AN210998

PSoC® 4 低功耗 CapSense®设计

作者：Vijay K M

相关项目：有

相关器件系列：PSoC 4 系列

软件版本：PSoC Creator™ 3.3 及更高版本

相关应用笔记：要想获取完整的列表，请点击此处。

想要获取本应用笔记的最新版本或相关项目文件，请访问 http://www.cypress.com/go/AN210998。

AN210998 向您介绍了使用 PSoC® 4 器件的资源设计低功耗 CapSense®应用的方法。本应用笔记还说明了如何计算 CapSense 应用的功耗并提供了硬件、固件和系统级指南以尽量降低功耗。

目录

1 简介 1
2 低功耗设计简介 2
2.1 电源预算要求 2
2.2 影响平均电流的因素 2
3 低功耗硬件设计中的注意事项 6
3.1 寄生电容 6
3.2 覆盖层 7
4 固件设计注意事项 8
4.1 手动调校与自动调校 8
4.2 低功耗的 CapSense 组件设置 8
4.3 动态转换功耗模式 10
4.4 刷新率 12
5 低功耗系统注意事项 12
5.1 扫描组合传感器 12
5.2 通过接近感应传感器实现接近时唤醒 13
6 示例项目 13
7 平均电流测量 15
8 总结 16
9 相关资源 16
A 附录 A. 功耗模式总结 17
A.1 活动模式 17
A.2 睡眠模式 17
A.3 深度睡眠模式 17
A.4 模式进入和唤醒源 17
B 附录 B：术语表 19
全球销售和设计支持 22
产品 22
PSoC®解决方案 22
赛普拉斯开发者社区 22
技术支持 22

1 简介

基于电容感应的设计日益被应用于各种电池供电的便携式电子电器。在这些便携式电子电器（其中电池使用寿命是最重要的因素）的设计中，功耗是一个关键参数。本应用笔记说明了如何使用 PSoC 4 器件的资源来设计低功耗 CapSense 应用。在这个过程中，它会介绍影响 CapSense 系统功耗的因素并说明进行低功耗设计时的硬件、固件和系统级注意事项。

本文档假设您已经基本了解了 PSoC 4 架构、PSoC 4 CapSense，并懂得使用 PSoC Creator™集成开发环境（IDE）为 PSoC 4 进行应用开发的方法。要想获得 PSoC 4 的基本知识，请参考 AN79953 — PSoC 4 入门手册。如果您尚未了解 PSoC 4 CapSense，请参考 AN64846 — CapSense 入门和 AN85951 — PSoC 4 CapSense 设计指南。如果您尚不熟悉 PSoC Creator，请参考 PSoC Creator 主页。

本应用笔记的第一部分介绍了影响功耗的各种因素。后面各部分则说明了进行低功耗设计时的硬件、固件和系统级注意事项。最后一部分则介绍了一个示例项目，介绍如何实现低功耗 CapSense 设计。

2 低功耗设计简介

该部分说明了功耗预算要求在系统定义中的重要性。然后简单介绍了影响 CapSense 控制器功耗的各个因素（如活动时间、睡眠时间、工作电压、工作频率、外设和 GPIO 的活动），将这些因素分为影响活动时间的因素和影响瞬间电流的因素。

2.1 电源预算要求

在低功耗电容感应系统的系统定义中，功耗预算要求是其中一项主要成分。目标功耗的初始定义为系统设计人员提供了硬件和固件开发的出发点。

功耗预算要结合电池容量和电池使用寿命得出。可以根据物理外形、成本和大小等因素决定电池容量。要想确定功耗预算，设计人员需要了解电池的使用寿命，即系统能够无需更换电池或给电池重新充电的情况下持续操作的时长。电池使用寿命的要求因应用而异。对于某些应用（如手机/便携式 gadget），电池使用寿命的要求可以以天为单位指定。对于被密封或放置在偏远区域的其他系统，要求可以以月或年为单位指定所需的电池使用寿命。在这种系统中，更换电池非常困难，因此对功耗运算的要求十分严格。所需电池的使用寿命和（能够在系统中使用的）最大电池容量被确定时，便可以计算出平均电流。更多有关电池容量的信息，请参考选定电池的数据手册。

使用以下公式，可以计算得出电池充电系统的平均电流（功耗预算）：

$$\text{最大平均电流} = \frac{Battery\ Capacity\ [mA-H]}{Required\ Battery\ Life\ [H]}$$

专门设计系统任务，以便使系统的总平均电流小于功耗预算特定的最大平均电流。在电池供电的应用中，平均功耗是实现长电池使用寿命的重要因素。

2.2 影响平均电流的因素

器件处于活动模式和非活动模式的时间（在活动和非活动模式下所消耗的）以及电流是决定 CapSense 系统功耗的主要因素。在活动模式下，CapSense 硬件会扫描传感器，检测出任何人体接触；在非活动模式下，器件将进入睡眠模式。在典型应用中，CapSense 控制器不需要一直处于活动模式。CapSense 硬件会定期扫描各个传感器。传感器被扫描时，器件处于活动模式。传感器不被扫描时，器件处于非活动模式。

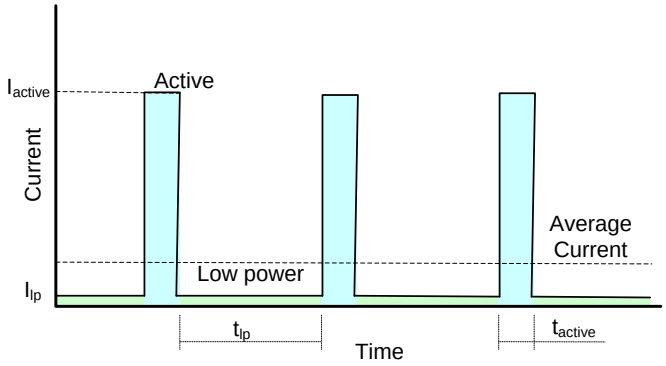
PSoC 4 支持各种低功耗模式，从而降低器件的平均功耗。要想降低平均功耗，在非活动模式下，器件应处于最低功耗模式（即深度睡眠模式）。某些 PSoC 4 器件支持休眠和停止模式，在这些模式下，器件的功耗小于深度睡眠模式下的功耗。然而，当器件通过一个中断从这些模式唤醒时会引起软件复位。如果使用了 CapSense，则这种复位需要进行特殊处理。因此，对于大部分系统，建议您使用深度睡眠模式。

由于在深度睡眠模式下 CapSense 硬件被禁用，因此必须定义唤醒器件，以扫描人体接触。在周期间隔中，可以通过使用 PSoC 4 中的看门狗定时器（WDT）将器件从深度睡眠模式唤醒。

就算在 CapSense 扫描期间（活动模式），由于在开始和结束扫描之间不需要 CPU 的干预，因此该器件可以进入睡眠模式。如果固件除等待扫描完成外没有任何其他任务，则在启动扫描后可使器件进入睡眠模式，以节省功耗。当 CapSense 硬件完成扫描操作时，它会生成一个中断，使 CPU 从睡眠模式唤醒，并返回到活动模式。

一个为了实现低功耗优化的系统，会让该系统大部分的时间处于低功耗模式，与此同时需要确保系统可以可靠的工作。图 1 和公式 1 显示的是平均电流和活动时间的关系。

图 1. 典型的低功耗系统时序图



如果器件从低功耗模式定期唤醒并在一段固定的时间内处于活动模式，那么您可以通过以下公式计算得出器件所消耗的近似平均电流：

$$\text{平均电流}(I_{AVE}) = \frac{((t_{active} * I_{active}) + (t_{lp} * I_{lp}))}{t_{active} + t_{lp}}$$

