

Errata

MSPM0L111x 微控制器



摘要

本文档介绍了功能规格的已知例外情况（公告）。

内容

1 功能公告.....	1
2 预编程软件公告.....	3
3 仅调试公告.....	3
4 编译器修复公告.....	3
5 器件命名规则.....	3
6 公告说明.....	5
7 修订历史记录.....	25

商标

所有商标均为其各自所有者的财产。

1 功能公告

影响器件运行、功能或参数的公告。

✓ 复选标记表示指定版本中存在该问题。

勘误编号	修订版 A、B
ADC_ERR_05	✓
AES_ERR_01	✓
CLK_ERR_02	✓
CPU_ERR_02	✓
CPU_ERR_03	✓
CPU_ERR_04	✓
DMA_ERR_02	✓
FCC_ERR_01	✓
FLASH_ERR_03	✓
FLASH_ERR_04	✓
FLASH_ERR_05	✓
FLASH_ERR_08	✓
FLASH_ERR_09	✓
GPIO_ERR_05	✓
GPIO_ERR_06	✓
I2C_ERR_04	✓
I2C_ERR_05	✓
I2C_ERR_06	✓
I2C_ERR_07	✓
I2C_ERR_08	✓
I2C_ERR_09	✓

勘误编号	修订版 A、B
I2C_ERR_10	✓
I2C_ERR_13	✓
I2C_ERR_15	✓
KEYSTORE_ERR_01	✓
PMCU_ERR_10	✓
PMCU_ERR_12	✓
PMCU_ERR_14	✓
RST_ERR_01	✓
RST_ERR_02	✓
SPI_ERR_02	✓
SPI_ERR_04	✓
SPI_ERR_05	✓
SPI_ERR_06	✓
SPI_ERR_07	✓
SPI_ERR_10	✓
SYSCTL_ERR_01	✓
SYSCTL_ERR_02	✓
SYSCTL_ERR_03	✓
SYSCTL_ERR_04	✓
SYSCTL_ERR_05	✓
SYSCTL_ERR_06	✓
SYSCTL_ERR_10	✓
SYSOSC_ERR_01	✓
SYSOSC_ERR_02	✓
SYSOSC_ERR_04	✓
SYSOSC_ERR_05	✓
TIMER_ERR_04	✓
TIMER_ERR_06	✓
TIMER_ERR_07	✓
UART_ERR_01	✓
UART_ERR_02	✓
UART_ERR_04	✓
UART_ERR_05	✓
UART_ERR_06	✓
UART_ERR_07	✓
UART_ERR_08	✓
UART_ERR_10	✓
UART_ERR_11	✓

2 预编程软件公告

影响出厂编程软件的公告。

✓ 复选标记表示指定版本中存在该问题。

3 仅调试公告

仅影响调试操作的公告。

✓ 复选标记表示指定版本中存在该问题。

4 编译器修复公告

由编译器权变措施解决的公告。请参阅每个公告，以了解 IDE 和编译器版本及权变措施。

✓ 复选标记表示指定版本中存在该问题。

5 器件命名规则

为了标示产品开发周期所处的阶段，TI 为所有 MSP MCU 器件的器件型号分配了前缀。每个 MSP MCU 商用系列产品都具有以下两个前缀之一：MSP 或 XMS。这些前缀代表了产品的发展阶段，即从工程原型 (XMS) 直到完全合格的生产器件 (MSP)。

XMS - 实验器件，不一定代表最终器件的电气规格

MSP - 完全合格的生产器件

支持工具命名前缀：

X：还未经德州仪器 (TI) 完整内部质量测试的开发支持产品。请注意，带有 X 标记的开发样片应在 -40°C 至 85°C 温度范围内运行。

null：完全合格的开发支持产品。

XMS 器件和 X 开发支持工具在供货时附带如下免责条款：

“开发中的产品用于内部评估用途。”

MSP 器件的特性已经全部明确，并且器件的质量和可靠性已经完全论证。TI 的标准保修证书对该器件适用。

预测显示原型器件 (XMS) 的故障率大于标准生产器件。由于这些器件的预计最终使用故障率尚不确定，德州仪器 (TI) 建议不要将它们用于任何生产系统。请仅使用合格的生产器件。

TI 的器件命名规则还包含具有器件产品系列名称的后缀。此后缀表示温度范围、封装类型和配送形式。

下方的器件封装图展示了封装编号法方案。

下方的封装图展示了封装编号法方案，并且 x 定义了器件修订版与版本 ID 间的映射关系。

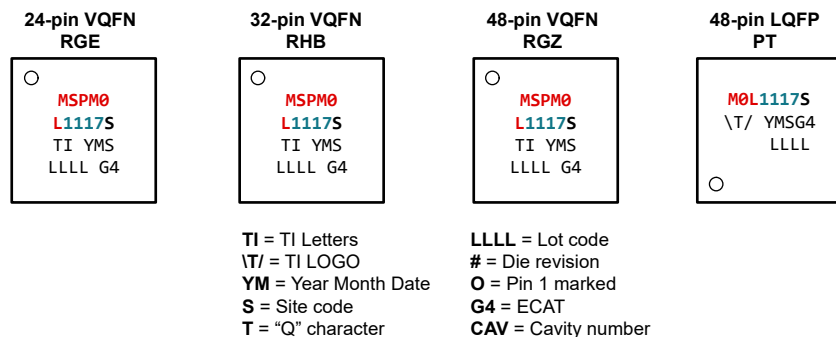


图 5-1. 封装符号

芯片修订版本

修订版字母 (封装标记)	版本 (在器件的常量存储器中)
A	0x0
B	0x0

修订版字母表示产品硬件修订版本。根据修订字母，本文档中的公告标记为“适用于”或“不适用于”给定器件。该字母映射到器件存储器中存储的整数，可用于使用应用软件或已连接的调试探针来查找修订版本。

