

Errata

MSPM0L112x 和 MSPM0L211x 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.....2

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.....4

 5.1 器件编号法和修订版本标识.....4

6 公告说明.....6

7 修订历史记录.....21

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
AES_ERR_01	✓
COMP_ERR_06	✓
CPU_ERR_02	✓
CPU_ERR_03	✓
FCC_ERR_01	✓
FLASH_ERR_03	✓
FLASH_ERR_04	✓
FLASH_ERR_05	✓
FLASH_ERR_08	✓
KEYSTORE_ERR_01	✓
PMCU_ERR_14	✓
RST_ERR_01	✓
RST_ERR_02	✓
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	✓
TIMER_ERR_04	✓
TIMER_ERR_06	✓
TIMER_ERR_07	✓
UNICOMMI2CC_ERR_01	✓
UNICOMMI2CC_ERR_02	✓
UNICOMMI2CC_ERR_03	✓
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_09	✓

勘误编号	修订版 A
UNICOMMUART_ERR_10	✓

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。

该器件没有关于此类别的任何勘误。

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。

该器件没有关于此类别的任何勘误。

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，图 5-1 定义了器件修订版到版本 ID 的映射。

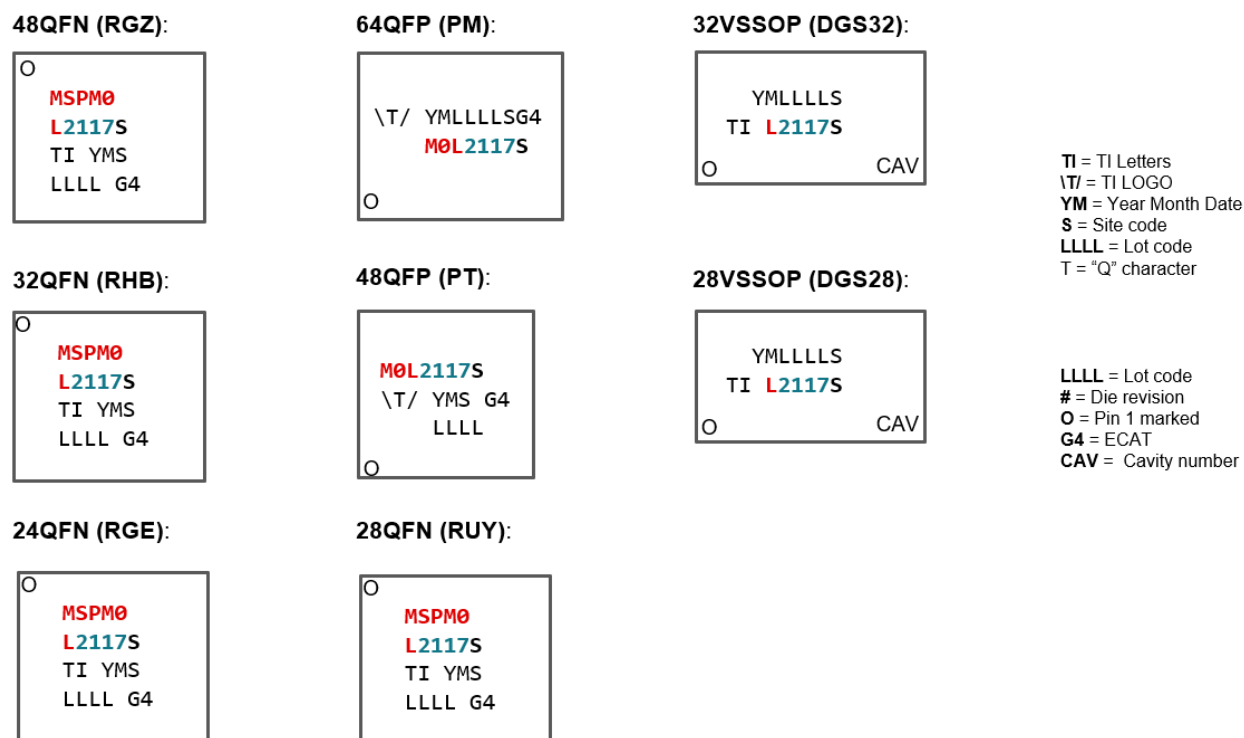


图 5-1. 封装符号

表 5-1. 芯片修订版本

修订版字母	产品开发状态	版本 (在器件的出厂常量存储器中)
A	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

