Errata

MSPM0G511x、MSPM0G5187 微控制器

TEXAS INSTRUMENTS

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
5.1 器件编号法和修订版本标识.....3
6 公告说明.....5
7 商标.....30
8 修订历史记录.....30

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 B（原型 X 标记产品）	修订版 C
AES_ERR_01	✓	✓
BSL_ERR_01	✓	✓
BSL_ERR_02	✓	
COMP_ERR_06	✓	✓
CPU_ERR_02	✓	✓
CPU_ERR_03	✓	✓
FCC_ERR_01	✓	✓
FLASH_ERR_03	✓	✓
FLASH_ERR_04	✓	✓
FLASH_ERR_05	✓	✓
FLASH_ERR_08	✓	✓
FLASH_ERR_09	✓	✓
GPIO_ERR_05	✓	✓
GPIO_ERR_06	✓	✓
I2S_ERR_01	✓	✓
KEYSTORE_ERR_01	✓	✓
PMCU_ERR_14	✓	✓
PMCU_ERR_16	✓	✓
RST_ERR_01	✓	✓
RST_ERR_02	✓	✓

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

勘误编号	修订版 B（原型 X 标记产品）	修订版 C
SPI_ERR_10	✓	✓
SYSCTL_ERR_01	✓	✓
SYSCTL_ERR_02	✓	✓
SYSCTL_ERR_03	✓	✓
SYSCTL_ERR_04	✓	✓
SYSCTL_ERR_05	✓	✓
SYSCTL_ERR_06	✓	✓
SYSCTL_ERR_11	✓	✓
SYSCTL_ERR_12	✓	✓
SYSOSC_ERR_01	✓	✓
SYSOSC_ERR_02	✓	✓
SYSOSC_ERR_04	✓	✓
SYSOSC_ERR_05	✓	✓
SYSPLL_ERR_01	✓	
TIMER_ERR_04	✓	✓
TIMER_ERR_06	✓	✓
TIMER_ERR_07	✓	✓
UNICOMMI2CC_ERR_01	✓	✓
UNICOMMI2CC_ERR_02	✓	✓
UNICOMMI2CC_ERR_03	✓	✓
UNICOMMI2CT_ERR_01	✓	✓
UNICOMMI2CT_ERR_02	✓	✓
UNICOMMI2CT_ERR_03	✓	✓
UNICOMMSPI_ERR_01	✓	✓
UNICOMMSPI_ERR_02	✓	✓
UNICOMMSPI_ERR_03	✓	✓
UNICOMMSPI_ERR_04	✓	✓
UNICOMMUART_ERR_01	✓	✓
UNICOMMUART_ERR_02	✓	✓
UNICOMMUART_ERR_03	✓	✓
UNICOMMUART_ERR_04	✓	✓
UNICOMMUART_ERR_05	✓	✓
UNICOMMUART_ERR_06	✓	✓
UNICOMMUART_ERR_07	✓	✓
UNICOMMUART_ERR_08	✓	✓
UNICOMMUART_ERR_09	✓	✓
UNICOMMUART_ERR_10	✓	✓
UNICOMMUART_ERR_11	✓	✓
UNICOMMUART_ERR_12	✓	✓
UNICOMMUART_ERR_13	✓	✓
UNICOMMUART_ERR_14	✓	✓
UNICOMMUART_ERR_15	✓	✓
USB_ERR_01	✓	✓
USB_ERR_02	✓	✓
USB_ERR_03	✓	
USB_ERR_04	✓	✓

Copyright © 2026 Texas Instruments Incorporated

勘误编号	修订版 B（原型 X 标记产品）	修订版 C
VREF_ERR_04	✓	✓

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开发的发展阶段，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。

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。

5.1 器件编号法和修订版本标识

下面的封装图展示了封装编号法方案，表 5-1 定义了器件修订版到版本 ID 的映射。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

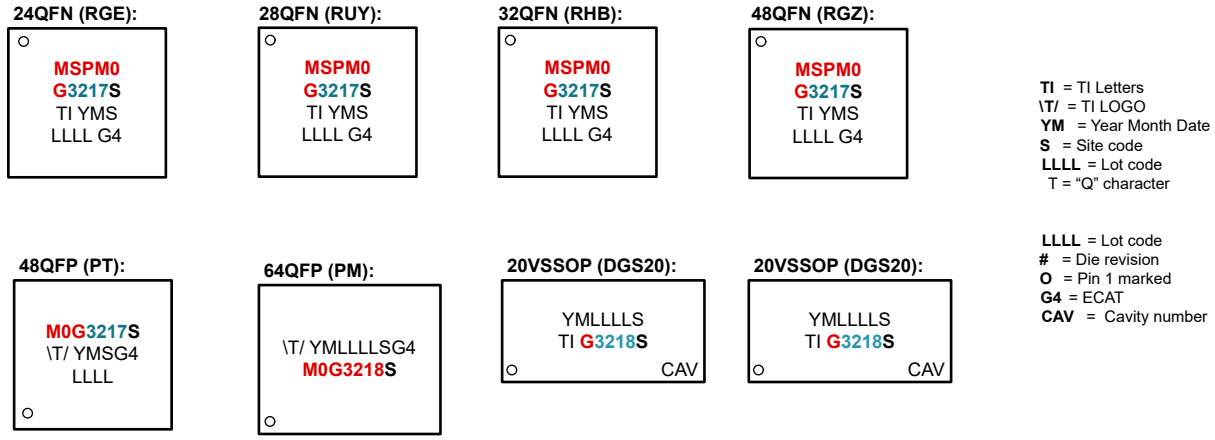


图 5-1. 封装符号

表 5-1. 芯片修订版本

修订版字母	产品开发状态	DEVICEID.VERSION
A	原型（X 标记）	0
B	RTM	0

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。该字母映射到器件出厂常量存储器中存储的整数（DEVICEID.VERSION 字段），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。

备注

DEVICEID.VERSION 字段随 RTM（按市场发布）后产品硬件修订版本递增。在相同的 VERSION 值内。用户可以通过丝印上的“X”字符识别具体型号。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

