

MSP430G2452 Device Erratasheet

1 Revision History

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

Errata Number	Rev B	Rev A
BCL12	✓	✓
BCL14	✓	✓
CPU4	✓	✓
EEM20	✓	✓
SYS15	✓	✓
TA12	✓	✓
TA16	✓	✓
TA21	✓	✓
TAB22	✓	✓
USI4	✓	✓
USI5	✓	✓
XOSC5	✓	✓

2 Package Markings


RSA16

QFN (RSA), 16 Pin

○ M430G	YM = Year and Month Date Code
xxxx	LLLL = Assembly Lot Code
TI YMS #	S = Assembly Site Code
LLLL	# = DIE Revision
	o = PIN 1


PW14

TSSOP (PW), 14 Pin

Gxxxx	YM = Year and Month Date Code
 YMSG4	LLLL = Assembly Lot Code
○ LLLL #	S = Assembly Site Code
	# = DIE Revision
	o = PIN 1


PW20

TSSOP (PW), 20 Pin

Gxxxx	YM = Year and Month Date Code
 YMSG4	LLLL = Assembly Lot Code
○ LLLL #	S = Assembly Site Code
	# = DIE Revision
	o = PIN 1

N20

PDIP (N), 20 Pin

 YMLLLLS #	YM = Year and Month Date Code
○ MSP430Gxxxx	LLLL = Assembly Lot Code
	S = Assembly Site Code
	# = DIE Revision

3 Detailed Bug Description

BCL12 *BCS Module*

Function Switching RSELx or modifying DCOCTL can cause DCO dead time or a complete DCO stop

Description After switching RSELx bits (located in register BCSTL1) from a value of >13 to a value of <12 OR from a value of <12 to a value of >13, the resulting clock delivered by the DCO can stop before the new clock frequency is applied. This dead time is approximately 20 us. In some instances, the DCO may completely stop, requiring a power cycle.

Furthermore, if all of the RSELx bits in the BCSTL1 register are set, modifying the DCOCTL register to change the DCOx or the MODx bits could also result in DCO dead time or DCO hang up.

Workaround - When switching RSEL from >13 to <12, use an intermediate frequency step. The intermediate RSEL value should be 13.

Current RSEL	Target RSEL	Recommended Transition Sequence
15	14	Switch directly to target RSEL
14 or 15	13	Switch directly to target RSEL
14 or 15	0 to 12	Switch to 13 first, and then to target RSEL (two step sequence)
0 to 13	0 to 12	Switch directly to target RSEL

AND

- When switching RSEL from <12 to >13 it's recommended to set RSEL to its default value first (RSEL = 7) before switching to the desired target frequency.

AND

- In case RSEL is at 15 (highest setting) it's recommended to set RSEL to its default value first (RSEL = 7) before accessing DCOCTL to modify the DCOx and MODx bits. After the DCOCTL register modification the RSEL bits can be manipulated in an additional step.

In the majority of cases switching directly to intermediate RSEL steps as described above will prevent the occurrence of BCL12. However, a more reliable method can be implemented by changing the RSEL bits step by step in order to guarantee safe function without any dead time of the DCO.

Note that the 3-step clock startup sequence consisting of clearing DCOCTL, loading the BCSTL1 target value, and finally loading the DCOCTL target value as suggested in the in the "TLV Structure" chapter of the [MSP430x2xx Family User's Guide](#) is not affected by BCL12 if (and only if) it is executed after a device reset (PUC) prior to any other modifications being made to BCSTL1 since in this case RSEL still is at its default value of 7. However any further changes to the DCOx and MODx bits will require the consideration of the workaround outlined above.

BCL14 *BCS Module*

Function Oscillator fault forced in bypass mode when P2SEL.7 bit is not set

Description When the LFXT1 oscillator is used in bypass mode and P2SEL.7 is not set, the oscillator fault flag (OFIFG) will be forced to set and cannot be cleared. Due to the failsafe logic, LFXT1 cannot be used as MCLK in this case. The bug only affects the behavior of the oscillator fault, the clocking itself works properly.

