

Application Note

使用 TI 传感器控制器实现楼宇自动化应用的超低功耗传感技术



Eyal Cohen, Ahmad Ibrahim, Svein Vetti

摘要

传感器控制器是 CC13x2/4 和 CC26x2/4 无线器件中配备的超低功耗辅助处理器，本应用手册探讨了它的多功能性，尤其是在楼宇自动化应用环境中。本文档阐明了传感器控制器的电源效率和强大的处理能力，其设计用于在低功耗连续传感任务中管理传感器并与之连接。

内容

1 引言.....	2
1.1 楼宇自动化中的传感器控制器.....	2
1.2 TI 器件.....	2
2 传感器控制器.....	5
2.1 特性.....	6
2.2 传感器控制器电源模式.....	7
2.3 功率测量设置.....	10
3 使用传感器控制器的楼宇自动化用例与技术.....	12
3.1 PIR 运动检测.....	12
3.2 玻璃破裂检测.....	16
3.3 门窗传感器.....	19
3.4 低功耗 ADC.....	20
3.5 使用 BOOSTXL-ULPSENSE 的不同传感器读数.....	22
4 总结.....	35
5 参考资料.....	36

商标

SimpleLink™, EnergyTrace™, and Code Composer Studio™ are trademarks of Texas Instruments.

Wi-Fi® is a registered trademark of Wi-Fi Alliance.

Bluetooth® is a registered trademark of Bluetooth Sig, Inc.

Zigbee® is a registered trademark of Zigbee Alliance.

所有商标均为其各自所有者的财产。

1 引言

以下两节介绍了使用传感器控制器引擎（尤其是在楼宇自动化应用中）的理念，还简要介绍了我们配备该引擎的无线 MCU（低于 1GHz 和/或 2.4GHz 频段）。

1.1 楼宇自动化中的传感器控制器

传感器控制器是一款可编程、自主式超低功耗 CPU，具有快速唤醒功能和极低的电流传感器读数。大多数楼宇自动化系统需要每秒多次唤醒并执行小任务，而启动和关闭能耗很容易成为应用总能耗的主要影响因素。大型高速 MCU 系统通常需要大量模块/例程，这会大幅增加从待机模式切换到工作模式的能耗。例如，更大的 MCU 系统可能需要功能更强的 PRCM（电源和时钟模块）系统。为解决这一问题，TI 推出了传感器控制器引擎，其可从待机状态唤醒（执行任务然后恢复待机状态），同时尽可能减少能耗。传感器控制器是 CC13x2/4 和 CC26x2/4 无线器件中配备的超低功耗辅助处理器，本应用手册探讨了它的多功能性，重点介绍楼宇自动化应用。

1.2 TI 器件

1.2.1 CC13x4 无线 MCU

SimpleLink™ CC13x4 器件系列是低功耗、低于 1GHz（或采用 CC1354 的多频段低于 1GHz 和 2.4GHz）无线微控制器 (MCU)，适用于需要增强安全性、片上无线更新功能以及支持高级应用或大型无线协议的应用。它支持 IEEE 802.15.4、支持 IPv6 的智能对象 (6LoWPAN)、无线 M-Bus、Wi-SUN、Amazon Sidewalk、mioty 和专有系统，包括 TI 15.4-Stack（低于 1GHz）和 2.4GHz 协议（例如 BLE 5、Zigbee 和 Thread）。

除了软件兼容之外，在低于 1GHz 无线 MCU 中，7mm × 7mm QFN 封装的 32kB 闪存到最高 1MB 闪存都是引脚对引脚兼容的，以更大限度提高设计的可扩展性。有关 TI 的低于 1GHz 器件的更多信息，请参阅 [低于 1GHz 产品](#)。

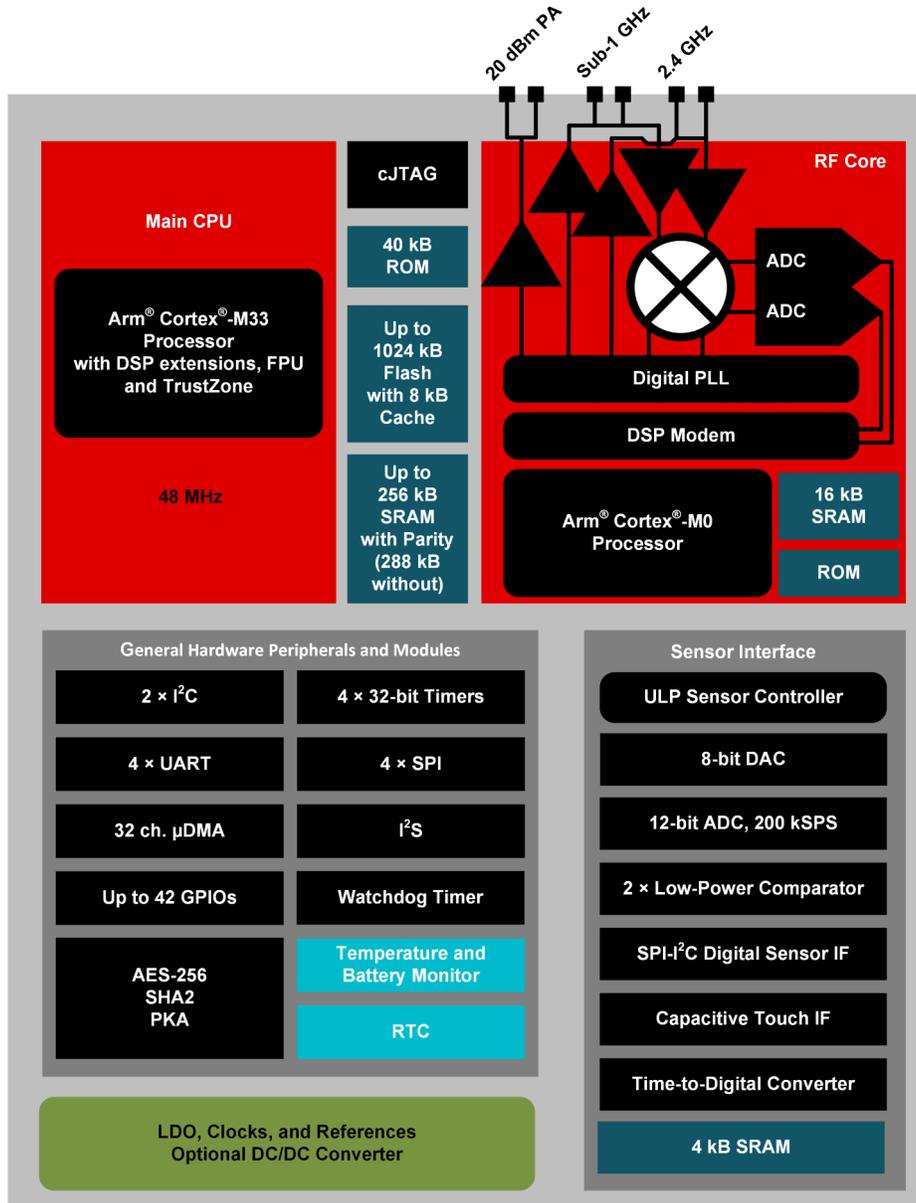


图 1-1. CC1354 方框图

1.2.2 CC26xx 无线 MCU

SimpleLink™ CC26xx 器件系列是一系列低功耗、多协议 2.4GHz 无线微控制器 (MCU)，旨在面向物联网应用。这些器件支持 Thread、Zigbee、低功耗 Bluetooth 5.3、IEEE 802.15.4、支持 IPv6 的智能对象 (6LoWPAN)、专有系统 (包括 2.4GHz 的 TI 15.4-Stack) 和通过动态多协议管理器 (DMM) 驱动程序实现的多并发协议。此器件由 ARM Cortex 驱动，后者因型号 (M3、M4) 不同而有所差异，并采用搭载 M33 处理器的最新器件。存储器配置范围从 128kB 闪存/28kB RAM 到 1MB 闪存/296kB RAM 不等。

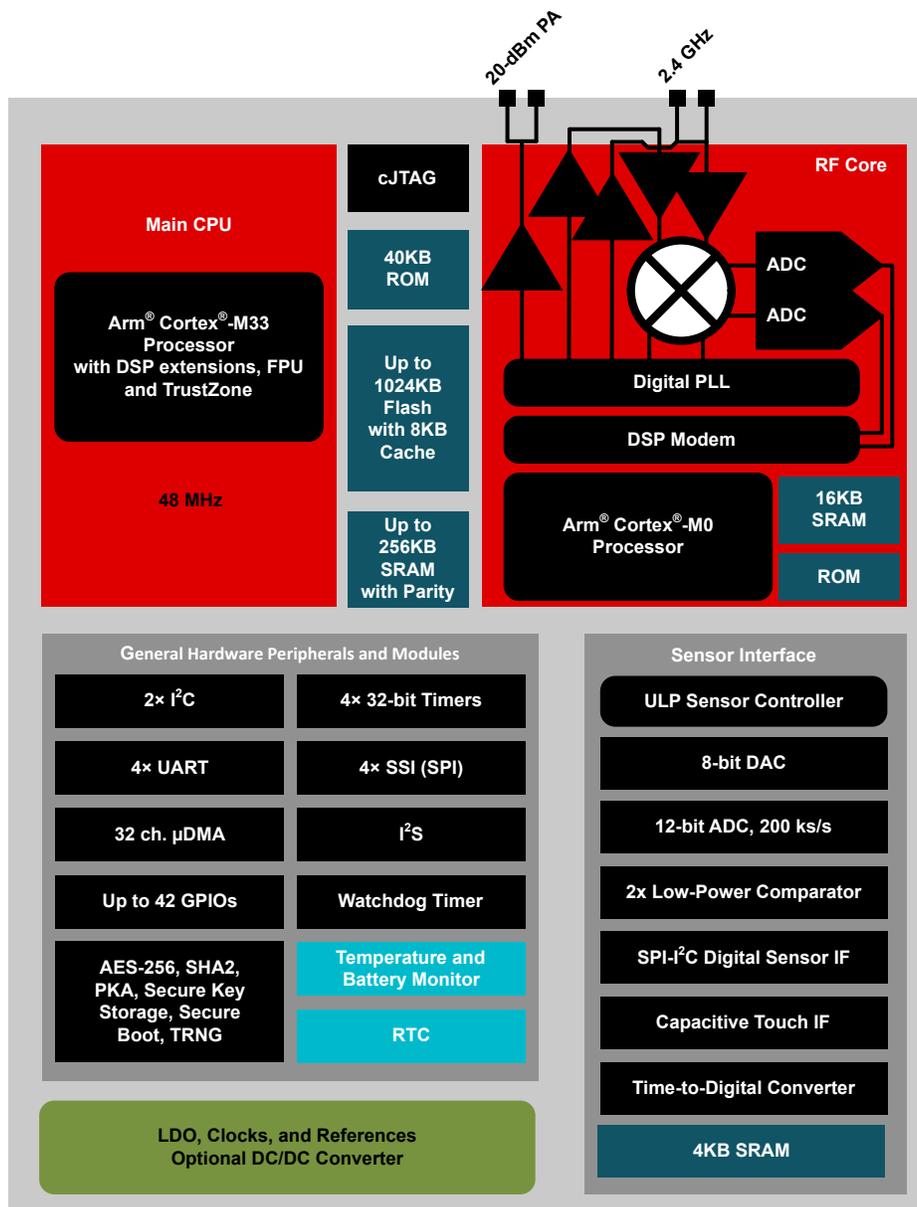


图 1-2. CC2674P10 方框图

CC13xx 器件是 SimpleLink™ MCU 平台的一部分，该平台包括 Wi-Fi®、低功耗 Bluetooth®、Thread、Zigbee®、低于 1GHz MCU 和主机 MCU，它们共用一个通用、易于使用的开发环境，其中包含单核软件开发套件 (SDK) 和丰富的工具集。借助一次性集成的 SimpleLink™ 平台，可以将产品组合中的任何器件组合添加至您的设计中，从而在设计要求变更时实现 100% 的代码重用。更多详细信息，请参阅 [SimpleLink MCU 平台](#)。

2 传感器控制器

传感器控制器是专为低功耗应用而设计的 — 使开发人员能够创建只需使用纽扣电池就能运行数年的智能传感器。传感器控制器可编程，允许用户读取和处理数据，以在系统其他部分休眠时做出低级决策。然后，传感器控制器可以唤醒系统，执行计算量更大的任务或通过无线电传输消息。

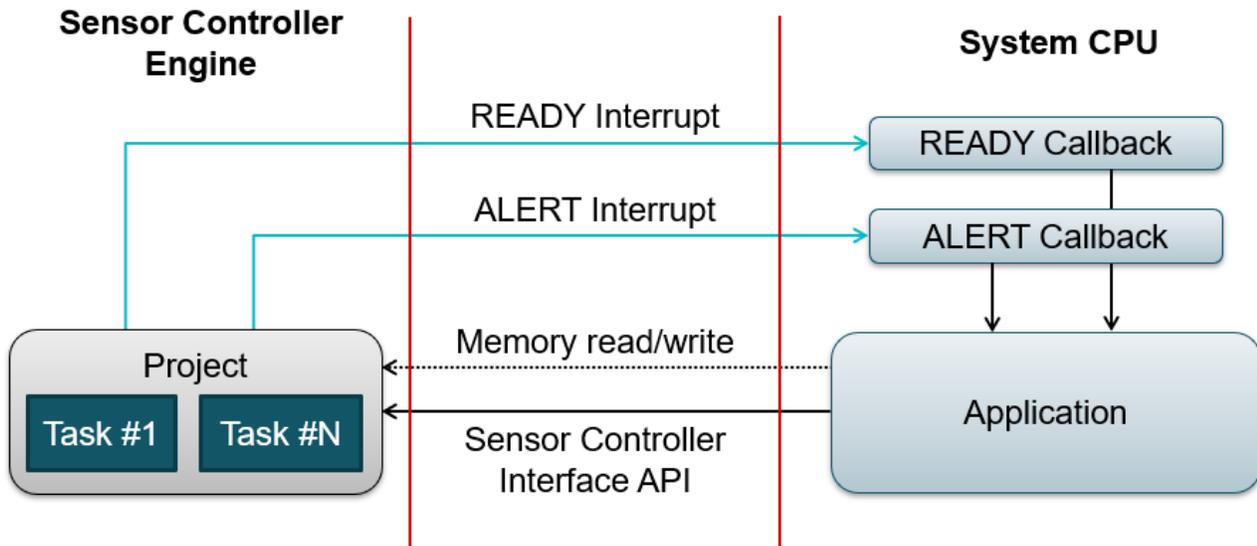


图 2-1. 系统 CPU 与传感器控制器交互

传感器控制器可实现串行运行的任务，而每个任务算法又分为四种类型的代码块：

- **初始化代码**：通过任务控制界面启动任务时运行一次
- **执行代码**：每次计划执行任务时运行（基于实时计数器 (RTC) 的周期性计时脉冲）
- **事件处理程序代码**：当设置的触发器发生时运行一次（例如，AUX I/O 引脚上的边沿或电平，或在可变延迟后）
- **终止代码**：通过任务控制界面停止任务时运行一次

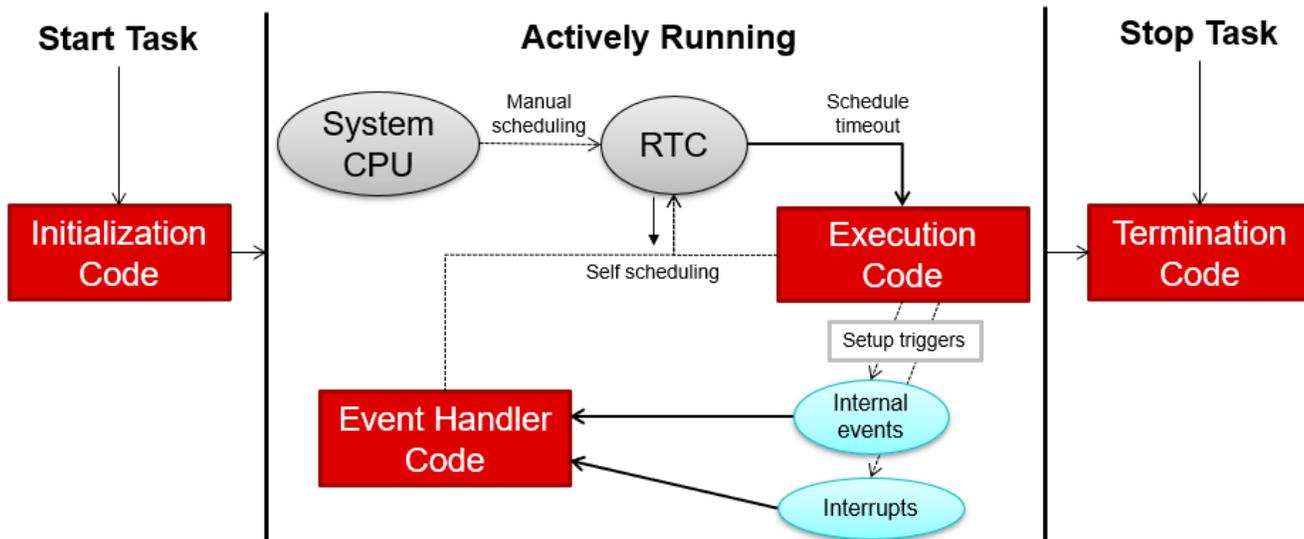


图 2-2. 任务执行流程

2.1 特性

传感器控制器可以访问模拟和数字外设，其中包括：

- 12 位 ADC，能够对支持模拟 I/O 或内部芯片电压进行采样。
- 两个比较器：一个高速连续时间比较器 (CompA) 和一个低功耗时钟比较器 (CompB)，更新频率为 32kHz。
- 8 位基准 DAC，能够支持具有基准电压的比较器。
- SPI 主器件接口。
- 位操作串行接口，包括 I2C 主器件、UART 等。
- 时间数字转换器，能够测量可配置的启动和停止触发器之间的时间。
- 可编程电流源，能够提供 0 到 20 μ A 的电流。
- 两个简单的 16 位计时器。
- 具有 PWM 和四条捕获/比较通道的计时器，无需唤醒传感器控制器即可设置和清除事件。
- 具有 4 条捕获或比较通道的 16 位异步通用计时器。
- 超低功耗 16 位脉冲计数器，能够对任何数字输入引脚或比较器输出的上升沿进行计数。
- 可访问所有 GPIO 引脚。

