噪比

调校好的 CapSense 系统能够准确地识别传感器的 ON 和 OFF 状态。要实现该性能级别，CapSense 信号必须远远大于 CapSense 噪声。测量时比较信号与噪声，由此得出信噪比（SNR）。在讨论 CapSense SNR 的含义时，首先需要定义触摸感应环境中信号和噪声的概念。

4.1.1.1 CapSense 信号

CapSense 信号表示在手指触摸传感器时，传感器响应的变化，如图 4-1。传感器输出是一个表示传感器电容值的数字计数器值。在该示例中，在手指没有触摸传感器时，平均输出为 5925 次计数。当手指放在传感器上时，平均输出增至 6060 次计数。由于 CapSense 信号跟踪手指触摸导致的计数值变化，因此信号 = 6060 – 5925 = 135 次计数。

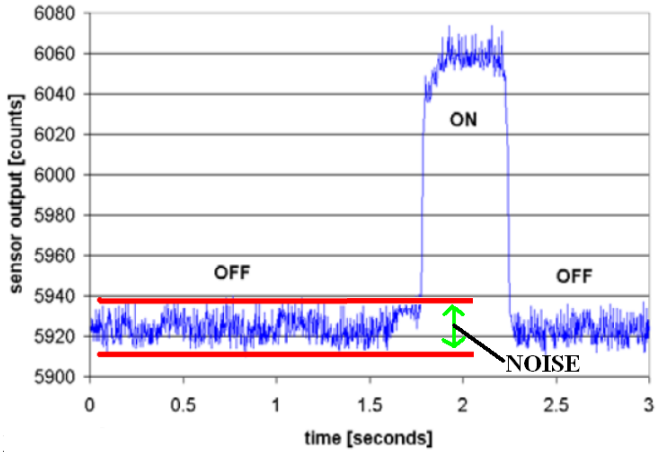
图 4-1. CapSense 信号的示例



4.1.1.2 CapSense Noise（CapSense 噪声）

CapSense 噪声是手指未触摸时传感器响应的峰-峰值变化，如图 4-2 所示。在该示例中，手指未触摸时输出波形跳跃限幅为 5912 和 5938 次计数。由于噪声为该波形最大计数与最小计数之差，所以噪声 = 5938 – 5912 = 26 次计数。

图 4-2. CapSense 噪声示例



4.1.1.3 CapSense SNR

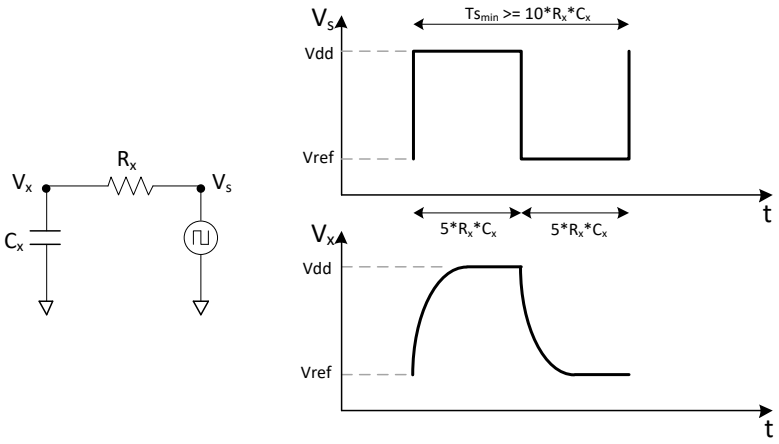
CapSense 信噪比是信号和噪声的简单比例。继续该示例，如果信号为 135 次计数，噪声为 26 次计数，那么 SNR 将为 135:26，这就使 SNR 降低到 5.2:1。CapSense 的最小建议 SNR 为 5:1，这表示信号是噪声的 5 倍。滤波器通常通过固件实现，以降低噪声。更多有关信息，请参见软件过滤。

4.1.2 充电/放电率

在调校过程中，要获得最大灵敏度，必须在每个周期内对传感器电容器进行完全充电和放电。充电/放电路径按照特定频率在两种状态的之间切换，该频率由用户模块中的时钟参数设置。

充电/放电路径包含串联电阻，用于降低电荷转移的速度。该电荷转移的变化率使用 RC 时间常数描绘（该时间常数是通过传感器电容和串联电阻计算得出），如图 4-3 所示。

图 4-3. 充电/放电波形



请确保将充电/放电率设置为与 RC 时间常数兼容的级别。基本原则容许每次切换的周期为 5RC，每个周期有两次切换（一次充电、一次放电）。通过以下公式，可以分别计算得出最小值周期和最大频率：

$$Ts_{min} = 10 \times R_X C_X$$ 公式 5

$$fs_{max} = \frac{1}{10 \times R_X C_X}$$ 公式 6

例如，假定串联电阻包含一个 560 Ω 的外部电阻和一个 800 Ω 的内部电阻，且传感器的电容为标准电容：

R_X = 1.4 kΩ

C_X= 24 pF

在本示例中，时间常数值和前端切换最大频率分别为：

Ts_{min} = 0.34 µs

fs_{max} = 3 MHz

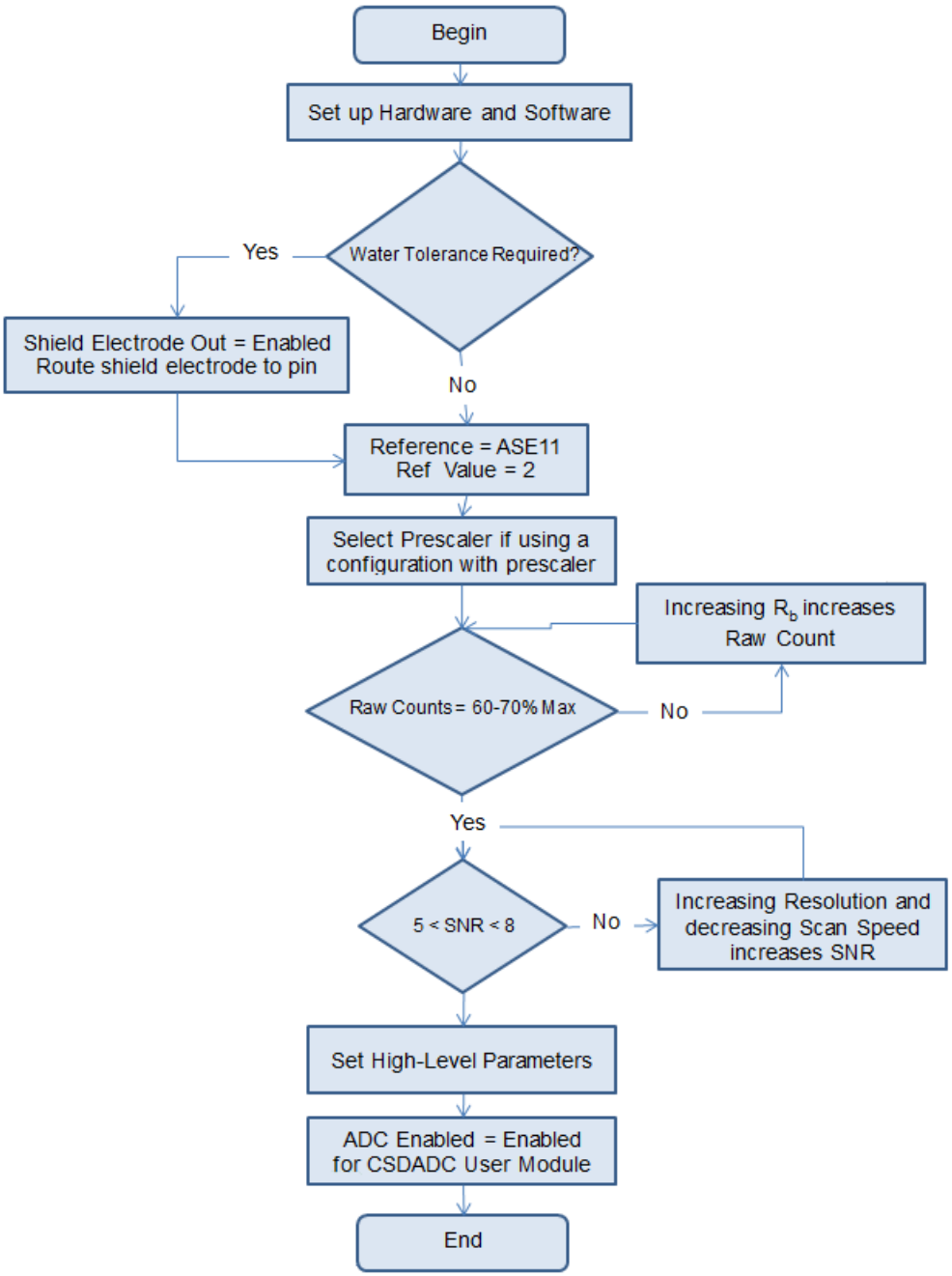
4.1.3 基准线更新阈值验证的重要性

温度和湿度都会引起平均计数随时间浮动。基准线是 CapSense 测量值的参考计数量，这对于环境影响补偿而言起着重要作用。高级决策（例如，手指触摸和手指离开状态）是基于基准线建立的参考级别。由于每个传感器具有相应的独特寄生电容，因此所有电容式传感器均有各自的基准线。

基准线以基准线更新阈值参数所设置的速率来跟踪计数变化。确保更新速度与预期应用相互匹配。如果更新速度过快，那么，基准线将补偿手指触摸产生的所有变量，并不予检测手指移动的情况。如果更新速度过慢，那么，相对慢的环境变化可能导致误触摸。开发期间，应对基准线更新阈值设置进行验证。

4.2 调校 CSD 和 CSDADC 用户模块

图 4-4. 调校 CSD 和 CSDADC 用户模块



4.2.1 设置硬件和软件，以进行调校

调校过程开始前，需要准备一些如概述一节所列出的硬件和软件工具。调校时，需要通过通信接口监测传感器的原始计数、基准线和计数差值。可以在此处下载使用 I²C 通信协议与个人计算机进行通信的代码示例。下载示例后：

1. 使用 PSoC Designer 打开代码示例。在 Workspace Explorer 中，右键单击 **CSD User Module**，然后选择 **CSD Wizard**。可以显示分配给引脚的 4 个 CapSense 按键。根据您的应用需求更改按键和引脚的分配数量。
2. 如有必要，可以更改调制器电容引脚和反馈电阻引脚。
3. 构建和编译项目。
4. 使用十六进制文件编程 CapSense 控制器。
5. 有关如何监测原始计数、基准线和计数差值的详细内容，请参见文档的代码示例中的“使用 USB-I2C 工具监测 CapSense 数据的说明”部分。

