Errata

MSPM0G3x0x, MSPM0G1x0x Microcontrollers

ABSTRACT

This document describes the known exceptions to the functional specifications (advisories).

Table of Contents

1 Functional Advisories........ 1
2 Preprogrammed Software Advisories........ 1
3 Debug Only Advisories........ 2
4 Fixed by Compiler Advisories........ 2
5 Device Nomenclature........ 2
6 Advisory Descriptions........3
7 Revision History........ 18

1 Functional Advisories

Advisories that affect the device's operation, function, or parametrics.

✓ The check mark indicates that the issue is present in the specified revision.

Errata Number	Rev B
ADC_ERR_01	✓
ADC_ERR_02	✓
BSL_ERR_01	✓
COMP_ERR_02	✓
COMP_ERR_03	✓
CPU_ERR_01	✓
GPIO_ERR_01	✓
I2C_ERR_01	✓
I2C_ERR_02	✓
I2C_ERR_03	✓
PWREN_ERR_01	✓
RTC_ERR_01	✓
SPI_ERR_01	✓
SPI_ERR_02	✓
SYSOSC_ERR_01	✓
TIMER_ERR_01	✓
VREF_ERR_01	✓
WWDT_ERR_01	✓
WWDT_ERR_02	✓

2 Preprogrammed Software Advisories

Advisories that affect factory-programmed software.

✓ The check mark indicates that the issue is present in the specified revision.

3 Debug Only Advisories

Advisories that affect only debug operation.

✓ The check mark indicates that the issue is present in the specified revision.

4 Fixed by Compiler Advisories

Advisories that are resolved by compiler workaround. Refer to each advisory for the IDE and compiler versions with a workaround.

✓ The check mark indicates that the issue is present in the specified revision.

5 Device Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all MSP MCU devices. Each MSP MCU commercial family member has one of two prefixes: MSP or XMS. These prefixes represent evolutionary stages of product development from engineering prototypes (XMS) through fully qualified production devices (MSP).

XMS – Experimental device that is not necessarily representative of the final device's electrical specifications

MSP – Fully qualified production device

Support tool naming prefixes:

X: Development-support product that has not yet completed Texas Instruments internal qualification testing.

null: Fully-qualified development-support product.

XMS devices and X development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

MSP devices have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (XMS) have a greater failure rate than the standard production devices. TI recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the temperature range, package type, and distribution format.

6 Advisory Descriptions

ADC_ERR_01 *ADC is unable to trigger fast clock in STANDBY1 mode*

Revisions Affected

Rev B

Details

When the device is operating in STANDBY1 mode, the ADC module may not correctly assert an asynchronous fast clock request when it is triggered through the event system (for example, when an event publisher such as a timer generates an event to the ADC through the ADC event subscriber port).

Workaround

Use STANDBY0 or higher power modes when triggering ADC conversions via the event fabric.

ADC_ERR_02 *Increased current in low power mode when ADC is in Repeat mode*

Revisions Affected

Rev B

Details

When the ADC is configured to Repeat Sequence or Repeat Single mode with EVENT trigger in low power mode (LPM), the ADC module will consume additional current until disabled.

Workaround

Configure ADC to be in Single or Sequence mode (no Repeat), and use an interrupt after conversion to re-enable ADC for next trigger

BSL_ERR_01 *Invoking the BSL through application software will fail under certain conditions*

Revisions Affected

B

Details

IF invoking the BSL within an application (BSL Software Invoke),

AND an application uses the following SRAM memory locations:

- 0x20000EE0 - 0x20000FFF
- 0x20007FF4 - 0x20007FFF

THEN the BSL will fail to start after the reset and the device will go back to the application.

This errata does not apply to BSL invocation through hardware invoke methods.

Workaround

For applications needing a software invocation of the BSL, reserve the mentioned SRAM ranges for BSL use within the linker file. The MSPM0 bsl_software_invoke example within the MSPM0 SDK v 1.20.00.xx or higher contains a linker file with this fix.