6 公告说明

ADC_ERR_05 **ADC 模块**

类别

功能

功能

启用 IP 之前生成的 HW 事件，ADC 触发器将保持在队列中

说明

当 ADC 配置为 HW 事件触发模式并且在启用 ADC 之前生成触发信号时，ADC 触发信号将保持在队列中。启用 ADC 后，会触发采样和转换。

权变措施

将 ADC 配置为 HW 触发模式后，先启用 ADC，然后再进行外部触发。

AES_ERR_01 **AES 模块**

类别

功能

功能

AES 保存的上下文就绪中断未按预期生成

说明

未生成已保存的上下文就绪中断。如果对任何 AES 寄存器进行了访问（读取或写入），则会生成中断。

权变措施

使用基于轮询的机制，检查 CTRL 寄存器中已保存的上下文就绪的状态位，而不是中断。

CLK_ERR_02 **CLK 模块**

类别

功能

功能

当 HFCLK_IN 作为 HFCLK 和 HSCLK 的源时，MFCLK 会错误地以 BUSCLK 为源

说明

当 HSCLK 完全以 HFCLK 为源，并且 HFCLK_IN 被配置为 HFCLK 的源时，MFCLK 会被错误地设置为 BUSCLK，而非 4MHz。

权变措施

无权变措施

CPU_ERR_02 **CPU 模块**

类别

功能

功能

禁用 CPUSS 预取/缓存具有限制

说明

如果存在待处理的闪存访问，CPU 预取/缓存禁用将不会生效

CPU_ERR_02

(续)

CPU 模块**权变措施**

禁用高速缓存和预取器（顺序无强制要求），然后对 SYSTCL 中的关断存储器 (SHUTDNSTORE) 发出存储器访问（更多参考信息，请查看器件 TRM 中的 SHUTDNSTORE 寄存器）。

存储器访问完成后，预取器/高速缓存将被禁用。

示例：

```
CPUSS->CTL = ( CPUSS_CTL_PREFETCH_DISABLE |
CPUSS_CTL_ICACHE_DISABLE); //禁用指令缓存和预取
DL_SYSTCL_setShutdownStorageByte(DL_SYSTCL_SHUTDOWN_STORAGE_BYTE_X
, data) // 将一个字节保存到 SHUTDOWN 存储器
```

CPU_ERR_03**CPU 模块****类别**

功能

功能

在转换到低功耗模式时，预取器可能会获取错误的指令

说明

转换到低功耗模式且存在待处理的预取时，预取器可能会错误地获取不正确的数据（全 0）。当器件唤醒时，如果预取器和高速缓存未被 ISR 代码覆盖，则从闪存执行的主代码可能会损坏。例如，如果 ISR 位于 SRAM 中，则从闪存预取的不正确的数据不会被覆盖。当 ISR 返回损坏的数据时，CPU 可能会提取预取器中的数据，从而导致指令不正确。硬件事件唤醒是将唤醒器件但不刷新预取器的进程的另一个示例。

权变措施

进入低功耗模式之前禁用预取器。

示例：

```
CPUSS->CTL &= 0x6; // 禁用预取器，保持其他设置
SYSTCL.SOCLOCK.SHUTDNSTORE0 // 从 SHUTDOWN 存储器读取
__WFI(); // or __WFE(); 该函数调用转换到低功耗模式
CPUSS->CTL |= 0x1; // 启用预取器
```

CPU_ERR_04**CPU 模块****类别**

功能

功能

高速缓存无法从导致硬故障的 FLASH 地址返回正确的数据

说明

如果器件因某个 FLASH 地址进入硬故障状态，在器件从该硬故障恢复并尝试从其高速缓存（存储来自该 FLASH 地址的数据）中检索信息后，返回值变为零。此外，如果访问相同的 FLASH 地址，则不会触发新的硬故障。

CPU_ERR_04

(续)

CPU 模块

权变措施

通过更改 CPUSS.CTL.ICACHE 位禁用然后重新启用高速缓存。

示例：

CPUSS->CTL &= ~(CPUSS_CTL_ICACHE_ENABLE); ///禁用指令缓存

CPUSS->CTL |= CPUSS_CTL_ICACHE_ENABLE; ///启用指令缓存

DMA_ERR_02

DMA 模块

类别

功能

功能

64 位到 128 位以及 128 位到 64 位的 DMA 混合字节传输无法正常工作

说明

当执行源宽度为 128 位且目标宽度为 64 位，或者源宽度为 64 位且目标宽度为 128 位的 DMA 传输时，传输将无法正确完成。

权变措施

目前没有权变措施。仅使用宽度匹配的传输 (64 位到 64 位或 128 位到 128 位)。

FCC_ERR_01

FCC 模块

类别

功能

功能

当 BUSCLK 以 LFCLK 为源时，FCC 行为异常

说明

当 BUSCLK 以 LFCLK 为源时，FCC 无法按预期工作。表现为 FCC 完成信号未置为有效，或者报告的 FCC 值不正确。

权变措施

无权变措施

FLASH_ERR_03

FLASH 模块

类别

功能

功能

在 2 个等待状态下访问闪存，然后访问无效引导代码，将导致下一次闪存访问也出现违规

说明

在 2 个等待状态下访问闪存，然后访问 BOOTCODE，将导致下一次闪存访问也出现违规。

权变措施

请勿尝试在启动阶段后访问引导代码区域。否则，在引导代码违规和下一次正确闪存访问之间需要间隔 4 个时钟周期。

FLASH_ERR_04	FLASH 模块
类别	功能
功能	如果错误未发生在主闪存区域，则 SYSCTL_DEDERRADDR 中会报告错误地址。
说明	
权变措施	如果 SYSCTL_DEDERRADDR 的返回地址返回 0x00Cxxxxx，请使用 0x41000000 进行“OR”操作，以获取 NONMAIN 或 Factory 区域返回地址的正确地址。例如，如果 SYSCTL_DEDERRADDR = 0x00C4013C，则实际地址为 0x41C4013C。 如果 SYSCTL_DEDERRADDR 的返回地址返回 0x00Dxxxxx，请使用 0x41000000 进行“OR”操作，以获取数据银行返回地址的正确地址。例如，如果 SYSCTL_DEDERRADDR = 0x00D0012A，则实际地址为 0x00D0012A。
FLASH_ERR_05	FLASH 模块
类别	功能
功能	DEDERRADDR 可能具有不正确的复位值
说明	SYSCTL->DEDERRADDR 的复位值可能返回 0x00C4013C，而不是正确的 0x00000000。错误发生在出厂微调区域，并不表示出现故障，可适当忽略。在器件上对 NONMAIN 进行编程后，复位值通常会发生变化。
权变措施	接受 0x00C4013C 作为另一个复位值，因此引导后的默认值可能为 0x00000000 或 0x00C4013C。返回值超出了器件 MAIN 闪存的范围，因此该返回值不可能来自实际的 FLASH DED 状态。
FLASH_ERR_08	FLASH 模块
类别	功能
功能	不会为典型的无效存储器区域生成硬故障
说明	在尝试访问如下所示的非法存储器地址空间时，不会生成硬故障：1.0x010053FF - 0x20000000 2.0x40BFFFFFF - 0x41C00000 3.0x41C007FF - 0x41C40000
权变措施	否
FLASH_ERR_09	FLASH 模块
类别	功能

