CY8C21x34/B

CapSense® 设计指南

文档编号 001-78594 修订版 **

赛普拉斯半导体

198 Champion Court

San Jose，CA 95134-1709

电话（美国）：1.800.541.4736

电话（国际）：408.943.2600

http://www.cypress.com

版权所有

© 赛普拉斯半导体公司，2011-2012。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任，也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，赛普拉斯不授权将其产品用作生命支持系统的关键组件，如该关键组件之运转异常和故障将会对用户造成严重伤害。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

商标

PSoC Designer™、Programmable System-on-Chip™、PSoC Creator™ 和 SmartSense™ 是赛普拉斯半导体公司的商标，PSoC® 和 CapSense® 是赛普拉斯半导体公司的注册商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

源代码

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

免责声明

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

产品使用可能受适用的赛普拉斯软件许可协议限制。

目录

1. 简介 8

1.1 摘要 8

1.2 赛普拉斯 CapSense 文档体系 8

1.3 CY8C21x34/B CapSense Plus 系列功能 10

1.3.1 高级触摸式感应功能 10

1.3.2 器件功能 10

1.4 文档规范 11

2. CapSense 技术 12

2.1 CapSense 基本原理 12

2.2 CY8C21x34/B 中的 CapSense 方法 14

2.2.1 通过 Sigma Delta 调制器 (CSD) 实施 CapSense 14

2.2.2 CSD 带 ADC (CSDADC) 的功能 15

2.2.3 SmartSense™ 自动调校 15

3. CapSense 设计工具 17

3.1 CapSense 设计工具概述 17

3.1.1 PSoC Designer™ 和用户模块 17

3.1.2 通用的 CapSense 控制器套件 18

3.1.3 通用 CapSense 控制器模块电路板 19

3.1.4 CapSense 数据查看工具 20

3.2 用户模块概述 20

3.3 CapSense 用户模块全局阵列 20

3.3.1 原始计数 21

3.3.2 基准线 21

3.3.3 差值计数（信号） 21
3.3.4 传感器状态 22
3.4 CSD 用户模块配置 22
3.4.1 CSD 不带有时钟预分频器 22
3.4.2 CSD 带有时钟预分频器 22
3.5 CSD 用户模块参数 22
3.6 CSD 用户模块高阶参数 23
3.6.1 手指阈值 23
3.6.2 噪声阈值 23
3.6.3 基准线更新阈值 23
3.6.4 传感器自动复位 24
3.6.5 迟滞 24
3.6.6 防反跳 24
3.6.7 负噪声阈值 24
3.6.8 低基准线复位 25
3.6.9 高阶参数建议 25
3.7 CSD 用户模块低阶参数 25
3.7.1 扫描速度 (Scanning Speed) 25
3.7.2 分辨率 25
3.7.3 参考 26
3.7.4 参考值 (Ref Value) 26
3.7.5 预分频器周期 (Prescaler Period) 27
3.7.6 屏蔽电极输出 27
3.8 CSDADC 用户模块配置 27
3.8.1 CSDADC 带有 PRS16 时钟源 27
3.8.2 CSDADC 带有 PRS8 时钟源 27
3.8.3 CSDADC 带有 PWM8 时钟源 27
3.8.4 CSDADC 带有 VC2 时钟源 28
3.9 CSDADC 用户模块参数 28
3.10 低阶参数 28
3.10.1 扫描速度 (Scanning Speed) 28
3.10.2 分辨率 28
3.10.3 参考 28
3.10.4 参考值 (Ref Value) 29

3.10.5 预分频器周期 (Prescaler Period) 29
3.10.6 屏蔽电极输出 29
3.10.7 启用 ADC 29
3.11 SmartSense 用户模块参数 30
3.12 低阶参数 30
3.12.1 屏蔽电极输出 30
3.12.2 调制器电容引脚 (Modulator Capacitor Pin) 30
3.12.3 反馈电阻引脚 (Feedback Resistor Pin) 30
3.12.4 阈值设置模式 30
3.12.5 灵敏度级别 31
3.12.6 手指阈值 31

4. 使用用户模块调校的 CapSense 性能 32

4.1 一般注意事项 32
4.1.1 信号、噪声和 SNR 32
4.1.2 充电/放电率 33
4.1.3 基准线更新阈值验证的重要性 34
4.2 调校 CSD 和 CSDADC 用户模块 35
4.2.1 设置硬件和软件作调校 36
4.2.2 选择预分频器 36
4.2.3 设置原始计数范围 R_b 37
4.2.4 设置高阶参数 37
4.3 配置 SmartSense 用户模块 38

5. 设计考虑事项 41

5.1 外覆层选择 41
5.2 ESD 保护 42
5.2.1 防止 42
5.2.2 重定向 42
5.2.3 钳制 42
5.3 电磁兼容性 (EMC) 注意事项 43
5.3.1 辐射干扰 43
5.3.2 辐射 43
5.3.3 传导式抗噪声和抗辐射 43

5.4 软件滤波 43
5.5 功耗 44
5.5.1 系统设计建议 44
5.5.2 睡眠-扫描方法 45
5.5.3 响应时间与功耗 45
5.5.4 测量平均功耗 45
5.6 引脚分配 46
5.7 PCB 布局准则 47

6. 耐水性 48

6.1 屏蔽电极和保护传感器 48
6.1.1 屏蔽 48
6.1.2 保护传感器 51
6.2 设计建议 53

7. 近程传感 54

7.1 近程传感器的类型 54
7.1.1 按键 54
7.1.2 导线 54
7.1.3 PCB 走线 54
7.1.4 传感器机械连接 55
7.2 设计建议 55

8. 低功耗设计要素 56

8.1 其他功耗节约技术 56
8.1.1 将驱动模式设置为模拟 HI-Z 56
8.1.2 全部放在一起 57
8.1.3 睡眠模式复杂性 57
8.1.4 待处理中断 57
8.1.5 启用全局中断 57
8.2 唤醒后执行序列 58
8.2.1 启用 PLL Mode（PLL 模式） 58
8.2.2 执行 Global Interrupt Enable（全局中断启用） 58
8.2.3 I^2C 从器件处于睡眠模式 58
8.2.4 睡眠定时器 58

9. 资源 60

9.1 网站 60
9.2 数据手册 60
9.3 技术参考手册 60
9.4 开发套件 60
9.4.1 通用的 CapSense 控制器套件 60
9.4.2 通用的 CapSense 模块电路板 61
9.4.3 线上仿真 (ICE) 工具包 61
9.5 PSoC 编程器 61
9.6 MultiChart 61
9.7 PSoC Designer 62
9.8 代码示例 62
9.9 设计支持 62

1. 简介

1.1 摘要

此文档是使用 CapSense Plus 控制器 CY8C21x34/B 系列电容式触摸传感 (CapSense®) 功能的设计指南。本指南包括以下主题：

- CapSense Plus 控制器 CY8C21x34/B 系列的功能
- CapSense 的操作原理
- CapSense 设计工具简介
- 调校 CapSense 触摸传感系统以获得最佳性能的深入指导
- 高级功能，例如耐水性和接近检测
- 电气和机械系统设计注意事项
- 在系统中设计 CapSense 所使用的附加资源和支持

1.2 赛普拉斯 CapSense 文档体系

图 1-1 和 表 1-1 概述了赛普拉斯 CapSense 文档体系。这些资源有助于用户快速访问需要成功完成 CapSense 产品设计所需的信息。

图 1-1 显示典型的电容式感应产品设计周期流程；本指南中的信息与突出显示为绿色的主题最相关。表 1-1 包含图 1-1 中每个编号任务的支持文档链接。

图 1-1. 典型的 CapSense 产品设计流程

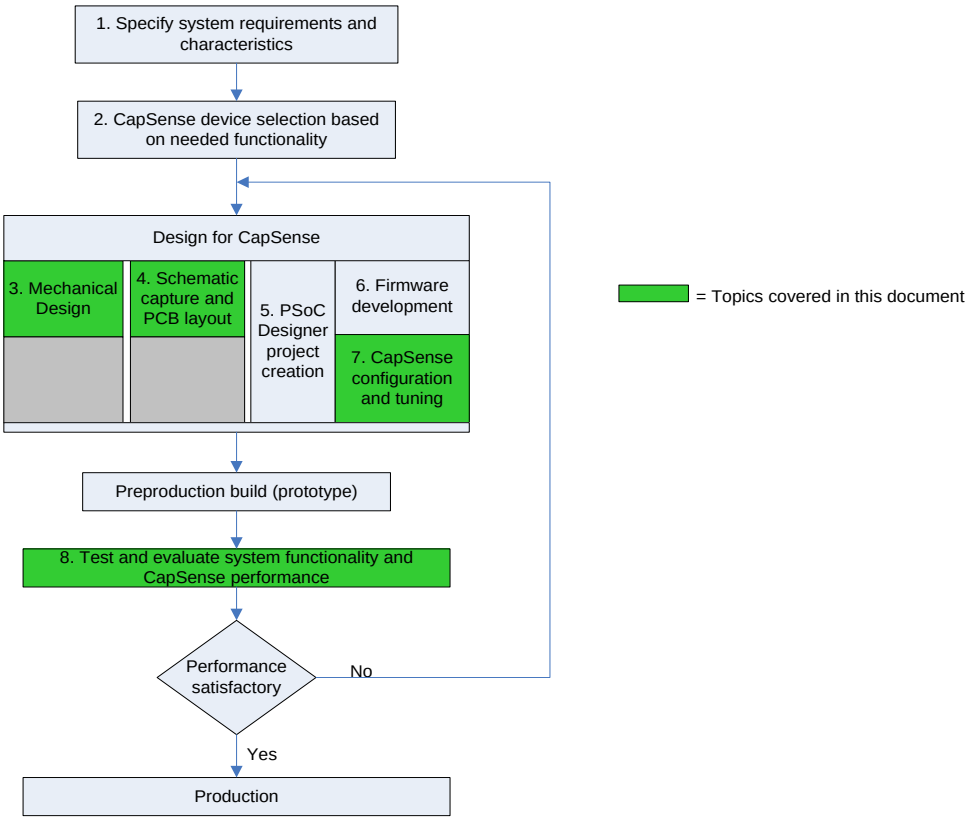


表 1-1. 图 1-1 中编号设计任务的相应赛普拉斯支持文件

图 1-1 中设计任务的编号	赛普拉斯 CapSense 支持文档
1	CapSense 入门
2	CapSense 入门 CY8C21x34/B CapSense 器件数据手册
3	CapSense 入门 PSoC 系列特定的 CapSense® 设计指南（本文档） CapSense 应用笔记
4	CapSense 入门 PSoC 系列特定的 CapSense® 设计指南（本文档） CapSense 应用笔记
5	PSoC Designer 用户指南
6	CapSense 应用笔记 CapSense 代码示例 PSoC 系列特定的技术参考手册（适用于 CY8C21x34/B）

7	PSoC 系列特定的 CapSense 设计指南（本文档） PSoC 系列特定的 CapSense 用户模块数据手册（CSD、CSDADC 和 SmartSense） PSoC 系列特定的技术参考手册（适用于 CY8C21x34/B）
8	PSoC 系列特定的 CapSense 设计指南（本文档） CapSense 代码示例

1.3 CY8C21x34/B CapSense Plus 系列功能

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

1.3.1 高级触摸式感应功能

- 可编程电容式感应元件
  - 使用 CSD 和 CSDADC 电容式感应技术支持 CapSense 按键、滑条和接近传感器的组合
  - SmartSense™ 自动调校 (CY8C21x34B)
  - 包含用来实施按键和滑条的集成式 API
  - 支持高达 24 个电容式按键和 4 个滑条
  - 支持高达 5 厘米的接近感应系统（使用 PCB 板上引线）
  - 通过屏蔽电极实现耐水性能

1.3.2 器件功能

- 高性能、低功耗 M8C Harvard 架构处理器
  - M8C 处理器的速度最高可达 24 MHz
- 灵活的片上存储器
  - 高达 8 KB 闪存和 512 字节的 SRAM
  - 支持仿真型 EEPROM
- 高精度的可编程时钟
  - 内部主振荡器 (IMO)：24/48 MHz ± 2.5%
  - 32 kHz 频率内部低速振荡器 (ILO)，用作看门狗和睡眠功能
- 增强型通用输入/输出 (GPIO) 功能
  - 高达 28 个通用 I/O (GPIO)，具有可编程引脚配置
  - 所有 GPIO 均具有 25-mA 的灌电流能力
  - 所有 GPIO 均可采用内部电阻上拉、HI-Z、开漏驱动和强驱动模式
- 外设功能
  - 计数器、定时器、PWM 和脉宽鉴别器 (PWD) 和硬件 7 段管驱动
  - I^2C 可作为主控、从器件和多主控配置通信
  - SPI、UART、IRDA 和一线通信协议

- 运行条件
  - 工作电压：2.4 V - 5.25 V，使用片上开关电压泵时工作电压可低至 1.0 V
  - 温度范围：–40 ^{o}C - +85 ^{o}C
- 封装
  - 16 引脚 SOIC (150-MIL) - 32 引脚 QFN (5 x 5 mm)
  - AEC 合格的汽车级部件 – CY8C21334 和 CY8C21534

1.4 文档规范

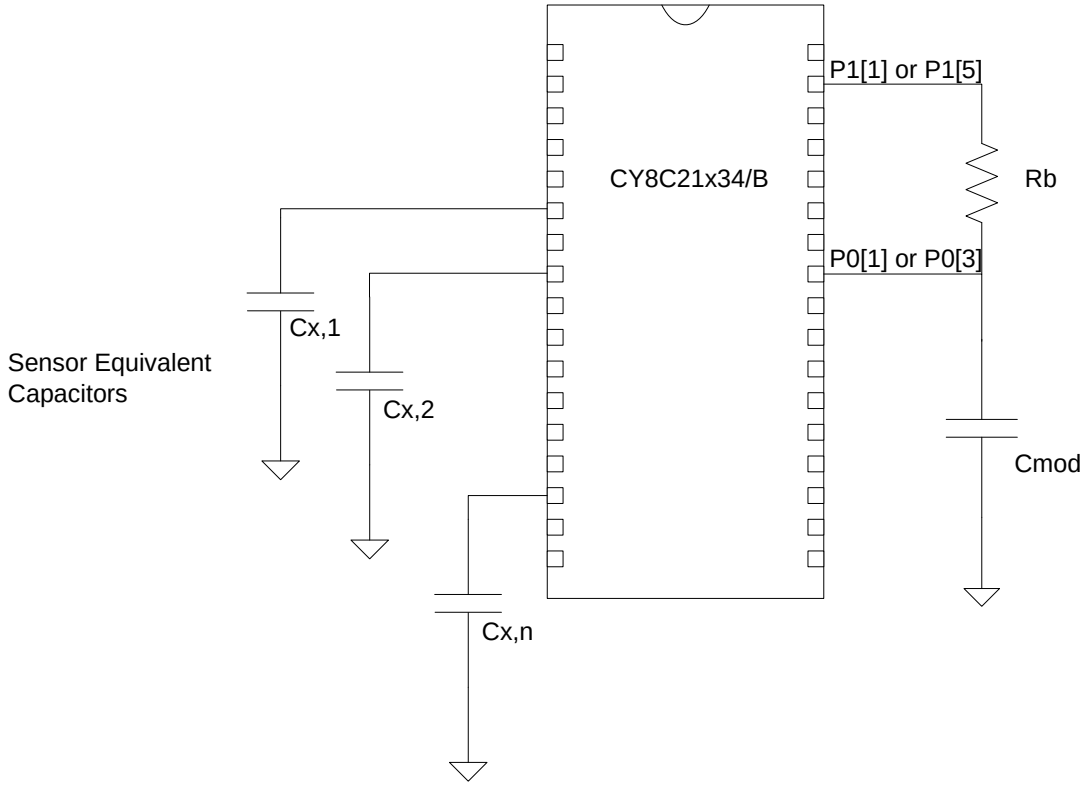
格式	用途
Courier New	显示文件位置、用户输入的文本和源代码： C:\ ...cd\icc\
斜体	显示文件名和参考文档： 读取 *PSoC Designer 用户指南*中的 *sourcefile.hex* 文件。
[Bracketed, Bold]	显示程序中的键盘命令： [**Enter**] 或 [**Ctrl**] [**C**]
File > Open	表示菜单路径： File（文件）> Open（打开）> New Project（新建项目）
粗体	显示程序命令、菜单路径和图标名称： 单击 **File（文件）**图标，然后再单击 **Open（打开）**。
Times New Roman	显示公式： 2 + 2 = 4
文本显示在灰色框中	描述产品警告信息或独特功能。

2. CapSense 技术

2.1 CapSense 基本原理

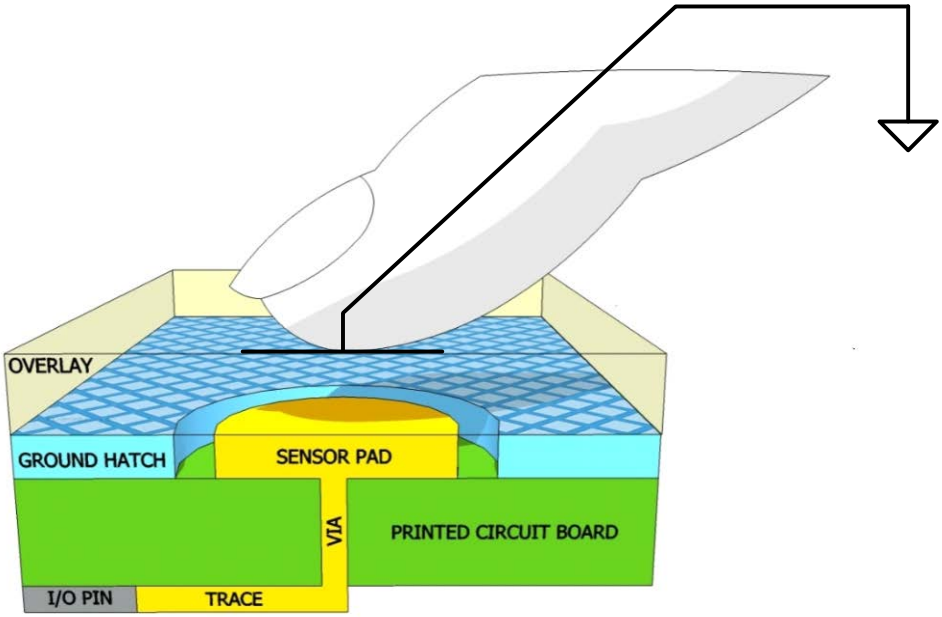
CapSense 是一种触摸式感应技术，其工作方式是在预先指定为传感器的 CapSense 控制器上测量每个 I/O 引脚的电容。如图 2-1 所示，各个传感器引脚的总电容可以建模为在同一设计中包含 $C_{X[1]}$ 至 $C_{X[n]}$ 等 n 个传感器的电容值的等效集总电容器。CY8C21x34/B 器件连接的内部电路将每个 C_X 数量级转换成数字代码，然后进行存储以备后期处理。其他组件即 Rb 和 C_{MOD} 由 CapSense 控制器内部电器来使用。

