AN85951

PSoC® 4 和 PSoC 6 MCU CapSense®设计指南

文档编号：001-87199 版本*I

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

© 赛普拉斯半导体公司，2013-2020 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约归赛普拉斯所有。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。如果软件没有附带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适用于个人的、非独占性、不可转让的许可（无转授许可权）（1）在版权保护下的软件（a）以源代码形式提供的软件,只能是在组织内部为了使用赛普拉斯的硬件去修改和复制。（b）以二进制代码形式从外部发到终端用户（直接或间接通过经销商和分销商）,仅用于赛普拉斯硬件产品单元。（2）在软件（由赛普拉斯公司提供，且未经修改）侵犯赛普拉斯专利的权利主张下，仅许可在赛普拉斯硬件产品上制造、使用、提供和导入软件。禁止对软件的任何其他使用、复制、修改、翻译或编译。

赛普拉斯不对此材料提供任何类型的明示或暗示保证，包括但不限于针对特定用途的适销性和适用性的暗示保证。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。在适用法律允许的范围内，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。贵方应赔偿并保护赛普拉斯免受所有索赔的损害，包括因人身伤害或死亡引起的索赔、费用、损失和其它责任。

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

目录 3

1 简介 6

1.1 摘要 6
1.2 简介 6
1.3 CapSense 功能 7
1.4 PSoC 4 和 PSoC 6 MCU CapSense Plus 特性 7
1.5 CapSense 设计流程 9

2 CapSense 技术 11

2.1 CapSense 基本原理 11
2.1.1 自电容感应 12
2.1.2 互电容感应 14
2.2 电容式触摸感应方法 15
2.2.1 CapSense Sigma Delta（CSD） 15
2.2.2 CapSense Crosspoint（CSX） 15
2.3 信噪比 17
2.4 CapSense Widget 18
2.4.1 按键（零维传感器） 18
2.4.2 滑条（一维传感器） 19
2.4.3 触摸板/触控板（二维传感器） 20
2.4.4 接近（三维传感器） 21
2.5 防水功能 21
2.5.1 水滴和水流对 CapSense 传感器产生的影响 22
2.5.2 驱动屏蔽信号和屏蔽电极 24
2.5.3 保护传感器 25

3 PSoC 4 和 PSoC 6 MCU CapSense 27

3.1 CapSense CSD 感应方式 27
3.1.1 GPIO 单元中电容-电流转换器 28
3.1.2 IDAC 源模式 28
3.1.3 IDAC 灌电流模式 29
3.1.4 CapSense 时钟发生器 30
3.1.5 Sigma Delta 转换器 31
3.1.6 模拟复用器 33
3.1.7 CapSense CSD 屏蔽 33

3.2 CapSense CSX 感应方式 33
3.2.1 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 中的 CapSense 架构 35
3.3 PSoC 4xxxM/4xxxL 系列中的 CapSense 36

4 CapSense 设计和开发工具 38

4.1 PSoC Creator 38
4.1.1 CapSense 组件 38
4.1.2 CapSense_ADC 组件 39
4.1.3 调谐器助手 39
4.1.4 示例项目 39
4.2 ModusToolbox 40
4.3 硬件套件 40

5 CapSense 功能调校 41

5.1 选择 SmartSense 还是手动调校 41
5.2 SmartSense 42
5.2.1 SmartSense 的组件配置 43
5.3 手动调校 51
5.3.1 概述 51
5.3.2 CSD 感应方法 53
5.3.3 CSX 感应方法 75
5.3.4 手动调校权衡 83
5.3.5 调校调试常见问题解答 84

6 CapSense 手势 87

6.1 CapSense 的手势支持 87
6.2 手势组 88

7 设计注意事项 89

7.1 固件 89
7.1.1 低功耗设计 90
7.2 传感器结构 91
7.3 覆盖层选择 92
7.3.1 覆盖层材料 92
7.3.2 覆盖层厚度 93
7.3.3 覆盖层粘合剂 93
7.4 PCB 布局指南 93
7.4.1 寄生电容 C_P 93
7.4.2 电路板的各个层 94
7.4.3 滑条设计 97
7.4.4 传感器和器件放置 103
7.4.5 走线长度和宽度 104
7.4.6 走线路由 104
7.4.7 串扰解决方案 105

7.4.8 过孔 106
7.4.9 地层 106
7.4.10 电源的布局建议 110
7.4.11 耐水性布局指导 110
7.4.12 原理图规格检查表 113
7.4.13 布局规格检查表 114
7.5 ESD 保护 117
7.5.1 防止 ESD 放电 117
7.5.2 重定向 118
7.5.3 ESD 保护器件 118
7.6 电磁兼容性（EMC）注意事项 119
7.6.1 辐射干扰及辐射 119
7.6.2 导电射频噪声 128

8 CapSense Plus 129

9 资源 133

9.1 网站 133
9.2 数据手册 133
9.3 技术参考手册 133
9.4 开发套件 133
9.5 PSoC Creator 133
9.6 ModusToolbox™ 133
9.7 应用笔记 134
9.8 设计支持 134

术语表 135

修订记录 140

文档修订记录 140

1 简介

1.1 摘要

CapSense®设计指南说明了如何使用 PSoC® 4 和 PSoC 6 MCU 器件系列所具有的 CapSense 功能来设计电容式触摸感应应用。这些器件中的 CapSense 性能能够提供前所未有的信噪比、一流的防水功能，以及各种传感器（如按键、滑条、触控板和接近感应传感器）。本指南介绍的是 CapSense 操作、CapSense 设计工具、PSoC Creator™ CapSense 组件的性能调校以及设计注意事项。该指南还介绍了赛普拉斯用于 CapSense 评估的新 ModusToolbox™设计工具。

赛普拉斯为各种不同的器件系列提供了 CapSense 功能。如果用户没有选择特定的器件或不熟悉的电容式感应，请参考 CapSense 设计指南入门。该指南有助于您了解 CapSense 比机械按键的优势、CapSense 技术的基本原理以及如何为您的应用选择合适的器件。它还指导您如何找到合适的文档、套件或工具，从而为您的设计提供帮助。

1.2 简介

电容式触摸传感器是用户接口器件，通过检测人体电容的变化来检测传感器周围是否存在手指触摸。赛普拉斯 CapSense 解决方案为您的产品提供了精致、可靠且使用方便的电容式触摸感应功能。

本设计指南侧重于 PSoC 4 和 PSoC 6 MCU 器件系列中的 CapSense 功能。这些器件是真正的可编程嵌入式片上系统，它在单芯片上集成了可配置的模拟和数字外设功能、存储器、无线模块和微控制器。这些器件非常灵活，除 CapSense 功能外，它还能够执行多项功能（如 ADC、DAC 和 BLE），从而加快产品的上市时间、集成关键的系统功能，并降低整个系统的成本。

该指南假定您已经熟悉了使用赛普拉斯 PSoC Creator™集成开发环境（IDE）对 PSoC 4 和 PSoC 6 MCU 进行开发应用。如果您尚未了解 PSoC 4，请参考 AN79953 — PSoC 4 入门手册或 AN92167 — PSoC 4 BLE 入门手册以加深对该产品的了解。如果您不熟悉 PSoC 6 MCU，请参考 AN221774 — PSoC 6 MCU 入门和 AN210781 — 具有蓝牙低功耗（BLE）连接的 PSoC 6 MCU 入门。如果您尚未了解 PSoC Creator，请参考 PSoC Creator 主页。

如果您不收悉 ModusToolbox，请参考 ModusToolbox™ IDEk 快速入门指南。

通过本指南，您会了解：

- PSoC 4 和 PSoC 6 MCU 中的 CapSense 技术
- 用于 PSoC 4 和 PSoC 6 MCU CapSense 的设计和开发工具
- PSoC 4 和 PSoC 6 MCU 的 CapSense PCB 布局指南
- PSoC 4 和 PSoC 6 MCU CapSense 组件的性能调校
- 采用 CapSense Plus™特性的应用（如电机控制系统和电磁炉）

1.3 CapSense 功能

PSoC 4 和 PSoC 6 MCU 中的 CapSense 具有以下特性：

- 支持自电容和互电容触摸感应
- 功能强大的 CapSense Sigma Delta（CSD）和 CapSense Crosspoint（CSX）感应技术，能够为基于自电容和互电容的触摸感应提供一流的信噪比。
- 在覆盖材料和厚度不同的条件下仍能提供高性能感应（请参考 CapSense 基本原理、覆盖层材料和覆盖层厚度）
- SmartSense™自动调试技术
- 接近感应的范围较大（接近感应距离可达 30 cm）
- 防水功能（请参见防水功能）
- 伪随机序列（PRS）时钟源可降低电磁干扰（EMI）
- 功耗较低：工作电压低至 1.71 V，休眠模式下的电流消耗低至 150 nA
- 所有 GPIO 引脚都支持电容式感应和屏蔽[1]
- 允许将 CapSense 模块重新配置为 ADC，并且所有 GPIO 引脚都支持 ADC 输入 [1]
- 通过使用可编程参考电压（V_{REF}），可提供优越的信噪比
- 支持用于降低电磁干扰（EMI）的扩频和可编程电阻开关
- 在 CapSense 扫描过程中，通过将初始化和配置过程分担给 CapSense 序列发生器，可降低 CPU 开销。

与 PSoC 4100S 器件相比，PSoC 4100S Plus 器件具有以下附加功能：

- 提供更大的闪存存储器和更多的 I/O
- 提供一个控制区域网络（CAN）模块
- 为加密应用提供用于安全密钥生成的真随机数
- 接受 4 至 33 MHz 频率范围的外部晶振（ECO）

1.4 PSoC 4 和 PSoC 6 MCU CapSense Plus 特性

您可创建带有电容式触摸感应和其他系统功能的 PSoC 4 CapSense Plus 应用。 除了 CapSense 外，这些器件还有以下关键特性：

- 带单周期乘法的 ARM® Cortex®-M0/M0+CPU，其工作频率为 48 MHz，并且它的指令执行速度高达 43 DMIPS
- 在环境温度范围为–40 至 85 °C 中，工作电压范围为 1.71 ~ 5.5 V
- 高达 128 KB 的闪存空间（CM0+的代码容量超过了采用 8 位解决方案的芯片的代码容量两倍）
- 高达 16 KB 的 SRAM
- 多达 94 个可编程 GPIO
- 独立的中心对齐 PWM，其特点包括互补死区可编程输出、已被同步化的 ADC 操作（在 PWM 周期中用户可指定的时间内触发 ADC 的能力）和同步刷新（在所有的 PWM 上对 PWM 占空比变化进行同步化操作，以避免发生异常波形）
- 基于比较器的 PWM 停止（Kill）信号触发（以便在检测到过电流条件时终止电机驱动）

[1] 对于 PSoC 6 系列器件，请遵循相应器件数据手册的 Errata 一节中所述的建议，以获得最佳的 CapSense 灵敏度和精度。

- 12 位的 1 Msps ADC，具有“采样和保持”（S&H）功能，且无需序列开销，因而，可将整个 ADC 带宽使用于信号转换，而无需将其用于定序器开销
- 具有比较器模式和 SAR 输入缓冲能力的运算放大器
- Segment LCD 直接驱动器支持 4 个共用引脚
- T/I2C 串行通信通道
- 蓝牙低功耗（BLE）通信与版本 4.0 和版本 4.1 的多项特性相兼容。
- 可编程的逻辑模块，每个模块包括八个宏单元和一个可级联的数据路径（即为通用数字模块（UDB）），用于有效地执行可编程外设（如 I2S）
- 控制器区域网络（CAN）
- 全面支持 PSoC Creator 设计入口、开发和调试环境，从而提供：
  - 设计入口和构建（了解模拟路由）
  - 用于所有的固定功能外设和通用的可编程外设的组件
  - 文档和培训模块
- 支持将编译内容导入到 MDK ARM 环境（前面将其称为 RealView）和其他环境内。

除了 CapSense 外，PSoC 6 MCU 器件还有以下主要特性：

- 单 CPU 器件（Arm Cortex-M4），双 CPU 器件（Cortex-M4 and Cortex-M0+）。硬件中支持处理器间的通信
- 在 1.71 V ~ 3.6 V 的器件工作电压范围内，用户可以选择 1.1 V 或 0.9 V 的内核逻辑操作
- 高达 2 MB 的闪存存储器以及高达 1 MB 的 SRAM
- 多达 78 个 GPIO，可用于模拟、数字、CapSense 或 segment LCD 功能
- 可编程模拟模块：两个运算放大器、可配置 PGA、比较器、12 位 SAR ADC（1 Msps）、12 位电压模式 DAC
- 可编程数字模块、通信接口
- 12 个 UDB、32 个 TCPWM，可配置为 16 位/32 位定时器、计数器、PWM 或正交解码器
- 九个串行通信模块（SCB），可配置为 I2C、SPI 或 UART 接口
- 具有一个 I2S 接口和两个 PDM 通道的音频系统
- SMIF 接口，支持从外部四路 SPI 闪存实现的芯片内执行（XIP）以及动态加密和解密。
- 符合 BLE 5.0 规范的蓝牙智能连接（仅适用于 BLE 器件系列的 PSoC 6 MCU）

请参考 AN64846 — CapSense 设计指南入门，准确选择满足您要求的 CapSense 器件。

1.5 CapSense 设计流程

图 1-1 显示的是典型的电容式感应产品的设计周期流程；本指南中提供的内容以绿色高亮显示。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 设计流程

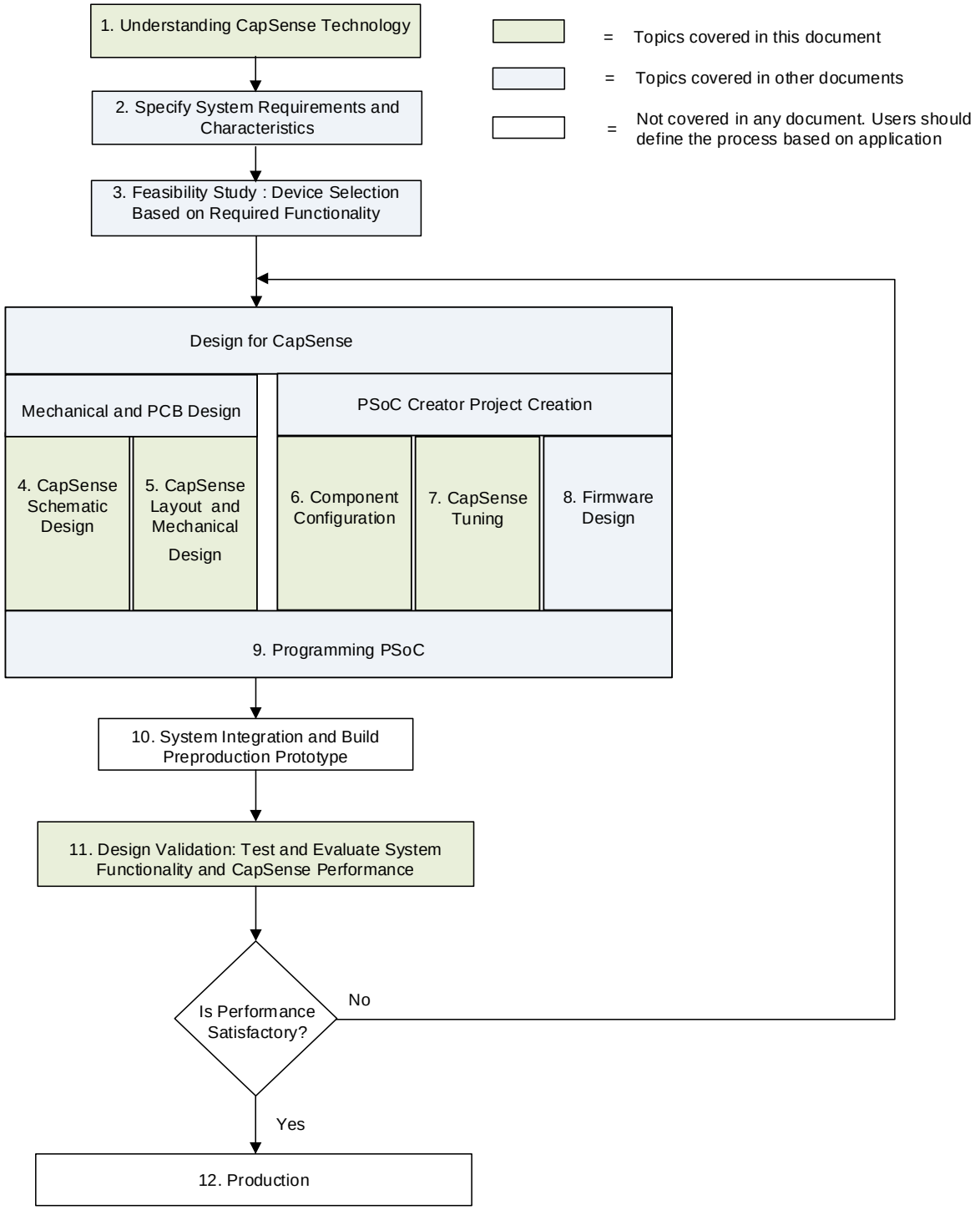


表 1-1. 支持文档

流程图中的步骤	支持赛普拉斯的文档	
	名称	章节
1. 了解 CapSense	CapSense 设计指南（本文档） CapSense 设计指南入门	第 2 章和 第 3 章 –
2. 指定要求	CapSense 设计指南入门	–
3. 可行性研究	PSoC 4 数据手册 PSoC 4 BLE 数据手册 PSoC 6 MCU 数据手册	–
	AN64846 — CapSense 设计指南入门 AN79953 — PSoC 4 入门 AN91267 — PSoC 4 BLE 入门 AN221774 — PSoC 6 MCU 入门手册	–
4. 原理图设计	CapSense 设计指南（本文档）	第 6 章
5. 布局设计	CapSense 设计指南（本文档）	第 6 章
6. 组件配置	CapSense 组件数据手册	–
	CapSense 设计指南（本文档）	第 5 章
7. 性能调校	CapSense 设计指南（本文档）	第 5 章
8. 固件设计	组件数据手册 PSoC Creator CapSense	–
	PSoC Creator 示例项目示例项目	–
	在此处下载 ModusToolbox™ 。 请参考下面与 ModusToolbox™相关的文档： ModusToolbox 发布说明 ModusToolbox™用户指南 ModusToolbox™快速入门指南 ModusToolbox™ CapSense®配置器指南 ModusToolbox™ CapSense®调谐器指南 PSoC® Creator™到 ModusToolbox™移植指南	
9. 编程 PSoC	有助于使用 IDE 进行编程的 PSoC Creator 用户指南 用于独立编程的 PSoC Programmer 主页和 MiniProg3 用户指南	–
10. 原型	–	–
11. 设计验证	CapSense 设计指南（本文档）	第 5 章
12. 生产	–	–

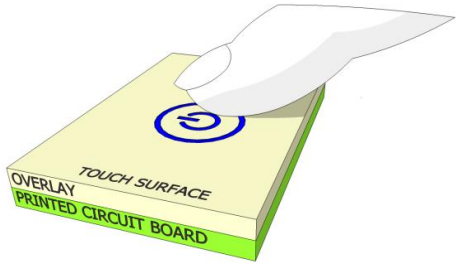
2 CapSense 技术

电容式触摸感应技术通过测量面板（传感器）和其环境之间的电容变化来检测触摸界面附近是否有手指。

2.1 CapSense 基本原理

一个典型的 CapSense 传感器包括蚀刻在印刷电路板（PCB）表面上的铜盘（带有合适的形状和大小）。绝缘覆盖层作为按键的触摸表面使用，如图 2-1 所示。

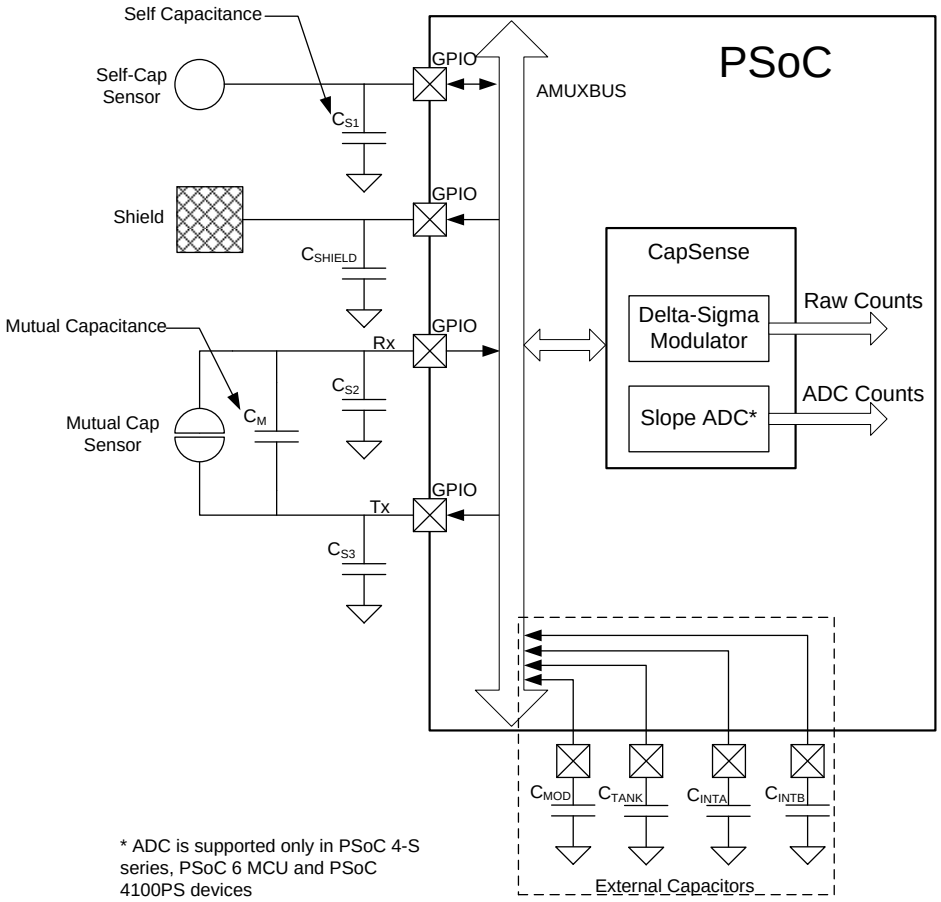
图 2-1. 电容式触摸传感器



通过 PCB 走线和过孔将传感器垫片连接到（被配置为 CapSense 传感器引脚的）PSoC GPIO。如图 2-2 所示，电极的自电容被标志为 C_{SX}，另外电极之间的互电容被标志为 C_{MXY}。PSoC 的内部 CapSense 电路将这些电容值转换为等效的数字计数值（请查看第 3 章了解相关信息）。然后，CPU 将处理这些数字计数值，以检测触摸。

对于自电容感应方法，CapSense 还需要使用外部电容 C_{MOD}，而对于互电容感应方法，则需要使用 C_{INTA} 和 C_{INTB}。这些外部电容用于将专用 GPIO 引脚接地。如果执行屏蔽电极来获得防水功能或获得较大的接近感应距离，那么可能需要使用一个额外的 C_{TANK}。表 7-6 列出了外部电容的推荐值。

图 2-2. PSoC 器件、传感器和外部电容

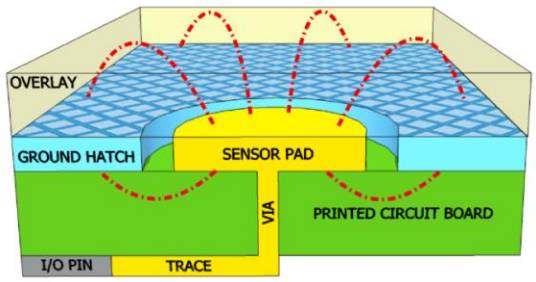


手指未接触传感器时的传感器电容称为寄生电容，C_P。寄生电容由传感器（包括传感器垫片、走线和过孔）和系统中的其他导体（如接地层、走线、产品机箱或外壳中的任何金属）之间的电场共同产生。PSoC 的 GPIO 和内部电容也会产生寄生电容。然而，与传感器电容相比，这些内部电容非常小。

2.1.1 自电容感应

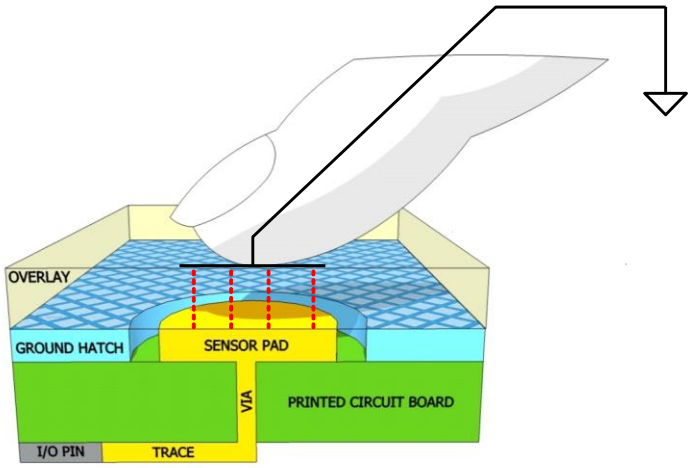
图 2-3 显示的是使用自电容感应方法时如何通过走线和过孔将 GPIO 引脚连接到传感器垫片。一般情况下，围绕传感器焊盘的接地网格将它与其他传感器和走线隔离开。 尽管图 2-3 显示了传感器垫片周围的电场线，但实际电场要复杂得多。

图 2-3. 寄生电容



当手指触摸传感器表面时，人体的导电性质和大质量构成接地的导电层（平行于传感器垫片），如图 2-4 所示。

图 2-4. 手指电容



该操作构成一个平行板电容器。传感器垫片与手指之间的电容值可通过公式 1 计算得到：

公式 2-1. 手指电容

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

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 = 覆盖层厚度

C_F 为手指电容。由于寄生电容 C_P 和手指电容 C_F 均可代表传感器引脚与地面的电容，因此两者性质相同。因此，当手指接触传感器时，传感器总电容 C_S 等于 C_P 与 C_F 之和。

公式 2-2. 手指触摸传感器时的总感应电容

$$C_S = C_P + C_F$$

如果没有触摸，C_S 等于 C_P。

PSoC 将电容 C_S 转换成相应的数字计数值，称为原始计数。由于手指触摸时会增加传感器引脚的总电容，因此原始计数的增量表示存在手指触摸。

对于 0.3 pF 的手指电容，PSoC 4 CapSense 支持高达 65 pF 的寄生电容；对于 0.1 pF 的手指电容，则支持高达 35 pF 的寄生电容。

2.1.2 互电容感应

图 2-5 显示了互电容感应技术的按键传感器布局。互电容感应将测量两个电极（即发送（Tx）和接收（Rx））间的电容。

在互电容感应系统中，为 Tx 引脚提供数字电压（VDDIO[2]或 VDDD[3]（如果器件不支持 VDDIO）和 GND 间的信号切换），并测量 Rx 引脚所接收的电荷。在 RX 电极上接收到的电荷与两个电极间的互电容（C_M）成正比。

在 Tx 和 Rx 电极间放置手指时，互电容会降低到 C^1_M，如图 2-6 所示。由于互电容降低，Rx 电极上接收到的电荷也会降低。CapSense 系统通过测量 Rx 电极上接收到的电荷来检测触摸/无触摸状态。

图 2-5. 互电容感应的工作原理

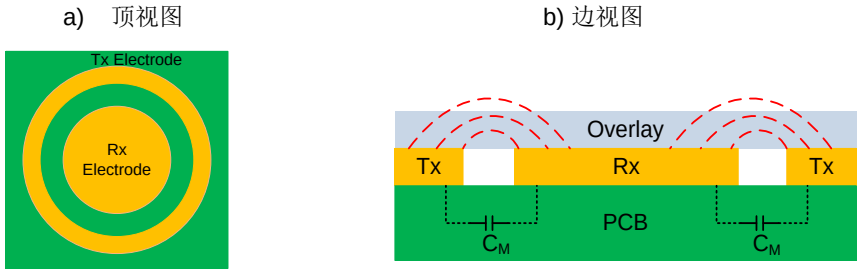
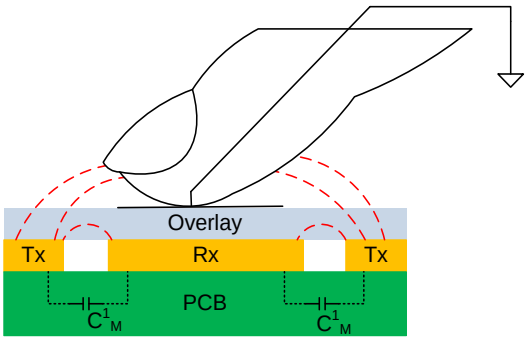


图 2-6. 发生手指触摸时的互电容



[2] VDDIO 是 I/O 引脚的电源。

[3] VDDD 是数字部分的器件电源。

2.2 电容式触摸感应方法

PSoC 4 采用赛普拉斯的电容式触摸感应方法（即 CapSense Sigma Delta（CSD））来实现自电容检测操作，并采用 CapSense Crosspoint（CSX）来实现互电容扫描操作。CSD 和 CSX 触摸感应方法能够提供行业中最佳的信噪比。这些感应方法是硬件和固件技术的组合。

2.2.1 CapSense Sigma Delta（CSD）

图 2-7 显示的是 CSD 方法的简化框图。

采用 CSD 方法时，每个 GPIO 都有一个开关电容电路，用以将传感器电容转换为等效电流。然后，模拟复用器会选择其中一个电流信号，并将该信号送到电流 - 数字转换器。电流-数字转换器的工作原理与 Delta Sigma ADC 的工作原理相似。电流-数字转换器的输出计数（被称为**原始计数值**）是一个与各个电极之间的自电容成比例的数字值。

公式 2-3. 采用 CSD 时原始计数值与传感器电容之间的关系

$$\text{raw count} = G_C C_S$$

其中，G_C 是 CSD 数字转换增益的容值。

C_S 是电极的自电容

图 2-7. CapSense Sigma Delta 方法的简化框图

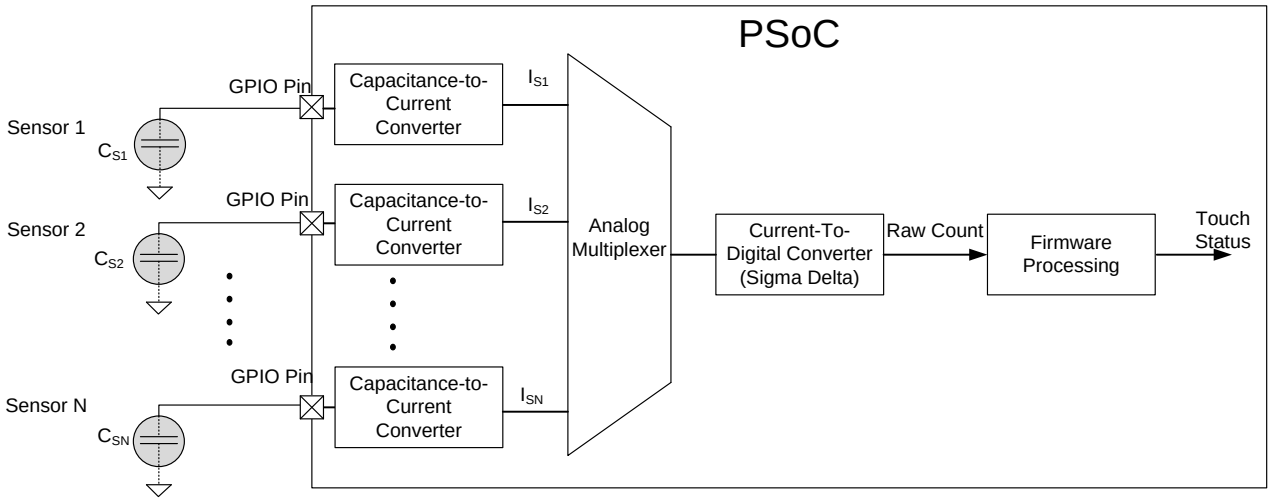
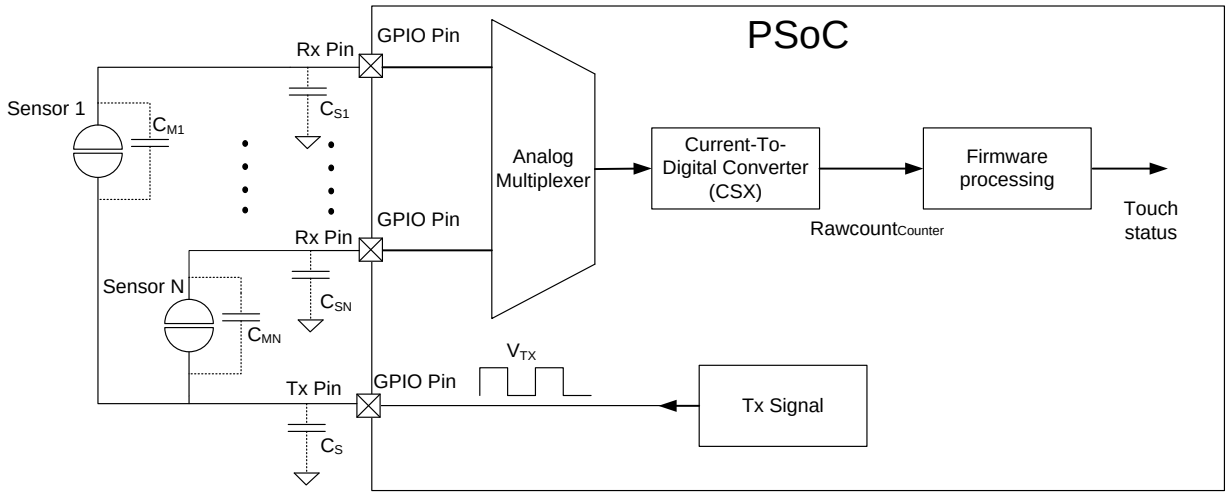


图 2-9 显示的是一段时间内的原始计数图。当有手指触摸传感器时，C_S 将从 C_P 增加到 $C_P + C_F$，同时，原始计数也会增加。通过将原始计数与某个预定的阈值进行对比，固件中的逻辑可决定传感器是否处于激活状态（即存在手指触摸）。

2.2.2 CapSense Crosspoint（CSX）

图 2-8 显示的是 CSX 方法的简化框图。

图 2-8. CSX 方法的简化框图



采用 CSX 方法时，Tx 引脚（或 Tx 电极）上的电压将电荷耦合到 RX 引脚上。该电荷与 Tx 和 Rx 电极之间的互电容成正比。然后，模拟复用器会选择其中一个 Rx 通道，并将该信道上的信号传送给电流-数字转换器。

电流-数字转换器的输出计数（被称为$\mathbf{Rawcountt}_{\text{计数器}}$原始计数值）是一个与 Rx 和 Tx 电极之间互电容成正比的数字值，如公式 2-4 所示。

公式 2-4. 采用 CSX 时原始计数值与传感器电容之间的关系

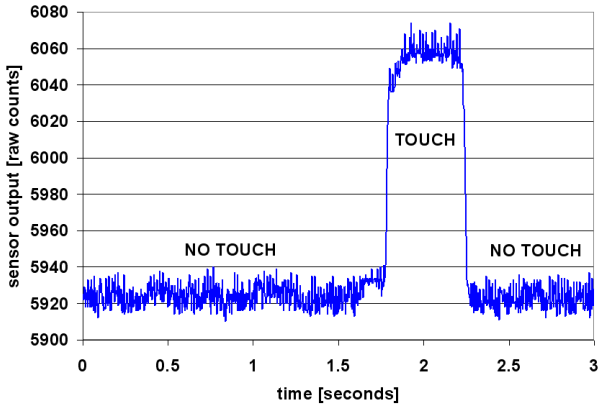
$$\mathrm{Rawcount}_{\mathrm{Counter}} = G_{CM}\, C_M$$

其中，G_{CM} 是互电容方法的数字转换增益的容值，

而 C_M 则是两个电极之间的互电容。

图 2-9 显示的是一段时间内的原始计数图。当手指接触传感器时，C_M 将从 C_M 降至 C^1_M（见图 2-6），因此计数器输出也相应降低。当 C_M 降低时，原始计数会增高。固件将其视为正常情况。这样会保持 CSD 和 CSX 方法之间的原始计数的相同视觉表示。通过将原始计数与某个预定的阈值进行对比，固件中的逻辑可决定传感器是否处于激活状态（即存在手指触摸）。使用公式 3-11 来计算增高的反转原始计数值。

图 2-9. 原始计数与时间



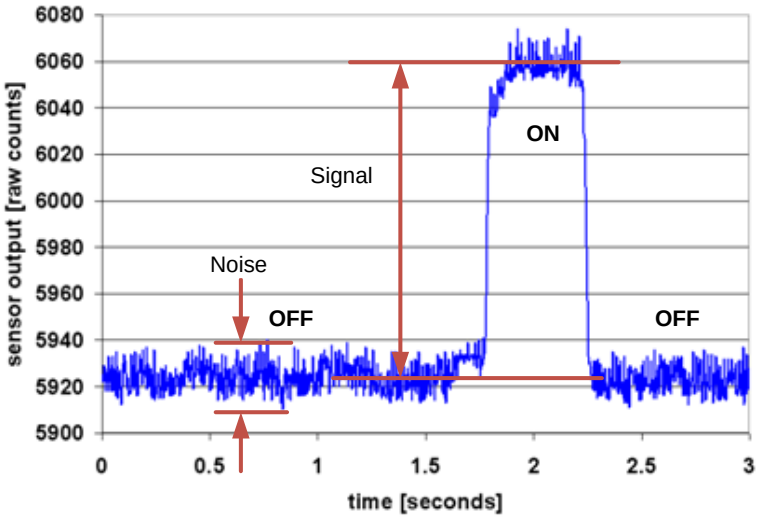
欲深入了解有关 PSoC 4 CapSense CSD 和 CSX 模块的信息，请参阅 PSoC 4 CapSense。

2.3 信噪比

实际上，原始计数会根据系统中难以消除的噪声而发生改变。CapSense 噪声是在没有触摸时原始计数中峰-峰的变化值，如图 2-10 所示。

调试较好的 CapSense 系统能够可靠地识别传感器的开关状态。要获得良好的性能，CapSense 信号必须远远大于 CapSense 噪声。信噪比（SNR）是指 CapSense 信号同 CapSense 噪声的比例，它是 CapSense 传感器最重要的性能参数。

图 2-10. SNR



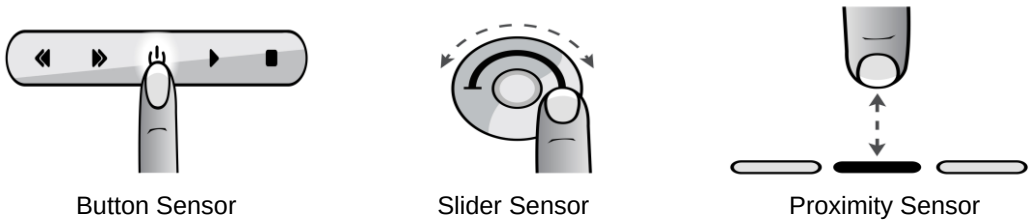
在该示例中，没有触摸时原始计数的平均水平为 5925 个计数。有手指放在传感器上时，平均原始计数值增大到 6060 个计数，因此信号为 6060 – 5925 = 135 个计数。OFF 状态下，原始计数的最小值为 5912 个计数，最大值为 5938 个计数。因此，CapSense 噪声为：5938 – 5912 = 26 个计数。这样会使 SNR 的值为 135 / 26 = 5.2。

建议 CapSense 传感器的最小 SNR 为 5。该 5:1 比例来自最佳的实践阈值设置（该设置确保信号和噪声之间的足够容限，从而获得可靠的 ON/OFF 操作）。

2.4 CapSense Widget

CapSense widget 包含一个或多个 CapSense 传感器，它作为特定用户界面类型的单位。CapSense widget 大致分为以下四类：按键传感器（零维）、滑条传感器（一维）、触摸板/触控板传感器（二维）和接近感应传感器（三维）。图 2-11 显示的是按键传感器、滑条传感器和接近感应传感器 Widget。本节介绍了各种 CapSense widget 的基本概念。更多有关传感器结构的详细信息，请参考传感器结构。

图 2-11. 几种 Widget 类型

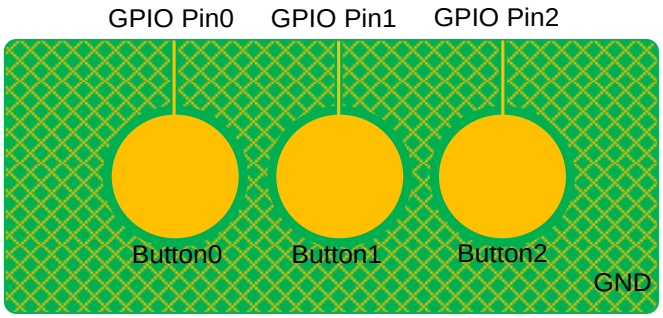


2.4.1 按键（零维传感器）

在众多应用如家用电器、医疗器件、白色家电、照明控制和多种其他产品中，CapSense 按键可替代机械按键。它是一种最简单的 CapSense Widget，包括一个简单的传感器。CapSense 按键提供两种输出状态：活动（有手指触摸）或非活动（无手指触摸）。另外，这两种状态还分别称为 ON 和 OFF 状态。

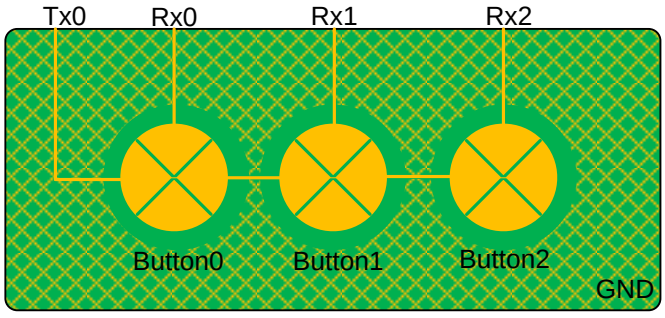
对于基于自电容的方法（例如，CSD 感应方法），一个简单的 CapSense 按键包括一个圆形的铜板（该铜板通过一根 PCB 走线连接到 PSoC GPIO 上）。按键周围应使用网格覆铜与其它的按键和走线隔离。另外，通过一个环形区域可以将按键和接地网格分开。每个按键都需要连接一个 PSoC GPIO。

图 2-12. 简单 CapSense 按键



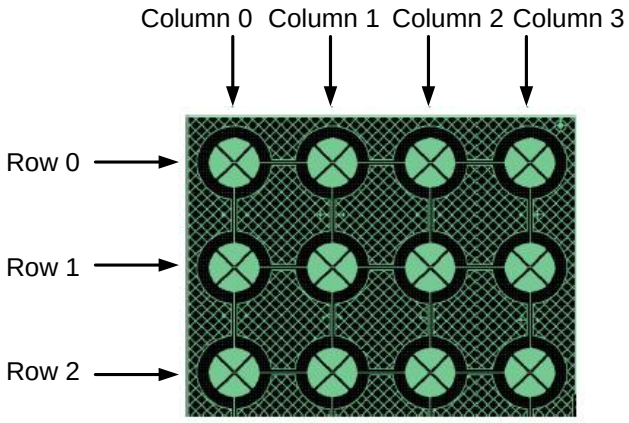
对于互电容方法（例如，CSX 感应方法），每个按键要求使用一个被配置为 Tx 电极的 GPIO 引脚和一个被配置为 Rx 电极的 GPIO 引脚。多个按键可共享 Tx 引脚，如图 2-13 所示。

图 2-13. 采用互电容感应方式的简单 CapSense 按键



如果应用需要大量的按键（如计算器键盘或 QWERTY 键盘），您可以按矩阵方式排列 CapSense 按键，如图 2-14 所示。这样，在设计中，可将每个 GPIO 使用于多个按键。例如， 图 2-14 中显示的 12 个按键设计仅要求使用 7 个 GPIO。

图 2-14. 矩阵按键



矩阵按键设计包括两组电容传感器：行传感器和列传感器。矩阵按键构架可用于自电容方法和互电容方法。互电容方法的优势在于它可以同时检测到多个手指。

在自电容模式下，每个按键包括一个行传感器和一个列传感器，如图 2-14 所示。当按键被触摸时，其行和列传感器均会被触发。矩阵支持的按键数量等于行数和列数的乘积。在自电容模式下，一次只能检测一个矩阵按键。如果有多个行或列传感器处于“触发”状态，则无法解析手指位置，且这种情况被视为无效状态。某些应用要求同时感应多个按键，如在键盘上同时按下组合键“Shift、Ctrl 和 Alt”。在这种情况下，您可以使用互电容感应方法，或者您应该独立设计 Shift、Ctrl 和 Alt 按键。

2.4.2 滑条（一维传感器）

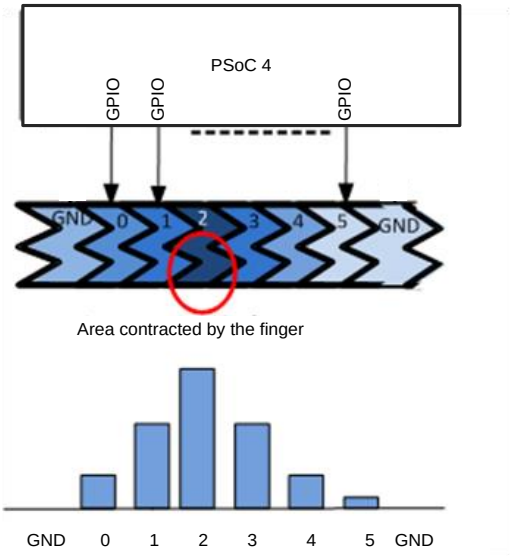
如果需要输入递增或递减，将使用滑条。示例应用包括照明控制（调光器）、音量控制、图形均衡器和速度控制。目前，PSoC Creator 中的 CapSense 组件仅支持基于自电容的滑条。组件的以后版本将支持基于互电容的滑条。

滑条由一系列称为段（相邻放置）的电容传感器构成。触摸某一个滑条段会激活邻近几个滑条段。固件通过处理被触摸的滑条段和相邻滑条段的原始计数来计算手指触摸的几何中心位置（**中心位置**）。

该中心位置的实际分辨率高于滑条段的数量。例如，包含五段的滑条至少可以解析 100 个手指的物理位置。当手指划过滑条时，高分辨率允许平缓过渡中心位置。

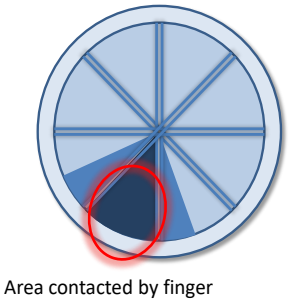
在一个线性滑条中，各滑条段按内联顺序安排，如图 2-15 显示。每个段均连接了一个 PSoC GPIO。滑条的段建议采用 Z 形图案（双 V 形图案）。这种布局可以确保在触摸某段时，还可以部分触摸邻近的段。这样会有助于几何中心位置的计算。

图 2-15. 线性滑条



辐射滑条类似于线性滑条，但辐射滑条是连续的。图 2-16 显示的是一个典型的辐射滑条。

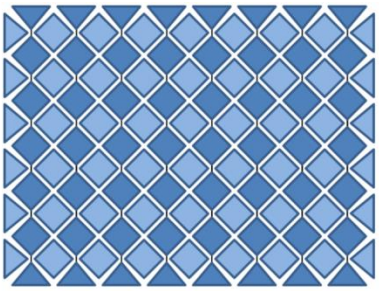
图 2-16. 辐射滑条



2.4.3 触摸板/触控板（二维传感器）

触控板（也称为触摸板）有两个按 X 和 Y 模式设置的线性滑条，从而，触控板可根据 X 和 Y 方向确定手指所在的位置。图 2-17 显示的是典型的触控板传感器安排。目前，PSoC Creator 中的 CapSense 组件仅支持基于自电容的触摸板。组件的以后版本将支持基于互电容的触摸板。

图 2-17. 触控板传感器安排

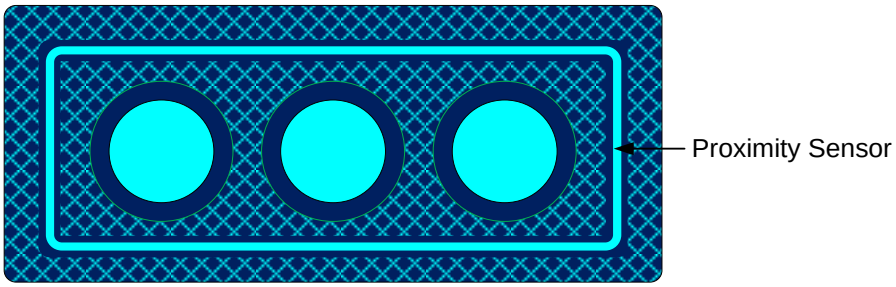


2.4.4 接近（三维传感器）

通过接近传感器可以检测传感器周围的三维空间内是否存在手指。然而，类似于 CapSense 按键，接近传感器的实际输出是一个 ON/OFF 状态。根据传感器的结构，在几厘米到几十厘米的距离内，接近感应传感器可以检测到一个手指。目前，PSoC Creator 中的 CapSense 组件仅支持基于自电容的接近感应传感器。组件的以后版本将支持基于互电容的接近感应传感器。

相比于按键与滑条，接近感应器需要感应更远距离处的电场变化。这便要求更大的传感器面积。然而，更大的传感器面积会产生较大的寄生电容 C_P，并且很难检测到该电容。因此要求在距离较远时，传感器有较高的电场强度，同时面积要小。使用宽度为 2-3 mm 的走线围绕其他传感器，如图 2-18 所示。

图 2-18. 接近感应传感器



实现接近传感器的另一种方法是将多个传感器放在一起。具体情况为：通过固件将多个传感器板组合成一个较大的传感器。该方法的缺点是会产生较大的寄生电容。请参阅 CapSense 组件数据手册，了解详细信息。

欲了解更多有关接近感应传感器的信息，请参考 AN92239 — 基于 CapSense 的接近感应和 CapSense 入门设计指南中‘接近感应’一节的内容。

2.5 防水功能

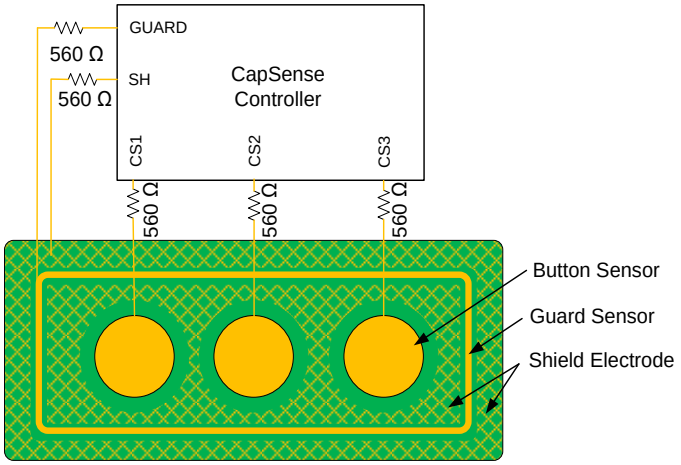
电容式感应应用于各种应用，如：家用电器、汽车级和工业级应用。这些应用要求在有薄雾、湿气、水、冰和湿度变化等环境下仍能进行强大的电容式感应操作。在电容式感应的应用设计中，当传感器表面上存在液体薄膜或液滴时，由于液体的导电性，可能会导致触摸或接近感应的误触发。赛普拉斯的 CSD 感应方法可以补偿上述原因所导致原始计数变化，并能够提供一个强大可靠的电容式感应应用的操作。

图 2-19. 洗衣机中基于 CapSense 的防水触摸用户界面



为了能补偿薄雾、湿气和湿度变化所导致的原始计数化，CapSense 感应方法连续调整传感器的基准线，以防止发生误触摸。要想补偿原始计数因液滴或水流导致的变化，用户应该安装一个屏蔽电极和保护传感器，以提供强大的触摸感应功能，如图 2-20 所示。当安装屏蔽电极时，即使传感器表面上存在水滴，CapSense 仍能正常工作并报告传感器的 ON/OFF 状态。存在水流时，保护传感器将检测是否存在水流，并不扫描传感器。因此，也不会报告传感器的 ON/OFF 状态。

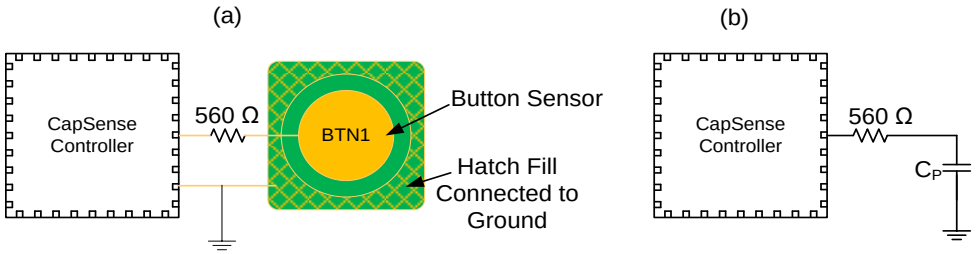
图 2-20. 将屏蔽电极（SH）和保护传感器（GUARD）连接至 CapSense 控制器



2.5.1 水滴和水流对 CapSense 传感器产生的影响

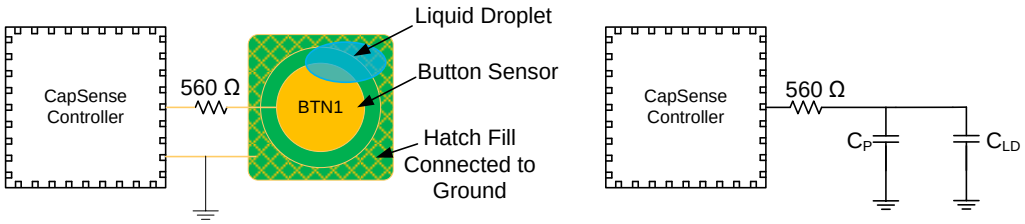
为了了解水对 CapSense 传感器的影响，应该考虑一个 Capsense 系统，其中围绕传感器的填充网格被接地，如图 2-21(a)所示。填充网格接地时会提高传感器的抗噪能力。传感器的寄生电容在图 2-21(b)中被标记为 C_P。