6 公告说明

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。

BSL_ERR_01 *BSL 模块*

类别

功能

功能

在某些情况下，通过应用软件调用 BSL 将失败

说明

由于 SRAM 错误代码，BSL 无法通过在应用程序内调用（BSL 软件调用）启动。复位后，由于错误，器件将返回应用程序，而非调用 BSL。

此勘误表不适用于通过硬件调用方法调用 BSL

解决方法

对于需要通过软件调用 BSL 的应用程序，请在复位器件之前用汇编代码清除整个 SRAM。MSPM0 SDK v 1.20.01.xx 或更高版本中的 MSPM0 bsl_software_invoke 示例包含“bsl_software_invoke” 示例中的修正示例。

BSL_ERR_02 *BSL 模块*

类别

功能

功能

DFU 模式下始终可用的存储器读回

说明

调用 DFU (USB BSL) 模式时，即使 BSL 未使用正确的密码解锁或在 NONMAIN 配置中禁用了读取特性，存储器回读命令也始终通过。存储器范围检查也会被旁路，用户可以读取所有存储器，包括外设、MAIN FLASH、NONMAIN FLASH 和 SRAM FLASH。

解决方法

使用辅助 BSL 覆盖安全特性。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

COMP_ERR_06 *COMP 模块*

类别

功能

功能

如果 SYSOSC 在输出就绪之前便已禁用，则 COMP 输出将不会变为就绪状态或生成中断

说明

启用 COMP 或更改任何其输入配置 (IPSEL/IMSEL) 后，一旦 COMP 输出就绪，就会触发 OUTRDYIFG 位置位。 在输出就绪之前，OUTRDYIFG 会阻断所有 COMP 中断，阻止它们触发。

COMP OUTRDYIFG 位要求 SYSOSC 至少为 4MHz 才能被置位。如果 SYSOSC 被禁用，OUTRDYIFG 将永远不会被置位，并且会永久阻断所有 COMP 中断触发。

解决方法

为了使 COMP 按预期工作，必须启用 SYSOSC，直到 OUTRDYIFG 位被置位为止。 无论何时，只要您启用 COMP 或配置 IPSEL/IMSEL，就必须先等待 OUTRDYIFG 位置位，然后才能进入低功耗模式，以禁用 SYSOSC 或将时钟更改为 LFCLK。
1.启用比较器
2.根据需要配置 IPSEL/IMSEL
3.等待 OUTRDYIFG 中断
4.进入低功耗模式 (LPM)，或根据需要将时钟更改为 LFCLK

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

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_SYSCTL_setShutdownStorageByte(DL_SYSCTL_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

Copyright © 2026 Texas Instruments Incorporated

CPU_ERR_03 (续) ***CPU 模块***

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
SYSCTL.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

FCC_ERR_01 ***FCC 模块***

类别

功能

功能

当 BUSCLK 的时钟源为 LFCLK 时，FCC 行为异常

说明

当 BUSCLK 的时钟源为 LFCLK 时，FCC 无法按预期工作。FCC 完成信号未置为有效，或者报告的 FCC 值不正确。

解决方法

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。

解决方法

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。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

FLASH_ERR_04 (续) — ***FLASH 模块***

说明

解决方法

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 2.0x40BFFFFF - 0x41C00000 3.0x41C007FF - 0x41C40000

权变措施

否

FLASH_ERR_09 — ***FLASH 模块***

类别

功能

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

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

FLASH_ERR_09
(续)

FLASH 模块

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。

解决方法

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。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

