

Application Note

使用 MSPM0C1104 进行宽范围和高分辨率 PWM 信号捕获



Joe Ji, Janz Bai

摘要

PWM 信号在系统设计中用作速度控制或反馈，因此务必在宽泛的周期范围内准确测量 PWM 信号，以满足系统控制精度要求。一般而言，需要高 MCU 时钟频率来实现高分辨率 PWM 信号捕获。本应用手册使用在 24MHz MCU 时钟频率上工作的 MSPM0C1104，在 100Hz 至 10kHz 频率范围内实现了输入 PWM 占空比和周期不到 1% 的捕获误差，让低功耗系统设计受益匪浅。

内容

1 简介.....2

1.1 PWM 信号捕获简介.....2

1.2 MSPM0C110x 简介.....2

2 PWM 信号捕获.....3

2.1 PWM 信号捕获方法.....3

2.2 使用 TIMx CC 块进行的 PWM 信号捕获.....3

2.3 使用 GPIO 中断进行的 PWM 信号捕获.....4

2.4 不同 PWM 信号捕获设计的比较.....5

3 软件实现.....6

3.1 识别上升沿和下降沿.....6

3.2 时序分类.....6

3.3 信号滤波器和结果计算.....8

4 系统测试.....9

4.1 测试设置.....9

4.2 变量监测.....9

4.3 PWM 信号捕获分辨率测试和比较.....9

5 总结.....11

6 参考资料.....11

商标

Arm® and Cortex® are registered trademarks of Arm Limited.
所有商标均为其各自所有者的财产。

1 简介

1.1 PWM 信号捕获简介

PWM 信号在系统设计中广泛用作速度控制或反馈。要实现高 PWM 捕获精度，需要相对较高的 MCU CLK 频率。MSPM0C110x 系列 MCU 凭借各种外设为系统设计带来福音，但 MCU CLK 仅为 24MHz，可能会导致 PWM 捕获误差。本应用报告针对使用 MSPM0C110x 系列 MCU (只有一个计时器和一个 GPIO) 时出现的 PWM 捕获误差提出了解决方法。该解决方法还可以识别 100% 和 0% PWM 占空比，且无需执行任何额外工作。

1.2 MSPM0C110x 简介

MSPM0C110x 微控制器 (MCU) 属于 MSP 高度集成的超低功耗 32 位 MCU 系列，该 MCU 系列基于增强型 Arm® Cortex®-M0+ 内核平台，工作频率最高可达 24MHz。这些 MCU 可提供高性能模拟外设集成。

MSPM0C110x 器件提供高达 16KB 的嵌入式闪存程序存储器和 1KB 的 SRAM。该器件集成了各种高性能模拟外设，例如一个以 VDD 作为电压基准的 12 位 1.5Msps ADC 和片上温度传感器。该些器件还提供智能数字外设，例如一个 16 位高级计时器、两个 16 位通用计时器、一个窗口化看门狗计时器和各种通信外设 (包括一个 UART、一个 SPI 和一个 I2C)。这些通信外设为 LIN、IrDA、DALI、Manchester、Smart Card、SMBus 和 PMBus 提供协议支持

