

# MSP430F5172 Device Erratasheet

---



---



---

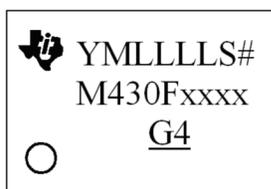
## 1 Revision History

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

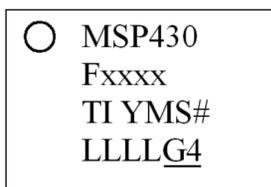
The revision of the device can be identified by the revision letter on the [Package Markings](#) or by the [HW\\_ID](#) located inside the TLV structure of the device

Errata Number	Rev C
<a href="#">ADC39</a>	✓
<a href="#">ADC42</a>	✓
<a href="#">BSL7</a>	✓
<a href="#">CPU40</a>	✓
<a href="#">DMA4</a>	✓
<a href="#">DMA7</a>	✓
<a href="#">DMA10</a>	✓
<a href="#">EEM11</a>	✓
<a href="#">EEM17</a>	✓
<a href="#">EEM19</a>	✓
<a href="#">EEM21</a>	✓
<a href="#">EEM23</a>	✓
<a href="#">JTAG26</a>	✓
<a href="#">PMAP1</a>	✓
<a href="#">PMM14</a>	✓
<a href="#">PMM15</a>	✓
<a href="#">PMM18</a>	✓
<a href="#">PMM20</a>	✓
<a href="#">PMM26</a>	✓
<a href="#">PORT15</a>	✓
<a href="#">PORT19</a>	✓
<a href="#">PORT21</a>	✓
<a href="#">SYS12</a>	✓
<a href="#">SYS16</a>	✓
<a href="#">TD1</a>	✓
<a href="#">TD2</a>	✓
<a href="#">UCS9</a>	✓
<a href="#">UCS11</a>	✓
<a href="#">USCI26</a>	✓
<a href="#">USCI31</a>	✓
<a href="#">USCI35</a>	✓
<a href="#">USCI39</a>	✓
<a href="#">USCI40</a>	✓

## 2 Package Markings

**DA38**
**TSSOP (DA), 38 Pin**


YM = Year and Month Date Code  
 LLLL = Assembly Lot Code  
 S = Assembly Site Code  
 # = DIE Revision  
 o = PIN 1

**RSB40**
**QFN (RSB), 40 Pin**


YM = Year and Month Date Code  
 LLLL = Assembly Lot Code  
 S = Assembly Site Code  
 # = DIE Revision  
 o = PIN 1

## 3 TLV Hardware Revision

Die Revision	TLV Hardware Revision
Rev C	30h

Further guidance on how to locate the TLV structure and read out the HW\_ID can be found in the device User's Guide.

## 4 Detailed Bug Description

<b>ADC39</b>	<b><i>ADC10_A Module</i></b>
<b>Function</b>	Erroneous ADC10 results in extended sample mode
<b>Description</b>	If the extended sample mode is selected (ADC10SHP = 0) and the ADC10CLK is asynchronous to the SHI signal, the ADC10 may generate erroneous results.
<b>Workaround</b>	1) Use the pulse sample mode (ADC10SHP=1) OR 2) Use a synchronous clock for ADC10 and the SHI signal.
<b>ADC42</b>	<b><i>ADC10_A Module</i></b>
<b>Function</b>	ADC stops converting when successive ADC is triggered before the previous conversion ends
<b>Description</b>	Subsequent ADC conversions are halted if a new ADC conversion is triggered while ADC is busy. ADC conversions are triggered manually or by a timer. The affected ADC modes are:  - sequence-of-channels - repeat-single-channel - repeat-sequence-of-channels (ADC12CTL1.ADC12CONSEQx)  In addition, the timer overflow flag cannot be used to detect an overflow (ADC12IFGR2.ADC12TOVIFG).
<b>Workaround</b>	1. For manual trigger mode (ADC12CTL0.ADC12SC), ensure each ADC conversion is completed by first checking ADC12CTL1.ADC12BUSY bit before starting a new conversion.  2. For timer trigger mode (ADC12CTL1.ADC12SHP), ensure the timer period is greater than the ADC sample and conversion time.  To recover the conversion halt:  1. Disable ADC module (ADC12CTL0.ADC12ENC = 0 and ADC12CTL0.ADC12ON = 0) 2. Re-enable ADC module (ADC12CTL0.ADC12ON = 1 and ADC12CTL0.ADC12ENC = 1) 3. Re-enable conversion
<b>BSL7</b>	<b><i>BSL Module</i></b>
<b>Function</b>	BSL does not start after waking up from LPMx.5
<b>Description</b>	When waking up from LPMx.5 mode, the BSL does not start as it does not clear the Lock I/O bit (LOCKLPM5 bit in PM5CTL0 register) on start-up.
<b>Workaround</b>	1. Upgrade the device BSL to the latest version (see Creating a Custom Flash-Based Bootstrap Loader (BSL) Application Note - SLAA450 for more details)  OR  2. Do not use LOCKLPM5 bit (LPMx.5) if the BSL is used but cannot be upgraded.