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);
}
```

I2S_ERR_01 ***I2S 模块***

类别

功能

功能

使能 I2S/TDM 模块后首次 TXIFG 中断未触发

说明

在使能 I2S/TDM 模块后，首次 TXIFG 中断不会被触发。

解决方法

向 INTCTL.INTEVAL 位写入 1，以生成初始 TXIFG 中断。

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

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

KEYSTORE_ERR_01 (续)

密钥库模块

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。

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_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。

解决方法

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。

PMCU_ERR_16

PMCU 模块

类别

功能

功能

在 3.6V 电源电压下，关断电流超出规格

说明

在 VDD 电源电压为 3.6V 的特定临界情况下，关断电流超过了数据表规格。具体来说，在 –40°C 下，关断电流 I_VDD 可能超过 200nA，而 USB 关断电流 I_VUSB 可能超过 2 µA。

解决方法

无权变措施

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

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

RST_ERR_01 (续) ***RST 模块***

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。

解决方法

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 47kΩ 电阻器和 10nF 电容器。

解决方法

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。请注意，所需的上拉最小电阻取决于外部复位控制器（如果适用）的 IO 驱动强度。已知符合标准的配置是 NRST 电路上的 4.7kΩ 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_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。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

SPI_ERR_10 (续) ***SPI 模块***

解决方法

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

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

SYSCTL_ERR_02 ***SYSCTL 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

SYSCTL_ERR_03 ***SYSCTL 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。

解决方法

无权变措施

Copyright © 2026 Texas Instruments Incorporated

SYSCTL_ERR_04 ***SYSCTL 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。

解决方法 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

解决方法 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_11 ***SYSCTL 模块***

类别 功能

功能 当器件处于 RUN2/SLEEP2 模式且启用 FCL 时，会发生意外的 BOR 复位周期

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

SYSCTL_ERR_11 (续) — ***SYSCTL 模块***

说明

当 FCL 启用且器件在 RUN2/SLEEP2 模式下运行时，BOR1/2/3 在 NMI 之后会出现意外的 BOR

解决方法

使用 BOR1/2/3 监控器时，应避免启用 FCL 并避免使器件在 RUN2 或 SLEEP2 模式下运行。

SYSCTL_ERR_12 — ***SYSCTL 模块***

类别

功能

功能

当器件处于 RUN2/SLEEP2 模式且启用 FCL 时，BOR 事件在 BOR1/2/3 之后保持已屏蔽状态

说明

当在 RUN2 或 SLEEP2 功耗模式下启用 FCL 时，BOR 屏蔽逻辑无法按预期工作，导致在首次 BOR 事件发生后，无法检测到所有后来的 BOR 事件。直到软件重新启用 SYSOSC，BOR MASK 才会被清除。

解决方法

权变措施 1：当需要 BOR1/2/3 检测时，避免启用 FCL 并使器件处于 RUN2/SLEEP2 模式 解决方法 2：在初始屏蔽周期结束后，手动清除 BOR/NMI 处理程序中的 BOR 掩码标志 解决方法 3：通过定期切换 Sysosc 启用/禁用来清除 BOR 掩码标志，实现基于软件的刷新机制

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

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。

解决方法

使用 STOP0 模式而不是 STOP1 模式，使用 STOP0 模式时没有 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）。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

SYSOSC_ERR_02 *SYSOSC 模块*

类别

功能

功能

在 LPM 下接收到异步时钟请求（在 FCL 模式下禁用了 SYSOSC）时，MFCLK 不工作

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/STANDBY1)。
3.从一些使用 MFCLK 作为功能时钟的外设接收到异步请求。
接收到异步请求时，SYSOSC 将被启用，ulpclk 将变为 32MHz。但 MFCLK 会断开，并且它根本不会切换，因为器件的设置仍然为 LPM。

权变措施

如果 SYSOSC 正在使用 FCL 模式 - 当您进入 LPM 模式（通常会关闭 SYSOSC）时，请勿启用外设的 MFCLK。

SYSOSC_ERR_04 *SYSOSC 模块*

类别

功能

功能

在 FCL ON 模式下，SYSOSC 精度会降低

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%。精度下降是由于 4MHz FCL 采样时钟与系统噪声之间的同步所致。

解决方法

如果使用 SYSPLL FCL ON 模式，请为 SYSPLL 频率选择一个非 4MHz 整数倍的值，例如：78MHz 或 79MHz
请勿将 SYSPLL 频率设为 16、32、48、40、64、80MHz 等
对于 78MHz：
将 SYSPLLCFG1.PDIV 设置为 0x3 且 SYSPLLCFG1.QDIV 设置为 38

并非 4MHz 倍数的 SYSPLL 工作频率仍会至少每微秒与 FCL 采样时钟对齐一次；当这两个时钟具有公因数时，对齐频率会更高。

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。

Copyright © 2026 Texas Instruments Incorporated

SYSOSC_ERR_05
(续)

SYSOSC 模块

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。

解决方法

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

SYSPLL_ERR_01

SYSPLL 模块

类别

功能

功能

启用后，SYSPLL 频率可能无法锁定到正确的频率。

说明

当在 SYSCTL HSCLKEN 寄存器中将 SYSPLLEN 位设置为 1 时，SYSPLL 将运行锁相环搜索。如果频率不会设置为正确的值，搜索可能会失败，相反，生成的频率将与配置的频率大不相同。

解决方法

频率验证流程
只要 SYSPLLEN 位设置为 1，就会使用频率时钟计数器 (FCC) 监控 SYSPLL 的频率输出。设置正确的频率后，它将保持稳定，直到 SYSPLL 被禁用并重新启用（SYSPLLEN 位从 0 切换到 1）。如果检测到频率不正确，请禁用并重新启用 SYSPLL 以再次执行验证。

权变措施 1：FCC 计数检查
使用 LFCLK 作为 FCC 触发源，以计算 SYSPLL 输出时钟频率。执行 FCC，并使用 LFCLK 作为参考，将测量值与配置的 SYSPLL 频率进行比对。
计算示例：
- SYSPLLCLK0 = 80MHz; LFCLK = 32.768kHz
- 测量的 FCC 计数 = 80,000,000/32,768 = 2,441
FCC 计数容许：
实际 FCC 计数会因组合的时钟精度（SYSPLLCLK0 和 LFCLK）而有所变化。建议为允许的 FCC 检查范围增加 +/- 5~10%。
- FCC 计数上限 = 2,441*1.05 = 2,563
- FCC 计数下限 = 2,441*0.95 = 2,318
时序注意事项：
- 时钟同步时间：5-6 个 LFCLK 周期
- FCC 触发时间：1-32 个 LFCLK 周期（用户可配置）
寄存器配置：
- FCC 设置：SYSCTL.GENCLKCFG.FCCTRIGSRC = 1;
- SYSCTL.GENCLKCFG.FCCLVLTRIG = 0;
- SYSCTL.GENCLKCFG.FCCTRIGCNT = 0;

Copyright © 2026 Texas Instruments Incorporated

SYSPLL_ERR_01
(续) ***SYSPLL 模块***

- SYSCTL.GENCLKCFG.FCCSELCLK = 4;
- 起始 FCC：SYSCTL.FCCCMD = 0x0E000001U
- 检查 FCC 完成状态：SYSCTL.CLKSTATUS.FCCDONE
- 读取 FCC 计数：SYSCTL.FCC

超时保护：
在监控 FCC 完成状态期间，实现软件超时机制，以避免当未锁定的 SYSPLL 频率低于 FCC 触发时钟 (LFCLK) 时等待时间延长。

