AN66269 - CY8C20x34

CapSense®设计指南

文档编号：001-79112 版本*D

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

版权所有

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

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

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

目录

1. 简介 5

1.1 摘要 5

1.2 赛普拉斯的 CapSense 文档体系 5

1.3 CY8C20x34 CapSense 系列特性 7

1.4 文档规范 8

2. CapSense 技术 9

2.1 CapSense 基本原理 9

2.2 CY8C20x34 中的 CapSense 方法 11

3. CapSense 设计工具 13

3.1 概况 13

3.2 用户模块概述 15

3.3 CapSense 用户模块全局阵列 15

3.4 CSA_EMC 用户模块参数 16

4. 使用用户模块来调试 CapSense 性能 21

4.1 基本注意事项 21

4.2 CSA_EMC 用户模块 — 调校指南 22

4.1 建议的 C_{INT} 值 23

4.2 测量传感器寄生电容 C_P 23

4.3 估算 CSA_EMC 时钟 24

4.4 设置建立时间 24

4.5 监测 CapSense 数据 25

4.6 提高信噪比的方法 25

4.7 调校示例 25

5. 设计的注意事项 32

5.1 覆盖层选择 32

5.2 ESD 保护 33

5.3 电磁兼容性（EMC）注意事项 34

5.4 软件滤波 35

5.5 功耗 36

5.6 引脚分配 37

5.7 PCB 布局指南 38

6. 资源 39

6.1 网站 39

6.2 数据手册 39
6.3 技术参考手册 39
6.4 开发套件 39
6.5 样本电路板文件 40
6.6 PSoC Designer 42
6.7 PSoC Programmer 42
6.8 CapSense 数据查看工具 42
6.9 代码示例 42
6.10 设计支持 43
术语表 44
文档修订记录 49
文档修订记录 49

1. 简介

1.1 摘要

本文档为了使用 CapSense 控制器 CY8C20x34 系列实现电容式触摸感应（CapSense®）功能提供了设计指南。本指南包含以下主题：

- CapSense 器件 CY8C20x34 系列的特性
- CapSense 工作原理
- CapSense 设计工具简介
- 调试 CapSense 触摸感应系统以获得最佳性能的深入指导
- CapSense 逐步调试示例
- 有关 CapSense 电气和机械系统设计的注意事项
- 在系统中设计 CapSense 所使用的附加资源和支持

1.2 赛普拉斯的 CapSense 文档体系

图 1-1 和表 1-1 汇总了赛普拉斯 CapSense 文档体系。利用这些资源，可快速获得成功完成 CapSense 产品设计所需的信息。图 1-1 给出了电容式感应系统产品设计周期的典型流程。本指南中的信息与突出显示为绿色的主题关联性最大。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. 设计周期流程

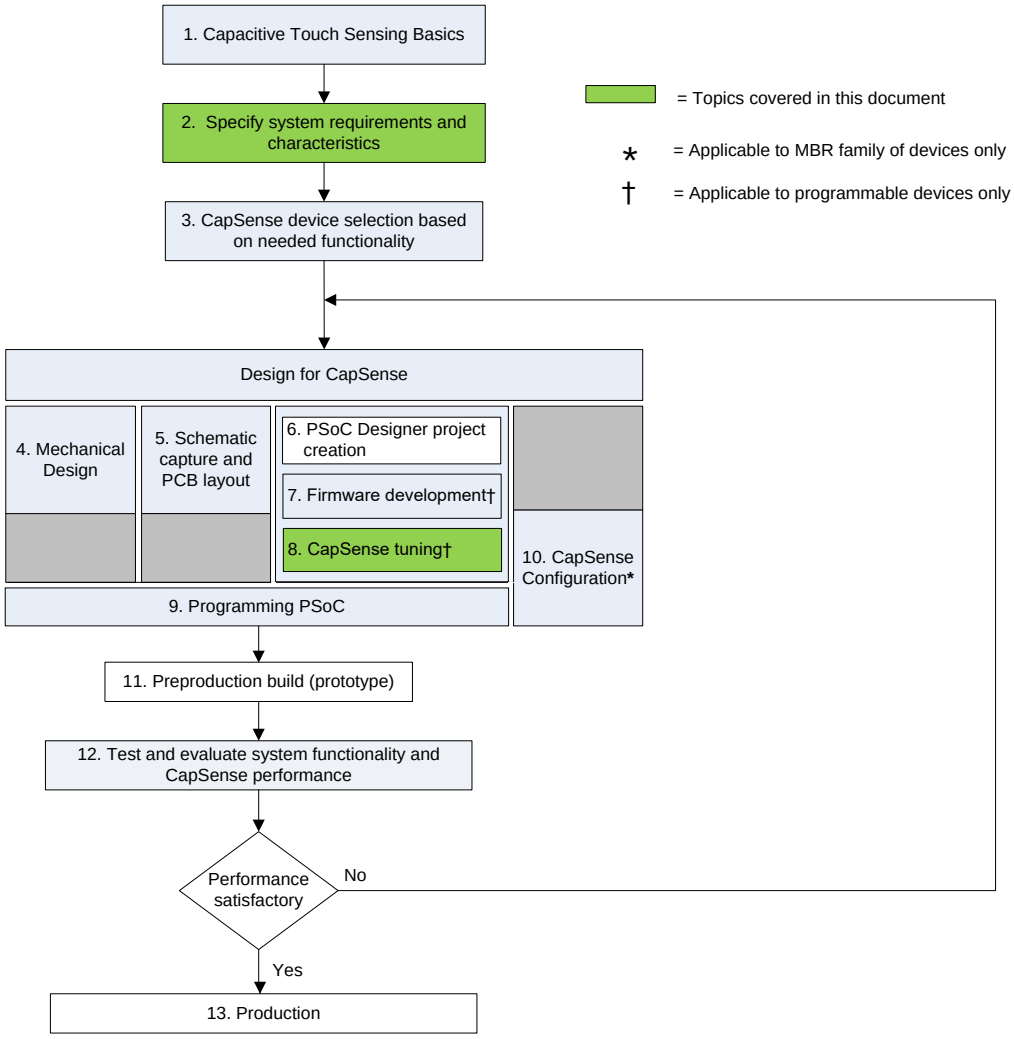


表 1-1. 与图 1-1 中每个已编号的任务相对应的赛普拉斯文档

图 1-1 中已编号的设计任务	相关的赛普拉斯 CapSense®文档
1	■ CapSense®入门
2	■ CapSense®入门 ■ CY8C20x34 CapSense®器件数据手册 ■ PSoC 系列特定的 CapSense®设计指南（本文档）
3	■ CapSense®入门
4	■ CapSense®入门
5	■ CapSense®入门
6	■ PSoC Designer 用户指南

图 1-1 中已编号的设计任务	相关的赛普拉斯 CapSense®文档
7	■ 汇编语言用户指南 ■ C 语言编译器用户指南 ■ CapSense 代码示例 ■ PSoC 系列特定的技术参考手册（适用于 CY8C20x34）
8	■ 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（CSA_EMC） ■ CapSense 数据查看工具 — AN2397 ■ CapSense 控制器代码示例设计指南
9	■ 编程器用户指南 ■ MiniProg3 用户指南
11	■ 代码示例

1.3 CY8C20x34 CapSense 系列特性

Cypress CY8C20x34 是低功耗、高性能、可编程的 CapSense 控制器系列，其特性如下：

1.3.1 高级触摸感应功能

- 可编程电容式感应元件
  - 支持一组使用 CSA_EMC 电容式感应技术的 CapSense 按键、滑条和接近感应传感器
  - 包含用于实施按键和滑条的集成式 API
  - 支持多达 25 个电容式传感器和 6 个滑条
  - CY8C20x24 支持 25 个电容式传感器和 1 个滑条
  - 支持长达 2 厘米的接近感应距离（使用板上 PCB 走线）
- 增强型抗噪能力
  - CSA_EMC 为传导和辐射噪声情况较严重的应用提供出色的抗噪性能
- 低功耗
  - 两种功耗模式，用于优化功耗
  - 1.5 mA 的活动模式电流；2.6 uA 的睡眠模式电流

1.3.2 器件特性

- 高性能、低功耗 M8C Harvard 架构处理器
  - 多达 2 个 MIPS，带有 12 MHz 内部时钟、外部时钟信号
- 灵活的片上存储器
  - 高达 8 KB 的闪存和 512 B 的 SRAM
  - 支持仿真型 EEPROM
- 高精度的可编程时钟
  - 内部主振荡器（IMO）：6/12 MHz ± 5%

- ☐ 供看门狗和睡眠定时器使用的 32 kHz 内部低速振荡器

- 增强型通用输入/输出（GPIO）功能
  - ☐ 多达 28 个通用 I/O（GPIO），具有可编程配置特性
  - ☐ 所有通用 I/O 均具有 20 mA 的灌电流能力
  - ☐ 在所有 GPIO 上支持内部电阻上拉、HI-Z、开漏和强驱动模式
- 外设功能
  - ☐ 13 位可编程定时器
  - ☐ 看门狗和睡眠定时器
  - ☐ I^2C 主设备 — 时钟速率为 100 kHz；I^2C 从设备 — 速率高达 400 kHz
  - ☐ SPI 主设备和从设备 — 可配置范围为 46.9 kHz - 3 MHz
- 工作条件
  - ☐ 工作电压范围：2.4 V 到 5.25 V
  - ☐ 温度范围：–40 °C 到+85 °C

1.4 文档规范

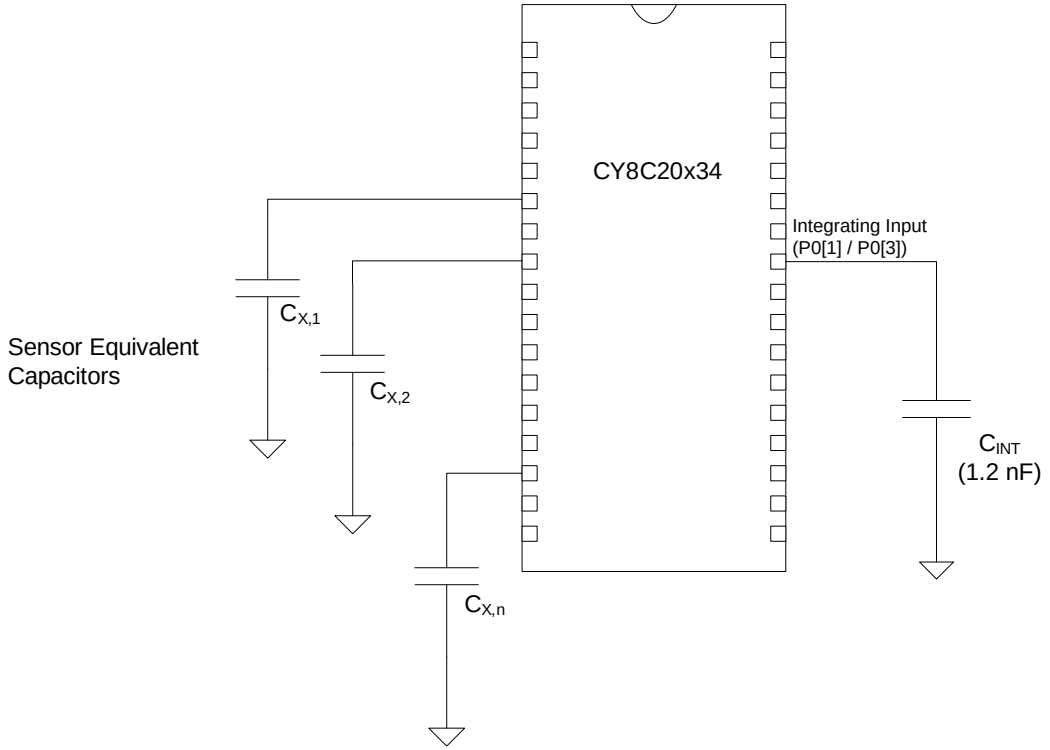
规范	使用说明
Courier New 字体	显示文件位置、用户输入的文本和源代码： `C:\ ...cd\icc\`
斜体字	用于显示文件名称和参考文档： 请阅读 *PSoC Designer 用户指南*中的 *sourcefile.hex* 文件。
[方括号、粗体]	显示程序中的键盘指令： **[Enter]**或**[Ctrl]** **[C]**
File（文件）> Open（打开）	表示菜单路径： **File > Open > New Project**
粗体字	用于显示操作过程中的各条指令、菜单路径和图标名称： 请点击 **File** 图标，然后点击 **Open**。
Times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

2. CapSense 技术

2.1 CapSense 基本原理

CapSense 是一种触摸式感应技术，其工作原理是测量 CapSense 控制器中每个 I/O 引脚上的电容，该控制器作为传感器使用。如图 2-1 所示，对于具有 n 个传感器的设计，每个传感器引脚的总电容可以模拟为 $C_{X,1}$ 到 $C_{X,n}$ 的等效集总电容。CY8C20x34 器件的内部电路将每个 C_X 数量级转换成数字代码，然后进行存储以供后期处理。其它组件，即积分电容 C_{INT}，由 CapSense 控制器内部电路所使用。