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



如图 2-2 所示，每个传感器 I/O 引脚均通过走线、过孔或在必要时同时使用二者连接至传感器板。外覆层是传感器板上面的非导电性覆盖层，这构成了该产品的触摸式接口。当手指与外覆层接触时，人体组织的导电性会产生一并行传感器板之接地导电层。如图 2-2 所示。此排列构成一个并联板电容器，通过公式 1 来计算该电容器的电容。

图 2-2. 传感器正被手指激活的之典型 CapSense PCB 横截面



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

其中：

C_F = 手指与传感器外覆层接触时发生作用的电容

ε_0 = 自由空间介电常数

ε_r = 外覆层的绝缘常数（相对介电常数）

A = 手指与传感器板的交叠面积

D = 外覆层厚度

除了平行板电容，手指与外覆层的接触也会在其自身与附近其他导体之间产生边缘电场。与平行板电容器相比，这些边缘电场的影响通常很轻微，通常可以忽略。

即使手指未触摸外覆层，传感器 I/O 引脚也有一些寄生电容 (C_P)。C_P 是由 CapSense 控制器内部寄生电容与耦合电场共同产生，其中后者是在传感器板、走线和过孔、以及接地层等系统中的其他导体、其他走线、产品机壳或外壳中的任何金属之间耦合产生的。CapSense 控制器可测量连接至传感器引脚的总电容 (C_X)。

当手指未接触传感器时：

$$C_X = C_P \qquad \text{公式 2}$$

当手指触摸传感器时：

$$C_X = C_P + C_F \qquad \text{公式 2}$$

通常， C_P 比 C_F 大一个数量级。C_P 的范围通常为 6 pF - 15 pF，但在极端情况下可以高达 50 pF。CF 的范围通常为 0.1 pF - 0.4 pF。调校 CapSense 系统时，C_P 的数量级至关重要，这将在“通过用户模块调校 CapSense 性能”这部分进行讨论。

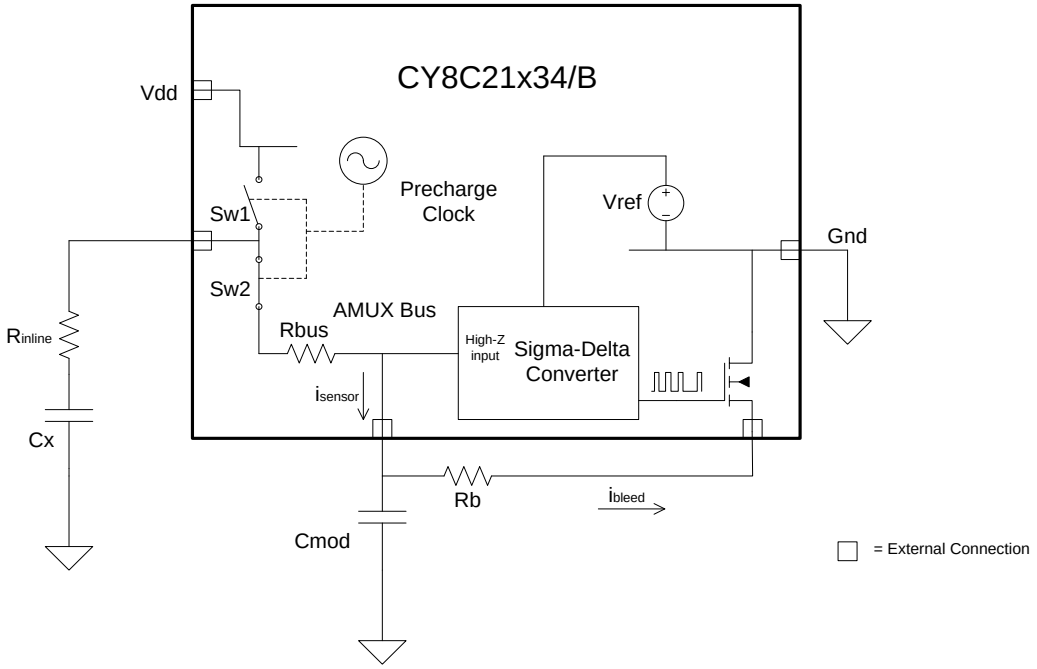
2.2 CY8C21x34/B 中的 CapSense 方法

CY8C21x34/B 器件支持多个用来将传感器电容 (C_X) 转换成数字代码的 CapSense 方法。这些方法是 CapSense Sigma Delta (CSD)、CSD 与 ADC (CSDADC) 和 SmartSense。这些方法的实施形式为 PSoC Designer 用户模块，有关描述详见本设计指南第 3 节。

2.2.1 通过 Sigma Delta 调制器 (CSD) 实施 CapSense

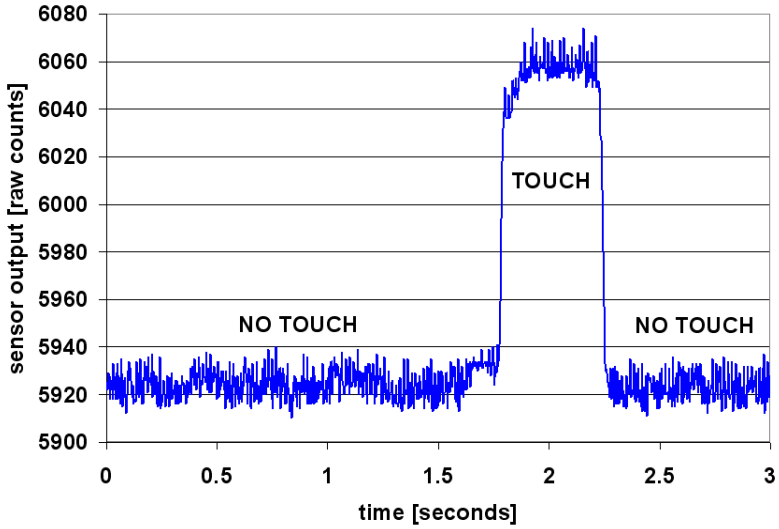
CY8C21x34/B 器件中的 CapSense Sigma-Delta (CSD) 方法将 C_X 嵌入开关式电容电路中，如图 2-3 所示。该传感器 (C_X) 分别通过负遮盖的开关 Sw1 和 Sw2 交替连接到 V_{DD} 和模拟 MUX (AMUX) 总线。Sw1 和 Sw2 由预充电时钟来驱动，用于生成输入 AMUX 总线的电流 (I_{SENSOR})。I_{SENSOR} 的数量级与 C_X 的数量级成正比。Sigma-Delta 转换器采集 AMUX 总线电压样本并生成一个用来控制恒流源 (I_{DAC}) 的调制位流，该恒流源对 AMUX 充电使 AMUX 总线的平均电压便保持在 V_{REF}。传感器释放调制电容器 (C_{MOD}) 中的电荷 I_{SENSOR}。C_{MOD} 结合 Rbus 形成低通滤波器，衰减在 Sigma-Delta 转换器输入处的预充电开关瞬态。

图 2-3. CSD 框图



保持 AMUX 平均电压为稳态值时 (V_{REF}) 时，Sigma-Delta 转换器通过控制位流占空比的方式保持平均的泄漏电流 (I_{BLEED}) 与 I_{SENSOR} 相互匹配。Sigma-Delta 转换器在传感器扫描持续期间内存储位流，累积结果为数字输出值，称为原始计数 (raw counts)，该值与 C_x 成正比。此原始计数由高层算法进行转换，以便分辨传感器的状态。图 2-4 表示从手指触摸传感器到释放这段时间内多次连续扫描得出的 CSD 原始计数。根据 CapSense 基本原理中的说明，手指触摸传感器导致 C_X 增加 C_F，并依次导致原始计数按比例增加。通过比较稳态下原始计数水平到预定阈值的转变，高级算法能够确定传感器是在 On（触摸）还是 Off （未触摸）状态。

图 2-4. 手指触摸期间的 CSD 原始计数



2.2.2 CSD 带 ADC (CSDADC) 的功能

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

2.2.2.1 ADC 特性

- 绝对和比例 ADC 输入模式带有运行时模式切换，这有助于您轻松扫描不同的传感器类型
- 单斜 ADC 用于绝对电压输入模式
- 单斜 ADC 的内置校准机制
- 用于比例输入模式的积分增量型 ADC

2.2.3 SmartSense™ 自动调校

调校触摸感应用户接口是确保系统正常运行和保持良好用户体验的重要步骤。典型的设计流程需要在初期设计阶段、系统集成期间调校传感器接口，并在最终生产时精调，方能作批量生量。调校是一个重复过程，可能非常耗时。SmartSense 自动调校的开发即用来简化用户接口开发的流程。SmartSense 易于使用，免除了整个产品开发周期（即从原型到批量生产期间）中的调校过程使得设计时间大幅缩短。SmartSense 在上电时自动调校每个 CapSense 传感器，然后在运行期间负责监测和维持传感器最佳性能。这项技术能适应于 PCB、外覆层的制造性误差和噪声源（如 LCD 反相器、交流电路噪声和开关模式电源），并自动将其修正。SmartSense 自动调校视乎设定需要介于 2.1 KB - 3.7 KB 之间的闪存，每个传感器另外需要 29B RAM。

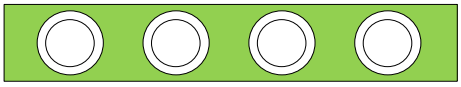
2.2.3.1 流程变化

CY8C21x34B SmartSense 用户模块 (UM) 设计与寄生电容在 5 pF - 45 pF 这一范围的传感器工作（典型的传感器 C_P 寄生电容值介于 10 pF - 20 pF 这一范围之间） 各个传感器的灵敏度参数会根据特定传感器的特性自动设置。这提高了批量生产的良率，因为每个传感器的一致性响应在 5-41 pF 这个指定范围内得到维护，而不须考虑各传感器之间的 C_P 变化。个别传感器的寄生电容可能由于 PCB 布局、PCB 制造流程各不相同或多源供应链中的 不同 PCB 供应商，有所差异。传感器的灵敏度取决于寄生电容的大小；C_P 值越高，传感器灵敏度就越低，进而导致手指触摸信号振幅降低。在有些情况下，C_P 值的变化会解调系统，致使传感器无法大到最佳性能（过于灵敏或不够灵敏），或在最坏的情况下，传感器不能正常工作。在以上情况，您必须重新调校系统，必要时，还需重新认证用户界面子系统的性能。SmartSense 自动调校能解决这些问题。

SmartSense 可以实现平台设计。设想在笔记本电脑上设置电容式多媒体触摸感应键；在两个按键之间的间距大小均取决于笔记本尺寸和键盘布局。在本例中，宽屏机型要比标屏机型的按键间距大许多。两个按键之间的间距越大，传感器与 CapSense 控制器之间的走线就越长，进而导致传感器寄生电容越高。这意味着在相同平台上设计的不同机型，CapSense 按键的寄生电容可能不同。虽然对于所有笔记本机型而言这些按键的功能均是相同的，但是，传感器必须针对每种机型进行调校。使用建议的最佳实践（参见“CapSense 入门”中的 PCB 布局），SmartSense 可以帮助您实现平台设计，调校功能会自动和有效的完成。

图 2-5. 针对 21 英寸笔记本机型设计多媒体按键

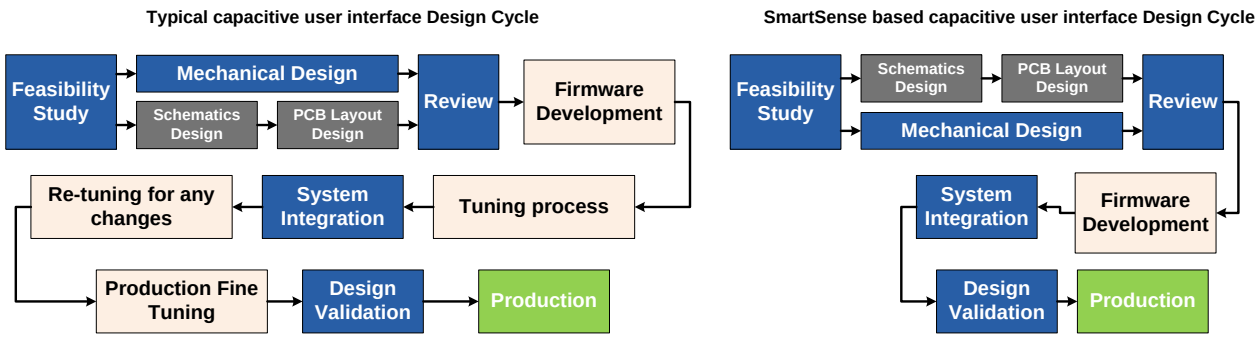
图 2-6. 针对具有相同功能和按键大小的 15 英寸笔记本机型设计多媒体按键



2.2.3.2 缩短设计周期时间

通常，电容式传感器接口设计最耗时的任务是固件开发和传感器调校。一般典型的触摸感应式控制器，当对不同机型的端口进行相同设计时，或者机器的 PCB 尺寸或传感器 PCB 布局发生变化时，均须重新调校传感器。SmartSense 设计解决了这些挑战，因为 SmartSense 需要较少的固件开发工程，无须调校和重新调校过程。这大大缩短了典型设计周期。图 2-7 比较了典型的触摸感应式控制器和 SmartSense 设计流程的周期。

图 2-7. 典型的电容式接口设计周期比较



3. CapSense 设计工具

3.1 CapSense 设计工具概述

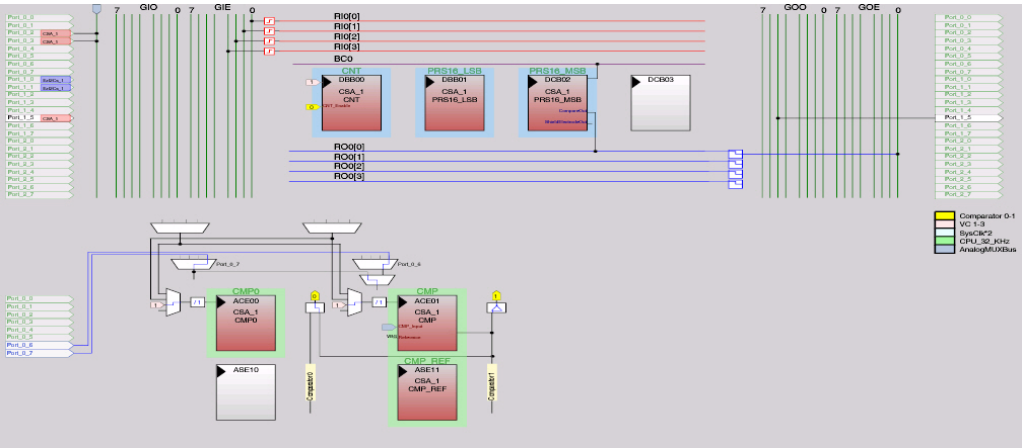
赛普拉斯提供用来开发 CapSense 电容式触摸传感应用的全套硬件和软件开发工具。CY8C21x34/B 系列的基本开发系统包括本章所讨论的组件。有关订购信息，请参见资源。

3.1.1 PSoC Designer™ 和用户模块

赛普拉斯独有的集成设计环境 PSoC Designer 支持模拟和数据模块配置、开发固件和调校及调试设计。在拖放式设计环境中使用用户模块库开发这些应用程序。用户模块配置既可以通过器件编辑器 GUI 来完成，也可以通过固件写入特定寄存器来完成。PSoC Designer 附带内置的 C 语言编译器和嵌入式编程器。现已推出用于复杂设计的专业版编译器。

CSD、CSDADC 和 SmartSense 用户模块使用开关式电容电路、模拟复用器、电压比较器、数据计数功能和高级软件例程 (API) 来实现电容式触摸传感器。其他模拟和数字外设用户模块可用于实现其他功能，例如，I^2C、SPI、TX8、定时器和 PWM。