FLASH_ERR_09

(续)

FLASH 模块

功能

在特定的闪存交换策略下，BANK1 的静态写保护在 bank1 擦除时无法正常工作

说明

如果器件闪存并非最大型号 (例如，MSPM0L111x 系列有 MSPM0L1117 和 MSPM0L1116 两种不同型号的器件，只有 MSPM0L1116 存在此问题)，将出现在以下情况时意外擦除 bank1 的问题：失败情况 1：BANK1 的静态写保护已启用，闪存交换策略已禁用，SECCFG_Reg.FLBANKSWP 寄存器的 USEUPPER 位为 0 或 1，通过存储体擦除命令意外擦除了 BANK1。失败情况 2：BANK1 的静态写保护已启用，闪存交换策略已启用，SECCFG_Reg.FLBANKSWP 寄存器的 USEUPPER 位为 0，通过存储体擦除命令意外擦除了 BANK1。

权变措施

1.对于 Bank1，使用扇区擦除代替存储体擦除。或 2.使用同一器件系列中闪存最大的器件。

GPIO_ERR_05

GPIO 模块

类别

功能

功能

当 DMA 传输正在进行时，可能会错过对 GPIO DOUTTGL 寄存器的写入

说明

当并发 DMA 传输正在进行时，GPIO DMAMASK 寄存器信息被错误地应用于 CPU 对 DOUTTGL 寄存器的写入。

权变措施

在应用程序代码中，确保针对 DOUTTGL 寄存器中的相应位将 GPIO DMAMASK 位设置为 1，然后再发出 CPU 对 DOUTTGL 寄存器的写入访问权限。如果不需要向任何 GPIO 寄存器进行 DMA 传输，则 GPIO DMAMASK 可在 IO 初始化步骤期间配置为 0xFFFFFFFF。这将解决此勘误表中的冲突。如果应用程序还需要对 GPIO 寄存器进行 DMA 写入传输，建议应用程序不要同时使用 DMA 和 CPU 来写入器件中同一 GPIO 模块的 DOUTTGL 寄存器。如果器件具有多个 GPIO 模块，则 DMA 和 CPU 可以同时向不同 GPIO 模块的 DOUTTGL 寄存器写入数据 (同时仍要求为 CPU 要写入的 GPIO 模块配置 GPIO DMAMASK)。

GPIO_ERR_06

GPIO 模块

类别

功能

功能

当 DMA 传输正在进行时，对 GPIO DOUT、DOUTSET 和 DOUTCLR 寄存器的写入可能会丢失

说明

DMA 无法访问 GPIO DOUT、DOUTSET 和 DOUTCLR 寄存器。由于实现中的错误，当并发 DMA 传输正在进行时，CPU 对 GPIO DOUT、DOUTSET 和 DOUTCLR 的访问也将被阻止。

GPIO_ERR_06

(续)

GPIO 模块**权变措施**

在应用程序代码中，软件不应写入 DOUT、DOUTSET 和 DOUTCLR 寄存器，而是应对 DOUTTGL 寄存器执行等效写入（有关 CPU 写入 DOUTTGL 寄存器的限制，请参阅权变措施 GPIO_ERR_05）。

在下面的伪代码中，“pins”表示待配置的 GPIO 模块中引脚的位向量。

```
DL_GPIO_setPins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & pins;
}

DL_GPIO_clearPins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = gpio->DOUT31_0 & pins;
}

DL_GPIO_writePins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & pins;
    gpio->DOUTTGL31_0 = gpio->DOUT31_0 & (~pins);
}

DL_GPIO_writePinsVal(GPIO_Regs* gpio, uint32_t pinsMask, uint32_t pinsVal)
{
    uint32_t doutVal = gpio->DOUT31_0;
    doutVal &= ~pinsMask;
    doutVal |= (pinsVal & pinsMask);
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & doutVal;
    gpio->DOUTTGL31_0 = gpio->DOUT31_0 & (~doutVal);
}
```

I2C_ERR_04**I2C 模块****类别**

功能

功能

当 SCL 为低电平且 SDA 为高电平时，目标 i2c 无法释放延展。

说明

- 1：SCL 线路接地并释放，器件不限期将 SCL 拉至低电平。
- 2：时钟后延展、超时和释放；如果线路上存在另一个时钟低电平，器件会不限期地将 SCL 拉至低电平。

权变措施

如果 I2C 目标应用在低功耗模式下不需要使用异步快速时钟请求进行数据接收，则建议默认禁用 SWUEN，包括在复位或功率周期期间也是一样。在这种情况下，不会出现错误描述 1 和 2。

如果 I2C 目标应用需要使用异步快速时钟请求在低功耗模式下接收数据，请在进入低功耗模式之前启用 SWUEN，并在退出低功耗模式后清除 SWUEN。即使在这种情况下，当 I2C

I2C_ERR_04 (续) I2C 模块

目标处于低功耗模式时，也可能出现错误描述 1 和 2，如果总线上的另一个器件引发连续时钟延展或超时，它也会无限期延长 SCL 线路。为了从这种情况中恢复，请在 I2C 目标器件上启用低电平超时中断，在低电平超时 ISR 内复位并重新初始化 I2C 模块。

I2C_ERR_05 I2C 模块

类别

功能

功能

如果我们在正在进行的事务期间切换 ACTIVE 位，I2C SDA 可能会卡在零电平的位置

说明

如果在正在进行的传输期间切换 ACTIVE 位，则其状态机将复位。但是，由主机驱动的 SDA 和 SCL 输出将不会复位。存在 SDA 为 0 且主机已进入 IDLE 状态的情况，在这种情况下，主机将无法从 IDLE 状态向前移动或更新 SDA 值。设置从机的 BUSBUSY (切换 ACTIVE 位会导致在线路上开始检测到开始)，并且 BUSBUSY 不会被清除，因为主机将无法驱动 STOP 来将其清除。

权变措施

在事务正在进行期间不要切换 ACTIVE 位。

I2C_ERR_06 I2C 模块

类别

功能

功能

当 I2C 时钟低于 24kHz 及更低时，SMBus 高电平超时功能会失败

说明

当 I2C 时钟速率低于 24kHz 及更低 (20kHz、10kHz) 时，SMBus 高电平超时功能会失败。根据 SMBUS 规格，活动事务期间 SCL 高电平时间的上限为 50us。从将 START MMR 位写入 SCL 低电平所需的总时间为 60us，即 >50us。它将触发超时事件，并使 I2C 主机进入 IDLE 状态，而不会在传输本身开始时完成事务。以下是详细说明。对于 SCL 配置为 20kHz 的情况，SCL 低电平周期和高电平周期分别为 30us 和 20us。首先，在高电平超时计数器开始递减的同时开始 MMR 位写入。然后，从开始 MMR 位写入到 SDA 变为低电平 (启动条件) 需要一个 SCL 低电平 (30us)。接下来，从 SDA 变为低电平 (启动条件) 到 SCL 变为低电平 (数据传输开始) 需要另一个 SCL 低电平周期 (30us)，此时应该停止高电平超时计数器。从计数器开始到结束总共需要 60us 的时间。但是，由于高电平超时计数器的上限 (50us)，尽管 I2C 事务会正常工作，而且不出现问题，但仍将触发超时事件。

