



Shuang Feng, Milen Stefanov

摘要

一天一次，一分钟一次，甚至每秒多次：设计人员必须针对其使用场景找到适合无线传感器标签或数据记录器的最优超低功耗架构。传统方法利用整个系统的电源占空比或 MCU 和传感器的待机模式来延长电池寿命。

德州仪器 (TI) 提供了本报告中介绍的一种新颖实现方案，该实现方案通过具有快速唤醒功能的可编程、自主超低功耗传感器控制器 CPU 实现高级传感。这种专用外设实现了与常规 MCU 待机方法相同或更低的功耗，同时从主 MCU 卸载传感器代码并将传感器数据存储到其超低泄漏 (ULL) 始终开启的 SRAM 存储器内的几 KB 大型缓冲区中。收集一系列传感器数据并立即传输，尽可能降低无线协议开销，并且与多个较短的数据包相比，每个数据包都具有冗余开销字节，从而进一步节省电能。

传感器控制器 CPU 可在 SimpleLink™ 高性能无线 MCU 中找到，例如 CC1352P 器件。

内容

1 引言.....	2
2 系统概述.....	3
3 系统运行.....	4
4 测试和验证.....	9
5 总结.....	12
6 参考文献.....	13
7 修订历史记录.....	13

插图清单

图 1-1. 传感器标签在冷链中的应用.....	2
图 1-2. 用于监测温度和湿度的 SensorTag 或数据记录器解决方案，具有超低功耗 (I2C 总线和 GPIO 引脚).....	3
图 2-1. 连接到 HDC2010，电源线是 P-MOSFET 门控 (低有效).....	4
图 3-1. Sensor Controller Studio 下的温度和湿度曲线.....	5
图 3-2. 传感器控制器引擎固件的流程图.....	5
图 3-3. 四个字节读取模式的比较 (多字节读取位于顶部).....	6
图 3-4. Sensor Controller Studio 中的温度和湿度数据缓冲器.....	7
图 3-5. 完成对 HDC2010 的初始 SCE 访问 (I2C 上拉、配置、数据读出和断电).....	7
图 3-6. 使用 SCE 定期读取传感器，时长为 198µs.....	8
图 4-1. +1dBm 输出的电流曲线，包含 SCE 和 Arm® Cortex®-M4F 活动.....	9
图 4-2. 待机、传感器控制器和占空比方法的超低功耗决策树.....	10
图 4-3. 经过简化的电流曲线 (比较).....	11
图 4-4. 系统解决方案比较 (每分钟读取一次传感器和传输一次数据).....	11
图 4-5. 由于数据缓冲和避免无谓的无线开销而实现的功效.....	12

表格清单

表 2-1. HDC2010 I ² C 从器件地址.....	3
表 3-1. 读取单字节.....	6
表 3-2. 读取多字节.....	6
表 4-1. 三种超低功耗架构的对比.....	9

商标

SimpleLink™, SmartRF™, BoosterPack™, and EnergyTrace™ are trademarks of Texas Instruments. Arm®, Cortex®, are registered trademarks of Arm Limited.

所有商标均为其各自所有者的财产。

1 引言

用于资产跟踪的无线传感器标签可监测温度、湿度、环境光或其他环境参数，并且通常通过低功耗短距离无线网络将数据传输到网关。若要对全球冷链提供支持则需要正确处理对时间和温度敏感的货物，并能够远程控制具有类似运输要求的食品、药品和其他货物。

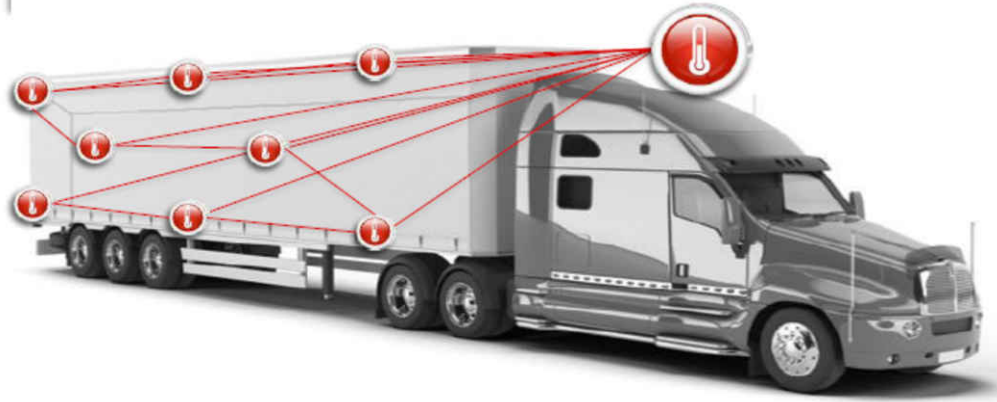


图 1-1. 传感器标签在冷链中的应用

在图 1-1 所示的应用中，关键功能是读取各种传感器以及在低于 1GHz、2.4GHz 或这两个射频频段中提供低功耗无线连接。超低功耗双频无线 MCU（例如 CC1352P 器件）将通用微控制器、专用传感器控制器引擎外设和无线连接的功能集成到单个器件中。

低功耗传感应用有两种传统解决方案：常见的是标准待机方法，在这种方法中，MCU 会始终维护实时时钟 (RTC) 及其 RAM 内容，在不访问传感器器件时以低功耗模式运行。

第二种解决方案是采用占空比方法，在这种方法中，会在两个传感器读取周期之间打开和关闭整个系统电源，MCU 会丢失其 RAM 和 RTC 内容，而纳米计时器器件跟踪占空比周期。这种方法可在关闭（断电）状态下实现超低功耗，通常在 50nA 的范围内，如可实现 10 年以上纽扣电池寿命且适用于低于 1GHz 的星形网络的湿度和温度传感器节点中所述。