图 2-21. 典型的 CapSense 系统布局



如图 2-22 中所示，当水滴滴在传感器表面上时，由于其导电性，它会为电场线提供强烈的耦合路径，从而使它返回接地；这样便引进了一个与 C_P 并联的 C_{LD} 电容。该电容从 AMUX 总线吸取了额外电荷（如 GPIO 单元中电容-电流转换器一节中所述），使得传感器的原始计数增加。在某些情况下（如咸水或含有矿物质的水），原始计数因水滴滴在传感器表面上引起的增量会等于手指触摸导致的增量，如图 2-23 所示。此时，可能会发生传感器误触发。

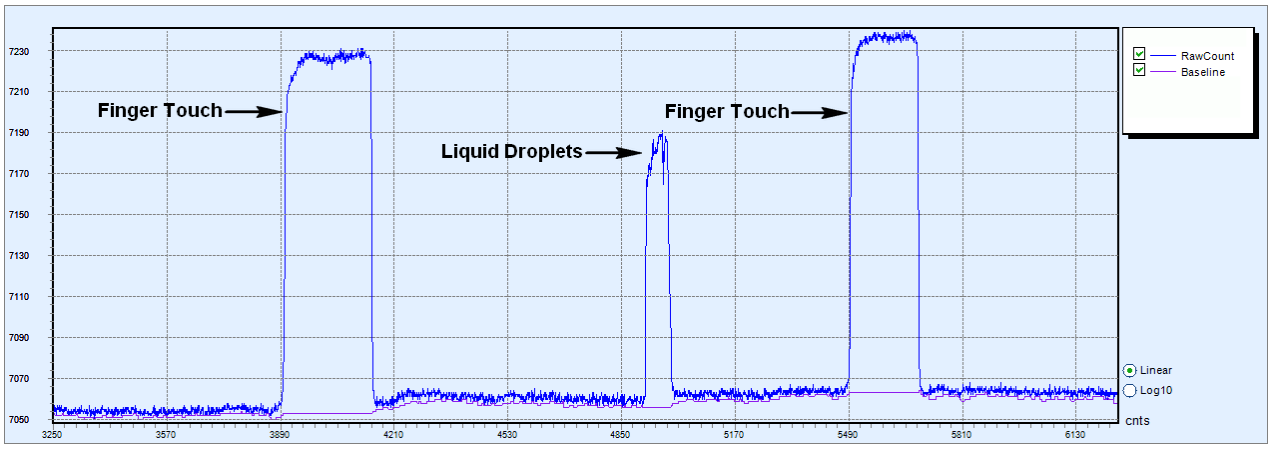
图 2-22. 填充网格接地时水滴所引入的电容



C_P — 传感器的寄生电容

C_{LD} — 水滴所引起的电容

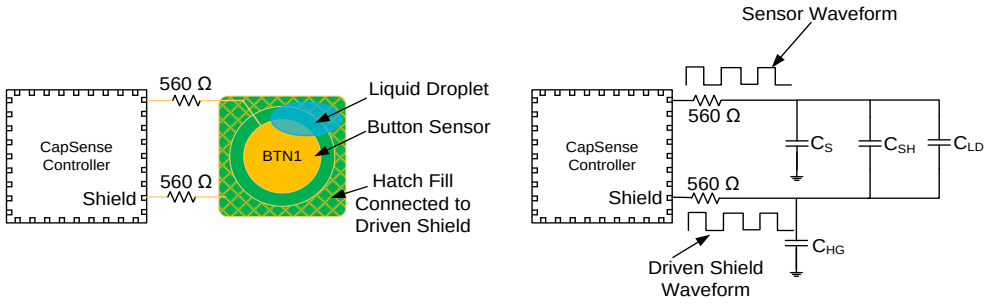
图 2-23. 水滴在围绕传感器的填充网格接地时产生的影响



为了消除水滴所添加的电容对 CapSense 电路产生的影响，您需要将围绕传感器的填充网格连接到驱动屏蔽信号。

如图 2-24 所示，当围绕传感器的填充网格连接到驱动屏蔽信号，并在水滴滴在触摸表面时，水滴两侧的电压保持相同。由于水滴所添加的电容 C_{LD} 不会从 AMUX 总线吸收任何额外电荷，因此电容 C_{LD} 的影响会被消除。从而，水滴滴在传感器上时所导致的原始计数的增量很小，如图 2-25 所示。

图 2-24. 围绕传感器的填充网格连接到驱动屏蔽信号时水滴所引起的电容



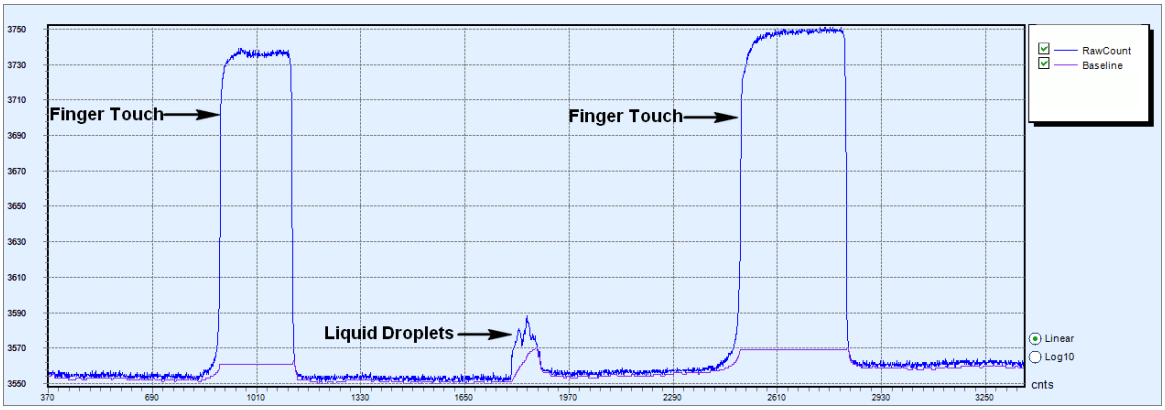
C_S — 传感器的寄生电容

C_{SH} — 传感器和填充网格之间的电容

C_{HG} — 填充网格和地面之间的电容

C_{LD} — 水滴所引起的电容

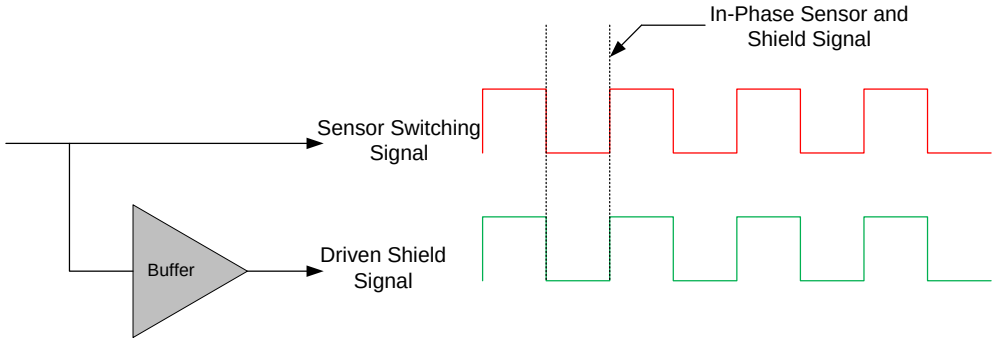
图 2-25. 水滴在围绕传感器的填充网格连接到驱动屏蔽信号时产生的影响



2.5.2 驱动屏蔽信号和屏蔽电极

驱动屏蔽信号是传感器输出被缓冲后的信号，如图 2-26 所示。驱动屏蔽信号和传感器开关信号的振幅、频率和相位相同。缓冲区为驱动屏蔽信号提供了足够大的电流，以便将填充网格的寄生电容驱动为高电平。当围绕传感器的填充网格连接到驱动屏蔽信号时，它被称为屏蔽电极。

图 2-26. 驱动屏蔽信号



屏蔽电极用于下述各目的：

- 实现具有防水功能的 CapSense 设计：屏蔽电极有助于使得 CapSense 设计防水，如上面内容所述。
- 提高存在浮动/接地导电物体时的接近感应距离：当屏蔽电极放置在接近感应传感器和浮动/接地导电物体间，它会降低这些物体对接近感应距离的影响，并有助于获得较大的接近感应距离。请参见 CapSense 入门设计指南中‘接近感应’一节的内容，了解更多详细信息。
- 降低传感器的寄生电容：当 CapSense 传感器具有较长的走线时，由于从传感器走线耦合到周围地面的传感器电场线会增加，因此传感器的 C_P 值会很高。通过实现屏蔽电极，会降低电场线耦合到地面的电容，从而降低传感器的 C_P 值。

请参考耐水性布局指导一节，了解屏蔽电极的布局指南。

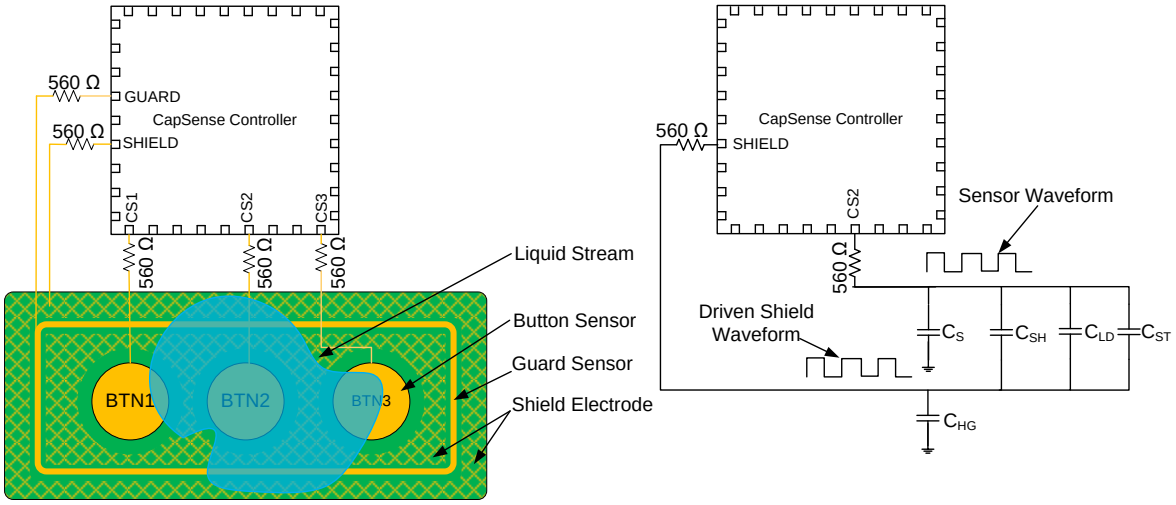
2.5.3 保护传感器

当传感器界面上存在水流时，水流会对 CapSense 传感器添加一个较大的容值（C_{ST}）。此电容可能比 C_{LD} 大数倍。正因为如此，屏蔽电极的影响完全被忽略，传感器的原始计数将与手指触值相同，甚至更高。在这些情况下，通过一个保护传感器，可以防止传感器发生误触发事件。

保护传感器是围绕 PCB 上所有传感器的铜走线，如图 2-27 所示。保护传感器类似于按键传感器，并用于检测是否存在水流。保护传感器触发时，固件会禁用扫描所有其他传感器（保护传感器除外），以便防止传感器发生误触发。

注意：由于保护传感器触发时不会扫描其他传感器，因此在触摸屏上存在水流时检测不到触摸。

图 2-27.存在水流时的电容测量

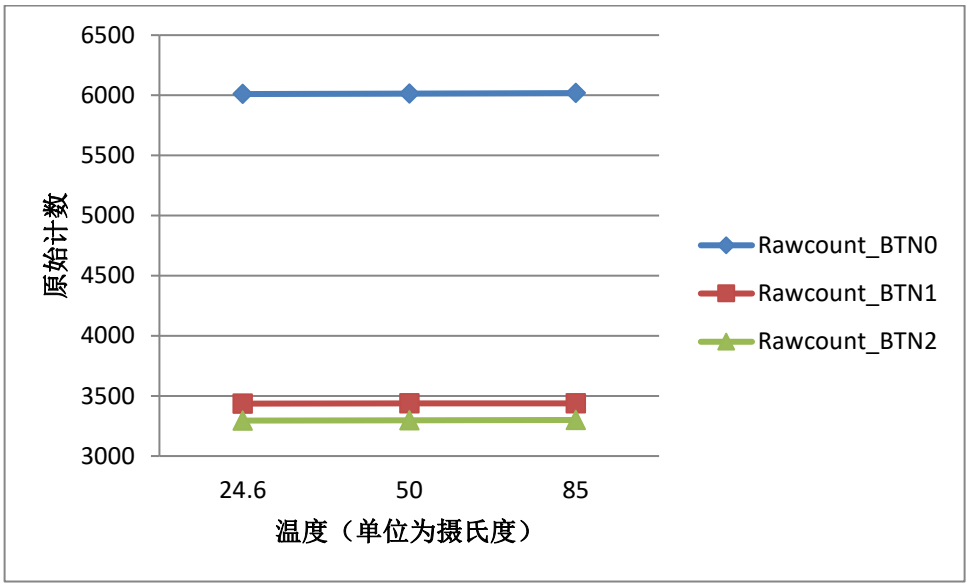


2.5.3.1 水属性对防水功能的影响

在某些应用中，CapSense 系统必须在存在不同液体（如肥皂水、海水和矿泉水）的环境中工作。此时，始终推荐通过考虑水滴导致最差情况的信号来调整传感器的 CapSense 参数。为了模拟最差的条件，建议通过将 40 克的食用盐（氯化钠）溶解在一公升水来测试传感器的防水性能。使用肥皂水也可以进行测试；结果证明肥皂水的影响和盐水的影响相同。因此，如果经过调校后可以抵抗盐水的干扰，则 Capsense 系统即使在肥皂水的环境中也可以工作。

在某些应用中（如电磁炉），在 CapSense 触摸屏上可能会附着热水。要想确定水滴温度对 CapSense 性能的影响，请在传感器上滴上不同温度的水滴，这样可以监控相应原始计数的变化。实验结果表明，热水滴和室温下的水滴产生的影响是相同的，如图 2-28 所示。这是因为热水滴的温度在落到触摸屏时会立即下降到室温。如果热水连续落在传感器上，并且覆盖层的温度因此而升高，则通过基准线算法补偿温度上升所引起的原始计数的增量，从而能够防止传感器的误触发。

图 2-28. 原始计数的变化与水温



要实现防水性设计，请按照以下步骤进行操作：

1. 如果您的应用要求抵抗水滴的干扰，请使用一个屏蔽电极。如果您的应用要求抵抗水流和水滴的干扰，则使用屏蔽电极和保护传感器。请按照耐水性布局指导章节中介绍的原理图和布局指南来设置屏蔽电极和保护传感器。
2. 在 CapSense 组件中，使能驱动屏蔽信号并将 **Inactive sensor connection** 选项设置为 **Shield**。
3. 如果使用了 SmartSense 算法，请设置保护传感器的手指电容（若使用），使得该传感器仅在触摸屏上存在水流时被触发。

   如果使用了手动调校方式，则需要设置保护传感器的灵敏度，使得该传感器仅在触摸屏上存在水流时被触发。

3 PSoC 4 和 PSoC 6 MCU CapSense

本章介绍的是如何在 PSoC 4 和 PSoC 6 MCU 中实现 CapSense CSD 和 CSX。请查看电容式触摸感应方法，了解 CapSense 的基本原理。有关 PSoC 4 器件架构的知识是本节的首先要具备的知识。如果您尚未了解 PSoC 4，请参考 AN79953 — PSoC 4 入门手册或 AN79953 — PSoC 4 BLE 入门手册以加深对该产品的了解。

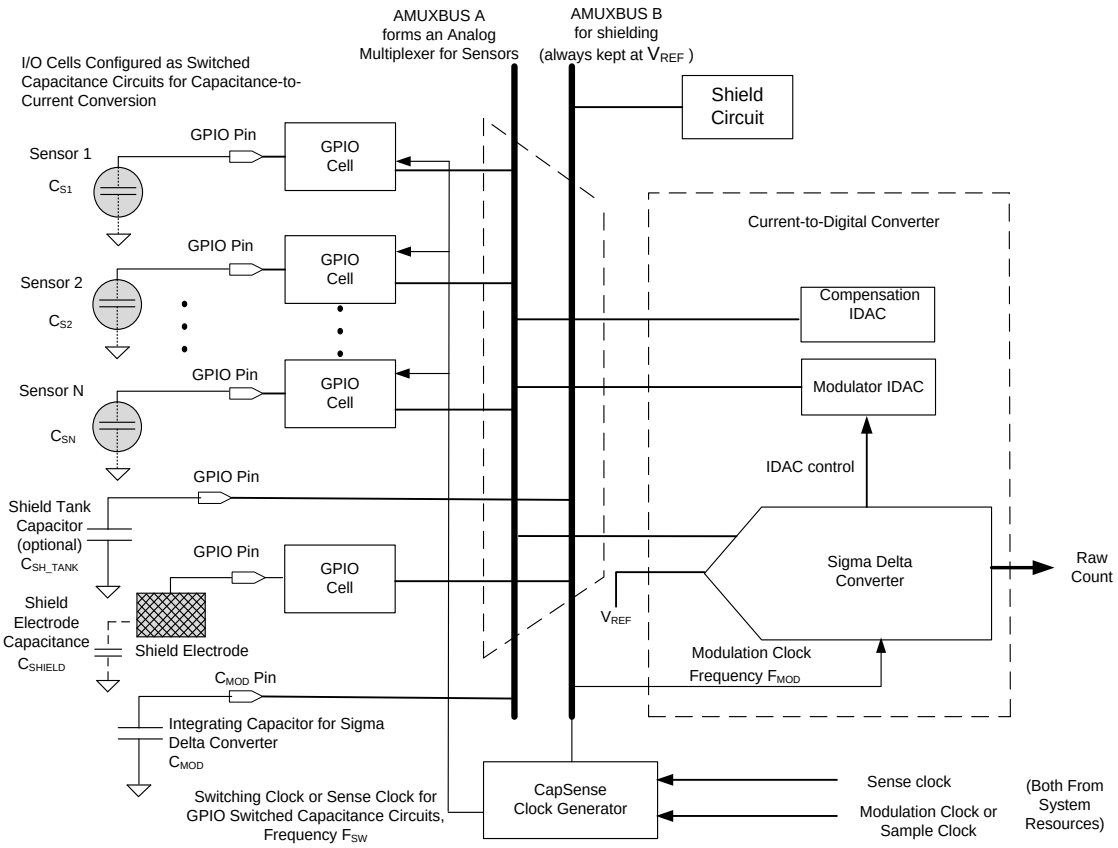
如果您使用了组件的自动调校功能（SmartSense），您可跳过该节。请参考 CapSense 功能调校章节，查找详细信息。

PSoC 4 器件系列拥有两种不同的 CapSense 构架。第 3.1 章节介绍了 PSoC 4000、PSoC 4200、PSoC 4200 BLE、PRoC BLE、PSoC 4200M 和 PSoC 4200L 器件间的架构区别，而 3.2.1 章节介绍了 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 器件系列间的 CapSense 构架区别。

3.1 CapSense CSD 感应方式

图 3-1 显示的是 CapSense 模块，该模块扫描了 CSD 感应模式下的 CapSense 传感器。

图 3-1. CapSense CSD 感应

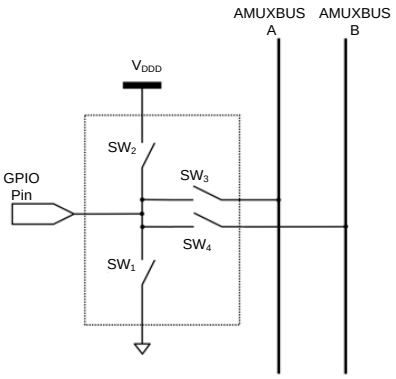


根据电容式触摸感应方法一节的内容，该模块的工作原理为：首先，将传感器的电容转换成等效的电流，然后，模拟复用器会选择其中一个电流信号，并将其送到电流-数字转换器。该电流-数字转换器包含一个 sigma-delta 转转器，能够控制调制 IDAC，使得在一段特定时间内 IDAC 的总拉电流或总灌电流与传感器电容的总拉电流或总灌电流相同。Sigma-delta 转换器的数字计数输出是传感器电容的指标，它被称为原始计数。可以在 IDAC 源模式中或在 IDAC 灌模式中对该模块进行配置。在 IDAC 源模式中，IDAC 为 AMUXBUS 提供电流，而 GPIO 单元从 AMUXBUS 吸收电流。在 IDAC 灌模式中，IDAC 从 AMUXBUS 吸收电流，而 GPIO 单元为 AMUXBUS 提供电流。

3.1.1 GPIO 单元中电容-电流转换器

在 CapSense CSD 系统中，GPIO 单元被配置为开关电容电路，用以将传感器电容转换成等效的电流。图 3-2 显示的是 GPIO 单元结构的简化框图。

图 3-2. GPIO 单元结构

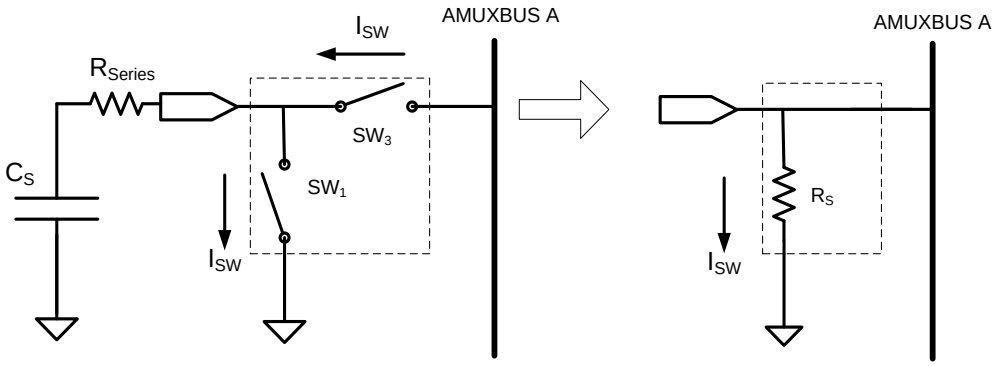


PSoC 4 和 PSoC 6 器件具备两条模拟复用器总线，其中：AMUXBUS A 适用于 CSD 感应；AMUXBUS B 适用于 CapSense CSD 屏蔽。GPIO 开关电容电路有两种配置：供给 AMUXBUS A 的拉电流或来自 AMUXBUS A 的灌电流。

3.1.2 IDAC 源模式

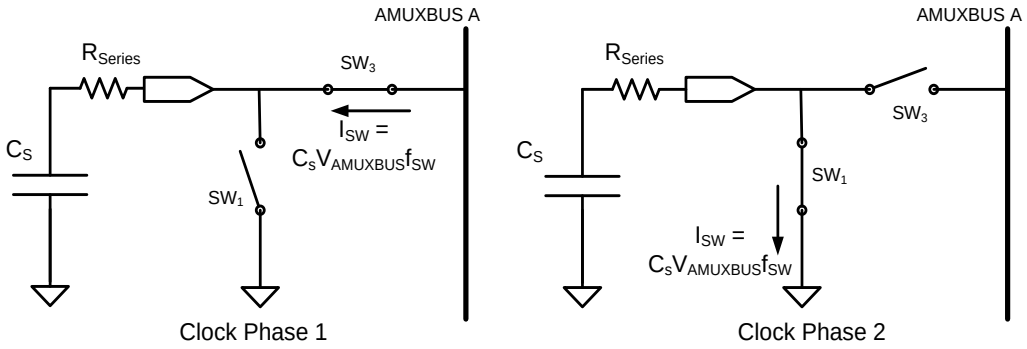
在 IDAC 源模式中，通过开关电容电路，GPIO 单元从 AMUXBUS A 吸收电流，如图 3-3 所示。

图 3-3. 从 AMUXBUS A 吸收电流的 GPIO 单元



两个非重叠、非重相位时钟的频率 F_{SW} 分别控制着开关 SW_1 和 SW_3，如图 3-4 所示。SW_1 和 SW_3 的连续开关形成了一个等效电阻 R_S，如图 3-3 所示。

图 3-4. 非重叠方式的 SW_1 和 SW_3



如果开关在足够低的频率 f_{SW} 运行，也就是说，传感器满充到 V_{REF} 和满放至接地电压仅用了 $T_{SW}/2$ 的时间，如图 3-4 所示，那么可按照公式 3-1 计算等效电阻 R_S 的值。

公式 3-1. 传感器等效电阻

$$R_S = \frac{1}{C_S F_{SW}}$$

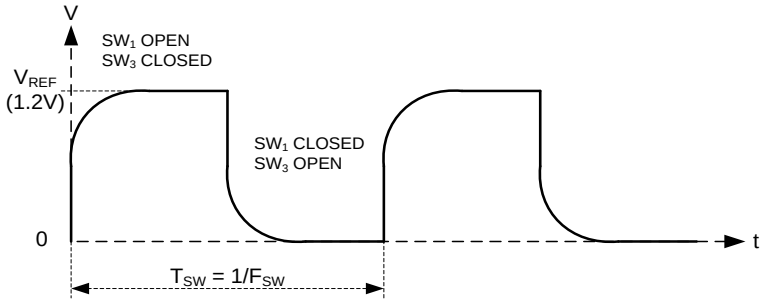
其中：

C_S = 传感器电容

F_{SW} = 感应时钟频率

Sigma-Delta 转换器将 AMUXBUS A 的电压保持为常量 V_{REF}（该过程如 Sigma Delta 转换器一节中所述）。图 3-5 显示的是得到的 C_S 的电压波形。

图 3-5.传感器电容的电压



通过公式 3-2 可以计算出从 AMUXBUS A 输出电流的平均值。

公式 3-2. 从 AMUXBUS A 通过 CapSense 传感器吸收电流以供给 GPIO 的平均电流（I_{CS}）

$$I_{CS} = C_S F_{SW} V_{REF}$$

3.1.3 IDAC 灌电流模式

在 IDAC 灌模式中，GPIO 单元通过开关电容电路为 AMUXBUS A 提供电流，如图 3-6 所示。图 3-7 显示的是传感器电容的电压波形。

由于该模式通过 VDDD 直接给 AMUXBUS A 充电，因此与 IDAC 源模式相比，该模式很容易受到电源噪声的影响。因此，建议将此模式使用一个 LDO 或一个非常稳定且低噪声的 VDDD。

图 3-6. 给 AMUXBUS A 提供电流的 GPIO 单元

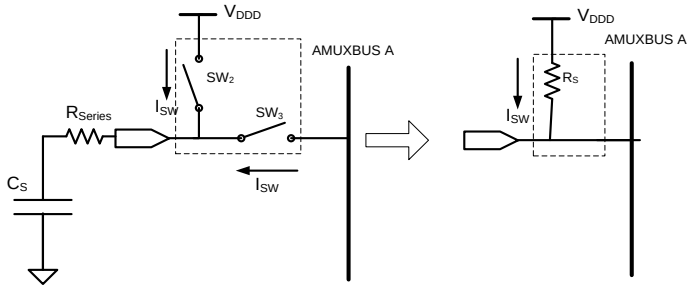
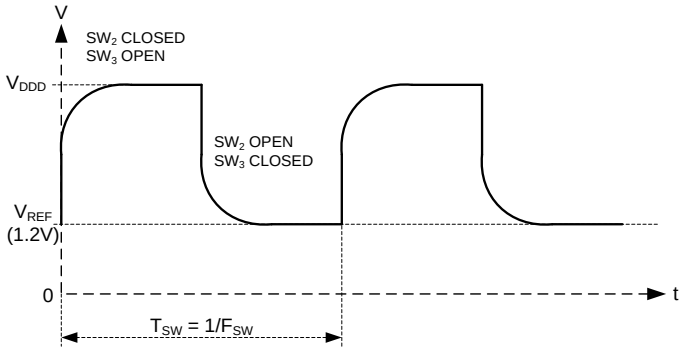


图 3-7. 传感器电容的电压



公式 3-3，可以计算得出提供给 AMUXBUS A 的平均电流。

公式 3-3. 从 GPIO 通过 CapSense 传感器供给 AMUXBUS A 的平均电流（I_{CS}）

$$I_{CS} = C_S F_{SW} (V_{DDD} - V_{REF})$$

3.1.4 CapSense 时钟发生器

根据 PSoC 器件系列，该模块从高频率系统时钟 HFCLK 分别生成感应时钟 F_{SW} 和调制时钟 F_{MOD}，如图 3-1 所示。

3.1.4.1 感应时钟

感应时钟还被称为开关时钟， 它将非重叠时钟驱动为 GPIO 单元的开关电容电路，以实现 GPIO 单元中电容-电流转换器。

感应时钟输出的选项共有三个：直接输出、8 位伪随机序列（PRS）输出和 12 位 PRS 输出。一些 PSoC 4 和 PSoC 6 器件还支持额外的扩频时钟（SSCx）模式。更多有关 PSoC 器件的支持模式信息，请参考 CapSense 组件数据手册。

直接时钟是恒定频率感应时钟源。选择该选项时，传感器引脚将切换为恒定频率时钟（其频率由 CapSense 组件配置窗口中指定）。

伪随机序列（PRS）时钟表示感应时钟由某个 PRS 模块驱动，该 PRS 模块可以生成 8 位或 12 位 PRS。PRS 时钟通过对输入时钟进行分频将感应时钟的频率扩展到较宽的频率范围。

SSCx 还扩展感应时钟的频率。它提供更好的抗噪能力并减少电磁辐射干扰。

请参考感应时钟的相关参数一节，了解时钟源和频率选择指南的详细信息。

3.1.4.2 调制器时钟

调制时钟由 Sigma Delta 转换器使用。该时钟根据以下公式配置传感器扫描时间。

公式 3-4. 传感器扫描时间

$$Sensor\ scan\ time = Hardware\ scan\ time + Sensor\ Initialization\ time$$

公式 3-5. 硬件扫描时间

$$Hardware\ scan\ time = \left(2^{Resolution} - 1\right) / Modulator\ Clock\ Frequency$$

其中，“分辨率”即扫描分辨率。

3.1.5 Sigma Delta 转换器

Sigma Delta 转换器将输入电流转换为一个相应的数字计数值。该转换器包括一个 Sigma-delta 转换器、一个时钟发生器和两个源电流/灌电流数模转换器（IDAC），如图 3-1 所示。

Sigma-delta 调制器以打开/关闭方式来控制调制器 IDAC 的电流。补偿 IDAC 始终处于打开或关闭状态。

Sigma-delta 转换器可在单 IDAC 模式或双 IDAC 模式下运行。

- 在**单 IDAC 模式下**，调制 IDAC 由 Sigma-delta 调制器控制；补偿 IDAC 始终为 OFF 状态。
- 在**双 IDAC 模式下**，调制 IDAC 由 Sigma-delta 调制器控制；补偿 IDAC 始终为 ON 状态。

Sigma-delta 转换器还要求一个外部集成电容（即调制电容 C_{MOD}），如图 3-1 所示。C_{MOD} 的建议值为 2.2 nF。

Sigma-delta 调制器保持 C_{MOD} 上的电压等于 V_{REF}。它在下列某种模式下工作：

- IDAC 源模式：在该模式中，开关电容电路将通过 AMUXBUS A 接收 CMOD 的电流，然后，IDAC 为 AMUXBUS A 供电，以使其电压平衡。
- IDAC 灌电流模式 ：在该模式中，IDAC 将通过 AMUXBUS A 接收 CMOD 的电流，然后，开关电容电路为 AMUXBUS A 供电，以使其电压平衡。

在这两种情况下，IDAC 电流根据 C_{MOD} 上微弱的电压变化在 ON 和 OFF 状态间进行切换，以保持 C_{MOD} 的电压为 V_{REF}。

Sigma delta 转换器的工作范围为 8 位到 16 位分辨率。在**单 IDAC 模式**下，原始计数值与传感器电容值成正比。如果“N”是 Sigma-delta 转换器的分辨率，I_{MOD} 是调制 IDAC 电流的值，那么，可通过公式 3-6 计算出 IDAC 源电流模式中原始计数的近似值。

公式 3-6. 单 IDAC 源电流模式中的原始计数

$$raw\ count = (2^N - 1)\frac{V_{REF}\ F_{SW}}{I_{MOD}}\ C_S$$

同样，IDAC 灌电流模式中原始计数的近似值为：

公式 3-7. 单 IDAC 灌电流模式中的原始计数

$$\text{raw count} = (2^N - 1)\frac{(V_{DD} - V_{REF})\, F_{SW}}{I_{MOD}}\, C_S$$

在这两种情况下，原始计数值均与传感器电容值 C_S 成正比。随后，CapSense CSD 固件将处理该原始计数，以便检测触摸。要想得到可靠的触摸检测，应该使用最佳值调试各个硬件参数（如 I_{MOD}、I_{COMP} 和 F_{SW}）和固件参数。欲深入了解调试的信息，请参阅 CapSense 功能调校一节的内容。

在**双 IDAC 模式**下，补偿 IDAC 始终处于打开状态。如果 I_{COMP} 是补偿 IDAC 电流，则通过下面的公式可计算出 IDAC 源电流模式下的原始计数值：

公式 3-8. 双 IDAC 源电流模式中的原始计数

$$\text{raw count} = (2^N - 1)\frac{V_{REF}\, F_{SW}}{I_{MOD}}\, C_S - (2^N - 1)\frac{I_{COMP}}{I_{MOD}}$$

IDAC 灌电流模式下的原始计数值可通过公式公式 3-9 计算得出。

公式 3-9. 双 IDAC 灌电流模式中的原始计数

$$\text{raw count} = (2^N - 1)\frac{(V_{DD} - V_{REF})\, F_{SW}}{I_{MOD}}\, C_S - (2^N - 1)\frac{I_{COMP}}{I_{MOD}}$$

请注意，原始计数值始终为正值。这是一个必要条件，用于确保在 IDAC 灌电流模式中，I_{COMP} 小于$(V_{DD} - V_{REF})\, C_S\, F_{SW}$，并且在 IDAC 源电流模式中，$I_{COMP}$ 小于$C_S\, F_{SW} V_{REF}$。如果$I_{COMP} > V_{REF} C_S\, F_{SW}$，则公式 3-8 不可用，且如果$I_{COMP} > (V_{DD} - V_{REF})\, C_S\, F_{SW}$，则公式 3-9 不可用；在这些情况下，原始计数值将为“0”。

表 3-1 说明了上述公式涉及的参数与 CapSense 组件参数间的关系。

表 3-1. CapSense 原始计数与 CapSense 硬件参数间的关系

SI. 序号	参数	说明	说明
1	N	扫描分辨率	扫描分辨率的配置范围为 8 位到 16 位
2	V_{REF}	不适用	V_{REF} 值为 1.2 V，或根据 PSoC 器件系列可配置为 0.6 V 到 V_{DDA}-0.6 V 间。请参阅 CapSense 组件数据手册，了解详细信息。
3	F_{SW}	感应时钟频率 感应时钟源	感应时钟频率和感应时钟源确定了传感器的切换频率。 更多信息，请参考感应时钟一节。
4	I_{MOD}	调制器 IDAC	I_{MOD} = 调制 IDAC 电流
5	I_{COMP}	补偿 IDAC	I_{COMP} = 补充 IDAC 电流
6	V_{DD}	不适用	该参数是器件的供电电压。
7	C_S	不适用	该参数是传感器的寄生电容。
8	不可用	调制器时钟频率	调制器时钟分频器不会影响原始计数。 请参见调制器时钟一节了解更多信息

3.1.6 模拟复用器

Sigma-delta 转换器每次扫描一个传感器。模拟复用器选择一个 GPIO 单元，并将其连接至 Sigma-delta 转换器的输入端，如图 3-1 所示。AMUXBUS A 和 GPIO 单元开关共同组成该模拟复用器（请参见图 3-6 中的 SW_3）。AMUXBUS A 连接至支持 CapSense 功能的所有 GPIO。请参考您相应器件的数据手册，以获取支持 CapSense 的端口引脚列表。AMUXBUS A 还将集成电容 C_{MOD} 连接至 Sigma-delta 转换器电路。AMUXBUS B 用于屏蔽，并使能屏蔽时保持为 V_{REF}。

3.1.7 CapSense CSD 屏蔽

PSoC 4 和 PSoC 6 MCU CapSense 支持用于防水和接近感应性能的屏蔽电极。CapSense 具有一个屏蔽电路，该电路会使用传感器开关信号的副本来驱动屏蔽电极，这样可以避免传感器与屏蔽电极间潜在的差异。请参考驱动屏蔽信号和屏蔽电极和水滴和水流对 CapSense 传感器产生的影响部分，以了解它对防水功能的效果。

在感应电路中，Sigma-delta 转换器保持 AMUXBUS A 的电压等于 V_{REF}（请查看 Sigma Delta 转换器一节）。通过在 AMUXBUS A 和供电轨（根据配置，该电压可以为 V_{DD} 或接地）间切换传感器，GPIO 单元可生成传感器波形。屏蔽电路以相同的方式工作；AMUXBUS B 电压始终保持为 V_{REF}。GPIO 单元会切换 AMUXBUS B 与供电轨（等于 V_{DD} 或接地，该配置与传感器配置相同）之间的屏蔽。该过程在屏蔽电极上生成传感器开关波形的副本。

对于带有高的寄生电容的大屏蔽层，外部电容（Csh 槽电容）用于增强屏蔽电极的驱动能力。通过 CapSense 组件的 **CSD Settings** 选项卡中 **Shield tank capacitor** 选项，可以配置屏蔽驱动。

3.2 CapSense CSX 感应方式

图 3-8 显示的是 CSX 感应单路的简单图形。该方法使用了 CSD HW 的下列子模块。

- 一个 8 位 IDAC、比较器（CMP）和计数器和数字逻辑
- AMUX-A、Tx 时钟和调制器时钟，Tx 和 Rx 电极和外部电容的 V_{REF} 和端口引脚。
- 两个外部电容（CINTA 和 CINTB）

注意： PSoC 4100 不支持 CSX 感应方法。

图 3-8. CapSense CSX 感应方法配置

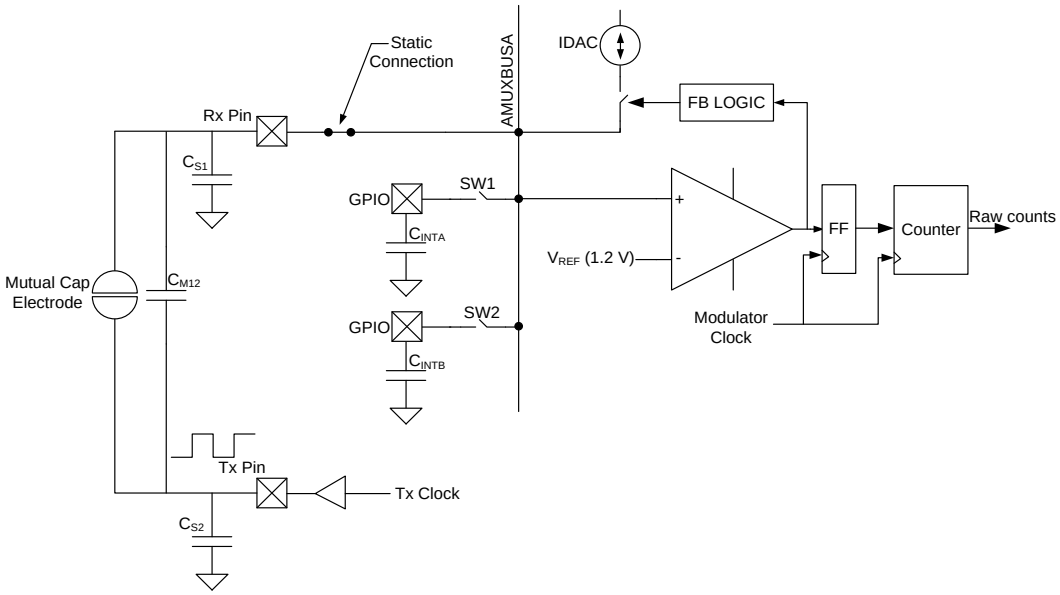
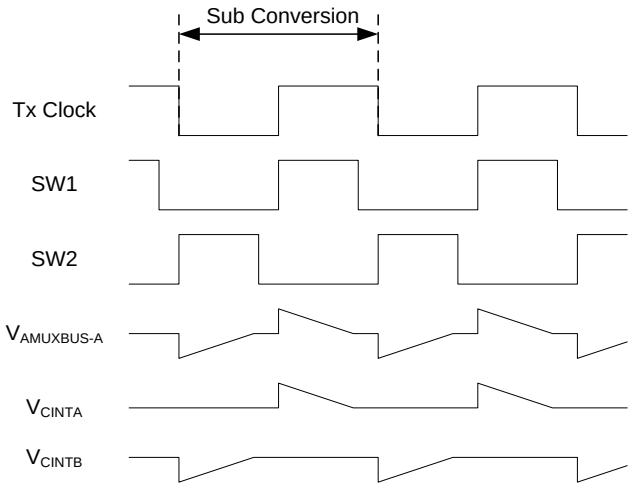


图 3-9. CSX 感应波形



CSX 感应方法用于测量 Tx 电极和 Rx 电极间的互电容，如图 3-8 所示。Tx 电极是由数字波形（Tx 时钟）激发的，该波形在 VDDIO（如果 VDDIO 不可用，则使用 VDDD）和接地间进行切换。Rx 电极被静态连接到 AMUXBUS A。CSX 方法要求使用两个外部集成电容，C_{INTA} 和 C_{INTB}。表 7-6 给出了这些电容值。

图 3-9 显示的是 Tx 电极、C_{INTA} 以及 C_{INTB} 电容上的波形。固件通过将两个外部电容上的电压初始化为 V_{REF} 并执行一系列子转换来启动采样，即生成样本的过程。一个采样中所有子转换的总和作为结果，并被称为“原始计数”。一个子转换是用于计算在一个 Tx 时钟周期中执行的转换次数的电容值。

在一个子转换中，SW1 和 SW2 的开关将在与 Tx 时钟的同一个相位进行。在 Tx 时钟的上升沿上，SW1 处于关闭状态（在这段时间内，SW2 处于打开状态），并且电荷从 Tx 电极流动到 Rx 电极。将该电荷集成到 C_{INTA} 电容上，从而使 C_{INTA} 上的电压递增。将 IDAC 配置为灌电流模式，使 C_{INTA} 上的电容等于 V_{REF} 电压。

在 Tx 时钟的下降沿上，SW2 处于关闭状态（在该时间内，SW1 处于打开状态），并且电荷从 Rx 电极流动到 Tx 电极。这样会使 C_{INTB} 上的电压小于 V_{REF}。将 IDAC 配置为源电流模式，以使 C_{INTB} 上的电压等于 V_{REF}。在两个周期中，Tx 和 Rx 电极间传输的电荷与电极间的互电容 C_M 成正比。IDAC 对外部电容进行充电或放电时，比较器输出将使能计数器。计数器将计算一个子转换中的调制器时钟周期的数量。执行多个子转换，并将该结果存储在同一个计数器内，从而生成传感器的“原始计数”。

通过调制器时钟，可以测量到在一个 Tx 时钟周期内对外部电容进行充电/放电所需的时间。因此，调制器时钟频率必须始终大于 Tx 时钟频率；调制器时钟的频率越高，准确度也越高。为了正常工作，需要设置 IDAC 电流，使 C_{INTA} 和 C_{INTB} 电容能在一个 Tx 时钟周期内完成充电或放电。CapSense 组件提供了一个自动校准 IDAC 的选项。建议使能该选项。

公式 3-10. 原始计数与互电容感应间的关系

$$\mathrm{Rawcount}_{\mathrm{Counter}} = \frac{2 * V_{TX} * F_{TX} * C_M * \mathrm{MaxCount}}{\mathrm{IDAC}}$$

$$\mathrm{MaxCount} = \frac{F_{Mod} * N_{Sub}}{F_{TX}}$$

其中：

IDAC — IDAC 电流

C_M — Tx 和 Rx 电极间的互电容

V_{TX} — Tx 信号幅度

F_{TX} – Tx 时钟频率

F_{Mod} – 调制器时钟频率

N_{Sub} – 子转换次数

将手指放在 CSX 按钮上时，Rx 和 Tx 电极间的互电容将被减小，同时降低原始计数。从硬件获取的原始计数由 CapSense 组件处理，以增加触摸事件的计数（类似于 CSD 中的计数）。您所看到的最终反转原始计数是通过公式 3-11 计算的。

公式 3-11. 确定公式 $Rawcount_{Component}$

$$Rawcount_{Component} = MaxCount - Rawcount_{Counter}$$

请参考 CSX 感应方法一节，了解更多有关上述 CSX 硬件参数的详细解释

3.2.1 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 中的 CapSense 架构

PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 中第四代 CapSense 架构是上一代 CapSense 架构的改进版本。表 3-2 列出了各个 PSoC 4 器件中 CapSense 架构间主要的区别。

表 3-2. CapSense 架构比较

特性	第三代 CapSense（PSoC 4、4-M、4-BLE 以及 4-L）	第四代 CapSense（PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU）	与第三代 Capsense 相比，第四代具有的优点
感应模式	支持自电容和互电容模式	支持自电容、互电容和 ADC 模式	硬件相同，但提供了其他 ADC 功能
传感器寄生电容（C_P）范围	5 pF ~ 60 pF	5 pF ~ 200 pF	支持高电容 C_P 的设计应用
V_{REF}	1.2 V	0.6 V ~ VDDA – 0.6 V	提高信噪比
IDAC LSB 大小	1.2 µA、2.4 µA	37.5 nA、300 nA、2.4 µA	提高灵敏度
IDAC 的分流功能	需要两个 IDAC	需要一个 IDAC	使用更少的资源实现相同的性能，并释放一个 IDAC 作为通用目的
降低电磁干扰（EMI）— 数字	–	扩频 — 由 CSD 控制	扩频时钟可通过硬件生成，CPU 处于完全空闲状态。
10 位 ADC	不支持	支持	不进行 CapSense 扫描时，可将相同 CSD 硬件作为 ADC 使用。
硬件状态机	不支持	支持	CPU 不再需要进行初始化或生成扩频感应时钟。

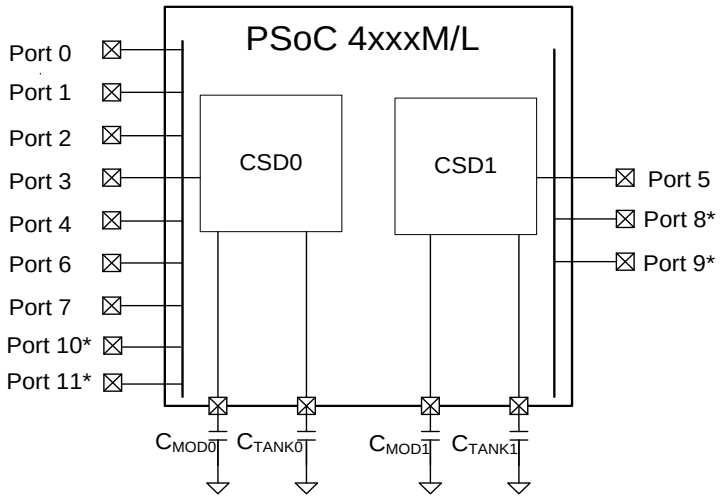
PSoC 4 S 器件系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 中的 CapSense 硬件支持基于自电容（CSD）和互电容（CSX）的电容式感应。另外，不进行 CapSense 扫描操作时，它仍能测量输入电压。

本节介绍了如何使用 CapSense 硬件来执行自电容和互电容感应。有关 CapSense 硬件的详细信息，请参考 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 器件技术参考手册中的 **CapSense** 一节。需要对 CapSense 结构有基本的了解，从而了解自电容和互电容的工作原理。

3.3 PSoC 4xxxM/4xxxL 系列中的 CapSense

PSoC 4xxxM/4xxxL 器件系列支持两个 CapSense 模块— CSD0 和 CSD1。每个模块拥有同样的功能和性能，如 CapSense CSD 感应方式一节中所述。PSoC 4xxxM 中 CSD0 和 CSD1 模块的主要区别在于：CSD0 模块可以扫描所有 GPIO 上的 CapSense 传感器（端口 5 引脚除外），而 CSD1 模块只能扫描端口 5 引脚上的 CapSense 传感器，如图 3-10 所示。

图 3-10. PSoC 4 M 系列中的 CapSense



*端口 8、9、10 和 11 仅适用于 PSoC 4xxxL 器件系列。PSoC 4xxxL 系列的端口 12 没有 CapSense 功能。

每个 CSD 模块都需要一个单独的 C_{MOD} 和 C_{SH_TANK} 电容。l 表 3-3 对 CSD0 和 CSD1 模块间的差异进行了汇总。

表 3-3. PSoC 4xxxM/L 系列中 CSD0 和 CSD1 模块间的差异

	CSD0	CSD1
C_{MOD}	请参见表 7-7 表 7-7 了解建议引脚。	
C_{SH_TANK}		
$C_{INTA/B}$		
CapSense 引脚	所有引脚（端口 5 引脚除外）（针对 PSoC 4xxxM）。 所有引脚（端口 5、8 和 9 引脚除外）（针对 PSoC 4xxxL）。	端口 5 中的所有引脚（针对 PSoC 4xxxM）。端口 5、8 和 9 中的所有引脚（针对 PSoC 4xxxL）。
屏蔽电极引脚	所有引脚（端口 5 引脚除外）（针对 PSoC 4xxxM）。 所有引脚（端口 5、8 和 9 引脚除外）（针对 PSoC 4xxxL）。	端口 5 中的所有引脚（针对 PSoC 4xxxM）。端口 5、8 和 9 中的所有引脚（针对 PSoC 4xxxL）。
CapSense 引脚最大数量	47 个（针对 PSoC 4xxxM） 68 个（针对 PSoC 4xxxL）	4 个（针对 PSoC 4xxxM） 22 个（针对 PSoC 4xxxL）

*表示最大引脚数量，不包括设计中用于 C_{MOD} 和 C_{SH_TANK} 的两个引脚。

注意：由于 CSD0 和 CSD1 模块使用不同的屏蔽电极引脚，因此需要使 CSD0 传感器的屏蔽网格与 CSD1 传感器的屏蔽网格隔离开。

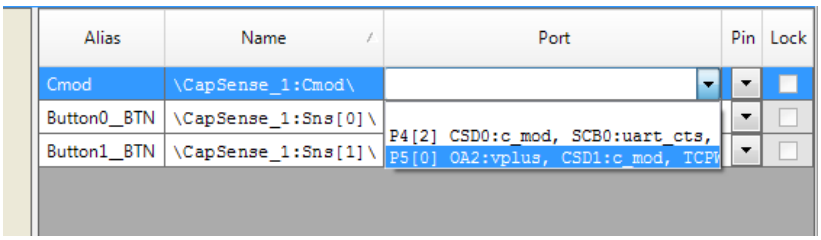
想要选择某个特定的 CSD 模块，请按照下面程序：

1. 将 CapSense CSD 组件放置在 PSoC Creator 原理图中。

   注意：CapSense v3.0 组件不支持使用 PSoC 4-M 系列中的 CSD1 模块进行感应功能。如果要使用这两个 CapSense 模块，则您应该使用 CapSense_CSD v2.40 组件。

2. 在 PSoC Creator cydwr pins 选项卡中，根据所需 CSD 模块分配 CMOD 引脚，如图 3-11 所示。例如，如果您需要使用 CSD0 模块，请选择 P4.2 作为 CMOD 引脚。

图 3-11. 选择 PSoC 4xxxM/L 系列中的 CSD0 或 CSD1 模块



3. 在同一个项目中，想要用被分配给 CSD0 和 CSD1 模块的端口上的 CapSense，请使用 CSD 组件的这两个外设 CSD0 和 CSD1。
   下面是一个示例代码段，用于同一个项目中的两个 CSD 模块：

```
/* Start CapSense Component */
    CapSense_1_Start();
    CapSense_2_Start();

    /* Initialize all baselines */
    CapSense_1_InitializeAllBaselines();
    CapSense_2_InitializeAllBaselines();


    for(;;)
    {
            /* Check that scanning is completed */
            if (0u == CapSense_1_IsBusy() && 0u == CapSense_2_IsBusy())
            {
                    /* Update all enabled baselines */
                    CapSense_1_UpdateEnabledBaselines();
                    CapSense_2_UpdateEnabledBaselines();

                    /* Start scanning all enabled sensors */
                    CapSense_1_ScanEnabledWidgets();
                    CapSense_2_ScanEnabledWidgets();
            }

     }
```

