Errata

MSPM0L222x, MSPM0L122x Microcontrollers

ABSTRACT

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

Table of Contents

1 Functional Advisories........ 1
2 Preprogrammed Software Advisories........ 3
3 Debug Only Advisories........ 3
4 Fixed by Compiler Advisories........ 3
5 Device Nomenclature........ 3
5.1 Device Symbolization and Revision Identification........ 4
6 Advisory Descriptions........ 5
7 Revision History........ 14

Trademarks

All trademarks are the property of their respective owners.

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 A
ADC_ERR_03	✓
ADC_ERR_05	✓
ADC_ERR_06	✓
COMP_ERR_03	✓
COMP_ERR_04	✓
GPIO_ERR_03	✓
GPIO_ERR_04	✓
I2C_ERR_03	✓
I2C_ERR_05	✓
I2C_ERR_06	✓
LFXT_ERR_01	✓
LFXT_ERR_02	✓
PMCU_ERR_07	✓
PMCU_ERR_08	✓
PMCU_ERR_10	✓
PMCU_ERR_12	✓
RTC_A_ERR_02	✓
SPI_ERR_03	✓
SPI_ERR_04	✓
SPI_ERR_05	✓

Errata Number	Rev A
SRAM_ERR_01	✓
SYSOSC_ERR_01	✓
TAMPERIO_ERR_01	✓
TIMER_ERR_01	✓
TIMER_ERR_04	✓
UART_ERR_01	✓
UART_ERR_02	✓
UART_ERR_03	✓
VREF_ERR_03	✓

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.

Errata Number	Rev A

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.

5.1 Device Symbolization and Revision Identification

The package diagrams below indicate the package symbolization scheme, and Table 5-1 defines the device revision to version ID mapping.

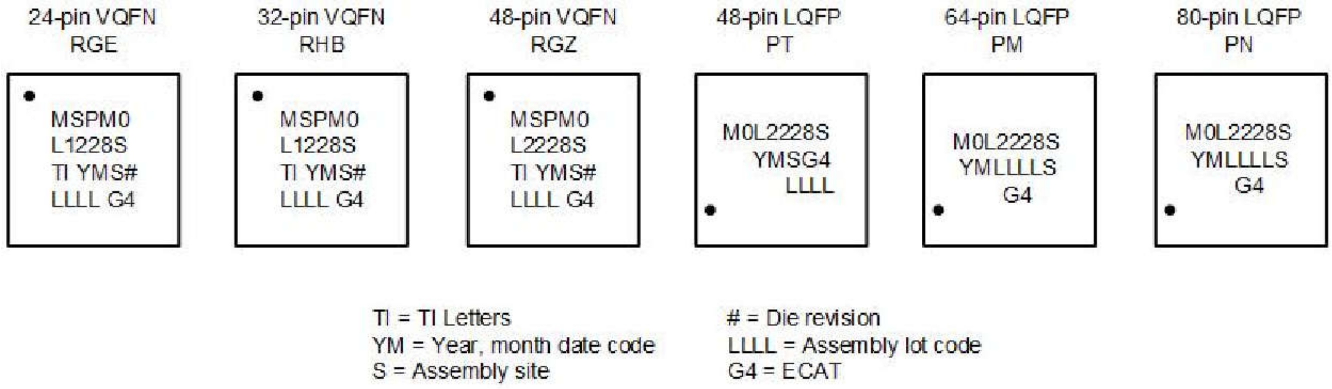

Figure 5-1. Package Symbolization

Table 5-1. Die Revisions

Revision Letter (package marking)	Version (in the device factory constants memory)
A	1

The revision letter indicates the product hardware revision. Advisories in this document are marked as applicable or not applicable for a given device based on the revision letter. This letter maps to an integer stored in the memory of the device, which can be used to look up the revision using application software or a connected debug probe.