公式 1

t_{active} = 器件处于活动模式的时间

I_{active} = 活动模式下消耗的电流

t_{lp} = 器件处于低功耗模式的时间

I_{lp} = 低功耗模式下消耗的电流

使用下面公式可计算器件平均功耗：

$$P_{AVE} = V_{DD} \times I_{AVE} \qquad \text{公式 2}$$

如公式 2 所示，功耗取决于工作电压和平均电流。工作电压是影响功耗的主要因素。功耗与工作电压直接成正比。因此，您应该尽可能降低器件的工作电压。

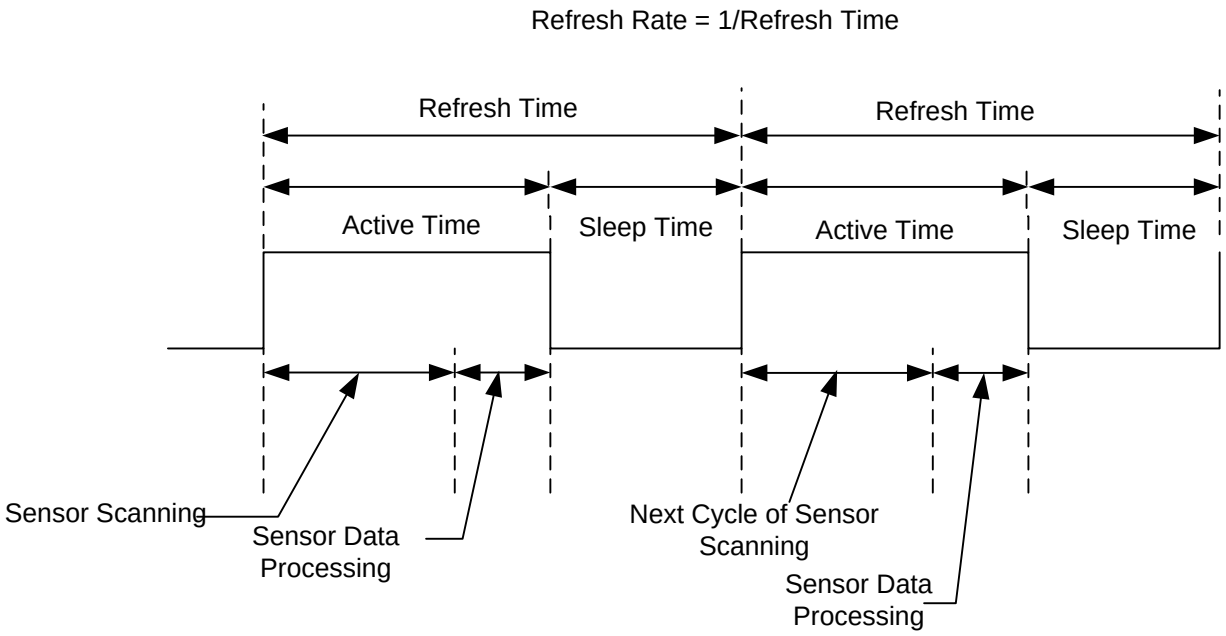
如公式 1 所示，通过降低（在活动和低功耗模式下的）活动时间和电流，可降低所耗的平均电流。

2.2.1 影响活动时间的因素

以下各因素影响活动时间，进而影响整体功耗。

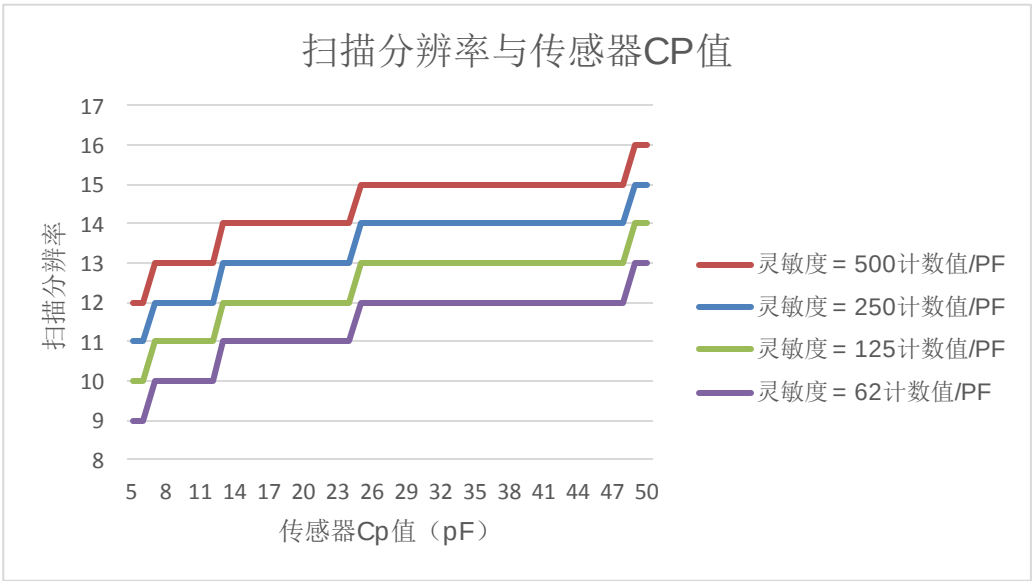
- **传感器数量**：传感器数量越多，则传感器的总扫描时间越长。对于一个固定的刷新率，如果延长传感器扫描时间，则器件处于睡眠模式的时间会减少。这样会使器件的功耗增加。
- **扫刷新率**：刷新率是每秒活动-睡眠的周期数量。该参数影响器件在固定的活动时间内处于睡眠模式的时长，如图 2 所示。通过降低I_{AVE}，可以降低平均功耗，如公式 1 所示。增加睡眠时间可使I_{AVE}降低。刷新率越低，功耗也越低。但过低的刷新率会减少响应时间。如果刷新率的时间过长，则会导致终端用户体验较差。因此，需要在响应时间和功率之间进行权衡。根据经验，刷新率最小要保持为 50 Hz，以便扫描快速手指敲击，并提供可接受的终端用户体验。根据用户是否正在触摸传感器还是离开较长时间，从而动态更改刷新率，这样能够进一步降低刷新率。在这种情况下，如果用户不靠近传感器，则以较低的刷新率来扫描传感器，而用户触摸传感器时，刷新率将被增加，以便检测出快速手指敲击。示例项目演示了如何优化响应时间并降低功耗。

图 2. 刷新率和睡眠时间的关系



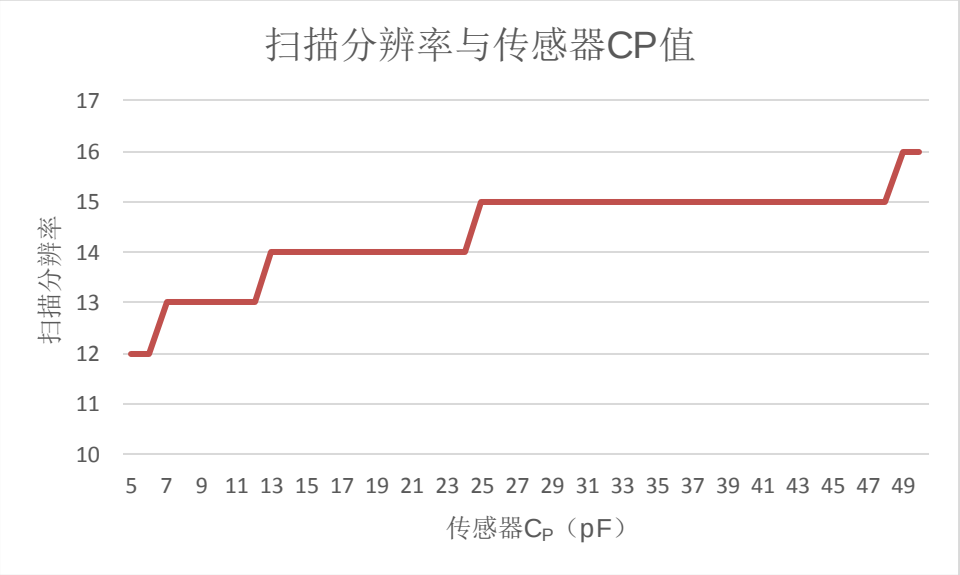
- **扫描分辨率/灵敏度：**扫描分辨率是原始计数计数器的分辨率。灵敏度参数与扫描分辨率参数相似，并被定义为每 1 pF 大小的手指电容（C_F）所获得的信号计数值。固件设计注意事项章节详细说明了这些因素。