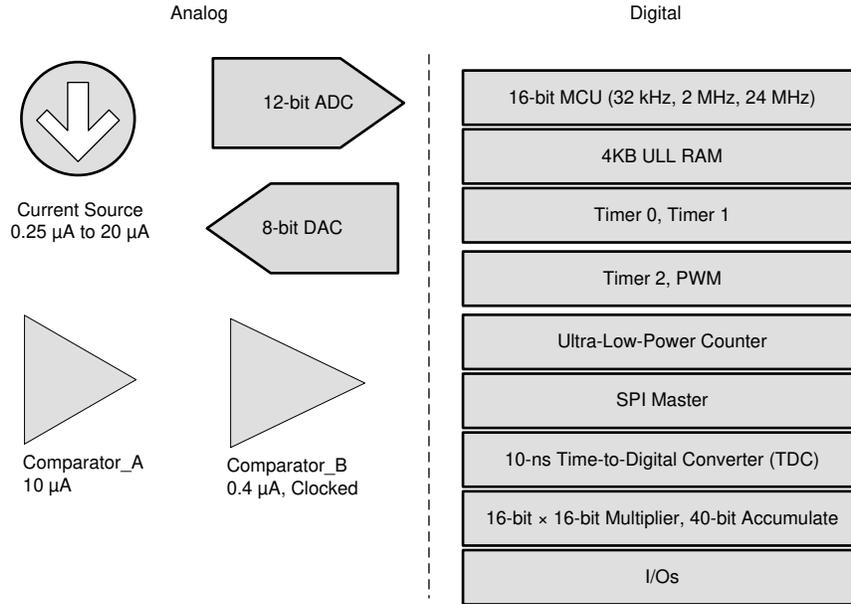


图 2-3. 传感器控制器外设

2.2 传感器控制器电源模式

传感器控制器以三种模式运行：工作 (24MHz) 模式、低功耗 (2MHz) 模式和待机模式。运行任务代码时，传感器控制器处于工作模式或低功耗模式，否则处于待机模式。

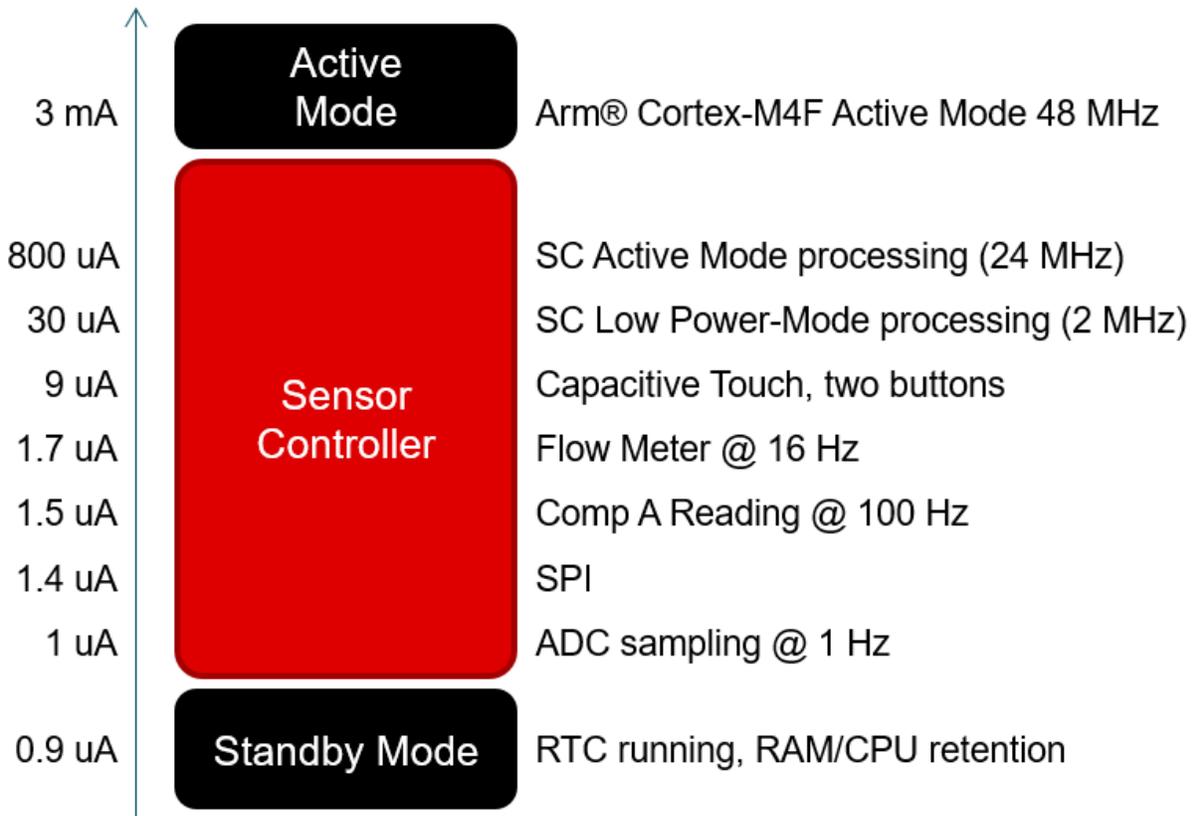


图 2-4. 传感器控制器功耗模式与 Arm Cortex M4F

传感器控制器可通过两种不同的内部电源运行 — 主电源管理系统或待机期间使用的电源方案。

主电源系统设计为提供数十毫安的电流，可为整个 IC 供电，但其缺点是从待机模式转换然后再恢复待机模式期间功耗较大。在工作模式 (24MHz) 下，需要此系统进行操作。

待机电源方案在待机模式下运行，但也可在低功耗模式下为传感器控制器供电。由于不需要更改工作模式，而且系统中有一个专用的 2MHz 振荡器（相对于更高频率的分频设计），因此低功耗 (2MHz) 运行模式专为超低功耗运行而设计。

2.2.1 工作模式

在工作模式 (24MHz) 下，传感器控制器以 24MHz 的最大速度运行，从而使控制器能够处理计算要求较高的任务。虽然在此模式下的功耗和唤醒时间较高，但与使用主 CPU 来执行类似任务相比，其相对更高效。除非系统 CPU 使用传感器，否则所有外设模块均可在工作模式下使用。

2.2.2 低功耗模式

在低功耗模式 (2MHz) 下，传感器控制器以降低的 2MHz 时钟速度运行。需要工作模式的外设模块 (SCLK_HF 和/或其他系统功能) 不可用。不可用的外设包括：

- 12 位 ADC
- 可编程电流源 (0.25µA 至 20µA)
- 时间数字转换器

然而，低功耗模式的优点是功耗更低，唤醒时间更快，同时仍能保持较简单操作的功能。

作为 12 位 ADC 的替代设计，可以使用在 2MHz 模式下运行的传感器控制器实现 8 位逐次逼近 (SAR) 型低功耗软件 ADC。本文档稍后会详细介绍这种设计。

传感器控制器可根据任务代码动态改变电源模式，并在工作模式与低功耗模式之间转换，从而在性能与功耗之间实现平衡。

例如，可以实现一种代码，只允许在某些唤醒期间进入工作模式：

```
// The task code block starts in low-power mode
// If some condition is met ...
if (...) {
    // Enter active mode
    pwrRequestAndWaitForActiveMode();

    // Get one ADC sample
    adcEnableSync(ADC_REF_FIXED, ADC_SAMPLE_TIME_2P7_US, ADC_TRIGGER_MANUAL);
    adcGenManualTrigger();
    adcReadFifo(output.adcValue);
    adcDisable();

    // Return to low-power mode
    pwrRequestAndWaitForLowPowerMode();
}

// The task code block ends in low-power mode, so unless changed by the static configuration,
// the Sensor Controller can start in low-power mode at the next wake-up
```

用于前一个代码的过程是 `pwrRequestAndWaitForActiveMode()`，它请求更改为工作电源模式，并等待更改生效；而 `pwrRequestAndWaitForLowPowerMode()` 则请求更改为低功耗模式，并等待更改生效。

2.2.3 待机模式

在待机模式下，传感器控制器基本上处于非工作状态，但会在尽可能减少功耗的同时保持状态 (RAM 保留)。

术语 *待机* 用于整个 MCU 系统处于 *暂停* 状态，但除了整个 MCU RAM 的存储器保留外，还有一个实时时钟在运行的模式。这意味着 MCU 系统可以在进入待机模式之前的位置继续运行。请注意，一些 MCU 制造商在说明其待机功耗时几乎不保留 RAM，这意味着重新开始运行需要花费更多的时间和能量。

CC13xx 系列采用 TI 设计的 65nm 工艺技术，可提供超低漏电晶体管。这些晶体管用于注重超低功耗的模块中——例如传感器控制器 MCU 系统和大多数 RAM 块。即使在高温条件下，这种专用工艺技术也能实现低功耗。

使用此功能的一项关键节能技术是传感器时序，其在需要对传感器进行采样之前的指定时间段启用传感器，使传感器控制器在传感器启动时保持最低功率状态，仅在传感器准备就绪时才激活。

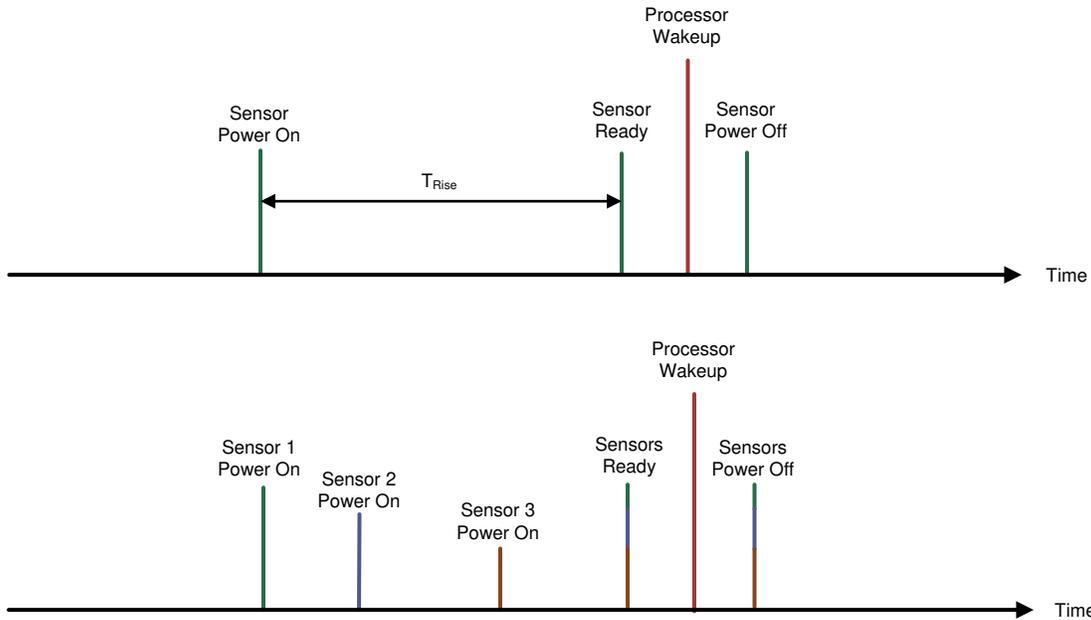


图 2-5. 带有一个和多个传感器的传感器时序

有关更多信息，请查看 [使用 CC13x2 和 CC26x2 传感器控制器的传感器时序](#)。

2.2.4 在电源模式之间切换

如本章开头所述，传感器控制器可以在两种速度下运行：**2MHz**（低功耗模式）和 **24MHz**（工作模式）。2MHz 模式在启动效率方面功耗最低，但在可用外设数量方面也存在一些限制。当传感器控制器处于待机模式（基本上处于非工作状态，但在尽可能减少功耗的同时保持状态（RAM 保留））时，系统可以唤醒至 2MHz 模式或 24MHz 模式。为了测量唤醒过程所消耗的电流（换句话说，并非真正用于应用，而只是由于唤醒过程所产生的额外能量），我们选择了一个非常简单但也可以测量的应用：从待机状态唤醒 -> 切换引脚然后恢复待机状态——每秒重复此操作 100 次 (100Hz)。鉴于应用的简单性，这并不会对总功耗造成太大影响，因此或多或少仅测量从待机模式进入工作模式 (24MHz) 然后再恢复待机模式所需的额外能量。

2.2.4.1 24MHz — 从待机状态启动并恢复待机状态的能量

放大下图以查看其中一个转换（绿色表示引脚切换，黄色表示电流消耗，x 轴表示时间）仪器测量得出的能量消耗为 145nC。如果使用此值来估算总消耗，响应为 $100\text{Hz} \times 145\text{nC} = 14.5\mu\text{A}$ 。仪器的测量值显示平均电流消耗为 13.5uA，因此在这种情况下，使用每次唤醒的启动能量测量值可以进行相当好的估算。

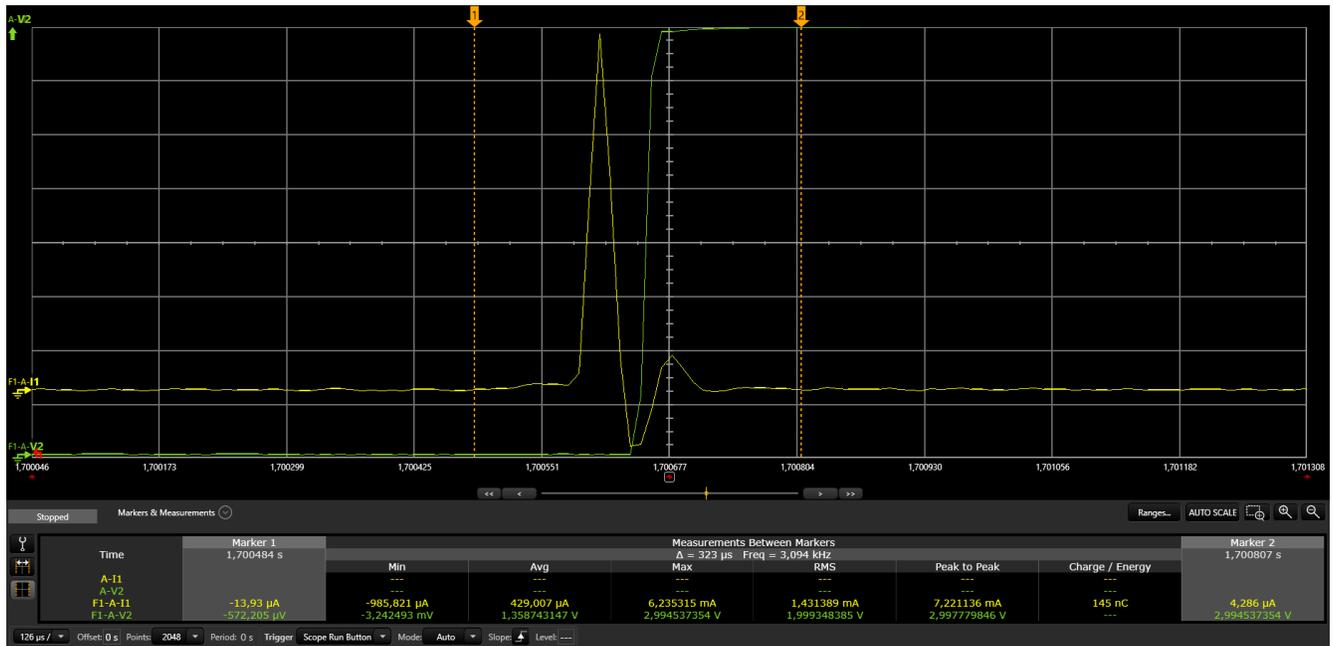


图 2-6. 24MHz 模式下的 100Hz 引脚唤醒测试案例

2.2.4.2 2MHz — 从待机状态启动并恢复待机状态的能量

放大图 2-7 以查看其中一个转换（绿色表示引脚切换，黄色表示电流消耗，x 轴表示时间）仪器测量得出的能量消耗为 8nC。如果我们使用此值来估算总消耗，得出的结果是 $100\text{Hz} \times 8\text{nC} = 0.8\mu\text{A}$ 。仪器的测量值显示平均电流消耗为 867nA，因此在这种情况下，使用每次唤醒的启动能量测量值可以进行不错的估算。