6 Advisory Descriptions

ADC_ERR_03 *ADC Module*

Category

Functional

Function

ADC SNR and DNL performance is worse when using VSSA for ADC ground reference.

Description

When the ADC is using VSS as the ground reference, noise sources on the VSS ground will impact the ADC results, which results in worsened SNR and DNL performance.

Workaround

Workaround 1: Use VREF+ and VREF- pins for the ADC reference. With the VREF- pin, the ADC will have it's analog ground reference lessening the influence of external noise sources. Workaround 2: Reduce noise sources on VSS; common sources are CPU or Digital switching noise

ADC_ERR_05 *ADC Module*

Category

Functional

Function

HW Event generated before enabling IP, ADC Trigger will stay in queue

Description

When ADC is configured in HW event trigger mode and the trigger is generated before enabling the ADC, the ADC trigger will stay in queue. Once ADC is enabled, it will trigger sampling and conversion.

Workaround

After configuring ADC in HW trigger mode, enable ADC first before giving external trigger.

ADC_ERR_06 *ADC Module*

Category

Functional

Function

ADC Output code jumps degrading DNL/INL specification

Description

The ADC may have errors at a rate as high as 1 in 2M conversions in 12-bit mode. When a conversion error occurs, it will be a significant random jump in the digital output of the ADC without a corresponding change in the ADC input voltage. The magnitude of this jump is larger near major transitions in the bit values of the ADC result (more bits transitioning from 1->0, or 0->1), and largest around midscale (2048 or 0x800). Depending on the application needs the best workaround may vary, but the following workarounds in software are proposed. Selection of the best workaround is left to the judgment of the system designer.

Workaround

Workaround 1: Upon ADC result outside of application threshold (via ADC Window Comparator or

ADC_ERR_06 (continued) — *ADC Module*

software thresholding), trigger or wait for another ADC result before making critical system decisions

Workaround 2: During post-processing, discard ADC values which are sufficiently far from the median or expected value. The expected value should be based on the average of real samples taken in the system, and the threshold for rejection should be based on the magnitude of the measured system noise.

Workaround 3: Use ADC sample averaging to minimize the effect of the results of any single incorrect conversion.

COMP_ERR_03 — *COMP Module*

Category

Functional

Function

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

Description

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.

COMP_ERR_04 — *COMP Module*

Category

Functional

Function

Connecting TEMPSENSE as COMP input channel in standby makes COMP output continuously toggle

Description

Temperature sensor is OFF in standby mode. Connecting TEMPSENSE as COMP input channel in standby mode makes COMP output continuously toggle.

Workaround

Do not use TEMSENSE as COMP input in standby mode.

GPIO_ERR_03 — *GPIO and DEBUGSS Module*

Category

Functional

GPIO_ERR_03 (continued) — *GPIO and DEBUGSS Module*

Function

On a debugger read to GPIO EVENT0 IIDX, interrupt is cleared.

Description

EVENT0's IIDX of GPIO, on a debugger read is treated as a CPU read and interrupt is getting cleared.

Workaround

During the debug, the IIDX of event0 can be read by software reading RIS.

GPIO_ERR_04 — *GPIO Module*

Category

Functional

Function

Configuring global fastwake is not allowing PAD data to go to DIN register

Description

When configuring the fastwakeonly bit of CTL register and forcing data to PAD in run mode, data in PAD is not reflecting in DIN register. This is because the configuration of CTL register does not allow any data to go from PAD to DIN register.

Workaround

Avoid to use GPIO fastwakeonly function when expecting data in PAD going into DIN register.

I2C_ERR_03 — *I2C Module*

Category

Functional

Function

I2C peripheral mode cannot wake up device when sourced from MFCLK

Description

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.

