Stellaris® LM3S811 RevC2 Errata

This document contains known errata at the time of publication for the Stellaris® LM3S811 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 PR141-PRDC-007452 v9.0.

Date	Revision	Description
July 2010	2.4	■ Added issue “The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled” on page 11.
April 2010	2.3	■ Removed issue "Setting Bit 7 in I2C Master Timer Period (I2CMTPR) register may have unexpected results". The data sheet description has changed such that this is no longer necessary. ■ Minor edits and clarifications.
February 2010	2.2	■ Added issue “The General-Purpose Timer match register does not function correctly in 32-bit mode” on page 9. ■ Added issue "Setting Bit 7 in I2C Master Timer Period (I2CMTPR) register may have unexpected results".
Jan 2010	2.1	■ "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 LM3S811 data sheet.
Dec 2009	2.0	Started tracking revision history.

Erratum Number	Erratum Title	Revision(s) Affected
1.1	JTAG interface is held in reset if TRST is configured as a GPIO	C0
1.2	Serial Wire Debug (SWD) interface is held in reset if TRST is configured as a GPIO	C0
2.1	JTAG INTEST instruction does not work	C0, C2
3.1	MOSC verification circuit does not detect all failures of the main oscillator	C0, C2
3.2	Excessive input pin current when level exceeds V_{DD}	C0, C2
3.3	LDO Current Limit interrupt does not function properly	C0, C2
3.4	LDO Power Unregulated interrupt unpredictable after LDO voltage adjustments	C0, C2
3.5	PLL Lock Raw Interrupt Status triggers when PLL is powered down	C0, C2
3.6	Brown-Out-Reset (BOR) state unknown after initial brown out condition	C0
3.7	Brown-Out-Reset (BOR) does not work when the PLL is enabled to drive the system clock	C0
3.8	Software Reset does not work when the PLL is enabled to drive the system clock	C0
3.9	Device is non-functional when V_{DD} drops below LDO voltage setting +500 mV	C0
3.10	I/O buffer 5-V tolerance issue	C0, C2

Erratum Number	Erratum Title	Revision(s) Affected
4.1	GPIO internal pull-up resistor does not pull up to 3.3 V	C0
4.2	PB6, PC5, and PC6 are not 5-V tolerant	C0
4.3	GPIO input pin latches in the Low state if pad type is open drain	C0, C2
4.4	GPIO pins may glitch during power supply ramp up	C0, C2
5.1	General-purpose timer Edge Count mode count error when timer is disabled	C0, C2
5.2	General-purpose timer 16-bit Edge Count mode does not load reload value	C0, C2
6.1	The General-Purpose Timer match register does not function correctly in 32-bit mode	C0, C2
7.1	Watchdog Timer reset does not work when the PLL is enabled to drive the system clock	C0
8.1	Use of "Always" triggering for ADC Sample Sequencer 3 does not work	C0, C2
8.2	Incorrect behavior with timer ADC triggering when another timer is used in 32-bit mode	C0, C2
8.3	ADC hardware averaging produces erroneous results in differential mode	C0, C2
9.1	The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled	C0, C2
10.1	PWM pulses cannot be smaller than dead-band time	C0, C2
10.2	PWM interrupt clear misses in some instances	C0, C2
10.3	PWM generation is incorrect with extreme duty cycles	C0, C2
10.4	PWMINTEN register bit does not function correctly	C0, C2
10.5	Sync of PWM does not trigger "zero" action	C0, C2
10.6	PWM "zero" action occurs when the PWM module is disabled	C0, C2

1 JTAG and Serial Wire Debug

1.1 JTAG interface is held in reset if TRST is configured as a GPIO

Description:

The JTAG interface consists of five pins: TDI, TDO, TCLK, TMS, and TRST. TRST is an optional JTAG signal and thus should be available to users as a GPIO. However, if the PB7/TRST pin is configured as the GPIO PB7 function, the JTAG module is incorrectly held in a reset state. In this 4-pin configuration, users will not be able to communicate with the JTAG interface.