4.2.2 选择预分频器

要选择预分频器的值，请执行以下流程：

1. 使用 LCR 表或 CapSense 控制器测量传感器寄生电容的最大值。如果使用 LCR 表，请跳过第 5 步。
2. 使用带时钟预分频器的 CSD 用户模块和以下参数：
   a. 扫描速度 = 正常
   b. 分辨率 = 16
   c. 参考电压 = VBG
   d. 预分频器周期 = 15（SysClk = 24 MHz）、7（SysClk = 12 MHz） 或 3（SysClk = 6 MHz）
   e. 屏蔽电极输出 = 使能或禁用（具体情况根据应用要求）
3. 编译项目并将项目加载到 CapSense 控制器内。
4. 仅监测所有传感器（包括滑条段）的原始计数，并记录最高原始计数值。
5. 使用公式 7 计算寄生电容的最大值。

$$C_X = \frac{\text{Highest Raw Coun}}{88.85e12\times(V_{DD}-1.3)}$$ 公式 7

6. 使用公式 8 计算 fs_{max}。

$$fs_{max} = \frac{1}{10\times R_X C_X}$$ 公式 8

7. 根据时钟配置计算预分频器的值。
8. 针对带有时钟预分频器配置的 CSD：

$$\text{Prescaler} \geq \frac{\text{SysClk}}{fs_{max}\times 2} -$$ 公式 9

9. 针对带有 PWM8 时钟源配置的 CSDADC：

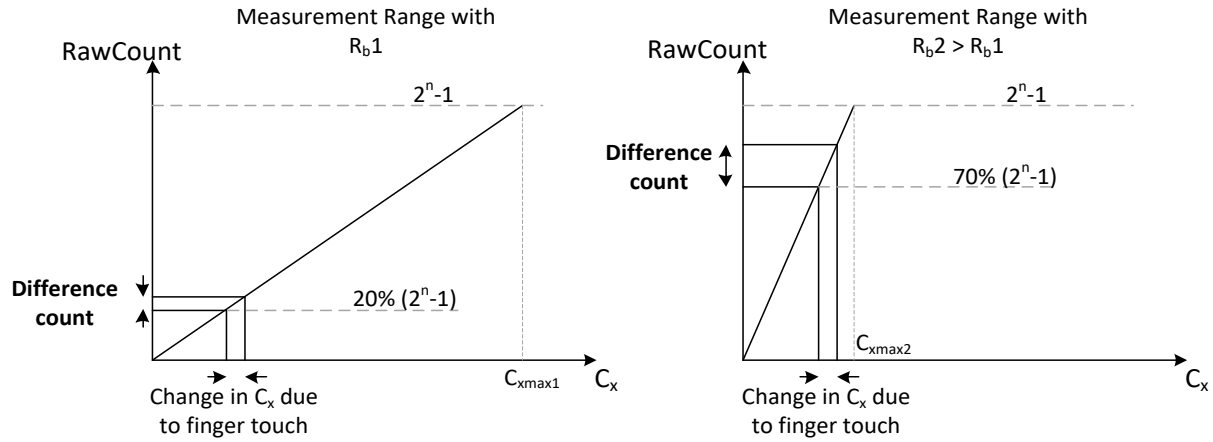
$$\text{Prescaler} \geq \frac{\text{SysClk}}{fs_{max}} -$$ 公式 10

10. 为了优化 SNR 性能，将预分频器的计算值取舍为最接近于 $2^n – 1$ 的值（1、3、7、15、31 等）。

4.2.3 通过 R_b 设置原始计数范围

泄露电阻的值 R_b 用于确定 CapSense 传感器的灵敏度。灵敏度是在触摸和释放这两个条件下分别测得的计数差值。增加 R_b，可以改善灵敏度并增加释放条件下的原始计数值。提高 R_b 还可以降低能被 CapSense 传感器测量的最大电容值。

图 4-5. CSD 测量范围



请查看图 4-5 中的两个原始计数测量值范围图。在左侧图形中，R_b = R_b1，系统可以测量到的电容值高达 C_{xmax1}。原始计数为其最大值的 20%。由手指触摸产生的计数差值是非常小，并且灵敏度也低。在右侧图形中，泄露电阻升高，R_b2 > R_b1。这时，原始计数为该最大值的 70%。在这种情况下，计数差值远远大于第一种情况的计数差值，这样会增加灵敏度。

R_b 的典型值介于 500 Ω 和 10 kΩ 之间。为了查找最佳电阻：

1. 从 R_b 2 kΩ 电阻开始。
2. 监测在所有传感器上测量到的原始计数。
3. 调整 R_b，直到原始计数达到选定扫描分辨率全量程范围的 70%为止。例如，如果选择 9 位分辨率，则全量程范围为 511 次计数。调整 R_b，直到原始计数为 358 次计数为止。

设计多个传感器时，可能很难将所有原始计数精确地设置到 70%。在这种情况下，可以将所有原始计数设置为其最大值的 60 - 70%之间。

4.2.4 设置高级参数

遵循高级参数建议部分给出的建议事项。

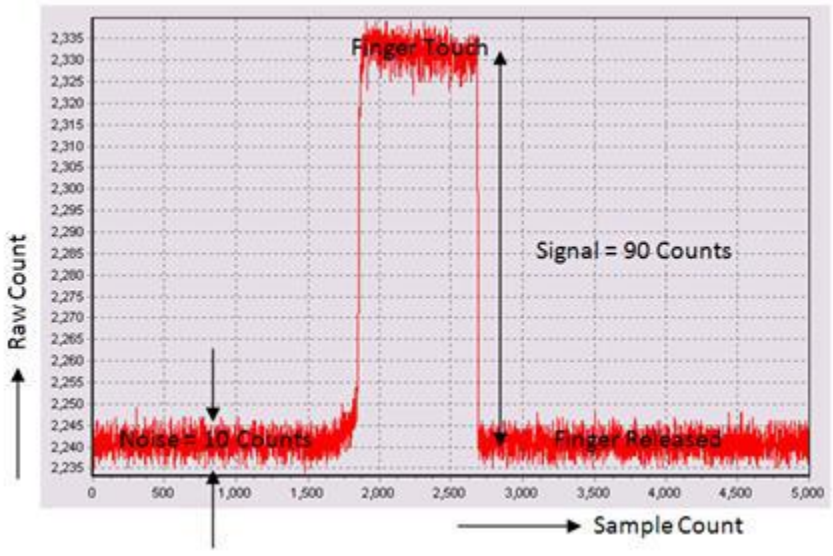
4.3 配置 SmartSense 用户模块

下面是在典型的 CapSense 应用中配置 SmartSense 用户模块的基本指南：

1. 准备目标电路板。组装目标应用 PCB，并在它上面固定覆盖层。使用非导电胶水或特殊胶带来粘贴覆盖层。避免在 PCB 与覆盖层之间留有空气间隙，否则会明显降低灵敏度，且由于您触摸时气隙会漂移，导致出现许多按键误触发情况。
2. 设置实时数据图工具，例如，Multi-Chart（参见 CapSense 数据查看工具一节），以便监测一个或多个数据系列。在调校过程中，必须观察传感器的原始计数（waSnsResult）、基准线（waSnsBaseline）、计数差值（waSnsDiff）、信号（baSnsSignal）和按键手指阈值（baBtnFThreshold）。使用 UART-USB 桥接器或 I2C-USB 桥接器硬件并通过 USB 端口来构建 CapSense 控制器与个人计算机之间的接口。请勿使用 LCD 或其它任何数字显示屏来监测数据，因为这些工具运行较慢，无法跟踪数据的变化。

   注意：使用 SmartSense 和 EzI2C 时，不能将 P1[1]连接到 R_b，因为它被分配给 SCL。
3. 开始时的默认配置：
   - 设置 IMO = 24 MHz、CPU_Clock = SysClk/1
   - 在 IMO = 24 MHz 的情况下，放置 SmartSense 用户模块
   - 禁用 **Sensor Autoreset**。将 **Debounce** 设置为 3，并将 **Shield Electrode Out** 设置为‘None’。
   - 在 CapSense 向导的 Global Settings 下方，在“Button”类别下分配一个传感器。然后选择调制器电容和反馈电阻引脚。将 **Threshold Setting Mode** 设置为 Manual，并禁用中值和 IIR 滤波器。在电路板上，确保 R_b = 15 KΩ，C_{mod} = 10 nF。
   - 在 CapSense 向导的 Sensor Settings 选项卡下，将 **Sensitivity Level** 设置为 Low，**Finger Threshold** 设置为 96。
4. 使用用户模块数据手册中提供的示例固件代码生成/编译项目。
5. 监测传感器原始计数，并计算 SNR。图 4-6 显示的是所监测的传感器原始计数。根据 CapSense 最佳实践，强大可靠设计的 SNR 必须大于 5。在这种情况下，监测 SNR 的值为 9。如果 SNR 的值小于 5，则阅读第 7 步中的 SNR 改进技术。

图 4-6. 所监测的传感器原始计数

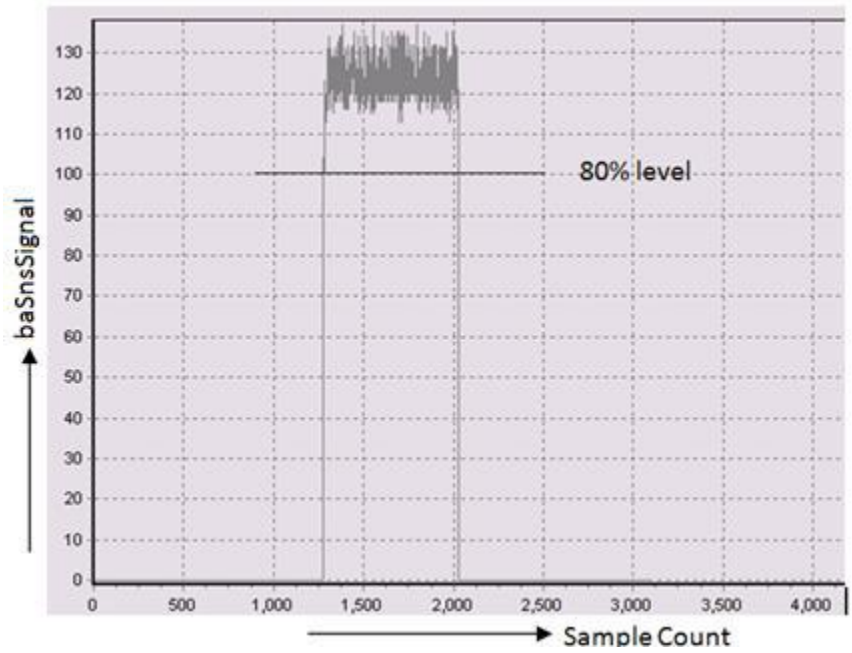


6. 如果应用环境不要求精确控制手指阈值，或者选择了自动阈值设置模式，则可以跳过该步骤。自动阈值模式使用一种算法，它可以根据系统中发现的噪声来动态调整手指阈值。在以下情况中，应在自动阈值模式上选择手动模式：

- 如果 RAM/ROM 封装小于它要支持的传感器数量。
- 要求精确调整传感器手指阈值的设计。
- 需要在嘈杂环境下运行的设计。

如果使用手动阈值设置模式，还需要设置好手指阈值，才能完成调校过程。监测传感器信号 baSnsSignal。确定感应器信号最大值的 80%，该值便是手指阈值。在该示例中，该信号接近 100 次计数，如图 4-7 所示。