图 2-1. CY8C20x34 PSoC 器件中的 CapSense 特定器件实施



如图 2-1 所示，每个传感器 I/O 引脚均通过走线、过孔或两者连接至传感器板。覆盖层是置于传感器板上的绝缘盖，也是该产品触摸界面的组成部分。当手指与覆盖层接触时，人体组织的导电性会产生一个与传感器板并行的接地导电层，如图 2-2 所示。该操作构成了平行板电容器，其电容值可通过以下公式得出：

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

其中：

C_F = 手指与传感器覆盖层接触时所产生的电容值

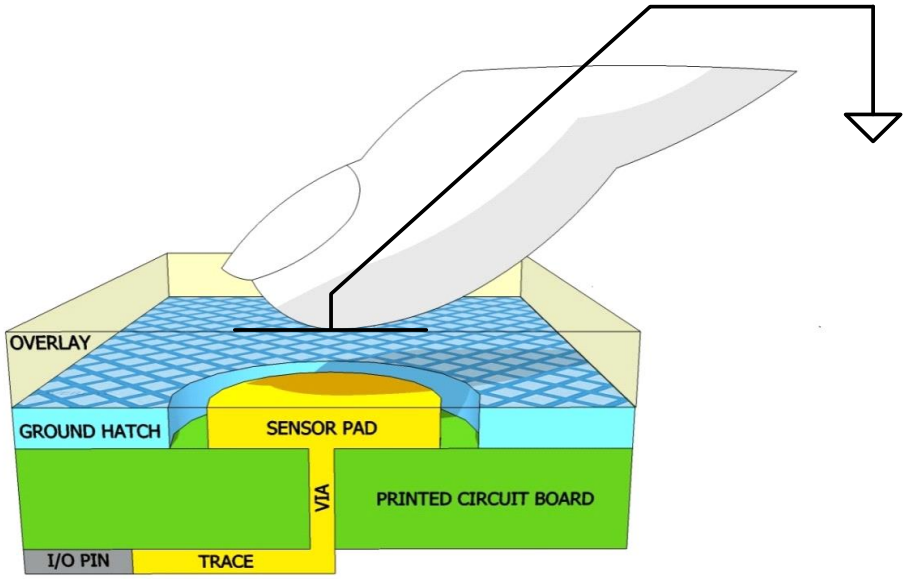
ε_0 = 空气介电常数

ε_r = 覆盖层的绝缘常数（相对介电常数）

A = 手指与传感器板覆盖层的接触面积

D = 覆盖层的厚度

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



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

即使手指未触摸覆盖层，传感器 I/O 引脚也有一些寄生电容（C_P）。C_P 是由 CapSense 控制器内部寄生电容与耦合电场共同产生的，其中电场是由传感器板、走线以及过孔与系统中其它导体（如地层、其它走线、产品机壳或外壳中的任何金属）之间的耦合产生的。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：

$$C_X = C_P \qquad \text{公式 2}$$

当手指接触传感器板上时，C_X 等于 C_P 与 C_F 之和：

$$C_X = C_P + C_F \qquad \text{公式 3}$$

通常，C_P 比 C_F 大几个数量级。C_P 的取值范围通常为 6 pF - 15 pF，但在极端情况下它的值可以高达 50 pF。C_F 的范围通常为 0.1 pF - 0.4 pF。调试 CapSense 系统时，C_P 的数量级至关重要，这将在使用用户模块来调试 CapSense 性能中进行讨论。

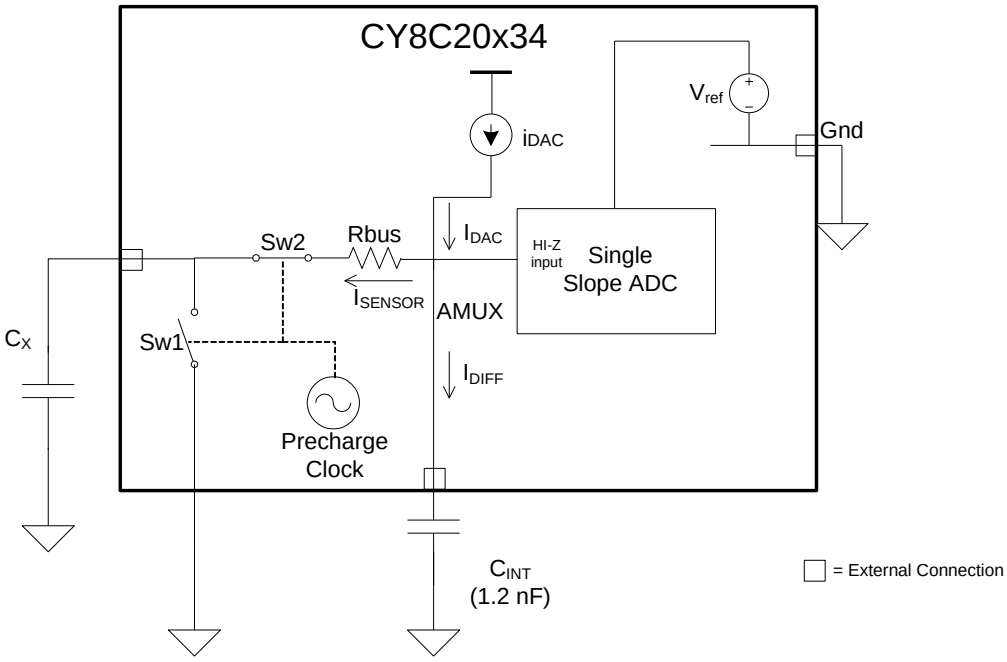
2.2 CY8C20x34 中的 CapSense 方法

CY8C20x34 器件支持 CapSense 逐次逼近电磁兼容性（CSA_EMC）方法，以便将传感器电容（C_X）转换成数字代码。CSA_EMC 方法在 PSoC Designer 中被作为用户模块使用，有关的详细内容请见后面一节。

2.2.1 CapSense 逐次逼近电磁兼容性（CSA_EMC）

CY8C20x34 器件使用的 CSA_EMC 方法将 C_X 整合至开关式电容电路中，如图 2-3 所示。

图 2-3. CSA_EMC 框图



恒流源（iDAC）为模拟复用器总线提供的电流量为 I_{DAC}。传感器（C_X）分别通过开关 Sw1 和 Sw2 交替连接到模拟复用器（AMUX）总线和 GND，通过 AMUX 总线泄放 I_{SENSOR} 电流量。I_{SENSOR} 的大小与 C_X 的大小成正比。开关 Sw1 和 Sw2 由非重叠的时钟（预充电时钟）提供时钟脉冲。

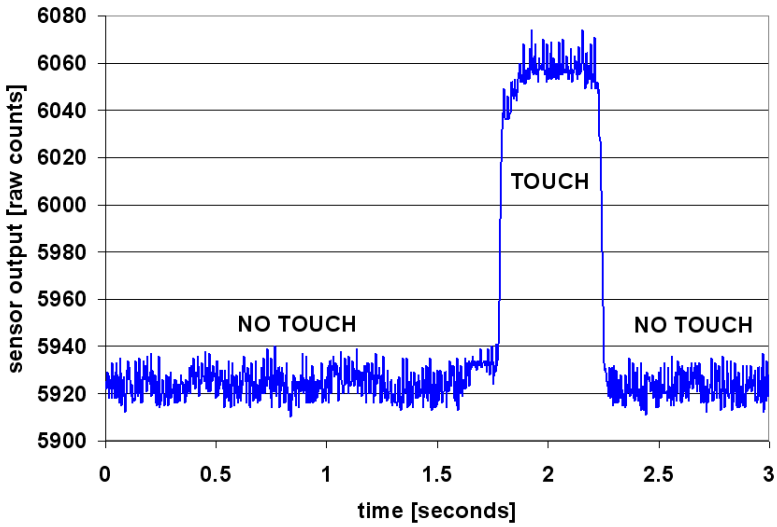
积分电容 C_{INT} 累计差分电流 I_{DIFF}（I_{DAC} 与 I_{SENSOR} 之间的差值）并提高潜能。持续进行此电荷积分，直到 C_{INT} 的展开潜能达到平衡水平，即 I_{SENSOR} 等于 IDAC。积分时间是指建立时间。

单斜 ADC 用于将 C_{INT} 上的平衡潜能转变为数字输出计数，称为原始计数，该值与 C_X 成比例。该原始计数由高级算法进行计算，以便求得传感器的状态。

通过使用逐次逼近法可以设置 I_{DAC} 电流，从而确保 C_{INT} 上的平衡电压位于 ADC 的线性转换区。

图 2-4 表示从手指触摸传感器到释放这段时间内，连续进行扫描得出的 CSA_EMC 原始计数。正如 CapSense 基本原理所解释，手指触摸使 C_X 增加到 C_F，使原始计数按比例增加。通过比较稳定状态下原始计数的偏移量与预定阈值，高级算法可以确定传感器处于 On（触摸）状态还是 Off（离开）状态。

图 2-4. 手指触摸期间的 CSA_EMC 原始计数



CSA_EMC CapSense 算法在出现 RF 干扰的情况下正常运行。当 CapSense 遭遇传导干扰、交流噪声以及变换器、变压器和电源等其它噪声源时，可使用 CSA_EMC。低级参数部分将详细讨论该主题。

3. CapSense 设计工具

3.1 概况

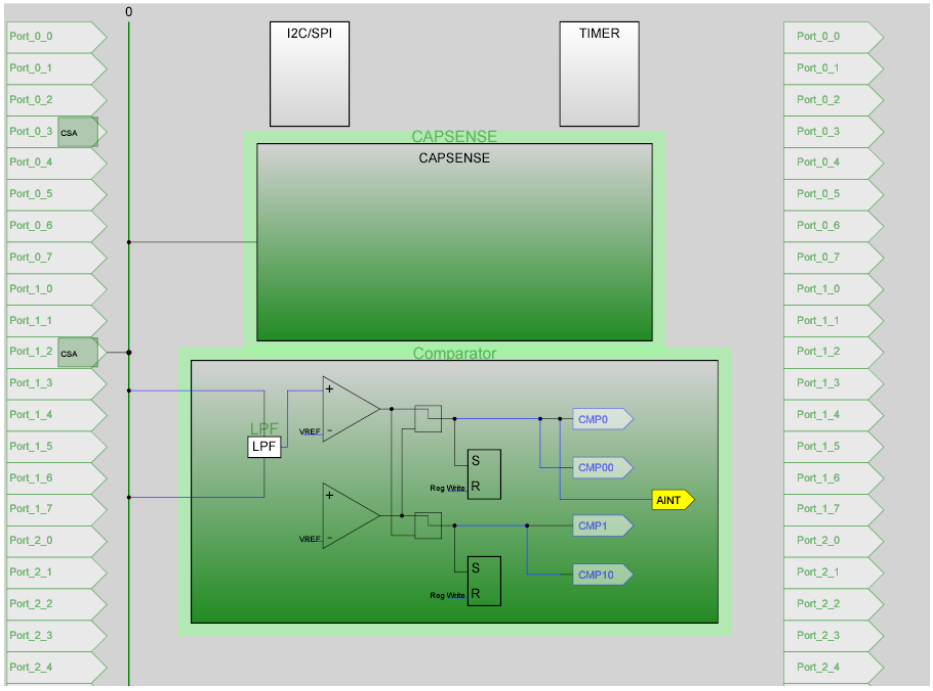
赛普拉斯可以提供用来开发 CapSense 电容式触摸感应应用的全套硬件和软件工具。CY8C20x34 系列的基本开发系统包括本章所讨论的组件。有关订购信息，请参见资源部分。

3.1.1 PSoC Designer 和用户模块

您可利用赛普拉斯专有的集成设计环境（PSoC Designer），配置模拟和数字模块、开发固件、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。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内来完成。PSoC Designer 具有内置的 C 语言编译器和嵌入式编程器。专业版编译器可用于复杂设计。

CSA_EMC 用户模块使用开关式电容电路、模拟复用器、比较器、数字计数功能和高级软件子程序（API）来实现电容式触摸传感器。其它模拟和数字外设用户模块可用于实现其它功能，例如，I²C、SPI、TX8 和定时器。

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 CY8C20x34 项目：

1. 新建 PSoC Designer 项目，其中，CY8C20x34 为目标器件。
2. 选择和放置任意需要专用引脚的用户模块（例如，I²C 或 LCD）。分配端口和引脚。
3. 选择和放置 CSA_EMC 用户模块。
4. 右键单击 CSA_EC 用户模块以访问 CSA_EMC 向导。
5. 设置您需要的传感器、滑条或辐射滑条的数量。
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. 调整用户模块数据手册中的采样代码来实现按键、滑条或触摸板。