I2C_ERR_05 — *I2C Module*

Category

Functional

Function

I2C SDA may get stuck to zero if we toggle ACTIVE bit during ongoing transaction

I2C_ERR_05 (continued) — *I2C Module*

Description

If ACTIVE bit is toggled during an ongoing transfer, its state machine will be reset. However, the SDA and SCL output which is driven by the master will not get reset. There is a situation where SDA is 0 and master has gone into IDLE state, here the master won't be able to move forward from the IDLE state or update the SDA value. Slave's BUSBUSY is set (toggling of the ACTIVE bit is leading to a start being detected on the line) and the BUSBUSY won't be cleared as the master will not be able to drive a STOP to clear it.

Workaround

Do not toggle the ACTIVE bit during an ongoing transaction.

I2C_ERR_06 — *I2C Module*

Category

Functional

Function

SMBus High timeout feature fails at I2C clock less than 24KHz onwards

Description

SMBus High timeout feature is failing at I2C clock rate less than 24KHz onwards (20KHz, 10KHz). From SMBUS Spec, the upper limit on SCL high time during active transaction is 50us. Total time taken from writing of START MMR bit to SCL low is 60us, which is >50us. It will trigger the timeout event and let I2C Master goes into IDLE without completing the transaction at the start of transfer itself. Below is detailed explanation. For SCL is configured as 20KHz, SCL low and high period is 30us and 20us respectively. First, START MMR bit write at the same time high timeout counter starts decrementing. Then, it takes one SCL low period (30us) from START MMR bit write to SDA goes low (start condition). Next, it takes another SCL low period (30us) from SDA goes low (start condition) to SCL goes low (data transfer starts) which should stop the high timeout counter at this point. As a total, it takes 60us from counter start to end. However, due to the upper limit(50us) of the high timeout counter, the timeout event will still be triggered although the I2C transaction is working fine without issue.

Workaround

Do not use SMBus High timeout feature when I2C clock less than 24KHz onwards.

LFXT_ERR_01 — *LFXT Module*

Category

Functional

Function

LFXT clock is not getting active until writing something into LFCLKCFG mmr

Description

After setting the STARTLFXT, the CLKSTATUS is updating as expected, but LFXT clock is not seen on LFCLK_OUT until a write into the LFCLKCFG is made. And if LFXT is configured as a source to LFCLK tree, VBAT supply goes down and comes up back again, in that case the LFCLK is blank until a write into LFCLKCFG is made.

Workaround

Manually configuring LFCLKCFG register (such as XT1DRIVE field) to make LFXT valid, every time LFXT starts from blanking state.

LFXT_ERR_02 *LFXT Module*

Category

Functional

Function

Incorrect LFCLKMUX status for LFXT after a VDD domain power-cycle

Description

LFXT (in LFSS) is configured by user. If there is a VDD power-cycle, then LFCLKMUX status is getting incorrectly reset to the default state (LFCLKMUX status showing LFOSC) after VDD boots-up. The device does continue to use LFXT as the LFCLK source.

Workaround

1.Ignore the LFCLKMUX bit status if there is a VDD power-cycle. 2.Re-configure the LFXT if there is a VDD power-cycle.

PMCU_ERR_07 *PMCU Module*

Category

Functional

Function

NRST<1sec pulse giving wrong rstcause in shutdown mode

Description

The rstcause value is wrong under the following condition. Though the expected rstcause is 0x05. (i) Device is configured for shutdown mode (ii) WFI() is called (iii) Give NRST<1sec pulse to bring device out from shutdown mode

Workaround

No workaround.

PMCU_ERR_08 *PMCU Module*

Category

Functional

Function

More wakeup time than expected when trigger is given to device while it is transitioning to LPM

Description

If wakeup signal is given to device when it is transitioning to low power mode, an additional wakeup time of approximately 3us will occur.

Workaround

No workaround.

PMCU_ERR_10 *PMCU Module*

Category

Functional

Function

VBOOST might have larger delay under certain operating conditions

PMCU_ERR_10 (continued) — *PMCU Module*

Description

VBOOST for analog MUX has large delay at VDD<1.8V, which delays settling time of other modules like HFXT, COMP, SYSOSC(FCL-external R),OPA and GPAMP.