图 3. 不同灵敏度值的传感器扫描分辨率与传感器的 C_P 值



- **传感器的寄生电容（C_P）：** 传感器的 C_P 直接影响着传感器的扫描时间。图 4 显示的是传感器 C_P 对传感器扫描分辨率参数的影响。与 C_P 较小的传感器相比，如果 C_P 值较大，则必须以较高的扫描分辨率对传感器进行扫描以获得所需信号。（对于固定的调制器时钟）较高的扫描分辨率会延长传感器扫描时间，从而导致活动时间增加和平均功耗增加。因此要进行以下布局的最佳实践，以最大程度降低传感器的 C_P 值。本应用笔记中的低功耗硬件设计中的注意事项章节提供了用于尽量降低 C_P 值的指南。

图 4. 传感器扫描分辨率与灵敏度为 500 计数值/pF 时的传感器 C_P 值



- **接近感应距离：**接近感应传感器要求扫描传感器具有高的扫描分辨率（15 或 16 位），这样可以获得较大的接近感应距离。扫描分辨率越高，扫描时间也越长，并会增长器件的活动时间，从而消耗更多的能量。因此，较大的接近感应距离需要的功耗也越大。

2.2.2 影响活动模式和低功耗模式电流的因素

平均电流取决于在活动和低功耗模式下瞬间消耗的电流。以下各因素会影响这些模式下瞬间消耗的电流：

- **操作频率：**器件在活动模式下的功耗与操作频率直接成正比。完成扫描操作后，CPU 会处理 CapSense 数据，因此较高的操作频率会使处理过程在短时间内完成。但对于不需要快速数据处理性能的系统，可将 CPU 频率降低到一个较低的值，从而减少 CPU 活动时间内的功耗。因此，在不同频率下，活动时间和活动电流之间都存在权衡。
- **未使用的外设：**只有应用需要这些外设时才给它们通电。应用不需要它们时，将它们保持在低功耗状态或关闭电源。
- **未使用的 GPIO：**应将未使用的 GPIO 配置为指定的电平（带有内部上拉电阻的输入），尽量避免悬浮输入，从而降低功耗。
- **主机通信：**在某些系统中，主机通过使用一个串行协议（I^2C、SPI 或 UART）与 CapSense 控制器进行通信。在典型的 CapSense 应用中，完成传感器扫描时，CapSense 器件会进入睡眠模式。如果（通过 I^2C/SPI/UART）对主处理器进行任何数据传输，器件将不进入睡眠模式或它会退出深度睡眠模式。因此，频繁的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，并开始进行数据操作（当 CapSense 控制器向主机发送一个主机中断脉冲时）。

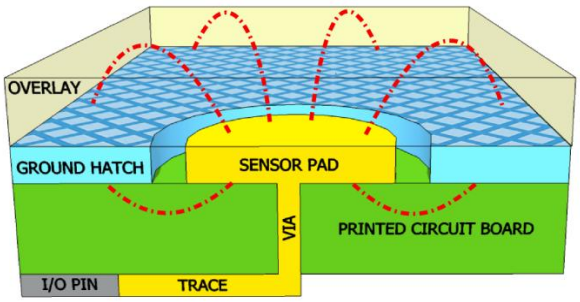
3 低功耗硬件设计中的注意事项

在 CapSense 应用中，电容性传感器是由印刷电路板（PCB）或柔性电路的走线构成的。一个良好的硬件设计可确保该设计的稳健性，并有助于实现较低的平均功耗。不好的 CapSense 硬件会增大功耗，固件算法或传感器调校无法完全补偿这种损害。以下各章节介绍了硬件设计中的注意事项，以便实现低平均功耗。

3.1 寄生电容

需要降低传感器寄生电容，从而降低功耗。在 CapSense 自电容系统（CSD）中，由控制器测量的传感器电容被称为“Cs”。当传感器上不存在手指触摸时，Cs 等于系统的寄生电容（C_P）。寄生电容是分布电容的简化形式，其中包括传感器垫块效应、覆盖层效应、CapSense 控制器引脚和传感器垫块之间走线的效应、穿过电路板的过孔的效应以及 CapSense 控制器引脚电容的效应，如图 5 所示。因此，C_P 的主要组成部分是走线电容和传感器电容。C_P 与传感器垫块周围的电场有关。增加的 C_P 值会增加功耗并降低灵敏度。

图 5. C_P 和电场



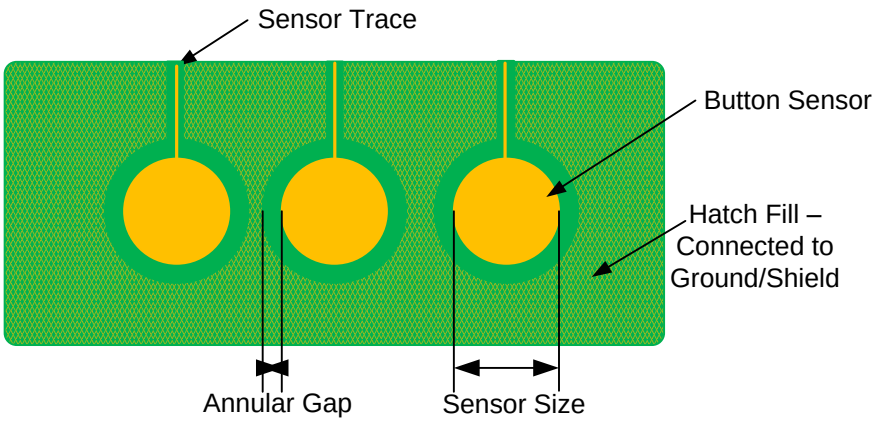
当手指触碰传感器表面时，手指和传感器垫片会通过覆盖层形成一个简单的平行板电容。这种电容被称为“手指电容”（Cf）。Cf 是分布电容的简化形式，其中包括人体和返回路径对电路板接地层的效应。

电容感应控制器会使用 CSD 方法将测量的其输入引脚上的电容转换为数字计数值。控制器使用数字计数值来识别由手指触摸导致的传感器电容增加。

要想正确检测出存在一个手指触摸，必须调校 CSD 中的扫描分辨率参数，以维持特定的灵敏度级别。如果 C_P 值较大，则需要增加传感器的扫描分辨率，从而以获得可靠的灵敏度。较高的分辨率会使转换时间（扫描时间）变长。如公式 3 所示，这样会增加电容感应应用的平均功耗。要想降低功耗，那么降低传感器的 C_p 值，从而能够设置一个较低的扫描分辨率。

- **传感器的尺寸：**传感器尺寸越小（图 6），C_P 值越低。由于 C_P 值较低，因此扫描分辨率也要低，以便获得所需信号，从而降低平均功耗。然而，如果对于覆盖层厚度和灵敏度传感器的尺寸太小，可能无法检测出手指触摸。根据覆盖层厚度的要求和灵敏度参数的设置选择最佳的传感器尺寸。请参考 AN85951 — PSoC 4 CapSense 设计指南。

图 6. CapSense 传感器布局



- **走线：**走线的长度和宽度会增加传感器的 C_P 值，另外要尽量降低这些值。将芯片放置在靠近传感器的位置可使走线长度最小，从而有助于降低 C_P 值。
- **环形间隙：**增大传感器和地面间的气隙会降低抗噪能力。因此，必须在功耗和抗噪能力之间进行权衡（根据系统要求）以获得最佳 C_P。有关环形间隙对传感器 CP 和灵敏度的详细详细，请参考 CapSense 入门。
- **填充接地网格：**使用网格接地层（并非实心接地层）可降低 C_P。建议您在传感器周围以及传感器下方的 PCB 板底层上使用网格接地层。
- **屏蔽：**为降低该传感器 C_P，推荐为顶层和底层上的填充网格发送一个驱动屏蔽信号。请在顶层上填充一个走线宽度为 0.17 mm（7 mil）、方格宽度为 1.143 mm（45 mil）的网格，并在底层上填充一个走线宽度为 0.17 mm（7 mil）、方格宽度为 1.778 mm （70 mil）的网格，然后使用驱动屏蔽信号驱动该网格。
- 通过使用环绕传感器，可以使接近感应传感器 C_P 最小。
- 对于 FPC，为了降低 C_P，请只在顶层保留填充网格。填充网格可以接地或连接一个驱动屏蔽信号。