有关 CapSense 用户模块的代码示例，请参见代码示例。

3.1.2 通用的 CapSense 控制器套件

通用 CY3280-20x34 CapSense 控制器套件使用预定义的控制电路和插入硬件，简化了原型设计和系统调试。套件包括 MiniProg 1，用于编程；也包括 I²C 连接 USB 的桥接器硬件（CY3240 - I2USB 桥接器），用于调校和数据采集。

3.1.3 通用 CapSense 控制器模块电路板

赛普拉斯模块电路板具有一系列传感器、LED 和接口，以满足您的应用需求。

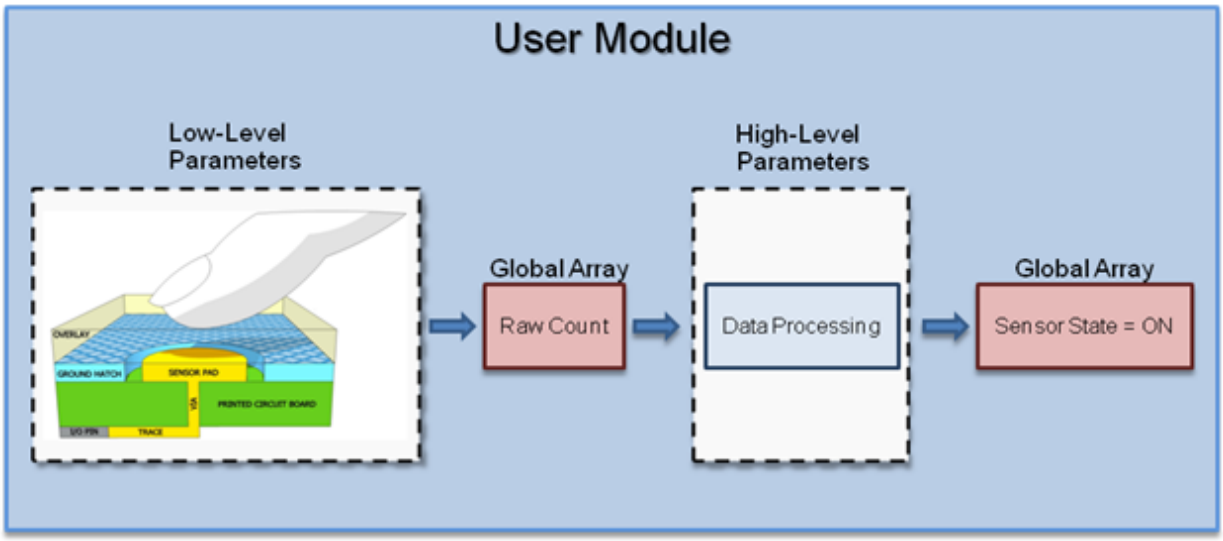
- CY3280-BSM 简单按键模块
- CY3280-BMM 矩阵按键模块
- CY3280-SLM 线性滑条模块
- CY3280-SRM 辐射滑条模块
- CY3280-BBM 通用 CapSense 原型设计模块

3.1.4 CapSense 数据查看工具

CapSense 设计过程中，您可能经常需要监测 CapSense 数据（原始计数、基准线、差值计数等），以进行调校和调试。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信息，请参见应用笔记 AN2397 — CapSense 数据查看工具。

3.2 用户模块概述

图 3-2. 用户模块框图



用户模块包含整个 CapSense 系统，即从物理感应到数据处理。用户模块的行为是通过使用各种参数定义的。这些参数影响感应系统的不同部件，分为低级参数和高级参数，它们之间使用全局阵列进行通信。

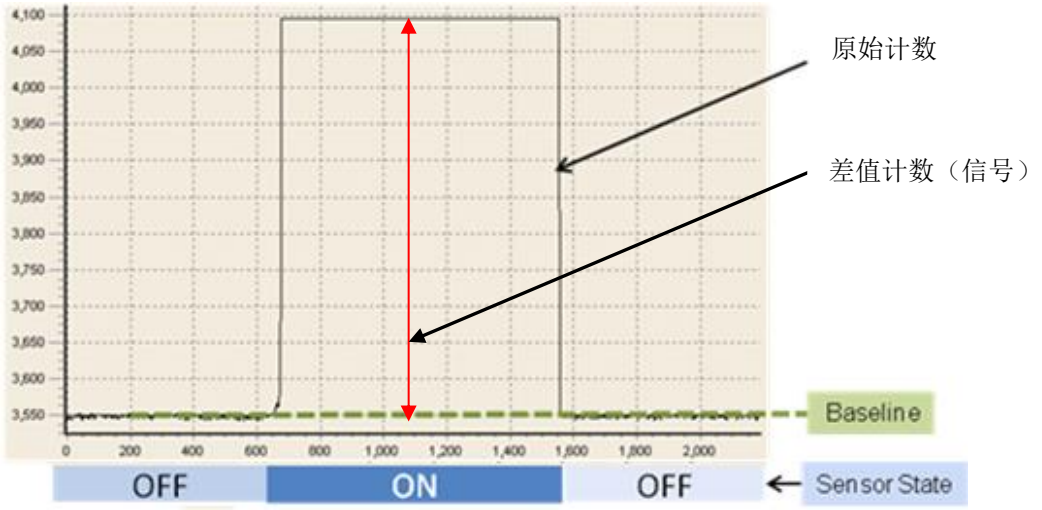
低级参数（如扫描传感器的速度和分辨率）定义物理层上感应方法的行为，并与电容到原始计数的转换相互关联。这些参数对于每一种感应方法而言均是唯一的，有关这些参数的描述，请参见低级参数。

高级参数（如防反跳计数和噪声阈值）定义如何处理原始计数来产生相关信息，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，有关这些参数的相关描述，请参见高级参数。

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。不应该对这些阵列手动更改，但如需调试，您可对其进行检查。

图 3-3. 原始计数、基准线、差值计数和传感器状态



3.3.1 原始计数

调用用户模块 API CSA_EMC_ScanSensor()时，CapSense 控制器中的硬件电路将测量传感器电容并以数字格式（称原始计数）存储这一结果。

传感器的原始计数与其电容成正比。当传感器电容值增加时，原始计数也随之增加。

CSA_EMC 用户模块将传感器原始计数值存储在 CSA_EMC_waSnsResult[]整数阵列中。该阵列被定义在头文件 CSA_EMC.h 中。

3.3.2 基准线

环境的缓慢改变（如温度和湿度）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准线算法来弥补这些差异。该算法使用基准线变量来完成该功能。基准线变量记录原始计数值的所有渐变差异。实际上，基准线变量存储数字低通滤波器的输出（该数字低通滤波器使用的是输入原始计数值）。

基准线算法由用户模块 API CSA_EMC_UpdateSensorBaseline 执行。

传感器的基准线值存储于 CSA_EMC_waSnsBaseline[]整数阵列中。该阵列被定义在头文件 CSA_EMC.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线值之间的差值。当传感器处于非活动状态时，差值计数为零。激活传感器（通过触摸方式）使差值计数值为正。

传感器的差值计数值存储于 CSA_waSnsDiff[]整数阵列中。该阵列被定义在头文件 CSA_EMC.h 中。

通过用户模块 API CSA_EMC_UpdateSensorBaseline()，可以更新差值计数变量。

3.3.4 传感器状态

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

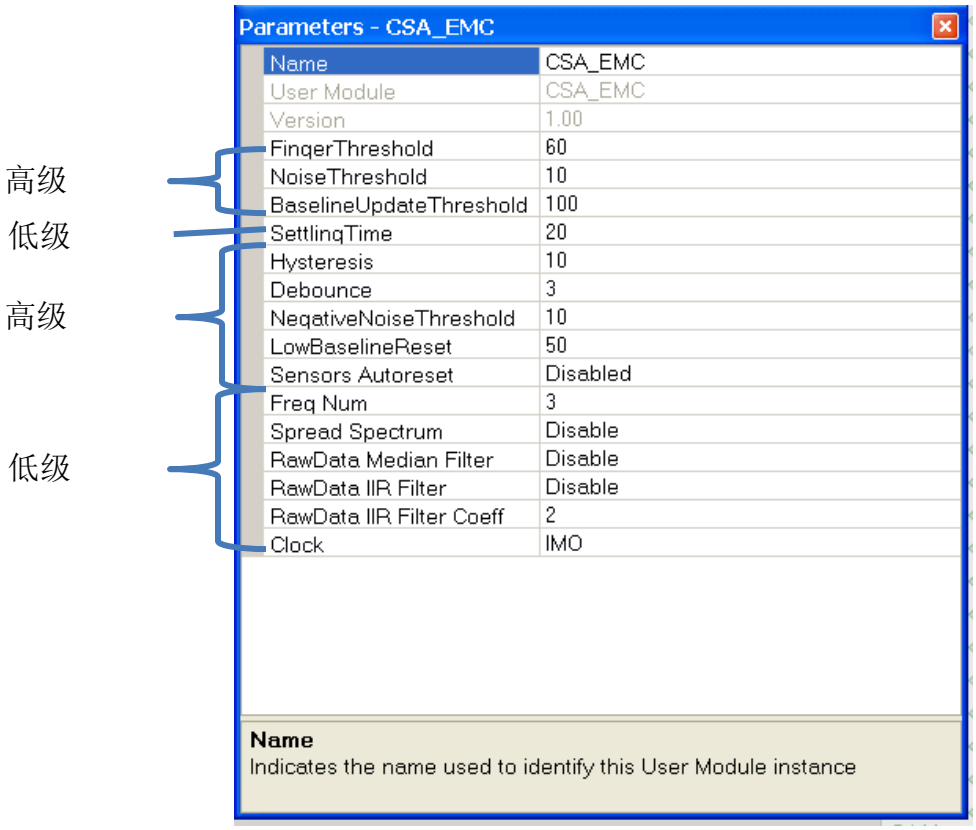
传感器状态存储在名为 CSA_EMC_baSnsOnMask[]的字节阵列中。每个阵列元素存储 8 个连续传感器的状态。该阵列被定义在头文件 CSA_EMC.h 中。