图 4-7. 监测传感器信号 baSnsSignal



此时，打开 CapSense 向导，在 Sensor Settings 选项卡下，将手指阈值设置为所计算的值（在本示例中，该值为 100，如图 4-7 所示）。

7. 重复上一步，完成 SmartSense 传感器调校过程。然而，第 5 步计算的 SNR 结果可能小于 5:1，例如，当使用的覆盖层较厚时，会出现这种情况。在这种情况下，请遵循这些建议：
   - 将灵敏度级别从‘Low’（低级）改为‘Medium’（中级）。如果 SNR 仍未达到 5:1，则将灵敏度级别更改为‘High’（高级）。
   - 有时，由于系统中观察的噪声冗余，因此 SNR 无法达到 5:1。在这些情况下，应使用软件滤波器。使用 IIR 滤波器，将比例系数调为 1/2。如果还未降低噪声，则将 IIR 的比例系数改为 1/4。

5. 设计的注意事项

在为您的应用设计电容式触摸感应技术时，一定要记住 CapSense 器件是更大设计框架里的一部分。认真考虑从 PCB 布局到用户界面到最终使用操作环境的各个细节层次，可形成强大且可靠的系统性能。有关 EMC 注意事项的更多信息，请参见 CapSense 入门一节。

5.1 覆盖层选择

在 CapSense 基本原理一节中，使用公式 1 计算手指电容，如下所示：

$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{D}$$

其中：

ε_0 = 空气介电常数

ε_r = 覆盖层的介电常数

A = 手指与传感器垫片覆盖层的接触面积

D = 覆盖层的厚度

若要增大 CapSense 信号强度，请选择具有较高介电常数的覆盖层材料，减小覆盖层厚度，并增大按键直径。

表 5-1. 覆盖层材料介电强度

材料	击穿电压（V/mm）	12 kV 时的覆盖层最小厚度（mm）
空气	1200 – 2800	10
干木材	3900	3
普通玻璃	7900	1.5
硼硅酸盐玻璃（Pyrex®））	13,000	0.9
PMMA 塑料（Plexiglas®）	13,000	0.9
ABS	16,000	0.8
聚碳酸酯（Lexan®）	16,000	0.8
福米卡	18,000	0.7
FR-4	28,000	0.4
PET 薄膜（Mylar®）	280,000	0.04
聚酰亚胺薄膜（Kapton®）	290,000	0.04

导电材料不能用做覆盖层，因为它会与电场模式相干扰。出于该原因，覆盖层不能使用含金属颗粒的油漆。

使用粘合剂将覆盖层粘贴到 CapSense PCB 上。3M™有一种产品号为 200MP 的透明丙烯酸粘合膜，可用于 CapSense 应用。这种特殊的粘合剂是从纸作胶带卷抽取（3M™编号为 467MP 和 468MP 的产品）的。

5.2 ESD 保护

仔细的系统设计会自然提供稳健的 ESD 耐受性。考虑到在您的产品（特别是用户界面产品）中发生的接触放电方式，可能需要抵受 18 kV 的放电事件，并且不能对 CapSense 控制器产生任何损害。

CapSense 控制器引脚可承受 2 kV 的直接电流。在大多数情况下，覆盖层材料能为控制器引脚提供充分的 ESD 保护。表 5-1 列出了使 CapSense 传感器避免经受 12 kV 放电所需的各种覆盖层材料厚度（按照 IEC 61000-4-2 的规定）。如果覆盖层材料未提供充分的 ESD 保护，则按照以下的顺序采取防静电措施：预防、重定向、钳制。

5.2.1 预防

确保触摸表面上所有路径的击穿电压均大于潜在高电压接触。此外，系统设计需要确保 CapSense 控制器和 ESD 源之间保持适当的距离。如果无法保持足够的距离，可在 ESD 源和 CapSense 控制器之间放置一个高击穿电压的保护层。Kapton®胶带厚度为 5 mil 的每一层可承受 18 kV 电压。

5.2.2 重定向

如果您的产品空间密集，则可能无法防止放电事件。在这种情况下，您可以通过控制放电发生源来保护 CapSense 控制器。通常是在连接到机壳接地的电路板周边放置一个保护环。按照 PCB 布局指南中的建议，在按键或滑条传感器周围提供一个网格接地层可重定向 ESD 事件，使其远离传感器和 CapSense 控制器。

5.2.3 钳制

因为是有意将 CapSense 传感器放置在接近于触摸表面的位置，所以重定向放电路径可能不太切合实际。在这种情况下，合理的做法是添加串联电阻或者专用 ESD 保护器件。

建议使用 560 Ω 的串联电阻。

更有效的方法是在易受影响的走线上提供特殊用途 ESD 保护器件。CapSense 的 ESD 保护器件必须是低电容的。表 5-2 列出了适用于 CapSense 控制器的建议器件。

表 5-2. 建议用于 CapSense 的低电容 ESD 保护器件

ESD 保护器件		输入电容	漏电流	接触放电的最大限制	空气放电的最大限制
制造商	器件型号				
Littelfuse（力特）	SP723	5 pF	2 nA	8 kV	15 kV
Vishay	VBUS05L1-DD1	0.3 pF	0.1 µA <	±15 kV	±16 kV
NXP	NUP1301	0.75 pF	30 nA	8 kV	15 kV

5.3 电磁兼容性（EMC）的注意事项

5.3.1 辐射干扰

辐射电能可能会影响系统测量并可能会影响处理器内核的运行。干扰可以从 PCB 层次进入 CapSense 控制器，通过 CapSense 传感器走线，和通过其它所有数字或模拟输入。最大限度减小射频干扰影响的布局指南：

- **接地层**：在 PCB 上使用一个接地层。
- **串联电阻**：在 CapSense 控制器引脚的 10 mm 范围内安装串联电阻。
  - 建议为 CapSense 输入线使用 560 Ω 的串联电阻。
  - 建议为通信线（I^2C 和 SPI）使用 330 Ω 的串联电阻。
- **走线长度**：要尽可能地缩短走线长度。
- **电流环路区域**：尽量减少电流的返回路径。在传感器和走线间 1 厘米范围内填充网格接地层（不是实心接地层），以减小寄生电容的影响。
- **RF 源位置**：使系统和带有噪声源（如 LCD 反相器和开关电源（SMPS））离开，以便保证干扰与 CapSense 输入相分隔。电源屏蔽是另一种防止干扰的通用方法。

5.3.2 辐射

想要降低来自 CapSense 传感器的辐射，请给开关式电容时钟选择较低频率。通过使用预分频器选项，固件可控制该时钟。增加预分频器的值可以降低时钟的开关频率。

5.3.3 抗传导干扰和辐射

通过与其它系统的互连而进入系统的噪声称为传导噪声。这些互连包括电源和通信线。因为 CapSense 控制器是低功耗器件，所以必须避免传导辐射。以下指南可帮助减少传导辐射和干扰：

- 按照数据手册的建议使用去耦电容器。
- 在系统电源的输入上添加双向滤波器。这对于传导辐射和干扰都有效。PI 型滤波器可以防止电源噪声影响敏感器件，同时防止器件的开关噪声反耦合到电源层。
- 如果 CapSense 控制器 PCB 通过线缆连接到电源，请尽量减小线缆长度并考虑使用屏蔽线缆。
- 您可以在电源或通信线路周围放置一个铁氧体磁珠以用来过滤高频噪声。

5.4 软件过滤

软件滤波器是处理系统高频噪声的技术之一。表 5-3 列出了对 CapSense 有用的滤波器类型。

表 5-3. CapSense 滤波器表

类型	说明	应用
均值	具有同样的加权系数的有限脉冲响应滤波器（无反馈路径）	来自电源的周期性噪声
IIR	具有与 RC 滤波器类似的阶跃响应的无限脉冲响应滤波器（有反馈路径）	高频白噪声（1/f 噪声）
中值	从大小为 N 的缓冲区计算中值输入值的非线性滤波器	来自电机和开关电源的噪声毛刺
抖动	根据之前输入来限制当前输入的非线性滤波器	来自厚覆盖层的噪声（SNR < 5:1），对滑条中心数据非常有用。
基于事件的滤波器	对传感器数据中观察到的模式发出预定义响应的非线性滤波器	通常在非触摸事件中使用，以阻止 CapSense 数据传输
基于规则的滤波器	对传感器数据中观察到的模式进行预定义响应的非线性滤波器	通常在触摸表面正常操作过程中使用，以响应特殊情况，例如，意外选择多个按键

可以在这里下载使用所有这些滤波器的代码示例。

表 5-4 详细说明了不同软件滤波方式对 RAM 和闪存的要求。每种滤波器所需的闪存数量取决于编译器的性能。这里列出的要求适用于 ImageCraft 编译器和 ImageCraft 专业版编译器

表 5-4. RAM 和闪存要求

滤波器类型	滤波器的阶位	RAM（每个传感器的字节）	闪存（字节）ImageCraft 编译器	闪存（字节）ImageCraft Pro 编译器
均值	2-8	6	675	665
IIR	1	2	429	412
	2	6	767	622
中值	3	6	516	450
	5	10	516	450
用于原始计数的抖动滤波器	N/A	2	277	250
用于滑条中心的抖动滤波器	N/A	2	131	109

可以在此处下载使用所有这些滤波器的代码示例。

5.5 功耗

5.5.1 系统设计建议

对于许多设计而言，最小化功耗是重要目标。有多种方法可降低 CapSense 电容式触摸感应系统的功耗。

- 将 GPIO 驱动模式设置为低功耗
- 关闭高功耗模块
- 优化 CPU 速度以降低功耗
- 在较低 V_{DD} 下操作

除了这些方法外，睡眠扫描方法也非常有效。

5.5.2 睡眠-扫描方法

在各典型应用中，CapSense 控制器不需要经常保持活动状态。可以使器件进入睡眠状态，从而使器件的 CPU 和主要模块停止运行。在睡眠状态下，器件所消耗的电流远远低于有效电流。

使用以下公式可计算器件在较长周期中所消耗的平均电流。

$$I_{AVE} = \frac{(I_{Act} \times t_{Act}) + (I_{Slp} \times t_{Slp})}{T}$$ 公式 11

其中：

I_{ACT} = 有功电流

T_{ACT} = 活动时间

I_{Slp} = 睡眠电流

t_{Slp} = 睡眠时间

T = 总计时间周期

器件的平均功耗可按以下公式计算：

$$P_{AVE} = V_{DD} \times I_{AVE}$$ 公式 12

其中：

V_{DD} = 电源电压

I_{AVE} = 平均电流

基于 Excel 的平均功耗计算器可用。单击此处下载计算器。

5.5.3 响应时间与功耗

如公式 12 所述，可以通过降低 I_{AVE} 或 V_{DD} 来降低平均功耗。增加睡眠时间可减少 I_{AVE}。延长睡眠时间到较高值将导致 CapSense 按键响应时间差。因此，睡眠时间的设置必须根据系统需求。

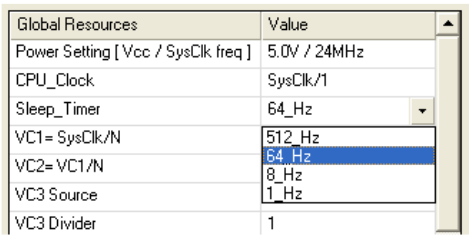
在任何应用中，如果功耗和响应时间均被视为重要参数，则可以使用优化方法，即合并持续性扫描和睡眠扫描模式。使用这种方法时，器件大部份时间在睡眠扫描模式下：器件扫描传感器，然后定期进入睡眠模式，因此功耗较少（如上一节所述）。当用户触摸传感器以操作系统时，则器件跳至持续扫描模式: 传感器被连续扫描而不进入睡眠模式，提供较佳响应时间。在指定的超时期间内，器件仍然处于持续扫描模式。如果在该超时周期内，用户未触摸任何传感器，则器件会切换回睡眠-扫描模式。