4 CapSense 设计和开发工具

赛普拉斯提供了一组完整的硬件和软件工具，有助于开发您的赛普拉斯应用。

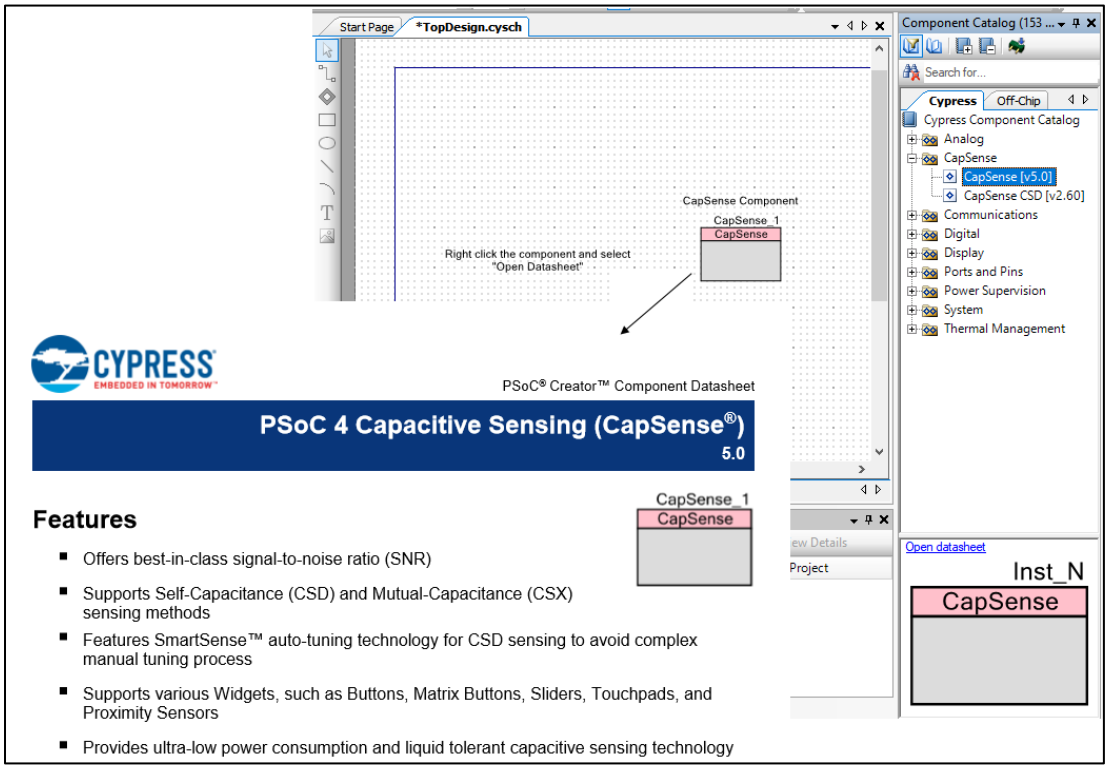
4.1 PSoC Creator

PSoC Creator 是最先进、简单易用的集成开发环境。它根据传统的原理图输入项提供了硬件配置和软件开发的独特组合。通过使用一个组件库，可以在拖放式设计环境中开发您的应用。更多详细信息，请参考 PSoC Creator 主页。

4.1.1 CapSense 组件

PSoC Creator 提供了 CapSense 组件，只要通过配置该组件，即可在 PSoC 中创建一个电容式触摸系统。该组件还提供了一个应用编程接口（API），用于简化固件开发。某些 PSoC 4 BLE 和 PSoC 6 MCU 器件还支持一个 CapSense Gesture 组件（请参考您相应器件的数据手册，以确定您的器件是否支持该组件）。

图 4-1. PSoC Creator 组件放置



每个组件都有一个相应的数据手册，它介绍了组件的详细信息。要想打开组件数据手册，请右击组件，然后选择 **Open Datasheet** 项。

CapSense 组件还包含一个 Tuner GUI（被称为调谐器助手），有助于执行调试过程。

4.1.2 CapSense_ADC 组件

CapSense_ADC 组件仅适用 PSoC 4 S 器件系列、PSoC 4100S Plus 和 SoC 4100PS 器件。需要执行 CapSense 和 ADC 操作时，应该使用该组件。使用 CapSense 模块执行 ADC 操作，以时间复用方式执行触摸功能。

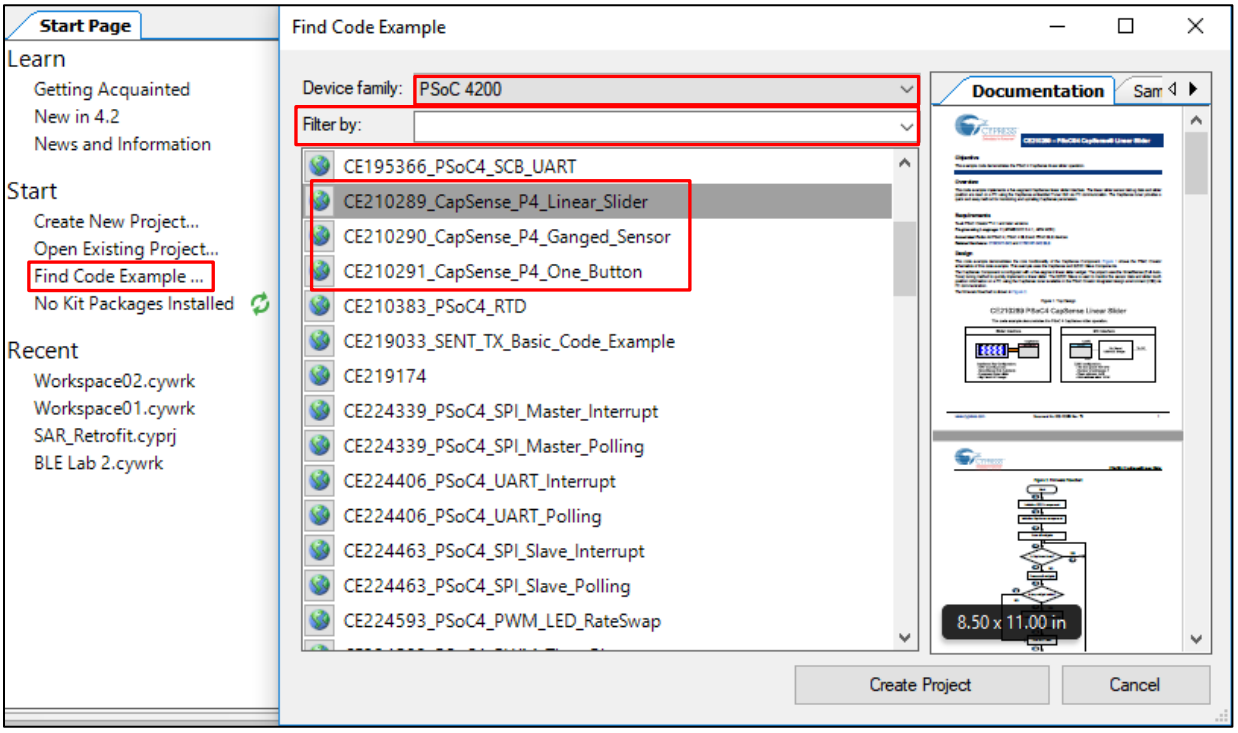
4.1.3 调谐器助手

CapSense 组件包含了调谐器助手，该助手帮助调校 CapSense 参数和监控原始计数、基准线以及计数差值等传感器数据。请参考相应的 CapSense 组件数据手册，了解调谐器助手的使用程序。

4.1.4 示例项目

您可以通过使用 PSoC Creator 中提供的 CapSense 示例项目学习原理图输入和固件开发。如果要查找 CapSense 实例项目，请进入 PSoC Creator 起始页，然后点击 **Find Code Example...**项，并选择相应架构，如图 4-2 所示。您还可以通过将某个项目的一部分名称或整个名称写入到 **Filter by** 框内来过滤该项目。

图 4-2. PSoC Creator 示例项目



4.2 ModusToolbox

赛普拉斯推出了ModusToolbox软件套件，用于开发基于PSoC 6的CapSense应用。您可以从此处下载ModusToolbox。开始使用该软件之前，赛普拉斯建议您先参阅快速入门指南和用户指南。如果您的系统安装了 ModusToolbox IDE，则可以创建一个 CapSense 应用。CapSense®配置器指南介绍了简单 CapSense 线性滑条示例的各开发步骤。您也可以参考 CapSense®调谐器指南来调试您的 CapSense 设计。

请注意，PSoC Creator 不支持所有 PSoC 6 器件。仅 ModusToolbox 支持 CY8C62x8 和 CY8C62xA 器件系列。ModusToolbox 和 PSoC Creator 4.2 支持 CY8C6xx7 器件系列。

4.3 硬件套件

表 4-1 列出了各种开发套件，这些开发套件支持 PSoC 4 和 PSoC 6 CapSense 的评估。

表 4-1. PSoC 4 和 PSoC 6 CapSense 开发套件

开发套件	支持 CapSense 性能
PSoC 4000 Pioneer 套件（CY8CKIT-040）	一个 5x6 CapSense 触摸板和一个连线接近感应传感器
PSoC 4 S 系列 Pioneer 套件（CY8CKIT-041）	两个自电容或互电容感应按键 一个 7×7 自电容或互电容感应触摸板
PSoC 4 S 系列原型套件（CY8CKIT-145）	三个自电容或互电容感应按键 一个 5 段自电容或互电容感应线性滑条
PSoC 4100S Plus 原型开发套件（CY8CKIT-149）	三个自电容或互电容感应按键 一个 6 segment 自电容或互电容感应线性滑条
PSoC 4 Pioneer 套件（CY8CKIT-042）	一个 5 segment 线性滑条
PSoC 4 BLE 蓝牙低功耗 Pioneer 套件（CY8CKIT-042-BLE）	一个 5 segment 线性滑条和一个连线接近感应传感器。
PSoC 4200M Pioneer 套件（CY8CKIT-044）	一个 5 segment 手势检测和两个接近连线传感器
PSoC 4200L Pioneer 套件（CY8CKIT-046）	一个 5 段手势检测、两个接近连线传感器和一个 8 元素的辐射滑条
PSoC 4100PS 原型开发套件（CY8CKIT-147）	无板上 CapSense 传感器。该套件可用于将外部传感器连接到任何 I/O 引脚上。
CapSense 接近屏蔽（CY8CKIT-024）	一个 4 segment 手势检测和一个接近环路传感器
CapSense®液体水平感应屏蔽（CY8CKIT-022）	一个 2 元素柔性 PCB 和 12 元素柔性 PCB
带有 PSoC 开发套件（CY8CKIT-001）的 PSoC 4 处理器模块（CY8CKIT-038）	一个 5 段线性滑条和两个按键
CapSense 扩展板套件（CY8CKIT-031）， 可将其用于 CY8CKIT-038 和 CY8CKIT-001	一个 10 段滑条、5 个按键和一个带有 LED 指示灯的 4x4 矩阵按键。
MiniProg3 编程和调试套件（CY8CKIT-002）	在 CY8CKIT-038 中 CapSense 的性能调试
PSoC 6 Wi-Fi BT Pioneer 套件（CY8CKIT-062-WiFi-BT Pioneer 套件）以及 PSoC 6 BLE Pioneer 套件（CY8CKIT-062-BLE Pioneer 套件）	一个 5 segment CapSense 滑条、两个 CapSense 按键、一个 CapSense 接近感应插座、一个接近感应
PSoC 6 Wi-Fi BT 原型开发套件（CY8CPROTO-063-4343W）	一个 5 segment CapSense 滑条和两个互电容 CapSense 按键

5 CapSense 功能调校

设计完接近感应传感器布局（请参见 PCB 布局指南）后，需要执行固件，并且调试 CapSense 参数，使接近感应传感器得到最佳性能。CapSense 感应方法是硬件和固件技术的组合。因此，它具有多个硬件和固件参数，用于进行正常操作。应将这些参数调整到最佳值，用以执行可靠的触摸测试和快速响应。必须手动调试市场上的多数电容式触摸解决方案。赛普拉斯为 PSoC 4 CapSense 提供了 SmartSense（也称为自动调校）这一独特性能。SmartSense 是一种固件算法，它可以自动将所有参数设置为最佳状态。

5.1 选择 SmartSense 还是手动调校

SmartSense 自动调校可以缩短设计周期时间，并为不同的 PCB 提供稳定的性能，但它需要其他的 RAM 和 CPU 资源，如组件数据手册中所述，从而允许调校 CapSense 参数的运行时间。SmartSense 主要用于简单按键和滑条 widget 的通用 CapSense 应用，目前它只支持自电容感应功能支持，但不支持互电容感应功能。

另一方面，需要通过手来调校，以获取最佳 CapSense 参数，但可以严格控制电容感应系统的性能，如响应时间和功耗。手动调校还使我们能够在通用按键和滑条应用中使用 CapSense，如接近和液体水平感应屏蔽。

建议将 SmartSense 调校用于按键和滑条 widget 的通用 CapSense 应用，这些 widget 的寄生电容（C_P）值必须处于 SmartSense 所支持的范围内。

对于按键 widget，如果手指电容大于 0.2 pF，则 SmartSense 将支持一个大小在 5 pF 到 45 pF 范围内的传感器寄生电容。如果手指电容低于 0.2 pF，但不小于 0.1 pF，则受支持的寄生电容范围为 5 pF 到 35 pF。

对于滑条 widget，每个单独滑条段的寄生电容 C_P 的范围与为按键 widget 支持的寄生电容值范围相同。另外，所有滑条段的 C_P 都必须大于滑条中最大 C_P 段 C_P 值的 75%。例如，在一个滑条中，如果最大 C_P 段的 C_P 值为 30 pF，那么其他段的 C_P 值都必须大于 22.5 pF。

如果将手动调校用于带有传感器寄生电容的 CapSense 应用，便不用遵守以上标准。您还可以使用手动调校，其中需要严格控制传感器扫描时间或其他 CapSense 参数。在这些场合中，您可以使用 SmartSense，以寻找最佳硬件参数（如感应时钟频率），然后将调校模式更改为手动调校模式用以调制 CapSense 参数。

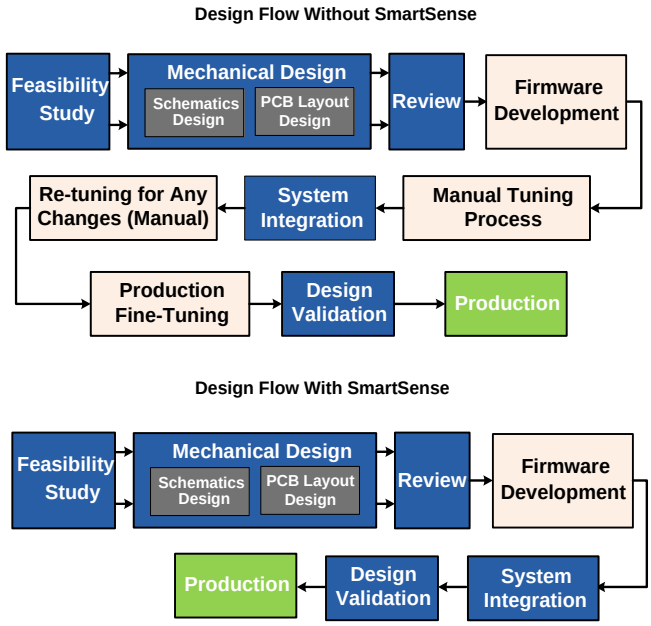
请注意，手动调校需要 I^2C 或 UART 同一台 PC 主机通信。

5.2 SmartSense

SmartSense 是一种固件算法，它可以自动将所有 CapSense 调校参数设置为最佳值。SmartSense 的部分优点（与手动调校相反）包括：

- **缩短了设计周期时长：**电容式触摸应用的设计流程涉及所有传感器的调校。如果在您的设计中有多个传感器，那么该步骤可能非常耗时。另外，如果在设计、PCB 布局或者机械设计中有变化，那么您必须重复进行调校。自动调校通过自动设置所有参数来解决该问题。图 5-1 显示的是带有 SmartSense 以及不带 SmartSense 的典型 CapSense 应用的设计流程。

图 5-1. 带有 SmartSense 以及不带有 SmartSense 的设计流程



- **各种 PCB 的性能相互独立：**个别传感器的寄生电容 CP 可能由于 PCB 制造过程中的工艺变化或多源供应链中供应商之间的误差而存在差异。如果产品批次中的 CP 明显发生变化，则必须调整每一批的 CapSense 参数。SmartSense 自动为每个器件设置参数，因此要注意 CP 的差异。
- **易于使用：**由于只需要了解 CapSense 的基本知识，所以 SmartSense 的使用更快速、更简单。

注意：可以通过多种方法使用 SmartSense：

1. **SmartSense（完全自动调校）** — 这是最快的调校方法。通过该方法，可以在运行时间中自动校准 CapSense 硬件和固件调校参数。它是我们对大部分设计推荐的调校方法。
2. **SmartSense（仅用于硬件参数）** — 该方法会自动调校 CapSense 的所有硬件参数，但允许设置用户定义的阈值。该方法消耗的闪存/RAM 资源比 SmartSense（完全自动调校）的少。此外，这种方法避免了计算自动阈值时所需要的额外处理，因此在给定的扫描速率条件下所消耗的功率少。在低功耗或有噪声的设计中或在需要受限存储的场合下，应该使用这种方法。
3. **SmartSense（用于开始调校）** — 您还可以将 SmartSense 用于开始调校，从而快速找到 CapSense 电路板的最佳设置，然后将其更改为手动调校。在严格要求响应时间或功耗的场合下，应使用这种方法。这是寻找最佳设置的快速方法，并非从头开始手动调校。

5.2.1 SmartSense 的组件配置

本节介绍的是 SmartSense 模式的组件配置。请参见手动调校一节，了解有关手动调校方面的详细信息。建议对所有较新的设计使用最新的 CapSense 组件。如果您的设计使用 CapSense 组件的更旧版本，请参阅相应的组件数据表以获取设计指南。

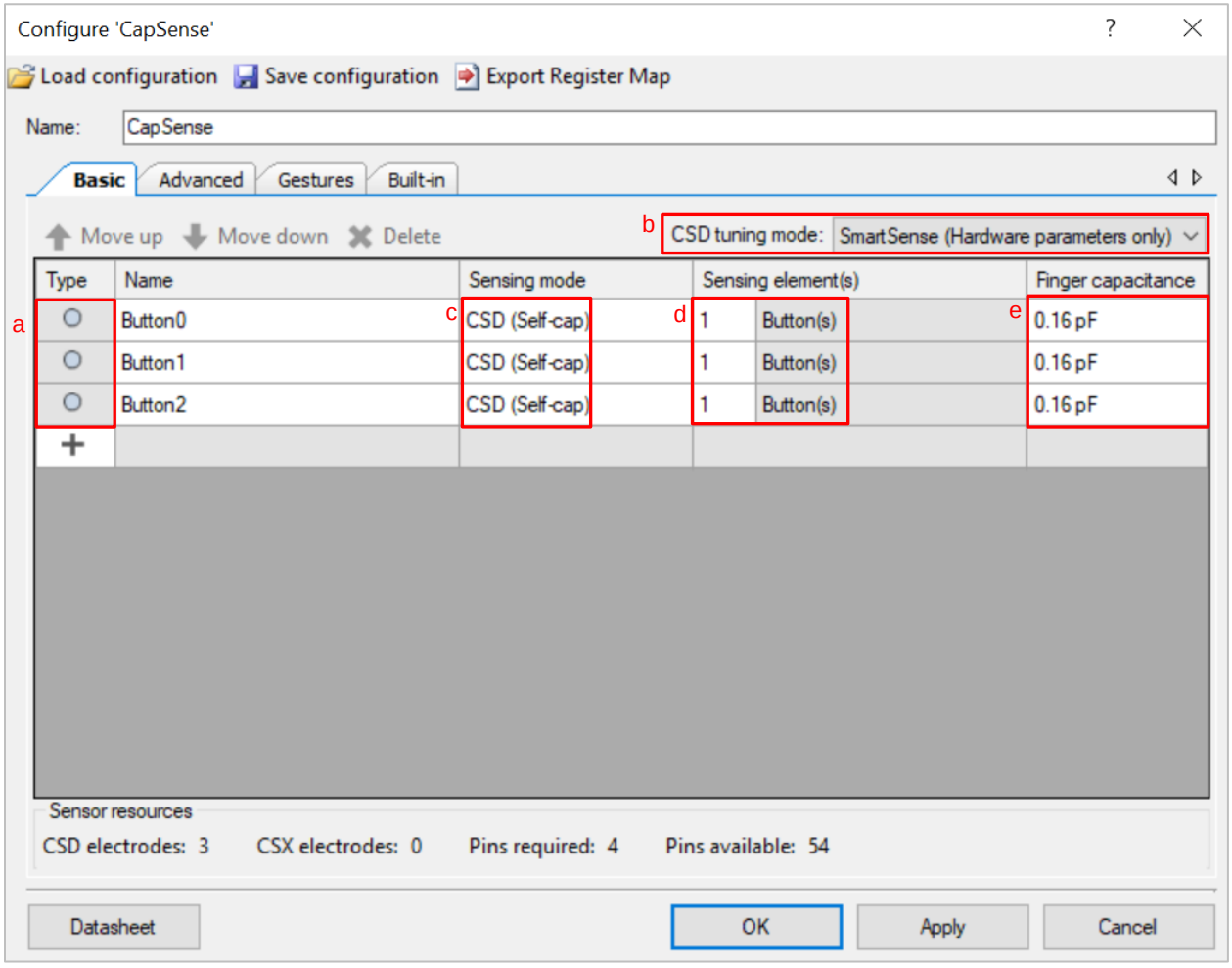
5.2.1.1 SmartSense 配置

请双击 CapSense 组件或右键点击该组件，并选择 **Configure** 项，以打开 CapSense 组件配置窗口（看图 5-2）。

5.2.1.1.1 常规设置

请按照图 5-2 中的内容设置 **Basic** 选项卡配置。

图 5-2.CapSense 组件的 **Basic** 选项卡



a. **Widget Type**：通过点击**+**符号并选择下拉列表中的 widget（如“button”、“liner slider”和“radial slider”等），以指定 widget 类型。重复该操作，直到所有 widget 被添加到组件中为止。

b. **CSD tuning mode**：要想自动调校所有硬件和软件参数，请选择 **SmartSense (Full Auto-Tune)**调校方式；否则，请选择 **SmartSense (Hardware parameters only)**选项手动控制阈值设置。表 5-1 列出了 SmartSense 根据选定模式自动设置的所有参数。

注意：目前，SmartSense 仅支持使用 **CSD (Self-cap)**的感应模式的 widget。**CSX (Mutual-cap)**的 widget 需要手动调校。

表 5-1. SmartSense 中被自动调校的 CapSense 参数

参数	计算频率	
	全自动调校模式	只有完全自动调校硬件参数模式
手指阈值	根据所选的**手指电容**在 CapSense 启动时计算一次，并在每次传感器扫描后更新。	手动选择
噪声阈值		
迟滞		
负噪声阈值		
扫描分辨率	启动 CapSense 时计算一次。	
补偿 IDAC		
调制器 IDAC		
感应时钟频率		
调制器时钟频率		
低基准线复位		

c. **Sensing mode**：根据您硬件设计的要求，为每个 widget 选择 CapSense 感应模式（CSD (Self-cap)或 CSX (Mutual-cap)。SmartSense 目前仅支持 CSD widget，不支持 CSX widget。CSX widget 需要进行手动调校。

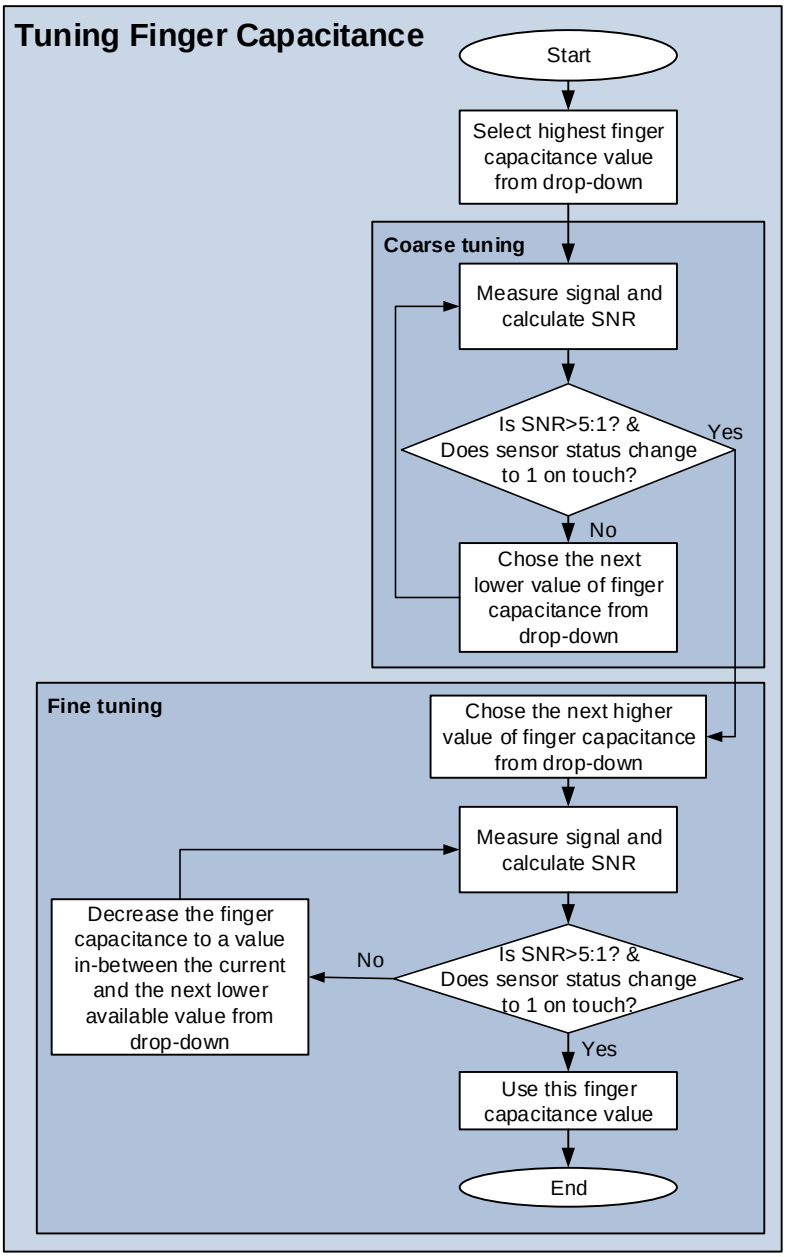
d. **Sensing element(s)**：该字段用于指定每个 CSD widget 中的传感器数量。该选项卡有助于使用多个相同的传感器（即具有相同的 C_P 和传感器位置）的应用。SmartSense 方法会将单个 widget 中的所有传感器调校为同样的灵敏度。单个 widget 中包含多个传感器会节省处理时间并降低存储器的空间。您可以配置传感器的数量，使之符合硬件设计。

e. **Finger Capacitance**：该参数指的是 CapSense 组件检测为有效触摸的最小 C_F 值。如果手指触摸按键传感器时增加的实际 C_F 小于组件配置窗口中指定的值，那么即使存在触摸，传感器状态也不会更改为 1。但如果增加的实际 C_F 高于组件配置窗口中指定的值，那么传感器状态会变为 1。

如果 C_F 值未知（可根据公式 2-1 估计 C_F 值），那么请按照下面内容设置**手指电容**：

- **For button sensors**：首先要指定最高值（从列表中的可用选项），然后检查发生触摸时 SNR 和按键的状态。逐渐降低 **Finger capacitance** 参数值，直到发生触摸时的按键状态转为 1 并且 SNR 大于 5。图 5-3 显示的是寻找设计中正确的**手指电容参数**值的详细步骤。您可以选择对工作设计的**手指电容**参数值进行粗调或细调。粗调可以满足大部分设计要求，但通过细调，您可以使用 SmartSense 选择最有效果的 CapSense 参数（例如，最短的传感器扫描时间）。如果 SNR 小于 5:1（即使选择了手指电容的允许最小值），请参考 PCB 布局指南、手动调校或调校调试常见问题解答（如 5.3.5.4、5.3.5.7 或 5.3.5.10）。

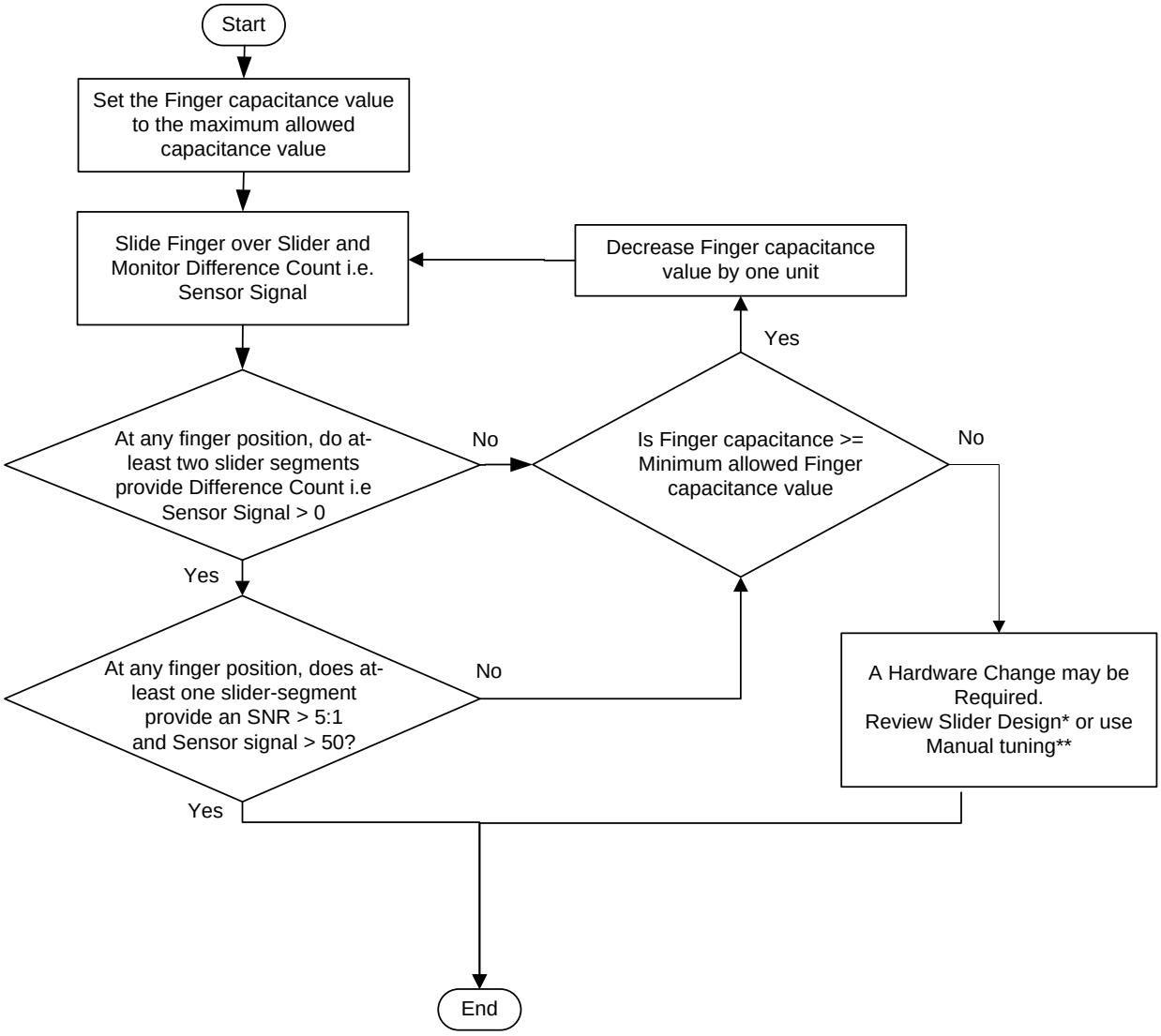
图 5-3. 在 PSoC Creator 中使用基于 SmartSense 自动调校的 CapSense 项目



- **对于滑条传感器：**最初将**手指电容**置为最高值。然后在滑条上滑动您的手指。如果在滑条的任意位置上至少一个滑条段状态为 ON 并且 SNR > 5:1，以及至少两个滑条段报告的“差值计数”（即“传感器信号”值）大于零，则使用该**手指电容。**否则，**手指电容**值将被降低，直到上述条件为真为止。

  如果将**手指电容**置为最小允许值后仍未满足上述条件，请进行手动调校或根据滑条设计注意事项中的内容修改硬件，或者参考调校调试常见问题解答。图 5-4 解释了为滑条进行手指电容值设置的过程。

图 5-4. 设置滑条的手指电容值



* 要查看滑块设计，请参阅设计注意事项一章中的滑条设计部分。.

**要进行手动调校，请参考 CapSense 功能调校章节的手动调校部分。

5.2.1.1.2 高级设置

通过使用 **Advanced** 选项卡，您可以指定 CapSense 时钟并设置阈值参数。在 SmartSense 自动调校模式中，该算法可自动调校大部分高级参数。因此，无需手动调校这些参数的值。

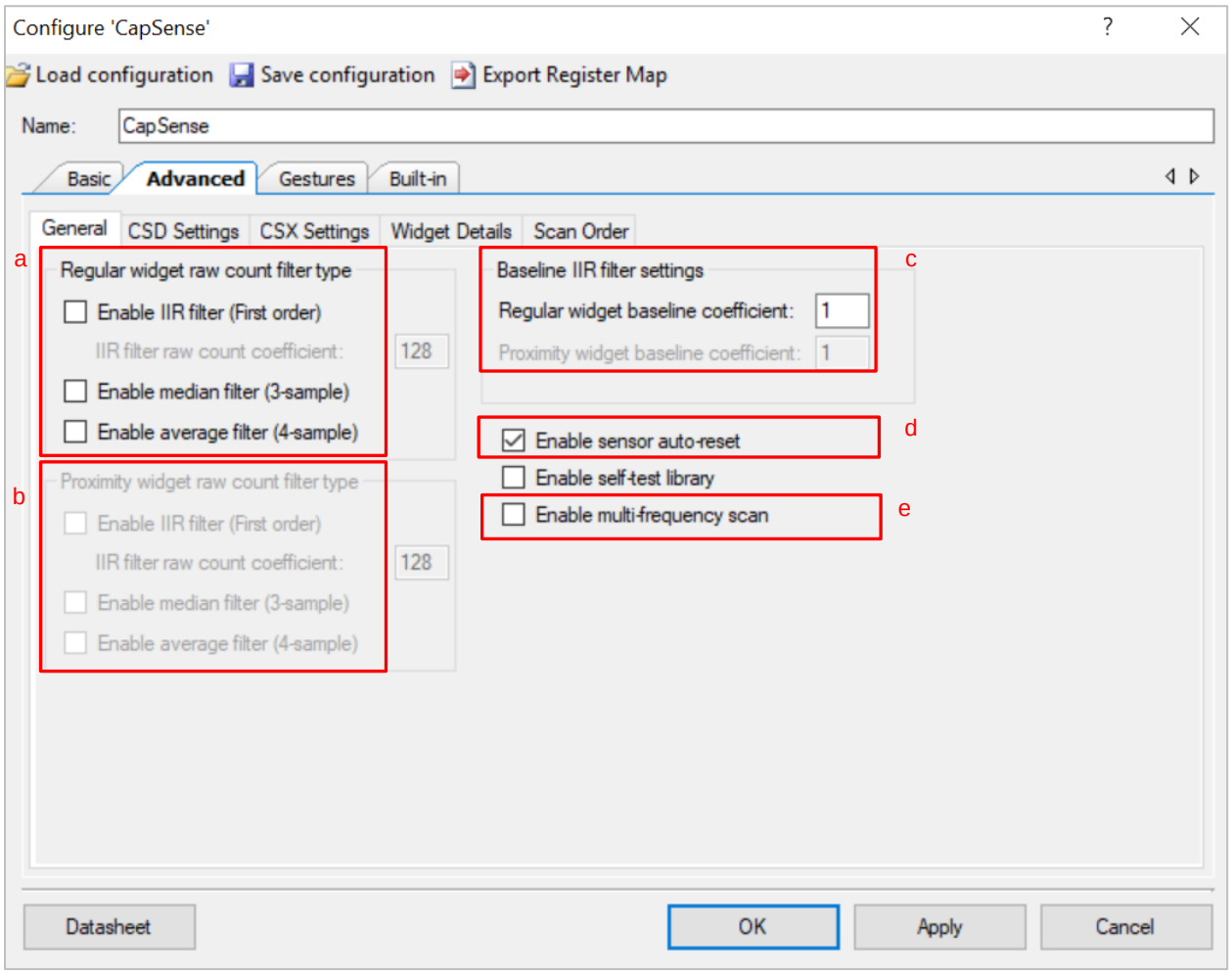
Advanced 选项卡包含以下的子选项卡：

5.2.1.1.2.1 General 选项卡

General 选项卡 包含所有 widget 的通用参数，不管这些 widget 所使用的感应方式如何。该选项卡允许您配置以下特性：

- 为所有 widget 使能并配置滤波器
- 配置基准线更新率
- 使能/禁用传感器自动复位功能

图 5-5.Advanced 设置中的 General 选项卡



a. **Regular widget raw count filter type**：通过该参数，您可以为所有 widget 选择固件滤波器（接近感应 widget 除外），以降低原始计数中的噪声。表 5-2 说明了各可用的滤波器及其应用。请勿在开始时使能滤波器；当您使用调谐器观察原始计数中的噪声时，请根据表 5-2 选择相应的滤波器。如果您选择使用 IIR 滤波器，首先请选择包含了一个更大过滤系数的滤波器，并递减过滤系数，直到获得 SNR ≥ 5:1 为止。通过 CapSense 3.0 组件，用户可以使能多个滤波器。

b. **Proximity widget raw count filter type**：该参数与 **Regular widget rawcount filter type** 相似，但它只能用于接近感应的 widget。与按键和滑条传感器相比，接近感应传感器的噪声更大，因此它需要不同的滤波器设置。与 **Regular widget rawcount filter type** 相同，在开始时请勿使能这些滤波器。当观察原始计数时，您可以使能滤波器，使 SNR ≥ 5:1。

c. **Baseline IIR filter settings**：该参数控制基准线更新速率。基准线是原始计数经过 IIR 过滤后的值。基准线系数值是 IIR 滤波器公式（如表 5-2 所示）中的参数‘N’。该参数控制基准线更新的速率。您可以使 N 的值从一个小值（N = 1）开始并递增该值，直到基准线值能跟踪原始计数中的缓慢变化为止。

d. **Enable sensor auto-reset**：使能该项会限制传感器保持开启状态的最大持续时间（通常为 5 到 10 秒）。如果由于发生了一个大的电源电压波动或是因为噪声条件突然发生了变化，使原始计数意外上升，则通过使能该项可以防止传感器的状态持续开启。使用该设置可以避免传感器在高噪声条件下被栓锁。

e. **Enable multi-frequency scan**：使能多频率扫描以不同的频率对传感器进行三遍扫描操作。传感器差值计数的中值被选择用于进一步处理。在特定传感器扫描频率下出现外部噪声时，可使用该功能获取稳健操作。

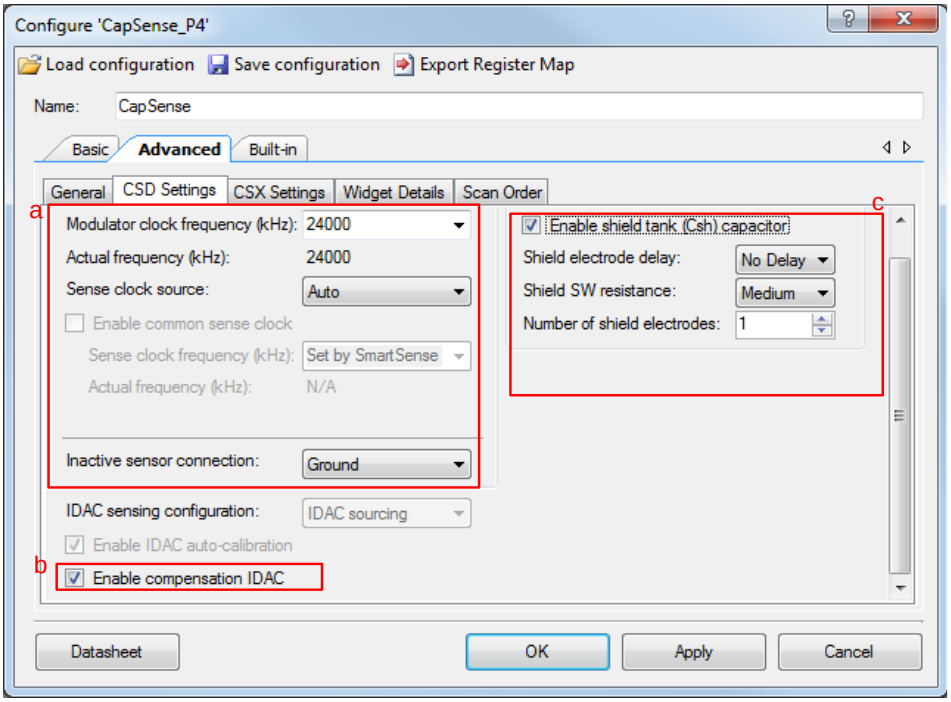
表 5-2. CapSense 组件中的原始数据噪声滤波器

滤波器	说明	数学描述	应用
中值滤波器	非线性滤波器采用了三个最新的样本，并计算出它们的中间值。	$y[i] = median\ (x[i], x[i-1], x[i-2])$	消除来自电机和开关电源的噪声
均值滤波器	具有同样的加权系数的有限脉冲响应滤波器（无反馈回路）。它采用了四个最新的样本，并计算出了它们的平均值。	$y[i] = \frac{1}{4} * (x[i] + x[i-1] + x[i-2] + x[i-3])$	消除周期性噪声（如电源导致的噪声）
一阶 IIR 滤波器	具有与 RC 低通滤波器相似的阶跃响应的无限脉冲响应滤波器（有反馈回路），所以可以通过低频信号（手指触摸响应）。 K 值被固定为 256。 N 是图 5-5 中可选的 **IIR 滤波器原始计数系数**。降低 N 值可降低噪声，但会使响应变慢。	$y[i] = \frac{1}{K} * \{N * x[i] + (K - N) * y[i-1]\}$	消除高频噪声。

5.2.1.1.2.2 CSD 设置

通过 **CSD Settings** 选项卡，您可以配置 CSD widget 的相关设置，如图 5-6 所示。

图 5-6. CapSense 组件中的 **CSD Settings**



a. **Scan settings**：该参数用于指定以下参数：

- **Modulator clock frequency (kHz)**：建议将该参数设置为最大值以缩短扫描时间，从而降低平均功耗。请参考调制器时钟的相关参数一节，了解该参数的建议值。
- **Sense clock source**：该参数用于指定感应时钟源。可用选项分别为‘Direct’、‘8-bit pseudo random sequence（即 8 位 PRS）、‘12-bit PRS、、‘Auto’和 SSCx（SSC — 扩频，仅适用于 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 系列器件）。‘Auto’选项可自动选择 PRS 顺序的长度（针对所有 PSoC 4 系列器件）或 SSCx 值（仅针对 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 系列器件）。如果您的设计对电磁兼容要求比较严格，那么建议您使用 PRS 或 SSCx，因为它能够降低电磁辐射。请参考感应时钟的相关参数一节，了解时钟源选择指南的详细信息。
- **Inactive sensor connection**：CapSense 一次扫描一个传感器。各个传感器未被扫描时，通过该项可确定传感器的连接情况。

  建议使用“Ground”选项，因为它可以降低对扫描的传感器带来的噪声。对于防水设计，请将该选项设置为“Shield”（屏蔽）。

  注意，建议勾选 **Enable shield electrode** 选项，以将“Inactive sensor connection”选项设置为“Shield”。

b. **Enable compensation IDAC**：该参数用于使能或禁用补偿 IDAC。通过使能补偿 IDAC，可以选择 CSD 操作的双向 IDAC 模式。支持双向 IDAC 的 SmartSense 算法有助于降低传感器的扫描时间，从而降低器件的平均功耗。

c. **Enable shield electrode**：该参数用于使能或禁用屏蔽电极。如果您的电路板中有一个用于接近感应、防水功能或降低传感器的 C_P 值的屏蔽电极，则应该使能屏蔽电极。更多有关信息，请参考驱动屏蔽信号和屏蔽电极一节的内容。

- Enable shield tank capacitor：如果您正在使用一个 C_{SH_TANK} 电容，请使能该项。更多相关信息，请参考 CapSense CSD 屏蔽一节的内容。
- **Shield electrode delay**：为了确保屏蔽电极的正常操作，屏蔽信号应和传感器信号的相位相互匹配。您可以使用示波器查看传感器和屏蔽信号，以验证该条件。如果它们没有对齐，那么使用该项可添加延迟到屏蔽信号，使信号对齐。可用的延迟值会因选定的器件不同而异。

- **Shield SW resistance**：通过该参数，可以控制屏蔽信号上升和下降的时间，从而降低 EMI。该参数仅适用于 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 系列器件。建议您从“Low” EMI 设置开始执行操作，因为它可动态控制屏蔽信号的转换速率。如果屏蔽切换产生电磁辐射，您可以将该参数设置为“low”、“medium （default）”或“high”值。
- **Number of shield electrode**：该参数指定了设计中所需屏蔽电极的数量。大部分设计都使用一个专用的屏蔽电极；但是，某些设计需要多个专用的屏蔽电极以便于实现 PCB 布局路由或尽量减少屏蔽层所需的 PCB 空间。

5.2.1.1.2.3 CSX 设置

通过 **CSX Settings** 选项卡，您可以配置 CSX widget 的相关设置。目前，由于 SmartSense 算法不支持 CSX widget，因此您必须手动调校该功能。更多有关手动调校过程的信息，请参考手动调校一节。

5.2.1.1.2.4 Widget 细节

通过 **Widget Details** 选项卡，您可以配置各个阈值（如图图 5-7 所示）并指定组合传感器的元素（如图图 5-8）。如果将图 5-2 中的 **CSD tuning mode** 设置为 **SmartSense (Full Auto-Tune)**，那么 CapSense 组件会自动调校各个阈值，因此在 **Widget Details** 选项卡中将显示 **Set by SmartSense**。

如果将 **CSD tuning method** 设置为 **SmartSense (Hardware parameters only)**，那么阈值的建议值会按照选择 CapSense 软件参数一节中的定义进行。

图 5-7. Widget Details 选项卡设置

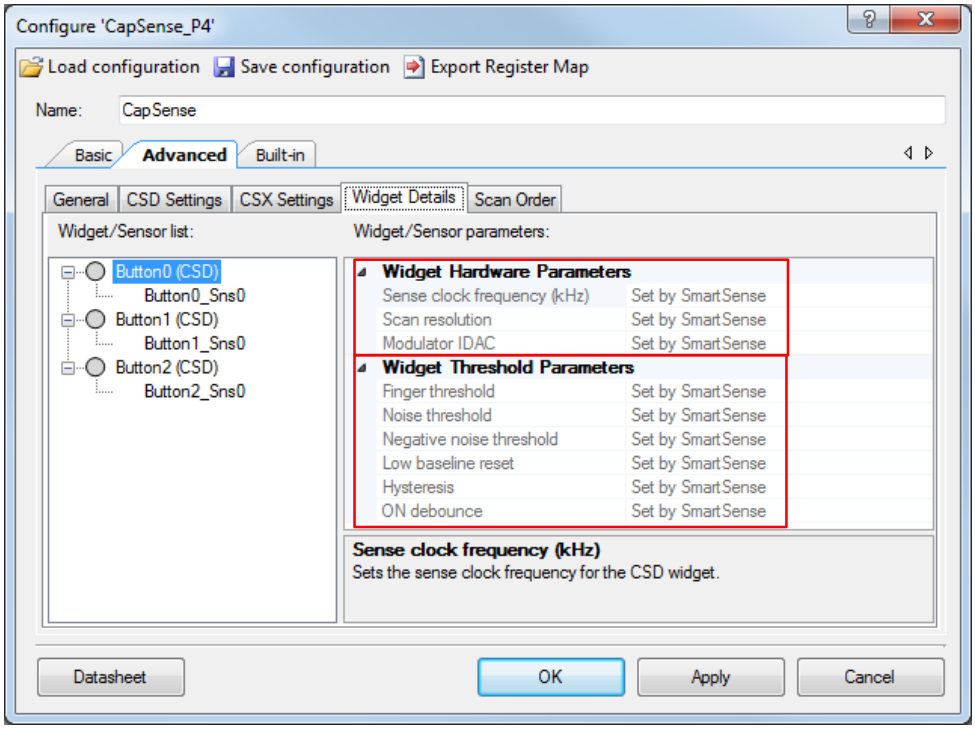
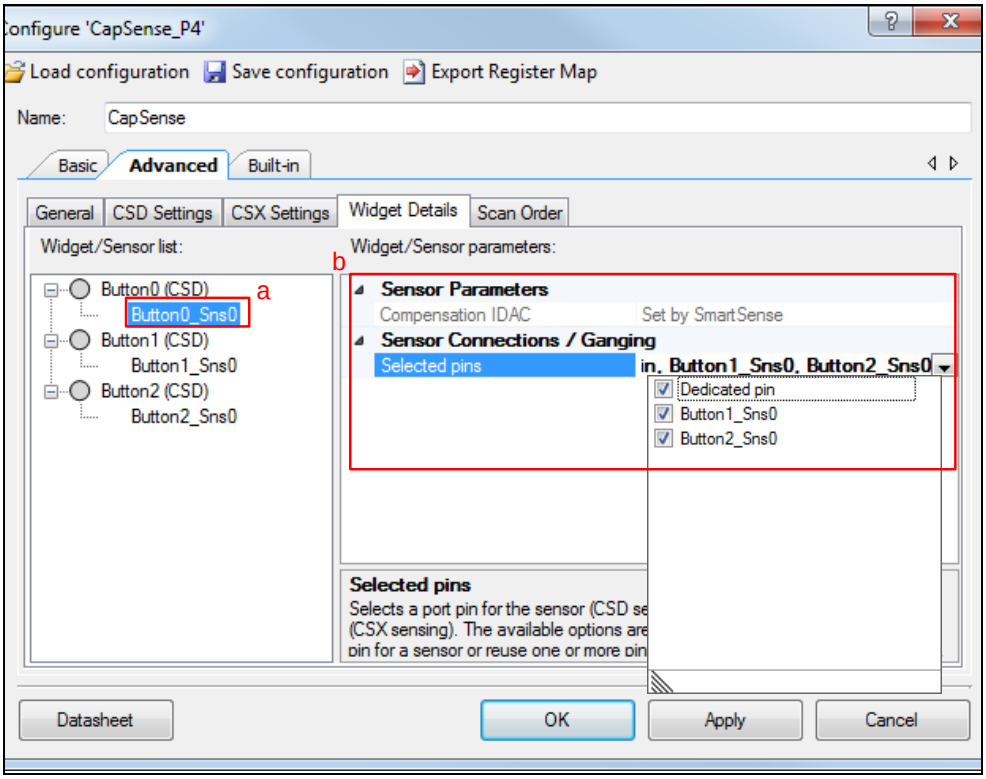


图 5-8. 指定 SmartSense 模式下的组合传感器元素



5.3 手动调校

5.3.1 概述

赛普拉斯的 SmartSense 技术允许器件进行自我校准，以获得最佳性能，并自动完成整个调校过程。这种技术可满足大多数设计的要求，但在 SmartSense 不能自动调校或拥有特定的 SNR 或电源要求的情况下，您可以修改各个 CapSense 参数以满足系统要求。通过手动调校可以实现该操作。

手动调校（与 SmartSense 自动调校相反）的部分优点包括：

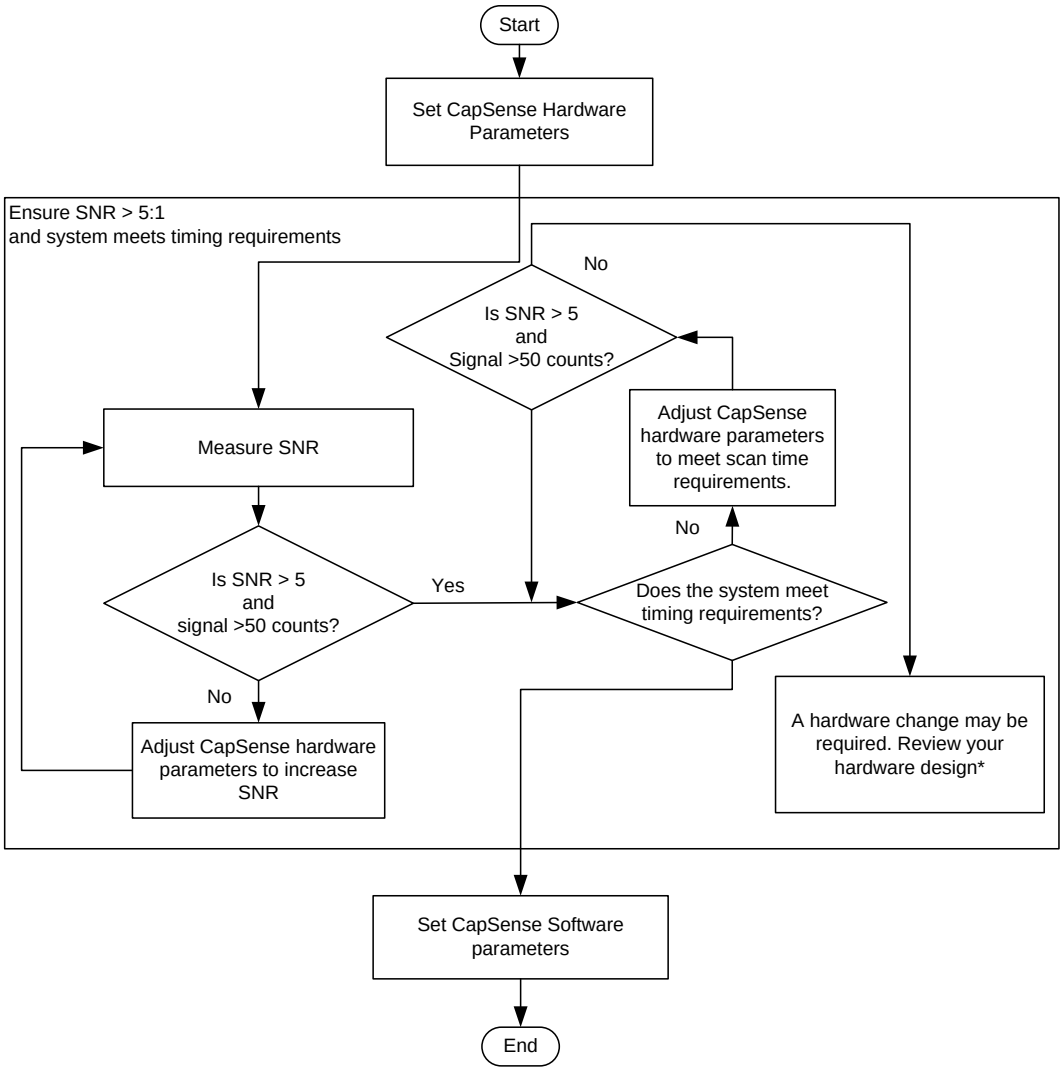
- 严格控制参数设置：SmartSense 会自动设置所有参数。但是在您需要严格控制各参数中，也有意外情况发生。例如，如果您需要严格控制 PSoC 4 扫描一组传感器所需的时间或者严格控制每个传感器的感应时钟频率（以降低各系统中的 EMI 值），请进行手动调校。
- 支持更高的寄生电容：对于 0.2 pF 的手指电容，SmartSense 支持高达 45 pF 的寄生电容；对于 0.1 pF 的手指电容，支持高达 35 pF 的寄生电容。如果寄生电容量大于 SmartSense 所支持的最大值，您需要进行手动调校。