```
fccTimeOutCounter = 0;
while (DL_SYSCTL_isFCCDone() == 0) {
delay_cycles(977); /* 1 个 LFCLK 周期 = 32MHz/32.768kHz */
fccTimeOutCounter++;
if(fccTimeOutCounter > 65) break;
/* 超时设置为大约 2ms（用户可配置）*/
}
```

FCC 检查重新启动：
如果 FCC 测量值超出预期范围，请禁用并重新启用 SYSPLL（将 SYSPLLEN 设为 0，再设为 1），然后重复 FCC 验证。

权变措施 2：FCC 比率检查
使用 LFCLK 作为 FCC 触发源，以计算 SYSPLL 输出和输入时钟频率。执行 FCC，并对比预期比率，验证测量的输出时钟与输入时钟的 FCC 检查值比率。

计算示例：
- SYSPLL = 80MHz；HFCLK = 40MHz；LFCLK = 32.768kHz
- 预期时钟比率 = 80MHz/40MHz = 2.0000
- 测量的 FCC 计数 (SYSPLL) = 80,000,000/32,768 = 2,441
- 测量的 FCC 计数 (HFCLK) = 40,000,000/32,768 =1,220
- 测量的时钟比率 = 2,441/1,220 = 2.0008

FCC 比率容许：
FCC 比率方法可消除组合时钟精度造成的误差，仅取决于 FCC 不确定性（2 个计数时钟周期）及计算取整误差。与 FCC 计数检查方法相比，这可实现更严格的容许范围，例如 +/- 0.3%。

时序注意事项：
- 时钟同步时间：5-6 个 LFCLK 周期
- FCC 触发时间：1-32 个 LFCLK 周期（用户可配置）
- 每次完整 FCC 比率检查的总时间：2*（同步时间 + 触发时间）

FCC 比率检查流程：
1.设置 FCC 以检测 SYSPLL 输出时钟（SYSPLL0 或 SYSPLL2X）
2.启动 FCC 并等待 FCC 完成（添加超时保护，请参阅 FCC 计数检查）
3.读取 FCC 检查计数
4.设置 FCC 以检测 SYSPLL 输入时钟（SYSOSC 或 HFCLK）
5.启动 FCC 并等待 FCC 完成（添加超时保护，请参阅 FCC 计数检查）
6.读取 FCC 检查计数
7.计算 FCC 检查比率，并与预期比率范围进行比较
8.如果 FCC 比率超出预期范围，请禁用并重新启用 SYSPLL（将 SYSPLLEN 设为 0，再设为 1），然后重复 FCC 比率验证。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

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

在单次模式下使用计时器时，如果接近零事件，则可能会错过计时器的重新启用。对计时器使能位进行硬件更新将需要一个功能时钟周期。例如，如果计时器的时钟源为 32.768kHz，时钟分频器为 3，则需要 ~100us 才能将使能位正确设置为 0。

权变措施

在重新启用计时器之前等待 1 个功能时钟周期，或者可以先禁用计时器，然后再重新启用。

通过 CTRCTL.EN = 0 禁用计数器，然后通过 CTRCTL.EN = 1 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。

解决方法

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。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMI2CC_ERR_01 — *UNICOMMI2CC 模块*

类别

功能

功能

I2C 控制器繁忙状态轮询问题

说明

当通过设置 BUSTRUN/FRAME_START 位来启动 I2C 控制器传输时，繁忙状态标志需要大约 2-3 个 I2C 功能时钟周期才能置为有效。如果在设置 BUSTRUN/FRAME_START 后立即轮询繁忙位，则应用可能会在该状态得到正确设置之前检查该状态。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，该问题会更加明显。

权变措施

在轮询繁忙状态之前添加软件延迟。建议的延迟应为：软件延迟 = 3 x I2C 功能时钟 = 3 x clock_divider x（CPU_CLK/所选时钟源频率），例如 clock_divider 为 2、时钟源为 4MHz (MFCLK)、CPU_CLK 为 80MHz：软件延迟 = 3 x 2 x (80 MHz / 4 MHz) = 120 个 CPU 周期

UNICOMMI2CC_ERR_02 — *UNICOMMI2CC 模块*

类别

功能

功能

当 I2C 控制器的 ACKOEN 位使能时，RXDONE 中断会被触发两次

说明

当 I2C 控制器的 CTR.ACKONE 位使能时，软件可以通过向 CTR.ACK 位写入 0 或 1 来控制 ACK/NACK 信号的发送。这一操作可以在停止条件发生前的 RXDONE 中断中进行。在写入 CTR.ACK 位之后，I2C 总线上会产生有效的 ACK/NACK 信号，但在随后的 I2C 停止状态下，还会再次触发一次额外的 RXDONE 中断。

解决方法

在写入 CTR.ACK 位之前，先禁用 RXDONE 中断；直到下一次传输开始前，再手动重新使能该中断。

UNICOMMI2CC_ERR_03 — *UNICOMMI2CC 模块*

类别

功能

功能

启用时钟延展功能时，I2C 控制器的 SCL 频率较低

说明

- 请务必不要建议使用 ETW 功能作为解决方法。- 请确保 ERRATA 提案不会影响此器件修订版或先前修订版的其他现有 ERRATA。启用时钟扩展功能 (CR.CLKSTRETCH = 1) 时，SCL 频率将低于设置值。例如，I2C 控制器总线速度设置为 400kHz 时，如果启用了时钟延展功能，则 I2C 控制器总线时钟将低于 400kHz（例如 360kHz 或 390kHz）。在其他总线速度（例如 100kHz 或 1MHz）下，也会出现相同现象。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMI2CC_ERR_03 (续) ***UNICOMMI2CC 模块***