图 3-1. PSoC Designer 器件编辑器



3.1.1.1 CapSense 用户模块入门

在 PSoC Designer 中新建 CY8C21x34/B 项目：

1. 新建 CY8C21x34/B PSoC Designer 项目。

2. 选择和放置需要专用引脚的任何用户模块（例如，I^2C 或 LCD），并分配端口和引脚。

3. 选择和放置 CSD、CSDADC 或 SmartSense 用户模块。

4. 右键单击用户模块，访问用户模块向导。

5. 设置传感器计数、滑条配置、引脚分配和关联性。

6. 设置引脚和全局用户模块参数。

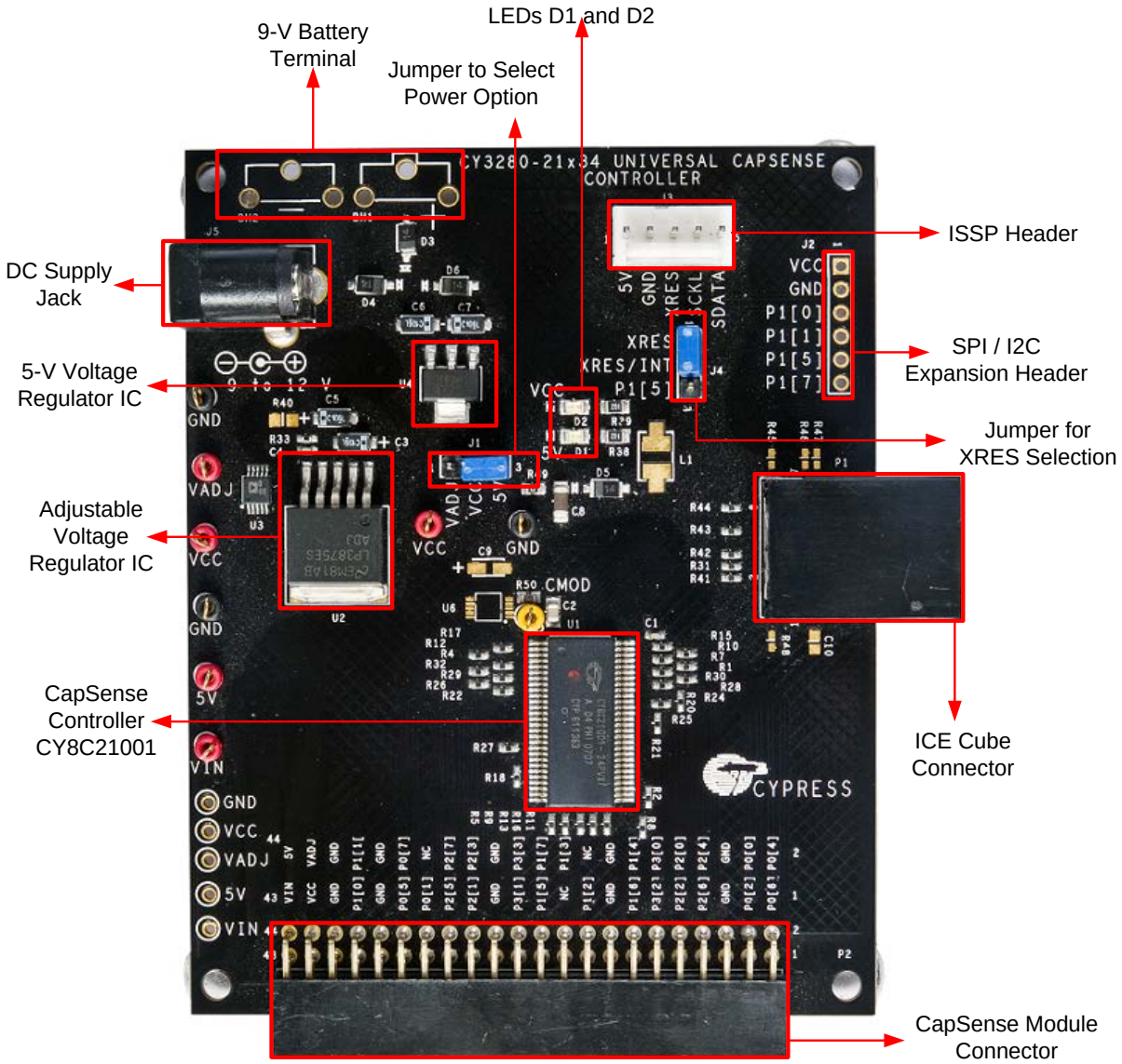
7. 生成应用并切换到应用编辑器。

8. 根据用户模块数据手册(CY8C21x34) 调整范例代码，用来实施按键或滑条。

3.1.2 通用的 CapSense 控制器套件

通用的 CY3280-BK1 CapSense 控制器套件带有预定义的控制电路和插入硬件，用来简化原型设计和调试系统。程序设计和 I^2C-to-USB 桥接器硬件也包含在内，以便进行调校和数据采集。

图 3-2. CY3280-21x34 CapSense 控制器套件



3.1.3 通用 CapSense 控制器模块电路板

赛普拉斯模块电路板带有一系列传感器、LED 和接口，用以满足您的应用需求。

- CY3280-BSM 简单按钮模块
- CY3280-BMM 矩阵按钮模块
- CY3280-SLM 线性滑条模块

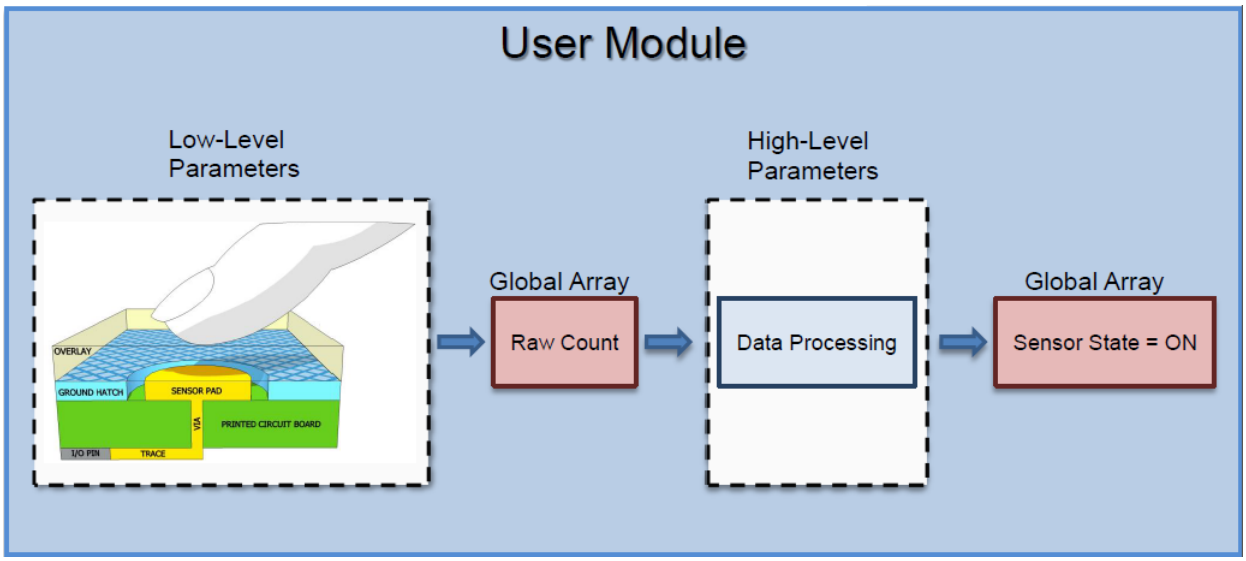
- CY3280-SRM 辐射滑条模块
- CY3280-BBM 通用的 CapSense 原型模块

3.1.4 CapSense 数据查看工具

CapSense 设计过程中，您需要多次监测用于调校和调试的 CapSense 数据（原始计数、基准线、差值计数等）。CapSense 数据查看工具共有两种，分别是 MultiChart 和桥控制面板。有关这些工具的详细说明，请参见应用笔记“AN2397 – CapSense 数据查看工具”。

3.2 用户模块概述

图 3-3. 用户模块框图



用户模拟包含整个 CapSense 系统，即从物理感应到数据处理。使用各种参数定义用户模块的行为。这些参数影响感应系统的不同部分，可分为低阶参数和高阶参数，在这些参数之间使用全局阵列进行通信。

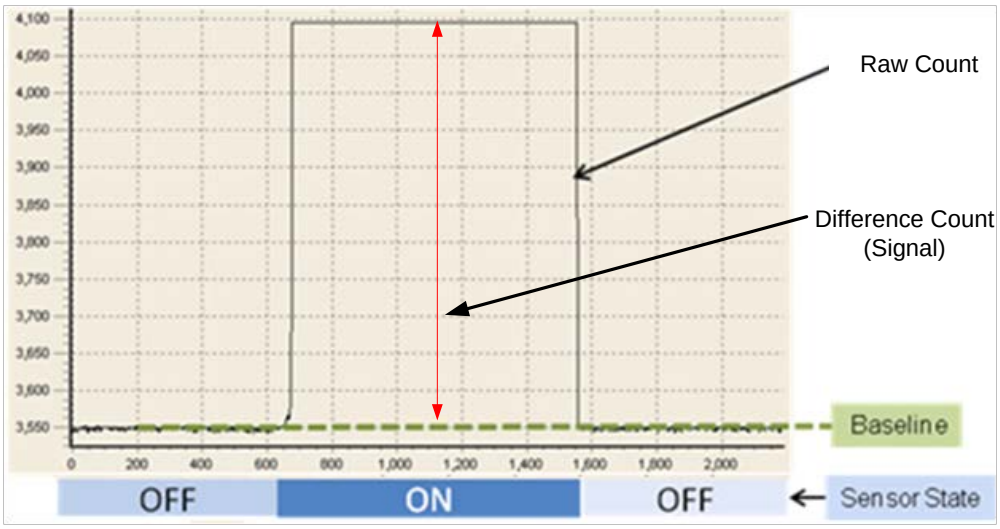
低阶参数（如扫描传感器的速度和分辨率）定义物理层上感应方法的行为，并与电容到原始计数的转换相互关联。每种感应方法具有独特的低阶参数，相关描述见“CSDADC 用户模块配置”、“CSDADC 用户模块参数”和“SmartSense 用户模块参数”这些部分。

高阶参数（如防反跳计数和噪声阈值）定义如何处理原始计数去产生相关信息，例如传感器 ON/OFF（触摸/离开）状态和手指在滑条上的大致位置。这些参数对于所有感应方法而言均是相同的，相关描述见“CSD 用户模块高级参数”这部分。

3.3 CapSense 用户模块全局阵列

学习 CapSense 用户模块参数之前，您必须熟悉 CapSense 系统所使用的某些全局阵列。这些阵列不能手动更改，但如需调试，您可对其进行检查。

图 3-4. 原始计数、基准线、差值计数和传感器状态



3.3.1 原始计数

CapSense 控制器中的硬件电路用于测量传感器的电容。调用用户模块 API *UMname*_ScanSensor()，该 API 以数字格式存储调用的原始计数的结果，这里假设 *UMname* 可能是 CSD、CSDADC 或 SmartSense。

传感器的原始计数与其传感器电容成比例。原始计数随着传感器电容值的增加而增加。

传感器的原始计数值存储在 *UMname*_waSnsResult[] 整数阵列。此阵列定义在头文件 *UMname.h* 中。

3.3.2 基准线

缓慢的环境改变（如温度和温度）会影响传感器原始计数，这会导致计数差异。

用户模块使用复杂的基线算法，用来补偿这些差异。该算法使用基准线变量来完成此功能。基准线变量记录原始计数值任何渐变差异。实际上，基准线变量存储数字低通滤波器的输出，而输入的原始计数值又馈送到数字低通滤波器中。

基线算法通过用户模块 API *UMname*_UpdateSensorBaseline 来执行，这里 *UMname* 可以是 CSD、CSDADC 或 SmartSense。

传感器的基线值存储在 *UMname*_waSnsBaseline[] 整数型阵列。此阵列定义在头文件 *UMname.h* 中。

3.3.3 差值计数（信号）

差值计数也称为传感器信号，被定义为传感器原始计数和基准值之间计数的差值。当传感器处于非活动状态时，差值计数为零。激活的传感器（通过触摸）产生正差值。

传感器的差值计数值存储在 *UMname*_waSnsDiff[] 整数型阵列，这里 *UMname* 可以是 CSD、SmartSense 或 CSA_EMC。此阵列定义在头文件 *UMname.h* 中。

差值变量通过用户模块 API *UMname*_UpdateSensorBaseline() 更新。

3.3.4 传感器状态

传感器状态表示物理传感器的活动/非活动状态。手指触摸时，传感器状态从 0 到 1 ， 手指放开时，回到 0 状态。

传感器的状态存储在名为 *UMname*_baSnsOnMask[]array 的字节阵列，这里 *UMname* 可以是 CSD、CSDADC 或 SmartSense。每个阵列元素存储八个连续传感器状态。

传感器状态通过用户模块 API *UMname*_bIsAnySensorActive() 来更新。

3.4 CSD 用户模块配置

CSD 用户模块有两个可选的时钟配置。这些配置使用 CSD 前端开关式电容器电路的不同信号源，如第 13 页的图 2-3 所示。

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

3.4.1 CSD 不带有时钟预分频器

在此配置中，16 位随机序列 (PRS) 用作 CSD 前端开关式电容器电路的信号源。系统时钟 (IMO) 是 PRS 的时钟源。PRS 扩展时钟的频率频谱，并提供优异的抗噪声功能。此配置需要 3 个数字模块。

3.4.2 CSD 带有时钟预分频器

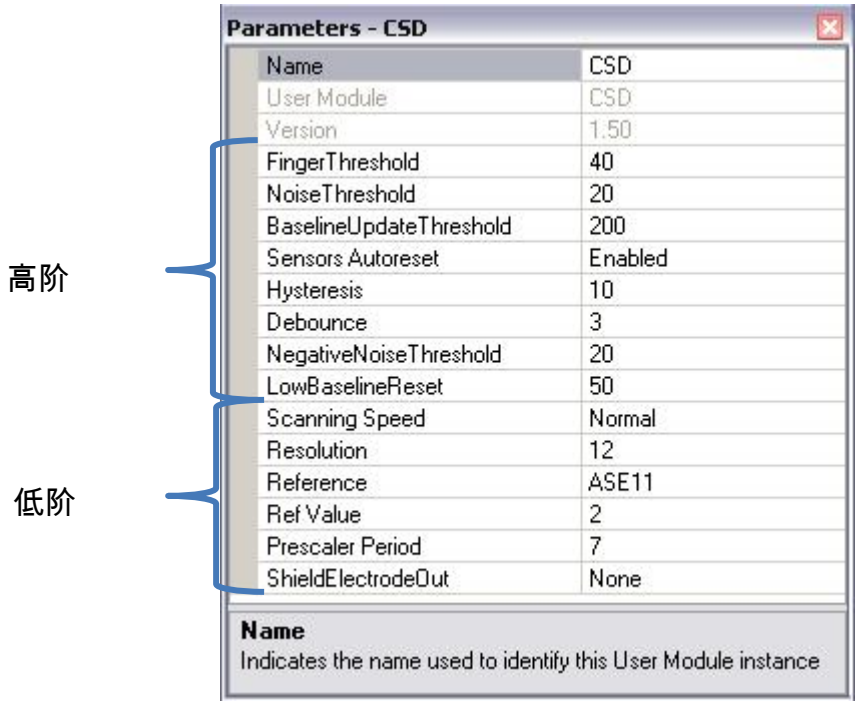
在此配置中，一个 8 位随机序列 (PRS) 用作 CSD 前端开关式电容器电路的信号源。IMO 由可调整式预分频器来分频，并用作 PRS 的时钟源。PRS 扩展时钟的频率频谱作抗噪声功能。注意：16 位 PRS 用于 CDS 不带时钟预分频器配置，可提供优异的抗噪声功能。此配置需要 3 个数字模块。

由于它需要较低的开关频率，在具有高 C_P 的环境下工作时，可以使用此配置。C_P 和开关频率之间的关系详见“选择预分频器”这部分。

3.5 CSD 用户模块参数

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

图 3-5. PSoC Designer - CSD 参数窗口



3.6 CSD 用户模块高阶参数

3.6.1 手指阈值

用户模块使用此手指阈值参数判断传感器是否为活动/非活动状态。如果传感器的差值计数值超过手指阈值，则传感器被判定为活动状态。此定义假定迟滞设为零，防反跳设为 1。

可能值为 5 到 255。

3.6.2 噪声阈值

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

对于滑条传感器而言，当差值计数大于噪声阈值时，质心计算暂停。

可能值为 3 - 255。对于正确的用户模块操作而言，噪声阈值设置应从不高于手指阈值与迟滞设置之差。

3.6.3 基准线更新阈值

当原始计数大于当前基线值，且差值计数小于噪声阈值是地，当前基线值与原始计数之间的差值累计到“桶型”时间段中。当时间段完全充满时，基线作出增量和时间段会被清空。基线更新阈值参数设置使基线值增加而时间段必须达到的阈值。

可能值为 0 到 255。

3.6.4 传感器自动复位

传感器自动复位参数确定基线值是需要随时进行更新，还是仅在差值计数小于噪声阈值时才予以更新。

启用 Sensors Autoreset（传感器自动复位）後，会不断地更新基准线。这限制了传感器的最大持续时间，但可以防止在无任何触摸的情况下，原始计数意外升高而导致传感器处于永久性打开状态。此突发性上升可能由电源电压、高能射频噪声源较大的波动或极快的温度变化波动所引起的。

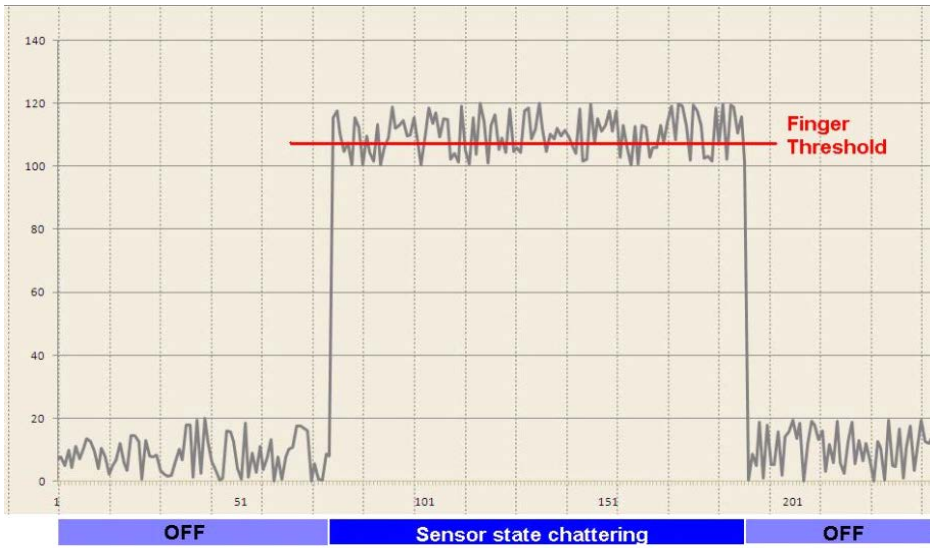
禁用 Sensors Autoreset（传感器自动复位）时，仅在差值计数小于噪声阈值参数时才予以更新基准线。

可能值为“启用”和“禁用”。

3.6.5 迟滞

该迟滞设置防止传感器 ON（触摸）状态由于系统噪声而发生振动。在公式 4 中给出此迟滞功能。该公式假设防反跳设置为 1。

图 3-6. 传感器状态与设为零的迟滞差值计数



$$if\,DifferenceCount \geq FingerThreshold + Hysteresis,\ SensorState = ON$$
$$if\,DifferenceCount \leq FingerThreshold - Hysteresis, \qquad \text{公式 3}$$