手动调校过程可总结为以下三个步骤，如图 5-9 所示。

1. 设置 CapSense 组件硬件参数的初始值。
2. 调校 CapSense 组件硬件参数，以确保信噪比大于 5:1（信号不小于 50 各计数），并且系统满足各时序要求。
3. 设置 CapSense 组件硬件参数的最佳值（阈值参数）

下面各节详细介绍了手动调校的基本原理以及上述三个步骤的操作。有关 PSoC 4 CapSense 架构的知识是这些章节的首要介绍的内容。请参考电容式触摸感应方法和 CapSense CSD 感应，以熟悉 PSoC 4 CapSense 构架。根据选定的感应方法，手动调校过程存在差异。请参考 CSD 感应方法一节，了解如何使用 CSD 进行手动调校（对于自电容感应）。另外参考 CSX 感应方法一节，了解如何使用 CSX 进行手动调校（对于互电容感应）。如果您不打算在您的设计中使用手动调校方法，可跳过这些章节。

图 5-9. 手动调校流程图



*要查看硬件设计，请参考设计注意事项章节中的传感器结构和 PCB 布局指南节的内容。

5.3.2 CSD 感应方法

本节说明了使用 CSD 感应方法进行手动调校的基础知识。另外，它还介绍了影响手动调校过程的各个硬件和软件参数。最终，它为您提供了手动调校的三个示例，包括按键、滑条和接近感应的 widget。

5.3.2.1 基础知识

5.3.2.1.1 转换增益和 CapSense 信号

转换增益会影响系统可观查到的信号状态。增益越大，信号越高，从而可获得的信噪比也越大。请注意，增大增益可能同时增强信号和噪声。但若有需要，您可以使用固件滤波器来降低噪声。有关可用固件滤波器的详细信息，请参考表 5-2。

5.3.2.1.1.1 单 IDAC 模式下的转换增益

在单 IDAC 模式下，原始计数值与传感器电容值成正比。

公式 5-1. 原始计数与传感器电容的关系

$$\text{raw count} = G_C C_S$$

其中，C_S 为传感器电容。如果传感器上不存在手指触摸，$C_S = C_P$。传感器上存在手指触摸时，$C_S = (C_P + C_F)$。G_C 是 CapSense CSD 电容值转换成数字值的增益。根据公式 3-6 和公式 5-1，使用所推荐的 IDAC 源电流模式的转换增益的近似值如下：

公式 5-2. 数字转换器增益的电容

$$Gc = (2^N - 1)\frac{V_{REF}\, F_{SW}}{I_{MOD}}$$

其中：

V_{REF} 是比较器的参考电压。所有 PSoC 4 系列器件（PSoC 4 S 系列除外）的 V_{REF} 值被固定为 1.2 V。对于 PSoC 4100S Plus、PSoC 4100PS 系列以及 PSoC 6[1] 系列，V_{REF} 值的范围为 0.6 V 到 V_{DDA} – 0.6 V。

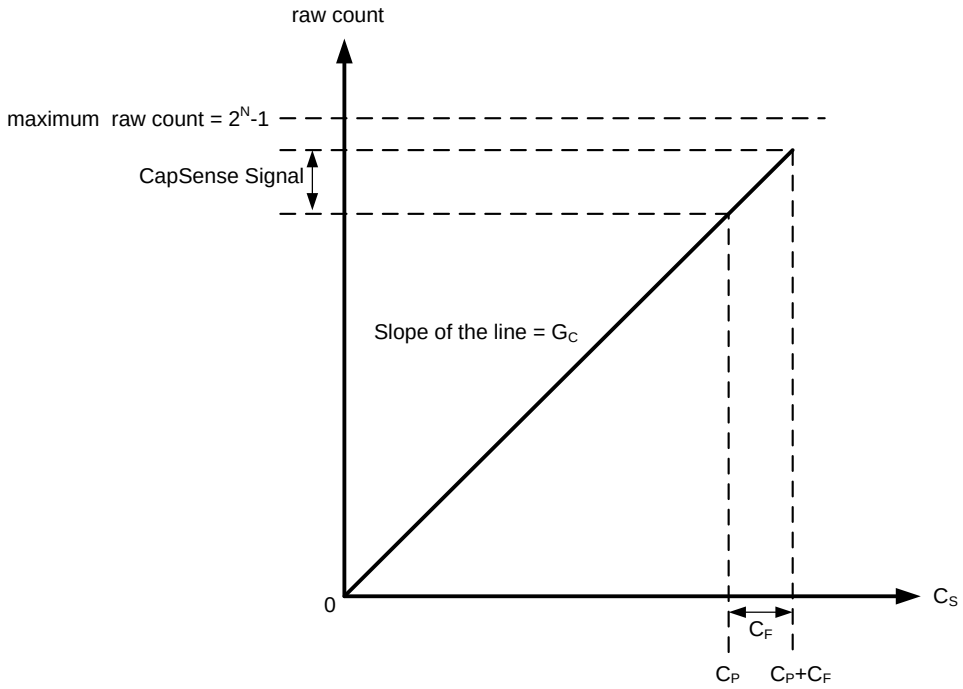
F_{SW} 是感应时钟频率

I_{MOD} 是调制 IDAC 电流

转换增益的各可调校参数分别为 V_{REF}（仅针对 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6MCU 系列）、F_{SW}、I_{MOD} 和 N。图 5-10 显示的是原始计数和传感器电容的关系图。

注意：CapSense 组件会根据 V_{DDA} 电压自动选择 V_{REF}。

图 5-10. 原始计数与传感器电容



当传感器上有手指触摸时，原始计数将发生变化，这种变化被称为 CapSense 信号。图 5-11 显示了信号变化值和转换增益的关系。

图 5-11. 不同转换增益中的信号值

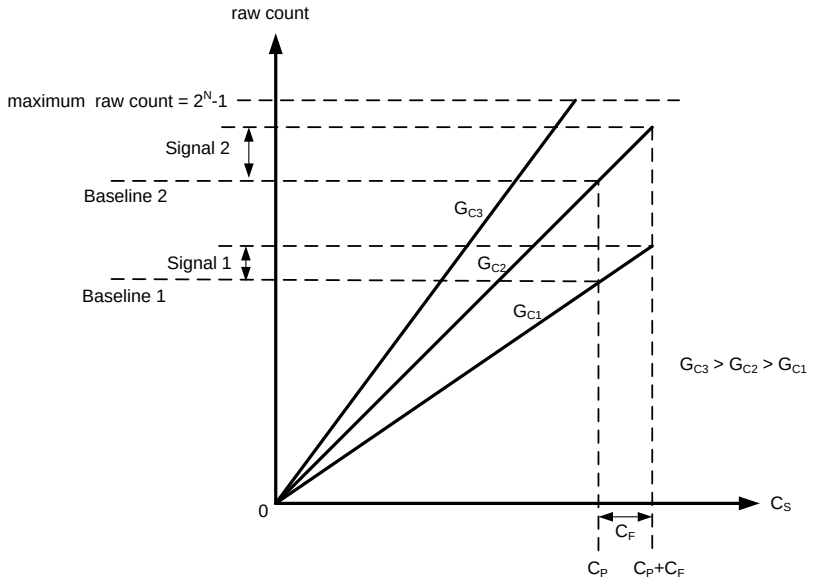
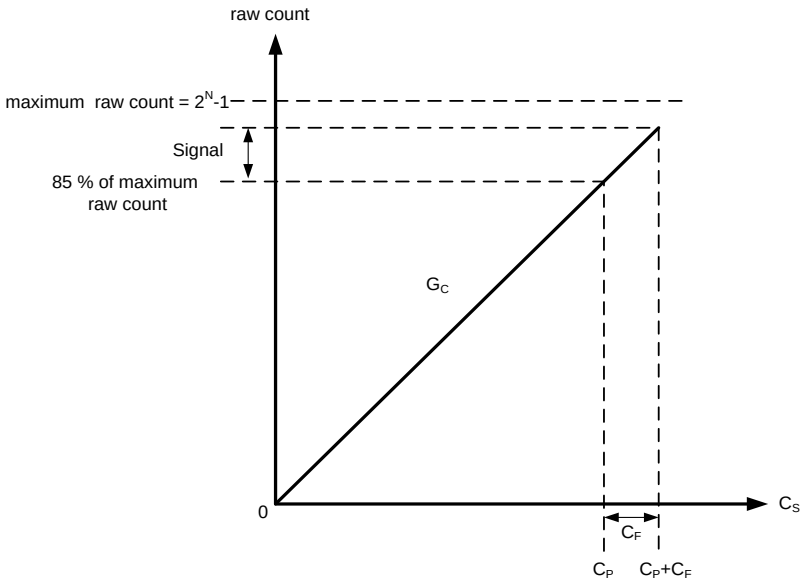


图 5-11 显示的是与三个转换增益值相对应的三条关系线：G_{C3}、G_{C2} 和 G_{C1}。转换增益增大会使信号值增大。但增大转换增益值也会使与 C_P（即基线）相对应的原始计数值越来越靠近原始计数的最大值（2^N-1）。增益值过大，原始计数会饱和，如 G_{C3} 所示。因此，为避免原始计数饱和，您应调整转换增益的值，以得到良好的信号值。对 CSD 参数进行调校。当传感器上没有发生手指触摸时（即 $C_S = C_P$）原始计数 = (2^N-1)的 85%，如图 5-12 所示。这样能够确保在增益值最大并且环境发生变化时，可以尽可能增大原始计数值，并且手指触摸位置上不会发生饱和。

图 5-12. 建议的调校方式



5.3.2.1.1.2 双 IDAC 模式下的转换增益

从公式 5-2 和公式 3-8，可推出双 IDAC 模式下的原始计数计算公式：

公式 5-3. 双 IDAC 模式的原始计数

$$\text{raw count} = G_C C_S - (2^N - 1)\frac{I_{COMP}}{I_{MOD}}$$

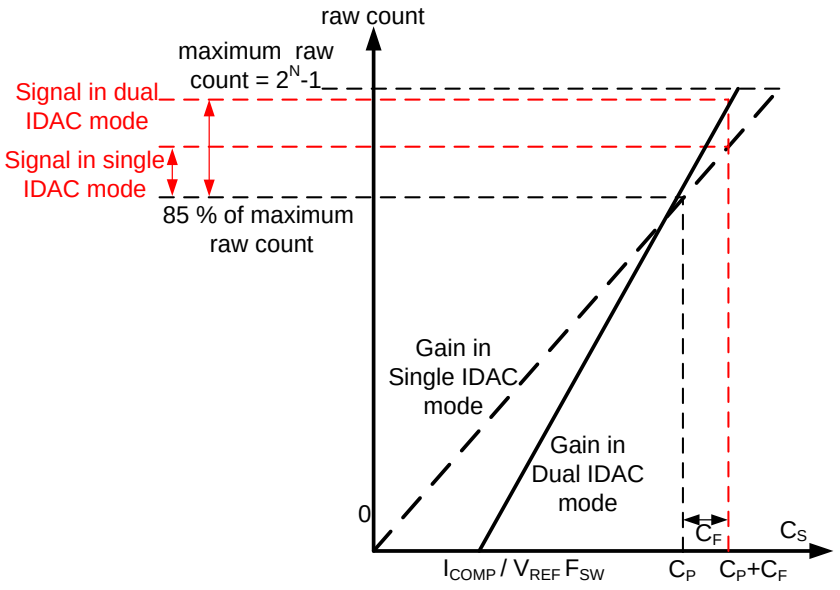
其中：

I_{COMP} 是补充 IDAC 电流

G_C 是通过公式 5-2 计算出的

在单 IDAC 模式和双 IDAC 模式对 CSD 参数进行调校时，如果传感器上没有发生手指触摸（即 $C_S = C_P$）时，原始计数 = (2^N-1)的 85%如图 5-13 所示，从而确保高转换增值，避免出现扁平点以及避免因环境变化造成的原始计数饱和情况。

图 5-13. 双 IDAC 模式下的建议调校方式



如图 5-13 所示， 在单 IDAC 模式下 85%的要求被限制为固定增值，而在双 IDAC 模式下，通过将 C_S 轴拦截向右移动（增加 I_{COMP}）以及相应降低调制器 IDAC（I_{MOD}）可以增大增值。$C_S = C_P$ 时仍获取的原始计数为(2^N-1)的 85%。

这样使用双 IDAC 模式会使原始计数与 C_P 值的关系图发生以下变化：

a. 使用补偿 IDAC 会引起 C_S 轴上的一个非零拦截，可使用下面公式计算该值：

公式 5-4. 关于 I_{COMP} 的 Cs Axis Intercept

$$Cs\ \text{axis intercept} = \left(\frac{I_{COMP}}{V_{REF}\ F_{SW}}\right)$$

b. 由于双 IDAC 模式下的 I_{MOD} 值为单 IDAC 模式下 I_{MOD} 值的一半（所有其他的参数都相同），因此，从公式 5-2 中可看出：双 IDAC 模式下的增益 G_C 是单 IDAC 模式下增益值的两倍。因此，在给定的分辨率 N 上，双 IDAC 模式的信号比单 IDAC 模式的信号大一倍。

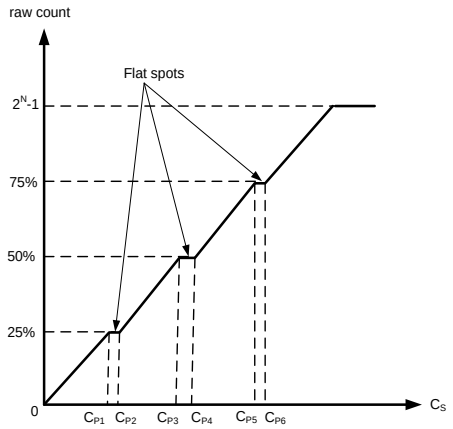
在进行手动调校传感器过程中，请注意使用公式 5-2、公式 5-3 和以下注意事项：

1. 增益越高，灵敏度则越大，并且整个系统的性能也越好。但设置增益时要注意避免原始计数饱和的情况，如图 5-11 中增益 G_{C3} 的关系线所示。建议您设置增益以让对应于 C_P 的原始计数为最大原始计数值的 85%（对于单 IDAC 模式和双 IDAC 模式都是这样）。
   设置感应时钟频率（F_{SW}）时请务必小心。频率越高，增益也越大，但该频率要足够低，以完成传感器充电和放电过程，如图 3-6 传感器电容的电压所示。
2. 勾选补偿 IDAC 时可增大增益值。如果按照上述部分设置补偿 IDAC，可使增益值增加一倍。因此，如果没将补偿 IDAC 用于通用应用，建议您始终使能该参数。
3. 调制器 IDAC 电流越低，获得的增益越大。调整您的 IDAC 以得到最大增益，但需要保证相应于 C_P 的原始计数在环境条件发生变化（如温度改变）时可以尽量调整，如图 5-12 和图 5-13 所示。
4. 通过增大分辨率，可以增大增益值。如果将分辨率递增一位数，系统的增益会增加一倍，但扫描时间也随着增加一倍，如公式 3-4 所示。因此，建议适当调整分辨率，从而使扫描时间和增益值得到平衡。

5.3.2.1.2 扁平点

理想情况下，原始计数和传感器电容间为线性关系，如图 5-10 和图 5-13 所示。但实际上，sigma delta 调制器有多个非灵敏区域，它也称为扁平点或死区 — 在某个传感器电容范围内，该 sigma delta 调制器可能提供同样的原始计数值，如图图 5-14 所示。

图 5-14. 原始计数与传感器电容的关系图中的扁平点（使用直接时钟时）



使用 CapSense CSD 时，这些扁平点发生于靠近最大原始计数值的 25%、50%和 75%处（即 2^N-1 的 25%、50%和 75%处，其中 N 表示扫描分辨率）。当直接时钟作为感应时钟源时，这些扁平点变得最明显。而在 PRS 被用作感应时钟时，将不存在扁平点（另请参考感应时钟的相关参数一节的内容）。

对于大部分系统，建议您将 PRS 作为感应时钟源使用，因为它能够限制扁平点的影响并降低 EMI/EMC 噪声，如感应时钟源一节所述。如果您的系统需要一个直接时钟，那么必须使用自动校准功能或者在使用手动校准功能时避免该原始计数范围。

5.3.2.2 选择 CapSense 硬件参数

CapSense 硬件参数控制着转换增益和 CapSense 信号。表 5-3 列出了适用于 CSD 感应方法的各个 CapSense 硬件参数。通过下面部分介绍的指导内容，您可以调整这些参数以得到特定系统的最佳性能。

表 5-3. CapSense 组件硬件参数

Sl. 序号	CapSense 组件参数名称
1	感应时钟频率
2	感应时钟源
3	调制器时钟频率
4	调制器 IDAC
5	补偿 IDAC
6	扫描分辨率

5.3.2.2.1 感应时钟的相关参数

对于一个给定的分辨率 N，您可以更改 F_{SW} 和各 IDAC 值，使对应于 C_P 的原始计数等于其最大值的 85%。但是，如果所选的感应时钟频率（F_{SW}）不准确，它会影响 CapSense CSD 的操作。根据公式 5-2，建议您使用最大时钟频率，但要确保传感器电容已完成充电或放电过程，如图 3-5 所示。

选择最佳时钟频率的最简单的方法是使能器件的 SmartSense 功能，然后回读各个数值，如按键 Widget 示例一节所述。SmartSense 将自动读取传感器的寄生电容，并设置感应时钟的最大可用频率以确保正确的充电和放电操作。另外，您也可以按照以下信息手动确定感应时钟的频率：

传感器电容的充电和放电路径包含各串联电阻（即图 3-3、图 3-5 中的 R_{Series} 和 GPIO 单元开关），它们会使充电和放电过程变慢。

如果 $R_{SeriesTotal}$ 是 GPIO 电阻与外部串联电阻之和，而 C_P 是传感器的寄生电容，那么您应该选择足够低的开关频率，从而允许传感器电容完成充电和放电过程。根据经验，可以选择 $5R_{SeriesTotal}C_P$ 的周期用于充电和放电循环。感应时钟的最小时间周期和最大频率的计算公式分别为：

公式 5-5. 感应时钟的最小时间周期

$$T_{SW}(\text{minimum}) = 10R_{SeriesTotal}C_P$$

公式 5-6. 感应时钟最大频率

$$F_{SW}(\text{maximum}) = \frac{1}{10R_{SeriesTotal}C_P}$$

GPIO 电阻的典型值为 500 Ω，推荐的外部电阻大小为 560 Ω（请参考 CapSense 引脚上的串联电阻一节，了解详细信息）。因此，当计算最大开关频率时，使 R_S 的值等于 1.06 kΩ。提供 F_{SW} 的源时钟被称为**感应时钟源**。

默认情况下，系统将直接时钟作为可提供 F_{SW} 的感应时钟源。通常，由于存在扁平点，而且您的系统容易受到 EMI/EMC 噪声的影响，因此建议您使能 PRS 时钟选项。除了直接时钟和 PRS 时钟源外，PSoC 4 S 系列和 PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 系列器件还支持扩频时钟源。在这种情况下，可以在某个预定范围内增大感应时钟的频率，从而降低 EMI 噪声。通过选择时钟分频器为 SSC2 至 SSC5 可以指定频率扩展范围，建议为感应时钟所选中心频率的±10%。更多有关扩频架构的信息，请参考 PSoC 4000S 技术参考手册的内容。

注意：选择 SSC 分频器时需要满足以下条件：2^{SSCn} 不大于调制器时钟和感应时钟的频率比例的 10%。

可选的 PRS 时钟源包括：PRS-8 和 PRS-12。一般情况下，PRS-12 比 PRS-8 好，但只有扫描时间（整个 PRS 序列等于一个扫描周期）较长时，才选择 PRS-12。为了确定需要使用的时钟源，您需要计算扫描周期和 PRS 序列周期。

公式 5-7. PRS 扫描周期

$$T_{SCAN} = \frac{2^{\text{Resolution}} - 1}{\text{Modulator Clock Frequency}}$$

公式 5-8. PRS 序列周期

$$T_{PRS} = \frac{2^{\text{PRS_Resolution}} - 1}{\text{Sense Clock Frequency}}$$

N 为 8 或 12

使能 PRS 后，请确保 $T_{SCAN} \geq T_{PRS}$。通过选择 PRS-Auto 选项，该过程可自动完成。在 CapSense 配置窗口中选择 PRS 的感应时钟频率后，最大频率和平均频率将显示在配置窗口中。 平均频率为最大感应时钟频率的一半。应针对最大感应时钟频率对传感器进行完全充电和放电，并需要满足公式 5-6。

要想查找串联电阻为 500 Ω 的情况下 C_P 和感应时钟频率的参考值，请参考 CapSense 组件数据手册。

5.3.2.2.2 调制器时钟的相关参数

调制器时钟会影响扫描时间（公式 3-4）和信噪比。通常，建议选择调制器时钟的最高频率。这样能提供最佳扫描时间，并尽可能降低测量误差。

5.3.2.2.3 调制 IDAC 和补偿 IDAC

CSD 支持用于给 C_{MOD} 充电的两个 IDAC、一个调制器 IDAC 和一个补偿 IDAC，如图 3-1 所示。它们控制着转换增益和 CapSense 信号，以便进行电容-数字转换。

使用 CapSense 组件时，可以对 IDAC 进行以下配置：

- 启用或禁用补偿 IDAC
- 启用或禁用 IDAC 的自动校准功能
- 禁用自动校准并使用手动调试时调制器和补偿 IDAC 的 DAC 代码选择。

5.3.2.2.3.1 补偿 IDAC

在大多数情况下，建议启用补偿 IDAC 以增加转换增益和 CapSense 信号。当需要测量绝对电容或想将 IDAC 用于其它模拟功能时，应禁用补偿 IDAC。

5.3.2.2.3.2 自动校准

通过该特性，固件可以自动对 IDAC 进行校准，以获得所需的校准目标，即 85%。在大多数情况下启用自动校准功能。要想使用别的校准等级，例如，降低校准百分比，以避免因环境发生变化而引起的原始计数饱和，则可以通过 API `CapSense_CSDCalibrateWidget()`对 IDAC 自动校准百分比进行更改。

当需要通过在复位期间测量原始计数值来检测 Cp 的变化的时候，请禁用 IDAC 自动校准功能，例如：

- 检测电路板上或由布局问题引起的传感器 Cp 较大变化
- 在复位时检测到手指触摸
- 采用高级 CapSense 方法，例如液位感应，以便在复位时为不同液位产生不同的原始计数值

5.3.2.2.3.3 选择 DAC 代码

在大多数情况下，建议启用自动校准功能。不过，如果您因某种原因需要禁用该功能，在配置 CapSense 组件时请先启用自动校准，然后通过 CapSense 调谐器读回已校准的 IDAC 值。接着，请重新配置 CapSense 组件，以禁用自动校准，并使用从调谐器读回的固定 DAC 代码。

5.3.2.2.4 扫描分辨率

需要选择扫描分辨率，以维持扫描时间和信号间的平衡。由于分频率越高，信号电平则越大（公式 5-2），因此扫描时间也越长（根据公式 3-4 计算）。针对每位，信号强度和扫描时间会倍增。对于大多数传感器设计，建议从最低分辨率开始进行，并逐渐增大该值，以增强信号。如果时序预算还有调整空间，则可提高分辨率以获得比所建议的最小比率（5:1）更高的 SNR，从而提高系统的稳定性。

5.3.2.3 选择 CapSense 软件参数

CapSense 软件参数根据传感器的原始计数值控制该传感器的状态。表 5-4 提供的是 CapSense 软件参数的列表。这些参数适用于 CSD 和 CSX 感应方法。通过使用基准线部分的内容，该部分会定义这些参数并提供如何调整参数，以获得最佳性能设计的指导。

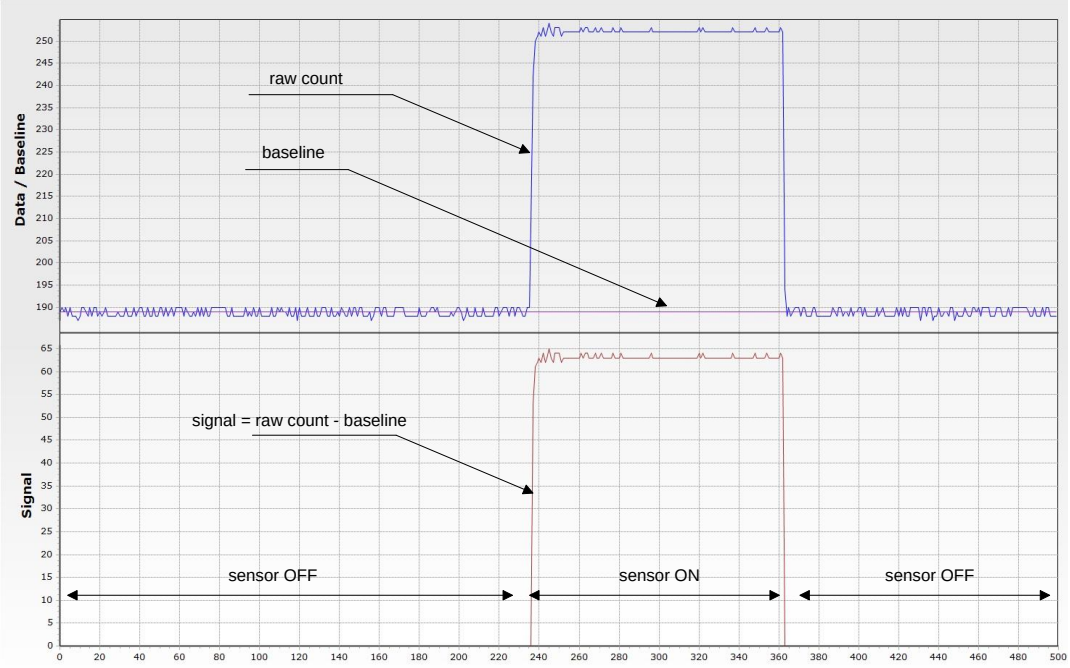
表 5-4. CapSense 组件软件参数

Sl. 序号	CapSense 组件参数名称
1.	手指阈值
2.	噪声阈值
3.	迟滞
4.	ON 去抖动
5.	传感器自动复位
6.	低基准线复位
7.	负噪声阈值

5.3.2.3.1 基准线

为给定 C_P 调校 CapSense 组件后，传感器的原始计数值随着温度和湿度等环境因素的变化而逐渐变化。因此，通过低通滤波原始计数值，CapSense 组件创建了一个新的计数值，即基准线。**基准线**会跟踪并补偿原始计数的变化。基准线可降低由于触摸引起原始计数的突变。因此，基准线值为计算信号提供了一个参考值。图 5-15 显示的是原始计数、基准线以及信号的概念。

图 5-15. 原始计数、基准线和信号



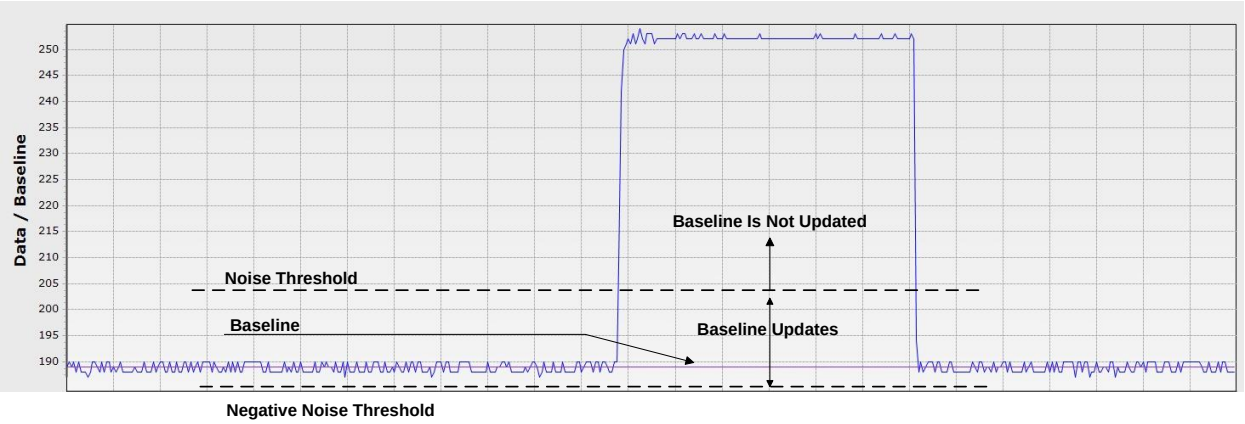
每次扫描 CapSense 传感器后，需要调用 `CapSense_UpdateEnabledBaselines()` API，以更新基准线。

5.3.2.3.2 基准线更新算法

要想正确调校 CapSense 软件（即阈值参数），必须了解使用 `CapSense_UpdateEnabledBaselines()` API 计算基准线的方法以及阈值参数对基准线更新的影响。

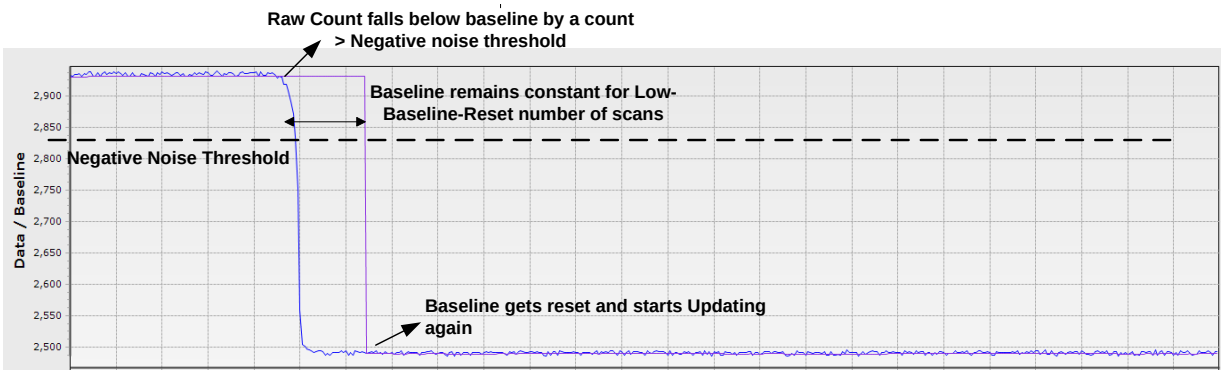
基本上，基准线就是经过低通滤波后的原始计数。如图 5-16 所示，如果当前原始计数值位于（基准线 – 负噪声阈值）到（基准线 + 噪声阈值）的范围内，则可以使用低通滤波原始计数值更新基准线。如果当前原始计数值大于基准线值，并且该差值大于噪声阈值，基准线值将保持不变，并要等于先前基准线值。

图 5-16. 基准线更新算法



如果当前原始计数值小于低于基准线值减去负噪声阈值的差值，基准线保持为在先前低基准线复位时的值（复位次数等于传感器扫描次数）。如果原始计数值在低基准线复位（复位次数等于传感器扫描次数）时一直低于基准线值减去噪声阈值的差值，那么基准线将被复位为当前原始计数值并重新更新，如图 5-17 所示。

图 5-17. 低基准线复位



5.3.2.3.3 手指阈值

手指阈值参数与迟滞参数一起使用，用于确定传感器的状态，如公式 5-9 所示。

公式 5-9. 传感器状态

$$\text{Sensor State} = \begin{cases} \text{ON} & \text{if (Signal} \geq \text{Finger Threshold} + \text{Hysteresis)} \\ \text{OFF} & \text{if (Signal} \leq \text{Finger Threshold} - \text{Hysteresis)} \end{cases}$$

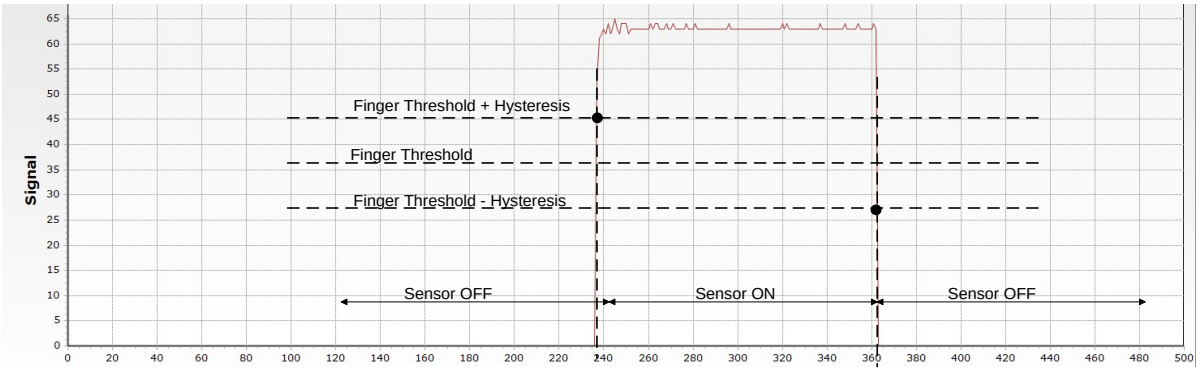
请注意，以上公式中的信号值表示在传感器有触摸时的原始计数值减去基准线值的差值，如图 5-15 所示。

建议您将手指阈值设置为该信号的 80%。该设置所提供的余量足以在各种 PCB 上的信号可能会发生变化的情况下能够可靠地检测传感器的 ON/OFF 状态。

5.3.2.3.4 迟滞

迟滞参数与手指阈值参数一起使用，用于确定传感器的状态，如公式 5-9 和图 5-18 所示。迟滞提供对传感器状态噪杂切换的抗干扰能力。迟滞参数的设置必须低于手指阈值参数设置。

图 5-18. 迟滞



建议您将迟滞设置为该信号的 10%。

5.3.2.3.5 噪声阈值

对于单传感器 Widgets（如按键和接近传感器），噪声阈值参数设置了原始计数上限，超过该值时便不更新基准线，如图 5-16 所示。换而言之，只要原始计数值高于*基准线* + *噪声阈值*，则基准线保持不变。这样可防止发生手指触摸时基准线随着原始计数的变化。

噪声阈值始终低于*手指阈值* – *迟滞*。建议您将噪声阈值设置为该信号的 40%。

当噪声阈值被设为低值时，如果由于电源稍微变化或 GPIO 高灌电流导致的地电压变化或其他原因，原始计数突然稍微增加，那么基准线将会保持不变。

换句话说，如果噪声阈值被设为接近*手指阈值* – *迟滞*的值，那么即使传感器上发生触摸，基准线仍会连续更新。这样做，信号值将得到降低（请注意，*信号值* = *原始计数值* – *基准线值*），并且误报了传感器状态。

5.3.2.3.6 负噪声阈值

负噪声阈值参数设置了原始计数下限，如果低于该值，基准线不更新由低基准线复位参数指定的采样次数，如图 2-26 所示。

负噪声阈值确保基准线不会因原始计数的任何重复的高振幅负噪声尖峰而下降，这些负噪声尖峰是由各种噪声源（如 ESD 事件）造成的。

建议将负噪声阈值参数值设置为等于噪声阈值参数值。

5.3.2.3.7 低基准线复位

该参数与负噪声阈值参数配合使用。该参数计算复位基准线所需的异常低原始计数的次数，如图 2-26 所示。

器件启动期间，如果传感器上发生手指触摸，那么启动时，基准线将被初始化为高原始计数值。抬起手指时，原始计数会下降到较低的值。在这种情况下，基准线应该跟踪低原始计数值。通过使用低基准线复位参数，可以处理该事件。当低采样次数达到低基准线复位次数时，它将基准线复位为低原始计数值。请注意，在这种情况下，手指抬起时，传感器不会响应，直到公式 5-10 所示的低基准线复位时间结束为止。

公式 5-10. 低基准线复位时间

$$\text{Low Baseline Reset Time} = \frac{\text{Low Baseline Reset parameter value}}{\text{Scan Rate}}$$

低基准线复位参数需要满足低基准线复位部分中所列出的以下各个条件：

- 低基准线复位时间大于负噪声（由噪声源，如 ESD 事件导致）所存在的时间。
- 低基准线复位时间需要小于在器件启动期间触摸传感器的手指抬起后传感器再次响应所需的时间。

低基准线复位参数通常被设为 30。

5.3.2.3.8 Debounce（去抖动）

通过该参数可以选择 CapSense 连续扫描的次数，在该过程中传感器必须有效，以使组件生成一个 ON 状态。去抖动可确保高频率高振幅的噪声不会导致误检。

公式 5-11. 传感器状态与去抖动间的关系

$$\text{Sensor State} = \begin{cases} \text{ON} & \text{if (Signal} \geq \text{Finger Threshold} + \text{Hysteresis) for scans} \geq \text{debounce} \\ \text{OFF} & \text{if (Signal} \leq \text{Finger Threshold} - \text{Hysteresis)} \\ \text{OFF} & \text{if (Signal} \geq \text{Finger Threshold} + \text{Hysteresis) for scans} < \textit{debounce} \end{cases}$$

去抖动参数会影响 CapSense 系统的响应时间。由于发生手指触摸，原始计数值增加到高于手指阈值加上迟滞，此时，可通过以下公式计算出传感器报告“ON”状态所需的时间：

公式 5-12. 去抖动与传感器响应时间的关系

$$\text{Sensor response time} = \frac{\text{Debounce}}{\text{Scan Rate}}$$

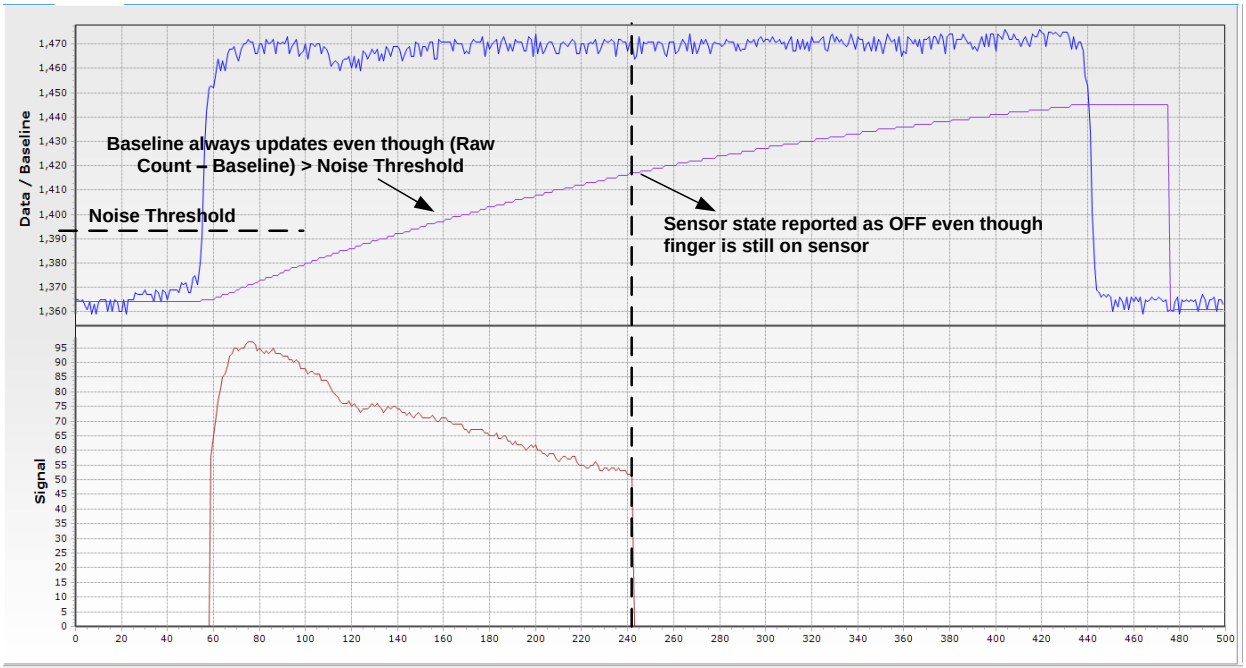
去抖动参数通常被设为‘3’以进行可靠的传感器状态检测。它可以随着终端用户系统的噪声发生变化。

5.3.2.3.9 传感器自动复位

使能传感器自动复位参数使基准线一直更新，无论信号高于或低于噪声阈值。

如果该参数被禁用，仅在当前原始计数处于（*基准线 – 负噪声阈值*）到（*基准线 + 噪声阈值*）的范围内，基准线才被更新，如图5-16所示和基准线更新算法部分的说明。然而，当该参数被使能时，如果当前原始计数高于（*基准线 – 负噪声阈值*）时，则基准线便一直更新，如图5-19所示。

图 5-19. 传感器自动复位被使能时的基准线更新状况



由于传感器自动复位参数被使能时，基准线一直被更新，因此该设置会限制上报传感器ON状态的最长时间。然而，如果传感器上没有发生任何触摸而原始计数突然上升，使能该参数可以阻止传感器始终被打开。较大的电源电压波动、高能射频噪声源或极快的温度变化都会导致这种突然升高。

如果遇到上面所述的问题，应该使能该参数。

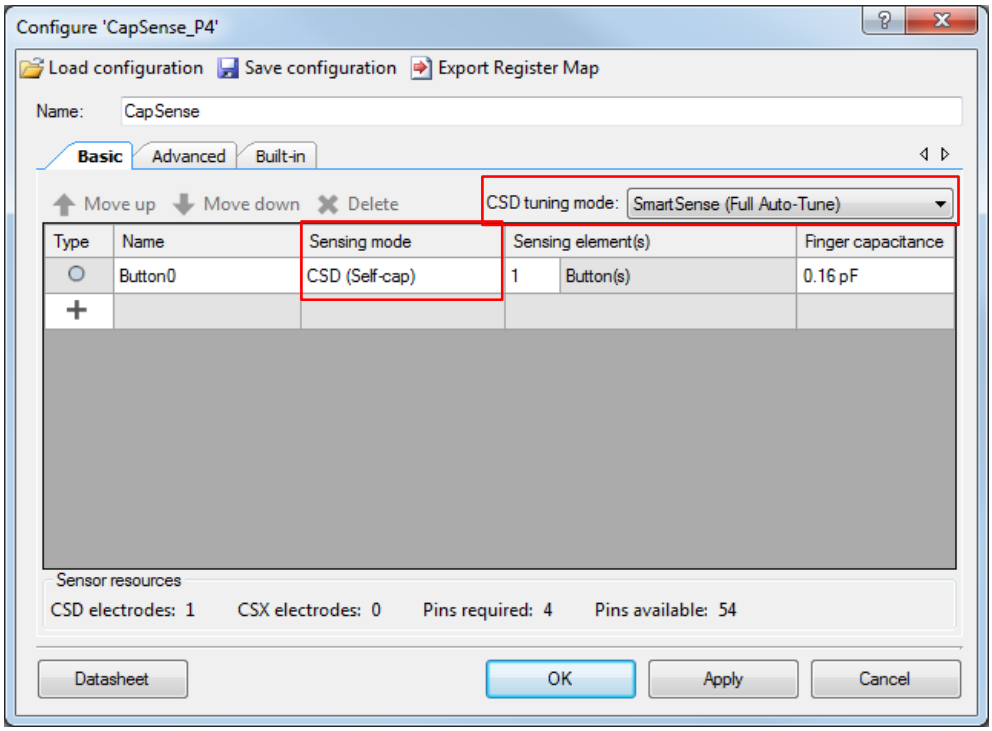
5.3.2.4 按键 Widget 示例

以下示例介绍的是使用调谐器助手调校 PSoC Creator CapSense 组件中基于自电容的按键 Widget。有关组件和所有相关参数的详细信息，请参考组件数据手册。

按照以下步骤中所列出的详细流程，可以手动设置所有调校参数。

1. 双击该组件或右键单击 **Component** 并选择 **Configure** 打开 CapSense 组件配置窗口。
2. 在 **Basic** 选项卡中，点击**+**符号以添加 Widget 按键。将 **Sensing mode** 选为 **CSD (Self-cap)**以及将 **CSD tuning mode 选为 Manual tuning**，如图 5-20 所示。

图 5-20. 选择“Manual Tuning”模式



3. 在 **Advanced** 选项卡的 **General** 设置窗口中，将所有滤波器参数设置为默认值。根据 SNR 和响应时间的要求，使能滤波器参数，如第 11 步所述。
4. 在 **Advanced** 选项卡的 **CSD Settings** 窗口中，根据图 5-21 和表 5-5 所示的内容设置各个参数。

图 5-21. CSD Widget 设置

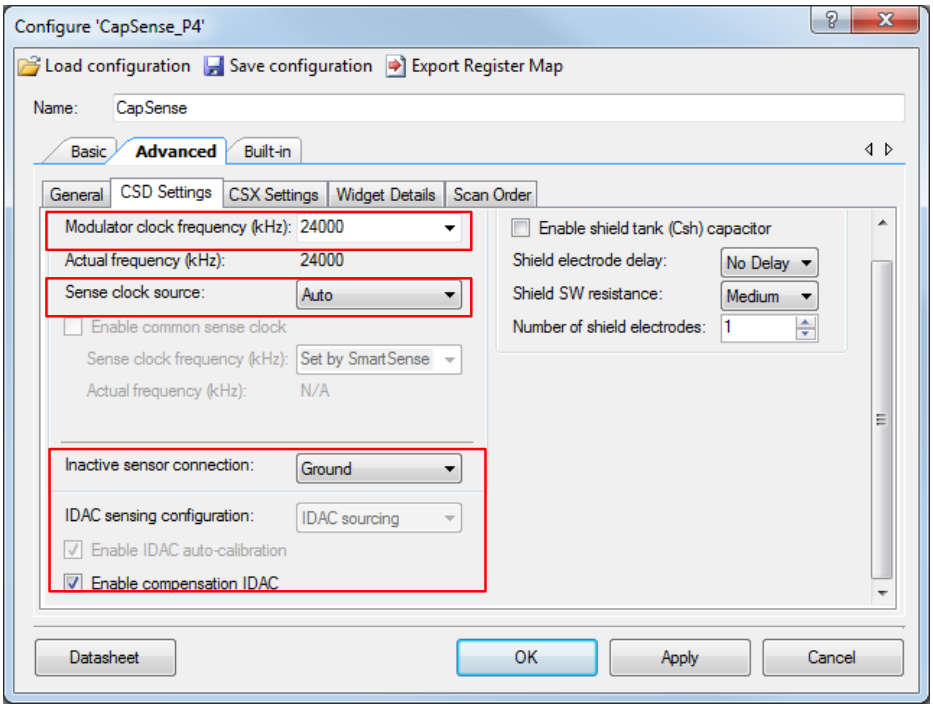


表 5-5. CapSense 组件 CSD 配置窗口

参数	数值	说明
调制器时钟频率	可用最大值选项	较高的调制器时钟频率会缩短传感器扫描时间，从而节省功耗。因此，建议您使用最高的频率。
感应时钟源	Auto（自动）	将感应时钟源自动选择为 **Auto**，这样可选择良好的扩频时钟（SSC）或 PRS 时钟，从而可处理 EMC/EMI 或扁平点 等问题。更多有关选择“直接”或“SSC 时钟”的信息，请参考感应时钟和有关感应时钟的参数等部分的内容。
无效传感器连接	接地（默认）	无效传感器将被接地以提供良好的噪声源屏蔽性能。对于防水的设计或包含接近传感器的设计，可以使用无效传感器连接作为屏蔽。更多有关信息，请参考防水功能部分和 AN92239 — 基于 CapSense 的接近感应应用笔记的内容。
Enable IDAC auto-calibration（使能 IDAC 自动校准）	使能	通过使能自动校准，器件可以自动选择 IDAC 的最佳校准点（在单 IDAC 模式和双 IDAC 模式下，该值分别为 85%和 70%）。对于因环境因素而需要一个不同的校准点的系统，请参考调制 IDAC 和补偿 IDAC 部分的内容。
使能补偿 IDAC	使能	通过使能补偿 IDAC 选择 CSD 的双 IDAC 模式操作。与单 IDAC 模式相比，双 IDAC 模式能够提供更高的信号值，如调制 IDAC 和补偿 IDAC 部分所述。
使能屏蔽电极	根据 PCB 设计	如果您的设计需要较大的接近感应距离和防水功能功能，或使用屏蔽以降低传感器的 C_P，则使能屏蔽。

5. 在 **Advanced** 选项卡的 **Widget Details** 窗口中，根据图 5-22、图 5-23、表 5-6 和表 5-7 所示的内容设置各个参数。

图 5-22. CapSense 组件的“Widget Details”窗口

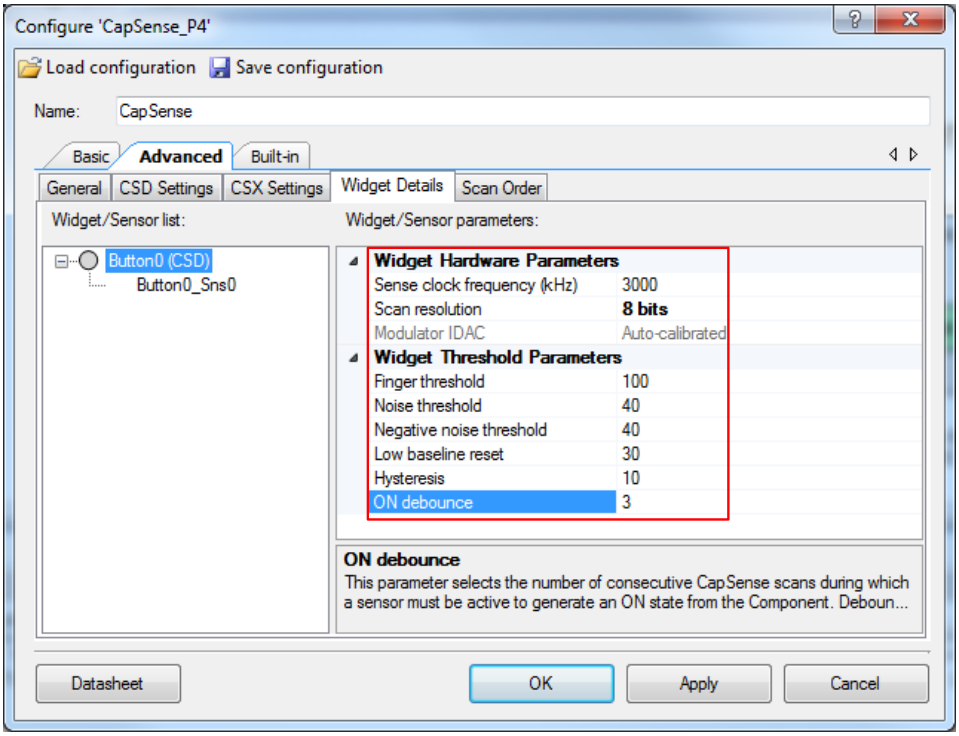


图 5-23. CapSense 组件的“Widget Details”窗口 — 传感器设置

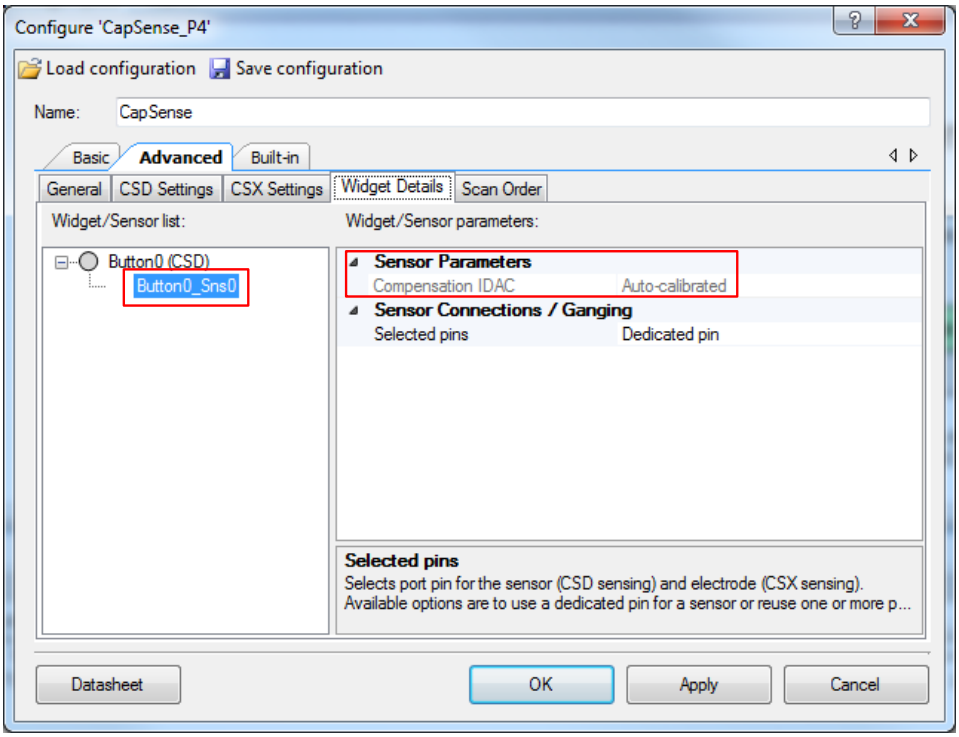


表 5-6. CapSense 组件的“Widget Details”窗口

参数	数值	说明
感应时钟频率	$\frac{1}{10R_{SeriesTotal}C_P}$	请参考感应时钟的相关参数部分以选择合适的频率。
扫描分辨率	8 位	建议选择初始值为“8 位”，该分辨率适用于某些信号，并可确保快速扫描时间。若需要，可以调整该值，如第 8 步中所述。
调制器 IDAC	NA	通过使能自动校准功能，器件可以自动选择该值。请参考调制 IDAC 和补偿 IDAC 部分的内容，从而选择另一个 IDAC 值。
手指阈值（FT）	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。
噪声阈值	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。
负噪声阈值	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。
低基准线复位	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。
迟滞	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。
ON 去抖动	默认值	在调校过程的第 11 步中，调整 Widget 阈值参数。