解决方法

不建议禁用时钟延展功能，因为这对 I2C 通信功能不会产生很大影响。禁用时钟延展功能后，即可看到预期的 I2C 总线时钟速度。

UNICOMMI2CT_ERR_01 ***UNICOMMI2CT 模块***

类别

功能

功能

当单独写入 CTR 时，I2C 目标将保持空闲状态

说明

在以下情况中，I2C 目标器件将保持空闲状态：1.I2C 使用 MFCLK 作为时钟源。2.CLKDIV=0。3.在第一次寄存器写入中设置 CTR.START、CTR.STOP 或 CTR.BLEN 位。4.在第二次寄存器写入中设置 CTR.FRM_START。

权变措施

当 I2C 使用 MFCLK 作为时钟源且 CLKDIV=0 时，建议在一次寄存器写入中设置 CTR.START、CTR.STOP、CTR.BLEN 或 CTR.FRM_START 位。

UNICOMMI2CT_ERR_02 ***UNICOMMI2CT 模块***

类别

功能

功能

RXFIFO 数据完整性问题

说明

在 UNICOMMI2CT RX 模式下，RXDONE 中断标志在接收到的帧的最后一位之后立即置为有效。但是，RXFIFO 缓冲器内容更新会表现出相对于 RXDONE 置为有效的额外功能时钟（I2C 功能时钟）延迟。如果 CPU 在这段时间之前启动 RXFIFO 读取操作，则可能访问过时的 FIFO 数据，从而导致违反数据完整性。当 UNICOMMI2CT 功能时钟比 CPU 时钟慢得多时，此问题尤其会造成问题。

解决方法

在读取 RXFIFO 之前添加固定延迟，以确保数据可用。建议的延迟应为：软件延迟 = 1 × I2C 功能时钟 = 1 × clock_divider（CPU_CLK/所选时钟源频率），例如 clock_divider=2、时钟源=4MHz (MFCLK)、CPU_CLK=80MHz：软件延迟 = 1 × 2 × (80MHz / 4MHz) = 40 个 CPU 周期

UNICOMMI2CT_ERR_03 ***UNICOMMI2CT 模块***

类别

功能

功能

TSTART 标志在 SR ADDRMATCH 更新之前更新

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMI2CT_ER R_03 (续)

UNICOMMI2CT 模块

说明

在 UNICOMM_I2CT 中，目标 START 状态位 (RIS.TSTART) 的中断状态位在地址匹配状态 (SR.ADDRMATCH) 位之前设置 2-3 个 I2C 功能时钟周期。如果在此期间有 CPU 访问，则可能会读取过时值。当 UNICOMMI2CT 功能时钟比 CPU 时钟慢得多时，此问题尤其会造成问题。

解决方法

在读取 ADDRMATCH 之前添加固定延迟。建议读取前的软件延迟：软件延迟 = 3 × I2C 功能时钟 = 3 × clock_divider（CPU_CLK/所选时钟源频率），例如 clock_divider=2、时钟源 =4MHz (MFCLK)、CPU_CLK=80MHz：软件延迟 = 3 × 2 × (80MHz / 4MHz) = 120 个 CPU 周期

UNICOMMSPI_ER R_01

UNICOMMSPI 模块

类别

功能

功能

如果同时对 TXFIFO 进行读取/写入，则可能不会生成 SPI 下溢事件

说明

在 SPI 外设模式下运行时，存在一个竞态条件，这可能会导致状态标志不准确。此问题在以下特定条件下发生：1.发送 FIFO 变为空 2.SPI 外设接收发送数据的请求（通常来自外部控制器）3.CPU/DMA 尝试将新数据写入发送缓冲区 4.此 CPU/DMA 写入操作与控制器的新发送请求同时发生。在这些时序敏感型情况下，状态标志（例如 FIFO 空/下溢标志）可能会报告不正确的状态。这可能会导致数据传输错误或者中断丢失。此问题可能导致：错误的状态标志读数可能的数据传输错误中断生成不可靠

解决方法

要缓解此问题，请执行以下操作：1.谨慎管理 CPU/DMA 与 SPI 外设 2 之间的数据速率。对于关键应用，请考虑对到 SPI 发送缓冲区的 DMA 传输使用更高的优先级。

UNICOMMSPI_ER R_02

UNICOMMSPI 模块

类别

功能

功能

控制器端错误读取繁忙状态

说明

SPI 繁忙状态并不取决于 TXFIFO 的数据元素，因此本质上，在我们向 SPI TXFIFO 加载某些数据后，如果 CLKSEL 选择的是较慢的时钟（例如 LFCLK），或者 CLKDIV > /1，则繁忙位不会立即变为高电平。由于 SPI 源时钟较慢，这会导致在向 TXFIFO 加载数据之后到通信实际开始之前的这段时间内读取到错误的状态。

解决方法

可以在写入应用程序代码时引入对 TXFIFO 空状态位的检查。

Copyright © 2026 Texas Instruments Incorporated

UNICOMMSPI_ER R_03 ***UNICOMMSPI 模块***

类别

功能

功能

使用 CPU 清除 RXFIFO 时，第二次不会设置 RTOUT

说明

在启用 rxtimeout 的情况下完成 SPI 通信并设置 RTOUT 后，如果使用 CPU 读出 RXFIFO 数据并读取 IIDX 以清除中断，但未执行任何复位或禁用 IP 的操作，那么当第二次进行相同传输时，即使 rxtimeout 计数器为“0”且 RXFIFO 非空，也不会设置 RTOUT 中断。

解决方法

使用 DMA 读出 RXFIFO 数据，而不要直接使用 CPU。或在直接使用 CPU 读出 RXFIFO 数据后复位 UNICOMM SPI。

UNICOMMSPI_ER R_04 ***UNICOMMSPI 模块***

类别

功能

功能

在 DSS 不匹配的上一次传输之后重新进行传输时，RX 中存储的数据不正确

说明

如果控制器端与外设端使用不同的数据包大小设置 (DSS) 发送数据，随后执行 RXCLR 清除 FIFO，再以正确的数据包大小重新发送数据，在第一个 SCLK 脉冲到来时，仍会有旧数据进入 RXFIFO。

解决方法

请确保控制器端和外设端的数据包大小设置 (DSS) 相同。

UNICOMMUART_E RR_01 ***UNICOMMUART 模块***

类别

功能

功能