<b>CPU40</b>	<b><i>CPUXv2 Module</i></b>
<b>Function</b>	PC is corrupted when executing jump/conditional jump instruction that is followed by instruction with PC as destination register or a data section
<b>Description</b>	<p>If the value at the memory location immediately following a jump/conditional jump instruction is 0X40h or 0X50h (where X = don't care), which could either be an instruction opcode (for instructions like RRCM, RRAM, RLAM, RRUM) with PC as destination register or a data section (const data in flash memory or data variable in RAM), then the PC value is auto-incremented by 2 after the jump instruction is executed; therefore, branching to a wrong address location in code and leading to wrong program execution.</p> <p>For example, a conditional jump instruction followed by data section (0140h).</p> <pre>@0x8012 Loop DEC.W R6 @0x8014 DEC.W R7 @0x8016 JNZ Loop @0x8018 Value1 DW 0140h</pre>
<b>Workaround</b>	<p>In assembly, insert a NOP between the jump/conditional jump instruction and program code with instruction that contains PC as destination register or the data section.</p> <p>In C, no workaround is necessary since the compiler automatically generates the necessary NOPs.</p>
<b>DMA4</b>	<b><i>DMA Module</i></b>
<b>Function</b>	Corrupted write access to 20-bit DMA registers
<b>Description</b>	When a 20-bit wide write to a DMA address register (DMAxSA or DMAxDA) is interrupted by a DMA transfer, the register contents may be unpredictable.
<b>Workaround</b>	<ol style="list-style-type: none"> <li>1. Design the application to guarantee that no DMA access interrupts 20-bit wide accesses to the DMA address registers.</li> </ol> <p>OR</p> <ol style="list-style-type: none"> <li>2. When accessing the DMA address registers, enable the Read Modify Write disable bit (DMARMWDIS = 1) or temporarily disable all active DMA channels (DMAEN = 0).</li> </ol> <p>OR</p> <ol style="list-style-type: none"> <li>3. Use word access for accessing the DMA address registers. Note that this limits the values that can be written to the address registers to 16-bit values (lower 64K of Flash).</li> </ol>
<b>DMA7</b>	<b><i>DMA Module</i></b>
<b>Function</b>	DMA request may cause the loss of interrupts
<b>Description</b>	If a DMA request starts executing during the time when a module register containing an interrupt flags is accessed with a read-modify-write instruction, a newly arriving interrupt from the same module can get lost. An interrupt flag set prior to DMA execution would not be affected and remain set.
<b>Workaround</b>	<ol style="list-style-type: none"> <li>1. Use a read of Interrupt Vector registers to clear interrupt flags and do not use read-modify-write instruction.</li> </ol> <p>OR</p>

2. Disable all DMA channels during read-modify-write instruction of specific module registers containing interrupts flags while these interrupts are activated.

**DMA10**
***DMA Module***


---

**Function**

DMA access may cause invalid module operation

**Description**

The peripheral modules MPY, CRC, USB, RF1A and FRAM controller in manual mode can stall the CPU by issuing wait states while in operation. If a DMA access to the module occurs while that module is issuing a wait state, the module may exhibit undefined behavior.

**Workaround**

Ensure that DMA accesses to the affected modules occur only when the modules are not in operation. For example with the MPY module, ensure that the MPY operation is completed before triggering a DMA access to the MPY module.

**EEM11**
***EEM Module***


---

**Function**

Conditional register write trigger fails while executing rotate instructions

**Description**

A conditional register write trigger will fail to generate the expected breakpoint if the trigger condition is a result of executing one of the following rotate instructions: RRUM, RRCM, RRAM and RLAM.