此报告介绍了一种全新解决方案，被称为传感器控制器方法，这种方法是标准和占空比方法的组合，具有下列功能：

1. CC1352P MCU 在待机模式下运行，直接由电池供电。因此不需要任何外部器件，例如纳米计时器和超低泄漏负载开关。
2. 传感器控制器引擎 (SCE) 外设实现了与传感器之间的 I2C 通信和数据传输（而在其他两种解决方案中，此任务是由 Arm® Cortex®-M4F MCU 内核处理的）。
3. 使用一个通用 IO 引脚，当传感器关闭且未进行任何 I2C 通信时，SCE 会关断 SCL 和 SDA 线的两个上拉电阻器。
4. 在待机模式下，SCE 会将传感器数据存储存储在用户可配置的缓冲区中（最长 3K 字节），并一次性将这些缓冲区数据块传递到 Arm Cortex-M4F 内核。
5. Arm Cortex-M4F 内核根据所使用的数据协议创建一个具有用户可配置有效载荷长度的数据包，并以无线方式发送。
6. 可选：使用一个通用 IO 引脚，SCE 为传感器器件通电或断电（具体取决于传感器读取的占空比）。

图 1-2 所示为传感器控制器（或基于 SCE）方法的方框图。请注意，尽管此图看起来与常用的待机方法相同，但专用 SCE 超低功耗外设自主处理 I2C 协议，因此需要考虑硬件差异。两个专用 IO 引脚用于为传感器（此处为 HDC2010 器件）和 I2C 上拉电阻器提供电源。这些由 SCE 控制的 IO 引脚可支持传感器电源引脚（如果需要）和电阻器的占空比，从而实现真正优化的低功耗解决方案。

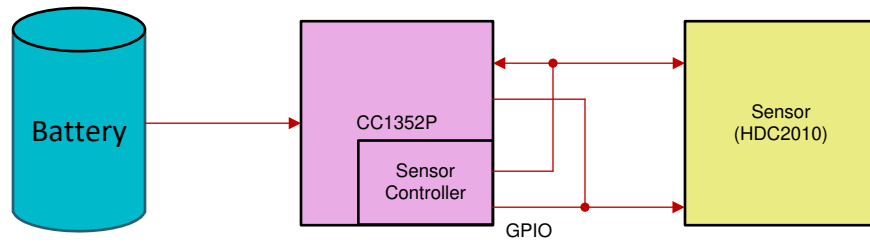


图 1-2. 用于监测温度和湿度的 SensorTag 或数据记录器解决方案，具有超低功耗 (I2C 总线和 GPIO 引脚)

本报告介绍了一种解决方案，其中 CC1352P 器件利用其传感器控制器引擎配置 HDC2010 以定期读取温度和湿度数据。该解决方案可用作温度和湿度监测或数据记录器器件的开发平台。如果这仅取决于 HDC2010 或 HDC2080 器件的功能，则温度和湿度的监测周期是可配置的，跨度从毫秒到秒不等，因为 CC1352P MCU 会持续供电并使用实时时钟运行 TI RTOS。根据应用的要求，可在任何由用户定义的时间段之后进行唤醒和传感器读数。

2 系统概述

2.1 软件

以下软件用于创建和使用 CC1352P MCU 中的传感器控制器 CPU (在进行本文档中所述的性能测试时会用到这些修订号)。

- [Sensor Controller Studio \(SCS\) v2.5.0](#)
- [Code Composer Studio \(CCS\) 9.1.0](#)
- [SDK3.20.00.20](#) (适用于 CC1352P 器件的 SDK，代码示例 rfPacketTX)
- [SWRC367 固件](#)

首先，通过读取制造商和供应商 ID 预定义值以及监控 I2C 总线事务开发和验证了用于连接 HDC2010 传感器的传感器控制器代码，如图 3-3 所示。

接下来，传感器控制器 I2C 总线功能被集成到 SDK 中现有的 rfPacketTX 代码示例中，重点是超低功耗和周期性射频传输。所有数据包均使用 SmartRF™ Studio 下的 CC1350 LaunchPad 以“Packet RX”模式进行监控，从而捕获和可视化数据内容。

最后，基于 rfPacketTX 开发了第二个 CCS 项目，现在使用主 MCU 来通过 I2C 总线控制 HDC2010 器件，而不使用 SCE 外设。

2.2 硬件

对于测量，使用了以下硬件 (在 [TI.com](#) 上提供)：

- [CC1352P LaunchPad](#)
- [BOOSTXL-BASSENSORS](#) (包括 HDC2010 传感器、TMP116 传感器和其他传感器)

硬件修改：移除了 CC1352P LaunchPad 上的 I2C 总线上拉电阻器，因为它们与 BOOSTXL-BASSENSORS 上的 I2C 上拉电阻器平行。BoosterPack™ 插件模块上的上拉电阻器连接到 CC1352P DIO 引脚 (额外接线)；连接到 CC1352P LaunchPad 的 +3.3V 引脚的覆铜线迹在 PCB 上被切断。在这种功率优化设计中，新布线允许仅在数据传输时为 I2C 上拉电阻器供电。

BOOSTXL-BASSENSOR 中 HDC2010 器件的 ADDR 引脚连接到 GND，因此其从 i2C 地址为 **0x40** (参见图 2-1)。用在 Sensor Controller Studio 中时，0x40 地址变为 **0x80 (写)** 或 **0x81 (读)**，Sensor Controller Studio 会在 7 位地址后插入读/写位作为最后一位，如表 2-1 所示。

表 2-1. HDC2010 I²C 从器件地址

ADDR	地址 (7 位地址)	BOOSTXL-BASSENSORS (8 位地址)
GND	1000000	0x80(W)/0x81(R)
VDD	1000001	0x82(W)/0x83(R)

NOTE

HDC_V+ 引脚使用低电平有效电压来控制 P-MOSFET，从而对 HDC2010 传感器的电源进行门控。

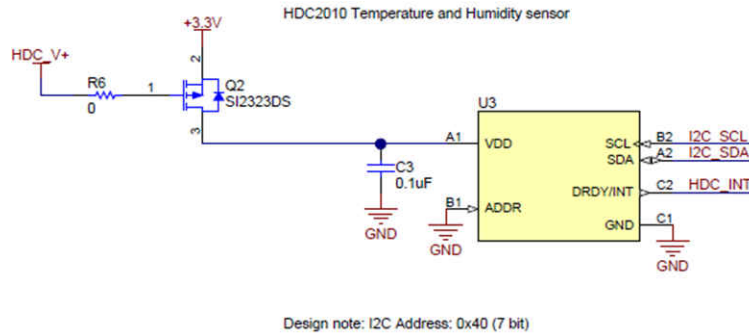


图 2-1. 连接到 HDC2010，电源线是 P-MOSFET 门控（低有效）

上电后，HDC2010 器件的采样频率可配置为从每两分钟一次采样到每秒五次采样。必须计算 HDC2010 传感器提供的原始值或将其转换为人类可读的温度和湿度格式，请参见方程式 1 和方程式 2。

$$\text{Temperature (}^{\circ}\text{C)} = \left(\frac{\text{TEMPERATURE [15:0]}}{2^{16}} \right) \times 165 - 40 \quad (1)$$

$$\text{Humidity (\%RH)} = \left(\frac{\text{HUMIDITY [15:0]}}{2^{16}} \right) \times 100 \quad (2)$$