3.2 覆盖层

覆盖层为非导电材料，可用作为 CapSense 传感器的触摸表面。覆盖层的材料和厚度决定了手指电容（C_F）手指电容与覆盖层材料的介电常数直接成正比，与覆盖层厚度成反比，如公式 4 所示。手指电容越低，信号值越低。要想补偿这种现象，需要增大分辨率，这样会加长扫描时间，从而增加功耗。

$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{d}$$ 公式 3

因此，要想实现低功耗，那么需要降低覆盖层厚度并使用具有介电常数的覆盖层材料。一些通用覆盖层材料的介电常数列在表 1 内。介电值在 2.0 和 8.0 间的材料适合于电容感应应用。

表 1. 通用材料的介电常数

材料	ε_r
空气	1.0
福米卡®	4.6 – 4.9
玻璃（标准）	7.6 – 8.0
玻璃（陶瓷）	6.0
PET 薄膜（Mylar®）	3.2
聚碳酸酯（Lexan®）	2.9 – 3.0

材料	ε_r
丙烯酸（Plexiglass®）	2.8
ABS	2.4 – 4.1
木表和桌面	1.2 – 2.5
石膏（石膏板）	2.5 – 6.0

4 固件设计注意事项

本章节介绍了用于降低功耗同时确保可靠操作的固件技术。其主要目的是为优化各任务，从而在大部分时间内器件均处于低功耗模式。PSoC 4 的低功耗模式允许降低总功耗，同时保留其基本功能。

4.1 手动调校与自动调校

CapSense CSD 方法是硬件和固件技术的结合。因此，它具有多个硬件和固件参数，用于进行正常操作。应该调校这些参数，从而使性能最佳。市场上的多数电容式触摸解决方案必须手动调试。赛普拉斯为 PSoC 4 CapSense 提供了 SmartSense（也称为自动调校）这一独特性能。

SmartSense 是一种固件算法，它可以自动将所有参数设置为最佳状态。SmartSense 自动调校性能可以缩短设计周期，并为不同的 PCB 提供稳定的性能，但它需要额外的 RAM 和 CPU 资源，以便允许运行调试 CapSense 参数。

此外，需要通过手动调校来获取最佳 CapSense 参数，但允许严格控制电容感应系统，如响应时间和功耗。通过手动调校，您便可以严格控制传感器扫描时间，使应用工程师能够将功耗尽可能降低。

各种调校模式间的功耗按照以下顺序增加：

手动调校 ≤ SmartSense（硬件参数） < SmartSense（全自动调校）

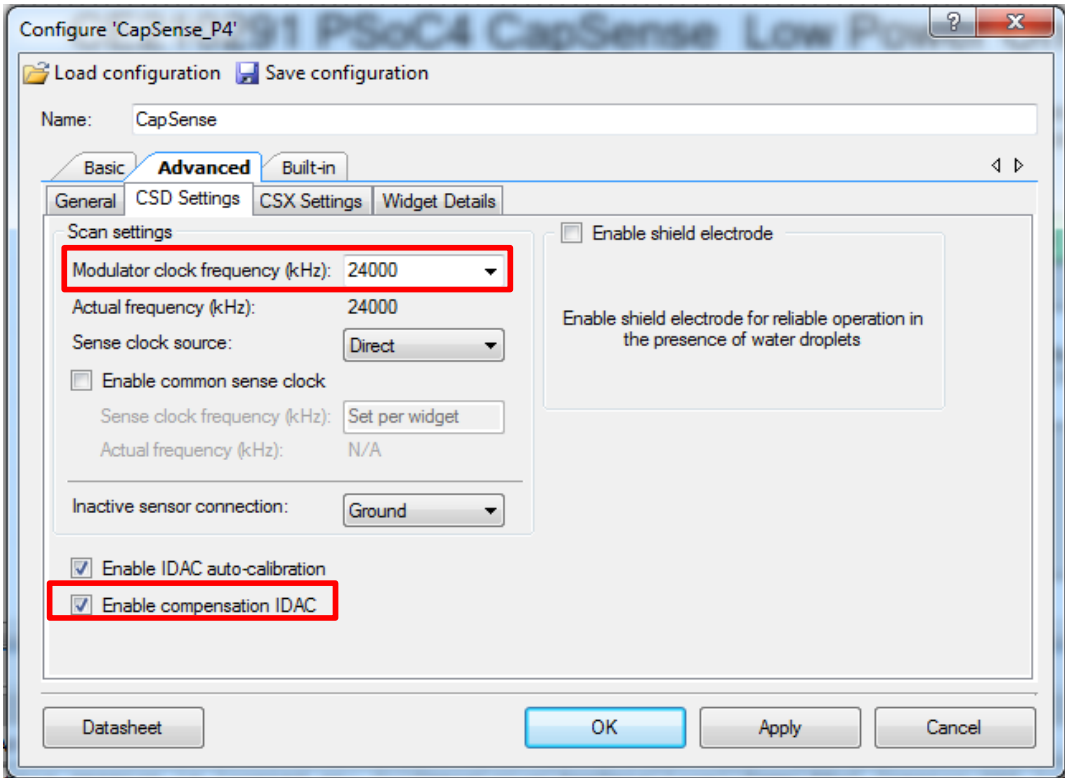
4.2 低功耗的 CapSense 组件设置

设计人员可以调校 CapSense 组件中的配置参数，以降低平均功耗。通过将组件参数配置为最佳值，可以缩短扫描时间，从而直接缩短器件处于活动模式的时间，延长睡眠时间，并降低功耗。在 CapSense 组件中，可以配置三个关键参数，以降低平均功耗，这三个参数分别为补偿 IDAC、调制器频率和扫描分辨率。

使能补偿 IDAC：由于系统可以使用较低的分辨率，从而获得同样的信号，使能补偿 IDAC 参数会降低功耗。较低的分辨率会缩短扫描时间并降低功耗。可以在 Advanced 选项卡/CSD Settings 项中设置补偿 IDAC 参数，如图 7 所示。如果在通用 DAC 要求中不需要使用补偿 IDAC，且传感器寄生电容大于 20 pF。

调制器频率：较高的调制器频率可缩短扫描时间，从而降低功耗。可以在 Advanced 选项卡/CSD Settings 项中设置该参数，如图 7 所示。

图 7. CapSense 组件中 Advanced 选项卡/CSD Settings 项



扫描分辨率/灵敏度：扫描分辨率是原始计数计数器的分辨率。传感器扫描时间与扫描分辨率参数直接成正比，如公式 3 所示。如果增大一位分辨率，扫描时间会增加一倍。

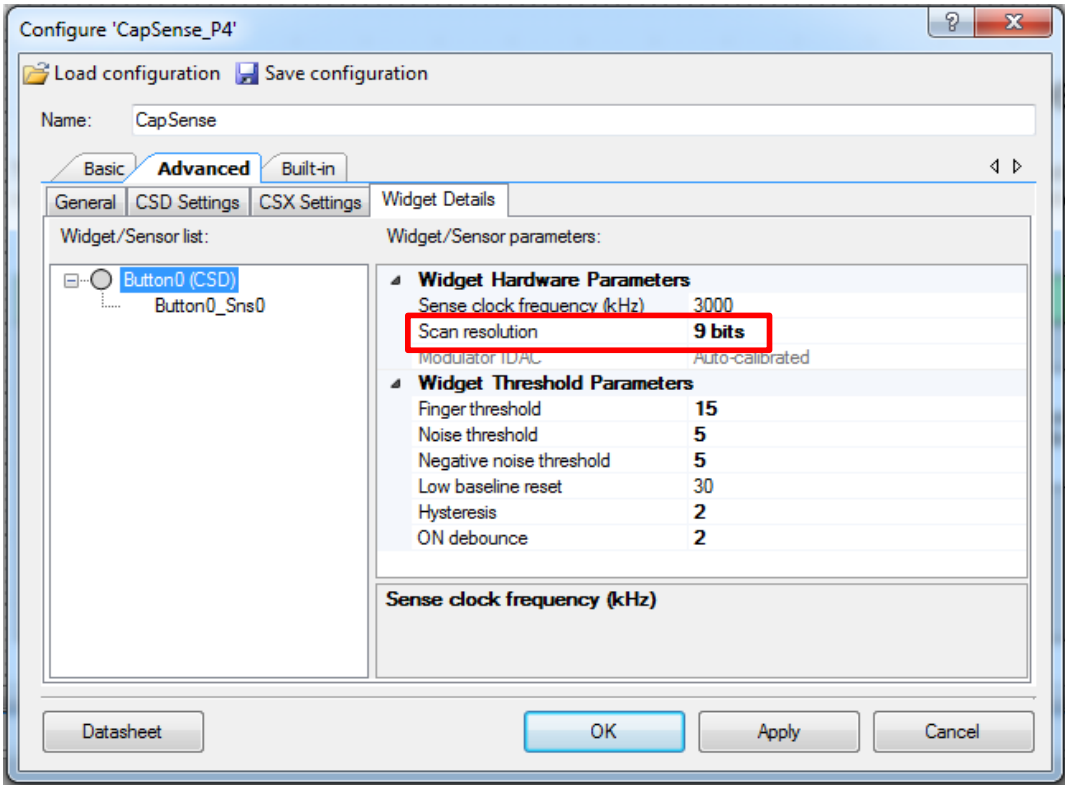
$$Sensor\ scan\ time = \left(2^{Resolution} - 1\right) \Big/ Modulator\ Clock\ Frequency \qquad \text{公式 4}$$