**Workaround**

None

---

**NOTE:** This erratum applies to debug mode only.

---

**EEM17**
***EEM Module***


---

**Function**

Wrong Breakpoint halt after executing Flash Erase/Write instructions

**Description**

Hardware breakpoints or Conditional Address triggered breakpoints on instructions that follow Flash Erase/Write instructions, stops the debugger at the actual Flash Erase/Write instruction even though the flash erase/write operation has already been executed. The hardware/conditional address triggered breakpoints that are placed on either the next two single opcode instructions OR the next double opcode instruction that follows the Flash Erase/Write instruction are affected by this erratum.

**Workaround**

None. Use other conditional/advanced triggered breakpoints to halt the debugger right after Flash erase/write instructions.

---

**NOTE:** This erratum affects debug mode only.

---

**EEM19**
***EEM Module***


---

**Function**

DMA may corrupt data in debug mode

**Description**

When the DMA is enabled and the device is in debug mode, the data written by the DMA may be corrupted when a breakpoint is hit or when the debug session is halted.

**Workaround**

This erratum has been addressed in MSPDebugStack version 3.5.0.1. It is also available in released IDE EW430 IAR version 6.30.3 and CCS version 6.1.1 or newer.

If using an earlier version of either IDE or MSPDebugStack, do not halt or use breakpoints during a DMA transfer.

---

**NOTE:** This erratum applies to debug mode only.

---

**EEM21**
***EEM Module***


---

**Function**

LPMx.5 debug limitations

**Description**

Debugging the device in LPMx.5 mode might wake the device up from LPMx.5 mode inadvertently, and it is possible that the device enters a lock-up condition; that is, the device cannot be accessed by the debugger any more.

**Workaround**

Follow the debugging steps in Debugging MSP430 LPM4.5 [SLAA424](#) .

**EEM23**
***EEM Module***


---

**Function**

EEM triggers incorrectly when modules using wait states are enabled

**Description**

When modules using wait states (USB, MPY, CRC and FRAM controller in manual mode) are enabled, the EEM may trigger incorrectly. This can lead to an incorrect profile counter value or cause issues with the EEMs data watch point, state storage, and breakpoint functionality.

**Workaround**

None.

---

**NOTE:** This erratum affects debug mode only.

---

**JTAG26**
***JTAG Module***


---

**Function**

LPMx.5 Debug Support Limitations

**Description**

The JTAG connection to the device might fail at device-dependent low or high supply voltage levels if the LPMx.5 debug support feature is enabled. To avoid a potentially unreliable debug session or general issues with JTAG device connectivity and the resulting bad customer experience Texas Instruments has chosen to remove the LPMx.5 debug support feature from common MSP430 IDEs including TIs Code Composer Studio 6.1.0 with msp430.emu updated to version 6.1.0.7 and IARs Embedded Workbench 6.30.2, which are based on the MSP430 debug stack MSP430.DLL 3.5.0.1 <http://www.ti.com/tool/MSPDS>

TI plans to re-introduce this feature in limited capacity in a future release of the debug stack by providing an IDE override option for customers to selectively re-activate LPMx.5 debug support if needed. Note that the limitations and supply voltage dependencies outlined in this erratum will continue to apply.

For additional information on how the LPMx.5 debug support is handled within the MSP430 IDEs including possible workarounds on how to debug applications using LPMx.5 without toolchain support refer to [Code Composer Studio User's Guide for MSP430 chapter F.4](#) and [IAR Embedded Workbench User's Guide for MSP430 chapter 2.2.5](#).

**Workaround**

1. If LPMx.5 debug support is deemed functional and required in a given scenario:

a) Do not update the IDE to continue using a previous version of the debug stack such as MSP430.DLL v3.4.3.4.

OR

b) Roll back the debug stack by either performing a clean re-installation of a previous version of the IDE or by manually replacing the debug stack with a prior version such as MSP430.DLL v3.4.3.4 that can be obtained from <http://www.ti.com/tool/MSPDS>.

2. In case JTAG connectivity fails during the LPMx.5 debug mode, the device supply voltage level needs to be raised or lowered until the connection is working.

Do not enable the LPMx.5 debug support feature during production programming.

## **PMAP1**

### ***PMAP Module***

---

#### **Function**

Port Mapping Controller does not clear unselected inputs to mapped module.

#### **Description**

The Port Mapping Controller provides the logical OR of all port mapped inputs to a module (Timer, USCI, etc). If the PSEL bit (PxSEL.y) of a port mapped input is cleared, then the logic level of that port mapped input is latched to the current logic level of the input. If the input is in a logical high state, then this high state is latched into the input of the logical OR. In this case, the input to the module is always a logical 1 regardless of the state of the selected input.

