Stellaris® LM3S2678 RevA0 Errata

This document contains known errata at the time of publication for the Stellaris LM3S2678 microcontroller. The table below summarizes the errata and lists the affected revisions. See the data sheet for more details.

See also the ARM® Cortex™-M3 errata, ARM publication number PR326-PRDC-009450 v2.0.

Table 1. Revision History

Date	Revision	Description
September 2011	2.8	■ Added issue “Boundary scan is not functional” on page 4.
August 2011	2.7	■ Clarified issue “General-purpose timer 16-bit Edge Count or Edge Time mode does not load reload value” on page 10 to include Edge-Time mode. ■ Added issue “Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling” on page 13. ■ Added issue “PWM fault latch does not operate correctly” on page 16. ■ Minor edits and clarifications.
September 2010	2.6	■ Added issue “MOSC valid detect circuit should not be enabled” on page 5. ■ Removed the "ROM_I2CMasterErr function is incorrect" issue because the data sheet has been changed such that the ERROR bit no longer is set when the ARBLST bit is set. ■ Minor edits and clarifications.
July 2010	2.5	■ Added information that “No interrupt generated with software-triggered DMA transfer using channel 30” on page 6 is fixed for parts with certain date codes. ■ Added issue “PB0 and PB1 have permanent internal pull-down resistance” on page 8. ■ Added issue “The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled” on page 14. ■ Minor edits and clarifications.
June 2010	2.4	■ Added issue “External reset does not reset the XTAL to PLL Translation (PLLCFG) register” on page 5.
May 2010	2.3	■ Minor edits and clarifications.
April 2010	2.2	■ Minor edits and clarifications.
April 2010	2.1	■ Minor edits and clarifications.
February 2010	2.0	■ Added issue “The General-Purpose Timer match register does not function correctly in 32-bit mode” on page 11.

Date	Revision	Description
Jan 2010	1.9	■ "Hard Fault possible when waking from Sleep or Deep-Sleep modes and Cortex-M3 Debug Access Port (DAP) is enabled" has been removed and the content added to the LM3S2678 data sheet. ■ Added issue “The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules” on page 7. ■ Added issue “A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode” on page 11.
Dec 2009	1.8	Started tracking revision history.

Table 2. List of Errata

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
1.1	Subsequent attempts to mass erase the Flash memory following a locked device recovery will fail	JTAG and Serial Wire Debug	A0
1.2	JTAG INTEST instruction does not work	JTAG and Serial Wire Debug	A0
1.3	Boundary scan is not functional	JTAG and Serial Wire Debug	A0
2.1	Debugger cannot halt processor when in Sleep mode	System Control	A0
2.2	I/O buffer 5-V tolerance issue	System Control	A0
2.3	External reset does not reset the XTAL to PLL Translation (PLLCFG) register	System Control	A0
2.4	MOSC valid detect circuit should not be enabled	System Control	A0
3.1	FMPPE and FMPRE registers cannot be committed to non-volatile storage	Flash Controller	A0
4.1	ROM_SSIConfigSetExpClk function is incorrect	ROM	A0
5.1	No interrupt generated with software-triggered DMA transfer using channel 30	µDMA	A0
5.2	The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules	µDMA	A0
6.1	GPIO commit register control not working as expected	GPIO	A0
6.2	PB0 and PB1 have permanent internal pull-down resistance	GPIO	A0
7.1	General-purpose timer 16-bit Edge Count or Edge Time mode does not load reload value	General-Purpose Timers	A0
7.2	The General-Purpose Timer match register does not function correctly in 32-bit mode	General-Purpose Timers	A0
7.3	A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode	General-Purpose Timers	A0
8.1	Use of "Always" triggering for ADC Sample Sequencer 3 does not work	ADC	A0
8.2	Using the PWM unit to trigger the ADC results in only one sample taken	ADC	A0
8.3	TIMER3 cannot trigger ADC	ADC	A0

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
8.4	Incorrect behavior with timer ADC triggering when another timer is used in 32-bit mode	ADC	A0
8.5	ADC hardware averaging produces erroneous results in differential mode	ADC	A0
8.6	Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling	ADC	A0
9.1	The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled	UART	A0
10.1	PWM generation is incorrect with extreme duty cycles	PWM	A0
10.2	PWM sync status is not properly cleared	PWM	A0
10.3	Sync of PWM does not trigger "zero" action	PWM	A0
10.4	PWM "zero" action occurs when the PWM module is disabled	PWM	A0
10.5	PWM fault latch does not operate correctly	PWM	A0
11.1	QEI index resets position when index is disabled	QEI	A0
11.2	QEI hardware position can be wrong under certain conditions	QEI	A0
12.1	ROM-resident boot loader does not operate	Boot Loader	A0
13.1	Certain pins do not fully comply with the JEDEC ESD standard	Electrical Characteristics	A0