在 IrDA 模式中引入干扰时的数据完整性问题

说明

UNICOMM-UART 支持 IrDA，但在嘈杂的环境中有限制。由于没有干扰滤波器，当数据线路上出现干扰时，IrDA 模式可能会发生数据完整性问题。由于 IrDA 依赖于边沿检测逻辑，因此这些干扰会被误解为有效脉冲，从而导致 FIFO 中出现意外数据和 LTOUT 计算问题。

权变措施

否

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMUART_ERR_02 *UNICOMMUART 模块*

类别 功能

功能 IDLELINE 模式下有干扰时的空闲周期检测问题

说明 空闲期间的干扰可能会破坏接收器的空闲检测，从而导致 Rx 侧丢弃后续地址字节。重要的是，此故障模式与时序有关：只有在特定时间插入的相对于空闲期间的干扰才会触发问题。并非空闲期间内的每个干扰都会导致损坏。

解决方法 否

UNICOMMUART_ERR_03 *UNICOMMUART 模块*

类别 功能

功能 Manchester 模式下的干扰数据完整性问题

说明 Manchester 编码数据流中的干扰会破坏信号的精确边沿时序，从而导致错误的接收数据以及数据完整性问题。

权变措施 否

UNICOMMUART_ERR_04 *UNICOMMUART 模块*

类别 功能

功能 在 LIN 模式下的中断域和/或同步域期间，数据将因干扰而丢失

说明 中断场场景（高脉冲干扰）：在中断域周期期间，负边沿会将计数器复位为零，导致中断域检测失败并造成数据丢失。同步域方案：同步字段期间的干扰将触发错误的 LINC0/LINC1 中断，在 Rx 线路负边沿复位计数器，导致同步字段验证失败并造成数据丢失。

权变措施 否

UNICOMMUART_ERR_05 *UNICOMMUART 模块*

类别 功能

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMUART_ERR_05 (续) ***UNICOMMUART 模块***

功能

UART 干扰滤波器不存在导致的问题

说明

1. IRDA 模式数据完整性：由于没有干扰滤波器，当数据线路上出现干扰时，IrDA 模式可能会发生数据完整性问题。由于 IrDA 依赖于边沿检测逻辑，因此这些干扰会被误解为有效脉冲。2. Ideline 检测：在 IDLELINE 检测期间，信号干扰可能会通过将内部空闲线路计数器复位为零来破坏检测过程。这种干扰会导致 IDLELINE 状态标志无法正确置位，从而使模块无法识别通信线路上的有效空闲状态。但是，如果在这种中断后，空闲线路状态持续保持，且没有额外的干扰，则检测机制最终将恢复并正确检测空闲状态。3.Manchester 模式数据完整性：在曼彻斯特编码传输的数据采样期间，如果出现信号干扰，接收器将捕获错误数据，从而导致位被错误解析。这些采样错误会导致接收缓冲区中存储损坏的数据值，从而导致接收消息中出现数据完整性故障。在采样窗口之外出现的干扰通常不会影响数据解释。4. LIN 模式下 BREAK / SYNC 字段的敏感性：a. BREAK 字段：在 LIN 通信期间，如果信号干扰刚好发生在 BREAK 字段的时钟采样期间，检测机制可能会提前终止。但是，该系统具有复原力，如果中断过后 BREAK 字段的状态仍在持续且没有新的干扰出现，检测逻辑将会恢复并成功识别出 BREAK 字段。b. SYNC 字段：该系统在 SYNC 字段校准期间也表现出类似的脆弱性。在此期间，干扰会干扰下降沿和上升沿 (RXNE 和 RXPE) 的检测。5. 干扰导致的 LTOUT/RTOUT 检测问题：当空闲线路检测期间出现信号干扰时，这些干扰会暂时中断 LTOUT/RTOUT 的时间计算，并将超时计数器重置为 0。如果线路在中断后没有其他干扰，则系统将在配置的时间段之后检测到 LTOUT/RTOUT 条件。

权变措施

没有可用的解决方法。

UNICOMMUART_ERR_06 ***UNICOMMUART 模块***

类别

功能

功能

由于停止位处理而导致 RTOUT/LTOUT 计算问题

说明

在接收器侧，功能状态机在停止位中间从停止位转换为空闲。这会使接收超时 (RTOUT) 和线路超时 (LTOUT) 计数器在停止位的中间而非末端开始计数。这会导致 RTOUT/LTOUT 提前半个波特周期触发。这在 UART 功能时钟频率较高的低波特率下尤其明显。

权变措施

向 RTOUT 计数器添加具有半个停止位周期的补偿。

UNICOMMUART_ERR_07 ***UNICOMMUART 模块***

类别

功能

功能

如果在 RS – 232 模式下禁用 UART，则 RTS 线路不会变为高电平

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMUART_ERR_07 (续) — ***UNICOMMUART 模块***

说明

当 UART 被禁用时，RTS 线路无法恢复到其空闲状态（高电平），仍然卡在低电平。

权变措施

使用软件来启用内部上拉电阻，并将 RTS 线路 IO 设置为 Hiz 模式。

UNICOMMUART_ERR_08 — ***UNICOMMUART 模块***

类别

功能

功能

连续 LTOUT 中断

说明

线路超时 (LTOUT) 中断发生后，计数器会重新启动并继续计数，在每个超时到期间隔生成 LTOUT 中断。此序列也适用于 RTOUT 事件。

解决方法

在 LTOUT ISR 中禁用 LTOUT 检测。接收到新字符后重新启用。

UNICOMMUART_ERR_09 — ***UNICOMMUART 模块***

类别

功能

功能

ISO-7816 智能卡模式波特率限制

说明

为了在 ISO-7816 智能卡模式下实现 9600 波特率，由于以下限制，UARTCLK 频率需要超过 57MHz：1.ISO-7816 标准要求每位 2 有 372 个时钟周期。在 MSPM0 ISO-7816 模式下，UART 外设中的过采样率 (OVS) 固定为 16 倍 最小 UARTCLK 计算方式：所需的 UARTCLK = 9600 * 372 * 16 = 57.139MHz

权变措施

具有较低输入频率的 UART 将无法支持智能卡模式。

UNICOMMUART_ERR_10 — ***UNICOMMUART 模块***

类别

功能

功能