5.5.4 测量平均功耗

下列指南介绍了使用睡眠-扫描方法时确定平均功耗的各个步骤：

1. 编译一个扫描所有传感器而不进入睡眠模式（持续扫描模式）的项目。扫描传感器前，代码需要具备引脚切换功能。输出引脚的状态切换可当做时间标记，该时间标记可通过示波器跟踪。
2. 将项目加载到 CapSense 器件中，并测量电流消耗。将所测量的电流分配给 I_{ACT}。
3. 从数据手册中获取睡眠电流信息，然后将其分配到 I_{Slp}。
4. 使用示波器监测输出引脚切换，并测量两次切换之间的周期。这样会得到活动时间。将该值分配到 t_{Act}。
5. 对项目应用睡眠扫描周期。通过在全局资源窗口中选择睡眠定时器频率来设置睡眠扫描周期的时间 T，如图 5-1 所示。
6. 从睡眠扫描周期时间中减去活动时间，得出睡眠时间，$t_{Slp} = T - t_{Act}$。
7. 使用公式 11 计算平均电流。
8. 使用公式 12 计算平均功耗。

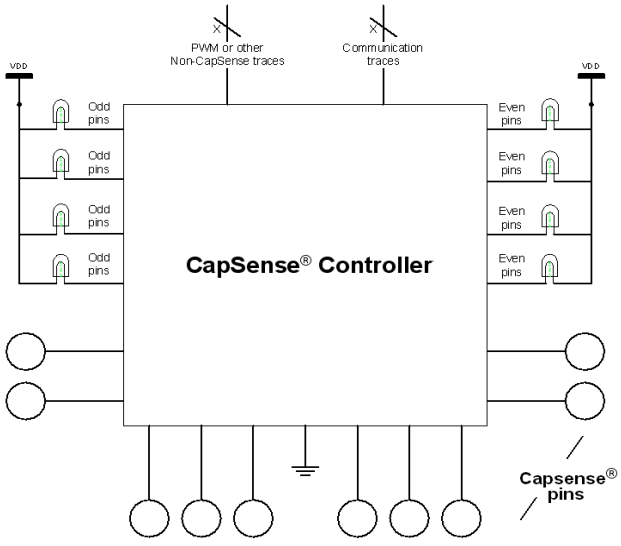
图 5-1. 全局资源窗口



5.6 引脚分配

减少 CapSense 传感器走线和通信线以及非 CapSense 走线之间相互作用的一种有效方法是使用端口分配将它们隔离开。图 5-2 显示了针对 32-QFN 封装进行此隔离的通用示例。由于各个功能已被隔离，因此 CapSense 控制器得到定向，为了确保通信、LED 和感应走线之间不存在交叉。

图 5-2. 推荐的：通信、CapSense 和 LED 的端口隔离



CapSense 控制器架构对偶数和奇数端口引脚数需要电流预算。如果奇数端口引脚的电流预算为 100 mA，那么所有奇数端口引脚所抽取的总电流不能超过 100 mA。除电流预算限制外，每个端口引脚还有最大的电流限制，相关定义见 CapSense 控制器数据手册。

所有 CapSense 控制器都提供高灌电流和拉电流的端口引脚。使用来自端口引脚的高灌电流或拉电流时，应使用距离器件接地引脚最近的端口，以最小化噪声。

5.7 PCB 布局指南

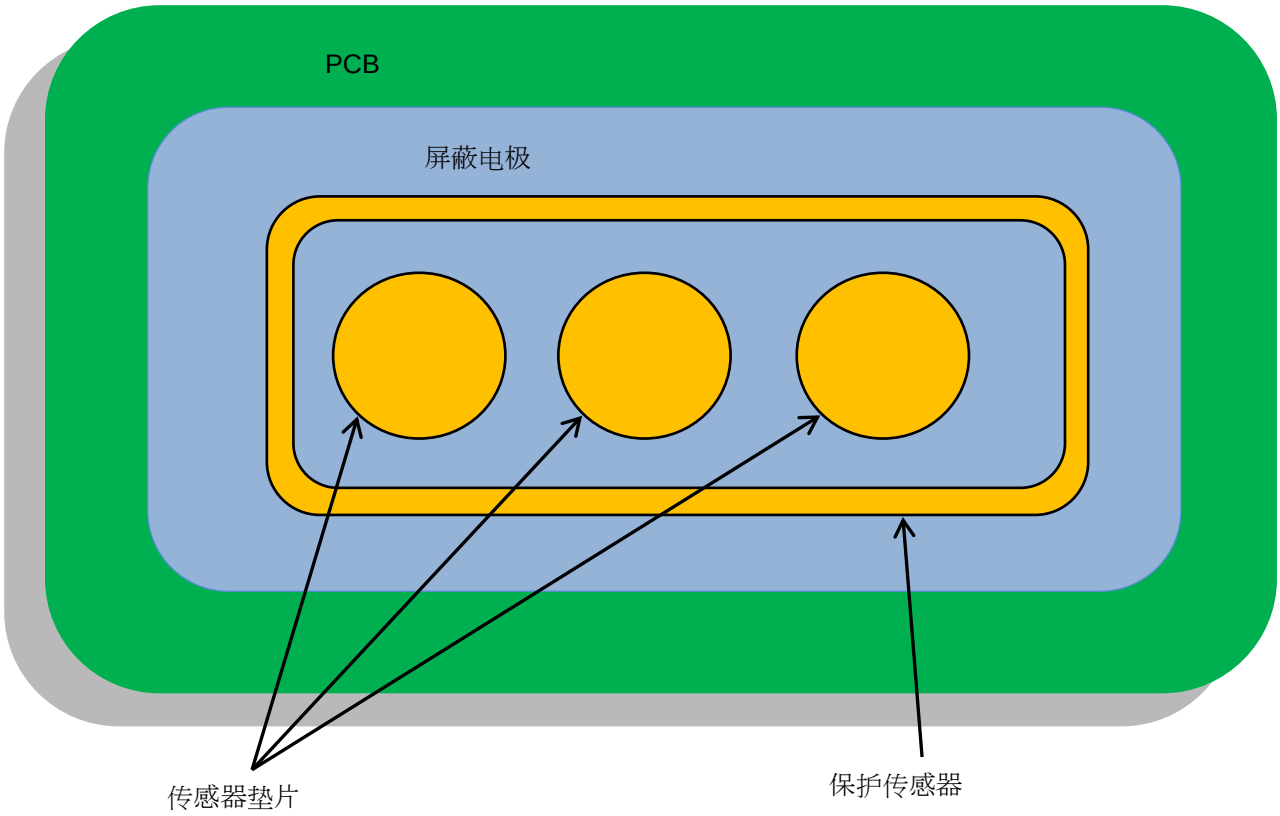
CapSense 入门手册中提供了详细的 PCB 布局指南。

6. 防水功能

某些 CapSense 电容式触摸感应应用需要在带有水份环境进行可靠的工作。白色家电、汽车应用和工业应用系统均在含水、冰和湿度变化的环境下运行。对于这样的应用，屏蔽电极和保护传感器可以提供稳健的触摸感应功能。

6.1 屏蔽电极和保护传感器

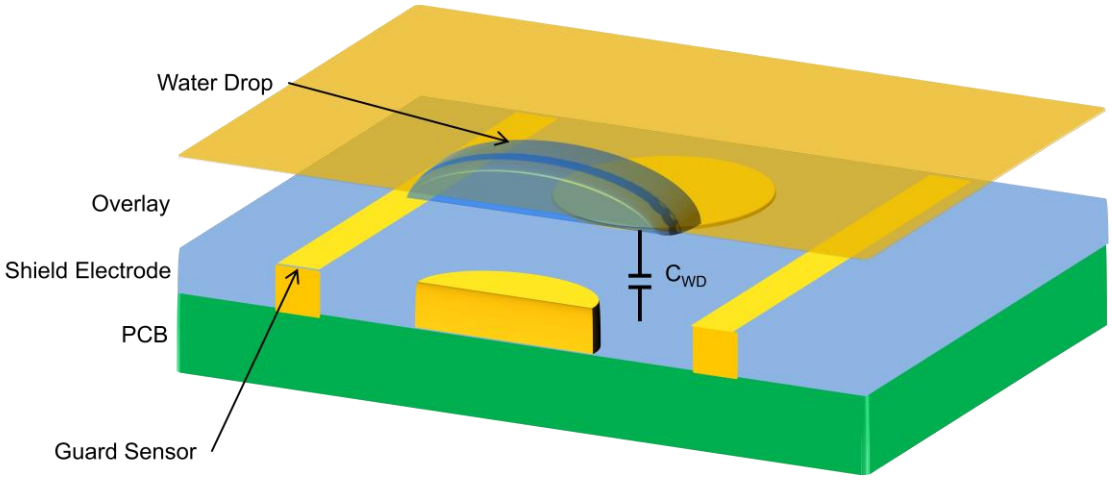
图 6-1. 使用屏蔽电极和保护传感器的 PCB 布局



6.1.1 屏蔽电极

屏蔽电极防止 CapSense 按键传感器检测由水滴引起的误触摸。当水滴落到覆盖层表面上时， 屏蔽电极和传感器垫片之间的耦合增加了 C_{WD}，如图 6-2 所示。

图 6-2. 带水滴的电容测量



- C_{WD} — 水滴与屏蔽电极之间的电容

屏蔽电极的目的是在触摸传感器周围建立电场，用来消弱水滴产生的影响。屏蔽电极的工作方式是对照屏蔽电极上的触摸传感器的电压。

请遵循以下准则以确保屏蔽电极正常工作：

- 原理图
- 布局
- 固件开发

6.1.1.1 原理图

选择正确的引脚以驱动屏蔽电极输出信号。请勿使用以下引脚来驱动屏蔽电极输出信号。

- 调制器电容引脚（C_{MOD}）：P0[1]或 P0[3]
- 反馈电阻引脚（R_B）：P1[1]或 P1[5]
- 其它端口引脚：P0[0]、P0[4]、P1[0]、P1[4]、P2[0]、P2[4]和 P3[0]

6.1.1.2 布局

遵循 CapSense 入门中提供的布局指南。

6.1.1.3 固件开发

第一步：可以从其中一个备用数字行总线（Row_0_Output_1 到 Row_0_Output_3）中选择屏蔽电极信号源。

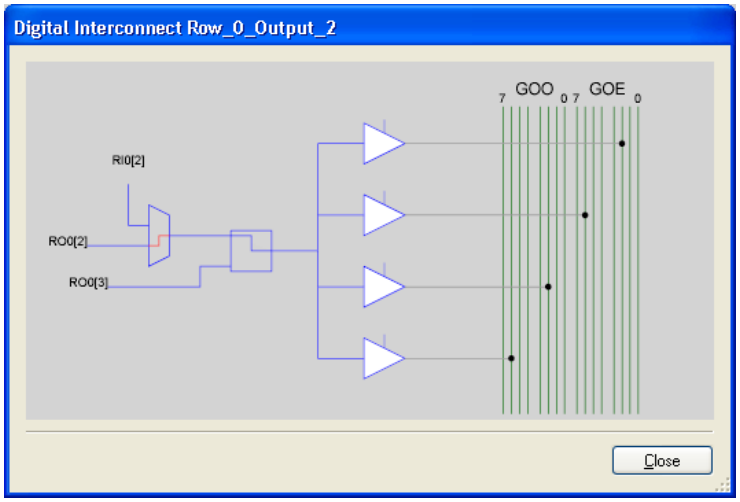
- 如果屏蔽引脚是 P0[5]、P1[5]、P2[1]、P2[5]，则选择 Row_0_Output_1
- 如果屏蔽引脚是 P0[2]、P0[6]、P1[2]、P1[6]、P2[2]、P2[6]、P3[2]，则选择 Row_0_Output_2
- 如果屏蔽引脚是 P0[7]、P1[3]、P1[7]、P2[3]、P2[7]，则选择 Row_0_Output_3