3.6.6 防反跳

防反跳参数防止原始计数尖峰脉冲将传感器状态从 OFF（释放）更改为 ON（触摸）。若要使传感器状态从关变为开，在若干特定的样本中，差值必须持续大于手指阈值与迟滞水平之和。

可能值为 1 - 255。设置为 1 则为无防反跳。

3.6.7 负噪声阈值

负噪声阈值参数充当负差值计数阈值。对于由低基线复位参数指定的采样数而言，如果原始计数小于基线值与负噪声阈值之差，则将基线值设为新的原始计数值。

可能值为 0 到 255。

3.6.8 低基准线复位

低基准线复位参数与负噪声阈值参数结合使用。它计算重设基准线所需的异常低的采样的次数。用于更正启动时手指已经在传感器上面的情况。

可能值为 0 到 255。

3.6.9 高阶参数建议

下列建议是选择理想参数设置的出发点。

- **手指阈值**：传感器为 ON（触摸）时设置为 75% 的差值计数
- **噪声阈值**：传感器为 OFF（释放）时设置为 40% 的差值计数
- **基准线更新阈值**：设置为噪声阈值的两倍
- **传感器自动复位**: 根据设计要求
- **迟滞**：传感器为 ON（触摸）时设置为 15% 的差值计数
- **防反跳**：根据设计要求
- **负噪声阈值**：设置为等于噪声阈值
- **低基准线复位**：设置为 10

3.7 CSD 用户模块低阶参数

除高阶参数外，CSD 用户模块还具有多个低阶参数。这些参数是 CSD 感测方法所特有的，决定了如何从传感器中获取原始计数数据。

3.7.1 扫描速度 (Scanning Speed)

此参数用来设置传感器扫描速度。尽管扫描速度快能使反应时间缩短，但慢速扫描有以下优势：

- 信噪比提高
- 更好地应对电源和温度的变化
- 较少需要系统中断延迟；可以处理较长的中断
- 有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。
- 这些值可能为超快、偏快、正常和偏慢。

3.7.2 分辨率

此参数确定扫描分辨率（以“位”为单位）。N 位扫描分辨率的原始计数最大值为 2^{N-1}。提高分辨率可以改善灵敏度和 SNR，但却可以缩短扫描时间。有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。这些值可能为 9 - 16 位。

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

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

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

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

3.7.3 参考

要保证 CSD 正常工作要求一个电压比较器的输入电压参考值。

可能值有：

- VBG：内部电压参考 1.3 V 衍生于一个固定带隙参考值。
- ASE11：内部变量电压参考值衍生于 PWM。
- AnalogColumn_InputSelect_1：外部电压参考，如电阻分压器或外部滤波的 PWM/PRSPWM 信号。此参考电压可以连接到 CapSense 控制器引脚，并用作电压比较器参考源。

3.7.4 参考值 (Ref Value)

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

可能的值为：0（¼V_{DD} 最小值）- 8（¾ V_{DD} 最大值）

3.7.5 预分频器周期 (Prescaler Period)

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

这些值可能为 1 - 255。

3.7.6 屏蔽电极输出

要求耐水性的应用程序应启用 Shield Electrode Out（屏蔽电极输出）。可以从其中一个备用数字行总线 (Row_0_Output_1 - Row_0_Output_3) 中选择屏蔽电极信号源。每行输出都可以路由到三个引脚当中的一个。将行 LUT 功能设置为 A。

可能值为“启用”和“禁用”。

3.8 CSDADC 用户模块配置

CSDADC 组合电容感应与 ADC 功能。它通过重新使用 CSD 和 ADC 的共同模块来节省代码空间。当您同时需要电容式感应与 ADC 功能时，您应使用 CSDADC。仅用到其中一个功能或其他功能的应用程序则应使用 CSD 用户模块，或使用 ADC8 和 ADC10 用户模块。

CSDADC 用户模块有 4 个可选的时钟配置：PRS16、PWM8、PRS8 和 VC2。这些配置使用 CSD 前端开关式电容器电路的不同信号源

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

3.8.1 CSDADC 带有 PRS16 时钟源

在此配置中，16 位随机序列 (PRS) 用作 CSD 前端开关式电容器电路的信号源。系统时钟 (IMO) 是 PRS 的时钟源。PRS 扩展时钟的频率频谱，并提供优异的抗噪声功能。此配置需要 3 个数字模块。

3.8.2 CSDADC 带有 PRS8 时钟源

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

由于需要较低开关频率，因此在 C_P 偏高的环境下工作时，应使用此配置，C_P 与开关频率之间的关系详见“通过用户模块调校 CapSense 性能”这部分。

3.8.3 CSDADC 带有 PWM8 时钟源

在此配置中，IMO 由可调式预分频器分频，并用于 CSD 前端开关式电容器电路。此配置对工作频率上之谐波及噪声较敏感。此配置需要 2 个数字模块。

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

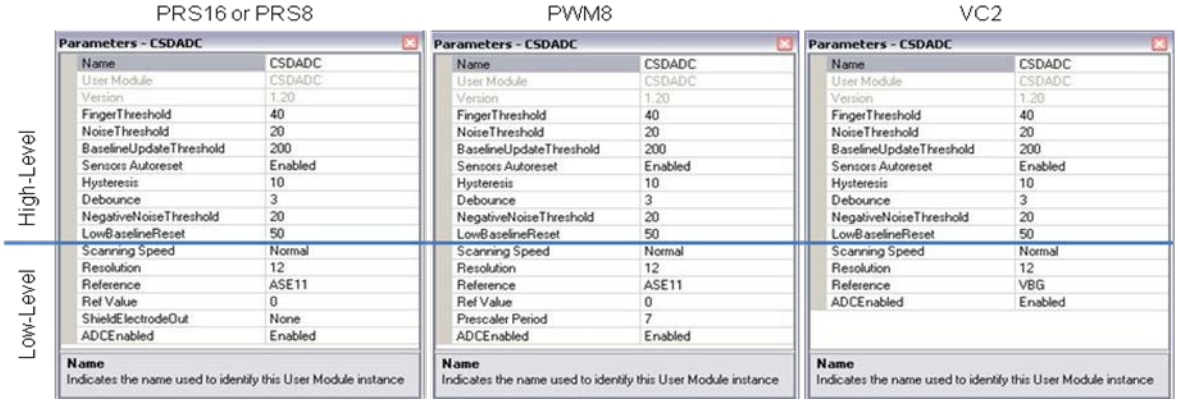
3.8.4 CSDADC 带有 VC2 时钟源

在此配置中，系统时钟使用可调式预分频器 VC2 予以分频，并用于 CSD 前端开关式电容器电路。此配置对工作频率上之谐波及噪声较敏感 此配置需要 1 个数字模块。

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

3.9 CSDADC 用户模块参数

图 3-7. PSoC Designer - CSDADC 参数窗口



3.10 低阶参数

3.10.1 扫描速度 (Scanning Speed)

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

- 信噪比提高
- 更好地应对电源和温度的变化
- 较少需要系统中断延迟；可以处理较长的中断

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

3.10.2 分辨率

此参数确定扫描分辨率（以“位”为单位）。N 位扫描分辨率的原始计数最大值为 2^{N-1}。提高分辨率可以改善灵敏度和 SNR，但却可以缩短扫描时间。有关扫描速度和分辨率性能的详细信息，请参见表 3-1 和表 3-2。

这些值可能为 9 - 16 位。

3.10.3 参考

要保证 CSD 正常工作要求一个电压比较器的输入电压参考值。

可能值有：

- VBG：内部电压参考 1.3 V 衍生于固定带隙参考值。
- ASE11：内部变量电压参考值衍生于 PWM。
- AnalogColumn_InputSelect_1：外部电压参考，如电阻分压器或外部滤波的 PWM/PRSPWM 信号。此参考电压可以连接到 CapSense 控制器引脚，并用作电压比较器参考源。

3.10.4 参考值 (Ref Value)

参考值的影响取决于参考电压参数。当参考值设置为 ASE11，使用 Ref Value（参考值）来设置参考电压值。

这些值可能为 0（¼ V_{DD} 最小值）- 8（¾ V_{DD} 最大值）。

提高 Ref Value（参考值）时，灵敏度会降低，但这会增加对屏蔽电极产生的影响。如果设计传感器的电容有明显差异（传感器板的尺寸不同），则使用 API 功能来平衡原始计数，方法是对较大电容的传感器设置较高的参考值。

当参考值设置为 VBG 或 AnalogColumn_InputSelect_1 时，Ref Value（参考值）不产生任何影响。Ref Value（参考值）不用于带有 VC2 时钟源配置的 CSDADC。

3.10.5 预分频器周期 (Prescaler Period)

预分频周期确定预充电开关输出频率。

可能值为 1 - 255。建议值为 2^{n-1}，这样可以获得最大信噪比（1、3、7、15、31、63、127 或 255）。其他值会导致更多噪声，尤其是在低分辨率和高扫描速度的情况下。

预分频器周期仅用于带有预分频器的配置。

3.10.6 屏蔽电极输出

可以从其中一个备用数字行总线 (Row_0_Output_1 - Row_0_Output_3) 中选择屏蔽电极信号源。每行输出都可以路由到三个引脚当中的一个。将行 LUT 功能设置为 A。

可能值为“启用”和“禁用”。

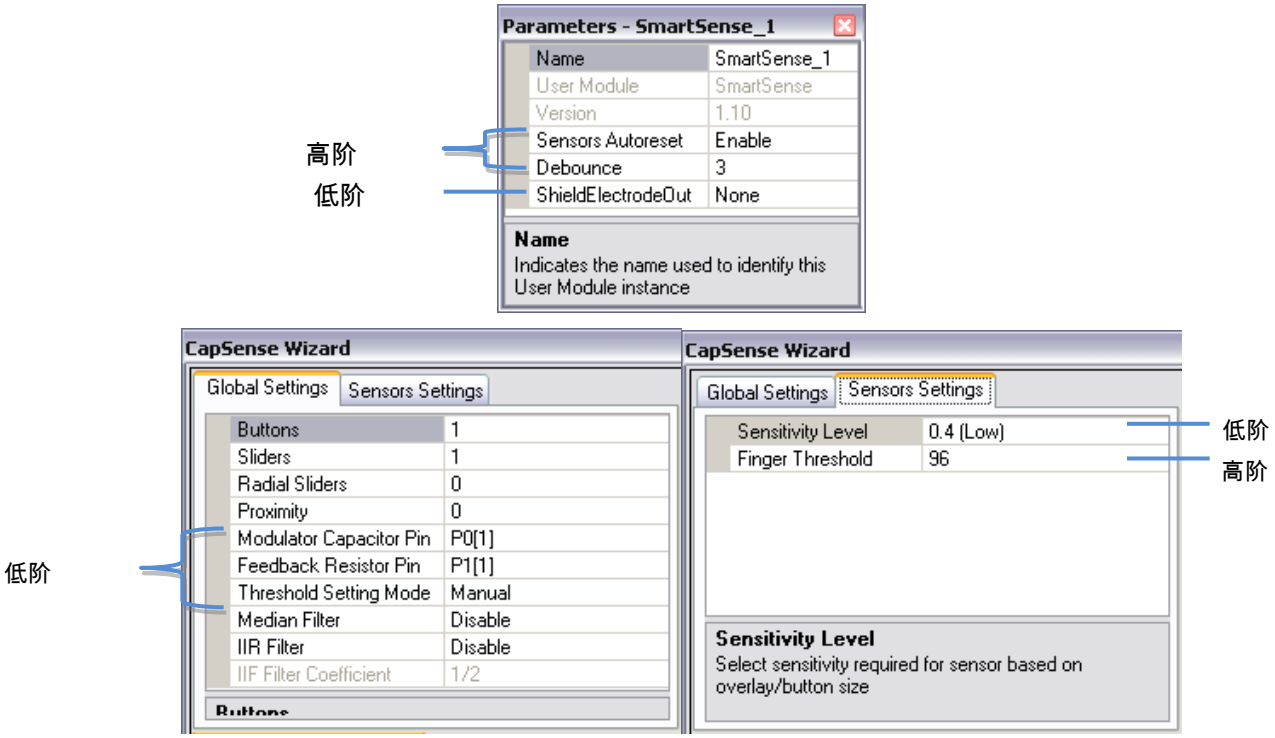
此参数仅用于带有 PRS8/PRS16 时钟源配置的 CSDADC。在带有 PWM8 时钟源配置的 CSDADC 中，屏蔽电极信号永久连接至 Row_0_Output_0。

3.10.7 启用 ADC

当此参数设置为 Enabled（启用）时，ADC 线程会嵌入编译代码中，可以从用户代码中调用该线程。当此参数设置为“禁用”时，将不包含 ADC 线程。此参数设置为 Disabled（禁用）时，如果您在设计中不再需要 ADC，可以使用此设置节省闪存。此参数设置为 Disabled（禁用）时，CSDADC 用户模块的行为与 CSD 用户模块相同。

3.11 SmartSense 用户模块参数

图 3-8. PSoC Designer SmartSense 参数



3.12 低阶参数

3.12.1 屏蔽电极输出

屏蔽电极用于降低寄生电容。屏蔽电极信号可以路由至其中一个备用数字行总线 (Row_0_Output_1 – Row_0_Output_3)。然后，每行输出都可以路由到容许引脚当中的一个。将行 LUT 功能设置为 A。

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

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

3.12.3 反馈电阻引脚 (Feedback Resistor Pin)

此参数设置引脚以连接外部反馈电阻 (R_b)。从以下可用引脚中选择：P1[1]、P1[5]、P3[1]。在某些器件封装上，某些引脚不可用。使用引脚 P1[5] 或 P3[1] 来避免编程问题。

3.12.4 阈值设置模式

使用此参数选择自动阈值设置或手动阈值设置。

3.12.5 灵敏度级别

此参数用于升高或降低传感器的灵敏度。这些值可能是 0.1 pF（最高灵敏度）、0.2 pF 和 0.4 pF（最低灵敏度）。较高灵敏度设置用于较厚外覆层的电容式传感器。

3.12.6 手指阈值

此参数仅适用于阈值设置模式设置为手动模式这种情况。建议将此值设置为存储在 SmartSense_baSnSSignal[] 阵列格式传感器信号的 80%，。使用 I^2C 或 UART 通信协议，可以轻松监测此阵列。

4. 使用用户模块调校的 CapSense 性能

理想的用户模块参数设置取决于电路板布局、按键尺寸、外覆层材料和应用需求。这些因素将在“设计注意事项”中进行讨论。调校是确定最佳的参数设置，以创造健壮可靠的传感器运行环境。

4.1 一般注意事项

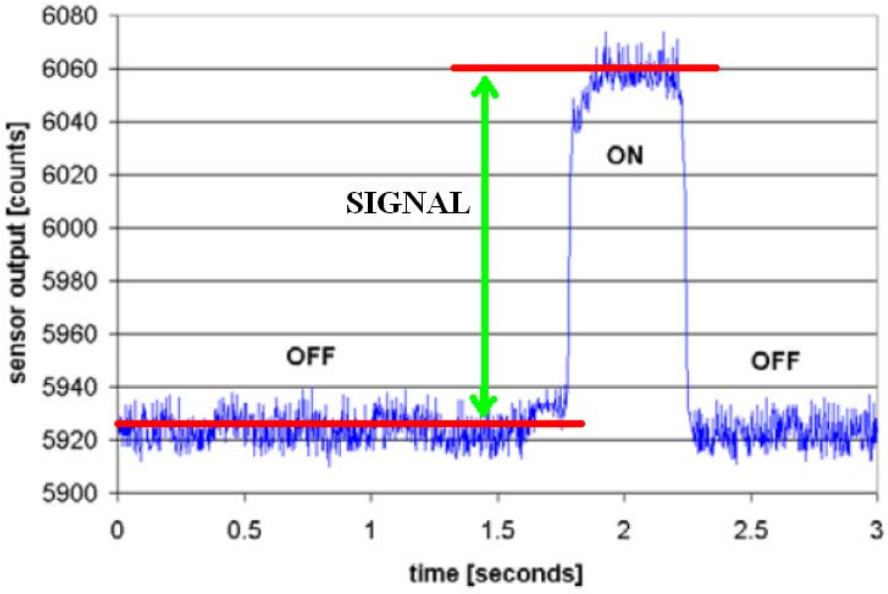
4.1.1 信号、噪声和 SNR

最佳调校的 CapSense 系统可靠地分辨 ON（触摸）和 OFF（离开）传感器状态。要实现此性能级别，CapSense 信号必须远远大于 CapSense 噪声。测量时比较信号与噪声，由此得出信噪比 (SNR)。在探讨信噪比的含义之前，首先需要定义触摸感应下的信号和噪声。

4.1.1.1 CapSense 信号

CapSense 信号是手指放在传感器上时传感器响应的变化，如图 4-1 所示。传感器输出是数字计数器，并对应传感器电容变化的值。在此示例中，手指离开传感器后，平均性能级别为 5925 次计数。当手指放在传感器上时，平均输出增至 6060 次计数。传感器信号追踪手指触摸前后数值的变动，信号值 = 6060 - 5925 = 135。

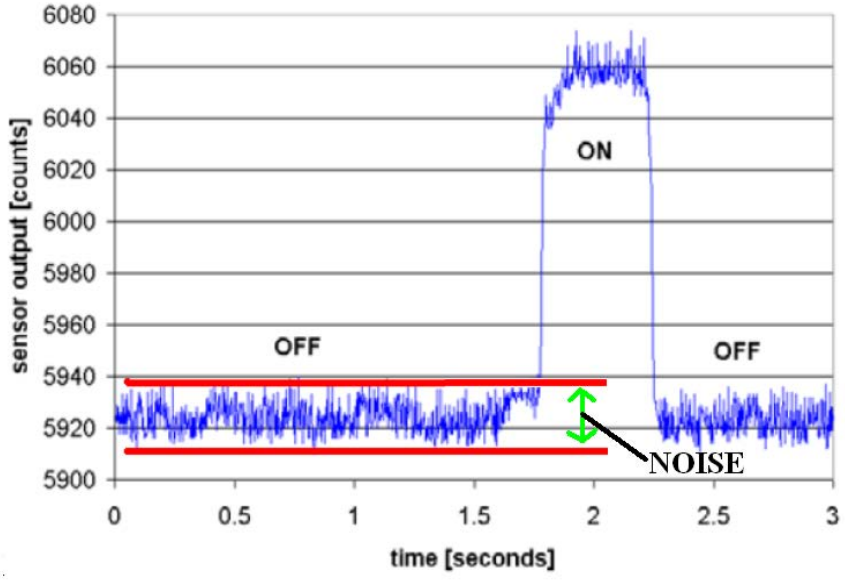
图 4-1. CapSense 信号的示例



4.1.1.2 CapSense 噪声

CapSense 噪声是手指未触摸时传感器响应的峰至峰变化，如图 4-2 所示。 在此示例中， 手指未触摸时输出波形跳跃限幅为最大为 5938 个计数和最小为 5912 个计数。 CapSense 噪声是波形最大值和最小值的差， 因而噪声值 = 5938 - 5912 = 26。