调制器时钟频率是原始计数计数器使用的分辨率。

灵敏度参数与扫描分辨率参数相似，并且被定义为每 1 pF 大小的手指电容（C_F）所获得的信号计数值。如果使用 SmartSense 设置调校模式，那么应该使用该参数，而不是扫描分辨率。灵敏度参数可用于提高或降低传感器信号的强度。如图 3 所示，灵敏度参数直接影响传感器的扫描分辨率和扫描时间。增大灵敏度时，这两个参数也会增加。因此，灵敏度越高，功耗则越高。

进行手动调试时，分辨率越低，扫描时间越短，功耗也会越低。但是，分辨率过低会对触摸响应和 SNR 产生影响。这是设计工程师设置分辨率时需要考虑的内容。该参数位于 Advanced 选项卡下 Widget Details 设置中，如图 7 所示。使用 SmartSense 时，通过灵敏度参数可以确定传感器扫描时间和功耗。灵敏度越低，功耗越低。需要为该参数设置一个合适的数值，使 SNR 大于 5:1。

图 8. CapSense 组件中 Advanced 选项卡下的 Widget Details 设置



4.3 动态转换功耗模式

通过动态转换器件的功耗模式，可以降低器件的平均功耗。

4.3.1 各扫描间隙期间采用深度睡眠模式

使用 PSoC 4 的深度睡眠模式可以大幅度降低 CapSense 设计的功耗。 在各 CapSense 扫描期间，使系统进入最低功耗模式（深度睡眠），便可以降低系统的平均功耗。

在深度睡眠模式下，CapSense 硬件被禁用。因此，必须频繁唤醒器件，以便扫描触摸并执行其它系统所要求的操作。在频繁的时间间隔中，可以通过使用 PSoC 4 中的看门狗定时器（WDT）将器件从深度睡眠模式唤醒。

典型应用可以通过以下方法将器件从深度睡眠模式中唤醒。

使用 WDT 定期扫描

您可以利用 WDT 使 PSoC 4 器件进入深度睡眠模式一段指定的时间，并且偶尔唤醒它以扫描 CapSense 传感器。该技术非常有用，这是因为人与用户界面（UI）系统的交互一般相当缓慢，用户并没有意识到 CapSense 系统在深度睡眠间隔中的非活动状态。

CapSense 应用的系统唤醒频率由人为响应时间决定。一个有效手指触摸的时长一般为数十秒到数百秒。因此，不需要每秒唤醒 PSoC 4 器件一次并扫描是否存在手指触摸。通过唤醒间隔可以确定 CapSense 扫描频率；该数值会因应用的不同而存在差异。

GPIO 中断

可以使用一个中断从低功耗模式中唤醒 PSoC 4 器件，该中断是由 GPIO 数字输入信号的上升沿、下降沿或双边沿生成的。使用 GPIO 中断时，主处理器、传感器或用户按键将在 GPIO 上触发一个中断，进而唤醒器件。在深度睡眠模式下，基于 GPIO 的唤醒消耗的电流最低。

I^2C 地址匹配

PSoC 4 器件中的串行通信模块（SCB）提供了一个 I^2C 功能，它可作为主设备或从设备运行。将 I^2C 配置为从设备时，它能够将 PSoC 器件从深度睡眠模式中唤醒。使用该方法时，远程主设备可以通过一次 I^2C 地址匹配来唤醒目标 PSoC 器件。PSoC 器件可以在活动模式下完成所需操作，将结果发送到 I^2C 主设备上，然后返回深度睡眠模式。

I^2C 地址匹配方法允许器件进入任意功耗模式，而不顾主控制器的功耗模式或操作模式。主机和从设备（PSoC 4）间不需同步。若需要，唤醒从设备，让其完成所需操作，然后再次返回到深度睡眠模式。这样可以避免在主机与 I^2C 总线上的其它器件进行通信时发生意外唤醒事件。

请参考 AN90799 — PSoC 4 中断，了解有关配置这些中断源的详细信息。

4.3.2 在 CapSense 扫描期间进入并保持睡眠模式

CapSense 扫描为非阻塞操作。从开始到结束 CapSense 扫描之间无需 CPU 的干预。启动一次扫描后，可以使器件进入睡眠模式以节省电源。当 CapSense 硬件完成扫描操作时，它会生成一个中断，使器件返回到活动模式。图 9 和图 10 显示了这个过程。

图 9. 不存在触摸时的时序框图

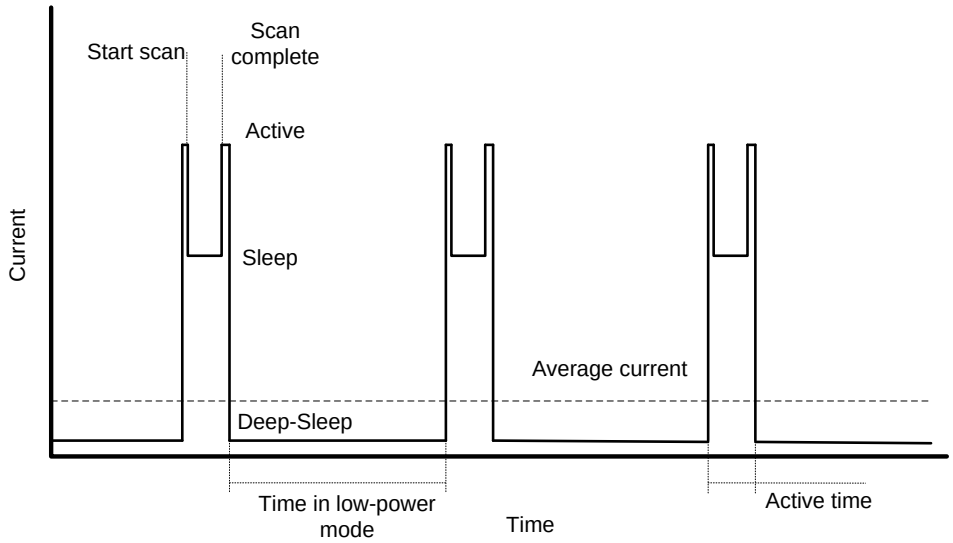
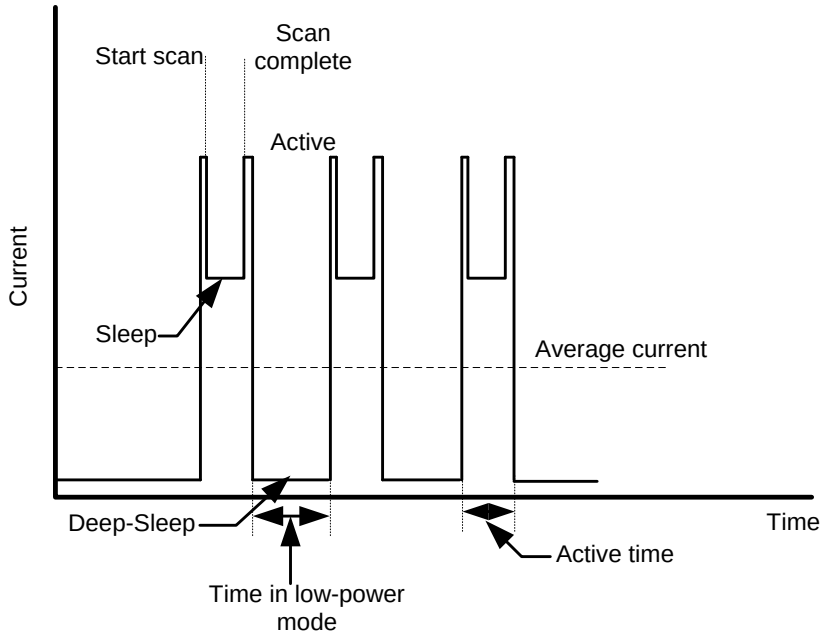