第二步：将屏蔽电极输出路由到屏蔽引脚。执行下列步骤：

1. 选择屏蔽电极输出。
2. 将行输出信号路由到全局输出奇数/偶数引脚。
3. 将全局输出奇数/偶数引脚连接到端口引脚上。例如，如果将 P0[2]选择为屏蔽引脚，则从用户模块属性中在 Row_0_Output_2 上选择屏蔽电极输出。

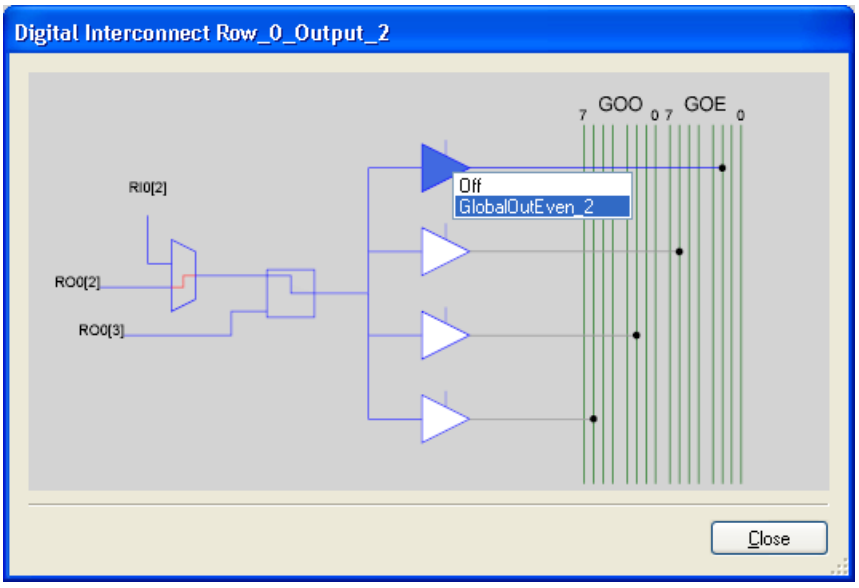
4. 单击 **Row_0_Output_2**，打开数字互连视图，如图 6-3 所示。

图 6-3. 数字互连原理图



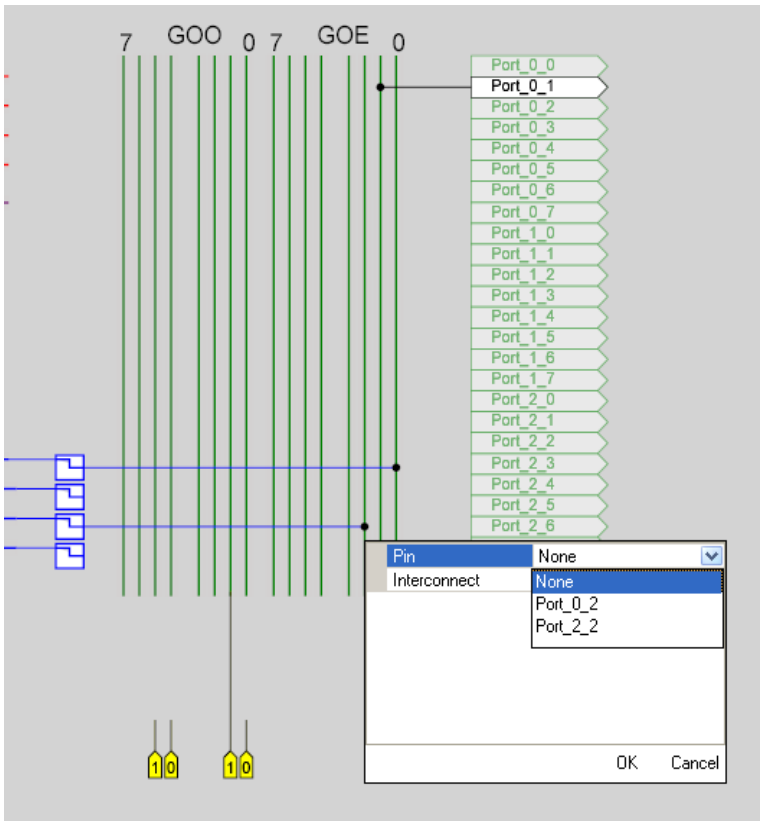
5. 选择 **Row_0_Output_2_Drive_0**，然后再选择 **GlobalOutEven_2**，如图 6-4 所示。

图 6-4. 输出选择原理图



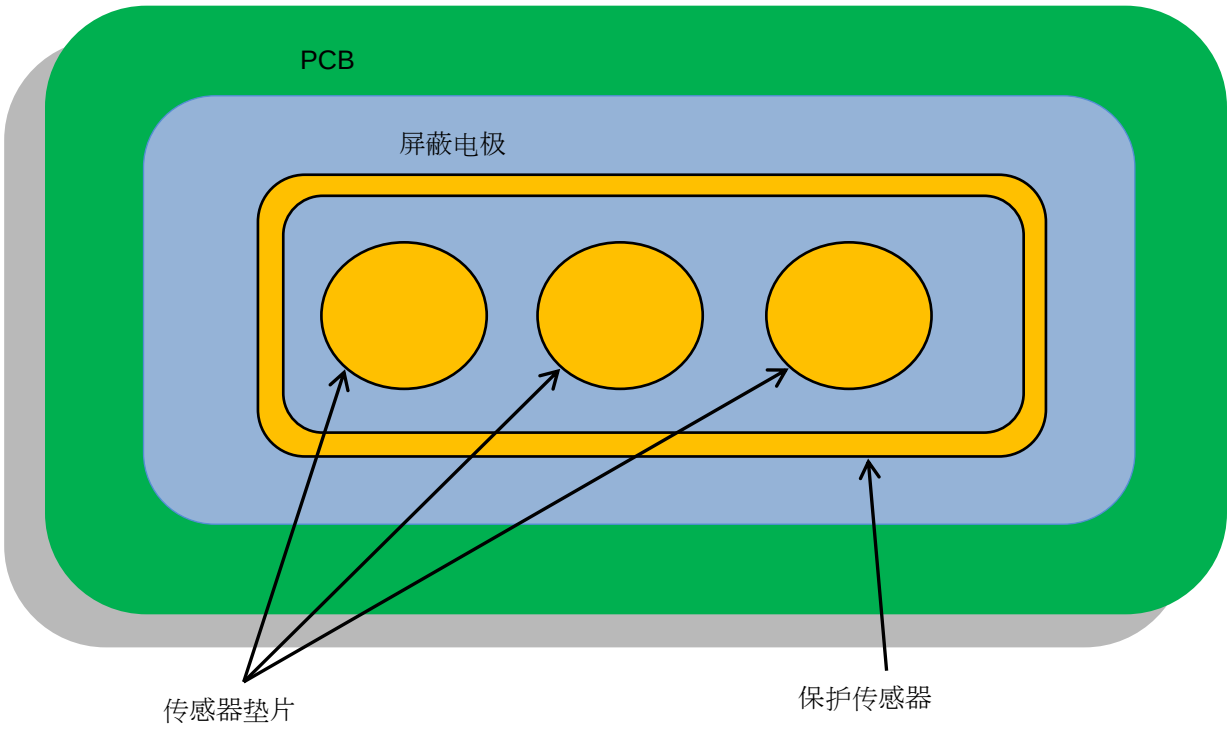
6. 点击 **GlobalOutEven_2**，然后在‘Pin’选项中选择 **P0[2]**，如下图所示。

图 6-5. 使用屏蔽和保护传感器的 PCB



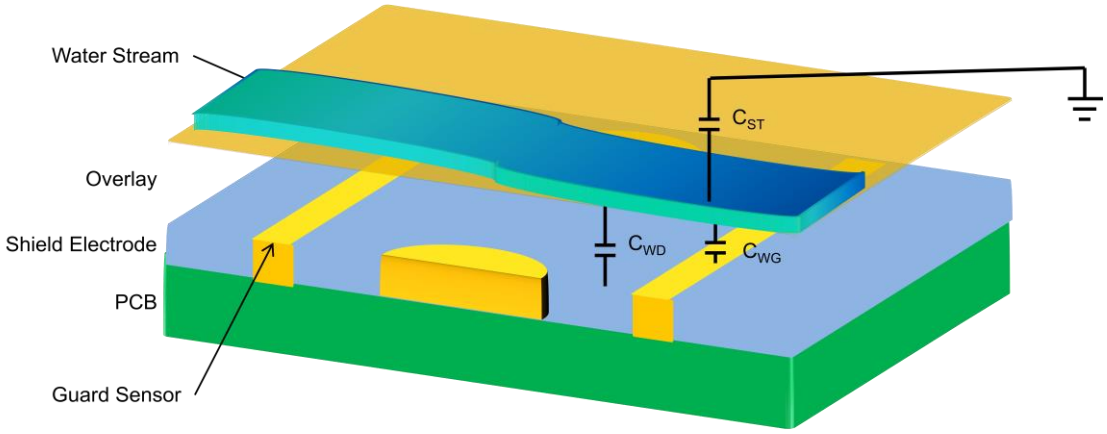
6.1.2 保护传感器

图 6-6. 使用屏蔽和保护传感器的 PCB



保护传感器是铜走线，如图 6-6 所示，该走线环绕 PCB 上的所有传感器，用来检测是否存在水流。当在感应表面存在水流时，在系统中将添加较大的电容 C_{ST}，如图 6-7 所示。该电容可能比 C_{WD} 大数倍。因此，屏蔽电极的影响被完全屏蔽，传感器所测量的原始计数将等于，甚至高于手指触值。在这种情况下，应该使用保护传感器；当检测到水流时，该传感器将防止其它传感器触发。