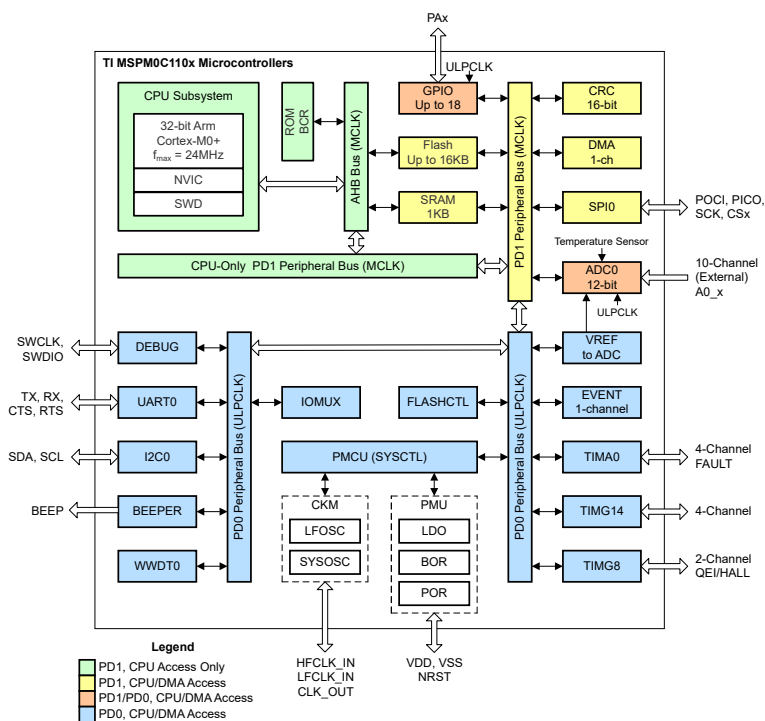


图 1-1. MSPM0C110x 功能方框图

2 PWM 信号捕获

2.1 PWM 信号捕获方法

本节讨论了两种捕获 PWM 输入信号的方法。第一种方法是使用 TIMx CC 块捕获 PWM 信号。第二种方法是使用 GPIO 中断捕获 PWM 信号。

2.2 使用 TIMx CC 块进行的 PWM 信号捕获

捕获和比较 (CC) 块用于捕获事件或比较事件。使用 MSPM0C 系列 MCU 时, TIMG 具有多达 2 个相同的捕获和比较块, TIMA 具有多达 4 个相同的捕获和比较块, 用于支持外部或内部信号。计时捕获模式用于生成捕获事件并记录时间间隔, 这对于速度计算或时间测量很有用。请参阅 *MSPM0 C 系列 24MHz 微控制器技术参考手册*, 了解有关 CC 锁定设置和关键寄存器的更多信息, 以便配置捕获模式。

使用两个捕获寄存器可以同时捕获单个输入波形的脉宽和周期。输入信号可以从外部连接到 CCP 的通道 0, IFCTL_01[1] 寄存器可以配置为在内部将输入端连接到 CCP 的通道 1, 从而使捕获寄存器 0 (TIMx.CC0) 来捕获脉宽, 捕获寄存器 1 (TIMx.CC1) 来捕获周期。组合脉宽和周期捕获的预期内部时序如图 2-1 所示。

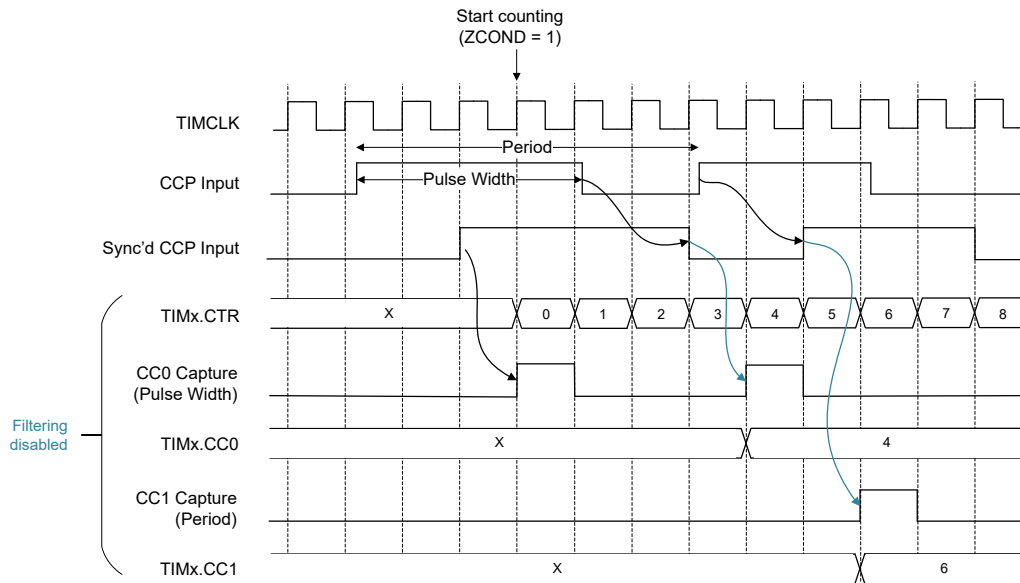


图 2-1. 组合的脉宽和周期捕获

下面是占空比和周期计算公式：

$$Period = T_{Load} - T_{Cc1} \quad (1)$$

$$Duty = \frac{(T_{load} - T_{CC0})}{Period} \quad (2)$$

- T_{load} 是在 TIMx.CTR 中加载的最大值
- T_{cc0} 是 TIMx.CC0 中捕获的值
- T_{cc1} 是 TIMx.CC1 中捕获的值

