



AN90071-CY8CMBR3xxx

CapSense®设计指南

文档编号: 001-91599 版本*C

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

© 赛普拉斯半导体公司，2014-2018 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）

（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

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

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

目录

1. 简介	6
1.1 摘要	6
1.2 简介	6
1.3 CY8CMBR3xxx 系列特性概述	7
1.4 CY8CMBR3xxx 系列的特性比较	9
1.5 CY8CMBR3xxx CapSense 的系统概述	10
1.6 CapSense 设计流程	10
2. CapSense 技术	13
2.1 CapSense 基本原理	13
2.2 CY8CMBR3xxx 中的电容触摸感应方法	15
2.3 CapSense 调试	16
2.4 SmartSense 自动调试	16
2.5 传感器类型	17
2.5.1 按键（零维传感器）	17
2.5.2 滑条（一维传感器）	17
2.5.3 接近（三维传感器）	19
2.6 防水功能	19
2.6.1 屏蔽电极	20
2.6.2 保护传感器	21
3. CapSense 原理图设计	22
3.1 原理图设计注意事项	22
3.1.1 选择特殊输出引脚	22
3.1.2 选择传感器引脚	23
3.1.3 选择滑条引脚	24
3.1.4 选择 GPO	25
3.2 原理图检查表	26
3.3 示例原理图	26
3.3.1 手机的触摸按键	26
3.3.2 家用电器中的触摸按键	28
3.3.3 接近感应	29
4. CapSense 的布局指南	31
4.1 设计工具箱	31
4.1.1 通用布局指南	31
4.1.2 布局估计	34

4.1.3	C _P 、功耗以及响应时间计算	35
4.1.4	验证设计	38
4.2	接近传感器的布局指南	40
4.2.1	传感器设计指南	40
4.2.2	金属物体对接近距离产生的影响	42
4.3	防水性的布局指南	43
4.3.1	屏蔽电极结构	43
4.3.2	保护传感器	44
4.4	示例布局	45
4.4.1	FPC 上的触摸按键	45
4.4.2	FR4 PCB 上的触摸按键	46
4.4.3	FPC 上的接近传感器	47
4.4.4	FR4 和 FPC 上的 CapSense 设计	48
5.	配置 CY8CMBR3xxx	49
5.1	CY8CMBR3xxx 的配置	49
5.1.1	传感器配置	49
5.1.2	GPO 配置	53
5.1.3	蜂鸣器配置	56
5.1.4	器件配置	58
5.2	配置 CY8CMBR3xxx	59
5.2.1	使用 EZ-Click 配置 CY8CMBR3xxx	60
5.2.2	使用第三方编程器配置 CY8CMBR3xxx	62
5.2.3	使用主机 API 配置 CY8CMBR3xxx	62
5.2.4	使用 BCP (Bridge Control Panel) 软件配置 CY8CMBR3xxx	68
6.	CapSense 功能调试	77
6.1	通用注意事项	77
6.1.1	信号、噪声和信噪比	77
6.2	调试按键、滑条和保护传感器	79
6.2.1	确保最低信噪比的方法	79
6.2.2	按键阈值参数	80
6.2.3	设置按键、滑条和保护传感器的阈值	82
6.3	调试接近传感器	84
6.3.1	确保最低信噪比的方法	84
6.3.2	高级低通 (ALP) 滤波器	86
6.3.3	接近传感器阈值参数	89
6.3.4	设置接近传感器阈值参数	90
7.	低功耗设计中的注意事项	92
7.1	CY8CMBR3xxx 的工作模式	92
7.1.1	活动模式	93
7.1.2	查找触摸 (Look-for-touch) 模式	93
7.1.3	查找接近 (Look-for-Proximity) 模式	94
7.1.4	深度睡眠模式	94

7.1.5	配置模式	94
7.2	影响功耗的因素	94
7.3	使功耗降低的系统设计建议.....	95
8.	资源	96
8.1	网站	96
8.2	数据手册	96
8.3	寄存器技术参考手册	96
8.4	设计工具箱.....	96
8.5	EZ-Click™ 2.0	96
8.6	开发套件	96
8.7	设计支持	97
9.	附录	98
9.1	顶层 API.....	98
9.2	底层 API.....	104
	术语表	106

1. 简介



1.1 摘要

本指南介绍的是如何使用 CY8CMBR3xxx CapSense 控制器系列设计电容触摸传感应用。CY8CMBR3xxx CapSense 控制器系列产品具有绝佳的信噪比(SNR)、一流的防水功能，并能连接各类传感器，如按键、滑条和接近传感器。本指南说明了 CY8CMBR3xxx CapSense 的设计流程、设计工具、设计注意事项和性能调试。

1.2 简介

电容触摸传感器是用户界面设备，通过使用人体的电容来检测在传感器上或周围是否有手指。电容传感器外观漂亮，容易使用，且寿命长。赛普拉斯CapSense解决方案为您的产品提供精致、可靠和使用方便的电容式触摸感应功能。赛普拉斯CapSense解决方案已经替换了四十亿个机械按键。

CY8CMBR3xxx CapSense Express™控制器通过使用电容式触摸感应技术，带来了高级且容易实现的用户界面解决方案。该寄存器可配置的控制器系列可支持多达16个电容式感应的输入，消除设计周期中费时的固件开发和调试过程。因此，当需要实现电容按键、滑条和接近感应解决方案时，CapSense控制器是理想的选择，可以带来最小的开发周期时间。CY8CMBR3xxx控制器可以消除由薄雾、湿气、水滴、液体流导致的误触摸，因而可以使能开发强大防水功能的CapSense应用。

本设计指南说明了如何使用 CapSense 控制器的 CY8CMBR3xxx 系列来实现电容式触摸感应功能。该指南假设您熟悉赛普拉斯的 CapSense 技术。如果您不熟悉 CapSense 技术，请参考 [CapSense 入门指南](#) 了解其基础。

通过本指南，您会了解：

- CY8CMBR3xxx CapSense 控制器系列的特性
- CY8CMBR3xxx 系列使用 CapSense 技术
- 原理图和布局设计注意事项
- 如何配置 CY8CMBR3xxx 系列
- 性能调试
- 低功耗设计注意事项

1.3 CY8CMBR3xxx 系列特性概述

CY8CMBR3xxx 系列控制器是高性能、低功耗的 CapSense Express™ 控制器，通过使用电容式触摸感应技术，可以开发出高性能且易实现的用户接口解决方案。

CY8CMBR3xxx 系列具有以下特性：

- 寄存器可配置的 CapSense Express 控制器
 - ☐ 可通过 I²C 接口进行配置
 - ☐ 不需要固件开发或器件编程
 - ☐ 支持多达 16 个电容式感应输入
 - ☐ 支持多达 8 个通用输出（GPO）
 - ☐ GPO 可连接到 CapSense 传感器，也可由主机处理器控制
 - ☐ GPO 支持直接 LED 驱动功能
- SmartSense™ 自动调试
 - ☐ CapSense 算法可以连续补偿由系统、生产过程和环境的变化引起的影响
 - ☐ 自动配置电容按键参数，以得到优化性能
 - ☐ 不需要手动调试系统
 - ☐ 寄生电容（C_P）的较广工作范围（5 到 45 pF）
 - ☐ 如果需要，允许用户手动设置各阈值
- 高级功能
 - ☐ 接近触摸时的唤醒性能
 - 发生接近事件时，使系统从低功耗模式唤醒到活动模式
 - ☐ 防水功能
 - 存在水滴、水流或薄雾的情况下，电容式按键仍可以正常工作
 - ☐ GPO 引脚上的 PWM 输出
 - 传感器为 ON/OFF 状态时，用户可以配置 LED 的亮度
 - ☐ 侧向传感器抑制（FSS）
 - 区分紧密排列按键时的触摸状态
 - ☐ 提高信噪比的固件滤波器
 - ☐ 通过蜂鸣器信号输出可以发出音频触摸反馈
 - ☐ 使用外部电阻桥接输出模拟电压
 - ☐ 主机中断输出，向主机警报传感器状态的任何更改
 - ☐ 通过 I²C 接口获得的系统诊断数据
 - 简化生产线测试和系统调试的流程
- 抗噪能力
 - ☐ 伪随机序列（PRS）时钟源，用于最小化电磁干扰
 - ☐ 电磁兼容性（EMC）特性针对外部辐射噪声和外部传导噪声提供卓越的抗噪能力

- 系统诊断可检测：
 - ☐ 传感器短接到 VDD 和接地
 - ☐ 传感器与传感器短接
 - ☐ 传感器短路屏蔽
 - ☐ 错误的调制电容值 (C_{MOD})
 - ☐ 超出范围的传感器寄生电容 (C_P)
- I²C 接口
 - ☐ 支持达 400 kHz 的速度
 - ☐ 硬件地址匹配时唤醒
 - ☐ 无需时钟延展
- 宽广的工作电压范围
 - ☐ 1.71 V ~ 5.5 V 的工作范围
- 低功耗
 - ☐ 支持各种操作模式以降低器件的功耗
 - ☐ 扫描周期为 120 ms 时, 每个传感器的平均电流消耗将为 22 μ A
 - ☐ I²C 为 ON 时, 深度睡眠电流为 2.5 μ A
- 工业温度范围: -40 °C 到+85 °C
- 封装选项
 - ☐ 8 引脚 SOIC (150 mil)
 - ☐ 16 引脚 SOIC (150 mil)
 - ☐ 16 引脚 QFN (3 × 3 × 0.6 mm)
 - ☐ 24 引脚 QFN (4 × 4 × 0.6 mm)

1.4 CY8CMBR3xxx 系列的特性比较

表 1-1 对 CY8CMBR3xxx 系列的各种控制器特性进行了比较。这些控制器可以根据传感器数量和支持的功能区分。通过该表可以选择满足您设计要求的器件。

表 1-1.CY8CMBR3xxx 控制器的比较

序号	特性	MBR3116	MBR3106S	MBR3110	MBR3108	MBR3102	MBR3002
1	最大按键数量	16	11	10	8	2	2
2	最大滑条段数量	×	10	×	×	×	×
3	最大接近传感器数量	2	2	2	2	2	×
4	屏蔽电极	✓	✓	✓	✓	✓	×
5	保护传感器	✓	×	✓	✓	×	×
5	接近触摸时的唤醒	✓	✓	✓	✓	✓	×
6	防水功能	✓	×	✓	✓	✓	×
7	GPO/LED 驱动的最大输出数量	8	0	5	4	1	2
8	LED 亮度控制	✓	×	✓	✓	✓	×
9	I ² C 接口	✓	✓	✓	✓	✓	×
10	蜂鸣器驱动输出	✓	✓	✓	✓	×	×
11	主机中断输出	✓	✓	✓	✓	×	×
12	FSS	✓	✓	✓	✓	✓	×
13	中值滤波器和 IIR 滤波器	✓	✓	✓	✓	✓	✓
14	高级低通滤波器	✓	×	✓	✓	✓	×
15	电磁兼容性 (EMC)	✓	✓	✓	✓	✓	×
16	灵敏度控制	✓	✓	✓	✓	✓	×
17	自动手指阈值	✓	✓	✓	✓	✓	✓
18	灵活的手指阈值	✓	✓	✓	✓	✓	×
19	LED 点亮保持时间	✓	×	✓	✓	✓	×
20	切换模式	✓	×	✓	✓	✓	×
21	系统诊断	✓	✓	✓	✓	✓	✓
22	传感器自动复位	✓	✓	✓	✓	✓	✓

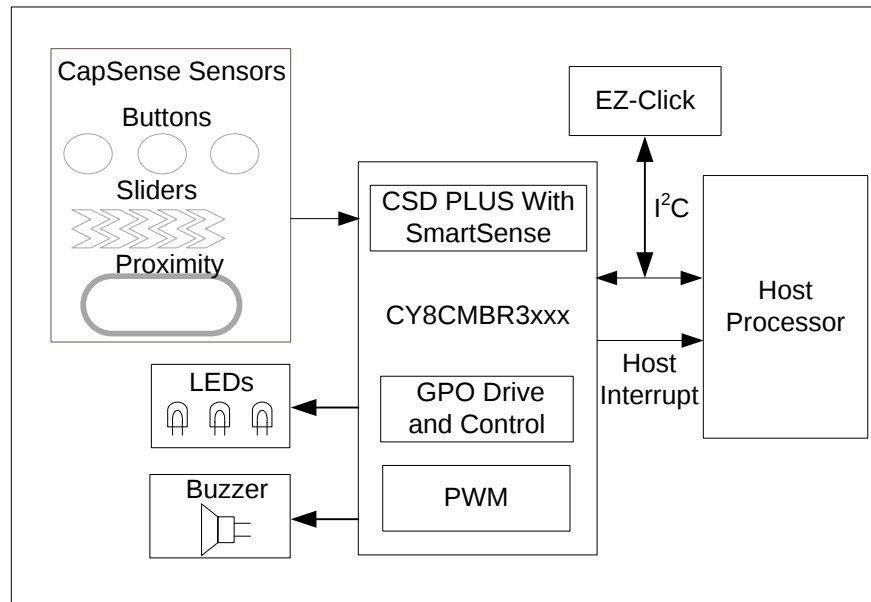
注意:

- CY8CMBR3102控制器不支持保护传感器, 因此它仅针对水滴提供防水功能而不为水流提供防水功能。
- CY8CMBR3002的传感器自动复位时间固定为20秒且该值不可配置。

1.5 CY8CMBR3xxx CapSense 的系统概述

图 1-1 显示的是使用 CY8CMBR3xxx CapSense 控制器的 CapSense 应用的典型系统级框图。CY8CMBR3xxx 控制器作为从器件工作，在用户界面屏幕上检测手指触摸，并向主机处理器报告触摸状态。每次传感器状态更改时，CY8CMBR3xxx 控制器将中断脉冲发送到主机处理器，以便主机可以通过 I²C 接口读取传感器状态。CY8CMBR3xxx 控制器通过驱动 GPO 和专用蜂鸣器输出提供触摸反馈。EZ-Click™ 工具通过 I²C 接口配置和调试 CY8CMBR3xxx 控制器。

图 1-1. 使用 CY8CMBR3xxx 控制器的 CapSense 系统框图



1.6 CapSense 设计流程

图 1-2 显示的是使用 CY8CMBR3xxx 控制器的电容式感应的典型产品设计周期流程。表 1-2 列出了受支持的文档和体系，用于设计使用 CY8CMBR3xxx 控制器的 CapSense 应用。

图 1-2. 典型的 CapSense 产品设计流程

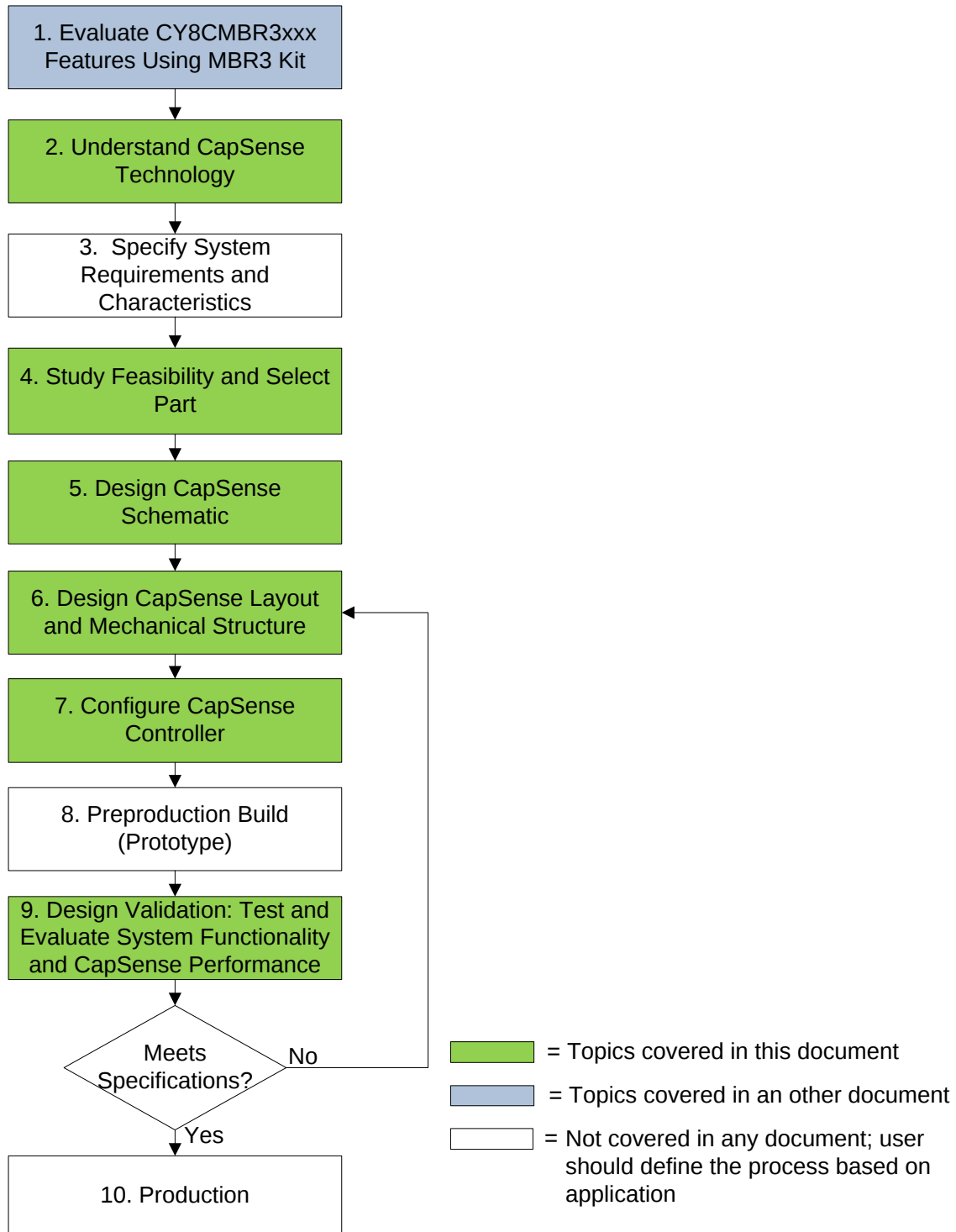


表 1-2.支持文档和生态系统

流程图中的各步骤	支持赛普拉斯的文档	
	名称	章节
1. 使用 MBR3 套件评估 CY8CMBR3xxx 的性能	CY3280-MBR3 评估套件的用户指南	NA
2. 了解 CapSense 技术	CapSense 入门 CY8CMBR3xxx CapSense 设计指南（本文档）	第二章节
3. 特定系统要求和特性	未在任何文档中描述；用户应该根据应用定义该过程	NA
4. 可行性研究：器件选择	CY8CMBR3xxx CapSense 设计指南（本文档） CY8CMBR3xxx 数据手册	第一章节
5. 设计 CapSense 原理图	CY8CMBR3xxx CapSense 设计指南（本文档）	第三章节
6. 设计 CapSense 布局和机械结构	CY8CMBR3xxx CapSense 设计指南（本文档）	第四章节
7. 配置 CapSense 控制器	CY8CMBR3xxx CapSense 设计指南（本文档）	第五章节
8. 预生产（样板）	未在任何文档中描述；用户应该根据应用定义该过程	NA
9. 设计验证	CY8CMBR3xxx CapSense 设计指南（本文档）	第 6 章节
10. 生产	未在任何文档中描述；用户应该根据应用定义该过程	NA

2. CapSense 技术

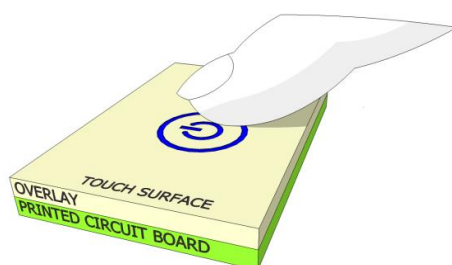


通过电容触摸感应技术可以测量一个板（传感器）和其环境之间的电容变化，这样能够检测在触摸界面上或附近是否有手指。

2.1 CapSense 基本原理

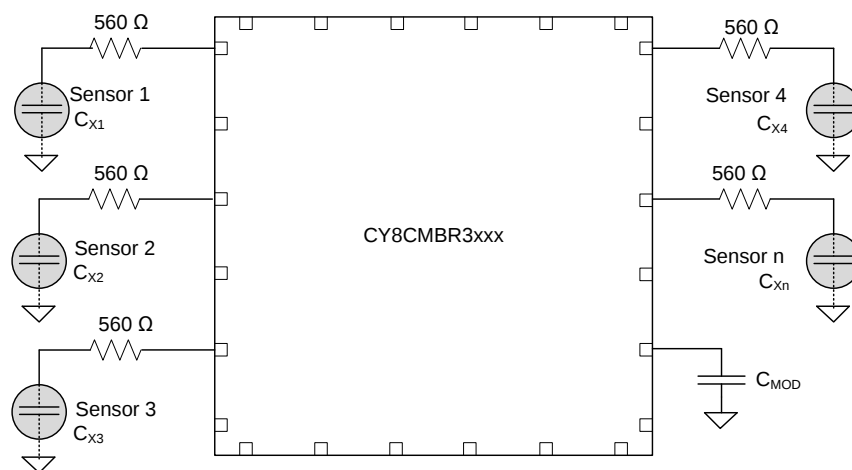
一个典型的 CapSense 传感器包括蚀刻在印刷电路板（PCB）表面上的铜盘。绝缘覆盖层作为传感器的触摸表面使用，如图 2-1 所示。

图 2-1. 电容式触摸感应



PCB 走线和过孔将传感器垫片连接到 CY8CMBR3xxx 控制器的 CapSense 输入引脚。每个传感器引脚的所有电容值可以模拟为等效的一个总电容值，从 C_{X1} 、 C_{X2} 到 C_{Xn} ，如图 2-2 所示。CY8CMBR3xxx 控制器中的 CapSense 电路将这些电容值转换为相应的数字计数。然后，控制器将处理这些数字计数，以检测触摸。

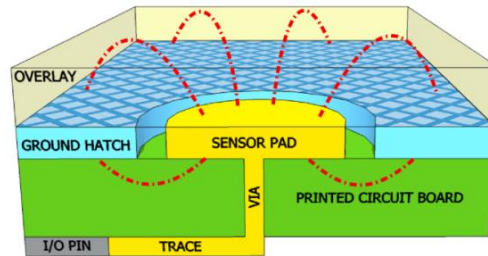
图 2-2. CapSense 在 CY8CMBR3xxx 器件中的实现



注意： CY8CMBR3xxx 控制器需要一个连接 CMOD 引脚和地面的外部电容 C_{MOD} 。

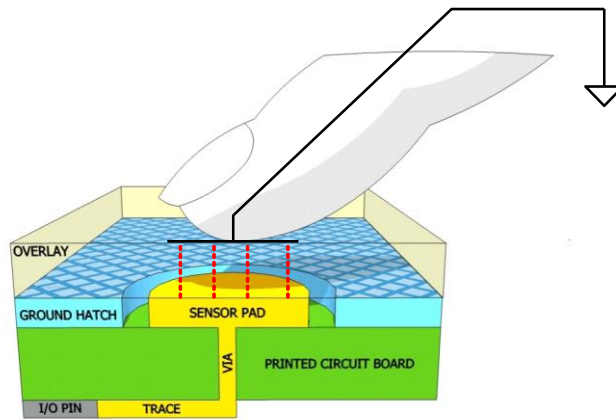
手指未接触传感器时的传感器电容称为寄生电容， C_P 。寄生电容由传感器（包括传感器垫片、走线和过孔）和系统中的其他导体（如接地层、走线、产品机箱或外壳中的任何金属等）之间的电场共同产生的。CY8CMBR3xxx 控制器的传感器引脚和内部电容也会带来寄生电容。然而，与传感器电容相比，这些内部电容是非常小的。图 2-3 显示的是 CY8CMBR3xxx 传感器引脚如何通过走线和过孔连接到传感器垫片。一般情况下，围绕传感器垫片的接地网络将其与其他传感器及走线分开。尽管该框图显示了传感器垫片周围的磁场线，但实际磁场比这复杂得多。

图 2-3.寄生电容



当手指触碰传感器表面时，人体的导电性质和大质量形成接地的导电层（平行于传感器垫），如图 2-4 所示。

图 2-4.典型的 CapSense PCB 与通过手指激活的传感器之间的横截面图



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：

$$C_F = \frac{\epsilon_0 \epsilon_r A}{d} \quad \text{公式 1}$$

其中： ϵ_0 = 空气介电常数

ϵ_r = 覆盖层的绝缘常数（相对介电常数）

A = 手指与传感器垫片覆盖层的接触面积

d = 覆盖层的厚度

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的电容，因此两者是同性质的。因此，当手指接触传感器时，传感器总电容 C_X 等于 C_P 与 C_F 之和。

$$C_X = C_P + C_F \quad \text{公式 2}$$

如果没有触摸， C_x 等于 C_P 。

CY8CMBR3xxx 控制器将电容 C_x 转换为相应的数字计数，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将表示有手指触摸。由于寄生电容 C_P 增加，因此 C_F 与 C_P 之间的比率会降低 — 电容的每一单位充电量的变化（与手指触摸相对应）相应降低。这样，当 C_P 增加时，将更难以检测触摸。

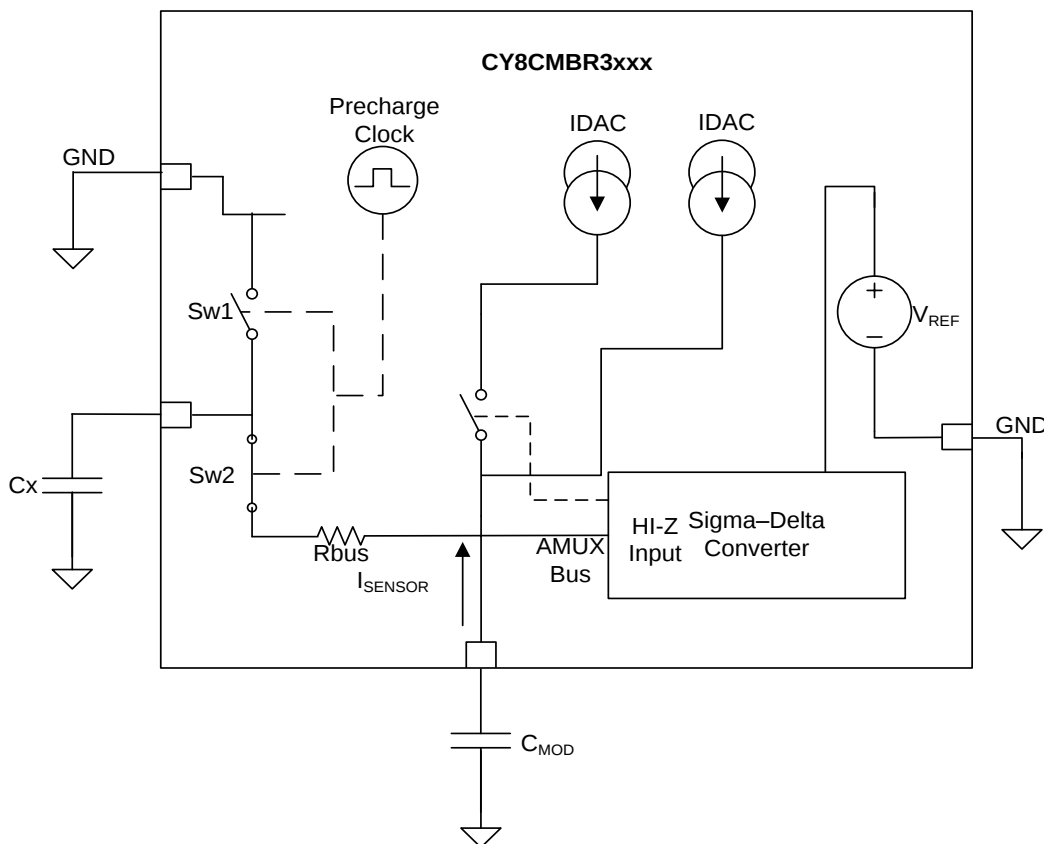
通常， C_P 比 C_F 大一个数量级以上。 C_P 的范围通常为 10 至 20 pF，但在特殊情况下可以高达 45 pF。 C_F 的范围通常为 100 至 400 fF。

2.2 CY8CMBR3xxx 中的电容触摸感应方法

CY8CMBR3xxx 控制器使用电容触摸感应方法 — **CapSense Sigma Delta PLUS (CSD PLUS)** — 将电容的变化转换为数字计数。CSD PLUS 触摸感应方法提供一个较好的信噪比 (SNR)，以确保在极端噪声环境下触摸的准确性。

CY8CMBR3xxx 器件中的 CSD PLUS CapSense 感应方法是将 C_x 整合至开关式电容电路中，如图 2-5 所示。通过开关 Sw1 和 Sw2，选择性地传感器 (C_x) 连接到 GND 和模拟复用器 (AMUX) 总线。Sw1 和 Sw2 由预充电时钟驱动，通过模拟复用器 (AMUX) 总线释放电流 (I_{SENSOR})。 I_{SENSOR} 的数量级与 C_x 的数量级成正比。Sigma-Delta 转换器对 AMUX 总线电压进行采样，并产生一个调制的位流（该位流控制恒流源，IDAC）。IDAC 对 AMUX 进行充电，以便维持 AMUX 总线电压的平均值为 V_{REF} 。传感器释放调制电容 (C_{MOD}) 中的电流 I_{SENSOR} 。 C_{MOD} 与 R_{bus} 共同构成一个低通滤波器，该滤波器削弱了 Sigma-Delta 转换器输入的预充电转换瞬态。

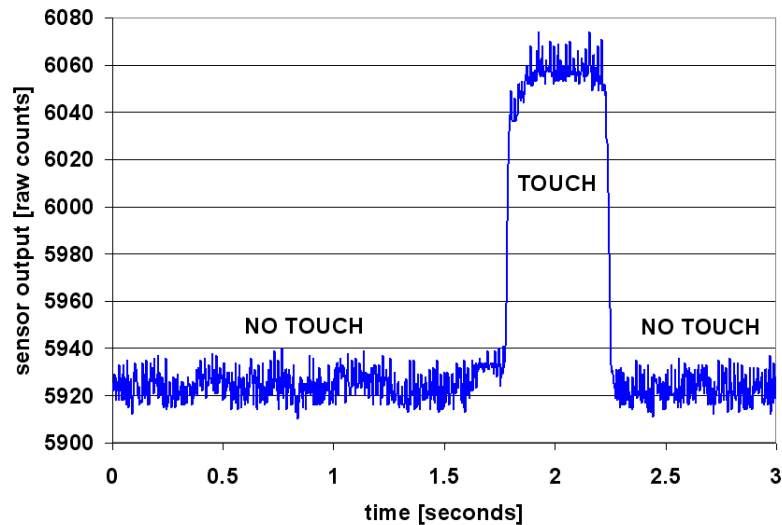
图 2-5.CSD PLUS 框图



为了维持 AMUX 总线的电压为 V_{REF} ，Sigma-Delta 转换器通过控制位流占空比将 $IDAC$ 输出电流与 I_{SENSOR} 进行匹配。Sigma-Delta 转换器在传感器扫描期间内存储位流，累积结果为数字原始计数输出，该值与 C_x 成正比。该原始计数由高级算法进行计算，以便求得传感器的状态。图 2-6 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 PLUS 原始计数。正如 [CapSense 基本原理](#) 所解释，手指触摸使 C_x 增加 C_F ，依次使原始计数按比例增加。通过

比较基于稳定状态下原始计数的变化量和预定阈值，高级算法能够确定传感器是在 ON（触摸）还是 OFF（未触摸）状态。更多有关原始计数、手指阈值和信噪比（SNR）的信息，请参考《CapSense 入门手册》。

图 2-6.手指触摸期间的原始计数



2.3 CapSense 调试

CapSense 系统性能取决于电路板布局、传感器尺寸、覆盖层材料和应用要求。这些因素在 [CapSense 的布局指南](#) 中介绍。除了这些因素以外，CSD PLUS 感应方法还需要设置一些硬件和软件参数，以确保稳健操作。为这些参数设置最佳值的过程称为手动调试。手动调试是一个费力和费时的过程。赛普拉斯 SmartSense 自动调试算法通过自动设置所有硬件和软件参数消除手动调试过程。

2.4 SmartSense 自动调试

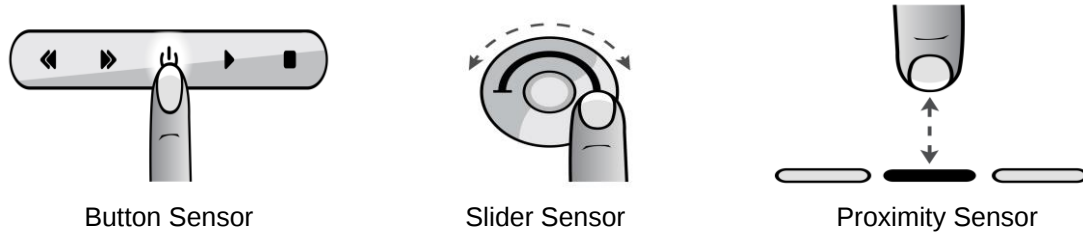
调试触摸感应用户界面对于确保系统正常运行和用户的良好体验非常重要。典型设计流程包括，在初始设计阶段和系统整合过程中调试传感器，并在投入量产前进行最后的生产微调。调试是一个重复过程，可能非常耗时。

SmartSense 自动调试是一个自动配置电容按键参数以得到最佳性能的 CapSense 算法。它连续补偿系统、生产过程和环境的变化，并简化用户界面的开发周期。它易于使用，并且在样板和制造阶段通过消除手动调试缩短了设计周期。SmartSense 自动调试功能在每个 CapSense 按键加电时对其进行自动调试，并维持运行时的最佳按键性能。SmartSense 自动调试适用于解决 PCB 和覆盖层的制造误差，这种方法能够自动消除噪声源（如 LCD 反相器、交流线和开关模式电源）产生的噪声。欲了解 SmartSense 算法的工作方式，请参考 [CapSense 入门指南](#)。

2.5 传感器类型

CY8CMBR3xxx 支持三种电容传感 — 按键、滑条和接近传感器，如图 2-7 所示。

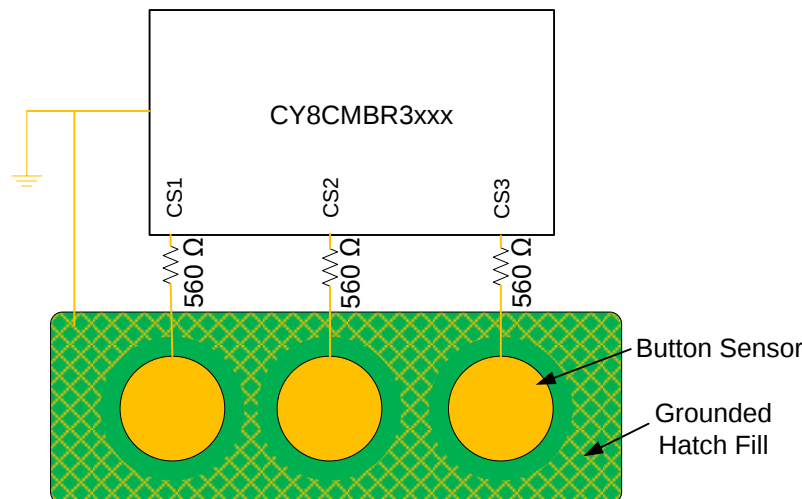
图 2-7.CY8CMBR3xxx 支持的各种电容传感器



2.5.1 按键（零维传感器）

在众多应用如家用电器、医疗器件、照明控制和许多其他产品中，CapSense 按键可替代机械按键。它是一种最简单的 CapSense 传感器，包括一个简单传感器。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，两种状态还分别称为 ON 和 OFF 状态。简单 CapSense 按键包含一个圆形的铜板。它连接到一个引脚（该引脚被标记为 CY8CMBR3xxx 控制器上的 CapSense 按键），如图 2-8 所示。在 CY8CMBR3xxx 中，CSx 引脚表示按键传感器可以连接到它。围绕该按键的接地覆铜将其与其他按键和走线分开。另外，通过一个圆形板可以将按键板和接地网格分开。有关布局建议，请查看[通用布局指南](#)章节中的内容。

图 2-8.简单 CapSense 按键



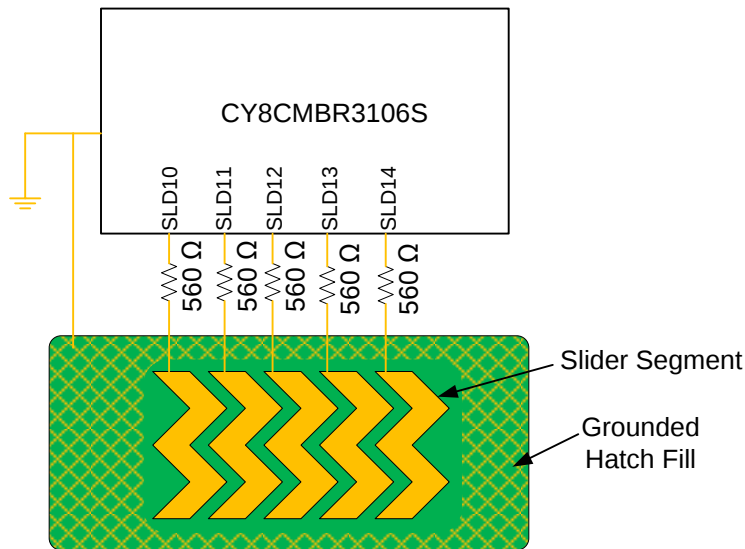
2.5.2 滑条（一维传感器）

当所需要的输出递增或递减时，将使用滑条。例如照明控制（调光器）、音量控制、图示均衡器和速度控制。滑条由一系列称为段（相邻放置）的电容传感器构成。触摸某一个滑条段也会激活邻近滑条段的一部分。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多于滑条段数量。例如，包含五段的滑条至少可以解析 100 个手指物理位置。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 2-9 显示。

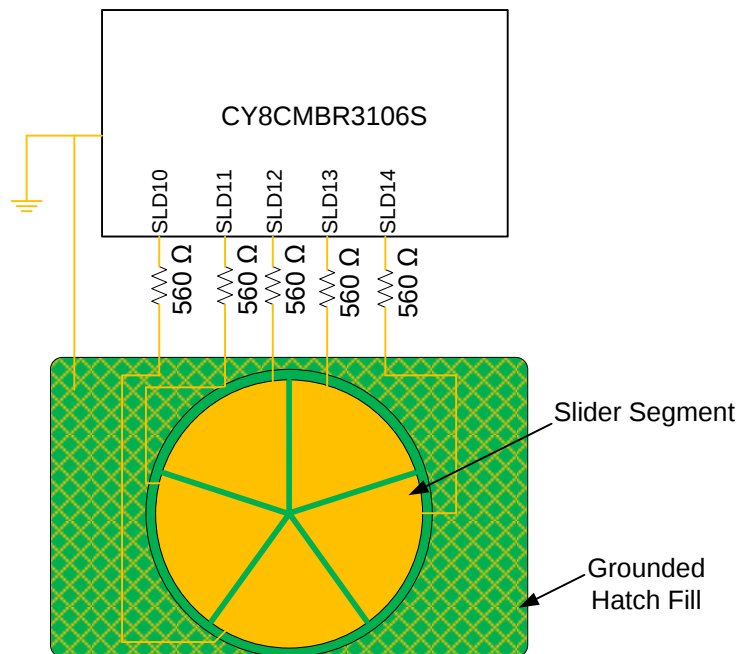
每个滑条段与 CY8CMBR3106S 控制器上用于滑条功能的引脚相连。在 CY8CMBR3106S 中，SLDxx 引脚表示一个滑条段可以连接到它。滑条段建议采用 Z 字形图案（双 V 形图案）。这种布局可以确保在触摸某段时，还可以部分触摸邻近的段。这样会有助于中心位置的计算。有关布局建议，请查看[通用布局指南](#)章节中的内容。