LIN 寄存器 CLKDIV 限制

说明

当 CLKDIV 值不是 0 时，写入 LINC0/1 将不会产生任何影响。

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

UNICOMMUART_ERR_10 (续) ***UNICOMMUART 模块***

权变措施

要正确配置 LIN 寄存器：1.首先将 CLKDIV 设置为“0” 2.使用所需的值更新 LINC0/1 寄存器配置 3.将 CLKDIV 恢复至其预期工作值

UNICOMMUART_ERR_11 ***UNICOMMUART 模块***

类别

功能

功能

UART 禁用后，BUSY 位保持设置状态

说明

即使 UNICOMMUART 被禁用，当 FIFO 中有数据时，STAT.BUSY 位也会保持高电平（置为有效）。

解决方法

在禁用 UNICOMMUART 之前，设置 IFLS.TXCLR 位以清除 TXFIFO 内容。

UNICOMMUART_ERR_12 ***UNICOMMUART 模块***

类别

功能

功能

CTL0.TXE 位在帧中间设置为 0 会导致当前传输出错

说明

开始发送数据包，然后在传输过程中，将 CTL0.TXE 位设置为 0，当前数据包传输不会完成，TX 线路将下拉至低电平，显示 BUSY 行为（表示繁忙）。仅当 CTL0.TXE = 1 时才会释放 TX 线，在此之前，BUSY 保持为 1。

解决方法

用户必须轮询 BUSY 状态以变为低电平和 CTL0.TXE = 0，以避免出现此问题。

UNICOMMUART_ERR_13 ***UNICOMMUART 模块***

类别

功能

功能

RTOUT 中断提前 1 个 fclk 周期出现

说明

RTOUT（接收超时）中断比应出现的确切时间提前了 1 个 fclk 周期。

解决方法

用户可以通过预判 RTOUT 中断会提前一个 fclk 周期出现来处理这一情况。

Copyright © 2026 Texas Instruments Incorporated

UNICOMMUART_ERR_14 — *UNICOMMUART 模块*

类别
功能

功能
CTS 中断在 CTS 线路的下降沿和上升沿均会切换

说明
CTS 中断在 CTS 线路的上升沿和下降沿均会切换。理想情况下，它应该仅在 CTS 线路的下降沿切换。

解决方法
用户可以监视 STAT.CTS 位切换为 1，而不是 RIS 位，因为 STAT.CTS = 1 的这种切换表示遇到了 CTS 线路下降。

UNICOMMUART_ERR_15 — *UNICOMMUART 模块*

类别
功能

功能
未生成 fclk 和新 IDLELINE 帧之间的 LCRH.SENDIDLE 访问丢失

说明
如果在两个 fclk 边沿之间将 LCRH.SENDIDLE 从 0 切换到 1，然后从 1 切换到 0；或者从 1 切换到 0，然后从 0 切换回 1，则将错过切换，并且不会发送新的 IDLELINE 帧。

解决方法
切换 LCRH.SENDIDLE 后，不要立即将其切回，请维持多个 fclk 周期，因为 LCRH.SENDIDLE 是在 fclk 上进行采样的。

USB_ERR_01 — *USB 模块*

类别
功能

功能
将 FLLEN 位之前的 CLKEN 位清零不会禁用 USBFLL

说明
如果在 FLLEN 位保持置位的情况下清除 CLKEN 位，USBFLL 时钟将保持其锁定状态且未按预期禁用。

解决方法
在清除 CLKEN 位之前清除 FLLEN 位，以正确禁用 USBFLL 时钟。

USB_ERR_02 — *USB 模块*

类别
功能

功能
在 IMASK、RIS 和 MIS 寄存器中交换 DMADONE 及 DMAPREIRQ 位

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

USB_ERR_02 (续) ***USB 模块***

说明

IMASK、RIS 和 MIS 寄存器包含位分配错误，其中 DMADONE 和 DMAPRERD 位的位置被错误地交换：- DMAPREIRQ 位 (19:12) 指示相应的 DMADONE 中断状态（与其预期功能相反）- DMADONE 位 (11:4) 指示相应的 DMA 前中断状态（与其预期功能相反）此位交换也会影响 IIDX 寄存器读回值。例如：BIT4 是为 DMADONEARX 设计的，它实际上表示 DMAPREARX (BIT12) - BIT5 是为 DMADONEATX 设计的，而它实际上表示 DMAPREATX (BIT13) - BIT11 是为 DMADONEDTX 设计的，而实际上表示 DMAPREDTX (BIT19)

解决方法

无权变措施。

USB_ERR_03 ***USB 模块***

类别

功能

功能

当将 SOF 设置为 FLL 基准时钟时，USB FLL 频率锁定时间会增加

说明

当 SOF（帧起始）用作 USBFLL 基准时，存在一种临界情况，USBFLL 可能需要延长时间来锁定正确的频率。测试结果表明锁定时间最长大约可达 30ms。

解决方法

1.使用 LXFT 而不是 SOF 作为 USB FLL 基准时钟源（将 USBFLLCTL.REFSEL 设置为 1h）2.选择 SYSPLL 作为 USB 时钟源（将 GENCLKCFG.USBCLKSRC 设置为 1h）

USB_ERR_04 ***USB 模块***

类别

功能

功能

配置多个端点时，端点中断可能会丢失

说明

当在配置的端点接收或发送数据时，相应的 EP 中断会记录在 INTRTX 或 INTRRX 寄存器中。这些寄存器上的任何新事件都将向 CPU 触发中断事件。ISR 处理程序需要读取 INTRTX 和 INTRRX，以识别中断来源并清除中断寄存器。中断寄存器的自动清除操作需要三个 CPU 时钟周期。在具有多个配置端点的系统中，可能会发生新的 EP 中断进入，而自动清除过程仍在进行中。因此，INTRTX 或 INTRRX 寄存器将保持置位状态，而不会识别 CPU 生成的新中断事件。

解决方法

读取 INTRTX 或 INTRRX 寄存器时，多次读取 INTRTX 或 INTRTX，以确保中断寄存器真正被清除。这样，可确保新的中断被识别为新的中断事件并正确报告给 CPU。

下面是代码实现的示例：