Workaround

Keep VDD>=1.8V and use VBOOST in ONALWAYS mode using GENCLKCFG[23:22]=0x2.

PMCU_ERR_12 — *PMCU Module*

Category

Functional

Function

BOR cold boot spec violation

Description

The BOR cold boot spec could be violated, the max spec at is 1.65V.

Workaround

No Workaround. While cold booting the device operate the device above 1.65V.

RTC_A_ERR_02 — *RTC Module*

Category

Functional

Function

Contents of LFSS-RTC TS data registers not reset on LFSS SW POR reset (LFSSRST)

Description

On LFSS SW POR reset (LFSSRST), the RTC TS(Time Stamp) data registers do not reset to 0x0.

Workaround

Use TSCLR register to manually clear the value in RTC TS data registers.

SPI_ERR_03 — *SPI Module*

Category

Functional

Function

When configured as peripheral for a multi-peripheral application, received data will have a right shift

Description

In multi-peripheral scenario, SPI controller first communicates with peripheral0 and then communicates with peripheral1. After finishing communication with peripheral1, the controller again communicates with peripheral0. During the second communication with peripheral0, received data of peripheral0 will have a right shift in the first frame. The peripheral0 is getting first data as 0x3B when the controller sent data 0x76.

SPI_ERR_03
(continued)

SPI Module

Workaround

To support multi peripheral scenario CSCLR needs to be enabled at peripheral end to reset it's RX and TX the bit counters, when there is no active communication happening with that peripheral (CS of that peripheral will be disabled).

SPI_ERR_04

SPI Module

Category

Functional

Function

IDLE/BUSY status toggle after each frame receive when SPI peripheral is in only receive mode.

Description

In case of SPI peripheral in only receiving mode, the IDLE interrupt and BUSY status are toggling after each frame receive while SPI is receiving data continuously(SPI_PHASE=1). Here there is no data loaded into peripheral(slave) TXFIFO and TXFIFO is empty.

Workaround

Do not use SPI peripheral only receive mode. Set SPI in peripheral(slave) simultaneous transmit and receive mode.

SPI_ERR_05

SPI Module

Category

Functional

Function

SPI Peripheral Receive Timeout interrupt is setting irrespective of RXFIFO data

Description

When using SPI timeout interrupt, the RXTIMEOUT counter started decrementing from the point that peripheral is stopped receiving SPI clock and setting the RXTIMEOUT interrupt irrespective of data exists in RXFIFO or not, which does not match the description in the TRM: SPI peripheral receive timeout(RTOUT) interrupt is "asserted when the receive FIFO is not empty, and no further data is received in the specified time at CTL1.RXTIMEOUT.

Workaround

Repeat load RXTIMEROUT counter value while receive FIFO is empty, and start timeout counting only when receive FIFO gets any data.

SRAM_ERR_01

SRAM Module

Category

Functional

Function

Mixing No-ECC/Parity aperture and ECC aperture accesses from CPU/DMA can lead to functional errors

SRAM_ERR_01 (continued) *SRAM Module*

Description

If CPU accesses are configured to use the No-ECC/Parity aperture and DMA to use the ECC aperture, it may lead to intermittent faults.

Workaround

Do not mix and match No-ECC/Parity/ECC aperture accesses between CPU and DMA. Use the same type of aperture for both CPU and DMA accesses. If a safety mechanism requires injecting faults, then avoid using DMA concurrently while this safety diagnostic mechanism is being exercised by the CPU software.

SYSOSC_ERR_01 *SYSOSC Module*

Category

Functional

Function

MFCLK drift when using SYSOSC FCL together with STOP1 mode

Description

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.

TAMPERIO_ERR_01 *TAMPERIO Module*

Category

Functional

Function

Tamper IOMUX state will lose latch status when VDD supply lost

Description

LFSS I/O state (when configured in IOMUX mode) will be latched & retained over SHUTDOWN mode transition of SoC. But over a VDD power-cycle the state will be lost.