权变措施

当 I2C 时钟速率低于 24KHz 及更低时，请勿使用 SMBus 高电平超时功能。

I2C_ERR_07 I2C 模块

类别

功能

I2C_ERR_07 (续) I2C 模块

功能	背对背控制器控制寄存器的写入将导致 I2C 无法启动。
说明	背对背 CTR 寄存器写入将导致后续的 CTR.START 不会正确地引起启动条件。
权变措施	以单次写入方式写入所有 CTR 位 (包括 CTR.START)，或者在 CTR 写入和 CTR.START 写入之间等待一个时钟周期。

I2C_ERR_08 I2C 模块

类别	功能
功能	RXDONE 中断后直接读取 FIFO 会导致读取错误数据
说明	发生 RXDONE 中断时，FIFO 并不总是会更新为最新数据。
权变措施	等待 2 个 I2C CLK 周期，让 FIFO 确保获得最新数据。I2C CLK 基于 I2C 寄存器中的 CLKSEL 寄存器。

I2C_ERR_09 I2C 模块

类别	功能
功能	如果以低速运行 I2C，则 ISR 读取时可能不会及时更新起始地址匹配状态。
说明	如果以低于 100kHz 的 I2C 速度运行，可能无法及时将 ADDRMATCH 位 (TSR 寄存器中的地址匹配) 置位来通过中断读取。
权变措施	如果在 I2C 上以低于 100kHz 的速度运行，请等待至少 1 个 I2C CLK 周期，然后再读取 ADDRMATCH 位。

I2C_ERR_10 I2C 模块

类别	功能
功能	启用 I2C 忙碌状态，防止进入低功耗模式
说明	在 I2C 目标模式下，如果没有 STOP 位，I2C 忙碌状态会在事务后保持高电平。

I2C_ERR_10 (续) I2C 模块

权变措施

对 I2C 控制器进行编程，以发送 STOP 位并且不为最后一个字节发送 NACK。通过 STOP 条件终止任何 I2C 传输，以保持正确的 BUSY 状态和异步时钟请求行为（用于重新进入低功耗模式）。

I2C_ERR_13 I2C 模块

类别

功能

功能

轮询 I2C BUSY 位可能无法保证控制器传输完成

说明

在设置 CCTR.BURSTRUN 位来启动 I2C 控制器传输后，大约需要 3 个 I2C 功能时钟周期才能将 BUSY 状态置为有效。如果在设置 CCTR.BURSTRUN 后立即轮询 BUSY 位以等待传输完成，可能会在 BUSY 状态尚未置位时就完成状态检查。在 CLKDIV 值较高（导致 I2C 功能时钟较慢）或编译器优化级别较高的情况下，更有可能发生该问题。

权变措施

在轮询 BUSY 状态之前添加软件延迟。软件延迟 = $3 \times \text{CPU CLK} / \text{I2C 功能时钟} = 3 \times \text{CPU CLK} / (\text{CLKSEL} / \text{CLKDIV})$ 例如，时钟分频器 (CLKDIV) 为 8、时钟源为 4MHz(MFCLK)，CPU CLK 为 32MHz 时：软件延迟 = $3 \times 32\text{MHz} / (4\text{MHz} / 8) = 192$ 个 CPU 周期

I2C_ERR_15 I2C 模块

类别

功能

功能

I2C SDA 上的干扰会导致 I2C 外设低功耗模式下处于活动状态

说明

I2C SDA 上 4us 内的干扰会导致 I2C 外设开关在低功耗模式 (STOP/STANDBY) 下进入活动状态，并且无法返回低功耗模式。

权变措施

1.增加 I2C 总线上的电容，但确保其总电容不超过 I2C 标准允许的值。2.增加 I2C 总线的电压。3.将 I2C 线移至远离噪声源。4.定期唤醒器件以检查 I2C 外设状态，如果其未处于 IDLE 状态，则复位 I2C 并将其重新初始化。

KEYSTORE_ERR_01 密钥库模块

类别

功能

功能

STATUS.STAT 值可以是 0 或 1，无需密钥访问

说明

STATUS.STAT 的复位值为 1 并在以下条件下变为 0：1.复位后，调试器通过寄存器窗口访问将返回 0x00。2.复位后，第一次 CPU 读取将返回 0x01，后续 CPU 读取将返回 0x00。3) 复位后，首先读取任何其他密钥库寄存器，然后读取 STATUS.STAT 将返回 0x00。

KEYSTORE_ERR_

01 (续)

密钥库模块
权变措施

STATUS.STAT = 0x0 表示“无错误”。要检查插槽是否有效 (是否有密钥), 请检查 STATUS.VALID。

PMCU_ERR_10

PMCU 模块
类别

功能

功能

在某些工作条件下, VBOOST 可能会存在更高的延迟

说明

模拟多路复用的 VBOOST 在 $VDD < 1.8V$ 时具有较大的延迟, 这会延迟 HFXT、COMP、SYSOSC (FCL 外部 R)、OPA 和 GPAMP 等其他模块的稳定时间。

权变措施

使用 GENCLKCFG[23:22]=0x2, 保持 $VDD \geq 1.8V$ 并在 ONALWAYS 模式下使用 VBOOST。

PMCU_ERR_12

PMCU 模块
类别

功能

功能

BOR 冷启动规格违例

说明

可能会违反 BOR 冷启动规格, 最大规格为 1.65V。

权变措施

无权变措施。在冷启动时, 器件会在 1.65V 以上运行。

PMCU_ERR_14

PMCU 模块
类别

功能

功能

在使能 FCL 的低功耗模式下, 由 GPIO 触发的 DMA 传输会处于挂起状态, 直到器件唤醒

说明

当使能 FCL 且 FCL 处于低功耗模式 (STOP2/STANDBY0/STANDBY1) 时, 如果 DMA 由来自 GPIO 的事件通道触发, 则该 DMA 传输在器件被事件或中断唤醒之前不会启动, 并且在唤醒前会出现额外的功耗。

