MSP430FG461x Device Erratasheet

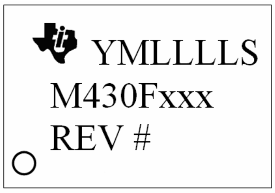
Current Version

Devices	Rev:	ADC18	CPU7	CPU8	DMA3	DMA4	FLL3	RTC1	TA12	TA16	TA18	TB2	TB16	TB18	USCI19	USCI20	WDG2
MSP430FG4616	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4617	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4618	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4619	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: See Appendix for prior revisions

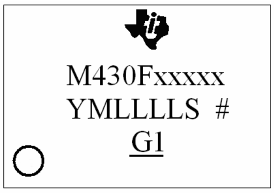
Package Markings

PZ100: LQFP(PZ) 100-pin



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

ZQW113: BGA(ZQW) 113-pin



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

Detailed Bug Description

ADC18 ADC18 - Bug description

Module: ADC12, incorrect conversion result in extended sample mode

The ADC12 conversion result can be incorrect in case where the extended sample mode is selected (SHP = 0), the conversion clock is not the internal ADC12 oscillator (ADC12SSEL > 0), and one of the following two conditions is true:

1) The extended sample input signal SHI is asynchronous to the clock source used for ADC12CLK and the undivided ADC12 input clock frequency exceeds 3.15 MHz
Or
2) The extended sample input signal SHI is synchronous to the clock source used for ADC12CLK and the undivided ADC12 input clock frequency exceeds 6.3 MHz.

Workaround:
1) Use the pulse sample mode (SHP = 1).
Or
2) Use the ADC12 internal oscillator as the ADC12 clock source.
Or
3) Limit the undivided ADC12 input clock frequency to 3.15 MHz.
Or
4) Use the same clock source (such as ACLK or SMCLK) to derive both SHI and ADC12CLK in order to achieve synchronous operation, and also limit the undivided ADC12 input clock frequency to 6.3 MHz.

CPU7 CPU7 - Bug description

Module: CPU, CPU CALL, and PUSH instructions

CALL and PUSH instructions using @SP+, @SP, and X(SP) addressing modes behave differently from original CPU. The original CPU predecremented the SP then used the predecremented value to calculate the address. The new CPU uses the SP to calculate the address, then decrements it.

Workaround: (Fixed in the compiler for C source. No silicon fix planned.)
None.

CPU8 CPU8 - Bug description

Module: CPU, using odd values in the SP register

The SP can be written with odd values. In the original CPU, an odd SP value could be combined with an odd offset (i.e., mov. #value, 5(SP)). In the new CPU, the SP can be written with an odd value, but the first time the SP is used, the LSB is forced to 0.

Workaround:
Do not use odd values with the SP.

Detailed Bug Description (continued)

DMA3 DMA3 - Bug description

Module: DMA, Function: Read-modify-write instructions may corrupt DMA address registers

When a 16-bit wide read-modify-write instruction (such as add.w and sub.w) is directly used on a DMA address register (DMAxSA or DMAxDA), the register contents will get corrupted.

Workaround:
1) Do not use 16-bit wide read-modify-write instructions on DMA address registers. Instead, in case address calculations are necessary, do the calculations first, and then assign the result to the DMA address registers.
Or
2) Use 20-bit wide read-modify-write instructions (such as addx.a, subx.a) on the DMA address registers if needed.

DMA4 DMA4 - Bug description

Module: DMA, Function: Corrupted write access to 20-bit DMA registers

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.

Workaround:
1) Ensure that no DMA access interrupts 20-bit wide accesses to the DMA address registers. This can be achieved by temporarely disabling all active DMA channels (DMAEN = 0), or by enabling the DMA on fetch feature (DMAONFETCH = 1). Also, it may be possible to ensure this by system design.
or
2) Use word access for accessing the DMA address registers. Note that this limits the address values that can be written to word values.

FLL3 FLL3 - Bug description

Module: FLL+, Function: FLLDx = 11 for /8 may generate an unstable MCLK frequency

When setting the FLL to higher frequencies using FLLDx = 11 (/8), the output frequency of the FLL may have a larger frequency variation (e.g., averaged over 2 seconds) and a lower average output frequency than expected when compared to the other FLLDx bit settings.

Workaround:
None

Detailed Bug Description (continued)

RTC1 RTC1 - Bug description

Module: RTC, Function: Incorrect RTCDAY count in BCD mode

When using the RTC in BCD mode, RTCDAY will count from 0x29 to 0x31 instead of counting to 0x30 in the month of December (RTCMON = 0x12). Furthermore, due to a malfunction in the leap year detection logic, RTCMON/RTCDAY may incorrectly count from 0x02/0x28 to 0x03/0x01 instead of 0x02/0x29, or it may incorrectly count from 0x02/0x28 to 0x02/0x29 instead of 0x03/0x01.