TIMx.CTR 设置为向下计数模式时, 公式成立。

图 2-2 展示了这种方法的捕获时序, 因为在 T1 捕获第一个输入边沿信号后, 必须手动重新加载 TIMx.CTR。T1 到 T2 之间的延时时间导致该误差。

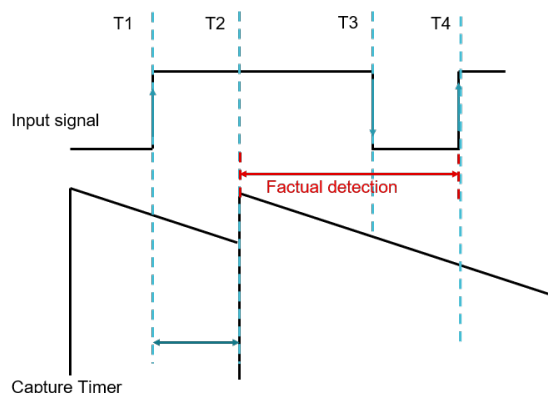


图 2-2. 使用 Timx CC 块捕获 PWM 占空比和周期的计时器顺序

- T1：触发中断以将 PRD 值加载到 CTR 中
- T2：将计时器计数值复位，以将值加载到中断处理程序中
- T2 - T1：进入中断处理程序以加载值的时间间隔
- T3：加载到 CC0 中的计数值
- T4：加载到 CC1 中的计数值并触发中断

2.3 使用 GPIO 中断进行的 PWM 信号捕获

PWM 占空比和周期捕获也可以通过软件中的 GPIO 中断来实现。可以使用 GPIO 第一个事件发布者在 GPIO 下降沿和上升沿生成 CPU 中断。用户可以控制 GPIO 中断中的 CPU 计时器操作，以捕获 PWM 占空比和周期。以下各部分介绍了捕获时序。以下各章介绍了软件详细信息。为了提高捕获精度，一旦 GPIO 中断进入，计时器必须尽快启动或停止，以验证 T1、T2 和 T3 之间且短暂且相等的 SW 延迟。这种 PWM 占空比和周期捕获方法的时序如图 2-3 所示。

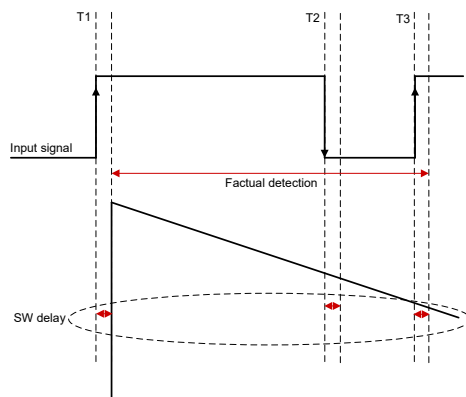


图 2-3. 使用 GPIO 中断捕获 PWM 占空比和周期的计时器顺序

- T1：在上升沿触发 GPIO 中断并启动计时器。
- T2：在下降沿触发 GPIO 中断并记录 TIMx.CTR 值。
- T3：在上升沿触发 GPIO 中断并停止计时器。

2.4 不同 PWM 信号捕获设计的比较

表 2-1 列出了两种 PWM 捕获方法之间的差异，请参阅表格“使用 TIMx CC 块进行的 PWM 信号捕获 (用 MCU 硬件实现) ”。使用 GPIO 中断进行 PWM 信号捕获可获得更高的精度和可行性：

表 2-1. PWM 信号捕获方法比较

项目	使用 TIMx CC 块进行的 PWM 信号捕获	使用 GPIO 中断进行的 PWM 信号捕获
捕获延迟	由计时器捕获块决定	由中断和软件决定
捕获误差	由计时器捕获块决定	可由软件管理
绝对误差	因 PWM 频率和占空比而异	低于 1%，容易补偿误差