图 10. 存在触摸时的时序框图



4.4 刷新率

影响活动时间的因素如影响活动时间的因素中所述，可以通过固件配置刷新率参数，以优化系统功耗和性能。示例项目演示了如何优化响应时间并降低功耗。

5 低功耗系统注意事项

5.1 扫描组合传感器

通过对组合传感器进行采样，可以降低平均功耗。使用该方法时，在系统待机模式下，将所有可唤醒系统的物理传感器连接在一起，变成设计中的单一虚拟组合传感器。扫描组合传感器的时间比扫描所有传感器的时间短，因此，电容式控制器处于睡眠模式的时间更长，这样便可以降低平均功耗。

如果触摸了任意物理传感器，组合传感器的传感器电容会增大，表示检测到触摸。但是，仅检测到组合传感器上存在的触摸时，无法确定待机模式中触摸了哪个按键。

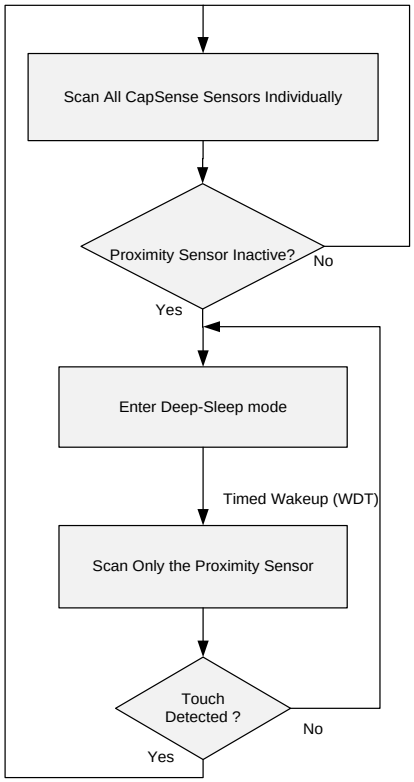
要想确定在待机模式下被触摸的具体按键，则需要唤醒电容式控制器，并使其进入活动模式。需要断开组合传感器中各个物理传感器的连接，并进行单独扫描，以确定被触摸的传感器。

在该方法中，组合传感器将多个物理传感器组合成一个单虚拟组合传感器，仅在存在触摸时才会使电容式控制器返回活动模式，因此能够降低平均功耗。电容式控制器转为活动模式后，如果在一段指定的时长内没有检测到 UI 面板上存在触摸，则电容式控制器再次进入组合传感器模式。 代码示例 CE210290 — PSoC4 CapSense 低功耗组合传感器描述了如何实现组合滑条以便降低平均功耗。

5.2 通过接近感应传感器实现接近时唤醒

对于扫描所有传感器，设计中使用的传感器数量越多，器件处于活动模式的时间就越长。这样会增加平均功耗。如果设计中有多个传感器，应添加一个围绕着各个传感器的单独接近感应传感器环路。该方法不需要物理触摸也可以唤醒系统。从深度睡眠模式唤醒时，器件只扫描该接近感应传感器，以便缩短总扫描时间，进而降低平均功耗。如果接近感应传感器已被激活（当用户的手接近 UI 面板时），则器件必须处于活动模式，并扫描其它传感器。如果接近感应传感器未被激活，则器件可返回深度睡眠模式。图 11 显示了该过程。

图 11. 通过接近感应传感器实现接近时唤醒

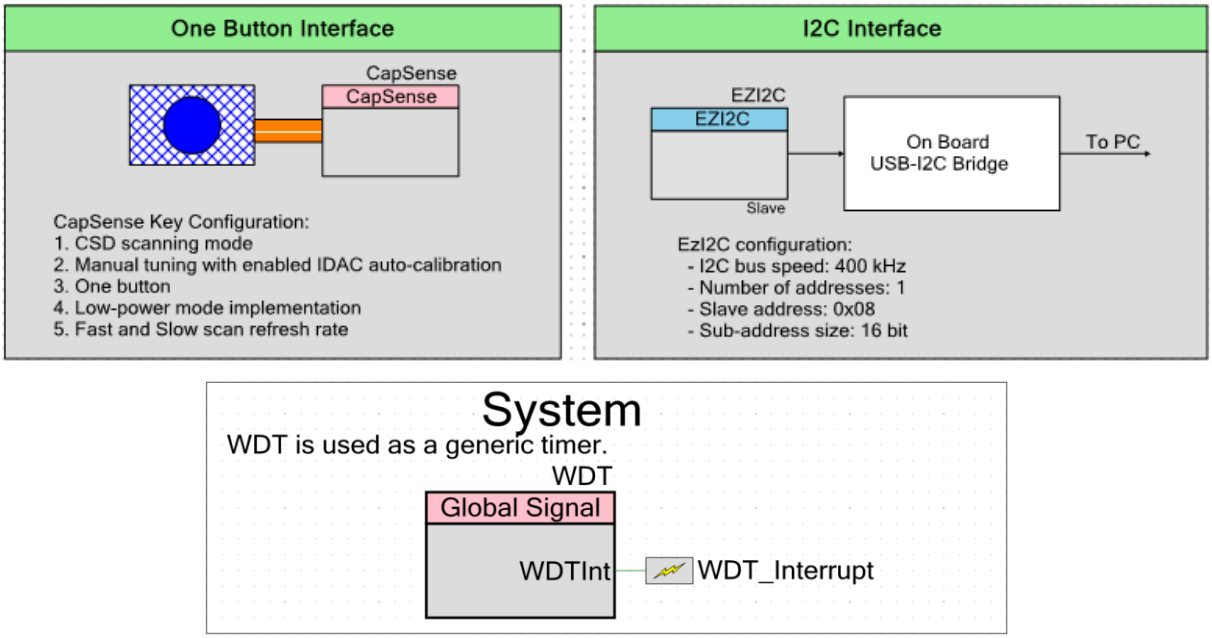


另外，为降低功耗，可以使用接近感应传感器来控制 UI 面板上的背光。每当电容式控制器处于待机模式（即所有传感器均无效）时，可以关闭背光，这样便表示设备的非活动状态。用户手接近面板，并且接近感应传感器检测到接近触摸时，将打开背光，协助用户触摸正确的按键。这样做也可以降低终端系统的总功耗。

6 示例项目

本代码示例 CE210291 — 单键的 PSoC4 低功耗 CapSense®描述了如何降低 CapSense 系统中的平均功耗，而不影响响应时间。CapSense 组件具有单键 Widget。该项目使用了使能 IDAC 自动校准的 CSD 手动调试模式。EZI2C 组件通过 I^2C 将传感器数据和按键触摸位置信息发送到外部主机/调试器上。将全局信号参考组件配置为看门狗定时器（WDT），用于设置刷新率。

图 12. 组件原理图



本代码示例执行两个刷新频率：活动模式中的 45[1] Hz 和 Look-For-Touch（LFT）模式中的 8 Hz。根据功耗预算和最佳的响应时间来选择这些刷新率。启动时，器件处于 LFT 模式，传感器的扫描频率为 8 Hz，如图 13 所示。检测到触摸时，器件转为活动模式，这时传感器的扫描频率为 45 Hz，这样便能够检测快速触摸的手指敲击，如图 14 所示。如果在指定的时长内没有检测到触摸，则器件的刷新率更改为 8 Hz，以降低平均功耗。在活动模式下，测量到的平均功耗大概为 20 µA，Look-For-Touch 模式下的平均功耗为 6 µA。