Workaround:
Do not operate the RTC module in BCD mode. Use the RTC in hexadecimal format mode (RTCBCD = 0) instead. Convert RTC registers to BCD on demand using software.
NOTE: The CPU instruction DADD.B/.W can be used to efficiently implement a hex to BCD conversion. An Assembly language example of such an optimized 8-bit conversion is shown below:

```
        mov.b      #8,R14       // Loop counter, process 8 bits
        clr.b      R12          // Result will get assembled in R12
loop    rlc.b      R13          // Get MSB from input variable in R13
        dadd.b     R12,R12
        dec.b      R14
        jnz        loop
```

TA12 TA12 - Bug description

Module: Timer_A, Function: Interrupt is lost (slow ACLK)

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 1 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 TA16 - Bug description

Module: Timer_A, Function: First increment of TAR erroneous when IDx > 00

The first increment of TAR after any timer clear event (POR/TACLR) 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

Detailed Bug Description (continued)

TA18 TA18 - Bug description

Module: Timer_A, MOV to TACTL may clear TAR

When TACTL is modified with a MOV instruction, the contents of TAR may be cleared, even when TACLR is not set.

Workaround:
Use BIS or BIC instructions to modify TACTL.
NOTE: A DMA transfer must not occur while these BIS and BIC instructions execute. This can be prevented by disabling the DMA prior to these instructions, or by using the DMAONFETCH bit to align DMA transfers to instruction fetch boundaries.

TB2 TB2 - Bug description

Module: Timer_B, Interrupt is lost (slow ACLK)

Timer_B counter is running with slow clock (external TBCLK or ACLK) compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by 1 with the occurring compare interrupt (if TBR = CCRx).
Due to the fast MCLK, the CCRx register increment (CCRx = CCRx + 1) happens before the Timer_B counter has incremented again. Therefore, the next compare interrupt should happen at once with the next Timer_B counter increment (if TBR = 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.

TB16 TB16 - Bug description

Module: Timer_B, Function: First increment of TBR erroneous when IDx > 00

The first increment of TBR after any timer clear event (POR/TBCLR) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK or TBCLK). This is independent of the clock input divider settings (ID0, ID1). All following TBR increments are performed correctly with the selected IDx settings.

Workaround:
None

TB18 TB18 - Bug description

Module: Timer_B, MOV to TBCTL may clear TBR

When TBCTL is modified with a MOV instruction, the contents of TBR may be cleared, even when TBCLR is not set.

Workaround:
Use BIS or BIC instructions to modify TBCTL.
NOTE: A DMA transfer must not occur while these BIS and BIC instructions execute. This can be prevented by disabling the DMA prior to these instructions, or by using the DMAONFETCH bit to align DMA transfers to instruction fetch boundaries.

Detailed Bug Description (continued)

USCI19 USCI19 - Bug description

Module: USCI, Function: LPM4 may affect USCI operation

When SMCLK is used as the USCI clock source, and SMCLK gets deactivated due to a LPM4 entry, ongoing SPI master, I2C master, and UART transmit transaction will be interrupted. Also, while in LPM4, UART receive operation is nonfunctional.

Workaround:
Do not enter LPM4 while SPI master, I2C master, or UART transmit operations are active. Wait for the operation to be completed prior entering LPM4, or enter a different low-power mode instead. Also, do not use LPM4 in case of UART receive operation. Instead, use a different low-power mode.

USCI20 USCI20 - Bug description

Module: USCI, Function: I2C Mode multi-master transmitter issue

When configured for I2C master-transmitter mode and used in a multi-master environment, the USCI module can cause unpredictable bus behavior if all of the following four conditions are true:

1) Two masters are generating SCL.
And
2) The slave is stretching the SCL low phase of an ACK period while outputting NACK on SDA.
And
3) The slave drives ACK on SDA after the USCI has already released SCL, and then the SCL bus line gets released.
And
4) The transmit buffer has not been loaded before the other master continues communication by driving SCL low.

The USCI will remain in the SCL high phase until the transmit buffer is written. After the transmit buffer has been written, the USCI will interfere with the current bus activity and may cause unpredictable bus behavior.

Workaround:
1) Ensure that slave does not stretch the SCL low phase of an ACK period.
Or
2) Ensure that the transmit buffer is loaded in time.
Or
3) Do not use the multi-master transmitter mode.

WDG2 WDG2 - Bug description

If a key violation is caused by incorrectly accessing a flash control register, the watchdog interrupt flag is set in addition to a correctly generated PUC.

Workaround:
None

Appendix: Prior Versions

Devices	Rev:	ADC18	CPU7	CPU8	DMA3	DMA4	FLL3	RTC1	TA12	TA16	TA18	TB2	TB16	TB18	USCI16	USCI19	USCI20	WDG2
MSP430FG4616	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4617	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4618	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430FG4619	G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
	E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Detailed Bug Description

USCI16 USCI16 - Bug description

Module: USCI, Function: UART/IrDA mode lost characters

When configured for UART/IrDA mode, the USCI baud rate generator may halt operation under the following conditions:
1) IrDA mode: repeated invalid start bits on the receive line
Or
2) UART/IrDA modes: positive pulse on the receive line during break character reception inside the stop bit time slot (the second stop bit time slot in case of UCSPB = 1) with a pulse width that passes the deglitch filter but is shorter than half a bit time.

After halting, additional characters will be ignored. Transmit functionality is not affected.

Workaround:
Check the UCBUSY flag status periodically in software. If the flag is set and no character has been received in the expected time, reset the USCI module in software. To reset the USCI module, toggle UCSWRST and reenable the USCI interrupts.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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