3 软件实现

可通过 E2E 论坛下载演示代码，该论坛的链接随附在参考资料部分。要验证使用 GPIO 中断功能进行 PWM 信号捕获，请在 CCS 项目预定义符号窗口中添加“Capture_IO”；要验证使用 TIMx CC 块进行 PWM 信号捕获，请在 CCS 项目预定义符号窗口中添加“Capture_CC”。

3.1 识别上升沿和下降沿

GPIO 中断可在 GPIO 上升沿和下降沿触发。根据表 3-1，读取 GPIO 状态，以确定是 GPIO 上升沿触发中断，还是 GPIO 下降沿触发中断。

表 3-1. GPIO 下降沿和上升沿中断

GPIO 状态	触发中断的 GPIO 边沿
GPIO 输入= 低电平	下降沿
GPIO 输入= 高电平	上升沿

3.2 时序分类

本演示将 GPIO 下降沿作为 PWM 脉冲的起始。正确的时序必须是：检测到第一个下降沿，检测到上升沿，然后检测到第二个下降沿。在某些极端条件下，PWM 占空比小于 0.2% 或 PWM 占空比大于 99.7%。由于存在噪声，时序可能不准确。在演示中设置了 gflag 和 gflag_2，用于确定正确的时序。表 3-2 列出了不同情况下的相应操作

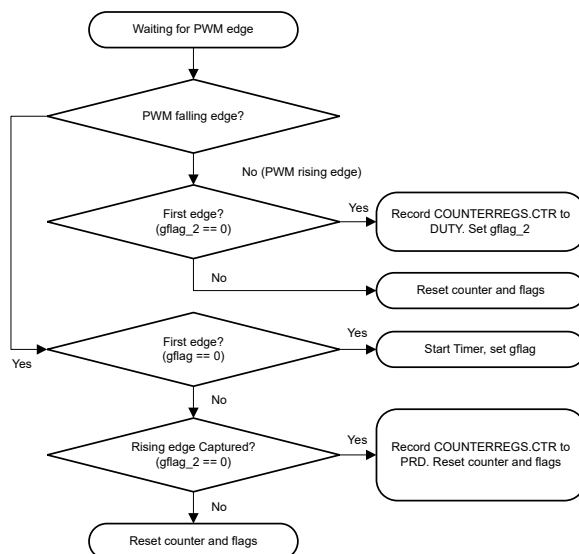


图 3-1. 使用 GPIO 中断进行 PWM 占空比和周期捕获的软件示意图

表 3-2. 在不同条件下 GPIO 中断处理程序中的操作

GPIO 边沿	gflag	gflag_2	操作
上升沿	0	0	不采取任何措施
	0	1	未发生
	1	0	将 COUNTERREGS.CTR 记录到 DUTY，设置 gflag_2
	1	1	复位计数器和标志
下降沿	0	0	启动计时器，设置 gflag
	0	1	未发生
	1	0	复位计数器和标志
	1	1	将 COUNTERREGS.CTR 记录到 PRD，复位标示

3.3 信号滤波器和结果计算

为了避免输入抖动和噪声的影响，演示代码为捕获的占空比和周期设置了一个滤波器窗口，该窗口可以通过宏定义 *FilterWindow* 进行更改。信号滤波器和结果计算流程在图 3-2 进行介绍。

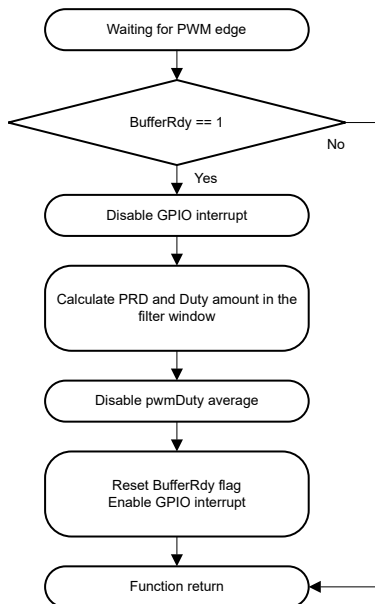


图 3-2. 信号滤波器和结果计算软件示意图

4 系统测试

本部分讨论 PWM 捕获方法的测试设置和测试结果。

4.1 测试设置

需要使用波形发生器来获取测试中使用的基准 PWM。波形发生器也可以替换成为可以生成精确 PWM 的另一个 MCU 或 DSP。将 PWM 信号连接到 LP-MSPM0C1104 的 PA23 和 GND。使用 Micro-USB 电缆将 LP-MSPM0C1104 连接到 PC，如图 4-1 所示。

