Errata

MSPM0C1103、MSPM0C1104、MSPM0C1103-Q1、MSPM0C1104、MSPS003F3 和 MSPS003F4 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

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
ADC_ERR_05	✓
CPU_ERR_01	✓
CPU_ERR_02	✓
CPU_ERR_03	✓
FLASH_ERR_04	✓
FLASH_ERR_05	✓
FLASH_ERR_06	✓
FLASH_ERR_08	✓
GPIO_ERR_03	✓
GPIO_ERR_04	✓
I2C_ERR_03	✓
I2C_ERR_04	✓
I2C_ERR_05	✓
I2C_ERR_06	✓
I2C_ERR_07	✓
I2C_ERR_08	✓
I2C_ERR_09	✓
I2C_ERR_10	✓
I2C_ERR_13	✓

勘误编号	修订版 B
RST_ERR_01	✓
SPI_ERR_03	✓
SPI_ERR_04	✓
SPI_ERR_05	✓
SPI_ERR_06	✓
SPI_ERR_07	✓
SYSCTL_ERR_03	✓
SYSOSC_ERR_02	✓
TIMER_ERR_01	✓
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_09	✓
UART_ERR_10	✓
UART_ERR_11	✓

2 预编程的软件公告

影响出厂编程软件的公告。

✓ 复选标记表示问题存在于指定的版本中。

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。

勘误编号	修订版 A	修订版 B
GPIO_ERR_03	✓	✓

4 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。

4.1 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 4-1 定义了器件修订版到版本 ID 的映射。

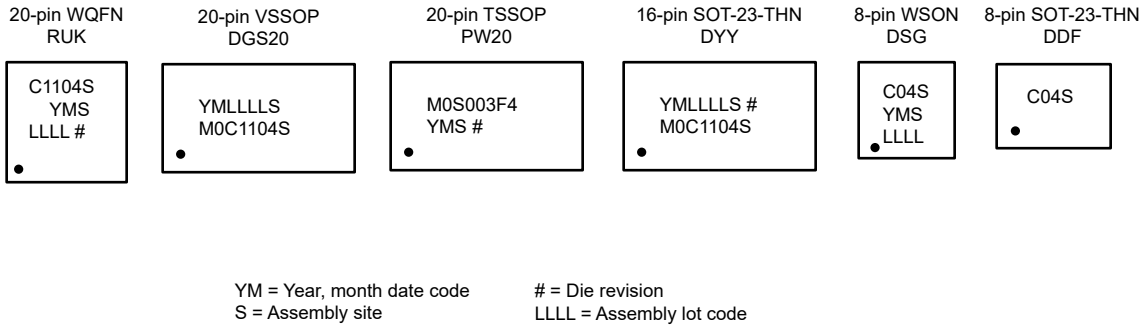

图 4-1. 封装符号

表 4-1. 芯片修订版本

修订版字母（封装标记）	版本（在器件的出厂常量存储器中）
B	2

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。

5 公告说明

5.1 已通过编译器公告修复

使用编译器解决方法解决的公告。请参阅每个公告，了解 IDE 和编译器版本及解决方法。

✓ 复选标记表示问题存在于指定的版本中。

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 收到 HW 事件触发信号时，即使 CTL0.ENC = 0x0，待处理的事件也将进入 ADC 触发队列。一旦 ADC 设置为 CTL1.TRIGSRC = 0x1（HW 触发）且 CTL0.ENC = 1，排队的 HW 触发信号将启动 ADC 采样和转换过程。即使 CTL1.TRIGSRC = 0x0（SW 触发），待处理的 HW 请求也可进入队列，但仅在 CTL1.TRIGSRC = 0x1 && CTL0.ENC = 0x1 时，才会启动 ADC 采样。

权变措施

仅当需要使用 HW 触发时，才配置 ADC F_SUB；否则，待处理的请求可能会排入队列。如果在 SW 和 HW 触发模式之间切换，ADC (RSTCTL) 上的复位将清除所有待处理的队列，但需要重新配置 ADC。

CPU_ERR_01 ***CPU 模块***

类别

功能

功能

在主闪存和其他闪存区域之间切换时，CPU 高速缓存内容可能会损坏。

说明

在访问主闪存存储器和其他非易失性内存区域（如 NONMAIN 或 Factory 区域）之间切换时，可能发生高速缓存损坏。

权变措施

使用以下步骤安全访问主内存以外的区域：
1.设置 CPUSS.CTL.ICACHE = 0x0，以禁用高速缓存。
2.从 SHUTDOWN 存储器读取 SYSCTL.SOCLOCK.SHUTDNSTORE0
3.对 NONMAIN 或 Factory 区域存储器执行所需的访问。
4.设置 CPUSS.CTL.ICACHE = 0x1 以重新启用高速缓存