图 4-2. CapSense 噪声示例



4.1.1.3 CapSense SNR

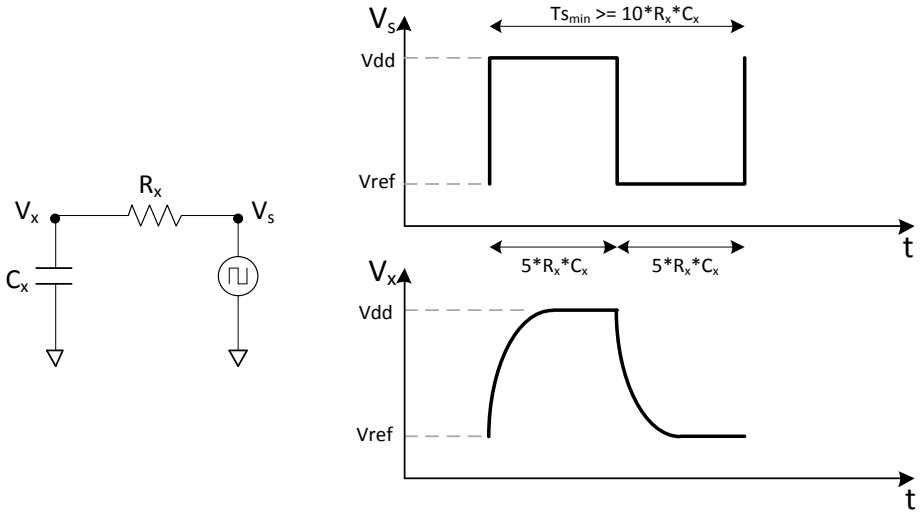
CapSense SNR 是信噪简比。继续该示例，如果信号为 135 个计数，噪声为 26 个计数，那么，SNR 为 135:26，这就将 SNR 降低到 5.2:1。建议 CapSense SNR 最小为 5:1，这表示信号比噪声大 5 倍。滤波器通常实施在固件中以降低噪声。更多信息，请参见软件滤波。

4.1.2 充电/放电率

调校过程中，要实现最大灵敏度，必须在每个周期内对传感器电容器进行完全充电和放电。充电/放电路径开关在两种状态下来回切换，根据用户模块中的时钟参数来设置速度。

充电/放电路径包含串联电阻，这降低了电荷转移的速度。此电荷转移变化率由 RC 时间常数描绘，涉及传感器电容和串联电阻，如图 4-3 所示。

图 4-3. 充电/放电波形



确保将充电/放电率设置为与 RC 时间常数兼容的级别。基本原则容许每次跃迁周期为 5RC，每个周期有两次跃迁（一次充电、一次放电）。最小时间周期和最大频率的计算公式如下：

$$Ts_{min} = 10 \times R_X C_X Ts_{min} = 10 \times R_X C_X \qquad \text{公式 4}$$

$$fs_{max} = \frac{1}{10 \times R_X C_X} fs_{max} = \frac{1}{10 \times R_X C_X} \qquad \text{公式 5}$$

例如，假设串联电阻包括 560-Ω 外部电阻和高达 800 Ω 的内部电阻，传感器电容为典型值：

R_X = 1.4 kΩ

C_X= 24 pF

在本例中，时间常数值和前端开关最大频率为：

Ts_{min} = 0.34 µs

fs_{max} = 3 MHz

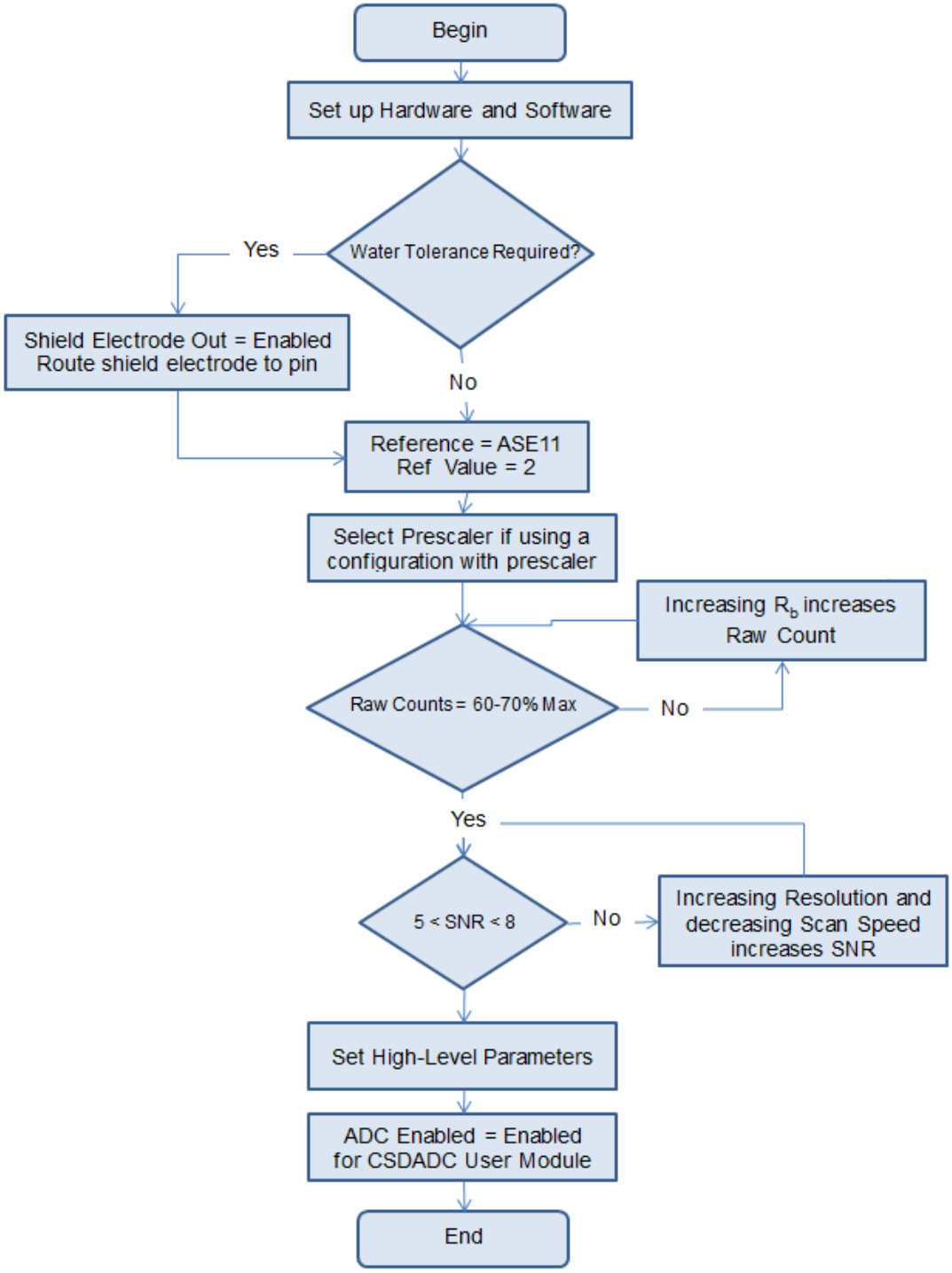
4.1.3 基准线更新阈值验证的重要性

随着时间的推移，温度和温度会导致平均计数漂移。基准线是 CapSense 测量值的参考计数量，这对于环境影响补偿而言起着重要作用。高级决策基于基准线建立的参考级别，例如，手指触摸和手指离开状态。由于每个传感器都有与配套的寄生电容，每个电容传感器都独立拥有基准线。

基准线跟踪由基准线更新阈值参数设置的速率计数变化。确保更新速率与预期应用程序相互匹配。如果更新速率过快，那么，基准线将补偿手指产生的所有变量，并不予检测手指移动的情况。如果更新速率过慢，那么，相对较慢的环境变化可能导致手指出错。开发过程中，应当验证基准线更新阈值的设置。

4.2 调校 CSD 和 CSDADC 用户模块

图 4-4. 调校 CSD 和 CSDADC 用户模块



4.2.1 设置硬件和软件作调校

调校流程开始前，需要准备某些硬件和软件工具。这些工具列在“CapSense 设计工具概述”这部分。调校时，需要通过通信接口监测传感器的原始计数，基准线和差值计数。可以在这里下载使用 I^2C 通信协议与个人计算机进行通信的代码示例。下载完示例项目后：

1. 使用 PSoC Designer 打开代码示例。在 Workspace Explorer 中，右键单击 **CSD User Module（CSD 用户模块）**，然后选择 **CSD Wizard（CSD 向导）**。可以显示分配到该引脚的 4 个 CapSense 按钮。根据您的应用需求更改按钮和引脚的分配数量。
2. 如有必要，可以更改调制器电容引脚和反馈电阻引脚。
3. 生成和创建项目。
4. 使用 Hex 文件编程 CapSense 控制器。
5. 有关如何监测原始计数、基准线和差值计数的详细内容，请参见“使用 USB-I2C 工具监测 CapSense 数据的说明”这部分，这可以在说明文档的代码示例中找到。

4.2.2 选择预分频器

要选择预分频器的值，请执行以下程序：

1. 使用 LCR 或 CapSense 控制器测量传感器寄生电容的最大值。如果使用 LCR 计量仪，请跳过第 5 步。
2. 使用 CSD 用户模块，包含时钟预分频器和以下参数：
   a. 扫描速度 = 正常
   b. 分辨率 = 16
   c. 参考 = VBG
   d. 预分频器周期 = 15 (SysClk = 24 MHz)、7 (SysClk = 12 MHz) 或 3 (SysClk = 6 MHz)
   e. ShieldElectrodeOut = 根据应用启用或禁用
3. 创建项目并将项目下载到 CapSense 控制器。
4. 仅监测所有传感器（包括滑条段）的原始计数，记录最高原始计数值。
5. 使用公式 7 计算寄生电容的最大值。

$$C_X = \frac{\text{Highest Raw Count}}{88.85e12 \times (V_{DD} - 1.3)}$$ 公式 6

6. 使用公式 8 计算 fs_{max}。

$$fs_{max} = \frac{1}{10 \times R_X C_X}$$ 公式 7

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

$$\text{Prescaler} \geq \frac{\text{SysClk}}{fs_{max} \times 2} - 1$$ 公式 8

9. 针对带有 PWM8 时钟源配置的 CSDADC：

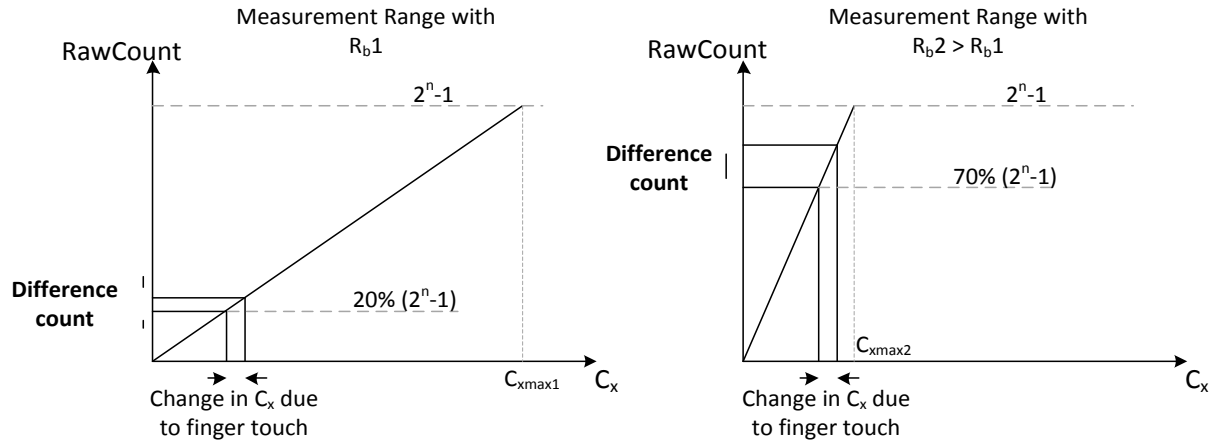
$$\text{Prescaler} \geq \frac{\text{SysClk}}{fs_{max}} - 1$$ 公式 9

10. 为优化 SNR 性能，将预分频器的计算值舍入到最近位 $2^n - 1$ 的值（1、3、7、15、31 等）

4.2.3 设置原始计数范围 R_b

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

图 4-5. CSD 测量范围



注意这两个原始计数测量值范围图（图 4-5）。在左侧图形中，$R_b = R_b1$，系统可以测量的电容值高达 C_{xmax1}。 原始计数为其最大值的 20%。 由手指触摸面积非常小， 低灵敏度产生的差值计数。 在右侧图形中， 泄露电阻升高， $R_b2 > R_b1$。 此时， 原始计数为最大值的 70%。 在这种情况下， 差值计数远远大于第一种情况的差值计算， 灵敏度增加。

R_b 的典型值介于 500 Ω 和 10 kΩ 之间。查找最佳电阻：

1. 从 R_b 2-kΩ 电阻开始。
2. 监测所有传感器的原始计数测量值。
3. 调校 R_b，直到原始计数达到选定扫描分辨率全量程范围的 70% 时为止。例如，如果选择 9 位分辨率，则全量程范围为 511 个计数。R_b 应调整到原始计数为 358 个计数。

设计多个传感器时，可能很难将所有原始计数精确地设置到 70%。在此情况下，可以将所有原始计数最大值设置到 60-70% 之间。

4.2.4 设置高阶参数

遵循“高阶参数建议”这部分给出的建议值。

4.3 配置 SmartSense 用户模块

下面是在典型的 CapSense 应用环境下配置 SmartSense 用户模块的基本准则：

1. 准备目标电路板。组装目标应用 PCB ，在它上面固定外覆层。使用不导电胶水或特殊胶带来粘贴外覆层。避免在 PCB 与外覆层之间留有空气间隙，否则会极大降低灵敏度，且由于您触摸时气隙会漂移，导致出现许多假按键触发情况。

2. 设置实时组图工具，例如，Multi-Chart（参见第 3.1.4 节），以便实时监测一个或多个数据系列。在调校过程中，必须实时观察传感器原始计数 (waSnsResult)、基准线 (waSnsBaseline)、差值计数 (waSnsDiff)、信号 (baSnsSignal) 和按键手指阈值 (baBtnFThreshold)。使用 UART-USB 桥或 I2C-USB 桥硬件并通过 USB 端口来构建 CapSense 控制器与个人计算机之间的接口。请勿使用 LCD 或其他任何数字显示器来监测数据，因为这些工具运行较慢，无法可视化动态数据。

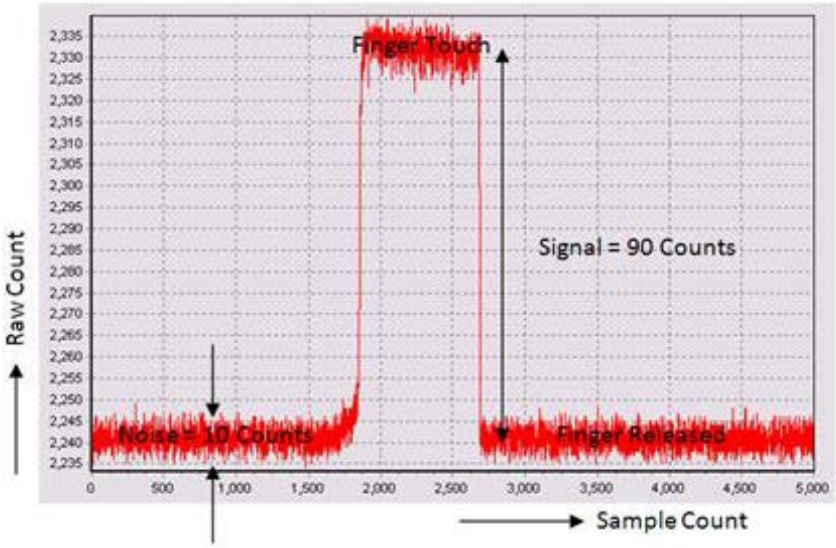
3. 开始时的默认配置：

   - 设置 IMO = 24 MHz、CPU_Clock = SysClk/1
   - 放置 SmartSense 用户模块配置，针对 IMO = 24 MHz
   - 禁用“Sensor Autoreset”（传感器自动复位）。将“Debounce”（防反弹）设置为 3，“ Shield Electrode Out”（屏蔽电极输出）设置为 None（无）。
   - 在 CapSense 向导的 Global Settings（全局设置）下，在“Button”（按钮）类别下分配一个传感器。然后再选择调制电容和反馈电阻引脚。将阈值设置模式设置为手动模式，然后禁用中值和 IIR 滤波器。在电路板上，确保 Rb 值为 15K，Cmod 为 10 nF。
   - 在 CapSense 向导的 Sensor Settings（传感器设置）选项卡下，将“Sensitivity Level”（灵敏度级别）设置为 Low（低），“Finger Threshold”（手指阈值）设置为 96。

4. 使用用户模块数据手册中提供的样本固件代码生成/创建项目。

5. 监测传感器原始计数，计算 SNR。图 4-6 显示监测的传感器原始计数。根据 CapSense 最佳实践，强健设计的 SNR 必须大于 5。在这种情况下，监测 SNR 的值为 9，大于 5。如果 SNR 的值小于 5，则阅读第 7 步中的 SNR 改进技术。