Workaround:

The PB7/TRST pin must be configured as JTAG TRST in order for the JTAG interface to function correctly.

Silicon Revision Affected:

C0

1.2 Serial Wire Debug (SWD) interface is held in reset if TRST is configured as a GPIO

Description:

The SWD interface consists of three pins: SWDIO, SWCLK, and SWO. Even though the TRST signal is not a pin used in the SWD interface, if the PB7/TRST pin is configured as the GPIO PB7 function, the SWD interface is held in a reset state. In this configuration, users will not be able to communicate over the SWD interface.

Workaround:

The PB7/TRST pin must be configured as JTAG TRST in order for the SWD interface to function correctly.

Silicon Revision Affected:

C0

2 JTAG

2.1 JTAG INTEST instruction does not work

Description:

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

Workaround:

None.

Silicon Revision Affected:

C0, C2

3 System Control

3.1 MOSC verification circuit does not detect all failures of the main oscillator

Description:

The MOSC verification circuit does not detect all MOSC (main oscillator) failures. In the case where the MOSC clock verification timer function has been enabled by the MOSCVER bit in the **Run-Mode Clock Configuration (RCC)** register, and a MOSC failure is detected, the main clock tree is supposed to immediately switch to a working clock and generate an interrupt to the core. The MOSC verification circuit does not always detect the failure, so the device continues to be clocked by the failed MOSC clock source.

Workaround:

An external clock verification circuit must be used to guarantee the detection of a main oscillator failure. For example, a general-purpose timer can be used to generate a periodic signal to an external circuit that monitors the oscillation of that signal. If the external circuit detects that the signal has stopped oscillating, the circuit can assert the hardware reset pin of the controller.

Silicon Revision Affected:

C0, C2

3.2 Excessive input pin current when level exceeds V_{DD}

Description:

When the voltage on an input pin is driven above V_{DD}, excessive current is sunk through the input pin. For example, if a pin is configured as a GPIO input and pulled-up to 5 volts with a 1K W resistor, the V_{IH} level is only 4.4 volts, meaning I_{IH} is 0.64 mA. This violates the 5-V tolerance specification.

Workaround:

If the device driving the input pin above V_{DD} cannot source enough current to drive the signal to a logic High value, a resistor can be used in series to limit the amount of current being sunk through the pin.

Silicon Revision Affected:

C0, C2

3.3 LDO Current Limit interrupt does not function properly

Description:

The System Control module can be programmed by software to generate an interrupt when the LDO exceeds its current limit. This feature does not function properly in all over-current conditions.

Workaround:

The Current Limit interrupt should not be used. The over-current condition should be ignored by the interrupt controller by always masking the Current Limit interrupt. The `CLIM` bit in the **Interrupt Mask Control (IMC)** register should always be cleared (0).

Silicon Revision Affected:

C0, C2

3.4 LDO Power Unregulated interrupt unpredictable after LDO voltage adjustments

Description:

The System Control module can be programmed by software to generate an interrupt when the LDO power is unregulated. The interrupt is unpredictable while the LDO voltage is being adjusted. This could incorrectly cause an interrupt or, if the `LDOARST` bit in the **Allow Unregulated LDO to Reset the Part (LDOARST)** register is set, a system reset.

Workaround:

The Power Unregulated interrupt should not be used while the LDO voltage is adjusted. While the LDO voltage is adjusted, the unregulated power condition should be ignored by the interrupt controller by always masking the Power Unregulated interrupt. This condition is masked by clearing the `LDOIM` bit in the **Interrupt Mask Control (IMC)** register.

Silicon Revision Affected:

C0, C2

3.5 PLL Lock Raw Interrupt Status triggers when PLL is powered down

Description:

The **PLL Lock Raw Interrupt Status (PLLLRIS)** register incorrectly triggers when the PLL is powered down. After reset, by default, the PWRDN bit in the **Run-Mode Clock Configuration (RCC)** register is set, powering down the PLL. This interrupt will not be promoted to the interrupt controller unless the PLL Lock Interrupt Mask (PLLLIM) bit is set in the **Interrupt Mask Control (IMC)** register.