图 2-9.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 2-10 显示的是一个典型的辐射滑条。

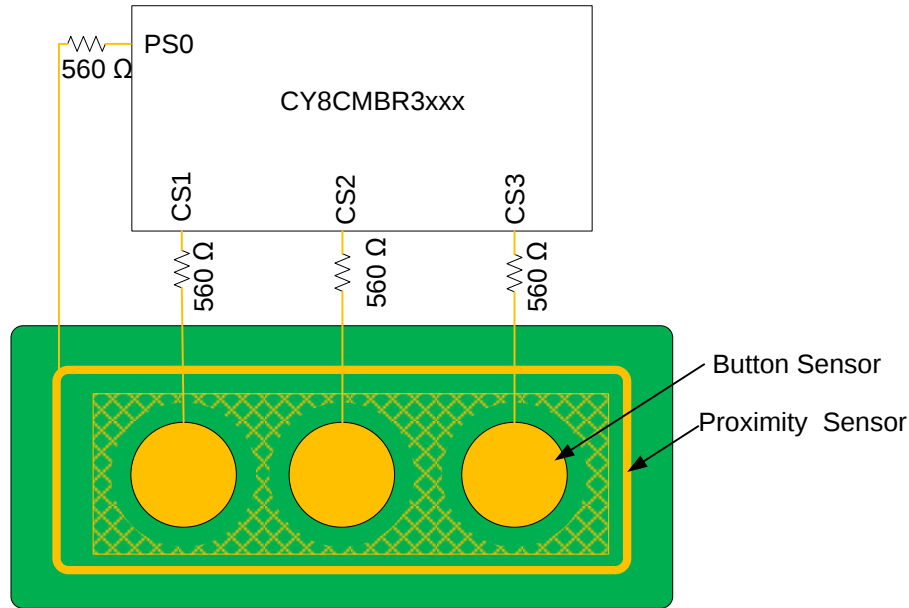
图 2-10.辐射滑条



2.5.3 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有手指。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。根据传感器的大小和结构，在几厘米到几十厘米的距离内，接近传感器可以检测到一个手指。每个接近传感器与 CY8CMBR3xxx 控制器上用于接近感应功能的引脚相连。在 CY8CMBR3xxx 控制器中，PSx 引脚表示接近传感器可以连接到该引脚。接近感应的电场设计要求的比按键和滑条大得多，它需要更大的传感器面积。接近传感器可以是 PCB 上的走线，也可以是简单的线环，如图 2-11 所示。更多有关信息，请参考[接近传感器的布局指南](#)。

图 2-11.接近传感器



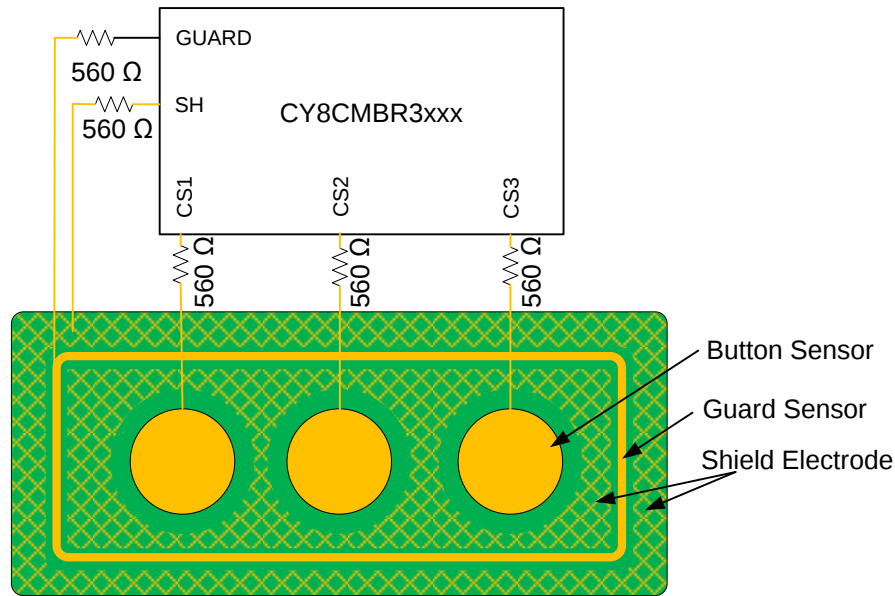
2.6 防水功能

在 CapSense 设计中，如果触摸界面上存在水膜或水滴，可能会发生误触摸感应。家用电器、汽车应用和工业应用均会涉及到水、冰和湿度变化的环境，传感器系统必须在这种环境下进行操作。对于这样的应用而言，屏蔽电极和保护传感器可以提供强健的触摸感应。

如果您的应用要求抵抗水滴和潮气的干扰，则应使用屏蔽电极。如果您的应用还需要接触界面能够耐水流，则应同时使用屏蔽电极和保护传感器。即使启用了 SHIELD，非活动传感器也会接地。因此，建议将按钮传感器相隔较远放置，使得存在于一个传感器上的液滴不会导致相邻传感器的错误触发。

注意：滑条传感器不支持防水功能。

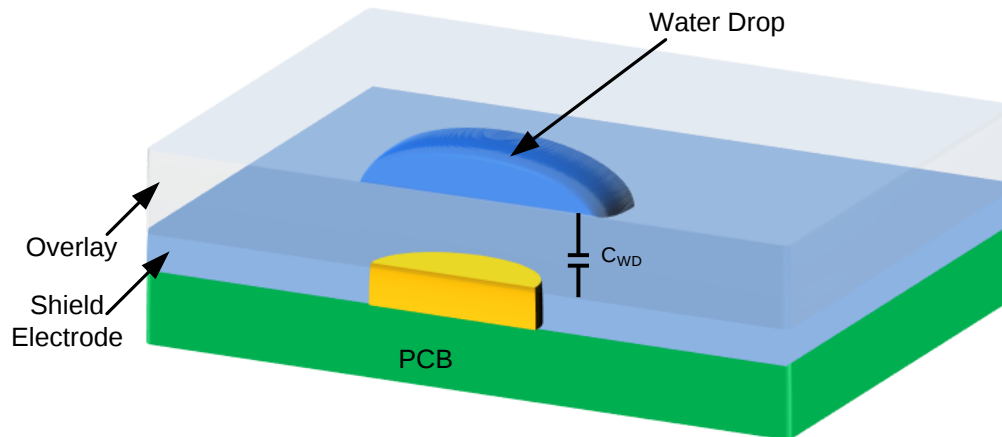
图 2-12. 连接到 CY8CMBR3xxx 的屏蔽电极 (SH) 和保护传感器 (GUARD)



2.6.1 屏蔽电极

屏蔽电极防止 CapSense 按键传感器检测由水滴引起的误触摸。当水滴落到覆盖层表面上时，屏蔽电极和传感器垫片之间的耦合增加了 C_{WD} ，如图 2-13 所示。

图 2-13. 存在水滴时的电容测量



- C_{WD} — 水滴与屏蔽电极之间的电容

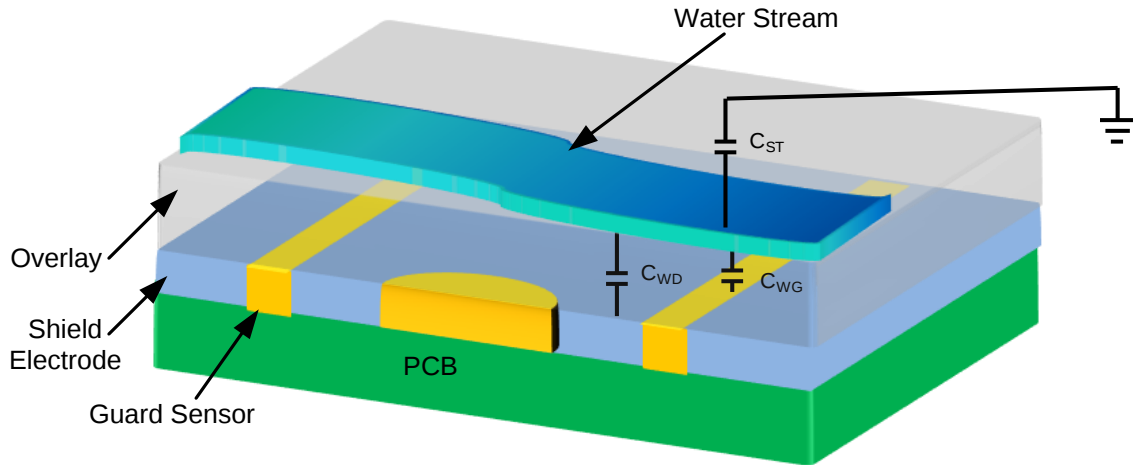
屏蔽电极的目的是在触摸传感器周围建立电场，用来削弱水滴产生的影响。屏蔽电极的工作方式是将触摸传感器的电压反映到屏蔽电极上。由于屏蔽电极和传感器由相同信号驱动，因此它们之间没有潜在区别，两者之间的任何电容不能引起电荷转移。这样，即使传感器和屏蔽区域上存在水膜或水滴，也不会改变传感器的电容，从而允许 CapSense 在有水膜或水滴的情况下正常工作。

2.6.2 保护传感器

保护传感器是围绕 PCB 上所有传感器的铜走线，如图 2-12 所示。该传感器用于检测水流的存在。水流出现在感应表面上时，在系统中将添加较大电容 (C_{ST})，如图 2-14。该电容可能比 C_{WD} 大数倍。正因为如此，屏蔽电极的影响完全被屏蔽，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，当保护传感器检测水流时，它将防止触发其他传感器。但，存在水流时不可检测到触摸。

在 MBR3 中，保护传感器（如果已启用）仅在活动和查找触摸模式下扫描。因此，当设备处于 Look-for-Proximity 模式时，当存在液体流时可能发生错误接近触发。

图 2-14. 存在水流时的电容测量



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

要实现防水性设计，请按照以下步骤进行操作：

1. 如果您的应用要求耐水滴和潮气，则应使用具有屏蔽电极的 CY8CMBR3xxx 控制器。如果您的应用还要求耐水流，则应使用具有屏蔽和保护功能的 CY8CMBR3xxx 控制器。请参考表 1-1 选择控制器。
2. 请按照防水性的布局指南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。
3. 使能 EZ-Click 中的屏蔽和保护传感器特性，如屏蔽和保护传感器配置章节中所述。
4. 调试保护传感器，如调试按键、滑条和保护传感器章节中所述。

3. CapSense 原理图设计



CY8CMBR3xxx 控制器支持专用引脚上的各按键、滑条和接近传感器。另外，在该控制器中，支持按键、滑条和接近传感器的数量是固定的。设计 CapSense 应用的第一个步骤是选择传感器引脚。本章节为选择 CapSense 原理图的传感器引脚提供了指南。

3.1 原理图设计注意事项

使用 CY8CMBR3xxx 控制器设计原理图时需要进行下列步骤：

1. 特殊输出的引脚选择
2. 传感器引脚选择
3. GPO 引脚选择
4. 电源和接地连接
5. I²C 线连接

本章节介绍的是如何选择特殊输出、传感器和 GPO 引脚。更多有关电源、接地和 I²C 线的连接的信息，请参考 [CY8CMBR3xxx 数据手册](#) 中介绍的内容。

3.1.1 选择特殊输出引脚

屏蔽电极、保护传感器、蜂鸣器和主机中断引脚被称为特殊输出引脚（SPO）。这些特性被复用到传感器引脚，且仅在专用的引脚上受支持。因此，在连接传感器引脚之前，需要选择这些特殊输出引脚。

例如，考虑一个需要七个按键、蜂鸣器输出和防水功能的 CapSense 设计。如表 1-1 所列，CY8CMBR3116 和 CY8CMBR3110 控制器满足该设计的要求。为了更好的理解，选择 CY8CMBR3110 为例说明。设计原理图中的第一个步骤是选择蜂鸣器（BUZ）、屏蔽（SH）和保护（GUARD）引脚，因为这些引脚被复用到传感器和 GPO 引脚，如图 3-1 所示。有关同一个引脚上受支持的各种特性的信息，请参考 [CY8CMBR3xxx 数据手册](#) 中的“引脚分布”章节。选择特殊的输出引脚后，继续选择按键传感器 CS0、CS1、CS3、CS5、CS6、CS7 和 CS8，如图 3-2 所示。

图 3-1. 单引脚上受支持的各种特性

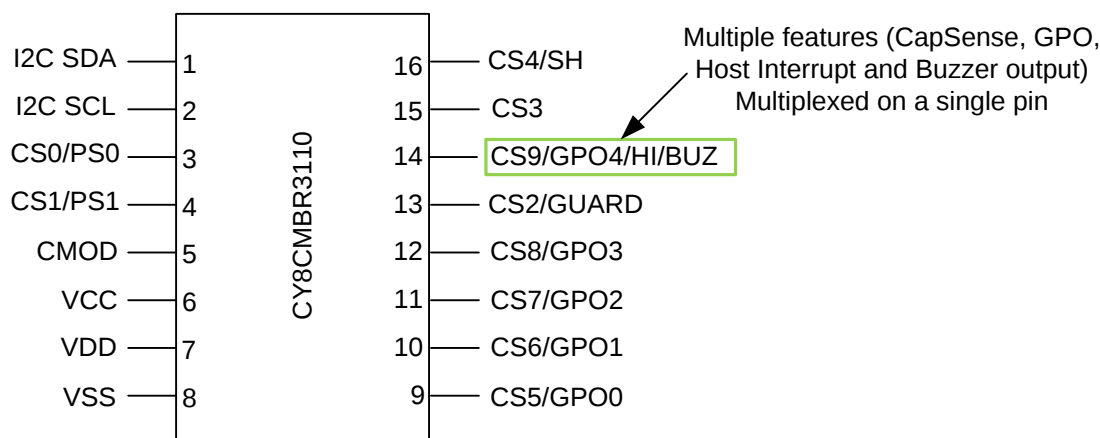
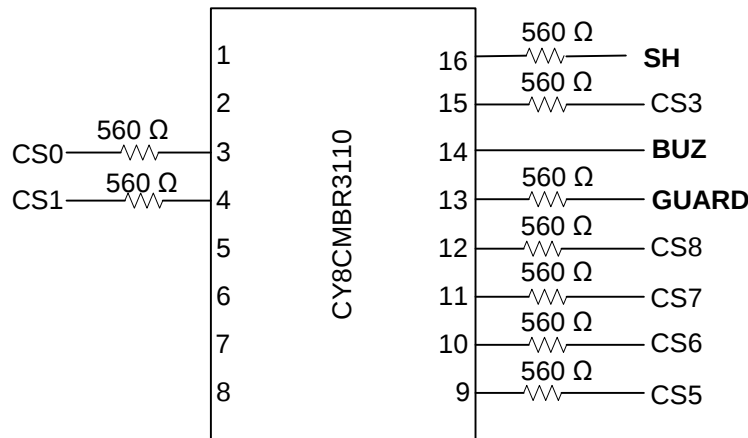


图 3-2.特殊输出（SPO）的引脚选择



3.1.2 选择传感器引脚

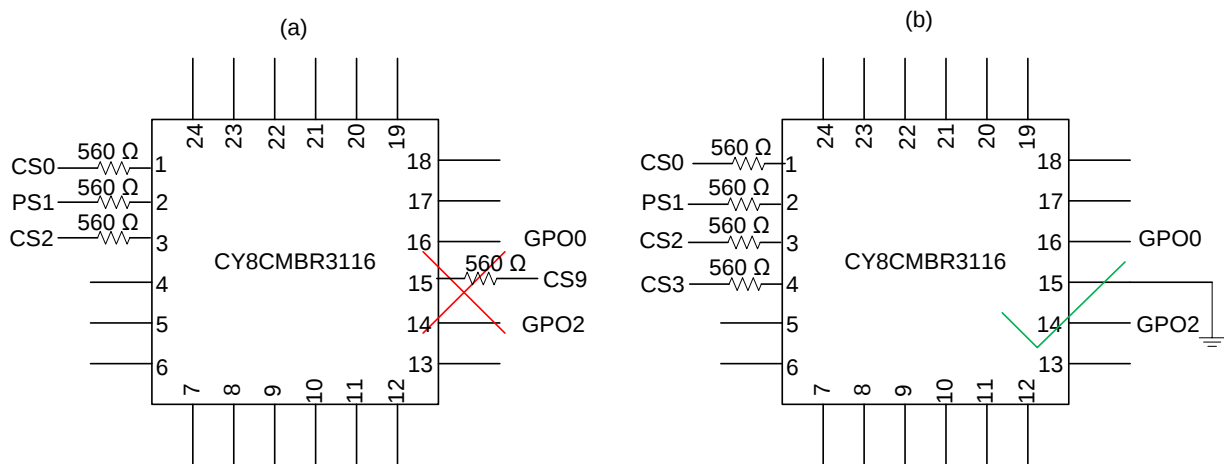
当选择传感器引脚时，请勿选择接近 GPO 引脚或 GPO 引脚之间的传感器引脚（CSx、SLDx、PSx 或 GUARD）。选择接近 GPO 引脚的传感器引脚会导致一个串扰。如果传感器走线和 GPO 走线彼此并行，那么切换 GPO 时会使接近 GPO 引脚的传感器进行误触发。有关避免串扰的指南，请参考 [CapSense 入门](#) 中的“PCB 布局指南”部分。

例如，图 3-3 (a) 显示的是一个原理图，其中传感器 CS9 位于 GPO0 和 GPO2 引脚之间。在这种情况下，如果 CS9 走线与 GPO0 或 GPO2 走线彼此并行，GPO0 或 GPO2 切换时，传感器 CS9 会进行误触发。如果不选择接近 GPO 引脚的传感器引脚 CS3，如图 3-3 (b) 所示，或确保传感器走线和 GPO 走线不彼此并行，则可以避免误触发。

如果您的设计需要一个接近传感器，在选择按键（CSx）引脚之前，需要选择接近传感器引脚（PSx），因为接近传感器仅在两个引脚上受支持。

要想增加抗噪能力，需要将 560 Ω 串联电阻连接到传感器引脚上。如果未使用 CapSense 引脚，建议将其配置为传感器引脚并连接至地。有关如何将一个引脚配置为 CapSense 输入的信息，请参考 [传感器配置](#) 章节中介绍的内容。

图 3-3. 传感器引脚选择指南



3.1.3 选择滑条引脚

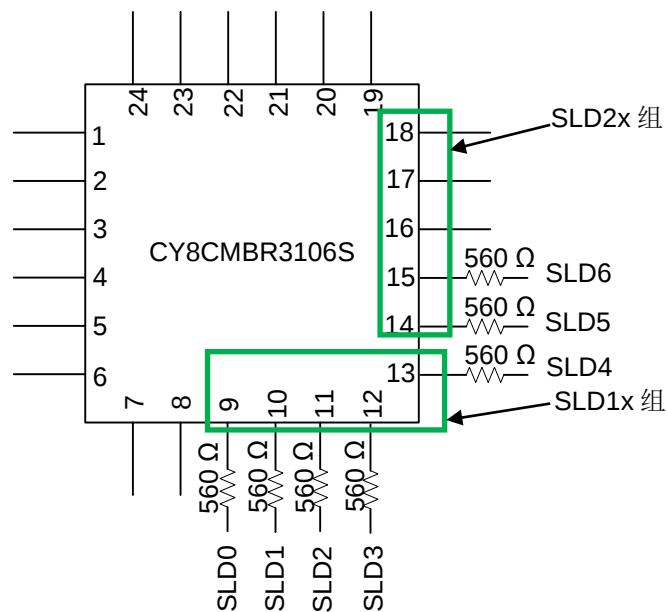
要想选择滑条引脚，请参考表 3-1.

表 3-1.滑条引脚选择指南

滑条段数量	建议
小于或等于 5	从 SLD1x 组中选择引脚。SLD10 作为第一个滑条段并且 SLD14 为第五个滑条段。
多于五	从 SLD1x 组中为前五个滑条段选择五个引脚，然后从 SLD2x 组中为剩下的滑条段选择引脚。SLD10 是滑条的第一段，SLD20 是第六段，SLD24 是第十段。

图 3-4 显示的是七段滑条的原理图示例。在该原理图中，滑条段 SLD0 到 SLD4 使用控制器 SLD1x 组中的引脚，SLD5 和 SLD6 段使用 SLD2x 组中的引脚。可以将 SLD2x 组中剩下的各引脚（SLD22、SLD23 和 SLD24）配置为按键。如果未使用，应该将它们配置为按键并连接至地¹。

图 3-4.滑条引脚选择

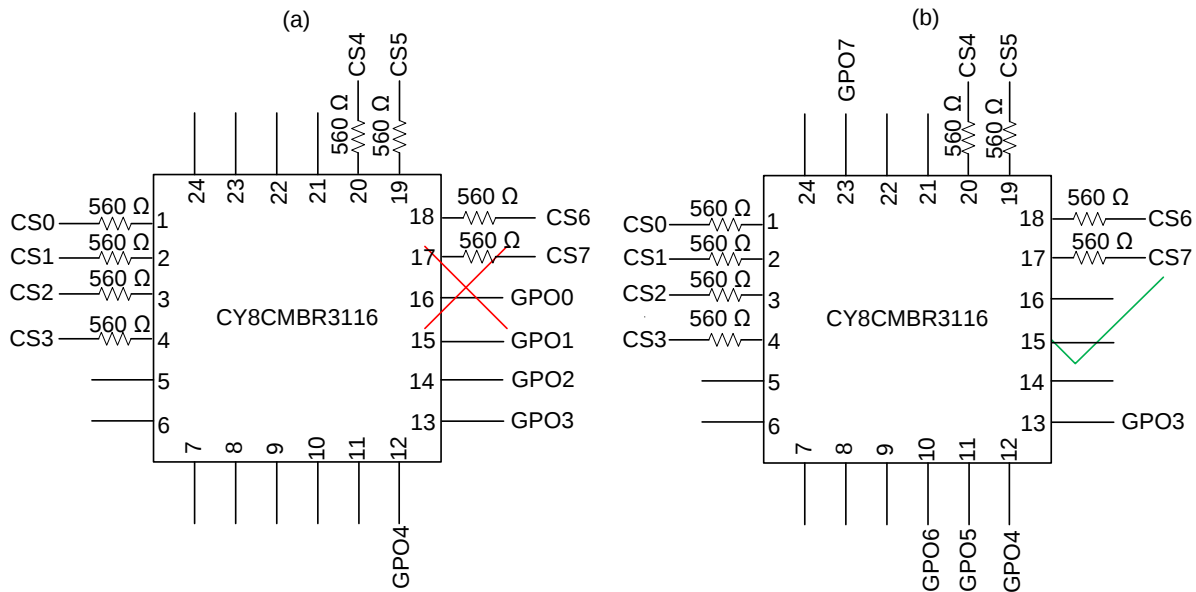


¹ 当不使用引脚#18（即 CY8CMBR3106S 中的 CS15/SLD24）时，应将其处于悬空状态。在启动过程中，该 I/O 作为复位（AXRES）引脚使用。请确保上电时该引脚不接地，以便器件可以正确启动。启动后，该 I/O 按照引脚名称运行。

3.1.4 选择 GPO

如果 GPO 由主控制器控制，那么任何有效 GPO 可用于提供触摸反馈。建议根据 GPO7 到 GPO0 的顺序选择 GPO。例如，如果您的设计需要八个按键和五个 GPO，则选择 CS0-CS7 用于按键和 GPO7-GPO3 用于 GPO。这样可以避免选择接近 GPO 引脚的传感器引脚，如图 3-5 所示。

图 3-5. GPO 主机控制被使能时选择 GPO 引脚



如果禁用了 GPO 主机控制，那么 CS0 到 CS7 的每一个传感器将分别映射到 GPO0 到 GPO7。这时，选择相应于已选传感器的 GPO。例如，当禁用 GPO 主机控制而且将 CS0、CS1 和 CS2 配置为按键时，选择 GPO0、GPO1 和 GPO2 作为 GPO 引脚，用于驱动 LED。

CY8CMBR3xxx 控制器支持拉电流和漏电流配置，以便可以在 GPO 引脚上驱动 LED。由于 GPO 的灌电流比它的拉电流更高，因此建议将 LED 连接到灌电流。有关拉电流和灌电流的最大限制，请参考 [CY8CMBR3xxx 数据手册](#) 中介绍的直流 I/O 端口规范。

要想限制该电流，需要根据供电电压添加合适的串联电阻值。如果您的设计需要灌电流超过最大限制，则将外部晶体管连接到 GPO 引脚并通过使用 [EZ-Click](#) 将 GPO 配置为强驱动模式。

3.2 原理图检查表

请使用表 3-2 中的检查表验证您的原理图设计。

表 3-2.原理图设计检查表

序号	类目	建议/备注
1.	VDD ² 引脚上的去耦电容	并行连接 1 μ F 和 0.1 μ F 电容
2.	VCC ³ 引脚上的去耦电容	如果 VDD > 1.8 V，连接一个 0.1 μ F 电容 如果 VDD 范围为 1.71 V 到 1.89 V，则将该引脚短接到 VDD
3.	VDDIO ⁴ 引脚上的去耦电容	将 1 μ F 和 0.1 μ F 电容并联
3.	C _{MOD}	连接到 2.2 nF、5 V、X7R 或 NPO 电容
4.	CSx、PSx、SLDx、GUARD 和 SH 引脚上的串联电阻	将 560 Ω 电阻串连到该引脚
5.	I ² C 线上的串联电阻	将 330 Ω 电阻串连到 SDA ⁵ 和 SCL ⁶ 线
6.	I ² C 线上的上拉电阻	根据 I ² C 接口电压选择合适值
7.	HI 线上的上拉电阻	根据接口电压选择适当值
8.	未使用的 CapSense 引脚 ⁷ （CSx、PSx、SLDx 和 GUARD）	有关未使用的引脚连接信息，请参考 CY8CMBR3xxx 数据手册中“引脚分布”一节的内容。

3.3 示例原理图

3.3.1 手机的触摸按键

通过手机的触摸按键可以快速访问各种应用。一个典型的手机可拥有四个触摸按键并需要 LED 背光。

图 3-6 中显示了一个示例原理图，说明如何在手机中实现各触摸按键。

在图 3-6 中，CY8CMBR3108 控制器的配置如下：

- CS0-CS3: CapSense 按键
 - 每个 CapSense 引脚必须有大小为 560 Ω 的串联电阻（放置在接近芯片的位置），以便提高抗噪能力。
- GPO0-GPO3: 连接到外部 LED
 - 在灌电流模式中连接 LED，因为 CY8CMBR3xxx 控制器具有高灌电流能力。
 - 对各个串联电阻进行连接，以使 GPO 电流受 I_{IL} 的限制。
- CMOD 引脚: 通过大小为 2.2 nF 的电容连接至地
- VCC 引脚: 通过大小为 0.1 μ F 的电容连接至地。
 - 供电电压为 3.3 V，所以，VCC 引脚和 VDD 引脚并没有短接。
- VDD 引脚: 连接至外部供电电压
 - 1 μ F 和 0.1 μ F 去耦电容连接至 VDD 引脚。
- VDDIO 引脚: 连接至供电电压 ($\leq$ VDD)
 - VDDIO 给 I²C 和 HI 线供电。
- I²C_SCL 和 I²C_SDA 引脚: 通过大小为 330 Ω 的电阻连接至 I²C 插座

² VDD: 这是芯片的主要电源，其范围值为 1.8 V $\pm$ 5% 或 1.8 至 5.5 V。

³ VCC: 指的是内部电压调节器输出，用于给内核和电容式感应电路供电。

⁴ VDDIO: 这是 I²C SDA、I²C SCL、和 CY8CMBR3108 中 HI 线的电源输入。这些 I/O 的信号电平都引用 VDDIO 电源。更多有关电源的信息，请参考 CY8CMBR3xxx 数据手册。

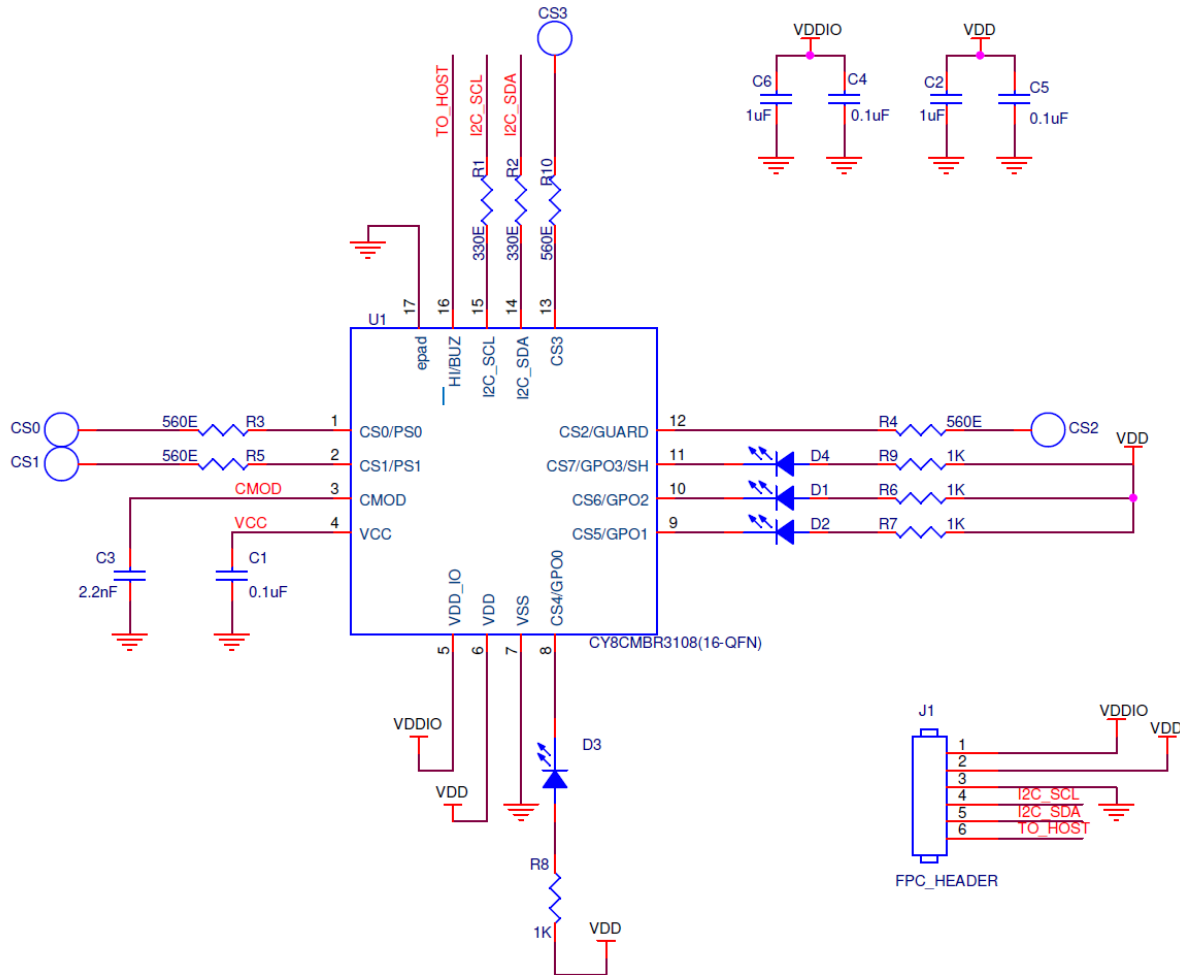
⁵ SDA — 串行数据（SDA）线用于在 I²C 主器件和 I²C 从器件间发送数据。

⁶ SCL — 串行时钟（SCL）线用于将 I²C 从器件与 I²C 主器件进行同步。

⁷ 有关未使用时的引脚连接的信息，请参考 CY8CMBR3xxx 数据手册中“引脚分布”一节的内容。

- HI引脚：连接至主机
 - 提示主机启动一个I²C数据传输，以读取已被更改的传感器状态。

图 3-6. 手机触摸按键的原理图



注意：

- 请将电容 C1、C2、C3、C4、C5、C6 与 560 Ω 串联电阻尽可能放在 CapSense 芯片的近处
- I²C线（SDA和SDL）和HI线要求主机端上的外部上拉电阻。

表 3-3. 材料清单

序号	数量	参考	说明	制造商	制造商芯片型号
1	2	C1、C4、C5	CAP CER 0.1 μF 10 V 10% X5R 0402	Taiyo Yuden	LMK105BJ104KV-F
2	1	C2、C6	CAP CER 1 μF 10 V 10% X5R 0402	TDK Corporation	C1005X5R1A105K050BB
3	1	C3	CAP CER 2200 pF 50 V 5% X7R 0402	Murata	GRM155R71H222JA01J
4	2	R1、R2	RES 330 Ω 1/10 W 5% 0402 SMD	Panasonic Electronic Components	ERJ-2GEJ331X
5	4	R3、R4、R5、R10	RES 560 Ω 1/16 W 5% 0402 SMD	Yageo Corporation	RC0402JR-07560RL

序号	数量	参考	说明	制造商	制造商芯片型号
6	4	R6、R7、R8、R9	RES 1 kΩ 1/10 W 5% 0402 SMD	Panasonic Electronic Components	ERJ-2GEJ102X
7	1	U1	CY8CMBR3108-LQXI-16QFN	Cypress Semiconductor	CY8CMBR3108-LQXI - 16QFN
8	4	D3、D2、D1、D4	LED CHIP ANGAN HER RA 0603	Avago Technologies US Inc.	516-2277-1-ND

3.3.2 家用电器中的触摸按键

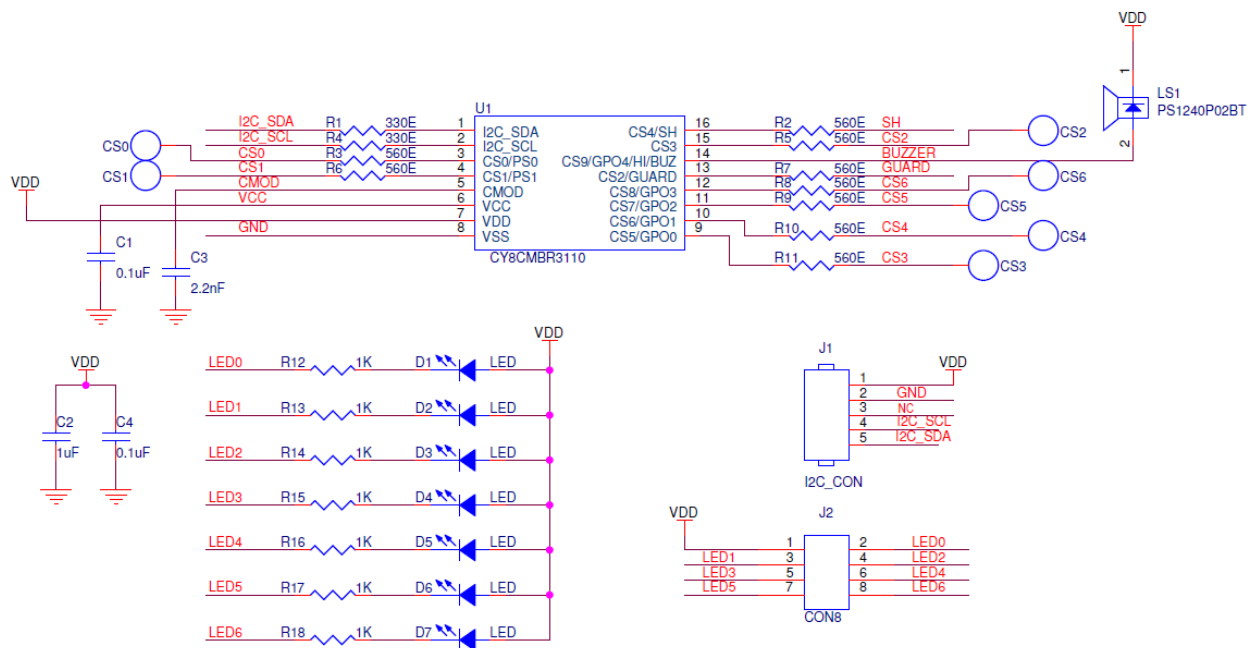
家用电器中使用了触摸按键来取代机械按键。这些应用中的关键要求是防水性 CapSense 按键和音频反馈。

图 3-7 中显示的原理图为电磁灶中实现触摸按键的示例。

在图 3-7 中，CY8CMBR3110 控制器的配置如下：

- CS0、CS1、CS3、CS5、CS6、CS7、CS8: CapSense 按键
 - 每个 CapSense 引脚必须有大小为 560 Ω 的串联电阻（放置在接近芯片的位置），以便提高抗噪能力。
- SH: 屏蔽引脚
 - 每个屏蔽引脚必须有大小为 560 Ω 的串联电阻（放置在接近芯片的位置），以便提高抗噪能力。
- GUARD: 保护传感器
 - 每个保护传感器必须有大小为 560 Ω 的串联电阻（放置在接近芯片的位置），以便提高抗噪能力。
- BUZZER: 连接到 1 引脚的蜂鸣器
- CMOD 引脚: 通过大小为 2.2 nF 的电容器连接至地
- VCC 引脚: 通过大小为 0.1 μF 的电容器连接至地（器件）
 - 供电电压为 5 V，所以，VCC 引脚和 VDD 引脚并没有短接。
- VDD 引脚: 连接至外部供电电压
 - 1 μF 和 0.1 μF 去耦电容连接至 VDD 引脚。
- I²C_SCL 和 I²C_SDA 引脚: 通过大小为 330 Ω 的电阻连接至 I²C 插座。

图 3-7. 电磁炉中触摸按键的原理图



注意:

- 请将电容 C1、C2、C3、C4 和 560 Ω 串联电阻尽可能放置接近于 CapSense 芯片
- 该原理图中所示的LED由主机控制器根据传感器状态来驱动。
- I²C线（SDA和SDL）要求主机端上的外部上拉电阻。

表 3-4. 材料清单

序号	数量	参考	说明	制造商	制造商芯片型号
1	2	R1、R4	RES 330 Ω 1/10 W 5% 0603 SMD	Yageo Corporation	RC0603JR-07330RL
2	9	R3、R6、R2、R5、R7、R8、R9、R10、R11	RES 560 Ω 1/10 W 5% 0603 SMD	Yageo Corporation	RC0603JR-07560RL
3	1	C1、C4	CAP CER 0.1 μ F 10 V 10% X5R 0603	AVX Corporation	0603ZD104KAT2A
4	1	C2	CAP CER 1 μ F 10 V 10% X5R 0603	TDK Corporation	C1608X5R1A105K080AC
5	1	C3	CAP CER 2200 pF 50 V 5% NP0 0603	Murata	GRM1885C1H222JA01D
6	1	U1	CY8CMBR3110-SX2I-16SOIC	Cypress Semiconductor	CY8CMBR3110-SX2I-16SOIC
7	1	LS1	Buzzer Piezo 4 kHz 12.2 mm PC MNT	TDK Corporation (VA)	PS1240P02BT
8	7	D1、D2、D3、D4、D5、D6、D7、D8	LED Red Clear 1206 REAR MNT SMD	Stanley Electric Co	BR1111R-TR
9	7	R12、R13、R14、R15、R16、R17、R18	RES 1.0 k Ω 1/10 W 5% 0603 SMD	Yageo	RC0603JR-071KL
10	1	J1	CONN HDR 100" SNGL PCB 5POS	Sullins Connector Solutions	SWR25X-NRTC-S05-ST-BA
11	1	J2	CONN HEADER .100" DUAL STR 80POS	Sullins Connector Solutions	PRPC040DAAN-RC