表 5-7. CapSense 组件的“Widget Details”窗口 — 传感器设置

参数	数值	说明
补偿 IDAC	NA	通过使能自动校准功能，器件可以自动选择该值。请参考调制 IDAC 和补偿 IDAC 部分的内容，从而选择另一个 IDAC 值。
所选引脚	默认设置	通过使用该参数，您可以将多个传感器组合起来作为一个传感器进行扫描。

6. 然后，使用调谐器观察调谐器 GUI 窗口的 Graphing 选项卡中的原始计数值并计算传感器的信噪比。请参考 CapSense 组件数据手册，了解将调谐器添加到您项目的详细操作。要想计算信噪比（SNR），首先要测量存在手指触摸时的原始计数值，然后将该值除以无手指触摸时的原始计数的峰峰值噪声。根据您的终端系统设计，使用尺寸符合正常用例的手指进行测试。通常使用的手指尺寸大概为 8 到 9 mm。
7. 如果 SNR 的初始值大于 5，您可以转到步骤 9。否则，请转到步骤 8。
8. 如果 SNR 小于 5，将其增大，以获得正常性能。分频率和滤波器是影响 SNR 的主要参数。根据表 5-2

   扫描分频率 — 可以增大分频率，从而以与噪声不成比例的速率增加信号值，以提高整体 SNR。根据公式 3-4，增大分辨率将相应增加总扫描时间。

   滤波器 — 滤波器有助于减少噪声而无需增加信号。根据噪声类型，您可以使能一个滤波器来提高 SNR。每个被使能的滤波器会占用存储空间并增加处理时间。

   需要平衡分频率和滤波器间的关系，以得到最好的调校效果。如果您的系统非常噪杂（计数值大于 20），那么您可能需要先添加一个滤波器。但如果您的系统噪声较低（计数值小于 10），您需要优先提高分辨率，因为这样做可以提高系统的灵敏度和信号强度。

   请注意，可以直接在 Widget/Sensor Parameter 窗口中更新分辨率（如图 5-24 所示），但如果要调整滤波器设置，您需要打开 CapSense 配置窗口（Configure‘CapSense_CSD_P4’对话框），并选择相应的滤波器（如图 5-25 所示），然后重新编程器件，以更新滤波器设置。请参阅 CapSense 组件数据手册，了解有关滤波器的详细信息。

图 5-24. 在调谐器 GUI 窗口中更新扫描分辨率

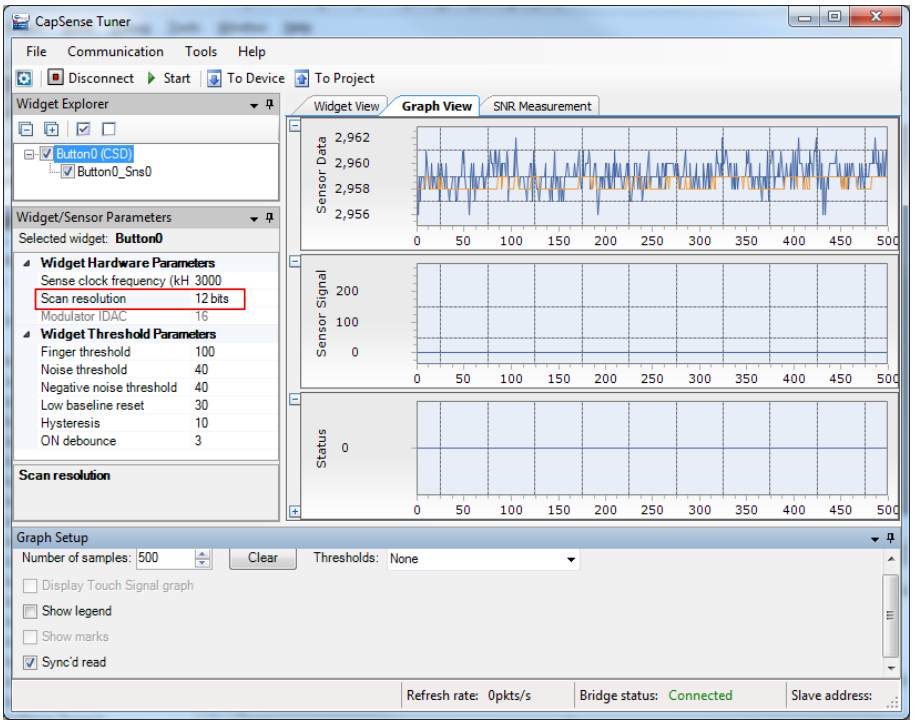
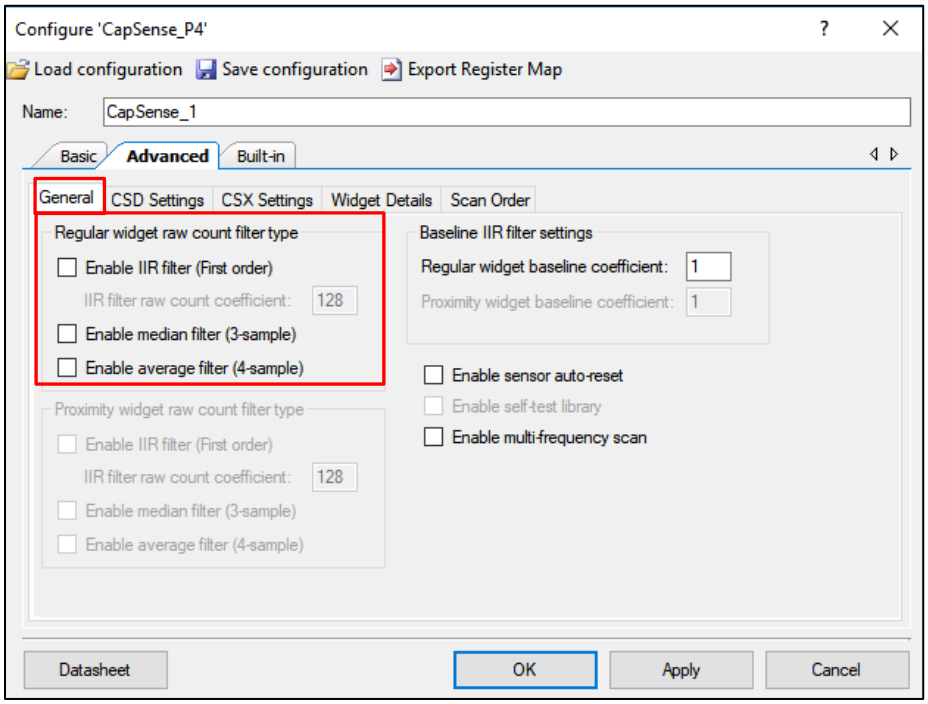
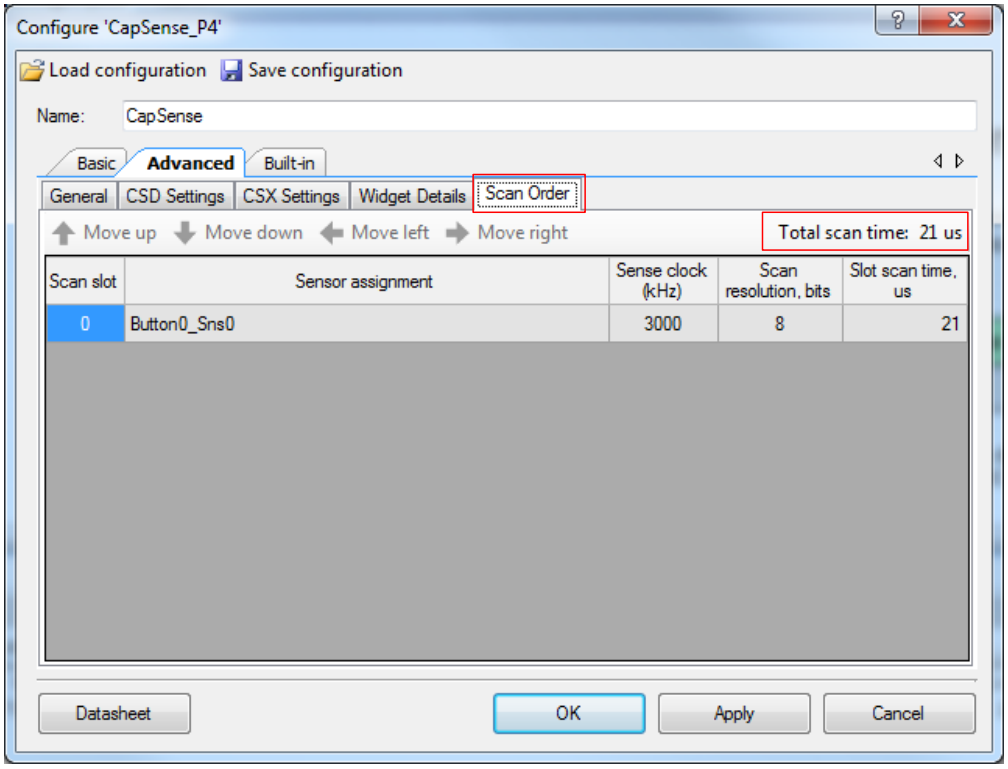


图 5-25. 在 Configure ‘CapSense_CSD_P4’对话框中更新滤波器设置



9. 检查总扫描时间（请参考图 2-26）是否满足系统要求。该时间会影响响应时间，在 CapSense 应用中，它是影响器件整体功耗的重要因素，如功耗与响应时间部分所述。

图 5-26. Scan Order 选项卡中的传感器扫描时间



10. 如果您设计中的扫描时间符合系统时序要求，请转到第 11 步。否则，请调整调校操作以缩短扫描时间。如果所有传感器上的 SNR 都大于 10，那么您可以降低分辨率或移除滤波器，从而缩短扫描时间，但要保持您的 SNR 大于 5。如果您的设计不能保证时序参数和 SNR 大于 5 这类要求，请转到第 12 步。

11. 确认您的设计符合时序参数，且 SNR 大于 5 后，则按照下面所示的信息设置您设计的 widget 阈值参数。请确保您在调谐器 GUI 窗口的 Graph View 选项卡中观察到的是计数差值（**信号**输出），*而不是*原始计数输出。根据您的终端系统设计，使用尺寸符合正常用例的手指进行测试。通常使用的手指尺寸大概为 8 到 9 mm。应该通过使用（系统“拒绝”的）尺寸较小的手指进行测试，以确保所引起的信号达不到手指阈值。

    手指阈值 = 信号值的 80%

    噪声阈值 = 信号值的 40%

    负噪声阈值 = 信号值的 40%

    迟滞 = 信号值的 10%

    去抖动值 = 3

    首先，可以在调谐器 GUI 窗口中设置这些参数（如图 5-27 所示），或者在 CapSense 组件自定义程序中直接输入这些设置，如图 5-28 所示。

    更多有关这些参数设置的信息，请参考选择 CapSense 软件参数。

12. 如果您没有获得大于 5 的 SNR 或无法满足时序要求，请参考手动调校的基础知识的内容了解调校您系统的方法，参考 PCB 布局指南或调校调试常见问题解答（如 5.3.5.4, 5.3.5.7, or 5.3.5.10）。您可能需要修改 CapSense 组件的高级参数和/或调整您的硬件设计，以满足终端系统的要求。

图 5-27. 在调谐器 GUI 窗口中更新阈值参数

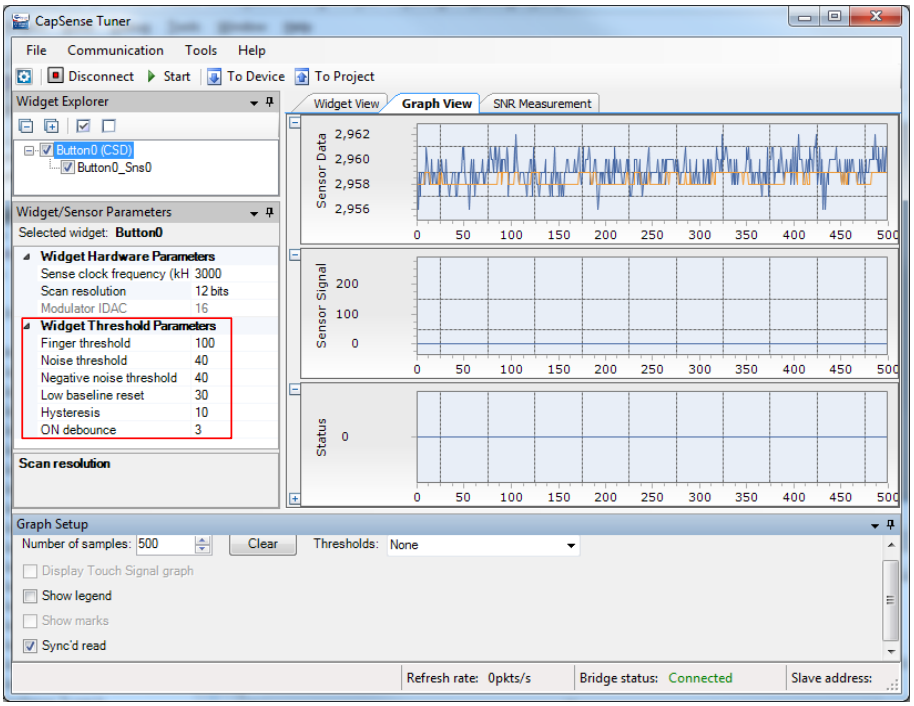


图 5-28. 在 Configure ‘CapSense_CSD_P4’对话框中更新阈值参数

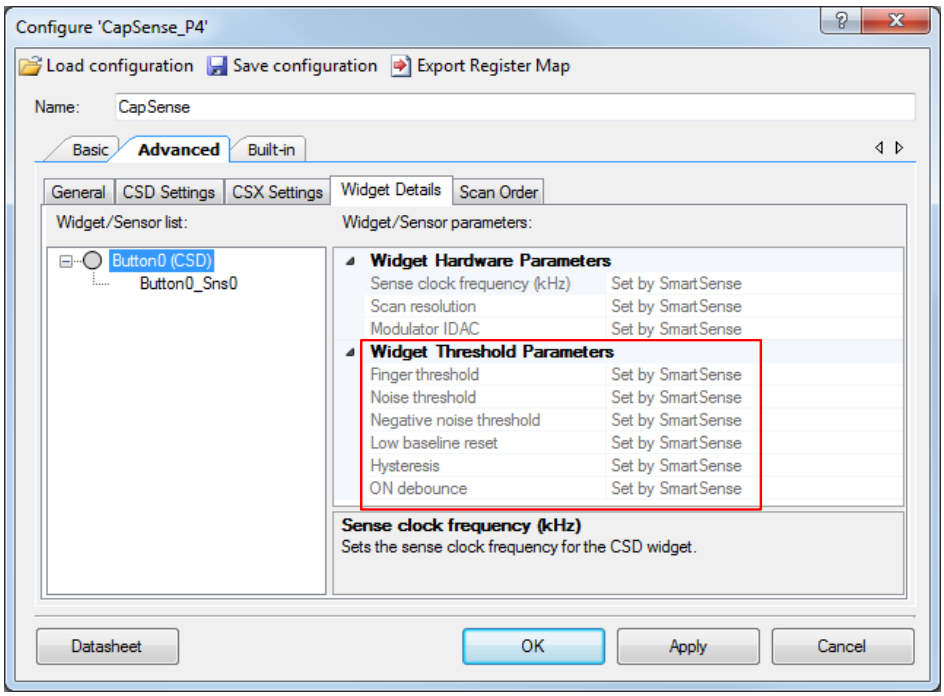
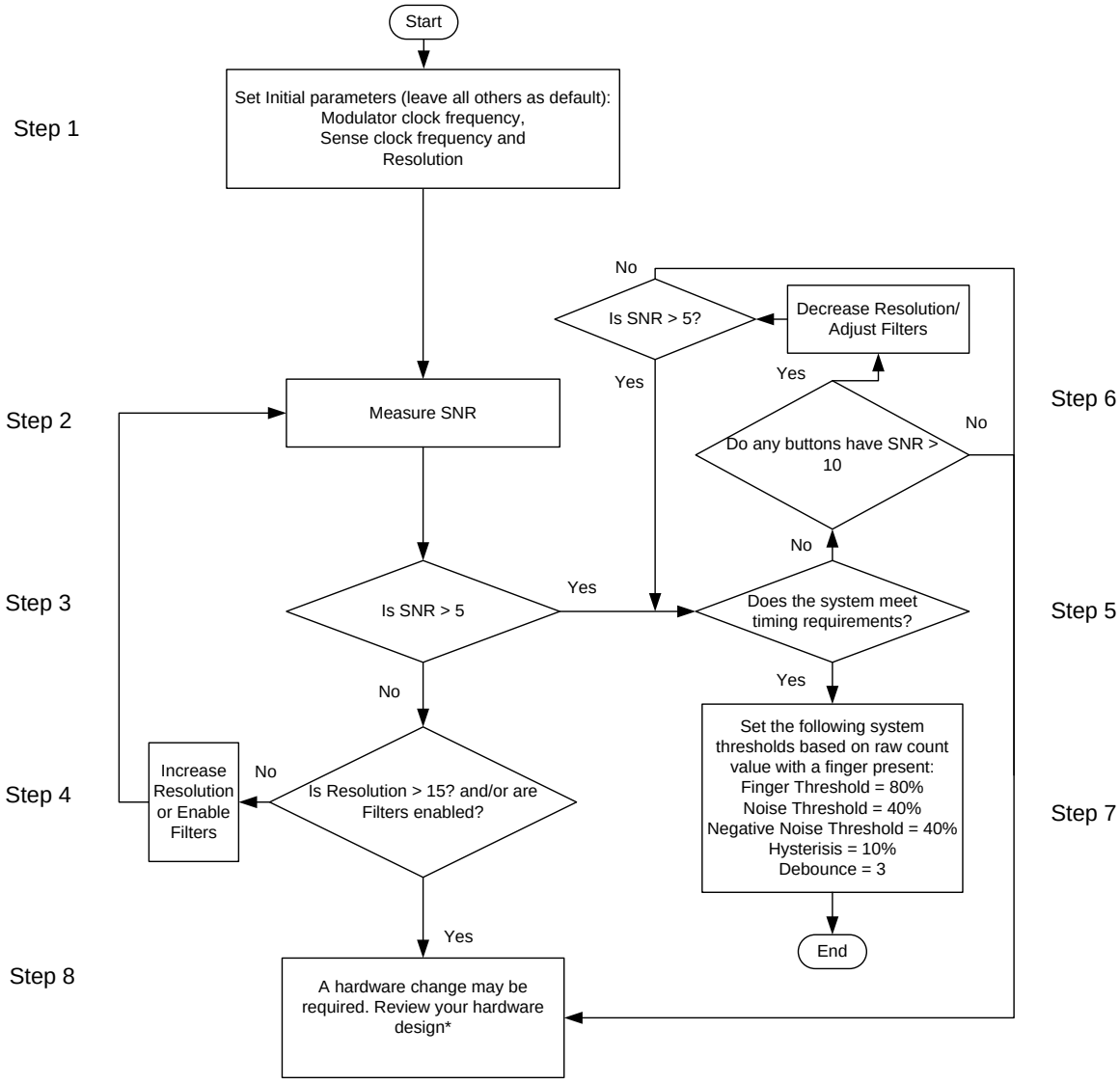


图 5-29. CSD 按键 Widget 调校流程图



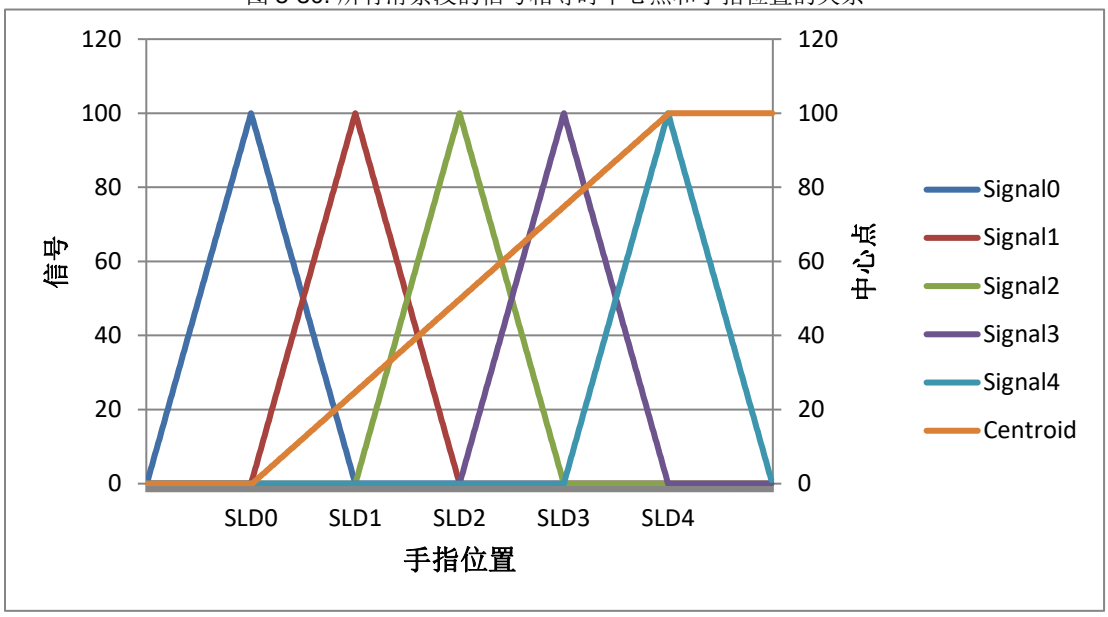
*要查看硬件设计，请参考设计注意事项章节中的传感器结构和 PCB 布局指南节的内容。另外，请参见调校调试常见问题解答一节，了解有关高级调试的指南。

5.3.2.5 滑条 Widget 示例

一个滑条包含多个段，每一段都连接至 PSoC 4 器件的 CapSense 输入引脚。与工作原理简单（打开/关闭）的按键 widget 传感器不同，各滑条 widget 传感器会一起工作，以便跟踪一个手指或其他导电物体的位置。因此，滑条布局设计应确保滑条中所有段的 C_P 值尽可能相近。如果各传感器具有相似的 C_P 值，那么可以减轻调校工作并确保整个滑条上的响应都相同。

为避免质心位置的非线性，必须确保所有滑条段的信号值都一样大，如图 5-30 所示。此处，将手指放置在滑条段中心时，每个滑条段的信号值等于原始计数值减去噪声阈值的差值。噪声阈值如果滑条段的信号不同，则质心位置为非线性的，如图 5-31 所示。在滑条上没有检测到任何手指触摸时，或所有滑条段的计数差值均小于手指阈值时，中心位置值将被报告为 0xFF。

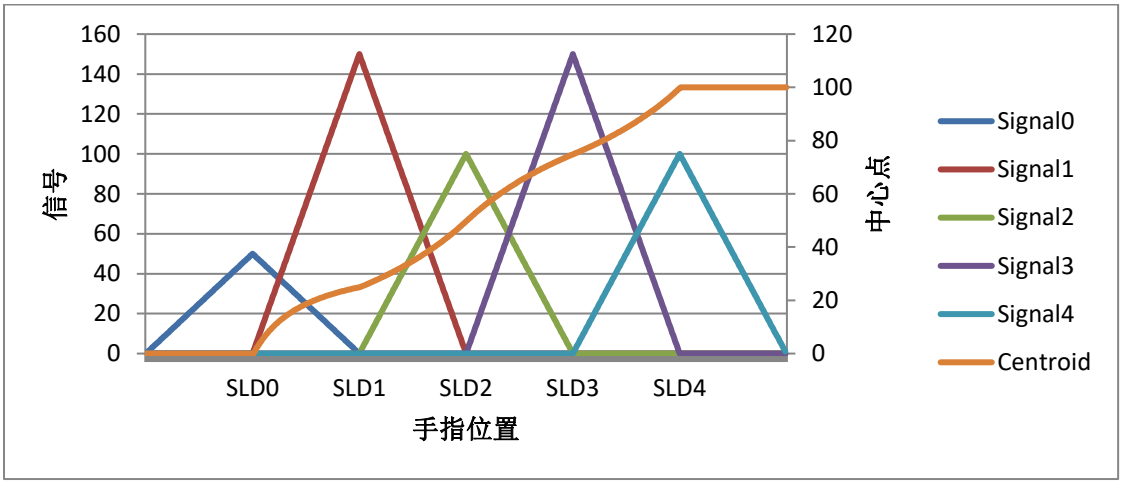
图 5-30. 所有滑条段的信号相等时中心点和手指位置的关系



注意：**信号 = 计数差值 – 噪声阈值**

计数差值 = 原始计数 – 基准线值

图 5-31. 所有滑条段的信号相差时中心点和手指位置的关系



请根据下面的流程手动调校滑条段，使所有滑条段的信号相等：

1. 使用 IDAC 自动校准（后续步骤推荐使用）时，滑条中其他段的 C_P 值需要大于该滑条中最大 C_P 值的 75%。例如，在一个滑条中，如果某一段的最大 C_P 值为 30 pF，则其他段的 C_P 值必须大于 22.5 pF。请确保您的硬件满足该要求。
2. 根据图 5-29 所示的手动调试流程，调试所有滑条段中 C_P 值最大的那一个。

3. 对于其他滑条段，请将下面各参数的值设置为具有最高 C_P 值的滑条段的相应值：
   - 扫描分辨率
   - 感应时钟分频器
   - 调制器时钟分频器
   - 调制 IDAC

   将所有段的这些参数设置为相同的值可确保所有传感器的灵敏度（即由 C_F 值的改变导致的原始计数变化）都一样。

   上述参数是 Widget 级参数，不需要单独设置这些参数。
4. 测量每个滑条段的信噪比，并确保所有滑条段的信噪比都大于 5:1。如果其中某个滑条段的 SNR 不大于 5:1，则需要将所有滑条段的分辨率增加 1 位，直到所有滑条段的 SNR 都大于 5:1 为止。
5. 确认所有滑条段的 SNR 都大于 5:1 后，将下面各阈值设置为表 5-8 所列出的值。

表 5-8. 阈值参数值

阈值参数	建议
手指阈值	信号值的 80%
噪声阈值	信号值的 40%
负噪声阈值	信号值的 40%
低基准线复位	设置为 30

5.3.2.6 接近 Widget 示例

有关调校一个接近感应传感器的信息，请参考 AN92239 — CapSense 接近感应。

5.3.3 CSX 感应方法

本节说明了使用 CSX 感应方法进行手动调校的基础知识。另外，它还介绍了影响手动调校过程的硬件参数。本节以手动调校按键 Widget 的示例结束。

5.3.3.1 基础知识

CapSense 组件版本 3.0 和更高版本都支持互电容感应。在互电容感应系统中，原始计数与 Tx 和 Rx 传感器间的互电容成正比，如公式 5-13 所示。

公式 5-13. 原始计数与传感器电容的关系

$$\text{raw count} = G_{CSX} C_M$$

其中：G_{CSX} 为 CapSense CSX 数字转换增益的电容值，C_M 为 Tx 和 Rx 电极间的互电容。这个转换增益的近似值为：

公式 5-14. 数字转换器增益的电容

$$G_{CSX} = 2 * (2^N - 1) \frac{V_{TX} F_{SW}}{I_{DAC}}$$

通常 V_{TX} 等于 V_{DDIO}。如果 V_{DDIO} 无效，则该电压等于 V_{DDD}。转换增益可调校的参数为 F_{SW}、I_{DAC} 以及 N。

其中：F_{SW} 为 Tx 时钟频率，I_{DAC} 为用于充电和放电 C_{INT} 电容的电流，并且 N 为计数器分辨率。

注意：N 值受子转换次数参数的控制，如 CapSense CSX 感应方式一节中的介绍。

当传感器上有手指触摸时，原始计数将发生变化，这种变化被称为 CapSense 信号。通过增加 N 值和 F_{SW} 值同时降低 I_{DAC} 值，可以增加信号。

5.3.3.2 选择 CapSense 硬件参数

CapSense 硬件参数控制着转换增益和 CapSense 信号。表 5-9 列出了适用于 CSX 感应方法的 CapSense 硬件参数。通过下面部分介绍的指导内容，您可以调整这些参数以得到特定系统的最佳性能。

表 5-9. CapSense 组件硬件参数

Sl. 序号	CapSense 组件参数名称
	CapSense v3.0 或更高版本
1	调制器时钟频率
2	Tx 时钟源
3	Tx 时钟频率
4	子转换次数

5.3.3.2.1 调制器时钟频率

最好为给定器件选择可允许的最高时钟频率，因为调制器时钟频率越高，灵敏度越高，从而提高给定 C_M-数字计数转换的准确度。另外，F_{mod}/F_{tx} 值越高，会确保 C_M-原始计数转换中的扁平点宽度越低。

5.3.3.2.2 Tx 时钟源

该 Tx 时钟源仅适用于 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 系列器件。Tx 时钟采用以下时钟源：

- 直接 — 具有 50%占空比的时钟信号。
- 扩频时钟 — 时钟信号频率在预定范围内动态扩展，以降低 EMI。
- 自动 — 扩频范围自动被选择，这样时钟的最大抖动被限制为±10 %，并保持 HFCLK 频率和 SenseClk 频率之间的比例为 160 或更高，在传感器内部扫描过程中至少执行一个完整的扩频多项式。

建议采用“自动”时钟源，以便获得强大的调校操作和最佳性能。

5.3.3.2.3 Tx 时钟频率

Tx 时钟频率确定了每个子转换所需的时间，如 CapSense CSX 感应方式一节所述。Tx 时钟信号必须对传感器寄生电容进行完全充电和放电。可以通过观察示波器上的信号验证该事件，或通过下面公式对其进行设置

$$F_{Tx} < \frac{1}{10 R_{SeriesTotal} C_P}$$

为了尽可能缩短扫描时间，建议使用组件下拉列表中满足上述条件可用的最大时钟频率。

5.3.3.2.4 子转换次数

传感器的灵敏度取决于子转换参数的数量。请在满足下面条件的基础上选择这些参数的数量。

$$\left(\frac{Modulator\ Clock}{Tx\ Clock} * Number\ of\ Sub\ Conversions\right) < 2^{16}$$

其中，2^{16} 为计数器的最大值

增加子转换次数将增大信号，并延长传感器的扫描时间。例如，对于固定的调制器时钟和 Tx 时钟，以上公式显示 CapSense 原始计数器的分辨率与“子转换次数”参数成正比。将子转换次数降低一半会使计数器分辨率减少 1 位，而且扫描时间缩短一半，从而减少传感器信号，如公式 3-10 所示。请参考按键 Widget 示例一节介绍的内容，了解为“子转换次数”参数配置最佳值的方式。

5.3.3.3 选择 CapSense 软件参数

互电容的 CapSense 软件参数与自电容的相同，因此可以使用同一种方法选择这些参数。更多有关这些参数的详细信息，请参考“CSD 感应方法”一节中显示的选择 CapSense 软件参数。

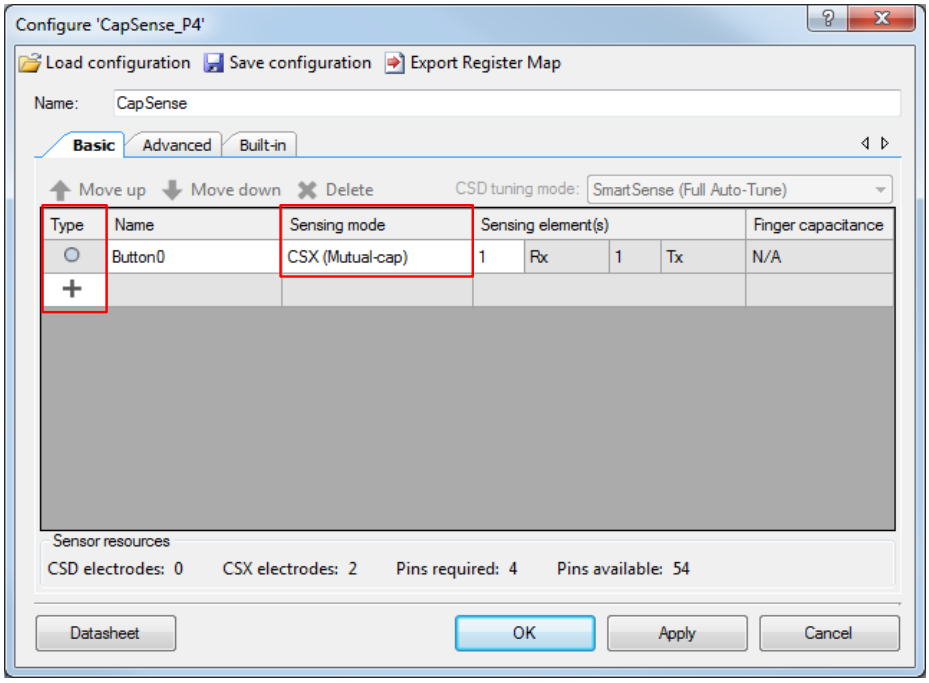
5.3.3.4 按键 Widget 示例

该示例介绍了使用调谐器助手调校 PSoC Creator CapSense 组件中基于互电容的按键 Widget。有关组件和所有相关参数的详细信息，请参考组件数据手册。

按照以下步骤中所列出的详细流程，可以手动设置所有调校参数。请参考图 5-39，快速参考流程图。

1. 双击该组件或右键单击 **Component** 并选择 **Configure** 打开 CapSense 组件配置窗口。
2. 在 **Basic** 选项卡中，点击**+**符号以添加 Widget 按键。将 **Sensing mode** 选为 **CSX (Mutual-cap)** ，如图 5-32 所示。

图 5-32. 添加 CSX 按键 Widget



3. 在 **Advanced** 选项卡**的 General** 窗口中，保持所有滤波器参数的默认设置。根据 SNR 和响应时间的要求使能滤波器参数。
4. 在 **Advanced** 选项卡**的 CSX Settings** 窗口中，按照图 5-33 设置各个参数。
5. 在 **Advanced** 选项卡**的 Widget Details”** 窗口中，按照图 5-34 设置各个参数。

图 5-33. CSX Widget 设置

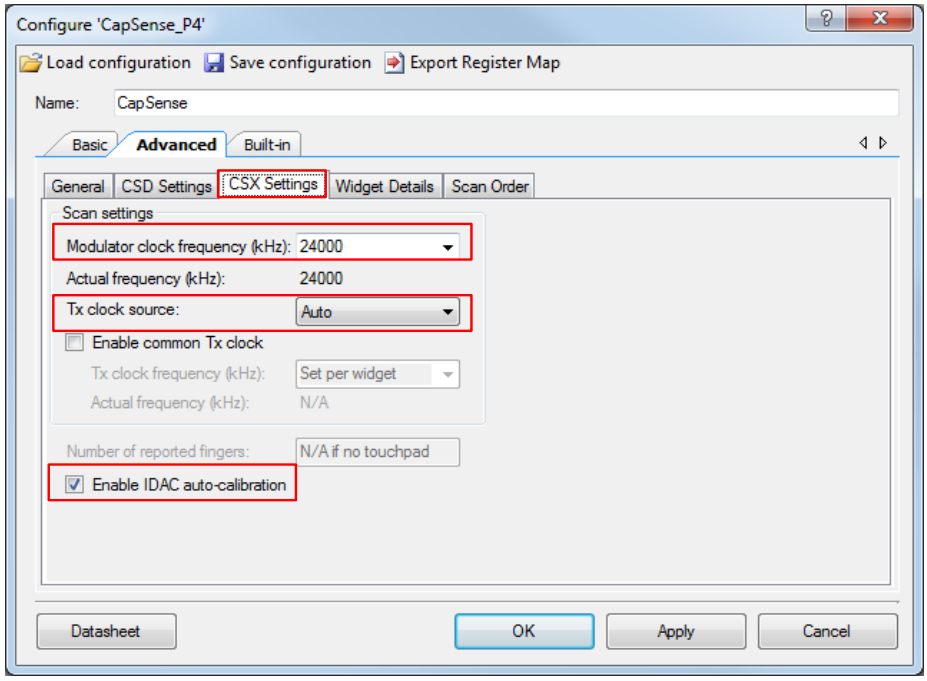


表 5-10. CapSense 组件的“General”配置窗口

参数	数值	说明
调制器时钟频率	可用最大值选项	较高的调制器时钟频率会降低传感器的扫描时间，从而节省功耗，并降低原始计数的噪声。因此，建议使用最高频率。
Tx 时钟源	Auto（自动）	该选项仅适用于 PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC 6 MCU 系列器件。对于其他器件，Tx 时钟源始终为“直接”。 使能“自动”选项（选择最佳扩频时钟值）有助于处理 EMC/EMI。更多有关选择“直接”还是“SSC 时钟”的信息，请参考 Tx 时钟源部分的内容。
Enable IDAC auto-calibration（使能 IDAC 自动校准）	使能	通过使能自动校准，器件可以自动选择 IDAC 的最佳校准点（对于 CSX，最佳校准点为最大值的 40%）。

图 5-34. CapSense 组件的“Widget Details”窗口

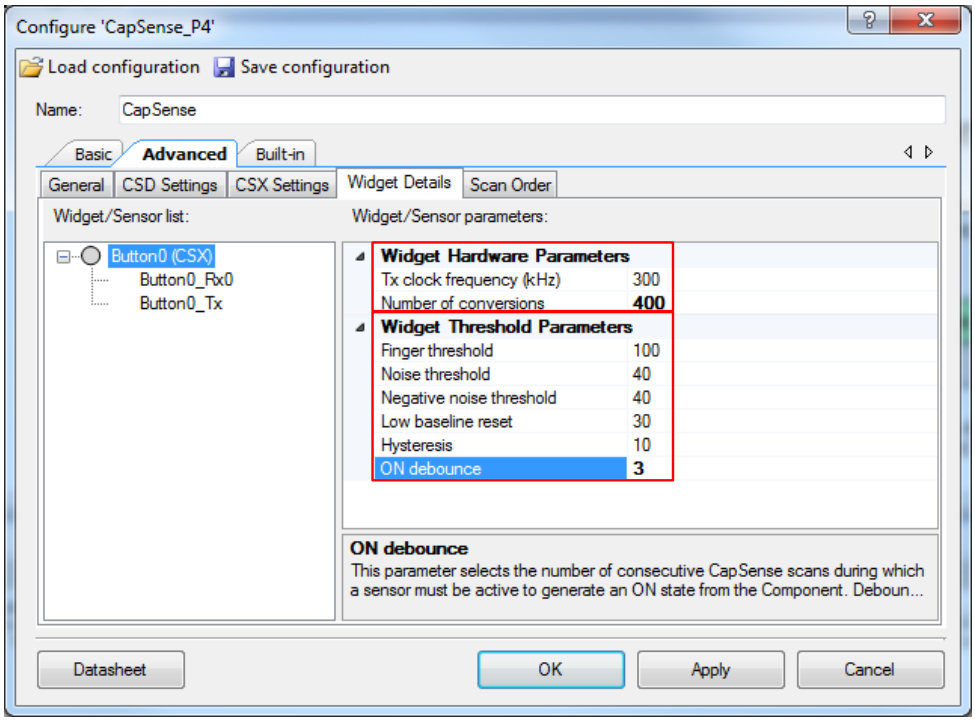


表 5-11. CapSense 组件的“Widget Details”窗口

参数	数值	说明
Tx 时钟频率	$\frac{1}{10R_{SeriesTotal}C_P}$	建议设置满足条件的最高时钟频率。 $Tx\ Clock\ Freq < \frac{1}{10R_{SeriesTotal}C_P}$ 可以使用示波器探测 Tx 引脚，以检查传感器完成的是充电还是放电。
子转换数量	$< \frac{1}{2}\left\{\frac{2^{16} * Tx\ Clock}{Modulator\ Clock}\right\}$	建议使用可获得的最大灵敏度的一半开始操作。若需要，可以调整该值，如步骤 8 中所述。
手指阈值（FT）	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。
噪声阈值	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。
负噪声阈值	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。
低基准线复位	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。
迟滞	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。
ON 去抖动	默认值	在调校过程的步骤 10 内，对阈值设置进行调整。

6. 然后，使用调谐器观察调谐器 GUI 窗口中 Graph View 选项卡的原始计数值，并计算传感器的信噪比。要想计算信噪比，先要测量存在手指触摸时的原始计数值，然后将该值除以无手指触摸时的原始计数的峰-峰噪声。根据您的终端系统设计，使用符合常用手指的尺寸进行测试。通常使用的手指尺寸大概为 8 到 9 mm。
7. 如果 SNR 的初始值大于 5，您可以转到步骤 9。否则，请转到步骤 8。
8. 如果 SNR 小于 5，将其增大，以获得正常性能。子转换和滤波器是影响 SNR 的主要参数。根据表 5-2 为您的应用程序选择相应的滤波器。

   子转换次数 — 可以增大转换次数，从而使用一个与噪声不成比例的速率增加信号值，以提高整个 SNR。增大子转换次数会增大计数器的分辨率和整个扫描时间。

   滤波器 — 滤波器有助于减少噪声而无需增加信号。根据噪声类型，您可以使能一个滤波器来提高 SNR。使用滤波器将需要更多的处理时间和存储器容量。

   需要在子转换和滤波器间进行权衡，以得到最好的调校效果。如果您的系统噪杂很大（计数值大于 20），那么您可能需要优先添加一个滤波器。但如果您的系统噪声较低（计数值小于 10），您需要优先提高分辨率，因为这样做可以提高系统的灵敏度和信号强度。

   请注意，可以直接在 Widget/Sensor Parameters 选项卡中更新子转换次数（如图 5-35 所示），但如果想要调整滤波器设置，则需要打开 CapSense 配置窗口并选择相应的滤波器（如图 5-36 所示），然后重新编程器件以更新滤波器设置。请参阅 CapSense 组件数据手册，了解有关滤波器的详细信息。

图 5-35. 在调谐器 GUI 中更新子转换数量

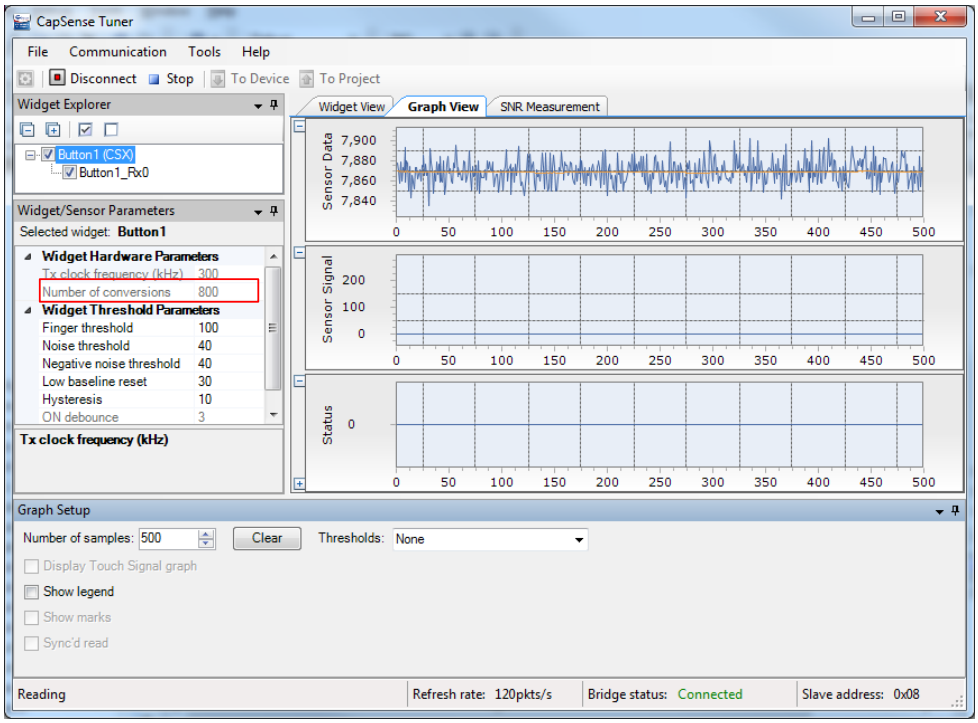
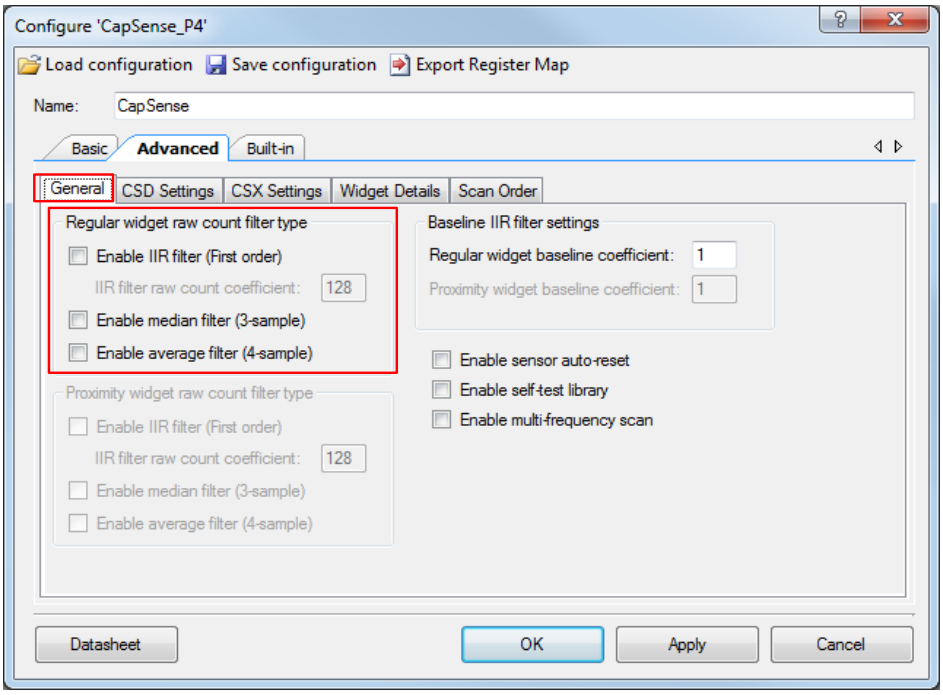


图 5-36. 更新滤波器设置



9. 如果传感器扫描的总时间满足您的要求，请转到步骤 10。否则，请调整调校操作，以缩短扫描时间。如果任意传感器的 SNR 大于 10，您可以降低“子转换次数”参数或清除滤波器，从而缩短扫描时间，但需要保持 SNR 大于 5。

10. 确认您的设计符合时序参数以及 SNR 大于 5 后，请按照下面内容设置设计阈值。请确保您在调谐器 GUI 窗口的 Graph View 选项卡中观察到的是差值计数（即**信号**输出），*而不是*原始计数输出，以便设置这些阈值。根据您的终端系统设计，使用符合正常尺寸的手指进行测试。通常，使用的手指尺寸大概为 8 到 9 mm。应该通过使用（系统“拒绝”的）尺寸较小的手指进行测试，以确保所引起的信号不会达到手指阈值。测量信号后，建议按下面内容设置阈值：

    手指阈值 = 信号值的 80%

    噪声阈值 = 信号值的 40%

    负噪声阈值 = 信号值的 40%

    迟滞 = 信号值的 10%

    去抖动值 = 3

    首先，可以在调谐器 GUI 中设置这些参数，如图 5-37 所示，并通过在调谐器的**“File”**菜单中点击 **Apply to Project** 来保存这些设置，或者在 CapSense 组件自定义程序中直接输入这些设置，如图 5-38 所示。更多有关这些参数设置的信息，请参考选择 CapSense 软件参数。

图 5-37. 在调谐器 GUI 窗口中更新阈值参数

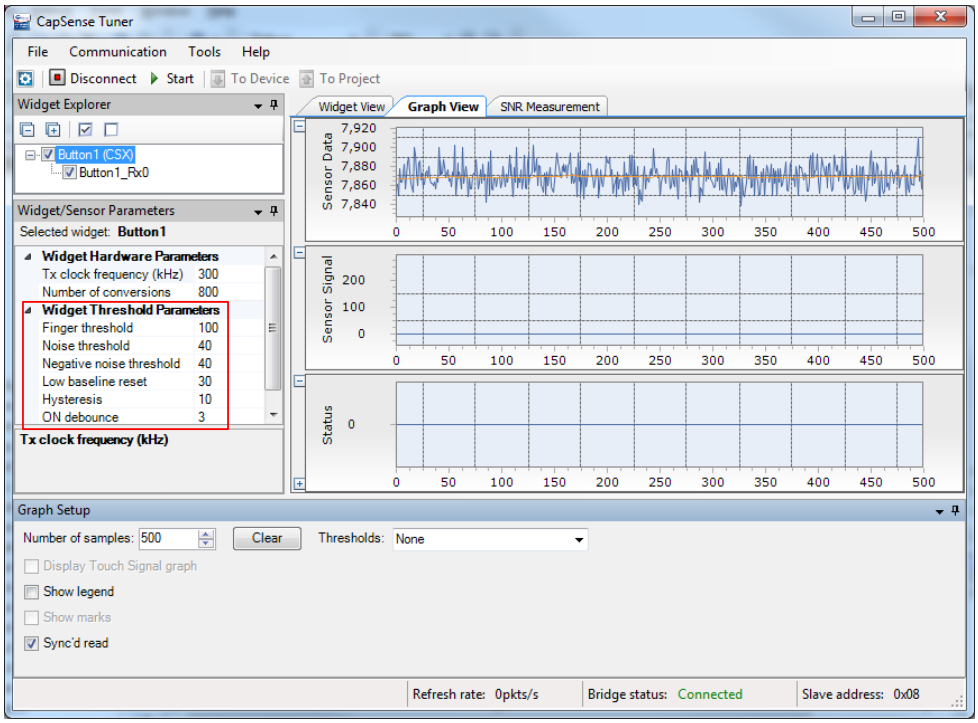


图 5-38 在 Configure ‘CapSense_CSD_P4’对话框中更新阈值参数

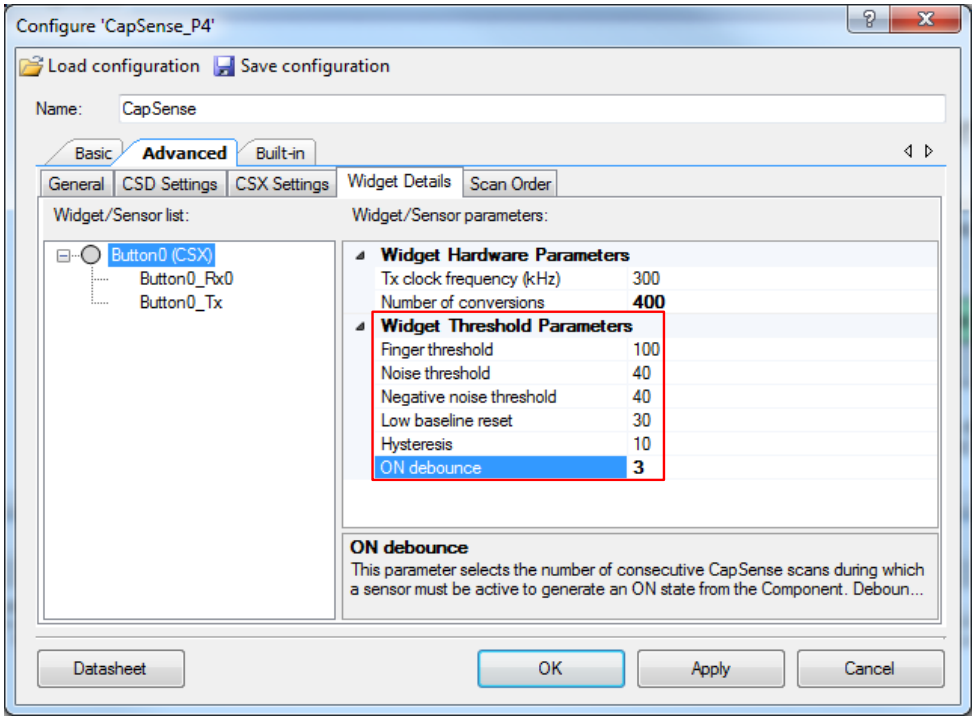
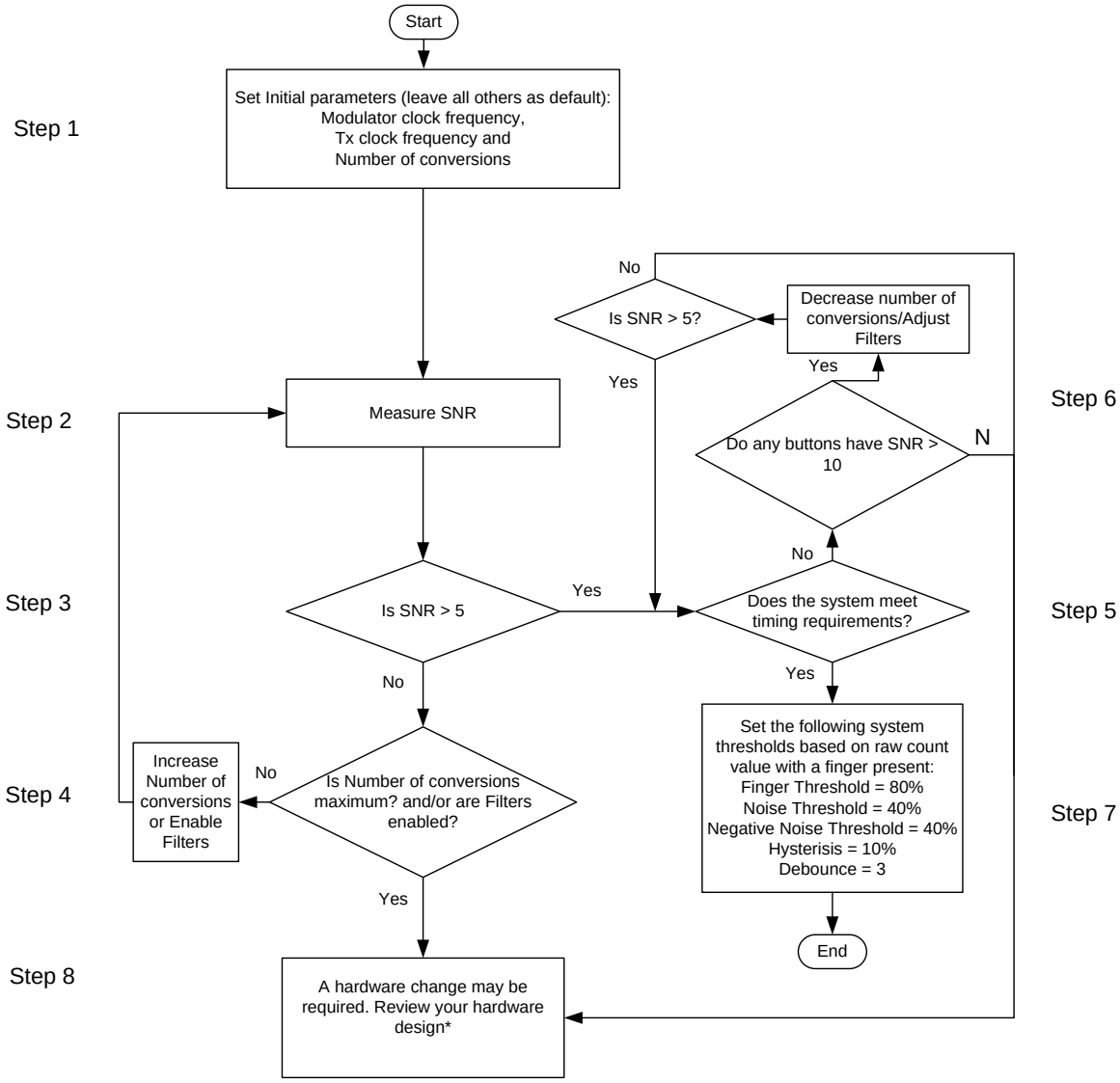