Workaround Set both P2SEL.6 and P2SEL.7 if the application requires correct function of the oscillator fault flag (e.g. MCLK failsafe logic).

NOTE: Setting P2SEL.7 bit disables the GPIO functionality and enables the input schmitt trigger of the pin. P2.7 should be tied to a fixed voltage level (VCC or GND) to prevent cross current.

CPU4

CPU Module

Function PUSH #4, PUSH #8CPU4 - Bug

Description The single operand instruction PUSH cannot use the internal constants (CG) 4 and 8. The other internal constants (0, 1, 2, -1) can be used. The number of clock cycles is different:

PUSH #CG uses address mode 00, requiring 3 cycles, 1 word instruction

PUSH #4/#8 uses address mode 11, requiring 5 cycles, 2 word instruction

Workaround Refer to the table below for compiler-specific fix implementation information.

IDE/Compiler	Version Number	Notes
IAR Embedded Workbench	IAR EW430 v2.x until v6.20	User is required to add the compiler flag option below. --hw_workaround=CPU4
IAR Embedded Workbench	IAR EW430 v6.20 or later	Workaround is automatically enabled
TI MSP430 Compiler Tools (Code Composer Studio)	v1.1 or later	
MSP430 GNU Compiler (MSP430-GCC)	MSP430-GCC 4.9 build 167 or later	

EEM20

EEM Module

Function Debugger might clear interrupt flags

Description During debugging read-sensitive interrupt flags might be cleared as soon as the debugger stops. This is valid in both single-stepping and free run modes.

Workaround None.

SYS15

SYS Module

Function LPM3 and LPM4 currents exceed specified limits

Description LPM3 and LPM4 currents may exceed specified limits if the SMCLK source is switched from DCO to VLO or LFXT1 just before the instruction to enter LPM3 or LPM4 mode.

Workaround After clock switching, a delay of at least four new clock cycles (VLO or LFXT1) must be implemented to complete the clock synchronization before going into LPM3 or LPM4.

TA12

TIMER_A Module

Function	Interrupt is lost (slow ACLK)
Description	Timer_A counter is running with slow clock (external TACLK or ACLK) compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by one with the occurring compare interrupt (if $TAR = CCRx$). Due to the fast MCLK the CCRx register increment ($CCRx = CCRx + 1$) happens before the Timer_A counter has incremented again. Therefore the next compare interrupt should happen at once with the next Timer_A counter increment (if $TAR = CCRx + 1$). This interrupt gets lost.
Workaround	Switch capture/compare mode to capture mode before the CCRx register increment. Switch back to compare mode afterwards.

TA16

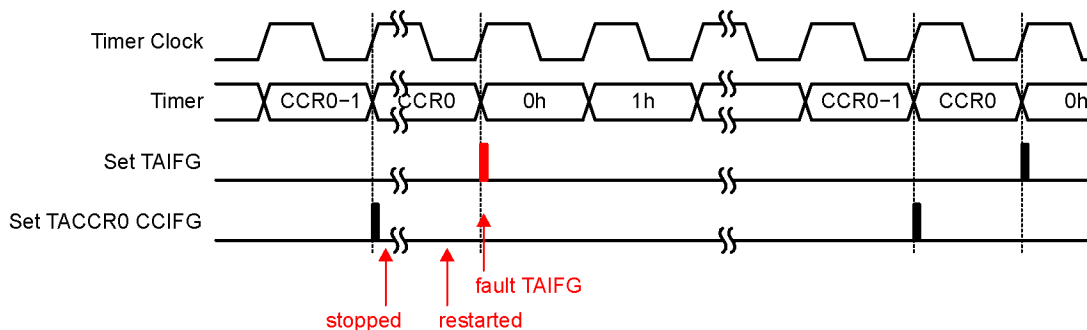
TIMER_A Module

Function	First increment of TAR erroneous when $IDx > 00$
Description	The first increment of TAR after any timer clear event (POR/TACLRL) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK or TACLK). This is independent of the clock input divider settings ($ID0$, $ID1$). All following TAR increments are performed correctly with the selected IDx settings.
Workaround	None