1 JTAG and Serial Wire Debug

1.1 Subsequent attempts to mass erase the Flash memory following a locked device recovery will fail

Description:

If a user performs a locked device recovery as described in the data sheet, subsequent attempts to mass erase the Flash memory will fail. The failure status in the **Flash Controller Raw Interrupt Status (FCRIS)** register following the attempted mass erase operation has the `Access Raw Interrupt Status (ARIS)` bit set, indicating an access error. The access error is generated because the recovery sequence incorrectly restores **Flash Memory Protection Enable 2 (FMPPE2)** and **FMPPE3** to the value 0xFFFF.FFFF. During subsequent mass erase attempts, the logic (that ensures write-protected Flash memory is not erased) incorrectly evaluates the protected condition and inhibits the erase operation.

Note: This erratum may affect the ROM-resident boot loader because this module utilizes Flash memory mass erase. The boot loader will not function following a recovery sequence because the boot loader attempts a mass erase, detects the error code, and terminates.

Workaround:

There is a dynamic workaround in which the **FMPPEn** registers are written to properly set the **FMPPEn** state before a mass erase is attempted. The **FMPPEn** registers must be written to the values indicated in the following table based on the size of Flash memory available on the microcontroller. The **FMPPEn** register must be written before each mass erase operation that is attempted.

Table 3. FMPPEn Default Values Based on Flash Memory Size

Flash Size (KB)	FMPPE3	FMPPE2	FMPPE1	FMPPE0
128	0x0000.0000	0x0000.0000	0xffff.ffff	0xffff.ffff
96	0x0000.0000	0x0000.0000	0x0000.ffff	0xffff.ffff
64	0x0000.0000	0x0000.0000	0x0000.0000	0xffff.ffff
32	0x0000.0000	0x0000.0000	0x0000.0000	0x0000.ffff

Silicon Revision Affected:

A0

1.2 JTAG INTEST instruction does not work

Description:

The JTAG INTEST (Boundary Scan) instruction does not properly capture data.

Workaround:

None.

Silicon Revision Affected:

A0

1.3 Boundary scan is not functional

Description:

The boundary scan is not functional on this device.

Workaround:

None.

Silicon Revision Affected:

A0

Fixed:

Fixed on devices with date codes of 0x1A (October, 2011) or later.

2 System Control

2.1 Debugger cannot halt processor when in Sleep mode

Description:

When the processor is in Sleep mode, the debugger is unable to bring the processor out of Sleep mode in order to initiate a debug session.

Workaround:

Do not use Sleep mode when attempting to debug the application.

Silicon Revision Affected:

A0

2.2 I/O buffer 5-V tolerance issue

Description:

GPIO buffers are not 5-V tolerant when used in open-drain mode. Pulling up the open-drain pin above 4 V results in high current draw.

Workaround:

When configuring a pin as open drain, limit any pull-up resistor connections to the 3.3-V power rail.

Silicon Revision Affected:

A0

2.3 External reset does not reset the XTAL to PLL Translation (PLLCFG) register

Description:

Performing an external reset (anything but power-on reset) reconfigures the XTAL field in the **Run-Mode Clock Configuration (RCC)** register to the 6 MHz setting, but does not reset the **XTAL to PLL Translation (PLLCFG)** register to the 6 MHz setting.

Consider the following sequence:

1. Performing a power-on reset results in XTAL = 6 MHz and **PLLCFG** = 6 MHz
2. Write an 8 MHz value to the XTAL field results in XTAL = 8 MHz and **PLLCFG** = 8 MHz
3. $\overline{\text{RST}}$ asserted results in XTAL = 6 MHz and **PLLCFG** = 8 MHz

In the last step, **PLLCFG** was not reset to its 6MHz setting. If this step is followed by enabling the PLL to run from an attached 6-MHz crystal, the PLL then operates at 300MHz instead of 400MHz. Subsequently configuring the XTAL field with the 8 MHz setting does not change the setting of **PLLCFG**.