通过用户模块 API CSA_bIsAnySensorActive()，可以更新传感器状态。

3.4 CSA_EMC 用户模块参数

CSA_EMC 用户模块参数分为高级参数或低级参数。有关 CSA_EMC 用户模块参数的列表和分类的信息，请参见图 3-4。

图 3-4. PSoC Designer — CSA_EMC 参数窗口



3.4.1 高级参数

3.4.1.1 手指阈值

用户模块使用该参数以判断传感器的活动/非活动状态。如果传感器的差值计数超过手指阈值，则传感器被判定为活动状态。该定义假定迟滞设为 0，防反跳设为 1。

取值范围为 3 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.2 迟滞

通过迟滞设置可以防止传感器因系统噪声而一直处于 ON（触摸）状态。公式 4 给出了该迟滞函数。该公式假设防反跳被设置为 1。

图 3-5. 传感器状态与差值计数（迟滞被设置为零）

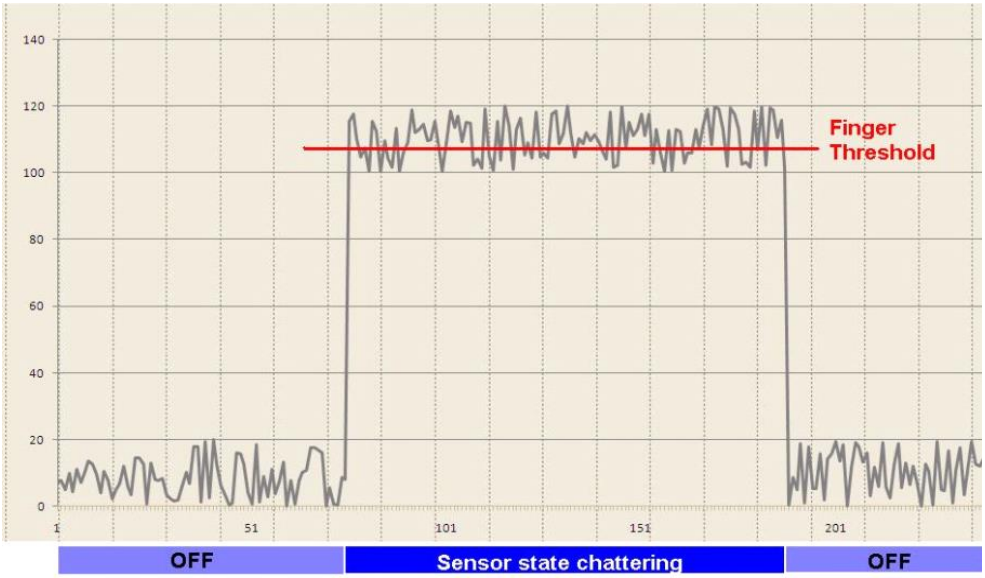
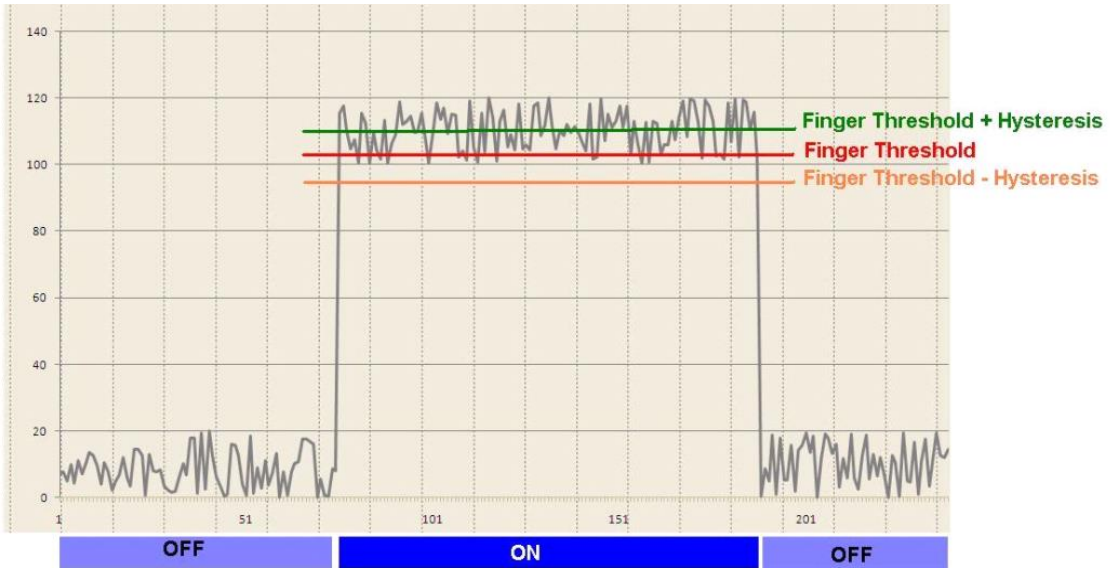


图 3-6. 传感器状态与差值计数（迟滞非零）



如果差值计数 ≥ 手指阈值 + 迟滞，传感器状态为 ON

如果差值计数 ≥ 手指阈值 - 迟滞，传感器状态为 OFF 公式 4

取值范围为 0 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.3 防反跳

防反跳防止原始计数的尖峰脉冲将传感器状态从 OFF 改为 ON。对于传感器状态从 OFF 到 ON 的瞬变而言，差值计数必须在指定的采样数内保持超过手指阈值与迟滞之和。

取值范围为 1 到 255。如果将该参数设置为 1，则不提供防反跳。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.4 基准线更新阈值

如前所释，基准线变量会追踪原始计数的渐进变化。也就是说，基准线变量存储数字低通滤波器的输出（该数字低通滤波器使用输入原始计数值）。基准线更新阈值参数用于调整该低通滤波器的时间常数。

该基准线更新阈值与该滤波器的时间常数成正比。基准线更新阈值越高，时间常数就越大。

取值范围为 1 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.5 噪声阈值

噪声阈值有助于用户模块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并两者之间的差值大于该阈值时，基准线更新算法会暂停。

对于滑条传感器而言，当差值计数大于噪声阈值时，质心计算会暂停。

取值范围为 3 到 255。为了保证正确的用户模块操作，噪声阈值不会高于手指阈值与迟滞之差。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.6 负噪声阈值

负噪声阈值有助于用户模块了解原始计数中噪声计数的下限。当原始计数低于基准线且两者之间的差值大于负噪声阀值时，基线更新算法将暂停。

取值范围为 0 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.7 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。如果在已指定的采样数内采样计数值小于基准线值与负噪声阈值之差，则将基线值设置为新的原始计数值。该参数实际上计数了复位基准线所需的异常低的样本数值。它用来修正启动时手指触摸传感器的情况。

取值范围为 0 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.1.8 传感器自动复位

传感器自动复位参数确定基准线是随时进行更新，还是仅在差值计数小于噪声阈值时才更新。

启用传感器自动复位后，会不断地更新基准线。该操作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一直处于打开状态。原始信号突然上升可能是由电源电压剧烈波动、高能射频噪声源或温度快速变化所导致的。

如果禁用传感器自动复位，仅在差值计数小于噪声阈值参数时才发生基准线更新。

可能值为“使能”和“禁用”。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.2 低级参数

除高级参数外，CSA_EMC 用户模块还具有多个低级参数。这些参数是 CSA_EMC 感应方法的特定参数，用于确定如何从传感器中采集原始计数数据。

3.4.2.1 建立时间

建立时间参数控制软件延迟，该延迟允许 C_{INT} 电容上的电压达到积分平衡潜能。

取值范围为 2 到 255。

有关建议值的信息，请参见 CSA_EMC 用户模块 — 调校指南。

3.4.2.2 抗噪声级别

抗噪声级别参数定义了抵抗噪声的级别。可用选项有“低”和“高”。

选择“高”选项，可以改善高噪声环境下的性能。然而，这样将增加存储器占用空间，因而降低了支持传感器的最大数量。

此外，选择“高”设置时，所需的扫描时间比“低”设置情况下多两倍。

3.4.2.3 扩频

使能扩频参数可以改善 CSA_EMC 算法在出现外部 RF 干扰情况下的性能。它还可以改善该算法的辐射特性。

可能值为“使能”和“禁用”。

3.4.2.4 原始数据中值滤波器

中值滤波器将查看传感器的最新三个样本并报告中值。该滤波器用于消除短时噪声尖峰。它具有单个采样延迟，并在默认情况下处于禁用状态。

可能值为“使能”和“禁用”。

3.4.2.5 原始数据 IIR 滤波器

该无限脉冲响应（IIR）滤波器可减少转换结果（原始计数）中的噪声。默认情况下，它被禁用。

可能值为“使能”和“禁用”。

3.4.2.6 原始数据 IIR 滤波器系数

该参数设置了原始计数 IIR 滤波器的系数。

可能值为 2（½ 前一个样本 + ½ 当前样本）和 4（¾ 前一个样本 + ¼ 当前样本）。

3.4.2.7 时钟

该时钟参数设置非重叠性预充电时钟的频率，该时钟用来控制开关 Sw1 和 Sw2。Sw1 和 Sw2 用于选择性地将传感器电容连接到 AMUX 总线和 GND。要想获得最大信号，应合理设置预充电时钟的频率，这样可以保证传感器电容在每个周期内完成整个充电和放电过程。

可能值为 IMO、IMO/2、IMO/4 和 IMO/8。

4. 使用用户模块来调试 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、覆盖层材料和应用需求。这些因素将在设计的注意事项中进行讨论。调校是确定最佳参数设置的过程，它的目的是保证传感器操作的稳健和可靠性。

4.1 基本注意事项

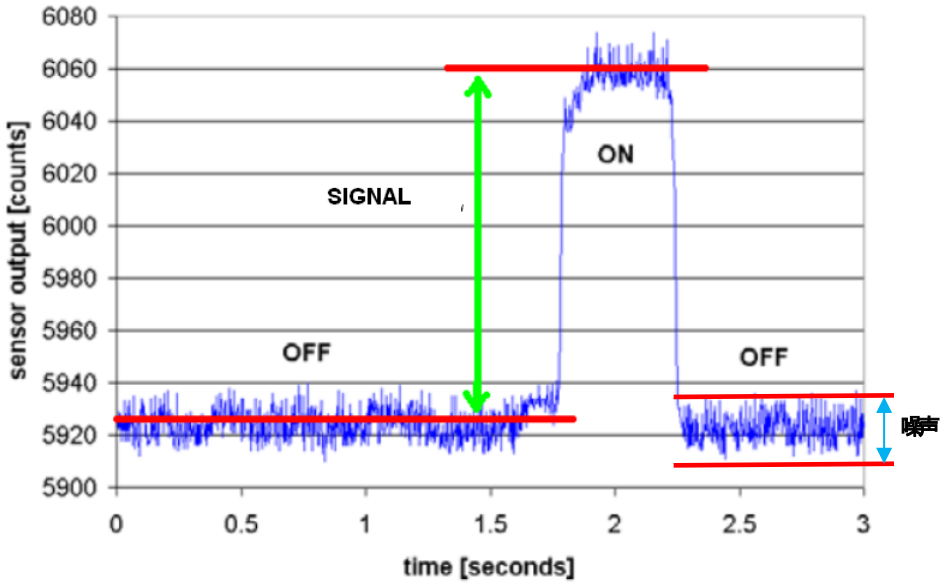
4.1.1 信号、噪声和 SNR

调校好的 CapSense 系统能够准确地识别传感器的 ON 和 OFF 状态。要实现该性能级别，CapSense 信号必须远远大于 CapSense 噪声。通过使用信噪比（SNR）的数值将 CapSense 信号与 CapSense 噪声作比较。讨论 SNR 对 CapSense 的意义之前，需要理解触摸感应环境中信号和噪声这两个概念。

4.1.1.1 CapSense 信号

CapSense 信号是指手指接触传感器时，传感器响应所发生的变化，如图 4-1 所示。传感器的输出是一个数字计数器，该计数器值表示传感器电容。在该示例中，在手指没有触摸传感器时，平均输出为 5925。当手指触摸传感器时，平均输出增至 6060。由于 CapSense 信号表示手指触摸产生的计数变化，因此信号值 = 6060 – 5925 = 135。

图 4-1. CapSense 信号与噪声



4.1.1.2 CapSense 噪声

CapSense 噪声是传感器响应在不存在手指触摸时的峰至峰变化，如图 4-1 所示。在该示例中，手指未触摸时输出波形跳跃限幅为：最大值为 5938，最小值为 5912。因为噪声是波形最大值与最小值之间的差别，因此噪声值 = 5938 – 5912 = 26。

4.1.1.3 CapSense 信噪比

CapSense SNR 简单是信号和噪声的比例。如果信号值为 135，噪声值为 26，那么，SNR 为 135:26，相当于 5.2:1。推荐的 CapSense SNR 最小为 5:1，这表示信号是噪声的 5 倍。用以降低噪声的滤波器是在固件中实现的。有关的更多信息，请参见软件滤波。

4.1.2 基准线更新阈值验证的重要性

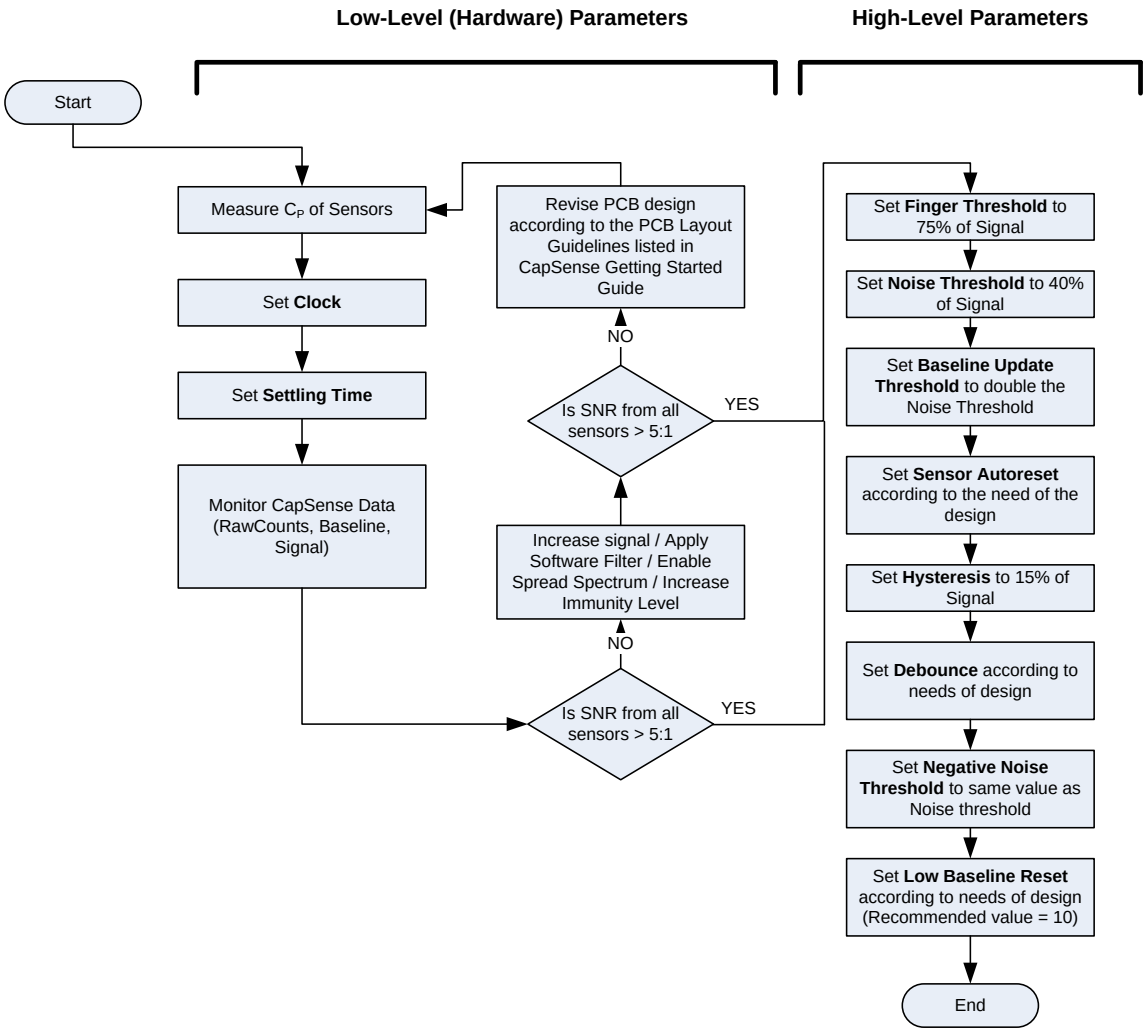
温度和湿度均使平均计数随时间浮动。基准线是用于 CapSense 测量的参考计数级别，对环境影响补偿起着重要的作用。高级决策（例如，手指触摸和手指离开状态）基于基准线建立的参考级别。由于每个传感器具有相应的寄生电容，因此所有电容式传感器均有各自的基准线。