出于演示目的，在传感器控制器代码示例中采用了这两个公式；但是，额外的代码减少了用于存储传感器数据的可用缓冲存储器，并对无线传输的数据量产生负面影响。

为实现最低功耗，TI 建议通过无线方式读取和传输原始 16 位湿度和 16 位温度传感器值（在创建具有所需有效载荷长度的数据包后），而转换为人类可读格式方程式 1 和方程式 2 则可在接收端或数据显示节点上完成。

3 系统运行

3.1 HDC2010 传感器读取

在调试模式下使用 Sensor Controller Studio，测试 HDC2010 功能的一种简单方法是在 HDC2010 传感器前呼气，这会导致湿度值发生非常大的变化，而温度几乎没有升高。将手指放在 HDC2010 传感器上几秒钟，使温度升至约 30.8°C（所有测试均在室温下的办公环境中进行）。x 轴表示时间，单位是秒。y 轴表示湿度和温度值。使用节 2.2 中的公式计算实际值。

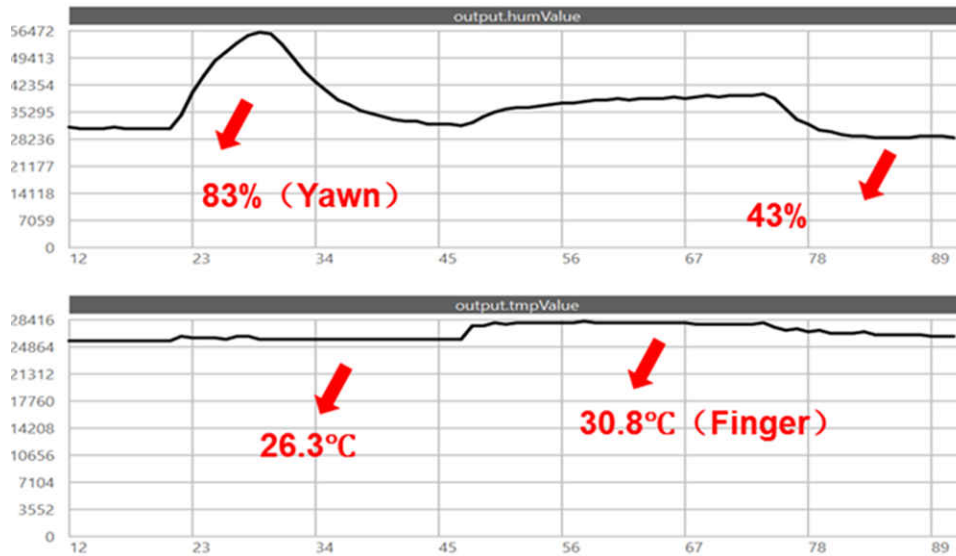


图 3-1. Sensor Controller Studio 下的温度和湿度曲线

图 3-2 中的软件流程图描述了 HDC2010 传感器配置。首先，将 DIO 引脚设置为低电平状态，打开 P_MOSFET 器件并为 HDC2010 传感器通电，然后在最小启动延时时间后初始化几个配置寄存器。这些寄存器值设置传感器采样频率（进行测量的频率）和数据分辨率，此处设置了 14 位最高精度。最后，将 0x0F 寄存器设置为 0x01 以开始测量。仅 1.24ms 后，系统会读出温度和湿度值并将其存储到传感器控制器 ULL RAM 存储器内的数据缓冲区中。

固件的最终版本始终使 HDC2010 传感器保持供电状态，因为它在两次温度和湿度测量之间的待机模式下通常消耗 50nA。还实施并测试了 HDC2010 传感器的完全断电模式。在这种情况下，每个 HDC2010 读数都需要完整的初始化周期和延迟来进行测量。根据传感器读取周期，尤其是在不活动时间周期很长的情况下，断电可能会更省电。对于此报告中分析过的 60 秒期间，待机模式实现了更低的功耗。

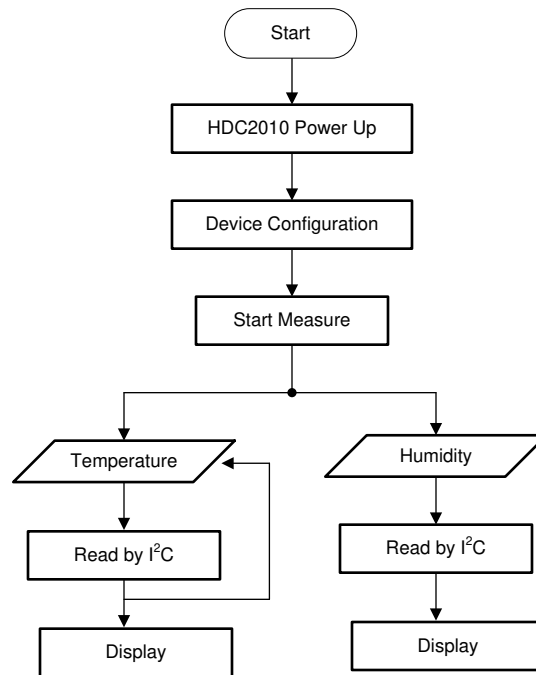


图 3-2. 传感器控制器引擎固件的流程图

3.2 针对低功耗的 I2C 协议和数据缓冲

在本设计中，HDC2010 传感器的多字节读取功能用于减少 I2C 通信时间；表 3-2 中介绍了单字节模式和多字节模式的区别。多字节模式只需要一个带有连续 ACK 请求的读命令即可读出从地址 0x00 开始的 4 个字节。这有助于减小代码大小，缩短 I2C 通信活动时间和降低整体系统功耗。