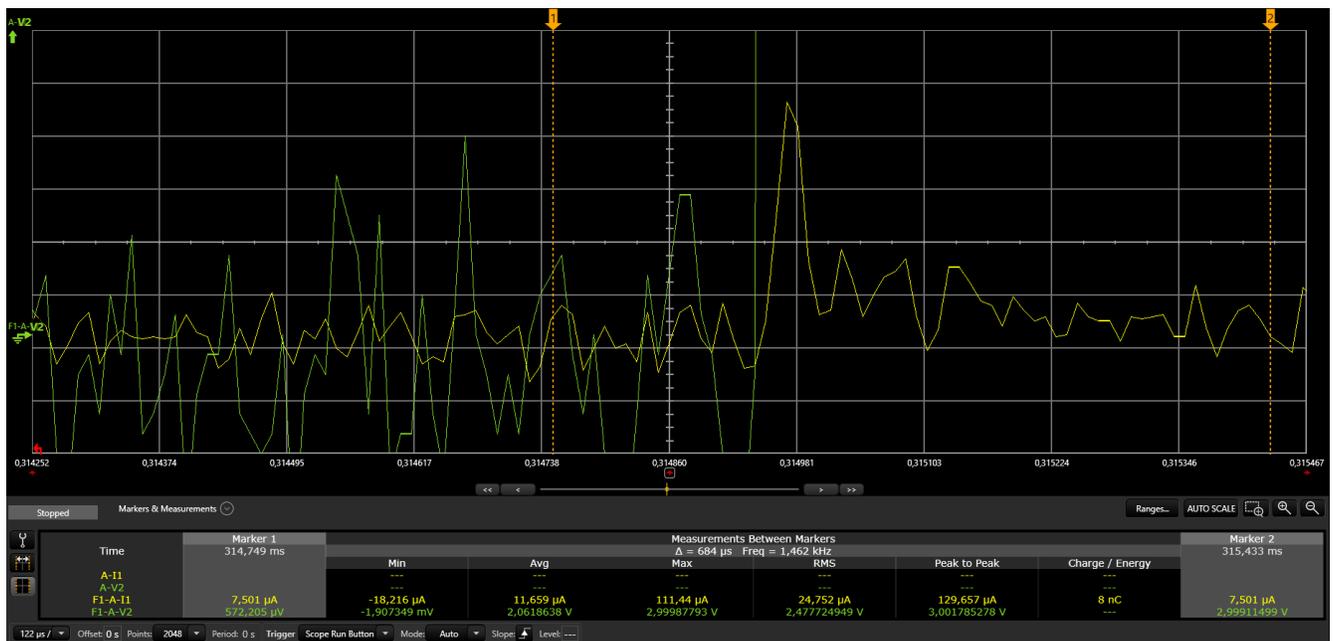


图 2-7. 2MHz 模式下的 100Hz 引脚唤醒测试案例

2.3 功率测量设置

本应用手册中提到的所有测量均使用 LAUNCHXL-CC1352P 修订版 B 电路板进行，其配备了所需的 Booster Pack 或电路板，但适用于所有带传感器控制器的器件。所使用的任何附加设备均在相应的应用中说明。

2.3.1 EnergyTrace™ 软件

对于电流测量，传统上使用功率分析仪进行功率/电流消耗测量，这种方法可提供详细的见解，但通常需要昂贵、笨重且并非始终可用的设备。采用 [CC13x2/CC26x2 的超低功耗传感应用中展示了如何使用 LAUNCHXL-CC1312R1 上的功率分析仪进行测量的示例。](#)

另一种设计是 EnergyTrace™ 软件，该软件是 Code Composer Studio™ (CSS) IDE 6.0 及更高版本中包含的、基于能量的代码分析工具。EnergyTrace™ 显著简化了该过程，允许开发人员在开发过程中直接从电路板测量和分析实时能耗。EnergyTrace™ 可实时跟踪能量和电流信号，并提供程序运行期间的功率、电压和电流平均值以及电池寿命估算。

2.3.2 软件

使用以下软件：

- Sensor Controller Studio 2.9.0.208
- Code Composer Studio 12.7.1
- SimpleLink CC13x2 / CC26x2 SDK 版本 7.40.00.77

2.3.3 电流消耗测量

对于每次测量，XDS110 调试器跳线最初都保存在 LAUNCHXL-CC1352P 电路板上。然后，使用 CCS IDE 将相关的应用软件刷写到 CC1352 上。完成后，将电路板与 PC 器件断开连接，并移除 Launchpad 上的所有跳线，XDS110 GND、3.3V、RXD<< 和 XDS110 Power 跳线除外，因为这些跳线对于 EnergyTrace™ 的功能至关重要。然后，使用 USB micro 将电路板重新连接到 PC 器件，并启动 EnergyTrace™。

每个报告的电流值都是四块电路板上 10 次测量的平均值（总共进行 40 次测量），并且是指定程序运行 30 秒时间窗口的平均值。所有测量均在室温环境下进行。

本应用手册使用以下软件和硬件来确定传感器控制器的电流消耗：

2.3.4 硬件

使用以下硬件：

- [LAUNCHXL-CC1352P](#)，套件修订版 B
- [BOOSTXL-ULPSENSE](#)，套件修订版 A
- [门窗传感器评估平台简介和性能概览](#)，应用手册

3 使用传感器控制器的楼宇自动化用例与技术

传感器控制器为开发高效楼宇自动化设计提供了一个平台。超低功耗运行和处理实时数据收集的能力使传感器控制器非常适合基于 PIR 的运动检测、玻璃破裂检测、门窗和感应等关键用例：

- **运动检测 — 模拟、数字 PIR 传感器：**

传感器控制器能够以为应用设计的频率定期从 PIR 传感器的数据进行采样，应用数字处理方法来优化接收到的信号，并使用基本算法（基于阈值以及其他方法）来检测运动。这样，主 CPU 就可以一直处于睡眠模式，直到检测到运动。

- **门窗传感器 — 簧片开关和磁铁传感器：**

传感器控制器能够定期检查簧片开关的状态，滤除噪声或任何误触发（例如通过对开关进行去抖动处理），确保仅处理合法事件。当检测发生时，传感器控制器可唤醒系统 CPU，而同时开关在大部分数例程期间以低功耗模式运行。

- **玻璃破裂检测 — 麦克风、压电式传感器：**

对于玻璃破裂检测，传感器控制器可使用麦克风监测音频或振动信号，也可以使用外部压电式压力传感器测量压力。然后，通过分析传入的信号是否存在特定频率模式来判断玻璃是否破裂。如有需要，主处理器还可进一步处理其他的处理。

以下各章将演示如何使用传感器控制器实现上述每种应用，并重点关注低功耗以延长电池寿命。

3.1 PIR 运动检测

可以使用多种不同类型的传感器来检测基本运动，但过去十年最常见的设计可能是使用 PIR 传感器。PIR 传感器基于维恩位移定律；根据该定律，不同温度下的黑体辐射曲线可在与温度成反比的不同波长下达到峰值。基本上，要监测红外频谱，不同温度下的物体可辐射不同水平的能量。图 3-1 在使用红外摄像头拍摄的图像中看到的内容：

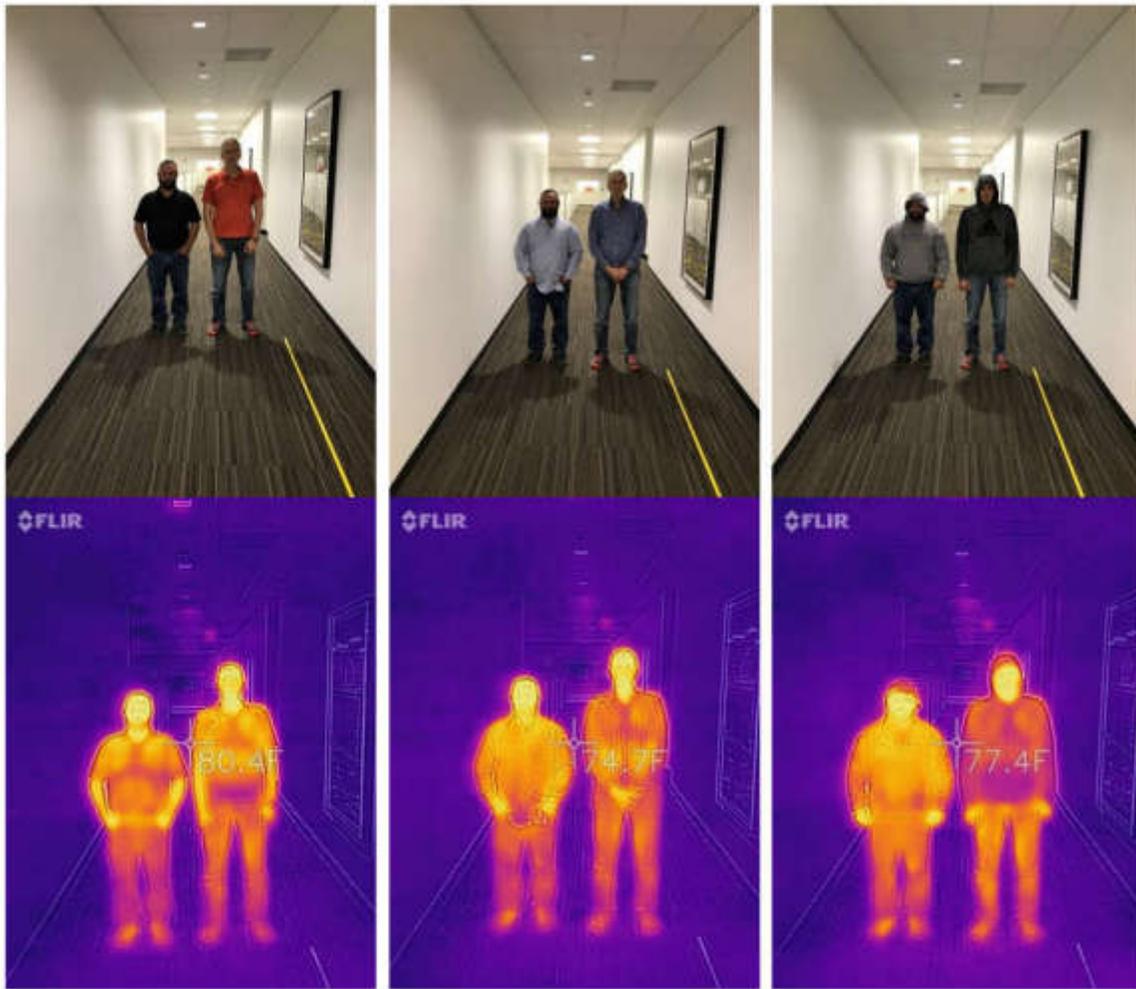


图 3-1. 红外图像

3.1.1 PIR 传统信号链

在传统的 PIR 运动检测信号链中，来自 PIR 传感器的信号通常经过一系列增益带通滤波器（通常包含直流阻断电容器），然后馈送到一组比较器以进行低侧和高侧波形检测（用作窗口比较器）。

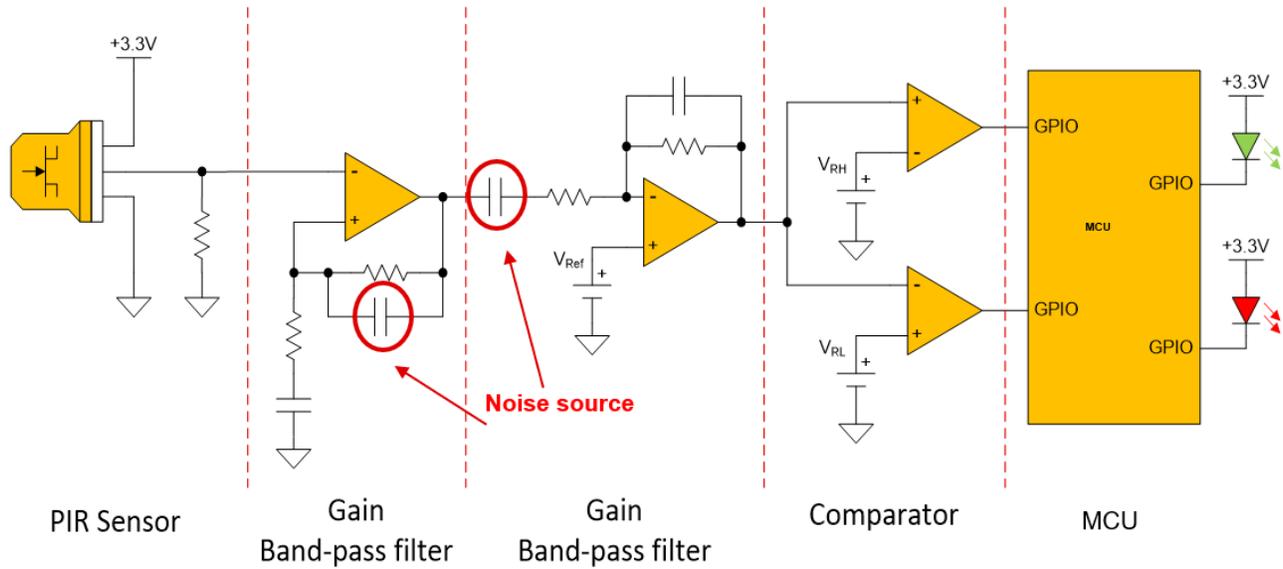


图 3-2. PIR 传统信号链

对于上述情况，使用纯硬件元件的主要缺点之一是，信号链是固定且有限的。对于带通滤波器，需专门针对特定的检测范围和运动速度来配置增益和截止频率。典型截止频率约为 0.7Hz 至 30Hz，总体信号增益可高达 1000 倍。我们在调查过程中发现此信号链的另一个缺点是，这些直流滤波电容器实际上会成为信号链中非常大的噪声源。陶瓷表面贴装电容器通常由钛酸钡制成，而钛酸钡具有压电效应，这意味着任何噪声或振动实际上都会在电容器上产生小噪声信号。处于如此低频率的钽电容器也会给信号带来噪声。在信号链中将此噪声与高达 1000 倍的增益耦合可能会返回非常差的信噪比。

3.1.2 无电容器运动检测方框图

下面的方框图演示了使用传感器控制器引擎实现运动检测的过程。该方框图提供模拟 PIR 传感器的简单信号链，其中整合了一个用于缓冲信号的运算放大器 (op-amp)，以及另一个用于放大 PIR 信号的运算放大器。这种无电容器设计利用传感器控制器的内部 DAC 有效消除了 PIR 信号的直流分量。DAC 稳定至 PIR 信号的直流值，每隔几秒更新一次，以确保准确的检测和增强的系统性能。

阈值检测和运动频率可通过软件进行调整，从而实现灵活、可定制的运动检测。检测到运动时，传感器控制器会唤醒主 ARM CPU，然后激活无线电，以通过低于 1GHz 或 2.4GHz 发送信号。在没有运动的情况下，主 CPU 会保持在深度睡眠模式，从而确保超低功耗和延长电池寿命。这种设计有效地平衡了性能与能效，专为低功耗运动检测应用而设计。

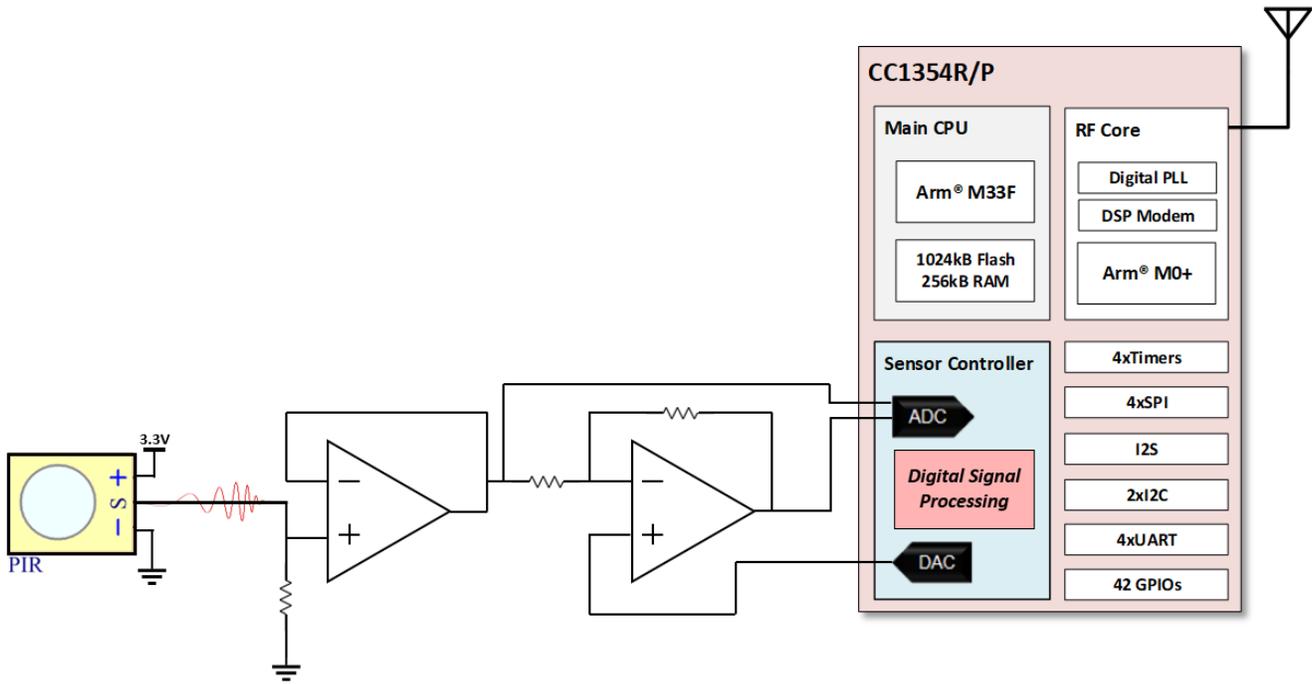


图 3-3. 无电容器 PIR 运动检测方框图

3.1.3 数字信号处理

以下一节详细介绍了 PIR 运动检测所实现的数字信号处理。此过程旨在优化系统的灵敏度和精度，同时更最大限度地降低噪声和减少误触发。

3.1.3.1 硬件

- CC1352P 传感器控制器
- Murata IRA-S210ST01 模拟 PIR 传感器
- 菲涅尔透镜 IML-0685

我们用于此运动检测的信号链主要是纯缓冲和增益级。为了取消传统电路中会给系统增加噪声的直流阻断电容器，需使用 DAC 对这些级进行偏置。信号馈送到模数转换器 (ADC)，而不是使用比较器，因此可以使 DAC 的反馈环路跟踪环境和环境温度的变化，并支持对信号进行数字滤波。