#### **Workaround**

1. Drive input to the low state before clearing the PSEL bit of that input and switching to another input source.

or

2. Use the Port Mapping Controller reconfiguration feature, PMAPRECFG, to select inputs to a module and map only one input at a time.

## **PMM14**

### ***PMM Module***

---

#### **Function**

Increasing the core level when SVS/SVM low side is configured in full-performance mode causes device reset

#### **Description**

When the SVS/SVM low side is configured in full performance mode (SVSMLCTL.SVSLFP = 1), the setting time delay for the SVS comparators is ~2us. When increasing the core level in full-performance mode; the core voltage does not settle to the new level before the settling time delay of the SVS/SVM comparator expires. This results in a device reset.

#### **Workaround**

When increasing the core level; enable the SVS/SVM low side in normal mode (SVSMLCTL.SVSLFP=0). This provides a settling time delay of approximately 150us allowing the core sufficient time to increase to the expected voltage before the delay expires.

## **PMM15**

### ***PMM Module***

---

#### **Function**

Device may not wake up from LPM2, LPM3, or LPM4

#### **Description**

Device may not wake up from LPM2, LPM3 or LPM4 if an interrupt occurs within 1 us after the entry to the specified LPMx; entry can be caused either by user code or automatically (for example, after a previous ISR is completed). Device can be recovered with an external reset or a power cycle. Additionally, a PUC can also be used to reset the failing condition and bring the device back to normal operation (for example, a PUC caused by the WDT).

This effect is seen when:

- A write to the SVSMHCTL and SVSMLCTL registers is immediately followed by an LPM2, LPM3, LPM4 entry without waiting the requisite settling time

((PMMIFG.SVSMLDLYIFG = 0 and PMMIFG.SVSMHDLYIFG = 0)).

or

The following two conditions are met:

- The SVSL module is configured for a fast wake-up or when the SVSL/SVML module is turned off. The affected SVSMLCTL register settings are shaded in the following table.

SVSL	SVSLE	SVSLMD	SVSLFP	AM, LPM0/1 SVSL state	Manual SVSMLACE = 0	Automatic SVSMLACE = 1	Wakeup Time LPM2/3/4
					LPM2/3/4 SVSL State	LPM2/3/4 SVSL State	
	0	x	x	OFF	OFF	OFF	t <sub>WAKE-UP FAST</sub>
	1	0	0	Normal	OFF	OFF	t <sub>WAKE-UP SLOW</sub>
	1	0	1	Full Performance	OFF	OFF	t <sub>WAKE-UP FAST</sub>
	1	1	0	Normal	Normal	OFF	t <sub>WAKE-UP SLOW</sub>
	1	1	1	Full Performance	Full Performance	Normal	t <sub>WAKE-UP FAST</sub>

SVML	SVMLE	SVMLFP	AM, LPM0/1 SVML state	Manual SVSMLACE = 0	Automatic SVSMLACE = 1	Wakeup Time LPM2/3/4
				LPM2/3/4 SVML State	LPM2/3/4 SVML State	
	0	x	OFF	OFF	OFF	t <sub>WAKE-UP FAST</sub>
	1	0	Normal	Normal	OFF	t <sub>WAKE-UP SLOW</sub>
	1	1	Full Performance	Full Performance	Normal	t <sub>WAKE-UP FAST</sub>

and

-The SVSH/SVMH module is configured to transition from Normal mode to an OFF state when moving from Active/LPM0/LPM1 into LPM2/LPM3/LPM4 modes. The affected SVSMHCTL register settings are shaded in the following table.

SVSH	SVSHE	SVSHMD	SVSHFP	AM, LPM0/1 SVSH state	Manual SVSMHACE = 0	Manual SVSMHACE = 1
					LPM2/3/4 SVSH State	LPM2/3/4 SVSH State
	0	x	x	OFF	OFF	OFF
	1	0	0	Normal	OFF	OFF
	1	0	1	Full Performance	OFF	OFF
	1	1	0	Normal	Normal	OFF
	1	1	1	Full Performance	Full Performance	Normal

SVMH	SVSHE	SVMHFP	AM, LPM0/1 SVSH state	Manual SVSMHACE = 0	Manual SVSMHACE = 1
				LPM2/3/4 SVSH State	LPM2/3/4 SVSH State
	0	x	OFF	OFF	OFF
	1	0	Normal	Normal	OFF
	1	1	Full Performance	Full Performance	Normal

#### Workaround

Any write to the SVSMxCTL register must be followed by a settling delay (PMMIFG.SVSMLDLYIFG = 0 and PMMIFG.SVSMHDLYIFG = 0) before entering LPM2, LPM3, LPM4.

and

1. Ensure the SVSx, SVMx are configured to prevent the issue from occurring by the following:

- Configure the SVSL module for slow wake up (SVSLFP = 0). Note that this will increase the wakeup time from LPM2/3/4 to twakeupslow (~150 us).

or

- Do not configure the SVSH/SVMH such that the modules transition from Normal mode to an OFF state on LPM entry. Instead force the modules to remain ON even in LPMx.