CPU_ERR_02 ***CPU 模块***

类别

功能

功能

禁用 CPUSS 预取具有限制

说明

如果存在待处理的闪存访问，CPU 预取禁用将不会生效。

CPU_ERR_02
（续）

CPU 模块

权变措施

禁用预取器，然后发出对 SYSCTL 中的关断存储器 (SHUTDNSTORE) 的存储器访问权限，这可以通过 SYSCTL.SOCLOCK.SHUTDNSTORE0; 来完成
存储器访问完成后，将禁用预取器。
示例：
CPUSS.CTL.PREFETCH = 0x0; //禁用预取器
SYSCTL.SOCLOCK.SHUTDNSTORE0; //对关断存储器的存储器访问

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.PREFETCH = 0x0；//禁用预取器
SYSCTL.SOCLOCK.SHUTDNSTORE0 // 从关断存储器读取
__WFI()；//或 __WFE ()；该函数调用转换到低功耗模式
CPUSS.CTL.PREFETCH = 0x1；//启用预取器

FLASH_ERR_04

FLASH 模块

类别

功能

功能

如果错误位于 NONMAIN 或 Factory 区域，则 SYSCTL_DEDERRADDR 中会报告错误地址

说明

当出现 FLASHDED 错误时，数据会截断最高有效字节。在器件的存储器限制范围内，最高有效字节对 MAIN 闪存的返回地址没有影响。对于 NONMAIN 闪存或 Factory 区域，MSB 应列出为 0x41xx.xxxx

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。

对于主闪存 DED，可按原样使用 SYSCTL_DEDERRADDR。

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_06 *闪存模块*

类别

功能

功能

CPU 和 DMA 无法同时访问闪存

详细信息

CPU 和 DMA 无法同时访问闪存；这种同时访问会导致从闪存读取不正确的数据。

权变措施

请勿同时通过 CPU 和 DMA 访问闪存。在进行编程/擦除操作、读验证/空白验证操作以及从闪存读取 DMA 数据等典型闪存操作时，软件需确保 CPU 不会在闪存繁忙时访问闪存。为此，可在闪存操作正在进行时将代码放入 SRAM，或将闪存移入 SRAM 以读取 DMA 需要读取的数据。

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

GPIO_ERR_03 *GPIO 模块*

类别

功能

功能

调试器读取 GPIO EVENT0 IIDX 时，中断被清除。

GPIO_ERR_03
（续）

GPIO 模块

说明

在调试器读取 GPIO EVENT0 的 IIDX 时，被视为 CPU 读取，中断被清除。

权变措施

在调试期间，可通过软件读取 RIS 来读取 event0 的 IIDX。

GPIO_ERR_04

GPIO 模块

类别

功能

功能

配置全局快速唤醒可防止 GPIO 引脚将数据发送到 DIN 寄存器。

说明

当配置 CTL 寄存器的仅快速唤醒位并在运行模式下强制将数据输入 GPIO 引脚时，器件将唤醒，但 GPIO 引脚上的数据不会出现在 DIN 寄存器中。这是因为 CTL 寄存器配置阻止任何数据从 GPIO 引脚流向 DIN 寄存器。

权变措施

当预期 GPIO 引脚上的数据进入 DIN 寄存器时，应避免使用 GPIO 仅快速唤醒功能。

I2C_ERR_03

I2C 模块

类别

功能

功能

I2C 外设模式在使用 MFCLK 源时无法唤醒器件

说明

如果 I2C 模块配置为外设模式
且 I2C 采用 MFCLK（中频时钟）时钟
且器件置于 STOP2 或 STANDBY0/1 功耗模式，
则 I2C 无法在接收数据时唤醒器件。

权变措施

若需在 I2C 外设模式下通过接收数据实现低功耗唤醒，则需将 I2C 的时钟源设置为 BUSCLK，而非 MFCLK。

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。

I2C_ERR_04（续）*I2C 模块*

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 目标处于低功耗模式时，也可能会出现错误描述 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，它也会不限期延长 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，由控制器驱动的 SDA 和 SCL 输出将不会复位。存在 SDA 为 0 且控制器已进入空闲状态的情况，在这种情况下，控制器将无法从空闲状态继续运行，也无法更新 SDA 值。目标的 BUSBUSY 会置位（ACTIVE 位的切换会导致线路上检测到起始信号），而由于控制器无法驱动 STOP 信号来清除，BUSBUSY 不会被清除。

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 I2C START 位写入到 SCL 变为低电平所需的总时间为 60us，超过了 50us。它将触发超时事件，并使 I2C 控制器在传输开始时进入 IDLE 状态，而不会完成事务。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 I2C START 位写入。然后，从 START 位写入到 SDA 变为低电平（启动条件）需要一个 SCL 低电平周期 (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。

I2C_ERR_06（续）*I2C 模块*

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

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

I2C_ERR_10（续）***I2C 模块***

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。

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 x CPU CLK/I2C 功能时钟 = 3 x CPU CLK/(CLKSEL/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 x 32MHz/(4MHz/8) = 192 个 CPU 周期

RST_ERR_01 ***RST 模块***

类别

功能

功能

当 LFCLK_IN 是 LFCLK 源且 LFCLK_IN 被禁用时，不会检测到 NRST 释放

说明

当 LFCLK = LFCLK_IN 且禁用 LFCLK_IN 时，会出现一种边界场景：NRST 脉冲边沿检测失效，且器件不会退出复位。如果 NRST 脉冲宽度低于 608us，则会出现此问题。NRST 脉冲超过 608us 时，复位可正常显示。