表 3-1. 读取单字节

主器件	启动	从器件地址 (W)		地址		启动	从器件地址 (R)				NACK	停止
从器件			ACK		ACK			ACK	数据			

表 3-2. 读取多字节

主器件	启动	从器件地址 (W)		地址		启动	从器件地址 (R)			ACK		ACK		NACK	停止
从器件			ACK		ACK			ACK	数据		数据				

在图 3-3 中，Saleae LogicPro16 分析仪的上图显示的是多字节读取，其中只读出一个写寄存器地址命令和 4 个字节（湿度和温度值），总时长为 0.1738ms。下图为要读出的每个字节值使用单独的写寄存器地址命令，总共持续 0.409ms。

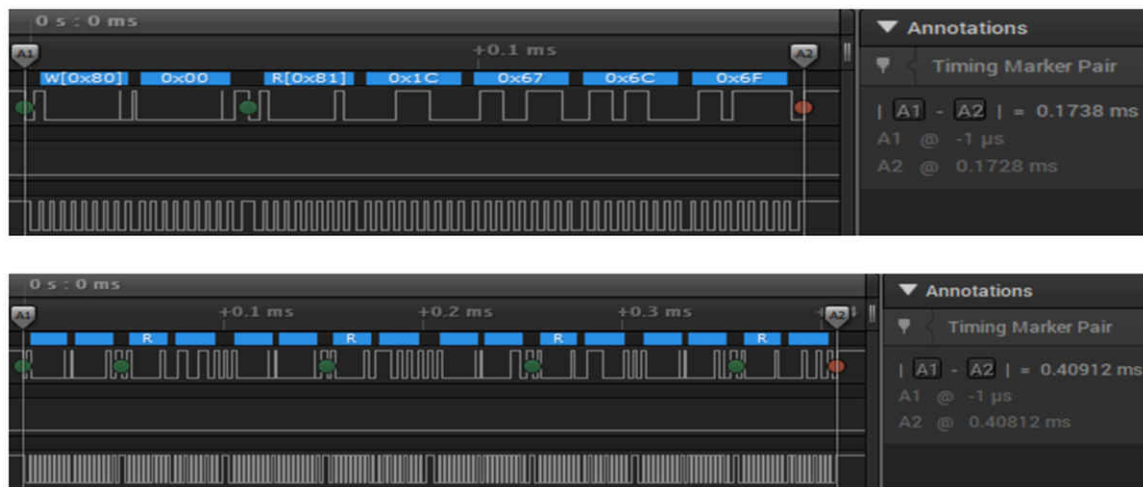


图 3-3. 四个字节读取模式的比较（多字节读取位于顶部）

采集到的传感器数据写入用户可配置的数据缓冲区，2 个字节的温度值和 2 个字节的湿度值一个接一个地被存储。当所需的缓冲区长度被数据填满时，例如，20 分钟后 80 字节，每分钟读取 1 次，通知 Arm Cortex-M4F MCU 获取数据缓冲区并将其传输。这显著减少了协议开销，前导码和 SYNC 只有相同的 8 个字节，但现在一次有 80 个字节的有效载荷。

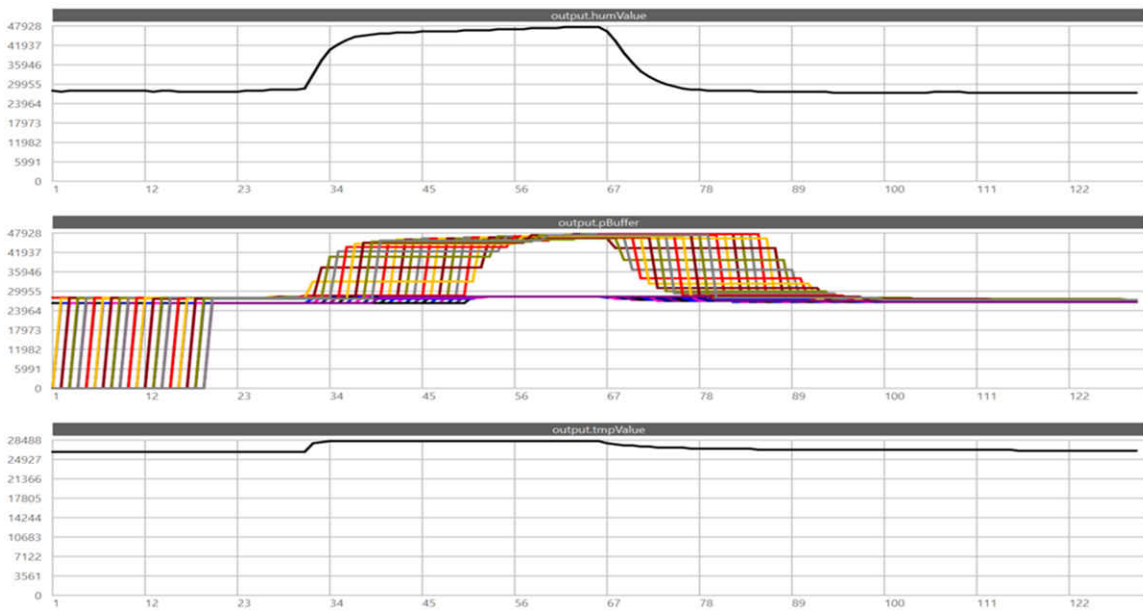


图 3-4. Sensor Controller Studio 中的温度和湿度数据缓冲器

80 字节的数据缓冲区大小是任意选择的，只是为了进行测试而选择了一个合理的大小。系统设计人员可能会增加缓冲区大小以利用 SCE 内的最大可用 ULL SRAM，其数量接近 3K 字节，请参阅所提供的 CC1352P + HDC2010 传感器控制器引擎项目的资源分配情况。

如果每分钟读取一次 HDC2010 器件，则缓冲区需要存储 4 个字节 × 60 分钟 = 1 小时内 240 个数据字节。通过添加 4 个字节的前导码、4 个字节的 SYNC 字和 2 个字节的连续数据包编号，数据包总长度之和为 250 个字节。包括 wM-Bus 和低功耗蓝牙 (BLE) 在内的多种无线协议使用的数据包长度达 256 字节。

图 3-5 所示为 SCE 通过 I2C 总线与 HDC2010 传感器进行交互的情况，其中显示了通道 0 (或 3.3V) 上到 I2C 上拉的正脉冲和通道 2 上 HDC2010 器件的通电脉冲 (低电平有效)。从断电开始，配置 HDC2010 寄存器，读出传感器数据的总时间仅为 3.36ms。