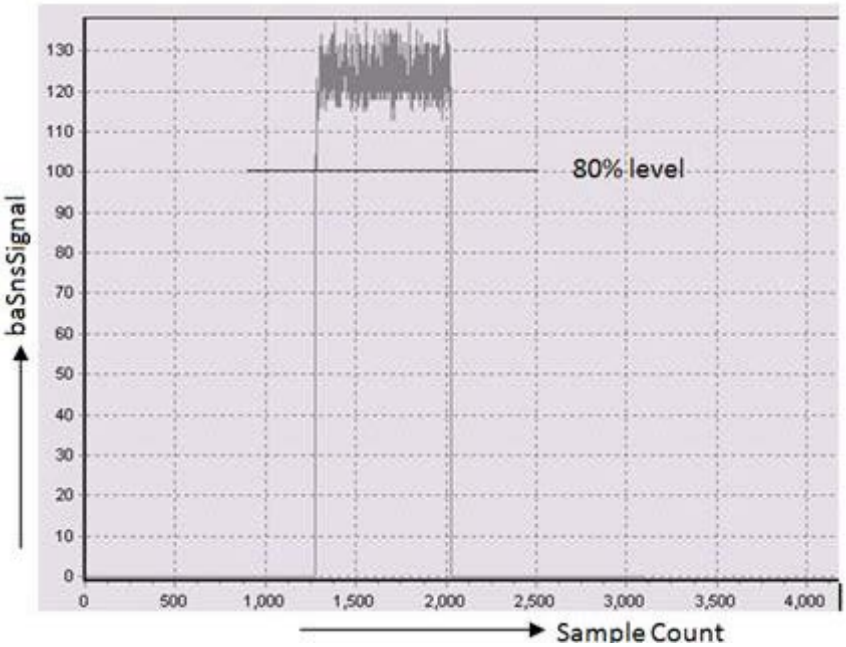
图 4-6. 监测传感器原始计数



6. 如果应用环境不要求精确控制手指阈值，或者选择了自动阈值设置模式，则可以跳过此步骤。自动阈值模式使用一种算法，它可以根据系统中发现的噪声来动态调整手指阈值。在以下场景中，应在自动阈值模式上选择手动模式：
   - 如果 RAM/ROM 封装小于传感器的数量。
   - 要求精确调整传感器手指阈值的设计。
   - 需要在噪声条件下运行的设计。

如果使用手动阈值设置模式，要完成调校流程，还必须设置手指阈值。监测传感器信号 baSnsSignal。确定感应器信号最大值的 80%，此值便是手指阈值。在本示例中，该信号接近 100 个计数，如图 4-6 所示。

图 4-7. 监测传感器信号 baSnsSignal



此时，打开 CapSense 向导，在 Sensor Settings（传感器设置）选项卡下，将手指阈值设置为计算值（在本例中，该值为 100，如图 4-7 所示）。

7. 重复上一步，完成 SmartSense 传感器调校流程。然而，第 5 步计算的 SNR 结果可能小于 5:1，例如，当使用的外覆层较厚时，会出现这种情况。在此情况下，请遵循这些建议：
   - 将灵敏度级别从“Low”（低级）改为“Medium”（中级）。如果 SNR 仍未达到 5:1，则将灵敏度级别更改为“High”（高级）。
   - 有时，由于系统中观察的噪声冗余，因此无法达到 5:1 这个要求。在这些情况下，应使用软件滤波器。启动 IIR 滤波器，将比例系统调至 1/2。如果这未降低噪声，则将 IIR 的比例系数改为 1/4。

5. 设计考虑事项

在您的应用中设计电容式触摸感应技术时，切记 CapSense 器件配置在较大规模的框架中，这一点至关重要。特别留意从 PCB 布局到用户接口及终端用户操作环境的每一个详细级别均可以带来强大而可靠的系统性能。有关 EMC 注意事项的更多信息，请参见"CapSense 入门"这个章节。

5.1 外覆层选择

在"CapSense 基本原理"这部分，使用公式 1 计算手指电容，如下所示：

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

其中：

ε_0 = 自由空间介电常数

ε_r = 外覆层的绝缘常数

A = 手指与传感器板的交叠面积

D = 外覆层厚度

要提高 CapSense 的信号强度，请选择介电常数较高的外覆盖层材料，降低外覆层厚度，并增加按键直径。

表 5-1. 外覆盖层材料的介电强度

材料	击穿电压 (V/mm)	最小值 外覆层厚度为 12 kV (mm)
空气	1200–2800	10
木材 – 干性	3900	3
玻璃 – 普通	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

导电性材料不能用作外覆层，因为这种材料对电场模式造成干扰。鉴于此，在覆盖层中不使用包含金属颗粒的油漆。

粘合剂常用来将外覆层粘贴到 CapSense PCB 上。由一种 3M™ 制造称为 200MP 的的丙烯酸透明粘附膜是适用于 CapSense 应用的材料。这种粘合剂是点胶式胶纸胶带卷（3M™ 产品编号 467MP 和 468MP）。

5.2 ESD 保护

强健的 ESD 耐受性是精密系统设计的自然副产品。考虑到终端产品的接触放电情况，特别是用户界面上的接触放电，18-kV 的放电事件可能不会对 CapSense 控制器造成任何损坏。

CapSense 控制器引脚可以承受 2-kV 电压的直接冲击事件。在大多情况下，外覆层为控制器引脚提供了充分的 ESD 保护。表 5-1 列出了用于保护 CapSense 传感器免受 12-kV 放电（IEC 61000-4-2 规范）冲击所需的各种外覆层材料的厚度。 如果外覆层材料未提供充分保护，则在以下情况下应用防静电措施：预防，重新定向，钳制。

5.2.1 防止

确保触摸表面上的所有路径均有大于潜在高电压接触的击穿电压。此外，设计您的系统，在 CapSense 控制器与可能的 ESD 来源之间保持适当的距离。如果无法保持充足的距离，则在 ESD 源和 CapSense 控制器之间放置可承受高击穿电压材料的保护层。一层 5 mil 厚的 Kapton®胶带可以经受住 18 kV。

5.2.2 重定向

如果产品包装紧密，则不可能防止高放电事件。在此情况下，可以通过控制放电事件发生的位置来保护 CapSense 控制器。标准做法是在连接到机壳接地的电路板周边放置一个护圈。如在《PCB 布局指南》中所示， 在按钮或者滑条传感器周围提供一个规划接地层，该举能够重新定向 ESD 放电事件，使其离开传感器和 CapSense 控制器。

5.2.3 钳制

因为 CapSense 传感器需要特别靠近触摸表面， 因此， 重定向放电路径可能是不可取的方法。 在这种情况下， 合理的做法是添加串联电阻或者专用 ESD 保护器件。

建议系列电容值为 560 Ω。

更有效的方法是在较弱引线使用 ESD 静电保护器件。适用于 CapSense 的 ESD 静电保护器件必须是低电容。表 5-2 列出了与 CapSense 控制器一起使用的建议器件。

表 5-2. 针对 CapSense 建议的低电容 ESD 保护器件

ESD 保护器件		输入电容	漏电流	接触放电的最大容限	空气放电的最高容限
制造商	部件编号				
Littlefuse	SP723	5 pF	2 nA	8 kV	15 kV
Vishay	VBUS05L1-DD1	0.3 pF	0.1 µA <	±15 kV	±16 kV
NXP	NUP1301	0.75 pF	30 nA	8 kV	15 kV

5.3 电磁兼容性 (EMC) 注意事项

5.3.1 辐射干扰

辐射电能可能影响系统测量，并潜在地影响处理器内核的正常工作。这种干扰在 PCB 级别上进入 CapSense 控制器，穿过 CapSense 传感器走线，然后穿过其他任何数字和模拟输入。最大限度减小射频干扰作用的布局指南：

- **接地层**：提供 PCB 上的接地层。
- **串联电阻**：在 CapSense 控制器引脚 10mm 的范围内安装串联电阻。
  - 建议 CapSense 输入线路串联电阻值为 560 Ω。
  - 建议通信路线的串联电阻值为 330Ω，例如，I^2C 和 SPI。
- **走线长度**：如有可能，请最大限度减少走线长度。
- **电流回路面积**：最小化电流的回路路径。在传感器和走线 1 厘米以内提供开口而非实芯式接地，从而减小寄生性电容的影响。
- **RF 源位置**：分隔带噪声源的系统，例如，LCD 反相器和开关模式的电源 (SMPS) ，保持与 CapSense 输入的隔离 。屏蔽电源是另一种常见的防干扰技术。

5.3.2 辐射

要降低来自 CapSense 传感器的辐射，请选择低频率的开关式电容时钟。使用 Prescaler（预分频器）选项在固件中控制此时钟。提升预分频器的值可以降低开关时钟的频率

5.3.3 传导式抗噪声和抗辐射

传导式噪声为通过与其他系统的相交进入系统的噪声。这些相交包括电源和通信线路。由于 CapSense 控制器是低功耗器件，必须避免传导式辐射。以下指南将有助于减少传导辐射和抗扰性：

- 使用数据手册中建议的耦合电容器。
- 在系统电源输入端添加双向滤波器。这对于消除传导式辐射和噪声是非常有效的。PI 滤波器可以防止电源噪声影响敏感部件，同时防止部件的开关噪声反耦合到电源层。
- 如果 CapSense 控制器 PCB 通过电缆连接至电源，则最小化电缆线的长度，并考虑使用屏蔽电缆。
- 您可以在电源或通信线路周围放置一个铁氧体磁珠以用来过滤高频噪声。

5.4 软件滤波

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

表 5-3. CapSense 滤波器表

类型	说明	应用
一般	有限脉冲响应滤波器（无反馈），具有相同的加	电源的周期性噪声

	权系数	
IIR	无限脉冲响应滤波器（有反馈），步响应类似于 RC 滤波器	高频白噪声（1/f 噪声）
中位数	非线性滤波器计算来自于缓冲区（大小为 N）的中位数输入值	来自于电机和开关电源的噪声尖峰
抖动	非线性滤波器根据上一次输入来限制电流输入	来自较厚外覆层的噪声 (SNR < 5:1)，特别适用于滑条质心数据
基于事件的	非线性滤波器导致一种预定义响应, 该响应因观察到传感器数据中的模式产生	通常在非触摸事件中使用，以阻止 CapSense 数据传输
基于规则的	非线性滤波器导致一种预定义响应, 该响应因观察到传感器数据中的模式产生	通常在触摸表面正常操作过程中使用，以响应特殊情况，例如，意外选择多个按键

可以在这里下载使用上述列出的所有滤波器的代码示例。

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

表 5-4. RAM 和闪存要求

滤波器类型	滤波器的阶位	RAM（每个传感器的字节）	闪存（字节）ImageCraft 编译器	闪存（字节）ImageCraft 专业版编译器
一般	2-8	6	675	665
IIR	1	2	429	412
	2	6	767	622
中位数	3	6	516	450
	5	10	516	450
关于原始计数的抖动滤波器	不可用	2	277	250
关于滑条质心的抖动滤波器	不可用	2	131	109

可以在这里下载使用上述列出的所有滤波器的代码示例。

5.5 功耗

5.5.1 系统设计建议

对于许多设计而言，最小化功耗是重要目标。有多种减少 CapSense 电容式触摸感应系统功耗的方法。

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

除这些方法外，应用睡眠-扫描方法也非常有效。

5.5.2 睡眠-扫描方法

在典型应用中，CapSense 控制器不需要经常保持活动状态。该器件可以进入睡眠状态，以停止器件的 CPU 和主要模块。睡眠状态下消耗的电流比活动状态下少很多。

使用以下公式计算该器件长时间运行所消耗的平均电流。

$$I_{AVE} = \frac{(I_{Act} \times t_{Act}) + (I_{Slp} \times t_{Slp})}{T}$$ 公式 10

其中：

I_{ACT} = 有功电流

T_{ACT} = 活动时间

I_{Slp} = 睡眠电流

t_{Slp} = 睡眠时间

T = 总计时间周期

器件的平均功耗计算如下：

$$P_{AVE} = V_{DD} \times I_{AVE}$$ 公式 11

其中：

V_{DD} = 电源电压

I_{AVE} = 平均电流

基于 Excel 的平均功耗计算器。单击这里下载计算器。

5.5.3 响应时间与功耗

如公式 12 所述，可以通过降低 I_{AVE} 或 V_{DD} 来降低平均功耗。I_{AVE} 可以通过延长睡眠时间来降低。 延长睡眠时间到较高值将导致 CapSense 按键响应时间差。因此，睡眠时间须基于系统需求进行设定。

在任何应用中，如果功耗和响应时间均被视为重要参数，则可以使用优化方法，即结合应用持续性扫描和睡眠扫描模式。 在此方法中，器件大部份时间在睡眠-扫描模式下：器件扫描传感器， 然后定期进入睡眠模式， 因此功耗较少（如上一节所述）。 当用户触摸传感器以操作系统时， 则器件跳至持续扫描模式: 传感器作连续扫描而不进入睡眠模式，提供较佳响应时间。在指定期间内，器件仍然处于持续扫描模式。如果在超时周期内，用户未对传感器进行操作，则器件会切换回睡眠扫描模式。

5.5.4 测量平均功耗

下列说明介绍在使用睡眠-扫描方法时如何确定平均功耗：

1. 构建一个扫描所有传感器而不进入睡眠模式（持续扫描模式）的项目。内含扫描传感器前代码引脚切换功能。切换输出引脚的状态用作时间标记，可以通过示波器进行跟踪。

2. 下载项目到 CapSense 器件中，测量当前功耗。将测量好的电流分配给 I_{ACT}。

3. 从数据表中获取睡眠电流信息，并分配给 I_{SLP}。

4. 使用示波器监测输出引脚切换，测量两次切换之间的时间期间。这给出活动时间。将此数值分配给 t_{ACT}。

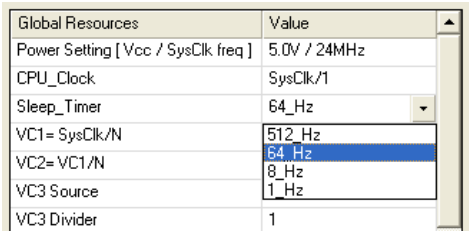
5. 对项目应用睡眠-扫描。通过在全局资源窗口中选择睡眠定时器频率来设置睡眠-扫描周期的时间周期 T， 如图 5-1 所示。

6. 从睡眠-扫描周期时间期间中减去活动时间，得出睡眠时间，$t_{Slp} = T - t_{Act}$。

7. 使用公式 11 计算平均电流。

8. 使用公式 12 计算平均功耗。

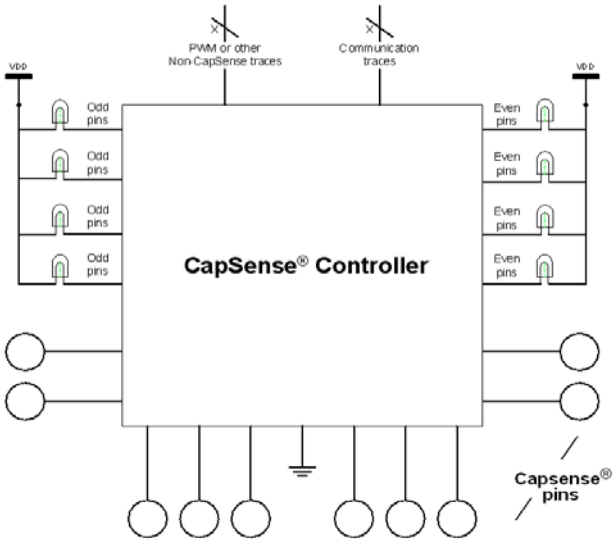
图 5-1. 全局资源窗口



5.6 引脚分配

减少 CapSense 传感器走线和通信与非 CapSense 走线之间交互作用的有效方法是通过端口分配来进行互相隔离。图 5-2 显示对 32 引脚 QFN 封装进行此隔离的基本版本。由于隔离了各个功能，定向了 CapSense 控制器中通信、LED 和感应走线之间不再存在交叉作用。

图 5-2. 建议：通信、CapSense 和 LED 的端口隔离



CapSense 控制器架构需要奇数和偶数端口引脚数的电流预算。如果奇数端口引脚的电流预算为 100 mA， 那么所有奇数端口引脚所抽取的总电流不应超过 100 mA。 除电流预算限制外， 每个端口引脚也有最大的电流限制， 相关定义见 “CapSense 控制器数据手册”这部分。