Note that this will cause increased power consumption when in LPMx.

Refer to the MSP430F5xx and MSP430F6xx Core Libraries ([SLAA448](#)) for proper PMM configuration functions.

Use the following function, PMM15Check (void), to determine whether or not the existing PMM configuration is affected by the erratum. The return value of the function is 1 if the configuration is affected, and 0 if the configuration is not affected.

```

unsigned char PMM15Check (void)
{
// First check if SVSL/SVML is configured for fast wake-up
if ( (!(SVSMLCTL & SVSLE)) || ((SVSMLCTL & SVSLE) && (SVSMLCTL & SVSLFP)) ||
(!(SVSMLCTL & SVMLE)) || ((SVSMLCTL & SVMLE) && (SVSMLCTL & SVMLFP)) )
{ // Next Check SVSH/SVMH settings to see if settings are affected by PMM15
if ((SVSMHCTL & SVSHE) && !(SVSMHCTL & SVSHFP))
{
if ( (!(SVSMHCTL & SVSHMD)) || ((SVSMHCTL & SVSHMD) &&
(SVSMHCTL & SVSMHACE)) )
return 1; // SVSH affected configurations
}
if ((SVSMHCTL & SVMHE) && !(SVSMHCTL & SVMHFP) && (SVSMHCTL &
SVSMHACE))
return 1; // SVMH affected configurations
}
return 0; // SVS/M settings not affected by PMM15
}
}

```

2. If fast servicing of interrupts is required, add a 150us delay either in the interrupt service routine or before entry into LPM3/LPM4.

## **PMM18**

### ***PMM Module***

#### **Function**

PMM supply overvoltage protection falsely triggers POR

#### **Description**

The PMM Supply Voltage Monitor (SVM) high side can be configured as overvoltage protection (OVP) using the SVMHOVPE bit of SVSMHCTL register. In this mode a POR should typically be triggered when DVCC reaches ~3.75V.

If the OVP feature of SVM high side is enabled going into LPM234, the SVM might trigger at DVCC voltages below 3.6V (~3.5V) within a few ns after wake-up. This can falsely cause an OVP-triggered POR. The OVP level is temperature sensitive during fail scenario and decreases with higher temperature (85 degC ~3.2V).

#### **Workaround**

Use automatic control mode for high-side SVS & SVM (SVSMHCTL.SVSMHACE=1). The SVM high side is inactive in LPM2, LPM3, and LPM4.