权变措施

使用 GPIO 中断通过软件来启动 DMA 传输, 而非使用由 GPIO 事件触发的 DMA。退出中断后, 再次重新进入低功耗模式。

RST_ERR_01

RST 模块

类别

功能

功能

在 LFXT 或 LFCLK_IN 丢失后，NRST 释放沿与 LFOSCGOOD 上升沿重合时，器件会保持复位状态

说明

在特定的极端情况下，如果 NRST 信号的释放沿（低电平到高电平转换）与 LFOSCGOOD 的上升沿重合，则器件无法检测到 NRST 置为无效，并会无限期地保持复位状态。可能会触发该问题的情况：当 LFCLK 源从 LFCLK_IN 或 LFXT 切换到 LFOSC 时，LFOSC 将重新启用，并且 LFOSCGOOD 状态在重新启用事件的 250us（最小值）至 1.5ms（最大值）内置为高电平。如果施加的 NRST 低电平脉冲使其释放沿（低电平到高电平转换）在 50ns 窗口内与 LFOSCGOOD 上升沿对齐，则不会检测到 NRST 置为无效，且器件保持复位状态。

权变措施

保持 NRST 低电平脉冲宽度高于 2ms 即可避免此问题

RST_ERR_02

RST 模块

类别

功能

功能

当 NRST 置为无效与初始 LFOSCGOOD 置为有效重合时，器件可能无法上电

说明

在上电事件中，一旦内部内核电压 (VCORE) 达到稳定电平，就会启用内部低频振荡器 (LFOSC)。LFOSCGOOD 信号在 VCORE 达到稳定后的 250us（最小值）至 1.5ms（最大值）内置为高电平。VDD 的 VBOR+ 阈值可用作 VCORE 变为稳定时的基准点。如果 NRST 置为无效（低电平到高电平转换）与 LFOSCGOOD 上升沿在 200ns 窗口内重合，则不会检测到 NRST 置为无效，且器件保持复位状态，无法退出启动序列的引导阶段。由于 NRST 置为无效时序由 NRST 电路上的 VDD 斜坡速率以及外部电阻器和电容器 (RC) 网络控制，因此实际上可能会触发这种情况 — 已知易受影响的配置包括 NRST 电路上的 47k Ω 电阻器和 10nF 电容器。

权变措施

权变措施 1：在 RC 滤波器中使用足够小的上拉电阻器，以确保 NRST 信号在 LFOSCGOOD 最小置为有效时间 (250us) 之前至少 50us 上升到 VDD 的 90%，从而确保 NRST 置为无效沿与 LFOSCGOOD 置为有效窗口之间有足够的间隔。已知符合标准的配置是 NRST 电路上的 1k Ω 电阻器以及 10nF 电容器。Workaround2：采用一个外部复位控制器，使 NRST 在 VDD 斜坡开始后保持在逻辑低电平至少 2.0ms，从而确保 NRST 置为无效发生在 LFOSCGOOD 最大置为有效时间 (1.5ms) 之后。

SPI_ERR_02

SPI 模块

类别

功能

功能

从低功耗模式 (LPM) 唤醒后 SPI 时钟和数据字节缺失

SPI_ERR_02 (续) SPI 模块

说明

器件从低功耗状态唤醒后，SPI 模块无法正确传播发送出的第一个字节的前几个时钟周期和数据位。

权变措施

为了在唤醒后确保 SPI 数据完整性，请在进入和退出 LPM 时使用以下序列：

1. 禁用 SPI 模块
2. 等待中断 (WFI) - 进入 LPM
3. 从 LPM 唤醒 (任何源)。
4. 启用 SPI 模块。

SPI_ERR_04 SPI 模块

类别

功能

功能

当 SPI 外设处于仅接收模式时，在每个帧接收后进行 IDLE/BUSY 状态切换。

说明

如果 SPI 外设处于仅接收模式，则在 SPI 连续接收数据 (SPI_PHASE=1) 时，每个帧接收后都会切换 IDLE 中断和 BUSY 状态。此处没有数据加载到外设 (从机) TXFIFO，且 TXFIFO 为空。

权变措施

不要使用 SPI 外设的仅接收模式。将 SPI 外设 (从机) 设置为同步传输和接收模式。

SPI_ERR_05 SPI 模块

类别

功能

功能

无论 RXFIFO 数据如何，都将设置 SPI 外设接收超时中断

说明

使用 SPI 超时中断时，无论 RXFIFO 中是否存在数据，RXTIMEOUT 计数器都会从外设停止接收 SPI 时钟起开始递减，并设置 RXTIMEOUT 中断，这与 TRM 中的说明不匹配：当接收 FIFO 不为空，且在 CTL1.RXTIMEOUT 所设定的时间内未再接收到任何数据时，SPI 外设接收超时 (RTOUT) 中断有效。

权变措施

在接收 FIFO 为空时，持续重复加载 RXTIMEROUT 计数器的值；仅当接收 FIFO 中存入任意数据后，才开始超时计数。

SPI_ERR_06 SPI 模块

类别

功能

功能

调试暂停置为有效时，IDLE/BUSY 状态不反映 SPI IP 的正确状态

SPI_ERR_06 (续) SPI 模块

说明

IDLE/BUSY 与暂停无关，它仅控制 RXFIFO/TXFIFO 写入/读取选通。因此，如果控制器正在发送数据，尽管未在 FIFO 中为其设置门锁，但正在设置 BUSY。在暂停期间，POCI 线路会在线路上传输先前传输的数据

权变措施

当 SPI IP 暂停时，请勿使用 IDLE/BUSY 状态。

SPI_ERR_07 SPI 模块

类别

功能

功能

如果同时在 SPI 外设对 TXFIFO 进行读取/写入，则可能不会生成 SPI 下溢事件

说明

当 SPI.CTL0.SPH = 0 且器件配置为 SPI 外设时。

如果在从 SPI 控制器发出读取请求时对 TXFIFO 进行了写入，则由于同时发生读取/写入请求，可能不会生成下溢事件。

权变措施

当 SPI 控制器对器件寻址时，确保 TXFIFO 不为空，这可以通过预加载数据来实现，以避免对同一 TXFIFO 地址进行写入和读取。或者，可以使用数据检查策略（如 CRC）来验证数据包是否已正确发送，在 CRC 不匹配时，可以重新发送数据。