图 6-7. 带水流时的电容测量



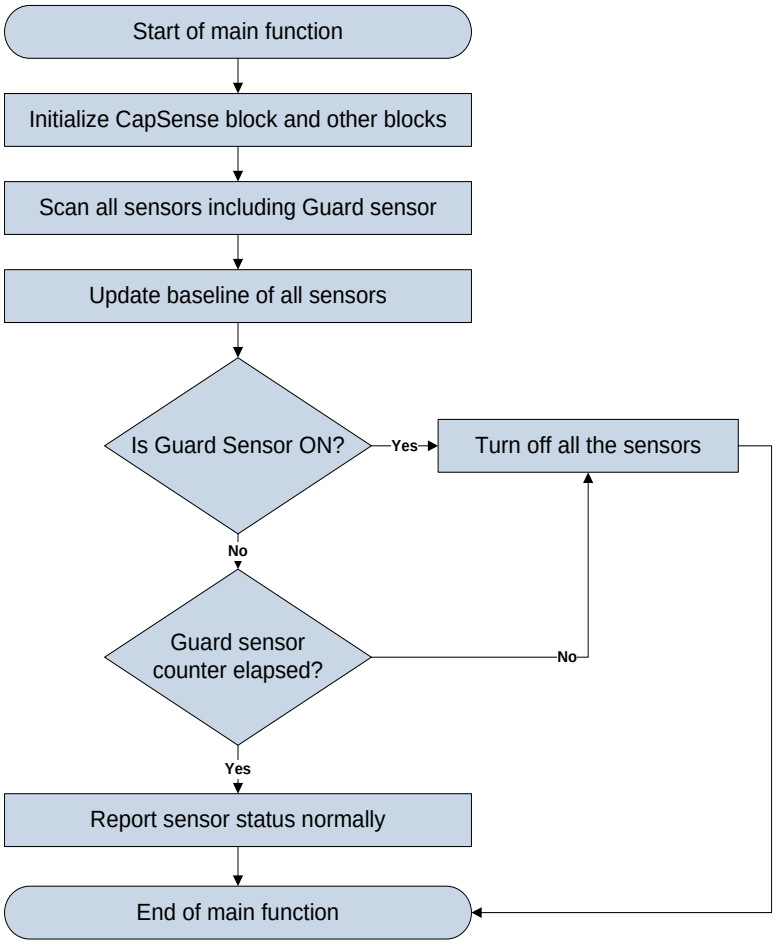
- C_{WD} — 水流与屏蔽电极之间的电容。
- C_{ST} — 水流与系统地之间的电容。
- C_{WG} — 水流与保护传感器之间的电容。

保护传感器应在固件中实施。实施保护传感器需要一个 CapSense 引脚和一个计数器（硬件/软件）。

当水流进入触摸板时，保护传感器检测到这个事件，然后禁用触摸传感器的处理逻辑。此外，保护传感器的“死”时间阻止传感器被过早解锁。水流消失时，保护计数器会在短时间暂时抑制触摸处理系统（保护传感器计数器）。这样会消除了遗留在触摸板上的水流触发的假触摸检测。

图 6-8 是显示如何在固件中实现保护传感器的流程图。

图 6-8. 实施保护传感器的流程图



6.2 设计建议

以下系统级和 PCB 布局建议适用于暴露于水份的 CapSense 系统。

- 屏蔽电极铜填充建议：
  - 顶层 — 走线宽度为 7 mil，栅格为 45 mil（填充 15%）
  - 底层 — 走线宽度为 7 mil，格子为 70 mil（填充 10%）
  - 按键之间的屏蔽电极应至少为 10 mm 宽
  - 只有传感器垫片和 CapSense 控制器周围的区域才应该填充地
- 传感器表面垂直放置，或与平面成一个角度，这样，水滴自然滑离传感器，而不会积存大量水滴。
- 使用防水和不吸水的覆盖层材料。这样可以最大程度地减少器件屏幕上的水纹和薄膜。这种材料特别适合具有较高导电性液体的情况，例如，海水。
- 在应用存在水流影响时，必须使用保护传感器。如果器件仅受到雨水影响，则无需使用保护传感器。

7. 接近感应

接近感应传感器在接触电容式触摸表面前检测到手指或其它导电性物体。假设在黑暗中，一只手伸出以打开汽车音频系统。接近检测功能在手指接近时能够使系统发亮。例如，当用户的手靠近时，音响系统按键的背光灯 LED 将点亮。

7.1 接近感应传感器类型

7.1.1 按键

具有较大 C_P 和较小计数差值（信号值）的按键可以作为接近感应传感器使用。当做按键的接近感应传感器的灵敏度相当高，远远超过了常规电容式触摸感应按键的灵敏度。

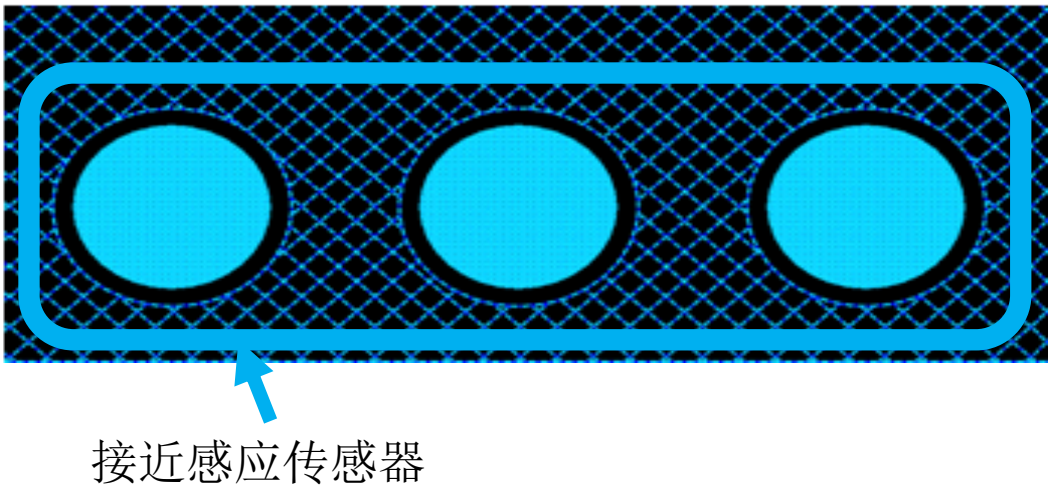
7.1.2 导线

一根导线也可以用作接近感应传感器。由于手指检测取决于随电场变化而变化的电容，影响到导线周围的电场的任何杂散电容或物体都会影响接近感应传感器的感应范围。由于制造成本和复杂性较高，所以导线传感器不适用于批量生产。

7.1.3 PCB 走线

一根较长的 PCB 走线可以构成一个接近感应传感器。走线可能是直线型，或环绕系统用户接口周边，如图 7-1 所示。这种方法适用于批量生产，但灵敏性不如导线传感器。

图 7-1. 使用 PC 走线的接近感应传感器



7.1.4 传感器机械连接

实施接近感应传感器的另一种方法是机械性连接多个传感器。实现方法是将多个传感器组合起来，形成一个大型传感器；使用固件将各个传感器连接到 PSoC Designer 内部模拟复用器总线上。使用这种方法时，请注意不要超过 C_P 设计限制。

7.2 设计建议

使用屏蔽电极有助于延长接近感应传感器的检测范围。当接近感应传感器周围存在金属物体时，这种方法特别有用。

导线传感器使屏蔽电极的作用能够更大的发挥，因为它可能被放置在离屏蔽电极更远的位置。

避免接近感应传感器内部范围较大且较坚实的接地填充区域的存在，因为这样会使灵敏度降低。

扫描速度越慢，所有位置的灵敏度越高。对于接近感应传感器，扫描速度应该非常低。

8. 低功耗设计注意事项

功耗是微控制器设计的重要因素。在降低 CapSense 控制器使用的平均电流的多种技术当中，睡眠模式的使用最为普遍。CapSense 控制器在不需要执行任何功能时会进入睡眠模式，这与手机背光灯在空闲期间变暗的情况相同。这样可以降低器件消耗的平均电流，进而满足所有电池应用的节能要求。CapSense 控制器通过向 CPU_SCR0 寄存器（位 3）中的 SLEEP（睡眠）位写入‘1’而进入睡眠模式。通过调用 M8C_Sleep 宏，可以完成此功能。进入睡眠模式后，中央 CPU 停止运行，内部主振荡器（IMO）被禁用，带隙电压参考关闭，闪存存储器模型也被禁用。保持工作的唯一电路是电源电压监视器和 32 kHz 的内部振荡器。除睡眠模式外，节省功耗技术包括：

- 禁用 CapSense（PSoC）模拟模块参考
- 禁用连续时间（CT）和开关电容（SC）模块
- 禁用 CapSense（PSoC）模拟输出缓冲区
- 将驱动模式设置为模拟高阻态

睡眠模式对设计造成负面影响。若在使用时不注意，可能会造成无法预见的操作。必须正确将 PSoC 从睡眠模式中唤醒，用户要认识到器件处于睡眠时容许的额外处理。

8.1 其它节能技术

除睡眠模式外，所有节能技术均是基于应用的技术，并且，有些技术可能造成不良后果。本章节将详细讨论每一种技术。

```
ABF_CR0 &= 0xc3; // Buffer Off
```

8.1.1 将驱动模式设置为模拟高阻态

CapSense 控制器驱动模式的状态可能影响功耗水平。你只能改变引脚不会对系统造成负面影响上的驱动模式。按顺序进行更改，这样可以避免产生线路故障。该序列取决于引脚的当前驱动模式和端口数据寄存器的状态。通过 CapSense 控制器驱动模式结构，在高阻态或强驱动模式之间切换时，引脚必须暂时处于电阻上拉或电阻下拉驱动模式。临时性驱动模式与引脚上的上一个值相反。因此，如果引脚被驱动为高电平，那么临时性驱动模式必须是电阻下拉式。这样可以确保引脚驱动模式非电阻性，消除了任何潜在的故障。

进入睡眠模式之前，使用软件对驱动模式进行手动设置。共有三个寄存器，分别是 PRTxDM0、PRTxDM1 和 PRTxDM2，用于控制驱动模式。每个寄存器向一个引脚分配一位。因此，通过对三个寄存器进行写操便可更改驱动模式。但由于整个端口是通过对三个寄存器进行相同的写操作来进行更改的，因此该操作很容易实现。模拟高阻态的正确位格式为 110b。0 端口暂时进入电阻下拉驱动模式，然后使用下列代码从强驱动切换为模拟高阻态。

```
PRT0DM0 = 0x00; // low bits
PRT0DM1 = 0xff; // med bits
PRT0DM2 = 0xff; //high bits
```

8.1.2 全部放在一起

下列代码是 28 引脚器件准备进入睡眠模式的典型序列的例子。在该序列中，中断被禁用，模拟电路被关闭，所有驱动模式均被设置为模拟高阻态，然后重新使能中断。

```
void PSoC_Sleep(void){
    M8C_DisableGInt;
    ARF_CR &= 0xf8; // analog blocks Off
    ABF_CR0 &= 0xc3; // analog buffer off
    PRT0DM0 = 0x00; // port 0 drives
    PRT0DM1 = 0xff;
    PRT0DM2 = 0xff;
    PRT1DM0 = 0x00; // port 1 drives
    PRT1DM1 = 0xff;
    PRT1DM2 = 0xff;
    PRT2DM0 = 0x00; // port 2 drives
    PRT2DM1 = 0xff;
    PRT2DM2 = 0xff;
    M8C_EnableGInt;
    M8C_Sleep;
}
```

8.1.3 睡眠模式复杂性

CapSense 控制器可以通过复位或中断的方式退出睡眠模式。CapSense 控制器中有三种复位方式：外部复位、看门狗复位和上电复位。通过其中的任意一种均可以使 CapSense 控制器退出睡眠模式；一旦复位无效后，CapSense 控制器便开始执行以 *Boot.asm* 开头的代码。用于唤醒 CapSense 控制器的可用中断有：睡眠定时器、低电压监控器、GPIO、模拟列以及异步。使用中断模式唤醒 CapSense 控制器或在睡眠状态下尝试数字通信时，会使睡眠模式的复杂性增加。这些注意事项将在后面的章节中进行讨论。