TA21

TIMER_A Module

Function	TAIFG Flag is erroneously set after Timer A restarts in Up Mode
Description	In Up Mode, the TAIFG flag should only be set when the timer counts from TACCR0 to zero. However, if the Timer A is stopped at $TAR = TACCR0$, then cleared ($TAR=0$) by setting the TACLRL bit, and finally restarted in Up Mode, the next rising edge of the TACLK will erroneously set the TAIFG flag.



Workaround	None.
-------------------	-------

TAB22

TIMER_A/TIMER_B Module

Function	Timer_A/Timer_B register modification after Watchdog Timer PUC
Description	Unwanted modification of the Timer_A/Timer_B registers TACTL/TBCTL and TAIV/TBIV can occur when a PUC is generated by the Watchdog Timer(WDT) in Watchdog mode and any Timer_A/Timer_B counter register TACCRx/TBCCRx is incremented/decremented (Timer_A/Timer_B does not need to be running).

Workaround	<p>Initialize TACTL/TBCTL register after the reset occurs using a MOV instruction (BIS/BIC may not fully initialize the register). TAIV/TBIV is automatically cleared following this initialization.</p> <p>Example code:</p> <pre>MOV.W #VAL, &TACTL</pre> <p>or</p> <pre>MOV.W #VAL, &TBCTL</pre> <p>Where, VAL=0, if Timer is not used in application otherwise, user defined per desired function.</p>
USI4	USI Module
Function	I2C Slave mode can generate a glitch at SCL
Description	USI I2C Slave Operation at slower communication rates (less than 20kbps). During I2C bus active operation, if USICNT is written while SCL is high, I2C module will generate a glitch on SCL that can corrupt the I2C bus sequence.
Workaround	Verify that SCL is low before writing USICNT register.
USI5	USI Module
Function	SPI master generates one additional clock after module reset
Description	Initializing the USI in SPI MASTER mode with the USICKPH bit set generates one additional clock pulse than defined by the value in the USICNTx bits on the SCLK pin during the first data transfer after module reset. For example, if the USICNTx bits hold the value eight, nine clock pulses are generated on the SCLK pin for the first transfer only.
Workaround	Load USICNTx with a count of N-1 bits (where N is the required number of bits) for the first transfer only.
XOSC5	XOSC Module
Function	LF crystal failures may not be properly detected by the oscillator fault circuitry
Description	The oscillator fault error detection of the LFXT1 oscillator in low frequency mode (XTS = 0) may not work reliably causing a failing crystal to go undetected by the CPU, i.e. OFIFG will not be set.
Workaround	None

4 Document Revision History

Changes from family erratasheet to device specific erratasheet.

1. Errata EEM20 was added
2. Errata TA22 was renamed to TAB22
3. Description for TAB22 was updated

Changes from device specific erratasheet to document Revision A.

1. USI5 Workaround was updated.

Changes from document Revision A to Revision B.

1. BCL12 Workaround was updated.

Changes from document Revision B to Revision C.

1. Errata TA21 was added to the errata documentation.

Changes from document Revision C to Revision D.

1. Silicon Revision B was added to the errata documentation.

Changes from document Revision D to Revision E.

1. BCL14 Workaround was updated.

Changes from document Revision E to Revision F.

1. Package Markings section was updated.

Changes from document Revision F to Revision G.

1. TA21 Description was updated.

Changes from document Revision G to Revision H.

1. Function for CPU4 was updated.
2. Workaround for CPU4 was updated.

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

Applications

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

TI E2E Community

e2e.ti.com