所有 CapSense 控制器均提供高灌电流和源电流能力端口引脚。使用来自端口引脚的高灌电流或源电流时，应使用距离器件接地引脚最近的端口，以最小化噪声。

5.7 PCB 布局准则

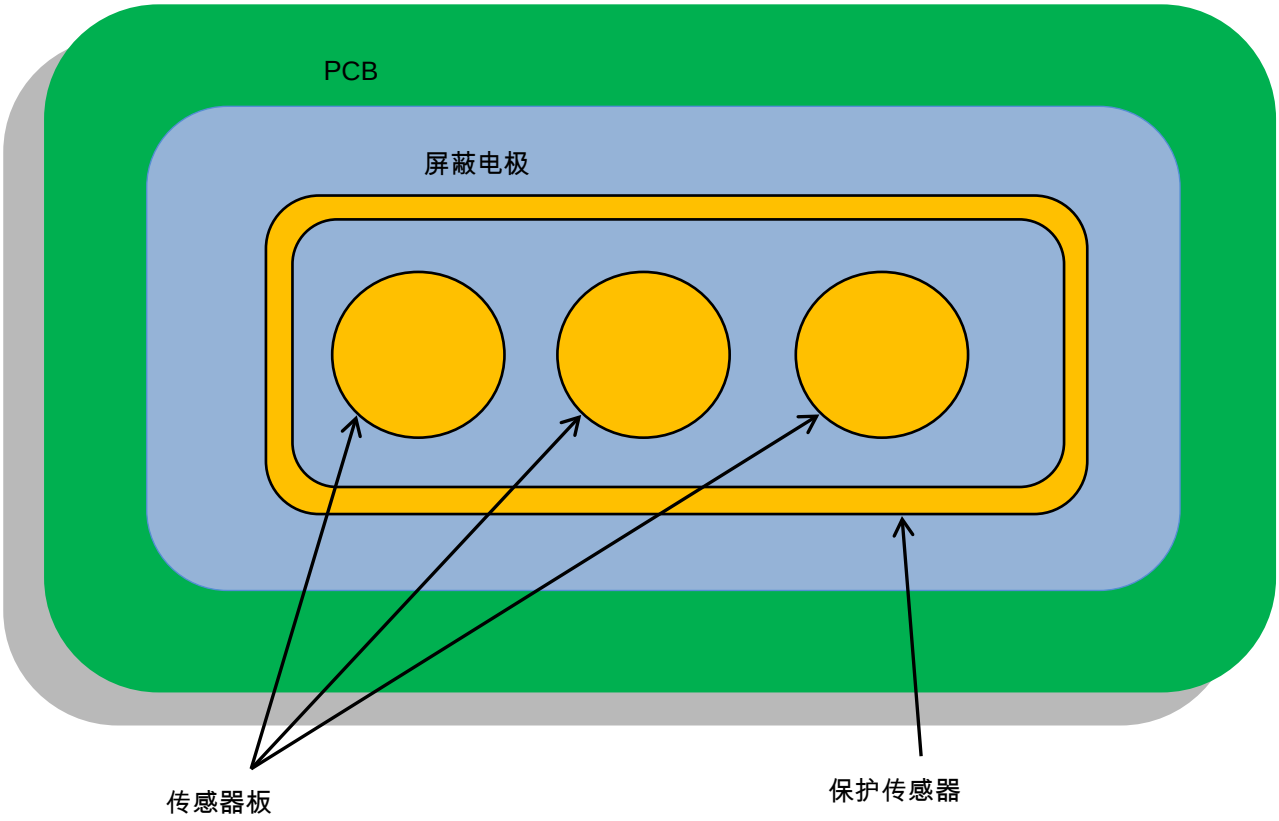
关于 PCB 布局指南的详细内容，请参见《CapSense 入门》。

6. 耐水性

某些 CapSense 电容式触摸感应系统应用需要在带有水份环境进行可靠的工作。白色家电、 汽车应用和工业应用均涉及水、冰和湿度变化的环境，传感器系统必须在这种环境下完成工作。对于这样的应用环境而言，屏蔽电极和防护传感器可以提供强健的触摸感应系统。

6.1 屏蔽电极和保护传感器

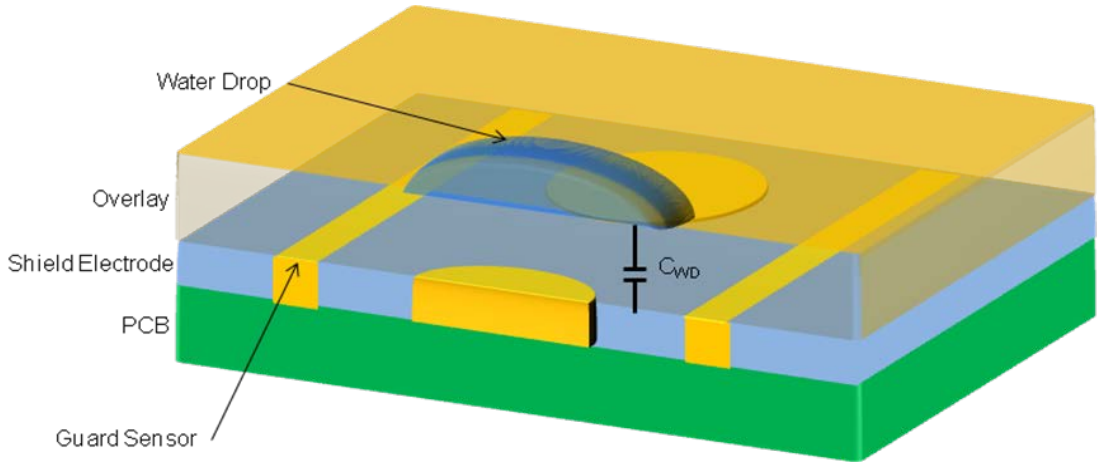
图 6-1. 屏蔽电极和保护传感器的 PCB 布局



6.1.1 屏蔽

屏蔽电极可以保护 CapSense 按键传感器避免检测由水滴引起的假触摸。当水滴落到外覆层表面上时，屏蔽电极和传感器板之间的耦合增加了 C_{WD}，如图 6-2 所示。

图 6-2. 带水滴的电容测量



- C_{WD} – 水滴与屏蔽电极之间的电容

屏蔽电极的目的是在触摸传感器周围建立电场，用来消弱水滴产生的影响。屏蔽电极的工作方式是将触摸传感器的电压镜像到屏蔽上。

遵循这些准则以确保屏蔽电极正常工作：

- 原理图
- 布局
- 固件开发

6.1.1.1 原理图

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

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

6.1.1.2 布局

遵循布局准则，这在“CapSense 入门”中给出。

6.1.1.3 固件开发

步骤 1：可以从其中一个备用数字行总线（Row_0_Output_1 - Row_0_Output_3）中选择屏蔽电极信号源。

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

步骤 2：路由屏蔽电极输出到屏蔽引脚。执行下列步骤：

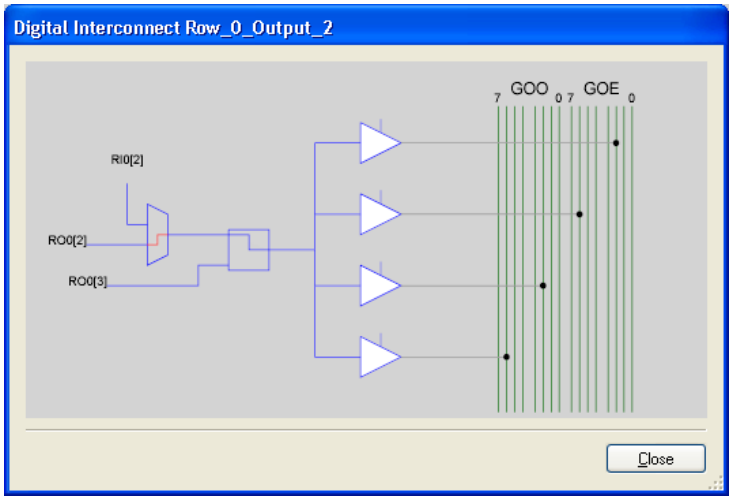
1. 选择屏蔽电极输出。

2. 将行输出信号路由到全局输出奇数/偶数。

3. 连接全局输出奇数/偶数到端口引脚。例如，如果选择 P0[2] 作为屏蔽引脚：从用户模块属性中，选择屏蔽电极输出（在 Row_0_Output_2 上）。

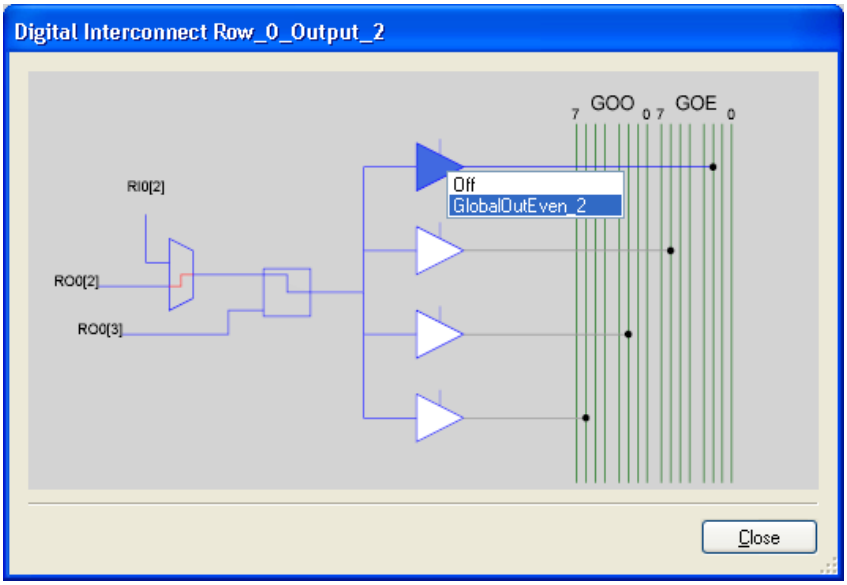
4. 单击 Row_0_Output_2，打开数字互连视图，如图 6-3 所示。

图 6-3. 数字互连原理图

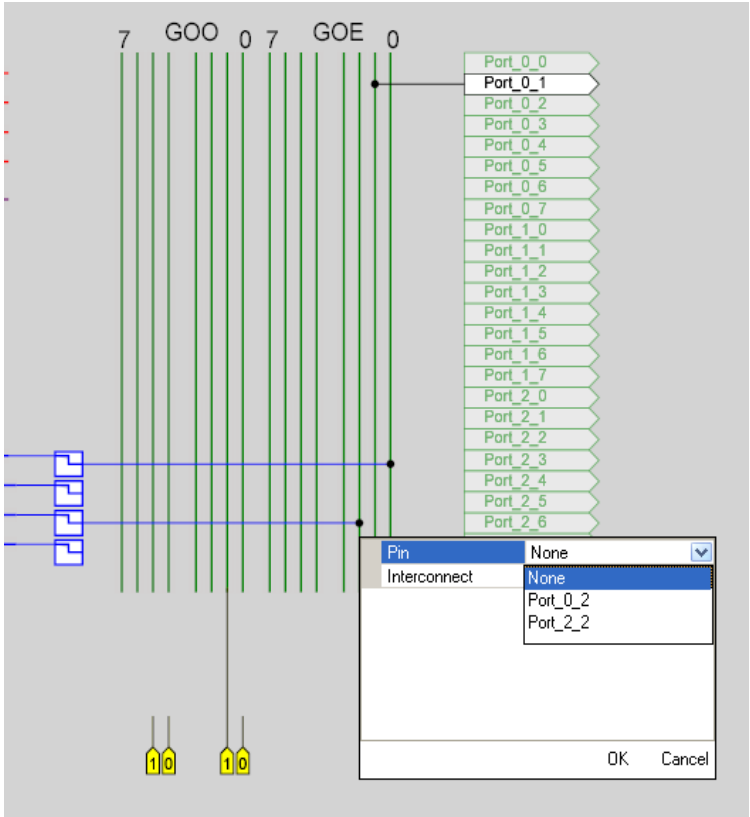


5. 选择 Row_0_Output_2_Drive_0，然后再选择 GlobalOutEven_2，如图 6-4 所示。

图 6-4. 输出选择原理图

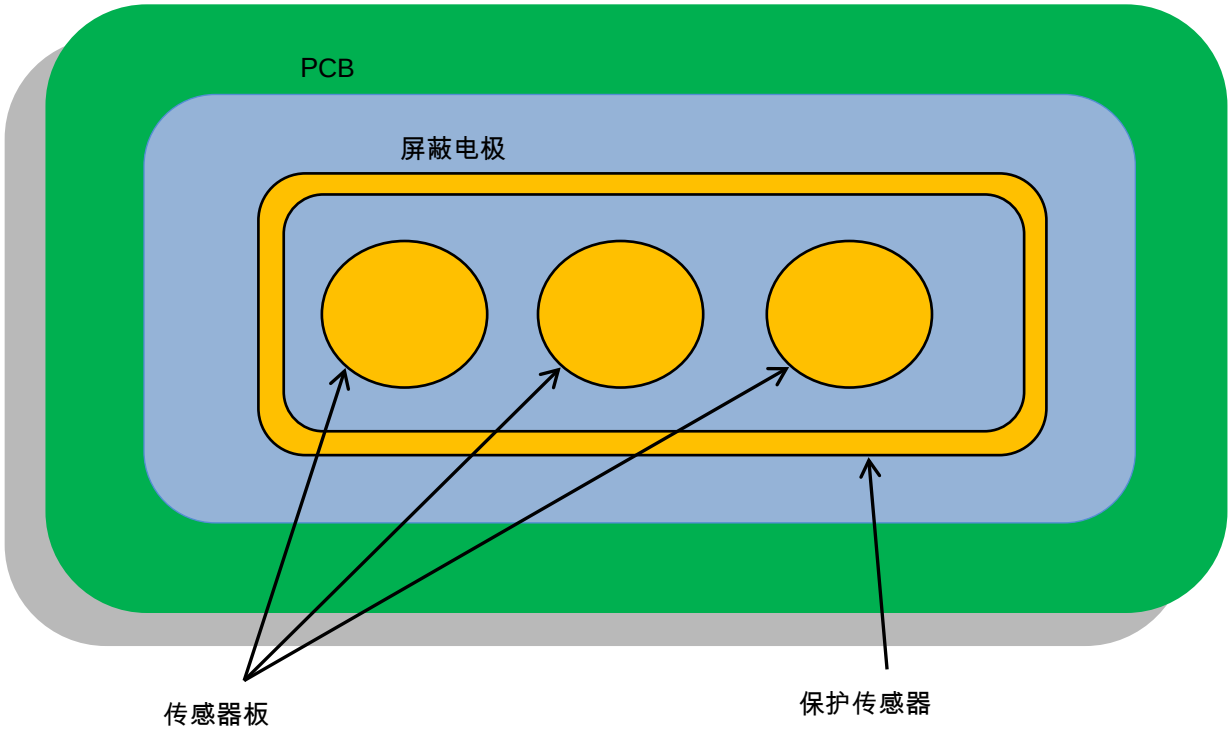


6. 单击 GlobalOutEven_2，然后在 Pin（引脚）选项中选择 P0{2]，如下图所示。



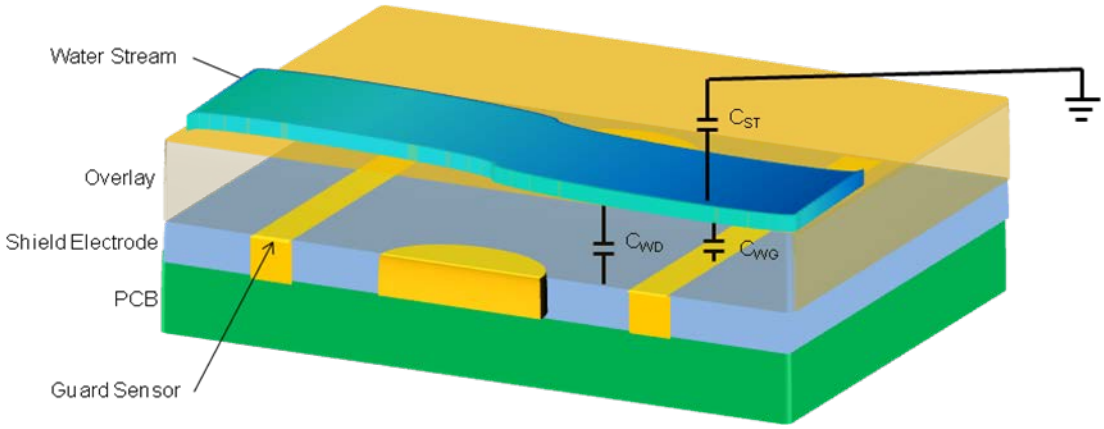
6.1.2 保护传感器

图 6-5. 屏蔽和保护传感器的 PCB



保护传感器是铜走线，如图 6-5 所示，该走线环绕 PCB 上的所有传感器，用来检测是否出现连续水滴。当水滴出现在感应表面时，在系统中将添加较大电容 C_{ST}，如图 6-6 所示。此电容可能比 C_{WD} 大数倍。正因为如此，屏蔽电极的影响被完全掩模，通过传感器测量的原始计数将与手指触值相同，甚至更高。在这种情况下，保护传感器将有所帮助；当检测水流时，该传感器将防止其他传感器触发。

图 6-6. 带水流的电容测量



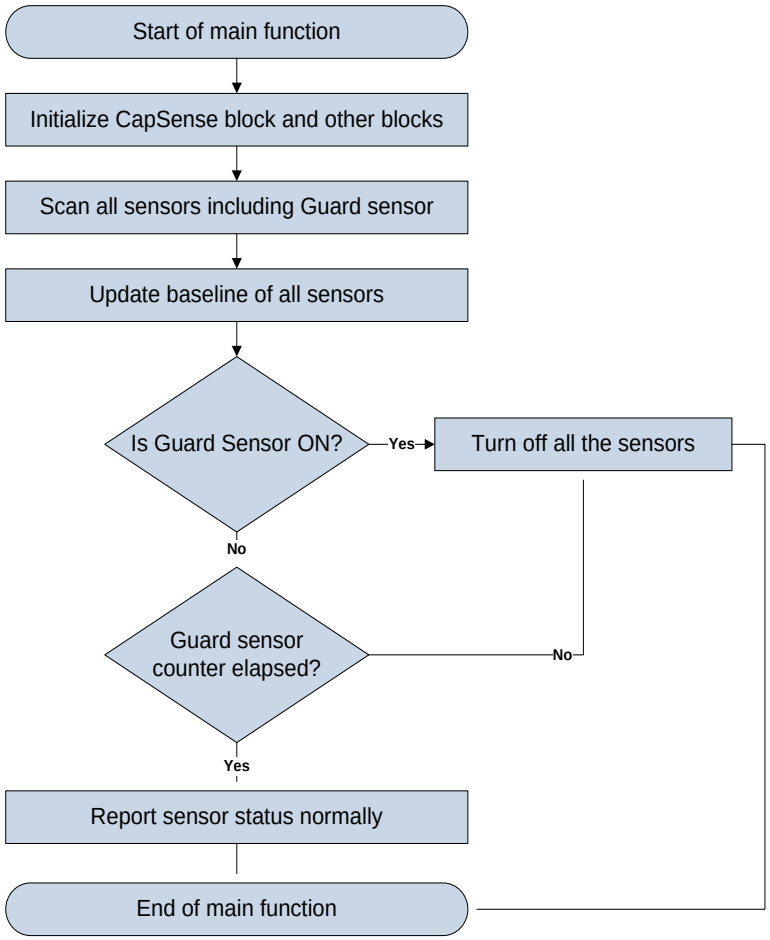
- C_{WD} - 水流与屏蔽电极之间的电容。
- C_{ST} - 水流与系统接地之间的电容。
- C_{WG} - 水流与保护传感器之间的电容

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

当水流进入触摸板时，保护性传感器检测到这个事件后，禁用触摸传感器的处理逻辑。附加保护性传感器的“死”时间阻止传感器被过早解锁。水流消失时，保护性计数器会短时间暂时抑制触摸处理系统（保护性传感器计数器）。这就消除了遗留在电路板上的水流触发的假触摸检测。