RTM (上市) 之后发生产品硬件修订时，DEVICEID.VERSION 字段会递增。

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。

解决方法

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。

COMP_ERR_06 *COMP 模块*

类别

功能

功能

如果在输出就绪之前禁用了 SYSOSC，则 COMP 输出就绪信号和中断生成将延迟

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 将延迟 2 个 LFCLK 周期（约 60us）。

解决方法

为避免延迟 2 个 LFCLK 周期，在进入会禁用 SYSOSC 的 LPM 或将时钟切换至 LFCLK 之前，应等待 OUTRDYIFG 位设置完成。

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)，或根据需要时钟更改为 LFCLK

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。

解决方法

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

CPU_ERR_02 (续) CPU 模块

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。

解决方法

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

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

FLASH_ERR_03

(续)

FLASH 模块**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

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。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。

解决方法

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

FLASH_ERR_08

(续)

FLASH 模块

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

解决方法

否

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。

解决方法

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。

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

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

RST_ERR_01 (续) RST 模块

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。

解决方法

保持 NRST 脉冲宽度高于 608us 即可以避免此问题。

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

发生上电事件时，内部低频率振荡器 (LFOSC) 将在 NRST 首次置为无效 (从低电平转换为高电平) 后启用。LFOSCGOOD 信号会在此 NRST 置为无效后的 250us (最小值) 至 1.5ms (最大值) 窗口内转换为高电平。如果后续出现 NRST 低电平脉冲，并且该脉冲置为无效的时间与 LFOSCGOOD 上升沿在 200ns 窗口内重合，则器件可能无法识别 NRST 置为无效事件。因此，器件会保持复位状态，且无法退出启动序列的引导阶段。如果外部电路是由 NRST 逻辑控制，则可能会发生这种情况。

解决方法

确保外部复位控制器在初始 NRST 置为无效后，可将 NRST 线路保持高电平 (置为无效) 至少 1.5ms。

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。

解决方法

否

SYSCTL_ERR_02 SYSCTL 模块

类别

功能

功能

SYSSTATUS.FLASHSEC 在 BOOTRST 之后为非零

SYSCCTL_ERR_02

(续)

SYSCCTL 模块

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。

解决方法

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

说明

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

否

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。

解决方法

否

SYSCCTL_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_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
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
权变措施	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

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，ulpclock 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。

解决方法

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。

SYSOSC_ERR_04

(续)

SYSOSC 模块**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。

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。

解决方法

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。

解决方法

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。

UNICOMMI2CC_ERR_01 ***UNICOMMI2CC 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 BUSRTRUN/FRAME_START 位来启动 I2C 控制器传输后，大约需要 2 个 I2C 功能时钟周期才能将 BUSY 状态置为有效。如果在设置 BUSRTRUN/FRAME_START 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。

解决方法

在轮询 BUSY 状态之前添加软件延迟。软件延迟 = $3 \times I2C \text{ 功能时钟} = 3 \times \text{clock_divider} \times (\text{CPU_CLK}/\text{所选时钟源频率})$ ，例如 clock_divider 为 8、时钟源为 4MHz (MFCLK)、CPU_CLK 为 32MHz：软件延迟 = $3 \times 8 \times (32 \text{ MHz} / 4 \text{ MHz}) = 192$ 个 CPU 周期

UNICOMMI2CC_E RR_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。

权变措施

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_E RR_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。

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。

UNICOMMSPI_ER R_02

UNICOMM SPI 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。

UNICOMMSPI_ER R_03

UNICOMM SPI 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

UNICOMM SPI 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。

解决方法

否

UNICOMMUART_E**RR_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_E**RR_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。