[1] 选中 45 Hz 刷新率，以满足活动模式下的 20 uA 平均电流消耗，同时确保最佳的按键响应时间。

图 13. 活动扫描模式下的时序框图

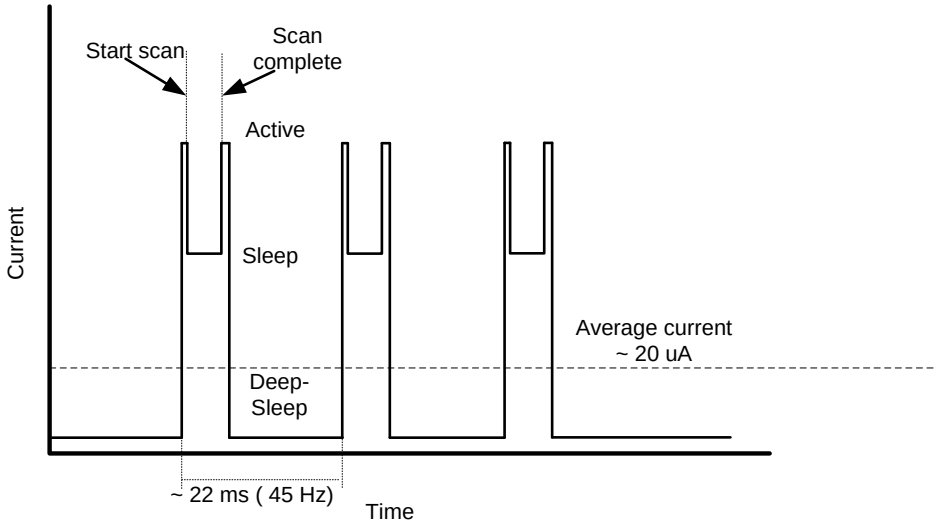
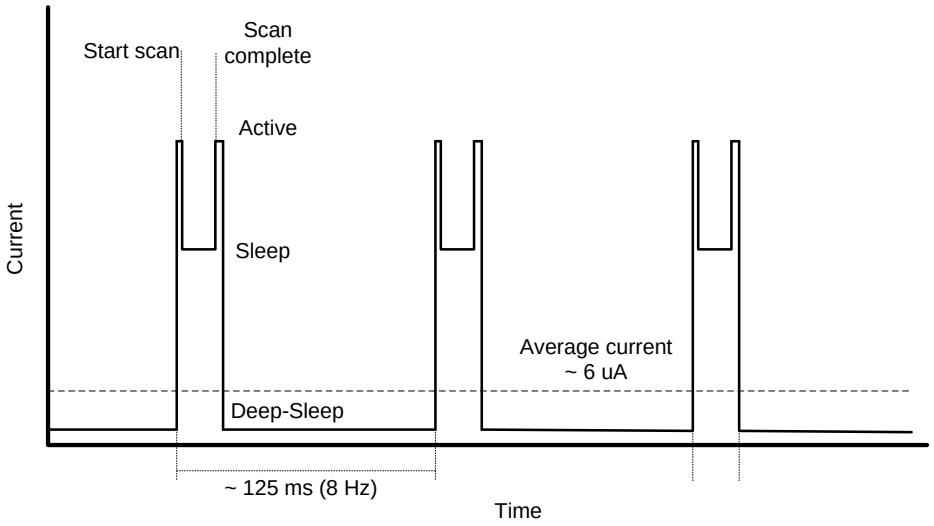


图 14. LFT 模式下的时序框图



本代码示例针对 PSoC 4000S 而设计，并且适用于 CY8CKIT-041-40xx 套件。通常通过更改器件和组件引脚分配，可以轻松将该设计移植到 PSoC 4 器件和套件中。更多有关实现固件的详细信息，请查阅 CE210291 文档。

7 平均电流测量

通过使用该示例项目，您可以测量平均电流消耗并观察电流配置文件。器件所消耗的瞬时电流不是一个固定值，而且它会随着芯片状态（功耗模式一切换，芯片状态就动态改变）而有所不同。因此，实际上您难以通过一个手持万用表来测量每一个单独的瞬时电流，因为这些电流突发的持续时间非常小。所以，建议您使用具有设置“光圈”选项的万用表进行测量。“光圈”指的是某个时间段“T”，在此期间，万用表将测量各个瞬时电流值，并将它们集成起来，然后显示出时间段“T”内的平均电流。相应的套件用户指南和代码示例文档提供了有关如何测量器件电流消耗的详细信息。以下各步骤总结了该流程：

1. 编译示例项目并将十六进制文件加载到套件内。
2. 将一个万用表（如 Agilent 34411A 6 1/2 数字万用表）连接至套件的电流测量跳线器上。
3. 根据扫描周期设置测量的光圈。如果不能确定精确的光圈，那么可以将其设置为扫描周期的整数倍。
4. 测量电流。测量到的电流就是单个间隔内的平均电流。

8 总结

AN210998 介绍了设计低功耗 CapSense 系统时需要注意的硬件和固件问题。本文档有关的示例项目描述了如何在基于 CapSense 应用中获得很低功耗，同时仍能确保最佳的触摸响应时间。

好的想法和良好的设计之间可能因为功耗问题而存在差异。通过使用 PSoC 4 中的多个节省功耗性能，您可以优化设计，并确保功耗量最低。

9 相关资源

这些应用笔记和代码示例提供了有关本文档中未能完整说明主题的更多信息：

- AN79953 — PSoC 4 入门
- AN85951 — PSoC 4 CapSense 设计指南
- AN64846 — CapSense 入门
- AN86233 — PSoC 4 低功耗模式和降低功耗技术
- AN77900 — PSoC 3 和 PSoC 5LP 的低功耗模式和降低功耗技术
- CE95288 — 使用 PSoC 4 实现低功耗 CapSense
- CE210291 — 单键的 PSoC4 低功耗 CapSense
- CE210290 — 使用组合传感器的 PSoC4 低功耗 CapSense

关于作者

姓名：Vijay Kumar Marrivagu

职务：系统工程师首席

背景：具有数年的数字设计和验证方面经验。

A 附录 A. 功耗模式总结

本部分详细描述了 PSoC 4 器件中可用的功耗模式。PSoC 4 器件具有三种工作模式：活动、睡眠和深度睡眠。这些功耗模式在功耗和外设可用性方面存在差异。

A.1 活动模式

活动模式是器件的主要工作模式，它是启动时的默认功耗模式。在该功耗模式中，源于内部主振荡器（IMO）的高频内部时钟（HFCLK）、CPU 和所有外设（除非固件特定禁用）均可用。一般情况下，活动模式中的功耗比其它模式的功耗高，这是因为芯片中的所有模块均可用。

可以从活动模式转换到其它任意模式。从其它模式中可以通过任意有效的中断或复位事件将 PSoC 4 器件返回活动模式。

A.2 睡眠模式

睡眠模式是类似于活动模式的低功耗模式，其中，器件中的所有外设均保持正常工作。在该模式中，CPU 的时钟（即系统时钟 SYSCLK）被关闭。可以通过中断从睡眠模式中唤醒 CPU。在睡眠模式中，所有外设（如 CapSense 或 I^2C 接口）可以正常运行，而 CPU 被关闭，以降低器件的功耗。

A.3 深度睡眠模式

深度睡眠是 PSoC 4 器件中可行的最低功耗模式。在该模式中，几乎所有外设（包括 CPU、IMO、CapSense 和 TCPWM）均被关闭或处于保持状态。

只有基本外设（如 I^2C、WDT 和 GPIO）可运行。根据具体的配置，I^2C 模块可以通过地址匹配事件从深度睡眠模式中唤醒器件。在深度睡眠模式中，ILO 时钟源也可以正常运行。在时钟模块中实现的 WDT 可以从深度睡眠模式中定期唤醒器件。

