

Errata

MSPM0G321x/MSPM0G320x 混合信号微控制器



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.....	2
3 仅调试公告.....	2
4 编译器修复公告.....	2
5 器件命名规则.....	2
6 公告说明.....	4
7 修订历史记录.....	16

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 AES 模块	✓
CPU_ERR_02 CPU 模块	✓
CPU_ERR_03 CPU 模块	✓
FLASH_ERR_03 FLASH 模块	✓
FLASH_ERR_04 FLASH 模块	✓
FLASH_ERR_05 FLASH 模块	✓
FLASH_ERR_08 FLASH 模块	✓
GPIO_ERR_05 GPIO 模块	✓
GPIO_ERR_06 GPIO 模块	✓
KEYSTORE_ERR_01 密钥库模块	✓
MATHACL_ERR_01 MATHACL 模块	✓
PMCU_ERR_13 PMCU 模块	✓
RST_ERR_01 RST 模块	✓
SYSCTL_ERR_01 SYSCTL 模块	✓
SYSCTL_ERR_02 SYSCTL 模块	✓
SYSCTL_ERR_03 SYSCTL 模块	✓
SYSCTL_ERR_04 SYSCTL 模块	✓
SYSCTL_ERR_05 LFCLK 模块	
SYSCTL_ERR_06 CLK_OUT 模块	
SYSOSC_ERR_01 SYSOSC 模块	✓
SYSOSC_ERR_02 SYSOSC 模块	✓

勘误编号	修订版 A
SYSOSC_ERR_04 SYSOSC 模块	✓
SYSPLL_ERR_01 SYSPLL 模块	✓
TIMER_ERR_04 TIMER 模块	✓
TIMER_ERR_06 TIMG 模块	✓
TIMER_ERR_07 初始重复计数器的周期比下一个重复模块少 1 个	✓
UNICOMMI2CC_ERR_01 UNICOMMI2CC 模块	
UNICOMMUART_ERR_06 UNICOMMUART 模块	✓
UNICOMMUART_ERR_07 UNICOMMUART 模块	✓
UNICOMMUART_ERR_09 UNICOMMUART 模块	
UNICOMMUART_ERR_10 UNICOMMUART 模块	✓
VREF_ERR_05 VREF 模块	✓

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。

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。

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。
CPU_ERR_02	CPU 模块
类别	功能
功能	禁用 CPUSS 预取功能的限制
说明	如果存在待处理的闪存访问，CPU 预取禁用将不会生效。
权变措施	禁用预取，然后对 SYSCTL 中的关断存储器 (SHUTDNSTORE) 发出存储器访问权限，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。
CPU_ERR_03	CPU 模块
类别	功能
功能	在切换到 SLEEP 模式时，预取器可能会导致数据完整性问题
说明	切换到 SLEEP0 时，预取器可能会错误地获取不正确的数据（全为 0）。退出睡眠模式时，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。
权变措施	进入 SLEEP 之前禁用预取器。
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。

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

在 SYSCTL->DEDERRADDR 中报告了错误的地址

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 可以列出为 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_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 2.0x40BFFFFFF - 0x41C00000 3.0x41C007FF - 0x41C40000

权变措施

否

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。

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_ERR_06

(续)

GPIO 模块

```

{
gpio->DOU TTGL31_0 = ~(gpio->DOU T31_0) & pins;
}

DL_GPIO_clearPins(GPIO_Regs* gpio, uint32_t pins)
{
gpio->DOU TTGL31_0 = gpio->DOU T31_0 & pins;
}

DL_GPIO_writePins(GPIO_Regs* gpio, uint32_t pins)
{
gpio->DOU TTGL31_0 = ~(gpio->DOU T31_0) & pins;
gpio->DOU TTGL31_0 = gpio->DOU T31_0 & (~pins);
}

DL_GPIO_writePinsVal(GPIO_Regs* gpio, uint32_t pinsMask, uint32_t pinsVal)
{
uint32_t doutVal = gpio->DOU T31_0;
doutVal &= ~pinsMask;
doutVal |= (pinsVal & pinsMask);
gpio->DOU TTGL31_0 = ~(gpio->DOU T31_0) & doutVal;
gpio->DOU TTGL31_0 = gpio->DOU T31_0 & (~doutVal);
}

```

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。

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。

MATHACL_ERR_01

MATHACL 模块

类别

功能

功能

MATHACL 状态错误位不会被清除

MATHACL_ERR_0

1 (续)

MATHACL 模块

说明

如果 mathacl 生成状态错误 (例如, 除以 0), 则状态寄存器永远不会被清除。

权变措施

复位外设以清除状态位。

PMCU_ERR_13**PMCU 模块**

类别

功能

功能

MCU 可能会在从 STOP2 和 STANDBY0 唤醒时卡住

说明

如果在器件转换到 STOP2 或 STANDBY 模式时预取访问处于挂起状态, 则当器件唤醒时, 挂起的预取会阻止器件恢复正常执行。如果 WFI 指令未按字对齐, 且闪存等待状态为 2, 则会发生勘误表所述问题。在这种情况下, 既不会处理 DMA 传输, 也不会处理挂起的中断。