3.3.3 接近感应

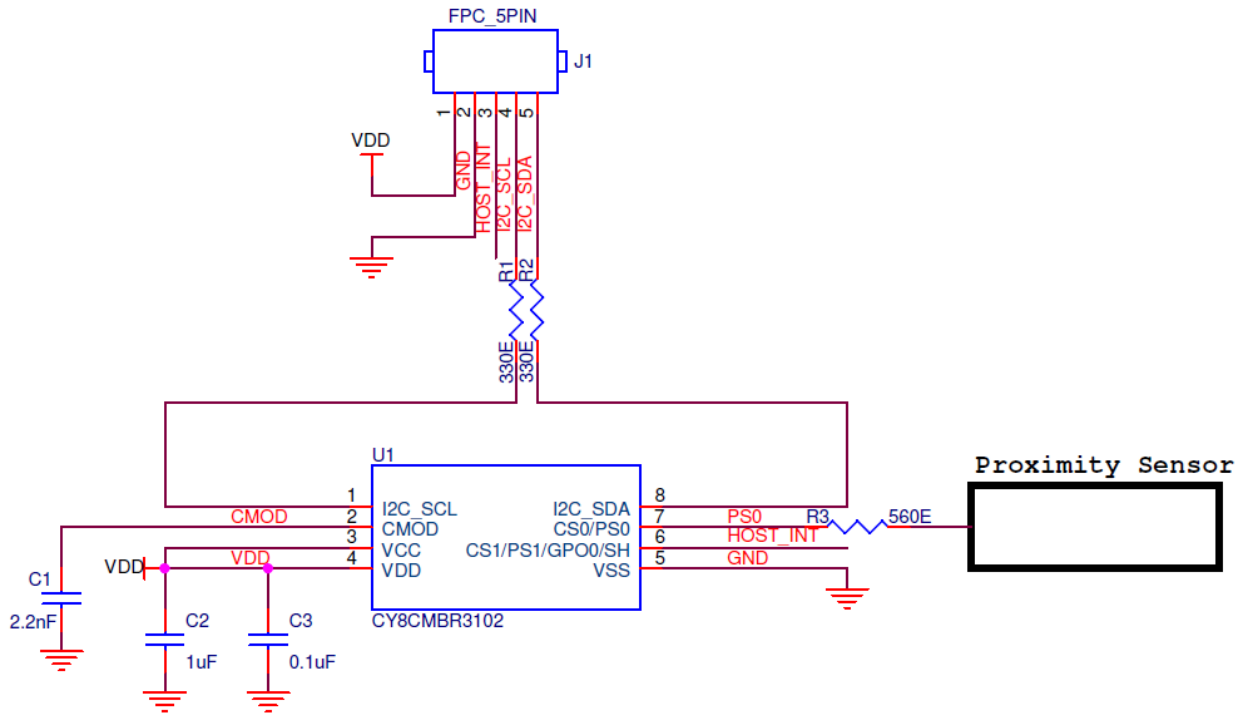
接近感应被用于手机/平板电脑，以便当用户正在通话并手机靠近耳朵时，将关闭显示屏/禁用键盘。接近感应也用于移动手机/平板电脑中的比吸收率（SAR）控制。

图 33-8 中的原理图显示的是移动手机/平板电脑中实现接近感应的示例。

在图 33-8 中，CY8CMBR3102 控制器的配置如下：

- PS0: 接近感应传感器
 - ☐ 每个接近感应传感器引脚必须有大小为 560 Ω 的串联电阻（放置在接近芯片的位置），以便提高抗噪能力。
- GPO0: 连接至主机
 - ☐ GPO 用于向主机控制器报告接近传感器状态
- CMOD 引脚: 通过大小为 2.2 nF 的电容连接至地
- VCC 引脚: 短接 VDD 引脚，以便实现 1.71 ~ 1.89 V 的操作
- VDD 引脚: 连接至外部供电电压
 - ☐ 1 μ F和0.1 μ F去耦电容连接至VDD引脚。
- I²C_SCL和I²C_SDA引脚: 通过大小为330 Ω 的电阻连接至I²C。

图 33-8. 接近感应应用的原理图



注意:

- 请将电容 C1、C2、C3 和 560 Ω 串联电阻放置尽可能接近 CapSense 芯片
- I²C 线 (SDA 和 SCL) 与 Host_INT 引脚要求主机端上的外部上拉电阻。

表 3-5. 材料清单

序号	数量	参考	说明	制造商	制造商芯片型号
1	1	C1	CAP CER 2200 pF 50 V 5% X7R 0402	Murata	GRM155R71H222JA01J
2	1	C2	CAP CER 1 μF 10 V 10% X5R 0402	TDK Corporation	C1005X5R1A105K050BB
3	1	C3	CAP CER 0.1 μF 10 V 10% X5R 0402	Taiyo Yuden	LMK105BJ104KV-F
4	1	R3	RES 560 Ω 1/16 W 5% 0402 SMD	Yageo Corporation	RC0402JR-07560RL
5	2	R1、R2	RES 330 Ω 1/10 W 5% 0402 SMD	Panasonic Electronic Components	ERJ-2GEJ331X
6	1	U1	CY8CMBR3102-SX1I-8SOIC	Cypress Semiconductor	CY8CMBR3102-SX1I-8SOIC

4. CapSense 的布局指南



在典型的 CapSense 应用中，电容式传感器由 FR4/FR2 或柔性印刷电路（FPC）上的走线构成。设计 CapSense 布局为设计阶段中的重要步骤。遵循 CapSense 布局的最佳实践会帮助您获得更高的抗噪能力、较低的寄生电容（ C_P ）以及更高的信噪比（SNR）。

设计 CapSense 布局时，需要考虑到下列因素：

- 寄生电容（ C_P ）：高传感器 C_P 值使之难以检测到传感器电容的小变化，因而降低了灵敏度。CapSense 布局的设计要满足传感器 C_P 为最小的条件。
- 走线长度：更长的走线长度会使传感器 C_P 增加并降低了传感器灵敏度。此外，长的走线，比如天线，会降低传感器的抗噪能力。
- 传感器尺寸：传感器的尺寸取决于覆盖层的厚度。更厚的覆盖层，传感器尺寸也会更大。
- 功耗：传感器的功耗取决于传感器的 C_P 。较高的传感器 C_P 将会延长传感器的扫描时间，并增大总功耗。要想降低功耗，需要降低传感器 C_P 。

请遵循布局最佳实践，以便得到低 C_P 、高 SNR 以及低功耗。CY8CMBR3xxx 设计工具箱有助于实现这些目标。

4.1 设计工具箱

CY8CMBR3xxx 设计工具箱是一个提供了通用布局指南、传感器尺寸估计、寄生电容估计、平均电流消耗以及传感器响应时间的电子表格。可以从 [EZ-Click](#) 访问该设计工具箱。可在 [CY8CMBR3xxx Design Toolbox](#) 中下载设计工具箱的最新版本。

该工具箱包括四个部分：

1. 通用布局指南
2. 布局估计
3. C_P 、功耗以及响应时间计算
4. 设计验证

4.1.1 通用布局指南

该数据表提供了有关传感器形状、最小和最大尺寸、传感器位置以及在 PCB 或 FPC 上布线等问题的建议。请按照这些指南来设计 CapSense 布局。

[表 4-1](#) 总结了用于设计 CapSense 应用的布局指南。有关这些指南的详细内容，请查阅 [CapSense 入门指南](#)。

表 4-1. 布局建议

序号	类别	详细信息	最小值	最大值	建议/备注
1	按键	按键寄生电容 (Cp)	5 pF	45 pF	降低按键 Cp，增大灵敏度并降低功耗
2		按键形状	NA	NA	实心圆形模型、带 LED 孔的圆形，或圆角的矩形
3		按键尺寸	5 mm	15 mm	请查看布局估计数据表
4		按键间距	等于按键离地间隙	NA	8 mm
5		按键离地间隙	0.5 mm	2 mm	等于最小/最大限制范围内的覆盖层厚度
6	滑条	滑条段寄生电容 (Cp)	5 pF	45 pF	降低滑条段 Cp，增大灵敏度并降低功耗
7		滑条段形状 (线性滑条)	NA	NA	Z 字形 (双箭头)
8		滑条段宽度	2 mm ⁸	8 mm	8 mm ⁹
9		滑条段高度	7 mm	15 mm	12 mm
10		滑条段之间的空隙	0.5 mm	2 mm	0.5 mm
11		网格填充和滑条之间的空隙	0.5 mm	2 mm	等于覆盖层厚度，但要位于最小/最大限制范围内
12	接近感应传感器	接近传感器寄生电容 (CP)	8 pF	45 pF	降低接近传感器 Cp，增大接近感应范围
13		接近传感器形状	NA	NA	- 在 PCB 上的圆形或矩形 (弯曲边缘) 回路 (大面积) - 实心填充的圆形或矩形 (小面积) - 建议在接近传感器周围布置接地回路
14		接近感应传感器走线宽度	1.5 mm		1.5 mm
15		接近回路与接地回路间的间隙	1 mm		1 mm
16		接地回路走线长度	1.5 mm		1.5 mm
17		接近传感器的回路直径	如推荐列表中所述	如推荐列表中所述	如果禁用 ALP 滤波器，回路直径至少要等于所需的接近距离。如果使能 ALP 滤波器，回路直径至少要等于所需的接近距离的一半。任何附近接近回路传感器的接地金属/平面将会缩小接近距离。

⁸ 滑条段的建议最小宽度取决于覆盖层厚度和所使用的覆盖层材料。有关取决于覆盖层厚度和所使用材料的最小滑条段宽度的信息，请参考 CapSense 设计入门指南中“滑条设计”部分的内容。

⁹ 推荐的滑条段宽度是根据 9 mm 的人手指平均直径设定的。有关详细信息，请参考 CapSense 设计入门指南中“滑条段形状、宽度和空隙”部分的内容。

序号	类别	详细信息	最小值	最大值	建议/备注
18	保护	保护传感器寄生电容 (CP)	5 pF	45 pF	降低保护传感器 Cp，增大灵敏度并降低功耗
19		保护传感器形状	NA	NA	弧形边缘的长方形 PCB 走线围绕着所有按键传感器
20		保护传感器厚度	2 mm		2 mm
21	地/屏蔽	地/屏蔽填充 — 顶层	如推荐列表中所述	如推荐列表中所述	网格地模式，其中走线宽度为 0.17 mm (7 mil)，网格尺寸为 1.143 mm (45 mil) (25%填充)
22		地/屏蔽层 — 底层	如推荐列表中所述	如推荐列表中所述	网格地模式，其中走线宽度为 0.17 mm (7 mil)，网格尺寸为 1.778 mm (70 mil) (17%填充) 对于 FPC 上的 CapSense 设计，底层没有被地填充。
23	布线	将各个组件放置在两层 PCB 上	NA	NA	- 顶层：传感器 - 底层：器件、其他组件以及走线
24		将各个组件放置在四层 PCB 上	NA	NA	- 顶层：传感器 - 第二层：CapSense 走线和 VDD。避免将 VDD 走线布置在传感器下方 - 第三层：网格地 - 底层：器件、其他组件以及非 CapSense 走线
25		从传感器焊盘到器件引脚的走线长度		520 mm	520 mm 是针对 (具有按键直径为 5 mm 的) FR4 PCB。对于不同的设计，请参阅布局估计数据表。走线越短，传感器 Cp 将越低。
26		走线宽度 (FR4/FR2/FPC)	0.17 mm	0.20 mm	0.17 mm (7 mil)
27		走线布局	NA	NA	应将走线布在无传感器的侧面。如将非 CapSense 走线穿过 CapSense 走线，则应确保其垂直相交。
28		传感器的过孔位置	NA	NA	过孔应接近按键边缘，以降低走线长度并增强灵敏度。
29		传感器走线的过孔尺寸			10 mil
30		传感器走线上的过孔数	1	2	1
31		CapSense 串联电阻的放置		10 mm	为实现噪声抑制，应将 CapSense 串联电阻置于靠近器件的位置。CapSense 电阻优先于 LED 电阻。先放置它们。
32		任何 CapSense 走线距接地层的距离	0.254 mm (10 mil)	0.508 mm (20 mil)	0.508 mm (20 mil)
33		器件放置	NA	NA	将器件安装在传感器的对面层。器件和传感器间的 CapSense 走线长度要为最小 (请查看上述的走线长度内容)
34		LED 背光	NA	NA	在传感器焊盘上切一个孔，并使用可背后安装的 LED
35		电路板厚度	如推荐列表中所述	如推荐列表中所述	基于 CapSense FR4 设计的标准电路板厚度为 1.6 mm

序号	类别	详细信息	最小值	最大值	建议/备注
36	覆盖层	覆盖层厚度	基于设计要求	5 mm	请使用布局估计数据表来决定使用覆盖层。已给的最大限制是针对塑料覆盖层
37		覆盖层材料	NA	NA	使用非导电材料（如玻璃、ABS 塑料、胶木、木材，等等）。PCB 与覆盖层之间不能有气隙。请使用粘合剂来粘附 PCB 和覆盖层。
38		覆盖层粘合剂	NA	NA	粘合剂应是非导电材料且介电同质。建议使用 3M 制成的 467MP 和 468MP 粘合剂。

4.1.2 布局估计

布局估计数据表根据最终系统要求和工业设计而提供推荐的最小传感器尺寸和最大走线长度，如图 4-1 所示。

该数据表的输入参数如下：

- 覆盖层厚度：输入设计中将要使用的覆盖层厚度。CY8CMBR3xxx 支持的最厚塑料覆盖层分别为 5 mm（对于按键传感器）和 2 mm（对于滑条传感器）。
- 覆盖层介电常数：输入覆盖层介电常数。有关某些材料的介电常数的详细信息，请查阅该数据表中的表 C。材料的介电常数更高，传感器的灵敏度将更高。
- 每英寸走线的电容：请参考该数据表中的表 C 并输入每英寸走线的电容。与 FPC 相比，FR4 PCB 的 C_p 更低。
- 按键灵敏度：该参数决定了按键直径和最大覆盖层厚度。选择 400 fF 值为系统的最低灵敏度，并选择 100 fF 为系统的最高灵敏度。
- 滑条灵敏度：如果您的设计有一个滑条，请输入滑条的灵敏度。覆盖层更厚，灵敏度便更低。
- 噪声条件：噪声条件是决定传感器尺寸的因素之一。如果您的设计需要在高噪声环境下工作，请选择“High”值。如果噪声条件不确定，选择“Medium”值。
- 接近传感器类型：如果您的设计需要接近传感器，请选择接近传感器类型。
- 所需的接近距离：选择所需的接近距离。有效范围为 1 到 30 cm。
- ALP 滤波器：如果您的设计要求更大的接近距离和更小的回路尺寸，便选择“Enabled”，否则，选择“Disabled”。当使能高级低通（ALP）滤波器时，接近传感器的响应时间将会增大。

基于所输入的参数，工具箱将估计下面内容：

- 用以获得 5:1 信噪比（SNR）的最小按键直径：如果信噪比（SNR）为 5:1 时所需的最小按键直径大于 15 mm，工具箱将显示一个警报信息，提示您降低覆盖层厚度或增大传感器灵敏度。对于 CY8CMBR3002 CapSense 控制器而言，灵敏度参数的固定值为 100 fF。为了避免引起传感器不正常，要使按键直径等于所建议的最小值。如果未检测到触摸，请增大按键直径，直到触摸响应最佳为止。如果按键直径过大，会导致悬空现象，并在手指触摸传感器前会打开它。
- 用以获得 5:1 信噪比（SNR）的最小滑条尺寸：当所需滑条尺寸超出范围时，工具箱将显示警报信息。在此情况下，需要降低覆盖层厚度或降低传感器灵敏度。
- 按键和滑条传感器的最大允许走线长度。
- 滑条段间的间隙和按键离地间隙。
- 在 SNR 为 5:1 的情况下，用以获得所需接近距离的最小接近回路直径。

图 4-1. 布局估计

Layout Estimator

Table A: Layout Estimator

Input Parameters	Value	Units	Comments
Overlay Thickness	2	mm	
Overlay - Dielectric constant	2.8	farad/m	
Capacitance of trace per inch	2	pF	
Button Sensitivity	200	fF	
Slider Sensitivity	100	fF	
Noise Condition	Medium		
Minimum Recommended Button Diameter	8	mm	
Minimum Recommended Slider Width	2	mm	
Minimum Recommended Slider Height	14	mm	
Maximum Trace Length for Button	508	mm	
Maximum Trace Length for Slider Segments	520	mm	
Clearance Between Slider Segments	2	mm	
Button to Ground clearance	2	mm	

Table B: Proximity Sensor Layout Estimator

Input Parameters	Value	Units	Comments
Proximity Sensor Type	PCB TRACE		
Required Proximity Distance	5	cm	
ALP Filter	Disabled		
Minimum Recommended Proximity Sensor Loop Diameter	5	cm	

Table C: Industry Standard Reference Values

Overlay Material	Dielectric constant
Plastic	2.8
Plexi glass	8
Formica	4.6 - 4.9
Glass (Standard)	7.6 - 8.0
Glass (Ceramic)	6
Mylar	3.2
ABS Plastic	3.8 - 4.5
Wood	1.2 - 2.5
Trace and board type	Capacitance per inch in pF
Copper trace, PCB, 2 layer, 64m	2
Copper trace, flex PCB, 2 layer	8

Note: Search online for the dielectric constant of the materials which are not listed in Table C.

	input cells, edit with actuals
	output cells, based on inputs

4.1.3 C_P、功耗以及响应时间计算

电路板布局完成后，通过使用 C_P、功耗以及响应时间计算数据表，可以估计设计的 C_P、平均电流消耗以及响应时间。

4.1.3.1 估计 C_P 值

请按照下面步骤来估计传感器 C_P：

1. 从表 A 的下拉列表中，选择器件型号。
2. 如果选中了“CY8C3106S”器件型号，并设计中需要滑条，请输入滑条数量和每个滑条组中的段数量。
3. 从表 B 的“传感器类型”列中，选择所需的传感器类型。未连接到任何传感器或不适用于所选器件型号的引脚，请选择 NC（无连接）。有关引脚的详情，请参考 [CY8CMBR3xxx 数据手册](#)。
4. 输入所选传感器的尺寸。“滑条长度”列只应用于滑条段。如果输入的传感器尺寸超过最大范围，工具箱将提示您减少传感器尺寸。要查看最小和最大传感器尺寸，请参考[通用布局指南](#)内容。
5. 在“走线长度”列中，输入每个传感器的走线长度。
6. 输入完这些参数后，工具箱将估计传感器 C_P 值，并将其显示在“传感器寄生电容（C_P）”列中。如果估计的 C_P 值大于 45 pF，要减少走线长度，直到 C_P 值小于 45 pF 为止。如果该值满足最大限制条件，建议不要减少传感器尺寸大小。

图 4-2.估计传感器 C_P 值

Table B: Cp Calculator

Sensor	Sensor Dimension				Trace Length		Sensitivity		Parasitic capacitance (Cp) of sensors (Approx)		Comments
	Slider Length		Button Diameter/Slider Width/Proximity Loop								
PS0			10	mm	50	mm	100	fF	7	pF	
PS1			10	mm	55	mm	100	fF	7	pF	
CS2			10	mm	58	mm	100	fF	10	pF	
CS3			10	mm	56	mm	100	fF	10	pF	
CS4			10	mm	54	mm	100	fF	10	pF	
CS5			10	mm	57	mm	100	fF	10	pF	
SLD10	12	mm	2	mm	62	mm	100	fF	9	pF	
SLD11	12	mm	2	mm	64	mm	100	fF	9	pF	
SLD12	12	mm	2	mm	67	mm	100	fF	10	pF	
SLD13	12	mm	2	mm	69	mm	100	fF	10	pF	
SLD14	12	mm	2	mm	71	mm	100	fF	10	pF	
SLD20	10	mm	2	mm	45	mm	100	fF	7	pF	
SLD21	10	mm	2	mm	47	mm	100	fF	7	pF	
SLD22	10	mm	2	mm	49	mm	100	fF	8	pF	
SLD23	10	mm	2	mm	52	mm	100	fF	8	pF	
SLD24	10	mm	2	mm	54	mm	100	fF	8	pF	

4.1.3.2 估计平均电流消耗

输入传感器尺寸（如估计 C_P 值部分所述）后，请按照下面步骤来估计控制器的平均电流消耗：

1. 在表 B 的“灵敏度”列中，选择每个传感器的灵敏度。灵敏度值越高，电流消耗将越低。
2. 在表 C 中，选择扫描周期。扫描周期值越大，平均电流消耗将越低。
3. 选择状态超时时间。超时时间越低，功耗也越低。根据您的设计的要求来选择该值。
4. 如果使能了 EMC，请选择“Enabled”；否则，选择“Disabled”。
5. 如果使能了蜂鸣器，选择蜂鸣器为 ON 时间值；否则，选择 0。当使能蜂鸣器时，蜂鸣器为 ON 时间值的增大将会使平均电流消耗也增加。
6. 如果使能了 GPO 上的 PWM，请选择“Enabled”；否则，选择“Disabled”。
7. 输入平均一个小时中的触摸数，以及每个传感器触摸的时间。该值取决于用户触摸传感器的频率。
8. 输入完这些参数后，工具箱将显示估计平均每小时的电流消耗，如图 4-3 所示。

图 4-3. 平均电流消耗估计

Table C: Power Calculator

Scan Period	200	ms
Time Out Period	8	s
EMC	Enabled	
Buzzer ON Time	0	ms
PWM on GPOs	Disabled	
Wake On Proximity	Disabled	
Average Number of Button Touch per Hour	10	
Average Button Touch Time	100	ms
Estimated Average Current Consumption	1.08	mA

4.1.3.3 估计响应时间

传感器的响应时间取决于表 B 和表 C 中所输入值。要想估计按键和滑条的响应时间，请按照 4.1.3.1 和 4.1.3.2 节中所述的内容在表 B 和表 C 中分别输入参数。

请按照下面步骤来估计按键和滑条的响应时间：

1. 在表 D 中，选择去抖动参数值。
2. 如果使能了无限脉冲响应（IIR）和中值滤波器，在各自单元格中选择“Enabled”；否则，选择“Disabled”。
3. 当输入这些参数时，工具箱将为按键和滑条分别显示了第一触摸和后续触摸的响应时间，如图 4-4 所示。

要估计接近传感器的响应时间和器件时序参数，请按照下面步骤进行操作：

1. 要想估计不同工作模式下的接近传感器响应时间，请选择控制器的工作模式。有关每个工作模式的详情，请参考 CY8CMBR3xxx 的工作模式。
2. 如果使能了高级低通（ALP）滤波器，请选择“Enabled”；否则，选择“Disabled”。
3. 选择去抖动参数值。
4. 输入这些参数后，工具箱将显示接近检测和接近释放的响应时间，如图 4-4 所示。
5. 要估计 LED 持续时间的最大值和最小值，请选择 LED 持续时间。

图 4-4.响应时间估计

Table D: Response Time Calculator

Button and Slider Touch and Release Response Time		
Debounce	3	
IIR Filter	Enabled	
Median Filter	Enabled	
Response time for first button touch	754	ms
Response time for consecutive button touch	268	ms
Response time for first slider touch	665	ms
Response time for consecutive slider touch	178	ms
Response time for button/slider release	178	ms

Proximity Sensor Response Time				
Operating State	Look-for-Touch			
ALP Filter	Disabled			
Debounce	3			
	Minimum	Typical	Maximum	
Response time for proximity detection	354	354	354	ms
Response time for proximity release	400	400	400	ms

Device Timing Parameters		
LED Duration	500	ms
LED Duration (Minimum)	500	ms
LED Duration (Maximum)	594	ms
GPO Control Status Delay	200	ms
Device Wakeup Time	50	ms

4.1.4 验证设计

构建和测试原型板后，使用设计验证数据表来验证设计。要想使用设计验证数据表，必须在 [CP、功耗以及响应时间计算](#) 数据表的表 B 中输入参数。

“设计验证”数据表只会验证按键和滑条传感器的设计；它并不会验证或提供接近传感器设计的建议，这是因为接近传感器的设计取决于各种外部因素。

请按照下面步骤来验证您的设计：

1. 使用 [EZ-Click](#) 来测量各个按键和滑条的原始信号、噪声以及 C_P 值。
2. 在表 C 中，输入每个传感器的原始信号、噪声以及 C_P 值，如图 4-5 所示。
3. 工具箱基于这些参数而提供了推荐的传感器尺寸。如果在 [CP、功耗以及响应时间计算](#) 的表 B 中所输入的传感器尺寸大于或等于推荐的传感器尺寸，该设计的状态被显示为“通过”；否则，其状态为“失败”。

如果设计通过，原型阶段完成，此时设计可以就绪生产。

注意：如果设计中有接近传感器，工具箱将不会提供任何有关接近传感器的建议，并不会验证整个设计。

如果设计失败，请按照下面步骤进行操作：

1. 降低表 B 中的灵敏度参数值。
2. 如果设计仍失败，请降低表 A 中的覆盖层厚度，如图 4-6 所示。

- 降低灵敏度和覆盖层厚度后，如果设计仍失败，请根据表 C 中的推荐值在表 B 中输入新的传感器尺寸，如图 4-7 所示。

注意：必须根据表 C 中的推荐值重新设计该布局。

图 4-5. 设计验证 — 传感器尺寸建议表

Table C: Sensor Dimension Recommendation

Sensor	Values taken from I2C						Improvement Recommendations					
	Noise		Rawcount		Cp		Recommended Sensor Dimension			Comments	Maximum Trace Length	
PS0	20	counts	45000	counts	12	pF						
PS1	25	counts	45400	counts	13	pF						
CS2	10	counts	3000	counts	12	pF			6	mm	520	mm
CS3	10	counts	3000	counts	12	pF			6	mm	520	mm
CS4	10	counts	3000	counts	13	pF			6	mm	520	mm
CS5	10	counts	3000	counts	15	pF			7	mm	514	mm
SLD10	12	counts	3100	counts	9	pF	14	mm	2	mm	520	mm
SLD11	13	counts	3125	counts	10	pF	14	mm	2	mm	520	mm
SLD12	15	counts	3150	counts	10	pF	14	mm	2	mm	520	mm
SLD13	14	counts	3140	counts	9	pF	14	mm	2	mm	520	mm
SLD14	12	counts	3170	counts	10	pF	14	mm	2	mm	520	mm
SLD20	16	counts	3158	counts	8	pF	14	mm	2	mm	520	mm
SLD21	12	counts	3100	counts	7	pF	14	mm	2	mm	520	mm
SLD22	14	counts	3140	counts	8	pF	14	mm	2	mm	520	mm
SLD23	13	counts	3125	counts	9	pF	14	mm	2	mm	520	mm
SLD24	16	counts	3158	counts	7	pF	14	mm	2	mm	520	mm

图 4-6. 设计验证 — 实际设计值表

Table A: Actual Design Values

Input Parameters	Initial Value	New Value	Units
Overlay Thickness	2	1	mm
Overlay - Dielectric constant	2.8		farad/m
Capacitance of trace per inch	2		pF
Scan Period	200	200	ms
Time Out Period	8	8	s
EMC	Enabled	Enabled	
Buzzer ON Time	0	0	ms
Wake On Proximity	Disabled	Disabled	
Average Number of Button Touch per	10	10	
Average Button Touch Time	100	100	ms

新的覆盖层
厚度

图 4-7. 设计验证 — 实际传感器尺寸表

Table B: Actual Sensor Dimensions

Sensor	Sensor Dimension								Sensitivity			
	Slider Length		Button Diameter/Slider Width/Proximity Loop Diameter		Slider Length		Button Diameter/Slider Width/Proximity Loop Diameter					
									Initial Value			
CS0			10	mm				mm	100	fF		fF
CS1			10	mm								fF
CS2			10	mm								fF
CS3			10	mm								fF
CS4			10	mm								fF
CS5			10	mm								fF
SLD10	12	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD11	12	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD12	12	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD13	12	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD14	12	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD20	10	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD21	10	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD22	10	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD23	10	mm	2	mm	14	mm	2	mm	100	fF		fF
SLD24	10	mm	2	mm	14	mm	2	mm	100	fF		fF

新的传感器尺寸值

4.2 接近传感器的布局指南

电容式接近传感器可以感应接近而不需要接触的导电物体。接近传感器和按键传感器之间有两个不同点：传感器结构和传感器灵敏度。对于接近传感器结构，电场线路应该被预计为最大距离，以便获得更大的接近距离。接近距离取决于传感器的电场传播（电场强度）。传播距离更大，检测的范围将会更大。

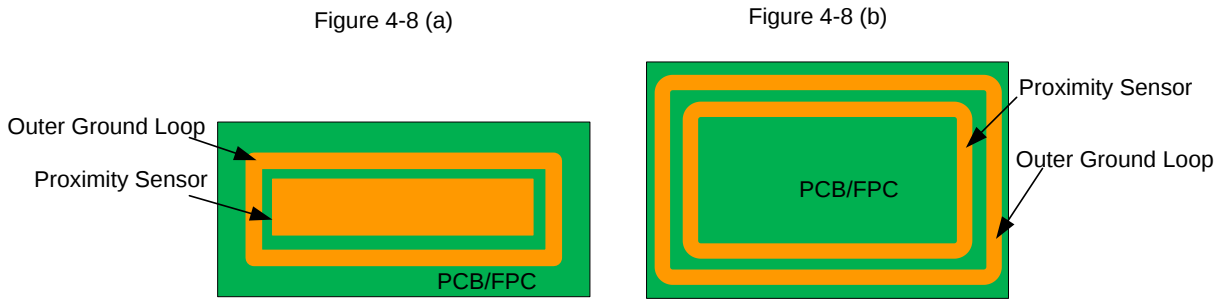
4.2.1 传感器设计指南

接近传感器结构取决于应用和所需的接近范围。手机中的人脸接近检测和平板电脑中的比吸收率（SAR）控制等应用要求最大接近距离为3厘米。为了得到该接近距离，需要有小面积且实心填充形状传感器。对于要求更大接近距离的应用，如相框、监视器和键盘，使用覆盖器件周围的大型传感器回路，以便得到更大的接近距离。

可以使用下面方式构造接近传感器：

- **PCB或FPC上的铜制走线：**长型PCB走线可以作为接近传感器使用。走线可能是直线型，或环绕系统用户接口周边，如图4-8所示。使用PCB走线构建接近传感器的这种方法适用于批量生产，但灵敏度不如导线传感器。建议在接近传感器周围使用接地回路，用以减少噪声并提供对ESD事件的抗扰性。需要权衡接近范围、抗噪性以及ESD性能。有关接近传感器和接地回路布局的建议，请查看设计工具箱中的[通用布局指南](#)。

图 4-8.PCB/FPC 上的接近传感器构建

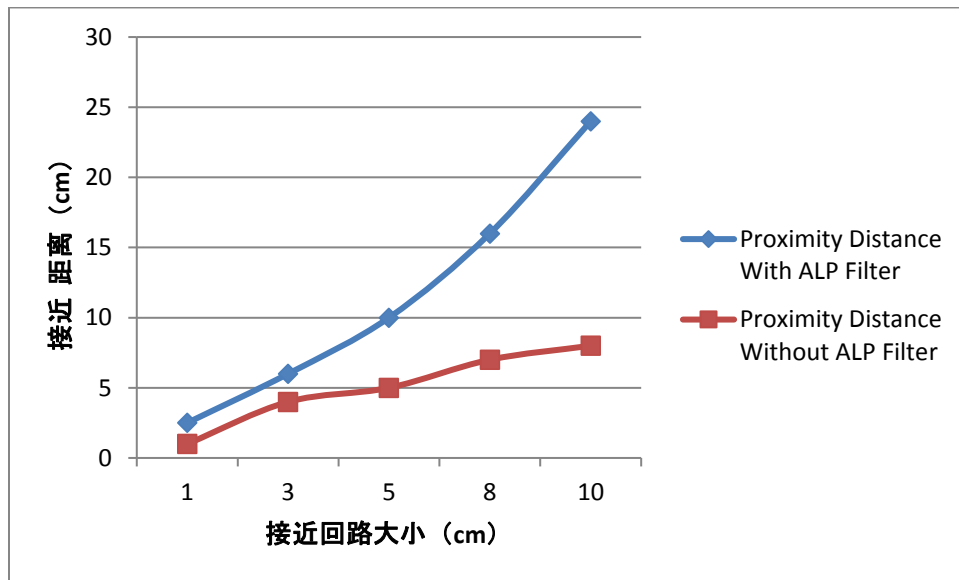


- 导线：一根导线可以用作良好的接近传感器。由于检测手指依赖于随电场变化而变化的电容，任何影响到导线周围的电场的杂散电容或物体都会影响接近传感器的感应范围。由于制造成本和复杂性较高，所以使用导线传感器对于批量生产而言并不是理想的解决方案。

请按照下面步骤来增大接近范围：

- 增大接近传感器回路大小：增加接近传感器的面积将会使接近距离增大，如图4-9所示。
- 将接近分辨率设置为最大值：增大接近距离将会使接近传感器分辨率增大。
- 使能高级低通（ALP）滤波器：使用ALP滤波器可以减少接近传感器中原始信号的噪声并增大接近距离，如图4-9所示。
- 使接近传感器屏蔽于任何接近的金属物体：放置在接近传感器附近的金属物体将会降低灵敏度并缩小接近距离。可以使用屏蔽电极来消除金属物体的影响并增大接近距离。

图 4-9.接近回路大小与接近距离的关系



注意：

- 图 4-9 中所显示的接近距离是在理想条件下测量到的。实际应用中的距离可因工作环境而被缩小。
- 图 4-9 中显示的接近距离是针对周围布置着接地回路的接近传感器进行测量的。接近回路与接地回路间的间隙为 1 mm。

4.2.2 金属物体对接近距离产生的影响

如果传感器附近有金属表面存在，接近检测范围明显被缩小了。大多数家电产品和汽车级应用有金属框架或外壳，所以要这些应用得到更大的接近范围具有挑战性。

当导电物体放置在传感器附近时，由于下面因素而明显缩小了接近检测范围：

- 传感器的寄生电容增加。更大的杂散电容通常要求降低传感器工作频率，因而缩小了接近距离。
- 金属接地层将捕捉一部分传感器电场，并降低手掌所产生的电容。

在接近传感器和金属物体之间放置一个屏蔽电极（如图4-12所示），可以减少金属表面的接近传感器的影响。

图 4-10. 单个传感器配置的电场传播（无金属物体）

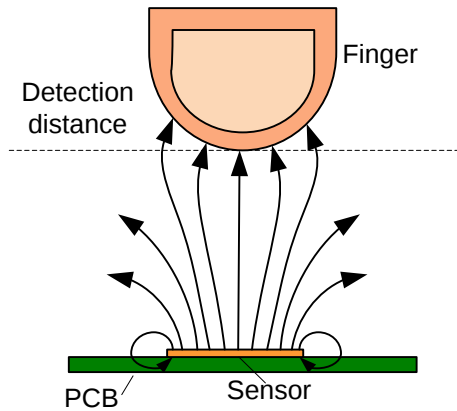


图 4-11. 单个传感器配置的电场传播（有固态金属物体）

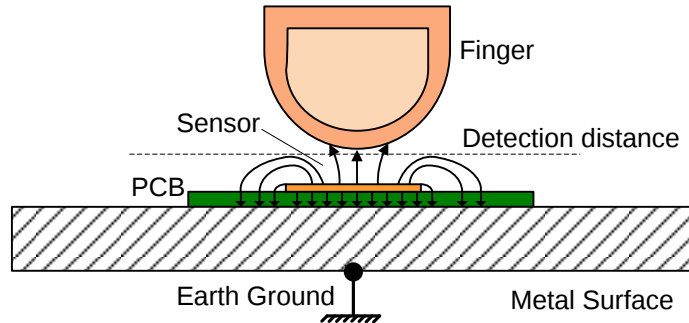
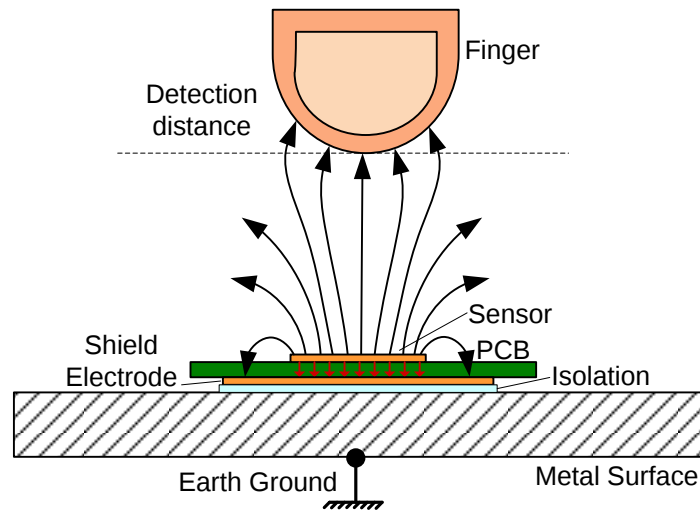


图 4-12. 使用屏蔽电极减少金属物体的影响



4.3 防水性的布局指南

覆盖层上的水滴或水流可能会错误触发电容式触摸传感器。使用屏蔽电极来消除传感器上的水滴导致的误触摸。同样，使用保护传感器来消除水流导致的误触摸。有关屏蔽和保护传感器的工作的详细信息，请查看[防水功能](#)一节的内容。

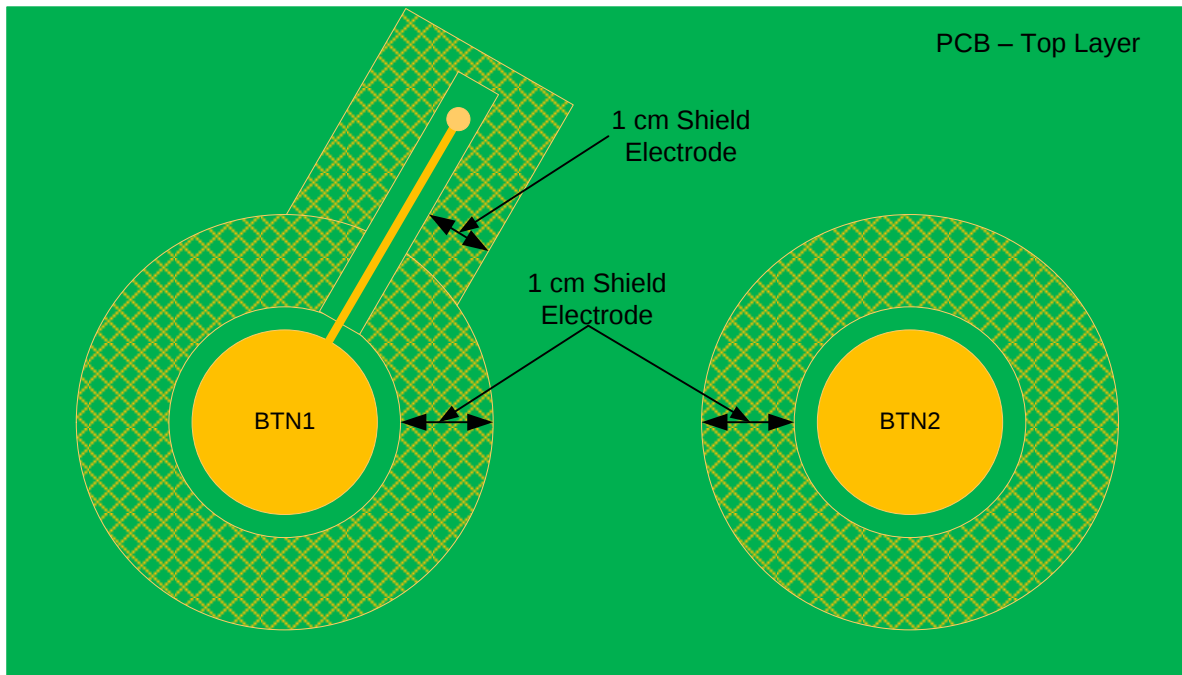
4.3.1 屏蔽电极结构

请按照下面指导来设计防水电容式感测应用：

- 在 PCB 顶层上，使用网格地（走线长度为 0.17 mm（7 mil），网格尺寸为 1.143 mm（45 mil），并使用屏蔽信号（SH 引脚）控制该网格模型。
- 将屏蔽电极放置在传感器焊盘和裸露走线（对于单层 PCB）的周围。