Workaround:

If the PLL is not being used, mask the PLL Lock interrupt by clearing the PLLIM bit in the **IMC** register. Only unmask the PLLRIS bit after the PLL has been powered on (PWRDN = 0).

Silicon Revision Affected:

C0, C2

3.6 Brown-Out-Reset (BOR) state unknown after initial brown out condition

Description:

Once a BOR condition is detected, the device has no way of knowing whether the condition has been rectified or if it still exists. After a BOR condition occurs, either an interrupt or reset is triggered, but there are no status flags for software to determine if the BOR condition still exists. This can result in the device executing the main application code while the supply voltage is below the BOR threshold voltage.

Workaround:

No workaround is available.

Silicon Revision Affected:

C0

3.7 Brown-Out-Reset (BOR) does not work when the PLL is enabled to drive the system clock

Description:

The internal brown-out detector does not properly reset the device when the PLL is enabled to drive the system clock while a brown-out condition exists. The brown-out detector triggers an internal reset, however, the device does not complete its reset routine, and the device does not come out of reset. The only way to recover from this condition is to assert the external hardware reset ($\overline{\text{RST}}$) or perform a Power-On Reset (POR).

Workaround:

If the PLL is used to drive the system clock, the BOR detector should not be used to generate a reset if a brown-out condition exists. The BOR detector only resets the device properly when the system clock is not driven from the PLL. The BYPASS bit in the **Run-Mode Clock Configuration (RCC)** register must be set to bypass the PLL and clock the system directly from the OSC source.

Silicon Revision Affected:

C0

3.8 Software Reset does not work when the PLL is enabled to drive the system clock

Description:

Software can initiate a reset sequence by setting the SYSRESETREQ bit in the ARM Cortex-M3 Application Interrupt and Reset Control register. When the PLL is enabled to drive the system clock and the SYSRESETREQ bit is set, the device does not properly complete the reset routine and the device does not come out of reset. The only way to recover from this condition is to assert the external hardware reset ($\overline{\text{RST}}$) or to perform a Power-On Reset (POR).

Workaround:

Because software sets the SYSRESETREQ bit, the user code can simply bypass the PLL before the SYSRESETREQ bit is set, allowing the device to reset properly. The BYPASS bit in the **Run-Mode Clock Configuration (RCC)** register can be set to bypass the PLL and clock the system directly from the OSC source.

Silicon Revision Affected:

C0

3.9 Device is non-functional when V_{DD} drops below LDO voltage setting +500 mV

Description:

The LDO voltage is adjustable between 2.25 V and 2.75 V and is set to a default value of 2.5 V after reset. The device becomes non-functional when the V_{DD} supply to the device drops below the LDO voltage plus 500 mV. For example, if LDO is set to its default value and the V_{DD} supply is dropped below 3.0 V, the device becomes non-functional. If the LDO is set to 2.75 V, then the device becomes non-functional when V_{DD} is dropped below 3.25 V. Because the Brown-Out Threshold (V_{BTH}) is a minimum of 2.85 V (below where the device becomes non-functional with the default LDO setting of 2.5 V), the internal Brown-Out Reset (BOR) circuitry is not able to reset the device before the device becomes nonfunctional.

Workaround:

It is recommended that the Brown-Out Reset (BOR) detector not be used and the internal LDO voltage regulator be kept at the default setting of 2.5 V, allowing the device to operate across the complete V_{DD} operating range (3.0 V to 3.6 V).

Silicon Revision Affected:

C0

3.10 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:

C0, C2

4 GPIO

4.1 GPIO internal pull-up resistor does not pull up to 3.3 V

Description:

Resetting a Stellaris® microcontroller causes configurable I/O signals, with the exception of the JTAG port, to be set to the default mode of a GPIO input with a weak internal pull-up resistor. The intent of this weak pull-up (~200k ohm) is to ensure that configurable signals are quiescent during the period between a reset operation and configuration of the microcontroller by the user. In current Stellaris® devices, these pull-up resistors are set too high (~2M ohm) and do not guarantee that the inputs are biased to a logic 1 as intended. As a result, it is possible that a non-driven input pin could float to a mid-level voltage between a logic 0 and 1. This may produce unintended switching of the input resulting in excessive current draw.