Workaround:

Set XTAL in **PLLCFG** to an incorrect value, and then to the desired value. The second change updates the register correctly. Do not enable the PLL until after the second change.

Silicon Revision Affected:

A0

2.4 MOSC valid detect circuit should not be enabled

Description:

If bit 0 of the **Main Oscillator Control (MOSCCTL)** register is set, the microcontroller is immediately reset and control is transferred to the NMI handler, even if the MOSC is functioning correctly.

Workaround:

None.

Silicon Revision Affected:

A0

3 Flash Controller

3.1 FMPPE and FMPRE registers cannot be committed to non-volatile storage

Description:

The Flash memory protection provided by the **Flash Memory Protection Read Enable n (FMPREn)** and **Flash Memory Protection Program Enable n (FMPPEn)** registers does not function correctly. Do not commit any of these registers or mass erase will not function.

This does not affect the Flash memory protection provided by the disabling of the JTAG/SWD port.

Workaround:

None.

Silicon Revision Affected:

A0

4 ROM

4.1 ROM_SSIConfigSetExpClk function is incorrect

Description:

If a non-Motorola format was specified in a call to the ROM_SSIConfigSetExpClk function, two lower bits of a clock divisor register could be corrupted. This corruption results in a small error in the actual clock rate.

Workaround:

Use the StellarisWare SSIConfigSetExpClk function in Flash memory.

Silicon Revision Affected:

A0

5 µDMA

5.1 No interrupt generated with software-triggered DMA transfer using channel 30

Description:

When performing a software-triggered data transfer using µDMA channel 30, no interrupt is triggered when the transfer completes.

Workaround:

Use any of the other channels (except for channel 30 and 31) that are not already assigned to a peripheral. See the "DMA Channel Assignments" table in the *Micro Direct Memory Access (µDMA)* chapter in the data sheet.

Silicon Revision Affected:

A0

Fixed:

This issue is fixed on parts with a date code of 06 (June 2010) or later. To determine the date code of your part, look at the third line in the part markings, at the fourth and fifth characters following the dash (highlighted in red below). The first number after the dash indicates the last decimal digit of the year. The next character indicates the month, in hexadecimal. So, in the below example, the 9B indicates a date code of November 2009.

The table below shows some example date codes:

Date Code	Date
9B	November 2009
9C	December 2009
01	January 2010
02	February 2010
03	March 2010
04	April 2010
05	May 2010
06	June 2010
07	July 2010
08	August 2010
09	September 2010
0A	October 2010
0B	November 2010
0C	December 2010

5.2 The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules

Description:

The µDMA controller fails to generate DMA requests from Timer A in the General-Purpose Timer modules when in the Event Count and Event Time modes.

Workaround:

Use Timer B.

Silicon Revision Affected:

A0

6 GPIO

6.1 GPIO commit register control not working as expected

Description:

The commit control registers provide a layer of protection against accidental programming of critical hardware peripherals, especially the JTAG/SWD pin functionality. Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)**, **GPIO Digital Enable (GPIODEN)**, and **GPIO Pull-up Select (GPIOPUR)** registers are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register has been unlocked and the appropriate bits of the **GPIO Interrupt Clear (GPIOCR)** register have been set. Registers **GPIODEN** and **GPIOPUR** should be protected by this mechanism but are not.

Workaround:

Extra caution should be taken by software when modifying the **GPIODEN** and **GPIOPUR** registers as they are not protected by the **GPIOCR** commit register mechanism. Writes to the register bits that affect the JTAG/SWD pins (`PC0-PC3`) should be done with great care as this interface may become permanently disabled if done incorrectly. The `NMI` pin (`PB7`) is not protected either.

Silicon Revision Affected:

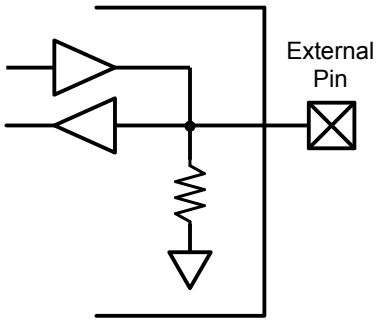
A0

6.2 PB0 and PB1 have permanent internal pull-down resistance

Description:

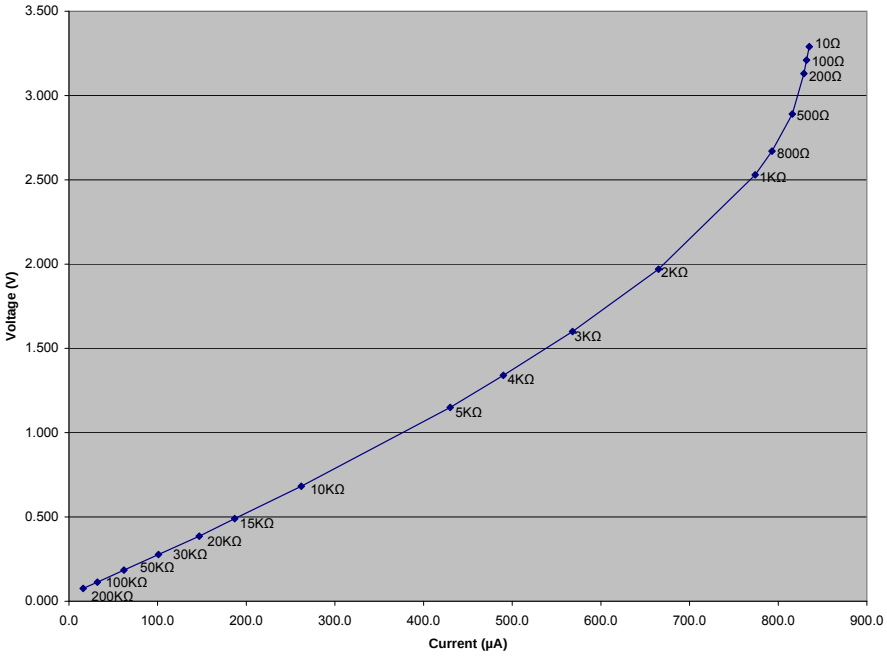
Regardless of their configuration, PB0 and PB1 have an internal pull-down resistance. The internal structure of these pins is shown in Figure 1 on page 8.

Figure 1. Internal Structure of PB0 and PB1



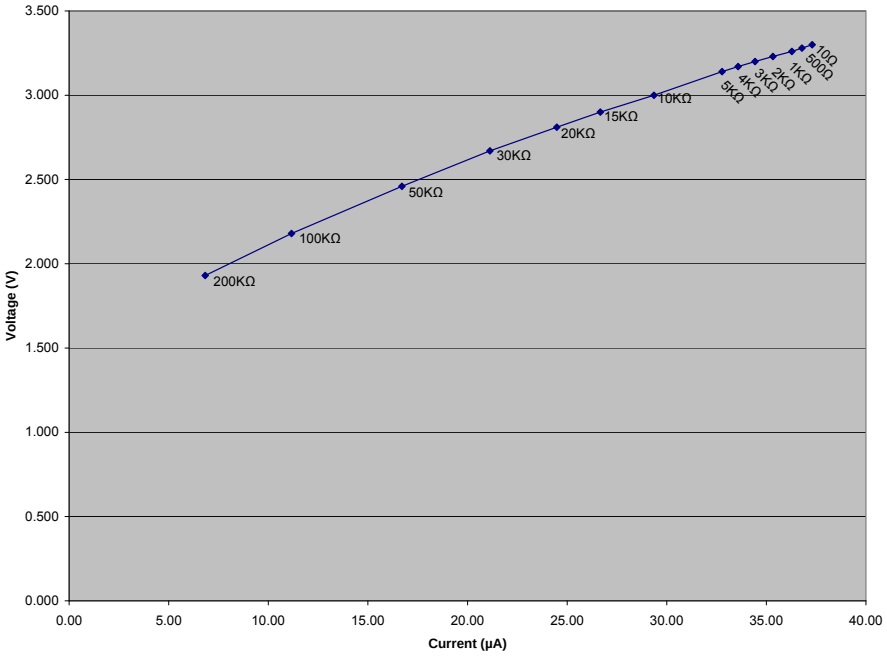
The characteristic of the pull-down for PB0 is shown in Figure 2 on page 9. The data labels for each data point show the external pull-up resistance in Ohms.

Figure 2. Voltage vs. Current for Various External Pull-up Resistors on PB0



The characteristic of the pull-down for PB1 is shown in Figure 3 on page 10. The data labels for each data point show the equivalent resistance in Ohms.

Figure 3. Voltage vs. Current for Various External Pull-up Resistors on PB1



Workaround:

When either of these pins is configured as an input, the external circuit must provide enough drive strength to over-drive the internal pull-down and achieve the necessary V_{IH} voltage level. If an external pull-up resistor is used, it must have a low value such as ~1 KΩ or less for PB0 and ~50 KΩ or less for PB1.