有关功耗的详细信息，请查阅器件数据手册。

A.4 模式进入和唤醒源

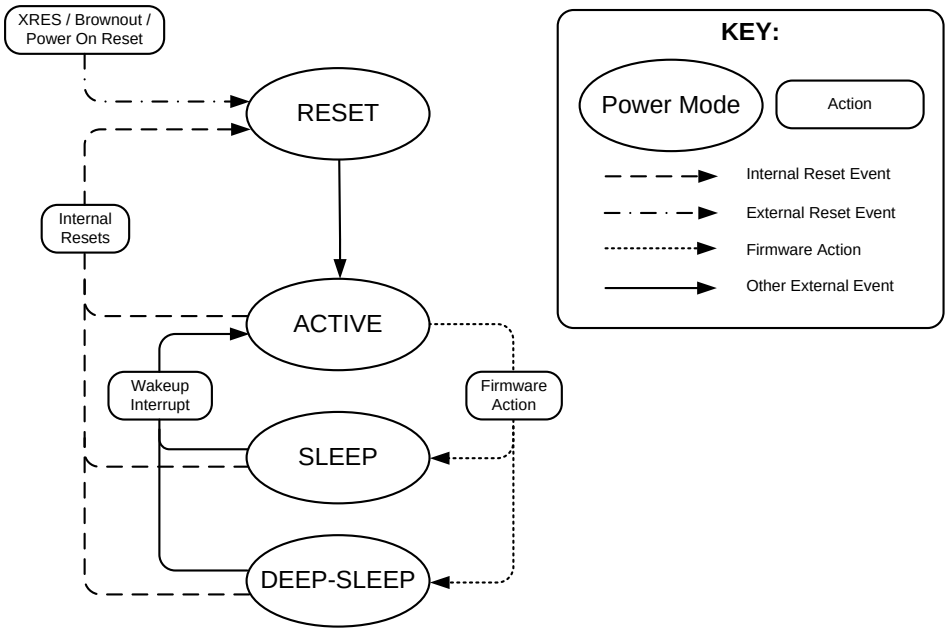
PSoC Creator 提供了一组被称为“应用编程接口（API）”的函数，用于将寄存器级操作进入低功耗模式。表 2 列出了用于进入低功耗模式的 API。它还指出退出低功耗模式并返回活动模式的唤醒源。图 12 显示的是各个功耗模式间的可行切换。

有关这些低功耗模式和唤醒源的详细信息，请参考 PSoC 4 架构技术参考手册（TRM）。请查阅 AN90799 — PSoC 4 中断，了解 PSoC 4 中可用中断的详细信息。

表 2. 模式切换详情

低功耗模式	进入低功耗模式的 API	唤醒源	唤醒事件
睡眠	CySysPmSleep()	所有中断源	中断
		所有复位源	复位
深度睡眠	CySysPmDeepSleep()	GPIO 中断	中断
		I^2C 地址匹配	中断
		看门狗定时器	中断/复位
		XRES（外部复位引脚），欠压	复位

图 15. 功耗模式切换


XRES / Brownout / Power On Reset
RESET
Internal Resets
ACTIVE
Wakeup Interrupt
Firmware Action
SLEEP
DEEP-SLEEP
KEY:
Power Mode
Action
Internal Reset Event
External Reset Event
Firmware Action
Other External Event

B 附录 B：术语表

术语	定义
电容式传感器	指的是对导致电容变化的触摸事件或接近物体作出响应的导体和基板（如印刷电路板（PCB）上的铜质按键）。
CapSense	赛普拉斯的触摸感应用户界面解决方案。
补偿 IDAC	指的是 CSD 使用的可编程恒流源，用于补偿多余的传感器寄生电容。与调制 IDAC 不同，该 IDAC 不受 CSD 模块中 Sigma-delta 调制器的控制。
CSD（CapSense Sigma Delta）	指的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，感应系统测量一个电极的自电容，并检测自电容的变化，从而确定是否有手指触摸。
驱动屏蔽	指的是 CSD 所使用的一种技术，用于使能防水功能，其中屏蔽电极由一个信号驱动，该信号的相位和幅度与传感器开关信号的相同。
电极	指的是导电材料，如 PCB 板、ITO 或 FPCB 板上的焊盘或物理层。电极被连接到 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。
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。该方法用于增加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，以检测被激活的传感器。PSoC 支持通过固件实现传感器组合，这便意味着，可以将多个传感器同时连接到 AMUXBUS，以进行扫描。
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，应将铜制地层放置在传感器周边，以获取良好的抗噪能力。但是实心填充地层会增大传感器的寄生电容（这种电容是不希望的）。因此，应以特殊网格方式填充地层。网格图案被紧密放置、纵横交错，像丝网一样，线宽度和两条线间的距离决定填充百分比。具有防水功能时，将通过屏蔽信号（而不是地层）驱动该网格填充（作为屏蔽电极使用）。
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。
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）。
手动调试	指的是手动设置（或调试）CapSense 参数的过程。
调制器时钟	指的是一个时钟源，在传感器扫描过程中用于对 CSD 模块输出的调制器进行采样。该时钟也是原始计数计数器的源。扫描时间（不包括预处理和后处理时间）的计算公式为(2N – 1)/调制器的时钟频率，其中 N 是扫描分辨率。
调制 IDAC	调制 IDAC 是可编程的恒流源，它的输出由 CSD 模块中的 Sigma-delta 调制器输出控制（ON/OFF），以保持 AMUXBUS 电压始终为 VREF。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	指的是一个电极（如 TX）与另一个电极（如 RX）间的电容。
噪声（CapSense 噪声）	传感器处于‘OFF’状态（无触摸）时原始计数的变量，即原始计数最大值减去原始计数最小值。
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（Cp）	寄生电容是由 PCB 走线、传感器焊盘、过孔以及气隙组成的传感器电极的内部电容。这是不希望发生的情况，因为它会使 CSD 的灵敏度降低。
接近感应传感器	指的是不需要物理接触却能够检测到附近的物体的传感器。
辐射滑条	指的是包含多于一个传感器的 Widget。这些传感器以特殊的圆形方式安排以检测手指的物理位置。
原始计数	指的是 CapSense 硬件模块的未处理数值输出，代表传感器的物理电容。
扫描分辨率	指的是 CSD 模块所引起的原始计数分辨率（位数）。
扫描时间	指的是完成一次传感器扫描所需的时间。

术语	定义
刷新率	指的是 CSD 模块扫描传感器的频率。刷新率决定从手指触摸到报告触摸的最短时间。最大刷新率被限制为传感器扫描时间。
自电容	指的是电极与地之间的电容。
灵敏度	指的是原始计数随传感器电容的变化，用计数/pF 表示。传感器灵敏度取决于电路板布局、覆盖层属性、感应方法以及调试参数。
感应时钟	用来实现 CSD 感应方法的开关电容前端的时钟源。
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发。屏蔽电极由 CSD 模块输出的屏蔽信号驱动。请参考驱动屏蔽内容。
信噪比（SNR）	有触摸时的传感器信号与无触摸时的传感器噪声信号的比例。
SmartSense 自动调试	设计阶段结束后，CapSense 算法会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、生产过程和环境的不同而引起的变化。
调试	指的是使 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。
Widget	指的是 CapSense 组件中包括一个传感器或一组类似传感器的用户界面元素。受支持的 Widget 包括按键、接近感应传感器、线性滑条、辐射滑条，矩阵按键和触控板。

文档修订记录

文档标题：AN210998 — PSoC® 4 低功耗 CapSense®设计

文档编号：002-12472

版本	ECN	变更者	提交日期	变更描述
**	5255821	RING	05/10/2016	本文档版本号为 Rev**，译自英文版 002-10998 Rev**。
*A	5818292	AESATMP9	07/14/2017	更新标志和版权。

全球销售和设计支持

赛普拉斯公司具有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想查找离您最近的办事处，请访问赛普拉斯所在地。

产品

ARM® Cortex® 微控制器	cypress.com/arm
汽车级产品	cypress.com/automotive
时钟与缓冲器	cypress.com/clocks
接口	cypress.com/interface
物联网	cypress.com/iot
存储器	cypress.com/memory
微控制器	cypress.com/mcu
PSoC	cypress.com/psoc
电源管理 IC	cypress.com/pmic
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线连接	cypress.com/wireless

PSoC®解决方案

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

赛普拉斯开发者社区

论坛 | WICED IoT 论坛| 项目 | 视频 | 博客 | 培训 | 组件

技术支持

cypress.com/support

此处引用的所有其他商标或注册商标归其各自所有者所有。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

赛普拉斯半导体公司，2016-2017 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。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。