Workaround

To retain a LFSS I/O state over a VDD power-cycle, configure it in Tamper mode.

TIMER_ERR_01 *TIMx Module*

Category

Functional

TIMER_ERR_01 (continued)

TIMx Module

Function

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

Description

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.

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

TIMER_ERR_04

TIMER Module

Category

Functional

Function

TIMER re-enable may be missed if done close to zero event

Description

When using a GPTIMER in one shot mode and CLKDIV.RATIO is not 0, TIMER re-enable may be missed if done close to zero event.

Workaround

TIMER can be disabled first before re-enabling.

UART_ERR_01

UART Module

Category

Functional

Function

UART start condition not detected when transitioning to STANDBY1 Mode

Description

After servicing an asynchronous fast clock request that was initiated by a UART transmission while the device was in STANDBY1 mode, the device will return to STANDBY1 mode. If another UART transmission begins during the transition back to STANDBY1 mode, the data is not correctly detected and received by the device.

Workaround

Use STANDBY0 mode or higher low power mode when expecting repeated UART start conditions.

UART_ERR_02 *UART Module*

Category

Functional

Function

UART End of Transmission interrupt not set when only TXE is enabled

Description

UART End Of Transmission (EOT) interrupt does not trigger when the device is set for transmit only (CTL0.TXE = 1, CTL0.RXE = 0). EOT successfully triggers when device is set for transmit and receive (CTL0.TXE = 1, CTL0.RXE = 1)

Workaround

Set both CTL0.TXE and CTL0.RXE bits when utilizing the UART end of transmission interrupt. Note that you do not need to assign a pin as UART receive.

UART_ERR_03 *UART Module*

Category

Functional

Function

UART RX interrupt erroneously set with 3x oversampling and MFCLK or BUSCLK as clock source

Description

When using BUSCLK or MFCLK for UART and 3x oversampling, the RXINT is getting set incorrectly. When using the UART module with BUSCLK or MFCLK as the source, and 3x oversampling mode, the RX interrupt may be set erroneously. Under these conditions TX data may also be corrupted.

Workaround

Use a higher oversampling rate when using BUSCLK or MFCLK. If 3x oversampling is required, utilize LFCLK.

VREF_ERR_03

Category

Functional

Function

Vref Module will turn off when in standby mode when COMP is also active

Description

When using the COMP and internal VREF in STANDBY mode, the VREF module will turn off.

Workaround

Use COMP and internal VREF in STOP2 Mode.

7 Revision History

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

Changes from May 1, 2024 to September 30, 2024 (from Revision * (May 2024) to Revision A (October 2024)) **Page**