基准线根据基准线更新阈值参数所设置的值来跟踪计数变化。请保证更新阈值符合所需应用要求。如果更新速度太快，那么，基准线将补偿手指产生的所有变化，因此不能检测到手指移动。如果更新速度太慢，则较慢的环境变化可被错误检测为手指触摸。在开发期间，建议验证基准线更新阈值参数的设置。

4.2 CSA_EMC 用户模块 — 调校指南

图 4-2 显示了 CSA_EMC 参数调校过程的流程图。CSA_EMC 用户模块参数可以分为两类：低级（硬件）参数和高级参数。这两种类别的参数以不同的方式影响电容式感应系统的行为。然而，各个传感器的灵敏度之间存在互补关系，该互补关系由硬件参数设置和许多高级参数设置决定。更改任何硬件参数时，必须考虑这一点，以便确保相关的高级参数得到相应的调整。调校 CSA_EMC 用户模块参数应始终从硬件参数开始。

图 4-2. CSA_EMC 用户模块参数调试流程图



4.1 建议的 C_{INT} 值

调校过程从大小 1.2 nF 的积分电容 C_{INT} 开始。在调校过程中，如果发现传感器信号不满足 5:1 的信噪比，则可以增加 C_{INT}。C_{INT} 的最大值建议为 5.6 nF。建议使用 X7R 或者 NPO 类型的电容，以确保 C_{INT} 在不同的温度条件下保持稳定状态。

4.2 测量传感器寄生电容 C_P

调校程序的第一步是测量传感器寄生电容（C_P）。请逐步执行以下操作：

1. 将 **CPU_CLK** 设置为 SysClk/2。
2. 将 **Clock**（时钟）设置为 IMO/8。
3. 将 **Settling Time**（建立时间）设置为 255。
4. 读取由特定传感器算法设置的 IDAC 代码。该值被存储在 `CSA_EMC_baDACCodeBaseline[]`阵列中
5. 测量与 IDAC 代码相对应的 IDAC 电流。

创建包含以下代码的 PSoC Designer 项目。该代码将 IDAC 连接到端口引脚 P1[4]。

```
//configure P1[4] to HI-Z
PRT1DM0 &= ~ (1<<4);
PRT1DM1 |= (1<<4);
//connect P1[4] to analog mux bus
MUX_CR1 = (1<<4);
// set IDAC to read back IDAC Code
IDAC_D = <IDAC CODE>
// turn ON IDAC
CS_CR2 = 0xD0;
```

将电流表放在引脚 P1[4]与地之间，然后测量电流。将该值命名为 $I_{MEASURED}$。

6. 使用公式 5 计算 C_P。

$$C_P = \frac{I_{MEASURED}}{(\frac{IMO}{8}\times 1.3)}$$ 公式 5

另外，也可以使用 LCR 仪表测量传感器的 C_P。将 LCR 仪表的一个终端连接到传感器引脚上，另一个终端连接到 GND，以测量 C_P。

4.3 估算 CSA_EMC 时钟

通过 CSA_EMC 时钟用户模块参数来设置预充电时钟。这是正确调校 CSA_EMC 设计时的最重要硬件 UM 参数。CSA_EMC 时钟的建议值如表 4-1。

表 4-1. 基于 C_P 和 IMO 的 CSA_EMC 时钟设置

	CSA_EMC 时钟	
C_P（pF）	IMO = 12MHz	IMO = 6MHz
< 5	IMO	IMO
5 到 10	IMO/2	IMO
10 到 15	IMO/4	IMO/2
15 到 20	IMO/4	IMO/2
20 到 25	IMO/8	IMO/4
25 到 30	IMO/8	IMO/4
30 到 35	IMO/8	IMO/4
35 到 40	IMO/8	IMO/4
40 到 45	IMO/8	IMO/4
45 到 50	IMO/8	IMO/8

4.4 设置建立时间

使用公式 6 估算建立时间参数的最小值。

$$Settling\ Time = \frac{5\times C_{INT}}{(Clock\times C_P\times 25\times \frac{1}{FCPU})}$$ 公式 6

其中：

C_{INT} = 积分电容值

时钟 = 预充电时钟频率（CSA_EMC 时钟）

C_P = 传感器的寄生电容值

F_{CPU} = CPU 时钟频率

4.5 监测 CapSense 数据

请参见 CapSense 数据查看工具中的内容。

4.6 提高信噪比的方法

想要提高 SNR，既可以通过降低噪声计数，也可以通过提高信号计数。

4.6.1 降低噪声

提高 SNR 的一种方法是降低噪声计数。您可以使用以下某个选项：

- 使用软件滤波器— 有关详细信息，请参见软件滤波一节。
- 使能扩频— 有关详细信息，请参见扩频内容。
- 提高抗噪级别— 有关详细信息，请参见抗噪声级别内容

4.6.2 提高信号

也可以通过提高信号来提高 SNR。使用这两种方法之一：

- 增加宏 `CSA_EMC_BASELINE` 定义的值。该宏位于 *CSA_EMC.inc* 文件内。该宏的默认值为 0x0800。
- 增加 C_{INT} 电容值。

4.7 调校示例

本节提供了使用以下指导来调校单个 CapSense 传感器的示例。在本示例中，假定您熟悉了 PSoC Designer 和 CapSense 数据查看工具数据查看工具。有关更多信息，请参见 PSoC Designer 用户模块和 AN2397 - CapSense 数据查看工具内容。

4.7.1 硬件要求

可以在您自己的电路板中或使用赛普拉斯的标准 CapSense UCC 电路板（下面给出链接）执行调校。

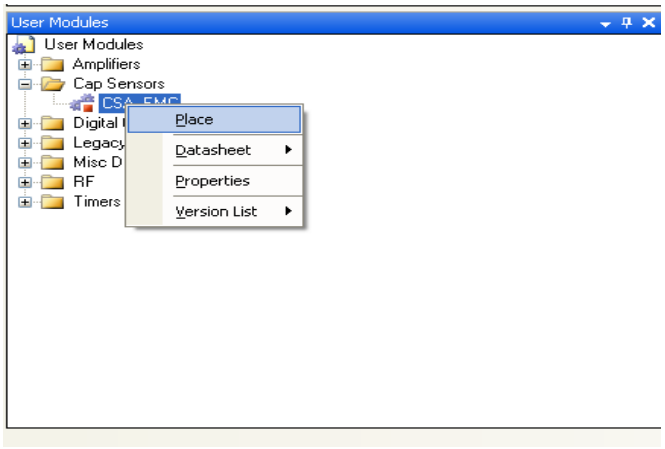
- CY3280-20x34 通用 CapSense 控制器
- CY3280-BSM 简单按键模块套件

下面给出的逐步调校程序使用 CapSense 传感器（端口引脚 P1[4]）和积分电容（端口引脚 P0[3]）。

4.7.2 逐步调校程序

1. 新建目标器件为 CY8C20x34 的 PSoC Designer 项目。
2. 转到 **User Module Catalog**（用户模块目录）窗口。选择和放置 Cap 传感器类别中的 CSA_EMC 用户模块。

图 4-3. 选择和放置 CSA_EMC 用户模块



3. 在 Workspace Explorer（工作区浏览器）中，右键单击 CSA_EMC 用户模块，以访问 CapSense 向导。分配传感器和 C_{INT} 的引脚。

图 4-4. 右键单击 CSA_EMC 用户模块以启动 CapSense 向导

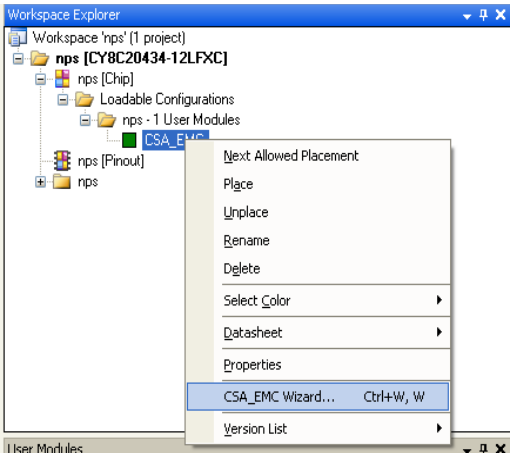
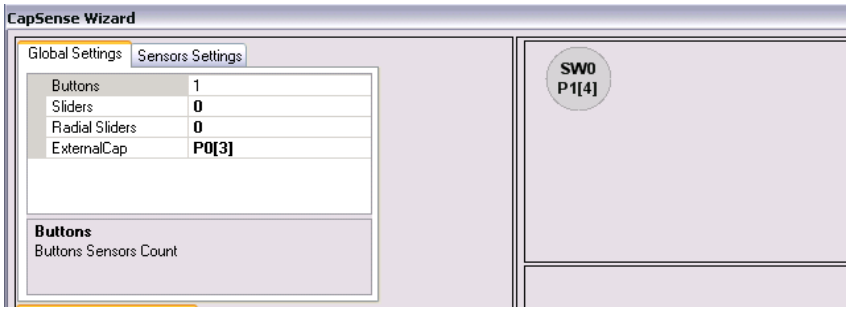
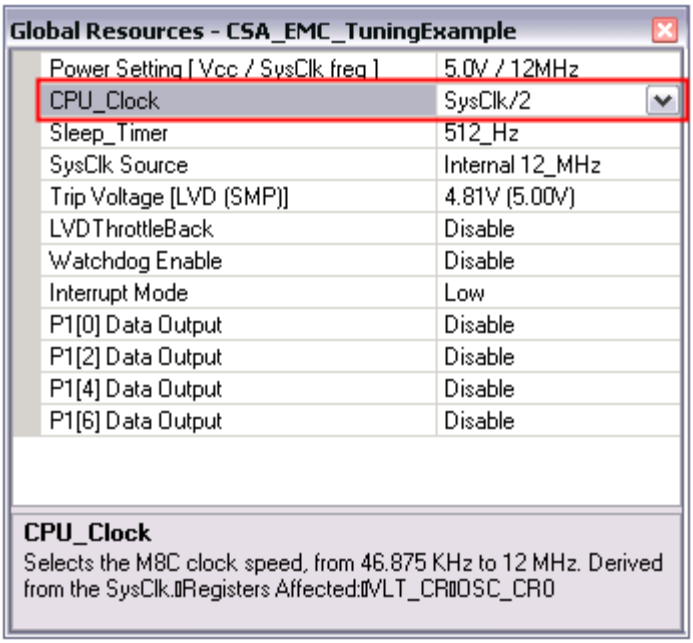