## **PMM20**

### ***PMM Module***

#### **Function**

Unexpected SVSL/SVML event during wakeup from LPM2/3/4 in fast wakeup mode

<b>Description</b>	<p>If PMM low side is configured to operate in fast wakeup mode, during wakeup from LPM2/3/4 the internal V<sub>CORE</sub> voltage can experience voltage drop below the corresponding SVSL and SVML threshold (recommendation according to User's Guide) leading to an unexpected SVSL/SVML event. Depending on PMM configuration, this event triggers a POR or an interrupt.</p> <hr/> <p><b>NOTE:</b> As soon the SVSL or the SVML is enabled in Normal performance mode the device is in slow wakeup mode and this erratum does not apply.</p> <p>In addition, this erratum has sporadic characteristic due to an internal asynchronous circuit. The drop of V<sub>core</sub> does not have an impact on specified device performance.</p> <hr/>
<b>Workaround</b>	<p>If SVSL or SVML is required for application (to observe external disruptive events at V<sub>core</sub> pin) the slow wakeup mode has to be used to avoid unexpected SVSL/SVML events. This is achieved if the SVSL or the SVML is configured in "Normal" performance mode (not disabled and not in "Full" Performance Mode).</p>
<b>PMM26</b>	<p><b><i>PMM Module</i></b></p> <hr/>
<b>Function</b>	<p>Device lock-up if RST pin pulled low during write to SVSMHCTL or SVSMLCTL</p>
<b>Description</b>	<p>Device results in lock-up condition under one of the two scenarios below:</p> <p>1) If RST pin is pulled low during write access to SVSMHCTL, with the RST/NMI pin is configured to reset function and is pulled low (reset event) the device will stop code execution and is continuously held in reset state. RST pin is no longer functional. The only way to come out of the lock-up situation is a power cycle.</p> <p>OR</p> <p>2) If RST pin is pulled low during write access to SVSMLCTL and only if the code that checks for SVSMLDLYIFG==1 is implemented without a timeout. The device will be stuck in the polling loop polling since SVSMLDLYIFG will never be cleared.</p>
<b>Workaround</b>	<p>Follow the sequence below to prevent the lock-up for both use cases:</p> <p>1) Disable RST pin reset function and switch to NMI before access SVSMHCTL or SVSMLCTL.</p> <p>then</p> <p>2) Activate NMI interrupt and handle reset events in this time by SW (optional if reset functionality required during access SVSMHCTL or SVSMLCTL)</p> <p>then</p> <p>3) Enable RST pin reset function after access to SVSMHCTL or SVSMLCTL</p> <p>To prevent lock-up caused by use case #2 a timeout for the SVSMLDLYIFG flag check should be implemented to 300us.</p>
<b>PORT15</b>	<p><b><i>PORT Module</i></b></p> <hr/>
<b>Function</b>	<p>In-system debugging causes the PMALOCKED bit to be always set</p>
<b>Description</b>	<p>The port mapping controller registers cannot be modified when single-stepping or halting at break points between a valid password write to the PMAPWD register and the expected lock of the port mapping (PMAP) registers. This causes the PMALOCKED bit to remain set and not clear as expected.</p> <p>Note: This erratum only applies to in-system debugging and is not applicable when</p>

operating in free-running mode.

**Workaround** Do not single step through or place break points in the port mapping configuration section of code.

## PORT19

### *PORT Module*

---

**Function** Port interrupt may be missed on entry to LPMx.5

**Description** If a port interrupt occurs within a small timing window (~1MCLK cycle) of the device entry into LPM3.5 or LPM4.5, it is possible that the interrupt is lost. Hence this interrupt will not trigger a wakeup from LPMx.5.

**Workaround** None

## PORT21

### *PORT Module*

---

**Function** Setting PxSEL bit for XTAL pins

**Description** Setting the PxSEL bit of XIN pin does not disable the digital function of the XOUT pin (in non-bypass mode). The primary port function will still be active on the XOUT pin.

**Workaround** Set the PxSEL bit of XOUT pin explicitly to disable the port function of the XOUT pin.

## SYS12

### *SYS Module*

---

**Function** Invalid ACCVIFG when DVcc in the range of 2.4 to 2.6V

**Description** A Flash Access Violation Interrupt Flag (ACCVIFG) may be triggered by the Voltage Changed During Program Error bit (VPE) when DVcc is in the range of 2.4 to 2.6V. Although this behavior is expected according to the user's guide, the VPE does not signify an invalid flash operation has occurred.

If the ACCVIE bit is set and a flash operation is executed in the affected voltage range, an unnecessary interrupt is requested. The bootstrap loader also cannot be used to execute write/erase flash operations in this voltage range, because it exits the flash operation and returns an error on an ACCVIFG event.

**Workaround** None

## SYS16

### *SYS Module*

---

**Function** Fast Vcc ramp after device power up may cause a reset

**Description** At initial power-up, after Vcc crosses the brownout threshold and reaches a constant level, an abrupt ramp of Vcc at a rate  $dV/dT > 1V/100\mu s$  can cause a brownout condition to be incorrectly detected even though Vcc does not fall below the brownout threshold. This causes the device to undergo a reset.