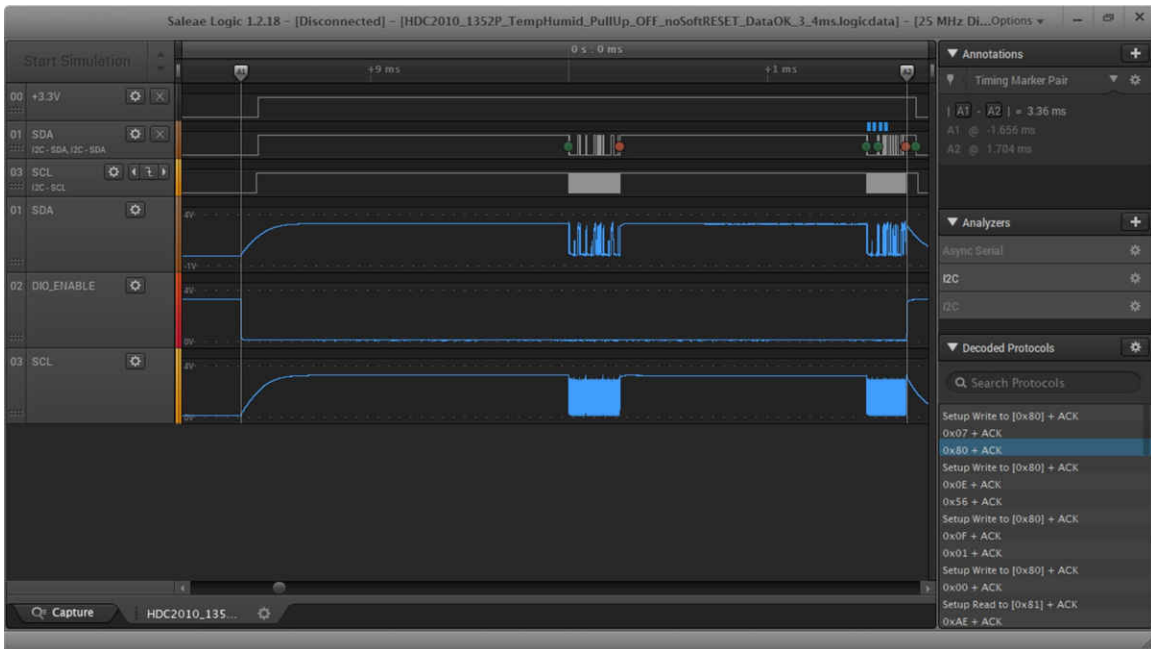


图 3-5. 完成对 HDC2010 的初始 SCE 访问 (I2C 上拉、配置、数据读出和断电)

请注意，如果 HDC2010 传感器不是每次都断电而是在两次测量之间保持待机状态，则所有后续数据读数都会更短。这两种解决方案均经过测试，最终固件在两次测量之间使 HDC2010 传感器保持待机模式。由于采用多字节读取模式，这会导致传感器数据读取速度非常快，时长仅为 198 μ s，如图 3-6 所示。

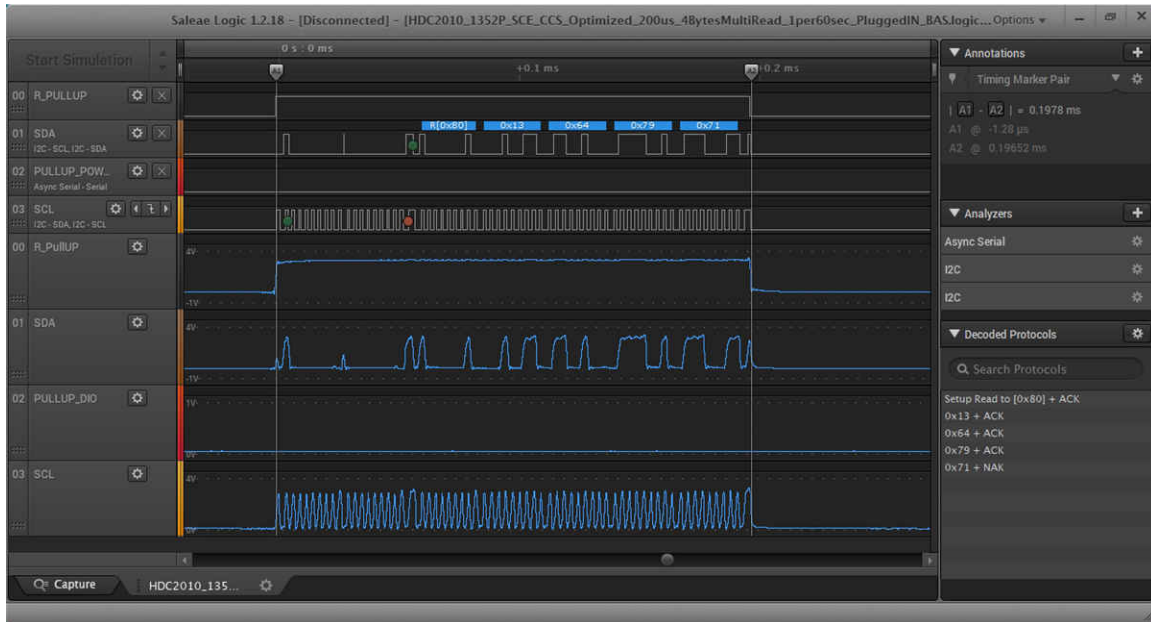


图 3-6. 使用 SCE 定期读取传感器，时长为 198 μ s

4 测试和验证

为了在待机和传感器控制器系统解决方案之间进行准确比较，具有周期性射频传输的 HDC2010 采样代码也在主 Arm Cortex-M4F MCU 上采用，而未使用 SCE。使用相同的 60 秒占空比及相同的射频参数和数据包长度，还添加了与 TIDA-00484 参考设计中相同的占空比方法。

4.1 EnergyTrace™ 结果

在此测试中，传感器读数（自动模式下的 HDC 读数）和 RF 数据包传输周期均设置为每分钟一次。RF 参数是 SmartRF Studio 中 50kbps、2-GFSK 在 +1dBm 下的默认设置。所有这些参数都等于 TIDA-00484 参考设计的设置，以支持对不同系统解决方案进行比较。图 4-1 显示的电流曲线，由 CCS 中的 EnergyTrace 函数捕获，总传输时间接近 6.2ms。此脉冲的最后 2.5ms 具有大约 9.4mA 的稳定电流，显示了 CC1352P 器件的无线数据传输（请注意，EnergyTrace 报告的是相对测量值，CC1352P 数据表中的值为 9mA）。