When either of these pins is configured as an output, the drive current needed to over-drive the internal pull-down resistance must be subtracted from the drive capabilities of the pin. In some applications, it may be necessary to select a higher drive strength (such as 4 mA instead of 2 mA) to achieve an acceptable output voltage on PB0.

Silicon Revision Affected:

A0

7 General-Purpose Timers

7.1 General-purpose timer 16-bit Edge Count or Edge Time mode does not load reload value

Description:

In Edge Count or Edge Time mode, the input events on the CCP pin decrement the counter until the count matches what is in the **GPTM Timern Match (GPTMTnMATCHR)** register. At that point, an interrupt is asserted and then the counter should be reloaded with the original value and counting begins again. However, the reload value is not reloaded into the timer.

Workaround:

Rewrite the **GPTM Timern Interval Load (GPTMTnILR)** register before restarting.

Silicon Revision Affected:

A0

7.2 The General-Purpose Timer match register does not function correctly in 32-bit mode

Description:

The **GPTM Timer A Match (GPTMTAMATCHR)** register triggers a match interrupt and a DMA request, if enabled, when the lower 16 bits match, regardless of the value of the upper 16 bits.

Workaround:

None.

Silicon Revision Affected:

A0

7.3 A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode

Description:

When the timer is in Input-Edge Time mode and rolls over after the terminal count, a spurious DMA request is generated.

Workaround:

Either ignore the spurious interrupt, or disable DMA requests shortly before the terminal count and enable them again shortly after the terminal count.

Silicon Revision Affected:

A0

8 ADC

8.1 Use of "Always" triggering for ADC Sample Sequencer 3 does not work

Description:

When using ADC Sample Sequencer 3 (SS3) and configuring the trigger source to "Always" to enable continuous sampling by programming the SS3 Trigger Select field (EM3) in the **ADC Event Multiplexer Select (ADCEMUX)** register to 0xF, the first sample will be captured, but no further samples will be updated to the sequencer FIFO. Interrupts are continuously generated after the first sample and the FIFO status remains empty.

Workaround:

Software must disable and re-enable the sample sequencer to capture another sample.

Silicon Revision Affected:

A0

8.2 Using the PWM unit to trigger the ADC results in only one sample taken

Description:

If the ADC is configured to trigger from a PWM generator source, the ADC unit is triggered on the first matched event in the **PWMn Interrupt and Trigger Enable (PWMnINTEN)** register; however, it will receive no further triggers at the subsequent event matches.

Workaround:

If the raw interrupt status is cleared before the next event match, the ADC correctly receives the next event signal. Therefore, the trigger condition must be cleared using the same interrupt condition in the **PWMn Interrupt Status and Clear (PWMnISC)** register.

Trigger condition	Interrupt status and clear
TrCntZero	IntCntZero
TrCntLoad	IntCntLoad
TrCmpAU	IntCmpAU
TrCmpAD	IntCmpAD
TrCmpBU	IntCmpBU
TrCmpBD	IntCmpBD

For example, consider the scenario where the ADC performs sequence 0 and generates a processor interrupt when triggered by PWM generator 2 CMPAD (compararator A value count down match). The DriverLib call to enable the PWM to trigger the ADC is:

```
PWMGenIntTrigEnable(PWM_BASE, PWM_GEN_2, PWM_TR_CNT_AD);
```

In the ADC sequence 0 interrupt handler, both the ADC interrupt status bit and the PWM generator 2 status bits must be cleared, as follows:

```
void
ADC0IntHandler(void)
{
     //
     // Clear the PWM trigger source
     // This step is for an errata workaround
     //
     PWMGenIntClear(PWM_BASE, PWM_GEN_2, PWM_INT_CNT_AD);
     //
     // Clear the ADC interrupt status
     //
     ADCIntClear(ADC_BASE, 0);

     ...remaining ADC interrupt handler code...
```

Silicon Revision Affected:

A0

8.3 TIMER3 cannot trigger ADC

Description:

The **ADC Event Multiplexer Select (ADCEMUX)** register can be used to select the event (trigger) that initiates sampling for each sample sequencer. However, Timer 3 cannot be used to trigger analog-to-digital conversions.

Workaround:

Use one of the other timers to trigger ADC.