图 4-5. CapSense 向导（传感器引脚 = P1[4]、C_{INT} 引脚 = P0[3]）



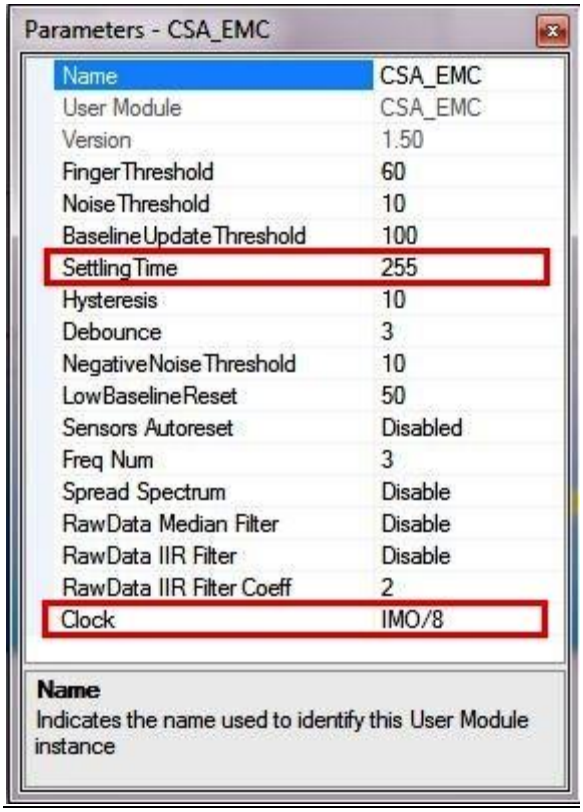
4. 在 **Global Resources** 窗口中，将 **CPU_CLK** 设置为 SysClk/2。根据您的设计需求，在 **Global Resources** 窗口中配置剩余参数。

图 4-6. 设置 CPU_CLK = SysClk/2



5. 在 **Parameters - CSA_EMC** 窗口中，设置**时钟** = IMO/8 和**建立时间** = 255。保存其它所有参数的默认值。

图 4-7. 设置时钟 = IMO/8，建立时间 = 255



6. 凭借数字通信接口（I²C 或 SPI），可以使用任意 CapSense 查看工具（MultiChart 或桥控制面板）读取 CSA_EMC_baDACCodeBaseline[]中的 IDAC 代码。有关更多信息，请参见 AN2397 - CapSense 数据查看工具一节。

   代码示例：

```
void main(void)
{
    M8C_EnableGInt;
    CSA_EMC_Start();
    …
        …
    while(1)
    {
    // OUTPUT CSA_EMC_baDACCodeBaseline[0] <- IDAC code for sensor 0;
    }
}
```

将该步中的读回代码为 0x08。

7. 测量与值为 0x08 的 I_{DAC} 代码相对应的 I_{DAC} 电流。

   代码示例：

```
void main(void)
{
        //configure P1[4] to HI-Z
        PRT1DM0 &= ~ (1<<4);
        PRT1DM1 |= (1<<4);
        //connect P1[4] to analog mux bus
        MUX_CR1 = (1<<4);
        // set IDAC to read back IDAC Code
        IDAC_D = 0x08;
        // turn ON IDAC
        CS_CR2 = 0xD0;
}
```

有关测量该电流的详细信息，请参见测量传感器寄生电容 CP 中的第 5 步。

电流测量值取为 23 µA。

8. 计算 C_P。

   根据公式 5，$C_P = I_{MEASURED} / ((IMO/8) * 1.3)$

   如果进行以下替换：

   $I_{MEASURED}$ = 23 µA

   IMO = 12 MHz（在本示例中，IMO 被设置为 12 MHz）

   那么，C_P = 12 pF

9. 计算 CSA_EMC 时钟。

   根据表 4-1，对于 C_P = 12 pF 和 IMO = 12 MHz，建议时钟为 IMO/4。

10. 计算建立时间。

    根据公式 5 中：

$$Settling\ Time = \frac{5 \times C_{INT}}{(Clock \times C_P \times 25 \times \frac{1}{FCPU})}$$

    如果进行以下替换：

    C_{INT}= 1.2 nF

时钟 = IMO/4 = 3 MHz

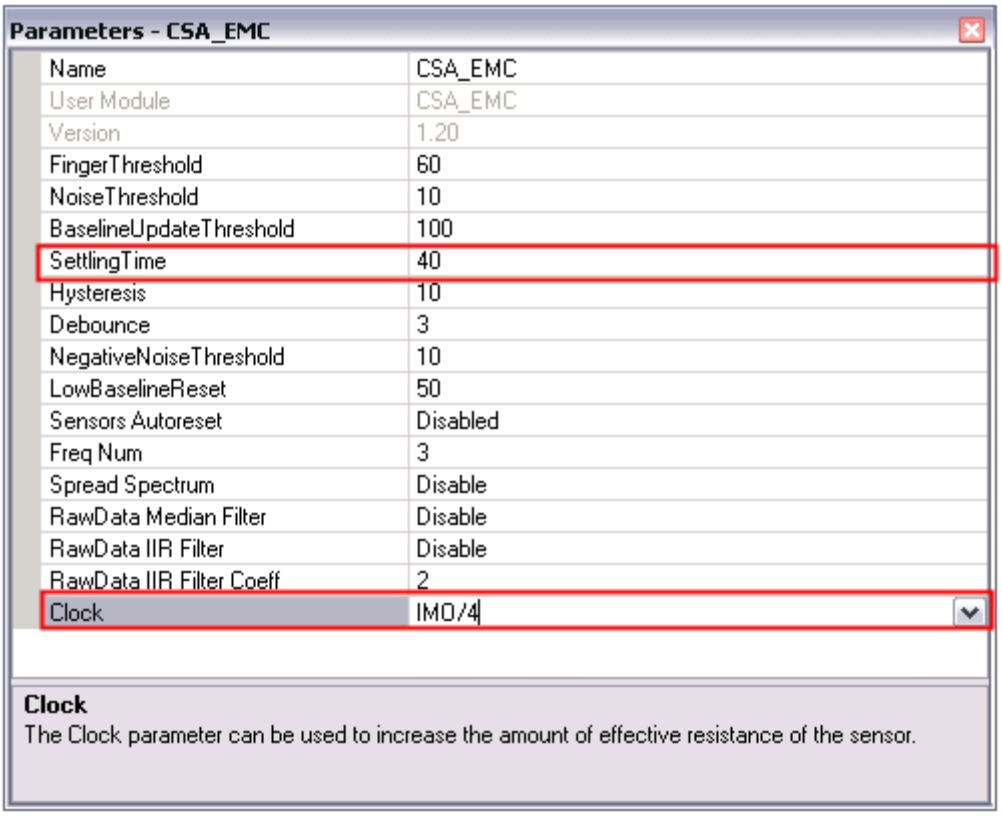
C_P = 12 pF

F_{CPU} = SysClk/2 = 12 MHz/2 = 6 MHz （在本示例中，SysClk 被设置为 12 MHz）

那么，建立时间 = 40

11. 在 **Parameters - CSA_EMC** 窗口中，更新时钟和建立时间值。

图 4-8. 设置时钟 = IMO/4，建立时间 = 40



12. 使用新的设置测量 SNR。

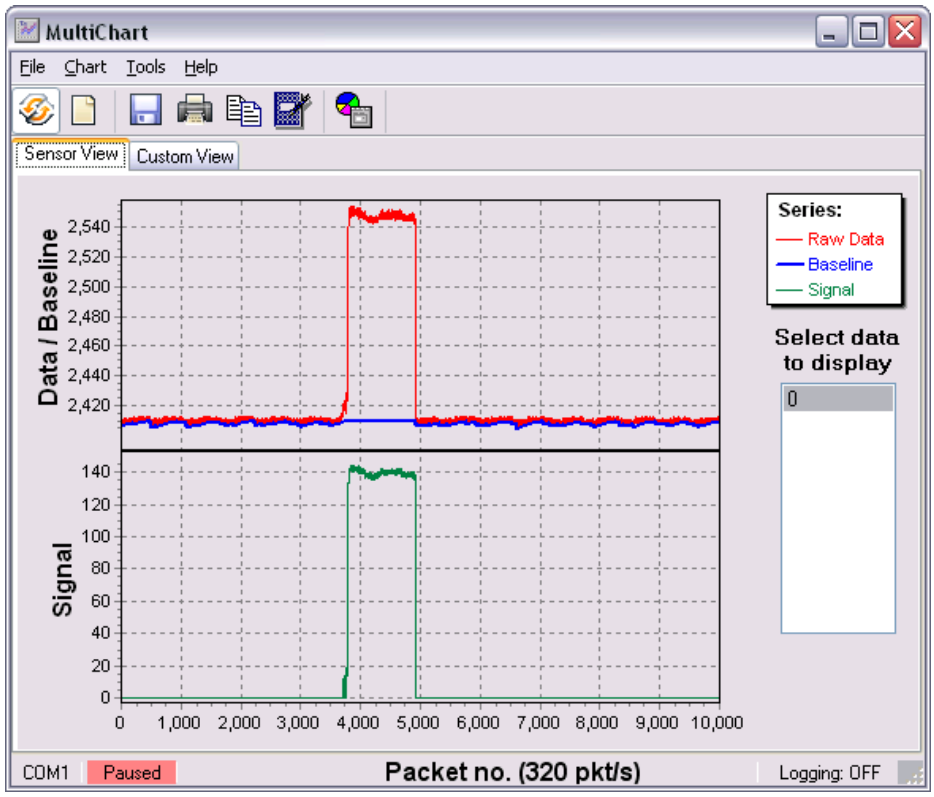
图 4-9 显示了观察到的原始计数、基准线和信号级别。

信号 = 140 计数（近似值）

噪声 = 10 计数（近似值）

因此，SNR = 14。该值大于 5。

图 4-9. 信号 = 140 计数，噪声 = 10 计数



13. 设置高级参数：

    手指阈值 = 信号的 75% = 0.75 × 140 = 105

    噪声阈值 = 信号的 40% = 0.40 × 140 = 56

    基准线更新阈值 = 2 × 噪声阈值 = 112

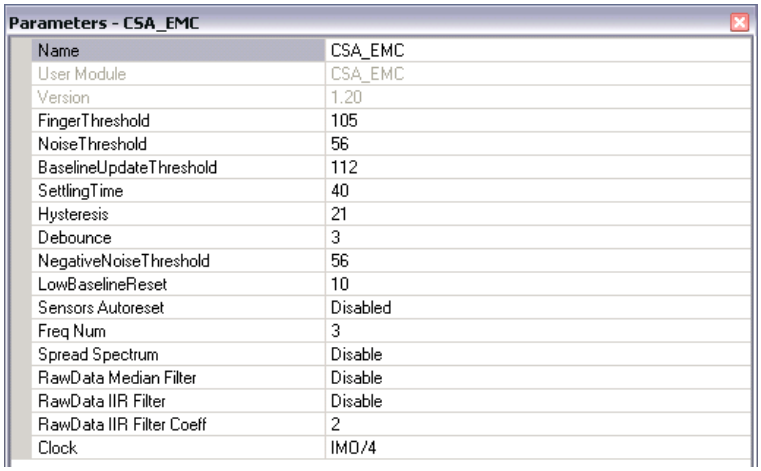
    迟滞 = 信号的 15% = 0.15 × 140 = 21

    负噪声阈值 = 噪声阈值 = 56

    低基准线复位 = 10

14. 根据上面计算结果更新 CSA_EMC 参数值。

图 4-10. 更新的 CSA_EMC 参数窗口



15. 使用新的设置，通过 *main.c* 文件中指定的设计代码，重新生成项目。

5. 设计的注意事项

为应用设计电容式触摸感应技术时，请注意：CapSense 器件是配置在较大规模的框架中的。认真考虑从 PCB 布局到用户界面、最终使用操作环境等各个细节问题，可以保证稳健且可靠的系统性能。有关详细信息，请参见 CapSense 入门内容。

5.1 覆盖层选择

CapSense 基本原理模型同时提供了 CapSense 等效模型与手指电容公式：

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

其中：

ε_0 = 空气介电常数

ε_r = 覆盖层的介电常数

A = 手指与传感器板覆盖层的接触面积（mm^2）

D = 覆盖层厚度（mm）

想提高 CapSense 的信号强度，请选择绝缘常数较高的覆盖层材料，降低覆盖层的厚度，并增加按键的直径。

表 5-1. 覆盖层材料介电强度

材料	击穿电压（V/mm）	12 kV 时的覆盖层最小厚度（mm）
空气	1200 – 2800	10
木材 – 干	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

导电材料不能用做覆盖层，因为它会与电场模式相干扰。因此，请勿在覆盖层中使用包含金属颗粒的油漆。

粘合剂常用来将覆盖层粘贴到 CapSense PCB 上。3M™的 200MP 膜产品是适用于 CapSense 应用的透明丙烯酸粘合膜。这种特殊的粘合剂是从纸作胶带卷抽取（3M™编号为 467MP 和 468MP 的产品）的。

5.2 ESD 保护

稳健的 ESD 耐受性是精密系统设计的自然副产品。经过考虑在最终产品（尤其是用户界面）中可能发生的接触放电，设计可以保证能够禁受 18 kV 放电事件，并且不会对 CapSense 控制器造成任何损害。

CapSense 控制器引脚 ESD 承受能力是 2 kV。在大多数情况下，覆盖层材料能为控制器引脚提供充分的 ESD 保护。表 5-1 列出了保护 CapSense 传感器免受 12 kV 放电（IEC 61000-4-2 规范）冲击所需的各种覆盖层材料的厚度。如果覆盖层材料无法提供足够的保护，将按以下顺序应用 ESD 对策：预防、重定向、钳制。