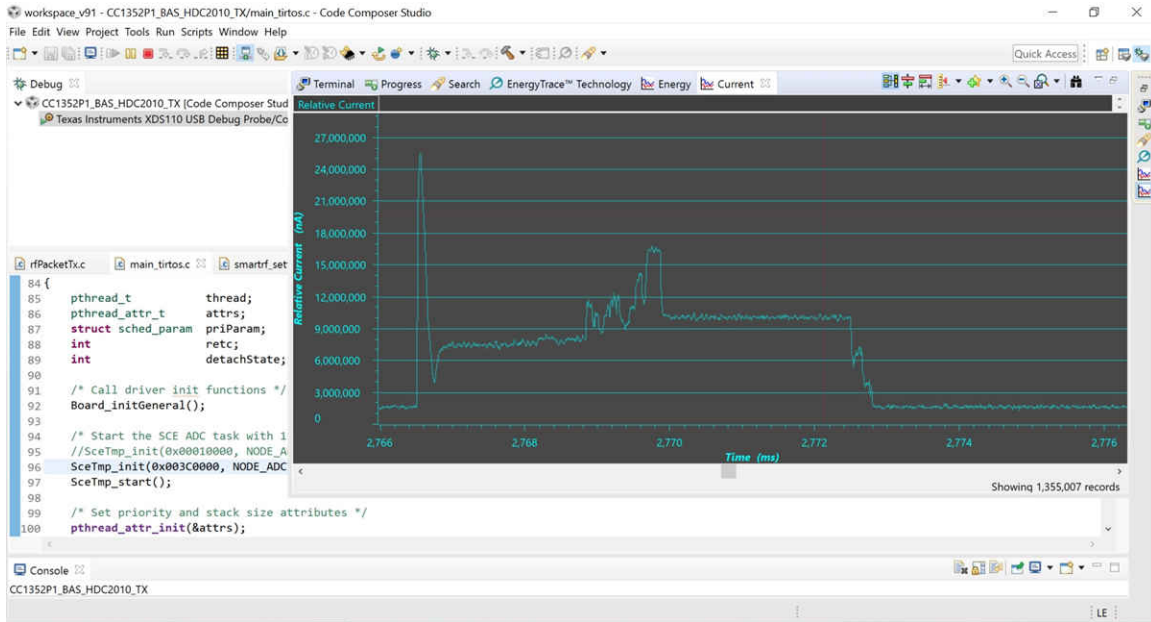


图 4-1. +1dBm 输出的电流曲线，包含 SCE 和 Arm® Cortex®-M4F 活动

4.2 平均电流消耗

使用 TI 专有工具进行了进一步测试，该工具在更高级的硬件和软件上使用 EnergyTrace™ 方法，结果与 Agilent Technologies N6705B 功率分析仪的偏差不到 3%。

该 TI 工具捕获了 100 秒的电流曲线，并计算了用户定义时间段（包括一次射频传输的 60 秒，或传输之间的任何时段）内的平均值。使用 3.0V 和 2.0V 电源进行了测试，以模拟 CR2032 纽扣电池或任何其他主 LiMnO2 电池的初始和使用寿命结束电压。开发并测试了一种特殊的固件，每 10 秒读取一次 HDC 传感器，而 RF 传输每 60 秒发生一次。若要在传输之前评估传感器控制器的效果以及传感器数据缓冲情况，需执行上述操作。占空比解决方案由 TIDA-00484 设计表述，其结果记录在 [可实现 10 年以上纽扣电池寿命且适用于低于 1GHz 的星形网络的湿度和温度传感器节点](#) 设计指南中。

表 4-1 的第 1 和第 2 列中显示的结果是由 N6705B 电源分析仪使用 Keysight 14585A 软件得出的。

表 4-1. 三种超低功耗架构的对比

	待机 (主 MCU 读取 HDC2010), I_{average} [60s]	传感器控制器 (SCE 读取 HDC2010), I_{average} [60s]	占空比 (在 TIDA-00484 中测量)
HDC2010 每 10 秒, RF 每 60 秒 [3.0V]/[2.0]	2181nA / 3040nA	1666nA / 2565nA	不适用
HDC2010 每 60 秒, RF 每 60 秒 [3.0 V]/[2.0]	1686nA / 2542nA	1693nA / 2576nA	1957nA

NOTE

结果显示，与待机解决方案相比，每 60 秒使用一次传感器控制器无法节省任何功耗（参见表 4-1）。比较 60 秒传感器读数和无线传输周期的 1686 和 1693nA 时，这两种解决方案的功耗值相同。表 4-1 中的值是电源电压、传感器采样率和待机或传感器控制器的每种组合的三次测试运行的平均结果。

如果每 10 秒读取一次传感器，则使用 SCE 外设的优势就变得显而易见了。更频繁地读取传感器会进一步放大利用 SCE 外设所带来的优势。占空比方法在 60 秒传感器采样周期时的能效较低，因为不活动时间太短，并且重新启动和重新初始化整个系统需要的能量多于在两者之间节省的能量。