3.1.3.2 数字信号处理

图 3-4 所示为随着环境温度和 PIR 传感器体温波动，PIR 信号随时间变化的情况。此处不存在运动检测事件，只有信号漂移。

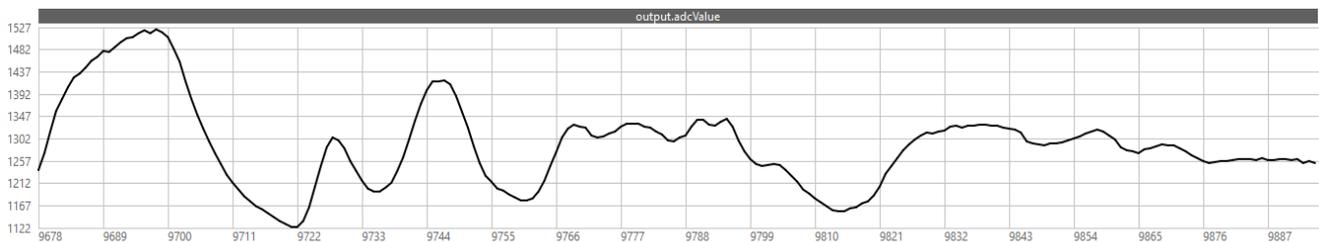


图 3-4. PIR 信号随时间变化的情况

PIR 原始信号可能会因环境变化（如温度波动或背景干扰）而产生一定的噪声。因此，应避免对原始信号使用简单的阈值，因为这可能会上下波动，从而导致检测不可靠。相反，我们要分析信号的一阶导数，以测量信号随时

间上升的速度。信号的快速变化会导致较高的一阶导数，我们可基于此设定阈值，以更可靠地检测运动。应用这种方法之前，我们会对原始信号进行过采样，并使用移动平均滤波器来消除小的峰值。

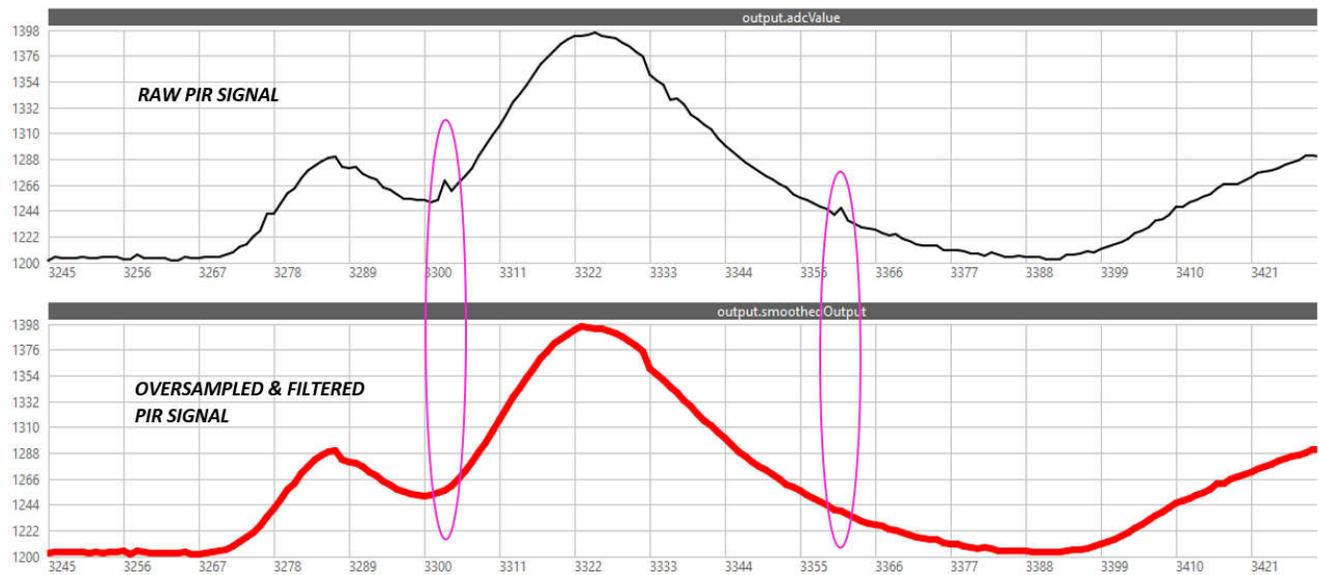


图 3-5. 平滑的 PIR 信号与原始信号

平滑原始 PIR 信号后，将计算一阶导数的绝对值。这一步骤使我们能够关注信号变化的幅度，而不受变化方向的影响。为了检测运动，我们在此绝对一阶导数上设置了软件定义的阈值。如果导数幅度超过阈值，则表示信号发生了快速变化，这与运动相对应。这种方法提供了一种稳健的运动检测方法，同时更大限度地减少了因信号逐渐波动或环境噪声造成的误触发。

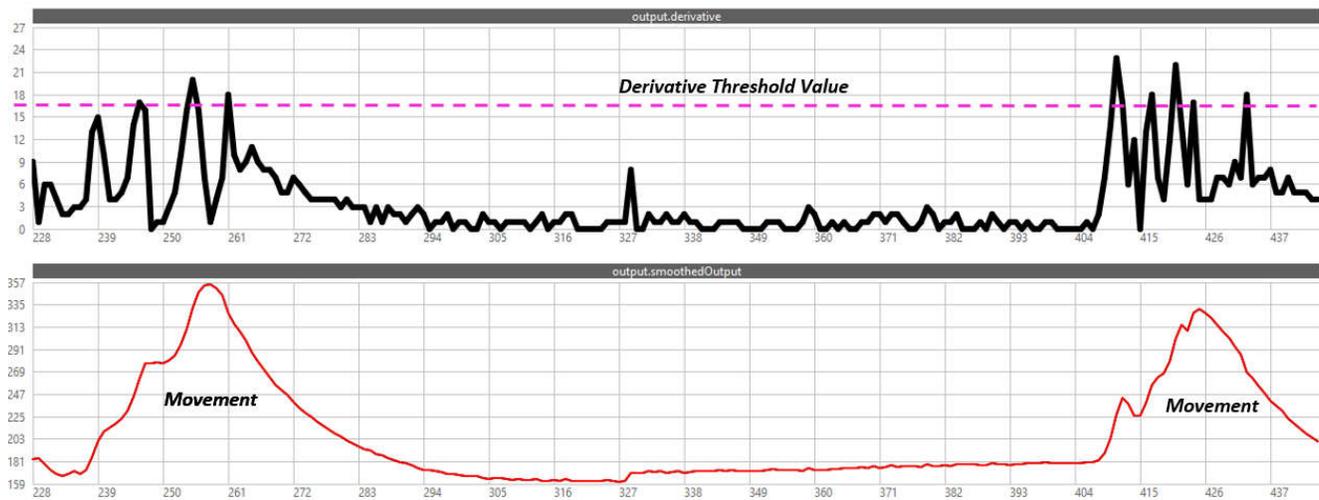


图 3-6. 平滑信号的一阶导数

3.2 玻璃破裂检测

为了在楼宇自动化中进行可靠的玻璃破裂检测，可以使用各种基于传感器的方法来识别与玻璃破裂相关的独特信号。通过结合声音与振动检测，或使用先进的频率滤波，这些方法有助于确保实现准确、迅速响应的监控并更大限度地减少误报。常见的方法包括使用压电式传感器检测冲击振动，使用麦克风进行音频模式识别，甚至使用加速度计进行多轴振动感应。每种方法都具有独特的优势，可以量身打造以满足自动化楼宇系统的特定安全需求和功率限制。

3.2.1 低功耗与低成本玻璃破裂方框图

图 3-7 说明了如何利用 CC13xx 的内部外设来设计具有外部压电式传感器的简化超低功耗玻璃破裂探测器。在此用例中，我们将内部比较器 B 以及基准 DAC 和 ADC 一起使用。COMPB 外设是一款低功耗时钟比较器，其更新频率为 32kHz。COMPB 可用于持续监控慢速信号，并从待机模式唤醒传感器控制器。监控的信号包括但不限于电源电压或模拟传感器输出。在 Sensor Controller Studio 中，可以使用内部基准 DAC 配置 COMPB 外设的阈值电压。一旦压电式传感器信号超过配置的阈值，系统就会使用内部 ADC 在指定时间内对整个压电信号进行采样。然后，该采样数据将用于计算信号的能量，从而区分玻璃上的敲击声和真正的裂探。

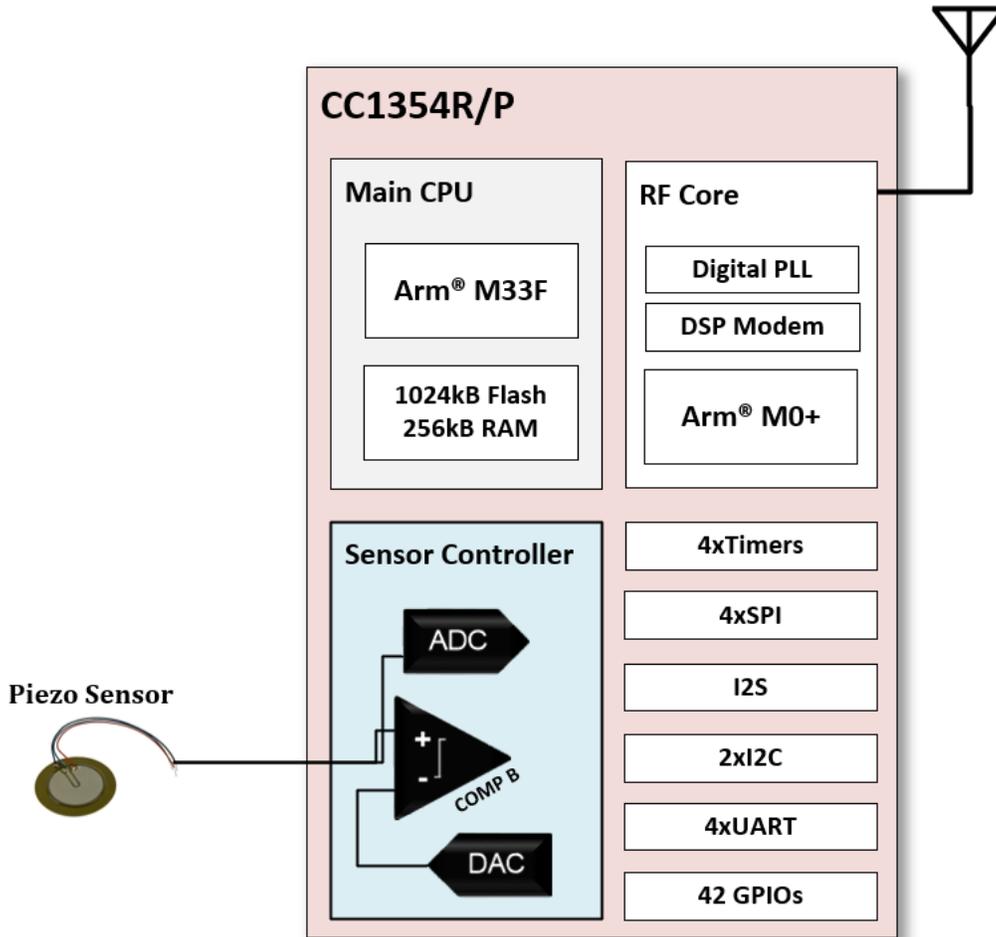


图 3-7. 高级玻璃破裂检测方框图

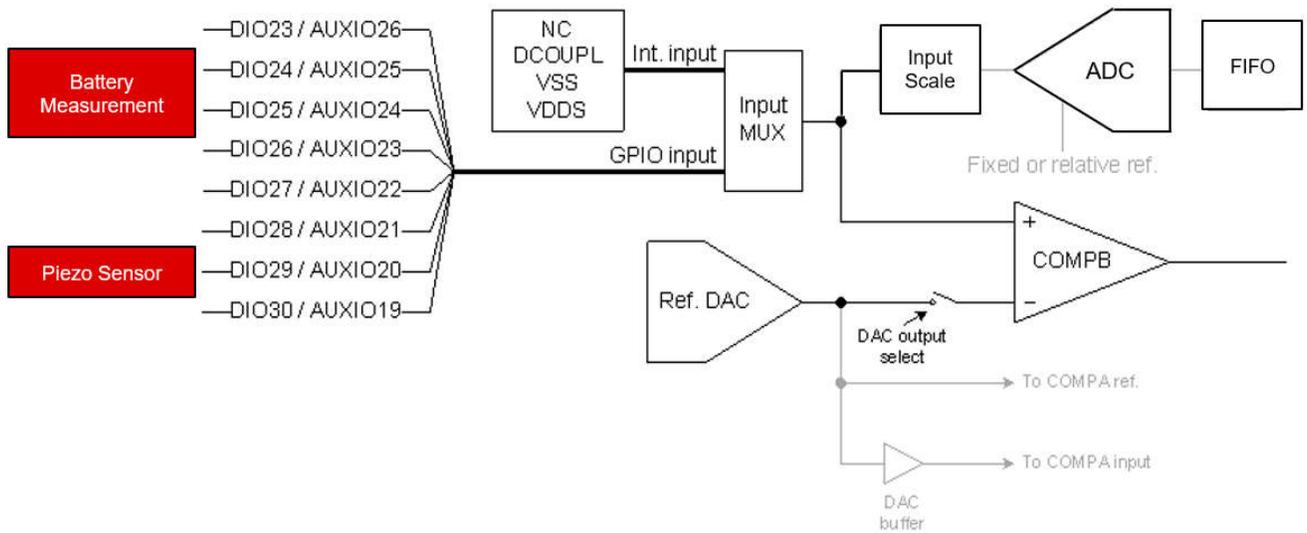


图 3-8. 传感器控制器外设的内部连接

图 3-9 是通过在前面提到的拓扑结构中连接压电传感器产生的。通过振动带有传感器控制器加压电式传感器的电路板，我们可以简单地绘制出 0.5V 至 5V（取决于压电式传感器负载电阻的大小）的不同峰值电压。然后，通过分析信号和能量，我们便可了解什么被视为破裂以及什么被视为正敲击或振动。

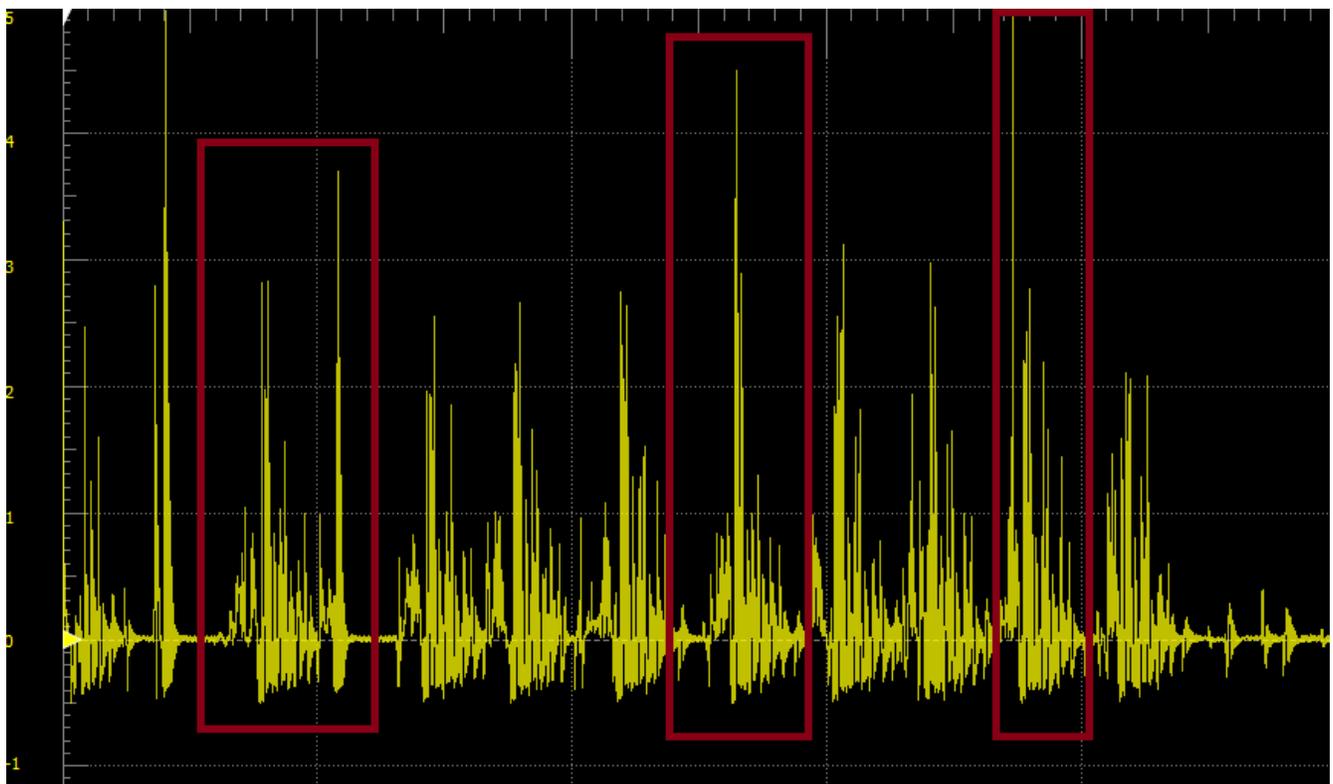


图 3-9. 压电式传感器信号

3.3 门窗传感器

簧片开关能够检测磁场。通常情况下，该开关为开路，但对该开关上施加磁场时，开关将闭合。簧片开关与上拉电阻器串联，因此 REED 信号可以低电平有效。簧片开关可用于多种应用；例如门窗传感器。图 3-10 显示对开门事件的检测。