**Workaround** Use a controlled Vcc ramp to power up the device.

## TD1

### *TIMER\_D Module*

---

**Function** Timer halt on EXTCLR event

<b>Description</b>	When the TEC module is configured to enable external asynchronous signals on the TECxCLR (TEC external clear) pin to clear the timer counter on the selected edge and the timer is halted after an external clear event but before next positive edge of the timer clock, then due to the erratum, it is not possible to write to the timer counter (TDxR) until the next positive timer clock edge occurrence. Halting of the timer right after an external clear event is possible if the system clock (MCLK) much greater than the Timer_D clock and the application halts the timer in the external clear interrupt service routine or if the application is doing random timer halts. This erratum does not cause any dead-lock conditions and the timer counter can be written into when the next positive timer clock edge is available.
<b>Workaround</b>	If required to halt the timer and change the timer counter value on an external clear event, wait until one timer clock period has elapsed to change the timer counter value.
<hr/>	
<b>TD2</b>	<b><i>TIMER_D Module</i></b>
<b>Function</b>	Up/Down mode with high resolution enabled
<b>Description</b>	The Timer_D Up/Down mode is not functional in high-resolution mode. The PWM signals are generated incorrectly in the down phase.  Timer_D can only be used in Up mode in high-resolution mode.
<b>Workaround</b>	None
<hr/>	
<b>UCS9</b>	<b><i>UCS Module</i></b>
<b>Function</b>	Digital Bypass mode prevents entry into LPM4
<b>Description</b>	When entering LPM4, if an external digital input applied to XT1 in HF mode or XT2 is not turned off, the PMM does not switch to low-current mode causing higher than expected power consumption.
<b>Workaround</b>	Before entering LPM4:  (1) Switch to a clock source other than external bypass digital input.  OR  (2) Turn off external bypass mode (UCSCTL6.XT1BYPASS = 0).
<hr/>	
<b>UCS11</b>	<b><i>UCS Module</i></b>
<b>Function</b>	Modifying UCSCTL4 clock control register triggers an erroneous clock source request
<b>Description</b>	Changing the SELM/SELS/SELA bits in the UCSCTL4 register might trigger the respective clocks to select an incorrect clock source which requests the XT1/XT2 clock. If the crystals are not present at XT1/XT2 or present but not yet configured in the application firmware, then the respective XT1/XT2 fault flag is falsely set.
<b>Workaround</b>	Clear all the fault flags in UCSCTL7 register once after changing any of the SELM/SELS/SELA bits in the UCSCTL4 register.
<hr/>	
<b>USCI26</b>	<b><i>USCI Module</i></b>
<b>Function</b>	Tbuf parameter violation in I2C multi-master mode
<b>Description</b>	In multi-master I2C systems the timing parameter Tbuf (bus free time between a stop

condition and the following start) is not guaranteed to match the I2C specification of 4.7us in standard mode and 1.3us in fast mode. If the UCTXSTT bit is set during a running I2C transaction, the USCI module waits and issues the start condition on bus release causing the violation to occur.

Note: It is recommended to check if UCBBUSY bit is cleared before setting UCTXSTT=1.

**Workaround** None

### **USCI31** *USCI Module*

---

**Function** Framing Error after USCI SW Reset (UCSWRST)

**Description** While receiving a byte over USCI-UART (with UCBBUSY bit set), if the application resets the USCI module (software reset via UCSWRST), then a framing error is reported for the next receiving byte.

**Workaround**

1. If possible, do not reset USCI-UART during an ongoing receive operation; that is, when UCBBUSY bit is set.
2. If the application software resets the USCI module (via the UCSWRST bit) during an ongoing receive operation, then set and reset the UCSYNC bit before releasing the software USCI reset.

Workaround code sequence:

```
bis #UCSWRST, &UCAxCTL1 ; USCI SW reset
```

```
;Workaround begins
```

```
bis #UCSYNC, &UCAxCTL0 ; set synchronous mode
```

```
bic #UCSYNC, &UCAxCTL0 ; reset synchronous mode
```

```
;Workaround ends
```

```
bic #UCSWRST, &UCAxCTL1 ; release USCI reset
```

### **USCI35** *USCI Module*

---

**Function** Violation of setup and hold times for (repeated) start in I2C master mode

**Description** In I2C master mode, the setup and hold times for a (repeated) START,  $t_{SU,STA}$  and  $t_{HD,STA}$  respectively, can be violated if SCL clock frequency is greater than 50kHz in standard mode (100kbps). As a result, a slave can receive incorrect data or the I2C bus can be stalled due to clock stretching by the slave.

**Workaround** If using repeated start, ensure SCL clock frequencies is < 50kHz in I2C standard mode (100 kbps).

### **USCI39** *USCI Module*

---

**Function** USCI I2C IFGs UCSTTIFG, UCSTPIFG, UCNACKIFG

**Description** Unpredictable code execution can occur if one of the hardware-clear-able IFGs UCSTTIFG, UCSTPIFG or UCNACKIFG is set while the global interrupt enable is set by software (GIE=1). This erratum is triggered if ALL of the following events occur in following order:

1. Pending Interrupt: One of the UCxIFG=1 AND UCxIE=1 while GIE=0

2. The GIE is set by software (e.g. EINT)
3. The pending interrupt is cleared by hardware (external I2C event) in a time window of 1 MCLK clock cycle after the "EINT" instruction is executed.