Silicon Revision Affected:

A0

8.4 Incorrect behavior with timer ADC triggering when another timer is used in 32-bit mode

Description:

When a timer is configured to trigger the ADC and another timer is configured to be a 32-bit periodic or one-shot timer, the ADC is triggered continuously instead of the specified interval.

Workaround:

Do not use a 32-bit periodic or one-shot timer when triggering ADC. If the timer is in 16-bit mode, the ADC trigger works as expected.

Silicon Revision Affected:

A0

8.5 ADC hardware averaging produces erroneous results in differential mode

Description:

The implementation of the ADC averaging circuit does not work correctly when the ADC is sampling in differential mode and the difference between the voltages is approximately 0.0V.

Workaround:

Do not use hardware averaging in differential mode. Instead, use the FIFO to store results and average them in software.

Silicon Revision Affected:

A0

8.6 Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling

Description:

Re-triggering a sample sequencer before it has completed its programmed conversion sequence causes the sample sequencer to continuously sample. If interrupts have been enabled, interrupts

are generated at the appropriate place in the sample sequence. This problem only occurs when the new trigger is the same type as the current trigger.

Workaround:

Ensure that a sample sequence has completed before triggering a new sequence using the same type of trigger.

Silicon Revision Affected:

A0

9 UART

9.1 The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled

Description:

The RTRIS (UART Receive Time-Out Raw Interrupt Status) bit in the **UART Raw Interrupt Status (UARTRIS)** register should be set when a receive time-out occurs, regardless of the state of the enable RTIM bit in the **UART Interrupt Mask (UARTIM)** register. However, currently the RTIM bit must be set in order for the RTRIS bit to be set when a receive time-out occurs.

Workaround:

For applications that require polled operation, the RTIM bit can be set while the UART interrupt is disabled in the NVIC using the IntDisable(n) function in the StellarisWare Peripheral Driver Library, where n is 21, 22, or 49 depending whether UART0, UART1 or UART2 is used. With this configuration, software can poll the RTRIS bit, but the interrupt is not reported to the NVIC.

Silicon Revision Affected:

A0

10 PWM

10.1 PWM generation is incorrect with extreme duty cycles

Description:

If a PWM generator is configured for Count-Up/Down mode, and the **PWM Load (PWMnLOAD)** register is set to a value N, setting the compare to a value of 1 or N-1 results in steady state signals instead of a PWM signal. For example, if the user configures PWM0 as follows:

- PWMENABLE = 0x00000001
  - PWM0 Enabled
- PWM0CTL = 0x00000007
  - Debug mode enabled
  - Count-Up/Down mode
  - Generator enabled

- PWM0LOAD = 0x00000063
  - Load is 99 (decimal), so in Count-Up/Down mode the counter counts from zero to 99 and back down to zero (200 clocks per period)
- PWM0GENA = 0x000000b0
  - Output High when the counter matches comparator A while counting up
  - Output Low when the counter matches comparator A while counting down
- PWM0DBCTL = 0x00000000
  - Dead-band generator is disabled

If the **PWM0 Compare A (PWM0CMPA)** value is set to 0x00000062 (N-1), PWM0 should output a 2-clock-cycle long High pulse. Instead, the PWM0 output is a constant High value.

If the **PWM0CMPA** value is set to 0x00000001, PWM0 should output a 2-clock-cycle long negative (Low) pulse. Instead, the PWM0 output is a constant Low value.

Workaround:

User software must ensure that when using the PWM Count-Up/Down mode, the compare values must never be 1 or the **PWMnLOAD** value minus one (N-1).

Silicon Revision Affected:

A0

10.2 PWM sync status is not properly cleared

Description:

When writing to the **PWM Time Base Sync (PWMSYNC)** register to sync the PWM generators, the sync bits are not automatically cleared by hardware. This condition results in the PWM counters being held and prevents them from outputting a PWM signal. The issue occurs only when the PWM clock divider is enabled in the **Run-Mode Clock Configuration (RCC)** register. When the USEPWMDIV bit is set, any divider (PWMDIV) exhibits the same behavior.

Workaround:

A software write to the **PWMSYNC** register to manually clear the sync bits releases the PWM counters and they will generate expected output waveforms.

Silicon Revision Affected:

A0

10.3 Sync of PWM does not trigger "zero" action

Description:

If the **PWM Generator Control (PWM0GENA)** register has the ActZero field set to 0x2, then the output is set to 0 when the counter reaches 0, as expected. However, if the counter is cleared by setting the appropriate bit in the **PWM Time Base Sync (PWMSYNC)** register, then the "zero" action is not triggered, and the output is not set to 0.