解决方法

否

UNICOMMUART_E**RR_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。

解决方法

否

UNICOMMUART_E**RR_05** *UNICOMMUART 模块*

类别

功能

UNICOMMUART_E

RR_05 (续)

UNICOMMUART 模块

功能

当线路空闲时，干扰会导致 LTOUT/RTOUT 周期计算问题

说明

在正常 UART 模式下，空闲线路条件下的干扰会通过使 LTOUT 事件 1-2 波特时钟移位来破坏 LTOUT 时序。

解决方法

否

UNICOMMUART_E

RR_06

UNICOMMUART 模块

类别

功能

功能

停止位长度不一致，导致 RTOUT/LTOUT 周期计算问题

说明

在接收器侧，功能状态机会使得状态在停止位的中间从停止位变为空闲，这会导致 RTOUT 计数器在实际启动之前开始计数。这会造成 RTOUT 周期不正确，RTOUT 中断的到来早于预期。

解决方法

向 RTOUT 计数器添加具有半个停止位周期的补偿

UNICOMMUART_E

RR_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

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。

解决方法

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_E

RR_08

UNICOMMUART 模块

类别

功能

功能

LTOUT 中断将在每次超时到期后持续发生

说明

一个 LTOUT 中断到来后，计数器会重新启动并继续计数，并提供另一个 LTOUT 中断，即使两者之间没有任何 Rx 帧也是如此。这将导致在每个超时到期间隔出现连续的 LTOUT 中

UNICOMMUART_E**RR_08 (续)****UNICOMMUART 模块**

断。该行为与 RTOUT 不同，因为 RTOUT 中断仅发生一次，直到发生新的 Rx 事务。产生该限制的原因是，RTOUT 和 LTOUT 事件使用了同一个计数器。

解决方法

UNICOMMUART_E**RR_09****UNICOMMUART 模块**

类别

功能

功能

当波特率低于 57MHz UARTCLK 时，ISO-7816 智能卡模式无法支持 9600 波特率

说明

为了在 ISO-7816 智能卡模式下实现 9600 波特率并考虑到以下限制，UARTCLK 频率需要超过 57MHz。1.ISO-7816 标准规定 1 位需要 372 个时钟周期。2.在 MSPM0 ISO-7816 模式下，UART 外设中的过采样率 (OVS) 固定为 16 倍。以下是 UARTCLK 最小值的计算结果：所需的 UARTCLK = 9600 * 372 * 16 = 57.139MHz

解决方法

否

UNICOMMUART_E**RR_10****UNICOMMUART 模块**

类别

功能

功能

仅当 CLKDIV 设置为 /1 时，才能访问 LIN 寄存器

说明

在 UNICOMMUART (UART+LIN) 型号中，LIN 寄存器配置仅在 CLKDIV 设置为 1 时保持有效。除 1 以外的任何 CLKDIV 值都会导致这些设置被丢弃。

解决方法

否

UNICOMMUART_E**RR_11****UNICOMMUART 模块**

类别

功能

功能

当 UNICOMMUART 被禁用且数据在 txfifo 中可用时，STAT.BUSY 位保持高电平

说明

当 UNICOMMUART 电源被禁用且数据在 txfifo 中可用时，STAT.BUSY 位保持高电平。

解决方法

复位 UNICOMMUART 以初始化所有 UNICOMMUART 寄存器。

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 1, 2025 to August 18, 2026 (from Revision A (Nov 2025) to Revision B (Aug 2026))