权变措施

保持 NRST 脉冲宽度高于 608us 即可以避免此问题。

SPI_ERR_03 ***SPI 模块***

类别

功能

功能

当配置为外设时，启用 CSCLR 将导致接收到的数据在 SPH=0 模式下产生右移

SPI_ERR_03（续）*SPI 模块*

说明

在外设模式下启用 CSCLR 时，如果当 CS 处于活动或无效时 SCK 线路上存在干扰，则接收到的数据将在下一个第一帧中具有 1 位右移。此问题在 SPH=0 时的 Motorola SPI 帧格式中出现，并将影响多外设模式，该模式在 CS 无效期间会切换 SCK。

权变措施

1. 设置 CSCLR=0h。
2. 在 SPH=0 模式下设置 CSCLR=1h 时，始终丢弃第一个帧。

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 TXFIFO，TXFIFO 为空。

权变措施

不要使用 SPI 外设的仅接收模式。将 SPI 外设设置为传输和接收模式。您无需在 TX FIFO 中为 SPI 设置任何数据。

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

当使用 SPI 超时中断时，即使在接收到最终 SPI CLK 后，RXTIMEOUT 也可以继续递减，这可能会导致错误的 RXTIMEOUT。

权变措施

接收到最后一个数据包后禁用 RXTIMEOUT（可以在 ISR 中完成）并在 SPI 通信再次开始时重新启用。

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

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 中为其设置闩锁，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

SPI_ERR_06（续）*SPI 模块*

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。

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。

权变措施

无权变措施

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。

TIMER_ERR_01 *TIMx 模块*

类别

功能

功能

使用硬件事件启动计时器时，捕获模式捕获的值不正确

说明

在捕获模式下使用任何定时器实例时，使用零 (ZCOND) 或负载 (LCOND) 条件启动定时器会导致定时器捕获零值或负载值，而非捕获相应 TIMx.CC 寄存器中的值。这会影响周期和脉宽捕获等周期性用例。

权变措施

使用以下软件流程计算周期或脉冲宽度。有关权变措施示例，请参阅 MSPM0-SDK 中的 timx_timer_mode_capture_duty_and_period。

1. 通过设置为 0h 禁用 ZCOND 或 LCOND。

TIMER_ERR_01
（续）

TIMx 模块

2. 当捕获发生时，捕获值将被正确捕获到 TIMx.CC 中
3. 将 TIMx.CTR 设置为重载值（负载或 0），以重新启动定时器。

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 ***TIMG 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 ***初始重复计数器的周期比下一个重复模块少 1 个***

类别

功能

功能

计时器

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。

TIMER_ERR_07
（续）

初始重复计数器的周期比下一个重复模块少 1 个

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

UART_ERR_04
（续）

UART 模块

。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

UART_ERR_07
（续）

UART 模块

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_09

UART 模块

类别

功能

功能

当以较慢的 UART 速度运行时，UART ADDR_MATCH 可能无法在读取时及时设置。

说明

在地址匹配中断期间，当代码跳转到 ISR 并读取 FIFO 时。由于地址匹配中断在 STOP 位之前产生，UART 无法接收作为地址发送到 RX 线上的数据。

权变措施

在读取数据之前等待 1 个 UART CLK 周期，以便设置 ADDR_MATCH。

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。

UART_ERR_10
（续）

UART 模块

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 RXTOSEL 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 RXTOSEL 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.RXTOSEL 寄存器选择在接收超时 (RTOUT) 中断触发之前的位时间。RXTOSEL 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 = (RXTOSEL - 0.5)/波特率

6 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 MARCH 31, 2025 to NOVEMBER 30, 2025 (from Revision B (March 2025) to Revision C (November 2025)) **Page**

Changes from Revision A (May 2024) to Revision B (March 2025) **Page**

- 更新了器件修订版本，更新了 ADC_ERR_06，并添加了 ADC_ERR_03、ADC_ERR_05、ADC_ERR_09、CPU_ERR_01、CPU_ERR_02、I2C_ERR_05、PMCU_ERR_06、PMCU_ERR_07、PMCU_ERR_13、SPI_ERR_03、SPI_ERR_04、SPI_ERR_05、SPI_ERR_06、SPI_ERR_07、SYSOSC_ERR_02、TIMER_ERR_01、TIMER_ERR_04、TIMER_ERR_06、UART_ERR_01、UART_ERR_02、UART_ERR_04、UART_ERR_05、UART_ERR_06、UART_ERR_07 和 UART_ERR_08 1

Changes from Revision * (October 2023) to Revision A (May 2024) **Page**

- 删除了 ADC_ERR_04、ADC_ERR_05、PMCU_ERR_04、PMCU_ERR_05、PMCU_ERR_06 和 UART_ERR_01；添加了 ADC_ERR_06、I2C_ERR_03 和 PMCU_ERR_07 1

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。

最后更新日期：2025 年 10 月