Workaround:

An external pull-up resistor should be used if the system design requires that a GPIO input be biased to a logical 1 after a reset. If a pin is unused or there is no requirement for the pin to be a High after reset, software may enable the internal pull-down resistors to prevent excess current draw on the pad. Note that the internal bias resistors are not intended to replace external resistors for open-drain type applications.

Silicon Revision Affected:

C0

4.2 PB6, PC5, and PC6 are not 5-V tolerant

Description:

The pins associated with GPIO signals PB6, PC5, and PC6 are not 5-V tolerant. Applying a voltage to any of these pins that is greater than V_{DD} (3.3 V) will have undetermined results.

Workaround:

No workaround is available.

Silicon Revision Affected:

C0

4.3 GPIO input pin latches in the Low state if pad type is open drain

Description:

GPIO pins function normally if configured as inputs and the open-drain configuration is disabled. If open drain is enabled while the pin is configured as an input using the **GPIO Alternate Function Select (GPIOAFSEL)**, **GPIO Open Drain Select (GPIOODR)**, and **GPIO Direction (GPIODIR)**

registers, then the pin latches Low and excessive current (into pin) results if an attempt is made to drive the pin High. The open-drain device is not controllable.

A GPIO pin is not normally configured as open drain and as an input at the same time. A user may want to do this when driving a signal out of a GPIO open-drain pad while configuring the pad as an input to read data on the same pin being driven by an external device. Bit-banging a bidirectional, open-drain bus (for example, I^2C) is an example.

Workaround:

If a user wants to read the state of a GPIO pin on a bidirectional bus that is configured as an open-drain output, the user must first disable the open-drain configuration and then change the direction of the pin to an input. This precaution ensures that the pin is never configured as an input and open drain at the same time.

A second workaround is to use two GPIO pins connected to the same bus signal. The first GPIO pin is configured as an open-drain output, and the second is configured as a standard input. This way the open-drain output can control the state of the signal and the input pin allows the user to read the state of the signal without causing the latch-up condition.

Silicon Revision Affected:

C0, C2

4.4 GPIO pins may glitch during power supply ramp up

Description:

Upon completing a POR (power on reset) sequence, the GPIO pins default to a tri-stated input condition. However, during the initial ramp up of the external V_{DD} supply from 0.0 V to 3.3 V, the GPIO pins are momentarily configured as output drivers during the time the internal LDO circuit is also ramping up. As a result, a signal glitch may occur on GPIO pins before both the external V_{DD} supply and internal LDO voltages reach their normal operating conditions. This situation can occur when the V_{DD} and LDO voltages ramp up at significantly different rates. The LDO voltage ramp-up time is affected by the load capacitance on the LDO pin, therefore, it is important to keep this load at a nominal 1 µF value as recommended in the data sheet. Adding significant more capacitance loading beyond the specification causes the time delay between the two supply ramp-up times to grow, which possibly increases the severity of the glitching behavior.

Workaround:

Ensuring that the V_{DD} power supply ramp up is a fast as possible helps minimize the potential for GPIO glitches. Follow guidelines for LDO pin capacitive loading documented in the electrical section of the data sheet. System designers must ensure that, during the V_{DD} supply ramp-up time, possible GPIO pin glitches can cause no adverse effects to their systems..

Silicon Revision Affected:

C0, C2

5 Timers

5.1 General-purpose timer Edge Count mode count error when timer is disabled

Description:

When a general-purpose timer is configured for 16-Bit Input Edge Count Mode, the timer (A or B) erroneously decrements by one when the Timer Enable (TnEN) bit in the **GPTM Control (GPTMCTL)** register is cleared (the timer is disabled).