可通过 E2E 论坛下载演示代码，该论坛的链接随附在参考资料部分。添加预定义符号 *Capture_CC*，然后点击“Build (构建)”以使用 TIMx CC 块调试 PWM 信号捕获，或者添加预定义符号 *Capture_IO*，然后点击“Build (构建)”以使用 GPIO 中断调试 PWM 信号捕获。在项目中，计时器时钟预分频器设置为 5，所需的计时器周期设置为 12ms，以捕获 100Hz 频率 PWM。

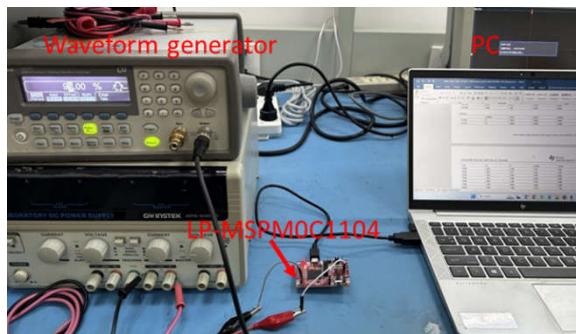


图 4-1. 测试的硬件设置

4.2 变量监测

两个项目中的监测变量都是 *pwmDuty*。将该值与波形发生器中设置的实际 PWM 占空比进行比较，从而获得捕获误差。

4.3 PWM 信号捕获分辨率测试和比较

为了比较这两种方法在全范围和极端条件下的表现，本测试使用了 100Hz/ 2KHz/ 5kHz/ 10kHz PWM 频率，PWM 占空比范围为 0.2% 至 99.8%。图 4-2 展示了使用 GPIO 中断 (蓝线) 和 TIMx CC 块 (橙线) 时的捕获误差比较，x 轴表示 PWM 占空比，为了正确显示极端条件，数字不成比例，y 轴表示捕获误差 (以 % 为单位)。表 4-1 和表 4-2 展示了使用 GPIO 中断 (蓝线) 和 TIMx CC 块 (橙线) 获得的捕获结果。

在所有 PWM 占空比范围内，GPIO 中断方法将捕获误差保持在 1% 以下，并且在 PWM 占空比 <1% 和 >99% 的极端条件下还具有稳健的性能，这可以避免异常数据。当 PWM 占空比较大 (尤其是 >99%) 时，TIMx CC 块方法显示出高分辨率，但当 PWM 占空比 <1% 时，必须手动校正捕获值，以避免 >100% 的错误结果。