SPI_ERR_10 SPI 模块

类别

功能

功能

DMA 未对最新 SPI 触发计数进行采样

说明

如果 SPI 配置为在发生接收超时 (RTOUT) 事件时触发 DMA 传输，且 DMA 通道在触发请求时正忙于为另一个传输提供服务，则在 DMA 应答请求之前 SPI 接收到的任何额外字节都不会包含在传输中。DMA 仅传输发出触发请求时接收到的字节数，使随后接收到的字节仍保留在 RX FIFO 中未被读取。

权变措施

将 DMA_TRIG_RX 配置为 RX 触发条件与 RTOUT 结合使用。当 RX FIFO 达到配置的阈值水平时，就会触发 RX 触发，从而确保在 DMA 服务开始之前，FIFO 中已存在全部预期字节数。这可保证无论 DMA 通道是否延迟应答请求，DMA 传输计数均准确无误。

SYSCTL_ERR_01 SYSCTL 模块

类别

功能

功能

SW-POR 功能与 HW-POR 结合使用

SYSCCTL_ERR_01

(续)

SYSCCTL 模块**说明**

当用户使用正确的密钥写入 LFSSRST 寄存器以生成软件触发的 POR 时，RSTCAUSE 寄存器将显示 0x2 (指示 NRST 触发的 POR)，而不是预期的 0x3 (软件触发的 POR)。这是因为 SW-POR 功能是与 HW-POR 路径结合使用的。

权变措施

否

SYSCCTL_ERR_02**SYSCCTL 模块****类别**

功能

功能

SYSSTATUS.FLASHSEC 在 BOOTRST 之后为非零

说明

在 BOOTRST/ 引导代码运行完成之后，SYSSTATUS.FLASHSEC 为非零。客户将在引导代码运行完成后看到此情况。

权变措施

否

SYSCCTL_ERR_03**SYSCCTL 模块****类别**

功能

功能

在执行 SYSRESET 或对 SYSSTATUSCLR 进行写入后，DEDERRADDR 仍然存在

详细信息

在执行 SYSRESET 或对 SYSSTATUSCLR 进行写入后，DEDERRADDR 寄存器仍然存在。仅当发生新的 FLASHDED 错误时，才会覆盖其值。这种行为不符合技术参考手册 (TRM)，手册中规定其初始复位值为零。

权变措施

无权变措施

SYSCCTL_ERR_04**SYSCCTL 模块****类别**

功能

功能

SYSRESET 之后不会清除 SYSSTATUS.FLASHSEC

说明

SYSRESET 之后不会清除 SYSSTATUS.FLASHSEC，只能通过写入 SYSSTATUSCLR 寄存器来清除。

权变措施

从 SYSRESET 返回后，使用 SYSSTATUSCLR = 0xCE000001 手动清除 FLASHSEC 状态。

SYSCTL_ERR_05 *LFCLK 模块*

类别	功能
功能	LFCLK 在退出关断模式时无法正常工作
说明	如果 LFCLK_IN 引脚配置为具有上拉的通用输入 (或) LFCLK_IN 功能，则在该配置中，退出关断模式将导致 LFCLK 卡住。
权变措施	选择以下任一选项：1. 在该 LFCLK_IN I/O 上启用下拉而不是上拉 2. 避免将其配置为输入

SYSCTL_ERR_06 *CLK_OUT 模块*

类别	功能
功能	当用户禁用 CLK_OUT 而异步时钟被选为 CLK_OUT 源时，外部时钟输出 (CLK_OUT) 上可能会出现干扰
说明	如果 CLK_OUT 的时钟源与当前总线时钟异步 (例如，总线时钟为 SYSOSC、而 CLK_OUT 的时钟源被选为 LFCLK)，则在这种情况下，当 CLK_OUT 引脚在启用一段时间后被禁用时，CLK_OUT 引脚上可能会出现干扰
权变措施	为避免外部引脚的干扰输出，请按照以下顺序操作：CLK_OUT 启用情况：IOMUX 启用配置应在 CLK_OUT 启用配置之后。CLK_OUT 禁用情况：IOMUX 禁用配置应在 CLK_OUT 禁用配置之前。

SYSCTL_ERR_10 *SYSCTL 模块*

类别	功能
功能	使用 ULPClk 4MHz 时钟进入 LPM 时，需要较长的唤醒时间
说明	当使用 ULPClk 4MHz (如 STOP0) 进入时，唤醒时间会比典型值额外增加约 10us。
权变措施	当 ULPClk 为 4MHz 时，避免直接进入低功耗模式。MCU 可以先唤醒至 RUN0 模式，然后再进入 LPM。

SYSOSC_ERR_01 *SYSOSC 模块*

类别	功能
功能	将 SYSOSC FCL 与 STOP1 模式一起使用时的 MFCLK 漂移

SYSOSC_ERR_01

(续)

SYSOSC 模块

说明

如果启用了 MFCLK 并且 SYSOSC 使用频率改正环路 (FCL) 模式并使用 STOP1 低功耗工作模式，则当 SYSOSC 从 4MHz 转换回 32MHz 时 (在退出 STOP1 进入 RUN 模式时或发生强制 SYSOSC 进入 32MHz 的异步快速时钟请求时)，MFCLK 可能会漂移两个周期。

权变措施

使用 STOP0 模式而非 STOP1 模式。使用 STOP0 模式时不会出现 MFCLK 漂移。

或

在使用 STOP1 时，在 FCL 模式下不使用 SYSOSC (使 FCL 保持禁用状态)。

SYSOSC_ERR_02 **SYSOSC 模块**

类别

功能

功能

MFCLK 在某些情况下无法启动。

说明

在以下情况下，MFCLK 不会开始切换：

1. 启用 FCL 模式，然后启用 MFCLK
2. 进入禁用 SYSOSC 的低功耗模式(SLEEP2/STOP2/STANDBY0/STANBY1)。
3. 当前已收到来自某个外设的异步时钟请求，该外设使用 MFCLK 作为其功能时钟。

权变措施

避免上述情况 - 在启用异步快速时钟请求的情况下，检查您的系统是否未使用 FCL 和 MFCLK 以及列出的低功耗模式。

SYSOSC_ERR_04 **SYSOSC 模块**

类别

功能

功能

在 FCL ON 模式下，SYSOSC 精度降低

说明

当使用内部振荡器 (SYSOSC) 的 FCL ON 模式时，精度可能会降低多达 +/-3%。精度下降是由 4MHz FCL 采样时钟与在 4MHz 谐波下运行的其他外设产生的系统噪声之间发生同步所致。