COMP_ERR_02 *COMP output toggles when DACCODEx is used for hysteresis*

Revisions Affected

B

Details

IF using the 8-bit DAC within the COMP module as an input to the COMP,

AND DAC output switches between values placed in DACCODEx,

THEN the COMP output toggles as if immediately crossing the new reference point.

This can happen regardless of the setting of the COMPx.CTL2.DACCTL bit.

This is most commonly seen in applications that utilize the two DACCODEx codes for custom or asymmetrical hysteresis for the COMP module.

Workaround

Utilize the established hysteresis values provided in COMPx.XTL1.HYST register bits.

COMP_ERR_03 *COMP hysteresis features are non-functional when using input exchange feature*

Revisions Affected

B

Details

When using hysteresis features of the COMP module, and exchanging the inputs of the COMP (COMPx.CTL1.EXCH = 1), the COMP module becomes unstable.

Workaround

Do not apply internal hysteresis methods when utilizing COMP module in input exchange feature.

CPU_ERR_01 *CPU cache content can get corrupted*

Revisions Affected

B

Details

Cache corruption can occur when switching between accessing Main flash memory, and other memory regions such as NONMAIN or Calibration data areas.

Workaround

Use the following procedure to access areas outside main memory safely:

1. Disable the cache by setting CTL.ICACHE to "0".
2. Perform needed access to memory area.
3. Re-enable cache by setting CTL.ICACHE to "1".

GPIO_ERR_01 *GPIO wakeup edges may be lost in STANDBY mode*

Revisions Affected

B

Details

After waking up once through a single GPIO edge, subsequent GPIO wakeup edges can be missed in STANDBY modes.

- Case 1: STANDBY0 wakeup - IF the MCU is set into STANDBY0 mode before one cycle of LFCLK AND you set the IO back to the “non-wake” state for <1 LFCLK cycle AND THEN assert it again, the next wakeup edge will not be detected.
- Case 2: STANDBY1 wakeup - IF a GPIO edge is used to wakeup AND the GPIO pulse is still active when the device returns to STANDBY1 THEN the device will not detect any subsequent wakeup edges.

Workaround

- Case 1:
  - Set GPIO wakeup edge to both falling and rising edges
  - OR Ensure GPIO wakeup pulse is longer than one LFCLK cycle
- Case 2:
  - Set GPIO wakeup edge to both falling and rising edges
  - OR Ensure GPIO wakeup pulse is not active before entering STANDBY1

I2C_ERR_01 *I2C module may hold the SDA line low when an SMBUS quick command is issued*

Revisions Affected

B

Details

IF the I2C module is in target mode AND configured for SMBUS ; AND IF the bus controller issues an SMBUS quick command addressed to the device with the R/W bit set to read, THEN the I2C module may hold the SDA line low preventing the STOP condition from completing successfully

Workaround

Load data into the I2C module transmit FIFO with the MSB set to 1 before the address ACK is completed to prevent the I2C module from driving the SDA line low.

.

I2C_ERR_02 *I2C quick command read mode only works with specific conditions*

Revisions Affected

B

Details

I2C quick command in read mode works only if TXFIFO has data with MSB set to 1 and clock stretching is disabled.

Workaround

Provide a dummy data in the TXFIFO with MSB set to 1 AND disable the clock stretching (CLKSTRETCH = 0).

I2C_ERR_03 *I2C peripheral mode cannot wake up device when sourced from MFCLK*

Revisions Affected

B

Details

IF I2C module is configured in peripheral mode AND I2C is clocked from MFCLK (Middle Frequency Clock) AND device is placed in STOP2 or STANDBY0/1 power modes, THEN I2C fails to wakeup the device when receiving data.

Workaround

Set I2C to be clocked by BUSCLK instead of MFCLK, if needing low power wakeup upon receiving data in I2C peripheral mode.

PWREN_ERR_01 *Peripheral registers are still accessible after disabling PWREN register*

Revisions Affected

B

Details

When disabling the power of a peripheral by setting the PWREN register to 0, the peripherals registers may appear to retain data values if read. Reading or writing to the registers when PWREN is 0 has no affect as the peripheral has no effect.