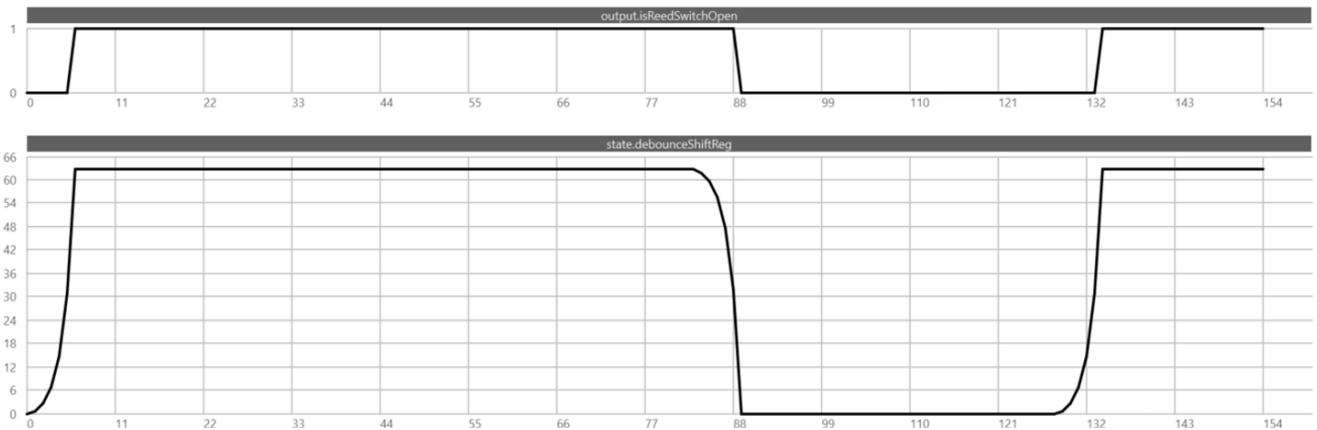


图 3-10. 检测开门事件

第一块磁铁在中心位置的最大检测距离约为 0.7 英寸，中心峰值的两侧各有一个次峰值。较深的红色阴影区域表示 Z 轴方向上的检测重叠，这在以下 3D 图中清晰显示：

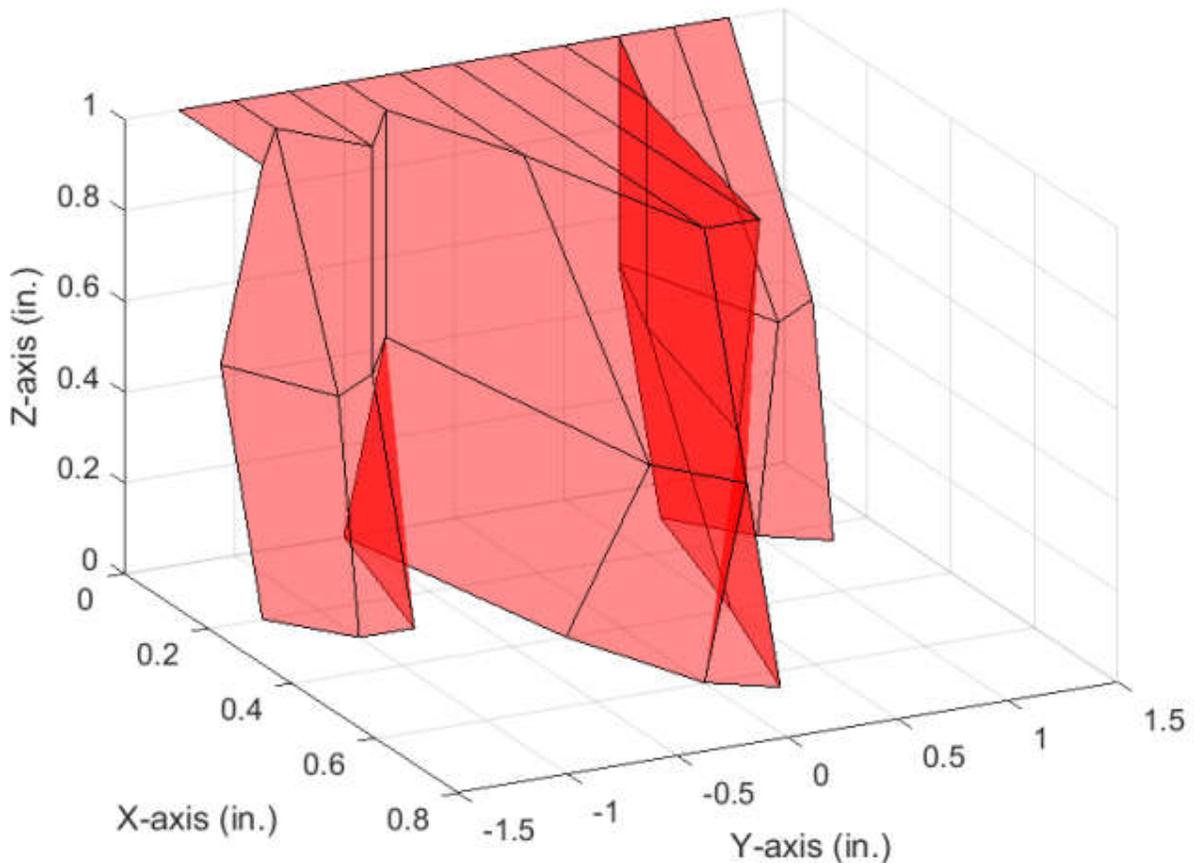


图 3-11. 簧片开关磁铁 1 检测场 (3D)

3.4 低功耗 ADC

如前所述，CC13x2 和 CC26x2 器件中包含 12 位模数转换器 (ADC) 外设块，其通常由传感器控制器使用，需要 SCLK_HF 信号，因此在低功耗 (2MHz) 模式下不可用。本节的目标是重点介绍一种替代设计，该设计使用在 2MHz 模式下运行的 CC13x2/4 和 CC26x2/4 传感器控制器创建 8 位逐次逼近 (SAR) 型低功耗 ADC。在此设计中，比较器和数模转换器 (DAC) 外设可在低功耗模式下使用。

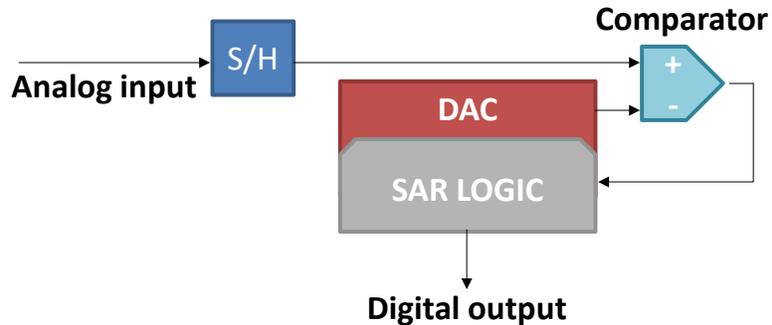


图 3-12. SAR ADC 方框图

SAR ADC 由四个主要块组成：

- 采样保持 (S/H) 块，在采样期间锁定模拟电压电平。
- SAR 逻辑块，使用传感器控制器复制。
- DAC，其输出由 SAR 逻辑块控制。
- 比较器，将模拟信号与 DAC 输出进行比较。

3.4.1 Sensor Controller Studio 中的代码实现

下面介绍 Sensor Controller Studio 中的 ADC 代码实现。

首先配置所需的数据结构和任务资源：

数据结构：

- cfg
 - 分辨率 = 8
- 输出
 - value
- state
 - isLarger
 - nextVal

任务资源：

- 模拟引脚
 - PIN_ANALOG
- COMPA
- 参考 DAC
- 基于 RTC 的执行调度
- 延迟插入

C 代码

初始化代码：

```
//Select SAR ADC input pin compaSelectGpioInput(AUXIO_A_PIN_ANALOG); fwScheduleTask(1);
```

执行代码：

```
//Enable comparator compaEnable(COMPA_PWRMODE_ANY);
state.nextVal = 0x0080; // 1/2 VDD
output.value = 0; //Set the reference DAC output to 1/2 VDD
refdacStartOutputOnCompaRef(state.nextVal); refdacEnable(REFDAC_PWRMODE_ANY,REFDAC_REF_VDDS);
U16 currBit = cfg.resolution;
U16 step = 64;
while(currBit > 0){ //Update DAC value refdacChangeOutputValue(state.nextVal); //wait for DAC to stabilize...
//An additional slight delay for DAC and comparator to stabilize...
refdacWaitForStableOutput();
fwDelayus(10); //Read comparator output compaGetOutput(state.isLarger); if(state.isLarger == 1){
state.nextVal += step; //Increase DAC value for next iteration
}
else{
state.nextVal -= step; //Decrease DAC value for next iteration
}
step >>= 1; //Divide step by 2
currBit -= 1;
//Update output value
output.value |= (state.isLarger << currBit);
}
refdacStopOutput();
compaSelectIntRef(COMPA_REF_VSS); //Disable peripherals
refdacDisable();
compaDisable();
fwScheduleTask(1);
```

终止代码：

```
//Disable peripherals
refdacDisable();
compaDisable();
```

3.4.2 测量

以下测量使用外部直流电源、EnergyTrace + CCS + 传感器控制器完成。

表 3-1. 软件和硬件 ADC 的不同电压测量值

电压 (V)	硬件 ADC	软件 ADC (8 位)
1.0	0.949	0.800
1.5	1.421	1.190
2.0	1.891	1.590
2.5	2.359	2.000
3.0	2.829	2.400

表 3-2. 软件和硬件 ADC 功耗

	电流消耗	单位	电池寿命 (CR123)
硬件 ADC (1Hz)	1.3	µA	13 年
硬件 ADC (100Hz)	16.5	µA	1 年零 4 个月
软件 ADC (1Hz)	0.8	µA	28 年零 6 个月

表 3-2. 软件和硬件 ADC 功耗 (续)

	电流消耗	单位	电池寿命 (CR123)
软件 ADC (100Hz)	4.8	μA	4 年零 8 个月

3.5 使用 BOOSTXL-ULPSENSE 的不同传感器读数

ULP Sense BoosterPack 是一款套件，其旨在轻松展示 CC13x2/CC26x2 RF SoC 中所含传感器控制器的超低功耗功能。可从中受益匪浅的应用示例包括：自动化门窗传感器（基于簧片开关）、电容式触控按钮和超低功耗加速度计处理（例如跌倒检测）。该电路板还包含一个模拟光传感器和一个电位器（可用于 ADC 测试案例）。

3.5.1 电容式触控

随着智能楼宇和物联网驱动型基础设施越来越受到重视，CapTouch 提供了直观的界面来控制照明、暖通空调和门禁控制等各种系统。例如，在**照明控制**应用中，CapTouch 面板允许用户通过简单的点击来调节亮度或安排照明变化。在**暖通空调系统**中，CapTouch 界面用触摸屏取代了传统的恒温器，其可控制温度、湿度和空气质量设置。在**门禁控制**应用中，它们也可以集成到触敏式键盘和生物识别设备中用于安全进入。

BOOSTXL-ULPSENSE 上的电容式触控示例在顶层使用一个较小的圆形覆铜区域，并在底层使用一个网格接地平面来形成一个小电容。传感器控制器能够利用 ISRC（电流源）和时间数字转换器 (TDC) 外设来测量电容。用指尖触摸时，系统的电容会发生变化。在不对电容式触控按钮进行任何输入以及在按钮激活的情况下，对电容式触控示例进行了测试。未检测到触摸按钮时，传感器控制器以 32Hz 的频率唤醒。如果检测到按钮按压操作，传感器控制器会将唤醒间隔增加到约 100Hz。传感器控制器还可以唤醒系统 CPU 并通知触摸事件。

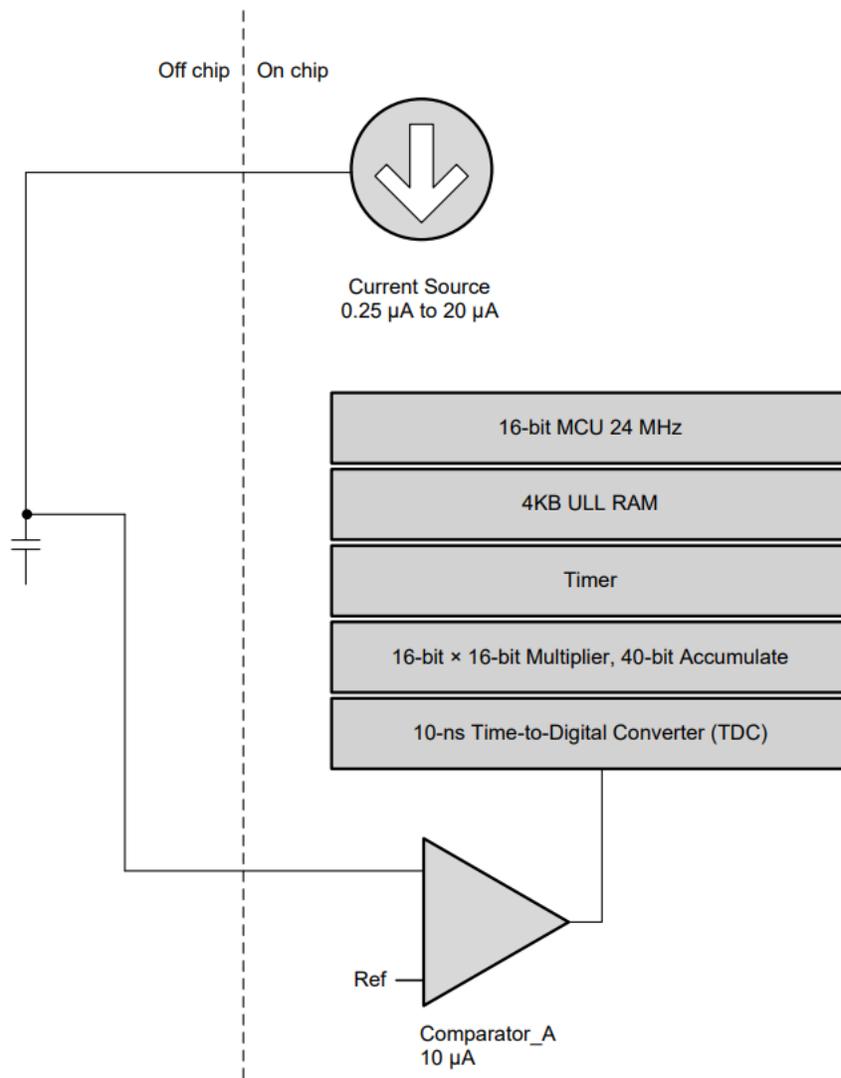


图 3-13. 电容式触控原理 (简化版)