5.2.1 预防

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

5.2.2 重定向

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

5.2.3 钳制

因为是有意使 CapSense 传感器的放置位置接近于触摸表面，所以重定向放电路径可能不太切合实际。在此情况下，可以使用包含串联电阻或特殊用途的 ESD 保护器件。

建议的串联电容值为 560 Ω。

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

表 5-2. 建议用于 CapSense 的低电容 ESD 保护器件

ESD 保护器件		输入电容	漏电流	接触放电的最大限制	空气放电的最大限制
制造商	器件编号				
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

辐射电能可以影响系统测量操作并且可能会影响处理器内核的运行。这种干扰通过 CapSense 传感器走线和其它任何数字或模拟输入进入 PCB 级别上的 PSoC 芯片。最大限度减小射频干扰的布局指南如下：

- **地层**：在 PCB 上提供一个地层。
- **串联电阻**：在离 CapSense 控制器引脚 10 mm 的范围内安装串联电阻
  - 用于 CapSense 输入线的建议串联电阻值为 560 Ω。
  - 用于通信线（I²C 和 SPI）的建议串联电阻值为 330 Ω。
- **走线长度**：建议尽量缩短走线长度。
- **电流环路区域**：建议尽量减少电流的返回路径。在传感器和走线间 1 厘米范围内使用网格地（不是实心填充的地层），以减小寄生电容的影响。
- **RF 源位置**：使系统和带有噪声源（如 LCD 反相器和开关模式电源（SMPS））离开，以便保证干扰与 CapSense 输入相分隔。电源屏蔽是另一种防止干扰的通用方法。

5.3.2 辐射

想要降低来自 CapSense 传感器的辐射，请给开关式电容时钟选择较低频率。在固件中使用预分频器选项控制该时钟。增加预分频器的值将降低开关时钟的频率。

5.3.3 抗传导干扰和辐射

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

- 按照数据手册的建议使用去耦电容。
- 在系统电源的输入上添加双向滤波器。这非常有助于抗传导辐射和干扰。PI 滤波器可以防止电源噪声影响敏感部件，同时也防止部件的开关噪声反耦合到电源层。
- 如果 CapSense 控制器 PCB 通过线缆连接到电源，请尽量缩短线缆的长度并考虑使用屏蔽线缆。
- 您可以在电源或通信线路周围放置一个铁氧体磁珠，用来过滤高频噪声。

有关 EMC 注意事项的更多信息，请参见 CapSense 入门这一章节。

5.4 软件滤波

软件滤波器是处理系统高级噪声的技术之一。表 5-3 列出了适用于 CapSense 的各种滤波器类型。

表 5-3. CapSense 滤波器表

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

表 5-4 详细说明了不同软件滤波器的 RAM 和闪存要求。每种滤波器所需的闪存空间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 Pro 编译器。

表 5-4. RAM 和闪存要求

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

有关软件滤波器的更多信息，请参见 CapSense 入门这部分。

5.5 功耗

5.5.1 系统设计建议

对于许多设计而言，功耗最小化是非常重要的目标。降低 CapSense 电容式触摸感应系统的功耗有多种方法。

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

除了这些方法之外，睡眠扫描方法也非常有效。

5.5.2 睡眠扫描方法

5.5.2.1 平均功耗

在典型应用中，CapSense 控制器不需要始终保持为活动状态。可以使器件进入睡眠状态，从而使 CPU 和器件的主要模块停止运行。在睡眠状态下，器件所消耗的电流远低于活动电流。

通过以下公式，可以计算出器件在较长的运行时间中所消耗的平均电流。

$$I_{AVE} = \frac{(I_{Act} \times t_{Act}) + (I_{Slp} \times t_{Slp})}{T}$$ 公式 7

其中：

I_{AVE} = 器件的平均电流

I_{Act} = 器件的活动电流

t_{Act} = 器件的活动时间

I_{Slp} = 器件的睡眠电流

t_{Slp} = 器件的睡眠时间

$T = t_{Act} + t_{Slp}$

器件的平均功耗可按以下公式计算：

$$P_{AVE} = V_{DD} \times I_{AVE}$$ 公式 8

其中：

P_{AVE} = 器件的平均功耗

V_{DD} = 器件的供电电压

I_{AVE} = 器件的平均电流

5.5.2.2 响应时间与功耗

如公式 8 所述，可以通过降低 I_{AVE} 或 V_{DD} 来降低平均功耗。增加睡眠时间可减少 I_{AVE}。将睡眠时间延长到较高值将导致 CapSense 按键响应时间不良。因此，睡眠时间必须基于系统需求调整。

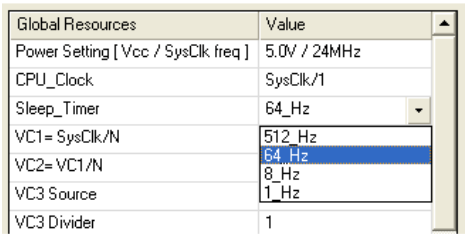
在功耗和响应时间均被视为重要参数的应用中，您可以使用一个优化方法，即是合并持续性扫描和睡眠扫描模式的方法。使用该方法时，器件在大部分时间将处于睡眠扫描模式。如先前章节所述，器件会周期性地扫描传感器并进入睡眠，从而使用较低的功耗。当用户触摸传感器以操作系统时，器件将进入持续扫描模式，在该模式下，器件将持续扫描传感器，而不进入睡眠，因而保证合理的响应时间。在指定的超时期间内，器件仍然处于持续扫描模式。如果在该超时周期内，用户未触摸任何传感器，则器件会切换回睡眠-扫描模式。

5.5.2.3 测量平均功耗

下列介绍了使用睡眠-扫描方法时确定平均功耗的各步骤：

1. 编译扫描所有传感器并不进入睡眠模式（持续扫描模式）的项目。扫描传感器前，代码需要具有引脚切换功能。切换输出引脚的状态可用作时间标记，可以通过示波器进行跟踪。
2. 将项目下载到 CapSense 器件中，然后测量电流消耗。测得的电流是 IAct。
3. 从数据手册中获取睡眠电流信息，该电流是 ISlp。
4. 使用示波器监测输出引脚切换，并测量两次切换之间的时间间隔。这给出活动时间。该时间值是 tAct。
5. 对项目应用睡眠扫描。如图 5-1 所示，在全局资源窗口中选择睡眠定时器频率，将会设置睡眠扫描循环的周期，T。
6. 从睡眠扫描循环期间中减去活动时间，会得出睡眠时间。tSlp = T – tAct。
7. 使用公式 7 计算平均电流。
8. 使用公式 8 计算平均功耗。

图 5-1. 全局资源窗口

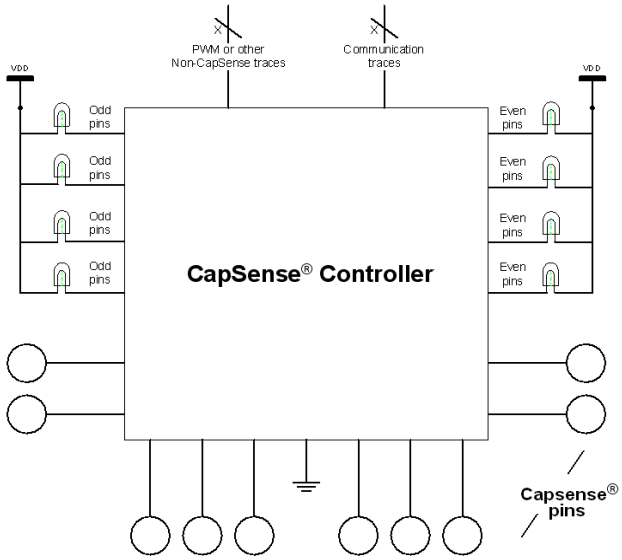


5.6 引脚分配

减少 CapSense 传感器走线和通信线以及非 CapSense 走线之间相互作用的一种有效方法是使用端口分配将它们隔离。图 5-2 显示了针对 32 引脚 QFN 封装进行隔离的基本示例。由于各项功能已被隔离，需要为 CapSense 控制器进行定向，并保证通信、LED 和感应走线之间不存在交叉。

对于具有多个 VSS 引脚的器件，应将所有 VSS 引脚连接到共同的 GND 地层。

图 5-2. 推荐：通信、CapSense 和 LED 的端口隔离



CapSense 控制器架构对奇数和偶数端口引脚数量的电流预算产生限制。对于 CapSense 控制器，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驱动的总电流不应超过 100 mA。除了总电流预算限制之外，每个端口引脚也有最大的电流限制，相关定义请见 CapSense 控制器数据手册这一部分。

所有 CapSense 控制器具有高灌电流和源电流能力的端口引脚。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。

电容式感应应用中的感应线路只能连接到传感器上。连接到其它电路板元件（ISSP 编程信头等）的感应线路对外部噪声更敏感，并具有较高的寄生性电容，这是由于导电路径表面面积增加所导致的。避免将传感器放置在编程引脚 P1[0] 和 P1[1]上。

请注意，P1[0]和 P1[1]对 POR 或 XRES 事件的响应与其它 I/O 引脚所做的响应不相同。XRES 事件之后，这两个引脚通过进入电阻为零的驱动模式而被下拉至地电压，直到达到 HI-Z 驱动模式时为止。发生 POR 事件后，P1[0]会处于高电平，然后进入电阻为零的状态，过一段时间后，将达到 HI-Z 驱动模式状态。发生 POR 事件后，P1[1]进入电阻为零的状态，然后再进入 HI-Z 驱动模式。在您的应用系统中使用这些引脚时，切记这一点。

5.7 PCB 布局指南

PCB 布局的详细准则可以在 CapSense 设计入门指南中获得。

6. 资源

6.1 网站

赛普拉斯 CapSense 控制器网站为您提供了本章节所列出的所有参考材料。

CY8C20x34 页面包含适用于 CapSense CY8C20x34 器件系列的各种技术资源。

6.2 数据手册

CapSense CY8C20x34 器件系列的数据手册可以从 www.cypress.com 上获取。

- CY8C20234、CY8C20334、CY8C20434、CY8C20534、CY8C20634
- CY8C20224、CY8C20324、CY8C20424、CY8C20524

6.3 技术参考手册

赛普拉斯已制定了下列技术参考手册，从而对 CapSense 控制器功能信息（包括顶级架构图、寄存器和时序图）的获取更快、更简单。

- PSoC® CY8C20x24、CY8C20x34 技术参考手册（TRM）

6.4 开发套件

6.4.1 通用的 CapSense 控制器 — 基本套件 1

CY3280-BK1 通用的 CapSense 控制器套件旨在通过预定义的控制电路和插入硬件来简化 CapSense 设计的原型开发和调试。

6.4.2 通用 CapSense 模块板

6.4.2.1 简单按键模块板

CY3280-BSM 简单的按键模块由 10 个 CapSense 按键和 10 个 LED 组成。该模块可连接至任何 CY3280 通用 CapSense 控制器电路板。

6.4.2.2 矩阵按键模块板

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

6.4.2.3 线性滑条模块板

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

6.4.2.4 辐射滑条模块板

CY3280-SRM 辐射滑条模块由四个 CapSense 按键、一个辐射滑条（带十个传感器）和四个 LED 组成。该模块可以连接至任何 CY3280 通用 CapSense 控制器电路板

6.4.2.5 通用 CapSense 原型设计模块

通过 CY3280-BBM 通用 CapSense 原型设计模块，可以访问所有连接到相连控制器电路板上 44 引脚连接器的信号。原型设计模块电路板与通用的 CapSense 控制器电路板一起使用，以实施其它功能（该功能不是其它单用途通用 CapSense 模块电路板所拥有的）。

6.4.3 在线仿真（ICE）套件

ICE 转接板是 CY3215-DK 线上仿真器与原型系统或 PCB 中的 PSoC 目标器件之间的内部互接，其方法是通过封装特定的转接板支脚和柔性线缆来完成。现已推出以下转接板。

- CY3250-20334QFN 线上仿真（ICE）转接板（2），用于调试 QFN CY8C20334 PSoC 器件
- CY3250-20434QFN 线上仿真（ICE）转接板（2），用于调试 QFN CY8C20434 PSoC 器件