保持从传感器的屏蔽电极网格宽度不超过 1 cm。如果该宽度超过 1 cm，将对系统性能产生微弱影响并可能增大辐射发射。如果电路板尺寸较宽，那么屏蔽电极范围 1 cm 外的区域应保留空白，如图 4-13 所示。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。

图 4-13. 屏蔽电极模型



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）。填充的网格应该与驱动屏蔽信号相连。

四层 PCB：

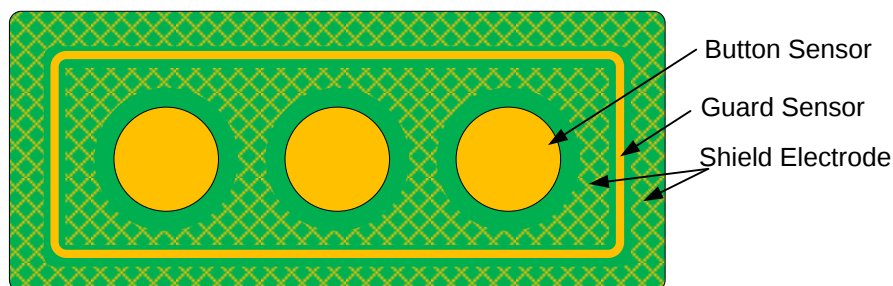
- 顶层：网格填充，走线宽度为 7 mil，网格宽度为 45 mil（25%填充）。填充的网格应该与驱动屏蔽信号相连。
- 第二层：网格填充，走线宽度为 7 mil，网格宽度为 70 mil（17%填充）。填充的网格应该与驱动屏蔽信号相连。
- 第三层：V_{DD}层
- 底层：网格填充，走线宽度为 7 mil，网格宽度为 70 mil（17%填充）。填充的网格接地。

建议传感器和屏蔽电极间的气隙为 1 mm。

4.3.2 保护传感器

保护传感器是围绕所有传感器感的铜走线，如图 4-14 所示。请确保屏蔽电极模型围绕着保护传感器和裸露走线，并离他们的间距不超过 1 cm。保护传感器的形状应为弯曲边缘的矩形。保护传感器的推荐厚度为 2 mm，并保护走线和屏蔽网格之间的最大间距为 1 mm。

图 4-14. 具有屏蔽电极和保护传感器的 PCB 布局



4.4 示例布局

4.4.1 FPC 上的触摸按键

图 4-15 和图 4-16 显示的是在 FPC 上实现触摸按键的设计布局的顶层和底层。有关该设计的原理图，请参考图 3-6。

请按照布局最佳实践进行设计：

1. 仅在顶层上填充地使传感器 C_P 变为最小。
2. 将芯片放置在传感器附近，以便使传感器的走线长度最小。
3. VDD、LED 和 I²C 线与传感器走线隔离开。
4. 传感器中间的间隙（用于 LED 背光）要为最小值，以得到最大灵敏度。

图 4-15. FPC 上的触摸按键 — 顶层

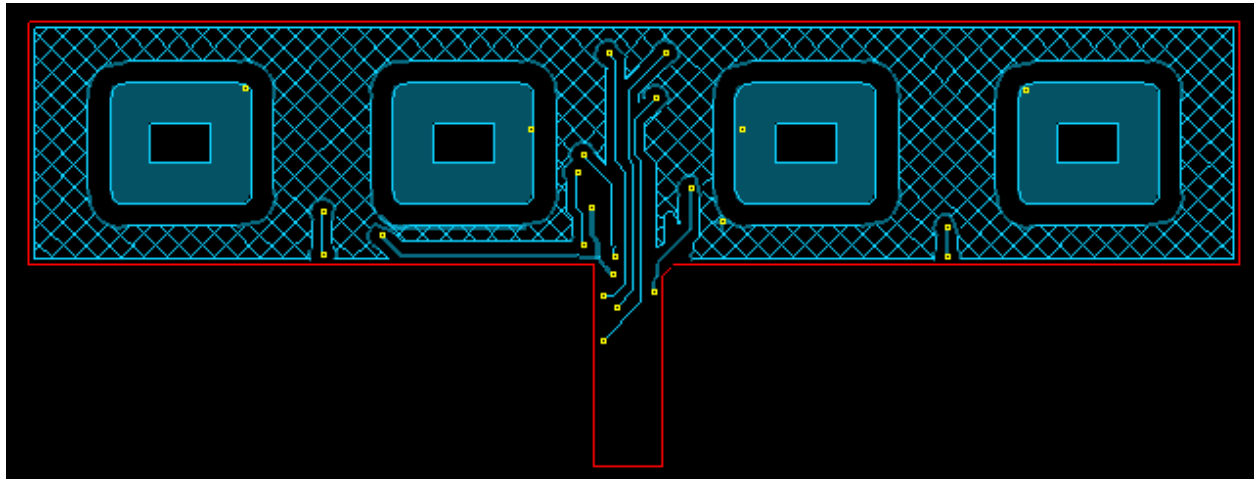
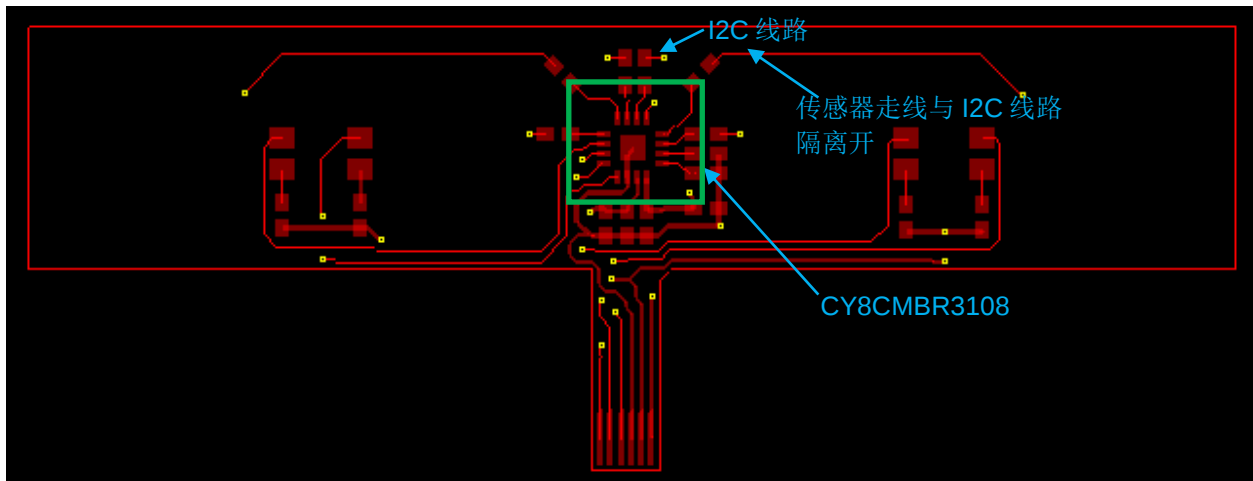


图 4-16. FPC 上的触摸按键 — 底层



4.4.2 FR4 PCB 上的触摸按键

图 4-17 和图 4-18 显示的是在 FR4 PCB 上实现防水触摸按键的设计布局的顶层和底层。有关该设计的原理图，请参考图 3-7。

请按照布局最佳实践进行设计：

1. 屏蔽和保护传感器布局的构建如[防水性的布局指南](#)一节中所述。
2. 将芯片放置在传感器附近，以便使传感器的走线长度最小。
3. VDD、LED 和 I²C 线路与传感器走线隔离开。
4. 传感器中间的间隙（用于 LED 背光）要为最小值，以得到最大灵敏度。

图 4-17. FR4 PCB 上的防水触摸按键 — 顶层

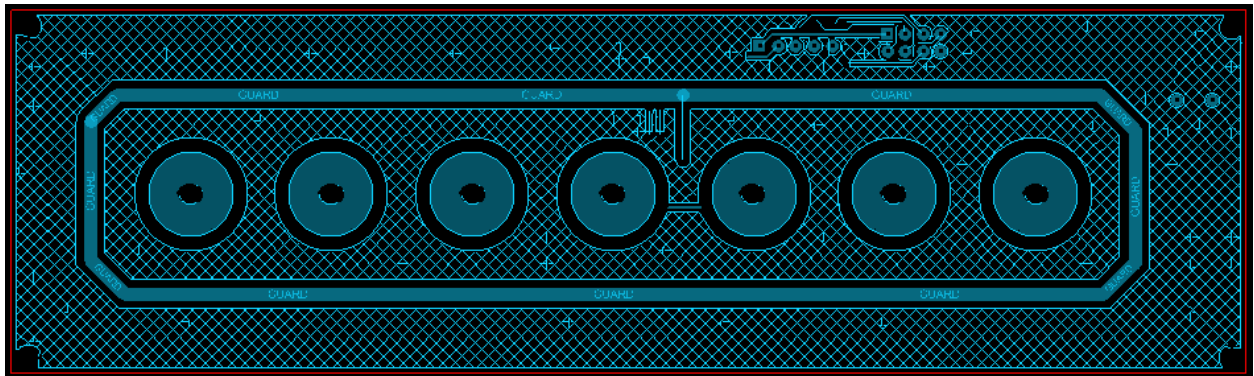
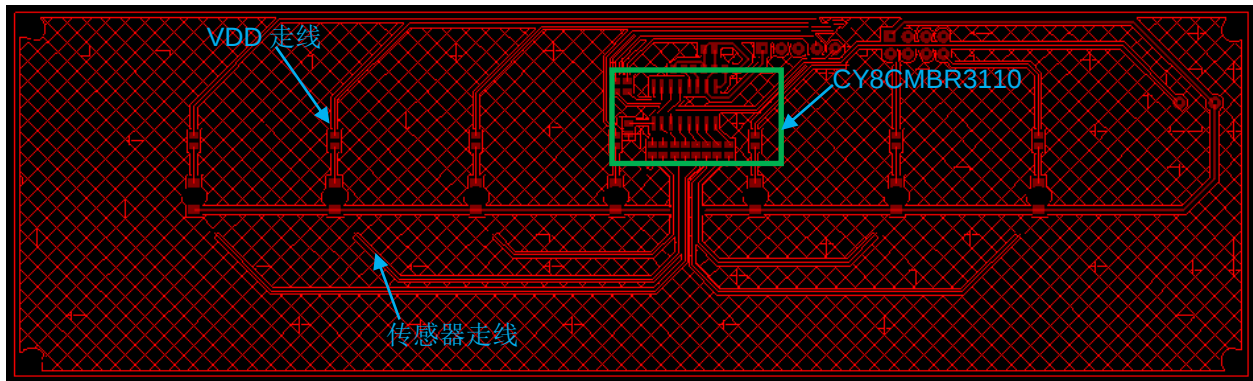


图 4-18. FR4 PCB 上的防水触摸按键 — 底层



4.4.3 FPC 上的接近传感器

图 4-19 和图 4-20 显示的是在 FPC 上实现接近传感器的设计布局的顶层和底层。有关该设计的原理图，请参考图 33-8。

请按照布局最佳实践进行设计：

1. 将接地回路围绕着接近传感器回路，以便提高抗噪性和 ESD 性能。
2. 通过使用循环传感器，可以使传感器 C_P 最小。
3. 将接地环围绕着接近传感器，使噪声为最小。这样能够提供对 ESD 事件的抗扰性。
4. 将芯片放置在传感器附近，以便使传感器的走线长度最小。
5. VDD 和 I²C 线路与传感器走线相隔离。

图 4-19. FPC 上的接近传感器 — 顶层

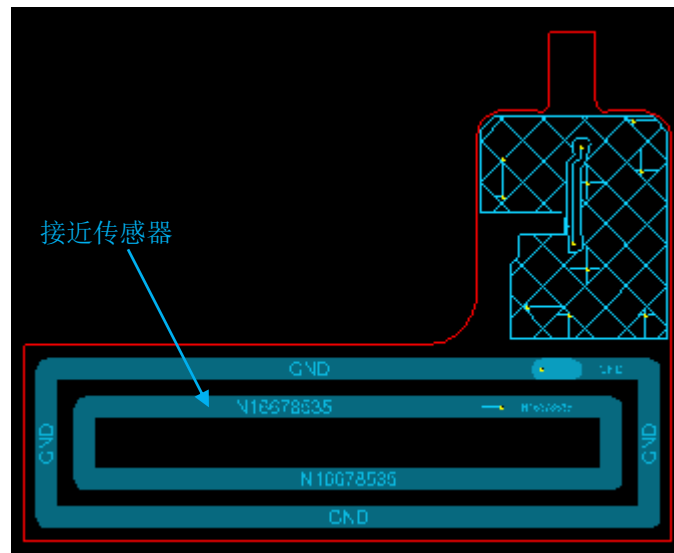
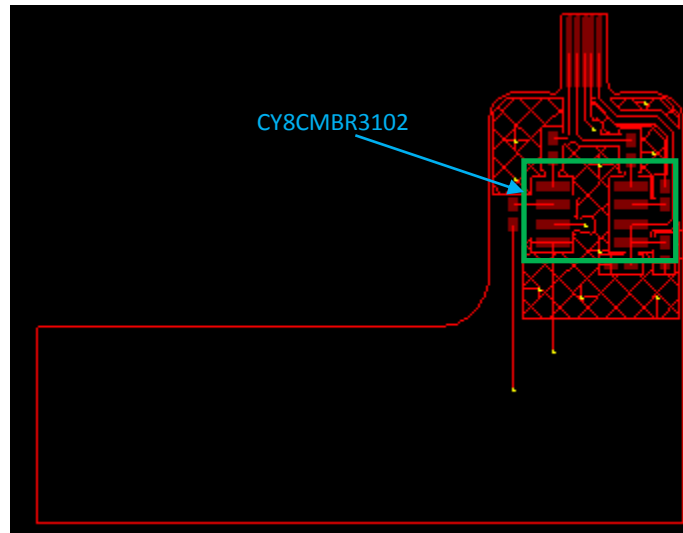


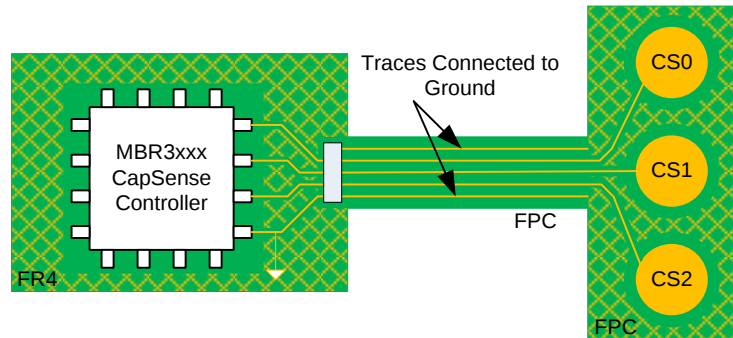
图 4-20. FPC 上的接近传感器 — 底层



4.4.4 FR4 和 FPC 上的 CapSense 设计

图 4-21 显示的是 CapSense 设计，其中在 FPC 上实现 CapSense 传感器，并将 CapSense 控制器放置在 FR4 PCB 上。通过使用 FPC 连接器将 CapSense 传感器同控制器连接在一起。

图 4-21. 在 FR4 和 FPC 上的 CapSense 设计



请按照这些指南来设计 CapSense，其中传感器位于 FPC 上，并且 CapSense 控制器位于 FR4 PCB 上：

1. 在 FPC 上，为降低传感器的寄生电容，在顶层上应该只存在填充网格。填充网格可以接地或连接到驱动的屏蔽信号。建议在顶层上只有填充网格，其走线宽度为 7 mil，并且网格宽度为 45 mil。
2. 在 FPC 连接器上，要通过传感器走线最小化耦合到 CapSense 系统的噪声，请降低传感器的走线长度，并将接地走线与传感器走线并行放置，如图 4-21 所示。
3. 在 FR4 PCB 上，您可以将 CapSense 控制器放置在顶层上，并在未被使用的区域内填充接地的网格。建议在顶层上填充网格，网格走线宽度为 7 mil，网格宽度为 45 mil。在底层上，您可以填充接地网格，其走线宽度为 7 mil，网格宽度为 70 mil。

5. 配置 CY8CMBR3xxx



CY8CMBR3xxx 为 I²C 可配置的 CapSense 控制器。该控制器提供了各种特性，并可通过 [EZ-Click](#) 工具轻松地配置；EZ-Click 是一个简单、直观的图形用户界面工具。CY8CMBR3xxx 系列的所有控制器（CY8CMBR3002 除外）都可以通过使用 EZ-Click 进行配置。CY8CMBR3002 控制器是一个固件不可修改的器件，不能对它进行配置。下面内容介绍了 CY8CMBR3xxx 系列中支持的各种配置，并说明如何配置他们。

5.1 CY8CMBR3xxx 的配置

CY8CMBR3xxx 控制器所支持的配置可分为下面四种：

1. 传感器配置
2. GPO 配置
3. 蜂鸣器配置
4. 器件全局配置

5.1.1 传感器配置

5.1.1.1 灵敏度控制

灵敏度参数可用于提高或降低传感器信号（差值计数）的强度。低的灵敏度值（100 fF）使传感器信号更强，但同时增大了响应时间和平均功耗。其他可用的灵敏度设置为 100 fF、200 fF、300 fF 和 400 fF。可以配置每个按键传感器、保护传感器和滑条组的灵敏度参数。该参数不适用于接近传感器。

传感器灵敏度受下列因素的影响：

1. 覆盖层厚度：更厚的覆盖层要求更低的灵敏度值。
2. 系统噪声：系统噪声增加时，需要降低灵敏度，以避免传感器的误触发。
3. 设计外形：相对大的传感器尺寸要求更高的灵敏度值。传感器尺寸更小，灵敏度值要更低。
4. 功耗：更低灵敏度设置使功耗增大。如果要求低功耗，需要将灵敏度参数设置为高值。

功耗会按照下面灵敏度的顺序而降低：100 fF > 200 fF > 300 fF > 400 fF。

可以通过 [EZ-Click](#) 的“CapSense sensor Configuration”选项卡来设置灵敏度参数。有关如何设置灵敏度参数的信息，请查阅 [CapSense 功能调试](#) 一节。对于 CY8CMBR3002 CapSense 控制器，该参数值固定为 100 fF，并且不可配置。

5.1.1.2 接近分辨率

接近距离取决于接近分辨率参数。接近分辨率越高，接近距离越大。需要设置每个接近传感器的分辨率参数。该参数的取值范围为 12 位到 16 位。可以通过 [EZ-Click](#) 的“CapSense sensor Configuration”选项卡来设置接近分辨率参数。有关如何设置该参数的信息，请查阅 [CapSense 功能调试](#) 一节。该参数不适用于 CY8CMBR3002 CapSense 控制器。

5.1.1.3 滑条分辨率

滑条分辨率是指控制器可处理的单一位置数量。分辨率更高的滑条可以处理更多位置。该参数仅适用于 CY8CMBR3106S 控制器。CY8CMBR3106S 支持最多两个滑条：SLD1x 组和 SLD2x 组。需要设置每个滑条组的分辨率参数。但如果两个滑条被组合成单个滑条，只需要设置一个分辨率。该参数的取值范围为 1 到 254。可以通过

[EZ-Click](#) 的“CapSense sensor Configuration”选项卡来设置滑条分辨率参数。

5.1.1.4 电磁兼容性 (EMC)

EMC 特性决定了器件对外部辐射和传导噪声的抗干扰能力，如功率放大器的音频噪声、无线发射器的音频噪声、ESD 以及电源线浪涌。对于在高噪声环境中工作的系统，需要使能 EMC 特性。使能 EMC 特性时，功耗和响应时间将会增加。

对于基于 CY8CMBR3106S 的设计，EMC 特性仅适用于传感器 CS0/PS0、CS1/PS1 以及 CS2 到 CS5。所有滑条段和从 CS11 到 CS15 的按键均不具备 EMC 特性。

对于基于 CY8CMBR3116 的设计，EMC 特性仅适用于传感器 CS0/PS0、CS1/PS1 以及 CS2 到 CS9。

在下面情况下，要禁用 EMC 特性：

- 在 CY8CMBR3116 控制器中使能了从 CS10 到 CS15 中的任何按键。
- 使能了滤波器（仅适用于基于 CY8CMBR3106S 的设计）
- 您的设计在低噪声环境下工作
- 使能了蜂鸣器

通过 EZ-Click 中的“Global Configuration”选项卡，可以使能 EMC 特性。在 CY8CMBR3002 CapSense 控制器中始终禁用了 EMC 特性。

5.1.1.5 侧向传感器抑制 (FSS)

FSS 特性用于区分靠近按键的传感器触摸，以避免误触摸。这样能够确保当同时触摸多个传感器时，系统仅识别被触摸的第一个按键。FSS 确保始终只有一个按键被报告为“ON”，如图 5-2 所示。如果手指触摸多个按键，则只有第一个感应到“触摸”的按键被报告为“ON”。如果同时按下多个按键，编号更低的传感器被报告为“ON”。

FSS 特性仅适用于按键，并可以使用 EZ-Click 中的“CapSense sensor Configuration”选项卡来使能或禁用每个按键。FSS 特性不适用于 CY8CMBR3002 CapSense 控制器。

图 5-1. 只有一个按键被按下时的 FSS



图 5-2. 有多个按键被按下时的 FSS（其中，只有第一个按键的状态为 ON）



5.1.1.6 固件滤波器

为了减少传感器原始信号的噪声，CY8CMBR3xxx 控制器支持下面滤波器：

- 中值滤波器
- IIR 滤波器
- 高级低通（ALP）滤波器

中值和 IIR 滤波器仅适用于按键、滑条和保护传感器。ALP 滤波器仅适用于接近传感器。

对于基于 CY8CMBR3106S 的设计，请注意以下事项：

- 不支持 ALP 滤波器
- 如果使能了 EMC 特性，请禁用中值和 IIR 滤波器

5.1.1.6.1 中值滤波器

中值滤波器为从大小为 N 的缓冲区计算出中值的非线性滤波器。该滤波器将去除传感器原始信号中的尖峰噪声。CY8CMBR3xxx 控制器中实现的中值滤波器为一个三阶滑动中值滤波器。有关何时使能该滤波器的信息，请查阅 [CapSense 功能调试](#) 一节。

如果使能了中值滤波器，禁用 [自动阈值](#) 性能并手动设置 [手指阈值](#) 参数。

5.1.1.6.2 无限脉冲响应（IIR）滤波器

IIR 滤波器产生了类似于 RC 滤波器的响应阶段。该滤波器减少了来自传感器原始信号的高频白噪声。有关何时使能该滤波器的信息，请查阅 [CapSense 功能调试](#) 一节。在 CY8CMBR3002 CapSense 控制器中始终使能了 IIR 滤波器。

中值滤波器和 IIR 滤波器均被使能时，在原始信号上先执行中值滤波器，然后才执行 IIR 滤波器。

5.1.1.6.3 高级低通（ALP）滤波器

ALP 滤波器用于减少接近传感器原始信号中的噪声。为了得到更大的接近距离，接近传感器要求更高的分辨率；然而，更高的分辨率值会使原始信号中的噪声增多。为了减少该高频噪声并得到大于 5:1 的信噪比（SNR），需要使能 ALP 滤波器。有关何时使能 ALP 滤波器以及使能时如何调试它的详细信息，请查阅 [调试接近传感器](#) 一节的内容。该特性不适用于 CY8CMBR3106S 和 CY8CMBR3002 CapSense 控制器。

5.1.1.7 自动阈值

自动阈值性能将根据环境的噪声，动态设置按键传感器的所有阈值参数。对于可变噪声环境，要使能该性能。该性能仅适用于按键传感器。在 CY8CMBR3002 CapSense 控制器中始终使能了该特性。

对于自动阈值性能，请注意下面内容：

- 如果使能了 EMC 性能，自动阈值性能将被禁用。此时，需要手动设置手指阈值参数。
- 如果该性能被禁用，只需要设置手指阈值参数。所有其他参数将根据手指阈值参数而被自动设置。
- 如果使能了中值滤波器，禁用自动阈值性能并手动设置手指阈值参数。

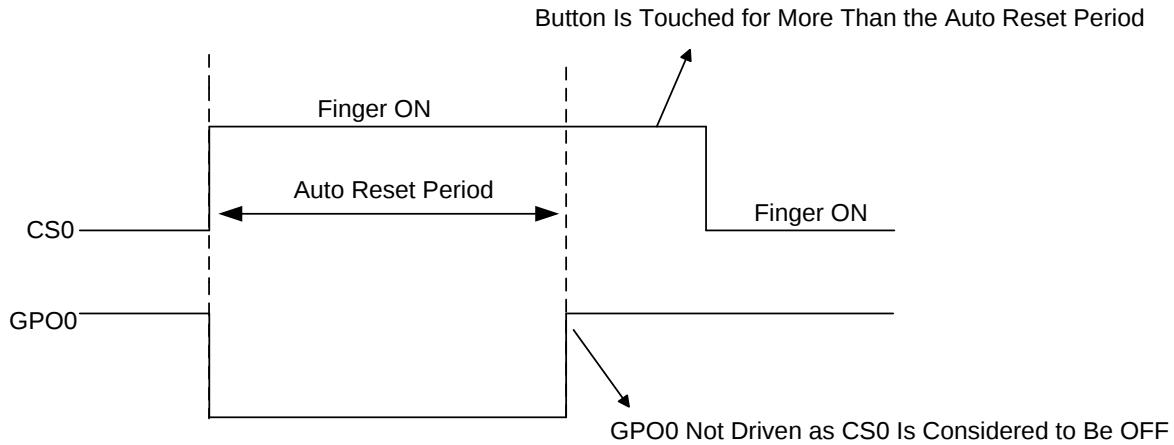
5.1.1.8 传感器自动复位时间

传感器自动复位时间决定了传感器被激活时处于 ON 状态的最大时间。自动复位周期结束后，传感器将转为 OFF 状态，如图 5-3 所示。如果有一个金属物体靠近传感器，该特性可阻止该传感器被卡住为 ON 状态。

共有两个自动复位参数集：自动复位和接近自动复位。自动复位参数可应用于按键和滑条。接近自动复位参数则应用于接近传感器。自动复位参数不应用于保护传感器。自动复位周期可以被禁止，或被设置为 5 或 20 秒。

在 CY8CMBR3002 CapSense 控制器中该参数被设置为 20 秒。

图 5-3.传感器自动复位



5.1.1.9 去抖动

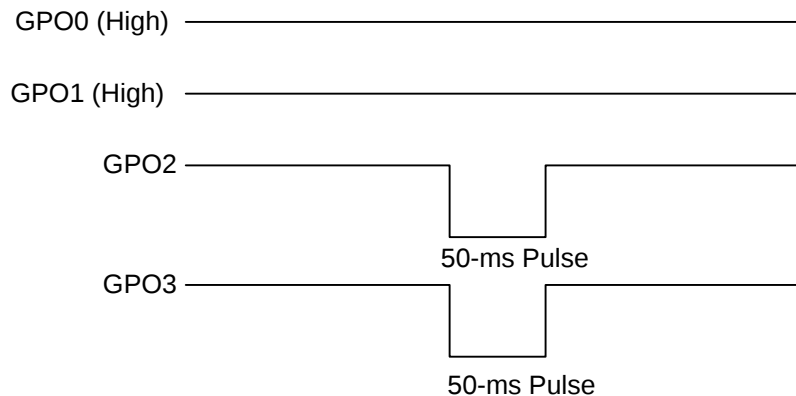
去抖动性能将指定一个传感器触摸被认为有效触摸的最小时间，从而可以避免由噪声尖峰或系统干扰导致的传感器错误触发。

根据传感器功能的不同，去抖动值也有所不同。例如，电源按键需要长去抖动时间，以避免意外切换系统的“ON”或“OFF”状态。更短的去抖动时间使器件对传感器触摸或接近事件的响应更快。去抖动参数可应用于除滑条传感器以外的所有传感器类型，并它的范围为 1 到 15。根据响应时间的要求来设置该参数。更多信息，请查阅[估计响应时间](#)一节的内容。在 CY8CMBR3002 CapSense 控制器中该参数被设置为 3。

5.1.1.10 系统诊断

系统诊断是一个内置的上电自测试（BIST）机制。如果使能该特性，在上电复位（POR）时或软件停止后将执行某些测试，有助于生产测试。如果检测到任何故障传感器，该传感器将被禁用，并在上电器件中的相应 GPO 上（若使能了 GPO）发出宽度为 50 毫秒的脉冲，如图 5-4 所示。在寄存器图中更新了系统诊断状态，以便主机可以通过 I²C 接口读取测试结果。欲了解更多信息，请参考 [CY8CMBR3xxx 寄存器技术参考手册](#)。

图 5-4.CS0、CS1 通过 BIST 并 CS2、CS3 失败的示例



在上电期间中执行下列测试：

1. 传感器短接至地：如果任何传感器短接至地，它将被禁用。
2. 传感器短接至 VDD：如果任何传感器短接至 VDD，它将被禁用。
3. 传感器间互相短接：如果两个或更多传感器互相短接，他们均被禁用。
4. 不合适的 C_{MOD} 值：C_{MOD} 的推荐值为 2.2 nF ±10%。如果 C_{MOD} 值小于 1 nF 或大于 4 nF，将禁用所有传感器，并在所有 GPO 上发出 50 毫秒宽的脉冲。
5. 按键 C_P > 45 pF：如果传感器 C_P 大于 45 pF，禁用它。CY8CMBR3xxx 估计的 C_P 容差为 ±2 pF。所以，如果传感器 C_P 处于 43 到 47 pF 的范围内，有可能将会禁用该传感器。
6. 如果屏蔽短接至 V_{DD}、地或一个传感器，将会禁用所有传感器，并在所有使能的 GPO 上发出 50 毫秒宽的脉冲。
7. 如果任何滑条段不通过上述测试，将禁用整个滑条。

如果系统诊断特性被使能，CY8CMBR3xxx 控制器的启动时间会增加。更多信息，请查阅 [CY8CMBR3xxx 数据手册](#) 中的系统规范一节。

注意： CY8CMBR3106S 不支持 GPO；所以，需要使用 I²C 接口读取寄存器图中的系统诊断结果。

5.1.1.11 屏蔽和保护传感器

在防水 CapSense 设计中要求使用屏蔽和保护传感器。如果您的设计有屏蔽电极，请选择 [EZ-Click](#) 中 “Global configuration” 选项卡下的 “Enable shield” 项。该选项允许控制器将屏蔽信号驱动到 SH 引脚。

如果您的设计有保护传感器，请选择 [EZ-Click](#) 中 “Global configuration” 选项卡下的 “Enable Water Tolerance” 项。该选项允许控制器配置引脚为保护传感器使用。请参考 [表 1-1](#)，了解器件支持屏蔽和保护传感器特性。

5.1.2 GPO 配置

CY8CMBR3xxx 控制器支持最多八个 GPO，用于驱动 LED 或驱动构成基本 D/A 转换器的外部电阻网。根据所配置的内容，GPO 可以输出 PWM 信号或输出直流电压电平。由于 GPO 引脚与 CapSense 传感器引脚被复用，所以使用 [EZ-Click](#) 来指定 CY8CMBR3xxx 引脚作为 CapSense 引脚还是 GPO 引脚。CY8CMBR3106S CapSense 控制器不支持 GPO，因此该特性不适用于该控制器。

5.1.2.1 GPO 驱动模式

CY8CMBR3xxx 中的 GPO 支持两个驱动模式：开漏低模式和强驱动模式。开漏低模式相当于一个常开（n/o）开关。当传感器无效时，相应的 GPO 将处于高阻态；当传感器被激活时，GPO 转为低态。该驱动模式有助于驱动一个外部电阻网，以实现 D/A 转换。强驱动模式可用于驱动 LED。CY8CMBR3002 CapSense 控制器的 GPO 驱动模式被固定设置为开漏低模式。

GPO 驱动模式为全局配置，并可应用于所有使能的 GPO。

5.1.2.2 GPO 逻辑电平

CY8CMBR3xxx 允许用户配置 GPO 引脚的逻辑电平（高电平有效或低电平有效）。该特性在控制基于传感器状态的 GPO 输出电压的情况下提供了灵活性，如 [表 5-1](#) 所示。该特性为全局，并可应用于所有使能的 GPO 引脚。CY8CMBR3002 CapSense 控制器的 GPO 逻辑电平是 “低电平有效”。如果将驱动模式设置为开漏（Open-Drain）低电平驱动模式，仅当引脚被外部拉到 V_{DD} 时，[表 5-1](#) 才有效。

表 5-1. GPO 逻辑电平

GPO 逻辑电平	传感器状态	GPO 上的输出电压
高电平有效	ON	V _{DD}
	OFF	GND
低电平有效	ON	GND
	OFF	V _{DD}

5.1.2.3 GPO 上的 PWM 输出

CY8CMBR3xxx 提供 GPO 引脚上的 PWM 输出。PWM 信号有利于控制 LED 的亮度。为了使能 GPO 上的 PWM 输出，在 [EZ-Click](#) 窗口中“Global configuration”选项卡内设置有效状态占空比和无效状态占空比的比例。比例为 100% 的有效占空比和比例为 0% 无效占空比导致 GPO 上的直流输出。该特性不适用于 CY8CMBR3106S 和 CY8CMBR3002 CapSense 控制器。

PWM 输出功能为全局，并可应用于所有使能的 GPO。PWM 信号的频率为 106.6 Hz。PWM 占空比可以以 6.67% 的步长对每个 GPO 进行配置。

5.1.2.4 GPO 主机控制

在 CY8CMBR3xxx 中，每个 GPO（GPO0-GPO7）被逻辑地链接到了 CapSense 引脚（CS0-CS7）。当传感器被激活时，相应的 GPO 将自动为 ON。然而，如果主机处理器需要控制 GPO 引脚，应使能 GPO 主机控制的功能。该特性允许主处理器通过写入到 GPO_OUTPUT_STATE 寄存器内来控制 GPO 引脚。欲了解更多信息，请参考 [CY8CMBR3xxx 寄存器技术参考手册](#)。该特性不适用于 CY8CMBR3106S 和 CY8CMBR3002 CapSense 控制器。

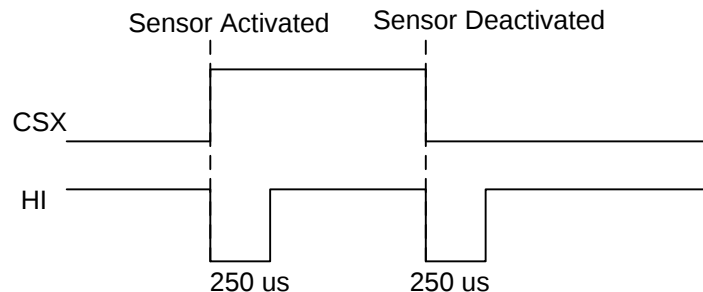
控制器在 [查找触摸（Look-for-touch）模式（Look-for-Touch）](#) 模式和 [查找接近（Look-for-Proximity）模式（Look-for-Proximity）](#) 模式下操作时，主机写指令至 CY8CMBR3xxx 控制器与这些指令响应的延迟时间等于一个 [扫描周期](#)。如果控制器在活动模式下操作，延迟时间取决于所有传感器的总扫描时间。

注意：在 EZ-Click 中，即使未使能 GPO，并且未将相应的 GPOx/CSx 引脚配置为 CapSense 传感器，但仍会根据相应传感器 ON/OFF 状态驱动 GPO（打开/关闭状态）。如果要禁用 GPO 引脚，您需要在 EZ-Click 中将引脚配置为 CapSense 输入，并将引脚接地。

5.1.2.5 主机中断

每当传感器状态或滑条位置具有改变时，主机中断功能在 [HI](#) 引脚上生成一个传输到主机处理器且时长为 250 μ s 的低电平有效脉冲信号，如 [图 5-5](#) 所示。要使能该特性，在 [EZ-Click](#) 中指定主机中断引脚。主机中断引脚具有开漏低电平驱动模式下的配置，并需要一个外部上拉电阻才能正常工作。请参见 [表 1-1](#)，了解支持该特性的器件列表。

图 5-5. CSx 按键被触摸时的主机中断线



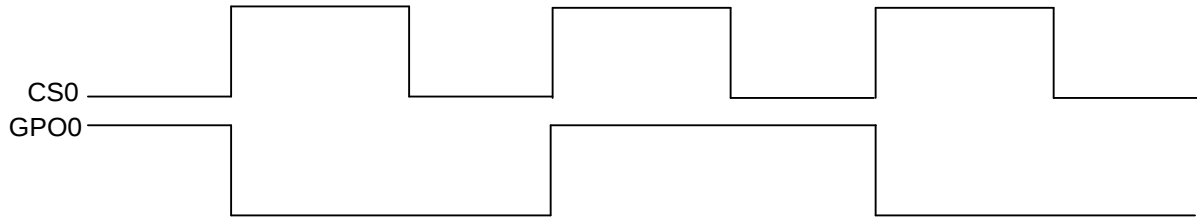
5.1.2.6 触发

当使能触发器功能时，GPO 引脚的状态改变在按键的每个上升沿和接近传感器状态上。该特性使 CapSense 控制器的机械式按键开关的行为能够实现。触发的配置显示在 [图 5-6](#) 中。对于每个 GPO 引脚，可以使能或禁用触发功能。触发功能不适用于 CY8CMBR3002 CapSense 控制器。

使用触发功能时，要考虑到以下注意事项：

- 触发和 LED 点亮保持时间功能是互斥的。如果上述两个功能均有效，则优先触发功能，并 LED 点亮保持时间功能将被禁用。
- 如果使能触发功能并打开任何 GPO，则器件不会进入 [查找接近（Look-for-Proximity）模式](#)。如果您的设计需要“接近触摸时的唤醒”功能，则禁用触发功能。

图 5-6.触发功能的示例

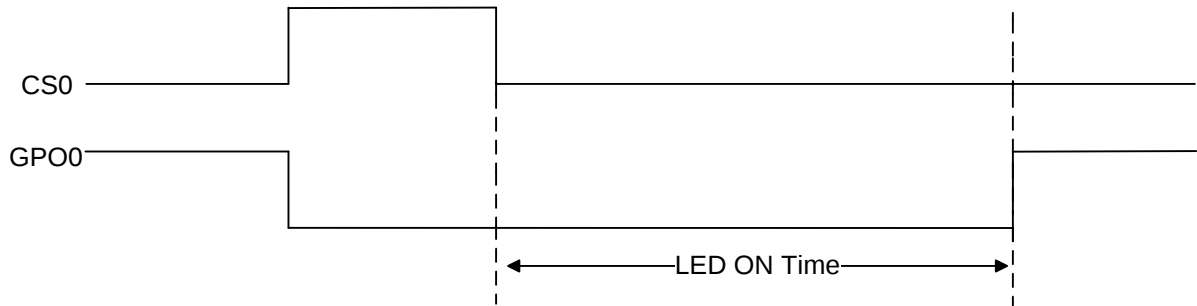


5.1.2.7 LED 点亮保持时间

LED 点亮保持时间指定释放相应传感器后 GPO 引脚被驱动为低电平或高电平时所需的时长，如图 5-7 所示。对于每个 GPO 引脚，可以使能或禁用 LED 点亮保持时间的功能。LED 点亮保持时间用于所有 GPO 并它的取值范围为 0 到 2000 ms，分辨率为 20 ms。LED 点亮保持时间功能不适用于 CY8CMBR3002 CapSense 控制器。