表 3-3. 电容式触控功耗

	平均电流消耗	单位	电池寿命 (CR123)
未触摸 (32Hz)	8.5	μA	2 年零 8 个月
触摸 (约 100Hz)	84.1	μA	3 个月

以下各图显示了 EnergyTrace™ 在 1 秒测量时间内捕获的电流信号以及单次测量值。

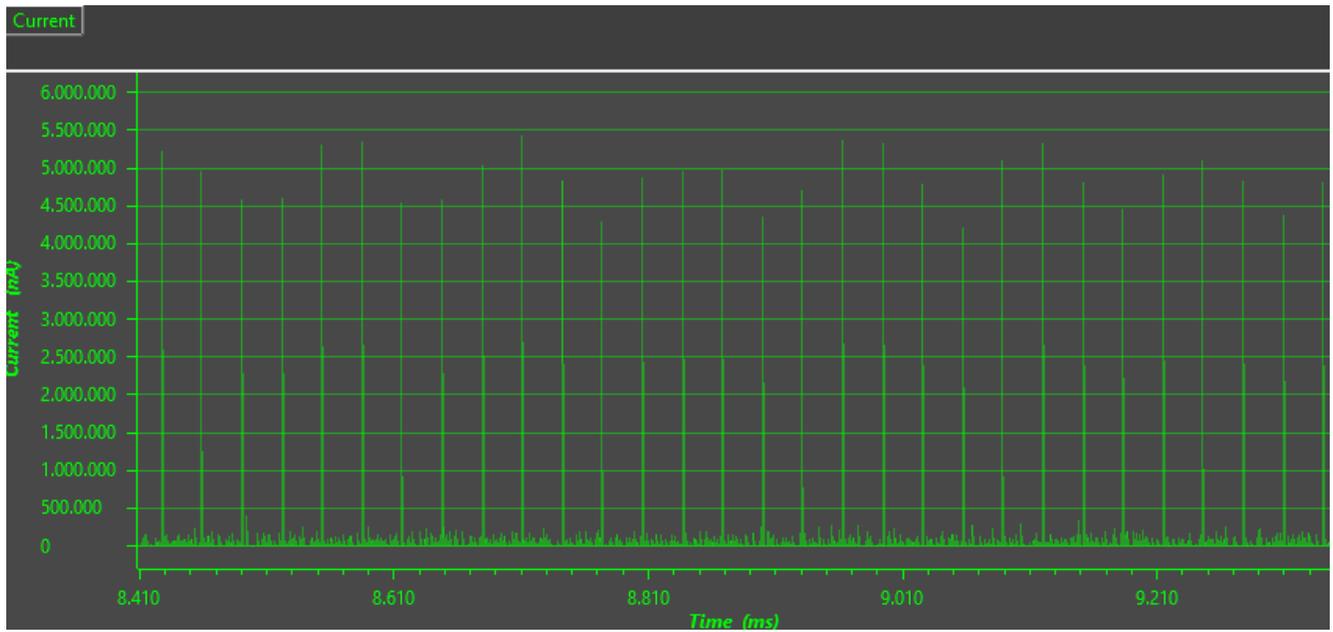


图 3-14. 电容式触控：未触摸时的测量值 — 1 秒

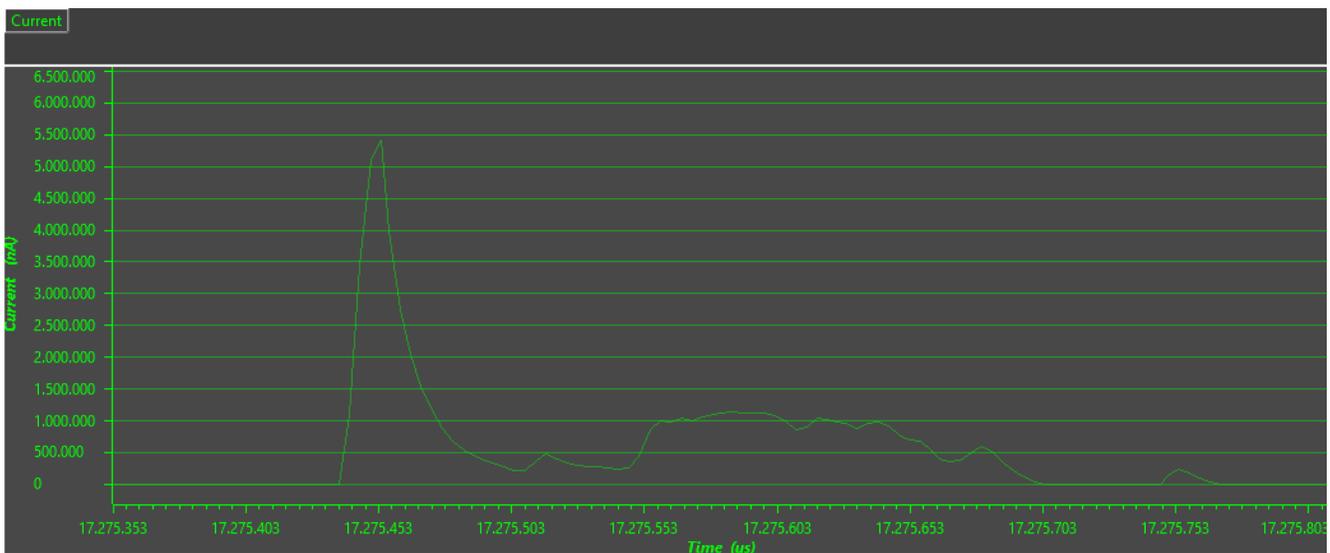


图 3-15. 电容式触控：未触摸时的测量值 — 一次测量

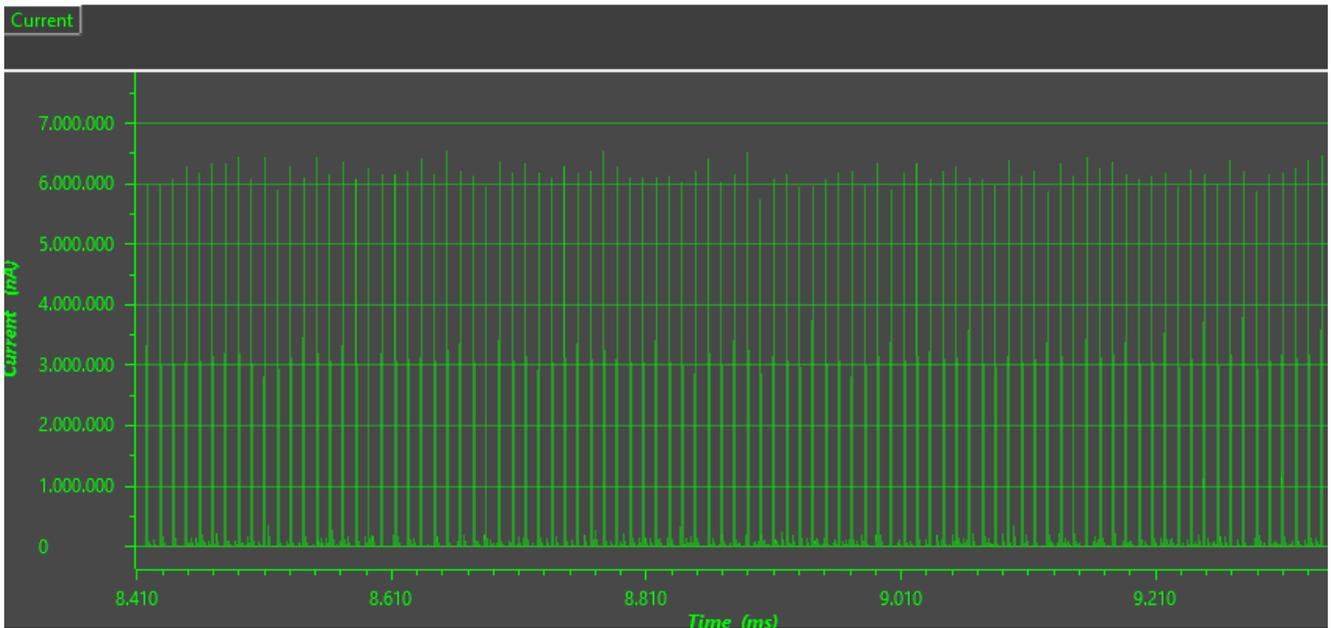


图 3-16. 电容式触控：触摸时的测量值 — 1 秒



图 3-17. 电容式触控：触摸时的测量值 — 一次测量

1 秒钟的测量窗口显示唤醒频率的增加（如前所述），每个峰值（一次测量）代表传感器控制器的一次唤醒。

3.5.2 模拟光传感器

模拟光传感器在楼宇自动化应用中发挥着至关重要的作用，它能提供环境光照水平的实时数据，从而实现对照明系统和其他环境设置的动态控制。例如，在**照明自动化**中，可以根据可用自然光量（日光采集）调节人工照明。在**暖通空调系统**中，模拟光传感器与其他传感器一起工作以检测日光强度，并根据季节和白天的情况相应地调节供暖或制冷系统。

模拟光传感器输出的电流取决于传感器受到的光照量。电流将馈入电阻器，从而产生可由 ADC 捕获的电压。传感器控制器使用外设计时器 2，在传感器控制器唤醒进行测量之前约 1ms 为模拟光传感器供电。如果光照水平高于阈值，传感器控制器可唤醒系统 CPU 进行测量处理。可通过 10Hz 的频率重复此过程。本示例在整个测试期间使用相同的环境光进行测试。

表 3-4. 模拟光传感器功耗

	平均电流消耗	单位	电池寿命 (CR123)
模拟光传感器 (10Hz)	9.1	μA	2 年零 6 个月

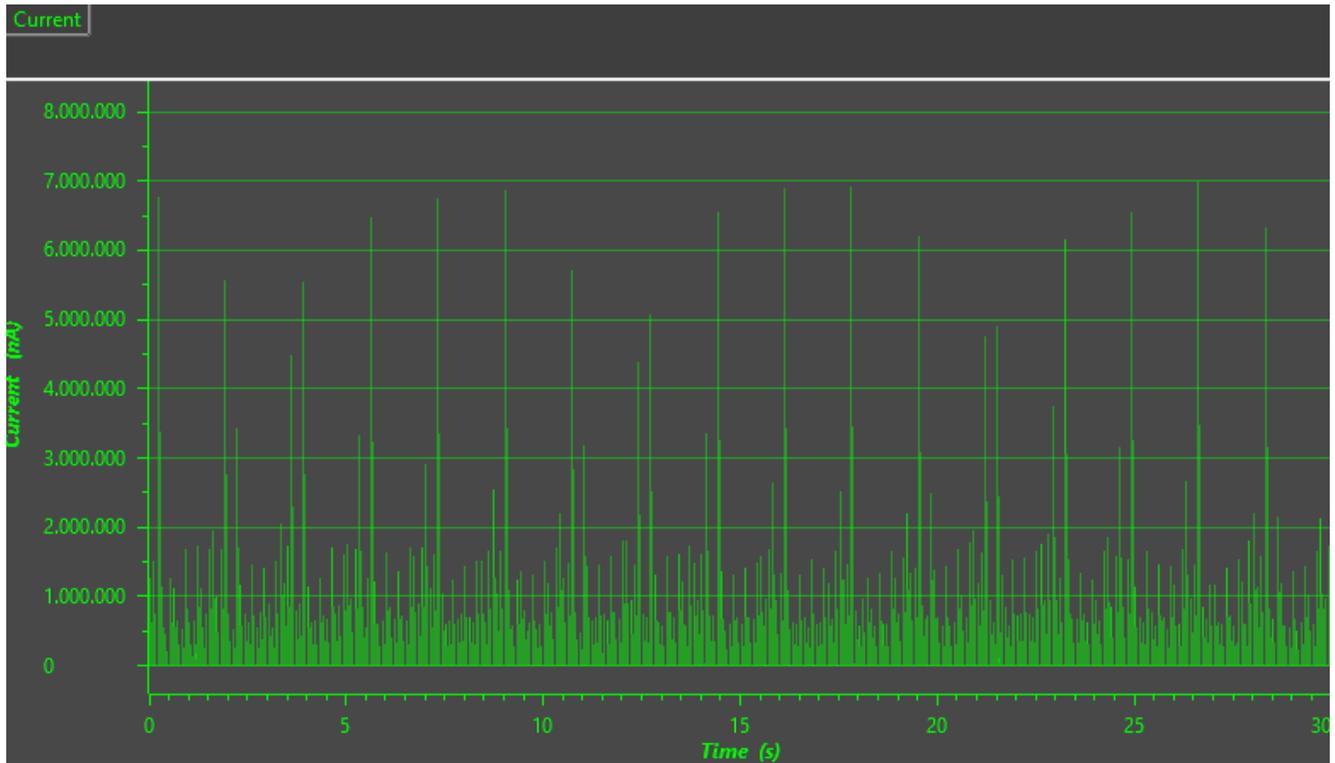


图 3-18. 模拟光传感器 : 30 秒

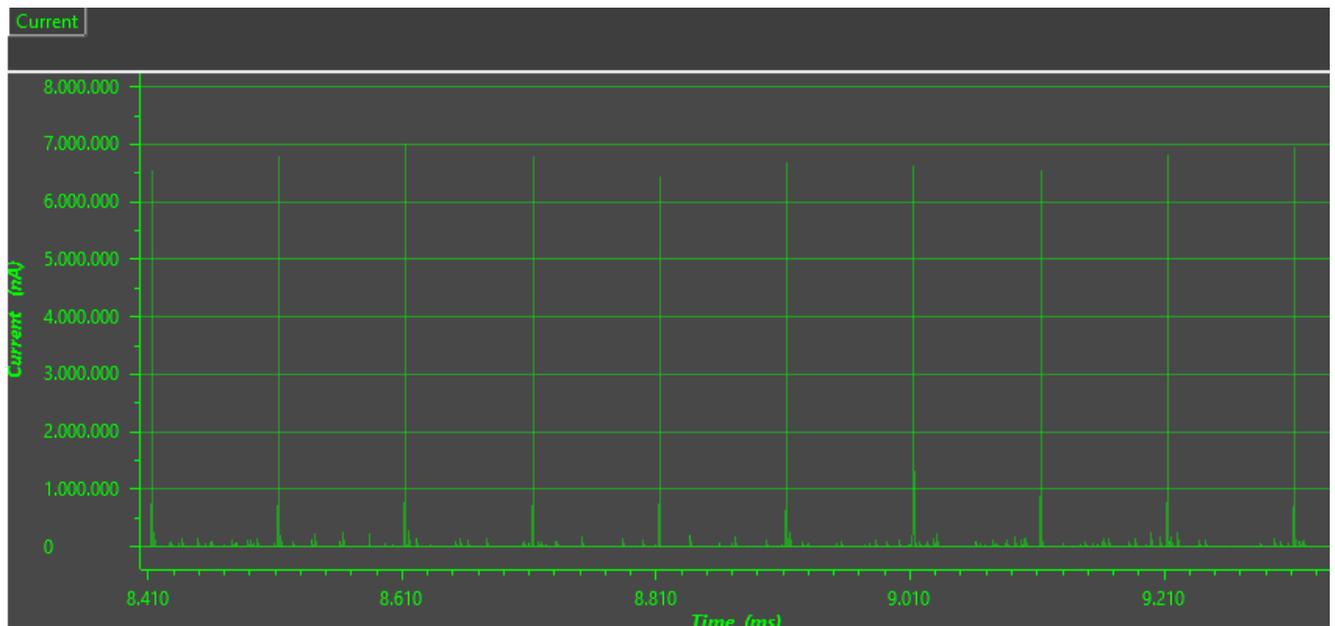


图 3-19. 模拟光传感器 : 2 秒

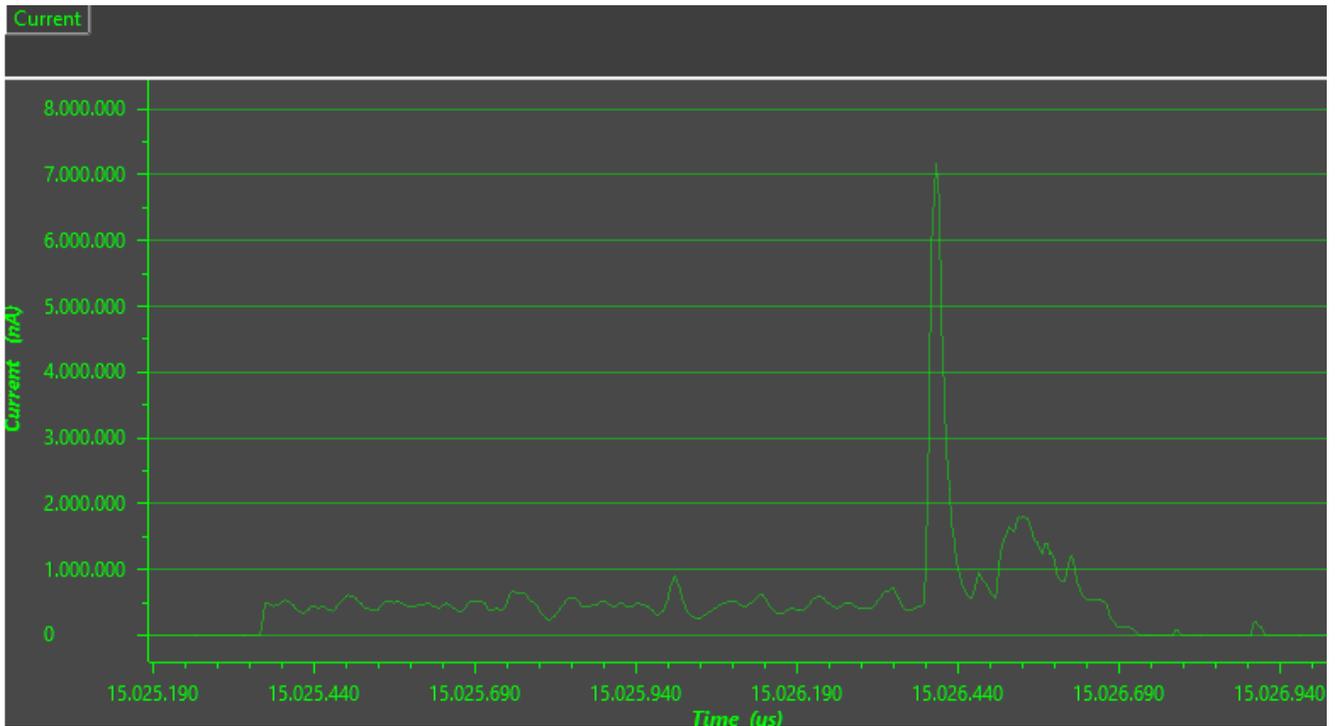


图 3-20. 模拟光传感器：一次测量

3.5.3 电位器 (0 至 200kΩ 范围)

电位器由电阻式元件和可拆卸刮片组成，刮片可沿元件移动调节电阻，从而提供可变的电压输出。例如，电位器设计用于光线控制的调光器开关，也可通过调节控制电路中的电阻来调节风扇速度、气流和温度设置。

BOOSTXL-ULPSENSE 上的电位器连接为电压分压器。该示例使用 COMPA 和基准 DAC 实现 SAR-ADC。电位器示例在电位器设置为中间位置时进行测试。传感器控制器能够以 25Hz 的速率唤醒，如果检测到变化，传感器控制器可以唤醒并通知系统 CPU。