**Workaround**

Disable the UCSTTIFG, UCSTPIFG and UCNACKIFG before the GIE is set. After GIE is set, the local interrupt enable flags can be set again.

Assembly example:

```
bic #UCNACKIE+UCSTPIE+UCSTTIE, UCBxIE ; disable all self-clearing interrupts
```

```
NOP
```

```
EINT
```

```
bis #UCNACKIE+UCSTPIE+UCSTTIE, UCBxIE ; enable all self-clearing interrupts
```

**USCI40**
***USCI Module***
**Function**

SPI Slave Transmit with clock phase select = 1

**Description**

In SPI slave mode with clock phase select set to 1 (UCAxCTLW0.UCCKPH=1), after the first TX byte, all following bytes are shifted by one bit with shift direction dependent on UCMSB. This is due to the internal shift register getting pre-loaded asynchronously when writing to the USCIA TXBUF register. TX data in the internal buffer is shifted by one bit after the RX data is received.

**Workaround**

Reinitialize TXBUF before using SPI and after each transmission.

If transmit data needs to be repeated with the next transmission, then write back previously read value:

```
UCAxTXBUF = UCAxTXBUF;
```

## 5 Document Revision History

Changes from family erratasheet to device specific erratasheet.

1. Errata CPU39 was removed
2. Errata EEM8 was removed
3. Errata EEM13 was removed
4. Errata FLASH37 was removed
5. Errata JTAG21 was removed
6. Errata PORT16 was removed
7. Errata MPY1 was removed
8. Errata PMM11 was removed
9. Errata PMM12 was removed
10. Errata SYS10 was removed
11. Errata UCS7 was removed
12. Errata UCS10 was removed
13. Errata USCI30 was removed
14. Revision A was removed
15. Revision B was removed

Changes from device specific erratasheet to document Revision A.

1. Errata PORT19 was added to the errata documentation.
2. Errata PMM18 was added to the errata documentation.

Changes from document Revision A to Revision B.

1. Errata DMA10 was added to the errata documentation.
2. Errata BSL7 was added to the errata documentation.

Changes from document Revision B to Revision C.

1. DMA10 Description was updated.
2. DMA10 Function was updated.

Changes from document Revision C to Revision D.

1. DMA10 Description was updated.
2. Errata EEM23 was added to the errata documentation.
3. Errata CPU43 was added to the errata documentation.

Changes from document Revision D to Revision E.

1. SYS16 Description was updated.
2. Errata TD1 was added to the errata documentation.
3. CPU43 Description was updated.
4. Errata PORT21 was added to the errata documentation.
5. Device TLV Hardware Revision information added to erratasheet.

Changes from document Revision E to Revision F.

1. Errata PMM20 was added to the errata documentation.
2. Errata USCI35 was added to the errata documentation.

Changes from document Revision F to Revision G.

1. BSL7 Workaround was updated.
2. BSL7 Function was updated.

Changes from document Revision G to Revision H.

1. EEM19 Workaround was updated.

2. EEM17 Workaround was updated.
3. Errata TD2 was added to the errata documentation.
4. CPU43 Description was updated.
5. EEM11 Workaround was updated.
6. EEM23 Workaround was updated.
7. EEM17 Description was updated.
8. EEM23 Description was updated.
9. Errata ADC39 was added to the errata documentation.
10. EEM19 Description was updated.

Changes from document Revision H to Revision I.

1. DMA10 Workaround was updated.
2. DMA10 Description was updated.
3. DMA10 Function was updated.

Changes from document Revision I to Revision J.

1. CPU40 Workaround was updated.
2. EEM19 Workaround was updated.
3. Errata USCI39 was added to the errata documentation.
4. Package Markings section was updated.
5. EEM23 Workaround was updated.
6. EEM23 Description was updated.
7. Errata ADC42 was added to the errata documentation.
8. EEM23 Function was updated.
9. EEM19 Description was updated.

Changes from document Revision J to Revision K.

1. Errata USCI40 was added to the errata documentation.
2. Errata CPU43 was removed from the errata documentation.
3. PMM18 Workaround was updated.

Changes from document Revision K to Revision L.

1. DMA7 Workaround was updated.
2. EEM23 Description was updated.
3. DMA7 Description was updated.

Changes from document Revision L to Revision M.

1. USCI39 Description was updated.

Changes from document Revision M to Revision N.

1. Errata JTAG26 was added to the errata documentation.

Changes from document Revision N to Revision O.

1. EEM19 Workaround was updated.
2. Errata PMM26 was added to the errata documentation.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)