8.1.4 挂起中断

如果待处理、启用和计划在将 SLEEP 位写入到 CPU_SCR0 寄存器内后处理中断，系统将拒绝进入睡眠模式。仍然执行指令，但 CapSense 控制器没有设置 SLEEP 位。相反，中断被服务，这样将导致 CapSense 控制器完全忽略睡眠指令。为了避免这种情况，准备进入睡眠模式时，应全部禁用中断并在写入 SLEEP 位之前重新使能中断。

8.1.5 全局中断使能

从中断中唤醒 CapSense 控制器时，无需使能“全局中断使能”寄存器（CPU_F）。通过中断从睡眠模式唤醒的唯一要求是使用 INT_MSKx 寄存器中的正确中断掩模掩码位，如下面示例所述。如果禁用全局中断，系统将不执行唤醒 CapSense 控制器的 ISR，但 CapSense 控制器仍然退出睡眠模式。

在这种情况下，您必须手动清除挂起中断，或使能全局中断以处理 ISR。在 INT_CLRx 寄存器中清除中断。

```
//Set Mask for GPIO Interrupts M8C_EnableIntMask(INT_MSK0, INT_MSK0_GPIO)

// Clear Pending GPIO Interrupt
INT_CLR0 &= 0x20;
```

8.2 唤醒后执行序列

如果通过复位唤醒 CapSense 控制器，程序会由引导代码起端开始执行。如果通过中断服务子程序唤醒 CapSense 控制器，首先执行的指令是直接跟在睡眠指令后面的那个指令。这是因为 CapSense 控制器完全进入睡眠模式之前，预先提取睡眠指令后面的第一个指令。因此，如果禁用了全局中断，指令将在睡眠启动前从它所停止的位置继续执行。

8.2.1 使能 PLL 模式

如果使能 PLL 模式，CPU 在进入睡眠模式前，频率必须降至最小值 3 MHz。这是因为 PLL 在 CapSense 控制器唤醒及重新使能后重新尝试锁定时经常过冲。此外，您应在唤醒后等待 10 ms，然后开始让 CPU 正常工作，从而确保正确执行。这表示使用睡眠模式和 PLL 模式时，软件必须能够在频率为 3 MHz 下继续执行。对 OSC_CR0 寄存器进行简单的写操作可以降低 CPU 的速度。然而，该寄存器仅设置 SYSCLK 分频器，这意味着 CPU 速度将由于器件系列的不同 SYSCLK 而有所不同。通常情况下，SYSCLK 为 24 MHz。

```
OSC_CR0 &= 0xf8; // CPU = 3 IMO = 24
```

8.2.2 执行全局中断使能

不建议在写入睡眠位的指令边界上中断。假设从中断（reti）指令返回时执行睡眠命令，这可能导致所有进入睡眠固件准备被跳过。要避免这种情况，请暂时禁用中断后，再准备进入睡眠，然后在进入睡眠前再重新使能中断。由于全局中断指令时序，中断在执行下一个指令过程中不能发生，这种情况可以设置睡眠位。

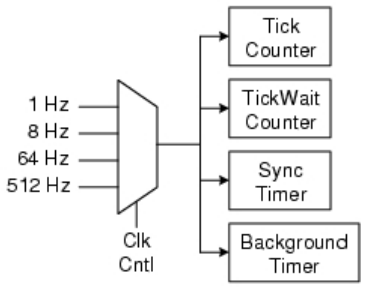
8.2.3 I²C 从设备处于睡眠模式

在睡眠模式下使用 I²C 从设备有许多复杂情况。由于在睡眠期间，IMO 和 CPU 都被关闭，CapSense 控制器不再执行任何处理。I²C 地址会产生这些问题。当将 I²C START 条件发送到特定地址时，CapSense 控制器无法处理该地址，因此发出 NAK 响应。典型的解决办法是在 I²C 总线的时钟或数据线路上设置下降沿中断。然后，主设备向其发送虚拟的 START 条件以唤醒 CapSense 控制器。唤醒与实现处理 I²C 寻址之间有一大段延迟时间，因此主设备需要推迟高达 200 µs 的时间后，再执行下一次发送或继续发送直到收到 ACK 为止。该解决方案还存在另一个问题，即为：CapSense 控制器将在所有 I²C 下降沿上被唤醒，这样引起更长的总活动时间和更高的睡眠电流。另一个解决方案是使用第三个 GPIO 引脚来唤醒 CapSense 控制器，在适当的延迟时间后发送初始的 START 条件。

8.2.4 睡眠定时器

CapSense 控制器提供睡眠定时器和睡眠定时器用户模块。当 CapSense 控制器处于睡眠模式时，可以使用这些器件，并这两个器件都执行类似功能。实际的睡眠定时器由永远不被关闭的内部低速振荡器控制。可选频率间隔为 1 Hz、8 Hz、64 Hz 和 512 Hz 时，定时器发生一个中断。对定期唤醒 CapSense 控制器执行某些处理或检查活动是非常有用。例如，定期唤醒以便扫描传感器。睡眠定时器用户模块使用睡眠定时器来生成某些额外功能。这些功能包括后台钟表计数器（用来生成定期中断）、程序环路延迟功能、可设置的递减计数器和控制回路时间的回路循环调节器。图 8-1 显示了该功能的简单框图。

图 8-1. 睡眠定时器用户模块框图



9. 资源

9.1 网站

赛普拉斯 CapSense 控制器网站提供了本章节所讨论的全部参考材料的访问权限。

CY8C21x34/B 网页包含适用于 CY8C21x34/B 器件系列的各种技术资源。

9.2 数据手册

CapSense CY8C21x34/B 器件系列的数据手册可以从 http://www.cypress.com 网站中获取。

- CY8C21234、CY8C21334、CY8C21434、CY8C21534、CY8C21634
- CY8C21234B、CY8C21334B、CY8C21434B、CY8C21534B、CY8C21634B

9.3 技术参考手册

赛普拉斯已创建了下列技术参考手册，用于对 CapSense 控制器功能的信息（包括顶级架构图、寄存器和时序图）进行快速而简易的访问。

- CY8CPLC20 、 CY8CLED16P01 、 CY8C29x66 、 CY8C27x43 、 CY8C24x94 、 CY8C24x23 、 CY8C24x23A 、 CY8C22x13、CY8C21x34、CY8C21x23、CY7C64215、CY7C603xx、CY8CNP1xx 以及 CYWUSB6953 PSoC® 可编程片上系统技术参考手册（TRM）

9.4 开发套件

9.4.1 通用的 CapSense 控制器套件

通用 CapSense 控制器套件具有预定义控制电路和即插即用硬件，易于进行原型设计和调试。该套件还包含了用于调校和数据采集的编程硬件和 I²C 至 USB 桥接器。

- CY3280-BK1 通用的 CapSense 控制器

9.4.2 通用 CapSense 模块板

9.4.2.1 简单触摸按键模块板

CY3280-BSM 简单按键模块由十个 CapSense 按键和十个 LED 构成。该模块可连接至任何 CY3280 通用的 CapSense 控制器电路板。

9.4.2.2 矩阵按键模块板

CY3280-BMM 阵列按键模块由 8 个 LED 和 8 个 CapSense 传感器组成（以 4×4 阵列格式组织，从而构成 16 个物理按键）。该模块可连接至任何 CY3280 通用 CapSense 控制器电路板。

9.4.2.3 线性滑条模块板

CY3280-SLM 线性滑条模块由五个 CapSense 按键、一个线性滑条（带十个传感器）和五个 LED 构成。该模块可连接至任何 CY3280 通用 CapSense 控制器电路板。

9.4.2.4 辐射滑条模块板

CY3280-SRM 辐射滑条模块由四个 CapSense 按键、一个辐射滑条（带 10 个传感器）和 4 个 LED 构成。该模块可连接至任何 CY3280 通用 CapSense 控制器电路板。

9.4.2.5 通用 CapSense 原型设计模块

CY3280-BBM 通用的 CapSense 原型模块提供了对路由至连有控制器的电路板上 44 引脚连接器的各个信号的连接。原型模块电路板与通用的 CapSense 控制器电路板一起组合用来实施通用 CapSense 模块电路板不具有的其它功能。

9.4.3 在线仿真器（ICE）套件

在原型系统中通过柔性线缆，ICE Pod 提供 CY3215-DK 在线仿真器与 PSoC 目标器件之间的连接，或者在 PCB 上通过封装特定的 Pod 支脚提供该连接。以下 Pod 可用：

- CY3250-20334QFN 在线仿真器（ICE）Pod 套件，用于调试 QFN CY8C21x34 PSoC 器件

9.5 PSoC Programmer

PSoC Programmer 是一种灵活的集成式编程应用，用于对 PSoC 器件进行编程。PSoC Programmer 可以与 PSoC Designer 和 PSoC Creator 一起使用，从而对 PSoC 器件上的任意设计进行编程。

PSoC Programmer 提供 API 硬件层，用来设计使用编程器和桥器件的特定应用。在 COM 指南文档中介绍了 PSoC Programmer 硬件层以及跨下列语言的示例代码：C#、C、Perl 和 Python。

9.6 MultiChart

MultiChart 是一种用于实时 CapSense 数据查看和记录的简单 PC 工具。借助该应用，您可以查看来自多达 48 个传感器的数据、保存和打印图表、保存数据以便日后在电子表格中进行分析。

9.7 PSoC Designer

赛普拉斯具备独有的集成设计环境 — PSoC Designer。使用 PSoC Designer，您可以配置模拟和数字模块、开发固件和调试设计。在拖放式设计环境中使用预先设定的模拟和数字功能库（包括 CapSense）来开发您的应用。PSoC Designer 带有内置 C 语言编译器和嵌入式编程器。专业版编译器可用于复杂设计。

9.8 代码示例

赛普拉斯提供多个代码示例以便实现您的设计并快速运行。

- CapSense 控制器代码示例设计指南
- CY8C21x34 上带有 EzI2C 从设备的 CSD 软件滤波器
- PSoC 与 SPI EEPROM 的接口
- 伪随机序列发生器
- CY8C21x34 上带有 I2CHW 从设备的 CSD
- CMPPRG 用户模块示例
- 使用 CYRF6936 和 CY8C27643 接收字符串
- CY8C21x34 上的 CSD 和 SPIS

9.9 设计支持

赛普拉斯具有许多设计支持渠道，由此确保您的 CapSense 解决方案成功。

- 知识库文章 — 按产品系列浏览技术文章，或根据不同 CapSense 主题进行搜索。
- CapSense 应用笔记 — 请参见本文档中说明信息所涉及的广泛应用笔记。
- 白皮书 — 了解高级电容式触摸接口主题。
- 赛普拉斯开发社区 — 与赛普拉斯技术社区联系并交换信息。
- CapSense 产品选型指南 — 请参见赛普拉斯 CapSense 产品线完整的产品系列。
- 视频库 — 使用视频教程提高学习速度。
- 质量和可靠性 — 赛普拉斯承诺满足客户的要求。在我们的“质量”网站上，您可以查找可靠性和产品资质报告。
- 技术支持 — 世界一流的在线技术支持。