表 3-5. 电位器功耗

	平均电流消耗	单位	电池寿命 (CR123)
电位器静止 (25Hz)	2.1	μA	10 年
电位器正在移动	14.7	μA	1 年零 6 个月

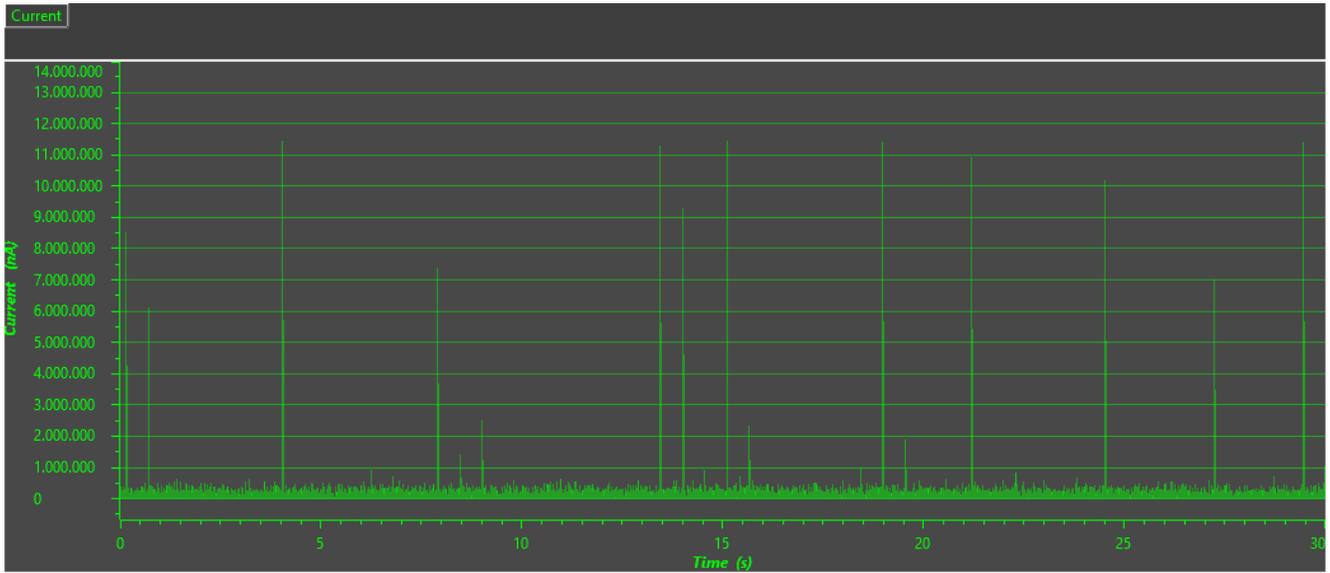


图 3-21. 电位器：中间 — 30 秒

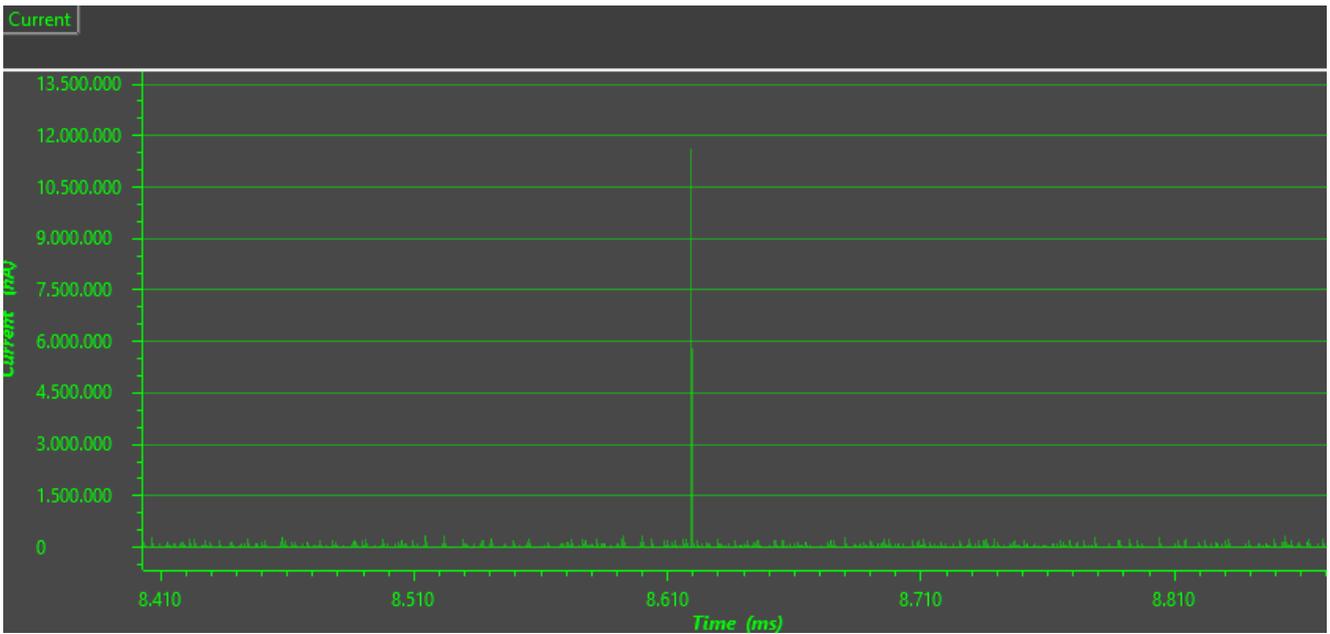


图 3-22. 电位器：中间 — 1 秒

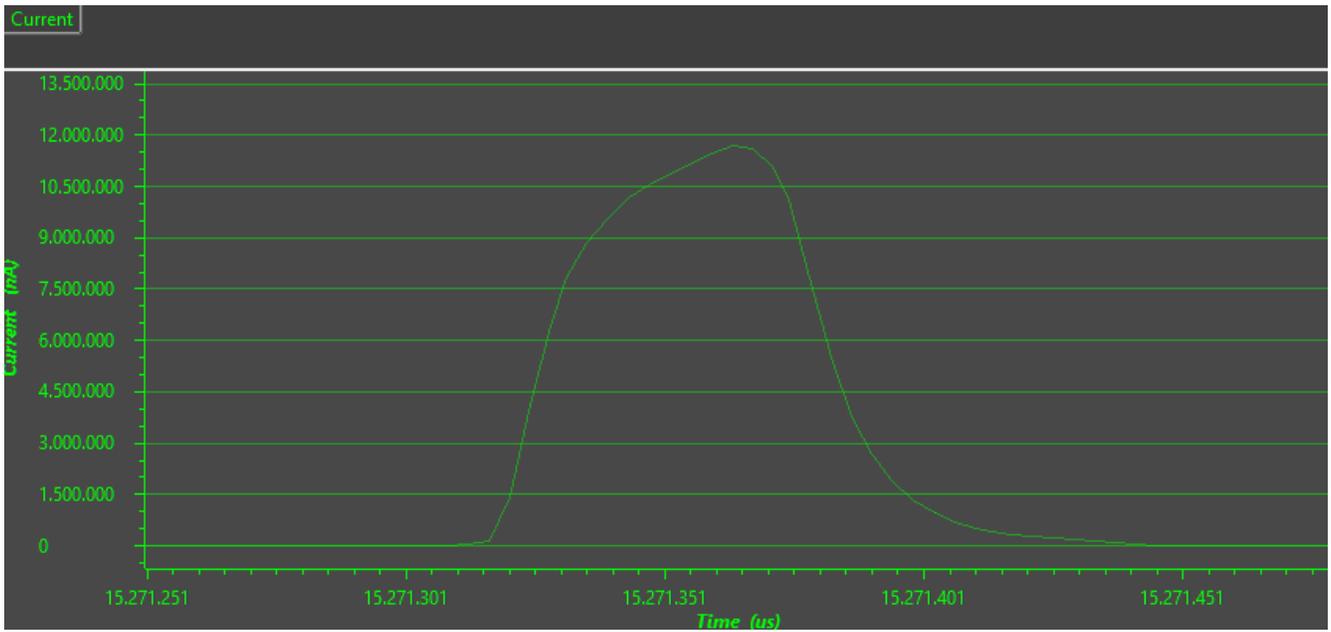


图 3-23. 电位器：中间 — 一次测量

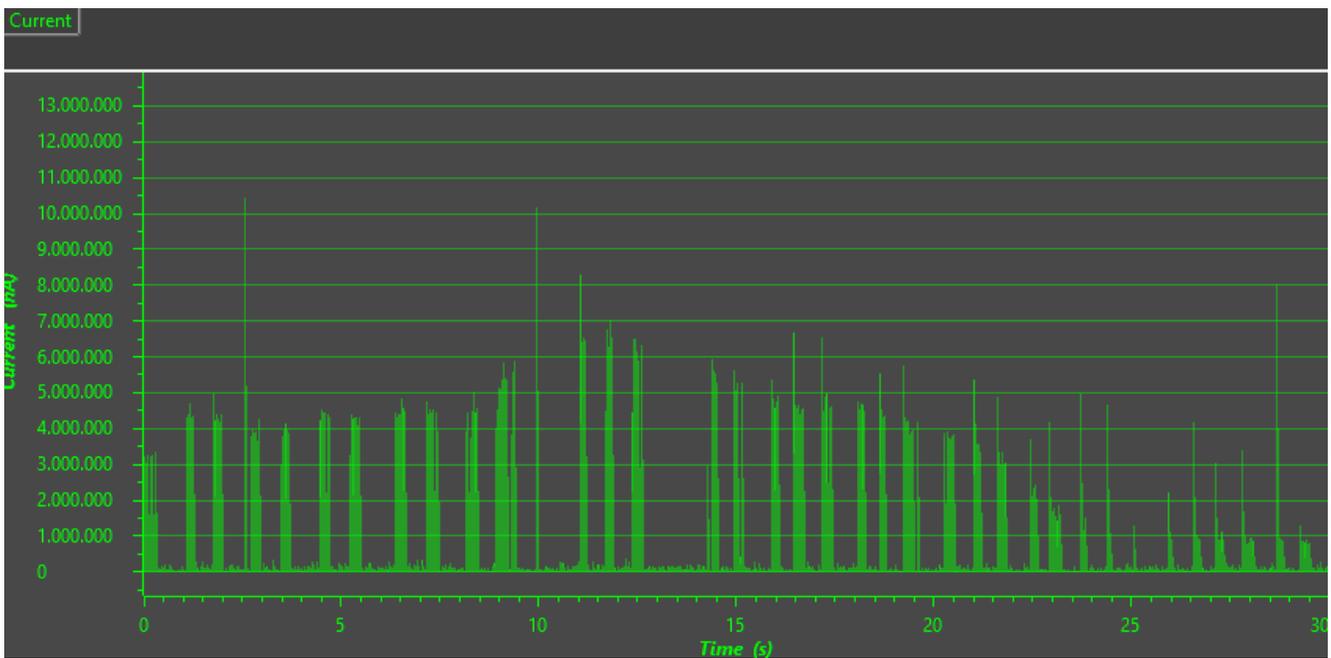


图 3-24. 电位器：移动中 — 30 秒

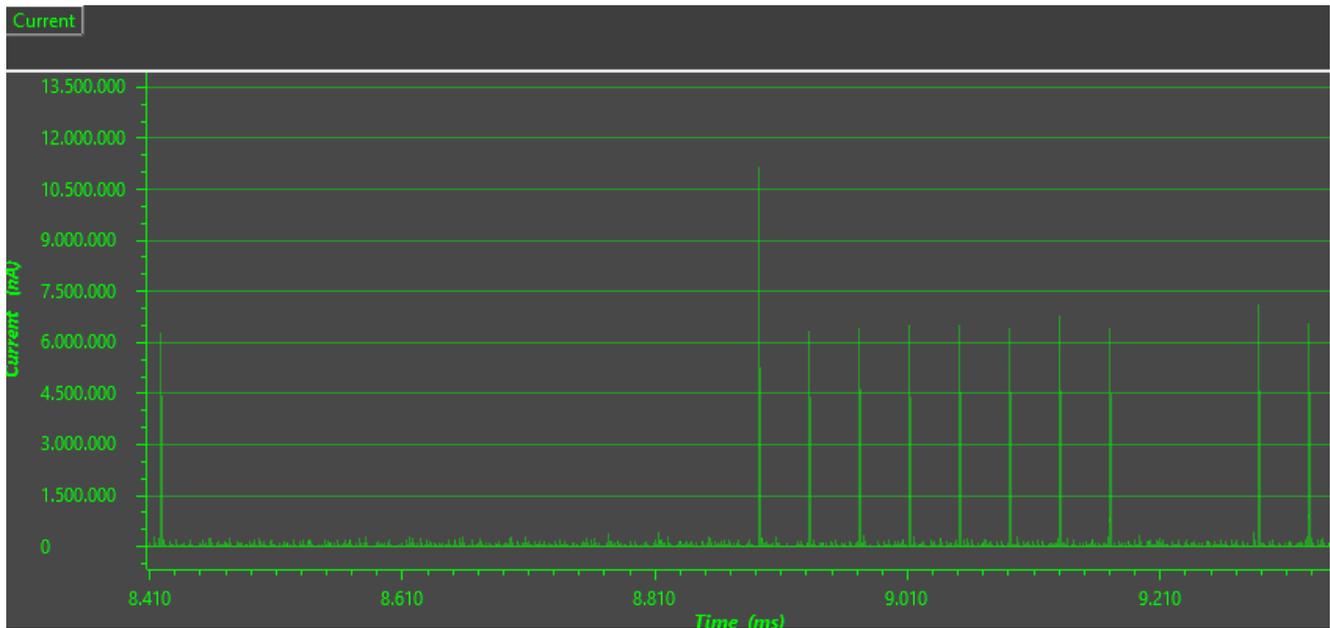


图 3-25. 电位器：移动中 — 1 秒

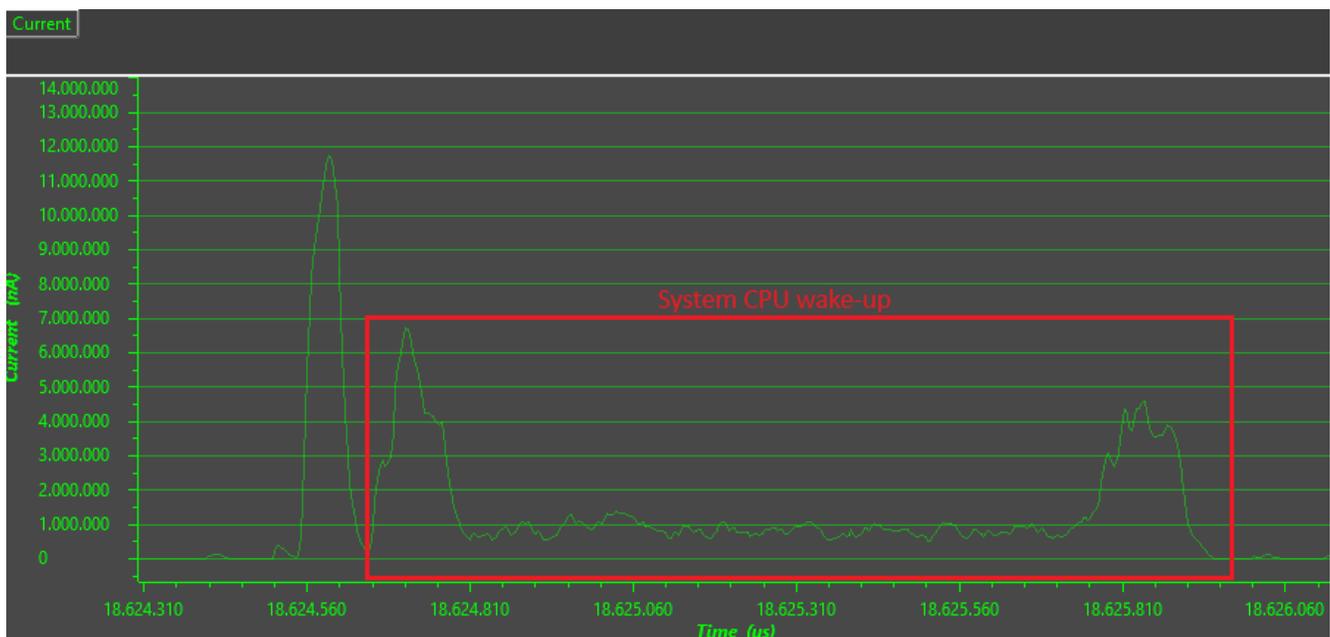


图 3-26. 电位器：移动中 — 一次测量

3.5.4 超低功耗 SPI 加速度计

超低功耗加速度计能够以最低的能耗检测运动、方向、振动甚至手势。这些传感器能够实现一系列应用，提高现代智能楼宇的效率、安全性和便利性。

在**占用检测**中，加速度计可用于感知房间内的运动或人员存在。由于其灵敏度较高，因此能够检测脚步声等细微的振动。这与照明和暖通空调系统的节能以及入侵检测等应用息息相关。

监测和诊断至关重要的系统也使用了加速度计。在电梯和暖通空调系统中，加速度计提供有关振动和运动模式的数据，因此可以检测到运行中的异常情况，并在需要时发出维护信号。

另外，我们还有一个用于加速度计数据处理的机器学习设计，利用 CC1352 平台的功能识别手势或运动模式（例如上下、左右或圆周运动）。您可以使用 Edge Impulse 平台轻松评估完整的设计，如需了解更多信息，请[从此处开始](#)。

BOOSTXL-UPLSENSE 配备了一个超低功耗加速度计。此传感器使用串行外设接口 (SPI) 与传感器控制器通信。加速度计能够以 100Hz 的频率向传感器控制器报告。一旦检测到变化超过某个阈值，传感器控制器就会唤醒系统 CPU，而系统 CPU 可点亮两个 LED 之一。

表 3-6. SPI 加速度计功耗

	平均电流消耗	单位	电池寿命 (CR123)
SPI 加速度计静止 (100Hz)	5.1	μA	4 年零 5 个月
SPI 加速度计移动 (100Hz)	8.5	μA	2 年零 8 个月