权变措施

目前没有解决方法可以完全消除系统中的噪声。但是，如果使用 FCL ON 模式，可通过让外设运行于非 4MHz 倍数频率，将精度降低程度降至最低。

SYSOSC_ERR_05 **SYSOSC 模块**

类别

功能

功能

启用 FCL 时，SYSOSC 在从 LPM 进行异步快速唤醒期间的运行频率可能低于基频

说明

FCL=enabled 时，在低功耗模式下且存在活动的异步快速时钟请求期间，SYSOSC 的运行频率可能比其基频最多低 10%。当器件处于低功耗模式时会发生这种情况，原因是即使 FCL = enabled，也始终会应用 FCL = Disabled 微调，而不是使用 FCL= enabled 微调。器件唤醒、退出 LPM 并处理请求后，SYSOSC 将应用正确的微调，并以其预期的目标频率恢复正常的 FCL = Enabled 运行。

大多数用例不会产生实质性影响，用户仍可根据需要继续同时使用 FCL 和异步快速时钟请求。此勘误表会产生实质性影响的主要应用是 UART 为异步快速时钟请求源的应用，因为这种临时移位可能会影响有效波特率，直至器件被完全唤醒。

SYSOSC_ERR_05

(续)

SYSOSC 模块**权变措施**

1. 保持 FCL = disabled
2. 如果需要 FCL = Enabled，则禁用异步快速时钟请求。

TIMER_ERR_04**TIMER 模块****类别**

功能

功能

如果接近零事件，则可能会错过计时器的重新启用

说明

在单次模式下使用 GPTIMER 且 CLKDIV.RATIO 不为 0 时，如果在接近零事件，则可能会错过计时器的重新启用。

权变措施

计时器可以先禁用，然后再重新启用。

TIMER_ERR_06**TIMER 模块****类别**

功能

功能

向 CLKEN 位写入 0 不会禁用计数器

说明

向计数器时钟控制寄存器 (CCLKCTL) 时钟使能位 (CLKEN) 写入 0 不会停止定时器。

权变措施

通过向计数器控制 (CTRCTL) 使能 (EN) 位写入 0 来停止定时器。

TIMER_ERR_07**TIMG 模块****类别**

功能

功能

初始重复计数器的周期比下一个重复少 1 个

说明

使用计时器重复计数器模式时，第一次重复的计数将比后续重复的计数少 1，因为以下重复计数器将包括 0 和加载值之间的转换。例如，如果 TIMx.RCLD = 0x3，则第一个重复计数器上将出现 3 个可观察到的零事件，并在以下重复计数器序列上显示 4 个可观察到的零事件。

权变措施

将初始 RCLD 值设置为比预期的 RCLD 大 1，然后在重复计数器归零事件 (REPC) 的 ISR 中将 RCLD 设置为预期的 RCLD 值。

例如，如果打算重复 4 次，请将初始 RCLD 值设置为 RCLD = 0x5，然后在 REPC 中断的计时器 ISR 中设置 RCLD = 0x4。现在，所有计时器重复将具有相同数量的零/加载事件。

UART_ERR_01	UART 模块
类别	功能
功能	在切换到 STANDBY1 模式时，未检测到 UART 启动条件
说明	器件处于 STANDBY1 模式时，由 UART 传输启动的异步快速时钟请求提供服务后，器件将返回 STANDBY1 模式。如果在转换回 STANDBY1 模式期间开始另一次 UART 传输，则器件无法正确检测到并接收数据。
权变措施	当预计存在重复的 UART 启动条件时，使用 STANDBY0 模式或更高的低功耗模式。
UART_ERR_02	UART 模块
类别	功能
功能	仅启用 TXE 时，不设置 UART 传输结束中断
说明	当器件设置为仅传输 (CTL0.TXE = 1, CTL0.RXE = 0) 时，不会触发 UART 传输结束 (EOT) 中断。当器件设置为传输和接收 (CTL0.TXE = 1, CTL0.RXE = 1) 时，EOT 会成功触发
权变措施	当使用 UART 传输结束中断时，设置 CTL0.TXE 和 CTL0.RXE 位。请注意，您不需要将引脚分配为 UART 接收。
UART_ERR_04	UART 模块
类别	功能
功能	当时钟从 SYSOSC 转换到 LFOSC 时，通过快速时钟请求接收到的错误 UART 数据会被禁用
说明	场景： 1.已选择 LFCLK 作为 UART 的功能时钟。 2.波特率为 9600，配置了 3 倍过采样 3.UART 快速时钟请求已被禁用 。如果在 UART RX 传输过程中 ULPCLK 从 SYSOSC 更改为 LFOSC，则会观察到一个位读取不正确
权变措施	在 LPM 模式下使用 UART 时启用 UART 快速时钟请求。

UART_ERR_05	UART 模块
类别	功能
功能	UART 模块中调试暂停功能的限制
说明	所有 Tx FIFO 元素都在通信进入暂停状态之前发出，预计完成现有帧并暂停。
权变措施	调试暂停置为有效后，请确保数据不会写入 TX FIFO。
UART_ERR_06	UART 模块
类别	功能
功能	UART 9 位模式下的 RTOUT/忙碌/异步异常行为
说明	在多节点场景中，UART 接收超时 (RTOUT) 无法正常工作，其中一个 UART 将用作控制器，其他 UART 节点作为外设，在 9 位 UART 模式下为每个外设配置不同的地址。第一个 UART 控制器与 UART 外设 1 通信，通过发送外设 1 的地址作为第一个字节，然后发送数据，外设 1 已看到地址匹配并接收到数据。控制器处理好外设 1 后，外设 1 在配置的超时期间后不设置 RTOUT、如果控制器立即开始与另一个 UART 外设 (外设 2) 的通信，该外设 1 在总线上配置了不同的地址。外设 1 RTOUT 计数器在与外设 2 和外设 1 通信过程中复位，仅在 UART 控制器完成与外设 2 的通信后才设置其 RTOUT。 在 BUSY 和异步请求中观察到类似行为。即使与总线上的其他外设的通信时地址不匹配，控制器也正在设置忙碌和 Async 请求。
权变措施	请勿在单个控制器连接到多个外设的多节点 UART 通信中使用 RTOUT/ BUSY / 异步时钟请求行为。
UART_ERR_07	UART 模块
类别	功能
功能	在 IDLE LINE MODE 下，RTOUT 计数器不会按预期计数
说明	在 UART 中的 IDLE LINE MODE 下，RTOUT 计数器会卡住，即使线路处于 IDLE 状态且 FIFO 有一些元素也是如此。这意味着 RTOUT 中断在 IDLE LINE MODE 下将不起作用。如果地址不匹配，当在 Rx 线上看到切换时，将重新加载 RTOUT 计数器。 在多响应器场景中，当命令发出位置和其他某个响应器之间正在通信时，这可能会导致产生 RTOUT 事件的无限延迟。
权变措施	在 IDLELINE 模式/多节点 UART 应用中使用 UART 模块时，请勿启用 RTOUT 功能。