图 5-39. CSX 按键 Widget 调校示例



*要查看硬件设计，请参考设计注意事项章节中的传感器结构和 PCB 布局指南节的内容。另外，请参考调校调试常见问题解答一节获取高级调试指南。

5.3.4 手动调校权衡

手动调校设计时，需要了解设置如何影响电容式感应系统的特性。所有 CapSense 设计都有三个主要特性：可靠性、功耗和响应时间。

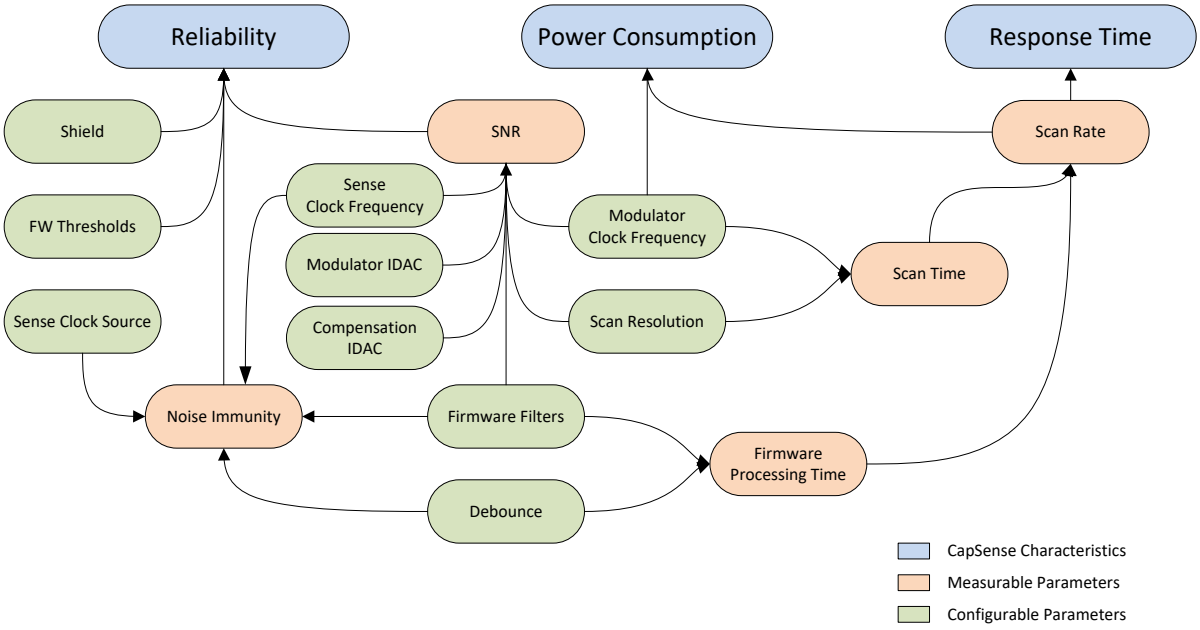
- **可靠性**是指 CapSense 系统在噪杂环境中或在有水滴等不利条件下的运行方式。可靠性高的设计将避免触发误触摸，并能确保在不利条件下发生所有预期触摸。
- **功耗**被定义为器件（包括扫描、处理和低功耗模式转换等操作）所消耗的平均功耗，如低功耗设计一节介绍的内容。通过缩短传感器的扫描和处理时间，可以缩短器件处于较高功耗状态的时间，并尽量延长器件处于较低功耗睡眠状态的时间。
- **响应时间**是指从手指触摸传感器到系统发送响应的时间。因为最低响应时间是由传感器的扫描和处理时间决定的，所以需要正确定义和遵循一个时序预算。总响应时间应该低于 100 ms。

上述这些特性互相依赖。调校过程就是为了找出一个满足项目特定要求的最佳比例。准备设计时，需要注意这些特性一般为反比关系。如您采取任何措施来改善某个特性，那么其他特性将被降低。

例如，如果您想将 CapSense 用到某个玩具上，则需要缩短响应时间并降低功耗。在另一个示例（如烘箱中的“开始/停止”按键）中，可靠性是最重要，而响应时间和功耗是次要的特性。

现在我们考虑影响每个特性的因素。下图显示的是 CapSense 特性、可测量参数及实际 CapSense 可配置参数之间的依赖关系。

图 5-40. CapSense 各参数之间的关系



5.3.4.1 可靠性

以下因素影响可靠性：

1. 信噪比（SNR）：

   SNR 对有效触摸信号提供一个置信测量。为了使 CapSense 可靠运行，SNR 值应大于 5。通过手动调校可以确保在特定设计中获得最佳 SNR。

2. 抗噪能力：

   是指系统能够抵抗外部或内部噪声的能力。外部噪声的典型示例包括：ESD 事件、RF 发送器（如 BLE）、开关继电器、供电电源等。内部噪声源可以是 PWM 驱动的 LED 或 I2C 或 SPI 通信。虽然设计中带有良好的 SNR，但由于它的抗噪能力差，所以它的性能可能很差。手动调校能够对频率和参数进行调校，通过选择不同参数来提供更多的控制，从而避免噪声干扰。

5.3.4.2 功耗与响应时间

以下因素影响功耗和响应时间：

1. 扫描速率

   扫描速率可被定义为扫描传感器的频率。扫描速率决定从手指触摸到报告触摸的最短时间。 传感器扫描时间限制着最大扫描速率。

2. 扫描时间

   指的是对一个特定传感器进行扫描和处理所占用的时间。扫描时间影响功耗（如低功耗设计一节所示）和扫描速率（如上所示）。手动调校可能获得特定的扫描时间，同时仍保持最小的 SNR。

3. 固件触摸延迟

   固件触摸延迟是由 Debounce（去抖动）过程或者使用原始数据噪声滤波器引起的（根据您正在使用的 CapSense 组件版本，请参考表 5-2. ）。这两个因素加长了传感器处理时间，并延迟了触摸报告，直到一定的连续采样数显示触摸信号为止，因此影响扫描时间。

   响应速率在两种情况下均被降低，但可靠性通常得到改善。

   要想了解在 PSoC Creator 中调校 CapSense CSD 参数的方式，以下部分将提供典型示例。另外，可以参考概述、选择 CapSense 硬件参数以及选择 CapSense 软件参数等各节，以便为您的设计提供最佳的手动调校功能。

5.3.5 调校调试常见问题解答

本节列出了关于 CapSense 组件调校一般性的调试问题。要想了解各种可能的原因和解决方案的快速信息，请转到您需要的问题。

5.3.5.1 调谐器不与器件通信

第一个原因： 您的器件未被编程。

解决方案 1：请确保启动调谐器前，使用最近更新的项目来编程器件。

第二个原因： 调谐器配置设置与 SCB 组件设置不匹配。

解决方案 2：打开 EzI2C 从设备配置窗口（即 Configure ‘SCB_P4’对话框），并验证设置是否与 “调谐器通信设置” 对话框的配置相匹配。更多有关调谐器使用情况的信息，请参考 CapSense 组件数据手册中介绍的内容。

第三个原因： 您的 I2C 引脚的配置不正确。

解决方案 3：在 Workspace Explorer 中打开.*cydwr* 文件，并确保引脚分配与物理连接到电路板上的引脚分配相匹配。

第四个原因： 您没有 CapSense TunerStart API 或其他所需的调谐器代码。

解决方案 4：将 CapSense 组件数据手册中列出的调谐器代码添加到 *main.c* 中，并重新编程器件。

5.3.5.2 我无法通过调谐器来更新器件上的参数

第一个原因：器件上的通信设置不正确。

解决方案 1：查看并确保“Configure SCB_P4”对话框中的设置与“调谐器通信设置”对话框中的设置相匹配。确保子地址的大小相等。

5.3.5.3 我可以连接到器件，但没有看到任何原始计数

第一个原因：您未将调谐器代码添加到项目中。

解决方案 1：查看调谐器助手一节，并将调谐器代码添加到您的 *main.c* 中，然后重新编程器件。

5.3.5.4 当传感器上存在手指触摸时，计数差值仅有少量的改变（10 到 20 个计数）

第一个原因：系统增益太低。

解决方案 1：查看本文档的调谐器助手一节。

第二个原因：您的传感器寄生电容值非常高。

解决方案 2：要确认该问题，请使用组件数据手册中记录的内置自测试（BIST）API。通过这些函数，您可以读取传感器寄生电容的估计值。另外，您还能够通过使用 LCR 测量仪独立确认该读取操作。

如果您的硬件具有用于使能驱动屏蔽信号和屏蔽电极的选项，那么在 CapSense 组件配置窗口的 Advanced 设置中使用该选项。围绕传感器的驱动屏蔽有助于降低寄生电容。使能该选项时，通过将 Advanced 设置中“Inactive Sensor connection”字段的设置改为“shield”，可为未使用的传感器使能驱动屏蔽。使能屏蔽后，如果 C_P 值仍大于 45 pF，那么请联系赛普拉斯技术支持，以便查看您的布局。

第三个原因：您的覆盖层可能太厚。

解决方案 3：查看与覆盖层材料相对应的覆盖层厚度。

第四个原因：原始计数可能太接近饱和值，因此触摸传感器时将发生饱和现象。

解决方案 4：调校 IDAC，对某个给定的传感器，确保将原始计数调校为分辨率的~85%（根据电磁兼容性（EMC）注意事项一节）。

5.3.5.5 调校系统后，在测试过程中，我看到了大量的辐射噪声

第一个原因：感应时钟频率导致系统中的辐射噪声。

解决方案 1：降低感应时钟频率，或者为您的传感器使能 PRS（根据电磁兼容性（EMC）注意事项一节）。如果已经使能了 PRS，请参考电磁兼容性（EMC）注意事项一节。

第二个原因：较大的屏蔽电极可能是导致大量辐射噪声的原因之一。

解决方案 2：根据耐水性布局指导一节，降低屏蔽电极的大小。

5.3.5.6 手动调校后，扫描时间不再满足系统要求

原因：系统的噪声和 C_P 为高，需要更多的扫描时间和过滤时间，从而使其可靠运行。

解决方案：需要降低 C_P 值。首先，在 CapSense 组件配置窗口的“Advanced”设置中使能驱动屏蔽信号和屏蔽电极，并确保通过查看 PCB 布局指南将增益设置为尽可能大的值。如果您的系统仍不能满足最终的要求，那么您需要更改电路板布局来进一步降低 C_P，请查看 PCB 布局指南。

5.3.5.7 我无法将系统校准为 85%

原因：您的传感器可能被短路接地。

解决方案：首先，使用万用表检查您的传感器和地层间是否被短接。如果存在短路情况，请查看您的原理图和布局，找到错误。

原因：可能是您的传感器 Cp 过高或过低。

解决方案：如果您的硬件具有用于使能驱动屏蔽信号和屏蔽电极的选项，那么在 CapSense 组件配置窗口的 Advanced 设置中使用该选项。围绕传感器的驱动屏蔽有助于降低寄生电容。如果您没有允许您使用扩展板的硬件选项，或者在启用扩展板后，您的 C_P 仍然超过器件支持的 Cp，请联系赛普拉斯技术支持，以协助您检查您的布局或获取特定应用的更多指南。

5.3.5.8 我的滑条中心位置的响应是非线性的

原因：布局可能不符合硬件设计指南，从而不能确保适当的线性度。

解决方案：通过使用 CapSense 配置窗口 General 选项卡中内置自测试项（BIST 项仅适用于 CapSense 版本 2.40）来检查传感器的 C_P 值，然后根据滑条设计一节更新布局。更多有关 BIST API 的信息，请参考 CapSense 组件数据手册中介绍的内容。

5.3.5.9 滑条段的 CP 值变化很大

原因：您的布局设计使传感器的 C_P 不平衡。

解决方案：需要更新您的布局。查看滑条设计，并根据要求更新布局。如果无法立即实现，应该重新调校每个传感器，以便获得类似的响应。这是一个长期的重复过程，首选方法是更新硬件（若可以）。

5.3.5.10 切换 GPIO 时，原始计数显示电平的转换或增加的噪声

原因 1：在 PCB 上传感器走线与切换 GPIO 并行布线。

解决方案：需要更新您的布局。请参阅设计指南中走线路由 和 串扰解决方案部分内容，并根据要求更新您的布局。如果在当前阶段无法更改布局，可以考虑使用固件滤波器，以降低噪声，从而提高信噪比。

原因 2： 正在在 GPIO 上灌入大量电流

解决方案：当扫描 CapSense 传感器时，限制通过 GPIO 灌入的直流灌电流。请参考原理图规格检查表。如果通过固件控制 GPIO 上的灌电流，并且由灌电流引起的不频繁原始计数电平转换与触摸信号的差异较大，那么，每次启用通过 GPIO 灌入灌电流的事件时，可以实现固件技术，例如复位/重新初始化 CapSense 基准线。

原因 3： 您没有遵循相应 PSoC6 器件系列数据手册中勘误表部分所述的指南。

解决方案：遵循相应 PSoC6 器件数据手册中勘误表部分所述的建议，例如驱动模式强度选择、切换频率限制、CapSense 引脚选择指南，等等

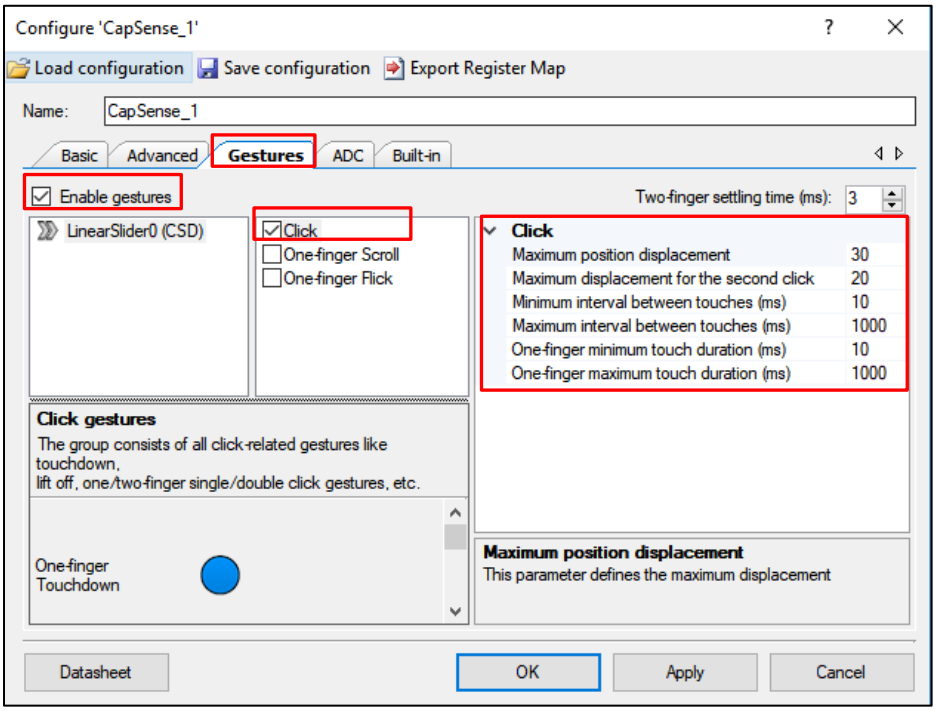
6 CapSense 手势

6.1 CapSense 的手势支持

PSoC 4 和 PSoC 6 中的 CapSense 组件支持手势检测功能，即根据预定义的触摸格式来识别不同的手势。

CapSense 组件中 **Gestures** 选项卡提供了与手势相关的所有配置参数。一旦使能了手势，将按照 widget/手势组系统性排列所有手势参数，如图 6-1 所示。

图 6-1. 配置 CapSense 组件手势



在右侧面板中点击某个手势组名称，显示与该手势组相关的参数。按照下面步骤来使能各手势并配置相应参数：

- 在左侧面板中（widget 面板）选择需要使能手势性能的 widget。
- 在中间面板中（手势组面板）选中所需的手势组。
- 配置左侧面板（参数面板）中的所有参数。

6.2 手势组

手势性能组包括：Click（点击）、One-finger Scroll（单手指滚动）、Two-finger Scroll（双手指滚动），Two-finger Zoom（双手指缩放)、One-finger Edge Swipe（单手指边缘滑动）、One-finger Flick（单手指快速滑动）、One-finger Rotate（单手指旋转）。

表 6-1 显示的是不同 widget 所支持的手势。

表 6-1. 不同 CapSense Widget 支持的手势

Widget 类型	手势组						
	点击	单手指滚动	双手指滚动	单手指快速滑动	单手指边缘滑动	双手指缩放	单手指旋转
按键							
线性滑条	✓	✓		✓			
辐射滑条	✓						
矩阵按键							
触摸板	✓	✓	✓	✓	✓	✓	✓
接近感应							

有关配置手势所使用的每个参数的更多详细信息，请参阅相应的组件数据表（支持手势部分）。

7 设计注意事项

本章节说明了 CapSense 的固件和硬件设计中的注意事项。

7.1 固件

PSoC CapSense 组件提供了多个用以简化固件开发的应用编程接口。CapSense 组件数据手册提供了可用 API 的详细名单以及说明内容。您可以通过使用 PSoC Creator 中提供的 CapSense 示例项目来了解原理图输入和固件开发。请参见第 4 章了解更多详细信息。

CapSense 扫描具有无阻塞的性质。开始和结束 CapSense 扫描之间无需 CPU 的干预。所以您可以使用 CPU 执行其他任务，同时仍能够进行 CapSense 扫描。然而，请注意 CapSense 是一种高灵敏度的模拟系统。因此，在设备中的电流突然发生变化时，可能增加原始计数中存在的噪声。如果您正在使用需要高灵敏度的 widget（如接近感应传感器或覆盖层较厚的按键），那么请使用阻塞扫描方式。下面显示的是严格无阻塞扫描的固件示例。

```
/* Enable global interrupts */
    CyGlobalIntEnable;

  /* Start EZI2C component */
    EZI2C_Start();

    /*
    * Set up communication data buffer to CapSense data structure to be
    * exposed to I2C master at primary slave address request.
    */
    EZI2C_EzI2CSetBuffer1(sizeof(CapSense_dsRam),
    sizeof(CapSense_dsRam),
    (uint8 *)&CapSense_dsRam);

    /* Initialize CapSense component */
    CapSense_Start();
    /* Scan all widgets */
    CapSense_ScanAllWidgets();

    for(;;)
    {
        /* Do this only when a scan is done */
        if(CapSense_NOT_BUSY == CapSense_IsBusy())
        {/* Process all widgets */
            CapSense_ProcessAllWidgets();
            /* Scan result verification */
            if (CapSense_IsAnyWidgetActive())
            {
                /* Add any required functionality
                based on scanning result */
            }
            /* Include Tuner */
            CapSense_RunTuner();
            /* Start next scan */
            CapSense_ScanAllWidgets();
        }
        /* CPU Sleep */
```

```
        CySysPmSleep();
    }
}
```

进行 CapSense 扫描的过程中，需要避免中断的代码、功耗模式的转换以及外设的 ON/OFF 切换。然而，如果您没有采用高灵敏度的 widget，则可以使用 CPU 来执行其他工作。另外，您还可以通过使用 PSoC 4 的低功耗模式来降低 CapSense 系统的平均功耗，如下一节中介绍的内容。**如果您正在使用无阻塞代码，则推荐使用调试器 GUI 来监控并验证原始计数值和信噪比。**

要通过 ModusToolbox 软件开发固件，请参考本文档中的 ModusToolbox 一节。

7.1.1 低功耗设计

PSoC 4 的低功耗模式允许降低总功耗，同时保留其基本功能。请参考 AN86233：PSoC 4 低功耗模式和降低功耗技术，了解 PSoC 4 低功耗模式的基础知识，并参考 AN210998：PSoC® 4 低功耗 CapSense®设计，了解对低功耗 CapSense 应用进行的设计。

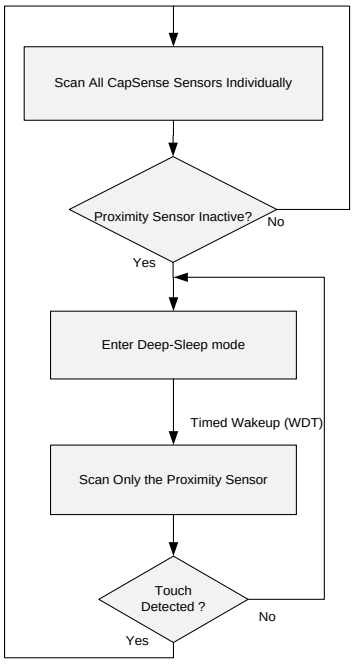
在 CapSense 扫描过程中，无需 CPU 的干预。如果固件除了等待扫描完成外不用执行任何其他任务，那么在启动扫描后可使器件进入睡眠模式，以节省功耗。当 CSD 硬件完成了扫描操作时，它会生成一个中断，使器件返回活动模式。

如果使用 API 同时扫描多个传感器，则在单一传感器的扫描完成后，器件会返回活动模式。因此，如果设计中有多个传感器，应该单独扫描每一个传感器。

可以使用 PSoC 4 的深度睡眠模式来大幅度降低 CapSense 设计的功耗。但是在深度睡眠模式下，CapSense 硬件被禁用。由此，必须经常唤醒器件，以扫描触摸。在频繁的时间间隔中，可以通过使用 PSoC 4 中的看门狗定时器（WDT）将器件从深度睡眠模式唤醒。增大扫描的频率会加快 CapSense 系统的响应速度，但同时也会增大平均功耗。

由于设计中的传感器数量增加，器件必须处于更长时间的活动模式，以扫描所有传感器。这样会增大平均功耗。如果设计中存在多个传感器，则应添加一个围绕着各个传感器的单独接近感应传感器环路。当器件从深度睡眠模式中唤醒时，只能扫描这个接近感应传感器。如果接近感应传感器已被激活，则器件必须处于活动模式并扫描其他传感器。如果接近感应传感器未被激活，则器件可返回深度睡眠模式。图 7-1 显示了该过程。

图 7-1. 低功耗 CapSense 设计



要进一步降低功耗，应在无需 CPU 的活动但需要高速度的外设（如系统定时器和 I²C）时，可以使器件进入睡眠模式。

注意：在 PSoC 4000 器件系列中，如果正在进行 CapSense 扫描，不推荐使器件进入睡眠模式。

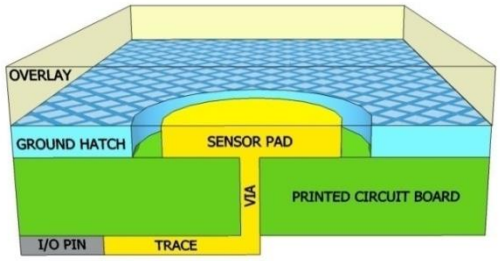
另外，您还可以在设计中添加“屏蔽网格”，如驱动屏蔽信号和屏蔽电极一节所示，以降低寄生电容以及缩短扫描时间。通过准确调校 CapSense 和唤醒的时间间隔，可以获取超低的系统电流，同时仍保持良好的触摸响应。

7.2 传感器结构

根据应用的要求，可以使用各种材料来构建电容式传感器。在典型的传感器结构中，用于检测触摸的导电垫片或表面会通过一条导电走线或链接与电容式控制器的引脚相连。整个布置在一个绝缘的覆盖层下面进行的，用户则在覆盖层的顶层上进行交互操作。

图 7-2 显示的是 CapSense 传感器的最通用结构。

图 7-2. CapSense 传感器结构

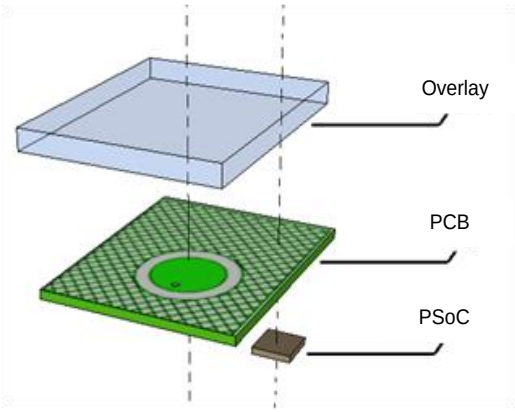


将蚀刻在 PCB 表面的覆铜作为 CapSense 传感器使用。绝缘覆盖层作为触摸表面。覆盖层也保护传感器不受环境的影响，并防止手指的直接接触。围绕传感器板的网格地将它与其他传感器及其走线分开。

如果需要防水功能，应使用屏蔽网格而不使用接地网格。即使用屏蔽信号驱动网格，而不用使其接地。请参考防水功能一节，了解详细信息。

最简单的 CapSense PCB 设计是一种顶端含传感器垫片和网格接地，底端含电子组件的两层电路板。图 7-3 显示的是 CapSense 硬件的分解图。

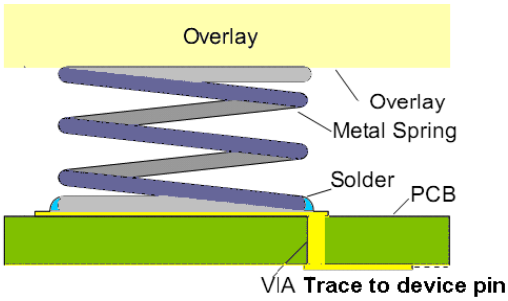
图 7-3. CapSense 硬件



另外，通过使用铜质除外的材料（如铟锡氧化物（ITO））或者基板上的电子印墨（如玻璃或柔性 PCB）可以构成传感器。在某些情况下，弹簧也可以作为CapSense传感器使用，如图7-4所示，用于创建抬动式传感器，支持将覆盖层放置在离PCB

一定距离的位置上。请参考 CapSense 设计指南入门，了解指定于弹簧传感器和其他非铜传感器（如 ITO、电子印墨）的 PCB 设计注意事项。

图 7-4 将弹簧作为传感器使用的传感器结构



7.3 覆盖层选择

7.3.1 覆盖层材料

覆盖层是 CapSense 硬件的关键部分，因为它决定了手指电容值的大小。手指电容值与覆盖层材料的相对介电常数成正比。请查看手指电容了解详情。

表 7-1 显示的是一些常见覆盖层材料的相对介电常数。具有相对介电常数介于 2.0 和 8.0 之间的材料很适合用作 CapSense 覆盖层。

表 7-1. 覆盖层材料的相对介电常数

材料	ε_r
空气	1.0
Formica（福米卡）	4.6 – 4.9
玻璃（标准）	7.6 – 8.0
玻璃（陶瓷）	6.0
PET 薄膜（Mylar®）	3.2
聚碳酸酯（Lexan®）	2.9 – 3.0
丙烯酸（Plexiglas®）	2.8
ABS	2.4 – 4.1
木表和桌面	1.2 – 2.5
石膏（石膏板）	2.5 – 6.0

注意：由于导电材料干扰电场模板。因此，导电材料不应用作盖板。也不应在覆盖层上使用包含金属颗粒的油漆。

7.3.2 覆盖层厚度

手指电容与覆盖层的厚度成反比。因此，与较厚的覆盖层相比，较薄的覆盖层可提供更大的信号。请查看手指电容了解详情。

表 7-2 列出了针对不同 CapSense Widget 所推荐的丙烯酸覆盖层的最大厚度。

表 7-2. 丙烯酸覆盖层的最大厚度

Widget	最大厚度（mm）
按键	5
滑条	5[4]
触控板	0.5

由于手指电容还受覆盖层的介电常数的影响，因此介电常数在选择覆盖层的最大厚度中也起到重要作用。常用玻璃的介电常数 ε_r 大约为 8，而丙烯酸的介电常数 ε_r 约为 2.5。ε_r/2.5 的比例是塑料覆盖层厚度的估计值，这样可以保持相同的灵敏度。按照该经验，就算常用玻璃覆盖层的厚度是塑料覆盖层厚度的 3 倍，但是仍能够保持相同的灵敏度。

对于 CSX 感应，建议覆盖层厚度最少为 0.5 mm。

7.3.3 覆盖层粘合剂

覆盖层与 PCB 之间要有良好的机械接触。应该使用绝缘粘合剂薄膜将覆盖层粘接到 PCB 上。该薄膜能够消除覆盖层和传感器板间的空气间隔，所以会增大系统的灵敏度。3M™提供了一种型号为 200MP 的高性能丙烯酸粘合剂。在各个 CapSense 应用中广泛采用了该粘合剂。其包装为粘合剂转移胶带；例如，产品型号为 467MP 和 468MP。

7.4 PCB 布局指南

通过 PCB 布局指南，您可以设计具有良好的灵敏度和较高信噪比的 CapSense 系统。

7.4.1 寄生电容 C_P

C_P 的主要组成部分是走线电容和传感器电容。请参考 CapSense 基本原理，了解详细信息。C_P 与 PCB 布局特性之间的关系并不简单。发生下面任何情况时，会增加 C_P：

- 传感器垫片大小增加
- 走线长度和宽度增加
- 传感器垫片和接地网格之间的间隙减少

应尽量减少走线的长度和宽带，以降低 C_P 值。降低走线长度会增加抗噪能力。不推荐降低传感器垫片的尺寸，因为这样也会降低手指的电容。

降低 C_P 值的另一种方法是加大传感器垫片和接地网格之间的间隙。然而，加大间隙同时也会降低抗噪能力。此外，通过使用屏蔽信号驱动网格，也可以降低 C_P 值。

更多有关信息，请参考驱动屏蔽信号和屏蔽电极一节的内容。

如果由于较长的走线或接近地层所引起的传感器 C_P 的值非常大，您可以使用互电容感应方法，以便在 C_P 值较高的情况下仍不降低灵敏度。使用互电容感应方法时，CapSense 传感器的灵敏度独立于传感器 C_P。

[4] 对于厚度为 5 mm 的丙烯酸覆盖层，SmartSense 组件要求手指直径最少为 9 mm，以确保滑条能够正常运行。如果手指直径小于 9 mm，应使用手动调校方法。

7.4.2 电路板的各个层

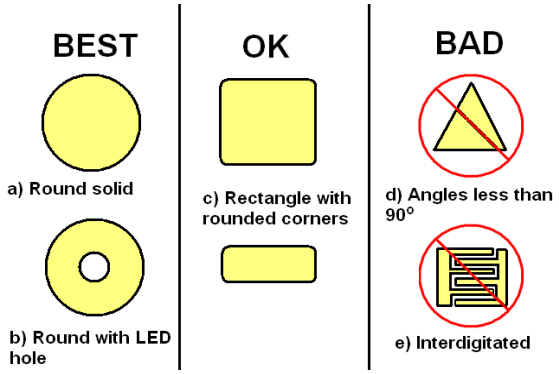
大多数应用都采用双层电路板，传感器垫片和网格地在顶层，其他所有的组件都在底层。更复杂的 PCB 会有四层。在基于 FR4 的 PCB 设计中，如果电路板的厚度介于 0.020 英寸（0.5 mm）到 0.063 英寸（1.6 mm）之间，那么可保证该设计的操作良好。

可以在 CapSense 设计中使用柔性电路。推荐在弧形表面上使用柔性电路。本文档中介绍的所有 PCB 指南也适用于柔性电路。应该为 CapSense 使用厚度为 0.01 英寸（0.25 mm）或更厚的柔性电路。用于柔性电路的 Kapton®材料的崩溃电压（290 kV/mm）为 CapSense 传感器提供了内置的 ESD 保护。

7.4.2.1 自电容按键设计

推荐为 CapSense 按键使用圆形传感器垫片。也可以使用圆角矩形垫片。应避免尖角（小于 90º 角）垫片，这是因为它们会集中电场。图 7-5 显示的是推荐的按键形状。

图 7-5. 推荐的按键形状

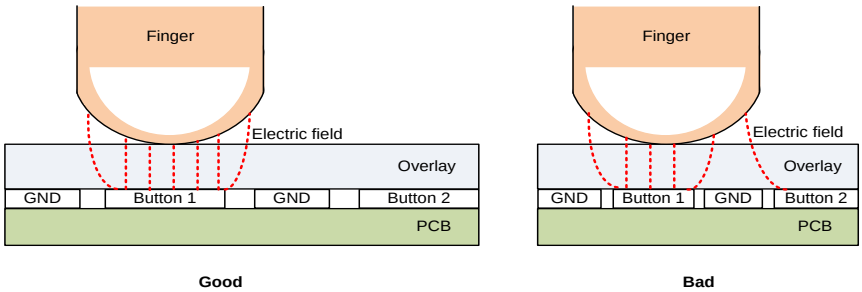


按键直径范围应介于 5 mm 到 15 mm 之间，其中 10 mm 适用于大多数应用。对于较厚的覆盖层，则要使用较大的按键直径。

传感器垫片和地网格之间的间隙宽度应等于覆盖层的厚度，其范围为 0.5 mm 到 2 mm 之间。例如，如果覆盖层的厚度为 1 mm，推荐使用 1 mm 宽的间隙。但对于 3 mm 厚的覆盖层，则只应使用 2 mm 宽的间隙。

适当选择两个相邻传感器间的距离，使得在触摸按键时手指不会接近其他传感器和接地网格之间的间隙，以避免旁边按键上的误触摸检测，如图 7-6 所示。

图 7-6. 各按键之间的距离



7.4.2.2 互电容按键

互电容感应将测量两个电极间的电容耦合的变化。传感器模型应按照这种方式设计，以便使手指最大程度影响各电极间的电场。图 7-7 和图 7-8 分别显示的是对厚度小于 1.5 mm 和大于 1.5 mm 的覆盖层的互电容按键设计。

传感器面积是互电容传感器的重要参数。对于覆盖层厚度（t），最小的传感器面积（a）可通过以下公式得出。

$$A = \frac{0.25pF \times t}{\varepsilon_0 \times \varepsilon_r}$$

其中，ε_0为空气的介电常数（8.85 x 10-15 F/mm），ε_r为覆盖层的介电常数（也称为相对介电常数）。

图 7-8 和图 7-9 中所示的最小电极大小要等于覆盖层厚度的 2 倍，以确保信号良好。

图 7-7. 薄覆盖层的互电容按键设计

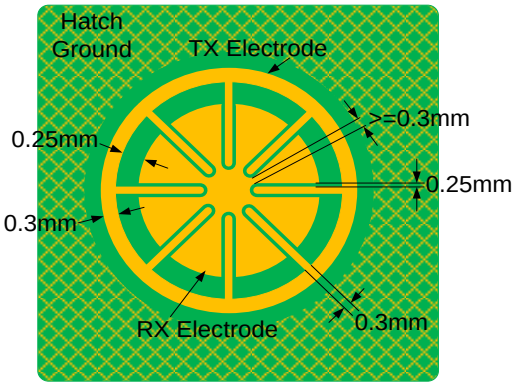
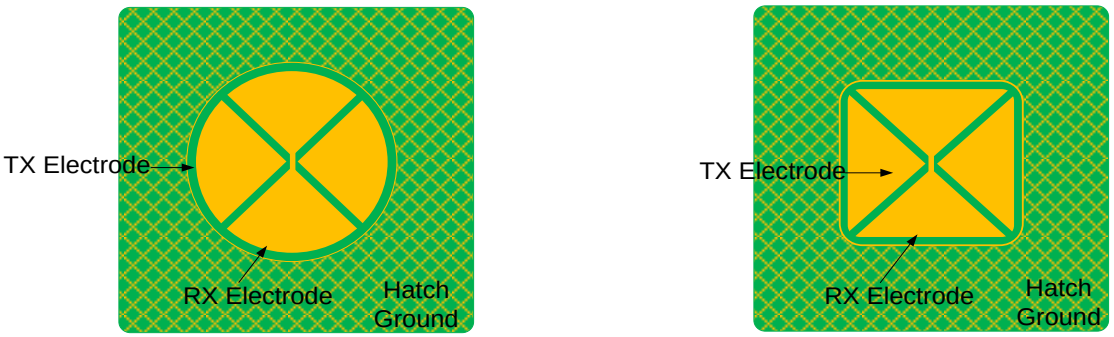
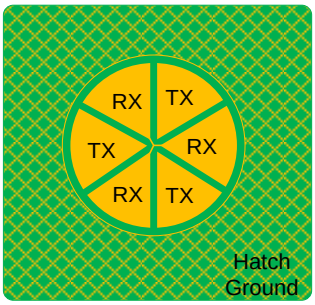


图 7-8. 厚覆盖层的互电容按键设计



对于较厚的覆盖层（> 1.5 mm），按键为大尺寸，在这种情况下，建议增加 Tx 和 Rx 段的数量，如图 7-9 所示。这样会增加各电极间的耦合，并且手指触摸传感器时，会增加信号。

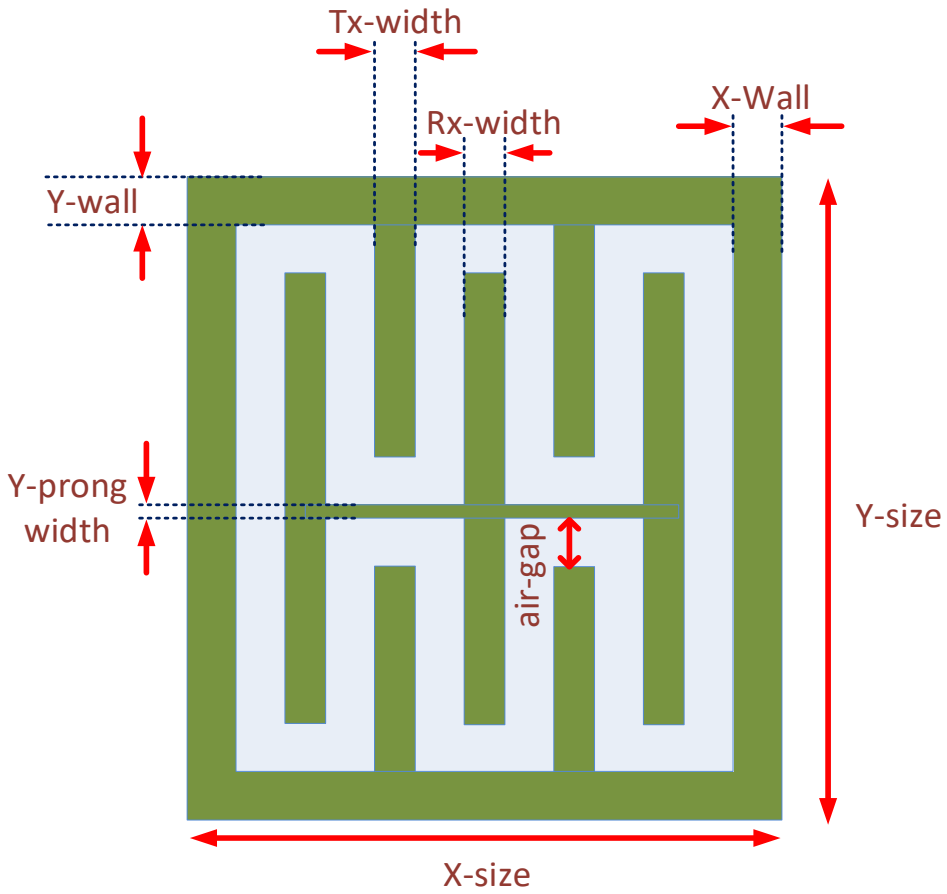
图 7-9. 具有多个 Tx 和 Rx 段的互电容按键设计



围绕传感器的网格接地会降低传感器原始计数中的噪声，从而提高信噪比。对于自电容按键，网格填充的百分比可以与表 7-9 中列出的一样。

齿或鱼骨也是互电容按键的标准形状。Tx 在按键的外侧形成一个盒子或环，从而降低 Rx 的噪声。边界内部有交错的 Tx 和 Rx 插脚，以形成电场。图 7-10 显示了三叉鱼骨传感器结构的示例。Tx 电极和 Rx 电极之间的气隙应为覆盖层厚度的一半。Tx 电极的外壁与共平面网格地之间的间隙应大于 Tx 和 Rx 电极的气隙。

图 7-10. 互电容按键设计的鱼骨图案



7.4.3 滑条设计

图 7-11 显示的是一个推荐的线性滑条模型，而表 7-3 显示的是推荐的每个线性滑条尺寸。下面各节中介绍了有关推荐的布局指南的详细说明。

图 7-11. 典型的线性滑条模型

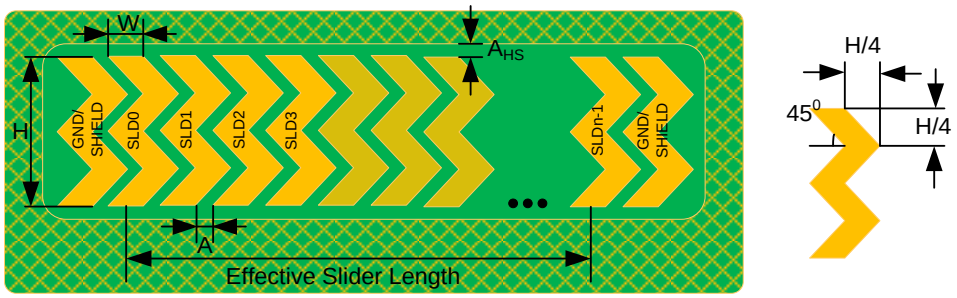


表 7-3. 线性滑条尺寸

参数	丙烯酸覆盖层的厚度	最小值	最大值	建议
段的宽度（W）	1 mm	2 mm	-	8 mm[5]
	3 mm	4 mm	-	
	4 mm	6 mm	-	
段的高度（H）	-	7 mm[6]	15 mm	12 mm
各段间的气隙（A）	-	0.5 mm	2 mm	0.5 mm
网格和滑条间的气隙（A_{HS}）	-	0.5 mm	2 mm	等于覆盖层的厚度

7.4.3.1 滑条段的形状、宽度和气隙

所报告的手指位置（即为中心位置）和滑条上的实际位置间的线性响应要求一个滑条设计，在 SLD0 段中点和 SLDn-1 段中点之间的任意位置上（而不是触摸在滑条段的中心位置）存在触摸点时，只有两个传感器报告有效信号[7]。如果手指触摸任何滑条段的中心位置，则相邻传感器会报告计数差值等于手指阈值。因此，推荐使用双 V 字形，如图 7-11 所示。通过这种形状，可以使中心响应接近理想响应，如图 7-12 和图 7-13 所示。出于同样原因，滑条段的宽度和气隙（分别为图 7-11 中所标记的“W”和“A”尺寸）应该遵循公式 7-1 所提及的关系。

[5] 推荐的滑条段宽度是根据 9 mm 的人手指平均直径而设定的。更多信息，请参阅滑条段的形状、宽度和气隙一节。

[6] 推荐的 7 mm 最小滑条段高度是根据 7 mm 的最小人手指直径而设定的。当手指触摸任何段的中点时，如果覆盖层厚度和对 CapSense 进行调校使之滑条获得信噪比≥ 5:1，则可将滑条高度设置为小于 7 mm。

[7] 此处，有效信号意味着已给滑条段的计数差值不得小于噪声阈值。

图 7-12. 理想滑条段信号和中心位置的响应

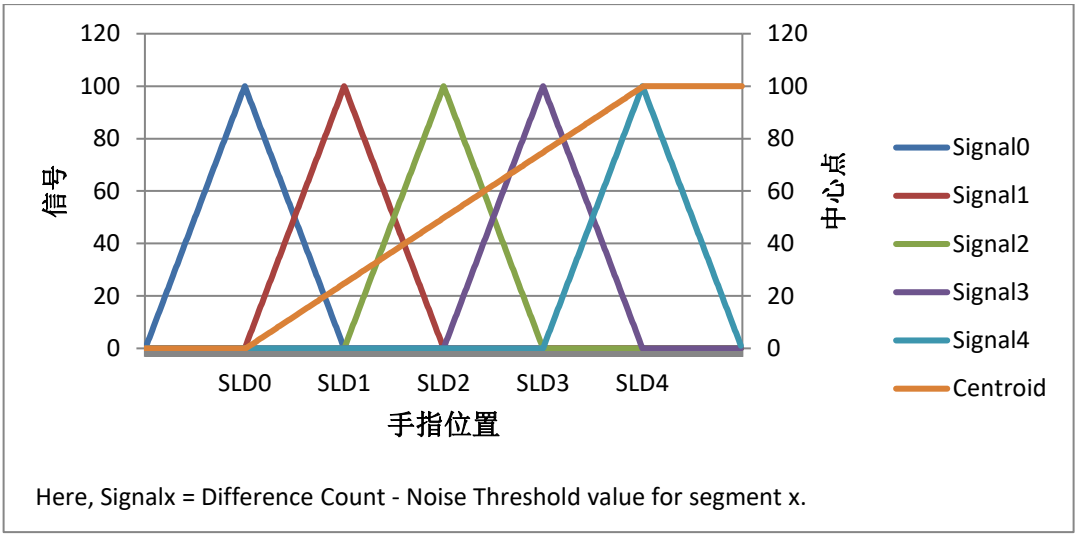
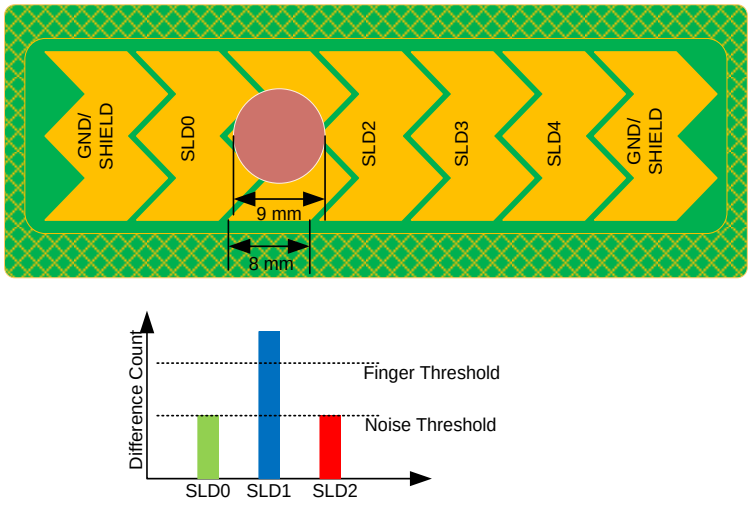


图 7-13. 理想滑条信号



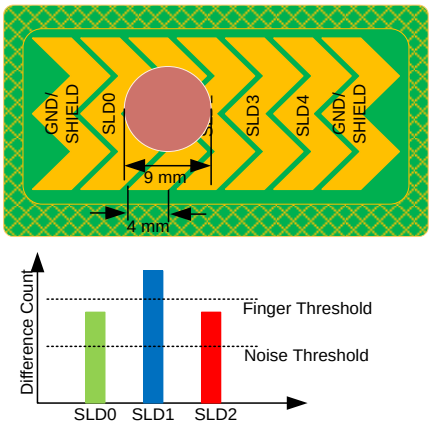
公式 7-1. 滑条段宽度、气隙和手指直径的关系

$$W + 2A = finger\ diameter$$

通常情况下，人手指的平均直径大约为 9 mm。根据这个手指的平均直径和公式 7-1，建议的滑条段宽度和气隙分别为 8 mm 和 0.5 mm。

如果 *slider-segment-width + 2 * air-gap* 小于公式 7-1 所要求的 *finger diameter*，则中心响应是非线性的。这是因为滑条上放置的手指会引进电容，从而在特定位置上向两个以上的滑条段提供有效信号，如图 7-14 所示。因此，通过公式公式 7-2 计算得出的中心位置是非线性的，如图 7-15 所示。

图 7-14. 滑条段宽度小于推荐值时，手指会导致在多于两个滑条段上生成有效信号



公式 7-2. 在 PSoC Creator 中 CapSense 组件使用的中心算法

$$Centroid\ position = \left(\frac{S_{x+1} - S_{x-1}}{S_{x+1} + S_{x0} + S_{x-1}} + maximum\right) * \frac{Resolution}{(n-1)}$$

其中：

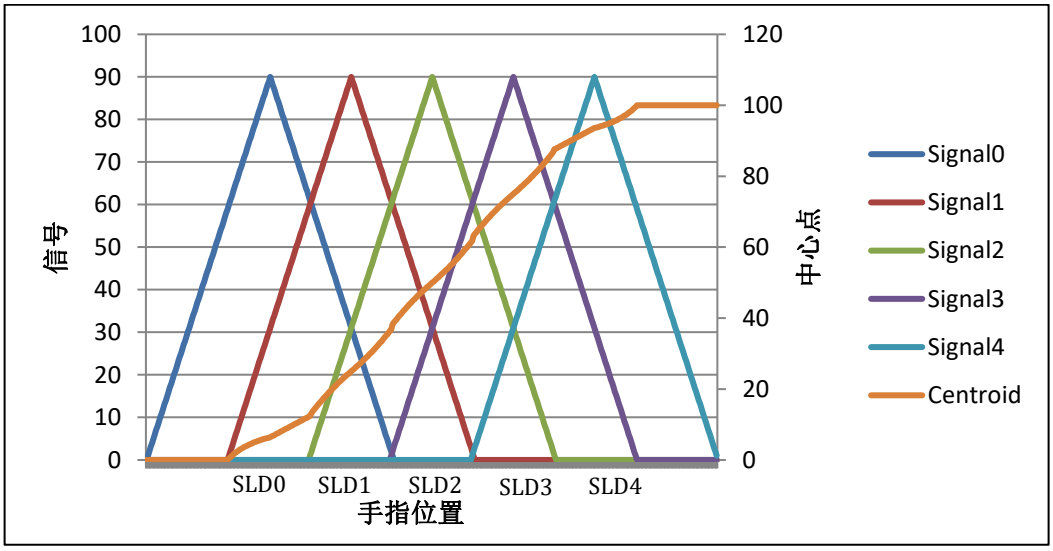
Resolution — CapSense 组件自定义程序中所设置的 API 分辨率

n — CapSense 组件自定义程序中传感器元素的数量

maximum — 提供了最大信号的元素指数

Si — 靠近信号强度最大位置的计数差值（已经减去了手指阈值）

图 7-15. 滑条段宽度小于推荐值时的非线性中心响应



请注意，虽然 *slider-segment-width* 值小于 *finger diameter – 2 * air-gap* 时中心响应是非线性的（如图 7-15 所示），但它仍可用于终端应用，其中所报告的中心位置与实际的手指位置的线性关系并不重要。但是，必须根据覆盖层的厚度保持滑条段的最小宽度，以便在有效滑条长度内任意位置都至少有一个滑条段能提供 ≥ 5:1 的信噪比（即信号值 ≥ 手指阈值参数）。如果滑条段宽度过小，那么手指不能耦合足够的容值，因此所有滑条段的信噪比不能达到 5:1，从而会将中心位置值报告为 0xFF[8]，如图图 7-16 所示。