触发和 LED 点亮保持时间功能是互斥的。如果上述两个功能均有效，则优先触发功能，并 LED 点亮保持时间功能将被禁用。

图 5-7.LED 点亮保持时间



5.1.2.8 模拟电压输出

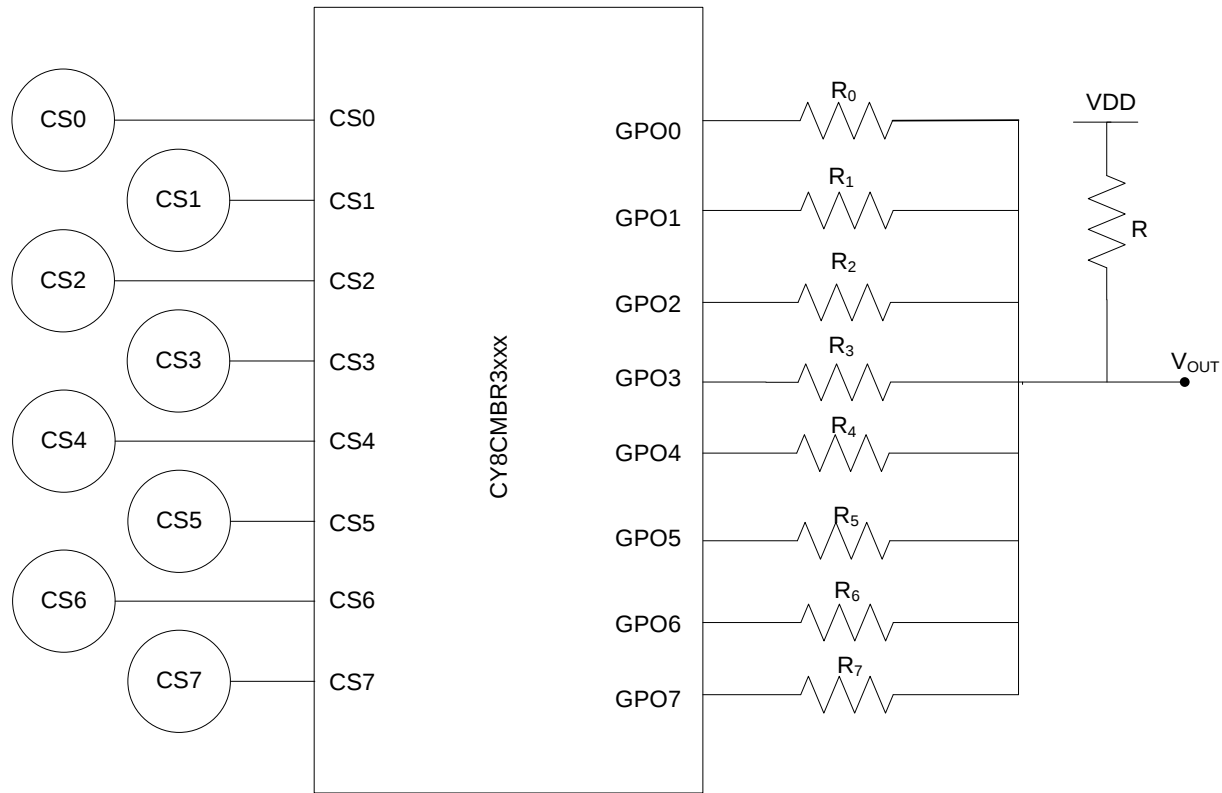
模拟电压输出功能使用一个电阻网络来指示传感器状态作为一个输入主机的模拟电压，如图 5-8 所示。在这种配置中，应将 GPO 驱动模式设置为开漏低电平驱动模式。当触摸传感器时，驱动相应的 GPO 为逻辑低电平信号，并在电压为 V_{OUT} 时形成电压分频器。释放传感器时，相应的 GPO 处于高阻状态并电压 V_{OUT} 会等于 V_{DD} 。此外，还要使能 FSS 功能，以便每次只有一个按键被报告为“ON”。

计算模拟电压 V_{OUT} 的公式如下：

$$V_{out} = \frac{V_{DD} \times R_n}{R + R_n} \quad \text{公式 3}$$

其中：R 是 V_{DD} 和 V_{OUT} 之间的电阻
 R_n 是 V_{OUT} 和接地之间的有效电阻

图 5-8.使用 GPO 和电阻网的模拟电压输出



5.1.3 蜂鸣器配置

5.1.3.1 蜂鸣器输出

CY8CMBR3xxx 支持单输入压电蜂鸣器。蜂鸣器由 PWM 信号驱动；它的频率是可配置的。要启用蜂鸣器，请选择 [EZ-Click](#) 中的蜂鸣器引脚。该特性不适用于 CY8CMBR3102 和 CY8CMBR3002 CapSense 控制器。

CY8CMBR3xxx 支持频率为 1.0、1.1、1.3、1.6、2.0、2.6 和 4.0 kHz 的蜂鸣器以满足不同压电蜂鸣器的驱动要求并提供不同的谐波。蜂鸣器输出的占空比固定为 50%。

5.1.3.2 蜂鸣器开启时间

蜂鸣器开启时间功能允许用户指定激活传感器时要使蜂鸣器输出有效的期间，如图 5-9 所示。蜂鸣器信号输出由所配置的时长驱动，并且它并不取决于传感器的触摸时间。蜂鸣器开启的时间结束后，即使激活传感器，该输出也会进入闲置状态。

蜂鸣器开启的时间结束前，如果触摸同一传感器或任何其他传感器，则蜂鸣器信号输出不会重启，如图 5-10 和图 5-11 所示。蜂鸣器开启时间的取值范围为 100 ms 到 12.7 s，它的分辨率为 100 ms。

图 5-9. 发生触摸事件时激活蜂鸣器

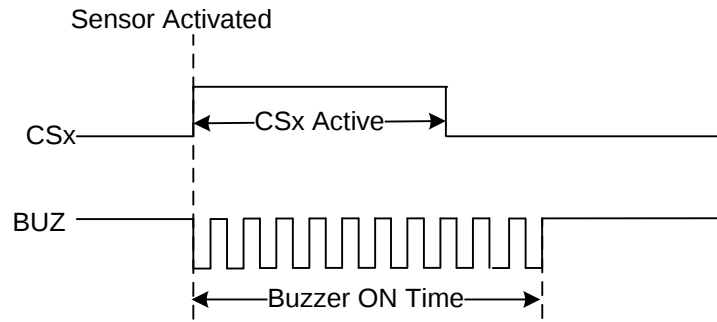


图 5-10. 在相同传感器上发生连续触摸时的蜂鸣器操作

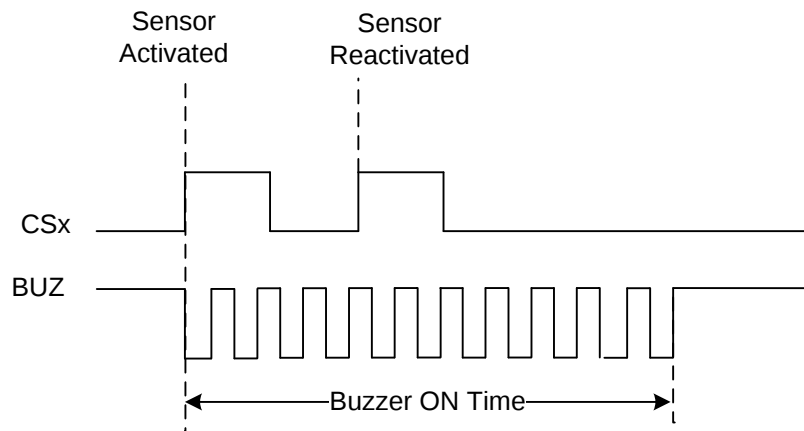
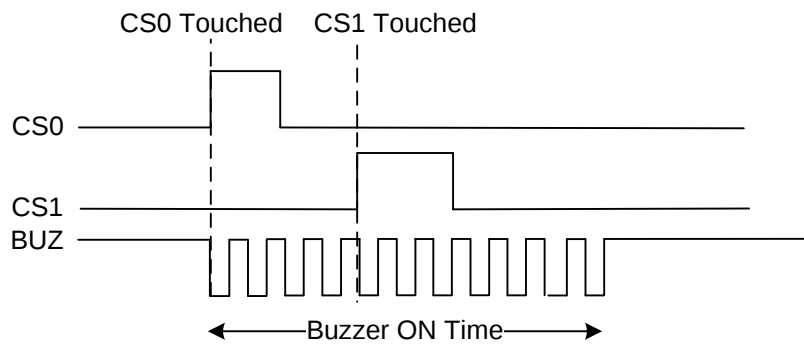


图 5-11. 激活多个传感器时的蜂鸣器操作



5.1.4 器件配置

5.1.4.1 供电电压

CY8CMBR3xxx 控制器的供电电压应在 [EZ-Click](#) 中指定，以便进行正常操作。如果工作电压处于 1.71 ~ 1.89 V 的范围内，请选择 1.8 (±5%) V 的选项；否则选择 1.8 ~ 5.5 V 的选项。该特性不适用于 CY8CMBR3002 CapSense 控制器。

5.1.4.2 I²C 地址

CY8CMBR3xxx 控制器通过 I²C 接口与主处理器进行通信。在典型的 I²C 通信中，I²C 总线具有一个以上的从设备。为了避免从设备之间 I²C 地址发生冲突，CY8CMBR3xxx 提供一个选项，用于指定控制器的 I²C 地址。I²C 地址是一个 7 位值，并它的范围为 8 (0x08) 到 119 (0x77)。CY8CMBR3002 控制器不支持 I²C 通信。

5.1.4.3 扫描周期

扫描周期参数指的是在[查找触摸 \(Look-for-touch\) 模式](#)和[查找接近 \(Look-for-Proximity\) 模式](#)下执行连续传感器扫描之间的时长。该参数的取值范围为 20 到 500 ms，它的分辨率为 20 ms。对于 CY8CMBR3002 CapSense 控制器，该参数值固定为 120 ms。

控制器的响应时间和功耗取决于扫描周期。有关更多信息，请参考[低功耗设计中的注意事项](#)的内容。

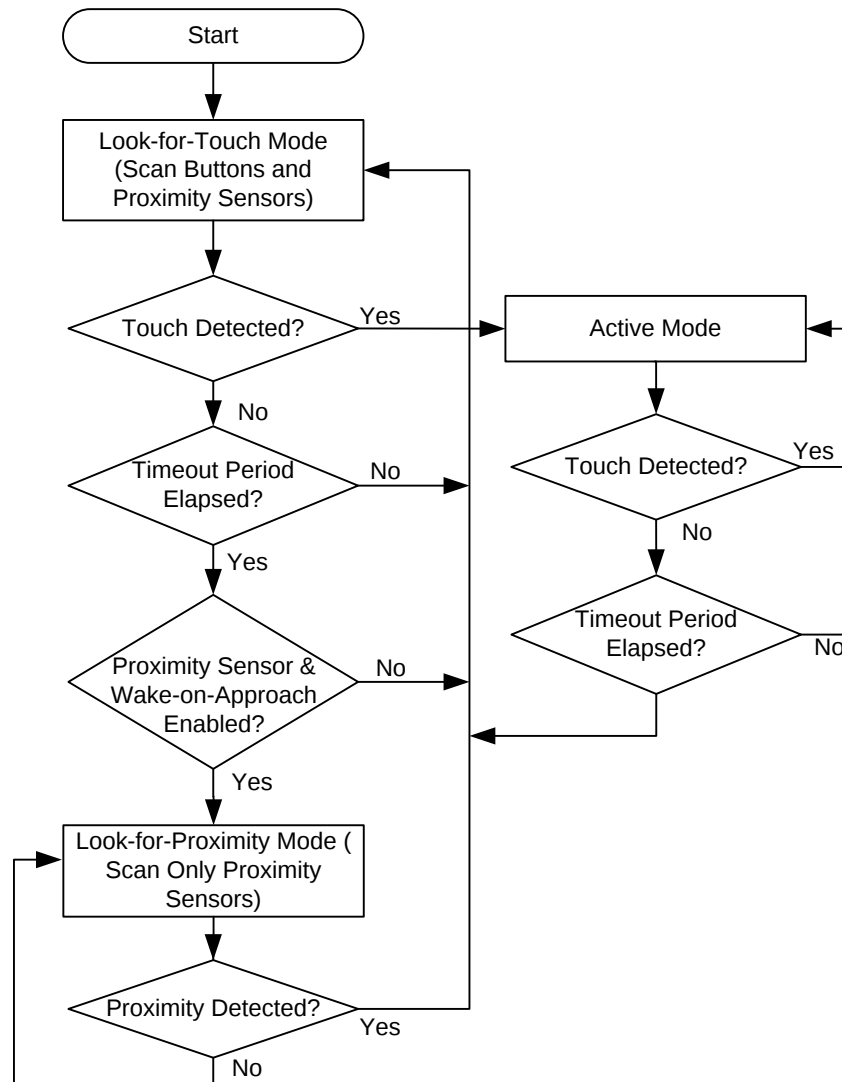
5.1.4.4 状态超时

状态超时时间间隔决定当不触摸传感器时器件从一种操作模式切换到另一种所需的时间。该参数影响到传感器连续触摸的响应时间。根据设计的要求，应设置该参数。这样，控制器能保留活动模式，并对连续触摸提供快速响应。该参数的取值范围为 0 到 63 s，它的分辨率为 1 s。对于 CY8CMBR3002 CapSense 控制器，该参数值固定为 10 s。

5.1.4.5 接近触摸时唤醒

为了降低功耗，CapSense 设计使用“接近触摸时唤醒”功能。使能接近传感器时，该特性允许器件进入[查找接近 \(Look-for-Proximity\) 模式](#)。在查找接近模式下，只扫描接近传感器；当检测到接近时，控制器进入[查找触摸 \(Look-for-touch\) 模式](#)，其中已经扫描所有传感器，如[图 5-12](#)所示。有关更多信息，请参考[低功耗设计中的注意事项](#)的内容。该特性不适用于 CY8CMBR3002 CapSense 控制器。

图 5-12. 从查找触摸模式切换到查找接近模式



5.2 配置 CY8CMBR3xxx

CY8CMBR3xxx系列是一款寄存器可配置的CapSense控制器。该控制器具有128字节的配置寄存器，从地址0x00开始到0x7E。除了CY8CMBR3002 CapSense控制器外，可通过I²C接口来配置CY8CMBR3xxx系列中所有其他控制器。将配置数据保存到非易失性存储器内，以便在控制器断电时仍能够保留配置数据。

有关CY8CMBR3xxx寄存器的详细信息，请参考CY8CMBR3xxx寄存器技术参考手册《TRM》。

使用下面的方法可通过I²C接口配置CY8CMBR3xxx配置寄存器：

1. EZ-Click 2.0
2. 第三方编程器
3. 主机 API
4. 使用 Cypress 提供的 Bridge Control Panel 软件

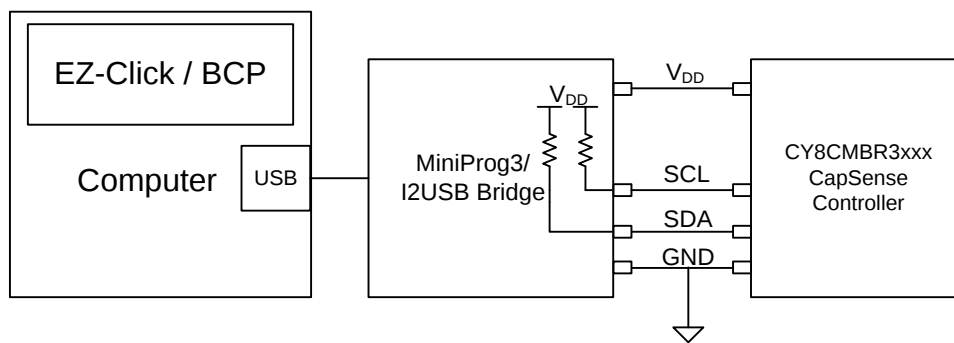
5.2.1 使用 EZ-Click 配置 CY8CMBR3xxx

EZ-Click 为一个简单且直观的图形用户界面工具（GUI），用于配置 CY8CMBR3xxx CapSense 控制器。它取所需的参数，并使用 I²C 接口配置器件。

5.2.1.1 硬件连接

图 5-13 显示的是使用 EZ-Click 2.0 或 BCP 通过 I²C 接口进行配置 CY8CMBR3xxx CapSense 控制器的硬件设计。EZ-Click/BCP 通过 MiniProg3 或 CY3240-I2USB 桥接器与 CY8CMBR3xxx 控制器进行通信。另外，图 5-13 显示的是 MiniProg3 或 I2USB Bridge 和 CY8CMBR3xxx 控制器的连接。因为 MiniProg3 或 I2USB 桥接器的 SCL 和 SDA 引脚被内部上拉到 V_{DD}（有关上拉电阻值的信息，请参考 MiniProg3/I2USB 桥接器套件用户指南），所以请勿将任何外部上拉电阻连接到这些引脚上。不强制将 MiniProg3 或 CY3240-I2USB 桥接器中的 V_{DD} 电压提供给 CY8CMBR3xxx 控制器。CY8CMBR3xxx 控制器还可以通过外部供电。在这种情况下，在 EZ-Click 和 BCP 中请确保电源为 ON（打开）状态，并为 MiniProg3/I2USB 桥接器设置的电压值等于 CY8CMBR3xxx V_{DD} 电压值。

图 5-13. 使用 EZ-Click 或 BCP 配置 CY8CMBR3xxx 的硬件设置



注意：在 CY3280-MBR3 套件中，PSoC 5LP 器件被作为 I2USB 桥接器使用，因此您不能使用 MiniProg3 或 CY3280-I2USB 桥接器与 CY8CMBR3116 CapSense 控制器进行通信。更多有关信息，请参考 CY3280-MBR3 套件用户指南。

EZ-Click

本部分对 EZ-Click 进行了简单介绍。有关如何配置 CY8CMBR3xxx 控制器的详细信息，请参考 EZ-Click 用户指南中介绍的内容。

EZ-Click 提供以下功能：

- 一个图形用户界面工具（GUI）用于配置 CY8CMBR3xxx 控制器。
- 允许调试传感器参数，并实时查看传感器调试数据。
- 支持生产线测试，并显示系统的结果以及 CapSense 传感器的信噪比。
- 允许保存配置，并在多个样本上使用它。
- 生成包括 128 字节的配置数据阵列的头文件，以便使用主机 API 配置 CY8CMBR3xxx 控制器，如图 5-16 所示。更多有关信息，请参考 5.2.3 一节。
- 生成配置的十六进制文件，以通过第三方编程工具来配置器件。有关详细信息，请参考 5.2.2 一节。
- 生成配置文件，包括所需的 I²C 指令，以便使用 BCP 来配置器件，如图 5-23 所示。有关更多信息，请参考 5.2.4 一节。

EZ-Click 共有 5 个选项卡，如图 5-15 所示：

- **Start Page**（起始页面）：该页面显示了创建新的项目、生成配置文件和配置控制器的 MBR 系列的各步骤。
- **CapSense sensor configuration**（CapSense 传感器配置）：在该选项卡中，可以选择传感器数，并对每个传感器设置多种参数和功能。
- **Global configuration**（全局配置）：在该选项卡中，可以使能 GPO 并指定全局参数。
- **CapSense output**（CapSense 输出）：在该选项卡中，可以监控传感器调试数据，用于调试。

- System diagnostics（系统诊断）：在该选项卡中，可以查看每个传感器的系统诊断结果。

注意：仅当创建一个新的项目或者打开一个现有的项目时，才能访问“CapSense 传感器配置”、“全局配置”、“CapSense 输出”和“系统诊断”。

图 5-14. 使用各种 EZ-Click 生成的文件配置 CY8CMBR3xxx

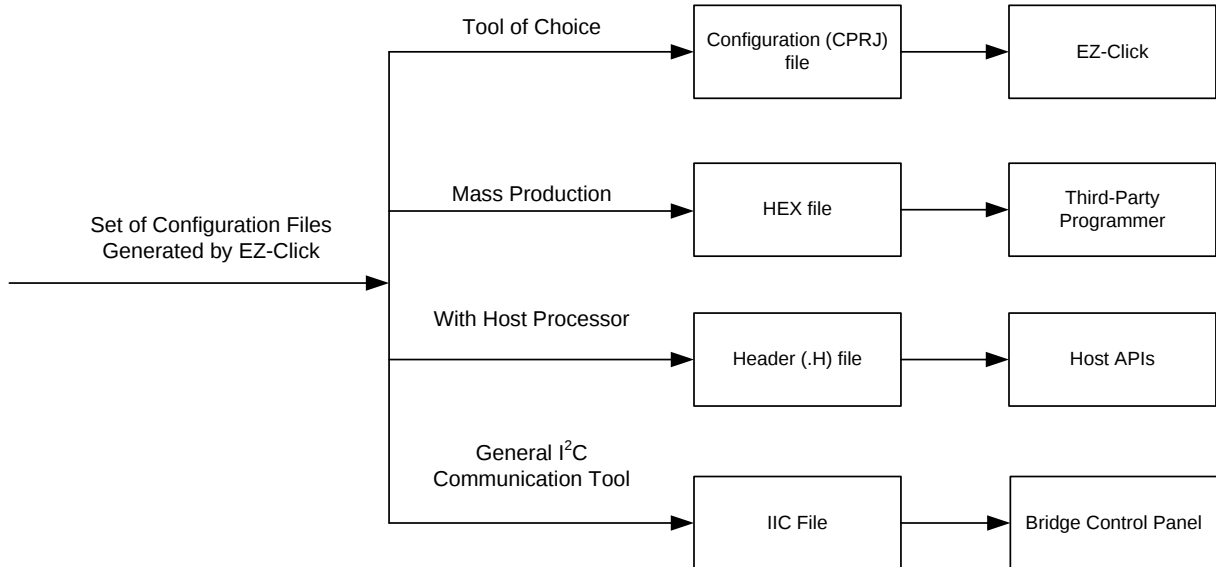


图 5-15. EZ-Click 定制器工具

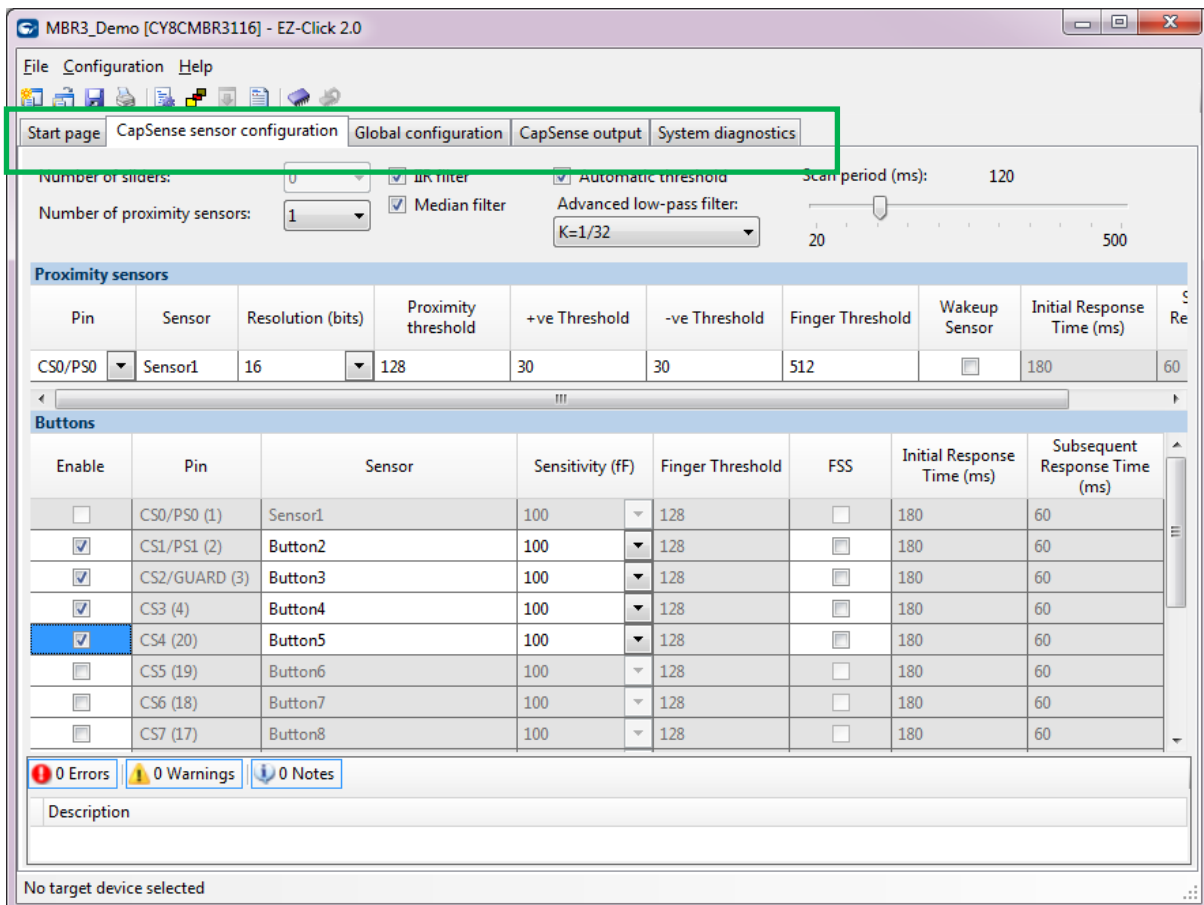


图 5-16. 由 EZ-Click 定制器工具生成的 C 配置文件

```

1  /* Project: C:\MBR3_Demo\MBR3_Demo.cprj
2  * Generated: 02/18/2014 11:20:08 AM +05:30 */
3  const unsigned char CY8CMBR3116_configuration[128] = {
4      0x78u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
5      0x00u, 0x00u, 0x00u, 0x00u, 0x7Fu, 0x7Fu, 0x7Fu, 0x80u,
6      0x80u, 0x80u, 0x80u, 0x7Fu, 0x7Fu, 0x7Fu, 0x7Fu, 0x7Fu,
7      0x7Fu, 0x7Fu, 0x7Fu, 0x7Fu, 0x03u, 0x00u, 0x00u, 0x00u,
8      0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x80u,
9      0x05u, 0x00u, 0x00u, 0x02u, 0x00u, 0x02u, 0x00u, 0x00u,
10     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
11     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x01u, 0x01u,
12     0x00u, 0xFFu, 0xFFu, 0xFFu, 0xFFu, 0xFFu, 0xFFu, 0xFFu,
13     0xFFu, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x03u, 0x01u, 0x58u,
14     0x00u, 0x37u, 0x06u, 0x00u, 0x00u, 0x0Au, 0x00u, 0x00u,
15     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
16     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
17     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
18     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u,
19     0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x00u, 0x41u, 0x41u
20 };

```

5.2.2 使用第三方编程器配置 CY8CMBR3xxx

为了在大规模生产过程中快速配置大量的器件，赛普拉斯建议使用 RPM Systems Corporation 的第三方编程器。为了使用第三方编程器配置 CY8CMBR3xxx 控制器，请使用您配置的十六进制文件（该配置由 EZ-Click 生成）。欲了解更多信息，请联系 RPM Systems Corporation。

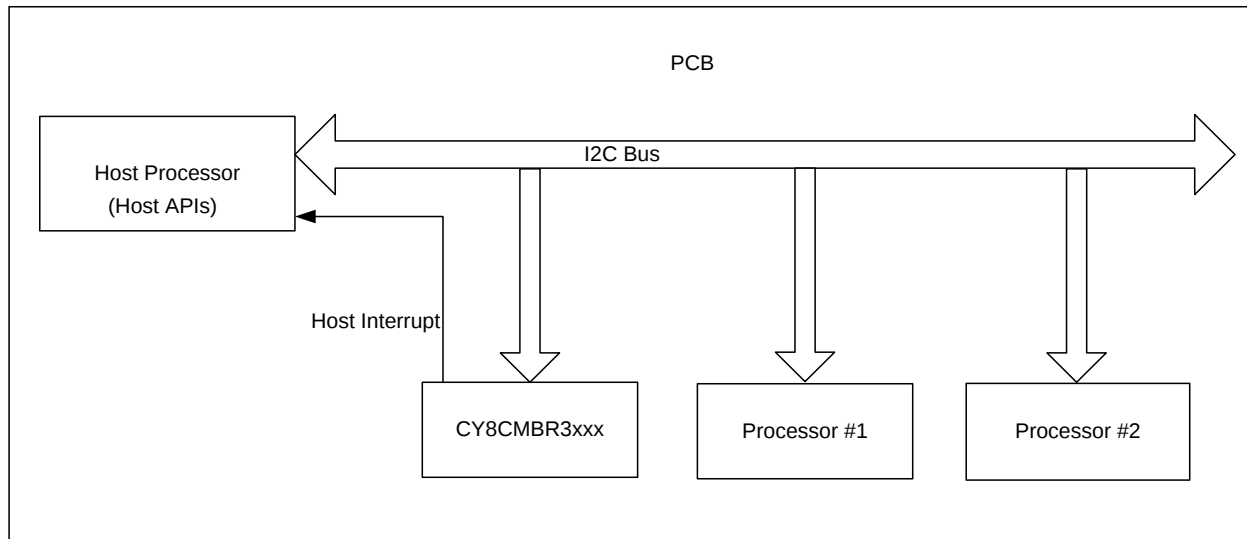
有关 CY8CMBR3xxx 系列编程规范的信息，请参考 [CY8CMBR3xxx 器件编程规范](#)。

5.2.3 使用主机 API 配置 CY8CMBR3xxx

图 5-17 显示的是主处理器通过 I²C 与 CY8CMBR3xxx 控制器以及同一个 PCB 上其它处理器进行通信的系统。主机可以在运行时从 CY8CMBR3xxx 控制器中回读传感器状态；如果需要的话，可调整 CY8CMBR3xxx 控制器操作，并且还重新配置它，以符合各种要求。

为了使用主处理器配置板上的 CY8CMBR3xxx 控制器，赛普拉斯提供了一组库文件（称为主机 API），可将这些库文件集成到主处理器固件内。该部分介绍了主机 API，并解释了如何使用它们来配置 CY8CMBR3xxx 控制器。

图 5-17. 系统演示主机 API 用法



5.2.3.1 主机 API 概况

主机 API 是赛普拉斯提供的一系列 C 型功能，用于使能主处理器通过 I²C 接口直接与 CY8CMBR3xxx 控制器通信。

这些 API 可以用于：

1. 配置 CY8CMBR3xxx 控制器
2. 将 I²C 指令发布给 CY8CMBR3xxx 控制器
3. 从 CY8CMBR3xxx 寄存器图中读取信息

5.2.3.2 主机 API 的介绍和用法

主机 API 分为两部分：顶层 API 和底层 API。

5.2.3.2.1 顶层 API

顶层 API 实现 C 型软件所需的功能，并依赖于底层 API，以便与 CY8CMBR3xxx 控制器建立物理连接。表 5-2 中列出了顶层 API 和它们的用法。附录中介绍了原型、参数和返回值。

表 5-2. 顶层主机 API

序号	API	说明和用法
1	CY8CMBR3xxx_WriteData()	通用 API 用于将数据写入到器件内。使用该 API 可以将数据写入到一个或多个 CY8CMBR3xxx 控制器寄存器内。
2	CY8CMBR3xxx_ReadData()	通用 API 用于从 CY8CMBR3xxx 控制器中读取数据。使用该 API 可以从一个或多个 CY8CMBR3xxx 控制器寄存器中读取数据。
3	CY8CMBR3xxx_WriteDualByte()	通过该 API 可以将数据写入到 CY8CMBR3xxx 控制器中的两字节寄存器内。 例如，通过使用该 API 可以将数据写入到 16 位传感器内，以使能寄存器。
4	CY8CMBR3xxx_ReadDualByte()	通过该 API 可以从 CY8CMBR3xxx 控制器中的两字节寄存器中读取数据。 例如，通过使用该 API 可以从 16 位传感器状态寄存器中读取数据。

5	CY8CMBR3xxx_SendCommand()	该 API 用于将指令发送给 CY8CMBR3xxx 控制器，比如将配置保存到非易失性存储器内、复位 CY8CMBR3xxx 控制器等。
6	CY8CMBR3xxx_CheckCommandStatus()	该 API 用于检查发送到 CY8CMBR3xxx 控制器中的最后指令状态。建议先在主处理器中调用该 API，然后将新的指令发送到 CY8CMBR3xxx 控制器内。
7	CY8CMBR3xxx_Configure()	该 API 用于配置 CY8CMBR3xxx 控制器整个 128 字节的配置寄存器。 要使用该 API，请在 EZ-Click 中创建一个配置，并在主机固件中使用所生成的头文件。
8	CY8CMBR3xxx_CalculateCrc()	该 API 用于计算已给定配置的 CRC 校验和。
9	CY8CMBR3xxx_VerifyDeviceOnBus()	该 API 用于验证 I ² C 总线上的 CY8CMBR3xxx 器件是否是所需要使用的 ¹⁰ 。从主处理器中调用该 API，以确认正确的 I ² C 从设备是否位于总线上。
10	CY8CMBR3xxx_SetDebugDataSensorId()	该 API 用于为传感器调试数据设置编号，该数据必须从 CY8CMBR3xxx 控制器中读取。
11	CY8CMBR3xxx_ReadSensorDebugData()	该 API 用于读取传感器调试数据。该 API 有助于了解传感器的性能。
12	CY8CMBR3xxx_ReadDiffCounts()	该 API 用于读取 CY8CMBR3xxx 控制器中所有传感器的差值计数。
13	CY8CMBR3xxx_ReadSensorStatus()	该 API 用于读取 CY8CMBR3xxx 控制器中所有传感器的状态。当 CY8CMBR3xxx 控制器发出主机中断脉冲时，调用该 API 可以知道被触发的传感器。

5.2.3.2.2 低层 API

低层 API 负责通过 I²C 接口实现主处理器和 CY8CMBR3xxx 器件之间的物理通信，因此它依赖于硬件。这些 API 默认支持 PSoC4 架构，并要求进行修改以支持特定的主处理器。

CY8CMBR3xxx 控制器处于深度睡眠模式时，第一个 I²C 数据传输未被确认（NACK）。如果 CY8CMBR3xxx 控制器未确认 I²C 数据传输，那么底层 API 必须在接下来的 340 ms 内重试两次，以成功完成数据传输。请参考 CY8CMBR3xxx 数据手册中的“主机通信协议”。为能实现该操作，请禁用主处理器中的所有中断（I²C 的中断除外），或者确保在 340 ms 的时间内完成所有挂起中断的传输。如果这里两次连续的 I²C 数据传输消耗的时间间隔大于 340 ms 的延迟，则器件可能进入深度睡眠模式，并且主机收到一个 NACK。

底层 API 包括三个，即：Host_LowLevelWrite()、Host_LowLevelRead() 以及 Host_LowLevelDelay()。附录中介绍了原型、参数和返回值。

表 5-3. 底层主机 API

序号	API	说明和用法
1	Host_LowLevelWrite()	该 API 用于通过 I ² C 接口将数据写入到 CY8CMBR3xxx 控制器中寄存器图内。
2	Host_LowLevelRead()	该 API 用于通过 I ² C 接口读取 CY8CMBR3xxx 控制器中寄存器图的数据。
3	Host_LowLevelDelay()	该 API 实现了顶层 API 所使用的延时功能。延迟周期的单位为毫秒。在所需时间内执行一个代码段即可得到该延迟。

¹⁰ 主机 API 不可用于在同一 I²C 总线上配置 CY8CMBR3xxx 系列的不同控制器。

5.2.3.3 主机 API 文件结构

表 5-4 中列出的文件构成了主机 API 库。

表 5-4. 主机 API 的文件结构

序号	文件名称	说明
1	CY8CMBR3xxx_Device.h	该文件包括的各个宏用于定义 CY8CMBR3xxx 系列中各种器件。另外，它还定义了主机 API 正在使用的器件。
2	CY8CMBR3xxx_Registers.h	该文件包括的各个宏用于定义 CY8CMBR3xxx 系列中所有寄存器的偏移地址。使用这些宏来参考寄存器，替换直接使用地址。只有寄存器可用于目前所使用的器件时，才能访问每个寄存器的宏。
3	CY8CMBR3xxx_CommandsAndConfig.h	该文件包括用于定义指令寄存器（CTRL_CMD）的各种操作码的宏。
4	CY8CMBR3xxx_APIs.h	该文件包含所有顶层 API 的声明。该文件还指明了结构定义，用于保持从设备中读取的传感器调试数据和传感器状态。
5	CY8CMBR3xxx_CRC.c	该文件包含了 API，用于为某个给定配置的 CY8CMBR3xxx 器件进行计算 CRC。
6	CY8CMBR3xxx_APIs.c	除了用于计算 CRC 的 API 外，该文件还包含了所有顶层 API 的定义。
7	CY8CMBR3xxx_HostFunctions.h	该文件包含了与主机相关的底层 API 的声明。
8	CY8CMBR3xxx_HostFunctions.c	该文件包含了对底层 API 的定义。修改这些 API 的内容，以符合主处理器的 I²C 实现。

5.2.3.4 宏

具有两个宏，即“CY8CMBR3XXX_DEVICE”和“CY8CMBR3xxx_SYNC_COUNTER_MATCH_RETRY”，它们被分别定义在“CY8CMBR3xxx_Device.h”和“CY8CMBR3xxx_APIs.h”中。需要修改这两个宏，以便在同步计数器失谐的情况下定义 CY8CMBR3xxx 控制器和 I²C 的重试次数。

1. CY8CMBR3xxx_DEVICE

该宏定义了主机所访问的控制器。该控制器可以是 CY8CMBR3xxx 系列中除 CY8CMBR3002 外（因为该控制器不支持 I²C 通信）的另一个控制器。在 CY8CMBR3xxx_Device.h 中定义了 CY8CMBR3xxx 系列中的另一个控制器。

2. CY8CMBR3xxx_SYNC_COUNTER_MATCH_RETRY

该宏定义在读取操作中与同步计数器不匹配时，API CY8CMBR3xxx_ReadSensorDebugData() 和 CY8CMBR3xxx_ReadDiffCounts() 重新尝试将 I²C 通信与 CY8CMBR3xxx 控制器建立连接的次数。通常，如果读取操作在第一次尝试时失败，它会在第二次尝试中成功。建议将尝试次数设置为一个小于 10 的值。更多有关同步计数器的信息，请参考 CY8CMBR3xxx 寄存器技术参考手册《TRM》。

5.2.3.5 演示项目及其用法

要快速熟习使用主机 API，使用 PSoC4 创建一个示范项目作为主处理器使用。该演示展示了如何使用主机 API 通过 PSoC4 Pioneer 套件来配置 CY3280 - MBR3 评估套件中的 CY8CMBR3116 控制器。

下面列出了该演示所需的硬件：

- CY3280-MBR3 评估套件
- CY8CKIT-042 PSoC4 Pioneer 套件
- USB A 至 mini-B 的电缆
- 个人计算机/笔记本电脑