```
uint_fast8_t intr = 0; /* 中断状态 */
uint_fast8_t intr_reg = 0; /* 临时中断寄存器 */
intr_reg = musb_regs->intr_tx; /* 读取 intr_tx 会清空寄存器，但可能会出现新的寄存器 */
```


Copyright © 2026 Texas Instruments Incorporated

USB_ERR_04 (续) ***USB 模块***

```
while (intr_reg) { /* 多次读取，直到 intr_tx 被清除 */
intr |= intr_reg; intr_reg = musb_regs->intr_tx;
}
intr &= musb_regs->intr_txen; /* 清除禁用中断 */
/* process EP0 */
if (intr & TU_BIT(0)) {
process_ep0(rhport); intr &= ~TU_BIT(0);
}
/* process EPn */
while (intr) {
unsigned const num = __builtin_ctz(intr);
process_edpt_n(rhport, tu_edpt_addr(num, TUSB_DIR_IN));
intr &= ~TU_BIT(num);
}
```

VREF_ERR_04 ***VREF 模块***

类别

功能

功能

当比较器处于采样（超低功耗）模式时，ADC 无法使用 VREF

说明

如果比较器 (COMP) 处于采样（超低功耗）模式，则以下 2 个用例不允许 ADC 使用 VREF。
情况 1：
配置：ADC 配置为使用内部 VREF 作为其参考，而 COMP 使用 VDDA 或启用采样和保持模式的外部参考
问题：内部 VREF 模块在由 COMP 的 VDDA/外部基准请求触发时进入采样保持模式，从而使 ADC 无法使用 VREF。
情况 2：
配置：ADC 和 COMP 均配置为使用内部 VREF 作为其基准，COMP 在采样模式下运行。
问题：由于 COMP 的采样保持请求，VREF 模块会进入采样保持模式，从而阻止 ADC 访问 VREF。

解决方法

在启用 ADC 采样之前，请确保 COMP 未配置为采样（超低功耗）模式。
示例：
COMP.CTL2.REFMODE = 0;

7 商标

所有商标均为其各自所有者的财产。

8 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

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

Changes from November 30, 2025 to September 31, 2026 (from Revision * (November 2025) to Revision A (September 2026)) **Page**

- 初始发行版 1
- 添加了 BSL_ERR_01 公告 5
- 添加了 BSL_ERR_02 公告 5
- 添加了 COMP_ERR_06 公告 6
- 已更新 CPU_ERR_02 功能 6
- 已更新 CPU_ERR_02 说明 6
- 已更新 CPU_ERR_02 权变措施 6
- 已更新 CPU_ERR_03 权变措施 6
- 添加了 FCC_ERR_01 公告 7
- 添加了 FLASH_ERR_09 公告 8
- 已更新 GPIO_ERR_06 说明 9
- 已更新 GPIO_ERR_06 权变措施 9
- 添加了 I2S_ERR_01 公告 10
- 添加了 PMCU_ERR_14 公告 11
- 添加了 PMCU_ERR_16 公告 11
- 已更新 RST_ERR_01 权变措施 11
- 添加了 RST_ERR_02 公告 12
- 添加了 SPI_ERR_10 公告 12
- 已更新 SYSCTL_ERR_04 权变措施 14
- 添加了 SYSCTL_ERR_05 公告 14
- 添加了 SYSCTL_ERR_06 公告 14
- 添加了 SYSCTL_ERR_11 公告 14
- 添加了 SYSCTL_ERR_12 公告 15
- 添加了 SYSOSC_ERR_04 公告 16
- 添加了 SYSOSC_ERR_05 公告 16
- 已更新 SYSPLL_ERR_01 权变措施 17
- 已更新 TIMER_ERR_06 模块 19
- 已更新 UNICOMMI2CC_ERR_01 功能 20
- 已更新 UNICOMMI2CC_ERR_01 说明 20
- 已更新 UNICOMMI2CC_ERR_01 权变措施 20
- 添加了 UNICOMMI2CC_ERR_02 公告 20
- 添加了 UNICOMMI2CC_ERR_03 公告 20
- 已更新 UNICOMMI2CT_ERR_02 功能 21
- 已更新 UNICOMMI2CT_ERR_02 说明 21
- 已更新 UNICOMMI2CT_ERR_02 权变措施 21
- 已更新 UNICOMMI2CT_ERR_03 功能 21
- 已更新 UNICOMMI2CT_ERR_03 说明 21
- 已更新 UNICOMMI2CT_ERR_03 权变措施 21
- 已更新 UNICOMMSPI_ERR_01 说明 22
- 已更新 UNICOMMSPI_ERR_01 权变措施 22
- 添加了 UNICOMMSPI_ERR_02 公告 22
- 添加了 UNICOMMSPI_ERR_03 公告 23
- 添加了 UNICOMMSPI_ERR_04 公告 23
- 已更新 UNICOMMUART_ERR_05 功能 24
- 已更新 UNICOMMUART_ERR_05 权变措施 24
- 已更新 UNICOMMUART_ERR_05 说明 24
- 已更新 UNICOMMUART_ERR_06 功能 25
- 已更新 UNICOMMUART_ERR_06 说明 25
- 已更新 UNICOMMUART_ERR_06 权变措施 25

English Document: SLVSJM6
Copyright © 2026 Texas Instruments Incorporated

- 已更新 UNICOMMUART_ERR_08 功能 26
- 已更新 UNICOMMUART_ERR_08 说明 26
- 已更新 UNICOMMUART_ERR_08 权变措施 26
- 已更新 UNICOMMUART_ERR_09 功能 26
- 已更新 UNICOMMUART_ERR_09 说明 26
- 已更新 UNICOMMUART_ERR_09 权变措施 26
- 已更新 UNICOMMUART_ERR_10 功能 26
- 已更新 UNICOMMUART_ERR_10 说明 26
- 已更新 UNICOMMUART_ERR_10 权变措施 26
- 已更新 UNICOMMUART_ERR_11 功能 27
- 已更新 UNICOMMUART_ERR_11 说明 27
- 已更新 UNICOMMUART_ERR_11 权变措施 27
- 添加了 UNICOMMUART_ERR_12 公告 27
- 添加了 UNICOMMUART_ERR_13 公告 27
- 添加了 UNICOMMUART_ERR_14 公告 28
- 添加了 UNICOMMUART_ERR_15 公告 28
- 添加了 USB_ERR_01 公告 28
- 添加了 USB_ERR_02 公告 28
- 添加了 USB_ERR_03 公告 29
- 添加了 USB_ERR_04 公告 29
- 添加了 VREF_ERR_04 公告 30

Copyright © 2026 Texas Instruments Incorporated

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出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