Errata

MSPM0C1103, MSPM0C1104, MSPS003F3, MSPS003F4 Microcontrollers

ABSTRACT

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

Table of Contents

1 Functional Advisories

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

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

Errata Number	Rev C
ADC_ERR_06	✓
I2C_ERR_03	✓
PMCU_ERR_07	✓

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 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.

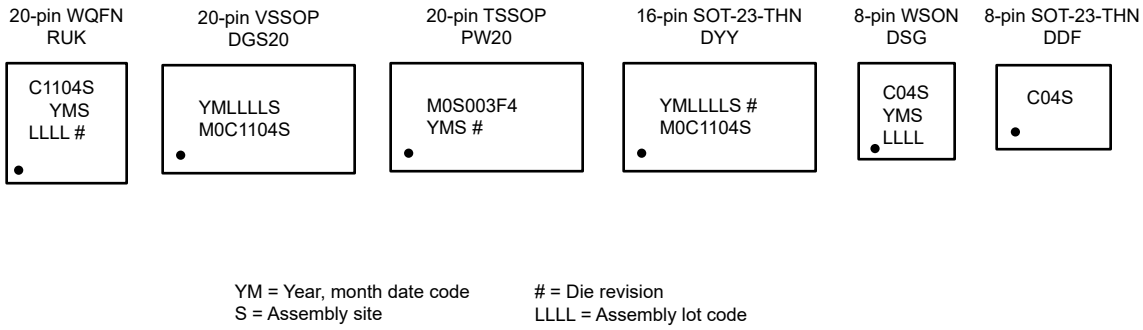

Figure 5-1. Package Symbolization

Table 5-1. Die Revisions

Revision Letter (package marking)	Version (in the device factory constants memory)
B	2

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_06 *ADC offset error needs to be calibrated in application code*

Revisions Affected

Rev C

Details

The calibration data of ADC offset error is not applied correctly AND needs to be implemented in application code.

Workaround

The calibration data is stored at address 0x41C40040 in factory region. Two DriverLib APIs ***DL_ADC12_getADCOffsetCalibration*** AND ***DL_FactoryRegion_getADCOffset*** have been implemented in the SDK to facilitate this.

```
__STATIC_INLINE int16_t DL_ADC12_getADCOffsetCalibration(float userRef)
{
float adcBuff = DL_FactoryRegion_getADCOffset() * (3.3 / userRef);
return (int16_t)(round(adcBuff));
}
__STATIC_INLINE float DL_FactoryRegion_getADCOffset(void)
{
return ((float) (*(int16_t *) (0x41C40040)));
}
```

The calibration data can be saved into a variable AND subsequently applied to the ADC conversion result.

Below is the example code demonstrating how to apply the calibration data, which has been integrated into the ADC examples provided in the SDK:

```
volatile uint16_t gAdcResult;
volatile int16_t gADCOffset;
gADCOffset =
DL_ADC12_getADCOffsetCalibration(ADC12_0_ADCMEM_0_REF_VOLTAGE_V);
gAdcResult = DL_ADC12_getMemResult(ADC12_0_INST, DL_ADC12_MEM_IDX_0);
int16_t adcRaw = (int16_t) gAdcResult + gADCOffset;
if (adcRaw < 0) {
adcRaw = 0;
}
if (adcRaw > 4095) {
adcRaw = 4095;
}
gAdcResult = (uint16_t) adcRaw;
```

The ADC offset calibration data can be applied for the example use cases below:

ADC_ERR_06 (continued)

ADC offset error needs to be calibrated in application code

- Utilizing ADC without DMA - the offset needs to be added to ADC result from the register MEMRES.
- Utilizing ADC with DMA - the offset needs to be added to ADC result stored in the memory address.
- Utilizing ADC window comparator - the offset needs to be subtracted from window threshold used when configuring the comparator feature.

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

Revisions Affected

Rev C

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.

PMCU_ERR_07 ***The LFOSCGOOD bit in the CLKSTATUS register does not indicate the correct LFOSC status***

Revisions Affected

Rev C

Details

When LFOSC startup has completed, the LFOSCGOOD bit has not been asserted by the LFOSC startup monitor.

Workaround

Avoid using the LFOSCGOOD bit to check if the LFOSC is ready or not.

7 Revision History

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

Changes from Revision * (October 2023) to Revision A (May 2024)	Page
• Removed ADC_ERR_04, ADC_ERR_05, PMCU_ERR_04, PMCU_ERR_05, PMCU_ERR_06, and UART_ERR_01; Added ADC_ERR_06, I2C_ERR_03, and PMCU_ERR_07	1

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