下面列出了该演示所需的软件：

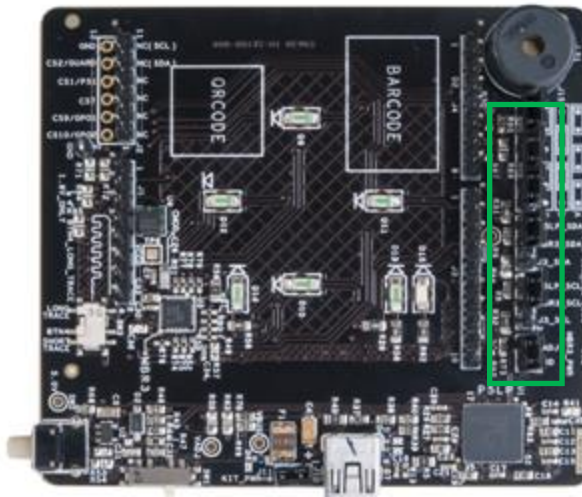
- PSoC Programmer 3.20.0

注意：应该安装好 PSoC4 pioneer 套件的驱动程序。有关安装套件驱动程序的详细信息，请参考 CY8CKIT-042 PSoC 4 Pioneer 套件指南。

为了使用 PSoC4 Pioneer 套件进行配置 MBR3 评估套件，请按照下面的步骤进行：

1. 更改 MBR3 评估套件中接头 J13 的跳线器位置，以便将引脚 J13-2 和 J13-3 连接在一起。
2. 更改 MBR3 评估套件上接头 J14 的跳线器位置，以便将引脚 J14-2 和 J14-3 连接在一起。
3. 将 MBR3 评估套件上接头 J15 的跳线器位置更改为位置 A。这样套件能够支持蜂鸣器和主机中断输出。

图 5-18. MBR3 评估套件上跳线器 J13、J14 和 J15 的位置



4. 将 MBR3 套件安装在 PSoC4 Pioneer 套件顶层上，以便使两个套件的 USB 端口相互对齐。USB 端口相互对齐时，将 PSoC 4 Pioneer Kit 上的 J1、J2、J3 和 J4 接头分别连接至 CY3280-MBR3 EVK 上的 J1、J2、J3 和 J4 接头。
5. 通过 USB A 至 mini-B 的线缆将 PSoC4 Pioneer 套件连接到个人计算机上，使之作为 MBR3 评估套件的一部分。MBR3 套件上电源指示灯（红色）发光。图 5-19 显示了最终的连接情况。完成硬件设置时，可以将项目下载到 PSoC4 主机内。

图 5-19. 安装在 PSoC4 Pioneer 套件顶层上的 MBR3 套件



6. 在此处下载“CY8CMBR3xxx_Host_APIs_Demo_Project.zip”项目，并从.zip文件中提取内容。
7. 打开 PSoC Programmer。它的位置为“Windows 启动菜单→All programs→Cypress→PSoC Programmer 3.20.0”。如果尚未在 PC 上安装 PSoC Programmer，请从 PSoC Programmer 3.20.0 中下载并安装它。
8. 在 PSoC Programmer 中，点击 File→File Load，然后导航到“CY8CMBR3xxx_Host APIs_Demo_Project”被提取的目录位置。
9. 依次打开“CY8CMBR3xxx_Host APIs_Demo_Project→PSoC4_Pioneer_Kit_Demo→PSoC4_Pioneer_Kit_Demo.cydsn→CortexM0→ARM_GCC_473→Release→PSoC4_Pioneer_Kit_Demo.hex”，然后点击打开。
10. 请确保 PSoC Programmer 被连接到 KitProg，如图 5-20 中所示。点击 Program（编程）按键，从而能够使用给定的十六进制文件来编程 PSoC4 Pioneer 套件，如图 5-20 中所示。
11. 一旦 PSoC4 Pioneer 套件被编程，PSoC4 Pioneer 套件将根据演示配置来配置 MBR3 评估套件上的 CY8CMBR3116 控制器。如果成功配置了 CY8CMBR3116 控制器，那么 PSoC4 Pioneer 套件上 RGB LED D9 会发出绿色光亮。如果未成功配置 CY8CMBR3xxx 控制器，则 PSoC4 Pioneer 套件上 RGB LED D9 会发出红色光亮。

MBR3 评估套件的配置如下：

- 4 个 CapSense 按键传感器
- 一个接近传感器
- 一个接近 LED
- 蜂鸣器输出
- 主机中断输出
- 主机控制的 GPO

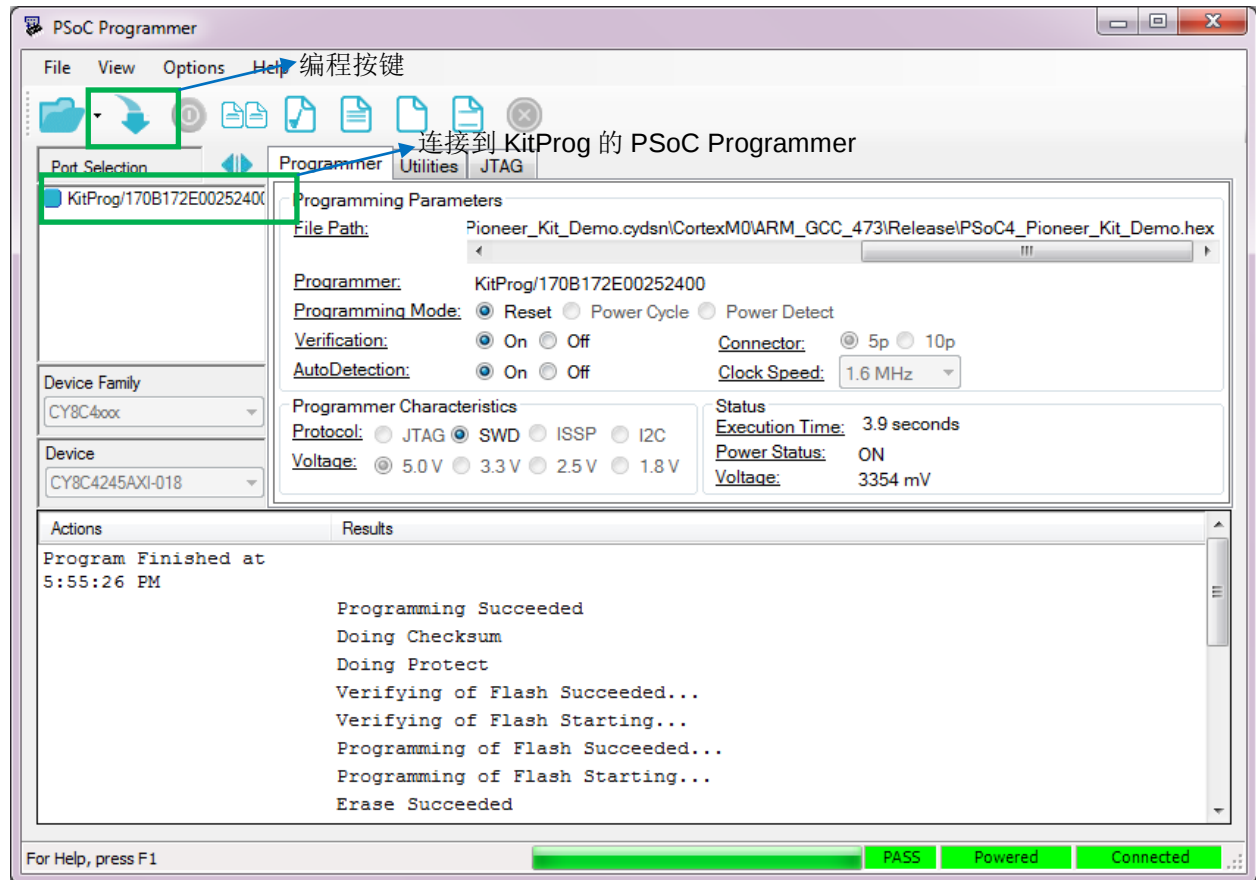
每当 CY8CMBR3116 控制器发送一个主机中断脉冲时，主处理器（PSoC4）都会打开/关闭 MBR3 评估套件上的 LED。主机中断被触发时，主机会读取 CY8CMBR3116 控制器中所有传感器的状态，如果传感器状态被设置为 1，则会打开相应 LED。

12. PSoC4 主机配置 CY8CMBR3116 控制器后，验证 MBR3 评估套件的下述各项功能：
 - 接近：以 3 厘米的高度将手/手指悬浮在 MBR3 评估套件上方；接近 LED 闪烁一次，以指示检测到接近物体。触摸接近传感器（在 MBR3 套件的边缘上标记为白色矩形）；蜂鸣器在 500 毫秒时间内发出嘟嘟声，并且只

要接近传感器被触摸，接近 LED 仍会保持打开状态。当手/手指不再接触接近传感器时，接近 LED 将被关闭，以指示没有检测到接近触摸。将手/手指进一步远离接近传感器，接近 LED 至少会闪烁一次，以指示没有检测到接近触摸。

- CapSense 按键：按下任何一个 CapSense 按键，相应的 LED 为打开状态，并且蜂鸣器会在 500 ms 时间内发出嘟嘟声。

图 5-20. 使用 PSoC Programmer 进行编程 PSoC4 Pioneer 套件



5.2.4 使用 BCP（Bridge Control Panel）软件配置 CY8CMBR3xxx

Bridge Control Panel（BCP）是一个软件工具，该工具可作为 I²C 主设备使用，从而能够通过使用 CY3240-I2USB 或 MiniProg3 与 I²C 从设备进行通信，如图 5-21 中所示。

图 5-21. BCP 与 I²C 从设备通信



注意： CY3280-MBR3 评估套件不需要额外的 I2USB/MiniProg3 硬件。因为该套件具有一个板上 I2C-USB 桥接器，用于连接到 BCP。要想将板上 I2C-USB 桥接器连接到 BCP，请更改 MBR3 评估套件上接头 J13 跳线器的位置，以便将 J13-1 引脚和 J13-2 引脚连接在一起，同时更改接头 J14 跳线器的位置，以便将 J13-1 和 J13-2 引脚连接在一起。

该工具可用于使用 I²C 接口来配置 CY8CMBR3xxx 控制器，并读取调试数据。强烈建议使用 EZ-Click 配置 CY8CMBR3xxx 控制器，而不是使用 BCP。但 BCP 仍可以用于配置 CY8CMBR3xxx 控制器。例如，要覆盖阈值参数并手动指定它们，请使用 BCP 工具。有关使用 BCP 工具的更多信息，请点击 BCP 工具中 BCP “Help” 菜单。

本部分介绍了 CY8CMBR3xxx 控制器中的各种寄存器名称。有关这些寄存器的定义和地址信息，请参考 CY8CMBR3xxx 寄存器 TRM。

5.2.4.1 使用 BCP (Bridge Control Panel) 软件配置 CY8CMBR3xxx 的步骤

图 5-24 显示的是使用 BCP 工具配置 CY8CMBR3xxx 的流程图。请参考图 5-13，了解使用 BCP 与 CY8CMBR3xxx 控制器进行通信时所需要的硬件连接。

使用 BCP 配置 CY8CMBR3xxx 的流程可以分为以下四个步骤：

1. 将配置数据发送到 CY8CMBR3xxx 控制器内。下面列出了各个配置数据：

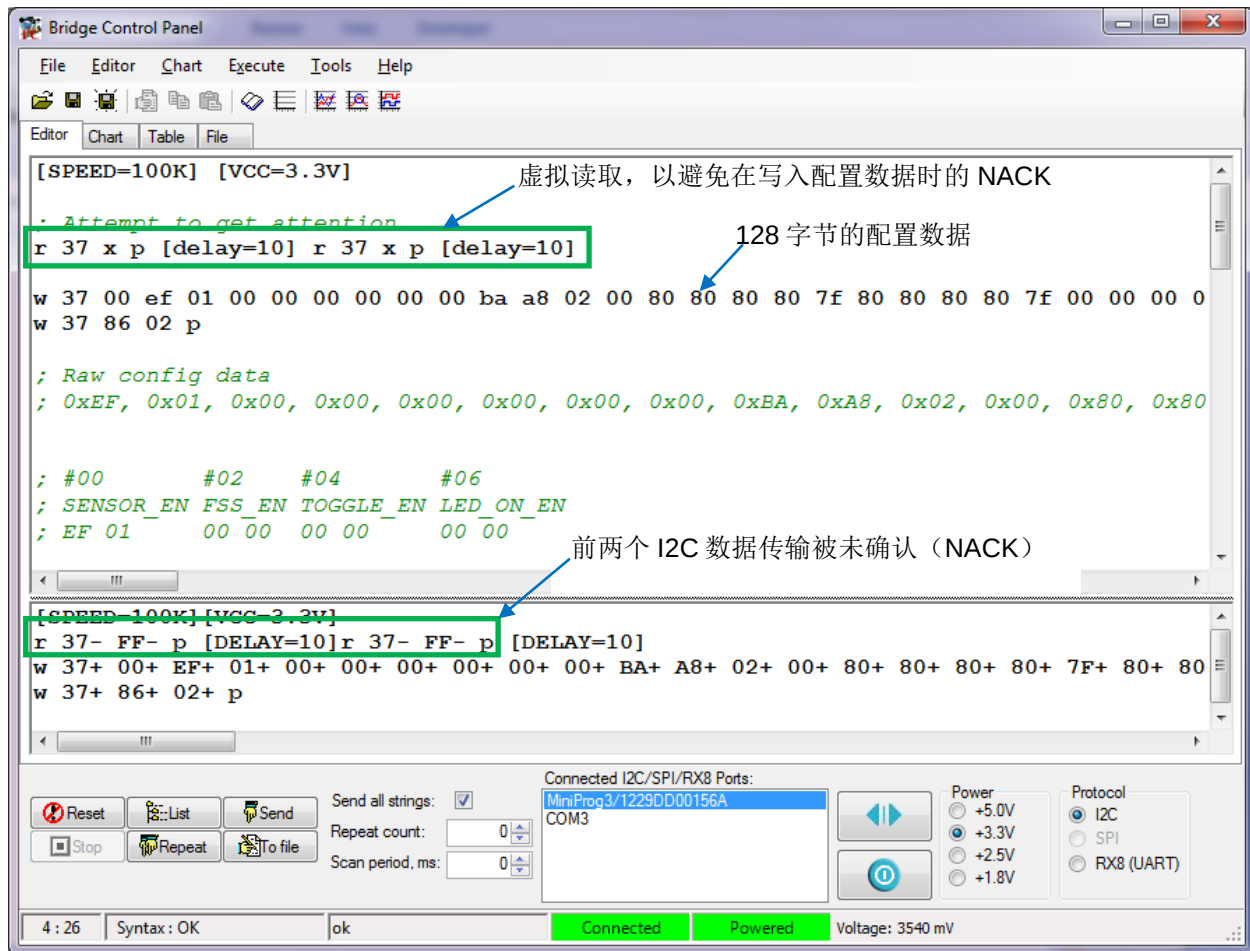
a) EZ-Click 所生成的配置文件（<project_name>.iic 文件）：

图 5-22 显示了使用 EZ-Click 所生成的配置文件来配置 CY8CMBR3xxx 控制器。该方法很少使用，因为 CY8CMBR3xxx 可以直接使用 EZ-Click 进行配置。请点击 BCP 工具中的 “Help”（帮助）选项，了解打开 BCP 工具中的任意 IIC（配置）文件并发送指令到从设备的介绍。

图 5-22. EZ-Click 生成 IIC 文件，从而使用 BCP 来配置 CY8CMBR3xxx 控制器

Name	Date modified	Type	Size
 MBR3_Demo		ess EZ-Click ...	2 KB
 MBR3_Demo	02/17/2014 8:29 PM	H File	2 KB
 MBR3_Demo.hex	02/17/2014 8:29 PM	HEX File	1 KB
 MBR3_Demo.iic	02/17/2014 8:29 PM	IIC File	5 KB

图 5-23. 由 EZ-Click 定制器工具生成的配置文件示例



b) 输入 BCP 工具的自定义指令：

通过在 BCP 中输入自定义指令来配置 CY8CMBR3xxx，如图 5-26 所示。使用这种方法时，可以配置所有 128 字节的配置寄存器或任何特定寄存器。使用自定义指令配置 CY8CMBR3xxx 时，需要计算配置数据前 126 个字节的 CRC（循环冗余校验），并将其写入到 CONFIG_CRC (0x7E)寄存器内。CY8CMBR3xxx 控制器将使用该 CRC 值来验证配置数据的完整性。

注意：当 I²C 指令（读/写）被发送到 CY8CMBR3xxx 控制器内时，它可能否定 I²C 数据传输。在这种情况下，在写入或读取任意寄存器时请插入虚拟 I²C 读取操作，如图 5-26 所示。

可以通过以下两种方法来计算配置数据前 126 个字节的 CRC 值：

- 使用 CY8CMBR3xxx 控制器计算 CRC 值：当 CMD_OP_CODE 值（即 3）被写入到 CTRL_CMD (0x86) 寄存器内时，CY8CMBR3xxx 控制器会计算配置数据的校验和，如图 5-25 所示。经过 220 ms 时间后，已计算的 CRC 值将被存储到 CALC_CRC (0x94)寄存器中。用户必须从 CALC_CRC (0x94)寄存器中读取 CRC 值，并手动将它写入到 CONFIG_CRC (0x7E)寄存器内。只有对 CY8CMBR3xxx 控制器进行测试和调试时才会使用这种方法。为了在生产过程中配置 CY8CMBR3xxx 控制器，请使用与主机 API 一起提供的 CRC API。

- 使用主机 API 来计算 CRC 值：赛普拉斯提供了主机 API，用以使用主处理器配置 CY8CMBR3xxx。主机 API 具有源代码，用于计算 126 个配置数据字节的 CRC 值。用户应使用该代码来计算 CRC，并将计算得出的 CRC 值写入到 CONFIG_CRC (0x7E) 寄存器内，如图 5-26 所示。5.2.4.2 一节说明了计算 CRC 的流程。强烈建议使用这种方法来计算 CRC 值。
2. 一旦将配置数据和 CRC 值写入到 CY8CMBR3xxx 控制器寄存器内，请将 CMD_OP_CODE 的数值 2 写入到 CTRL_CMD (0x86) 寄存器内，这样能够将配置数据保存到非易失性存储器内。
 3. 将 CMD_OP_CODE 的数值 2 写入到 CTRL_CMD (0x86) 寄存器内后等待 220 ms，然后再读取 CTRL_CMD_STATUS (0x88) 寄存器，以便检查配置数据是否成功被存储到非易失性存储器内。
如果成功，CTRL_CMD_STATUS (0x88) 寄存器的值为 0。如果 CTRL_CMD_STATUS 寄存器的值为 ‘0’，通过将 CMD_OP_CODE 的数值 255 写入到 CTRL_CMD (0x86) 寄存器内来发送复位指令。

如果 CTRL_CMD_STATUS (0x88) 寄存器的值为 ‘1’，便表示配置数据未被保存到非易失性存储器内。这时，请读取 CTRL_CMD_ERR (0x89) 寄存器，以了解保存配置数据到非易失性存储器内失败的原因。

- CTRL_CMD_ERR 中的数值 253 表示写入到非易失性存储器内失败。在这种情况下，应重新写入配置数据，并从第二步重新开始。
- CTRL_CMD_ERR 寄存器中的数值 254 表示写入到 CONFIG_CRC (0x7E) 的 CRC 值不正确。这时，对全部 126 个配置数据字节进行重新计算 CRC 值，并写入 CONFIG_CRC (0x7E) 寄存器，然后从第二步重新开始。
- CTRL_CMD_ERR 寄存器中的数值 255 表示写入到 CTRL_CMD (0x86) 寄存器内的 CMD_OP_CODE 值无效。这时，应发送一个有效的 CMD_OP_CODE 值（即 ‘2’），并重复第三步。

图 5-24. 使用 BCP 配置 CY8CMBR3xxx 的流程图

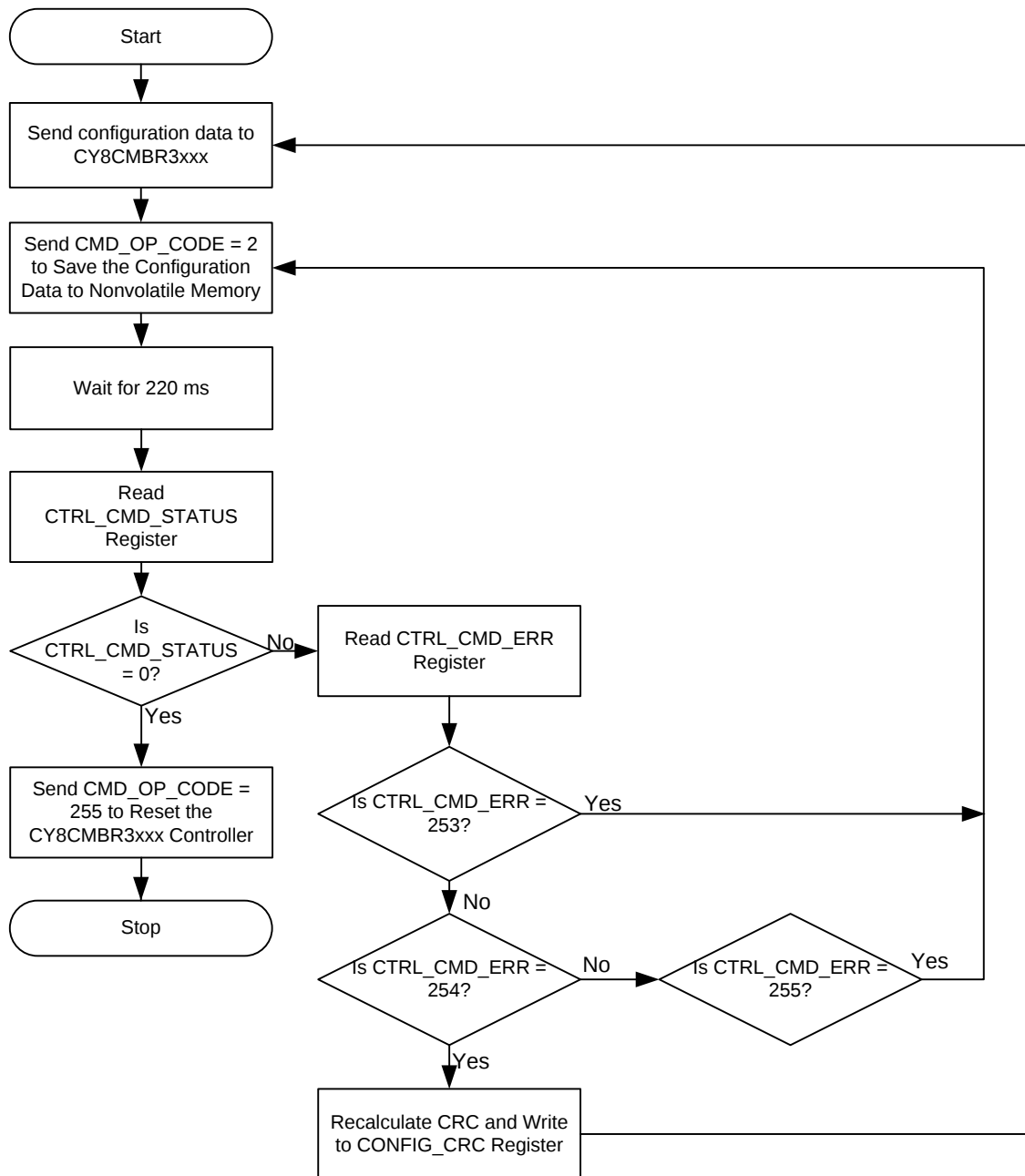
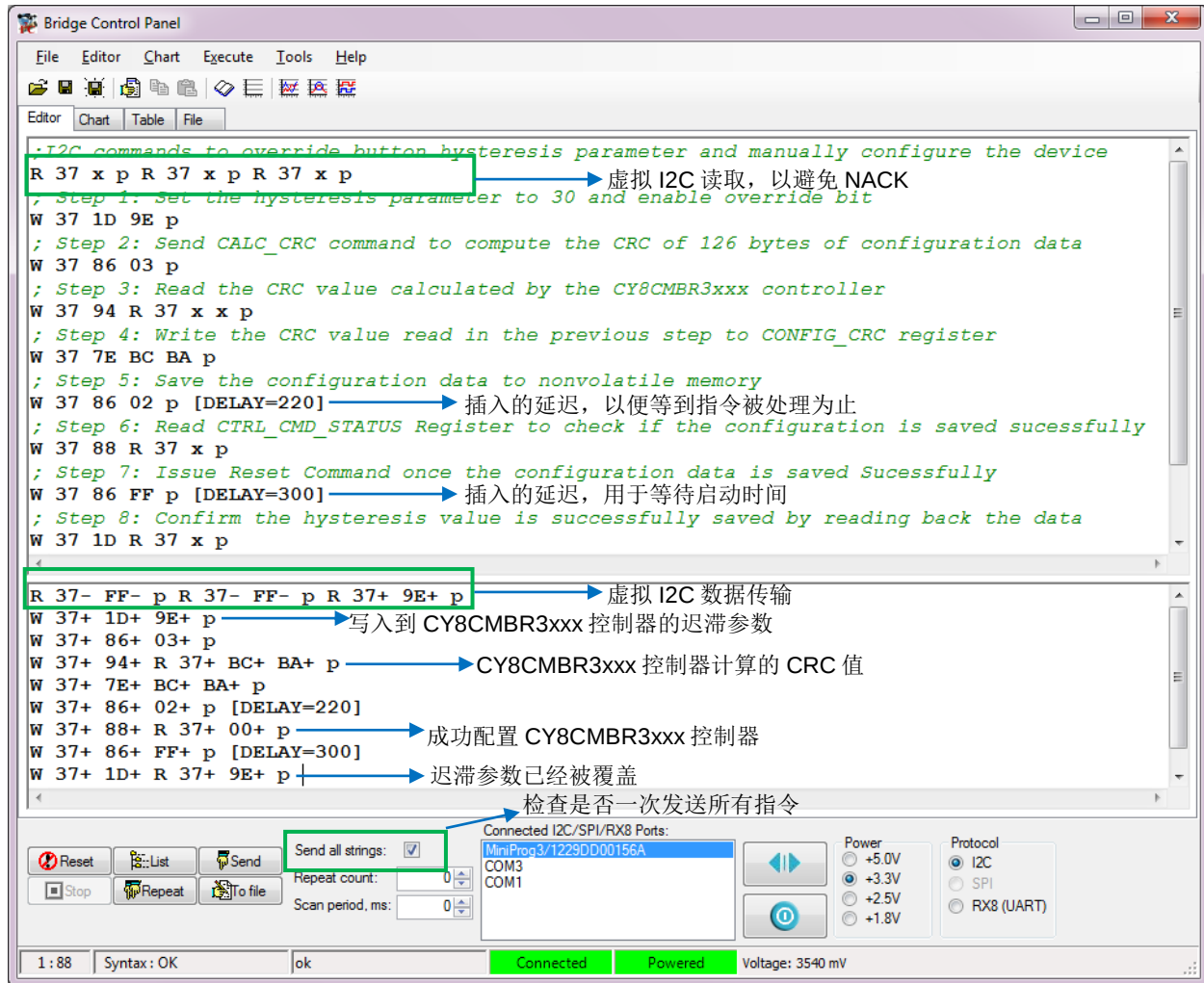
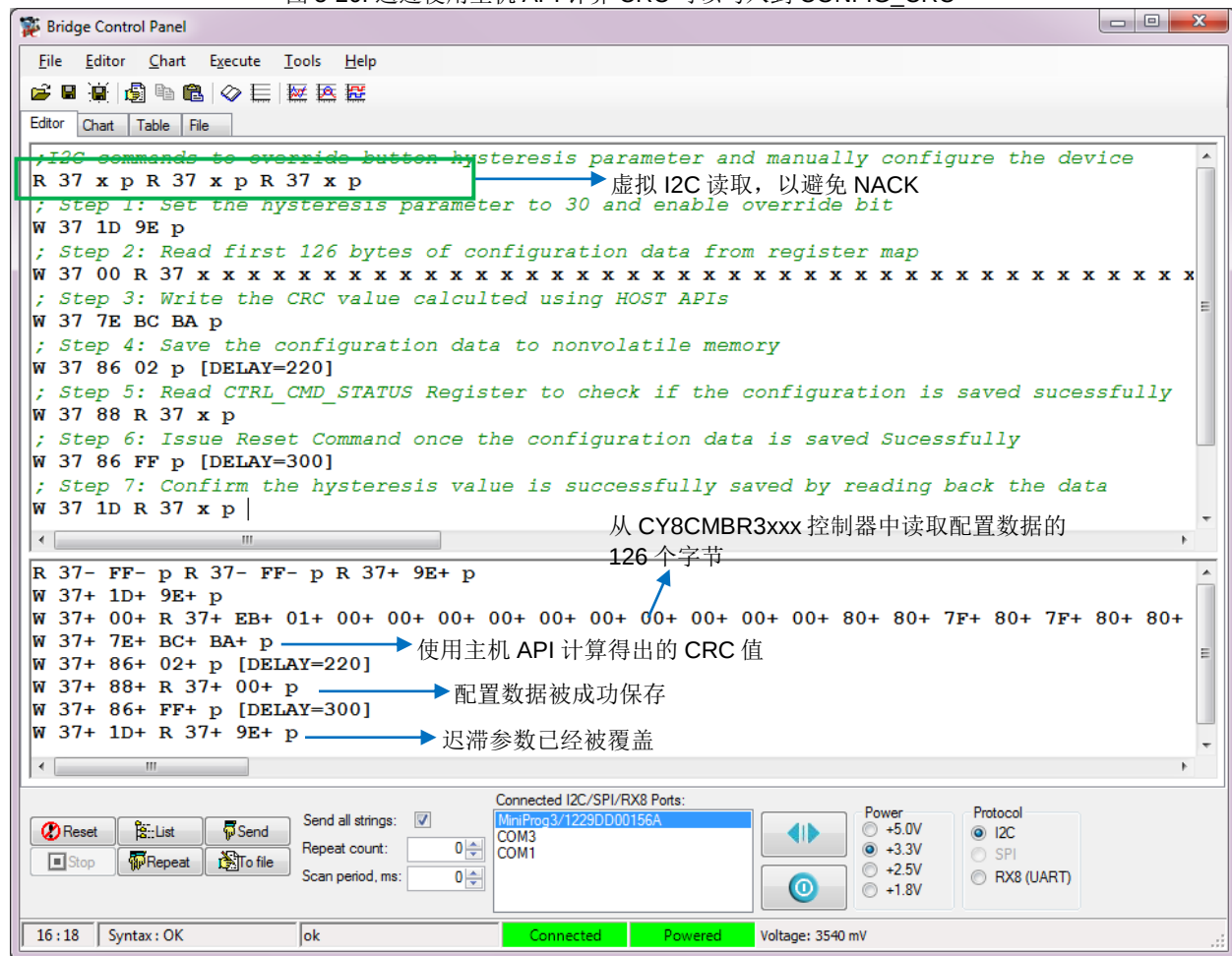


图 5-25. 通过 CY8CMBR3xxx 控制器计算 CRC



注意：通过检查“send all strings”选框并点击“send（发送）”按键，可以同时发送输入到 BCP 中的所有指令。

图 5-26. 通过使用主机 API 计算 CRC 可以写入到 CONFIG_CRC



5.2.4.2 使用主机 API 计算 CRC16-CCITT 的流程

CY8CMBR3xxx 控制器要求用户指定 126 个配置数据字节的 CRC 值。CY8CMBR3xxx 控制器中实现的 CRC 算法是 CRC16-CCITT。

请按照下面流程，使用主机 API 进行计算 CRC 值：

1. 下载一个 C IDE。在图 5-29 所示的示例中，使用 C/C++ 的 eclipse IDE。
2. 创建一个 C 项目，并且项目中包含 CY8CMBR3xxx_APis.h 和 CY8CMBR3xxx_CRC.c 文件。这些文件是主机 API 的一部分。
3. 创建一个带有适合标题的新源文件。在图 5-29 所示的示例中，源文件名称为“main.c”。
4. 在 main.c 文件中，输入下面的代码（在图 5-29 中也显示了这些代码）。

```
#include <stdio.h>
```

```
/* Variable to store 126 bytes of configuration data */
unsigned char CY8CMBR3xxx_configuration[126]= { };
```

```
/* The main() function invokes the CY8CMBR3xxx_CalculateCrc API to compute the CRC for 126 bytes of
 * configuration data */
int main(void)
```

```

{
    /* Variable to store the 16-bit CRC value*/
    unsigned short crcValue;

    /* Call the "CY8CMBR3xxx_CalculateCrc" API with configuration data address as the argument*/
    crcValue = CY8CMBR3xxx_CalculateCrc(CY8CMBR3xxx_configuration);

    /* Print the calculated 16-bit CRC value*/
    printf("The CRC for the given configuration data is: 0x%x",crcValue);

    return 0;
}

```

5. 通过使用 BCP 从 CY8CMBR3xxx 控制器中读取 126 个配置数据字节，并将它们复制到一个文本编辑器软件内（如记事本），以便格式化数据，即删除所有来自复制阵列中的“+”符号，并在每个字节数据前使用“0x”字符代替它，如图 5-27 和图 5-28 中所示。

图 5-27. 从 BCP 复制到记事本内的 CY8CMBR3xxx 配置数据

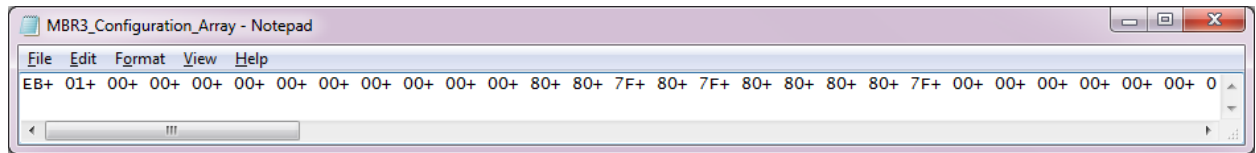
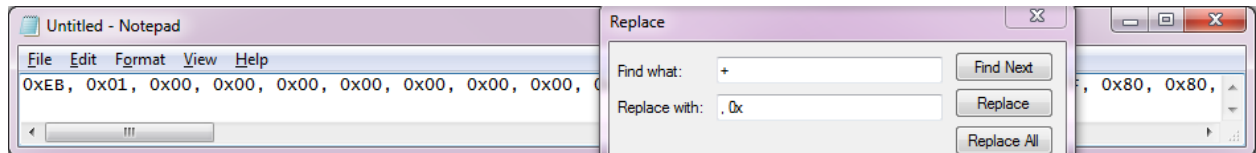
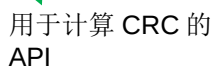


图 5-28. 在记事本中被格式化的 CY8CMBR3xxx 配置数据



6. 将该格式化的阵列复制到 main.c 中的“CY8CMBR3xxx_configuration [126]”阵列，如图 5-28 所示。
7. 保存并编译项目。请确保没有任何错误。
8. 成功编译项目后，请点击“Run”按键以获得 CRC 值。

图 5-29.使用 Eclipse C/C++ IDE 中的主机 API 计算 CRC 值



6. CapSense 功能调试



如 [CapSense 技术](#) 章节中解释，赛普拉斯的 SmartSense 自动调试算法通过自动调试硬件和固件参数来消除手动调试。然而，SmartSense 算法要求设置传感器的灵敏度参数值要确保所有传感器的信噪比大于 5:1。该 CY8CMBR3xxx 允许手动设置阈值参数。本节显示如何设置传感器的灵敏度和阈值参数，以检测到可靠触摸。CY8CMBR3002 CapSense 控制器不支持手动调试。在该控制器中，SmartSense 算法自动对所有硬件和固件参数进行调试。因此，下面各节不适用于 CY8CMBR3002 控制器。

6.1 通用注意事项

6.1.1 信号、噪声和信噪比

调试较好的 CapSense 系统能可靠地识别传感器的开关状态。为了获取强大性能，CapSense 信号必须远远大于 CapSense 噪声。可以使用一个称为信噪比（SNR）的数值，将 CapSense 信号与 CapSense 噪声进行比较。有关信噪比的更多信息，请参阅 [CapSense 入门](#) 中“CapSense 技术”的内容。根据 CapSense 最佳实践，需要所有传感器的最小信噪比为 5:1。传感器的信噪比的公式如下：

$$\text{SNR} = \text{公式 4} \frac{\text{Signal}}{\text{Peak-to-Peak Noise}}$$

其中：信号 = 原始信号 - 基线

峰峰值噪声 = 通过 3000 个样本测量原始信号的峰峰值噪声

因此，要获得高 SNR 的关键事项是增加信号并降低噪声。

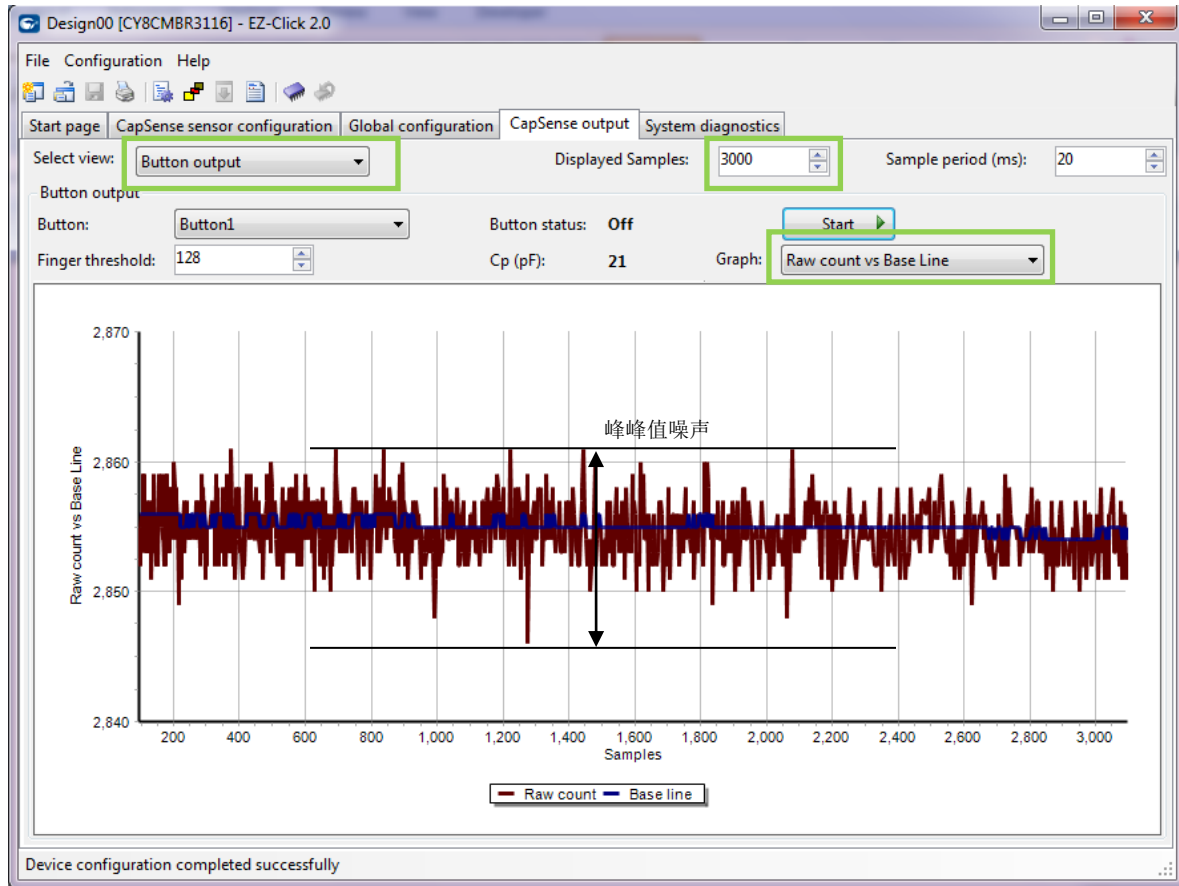
6.1.1.1 测量噪声信号

为了计算 SNR，需要测量该传感器的信号和噪声信号。

要使用 [EZ-Click](#) 测量噪声信号，请按照以下各步骤：

1. 生成配置文件并向器件下载配置。
2. 进入“CapSense output”选项卡并选择“Raw count vs Baseline”（原始信号与基线）图形。将“Displayed Samples”字段设置为 3000。
3. 点击“Start”按键，并等到获取 3000 个样本。
注意： 在获取原始信号样本过程中，不应触摸传感器，以测量噪声信号。
4. 测量噪声信号，该噪声信号等于原始信号（最大值）与原始信号（最小值）之差，如图 6-1 所示。图 6-1 中图形的噪声信号为 $2860 - 2845 = 15$ 个计数。