图 7-16. 滑条段宽度过小时所导致的中心位置错误报点

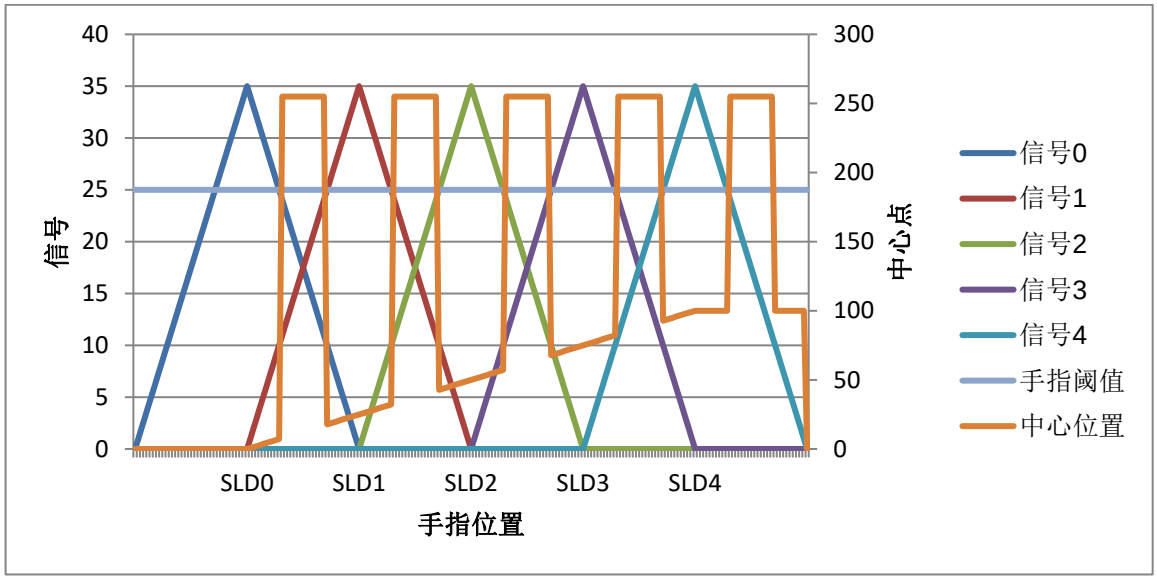
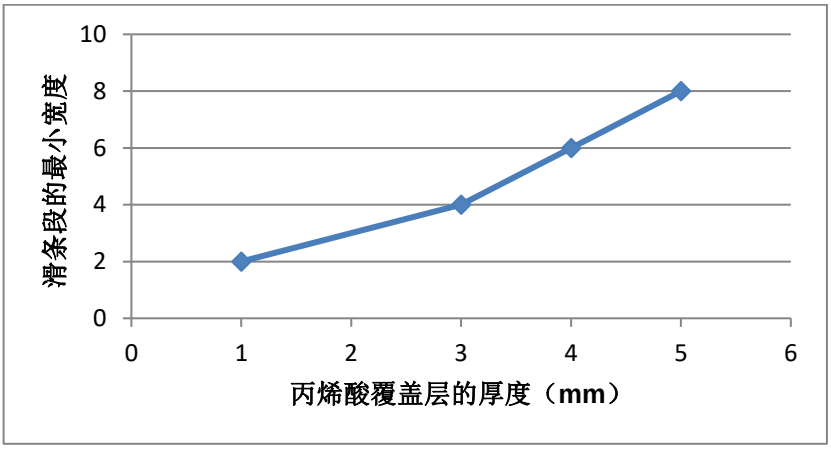


表 7-3 针对丙烯酸覆盖层的具体厚度提供了滑条段宽度的最小值。 对于表 7-3 中未指定的丙烯酸覆盖层厚度，可以使用图 7-17 来评估滑条段的最小宽度。

图 7-17. 滑条段的最小宽度与丙烯酸覆盖层厚度间的关系



[8] 在滑条上没有检测到任何手指触摸时，或所有滑条段的计数差值均小于手指阈值时，PSoC Creator 中 CapSense 组件会将中心位置值报告为 0xFF。

如果 *slider-segment-width + 2 * air-gap* 大于公式 7-1 所要求的 *finger diameter*，则中心响应中会有扁平点，换句话说，如果手指移动到任何滑条段的中点，则所报告的中心位置会保持不变，如图 7-18 所示。这是因为手指放置在滑条段的中间位置时，即使手指向相邻滑条段移动了极小的位置，它也只会报告该滑条段的信号有效，如图 7-19 所示。

图 7-18. 滑条段宽度大于推荐值时的扁平点（无响应的中心位置）

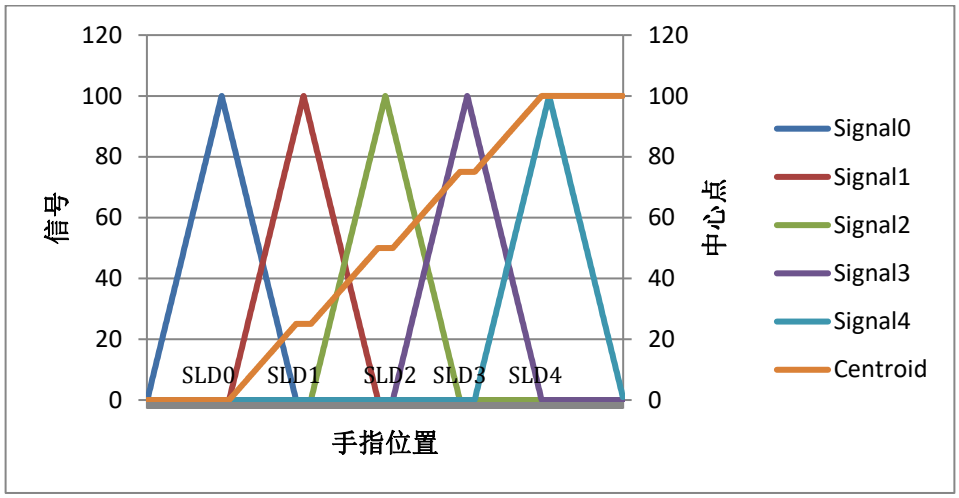
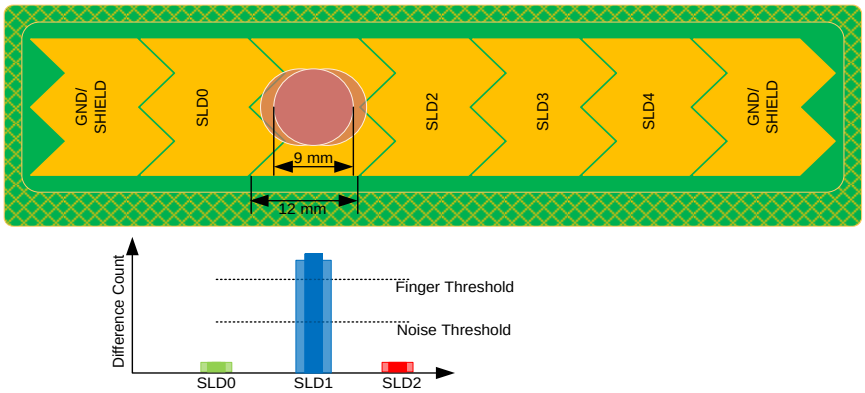


图 7-19. 滑条段宽度大于推荐值时滑条段上的信号



请注意，如果 *slider-segment-width + 2 * air-gap* 大于 *finger diameter*，则可能增大并调整所有滑条段的灵敏度，以便在手指触摸到滑条段的中间位置时，相邻传感器报告的计数差值仍能等于噪声阈值（如图 7-12 所示）；但是这样会引起悬浮效果，即虽然手指悬浮在滑条上并未触摸到滑条，滑条仍会报告中心位置。

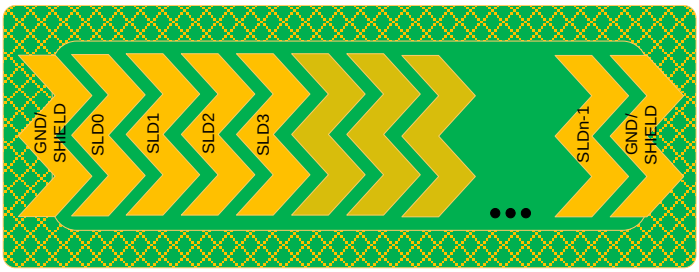
7.4.3.2 滑块两端的虚拟段

在 CapSense 设计中，当扫描某一段时，根据 CapSense CSD 组件中“Inactive sensor connection”参数所指定的选项，将相邻的段接地或连接至驱动屏蔽信号。对于线性中心响应，滑条要求所有段的灵敏度相同，即要求手指放置在滑条段上时，所有段的原始计数（信号）的增量都要相同。为保持所有滑条段信号具有统一的电平，建议将滑条两端的两个段物理接地，或连接至驱动屏蔽信号。选择接地还是连接至驱动屏蔽信号，取决于“Inactive sensor connection”参数所指定的值。因此，如果您的应用需要‘n’个段的滑条，应创建 n + 2 个段，如图 7-11 中所示。

如果因空间限制而无法在滑条两端上增加 2 个段，您可以利用顶层的填充网格来创建这些段，如图 7-20 所示。此外，如果可用的总空间仍受限制，则可将这些段的宽度设为小于段 SLD0 到 SLDn-1 间的宽度，甚至可以移除这些虚拟段。

如果滑条两端的 2 个段被连接到顶层填充网格，则应该将顶层填充网格连接至“Inactive sensor connection”参数所指定的信号。如果滑条需要具备防水功能，那么应该分别将围绕滑条的填充网格、最后两端以及未激活的滑条段连接到驱动屏蔽信号。更多有关信息，请参考水滴和水流对 CapSense 传感器产生的影响一节。

图 7-20. 第一个和最后一个滑条连接至顶层填充网格时的线性滑条图案



7.4.3.3 决定滑条尺寸

通过执行下列操作，可为特定设计选择滑条尺寸：

a. 根据应用要求决定滑条长度（L）。该长度和图 7-11 所示的“有效滑条长度”相同。
b. 根据电路板上的可用空间决定段高度。请使用所允许的最大段高度（15 mm）（若可行）；否则使用较小的高度，但要确保该高度大于表 7-3 中所指定的最小值。
c. 应该将滑条段宽度和各个滑条段之间的气隙设置为表 7-3 中所建议的值。对于 9 mm 大小的手指平均直径，推荐的滑条段宽度和气隙分别为 8 mm 和 0.5 mm。
d. 对于特定的滑条长度 *L*，通过下面公式可计算得出所需段数：

公式 7-3. 一个滑条所需的段数

$$Number\ of\ segments = \frac{slider\ length}{slider\ segment\ width + air\ gap} + 1$$

请注意，一个滑条最少需要使用两个滑条段实现。

如果 CapSense 的可用引脚略少于特定应用所计算的段数，那么您应该增大滑条段的宽度，从而获取所需滑条长度和可用的引脚数量。例如，10.2 cm 的滑条需要 13 个段。但是，如果只有 10 个引脚可用，则可能要将滑条段宽度增大到 10.6 cm。这样会引起非线性响应（如图 7-18 所示）或悬浮效果，但如果终端应用对线性要求不高，可以使用该布局。

请注意，PCB 长度大于所需要的滑条长度，如图 7-11 所示。PCB 长度可与滑条长度相关联，具体如下：

公式 7-4. PCB 的最小长度和滑条长度间的关系

$$PCB\ length = Slider\ Length + 3 * slider\ segment\ width + 2 * air\ gap$$

如果 PCB 的长度小于该公式要求的值，则可以移除虚拟段。

这时，PCB 所需的最小长度将为：

$$PCB\ length = Slider\ Length + Slider\ Segment\ Width$$

7.4.3.4 布置滑条段走线

一个滑条具有多个段，每个段被分别连接到器件的 CapSense 输入引脚。可以单独扫描每个段，并且所有段的信号值最终采用了中心算法来计算中心位置。SmartSense 算法通过实现滑条的特定调试方法，可以避免因各段的 C_P 值不同而导致的中心非线性。但是必须满足下面各个布局条件才能使滑条正常工作：

1. 所有滑条段的 C_P 值应始终位于 5 ~ 45 pF 的受支持范围内。
2. 该滑条中其他段的 C_P 值需要大于最大 C_P 值的 75%。例如，在一个滑条中，如果某一段的最大 C_P 值为 30 pF，则其他段的 Cp 值必须大于 22.5 pF。

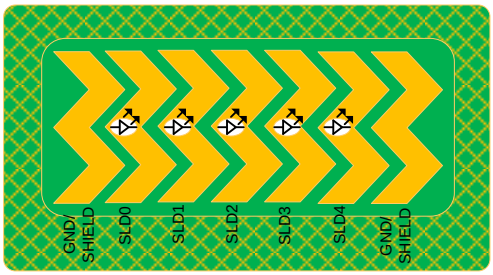
为了满足该条件，请实现下面的布局设计规则：

- 尽量为所有滑条段使用相同的形状。
- 对于所有滑条段，用于将各段连接至器件的各走线的长度和宽度要求相同。
- 保持传感器/走线和接地层/填充网格间的气隙宽度相同。

7.4.3.5 带有 LED 的滑条设计

在某些应用中，可能需要通过驱动 LED 来显示手指所在的位置。您可以将 LED 安装在滑条段上，也可以在某个滑条段中间位置钻一个孔用于背光显示 LED，如图 7-21 所示。钻好用于放置 LED 的孔时，会减少滑条段的有效面积。为实现 SNR > 5:1，需要使用一个宽度大于 LED 孔大小的滑条段。请参考表 7-3，了解针对特定覆盖层厚度的最小滑条宽度，以实现 SNR > 5:1。请按照串扰解决方案所提供的指南进行布置 LED 走线。

图 7-21. 带 LED 背光的滑条设计



7.4.4 传感器和器件放置

将传感器和器件放置在您的 PCB 设计中时，请按照下面的指南进行操作：

- 尽量减短器件引脚和传感器垫片间的走线长度。
- 在离器件引脚 10 mm 的范围内安装串联电阻，以降低射频干扰并提供 ESD 保护。请参考 CapSense 引脚上的串联电阻，了解详细信息。
- 将器件和其他组件安装在 PCB 的顶层上。
- 将传感器的开关信号（如 PWM、I2C 通信线路和 LED）和传感器走线隔离开。它们的最小间距必须为 4 mm。此外，应在 CapSense 走线和开关信号间填充网格地以避免串扰。
- LED 和 I2C 引脚等直流负载应尽可能通过完整端口与 CapSense 引脚物理隔离。例如，如果端口 P1 中有 LED 引脚，建议避免在同一端口上安装 CapSense 引脚。此外，CapSense 模块扫描传感器时，推荐将 GPIO 的源电流/灌电流限制在 40 mA 以内。如果在扫描 CapSense 传感器过程中，灌电流超过了 40 mA，则可能会使传感器的原始计数出现过大噪声。
- 避免在传感器和器件引脚间使用连接器，因为连接器会增加 CP 值和噪声。

7.4.5 走线长度和宽度

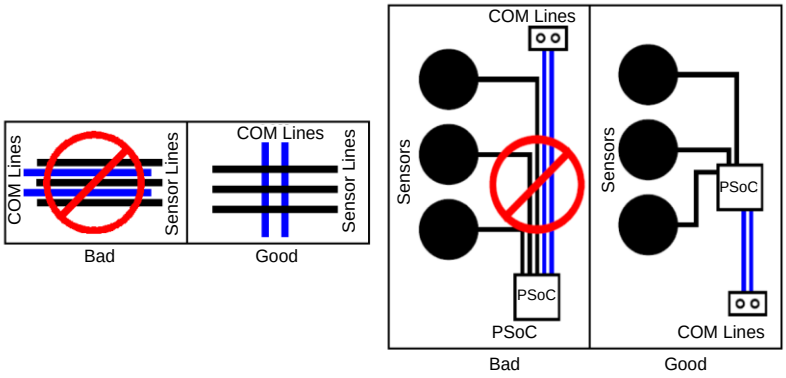
请使用既短又窄的 PCB 走线，以最小化传感器的寄生电容。对于标准的 PCB，建议使用的最大走线长度为 12 英寸（300 mm）；对于柔性电路，最大走线长度应为 2 英寸（50 mm）。走线的最大宽度应为 7 mil（0.18 mm）。应使用网格地或网格屏蔽围绕 CapSense 走线，走线与接地间的间距介于 10 mil 到 20 mil（0.25 mm 到 0.51 mm）之间。

7.4.6 走线路由

应在 PCB 的底层上路由传感器走线，以便手指不能与走线交互。不要将走线直接路由在任何传感器垫片的下方，除非走线是与该传感器连接的。

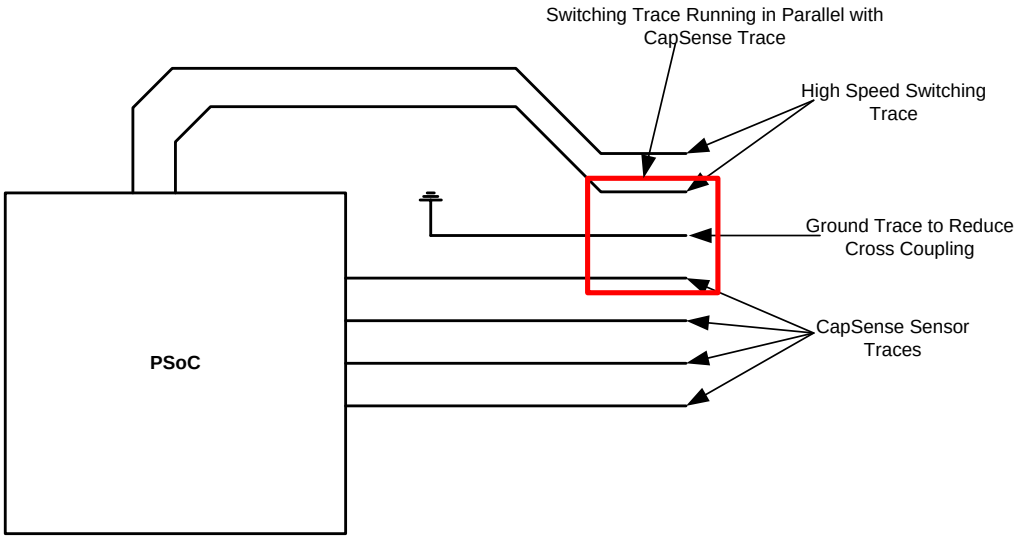
电容式感应走线与切换信号或通信线路之间的距离必须不低于 0.25 mm。通过加大感应走线和其他信号间的距离，可以增强抗噪能力。如果传感器引脚必须与通信线路交叉，确保交叉点呈直角，如图图 7-22 所示。

图 7-22. 传感器和通信线路的路由



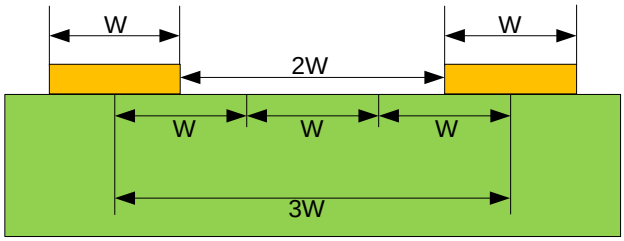
如果因距离有限而使传感器走线和高速开关走线（如 I²C 通信线或 BLE 天线走线）相互并行，则建议在传感器走线和高速走线之间放置一条接地线，如图 7-23 所示。如果想降低 CapSense 传感器走线和精度模拟走线（如从温度传感器到 PSoC 器件的走线）之间的串扰，也可以采用上述同样的方法。接地线的厚度可为 7 mil，传感器走线和接地线间的距离至少为 10 mil，以降低 Capsense 传感器的 C_P 值。

图 7-23. 降低高速开关走线和 CapSense 走线之间的串扰



如果在高速开关走线和 CapSense 走线间不能放置一个接地线，则可以遵循 3W 规则来降低它们之间的串扰。3W 规则指出：“要想降低相邻走线间的串扰，则两条线的边沿间的距离至少为走线宽度的两倍”，如图 7-24 所示。

图 7-24. 3W 走线的距离，用于最小化串扰

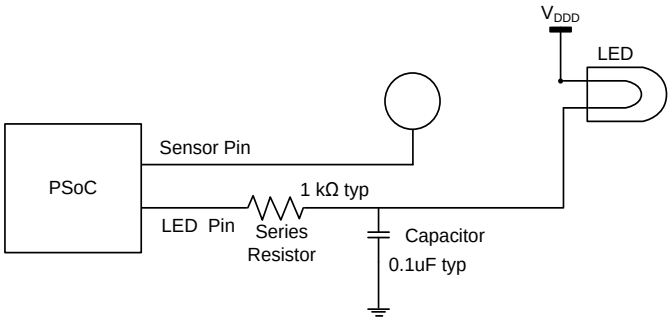


7.4.7 串扰解决方案

用于屏幕的一个通用背光技术是在传感器垫片下方安装一个 LED，这样可以通过传感器垫片中间的小孔看到该 LED。当 LED 打开或关闭时，LED 走线上的电压跳变会造成电容式传感器输入中的串扰，从而引起嘈杂的传感器数据。想要避免串扰现象，请将 CapSense 和 LED 走线隔离开，如第 6.3.7 节中所述。

另外，也可以通过利用滤波器去除 LED 驱动电压的快速转换来减少串扰，如图 7-25 所示。根据所需的 LED 响应速度设计该滤波器。

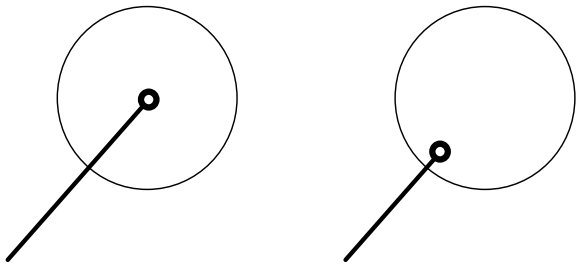
图 7-25. 降低串扰



7.4.8 过孔

在布局 CapSense 信号时使用最少的过孔数量，以尽量减少寄生电容。为了缩短走线长度，请将过孔放置在传感器垫片的边缘上，如图 7-26 所示。

图 7-26. 传感器垫片上的过孔位置



7.4.9 地层

设计接地层时，请按照下面的指南进行操作：

- 传感器周围的接地层应该是网格图案。如果 PCB 的顶层和底层上均使用了接地层或驱动屏蔽，顶层应占用 25%的网格（7 mil 的线路，45 mil 的间距），底层应占用 17%的网格（7 mil 的线路，70 mil 的间距）。
- 对于电路板上与 CapSense 无关的其他部分，尽可能使用实心接地层。
- 根据 PCB 的生产成本，尽量把不同层上的接地层互相拼接在一起。大量拼接各层可降低接地电感，并使芯片接地层更加靠近电源接地层。这样的设计非常重要，尤其是在大灌电流通过接地层（例如，在进行射频过程中）的情况下。
- 用于 CapSense 的所有接地层应按星形连接到中心点，并且该中心点应该是返回电源接地层的唯一路径。特别是：
  - 所有传感器的网格接地层终端都必须位于中心点
  - C_{MOD}、C_{INTX} 的接地层终端位于中心点
  - C_{SH_TANK} 的接地层终端必须位于中心点

图 7-27 显示的是星型连接。

表 7-4 显示的是不同系列的中心点。

图 7-27. 接地层的星形连接

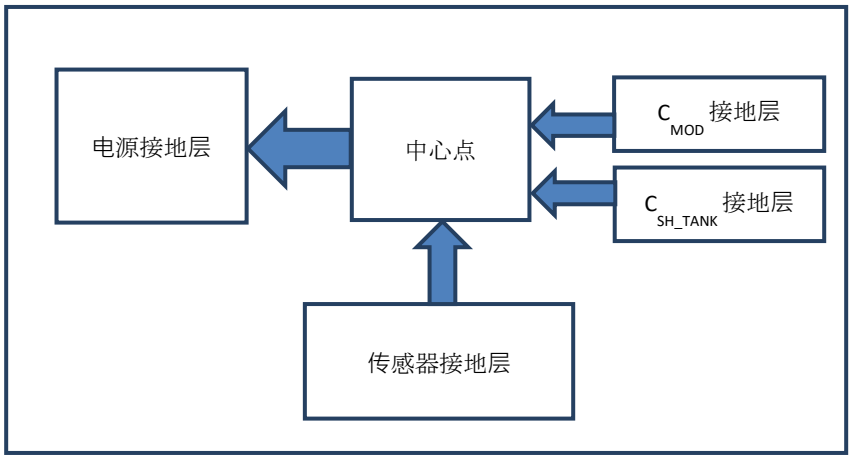


表 7-4. 星形连接的中心点

产品系列	中心点
PSoC 4000	VSS 引脚
PSoC 4100/4100M	VSS 引脚
PSoC 4200/4200M/4200L/PSoC 4-S/PSoC 4100PS	VSS 引脚
PSoC 4100-BL	E-pad
PSoC 4200 BL	E-pad

- CapSense 所有接地层的电感要比中心点的电感小 0.2 nH。要得到这样的电感，请将 C_{MOD}、C_{INTx} 和 C_{SH_TANK} 电容焊盘尽量放置在靠近芯片的地方，并使其接地层的厚度足够大。

7.4.9.1 使用不带 E-pad 的封装

如果未使用 E-pad，VSS 引脚应该为中心点和返回电源接地层的唯一路径。

图 7-28 和图 7-29 分别显示的是使用不带 E-pad 芯片时的电路板顶层和底层的高级布局框图。

.

图 7-28. 使用不带 E-pad 的芯片的 PCB 顶层布局

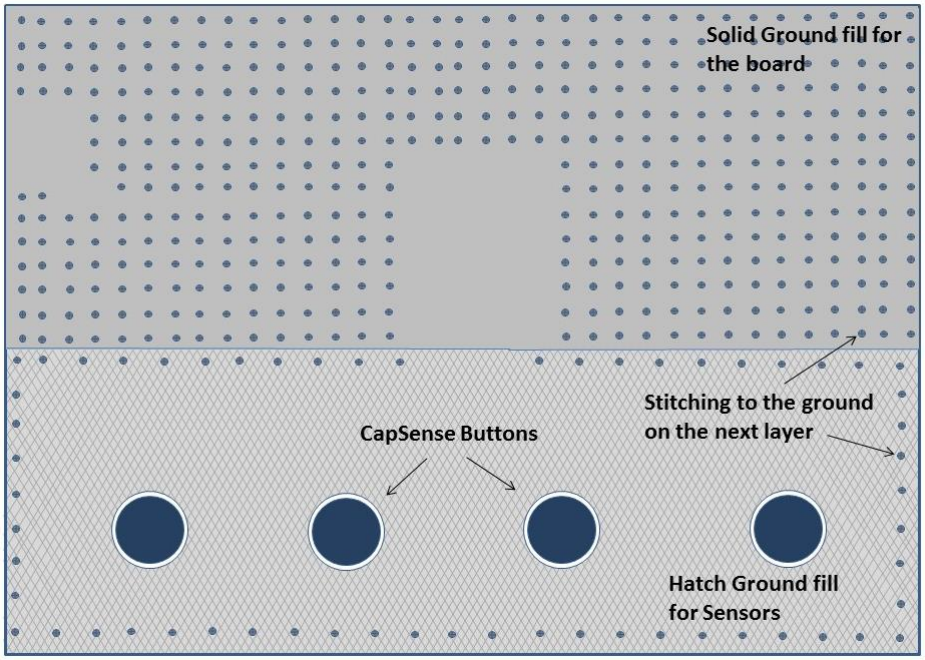
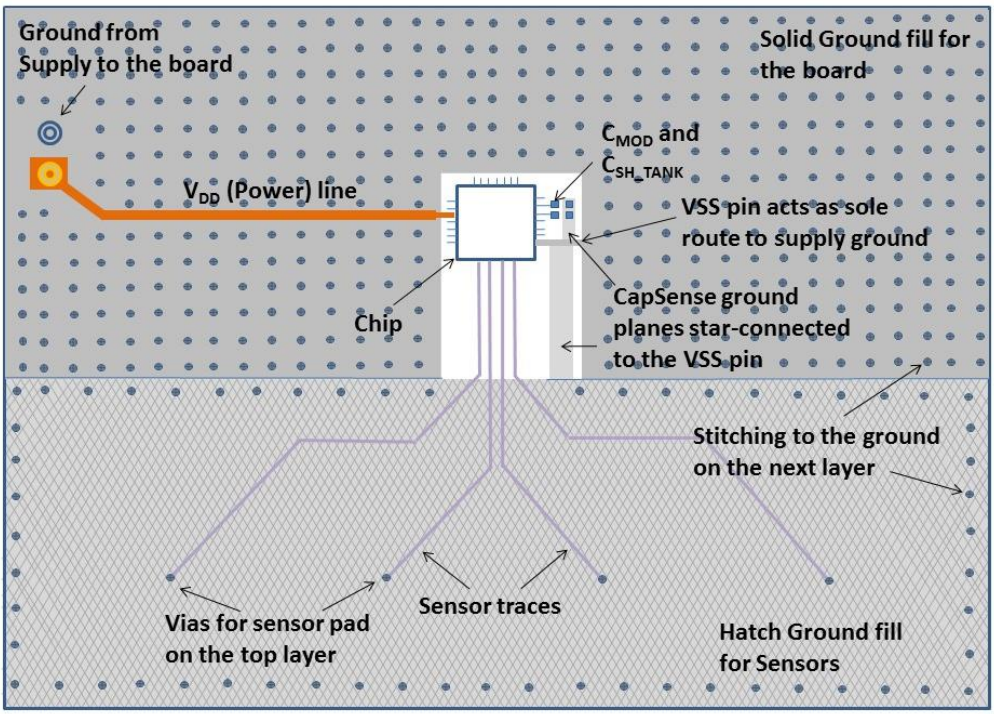


图 7-29.使用不带 E-pad 的芯片的 PCB 底层布局



7.4.9.2 使用带有 E-pad 的封装

如果您正在使用带有 E-pad 的封装，那么请按照下面的指南进行操作：

- E-pad 应该为质心点和返回电源接地层的唯一路径。
- E-pad 下面必须存在过孔，用于连接到下一层，从而为 E-pad 加地。考虑到成本问题，通常在设计中未填充过孔，但建议您使用含银的环氧树脂进行填充过孔，以得到最佳性能，因为这些过孔会使接地路径的电感最低。

7.4.9.3 使用 PSoC 4 BLE 芯片

带有 E-pad 的 QFN 封装中使用 PSoC 4 BLE 芯片时：

- 采用（上述）接地层的通用指南。
- 采用第 7.4.9.2 节中的 E-pad 使用指南。
- VSSA 引脚应在芯片下面连接至 E-pad。
- E-pad 下面的过孔建议采用 5 x 5（总共 25 个）矩阵的 10 mil 尺寸过孔形式。

图 7-30 和图 7-31 分别显示了使用 PSoC 4 BLE 或 PRoC BLE 芯片时的电路板顶层和底层的高级布局框图。

图 7-30. 使用 PSoC 4 BLE（带 E-pad）时的 PCB 顶层布局

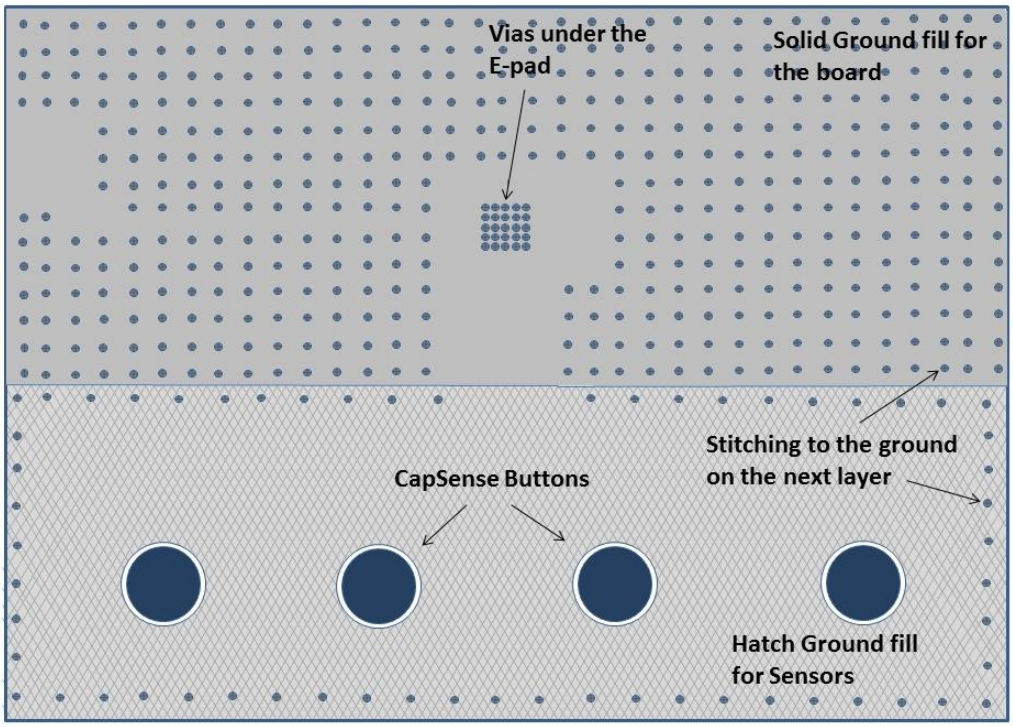
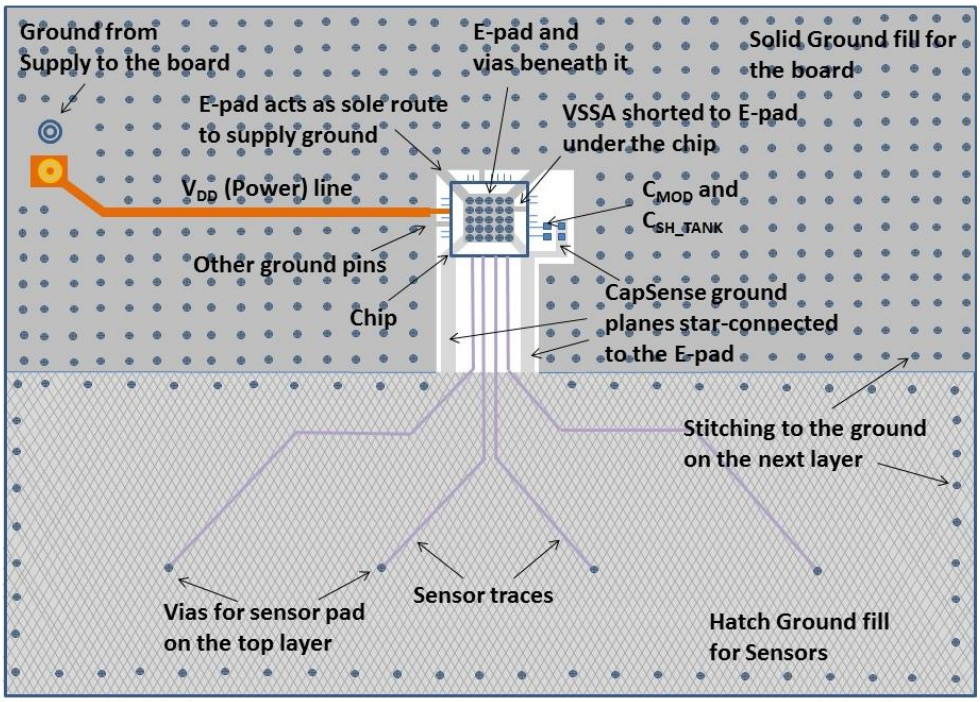


图 7-31. 使用 PSoC 4 BLE（带 E-pad）时的 PCB 底层布局



7.4.10 电源的布局建议

CapSense 是一种高灵敏度的模拟系统。因此，较差的 PCB 布局会使高灵敏度的传感器配置（如接近传感器和覆盖层厚度 > 1 mm 的按键）产生噪声。要想在高灵敏度的 CapSense 设计中降低噪声，应在 PCB 布局中的电源线上设置去耦电容，如表 7-5 所示。

表 7-5. 电源线上的去耦电容

电源线	去耦电容	相应的接地终端	可用的器件系列
VDD	0.1 µF 和 1 µF	VSS	PSoC 4000
VDDIO	0.1 µF 和 1 µF	VSS	PSoC 4000、PSoC 6 MCU
VDDD	0.1 µF 和 1 µF	VSS	PSoC 4100、PSoC 4200、PSoc 6 MCU
	0.1 µF 和 1 µF	VSSD	PSoC 4100-BL、PSoC 4200-BL、PSoC 4200 L、PSoC 4 S-Series、PSoC 4100S Plus
VDDA[9]	0.1 µF 和 1 µF（电池供电）	VSSA	PSoC 4100、PSoC 4200、PSoC 4100-BL、PSoC 4200-BL、PSoC 4200 L、PSoC 4 S-Series、PSoC 4100S Plus、PSoC 4100PS、PSoC 6 MCU
	0.1µF 和 10 µF（主电源供电）	VSSA	PSoC 4 S-Series、PSoC 4100S Plus、PSoC 4100PS
VDDR	0.1 µF 和 1 µF	VSSD	PSoC 4100-BL、PSoC 4200-BL、带有 BLE 连接功能的 PSoC 6 MCU
VCCD	请参见器件数据手册中介绍的内容。	VSS（PSoC 4000）或 VSSD（所有其他器件）	所有器件系列

必须将去耦电容和 C_{MOD} 电容尽放置在尽可能靠近芯片的地方，以尽量降低接地阻抗和缩短电源走线长度。

更多有关旁路电容的信息，请查看器数据手册中的“电源”章节。

7.4.11 耐水性布局指导

正如防水功能一节所述，通过使用一个屏蔽电极和保护传感器，可以实现 CapSense 系统的防水功能。本节介绍了屏蔽电极和保护传感器的使用方法。

屏蔽电极的面积取决于水滴大小以及电路板上可用于屏蔽电极的面积。屏蔽电极应围绕着传感器垫片和走线，并与它们的距离不超过 1 cm。如果该宽度超过 1 cm，将对系统性能产生微弱影响。

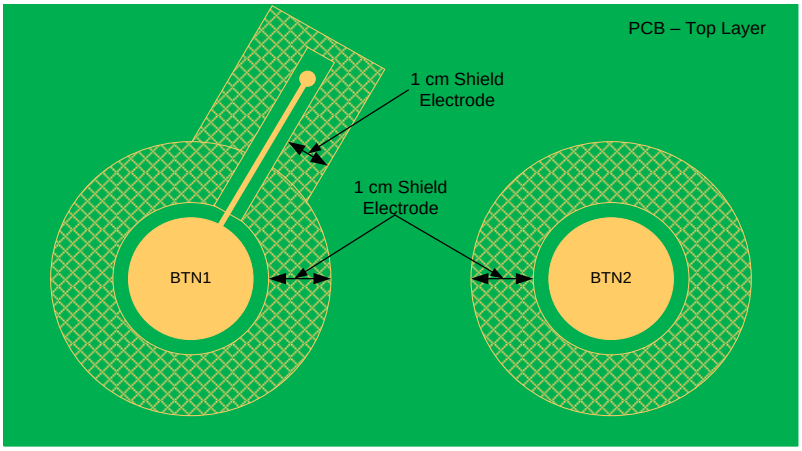
此外，使用更大的屏蔽电极会增大辐射。如果电路板尺寸较宽，则屏蔽电极范围 1 cm 外的区域应保留空白，如图 7-32 所示。为提高防水性能，在 PCB 的顶层和底层内不能有任何填充网格或走线接地。

如果存在接地网格填充或走线，当有水滴落在触摸表面时，会引起传感器误报。即使传感器和地面之间存在屏蔽电极，那么屏蔽电极的效果也会被完全屏蔽，反而会导致传感器误报。

在某些应用中，PCB 上没有足够的区域用于实现屏蔽电极。这时，屏蔽电极可以小于 1 cm；屏蔽电极的最小面积可以等于实现传感器后电路板上剩下的区域。

[9] 根据不同的供电电源，PSoC 4 S 系列和 PSoC 4100PS 系列中的 V_{DDA} 引脚需要不同的大容量电容。如果器件由电池供电，则建议使用并联的 0.1 µF 和 1 µF 电容；如果器件由主电源供电，则建议使用并联的 0.1 µF 和 10 µF 电容。这样做可以改善 CapSense 中使用的参考生成器（REFGEN）的电源抑制比。

图 7-32. 传感器走线被布线在顶层和底层时的屏蔽电极



要想在双层和四层 PCB 上实现屏蔽电极，请按照下面的指南进行操作：

对于双层 PCB：

- 顶层：网格填充，走线宽度为 7 mil，网格宽度为 45 mil（25%填充）。填充的网格应该与驱动屏蔽信号相连。
- 底层：网格填充，走线宽度为 7 mil，网格宽度为 70 mil（17%填充）。填充网格应该与驱动屏蔽信号相连。

对于四（或更多）层 PCB：

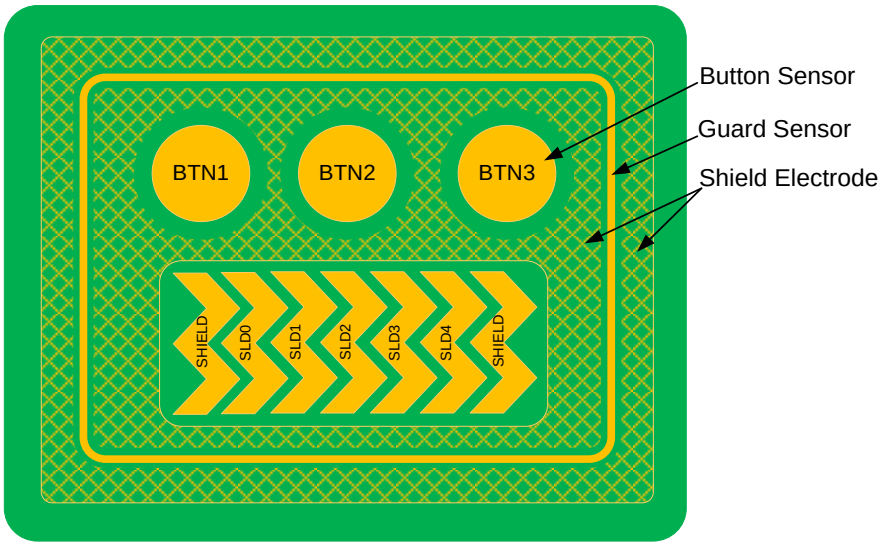
- 顶层：填充网格，走线宽度为 7 mil 和网格宽度为 45 mil（25%填充）。填充的网格应该与驱动屏蔽信号相连。
- 第二层：网格填充，走线宽度为 7 mil，网格宽度为 70 mil（17%填充）。填充的网格应该与驱动屏蔽信号相连。
- 第三层：VDD 层
- 底层：网格填充，走线宽度为 7 mil，网格宽度为 70 mil（17%填充）。填充的网格接地。

建议传感器和屏蔽电极间的气隙为 1 mm。

7.4.11.1 保护传感器

如保护传感器一节所述，保护传感器是围绕所有传感器的铜走线，如图 7-33 所示。

图 7-33. 使用屏蔽电极和保护传感器的 PCB 布局



仅在触摸表面上存在水流时才要触发保护传感器。应确保屏蔽电极格式围绕保护传感器，以避免它由存在水滴而被打开。请根据以下条件放置保护传感器：

- 当触摸表面上有水流时，它是第一个被打开的传感器。这样，通常使保护传感器围绕其他所有传感器。
- 当按下某个按键或滑条传感器时，应避免意外触摸该传感器。否则，将禁用按键传感器和滑条传感器的扫描过程，而且在关闭保护传感器前，CapSense 系统无法工作。为了确保不会意外触摸保护传感器，请保持它离其他传感器的距离至少为 1 cm。

请按照以下指南实现保护传感器：

- 保护传感器的形状应为弯曲边缘的矩形，另外保护传感器应围绕所有其他传感器。
- 保护传感器的建议厚度为 2 mm。
- 保护传感器和屏蔽电极间的建议间隙为 1 mm。

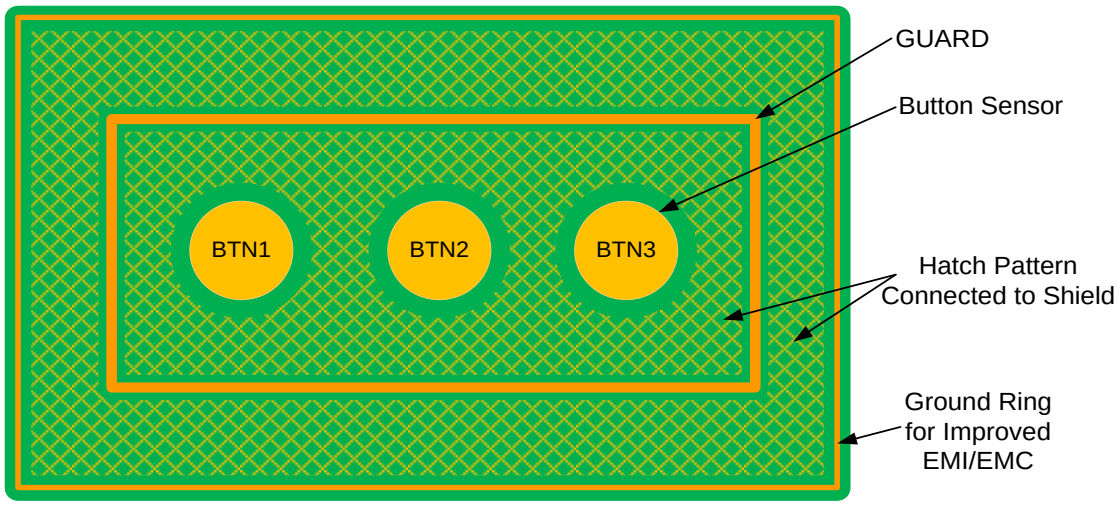
如果 PCB 上没有空间用于实现保护传感器，那么可以在固件中实现保护传感器的功能。例如，可以使用不同传感器的 ON/OFF 状态检测水流。通过使用以下各条件可检测触摸表面上的水流：

- 存在水流时，同时可激活多个按键传感器。如果您的设计不需要多点触摸感应性能，您可以检测它并抑制所有按键传感器的传感器状态，以防止发生错误触发。
- 在一个滑条中，如果被打开的不是相邻滑条段，那么您可以复位滑条段状态或抑制被计算的滑条中心值。

7.4.11.2 使用地环的防水性能

在某些应用中，电路板边缘需要一个地环（实心走线或网格填充）来提高 ESD 和 EMI/EMC 的性能，如图 7-34 所示。这样，当传感器和接地传感器之间存在水滴时，传感器会误报。因此，建议顶层上不要存在任何地环。如果该设计的顶层上要求存在地环，请使用厚度最小（8 mils）的地环。

图 7-34. 带有地环的 CapSense 设计可提高 ESD 和 EMI/EMC 性能



7.4.12 原理图规格检查表

您可通过所提供的检查表验证您的 CapSense 原理图。

表 7-6. 原理图规格检查表

序号	类别	推荐值/备注
1	C_{MOD}	大小为 2.2 nF。对于引脚的选择，请参考表 7-7 表 7-7。
2	C_{SH_TANK}	如果使用了屏蔽电极，大小为 10 nF。否则无需使用该电容。请分别参考驱动屏蔽信号和屏蔽电极和 CapSense CSD 屏蔽，了解有关屏蔽电极和 C_{SH_TANK} 使用情况的详细信息。 对于引脚的选择，请参考表 7-7。
3	C_{INTA}/C_{INTB}	大小为 470 pF。对于引脚的选择，请参考表 7-7 表 7-7。
3	输入线路上的串联电阻	自电容电阻为 560 Ω 以及互电容电阻为 2 kΩ。请参考 CapSense 引脚上的串联电阻，了解详细信息。
4	传感器引脚选择	如果可能，请避免使用接近带有开关/通信信号的 GPIO 的引脚。在一个端口中，请尽量将直流负载（如 LED 和 I²C 引脚）与 CapSense 引脚隔离开。请参考调校调试常见问题解答切换 GPIO 时，原始计数显示电平的转换或增加的噪声，以了解更多信息。 **注意**：对于 PSoC 6 系列器件，最佳的 CapSense 灵敏度和精度，请遵循相应器件数据手册（PSoC 61、PSoC 62 和 PSoC 63）的 Errata 一节中所述的建议，
5	GPIO 源/灌电流	要确保在 CapSense 模块正在对传感器进行扫描时，通过 GPIO 的总灌电流不大于 40 mA。

7.4.12.1 外部电容引脚选择

如 CapSense 基本原理一节所述，为了确保稳定操作，CapSense 需要外部电容 C_{MOD}（CSD 感应方法）、C_{TANK}（仅在实现屏蔽时）以及 C_{INTX}（CSX 感应方法）。从 PSoC Creator 3.3 SP2 开始，支持 C_{MOD} 和 C_{SH_TANK} 的引脚数量得到增加，用于改善设计的灵活性。表 7-7 表 7-7 列出了用于 PSoC Creator 3.3 SP2 或更高版本的 C_{MOD}、C_{INTX} 和 C_{SH_TANK} 电容的推荐引脚。

注意：在 PSoC 4100/PSoC 4200 器件中，如果为 C_{MOD} 选择了其他引脚（不是 P4[2]），则 P4[2]将无法实现其他功能。例如，在 PSoC 4200 器件中，如果您尝试将 C_{MOD} 布线到 PSoC Creator 的 P2[0]引脚上，那么它将使用 P2[0]和 P4[2]。

表 7-7. 外部电容的建议引脚（PSoC Creator 3.3 SP1 或更低版本）

器件	C_{MOD}	C_{SH_TANK}
PSoC 4000	P0[4]	P0[2]
PSoC 4100/ PSoC 4200	P4[2]	P4[3]
PSoC 4200M/ PSoC 4200L	CSD0: P4[2]	CSD0: P4[3]
	CSD1: P5[0]	CSD1：P5[1]
PSoC 4 BLE	P4[0]	P4[1]

表 7-8. 外部电容的建议引脚（PSoC Creator 3.3 SP2 或更高版本）

器件	C_{MOD}	C_{SH_TANK}	C_{INTA}	C_{INTB}
PSoC 4000	Port0[0:7]、Port1 [0:7] P2[0]	Port0 [0:7]、Port1 [0:7] P2[0]	P0[4]	P0[2]
PSoC 4100	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] P4[2]	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] P4[3]	不支持	不支持
PSoC 4200	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] P4[2]	Port0 [0:7]、Port1 [0:7]、 Port2 [0:7]、Port3 [0:7] P4[3]	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7]	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7]
PSoC 4200M	CSD0： Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] Port4 [0:6]、Port6 [0:5] Port7 [0:1]	CSD0： Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7]、 Port4 [0:6]、Port6 [0:5] Port7 [0:1]	CSD0：P4[2]	CSD0：P4[3]
	CSD1：不受支持	CSD1： 不受支持	CSD1： 不受支持	CSD1： 不受支持
PSoC 4200L	CSD0： Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] Port4 [0:6]、Port6 [0:5] Port7 [0:7]、Port10 [0:7] Port11 [0:7]	CSD0： Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] Port4 [0:6]、Port6 [0:5] Port7 [0:7]、Port10 [0:7] Port11 [0:7]	CSD0：P4[2]	CSD0：P4[3]
	CSD1： Port5 [0:7]、Port8 [0:7] Port9 [0:7]	CSD1： Port5 [0:7]、Port8 [0:7] Port9 [0:7]	CSD1：P5[0]	CSD1：P5[1]
PSoC 4 BLE	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] Port4 [0:1]、Port5 [0:1] Port6 [0:1]	Port0 [0:7]、Port1 [0:7] Port2 [0:7]、Port3 [0:7] Port4 [0:1]、Port5 [0:1] Port6 [0:1]	P4[0]	P4[1]
PSoC 6 MCU	P7[1]或 P7[2]或 P7[7]	P7[1]或 P7[2]或 P7[7]	P7[1]	P7[2]
PSoC 4 S-Series、PSoC 4100S Plus	P4[2]、P4[3]、P4[1]	P4[2]、P4[3]、P4[1]	P4[2]	P4[3]
PSoC 4100PS	P5[0]、P5[2]、P5[3]	P5[0]、P5[2]、P5[3]	P5[2]	P5[3]

7.4.13 布局规格检查表

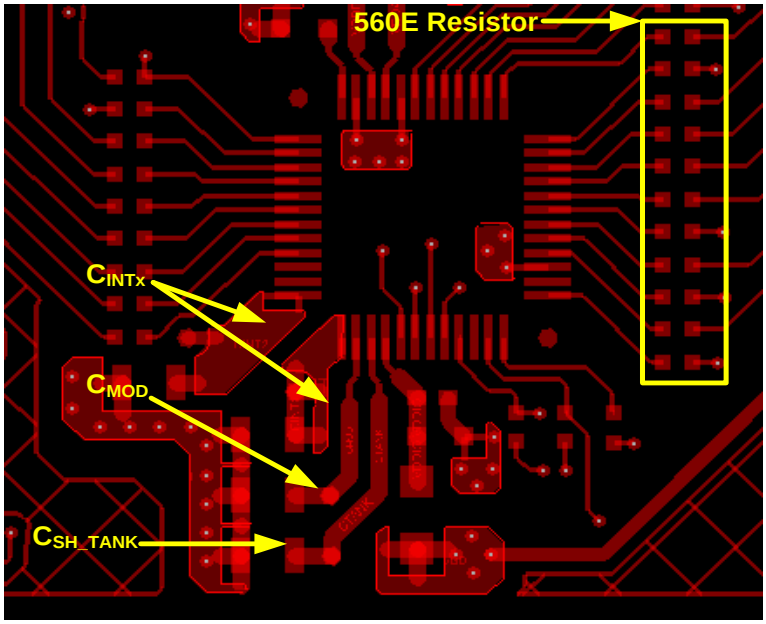
您可通过使用表 7-9 中提供的检查表验证您的布局设计。

表 7-9. 布局规格检查表

序号	类别		最小值 数值	最大值	推荐 /备注
1	按键	形状	N/A	N/A	弯曲边缘的圆形和矩形
		大小	5 mm	15 mm	10 mm
		与接地网格的间隙	0.5 mm	2 mm	应等于覆盖层厚度
2	滑条	滑条段宽度	1.5 mm	8 mm	8 mm
		段之间的间隙	0.5 mm	2 mm	0.5 mm
		滑条段的高度	7 mm	15 mm	12 mm
3	覆盖层	类型	N/A	N/A	具有较高相对介电常数的材料（导电材料除外） 去除传感器电路板和外壳的覆盖层 / 前版间的空气间隙。
		按键厚度	N/A	5 mm	
		滑条厚度	N/A	5 mm	
		触摸板厚度	N/A	0.5 mm	
4	传感器走线	宽度	N/A	7 mil	使用符合当前使用的 PCB 板技术的最小宽度。
		长度	N/A	对于标准的（FR4）PCB 板，长度为 300 mm 对于柔性 PCB 板，长度为 50 mm	该值越小越好
		与地面和其他走线的间隙	0.25 mm	N/A	使用最大的间隙，同时尽量缩短走线长度
		布线	N/A	N/A	请在传感器层的反面上进行布线。该走线与其他走线隔离开。如将非 CapSense 走线穿过 CapSense 走线，则应确保其垂直相交。走线的转向角不可为锐角。
5	过孔	过孔数	1	2	至少需要通过一个过孔在传感器层的背面路由走线
		过孔大小	N/A	N/A	10 mil
6	接地	网格填充百分比	N/A	N/A	铺设网格状的地覆铜，以降低寄生电容。通用网格： 顶层上铺设 25%的网格（7 mil 的线路，45 mil 的空间） 底层上铺设 17%的网格（7 mil 的线路，70 mil 的空间）
7	串联电阻	连接布置	N/A	N/A	将电阻设置在离 PSoC 引脚 10 mm 的范围内。 请参考图 7-35，了解有关在电路板上放置串联电阻的示例。
8	屏蔽电极	间距	N/A	1 cm	如果 PCB 板空间不受限制，请留出 1 cm 的间距。
9	保护传感器（用于防水功能）	形状	N/A	N/A	弯曲边缘的矩形
		厚度	N/A	N/A	建议保护走线的厚度和保护走线与屏蔽电极间的距离分别为 2 mm 和 1 mm。