- ADC_ERR_03 Module was updated 5
- ADC_ERR_03 Function was updated 5
- ADC_ERR_03 Description was updated 5
- ADC_ERR_03 Workaround was updated 5
- ADC_ERR_05 Module was updated 5
- ADC_ERR_05 Workaround was updated 5
- ADC_ERR_05 Function was updated 5
- ADC_ERR_05 Description was updated 5
- ADC_ERR_06 Category was updated 5
- ADC_ERR_06 Module was updated 5
- ADC_ERR_06 Function was updated 5
- ADC_ERR_06 Description was updated 5
- ADC_ERR_06 Workaround was updated 5
- COMP_ERR_04 Category was updated 6
- COMP_ERR_04 Module was updated 6
- COMP_ERR_04 Function was updated 6
- COMP_ERR_04 Workaround was updated 6
- COMP_ERR_04 Description was updated 6
- GPIO_ERR_03 Module was updated 6
- GPIO_ERR_03 Function was updated 6
- GPIO_ERR_03 Description was updated 6
- GPIO_ERR_03 Workaround was updated 6
- GPIO_ERR_04 Module was updated 7
- GPIO_ERR_04 Function was updated 7
- GPIO_ERR_04 Description was updated 7
- GPIO_ERR_04 Workaround was updated 7
- I2C_ERR_05 Category was updated 7
- I2C_ERR_05 Module was updated 7
- I2C_ERR_05 Function was updated 7
- I2C_ERR_05 Description was updated 7
- I2C_ERR_05 Workaround was updated 7
- I2C_ERR_06 Module was updated 8
- I2C_ERR_06 Function was updated 8
- I2C_ERR_06 Description was updated 8
- I2C_ERR_06 Workaround was updated 8
- LFXT_ERR_01 Module was updated 8
- LFXT_ERR_01 Function was updated 8
- LFXT_ERR_01 Description was updated 8
- LFXT_ERR_01 Workaround was updated 8
- LFXT_ERR_02 Module was updated 9
- LFXT_ERR_02 Function was updated 9
- LFXT_ERR_02 Workaround was updated 9
- LFXT_ERR_02 Description was updated 9
- PMCU_ERR_07 Category was updated 9
- PMCU_ERR_07 Module was updated 9
- PMCU_ERR_07 Function was updated 9
- PMCU_ERR_07 Description was updated 9
- PMCU_ERR_07 Workaround was updated 9
- PMCU_ERR_08 Module was updated 9
- PMCU_ERR_08 Function was updated 9
- PMCU_ERR_08 Workaround was updated 9
- PMCU_ERR_08 Description was updated 9
- PMCU_ERR_10 Category was updated 9

- PMCU_ERR_10 Module was updated.... 9
- PMCU_ERR_10 Function was updated.... 9
- PMCU_ERR_10 Description was updated.... 9
- PMCU_ERR_10 Workaround was updated.... 9
- PMCU_ERR_12 Category was updated.... 10
- PMCU_ERR_12 Module was updated.... 10
- RTC_A_ERR_02 Module was updated.... 10
- RTC_A_ERR_02 Function was updated.... 10
- RTC_A_ERR_02 Workaround was updated.... 10
- RTC_A_ERR_02 Description was updated.... 10
- SPI_ERR_03 Workaround was updated.... 10
- SPI_ERR_03 Module was updated.... 10
- SPI_ERR_03 Function was updated.... 10
- SPI_ERR_03 Description was updated.... 10
- SPI_ERR_04 Category was updated.... 11
- SPI_ERR_04 Module was updated.... 11
- SPI_ERR_04 Function was updated.... 11
- SPI_ERR_04 Description was updated.... 11
- SPI_ERR_04 Workaround was updated.... 11
- SPI_ERR_05 Category was updated.... 11
- SPI_ERR_05 Module was updated.... 11
- SPI_ERR_05 Function was updated.... 11
- SPI_ERR_05 Description was updated.... 11
- SPI_ERR_05 Workaround was updated.... 11
- TAMPERIO_ERR_01 Module was updated.... 12
- TAMPERIO_ERR_01 Function was updated.... 12
- TAMPERIO_ERR_01 Description was updated.... 12
- TAMPERIO_ERR_01 Workaround was updated.... 12
- TIMER_ERR_04 Category was updated.... 13
- TIMER_ERR_04 Function was updated.... 13
- TIMER_ERR_04 Description was updated.... 13
- TIMER_ERR_04 Workaround was updated.... 13
- UART_ERR_01 Function was updated.... 13
- UART_ERR_01 Description was updated.... 13
- UART_ERR_01 Workaround was updated.... 13
- UART_ERR_02 Function was updated.... 14
- UART_ERR_02 Module was updated.... 14
- UART_ERR_02 Description was updated.... 14
- UART_ERR_02 Workaround was updated.... 14
- UART_ERR_03 Category was updated.... 14
- UART_ERR_03 Module was updated.... 14
- UART_ERR_03 Function was updated.... 14
- UART_ERR_03 Description was updated.... 14
- UART_ERR_03 Workaround was updated.... 14

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