UART_ERR_08	UART 模块
类别	功能
功能	STAT BUSY 并不代表 UART 模块的正确状态
说明	即使 UART 模块被禁用并且 TXFIFO 中有可用数据，STAT BUSY 也会保持高电平。
权变措施	轮询 TXFIFO 状态和 CTL0.ENABLE 寄存器位以识别 BUSY 状态。
UART_ERR_10	UART 模块
类别	功能
功能	对于 UART IrDA 模式，BUSY 位设置会延迟
说明	在 IrDA 模式下，UART.STAT.BUSY 位会在 IrDA 启动脉冲的第二个边沿设置；这意味着整个位传输将在正确设置 BUSY 状态之前完成。在此期间，如果软件轮询 BUSY 位，即使 IrDA 启动脉冲正在进行，也会出现 UART 不繁忙的错误指示。BUSY 状态将受到 UART 波特率的影响，UART 传输速度越慢，正确设置 BUSY 之前的时间就越长。
权变措施	检查 BUSY 状态之前位传输长度的延迟。或者，依次检查 UART.STAT.BUSY == 0x0 和 UART.STAT.BUSY == 0x1 是另一种使动态延迟独立于波特率或其他 ISR 的权变措施。
UART_ERR_11	UART 模块
类别	功能
功能	在 STOP 位事务期间，UART 接收超时比预期更早开始计数
说明	在 STOP 位事务期间，接收超时将在 STOP 位事务的中间开始计数，如果 RXTSEL 设置太小，这可能会导致意外的 RTOUT 中断。例如，如果波特率为 1Mbps 且 RXTSEL 设置为 1，则预期的 RTOUT 应该在 STOP 位事务后 1us 发生，而不是 RTOUT 中断设置为 0.5us。
权变措施	UART.IFLS.RXTSEL 寄存器选择在接收超时 (RTOUT) 中断触发之前的位时间。RXTSEL 值需要大于 1 才能防止提前中断。接收超时时间的计算公式如下：接收超时 = (RXTSEL - 0.5)/波特率

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 NOVEMBER 30, 2025 to JULY 31, 2026 (from Revision A (November 2025) to Revision B (July 2026))
Page

• 已更新 CLK_ERR_02 类别.....	5
• 已更新 CLK_ERR_02 模块.....	5
• 已更新 CLK_ERR_02 权变措施.....	5
• 已更新 CLK_ERR_02 功能.....	5
• 已更新 CLK_ERR_02 说明.....	5
• 已更新 CPU_ERR_02 功能.....	5
• 已更新 CPU_ERR_02 说明.....	5
• 已更新 CPU_ERR_02 权变措施.....	5
• 已更新 CPU_ERR_03 权变措施.....	6
• 已更新 CPU_ERR_04 功能.....	6
• 已更新 CPU_ERR_04 说明.....	6
• 已更新 CPU_ERR_04 权变措施.....	6
• 已更新 DMA_ERR_02 模块.....	7
• 已更新 DMA_ERR_02 功能.....	7
• 已更新 DMA_ERR_02 权变措施.....	7
• 已更新 DMA_ERR_02 说明.....	7
• 已更新 FCC_ERR_01 类别.....	7
• 已更新 FCC_ERR_01 权变措施.....	7
• 已更新 FCC_ERR_01 功能.....	7
• 已更新 FCC_ERR_01 说明.....	7
• 已更新 FCC_ERR_01 模块.....	7
• 已更新 FLASH_ERR_09 功能.....	8
• 已更新 FLASH_ERR_09 权变措施.....	8
• 已更新 FLASH_ERR_09 模块.....	8
• 已更新 FLASH_ERR_09 说明.....	8
• 已更新 GPIO_ERR_06 说明.....	9
• 已更新 GPIO_ERR_06 权变措施.....	9
• 已更新 I2C_ERR_15 功能.....	13
• 已更新 I2C_ERR_15 权变措施.....	13
• 已更新 I2C_ERR_15 说明.....	13
• 已更新 PMCU_ERR_14 类别.....	14
• 已更新 PMCU_ERR_14 模块.....	14
• 已更新 PMCU_ERR_14 功能.....	14
• 已更新 PMCU_ERR_14 说明.....	14
• 已更新 PMCU_ERR_14 权变措施.....	14
• 已更新 RST_ERR_01 功能.....	15
• 已更新 RST_ERR_01 说明.....	15
• 已更新 RST_ERR_01 权变措施.....	15
• 已更新 RST_ERR_02 类别.....	15
• 已更新 RST_ERR_02 模块.....	15
• 已更新 RST_ERR_02 功能.....	15
• 已更新 RST_ERR_02 权变措施.....	15
• 已更新 RST_ERR_02 说明.....	15
• 已更新 SPI_ERR_10 模块.....	17
• 已更新 SPI_ERR_10 功能.....	17
• 已更新 SPI_ERR_10 说明.....	17
• 已更新 SPI_ERR_10 权变措施.....	17
• 已更新 SYSCTL_ERR_04 权变措施.....	18
• 已更新 SYSCTL_ERR_05 模块.....	19

• 已更新 SYSCTL_ERR_05 功能.....	19
• 已更新 SYSCTL_ERR_05 说明.....	19
• 已更新 SYSCTL_ERR_05 权变措施.....	19
• 已更新 SYSCTL_ERR_06 模块.....	19
• 已更新 SYSCTL_ERR_06 功能.....	19
• 已更新 SYSCTL_ERR_06 说明.....	19
• 已更新 SYSCTL_ERR_06 权变措施.....	19
• 已更新 SYSCTL_ERR_10 模块.....	19
• 已更新 SYSCTL_ERR_10 功能.....	19
• 已更新 SYSCTL_ERR_10 说明.....	19
• 已更新 SYSCTL_ERR_10 权变措施.....	19
• 已更新 SYSOSC_ERR_04 权变措施.....	21
• 已更新 SYSOSC_ERR_04 功能.....	21
• 已更新 SYSOSC_ERR_04 说明.....	21
• 已更新 SYSOSC_ERR_05 类别.....	21
• 已更新 SYSOSC_ERR_05 模块.....	21
• 已更新 SYSOSC_ERR_05 功能.....	21
• 已更新 SYSOSC_ERR_05 权变措施.....	21
• 已更新 SYSOSC_ERR_05 说明.....	21
• 已更新 TIMER_ERR_06 模块.....	22

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、与某特定用途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、安保法规或其他要求。

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。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月