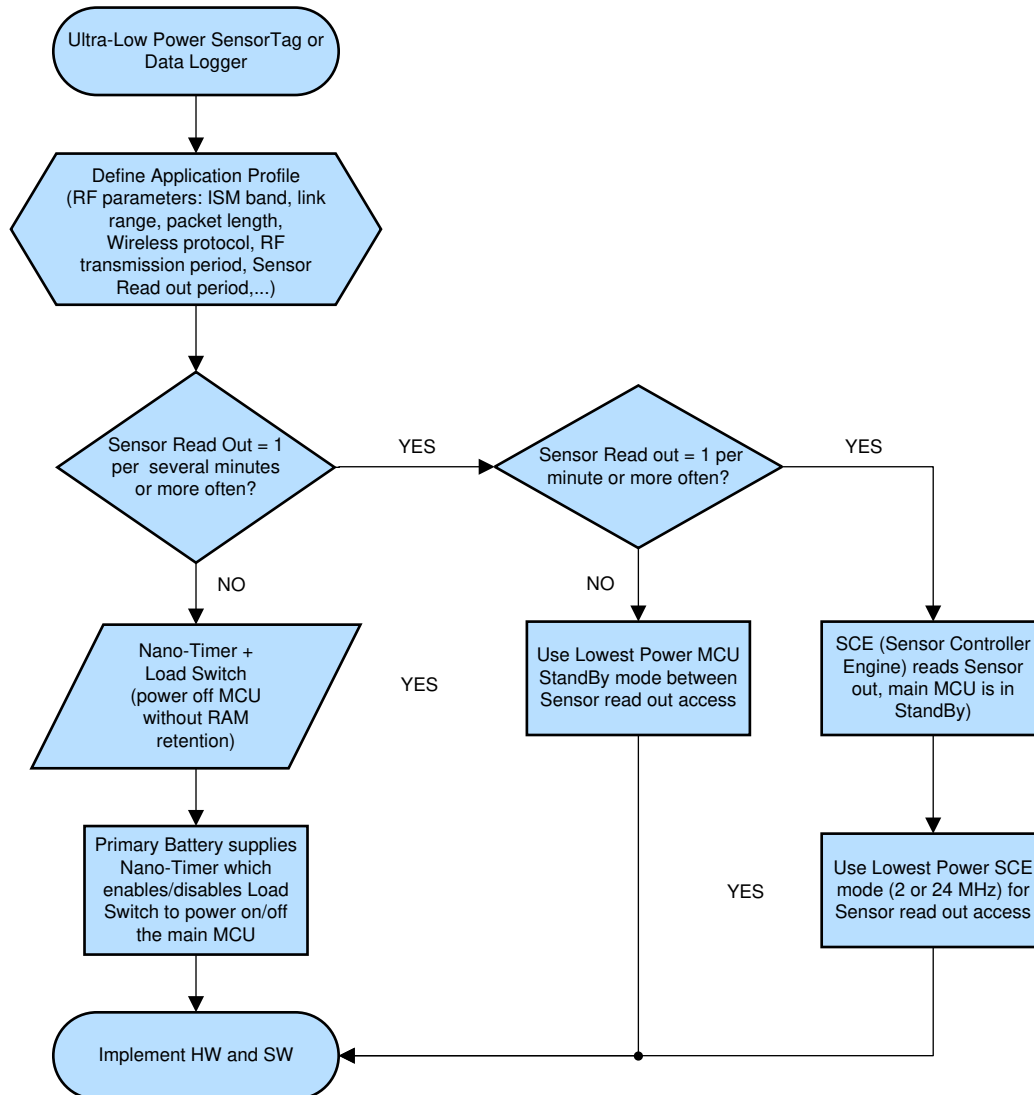


图 4-2. 待机、传感器控制器和占空比方法的超低功耗决策树

图 4-2 中的流程图是基于测试结果的决策树，用于在超低功耗是关键应用要求时轻松估计应考虑哪种系统方法。占空比方法（使用纳米计时器和负载开关）在不活动时间至少为 1-2 分钟或更长的低占空比情况下效率最高。

待机方法最容易实现（无需开发 SCE 代码），并且在 1 分钟传感器采样时间内的性能优于占空比方法。

当每分钟读取一次传感器时，传感器控制器方法的性能等同于待机方法（通过表 4-1 中的 N6705B 测量数据得到确认），前者可节省功耗并且传感器读取频率更高（例如 10 秒一次）。

图 4-3 中显示了三种解决方案的图形表示，以及不活动期间（传感器和无线 MCU 处于待机或完全断电状态）、传感器读取期间和随后 RF 数据包传输期间的平均电流。此处的电流值是使用 TI 的能源工具测得的。

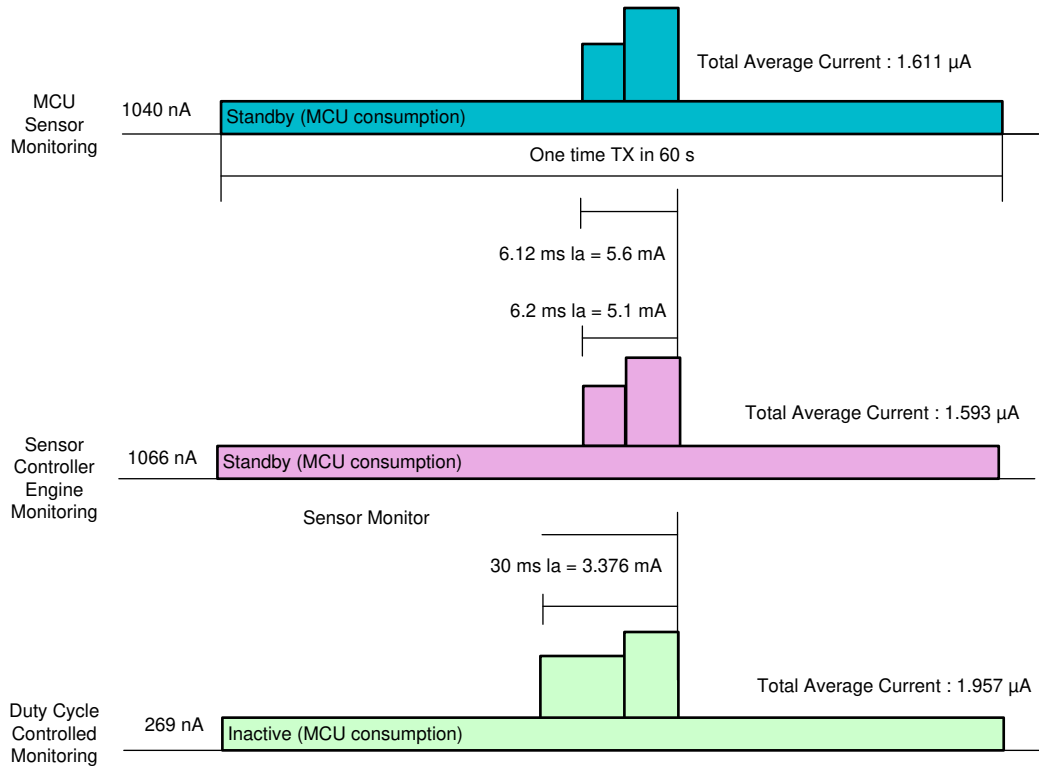


图 4-3. 经过简化的电流曲线 (比较)