图 6-1. 在 EZ-Click 定制器中测量噪声



6.1.1.2 信号测量

测量噪声信号后，需要测量传感器的信号，以计算 SNR。

请按照以下步骤测量信号：

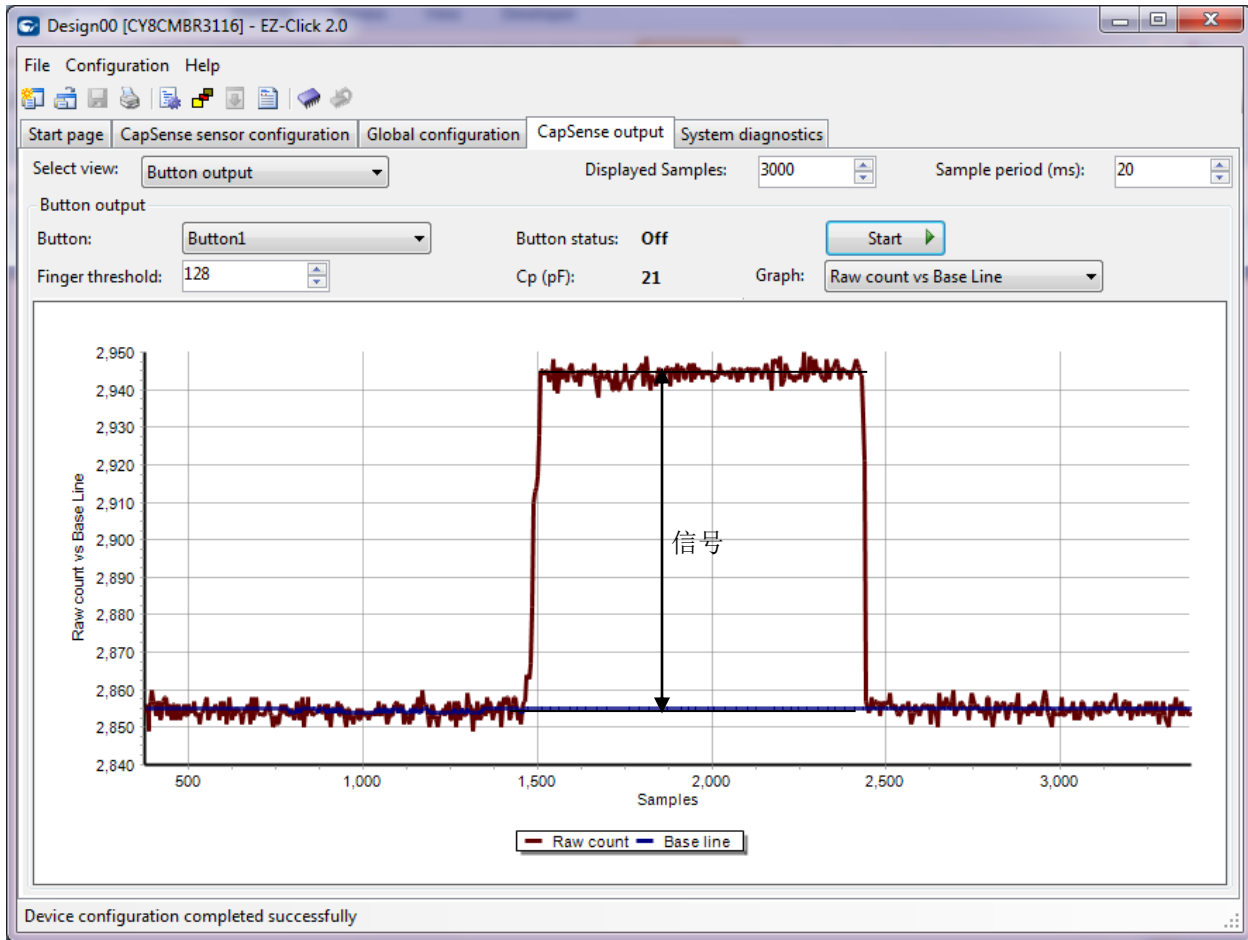
1. 在“CapSense output”选项卡中，点击“Start”按键，以查看“Raw count vs Baseline”图形。
2. 触摸传感器以获取 1000 个样本，然后释放手指。
3. 测量原始信号从无触摸到触摸状态的变化，如图 6-2 所示。

信号 = 原始信号（当触摸传感器时）的平均值 - 原始信号（当不触摸传感器时）的平均值

对于图 6-2 中所示的图形，信号 = 2945 - 2855 = 90 个计数

$$\text{SNR} = \frac{90}{15} = 6:1$$

图 6-2. 在 EZ-Click 定制器中测量信号



6.2 调试按键、滑条和保护传感器

图 6-3 显示的是调试按键、滑条和保护传感器的典型流程。

CY8CMBR3xxx 控制器中传感器的调试被分为两个部分：

1. 确保 SNR 大于 5:1
2. 将阈值参数设置为推荐值

6.2.1 确保最低信噪比的方法

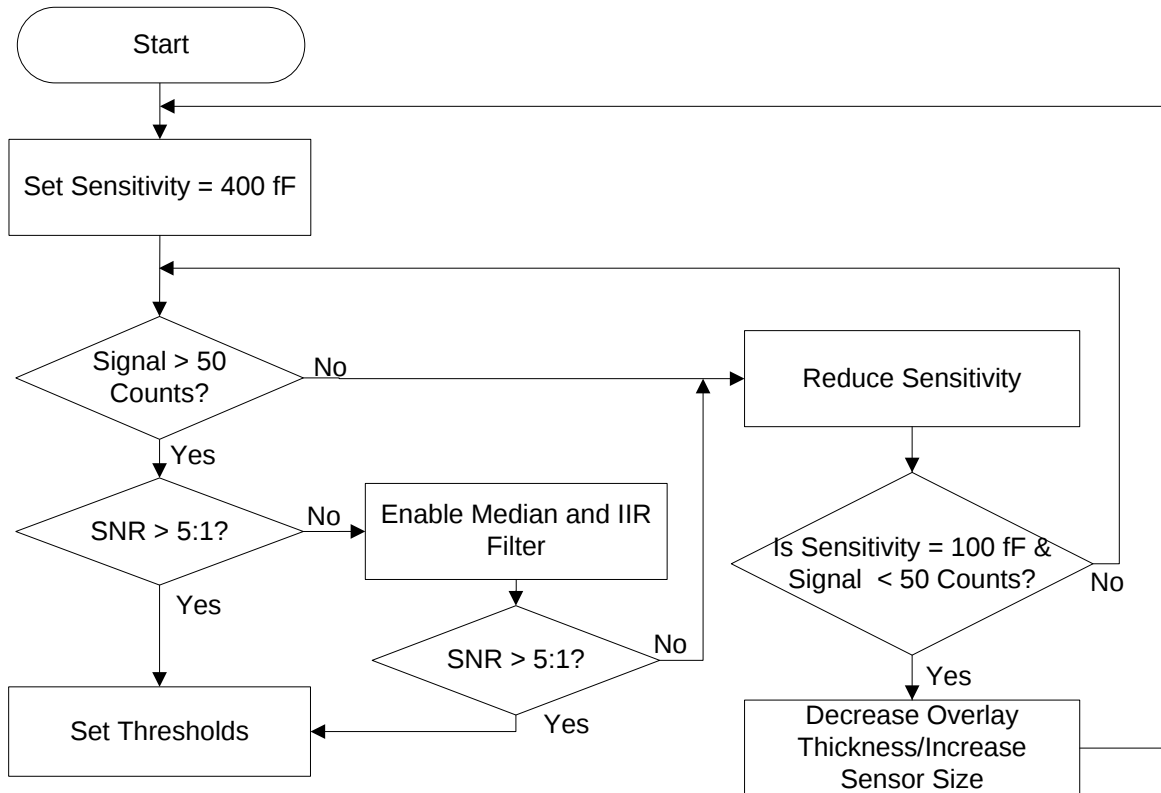
为了确保 SNR 大于 5:1，请按照以下步骤：

1. 将传感器的灵敏度设置为 400 fF，并测量信号。
2. 如果信号小于 50 个计数，降低灵敏度值，直到信号大于 50 个计数为止。如果设置最低灵敏度值（100 fF）后信号还不大 50 个计数，则降低覆盖层厚度或增加传感器大小。
3. 当信号大于 50 个计数时，测量噪声信号。如果噪声大于信号的 1/5，使能滤波器以降低噪声。使能 IIR 滤波器，并测量噪声信号。如果噪声仍大于信号的 1/5，使能中值滤波器。如果使能了中值滤波器，要禁用自动阈值性能并手动设置手指阈值参数。
4. 计算 SNR，并如果 SNR 大于 5:1，进行设置阈值参数。

5. 即使使能滤波器后，SNR 仍不大于 5:1，则降低敏感度值和计算 SNR。如果设置最低灵敏度值（100 fF）和使能滤波器后，SNR 还低于 5:1，则降低覆盖层厚度或增加传感器大小，以得到大于 5:1 的 SNR。

注意： 对于滑条，信号应始终小于 255，以得到准确的中心位置。如果信号大于 255，则增加敏感度参数，直到信号小于 255 为止。

图 6-3.按键、滑条和保护传感器调试



6.2.2 按键阈值参数

确保所有传感器的 SNR 大于 5:1 后，可以设置阈值参数。

以下阈值参数适用于按键和保护传感器：

- 手指阈值
- 迟滞
- 噪声阈值
- 负噪声阈值
- 低基线复位

6.2.3 一节介绍了何时以及如何设置这些参数。

6.2.2.1 手指阈值

该手指阈值参数用于判断传感器是否处于活动/非活动状态。如果传感器的差值大于手指阈值和迟滞之和，则传感器被判断为“ON”，并如果传感器的差值小于手指阈值和迟滞之差，则传感器被判断为“OFF”。手指阈值的范围为31到200；默认设置为128。关于设置该参数，请参考表6-2。

手指阈值和传感器“ON/OFF”状态之间的关系如下：

$$\text{Sensor State} = \begin{cases} \text{ON} & \text{if } (\text{Signal} \geq \text{Finger Threshold} + \text{Hysteresis}) \\ \text{OFF} & \text{if } (\text{Signal} \leq \text{Finger Threshold} - \text{Hysteresis}) \end{cases} \quad \text{公式5}$$

6.2.2.2 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“ON/OFF”状态。迟滞提供对传感器状态噪声切换的抗干扰能力。该参数仅适用于按键和保护传感器。该参数的范围为0到31；默认设置为12。关于设置该参数，请参考表6-2。

6.2.2.3 噪声阈值

噪声阈值参数设置原始信号限制，该限制位于不对基线进行更新的区域，如图6-4所示。换言之，只要原始信号高于（基线+ 噪声阈值），则基线保持不变。因此，当发生手指触摸时使该基线避免设置过高。该参数的范围为0到127；默认设置为51。关于设置该参数，请参考表6-2。

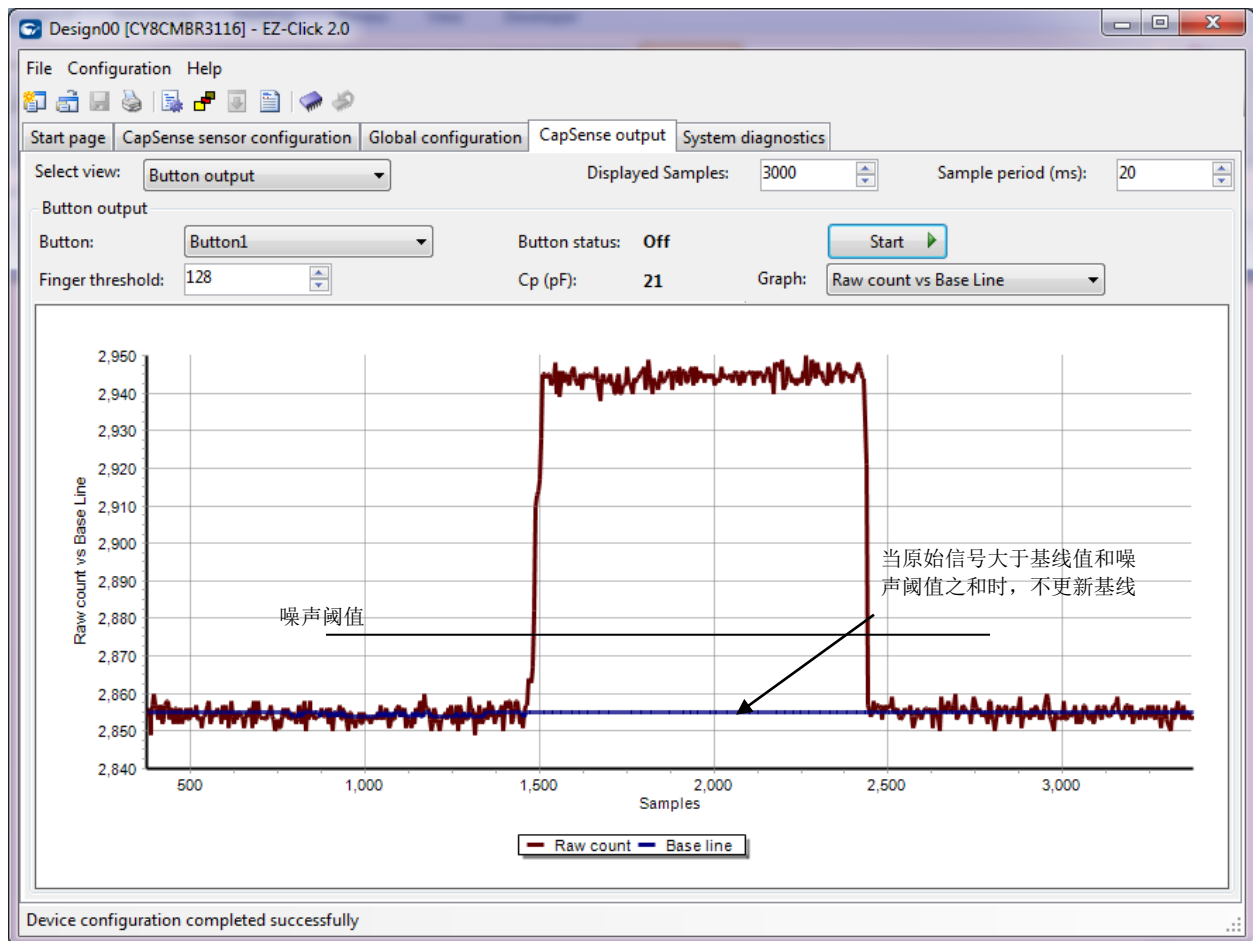
6.2.2.4 负噪声阈值

对于由低基线复位参数指定的采样数而言，如果原始信号小于基线值与负噪声阈值之差，则将基线值设为新的原始信号值。基线中的这种变化复位任何器件上电时卡处于活动状态的传感器，即控制器上电时传感器被触摸。该参数的范围为0到127；默认设置为51。关于设置该参数，请参考表6-2。

6.2.2.5 低基线复位

当基线值高于原始信号时，该参数和负噪声阈值参数用于复位基线。该参数计算复位基线所需的异常低的原始信号。如果器件启动过程中手指放在传感器上，然后抬起手指，则该参数用于复位基线。该参数的范围为0到127；默认设置为50。关于设置该参数，请参考表6-2。

图 6-4.噪声阈值



6.2.3 设置按键、滑条和保护传感器的阈值

表 6-1 显示的是适用于按键、滑条和保护传感器的阈值参数。

CY8CMBR3xxx 提供按键的自动阈值功能。应始终使能该功能，除非用户要求手动设置手指阈值。

滑条传感器需要用户手动设置手指阈值参数，因为自动阈值功能并不适用于滑条传感器。SmartSense 自动调试算法将自动设置所有其他阈值参数。

为了手动设置阈值参数（比如：噪声阈值、负噪声阈值和低基线复位），在相应的阈值寄存器中设置相应的阈值覆盖位，并手动设置这些阈值。建议不覆盖各阈值参数。使用桥接器控制屏幕（BCP）工具来设置阈值覆盖位并设置阈值参数（如：噪声阈值、负噪声阈值和低基线复位）。EZ-click 不提供用于覆盖或设置阈值参数的选项。

表 6-1. 按键、滑条和保护传感器的阈值参数

阈值参数	传感器类型		
	按键	滑条	保护
手指阈值	仅当禁用自动阈值功能时，才可设置	应始终手动设置	仅当禁用自动阈值功能时，才可设置
噪声阈值	仅当使能噪声阈值覆盖位时，才可设置	仅当使能滑条的噪声阈值覆盖位时，才可设置	仅当使能噪声阈值覆盖位时，才可设置
负噪声阈值	仅当使能负噪声阈值覆盖位时，才可设置	仅当使能滑条负噪声阈值覆盖位时，才可设置	仅当使能噪声阈值覆盖位时，才可设置
迟滞	仅当使能迟滞阈值覆盖位时，才可设置	不适用	仅当使能迟滞阈值覆盖位时，才可设置
低基线复位	仅当使能低基线复位覆盖位时，才可设置	仅当使能滑条低基线复位覆盖位时，才可设置	仅当使能低基线复位覆盖位时，才可设置

注意： CY8CMBR3xxx 具有用于按键和滑条传感器的单独寄存器，以设置噪声阈值、负噪声阈值和低基线复位参数。

请按照以下流程设置这些阈值参数：

1. 在“CapSense output”选项卡中，选择“Diff count vs. Finger threshold”图形并点击“Start”按键。
2. 触摸传感器并测量差值，如图 6-5 所示。

注意：

- 对 CY8CMBR3xxx 控制器的差值进行规范化处理。该值不会等于原始信号（触摸传感器时）和基线之差。
- 如果差值大于 255，将截断它为 255。在这种情况下，不可正确设置各阈值参数。增加灵敏度参数值，直到差值小于 255 个计数为止。按照 6.2.1 一节中介绍的流程，确保改变灵敏度参数值后 SNR 大于 5:1

3. 将阈值参数设置为如表 6-2 所示的值：

表 6-2. 阈值参数值

阈值参数	建议
手指阈值	差值的 80%
噪声阈值	差值的 40%
负噪声阈值	差值的 40%
迟滞	差值的 10%
低基线复位	设为 50

对于图 6-5 中所示的示例，差值约为 210 个计数，并设置阈值参数，如以下所示。

手指阈值 = $80\% \times 210 = 168$

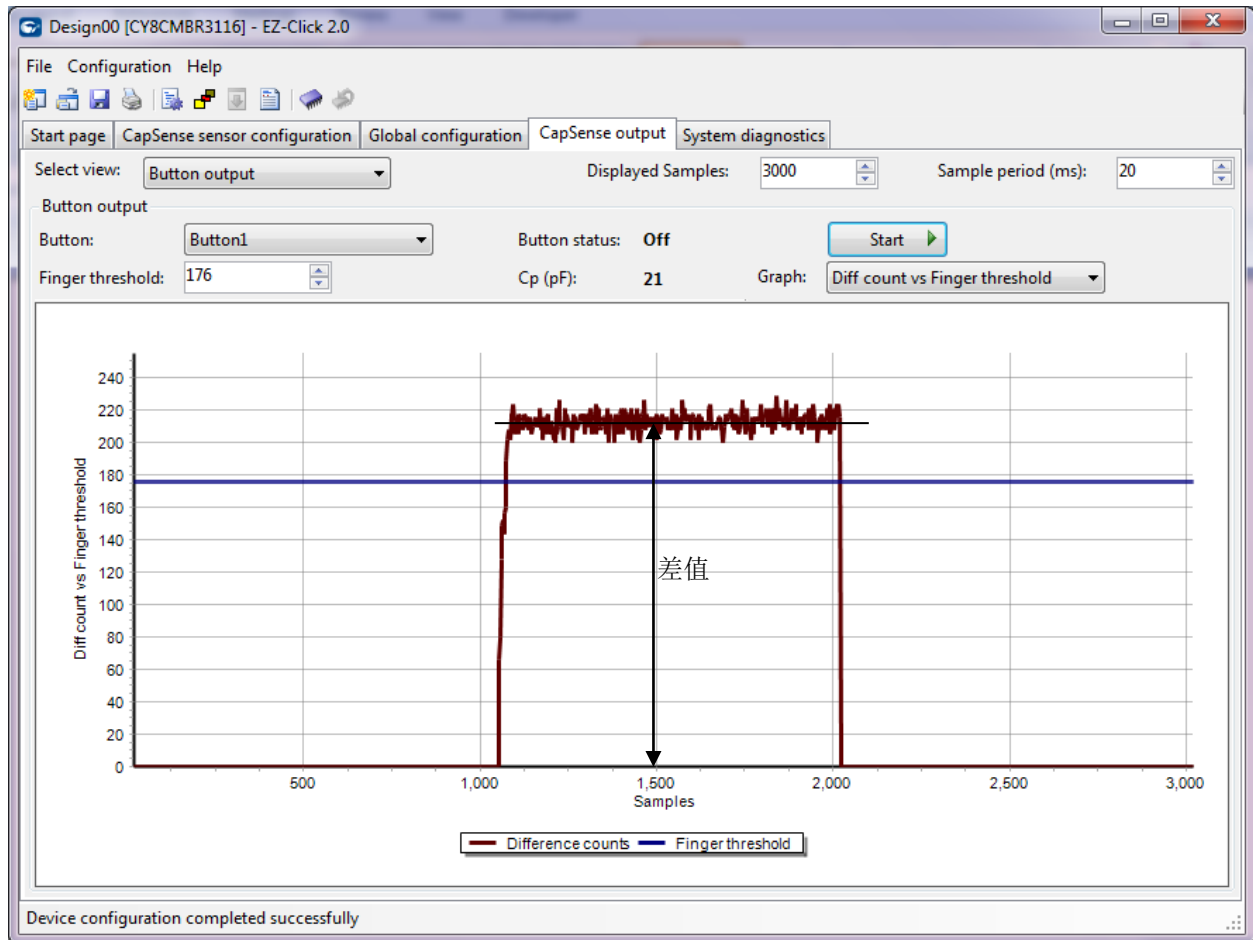
噪声阈值 = $40\% \times 210 = 84$

负噪声阈值 = 噪声阈值 = 84

迟滞 = $10\% \times 210 = 21$

低基线复位 = 50

图 6-5. 差值测量



6.3 调试接近传感器

图 6-6 显示的是对接近传感器进行调试的典型流程。

在 CY8CMBR3xxx 控制器内，通过两个阶段调试接近传感器：

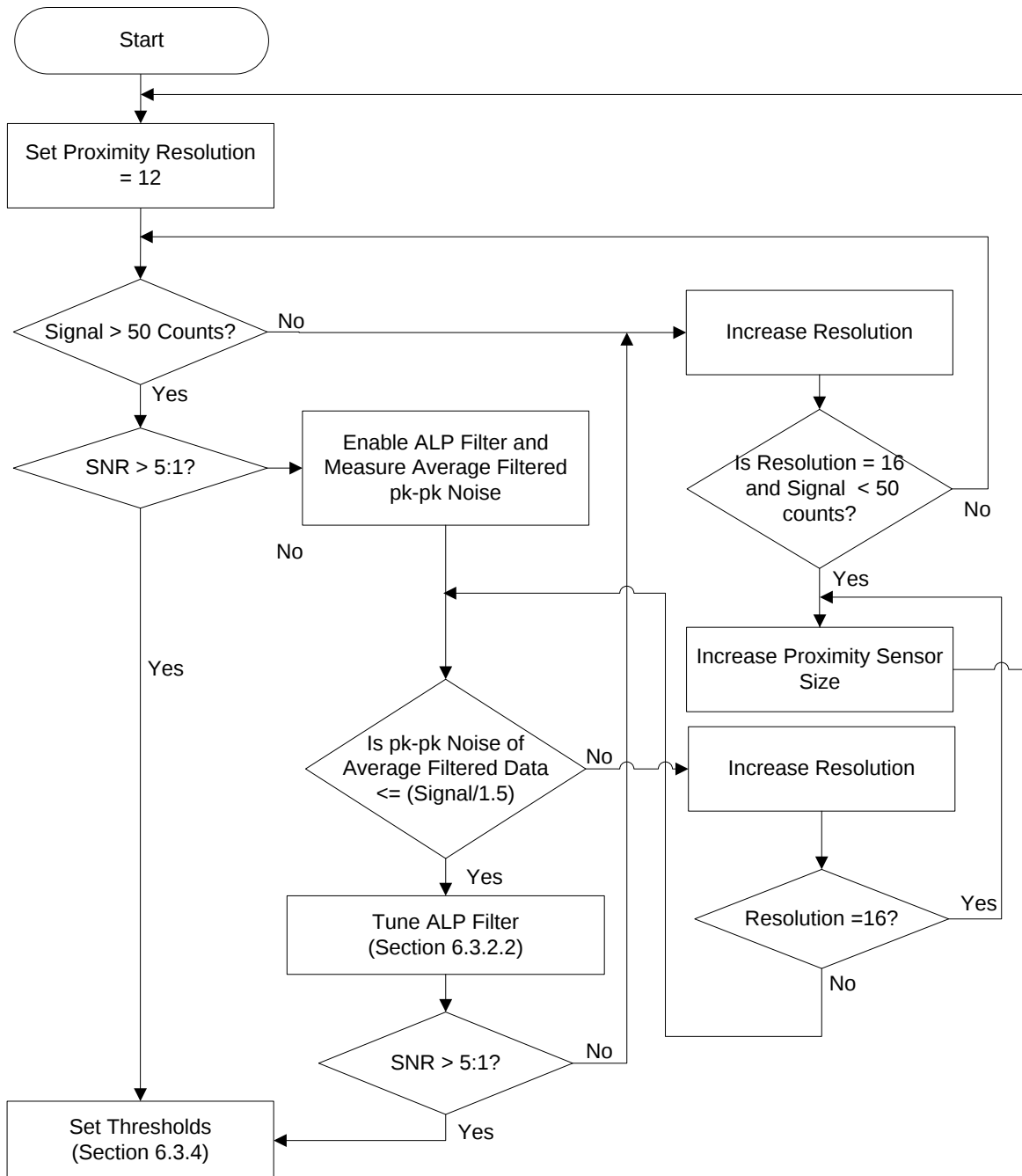
1. 确保 SNR 大于 5:1
2. 设置适当的阈值参数

6.3.1 确保最低信噪比的方法

为了确保 SNR 大于 5:1，请按照以下的步骤：

1. 设置接近传感器的分辨率为 12 位，并在所要求的接近距离测量信号。
2. 如果信号小于 50 个计数，增加接近分辨率，直到信号大于 50 个计数为止。即使设置分辨率为最高值（即 16 位）后信号仍不大于 50 个计数，则增加接近传感器大小。
3. 信号大于 50 个计数后，测量噪声信号，如图 6-7 所示。如果噪声大于信号的 1/5，使能 ALP 滤波器以降低噪声。
注意：中值滤波器和 IIR 滤波器不适用于接近传感器。
4. 计算 SNR，并如果 SNR 大于 5:1，继续设置阈值参数。
5. 即使使能 ALP 滤波器后，SNR 仍不大于 5:1，则增加分辨率并计算 SNR。即使设置分辨率为最高值（16 位）并使能 ALP 滤波器后，SNR 还低于 5:1，则增加接近传感器大小以获得大于 5:1 的 SNR。

图 6-6.接近传感器调试



6.3.2 高级低通（ALP）滤波器

ALP 滤波器由多个低通滤波器组合而成，特别设计它用于衰减接近传感器中的噪声。ALP 滤波器在多个低通滤波器之间切换，以获得最大的噪声衰减和提供最佳的响应时间。设置表 6-3 中所示的 ALP 参数，以进行正常操作。

表 6-3.ALP 滤波器参数

参数	建议
K 值	当 ALP 滤波器被使能时，设置该参数。
接近正阈值	当 ALP 滤波器被使能时，设置该参数。
接近负阈值	当 ALP 滤波器被使能时，设置该参数。

6.3.2.1 ALP 滤波器参数

6.3.2.1.1 K 值

ALP 滤波器的 K 值用于确定接近传感器原始信号中的噪声衰减。CY8CMBR3xxx 控制器有三个不同的 K 值：1/16、1/32 和 1/64。

噪声按照下面序列而降低： $1/64 > 1/32 > 1/16$ 。关于设置该参数的信息，请参考 [ALP 滤波器调试](#) 一节的内容。

6.3.2.1.2 接近正阈值

该参数用于确定接近传感器的开启时间，它的取值范围为 0 到 255，默认值为 30。关于设置该参数的信息，请参考 [ALP 滤波器调试](#) 一节的内容。

6.3.2.1.3 接近负阈值

该参数用于确定接近传感器的关闭时间，它的取值范围为 0 到 255，默认值为 30。关于设置该参数的信息，请参考 [ALP 滤波器调试](#) 一节的内容。

6.3.2.2 ALP 滤波器调试

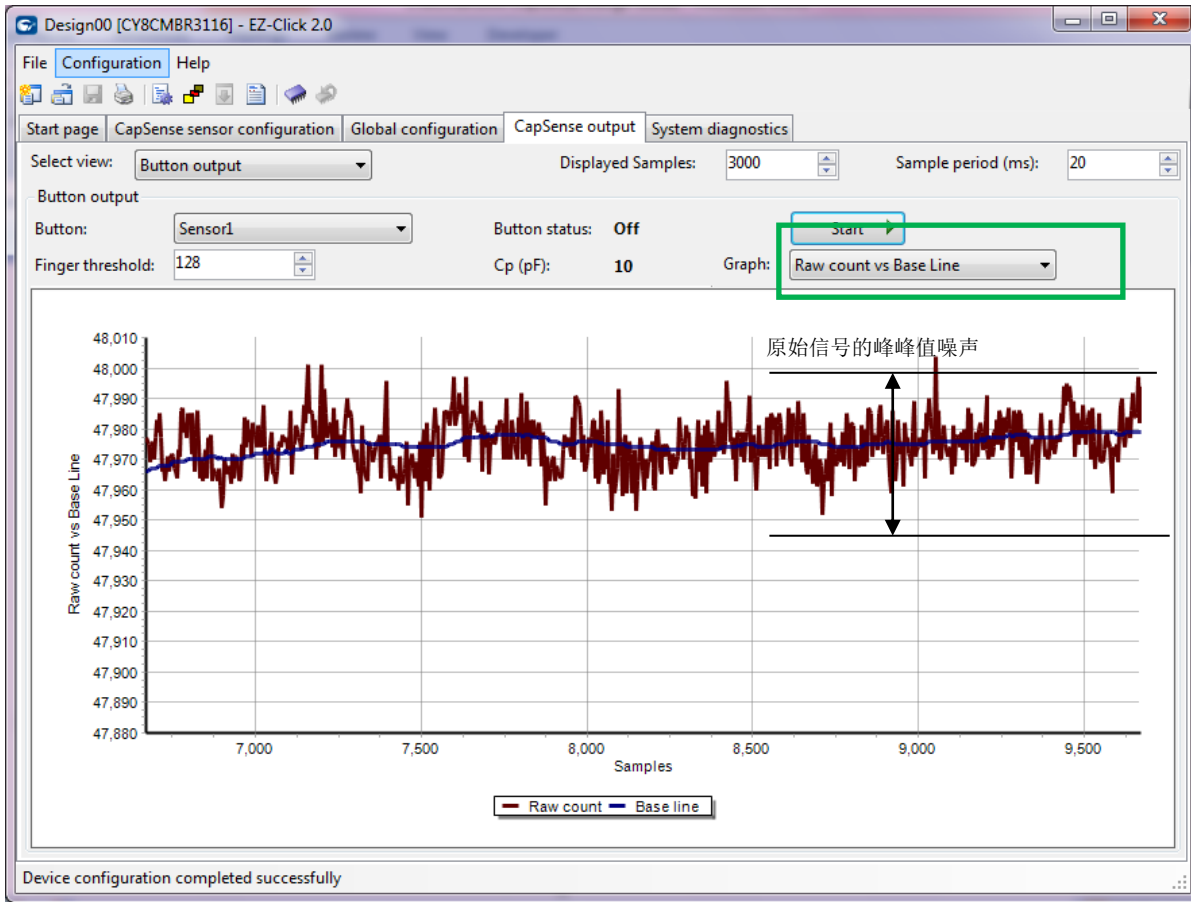
按照以下步骤调试 ALP 滤波器：

1. 禁用 ALP 滤波器，并在 3000 个样本中对原始信号的峰峰值噪声进行测量，如图 6-7 所示。有关更多信息，请参考 [测量噪声信号](#) 一节的内容。
2. 根据表 6-4 的映射情况，设置 EZ-Click 的“CapSense sensor configuration”选项卡中的 K 值。

表 6-4.选择 K 值

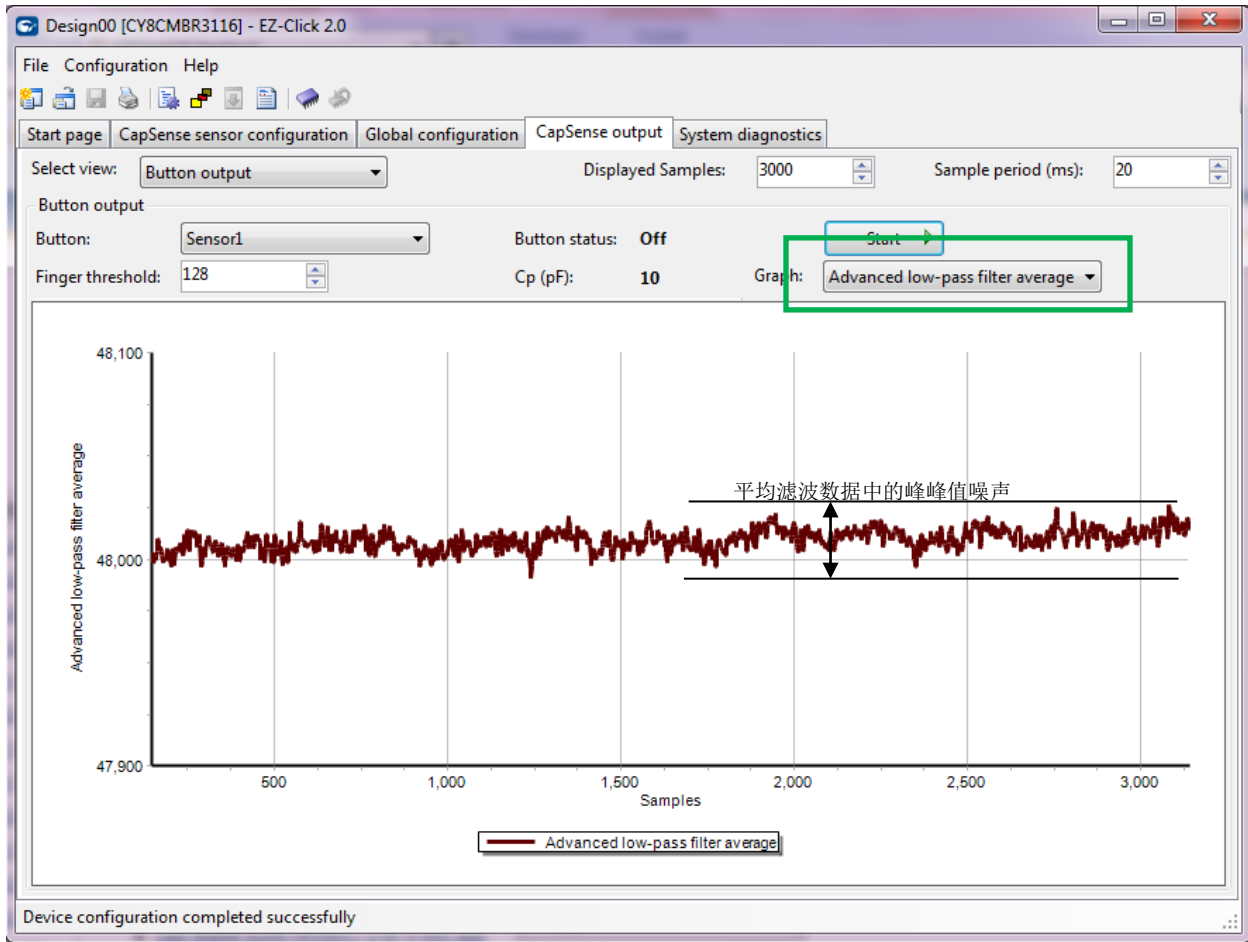
峰峰值噪声	推荐的 K 值
小于 32 个计数	1/16
大于 32 个计数	1/32
大于 64 个计数	1/64

图 6-7. 当禁用 ALP 滤波器时测量接近传感器的噪声信号



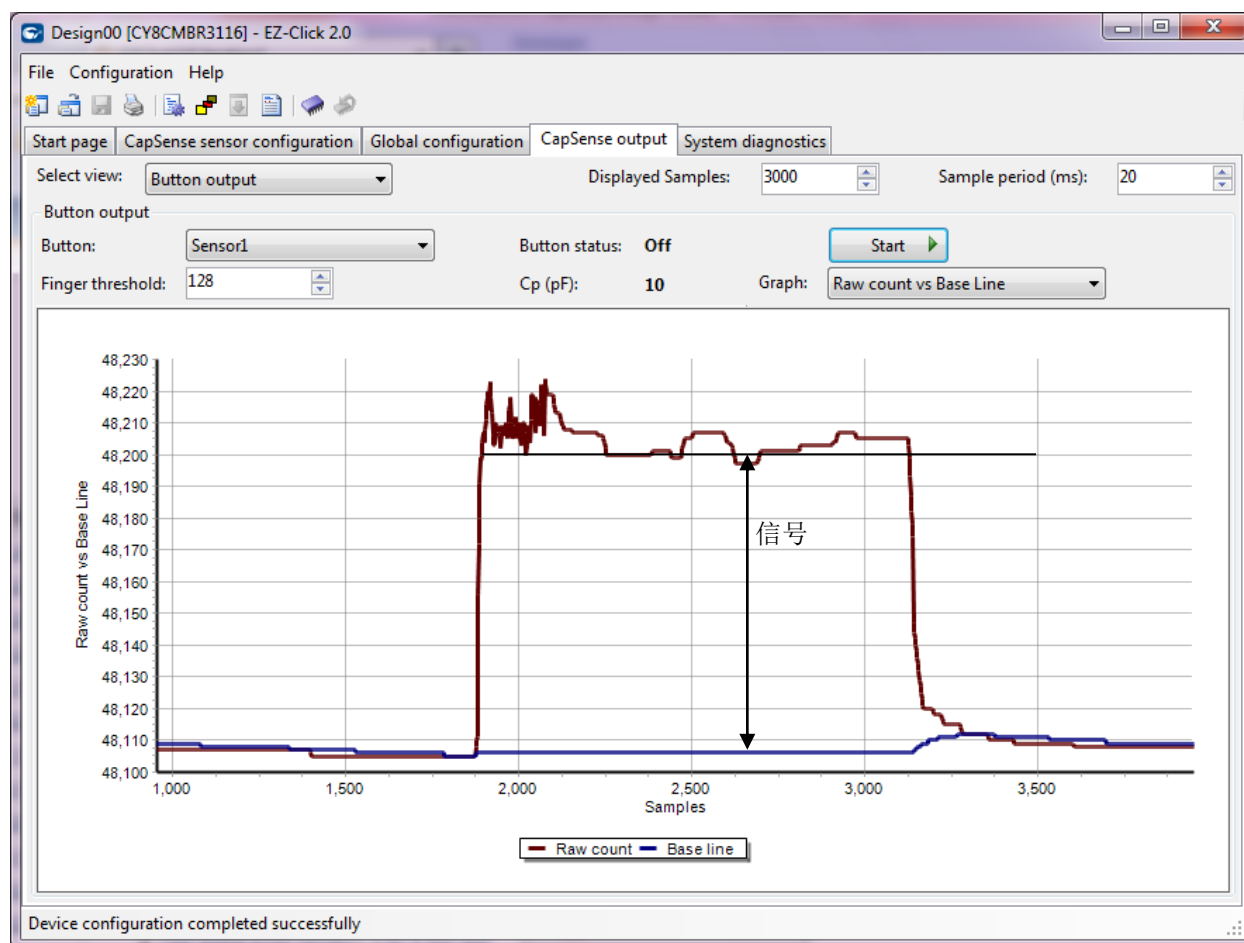
3. 使能 ALP 滤波器（如果设置 K 值，EZ-Click 自动使能 ALP 滤波器），并在 3000 个样本中通过高级低通平均滤波数据对峰峰值噪声进行测量，如图 6-8 所示。平均滤波数据可以查看在 EZ-Click 的“CapSense output”选项卡中。