	Page
• 已更新 COMP_ERR_06 类别.....	6
• 已更新 COMP_ERR_06 模块.....	6
• 已更新 COMP_ERR_06 功能.....	6
• 已更新 COMP_ERR_06 说明.....	6
• 已更新 COMP_ERR_06 权变措施.....	6
• 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 PMCU_ERR_14 类别.....	9
• 已更新 PMCU_ERR_14 模块.....	9
• 已更新 PMCU_ERR_14 功能.....	9
• 已更新 PMCU_ERR_14 说明.....	9
• 已更新 PMCU_ERR_14 权变措施.....	9
• 已更新 RST_ERR_02 类别.....	10
• 已更新 RST_ERR_02 模块.....	10
• 已更新 RST_ERR_02 功能.....	10
• 已更新 RST_ERR_02 说明.....	10
• 已更新 RST_ERR_02 权变措施.....	10
• 已更新 SYSCTL_ERR_05 模块.....	11
• 已更新 SYSCTL_ERR_05 功能.....	11
• 已更新 SYSCTL_ERR_05 说明.....	11
• 已更新 SYSCTL_ERR_05 权变措施.....	11
• 已更新 SYSCTL_ERR_06 类别.....	12
• 已更新 SYSCTL_ERR_06 模块.....	12
• 已更新 SYSCTL_ERR_06 功能.....	12
• 已更新 SYSCTL_ERR_06 说明.....	12
• 已更新 SYSCTL_ERR_06 权变措施.....	12
• 已更新 SYSCTL_ERR_11 类别.....	12
• 已更新 SYSCTL_ERR_11 模块.....	12
• 已更新 SYSCTL_ERR_11 功能.....	12
• 已更新 SYSCTL_ERR_11 权变措施.....	12
• 已更新 SYSCTL_ERR_11 说明.....	12
• 已更新 SYSCTL_ERR_12 类别.....	12
• 已更新 SYSCTL_ERR_12 模块.....	12
• 已更新 SYSCTL_ERR_12 功能.....	12
• 已更新 SYSCTL_ERR_12 说明.....	12
• 已更新 SYSCTL_ERR_12 权变措施.....	12
• 已更新 SYSOSC_ERR_04 模块.....	13
• 已更新 SYSOSC_ERR_04 权变措施.....	13
• 已更新 SYSOSC_ERR_04 功能.....	13
• 已更新 SYSOSC_ERR_04 说明.....	13
• 已更新 SYSOSC_ERR_05 类别.....	14
• 已更新 SYSOSC_ERR_05 模块.....	14

• 已更新 SYSOSC_ERR_05 功能.....	14
• 已更新 SYSOSC_ERR_05 权变措施.....	14
• 已更新 SYSOSC_ERR_05 说明.....	14
• 已更新 UNICOMMI2CC_ERR_02 类别.....	16
• 已更新 UNICOMMI2CC_ERR_02 模块.....	16
• 已更新 UNICOMMI2CC_ERR_02 功能.....	16
• 已更新 UNICOMMI2CC_ERR_02 说明.....	16
• 已更新 UNICOMMI2CC_ERR_02 权变措施.....	16
• 已更新 UNICOMMI2CC_ERR_03 类别.....	16
• 已更新 UNICOMMI2CC_ERR_03 模块.....	16
• 已更新 UNICOMMI2CC_ERR_03 功能.....	16
• 已更新 UNICOMMI2CC_ERR_03 说明.....	16
• 已更新 UNICOMMI2CC_ERR_03 权变措施.....	16
• 已更新 UNICOMMSPI_ERR_02 类别.....	16
• 已更新 UNICOMMSPI_ERR_02 模块.....	16
• 已更新 UNICOMMSPI_ERR_02 功能.....	16
• 已更新 UNICOMMSPI_ERR_02 说明.....	16
• 已更新 UNICOMMSPI_ERR_02 权变措施.....	16
• 已更新 UNICOMMSPI_ERR_03 类别.....	17
• 已更新 UNICOMMSPI_ERR_03 模块.....	17
• 已更新 UNICOMMSPI_ERR_03 功能.....	17
• 已更新 UNICOMMSPI_ERR_03 说明.....	17
• 已更新 UNICOMMSPI_ERR_03 权变措施.....	17
• 已更新 UNICOMMSPI_ERR_04 类别.....	17
• 已更新 UNICOMMSPI_ERR_04 模块.....	17
• 已更新 UNICOMMSPI_ERR_04 功能.....	17
• 已更新 UNICOMMSPI_ERR_04 说明.....	17
• 已更新 UNICOMMSPI_ERR_04 权变措施.....	17

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