如 TIDA-00484 设计所示，电压为 3V 时，占空比受控监控的平均电流为 1.957µA。新推出的传感器控制器引擎方法在电压为 3V 时消耗 1.593µA 电流，即在此应用中减少近 20% 的功耗。

以下 Excel 表格用于对三种解决方案进行比较，使用的是根据 TIDA-00484 设计中所述的相同 240mAh CR2032 纽扣电池容量和电池寿命公式。

Application Profile			
	Standby	Sensor Controller (SCE)	Duty-Cycle
*System Solution	240 mAh	240 mAh	240 mAh
*Battery Capacity	240 mAh	240 mAh	240 mAh
*Pulse Average Current	5.6 mA	5.1 mA	3.376 mA
*Pulse time	6.12 ms	6.2 ms	30 ms
*Inactive Mode Current	1040 nA	1066 nA	269 nA
*Sensor read out every 1 minute	1 per 60 s	1 per 60 s	1 per 60 s
*RF packet length (over-the-air)	112 bits	112 bits	104 bits
*Average Current	1.61109392 uA	1.59289 uA	1.956866 uA
*Power Consumption	0.09666564 mAs	0.095573 mAs	0.117412 mAs
*Battery Life time (85% derating factor)	14.4545709 Years	14.61976 Years	11.9005 Years

图 4-4. 系统解决方案比较 (每分钟读取一次传感器和传输一次数据)

4.3 RAM 中数据缓冲的节能效应

了解在超低泄漏 RAM 中读取和缓冲传感器数据的效果非常重要，这有助于以更少的协议开销传输一个更长的数据包。这可通过待机和传感器控制器解决方案来实现，因为有两个专用 RAM 区域可用：一个用于 Arm Cortex-M4F，另一个较小的用于 SCE 外设。

占空比方法无法缓冲数据，因为它会造成断电并且所有 RAM 内容都将丢失。因此，必须在断电之前无线发送传感器数据，从而进行额外的无线开销传输，这是位于每个无线发送的数据包的开始位置。在本报告中，低于 1GHz 协议开销包括 4 字节前导码 (0101.. 序列)，后跟 4 字节同步模式 (也称为 SYNC 字) 和 2 字节数据包序列号 (或仅用一个字节来编码数据包长度)。

假设总数据包长度为 250 字节 (包括 10 字节开销) , 请参见节 3.2 , 当每分钟读取一次传感器时, 待机和传感器控制器解决方案可发送在 1 小时内收集到的 60 个湿度和温度值数据。相比之下, 占空比方法将冗余传输 59 次相同的无线协议开销 (9 或 10 个字节) , 功耗几乎最高, 因为这些字节均是以无线方式发送的。

*System Solution	Standby	Sensor Controller (SCE)	Duty-Cycle
*Battery Capacity	240 mAh	240 mAh	240 mAh
*Average Current (including RF)	1611 nA	1593 nA	1957 nA
*Sensor read out every 1 minute	60 per hour	60 per hour	60 per hour
*RF packet length (over-the-air)	6720 bits	6720 bits	6240 bits
*Energy for 240 Bytes data (excluding RF protocol overhead)	4950 uAs	4885.2 uAs	7045.2 uAs
*Energy saving vs Duty-cycle solution	29.739 %	30.659172 %	0 %

图 4-5. 由于数据缓冲和避免无谓的无线开销而实现的功效

显然, 更长数据包的数据缓冲实现了进一步的节省, 达到了 30%, 请参见图 4-5。这是一个重要因素, 使用决策树时必须予以考虑, 请参见图 4-2。根据给定应用中的额外无线开销数量, 占空比解决方案甚至可能需要比最初估计的更长的非活动期才能提供超低的功耗。

使用 CC1352P MCU (以及 CC13xx 和 CC26xx 系列器件) 中的传感器控制器引擎进行传感器读取, 可实现与使用主 Arm Cortex-M4F MCU 相同或更低的功耗。如果应用在很多分钟或更长时段内不活动, 那么通过添加纳米计时器和负载开关器件, 整个系统的占空比可提供更低的功耗。

5 总结

本应用手册展示了一种超低功耗优化设计, 它利用 CC1352P MCU 中的传感器控制器引擎 (SCE) 来读取 HDC2010 湿度和温度传感器。测试了多个选项, 包括每 10 秒和每 60 秒的 HDC2010 读数, 并将后者与 TIDA-00484 设计中描述的基于纳米计时器的占空比解决方案进行比较。另外, 开发并测试了使用 Arm Cortex-M4F MCU 而非 SCE 读取传感器的待机解决方案。通过比较三种系统解决方案的优缺点, 推导出用于选择最高效超低功耗架构的决策树。

在了解了不活动时段的时长 (或传感器读取频率) 以及射频参数和射频协议开销后, 应用设计人员便知晓应该如何选择更低功耗系统架构。

事实证明, 新推出的传感器控制器引擎解决方案的性能等同于或优于标准待机方法 (在待机方法中, 使用 MCU 读取传感器) 。

如果应用不活动时段很长 (通常为数分钟甚至数小时) , 强烈建议使用占空比方法, 因为纳米计时器器件和负载开关可提供更高功率效率。

设计人员必须根据他们的应用规范和无线协议参数调整本报告中的注意事项。

6 参考文献

- 德州仪器 (TI) , [HDC2010 低功耗湿度和温度数字传感器数据表](#)
- 德州仪器 (TI) , [具有集成式功率放大器的 CC1352P SimpleLink™ 高性能双频带无线 MCU 数据表](#)
- 德州仪器 (TI) , [可实现 10 年以上纽扣电池寿命且适用于低于 1GHz 的星形网络的湿度和温度传感器节点设计指南](#)
- 德州仪器 (TI) , [SCE 和待机 HDC2010 传感器读数的代码示例](#)

7 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (January 2020) to Revision A (June 2021)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式。.....	2

重要声明和免责声明

TI 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安全、安保或其他要求。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 产品的应用。严禁对这些资源进行其他复制或展示。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 (<https://www.ti.com/legal/termsofsale.html>) 或 [ti.com](https://www.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。

邮寄地址: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2021, 德州仪器 (TI) 公司

重要声明和免责声明

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 产品的应用。严禁对这些资源进行其他复制或展示。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

Copyright © 2022，德州仪器 (TI) 公司