序号	类别		最小值 数值	最大值	推荐 /备注
10	C_{MOD}	放置	N/A	N/A	放置在靠近 PSoC 引脚的位置。请参考图 7-35，了解有关在 PCB 板上放置 C_{MOD} 的示例。
11	C_{SH_TANK}	放置	N/A	N/A	放置在靠近 PSoC 引脚的位置。请参考图 7-35，了解有关在电路板上放置 C_{SH_TANK} 的示例。
12	C_{INTA}	放置	N/A	N/A	放置在靠近 PSoC 引脚的位置。请参考图 7-35，了解有关在 PCB 板上放置 C_{INTA} 的示例。
13	C_{INTB}	放置	N/A	N/A	放置在靠近 PSoC 引脚的位置。请参考图 7-35，了解有关在 PCB 板上放置 C_{INTA} 的示例。

图 7-35. 将 C_{MOD}、C_{INTx}、C_{SH_TANK} 以及串联电阻放置在 PSoC 4200M 器件中的输入线上的示例



7.5 ESD 保护

CapSense 中使用的非导电覆盖层材料提供了固定的保护功能，用于阻止 ESD。表 7-10 列出了各种覆盖层材料的厚度，用以避免 CapSense 传感器受 12 kV 放电的影响（根据 IEC 61000 - 4 - 2 规范）。

表 7-10. ESD 保护的覆盖层厚度

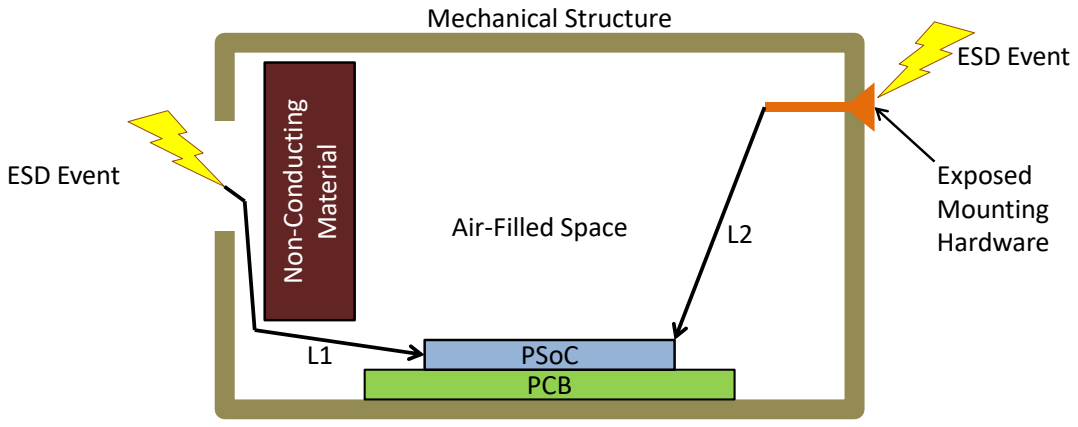
材料	击穿电压（V/mm）	用于避免受 12 kV ESD 影响的最小覆盖层厚度（mm）
空气	120 0 – 2800	10
干木材	3900	3
普通玻璃	7900	1.5
玻璃 — 硼硅酸盐（Pyrex®）	13,000	0.9
PMMA 塑料（Plexiglas®）	13,000	0.9
ABS	16,000	0.8
聚碳酸酯（Lexan®）	16,000	0.8
福米卡	18,000	0.7
FR-4	28,000	0.4
PET 薄膜（Mylar®）	280,000	0.04
聚酰亚胺薄膜（Kapton®）	290,000	0.04

如果覆盖层材料未提供全面保护（如在其他方向发生静电放电（ESD）），您可以按以下优先顺序采取防静电措施：预防、重定向、ESD 保护器件。

7.5.1 防止 ESD 放电

最好的对策是防止 ESD 放电回路波及 PSoC。请确保进入 PSoC 的所有路径的击穿电压都大于设备表面上最大的 ESD 电压。您还需要在 PSoC 和可能的 ESD 源间保持适当的距离。在图 7-36 所显示的示例中，如果 L1 和 L2 大于 10 mm，该系统可承受 12 kV ESD。

图 7-36. ESD 路径

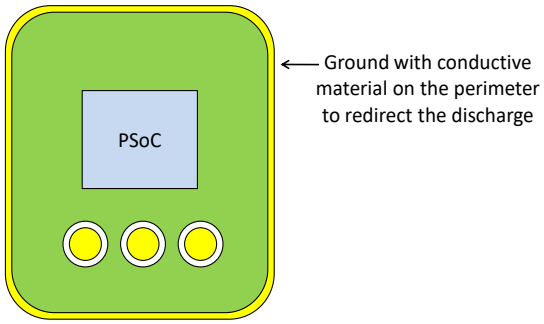


如果不能保持适当的距离，请在可能的 ESD 源和 PSoC 间放置一个具有高击穿电压的绝缘材料保护层。一层厚度为 5 mil 的 Kapton®胶带可承受 18 kV 的电压。有关其他材料的绝缘强度，请参见表 7-10。

7.5.2 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，您可通过重定向 ESD 来阻止它对 PSoC 产生影响。标准做法是在电路板周边放置一个地环，如图 7-37 所示。应将该地环连接至机盘地线。通过使用网格接地层围绕按键或滑条传感器，也可以阻止 ESD 事件对传感器和 PSoC 产生影响。

图 7-37. 地环



7.5.3 ESD 保护器件

您可在易受攻击的走线上使用 ESD 保护器件。应选择低输入电容的 ESD 保护器件，以避免 CapSense 灵敏度被降低。表 7-11 列出了所推荐的 ESD 保护器件。

表 7-11. ESD 保护器件

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

7.6 电磁兼容性（EMC）注意事项

EMC 与电磁能量的生成、发送和接收相关联，从而影响到电子系统的运行。电子设备需要符合特定的限制，以满足发出能源和易受外部事件的要求。一些监管机构设定了全球的地区性规格，以确保电子设备互不干扰。

CMOS 模拟和数字电路的输入阻抗很高。它们对外部电场非常敏感。因此，您需要特别注意，以确保这些电路在辐射和导电噪声存在的情况下仍可以正常工作。

在美国，FCC 根据 B 子部分中的第 15 节来调节计算器件，以阻止意外的辐射源。而欧洲和其他地区则遵循 CENELEC 标准。对于发射，将遵循 CISPR 标准（双标记为 ENxxxx 标准）；对于抗干扰和安全能力，则遵循 IEC 标准（双标记为 ENxxxx 标准）。

计算器件将根据通用发射规范，即 EN55022。该规范包含了辐射和导电发射的相关标准。在美国，医疗设备并非由 FCC 调节，而是受 FDA 规格限制。该规格包含了 EN55011 要求（即欧洲医疗设备规格）。具有电机控制的设备要遵循 EN55014 标准，而照明设备要满足 EN50015 标准。

这些规范对辐射和导电发生均有相同的关键性能限制。EN 61000-4 中的某些章节指定了抗辐射和抗传导干扰（抗扰）的性能要求。该标准还包含了线路电压跳变、静电放电 (ESD) 和某些安全问题。

7.6.1 辐射干扰及辐射

当 PSoC 4 和 PRoC BLE 提供强大的 CapSense 性能时，辐射电能会影响系统的测量，并可能影响 CapSense 处理器核心的运作。干扰会通过传感器走线，并通过其他的数字和模拟输入进入 PCB 级的 CapSense 芯片。CapSense 设备还发出可导致电磁兼容性（EMC）问题的辐射。

请使用以下技术尽量减少辐射干扰。

7.6.1.1 硬件的注意事项

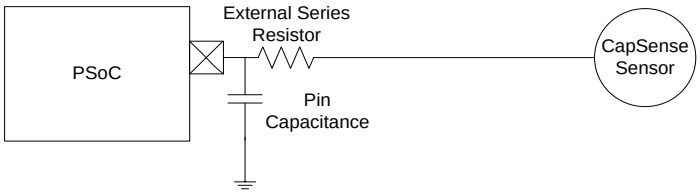
7.6.1.1.1 接地层

通常情况下，在 PCB 板上正确放置接地层会降低 RF 辐射和干扰。但是，接近 CapSense 传感器或接近用于将这些传感器连接至 PSoC 引脚的走线的实心地层会增大传感器的寄生电容。因此，建议在传感器周围以及传感器下方的 PCB 板底层上使用网格地层，如 PCB 布局指南中地层一节所介绍的内容。而在器件下方和 PCB 板上远离 CapSense 传感器和走线的其他电路部分下方应使用实心地层。不建议在 CapSense 传感器或走线的 1 cm 范围内使用实心地层。

7.6.1.1.2 CapSense 引脚上的串联电阻

每个 CapSense 控制器引脚都有一些与其关联的寄生电容 C_P。如图 7-38 所示，添加一个外部电阻可构成一个低通 RC 滤波器，用于减少耦合到引脚的 RF 噪声振幅。该电阻和 CapSense 传感器的寄生电容构成了低通滤波器，能够明显降低 RF 辐射。

图 7-38. RC 滤波器



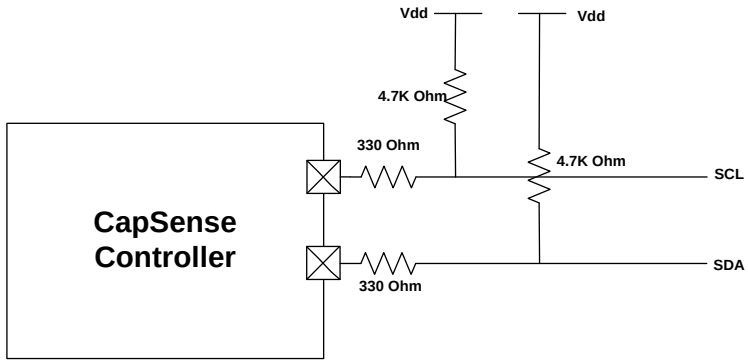
应将串联电阻置于靠近器件引脚的地方，以便能够在器件的输入端过滤掉走线所引起的辐射噪声。因此，建议在离引脚 10 mm 的范围内放置串联电阻。

对于在 PCB 上使用覆铜的 CapSense 设计，建议放置在 CapSense 输入线路上的串联电阻值为 560 Ω。添加该电阻会增加开关电容电路（将 C_P 转换成相应电阻的电路）的时间常量；请查看 GPIO 单元中电容-电流转换器。如果串联电阻值大于 560 Ω，开关电路的较低时间常量可抑制辐射和其干扰，但也会限制可传输的电荷量。这样会降低信号等级，进而降低信噪比（SNR）。小的电阻值有利于信噪比，但在阻止射频方面却不太有效。

7.6.1.1.3 数字通信线路上的串联电阻

通信线路，如 I²C 和 SPI，同样能受益于串联电阻，并且建议通信线路采用 330 Ω 的串联电阻。通信线路的走线较长，它们同 CapSense 走线一样，可作为天线使用。推荐在 I²C 通信线路上采用一个大小为 4.7 kΩ 的上拉电阻。因此，如果在这些线路上串联一个超过 330 Ω 的电阻，则 V_{IL} 和 V_{IH} 电压会超出要求。由于当 PSoC 输出为低电平时 V_{IL} 电平仍会保持为 I²C 规范所要求的 0.3 V_{DD}，因此 330 Ω 的电阻不会影响 I²C 的操作。

图 7-39. 通信线路上的串联电阻



7.6.1.1.4 走线长度

长走线比短走线引起的噪声更多。长走线还会增加 C_P 值。所以要尽量缩短走线长度。

7.6.1.1.5 电流环路面积

另一个重要的布局注意事项是，尽量缩短电流返回路径，因为电流是在环路中流通的。除非高速信号具有正确的返回路径，否则返回电流会通过一个较长的回路流通，从而形成一个较宽的环路，这样会增加辐射和干扰。

如果将 CapSense 地网格和围绕器件的地填充隔离开，传感器开关电流可能需要较长的返回路径，如图 7-40 所示。由于 CapSense 传感器以高频率切换，返回电流可能造成严重的 EMC 问题。因此，您应该使用单一接地网格，如图 7-41 所示。

图 7-40. 不合理的电流环路布局

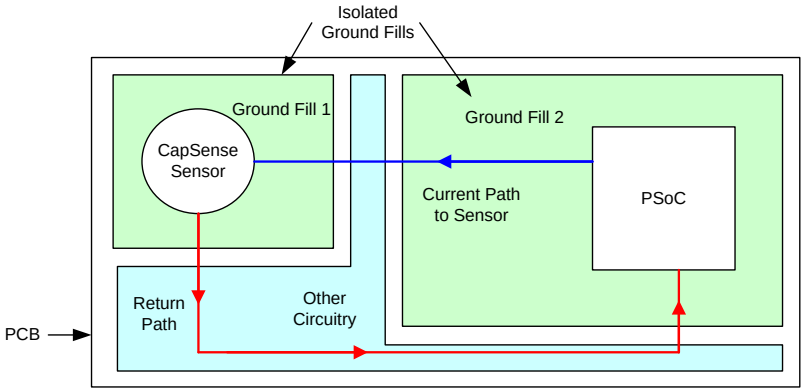
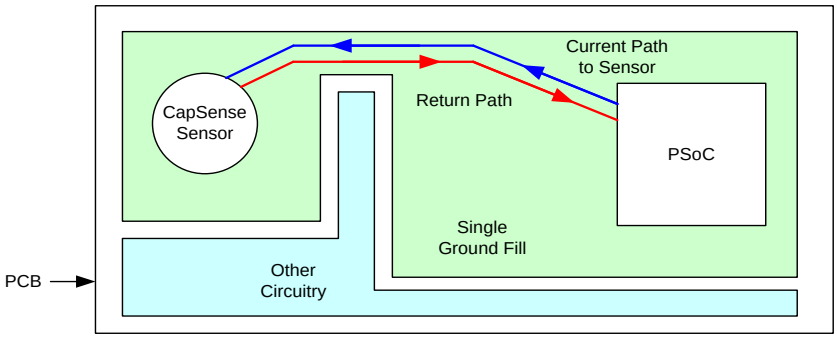


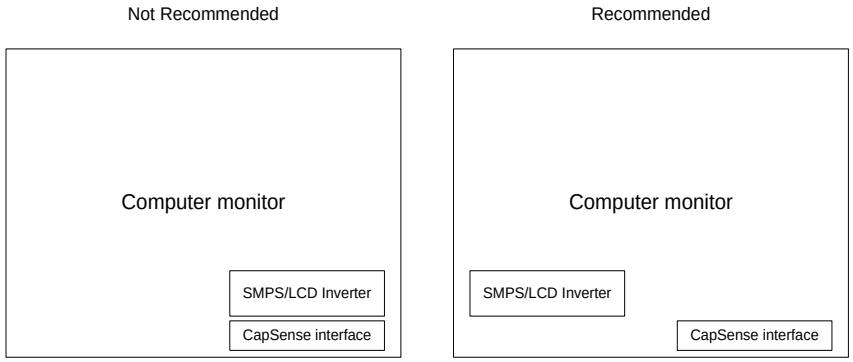
图 7-41. 合理的电流环路布局



7.6.1.1.6 射频源位置

如果您的系统有一个生成射频噪声的电路，如开关模式电源供应（SMPS）或一个变频器，您应该使这些电路远离 CapSense 接口。此外，也应该屏蔽这些电路，以降低射频。图 7-42 显示的是有关如何将射频噪声源和 CapSense 接口隔离开的一个示例。

图 7-42. 分离噪声源



7.6.1.2 固件的注意事项

在 CapSense 系统中下面各参数都会影响辐射（RE）：

- 器件的工作电压
- 器件的工作频率
- 传感器开关频率
- 屏蔽信号
- 传感器扫描时间
- 感应时钟源未激活的传感器终端

下面各节介绍了每个参数的影响。

7.6.1.2.1 器件的工作电压

辐射直接与发生切换的电压电平成比例。由于在任何输出引脚上切换信号的强度都直接取决于器件的工作电压，因此通过降低工作电压可以降低辐射。

PSoC 允许在较低的电压下进行操作，从而降低辐射。图 7-43 和图 7-44 显示的是工作电压对辐射产生的影响。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

图 7-43. V_{DD} 对辐射的影响（150 kHz – 30 MHz）

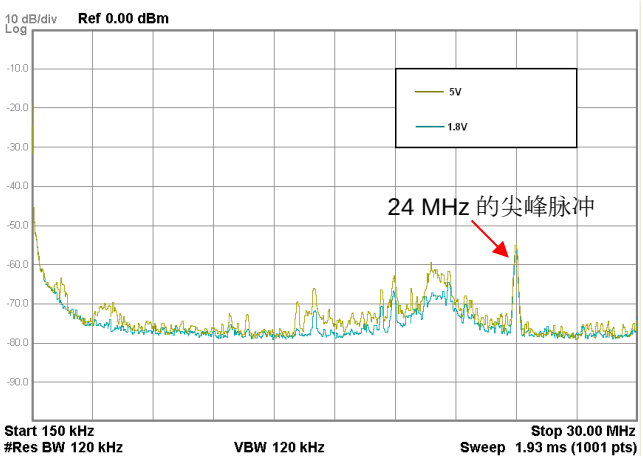
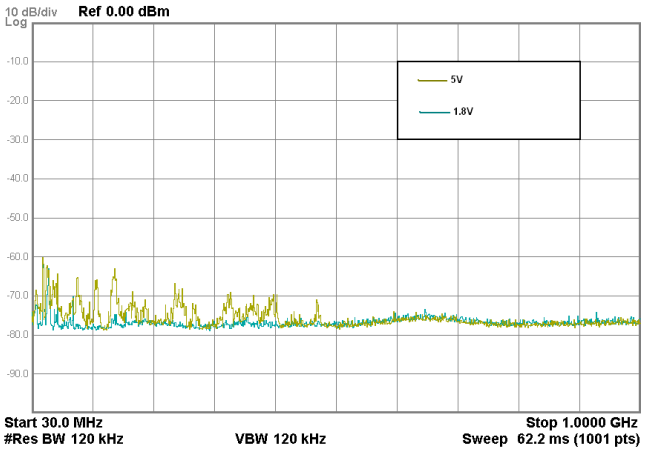


图 7-44. VDD 对辐射的影响（30 MHz – 1 GHz）



注意：频率轴上显示的是对数刻度值。

7.6.1.2.2 器件的工作频率

通过降低系统时钟频率（IMO 频率），可以降低辐射。但由于 IMO 频率会影响 CPU 时钟和所有其他系统时序，因此无法在所有应用中降低 IMO 频率。根据您的应用选择合适的 IMO 频率。

7.6.1.2.3 传感器开关频率

通过降低传感器的开关频率（请参考感应时钟一节）也可以降低辐射。请参考图 7-45 和图 7-46。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

图 7-45. 传感器开关频率对辐射的影响（150 kHz – 30 MHz）

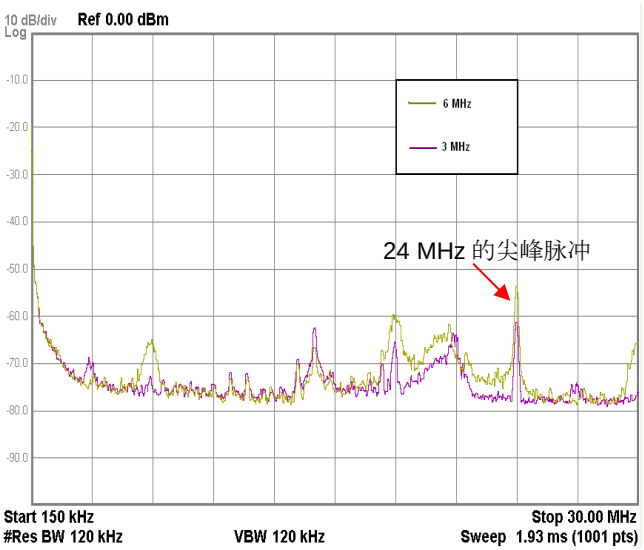
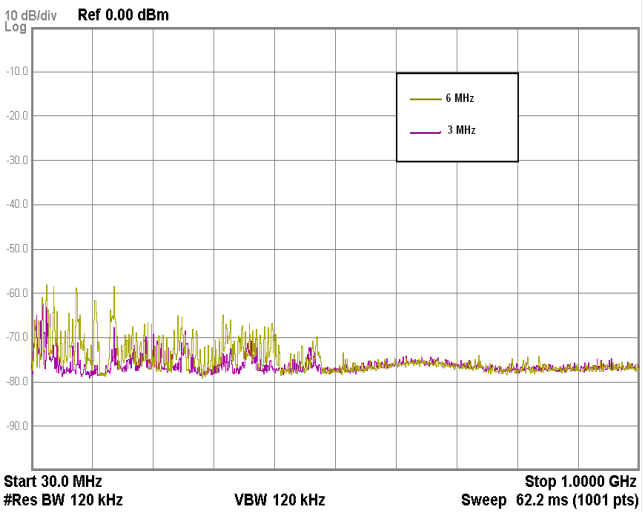


图 7-46. 传感器开关频率对辐射的影响（30 MHz – 1 GHz）



注意：频率轴上显示的是对数刻度值。

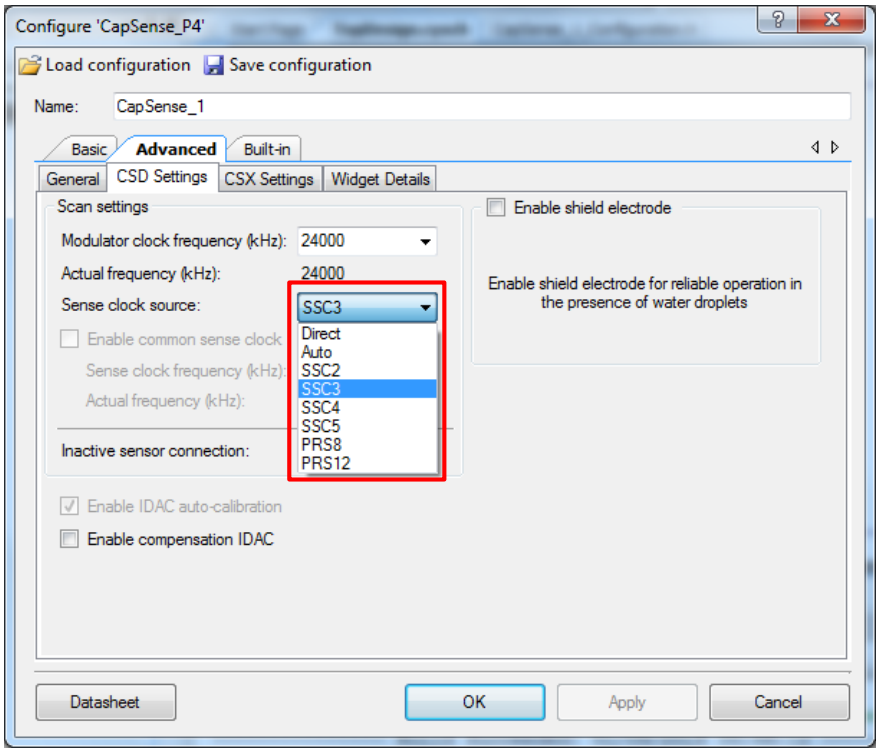
7.6.1.2.3.1 伪随机感应时钟

PSoC 4 器件支持生成基于 PRS 的感应时钟。使用 PRS，而不使用固定的时钟源；原因是通过使用 PRS 可以减少由固定频率源引起的 EMI，从而可以降低 CapSense 引脚上的噪声，并提高对其他频率源和它们谐波的抗 EMI 能力。

7.6.1.2.3.2 扩频感应时钟

除了生成基于 PRS 的时钟外，PSoC 4 S 系列、PSoC 4100S Plus、PSoC 4100PS 和 PSoC6 MCU 系列还支持一个叫做扩频感应时钟生成的独特功能，其中感应时钟频率被扩频到所需的范围内。该方法有助于降低峰值，并实现在某个频率范围内扩展辐射。通过将**感应时钟源**选为 **SSCn**，可以使能扩频时钟。频率扩展范围由寄存器的长度决定。更多有关在 PSoC 4 S 系列、PSoC 4100S Plus 和 PSoC 4100PS 系列中生成扩频时钟的信息，请参考相应器件技术参考手册，找到 CapSense 章节中扩频时钟一节的内容。

图 7-47. PSoC 4 S 系列、PSoC 4100S Plus 和 PSoC 4100PS 系列中的感应时钟源



7.6.1.2.4 屏蔽信号

使能网格图案上的屏蔽信号（请参考驱动屏蔽信号和屏蔽电极）会增大辐射。因此，仅在需要防水、接近感应功能或高寄生电容的设计中，才使能驱动屏蔽信号。此外，如果使用了屏蔽功能，那么请确保屏蔽电极面积与传感器的距离在 1 cm 范围内，如图 7-32 所示。

图 7-48 和图 7-49 显示的是使能传感器周围填充网格图案上的驱动屏蔽信号对辐射的影响。请注意，在这些图中，禁用驱动屏蔽信号时会将填充网格图案接地。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

图 7-48. 屏蔽电极对辐射的影响（150 kHz – 30 MHz）

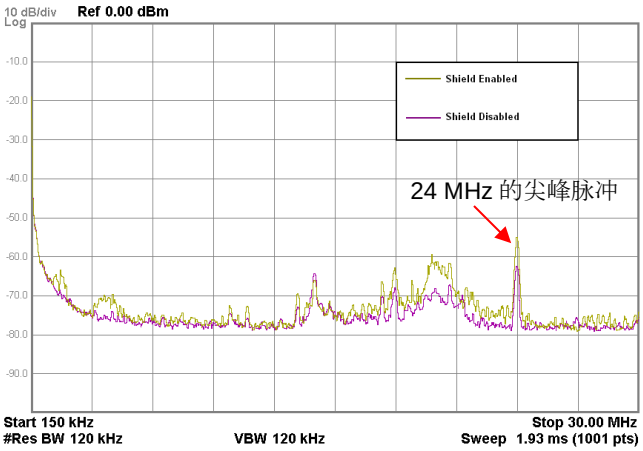
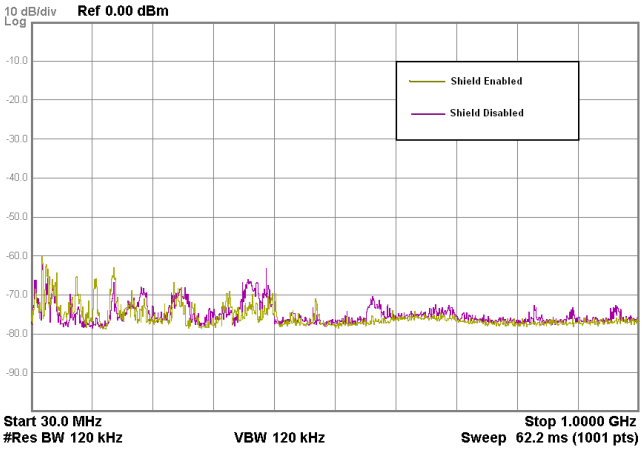


图 7-49. 屏蔽电极对辐射的影响（30 MHz – 1 GHz）



注意：频率轴上显示的是对数刻度值。

7.6.1.2.5 传感器扫描时间

降低传感器的扫描时间也能减少平均辐射。传感器的扫描时间取决于扫描分辨率和调制器时钟分频值（请参见公式 3-5）。增大扫描分辨率或调制器时钟分频值会延长扫描时间。图 7-50 和图 7-51 显示的是传感器扫描时间对辐射产生的影响。请注意，这里通过更改扫描分辨率可更改传感器扫描时间。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

表 7-12. 传感器扫描时间

参数	5 个按键的总扫描时间	
	0.426 ms	0.106 ms
调制器时钟分频值	2	2
扫描分辨率	10 位	8 位
单个传感器的扫描时间	0.085 ms	0.021 ms

图 7-50. 扫描时间对辐射的影响（150 kHz – 30 MHz）

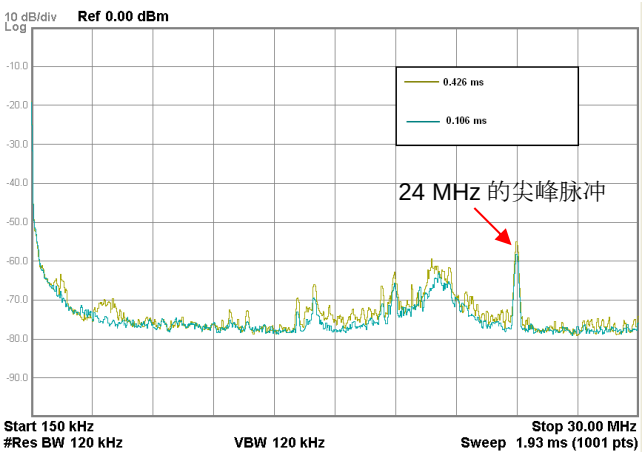
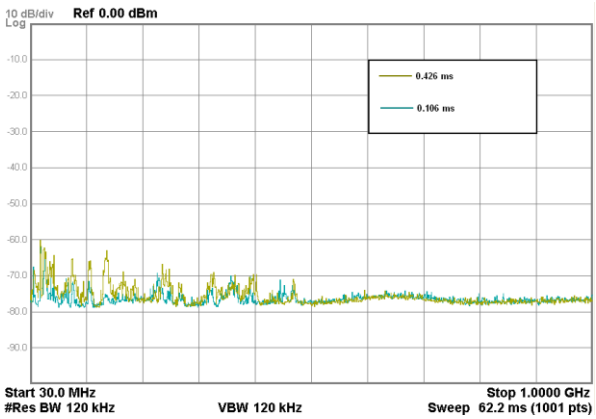


图 7-51. 扫描时间对辐射的影响（150 kHz – 30 MHz）



注意：频率轴上显示的是对数刻度值。

7.6.1.2.6 感应时钟源

使用 PRS 作为感应时钟源，而没有使用直接时钟驱动；这样做可以扩大辐射频谱，从而减少辐射。请参考图 7-52 和图 7-53。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

图 7-52. 感应时钟源对辐射的影响（150 kHz – 30 MHz）

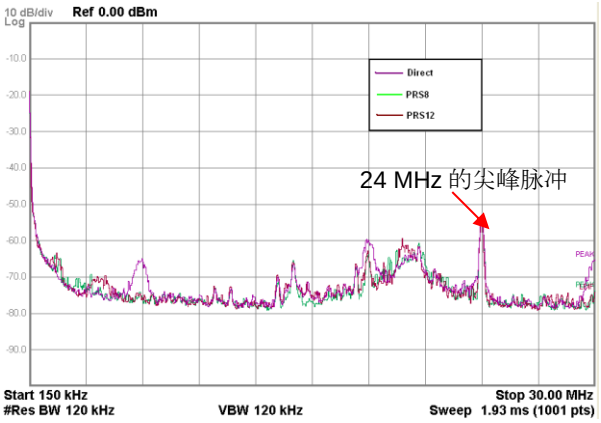
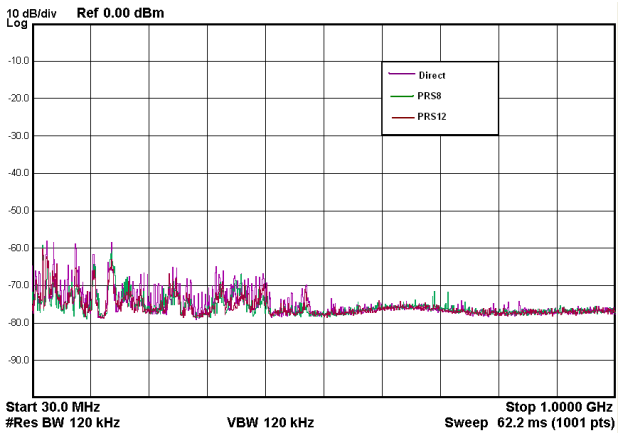


图 7-53. 感应时钟源对辐射的影响（30 kHz – 1 MHz）



注意：频率轴上显示的是对数刻度值。

7.6.1.2.7 未激活的传感器终端

将未激活的传感器接地要比将其连接至屏蔽层明显降低辐射。图 7-54 和图 7-55 显示的是各种无效传感器终端对辐射的影响。因为 IMO = 24 MHz，在频率为 24 MHz 时会出现一个尖峰脉冲，其他的尖峰脉冲是由器件硬件和固件的不同操作导致的。

图 7-54. 未激活传感器终端对辐射的影响（150 kHz – 30 MHz）

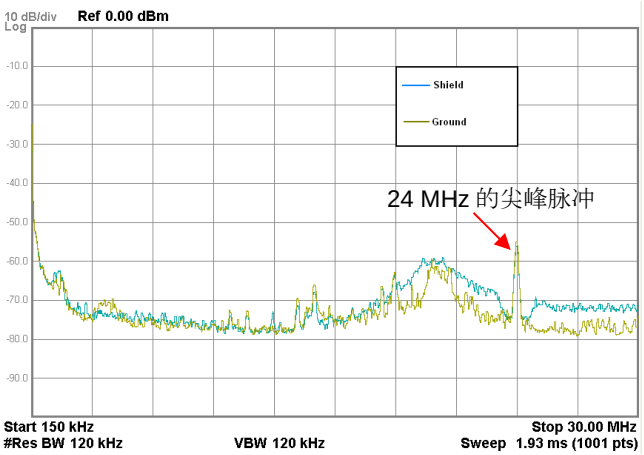
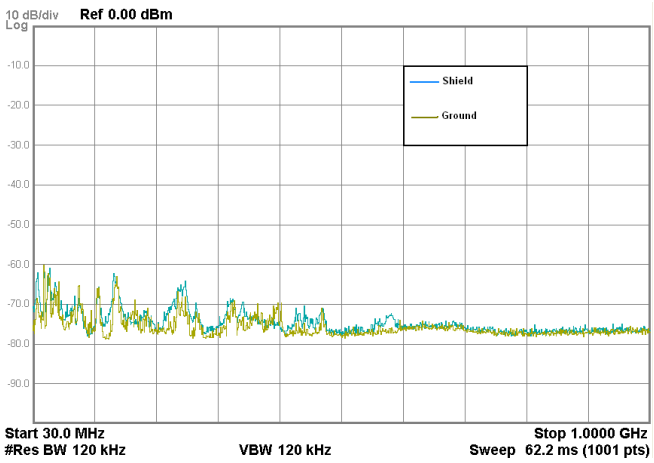


图 7-55. 未激活传感器终端对辐射的影响（30 MHz – 1 GHz）



注意：频率轴上显示的是对数刻度值。

7.6.2 导电射频噪声

通过电源和通信线路进入 CapSense 系统的噪声电流被称为导电噪声。您可通过使用以下技术来降低导电射频噪声。

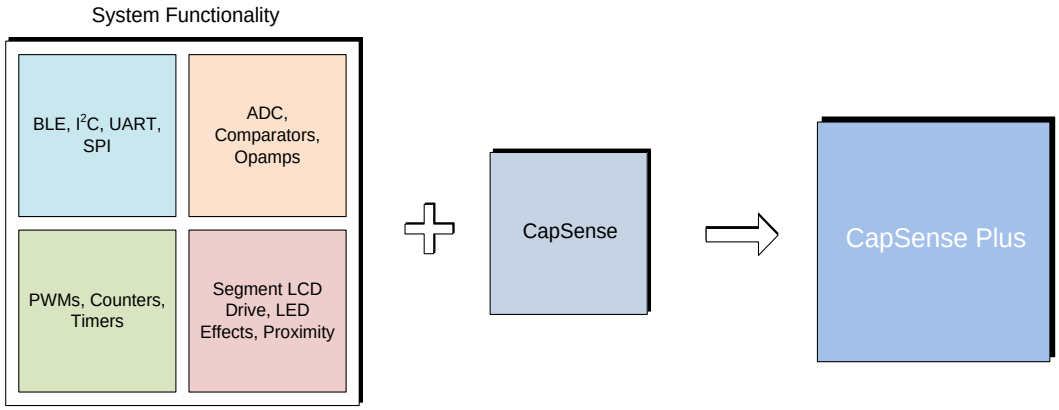
- 在电源供应引脚上使用去耦电容，以降低来自电源供应的导电噪声。请参见第 7.4.10 节和器件数据手册，了解详细信息。
- 在 PCB 上提供 GND 层（接地层）和 VDD 层，以降低电流环路。
- 如果使用线缆将 PSoC PCB 连接至电源供应，请尽可能缩短线缆长度，并考虑是否屏蔽该线缆。

为降低高频噪声，请在电源或通信线路周围放置一个铁氧体磁珠。

8 CapSense Plus

除了 CapSense 外，PSoC 4 可执行许多额外功能。该器件提供的各种功能允许您将不同的系统功能集成到单芯片内，如图 8-1 所示。这些应用被称为 CapSense Plus 应用。

图 8-1. CapSense Plus



PSoC 4 器件中的额外功能包括：

- 通信：BLE、I2C、UART、SPI、CAN 以及 LIN
- 模拟功能：ADC、比较器和运算放大器
- 数字功能：PWM、计数器、定时器和 UDB
- 段 LCD 驱动
- Bootloader
- 各种功耗模式：活动、睡眠、深度睡眠、休眠和停止

更多有关 PSoC 4 的信息，请查阅 AN79953 — PSoC 4 入门或 AN91267 — PSoC 4 BLE 入门。

PSoC 4 的灵活性和独有的 PSoC Creator IDE 能帮您快速修改您的设计，从而加快产品上市时间。各种系统功能的集成能明显降低了整个系统的成本。表 8-1 显示了一个应用示例列表，表明使用 CapSense Plus 可以明显降低成本。

表 8-1. CapSense Plus 示例

应用	CapSense	运算放大器	模数转换器（ADC）	比较器	PWM、计数器、定时器和 UDB	通信（BLE、I²C、SPI、UART）	LCD 驱动	GPIO
心率监测仪（手腕带）	用户界面：按键、线性滑条	TIA、缓冲器	心率测量、电池电压测量		LED 驱动	BLE	段式 LCD	LED 指示灯
LED 灯泡	用户界面：按键、辐射滑条	放大器	LED 电流测量	短路保护	LED 颜色控制（PrISM*）	BLE		LED 指示灯
洗衣机	用户界面：按键、辐射滑条		温度传感器	水位监测仪	蜂鸣器、FOC** 电机控制	I²C LCD 显示屏、UART 网络接口	段式 LCD	LED 指示灯
热水器	用户界面：按键、线性滑条		温度传感器、水流量传感器	水位监测仪	蜂鸣器	I²C LCD 显示屏、UART 网络接口	段式 LCD	LED 指示灯
IR 远程控制器	用户界面：按键、线性和辐射滑条、触摸板				Manchester 编码器			LED 指示灯
电磁炉	用户界面：按键、线性滑条		温度传感器				段式 LCD	LED 指示灯
电机控制系统	用户界面：按键、线性滑条				BLDC***和 FOC 电机控制			LED 指示灯
游戏 /模仿控制器	用户界面：按键、触摸板		阅读模拟游戏杆			I²C/SPI/UART 通信接口	段式 LCD	LED 指示灯
热敏打印机	用户界面：按键		过热保护，纸传感器		步进电机控制	SPI 通信接口		LED 指示灯

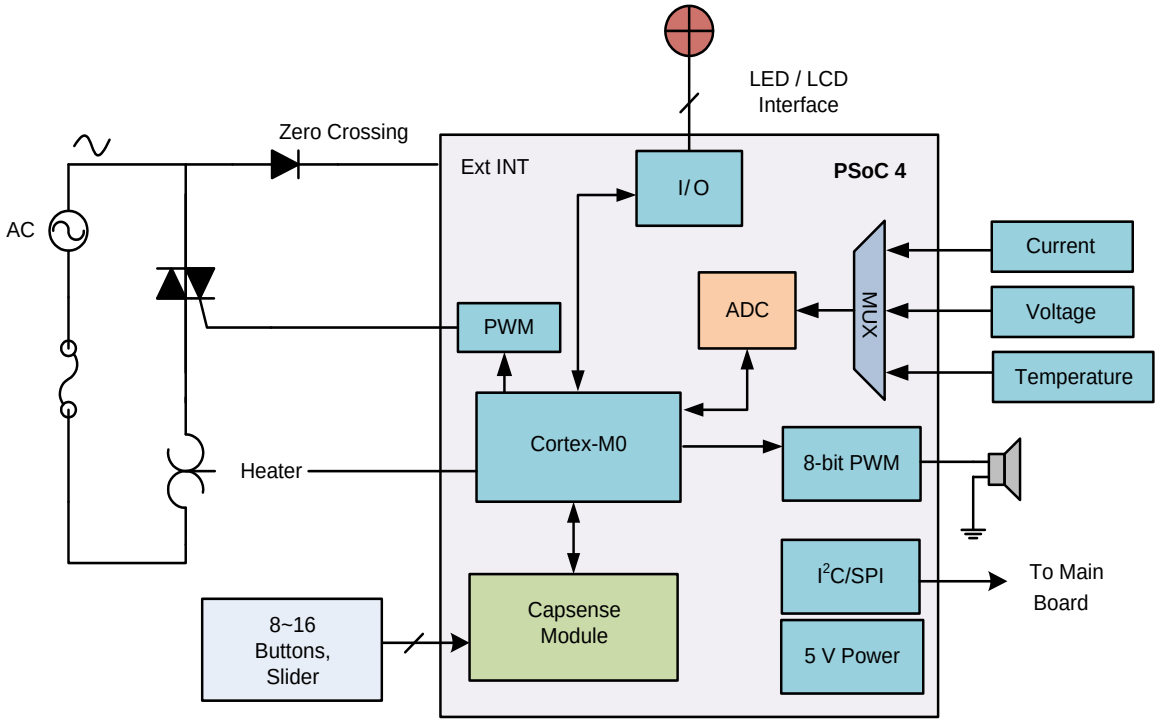
* PrISM 表示精确照明信号调制

**FOC 表示磁场定向控制

***BLDC 表示直流无刷电机

图 8-2 显示的是一个 CapSense Plus 应用（如电磁炉或微波炉）的通用框图。

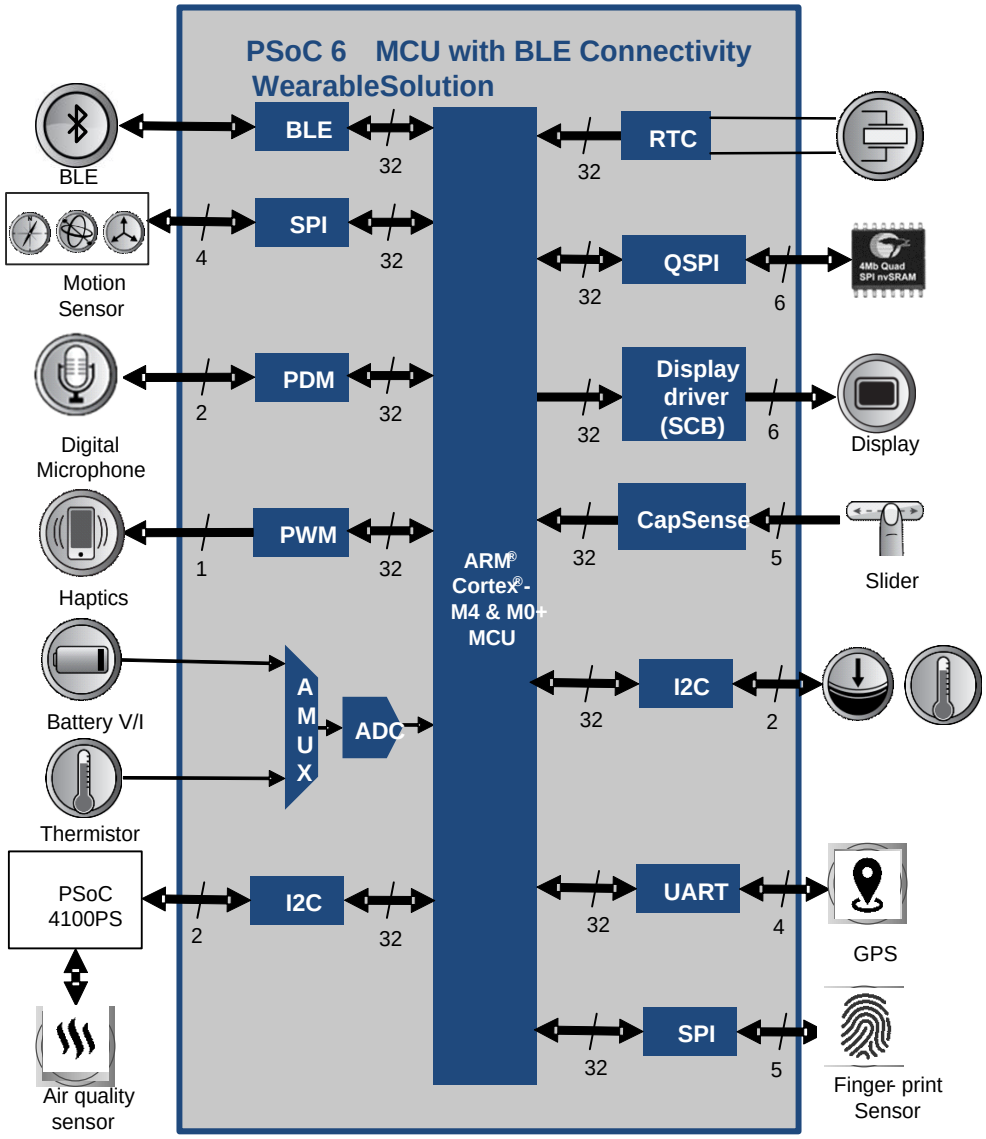
图 8-2. 带有 PSoC 4 的 CapSense Plus 系统



在本应用中，PSoC 4 中的 12 位 1 Msps SAR ADC 用于检测过电流、过电压和高温度等条件。PWM 输出用于驱动状态和警报铃声的扬声器。另一个 PWM 用于控制系统中的加热元件。用户界面由 CapSense 按键和滑条构成。PSoC 4 还可以驱动一个用于视觉输出的段式 LCD。PSoC 4 有一个连接至系统中主要电路板的串行通信模块。

图 8-3 显示的是一个健身追踪器的应用级框图。该追踪器是基于带 BLE 连接功能的 PSoC 6 MCU 的。该器件提供了单芯片解决方案和许多功能，包括活动监控，环境监控，用于用户界面的 CapSense，蓝牙低功耗（BLE）连接，等等。更多有关 PSoC 6 MCU 的信息，请参考 AN210781 — 具有蓝牙低功耗（BLE）连接的 PSoC 6 MCU 入门。

图 8-3. 带 BLE 连接功能的 PSoC 6 MCU 健身追踪器应用框图



通过 CapSense Plus 系统（如上述两个示例中的系统），可以降低电路板的大小、材料清单成本及功耗。

9 资源

9.1 网站

欲了解有关带 BLE 连接的 PSoC 4 和 PSoC 6 MCU 的信息，请访问 PSoC 4 入门、PSoC 4 BLE 入门、PSoC 6 MCU 入门以及带蓝牙低功耗（BLE）连接的 PSoC 6 MCU 等网页。

9.2 数据手册

- PSoC 4 数据手册
- PSoC 4 BLE 数据手册
- PSoC 6 MCU 器件

9.3 技术参考手册

通过 PSoC 4 技术参考手册（TRM）和 PSoC 6 技术参考手册（TRM），可以快速轻松查找有关 PSoC 4 和 PSoC 6 架构的信息，如顶层架构框图、寄存器汇总以及时序框图。

9.4 开发套件

表 4-1 列出了支持 PSoC 4 和 PSoC 6 CapSense 的赛普拉斯开发套件。

9.5 PSoC Creator

PSoC Creator 是一个最先进、简单易用的集成开发环境。请参考 PSoC Creator 主页。

9.6 ModusToolbox™

赛普拉斯推出了 ModusToolbox 软件套件，用于开发基于 PSoC 6 的 CapSense 应用。您可以从此处软件下载 ModusToolbox。相关文档如下：

- ModusToolbox 发布说明
- ModusToolbox 安装指南
- ModusToolbox 用户指南
- ModusToolbox 快速入门指南
- ModusToolbox CapSense 配置
- ModusToolbox CapSense 调谐器
- ModusToolbox Device 配置
- ModusToolbox SmartIO 配置
- PsoC Creator 到 ModusToolbox
- ModusToolbox 指令行

9.7 应用笔记

赛普拉斯提供了大量应用笔记，有助于您实现设计并使其快速运行。请参考 PSoC 4 应用笔记、PSoC 4 BLE 应用笔记和 CapSense 应用笔记和设计指南。下面是 CapSense 特定应用笔记的列表：

PSoC 3 和 PSoC 5LP 器件的设计指南

- PSoC® 3 和 PSoC® 5LP CapSense®的设计指南

CapSense Express 系列的设计指南

- CY8CMBR3XXX CapSense®设计指南
- CY8CMBR2110 CapSense®设计指南
- CY8CMBR2016 CapSense®设计指南
- CY8CMBR2010 CapSense®设计指南
- CY8CMBR2044 CapSense®设计指南
- CapSense® Express™：CY8C201XX 应用笔记

PSoC 1 器件的设计指南

- CY8C20XX7/S 设计指南
- CY8C20XX6A/H CapSense®设计指南
- CY8C21X34/B CapSense®设计指南
- CY8C20X34 CapSense®设计指南

入门应用笔记

- AN79953 — PSoC® 4 入门
- AN210781 — 具有蓝牙低功耗（BLE）连接的 PSoC 6 MCU 入门
- AN221774 — PSoC 6 MCU 入门手册

9.8 设计支持

赛普拉斯具有各种设计支持渠道，以确保 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。
- 白皮书 — 了解电容式触摸接口的高级主题。
- 赛普拉斯开发社区 — 与赛普拉斯技术社区联系并交换信息。
- 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 — 通过创建赛普拉斯支持案例，将您的设计发送给技术人员查看。您需要在赛普拉斯网站上进行注册和登录，这样才能获得技术支持。赛普拉斯建议您向技术支持发送原理图时使用其 PDF 打印版，并发送布局时使用带有层信息的 Gerber 文件。

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值的趋向。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（信号值）

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’。

电容式传感器

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开发工作，并且不需要固件开发。这些器件包括 CY8CMBR3XXX 和 CY8CMBR2XXX 系列。

质心或质心位置

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 质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的参数，只有存在手指触摸时，该参数才有效。该参数有助于抑制误触摸触摸信号。

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

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。当您触摸任何传感器时，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，以检测被激活的传感器。

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。

网格填充、网格地填充或网格铺地

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，应该以特殊网格方式填充接地层。网格图案被紧密放置、纵横交错，同丝网一样，线宽度和两条线间的距离决定了填充百分比。要求具有防水功能时，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

指的是至少包含一个传感器的 Widget。这些传感器以特殊的线性方式安排以检测手指的物理位置（在单轴上）。

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

指的是至少包含两个传感器（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。该时钟也是原始计数计数器的源。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、传感器焊盘、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设置，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

请参见电容式传感器。

传感器自动复位

指的是一种设置，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出现在传感器附近时造成的误触摸状态。

使能传感器自动复位时，即使计数差值大于噪声阈值，也更新基准线。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。

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：AN85951 — PSoC® 4 和 PSoC 6 MCU CapSense®设计指南 文档编号：001-87199				
版本	**ECN 编号**	**发布日期**	**变更者**	**变更说明**
**	3977992	04/23/2013	JOZH	新设计指南。
*A	4345299	04/14/2014	JOZH	本文档版本号为 Rev*A，译自英文版 001-85951 Rev*E。
*B	4514034	09/25/2014	LUFL	本文档版本号为 Rev*B，译自英文版 001-85951 Rev*G。
*C	4650978	02/04/2015	LUFL	本文档版本号为 Rev*C，译自英文版 001-85951 Rev*H。
*D	4723009	05/04/2015	LUFL	本文档版本号为 Rev*D，译自英文版 001-85951 Rev*I。
*E	4992984	11/11/2015	LUFL	本文档版本号为 Rev*E，译自英文版 001-85951 Rev*L。
*F	5264779	05/11/2016	YLIU	本文档版本号为 Rev*F，译自英文版 001-85951 Rev*O。
*G	5283302	05/24/2016	DCHE	No technical updates. Completing Sunset Review.
*H	5700406	04/20/2017	AESATP12	Updated logo and copyright.
*I	6840150	08/22/2019	YLIU	本文档版本号为 Rev*I，译自英文版 001-85951 Rev*W。