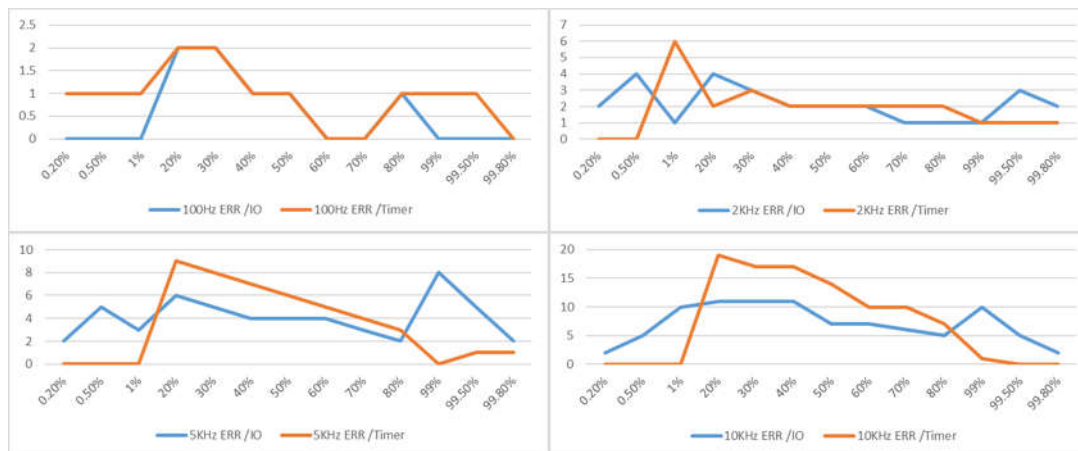


图 4-2. PWM 占空比捕获误差比较

表 4-1. 使用 GPIO 中断获得的 PWM 占空比捕获误差

输入信号	100Hz		2KHz		10KHz			
	计算结果	误差	计算结果	误差	计算结果	误差	计算结果	误差
0.2%	0.2%	0	0.0%	0.2%	0.0%	0.2%	0.0%	0.2%
0.5%	0.5%	0	0.9%	0.4%	0.0%	0.5%	0.0%	0.5%
1%	1.0%	0	1.1%	0.1%	2.3%	0.3%	0.0%	1.0%
20%	20.2%	0.2%	20.4%	0.4%	20.6%	0.6%	21.0%	1.0%
30%	30.2%	0.2%	30.3%	0.3%	30.5%	0.5%	31.0%	1.0%
40%	40.1%	0.1%	40.2%	0.2%	40.4%	0.4%	41.0%	1.0%
50%	50.1%	0.1%	50.2%	0.2%	50.4%	0.4%	50.7%	0.7%
60%	60.0%	0	60.2%	0.2%	60.4%	0.4%	60.7%	0.7%
70%	70.0%	0	70.1%	0.1%	70.3%	0.3%	70.6%	0.6%
80%	79.9%	0.1%	80.1%	0.1%	80.2%	0.2%	80.5%	0.5%
99%	99.0%	0	99.1%	0.1%	98.2%	0.8%	100.0%	1.0%
99.5%	99.5%	0	99.2%	0.3%	100.0%	0.5%	100.0%	0.5%
99.8%	99.8%	0	100.0%	0.2%	100.0%	0.2%	100.0%	0.2%

表 4-2. 使用 TIMx CC 块获得的 PWM 占空比捕获误差

输入信号	100Hz		2KHz		5KHz		10KHz	
	计算结果	误差	计算结果	误差	计算结果	误差	计算结果	误差
0.2%	0.1%	0.1%	0%	0.2%	0%	0.2%	0%	0.2%
0.5%	0.4%	0.1%	0%	0.5%	0%	0.5%	0%	0.5%
1%	0.9%	0.1%	0.4%	0.6%	0%	1.0%	0%	1.0%
20%	20.2%	0.2%	19.8%	0.2%	19.1%	0.9%	18.1%	1.9%
30%	30.2%	0.2%	29.7%	0.3%	29.2%	0.8%	28.3%	1.7%
40%	40.1%	0.1%	39.8%	0.2%	39.3%	0.7%	38.3%	1.7%
50%	50.1%	0.1%	49.8%	0.2%	49.4%	0.6%	48.6%	1.4%
60%	60.0%	0	59.8%	0.2%	59.5%	0.5%	59.0%	1.0%
70%	70.0%	0	69.8%	0.2%	69.6%	0.4%	69.0%	1.0%
80%	79.9%	0.1%	79.8%	0.2%	79.7%	0.3%	79.3%	0.7%
99%	98.9%	0.1%	98.9%	0.1%	99.0%	0	98.9%	1.0%
99.5%	99.4%	0.1%	99.4%	0.1%	99.4%	0.1%	99.5%	0
99.8%	99.8%	0	99.7%	0.1%	99.7%	0.1%	99.8%	0

5 总结

本应用手册针对使用 MSPM0C110x 系列 MCU (只有一个计时器和一个 GPIO) 时出现的 PWM 捕获误差提出了解决方法。该解决方法还可以识别 100% 和 0% PWM 占空比, 且无需执行任何额外工作。测试结果表明, 该方法可以在 100Hz 至 10kHz PWM 频率范围内将捕获误差保持在 1% 以下。测试结果还表明, 该方法在 PWM 占空比 <1% 和 >99% 的极端条件具有稳健性。

6 参考资料

- 德州仪器 (TI), [MSPM0 C 系列 24MHz 微控制器技术参考手册](#), 技术参考手册。
- 德州仪器 (TI), [MSPM0C110x、MSPS003 混合信号微控制器](#) 数据表。
- 德州仪器 (TI), [E2E 论坛](#), E2E 论坛。

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