图 6-7 是有关如何在固件中实施保护性传感器的流程图。

图 6-7. 实施保护性传感器的流程图



6.2 设计建议

以下系统级和 PCB 布局建议适用于暴露于水份的 CapSense 系统。

- 屏蔽电极铜填充建议：
  - 上层 – 7 mil 走线和 45 mil 栅格（填充 15%）
  - 下层 – 7 mil 走线和 70 mil 栅格（填充 10%）
  - 按键之间的屏蔽电极应至少为 10 毫米宽
  - 只有传感器板和 CapSense 的周围区域应接地
- 传感器表面垂直放置，或与平面成一个角度，这样，水流自然滑离传感器，而不会积存大量水滴。
- 使用防水和不吸水的外覆层材料。这样可以最大程度地减少器件面板上的水纹和薄膜。这特别适用于水具有较高导电性这种情况，例如，海水。
- 如果在连续水流的环境下应用，必须使用保护性传感器。如果器件仅暴露于阴雨条件下，无需使用保护性传感器。

7. 近程传感

近程传感器在接触电容性触摸表面前检测到手指或其他导电性物体。设想在黑暗中伸出手去操作汽车音响系统。近程检测可以在手指的作用下使系统点亮。例如，当用户将手靠近时，音响系统的按键将点亮背光灯 LED。

7.1 近程传感器的类型

7.1.1 按键

具有较大 C_P 和较小差值计数的按键可以用作近程传感器。实施为按键的近程传感器的灵敏度相当高，远远超过了常规电容式触摸感应按键的灵敏度。

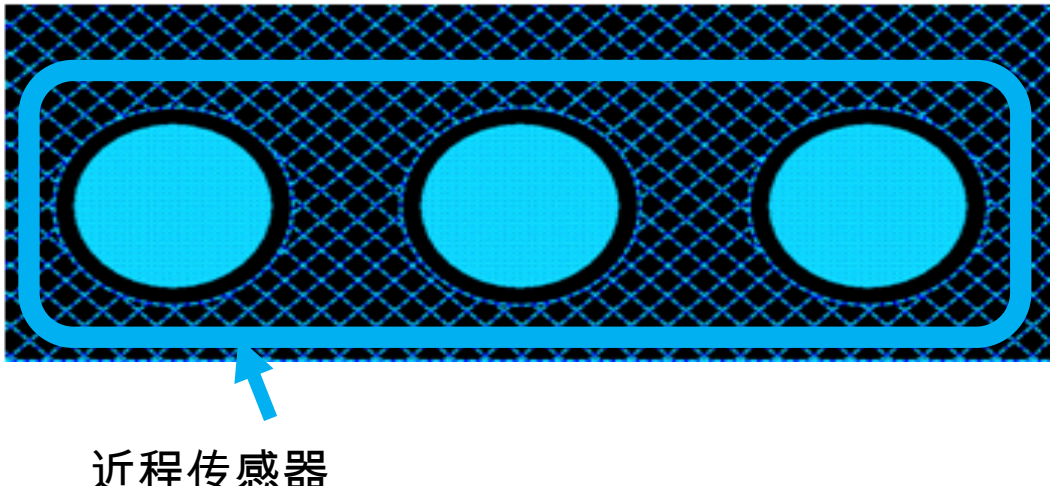
7.1.2 导线

一根导线段也可以用作良好的近程传感器。由于检测手指依赖于随电场变化而变化的电容，任何杂散电容或物体对导线周围的电场产生影响时，均会影响近程传感器的感应范围。由于制造成本和复杂性相对较高，使用导线传感器对于批量生产而言并不是理想的解决方案。

7.1.3 PCB 走线

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

图 7-1. 使用 PCB 走线的近程传感器



7.1.4 传感器机械连接

实施近程传感器的另一种方法是机械性连接多个传感器。实施方法是将多个传感器组合到一个大型传感器中，使用固件将传感器连接到 PSoC Designer 内部模块复用器总线上。使用此方法时，请注意不要超过 C_P 设计限制。

7.2 设计建议

使用屏蔽电极有效延长近程传感器的检测距离。当近程传感器周围存在金属物体时，这种方法特别有用。

导线传感器使屏蔽电极的作用得到更大的发挥，因为它可以较大程度地远离屏蔽电极。

避免近程传感器内部较大、坚实的地线填充区域，因为这样会降低灵敏度。

扫描速度越慢，所有距离的灵敏性越高。对于近程传感器而言，扫描速度应该非常低。

8. 低功耗设计要素

功耗是微控制器设计的重要方面。有效降低 CapSense 控制器使用的平均电流有多种技术，其中，睡眠模式的使用最普遍。在不需要执行任何功能时，CapSense 控制器使用睡眠模式，这类似于手机背光灯在空闲期间变暗这种情况。使用该模式，可以降低器件消耗的平均电流，所有电池应用所必须。CapSense 控制器通过向 CPU_SCR0 寄存器（位 3）中的 SLEEP（睡眠）位写入‘1’，便进入睡眠模式。通过调用 M8C_Sleep 宏，可以完成此功能。进入睡眠模式后，中央 CPU 停止运行，内部主振荡器 (IMO) 被禁用，带隙电压参考关闭，闪存存储器模型也被禁用。保持工作的唯一电路是电源电压监视器和 32-kHz 的内部振荡器。除睡眠模式以外的节省功耗技术：

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

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

8.1 其他功耗节约技术

除睡眠模式外，所有功耗节约技术均是基于应用和技术，并且，有些技术可能造成不良后果。后面章节将详细讨论每种技术。

ABF_CR0 &= 0xc3; // Buffer Off

8.1.1 将驱动模式设置为模拟 HI-Z

CapSense 控制器驱动模式的状态可能影响功耗水平。仅改变不会对系统造成负面影响的引脚上的驱动模式。按顺序进行更改，这样可以避免产生线路故障。此序列取决于引脚电流驱动模式和端口数据寄存器的状态。通过 CapSense 控制器驱动模式结构，在 HI-Z 或强驱动模式之间切换开关时，引脚必须暂时处于电阻上拉或电阻下拉驱动模式。临时性驱动模式与引脚上的上一个值相反。因此，如果引脚是高驱动，那么临时驱动模式必须是电阻下拉式。这就确保引脚的驱动模式不是电阻驱动模式，排除一切可能产生的故障。

进入睡眠模式之前，在软件中手动设置驱动模式。共有三个寄存器，分别是 PRTxDM0、PRTxDM1 和 PRTxDM2，可用于控制驱动模式。每个寄存器向一个引脚分配一个位。这样，要更改单个引脚的驱动模式，需要更改三个寄存器的

写入位。然而，由于整个端口由三个寄存器相同的写入位来进行更改，因此这一步很容易实现。模拟 HI-Z 的正确位模式为 110b。使用以下代码，通过调至电阻下拉模式，将端口 0 从强驱动模式转换为 HI-Z 模拟。

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

8.1.2 全部放在一起

下列代码是 28 位引脚端口典型睡眠准备序列的样本。在这个顺序，中断被禁用，模拟电路被关闭，所有驱动模式均设置为 Analog HI-Z，中断被重新启用。

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

8.1.3 睡眠模式复杂性

CapSense 控制器可以从复位或通过中断的方式退出睡眠模式。CapSense 控制器中有三种复位方式：外部复位、看门狗复位和加电复位。这三种复位方式中的任何一种均可以使 CapSense 控制器退出睡眠模式，一旦复位无效后，CapSense 控制器便开始执行以 *Boot.asm* 开头的代码。唤醒 CapSense 控制器可以使用的中断模式：睡眠定时器、低电压监视器、GPIO、模拟列和异步。使用中断模式唤醒 CapSense 控制器或在睡眠时尝试数字通信时，会提高睡眠模式的复杂性。以下章节将对这些注意事项进行详细说明。

8.1.4 待处理中断

如果待处理、启用和计划在写入 CPU_SCR0 寄存器睡眠位之后处理中断，系统将拒绝进入睡眠模式。仍然执行指令，但 CapSense 控制器没有设置睡眠位。相反，中断被服务，这样将导致 CapSense 控制器完全忽略睡眠指令。为避免这一问题，在睡眠模式准备期间，中断需全局禁用，并且在写入睡眠位之前重新启用。

8.1.5 启用全局中断

要从中断中唤醒 CapSense 控制器，无需启用 Global Interrupt Enable（启用全局中断）寄存器 (CPU_F)。唯一通过中断唤醒睡眠模式，需使用 INT_MSKx 寄存器中的正确中断掩模，如下例所述。如果全局中断被禁用，唤醒 CapSense 控制器的 ISR （中断服务子程序） 并未执行，但 CapSense 控制器仍可退出睡眠模式。

在此情况下，您必须手动清除待处理中断，或启用全局中断以支持 ISR 执行服务。中断在 INT_CLRx 寄存器内清除。

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

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

8.2 唤醒后执行序列

如通过复位唤醒 CapSense 控制器，程序会由引导代码起端开始执行。如果通过中断服务例程唤醒 CapSense 控制器，首先执行的指令是直接跟在睡眠指令后面的那个指令。这是因为 CapSense 控制器完全进入睡眠模式之前，预先提取睡眠指令后面的第一个指令。因此，如果全局中断被禁用，在睡眠开始前中断的指令将继续执行。

8.2.1 启用 PLL Mode（PLL 模式）

如果启用 PLL 模式，CPU 在进入睡眠模式前，频率必须降至最小值 3 MHz。这是因为 PLL 在 CapSense 控制器唤醒及重启后企图重新锁定时经常过冲。此外，您应在唤醒后等待 10 ms，然后开始让 CPU 正常工作，从而确保正确执行。这表示使用睡眠模式和 PLL 模式时，软件必须能够在频率为 3 MHz 下继续执行。对 OSC_CR0 寄存器写入可以降低 CPU 的速度。然而，此寄存器仅设置 SYSCLK 分频器，这意味着 CPU 速度将由于 部件系列的不同 SYSCLK 而有所不同。通常情况下，SYSCLK 为 24 MHz

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

8.2.2 执行 Global Interrupt Enable（全局中断启用）

不建议在写入睡眠位的指令边界上中断。假设从中断 (reti) 指令返回时执行睡眠命令，这可能导致所有进入睡眠固件准备被跳过。要避免这种情况，请暂时禁用中断后，再准备进入睡眠，然后在进入睡眠前再重新启用中断。由于全局中断指令的时序要求，中断不会发生在下一指令执行时，也就是设置睡眠位时。

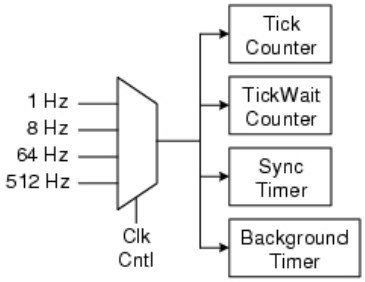
8.2.3 I^2C 从器件处于睡眠模式

在睡眠模式下使用 I^2C 从器件有许多复杂情况。由于在睡眠过程中，IMO 和 CPU 关闭，CapSense 控制器不再执行任何处理。I^2C 寻址会产生这些问题。当将 I^2C START 条件发送到特定地址时，CapSense 控制器无法处理该地址，因此发出 NAK 响应。典型的解决办法是在 I^2C 总线的时钟或数据线路上设置下降沿中断。然后，主控向其发送虚拟的 START（启动）条件以唤醒 CapSense 控制器。唤醒与实现处理 I^2C 寻址之间有一大段延迟时间，因此主控需要推迟高达 200 µs 的时间后，再执行下一次发送或继续发送直到 ACK 收到为止。此解决方案还存在另一个问题，那就是 CapSense 控制器将在任何 I^2C 下降沿流量拥堵时被唤醒，这将导致更长的总活动时间和更高的睡眠电流。另一种解决方案是，使用第三个通用 I/O 引脚，唤醒 CapSense 控制器，并在适当的延时后发送初始 START 条件。

8.2.4 睡眠定时器

CapSense 控制器提供睡眠定时器和睡眠定时器用户模块。当 CapSense 控制器处于睡眠模式时，可以使用这些器件，并执行类似功能。实际的睡眠定时器在内部低速振荡器关闭时支持，并且从不关闭。可选频率间隔为 1 Hz、8 Hz、64 Hz 和 512 Hz 时，定时器发生一次中断。定期唤醒 CapSense 控制器执行某些处理或检查活动非常有用。例如，定期唤醒以便扫描传感器。睡眠定时器用户模块使用睡眠定时器来生成某些额外功能。这些功能包括后台钟表计数器（用来生成定期中断）、程序环路延迟功能、可变倒计时计数器和控制回路时间的回路控制器。此功能的简单的框图如图 8-1 所示。

图 8-1. 睡眠定时器用户模块框图



9. 资源

9.1 网站

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

CY8C21x34/B Web 页面包含适用于 CY8C21x34/B 器件系列的各种技术资源。

9.2 数据手册

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

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

9.3 技术参考手册

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

- CY8CPLC20、CY8CLED16P01、CY8C29x66、CY8C27x43、CY8C24x94、CY8C24x23、CY8C24x23A、CY8C22x13、CY8C21x34、CY8C21x23、CY7C64215、CY7C603xx、CY8CNP1xx 和 CYWUSB6953 PSoC(R) Programmable 片上系统技术参考手册 (TRM)

9.4 开发套件

9.4.1 通用的 CapSense 控制器套件

通用的 CapSense 控制器套件的主要功能是预定义的控制电路和插入硬件，用来简化原型设计和调试系统。程序设计和 I^2C-to-USB 桥接器硬件也包含在内，以便进行调校和数据采集。

- CY3280-BK1 通用的 CapSense 控制器

9.4.2 通用的 CapSense 模块电路板

9.4.2.1 简单的按键模块电路板

CY3280-BSM 简单的按键模块由 10 个 CapSense 按键和 10 个 LED 组成。该模板与任意 CY3280 通用 CapSense 控制器板连接。

9.4.2.2 阵列按键模块电路板

CY3280-BMM 阵列按键模块由以下元件组成：8 个 LED、8 个 CapSense 传感器，组织格式为 4×4 阵列，从而构成 16 个物理按键。该模板与任意 CY3280 通用 CapSense 控制器板连接。

9.4.2.3 线性滑条模块电路板

CY3280-SLM 线性滑条模块由以下元件组成：5 个 CapSense 按键、1 个线性滑条（带有 10 个传感器）和 5 个 LED。该模板与任意 CY3280 通用 CapSense 控制器板连接。

9.4.2.4 辐射滑条模块电路板

CY3280-SRM 辐射滑条模块由以下元件组成：4 个 CapSense 按键、1 个辐射滑条（带有 10 个传感器）和 4 个 LED。该模板与任意 CY3280 通用 CapSense 控制器板连接。

9.4.2.5 通用的 CapSense 原型模块

CY3280-BBM 通用的 CapSense 原型模块提供了对路由至连有控制器的电路板上 44 引脚连接器的各个信号的连接。原型模块电路板与通用的 CapSense 控制器电路板一起用来实施其他功能，而将其作为其他单用途通用 CapSense 模块电路板的组成功能

9.4.3 线上仿真 (ICE) 工具包

ICE 转接板是 CY3215-DK 线上仿真器提供原型系统或 PCB 中的 PSoC 目标器件之间的内部互接，方法是通过柔性线缆或封装特定的转接板支脚。以下转接板可用：

- CY3250-20334QFN 线上仿真 (ICE) 转接板套件，用于调试 QFN CY8C21x34 PSoC 器件

9.5 PSoC 编程器

PSoC Programmer 是用来编程 PSoC 器件的灵活的集成式编程应用程序。PSoC Programmer 可以与 PSoC Designer 和 PSoC Creator 一起用来编程任何设计到 PSoC 器件上。

PSoC Programmer 提供 API 硬件层，用来设计使用编程器和桥器件的特定应用程序。在 COM 指导文档中包含 PSoC Programmer 硬件层的详细描述及以下所有语言的示例代码：C#、C、Perl 和 Python。

9.6 MultiChart

MultiChart 是用于实时 CapSense 数据查看和记录的简易的 PC 工具。该应用允许您查看 48 个传感器上的数据，保存和打印图表及将数据保存到试算表以备后期分析使用。

9.7 PSoC Designer

赛普拉斯提供独有的集成式设计环境 PSoC Designer。使用 PSoC Designer，您可以配置模拟和数字模块、开发固件和调校设计。在拖放式设计环境中使用预先设定的模拟和数字功能库（包括 CapSense）来开发你的应用程序。PSoC Designer 附带内置的 C 语言编译器和嵌入式编程器。现已推出用于复杂设计的专业版编译器。

9.8 代码示例

赛普拉斯提供多个示例代码以便实现您的设计并快速运行。

- CY8C21x34 器件上的 CapSense (CSD) 的功耗 - EP65489
- CY8C21x34 上带有 EzI2Cs 从器件的 CSD 软件滤波器
- 连接 PSoC 至 SPI EEPROM 接口
- CY8C21x34 上带有 EzI2Cs 从器件的 CSD
- 伪随机序列发生器
- CY8C21x34 上带有 TX8 的 CSD
- CY8C21x34 上带有 I2CHW 从器件的 CSD
- CMPPRG 用记模块示例
- 使用 CYRF6936 和 CY8C27643 接收字符串
- CY8C21x34 上 CapSense Sigma Delta (CSD) 带 LED 背光渐暗
- CY8C21x34 上的 CSD 带 SPIS

9.9 设计支持

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

- 知识库文章 – 参见产品系列的技术文章或搜索各种 CapSense 相关主题。
- CapSense 应用笔记 – 参见本文件中说明信息所涉及的广泛应用笔记。
- 白皮书 - 您可了解高级电容式触摸界面相关主题。
- 赛普拉斯开发者社区 - 您可与赛普拉斯开发者社区人员联系，交流信息。
- CapSense 产品选择器指南 – 参见赛普拉斯 CapSense 产品线完整的产品系列。
- 视频库 – 使用视频教程提高学习速度
- 质量和可靠性 – 赛普拉斯致力于实现客户满意度。在我们的“质量”页面上，您可找到可靠性与产品质量鉴定的相关报告。
- 技术支持 – 世界级在线技术支持。