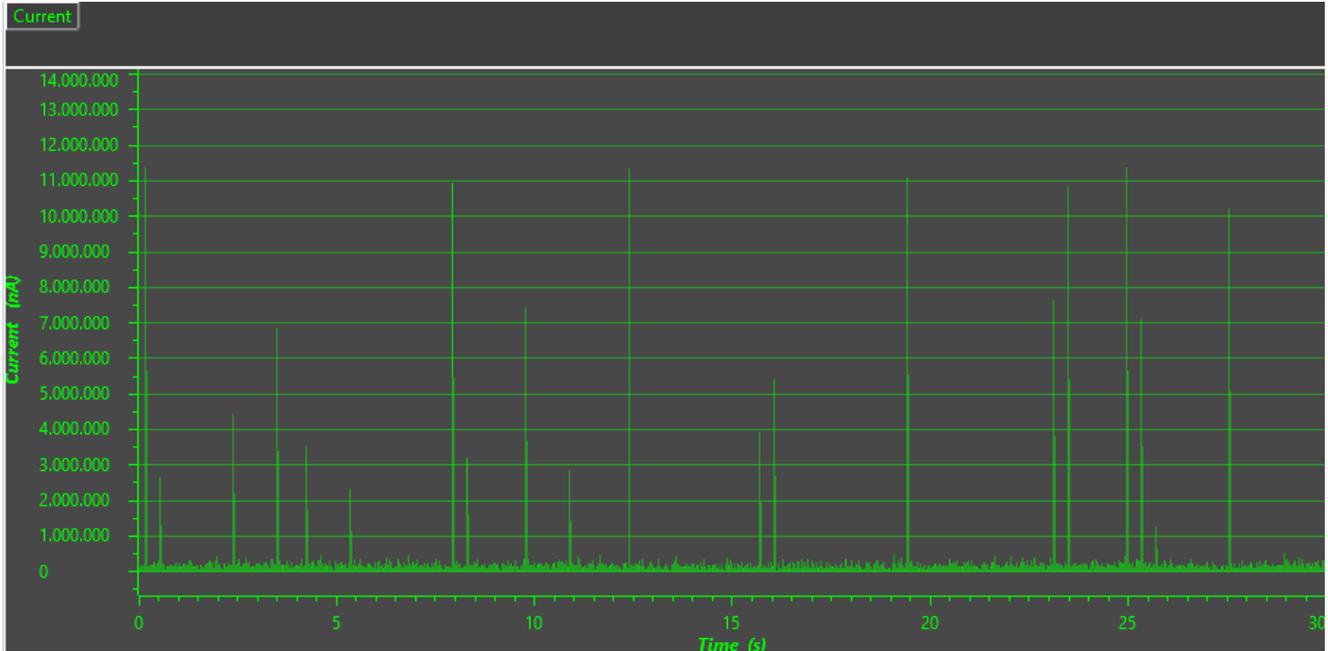


图 3-27. SPI 加速度计：稳定 — 30 秒

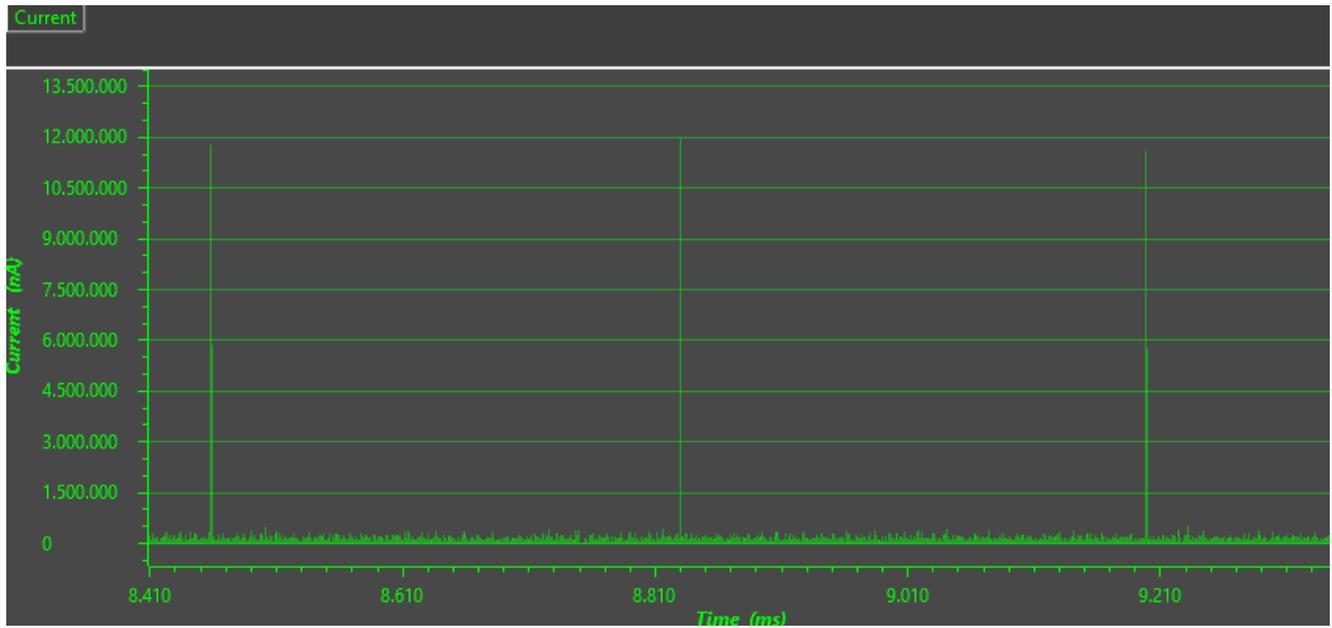


图 3-28. SPI 加速度计：稳定 — 1 秒

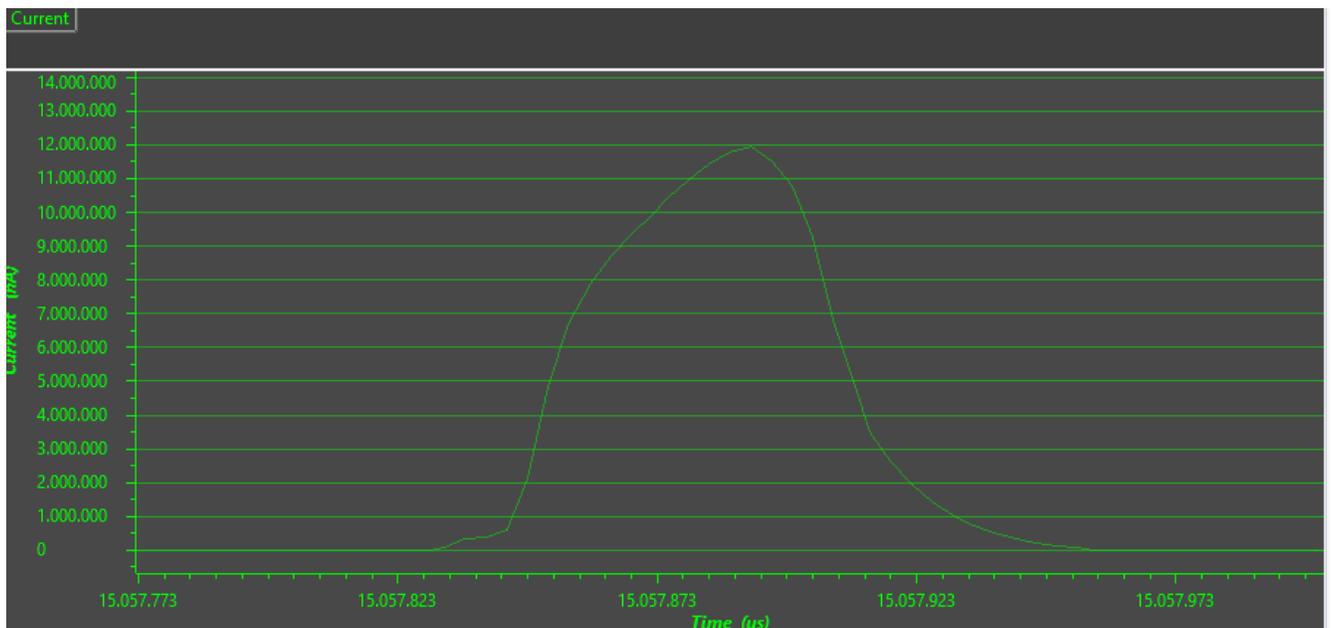


图 3-29. SPI 加速度计：稳定 — 一次测量

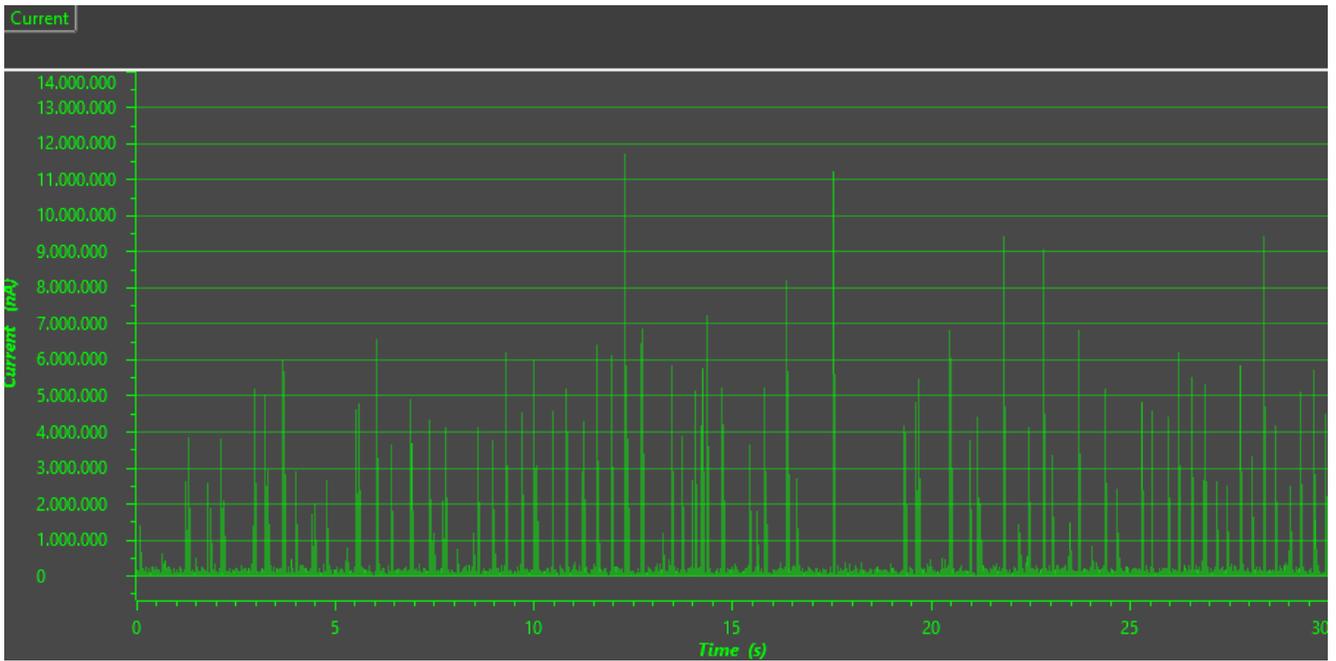


图 3-30. SPI 加速度计：移动中 — 一次测量

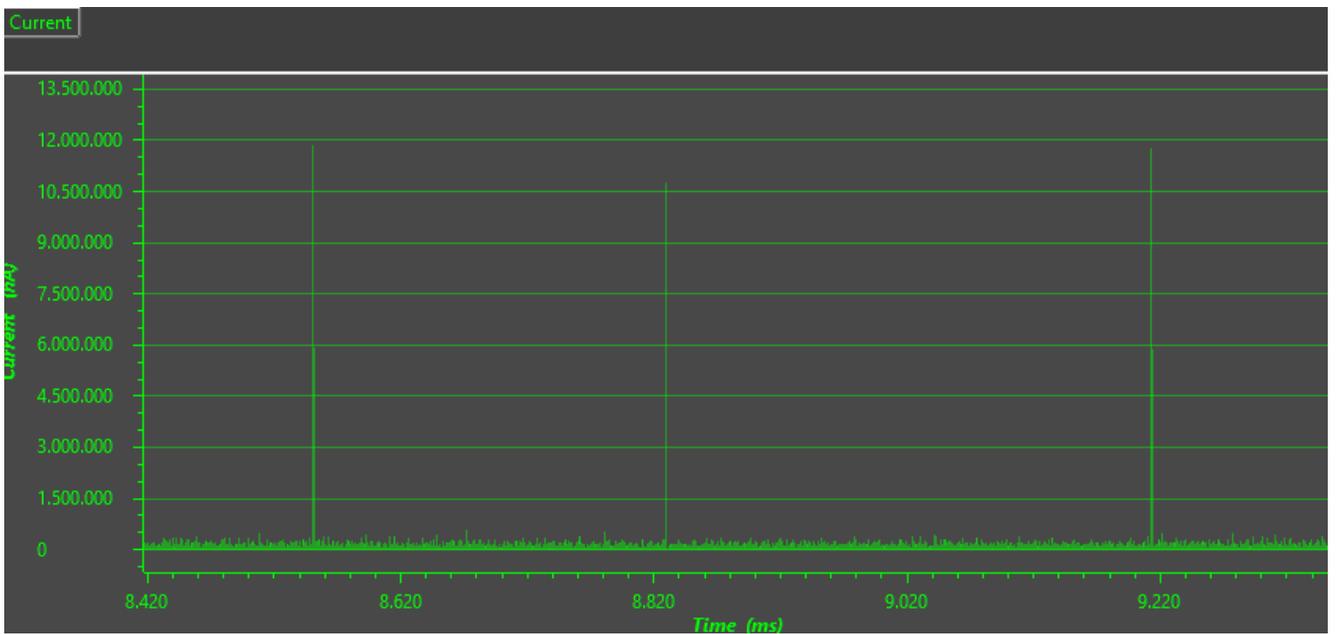


图 3-31. SPI 加速度计：移动中 — 一次测量

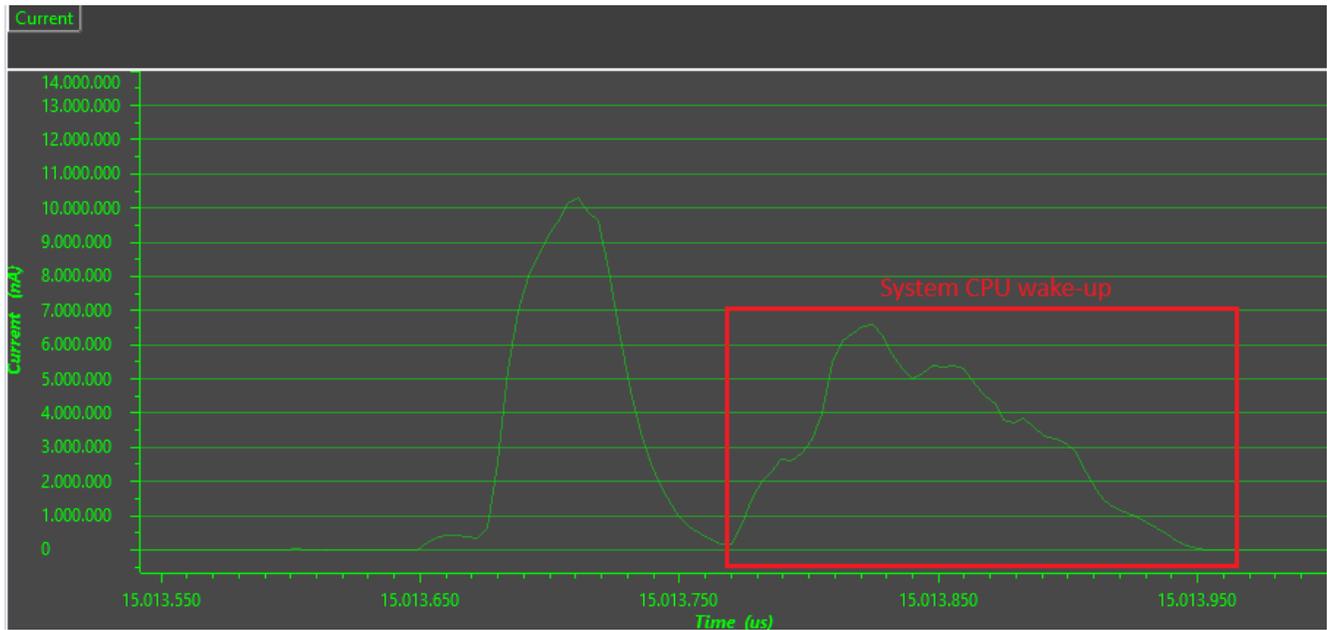


图 3-32. SPI 加速度计：移动中 — 一次测量

3.5.5 簧片开关

簧片开关测量与在节 3.3 中介绍的门窗传感器相关。

表 3-7. 簧片开关电源

	平均电流消耗	单位	电池寿命 (CR123)
不存在磁铁 (64Hz)	1.7	μA	13 年
存在磁铁 (64Hz)	8.5	μA	2 年零 7 个月

4 总结

表 4-1. 功耗和电池寿命

	平均电流消耗 (传感器控制器)	平均电流消耗 (Cortex M4F)	估计传感器控制器电池寿命 (CR123)	估计 ARM Cortex 电池寿命 (CR123)
PIR 运动检测 (20Hz)	7uA	大约 2mA	5 年零 1 个月	3 个月
玻璃破裂检测 (压电式传感器)	4uA	approximately 1mA	6 年	5 个月
门/窗 簧片开关读数	8.5uA	大约 1.5mA	5 年	3 个月
SPI 读数 (工作模式下为 4MHz)	3.7uA	243.8uA	6 年零 1 个月	2 个月零 3 天
SPI 读数 (低功耗模式下为 1MHz)	2.1uA	266.7uA	10 年零 9 个月	2 个月零 1 天
100Hz 引脚切换 (从待机模式切换到工作模式 (24MHz), 然后恢复待机模式)	14.5uA	大约 135uA	不适用 — 用例仅用于基准测试不同模式的功率差异	不适用 — 用例仅用于基准测试不同模式的功率差异
100Hz 引脚切换 (从待机模式切换到低功耗模式 (2MHz), 然后恢复待机模式)	0.8uA	大约 135uA	不适用 — 用例仅用于基准测试不同模式的功率差异	不适用 — 用例仅用于基准测试不同模式的功率差异

与系统 CPU 相比，传感器控制器的电流消耗要低得多。

本应用手册中展示的不同示例使用了许多不同的技术来降低功耗。以下所示是实现低功耗的一些方法：

- 尽可能用低功耗模式代替工作模式
- 尽可能用低功耗模式下的计时器 2 代替 TDC
- 尽可能用 SPI 代替 I2C
- 计时器 2 以 32kHz 频率为传感器上电，然后唤醒传感器控制器 (当传感器准备就绪时)
- 频率为 2MHz/32kHz 的计时器 2 作为状态 LED 的脉宽调制器 (PWM)
- 预处理传感器数据以检测相关活动，仅在需要时唤醒系统 CPU 应用。
- 通用输入/输出 (GPIO) 事件处理器，用于处理数字传感器的中断，并为定期产生中断的传感器设置超时。
- 更大限度减少了通过 I2C/SPI 进行的通信，一次操作即可读取多个外部器件寄存器。
- 测量后 (数据处理前) 尽快禁用外设。
- 传感器数据表明无活动时，降低传感器轮询频率。
- 低功耗模式下的基准 DAC 和 COMPA 可用作低精度 ADC。

5 参考资料

- 德州仪器 (TI), [使用 CC13x2 和 CC26x2 传感器控制器的传感器时序](#), 应用手册。
- 德州仪器 (TI), [适用于 CC13x2 和 CC26x2 的低功耗 ADC 解决方案](#), 应用手册。
- 德州仪器 (TI), [使用 MSPM0 进行 PIR 运动检测](#), 应用手册。

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司