The following peripherals are affected: comparator (COMP), operational amplifier (OPA), TimerA, TimerG, general-purpose input/output (GPIO), and windowed watchdog timer (WWDT), AES and TRNG.

Workaround

When the PWREN register of the peripheral is set to 0, the values of the associated registers should be disregarded or considered invalid.

RTC_ERR_01 *Some RTC Interrupts are not available in STANDBY1*

Revisions Affected

B

Details

When in STANDBY1, the RTCRDY and RTC_PRESCALER1 interrupts cannot wakeup the device.

Workaround

When waking up the device from STANDBY1 with the RTC, use other available interrupts such as RTC_ALARM and RTC_PRESCALER0.

SPI_ERR_01 *SPI Parity bit is not functional*

Revisions Affected

B

Details

SPI hardware parity modes are not functional.

Workaround

Do not enable hardware parity via the PTEN or PREN bit in the CTL1 register of the SPI module. Parity computation and checking may be implemented with application software.

SPI_ERR_02 *SPI Clock and data bytes after wake-up from low power mode (LPM) are missed*

Revisions Affected

B

Details

After device wake-up from a low power state, the SPI module may not properly propagate the first few clock cycles and data bits of the first byte sent out.

Workaround

To ensure SPI data integrity after a wakeup, use the following sequence when entering and exiting LPMs:

- Disable SPI module
- Wait for Interrupt(WFI)- enter LPM
- Wake up from LPM (any source).
- Enable the SPI module.

SYSOSC_ERR_01 *MFCLK may drift when using SYSOSC FCL mode and STOP1*

Revisions Affected

B

Details

IF MFCLK is enabled AND SYSOSC is using the frequency correction loop (FCL) mode AND the STOP1 low power operating mode is used, THEN the MFCLK may drift by two cycles when SYSOSC shifts from 4MHz back to 32MHz (either upon exit from STOP1 to RUN mode or upon an asynchronous fast clock request that forces SYSOSC to 32MHz).

Workaround

Use STOP0 mode instead of STOP1 mode. There is no MFCLK drift when STOP0 mode is used.

OR

Do not use SYSOSC in the FCL mode (leave FCL disabled) when using STOP1.

TIMER_ERR_01 *Capture mode captures incorrect value when using hardware event to start timer*

Revisions Affected

B

Details

When using any timer instance in capture mode, starting the timer using a zero (ZCOND) or load (LCOND) condition causes the timer to capture the zero or load value instead of the captured value in the respective TIMx.CC register. This affects periodic use cases such as period and pulse width capture.

Workaround

Use the below software flow to calculate the period or pulse width. See the timx_timer_mode_capture_duty_and_period in the MSPM0-SDK for an example of the workaround.

- Disable ZCOND or LCOND by setting to 0h.
- When a capture occurs, the capture value is correctly captured in TIMx.CC
- Restart the timer by setting TIMx.CTR to the reload value (load or 0)

VREF_ERR_01 *VREF READY bit does not get cleared after disabling VREF*

Revisions Affected

B

Details

The first time the VREF module is enabled after a SYSRST, the VREF READY bit can be used for its intended functionality. If the VREF module is disabled in application, the VREF READY bit does not get cleared. As a result of this errata, subsequent enabling of the VREF module cannot use the VREF READY bit to indicate the VREF module is stable.

Workaround

When re-enabling the VREF module in application, utilize a TIMER module to wait the maximum VREF startup time, before utilizing the VREF module. Please see the device data sheet for VREF startup time.

WWDT_ERR_01 *Watchdog timer 1 (WWDT1) event always does a SYSRST*

Revisions Affected

B

Details

WWDT1 event always does a SYSRST irrespective of the SYSTEMCFG.WWDTLP1RSTDIS bit setting, thus an WWDT1 event cannot trigger an NMI.

Workaround

Use WWDT0 for NMI purposes.

WWDT_ERR_02 *Window Watchdog Timer 1 (WWDT1) does not create a reset cause*

Revisions Affected

B

Details

After a reset caused by WWD1, the SYSCTL.RSTCAUSE register does not accurately portray that WWDT1 caused the reset.

Workaround

None.

7 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
October 2023	A	Silicon Revision B

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265