图 6-8. 平均滤波数据中的峰峰值噪声



4. 设置接近正阈值等于 $1.5 \times$ 平均滤波数据的峰峰值噪声。
5. 设置接近负阈值等于 $0.5 \times$ 平均滤波数据的峰峰值噪声。
6. 设置手指阈值等于接近正阈值。这是必需的，以确定 ALP 滤波器的响应。
7. 使用上述所示的设置配置器件，并测量原始信号的噪声。
8. 将手放在所需的接近距离内，并测量信号，如图 6-9 所示。
9. 计算 SNR。如 SNR 大于 5:1，则进行设置阈值，如表 6-5 所示。否则，增加接近分辨率并开始重复上述步骤。

图 6-9.测量接近传感器信号



6.3.3 接近传感器阈值参数

表 6-5 列出了适用于接近传感器的各阈值参数。

表 6-5.接近传感器阈值

参数	建议
接近阈值	应始终手动设置
接近触摸阈值	应始终手动设置
接近迟滞	当使能迟滞覆盖时，设置该参数
接近噪声阈值	当使能噪声阈值覆盖时，设置该参数
接近负噪声阈值	当使能负噪声阈值覆盖时，设置该参数
接近低基线复位	当使能低基线复位覆盖时，设置该参数

6.3.3.1 接近阈值

接近阈值参数类似于按键的手指阈值。它用于确定接近传感器的“ON/OFF”状态。如果差值大于接近阈值和迟滞之和，接近传感器的状态为“ON”。如果差值小于接近阈值和迟滞之差，接近传感器的状态为“OFF”。应手动设置该阈值。该参数的范围为 31 到 200；默认值为 128。有关推荐值的详细信息，请参考表 6-6。

6.3.3.2 接近触摸阈值

接近传感器有两个事件：接近检测事件和触摸检测事件（当触摸接近传感器时发生的）。接近阈值参数用于确定接近传感器事件的“ON/OFF”状态；接近触摸阈值用于检测接近传感器的触摸事件。该参数的范围为 62 到 65000；默认值为 512。有关推荐值的详细信息，请参考表 6-6。

6.3.3.3 接近迟滞

接近迟滞参数类似于按键迟滞。它与接近阈值参数一起使用，用于确定接近传感器的“ON/OFF”状态。该参数的范围为 0 到 127；默认值为 5。有关推荐值的更多信息，请参考表 6-6。

6.3.3.4 接近噪声阈值

对于按键，接近噪声阈值参数类似于噪声阈值参数。该参数的范围为 0 到 127；默认设置为 20。有关推荐值的详细信息，请参考表 6-6。

6.3.3.5 接近负噪声阈值

对于按键，接近负噪声阈值参数类似于负噪声阈值参数。该参数的范围为 0 到 127；默认值为 20。有关推荐值的详细信息，请参考表 6-6。

6.3.3.6 接近低基线复位

对于按键，接近低基线复位参数类似于低基线复位参数。该参数的范围为 0 到 127；默认值为 50。有关推荐值的详细信息，请参考表 6-6。

6.3.4 设置接近传感器阈值参数

对于接近传感器，只要设置接近阈值和接近触摸阈值。SmartSense 算法用于内部计算所有其他阈值。

如果由于某种原因，需要覆盖这些参数，则使用 BCP 发送 I²C 指令来实现。欲了解更多信息，请参考 [CY8CMBR3xxx 寄存器技术参考手册](#)。强烈建议不覆盖各阈值参数。

请按照以下流程设置接近传感器的阈值参数：

1. 在“CapSense output”选项卡中，为接近传感器选择“Diff count vs. Finger threshold”图形并点击“Start”按钮。
2. 将手放在所需的接近距离内，并测量差值，如图 6-10 所示。

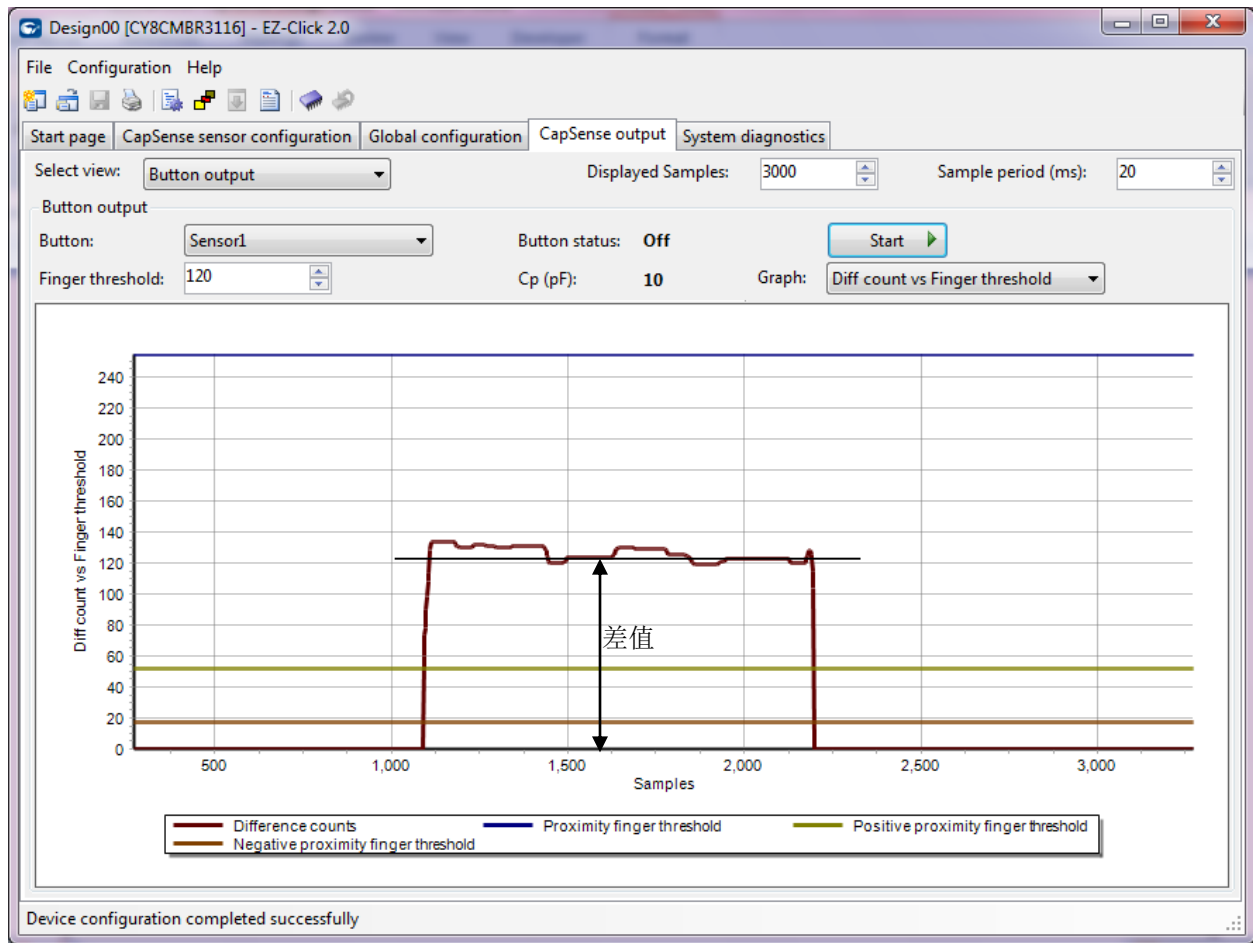
注意：

- 对 CY8CMBR3xxx 控制器的差值进行规范化处理。该值不会等于原始信号（触摸传感器时进行测量）和基线之差。
 - 如果差值大于 255，将截断它为 255。在这种情况下，不可正确设置各阈值参数。增加灵敏度参数值，直到差值小于 255 个计数为止。请确保更改灵敏度参数值后 SNR 大于 5:1。
3. 将阈值参数设置为如表 6-6 所示的值。
 4. 要设置接近触摸阈值，则触摸接近传感器，然后测量差值（接近传感器被触摸时测量）。
 5. 设置接近触摸阈值等于差值（接近传感器被触摸时测量）的 80%。

表 6-6.接近阈值设置

阈值参数	建议
接近阈值	差值的 80%
接近触摸阈值	差值的 80%（当接近传感器被触摸时）
接近噪声阈值	差值的 40%
接近负噪声阈值	差值的 40%
接近迟滞	差值的 10%
接近低基线复位	设为 50

图 6-10.测量差值以设置接近传感器阈值



7. 低功耗设计中的注意事项



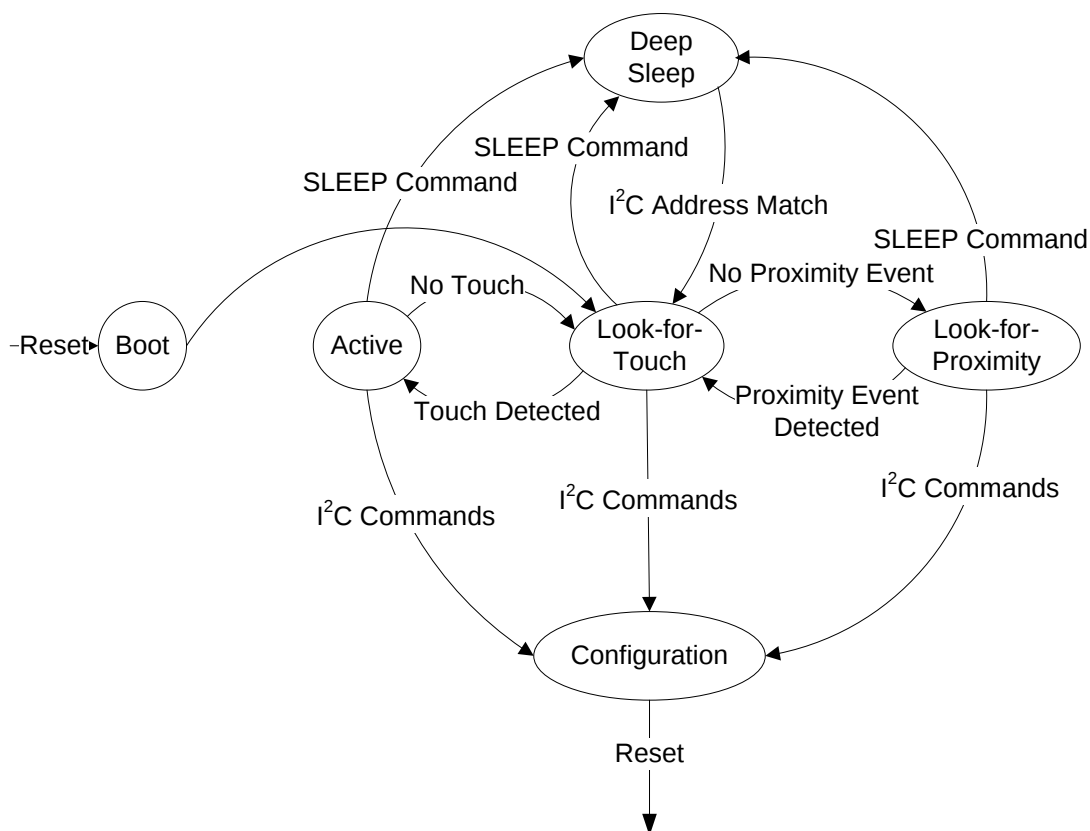
电容式触摸感应被用于电池供电、手持设备和便携式电子设备。因为这些设备是电池供电，所以功耗与能源效率变得非常重要的。本节介绍了对 CY8CMBR3xxx 控制器功耗的影响因素，并提供如何减少功耗的指南。

7.1 CY8CMBR3xxx 的工作模式

CY8CMBR3xxx 控制器支持以下五个工作模式：

1. 活动模式
2. 查找触摸（Look-for-touch）模式
3. 查找接近（Look-for-Proximity）模式
4. 深度睡眠模式
5. 配置模式

图 7-1.CY8CMBR3xxx 的工作模式



7.1.1 活动模式

CY8CMBR3xxx 控制器从复位模式进入活动模式，如图 7-1 所示。在活动模式下，控制器会扫描所有传感器、更新传感器数据，并根据传感器的状态，驱动任何已配置的输出（GPO、蜂鸣器和主机中断）。默认情况下，将活动模式中的扫描速率设置为 20 ms，以确保快速触摸响应。如果在 20 ms 内完成对传感器的扫描和处理功能，使控制器在持续时间（20 ms – 活动时间）内进入睡眠模式。当 20 ms 的定时器到期时，再次扫描传感器，并周期继续实现，如图 7-2 所示。

如果扫描所有传感器和处理传感器数据所需的时间超过 20 ms，则不将该器件进入睡眠模式。反而，对下一个周期开始进行扫描，如图 7-3 所示。

图 7-2.活动时间小于 20 ms 时活动模式下的扫描睡眠序列

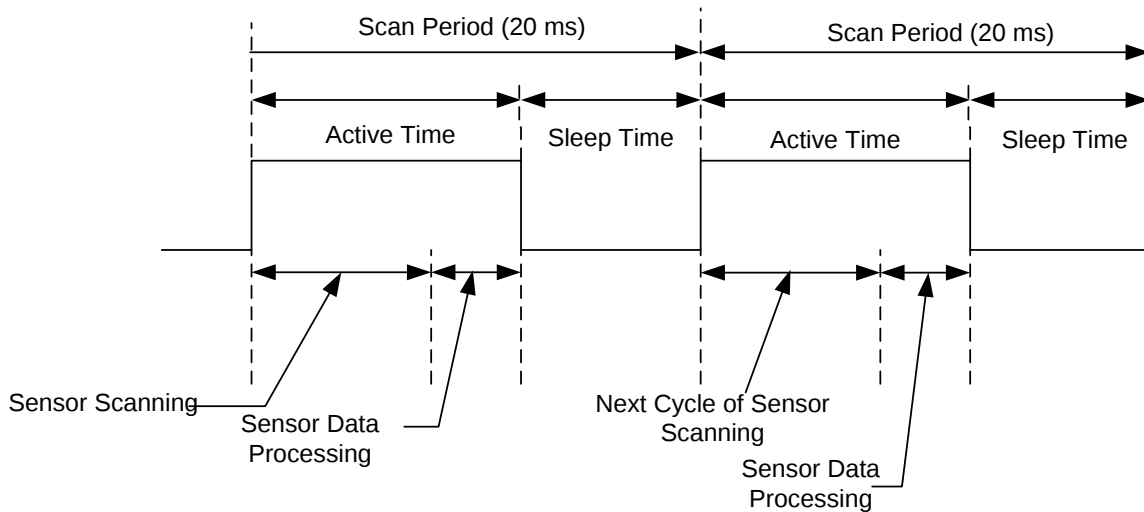
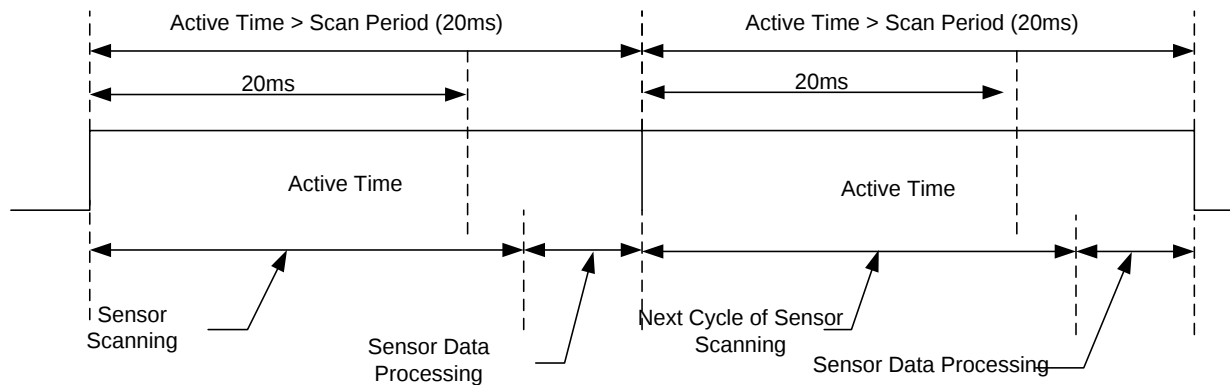


图 7-3.活动时间大于 20 ms 时活动模式下的扫描睡眠序列



7.1.2 查找触摸（Look-for-touch）模式

在等于状态超时值的期间内，如果在活动模式下检测不到任何触摸，CY8CMBR3xxx 会进入查找触摸模式，如图 7-1 所示。除非扫描周期是可配置的之外，查找触摸模式类似于活动模式。扫描周期的默认值为 120 ms。在这种模式下，扫描所有传感器，并处理传感器数据。如果任何传感器处于“ON”状态，输出（GPO、蜂鸣器和主机中断）被驱动，并控制器进入活动模式。

7.1.3 查找接近（Look-for-Proximity）模式

如果设计有接近传感器和“接近触摸时唤醒”功能被使能，则查找接近模式才可用。在大于状态超时值的期间内，如果在查找触摸模式下检测不到任何触摸，控制器进入查找接近模式，如图 7-1 所示。

在查找接近模式下，控制器只扫描接近传感器。如果检测到接近事件，则控制器进入查找触摸模式。如果在查找接近模式下检测到触摸事件，则控制器进入或模式。这种模式下的扫描周期是可配置的，并它等于查找触摸模式下的扫描周期；默认值为 120 ms。

7.1.4 深度睡眠模式

深度睡眠模式是最低功耗的模式。当主机发送睡眠指令时，CY8CMBR3xxx 器件进入深度睡眠模式。欲了解更多信息，请参考 [CY8CMBR3xxx 寄存器技术参考手册](#) 中介绍的 CTRL_CMD 寄存器的内容。在深度睡眠模式下，器件不扫描任何传感器，并禁用所有输出。当 I²C 地址匹配时，器件会唤醒；否认第一个 I²C 指令，然后进入查找触摸模式。

7.1.5 配置模式

当将 CMD_OP_CODE = ‘2’ 写入 CTRL_CMD 寄存器内，以将配置数据保存到非易失性存储器内时，器件进入配置模式。欲了解更多信息，请参考 [CY8CMBR3xxx 寄存器技术参考手册](#)。该器件只通过器件复位（软件或硬件复位）退出配置模式。

7.2 影响功耗的因素

影响到 CY8CMBR3xxx 控制器的功耗的各种因素：

- **传感器数量：**传感器数更多，则传感器的总扫描时间更长。对于一个固定的扫描周期，如果扫描时间增加，则器件处于睡眠模式的期间减少。这样导致该器件的功耗增加。
- **传感器的 C_P：**该值影响传感器的扫描时间。C_P 值更高，传感器扫描时间更长，导致功耗增加。按照 [CapSense 入门指南](#) 中所介绍的布局指南，使传感器 C_P 值最小。
- **灵敏度：**该参数影响传感器的扫描时间。传感器灵敏度参数值更低，传感器扫描时间更长。因此，灵敏度参数值更低，则功耗会更高。

功耗会按照下面灵敏度序列而降低：100 fF > 200 fF > 300 fF > 400 fF。

- **电磁兼容性（EMC）：**当使能 EMC 功能时，传感器扫描时间增加，导致功耗增加。
- **蜂鸣器的持续时间：**CY8CMBR3xxx 控制器处于活动状态，同时驱动蜂鸣器输出。更长的蜂鸣器持续时间导致活动时间增加和更高的功耗。
- **GPO 上的脉冲调制宽度（PWM）：**使能 GPO 上的脉冲调制宽度的影响与使能蜂鸣器输出的影响相同。如果在 GPO 引脚上 PWM 输出被使能，器件不进入睡眠模式，这样使功耗增加。
- **扫描周期：**扫描周期参数影响到器件处于睡眠模式时所需的期间。扫描周期更高，功耗将更低。
- **状态超时：**状态超时参数影响器件从活动模式进入查找触摸模式或从查找触摸模式进入查找接近模式时所需的时间。状态超时间隔更短，则功耗将更低。
- **I²C 通信：**如活动模式一节中解释，完成扫描传感器后 CY8CMBR3xxx 进入睡眠模式。如果对主处理器进行任何 I²C 数据传输，CY8CMBR3xxx 将不进入睡眠模式或它会退出深度睡眠模式。因此，频繁的 I²C 数据传输使平均功耗增加。

为了降低平均功耗，要实现一个专用的主机中断引脚，并开始进行 I²C 数据操作（仅当 CY8CMBR3xxx 发送一个主机中断脉冲时）。

根据 [估计平均电流消耗](#) 一节中介绍的设计输入，使用 [CY8CMBR3xxx 设计工具箱](#) 估计控制器的平均电流消耗。

7.3 使功耗降低的系统设计建议

赛普拉斯的 CY8CMBR3xxx 设计可满足电池供电应用的低功耗要求。请按照以下指南使功耗最小：

- 通过使用 [CapSense 入门](#)和 [CY8CMBR3xxx 设计工具箱](#)中所提的设计指南，使传感器 C_P 值最小。
- 设置传感器灵敏度的最佳值。请参考[敏感度控制](#)一节的内容。
- 根据需要使能 EMC。请参考[电磁兼容性（EMC）](#)一节的内容。
- 增加扫描周期，并降低状态超时周期。
- 使器件进入深度睡眠模式，或当不需要传感器扫描时使用电源管理器 IC（PMIC）来关闭输入器件的电源。
- 避免将 I²C 频繁读取/写入 CY8CMBR3xxx 控制器内，使用主机中断引脚，并当 CY8CMBR3xxx 发送一个主机中断脉冲后开始进行 I²C 数据传输。
- 使用接近传感器并使能“接近接触时唤醒”功能。
- 降低蜂鸣器的开启持续时间。
- 根据需要使能 GPO 上的 PWM。

8. 资源



8.1 网站

访问赛普拉斯 [CapSense 控制器](#) 网站可获取本文中讨论的所有参考资料。

在 CY8CMBR3xxx 网页上找到各种技术资源。

8.2 数据手册

[CY8CMBR3xxx 数据手册](#) 提供有关该器件的信息。

8.3 寄存器技术参考手册

[CY8CMBR3xxx 寄存器技术参考手册](#) 提供它的相关信息。

8.4 设计工具箱

交互式的 [设计工具箱](#) 使您能够设计一个既稳定又可靠的 CY8CMBR3xxx CapSense 解决方案。

8.5 EZ-Click™ 2.0

[EZ-Click 2.0](#) 是一个基于 GUI 的简单软件工具，用于定制 CY8CMBR3xxx 器件的配置。

8.6 开发套件

[CY3280-MBR3 评估套件](#) 展示 CY8CMBR3116 CapSense 控制器的功能。使用该套件可快速评估 CY8CMBR3xxx 解决方案的功能。

8.7 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对各种 CapSense 主题执行搜索。

[CapSense 应用笔记](#) — 以本文档提供的信息为基础的各种应用手册。

[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](#) — 请参见完整的 CapSense 产品线。

[视频资料库](#) — 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”网站，您可以找到可靠性和产品资质报告。

[技术支持](#) — 世界级在线技术支持。

9. 附录



9.1 顶层 API

表 9-1. 顶层主机 API

1	CY8CMBR3xxx_ReadSensorStatus	
	说明	该 API 将读取按键状态、被锁存的按键状态、接近状态、被锁存的接近状态、滑条 1 和 2 的位置以及从 CY8CMBR3xxx 器件抬起时滑条 1 和 2 的位置。 滑条位置和滑条抬起位置仅适用于 CY8CMBR3106S 器件。 器件触发主机中断脉冲时，您可以使用该 API 了解相应状态的变化。
	原型	<code>bool CY8CMBR3xxx_ReadSensorStatus(uint8 slaveAddress, CY8CMBR3XXX_SENSORSTATUS *status)</code>
	参数	slaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119
		*status 指向用户定义的缓冲区的指针，其中从 CY8CMBR3xxx 器件中读取的传感器状态得到存储。缓冲区结构被定义在 CY8CMBR3xxx_APIs.h 中。
	返回值	TRUE — 如果缓冲区被正确更新 FALSE — 如果缓冲区未被更新
2	CY8CMBR3xxx_WriteData	
	说明	该 API 会将一个或多个数据字节写入到 CY8CMBR3xxx 器件中的连续寄存器位置上。用户指定了最初的寄存器位置。
	原型	<code>bool CY8CMBR3xxx_WriteData(uint8 slaveAddress, uint8 *writeBuffer, uint8 numberOfBytes)</code>
	参数	SlaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119

		*writeBuffer 它是指将数据写入到器件内的缓冲区。 第一个元件应始终包含被写入的器件寄存器的位置。该值的范围为 0 – 134。 每个连续元件应包含需要写入到该寄存器和逐次寄存器的数据。这些元件值的范围为 0 – 255。数据字节数的范围为 1 到两个寄存器位置（128 和 135）中最小的那个值。
		numberOfBytes 被写入的字节数等于缓冲区中元件的数量。 有效范围：1 ~ 两个寄存器位置（129、136）中的最小值
	返回值	TRUE — 如果写操作成功 FALSE — 如果写操作不成功
3	CY8CMBR3xxx_ReadData	
	说明	该 API 会从 CY8CMBR3xxx 器件的连续寄存器位置中读取一个或多个数据字节。用户指定了寄存器最初的位置。
	原型	bool CY8CMBR3xxx_ReadData(uint8 slaveAddress, uint8 registerAddress, uint8 *readBuffer, uint8 numberOfBytes)
	参数	slaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119
		registerAddress 要读取的寄存器位置。被传输到该参数的值应是 CY8CMBR3xxx_Registers.h 文件中所定义的一个寄存器偏移宏。如果由 CY8CMBR3xxx_DEVICE 宏定义的器件支持寄存器宏，才会定义它。
		*readBuffer 使用从器件中读取的数据更新缓冲区。 后面每个元素都包含从相继寄存器中读取的数据。这些元件值的范围为 0 – 255。
		numberOfBytes 被读取的字节数等于缓冲区中元件的数量。 有效范围：1 – 252
	返回值	TRUE — 如果读操作成功 FALSE — 如果读操作不成功

4	CY8CMBR3xxx_WriteDualByte	
	说明	该 API 将数据写入到 CY8CMBR3xxx 器件的两个字节寄存器中。寄存器位置由用户指定。
	原型	bool CY8CMBR3xxx_WriteDualByte(uint8 slaveAddress, uint8 registerAddress, uint16 writeData)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		registerAddress 要写入的寄存器位置。传输到该参数的值应是 CY8CMBR3xxx_Registers.h 文件中定义的一个寄存器偏移宏。如果由 CY8CMBR3xxx_DEVICE 宏定义的器件支持寄存器宏，才会定义它。
		writeData 将两字节的值写入到寄存器内。LSB 构成第一个字节，MSB 构成第二个字节。 有效范围：0 – 65535
	返回值	TRUE — 如果写操作成功 FALSE — 如果写操作不成功
5	CY8CMBR3xxx_ReadDualByte	
	说明	该 API 用于从 CY8CMBR3xxx 器件的两字节寄存器中读取数据，并将级联值更新到读取缓冲区内。寄存器位置由用户指定。
	原型	bool CY8CMBR3xxx_ReadDualByte(uint8 slaveAddress, uint8 registerAddress, uint16 *readData)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		registerAddress 要读取的两字节寄存器。从该地址和相继地址中读取数据。 传输到该参数的值是 CY8CMBR3xxx_Registers.h 文件中定义的一个寄存器偏移宏。如果寄存器宏由 CY8CMBR3xxx_DEVICE 宏定义的器件支持，它才被定义。
		*readData 将从器件中读取的值更新到两字节缓冲区内。通过该 API 将器件中读取的第一个字节用于填充 LSB，使用第二个字节填充 MSB。该值的范围为 0 – 65535。
	返回值	TRUE — 如果读操作成功 FALSE — 如果读操作不成功

6	CY8CMBR3xxx_SendCommand	
	说明	该 API 将一个指令发送给 CY8CMBR3xxx 器件。调用该 API 前，您要检查器件是否已经通过调用 CY8CMBR3xxx_CheckCommandStatus() 就绪接收新的指令。
	原型	bool CY8CMBR3xxx_SendCommand(uint8 slaveAddress, uint8 command)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 - 119
		指令 要发送的操作码指令。传输到该参数的值应为 CY8CMBR3xxx_CommandsAndConfig.h 文件中所定义的一个操作码宏。
	返回值	TRUE — 如果成功发送指令 FALSE — 如果未成功发送指令
7	CY8CMBR3xxx_CheckCommandStatus	
	说明	该 API 将验证 CY8CMBR3xxx 器件是否准备好接收一条新指令，并检查先前发送的指令是否成功执行。
	原型	uint8 CY8CMBR3xxx_CheckCommandStatus(uint8 slaveAddress, uint8 *errorCode)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		*errorCode 该指针指向执行最后一条指令时返回的错误代码。如果最后指令失败，则 API 将更新该数据，以表示失败的原因。有效范围：0 到 255
	返回值	0 — 如果仍执行最后指令 1 — 如果成功执行最后指令 2 — 如果未成功执行最后指令或 I ² C 通信失败
8	CY8CMBR3xxx_Configure	
	说明	该 API 用于将用户指定的配置写入到 CY8CMBR3xxx 器件内、保存配置，并复位器件。 要配置器件，应先使用 EZ-Click 来创建配置。应该直接将 128 字节的配置数据（在 EZ-Click 项目中指定的.h 文件中生成）应用在该 API 上。 在配置保存指令执行和 CY8CMBR3xxx 器件复位过程中，该 API 会阻止 CPU 的执行。
	原型	bool CY8CMBR3xxx_Configure(uint8 slaveAddress, const unsigned char *configuration)

	参数	slaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119
		*configuration 128 字节的缓冲区用于存储器件的全部配置。该数据是从生成配置数据的 EZ-Click 中直接获得的，它包括配置寄存器数据的 126 个字节和 CRC 的两个字节。
	返回值	TRUE — 如果成功配置好器件 FALSE — 如果未成功配置器件
9	CY8CMBR3xxx_CalculateCrc	
	说明	该 API 用于计算 CY8CMBR3xxx 器件配置的 CRC 校验和。
	原型	uint16 CY8CMBR3xxx_CalculateCrc(uint8 *configuration)
	参数	*configuration 该 126 字节的缓冲区将保持用于存储器件设置的全部配置。
	返回值	已计算的 CRC 的两字节数值（较高的字节为 MSB）。
10	CY8CMBR3xxx_VerifyDeviceOnBus	
	说明	该 API 用于验证 I ² C 总线上的器件是否是一个有效的 CY8CMBR3xxx 器件。它会检查器件 ID、系列 ID 和器件版本，以验证器件。
	原型	bool CY8CMBR3xxx_VerifyDeviceOnBus(uint8 slaveAddress)
	参数	slaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119
	返回值	TRUE — 如果器件有效 FALSE — 如果器件无效
11	CY8CMBR3xxx_SetDebugDataSensorId	
	说明	该 API 设置传感器 ID，用于读取 CY8CMBR3xxx 器件中相应的调试数据。 为了读取器件中的调试数据，应先调用该 API，然后等待器件刷新时间间隔。这时再调用 CY8CMBR3xxx_ReadSensorDebugData()，实现从器件中回读调试数据。
	原型	bool CY8CMBR3xxx_SetDebugDataSensorId(uint8 slaveAddress, uint8 sensorId)
	参数	SlaveAddress CY8CMBR3xxx 器件的 I²C 地址。 有效范围：8 – 119

		sensorId 请求数据的传感器编号。最高编号既为器件所支持的传感器总数减去 1 得到的值。
	返回值	TRUE — 如果传感器 ID 写操作成功 FALSE — 如果传感器 ID 写操作不成功
12	CY8CMBR3xxx_ReadSensorDebugData	
	说明	该 API 将读取传感器的 Cp、差值、基线以及平均原始信号，并将数据更新到一个缓冲区内。 要读取调试数据，先要调试 CY8CMBR3xxx_SetDebugDataSensorId()，然后等待一段刷新时间间隔。刷新时间间隔完成后，应调用该 API 来读取器件中的数据。 如果器件的同步计数器得到匹配，那么该 API 才会更新缓冲区。如果不匹配，那么 API 将再次尝试读取数据。最多尝试读取的次数由 CY8CMBR3xxx_SYNC_COUNTER_MATCH_RETRY 定义。在 CY8CMBR3xxx_APIs.h 文件中可以设置该值。
	原型	bool CY8CMBR3xxx_ReadSensorDebugData(uint8 slaveAddress, CY8CMBR3XXX_SENSORDATA *debugData)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		*debugData 指针指向用户定义的缓冲区，其中从 CY8CMBR3xxx 器件中读取的数据得到存储。缓冲区结构被定义在 CY8CMBR3xxx_APIs.h 中。
	返回值	TRUE — 如果缓冲区被正确更新 FALSE — 如果缓冲区未被更新
13	CY8CMBR3xxx_ReadDiffCounts	
	说明	该 API 将读取 CY8CMBR3xxx 器件中所有传感器的差值，并将数据更新到缓冲区内。 如果器件的同步计数器得到匹配，则该 API 才会更新缓冲区。如果不匹配，那么 API 再尝试读取数据。最多尝试读取的次数由 CY8CMBR3xxx_SYNC_COUNTER_MATCH_RETRY 定义。在 CY8CMBR3xxx_APIs.h 文件中可以设置该值。
	原型	bool CY8CMBR3xxx_ReadDiffCounts(uint8 slaveAddress, uint16 *differenceCounts)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		*differenceCounts 它是用于存储所有传感器差值的缓冲区。缓冲区大小应等于器件中传感器的数量。

	返回值	TRUE — 如果缓冲区被正确更新 FALSE — 如果缓冲区未被更新
--	-----	--

9.2 底层 API

表 9-2. 底层主机 API

1	Host_LowLevelWrite	
	说明	该 API 通过使用 I ² C 通信协议将数据写入到 CY8CMBR3xxx 器件的寄存器图中。执行情况取决于主处理器，并要求更改 API 代码，使之符合特定的主处理器。 器件第一次执行 NACK I ² C 数据传输时，请确保该 API 在后面 340 ms 时间内重新尝试两次。.
	原型	bool Host_LowLevelWrite(uint8 slaveAddress, uint8 *writeBuffer, uint8 numberOfBytes)
	参数	SlaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119
		*writeBuffer 指的是将数据写入到器件内的缓冲区。 第一个元件应始终包含要写入的器件寄存器所在的位置。该值范围为 0 – 251。 每个连续元件都应包含要被写入到该寄存器和逐次寄存器内的数据。这些元件值的范围为 0 – 255。数据字节数的范围为 0 到 128。写零数据字节表示您正在更新想要读取器件寄存器映射信息的寄存器的位置。
		numberOfBytes 被写入的字节数等于缓冲区中元件的数量（即寄存器中数据字节数+ 1）。
	返回值	TRUE — 如果写操作成功 FALSE — 如果写操作不成功
2	Host_LowLevelRead	
	说明	使用该 API 通过 I2C 通信协议能够从 CY8CMBR3xxx 器件的寄存器图中读取数据。执行情况取决于主处理器，并要求更改 API 代码，使之符合特定的主处理器。 器件第一次执行 NACK I2C 数据传输时，请确保该 API 在后面 340 ms 时间内重新尝试两次。.
	原型	bool Host_LowLevelRead(uint8 slaveAddress, uint8 *readBuffer, uint8 numberOfBytes)
	参数	slaveAddress CY8CMBR3xxx 器件的 I ² C 地址。 有效范围：8 – 119

		*readBuffer 将从器件中读取的数据更新到的缓冲区内。 调用该 API 前，应先使用 Host_LowLevelWrite 设置需要读取的寄存器。每个后续元素都包含了从该寄存器和相继寄存器中读取的数据。这些元件值的范围为 0 – 255。
		numberOfBytes 要读取的数据字节数应等于缓冲区大小。 有效范围：1 – 252
	返回值	TRUE — 如果读操作成功 FALSE — 如果读操作不成功
3	Host_LowLevelDelay	
	说明	该 API 实现了顶层 API 所使用的延时功能。延迟周期的单位为毫秒。在所需时间内执行一个代码段即可实现该延迟。 执行情况取决于主处理器，并要求更改 API 代码，使之适合您的主机。
	原型	void Host_LowLevelDelay(uint16 milliseconds)
	参数	毫秒 需要等待的时间（单位为 ms）。 有效范围：0 – 65535
	返回值	无

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他传感器进行的扫描，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。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 $(2^N - 1) / \text{调制器的时钟频率}$ ，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，以及突然发生其他的环境变化。

噪声（CapSense 噪声）

传感器处于‘OFF’状态（无触摸）时原始计数的变量，使用峰至峰计数来测量。

噪声阈值

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

覆盖层

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

寄生电容 (C_P)

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

接近感应传感器

指的是不需要物理接触却能够检测到附近的物体的传感器。

辐射滑条

指的是包含多于一个传感器的 Widget。这些传感器以特殊的圆形方式设置，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。

刷新闻隔

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

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

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

传感器组合

请参见[组合传感器](#)。

屏蔽电极

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

屏蔽槽电容 (C_{SH})

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

信号 (CapSense 信号)

计数差值还被称为信号。请参见计数差值。

信噪比 (SNR)

有手指触摸时的传感器信号与无手指触摸时的传感器信号间的比例。

滑条分辨率

表示滑条上需要处理的手指位置总数的参数。

触摸板

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

触摸板

请参见[触摸板](#)。

调校

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

V_{REF}

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

Widget

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

修订记录



文档修订记录

文档标题: AN90071 - CY8CMBR3xxx CapSense®设计指南

文档编号: 001-91599

版本	提交日期	变更人	变更说明
**	04/21/2014	HENG	本文档版本号为 Rev**, 译自英文版 001-90071 Rev**。
*A	04/13/2015	HENG	本文档版本号为 Rev*A, 译自英文版 001-90071 Rev*E。
*B	04/27/2017	AESATMP9	更新徽标和版权。
*C	11/06/2018	XITO	本文档版本号为 Rev*C, 译自英文版 001-90071 Rev*H