Workaround:

None.

Silicon Revision Affected:

A0

10.4 PWM "zero" action occurs when the PWM module is disabled

Description:

The zero pulse may be asserted when the PWM module is disabled.

Workaround:

None.

Silicon Revision Affected:

A0

10.5 PWM fault latch does not operate correctly

Description:

If the LATCH bit is set in th **PWMnCTL** register, the PWM fault condition should be latched until the INTFAULTn bit in the **PWMISC** register is cleared. However, the PWM fault signal is not correctly latched and the PWM resumes programmed signalling after the fault condition is removed, regardless of whether the INTFAULTn bit is cleared.

Workaround:

Software can effectively address this issue with the addition of a few register writes in the ISR.

1. The **PWMnMINFLTPER** register can be used to ensure that the fault is asserted for a long enough period such that the ISR can be called to implement the workaround.
2. The PWM output can be disabled manually using the PWMnEN bit in the **PWMENABLE** register.
3. Software can perform computations to determine if the PWM can be restarted.
4. The INTFAULTn bit in the **PWMISC** is cleared by writing a 1 to it.
5. The PWM output can be manually re-enabled using the PWMnEN bit in the **PWMENABLE** register.

Note that when using this workaround, the PWM output is disabled manually, which means it does not go to the "pre-programmed" state from various fault registers but instead goes to 0.

Silicon Revision Affected:

A0

11 QEI

11.1 QEI index resets position when index is disabled

Description:

When the QEI module is configured to not reset the position on detection of the index signal (that is, the ResMode bit in the **QEI Control (QEICTL)** register is 0), the module resets the position when the index pulse occurs. The position counter should only be reset when it reaches the maximum value set in the **QEI Maximum Position (QEIMAXPOS)** register.

Workaround:

Do not rely on software to disable the index pulse. Do not connect the index pulse if it is not needed.

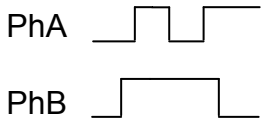
Silicon Revision Affected:

A0

11.2 QEI hardware position can be wrong under certain conditions

Description:

The **QEI Position (QEIPOS)** register can be incorrect if the QEI is configured for quadrature phase mode (SigMode bit in **QEICTL** register = 0) and to update the position counter of every edge of both PhA and PhB (CapMode bit in **QEICTL** register = 1). This error can occur if the encoder is stepped in the reverse direction, stepped forward once, and then continues in the reverse direction. The following sequence of transitions on the PhA and PhB pins causes the error:



Assuming the starting position prior to the above PhA and PhB sequence is 0, the position after the falling edge on PhB should be -3, however the **QEIPOS** register will show the position to be -1.

Workaround:

Configure the QEI to update the position counter on every edge on PhA only (CapMode bit in **QEICTL** register = 0). The effective resolution is reduced by 50%. If full resolution position detection is required by updating the position counter on every edge of both PhA and PhB, no workaround is available. Hardware and software must take this into account.

Silicon Revision Affected:

A0

12 Boot Loader

12.1 ROM-resident boot loader does not operate

Description:

The ROM-resident boot loader cannot be used to program the on-chip Flash. The boot loader enters the hard-fault state as a result of trying to access a peripheral that does not exist.

Workaround:

Use the Stellaris Flash-resident boot loader to update the on-chip Flash.

Silicon Revision Affected:

A0

13 Electrical Characteristics

13.1 Certain pins do not fully comply with the JEDEC ESD standard

Description:

The pins listed below do not fully meet the industry ESD standard of 2.0 KV Human Body Model (HBM) under all conditions. All Stellaris devices are tested using JEDEC Standard JESD22-A114. These pins fail only in the pin-to-power condition using a positive voltage transient. The pins pass at 500 V. All other HBM conditions pass on these pins. All other device pins fully comply with this HBM JEDEC standard.

- PB0
- PB1
- PF0
- OSC0
- OSC1

Workaround:

Extra caution should be taken to ensure proper ESD handling of these devices. Use appropriate caution when implementing ESD protection circuits on signals routed to these pins.

Silicon Revision Affected:

A0

Texas Instruments Incorporated
108 Wild Basin, Suite 350
Austin, TX 78746
http://www.ti.com/stellaris
http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm

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**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

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