权变措施

用户应禁用预取并发出关断存储内存读取, 从而防止发出新的预取并允许挂起的预取完成。

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。

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。

SYSCCTL_ERR_01

(续)

SYSCCTL 模块

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

否

SYSCCTL_ERR_05

LFCLK 模块

类别

功能

SYSCTL_ERR_05

(续)

LFCLK 模块**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后被禁用时，CLK_OUT 引脚上可能会出现干扰

权变措施

无权变措施

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

当启用 MFCLK，且 SYSOSC 使用频率校正环路 (FCL) 模式和 STOP1 低功耗工作模式时，若 SYSOSC 从 4MHz 切换回 32MHz (无论是从 STOP1 退出到运行模式，或是因异步快速时钟请求强制 SYSOSC 切换回 32MHz)，则 MFCLK 可能会漂移 2 个周期。

权变措施

Workaround1：使用 STOP0 模式而非 STOP1 模式。使用 STOP0 模式时不会出现 MFCLK 漂移。Workaround2：使用 STOP1 时，请勿在 FCL 模式下使用 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

SYSOSC_ERR_02

(续)

SYSOSC 模块

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

使用 SYSPLL 时，SYSOSC 精度在 FCL ON 模式下会降低

说明

当使用内部振荡器 (SYSOSC) 的 FCLON 模式时，在使用 SYSPLL 且 FCL ON 的情况下，精度可能会降低多达 +/-3%。精度下降是由于 4MHz SYSOSC 采样时钟与系统噪声之间的同步所致。

权变措施

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。

请勿将 SYSPLL 频率设为 16、32、48、64、80MHz 等

对于 78MHz：

请将 SYSPLLCFG1.PDIV 设为 0x3，SYSPLLCFG1.QDIV 设为 38。

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。

权变措施

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

SYSPLL_ERR_01

(续)

SYSPLL 模块

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.FCCLVLRIG = 0;
- SYSCTL.GENCLKCFG.FCCTRIGCNT = 0;
- 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 FCC 完成状态监控期间实施基于软件的超时，以防止无限等待：

```
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 = $80\text{MHz}/40\text{MHz} = 2.0000$
- 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

SYSPLL_ERR_01

(续)

SYSPLL 模块

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 (超时保护)
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 (超时保护)
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。

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。对计时器使能位的硬件更新将需要单个功能时钟周期, 如果使用 32.768kHz 和分频器 3, 则需要约 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 <code>TIMx.RCLD = 0x3</code> ，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 <code>RCLD = 0x5</code> ，然后在 REPC 中断的计时器 ISR 中设置 <code>RCLD = 0x4</code> 。现在，所有计时器重复将具有相同数量的零/加载事件。
UNICOMMI2CC_E RR_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 <code>BUSTRUN/FRAME_START</code> 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 <code>BUSTRUN/FRAME_START</code> 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 <code>CLKDIV</code> 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 = $3 \times \text{I2C 功能时钟} = 3 \times \text{clock_divider} \times (\text{CPU_CLK}/\text{所选时钟源频率})$ ，例如 <code>clock_divider</code> 为 2、时钟源为 4MHz (MFCLK)、 <code>CPU_CLK</code> 为 80MHz：软件延迟 = $3 \times 2 \times (80 \text{ MHz} / 4 \text{ MHz}) = 120$ 个 CPU 周期
UNICOMMUART_E RR_06	UNICOMMUART 模块
类别	功能
功能	由于停止位处理而导致 <code>RTOUT/LTOUT</code> 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 (<code>RTOUT</code>) 和线路超时 (<code>LTOUT</code>) 计数器在停止位的中间而非末端开始计数。这会导致 <code>RTOUT/LTOUT</code> 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。

UNICOMMUART_E

RR_06 (续) *UNICOMMUART 模块*

权变措施

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。

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_E

RR_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_E

RR_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。

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

VREF_ERR_05
VREF 模块
类别

功能

功能

当 COMP 配置为采样 (超低功耗) 模式时，VREF0 的 VREF READY0 状态变为低电平

说明

当比较器配置为采样 (超低功耗) 模式时，VREF0 的 VREF READY0 状态保持低电平。
当 REFMODE 位设置为 1 时，比较器配置为采样模式。

权变措施

将比较器配置为采样模式后，请勿轮询 VREF READY0 状态。在轮询 VREF0 的 READY0 状态之前，将比较器配置为静态 (快速) 模式，或在启用比较器之前轮询 READY0 状态。

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

日期	修订版本	注释
2026 年 2 月	*	初始发行版

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