术语表

AMUXBUS

指的是 PSoC 中的模拟复用器总线，通过它可将 I/O 引脚连接至多个内部模拟信号。

SmartSense™ 自动调校

设计阶段结束后，CapSense 算法自动设置各个感应参数以得到最佳性能，然后连续补偿由于系统、生产过程和环境不同引起的变化。

基准线

指的是从固件算法得到的数值。当传感器上没有手指触摸时，该算法将估计原始计数的值。基准线对原始计数突变的灵敏度较低，另外它还为计数差值的计算提供了参考点。

按键或按键 widget

指的是带有相关传感器的 widget，它会报告传感器的活动或非活动状态（即仅两种状态）。例如，它可以检测到传感器上是否有手指触摸。

计数差值

指的是原始计数与基准线间的差值。如果该差值为负，或如果它低于噪声阈值，则计数差值总是被设置为‘0’。

电容传感器

导体和基板（如印刷电路板（PCB）上的铜质按键）会对触摸事件或接近电容变化物体作出反应。

CapSense®

赛普拉斯的触摸感应用户界面的解决方案这是行业排名第一的解决方案，销量是排名第二的方案的四倍。

CapSense 机械按键替换（MBR）

将机械按键升级到电容式按键的赛普拉斯可配置解决方案仅需要很少的工程功耗，并且不需要固件开发。这些器件包括 CY8CMBR3XXX 和 CY8CMBR2XXX 系列。

中心或中心位置

是指在滑条分辨率所给定的范围内，表示滑条上的手指位置的数字。该数字由 CapSense 中心计算算法计算得出。

补偿 IDAC

指的是可编程的恒流源，CSD 通过使用该恒流源补偿多余的传感器 C_P。与调制 IDAC 不同，该 IDAC 没有受 CSD 模块中 Sigma-delta 调制器的控制。

CSD

CapSense Sigma Delta（CSD）是赛普拉斯专利方法，用于测量电容式感应应用的自电容。

在 CSD 模式下，感应系统测量电极的自电容，且检测自电容的变化，从而确定是否有手指触摸。

去抖动

用于定义连续扫描样本数量，其中应存在触摸事件，以使它变成有效的参数。该参数有助于抑制假的触摸信号。

对于连续扫描样本的去抖动数量，仅在计数差值大于手指阈值+迟滞时，手指触摸才被报告。

驱动屏蔽（Driven-Shield）

指的是 CSD 所使用的一种技术，用于使能防水功能，其中屏蔽电极由一个信号驱动，该信号的相位和幅度与传感器开关信号的相等。

电极

指的是导电材料，如 PCB 板、ITO 或 FPCB 板上的垫片或物理层。电极连接到 CapSense 器件的端口引脚，并作为 CapSense 传感器使用或用于驱动与 CapSense 功能相关的特定信号。

手指阈值

与 Hysteresis（迟滞）一起使用的参数，旨在确定传感器的状态。如果计数差值高于手指阈值+迟滞，传感器状态将显示‘ON’；如果计数差值低于手指阈值-迟滞，则传感器状态将显示‘OFF’。

组合传感器

这是将多个传感器连接在一起，并将它们作为单个传感器进行扫描的方法。该方法用于扩大接近感应的传感器面积，并降低功耗。

当系统处于低功耗模式时，为了降低功耗，需要将所有传感器连接在一起并将其作为单个传感器进行扫描（而不是单独扫描所有传感器），这样可以缩短扫描时间。当用户触摸任何传感器时，系统会进入活动模式，在该模式中，它会单独扫描所有传感器，以检测哪个传感器被激活。

PSoC 通过固件支持传感器组合，这意味着，可以将多个传感器同时连接到 AMUXBUS，以进行扫描。

手势

手势是一个由用户执行的动作，如滑动和线捏/缩放等等。CapSense 具有手势检测功能，即根据预定义的触摸格式来识别不同的手势。在 CapSense 组件中，只有触摸板 widget 支持手势功能。

保护传感器

指的是 PCB 板上围绕所有传感器的铜线，它类似于按键传感器并用于检测水流。触发保护传感器时，固件会禁用对所有其它传感器进行的扫描，以防止误触摸。

填充或网格地填充或网格布的铺设

当设计一个拥有电容式感应功能的 PCB 板时，应将铜制接地层放置在传感器周边，以获取良好的抗噪能力。但是实心接地层会使传感器的寄生电容增加（这种电容是不需要的）。因此，应以特殊网格方式填充接地层。紧密地放置网格地，十字交叉线像丝网一样，线宽度和两条线间的距离确定填充百分比。具有防水功能时，将通过屏蔽信号（而不是接地层）驱动该网格填充（作为屏蔽电极使用）。

迟滞

用于防止由系统噪声产生随机切换造成传感器状态的参数，它与手指阈值一起使用，以确定传感器状态。请查看手指阈值。

IDAC（电流输出的数模转换器）

PSoC 中的可编程恒流源，用于 CapSense 和 ADC 操作。

防水功能

存在水滴、水流或薄雾时，电容感应系统仍能够正常工作的能力。

线性滑条

指的是包含一个传感器以上的 widget。这些传感器以特殊的线性方式安排以检测手指（在单轴内）的物理位置。

低基准线复位

表示扫描样本最大数量的参数，其中原始计数异常低于负噪声阈值。如果超过了低基准线复位值，基准线将被复位到当前的原始计数。

手动调校

指的是手动设置（或调校）CapSense 参数的过程。

矩阵按键

指的是包含两个传感器以上（这些传感器以矩阵方式安排）的 widget。通过使用它可以在各个传感器（这些传感器以垂直方向和横向安排）的交点上检测是否有手指（触摸）。

如果 M 是横轴上的传感器数量，且 N 是纵轴上的传感器数量，那么矩阵按键 Widget 只需要使用 M + N 端口引脚就可以监控 M x N 总交叉点。

使用 CSD 感应方法（自电容）时，该 Widget 一次只能检测一个交叉点位置上的有效触摸。

调制电容（CMOD）

在自电容感应模式下 CSD 模块操作所需要的外部电容。

调制器时钟

指的是一个时钟源，在传感器扫描过程中用于采样从 CSD 模块输出的调制器。该时钟还为原始计数计数器提供时钟。扫描时间（不包括前处理和后处理时间）的计算公式为：$(2^N – 1)$/调制器的时钟频率，其中 N 是扫描分辨率。

调制 IDAC

调制 IDAC 是可编程的恒流源，它的输出由 CSD 模块中的 Sigma-delta 控制器输出控制（ON/OFF），以保持 AMUXBUS 电压始终为 V_{REF}。该 IDAC 提供的平均电流等于传感器电容引出的平均电流。

互电容

一个电极（假设为 TX）与另一个电极（假设为 RX）间的相对电容被称为互电容。

负噪声阈值

用于区分通常噪声与不想要的杂散信号的阈值。该参数与低基准线复位参数结合使用。

通过更新基准线，可以跟踪原始计数和负噪声阈值范围内的原始计数的变化，也就是基准线与原始计数之差（基准线 – 原始计数）小于负噪声阈值。

负方向的杂散信号可被触发的场合包括：上电时传感器上有手指触摸，除去传感器附近的金属物体，移除带有防水功能的 CapSense 产品上的水滴，以及突然发生其它的环境变化。

噪声（CapSense 噪声）

传感器处于‘OFF’状态（无触摸）时原始计数的变量，使用峰至峰计数来测量。

噪声阈值

用于区分传感器的信号和噪声的参数。如果原始计数 – 基准线的值大于噪声阈值，该参数将表示信号可能有效。如果差值小于噪声阈值，则该原始计数仅包括噪声。

覆盖层

指的是覆盖电容式传感器，并用作触摸表面的非导电材料（如塑胶和玻璃）。将带有多个传感器的 PCB 直接放置在覆盖层下面，或通过弹簧连接。产品的外壳常作为覆盖层使用。

寄生电容（C_P）

寄生电容是由 PCB 走线、传感器垫片、过孔以及气隙组成的传感器电极的内部电容。这是不想要的情况，因为它会使 CSD 的灵敏度降低。

接近感应传感器

指的是不需要物理接触却能够检测到附近的物体的传感器。

辐射滑条

指的是包含多于一个传感器的 widget。这些传感器以特殊的圆形方式安排以检测手指的物理位置。

原始计数

代表传感器物理电容的 CapSense 硬件模块的未处理数值输出。

刷新间隔

传感器两次连续扫描间的时间。

扫描分辨率

由 CSD 模块生产的原始计数分辨率（单位为位）。

扫描时间

完成传感器的扫描过程所需要的时间。

自电容

与电路接地和电极相关的电容。

灵敏度

指的是原始计数随传感器电容的变化，用计数/pF 来表示。传感器灵敏度取决于电路板布局、覆盖层属性、感应方法以及调校参数。

感应时钟

用来实现 CSD 感应方法的开关电容前端的时钟源。

传感器

请参见电容传感器。

传感器自动复位

用于防止传感器无限期地报告由系统故障或金属物体连续显示在传感器附近时造成的误触摸状态的设置。

使能传感器自动复位时，即使计数差值大于噪声阈值，也可以更新基准线。这样将防止传感器无限期地报告‘ON’状态。禁用传感器自动复位时，只有计数差值小于噪声阈值时才能更新基准线。

传感器组合

请参见组合传感器。

屏蔽电极

传感器周围填充铜，以便防止水滴或其它液体引起的误触摸。屏蔽电极由 CSD 模块输出的屏蔽信号驱动。请参见驱动屏蔽（Driven-Shield）。

屏蔽槽电容（C_{SH}）

指的是（当有一个带有高的寄生电容的大屏蔽层时，）用于增强 CSD 屏蔽的驱动能力的可选外部电容（C_{SH} 槽电容）。

信号（CapSense 信号）

计数差值还被称为信号。请参见计数差值。

信噪比（SNR）

有手指触摸时的传感器信号与无手指触摸时的传感器信号间的比例。

滑条分辨率

表示滑条上需要处理的手指位置总数的参数。

触摸板

指的是包含多个传感器的 Widget（这些传感器以特殊的横向和纵向安排），用于检测一个触摸的 X 和 Y 位置。

触摸板

请参见触摸板。

调校

“调校”是使 CapSense 操作中所需的各种硬件和软件或阈值参数达到最佳值的过程。

V_{REF}

PSoC 中的可编程参考电压模块，用于 CapSense 和 ADC 操作。

Widget

指的是 CapSense 组件中包括一个传感器或一组类似传感器的用户界面元素。受支持的 widget 包括按键、接近感应传感器、线性滑条、辐射滑条，矩阵按键和触摸板。

修订记录

文档修订记录

文档标题：AN66271 — CY8C21x34/B CapSense®设计指南 文档编号：001-78594			
版本	**提交日期**	**变更者**	**变更说明**
**	04/23/2012	GKL	本文档版本号为 Rev**，译自英文版 001-66271 Rev**。
*A	05/05/2015	GKL	本文档版本号为 Rev*A，译自英文版 001-66271 Rev*F。
*B	06/29/2016	GKL	本文档版本号为 Rev*B，译自英文版 001-66271 Rev*G。
*C	07/25/2017	AESATMP8	更新标志和版权。
*D	01/17/2019	XITO	译自英文版 001-66271 Rev*H。更新版权