6.5 样本电路板文件

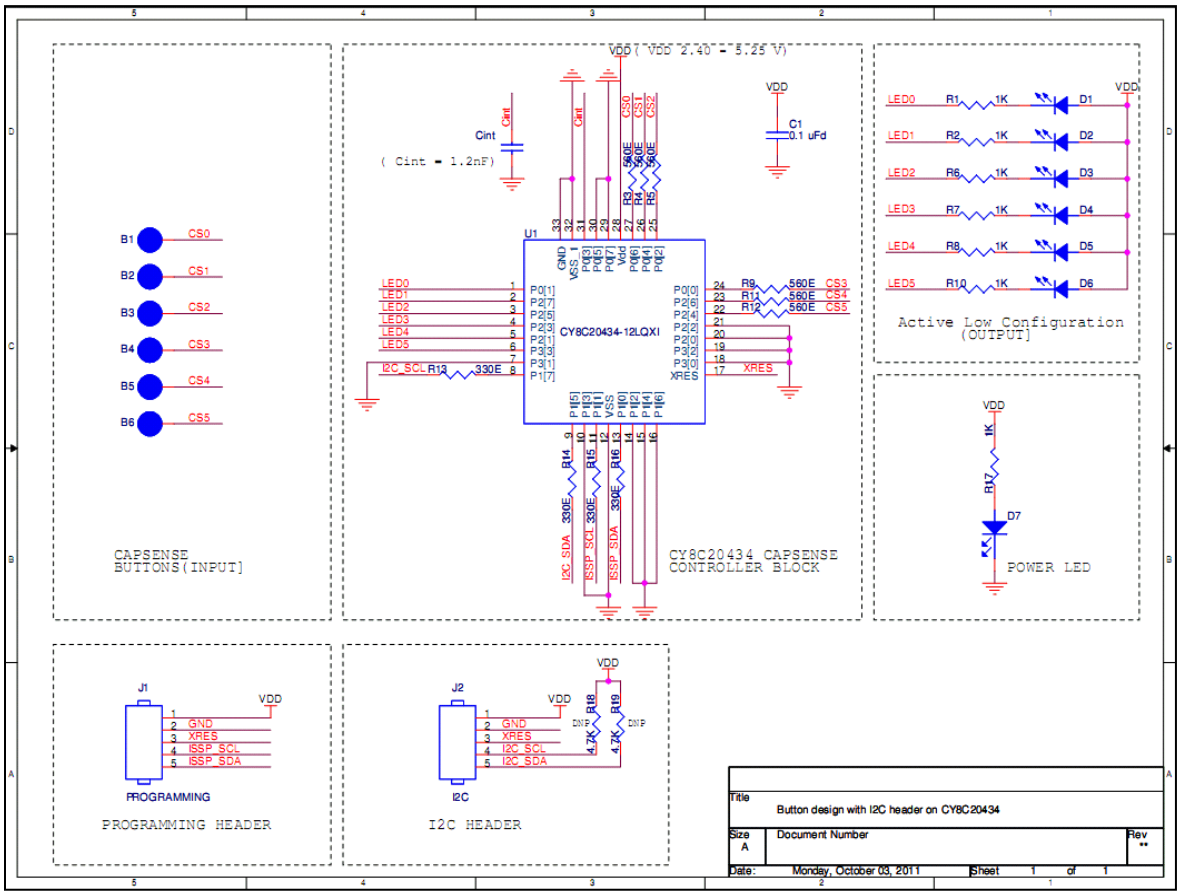
赛普拉斯提供了样本原理图和电路板文件，您可将其用作快速完成 PCB 设计流程的参考。

6.5.1 使用 CY8C20434 上的 I²C 信头来设计按键

图 6-1 显示了使用 CY8C20434 上的 I² 信头来设计按键的原理图。该设计支持：

- 6 个 CapSense 传感器。这些传感器被分配到 CY8C20434-12LQXI 器件的引脚 P0[0]、P0[2]、P0[4]、P0[6]、P2[4] 和 P2[6]上。
- 6 个 GPO，连接到 CY8C20434-12LQXI 的 P0[1]、P2[1]、P2[3]、P2[5]、P2[7]和 P3[3]上以分别驱动 LED D1、D2、D3、D4、D5 和 D6。
- CY8C20434-12LQXI 通过编程信头 J1 来编程。
- I²C 通过 I²C 信头 J2 与 CY8C20434-12LQXI 通信。

图 6-1. 使用 CY8C20434 上的 I²C 信头来设计按键 — 电路板原理图

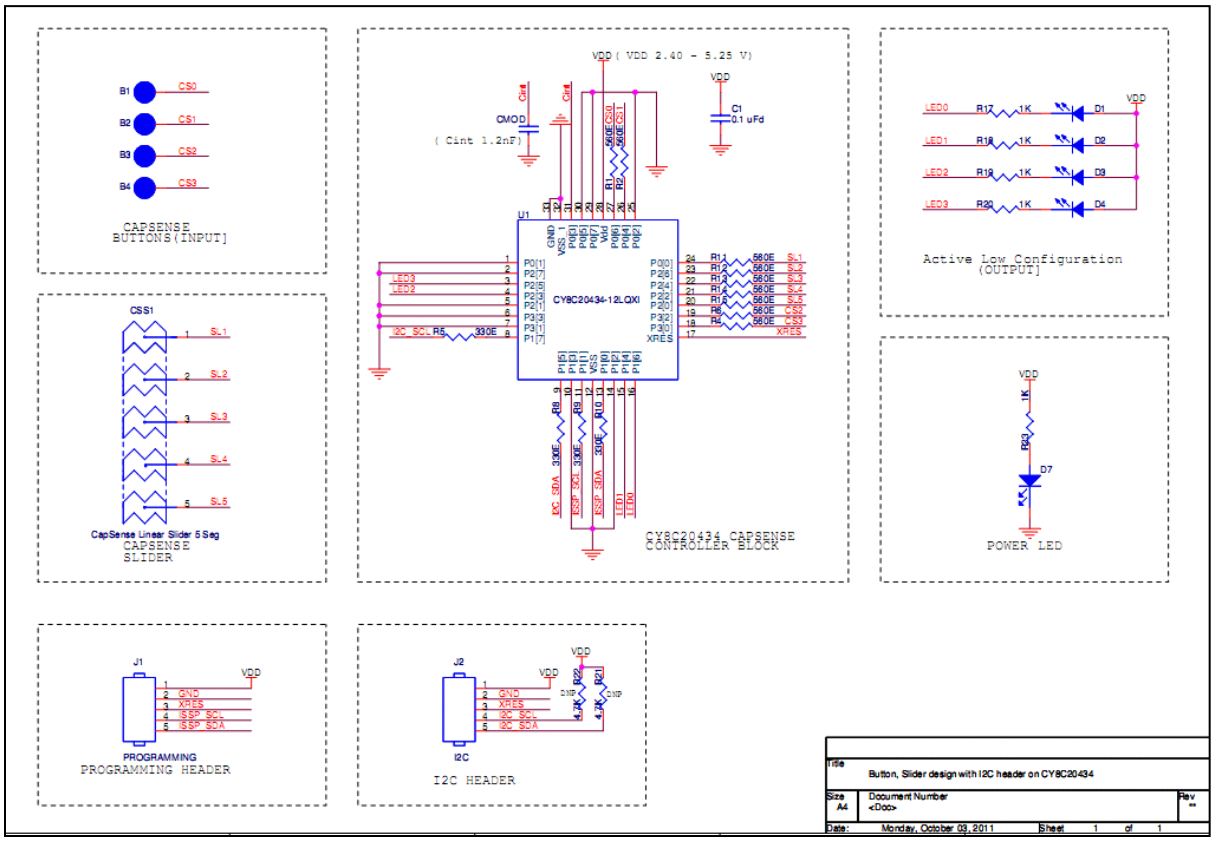


6.5.2 使用 CY8C20434 上的 I²C 信头来设计按键和滑条

图 6-2 给出了使用 CY8C20434 上的 I²C 信头来设计按键和滑条的原理图。该设计支持：

- 4 个 CapSense 传感器。这些传感器被分配到 CY8C20434-12LQXI 器件的引脚 P0[4]、P0[6]、P3[0]和 P3[2]。
- 带有 5 个段的线性滑条。这些段被分配到 CY8C20434-12LQXI 器件的引脚 P0[0]、P2[0]、P2[2]、P2[4]和 P2[6]。
- 4 个 GPO，连接到 CY8C20434-12LQXI 的引脚 P1[4]、P1[6]、P2[5]和 P2[7]，依次驱动 LED D1、D2、D3 和 D4。
- CY8C20434-12LQXI 通过编程信头 J1 来编程。
- I²C 通过 I²C 信头 J2 与 CY8C20434-12LQXI 通信。

图 6-2. 使用 CY8C20434 上的 I²C 信头设计按键和滑条



6.6 PSoC Designer

赛普拉斯具备独有的集成设计环境 — PSoC Designer。使用 PSoC Designer，您可以配置模拟和数字模块、开发固件和调整设计。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）来开发应用。PSoC Designer 带有内置 C 语言编译器和嵌入式编程器。专业版编译器可用于复杂设计。

6.7 PSoC Programmer

PSoC Programmer 是用于编程 PSoC 器件的灵活度且集成度较高的编程应用。PSoC Programmer 可以与 PSoC Designer 和 PSoC Creator 一起使用，以编程 PSoC 器件的任何设计。

PSoC Programmer 提供 API 硬件层，用于设计使用编程器和桥器件的特定应用程序。COM 指导文档包含了 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。

6.8 CapSense 数据查看工具

CapSense 设计过程中，您可能经常需要监测用于调校和调试过程的 CapSense 数据（原始计数、基准线、差值计数等）。

应用笔记 AN2397 — CapSense 数据查看工具为您提供有助于认识和正确使用 CapSense 数据查看和登录工具的重要信息。

6.9 代码示例

赛普拉斯提供了多个示例代码以便组织您的设计并保证快速运作。请参考 CapSense 控制器代码示例设计指南，以测量 CapSense 开发套件的各种性能。

6.10 设计支持

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

- 知识基础文章 — 包括产品系列的技术文章或有助于搜索各种 CapSense 相关主题。
- CapSense 应用手册 — 包括与本文档中的信息密切相关的广泛应用笔记。
- 白皮书 — 了解高级电容式触摸接口主题。
- 赛普拉斯开发社区 — 与赛普拉斯技术社区联系并交换信息。
- CapSense 产品选型指南 — 请参见赛普拉斯 CapSense 产品线完整的产品系列。
- 视频库 — 使用视频教程提高学习速度。
- 质量和可靠性 — 在质量网站查找产品可靠性和合格性的报告。赛普拉斯致力于满足客户的所有要求。
- 技术支持 — 世界一流的在线技术支持。

术语表

AMUXBUS

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

SmartSense™ 自动调校

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

基准线

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

按键或按键 widget

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

计数差值

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

电容传感器

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

CapSense®

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

CapSense 机械按键替换（MBR）

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

中心或中心位置

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

补偿 IDAC

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

CSD

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

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

去抖动

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

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

驱动屏蔽（Driven-Shield）

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

电极

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

手指阈值

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

组合传感器

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

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

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

手势

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

保护传感器

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

填充或网格地填充或网格布的铺设

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

迟滞

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

IDAC（电流输出的数模转换器）

PSoC 中的可编程恒流源，用于 CapSense 和 ADC 操作。

防水功能

存在水滴、水流或薄雾时，电容感应系统仍能够正常工作的能力。

线性滑条

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

低基准线复位

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

手动调校

指的是手动设置（或调校）CapSense 参数的过程。

矩阵按键

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

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

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

调制电容（CMOD）

在自电容感应模式下 CSD 模块操作所需要的外部电容。

调制器时钟

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

Modulation IDAC（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安排以检测手指的物理位置。

原始计数

代表传感器物理电容的 CapSense 硬件模块的未处理数值输出。

刷新间隔

传感器两次连续扫描间的时间。

扫描分辨率

由 CSD 模块生产的原始计数分辨率（单位为位）。

扫描时间

完成传感器的扫描过程所需要的时间。

自电容

与电路接地和电极相关的电容。

灵敏度

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

感应时钟

用来实现 CSD 感应方法的开关电容前端的时钟源。

传感器

请参见电容传感器。

传感器自动复位

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（Driven-Shield）。

屏蔽槽电容（C_{SH}）

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

信号（CapSense 信号）

计数差值还被称为信号。请参见计数差值。

信噪比（SNR）

有手指触摸时的传感器信号与无手指触摸时的传感器信号间的比例。

滑条分辨率

表示滑条上需要处理的手指位置总数的参数。

触摸板

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

触摸板

请参见触摸板。

调校

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

V_{REF}

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

Widget

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

文档修订记录

文档修订记录

文档标题：AN66269 — CY8C20x34 CapSense®设计指南 文档编号：001-79112			
版本	**ECN**	**提交日期**	**变更说明**
**	3617684	05/15/2012	本文档版本号为 Rev**，译自英文版 001-66269 Rev*D。
*A	4393776	05/30/2012	Sunset Review。
*B	4768892	05/19/2015	本文档版本号为 Rev*B，译自英文版 001-66269 Rev*G。
*C	5310423	06/15/2016	本文档版本号为 Rev*C，译自英文版 001-66269 Rev*I。
*D	6614199	07/05/2019	本文档版本号为 Rev*D，译自英文版 001-66269 Rev*J。