Workaround:

When the general-purpose timer is configured for Edge Count mode and software needs to “stop” the timer, the timer should be reloaded with the current count + 1 and restarted.

Silicon Revision Affected:

C0, C2

5.2 General-purpose timer 16-bit Edge Count mode does not load reload value

Description:

In Edge Count 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:

C0, C2

6 General-Purpose Timer

6.1 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 when the lower 16 bits match, regardless of the value of the upper 16 bits.

Workaround:

None.

Silicon Revision Affected:

C0, C2

7 WDT

7.1 Watchdog Timer reset does not work when the PLL is enabled to drive the system clock

Description:

The Watchdog Timer does not properly reset the device when the PLL is enabled to drive the system clock and a time-out condition exists. The Watchdog Timer can be configured to trigger a reset, however, the device does not complete its reset routine, and the device does not come out of reset. The only way to recover from this condition is to assert the external hardware reset (RST) or to perform a Power-On Reset (POR).

Workaround:

If the PLL is used to drive the system clock, the Watchdog Timer cannot be used to generate a reset when the timer times out. The Watchdog Timer only resets the device properly when the system clock is not driven from the PLL. The BYPASS bit in the **Run-Mode Clock Configuration (RCC)** register must be set to bypass the PLL and clock the system directly from the OSC source.

Silicon Revision Affected:

C0

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:

C0, C2

8.2 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:

C0, C2

8.3 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:

C0, C2

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:

C0, C2

Fixed:

Not yet fixed.

10 PWM

10.1 PWM pulses cannot be smaller than dead-band time

Description:

The dead-band generator in the PWM module has undesirable effects when receiving input pulses from the PWM generator that are shorter than the dead-band time. For example, providing a 4-clock-wide pulse into the dead-band generator with dead-band times of 20 clocks (for both rising and falling edges) produces a signal on the primary (non-inverted) output that is High except for 40 clocks (the combined rising and falling dead-band times), and the secondary (inverted) output is always Low.

Workaround:

User software must ensure that the input pulse width to the dead-band generator is greater than the dead-band delays.

Silicon Revision Affected:

C0, C2

10.2 PWM interrupt clear misses in some instances

Description:

It is not possible to clear a PWM generator interrupt in the same cycle when another interrupt from the same PWM generator is being asserted. PWM generator interrupts are cleared by writing a 1 to the corresponding bit in the **PWM Interrupt Status and Clear (PWMnISC)** register. If a write to clear the interrupt is missed because another interrupt in that PWM generator is being asserted, the interrupt condition still exists, and the PWM interrupt routine is called again. System problems could result if an interrupt condition was already properly handled the first time, and the software tries to handle it again. Note that even if an interrupt event has not been enabled in the **PWM Interrupt and Trigger Enable (PWMnINTEN)** register, the interrupt is still asserted in the **PWM Raw Interrupt Status (PWMnRIS)** register.

Workaround:

In most instances, performing a double-write to clear the interrupt greatly decreases the chance that the write to clear the interrupt occurs on the same cycle as another interrupt. Because each generator has six possible interrupt events, writing the **PWMnISC** register six times in a row guarantees that the interrupt is cleared. If the period of the PWM is small enough, however, this method may not be practical for the application.

Silicon Revision Affected:

C0, C2

10.3 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:

C0, C2

10.4 PWMINTEN register bit does not function correctly

Description:

In the **PWM Interrupt Enable (PWMINTEN)** register, the `IntPWM0` (bit 0) bit does not function correctly and has no effect on the interrupt status to the ARM Cortex-M3 processor. This bit should not be used.

Workaround:

PWM interrupts to the processor should be controlled with the use of the **PWM0-PWM2 Interrupt and Trigger Enable (PWMnINTEN)** registers.

Silicon Revision Affected:

C0, C2

10.5 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:

C0, C2

10.6 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:

C0, C2

Texas Instruments Incorporated
108 Wild Basin, Suite 350
Austin, TX 78746
http://www.ti.com/stellaris